

# Computer Graphics: Sketch Part 1

## Entertainment a driving force

Movies

Games

## Real Time – Apparent Realism

## Powerful Tools and Fast Computation

GPU is a SIMD device

Every pixel can be treated as a processor

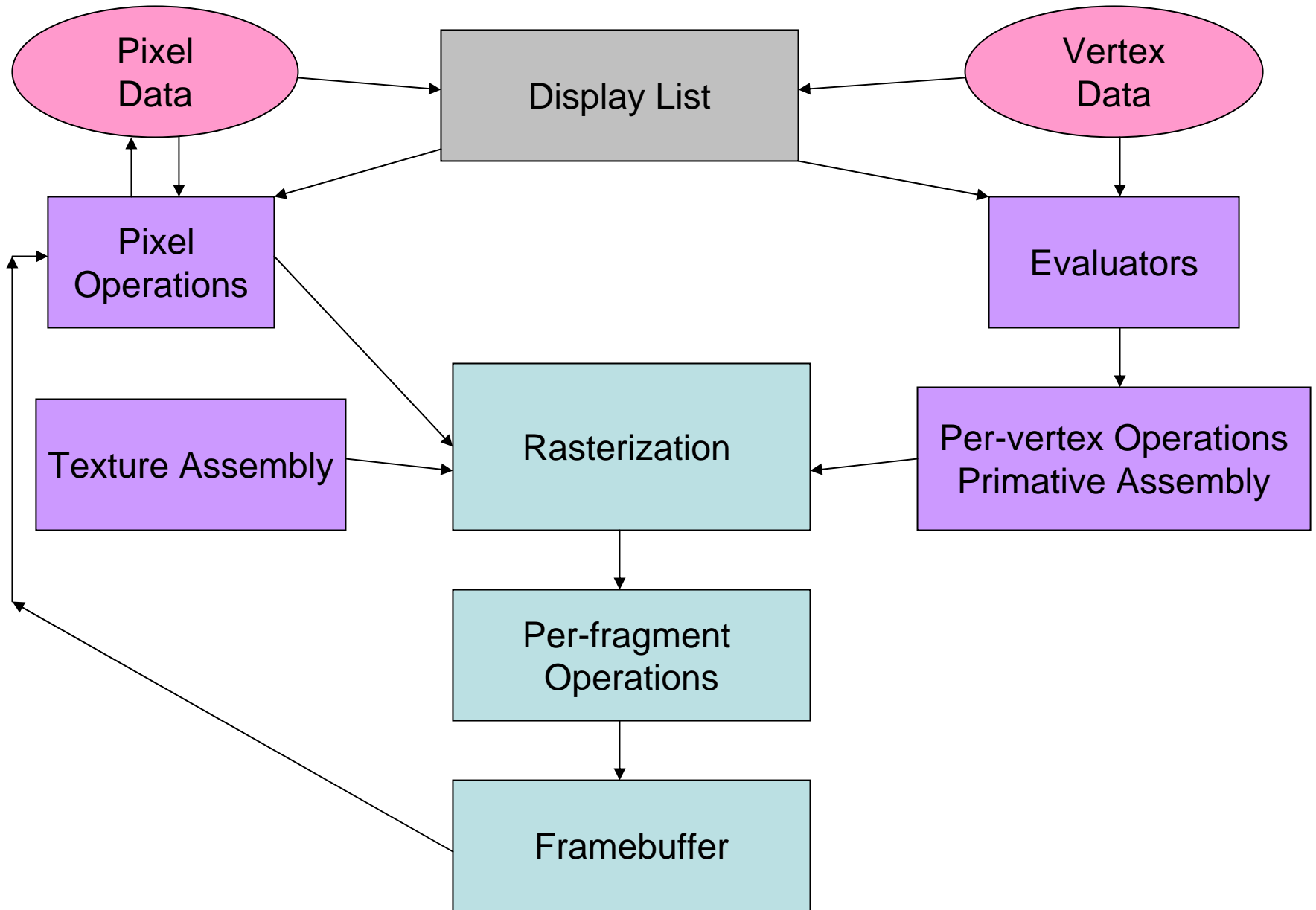
Fast FFT

Navier Stokes Equations

Optimization

## Begging for more use by scientists

# OpenGL Rendering Pipeline



# Rasterization and Fragment Operations

## Conversion of geometric and pixel data into fragments

A fragment square corresponds to a pixel in the frame buffer

Operation included:

- Line and polygon stiple

- Line width and point size

- Shading model and coverage

- Above provide support for antialiasing

Vertices connected into lines

Polygons filled

Depth and Color Calculated

## Fragment Operations

Operation may alter or throw out pixels

- Texturing and fog calculations

- Tests: scissor, alpha, stencil, depth

- Logical operations, blending, dithering, bitmasking

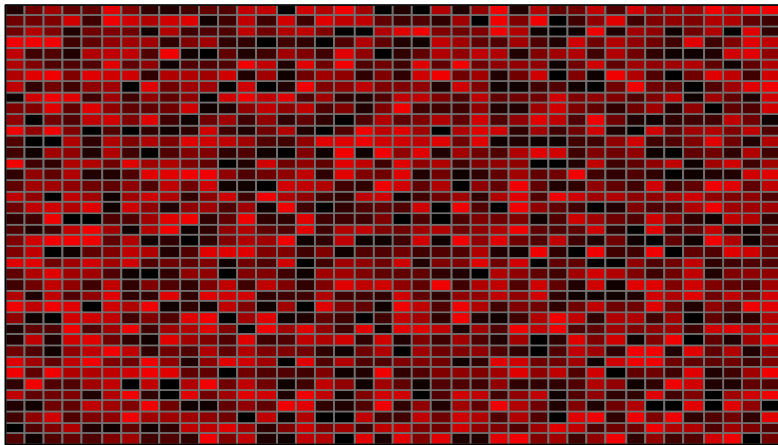
Result: pixels in a framebuffer

# Red, Green, and Blue Intensities (0-255 old graphics cards)

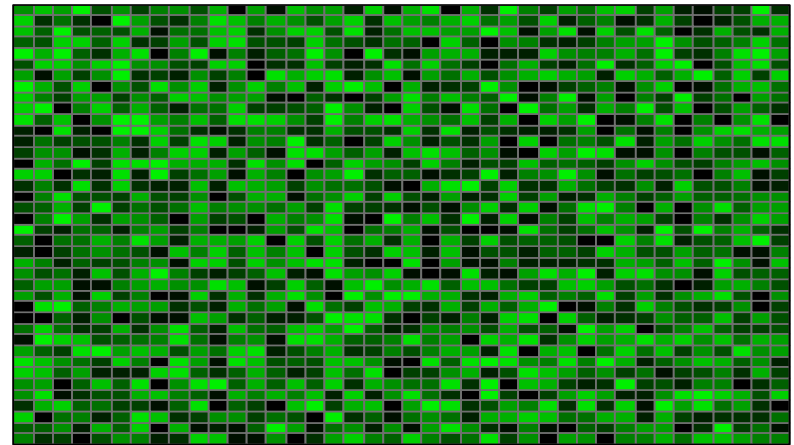
## Transformed with Gamma Correction to Voltage

### Controlling Phosphor Intensity

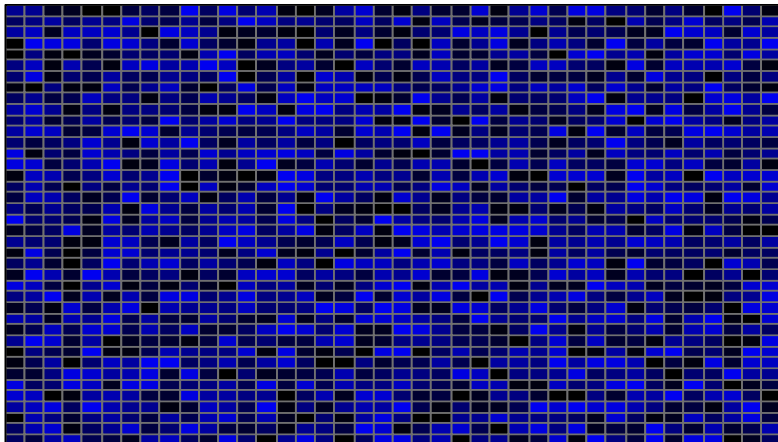
Red



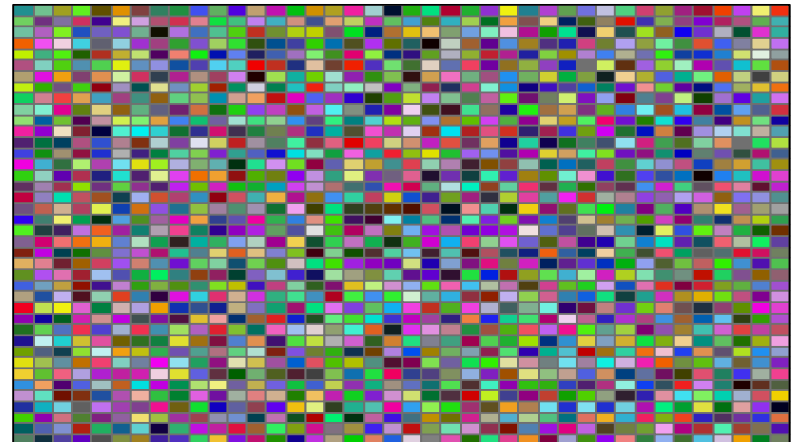
Green



Blue



Combined Result



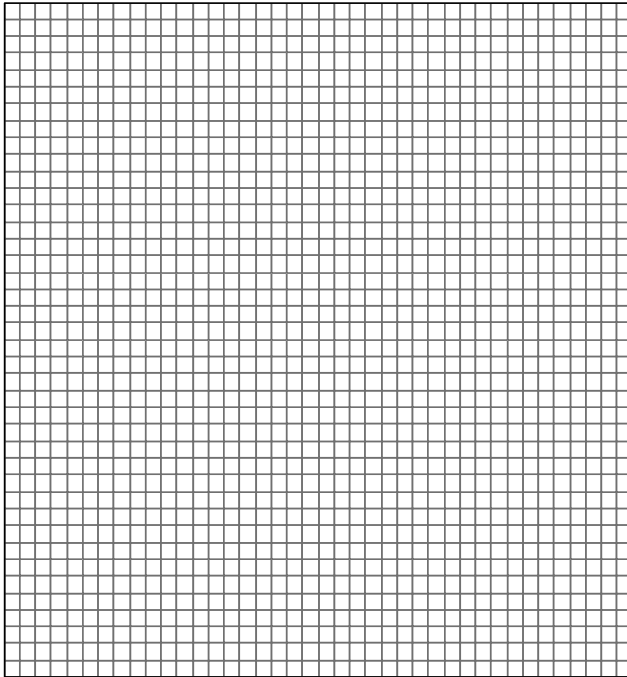
## Write on Pixel Control

**Alpha: weight or status of each pixel in the rgb planes**

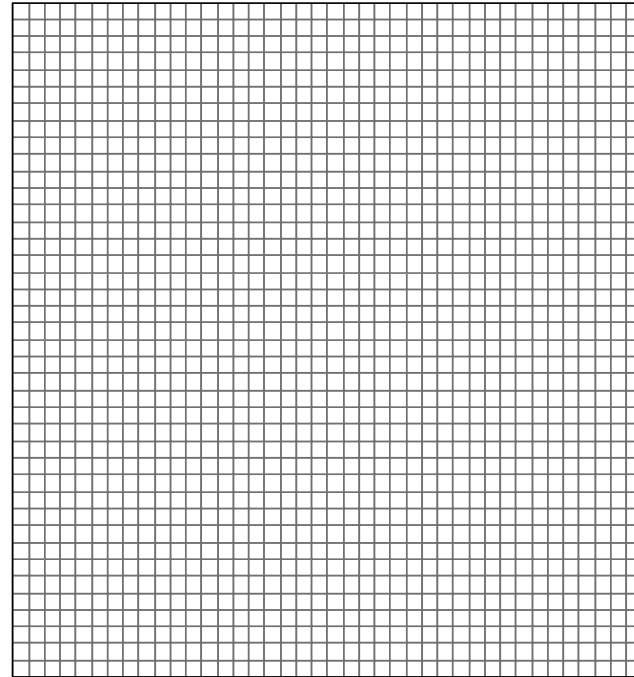
**Used with the input alpha and context for compositing**

**Z-buffer: depth of the rgb pixels in the rgb planes**

Alpha



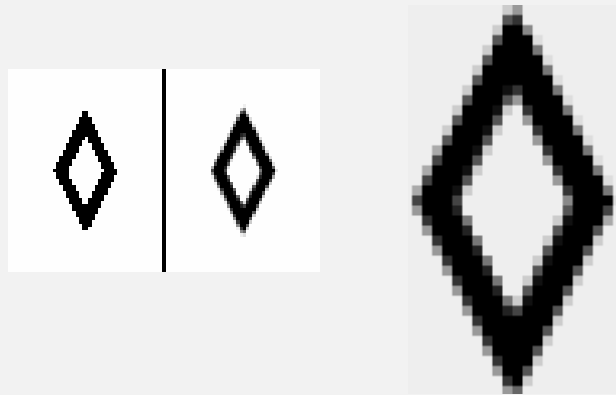
Z-Buffer



# Antialiasing

## From Wikipedia

**anti-aliasing** is the technique of minimizing aliasing (jagged, blocky or moiré patterns, popping, strobing, or unwanted sparkling) when representing a high-resolution signal at a lower resolution.



## Comments

Classic problem, rendering slanted lines with pixels

See Wikipedia for Mandelbrot examples and discussion of methods

Some graphics cards have antialiasing options

# Double Buffering

**See Wikipedia**

## **Problem**

Computer monitors constantly redraw the visible video page  
(at around 60 times a second)

Changes to the video page may show on the monitor before changes are complete

This results in ugly artifacts such as flickering, tearing and shearing.

## **Double buffering – class will use for animation**

Removes visible artifacts from the drawing process.

Hardware (page flipping solution)

Two graphics pages in VRAM.

One page is actively displayed

Draw into the background page

Flip pages when drawing is complete

Software approach, basic idea is similar

# Alpha compositing and blending

**See Wikipedia**

## **Problem**

Computer monitors constantly redraw the visible video page  
(at around 60 times a second)

Changes to the video page may show on the monitor before changes are complete

This results in ugly artifacts such as flickering, tearing and shearing.

## **Double buffering – class will use for animation**

Removes visible artifacts from the drawing process.

Hardware (page flipping solution)

Two graphics pages in VRAM.

One page is actively displayed

Clear, Update and/or Draw into the background page

Flip pages when drawing is complete

Software approach: the basic idea is similar

# Red, Green, Blue and Alpha channels

## Old video memory: 24 or 32 bits per pixel

8 bits each to control red, green, blue intensities

8 bits to control alpha compositing (not always included)

Integer intensity range 0-255 plus gamma correction

Options: Changeable color tables could reinterpret the bits

## Modern Graphic Processing Units (GPUs)

Floating point values with up to 4K x 4K pixel buffers

SIMD machine: conceptually a processor per pixel

Technically only  $n$  pixels handled at a time in parallel

$n$  is GPU dependent but a modest number like 24 or 32

Increasingly programmable

See *GPU Gems 2*, 2005. M. Phar Ed. Addison Wiley

Much faster for some tasks than CPUs

GPU speed doubling in 6 months rather than 18 months for CPUs