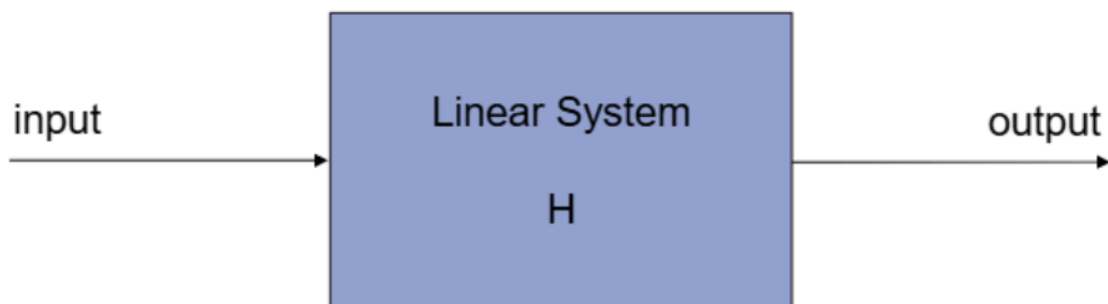


[수치해석] Digital Signal Processing1

Linear system

| Input을 output으로 매핑시켜주는 Syetem



Input: $x_1(t), x_2(t)$, **output:** $H(x_1(t)), H(x_2(t))$ 일 때, 아래 조건을 반드시 만족해야 한다.

$$\alpha x_1(t) + \beta x_2(t) = \alpha H(x_1(t)) + \beta H(x_2(t))$$

Superposition

Linearity를 만족하면 신호 처리에서 각 신호들을 각각 처리하고 나중에 합치는 방식으로 처리할 수 있다.

Signal

| 정보를 담고 있는 물리량의 변화

두 가지 형태가 있다.

1. Countinous signal: 실제 자연 그대로의 신호

- 변수로 표현한다.

2. Sampled signal: 컴퓨터가 이해할 수 있는 신호, Discretization되어 있음

- Sampling된 신호의 Indices로 표현된다.

One-dimensional signal

| 변수가 하나인 경우, 주로 시간에 의해 변화함

Continuous signal : $f(x), u(x), s(t), \dots$

Sampled signal : $u_n, u(n), \dots$

Two-dimensional signal

| 변수가 두개인 경우, 주로 공간 (x, y)에 의해 변화함

Continuous signal : $u(x, y), v(x, y), f(x, y), \dots$

Sampled signal : $u_{m,n}, v(m, n), u(i, j), \dots$

몇몇 Two-dimensional signal은 2D 신호를 1D 신호 두 개로 분리한 **Separable form**으로 나타낼 수 있다.

$$f(x, y) = f(x)f(y)$$

- x방향의 변화와 y 방향의 변화가 서로 독립적이다.

Delta function

| Signal을 다루기 위해 사용하는 함수

Dirac Delta function

| 아날로그 세계에서 Delta function

$$\delta(x, y) = \delta(x)\delta(y)$$

- $x = 0, y = 0$ 일때만 $\delta(x, y) = \infty$

Sampling: 연속 신호에서 특정 시간의 값만 뽑아낸다.

- $\iint f(x', y')\delta(x - x', y - y')dx'dy' = f(x, y)$
- $\lim_{\epsilon \rightarrow 0} \int_{-\epsilon}^{\epsilon} \int_{-\epsilon}^{\epsilon} \delta(x, y)dx dy = 1$

Scailing

- $\delta(ax) = \frac{\delta(x)}{|a|}, \delta(ax, by) = \frac{\delta(x, y)}{|ab|}$

Kronecker Dela function

| 디지털 세계에서 Delta function

$$\delta(m, n) = \delta(m)\delta(n)$$

- $m = 0, n = 0$ 일때만 $\delta(m, n) = 1$

Sampling

$$x(m, n) = \sum_{m'=-\infty}^{\infty} \sum_{n'=-\infty}^{\infty} x(m', n') \delta(m - m', n - n')$$

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(m, n) = 1$$

다른 함수들도 살펴볼 수 있다.

Function	Definition	Function	Definition
<i>Dirac delta</i>	$\delta(x) = 0, x \neq 0$	<i>Rectangle</i>	$\text{rect}(x) = \begin{cases} 1, & x \leq 1/2 \\ 0, & x > 1/2 \end{cases}$
	$\lim_{\epsilon \rightarrow 0} \int_{-\epsilon}^{\epsilon} \delta(x) dx = 1$	<i>Signum</i>	$\text{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$
<i>Sifting property</i>	$\int_{-\infty}^{\infty} f(x') \delta(x - x') dx' = f(x)$	<i>Sinc</i>	$\text{sinc}(x) = \frac{\sin \pi x}{\pi x}$
<i>Scaling property</i>	$\delta(ax) = \frac{\delta(x)}{ a }$	<i>Comb</i>	$\text{comb}(x) = \sum_{n=-\infty}^{\infty} \delta(x - n)$
<i>Kronecker delta</i>	$\delta(n) = \begin{cases} 0, & n \neq 0 \\ 1, & n = 0 \end{cases}$	<i>Triangle</i>	$\text{tri}(x) = \begin{cases} 1 - x , & x \leq 1 \\ 0, & x > 1 \end{cases}$
<i>Sifting property</i>	$\sum_{m=-\infty}^{\infty} f(m) \delta(n - m) = f(n)$		

Sampled signal을 Linear system에서 처리하는 방법을 생각해보자.



수식으로 나타내면 다음과 같다.

$$ay(m, n) = aH[x(m, n)]$$

$x(m, n) = \sum_{m'=-\infty}^{\infty} \sum_{n'=-\infty}^{\infty} x(m', n') \delta(m - m', n - n')$ 이기 때문에 다음과 같이 작성할 수 있다.

$$\begin{aligned}
 ay(m, n) &= aH\left[\sum_{m'=-\infty}^{\infty} \sum_{n'=-\infty}^{\infty} x(m', n')\delta(m - m', n - n')\right] \\
 &= a \sum_{m'=-\infty}^{\infty} \sum_{n'=-\infty}^{\infty} x(m', n')H[\delta(m - m', n - n')]
 \end{aligned}$$

- Superposition에 의해 H 입장에서 m', n' 은 상수 취급이기 때문에 \sum 밖으로 빼낼 수 있다.
- 모든 위치에서의 input 값과 Impulse response를 곱한 값을 더하면 현재 위치에서의 Output이 나온다는 것이다.

위 식에서 $H[\delta(m - m', n - n')]$ 은 Impulse response이다.

- **Impulse:** 특정 순간의 입력
- **Impulse response:** 특정 순간의 입력에 대한 System의 출력
 - $h(m, n; m', n') \equiv H[\delta(m - m', n - n')]$

Shift Invariance

$y(m, n) = H[x(m, n)]$ 이고, 모든 m_0, n_0 에 대해 $y(m - m_0, n - n_0) = H[x(m - m_0, n - n_0)]$ 인 경우이다.

- 즉, 특정 Input pixel의 위치만 옮겼을 때, Output의 값은 동일하고 위치만 변경되는 것이다.

결과적으로 아래와 같아진다.

$$h(m, n; m_0, n_0) = h(m - m_0, n - n_0)$$

- 좌변: 입력 위치가 (m_0, n_0) 일 때의 반응 함수를 구체적으로 정의
- 우변: 어차피 Shift invariance이므로 변수를 $m - m_0, n - n_0$ 으로 사용할 수 있다.
 - $h(m - m_0, n - n_0)$ 은 $h(m, n)$ 을 m_0, n_0 만큼 평행 이동한 것이다.

결국 **Linear Shift Invariant system**에서 신호 처리의 결과는 **Input과 Impulse response의 Convolution의 합**이다.

증명)

1. Linear system에 따르면 Output은 다음과 같다.

$$y(m, n) = H[x(m, n)]$$

2. Kronecker Delta function을 이용하여 나타내면 다음과 같다.

$$y(m, n) = H[\sum \sum x(m', n') \delta(m - m', n - n')]$$

3. Superposition에 의해 다음과 같이 쓸 수 있다.

$$y(m, n) = \sum \sum x(m', n') H[\delta(m - m', n - n')] = \sum \sum x(m', n') h(m, n, m', n')$$

4. H라는 System이 **Shift Invariant**하다면 다음과 같이 정리할 수 있다.

$$y(m, n) = \sum \sum x(m', n') h(m - m', n - n')$$

Stability

| 유한한 입력을 넣었을 때, 유한한 출력이 나오는 것

$$|x(m, n)| < \infty \rightarrow |H[x(m, n)]| < \infty$$

LSI System이 Stable할 조건

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |h(m, n)| < \infty$$

- LSI 시스템의 Output 공식에서 $x(m, n)$ 을 상수로 가정하면 위 공식을 얻을 수 있다.

Fourier transform

| 시간과 주파수간의 변환

1D Fourier transform

시간 → 주파수: $F(u) = \int f(x) e^{-j2\pi ux} dx$

주파수 → 시간: $F(x) = \int f(u) e^{j2\pi ux} du$

Discrete-Time Fourier Transform: Discrete한 Sample 데이터에서 연속적인 Frequency를 뽑아냄.

$$\begin{aligned} x(n) &= \int_{-0.5}^{0.5} X(u) \exp(j2\pi nu) du \\ \Updownarrow \\ X(u) &= \sum_{n=-\infty}^{\infty} x(n) \exp(-j2\pi nu), \quad -0.5 \leq u < 0.5 \end{aligned}$$

- 어차피 주기를 갖기 때문에 $-0.5 \sim 0.5$ 구간만 계산하여 전체 영역에 복사하면 된다.

2D Fourier transform

시간 → 주파수: $F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(-j2\pi(ux + vy)) dx dy$

주파수 → 시간: $f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp(j2\pi(xu + yv)) du dv$

Discrete-Time Fourier Transform

$$\begin{aligned} x(m, n) &= \int_{-0.5}^{0.5} \int_{-0.5}^{0.5} X(u, v) \exp(j2\pi(mu + nv)) du dv \\ \Updownarrow \\ X(u, v) &= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(m, n) \exp(-j2\pi(mu + nv)), \quad -0.5 \leq u, v < 0.5 \end{aligned}$$

- $X(u, v)$ 는 주기가 1인 주기함수이다.
- $X(u, v)$ 가 존재하기 위해선 아래 조건을 만족해야 한다.

$$\begin{aligned} |X(u, v)| &= \left| \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(m, n) \exp(-j2\pi(mu + nv)) \right| \\ &\leq \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |x(m, n)| |\exp(-j2\pi(mu + nv))| \leq \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |x(m, n)| < \infty \end{aligned}$$

Frequency domain

Fourier transformation을 이용하여 **Time domain**을 **Frequency domain**으로 변경할 수 있다.

- Frequency domain에서는 변화량이 클 수록 frequency가 크다.

$$f(t) \rightarrow F(w)$$

- w : 각주파수

$$f(x, y) \rightarrow F(u, v)$$

- u, v : Spatial frequency
- 공간적 좌표에 따라 밝기가 어떻게 변하는지
- Edge는 High spatial frequency이다.

특징

Uniqueness

- $f(x, y)$ 와 대응되는 $F(u, v)$ 는 유일하다.

Separability: 2d Transformation을 적분 2번으로 각각 나눠서 할 수 있다.

$$F(u, v) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(x, y) \exp(-j2\pi ux) dx \right] \exp(-j2\pi vy) dy$$

Convolution Theorem: 공간 영역에서의 **Convolution**이 주파수 영역에서의 곱과 동일하다.

$$g(x, y) = h(x, y) * f(x, y) \iff G(u, v) = H(u, v)F(u, v)$$

- 주파수 영역에서의 Covolution은 상수*공간 영역에서의 곱과 동일하다.

Inner product preservation: 좌표계를 변환해도 내적의 값은 변화되지 않는다.

- 함수의 표현을 바꿀 뿐, 함수 자체의 기하적 관계는 바꾸지 않는다.

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) h^*(x, y) dx dy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) H^*(u, v) du dv$$

- $\mathbf{h} = \mathbf{f}$ 인 상황에서도 동일하다.

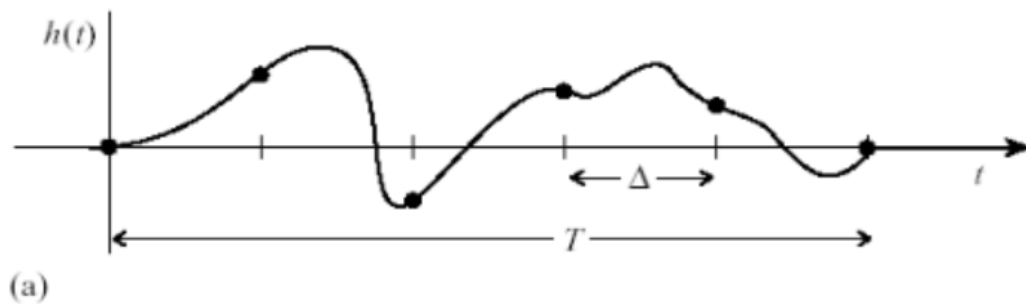
$$\int |f(x, y)|^2 dx dy = \int |F(u, v)|^2 du dv$$

Hankel transform: 2D Fourier Transform을 Cartesian 좌표가 아니라 Polar 좌표로 표현

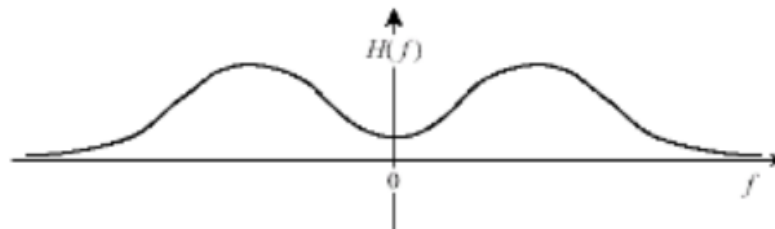
$$\begin{aligned} F_p(\xi, \phi) &\equiv F(\xi \cos \phi, \xi \sin \phi) \\ &= \int_0^{2\pi} \int_0^\infty f_p(r, \theta) \exp[-j2\pi r \xi \cos(\theta - \phi)] r dr d\theta \\ \text{where } f_p(r, \theta) &= f(r \cos \theta, r \sin \theta) \end{aligned}$$

Sampling

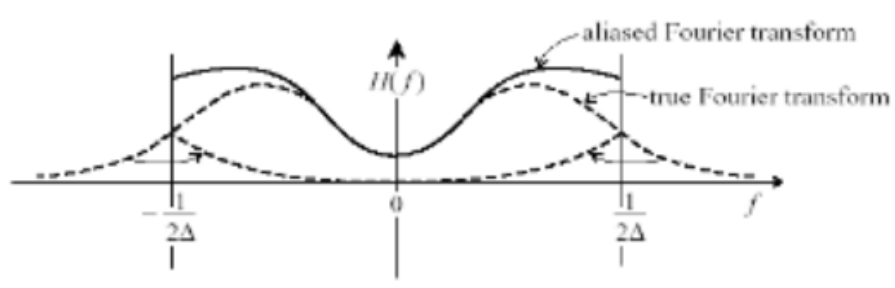
실제 $h(t)$ 에서 Δ 간격으로 Sampling한 것이다.



먼저, $h(t)$ 를 Fourier Transformation 했을 때의 $H(f)$ 는 다음과 같다.



하지만, Sampling Δ 의 간격에 따라 Frequency가 겹칠 수 있다. 이런 경우에 **Aliasing**이 발생한다.



Aliasing

먼저 Signal을 Sampling하면 다음과 같다.

$$h_s(t) = h(t) \cdot \sum_{k=-\infty}^{\infty} \delta(t - k\Delta)$$

Sampling 한 신호를 Fourier transform하면 다음과 같다.

- 시간 축에서의 곱은 주파수 축에서 Convolution이다.

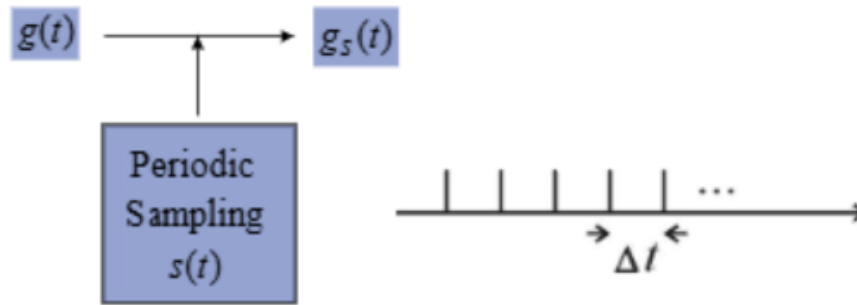
$$H_s(f) = H(f) * \frac{1}{\Delta} \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{\Delta})$$

- 즉, 동일한 $H(f)$ 가 $\frac{1}{\Delta}$ 마다 반복된다.

Δ 가 충분히 작지 않다면 Frequency가 겹쳐 Aliasing이 발생한다.

Sampling theory

먼저 Signal에서 Periodic Sampling을 이용하여 Sampling한다.

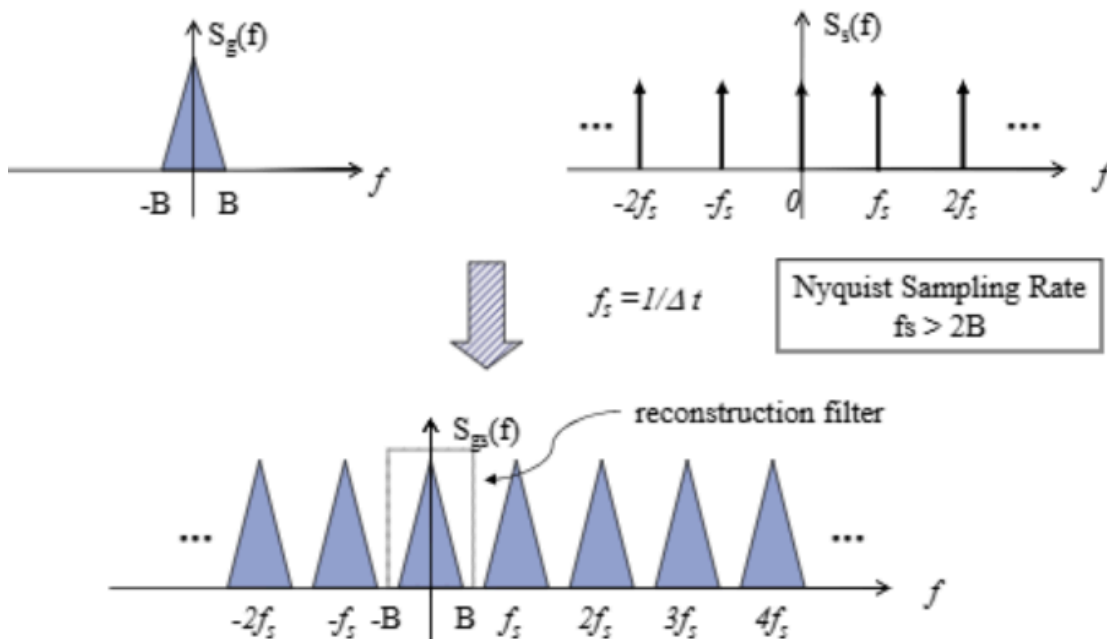


- 이때 사용하는 $s(t) = \sum_m \delta(t - m\Delta t)$

결국 Sampling 후 Signal $g_s(t)$ 는 $g(t) \cdot s(t)$ 이다.

Signal $g_s(t)$ 를 Fourier transformation하면 $S_{g_s}(f) = S_g(f) * S_s(f)$ 이다.

이는 곧 $S_g(f)$ 가 $\frac{1}{\Delta t}$ 마다 반복된다는 것이다.



- 여기서 $[-B, B]$ 는 원래의 Signal이 그 범위 내의 Frequency만 가진다고 가정하기 때문이다.

Aliasing은 두 Frequencies가 겹쳐야 발생하기 때문에, $f_s > 2B$ 라면 Aliasing이 발생하지 않는다.

- $f_s = \frac{1}{\Delta t} > 2B$
- **Nyquist sampling rate**

Sampling으로 인한 여러 복제본이 생겼을 때, 원본에 대응되는 부분만 잘라내는 것이 **Reconstruction filter**이다.

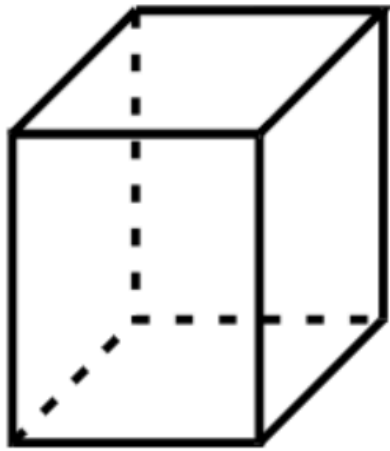
전체 과정은 다음과 같다.

1. 실세계의 신호를 컴퓨터가 이해할 수 있도록 **Sampling**한다.
 2. Sampling된 신호의 주파수 특성을 분석하기 위해 **Fourier transform**을 사용한다.
 - 신호를 주파수 영역에서 이해하고 조작하기 위함이다.
 - 신호 영역에서 Convolution을 주파수 영역에서 곱셈으로 다룰 수 있기에 계산 효율적이다.
 - LSI 시스템을 신호 영역보다 주파수 영역에서 정의하는 것이 효율적이다.
 3. 주파수 영역에서 복사된 스펙트럼 중 원래 신호에 대응되는 주파수 대역만 남기기 위해 **reconstruction filter**를 적용한다.
-

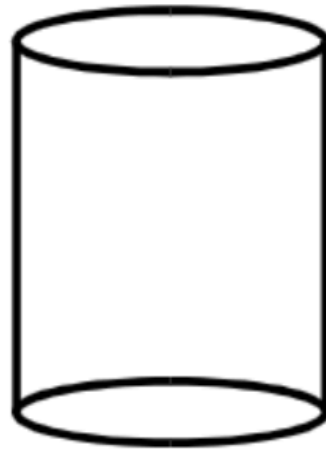
Reconstruction filter

샘플링 과정에서 주파수 영역에 무한히 복제된 스펙트럼들(Images) 중, 원래 신호에 해당하는 기본 대역 하나만 남기고 나머지는 모두 제거하는 필터

rectangular filter



cylindrical filter



$$h_r(x, y) = K \left(\frac{\sin \omega_x x}{\omega_x x} \right) \left(\frac{\sin \omega_y y}{\omega_y y} \right)$$
$$h_c(x, y) = \frac{2\pi\omega_0 J_1(\omega_0 \sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}}$$

Aliasing effect

2D image에서의 Aliasing effect를 살펴보자.

만약 $\Delta u < \Delta u_{Nyquist}$ 이거나 $\Delta v < \Delta v_{Nyquist}$ 이라면 Aliasing으로 인하여 이미지가 원래대로 복원이 되지 않을 것이다.

Aliasing로 인하여 원래 없던 패턴이 생기기도 한다.

- 원래 이미지에서 **촘촘히** 모여 있는 구간을 완벽히 복원하기 위해선 **Sampling** 간격이 **좁아야** 한다.
- 고주파 영역에서 Sampling rate가 더욱 중요하다.

- Frequency를 Sampling 횟수가 따라가지 못하면 중간 패턴을 보지 못 한다.
-

Limitations in Sampling

1. 실세계의 이미지는 Band-limited한 Frequency를 가지지 않는다.

- Sampling freq > 2B 조건을 만족시키기 어렵다.

해결 방법: Low pass filter

- High frequency가 문제이므로 Sampling 전에 Low-pass filter를 통해 High frequency 부분을 조정하고 Sampling하는 것이 도움이 된다.
- LPF(Low-pass Filter)는 High frequency 정보를 약화시킨다.
- 그러나 이 방법은 resolution loss를 유발하고 Blurring을 유발한다.

2. 현실 세계에서 이상적인 Low pass filter가 존재하지 않는다.