An Artist-Friendly Workflow for Panoramic HDRI

Sébastien Lagarde
Sébastien Lachambre
Cyril Jover

Hello,

My name is Sébastien Lagarde, I am a graphics programmer at Unity and with my two artist co-workers Sébastien Lachambre and Cyril Jover,
we have tried to setup an easy method to capture accurate panoramic High Dynamic Range Images (HDRIs).

Introduction

- Artist-oriented talk
- HDRI as a light source
  - Requires accurate HDRI

- Full range (no clipping)
- Minimal lens artifacts
- Linear
- No artistic effects

This talk is oriented toward artists.

We want to use HDRI as a light source to easily reproduce lighting from the real world and to better integrate a CG object inside an environment, a process that's widely used today

Such HDRI needs to be accurate:
- It must capture the full range of lighting in the scene, without clipping
- It must have minimal lens artifacts
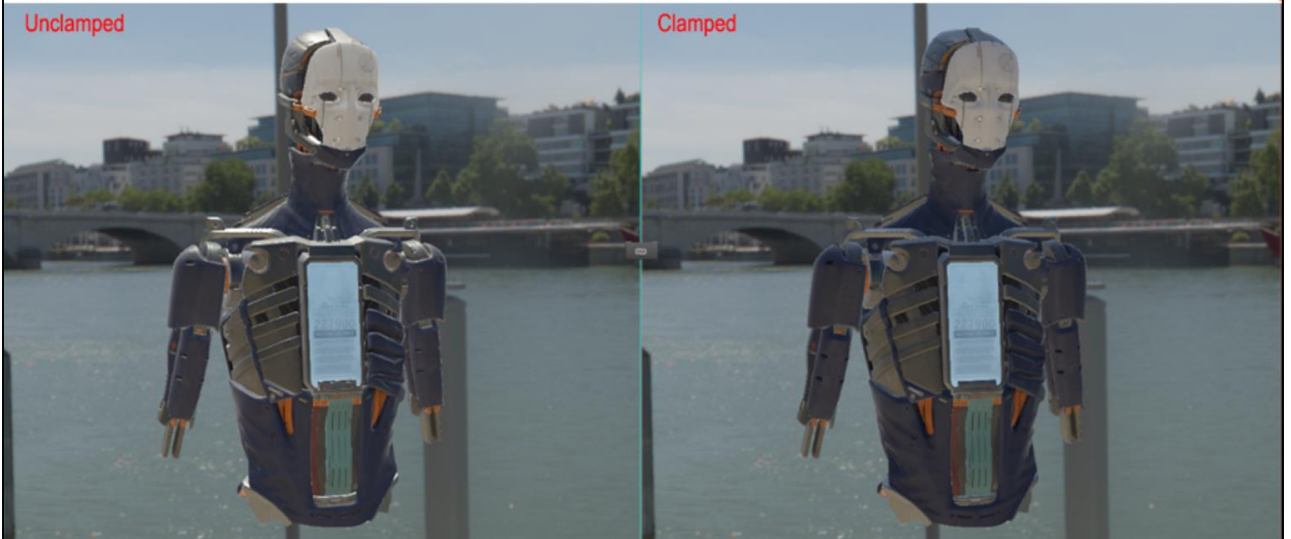- It must be linear
- It shouldn't include any artistic effects

When talking about full range HDRI, we mean this.
Left is unclamped HDRI. Right is clamped one.
When changing the exposure we can see that the sun disc is slightly less intense on the clamped one.

But this small difference has a huge impact on the lighting
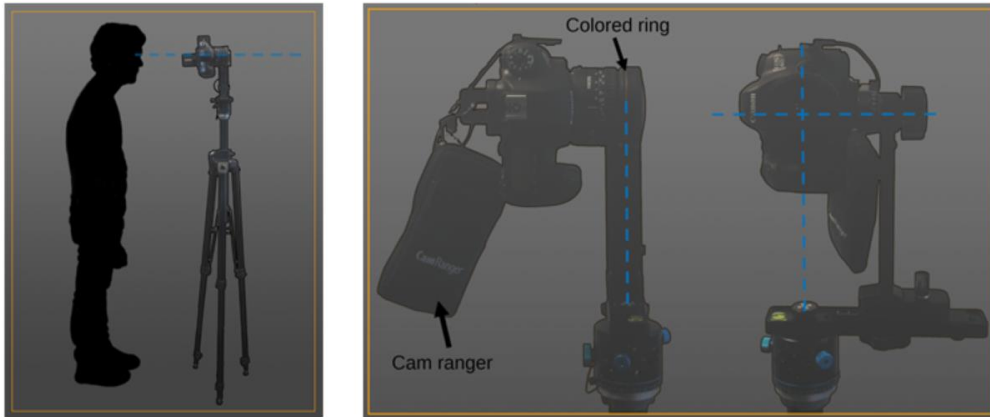
Equipment and software

This picture shows the equipment and the software we use at Unity.
We have tried to make our equipment setup affordable and have avoided using any custom devices to ensure our results can be reproduced.
We use a Canon 6D with a Sigma fisheye lens, a Manfrotto tripod with Nodal Ninja head.
We also tried to stick to a small number of software tools and favor artist-friendly ones, such as Photoshop and PTGui.

Capture

Equipment setup

Colored ring

Cam ranger

COPYRIGHT 2016 © UNITY TECHNOLOGIES

We use a heavy tripod in order to do stable capture in outdoor environments, since wind can be a problem, particularly with long exposure times.

The Nodal Ninja head is used to align the lens center to the pivot point.
The camera must be perfectly parallel to the ground plane.
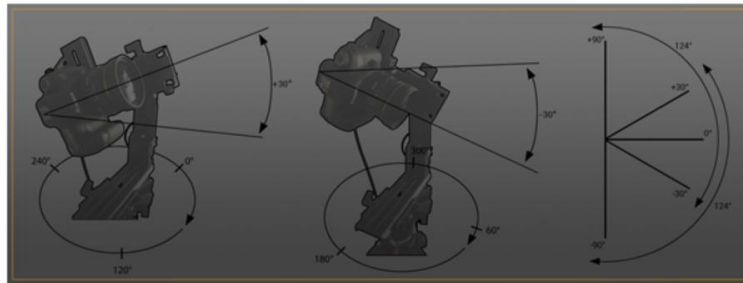Don't hesitate to spend time on alignment as it vastly reduces the time spent on stitching shots later.

We use a remote controller named CamRanger to record the sequence of shots that we want to do. It also allows us to reduce shaking during acquisition.

Note: The pivot point is at the height of a human eye.

To build a 360 panorama we use a fisheye lens. It provides 180° fov.
We have divided the 360° panorama in 6 views:

3 views oriented toward the sky
3 views oriented toward the ground

Several optical artifacts in the lens appear with distance from the lens center.
That is why we chose a 30° elevation instead of 45°.
The goal is to get more accuracy at the horizon, where there is often a lot of moving
elements like pedestrians that need to be removed.

Note: The 120 field of view of a fisheye lens results in the least number of optical
artifacts.

To remove the tripod we capture an extra bottom view
We place a marker on the ground under the tripod at the pivot point, move the tripod
then rotate the camera with the help of a Nadir Adapter (Nodal Ninja Nadir adapter)

Of course we shoot in manual mode with RAW format. We disable automatic white balancing and all color/contrast enhancing features

Camera settings

- An HDRI is a combination of shots of varying exposures
- Exposure value (EV)
  - ISO (S)
  - Aperture (N)
  - Shutter speed (t)

$$EV_{100} = \log_2\left(\frac{N^2}{t}\right) + \log_2\left(\frac{S}{100}\right)$$

- Fix ISO and aperture, vary shutter speed
- ISO 100 to minimise noise

An HDRI is a combination of shots of varying exposures
The exposure is based on 3 parameters: shutter speed (*t*), aperture (*N*) and ISO (*S*) and a given combination of these parameters can be summarized by an *exposure value* (*EV*).
Unity's HDRIs are acquired with fixed ISO and aperture values, and we only vary the shutter speed to control the exposure.
We set the ISO to 100 to minimise the noise

Aperture

- Want accurate images for both near and far scenes
- Small aperture (big f-stop)
  - Diffraction, softer result, significant flare
- Wide aperture (small f-stop)
  - Significant vignetting
  - Low depth of field
  - More optical aberrations
- Chose aperture should provide sharpest images

For the aperture, we want accurate images for both near and far scenes

Small apertures cause diffraction, produce softer results and significant flares. Wide apertures have significant vignetting, low depth of field and aberrations are more prevalent.

The best aperture should provide sharp images.
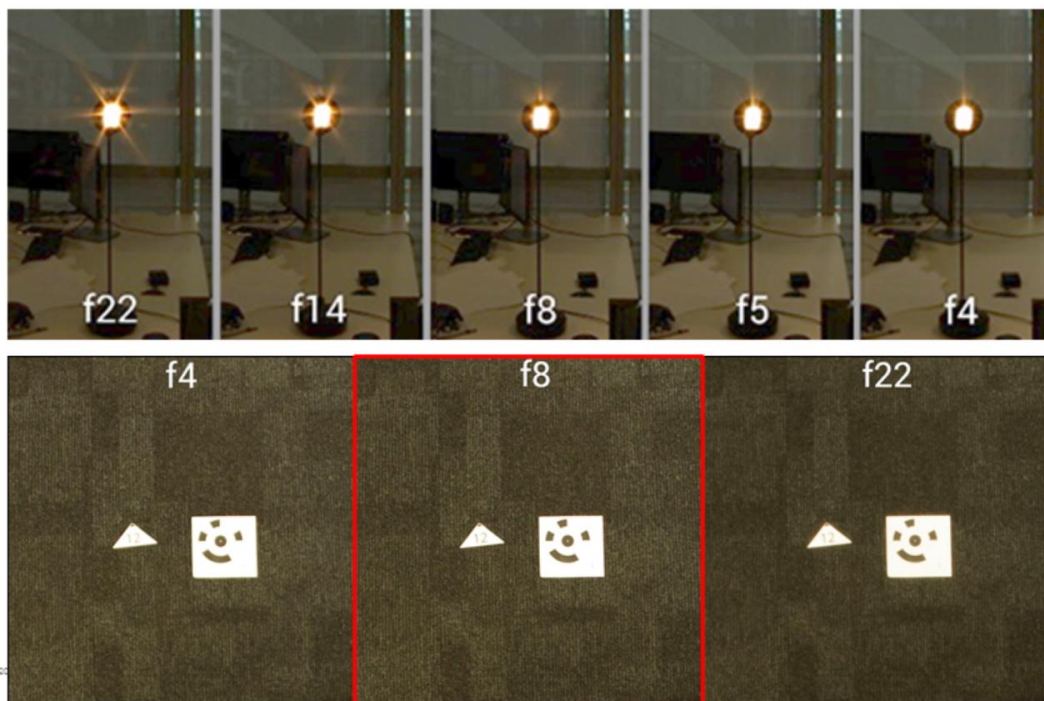
On our tests we found that f/8 provides the sharpest image (click) while at the same time minimizing lens artifacts

Shutter speed

- Varying shutter speed
- Depends on scene range

- Find brightest and darkest area
- Find shutter speed settings not clamping histogram
- Restrict shutter speed to power of two values

For the shutter speed, the value will depend on scene range.
Shutter speed controls the amount of time that the sensor is exposed to light

With camera on the tripod, we locate the darkest and brightest areas of the panorama.
We tweak the shutter speed until the histogram of the camera preview is not clipped.

To simplify the process, we restrict ourselves to power of two shutter speeds

Note: We avoid the brute force solution of using a large number of shots to minimize the amount of processing work later.
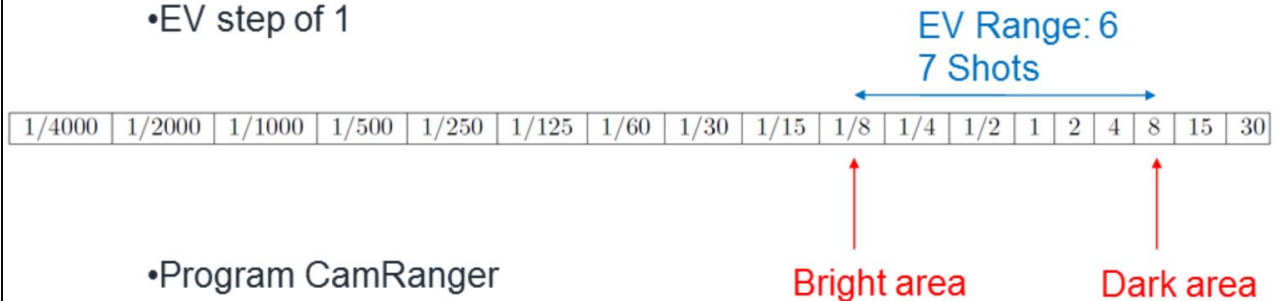
To get the scene EV (exposure value) range and the number of shots to perform we use the property that doubling the shutter speed is an increment of one EV.

We use this table to help us.
With shutter speed for bright (click) and dark areas (click) we calculate the EV range.
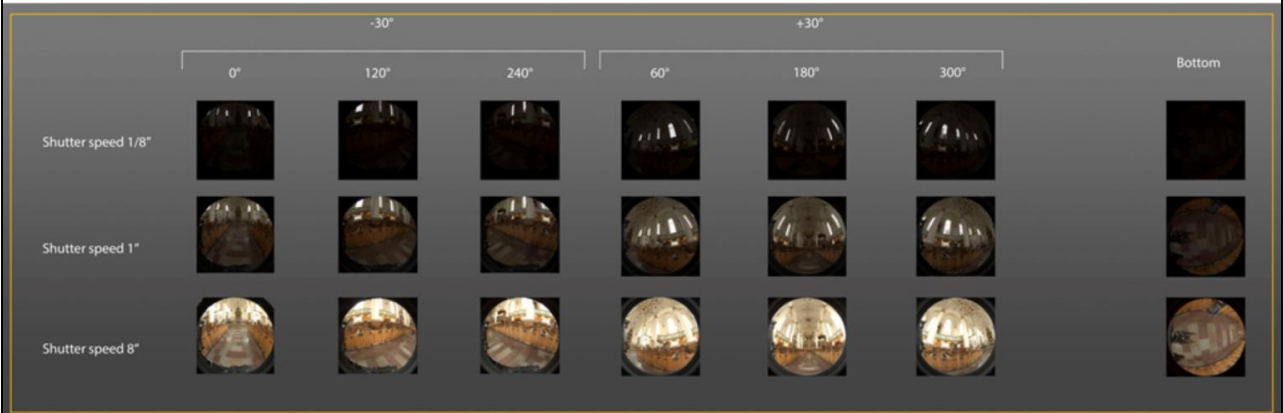Here it is 6 (Click).
With an EV step of 1 it means that we will require 7 shots with the shutter speeds shown in the table.

This shot sequence is programmed into CamRanger

Example of all the shots to take for the previous example and an EV step of 3

(wait)

Shutter speed

•Limited by minimal shutter speed of camera (1/4000)
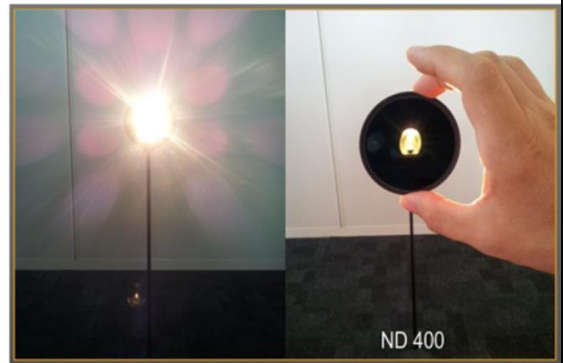•Very bright light sources are clipped

As we vary only shutter speed, we are limited by minimal shutter speed of camera :
1/4000 for a Canon 6D
Scenes with very bright light sources, like the sun or spot lights in a football stadium,
can emit up to 30 EV and will be clipped

Neutral density (ND) filter

- ND32, ND400, ND1024…
- EV reduction = log2(NDnumber)
  - ND32 = -5 EV
  - ND1024 = -10 EV
- Restrict to power of 2
- ND filters can be chained
  - ND32 + ND32 = ND1024

ND 400

To deal with very bright light sources we use Neutral Density filters.

A ND filter can reduce the amount of light entering the lens.
ND filter strength is expressed by a number that is the power of 2 of the EV reduction it provides
ND filters can be chained, their EV reduction adds
Again, we restrict ourselves to power of 2 ND filters to simplify our process

Shutter speed

•Face bright source for 0° view
• Shutter speed at 1/4000

•Use one or more ND filters
•Bright light must disappear

For scenes with a very bright light source the first step is to determine the ND filters to use.
For the 0° view of our process we turn the camera to face the light source.
We setup the shutter speed to the minimum allowed (1/4000" for Canon 6D).
Then we install one or more ND filters in front of the lens until the light disappears on the preview.

Once we know the ND filters to use we increase the shutter speed until the light source is not clipped.

Note: We can't always rely on the histogram here as the sun disc for example can be really small. We control the preview and judge when the light source disappears.

EV	Virtual	No filter	ND16	ND32	ND64	ND500	ND1000	ND500 + ND1000
0	1/4096	1/4000	1/250	1/125	1/60	1/8	1/4	
-1	1/8192		1/500	1/250	1/125	1/15	1/8	
-2	1/16384		1/1000	1/500	1/250	1/30	1/15	
-3	1/32768		1/2000	1/1000	1/500	1/60	1/30	
-4	1/65536		1/4000	1/2000	1/1000	1/125	1/60	
-5	1/131072			1/4000	1/2000	1/250	1/125	
-6	1/262144				1/4000	1/500	1/250	
-7	1/524288	Bright area →				1/1000	1/500	
-8	1/1048576					1/2000	1/1000	
-9	1/2097152					1/4000	1/2000	
-10	1/4194304						1/4000	1/8
-11	1/8388608							1/15
-12	Bright area							1/30
-13	1/33554432							1/60
-14	1/67108864							1/125
-15	1/134217728							1/250
-16	1/1000	1/500	1/250	1/125	1/60	1/30	1/15	1/500
-17								1/1000
-18	1/1073741824							1/2000
-19	1/2147483648							1/4000

To get the scene EV range and shutter speed for the shots, we use this big table.
It lists the power of two ND filters we use at Unity.

ND filters allow us to emulate a decrease of EV without changing the camera settings.
(Click)
You can think of it as if we were able to decrease the shutter speed below the
camera's limit, like a *virtual* shutter speed value (Click)

Let's say our light source disappears with an ND500 (Click). And we find the bright
area shutter speed to be 1/1000 (Click) . We get a list of shutter speed values(click).
We do not take into account the EV 0 as it is redundant.
(Click) I change the green vertical section to horizontal…
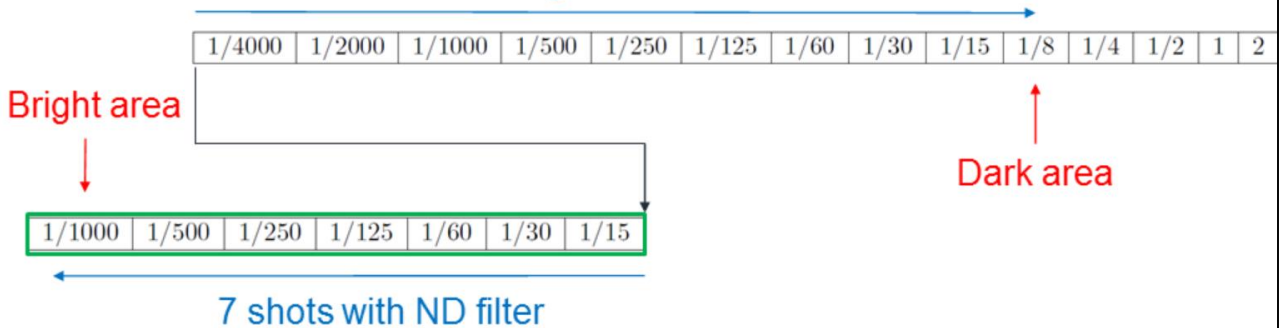
…and I concatenate it with our regular shutter speed table.

With an EV step of 1, the green table gives the number of shots to do with the ND filter and with which shutter speed values (Click).
After the ND filter shots we perform a second round of shots but without the ND filter (Click).
The scene EV range in this case is the number of shot minus one.

Example of shots that have been captured for a scene with 23 EV.

In practice the sun in a clear sky requires ND1000 + ND500 filters.
(wait)

White balance

- Human vision does chrominance adaptation
- White balance for photography
- Technically incorrect for panoramas
  - Human vision < 360°
  - Varying lighting conditions
  - Should be done when visualizing the HDRI
- Unity stays with *classic* process
- Extra bottom view with white target
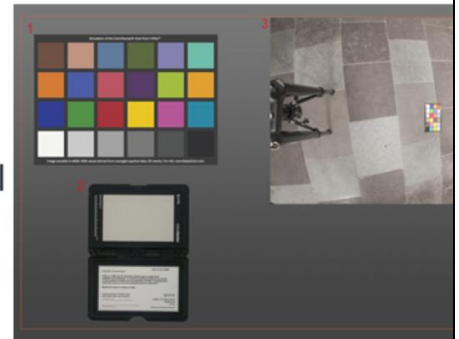
Human vision does chrominance adaptation to always perceive a white object as white whatever the lighting conditions.

In photography we try to obtain an image that is close to human vision. We simulate this adaptation with a process called white balancing.

Performing uniform white balance on the 360° view of a panorama is incorrect as a human eye has a smaller field of view.

Eyes can perceive different lighting conditions depending on orientation in the CG scene, best is to do white balancing at visualization time.

We chose to stay with classic white balancing for now.

We capture an additional bottom view with a color checker to do it later.

In addition to white balance we also want to perform luminance calibration.
With our workflow, the HDRI produced has relative luminance and we want to convert it to absolute luminance, matching the real world.
This calibration requires taking light measurement during the capture.

There are cheap and expensive devices to do this: lux meters, spot meters or luminance meters.
We use a lux meter for this step as it is a cheap and accessible device. It captures the light coming from upper hemisphere of the meter
We do the measurement of the illuminance of the scene at the camera pivot point.
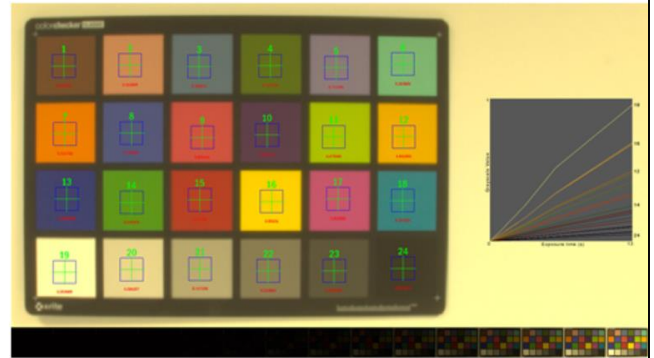
We use dcraw to convert our RAW files (.CR2 for Canon) to TIFF 16bit.
It allows us to not perform the white balancing stored in the metadata.
dcraw has also a good linear reconstruction that is important to have correct HDR values.
dcraw seems to be the only software that conserves linearity, other software such as Adobe Camera Raw applies some gamma conversion and artistic tweaks.

Note: Camera Raw also adds some contrast enhancement by default, and some black-box color tweaks for aesthetic reasons

Lens artifacts

- Lateral chromatic aberration
- Axial chromatic aberration
- Vignetting
- Distortion
- Veiling glare
- Lens flares

Software correction

Not corrected

Camera optics systems cause lots of artifacts.
We try to fix what we can.

For veiling glare and lens flares, we chose to not correct them.

Chromatic aberration is caused by the variation in refractive index for different light wavelengths of the lens.
To correct lateral chromatic aberration we use Photoshop's lens correction tool.
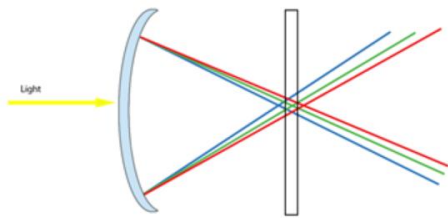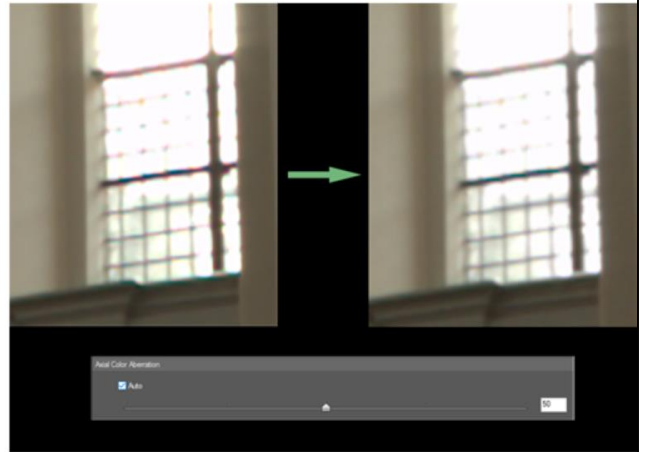
To correct axial chromatic aberration we use a software from Nikon called ViewNX2.

Vignetting

- Lens characterization for f/8 aperture
- White screen + moving white paper + dark room
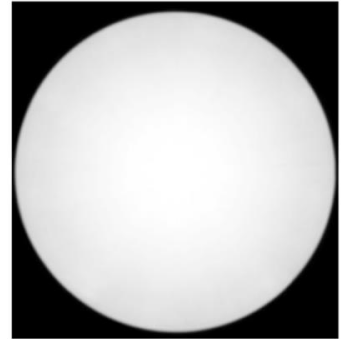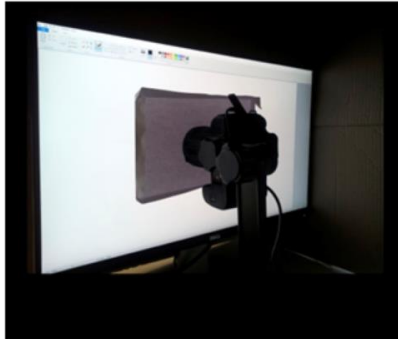
Vignetting can be significant. It depends on the lens and the aperture.
We have developed a simple method to characterize the lens vignetting.
We take a direct shot of a white piece of paper on a white screen.
Shooting the screen directly exhibits the pattern of LEDs. The paper also has a pattern, so we move it quickly during the shot to blur it a bit.
Then in Photoshop we blur and normalize the image.

Note: Lens vignetting for our fisheye lens at f/8 is not significant for our 120° field of view.
Note: The screen is to have a backlighting for the lens, else it is dark.

We use the same setup to characterize our ND filters.
To recover the vignetting we divide a picture taken with the ND filter by one taken without the ND filter.
Note: The shots must have the same exposure.
Note: The pink area is due to noise outside of the fisheye part.

Vignetting

•ND filters' vignetting exhibit chromatic shift

Visual remap
[0.9-1] to [0-1]

ND16   ND32   ND64   ND500

The ND filter is not always homogenous and has a very slight chromatic shift.
You can see it on the image here in yellow.
The image shows the vignetting characterization of our various Hoya ND filters.
It means that there is a very slight blue tint. The image has been rescaled.

The full characterization is lens vignetting multiplied by each ND filter's vignetting.
To apply vignetting correction, we divide vignett<u>ed</u> shot by the full characterization in Photoshop.
With our equipment and our 120° fov shots, we found that correction was not necessary.
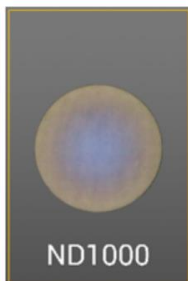
In addition to characterizing vignetting, our setup allows us to detect when an ND filter doesn't match its specification with the compensation to apply.
Our ND1000 is not a true 10 EV reduction and when using it we correct our shots.

Note: Lens characterization is still multiplied by ND1000 characterization in this case.
Note: Required to match our power of two process.

PTGui

- Handle distortion
- PTGui gamma corrects TIFF 16bit inputs!
  - Convert to 32bit TIFF
- Create HDRI and HDRI with X-Rite ColorChecker

We use PTGui to create our HDR panorama.
We correct the distortion inside PTGui, stitch our shots and generate two HDR panorama. One with and one without color checker.
PTGui applies a gamma correction on 16bit TIFF images so we need to convert our TIFF from 16bit to 32bit.

If the equipment is perfectly aligned, then the stitching should be easy.

PTGui

- PTGui uses exposure embedded in EXIF
- Shots with ND filters do not change EXIF metadata
- Produces wrong results

To build the HDR, PTGui uses exposure embed in EXIF
This information is not changed when using a ND filter and so it produces wrong results

We must edit *exposure* manually in PTGui (Click) with the value of the *virtual* shutter speed I mentioned earlier (Click)
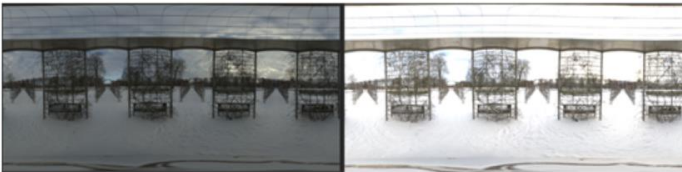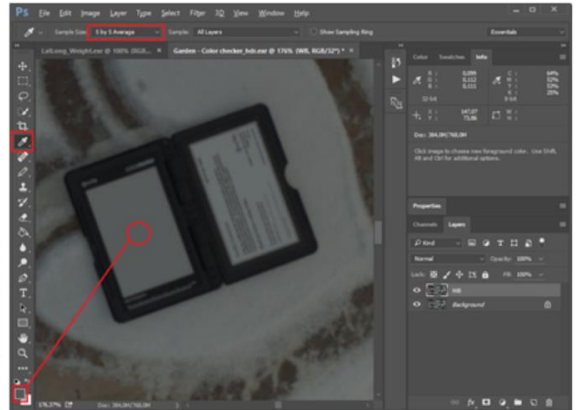
(wait)

This is an example of an outputted HDRI. You can see various exposures with the intense sun disc.

For white balance we have develop a Photoshop script.
We white balance the color checker contained in the HDRI with color checker and apply the gain to the HDRI without color checker.

Luminance calibration

- Ratio of light measurements

$$RGB\ calibrated = RGB * \frac{Real\ world\ measurement}{Image\ measurement}$$

- Illuminance from lux meter
- Illuminance from HDR panorama

$$\int_{\Omega_{upper}} L\cos(\theta)\sin(\theta)\mathrm{d}\theta\mathrm{d}\phi = \int_{\Omega_{upper}} L\cos(\theta)\sin(\theta)\frac{2\pi}{w}\cdot\frac{\pi}{h}$$

For luminance calibration we need to find the ratio between the real world illuminance that we get with the lux meter and the illuminance from the HDR panorama.
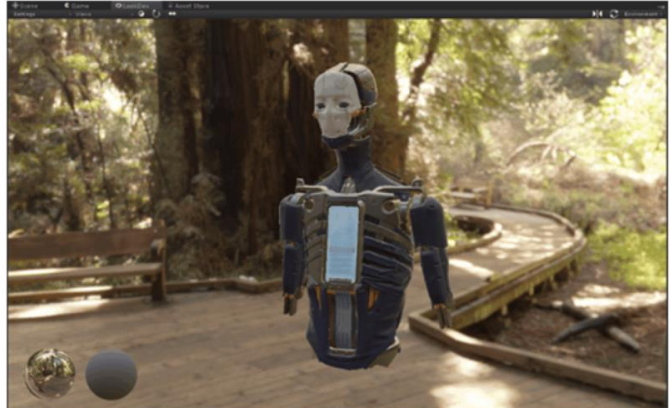The math to get the illuminance is not really artist friendly…

…But we have found a way to do this integral with a Photoshop script.
The script will apply the ratio of light measurements to the white balance HDRI.

Example of HDRI usage

At Unity we use HDRIs for our new look development tool.
It allows us to visualize an asset under various lighting conditions represented by an HDRI.
As the lighting is perfect, the artists can validate their materials.

Note: We have an HDRI browser in this tool and an option to generate hard shadows for one very bright light source.

Absolute HDRI

- Absolute HDRI usage?
- Play well with other games' lights
- For engines that support physical light units
  - See "Moving Frostbite to PBR" from SIGGRAPH 2014 PBS Course
  - Unity in the future!

Absolute HDRIs are in case the game engine uses physically based light units for other lights. This ensures that values are in the right range.
Unity doesn't support this yet but we have a plan for it.

Conclusion

Conclusion

- We think our process to capture accurate HDRI is:
  - A good balance between time, price, quality and complexity
  - Artist friendly
- Lens artifacts are very difficult to correct
- Plenty of details and more in course notes

To conclude, we think our process to capture accurate HDRI is a good balance between time, price, quality and complexity.
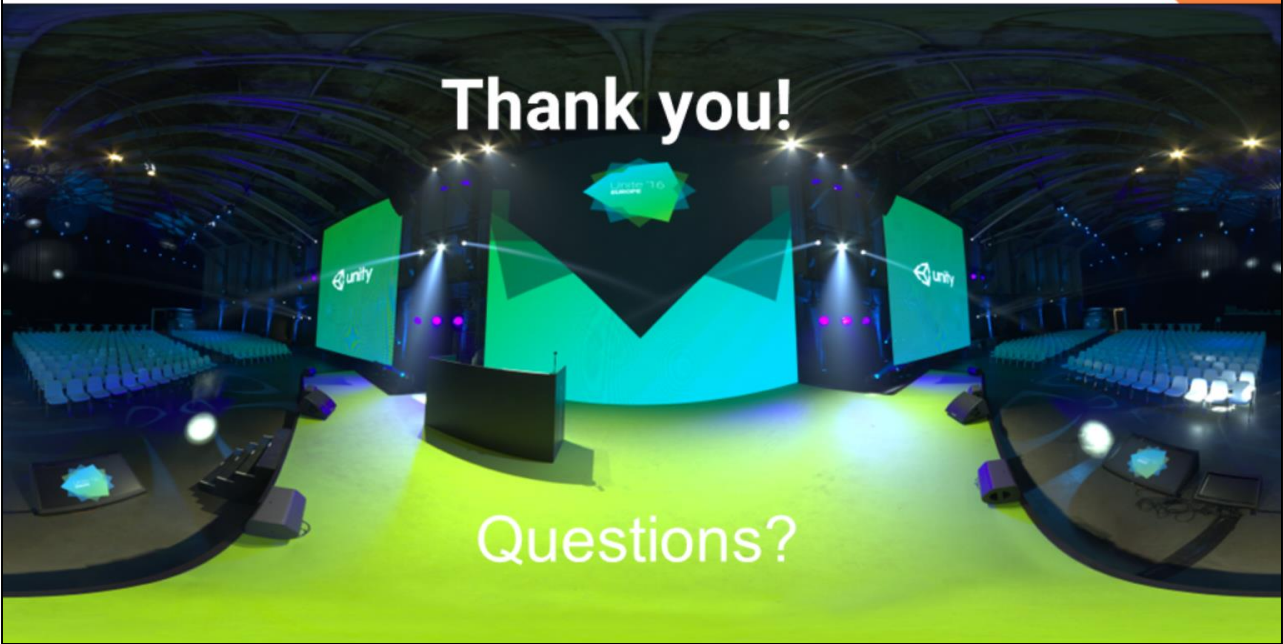We have tried to limit the number of software packages and avoided custom devices to remove lens artifacts.
Lens artifacts are very difficult to correct and we don't think we can improve it and stay artist friendly.
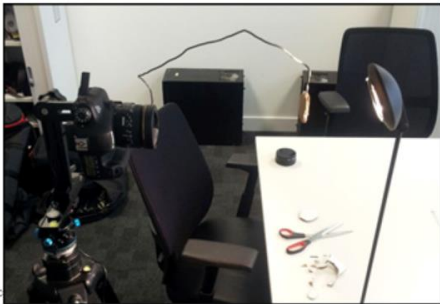There are plenty of details about our process, which we elaborate on in our course notes.

Bonus slides

We tried to hide the very bright light source with a piece of cardboard during the capture to remove lens flare.
Two HDRIs are processed, one with and one without the cardboard, then they are combined in Photoshop.

This works, but finding the correct position to get the correct occlusion takes time and the wind is a real problem outdoors!
We often have to block the wire with our hand to avoid it to moving too much

But…

… With more than one source to hide it is impractical.

At Unity we use this method in limited cases like the sun in a clear sky. With cloudy skies the sun flare is not visible.

Unity lookdev

- HDRI with sun
- Artists want hard shadow
  - Don't add extra directional light!
- Need to stay close to game look
  - Modulate shadow generated from sun direction
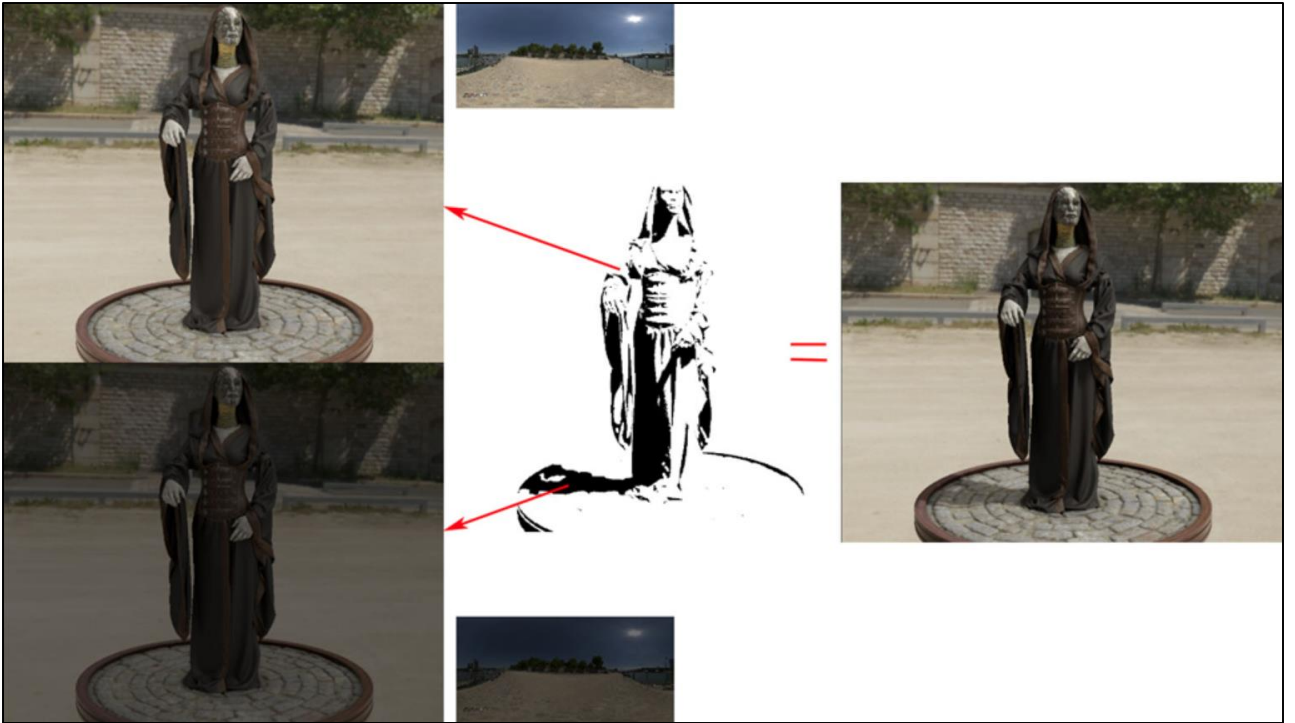
Highlight with modulated shadow   Shadow visualization

Sun removal

- Remove sun from HDRI with Photoshop stamp tool

Highlight with modulated shadow   Shadow visualization   Sun remove

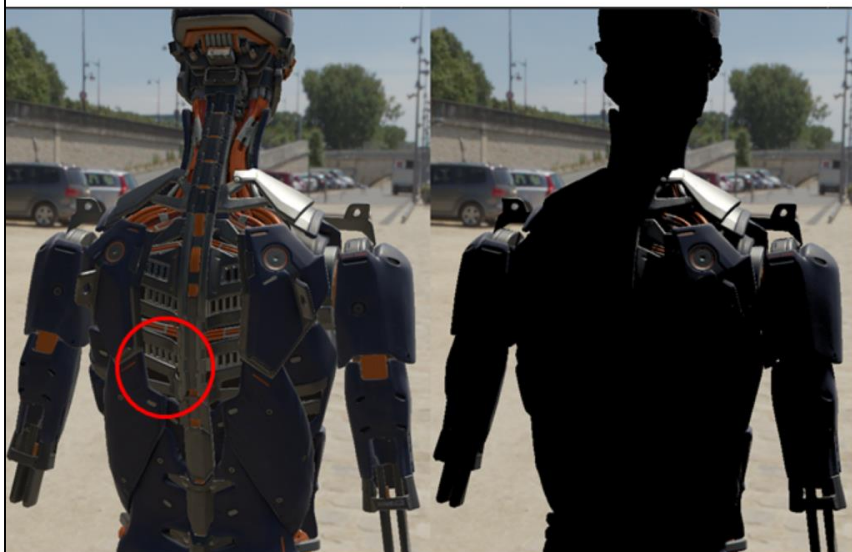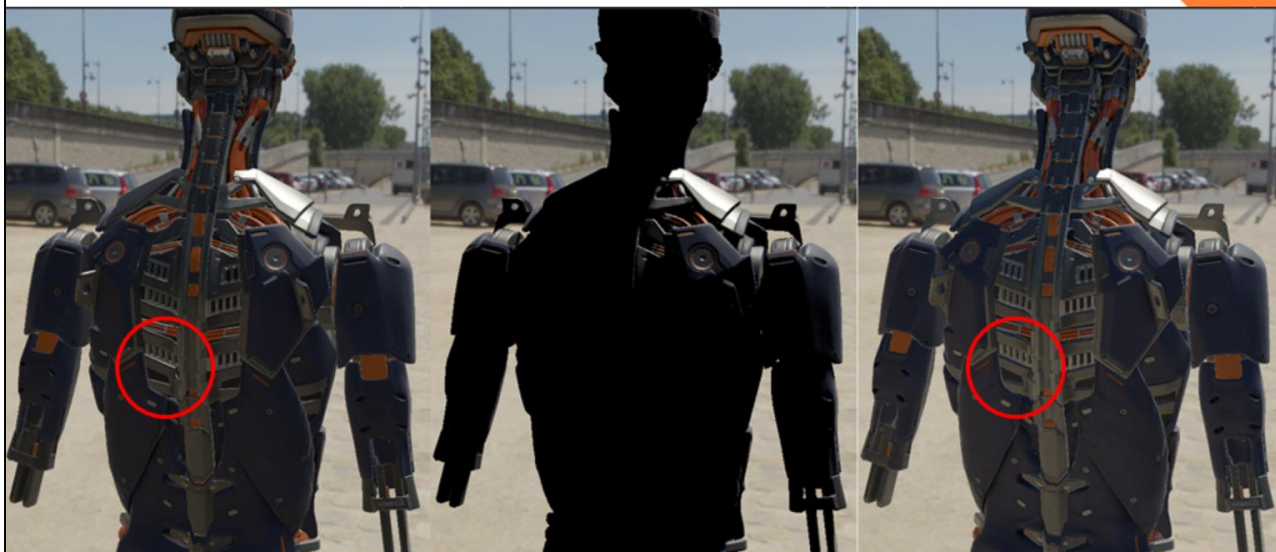# HDRI + directional

- Replace sun by a directional light
  - Remove sun from HDRI
  - Calculate illuminance difference (with Photoshop)
  - Solve equation:

$$\text{illuminance}_{\text{HDRI}} = \text{illuminance}_{\text{DirLight}}\cos(\theta) + \text{illuminance}_{\text{ShadowHDRI}}$$

$$\text{illuminance}_{\text{DirLight}} = \frac{\text{illuminance}_{\text{HDRI}} - \text{illuminance}_{\text{ShadowHDRI}}}{\cos(\theta)}$$

- Engine uses lux units for directional lights without knowing it ☺

Everything matches. The difference is due to a slight problem in Unity's HDRI integration.

Luminance calibration

- Validation with luminance meter measurement
- Method is good but measurements aren't
  - Changing lighting conditions
  - Pivot point location

Name	Interior Exterior	Stability	Lux	Nits	Recover Nits	Match
Station	int	stable	127	18.22	17.75	97%
Christian Church	int	unstable	64	9.75	10.30	94%
Trinitatis Church	int	unstable	73	8.94	10.32	84%
Unite	int	unstable	580	160	190	81%
Treasure Island	ext	unstable	51000	6700	6940	96%
Seine Bridge 02	ext	unstable	36	7.9	9.4	81%
Stadium	ext	stable	113	26.5	20.5	77%
SeaSide	ext	unstable	11800	1541	1910	76%
Marina Beach	ext	unstable	58000	6300	8270	69%
Jardin Plantes	ext	unstable	382	34	45.2	67%
Treasure Island 2	ext	unstable	31000	3600	4850	65%
KirbyCove	ext	unstable	1800	580	310	53%
Stade Small	ext	unstable	4000	468	708	49%
Garden	ext	unstable	7900	1000	1570	43%

Try to validate the method by comparing with the value taken with an accurate luminance meter device.
Compare illuminance and luminance measurements on several HDRI.

Our conclusion is that you can use either a lux meter or luminance meter, both method are valid. The true problem is about the measurement itself.
In case of stable lighting like in an indoor environment without windows, both methods match.
In case of outdoors with moving clouds, late or early hours, the measurement varies over the whole capture. The HDRI itself is acompound of different lighting condition.
In this case we believe that the lux meter could be more accurate than the luminance meter.

Future

- White balance
  - Try in-game automatic white balance
  - Try full color checker calibration
- Absolute HDRI
  - Improve measurement for better accuracy