Computer Graphics

Lecture 09

Aliasing - Point Sampling

We live in an analog world. There are an infinite amount of colors to paint an object (even if the difference is indiscernible to our eye), there are an infinite number of tones we can hear, and there are an infinite number of smells we can smell. The common theme among all of these analog signals is their **infinite** possibilities.

Digital signals and objects deal in the realm of the **discrete** or **finite**, meaning there is a limited set of values they can be. That could mean just two total possible values, anything as long as it's not ∞ (infinity).

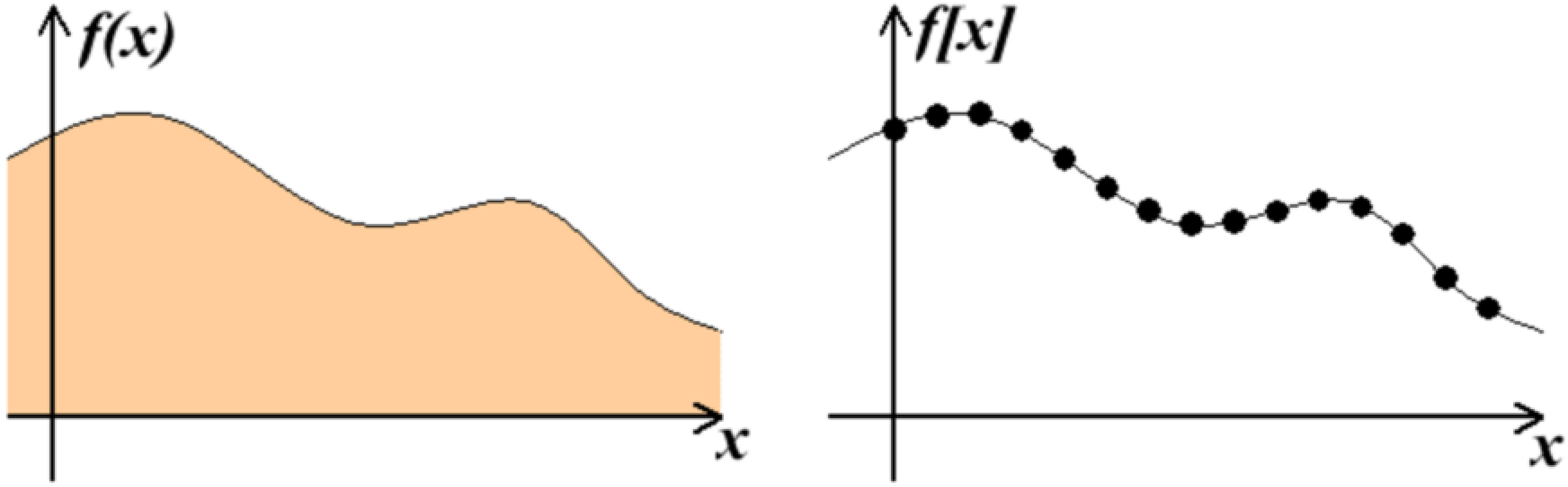
We think of most things in the real world as continuous, however, everything in a computer is discrete

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*

Point Sampling

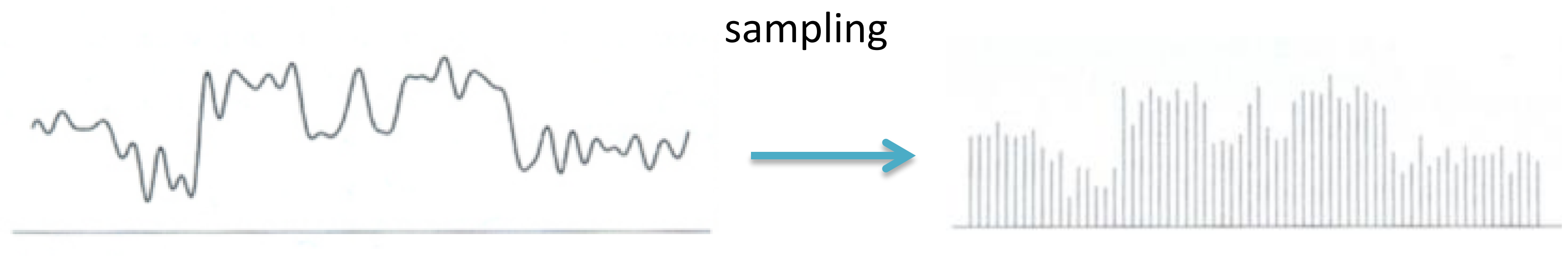
When we represent or render an image using a computer we must both sample and quantize



Sampling

How to store and compute with continuous functions?

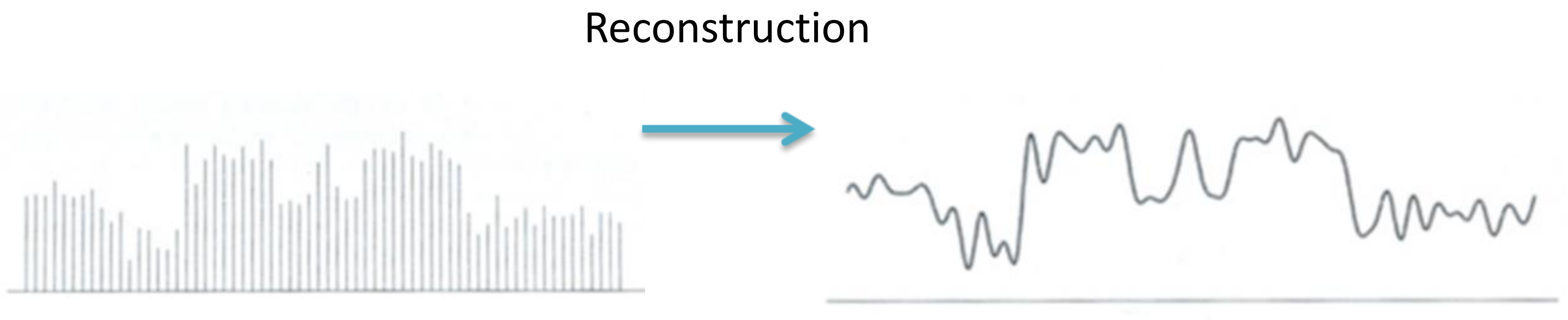
Common scheme for representation: samples – write down the function's values at many points



Reconstruction

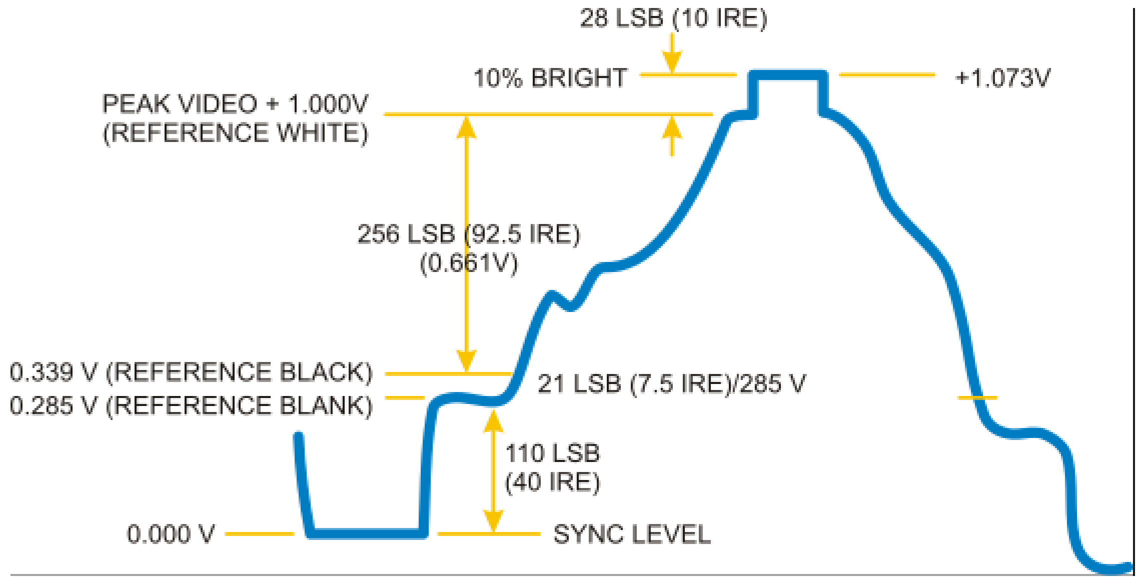
Making samples back into a continuous function

- for output (need realizable method)
- for analysis or processing (need mathematical method)
- amounts to “guessing” what the function did in between



Analog and Digital Signals

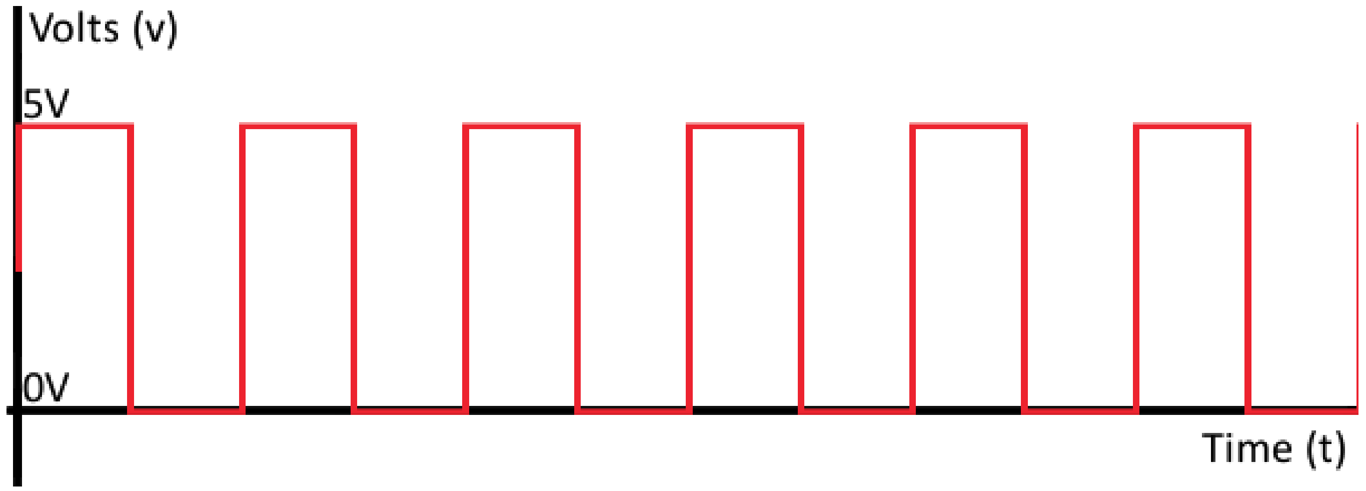
Video and audio transmissions are often transferred or recorded using analog signals. Tiny changes in the signal have a huge effect on the color or location of the video.



https://learn.sparkfun.com

Analog and Digital Signals

Digital signals must have a finite set of possible values. The number of values in the set can be anywhere between two and a-very-large-number-that's-not-infinity. Most commonly digital signals will be one of two values – like either 0V or 5V. Timing graphs of these signals look like square waves.

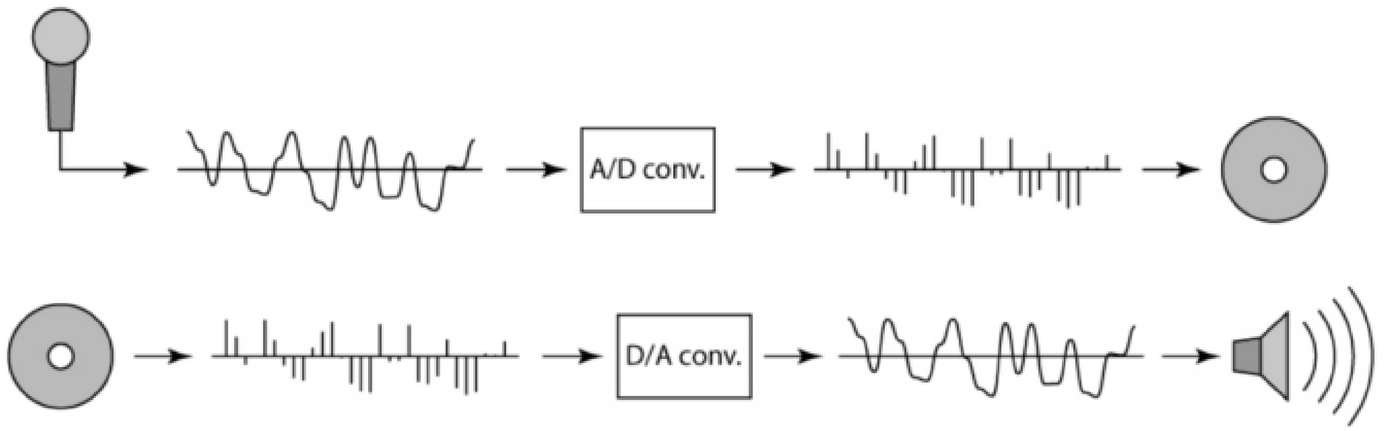


https://learn.sparkfun.com

Sampling examples

Recording: sound to analog to samples to disc

Playback: disc to samples to analog to sound again



Can we hear the original as is? No

http://www.cs.cornell.edu/

Aliasing

Missing out the information in the reconstruction process is called *Aliasing*

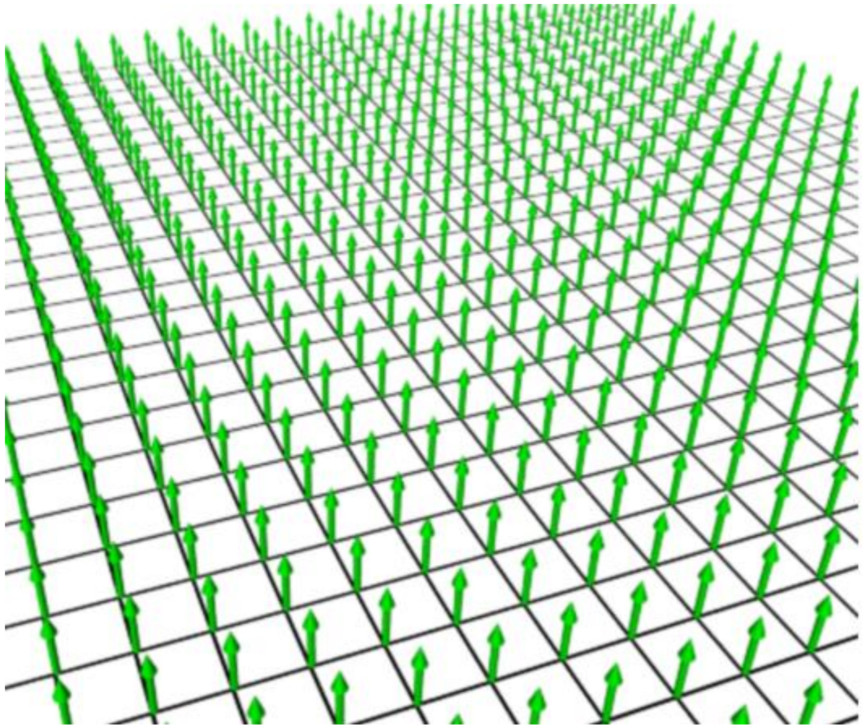
Aliasing occurs in computer graphics

- We are point sampling an analog signal
- The mathematical model of an image is a continuous analog signal which is sampled at discrete points (the pixel positions)
- A pixel is more than just a point, it is a sample

Picturing an Image as a 2D Function

- An ideal image can be viewed as a function, *f*(x, y), that gives an intensity for any given coordinate (x, y). We could plot this function as a height field. This plot would lowest at dark points in the image and highest at bright points.
- An image, in general, cannot be represented with such a simple form. Instead, we represent images as tabulated functions.

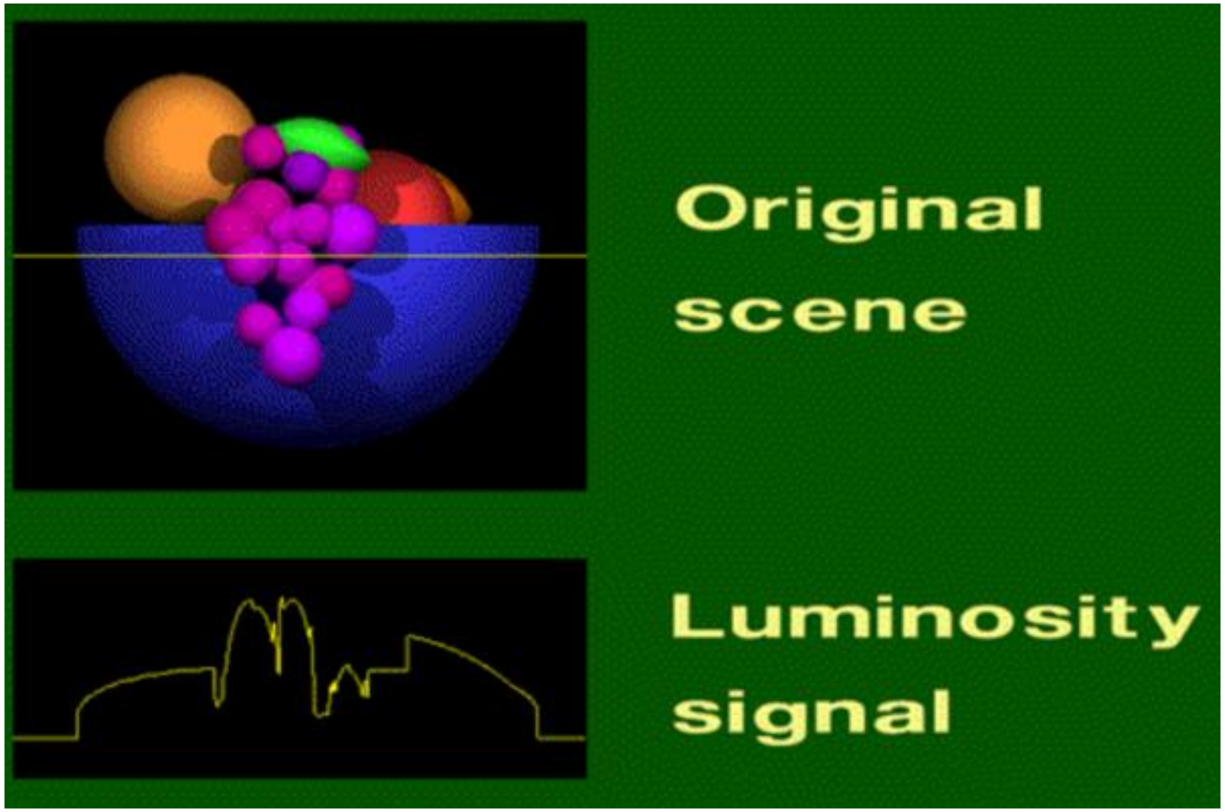
Sampling Grid



When a continuous image is multiplied by a sampling grid a discrete set of points are generated. These points are called samples. These samples are pixels. We store them in memory as arrays of numbers representing the intensity of the underlying function.

The most common way to generate the table values necessary to represent our function is to multiply the function by a sampling grid. A sampling grid is composed of periodically spaced

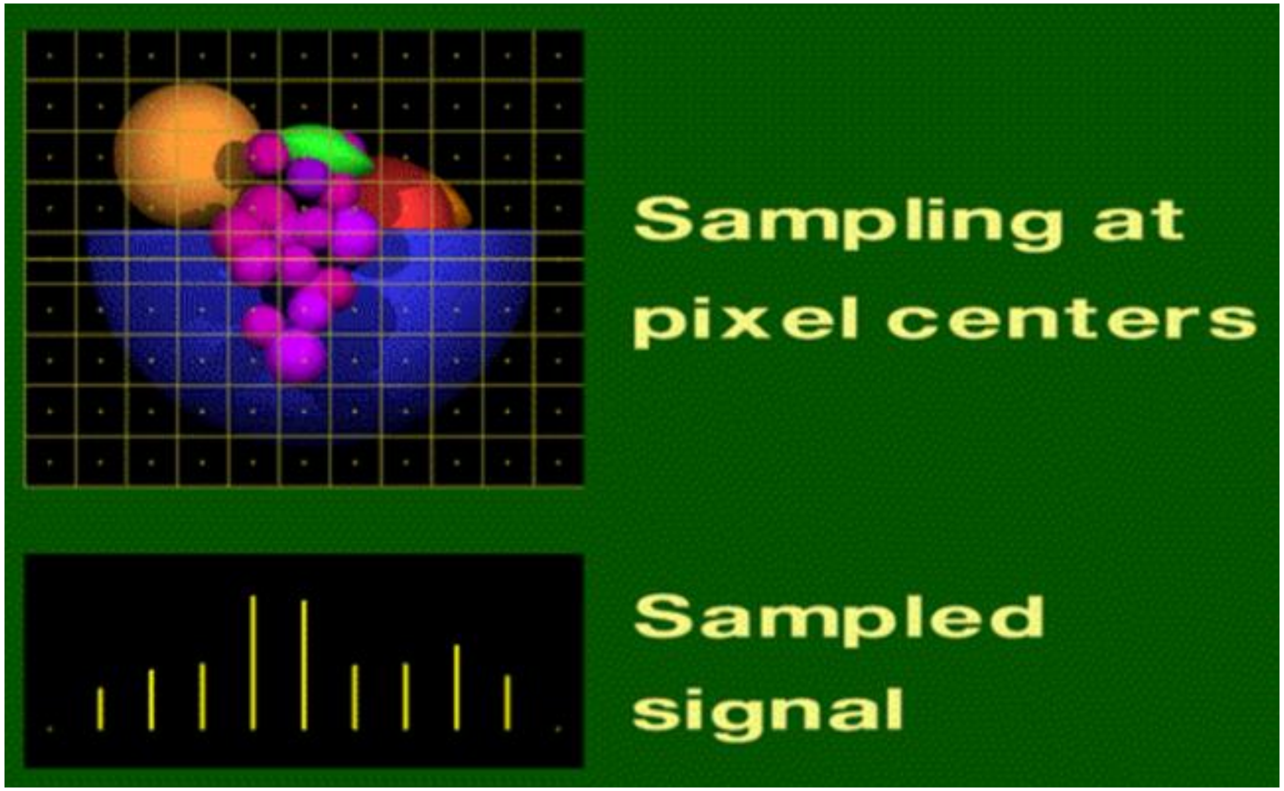
Aliasing in Computer Graphics



One scanline in the fruitbowl is highlighted.
The graph shows the luminosity (brightness) function of the highlighted scan line.

http://www.siggraph.org/education

Sampling the scene



The rectangular grid superimposed over the fruit bowl shows the size of the pixels

The dot in the middle of each square shows the position of a sample.

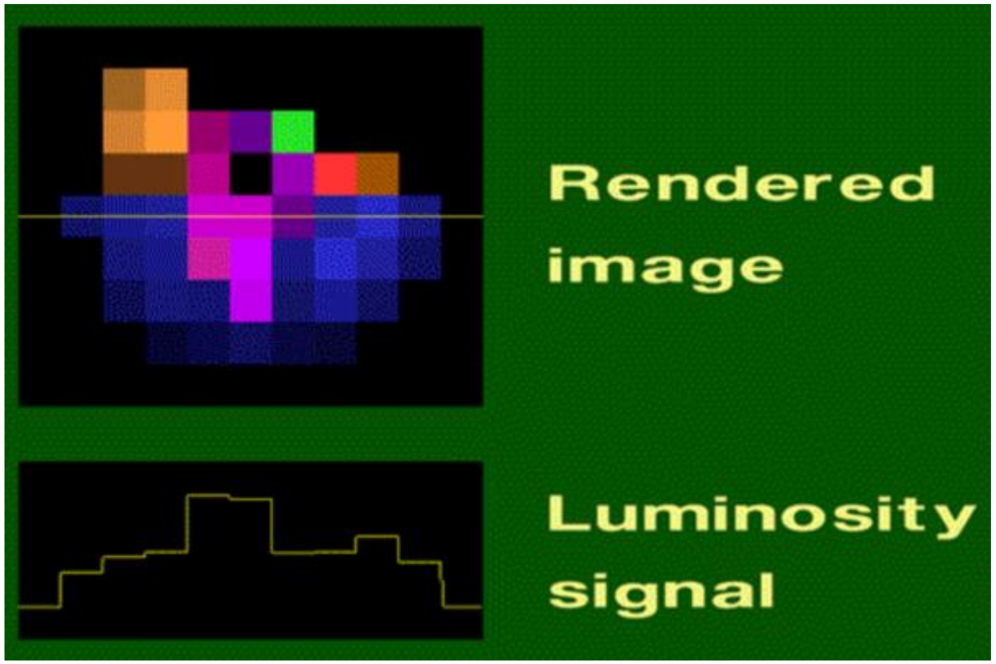
The color at the sample point will be the color of the pixel in the rendered image.

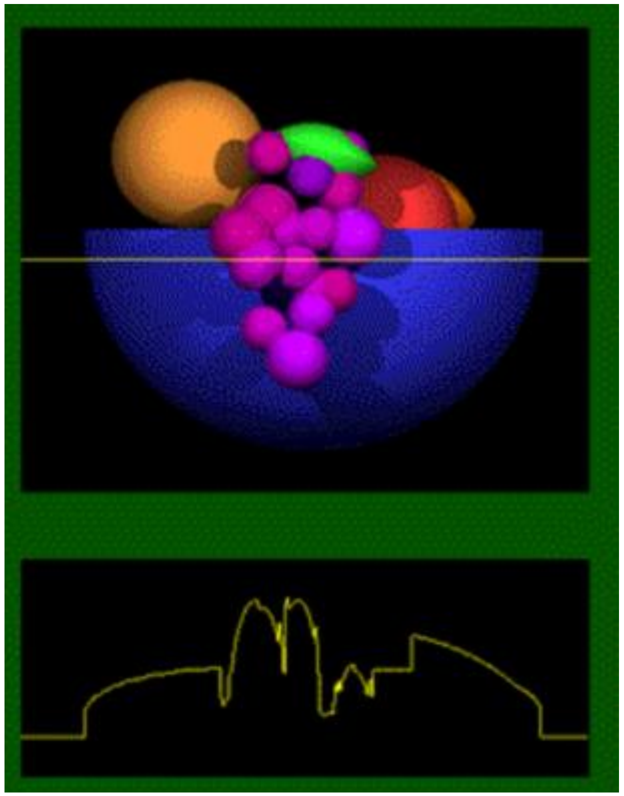
The graph shows the corresponding sampled luminosity function of the highlighted scanline.

Due to sampling a lot of information has been lost.

http://www.siggraph.org/education

Rendered image



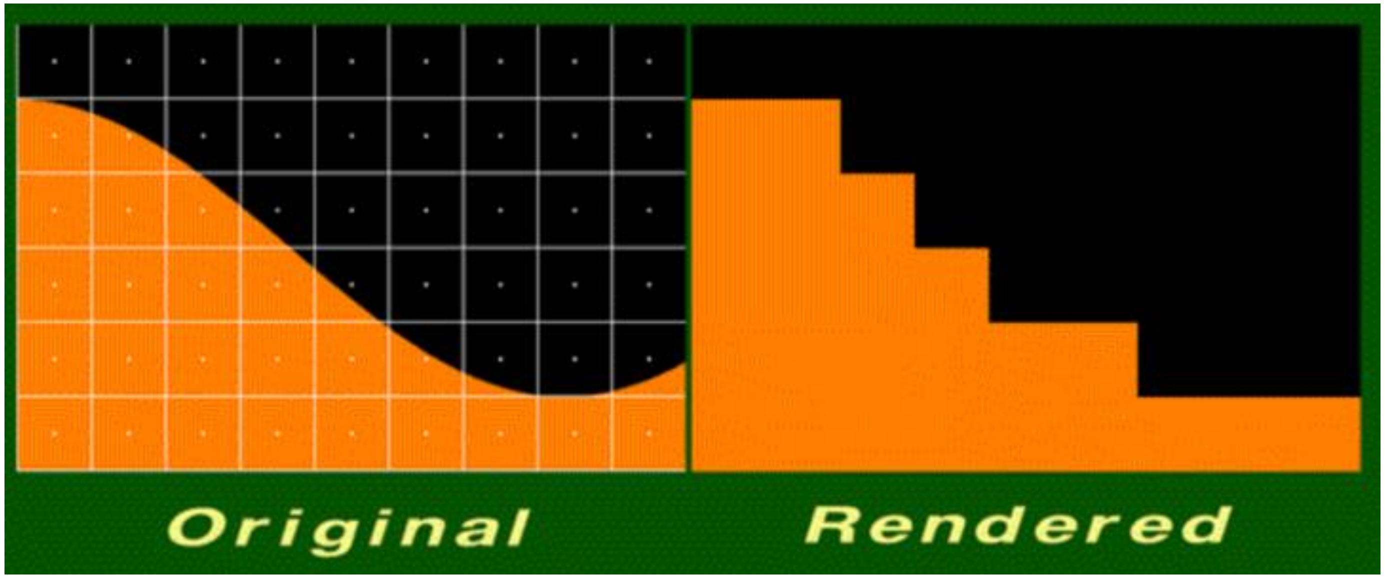


The rendered image differs greatly from the original scene, as does the luminosity signal. Notice that the green leaf has moved to the right in the rendered image.

http://www.siggraph.org/education

Aliasing Effects

Jagged profiles

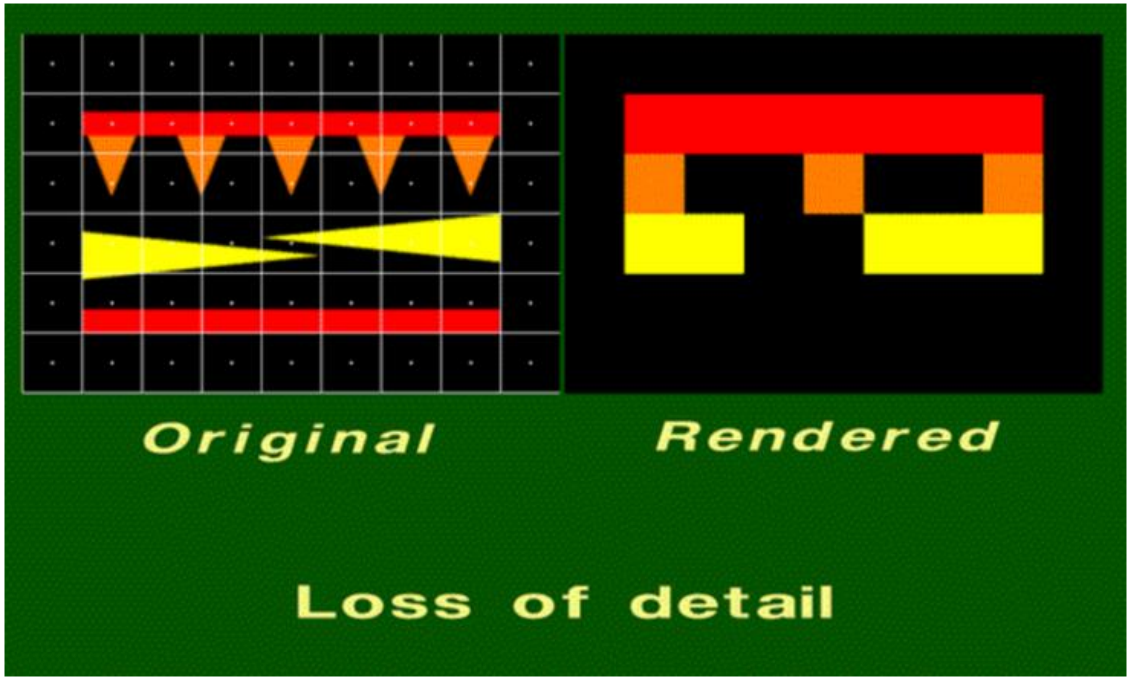


The picture on the left shows the sampling grid superimposed on the original scene. The picture on the right is the rendered image.

Also known as "jaggies", jagged silhouettes are probably the most familiar effect caused by aliasing.

Aliasing Effects

Improperly rendered details



The original scene on the left shows a group of small polygons.

In the rendered scene, one of the two red rectangles disappears entirely, and the other doubles in width.

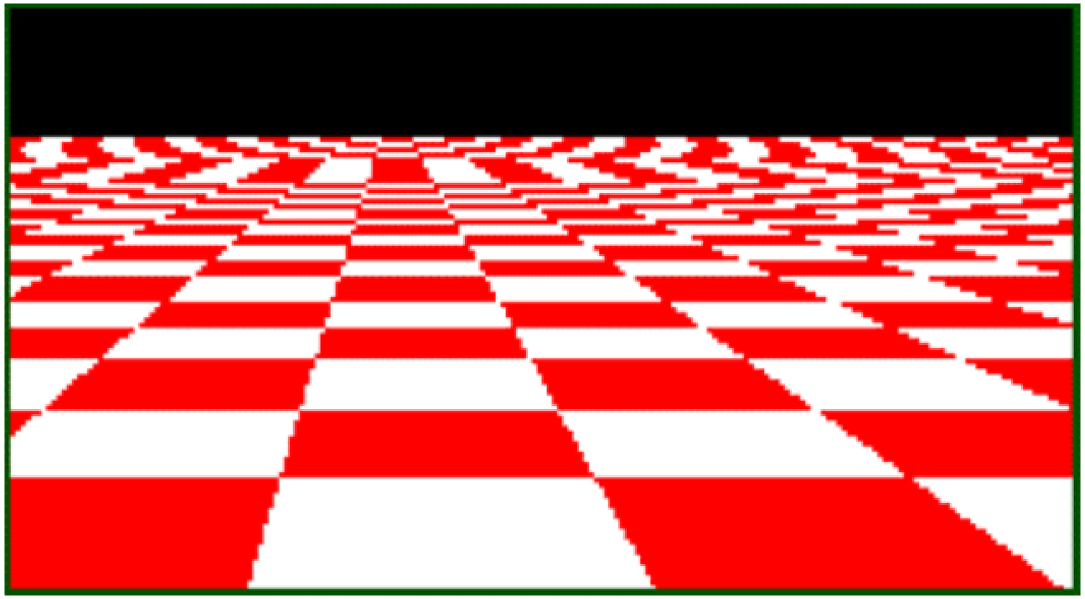
Two of the orange triangles disappear.

Although the two yellow triangles are identical in size, one is larger than the other in the rendered image.

http://www.siggraph.org/education

Aliasing Effects

Disintegrating textures



This is a checkered texture on a plane.

The checkers should become smaller as the distance from the viewer increases.

However, the checkers become larger or irregularly shaped when their distance from the viewer becomes too great.

Simply increasing the resolution will not remove this artefact.

Increasing the resolution will only move the artefact closer to the horizon.

http://www.siggraph.org/education

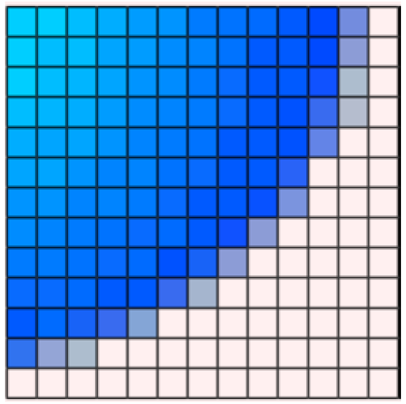
Antialiasing

antialiasing algorithms

- As aliasing problem is due to low resolution, one easy solution is to increase the resolution , causing sample points to occur more frequently. This increases the cost of image production.
- The image is created at high resolution and then digitally filtered. This method is called supersampling or postfiltering and eliminates high frequencies which are the source of aliases.
- The image can be calculated by considering the intensities over a particular region. This is called prefiltering.

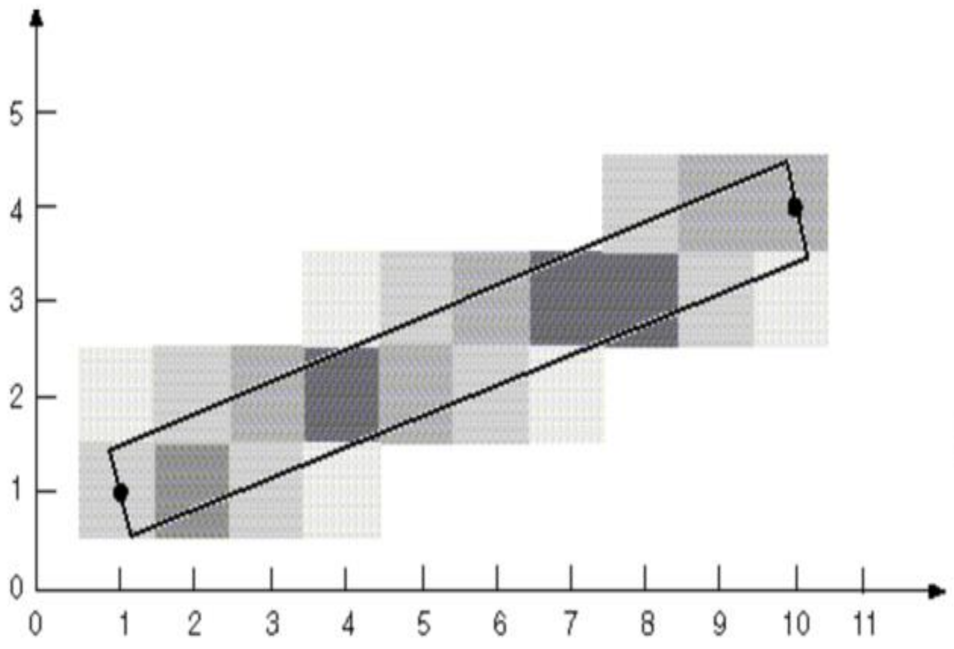
Prefiltering

- Prefiltering methods treat a pixel as an area, and compute pixel color based on the overlap of the scene's objects with a pixel's area.
- These techniques compute the shades of gray based on how much of a pixel's area is covered by a object.
- These techniques compute the shades of gray based on how much of a pixel's area is covered by a object.



Prefiltering

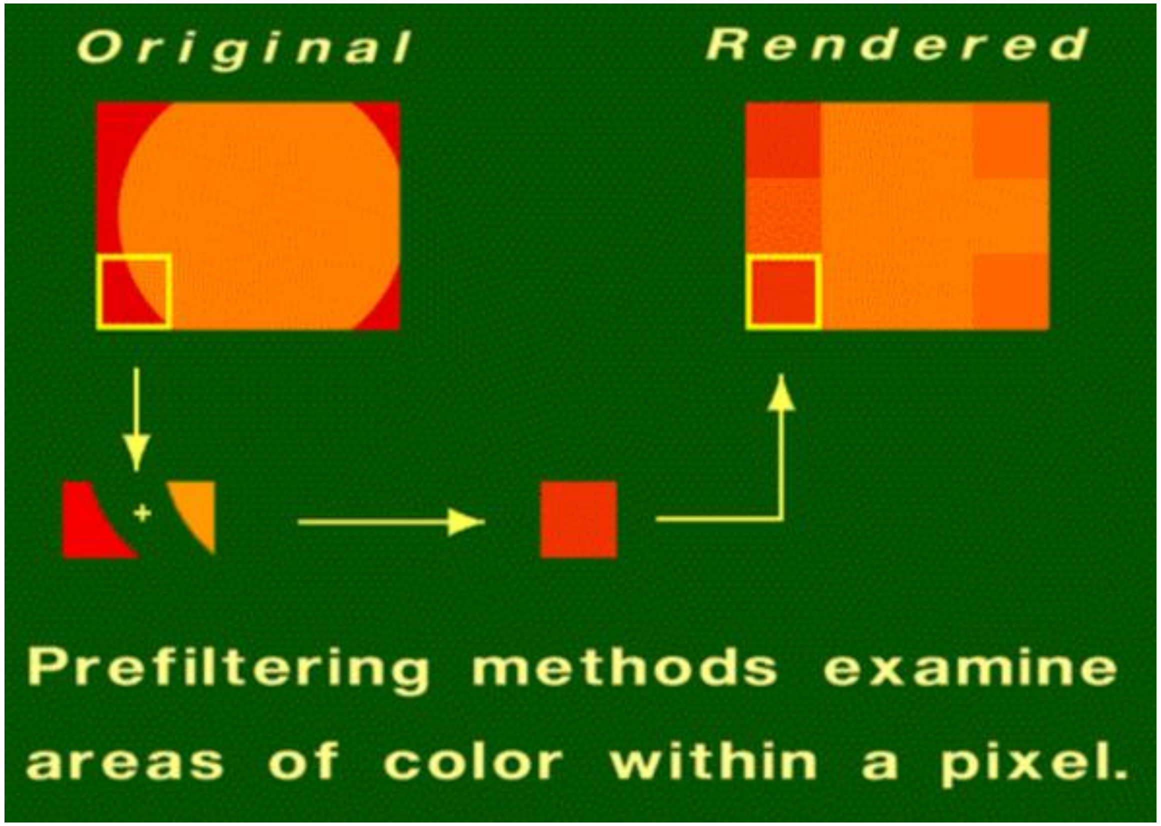
Antialiased Lines



Multiple pixels with varying intensity are used to represent the line. Each pixel intensity set as a function of the distance of the pixel to the line.

http://groups.csail.mit.edu/graphics

Basis for Prefiltering Algorithms



The original scene is a filled orange circle on a red background.

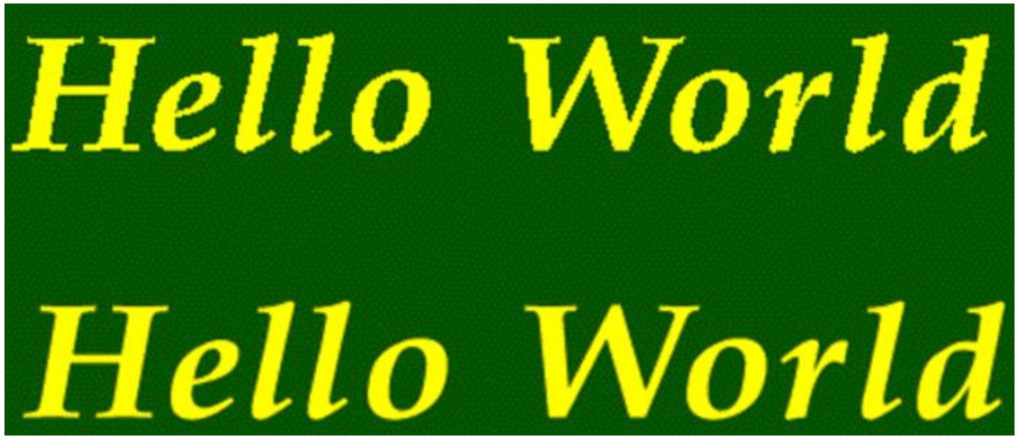
All of the pixels inside the circle are 100 percent orange.

All the pixels on the boundary of the circle have some area that is red and some area that is orange.

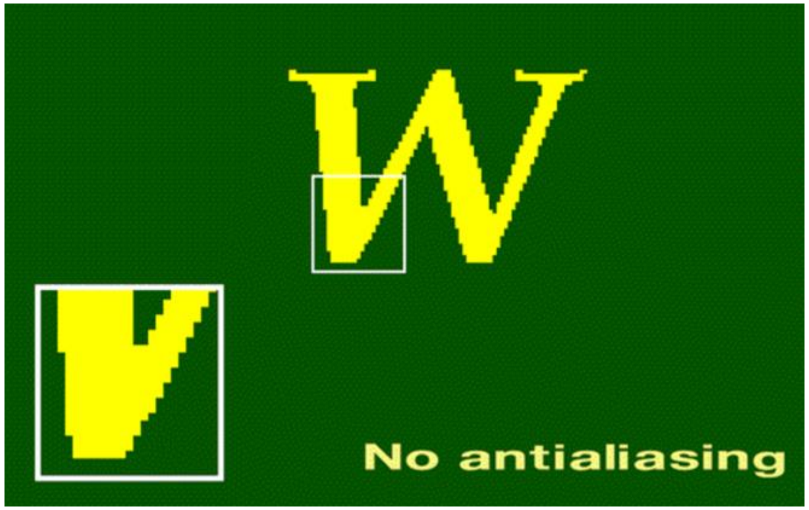
Forty percent of the highlighted pixel is orange and 60 percent of its area is red.

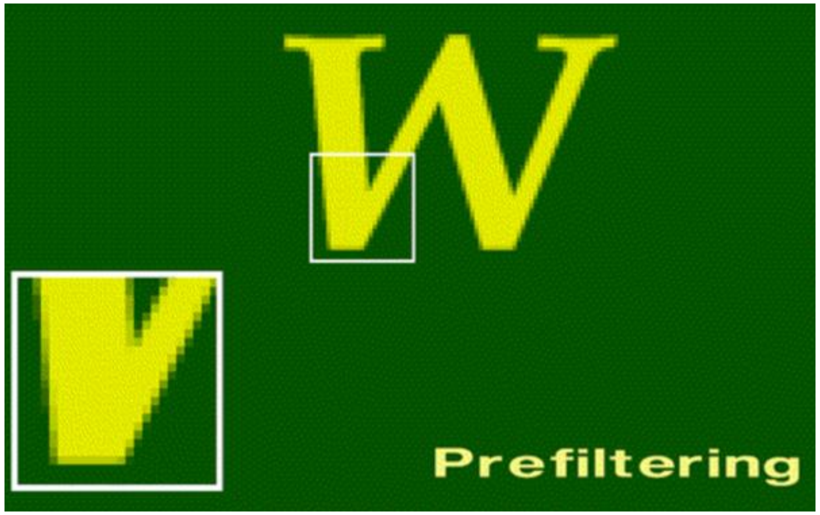
The computed color for the highlighted pixel is 40 percent orange and 60 percent red.

Prefiltering results



rendered at a resolution of 512x512





Postfiltering

Postfiltering is the more popular approach to antialiasing. For each displayed pixel, a postfiltering method takes several samples from the scene and computes an average of the samples to determine the pixel's color.

This process means calculating a virtual image at a higher spatial resolution than the frame store resolution and then averaging down to the final resolution

Postfiltering Algorithms

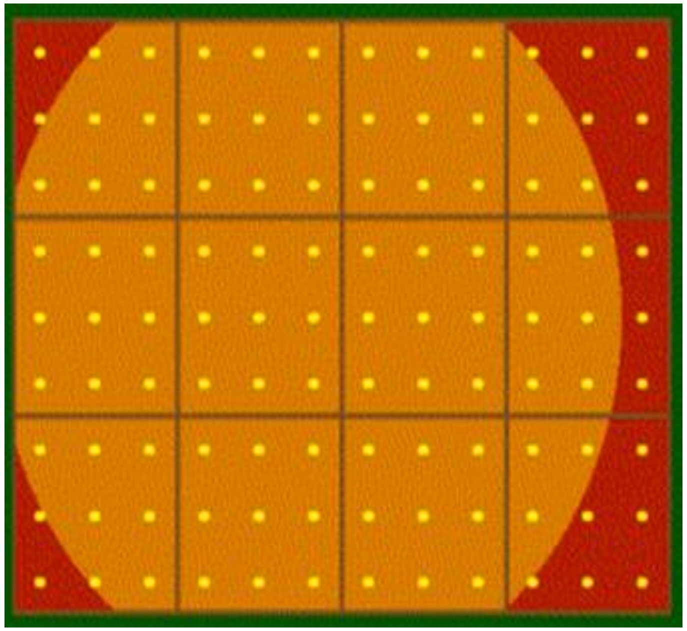
Example:

suppose the display resolution is 512x512. Sampling at three times the width and three times the height of the display resolution would yield 1536x1536 samples.

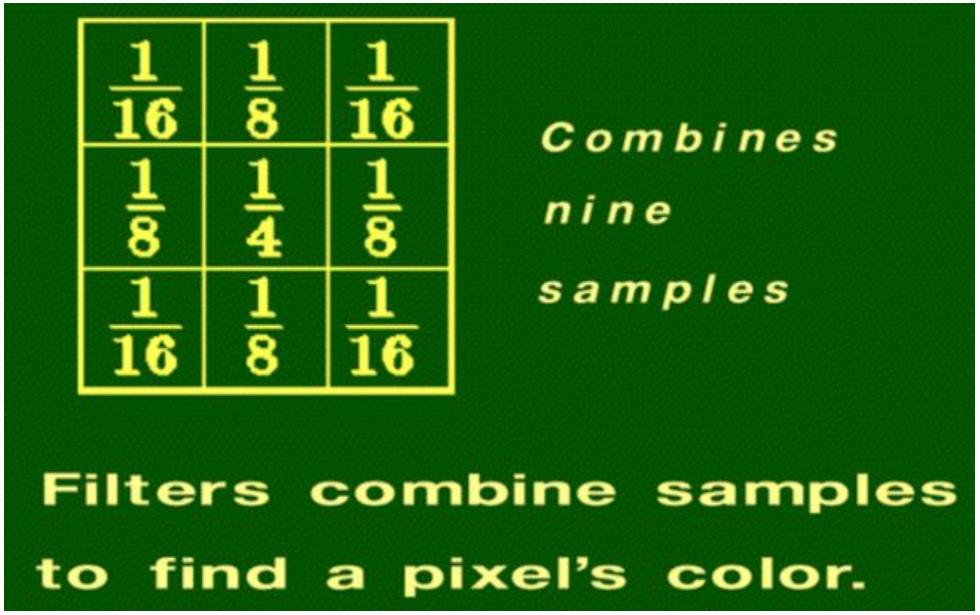
The color of each pixel in the rendered image will be an average of several samples.

if sampling were performed at three times the width and three times the height of the display resolution, then a pixel's color would be an average of nine samples.

A filter provides the weights used to compute the average.



Filtering



$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$
$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$
$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$

Combines nine samples

Filters combine samples to find a pixel's color.

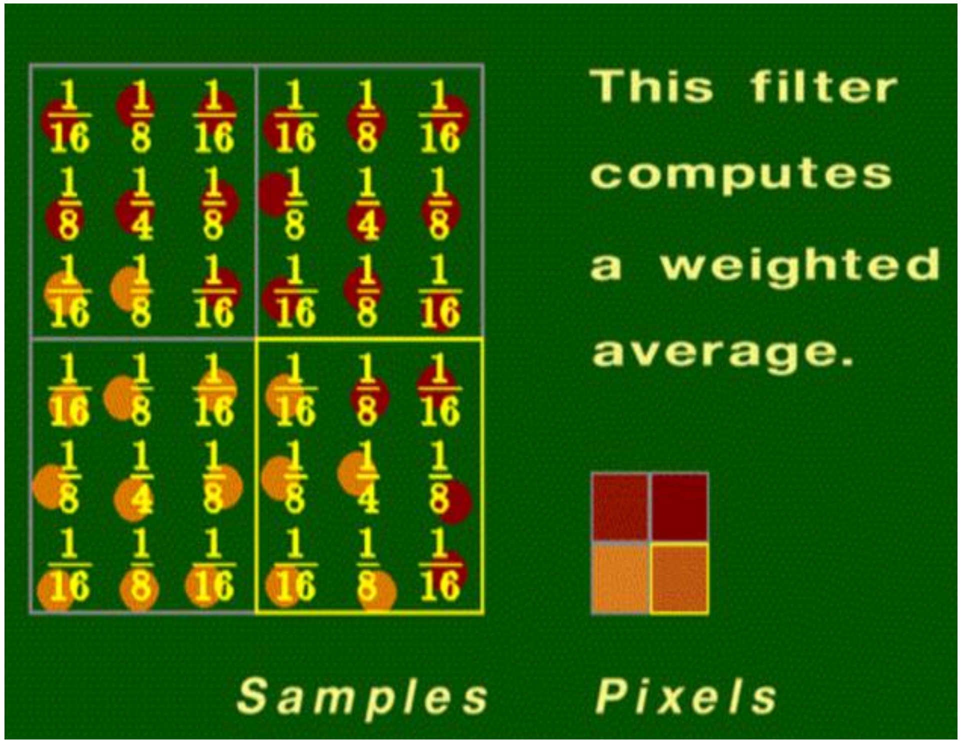
Filters combine samples to compute a pixel's color

Each sample is multiplied by its corresponding weight and the products are summed to produce a weighted average, which is used as the pixel color

The other type of filter is an unweighted filter. In an unweighted filter, each sample has equal influence in determining the pixel's color.

In other words, an unweighted filter computes an unweighted average.

filter to compute a pixel's color

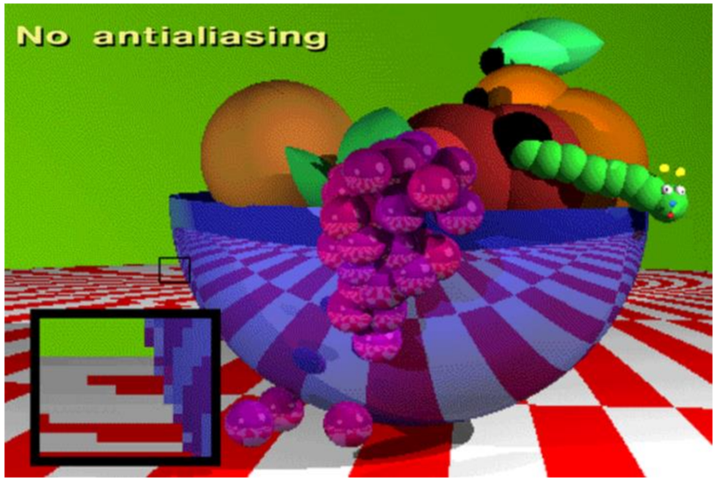


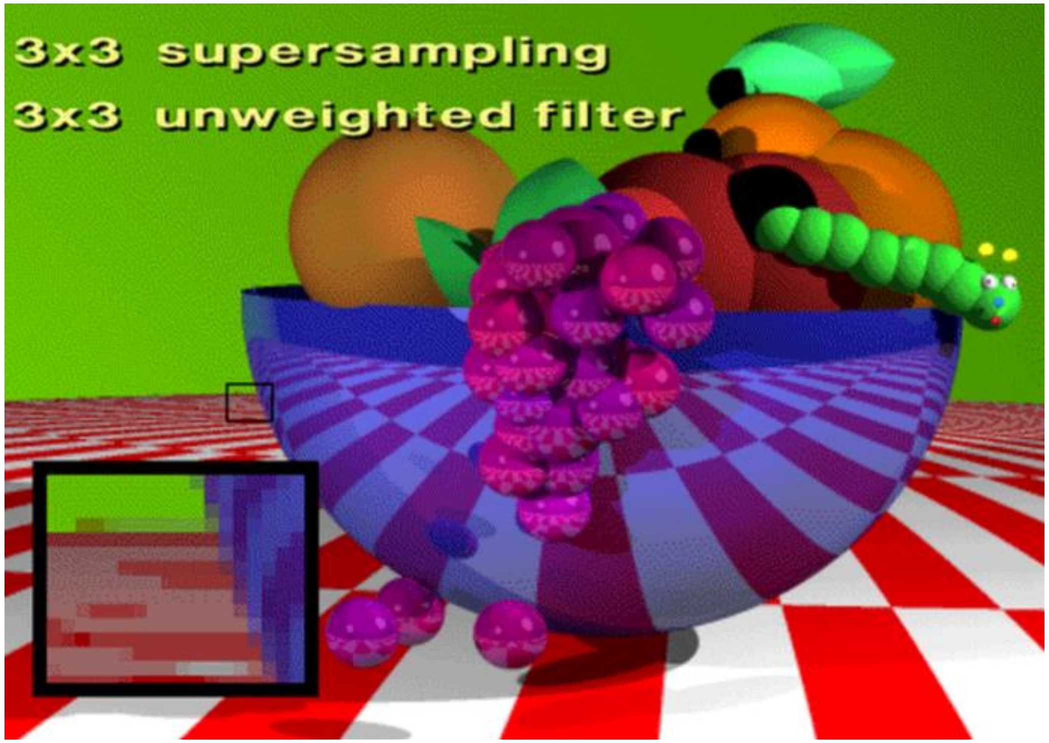
On the left is a group of samples that will be combined to produce pixel colors. Some of the samples are orange, and some are red.

The superimposed numbers are the weights from the filter shown in the last slide.

The highlighted group of samples on the left are combined using the filter to produce the color of the highlighted pixel on the right.

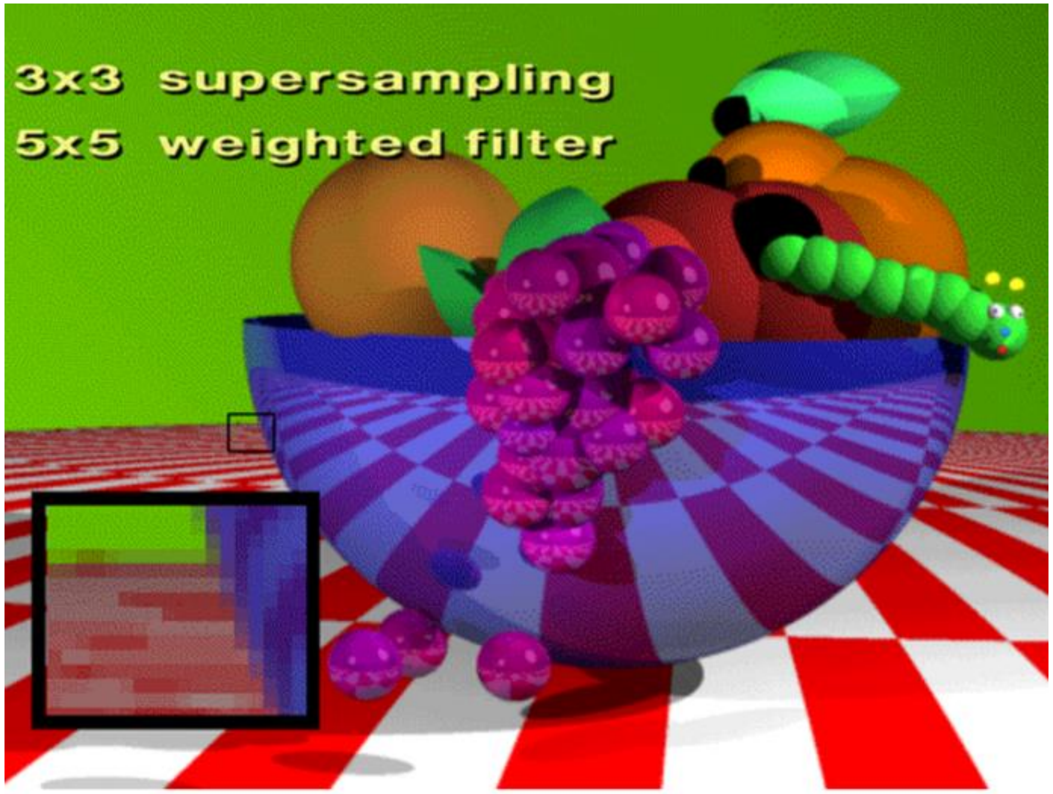
Postfiltering Example





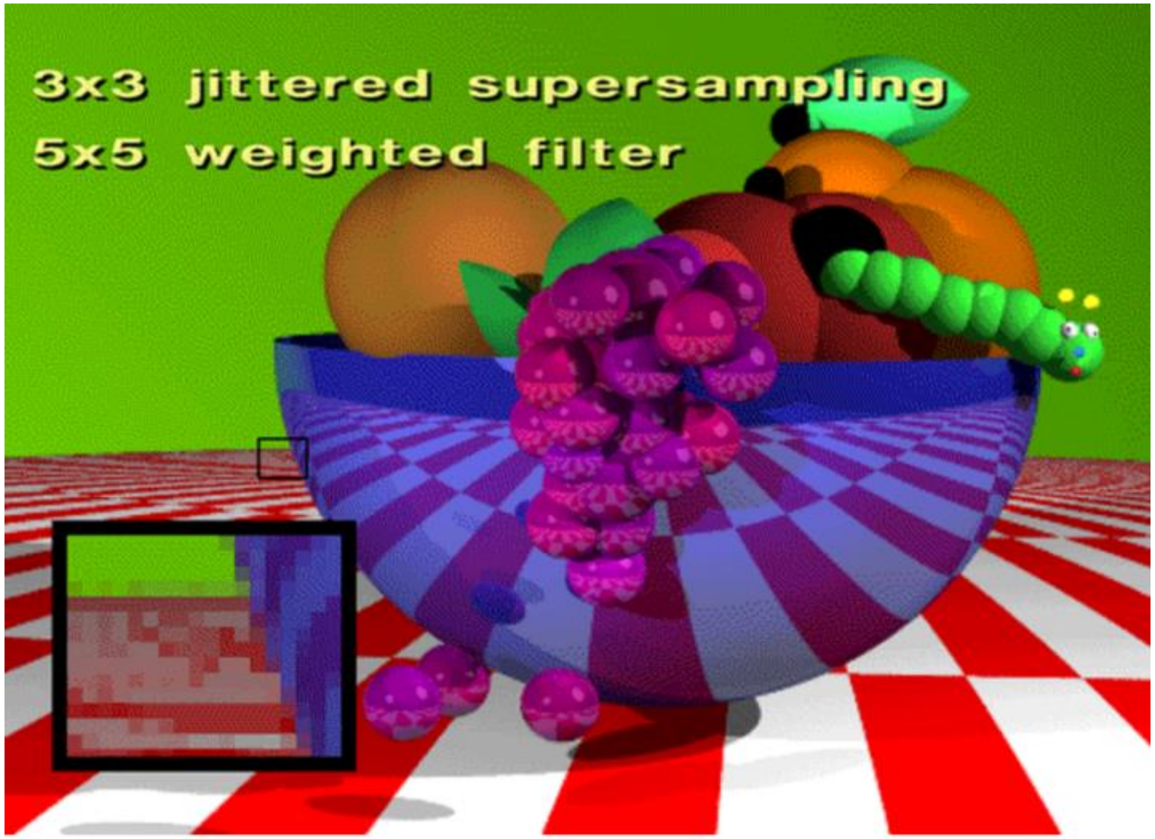
Using an unweighted filter. Each of the nine samples taken for a pixel had equal influence on the pixel's computed color.

Postfiltering Example



The samples are regularly spaced.
 Here's the weighted filter used in this rendering: 1/81 2/81 3/81 2/81 1/81 2/81 4/81 6/81 4/81 2/81 3/81 6/81 9/81 6/81 3/81 2/81 4/81 6/81 4/81 2/81 1/81 2/81 3/81 2/81 1/81 This filter allows samples taken from neighboring pixels to influence the color of the current pixel. The result is softer profiles and edges.

Postfiltering Example



Each sample is perturbed by a random amount in x and y. The random amount is constrained to be less than one sixth of the pixel's width.

Reference

http://www.siggraph.org/education/materials
http://groups.csail.mit.edu/graphics
http://www.cs.cornell.edu/