Filtering in Spatial Domain

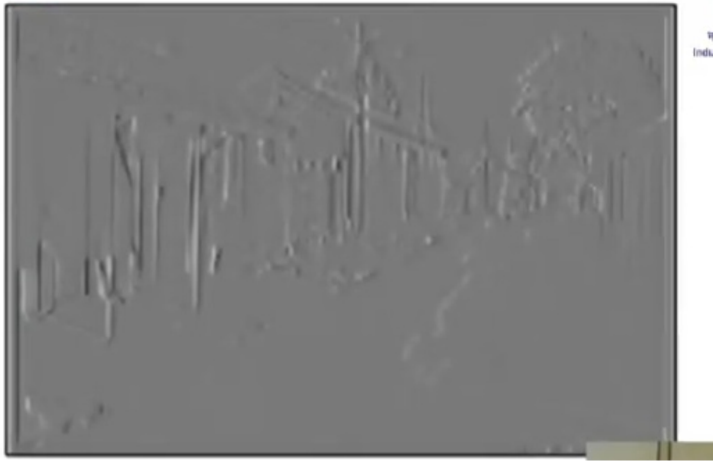
$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

output image

Intensity Image

$*$

$=$



highlighted

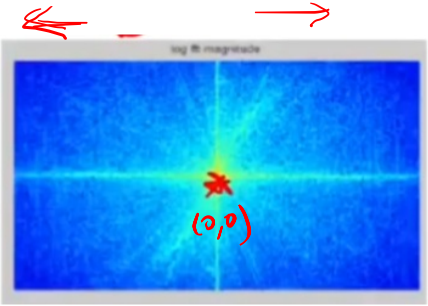
-If you have an intensity image and you apply convolution with an edge filter, you would get an output where only the edges are highlighted.

Filtering in Frequency domain

original image

frequency domain

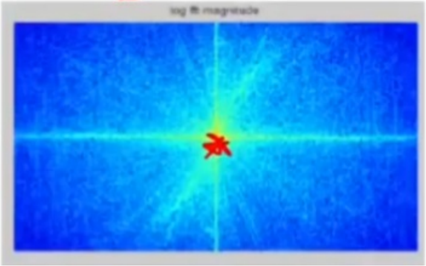
FFT



g

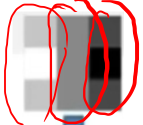
FFT $\rightarrow$

$F(g)$



$\times$

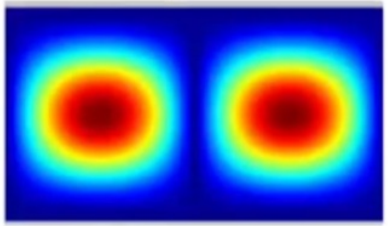
F

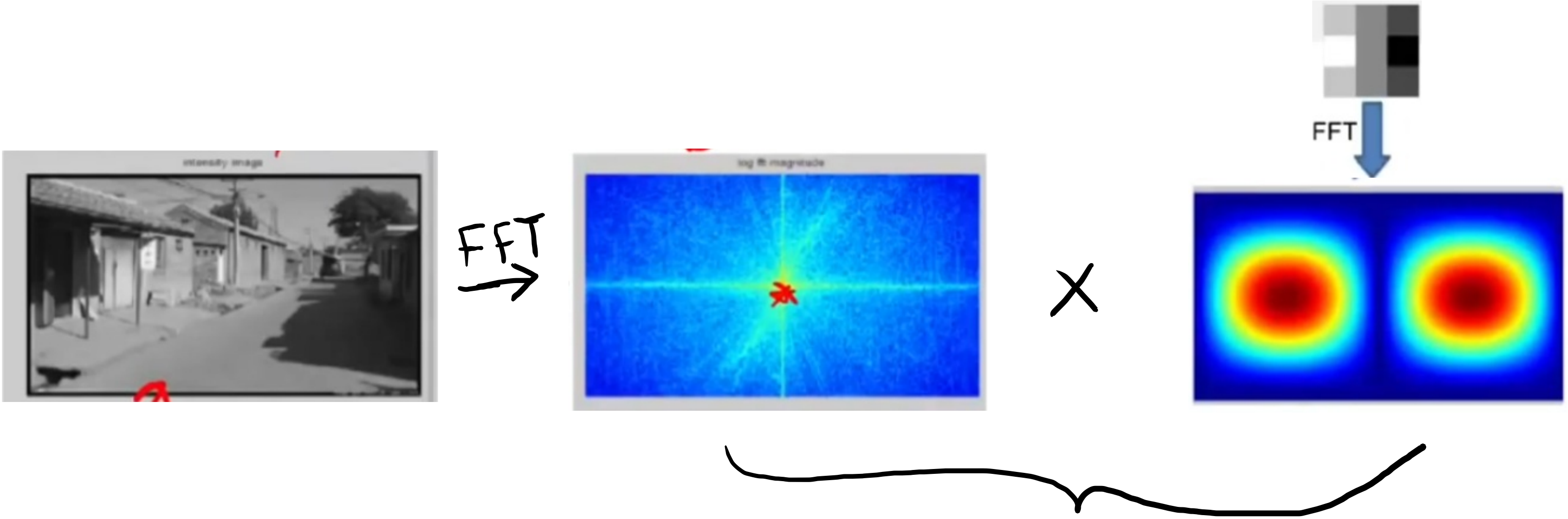


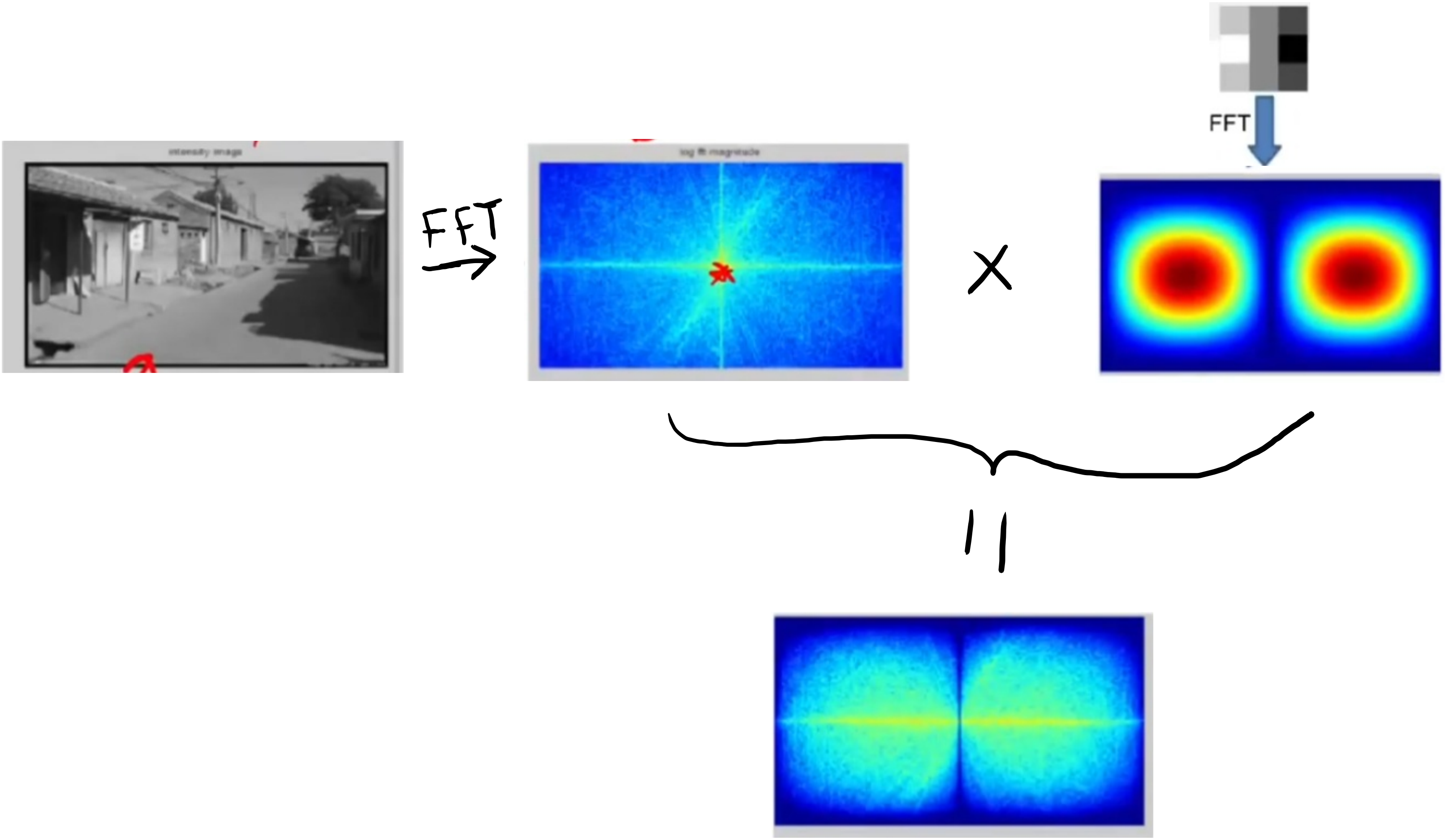
edge filter

FFT

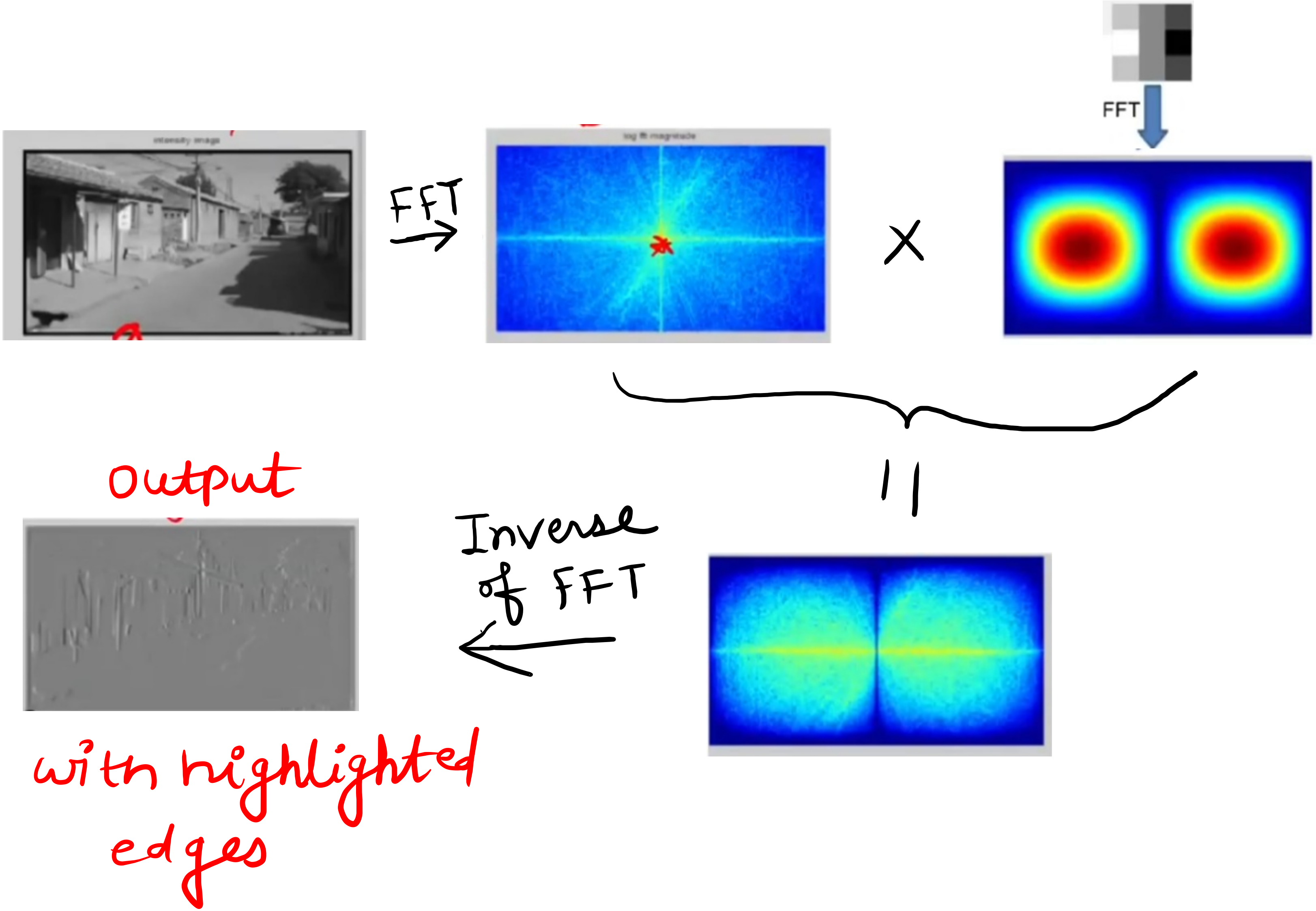
$F(h)$

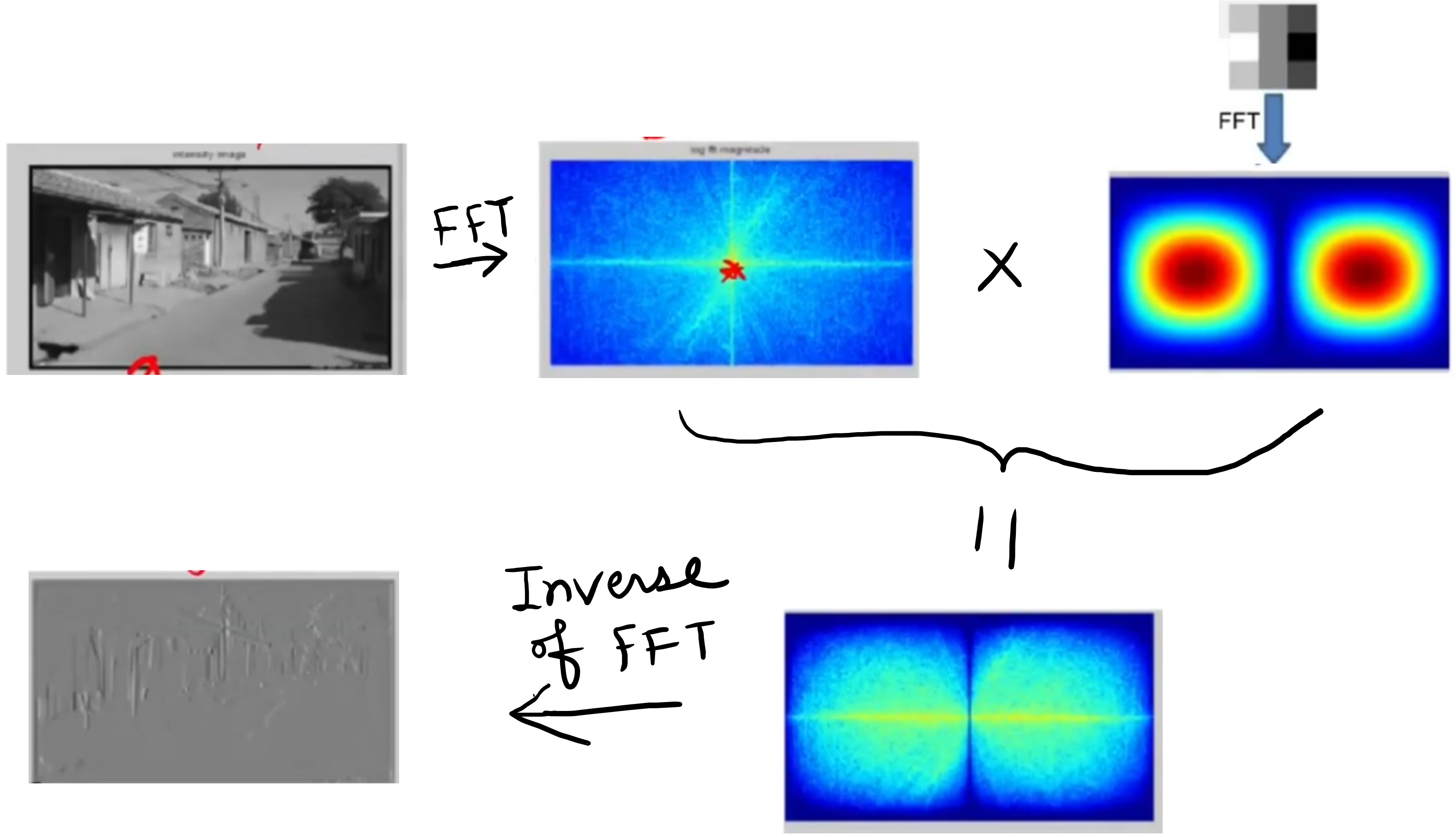



FFT
FFT

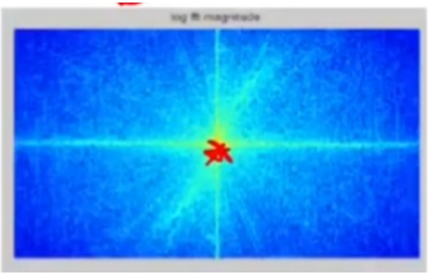


$F(g) \times F(h)$

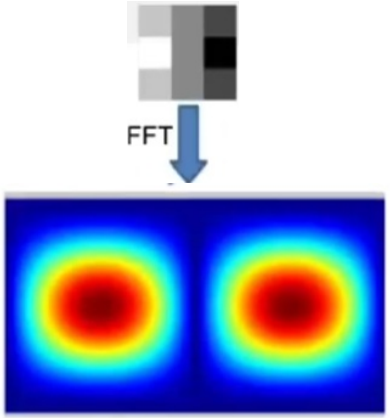

FFT
FFT
Inverse of FFT
Output
with highlighted edges


intensity image
FFT
log fft magnitude
X
FFT
=
Inverse of FFT

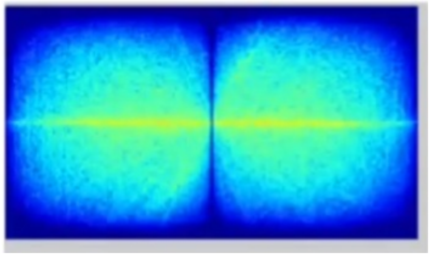
FFT $\rightarrow$



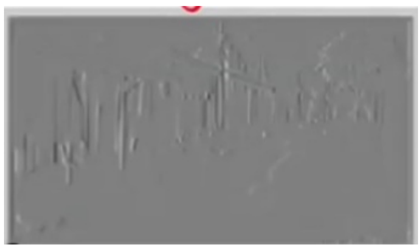
$\times$



=



Inverse of FFT $\leftarrow$



$$h * g = F^{-1}[F(h) \times F(g)]$$

Question to think

Cost improvement in convolution in spatial domain vs convolution in frequency domain

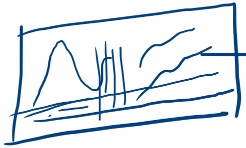
How much cost improvement?

Low Pass Filters vs High Pass Filters

1 Low pass filters- these filters allow low frequencies to go through and block high frequencies. What would be an example?

- Gaussian filter

2 High pass filters- these filters allow high frequencies to pass through and block low frequencies. Can you think of an example here?

edge filter  High frequency blocking low freq

Low Pass Filters vs High Pass Filters

Low pass filters- these filters allow low frequencies to go through and block high frequencies. What would be an example?

Gaussian

Remember that when you smoothen an image you remove certain edge information which are high frequency components. High frequency components simply mean that there is a significant change between one pixel and the pixel immediately next to it that is what we mean by a high frequency in an image

High pass filters- these filters allow high frequencies to pass through and block low frequencies. Can you think of an example here?

edge filter

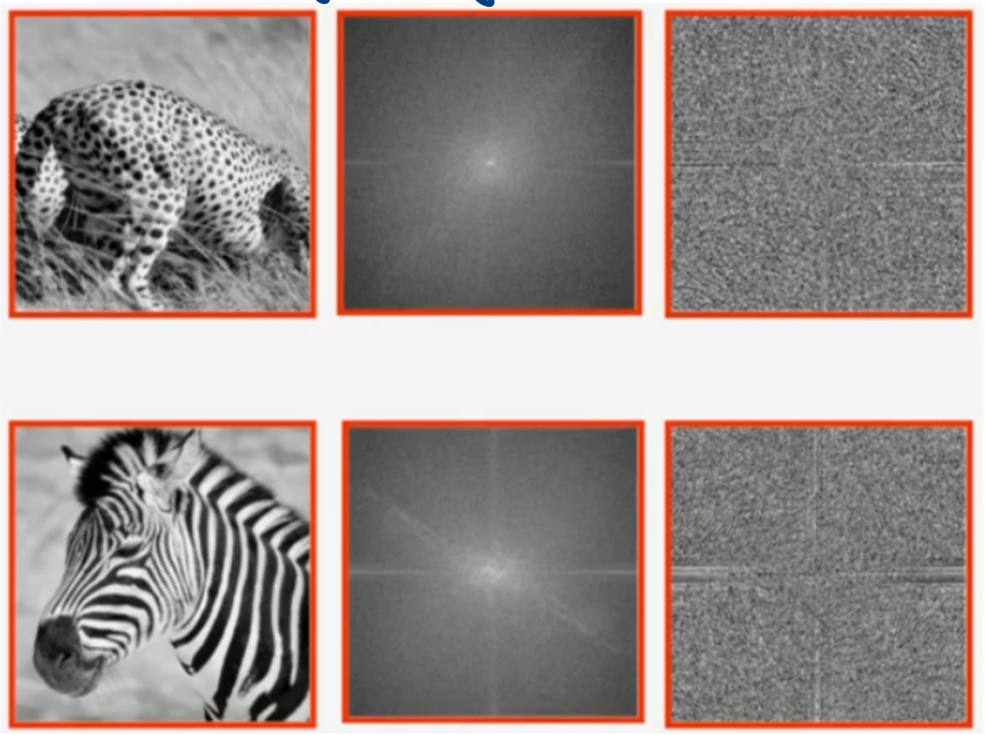
An edge is a high frequency component and it allows edges to go through, but blocks out your low frequency components across your image. When you go to a low resolution image from a high resolution image you actually lose your high frequency components. But it happens that the human eye is very good at working with medium frequency components that you can still make out the meaning in an image, but that is what you lose when you go to the lower resolution image.

Which has more information? Magnitute or Phase

original images magnitude phase

1

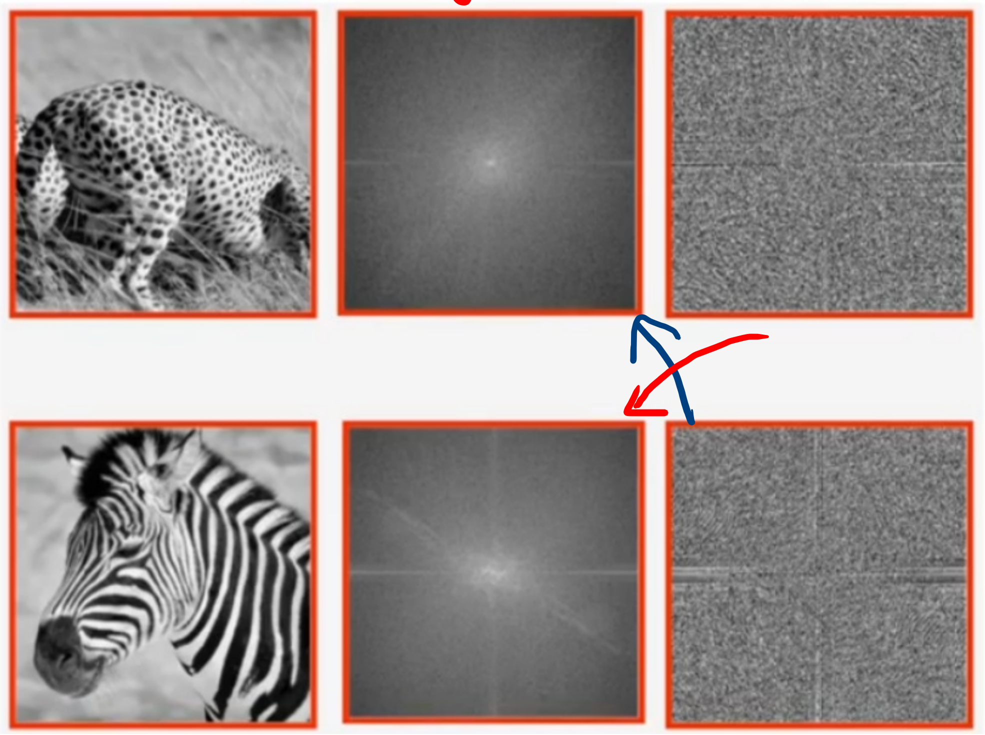
2

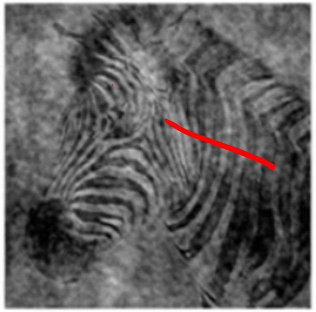


Which has more information? Magnitute or Phase

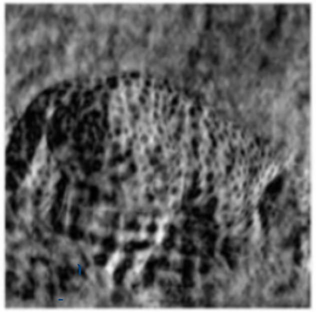
Magnitude   Phase

output image





zebra



zebra ✗

leopard

frequency   texture

Swapping the phases and combine with respective magnitude and doing an inverse fourier transform.

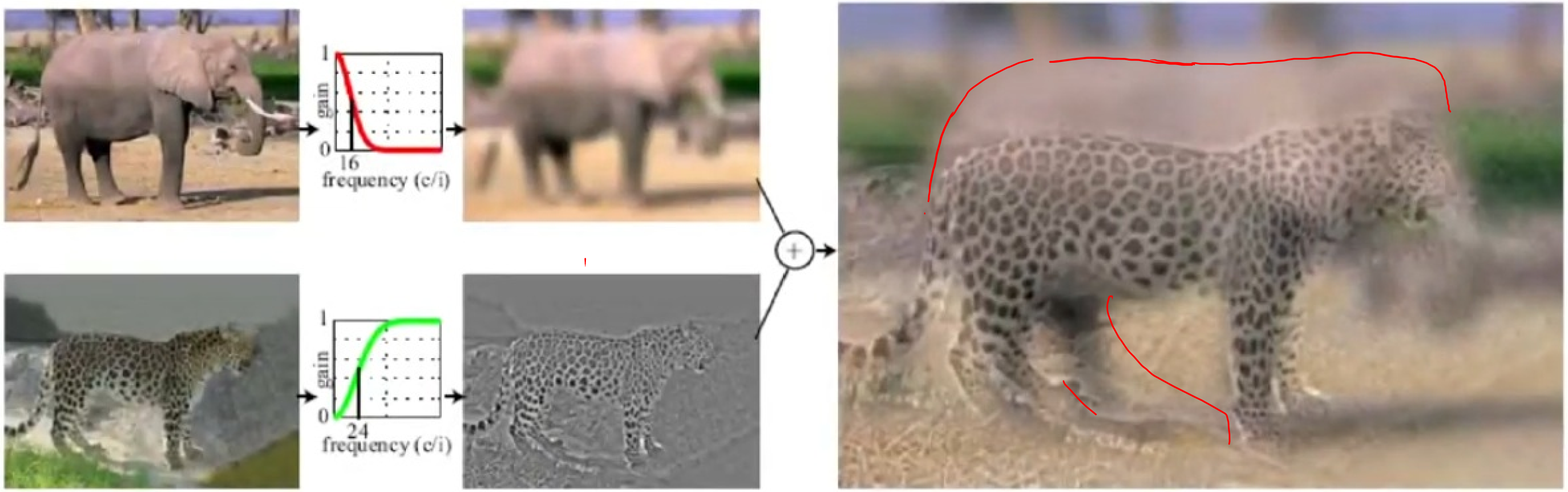
-In this case I think yes it is the magnitude of your leopard and your phase from the zebra

-Similarly you take the magnitude of the zebra and take the phase from the leopard and you get these images.

-The output tells us that the texture information comes from phase whereas the frequencies actually come from your magnitudes.

Hybrid Images

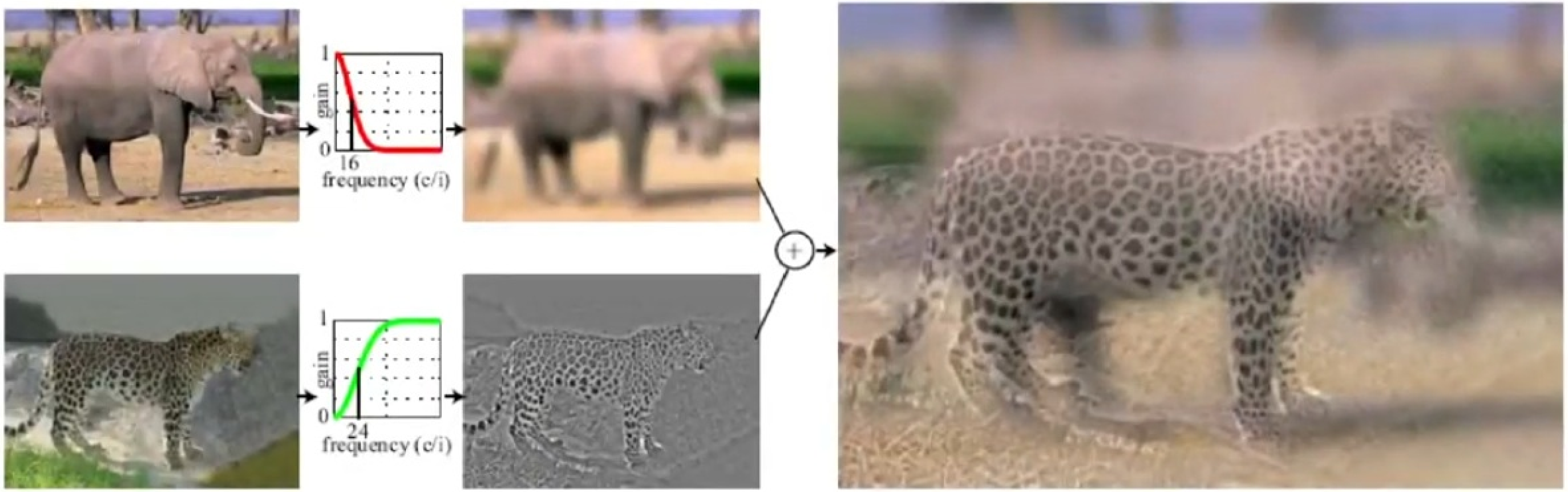
with different



output image

mid-2000

Hybrid Images



-You take an image, you simply first get a low resolution version of that by applying a Gaussian filter. You in fact can do it at multiple resolutions and then you swap your magnitudes and phases of different images and you can get pretty looking images that can be interesting.

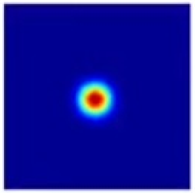
-These are called hybrid images and this was an interesting fact way back in the mid-2000s.

Match spatial domain images with frequency domain images

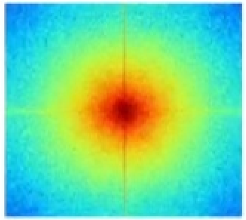
A – 2
B – 4

frequency domain

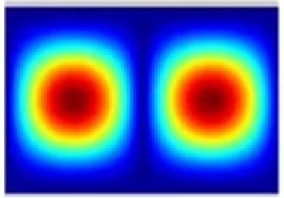
1



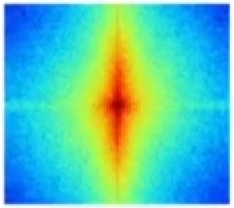
2



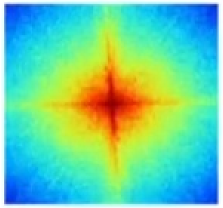
3



4

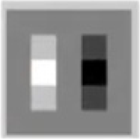


5



spatial domain images

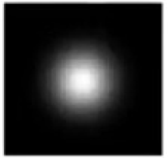
A



B

C

D



E

Subsampling ①

Original

②

Subsampled & zoomed

blurred

-What sense does a low-resolution image make to us?

-If we try to understand this from human perception perspective, it is understood today that early processing in human's filters in the human visual system are filters that look for different orientations and scales of frequency.
Sometimes people also relate them to what are known as Gabor filter.

-And secondly the perceptual cues in the human visual system are largely based on mid to high frequencies. So high frequency means an edge because there is a sudden change in intensity value as you move from one pixel to the next pixel along a row or a column.

-If it is along a row, you would call that a vertical frequency, if it is along a column, you would call that a horizontal edge. So perceptual cues depend on mid to high frequencies in general.

-When you see a low-resolution image, it is equivalent to sub-sampling image or if you are seeing an image from very far away you are actually sub-sampling the image.

-Which means you may be losing some of the high frequency information but you do get a lot of the low frequency and mid frequency information which is good enough to get a sense of what the object is but may not give you finer details.

Example of subsampling

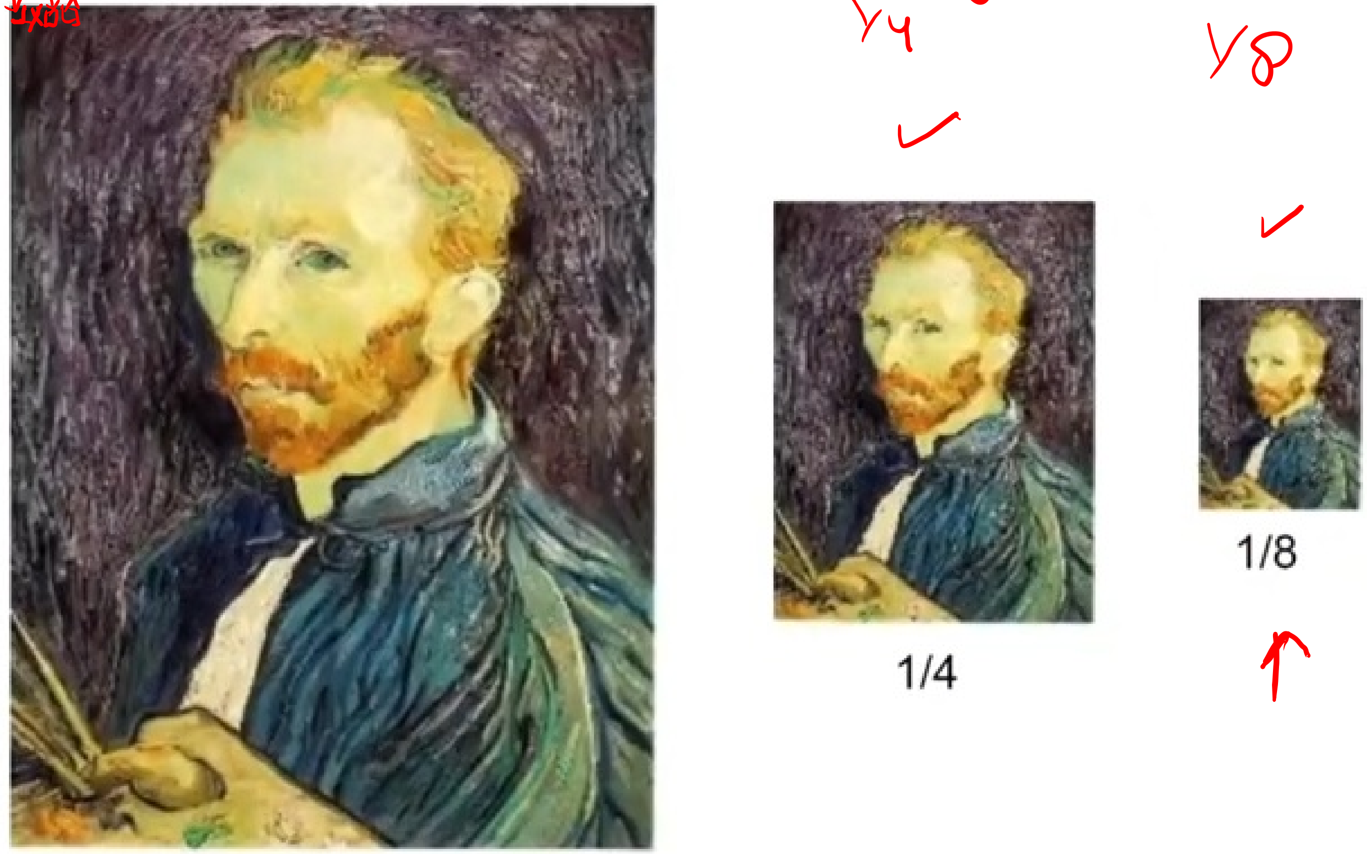
1/2 →

1/2

1/4

1/4

1/8 1/16



1/4

1/8

1/16

1/32

So, how do you sub-sample an image as an operation?

-The simplest way to sub-sample is throw away every other row and column.

-Consider only every alternate pixel both row wise and column wise.

-That straight away gives you a half size image.

-You can then go to repeating this to go to lower and lower resolutions.

first middle last

1/2

1/4 (2x zoom)

1/8 (4x zoom)

-When you actually zoomed into these low-resolution images, or the sub-sampled images, and looked at those images.

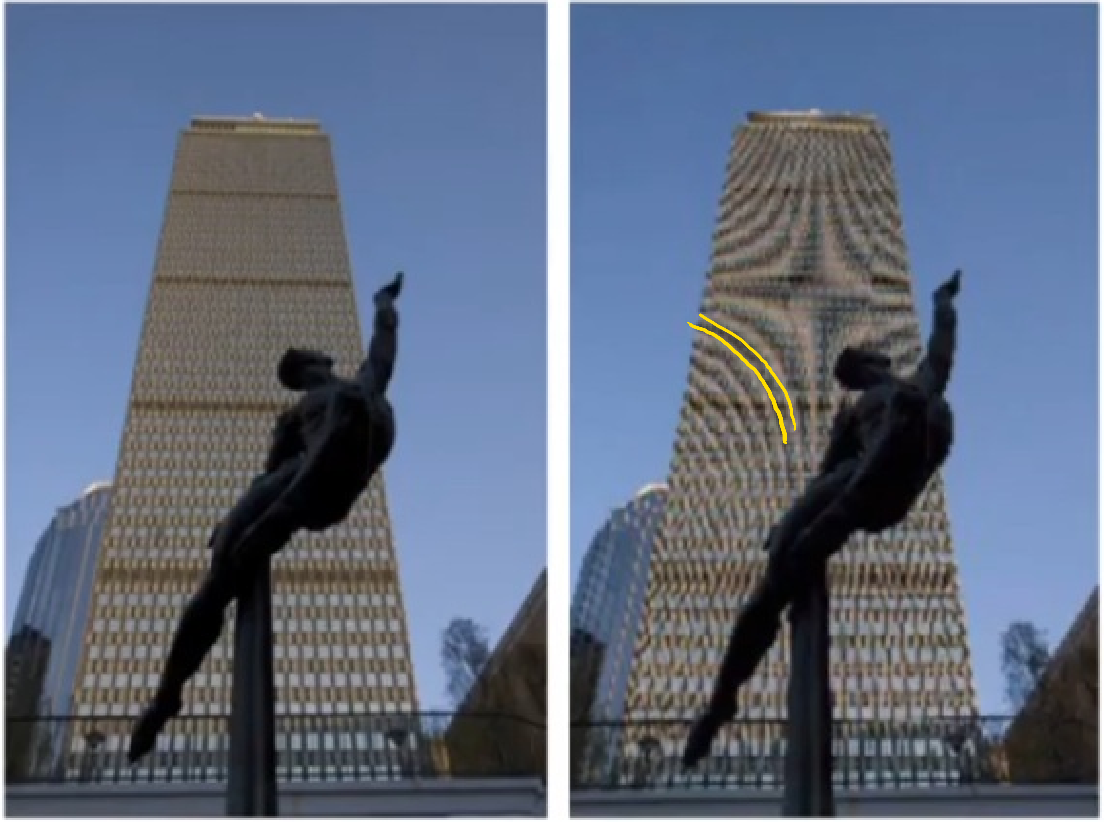
-The half sampled image gets you something like shown in first image, the quarter sampled image at a 2x zoom gets you something like in middle image, the 1 by 8th image at a 4x zoom looks like in last image.

-You may say that is expected if you sub-sample.

-Why do you worry about how it looks at close, that is a valid question.

-Here is the last image here looks pretty crufty, it has too many artifacts, can we at least make it look smooth? We understand that by subsampling you have lost some information but can it appear smooth when you zoom in,

-What we are looking for, we understand that you may have lost sharpness but we still want to look, make it look smooth. How would you achieve this?



what is happening in this kind of an image? So this is the original image, we have sub-sampled in some way

This effect is known as aliasing.

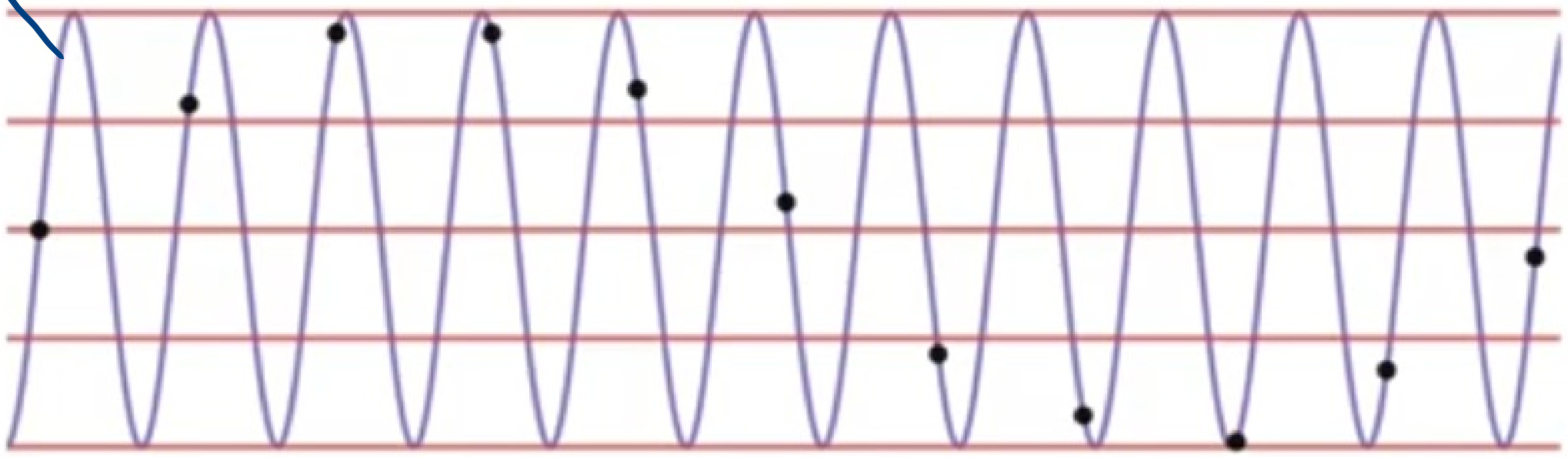
Aliasing generally happens when your sampling rate is not high enough to capture the amount of detail in the image.

We are going to see a couple of examples and then re-visit the painting that we saw and how do you ensure that it is not crufty.

Aliasing

-Whenever your sampling rate is not high enough, you will not get the details
- For example, you sub-sample a sinusoidal wave form.
-So if you only sub-sample at these black points, you may think that this is the actual sin wave.

original information



- black dot is sampling of original waveform

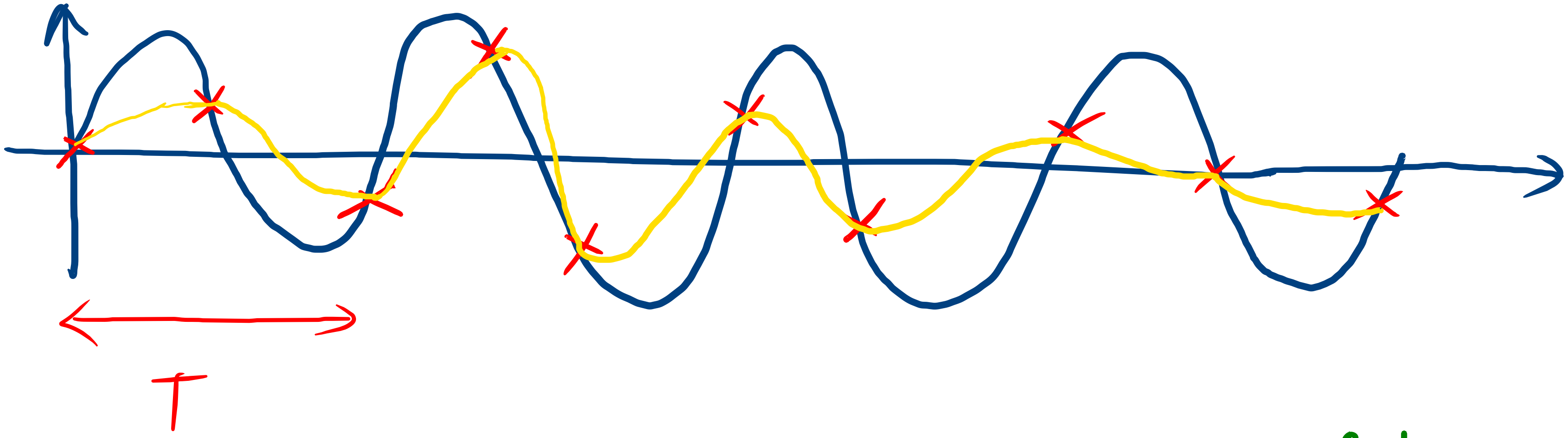
Sampling Rate

NO – entire infor with red sampling waveform

-Remember that the frequency of this red wave that we have just drew is very different from the original wave.

-And why is it so different, only because we did not sample the original signal.

-Here purple waveform is original signal. Because we did not sample it enough, that is the reason why we end up getting this kind of a shape which is very different from the original sinusoid wave.



$$f_s \geq 2 \times f_{max}$$

to avoid aliasing effect

f_s → sampling rate

f_{max} → maximum freq^n of original signal

Nyquist

-So what you need to do to get the sampling right

-Ideally you need to know the structure of your original signal

-And this is where an important concept called the Nyquist rate comes into the picture.

-That means the minimum sampling rate that is required to reconstruct your original signal or to capture the original signal in its actual form is called the Nyquist rate.

-So now what does the Nyquist rate mean?

Aliasing: Problem

-Nyquist rate is relevant to fundamentals from informational theory and signal processing where Shannon whose sampling theorem is very popular

-Shannon proposed that the minimum sampling rate or the Nyquist frequency is given by f_s which has to be at least $2 \times f_{max}$

-Where f_{max} here is the highest frequency in a signal, it is the highest frequency in a signal.

-Your sampling rate or Nyquist frequency should be at least twice the highest frequency that you have in your image or signal.

Image Aliasing

Subsampled

Striped shirt's pattern look weird on screen.

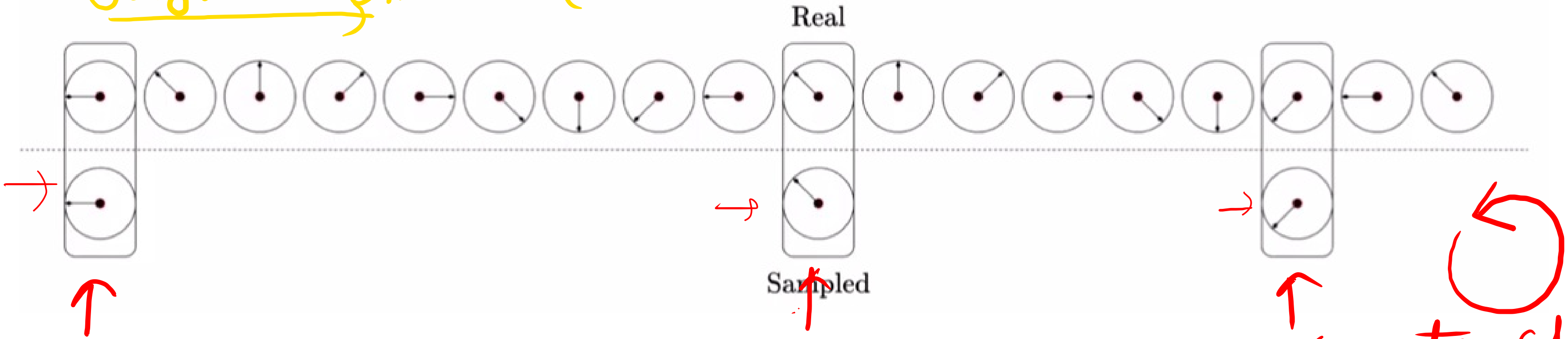
Why is this so?

Similar impacts can be seen in an image setting, in a video setting or even in a graphic setting.

So in an image setting if you do not sample your original signal at the appropriate frequency which is twice f_{max} your shirt is going to look very weird and this is due to aliasing.

Video Aliasing

original video (clockwise)



counter clockwise

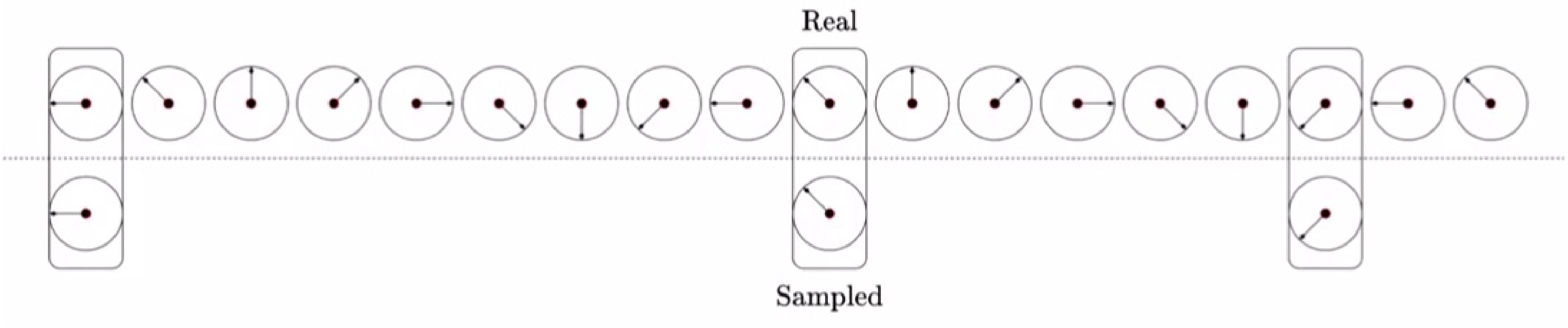
In a video, this is something that you may have observed a lot, sometimes when you go out on a road you see a bicycle, especially it happens with bicycles or car wheels.

Although car is moving forward, when you look at the wheels alone, it may appear that the wheels are circling backward. So this could be first sight bicycle or a car, try observing this if you have not already done this.

If you now take the motion of a wheel to be the top row here, so the top row here is the actual motion. As you can see the top row, you can see that the arrow here is moving in a clockwise direction.

So you can assume that this is something like a cycle wheel moving in a clockwise direction. But we have chosen to sample it only at three locations across all of these.

So the top row is the actual movement of the wheel but these boxes here are the ones that are actually sampled, they are the ones that are sampled.



So if you looked at only the sampled positions, you see that the arrow is first here, then it is here, then it is here, so on so forth.

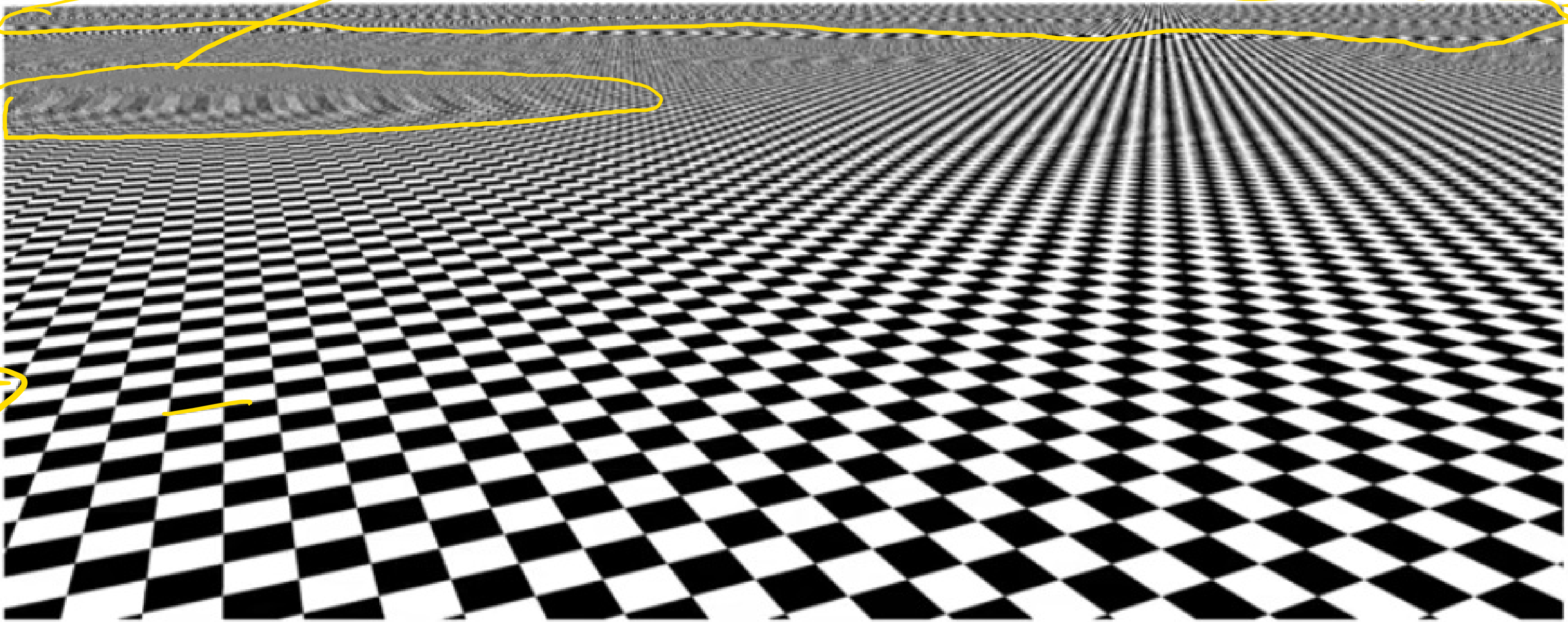
So probably in fact the second and third one could give you the reason why this effect happens, so you see that if you only took the second and the third one, you would actually think that your wheel is moving counter clockwise, because those are the only two samples you have and the arrows moving in the counter clockwise direction, but originally your wheel was moving in the clockwise direction.

This simply means that you are not sampling the wheel at the frequency at which it is rotating or at twice the maximum frequency at which it is rotating.

Aliasing: Graphics

frequency is very high

Rendering



Checkerboards disintegrate in ray tracing.

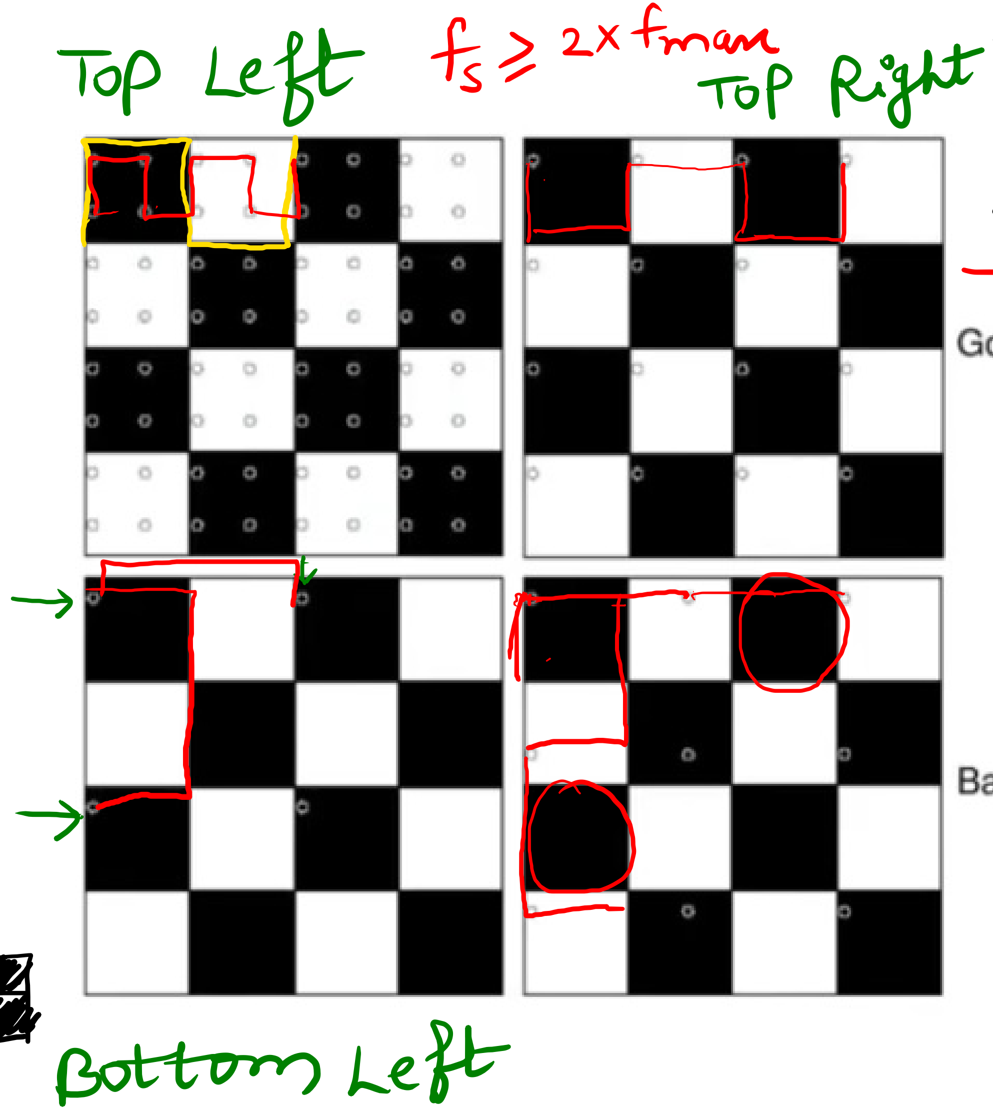
Here is another example from graphics and the reason is again the same. You can see in this particular case that the aliasing is prominent at these regions where the frequency is very, very high.

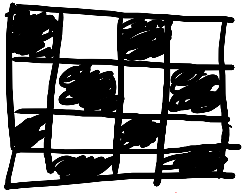
Remember once again, when we say frequency in an image, we mean how quickly the pixels change between black and white.

When we say black and white, it can be dark grey or light grey but going from black and white and that becomes very high, very closer to the horizon of this image.

And that is where the frequency is very high, which means your sampling rate has to be twice that frequency to be able to be true to the original image and this is an effect that we often see with graphics.

Let us try to intuitively understand why you need to do sample at twice the maximum frequency. Let us take this example of a chess board.

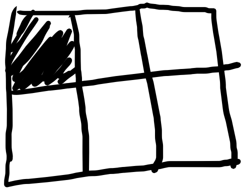




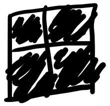
— objective

→ at least

Good sampling



Bad sampling



You have a chess board here and we argue now that the top row are examples if good sampling practices and the bottom row are bad sampling practices.

Bottom left, you can see that the samples are circles here, you can see those circles. And when you see those samples, you can clearly see that we are sampling all of them at only blacks.

So if you had only those four samples, you going to assume this entire board is a black board, that there are no white squares in between because you never sampled in the white squares.

So clearly you are not sampling at that appropriate frequency, the frequency here is how quickly does white and black change in how many pixels does it change.

Clearly you are not sampling at twice that maximum frequency and it is going to give you poor information.

Bottom right is a slightly better sampling but still not complete, so once again here you can see these hallow circles, you can see that you improved your sampling a little bit but once again you see a black, you see a white and again your third sample on that first row is a white again.

So, which means your perception or any further processing is going to imagine that this is black followed by a series of white, that is not checker board or a chess board, but it is just an entire block of white, after the first small square of black it is an entire block of white, which again is poor sampling.

So here is a sampling where we actually go at twice the frequency, you may ask why twice

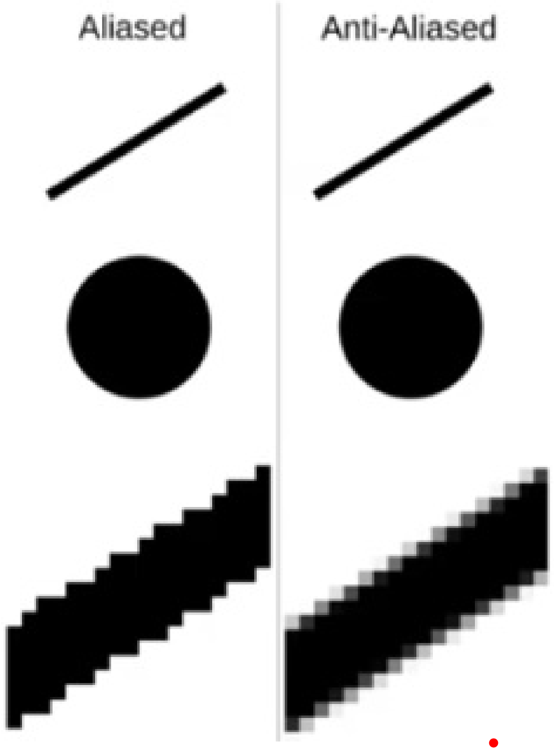
So in this case remember here that the actual frequency is when a signal completely goes through between its variations. So in this case a signal completes from one point to another point and we are sampling twice inside those two boxes. So which means that we are sampling twice the maximum frequency.
And when you sample like that you have got a sample in every black square and every white square and you now know this is a checker board.
Obviously if you sample more than that, that is even better for you.

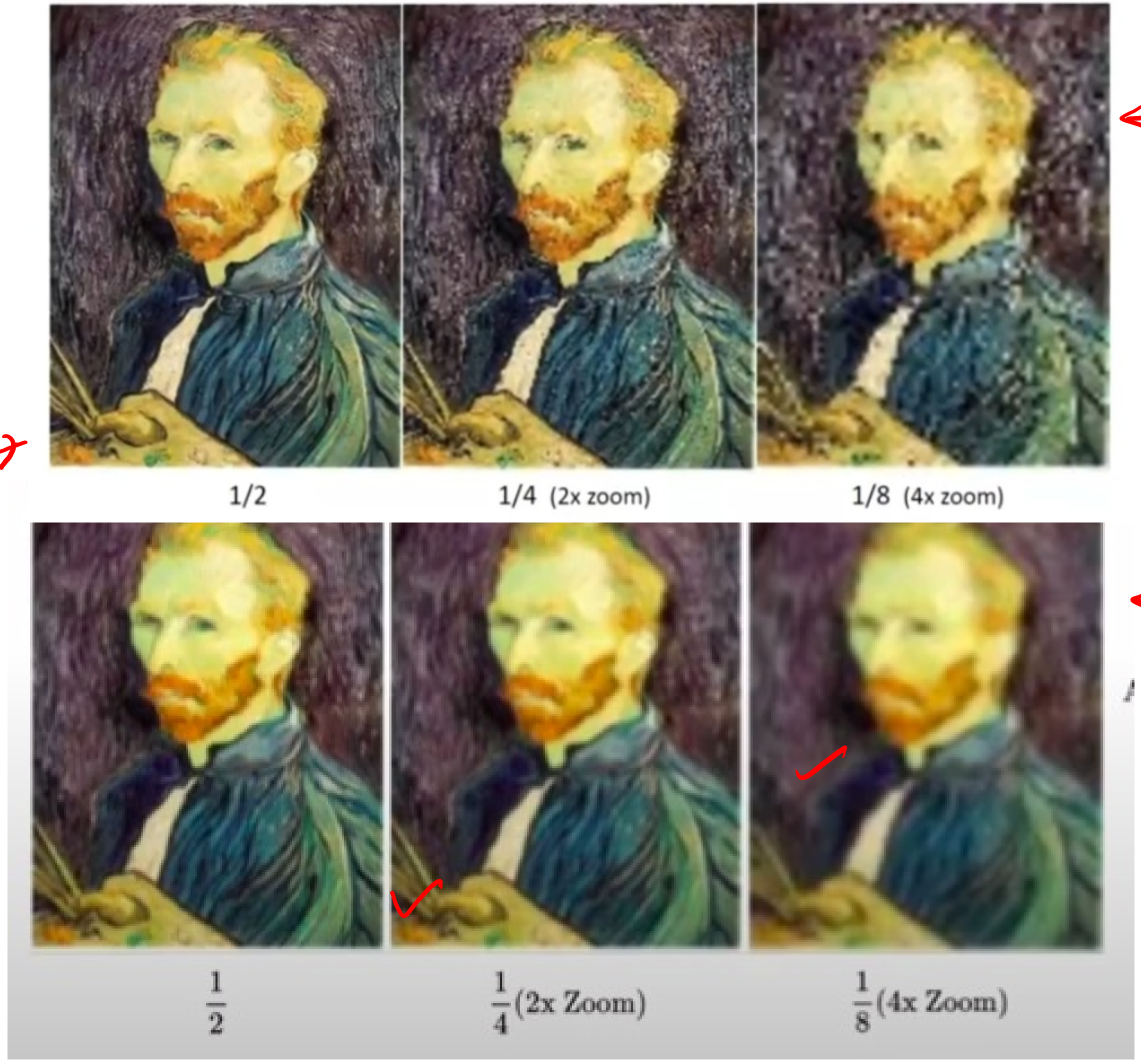
Hopefully that give you an intuitive understanding of why you need to sample any signal, any image at, at least twice the maximum frequency of content in the image.

Anti-aliasing

The example of a good anti-aliasing technique is Gaussian pre-filtering

That means, you going to lose some information but we at least wanted to make it look smooth.





So how do you do Gaussian pre-filtering?

Gaussian Pre-filtering

You first going to take an image and then do a Gaussian blur on it, that is Gaussian smoothing.

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

5x5

You can take a 3x3 Gaussian kernel.

So you first apply that kernel on your input image that gives you a blur version of the image. Now, you sub-sample, then before you sub-sample again blur then sub-sample, blur then sub-sample, blur then sub-sample, so on and so forth.

Since a Gaussian filter is a low pass filter, which means it removes high frequencies and only passes through the low frequencies, relatively lower frequencies which means it is going to remove, a certain amount of high frequencies.

So by applying the Gaussian blur, you have removed the high frequency which means your sampling rate now can actually be reduced.

Clarity is better

So you do a Gaussian blur, now your overall frequencies have come to a smaller range.

Now you can afford to sub-sample without having an aliasing effect or cruftiness in the image

And that is what we mean Gaussian pre-filter.

$f_{max} = 2$

$f_s \geq 2 \times 2$

$f_s \geq 4$

$f_s \geq 2 \times f_{max}$

4

$f_s \geq 8$

Remember again the Shannon sampling theorem states that you must sample at least twice the maximum frequency in the image.

If I reduce the maximum frequency in the image, I can sample at a lesser rate.

And that is the trick that we employ to be able to sub-sample but still have a smooth effect in your output image.