Sampling

Aliasing

- Discretization of signals
  - *Space*
  - *Time*
  - *Intensity*
- High Frequency information appears as low frequency
- Jaggies

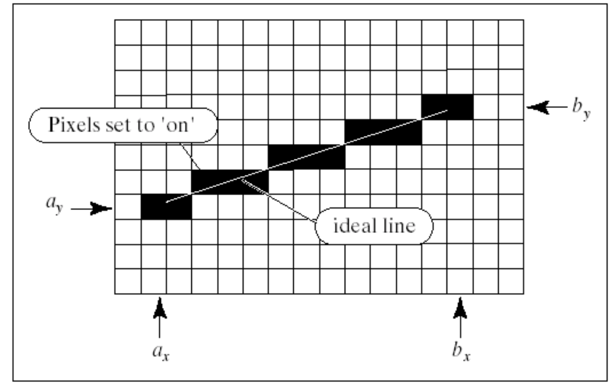


FIGURE 10.23 Drawing a straight-line-segment.

Sampling

Sampling Theorem (Nyquist or Shannon)

To avoid aliasing the sampling frequency must be greater than two times the highest frequency in the signal.

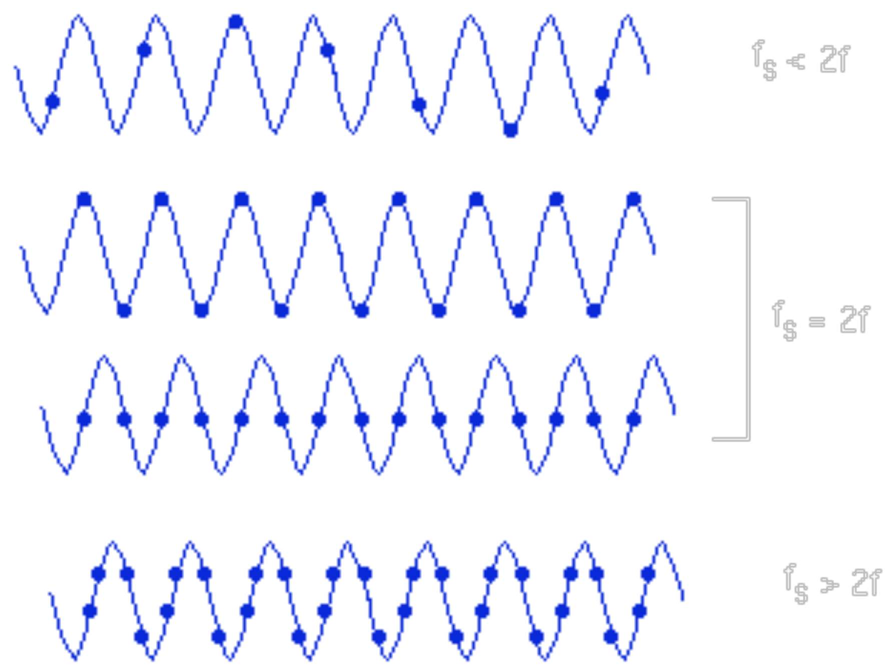
Sampling exactly equal to Highest frequency x 2 can produce both good and bad results

Sampling rate strictly greater than highest frequency x 2

Bad

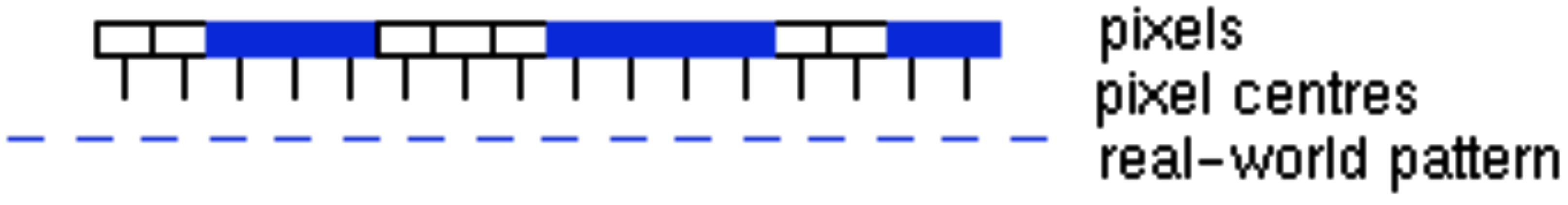
Good

Bad

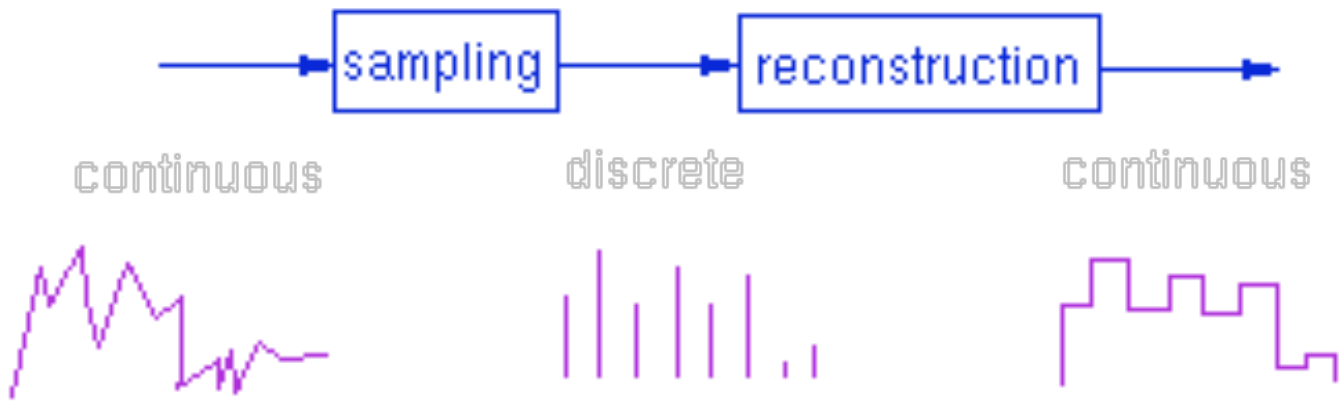


Sampling in graphics

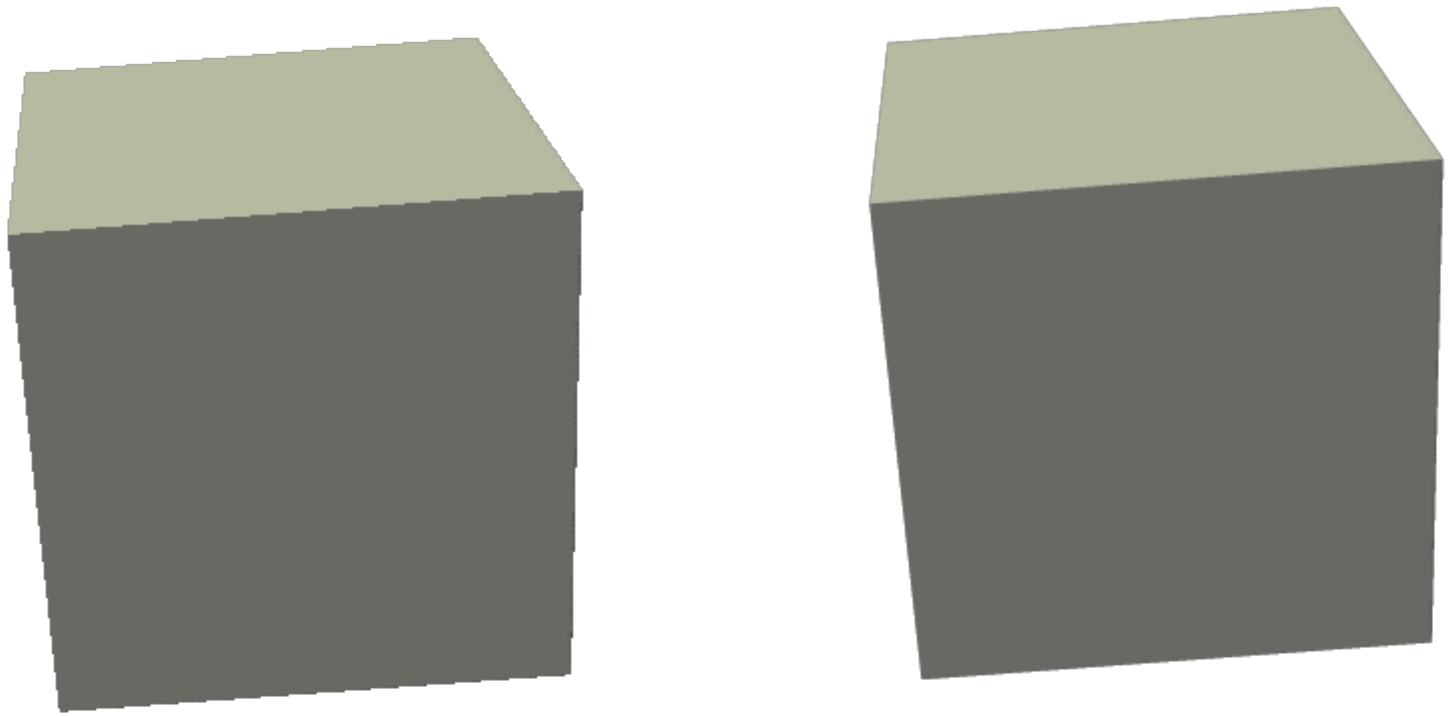
Example: Sampling at pixel centers



Sampling and reconstruction



Aliasing in Computer Graphics



Reducing aliasing

Only one way: Blur the image

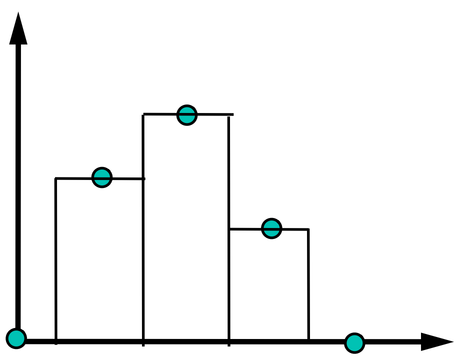
Strategies

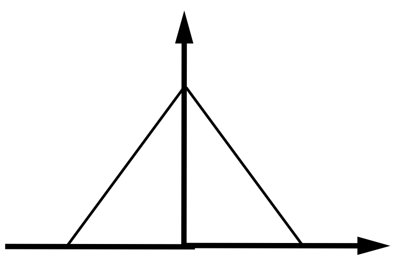
- Prefiltering (Before sampling)
  - *Pixel Coverages*
  - *Computationally expensive especially for non-polygonal objects*
- Postfiltering (After sampling)
  - *Weighted average of samples*

Filtering

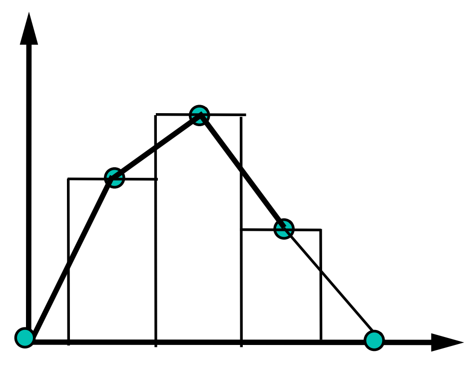
Filtering of step function using convolution with kernel g

$$P(x) = \int_{-s}^{s} f(x+u)g(u)du$$





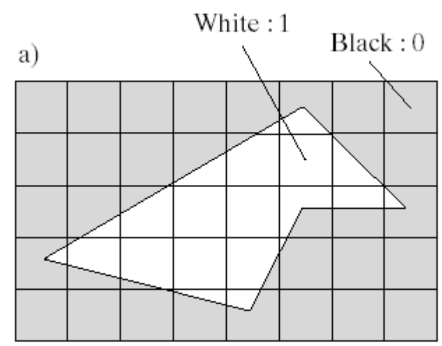
Tent kernel



Pre-filtering

Unweighted Area sampling

- Use average intensity of square pixel area



b)

0	0	0	0	1	6	0	0
0	0	0	6	13	15	8	0
0	3	11	15	15	9	7	3
3	11	14	15	12	2	0	0
0	0	1	6	5	0	0	0

FIGURE 10.49 Using the fraction of the pixel area covered by the object.

Black 0, White 1 (15)

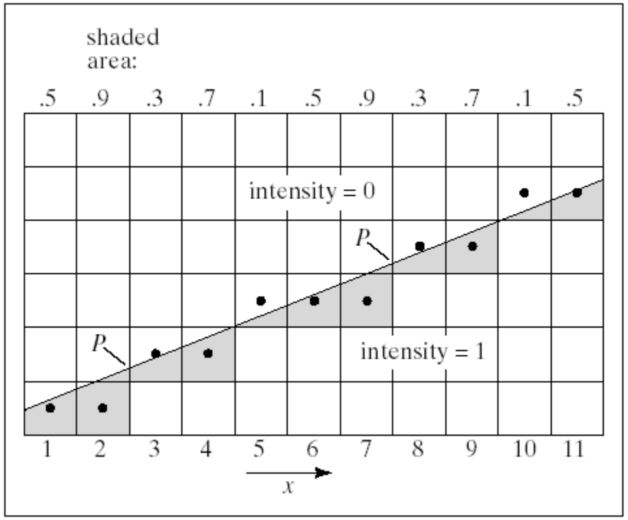
Pixel value: coverage*15

Incremental Polygon Antialiasing

Bresenham's Algorithm provides the dotted pixels (boundary)

- Incremental area calculation

FIGURE 10.50 Example of scan conversion with antialiasing.



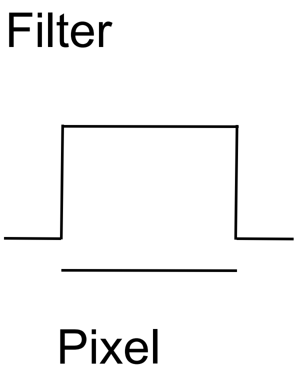
Inside pixels 1

Outside pixels 0

Boundary pixels fractions based on coverage

Box Filter

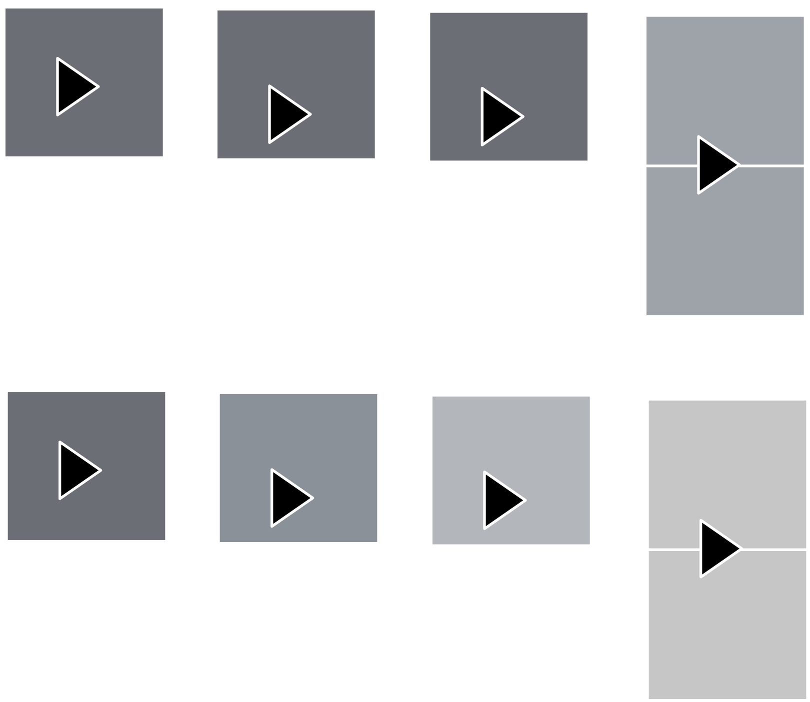
Area coverage approach corresponds to a box filter

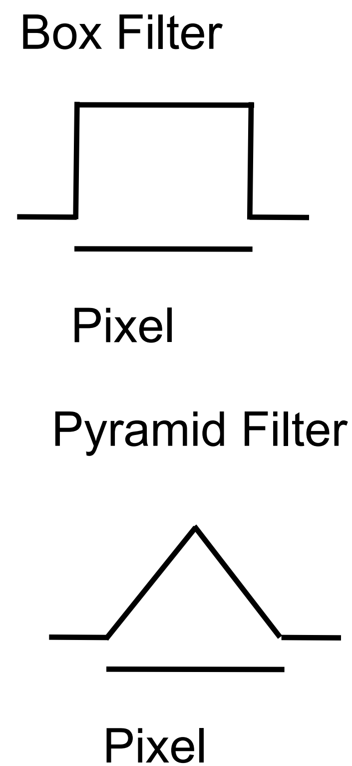


$$P(x) = \int_{-s}^{s} f(x+u)g(u)du$$

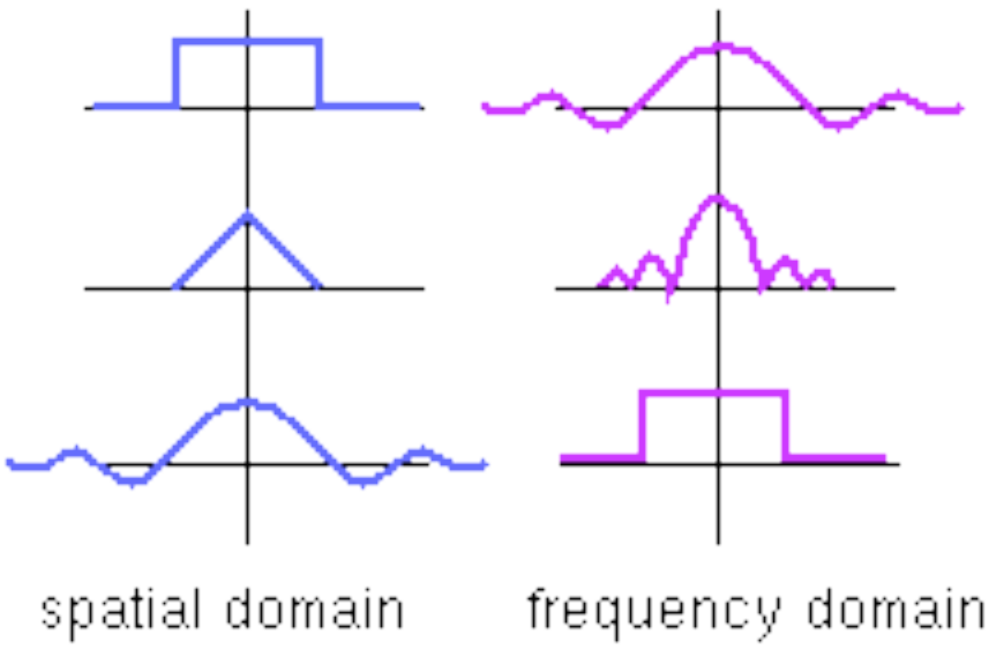
Problem with Box filter

Area Coverage Independent of position





Filters Kernels



Postfiltering

Super sampling

- Take many samples
- Combine them

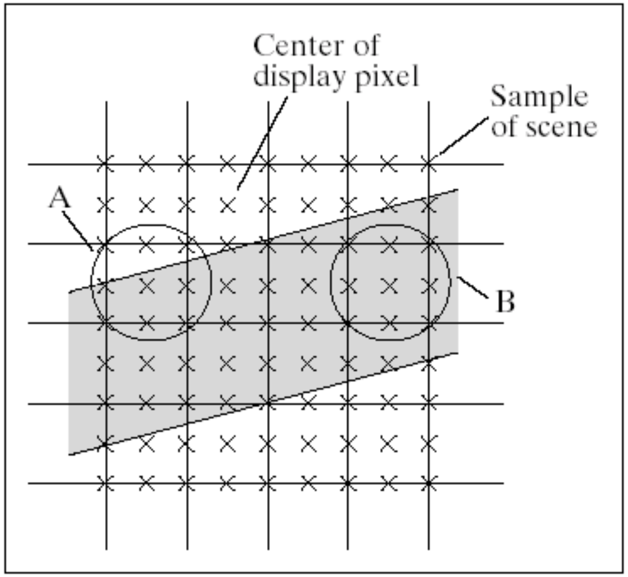


FIGURE 10.51 Antialiasing using supersampling.

Box Filter (Discreet version)

1/4	1/4
1/4	1/4

Bartlett window

FIGURE 10.55 Examples of window functions.

a) $$\frac{1}{8}\begin{bmatrix}0 & 1 & 0\\ 1 & 4 & 1\\ 0 & 1 & 0\end{bmatrix}$$

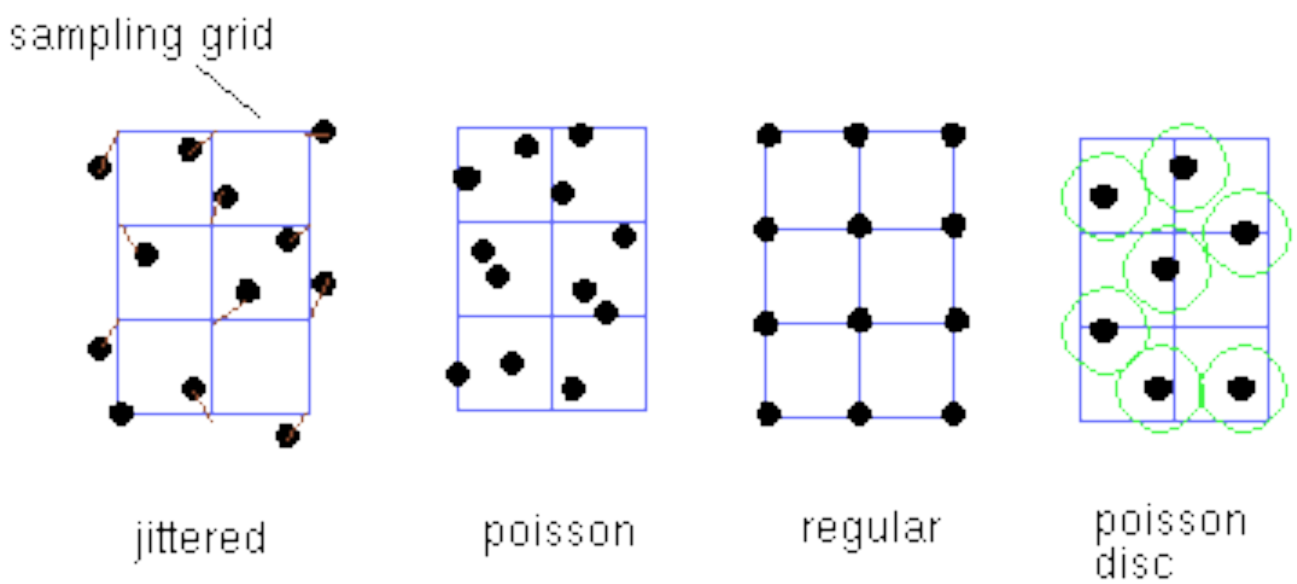
b) $$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

c) $$\frac{1}{81}\begin{bmatrix}1 & 2 & 3 & 2 & 1\\ 2 & 4 & 6 & 4 & 2\\ 3 & 6 & 9 & 6 & 3\\ 2 & 4 & 6 & 4 & 2\\ 1 & 2 & 3 & 2 & 1\end{bmatrix}$$

All weights add up to 1

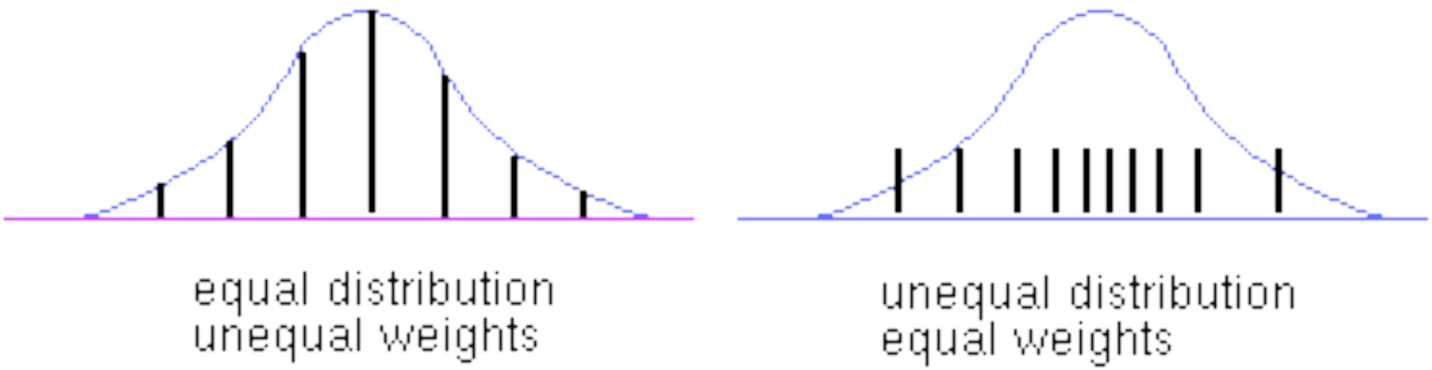
Stochastic supersampling

High frequency noise



Importance Sampling

Location vs density



Height indicates weight

Scene antialiasing

```
glClear(GL_ACCUM_BUFFER_BIT)
for(int i = 0 ; i < 8 ; i++ )
{
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
    cam.slide(jitter[i].x,jitter[i].y,0) ;  // move camera less than
                                             // a pixel in x and y
    display() ;
    glAccum(GL_ACCUM,1/8.0) ;
}
glAccum(GL_RETURN, 1.0) ;
// jitter is chosen from a particular distribution
```

Texture Antialiasing

Pixels have area

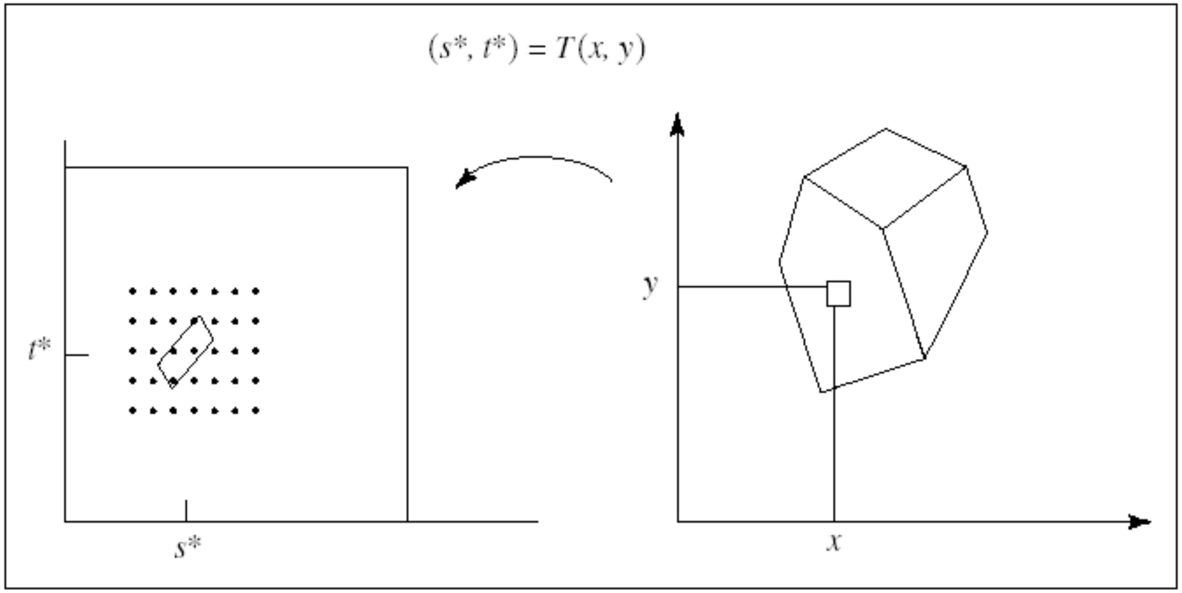
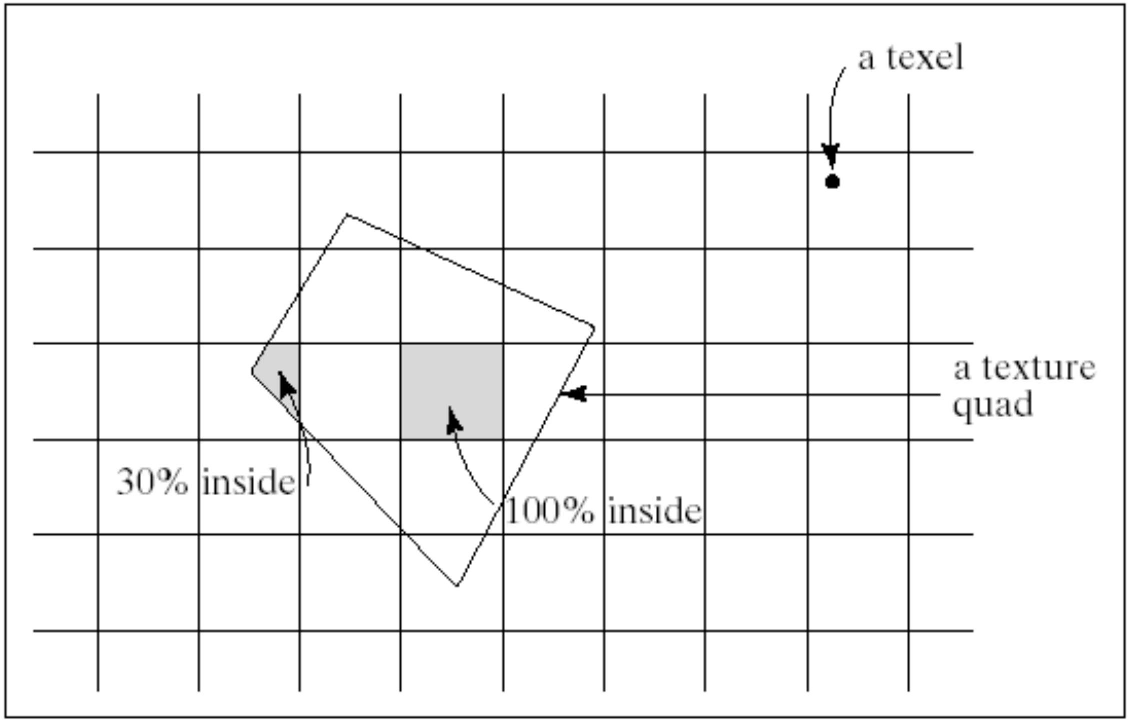


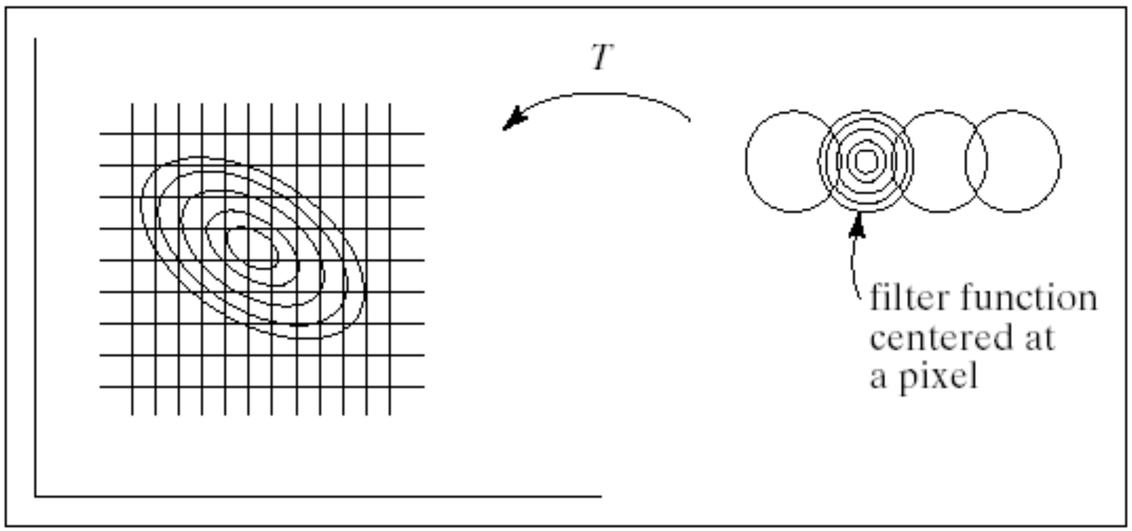
FIGURE 10.57 Cause of aliasing in rendering texture.

Area coverage

Two costly



Elliptical weighted average



Stochastic sampling

Average = $1/N_k$ texture($s + a_k$, $t+b_k$)

Where a_k, b_k are small random quantities and N_k the number of samples.

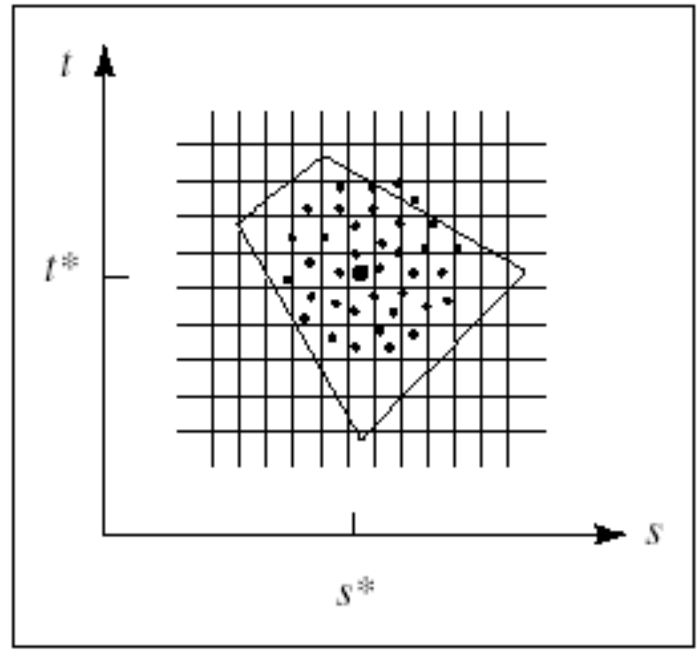


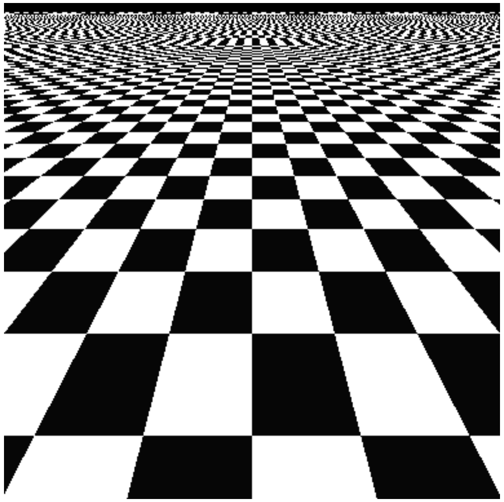
FIGURE 10.60 Antialiasing using stochastic sampling.

Examples

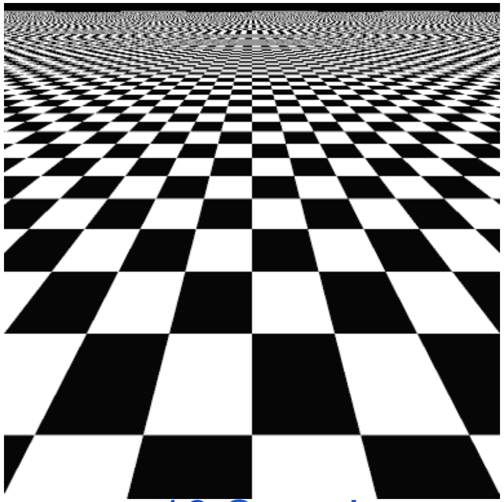
From: www.hpl.hp.com/research/mmsl/projects/graphics/antialiasing/index.html

Box vs Tent filtering over regular grids

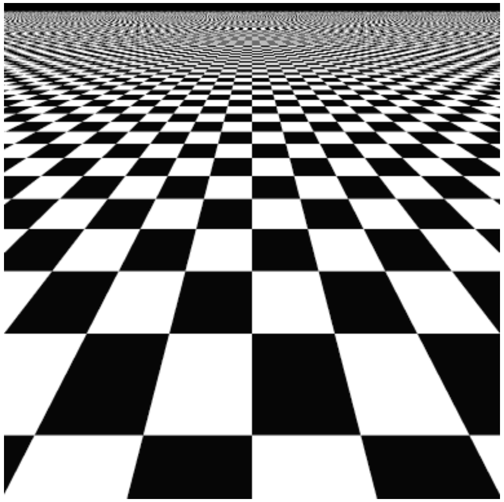
1 Sample



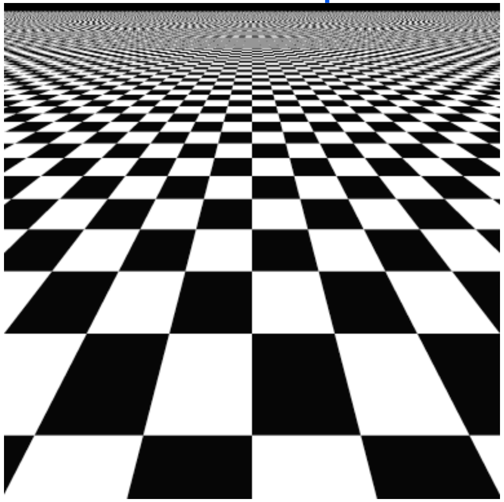
4 Samples



9 Samples



16 Samples



Box vs Tent filtering over regular grids --- Animation

Box vs Tent filtering over regular grids --- Animation