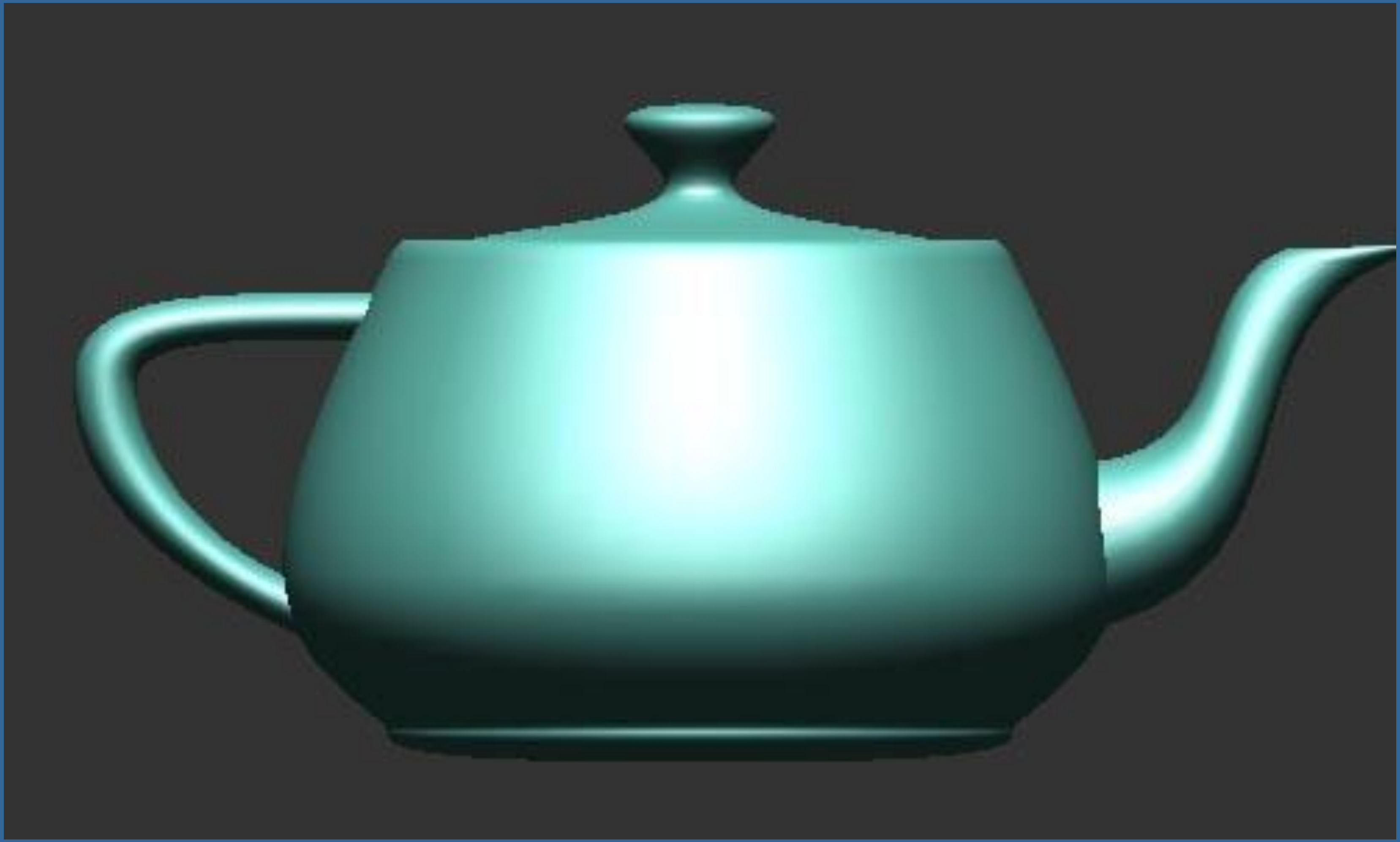
Filtering theory: Battling Aliasing with Antialiasing

Department of Computer Engineering
Chalmers University of Technology

What is aliasing?



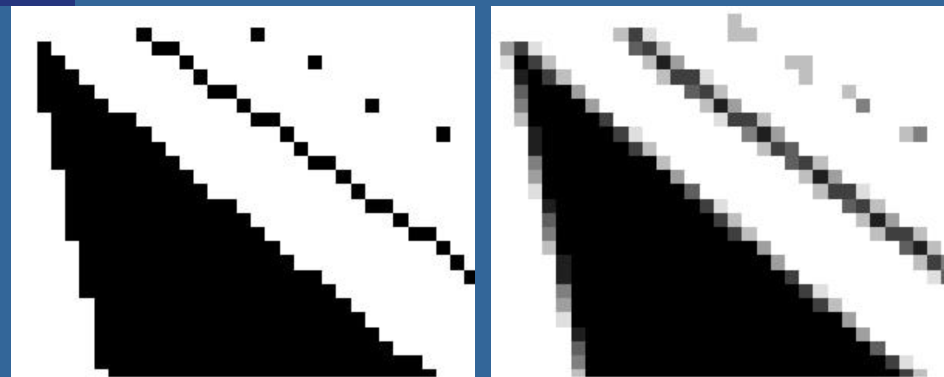
Example

With antialiasing techniques

Without antialiasing

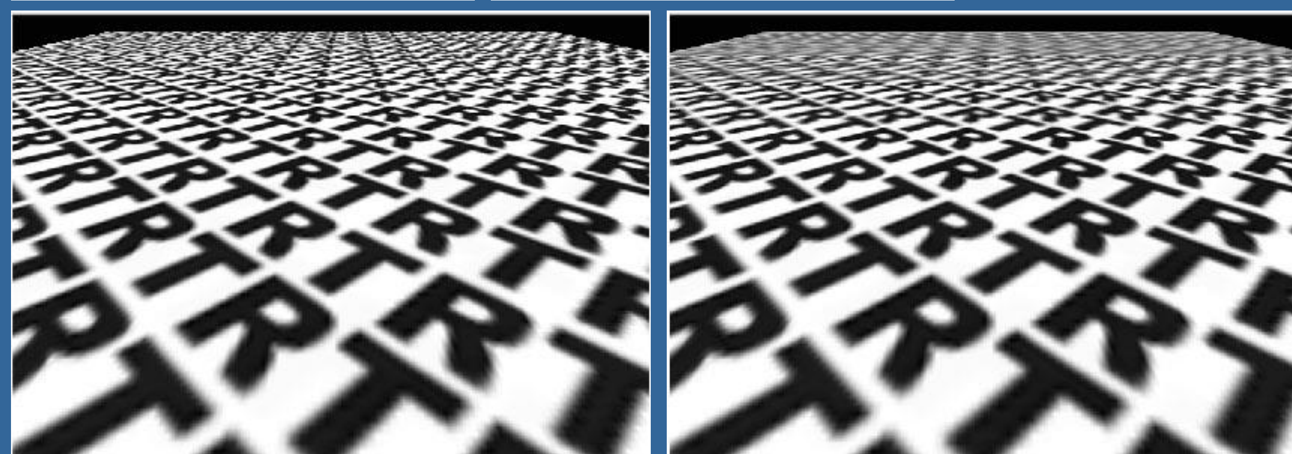
Computer graphics is a SAMPLING & FILTERING process!

- Pixels

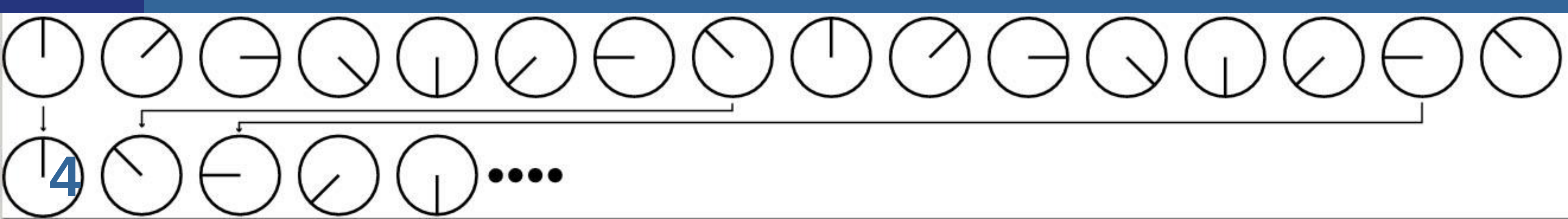


Demo

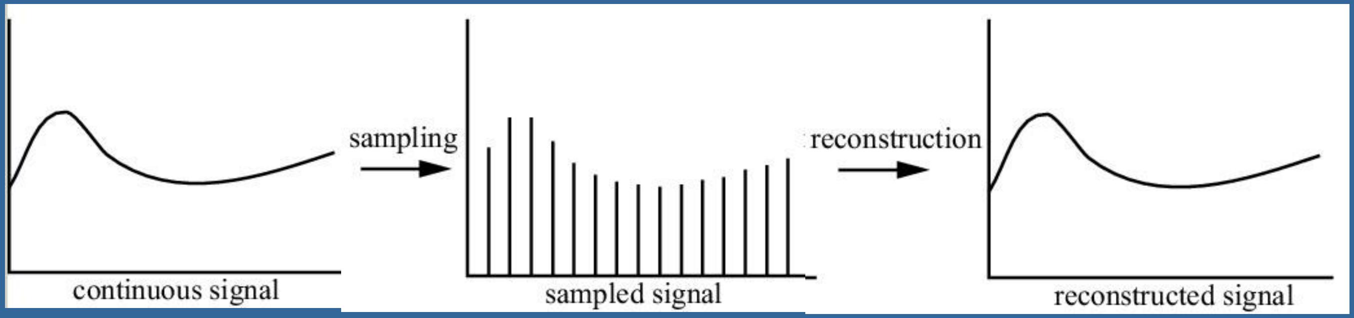
- Texture



- Time



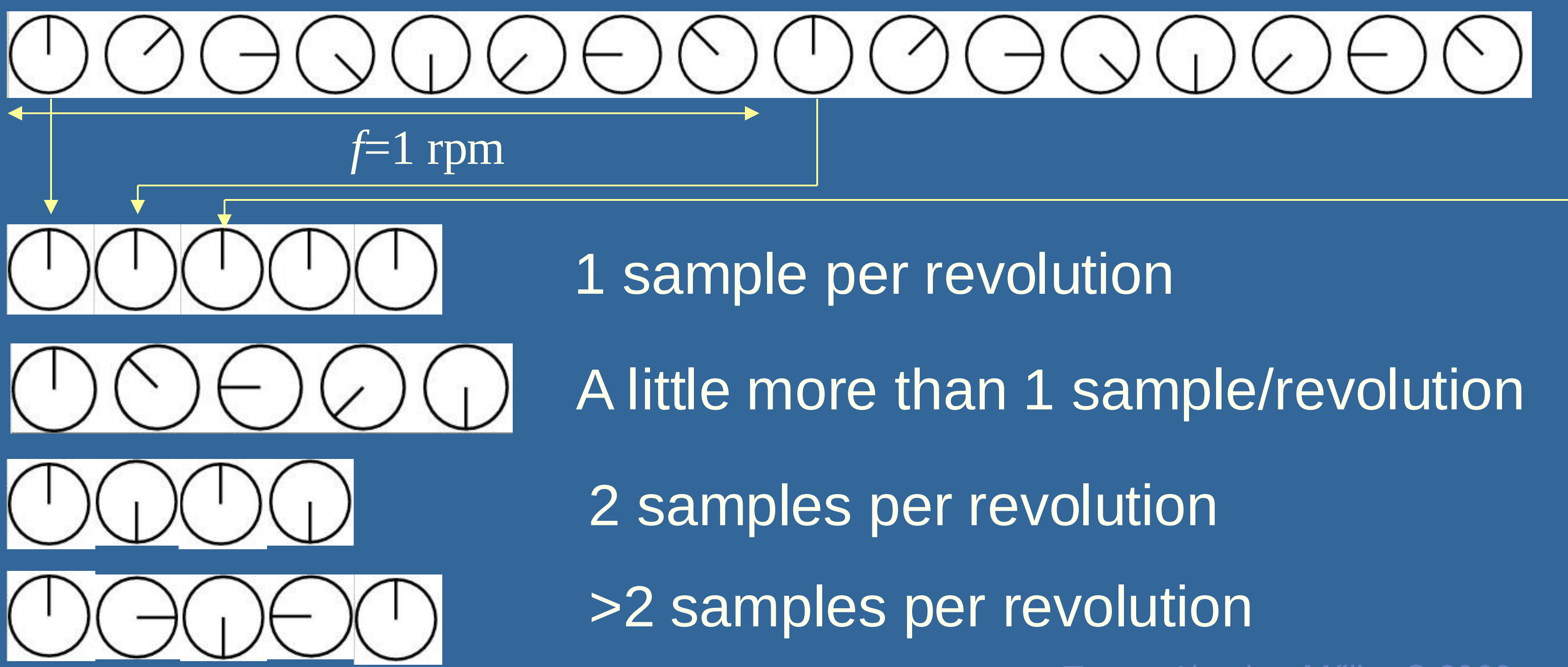
Sampling and reconstruction



- Sampling: from continuous signal to discrete
- Reconstruction recovers the original signal
- Care must be taken to avoid aliasing
- Nyquist theorem: *the sampling frequency should be at least 2 times the max frequency in the signal*
- Often impossible to know max frequency (bandlimited signal), or the max frequency is often infinite…

Sampling theorem

- Nyquist theorem: *the sampling frequency should be at least 2 times the max frequency in the signal*

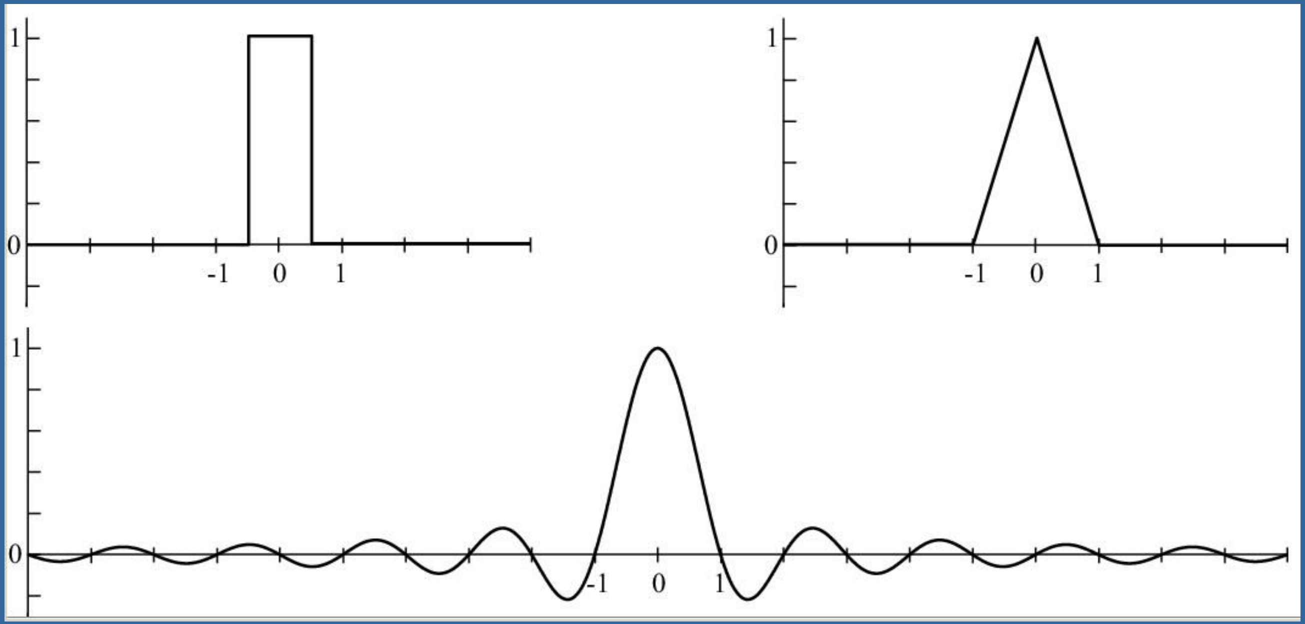


Motion blur

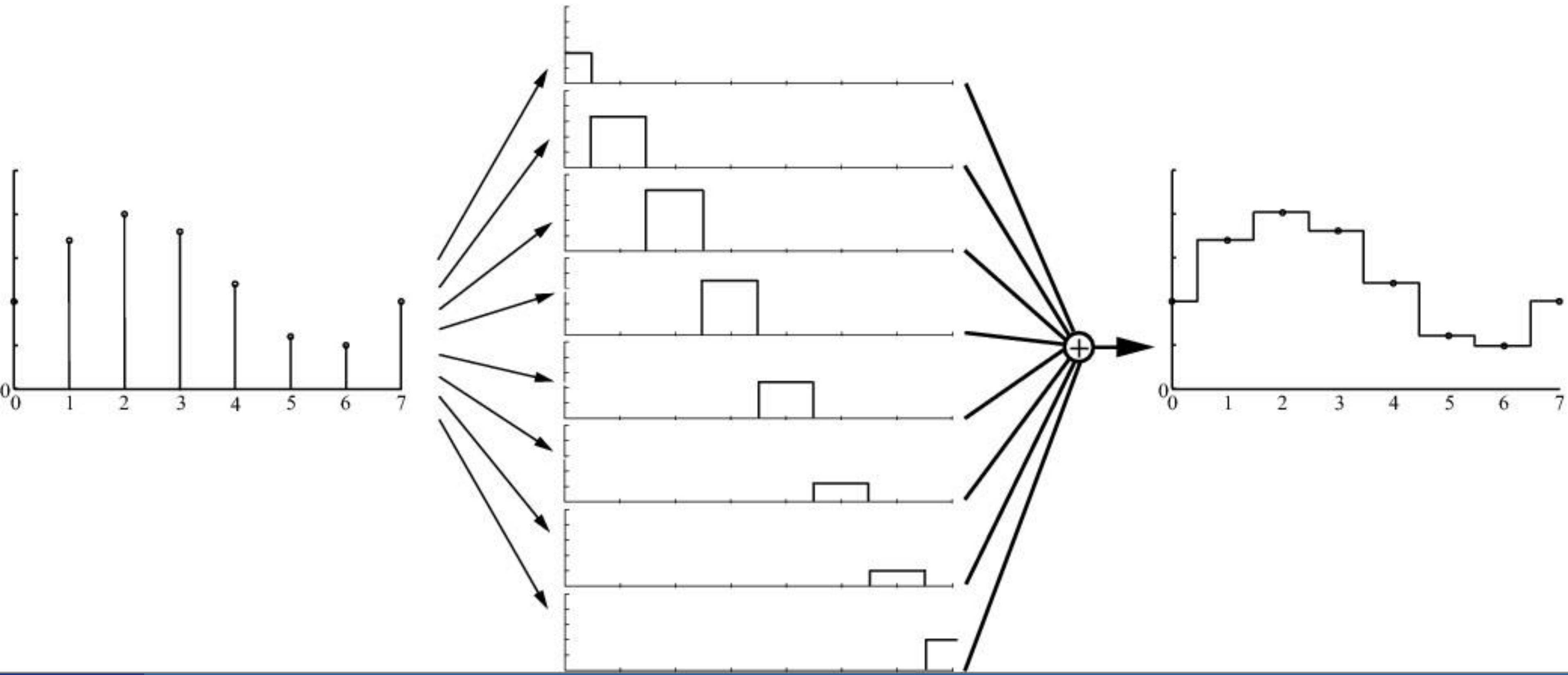
E.g., average several frames over the delta-time step between two frames.

Sampling is simple, now turn to: Reconstruction

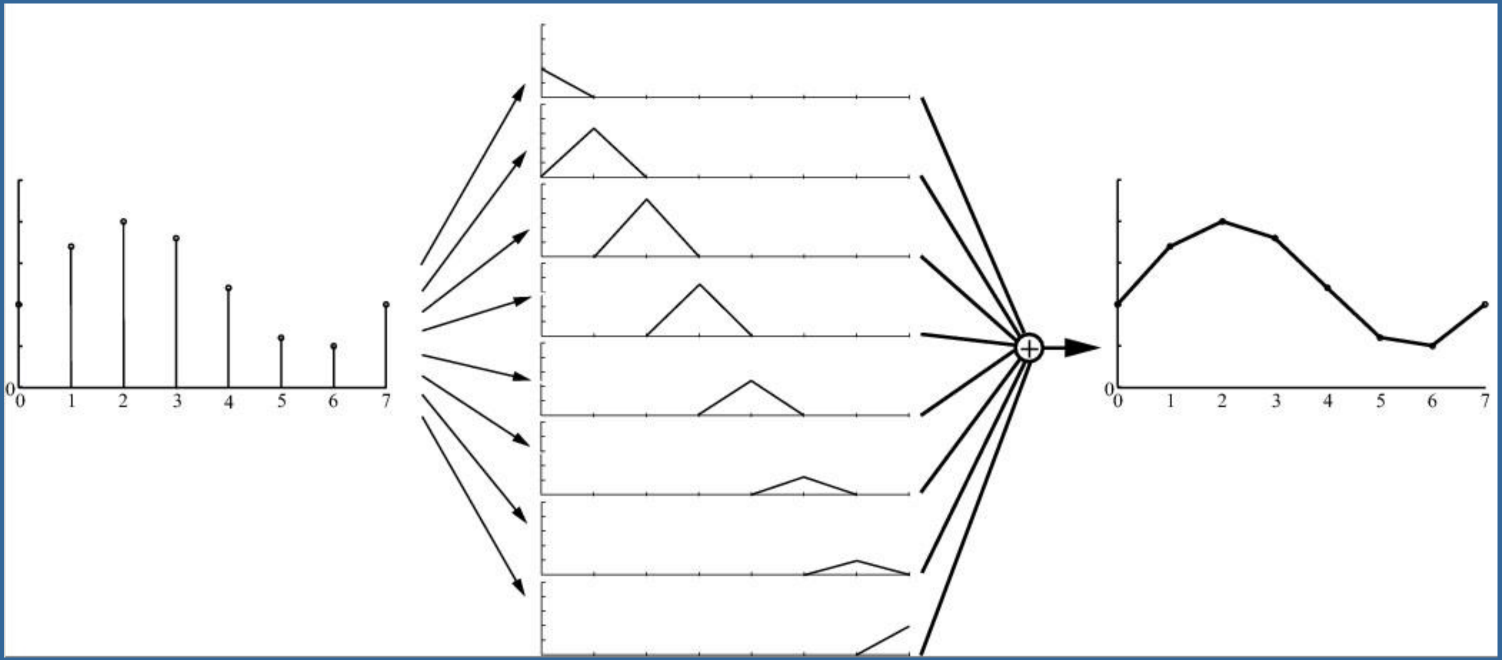
- Assume we have a bandlimited signal (e.g., a texture)
- Use filters for reconstruction



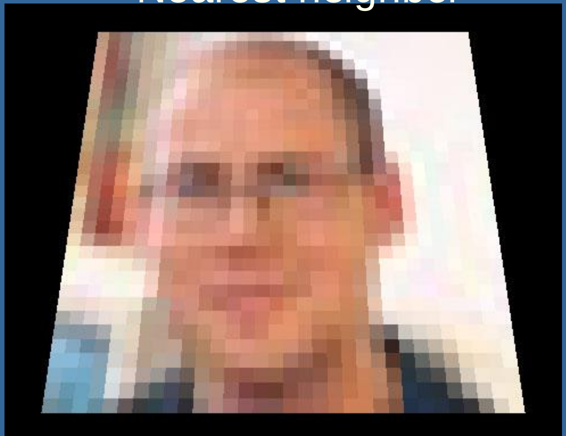
Reconstruction with box filter (nearest neighbor)



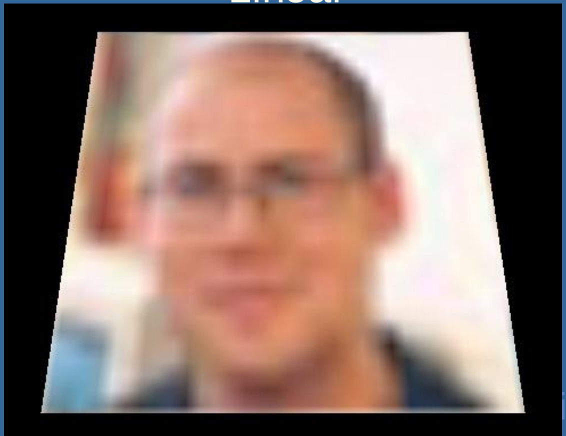
Reconstruction with tent filter



Nearest neighbor



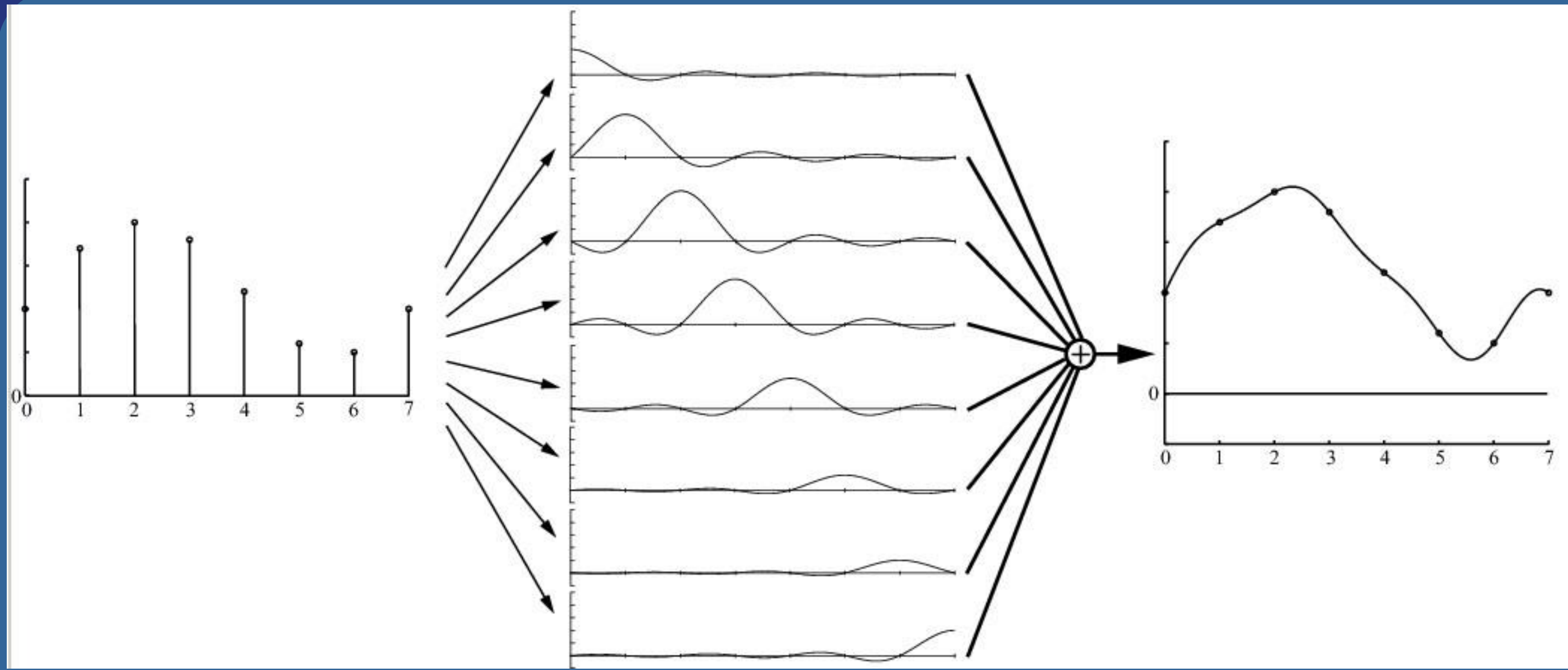
Linear



32x32
texture

$$\mathrm{sinc}(x) \equiv \begin{cases} 1 & \text{for } x = 0 \\ \dfrac{\sin x}{x} & \text{otherwise,} \end{cases}$$

Reconstruction with sinc filter



- In theory, the ideal filter
- Not practical (infinite extension, negative)

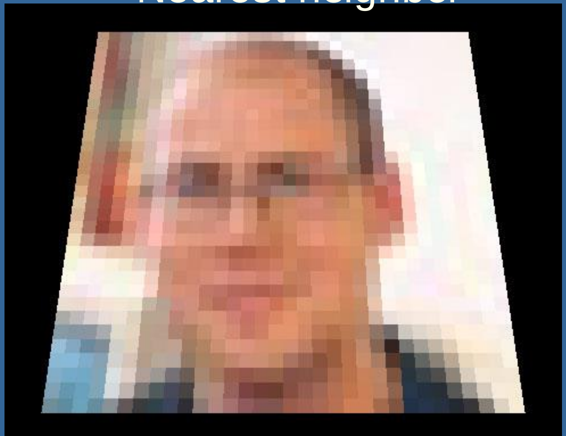
Resampling

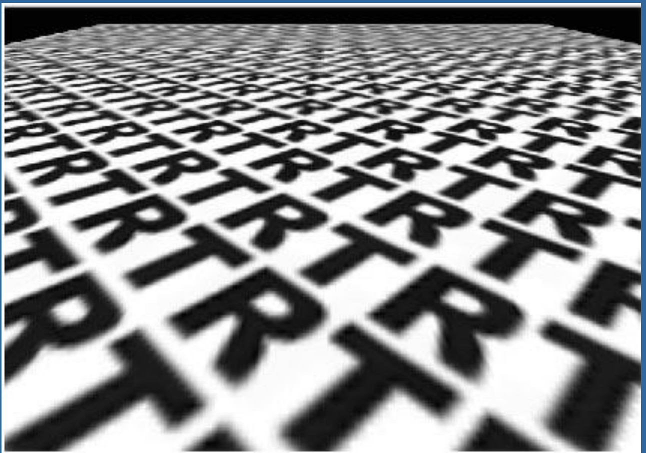
Enlarging or diminishing signals

- Enlarging easy: just use filter (e.g. box or tent) to compute intermediate values.
- For minification, one way is to take the average of the corresponding samples

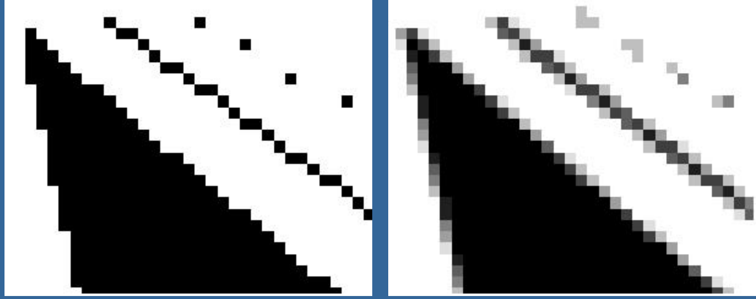
32x32 texture

Nearest neighbor

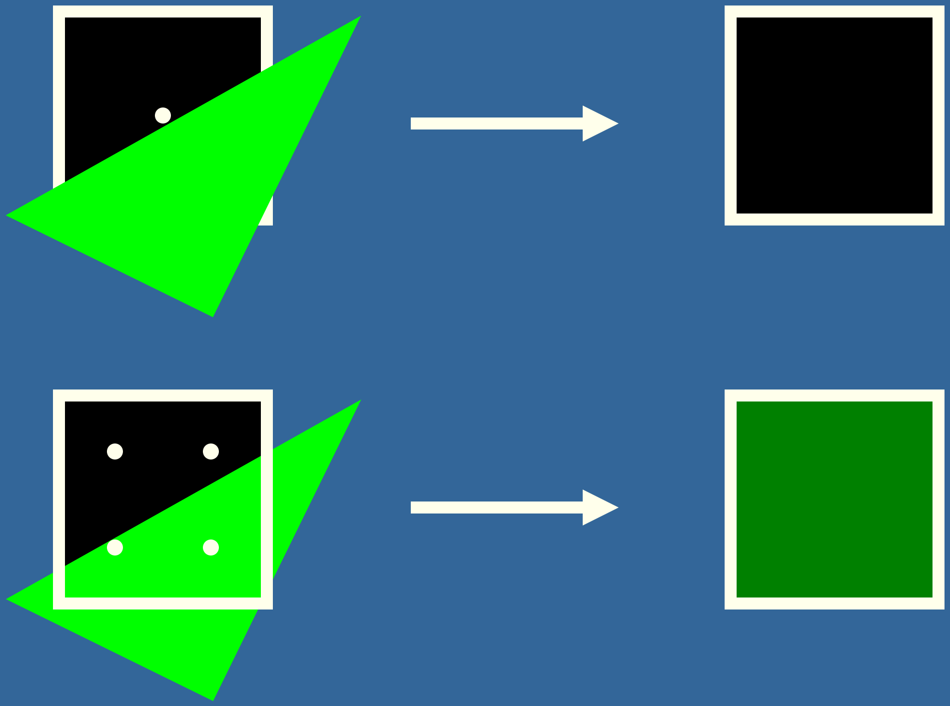




Screen-based Antialiasing



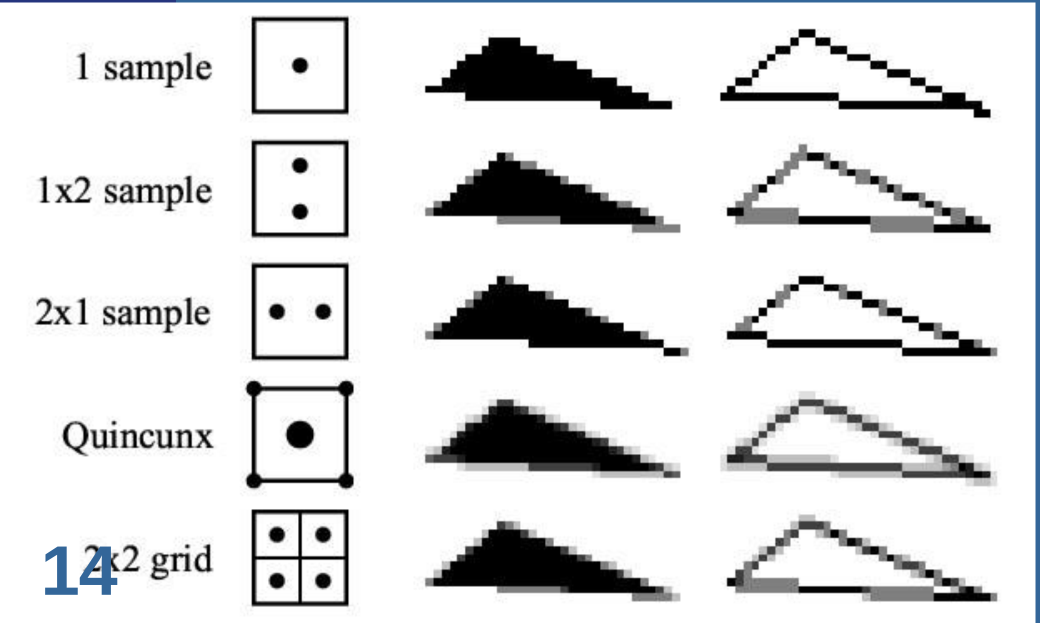
- Hard case: edge has infinite frequency
- Supersampling: use more than one sample per pixel

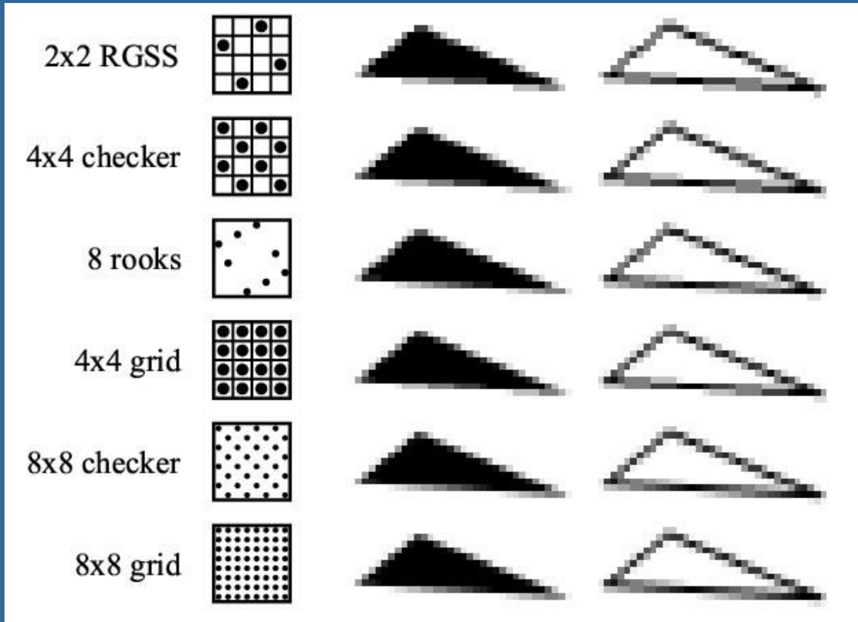


Formula and… examples of different schemes

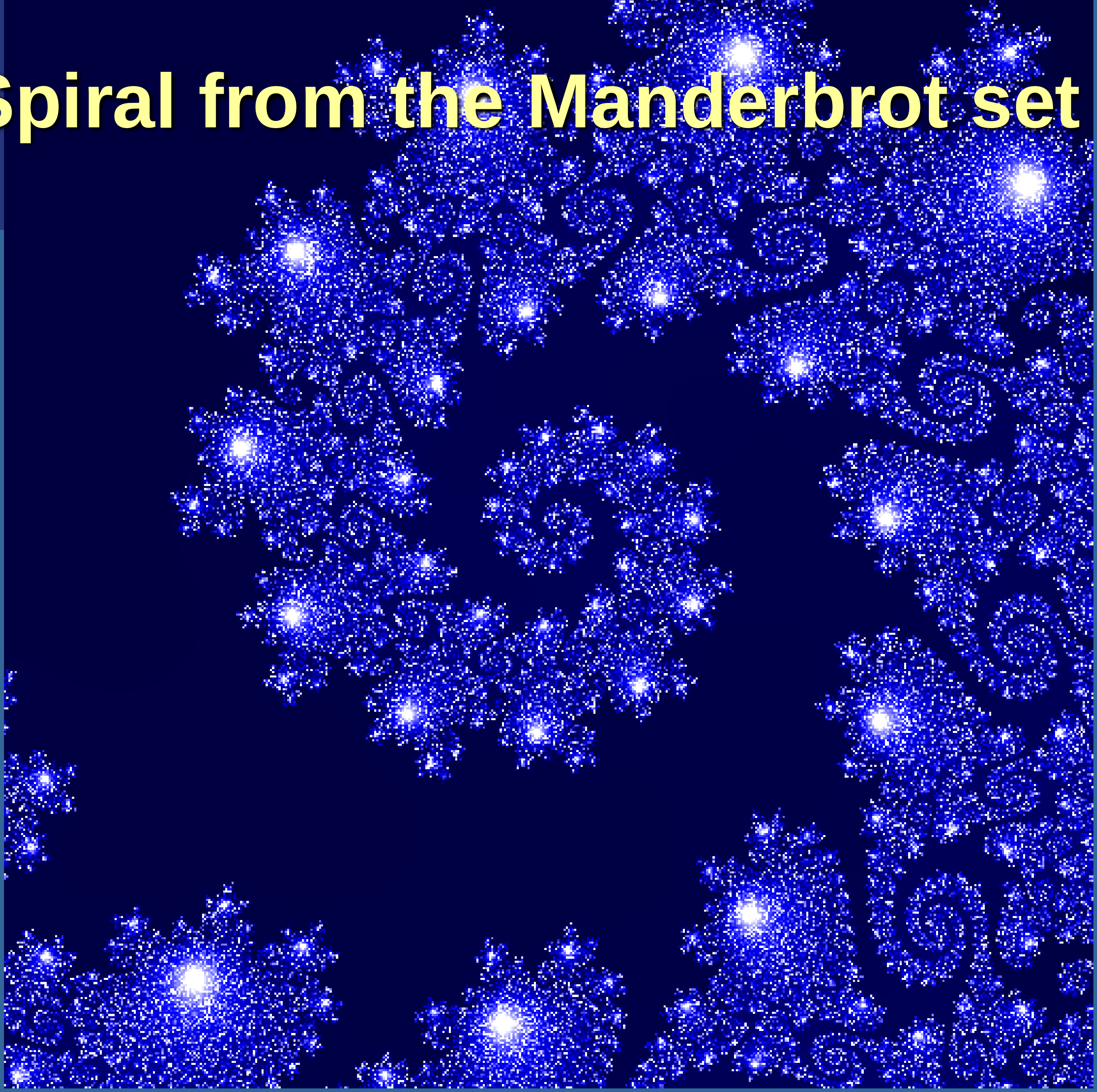
$$\mathbf{p}(x, y) = \sum_{i=1}^{n} w_i \mathbf{c}(i, x, y)$$

- w_i are the weights in [0,1]
- $\mathbf{c}(i,x,y)$ is the color of sample i inside pixel



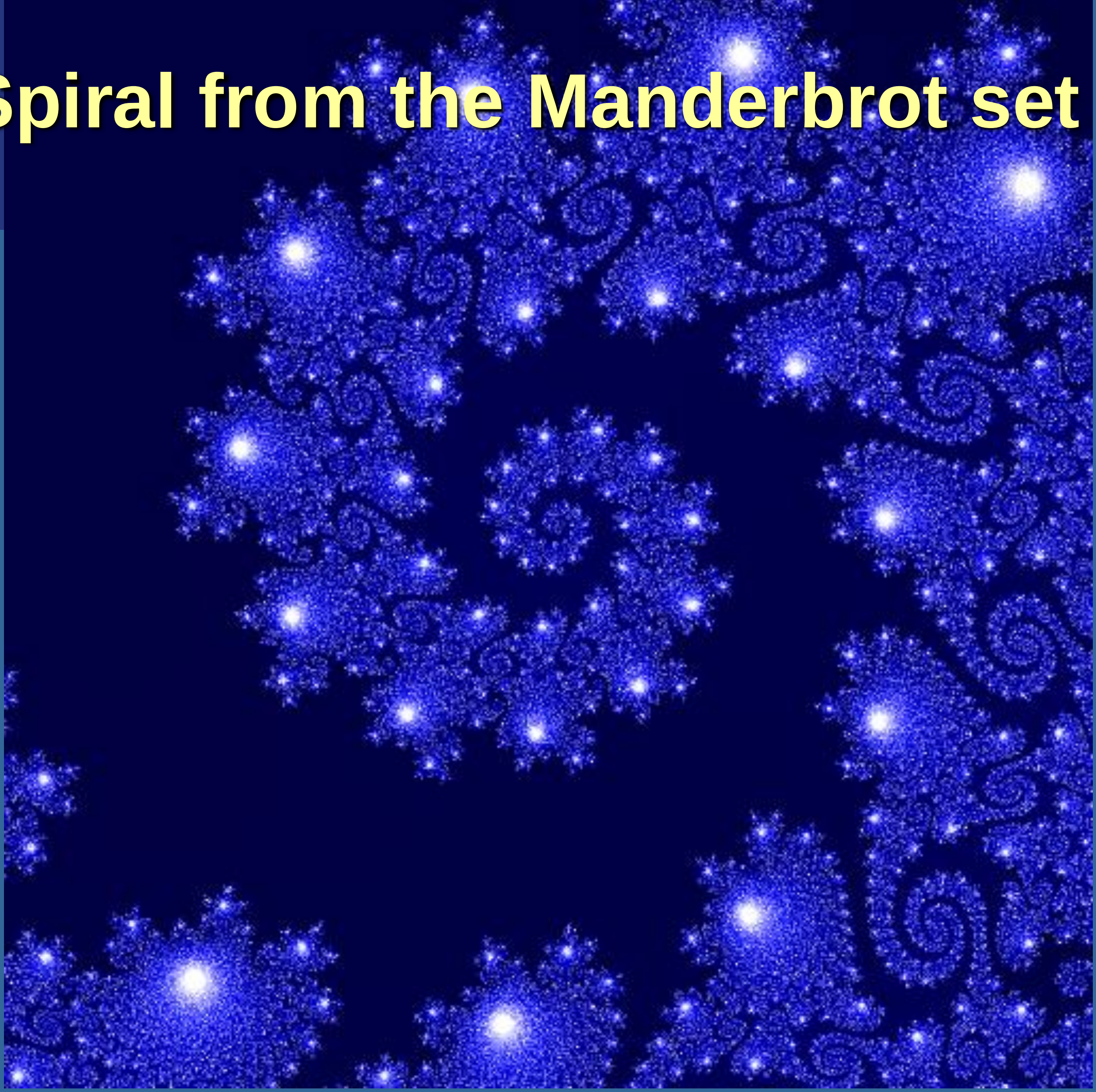


Spiral from the Manderbrot set



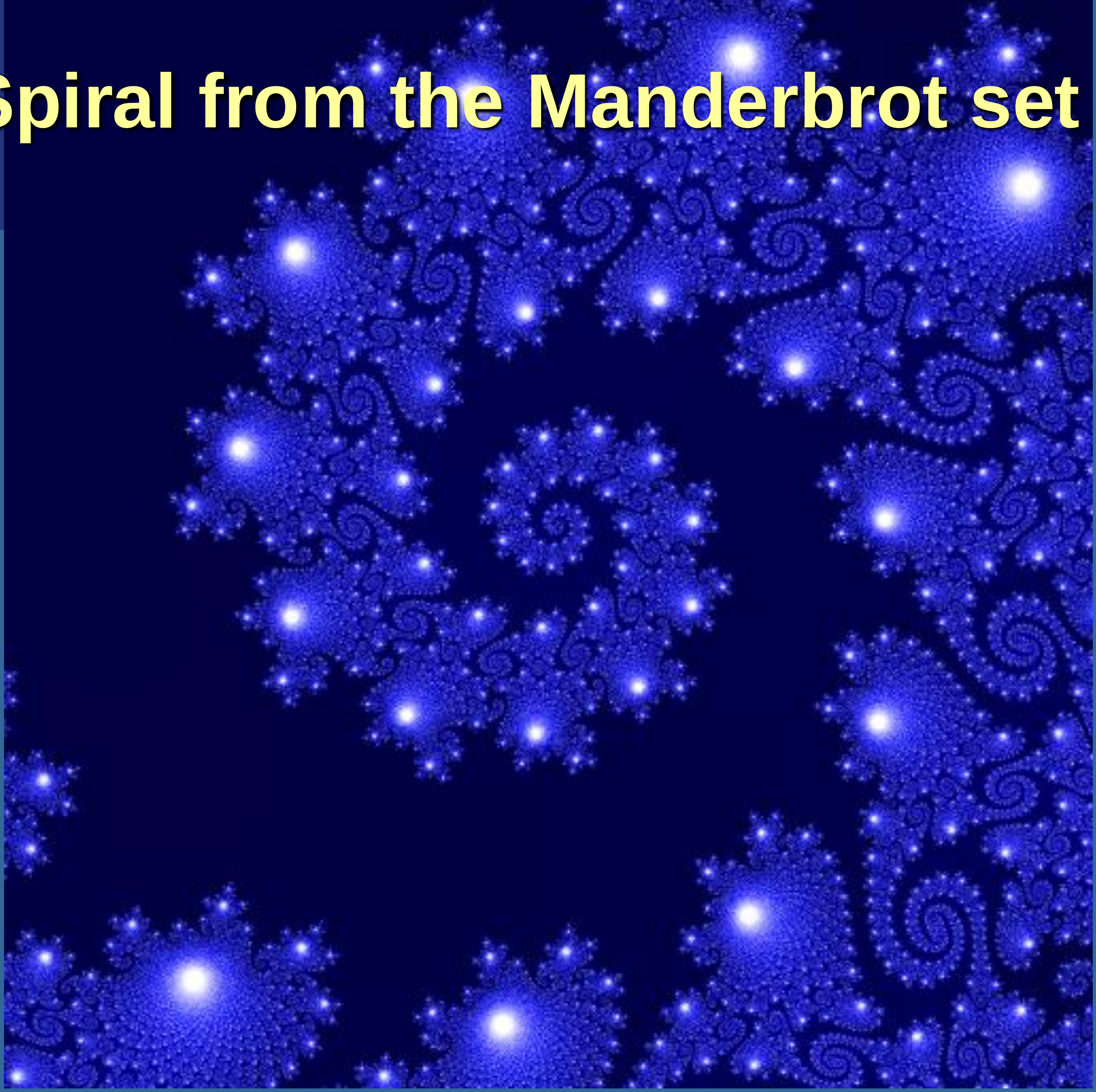
1 samples/pixel

Spiral from the Manderbrot set



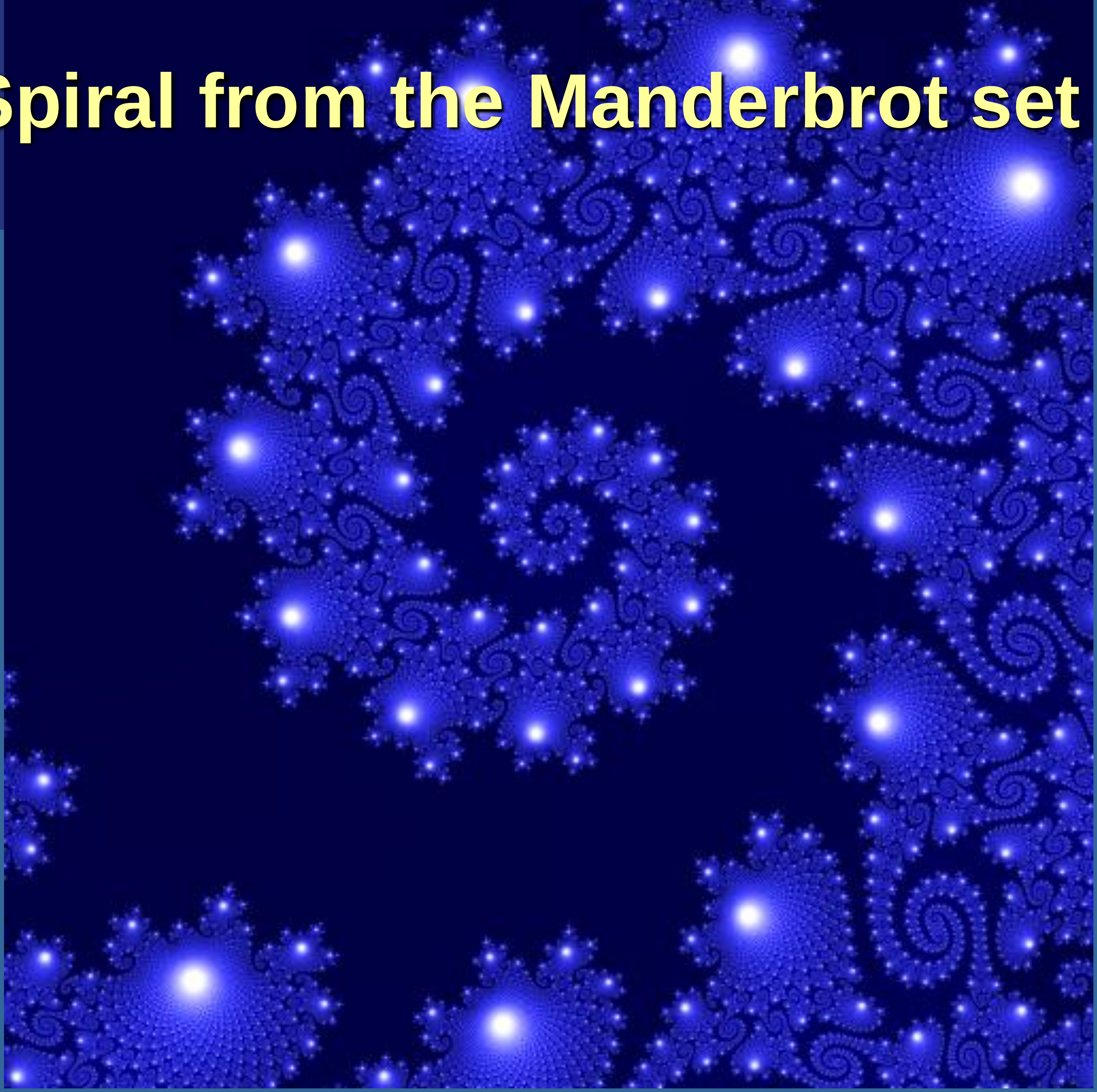
4 samples/pixel

Spiral from the Manderbrot set



25 samples/pixel

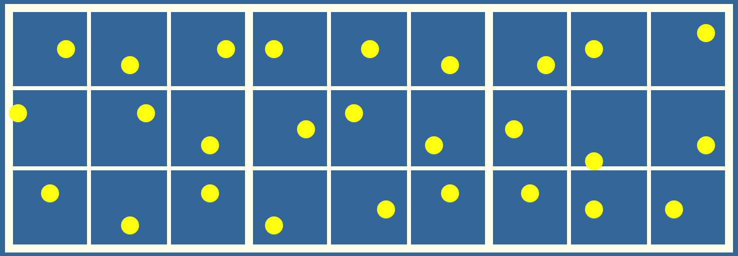
Spiral from the Manderbrot set



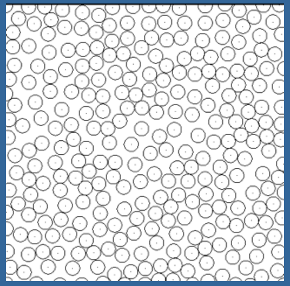
400 samples/pixel

Jittered sampling

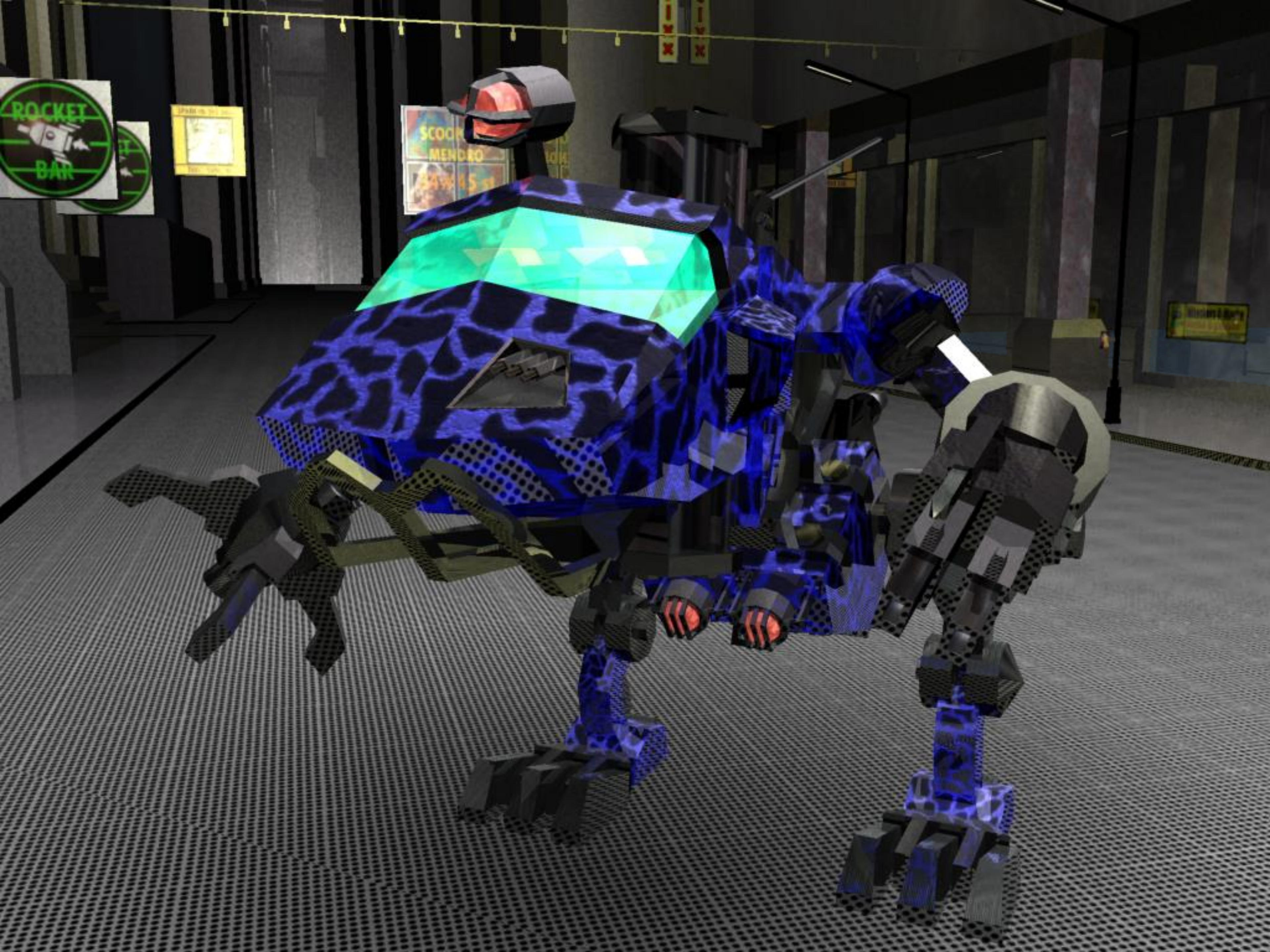
- Regular sampling cannot eliminate aliasing – only reduce it!
- Why?
- Because edges represent infinite frequency
- Jittering replaces aliasing with noise
- Example:



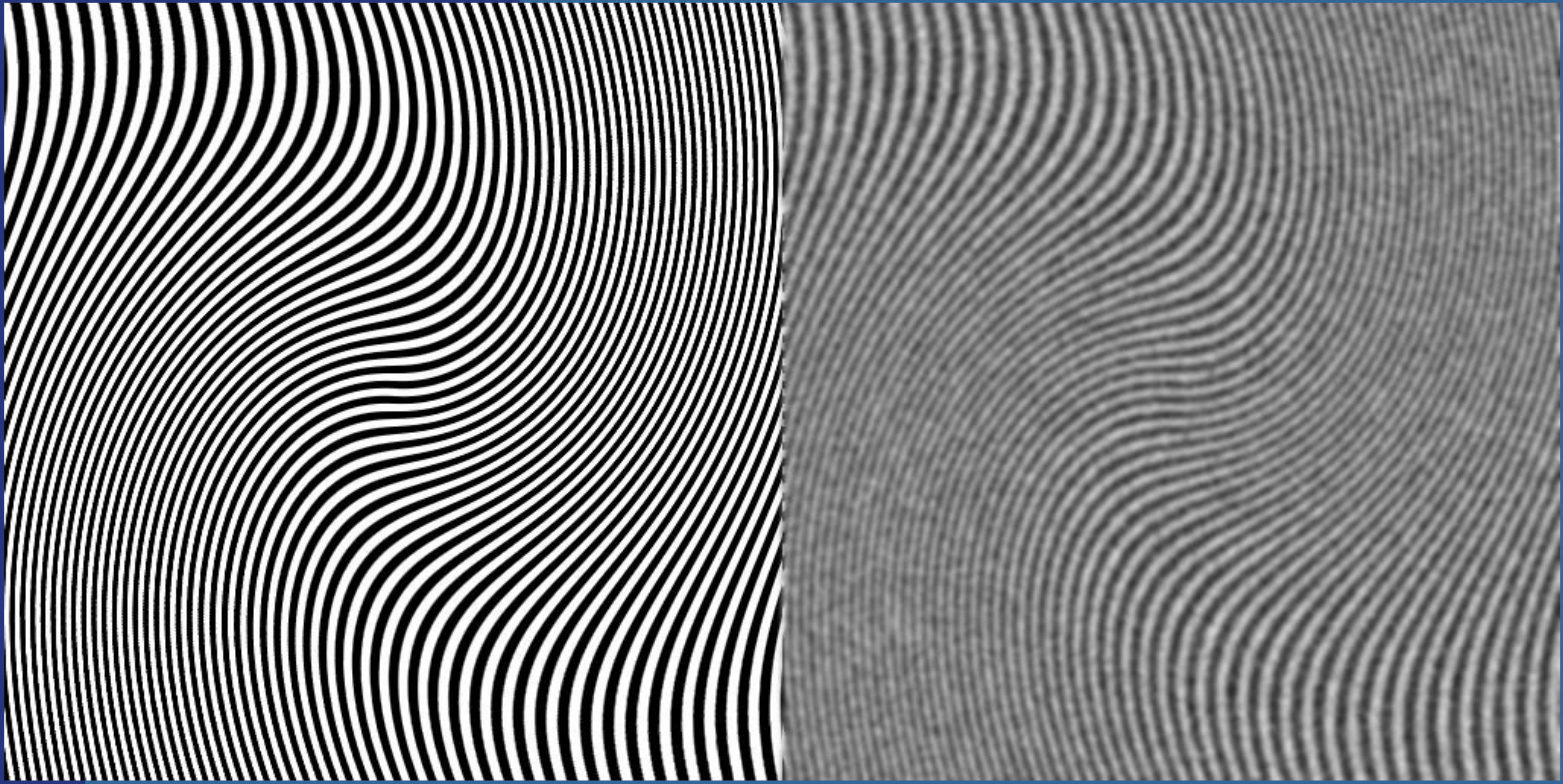
- But still has regularities due to one sample per subcell.
- Better (precomputed) stochastic or pseudo-random patterns (e.g., Poisson disk sampling) but often slower to compute.



Poisson


ROCKET
BAR

Moire example



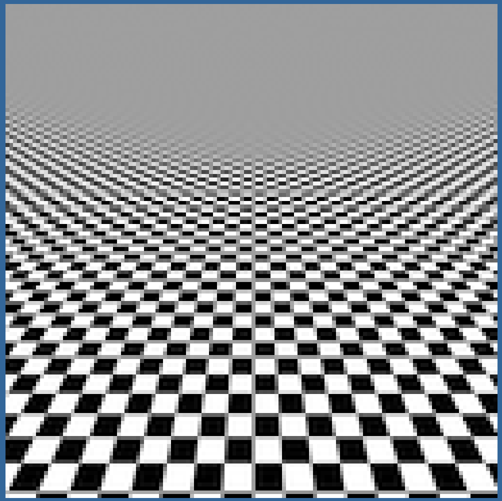
Moire patterns

Noise + gaussian blur

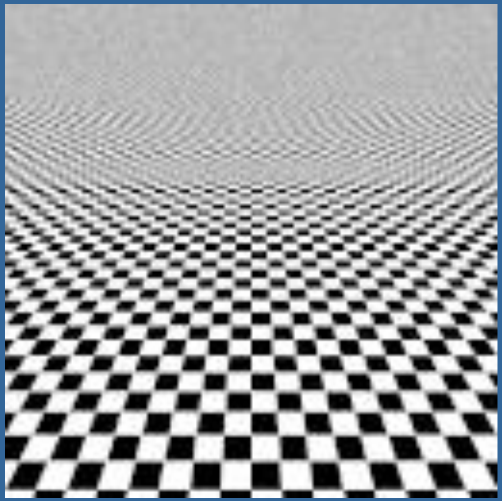
(no moire patterns)

Patterns

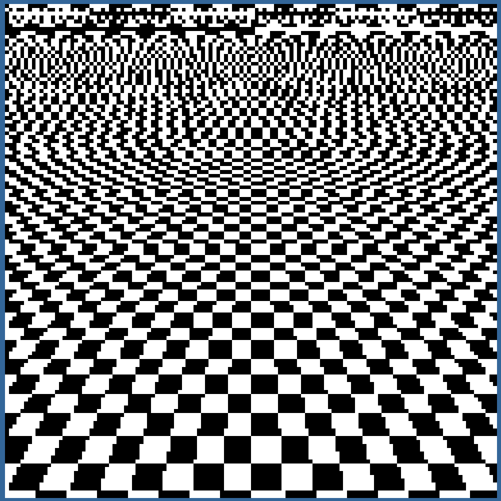
- Checker texture:



Sinc-filter AA



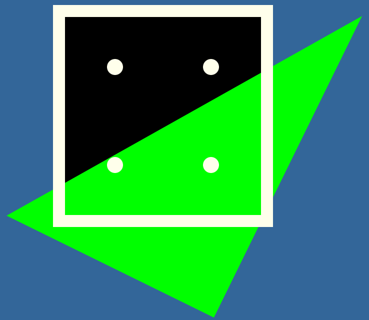
With simpler AA



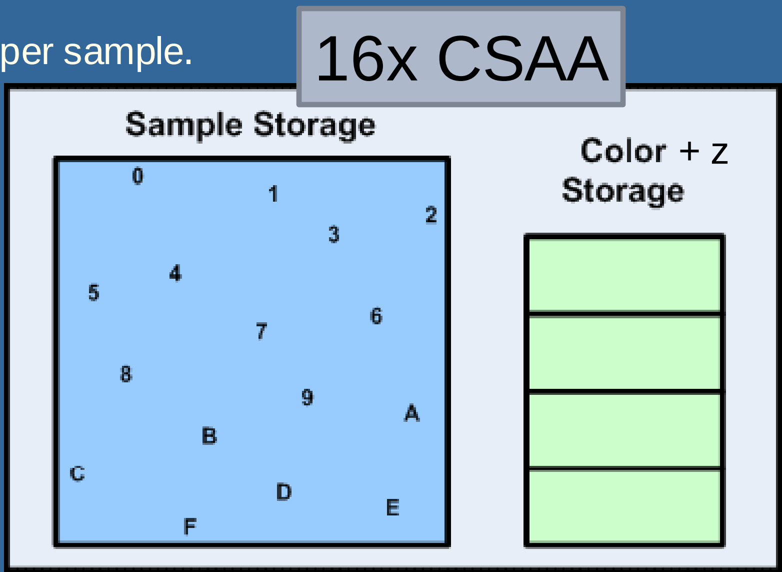
No AA

Point: good AA filtering is important for visual quality

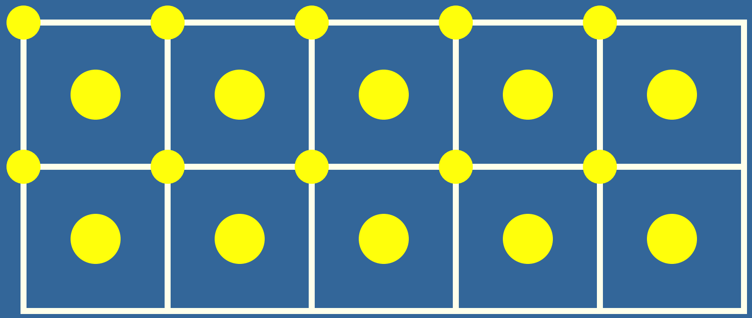
SSAA, MSAA and CSAA



- Super Sampling Anti Aliasing
  - Stores information (color, depth, stencil) for each sample and fragment shader is run for each sample.
  - Corresponds to rendering to an oversized buffer and downfiltering.
- Multi Sampling Anti Aliasing
  - Shares some information between samples. E.g:
    - **Result of Frament shader – Frag. shader is only run once per** rasterized **fragment**.
    - But stores a color per sample and typically also a stencil and depth-value per sample
- Coverage Sampling Anti Aliasing
  - Idea: Don't even store **unique** color and depth per sample. In each subsample, just store index into a per-pixel list of 4-8 colors+depths.
  - I.e., for up to 4 triangles, store their pixel coverage.
  - Fragment shader executed once per rasterized fragment
  - E.g., Each sample holds a 2-bit index into a table (a storage of up to four colors per pixel)

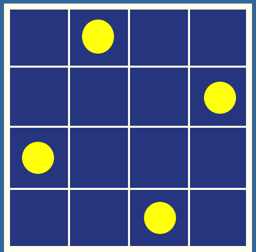


Another multisampling techniqe Quincunx

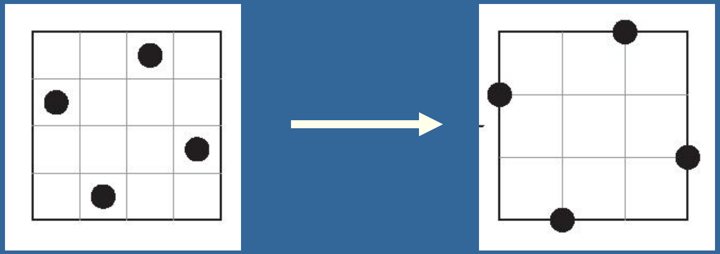


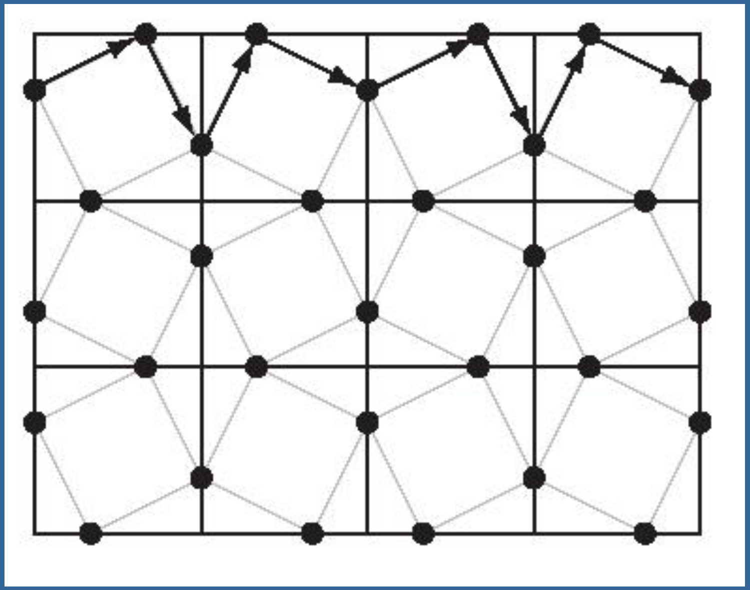
- Generate 2 samples per pixel at the same time
- w_1=0.5, w_2=0.125, w_3=0.125, w_4=0.125, w_5=0.125 (2D tent filter)
- All samples gives the same effect on the image (mid pixel = 0.5, corner pixels = 4*0.125=0.5)
- Was available on NVIDIA GeForce3 and up

Yet another scheme: FLIPQUAD multisampling

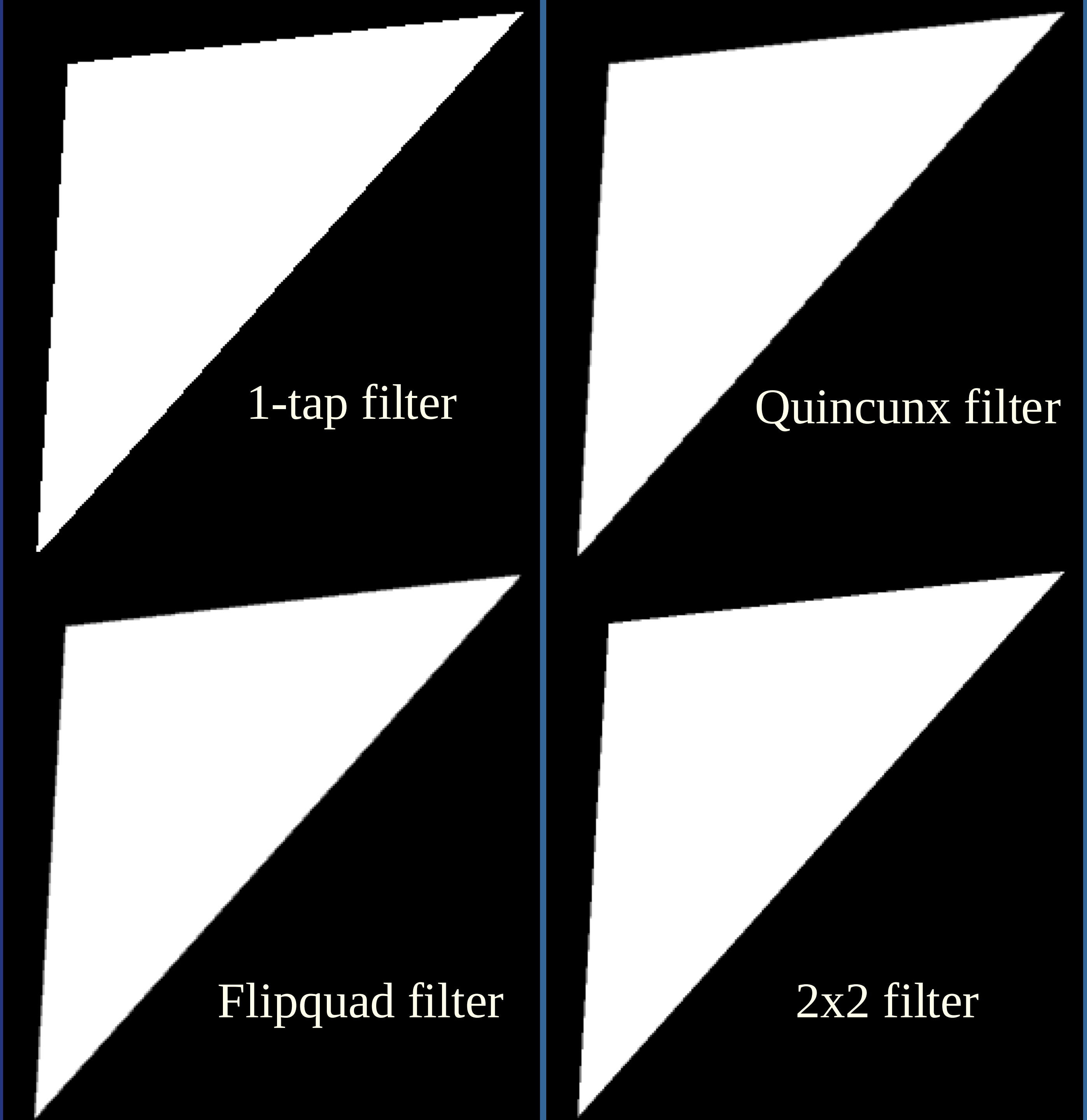


- Recap, RGSS:
  - One sample per row and column
- Combine good stuff from RGSS and Quincunx

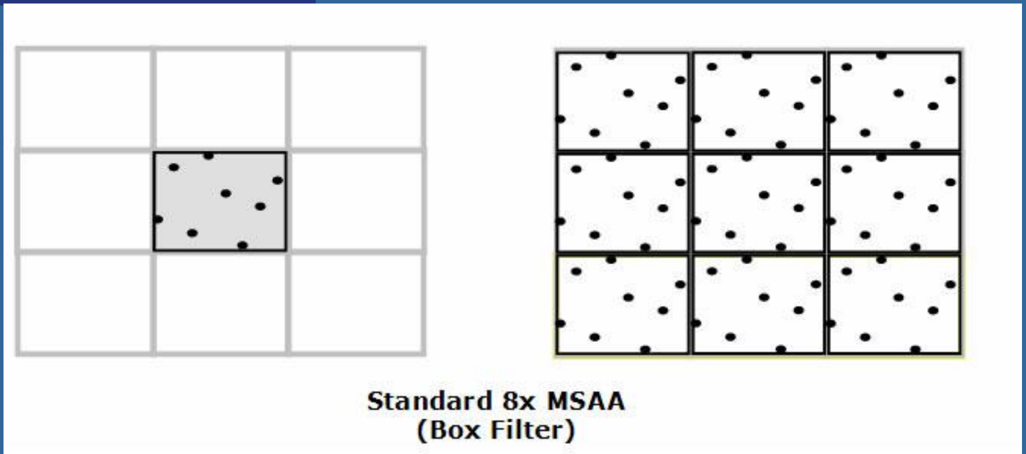


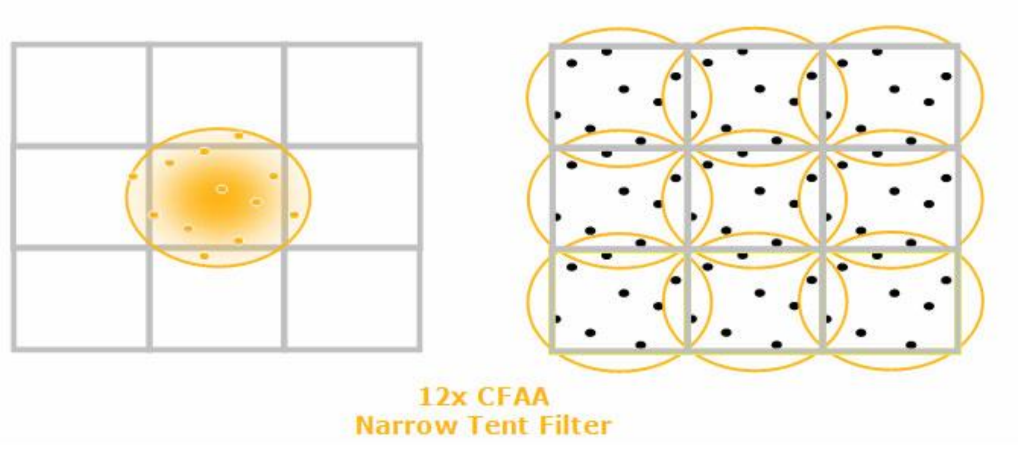


- Weights: 0.25 per sample
- Performs better than Quincunx


1-tap filter
Quincunx filter
Flipquad filter
2x2 filter

ATI Radeon 2900





From www.pcper.com

- Examples of 2 filter modes

Extra…

- Full screen anti aliasing (FSAA) – means super-/multi-/coverage- sampling the full screen. Default today.
- FXAA – fast approximate antialiasing, RTR p: 148. NVIDIA white paper. (2009)
- Subpixel Morphological Anti-Aliasing (SMAA)
  - Like FXAA but takes more samples per pixel along edges
- "Filmic SMAA: Sharp Morphological and Temporal Antialiasing" *Siggraph Advances in Real-Time Rendering in Games*, course notes. (2016)

  Roughly equal to:
  - Edge-detection blur
    + temporal filtering

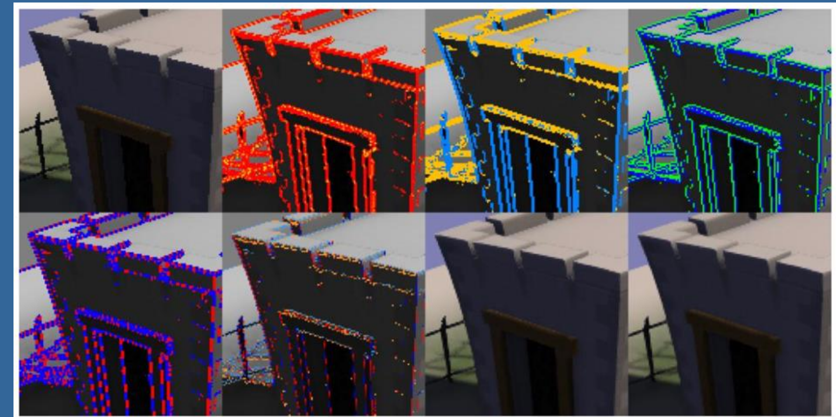
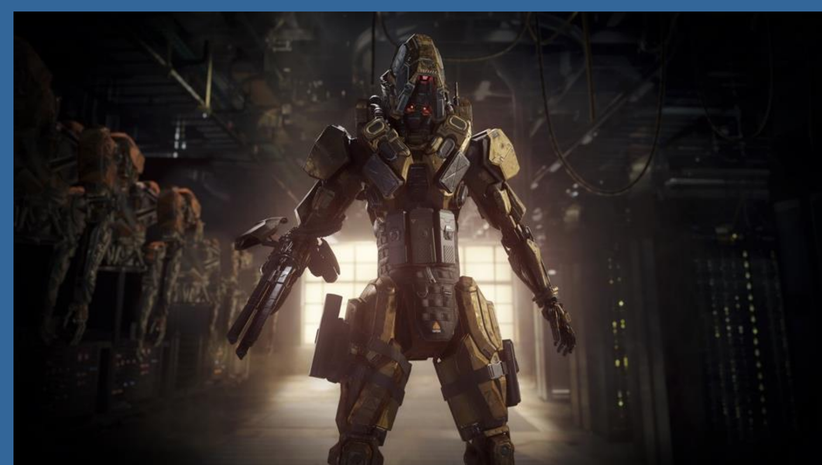


Figure 1: FXAA algorithm from right to left, top to bottom.

Detect the edge directions. Blur each edge orthogonally to its direction.



What is important:

- Aliasing in 3 different areas:
  - Pixels, textures, time
- Filter: box, tent, sinc
- Different sampling schemes
  - Quincunx, Grid, Rotated Grid Super Sampling (RGSS), checker, 8-rooks
- Jittering:
  - 1) How it works. 2) Trades undersampling artifacts for noise (typically prefered by humans)
- Supersampling, multisampling, (coverage sampling)
- Quincunx – pattern and weights
  - Good because costs only 2 samples/pixel on average, but uses 5 samples per pixel

More on filtering theory and practice

- Especially important for pixels and filtering of textures
- More about texturing in next lecture

THE END