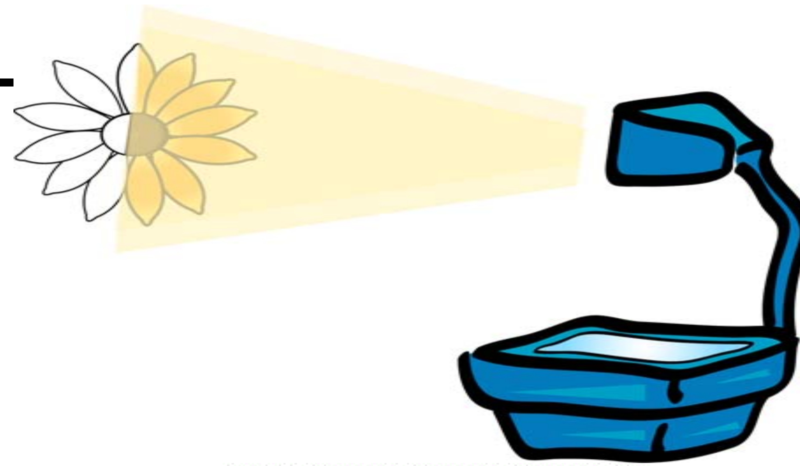
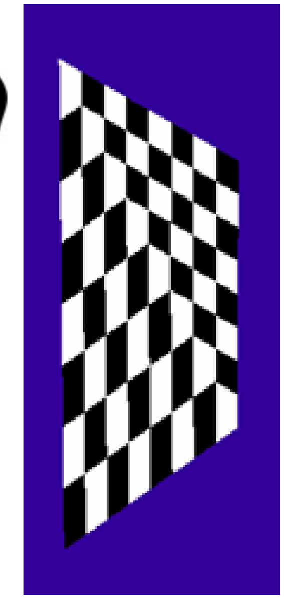
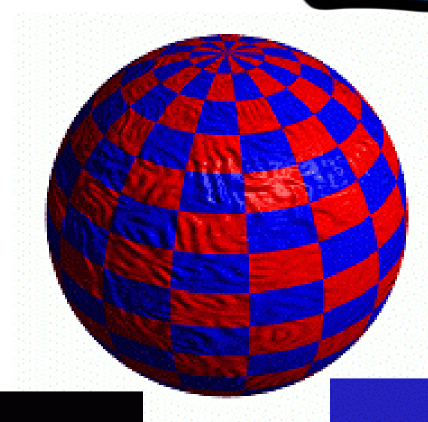
# Sampling, Aliasing, & Mipmaps

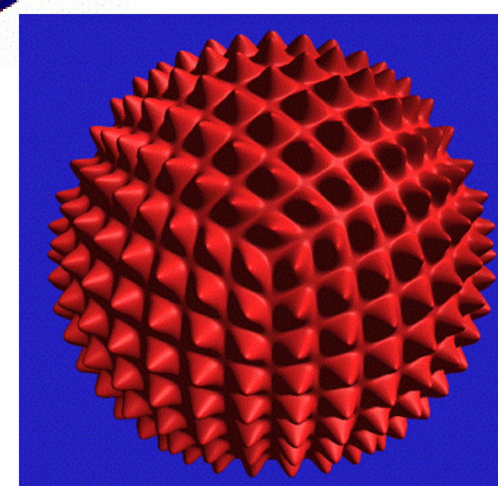
# Last Time?

- 2D Texture Mapping
- Perspective Correct Interpolation
- Common Texture Coordinate Projections
- Bump Mapping
- Displacement Mapping
- Environment Mapping









# Texture Maps for Illumination

- Also called "Light Maps"

# Today

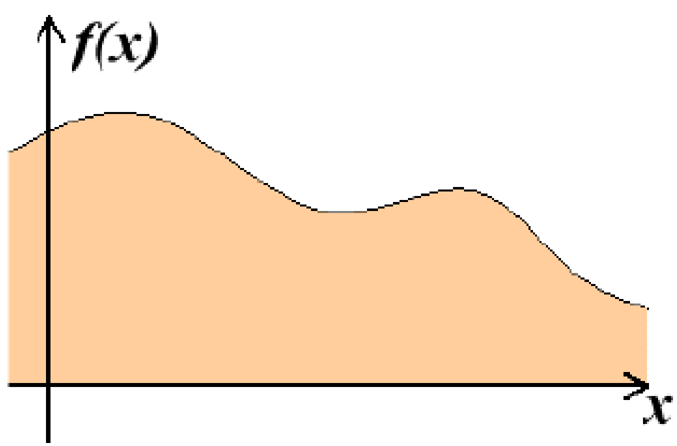
- What is a Pixel?
- Examples of Aliasing
- Signal Reconstruction
- Reconstruction Filters
- Anti-Aliasing for Texture Maps

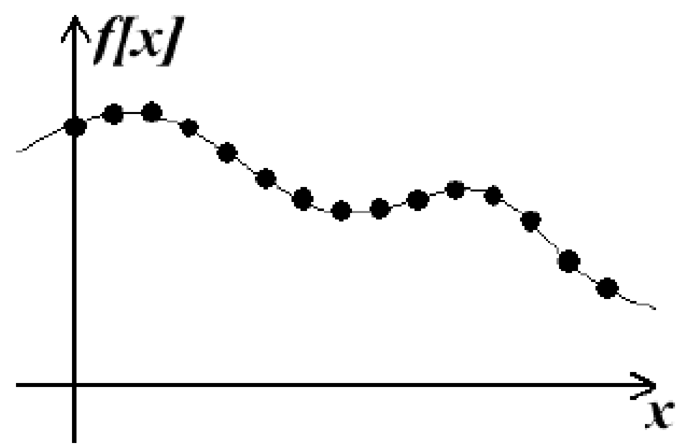
# What is a Pixel?

- A pixel is not:
  - a box
  - a disk
  - a teeny tiny little light
- A pixel is a point
  - it has no dimension
  - it occupies no area
  - it cannot be seen
  - it can have a coordinate
- A pixel is more than just a point, it is a sample!

# More on Samples

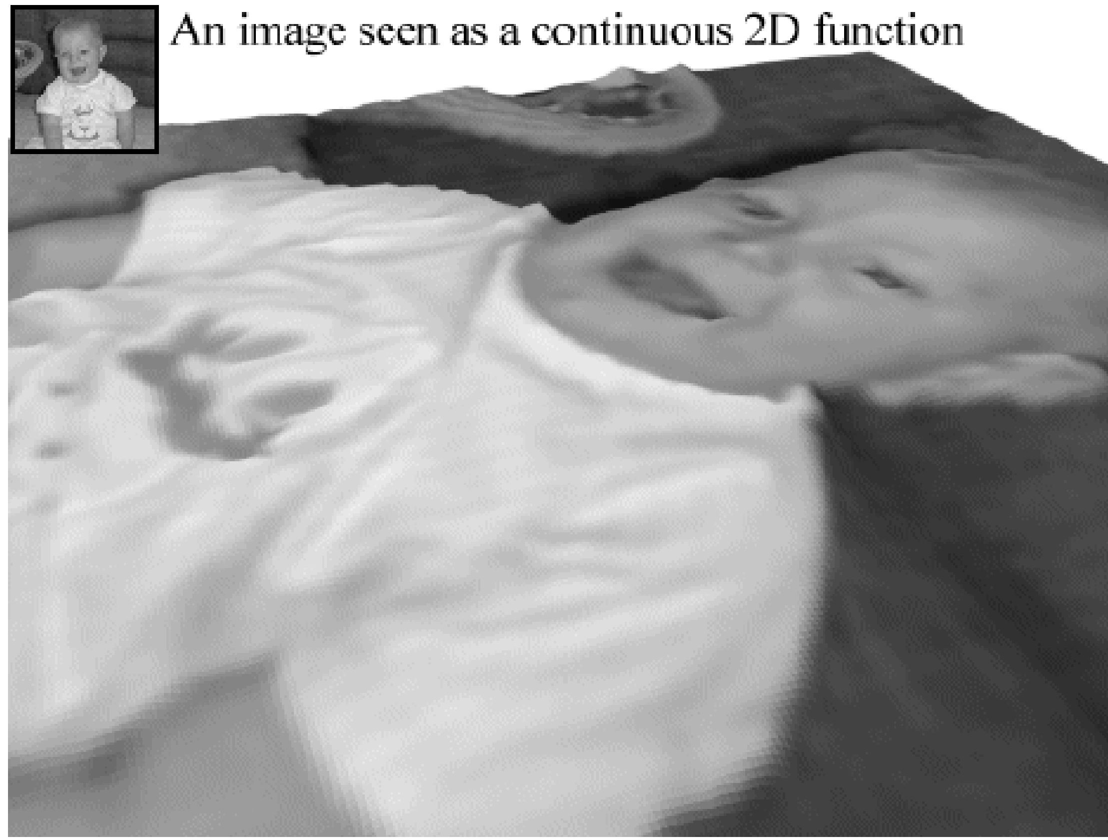
- Most things in the real world are *continuous*, yet everything in a computer is *discrete*
- The process of mapping a continuous function to a discrete one is called *sampling*
- The process of mapping a continuous variable to a discrete one is called *quantization*
- To represent or render an image using a computer, we must both sample and quantize



# An Image is a 2D Function

- An *ideal image* is a function I(x,y) of intensities.
- It can be plotted as a height field.
- In general an image cannot be represented as a continuous, analytic function.
- Instead we represent images as tabulated functions.
- How do we fill this table?



# Sampling Grid

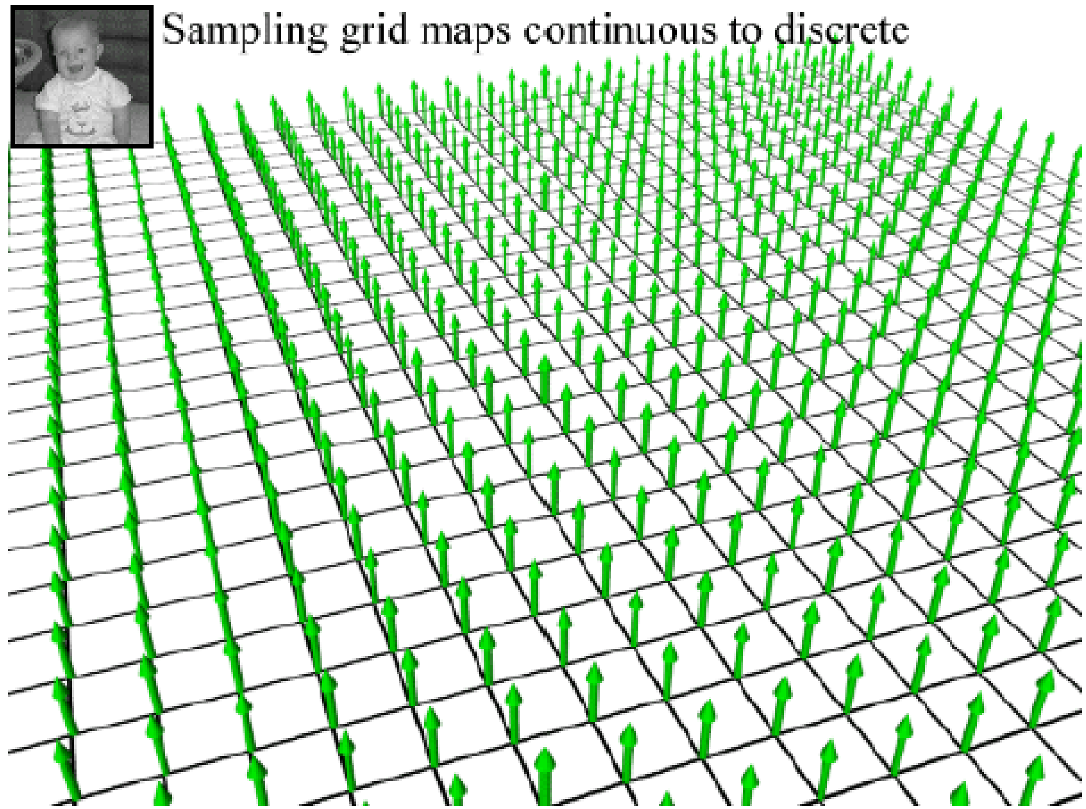
- We can generate the table values by multiplying the continuous image function by a sampling grid of Kronecker delta functions.

The definiton of the 2-D Kronecker delta is:

$$\delta(x,y) = \begin{cases} 1, & (x,y) = (0,0) \\ 0, & \text{otherwise} \end{cases}$$

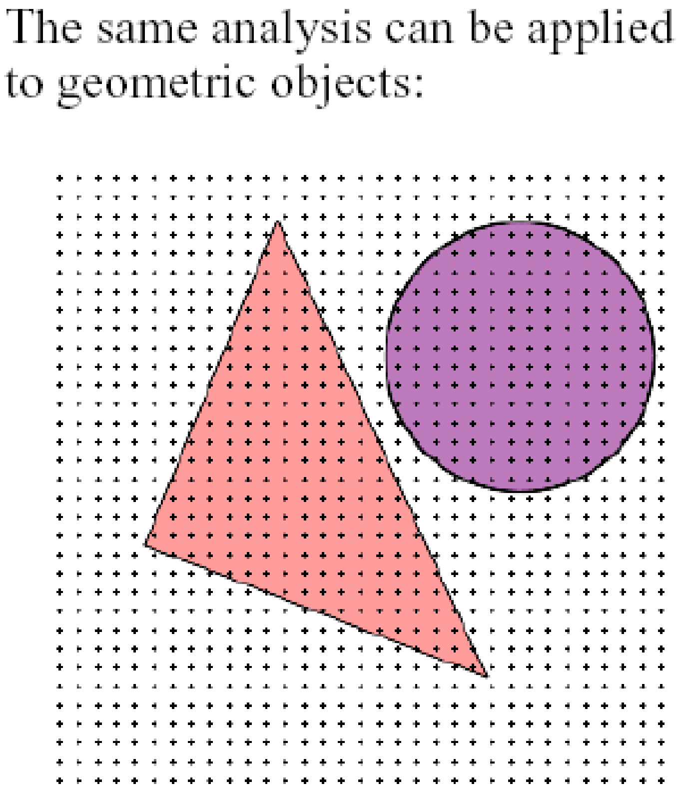
And a 2-D sampling grid:

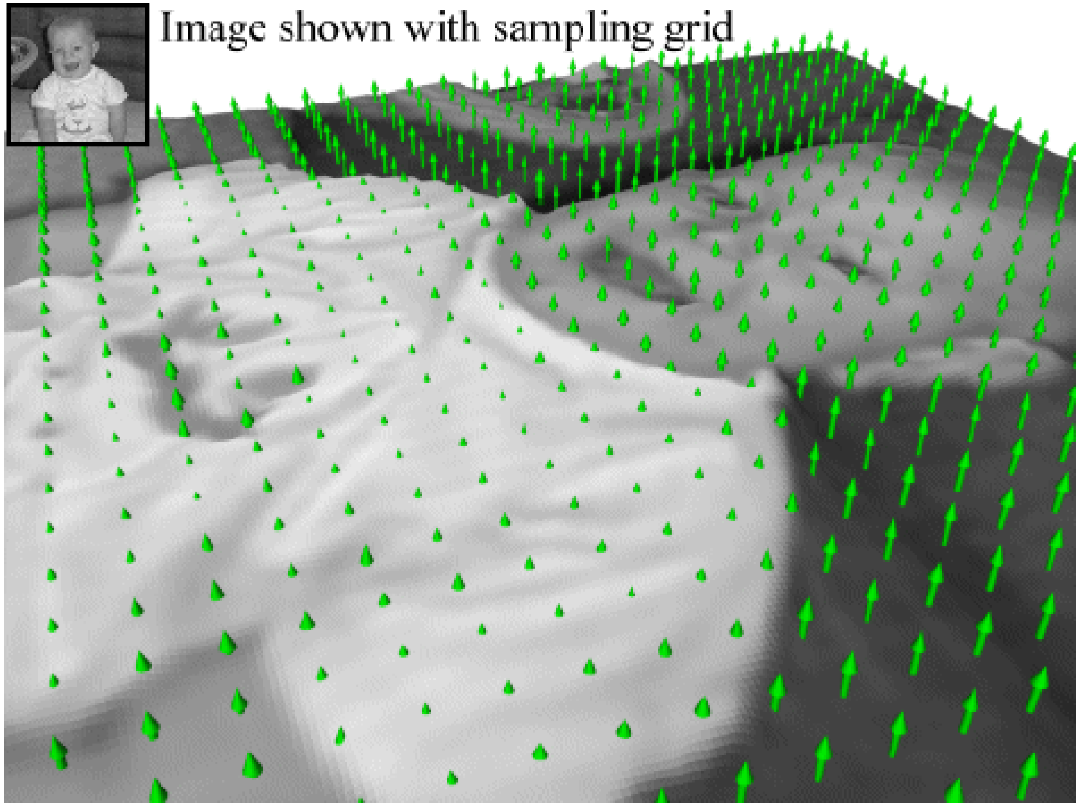
$$\sum_{j=0}^{h-1} \sum_{i=0}^{w-1} \delta(u-i, v-j)$$



# Sampling an Image

- The result is a set of point samples, or pixels.




Courtesy of Leonard McMillan, Computer Science at the University of North Carolina in Chapel Hill. Used with permission.

# Questions?

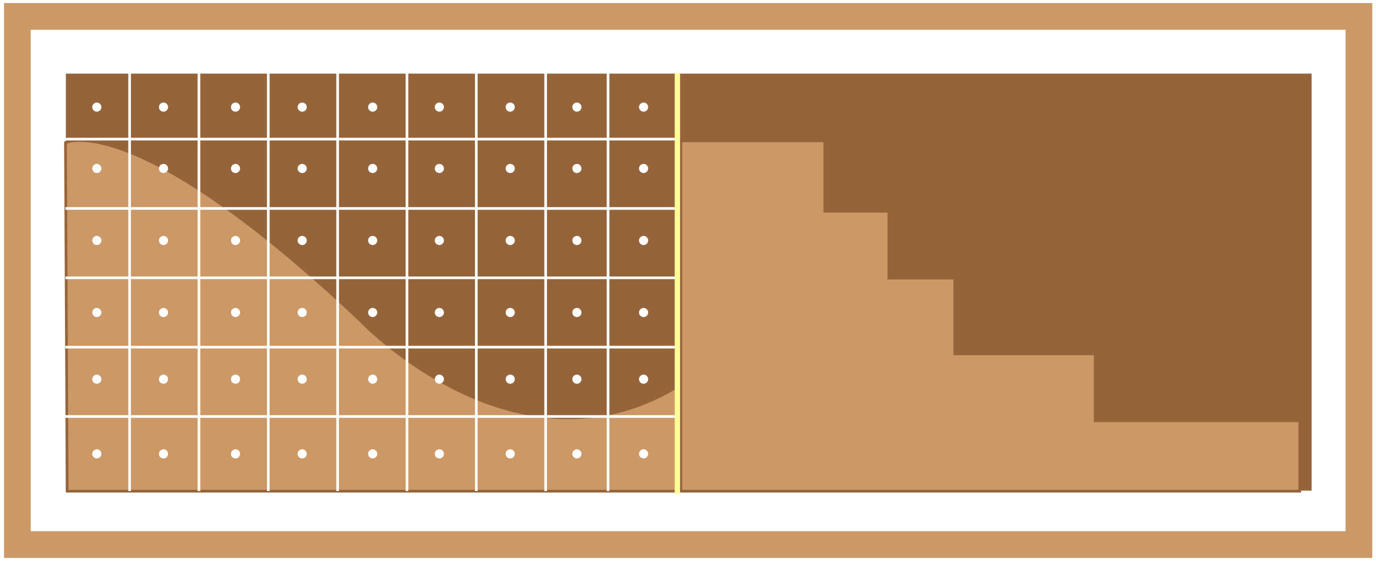
# Today

- What is a Pixel?
- Examples of Aliasing
- Signal Reconstruction
- Reconstruction Filters
- Anti-Aliasing for Texture Maps

# Examples of Aliasing

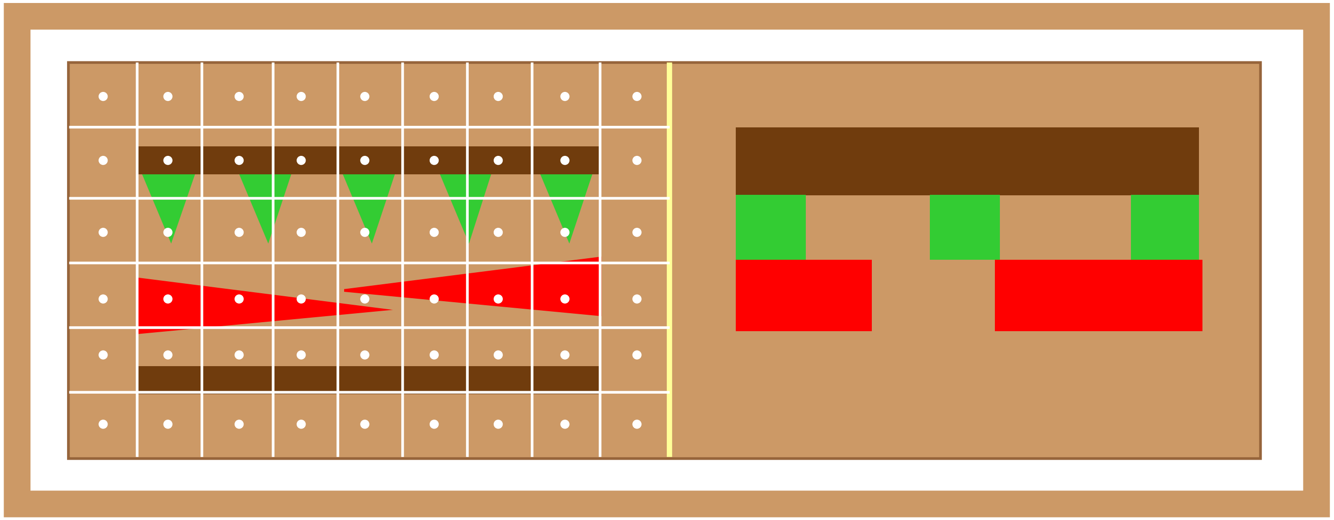
Image removed due to copyright considerations.

# Examples of Aliasing



JAGGED BOUNDARIES

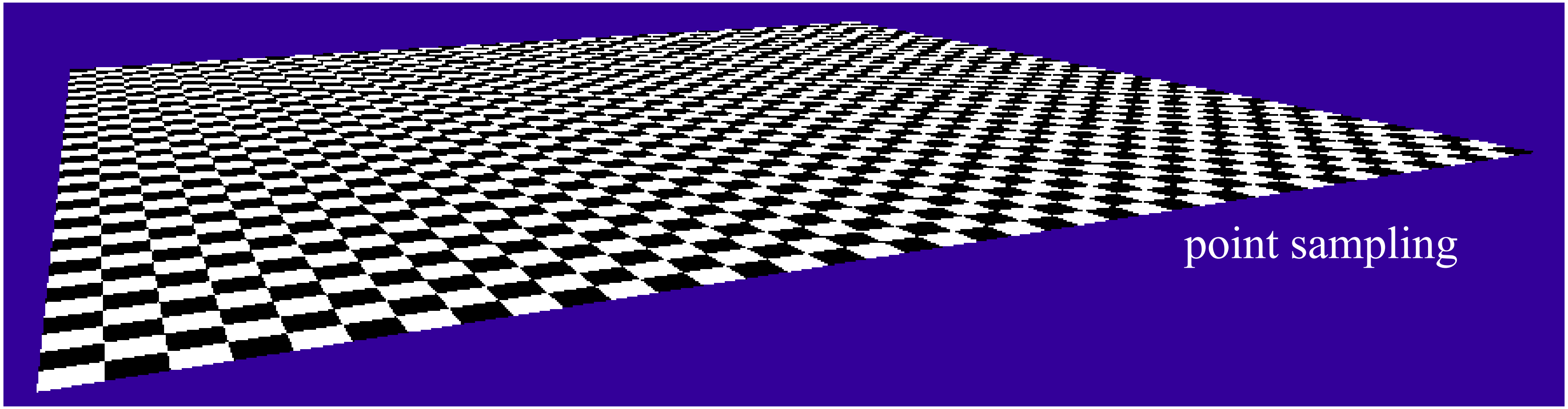
# Examples of Aliasing



IMPROPERLY RENDERED DETAIL

# Examples of Aliasing

## Texture Errors



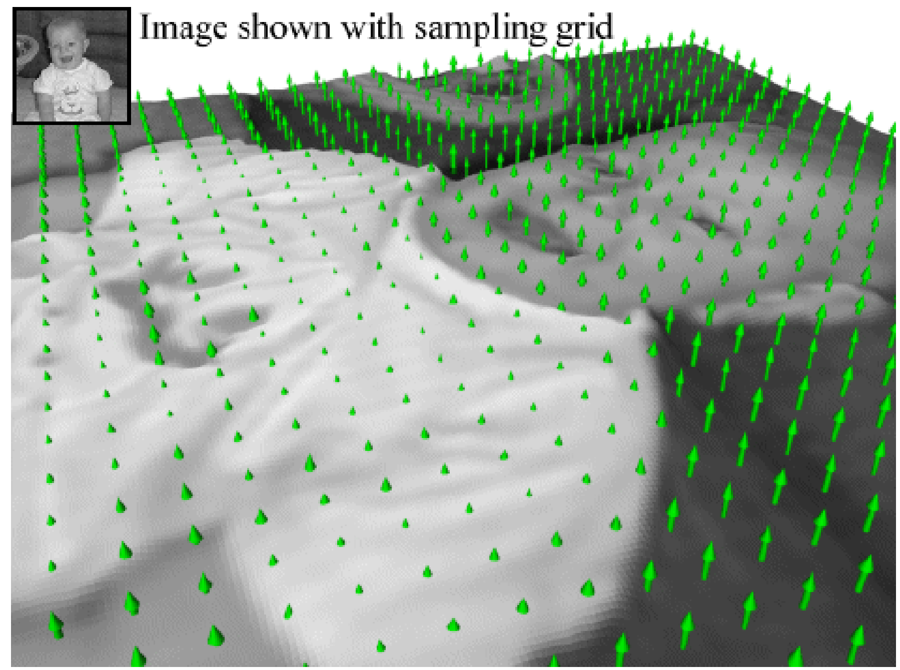
# Questions?

# Today

- What is a Pixel?
- Examples of Aliasing
- Signal Reconstruction
  - Sampling Density
  - Fourier Analysis & Convolution
- Reconstruction Filters
- Anti-Aliasing for Texture Maps

# Sampling Density

- How densely must we sample an image in order to capture its essence?
- If we under-sample the signal, we won't be able to accurately reconstruct it...



# Nyquist Limit / Shannon's Sampling Theorem

- If we insufficiently sample the signal, it may be mistaken for something simpler during reconstruction (that's aliasing!)

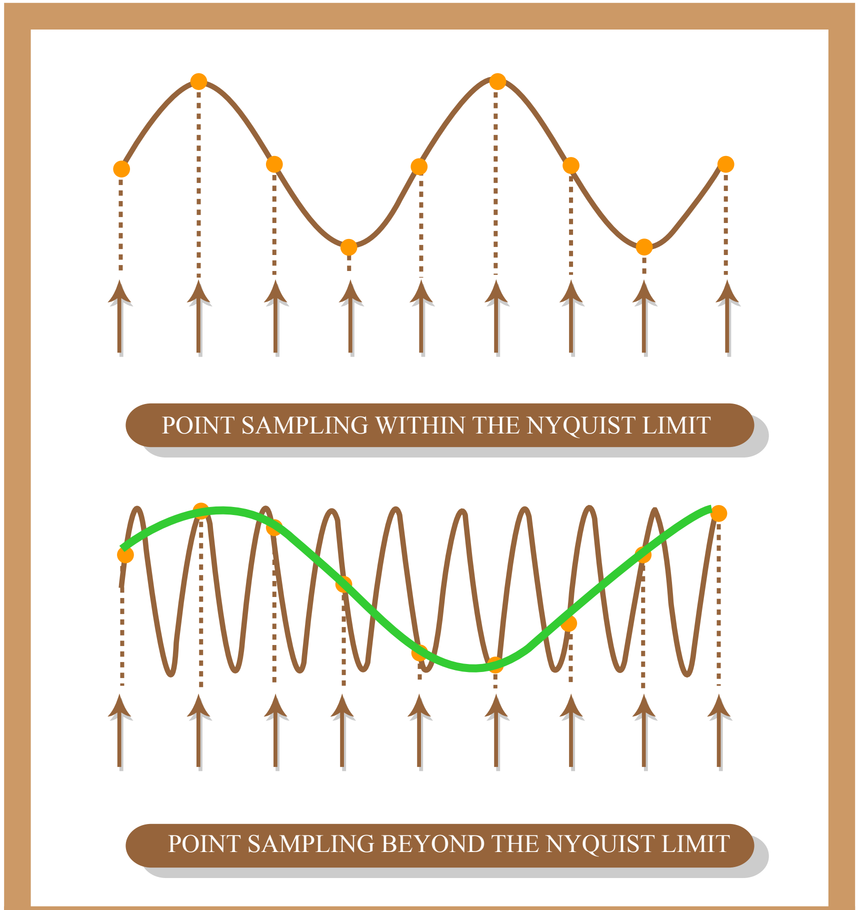
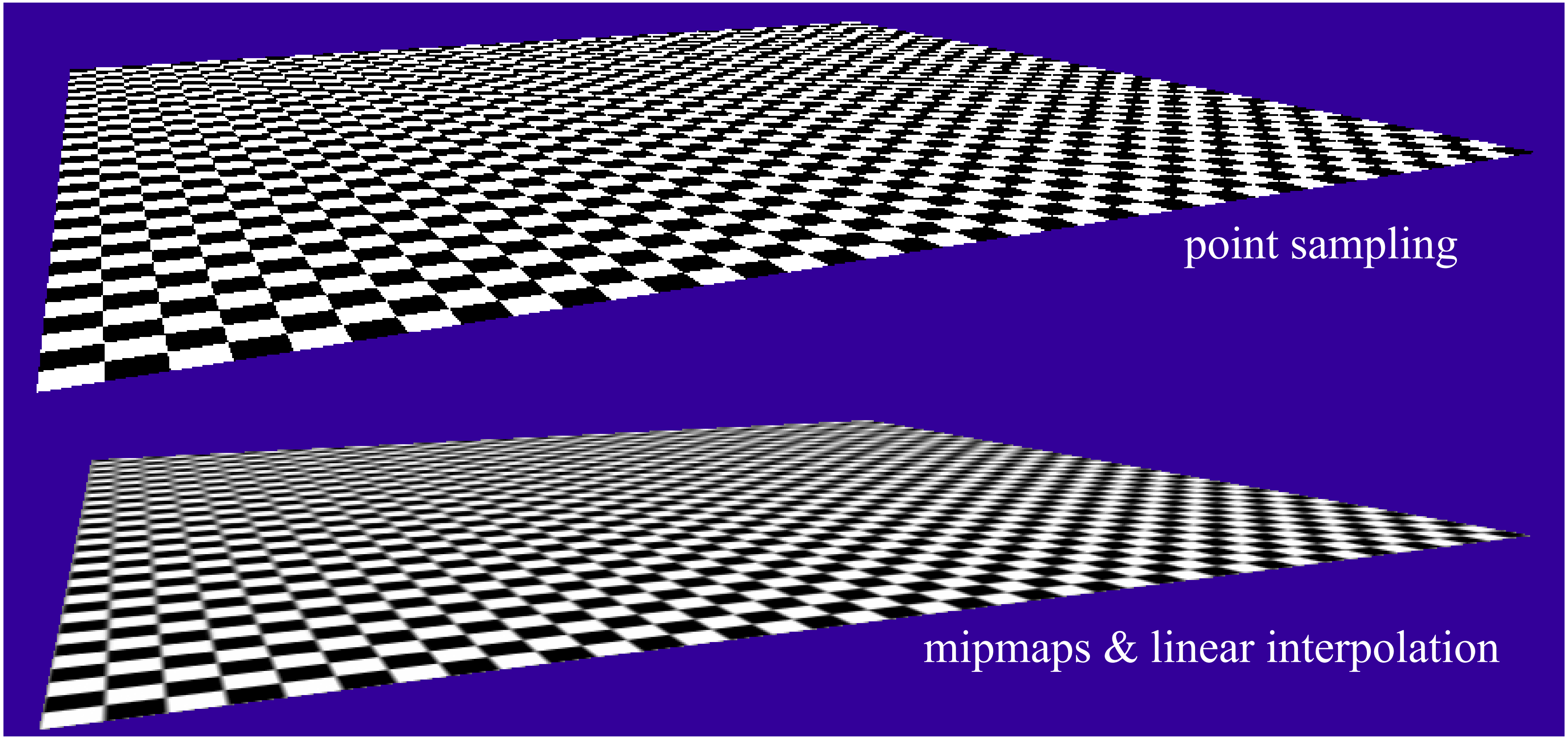



Image adapted from:
Robert L. Cook, "Stochastic Sampling and Distributed Ray Tracing." In *An Introduction to Ray Tracing.* Edited by Andrew Glassner. Academic Press Limited, 1989.

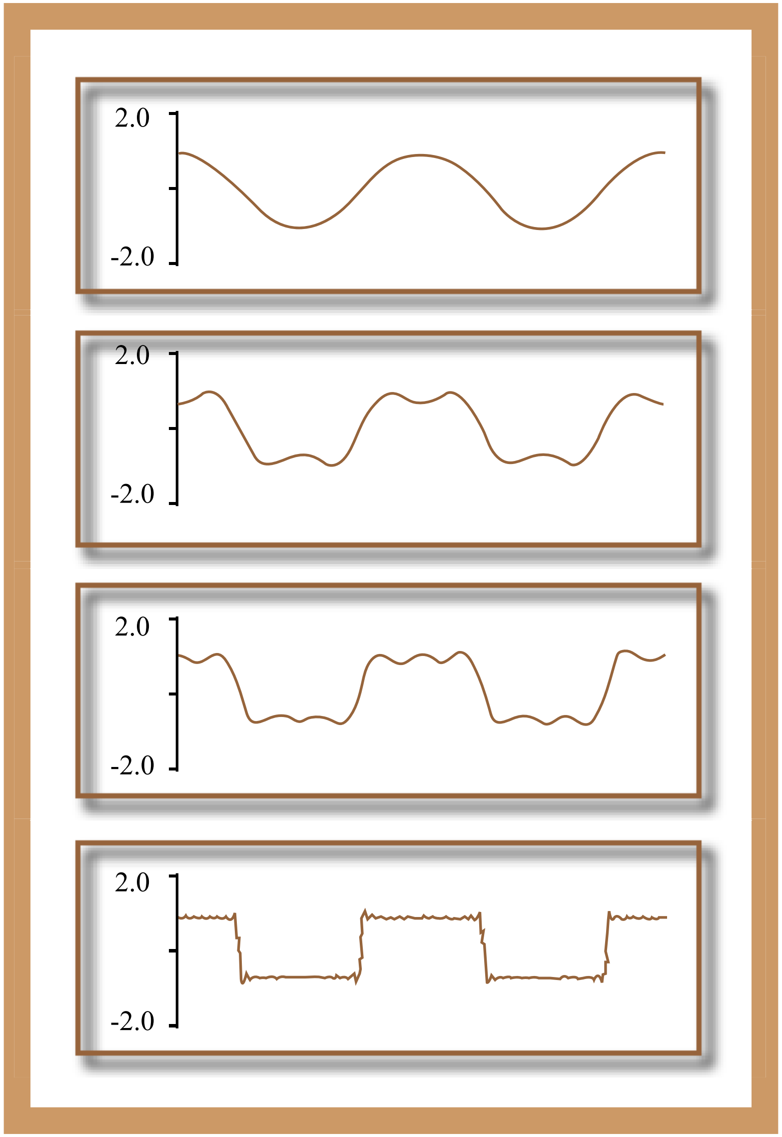
# Examples of Aliasing

## Texture Errors



# Remember Fourier Analysis?

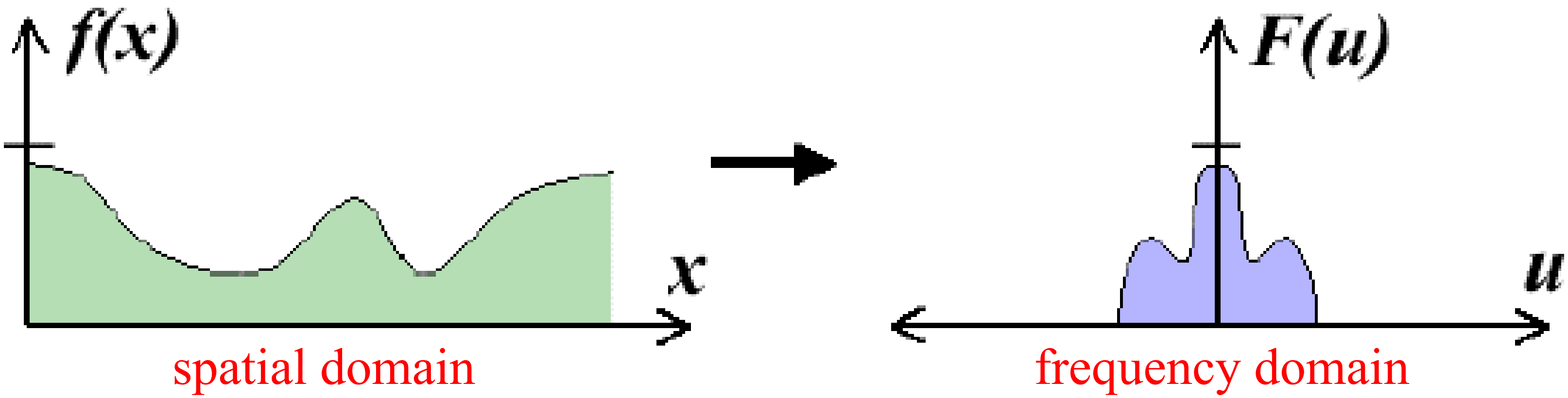
- All periodic signals can be represented as a summation of sinusoidal waves.




Images adapted from
http://axion.physics.ubc.ca/341-02/fourier/fourier.html

# Remember Fourier Analysis?

- Every periodic signal in the *spatial domain* has a dual in the *frequency domain*.




- This particular signal is *band-limited*, meaning it has no frequencies above some threshold

# Remember Fourier Analysis?

- We can transform from one domain to the other using the Fourier Transform.

frequency domain → $F(u,v)$; spatial domain → $f(x,y)$

Fourier Transform

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-i2\pi(ux+vy)}\, dx\, dy$$

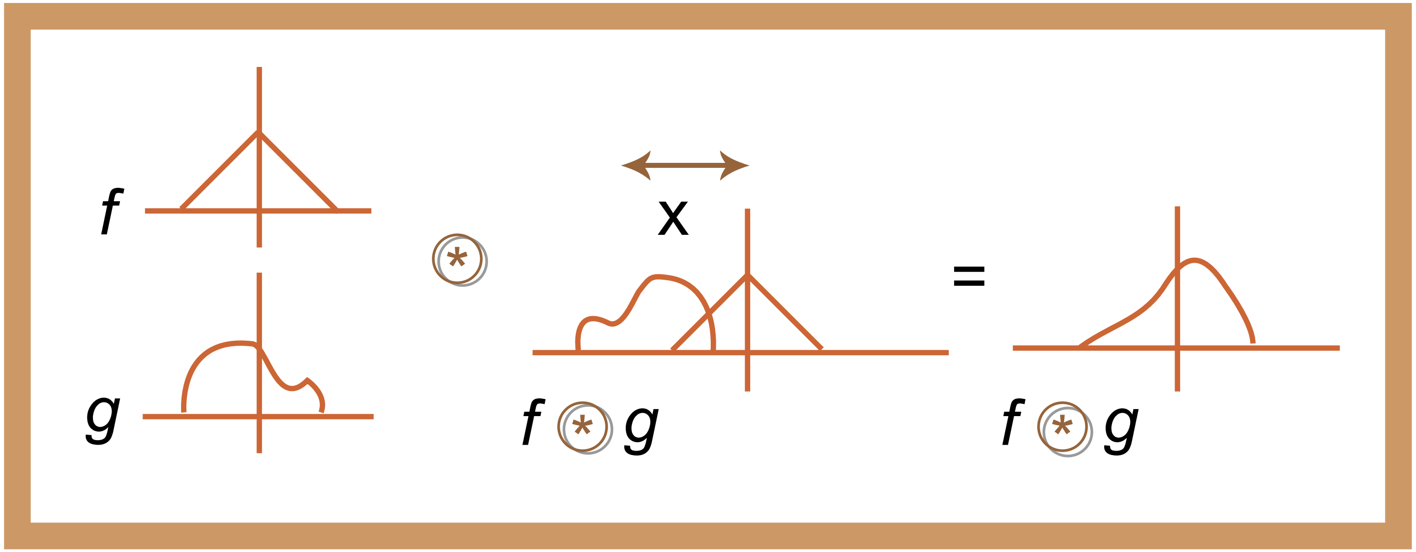
Inverse Fourier Transform

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\, e^{i2\pi(ux+vy)}\, du\, dv$$

# Remember Convolution?

Convolution describes how a system with impulse response, $h(x)$, reacts to a signal, $f(x)$.

$$f(x) * h(x) = \int_{-\infty}^{\infty} f(\lambda) h(x-\lambda) d\lambda$$



# Remember Convolution?

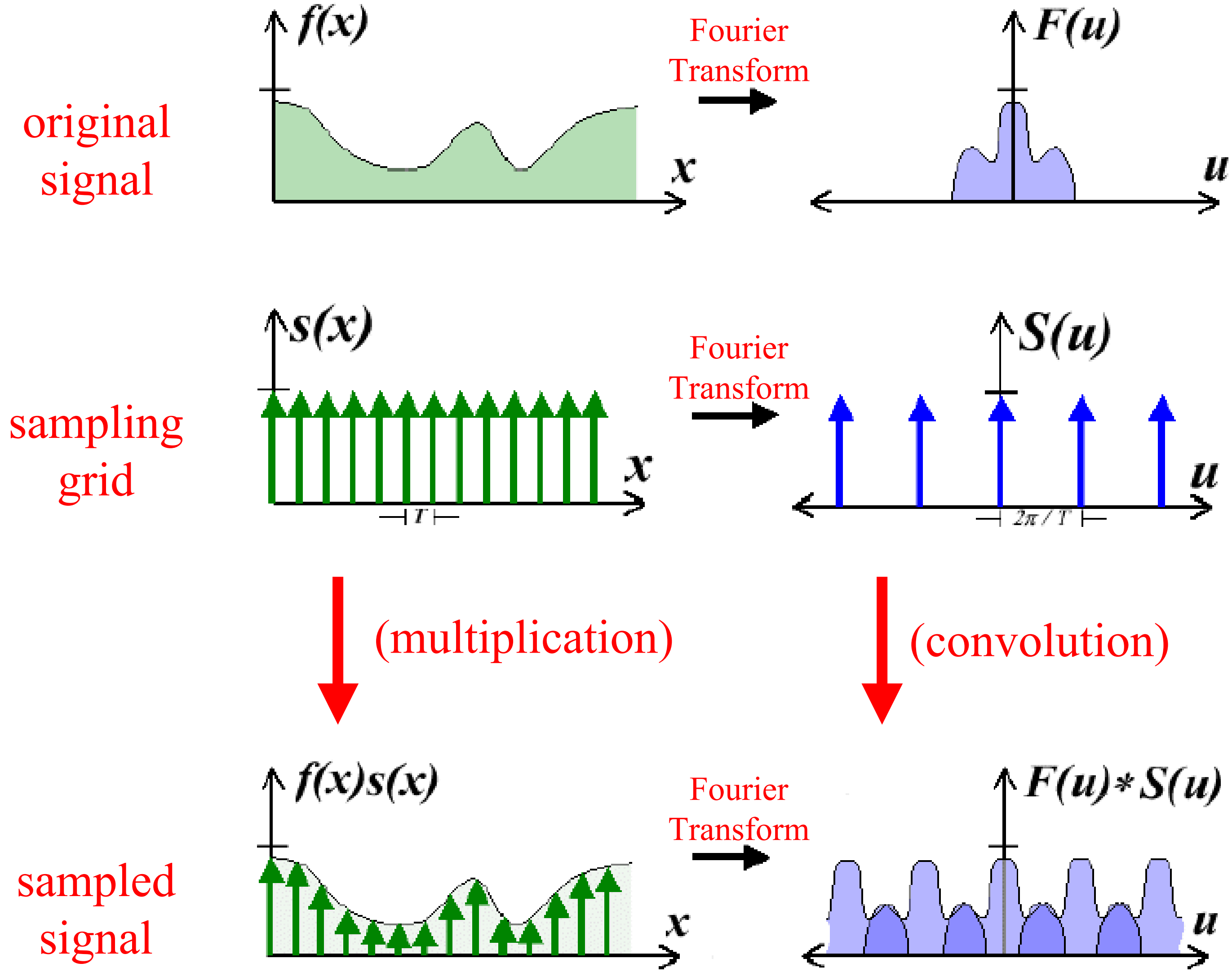
- Some operations that are difficult to compute in the spatial domain can be simplified by transforming to its dual representation in the frequency domain.
- For example, convolution in the spatial domain is the same as multiplication in the frequency domain.

$$f(x) * h(x) \rightarrow F(u)H(u)$$

- And, convolution in the frequency domain is the same as multiplication in the spatial domain

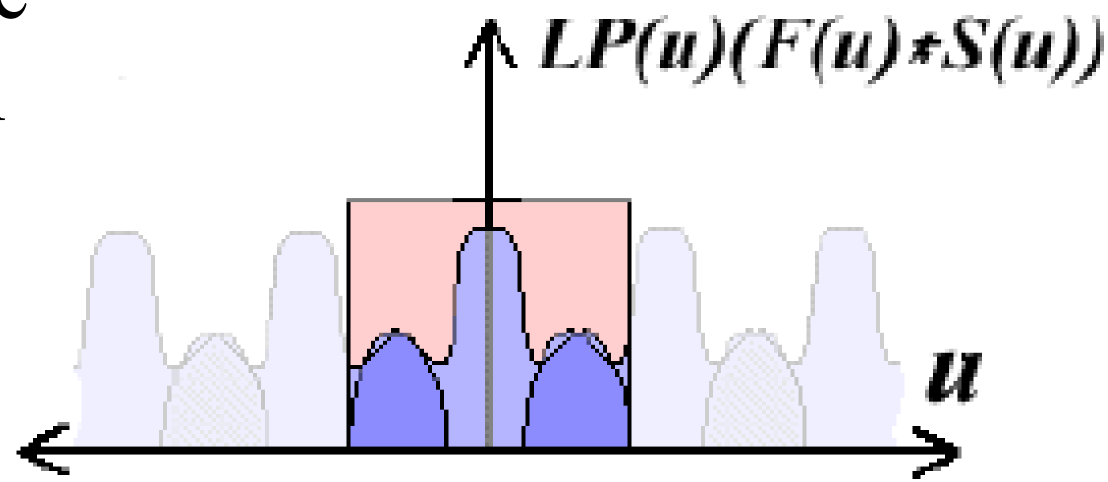
$$F(u) * H(u) \rightarrow f(x)h(x)$$

# Sampling in the Frequency Domain



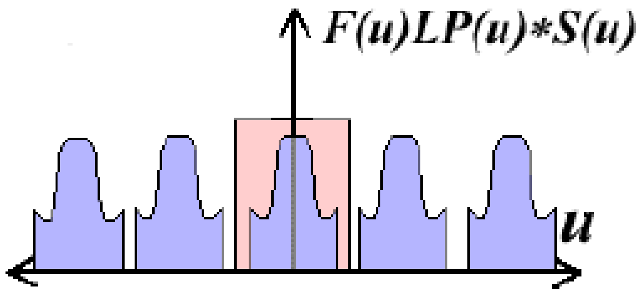
# Reconstruction

- If we can extract a copy of the original signal from the frequency domain of the sampled signal, we can reconstruct the original signal!
- But there may be overlap between the copies.

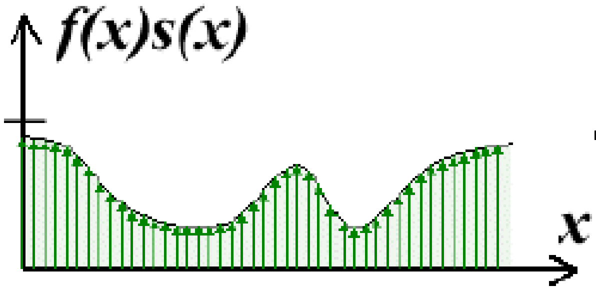


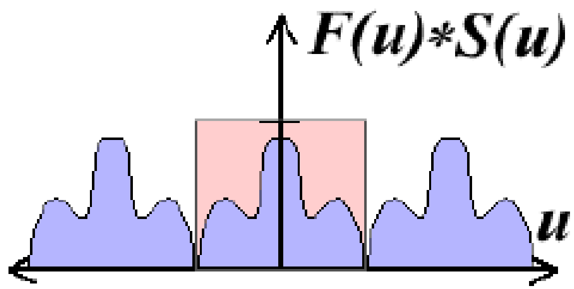
# Guaranteeing Proper Reconstruction

- Separate by removing high frequencies from the original signal (low pass pre-filtering)




- Separate by increasing the sampling density




- If we can't separate the copies, we will have overlapping frequency spectrum during reconstruction → *aliasing*.

# Questions?

# Today

- What is a Pixel?
- Examples of Aliasing
- Signal Reconstruction
- Reconstruction Filters
  - Pre-Filtering, Post-Filtering
  - Ideal, Gaussian, Box, Bilinear, Bicubic
- Anti-Aliasing for Texture Maps

# Pre-Filtering

- Filter continuous primitives
- Treat a pixel as an area
- Compute weighted amount of object overlap
- What weighting function should we use?

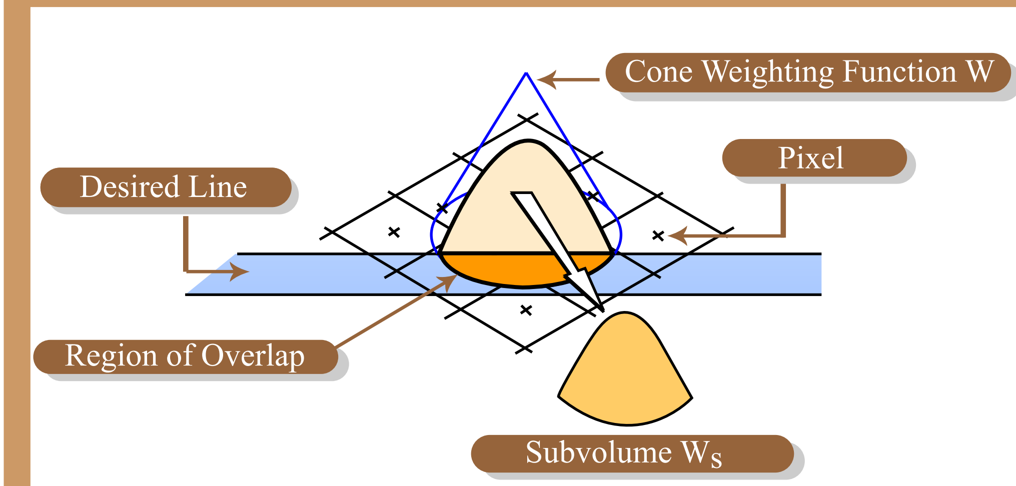



Image adapted from:
Foley et al. *Computer Graphics Principles and Practice*. 2nd Ed. Pearson Addison Wesley, June 1990. ASIN: 0201121107

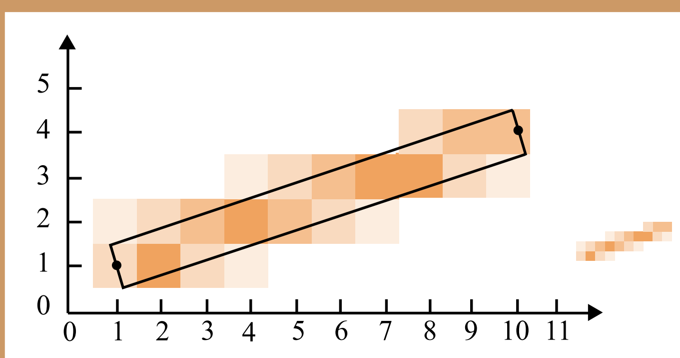
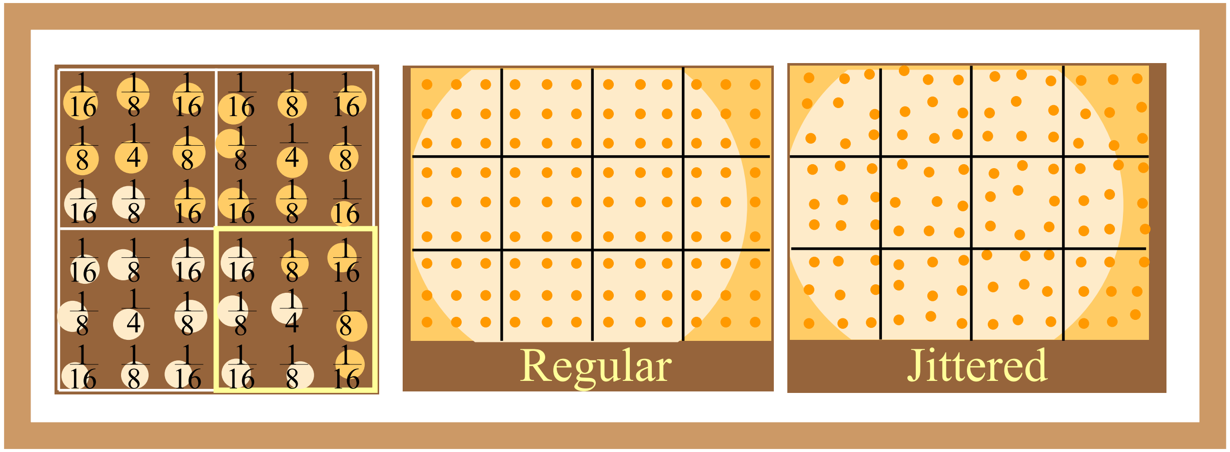



Image adapted from:
Foley et al. *Computer Graphics Principles and Practice*. 2nd Ed. Pearson Addison Wesley, June 1990. ASIN: 0201121107

# Post-Filtering

- Filter samples
- Compute the weighted average of many samples
- Regular or jittered sampling (better)



# Reconstruction Filters

- Weighting function
- Area of influence often bigger than "pixel"
- Sum of weights = 1
  - Each pixel contributes the same total to image
  - Constant brightness as object
    moves across the screen.
- No negative weights/colors (optional)

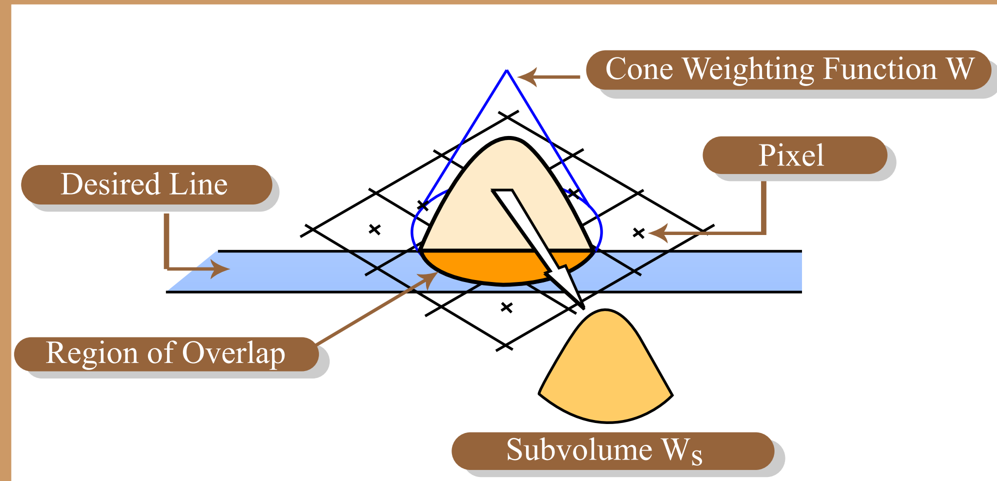
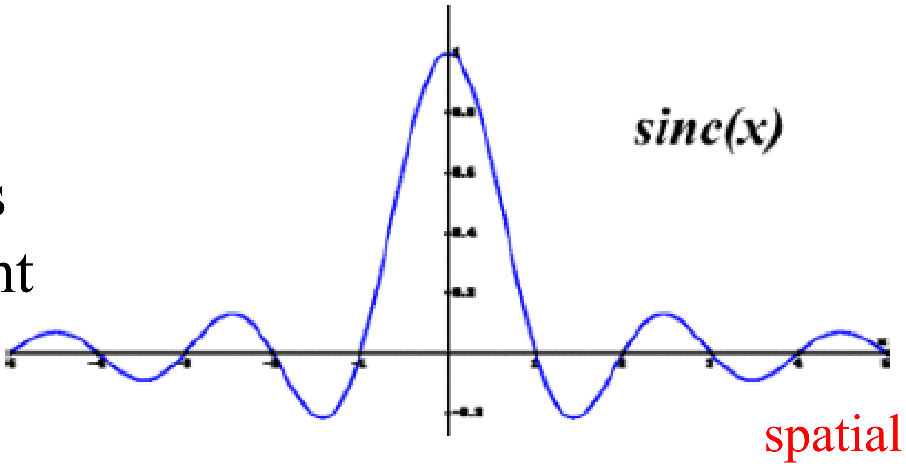
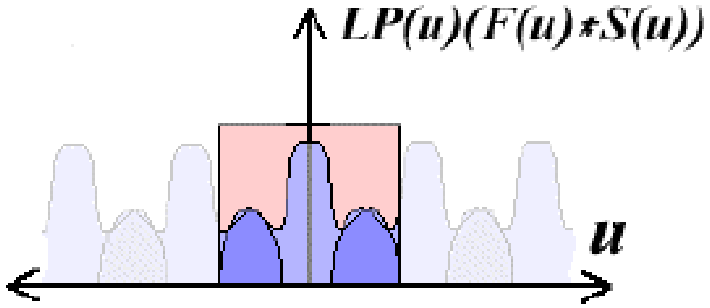


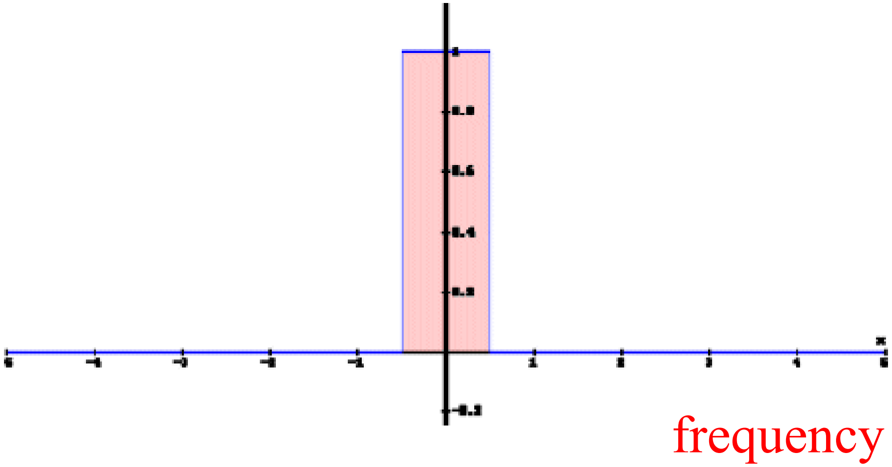

Image adapted from:
Foley et al. *Computer Graphics Principles and Practice*. 2nd Ed. Pearson Addison Wesley, June 1990.
ASIN: 0201121107

# The Ideal Reconstruction Filter

- Unfortunately it has *infinite* spatial extent
  - Every sample contributes to every interpolated point
- Expensive/impossible to compute

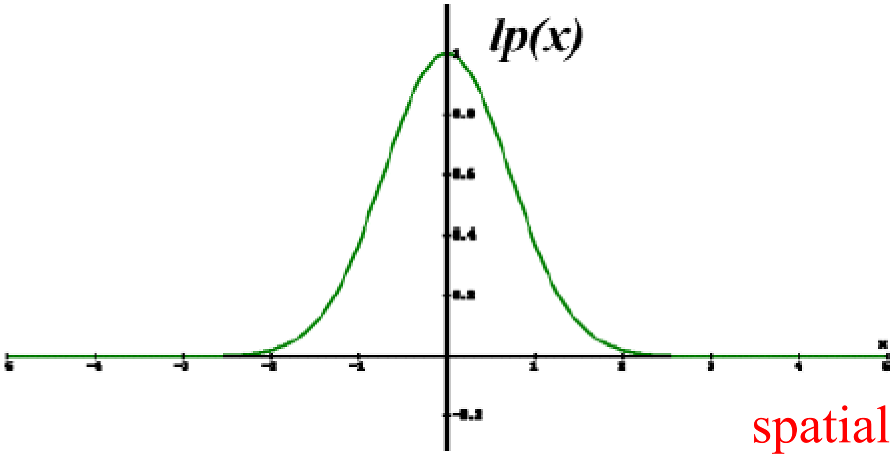


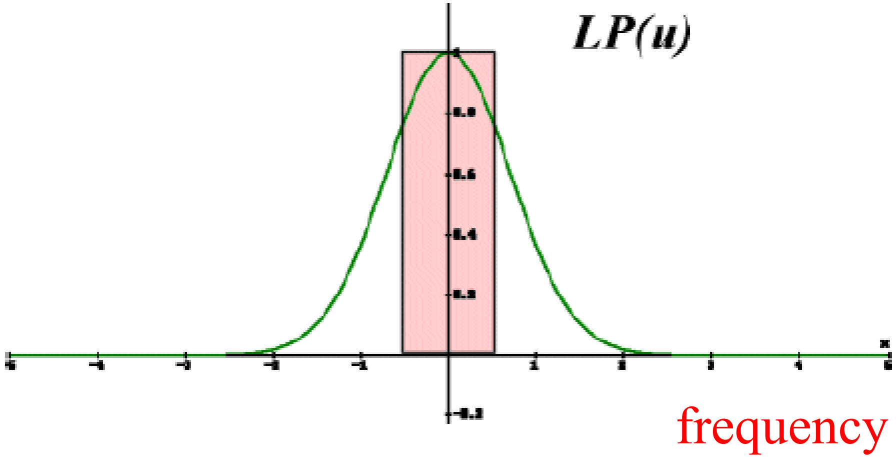



# Gaussian Reconstruction Filter

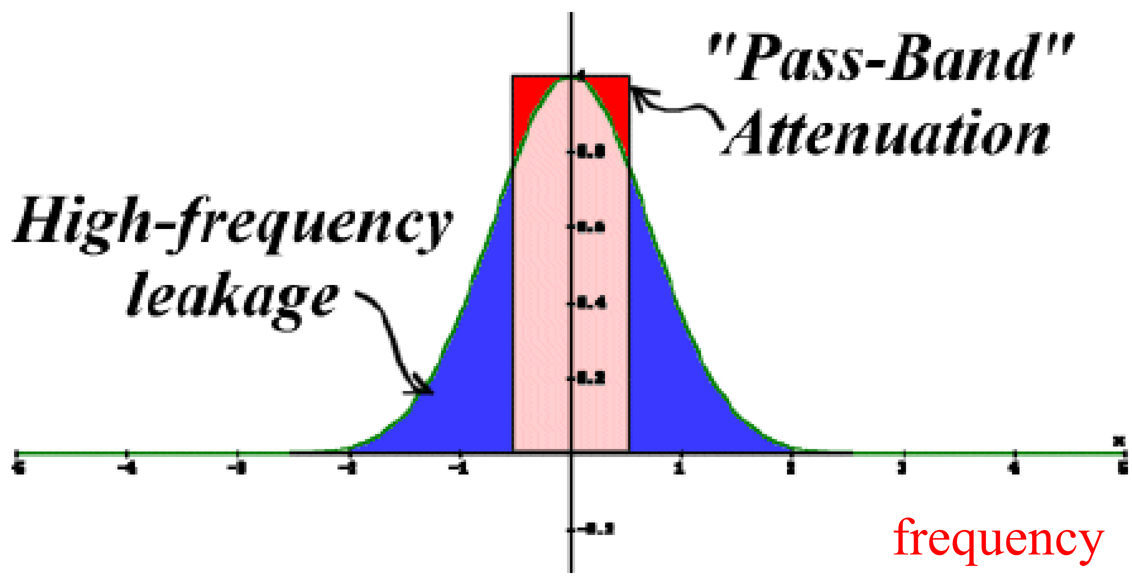
- This is what a CRT does for free!



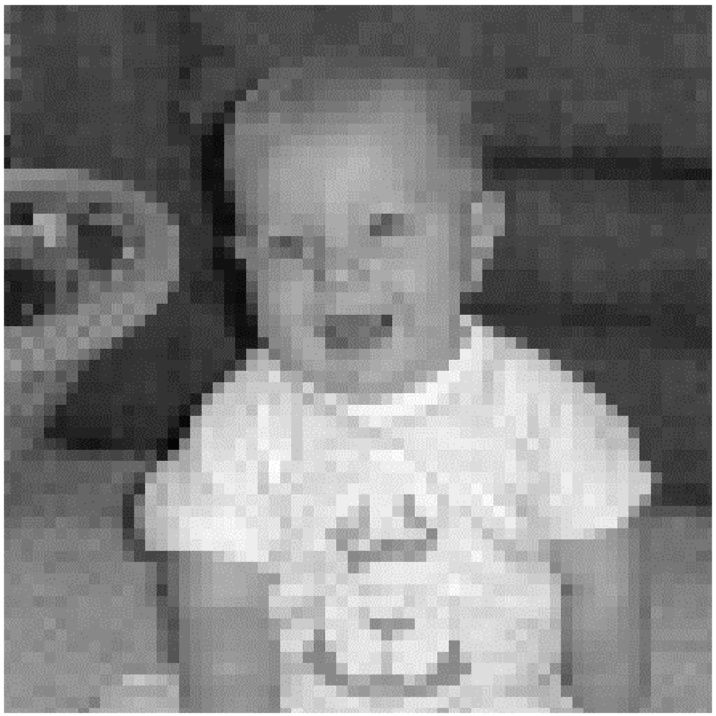
# Problems with Reconstruction Filters

- Many visible artifacts in re-sampled images are caused by poor reconstruction filters
- Excessive pass-band attenuation results in blurry images
- Excessive high-frequency leakage causes "ringing" and can accentuate th sampling grid
 (anisotropy)

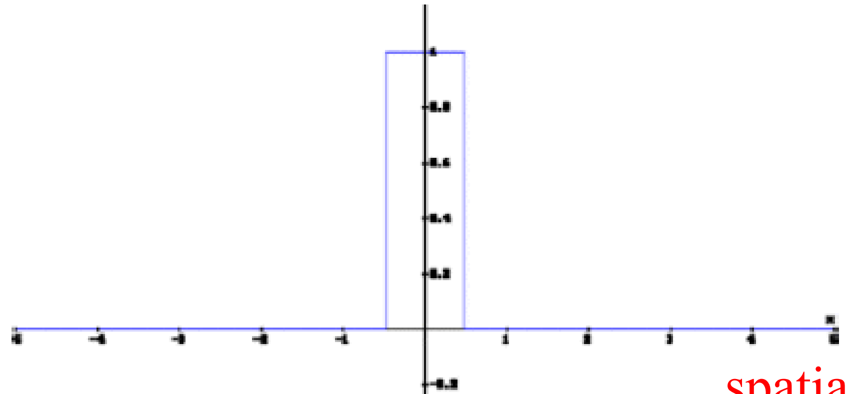


# Box Filter / Nearest Neighbor

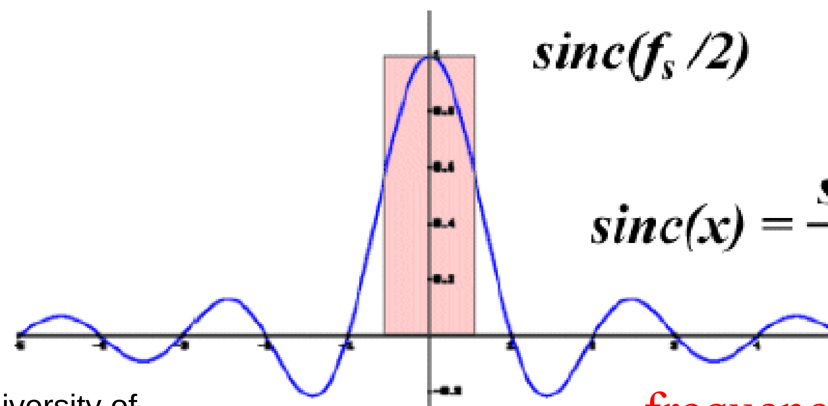
- Pretending pixels are little squares.



Courtesy of Leonard McMillan, Computer Science at the University of North Carolina in Chapel Hill. Used with permission.



spatial



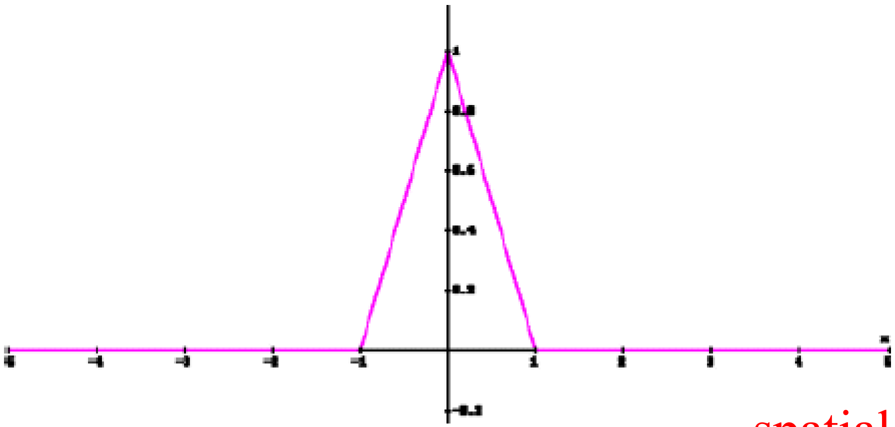
$sinc(f_s/2)$

$$sinc(x) = \frac{\sin(\pi x)}{\pi x}$$

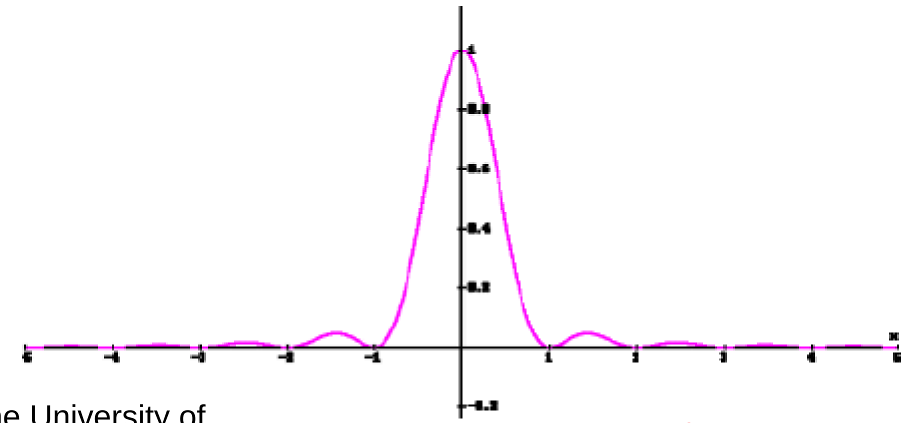
frequency

# Tent Filter / Bi-Linear Interpolation

- Simple to implement
- Reasonably smooth



spatial

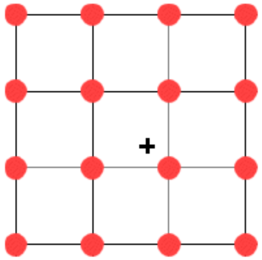


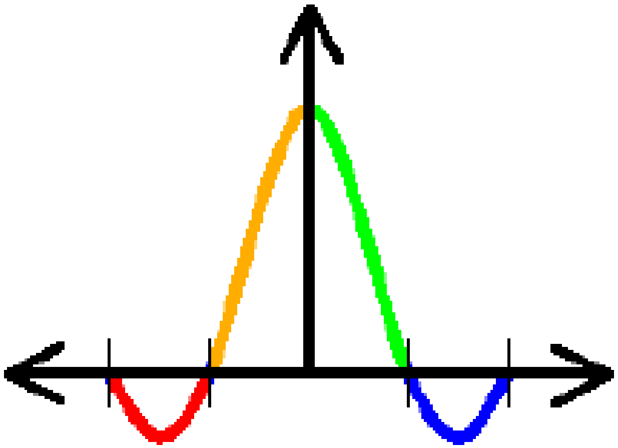
frequency

Courtesy of Leonard McMillan, Computer Science at the University of North Carolina in Chapel Hill. Used with permission.

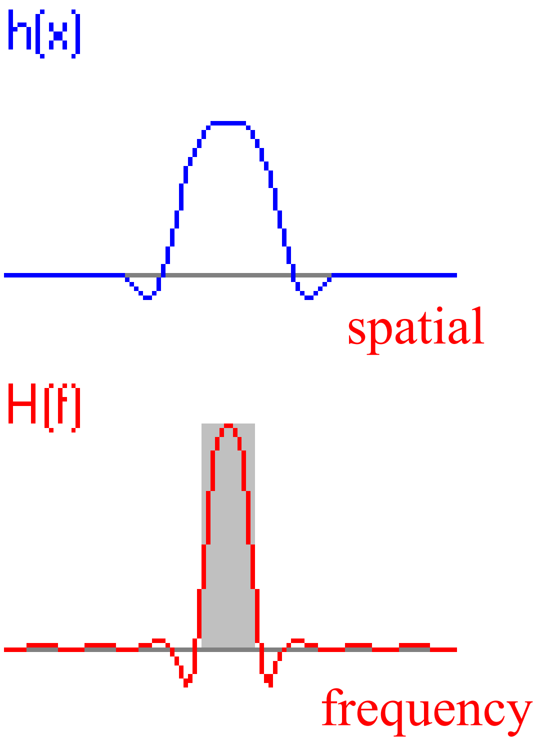
# Bi-Cubic Interpolation

- Begins to approximate the ideal spatial filter, the sinc function



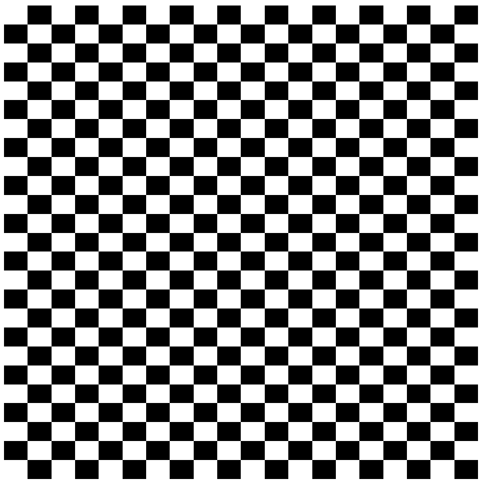


Courtesy of Leonard McMillan, Computer Science at the University of North Carolina in Chapel Hill. Used with permission.

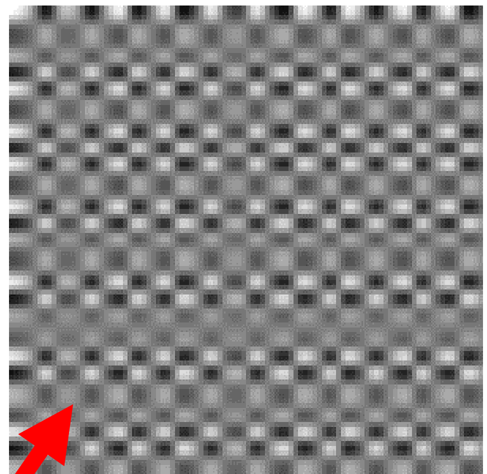


# Why is the Box filter bad?

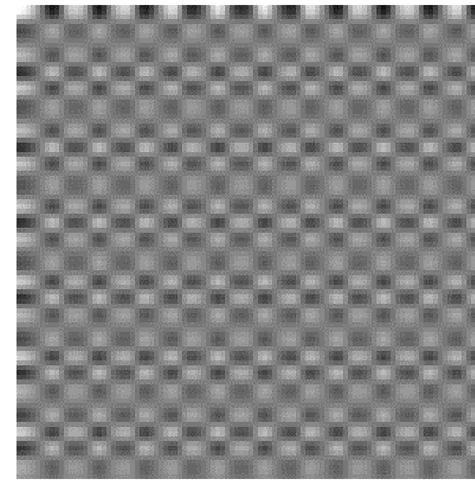
- (Why is it bad to think of pixels as squares)



Original high-resolution image

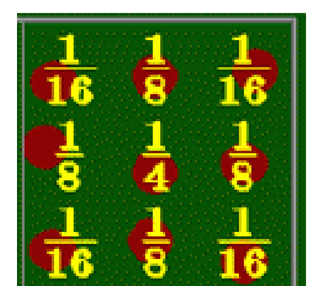


Down-sampled with a 5x5 box filter (uniform weights)



Down-sampled with a 5x5 Gaussian filter (non-uniform weights)

notice the ugly horizontal banding




Courtesy of Leonard McMillan, Computer Science at the University of North Carolina in Chapel Hill. Used with permission.

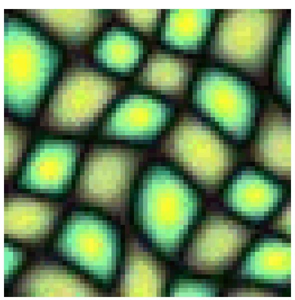
# Questions?

# Today

- What is a Pixel?
- Examples of Aliasing
- Signal Reconstruction
- Reconstruction Filters
- Anti-Aliasing for Texture Maps
  – Magnification & Minification
  – Mipmaps
  – Anisotropic Mipmaps

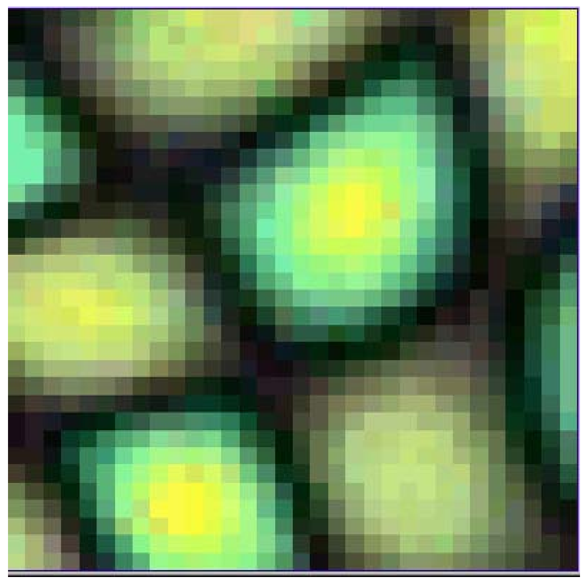
# Sampling Texture Maps

- When texture mapping it is rare that the screen-space sampling density matches the sampling density of the texture.

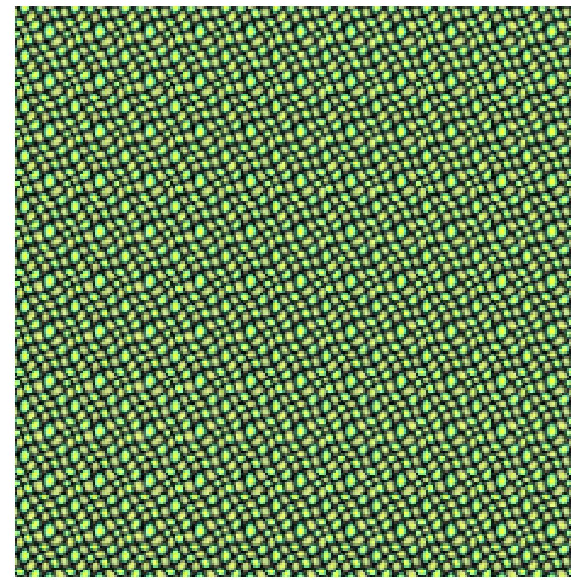


64x64 pixels

Original Texture



Magnification for Display

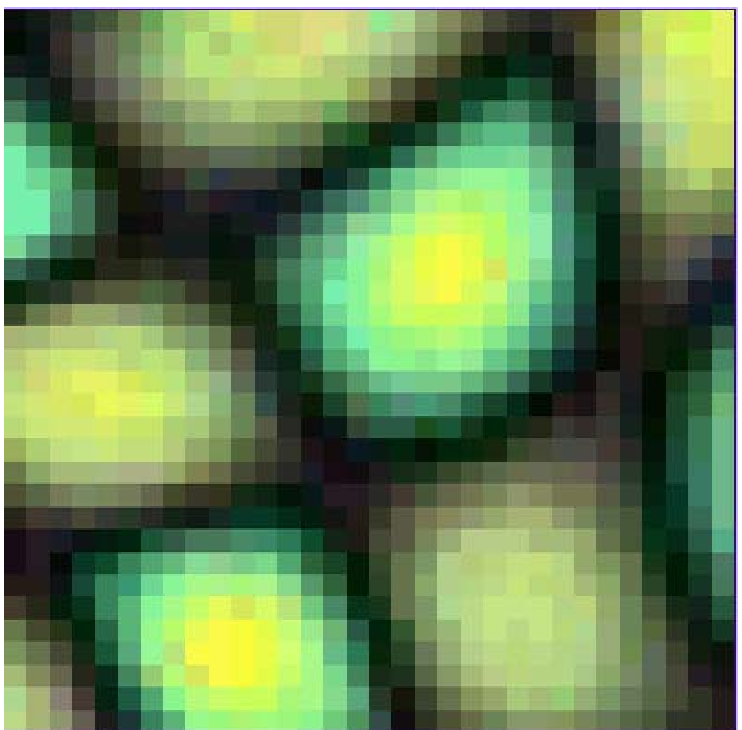


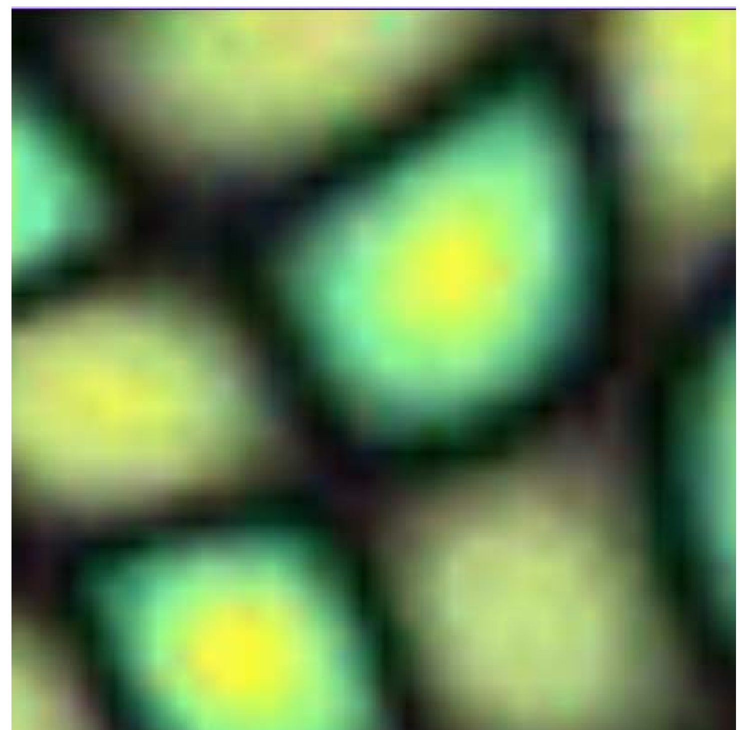
Minification for Display

for which we must use a reconstruction filter

# Linear Interpolation

- Tell OpenGL to use a tent filter instead of a box filter.
- Magnification looks better, but blurry
  - (texture is under-sampled for this resolution)





# Spatial Filtering

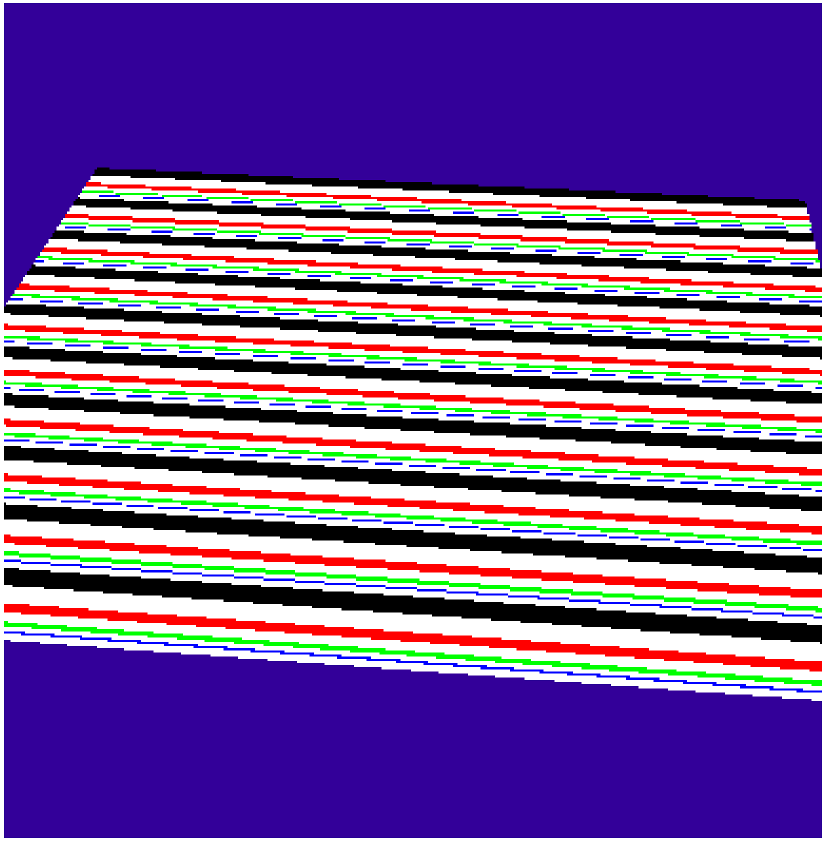
- Remove the high frequencies which cause artifacts in minification.
- Compute a spatial integration over the extent of the sample
- Expensive to do during rasterization, but it can be precomputed

# MIP Mapping

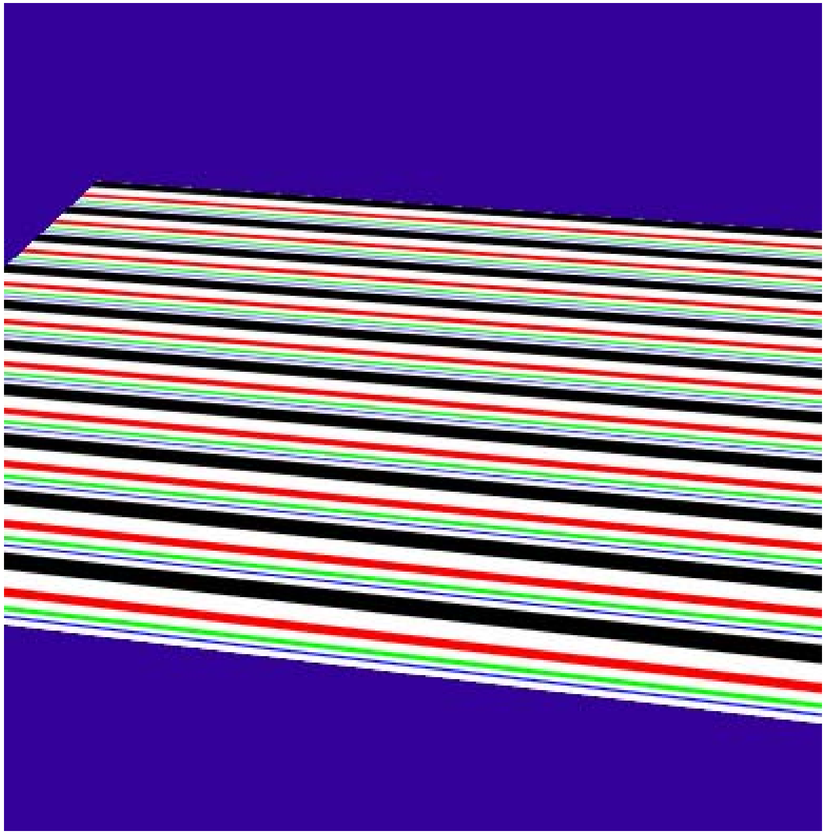
- Construct a pyramid of images that are pre-filtered and re-sampled at 1/2, 1/4, 1/8, etc., of the original image's sampling
- During rasterization we compute the index of the decimated image that is sampled at a rate closest to the density of our desired sampling rate
- MIP stands for *multium in parvo* which means *many in a small place*

# MIP Mapping Example

- Thin lines may become disconnected / disappear



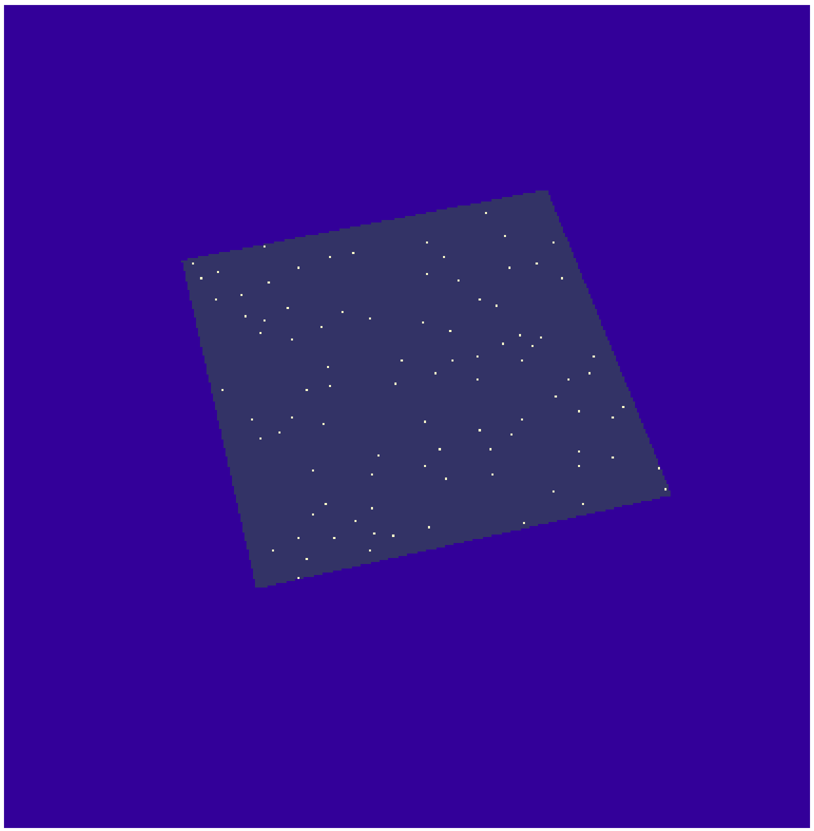
Nearest Neighbor



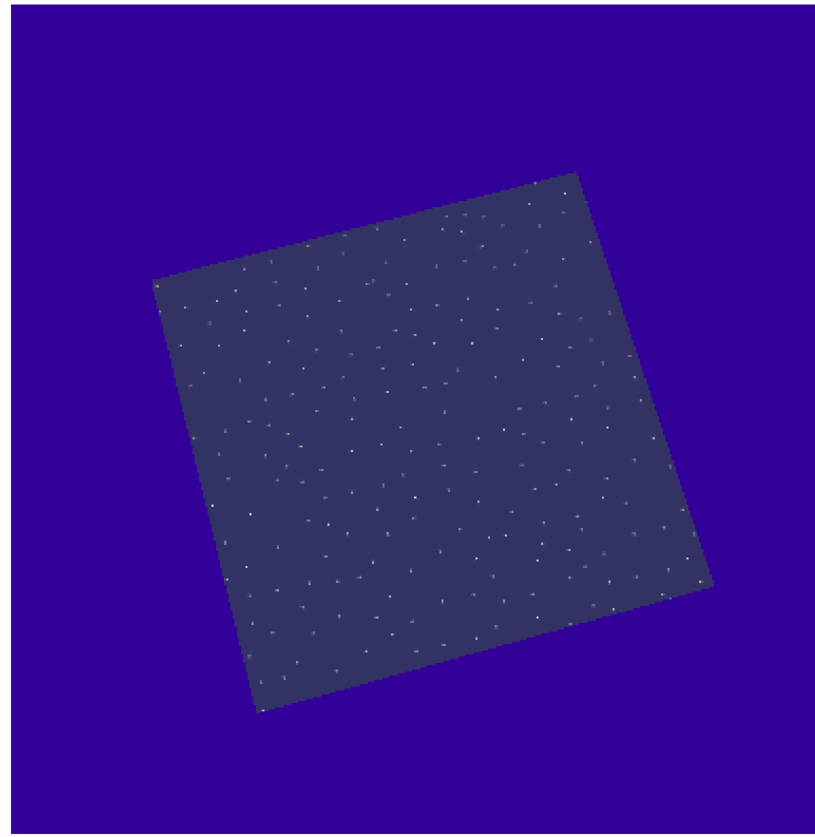
MIP Mapped (Bi-Linear)

# MIP Mapping Example

- Small details may "pop" in and out of view



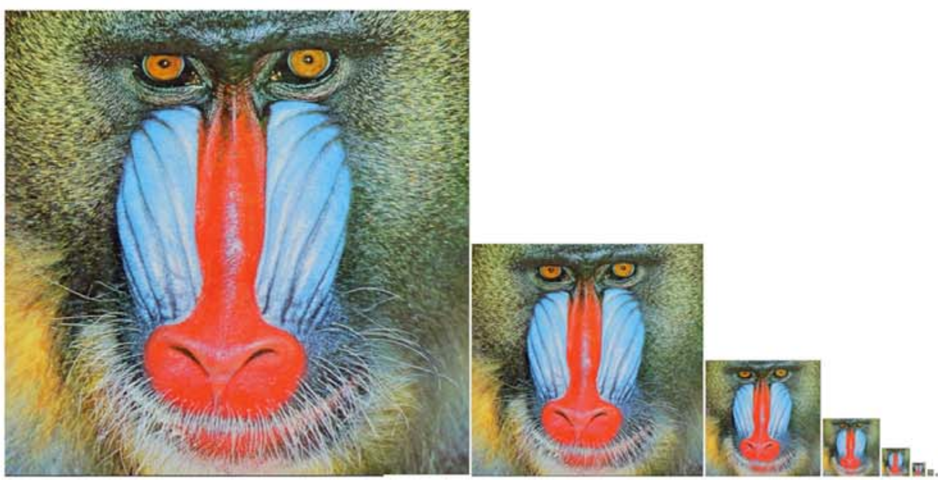
Nearest Neighbor



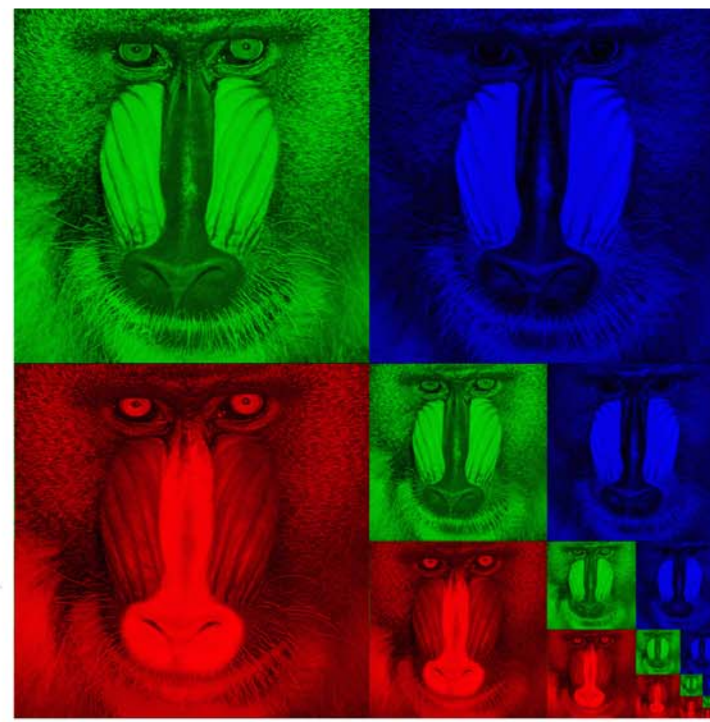
MIP Mapped (Bi-Linear)

# Storing MIP Maps

- Can be stored compactly
- Illustrates the 1/3 overhead of maintaining the MIP map



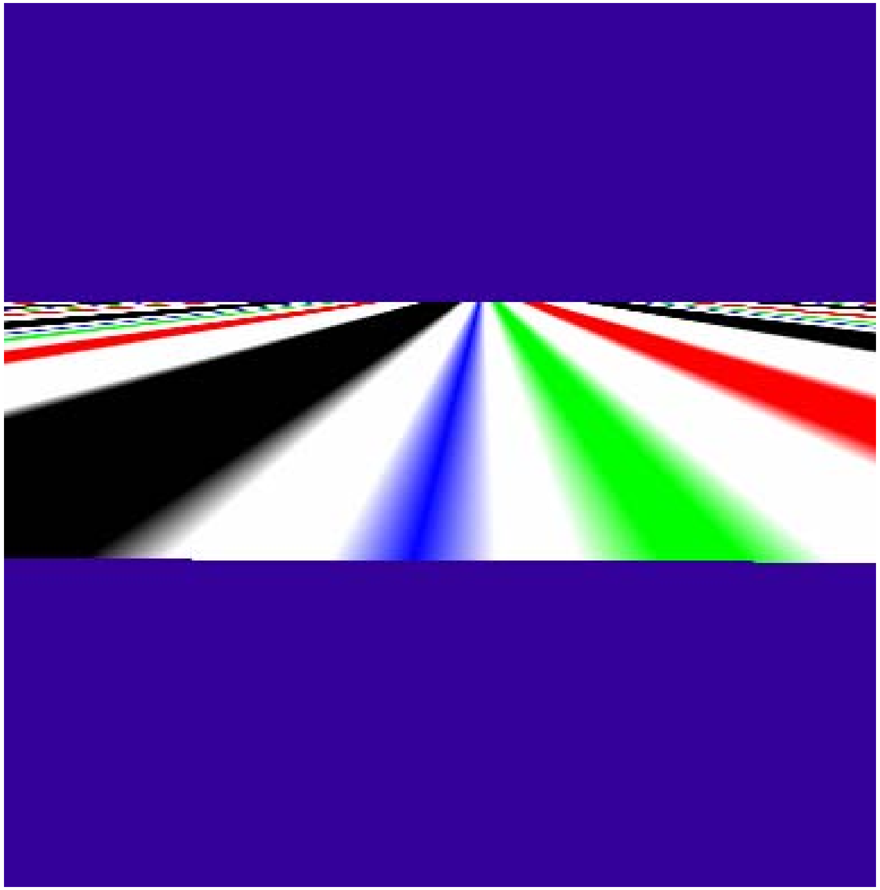
10-level mip map



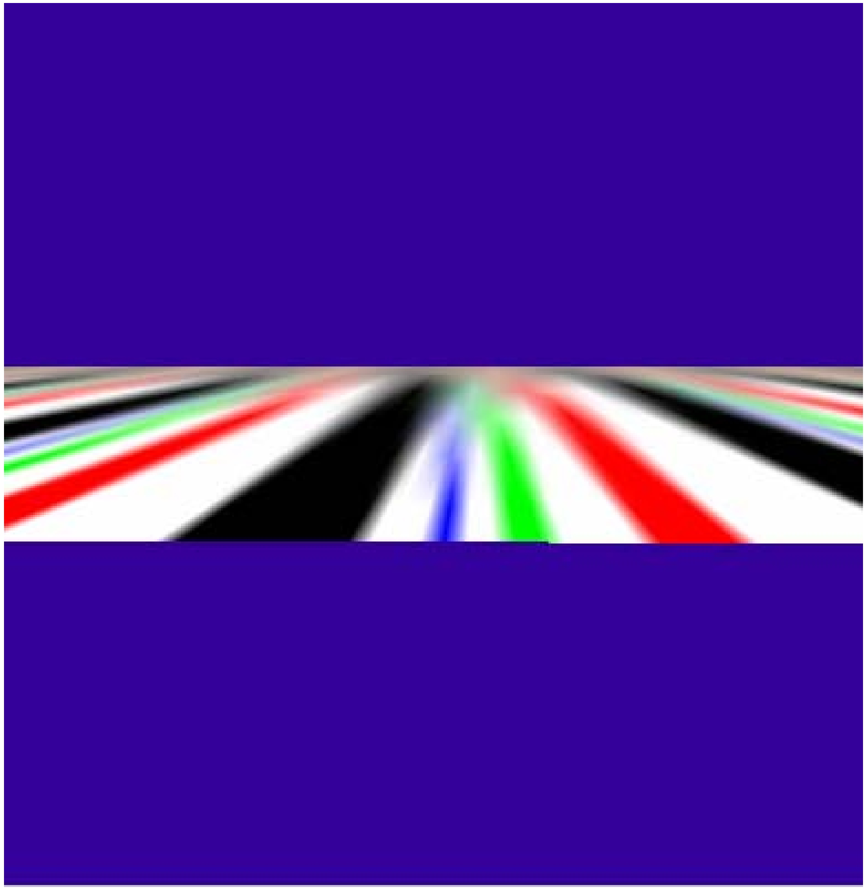
Memory format of a mip map

# Anisotropic MIP-Mapping

- What happens when the surface is tilted?



Nearest Neighbor



MIP Mapped (Bi-Linear)

# Anisotropic MIP-Mapping

- We can use different mipmaps for the 2 directions
- Additional extensions can handle non axis-aligned views

Image removed due to copyright considerations.

Images from http://www.sgi.com/software/opengl/advanced98/notes/node37.html

# Questions?

# Next Time: Last Class!

## Wrap Up &
## Final Project Review