HDR image based lighting: from acquisition to render

Artem Krizhanovskiy
Technical Artist Wargaming

Image based Lighting: applications

- PBR : Maxwell, Corona, V-ray, etc.
- Content creation software: SP, Marmoset, etc.
- Modern game engines
- Cinema and commercials

Image based Lighting:
V-ray


Image based Lighting:
V-ray
1

Image based Lighting:
V-ray


Image based Lighting:
Substance Painter

Image based Lighting:
Substance Painter

Image based Lighting:
Substance Painter

Image based Lighting:
Game Engine

Image based Lighting:
Game Engine

Image based Lighting:
Game Engine

Image based Lighting

- Photorealistic
- Physically correct
- Photometric units
- Environment matches Lighting

HDR : tonemapped output, linear radiance map

LDR : tonemapped radiance map, linear output. “JPEG” sky

Why are they different?

IBL as a sum of point lights

$$IBL = BRDF(pixel_1) + BRDF(pixel_2) + \cdots + BRDF(pixel_i) = \sum_{i=0}^{n} BRDF(pixel_i)$$

$$\text{Tonemap}\left[\sum BRDF(pixel_i)\right] \neq \sum BRDF(\text{Tonemap}[pixel_i])$$

<u>The function of sums does not equal the sum of functions</u>

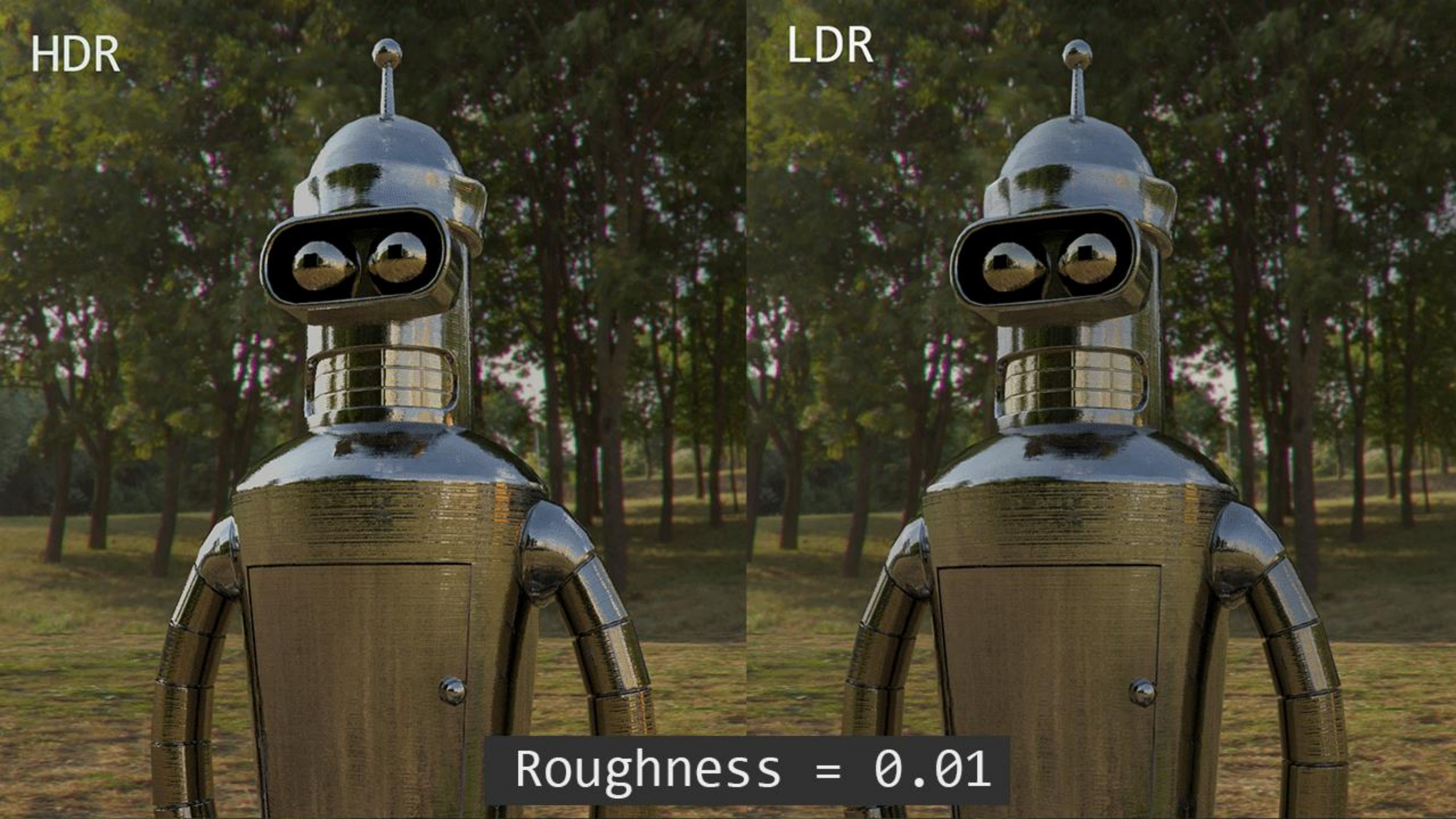

HDR
LDR
Roughness = 0.01

Image based Lighting

- Key to correct lighting is your radiance
- Tonemapped output, not assets
- Tonemapping is essential part of any true HDR rendering

Environment map as radiance:

- Full luminance range
- No post-processing
- No White Balance
- Linear

Radiance map: capture

Radiance map: equipment

- Full frame DSLR camera
- 8mm F3.5 Circular Fisheye
- Panoramic head
- Tripod
- Illuminometer

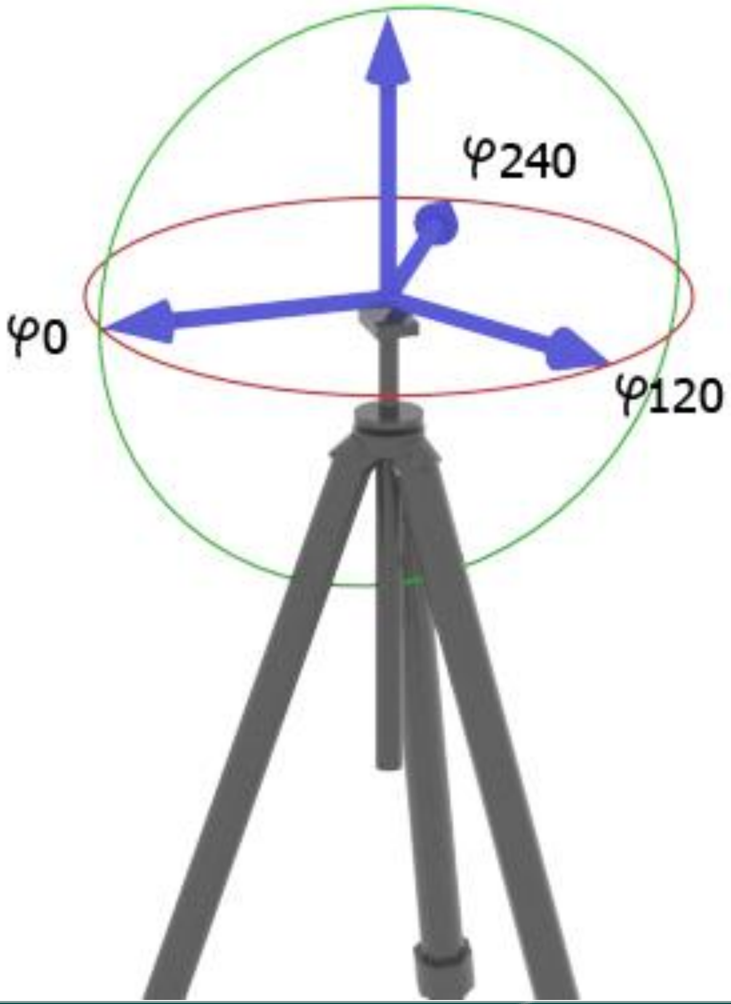
Radiance map: camera setup

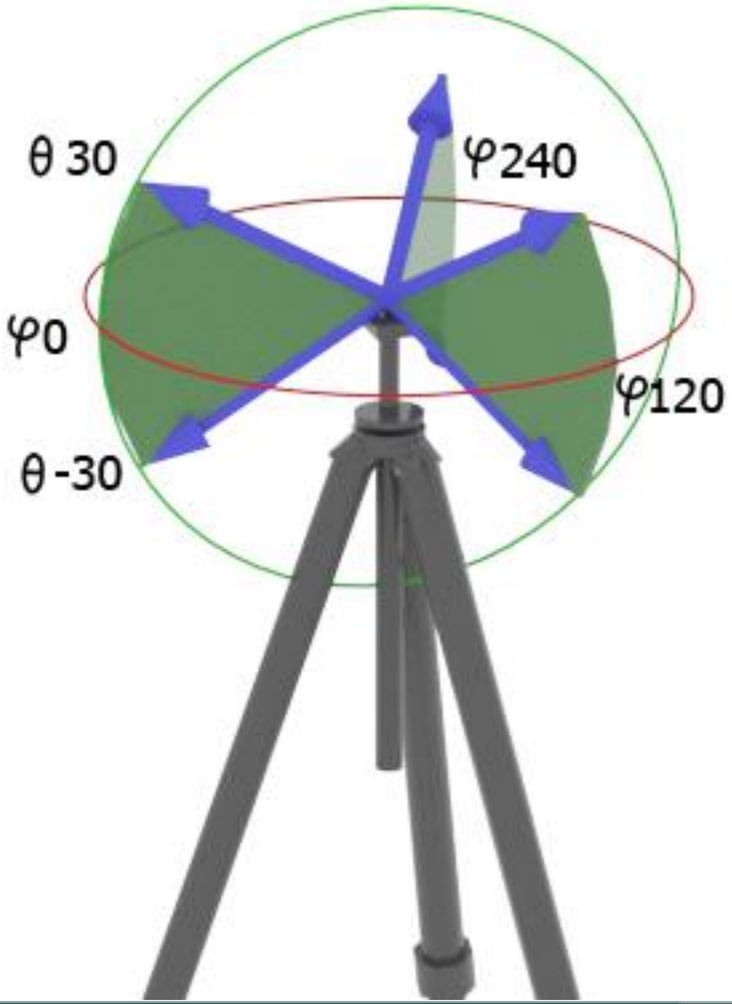
- Manual exposure
- All artistic modes off
- 6500k white balance
- Bracketing

Radiance map: exposure bracketing

- Depends on environment
- Variable shutter, constant aperture

Radiance map: camera position





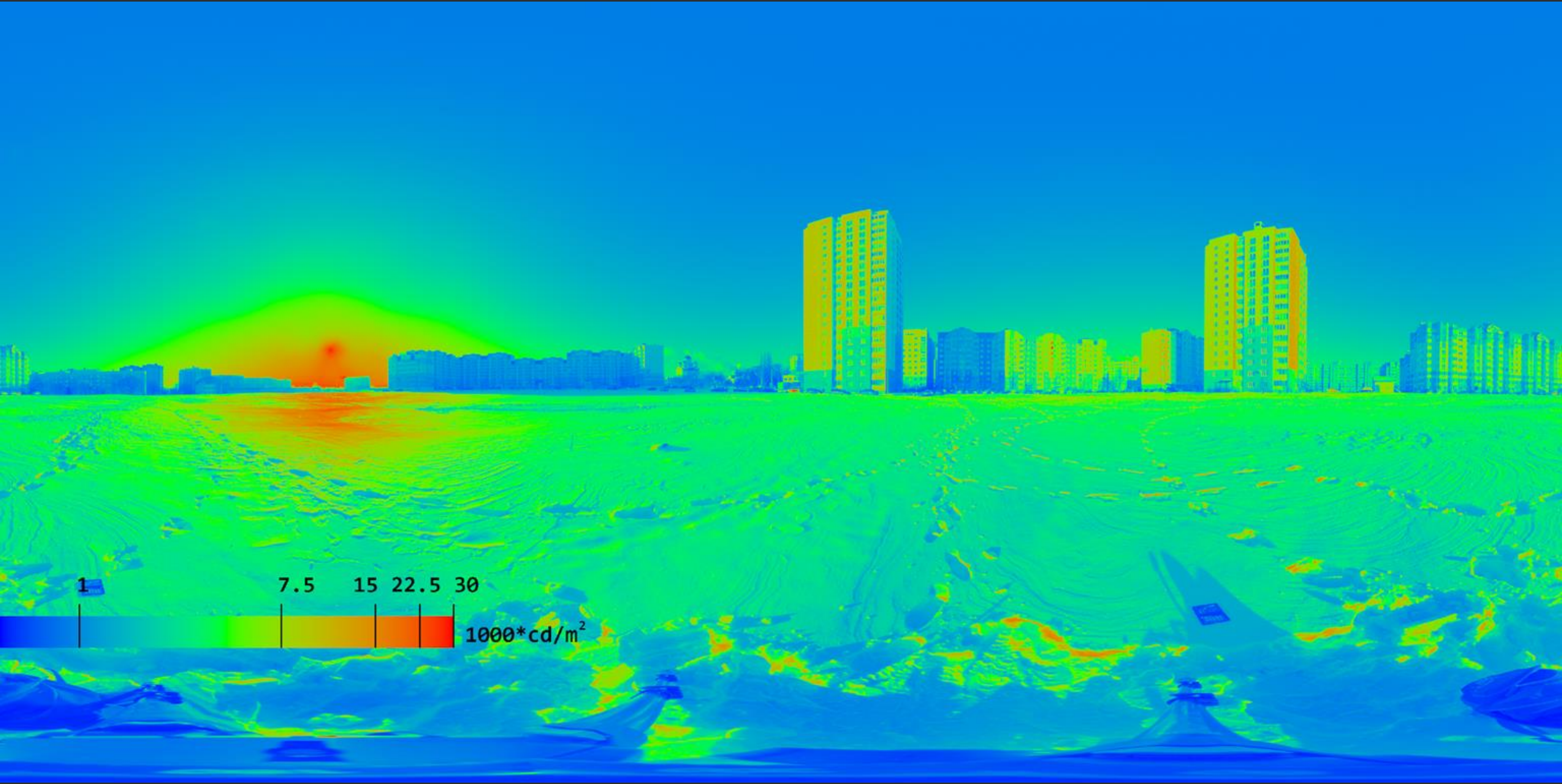
Radiance map: probe?

- Slightly lesser shots
- Too imperfect
- Dead end for HI-res HDRI

Radiance map: Exposure

- All color values are in range
- Best possible aperture


clipped


1
7.5
15
22.5
30
1000*cd/m²

Radiance map: Exposure

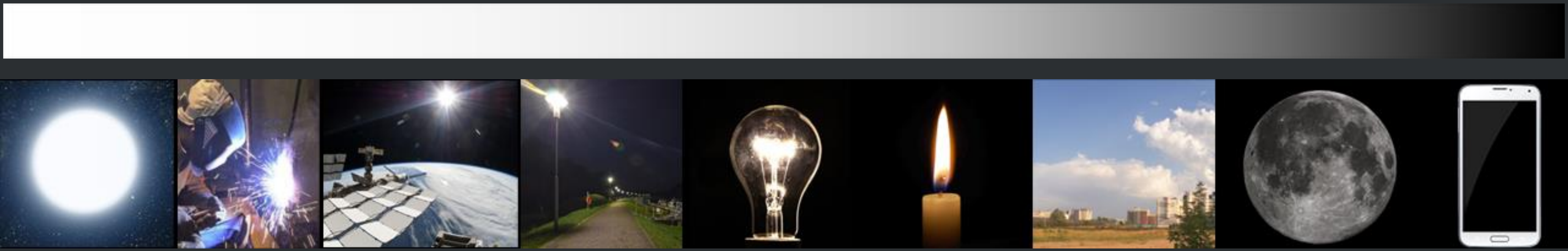
- Sun is really astronomically bright
- Blue sky is 1000-5000 $\frac{cd}{m^2}$
- Max capturable brightness is 4.6*10^6 $\frac{cd}{m^2}$

which is still
~350 times darker than Sun

Real world luminance values

99% of possible objects lies before 10^4 Nit, nonetheless remaining can affect the lighting greatly

10^{14} | 10^{11} | $1.6 * 10^9$ | 10^8 | 10^7 | 10^4 | $10^4 - 10^5$ | $\sim 2 * 10^3$ | $5 * 10^2$



Supergiant star | Electric spark | Sun on orbit | UHP lamp | Tungsten lamp | Candle | Sky | Moon | Screen

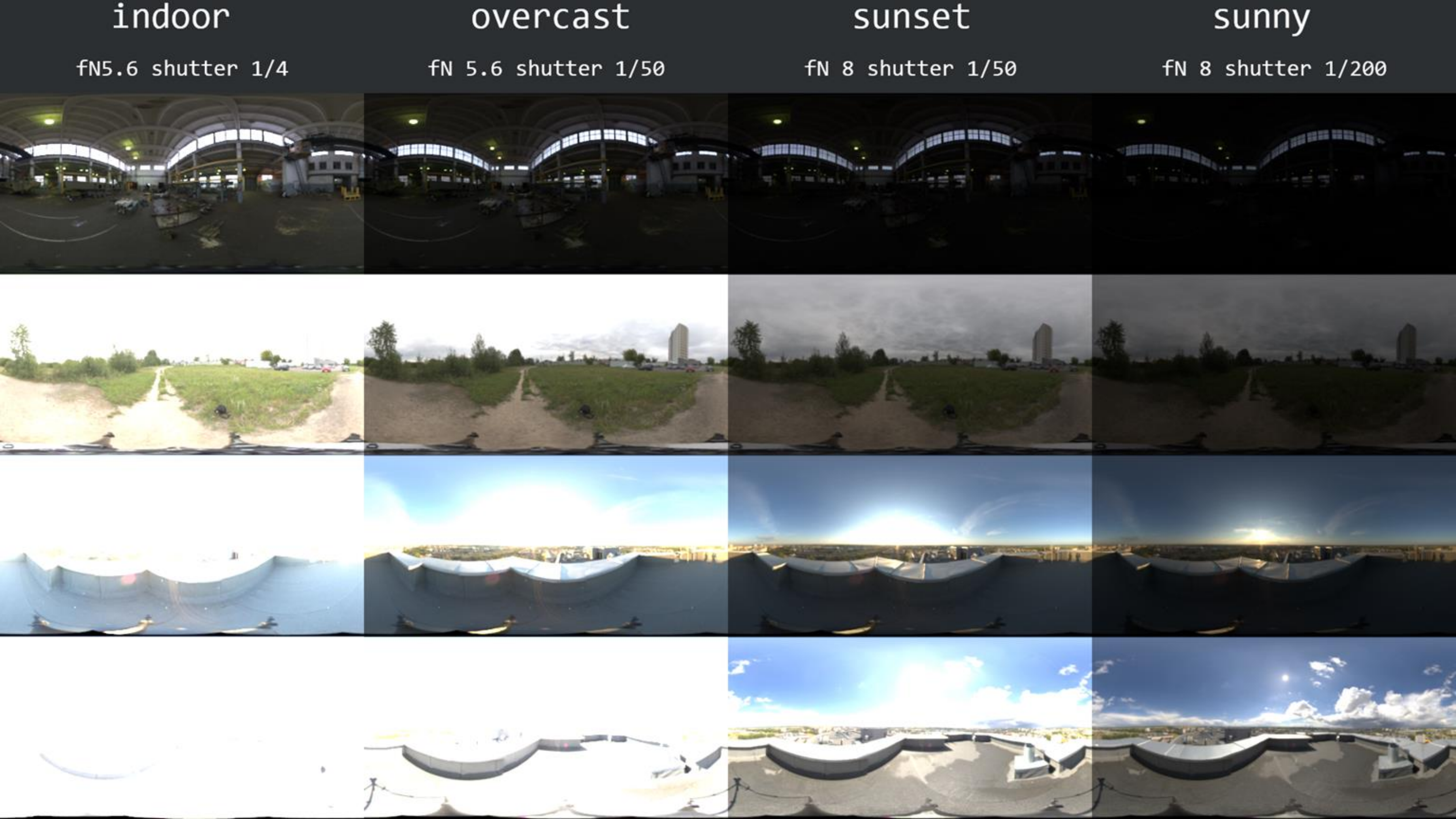
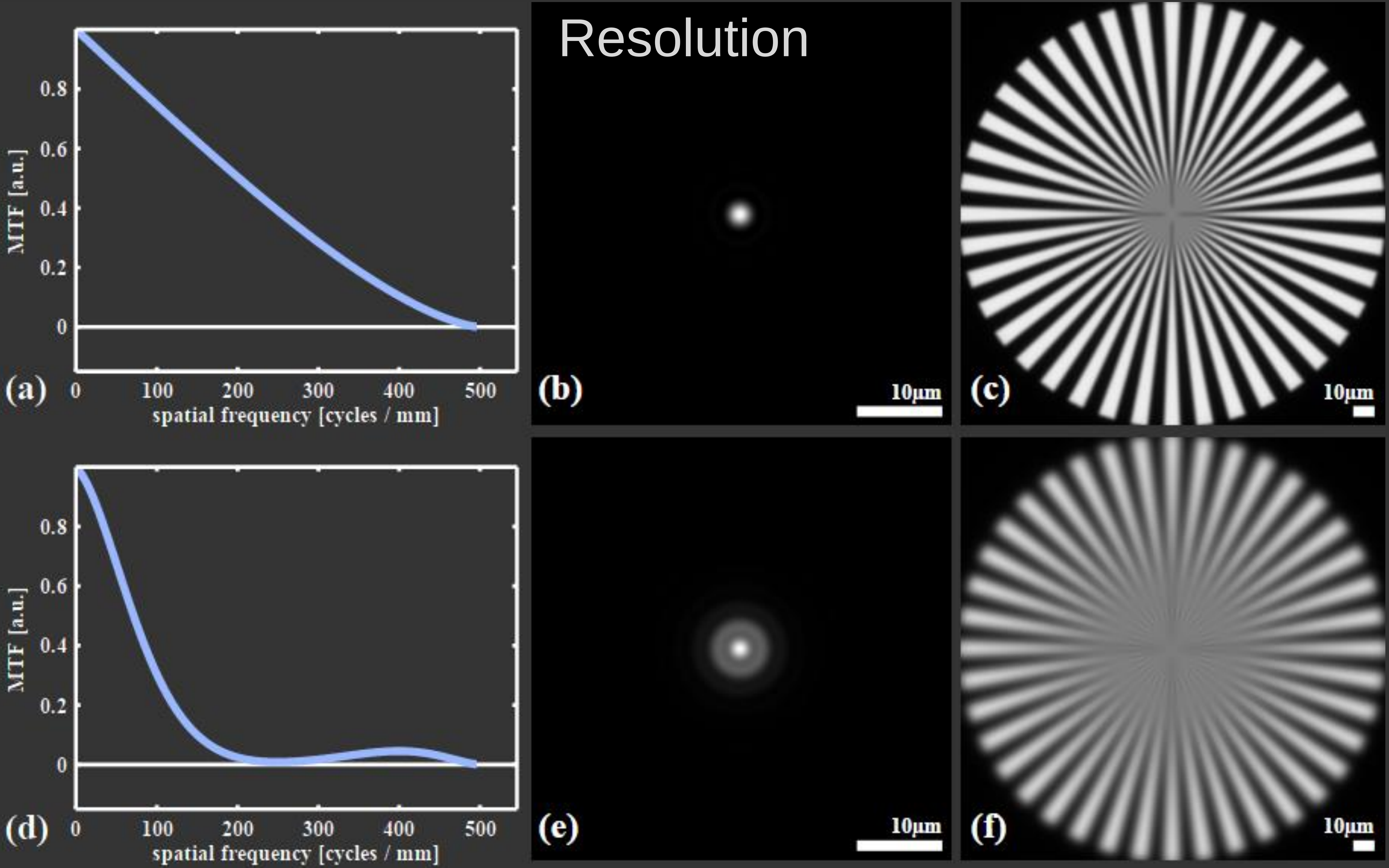

indoor
fN5.6 shutter 1/4
overcast
fN 5.6 shutter 1/50
sunset
fN 8 shutter 1/50
sunny
fN 8 shutter 1/200

Image quality

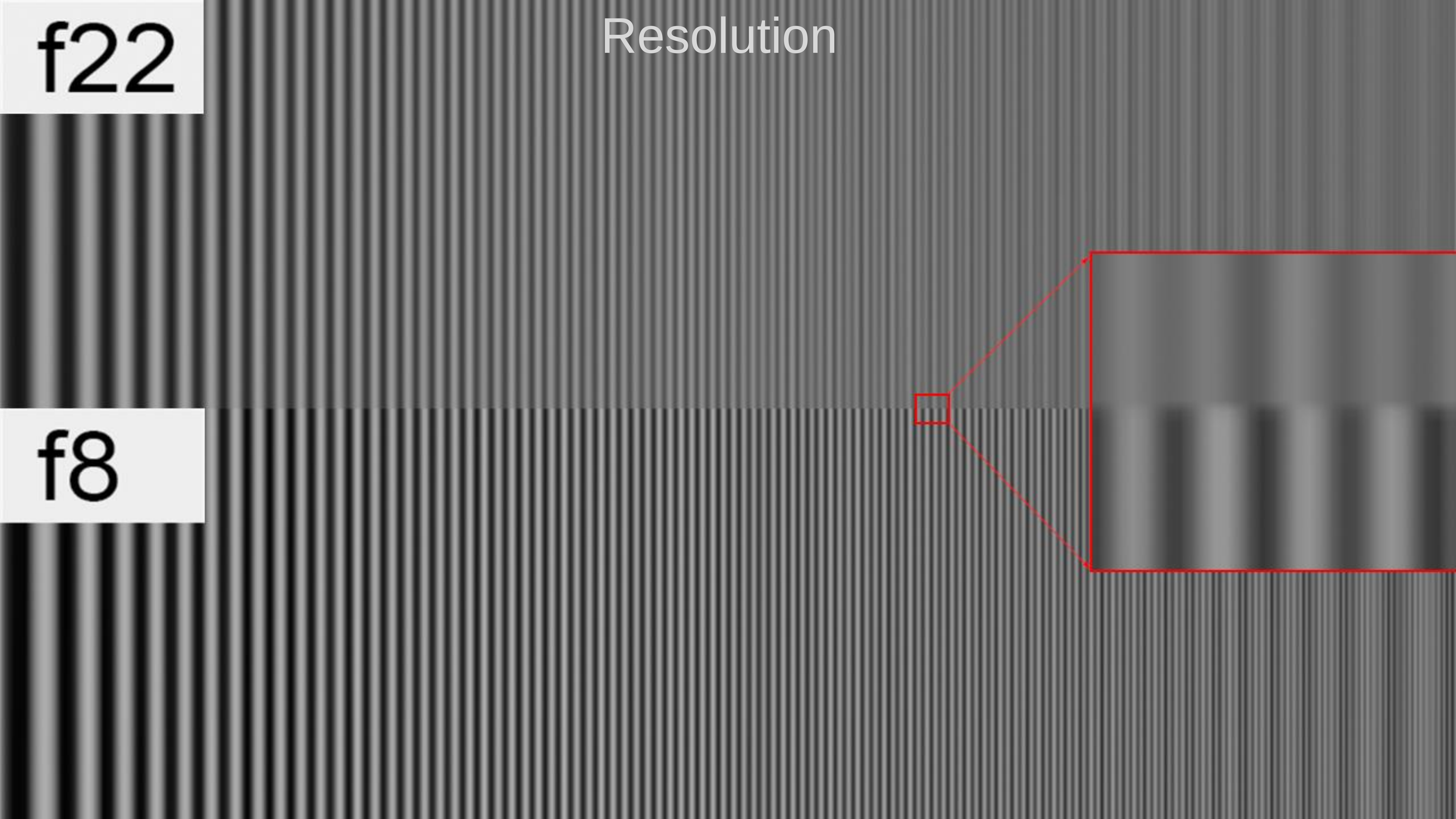
- Resolution
- Lens Flare
- Noise

Resolution

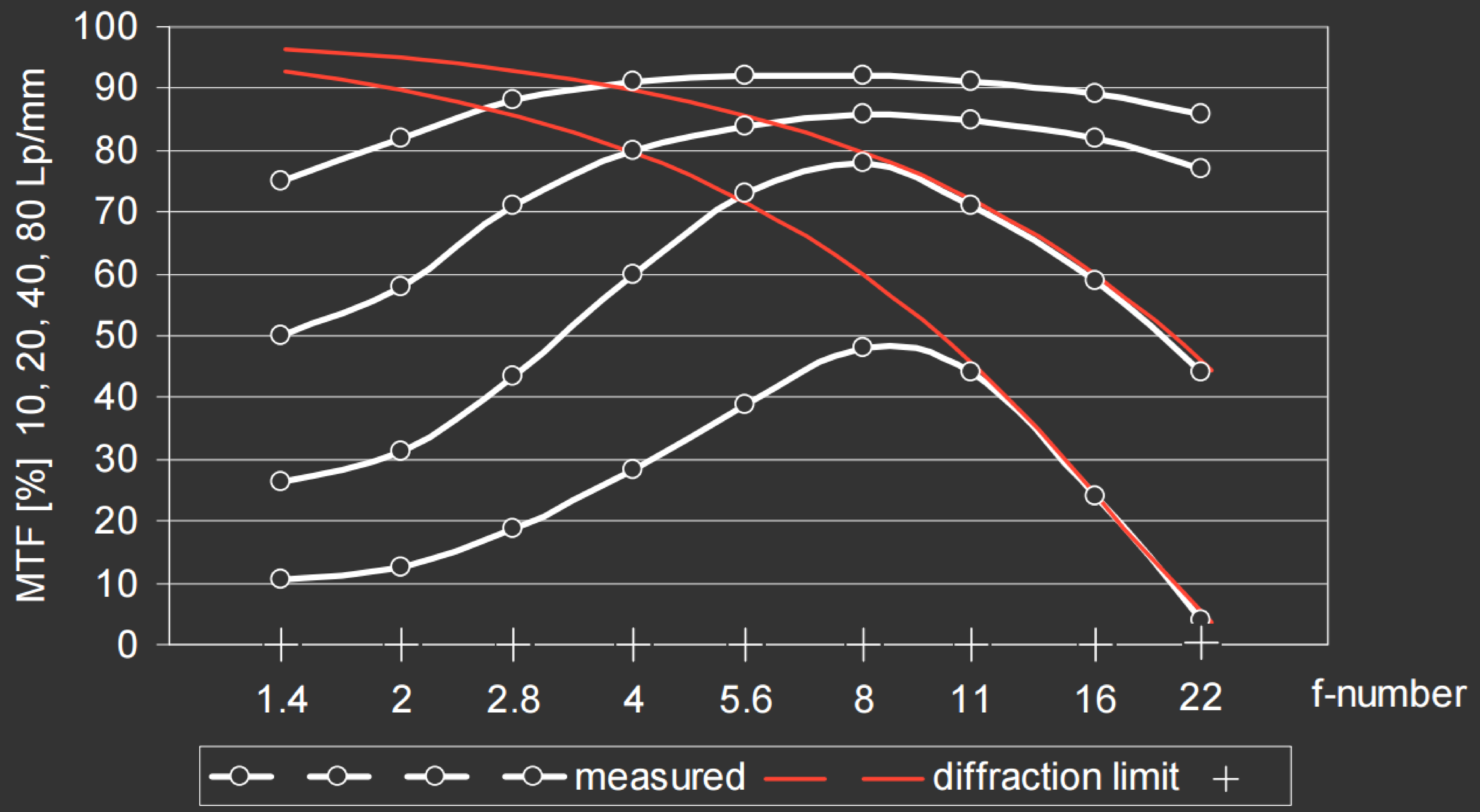


MTF [a.u.]
0.8
0.6
0.4
0.2
0
(a)
0
100
200
300
400
500
spatial frequency [cycles / mm]
(b)
10μm
(c)
10μm
MTF [a.u.]
0.8
0.6
0.4
0.2
0
(d)
0
100
200
300
400
500
spatial frequency [cycles / mm]
(e)
10μm
(f)
10μm

Resolution



Resolution



100
90
80
70
60
50
40
30
20
10
0
MTF [%] 10, 20, 40, 80 Lp/mm
1.4
2
2.8
4
5.6
8
11
16
22
f-number
measured
diffraction limit

Image quality: Lens flare

- Flare is not a light source
- Bigger fN means bigger Flare
- Occlude the sun if possible

Lens flare from aperture

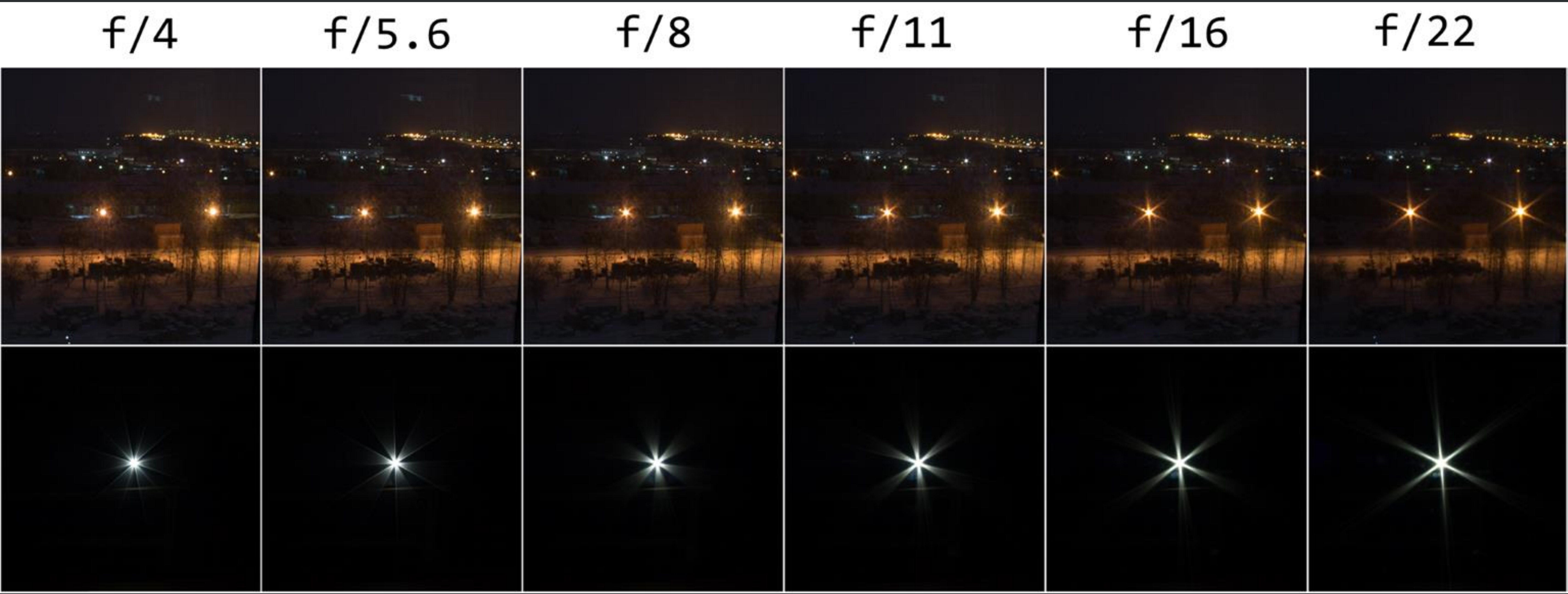
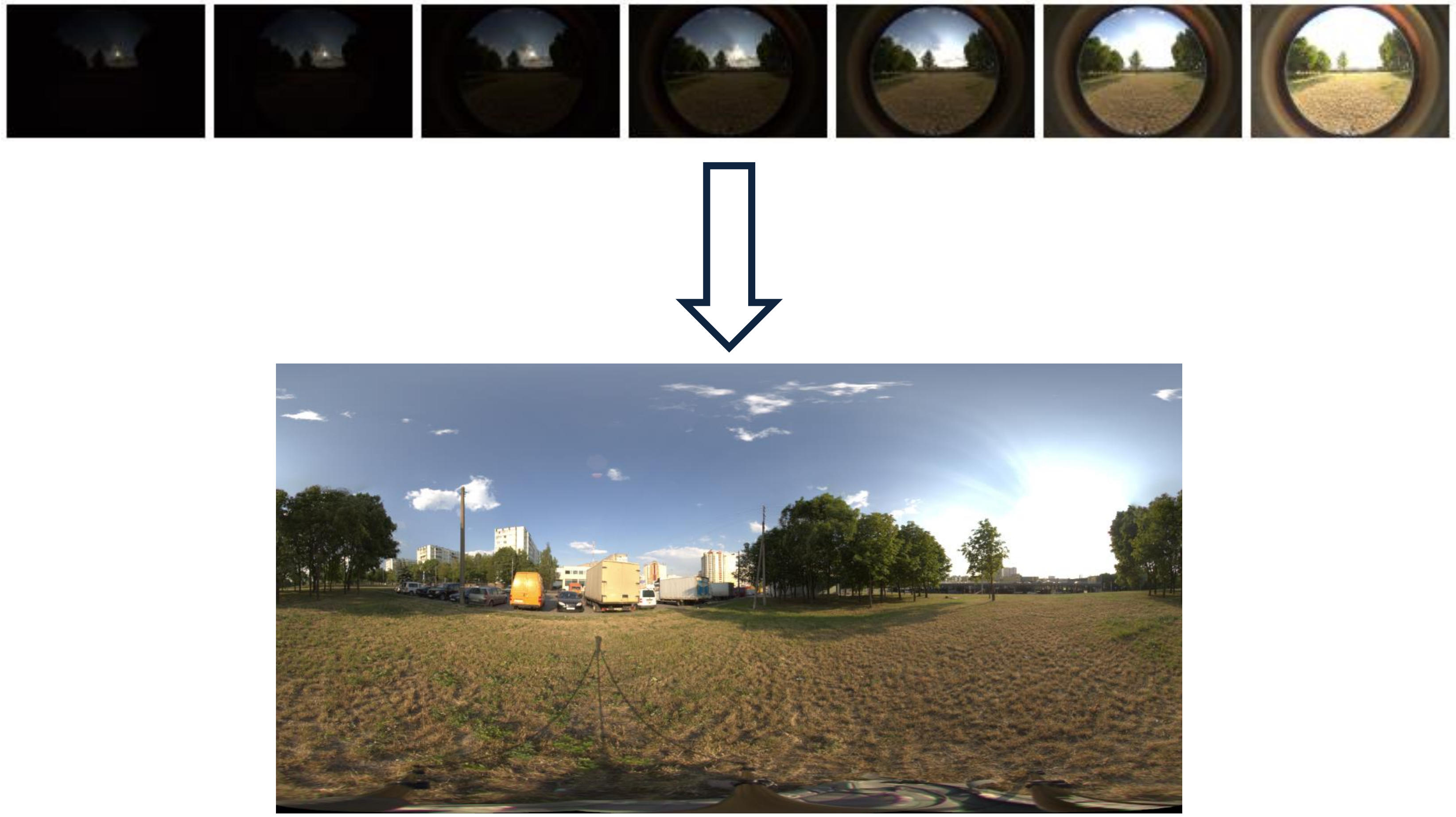


Image quality: noise

- ISO is multiplier of output signal
- Use ISO 100

Radiance map: assembly

- Raw pre-processing
- PTGui / HDR Shop
- Inverse response curve
- Photometric correction



Raw pre-processing: dcraw

- Raw input without White balance and tone curve

Raw pre-processing: dcraw

- Somewhat corrects chromatic aberrations

Raw pre-processing: dcraw

- Superior to Camera Raw and Photoshop
- PTGui uses it but with default params

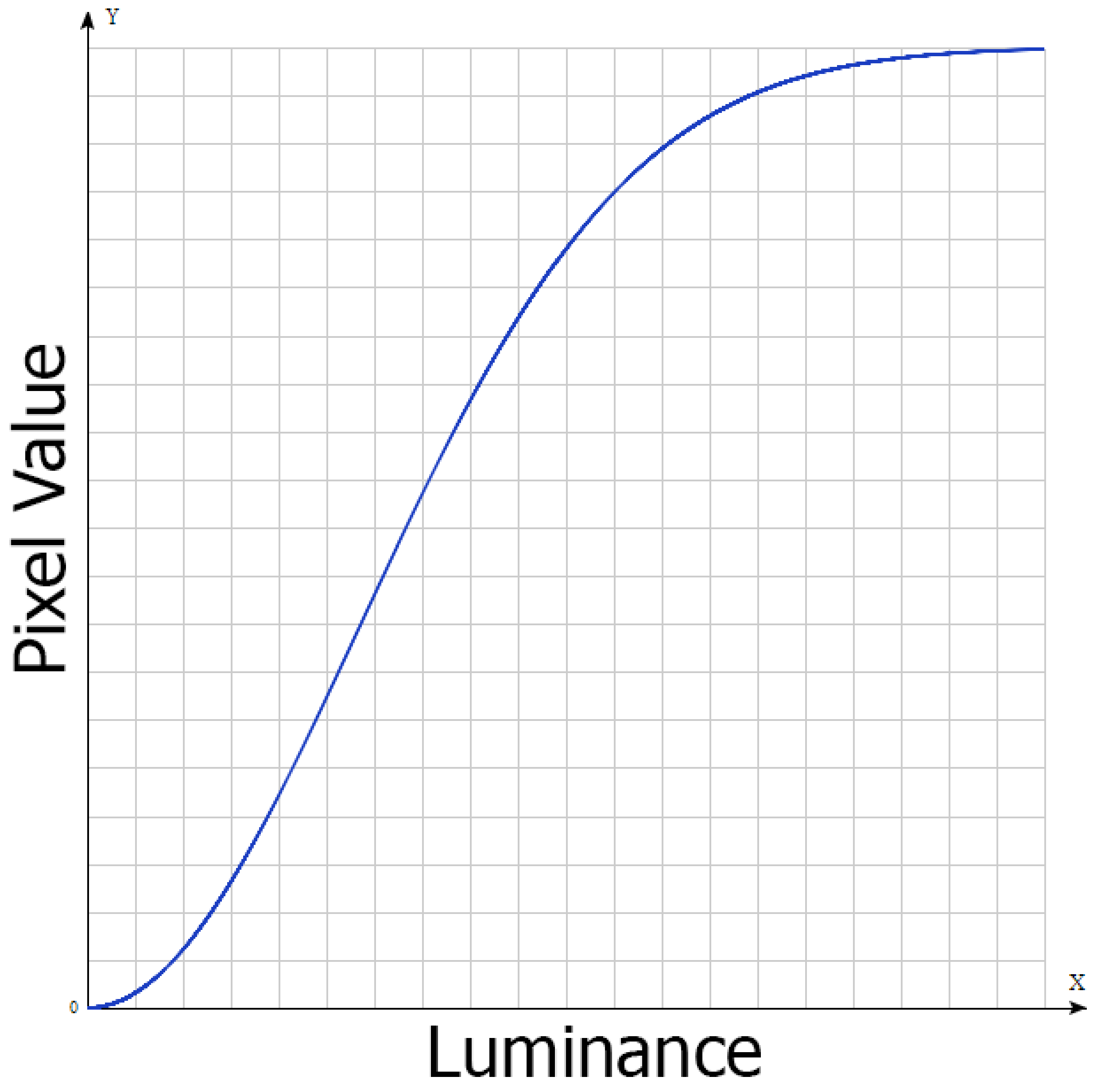
Raw pre-processing: dcraw

- Batch process
- Params: -6 -W -v -T -C 0.9992 0.9998
- Output is 16-bit TIFF in gamma
- Do not edit in Photoshop

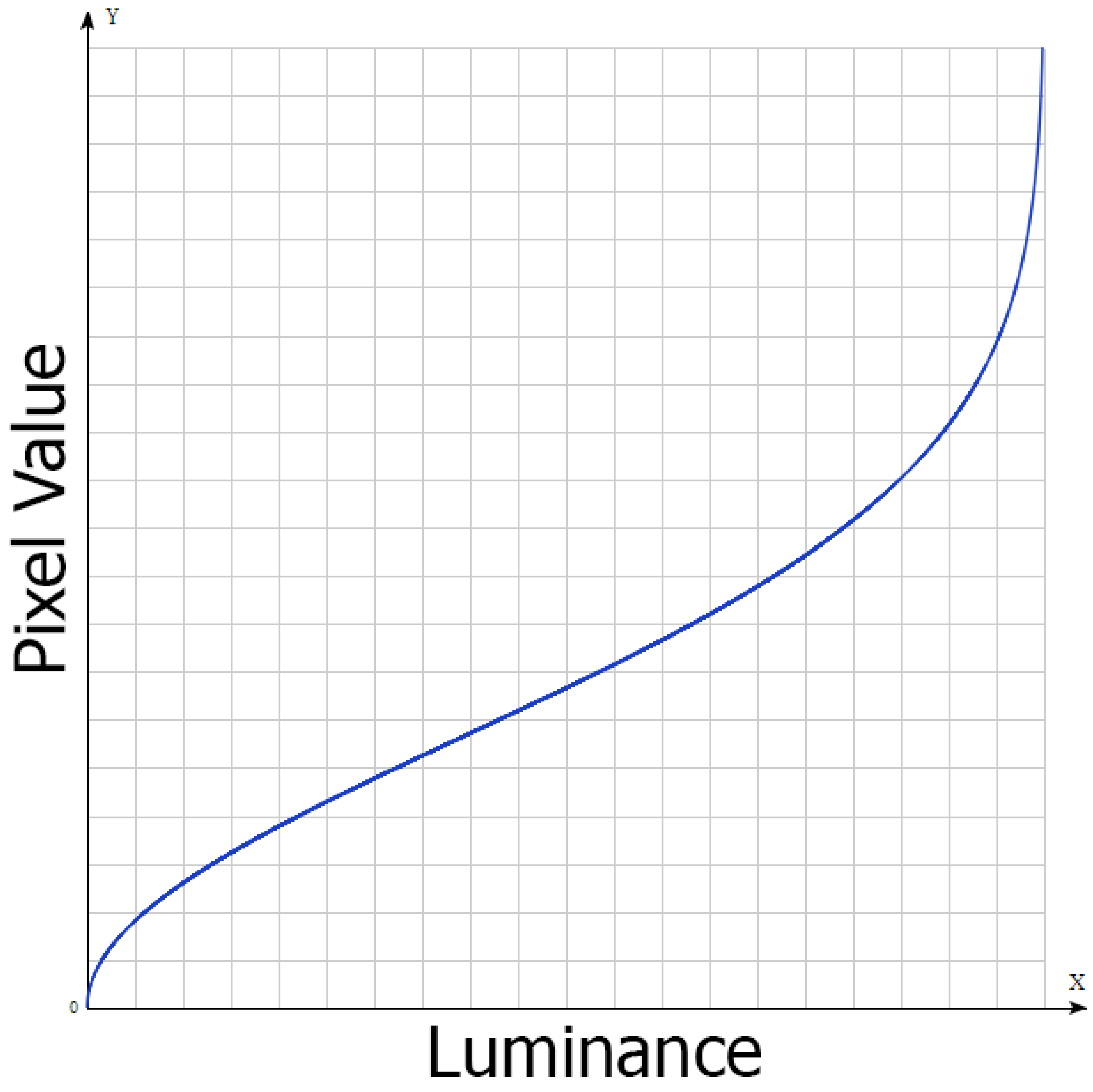
PTGui

- Software for panorama stitching and HDR assembly
- Use True HDR option
- Use pre-defined inverse response curve

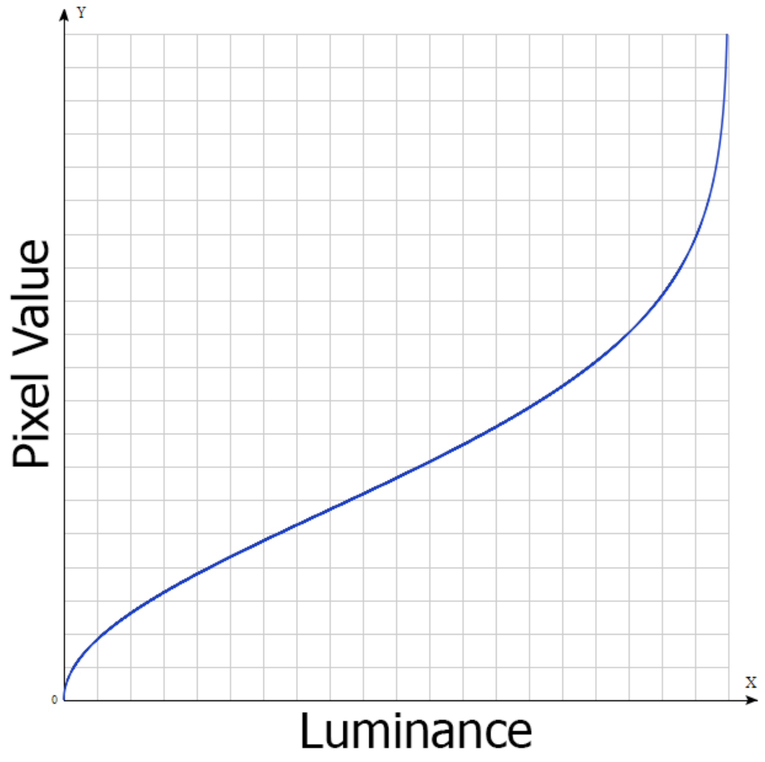
Response curve



Inverse Response curve

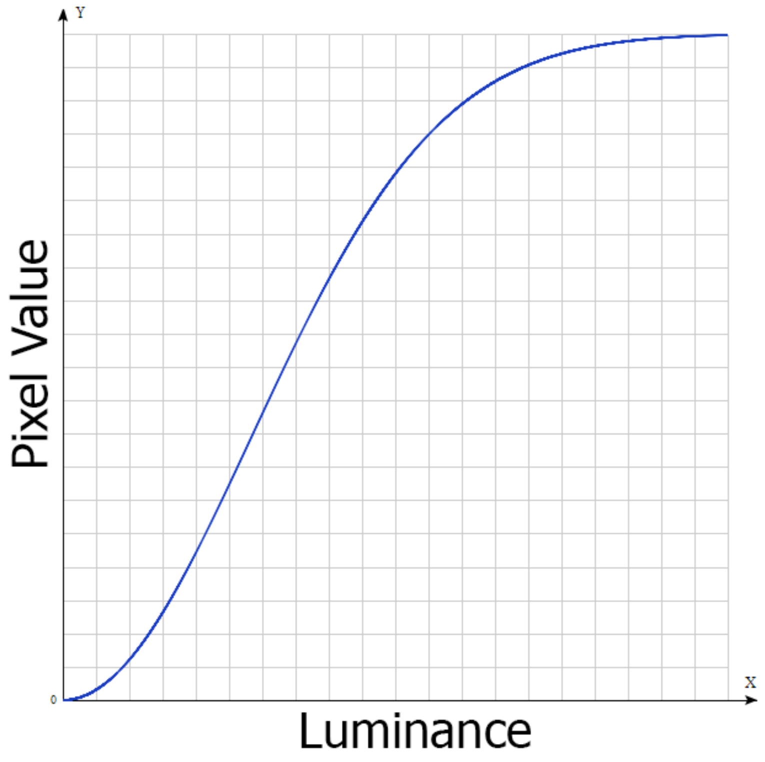


Inverse response curve



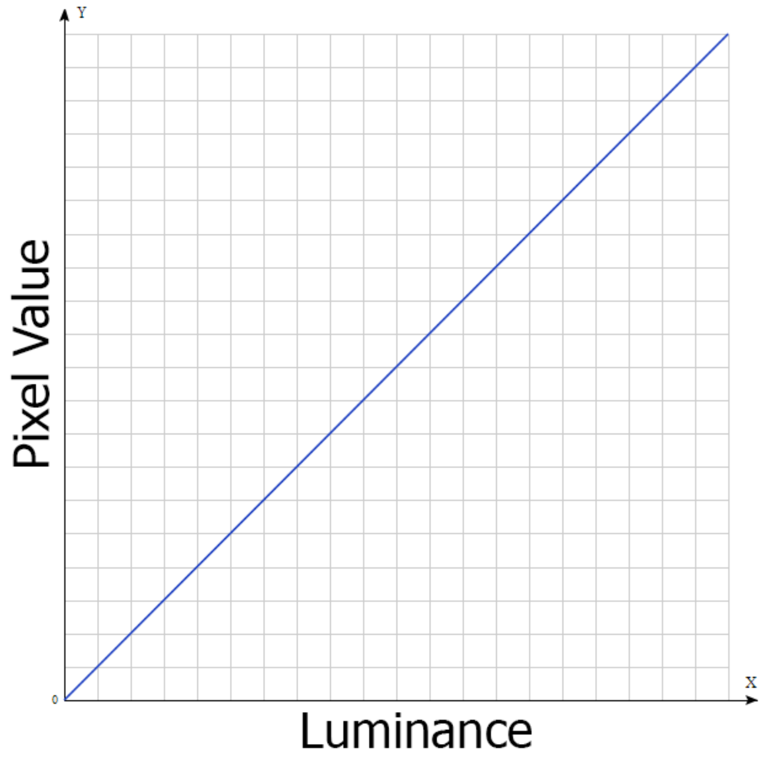
IRF(

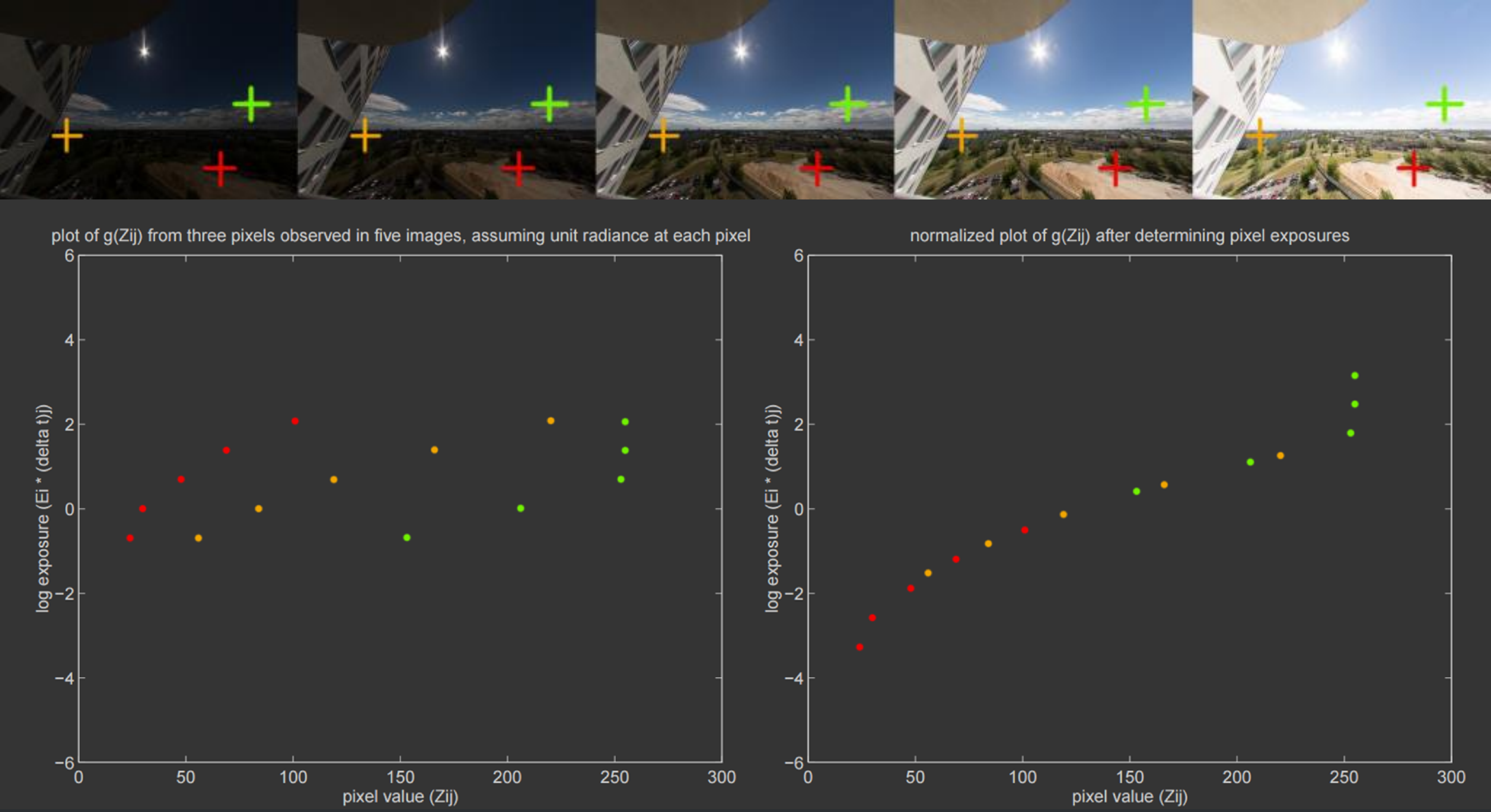
Response curve



) =

Linear

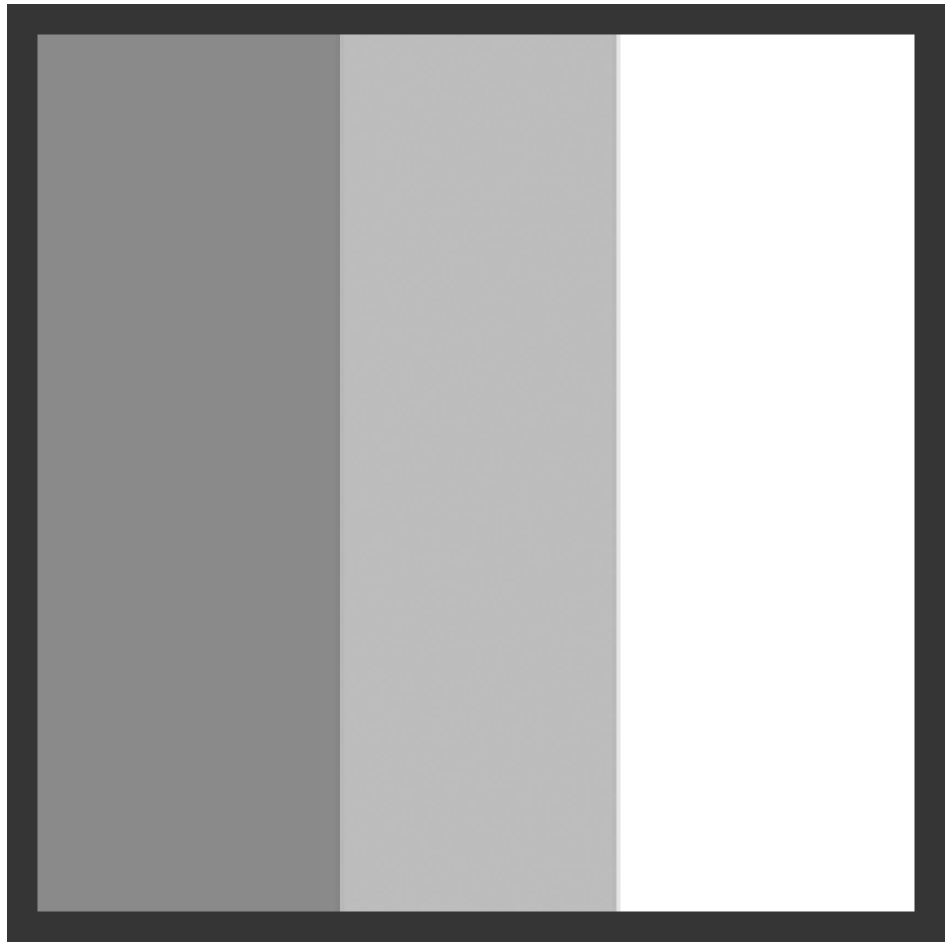




plot of g(Zij) from three pixels observed in five images, assuming unit radiance at each pixel
log exposure (Ei * (delta t)j)
pixel value (Zij)
6
4
2
0
−2
−4
−6
0
50
100
150
200
250
300
normalized plot of g(Zij) after determining pixel exposures
log exposure (Ei * (delta t)j)
pixel value (Zij)

Inverse response curve

- Restores linear relation
  between exposure and Luminance
- Capture once for single camera
- Use target for stability ⇨
- Lens and filter are independent



Photometric correction

- Exposure equation
- Units
- Storage
- Sun Luminance

Exposure equation

- Returns luminance from exposure

1. $Lmax = L * \frac{78 * N^2}{ISO * 0.65 * T}$

2. $Lmax = L * 2^{exposure}$

$$exposure = \log_2\left(\frac{78 * N^2}{ISO * 0.65 * T}\right)$$

Second equation is more preferable, because you can apply it as exposure in Photoshop

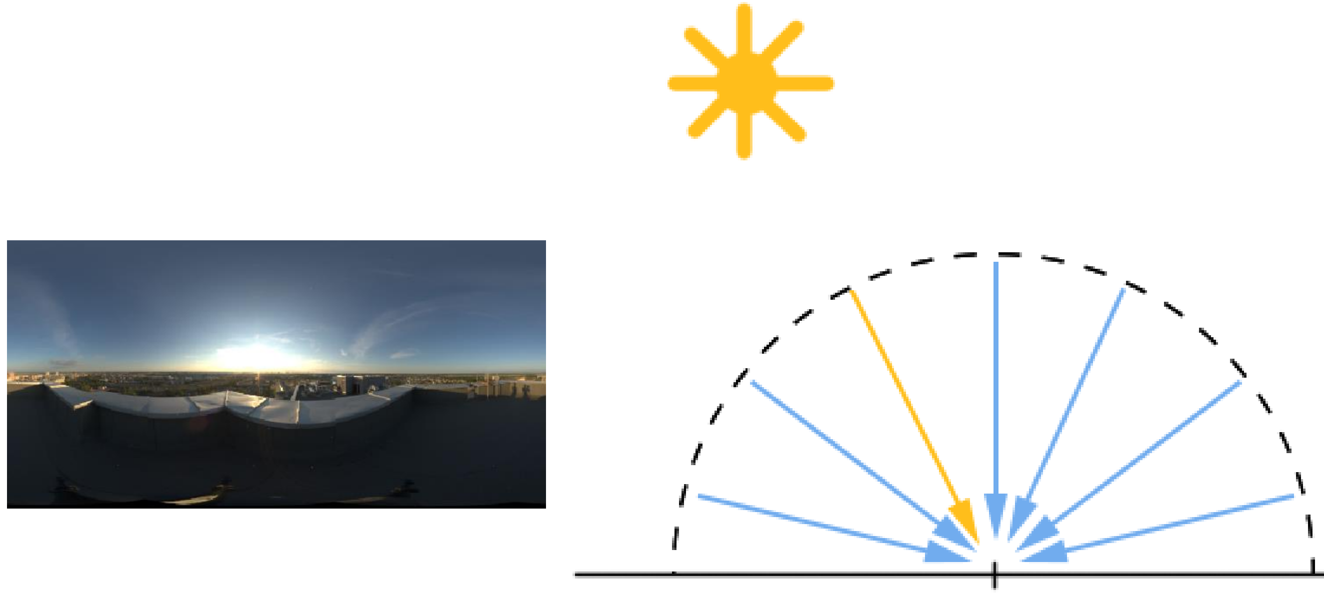
N- aperture, T- shutter speed

Units

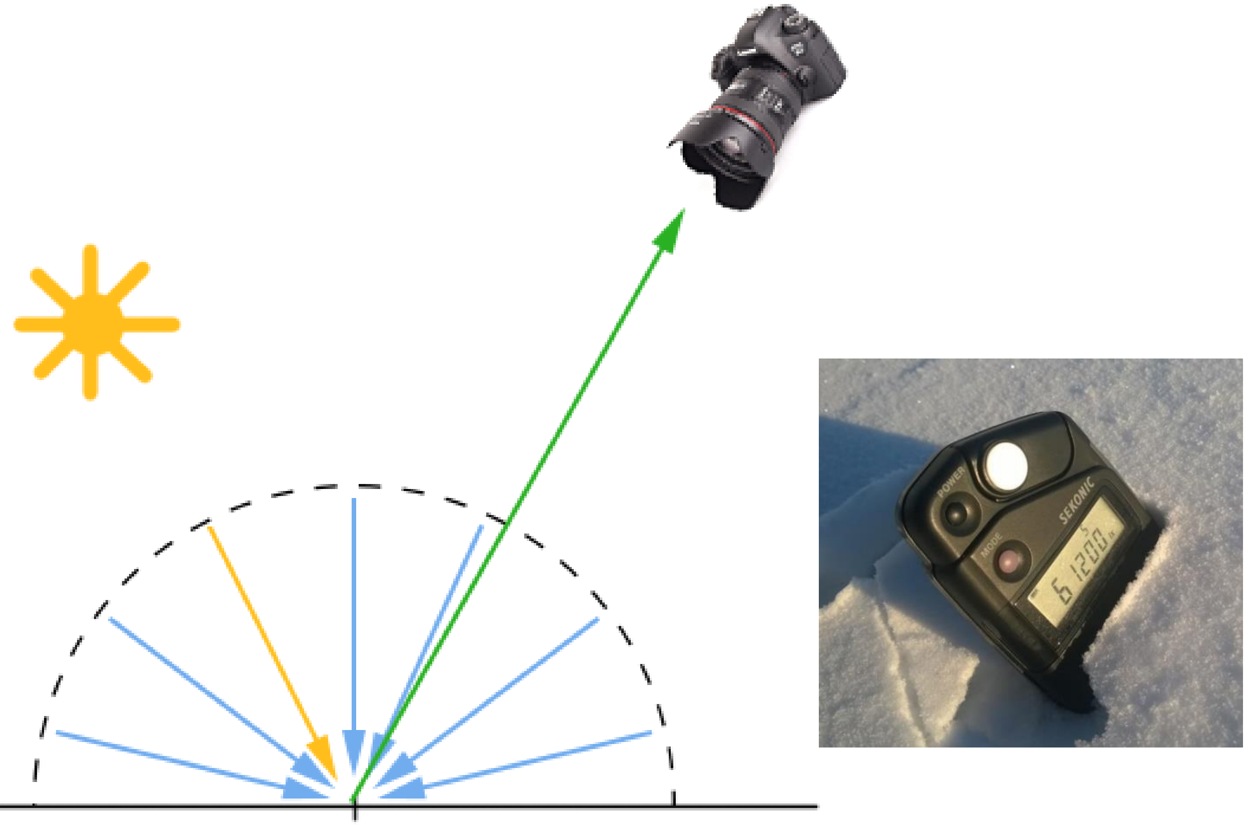
- Unit for luminance is cd/m^2, or Nit
- Good idea to work in 1000*Nit
- Most brightnesses are close to 1
- Unit for Illuminance is Lux

Units: Luminance vs Illuminance

Luminance, Nit



Illuminance, Lux



Units: Illuminance vs Luminance

$$Illuminance = \int_{\Omega} Luminance \cos(\omega i)\, d\omega$$

$$Illuminance = Luminance\ \pi\ sr$$

$$Luminance = \frac{Illuminance}{\pi\ sr}$$

so surface illuminated with *3.14* lux has a luminance of *1* cd/m2 or 1Lux = 1cd/m2/(π*sr)

Storage

- Store in absolute values
- More accurate packing if you can’t use float point formats
- Lighting settings are mostly the same for every environment

Sun luminance: two solutions

- ND filter 😐
- Analytical solution ☺

Sun luminance

1. Measure Illuminance with illuminometer
2. Shoot the white target on the color checker to restore color or luminance if you don’t have hardware
3. Do some math

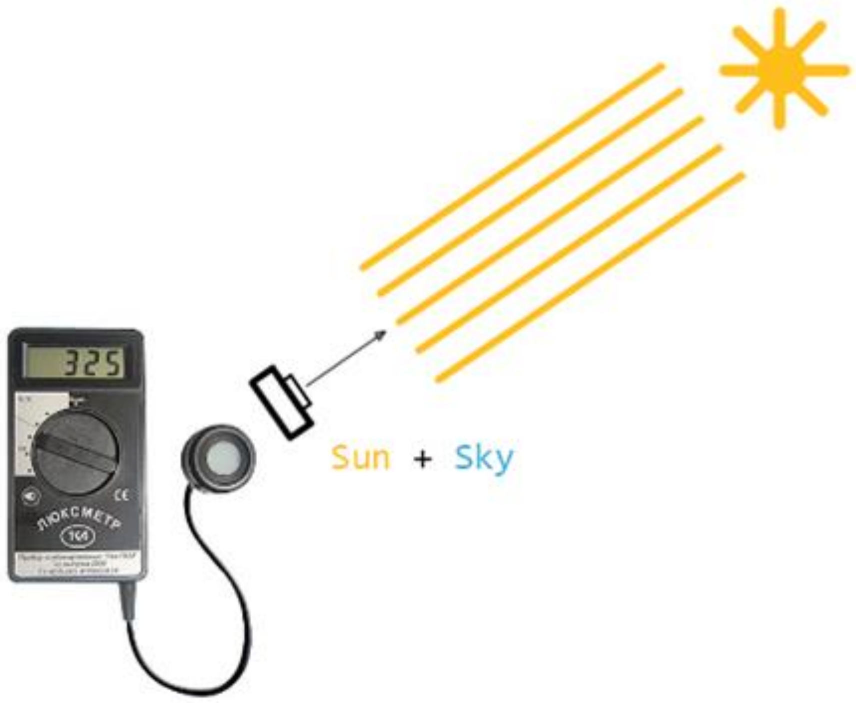
Sun luminance: illuminometer

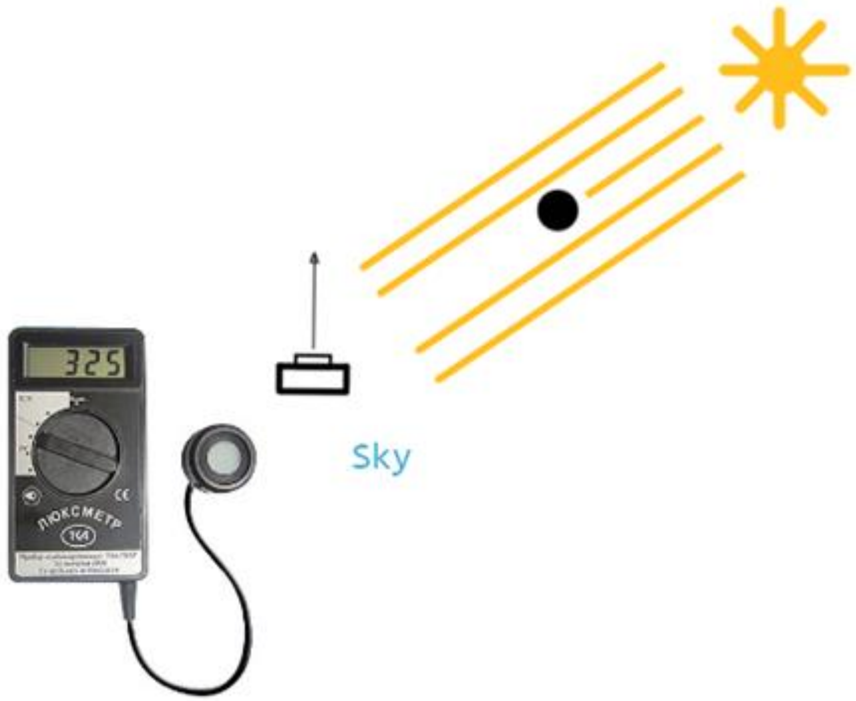
- Measure when sensor
  is normal faced to the Sun
- Use “flat” sensor

A. Faced toward the Sun
the Sun is not occluded

B. Faced toward the Sun
the Sun is occluded

C. Faced up
the Sun is occluded





Sun Illuminance = A - B

C is needed for further check

Sun luminance: restoration

- Just use direct light source with brightness equal to the Sun illuminance or luminance
- If you need exact Sun brightness, it can be restored from Illuminance and Sun angular size

$$Lsun = \frac{Isun}{\pi * 2.13 * 10^{-5}}$$

Sun luminance: restoration

$$Isun = Lsun * \int_{\Omega} SunAreaTest(\omega i)\cos(\omega i)\, d\omega$$

$$Isun = Lsun * \pi * 2.13 * 10^{-5}$$, *for Sun angle =0.53*

$$Lsun = \frac{Isun}{\pi * 2.13 * 10^{-5}}$$

Sun color

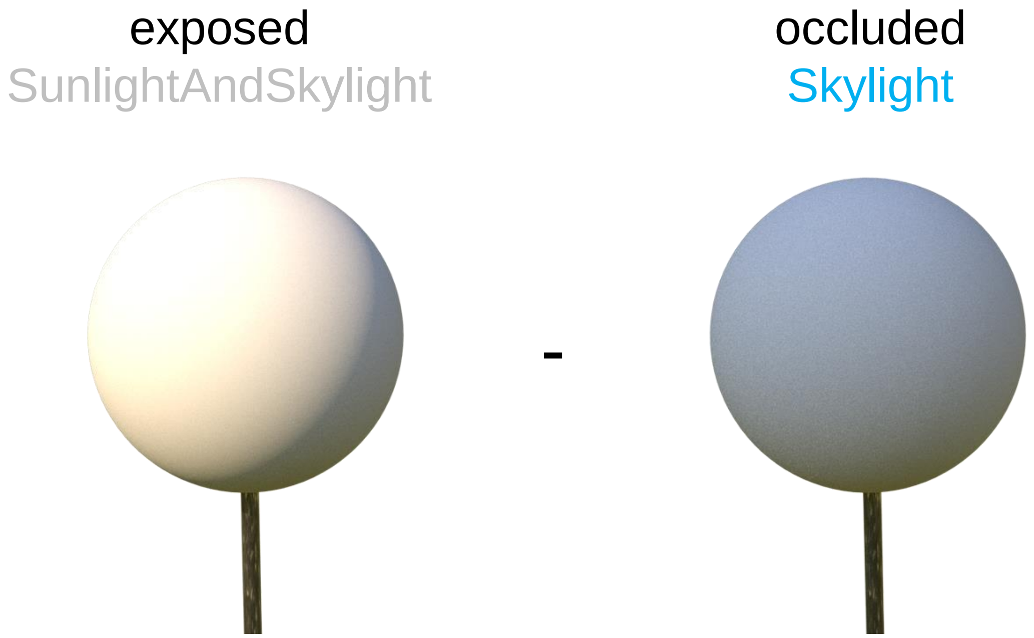
- Shoot white target, occluded and exposed to the sun(same as illuminometer), assemble HDRI, take the difference, normalize.
- Or use atmosphere model with the same conditions
- We used both approaches

- Shoot with the target exposed to the sun

exposed

SunlightAndSkylight

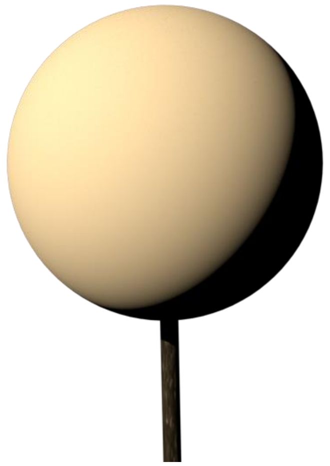
- Shoot with the target exposed to the sun
- Shoot with the sun occluded by some small object to get skylight



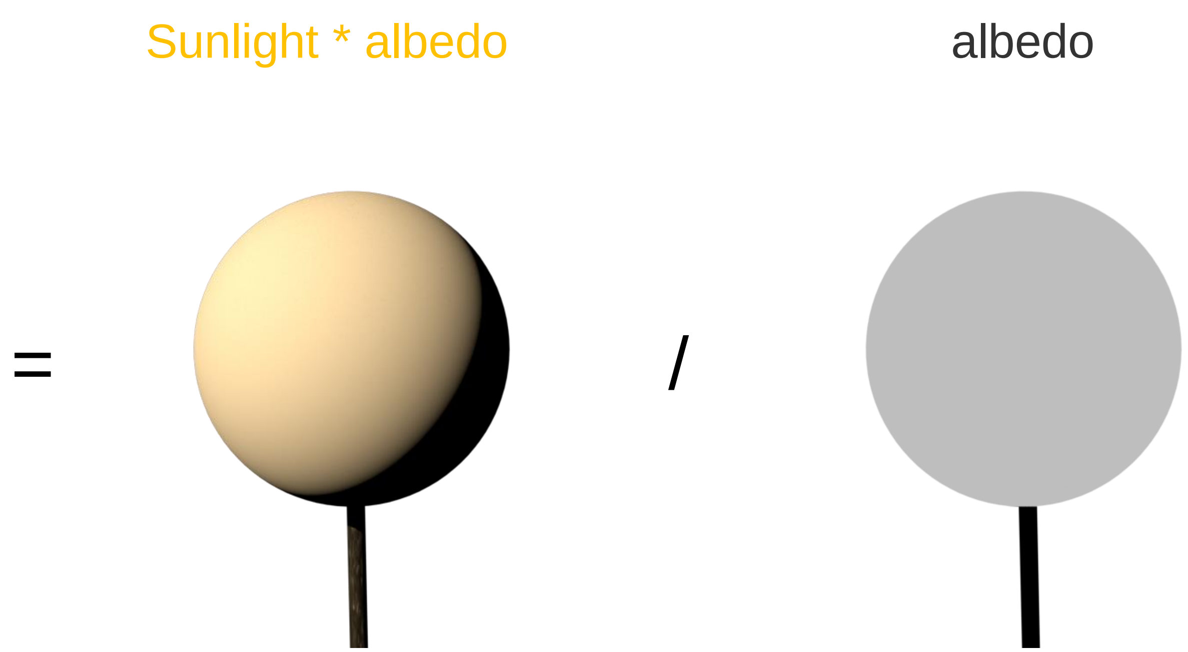
- Shoot with the target exposed to the sun
- Shoot with the sun occluded by some small object to get skylight
- Subtract the second from the first

Sunlight * albedo

=



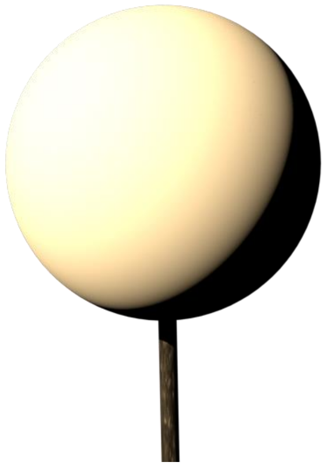
- Shoot with the target exposed to the sun
- Shoot with the sun occluded by some small object to get skylight
- Subtract the second from the first
- Divide by albedo



- Shoot with the target exposed to the sun
- Shoot with the sun occluded by some small object to get skylight
- Subtract the second from the first
- Divide by albedo
- Get your Illuminance

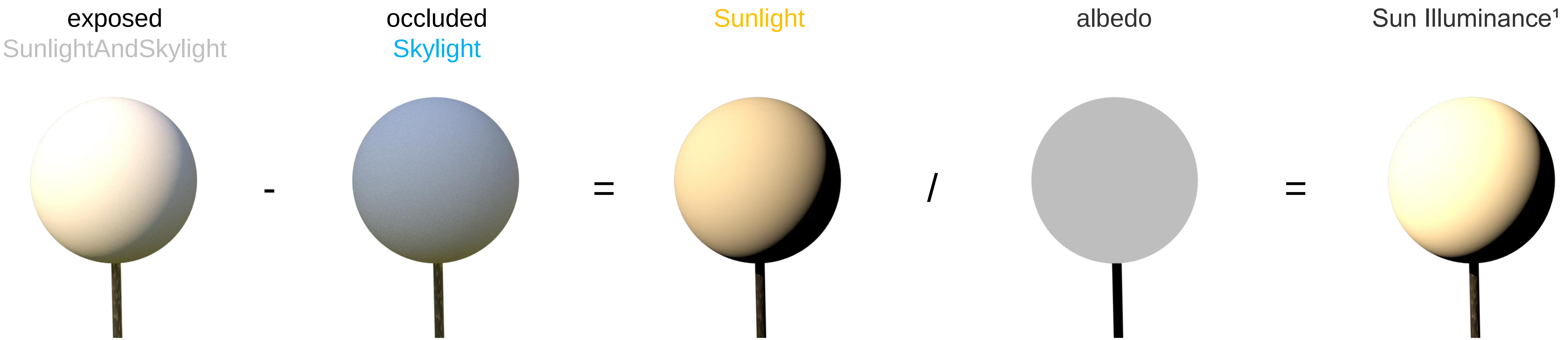
Sun Illuminance[1]

=



1. What you get is actually the luminance of the point illuminated by the Sun, Sun illuminance in Lux would be π times greater

- Shoot with the target exposed to the sun
- Shoot with the sun occluded by some small object to get skylight
- Subtract the second from the first
- Divide by albedo
- Get your Illuminance



1. What you get is actually the luminance of the point illuminated by the Sun, Sun illuminance in Lux would be π times greater

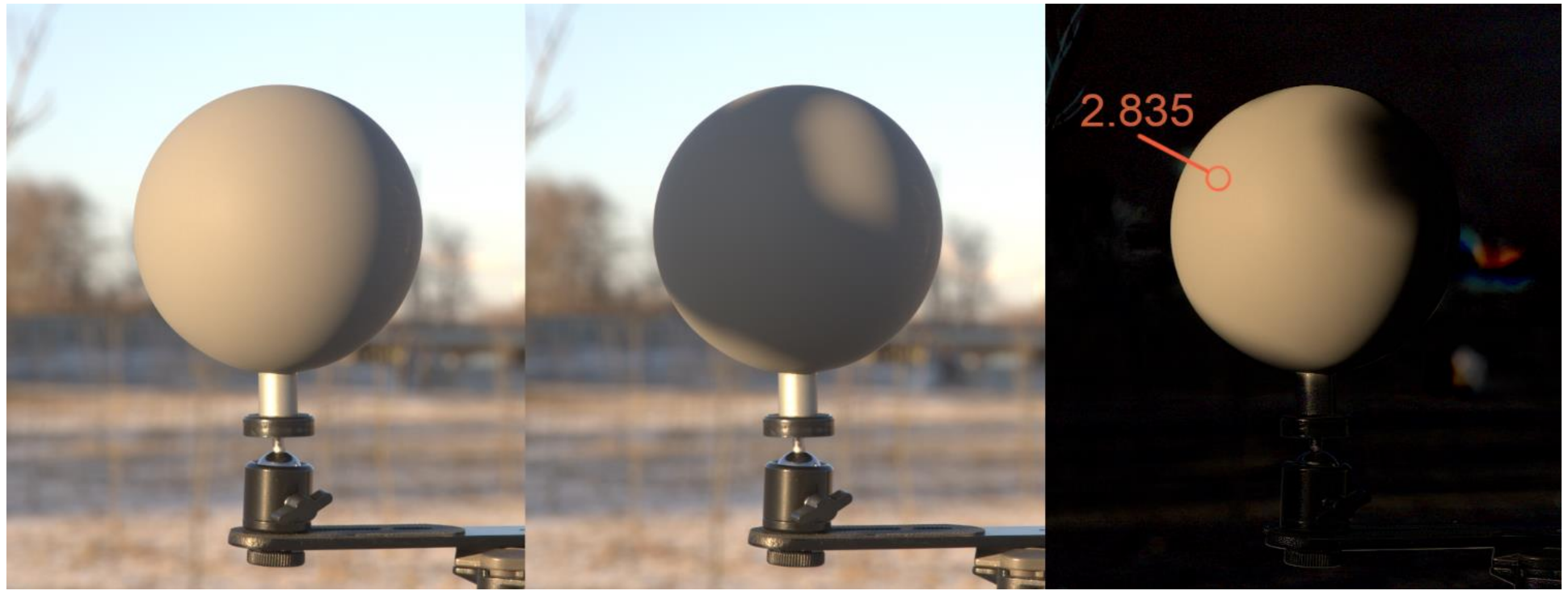
In practice

- Photometric correction
- Sun: from target
- Sun: from measurement
- Sun: from model
- Validation

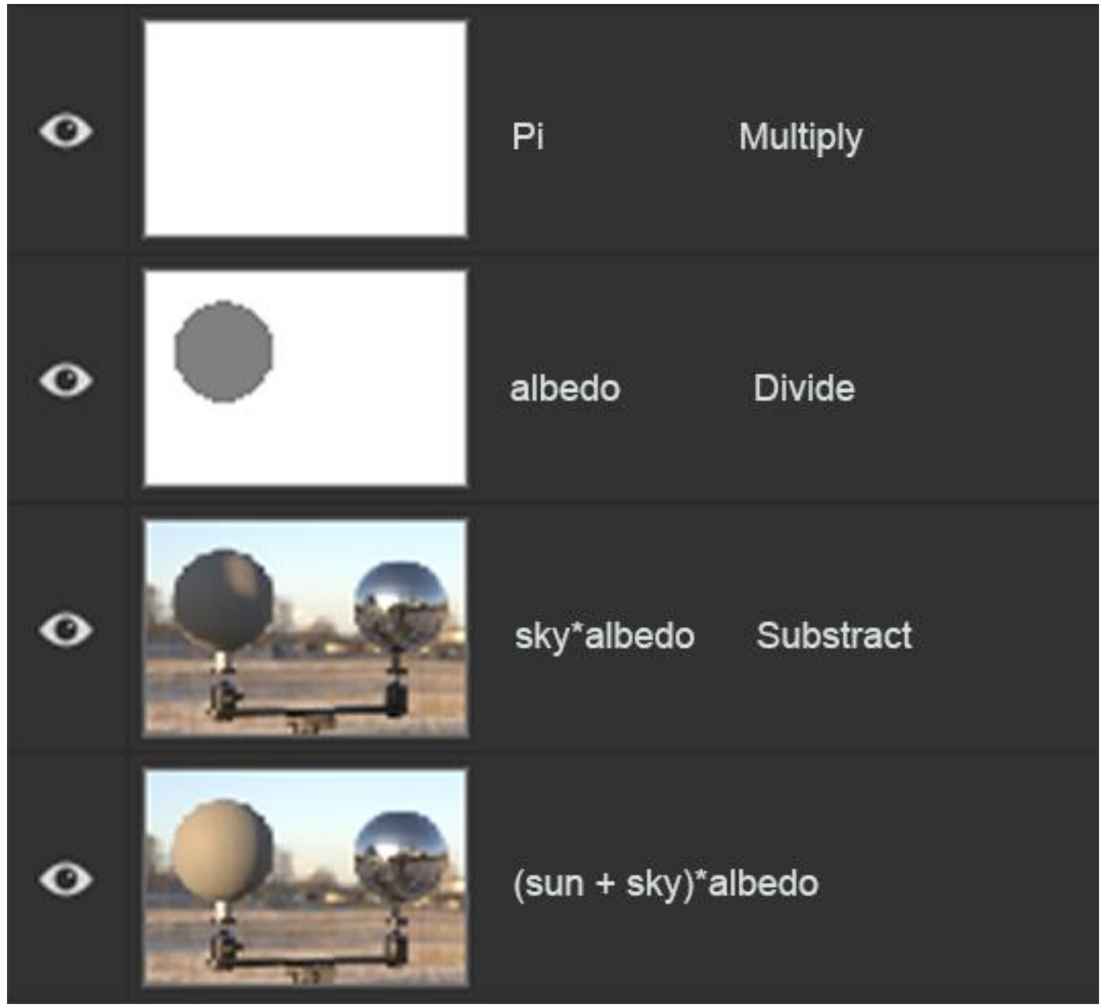
Photometric correction

- N = 8, T = 1/160s
- The final correction:

$$EV = Log_2 \frac{1}{1000} \frac{1.2 * N^2}{T} = 3.619$$



If you don't like numbers you can do all the math in Photoshop 32bit color mode ☺



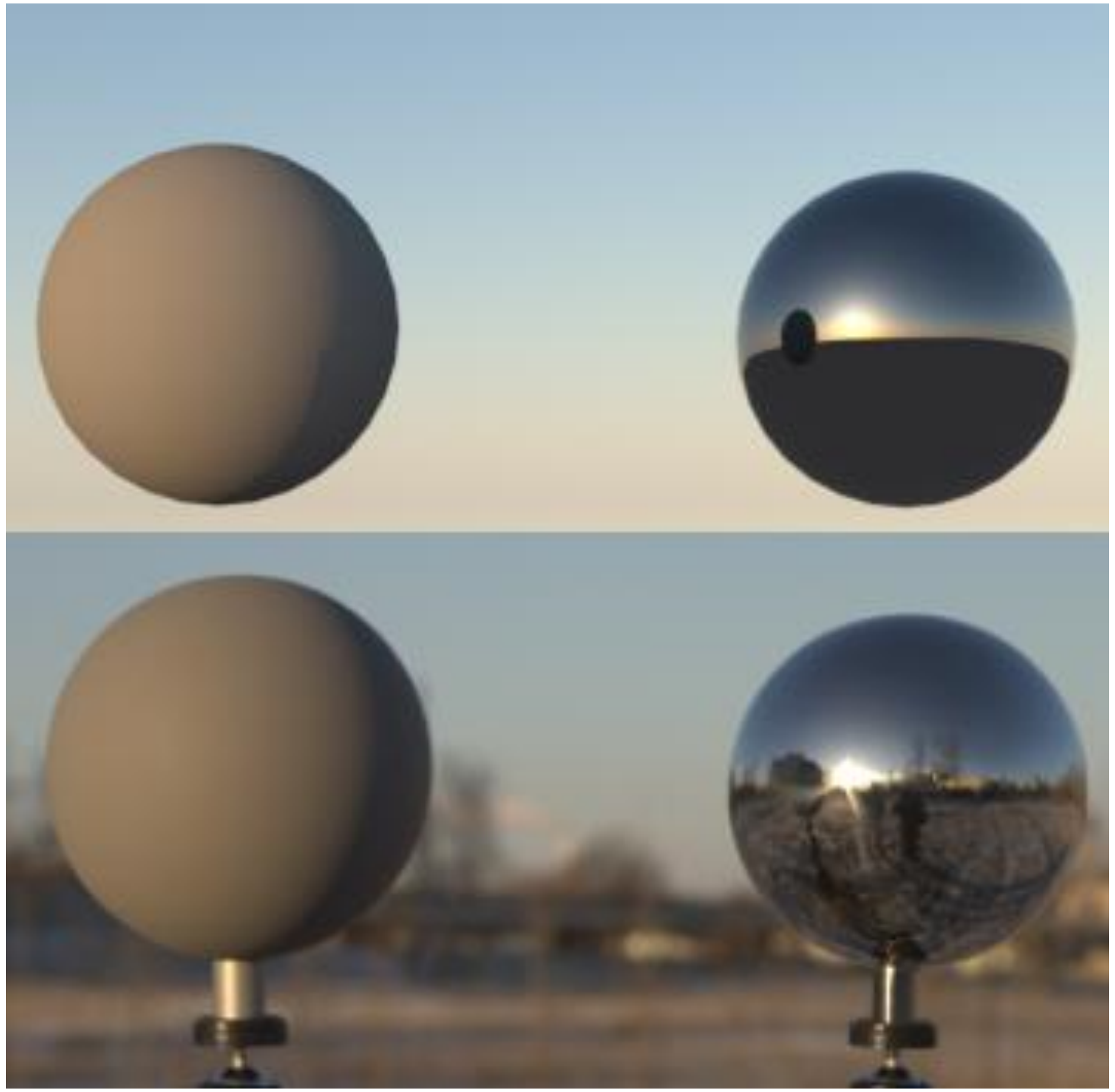
- Target's albedo = 0.5 sRGB or 0.2176 Linear
- Luminance towards Sun = 2.835×1000 Nit
- Sun Illuminance = $\frac{2835 * \pi}{0.2176}$ = 40930 Lux
- Do the same for all RGB channels to get color

- Pretty straightforward ☺
- You can’t restore color unless you have proper RGB filters ☹
- To restore color you still require a target

Sun Illuminance = 44700 Lux – 5680 Lux = 39020 Lux

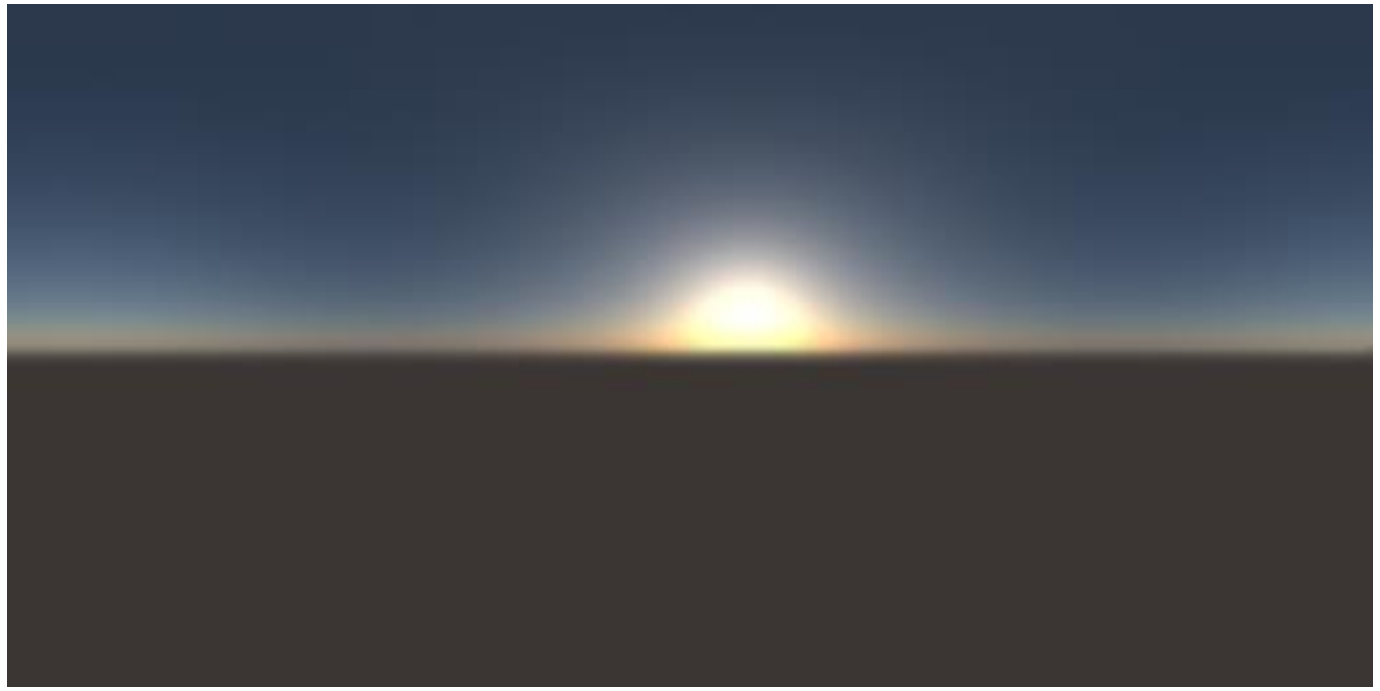
- Geo data and exposure from EXIF
- Use any Photometric sky model(I used Maxwell)
- Sun Illuminance = $\dfrac{2659 * \pi}{0.21076}$ = 39620 Lux
- Sun Color = 255, 199, 130

Method	Sun Illuminance(Lux)	Sun Color
Target Photo capture	40930	255, 204, 144
Measurement	39020	?
Sky model	39620	255, 199, 130



Sky model

(Maxwell Render)



Captured Sky

Summary

Pros:

- ✓ It works!
- ✓ Data proven
- ✓ Perfect reference
- ✓ Photographic lighting setup
- ✓ Process is adapted to scanning and capture
- ✓ Unification

Cons:

- × Static
- × Image quality is not enough
- × Bad artistic control
- × Geography problems☺

Radiance map as a reference

Radiance map as a reference

Questions?