CS148: Introduction to Computer Graphics and Imaging

Basic Signal Processing: Sampling, Aliasing, Antialiasing

No Jaggies

Key Concepts

Frequency space

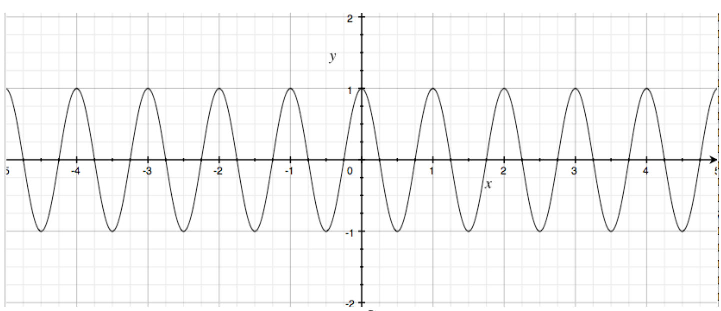
Filters and convolution

Sampling and the Nyquist frequency

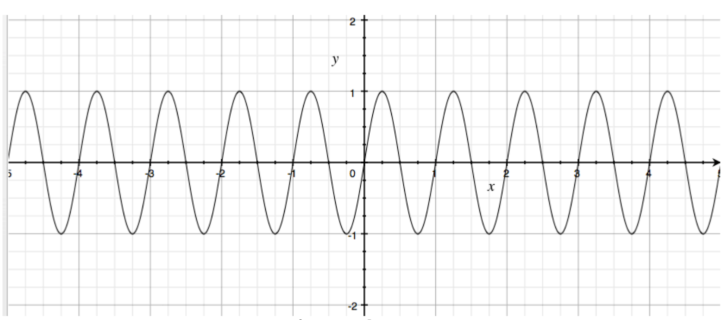
Aliasing and Antialiasing

Frequency Space

Sines and Cosines



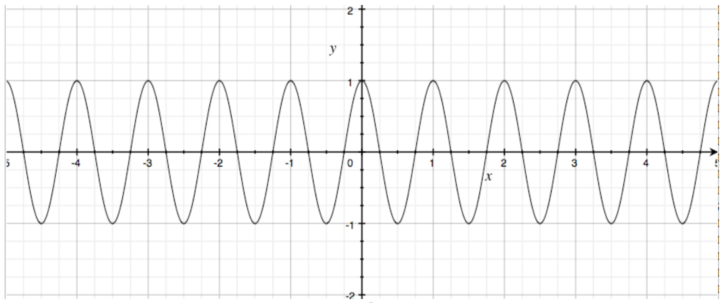
$$\cos 2\pi x$$



$$\sin 2\pi x$$

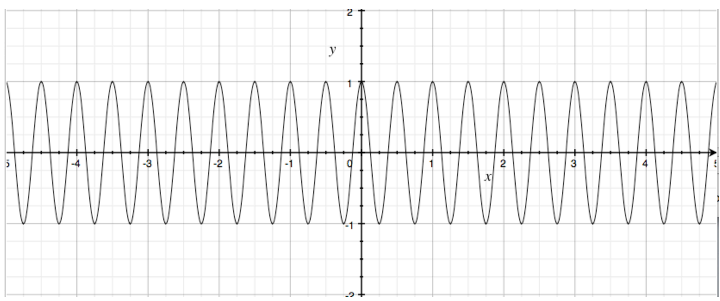
Frequencies $\cos 2\pi f x$

$$f = \frac{1}{T}$$



$$f = 1$$

$$\cos 2\pi x$$



$$f = 2$$

$$\cos 4\pi x$$

Recall Complex Exponentials

Euler's Formula

$$e^{jx} = \cos x + j \sin x$$

Odd (-x)

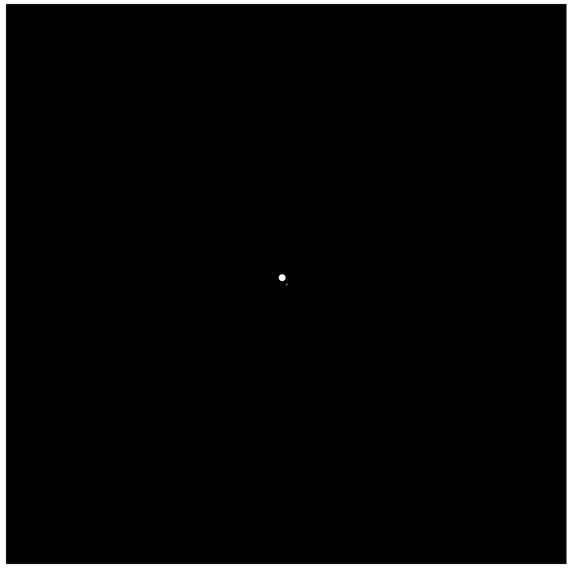
$$e^{-jx} = \cos -x + j \sin -x = \cos x - j \sin x$$

Therefore

$$\cos x = \frac{e^{jx} + e^{-jx}}{2} \quad \sin x = \frac{e^{jx} - e^{-jx}}{2j}$$

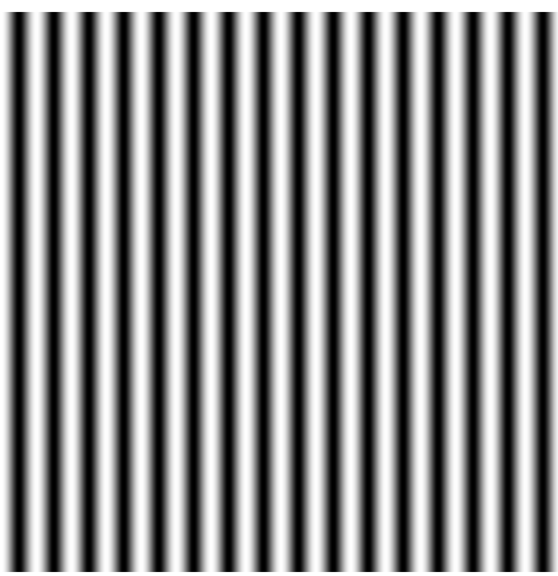
Hence, use complex exponentials for sines/cosines

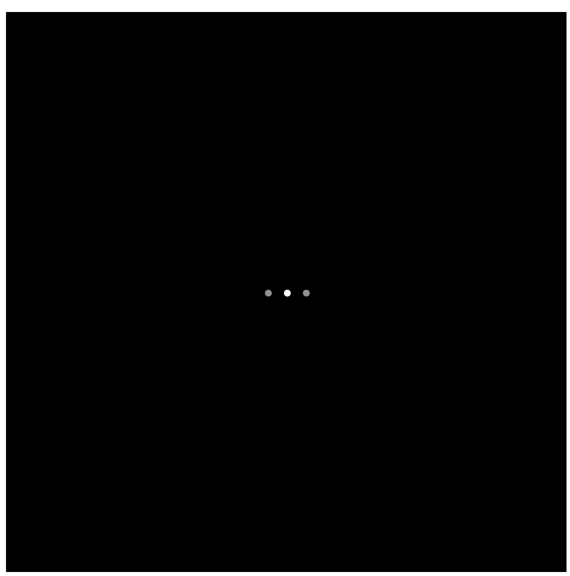
Constant



Spatial Domain **Frequency Domain**

$\sin(2\pi/32)x$

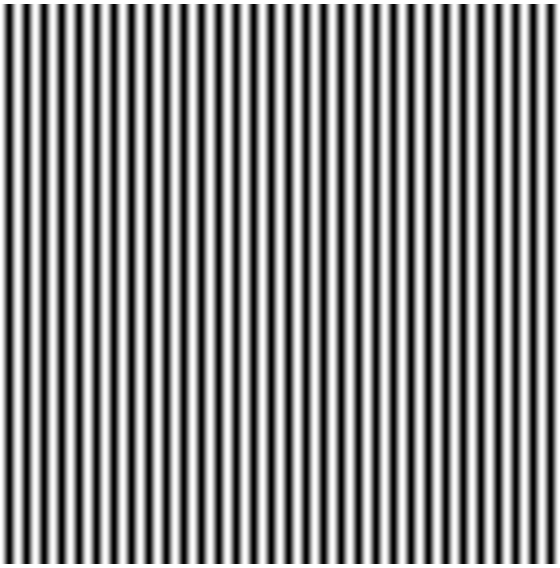


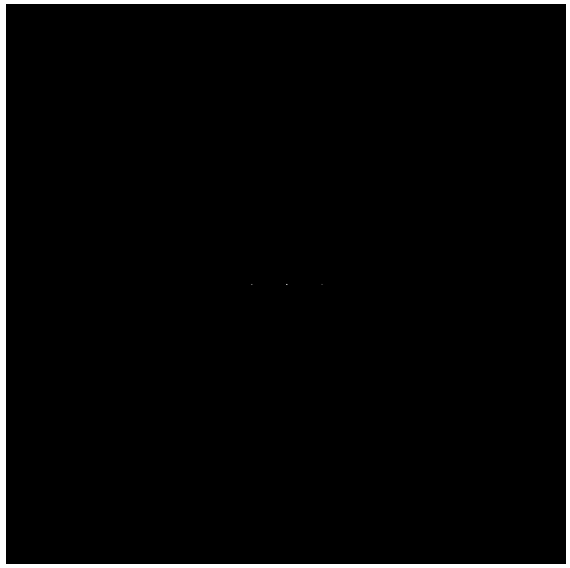


Spatial Domain **Frequency Domain**

Frequency = 1/32; 32 pixels per cycle

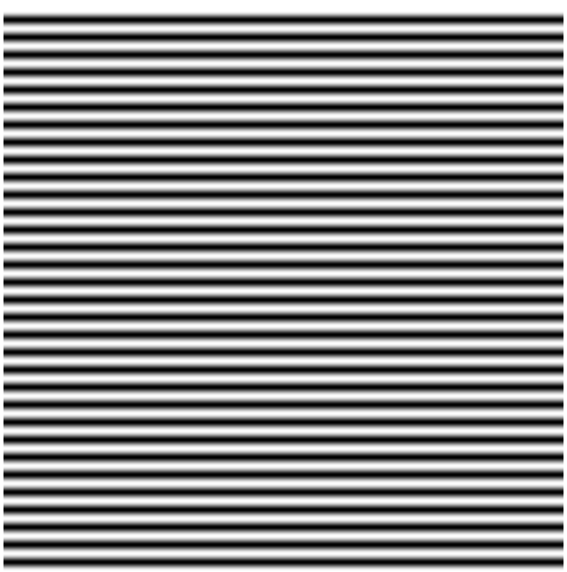
$\sin(2\pi/16)x$

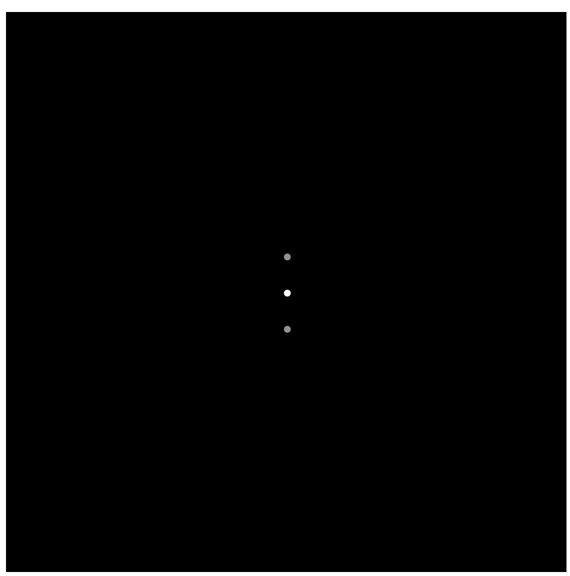




Spatial Domain **Frequency Domain**

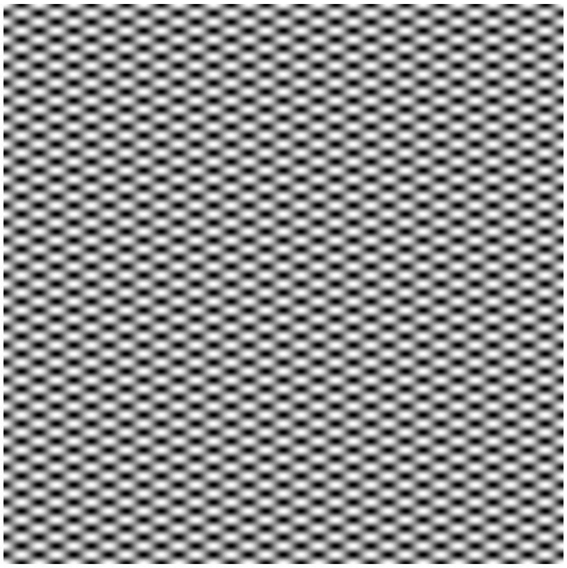
$\sin(2\pi/16)y$

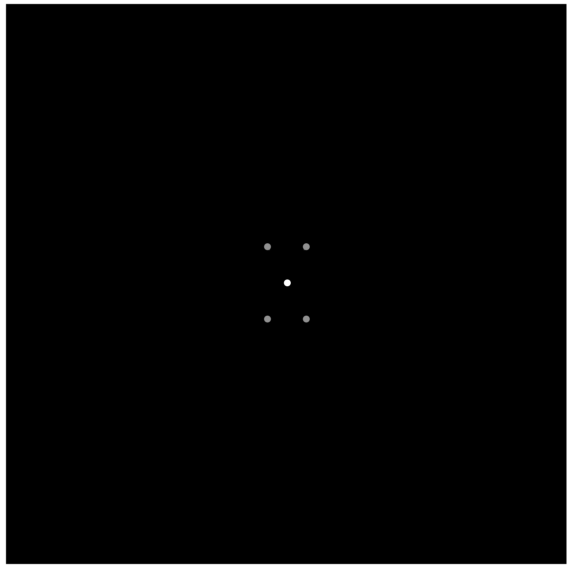




Spatial Domain **Frequency Domain**

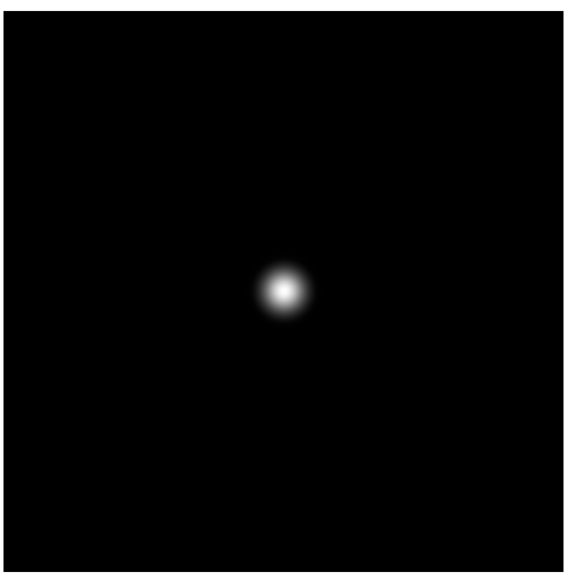
$\sin(2\pi/32)x \times \sin(2\pi/16)y$

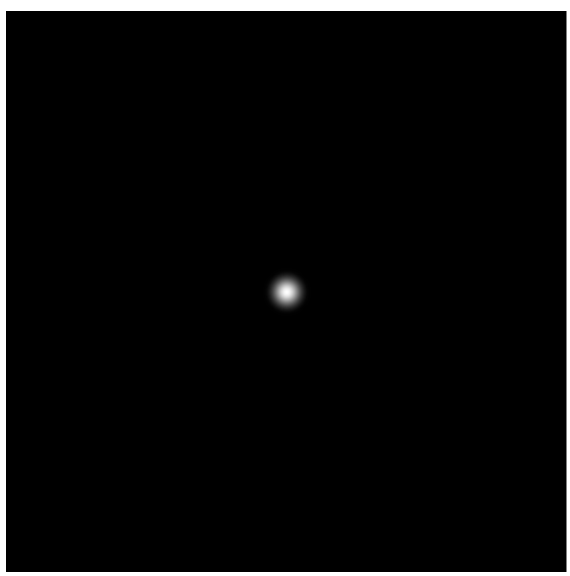




Spatial Domain **Frequency Domain**

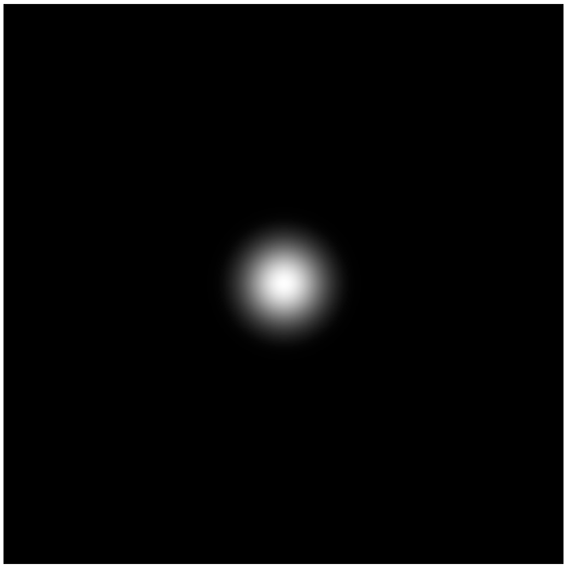
$e^{-r^2/16^2}$





Spatial Domain **Frequency Domain**

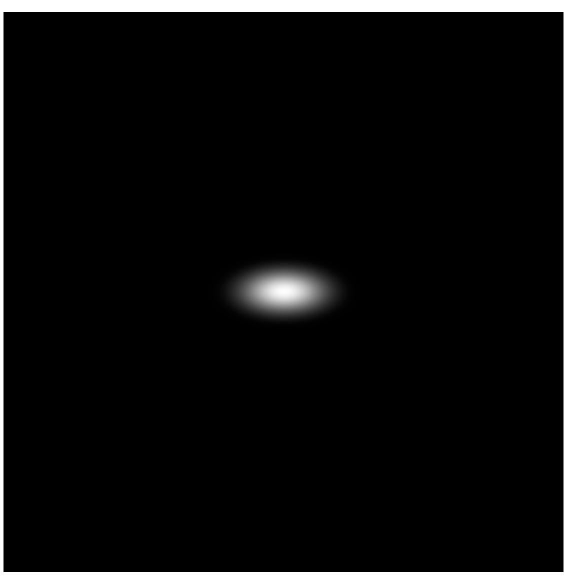
$e^{-r^2/32^2}$



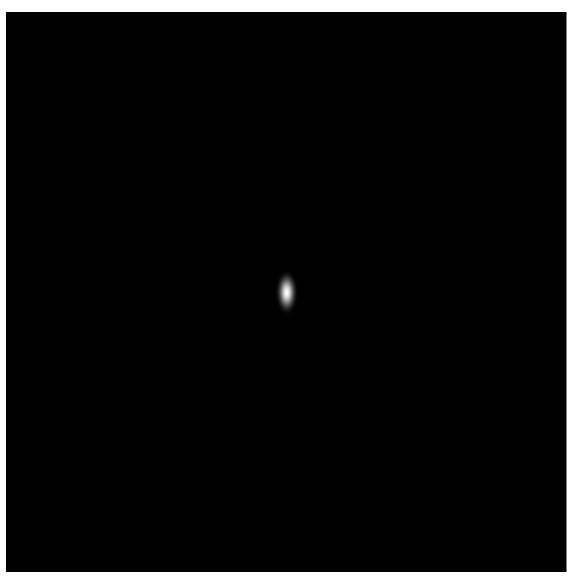
Spatial Domain

Frequency Domain

$e^{-x^2/32^2} \times e^{-y^2/16^2}$

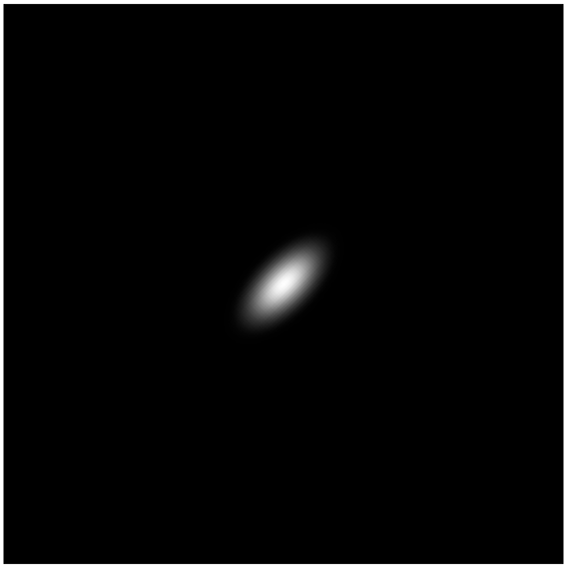


Spatial Domain



Frequency Domain

Rotate 45 $e^{-x^2/32^2} \times e^{-y^2/16^2}$

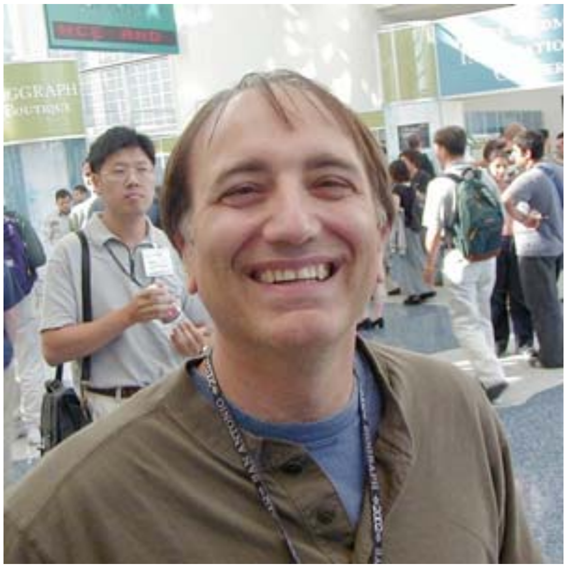


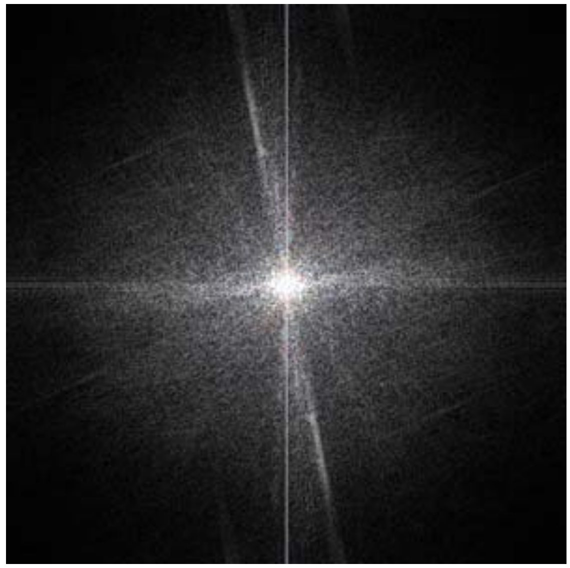
Spatial Domain

Frequency Domain

Filtering

My Humble Frequencies

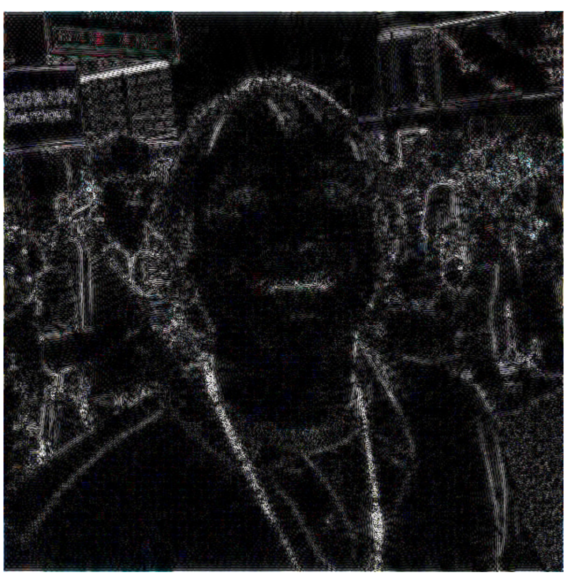


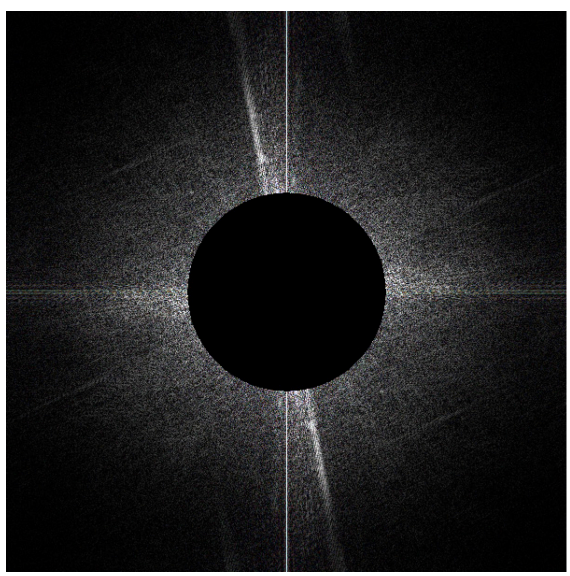


Spatial Domain

Frequency Domain

Remove Low Frequencies (Edges)

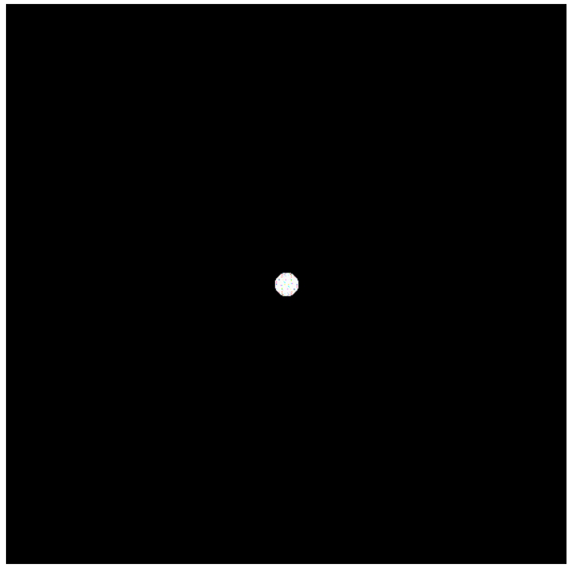




Spatial Domain

Frequency Domain

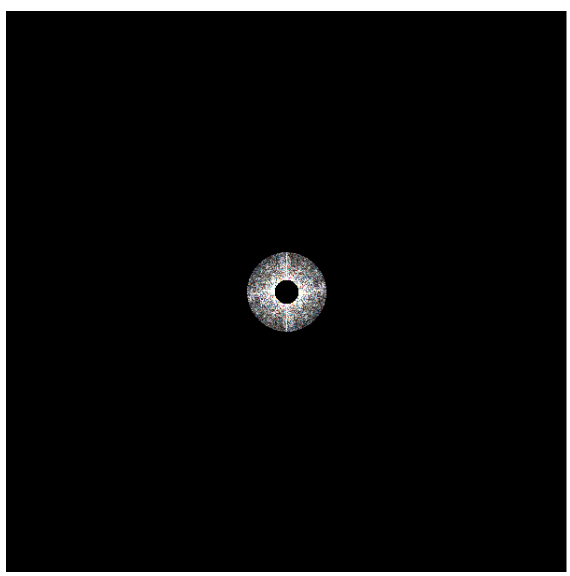
Remove High Frequencies (Blur)



Spatial Domain

Frequency Domain

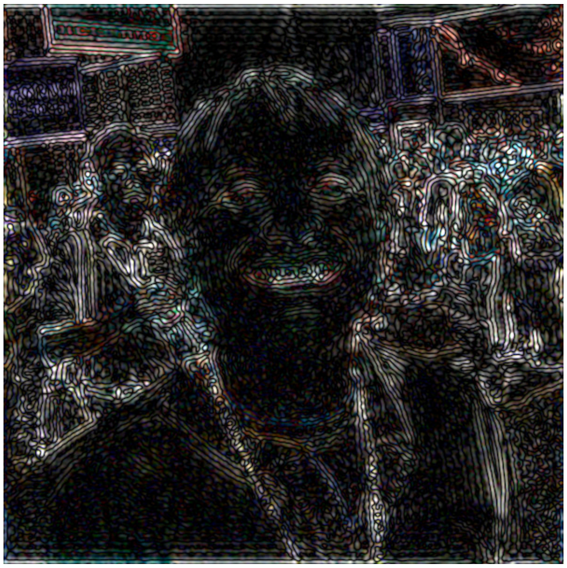
Remove Low and High Frequencies

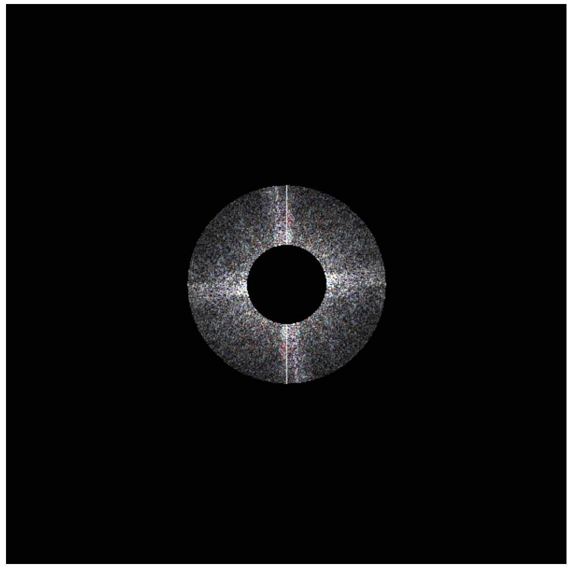


Spatial Domain

Frequency Domain

Remove Low and High Frequencies





Spatial Domain **Frequency Domain**

Filters = Convolution

Convolution

1	3	0	4	2	1
1	2				

Convolution

1	3	0	4	2	1
1	2				

1 * 1 + 3 * 2 = 7

7					

Convolution

1	3	0	4	2	1
	1	2			

3 * 1 + 0 * 2 = 3

7	3				

Convolution

1	3	0	4	2	1
		1	2		

0 * 1 + 4 * 2 = 8

7	3	8			

Convolution Theorem

A filter can be implemented in the spatial domain using convolution

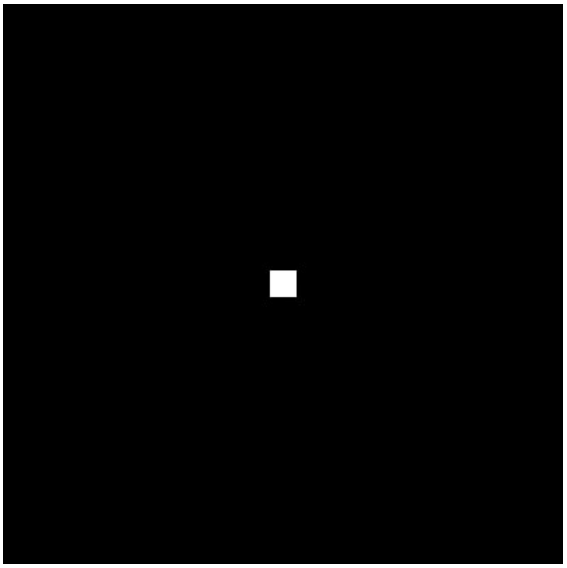
A filter can also be implemented in the frequency domain

- **Convert image to frequency domain**
- **Convert filter to frequency domain**
- **Multiply filter times image in frequency domain**
- **Convert result to the spatial domain**

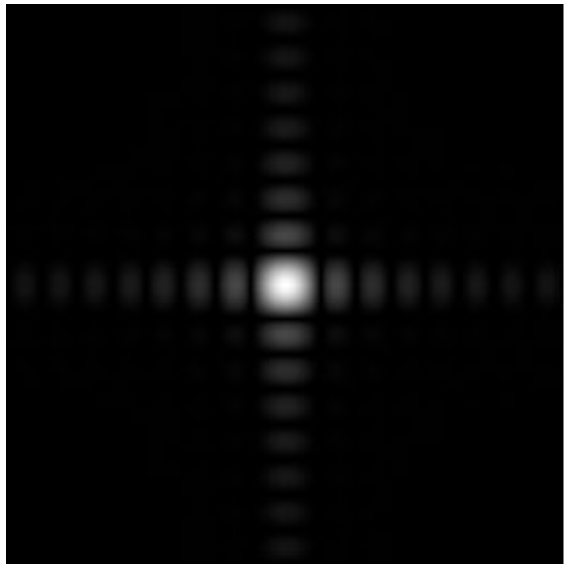
Box Filter

1	1
1	1

Box Filter = Low-Pass Filter

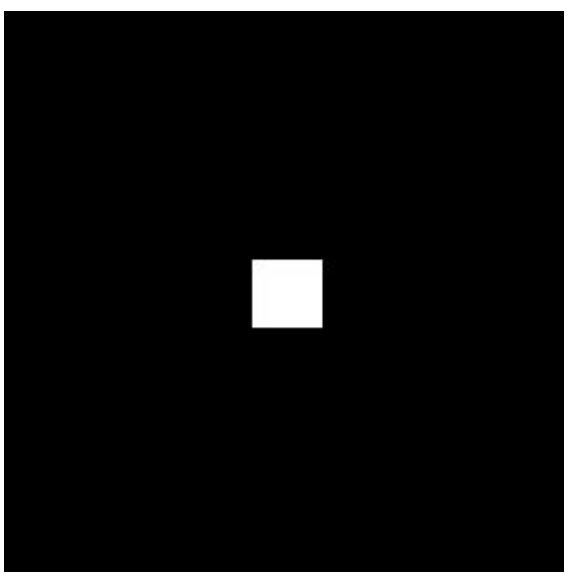


Spatial Domain

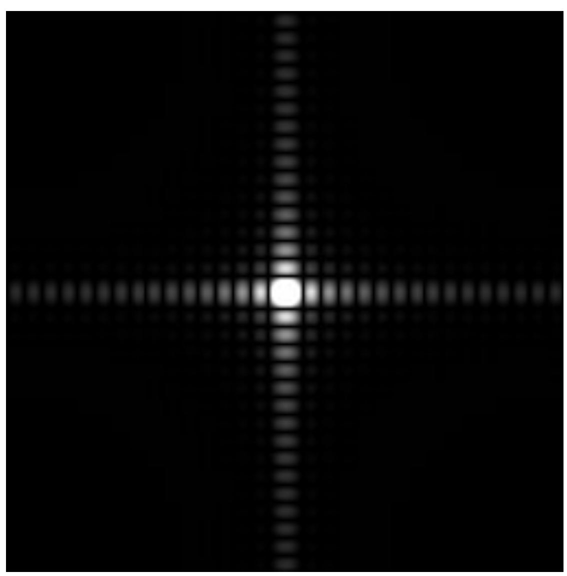


Frequency Domain

Wider Filters, Lower Frequencies



Spatial Domain



Frequency Domain

Size of Filter

As a filter is localized in space,
it spreads out in frequency

Conversely, as a filter is localized in frequency,
it spreads out in space

A box filter is very localized in space;
it has infinite extent in frequency space

Efficiency?

When would it be faster to apply the filter in the spatial domain?

When would it be faster to apply the filter in the frequency domain?

Sampling

Image Generation = Sampling

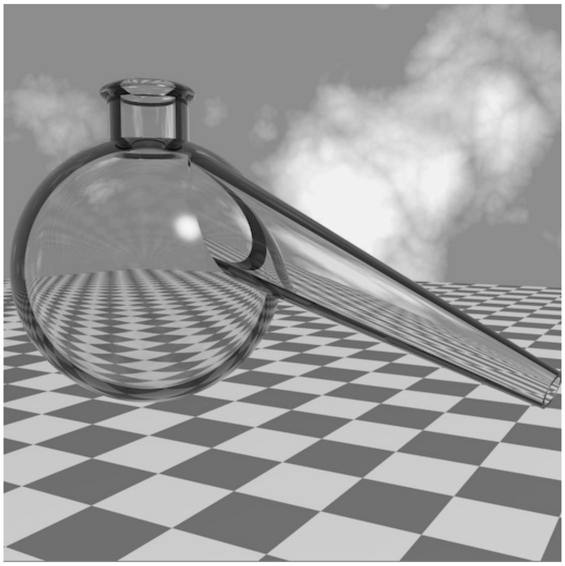
Evaluating a function at a point is sampling

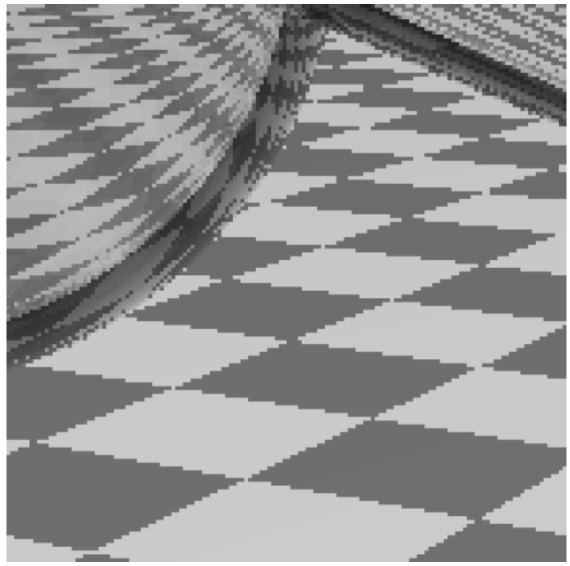
```
for( int x = 0; x < xmax; x++ )
  for( int y = 0; y < ymax; y++ )
    Image[x][y] = f(x,y);
```

Rasterization is equivalent to evaluating the function `inside(triangle,x,y)`

Sampling Causes Jaggies

Retort, by Don Mitchell





Staircase pattern or jaggies

Sampling in Computer Graphics

Artifacts due to sampling - *Aliasing*

- Jaggies – sampling in space
- Wagon wheel effect – sampling in time
- Temporal strobing – sampling in space-time
- Moire – sampling texture coordinates
- Sparkling highlights – sampling normals

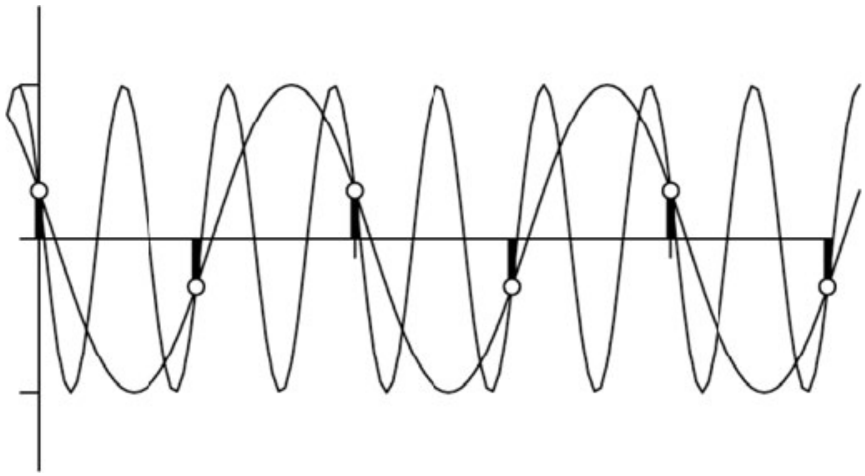
Preventing these artifacts - *Antialiasing*

Aliasing

Wagon Wheel Effect

http://www.michaelbach.de/ot/mot_wagonWheel/

"Aliases"



These two sine waves are indistinguishable
Indistinguishable frequencies are called "aliases"

Nyquist Frequency

Definition: The Nyquist frequency is ½ the sampling frequency (1/Ts)

Frequencies above the Nyquist frequency appear as aliases

No aliases appear if the function being sampled has no frequencies above the Nyquist frequency

Antialiasing

Antialiasing

Simple idea:

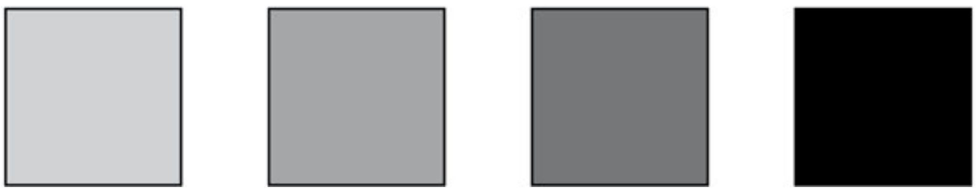
Remove frequencies above the Nyquist frequency before sampling

How? Filtering before sampling

Prefiltering by Computing Coverage

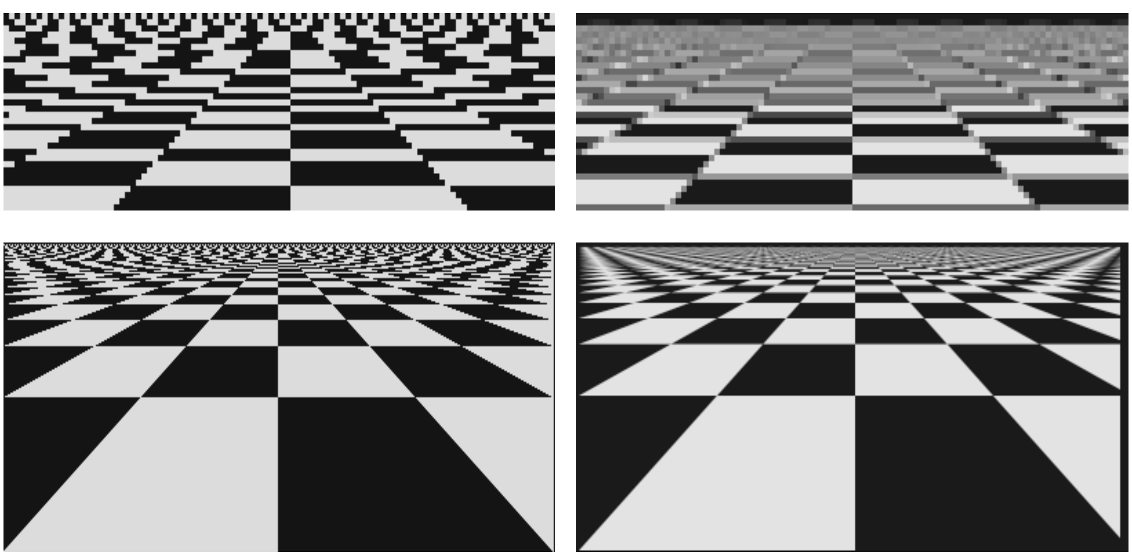
A 1 pixel box filter removes frequencies whose period is less than or equal to 1 pixel

Original



Filtered

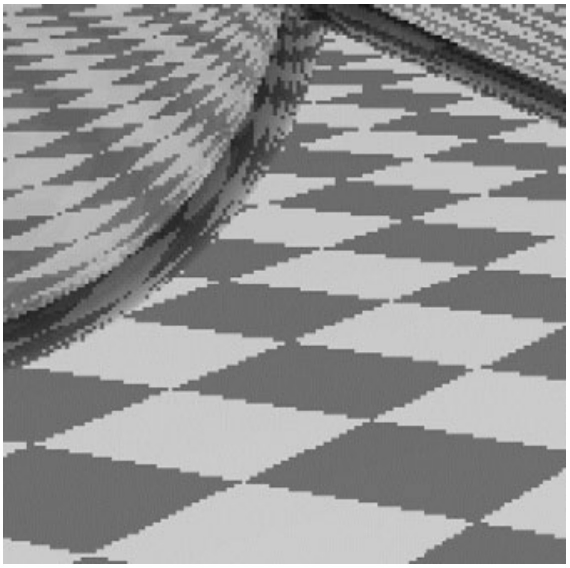
Point- vs. Area-Sampled

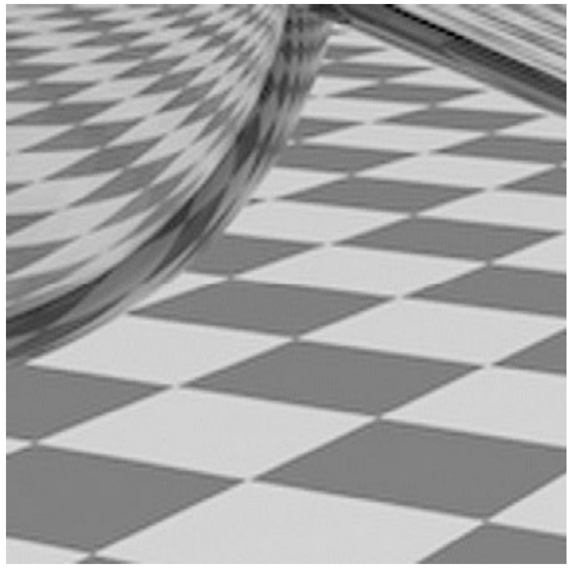


Point Area

Checkerboard sequence by Tom Duff

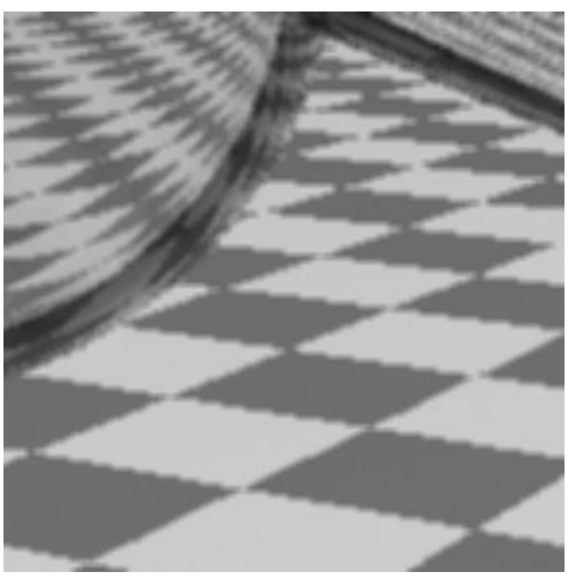
Antialiasing

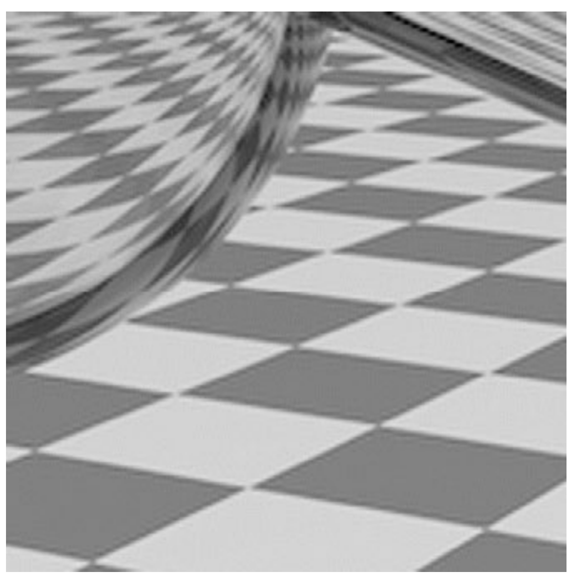




Jaggies **Prefilter**

Antialiasing vs. Blurred Aliases





Blurred Jaggies **Prefilter**

Things to Remember

Signal processing

- Frequency domain vs. spatial domain
- Filters in the frequency domain
- Filters in the spatial domain = convolution

Sampling and aliasing

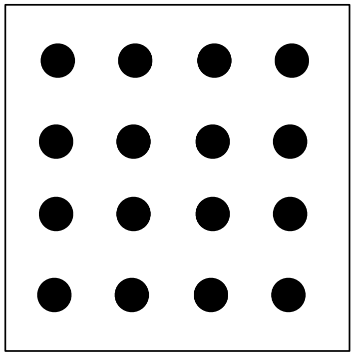
- Image generation involves sampling
  - May also sample geometry, motion, …
- Nyquist frequency is ½ the sampling rate
- Frequencies above the Nyquist frequency appear as other frequencies – aliases
- Antialiasing – Filter before sampling

Extra Slides

Supersampling

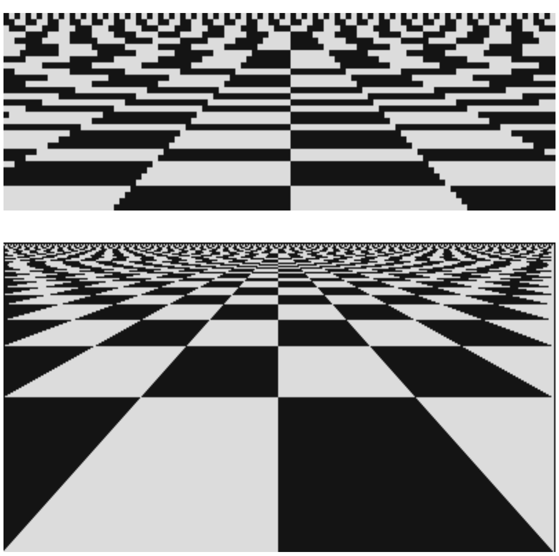
Supersampling

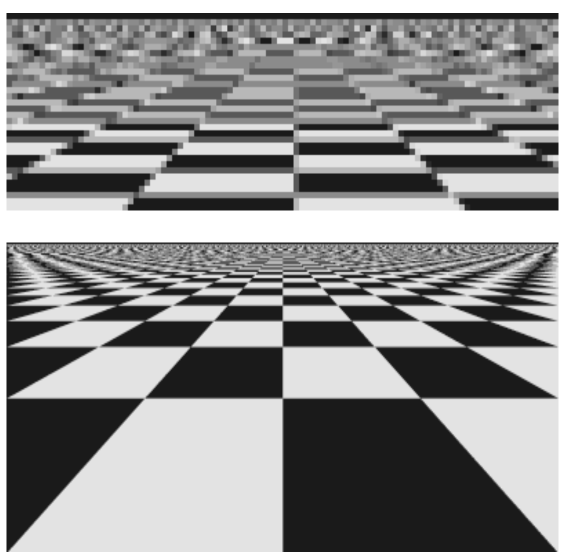
Approximate a box filter by taking more samples and averaging them together



4 x 4 supersampling

Point-sampling vs. Super-sampling

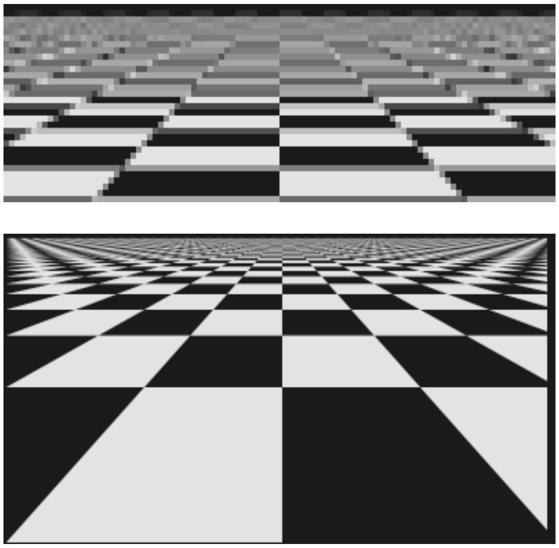




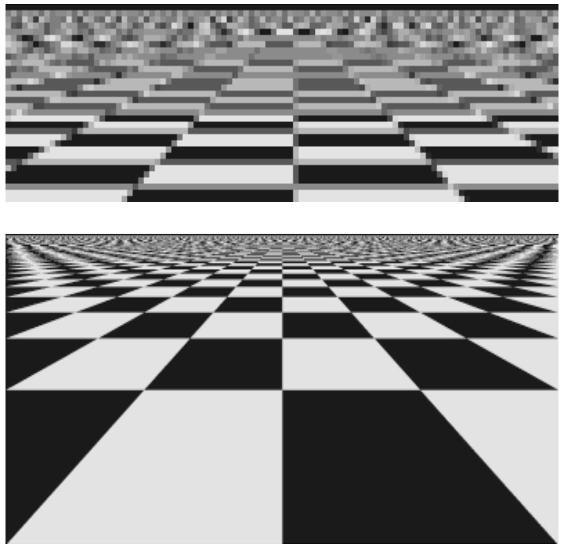
Point **4x4 Super-sampled**

Checkerboard sequence by Tom Duff

Area-Sampling vs. Super-sampling



Exact Area



4x4 Super-sampled