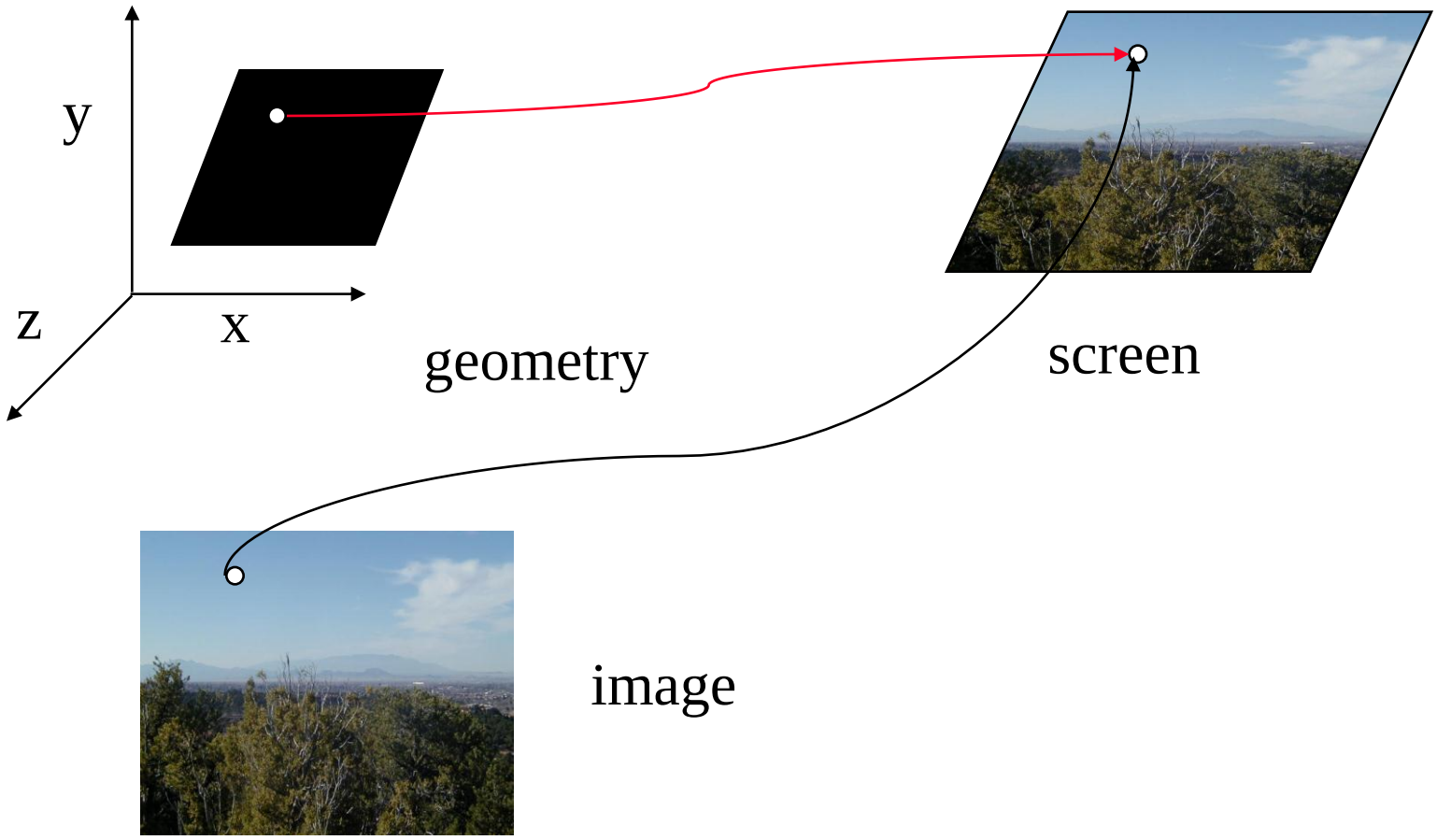
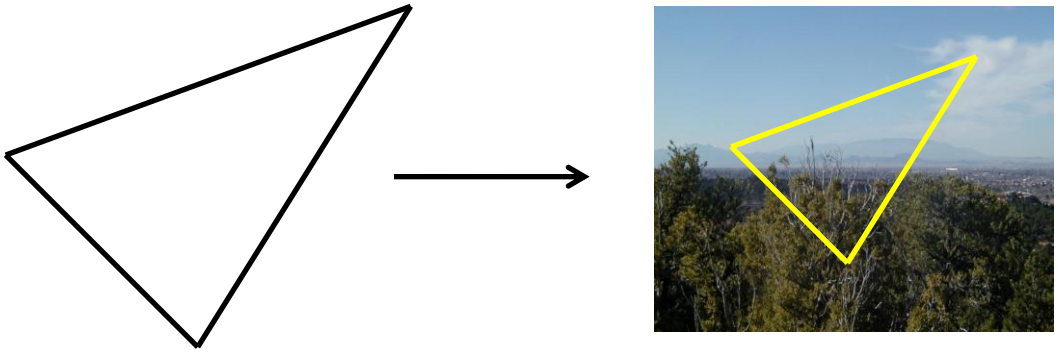
Texture mapping

Texture slides are based on E. Angel's slides



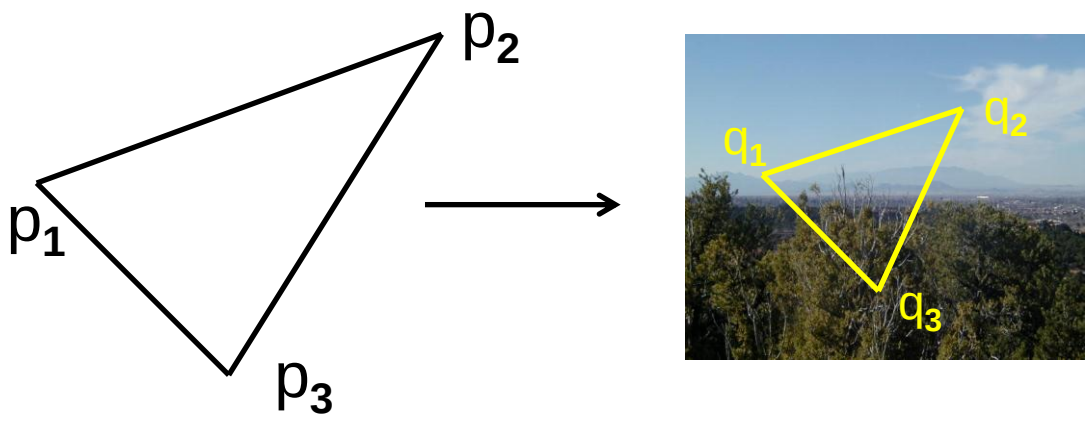
Texture mapping triangles



triangle in 3d **triangle in texture domain**

- Each vertex gets (s,t) coordinates in the texture plane
- image dimensions are always 1x1 in texture coordinates
- s,t are typically in 0..1 range but do not have to be

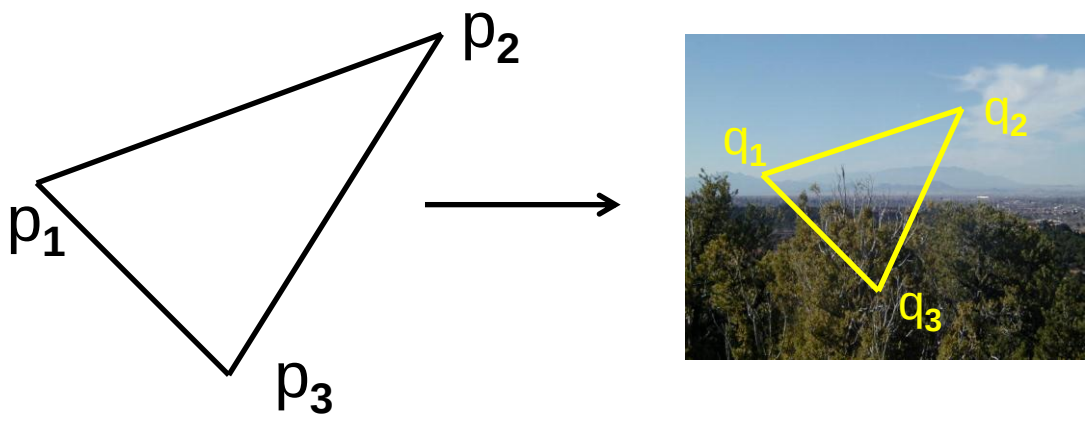
Texture mapping triangles



triangle in 3d **triangle in texture domain**

- Barycenteric coordinates: write each point inside the triangle as an affine combination of vertices
- $p = \alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3$, $\alpha_1 + \alpha_2 + \alpha_3 = 1$
- assign to p texture coordinate $q = \alpha_1 q_1 + \alpha_2 q_2 + \alpha_3 q_3$

Texture mapping triangles



triangle in 3d **triangle in texture domain**

Computing α_i

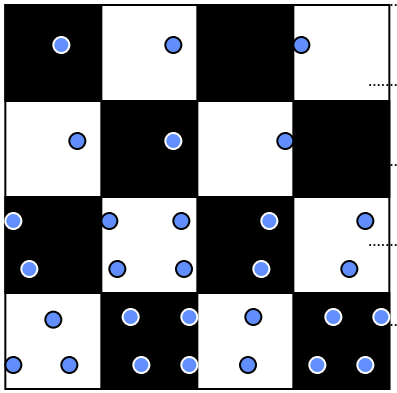
$p = \alpha_1 (p_1 - p_3) + \alpha_2 (p_2 - p_3) + p_3$, i.e.,

$p - p_3 = \alpha_1 (p_1 - p_3) + \alpha_2 (p_2 - p_3)$; taking dot product with $(p_2 - p_3)^\perp$

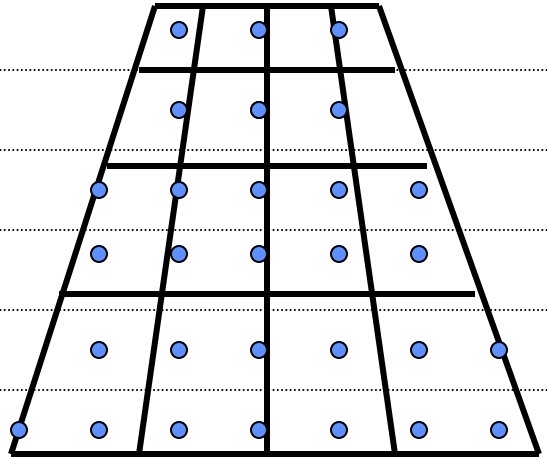
we get $\alpha_1 = (p - p_3) \cdot (p_2 - p_3)^\perp / (p_2 - p_3) \cdot (p_2 - p_3)^\perp$

similarly we can get the rest

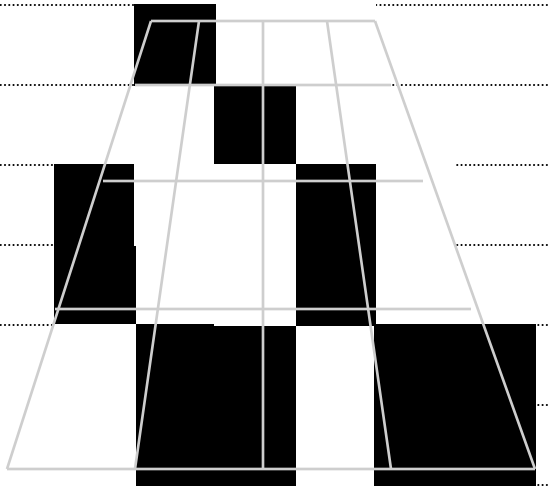
Sampling texture maps



Texture map



Polygon far from the viewer
in perspective projection



Rasterized and textured

the back row is a very poor representation of the true image

How computer images work?

Continuous real image

Digitization
(e.g. scanning)

square array of numbers
(abstract pixels)

each physical pixel
covers an area

display (physical pixels)

The eye blurs pixels into
continuous image

perceived image

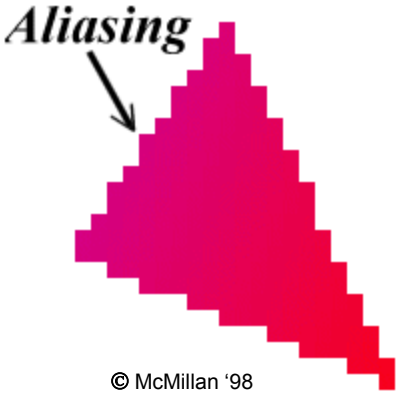
What can go wrong?

Pipeline: sample - process - reconstruct

All kinds of artifacts can appear

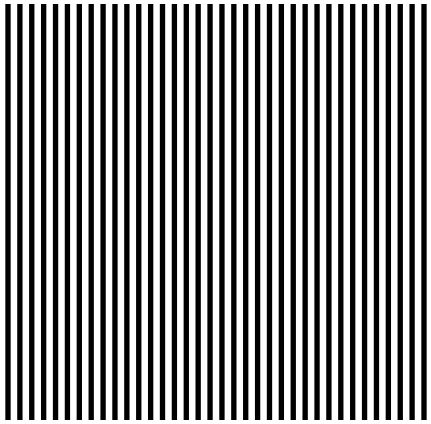
- **jaggies**
- **alias patterns**
- **moire patterns**
- **temporal aliasing (wheels going wrong way)**

Question: how do we avoid all this?

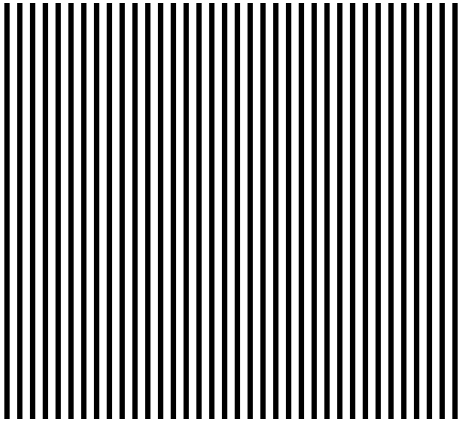


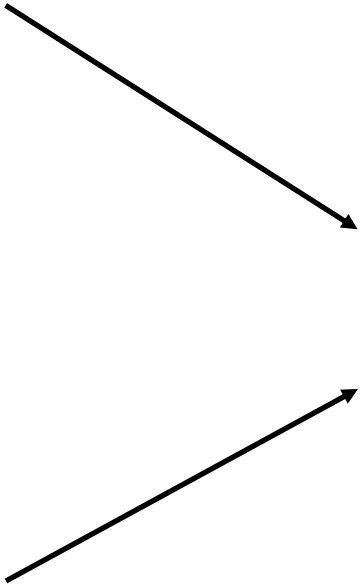
© McMillan '98

Aliasing



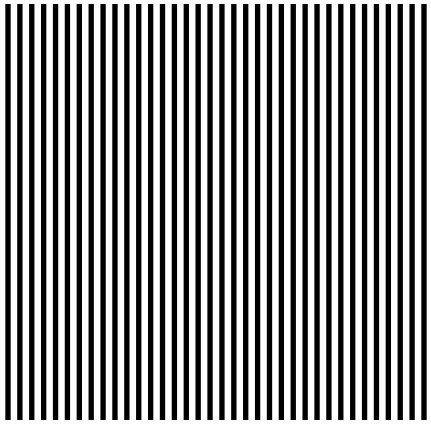
Slightly different frequency



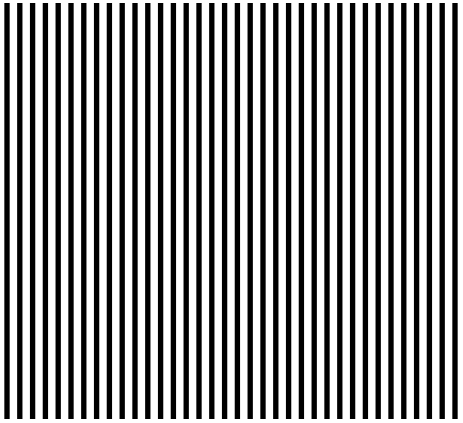


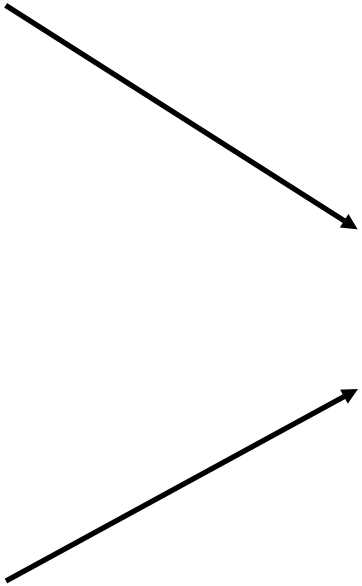
What can go wrong?

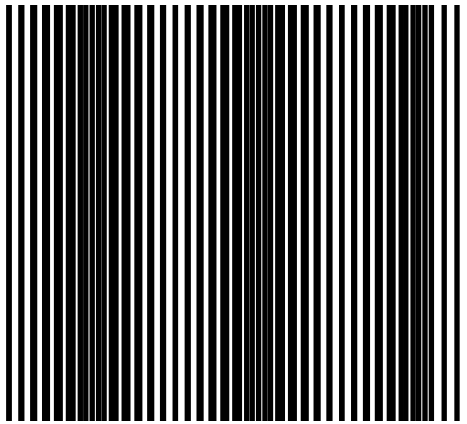
Aliasing



Slightly different frequency



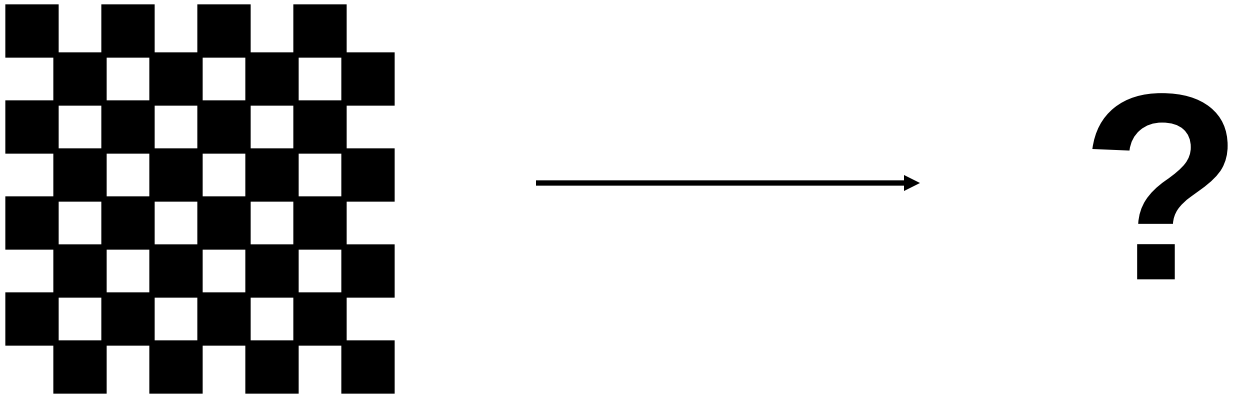




Lower frequency appears

Shrinking

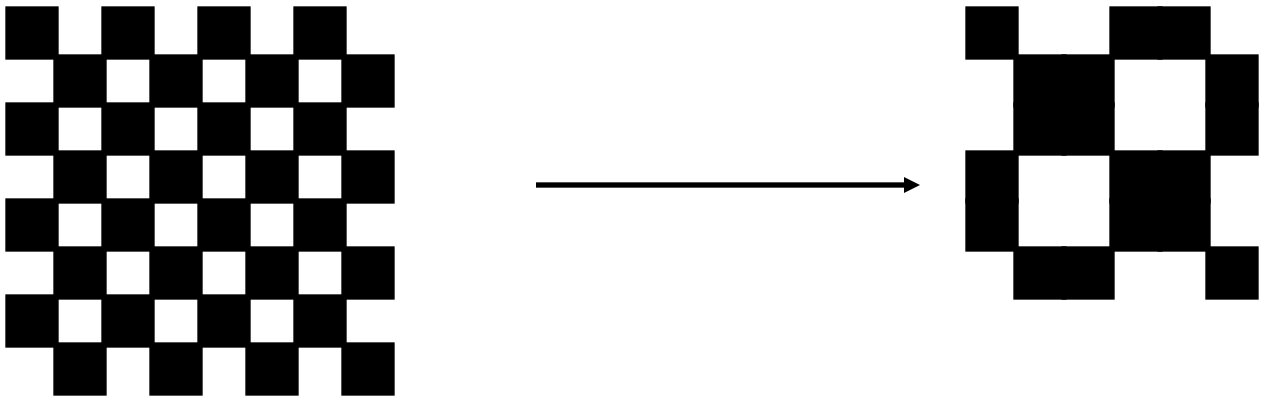
Naïve 1.5x shrinking : drop 2 out of 3



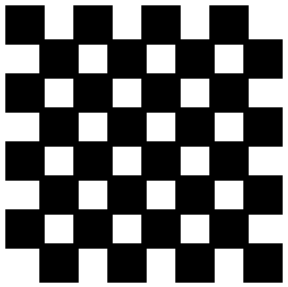
What do we get?

Shrinking

Naïve 1.5x shrinking: drop 1 out of 3



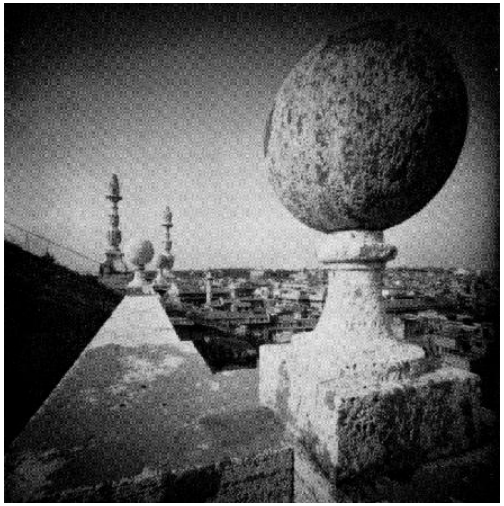
But we want



(impossible, not enough samples)

Shrinking

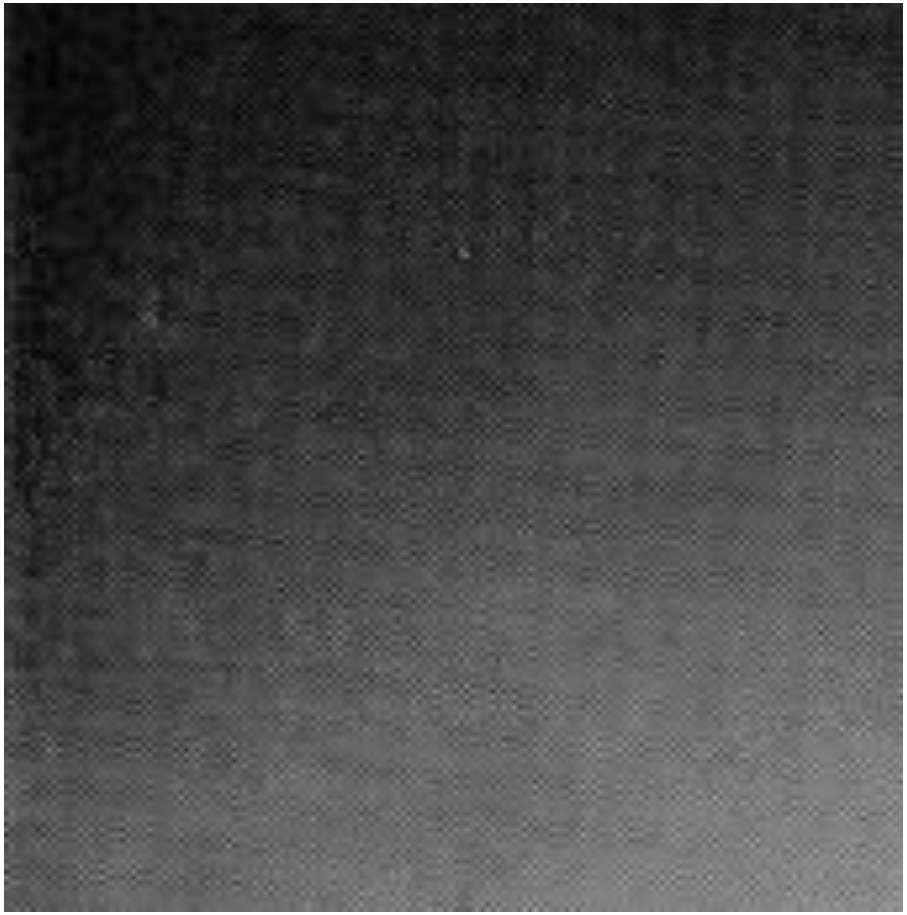
original



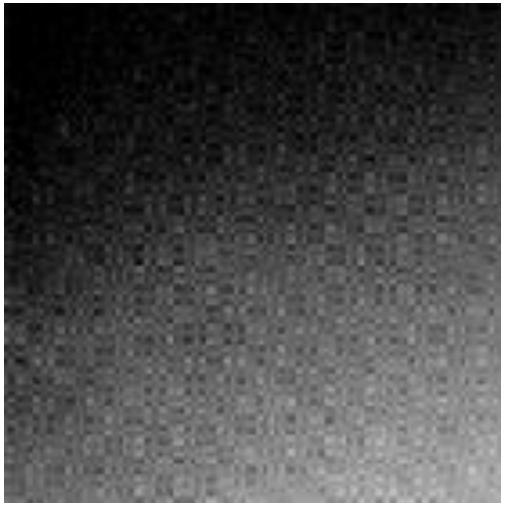
resized, nearest neighbor

resized, 11-point filter

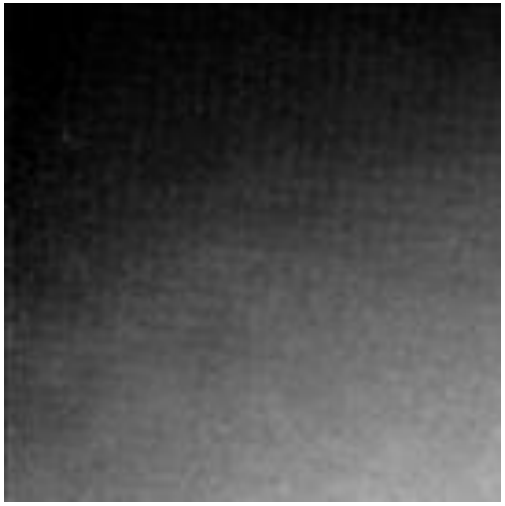
Shrinking



original



resized, nearest neighbor



resized, 11-point filter

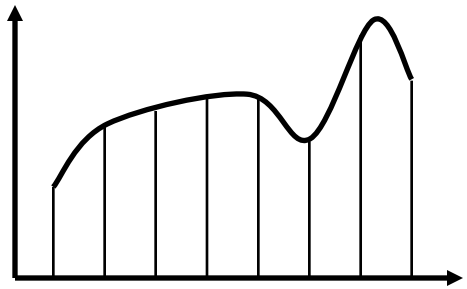
Frequency analysis

The key to fighting aliasing is to

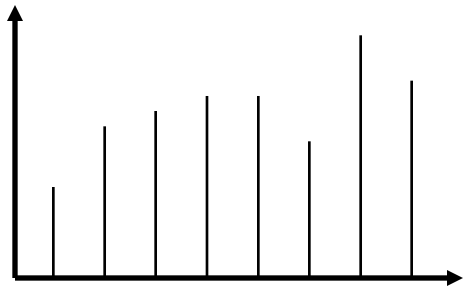
avoid frequencies we cannot represent

Big question:

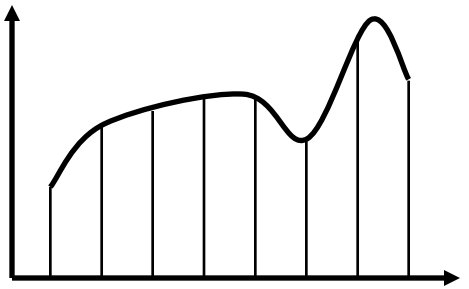
When can we reconstruct a continuous signal from samples?



original

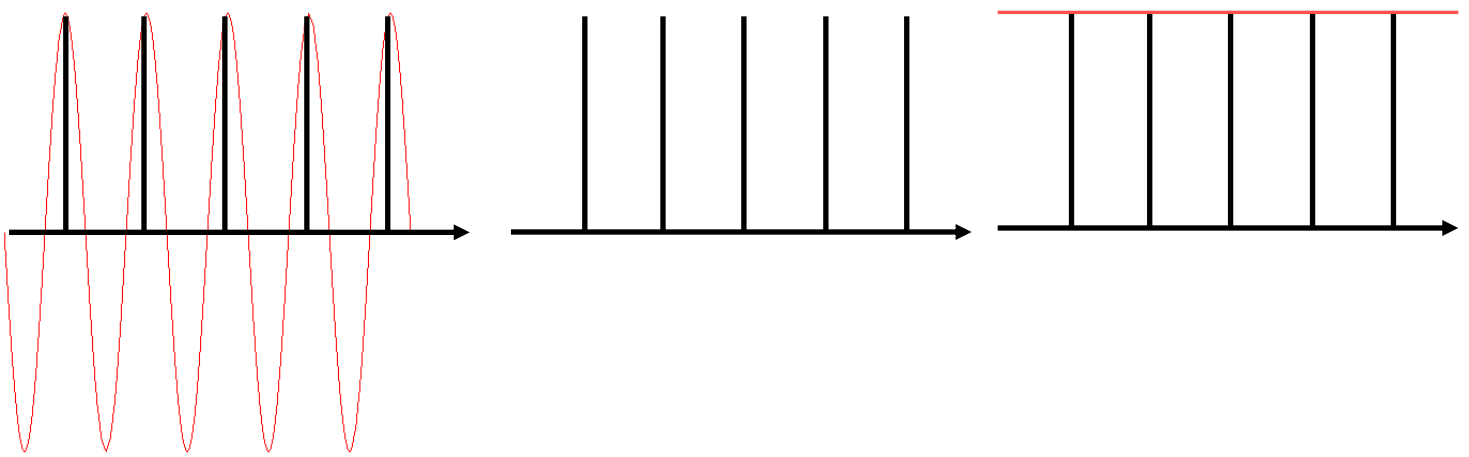


samples



reconstructed

Frequency analysis



original **samples** **reconstructed**

Need enough samples, to be more precise, sampling frequency should be more than twice the frequency of the wave