CS5182 Computer Graphics
Aliasing and Anti-aliasing

2024/25 Semester A

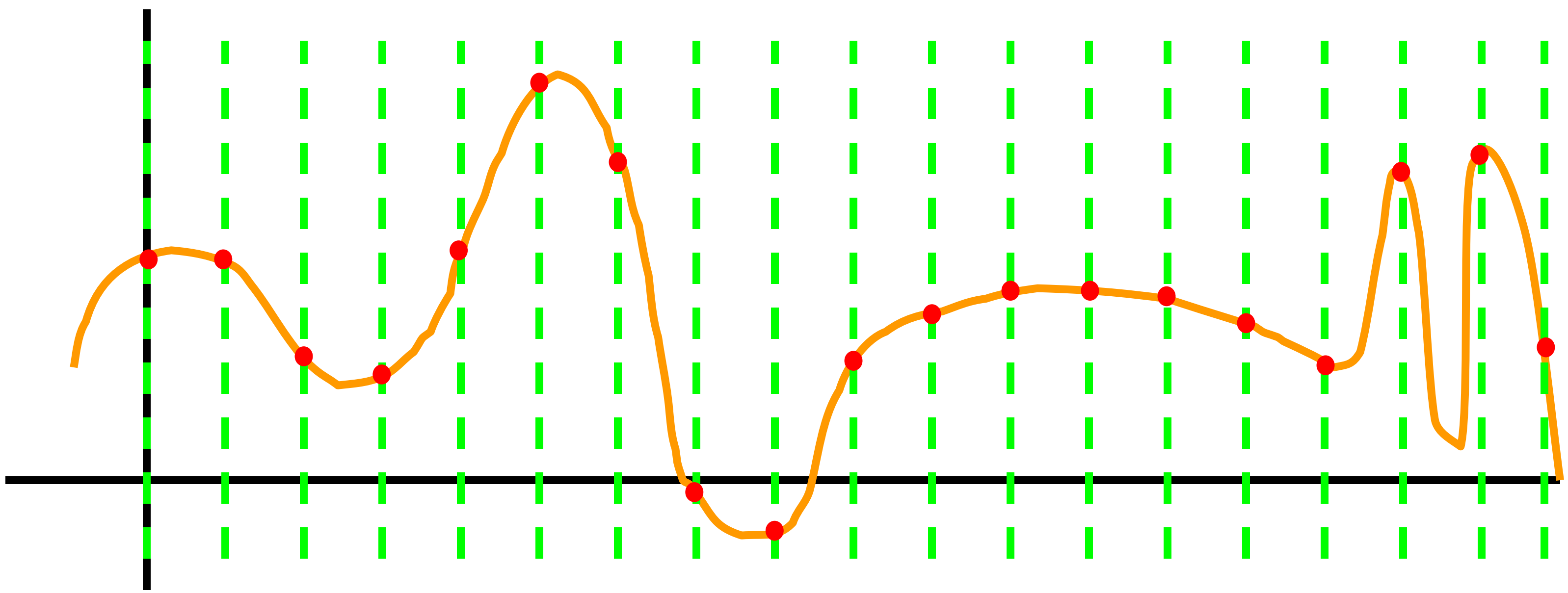
City University of Hong Kong (DG)

Aliasing

- Signal processing (SP) term with very specific meaning
- Computer graphics (CG) term for any unwanted visual artifacts

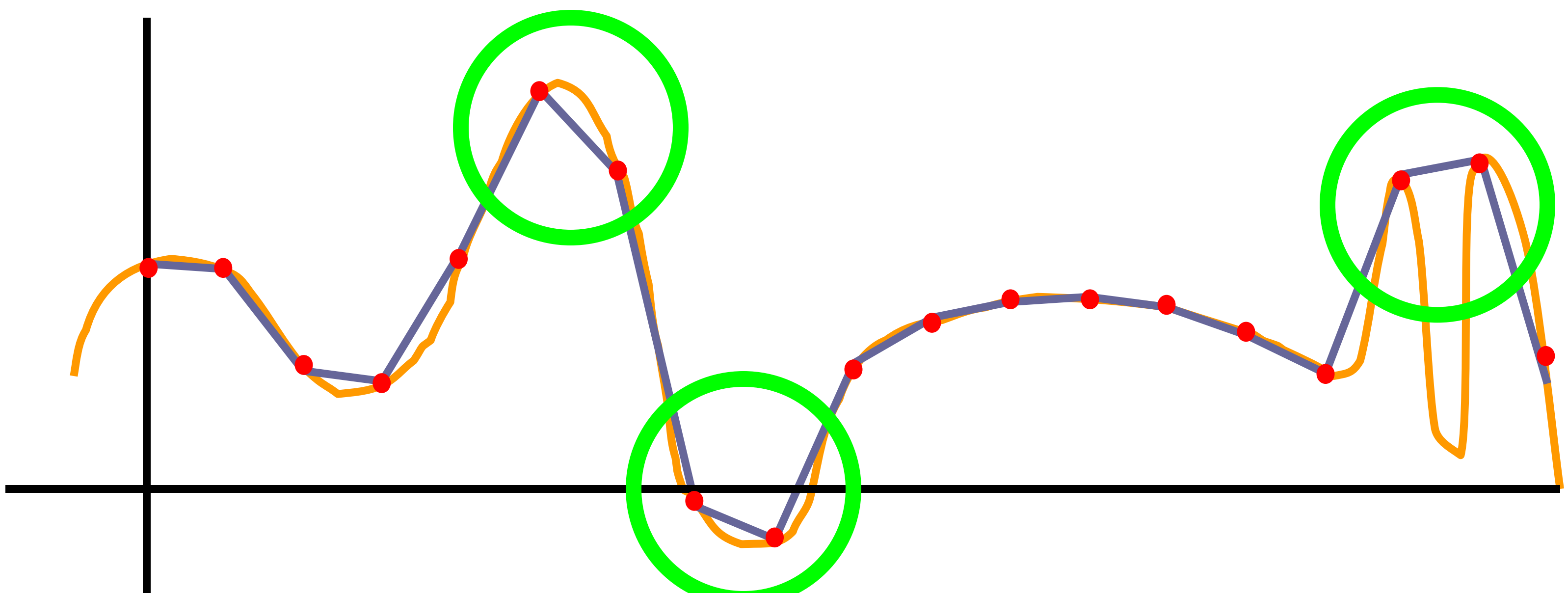
Aliasing in SP

- Regularly sampling a 1-D function:



Aliasing in SP

- Regularly sampling a 1-D function: What do you notice?
  - Jagged, not smooth
  - Lose information!



Aliasing in SP

- Regularly sampling a 1-D function: what do you notice?
  - Jagged, not smooth
  - Lose information!

- What can we do about these?
  - Use higher-order reconstruction
  - Use more samples (*How many more samples?)*

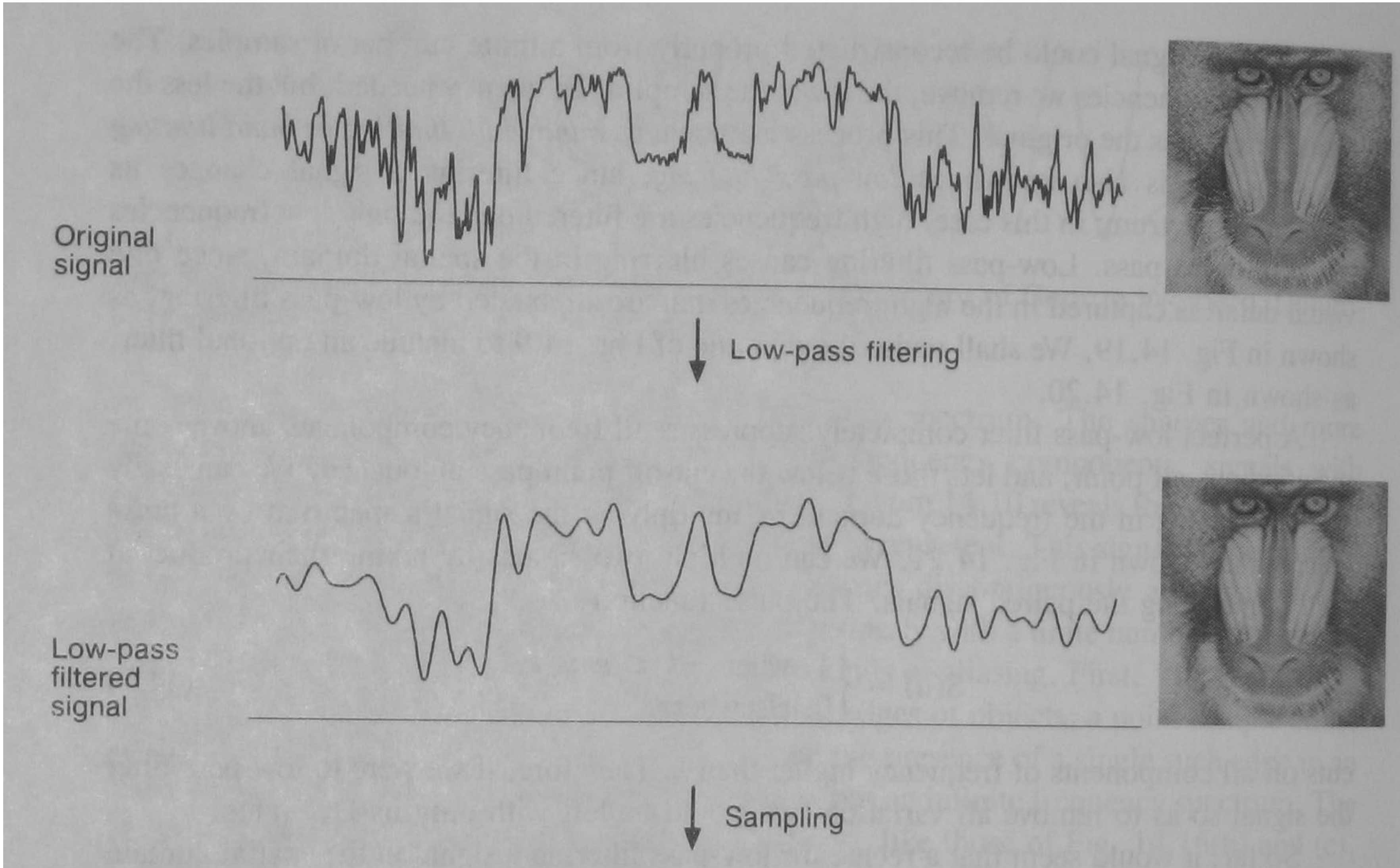
The Sampling Theorem

- Obviously, the more samples we take the better those samples approximate the original function
- The Nyquist sampling theorem:

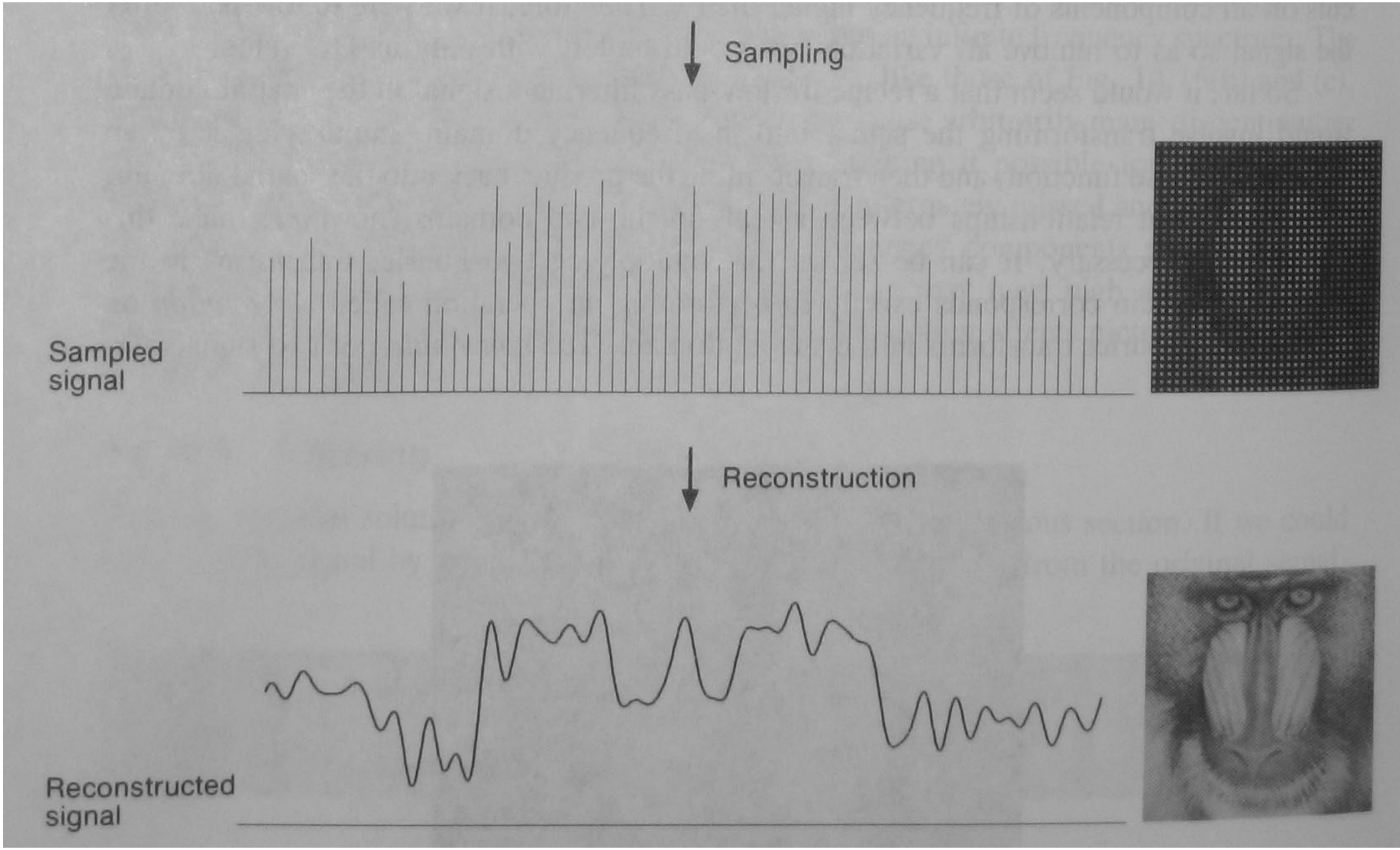
  A continuous bandlimited function can be completely represented by a set of equally spaced samples, if the samples occur at more than twice the frequency of the highest frequency component of the function.

- In other words, to adequately capture a function with maximum frequency F, we need to sample it at frequency $N = 2F$, where N is called the *Nyquist limit*.

Prefiltering for Anti-aliasing in SP



Prefiltering for Anti-aliasing in SP



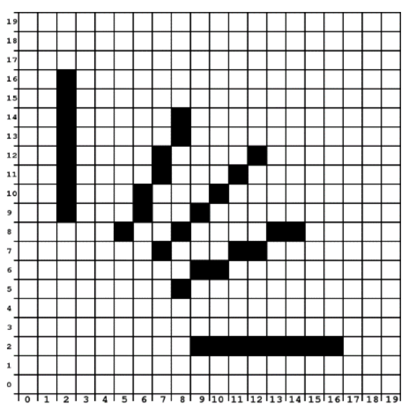

Normal Resolution S1.JPG
S1 - Without Low Pass Filter
High Resolution S1H.tif
14/08/19 11:32:20 AM
12000 x 8000
S1H - With Low Pass Filter
(High Res Mode)

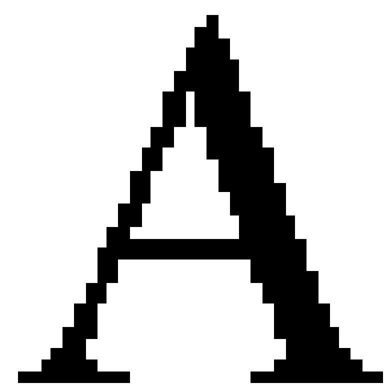
The Aliasing Problems in CG

- In 2D drawing, a straight line may look like staircases and a character may look like discretized. This kind of phenomenon is called *aliasing*.
  - Spatial aliasing is the aliasing that appears in static images.
  - Temporal aliasing is the aliasing that can be observed in video sequences. Two main reasons:

    -- As a result of spatial aliasing

    -- Undersampling in time domain.

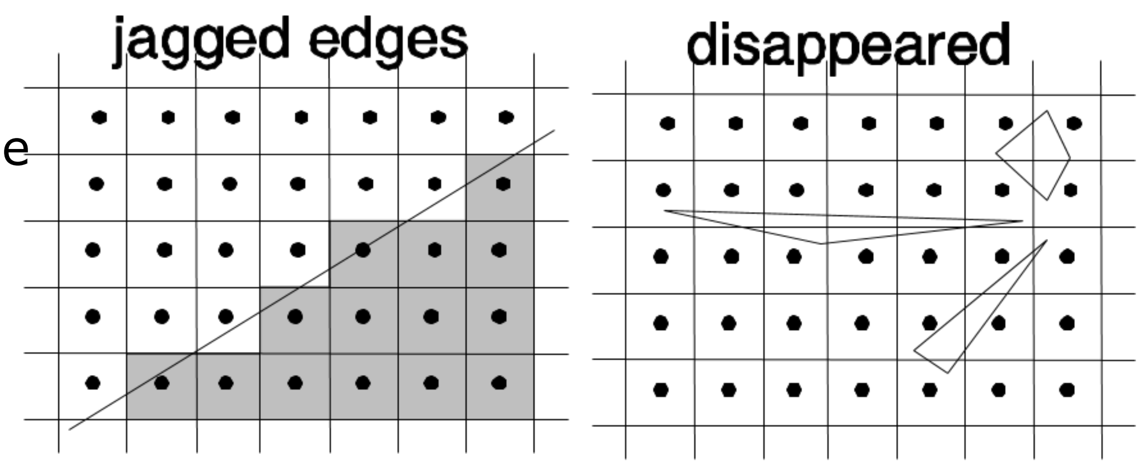




Spatial Aliasing in CG

- Spatial aliasing occurs mainly due to the point sampling nature of most drawing/rendering algorithms.
  - In 2D drawing, for example, if a line covers the sample point, we assume that it covers the whole pixel. However, if it does not cover the sample point, we assume that it does not cover the pixel at all. This is a rather coarse approximation.

In (a), a straight edge looks like staircases.

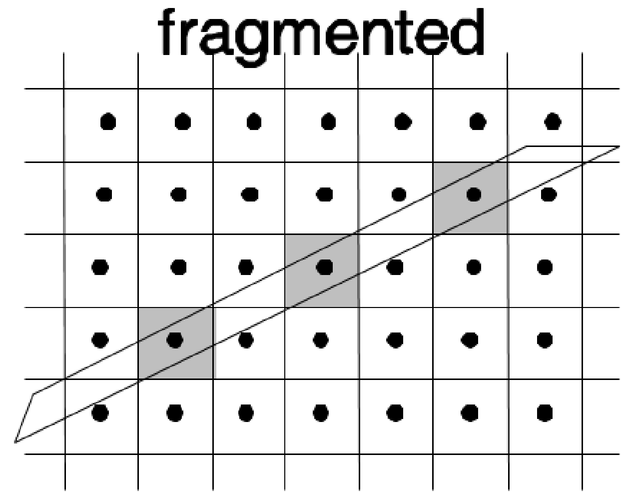


(a) (b)

In (b), because the triangles do not fall into any of the sample points, they simply disappear.

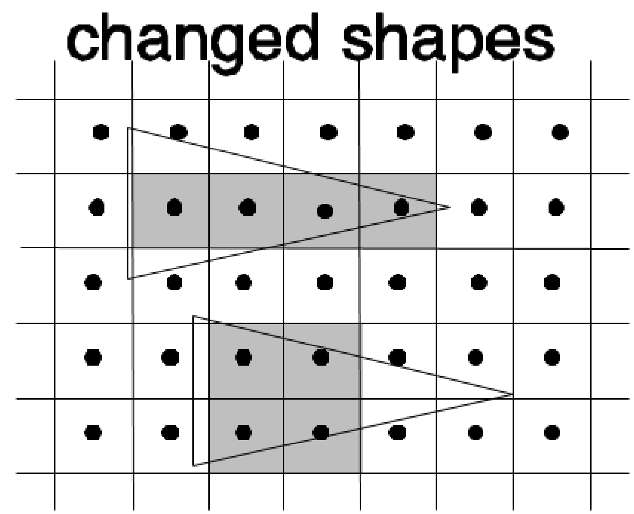
Spatial Aliasing in CG

In (c), some parts of a thin triangle do not fall into the sample points while the rest do. The triangle is fragmented into many pieces.



(c)

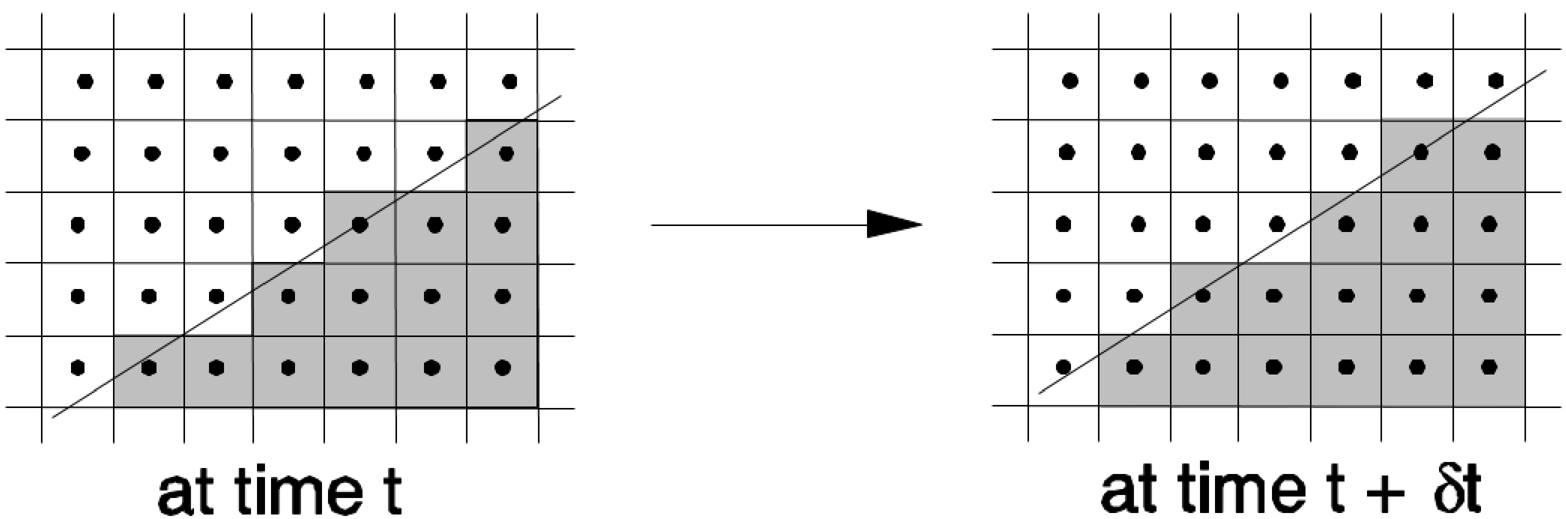
In (d), although we have two identical triangles, one appears as a rectangle and the other as a square. Hence, the appearance of small objects is affected by their physical positions on the pixel grid.



(d)

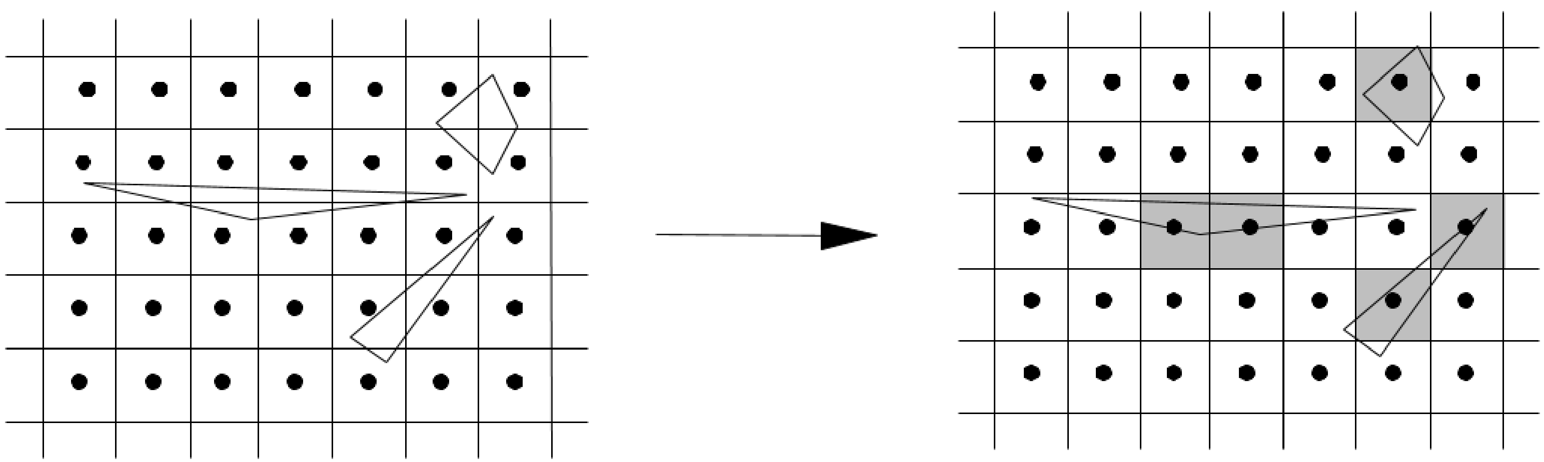
Temporal Aliasing in CG

- Temporal aliasing due to spatial aliasing
  - If the polygon represents an object which moves slightly upward from time t to time $t + \delta t$ in an animation sequence, the staircases along the edge may appear to be jumping around.



Temporal Aliasing in CG

- Temporal aliasing due to spatial aliasing
  - The changes may be even more obvious. At time t, the three triangles are invisible but at time $t + \delta t$, they become visible. The overall effect is that the objects appear to be blinking as they move.

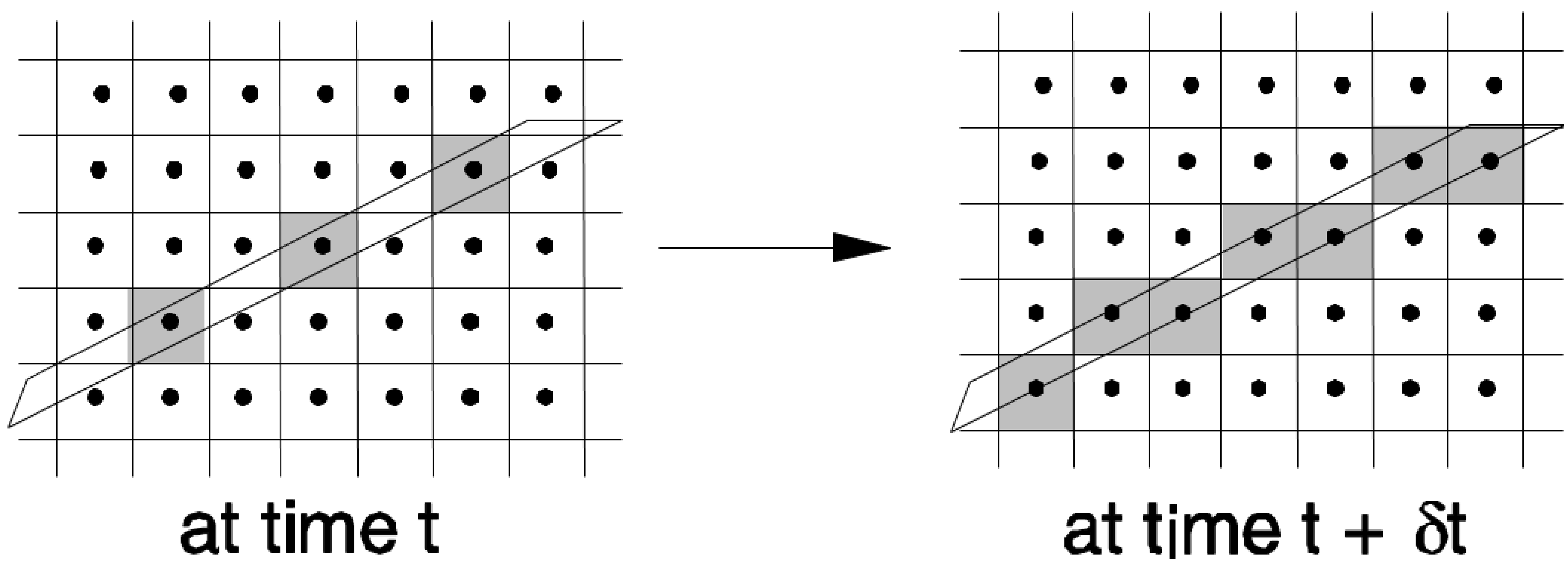


at time t

at time t + δt

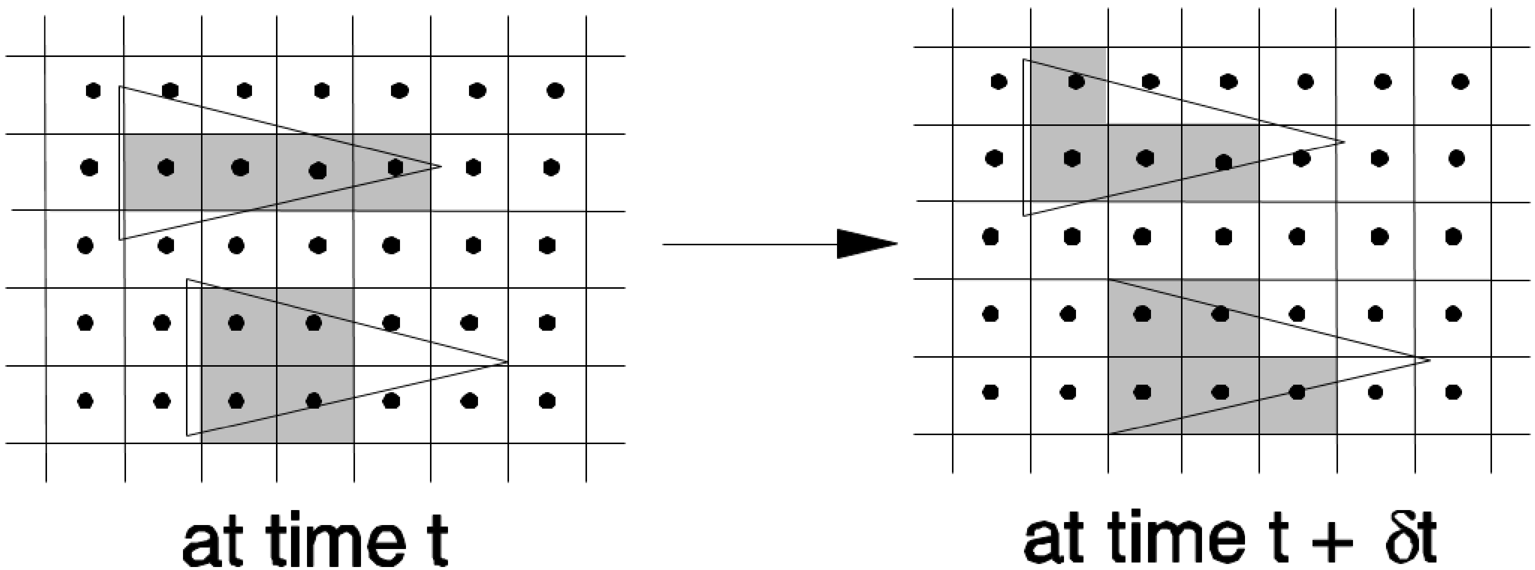
Temporal Aliasing in CG

- Temporal aliasing due to spatial aliasing
  - The triangle is fragmented into many disconnected pieces at time t. However, at time $t + \delta t$, it becomes a single connected object. The overall effect is that an object repeatedly decomposes and merges as it moves.



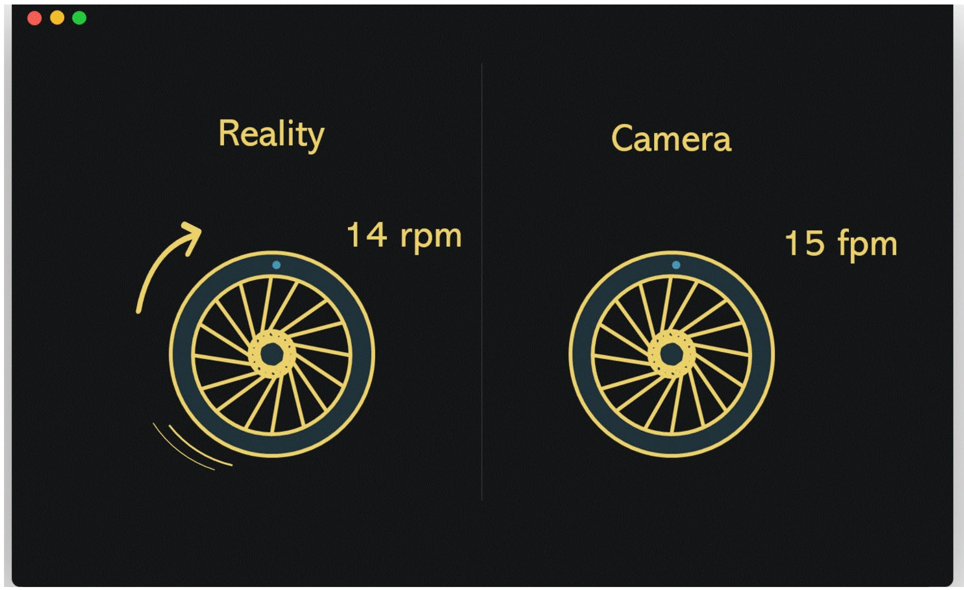
Temporal Aliasing in CG

- Temporal aliasing due to spatial aliasing
  - The two triangles appear to be changing shapes continuously as they move.



Temporal Aliasing in CG

- Temporal aliasing due to undersampling
  - Consider the following examples:



The wheel is in fact rotating clockwise. After sampling, the wheel appears rotating counter-clockwise

The vehicle is traveling forward. After sampling, the vehicle appears traveling backward.

Anti-aliasing in CG

- Anti-aliasing methods refer to techniques for solving or reducing the aliasing problem.
  - Supersampling
  - Accumulation buffer
  - Stochastic sampling
  - Catmull’s algorithm
  - The A-buffer method

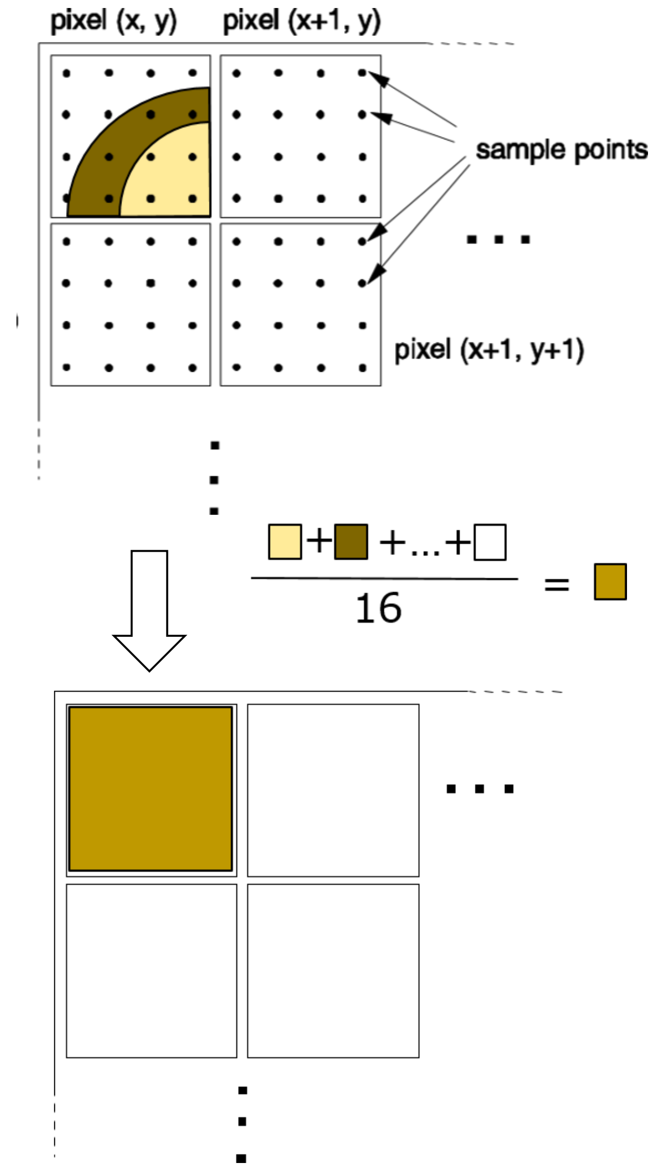
Anti-aliasing in CG

- Supersampling
  - This method increases the number of samples so as to reduce the aliasing effect.
  - The process:

    -- Create a virtual image at higher resolution than the final image

    -- Apply a low-pass filter

    -- Resample the filtered image
  - In practice, we combine steps 2&3, i.e., only convolve filter with image at new samples points



Anti-aliasing in CG

- Supersampling
  - Advantage

    -- Easy to implement the method in hardware. (Majority of the GeForce accelerator cards have implemented this method in hardware.)

  - Disadvantage

    -- It requires considerable amount of memory and processing time.

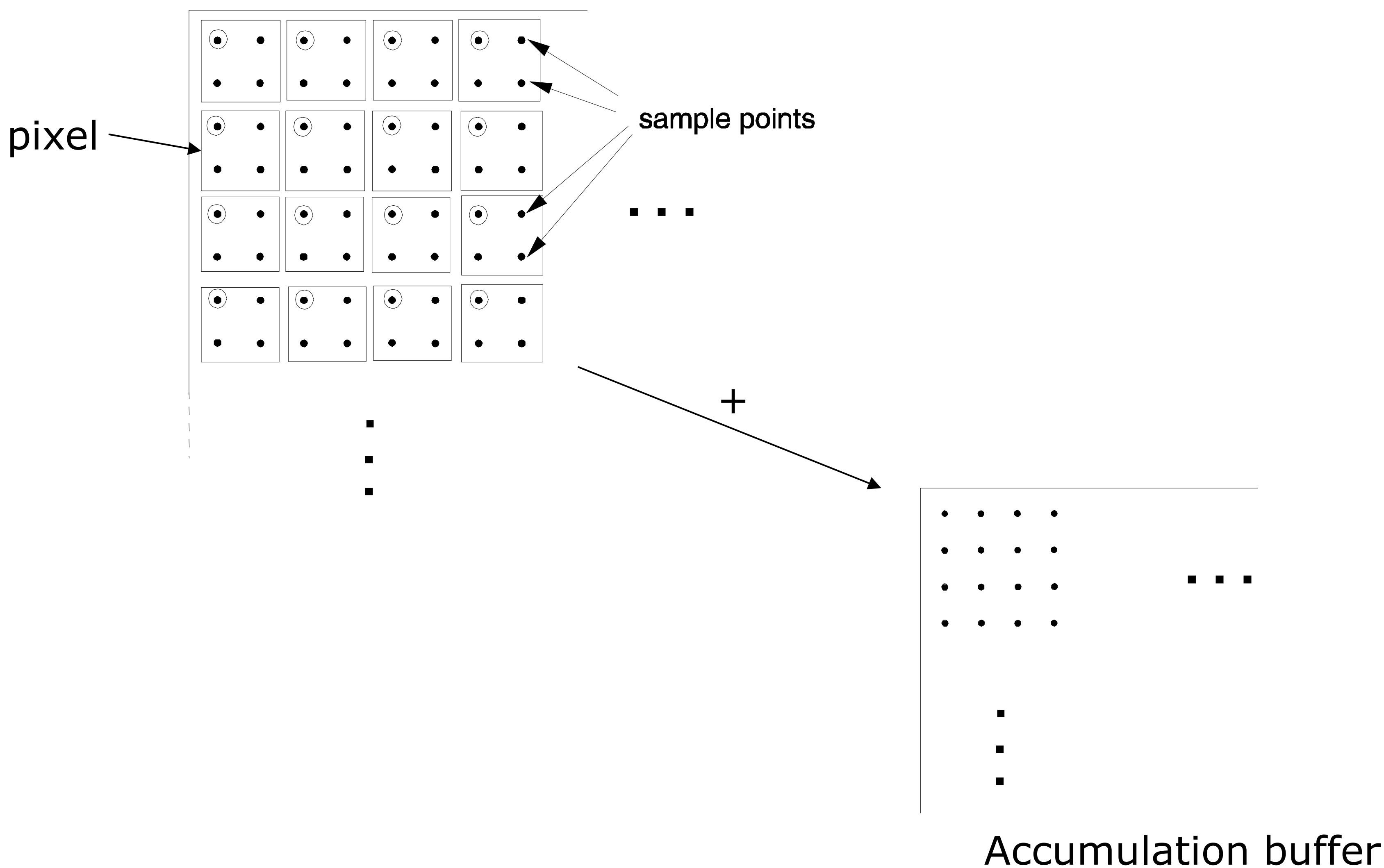
    -- It does not eliminate aliasing; it only reduces it. *Why?*

Anti-aliasing in CG

- Accumulation buffer
  - This method addresses the memory cost of the supersampling method, but with added advantages.
  - Instead of rendering all subpixels in a single pass, this method renders one subpixel at each pass.
  - Let the subpixel resolution be $a \times b$ and the final image resolution be $m \times n$. This method renders the image $a \times b$ times. An image of resolution $m \times n$ is generated in each pass. Each location of this image stores a subpixel color value from one pixel.
  - The output image is accumulated in a buffer called *Accumulation Buffer*, until all $a \times b$ images are rendered.

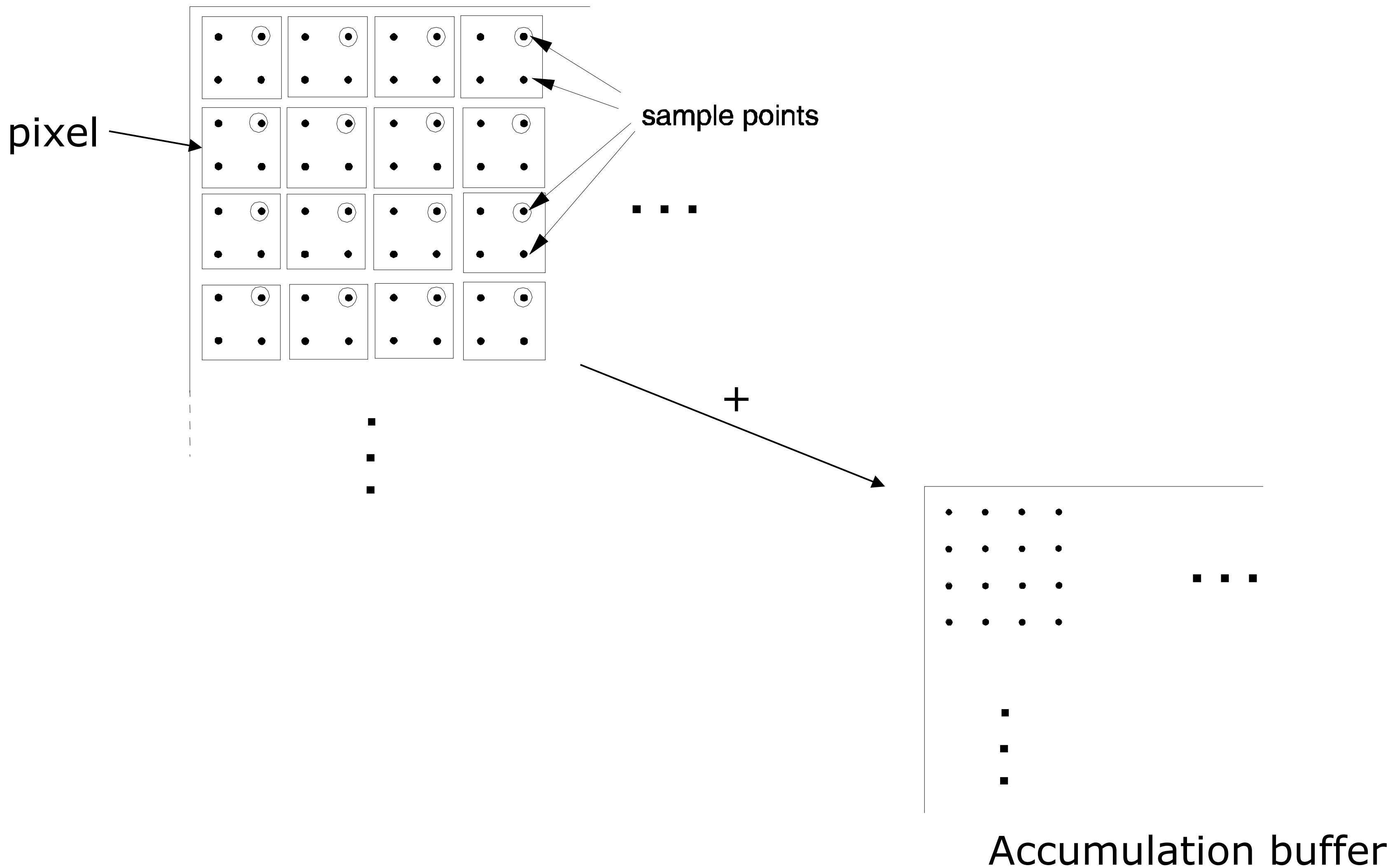
Anti-aliasing in CG

- Accumulation buffer



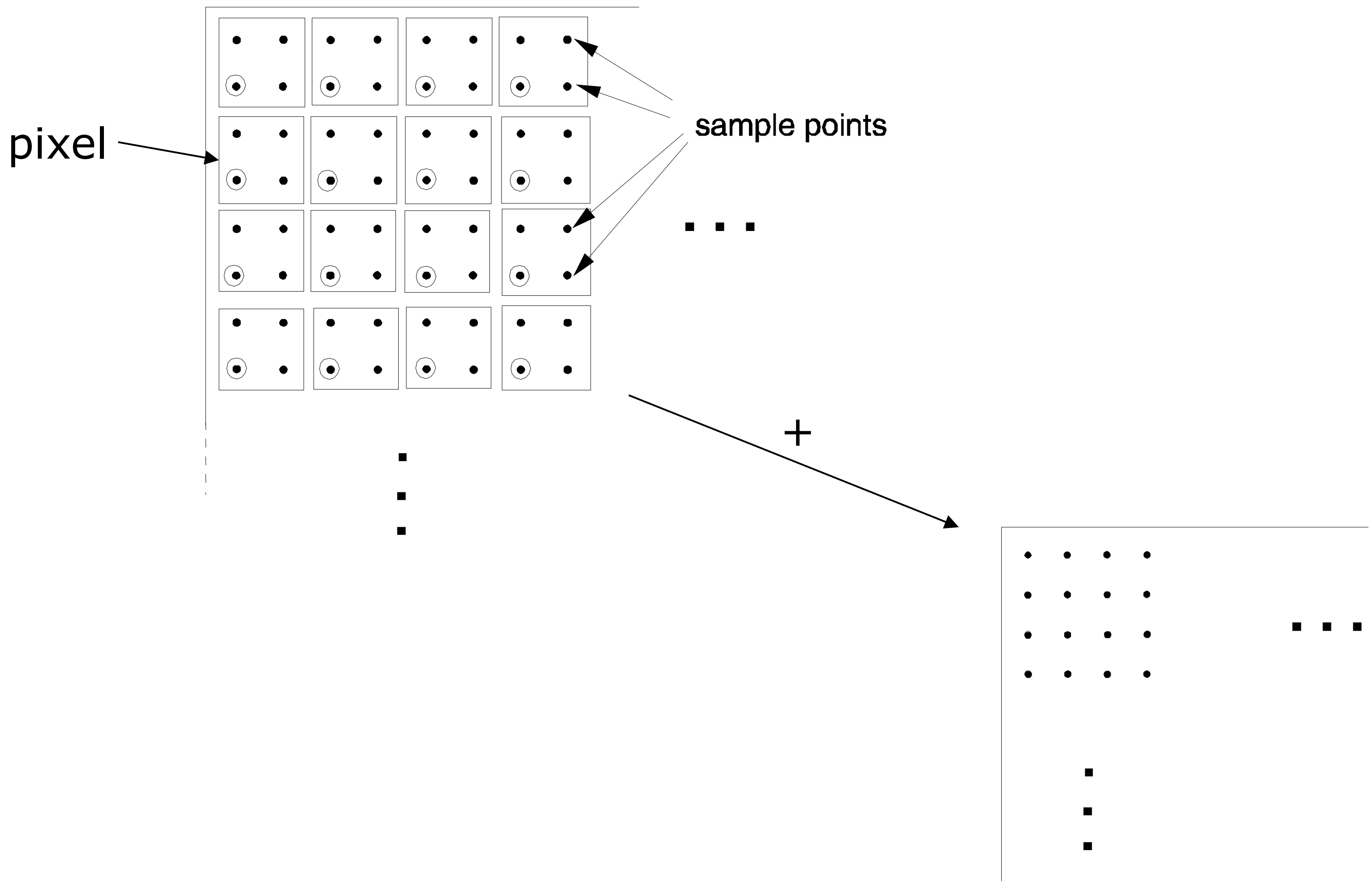
Anti-aliasing in CG

- Accumulation buffer



Anti-aliasing in CG

- Accumulation buffer



Accumulation buffer

Anti-aliasing in CG

- Accumulation buffer
  - Advantage

    -- Only require one more buffer with normal resolution.

    -- Can be integrated into existing graphics hardware easily.

    -- Can simulate motion blur by accumulating multiple images along the time axis.

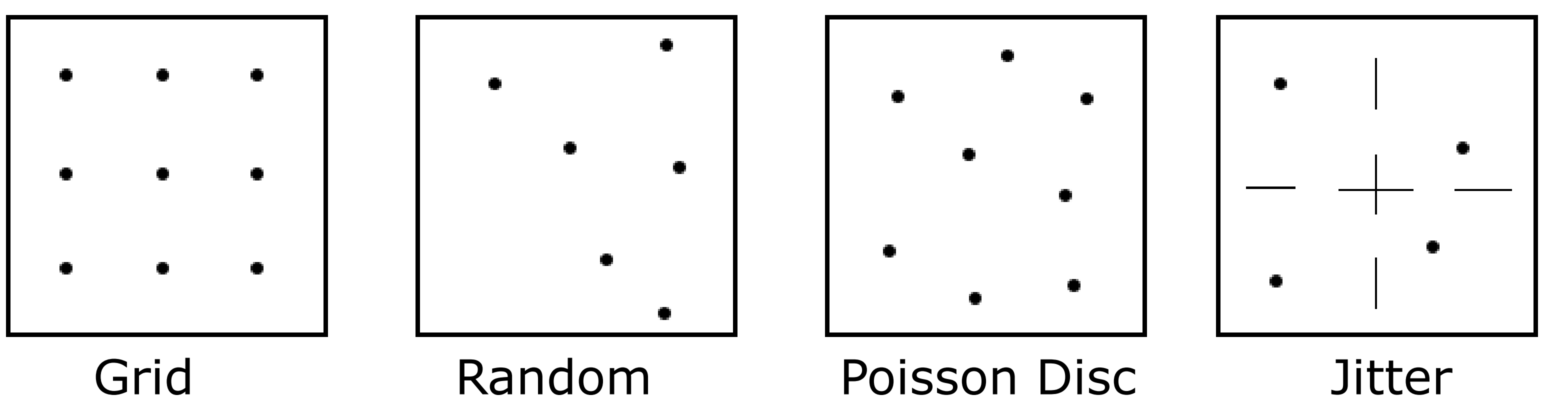
    -- Other effects such as depth of field can also be achieved by changing the camera position at each rendering pass.
  - Disadvantage

    -- Many rendering passes are needed to generate one image.

    -- The accumulation buffer needs to have more bits per pixel in order not to overflow.

Anti-aliasing in CG

- Stochastic sampling
  - An intuitive argument: in stochastic sampling, every region of the image has a finite probability of being sampled. Thus small features that fall between uniform sample points tend to be detected by non-uniform samples
  - This method replaces aliasing with noise by using random sample points in each pixel instead of regular grids as in supersampling.

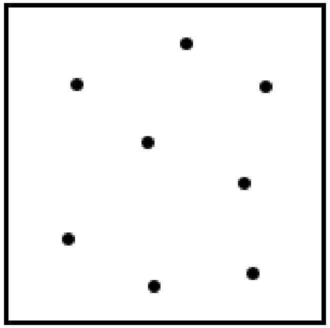


Anti-aliasing in CG

- Stochastic sampling
  - Poisson-disc distribution

  -- Poisson distribution, with minimum-distance constraint between samples

  -- Add points at random, removing again if they are too close to any previous points

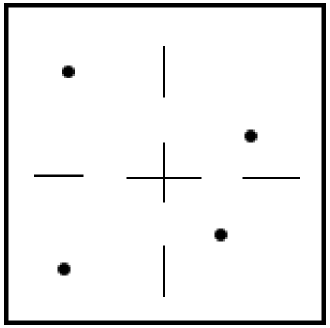


  - Jittered distribution

  -- Start with regular grid of samples

  -- Perturb each sample slightly in a random direction

  -- More “clumpy” or granular in appearance.



Anti-aliasing in CG

- Stochastic sampling
  - Advantage

    -- It can reduce aliasing better with less number of samples per pixel.

    -- With large enough number of samples, it can eliminate aliasing, at least theoretically.

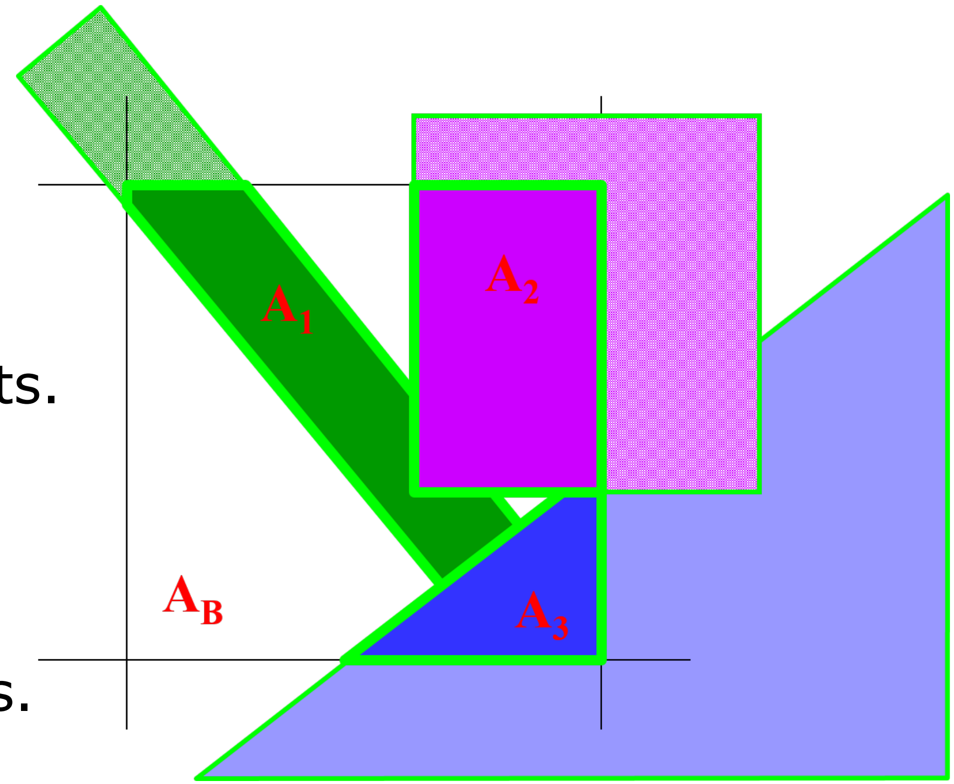
  - Disadvantage

    -- Random noise is added to the image.

    -- It still requires considerable amount of memory and processing time.

Anti-aliasing in CG

- Catmull's algorithm
  - Clip each polygon against each pixel to form polygon fragments.
  - Determine visible fragments.
  - Find fragment areas.
  - Multiply by fragment colors.
  - Sum for final pixel color.



Anti-aliasing in CG

- Catmull's algorithm
  - Advantage

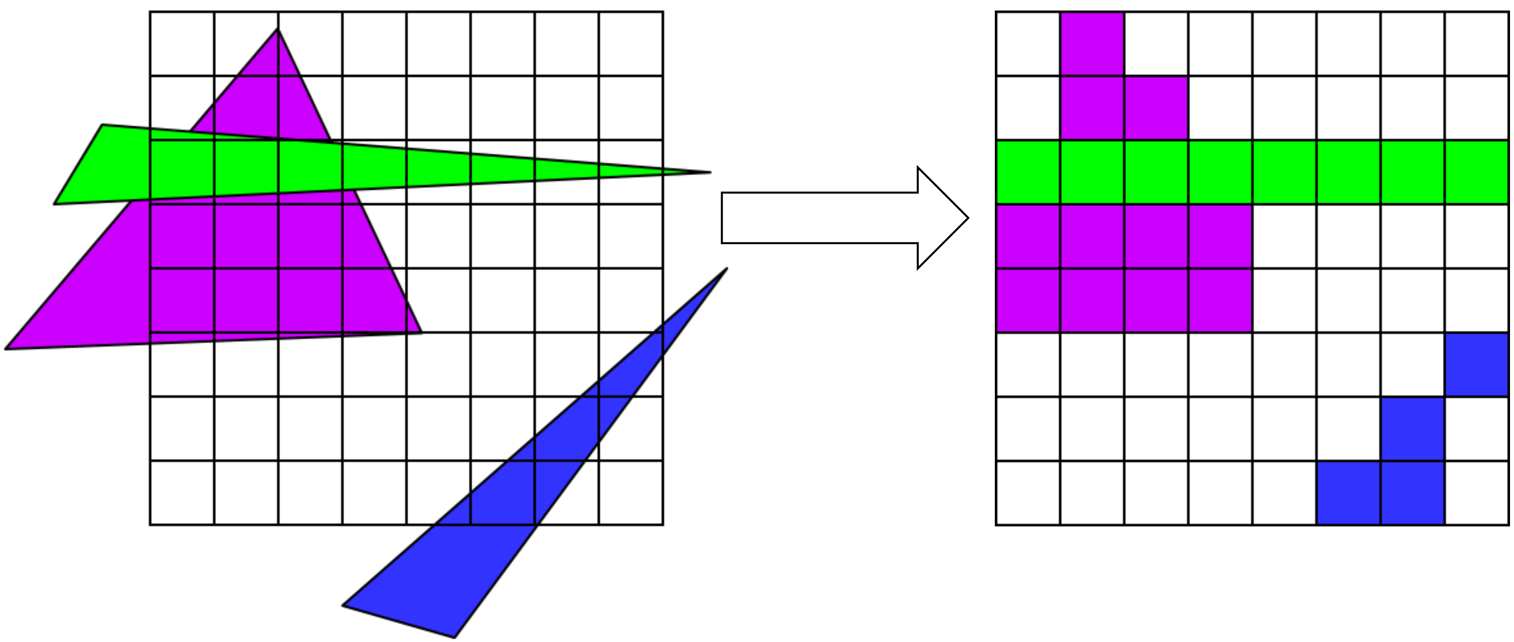
    -- It calculates exactly how much a polygon covers a pixel, i.e., its color contribution to the pixel.

  - Disadvantage

    -- Computationally very expensive due to polygon clipping and sorting polygon fragments by depth.

Anti-aliasing in CG

- The A-buffer method
  - This method uses subpixel sampling to make summing areas simple.



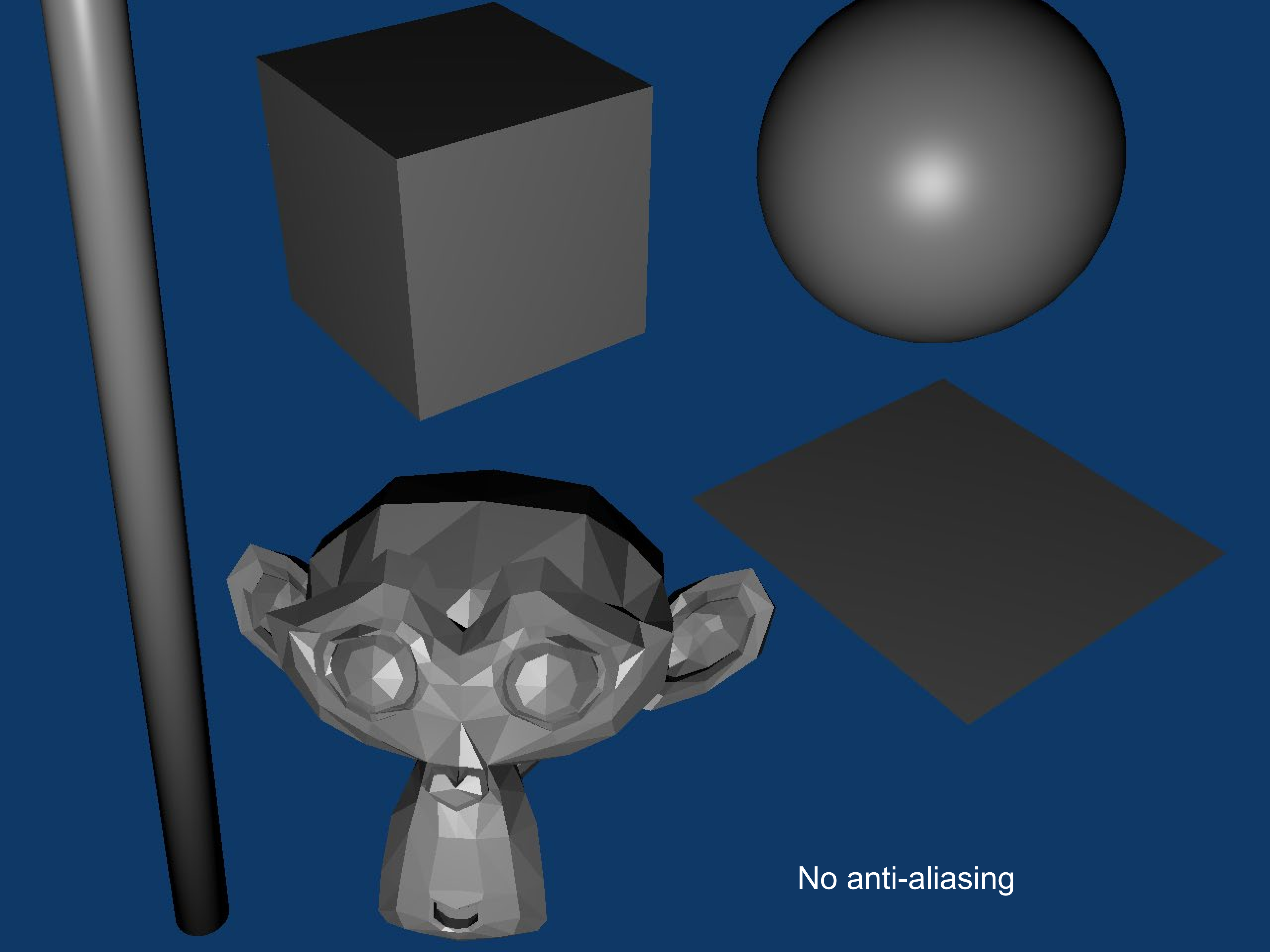
Anti-aliasing in CG

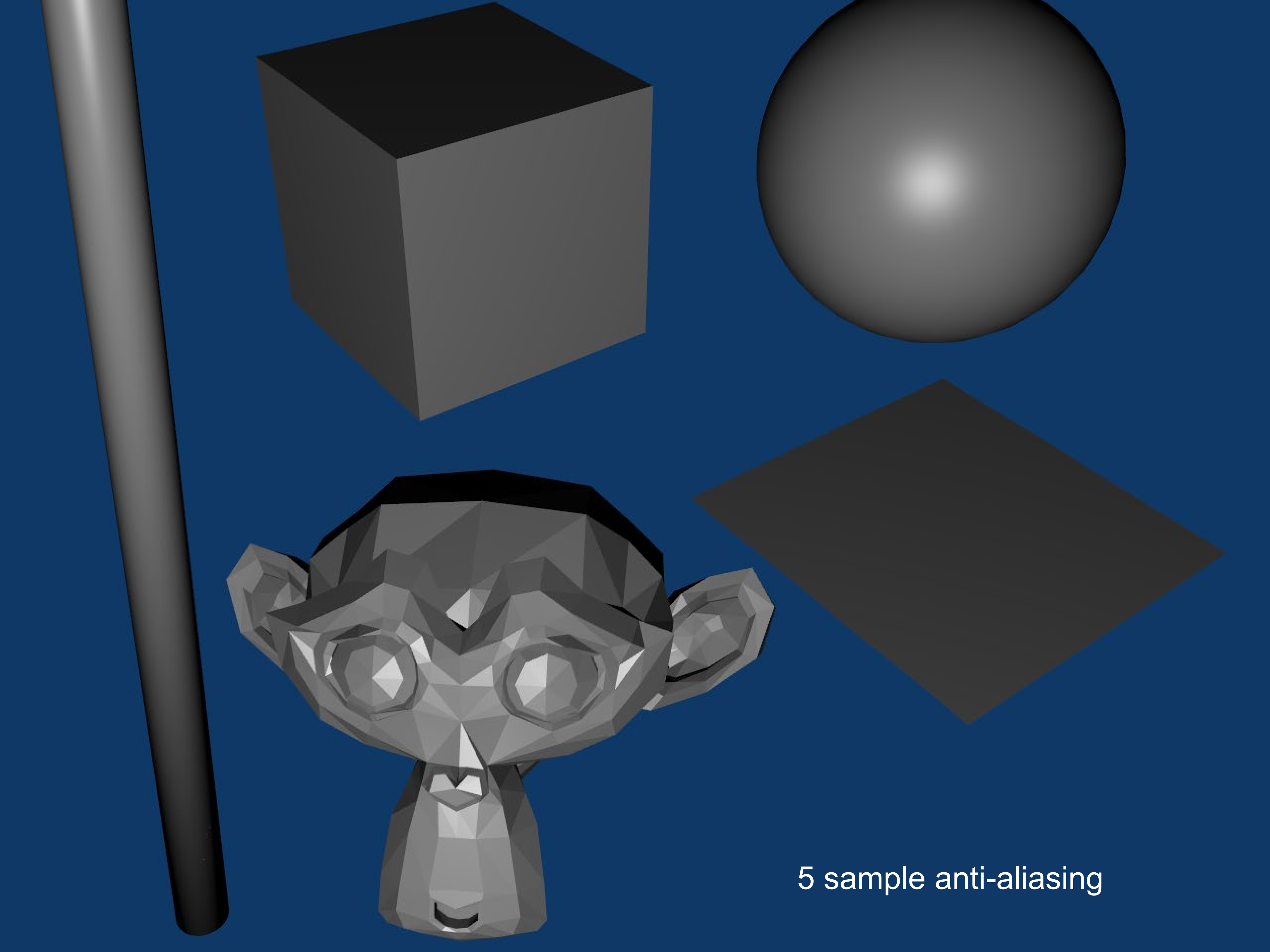
- The A-buffer method
  - Advantage

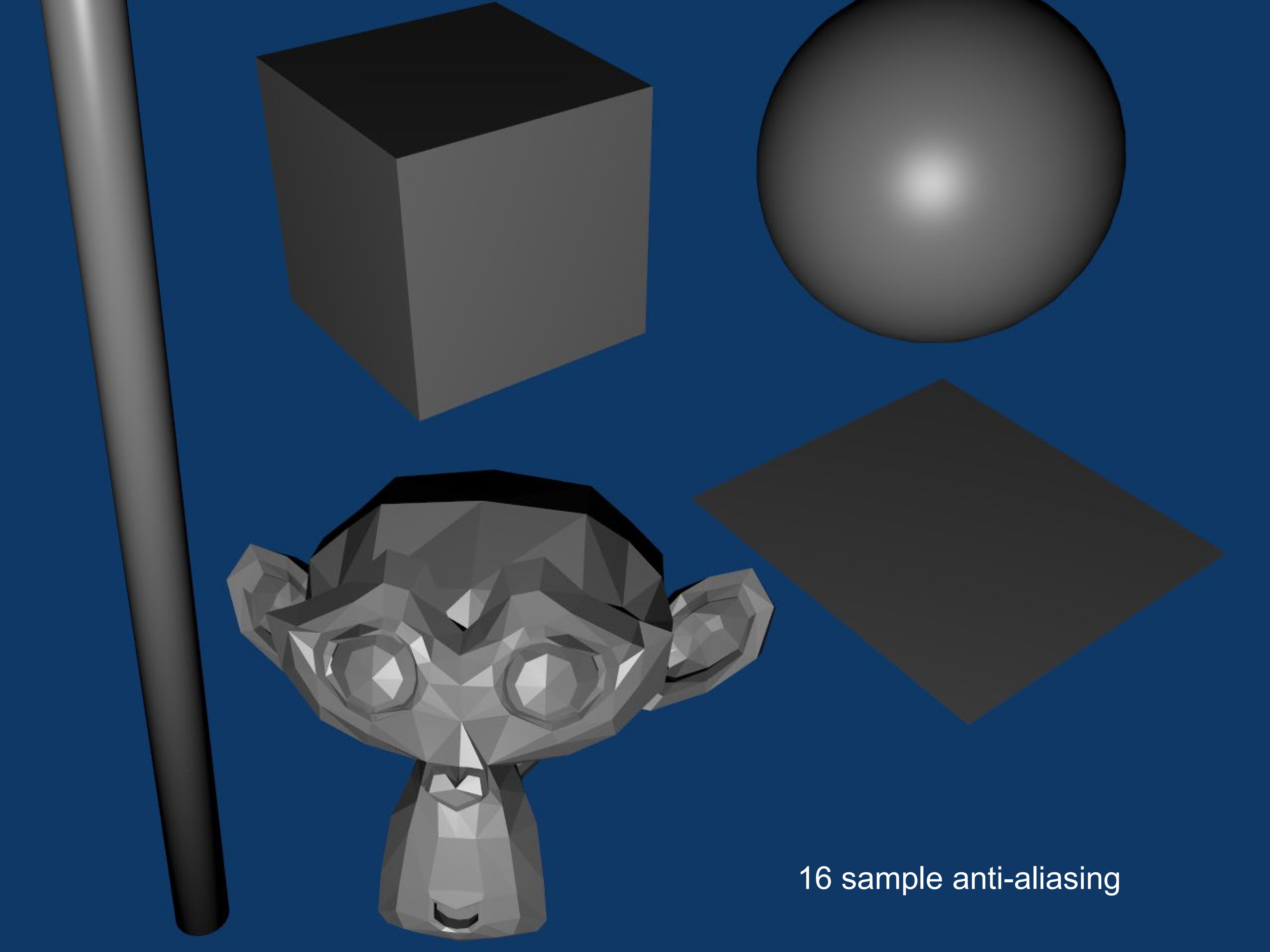
    -- Processing per pixel depends only on number of visible fragments

    -- Can be implemented efficiently using bitwise logical ops on subpixel masks
  - Disadvantage

    -- Still basically a supersampling algorithm


No anti-aliasing


5 sample anti-aliasing


16 sample anti-aliasing