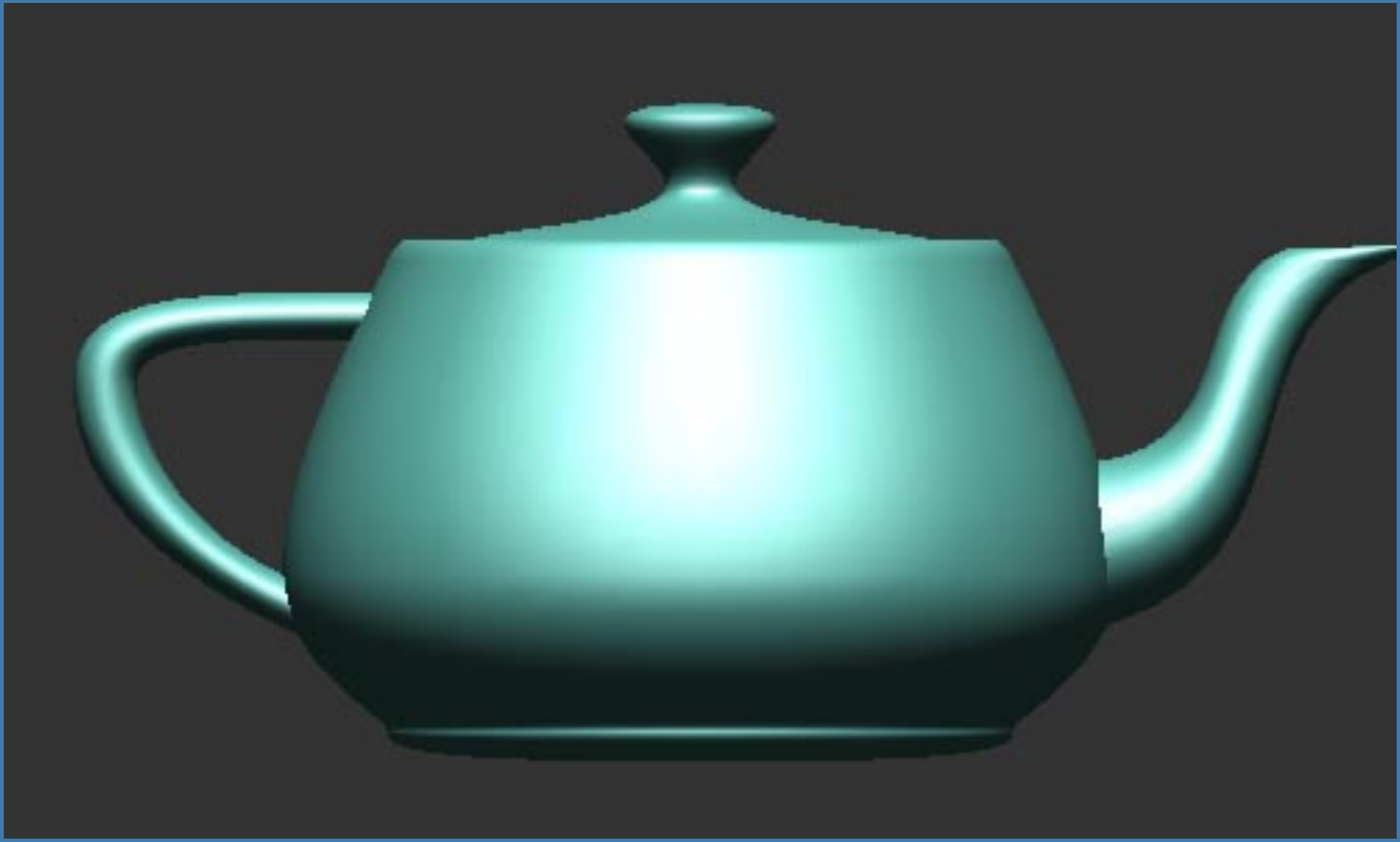
Filtering theory: Battling Aliasing with Antialiasing

Department of Computer Engineering
Chalmers University of Technology

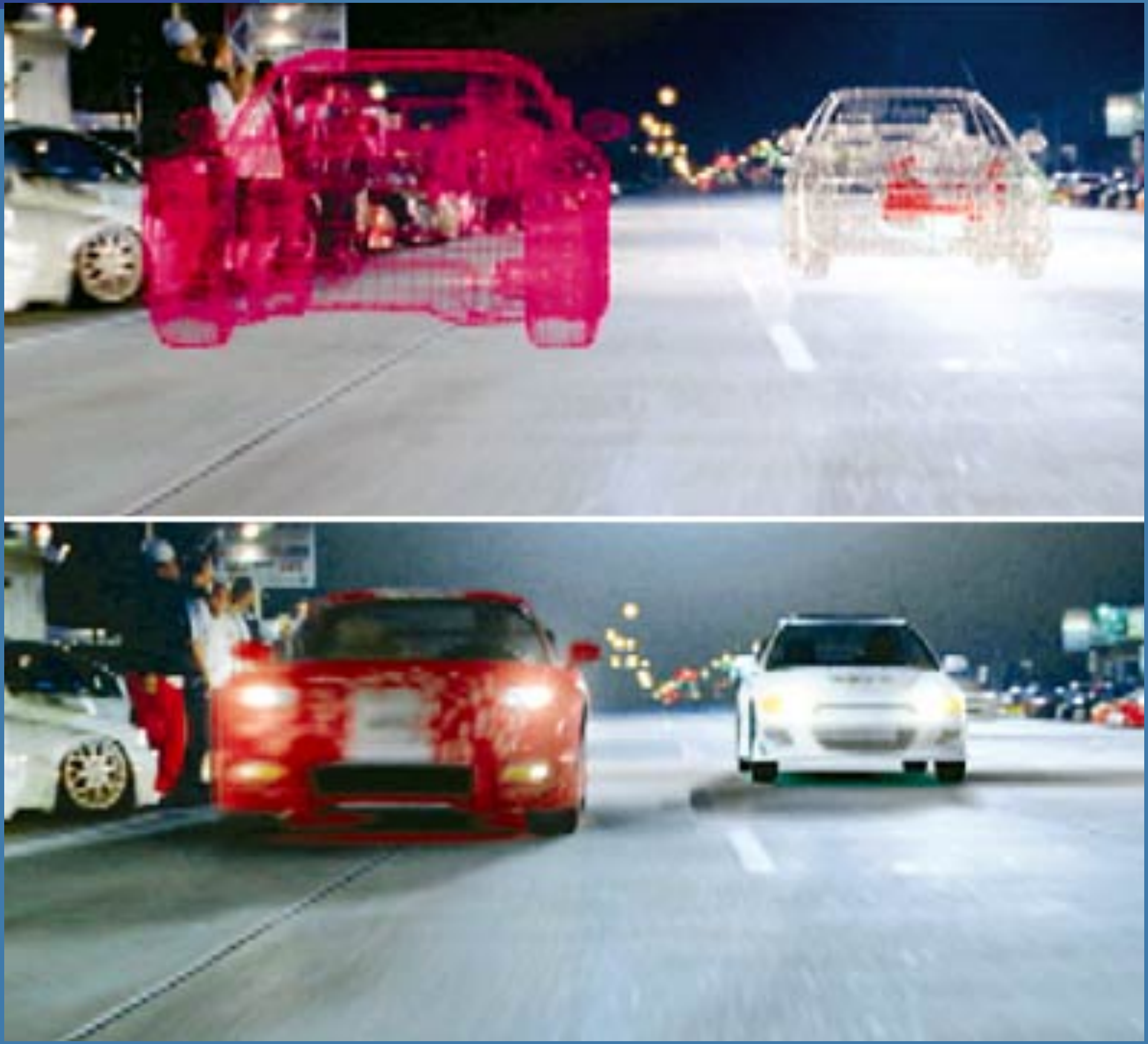
What is aliasing?



Why care at all?

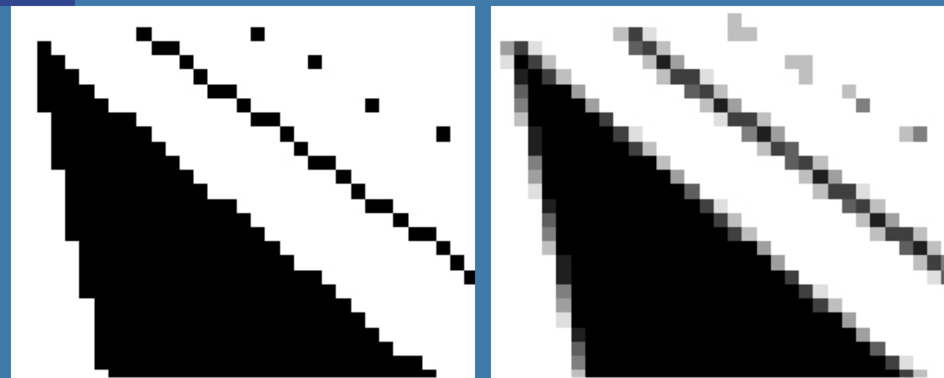
- Quality!!
- Example: Final fantasy
  - The movie against the game
  - In a broad way, and for most of the scenes, the only difference is in the number of samples and the quality of filtering

Physical correctness often less important than filtering



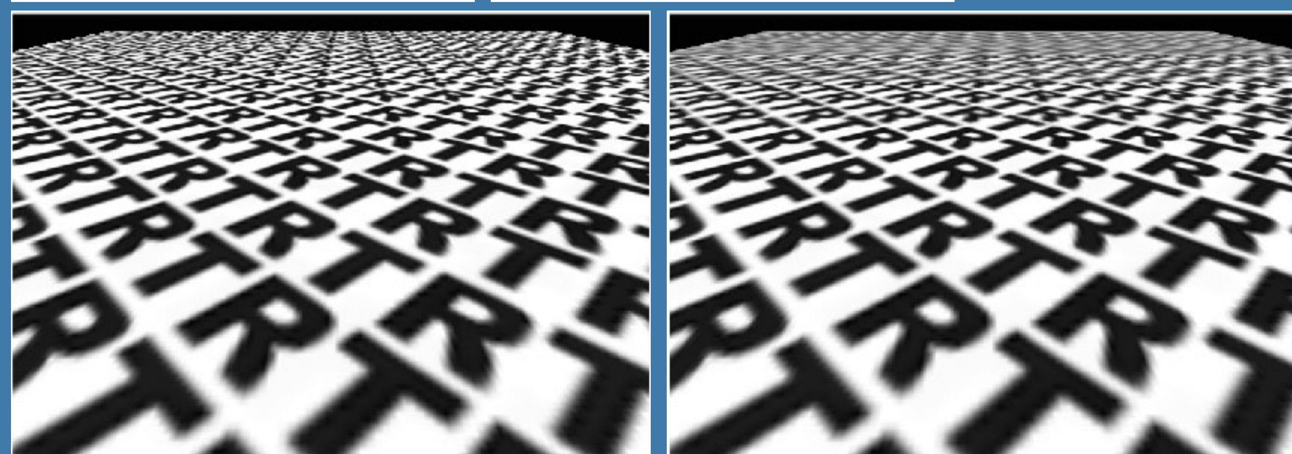
Computer graphics is a SAMPLING & FILTERING process!

- Pixels

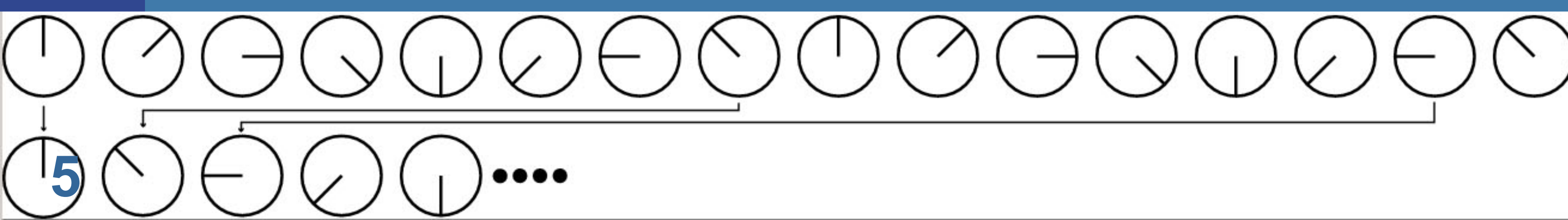


Demo

- Texture

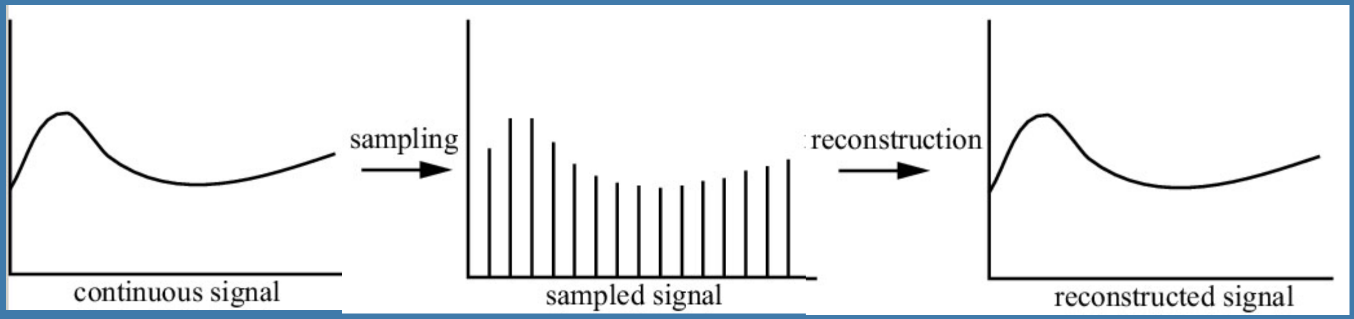


- Time



Motion blur (long exposure times)

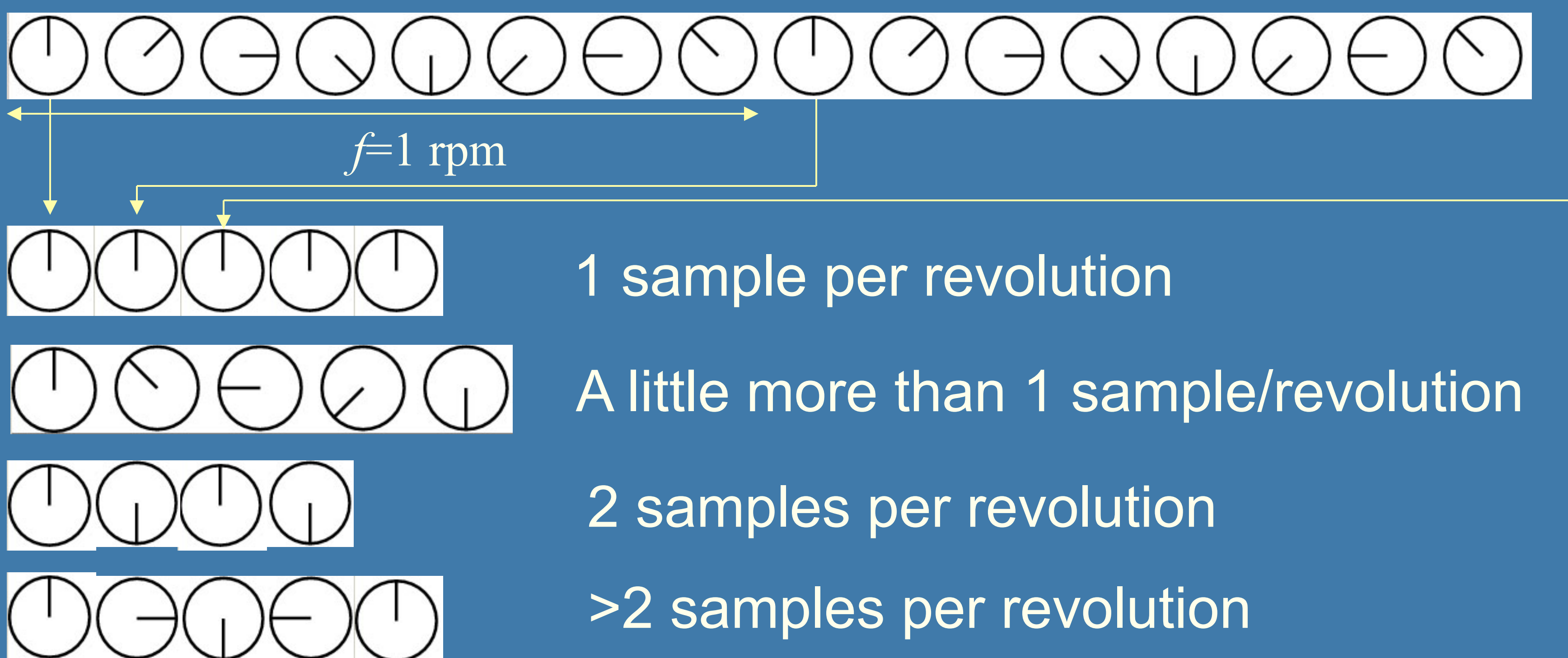
Sampling and reconstruction



- 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…

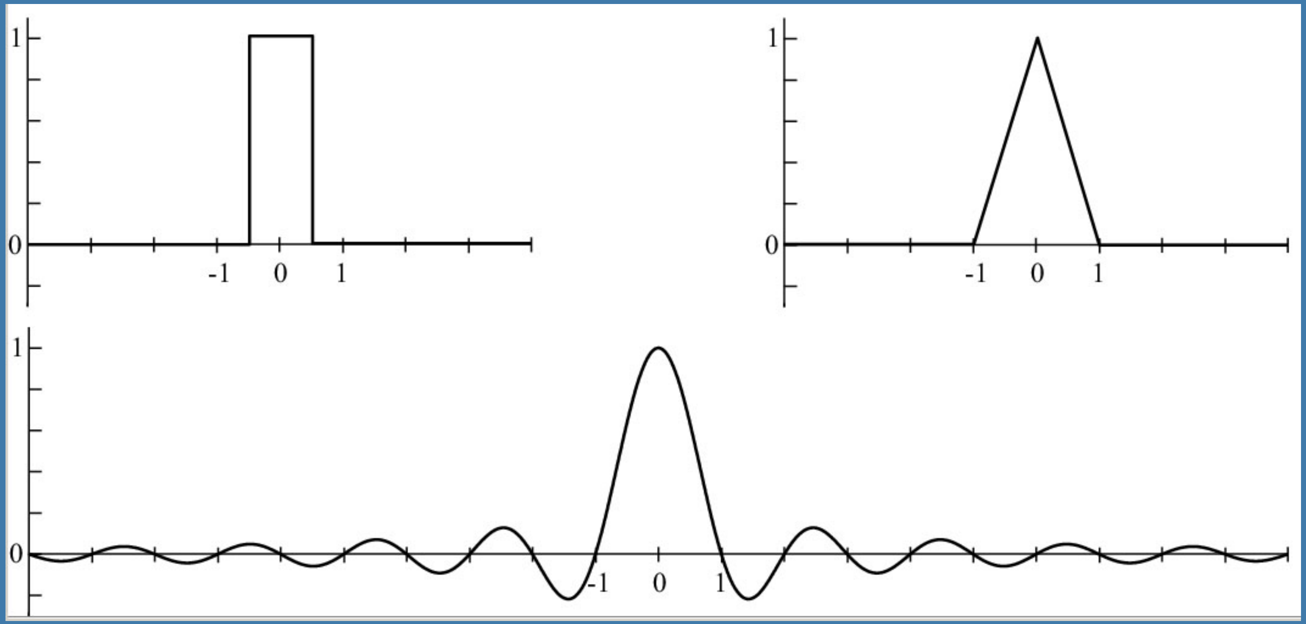
Sampling theorem

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

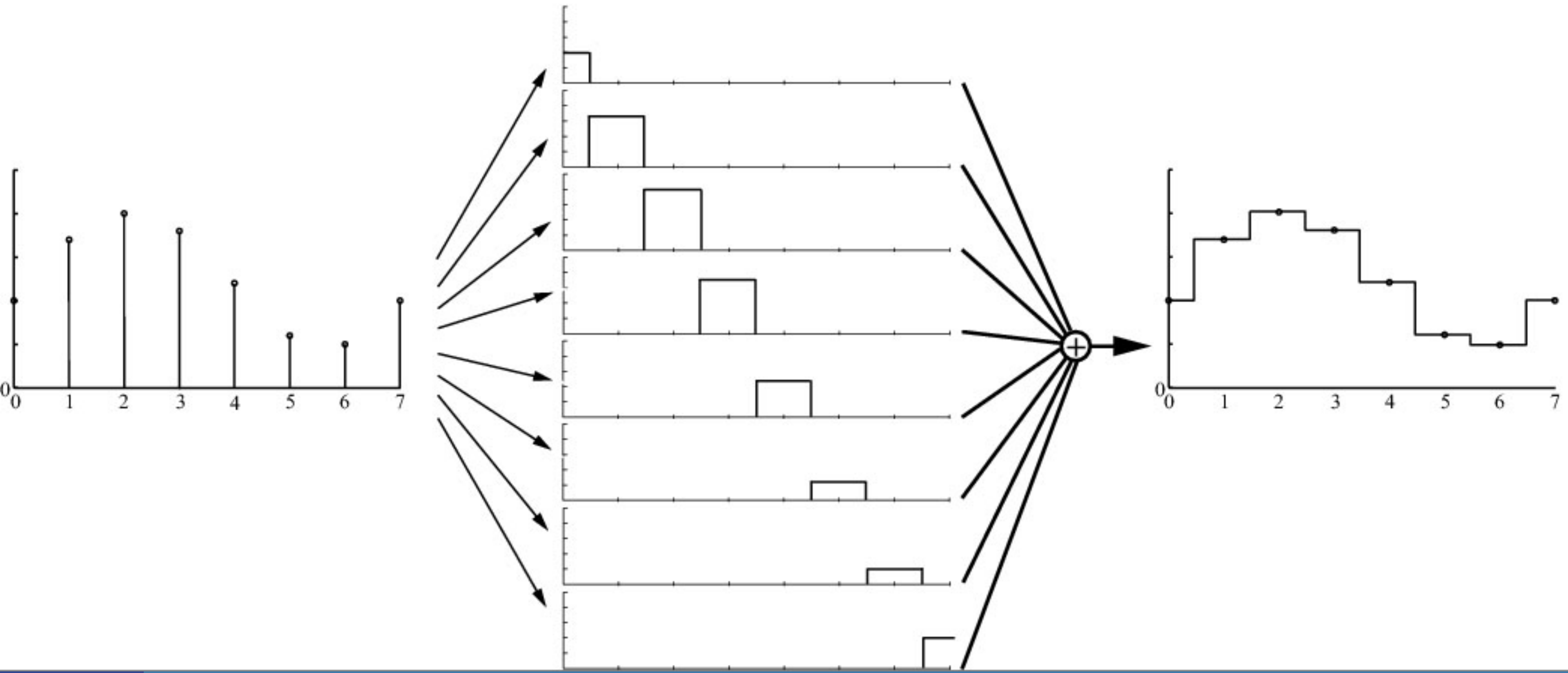


Sampling is simple, now turn to: Reconstruction

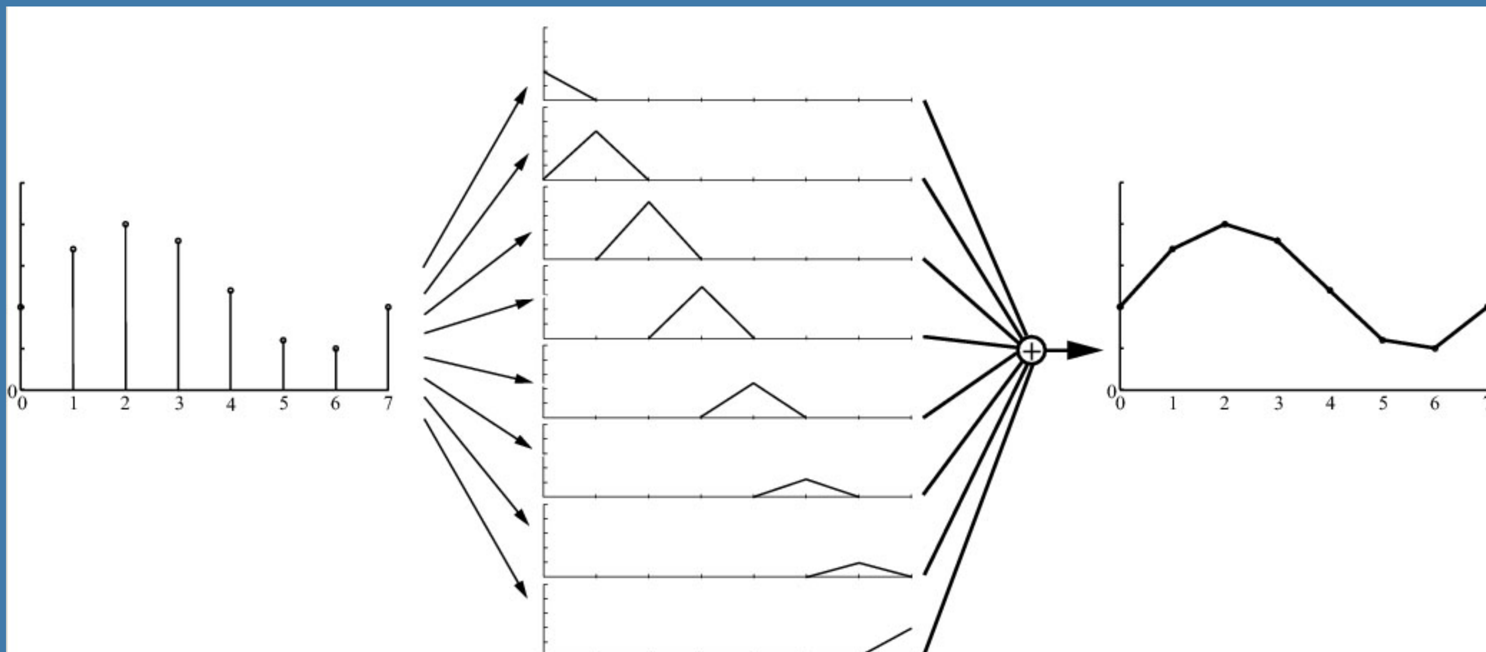
- Assume we have a bandlimited signal (e.g., a texture)
- Use filters for reconstruction



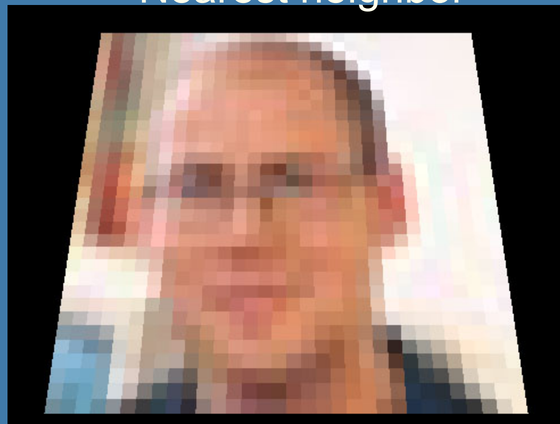
Reconstruction with box filter (nearest neighbor)



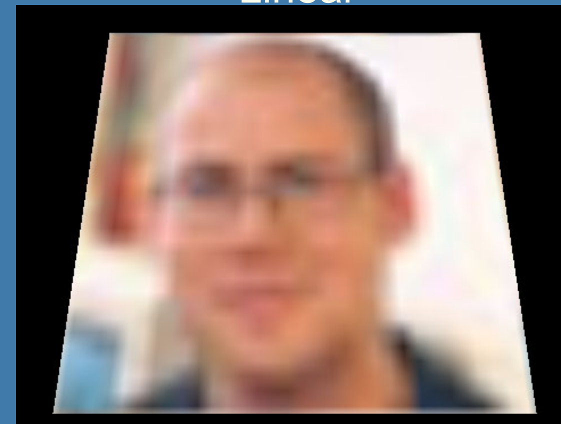
Reconstruction with tent filter



Nearest neighbor



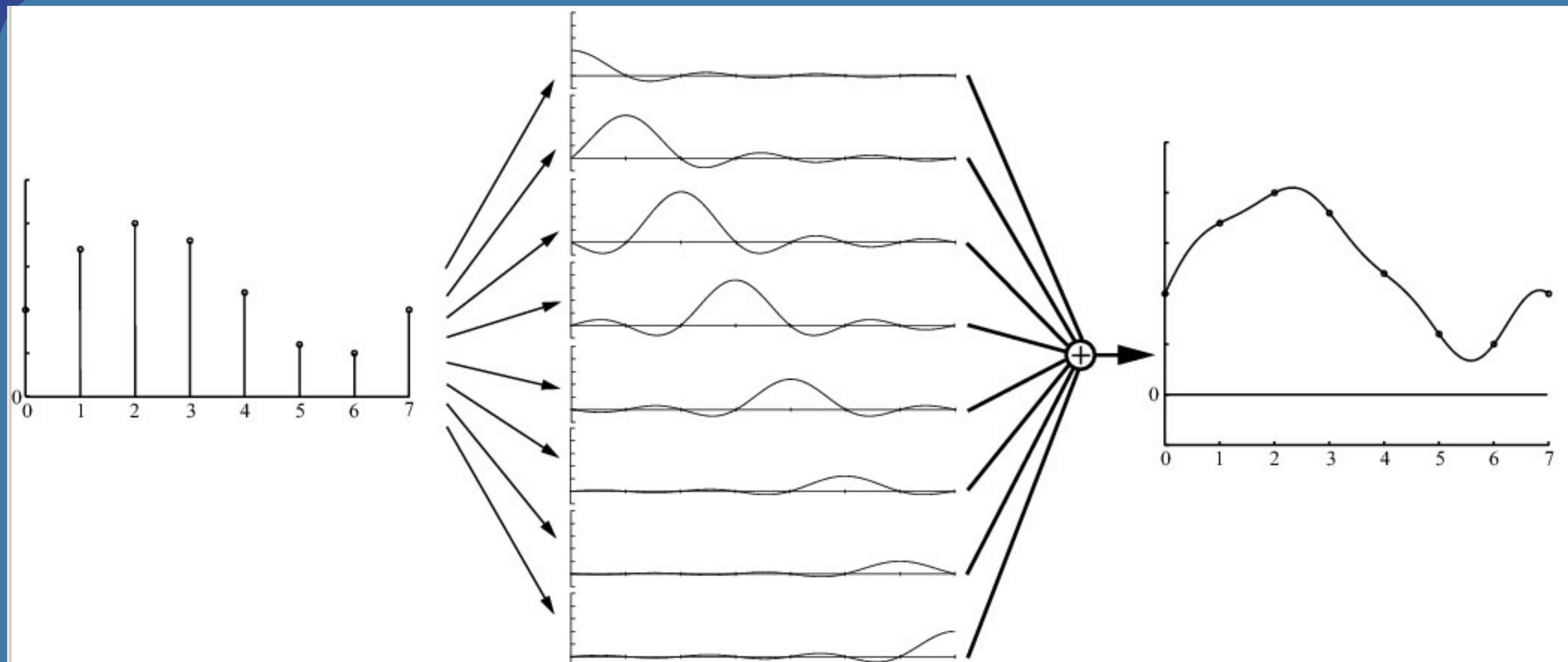
Linear



32x32
texture

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

Reconstruction with sinc filter



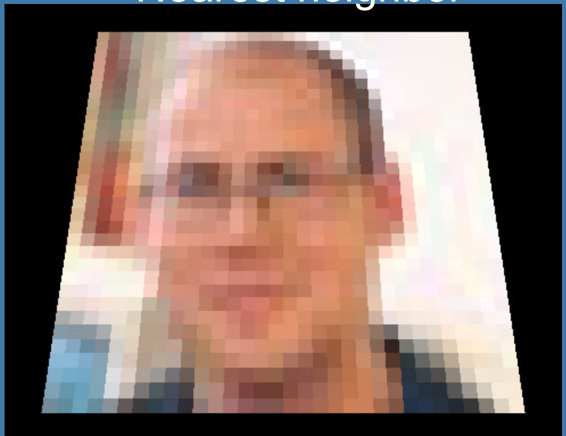
- In theory, the ideal filter
- Not practical (infinite extension, negative)

Resampling

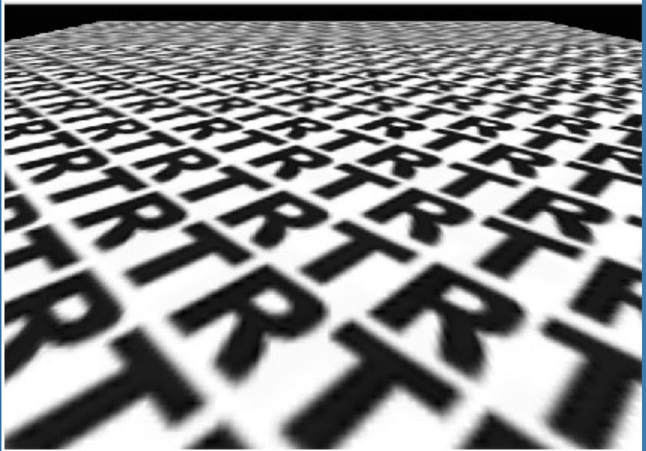
Enlarging or diminishing signals

- 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

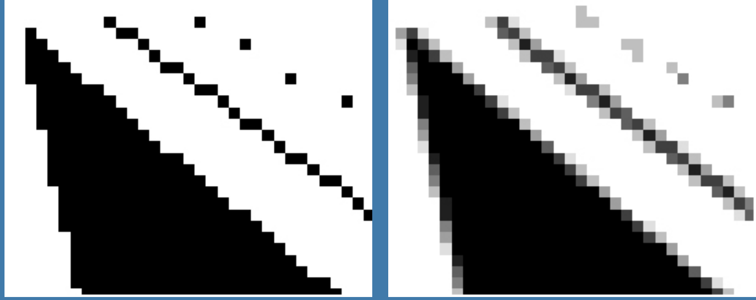
Nearest neighbor



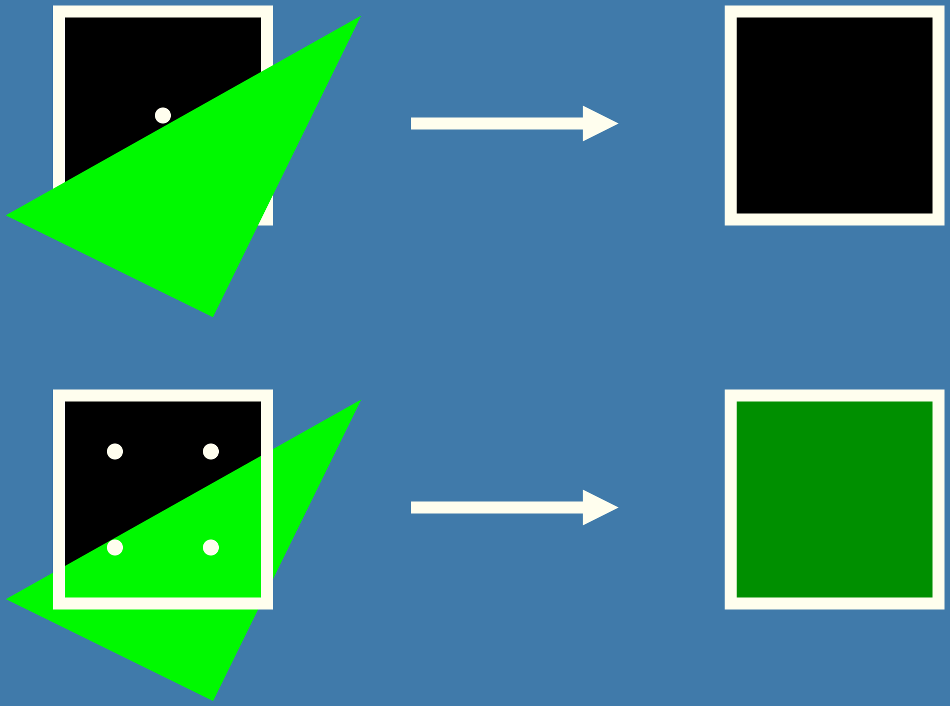
32x32 texture



Screen-based Antialiasing



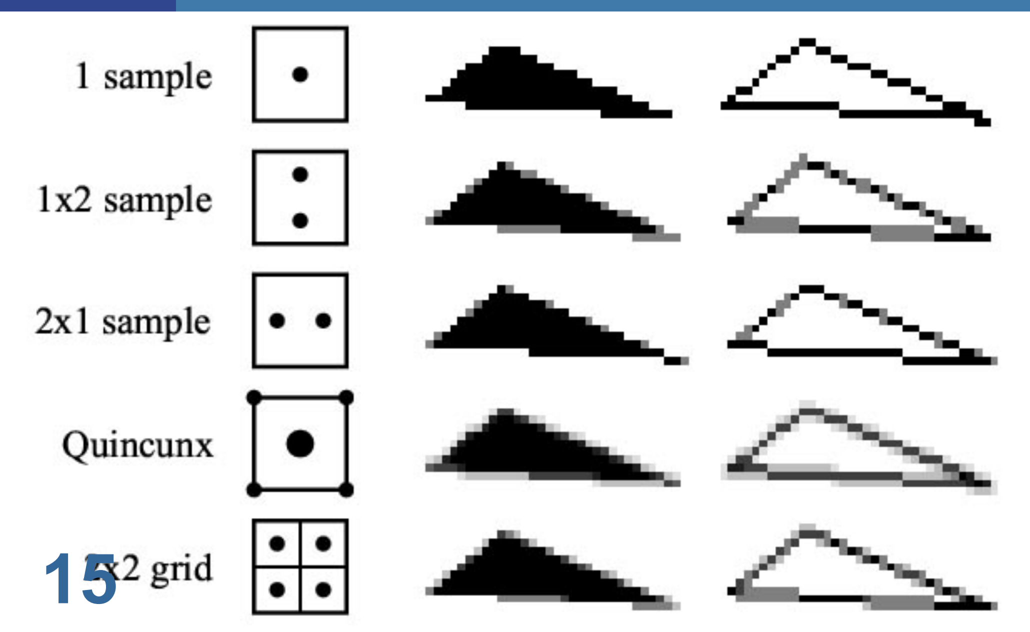
- Hard case: edge has infinite frequency
- Supersampling: use more than one sample per pixel

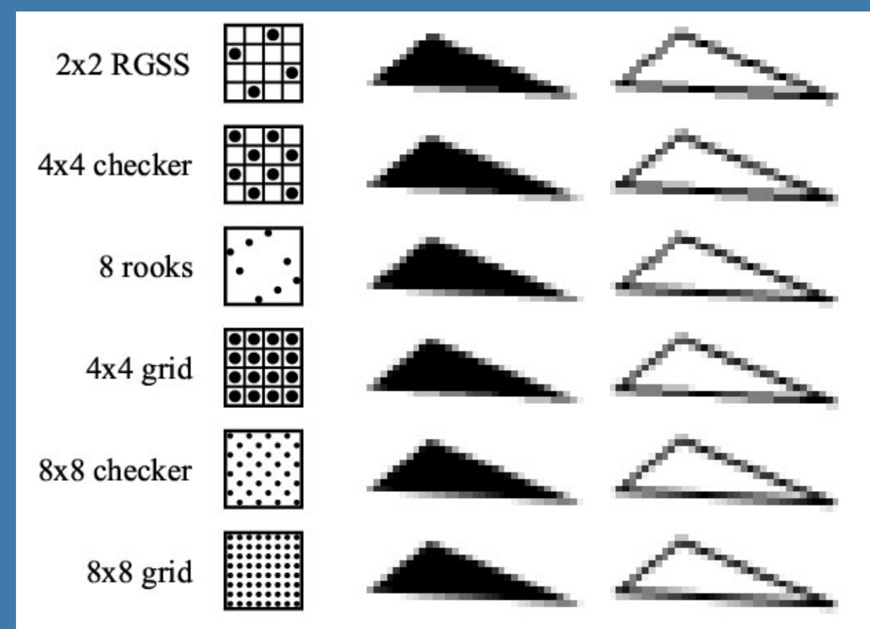


Formula and… examples of different schemes

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

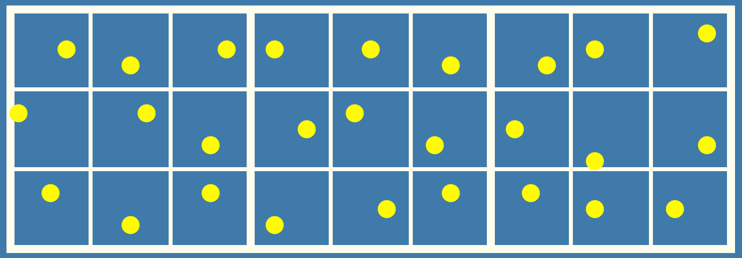
- w_i are the weights in [0,1]
- $\mathbf{c}(i,x,y)$ is the color of sample i inside pixel

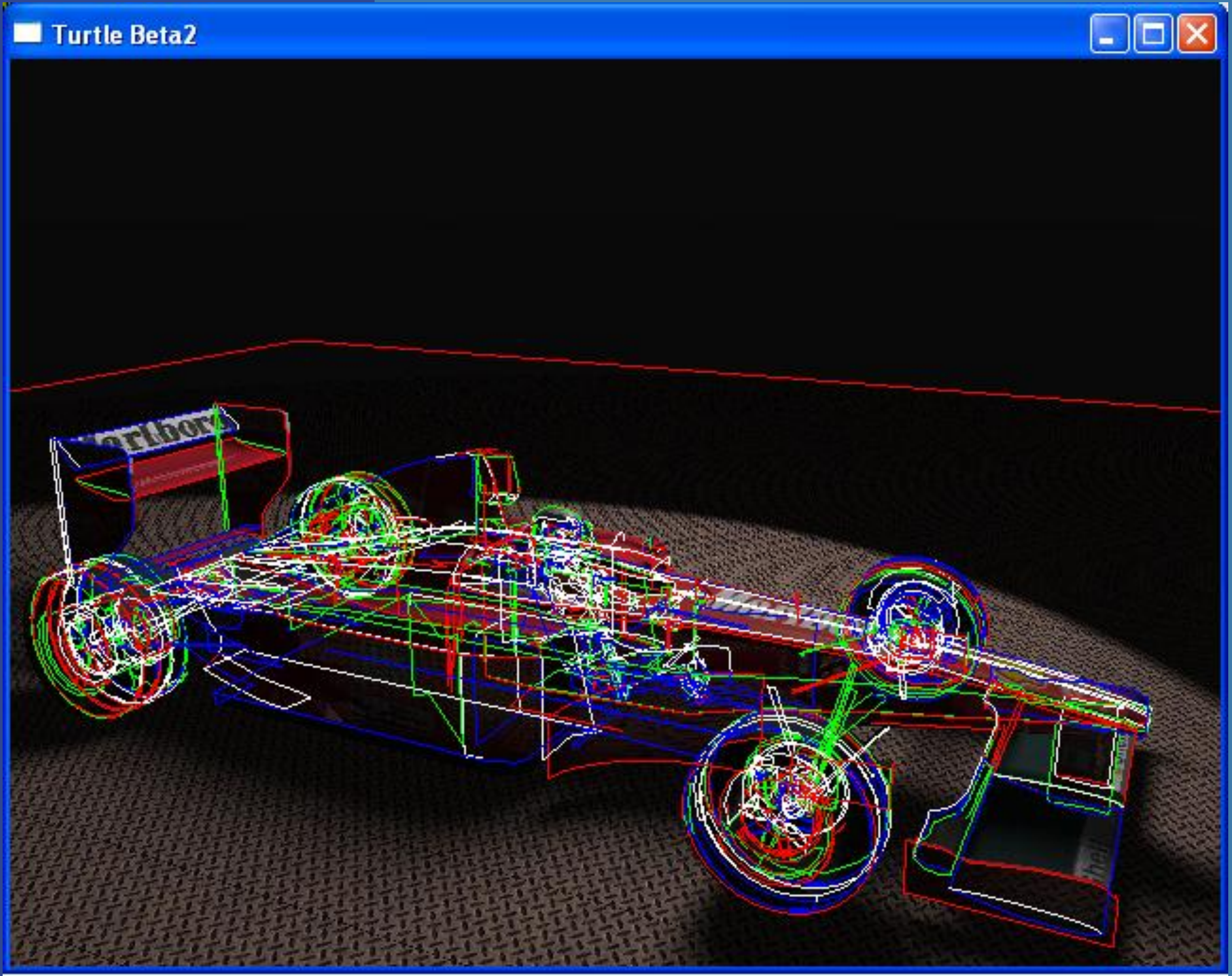




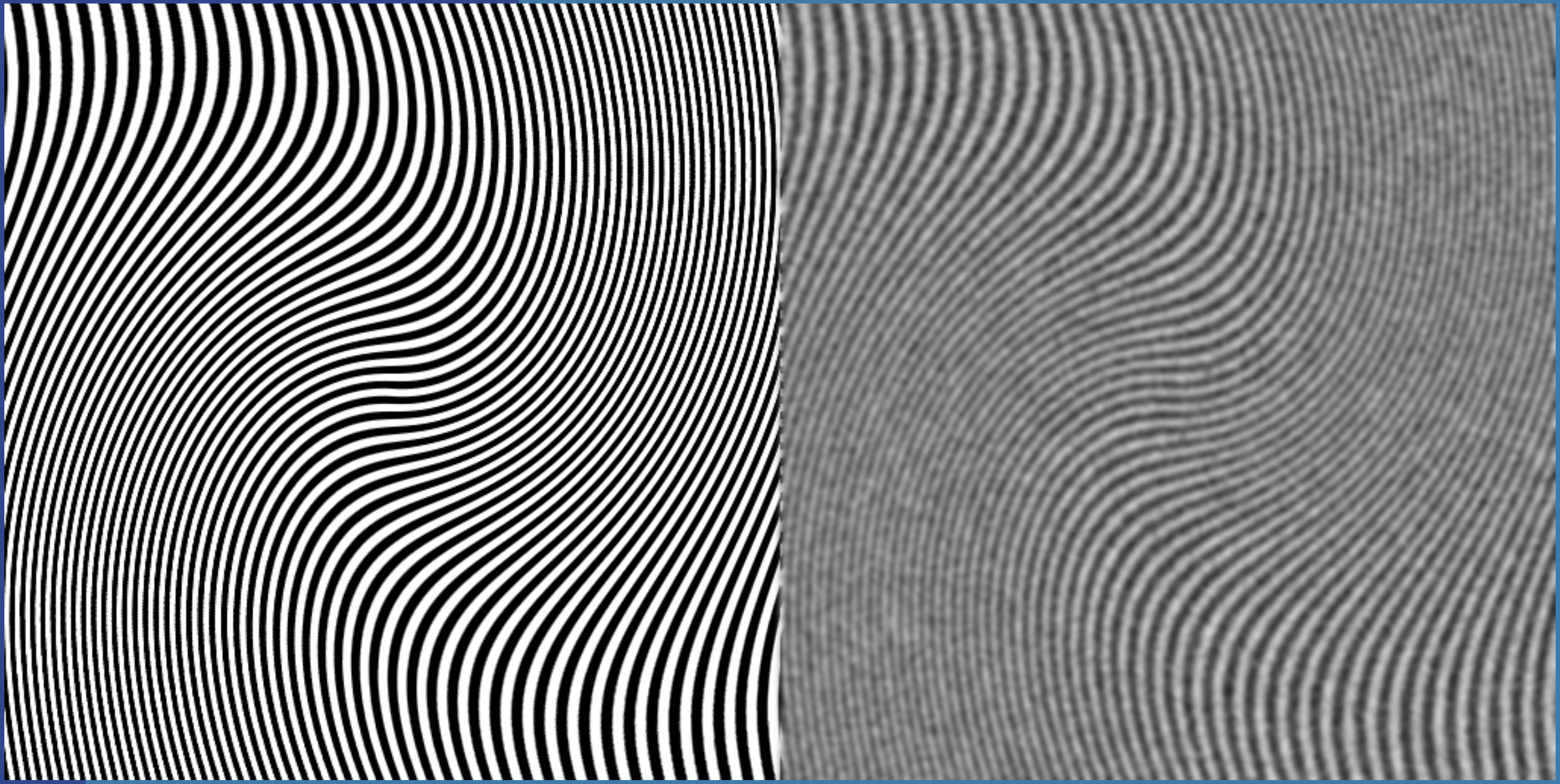
Jittered sampling

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




Turtle Beta2

Moire example



Moire patterns

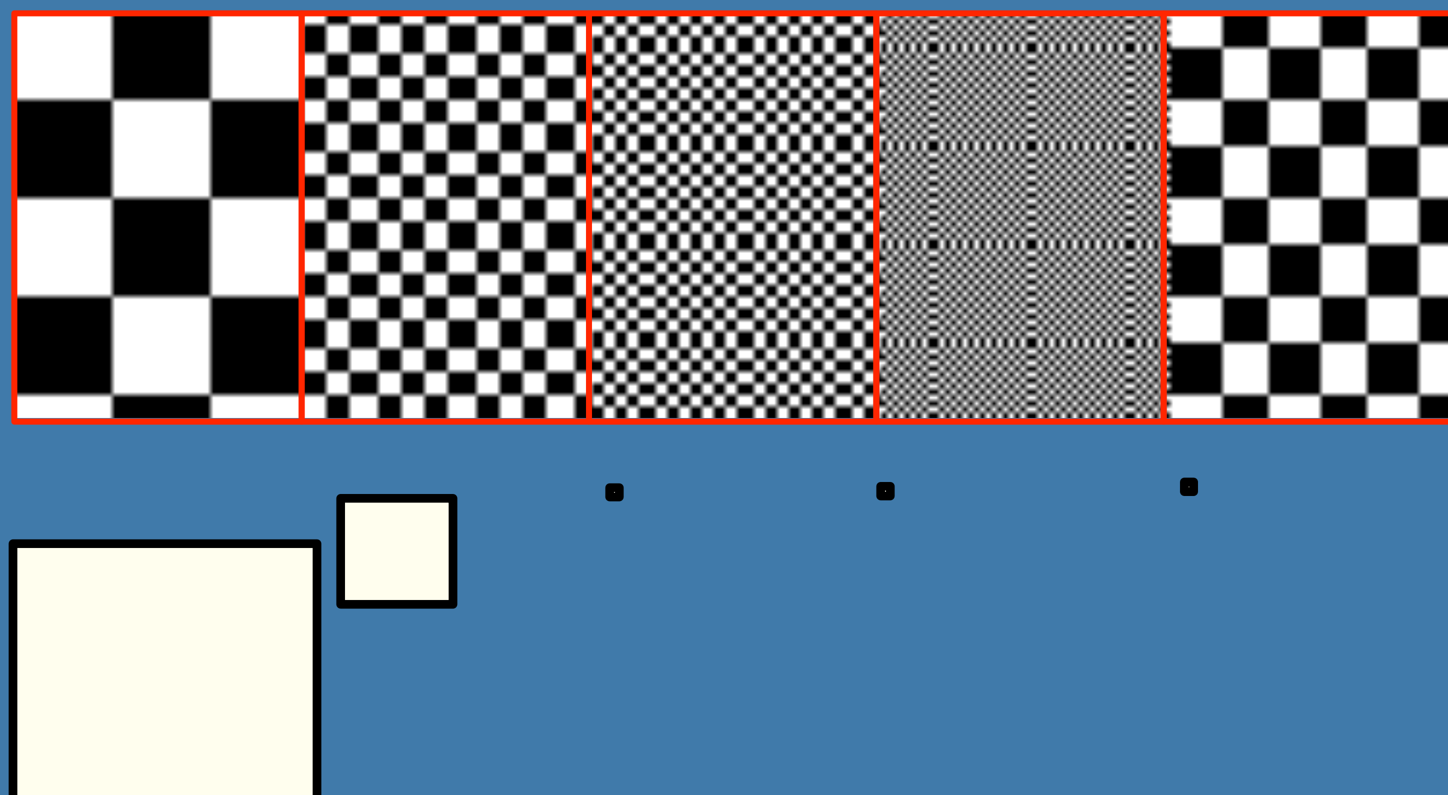
Noise + gaussian blur

(no moire patterns)

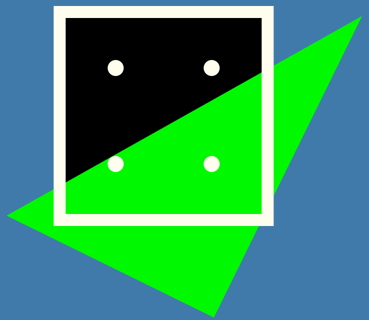
Ulf Assarsson, 2004

Patterns

- Texture zoomed out until square < 1 pixel

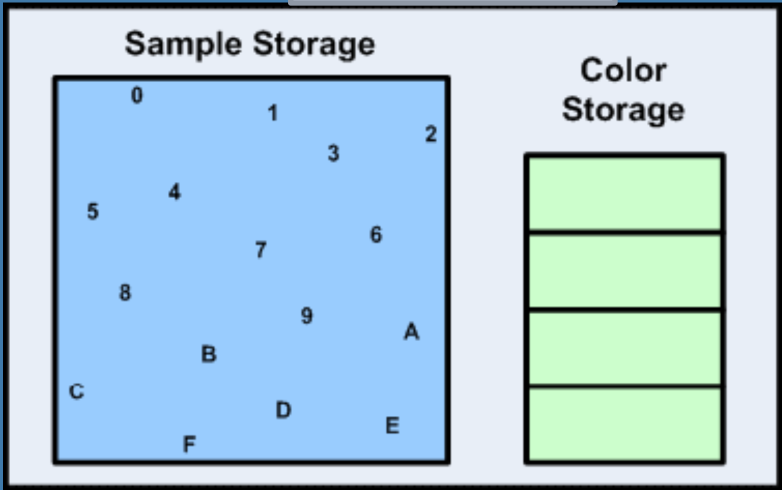


SSAA, MSAA and CSAA



- Super Sampling Anti Aliasing
  - Stores duplicate 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:
    - Frament shader only run once per fragment.
    - 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. Store index in each subsample, into a buffer per pixel of 4-8 colors+depths.
  - fragment shader executed once per fragment
  - E.g., Each sample holds a 2-bit index into a storage of up to four colors per pixel

16x CSAA

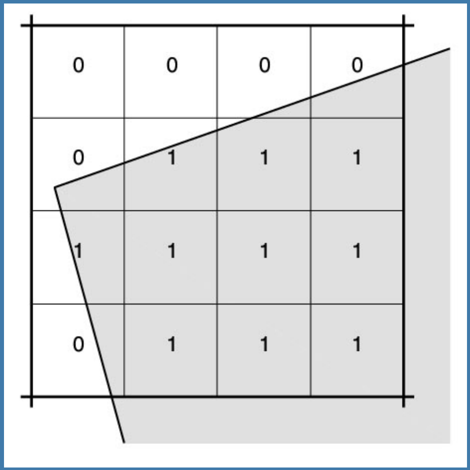


The A-buffer Multisampling technique

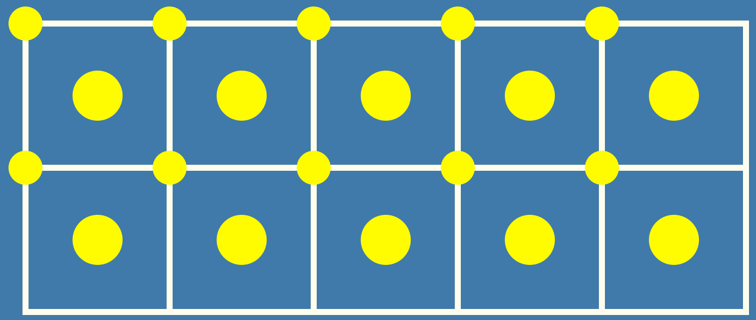
- Takes >1 samples per pixel, and shares compuations between samples inside a pixel
- Supersampling does not share computations (depth, fragment shading)
- Examples:
  - Lighting may be computed once per pixel
  - Texturing may be computed once per pixel
- Strength: anti-aliasing edges (and properly handling transparency)

The A-buffer

- To deal better with edges: use a coverage mask per pixel
- Coverage mask, depth, & color make up a fragment
- During rendering fragments are discarded when possible (depth test)
- When all polygons have been rendered, the fragments are merged into a visible color
  - Allows for sorting transparent surfaces as well
  - But costs memory



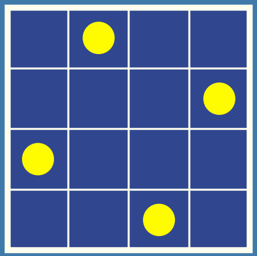
Another multisampling techniqe Quincunx



- 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

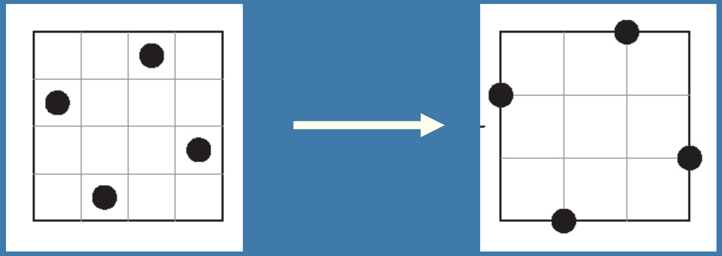
Yet another scheme: FLIPQUAD multisampling

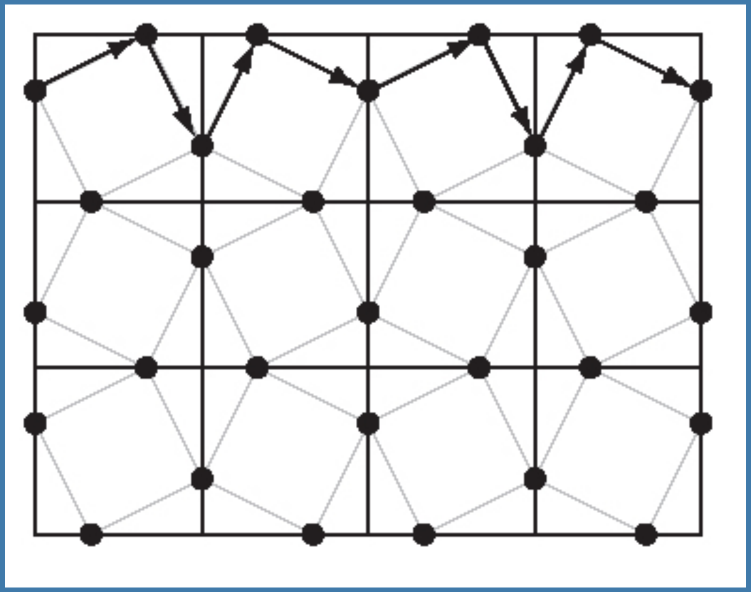
- Recap, RGSS:
  - One sample per row and column



- Combine good stuff from RGSS and Quincunx

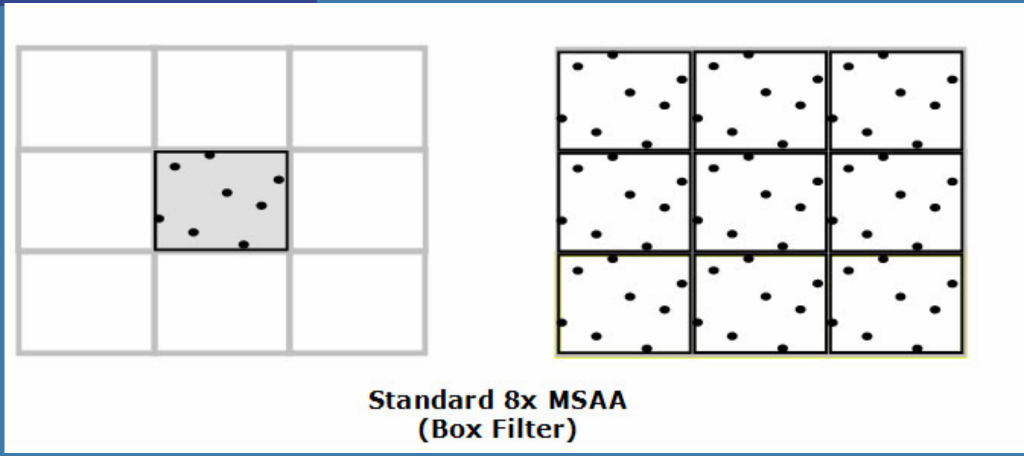
Demo

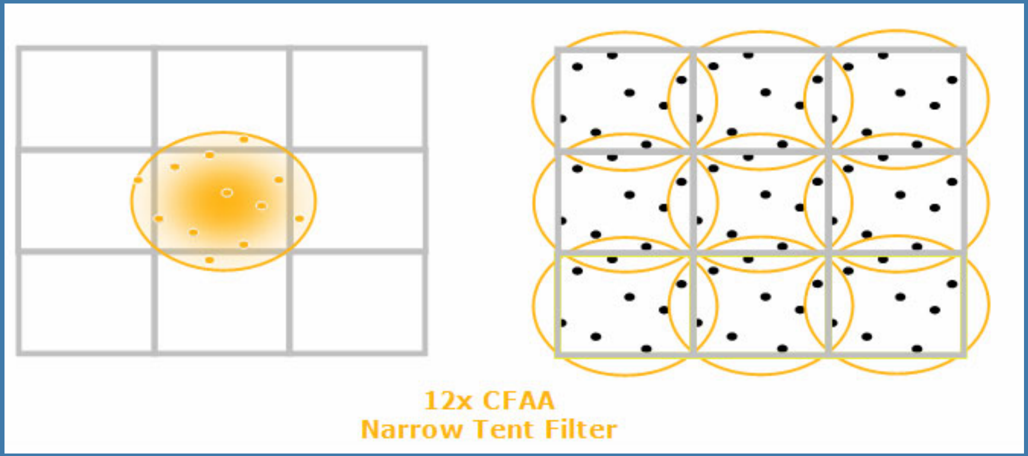




- Weights: 0.25 per sample
- Performs better than Quincunx

ATI Radeon 2900





From www.pcper.com

- Examples of 2 filter modes

More on filtering theory and practice

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

THE END