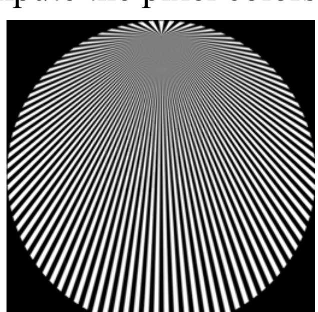
Sampling Theorem & Antialiasing

CS5502 Fall 2006
© Chun-Fa Chang

Motivations

- "My ray traced images have a lot more pixels than the TV screen. Why do they look like @#$%?"
- How to compute the pixel colors for the following pattern?



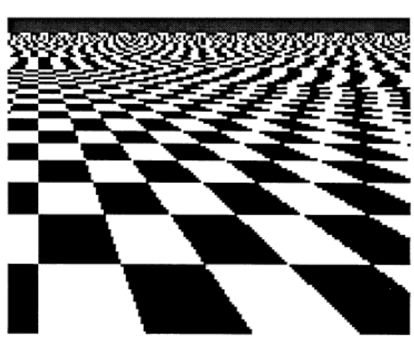
Antialiasing with Line Samples
Rendering Techniques '00 (Proceedings of the 11th Eurographics Workshop on Rendering), pp. 197-205
Thouis R. Jones, Ronald N. Perry

CS5502 Fall 2006
© Chun-Fa Chang

Part I: Sampling Theorem

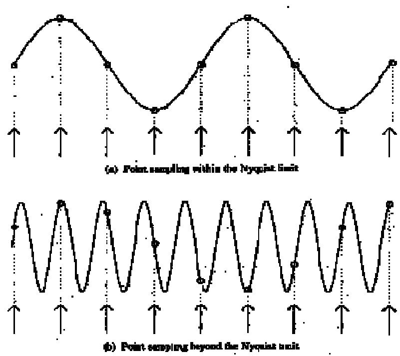
© Chun-Fa Chang

Example of Aliasing in Computer Graphics



© Chun-Fa Chang

Examples of Aliasing in 1D

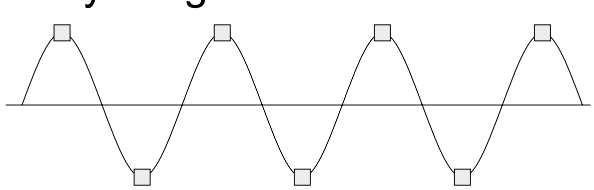


- See Figure 14.2 (p.394) of Watt's book for other examples.

CS5502 Fall 2006
© Chun-Fa Chang

An Intuition – Using a Single Frequency

- It's easy to figure out for a sin wave.

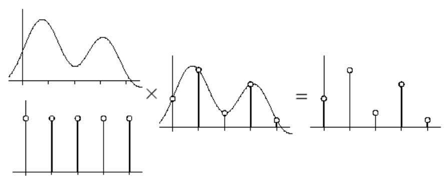


- What about any signal (usually a mixture of multiple frequencies)?
- Enter Fourier Transform…

CS5502 Fall 2006
© Chun-Fa Chang

Sampling

- 1D Signal: x → f(x) becomes i → f(i)

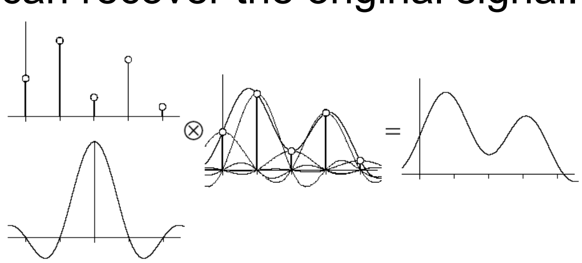


- 2D Image: x, y → f(x, y)
  - For grayscale image, f(x, y) is the intensity of pixel at (x, y).

© Chun-Fa Chang

Reconstruction

- If the samples are “dense” enough, then we can recover the original signal.

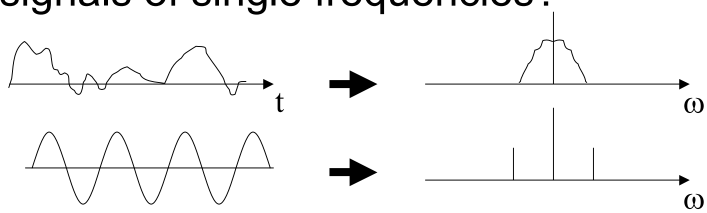


- Question is: How dense is enough?

© Chun-Fa Chang

Fourier Transform

- Can we separate signal into a set of signals of single frequencies?



$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-j\omega x} dx \qquad e^{-j\omega x} = \cos\omega + j\sin\omega$$

CS5502 Fall 2006
© Chun-Fa Chang

Basis Functions

- An example:

  $X=[x_1, x_2, \ldots, x_n]$

  $U=[u_1, u_2, \ldots, u_n]$

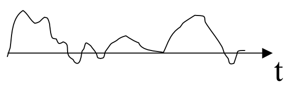
  $V=[v_1, v_2, \ldots, v_n]$

  Let X = a*U + b*V, how to find a and b?
- If U and V are orthogonal, then a and b are the projection of X onto U and V.

CS5502 Fall 2006
© Chun-Fa Chang

Compared to Fourier Transform

- Consider a continuous signal as a infinite-dimensional vector



[f(ε), f(2ε), f(3ε),…..]

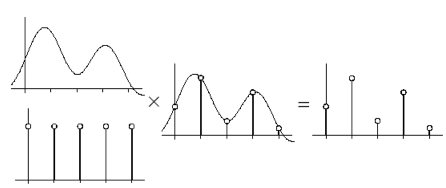
- Consider each frequency ω a basis, then F(ω) is the projection of f(x) onto that basis.

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-j\omega x} dx$$

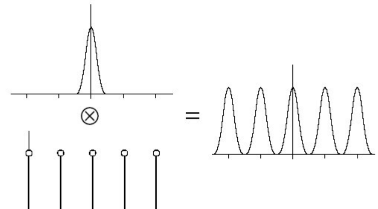
CS5502 Fall 2006
© Chun-Fa Chang

Sampling

- Spatial domain: multiply with a pulse train.



- Frequency domain: convolution!



CS5502 Fall 2006
© Chun-Fa Chang

Convolution

$$h(x) = f(x) \otimes g(x) = \int f(x')g(x - x')dx'$$

- To start with, image that f(x) is nonzero only in the range of [-a, a].
  - Then we only need to consider g(x) in the range of [x-a, x+a]
- Multiplication in spatial domain results in convolution in frequency domain (and vice versa).

© Chun-Fa Chang

An Intuition for Convolution

- Does it make sense to you that multiplication in one domain becomes convolution in the other domain?
- Look at this example:

$$P_1(x) = a_1x^3 + b_1x^2 + c_1x^1 + d_1$$
$$P_2(x) = a_2x^3 + b_2x^2 + c_2x^1 + d_2$$

- What are the coefficients of P_1*P_2?

CS5502 Fall 2006
© Chun-Fa Chang

- Consider x^n, …, x^2, x^1, x^0 as basis.

$$P_1(x) = a_1x^3 + b_1x^2 + c_1x^1 + d_1$$
$$P_2(x) = a_2x^3 + b_2x^2 + c_2x^1 + d_2$$

- Projections of P1 and P2 to the basis are (a_1, b_1, c_1, d_1) and (a_2, b_2, c_2, d_2)
- P1(x)*P2(x) results in: $(a_1, b_1, c_1, d_1) \otimes (a_2, b_2, c_2, d_2)$ in the transformed space.

© Chun-Fa Chang

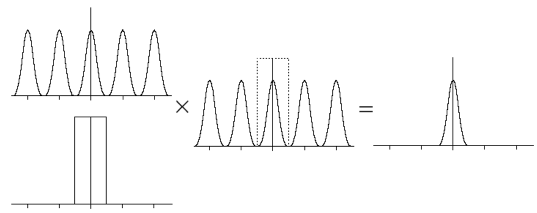
- The fact is: you have been doing convolution since elementary school!
- Example: 222*111 is computed as
  $(2,2,2) \otimes (1,1,1)$

$$
\begin{array}{rccccc}
\multicolumn{6}{l}{222 \times 111 =} \\
 & & & 2 & 2 & 2 \\
 & & 2 & 2 & 2 & \\
+ & 2 & 2 & 2 & & \\
\hline
 & 2 & 4 & 6 & 4 & 2
\end{array}
$$

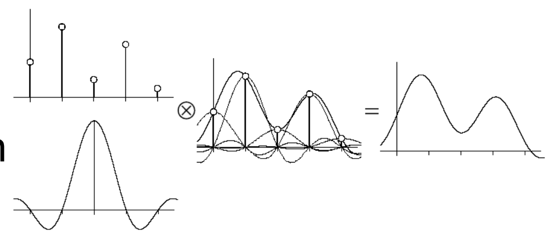
© Chun-Fa Chang

Reconstruction

- Frequency domain:



- Spatial domain: convolve with Sinc function



CS5502 Fall 2006
© Chun-Fa Chang

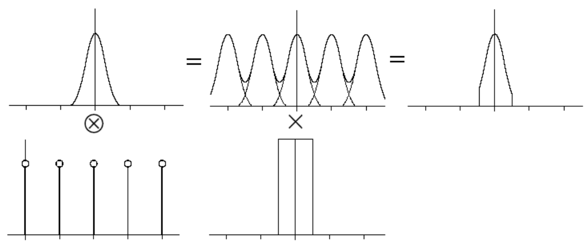
Reconstruction Kernel

- For perfect reconstruction, we need to convolve with the sinc function.
  - It's the Fourier transform of the box function.
  - It has infinite "support"
- May be approximated by Gaussian, cubic, or even triangle "tent" function.

CS5502 Fall 2006
© Chun-Fa Chang

Nyquist Limit

- Nyquist Limit = 2 * max_frequency
- Undersampling: sampling below the Nyquist Limit.



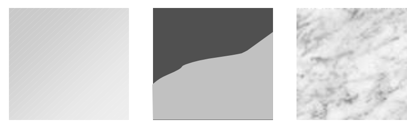
CS5502 Fall 2006
© Chun-Fa Chang

Part II: Antialiasing

CS5502 Fall 2006
© Chun-Fa Chang

Changes within a Pixel

- A lot can change within a pixel:
  - Shading
  - Edge
  - Texture

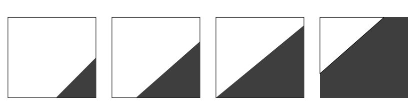


- Point sampling at the center often produces undesirable result.

© Chun-Fa Chang

Pixel Coverage

- What should be the pixel colors for these?



- Can we simply use the covered areas of blue and white? (Hint: convolve with box filter.)
- Do we have enough data to compute the coverage?

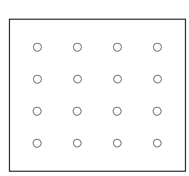
© Chun-Fa Chang

Antialiasing

- Consider a ray tracer. Is it often impossible to find the partial coverage of an edge.
- Each ray is a point sample.
- We may use many samples for each pixel → slower performance.

Antialiasing – Uniform Sampling

- Also called supersampling



- Wasteful if not much changes within a pixel.

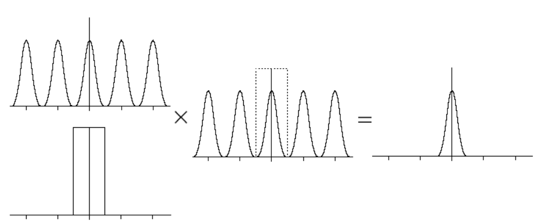
Filtering

- How do we reduce NxN supersamples into a pixel?
  - Average?
  - More weight near the center?
- Let's resort to the sampling theorem.

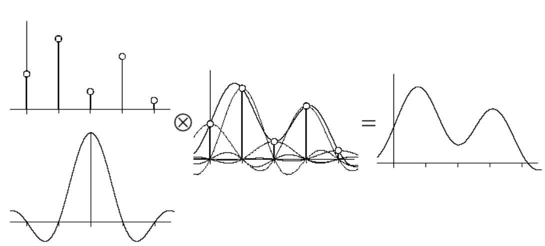
CS5502 Fall 2006
© Chun-Fa Chang

Reconstruction

- Frequency domain:



- Spatial domain:



CS5502 Fall 2006
© Chun-Fa Chang

A Few Observations

- In theory, a sample influences not only its pixel, but also every pixels in the image.
- What does it mean by removing high frequencies?

© Chun-Fa Chang

Other Than Uniform Sampling?

- So far, the extra samples are taken uniformly in screen space.
- Other ways to take extra samples:
  - Adaptive sampling
  - Stochastic (or randomized) sampling

© Chun-Fa Chang

Antialiasing – Adaptive Sampling

- Feasible in software, but difficult to implement in hardware.
- Increase samples only if necessary.
- But how do we know when is "necessary"?
  - Check the neighbors.

© Chun-Fa Chang

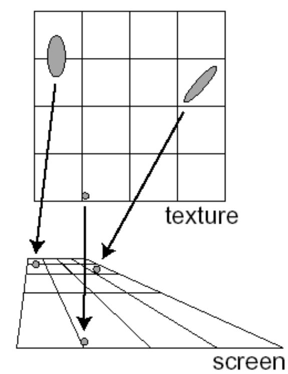
Antialiasing – Stochastic Sampling

- Keep the same number of samples per pixel.
- Replace the aliasing effects with noise that is easier to ignore.

© Chun-Fa Chang

EWA for Texture Mapping

- Paul Heckbert, "Survey of Texture Mapping" IEEE CG&A, Nov. 1986. (Figures)
- Green & Heckbert, "Creating Raster Omnimax Images from Multiple Perspective Views Using The Elliptical Weighted Average Filter" IEEE CG&A, 6(6), pp. 21-27, June 1986.



CS5502 Fall 2006
© Chun-Fa Chang