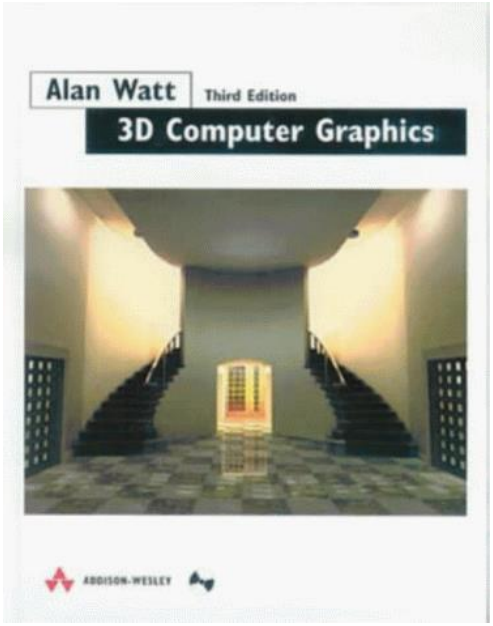
The University Of Sheffield.

COM3503/4503/6503: 3D Computer Graphics

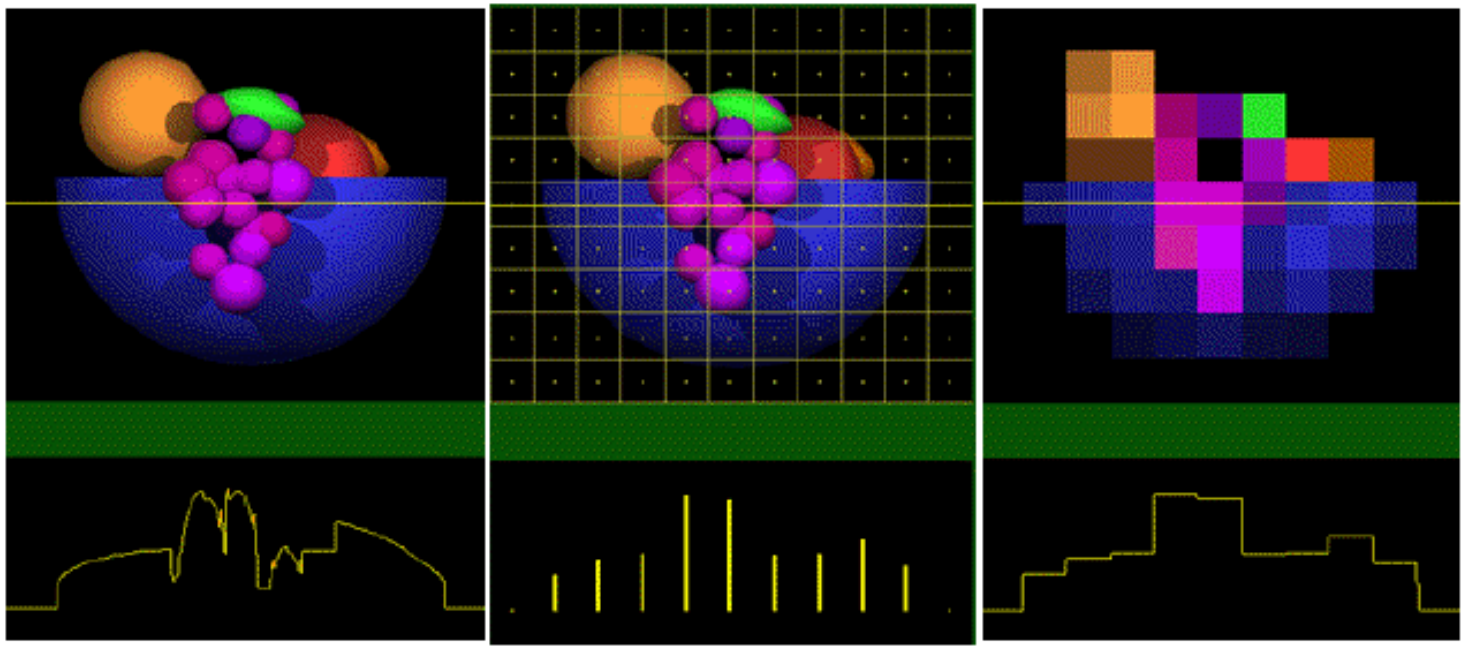
Lecture 11: Anti-aliasing



Dr. Steve Maddock
Room G011, Regent Court
s.maddock@sheffield.ac.uk

1. Introduction

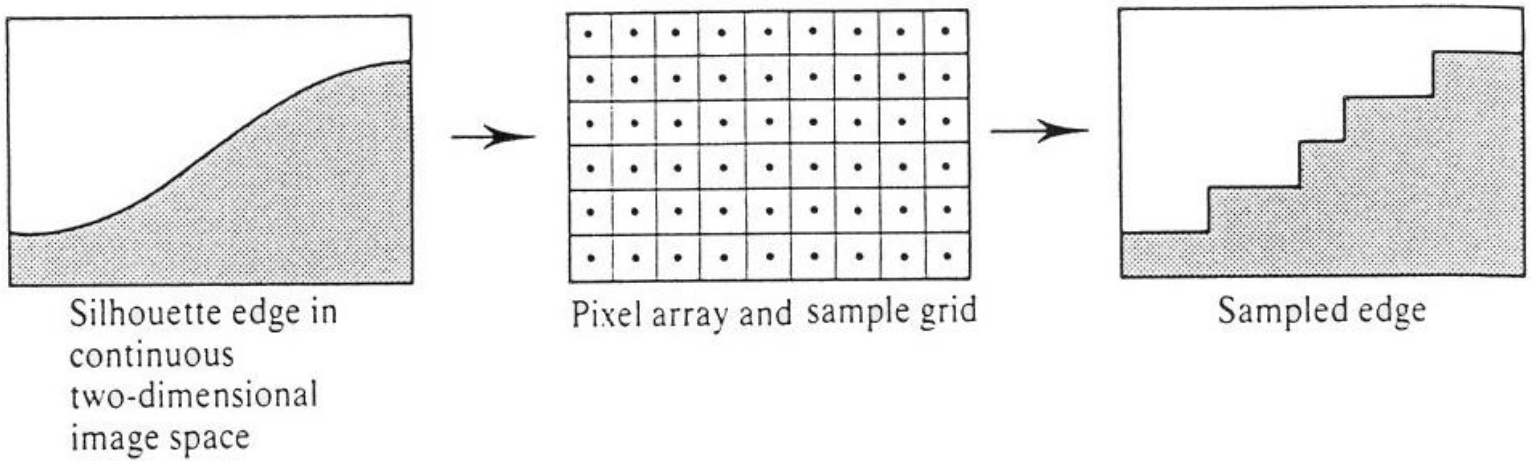
- World coordinate system – floating point
- Screen coordinate system – (x,y) integer space (and f.pt. z depth)
- Sample a continuous function using discrete (regular) locations:



Durand, F., 6.837 Intro to Computer Graphics, MIT, 2006, http://people.csail.mit.edu/fredo/

1.1 Artefacts that need to be addressed

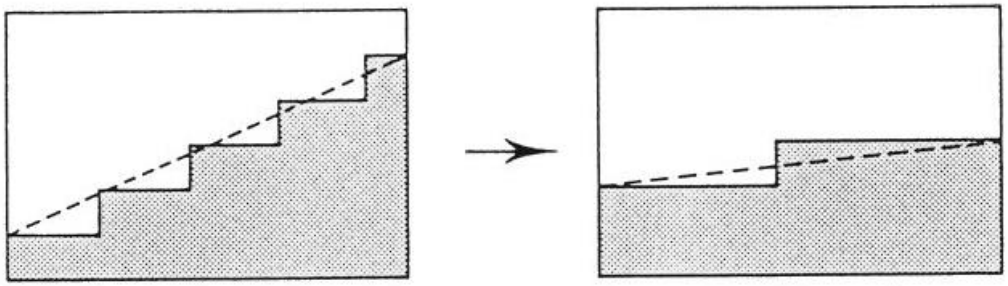
- Jagged silhouette edges – produces ‘crawling’ in animation



Silhouette edge in continuous two-dimensional image space

Pixel array and sample grid

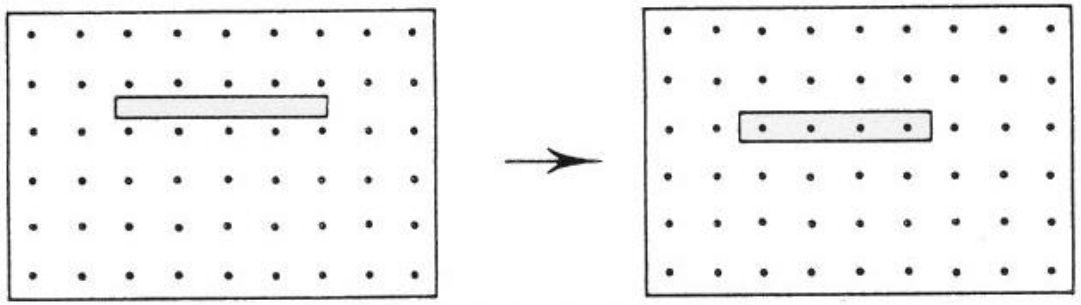
Sampled edge



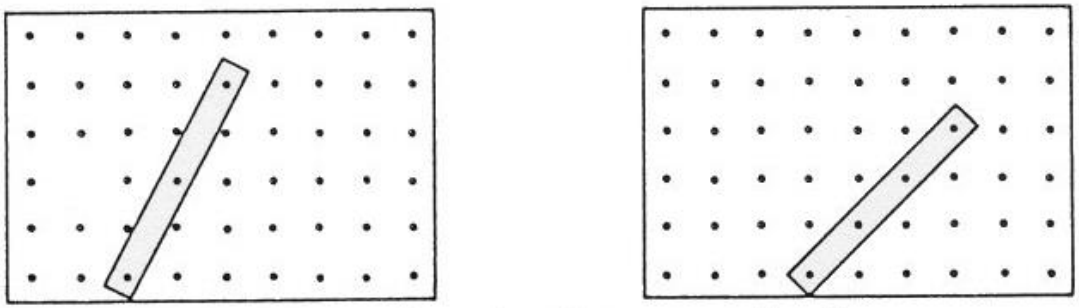
Jagged edge patterns change depending on line orientation

1.1 Artefacts that need to be addressed

- Small objects are missed – produces ‘scintillation’ in animation



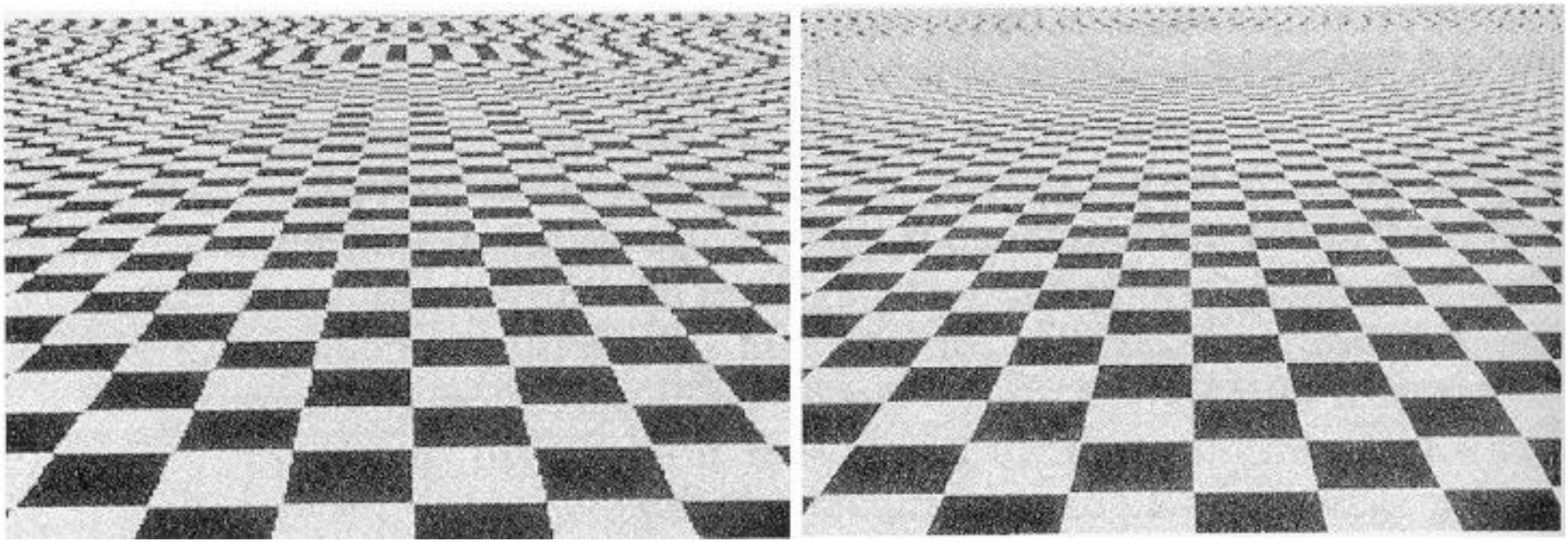
Long thin objects may completely disappear



Long thin objects break up unpredictably

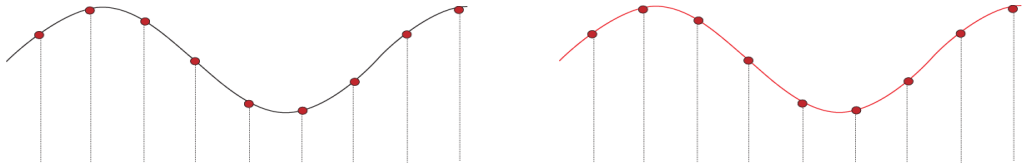
1.1 Artefacts that need to be addressed

- Texture breaks up in perspective views – produces Moiré patterns in animation



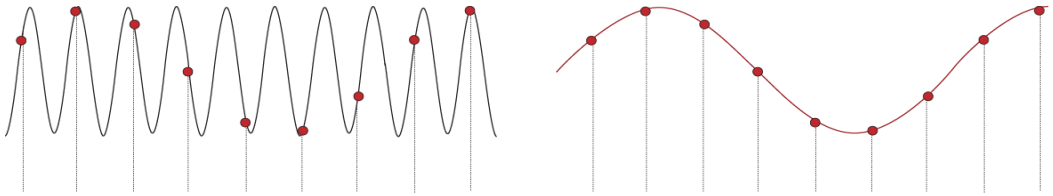
2. Sampling

- If sampling density is high enough, a signal can be reconstructed
  - Sample curve (the signal) at discrete (regular) locations…
  - … then fit a curve (fit the lowest frequency curve possible)



Input Reconstructed

- Insufficient sampling leads to aliases in the reconstruction
  - The signal is interpreted as a different (lower) frequency
  - Alias – a signal travelling in disguise as another frequency

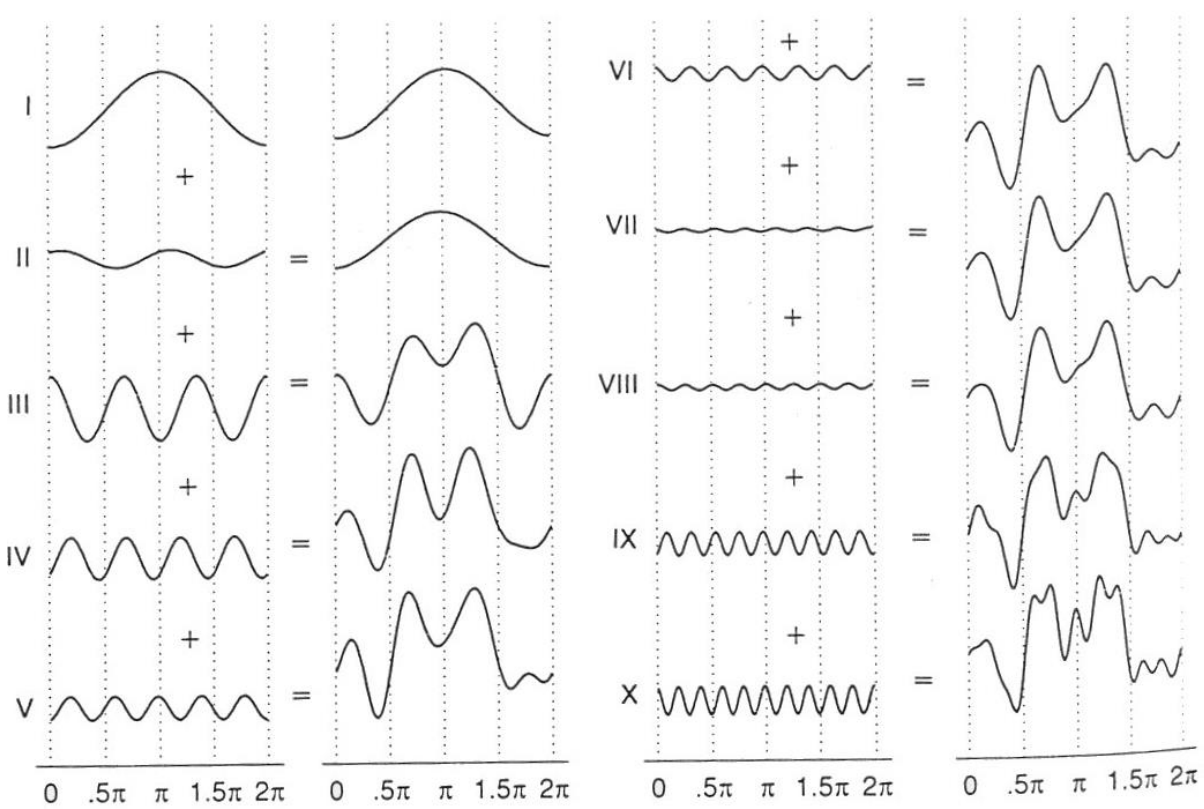


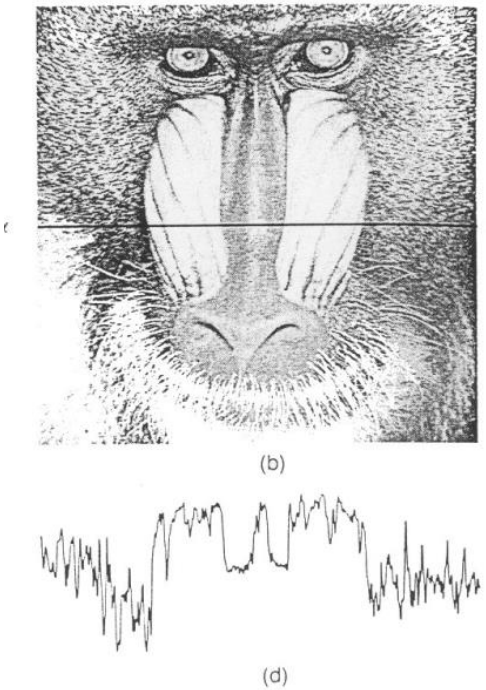
Input Reconstructed

Durand, 06

2.1 Mixtures of frequencies

- A complex signal is a mixture of frequencies
  - Further details – see Fourier analysis, Fourier transform



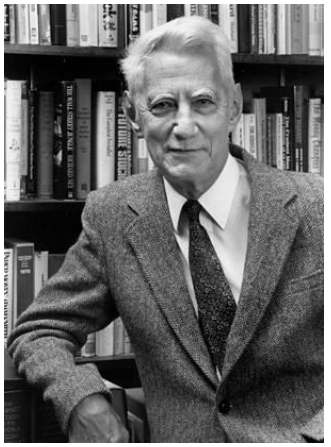


Mandrill. (c) Intensity plot of scan line α in
is courtesy of George Wolberg, Columbia

Foley et al, 90

2.2 The sampling theorem (Shannon, 1949)

- Important in all modern electronic technology involving information.
- Consider a signal that has no frequencies above a certain value B
  - Bandlimited
- If the frequency of samples is f_s, the sampling theorem says that these uniform samples are a complete representation if $B<f_s/2$
  - To avoid aliases, $f_s > 2B$
  - Aliasing will occur if the sampling occurs at less than twice the highest frequency component in the information being sampled
  - 2B is the Nyquist limit (Nyquist, 1928)
- Further details – see Fourier analysis, Fourier transform



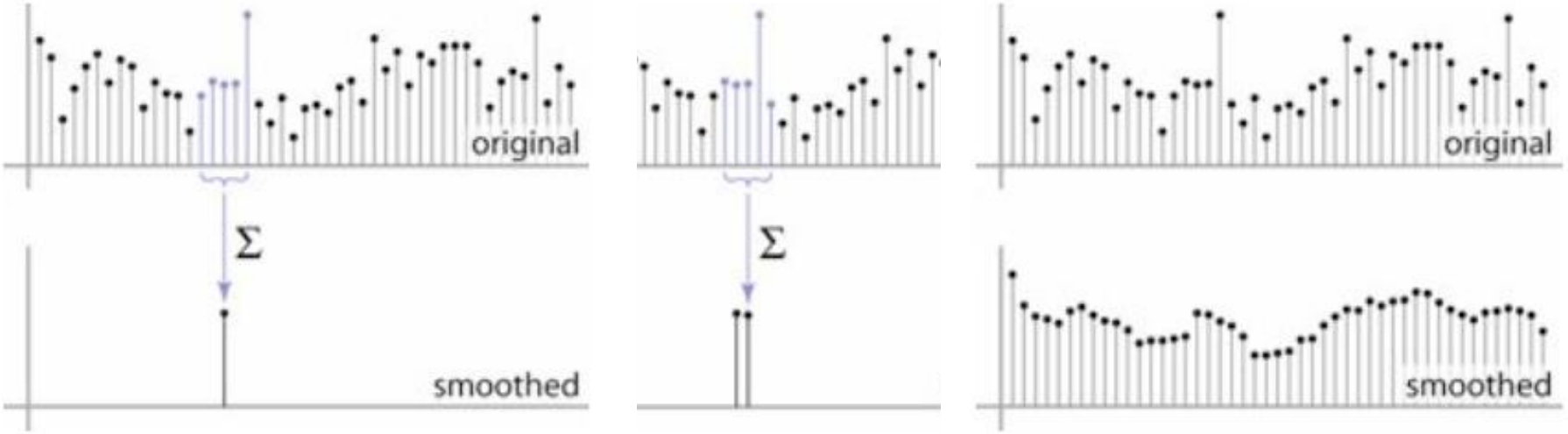
http://www.bell-labs.com/news/2001/february/26/1.html

3. Solutions?

- Sample more often
  - Can't keep doing this forever
  - Just raises Nyquist limit
  - (cf. The rise of mega-pixel resolutions for digital cameras)
- Remove the high frequencies in the image
  - Lose information, but better than aliasing
  - Produces **blurring**
- **Blurring in CG**
  - Rasterisation: compute at higher resolution; then filter / re-sample to produce lower resolution
  - Ray tracing: same problems as rasterisation, but can take advantage of nature of ray tracing to do it differently
  - Textures: blur the texture image before use

4. Blurring

- Basic idea: Averaging using a sliding window



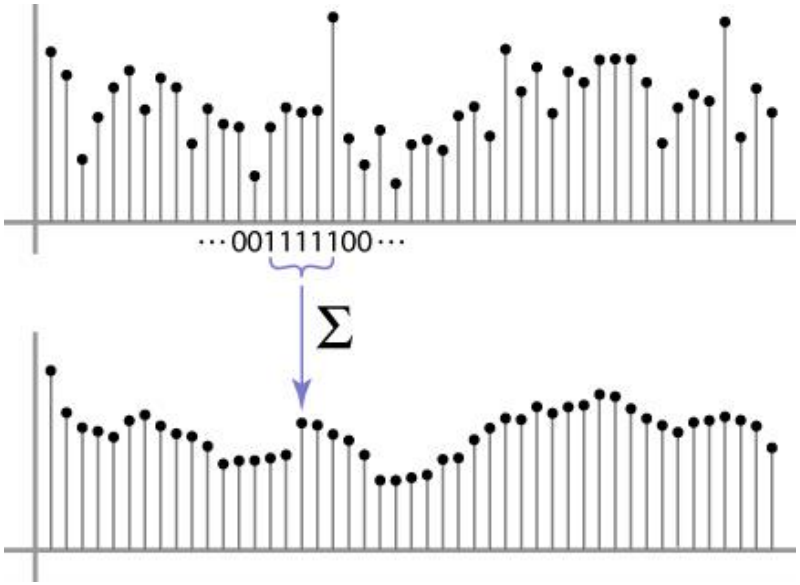
Simple averaging:

$$b_{\text{smooth}}[i] = \frac{1}{2r+1} \sum_{j=i-r}^{i+r} b[j]$$

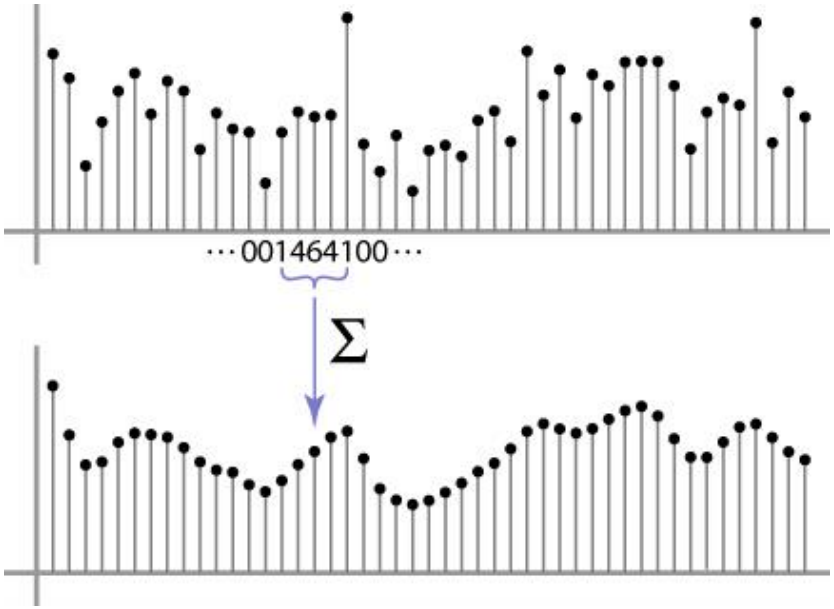
every sample gets the same weight

Steve Marschner, CS4620 Introduction to computer graphics, http://www.cs.cornell.edu/~srm/, 2008

4.1 Moving weighted average



Weights: […, 0,1,1,1,1,1,0,…]/5



Bell curve (gaussian-like) weights […, 1,4,6,4,1, …]/16

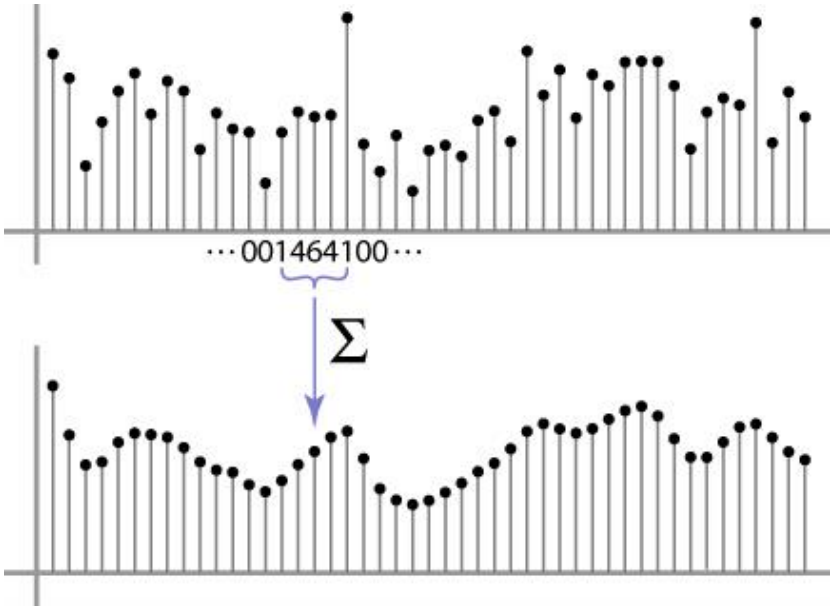
Marschner, 08

4.2 Filters

- Sequence of weights a[j] is called a filter
- Filter is non-zero over its *region of support*
- Filter is normalised so that it sums to 1.0
- Most filters are symmetric

$$(a \star b)[i] = \sum_j a[j]b[i-j]$$

each sample gets its own weight (normally zero far away)

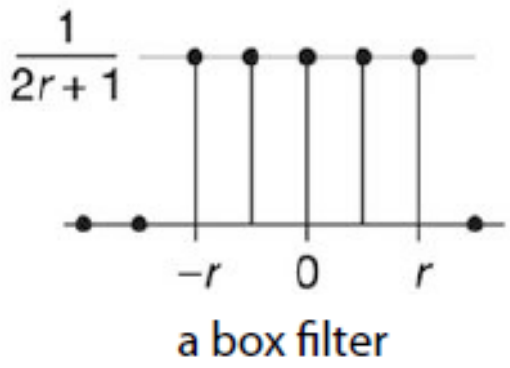


Bell curve (gaussian-like)
weights […, 1,4,6,4,1, …]/16

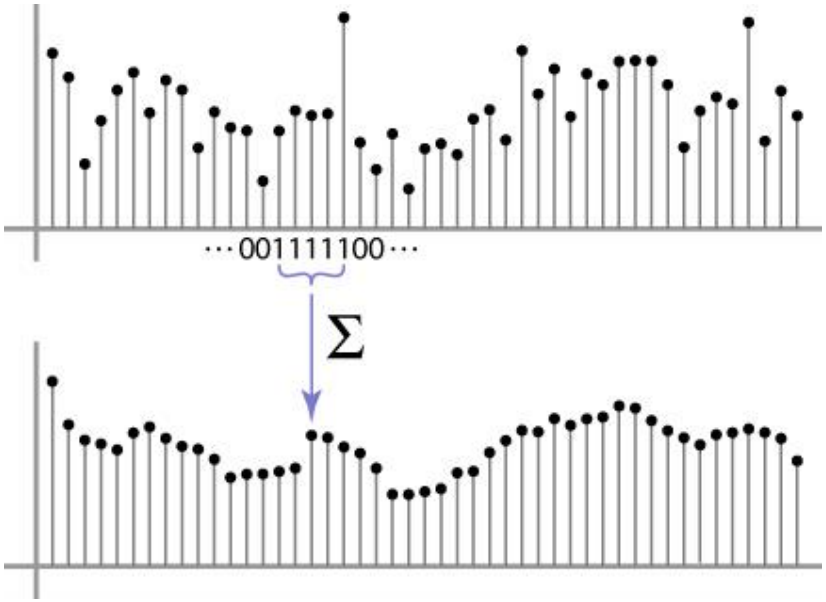
Marschner, 08

4.3 Convolution and filtering

- Moving average is convolution with a box filter



a box filter



Marschner, 08

4.4 2D

- A Gaussian kernel gives less weight to pixels further from the centre of the window

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

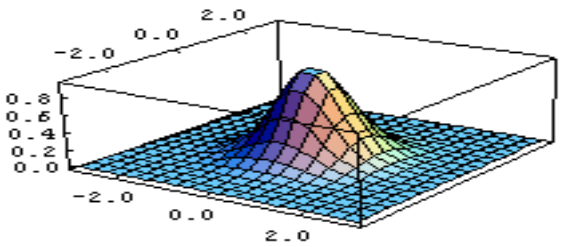
$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Marschner, 08

4.4 2D

- This Gaussian kernel is an approximation of a Gaussian function:

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



$$h(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$$

- In equation form:

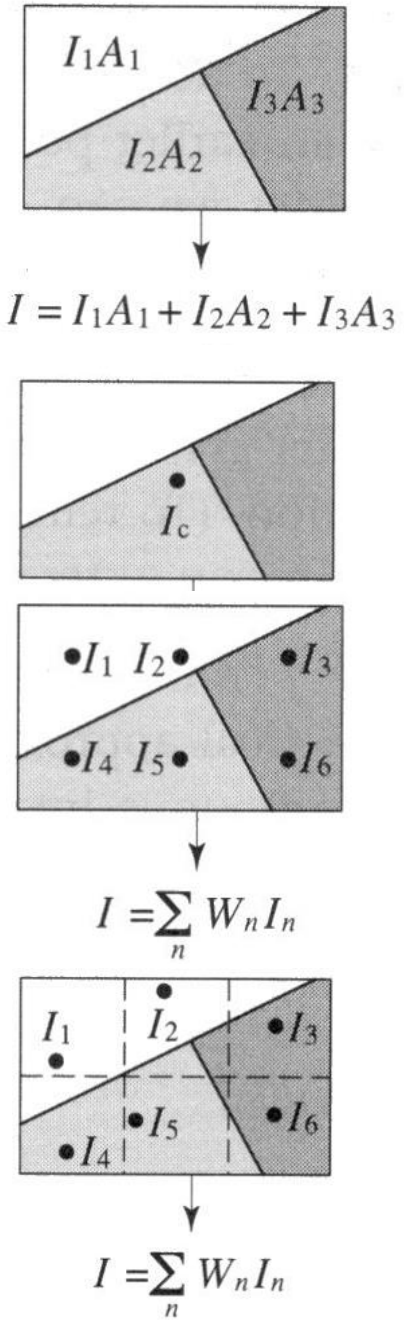
$$(a \star b)[i,j] = \sum_{i',j'} a[i',j']b[i-i',j-j']$$

Marschner, 08

5. Anti-aliasing and computer graphics

1. Pre-filtering – 'infinite' samples per pixel
   - Calculate the precise contribution of fragments of projected object structure as it appears in a pixel (Carpenter, 84; Catmull, 86)
2. No filtering – one sample per pixel
   - Do nothing – real-time animation, or as a preview method.
3. Post-filtering – n uniform samples per pixel
   - Commonest approach – render a virtual image at n times resolution of final screen image, then filter this
4. Post-filtering – stochastic sampling
   - Similar to (3), except pixel samples are 'jittered' (see ray tracing)

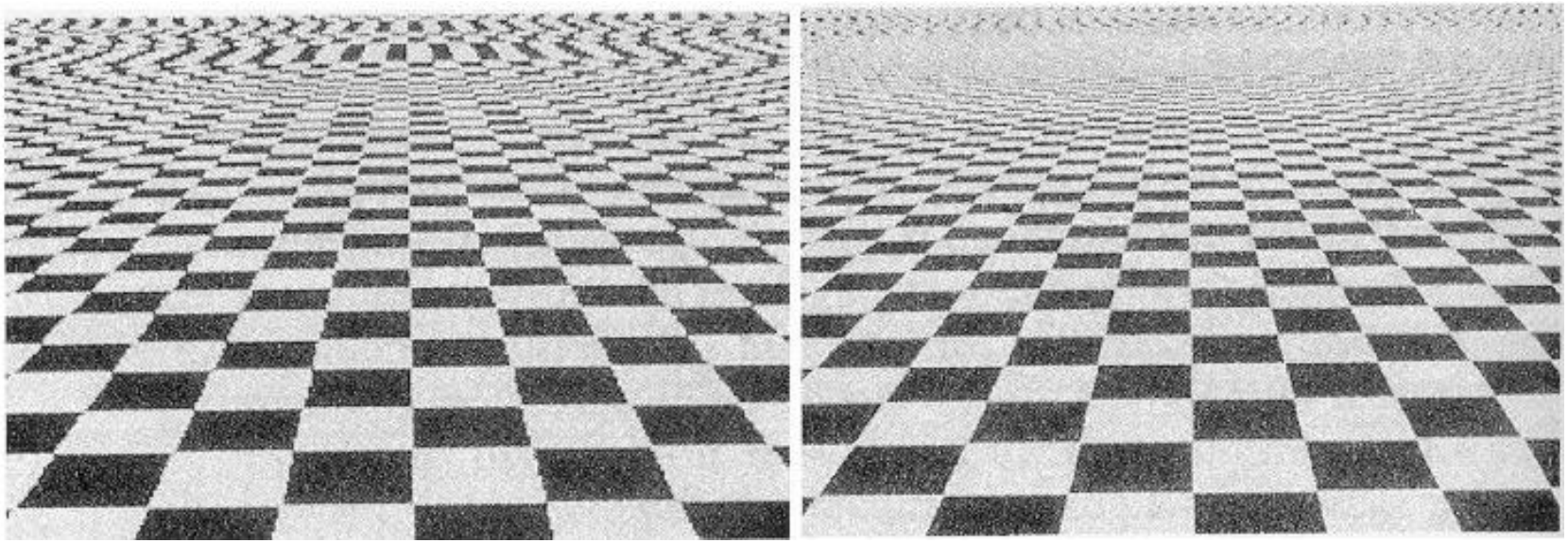
Consider a single pixel



Watt,00

6. Post-filtering – n uniform samples per pixel: Supersampling

- Increase frequency of sampling grid (i.e. increase spatial resolution of pixel array) and average results down to final screen resolution



Watt,00

6. Post-filtering – n uniform samples per pixel: Supersampling

- Sample at n times resolution → virtual image
- Re-sample at display resolution using a filter
- *Digital convolution*: window is centred on a supersample and a weighted sum of products is obtained by multiplying each supersample by the corresponding weight in the filter

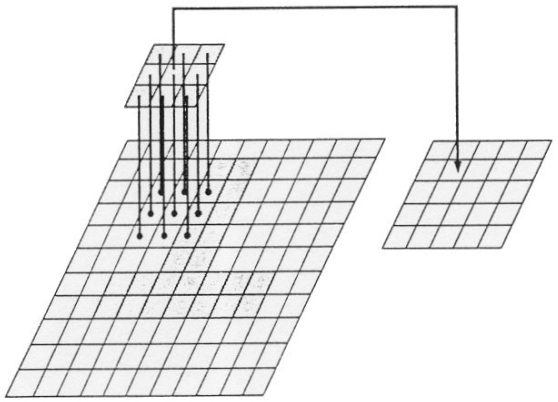


Table 4.1 Bartlett windows used in postfiltering a supersampled image.

3×3			5×5					7×7						
1	2	1	1	2	3	2	1	1	2	3	4	3	2	1
2	4	2	2	4	6	4	2	2	4	6	8	6	4	2
1	2	1	3	6	9	6	3	3	6	9	12	9	6	3
			2	4	6	4	2	4	8	12	16	12	8	4
			1	2	3	2	1	3	6	9	12	9	6	3
								2	4	6	8	6	4	2
								1	2	3	4	3	2	1

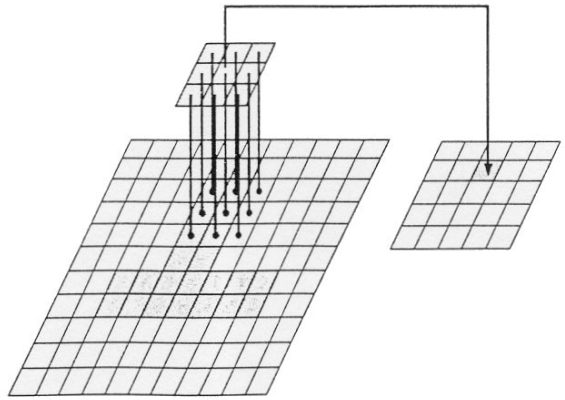


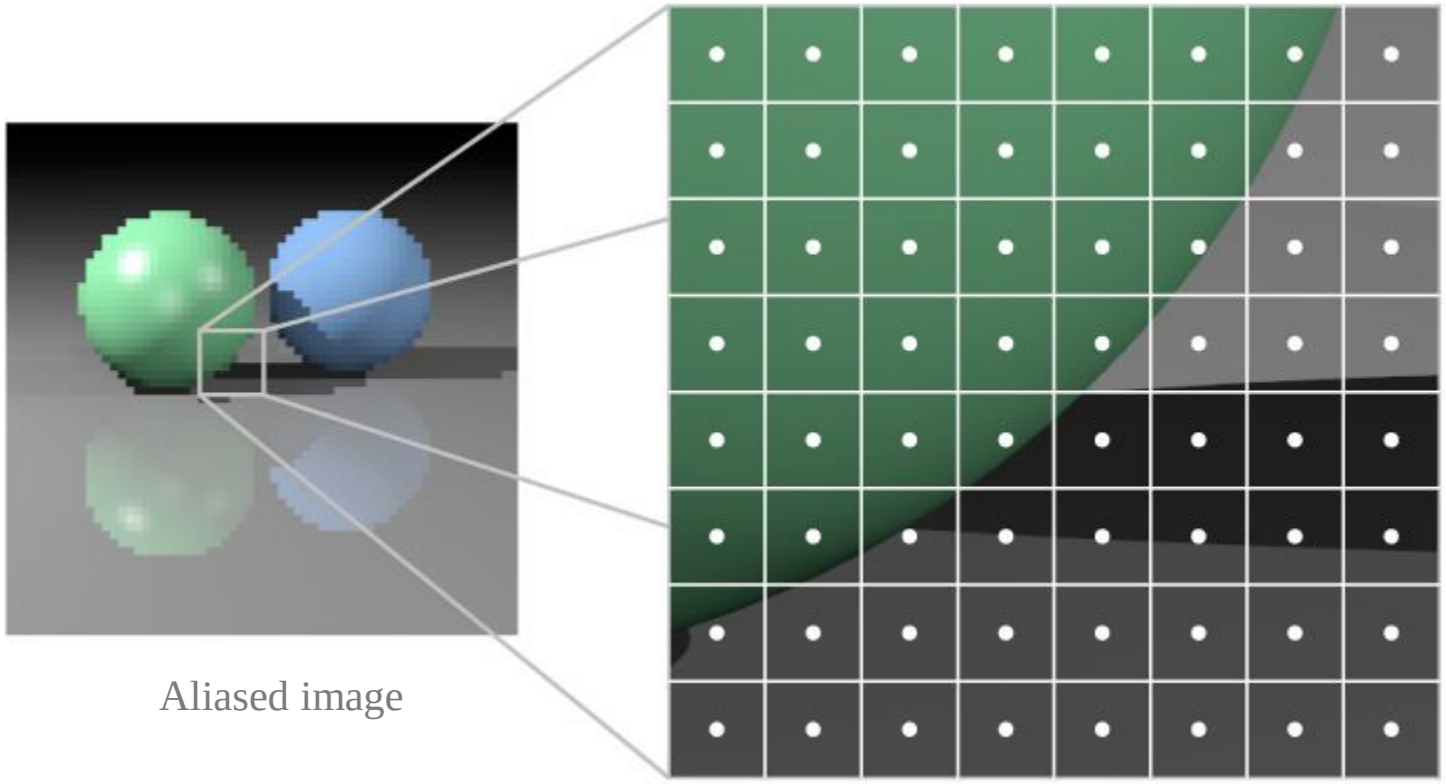
Fig. 14.32 Digital filtering. Filter is used to combine samples to create a new

↑ Resolution:

2x 3x 4x

6. Post-filtering – n uniform samples per pixel: Supersampling

- Ray tracing example:



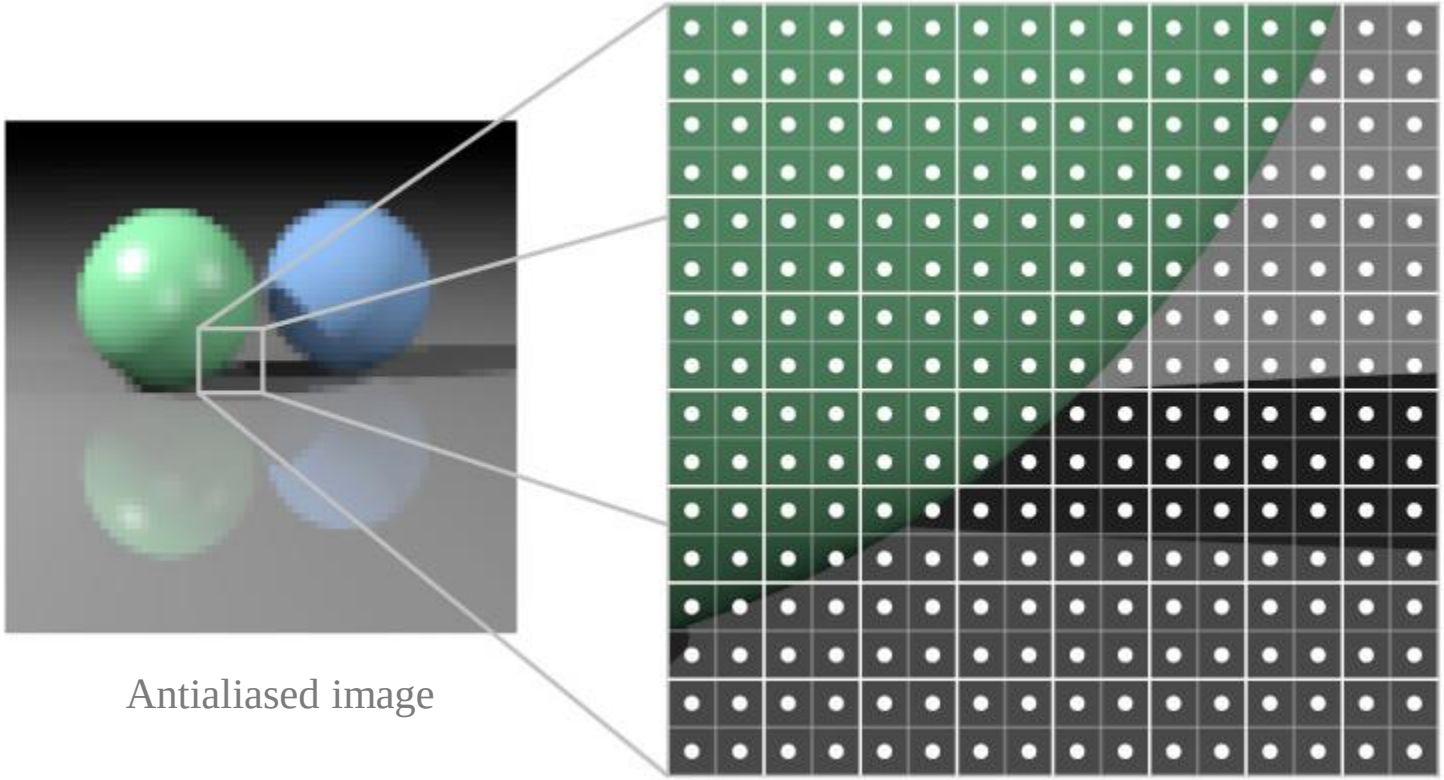
Aliased image

One sample per pixel

Marschner, 06

6. Post-filtering – n uniform samples per pixel: Supersampling

- Ray tracing example:



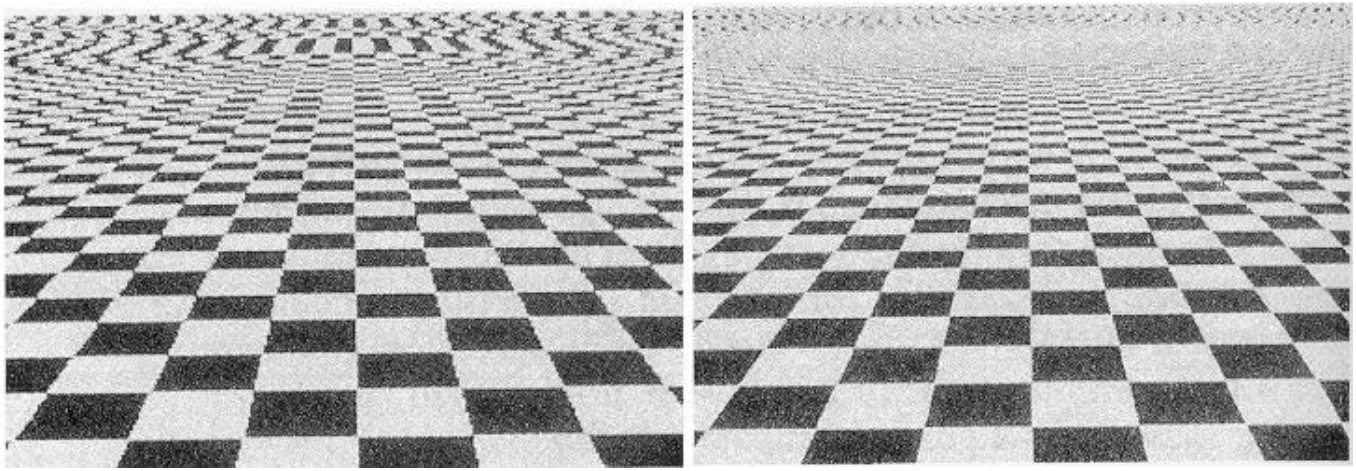
Antialiased image

Four samples per pixel

Marschner, 06

6. Post-filtering – n uniform samples per pixel: Supersampling

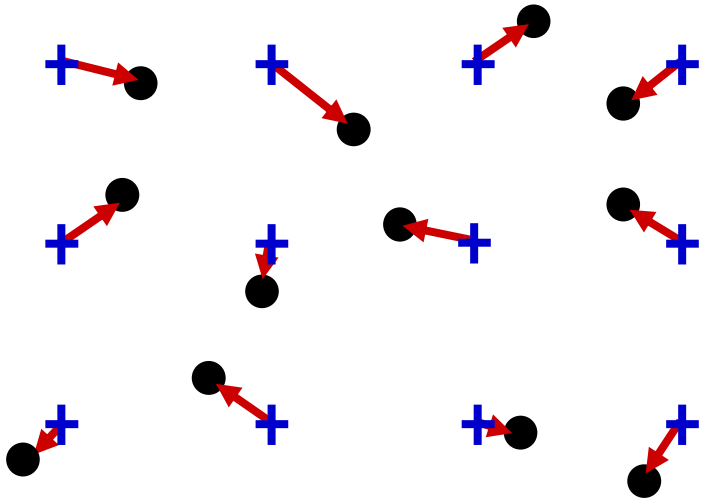
- Easily incorporated into (i) rasterization+z-buffer, and (ii) ray tracing
- Drawbacks:
  - 'Global' method – computation is not context dependent
  - Z-buffer: Memory requirements (although can be done as a multipass technique requiring only two buffers)
  - Only increases Nyquist limit
  - Trade off aliasing for blurring



Watt,00

7. Post-filtering – non-uniform sampling

- Incorporated into ray tracing, not standard rendering pipeline
- Based on idea that eye contains array of non-uniformly distributed photoreceptors
- **Jittering**: Uniform sample + random perturbation
- Sampling is now non-uniform
- In practice, adds noise to image – trades noise for aliasing



Durand, 06

Jittered supersampling

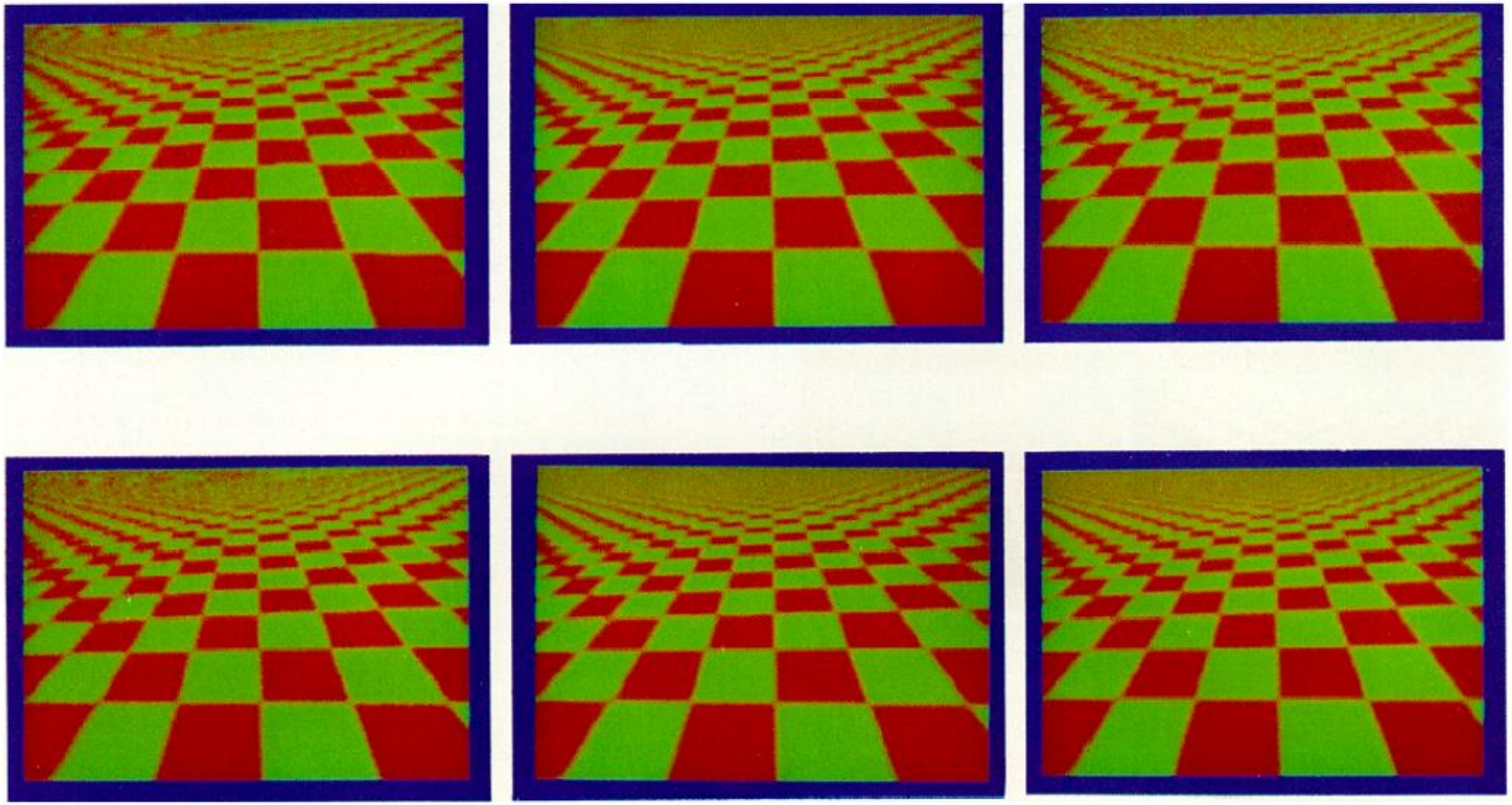
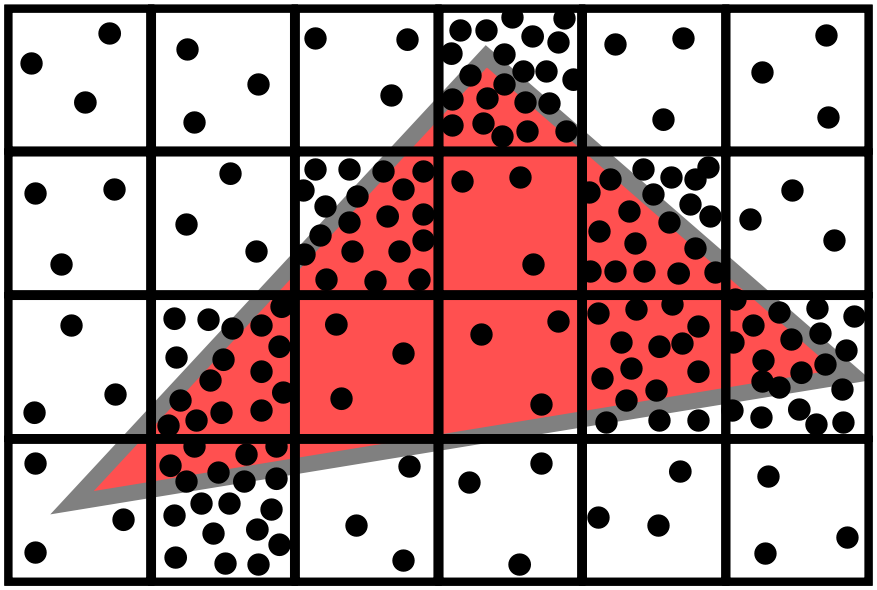


Figure 11
Jittered sampling of a slowly moving texture with jitter of 0, .5, and 1 from left to right and oversampling rates of 1 and 2 from top to bottom.

7.1 Post filtering – adaptive supersampling

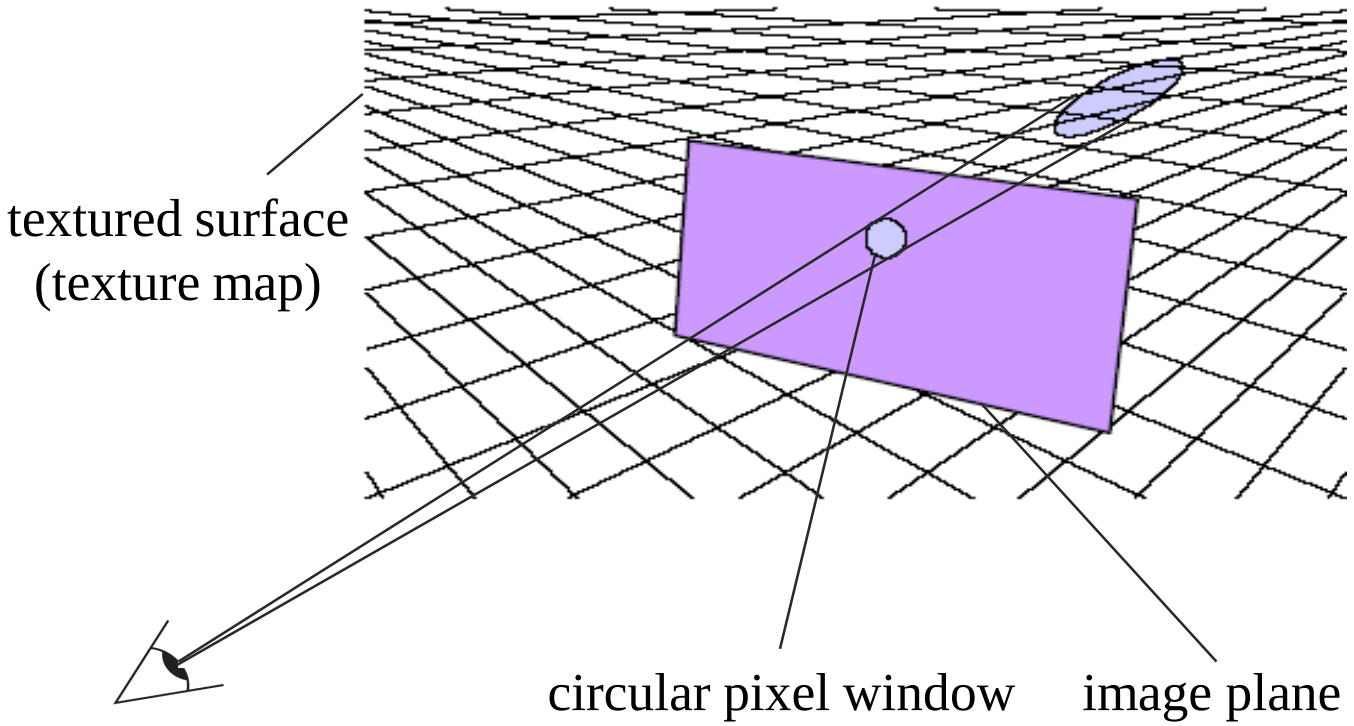
- For ray tracing: Use more sub-pixel samples around edges



Durand, 06

8. Anti-aliasing and texture mapping

- Necessary because small detail is often coherent, and there are compression issues
- How to map the texture area seen through the pixel window to a single pixel value?

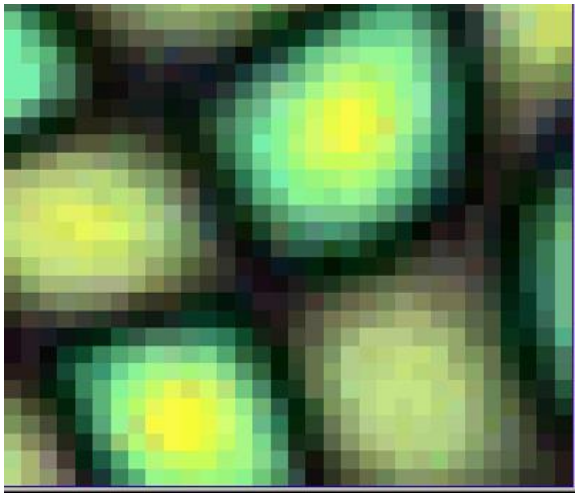


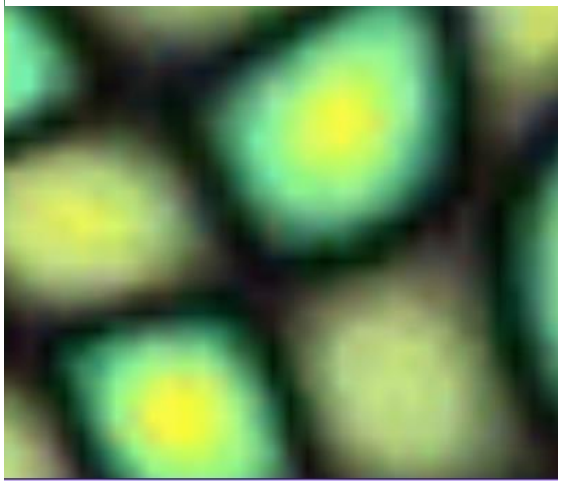
Durand, 06

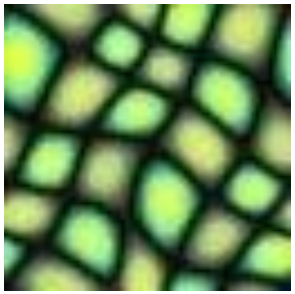
8. Anti-aliasing and texture mapping

Use a filter.
Magnification better, but blurry

Magnification for display

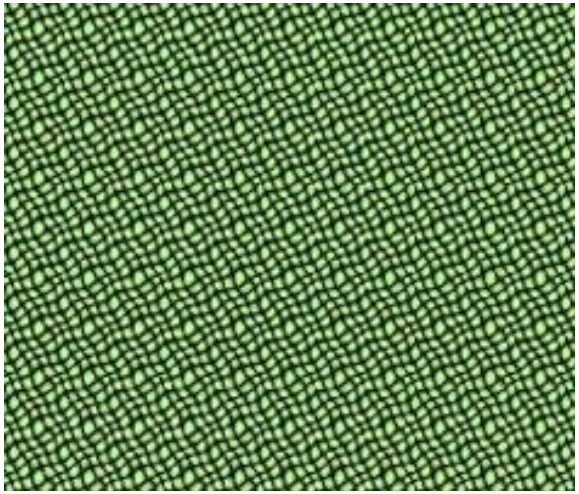


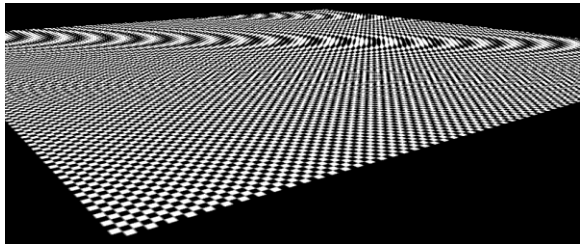




64x64 pixels

Original Texture





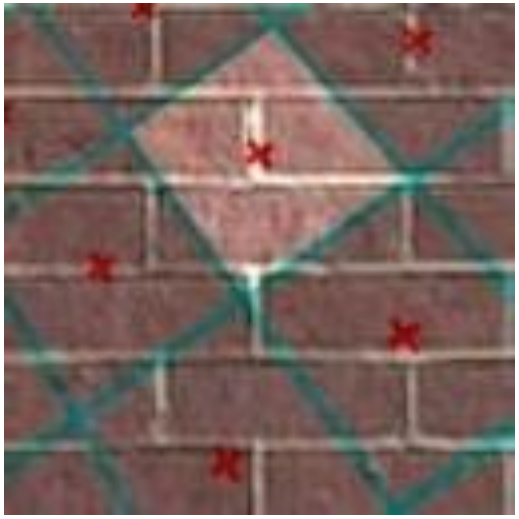
Durand, 06

Minification for display

8. Anti-aliasing and texture mapping

- Solution is to blur (pre-filter) the texture – remove the high frequencies in the texture
- Essentially, we want to convolve the texture with a filter kernel centered at the sample (i.e. pixel center)
- Expensive, but can be approximated

projected texture in image plane

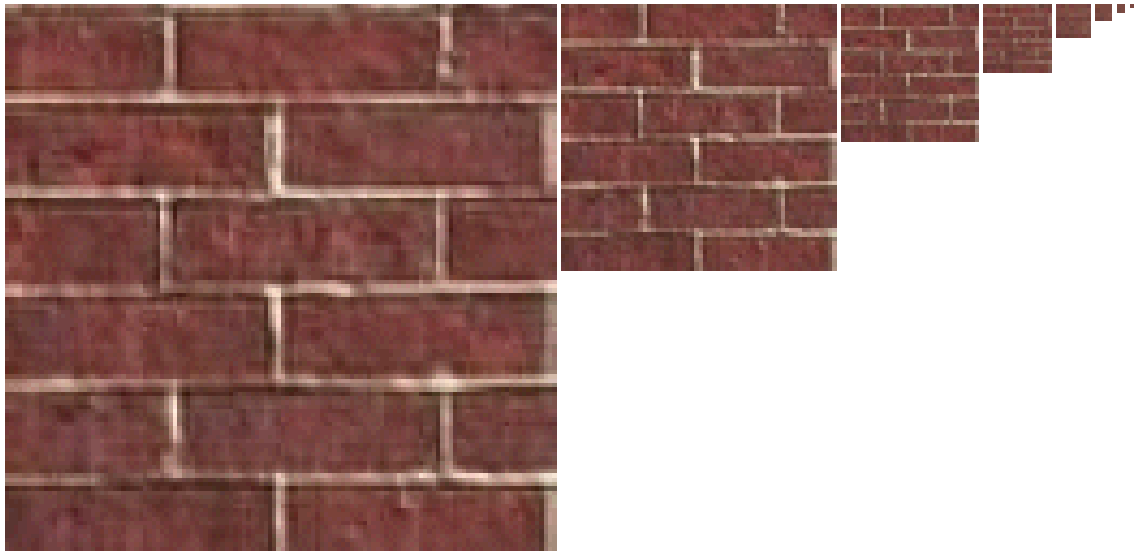


box filter in texture plane

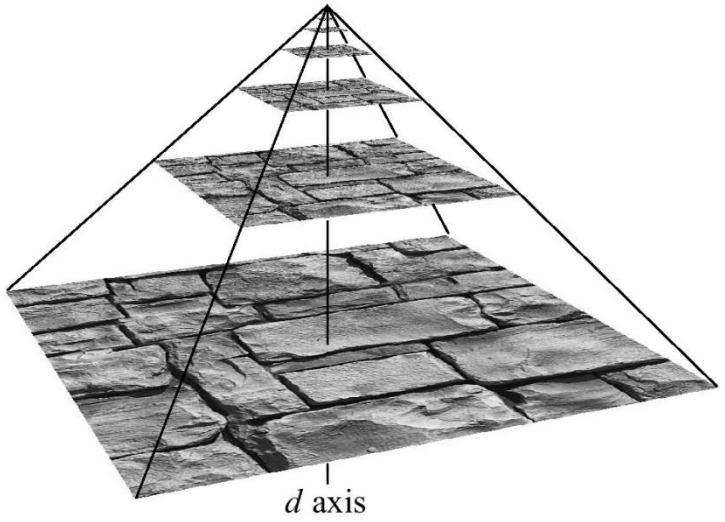
Durand, 06

8.1 Mip-mapping (Multum In Parvo: much in little space) (Williams, 83)

- Instead of a single texture map, many images are used, all derived by averaging down the original image
  - Image pyramid
- For example, a 64x64 image would be represented as the 64x64 image, a 32x32 image, a 16x16 image, and so on down to a 1x1 image.



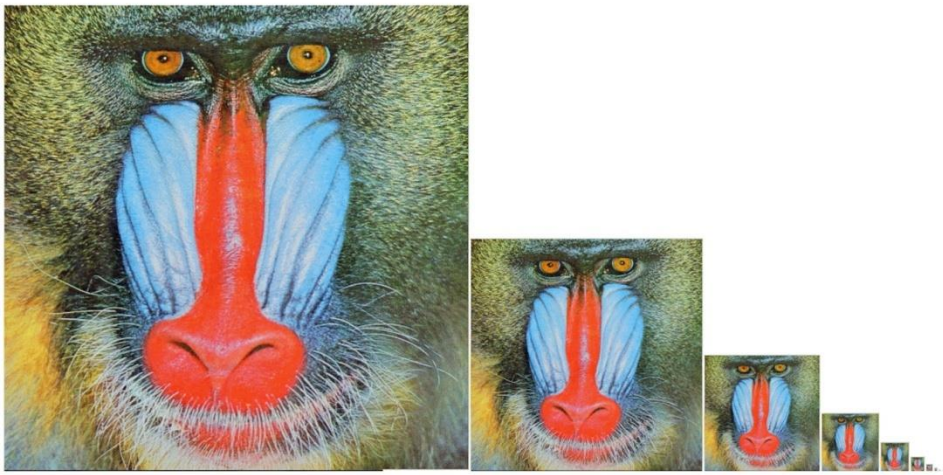
Durand, 06



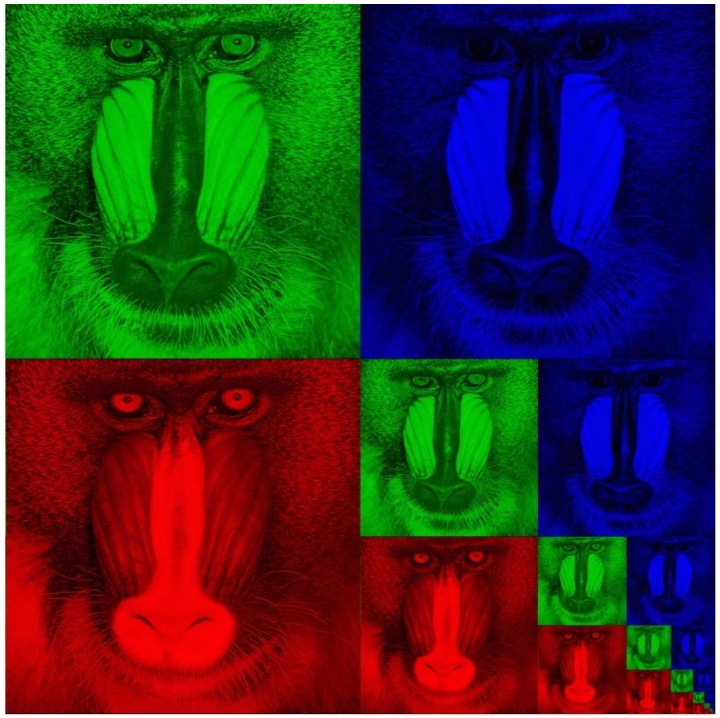
Marschner, 08

8.1 Mip-mapping

- The original texture is pre-filtered (blurred) to remove high frequencies



10-level mip map

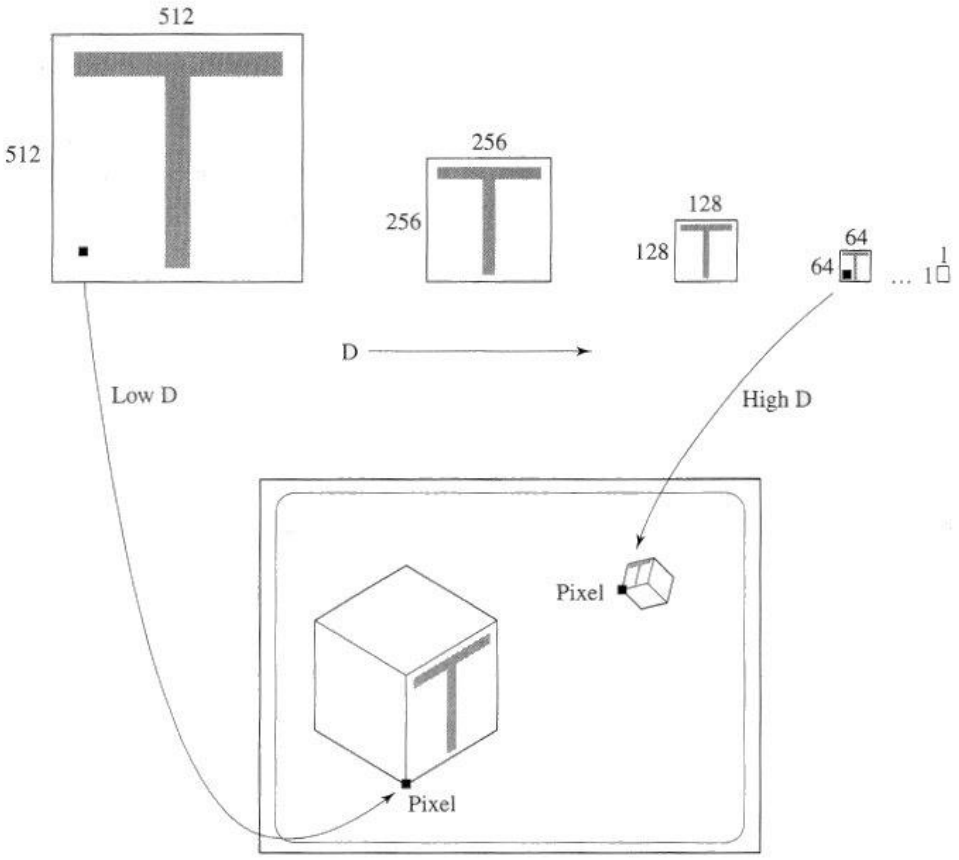


Memory format of a mip map

Durand, 06

8.1 Mip-mapping

- Based on distance a particular mip-map is used
- For close-up views the high-resolution texture map is used
- For distant views, a low resolution map is used

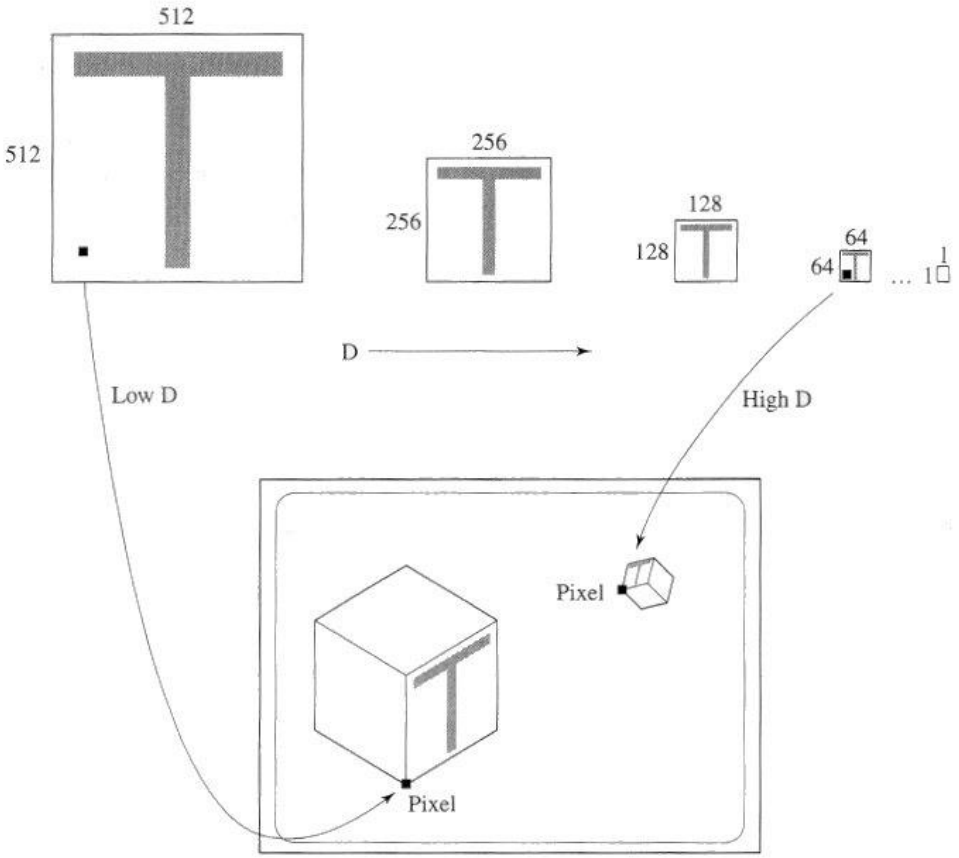


Watt,00

8.1 Mip-mapping

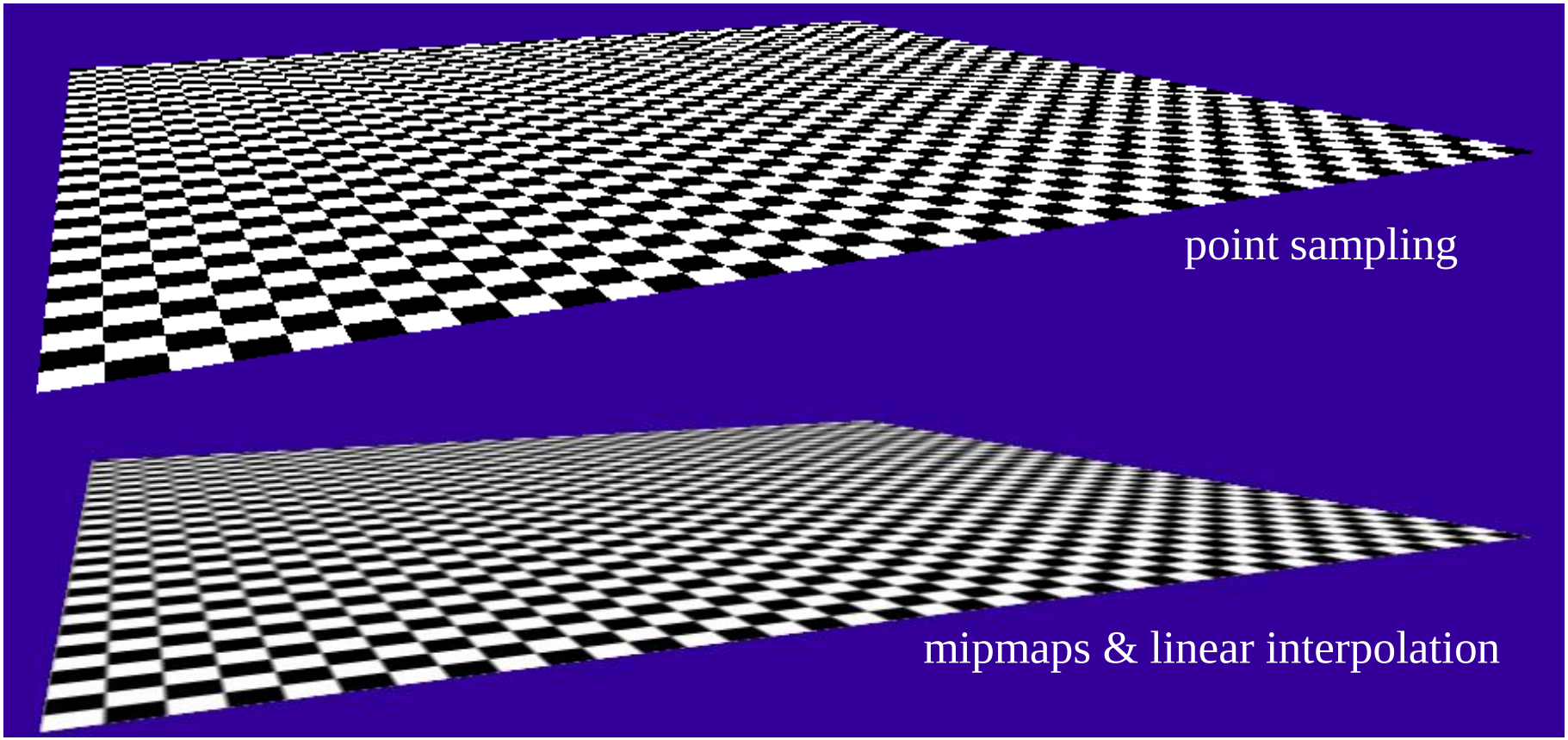
- To integrate into an interpolation process, the rate of (u,v) change over a polygon with respect to x and y is calculated and two levels in the mip-map set are chosen, with interpolation between mip-maps used for the final texture value

$$D = \text{maxOf}\left(\left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2\right)^{\frac{1}{2}}, \left(\left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right)^{\frac{1}{2}}\right)$$

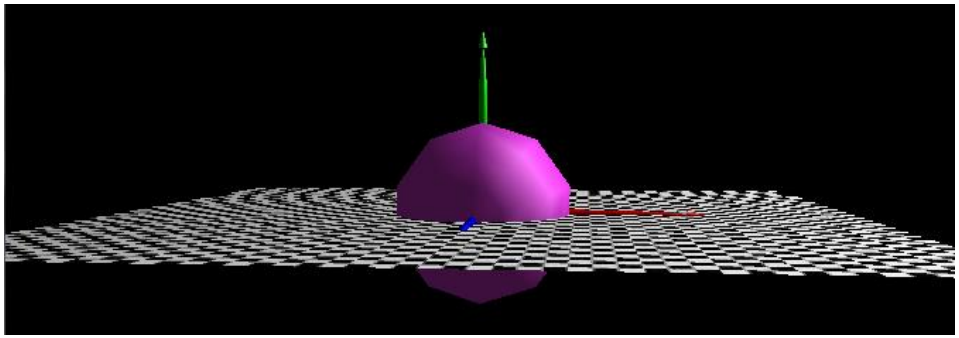


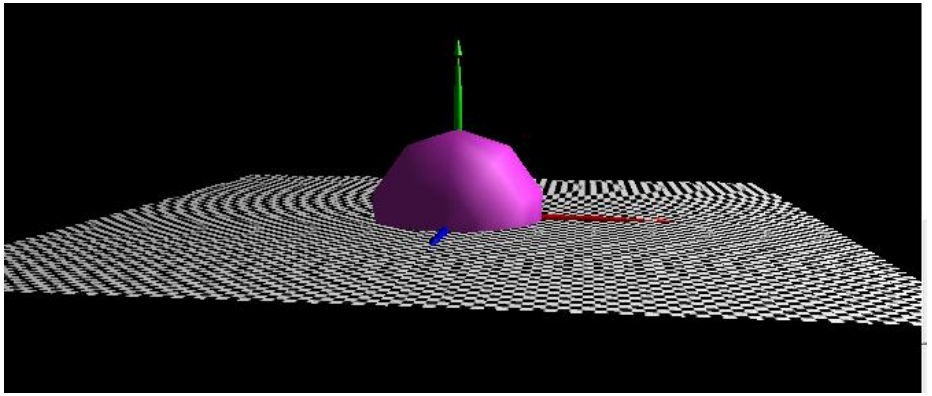
Watt,00

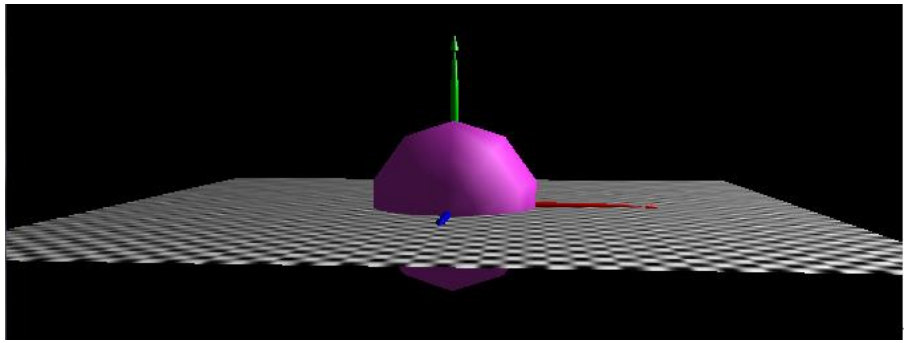
8.2 Mip-mapping example

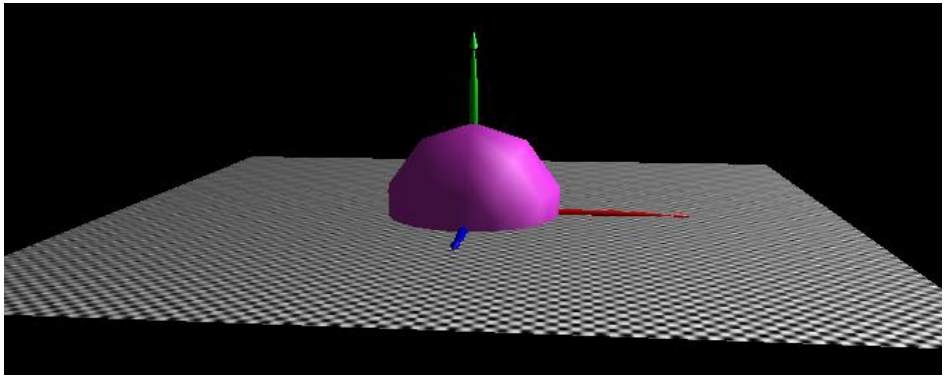


Durand, 06

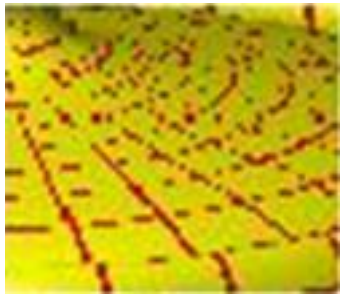




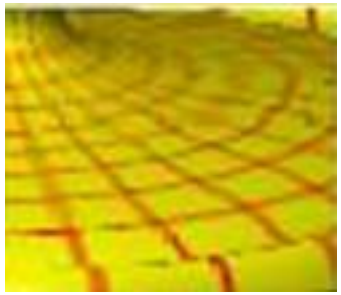




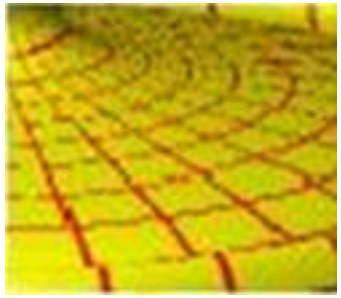
demo



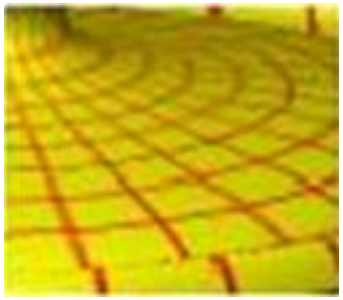
Not antialiased
Rendering time: 1



With mipmapping
Rendering time: ~2.3



Supersampled at a rate of nine samples to one pixel
Rendering time: ~3.5



9:1 supersampling and mipmapping
Rendering time: ~9

The SIGGRAPH 97 Education Slide set

9. Summary

- Insufficient sampling of a signal leads to aliases
- In computer graphics aliases manifest themselves in missing small objects and textures breaking up in perspective views
- There are four general approaches to anti-aliasing in computer graphics:
  1. Pre-filtering – 'infinite' samples per pixel
  2. No filtering – one sample per pixel
  3. Post-filtering – n uniform samples per pixel. Most common form used with rasterization and z-buffer for polygon mesh rendering
  4. Post-filtering – stochastic sampling. Easy to add in to ray tracing
- For 2D textures, most common anti-aliasing approach is *mip-mapping,* which is a pre-filtering approach