

Lab #2 - Imaging

Informática Gráfica

Adolfo Muñoz - Julio Marco
Pablo Luesia - J. Daniel Subías – Óscar Pueyo



Before we begin...

- The problem:



Your final work's output
(without post-processing)

Before we begin...

- The problem:



Your final work's output
(without post-processing)

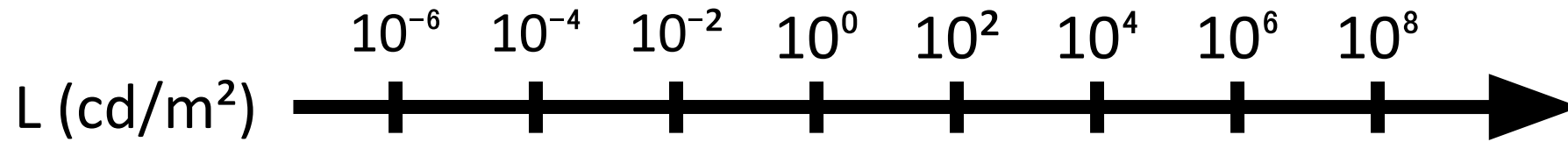


After tone mapping
(this session work!)

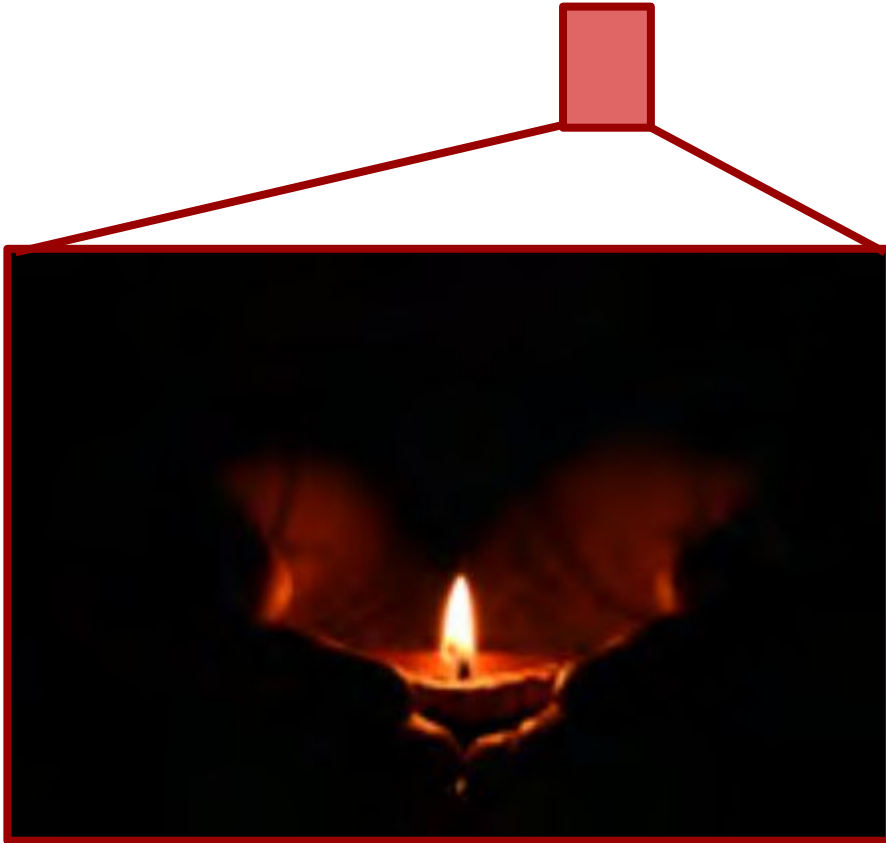
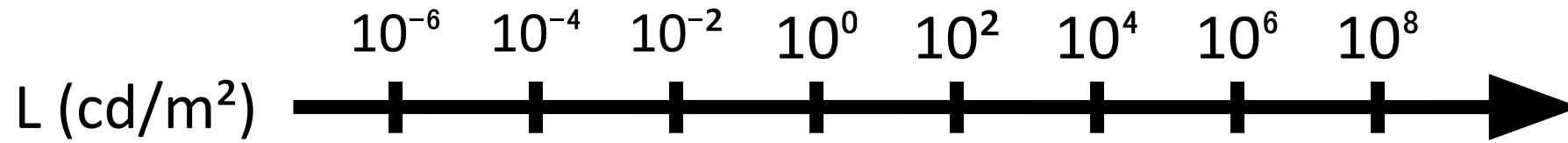
Before we begin...

- Intermediate assignment, again
 - No submission required
- To be completed before **October 2nd (without extensions)**
 - **Highly recommended**
- Remember: Final work is 80% of the final grade

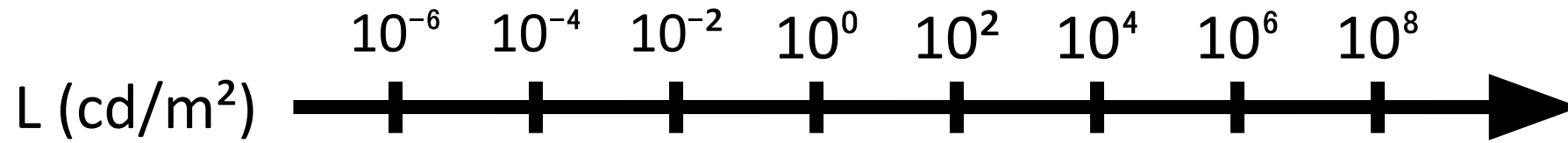
Problem statement, again



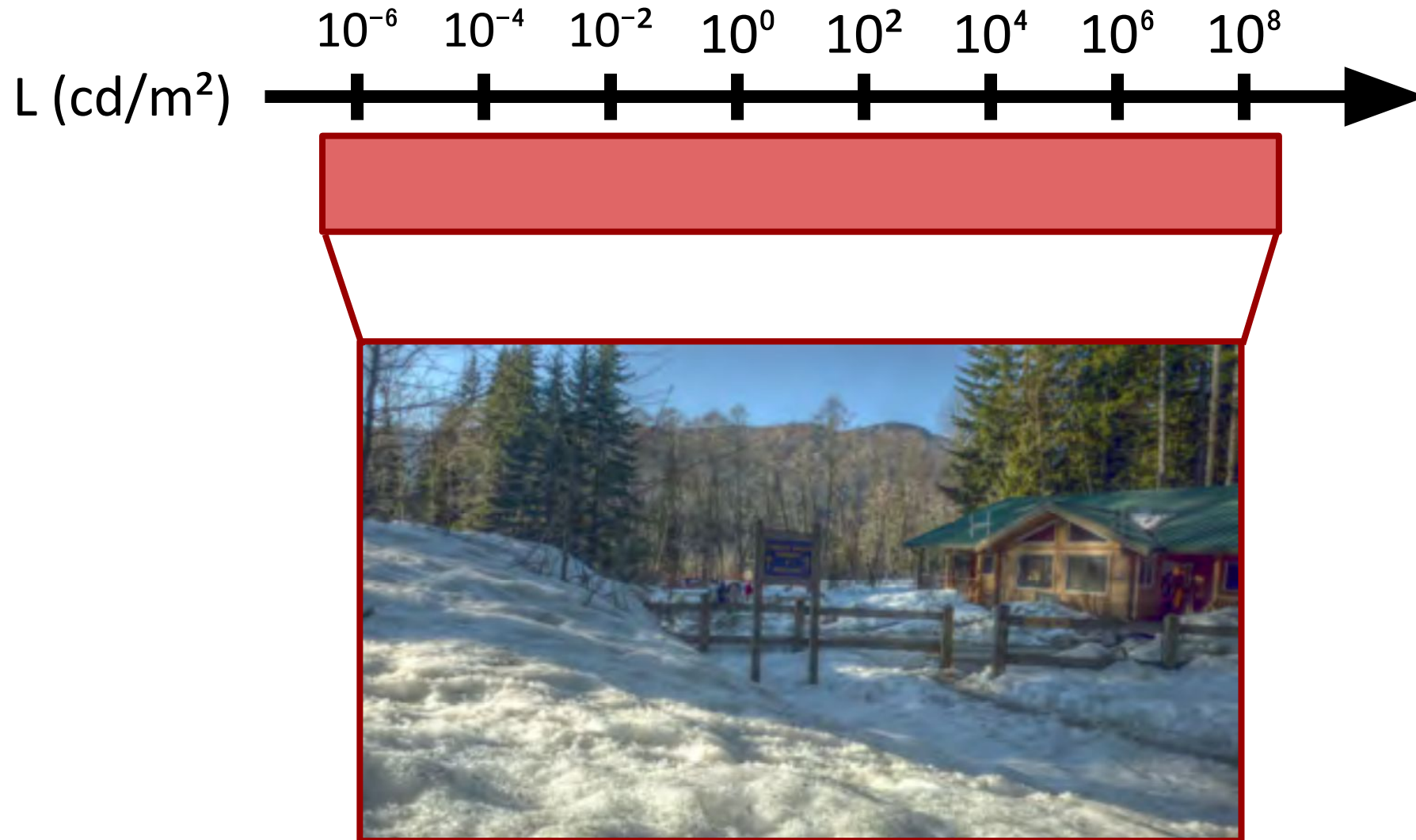
Problem statement, again



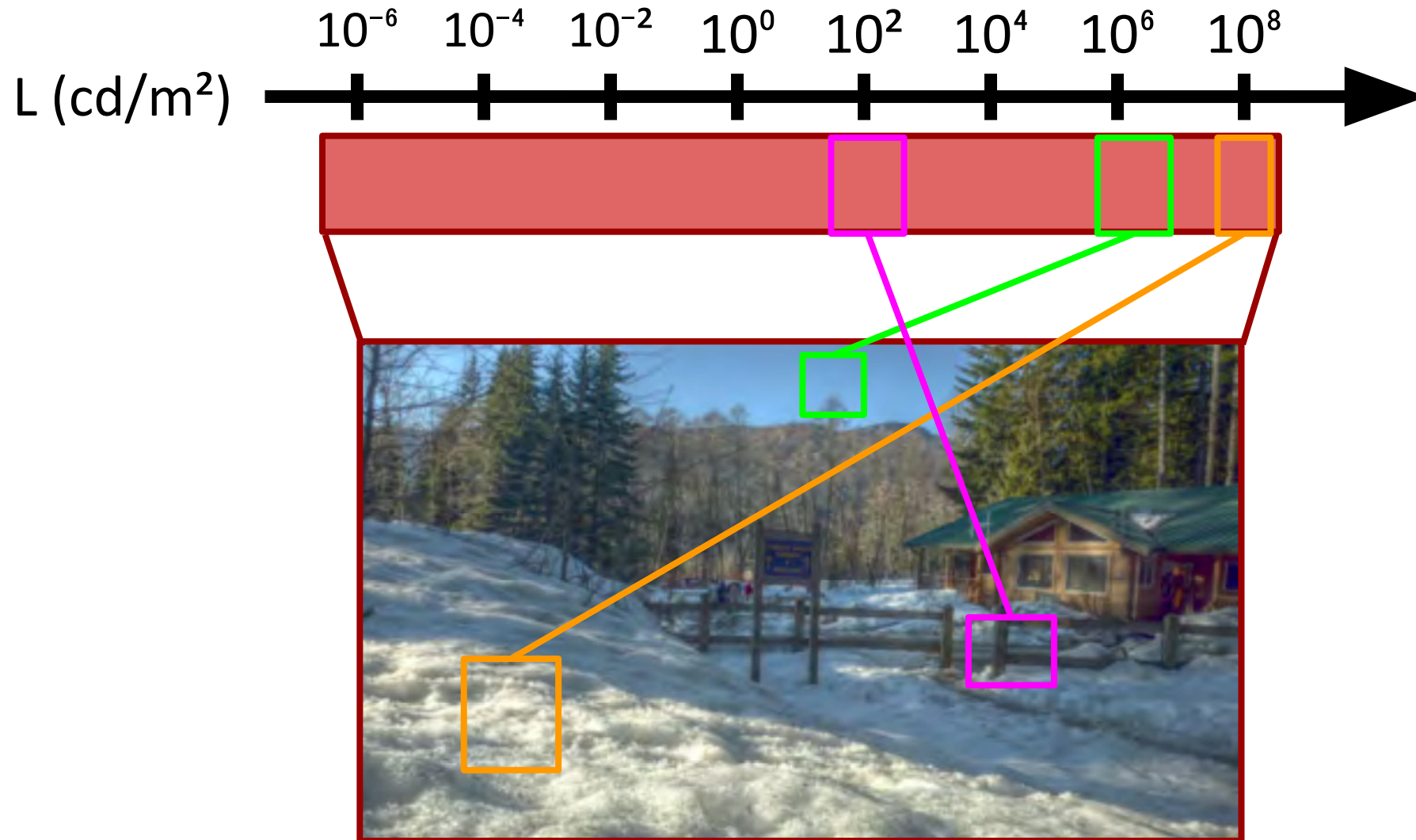
Problem statement, again



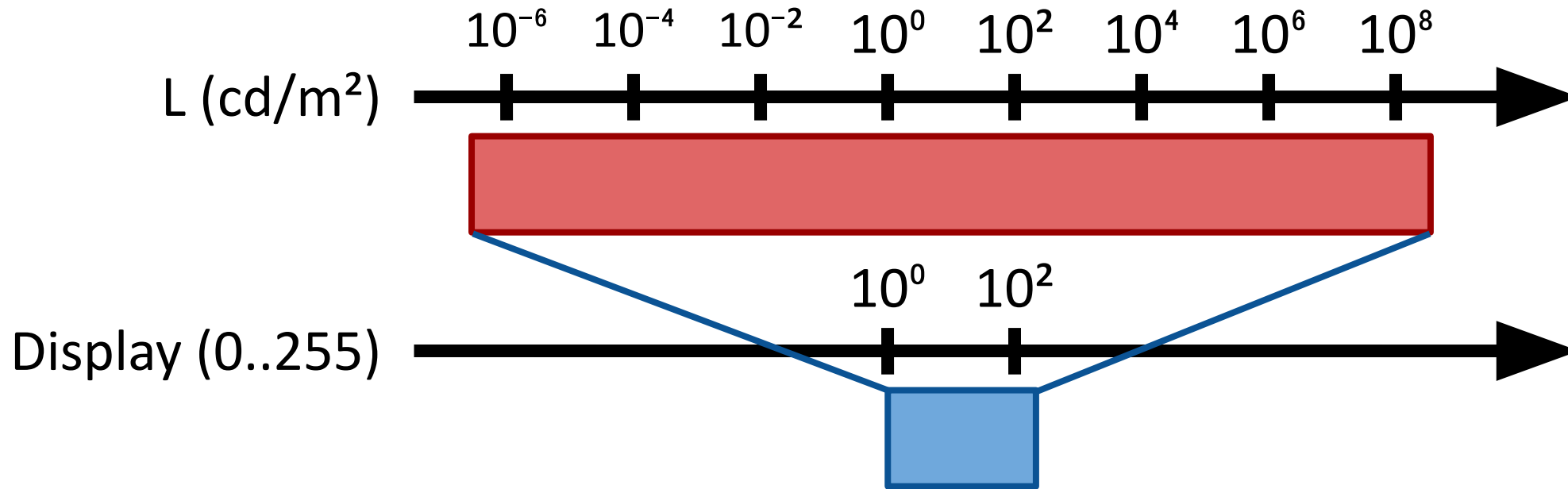
Problem statement, again



Problem statement, again

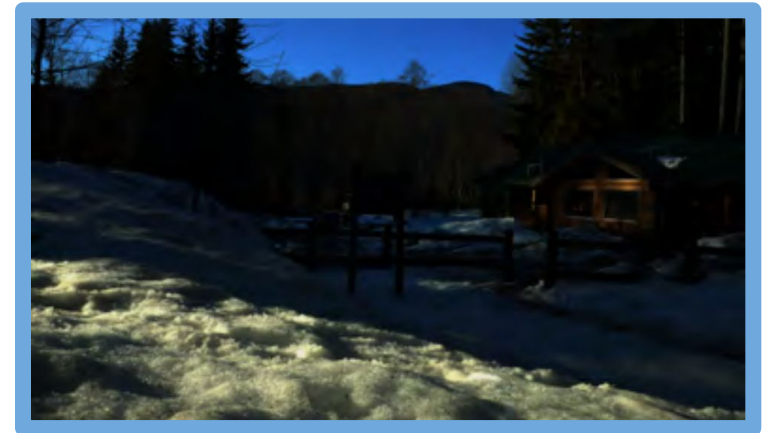
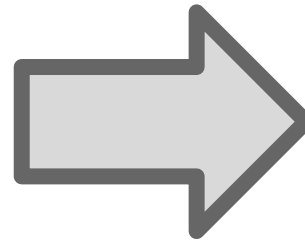
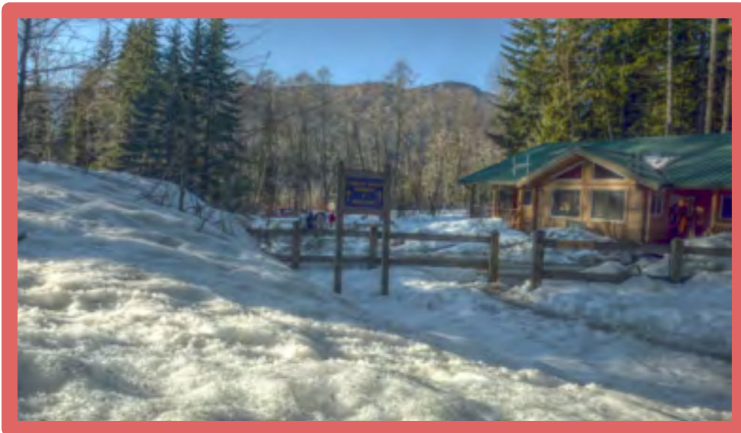
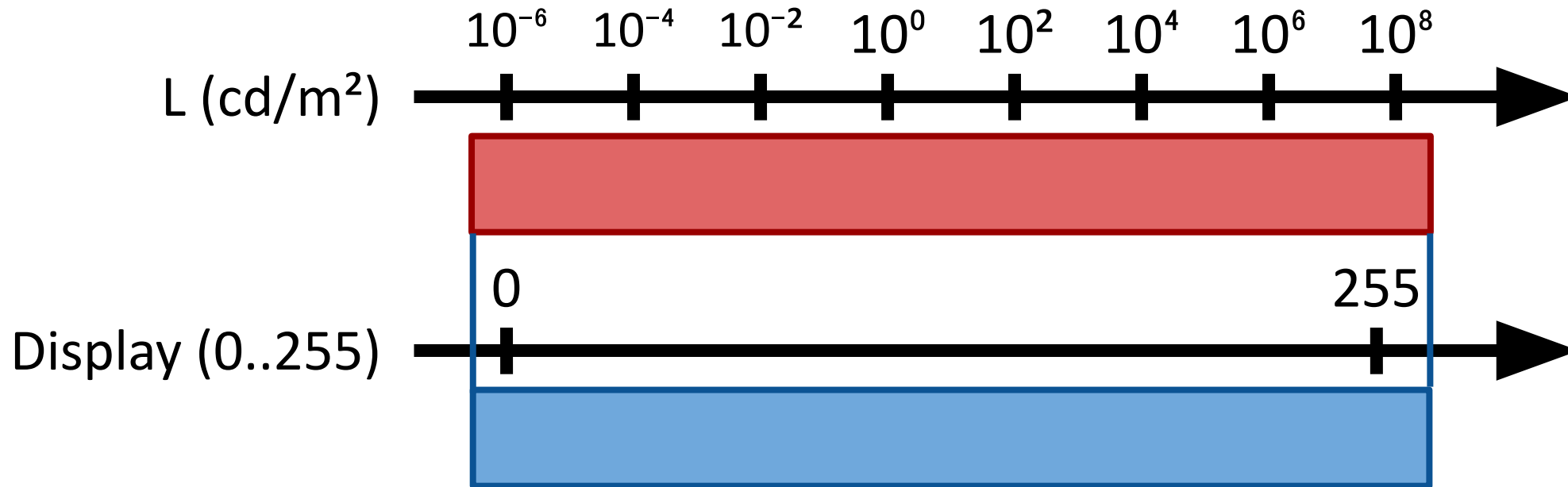


Problem statement, again

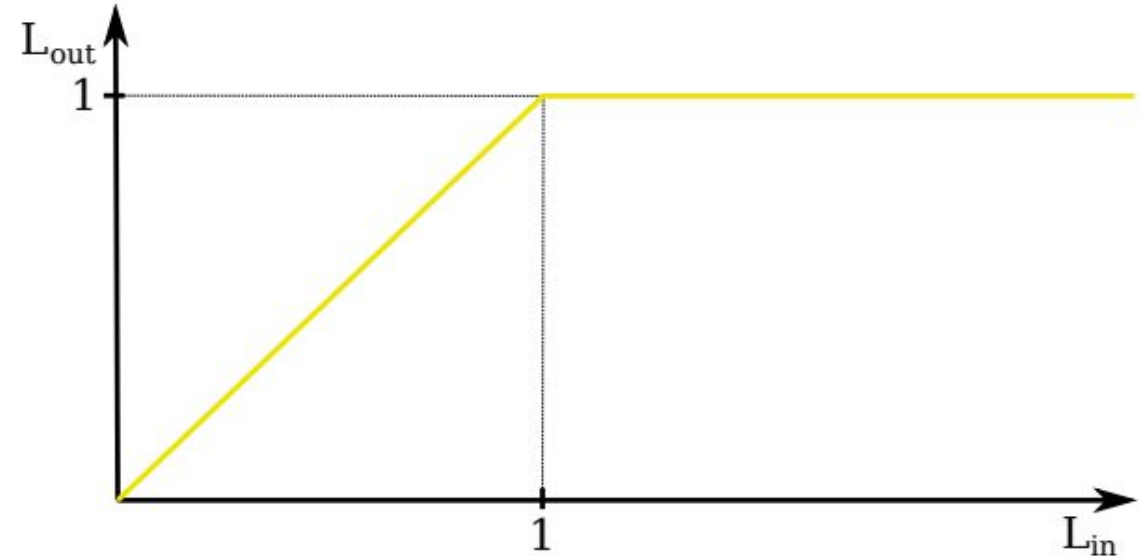
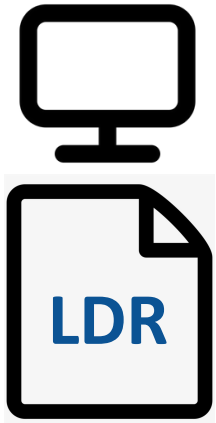


- A (typical) display only has 255 possible values for each color channel
- How to map the **red values** to the **blue values**?

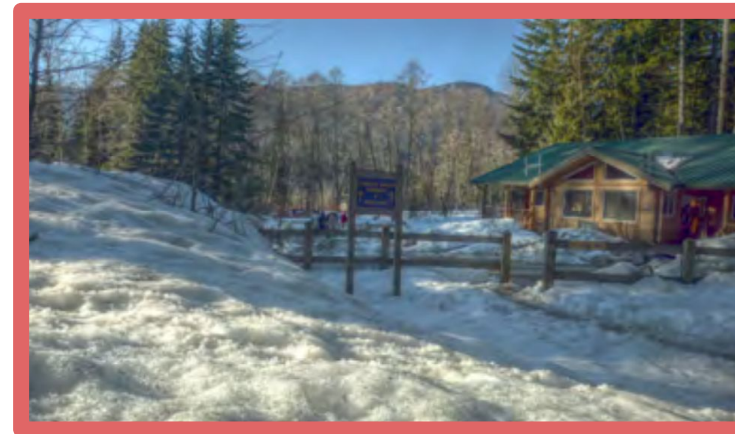
Problem statement, again



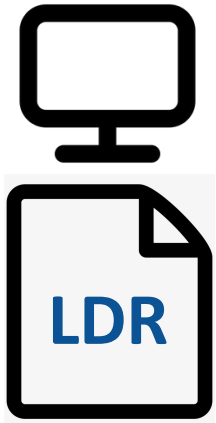
Tone mapping operators



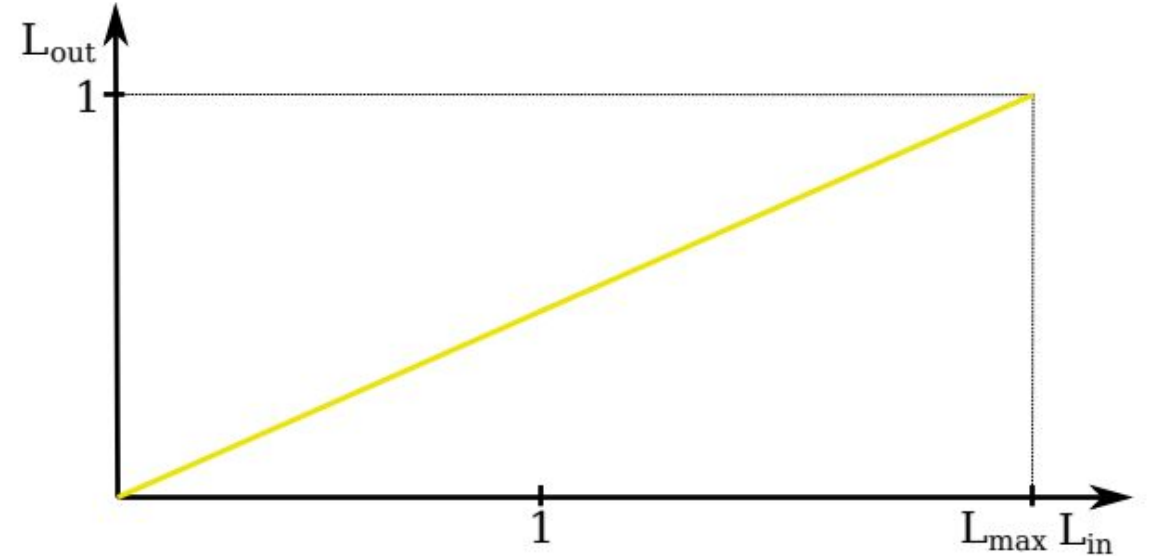
Clamping operator



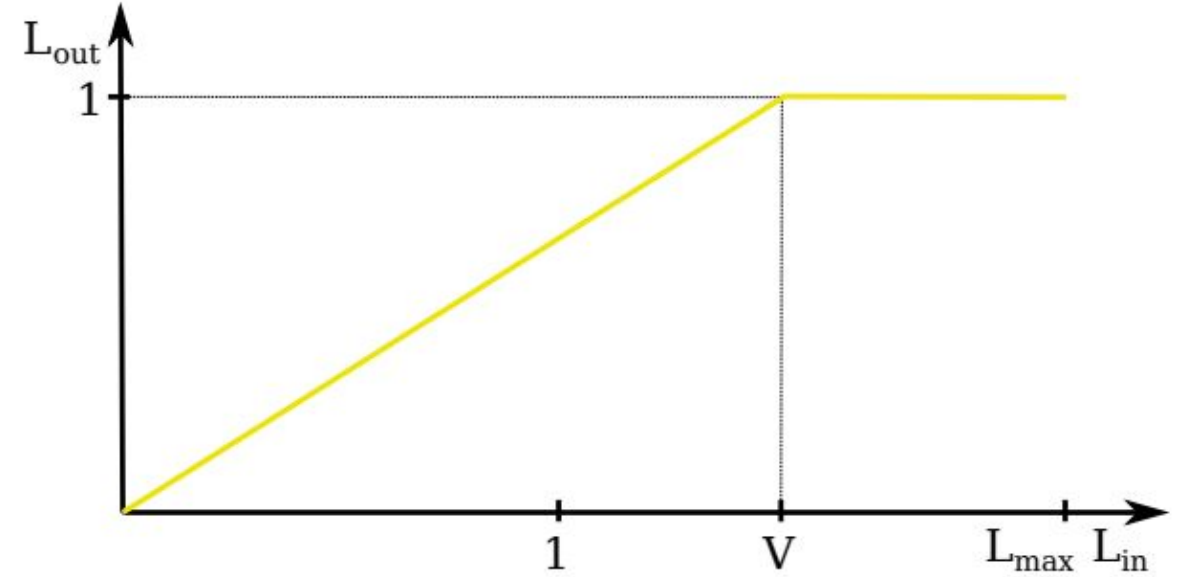
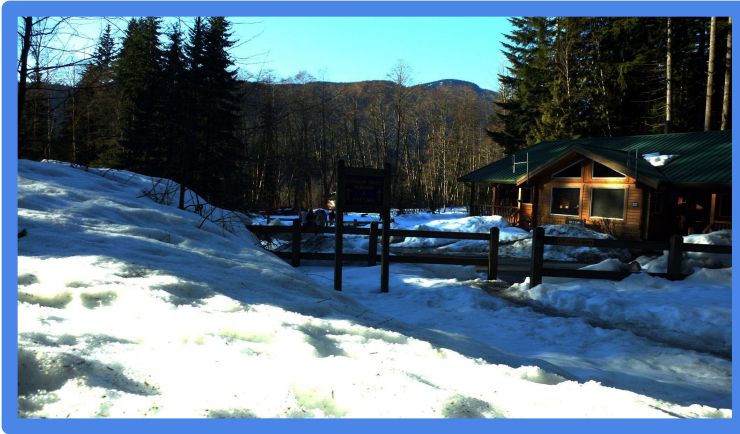
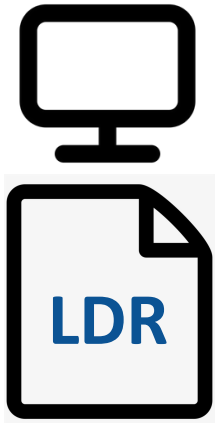
Tone mapping operators



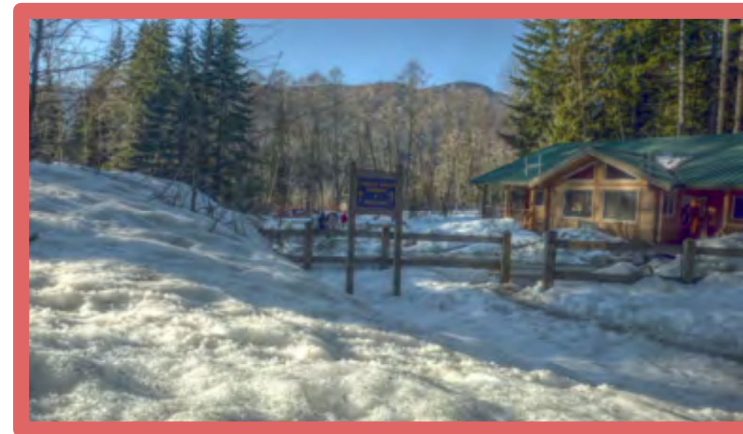
Equalization operator



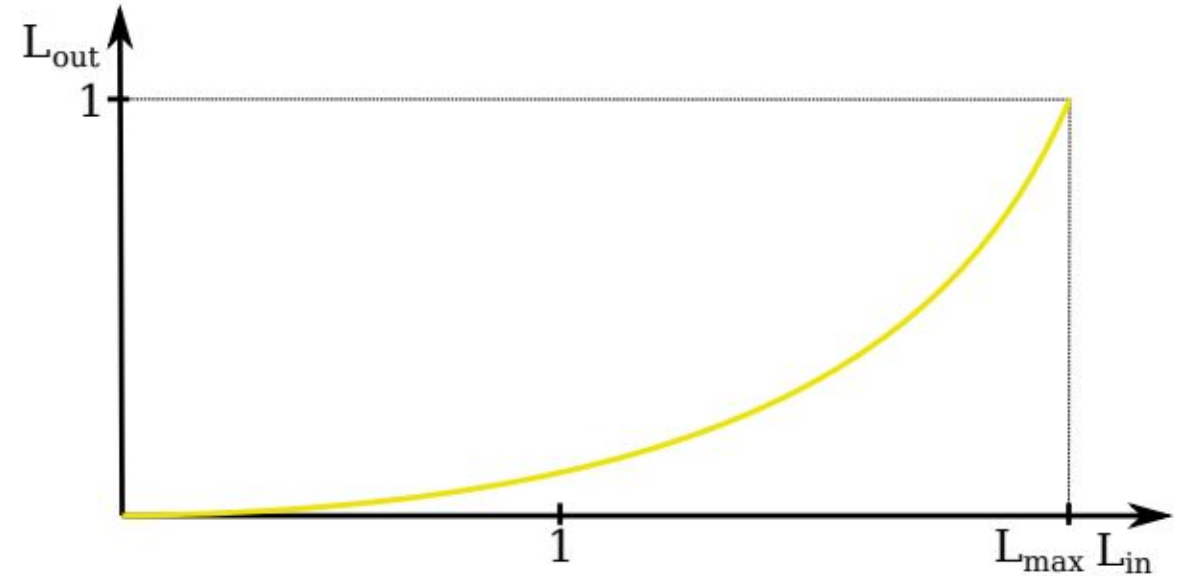
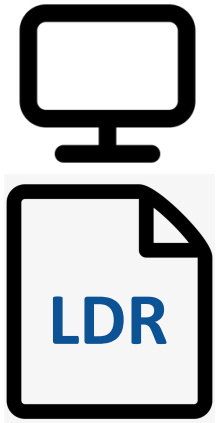
Tone mapping operators



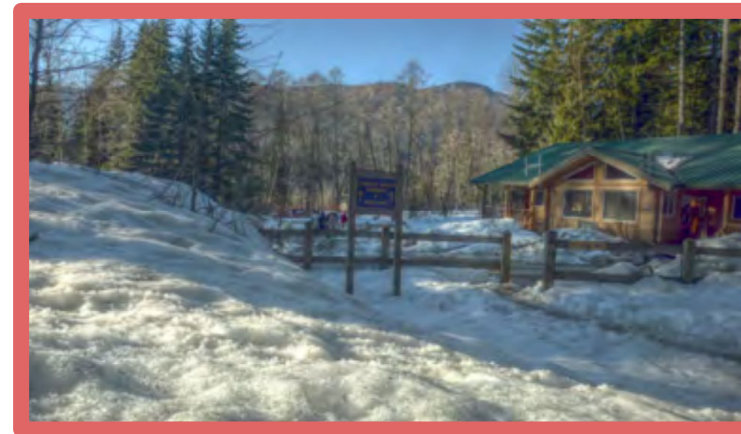
(Equalization + clamping)
operator



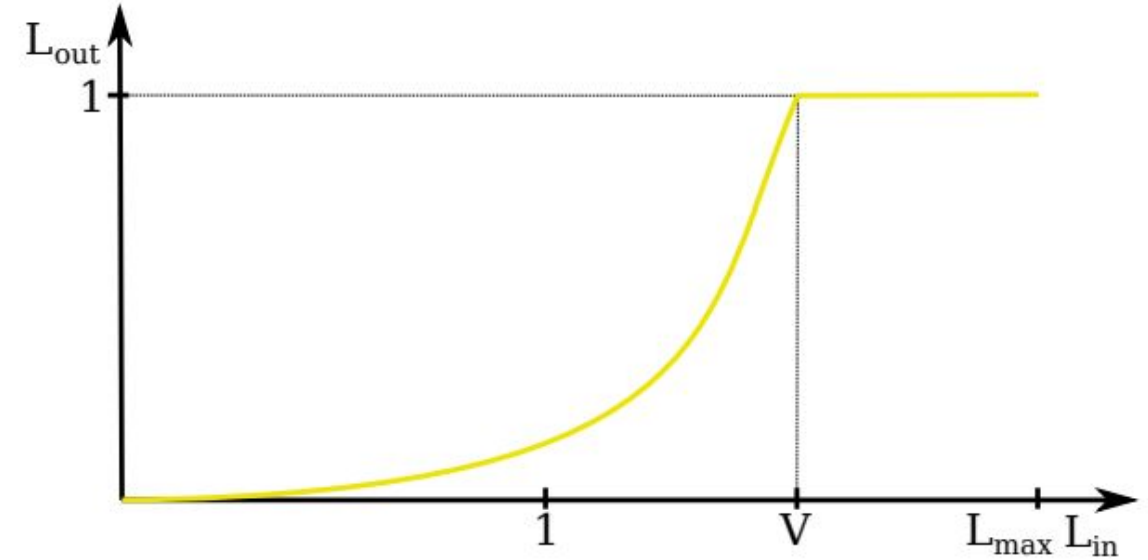
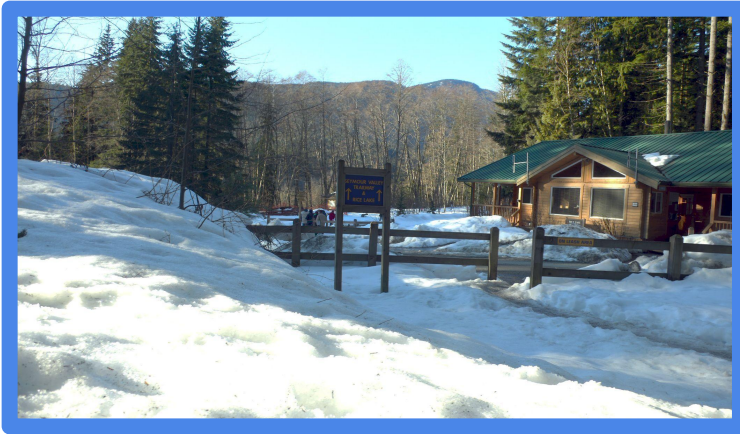
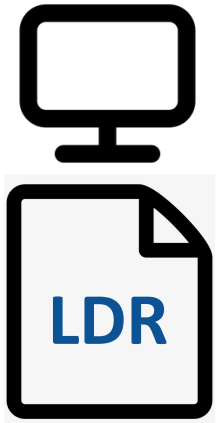
Tone mapping operators



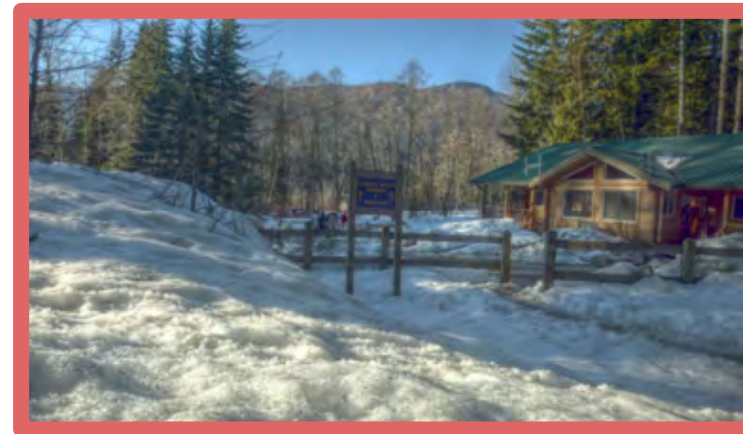
Gamma operator



Tone mapping operators



(Gamma + clamping)
operator



But... how do we store it?

But... how do we store it?

PPM file format

PPM files

- Readable by most viewers
- Simple text file

```
1  P3
2  # feep.ppm
3  4 4
4  15
5  0 0 0 0 0 0 0 0 0 0 15 0 15
6  0 0 0 0 15 7 0 0 0 0 0 0 0
7  0 0 0 0 0 0 0 15 7 0 0 0 0
8  15 0 15 0 0 0 0 0 0 0 0 0 0
```

Details in: <https://netpbm.sourceforge.net/doc/ppm.html>

PPM files

The diagram illustrates the P3 PPM file format with the following content and annotations:

```
1 P3
2 # feep.ppm
3 4 4
4 15
5 0 0 0 0 0 0 0 0 0 15 0 15
6 0 0 0 0 15 7 0 0 0 0 0 0
7 0 0 0 0 0 0 0 15 7 0 0 0
8 15 0 15 0 0 0 0 0 0 0 0 0
```

Annotations:

- Format:** Points to the **P3** header.
- Comment: line starts with #**: Points to the comment line **# feep.ppm**.
- Size resolution: width and height in that order**: Points to the resolution values **4 4**.
- Color resolution (its maximum value)**: Points to the maximum color value **15**.
- A pixel**: Points to a group of three values (0 0 0) representing a single pixel.
- R, G, B**: Labels for the Red, Green, and Blue color channels, positioned above the corresponding columns of values.

The color maximum problem

- Problem 1: PPM standard and HDR files
 - Color stored in disk: [0..15] - Color stored in memory: [0..1]
 - Pixel value [0..15] divided by maximum (15)
 - The standard only allows maximum up to 65535.
 - We will ignore that for HDR images.
 - HDR maximum up to 2^{30}

```
1  P3
2  # feep.ppm
3  4 4
4  15
5  0 0 0 0 0 0 0 0 0 15 0 15
6  0 0 0 0 15 7 0 0 0 0 0 0
```

The memory maximum problem

- Problem 2: Memory maximum value (v) is still only 1
 - Define a custom maximum in your PPM file reader
 - **Optional comment #MAX=<real maximum, decimal number>**

```
1  P3
2  # feep.ppm
3  #MAX=18.35
4  4 4
5  15
6  0 0 0 0 0 0 0 0 0 0 15 0 15
```

The memory maximum problem

- Problem 2: Memory maximum value v is still only 1
 - Define a custom maximum in your PPM file reader
 - **Optional comment #MAX=<real maximum, decimal number>**

$$v = \frac{s}{\frac{m}{c}} \quad \text{READING FILE}$$

$$s = v \frac{c}{m} \quad \text{SAVING FILE}$$

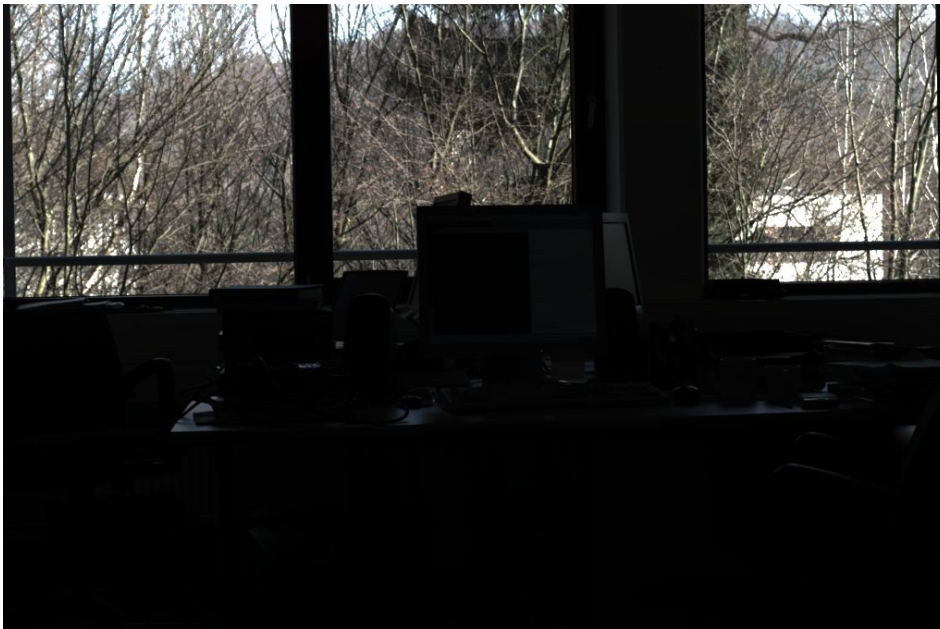
```
1  P3
2  # feep.ppm
3  #MAX=18.35
4  4 4
5  15
6  0 0 0 0 0 0 0 0 0 0 15 0 15
```

Example files

- There are some .ppm files in Moodle you can test with

Example files

- There are some .ppm files in Moodle you can test with
- Some examples (tone mapping RGB channels jointly)
 - **Clamp**



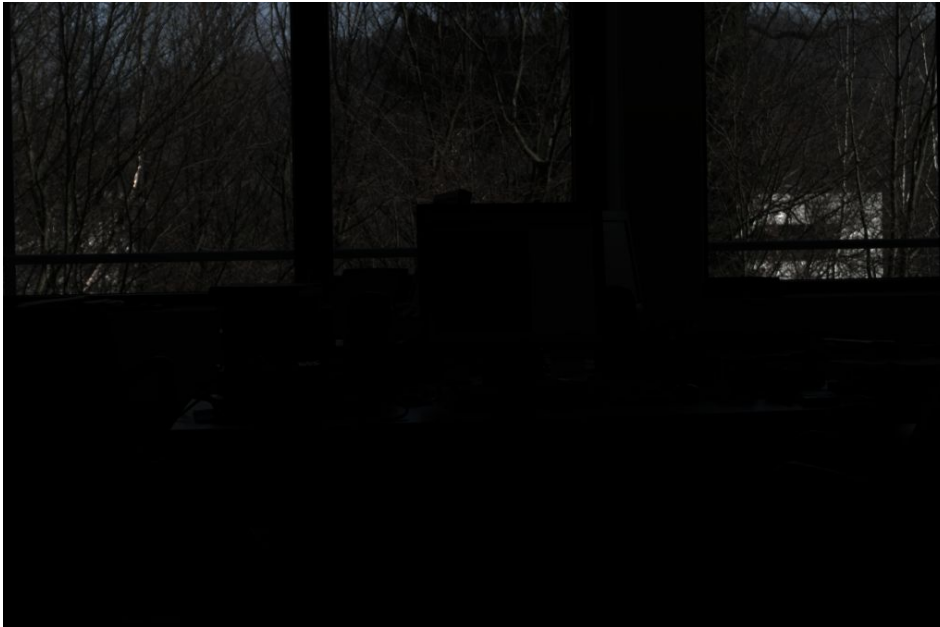
mpi_office.ppm



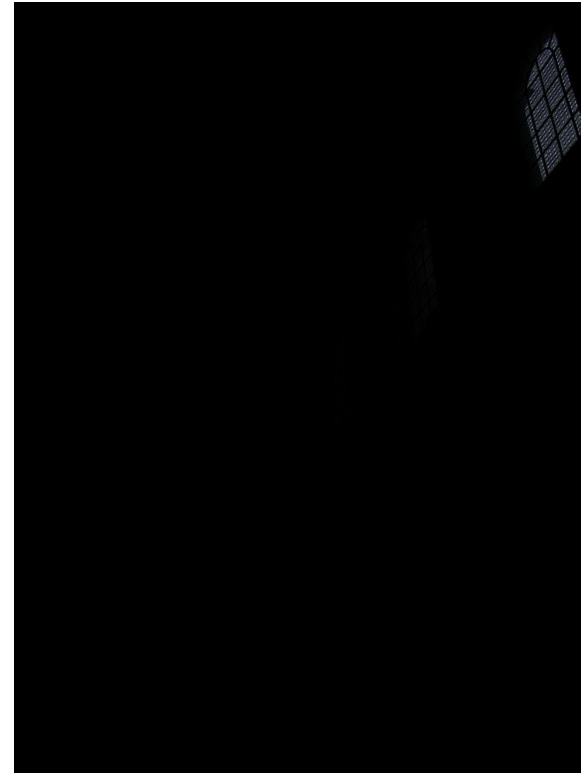
nancy_church_2.ppm

Example files

- There are some .ppm files in Moodle you can test with
- Some examples (tone mapping RGB channels jointly)
 - **Equalization**



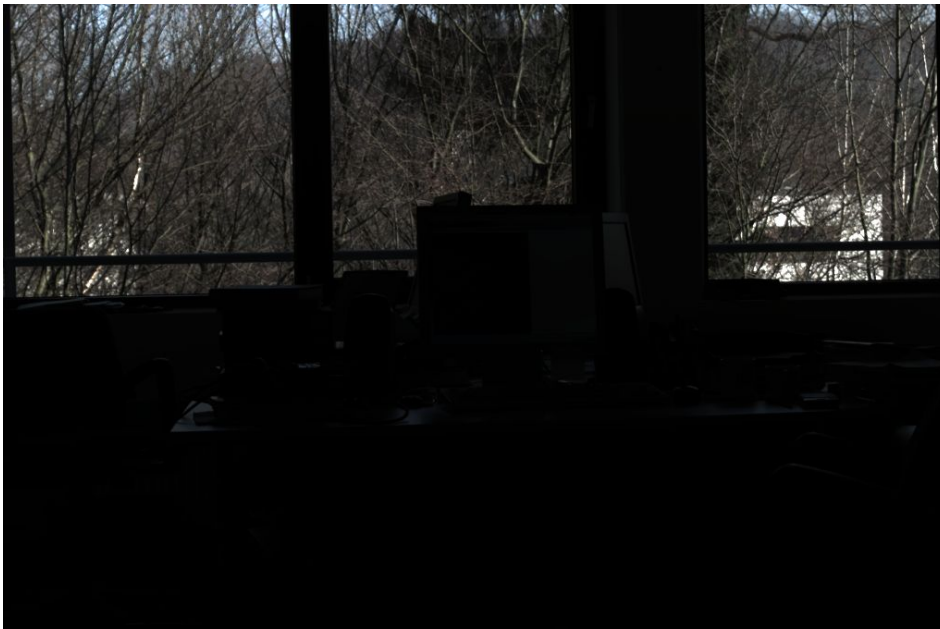
mpi_office.ppm



nancy_church_2.ppm

Example files

- There are some .ppm files in Moodle you can test with
- Some examples (tone mapping RGB channels jointly)
 - **Equalization + clamping**



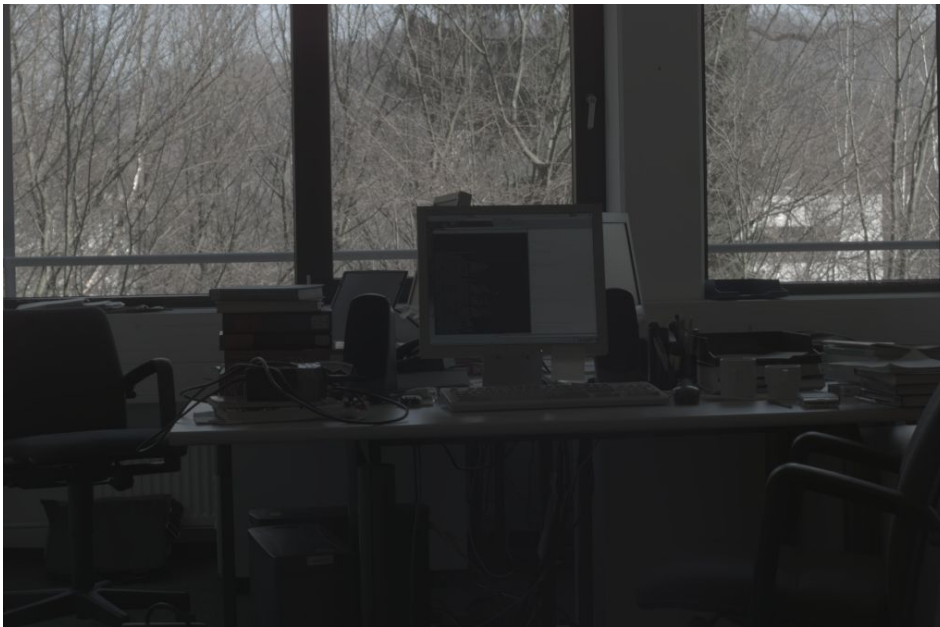
mpi_office.ppm



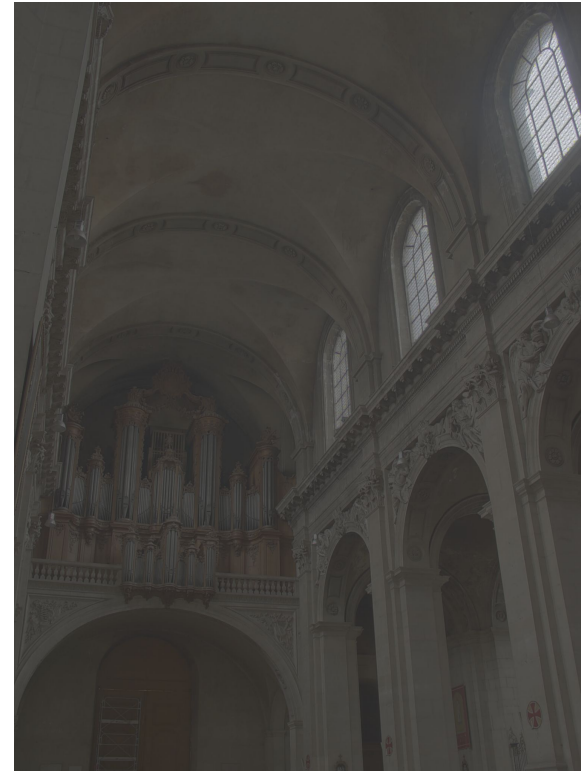
nancy_church_2.ppm

Example files

- There are some .ppm files in Moodle you can test with
- Some examples (tone mapping RGB channels jointly)
 - **Gamma**



mpi_office.ppm



nancy_church_2.ppm

Example files

- There are some .ppm files in Moodle you can test with
- Some examples (tone mapping RGB channels jointly)
 - **Clamp + gamma**



mpi_office.ppm



nancy_church_2.ppm

DO ASK questions, either now or after the lab

But be reasonable, please :)

pluesia@unizar.es | dsubias@unizar.es | o.pueyo@unizar.es

What to expect from this session

In the programming language of your choice, implement:

- **PPM file reader and writer**
 - With disk maximum >65535 and memory maximum >1
- **Implement tone mapping operators**
 - Clamping, equalization, gamma, combinations
- Recommended deadline: **October 2nd**. Extensions:
 - Use other color spaces (HSV, HSL) for tone mapping
 - New tone mapping operators (check class slides)
 - Other LDR formats to save the image
 - You can use existing programs (e.g. convert PPM to PNG), just keep in mind that we are not strictly following the PPM standard