



# Anti-aliasing



Michael Doggett  
Department of Computer Science  
Lund university

# Project

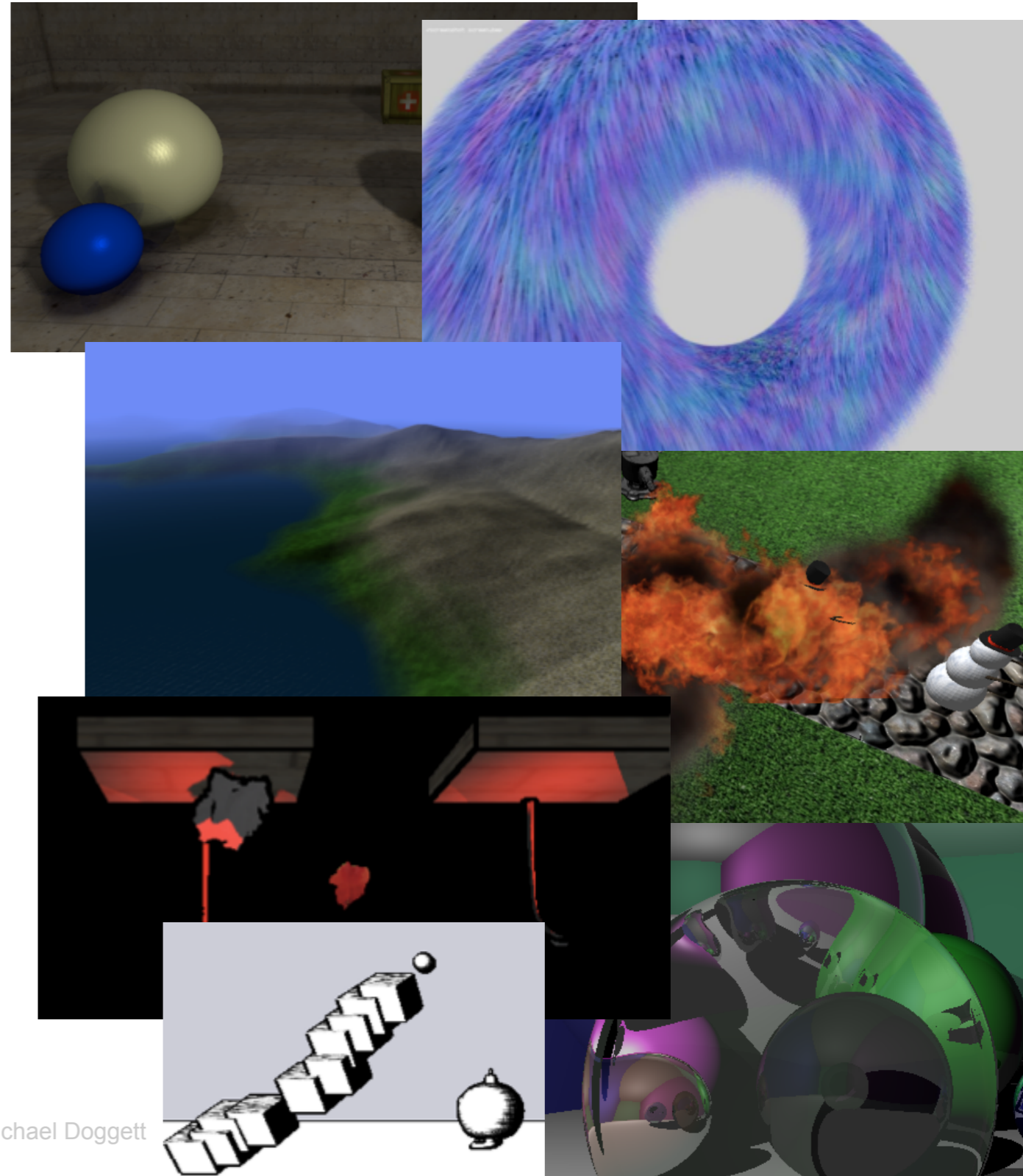
- Assignments are over, let's look at the project
- Should be done in a group of 2
- Each group writes a 2-4 page report
  - Check out LaTeX style on web page or use something similar
  - Make sure report includes references for web pages, papers, or other references used
- Submit a short video with report highlighting features of the project
- Final presentation includes a competition
- Total project corresponds to 3 points (3 weeks of work x 2 people)
- Project will be graded out of 10, and is worth 30% of final mark
- Finish by 2024-12-18, 13:00, Wednesday before last lecture

# 3D Graphics Project

- A chance to be creative
- A chance to create something that can be shown to potential employers (i.e. game company)
- For example :
  - Games (keep the mechanics simple, 2D)
  - Demo using 3D graphics algorithm
  - Check out previous year's projects on the web page
- Performance is not too important
  - But optimization is good, write about it in your report
- Formulate project, and get it approved
  - A long paragraph description of what your project will do, due Monday
- What is the best project?
  - A jury (with participants from industry) decides ...

# 3D Graphics Projects

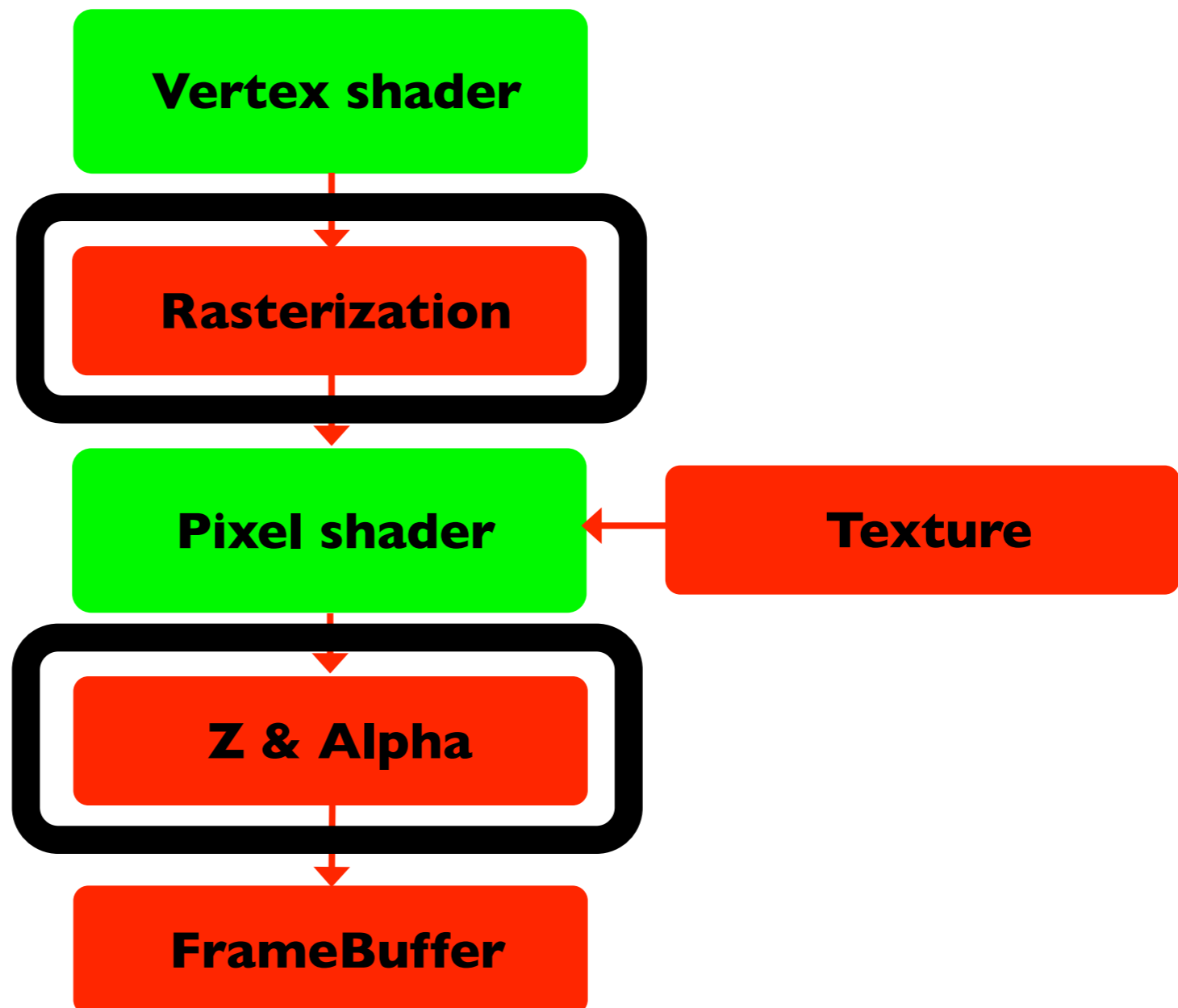
- SSAO
- Fur rendering
- Terrain Rendering
- Fire
- POM
- NPR - Outlines, hatching
- Temporal Anti-Aliasing
- Ray Tracing
- On web page look at :
  - Links
  - Previous years page



# 3D Graphics Projects

- Use Ray Tracer from Assignment 1
- Use Bonobo from Assignment 2
- For presentation/competition we provide a PC
  - or bring your own laptop
  - ~2 minute presentation

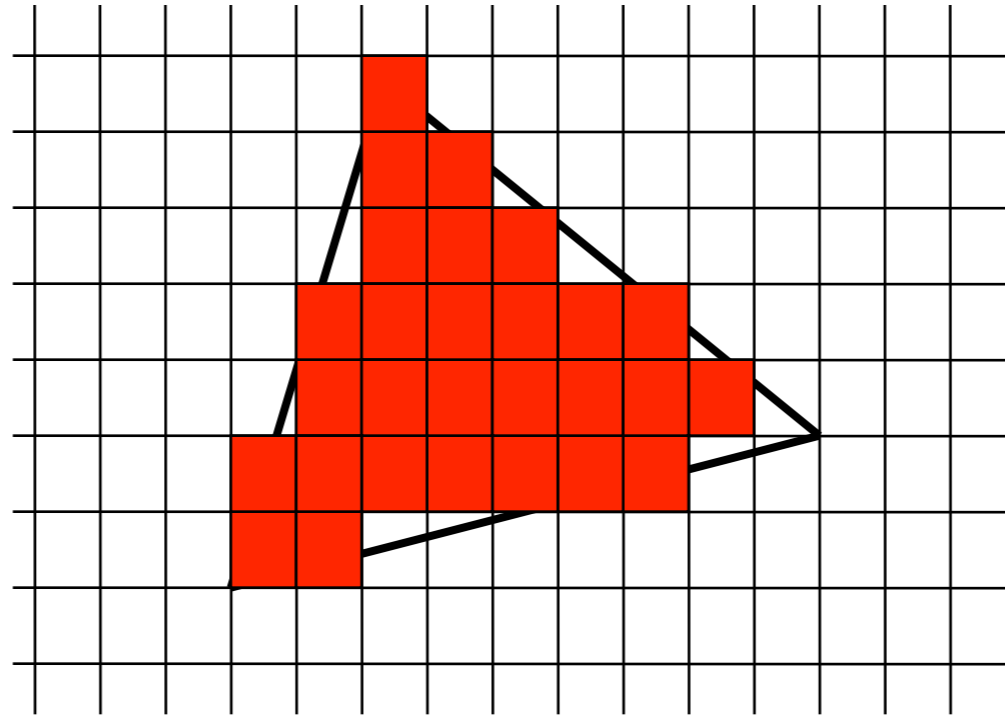
# Back to the Graphics Pipeline



# Anti-aliasing overview

- Sampling and reconstruction
- Supersampling
- Sampling Patterns
- Multisampling
- Screen Space, Resolve Pass AA
- Temporal AA

# Aliasing



- When rasterizing we get jagged edges
  - sometimes called “jaggies”
  - can look bad when animated
- Anti-aliasing is used to smooth out edges

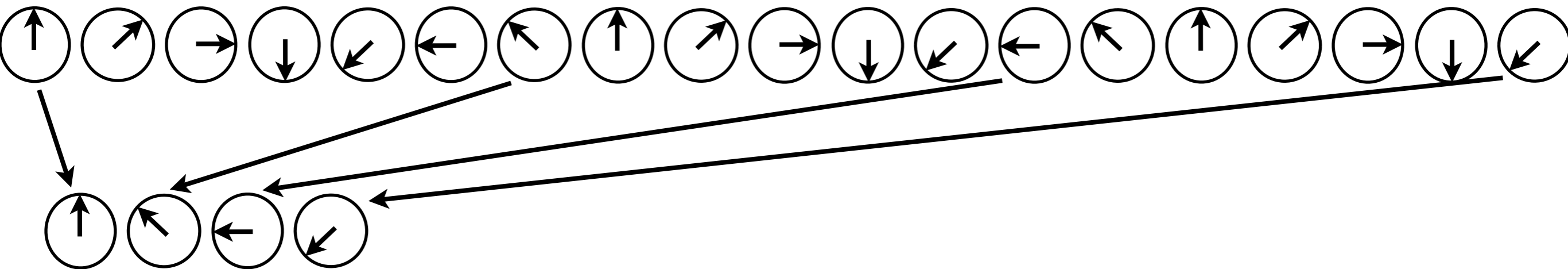


# Sampling

- Computer Graphics does a lot of :
  - Sampling, filtering and reconstruction
- Many objects are continuous, spheres, triangles
  - but are discretized by sampling

# Sampling Theory

- Sampling Theorem
  - A signal can be reconstructed from its samples without loss of information, if the original signal has no energy in frequencies at or above  $\frac{1}{2}$  the sampling frequency.
  - The minimum rate at which it must be sampled is the Nyquist frequency.



# Calculate everything that is in a pixel

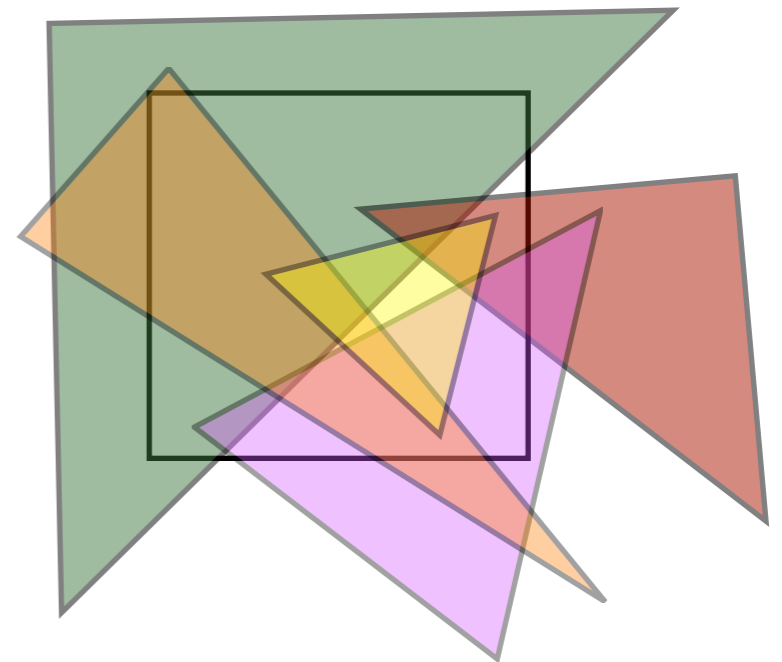
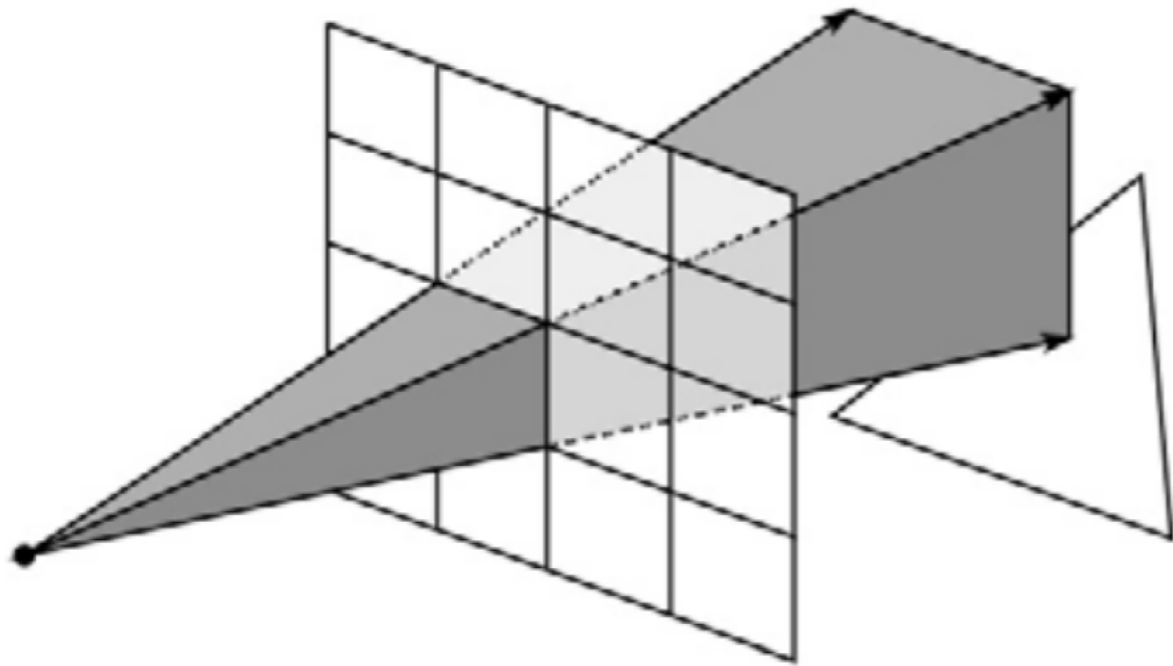
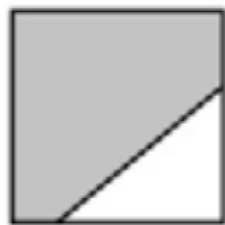


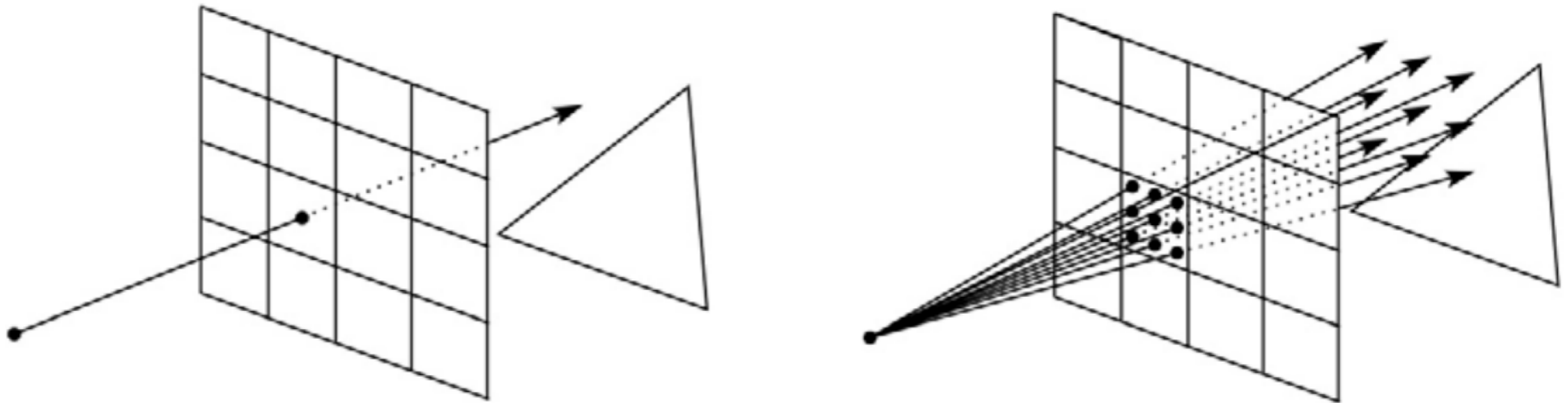
Image courtesy

Brian Curless CSE557

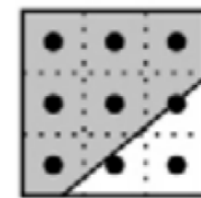
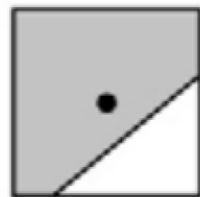


- Calculate area of triangle that covers pixel
- Divide by total area and multiply by color
- But this becomes complex ...

# Reduce aliasing by increasing samples

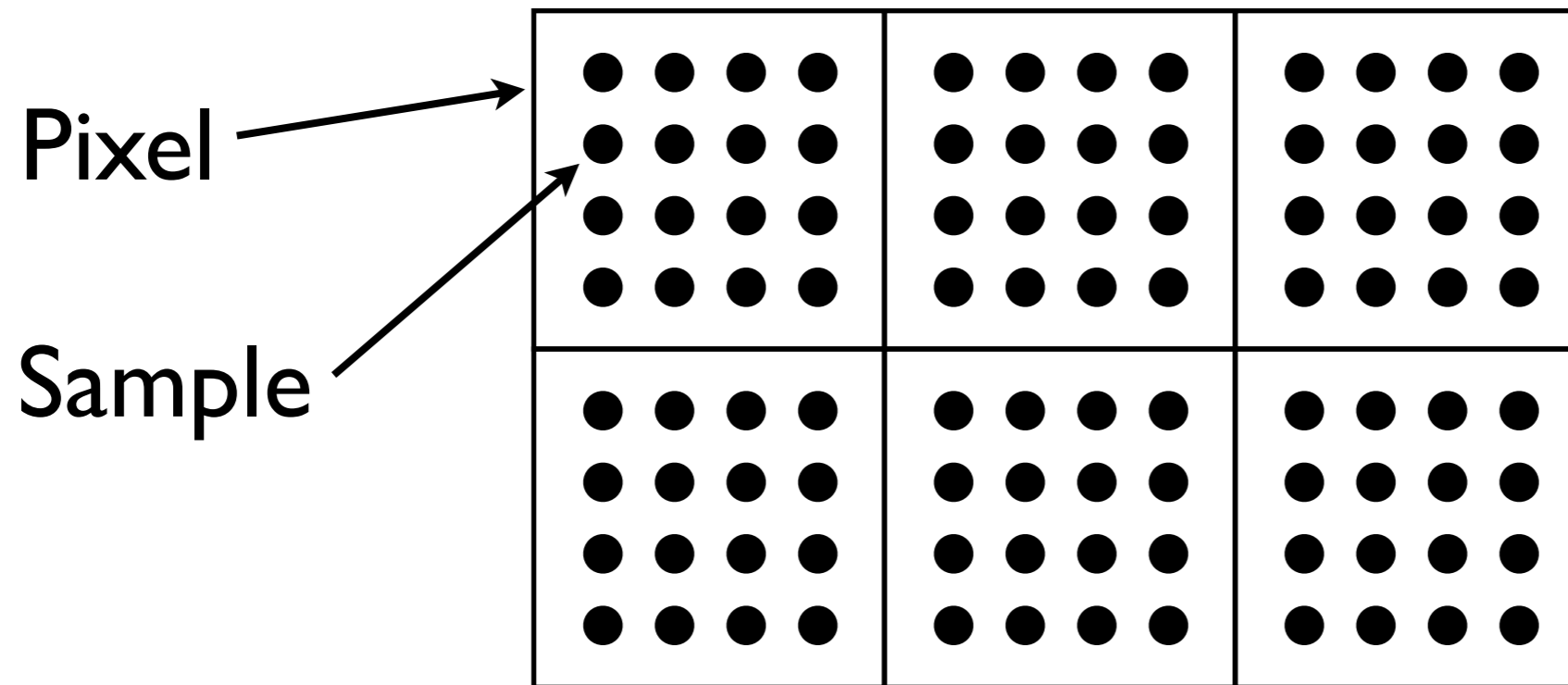


Images courtesy  
Brian Curless CSE557



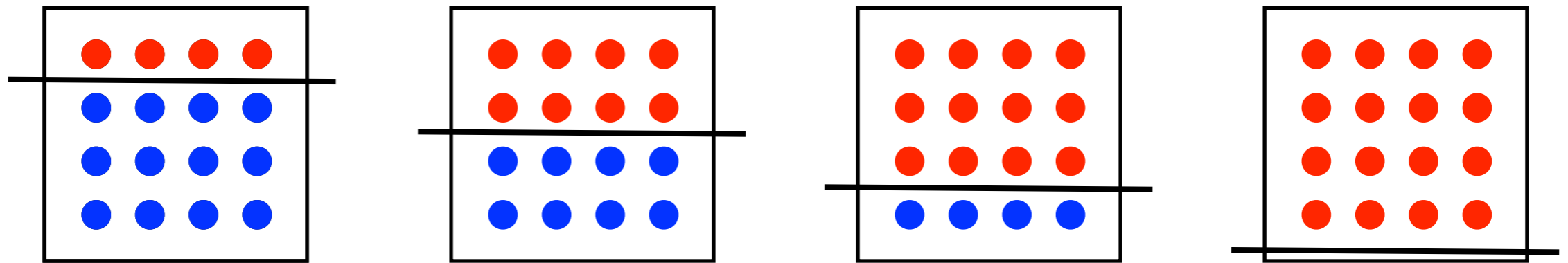
- So instead we sample in the center of the pixel
- To antialias we increase the number of samples in the pixel

# Supersampling



- Super Sampling AntiAliasing (SSAA)
- Render at 4x resolution in X and Y
- Average 4x4 pixels to get final color

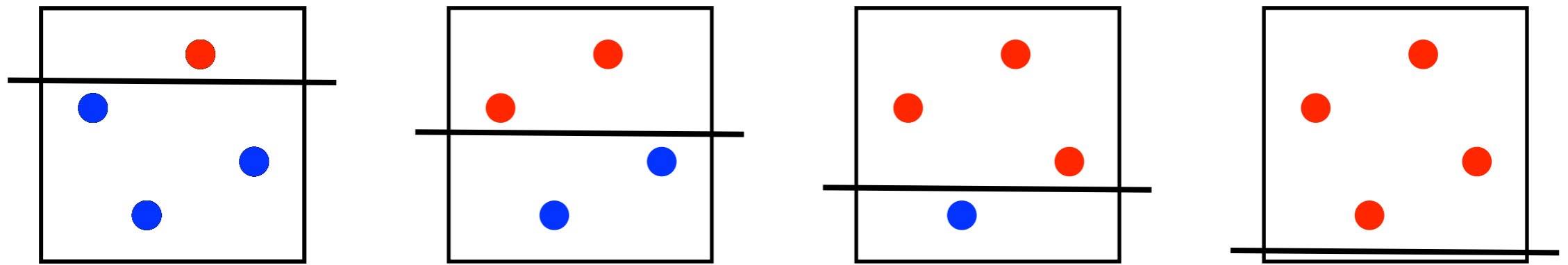
# Sampling Patterns



Only 4 different shades of red for 16 samples!

- Grid (Uniform Sampling)
  - Works well on diagonal lines
  - Not efficient on near horizontal and vertical lines

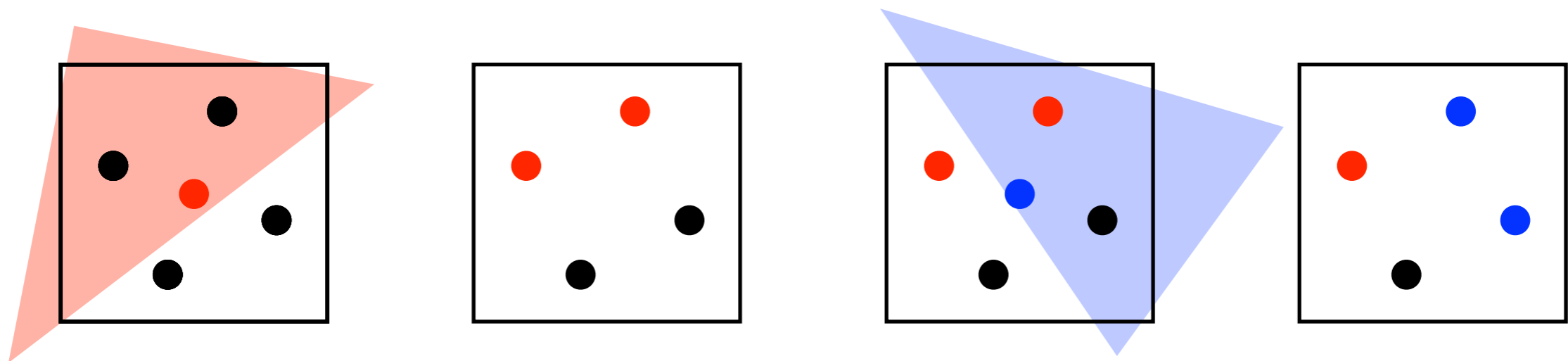
# Sampling Patterns



Now 4 different shades of red for 4 samples!

- Rotated Grid Super sampling (RGSS)
  - Rotate a grid to improve horizontal and vertical
  - Diagonal lines not as well covered, but average is much better
  - Important feature is no two samples on same horizontal or vertical line (called n-rooks pattern)

# Multisample AA (MSAA)



- Computing the color using a shader at every sample is expensive
- So only compute color at one sample, but compute depth at all samples
- Write color to every sample that passes the Z test
  - Store colors at up to 4 positions (RGSS)
- Still have to store the total number of Z/Color samples at each pixel
  - Since some triangles only cover some samples
- Implemented in hardware



# Extending MSAA

- Compute color at more than one sample
- i.e. 16 depth samples, 2 or 4 color samples
- NVIDIA Coverage Sample AA
- AMD Enhanced Quality AA

# Screen Space (Resolve Pass) AA

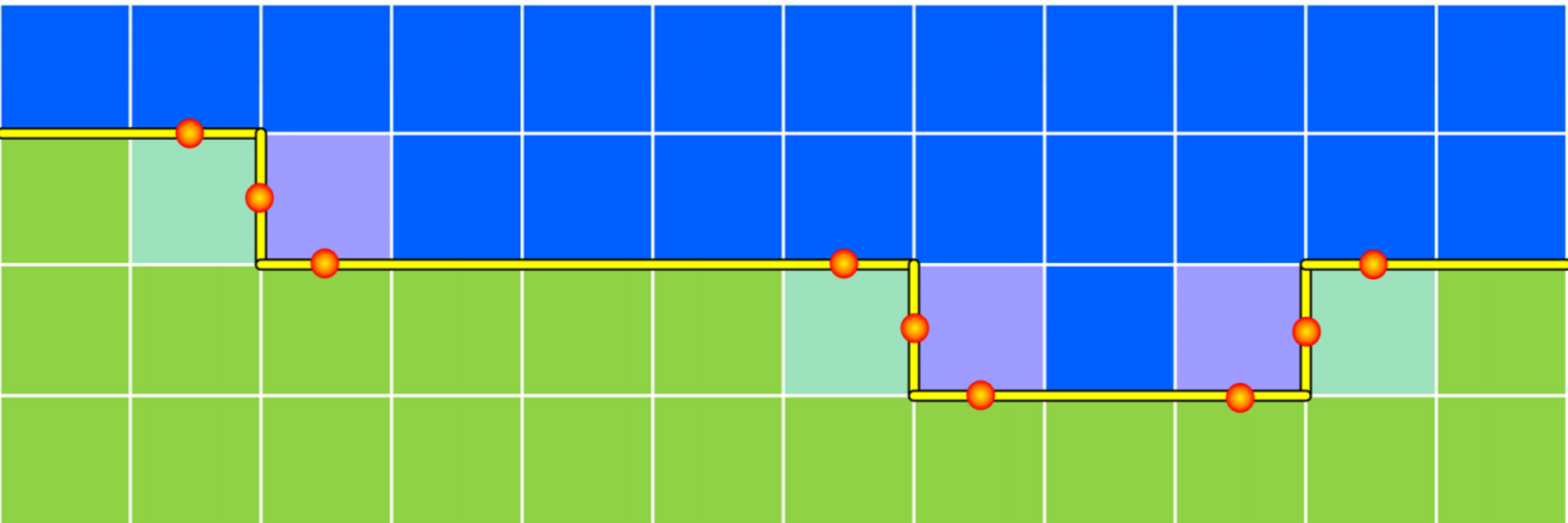
- Post-processing AA
- Purely software/shader based
- Many approaches (at least 27)
  - all use deferred shading color buffer pixels
- The basic idea is :
  - Detect colors that are different
  - Construct a silhouette
  - Filter colors based on constructed silhouette
- For more see the SIGGRAPH 2011 course, “Filtering Approaches for Real-Time Anti-Aliasing”

# Morphological Anti-Aliasing



## MLAA rule # 1 (out of 2)

silhouette segments start/end at edges of pixels at which horizontal and vertical separation lines intersect



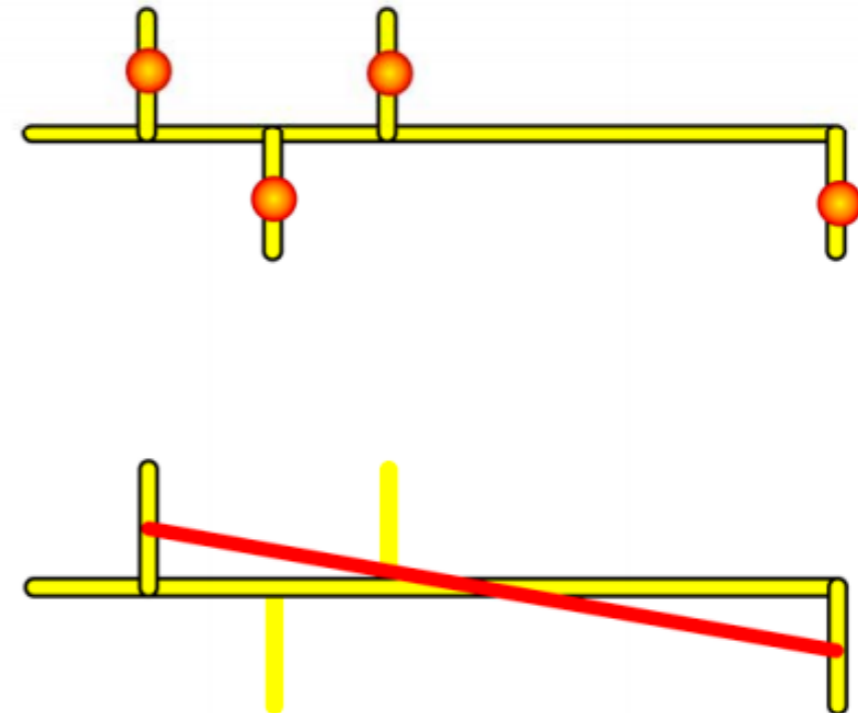
# Morphological Anti-Aliasing



## MLAA rule # 2

for each separation line

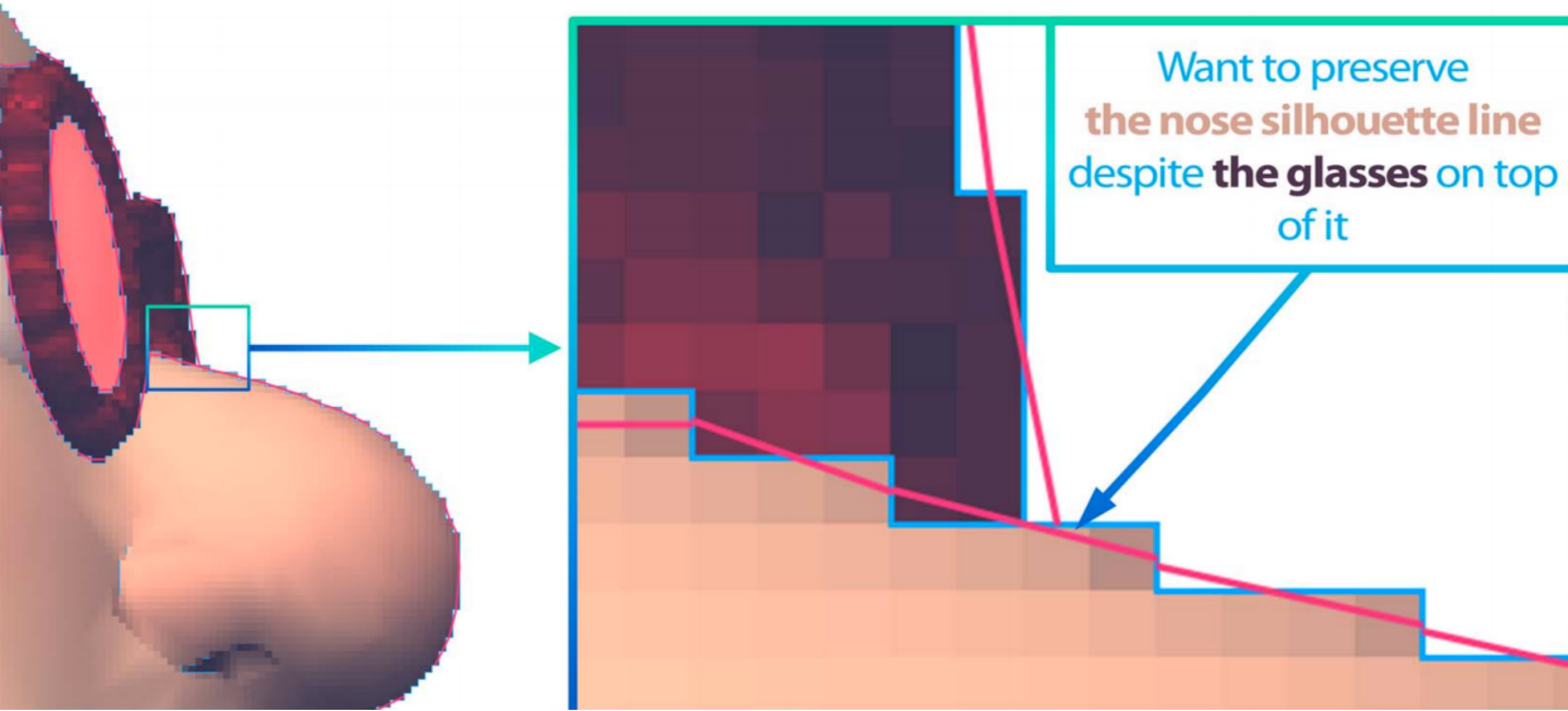
- look at all start/end points on adjacent orthogonal lines
- choose the longest segment



# Morphological Anti-Aliasing



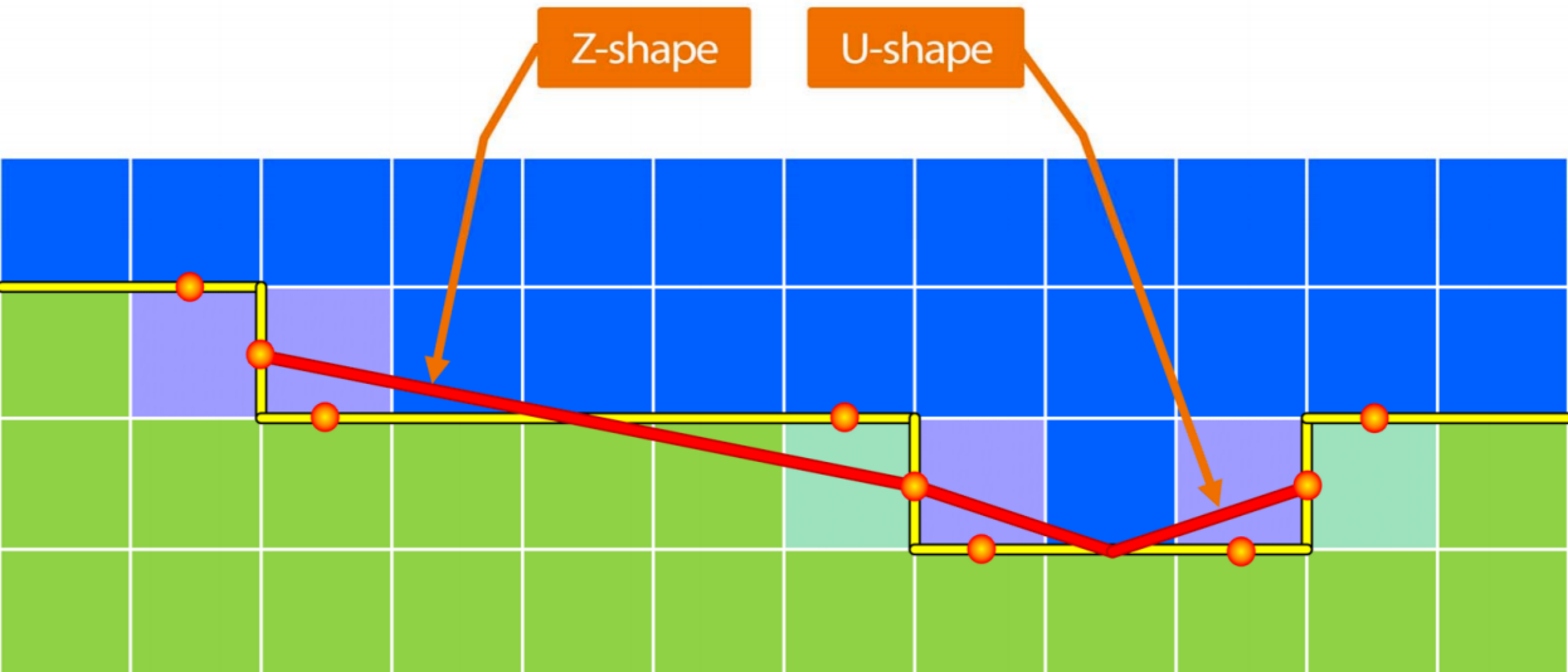
## Rationale: object intersection



# Morphological Anti-Aliasing



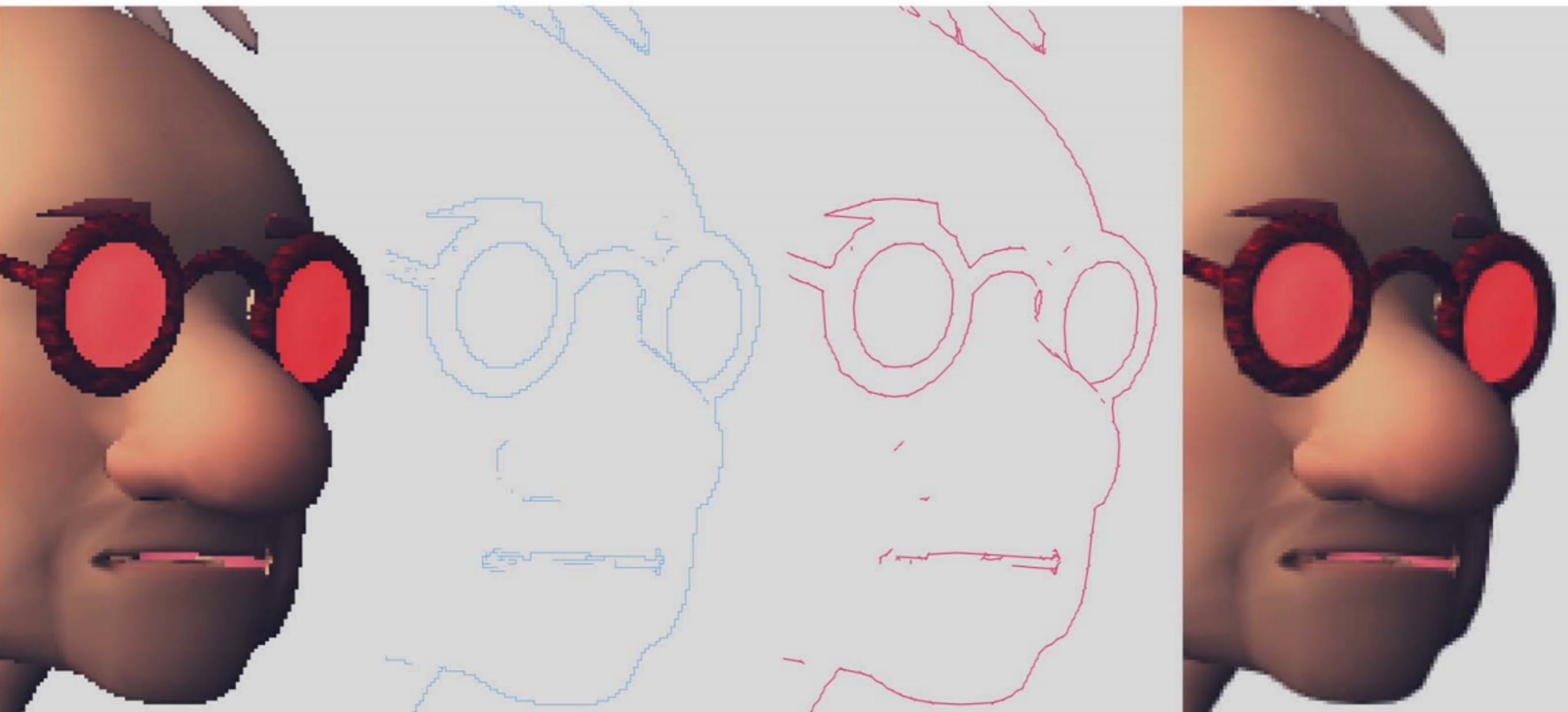
## Two types of shapes



# Morphological Anti-Aliasing



## This is what we will get



Original

Separation lines

Reconstructed silhouette

AA image

# MLAA on PS3

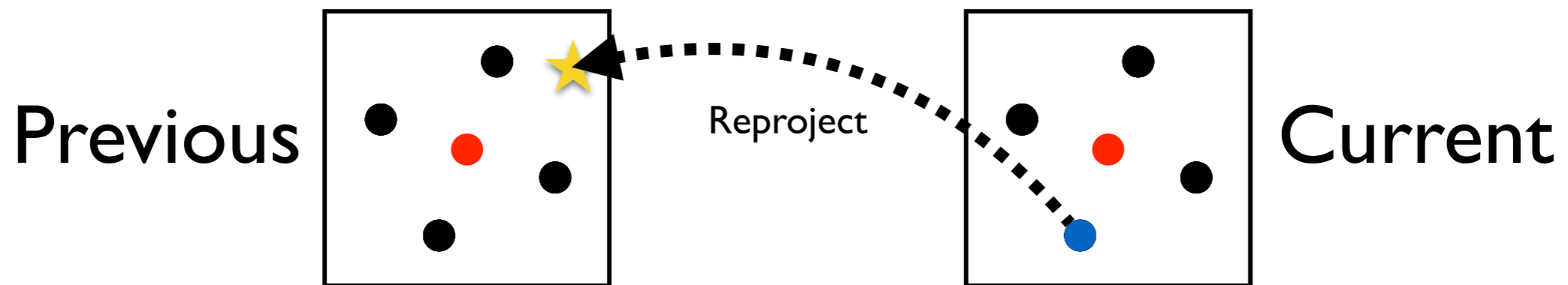




# Directionally Localized Anti-Aliasing (DLAA)



# Temporal Antialiasing (TAA)



- Create new sample in current pixel
- Reproject current pixel into previous frame
  - Use motion vectors
  - Sample history buffer (previous frame's pixels)
  - Validate history sample for changes in occlusion, etc
- Blend history color with new sample

More info : A Survey of Temporal Antialiasing Techniques, Yang, Liu, Salvi, 2020

<http://behindthepixels.io/assets/files/TemporalAA.pdf> Presentation : [https://youtu.be/Ya8xgT0\\_SpM](https://youtu.be/Ya8xgT0_SpM)

# AI/DL/Neural Network

## AA

- Train a NN with high quality images from games
- Use low-res image, G-Buffer, motion vectors, etc as input to NN
- Also used for upscaling
- Nvidia Deep Learning Super Sampling (DLSS)
- AMD FidelityFX Super Resolution (FSR)
- Intel Xe Super Sampling (XeSS)