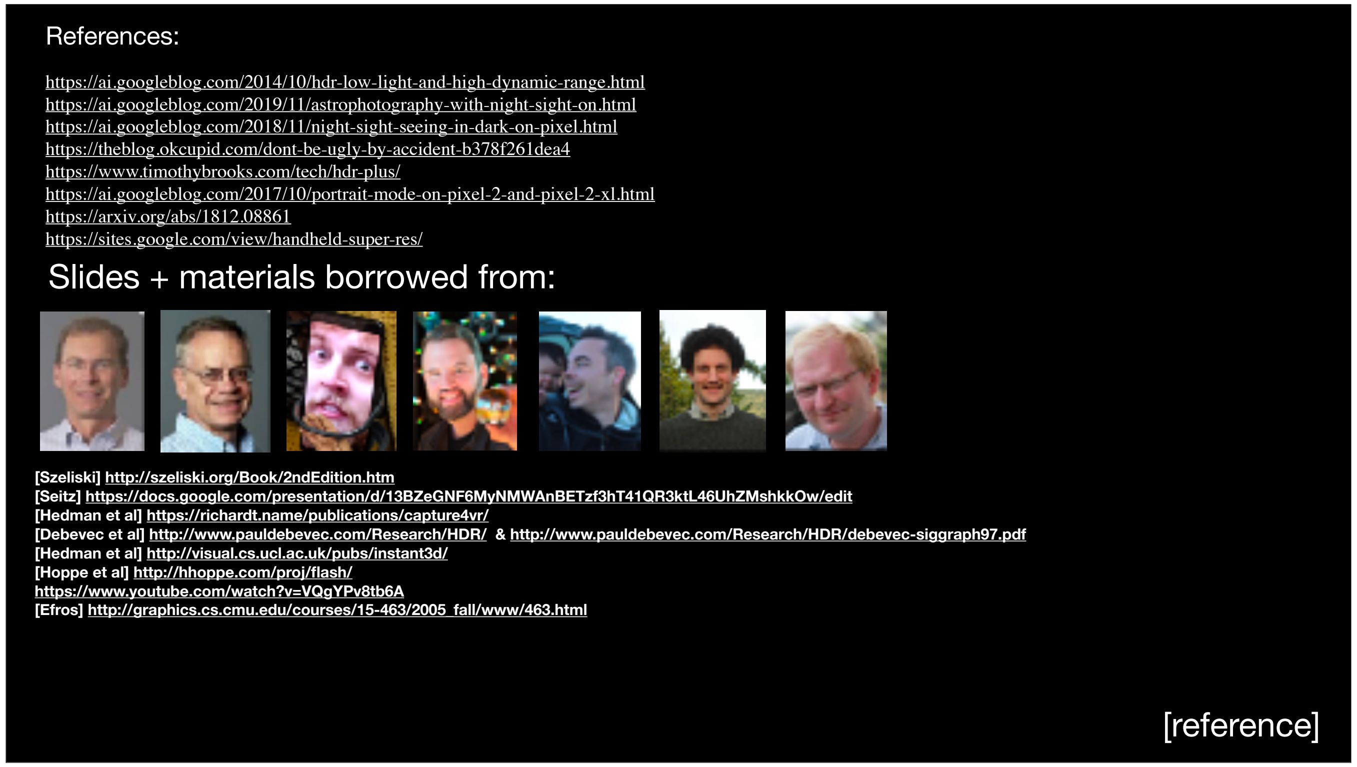
Computational Photography

CSE 576, Spring 2020

Aleksander Holynski



References:

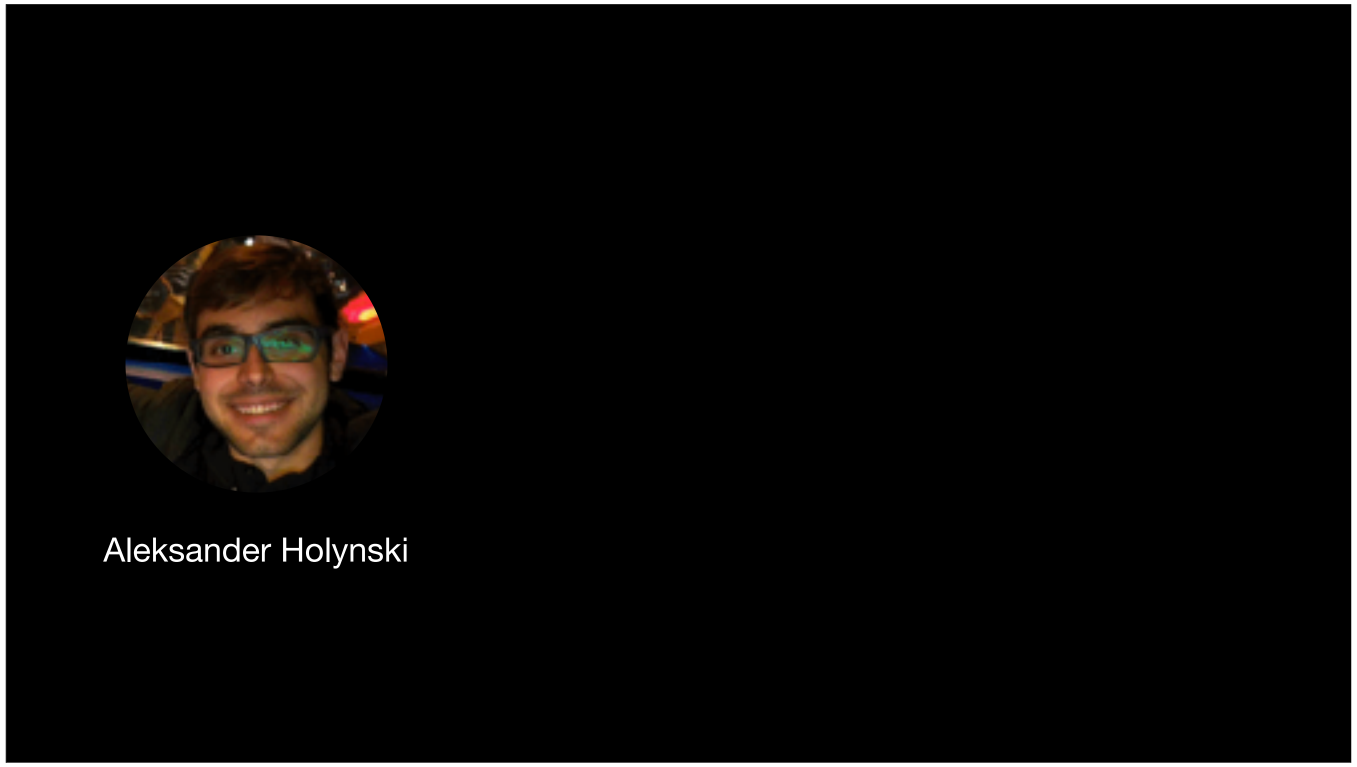
https://ai.googleblog.com/2014/10/hdr-low-light-and-high-dynamic-range.html
https://ai.googleblog.com/2019/11/astrophotography-with-night-sight-on.html
https://ai.googleblog.com/2018/11/night-sight-seeing-in-dark-on-pixel.html
https://theblog.okcupid.com/dont-be-ugly-by-accident-b378f261dea4
https://www.timothybrooks.com/tech/hdr-plus/
https://ai.googleblog.com/2017/10/portrait-mode-on-pixel-2-and-pixel-2-xl.html
https://arxiv.org/abs/1812.08861
https://sites.google.com/view/handheld-super-res/

Slides + materials borrowed from:

[Szeliski] http://szeliski.org/Book/2ndEdition.htm
[Seitz] https://docs.google.com/presentation/d/13BZeGNF6MyNMWAnBETzf3hT41QR3ktL46UhZMshkkOw/edit
[Hedman et al] https://richardt.name/publications/capture4vr/
[Debevec et al] http://www.pauldebevec.com/Research/HDR/ & http://www.pauldebevec.com/Research/HDR/debevec-siggraph97.pdf
[Hedman et al] http://visual.cs.ucl.ac.uk/pubs/instant3d/
[Hoppe et al] http://hhoppe.com/proj/flash/
https://www.youtube.com/watch?v=VQgYPv8tb6A
[Efros] http://graphics.cs.cmu.edu/courses/15-463/2005_fall/www/463.html

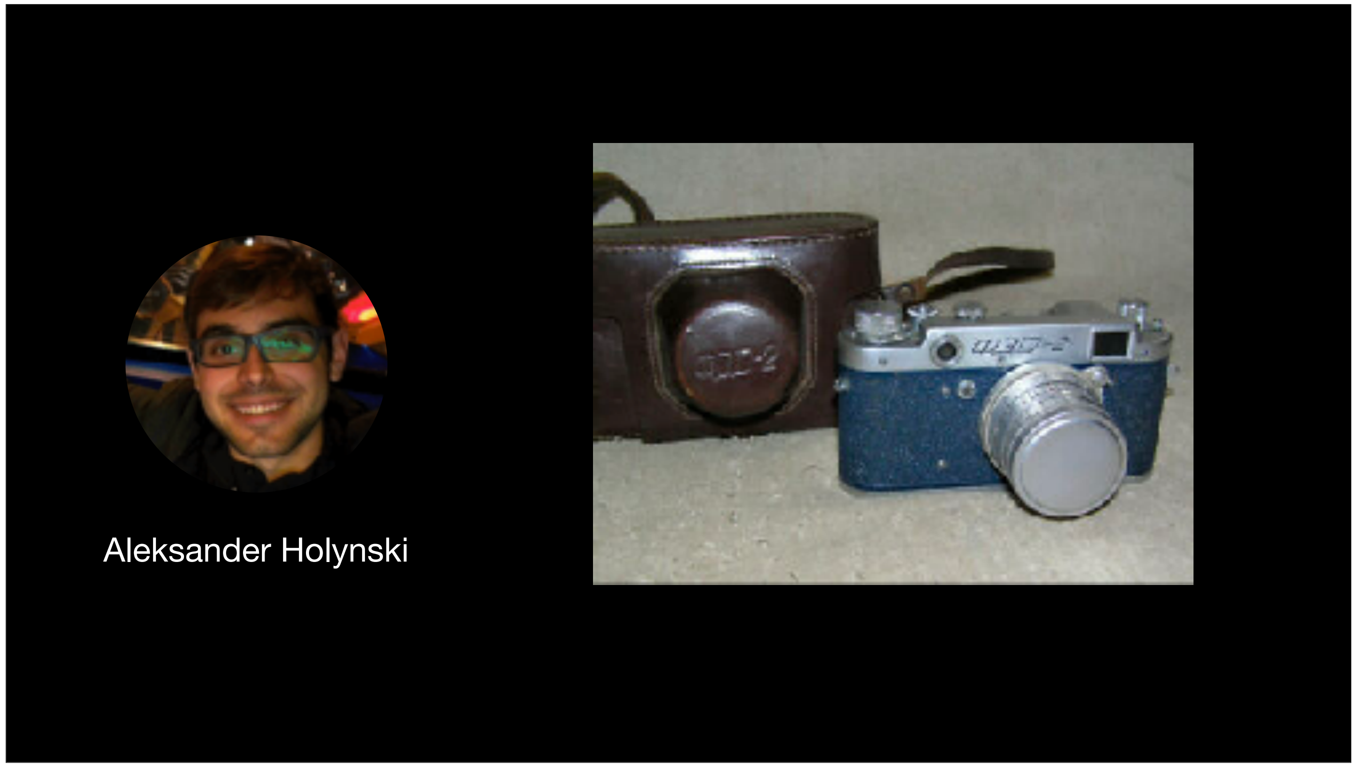
[reference]

Some of the great resources I used for this talk —



Aleksander Holynski

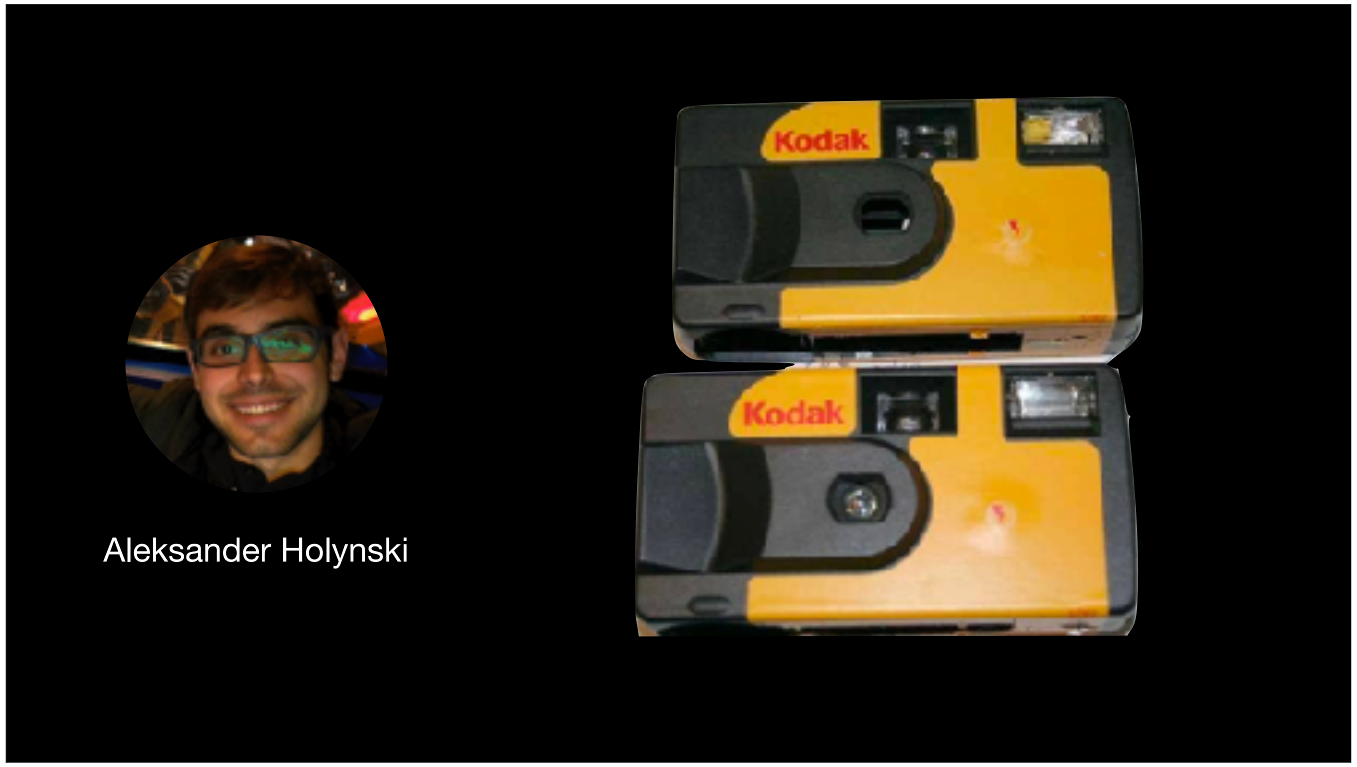
I’ll introduce myself through the cameras I’ve owned over the years.



Aleksander Holynski

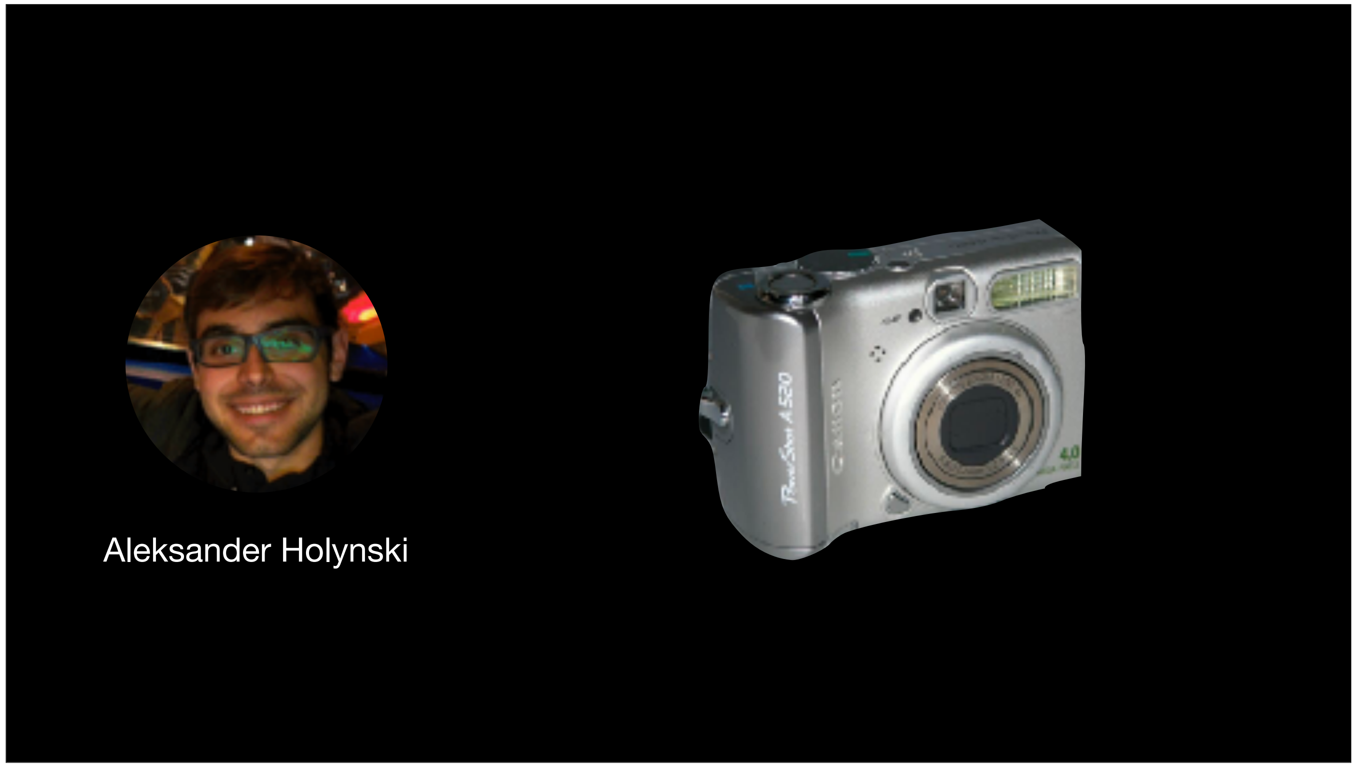
My first camera was actually an old Soviet camera I found in my parents’ closet, they had from the 70s

And I remember taking some pictures with it, and even got them developed, but I didn’t have a good grasp on how to use it, apart from pressing the shutter button.



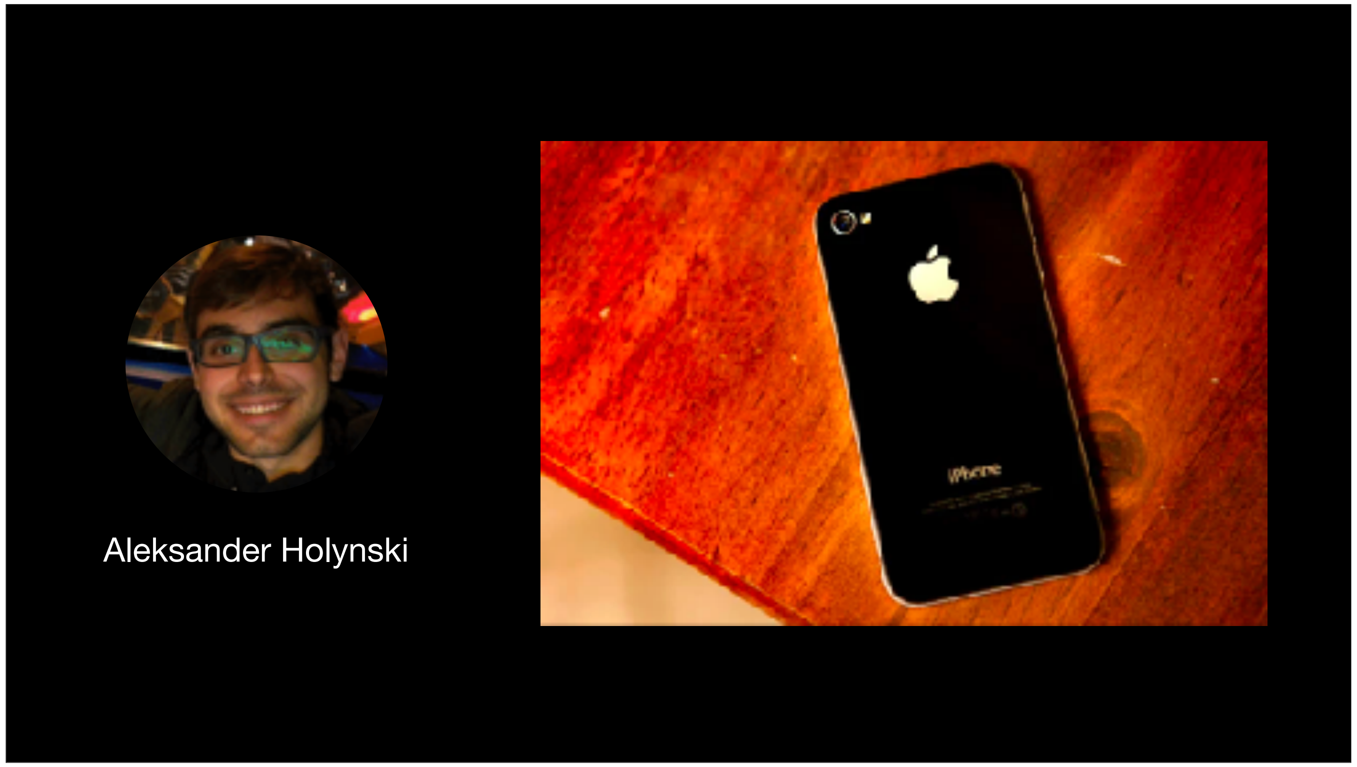
Aleksander Holynski

I took a bunch of pictures with disposable cameras that you could buy at the drugstore, and you'd take it back to get the film developed.



Aleksander Holynski

The next camera I owned was a point-and-shoot camera

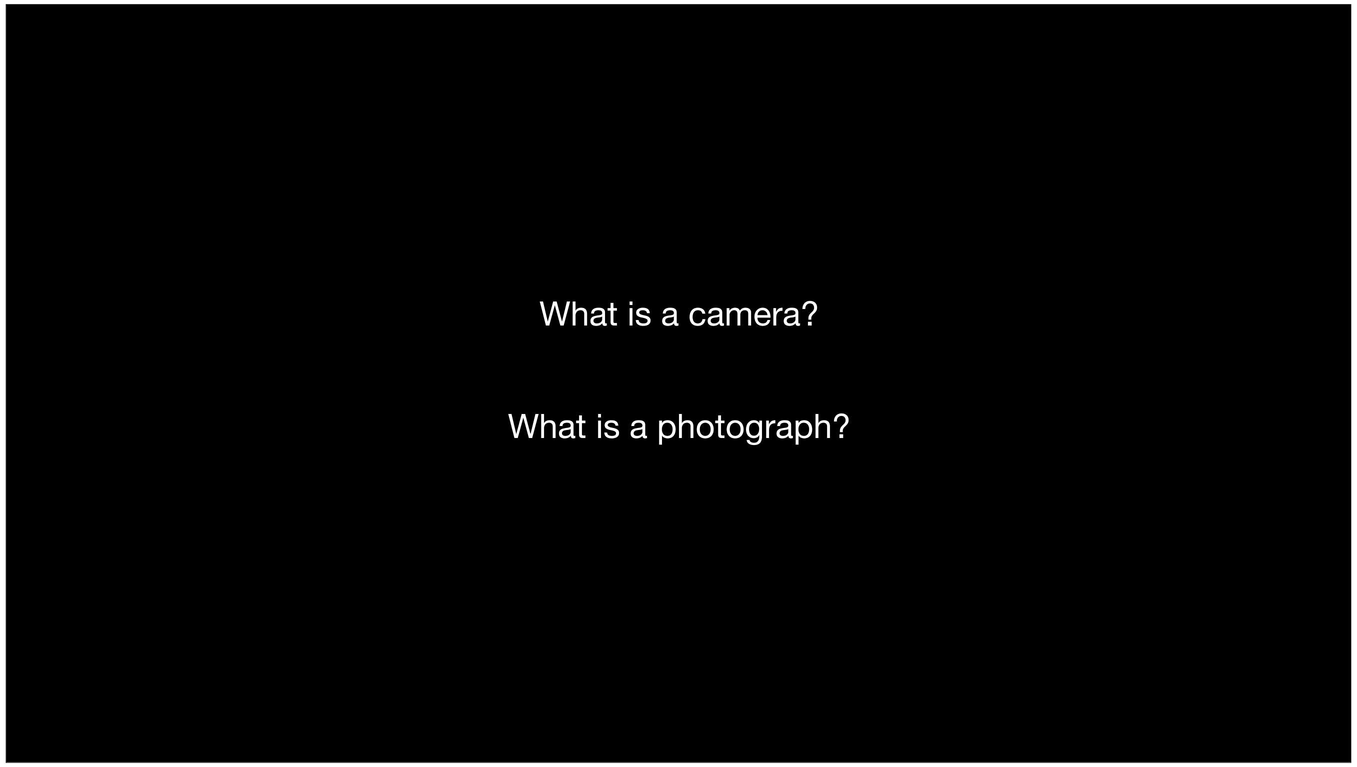


Aleksander Holynski

And then eventually when I got a smartphone, that was my camera.

The point I want to make is that I had always used cameras with an “Automatic Mode” which took the best pictures, so I never learned what any of the parameters or knobs on cameras did (e.g. aperture and ISO and focus), because the camera just figured it out.

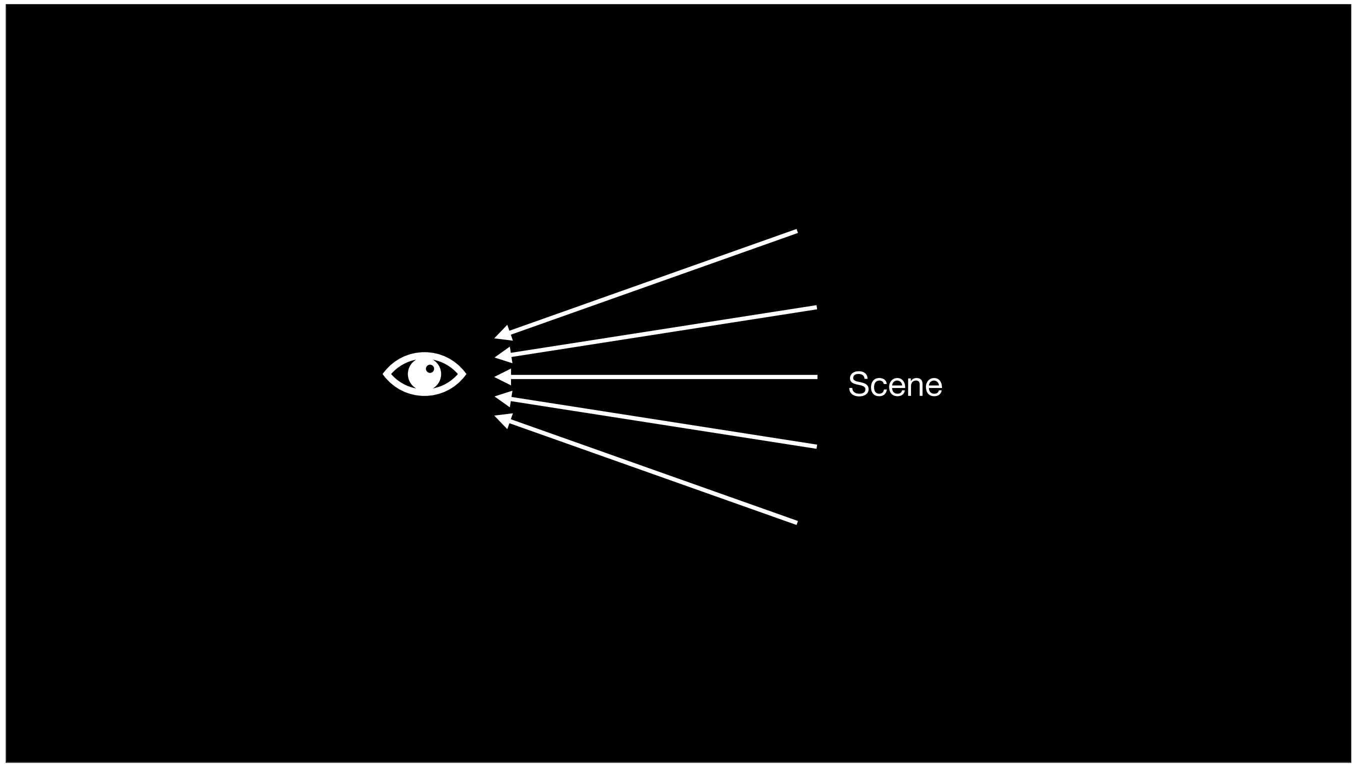
But cameras haven’t always been that way.
Let’s take a quick journey back in time through the history of cameras to help understand how they work.



What is a camera?

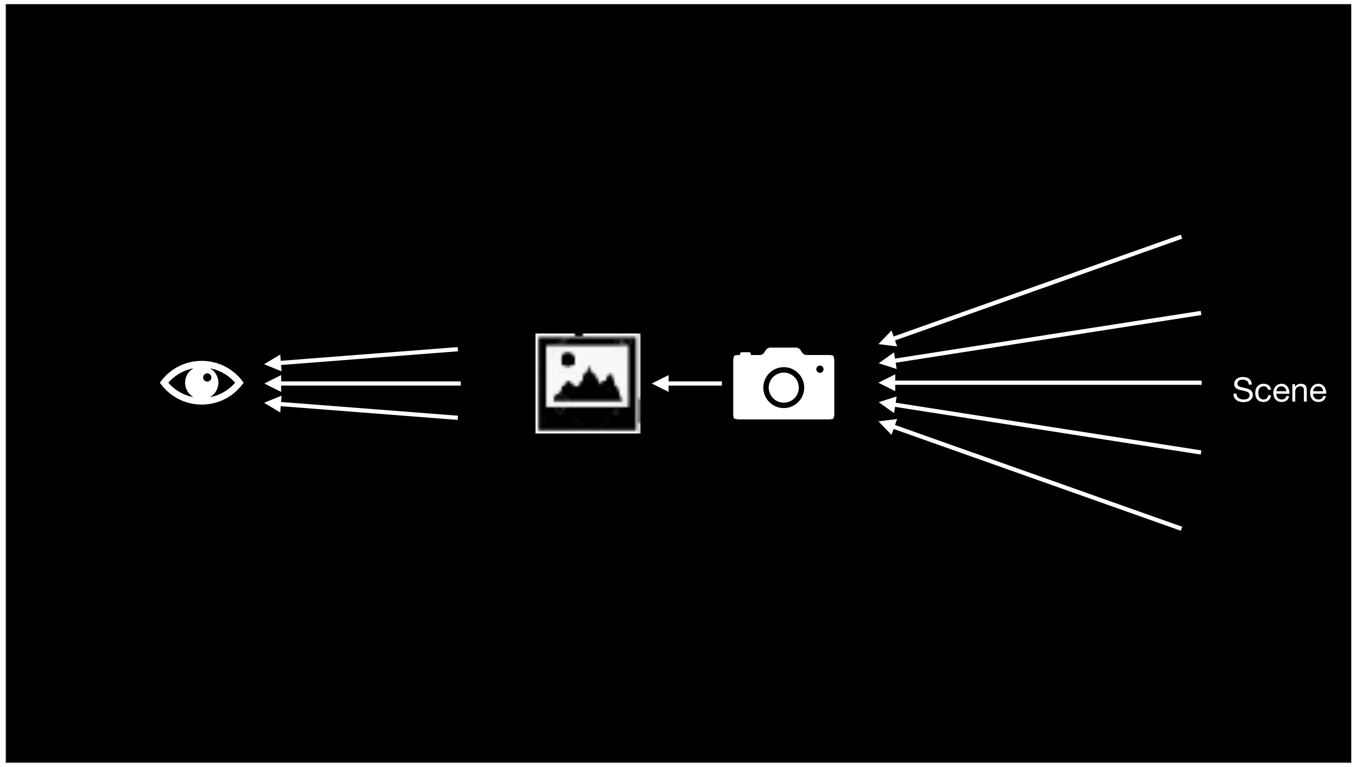
What is a photograph?

A good place to start would be to define what these two things are.



In an earlier lecture, we explained that we see things when light rays are captured by our eyes.

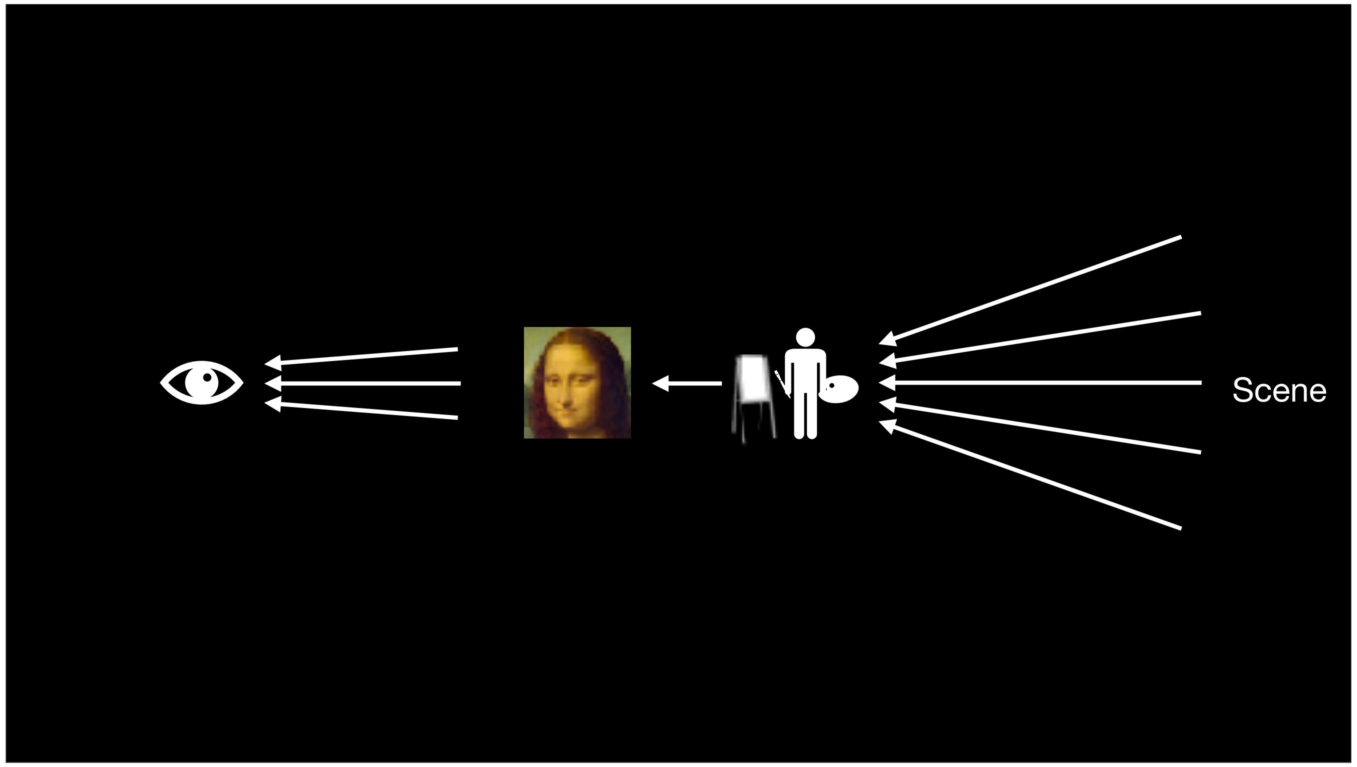
We can see cameras as an intermediate step in between our eyes and the world.



We can define a camera as any system which looks at the real world, captures the light rays, and packages them up in some way [slide], so that they can be seen by our eyes later. [slide]

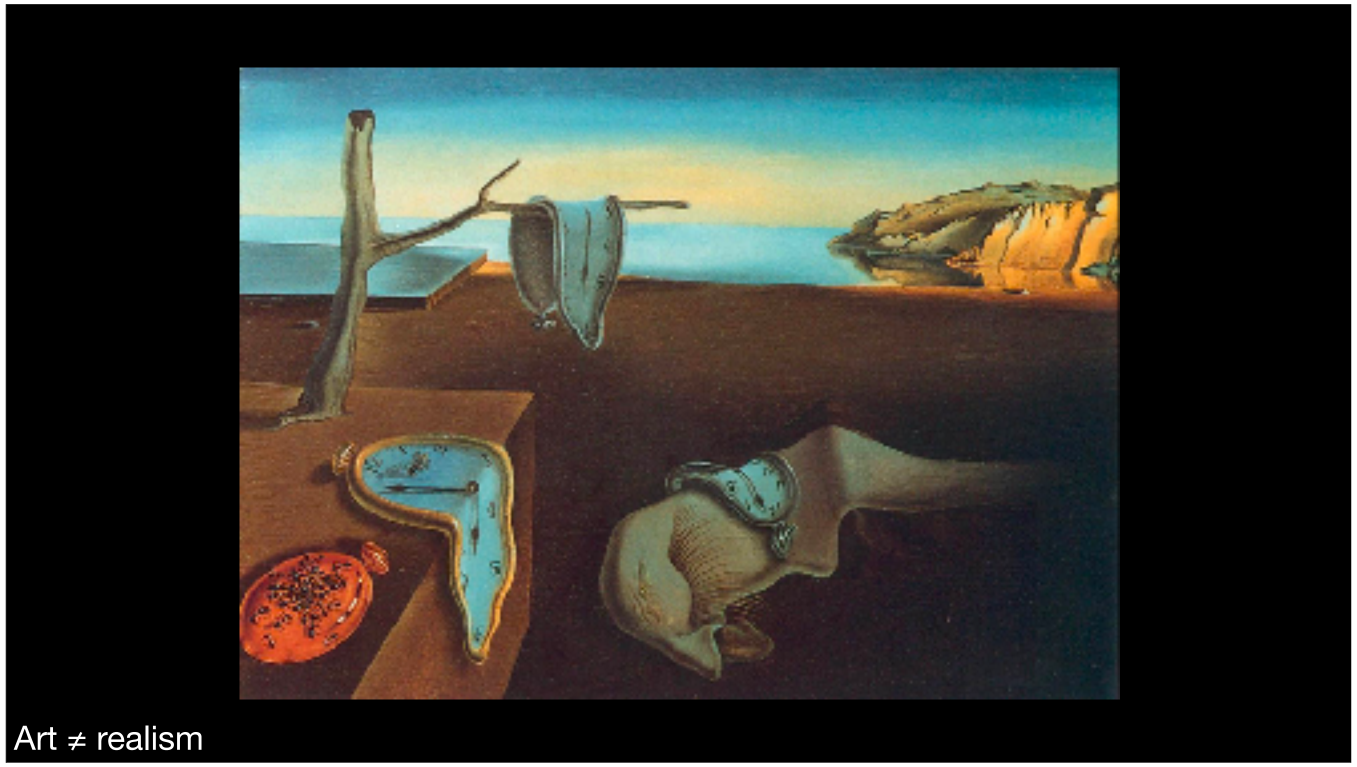
And this package is a photograph. A collection of information which tries to represent the light rays we would have seen when the picture was taken.

So if we go back to the *very* beginning….



The very first versions of cameras weren't even devices, they were humans. Like painters, artists. They did what a camera does.

They saw something and captured a representation of the scene that people could look at later.

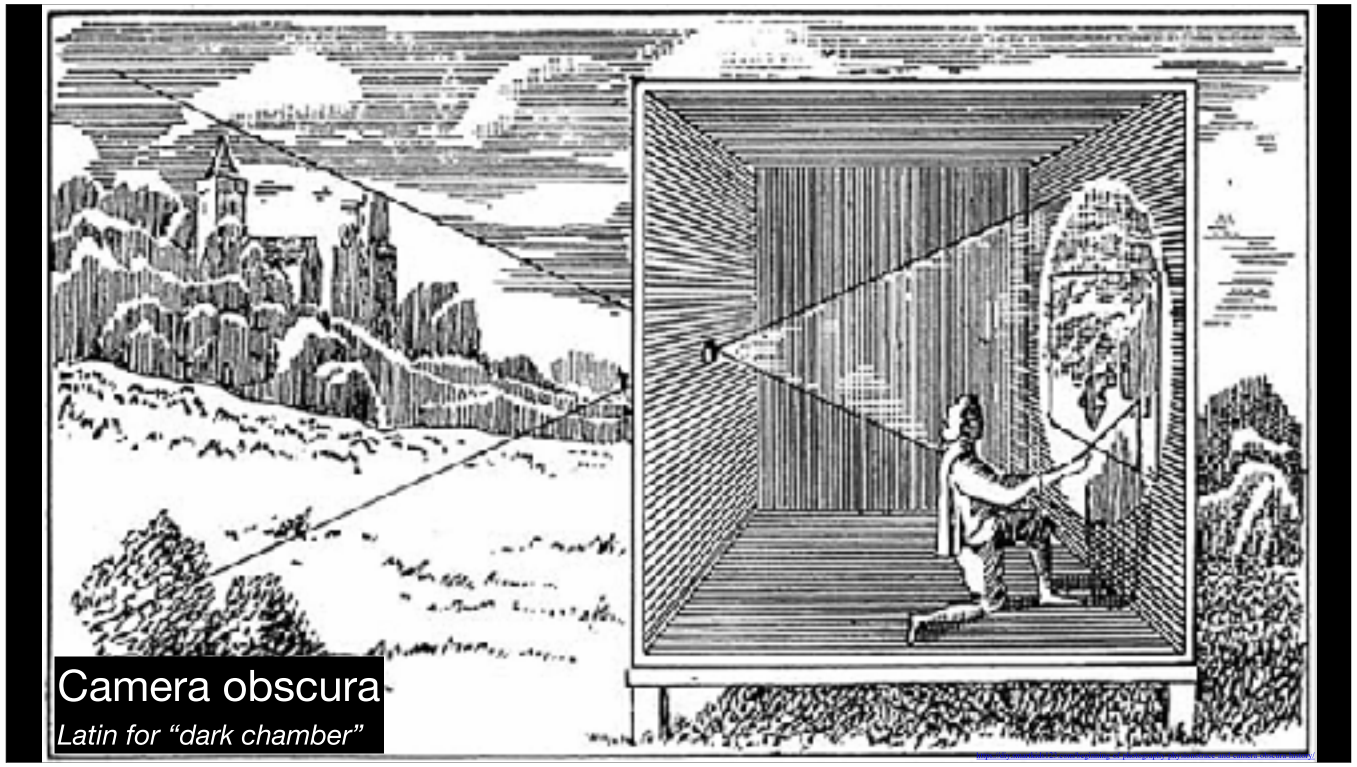


Art ≠ realism

But paintings aren’t always super realistic, right?

Realism is hard

And even when they are intended to replicate reality, it’s hard to perfectly capture subtle details, like shape, perspective, and shading.



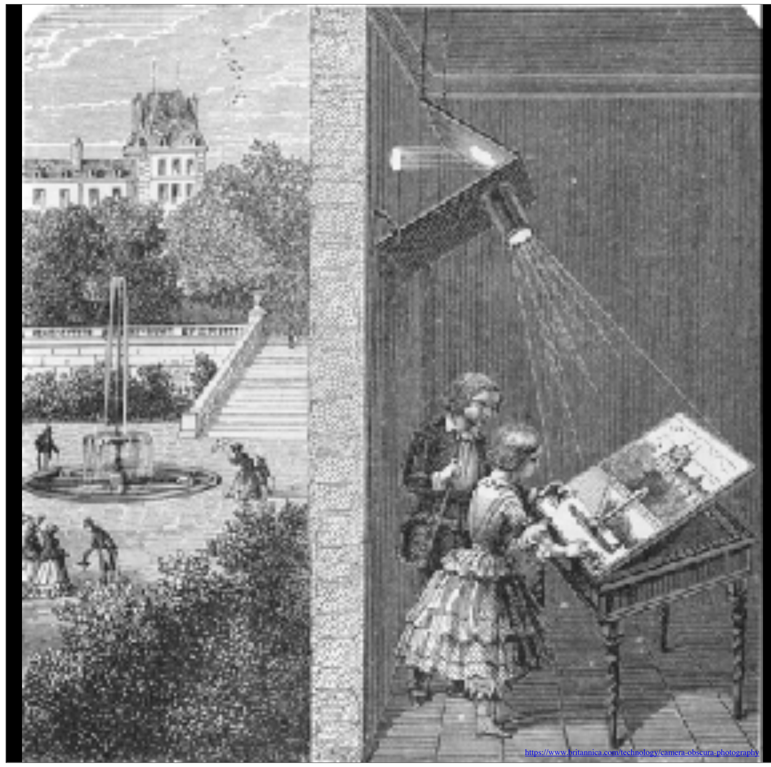
Camera obscura

Latin for "dark chamber"

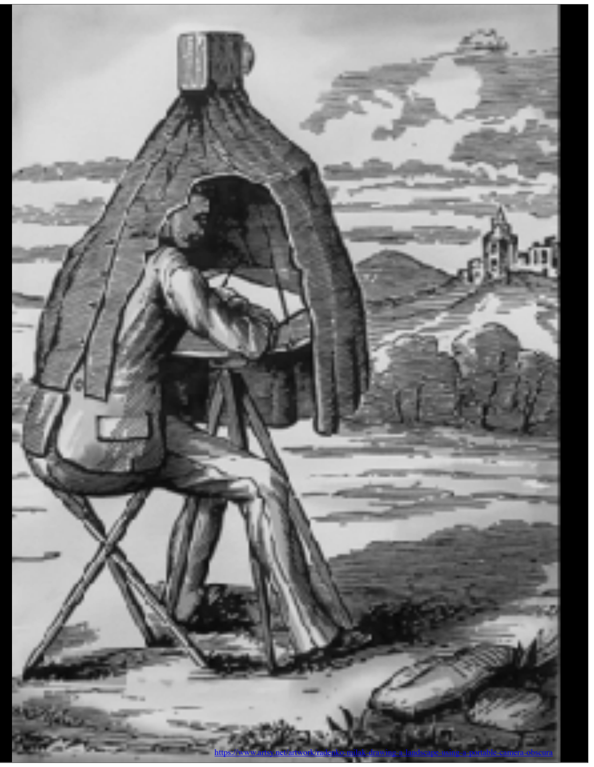
https://diy.smartkids123.com/beginning-of-photography-physionotrace-and-camera-obscura-history/

To make things easier, many artists began to use the technique known as 'camera obscura', which was effectively the pinhole camera we discussed in an earlier lecture.

They'd sit in a dark room, and a small hole would be cut in the wall, which would project the scene outside onto a painting surface, so that the artist could trace it.



https://www.britannica.com/technology/camera-obscura-photography



https://www.artsy.net/artwork/radenko-milak-drawing-a-landscape-using-a-portable-camera-obscura

This eventually developed into more advanced versions of the same thing, using mirrors, lenses, and even portable tents which the artist could take around with them.

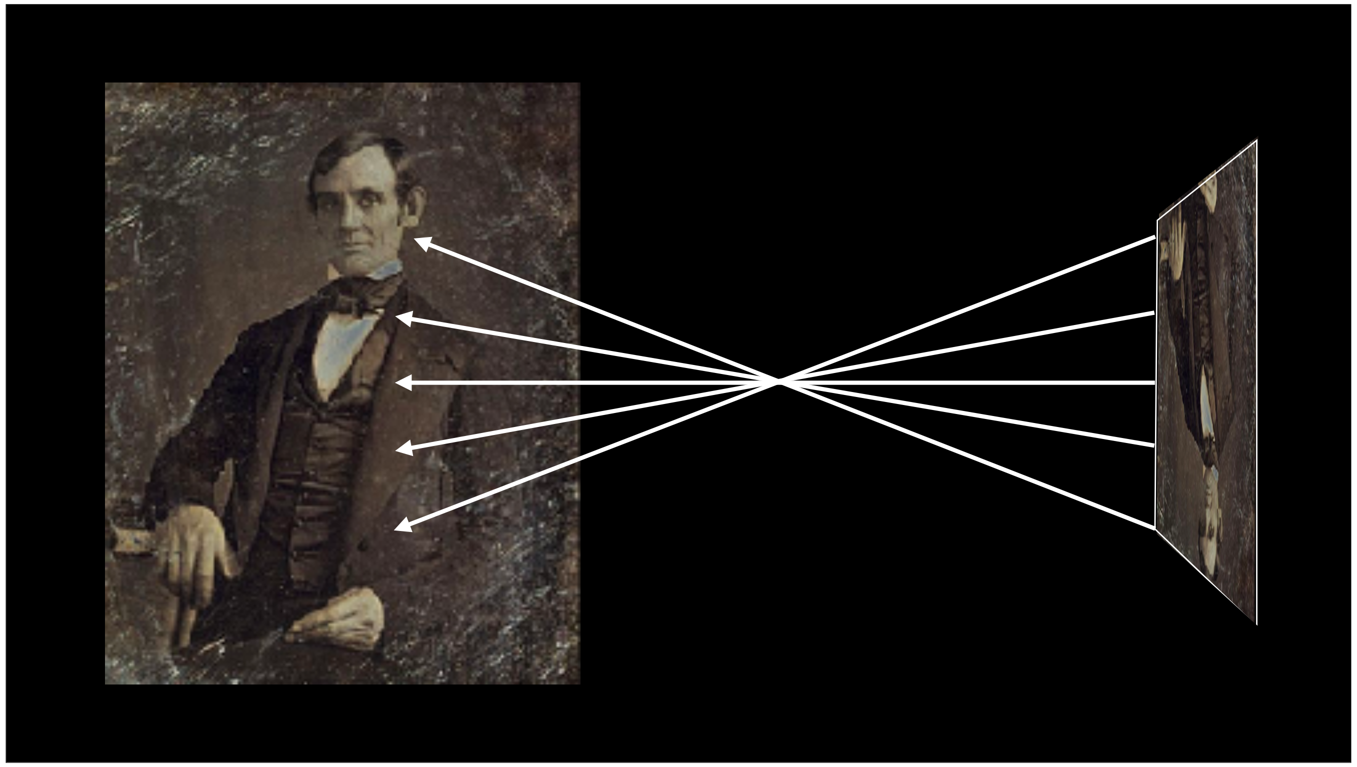
Dageurrotypes

Louis Daguerre & Nicéphore Niépce

But it wasn’t until the 1820s that a few artists and inventors realized that if you replaced the painting canvas in the camera obscura with a photosensitive plate [slide] (a plate that has chemical reactions when light touches it), you actually don’t have to do any drawing at all!

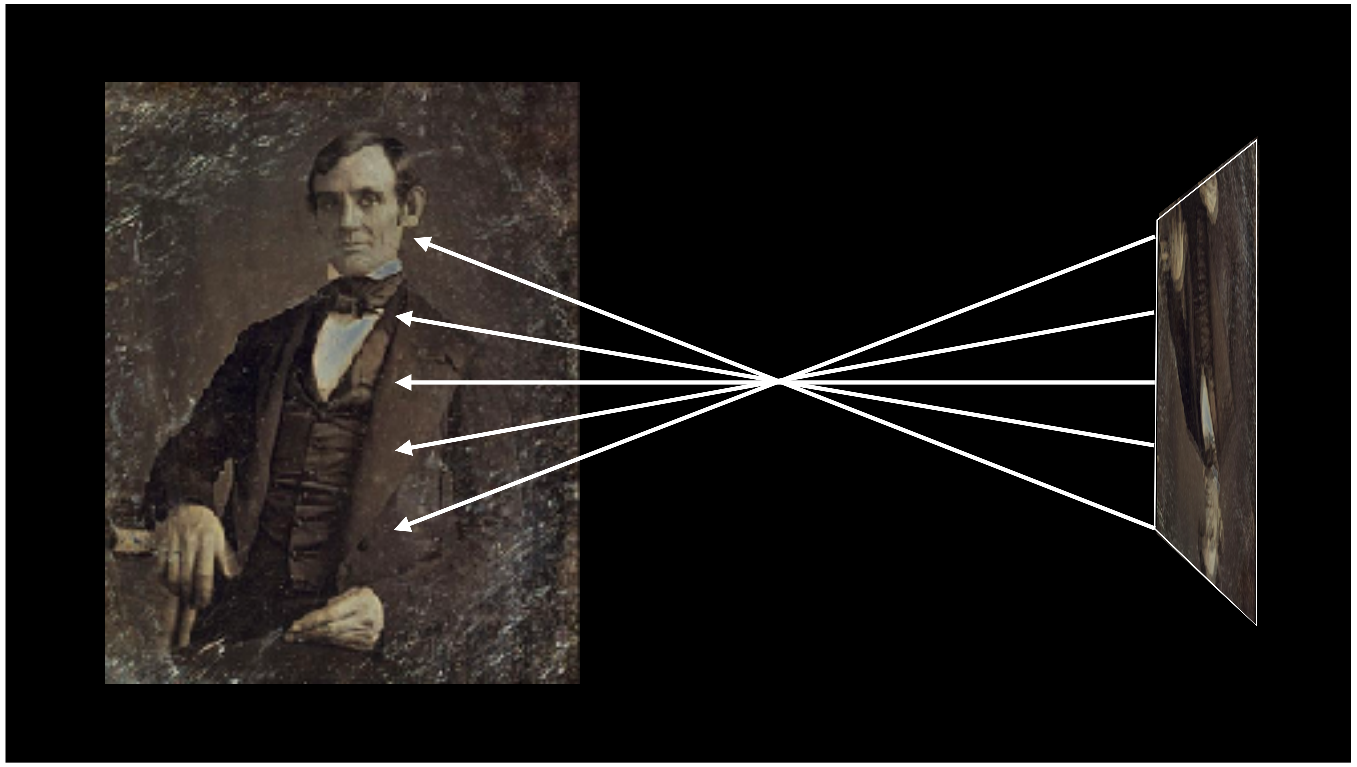
The plate would react more to brighter things, and less to darker things, so it would effectively capture light! [slide]

The most popular version of this camera system was called the daguerrotype, named after its inventor, Louis Daguerre.

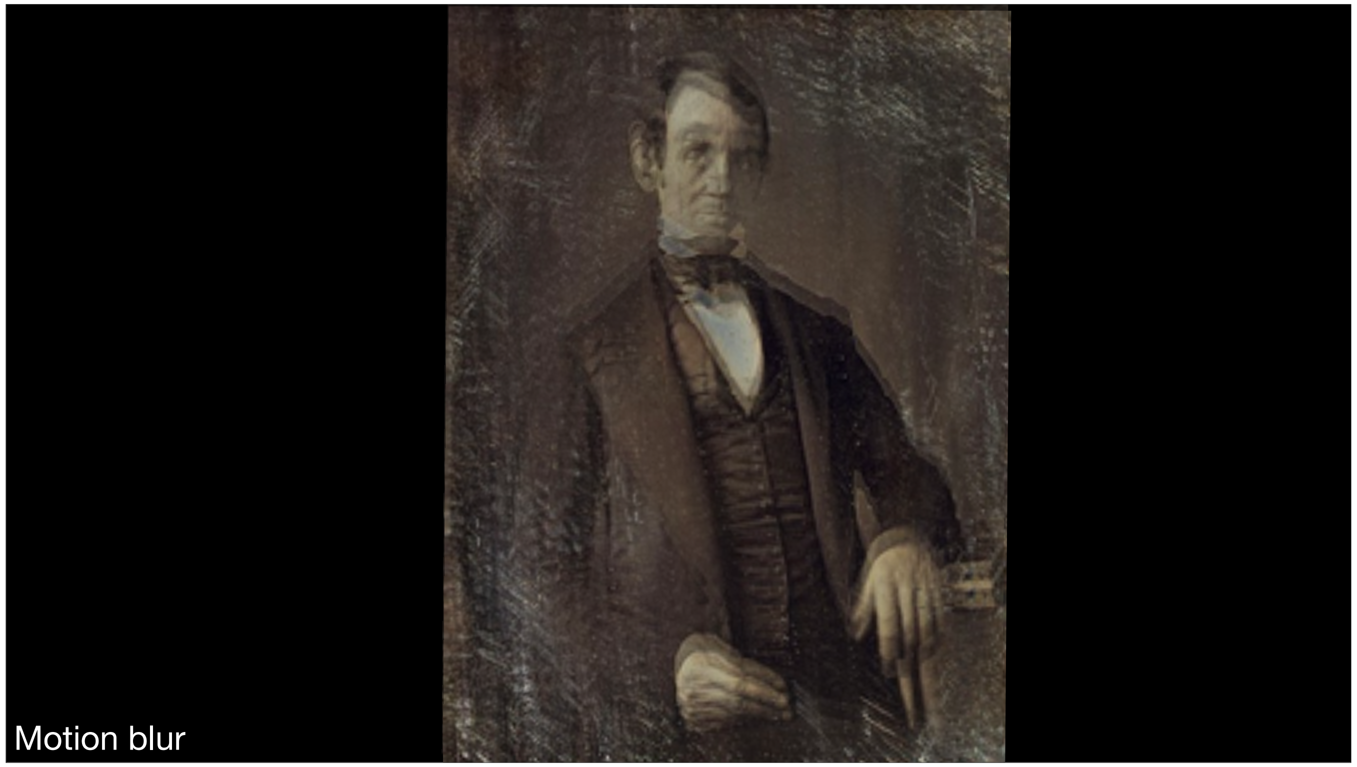


This was revolutionary, but it wasn't without its limitations. One of these limitations was exposure time.

Exposure time is the time that the photosensitive plate needs to be "exposed" to light in order for enough of a chemical reaction to occur such that the image becomes visible.
[slide]
Early versions required very long exposures, sometimes in the 10s of minutes, meaning that the subjects had to stay entirely still for very long. And if they didn't, the motion would cause the resulting image to look blurry.

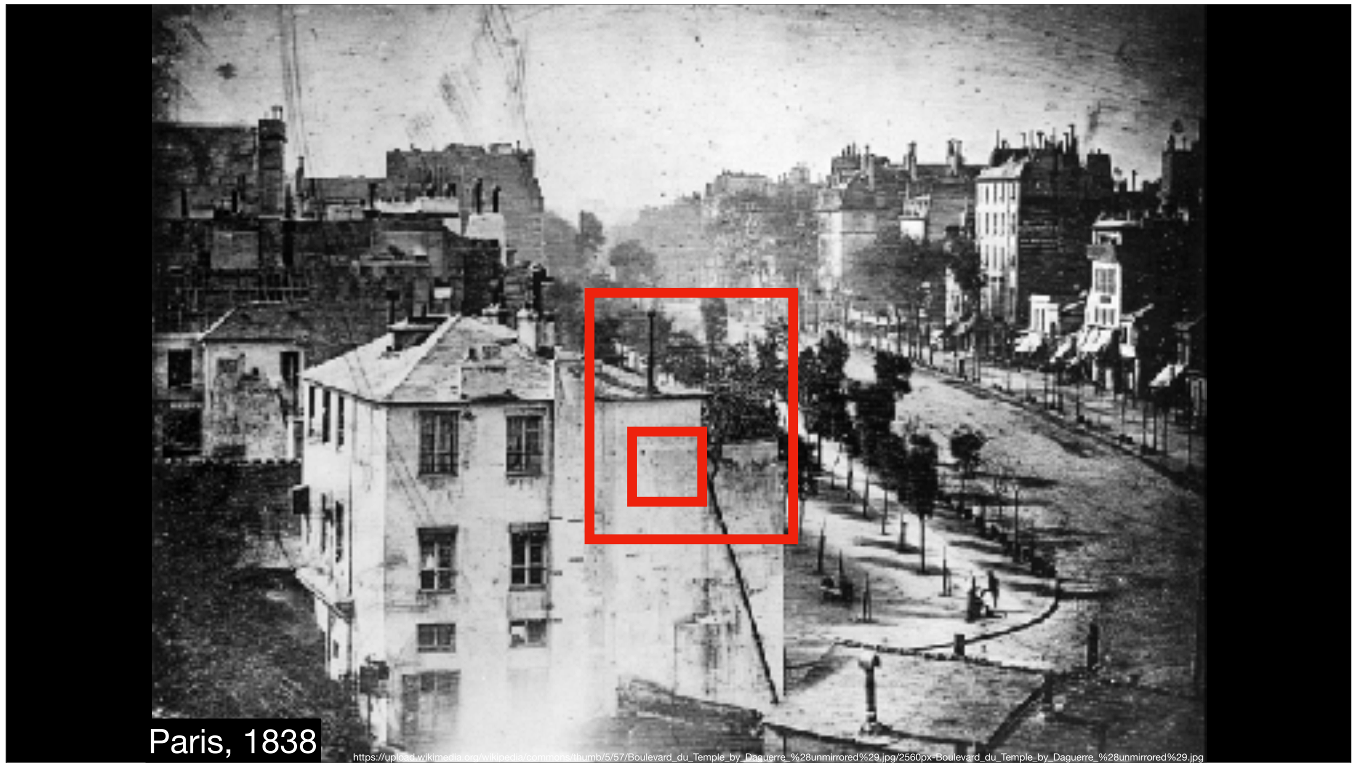


That’s because the photosensitive plate was just reacting to the light it saw. If the subject moved during the exposure, [slide] the camera would still be capturing light rays at the same angles, and would have no idea that the object emitting those light rays had moved, and it was now capturing something else. It would just add the two shifted things together, and the result would be a blurry image.



Motion blur

And that’s why a lot of people you see in older pictures aren’t smiling — because it’s hard to hold a smile for that long.



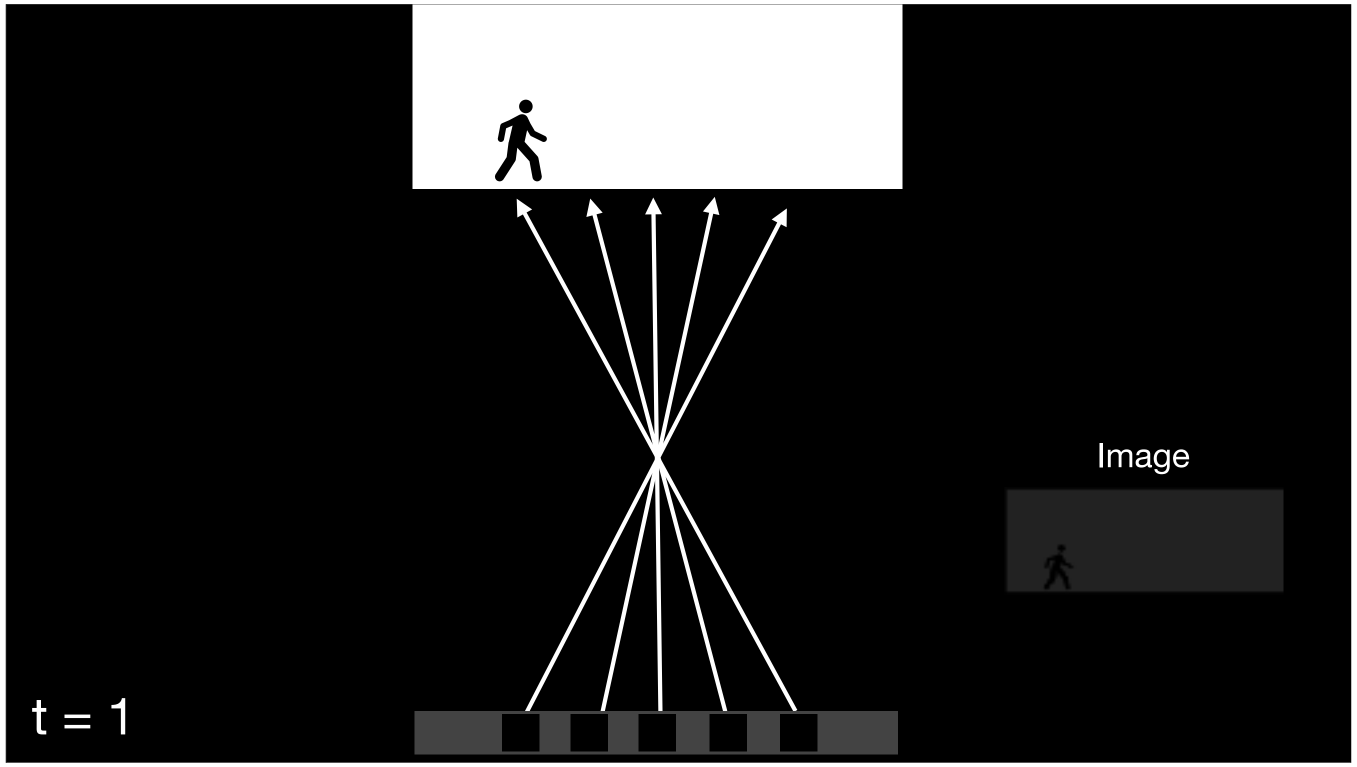
Paris, 1838

https://upload.wikimedia.org/wikipedia/commons/thumb/5/57/Boulevard_du_Temple_by_Daguerre_%28unmirrored%29.jpg/2560px-Boulevard_du_Temple_by_Daguerre_%28unmirrored%29.jpg

Here you can see an early picture captured by Daguerre in 1838, showing the streets of Paris.
[slide]
You can see the streets are suspiciously empty. This is basically for the same reason. When things move too fast, there isn't enough time for the difference in brightness between that object and its surroundings to have an effect on the photosensitive plate.

There are a couple other cool things to notice in this picture.
[slide]
Here you can see someone who stopped to get his shoe shined, so he was actually static for long enough to show up in the final image.
[slide]

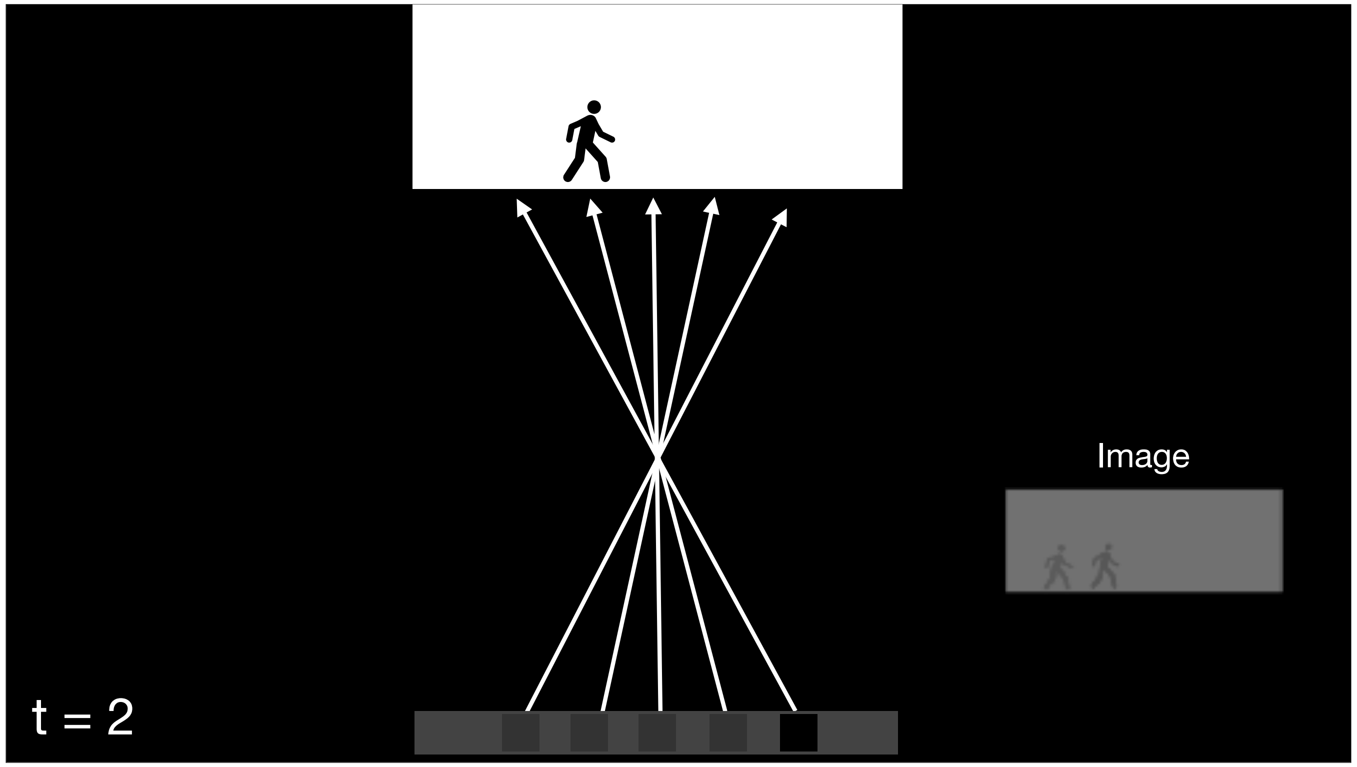
[slide]
And it's also speculated that in the top floor of the white building, there's a child looking out at Daguerre and his new camera.
[slide]



So this effect of disappearing objects, like the invisible people on the street, is effectively the same principle we saw in the blurry images.

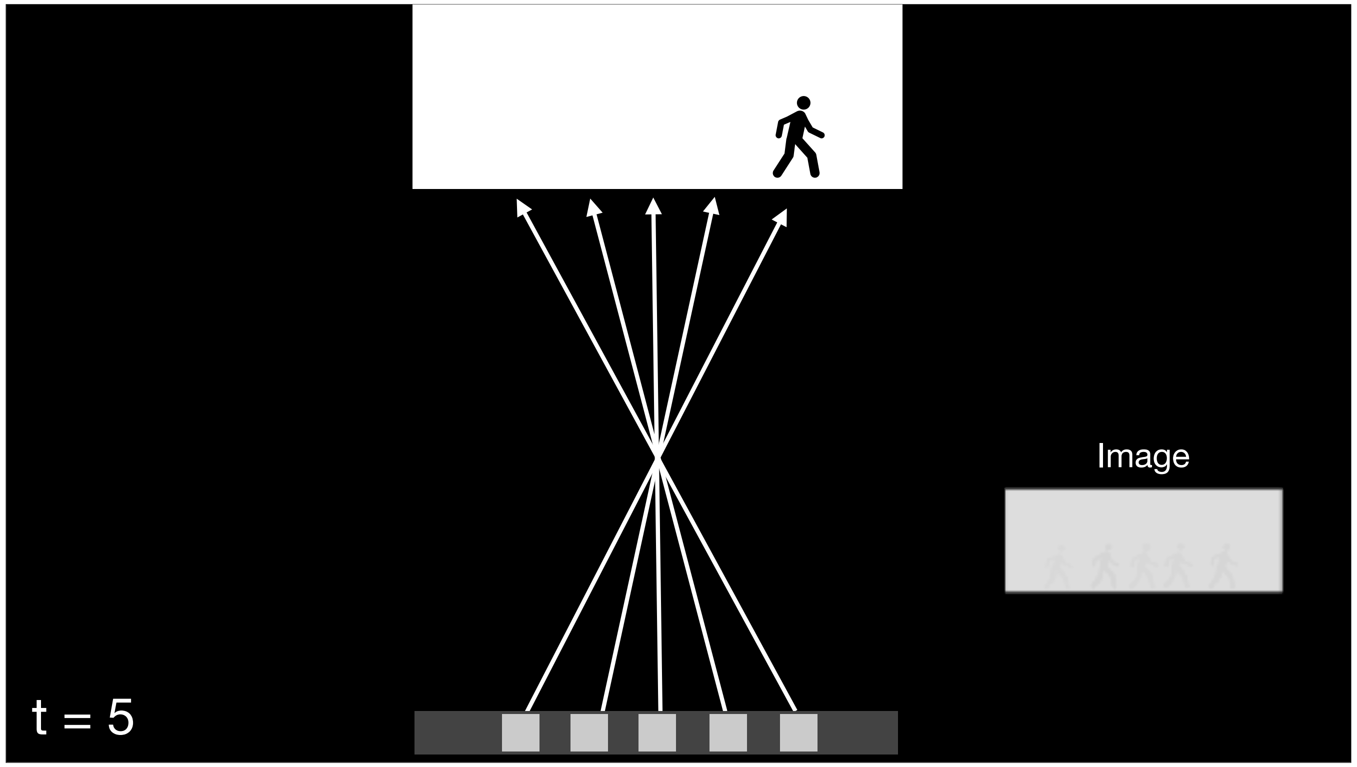
As an object moves through the scene, say this dark silhouette across a bright background, the longer our exposure time, the less likely it is that the moving object is going to make any impact on the final image.

So let’s say our exposure time is 5 seconds. [slide]
For the first second, the guy’s standing on the far left. We can see that during this time, the captured colors are clearly darker for the regions which correspond to the left part of the scene.



Then let’s say he moves over to the right and another second passes.
[slide]
We can see that now the one dark part of the image is not quite as dark anymore, because it has seen some of the bright background, and now there’s another part of the image which is also a bit darker.

And if this process repeats…



Eventually, by the end of it all, there’s almost no trace that this guy was even there, apart from leaving a bit of a dark trail along his path.

This principle is something people still take advantage of nowadays

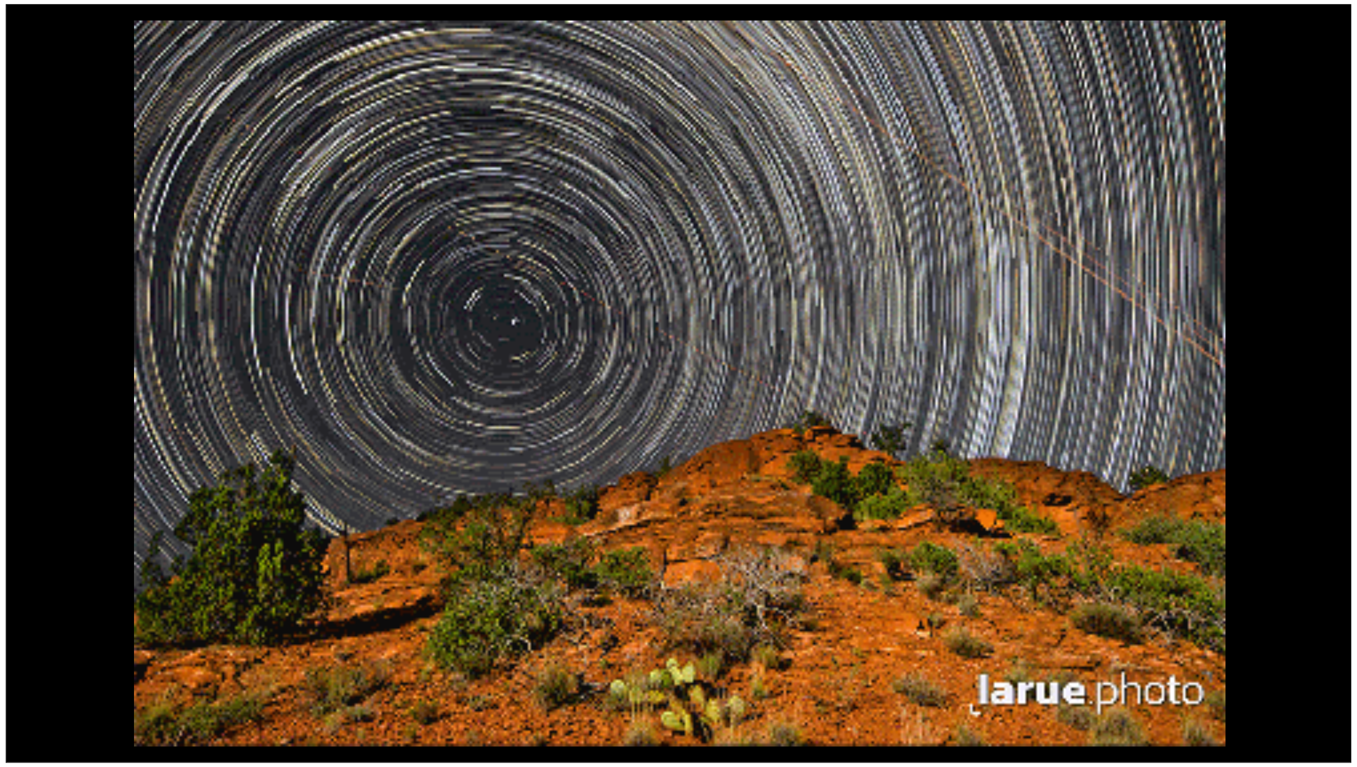
It’s often called long-exposure photography, and it can be used to create some pretty cool artistic effects.

Like here, if you take a very bright object, like a sparkler, and move it around a very dark scene, you can effectively “paint” patterns, or even [slide] draw words without the person who’s holding the light showing up in the image.

This principle is something people still take advantage of nowadays

It’s often called long-exposure photography, and it can be used to create some pretty cool artistic effects.

Like here, if you take a very bright object, like a sparkler, and move it around a very dark scene, you can effectively “paint” patterns, or even [slide] draw words without the person who’s holding the light showing up in the image.



The same goes for pictures you see of starry night skies that look like they're twirling, it's just a long exposure photograph with the natural orbit and rotation of the earth.

1830s

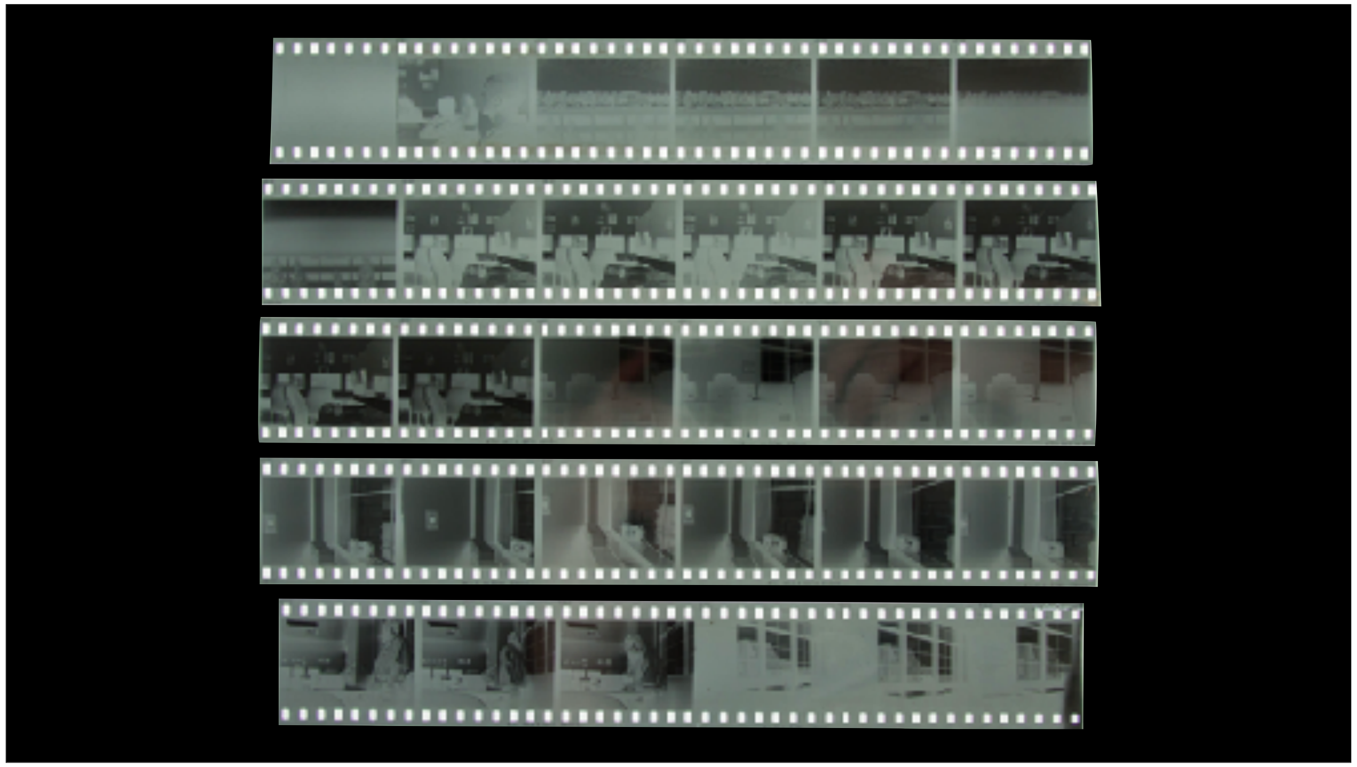
Back on track, with the history of cameras.

Over the years, cameras changed a lot.

1925

They started using portable film instead of big expensive metal plates.

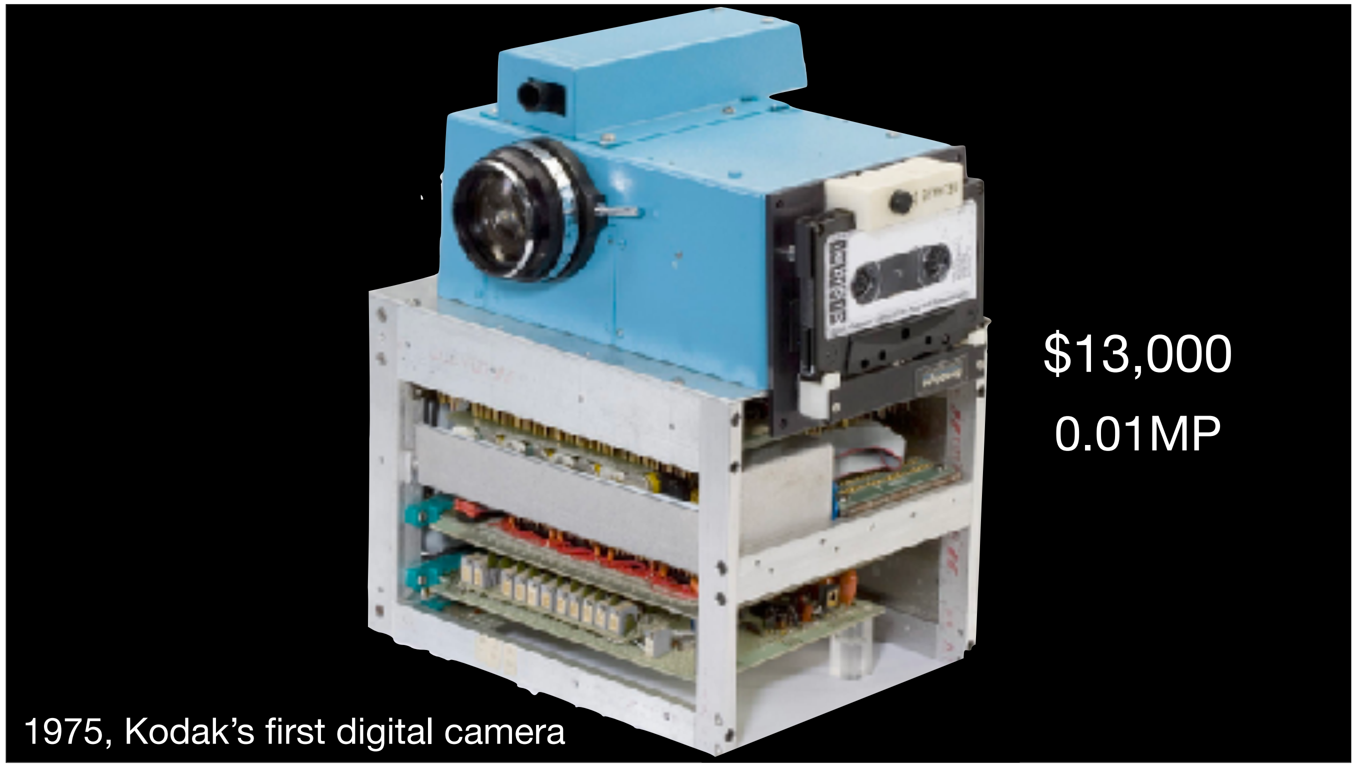
They also started printing on negatives.



That was an important milestone too — because you can develop negatives into actual photographs more than once, and that means you can make copies of photos, and that's something you couldn't do before.

Instant cameras

Over time, the process of taking and developing photographs eventually became faster, and even instant!



$13,000

0.01MP

1975, Kodak’s first digital camera

Eventually, with the advent of computers, cameras also became digital.

At first, they were incredibly expensive — things you’d only really see in research labs.

This one from Kodak was 13,000 dollars, and captured images with 0.01 megapixels.

$2000

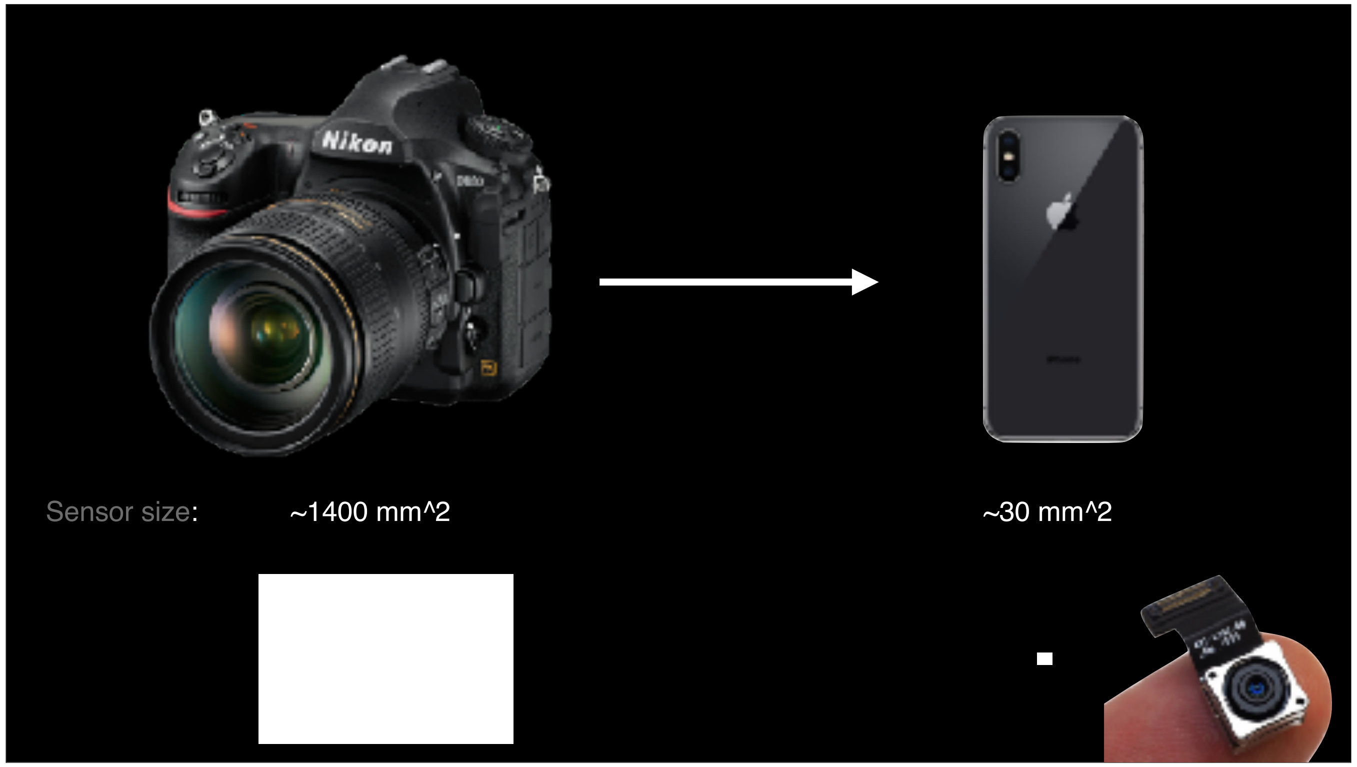
46MP

Today

For reference, the top-of-the-line modern consumer camera only costs about 2000 dollars, and will capture 46 megapixels, which is just under 5000x the number of pixels.

An impressive evolution!

But the single step in the history of the camera which we'll focus on the most in this lecture is the transition to having cameras in our smartphones.



Sensor size: ~1400 mm^2 ~30 mm^2

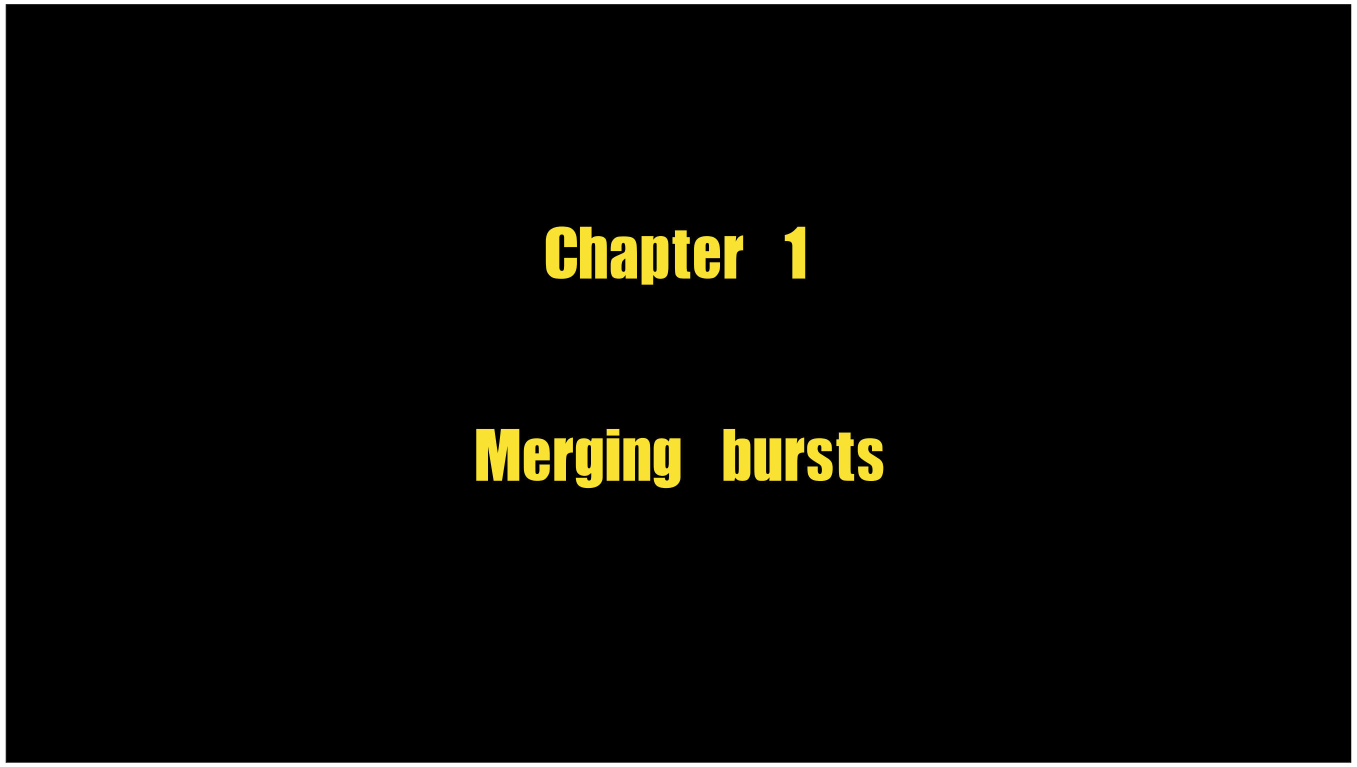
I think it’s worth diving into this transformation a bit because getting a camera to fit in your phone means that it has to be *really* small.

For reference, comparing to one of the newer iPhones, the sensor of the DSLR (the component which is actually capturing the light) [slide] is roughly 50 times larger.

[slide] Here’s what a camera from a smartphone looks like, on the tip of a finger. It’s tiny!

In order to make something that small, a few tradeoffs have to be made. But since our cameras are digital, and we can inspect and analyze pixel values, we can use computers and algorithms to find ways around some of these limitations!

In the rest of the talk, I’m going to go through a few clever solutions to some of these problems that are caused by putting cameras in our phones.

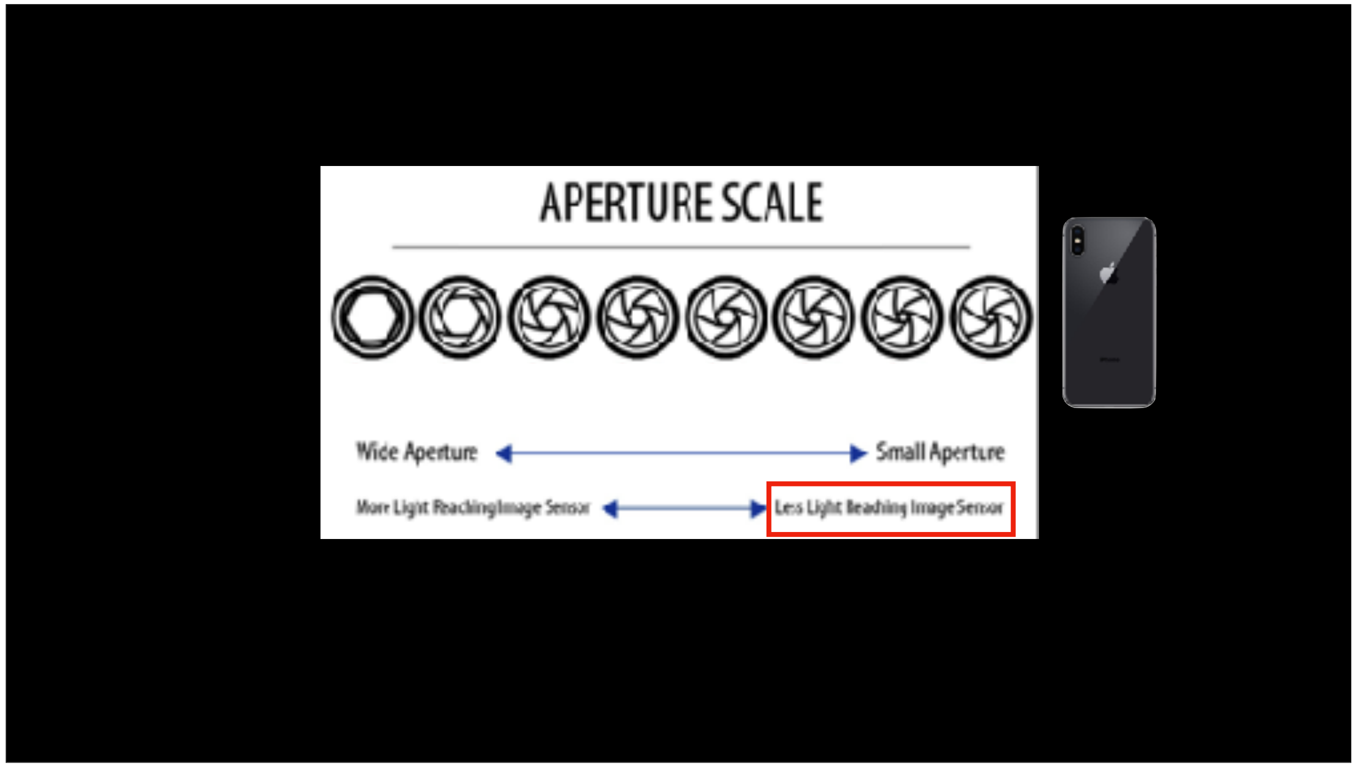


Chapter 1

Merging bursts

This brings us to chapter 1:

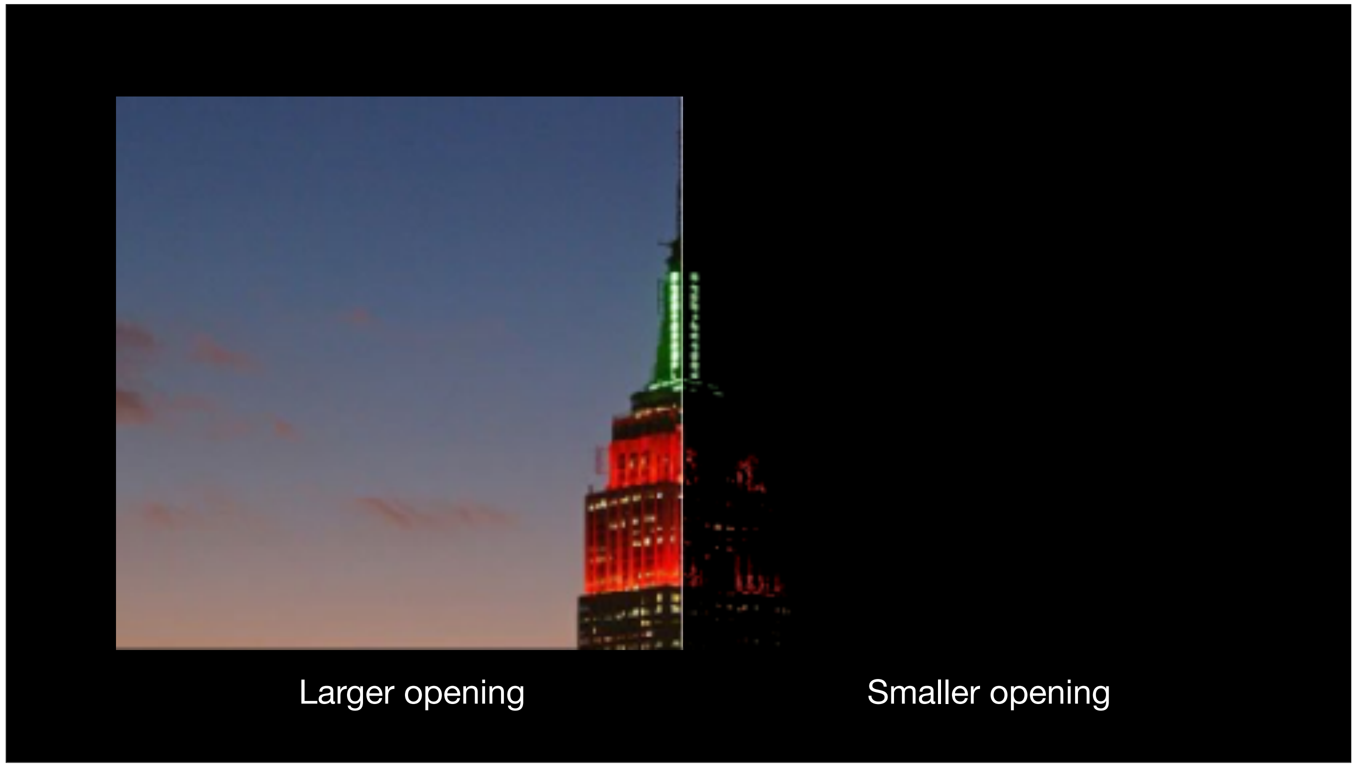
Merging bursts.


APERTURE SCALE
Wide Aperture
Small Aperture
More Light Reaching Image Sensor
Less Light Reaching Image Sensor

One of the things that a smaller phone intrinsically has to have is a smaller maximum aperture size, which is the maximum size of the camera opening when taking a picture.

And a smaller aperture also means that there's less overall light [slide] reaching the image sensor.

[P.S.]
For the rest of this section, I'm going to be calling this the opening size, and I'm specifically not using the word aperture because there are a couple meanings, and I don't want to cause any confusion — aperture in photography can refer to relative aperture, or how much you're choosing to open or close the shutter, but what we're talking about here is the maximum aperture size, or the maximum opening size when comparing two different cameras. (i.e. smartphones have a smaller maximum opening size). But the same principles we're explaining are valid among two concepts, both relative aperture and maximum aperture size.

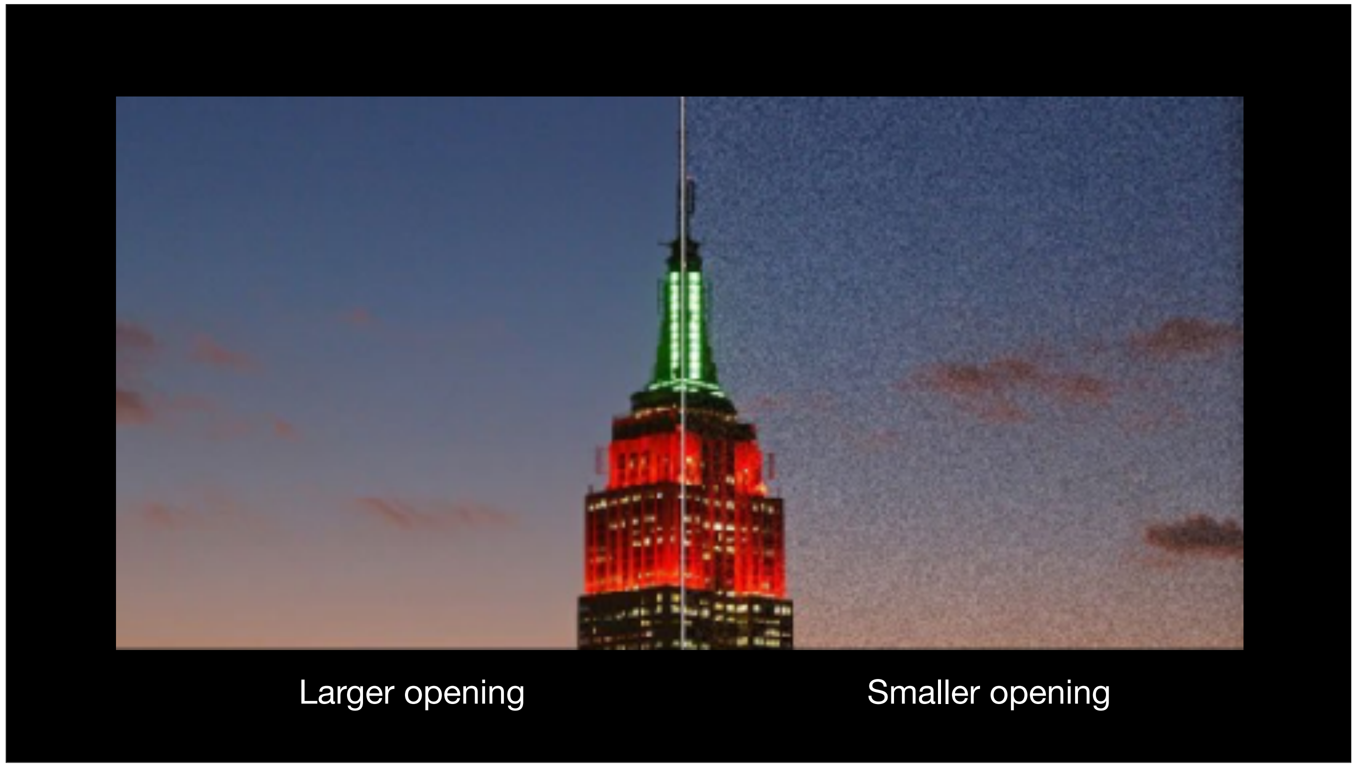


Larger opening

Smaller opening

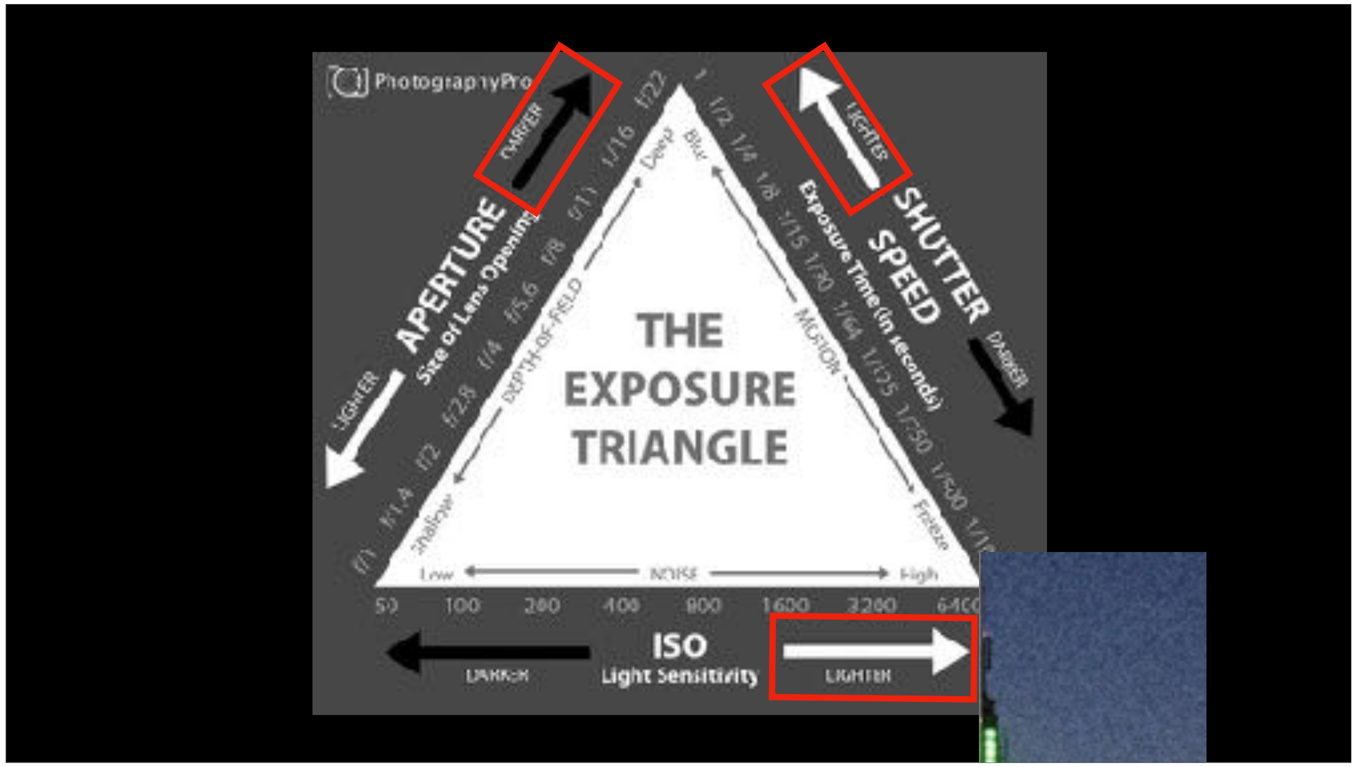
If we have an image captured with a larger opening, the same scene captured with a smaller opening is going to be darker.

We can apply gain to that same photograph, to try to match the other one [slide] but the result is going to be a very noisy image.



If we have an image captured with a larger opening, the same scene captured with a smaller opening is going to be darker.

We can apply gain to that same photograph, to try to match the other one [slide] but the result is going to be a very noisy image.



We're forced to have a small camera opening size [slide], and that makes the image dark.

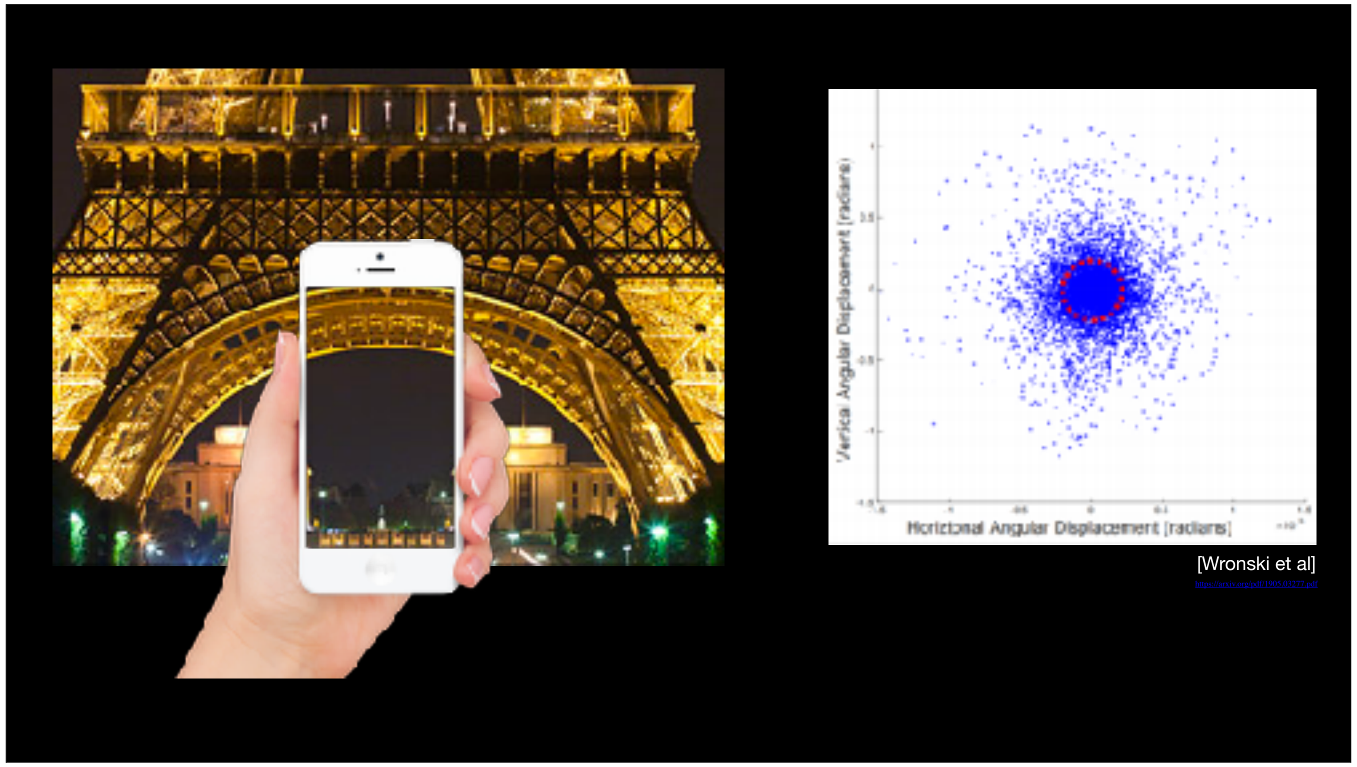
We've tried to compensate for this by increasing the sensitivity to light [slide], by applying gain to the signal, but that resulted in a noisier image.

Another way around this is to extend the exposure time [slide]

So if we just expose the image for longer, the image will be brighter, right?

Yes — but we should only do this if we have a tripod.

But we're talking about camera phones, and we don't want to carry around a tripod to take pictures. It ruins the convenience of the form factor.

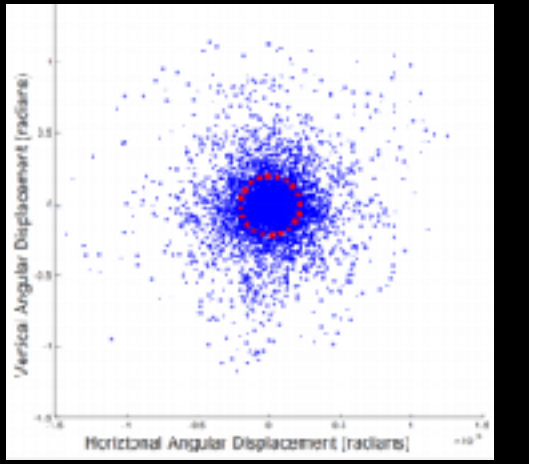


[Wronski et al]

https://arxiv.org/pdf/1905.03277.pdf

If we don’t have a tripod, that means we’re holding the camera in our hands.

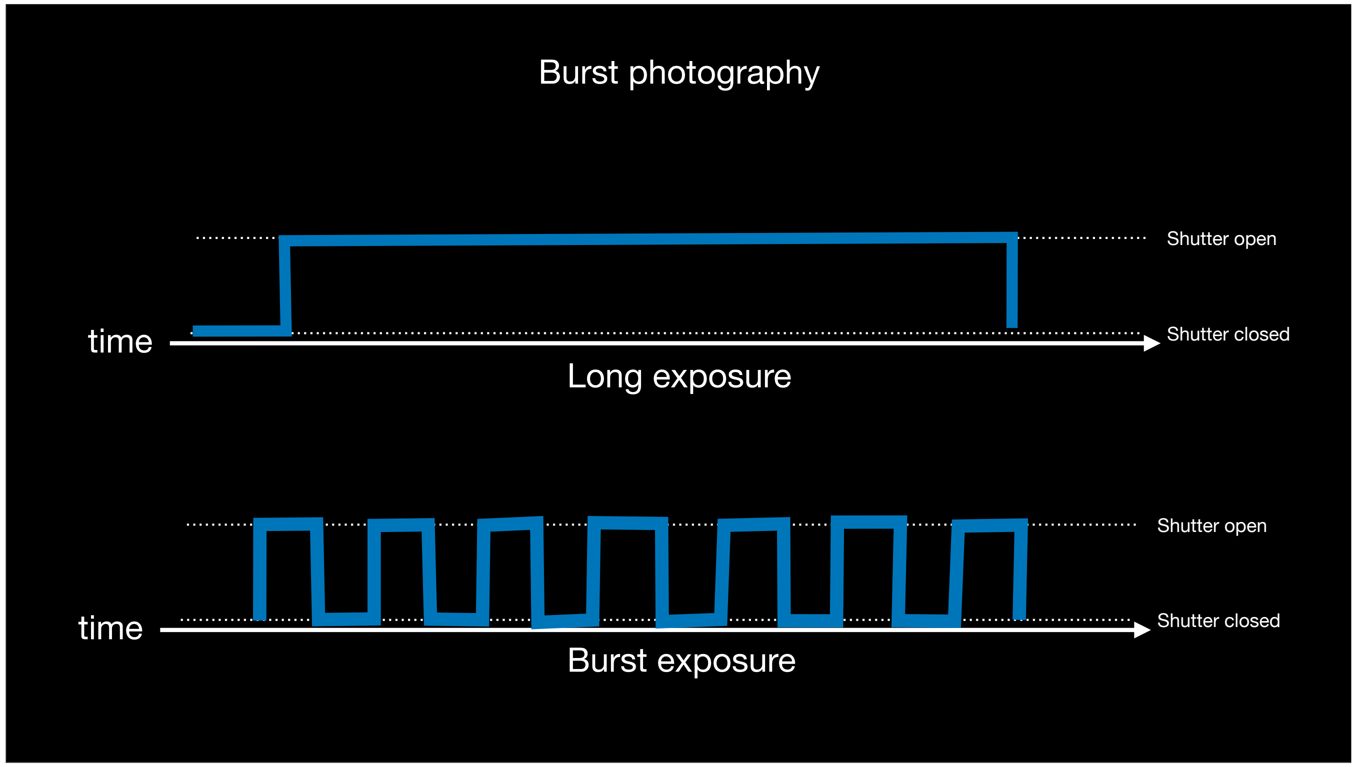
And no matter what we do, our hands are constantly moving, ever so slightly.
It’s totally natural, but it makes it impossible to hold the camera entirely still for long enough to get good picture, and we end up again with the same blurry image effect we saw before. [slide]


Vertical Angular Displacement [radians]
Horizontal Angular Displacement [radians]
1
0.5
0
-0.5
-1
-1.5

[Wronski et al]
https://arxiv.org/pdf/1905.03277.pdf

So what’s the answer to this problem?

Burst photography

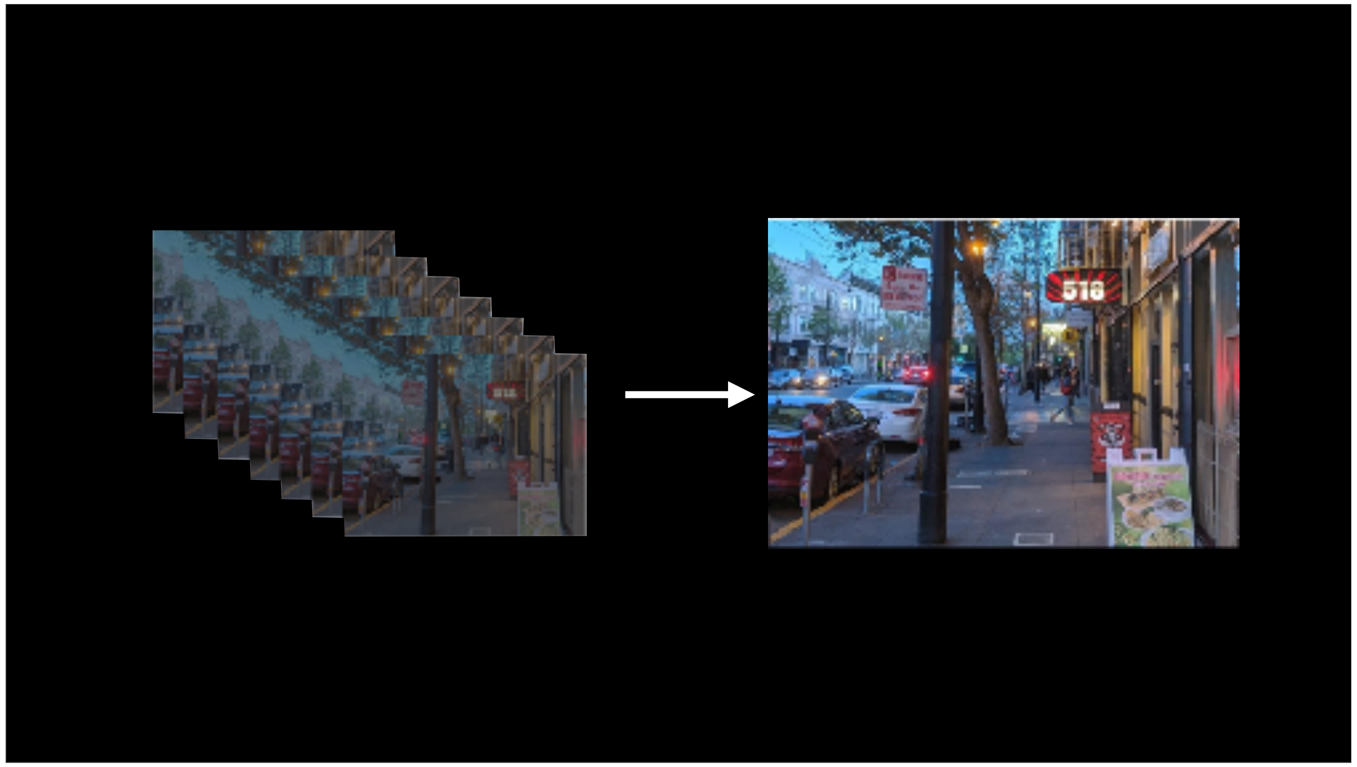


Burst photography

In contrast to a single long exposure [slide], which opens the shutter at one point and leaves it open for long enough to capture the light rays, capturing a brighter, but blurrier image…

[slide]

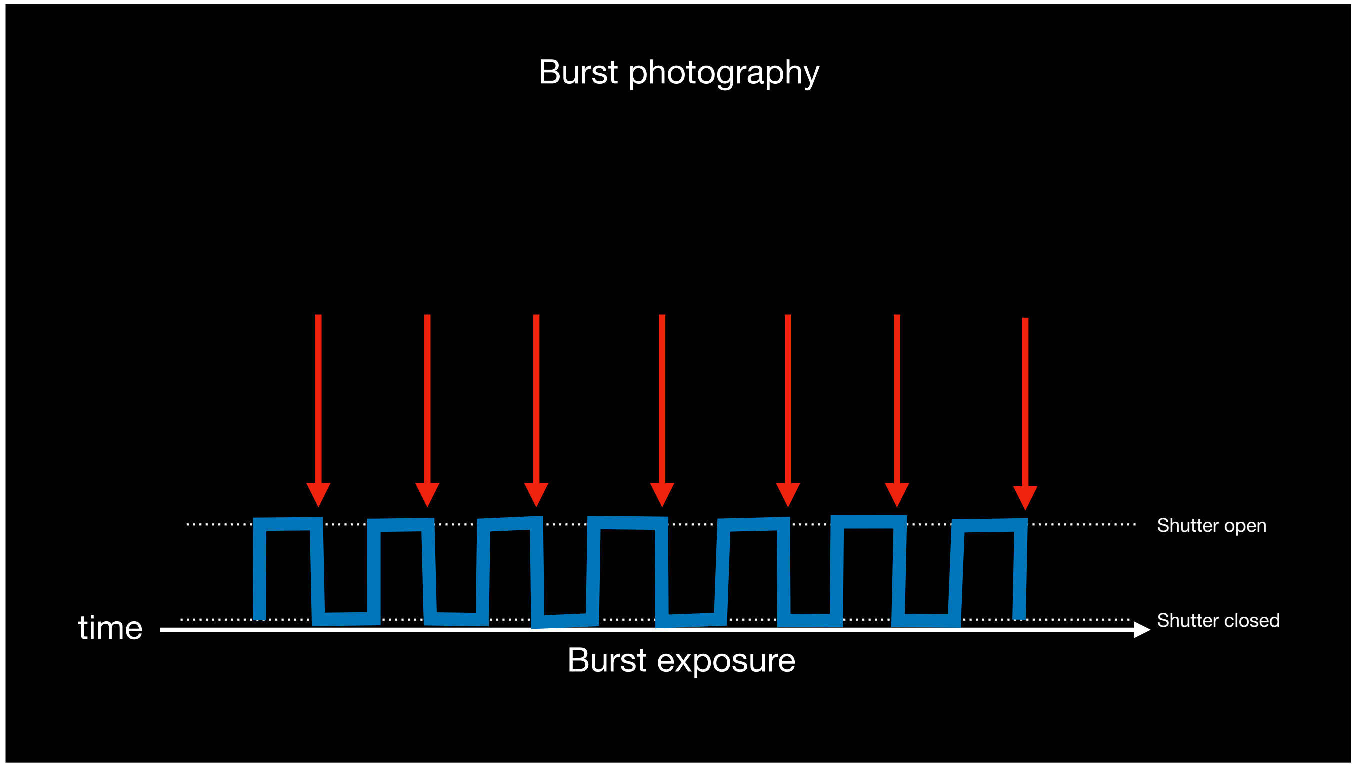
Burst photography will open and close the shutter several times in the same amount of time, and will effectively have captured many images, which are much darker, but won't be blurry.



Problem solved, right?

Add those darker images together, and you’ve got your final image.

Not quite! Remember the camera motion.



At each of these pictures in the burst, the camera may have already shifted, so just adding together the images will still end up being blurry.

What we need to do first is align the images.

This can be done using a lot of the same techniques discussed in previous lectures, either through feature point matching, or through dense optical flow estimation.

[Wronski et al]
https://arxiv.org/pdf/1905.03277.pdf

So here’s an example burst sequence captured by a phone, and it looks like there’s not much motion, but when you zoom in…

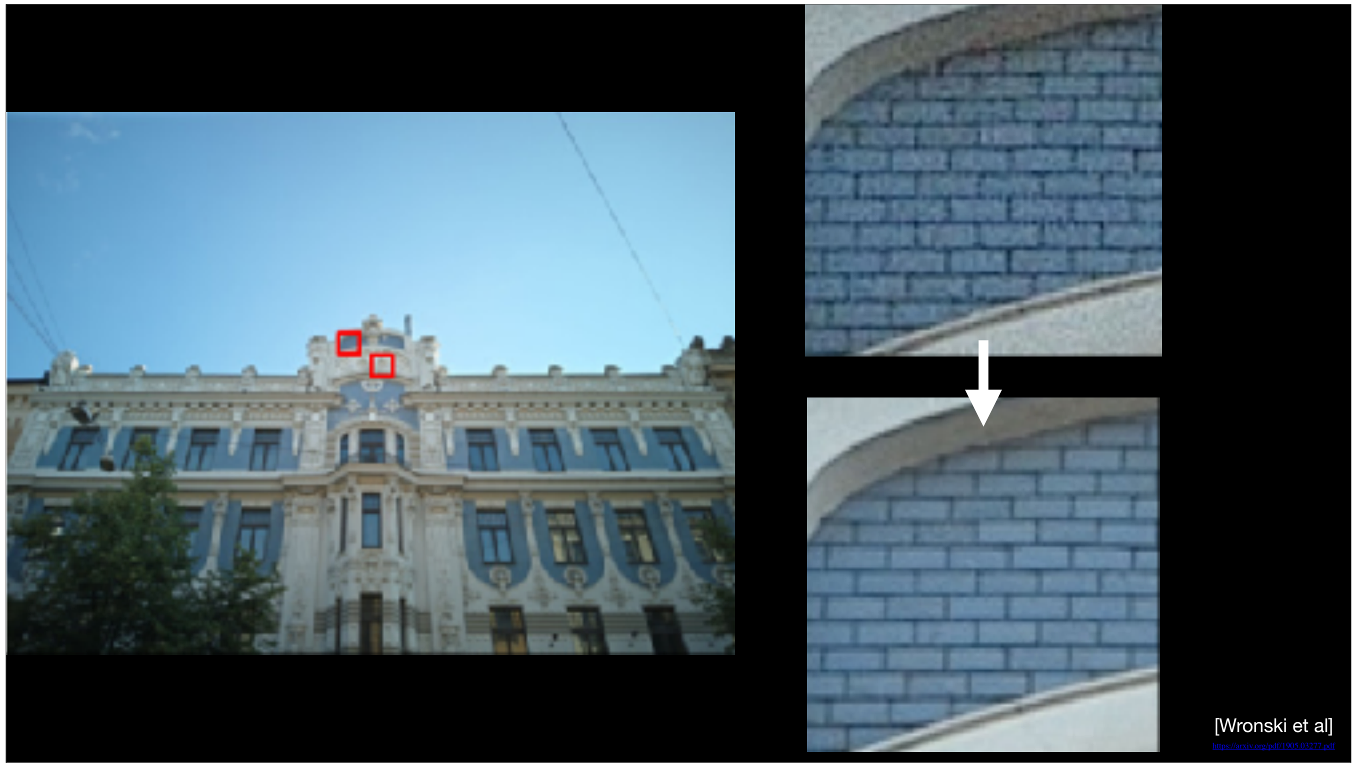
[Wronski et al]
https://arxiv.org/pdf/1905.03277.pdf

You’ll see that the camera is shaking quite a bit, due to the hand motion.

[Wronski et al]
https://arxiv.org/pdf/1905.03277.pdf

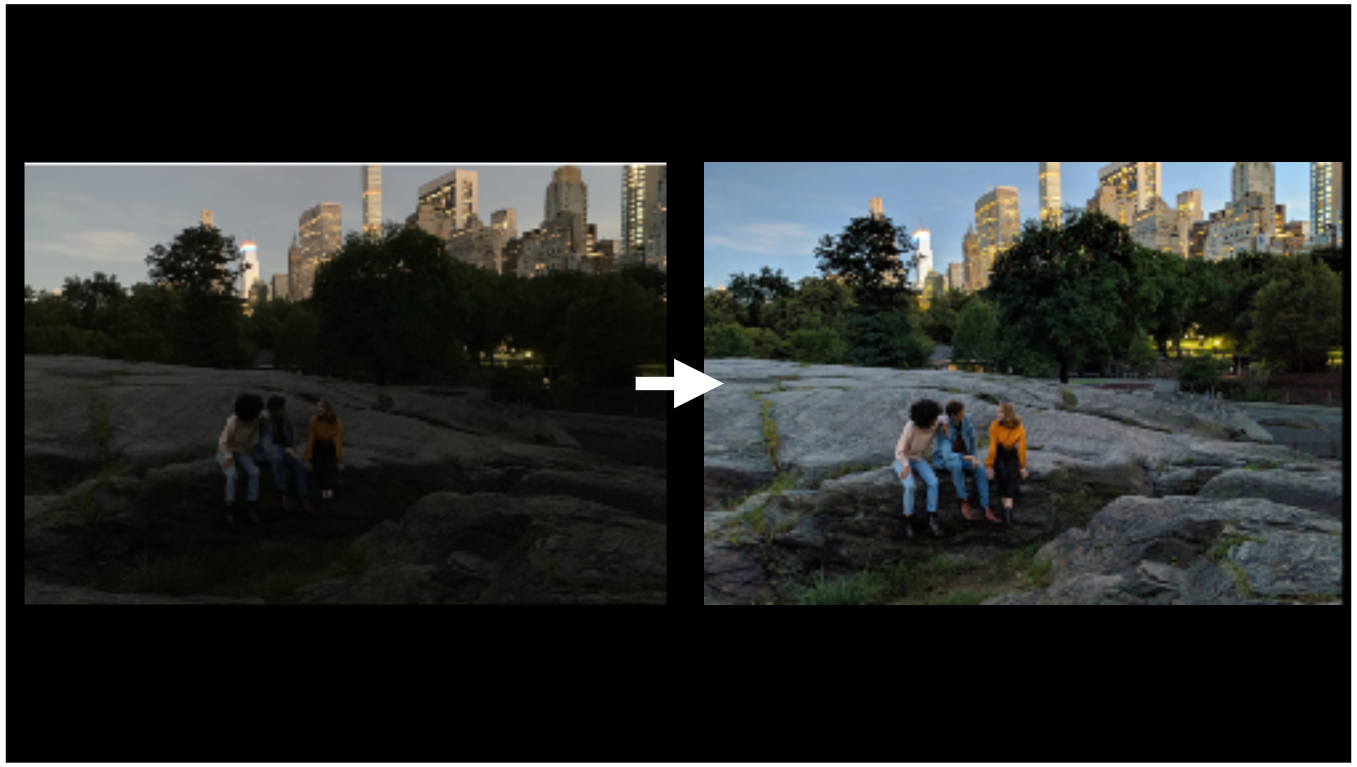
By doing optical flow alignment, we can get the frames to be nearly identical, with the only real difference between them being the noise.

And since the noise is effectively Poisson noise, and isn’t correlated across frames, if we average all of these captures frames, we get a much cleaner image as a result.



[Wronski et al]
https://arxiv.org/pdf/1905.03277.pdf

You can see this process significantly reduces the noise for these small red patches in the image.



Since this process effectively averages out the noise, we can now safely boost the signal of our images without worrying too much about amplifying the noise. And that lets us more easily take pictures when it's too dark.

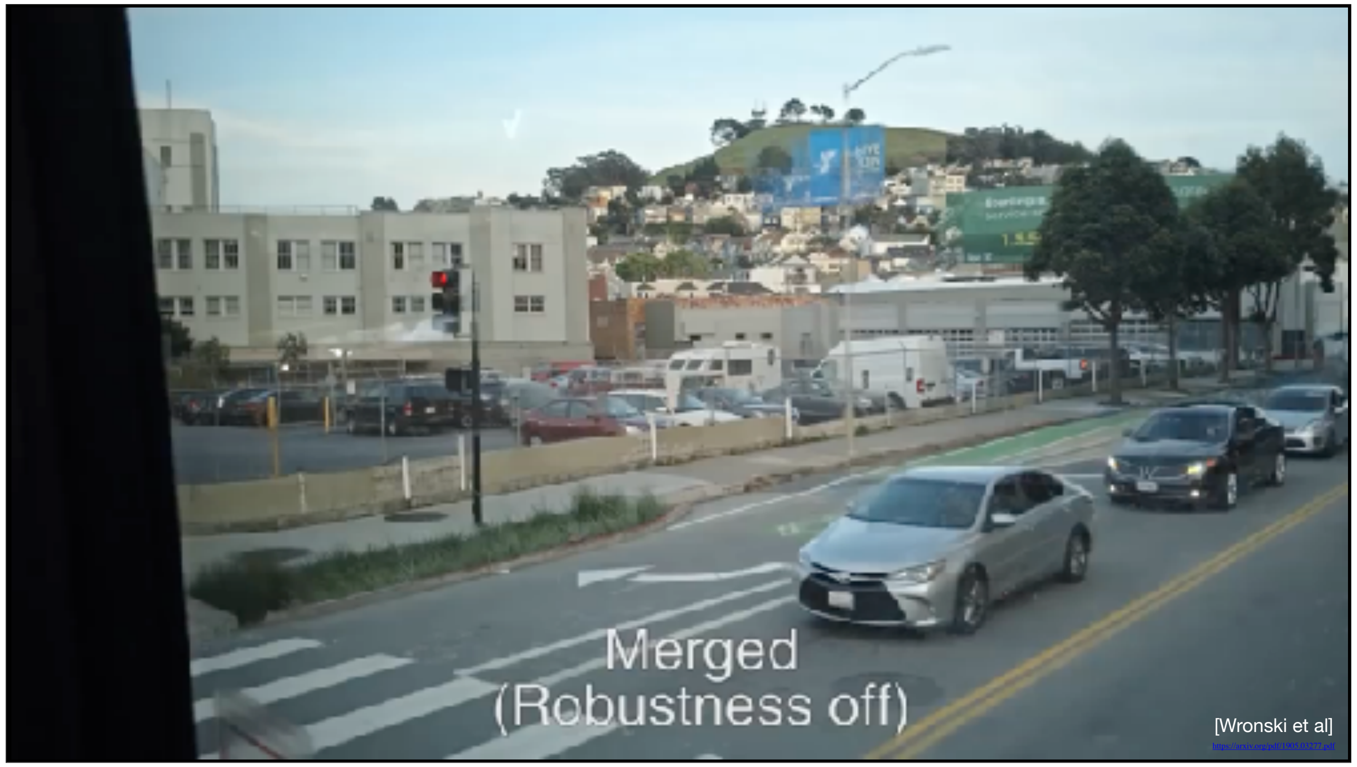
[Wronski et al]
https://arxiv.org/pdf/1905.03277.pdf

There are some things we do have to keep in mind, though.

For instance, certain types of motion can’t be aligned well, like large baseline camera translations, which introduce parallax.

[Wronski et al]
https://arxiv.org/pdf/1905.03277.pdf

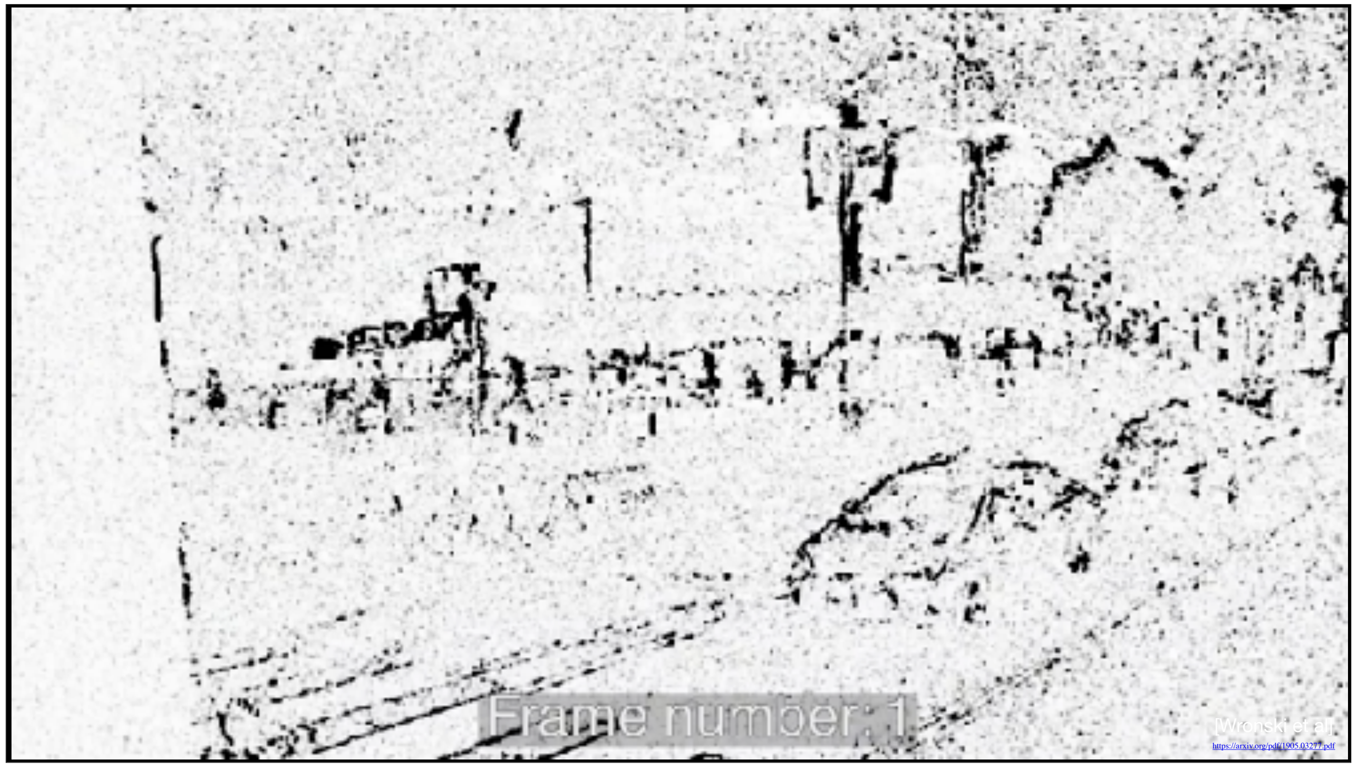
Since even when we align them, the images will seldom be identical because there are some regions which are visible in some frames, but not in others — places where we have occlusions



[Wronski et al]
https://arxiv.org/pdf/1905.03277.pdf

If we just naively merge those aligned images, we’re likely to get artifacts at a lot of these high-parallax regions. You can see there are ghosting artifacts in places where the alignment failed.

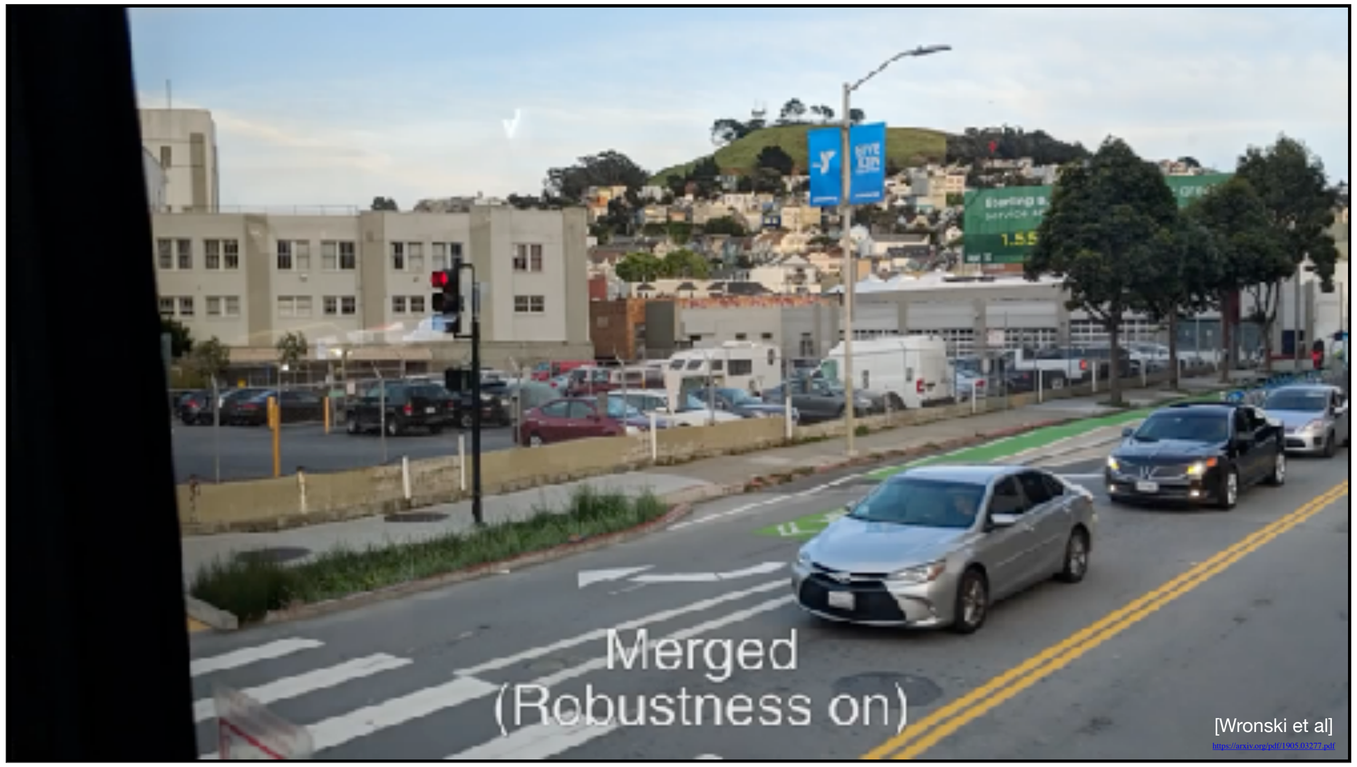
What we actually need to do is to weight different regions by their local statistics and consistency with one another. There are many ways of doing this, but one way is to just compare patches perceptually in the aligned stack of images. If the patch has a high temporal variance in color, we can apply a lower weight to the contribution from that particular frame.



[Wronski et al]
https://arxiv.org/pdf/1905.03277.pdf

Here’s what that kind of comparison could look like — we could compute a metric for each pixel or patch in each of the images based on how likely it was that the alignment was performed successfully — here white means “higher weight / better alignment” and black means “lower weight / worse alignment”.

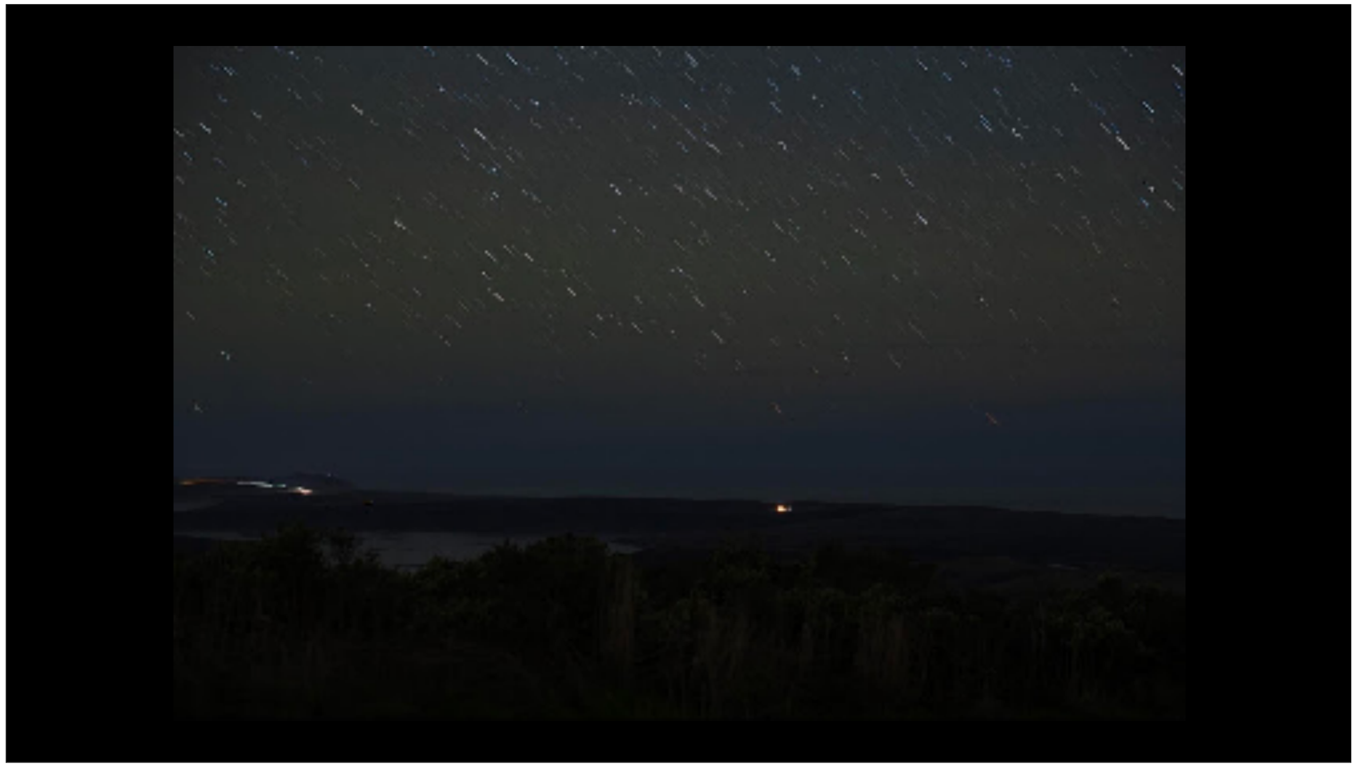
Then, we can do the averaging the same way we did before, but use these weights in averaging — and we get much better results for the final image.



[Wronski et al]
https://arxiv.org/pdf/1905.03277.pdf

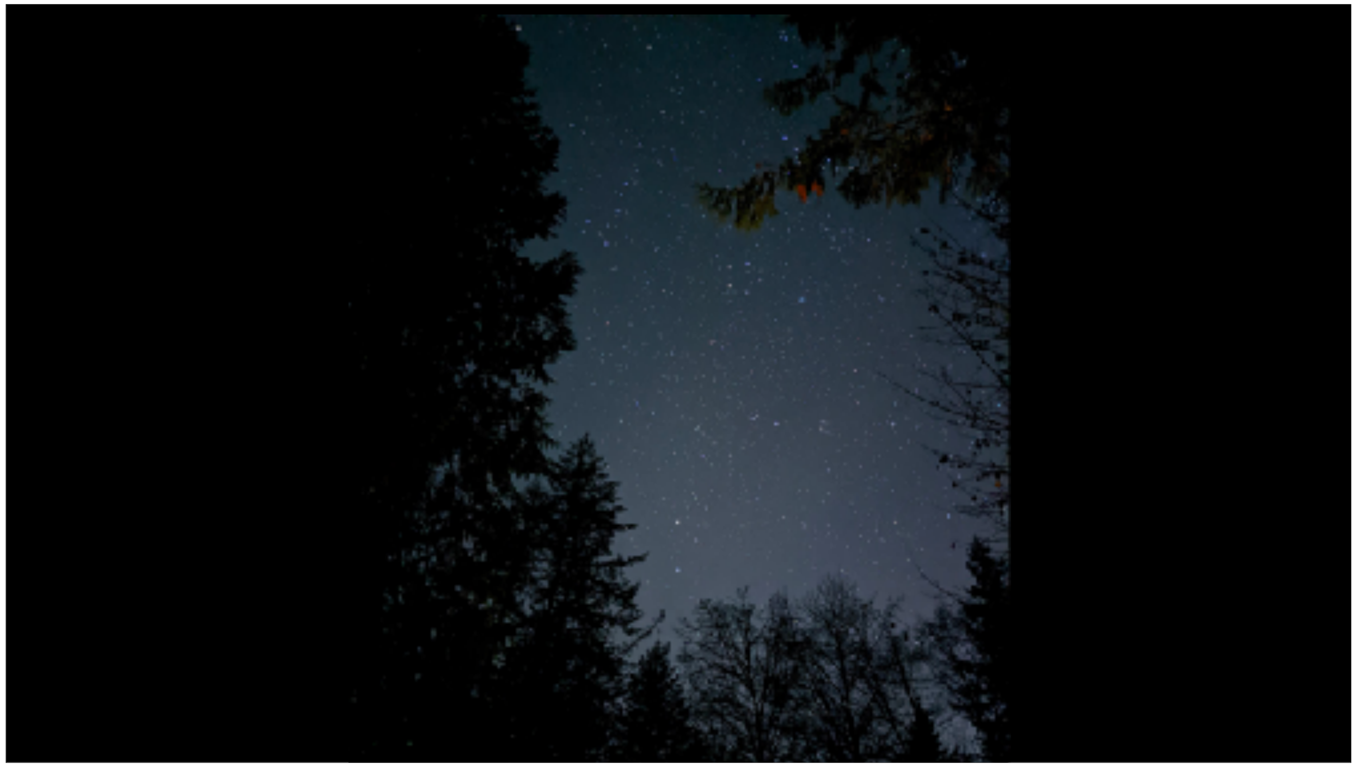
Here’s a comparison — with and without the weights in averaging.

The same method, if applied to shots taken over a course of minutes instead of seconds, can even take super impressive pictures of stars. Although in this case, since it's virtually impossible to hold a phone still enough for 5 minutes, it usually only works if you rest your phone against something.

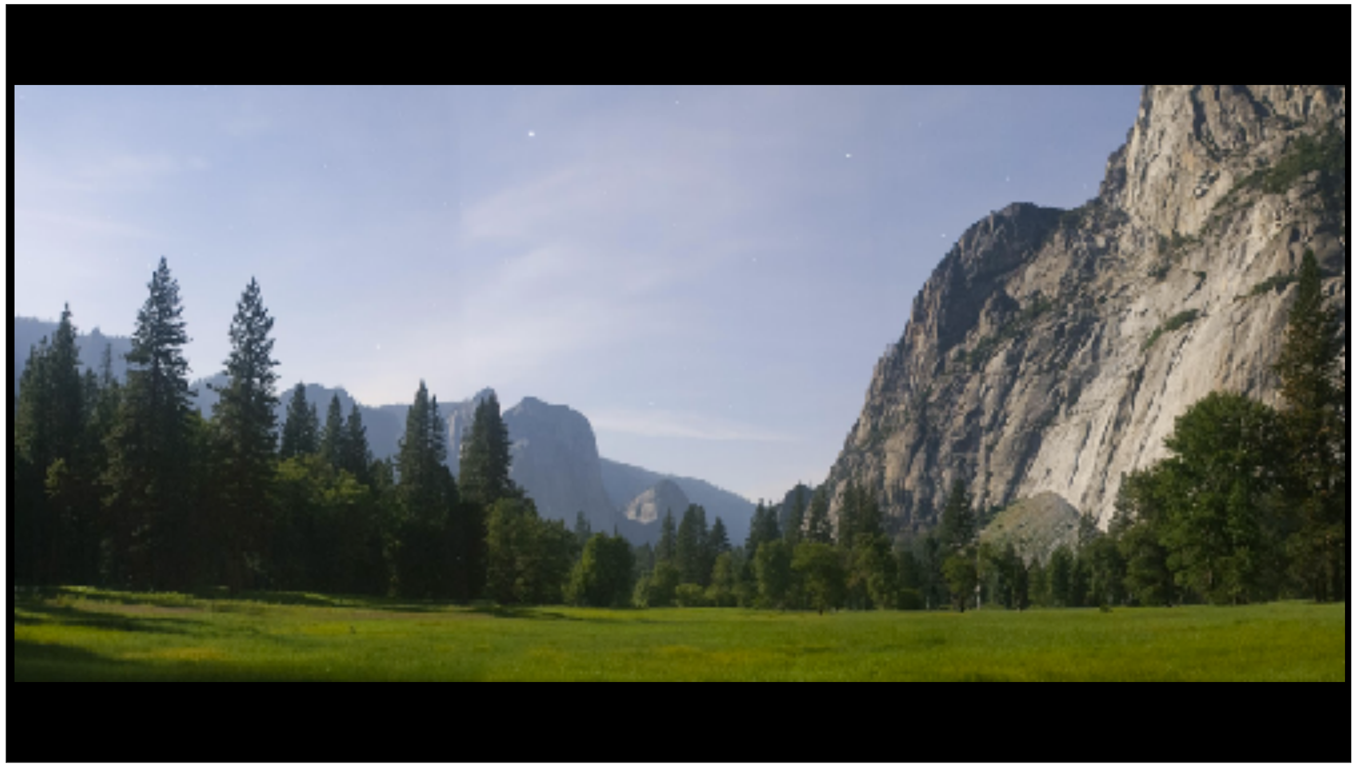


Here’s an example of what an image would look like if you did a single 2-minute long exposure.

Notice that each star shows up instead as a streak — this can be because of the natural motion of the earth, or a slow drift in the camera orientation.



And here’s the type of image you get when you use an aligned burst of images.

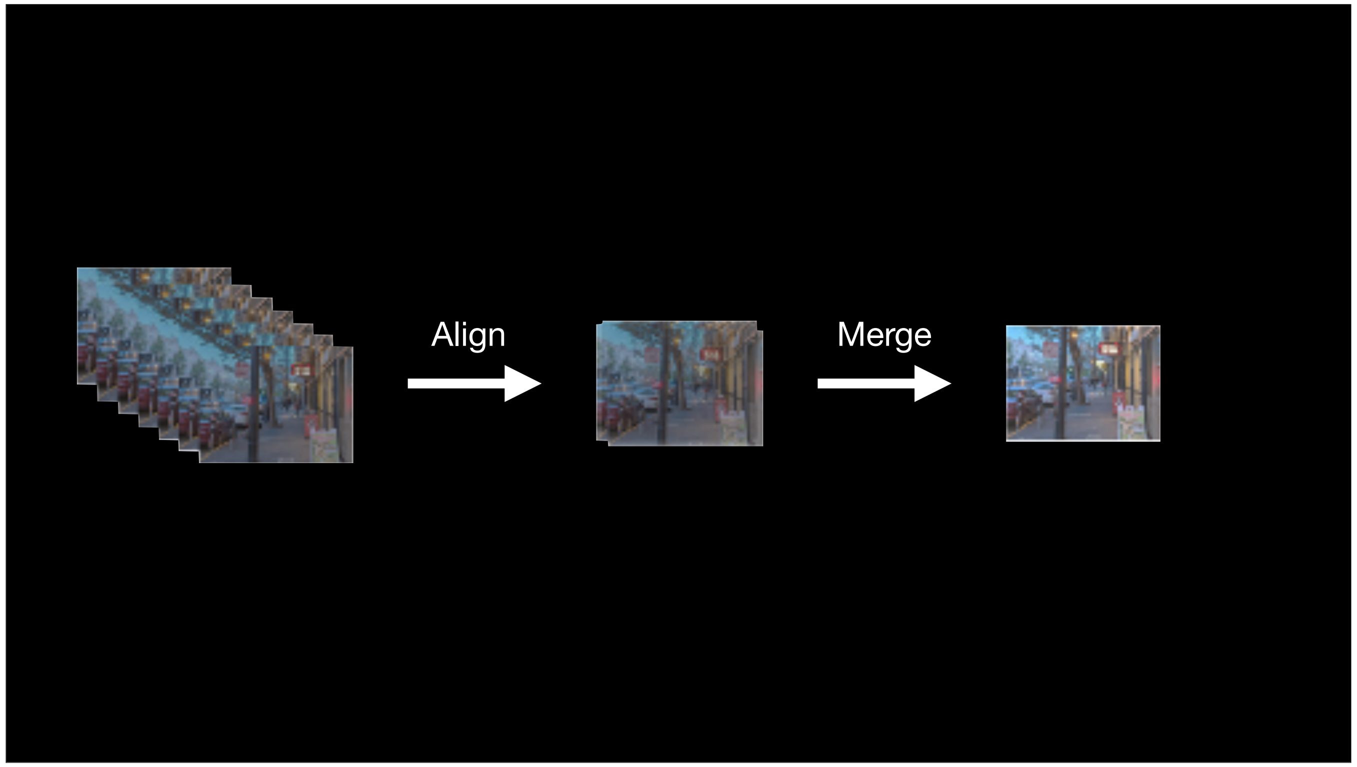


A cool fun-fact:

If you just take a long enough exposure photograph of a scene at night, you actually end up with an image which looks a lot like daytime.

This image is a 3-minute exposure taken in the middle of the night, and you can see the stars in the sky, but the grass is green and the sky is blue.

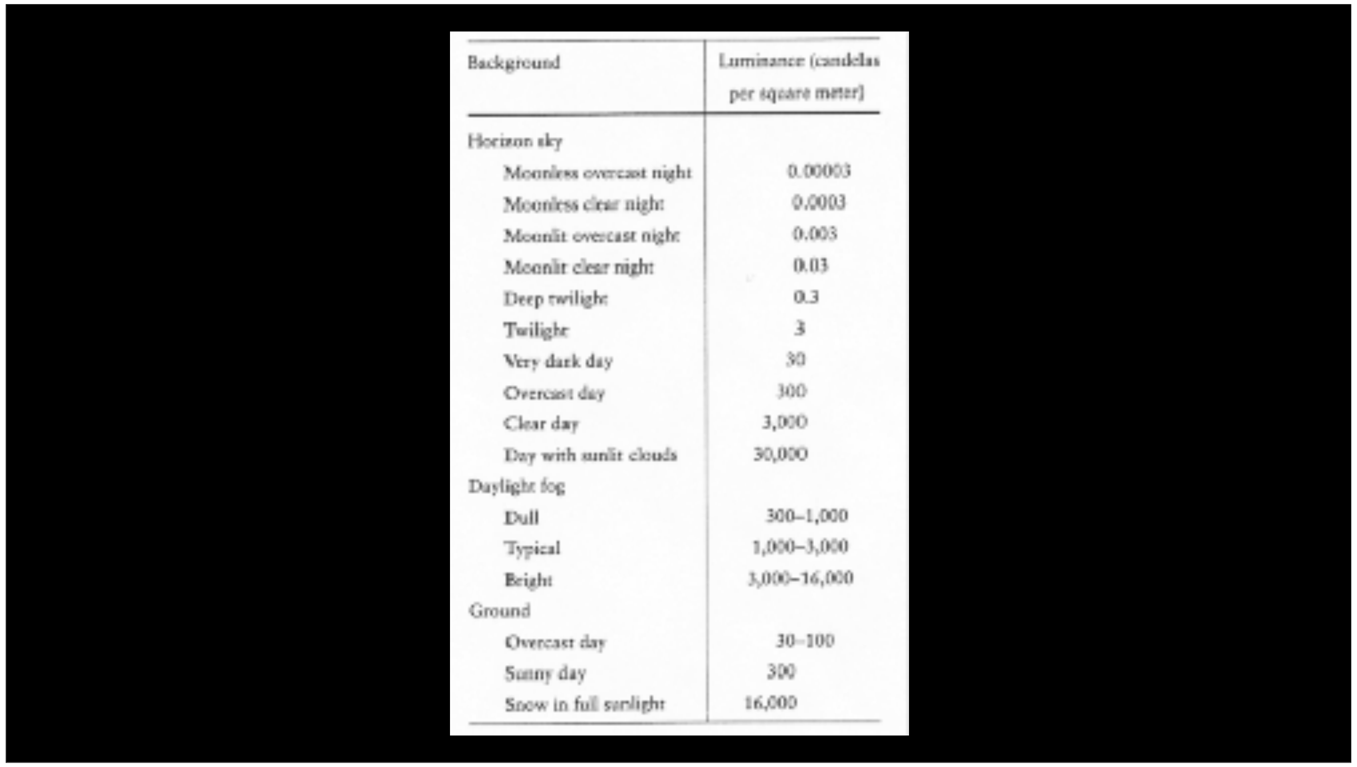
So, to make the images more “night-like” a lot of these night photography algorithms also apply a color curve…



Most camera phones have a lot of these features built in.

In most cases, when you open your camera app, your phone is actually constantly taking bursts of photographs, so that when you actually do press the shutter button, there’s no delay to capture a burst of images — you just use the last 5-10 frames that were captured.

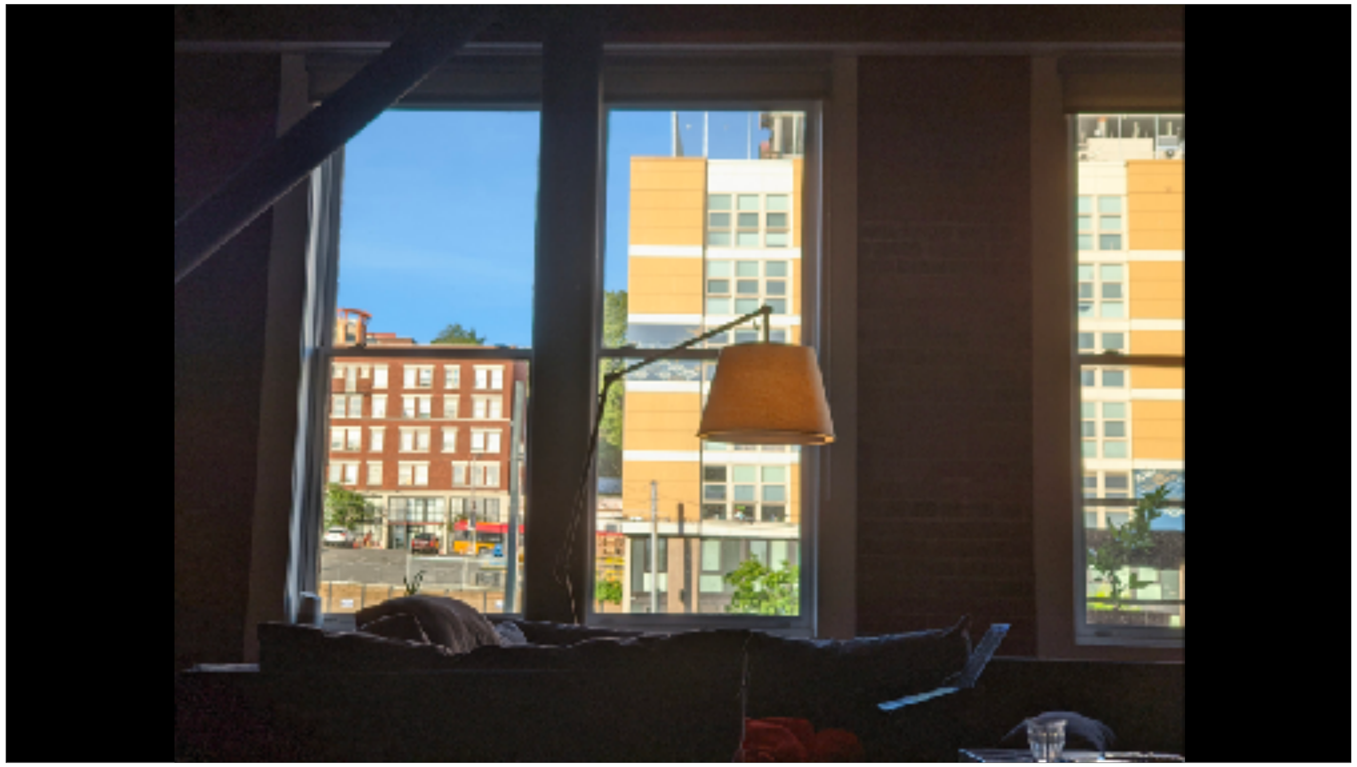
To recap, the basic idea we’ve seen here is to capture a burst of frames, align them, and combine information from them to produce a better image.
This same technique can be applied to a bunch of different applications.



Background	Luminance (candelas per square meter)
Horizon sky	
Moonless overcast night	0.00003
Moonless clear night	0.0003
Moonlit overcast night	0.003
Moonlit clear night	0.03
Deep twilight	0.3
Twilight	3
Very dark day	30
Overcast day	300
Clear day	3,000
Day with sunlit clouds	30,000
Daylight fog	
Dull	300–1,000
Typical	1,000–3,000
Bright	3,000–16,000
Ground	
Overcast day	30–100
Sunny day	300
Snow in full sunlight	16,000

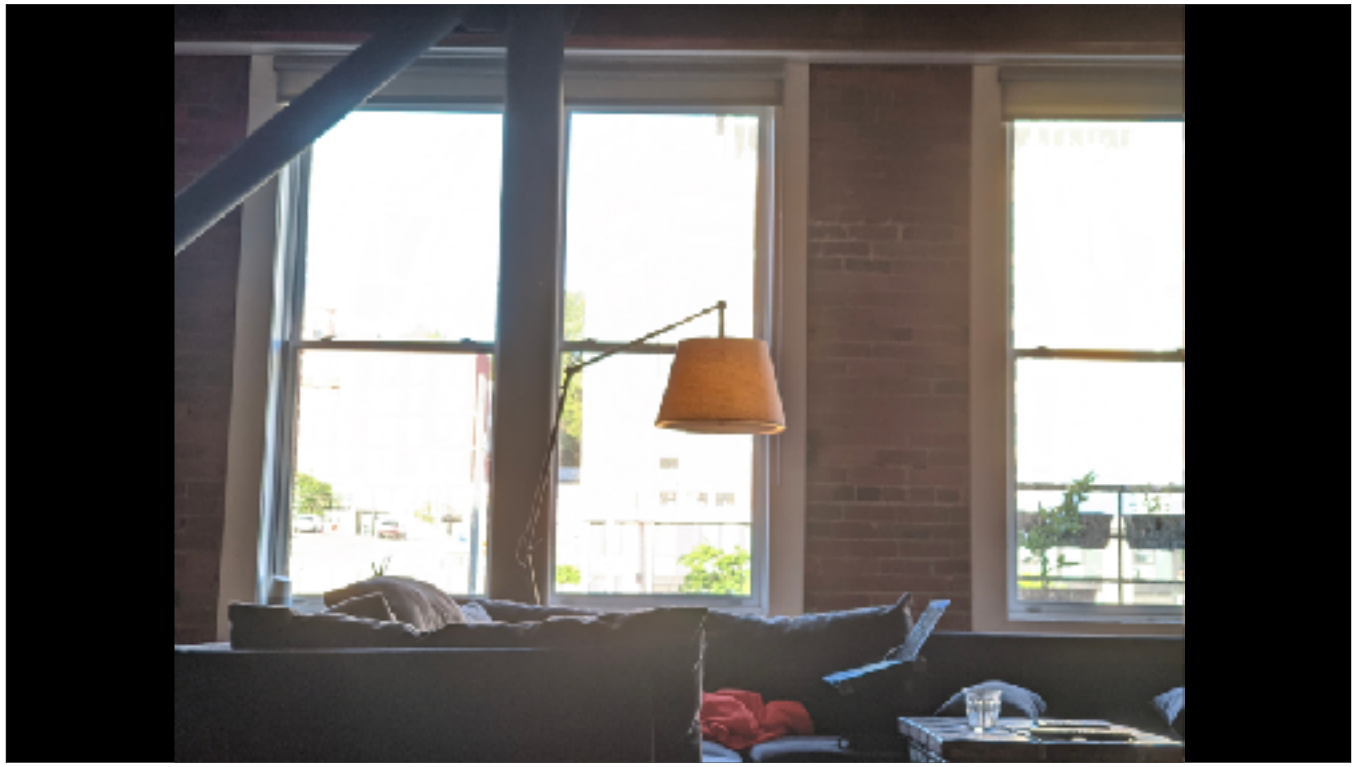
For example, dynamic range.

It’s impressive that our eyes are capable of seeing things which are both super bright and super dark. Our eyes have a high dynamic range, because we’re able to distinguish a pretty large range of brightness values at the same time, like if we’re sitting in a dark apartment…

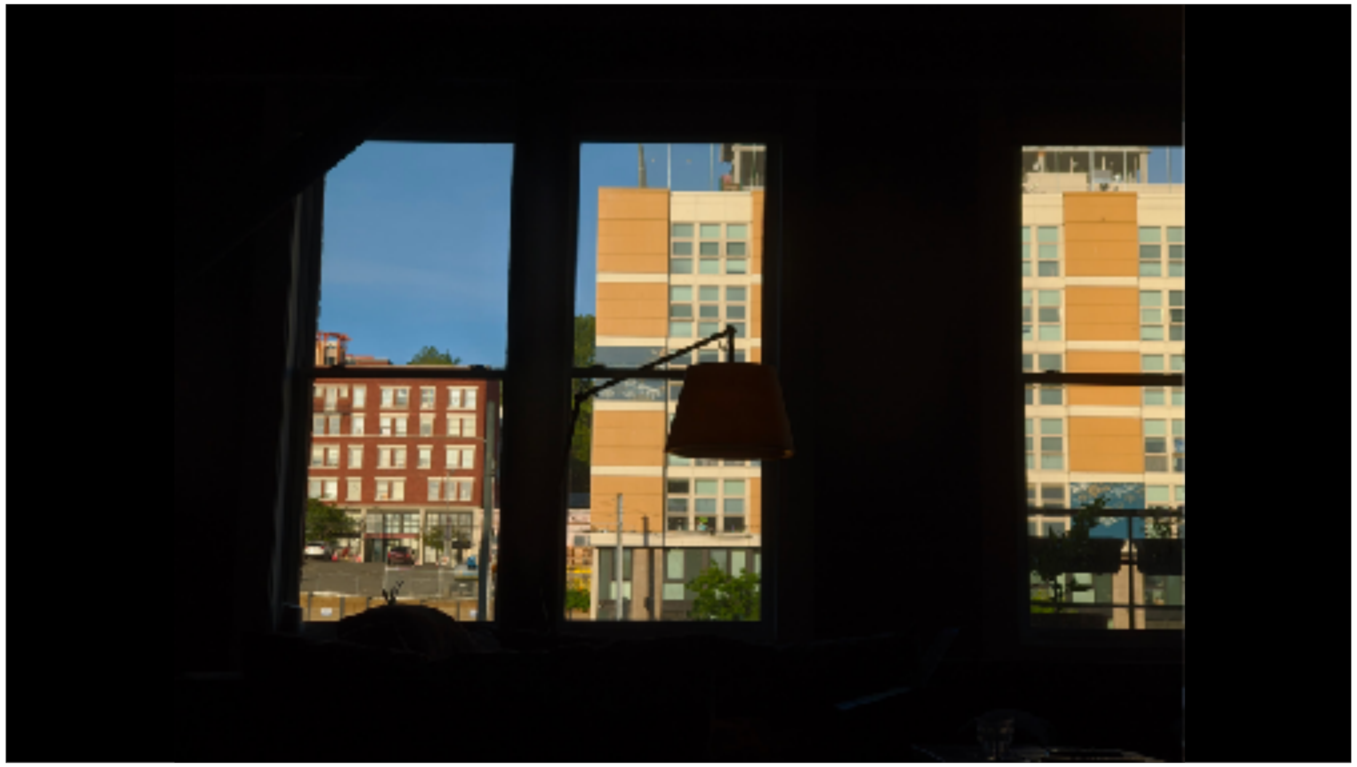


We’re able to see things both inside and outside the window. This image here might be something similar to what we’d see.

But a camera can’t do the same. Most cameras have really limited dynamic range, so you’ll either get…



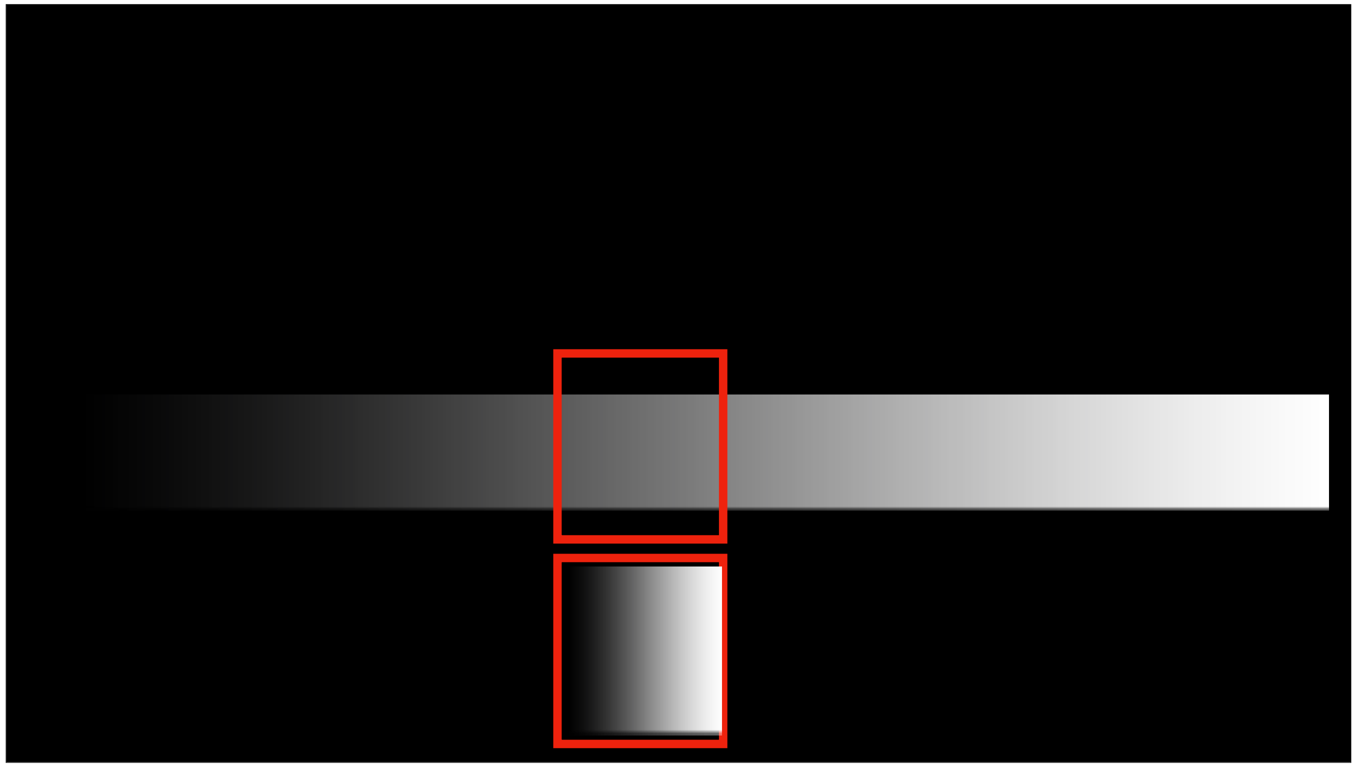
An image which is bright enough inside and totally overblown outside, or…



…the opposite, where we clearly see the outside, but everything inside is totally black.

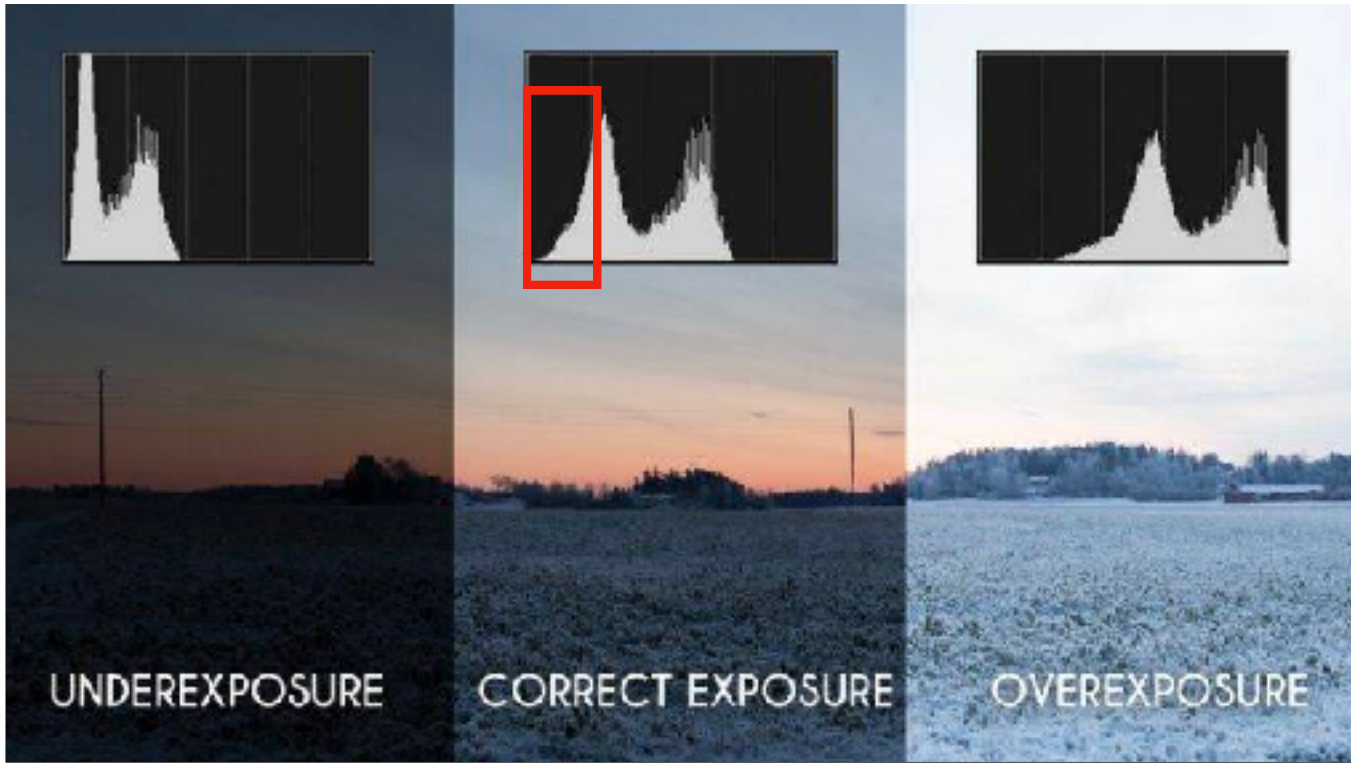
And this concept in photography is commonly referred to as overexposure and underexposure.

If we expose for not enough time, dark things will show up as black.
If we expose for too long, things which are very bright will show up washed out (white).



If this gradient contains all the different levels of luminance (or brightness) in the scene,
[slide]
a camera is only going to capture a small window of these, and the brightness values in this window will always be scaled to [0,255].

Things outside the window on the left and right will be clamped to 0 and 255, respectively.

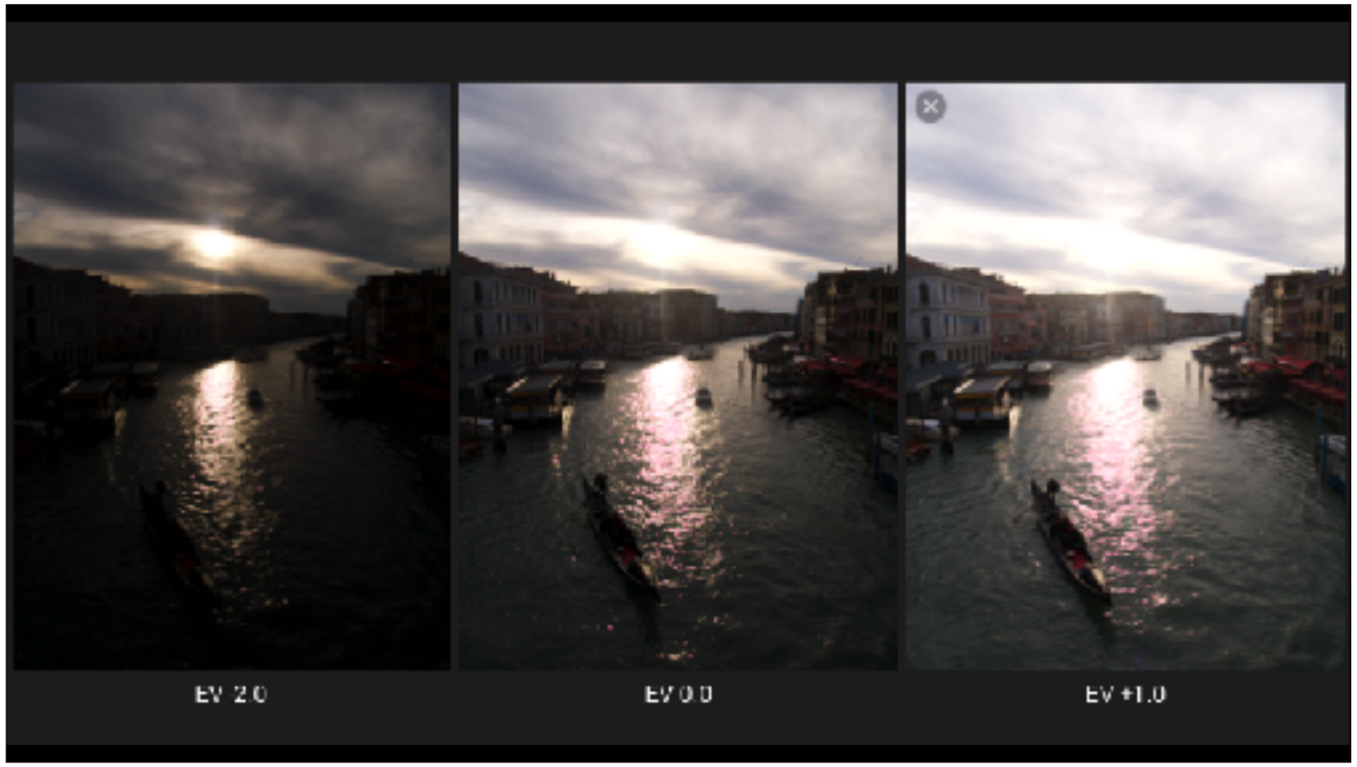


Here’s another figure which illustrates the same thing.

At the top you see what are referred to in image editing programs as color histograms or color curves, and they basically tell you how many pixels in the image have this particular value, where the X axis varies the brightness.

You can see in the middle image, when everything is correctly exposed, we have a pretty smooth distribution of intensity values, but in the left image, there’s an abnormally large spike at 0, which tells us that a lot of the pixels which were in [slide] this region of the curve have now been flattened or clamped to the same value.

What if we want our cameras to capture a larger range of values?
What if we want our images to have a higher dynamic range than our camera supports?

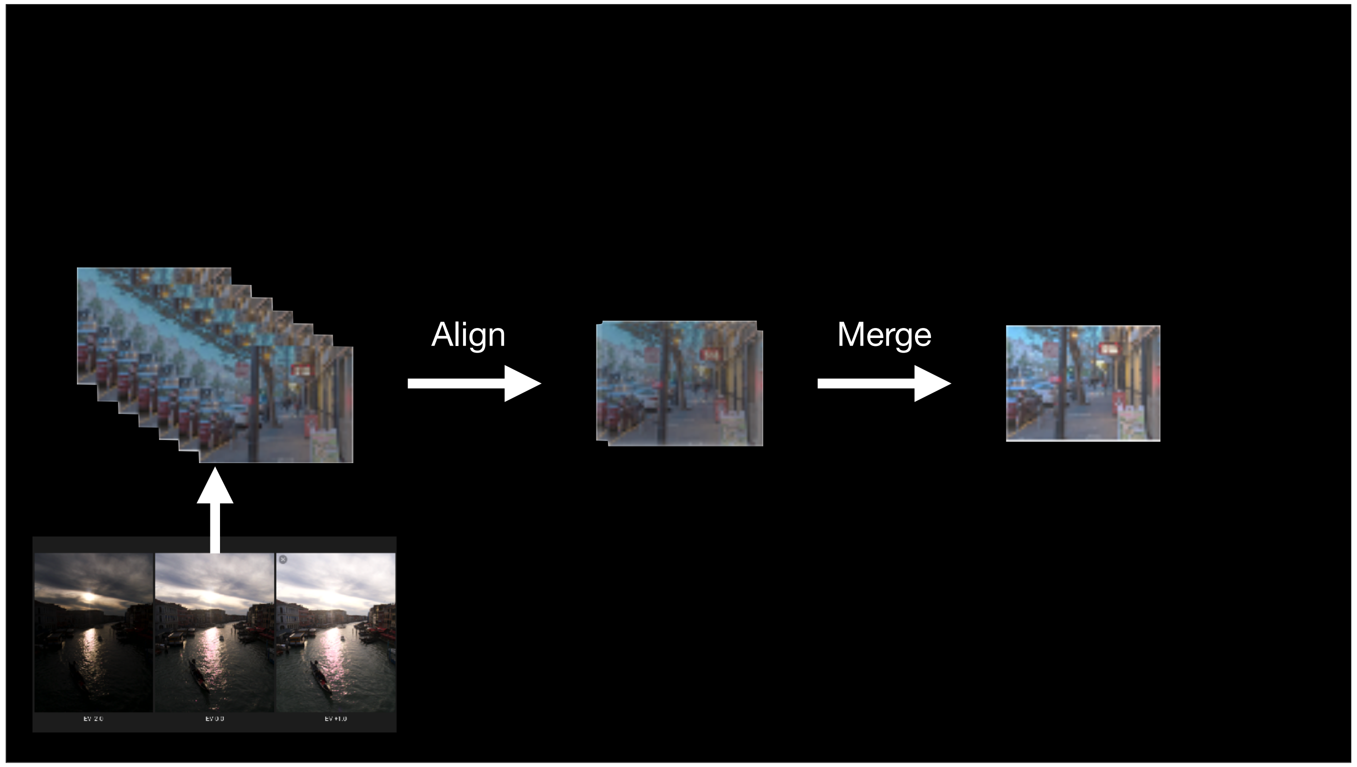


Traditionally, this was done through what’s known as exposure bracketing, where the camera would be put on a tripod and the same image would be taken at multiple exposures.

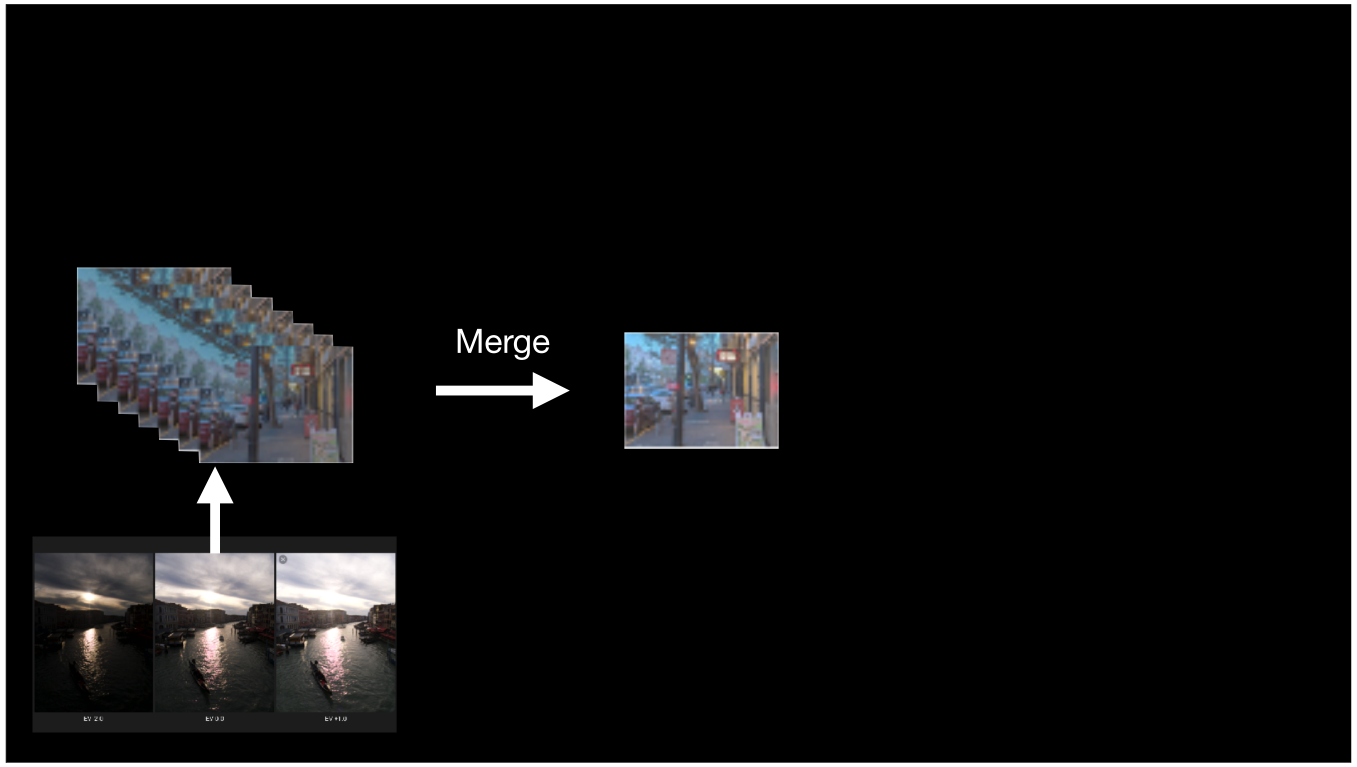
Then, in post, someone could go in and combine those images by cutting and pasting parts of the image which had the level of exposure that seemed appropriate.

Maybe the sky from the left photo, the water from the middle photo, and the buildings from the right photo.

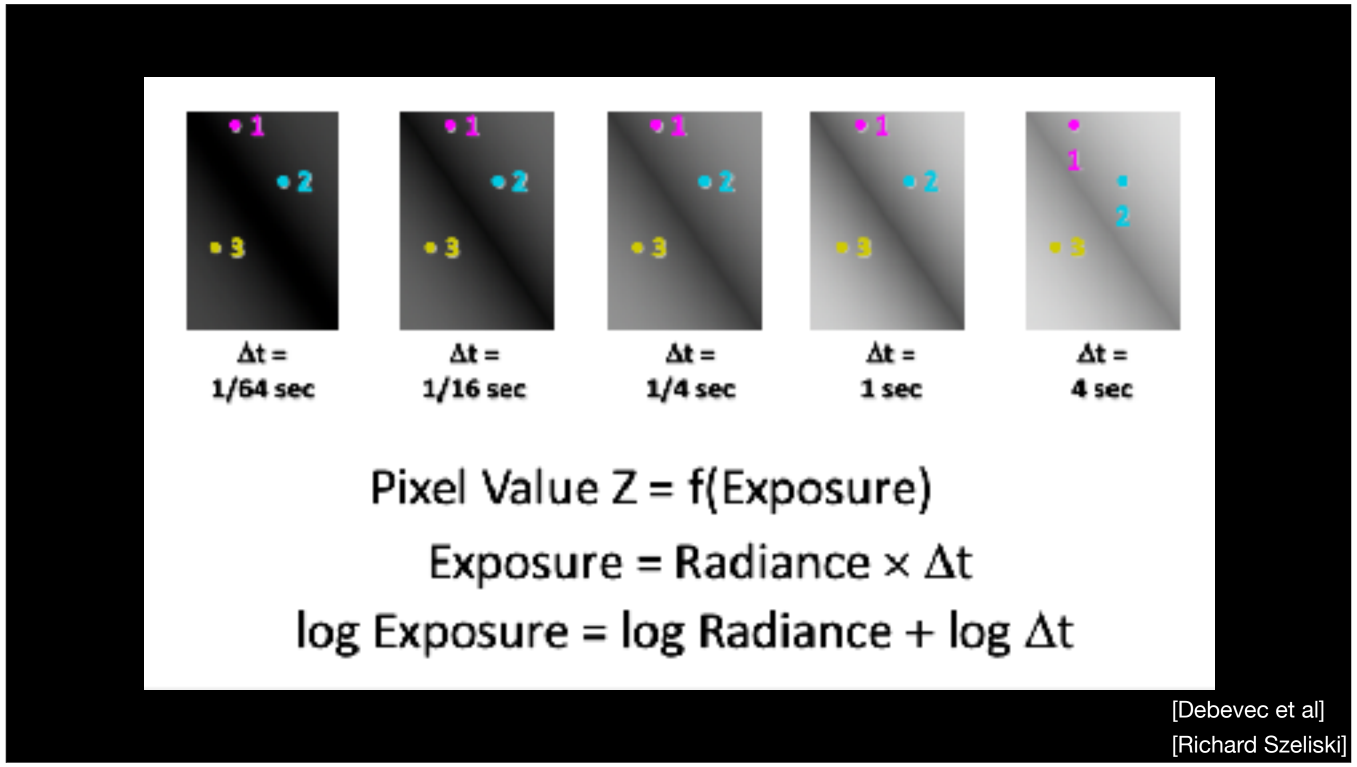
But again, with a smartphone, we don’t want to be carrying around a tripod, right?
So we can again use the same sort of pipeline we saw before…



The natural adaptation of this is to capture a set of exposure bracketed images in your burst, align them, just like we did before, and then use some smart LDR=>HDR merging algorithm to combine them into an HDR image.



For simplicity, let’s first look at the version of this problem without the alignment —



Pixel Value Z = f(Exposure)

Exposure = Radiance × Δt

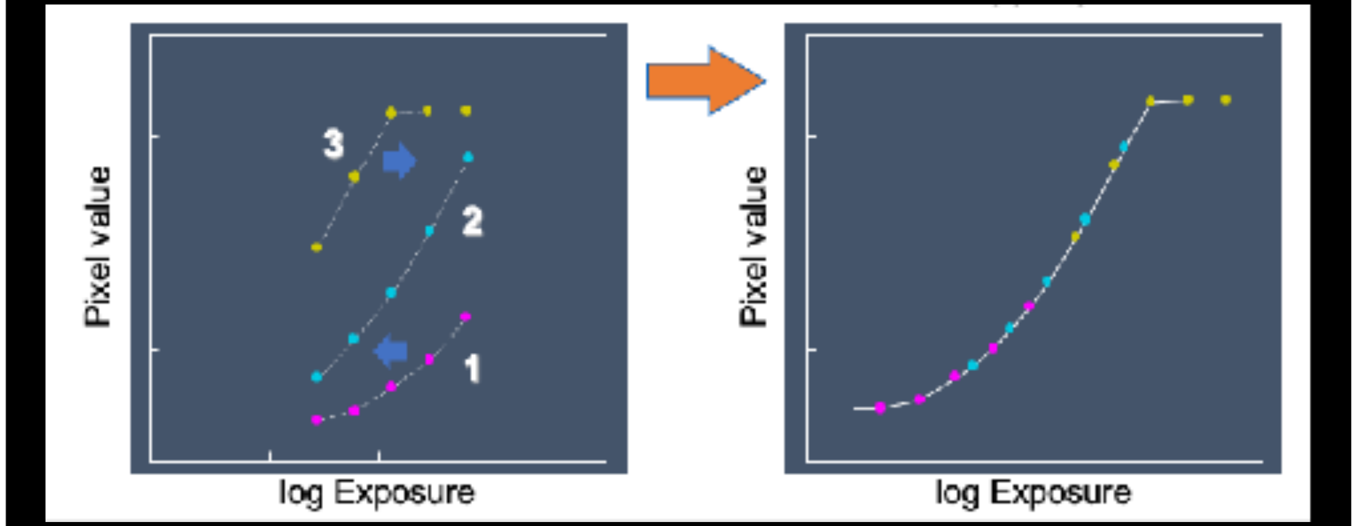
log Exposure = log Radiance + log Δt

[Debevec et al]
[Richard Szeliski]

Assuming we have a bunch of images which are aligned, or we have a camera on a tripod,

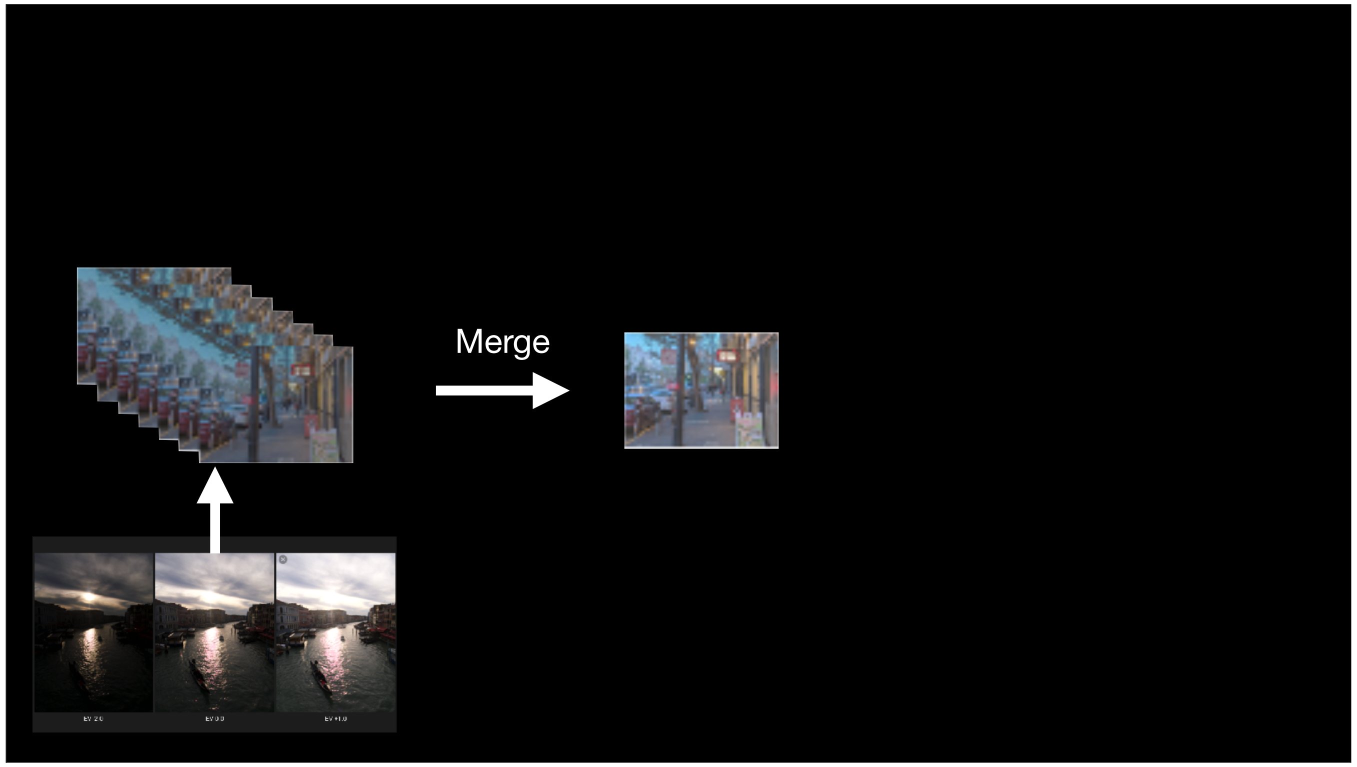
The algorithm works by looking at points which it knows has a constant radiance, or emitted light, at each point, and measures the pixel values which are captured across several different exposures.

And in doing so, since the exposure time is known, and the pixel values are known, we can solve for the unknown radiance for each point.

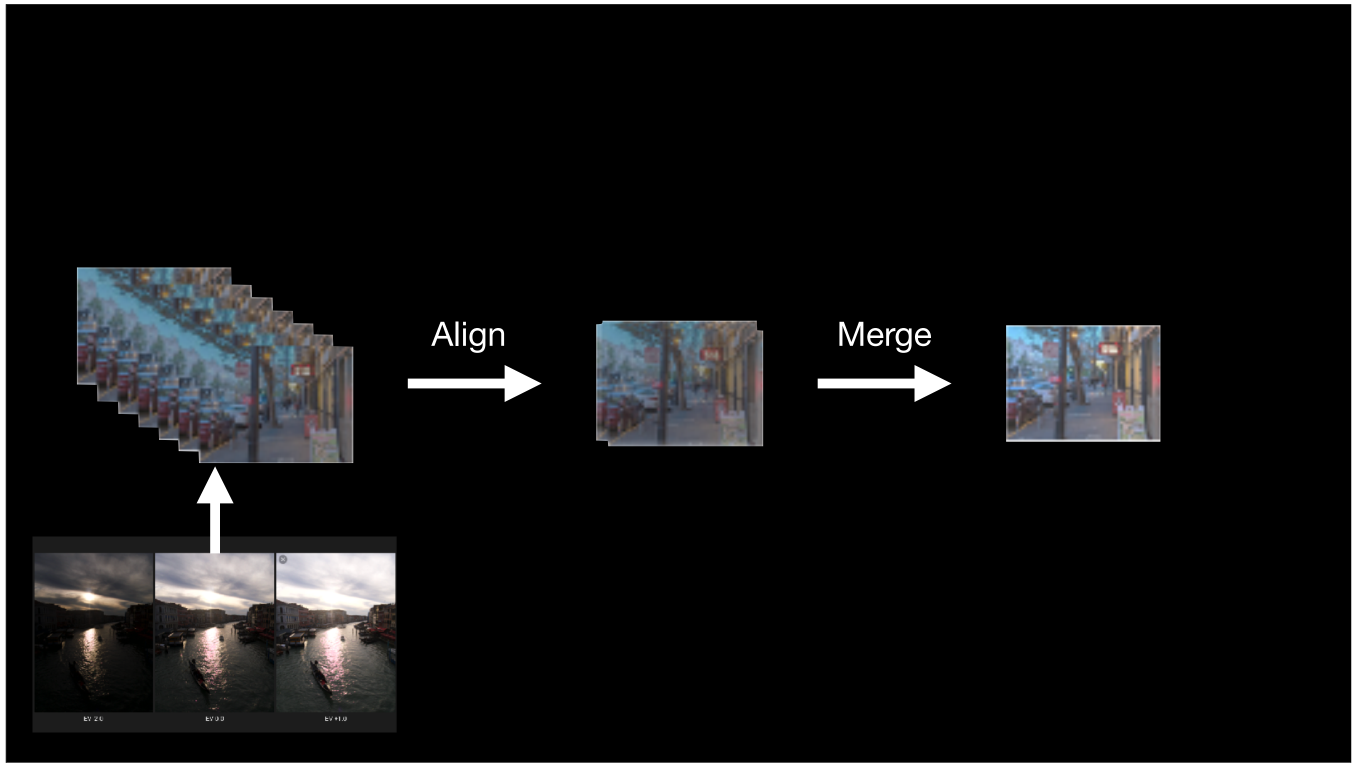


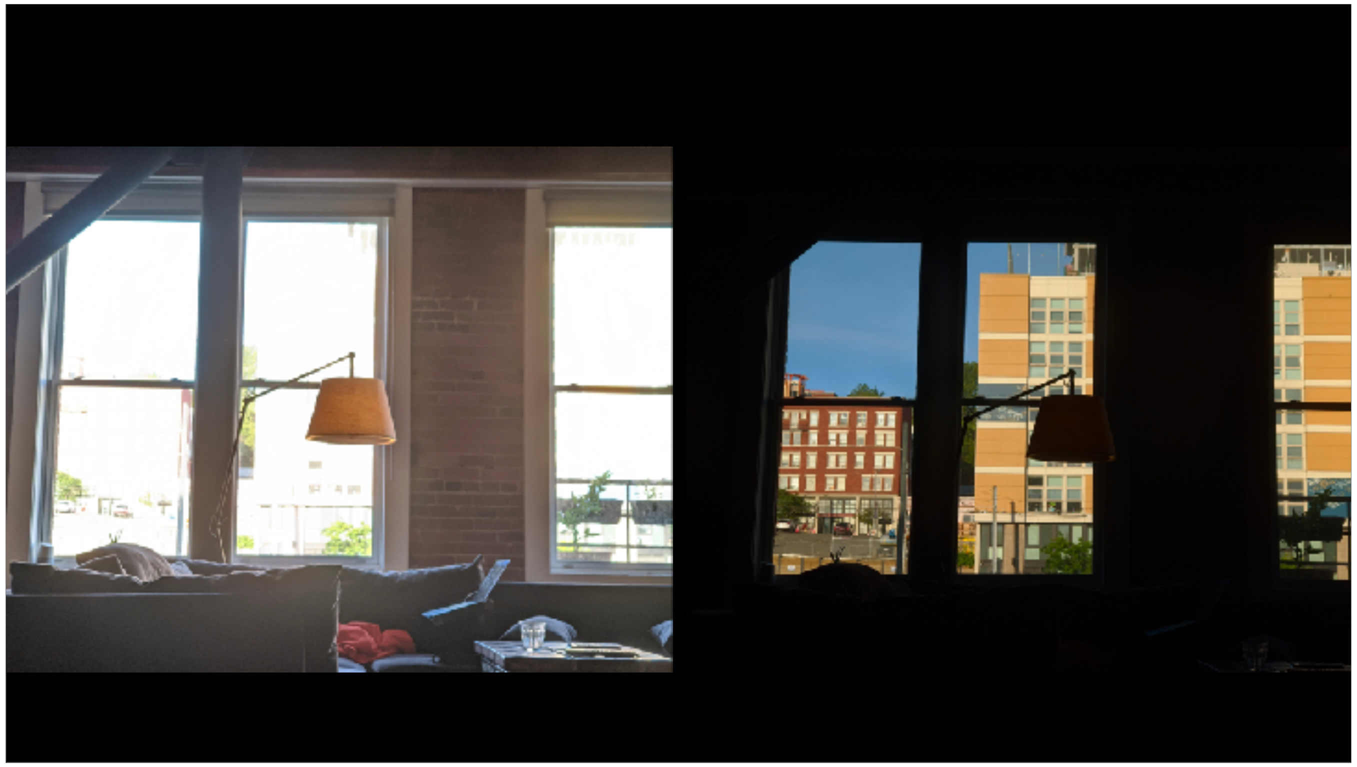
[Debevec et al]
[Richard Szeliski]

In turn, this tells us the true ratios in brightness between a pair of points, whereas a single exposure might not be as accurate, since in a single image, certain points may be too bright and will have clipped to the maximum value that the sensor could capture.


Merge

This is a great solution, but in practice, it’s not what most *phones* use nowadays, and the reason for that is basically because of alignment.

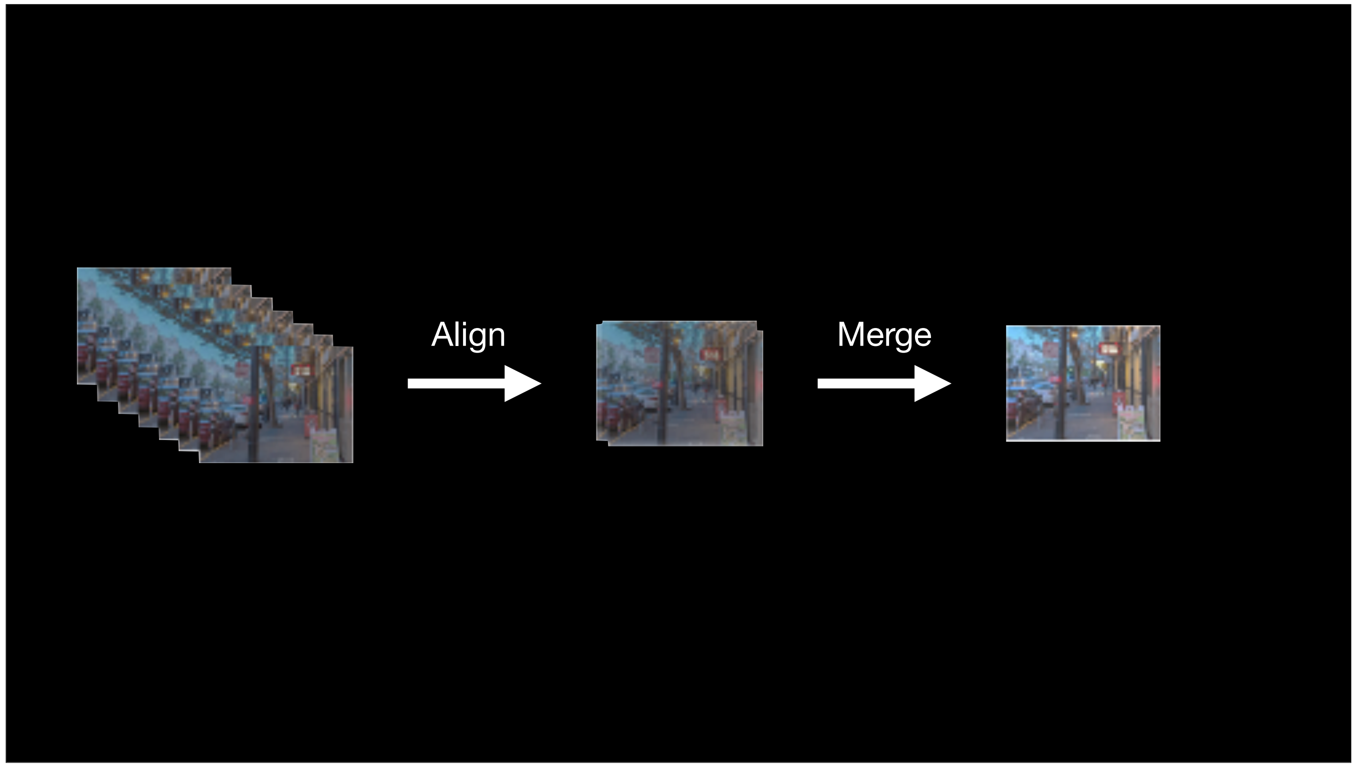

Align
Merge
EV -2.0
EV 0.0
EV +1.0



Say we have a set of bracketed exposures, containing this overexposed frame and this underexposed frame, and now we want to align them.
How do we do that?

Both images are seeing almost complementary parts of the scene, and there's pretty much no preserved detail in any of the overexposed regions for us to use in feature detection, matching, or optical flow.

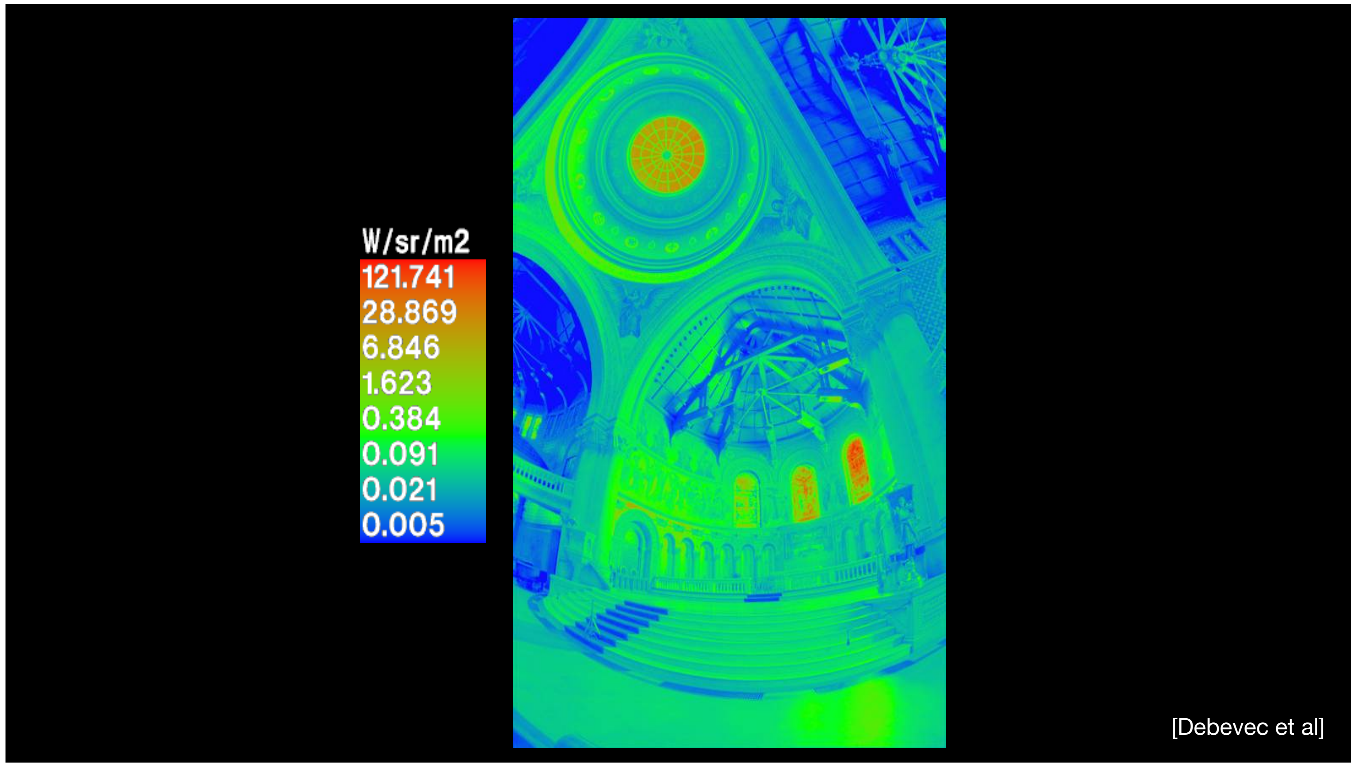
It effectively puts us in the same situation we saw in the stereo lecture, where we're trying to match textureless surfaces — a very difficult task.
So this approach is unfortunately not very robust to extreme lighting, and can lead to a lot of artifacts, like double images, or ghosting, as a result of unreliable alignment.



In practice, what most phones do is to actually just capture a bunch of short, identical exposures, just like with the de-noising pipeline.

The idea here is basically that by capturing everything with a low enough exposure, very few things will be totally overexposed, and things which are underexposed can be resolved by adding more images in the burst.

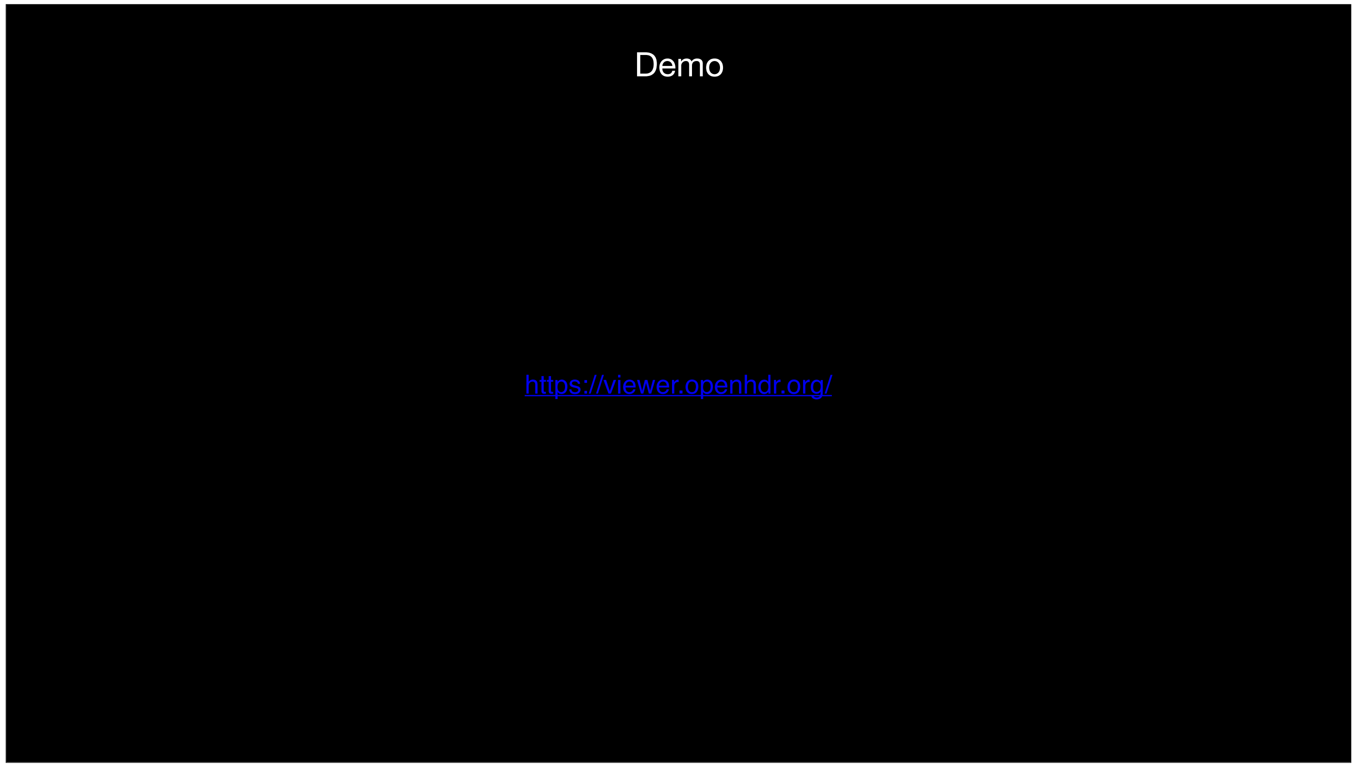
But there’s still one last step to this process…



[Debevec et al]

We talked about how to extract these radiance values for each pixel, but not about how we'd end up displaying this high-dynamic range image.

One obvious way of going about this is to just store these radiance values in a floating point binary, and when we want to look at the image, have the user pick an exposure, and we can synthetically generate that exposure and render what the image would have looked like with that camera exposure.



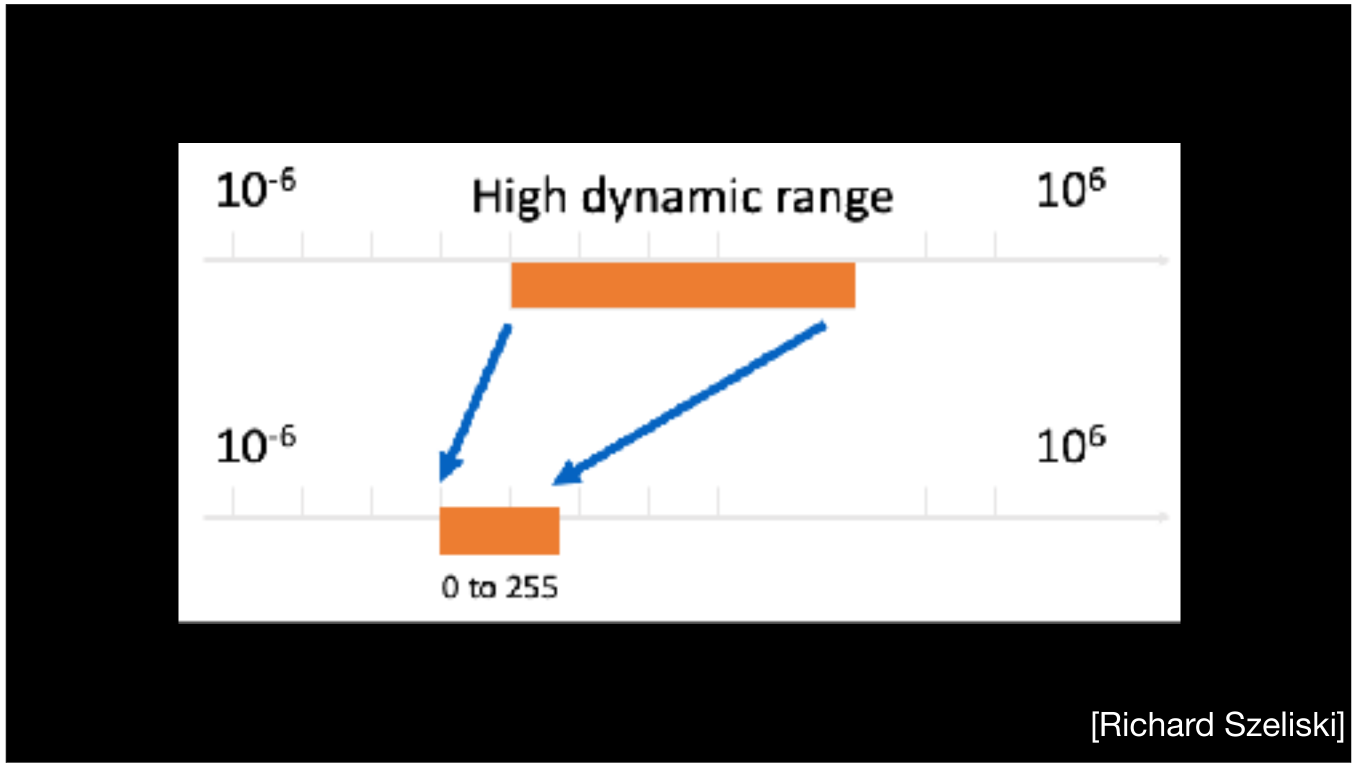
Demo

https://viewer.openhdr.org/

Demo.

This is different from just scaling the image, because as we move around to new exposures, we're discovering more detail in the image.

Notice the high exposure and low exposure, and how they're sometimes complementary.

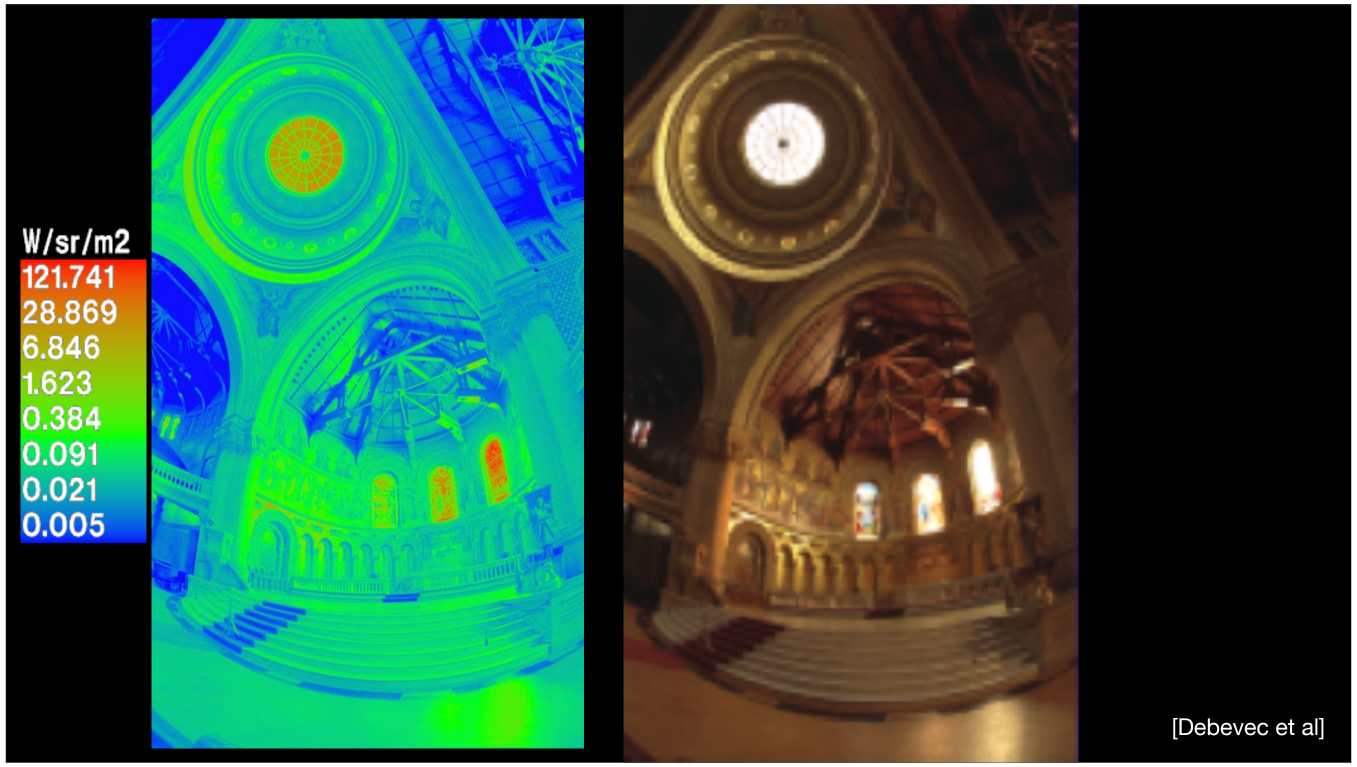


[Richard Szeliski]

That's cool, but it doesn't really accomplish the goal that motivated this whole discussion, which is that we want to create an image which has a higher dynamic range than our camera supports.

In order to do that, we have to do what's called 'tone mapping', which is the problem of taking these radiance values which have an arbitrary range, and somehow mapping them to pixel values within 0-255 so that we can either print them as images or display them on our screens. Basically, we need a function which transforms from radiance values to pixel values.

There are a bunch of different approaches we can take —

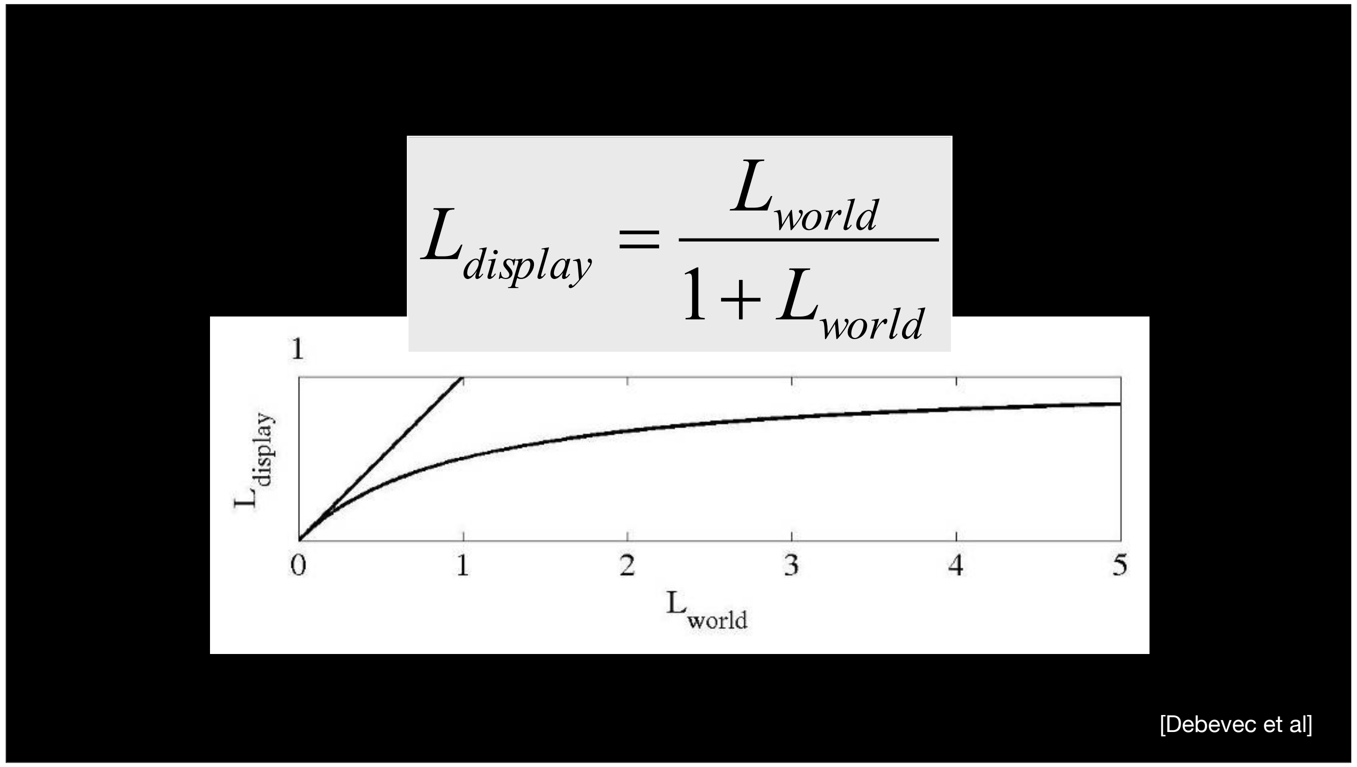


[Debevec et al]

Let’s consider the simplest version, which is just linearly scaling the radiance image so that the smallest value is 0 and the largest value is 255.

Perceptually, what that can result in, is our image only showing the very brightest parts. That’s because sometimes the brightest parts of the image are orders of magnitude brighter than the dark parts. [slide]

For reference, this is what one of the bracketed exposures looks like. [slide]
You can see there’s a lot of lost detail.



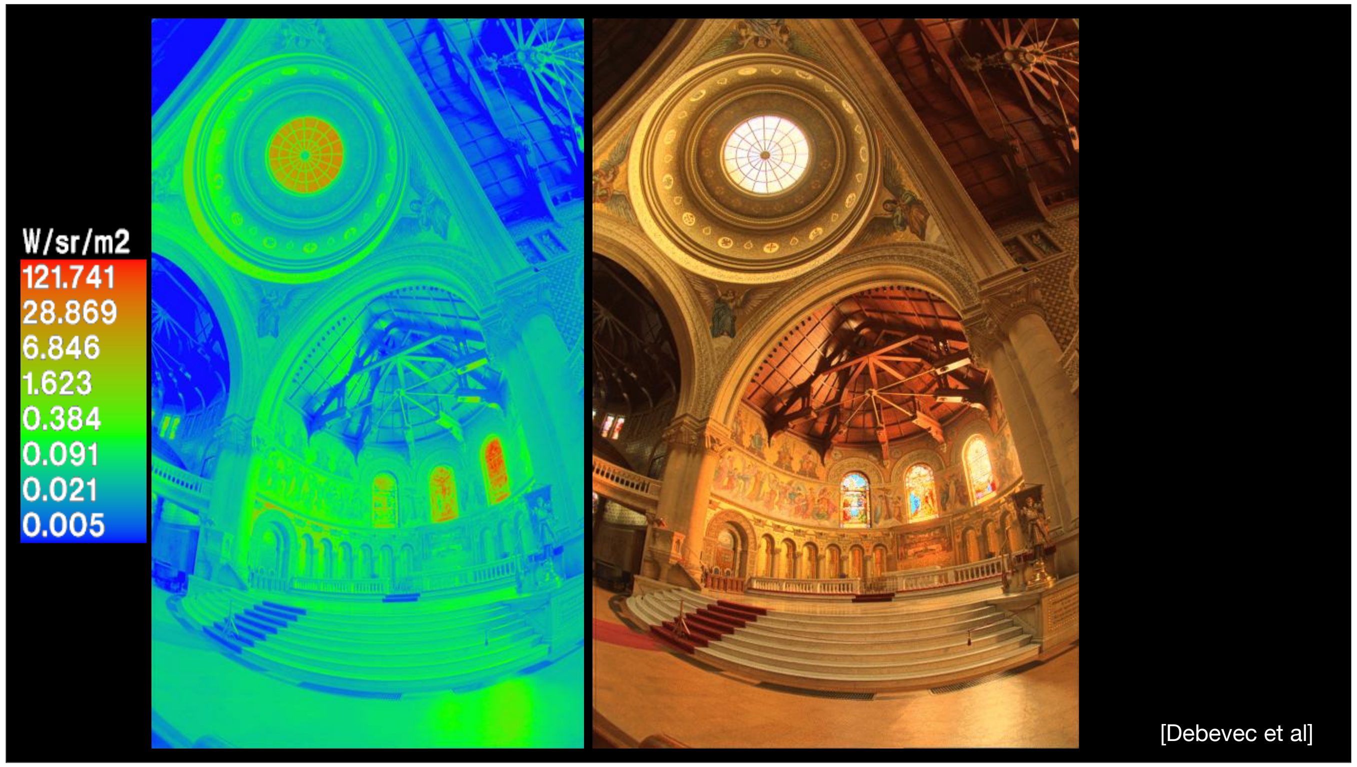
$$L_{display} = \frac{L_{world}}{1 + L_{world}}$$

[Debevec et al]

Another approach is to use what is often referred to as the Reinhart operator, which scales the radiance values so that brighter regions are more compressed than dark regions.

So in this formula, L_world is the luminance, or radiance, that we’ve estimated for a pixel, and L_display is the final pixel value that it’ll be mapped to (in this case, between 0 and 1 instead of 0 and 255).

You can see in this curve, values near zero in the input range get a lot of space in the output range (on the Y axis), but as the radiance increases, the output range gets more and more compressed.

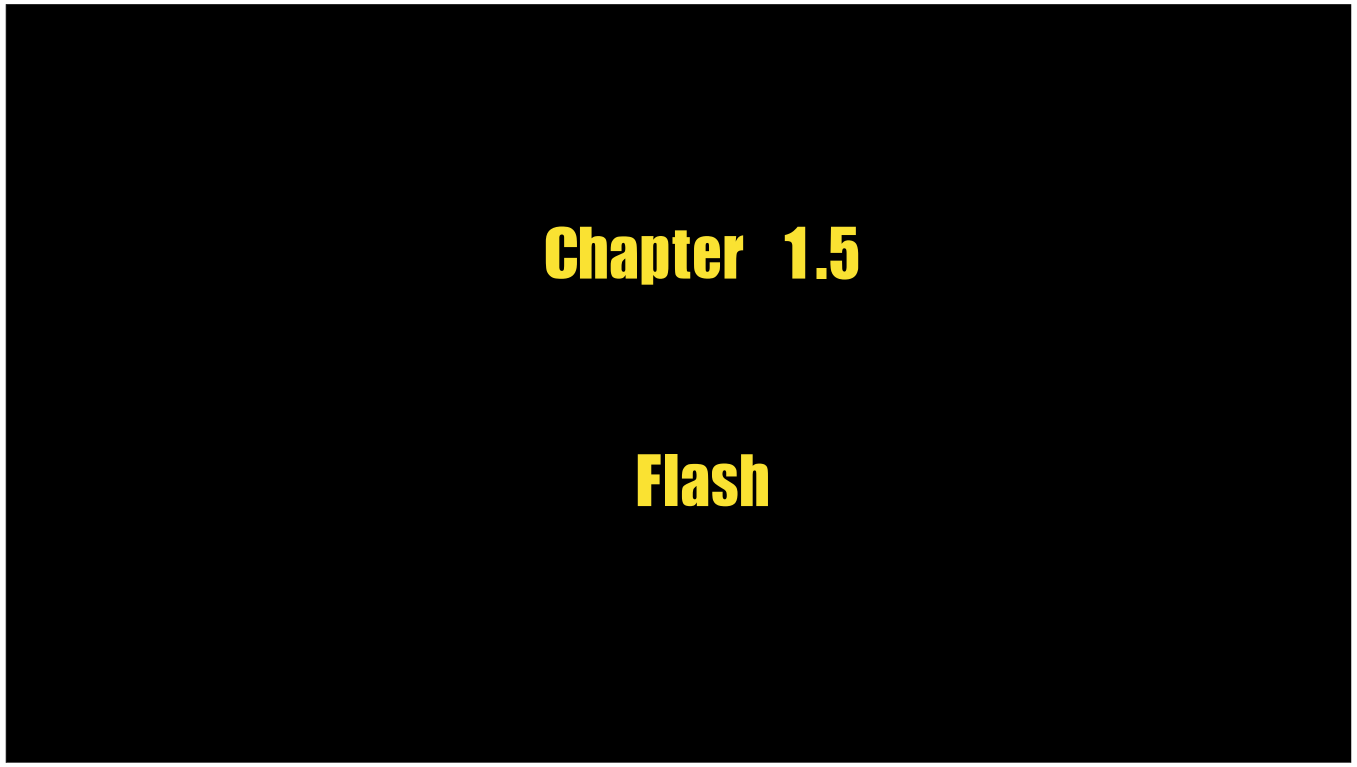


[Debevec et al]

And this results in a nicer looking image, like this one here.

And if you’re interested in the topic of more advanced tone mapping, I’ve left some more references at the end of the talk.

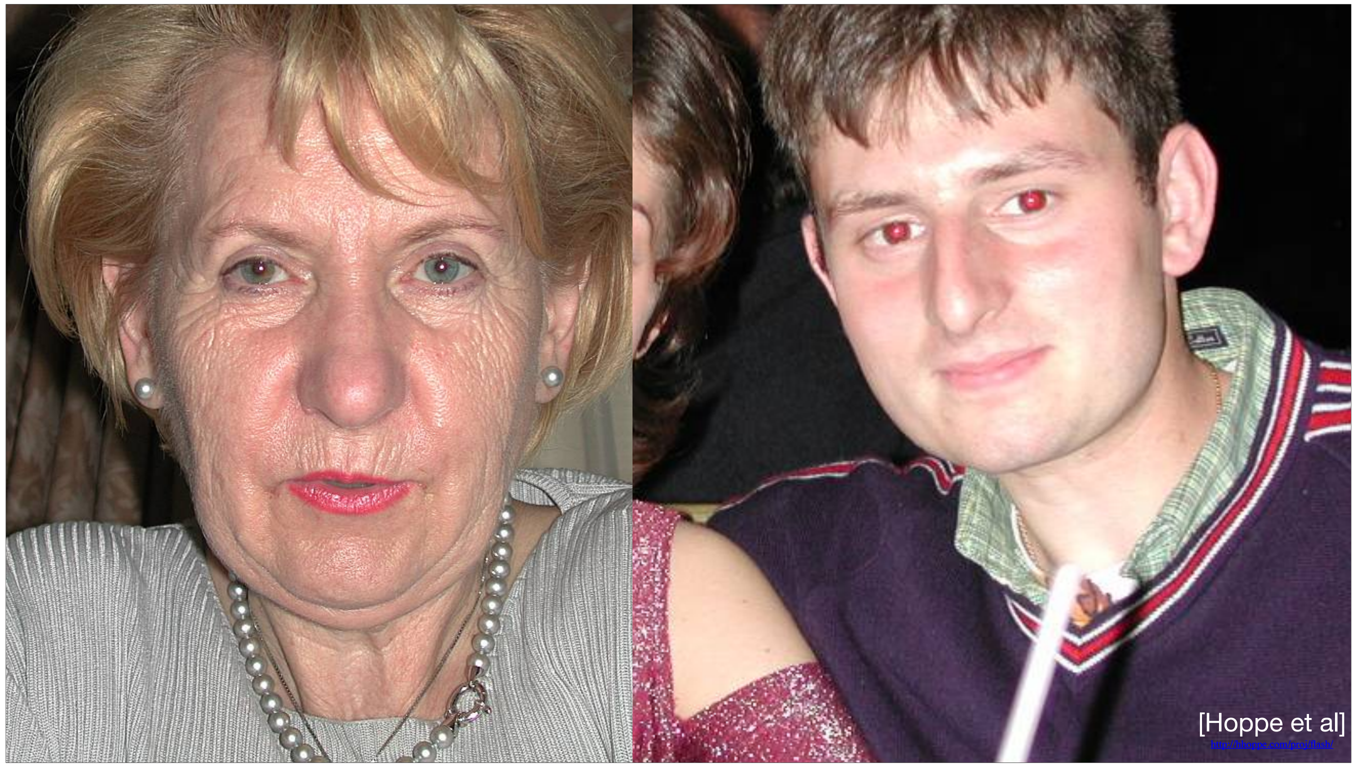
So that’s more or less how we can make images with higher dynamic range when our cameras have very limited dynamic range.



Chapter 1.5

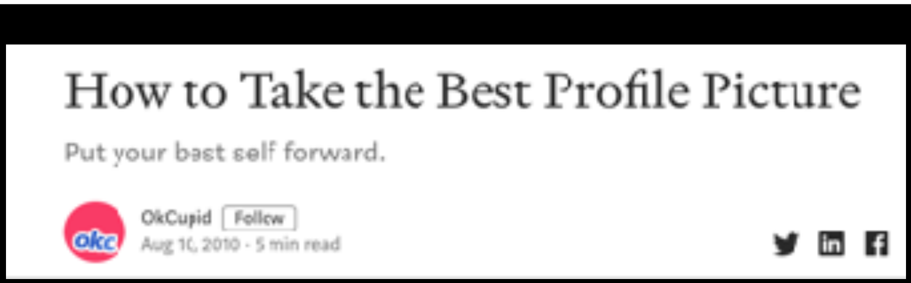
Flash

One more burst-related example before we move on.



[Hoppe et al]
http://hhoppe.com/proj/flash/

Remember when half the user-uploaded pictures online look like this? Flash photographs would give everyone red eyes, and would accentuate blemishes, wrinkles, and greasy skin. Everything looked worse.



How to Take the Best Profile Picture

Put your best self forward.

OkCupid Follow
Aug 10, 2010 · 5 min read

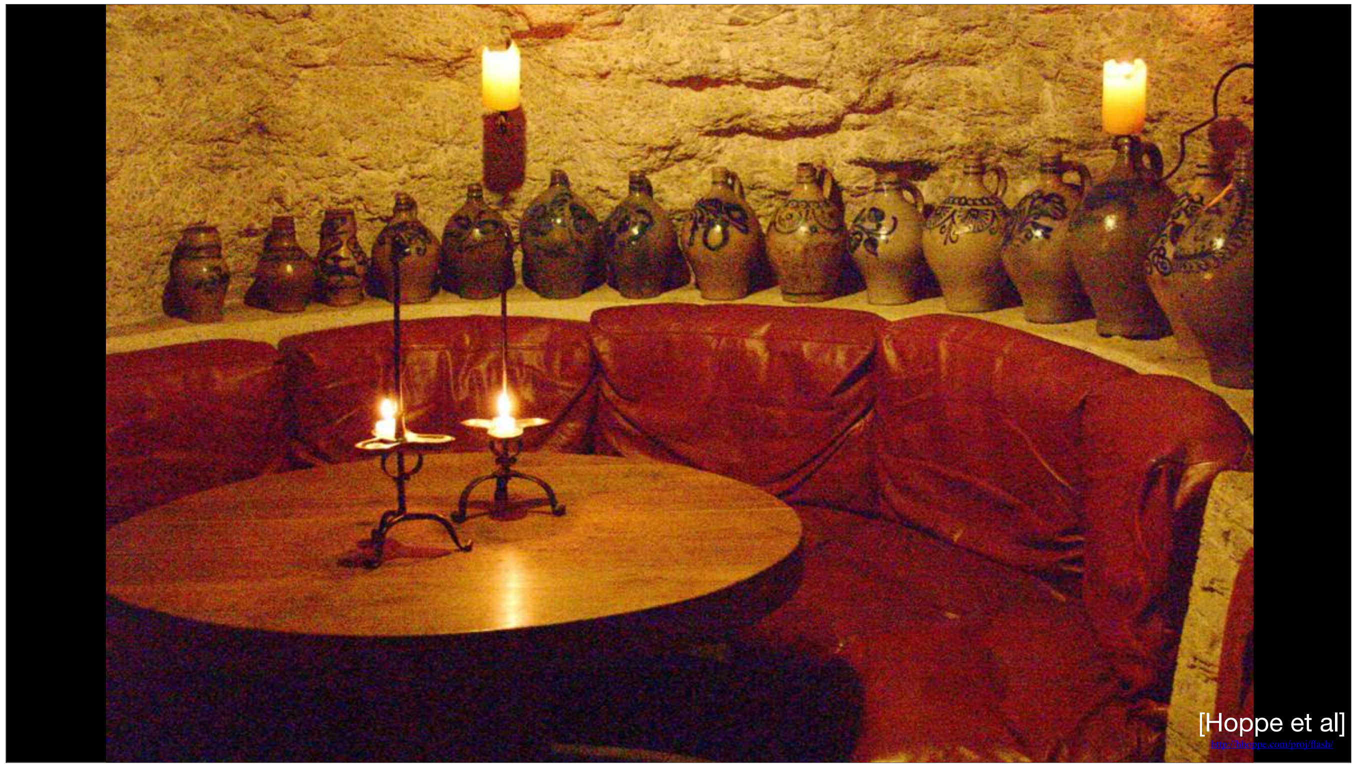
2. Flash is not your friend.

“The hard light of a flash often ages the appearance of its subject by accentuating wrinkles and blemishes [...] For example, a 28 year-old who used a flash looks similar to a 35 year-old who didn’t.”

https://theblog.okcupid.com/dont-be-ugly-by-accident-b378f261dea4

In fact, some data scientists at OK Cupid, an online dating website, said that in a sample size of 11.4 million people, those who used flash photographs as their profile picture were judged by others to be an average of 7 years older than those who didn’t.

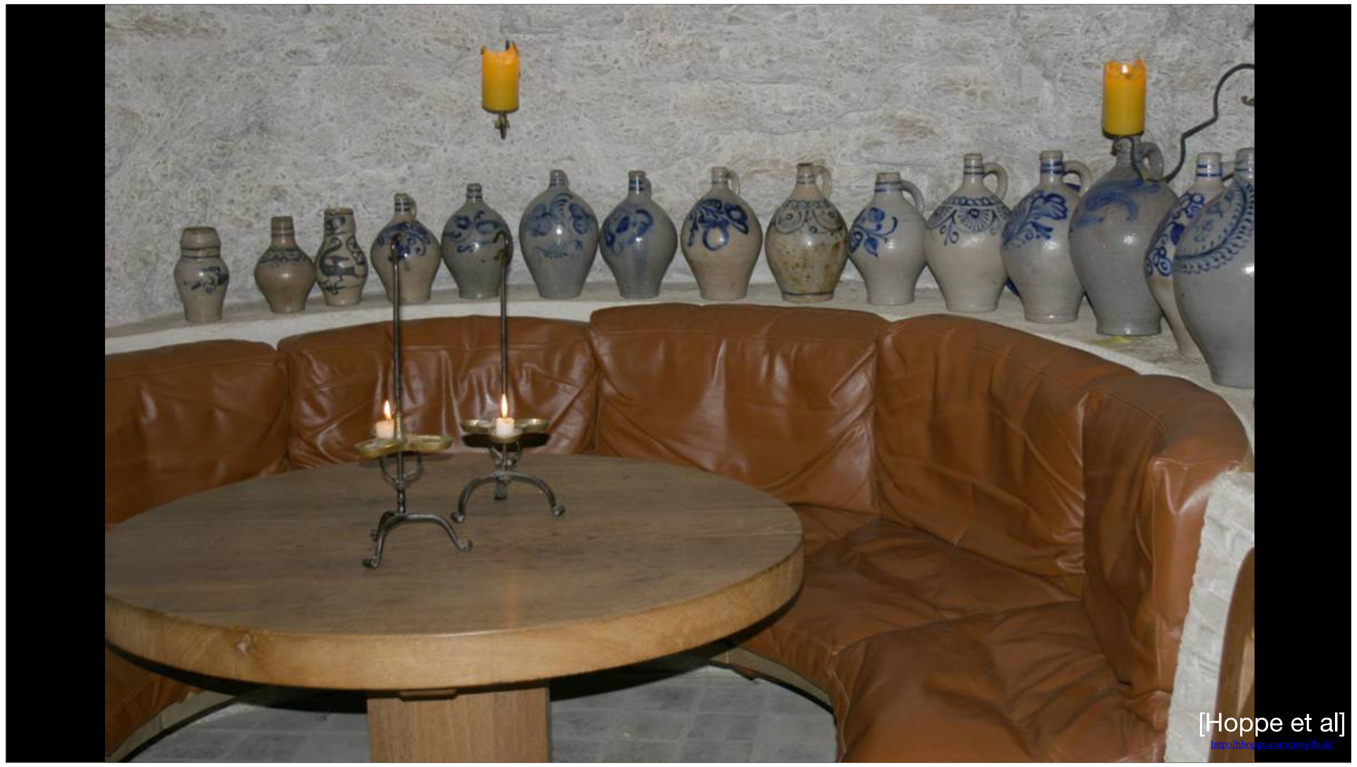
Well, these pictures were mostly from a time where your options were either to take a picture without flash, and have it be dark and noisy…



[Hoppe et al]

http://hhoppe.com/proj/flash/

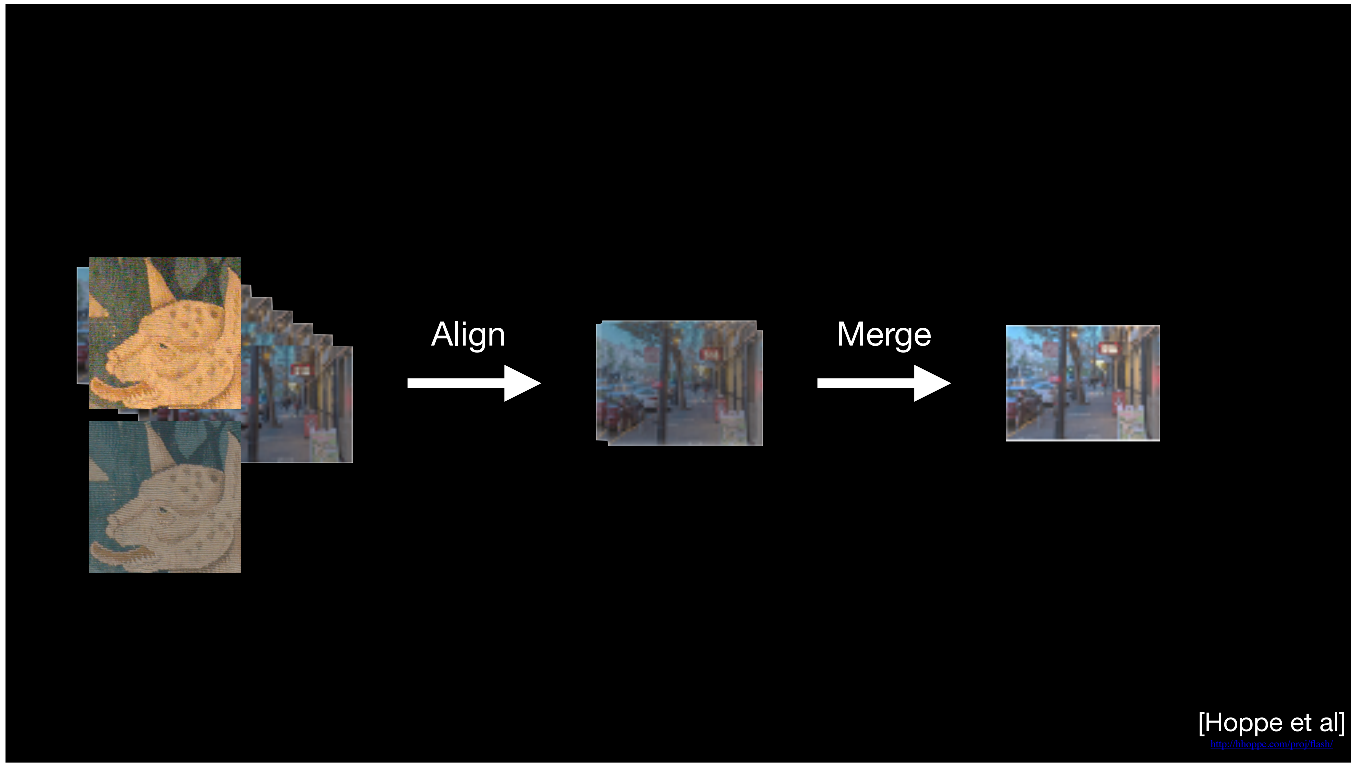
Or take a picture with flash and have it be sharp, noise-free…



[Hoppe et al]
http://hhoppe.com/proj/flash/

…but strangely lit, artificial, and jarring

We seldom see these types of photos sent around nowadays, even though flash is still used in a lot of cases. So why is that?

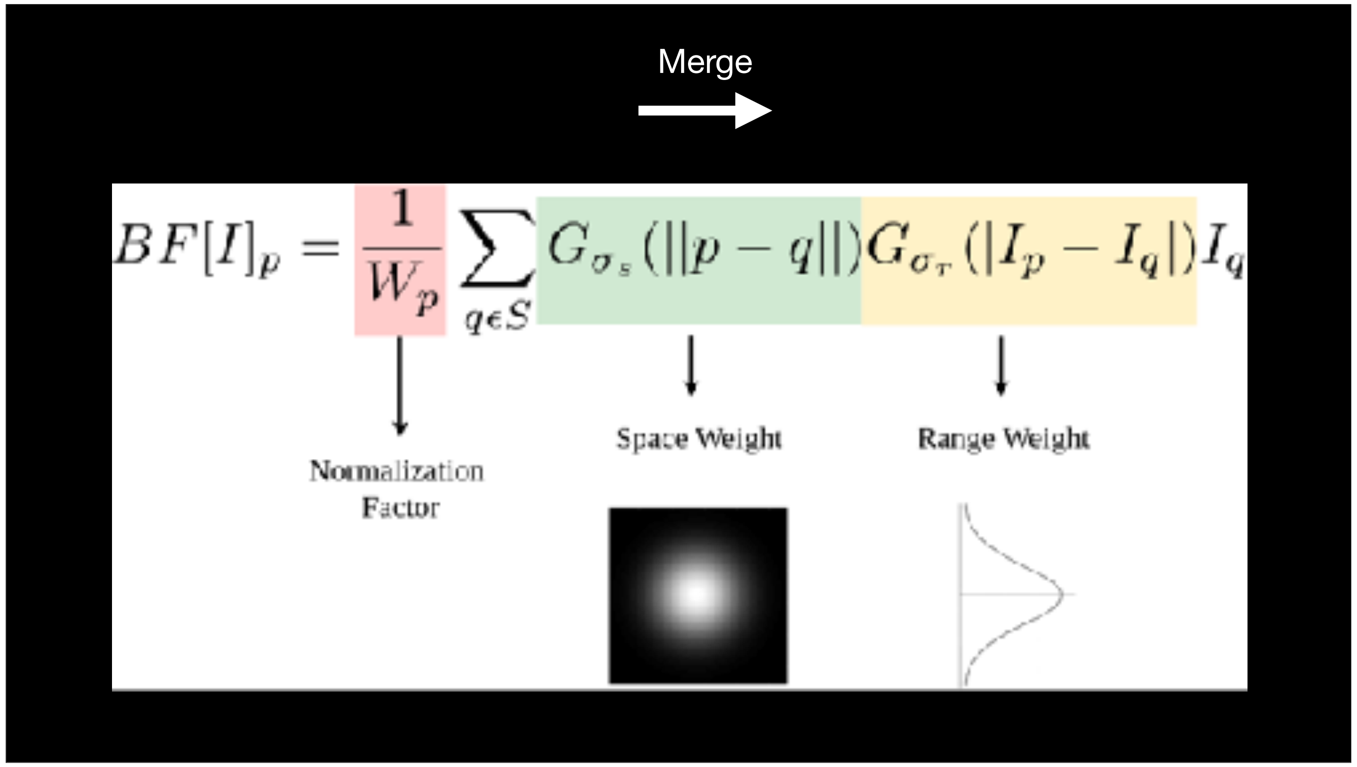


[Hoppe et al]
http://hhoppe.com/proj/flash/

The reason is that most phones and cameras nowadays use some version of this same pipeline. Except, instead of a burst of photos, they'll use two pictures, one with flash, and one without flash.

The alignment will happen in the same way as before, but in the merging stage, they need to do something a little bit smarter, because just averaging between the flash and no-flash image isn't guaranteed to give us anything better than either of them.

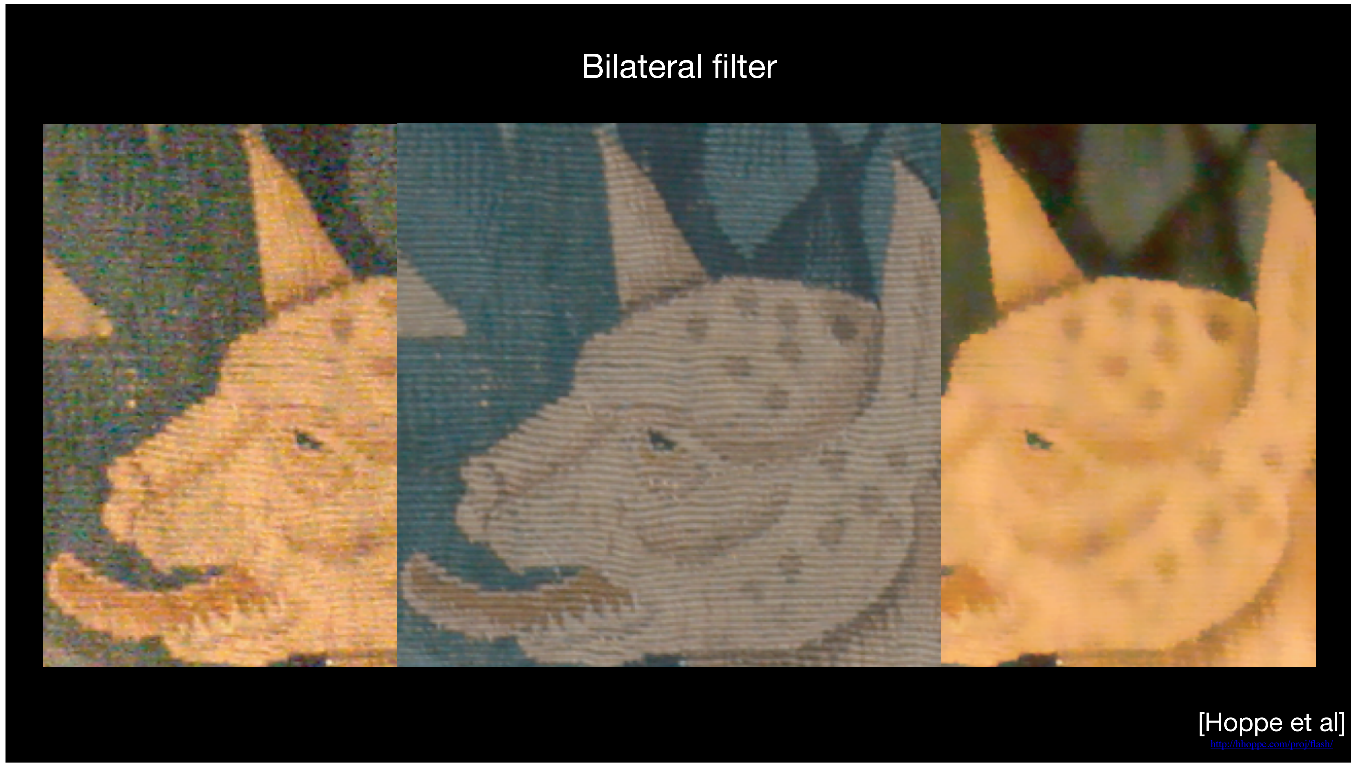
What we actually want is a final image which has the colors of the no-flash image, but without all the noise, and with the sharpness of the flash image.



$$BF[I]_p = \frac{1}{W_p} \sum_{q \epsilon S} G_{\sigma_s}(||p-q||) G_{\sigma_r}(|I_p - I_q|) I_q$$

For this, we look back at the bilateral filter.
The basic idea is that for each pixel, you have a filter kernel which is defined by spatial distance as well as color distance.

This means that pixels which are similar in color get blurred together more, and pixels which are more different are blurred together less.
This works very well as a filter for removing certain types of noise in images, because it preserves edges very effectively.

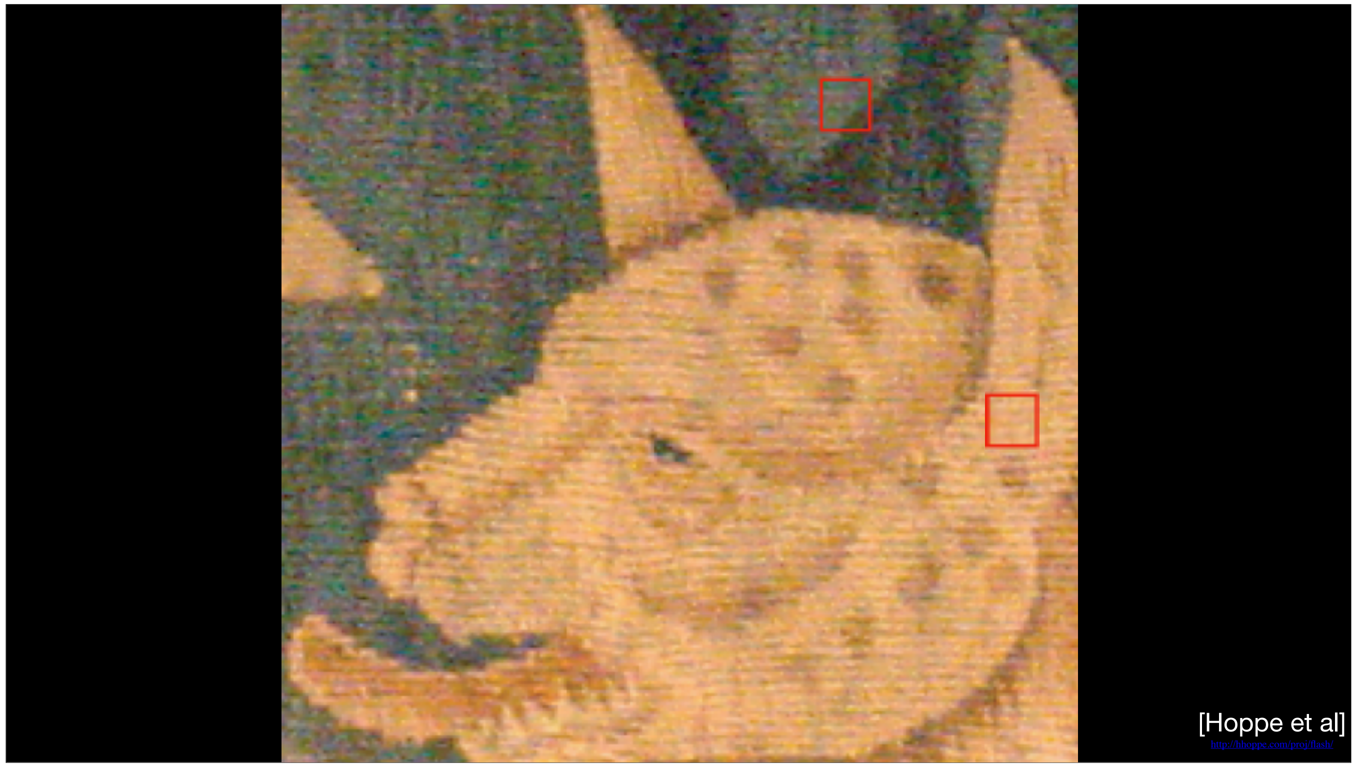


Bilateral filter

[Hoppe et al]
http://hhoppe.com/proj/flash/

We can see that applying the bilateral filter on the no-flash image definitely cleans up most of the noise, but it also results in a loss of detail.
We can tell this by looking at the flash image, for reference.
[slide]
This is clearly a rug or tapestry, but a lot of the details of the texture seem to be missing.

And the reason this happens is because…

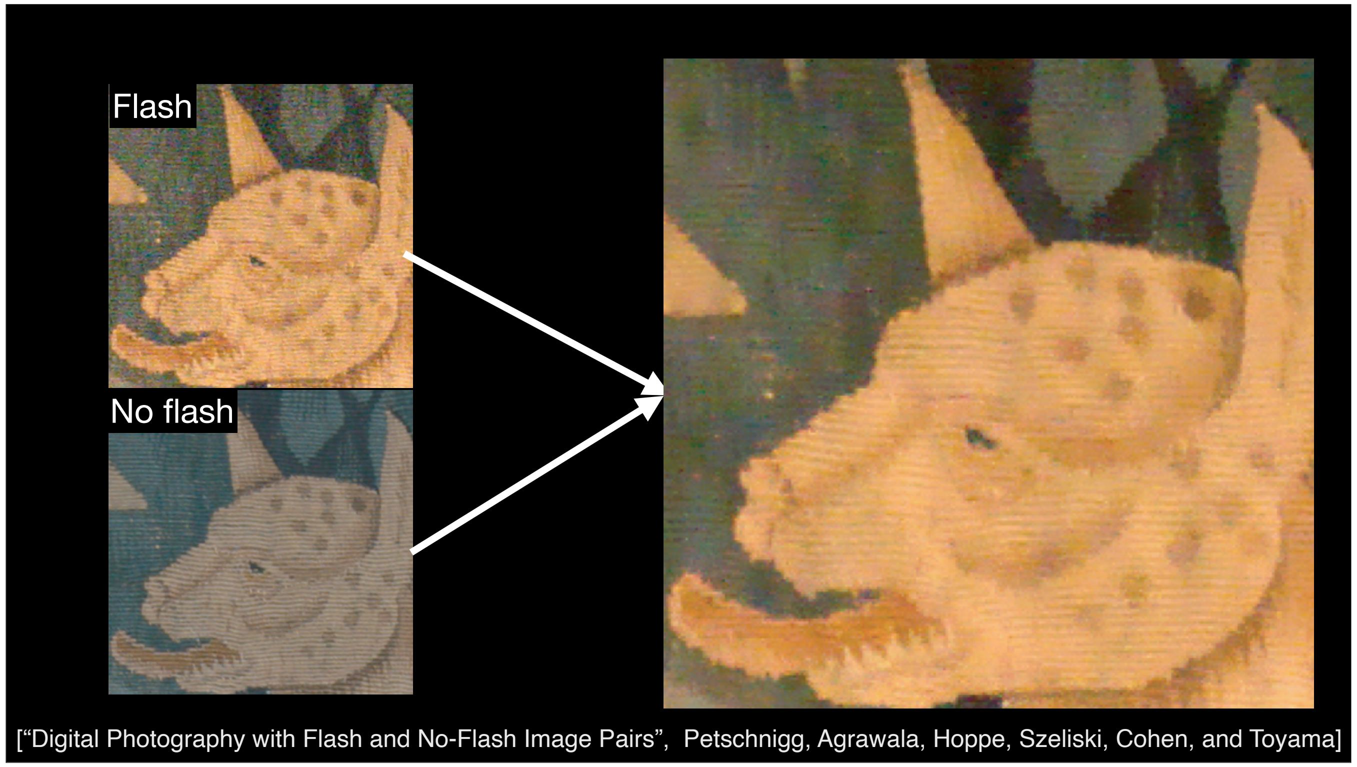


[Hoppe et al]

http://hhoppe.com/proj/flash/

At the pixel or kernel level, it’s very hard to tell apart real texture[slide], like this texture on the tapestry, from image noise[slide].

If we scale them up [slide], at this resolution they’re almost indistinguishable. Which one has the edge we want to keep?

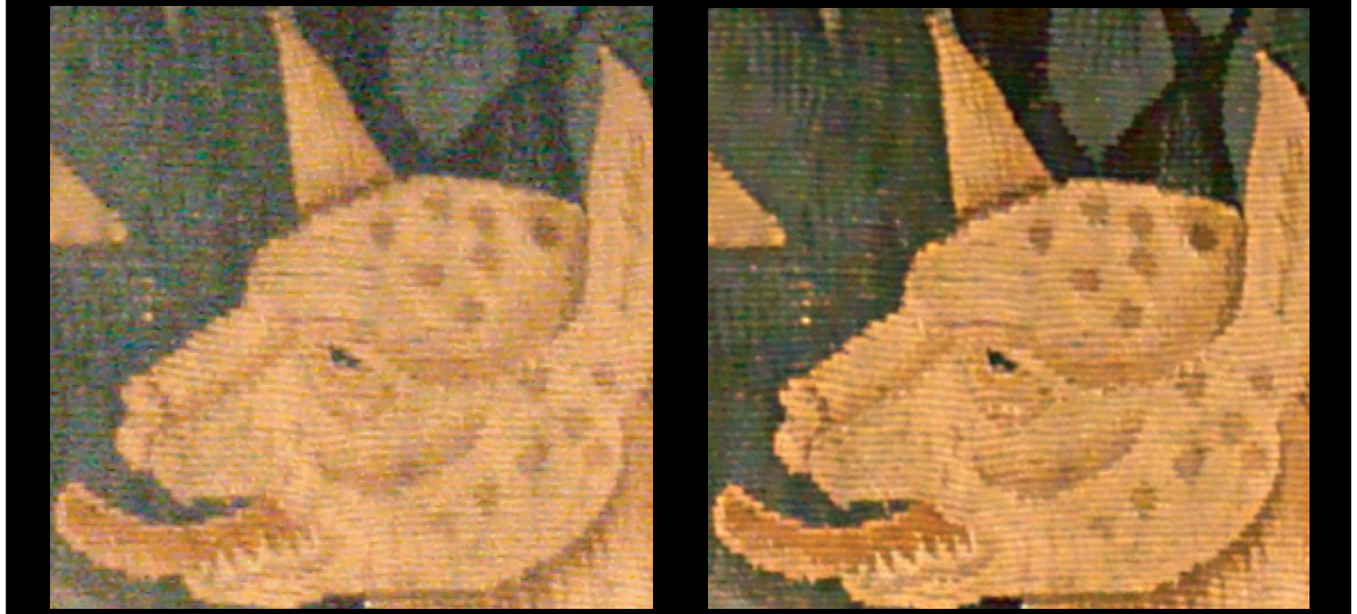


[“Digital Photography with Flash and No-Flash Image Pairs”, Petschnigg, Agrawala, Hoppe, Szeliski, Cohen, and Toyama]

The main takeaway, or the main trick, that was used to solve this problem was to use what’s called the JOINT bilateral filter, which is very similar to the regular bilateral filter (which computes weights based off of spatial distance and color distance) but instead of using the noisy no-flash image to compute the color distance, it computes the color distance using pixels in the flash image instead, since there’s less noise.

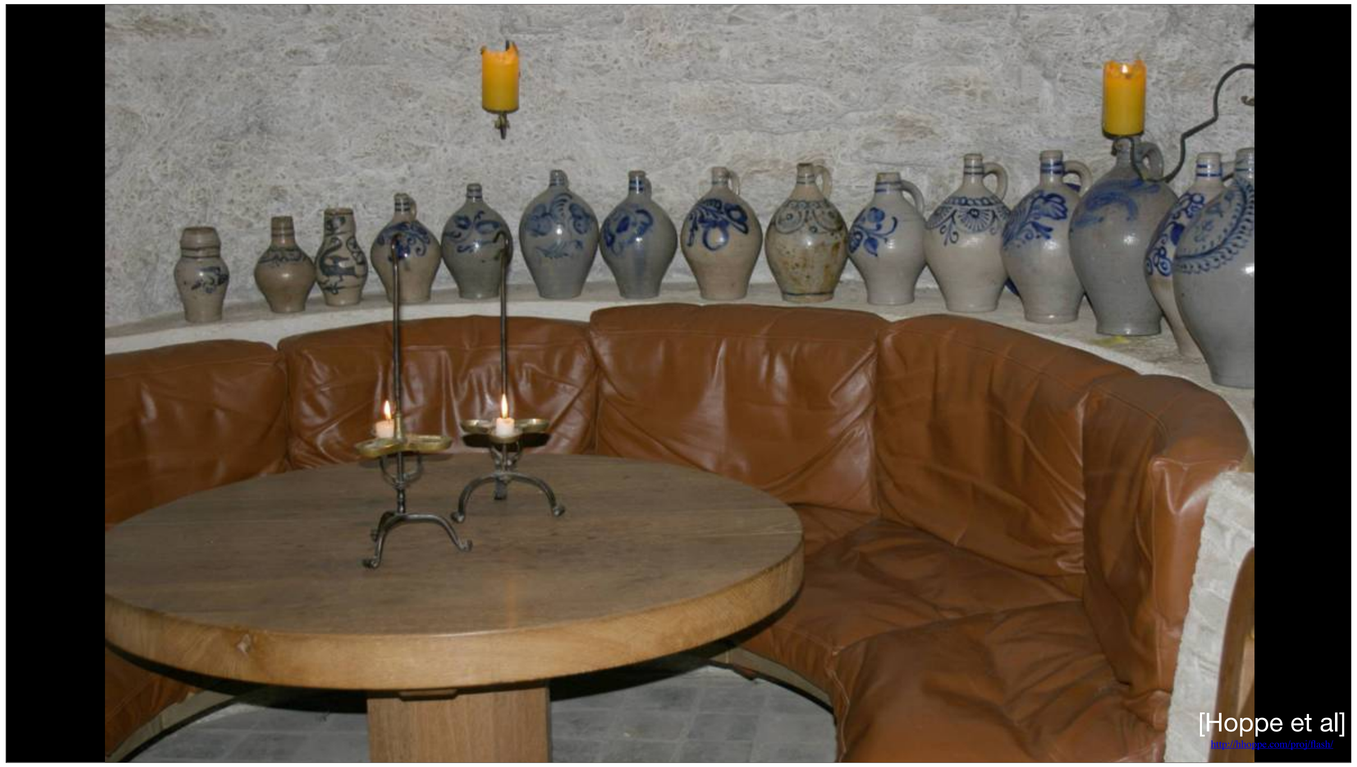
But it still only applies the filter to the no-flash image, which means that the resulting pixels will be weighted combinations of pixels in the no-flash image. This is what we want, because it’ll preserve the natural colors, but will get rid of noise and preserve edges based on the flash image.

There are actually a few more details if you want to get really sharp images with a lot of details, and you can take a look at the paper if you’re interested.



[Hoppe et al]
http://hhoppe.com/proj/flash/

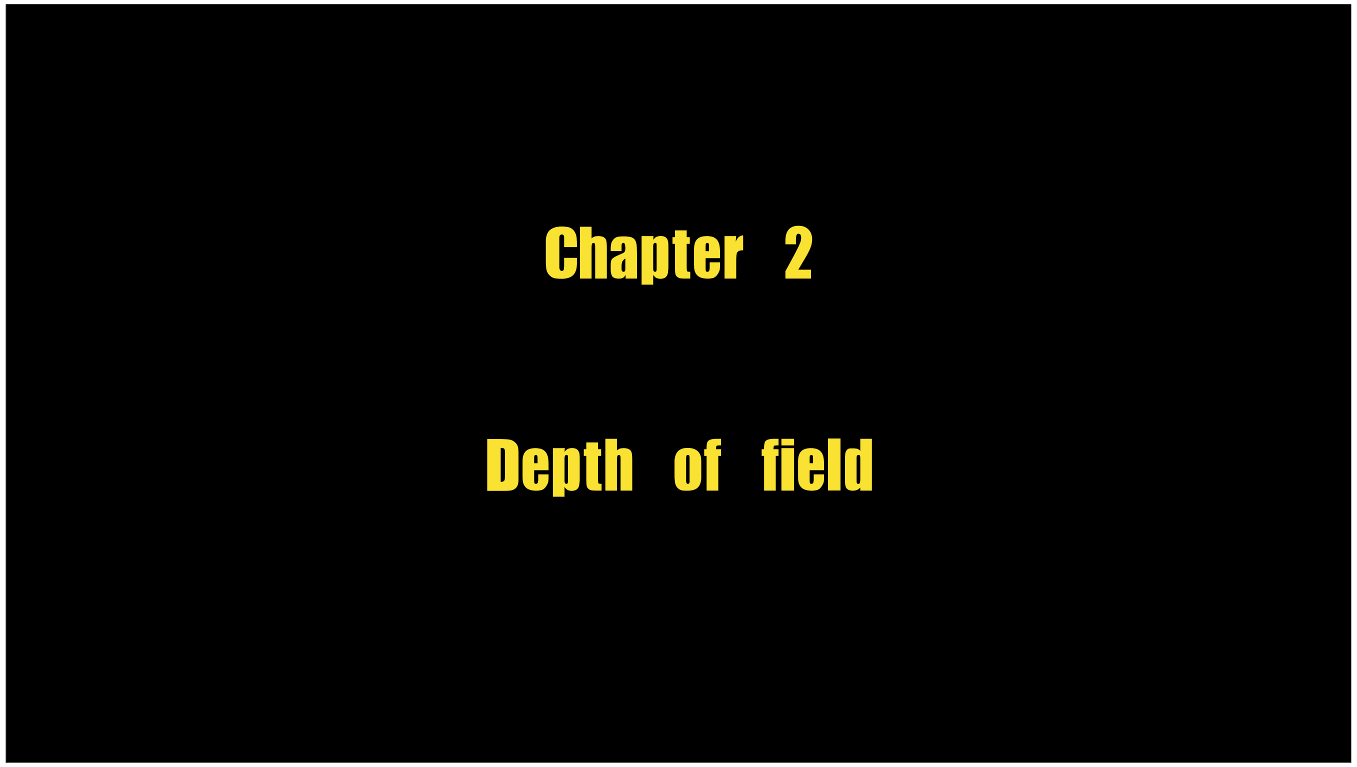
Here’s a comparison of the image before and after applying this flash-merging.



[Hoppe et al]
http://hhoppe.com/proj/flash/

The answer to why we don’t see too many images like this anymore isn’t necessarily because nobody is taking them…

But rather because cameras are smarter, and can combine other information it captures so that we never have to look at pictures like this.

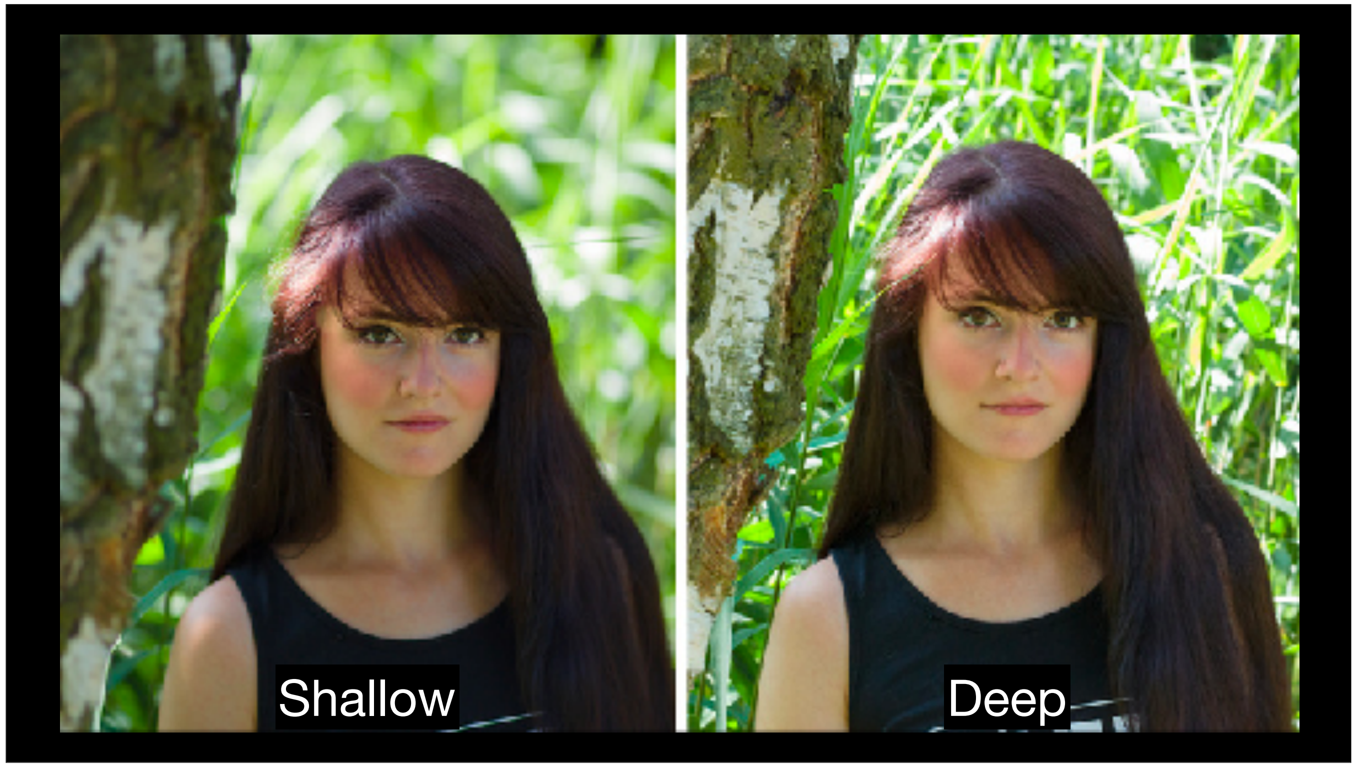


Chapter 2

Depth of field

Now for chapter 2…

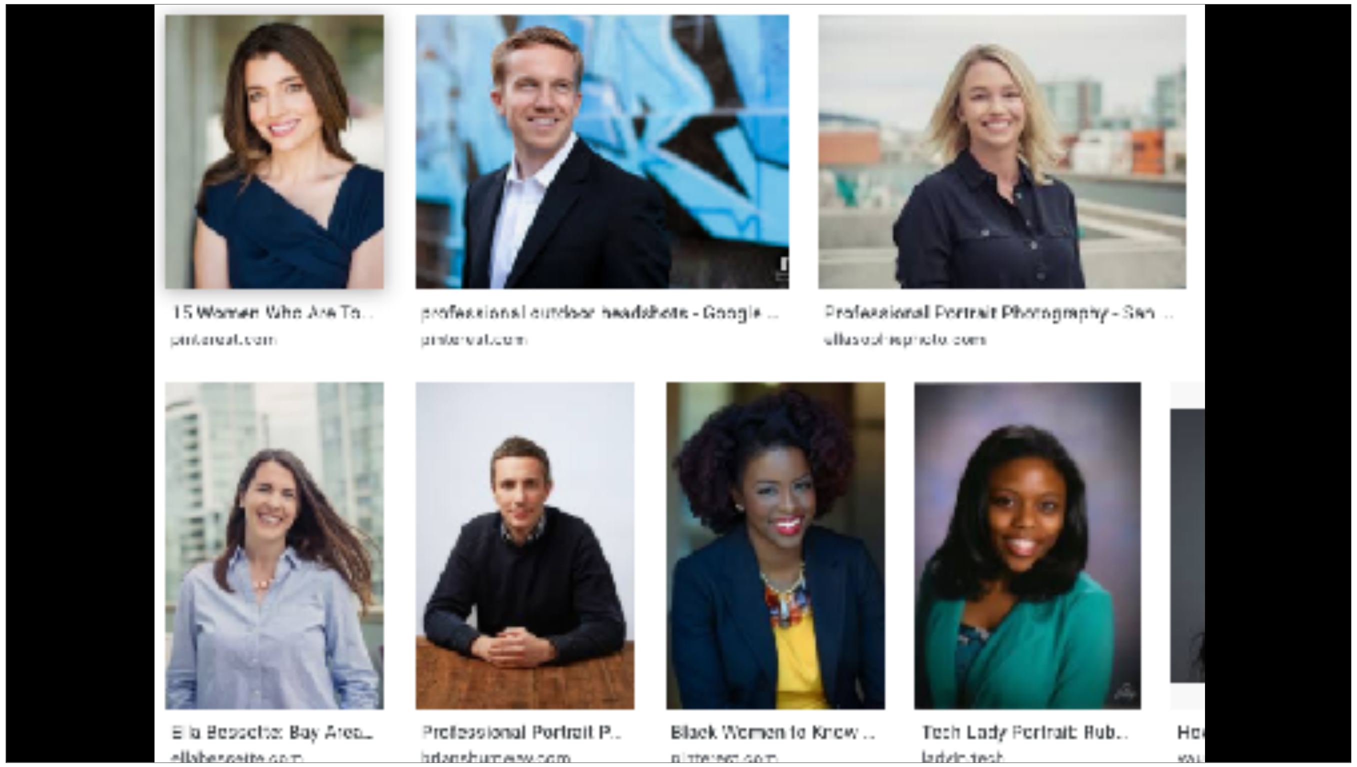
Depth of field.



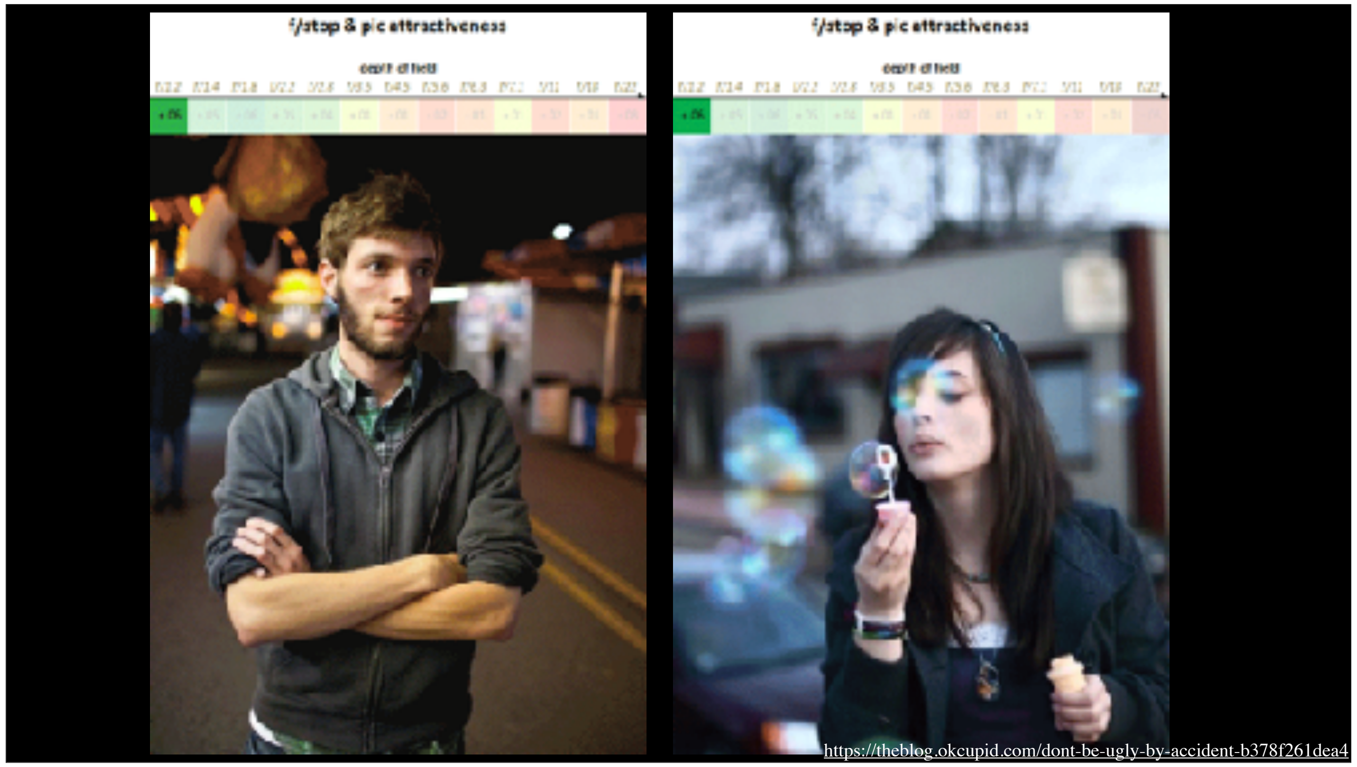
Depth of field is the idea that things at a certain distance (your focal distance) will be perfectly in focus.

As you move away from that distance, things will move out of focus, and become blurrier.

The depth of field basically defines how quickly that process happens.
With a shallow depth of field, anything like a couple inches further away or closer to the camera will be totally blurry,
But with a deep depth of field, sometimes everything looks like it’s in focus.

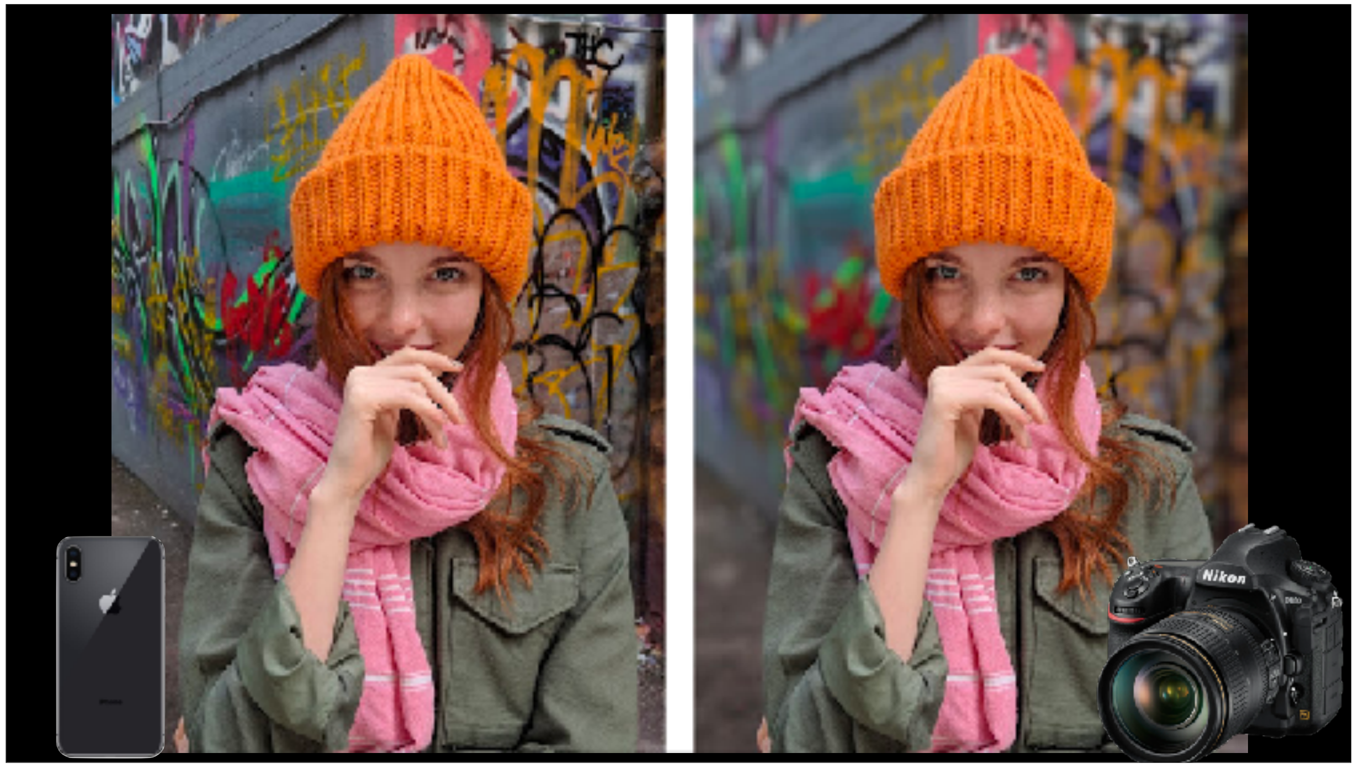

15 Women Who Are To...
pinterest.com
professional outdoor headshots - Google ...
pinterest.com
Professional Portrait Photography - San ...
ellasophiaphoto.com
Ella Bessette: Bay Area...
ellabessette.com
Professional Portrait P...
brianshumway.com
Black Women to Know ...
pinterest.com
Tech Lady Portrait Rub...
ladyin.tech
Ho

Over the years, shallow depth of field photography has come to be the standard for a professional looking photographs, and there are a number of theories for why this is the case, whether it be because there are less distractions in the background, or we’ve just associated that effect with fancier cameras.



https://theblog.okcupid.com/dont-be-ugly-by-accident-b378f261dea4

Actually, in the same online dating study cited earlier, they also show that shallow depth of fields were also correlated with how attracted people were to the photos, with many users saying that shallow depth of field photographs just seemed more “intimate” or “personal”.

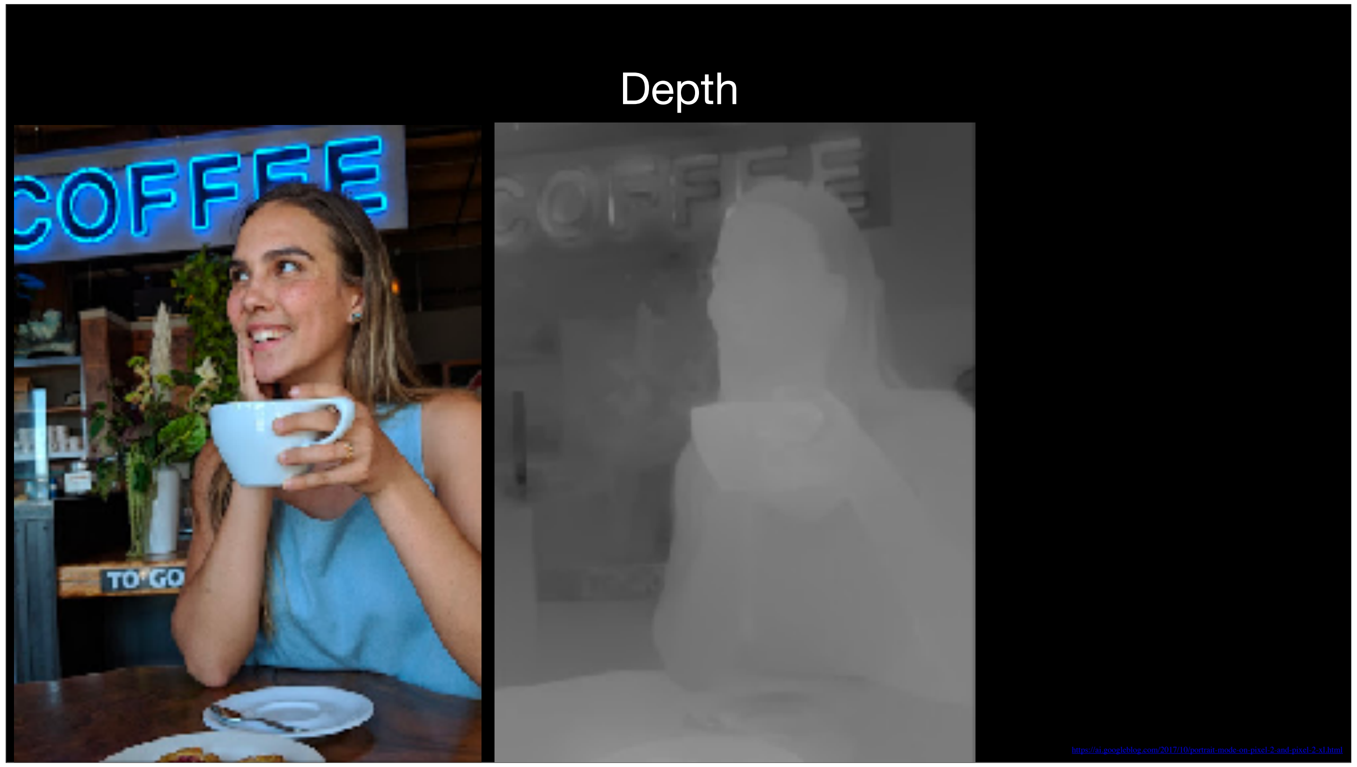


As we mentioned before, because they’re so small, smartphone cameras have some intrinsic limitations, like small apertures, and short focal lengths.

What that means is that all the photos they take are going to be nearly all-in focus, and can’t achieve the same shallow depth of field that larger cameras can.

So, how might we get around this obstacle? How can we simulate shallow depth of field in a smartphone photo?

There are two main techniques that you see in smartphone cameras nowadays, and we’ll discuss both of them.



Depth

https://ai.googleblog.com/2017/10/portrait-mode-on-pixel-2-and-pixel-2-xl.html

The first way is by using depth estimation.

This is actually one of the biggest reasons why lately, we’re starting to see…

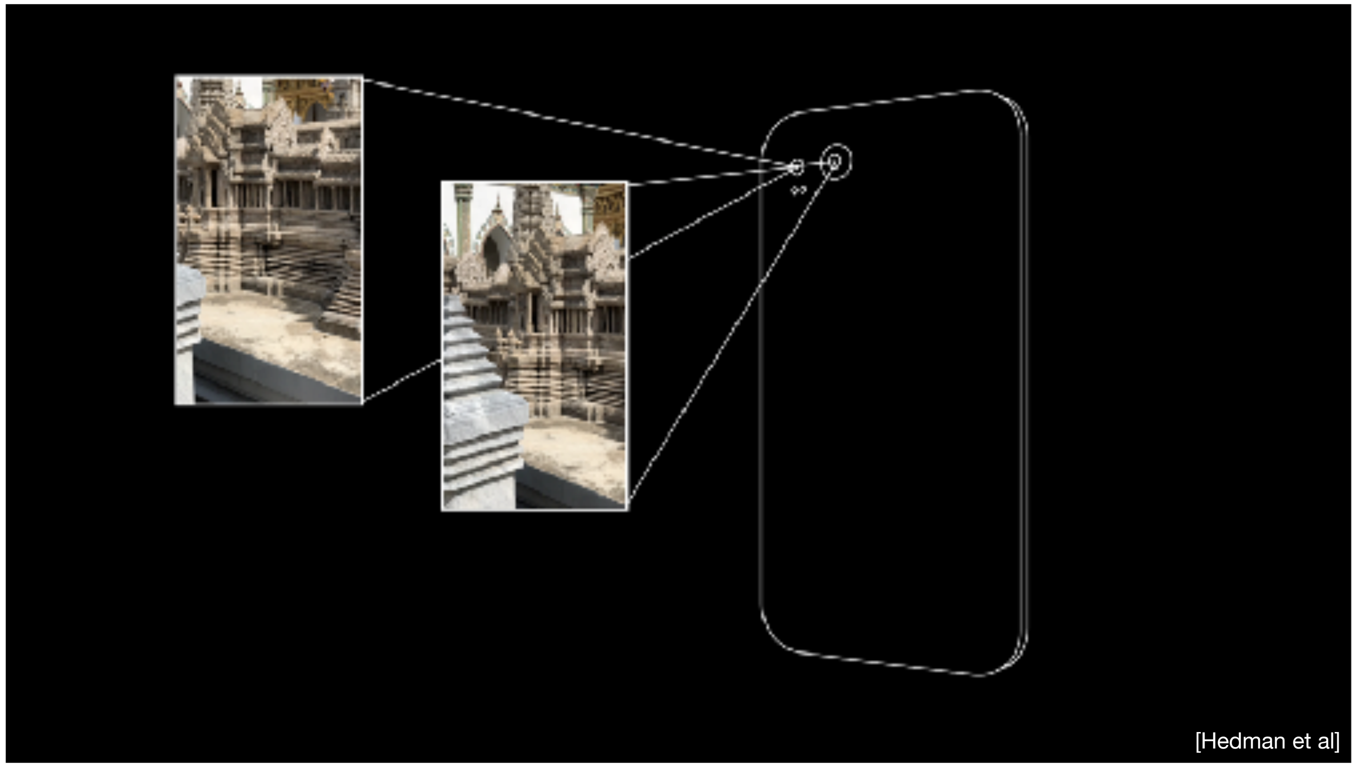
…multiple cameras on smartphones. We started by adding two cameras

…then three…

…then four…
[slide]

[Hedman et al]

There was a reason why people started adding two or more cameras to smartphones



[Hedman et al]

…and that's because it gives us the freedom to do stereo depth estimation

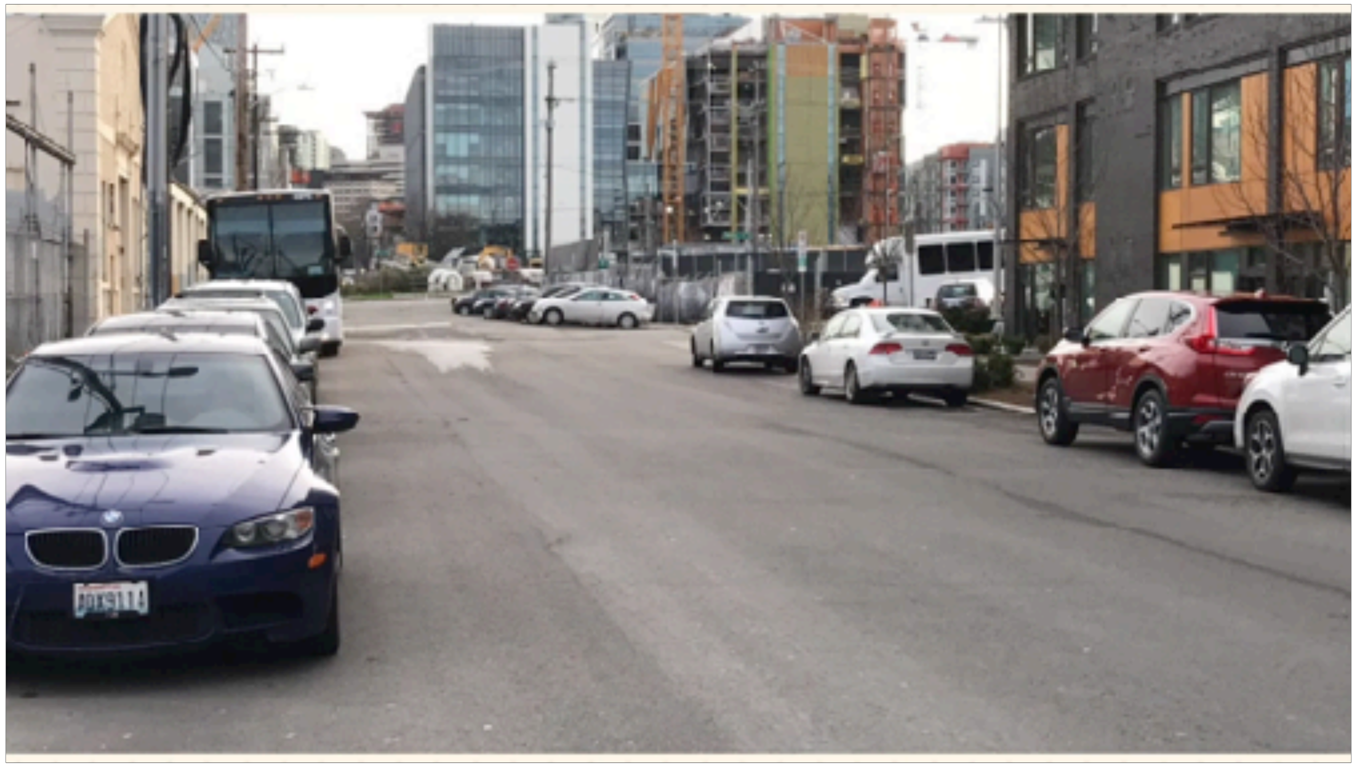
Of course, like with everything else we've seen in this lecture, it's not quite as simple as that, since…

[Hedman et al]

…usually when we’re doing stereo, we have cameras which are pretty far apart, but in this case they’re practically touching each other.

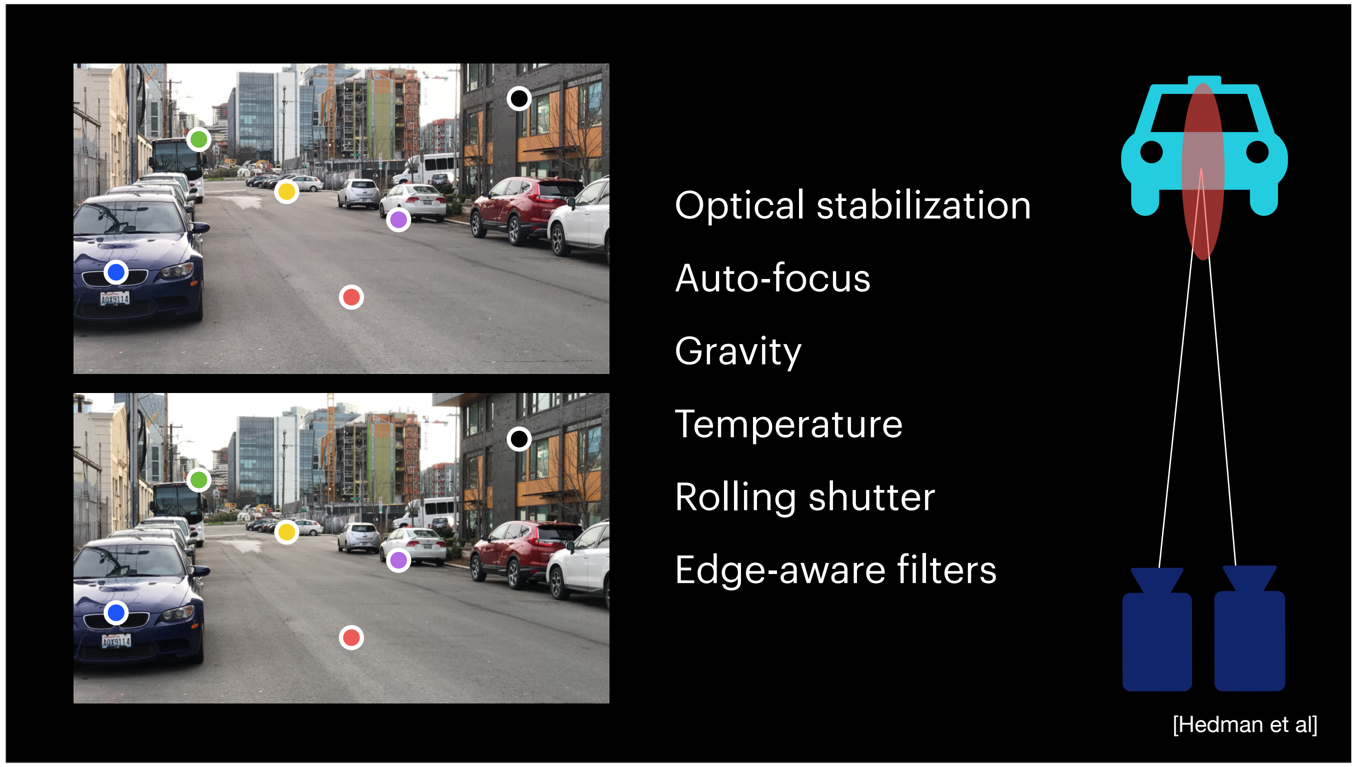
And that changes a lot for depth estimation.

For example, this is the image pair you get from a dual-camera phone.
Here’s the first image [slide]
And here’s the second image [slide]



Since the distance between the two cameras --- the baseline --- is so small, the images are barely different.

On one hand, this makes stereo matching fast, as you can keep the search region small.



Optical stabilization

Auto-focus

Gravity

Temperature

Rolling shutter

Edge-aware filters

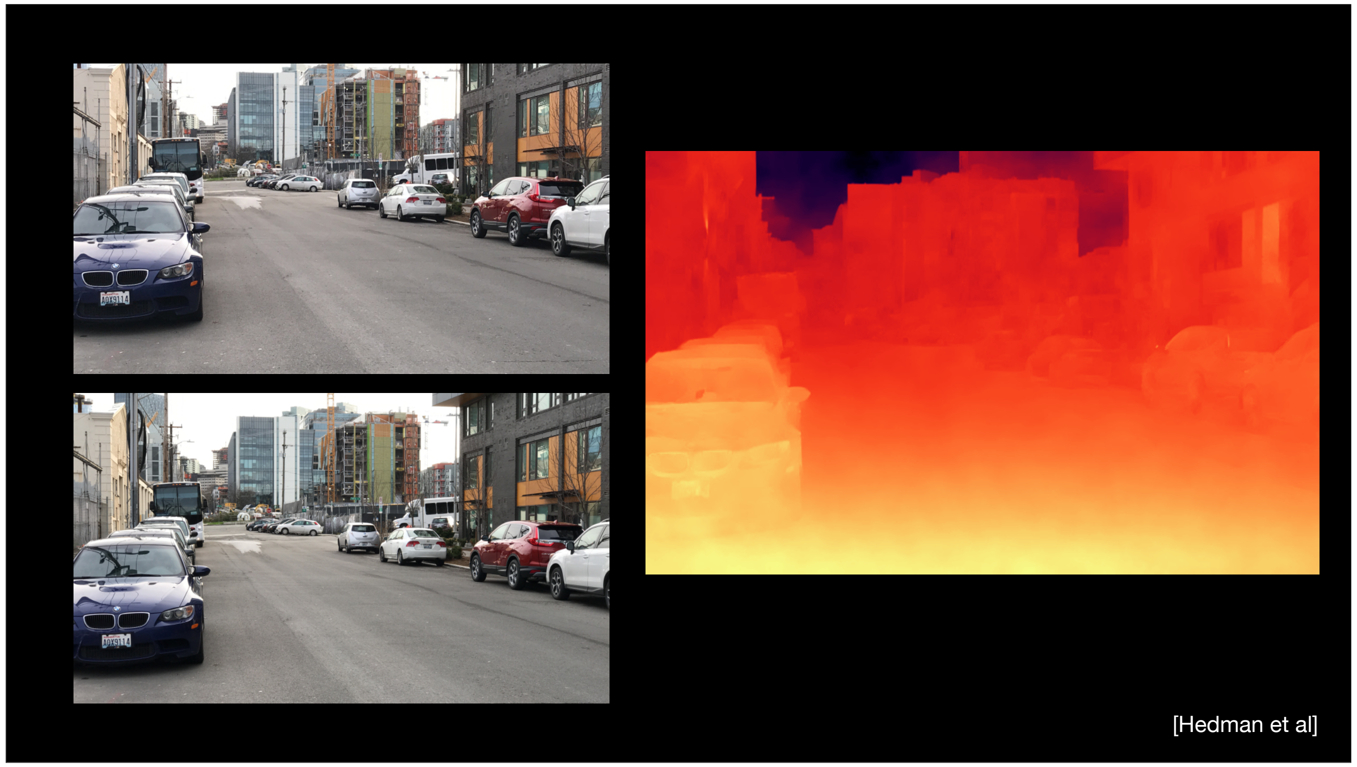
[Hedman et al]

But at the same time, for almost all points in the scene [slide] it makes the triangulation angle very narrow [slide], which means that the reconstructed depth has a very high uncertainty [slide], and can be very unreliable.

And it gets worse: in practice the cameras move and rotate independently of each other. [slide]

This happens because of otherwise desirable features in cell-phone cameras:
- Auto-focus
- Optical stabilization
- Rolling shutter

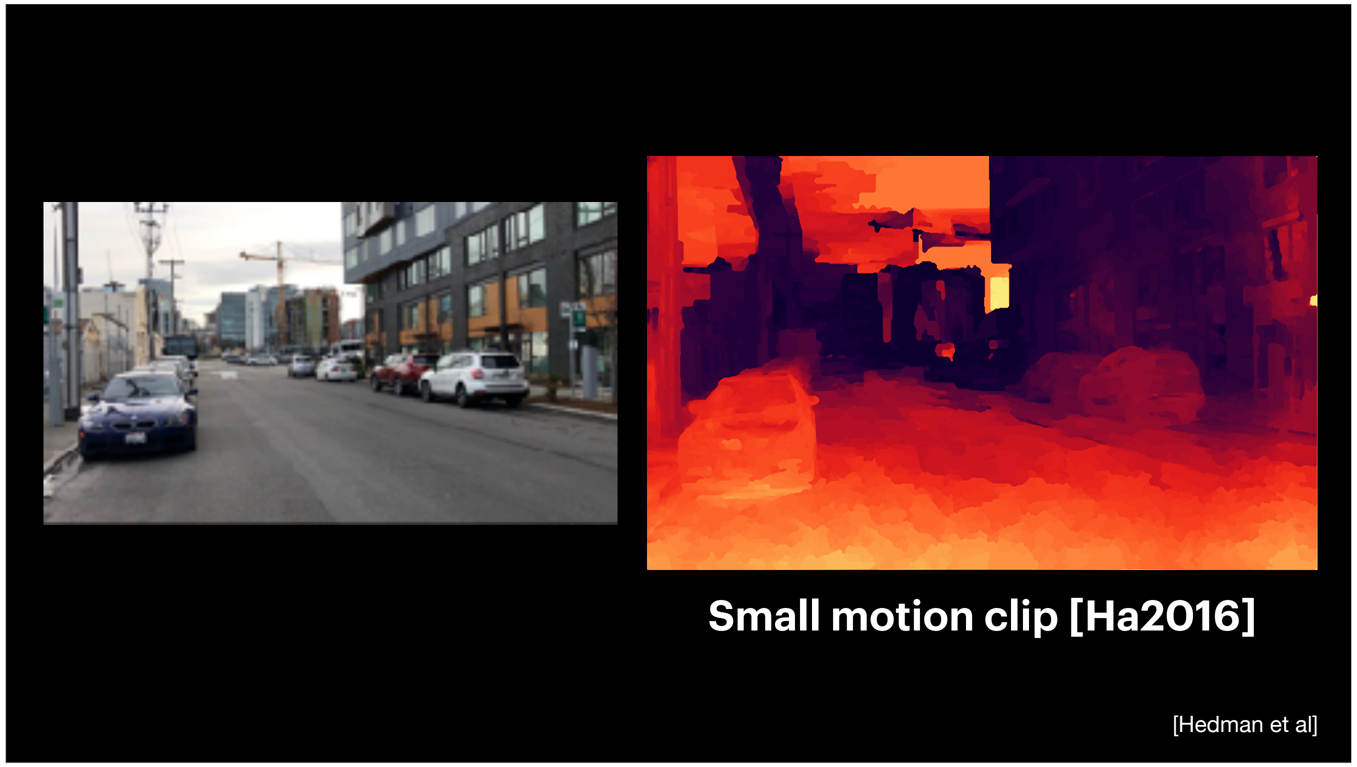
In fact, even gravity plays a role here.



[Hedman et al]

So for this reason, most algorithms which estimate depth from “micro-baseline” cameras use aggressive edge-aware filtering, like a bilateral filter

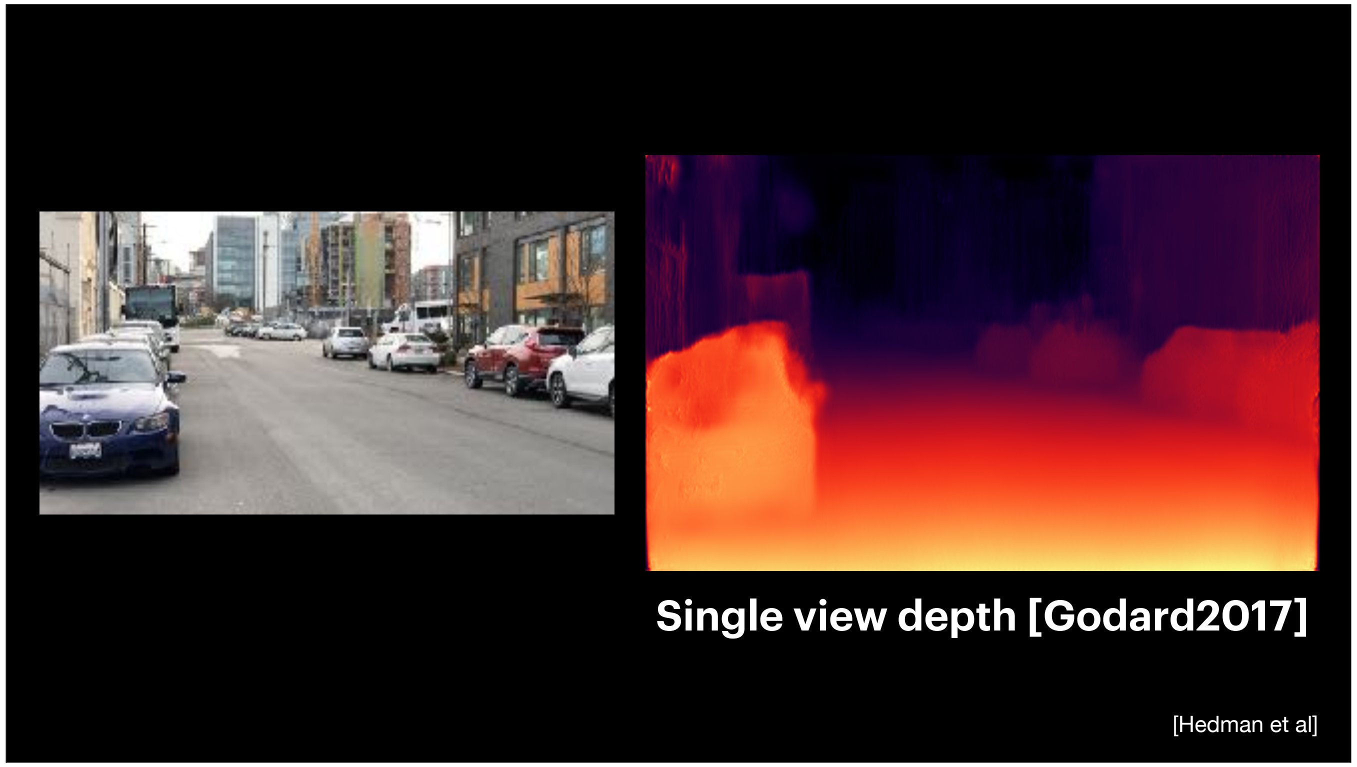
That results in smooth depth maps that also respect object boundaries, but also introduces a lot of low-frequency error to the estimated depth values.



Small motion clip [Ha2016]

[Hedman et al]

There are also other options, like methods which use a single camera and capture a short video clip, with your natural hand movement, and estimate depth based off the observed parallax from those frames.



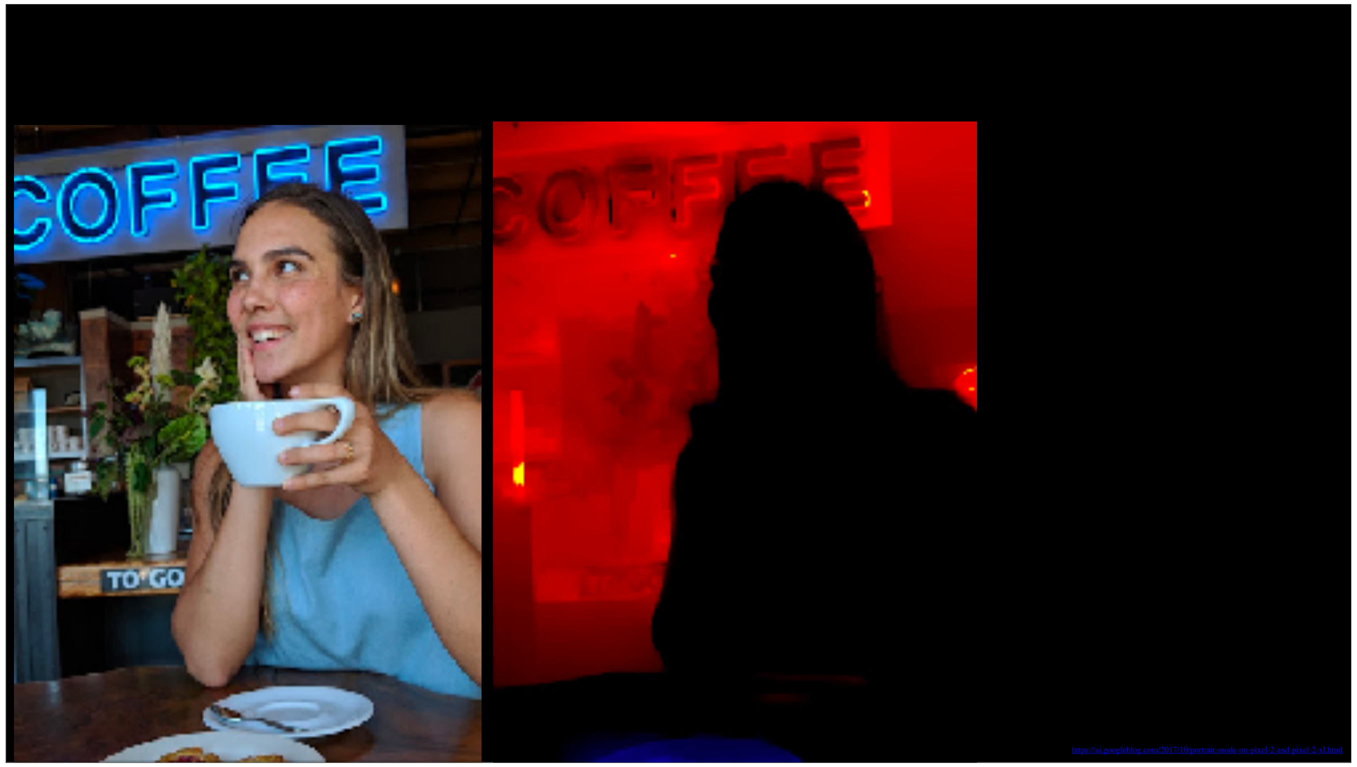
Single view depth [Godard2017]

[Hedman et al]

Or even methods that use neural networks to directly estimate depth maps from a single image.

But we’re trying to simulate depth of field?

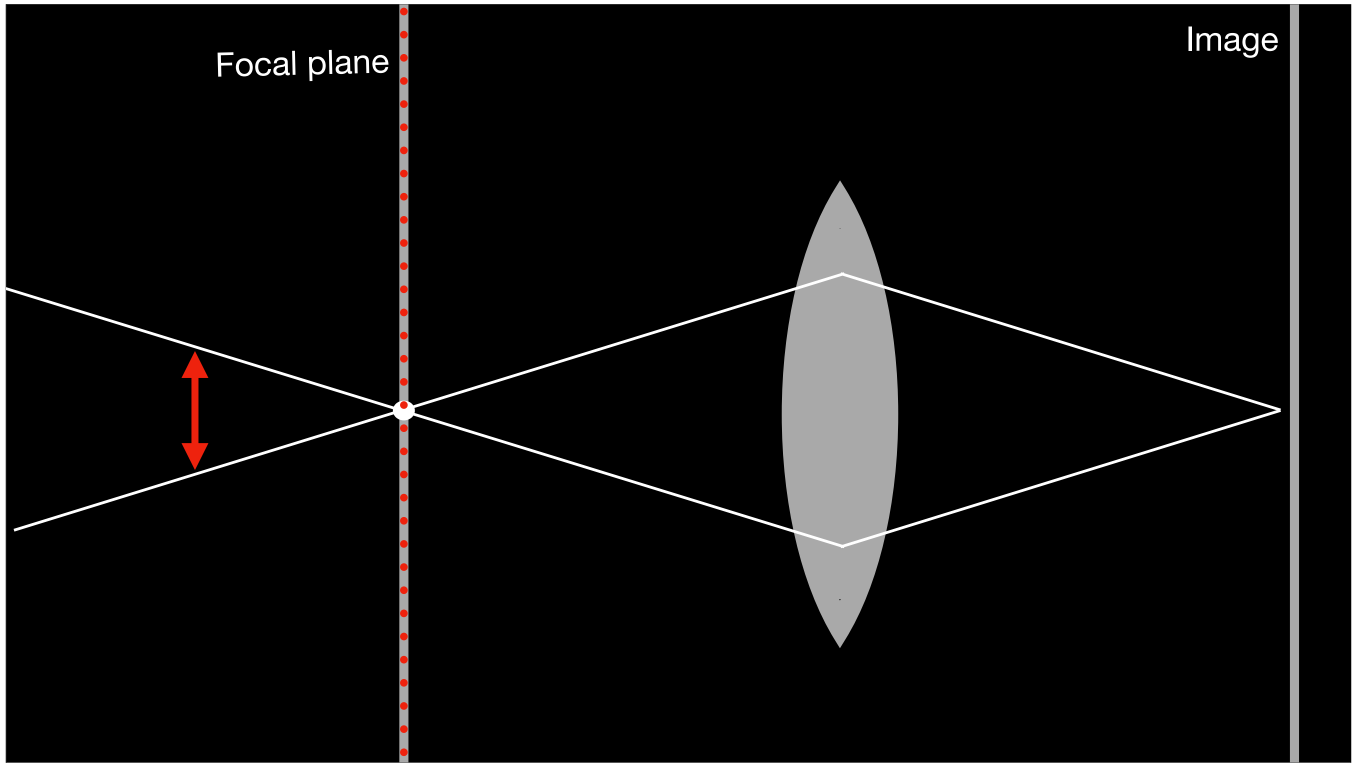
How does depth estimation tie into all of this?



https://ai.googleblog.com/2017/10/portrait-mode-on-pixel-2-and-pixel-2-xl.html

If we have a depth map, we can pick any subject, say the person in the photo, and blur out things which have a larger or smaller depth than the person.
[slide]
Here we can see a map which tells us how far each pixel's depth is from our artificially defined focal plane, which we've centered around the person.
As it gets brighter red, it means the pixel is further and further behind the focal plane.
As it gets brighter blue, it means that the pixel is closer and closer to the camera.

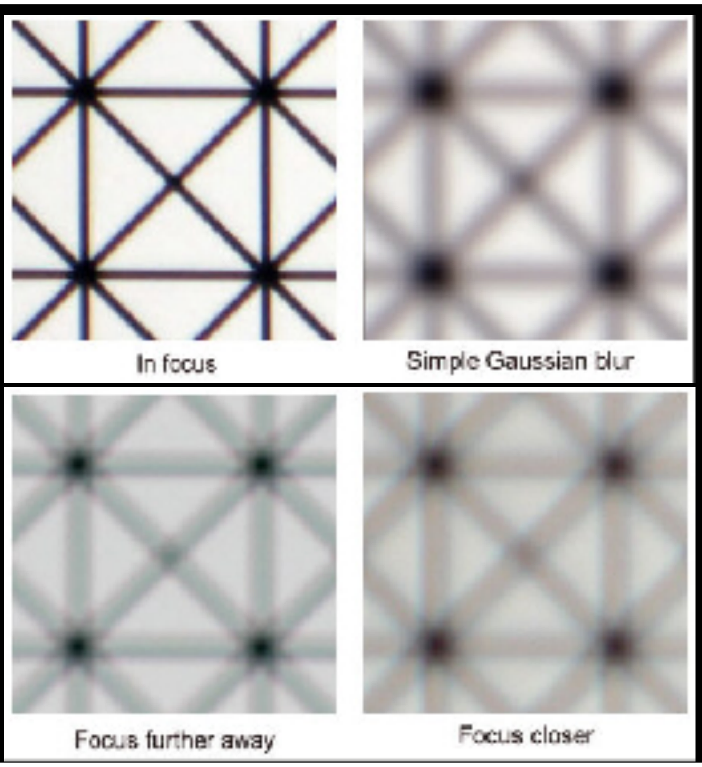
What do we do with this information? First, we need to refresh our memory on how depth of field works…



We have the image plane on the right, which is where the image is captured, and the have this focal plane, where everything is in focus.

As we move away from that focal plane, the circle of confusion increases in size. [slide]

This means that the further our object is from the focal plane, the more (or larger) blur we should be applying.



In focus

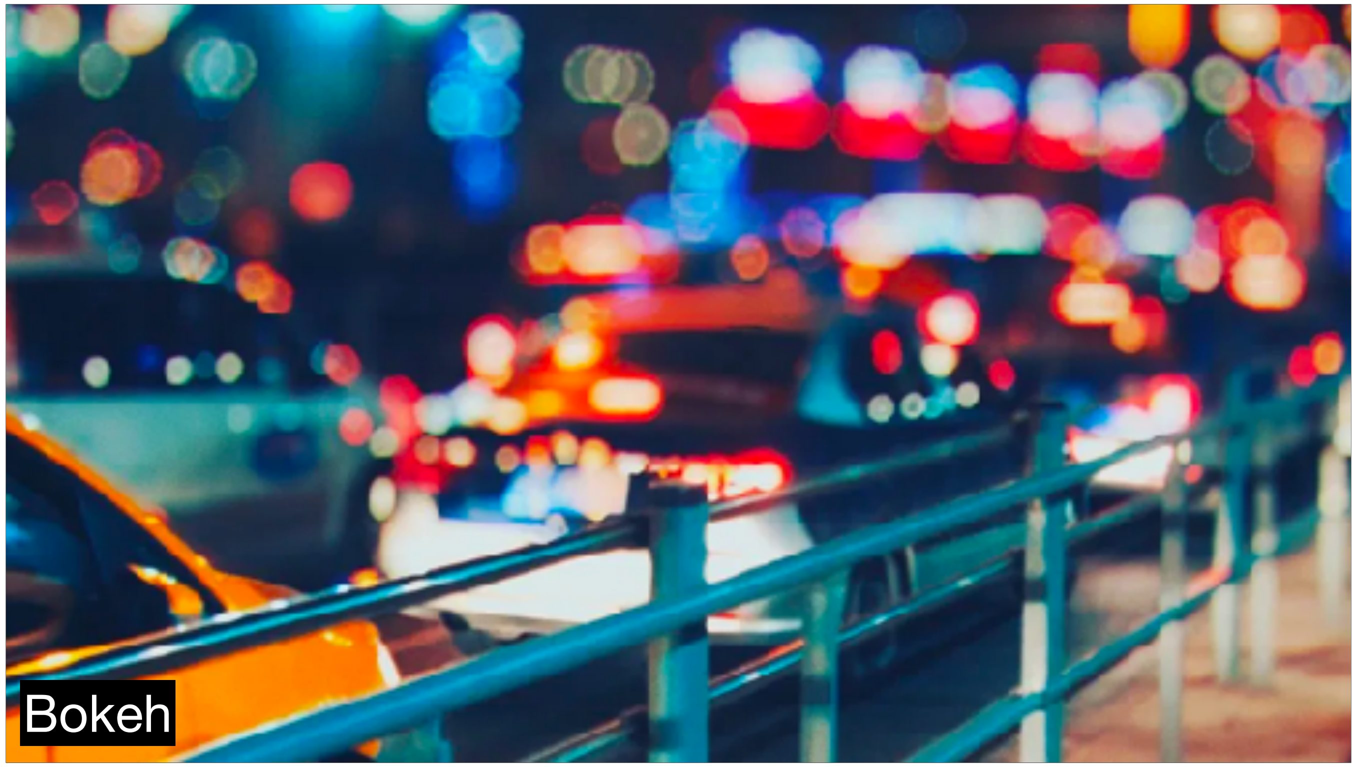
Simple Gaussian blur

Focus further away

Focus closer

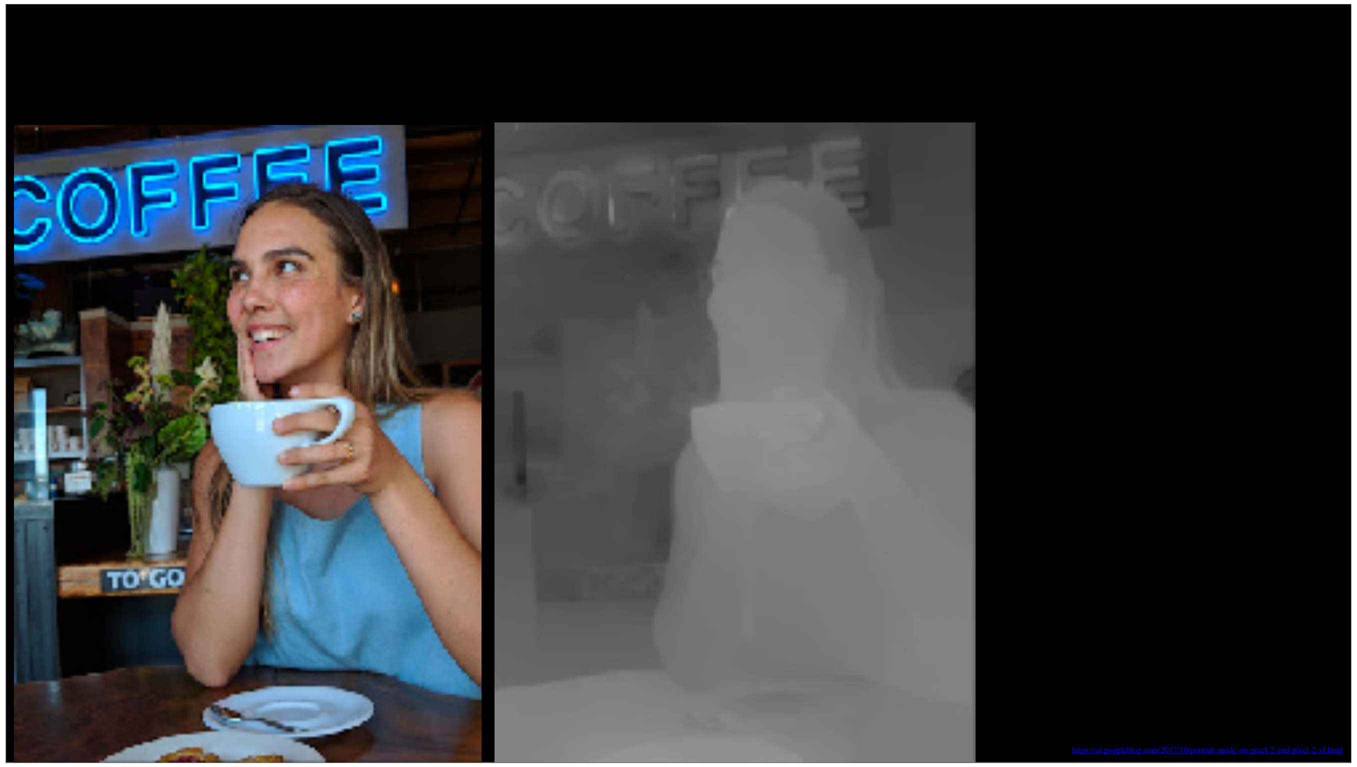
http://www.bobatkins.com/photography/technical/bokeh.html

But we also have to be careful what *kind* of blur we’re applying, since a simple Gaussian blur doesn’t necessarily match the type of blur we see in out of focus images.



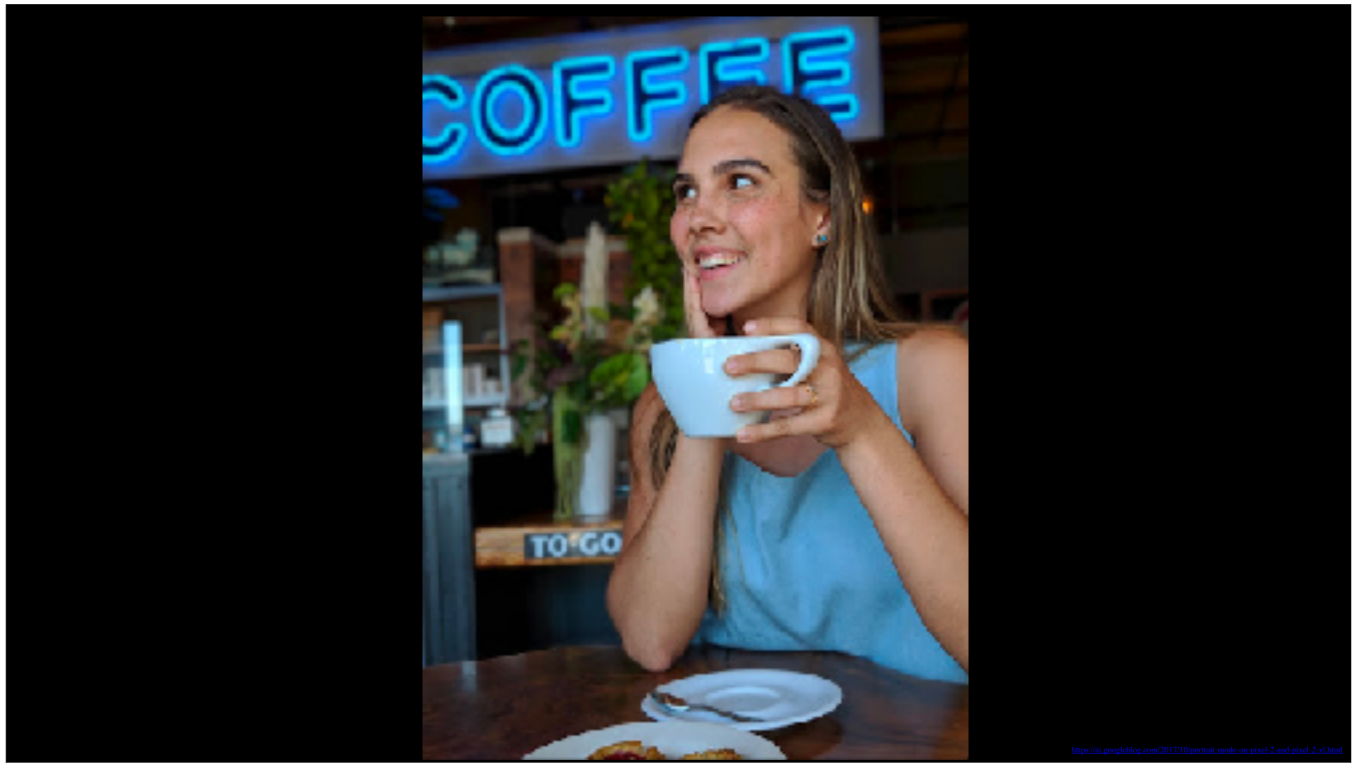
This type of blurry effect is often referred to as Bokeh.

The shape of these fuzzy circles you see are actually dependent on a lot of factors, like the shape of the camera’s aperture, the amount of lens distortion, the 2D position in the image, and so on.



https://ai.googleblog.com/2017/10/portrait-mode-on-pixel-2-and-pixel-2-xl.html

If done correctly…



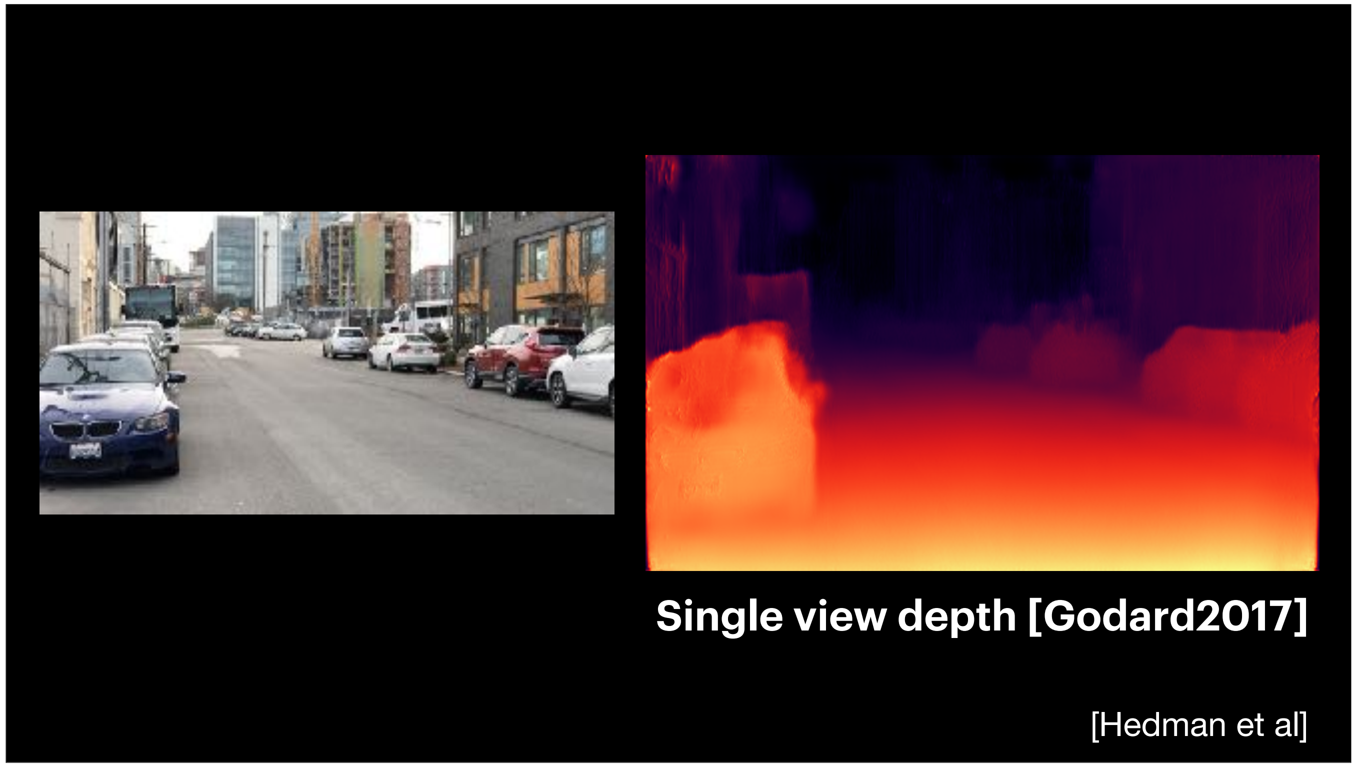
https://ai.googleblog.com/2017/10/portrait-mode-on-pixel-2-and-pixel-2-xl.html

It should be pretty much unnoticeable, and will look something like this, where you can see the background is clearly blurred out, but there are no jarring artifacts [slide]

This is what is being done in a lot of smartphone cameras, labeled as “Portrait mode”. [slide]

The effect isn’t super pronounced in this image, but here’s another example…

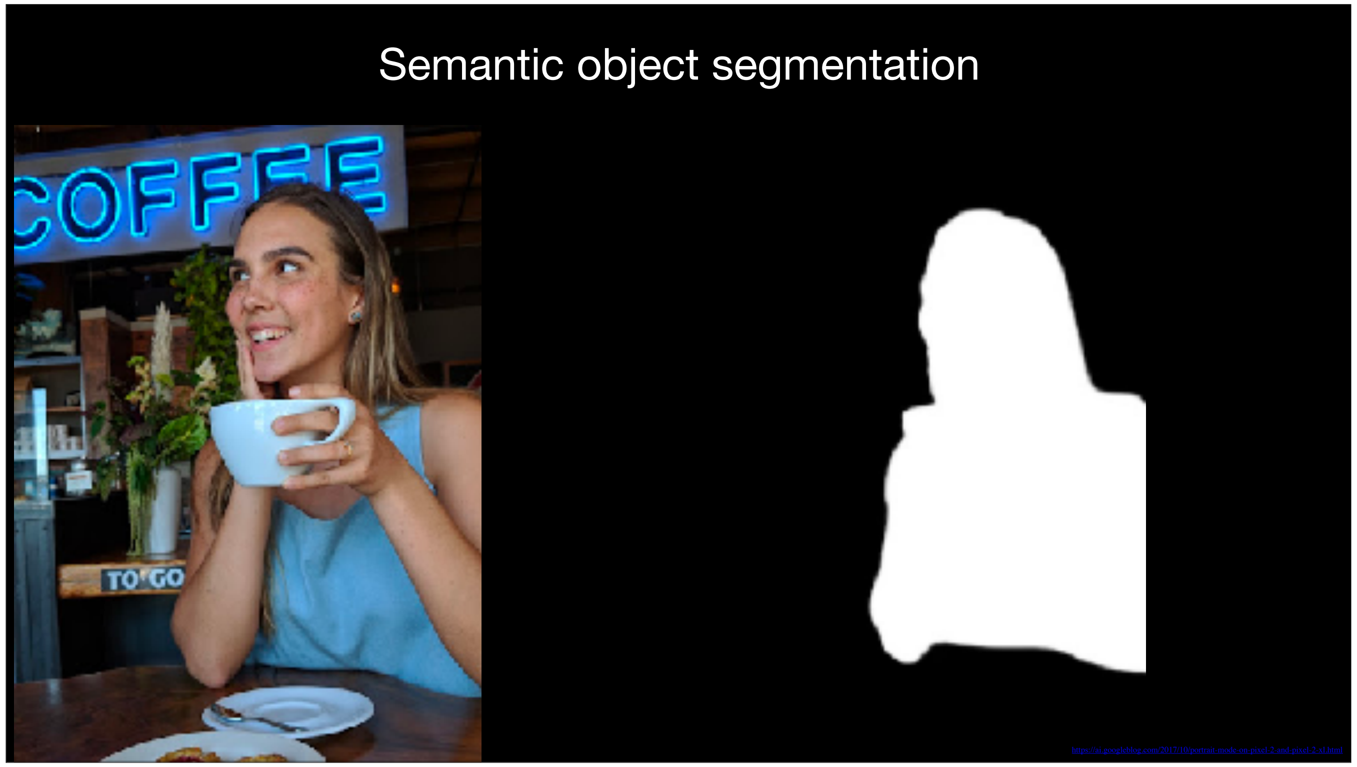
But you may be wondering — how come you can still do Portrait Mode on phones which don’t have multiple cameras? How are we getting the depth maps then?



Single view depth [Godard2017]

[Hedman et al]

They could be using that single-image depth estimation technique we talked about, but in practice, more often they use a simpler and more robust solution.



Semantic object segmentation

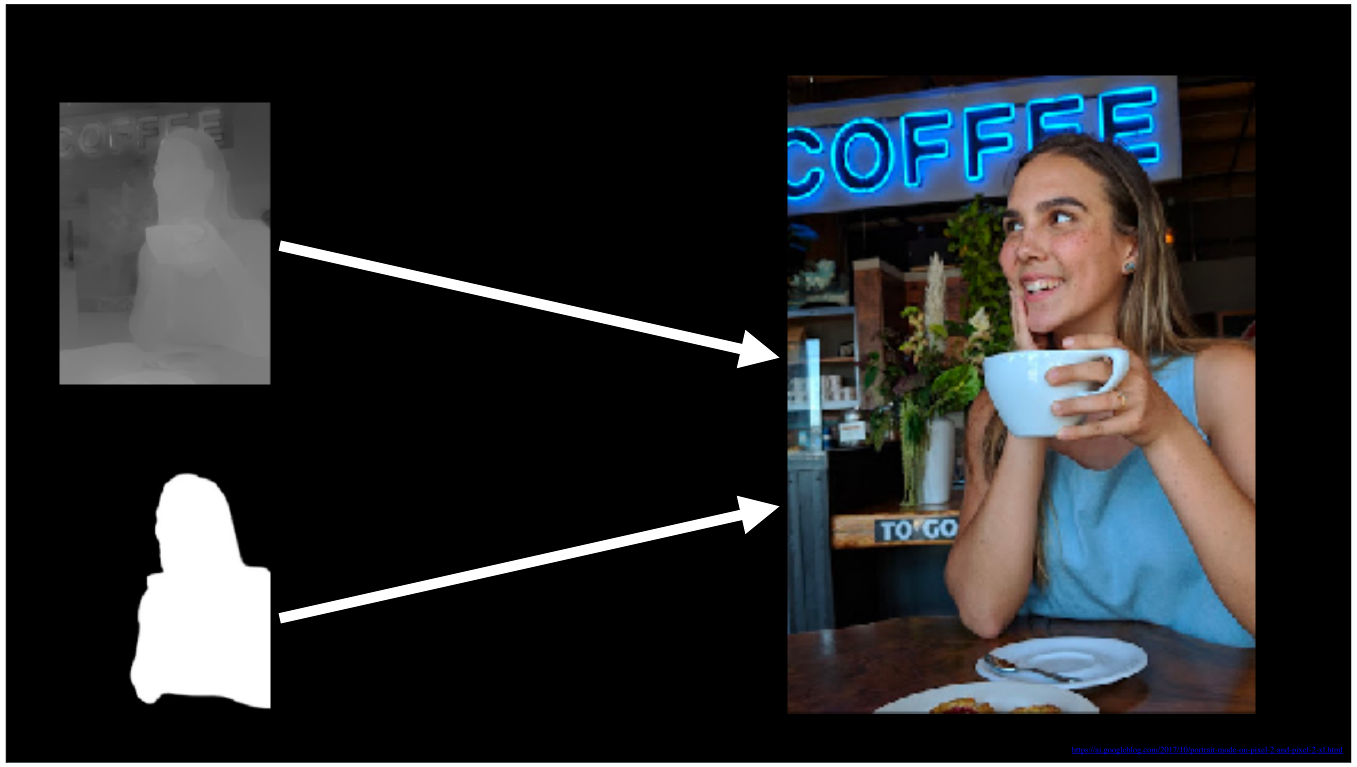
https://ai.googleblog.com/2017/10/portrait-mode-on-pixel-2-and-pixel-2-xl.html

Object segmentation.

We saw in the deep learning lectures that a lot of the new machine learning techniques have gotten really good at labeling objects, and in fact, if we give it a picture like this one, it'll produce a pretty good mask of where we have a person. [slide] [slide]

And since really most of the time, when we take Portrait Mode photos, we're taking them of people or pets, so there are very few classes of objects. So this technique takes advantage of that by segmenting out those objects of interest, and making everything else blurry.

This technique won't result in the correct amount of blur at different depths, but it turns out most people aren't really paying that much attention to that stuff anyways, so for the average consumer, it's okay.



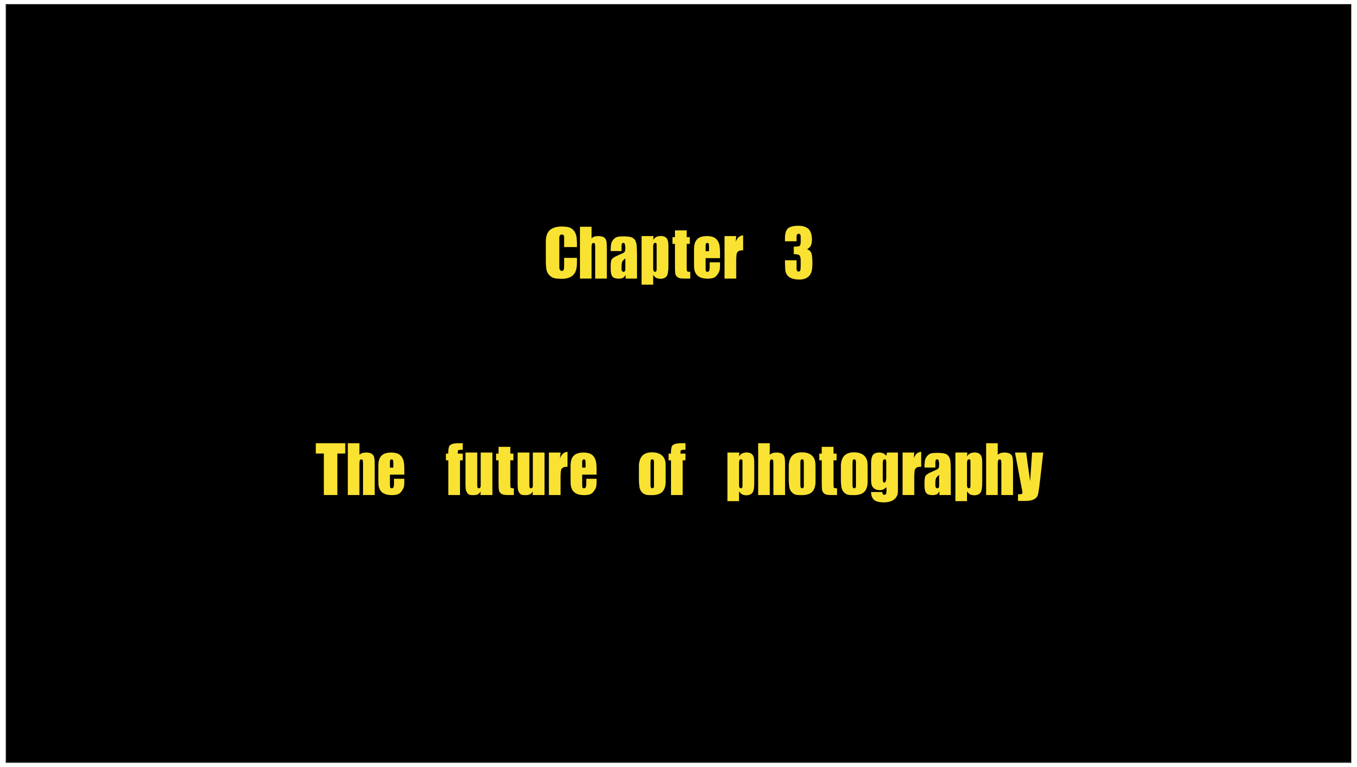
https://ai.googleblog.com/2017/10/portrait-mode-on-pixel-2-and-pixel-2-xl.html

In practice most smartphones use a combination of these two techniques and combine signals from both sources to get a more reliable result.

To recap, we set out to do synthetic depth of field so that we could emulate an effect that we get with more professional cameras, but in the end, we actually ended up accomplishing something which not even professional cameras can do.

Professional cameras, like a DSLR, can adjust the depth of field by changing the focus, or changing the lens, but once you take a picture, it’s locked in forever. In contrast, what we’re doing here is effectively changing the depth of field or focus after the picture has been taken — and that’s not something that could be done before.

This brings me to the next chapter…



Chapter 3

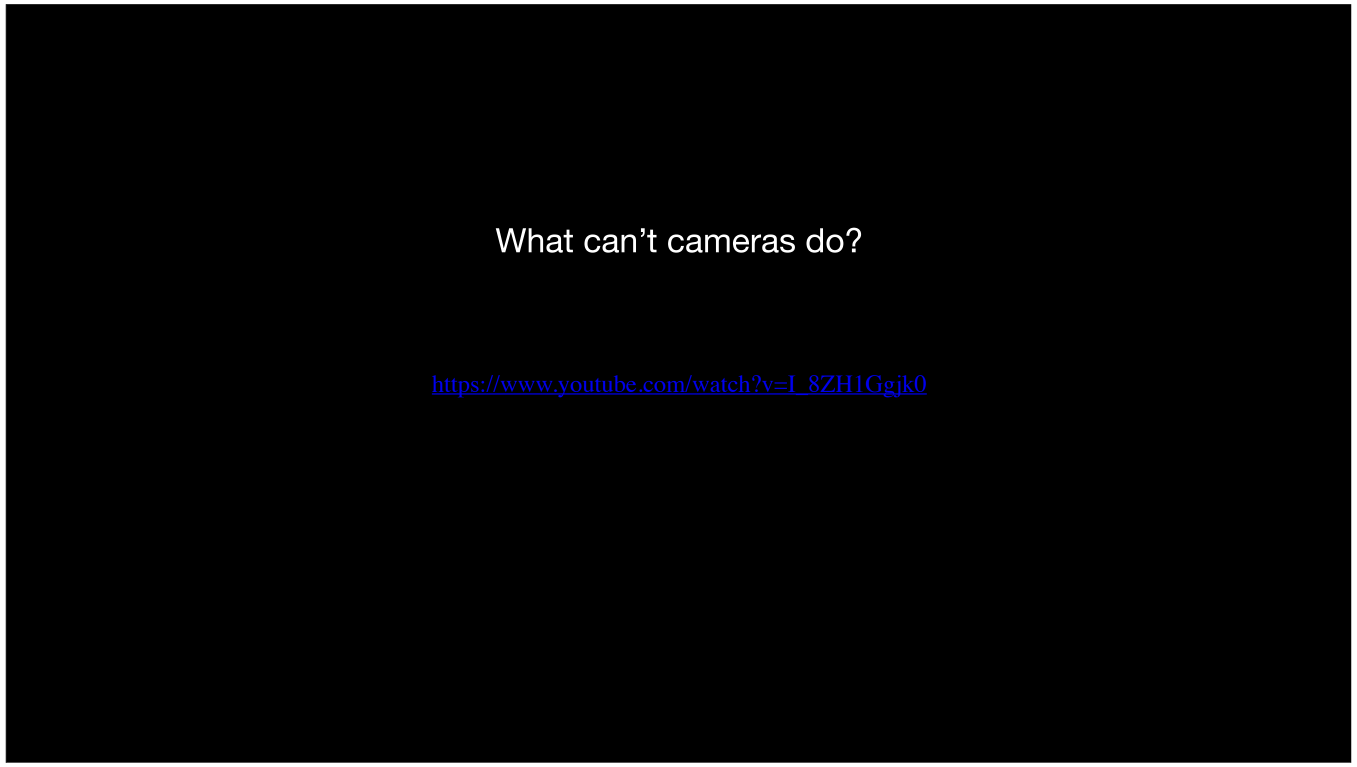
The future of photography

The future of photography.

In the past few examples, we’ve been focusing on using computers to enable us to match the quality of professional photographs using only the cameras in our smartphones.

If we think outside the box, what other things can we do?
Without limiting ourselves to the things which cameras already do, what can we do now that we couldn’t before?

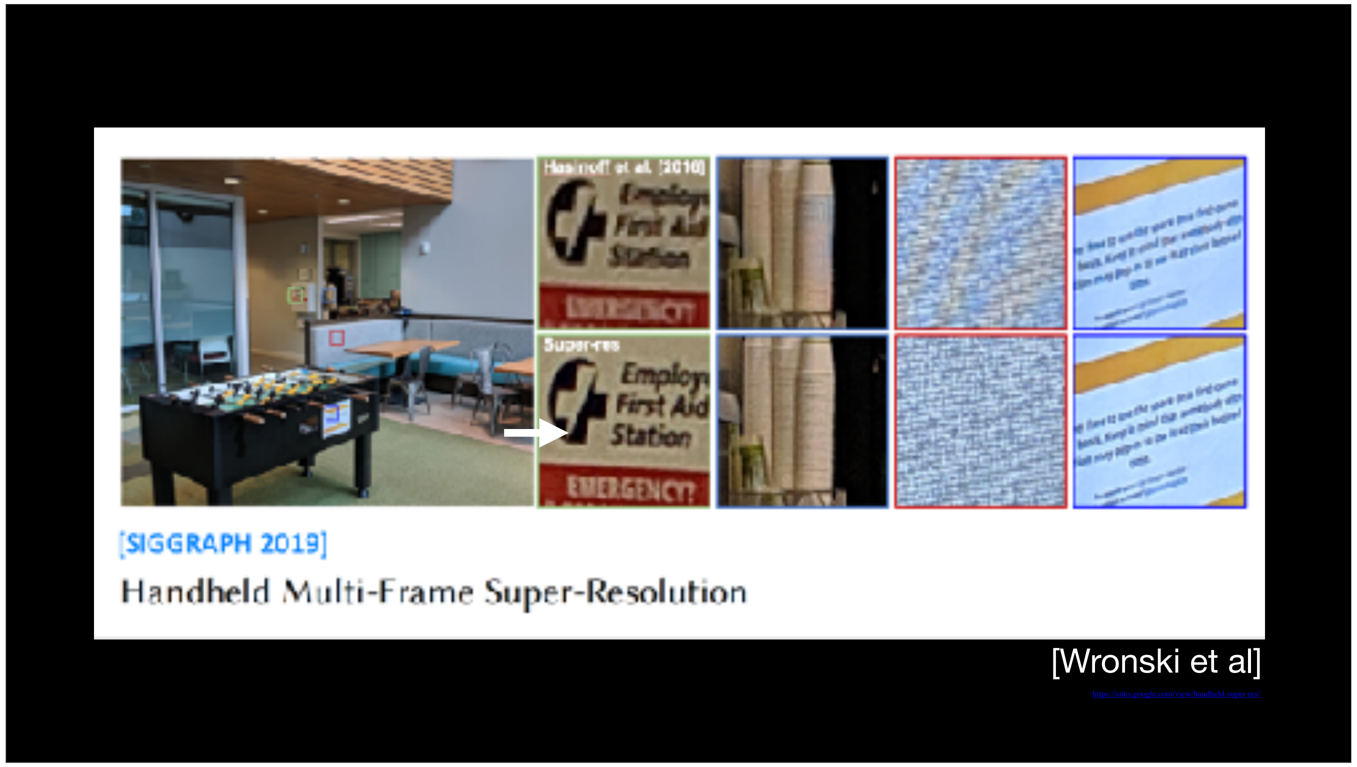
For this section we’re going to briefly touch on a bunch of cool new topics which have emerged over the past few years.



What can’t cameras do?

https://www.youtube.com/watch?v=I_8ZH1Ggjk0

So what can’t cameras do?



[SIGGRAPH 2019]

Handheld Multi-Frame Super-Resolution

[Wronski et al]

https://sites.google.com/view/handheld-super-res/

Actually, that example isn’t too far from the truth anymore.

This paper published last year does a really impressive job super-resolving, effectively “enhancing” images that are zoomed in.

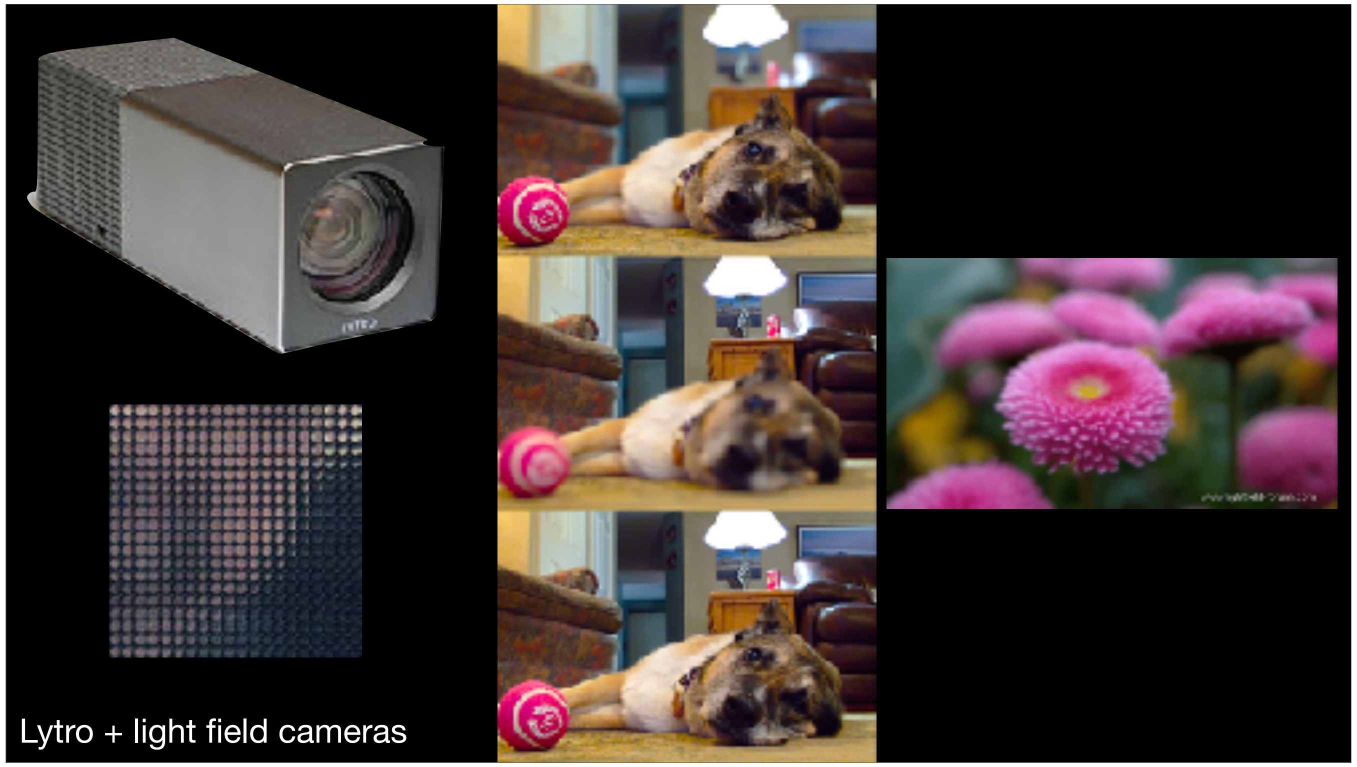
You can see the difference between naively zooming in on the image, and the final image they produce.

And of course, they’re using bursts.

But more seriously… one of the basic limitations that ALL cameras have is that they preserve a single moment in time. [slide]

No matter what happened at that moment, you're stuck with the picture you took. You can go into Photoshop and apply filters or make stylistic changes, but you can't easily go into Photoshop and press a button to change the viewpoint, or change someone's expression, or change someone's makeup — at least not without a lot of effort and time and fancy VFX.

Recently, there have been a lot of cool developments that have taken advantage of some of the concepts we've talked about to help LIFT these restrictions.



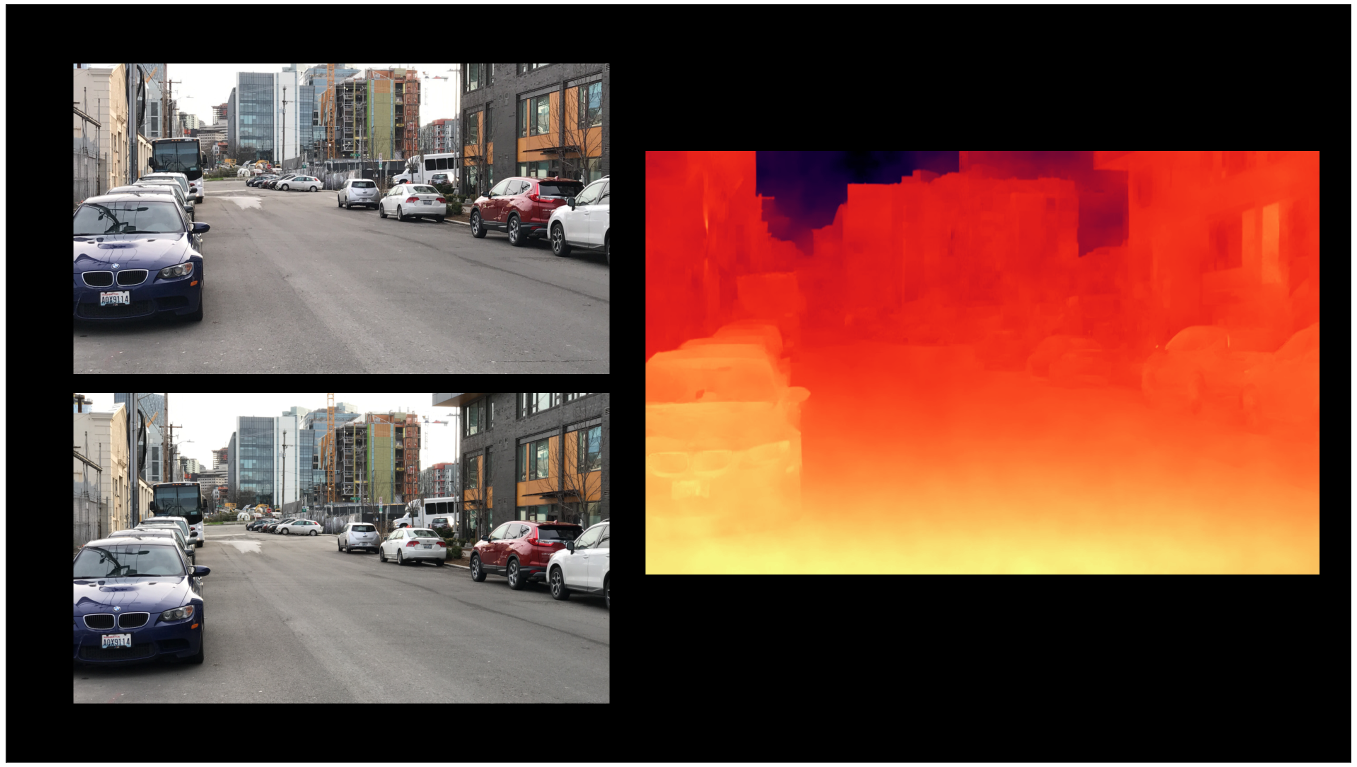
Lytro + light field cameras

Light field cameras are a great example of this — instead of just capturing a single image, light field cameras capture a wide spread of light rays, both in intensity, and in direction. Some of them work basically by breaking up the camera into a bunch of tiny little cameras, like you see at the bottom left.

That lets you do a lot of cool things, like synthetically change the aperture, or refocus the image to a different depth after the picture has been captured.

It can even let you move around the camera viewpoint after you captured the image, like when you're editing it in Photoshop, you can move the camera slightly to the left or right of where you originally took the picture.

There are actually other ways of doing this too,

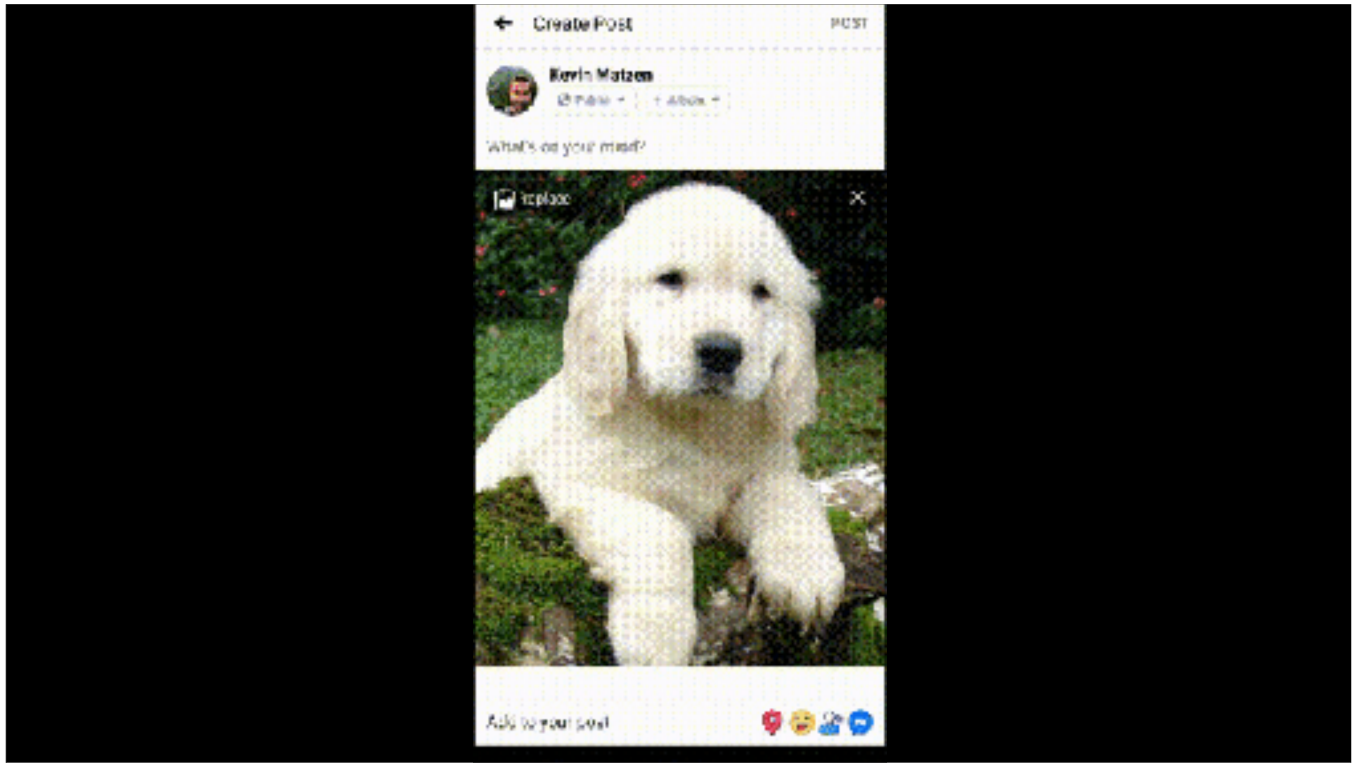


We know we can use micro-baseline or monocular depth estimation to produce a depth map for an image….

…and having depth information is the same as knowing where the objects in the color image are in 3D.

We can take advantage of that fact and and do cool things like this:

…where the picture is still frozen in time, but we can move around the camera to see the same scene from very different viewpoints.



Facebook implemented this and put it in their app, so when you take a photo to post to facebook, you have the option of making it 3D.

And since we're now in 3D, we can also look at these pictures we've captured in VR.

[Hedman et al]

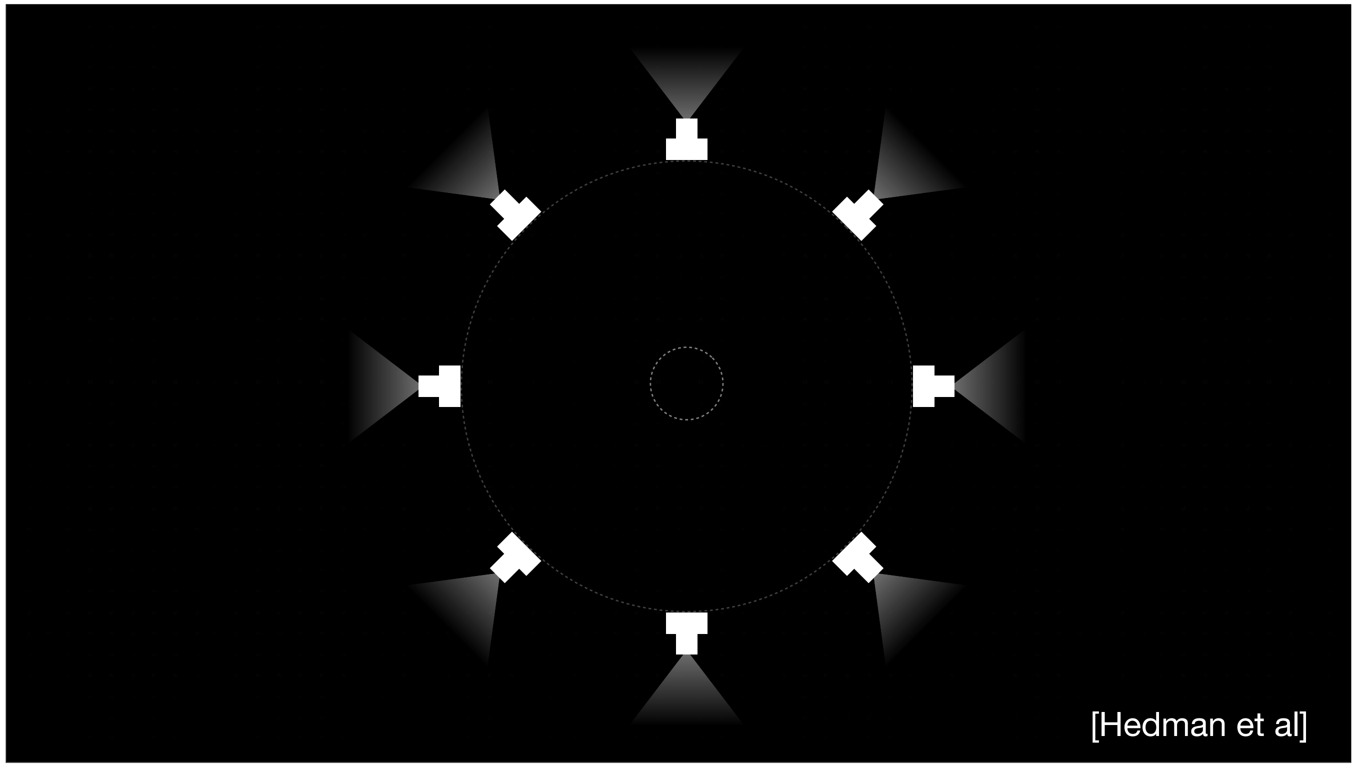
Which lets us basically teleport to new places, or share our favorite places with our friends…

This last example is actually more than just a single photo.

It’s a 3D panorama, which is effectively a whole 3D scene.

[Hedman et al]

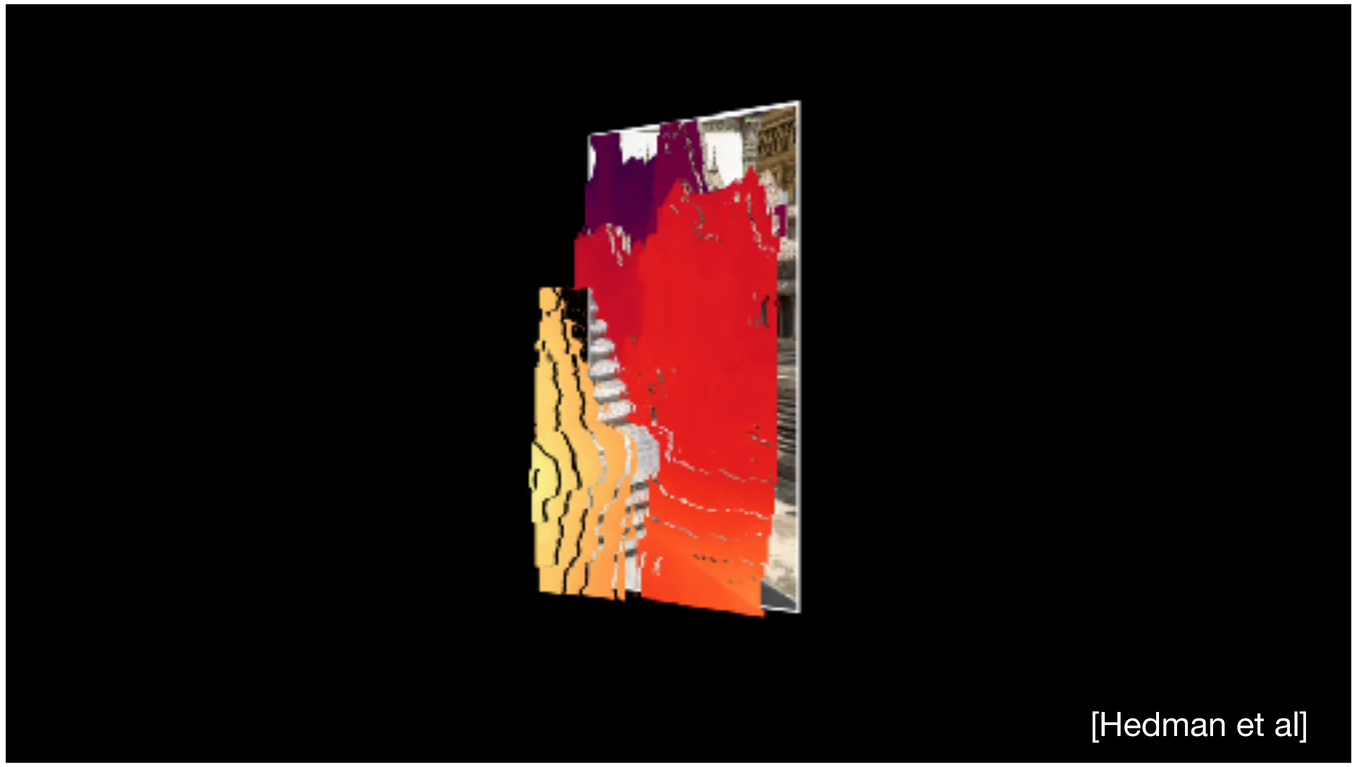
This work from Facebook takes advantage of the fact that when you're capturing a 360 panorama, and you are turning around in a circle, you're seldom going to be pivoting at exactly a single point, but rather



You’ll be capturing pictures along a ring.

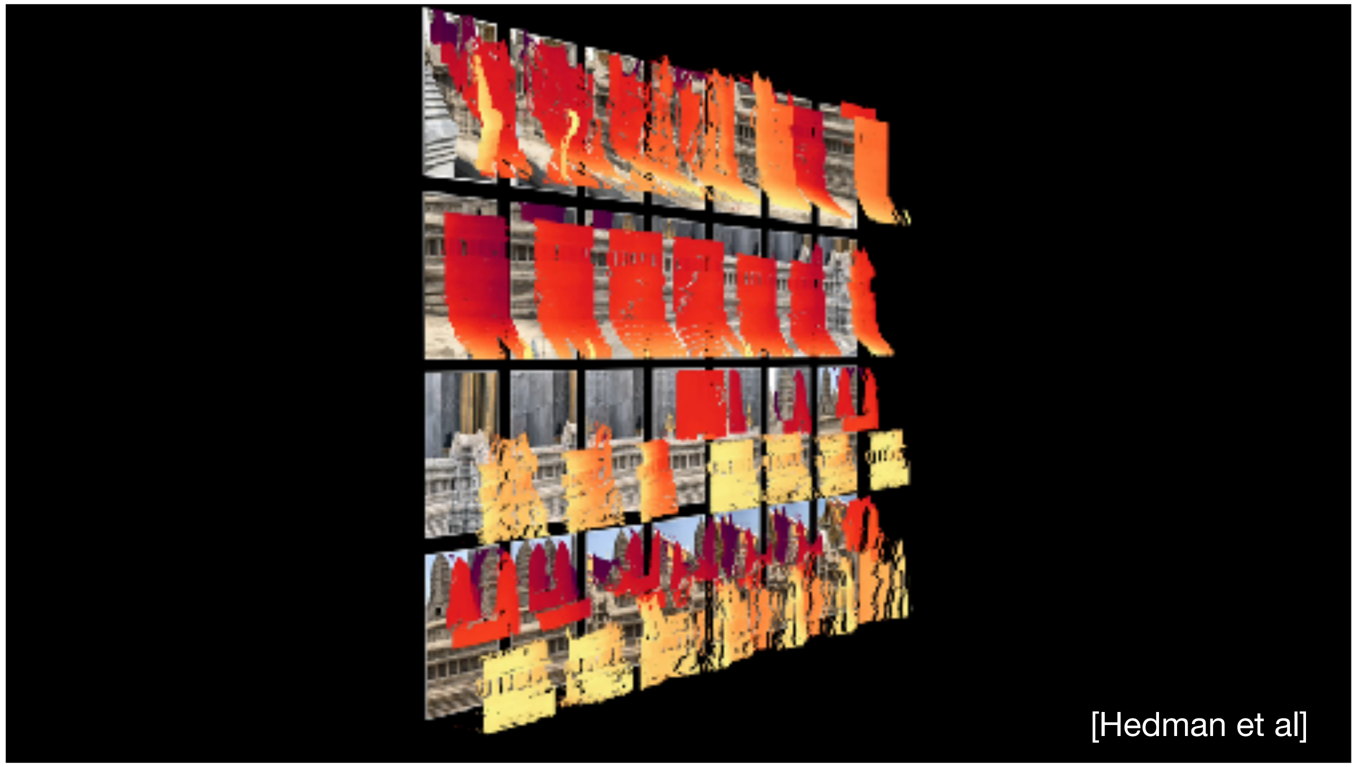
And since these pictures have relative translational parallax, we can do stereo depth estimation between them.

And on top of that, we can use the built-in dual-camera mode as another source of depth information.



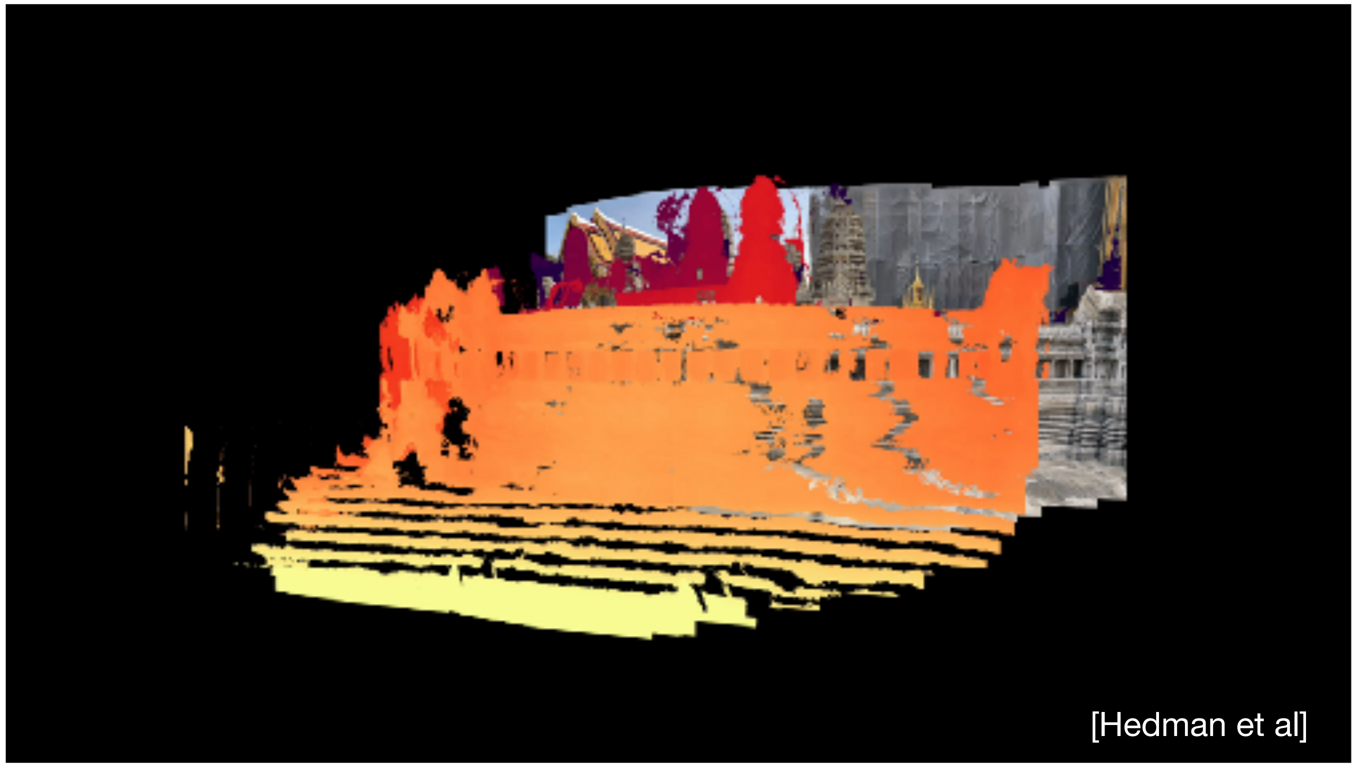
[Hedman et al]

If you capture a burst of these photos, like you would with a panorama, each with its corresponding depth map



[Hedman et al]

You can then stitch them together based on a lot of the techniques we’ve learned in combining images and panoramas, into two panoramas, one for depth and one for color.



[Hedman et al]

…and the resulting panoramas can be turned into a 3D mesh which we can view in VR and move around and explore.

[Hedman et al]

Here’s another visualization of the captured images, and how they fit together into two panoramas, one for color and one for depth.

When we render these 3D scenes, we can get really cool looking results. With depth, normals, parallax.

The illusion breaks if you move really far away, but 3D photos look great when staying close to the original views.

And since there's explicit 3D geometry, you can interact with the scene in ways you can't with regular photos.

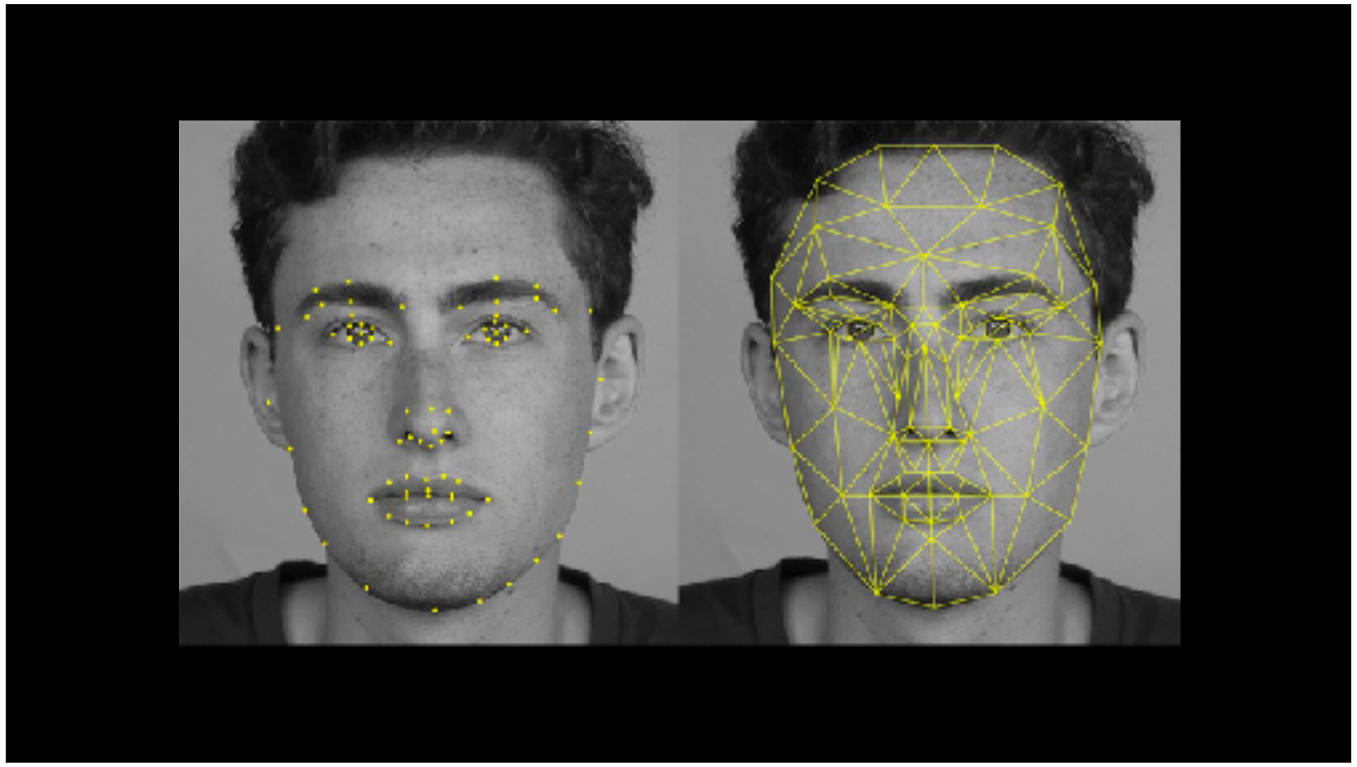
So you can fill the scene with water, or play around with lighting.

So that's an example of how we can change viewpoint — but what about other things…

…like changing our faces?

I’m sure most people know about face filters…

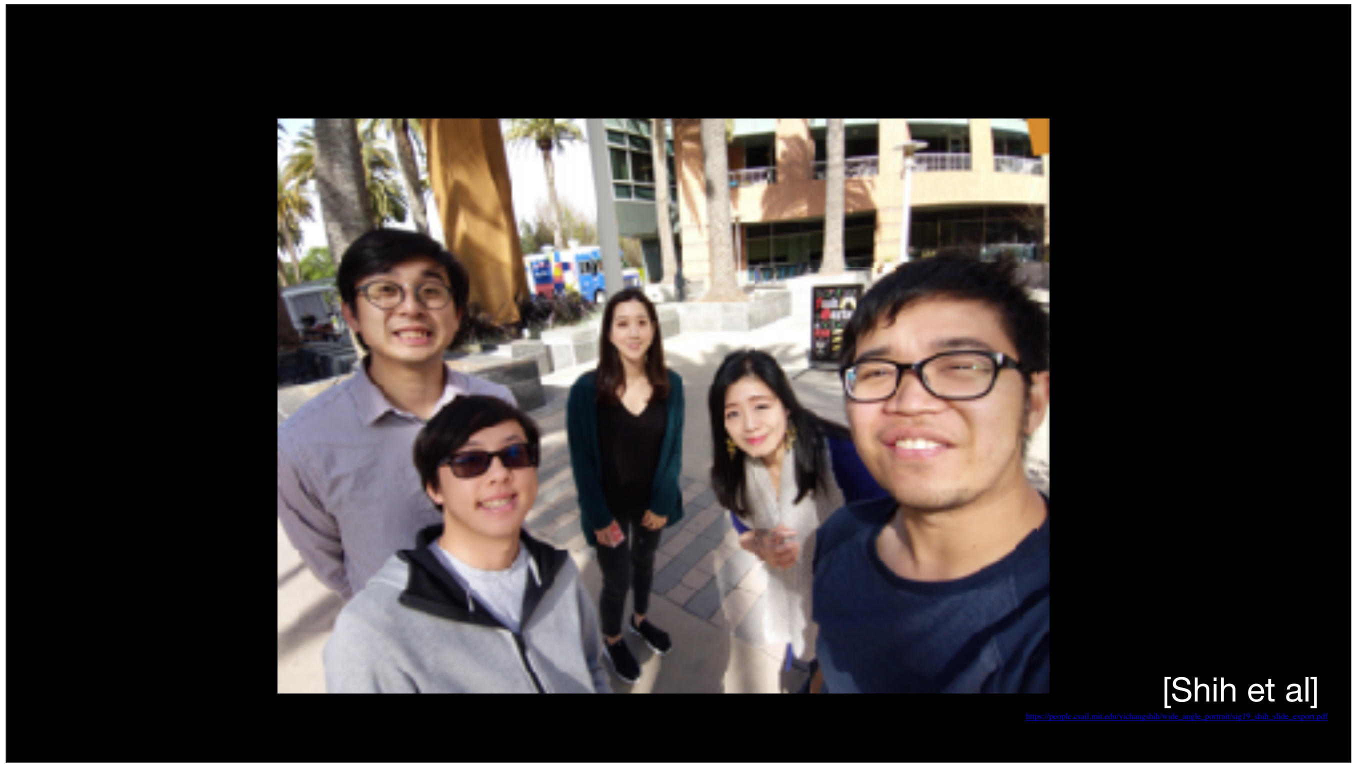
You can take a picture or video, and it’ll apply some kind of transformation, either aesthetic, or some kind of warp to your face.



It does this by tracking some feature points on your face.

https://technabob.com/blog/wp-content/uploads/2010/02/cosmetic-mirror-by-shiseido.jpg

This can let you do simple things like digitally try on makeup or glasses.



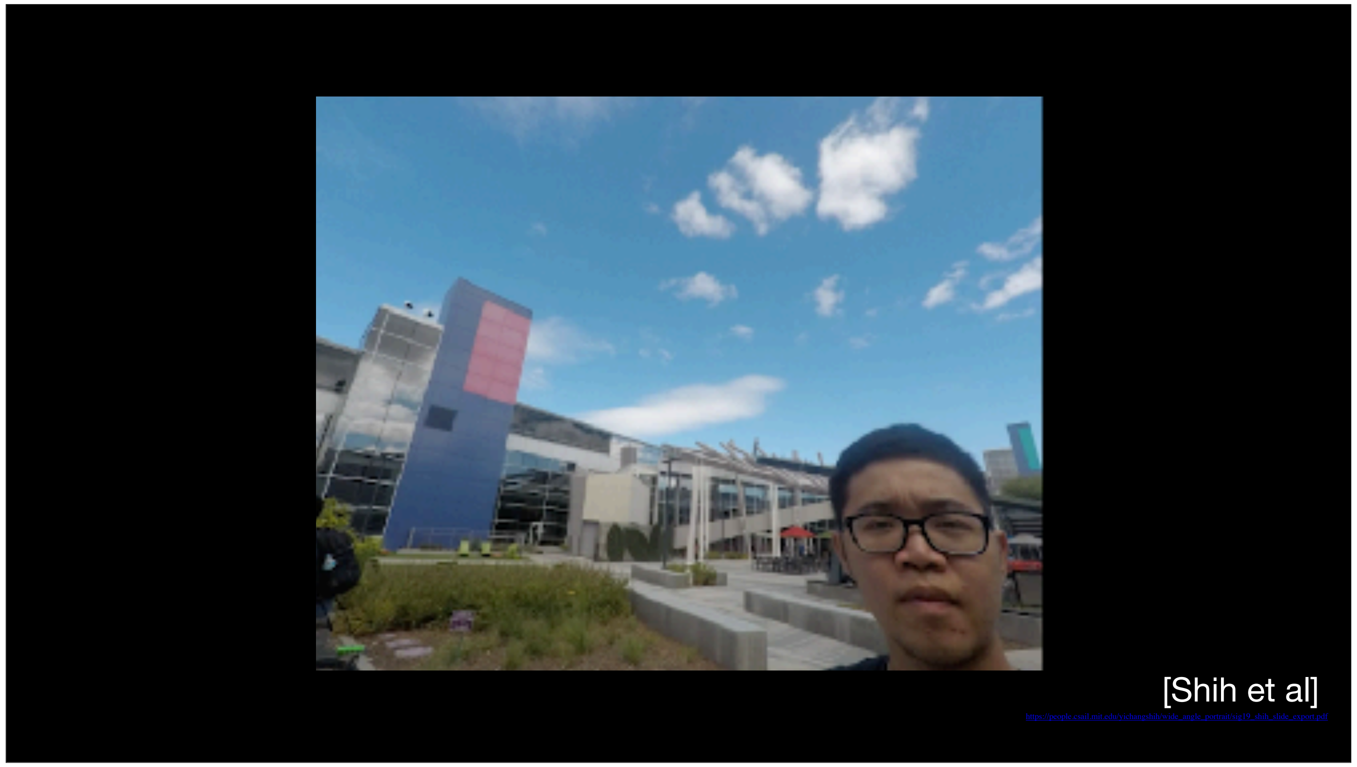
[Shih et al]

https://people.csail.mit.edu/yichangshih/wide_angle_portrait/sig19_shih_slide_export.pdf

But you can also do more complicated things, like in this paper from Google.

Let’s say you want to take a selfie with friends, but the field of view (the angle that the selfie camera can capture) is pretty limited.

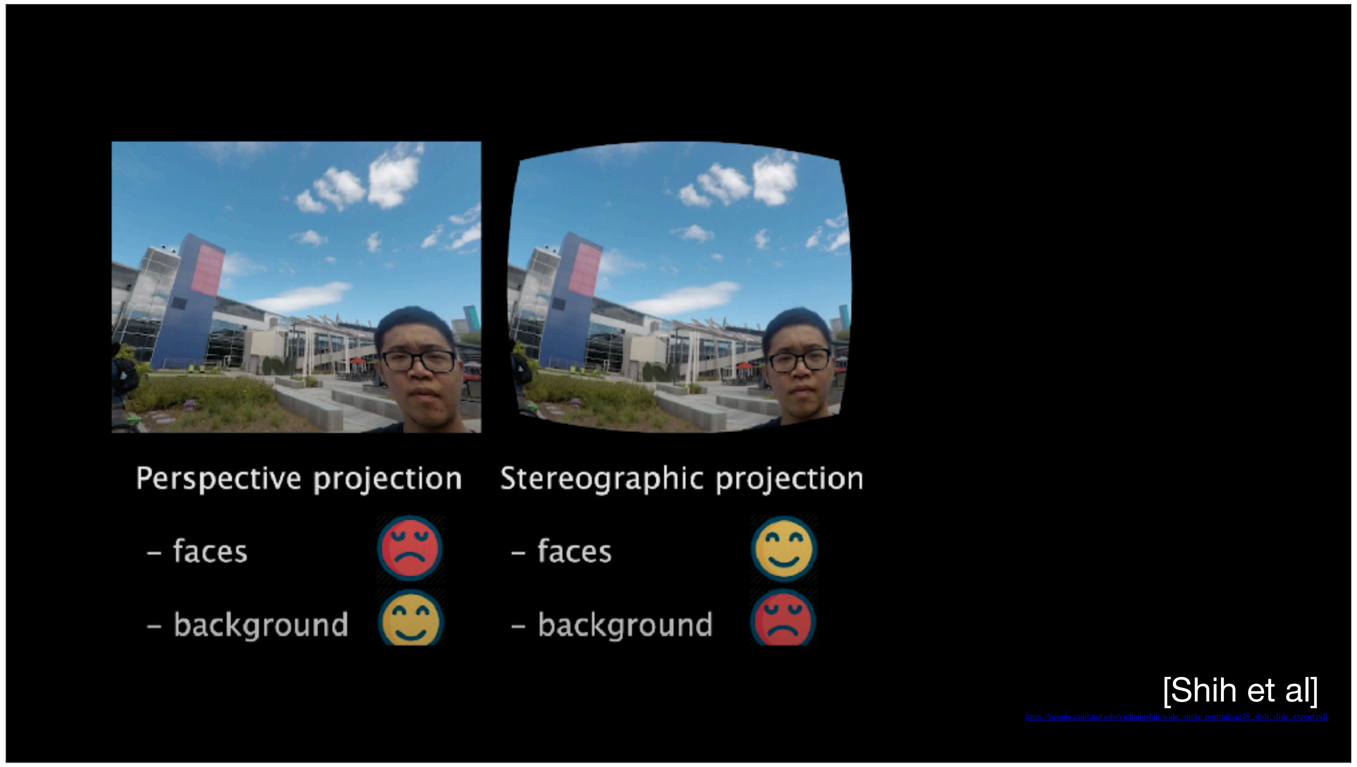
The natural solution is to increase the field of view (get a wider angle lens) [slide], but that results in a bunch of unpleasant effects, like…



[Shih et al]

https://people.csail.mit.edu/yichangshih/wide_angle_portrait/sig19_shih_slide_export.pdf

…really unpleasant perspective distortions on the face.



Perspective projection

- faces 🙁
- background 🙂

Stereographic projection

- faces 🙂
- background 🙁

[Shih et al]

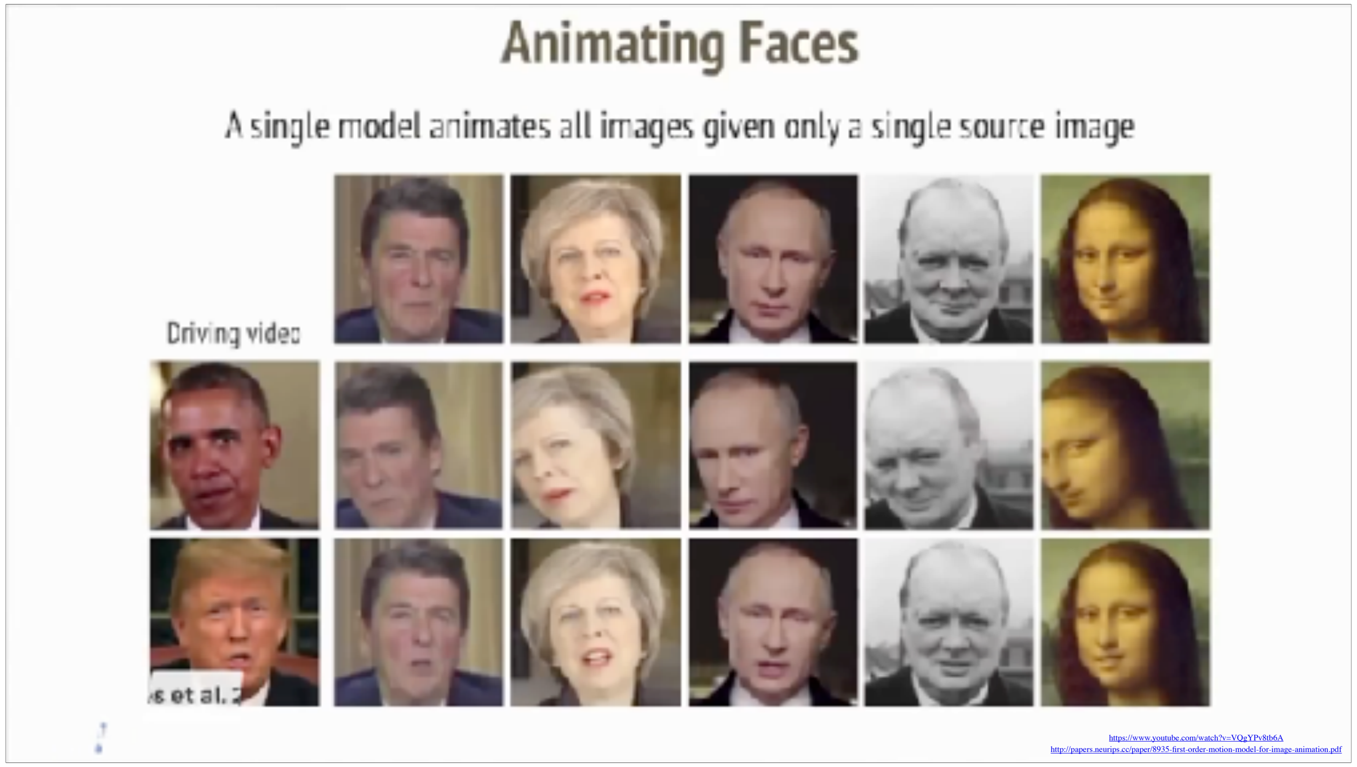
https://people.csail.mit.edu/yichangshih/wide_angle_portrait/sig19_shih_slide_export.pdf

There are other types of projections for wide-field of view images, which will make our faces look more normal, but that'll result in the background looking really warped, like buildings will be crooked, and things that are supposed to have straight lines will look more like curves.

This paper had a pretty cool solution…[slide]
They used this same type of face tracking to find the faces in the image, so they can warp those faces differently than the background, so we get the best of both worlds.

[Hedman et al]

But how about expressions - can we change expressions?



Animating Faces

A single model animates all images given only a single source image

https://www.youtube.com/watch?v=VQgYPv8tb6A
http://papers.neurips.cc/paper/8935-first-order-motion-model-for-image-animation.pdf

It turns out deep learning has gotten pretty good at that too.

There are techniques to do facial puppetry, which manipulate the expression of a person in a picture. All they need is a picture of the face that they want to manipulate, and they can change the expression based off of any other video of someone else. They can even drive it based off of an audio clip of someone else speaking.

Some people have found this to be a bit scary — there’s a good chance you’ve heard of this technology referred to in the media as deep-faking.

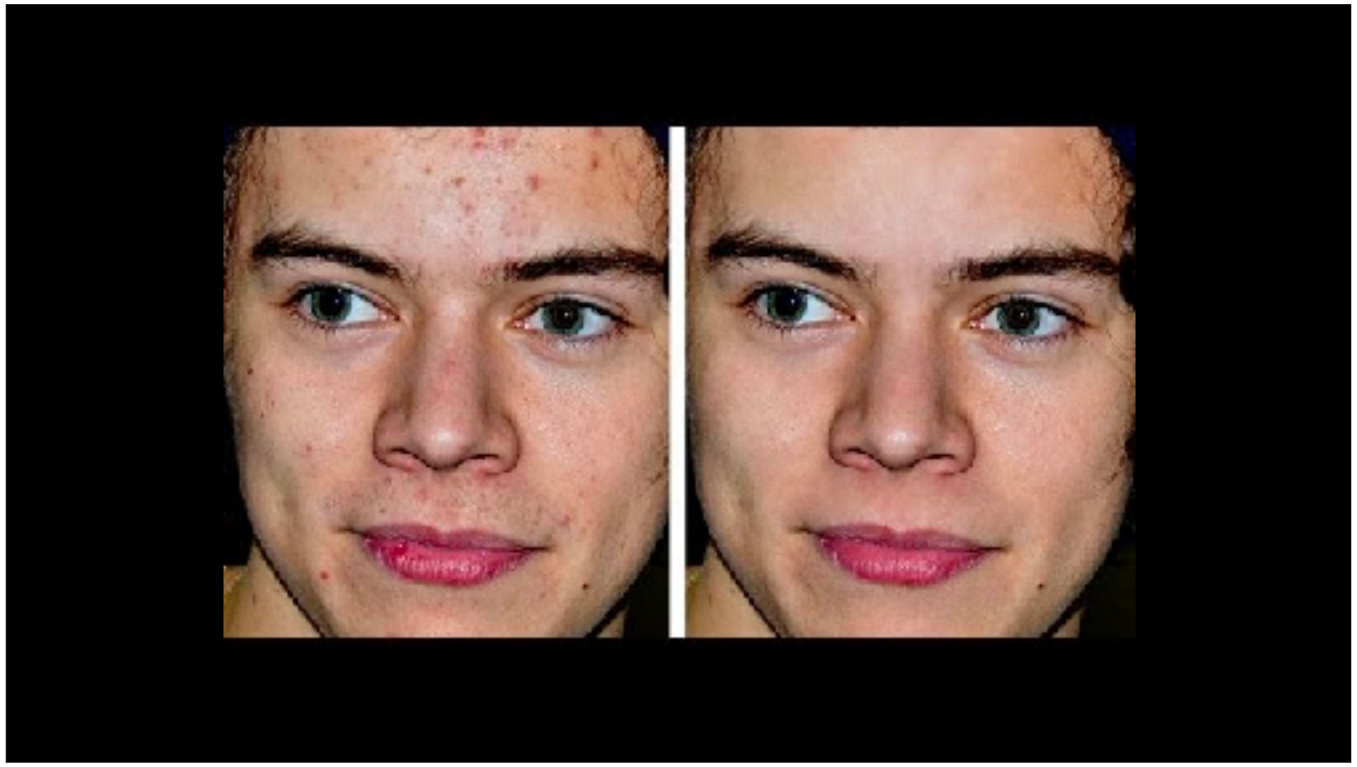
But where do we draw the line?

What makes a good photograph?

What is a good photograph, anyways?

is it the one where things look the best? Or is it the one that most accurately represents reality?

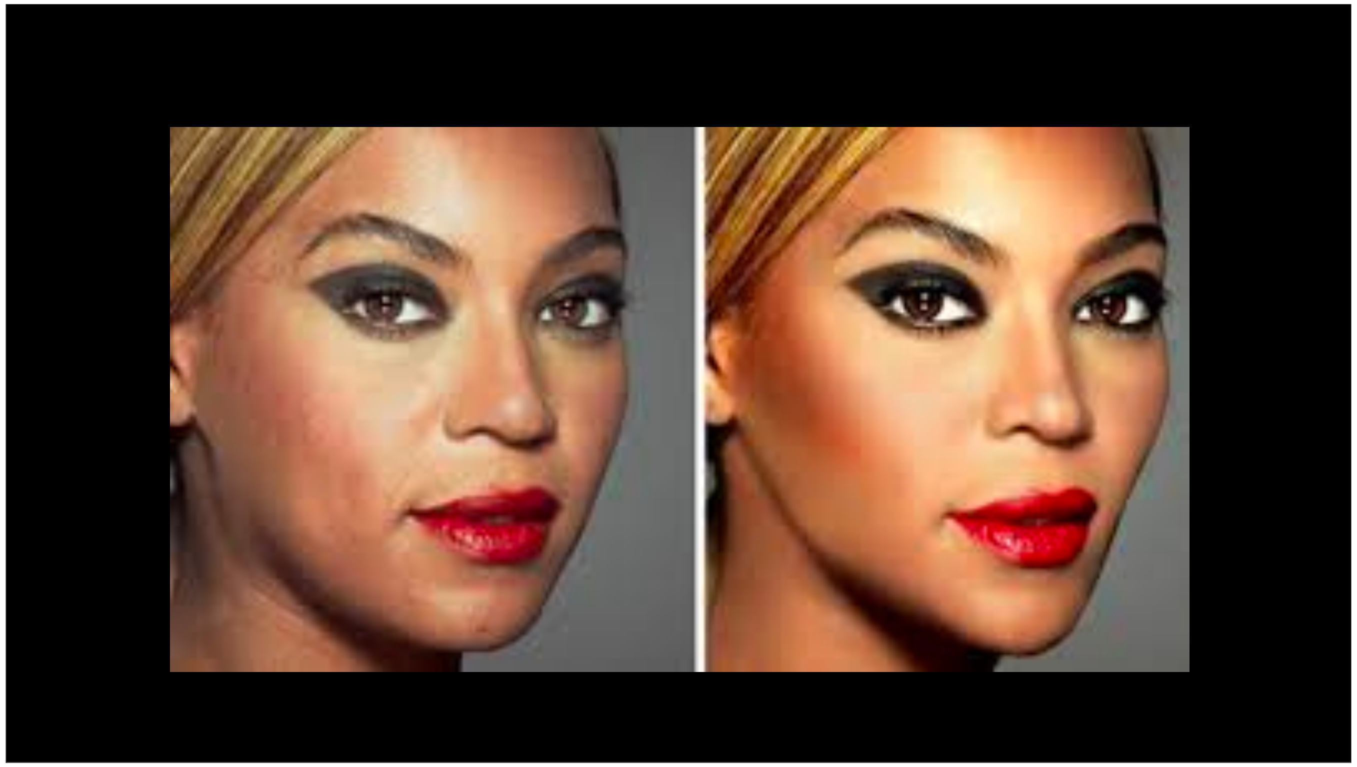
It’s a tricky question, and a lot of the popular face filters and skin-retouching algorithms also fall into this category.



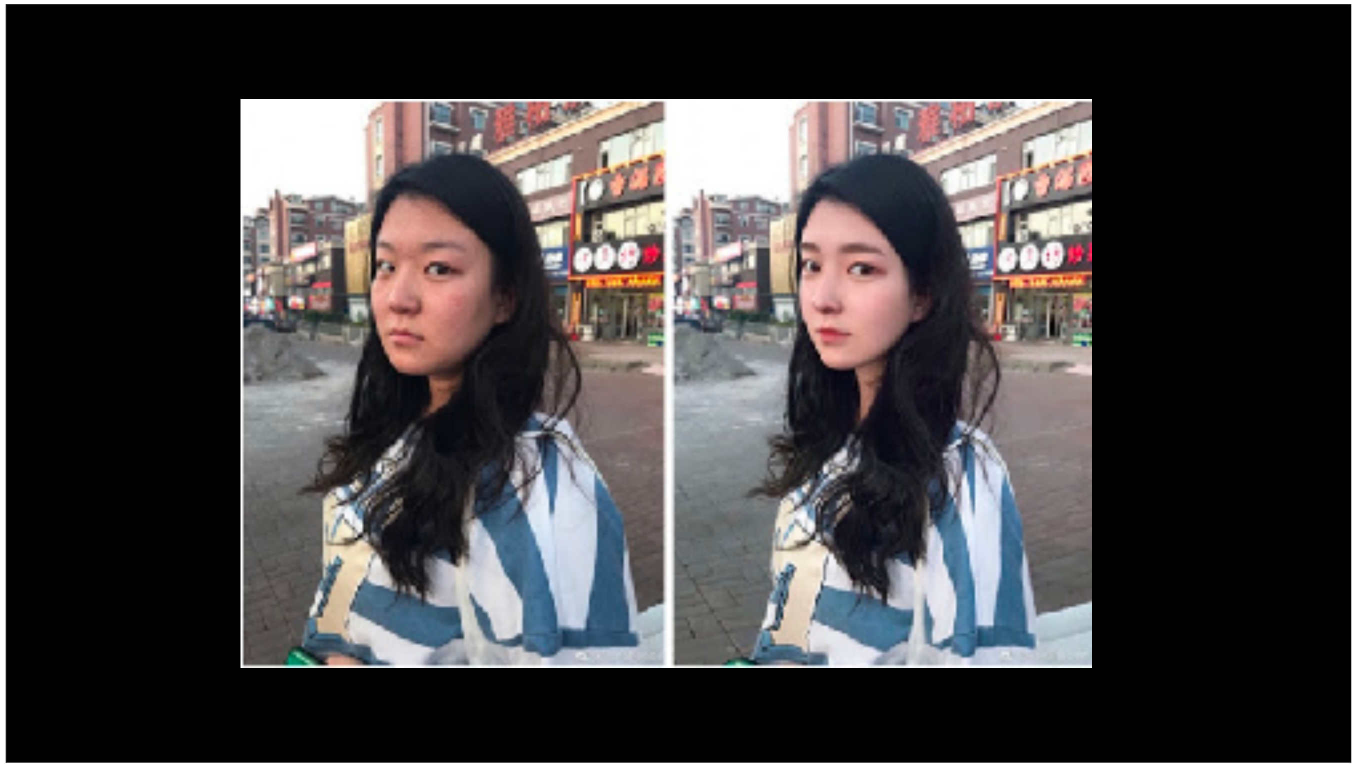
Deep learning algorithms, and even more classical edge-preserving filters do a good job of cleaning up blemishes on skin.

These are way more widely used than a lot of people realize — in fact, most photos which we see posted by models and famous people are usually edited in some way.

How do we know that we’ve gone too far?



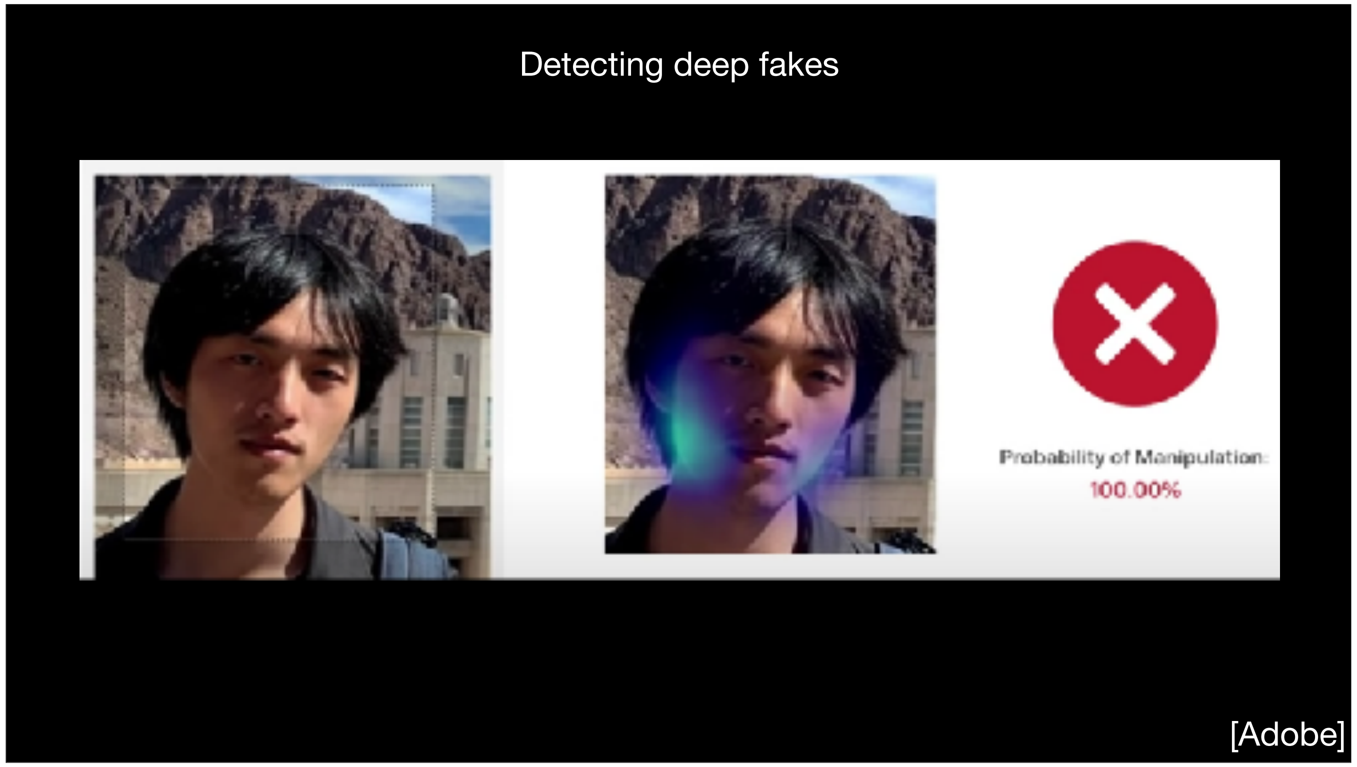
At what point does this image stop being an image of the same person?



Or does it even matter at all?

These are open questions.

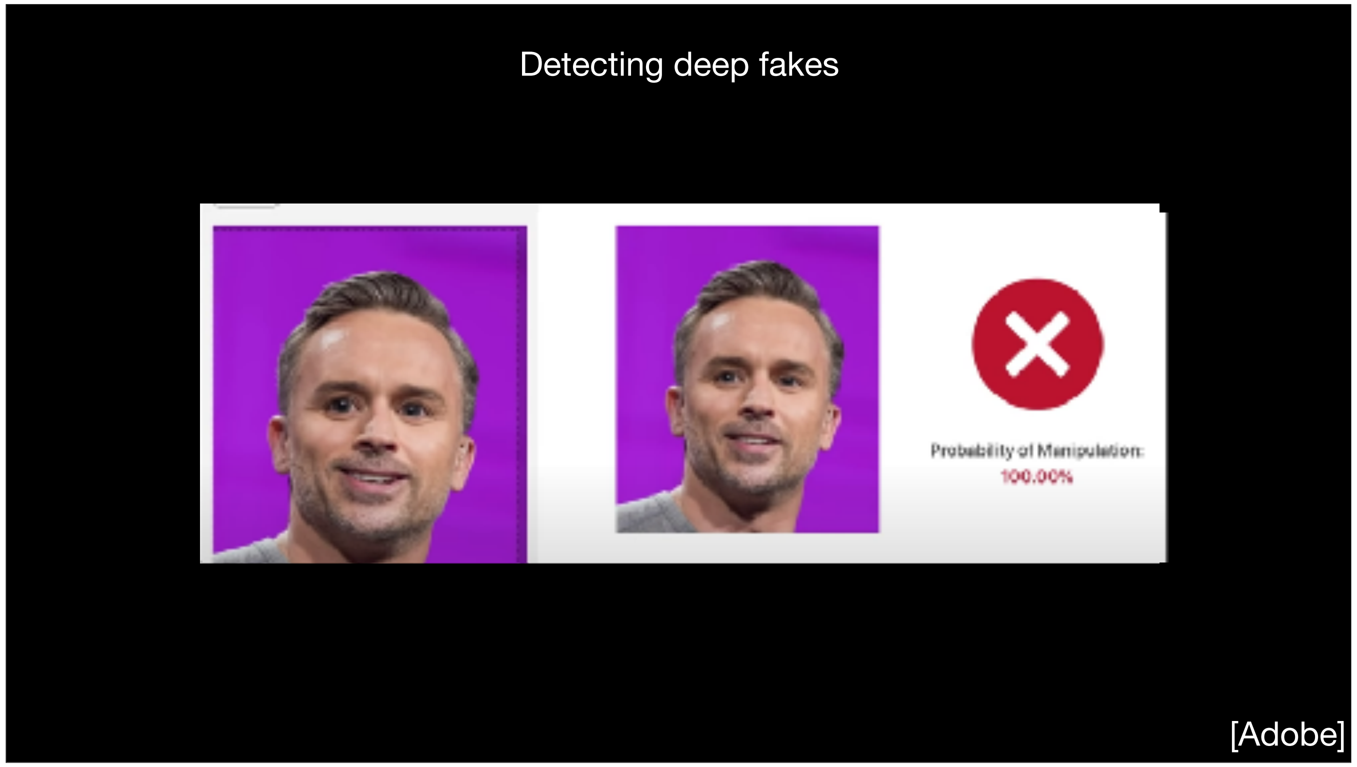
But there has been quite a bit of work put into detecting altered photographs.



Detecting deep fakes

[Adobe]

Last year, Adobe showed off an application which would look at images, and could tell you if it had been modified in any way.



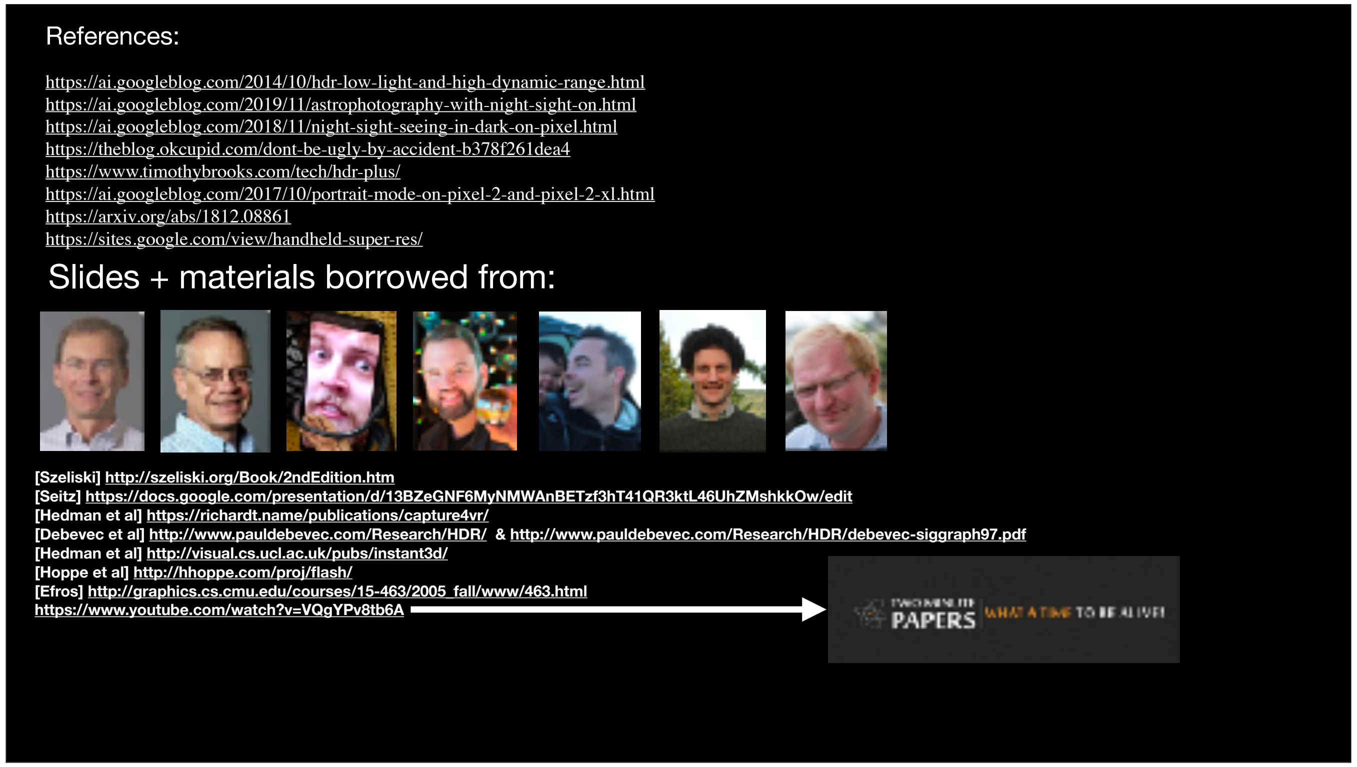
Detecting deep fakes

[Adobe]

On top of that, it would also tell you where in the image a modification was made.

And could even undo that suspected modification to try to recreate the original image.

It does this by looking at local image statistics — since most editing operations, or deep-faking methods, even if they look super realistic to us, often have some pixel-level statistics which give them away as being fake.



References:

https://ai.googleblog.com/2014/10/hdr-low-light-and-high-dynamic-range.html
https://ai.googleblog.com/2019/11/astrophotography-with-night-sight-on.html
https://ai.googleblog.com/2018/11/night-sight-seeing-in-dark-on-pixel.html
https://theblog.okcupid.com/dont-be-ugly-by-accident-b378f261dea4
https://www.timothybrooks.com/tech/hdr-plus/
https://ai.googleblog.com/2017/10/portrait-mode-on-pixel-2-and-pixel-2-xl.html
https://arxiv.org/abs/1812.08861
https://sites.google.com/view/handheld-super-res/

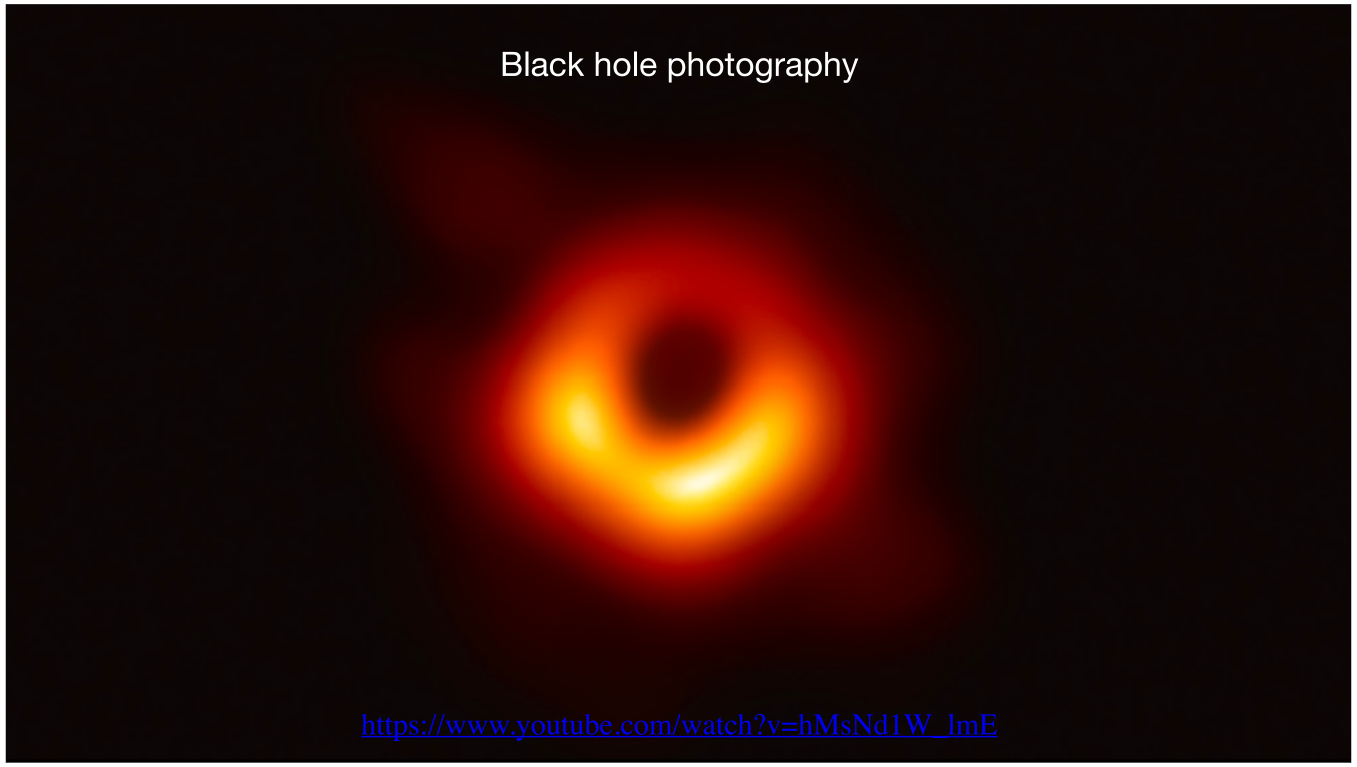
Slides + materials borrowed from:

[Szeliski] http://szeliski.org/Book/2ndEdition.htm
[Seitz] https://docs.google.com/presentation/d/13BZeGNF6MyNMWAnBETzf3hT41QR3ktL46UhZMshkkOw/edit
[Hedman et al] https://richardt.name/publications/capture4vr/
[Debevec et al] http://www.pauldebevec.com/Research/HDR/ & http://www.pauldebevec.com/Research/HDR/debevec-siggraph97.pdf
[Hedman et al] http://visual.cs.ucl.ac.uk/pubs/instant3d/
[Hoppe et al] http://hhoppe.com/proj/flash/
[Efros] http://graphics.cs.cmu.edu/courses/15-463/2005_fall/www/463.html
https://www.youtube.com/watch?v=VQgYPv8tb6A

Check out the Google AI blogs, they’re really well written.

Thanks to all the people shown here for their wonderful materials and slides.

This YouTube channel also served as great inspiration for some of these topics. They make short 2-minute videos about new cool papers in the field of graphics, AI and computer vision.



Black hole photography

https://www.youtube.com/watch?v=hMsNd1W_lmE

We can end on celebrating one of the pivotal achievements of computational photography over the past few years, when the entire globe was made into an image sensor, so we could capture a picture of a black hole millions of light-years away.

Let’s enhance

https://www.youtube.com/watch?v=Vxq9yj2pVWk