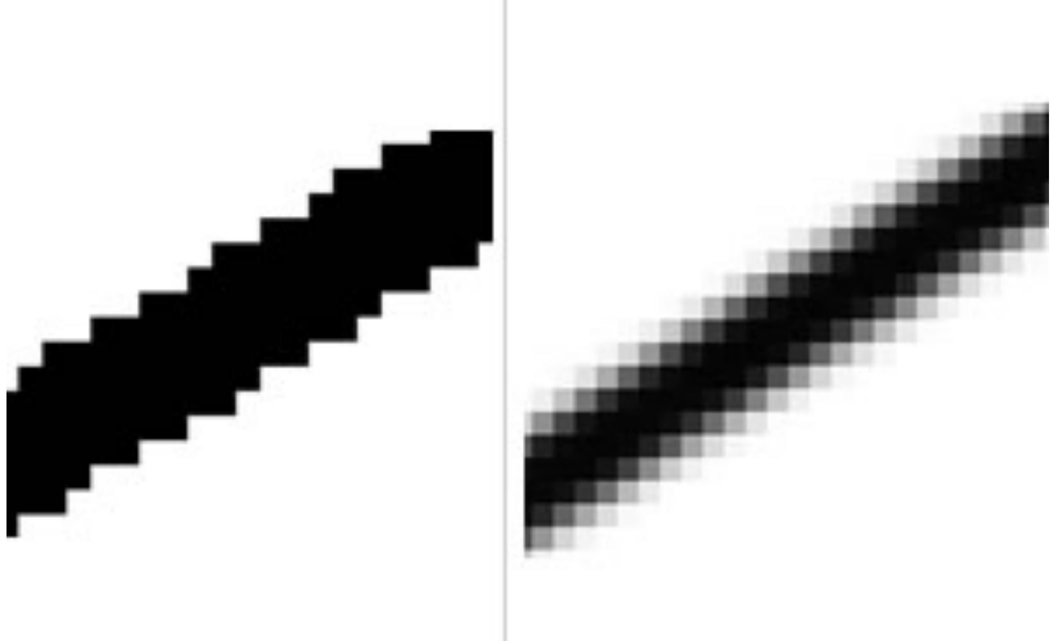
28 – SAMPLING. ALIASING AND ANTI-ALIASING

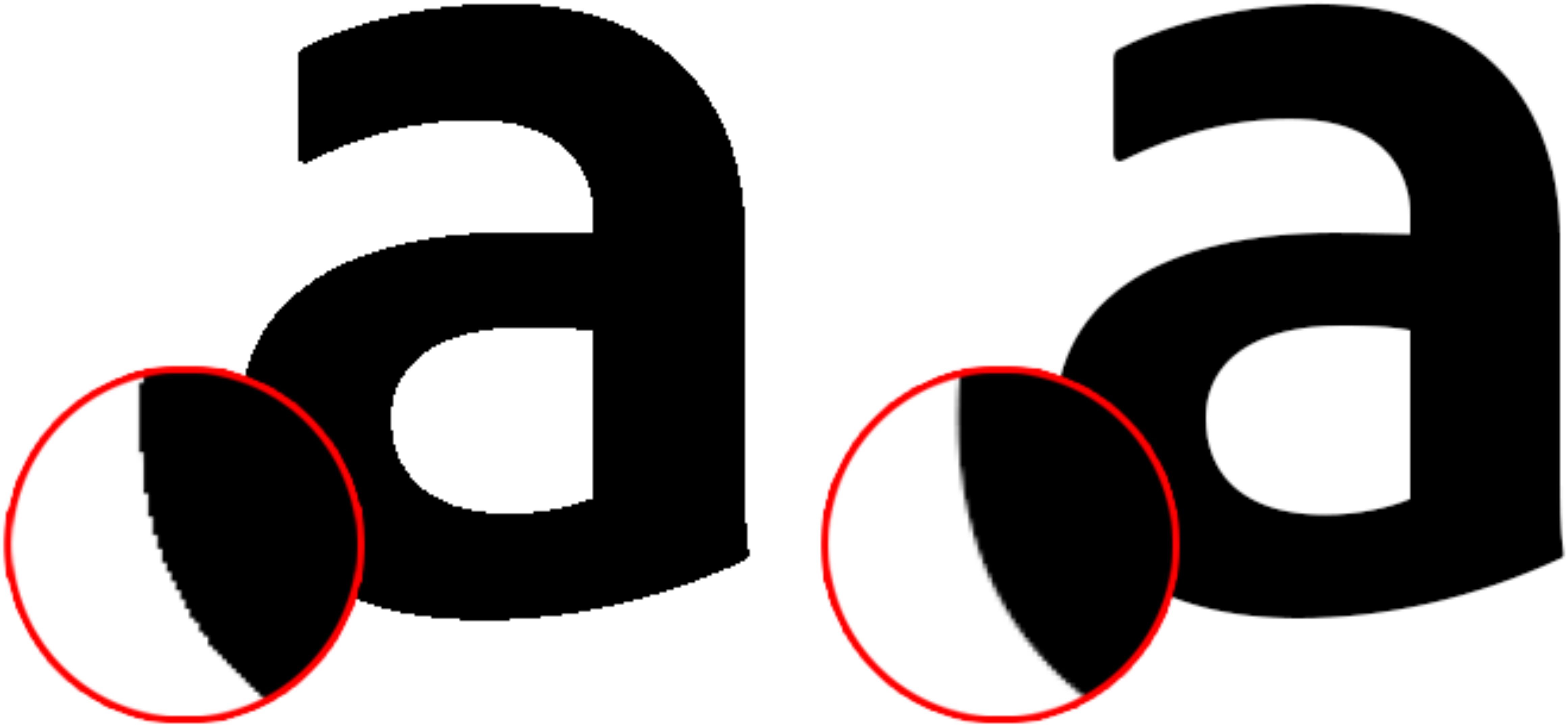
Textbook: 16



UGRAD.CS.UBC.CA/~CS314

Mikhail Bessmeltsev

ALIASING & ANTI-ALIASING

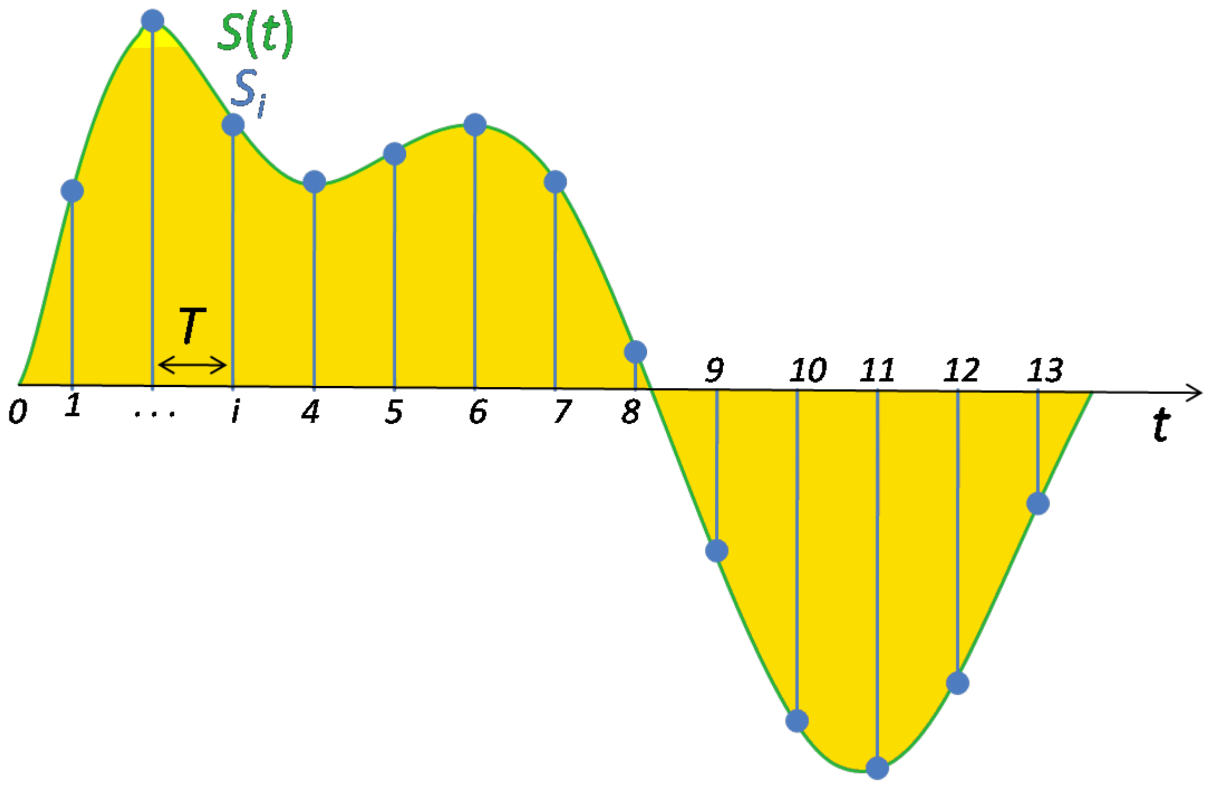


TWO VIEWS OF IMAGES

- A *continuous image*, $I(x_w, y_w)$, is a bivariate function.
  - range is a linear color space.
- A *discrete image* I[i][j] is a two dimensional array of color values.
- We associate each pair of integers i, j, with the continuous image coordinates $x_w = i$ and $y_w = j$

CONTINUOUS VS. DISCRETE

- Continuous -> Discrete: ***Sampling***
- Discrete -> Continuous: ***Reconstruction/Interpolation***



SAMPLING

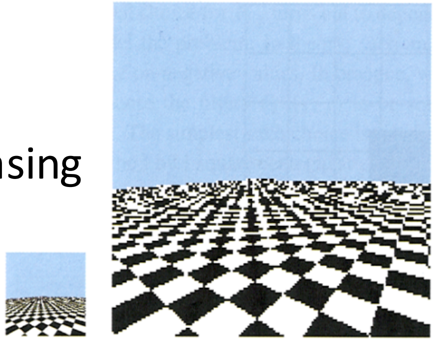
- The simplest and most obvious method to go from a continuous to a discrete image is by *point sampling*.
- To obtain the value of a pixel i, j, we sample the continuous image function at a single integer valued domain location:

$$\mathrm{I[i][j]} \leftarrow I(i, j)$$

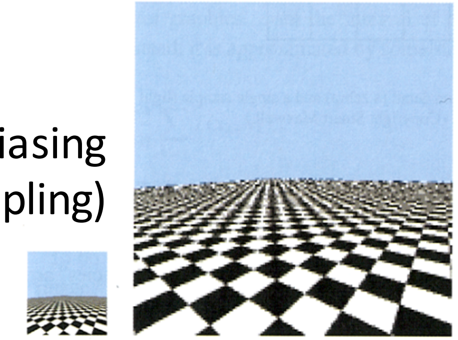
- This can results in unwanted artifacts.

ALIASING AND ANTI-ALIASING

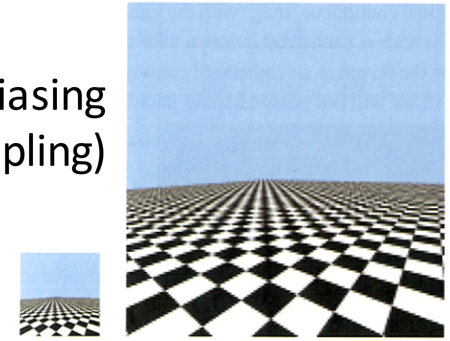
Aliasing



Anti-aliasing
(multi-sampling)



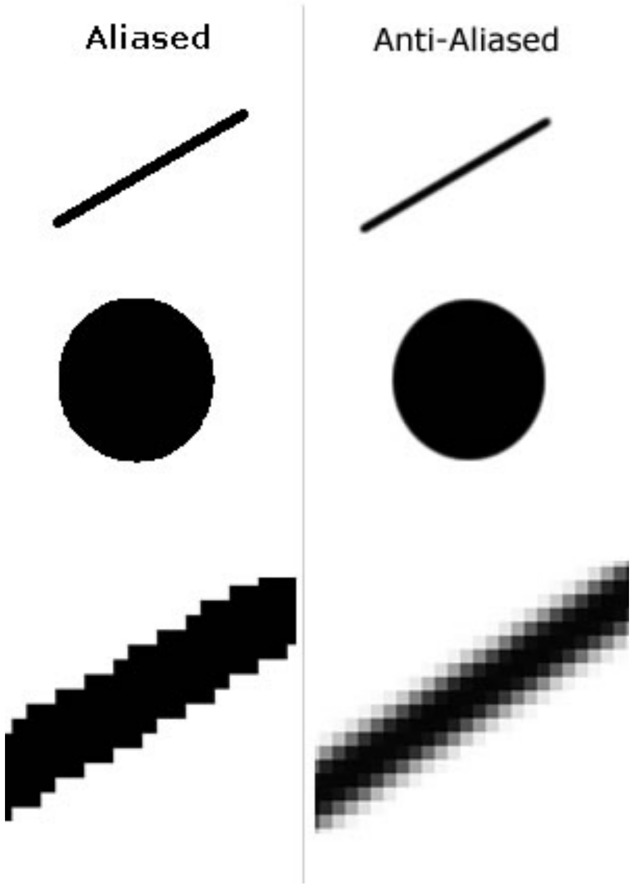
Anti-aliasing
(super-sampling)



ALIASING

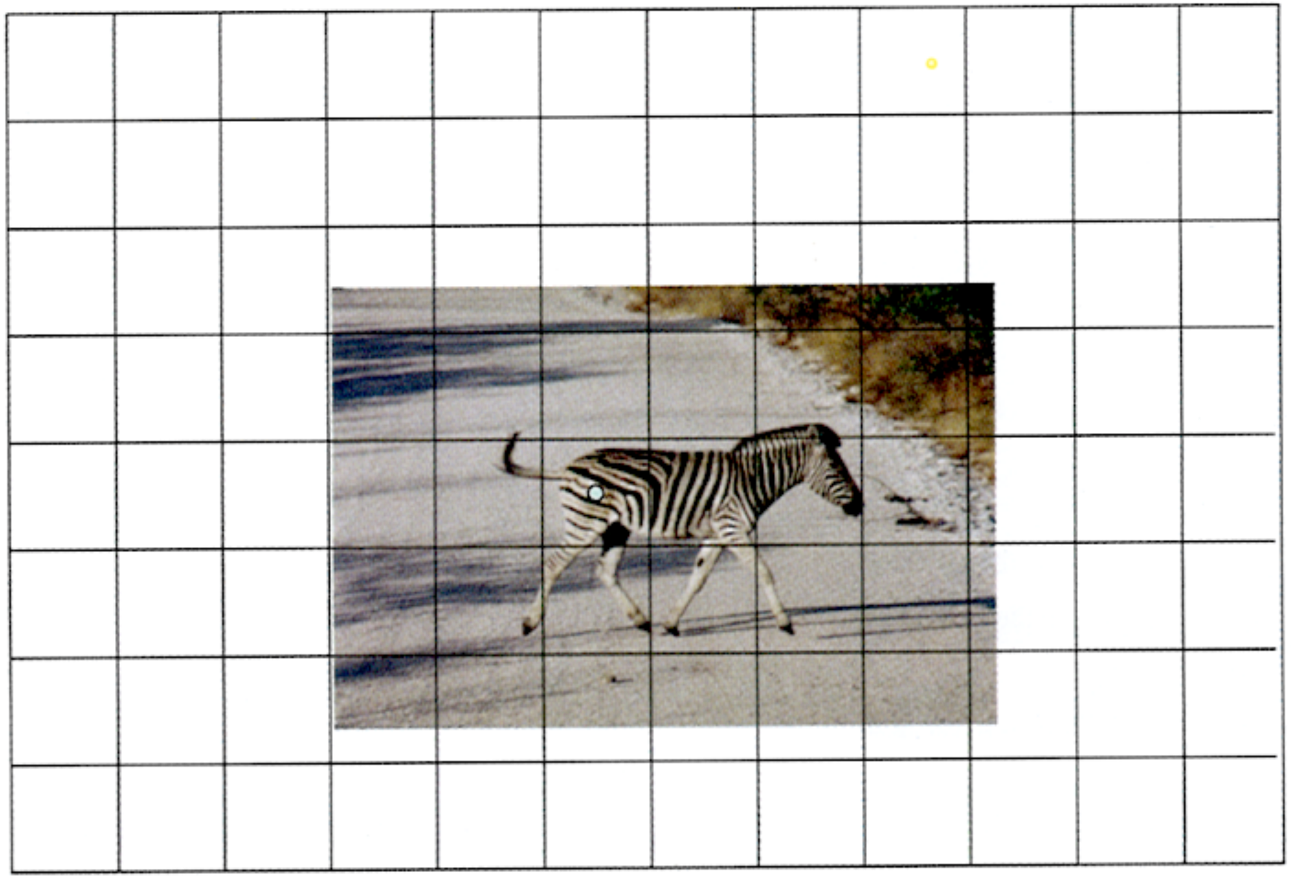
- Scene made up of black and white triangles: jaggies at boundaries
  - Jaggies will crawl during motion

- If triangles are small enough then we get random values or weird patterns.
  - Jaggies will crawl during motion



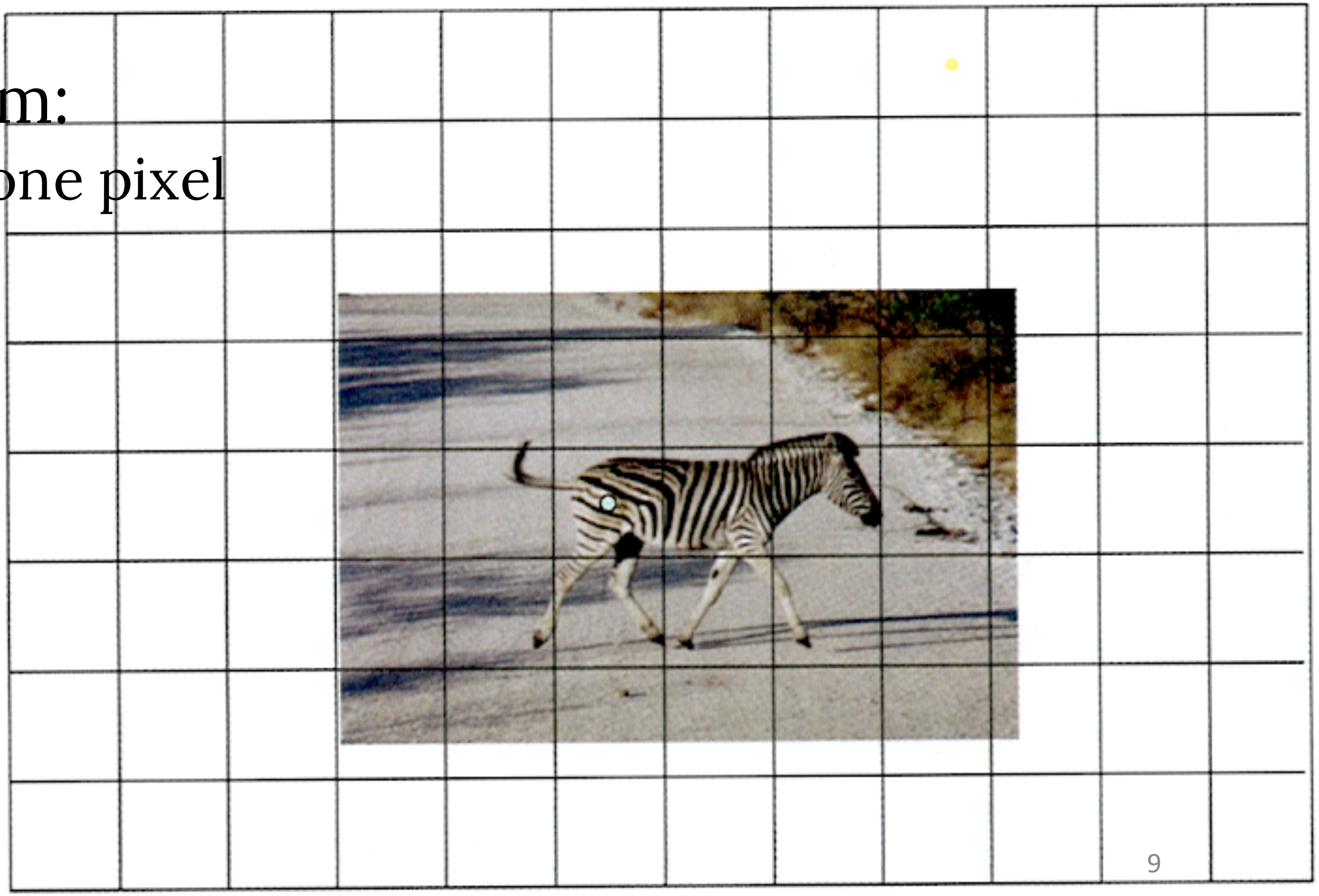
ALIASING

- Aliasing happens when hi-res image is drawn on low-res media



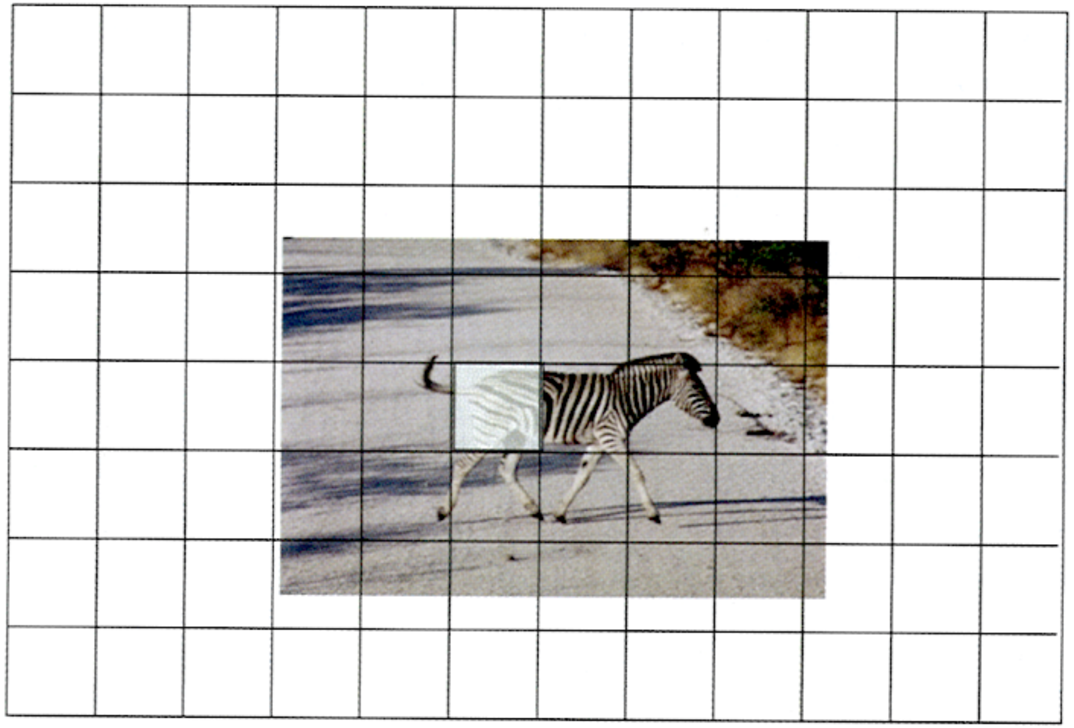
ALIASING

- Aliasing happens when hi-res image is drawn on low-res media
- The heart of the problem:
  too much information in one pixel



ANTI-ALIASING

- Intuitively:
  - single sample is a bad value
  - should use some kind of average value over some appropriate region.
- In the above examples, perhaps some gray value.



ANTI-ALIASING

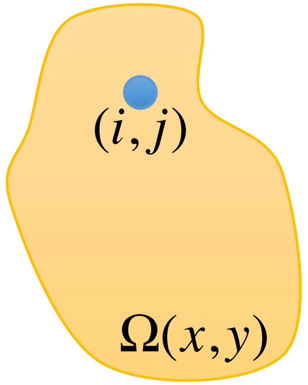
- Mathematically this can be modeled using *Fourier analysis*.
  - Breaks up the data by “frequencies” and figures out what to do with the un-representable high frequencies.

ANTI-ALIASING

- We can also model this as an optimization problem.
- These approaches lead to:

$$\mathrm{I[i][j]} \leftarrow \iint_{\Omega} I(x,y)F_{i,j}(x,y)\,dxdy$$

where $F_{i,j}(x,y)$ is some function measuring how strongly the continuous image value at $[x,y]^t$ should influence the pixel value $\mathrm{i,j}$



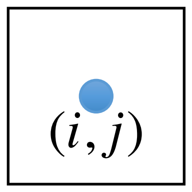
ANTI-ALIASING

- In this setting, the function $F_{i,j}(x,y)$ is called *a filter*.
    - In other words, the best pixel value is determined by performing some continuous weighted averaging near the pixel's location.
    - Effectively, this is like blurring the continuous image before point sampling it.

BOX FILTER

- We often choose the filters $F_{i,j}(x,y)$ to be something non-optimal, but that can more easily computed with.
- The simplest such choice is a *box filter*, where $F_{i,j}(x,y)$ is zero everywhere except over the 1-by-1 square center at $x=i, y=j$.

- Calling this square $\Omega_{i,j}$, we arrive at

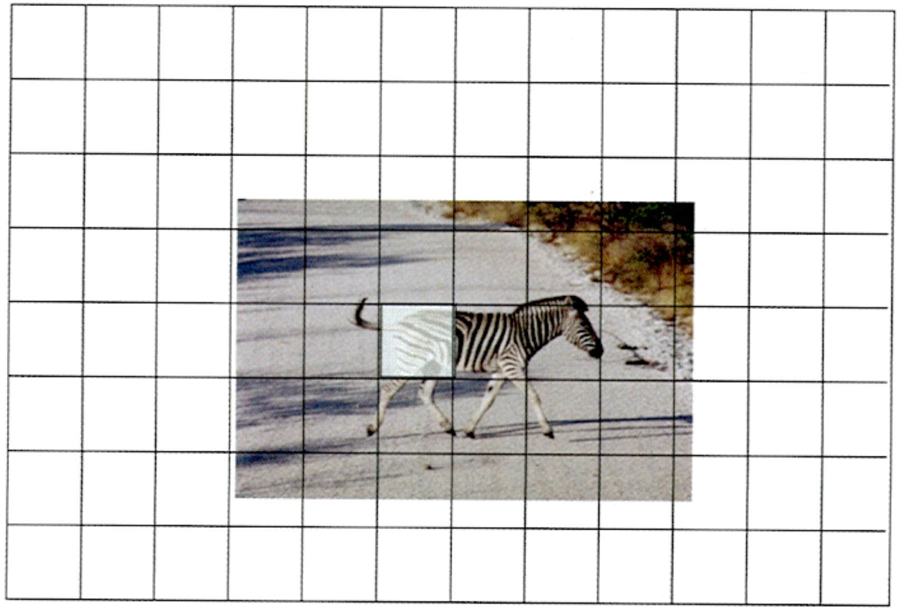


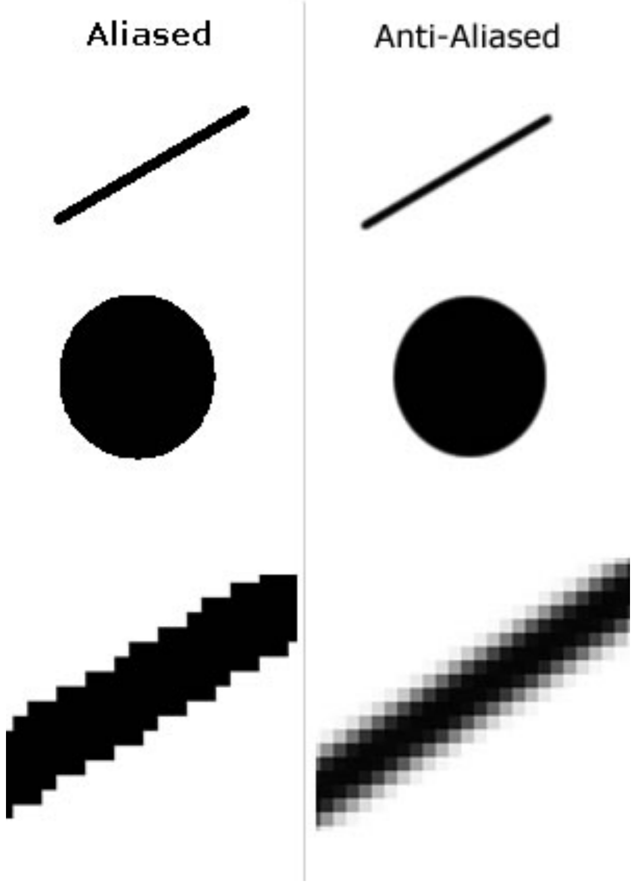
$\Omega(x,y)$

$$\mathrm{I[i][j]} \leftarrow \iint_{\Omega_{i,j}} I(x,y)dxdy$$

BOX FILTER

- In this case, the desired pixel value is simply the average of the continuous image over the pixel’s square domain.

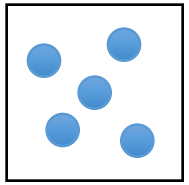




OVER-SAMPLING

- Even that integral is not really easy to compute
- Instead, it is approximated by some sum of the form:

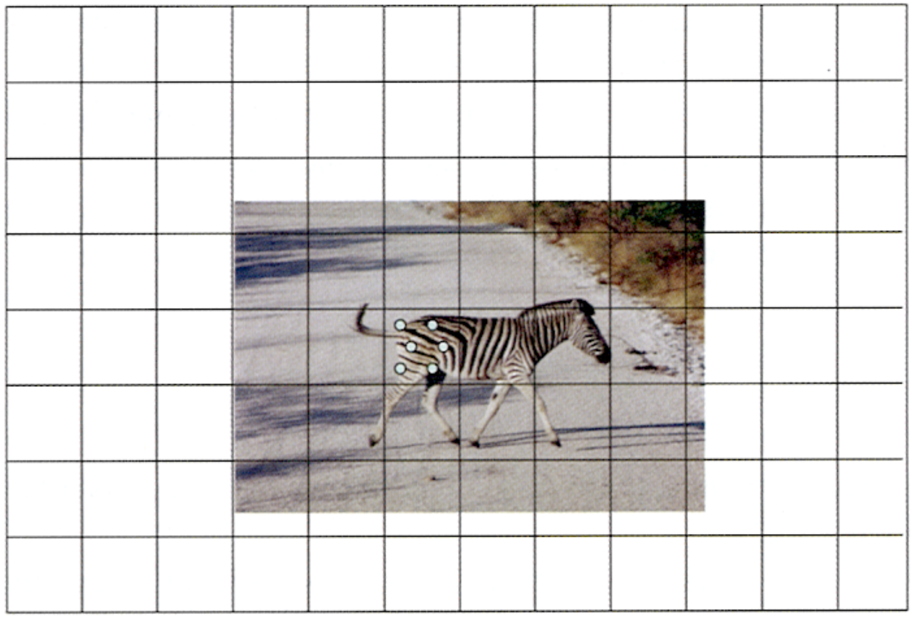
$$\mathrm{I[i][j]} \leftarrow \frac{1}{n} \sum_{k=1}^{n} I(x_k, y_k)$$



- where *k* indexes some set of locations (x_k, y_k) called the sample locations.
- The renderer first produces a "high resolution" color and z-buffer "image",
  - where we will use the term *sample* to refer to each of these high resolution pixels.

OVER-SAMPLING

- Then, once rasterization is complete, groups of these samples are averaged together, to create the final lower resolution image.



SUPER-SAMPLING

- If the sample locations for the high resolution image form a regular, high resolution grid, then this is called *super sampling.*
- We can also choose other sampling patterns for the high resolution "image",
  - Such less regular patterns can help us avoid systematic errors that can arise when using the sum to replace the integral.

MULTI-SAMPLING

- Render to a "high resolution" color and z-buffer
- *During the rasterization* of each triangle, "coverage" and z-values are computed at this sample level.
- But for efficiency, the fragment shader is only called **only once per final resolution pixel.**
  - This color data is shared between all of the samples hit by the triangle in a single (final resolution) pixel.
- Once rasterization is complete, groups of these high resolution samples are averaged together.

MULTI-SAMPLING

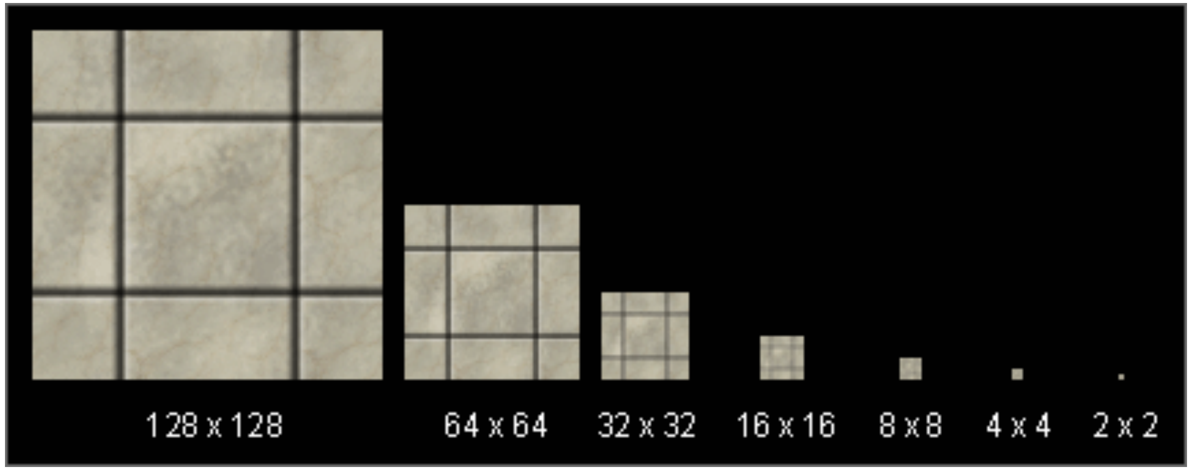
- Why is MS effective?

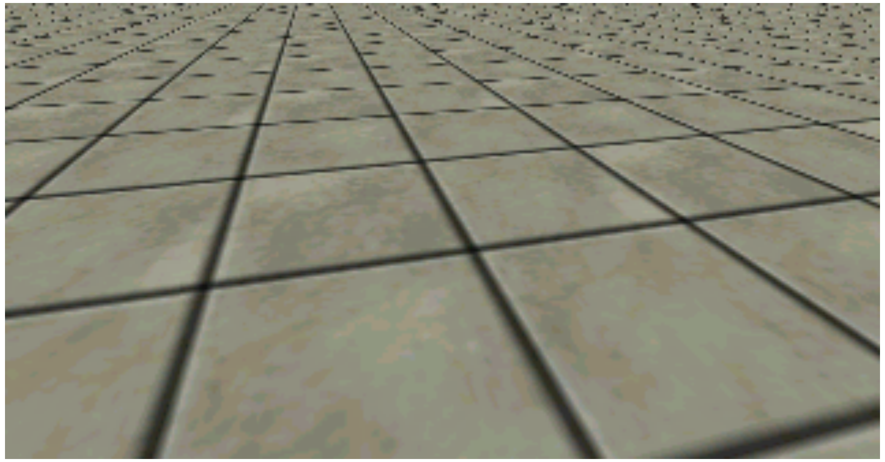
- Colors tend to vary quite slowly over each triangle
  - => no need to be computed at high spatial resolution
  - not true for textures

- For textures: pre-process the texture image itself
  - Our mipmaps!

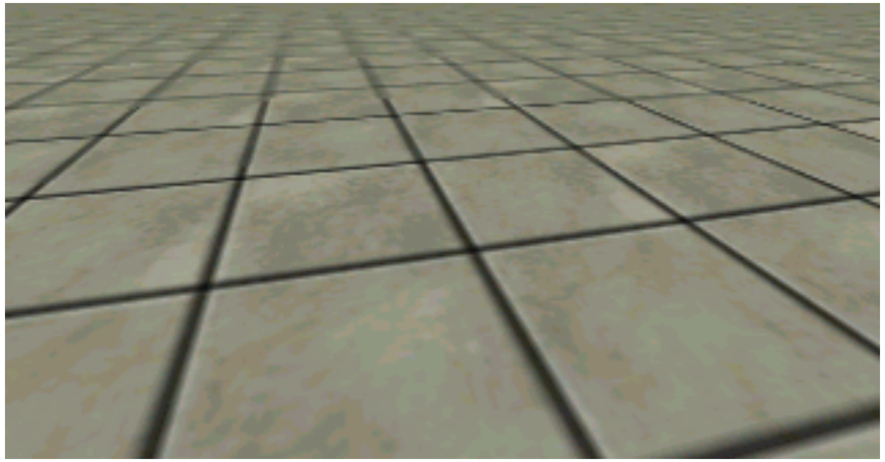
MIPMAPPING

use “image pyramid” to precompute averaged versions of the texture





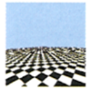
Without MIP-mapping

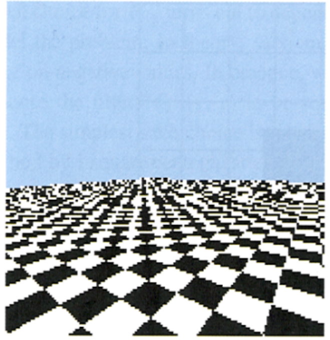


With MIP-mapping

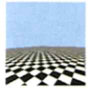
ALIASING AND ANTI-ALIASING

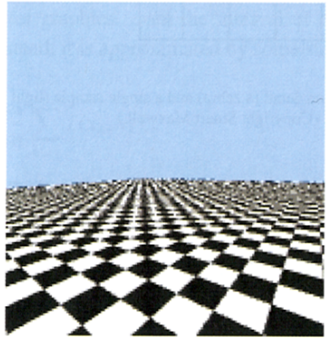
Aliasing



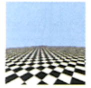


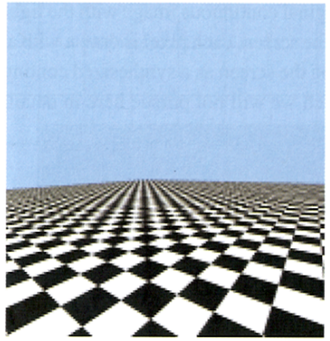
Anti-aliasing
(multi-sampling)





Anti-aliasing
(super-sampling)





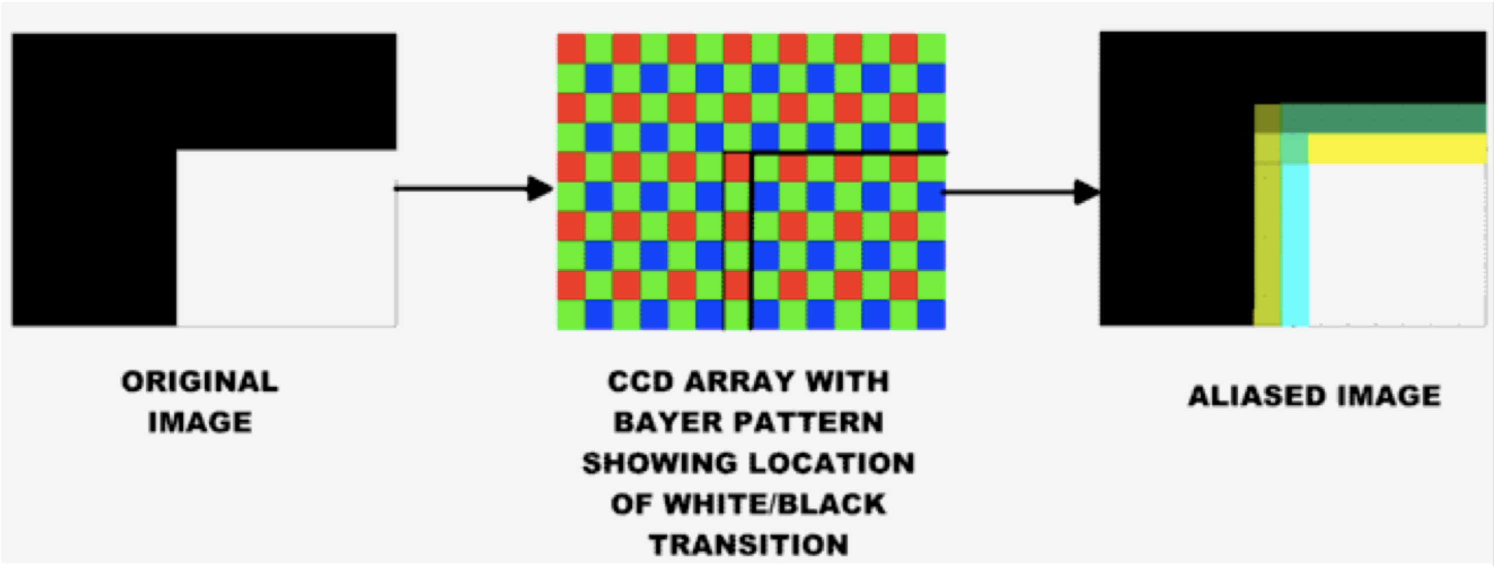
IN DIGITAL CAMERAS:

- Anti-aliasing is
  - spatial integration that happens over the extent of each pixel sensor
  - optical blurring (due to the lens).

  - sometimes extra additional optical elements specifically to blur the continuous image data before it is sampled at the sensors.

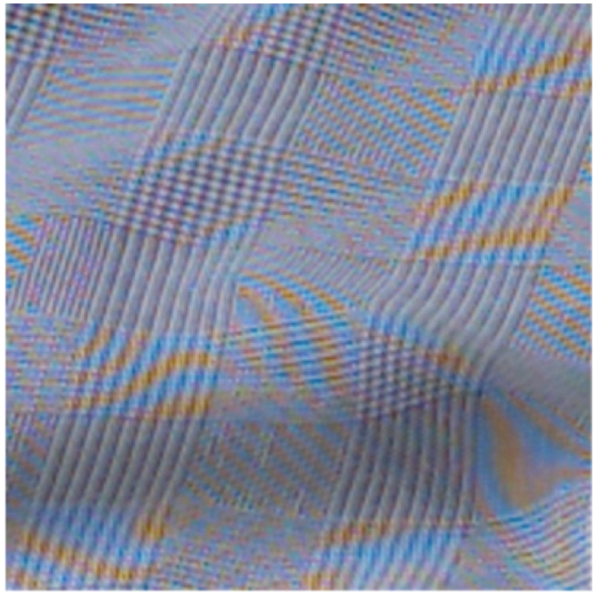
CAMERA: DEMOSAICING PROBLEM

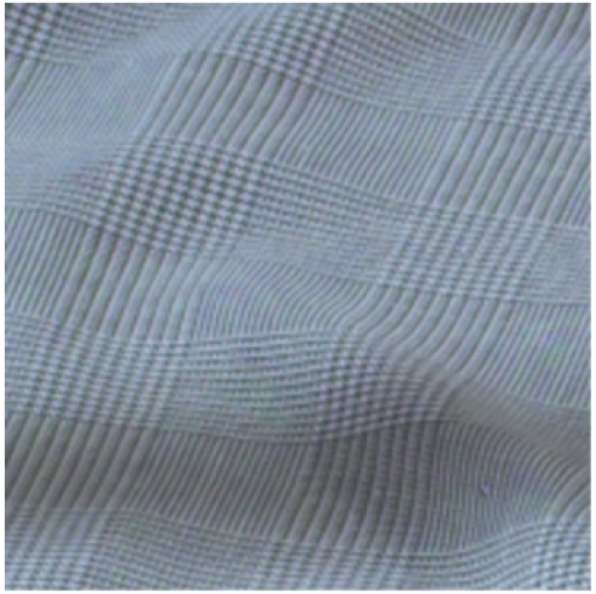
- Imagine a black-on-white corner
- Color aliasing happens

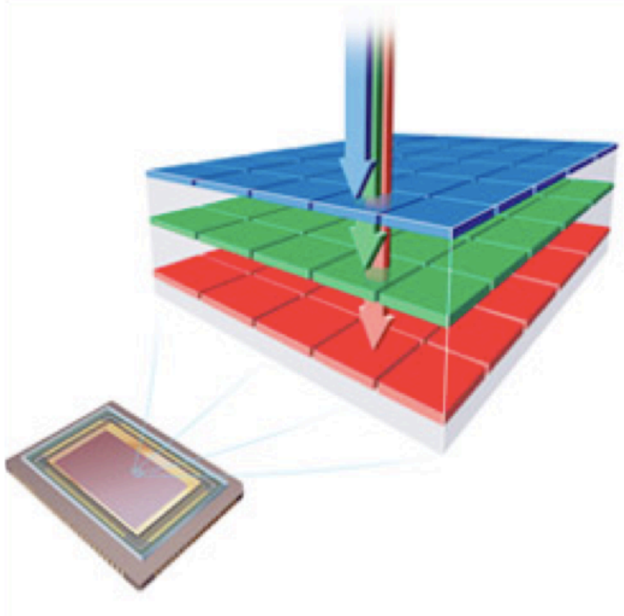


CAMERA: FOVEON SENSOR

- Transparent layers of RGB
- No demosaicing → no moiré artifact







HUMAN VISION

- In human vision, aliasing artifacts are not typically encountered.
  - Most of the antialiasing, at least in the foveal (central) region of vision, is due to the optical blurring of light, which happens well before it hits the receptor cells.
  - The irregular spatial layout of the sensor cells in the retina also helps by effectively providing spatial jitter (randomness) which turns noticeable aliasing into less conspicuous noise.