CMPE 362
Digital Image Processing

Frequency Domain Techniques – Part III

Sampling

Why does a lower resolution image still make sense to us?
What do we lose?

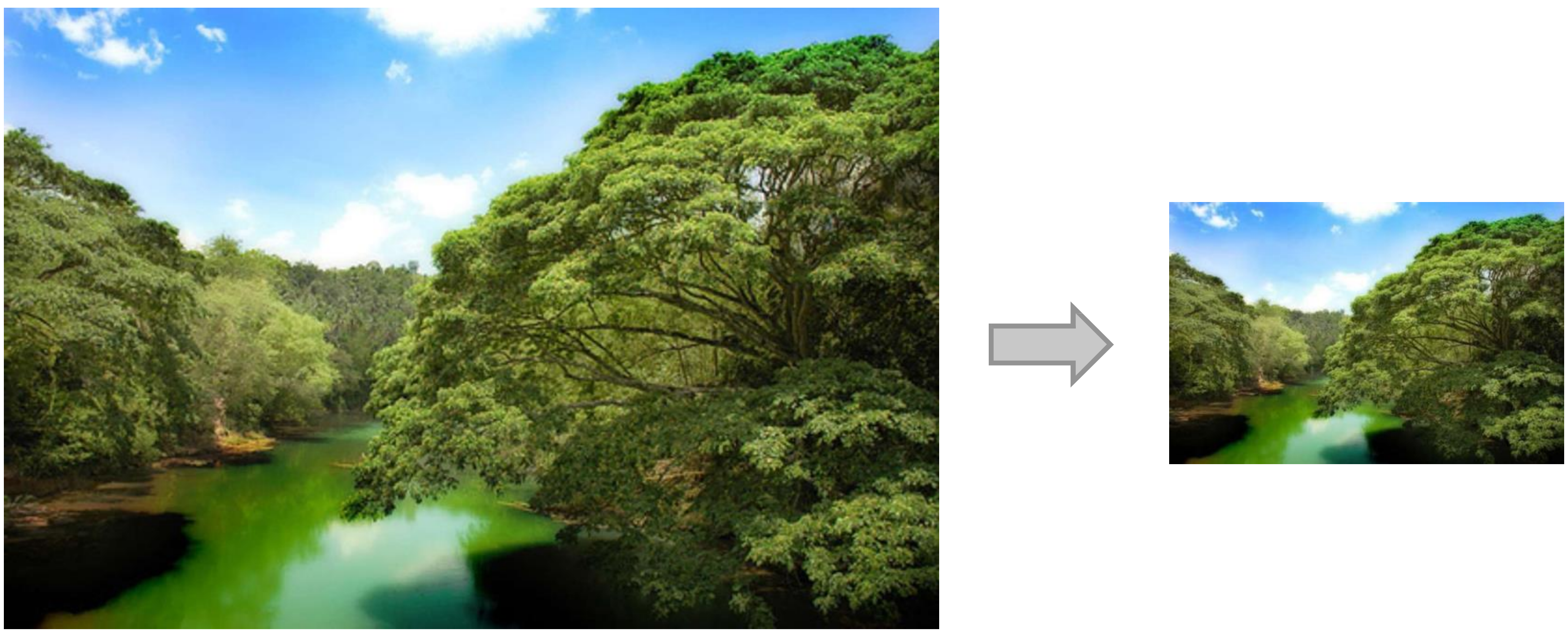
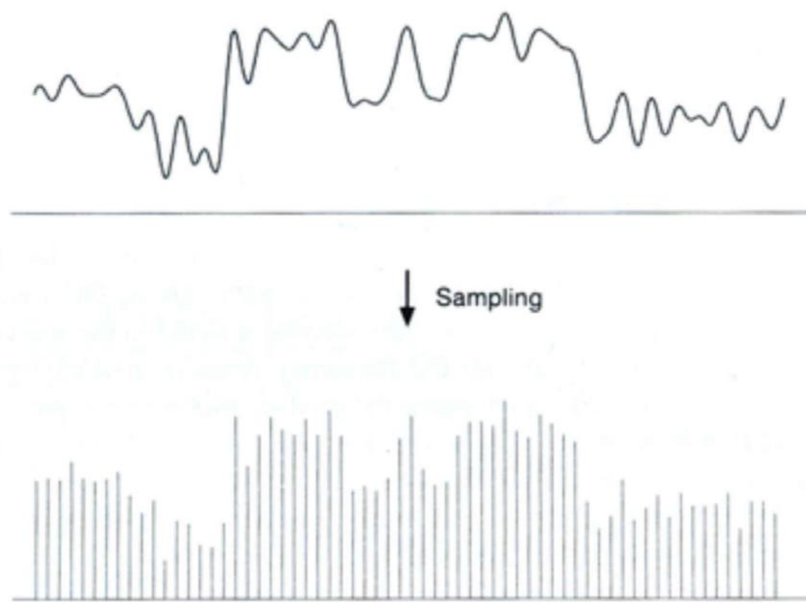


Image: http://www.flickr.com/photos/igorms/136916757/

Slide credit: D. Hoiem

Sampled representations

- How to store and compute with continuous functions?
- Common scheme for representation: samples
  – write down the function's values at many points

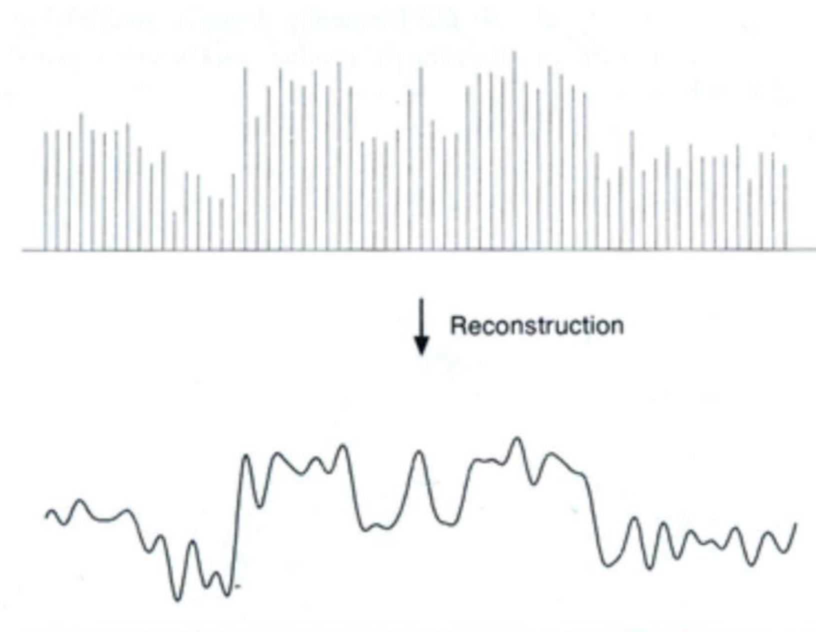


Slide credit: S. Marschner

Reconstruction

- Making samples back into a continuous function

– amounts to "guessing" what the function did in between

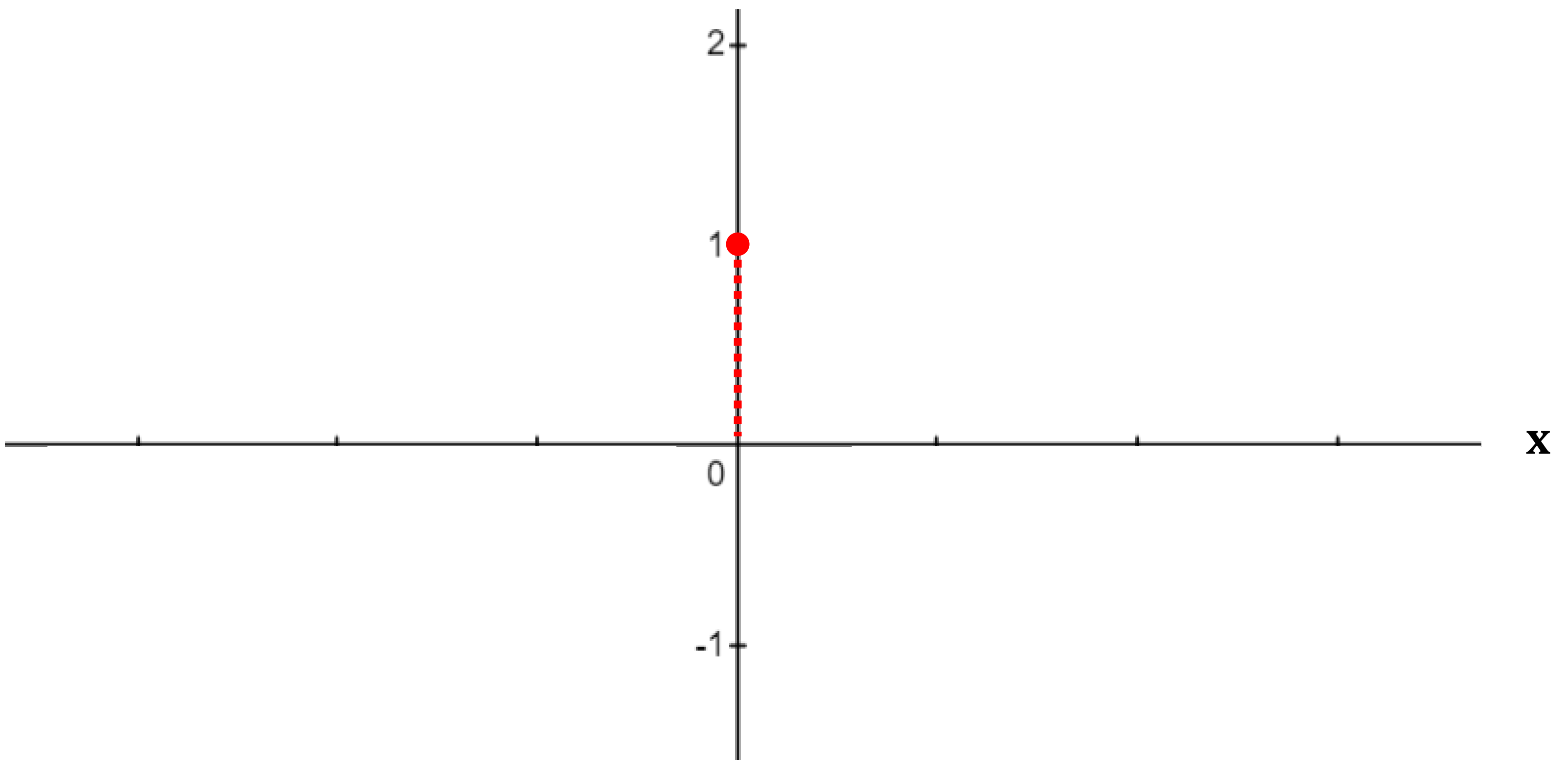


Slide credit: S. Marschner

Impulse function $\delta(x)$

$\delta(x) = 1 \quad \text{if } x = 0$

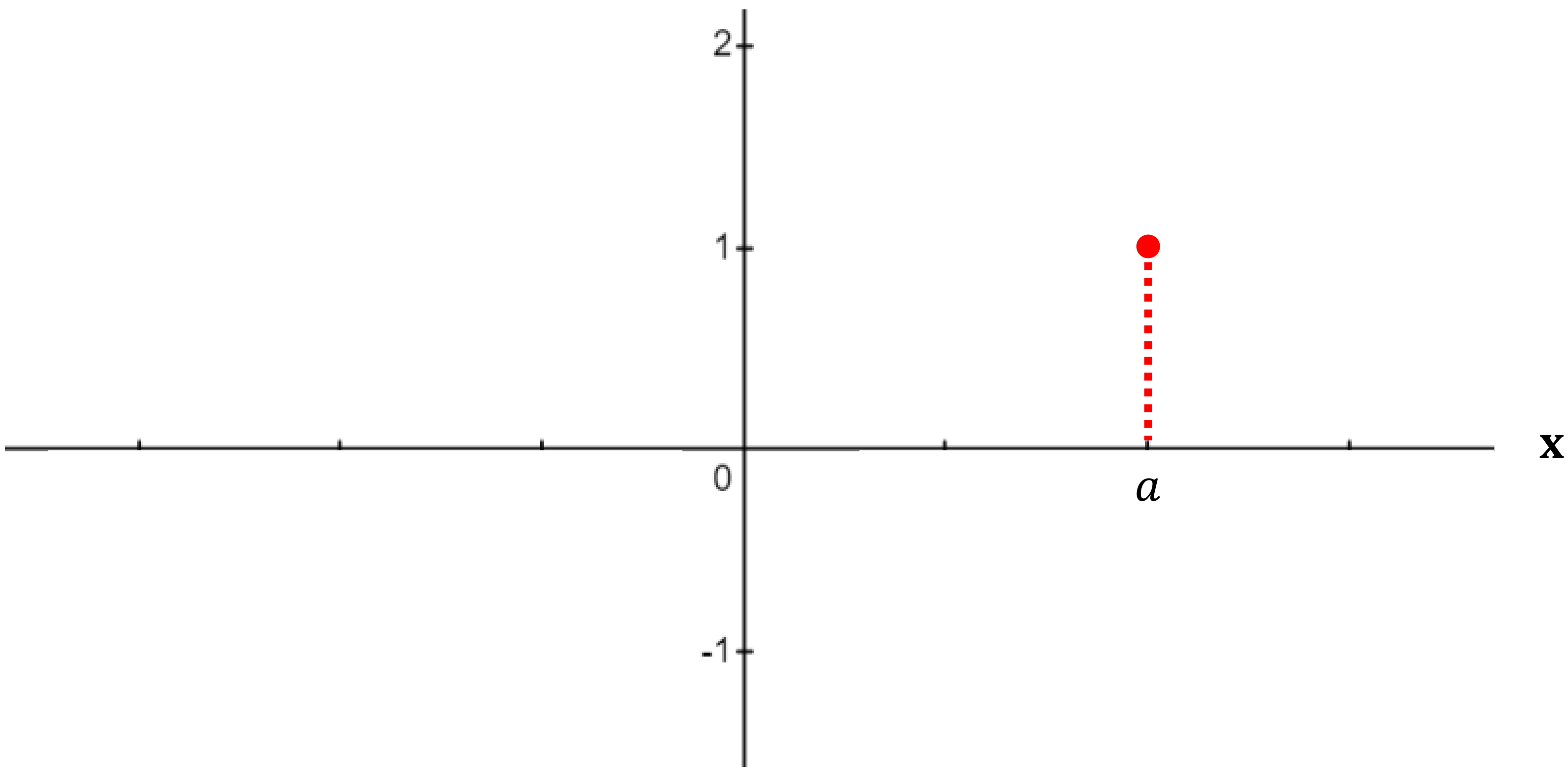
$\delta(x) = 0 \quad \text{otherwise}$



Impulse function $\delta(x - a)$

$\delta(x - a) = 1 \quad \text{if } x = a$

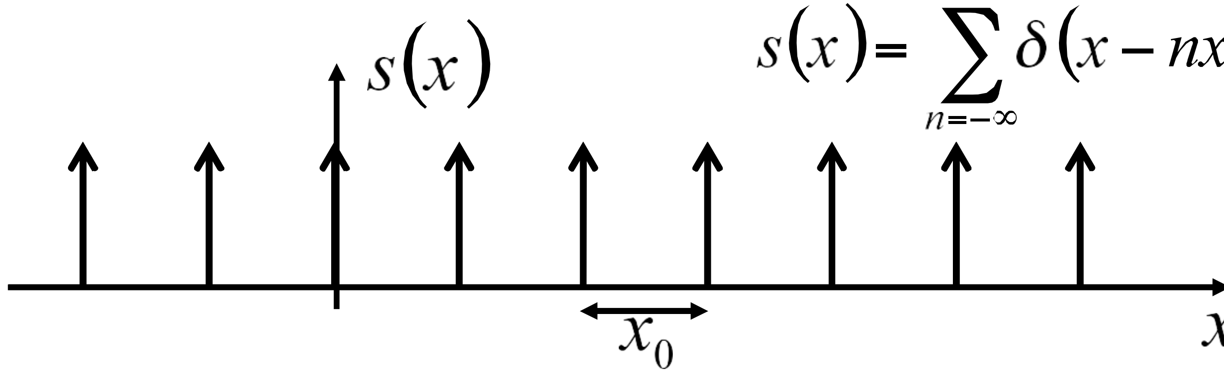
$\delta(x - a) = 0 \quad \text{otherwise}$



Impulse train

Shah function (Impulse train):

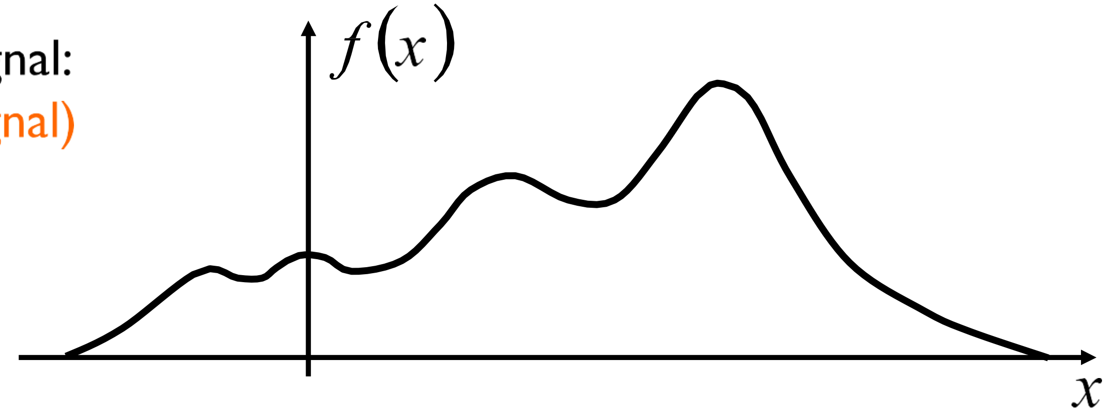
$$s(x) = \sum_{n=-\infty}^{\infty} \delta(x - nx_0)$$



Sampling Theorem

Continuous signal:

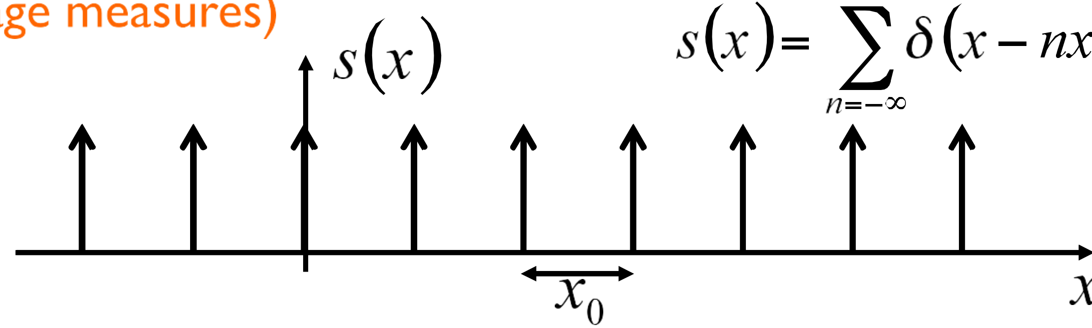
(Real world signal)



Shah function (Impulse train):

(What the image measures)

$$s(x) = \sum_{n=-\infty}^{\infty} \delta(x - nx_0)$$

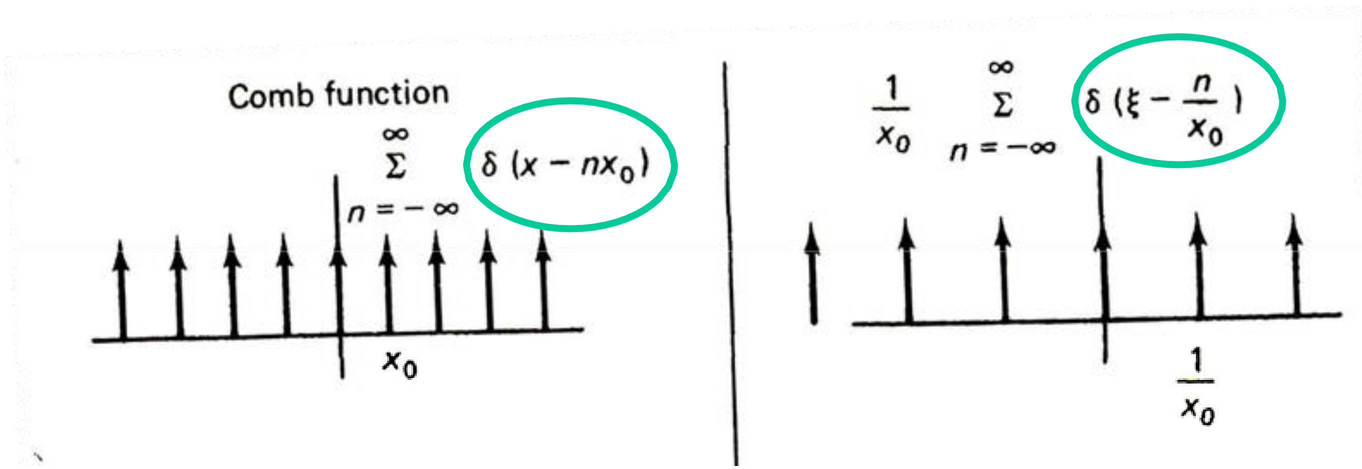


Sampled function:

$$f_s(x) = f(x)s(x) = f(x)\sum_{n=-\infty}^{\infty} \delta(x - nx_0)$$

Slide credit: S. Narasimhan

FT of an "impulse train" is an impulse train!



Slide credit: S. Narasimhan

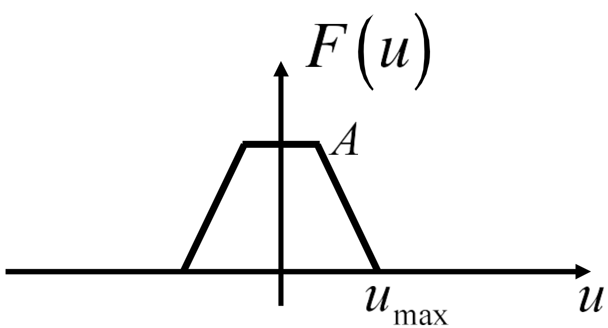
Sampling Theorem

Sampled function:

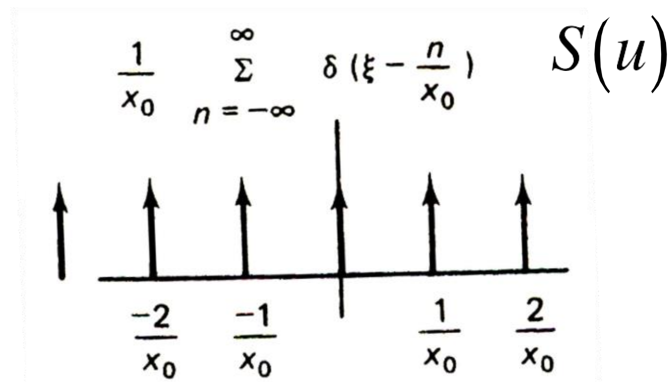
Sampling frequency $\frac{1}{x_0}$

$$f_s(x) = f(x)s(x) = f(x)\sum_{n=-\infty}^{\infty}\delta(x - nx_0)$$

$$F_S(u) = F(u) * S(u) = F(u) * \frac{1}{x_0}\sum_{n=-\infty}^{\infty}\delta\left(u - \frac{n}{x_0}\right)$$



$*$



Slide credit: S. Narasimhan

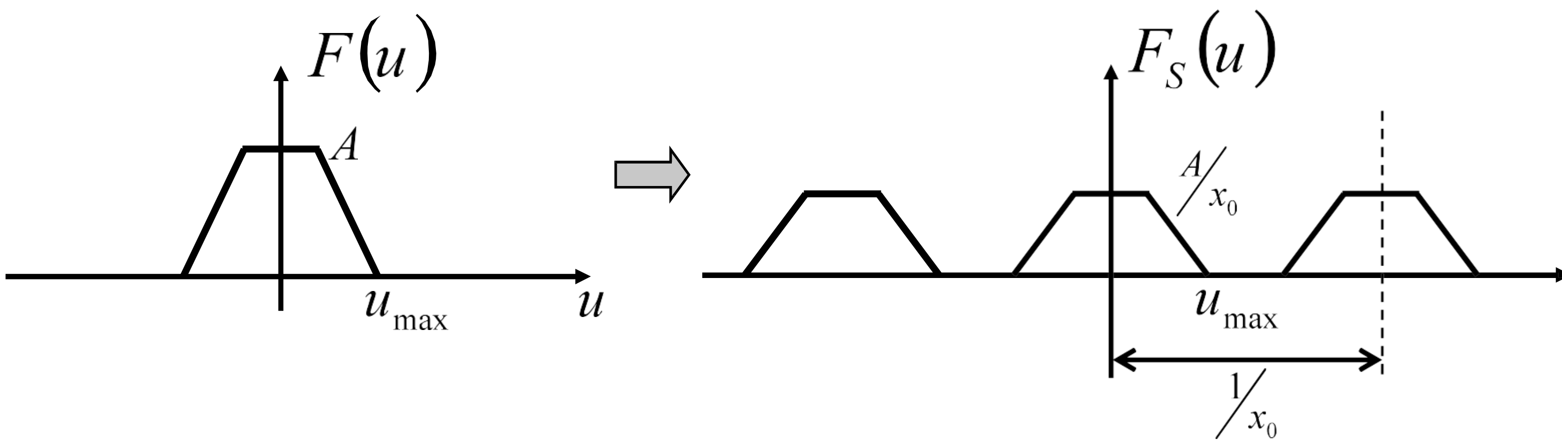
Sampling Theorem

Sampled function:

$$f_s(x) = f(x)s(x) = f(x)\sum_{n=-\infty}^{\infty}\delta(x - nx_0)$$

Sampling frequency $\frac{1}{x_0}$

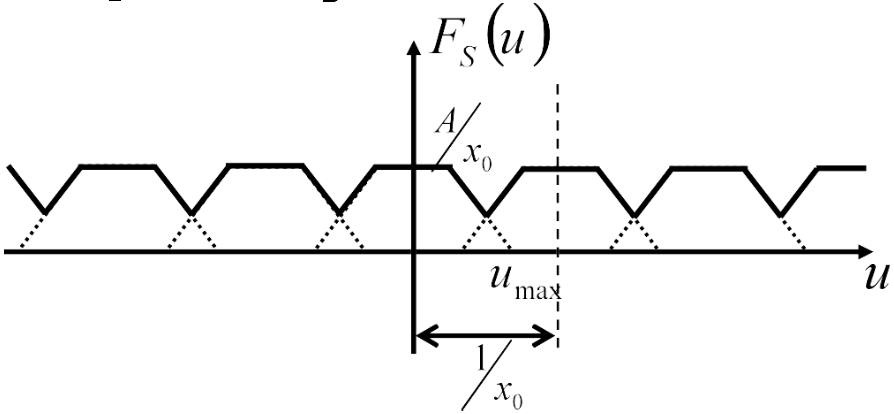
$$F_S(u) = F(u) * S(u) = F(u) * \frac{1}{x_0}\sum_{n=-\infty}^{\infty}\delta\left(u - \frac{n}{x_0}\right)$$



Slide credit: S. Narasimhan

Nyquist Frequency

If $u_{\max} \geq \frac{1}{2x_0}$



Aliasing

When can we recover $F(u)$ from $F_S(u)$?

Only if $u_{\max} < \frac{1}{2x_0}$ (Nyquist Frequency)

We can use

$$C(u) = \begin{cases} x_0 & |u| < 1/2x_0 \\ 0 & \text{otherwise} \end{cases}$$

Then $F(u) = F_S(u)C(u)$ and $f(x) = \mathrm{IFT}[F(u)]$

Sampling frequency must be greater than $2u_{\max}$

Slide credit: S. Narasimhan

Nyquist-Shannon Sampling Theorem

- When sampling a signal at discrete intervals, the sampling frequency must be $> 2 \times f_{max}$
- f_{max} = max frequency of the input signal
- This allows to reconstruct the original perfectly from the sampled version

Slide credit: D. Hoiem

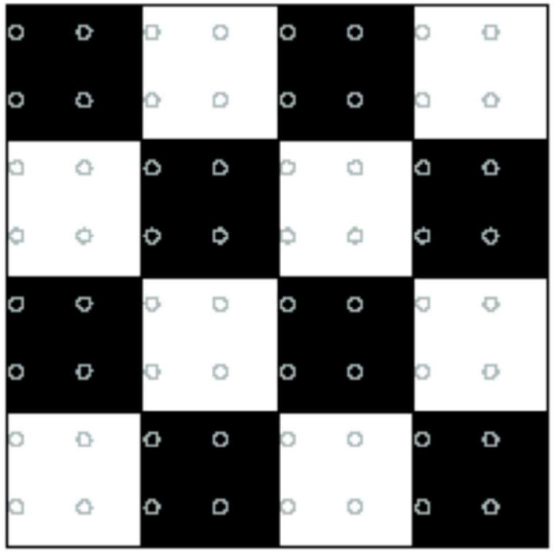
2D example

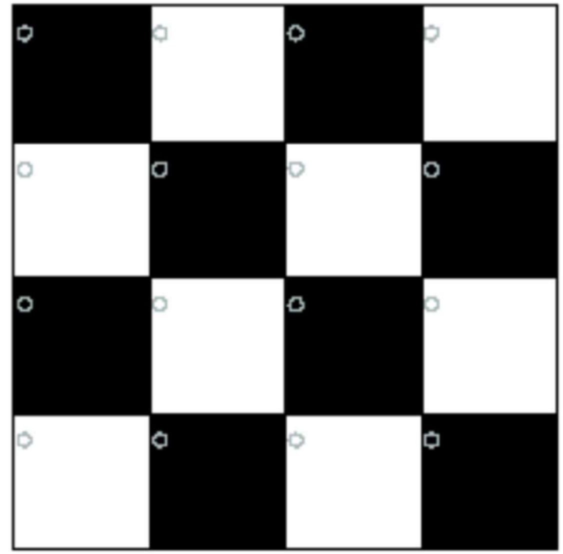
fmax = 1/2

1/x0 = 1/0.5
1/x0 > 2*fmax
1/0.5 > 2*(1/2)

✓

Good sampling





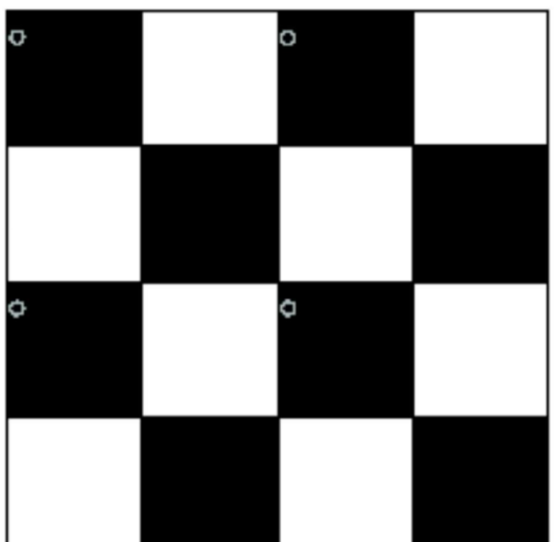
1/x0 = 1/1
1/x0 > 2*fmax
1/1 >= 2*(1/2)

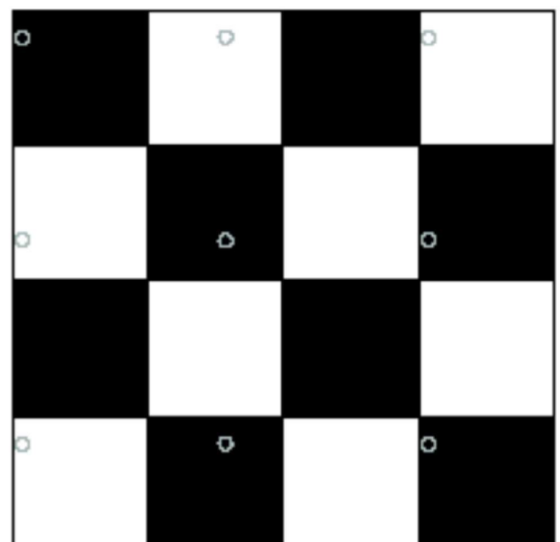
✓

Good sampling

1/x0 = 1/2
1/x0 ≯ 2*fmax
1/2 ≯ 2*(1/2)

Bad sampling





1/x0 = 1/1.5
1/x0 ≯ 2*fmax
1/1.5 ≯ 2*(1/2)

Bad sampling

Slide credit: N. Kumar

Aliasing problem

- Sub-sampling may be dangerous….
- Characteristic errors may appear:
  - "Wagon wheels rolling the wrong way in movies"
  - "Checkerboards disintegrate in ray tracing"
  - "Striped shirts look funny on color television"

Slide credit: D. Forsyth

Aliasing examples

Check out Moire patterns on the web.

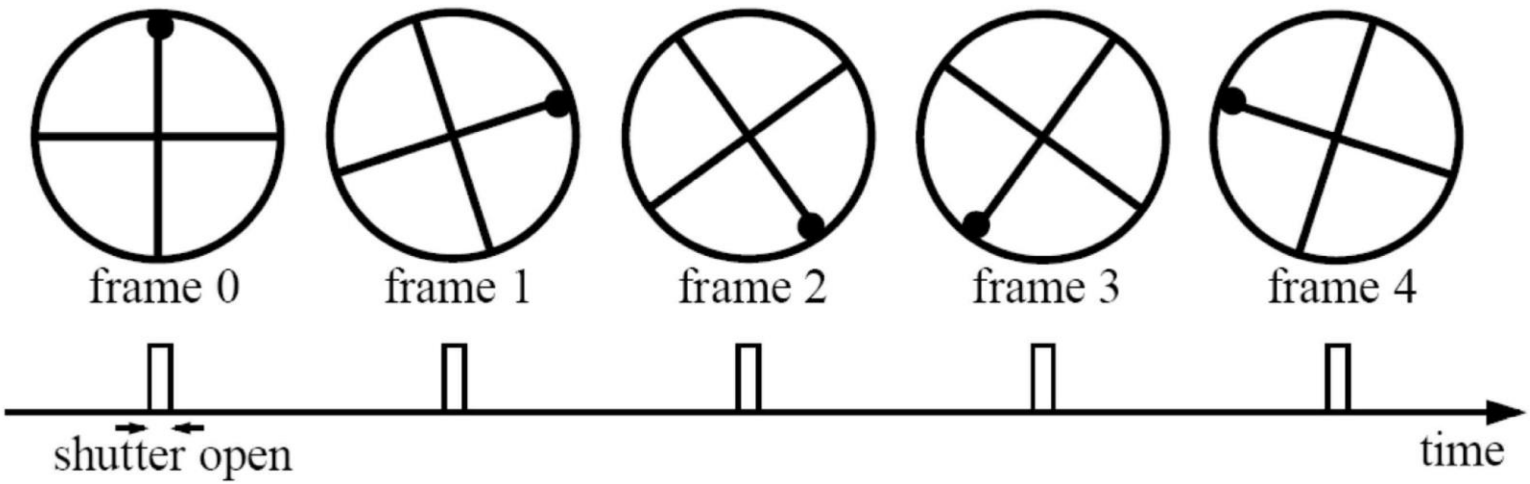
Slide credit: A. Farhadi

Aliasing in video

fmax = 1/90
1/x0 > 2*(1/90)
x0 < 45

Imagine a spoked wheel moving to the right (rotating clockwise). Mark wheel with dot so we can see what's happening.

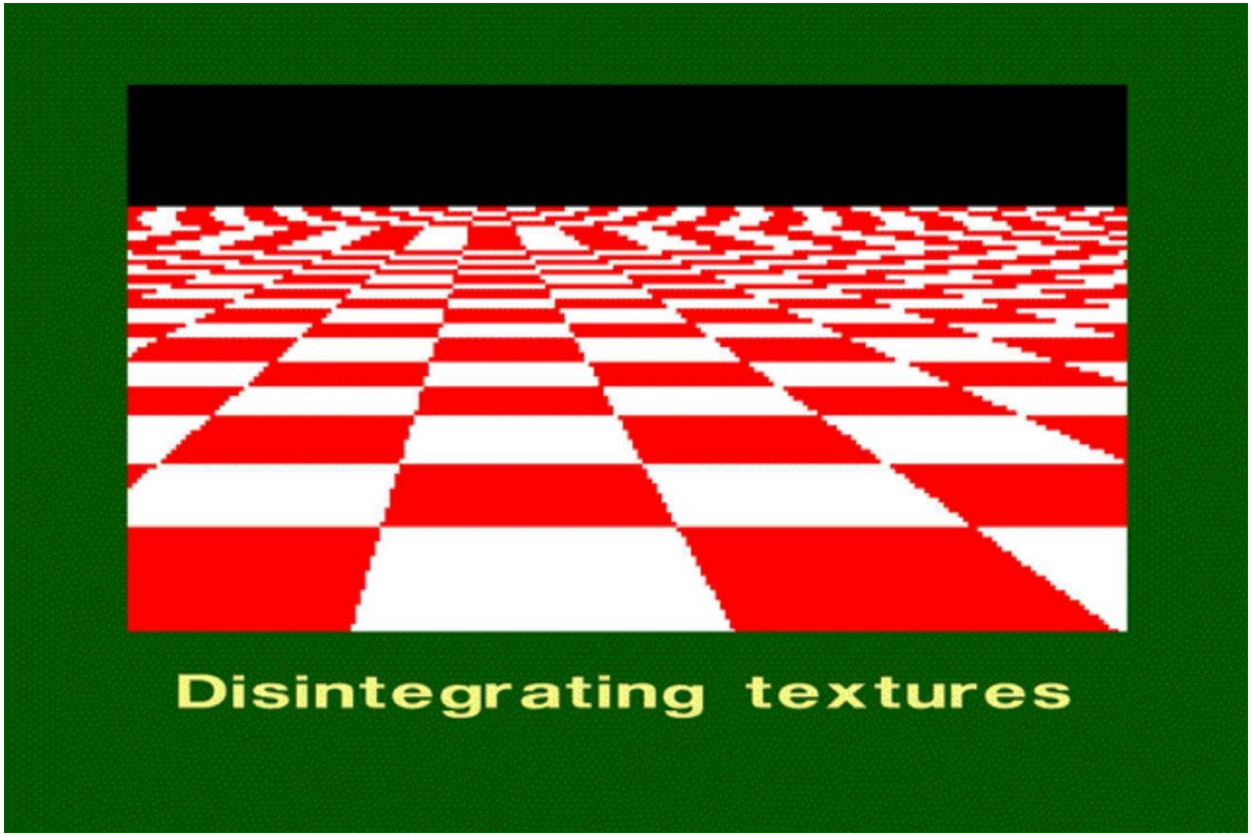
If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):



Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

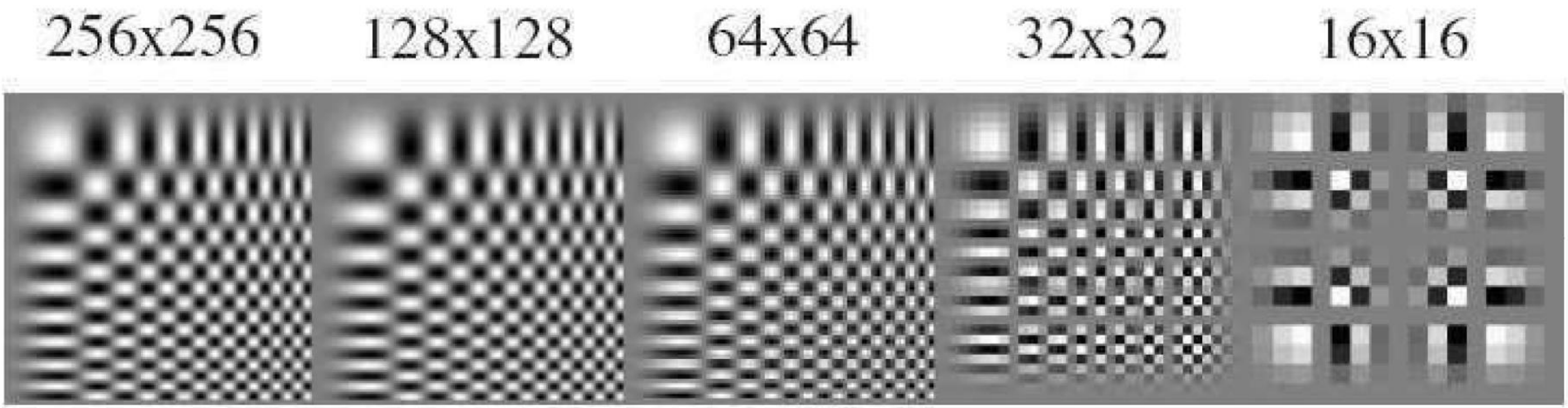
Slide credit: S. Seitz

Aliasing in graphics



Slide credit: A. Efros

Sampling and aliasing



Slide credit: D. Hoiem

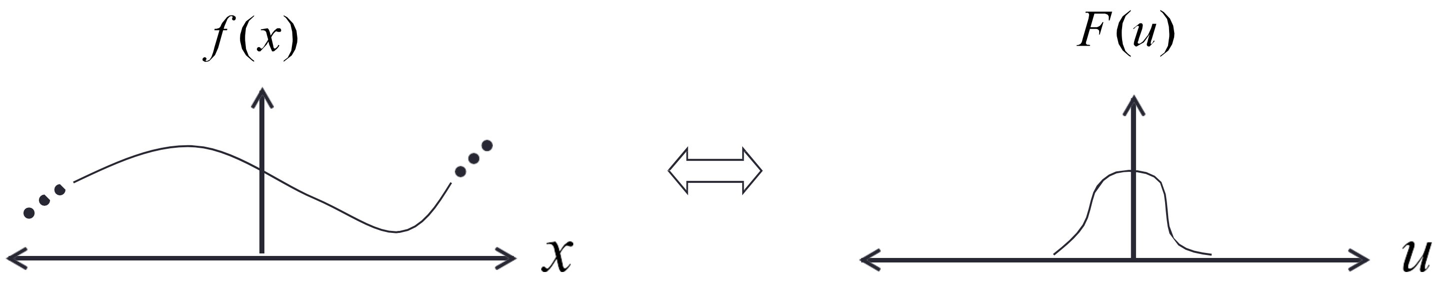
Anti-aliasing

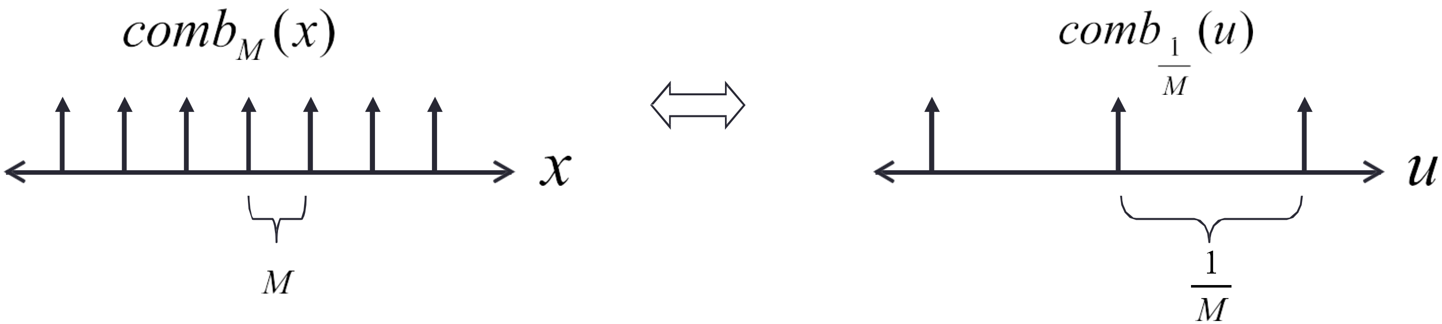
Solutions:

- Sample more often

- Get rid of all frequencies that are greater than half the new sampling frequency
  - Will lose information
  - But it's better than aliasing
  - Apply a smoothing filter

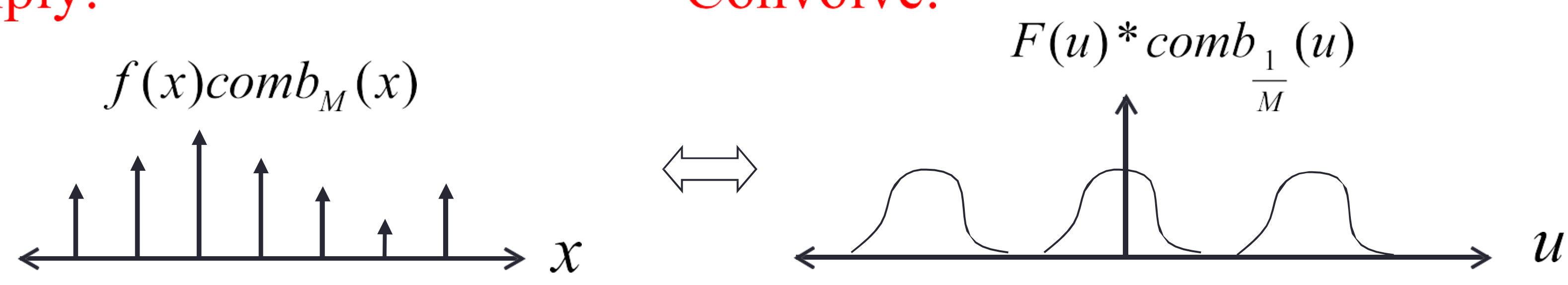
Sampling low frequency signal





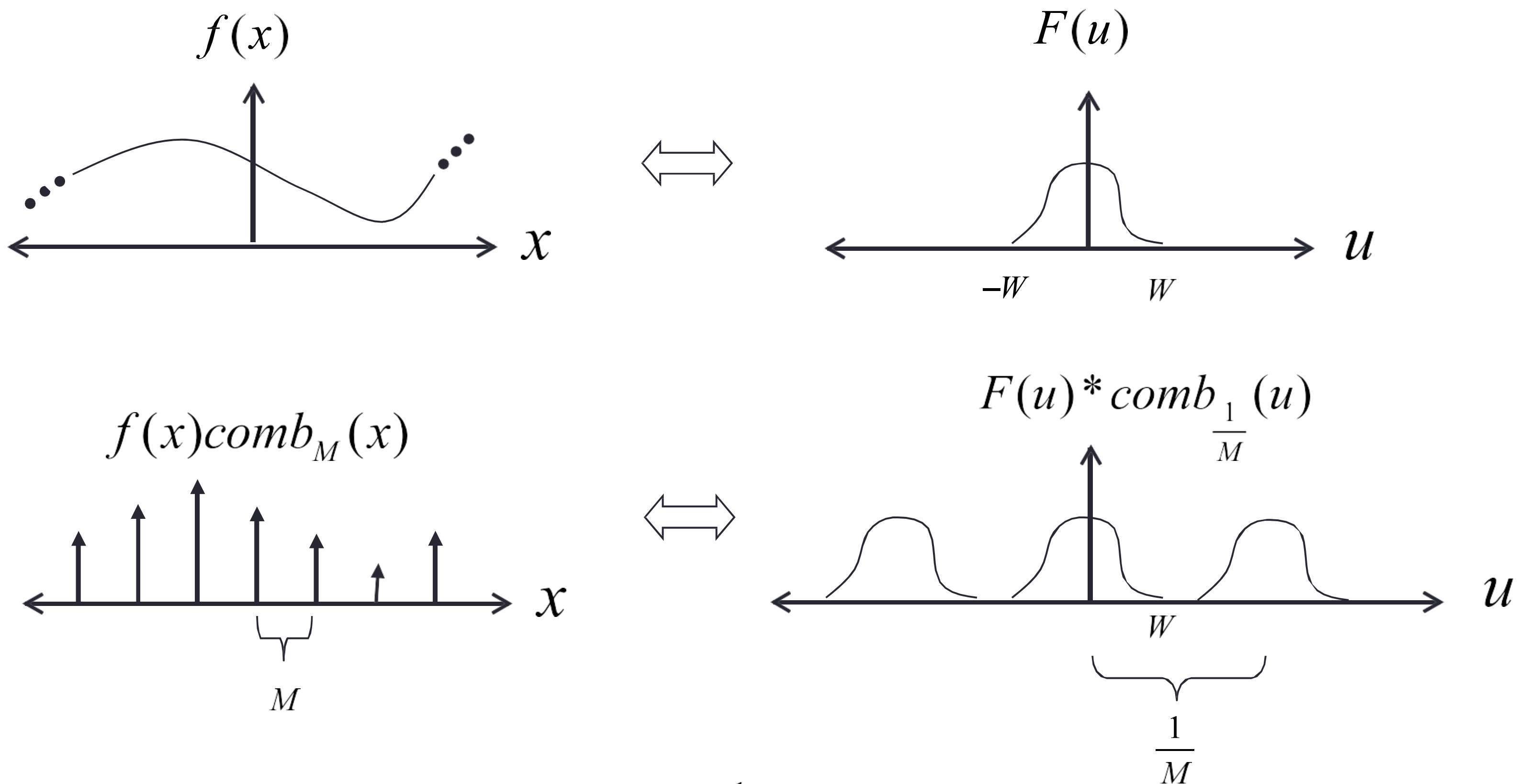
Multiply:

Convolve:



Slide credit: B. K. Gunturk

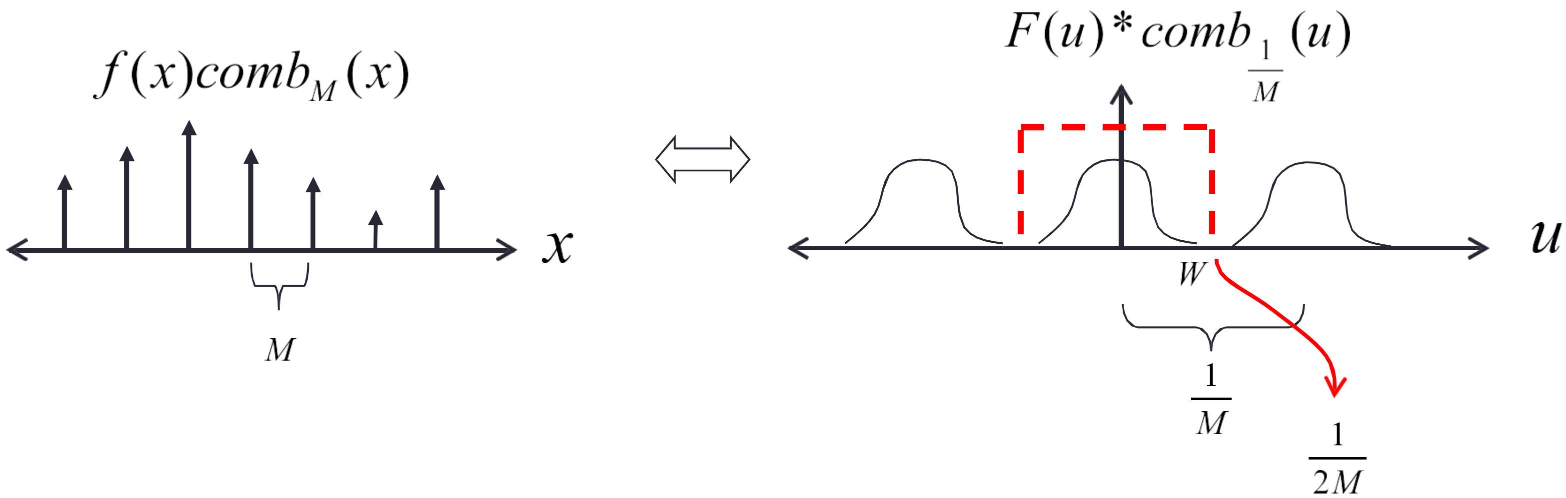
Sampling low frequency signal



"No problem" if $\dfrac{1}{M} > 2W$

Slide credit: B. K. Gunturk

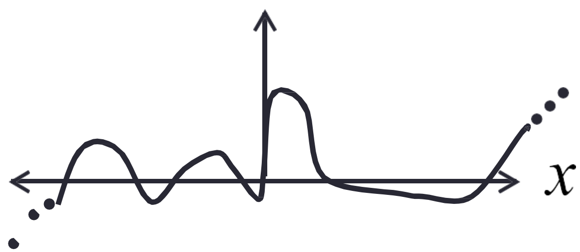
Sampling low frequency signal



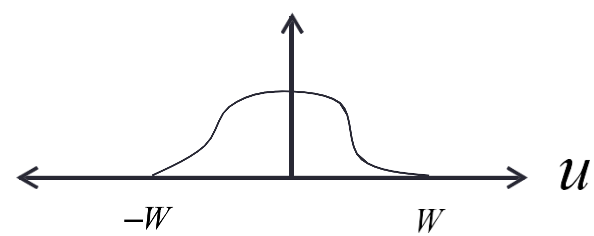
If there is no overlap, the original signal can be recovered from its samples by low-pass filtering.

Slide credit: B. K. Gunturk

Sampling high frequency signal

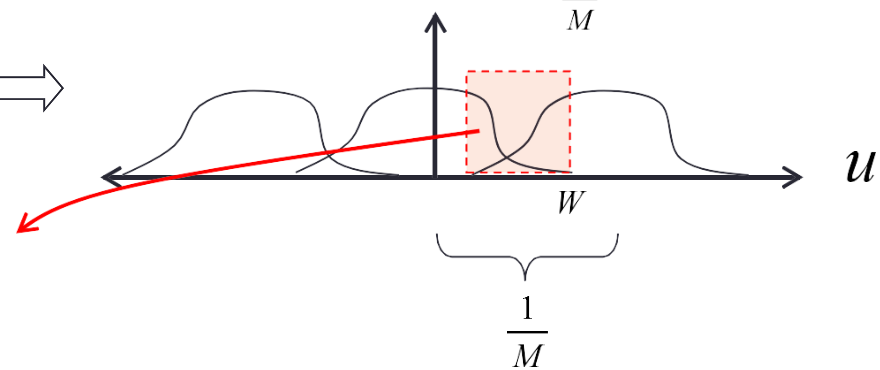


$\Leftrightarrow$



$$f(x)comb_M(x)$$

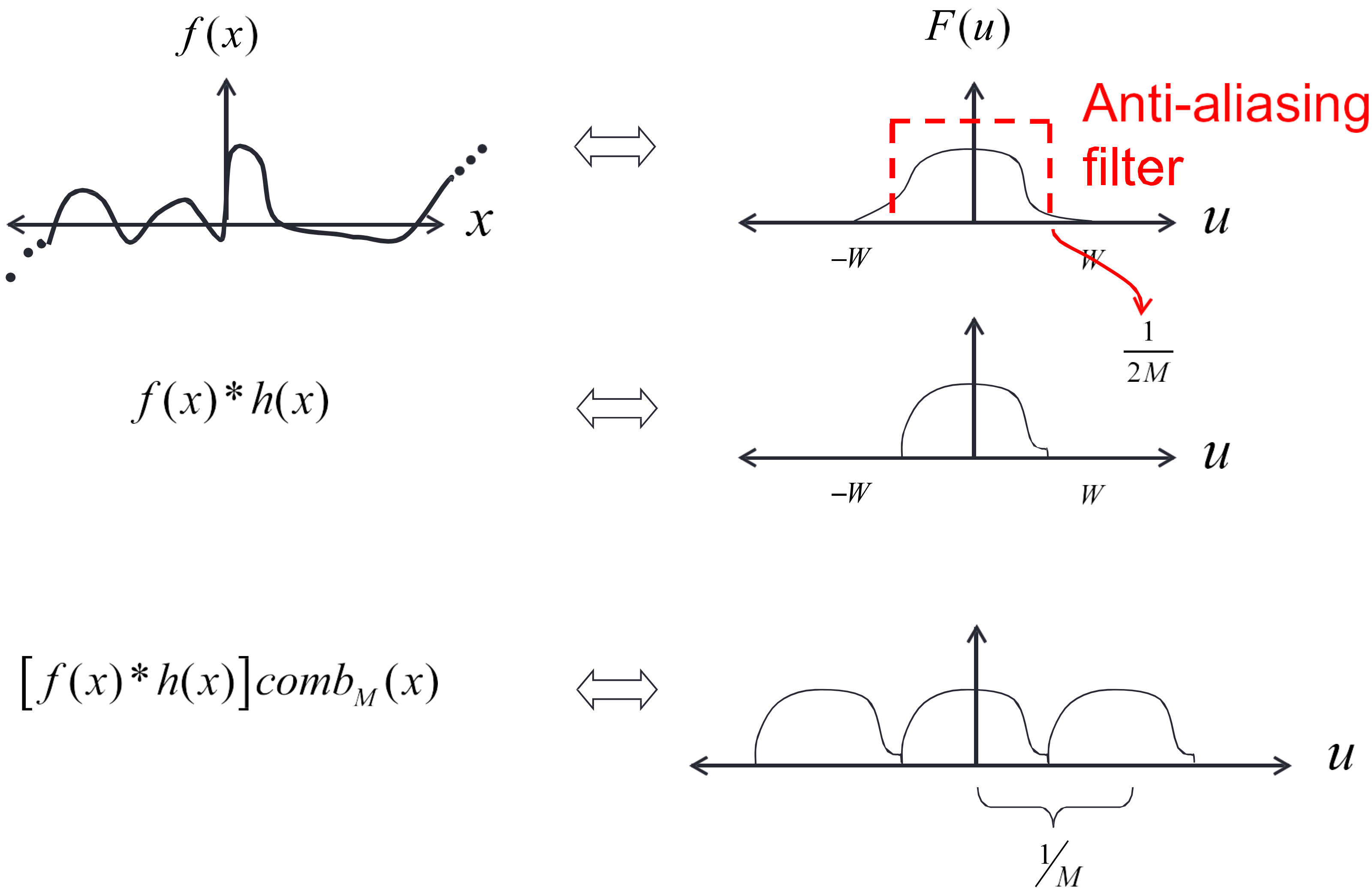
$\Leftrightarrow$



Overlap: The high frequency energy is folded over into low frequency. It is "aliasing" as lower frequency energy. And you cannot fix it once it has happened.

Slide credit: B. K. Gunturk

Sampling high frequency signal

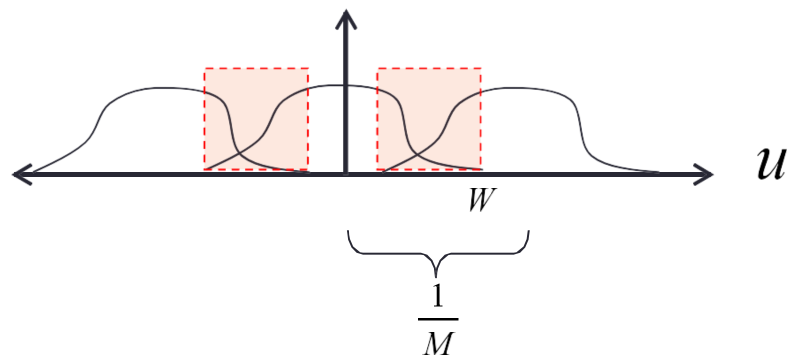


Slide credit: B. K. Gunturk

Sampling high frequency signal

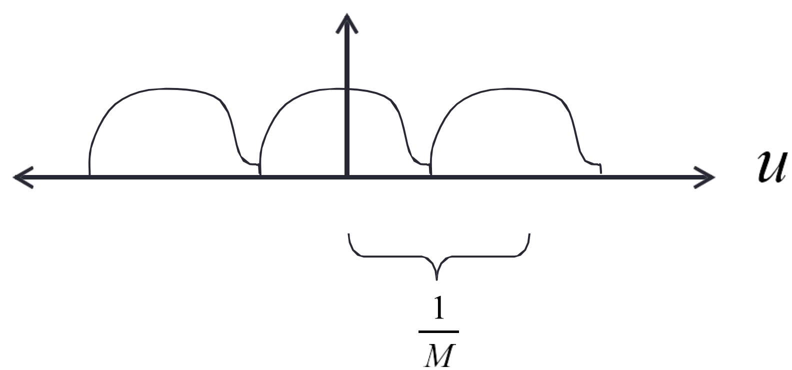
- Without anti-aliasing filter:

$$f(x)comb_M(x) \quad \Longleftrightarrow$$



- With anti-aliasing filter:

$$\left[f(x)*h(x)\right]comb_M(x) \quad \Longleftrightarrow$$



Slide credit: B. K. Gunturk

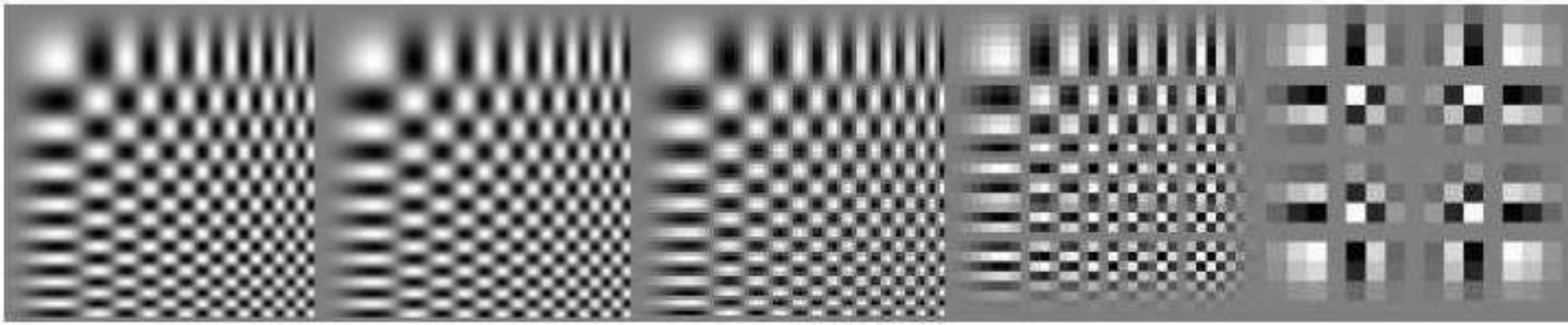
Algorithm for downsampling by factor of 2

1. Start with image
2. Apply low-pass filter
3. Sample every other pixel

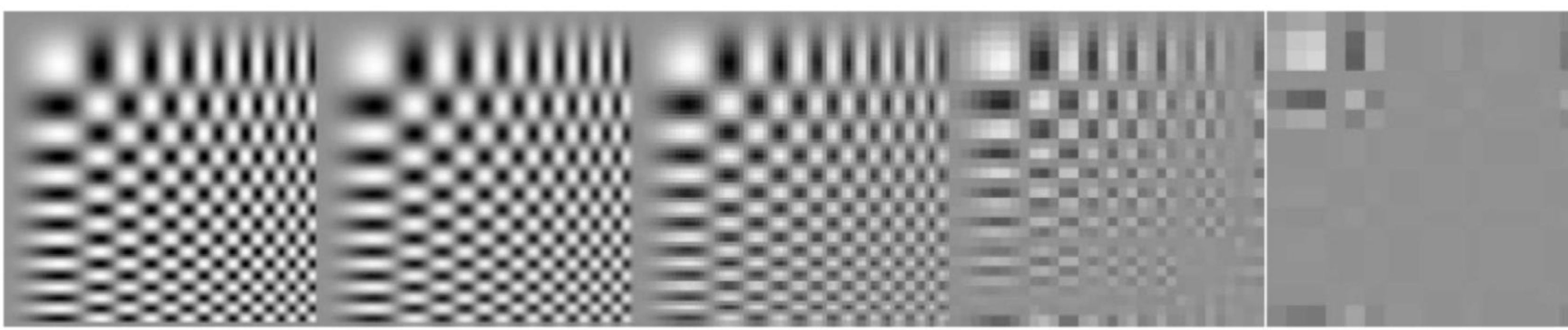
Slide credit: D. Hoiem

Anti-aliasing

256x256 128x128 64x64 32x32 16x16

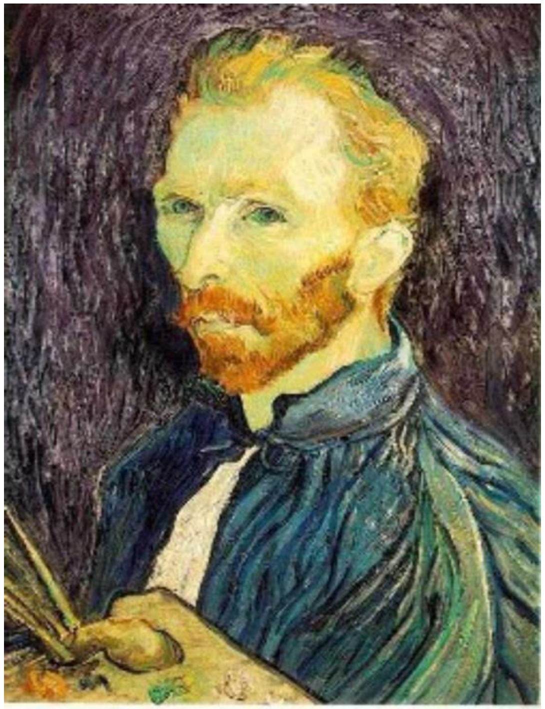


256x256 128x128 64x64 32x32 16x16



Slide credit: Forsyth and Ponce

Subsampling without pre-filtering



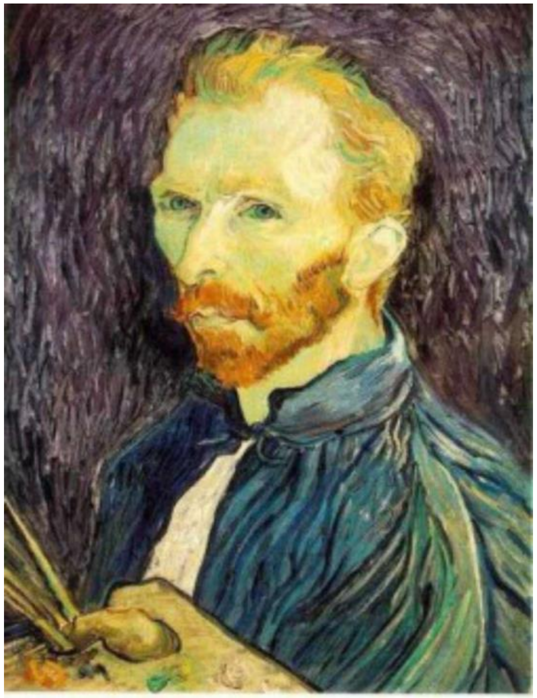
1/2

1/4 (2x zoom)

1/8 (4x zoom)

Slide credit: S. Seitz

Subsampling with Gaussian pre-filtering



Gaussian 1/2

G 1/4

G 1/8

Slide credit: S. Seitz

Week 08 – Hands on activity

Prepare and submit a Jupyter Notebook file containing the code and the results for the following Task

Task

Read an image of your choice and convert it into gray-scale image and display it.

Compute Fast Fourier Transform of the gray scale image and visualize its magnitude by using log transform (logarithm of 1 + its absolute value).

Shift the FFT of the image to the center and visualize its magnitude by using log transform (logarithm of 1 + its absolute value).

→ At this point, you should see the reason for shifting FFT to the center before visualization (it becomes easier to see the low frequency details).

Inverse the shifting using ifftshift.

Convert back the image from the Fourier domain to the spatial domain using Inverse Fast Fourier Transform.

Check and see that the image remains the same (after conversion from the spatial domain to the frequency domain and back from the frequency domain to the spatial domain).

→ Actually, the maximum of the absolute differences should be around 10^-13.