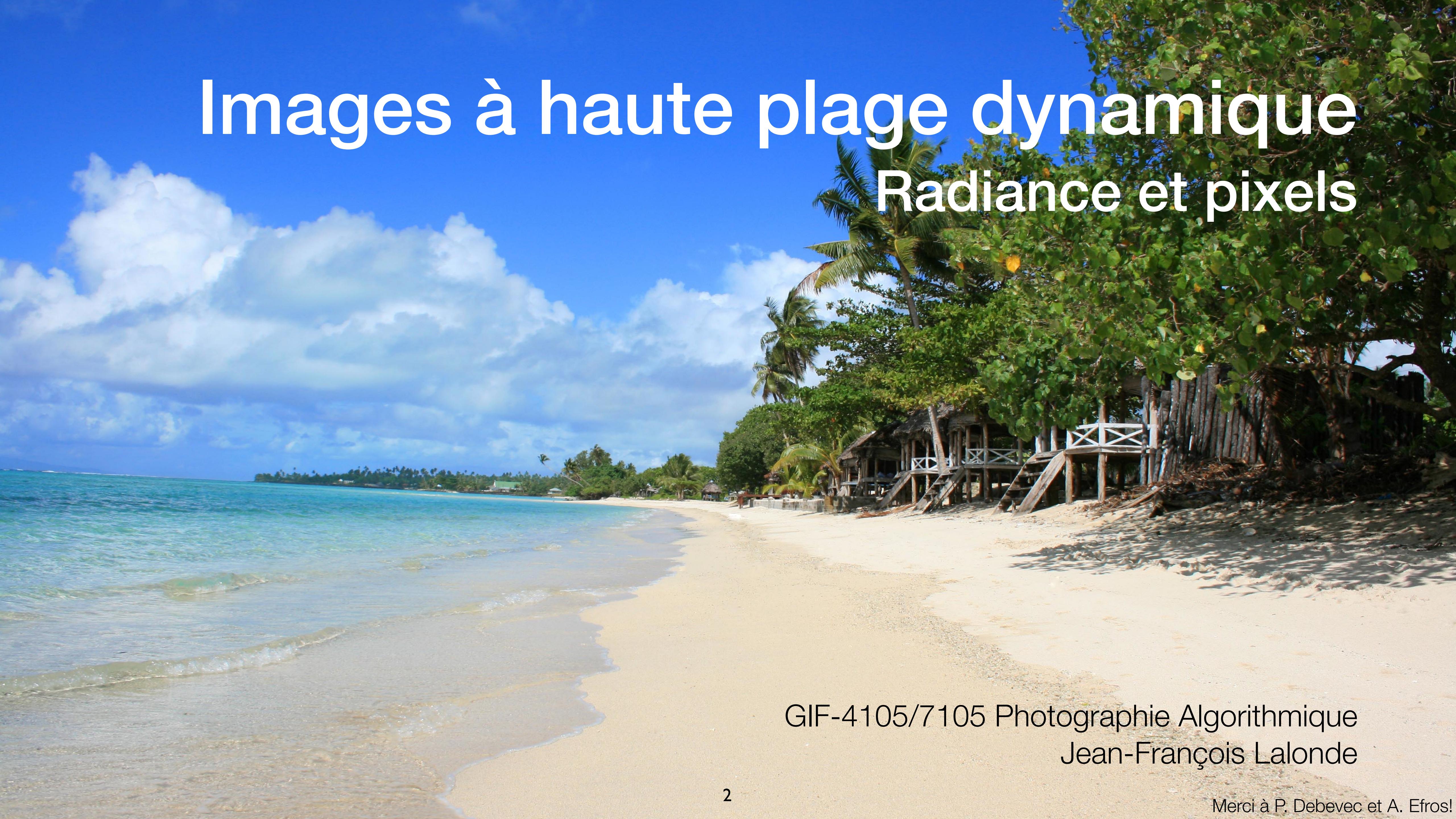


Images à haute plage dynamique



GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Images à haute plage dynamique Radiance et pixels



GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde



ou





Plage dynamique

1

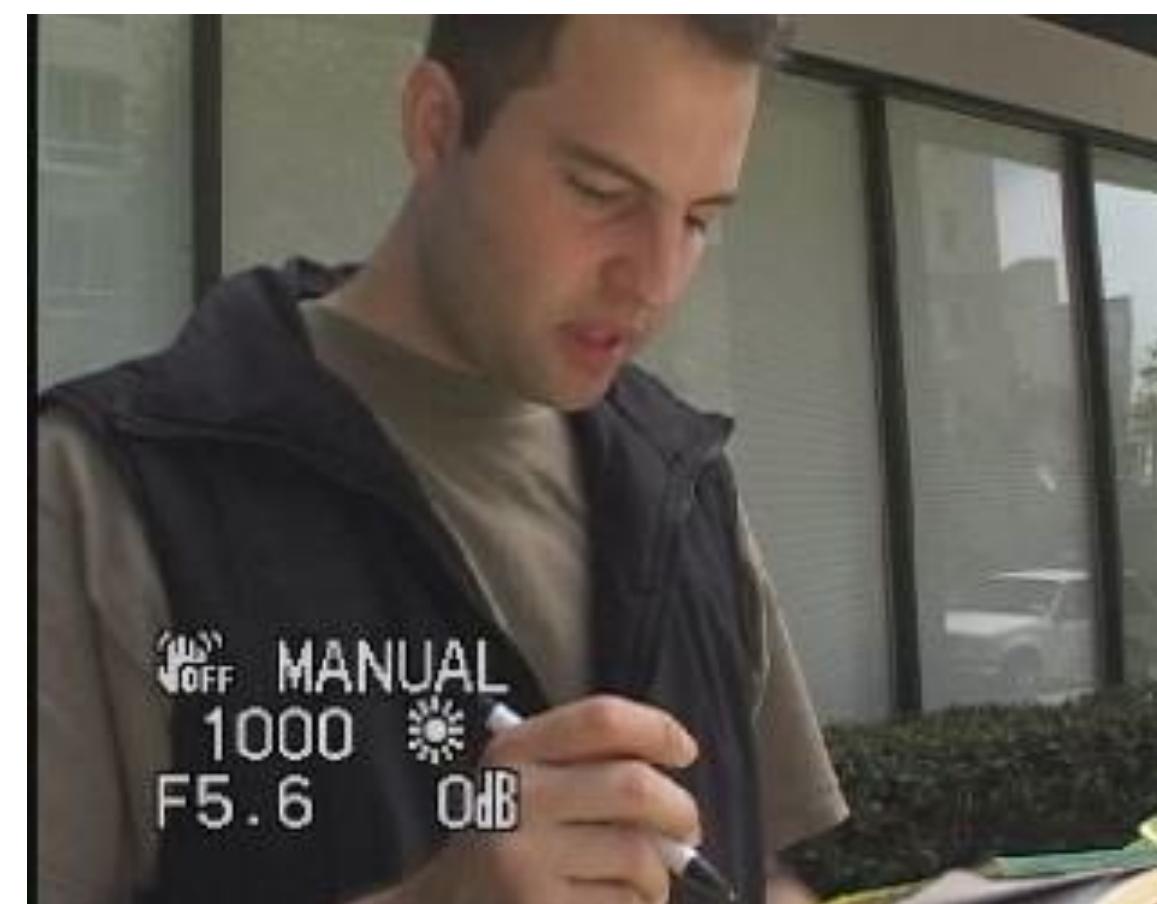


Plage dynamique : valeurs de radiance possibles
Le monde possède une (très) haute plage dynamique!

1 500



25 000



400 000



2 000 000 000



Et la caméra?

Plage dynamique : valeurs de radiance possibles
Le monde possède une (très) haute plage dynamique!

pixel = 42
42 quoi ? photons ?



Quelques définitions

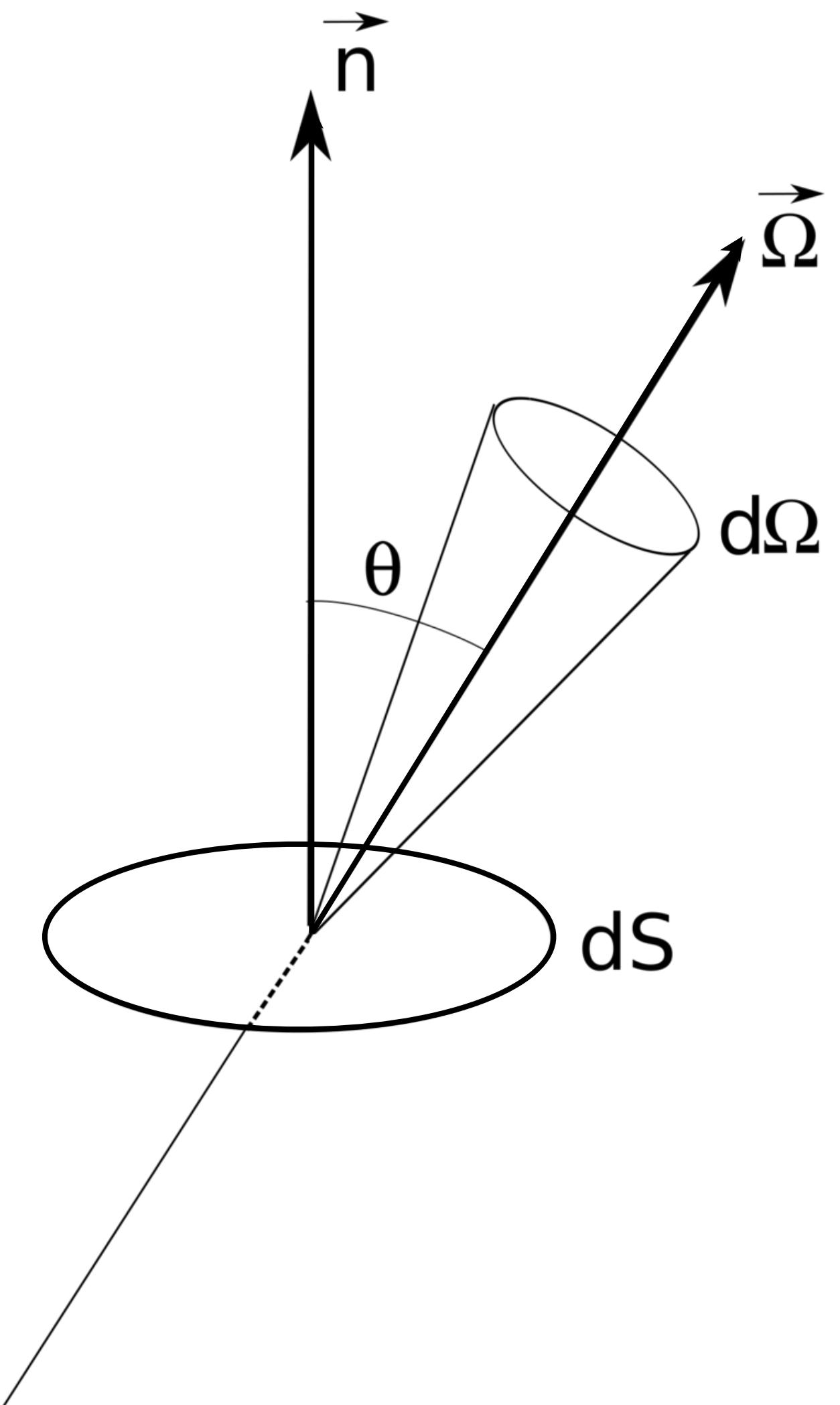
ra·di·ance | 'rādēəns |

Flux énergétique émis, ou reçu, en un point d'une surface, et dans une direction donnée par unité d'angle solide.

Elle s'exprime en $\text{W/m}^2/\text{sr}$.

flux énergétique

puissance d'un rayonnement électromagnétique



Quelques définitions

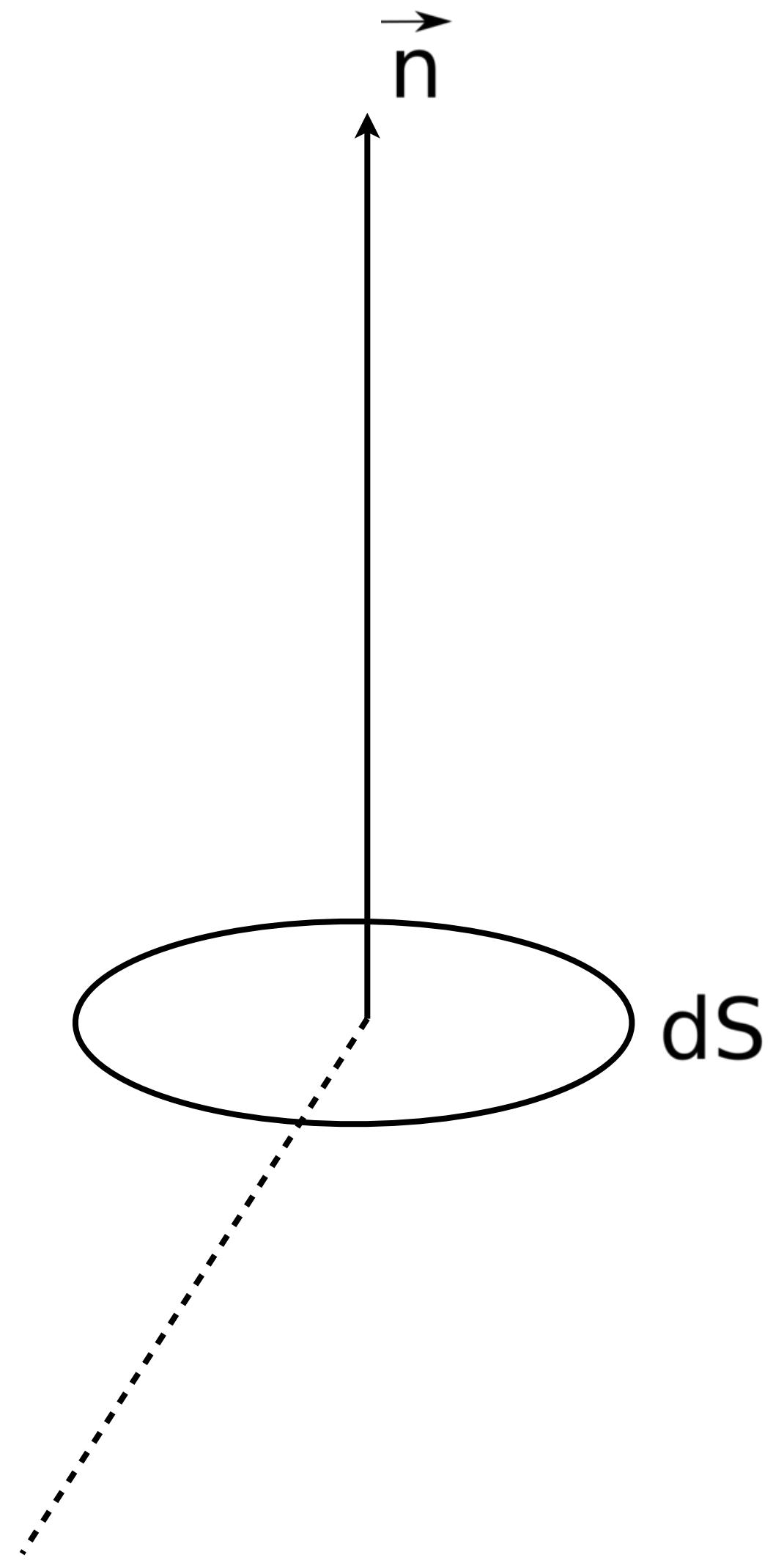
ir·ra·di·ance | i'rādēəns |

Flux énergétique perçue par une surface, par unité d'aire.

Elle s'exprime en W/m^2 .

flux énergétique

puissance d'un rayonnement électromagnétique



Radiance et irradiance

ra·di·ance | 'rādēəns |

Flux énergétique émis, ou reçu, en un point d'une surface, et dans une direction donnée par unité d'angle solide.

Elle s'exprime en $\text{W/m}^2/\text{sr}$.

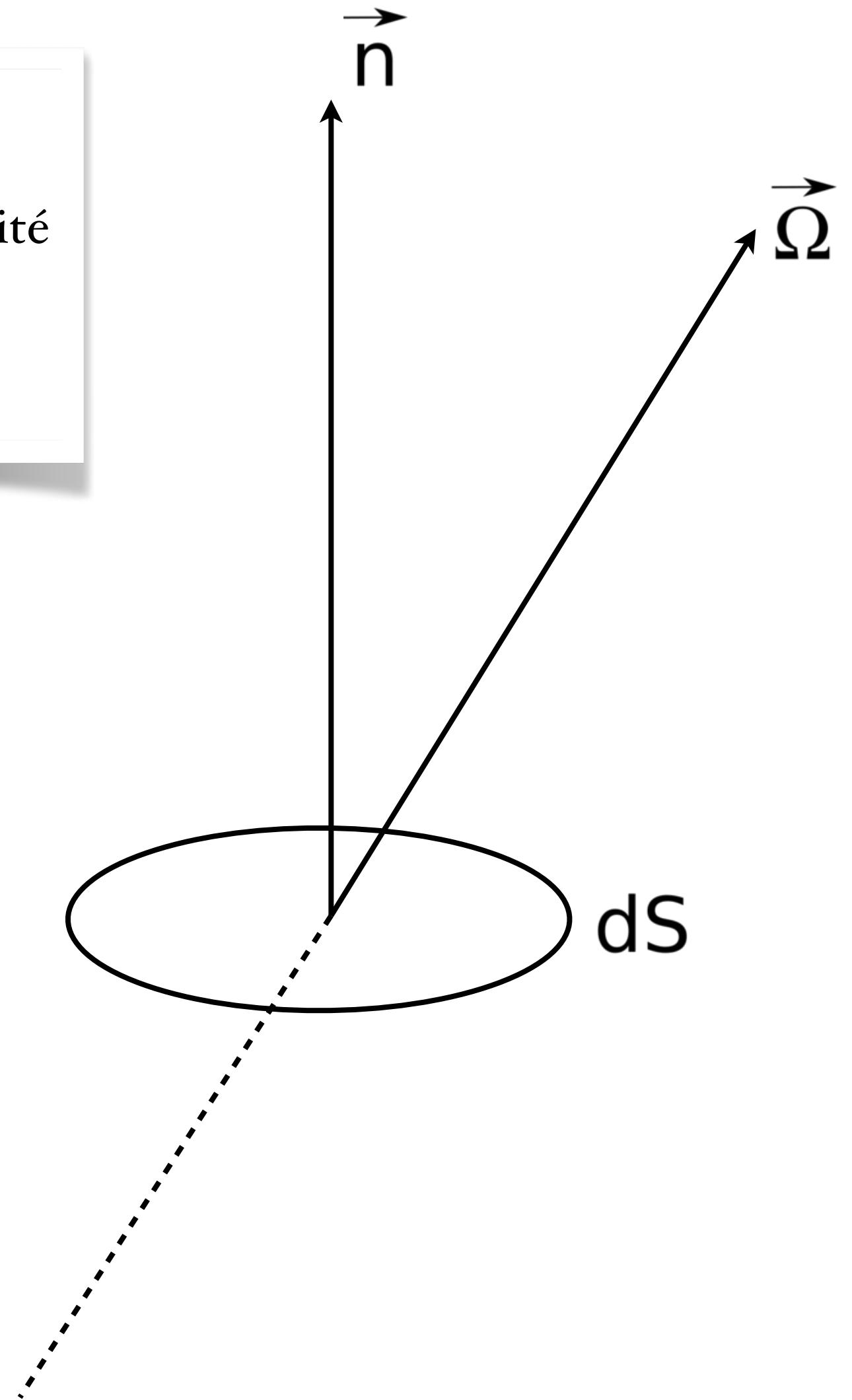
ir·ra·di·ance | i'rādēəns |

Flux énergétique perçue par une surface, par unité d'aire.

Elle s'exprime en W/m^2 .

$$E(\mathbf{x}) = \int_{\Omega} L(\mathbf{x}, \theta, \phi) \cos \theta d\Omega$$

L'**irradiance** est donc l'intégrale de la **radiance** (calculée sur un hémisphère) perçue par une surface.



Et la caméra?

Plage dynamique : valeurs de radiance possibles

Le monde possède une (très) haute plage dynamique!

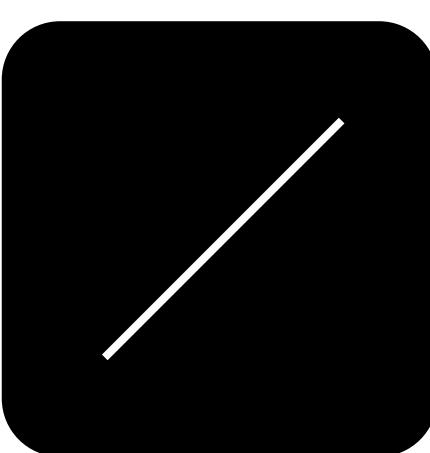
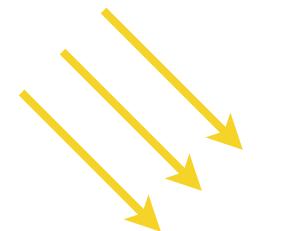
pixel = 42
42 quoi ? photons ? W/m²/sr ?



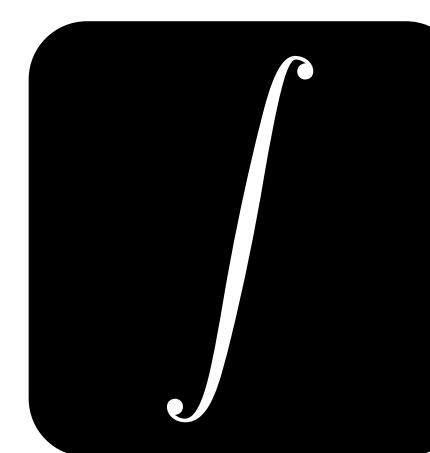
Dans une caméra...

irradiance \propto radianc

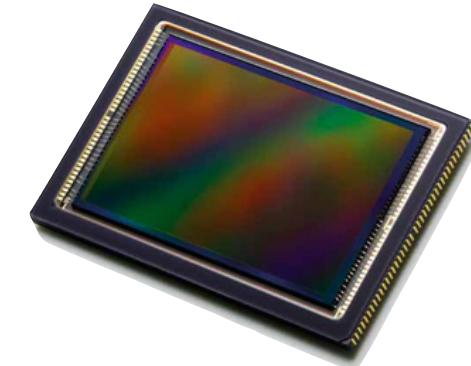
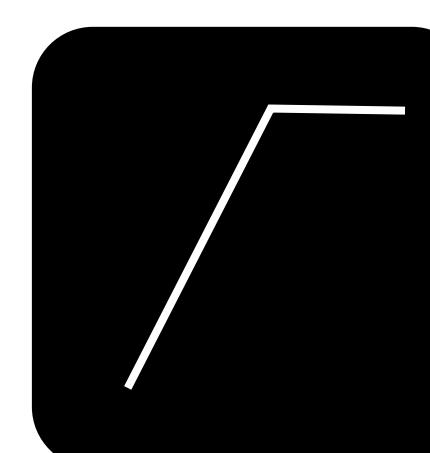
radiance



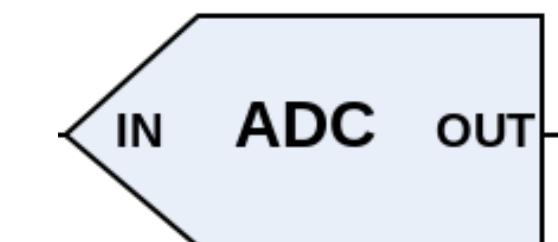
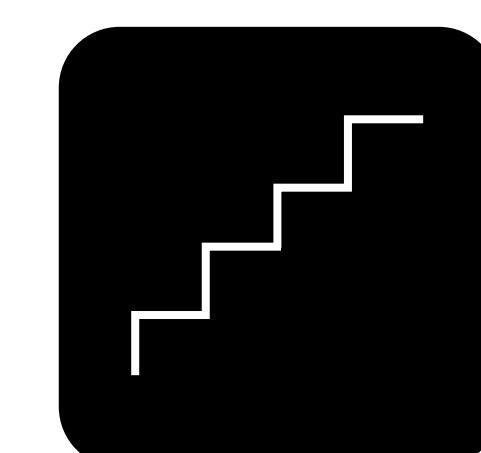
irradiance
au capteur



irradiance



voltage



pixel

Dans une caméra...

irradiance \propto radianc

Lentille

Obturateur

Plus précisément, pour une lentille mince :

$$E = \left(\frac{\pi}{4} \left(\frac{d}{z} \right)^2 \cos^4 \alpha \right) L$$

E : irradiance au capteur

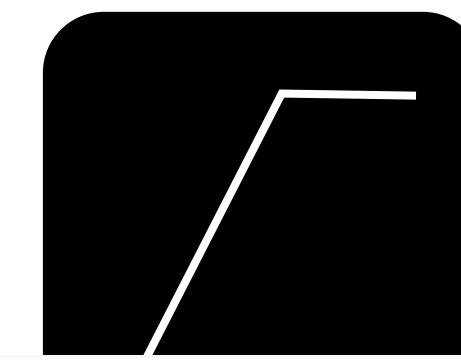
L : radianc émise

d : diamètre de la lentille

z : profondeur

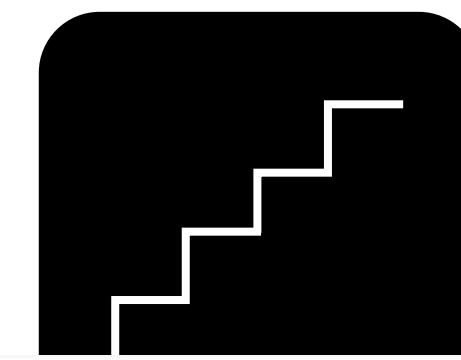
a: angle à l'axe optique

CCD



ce

A/D



voltage

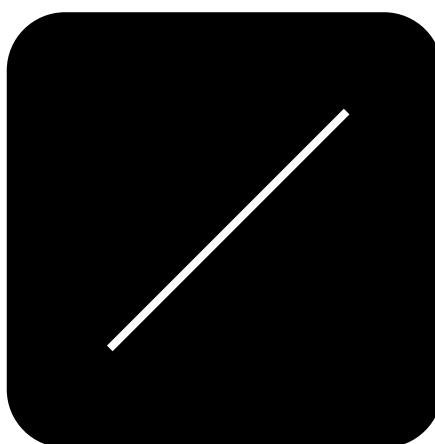
pixel

La plupart des lentilles « modernes » rendent ce rapport à peu près constant sur le plan du capteur!

Dans une caméra...

$$\text{irradiance} \propto \text{radiance}$$

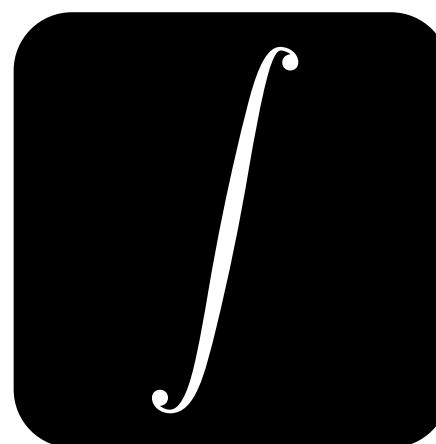
radiance



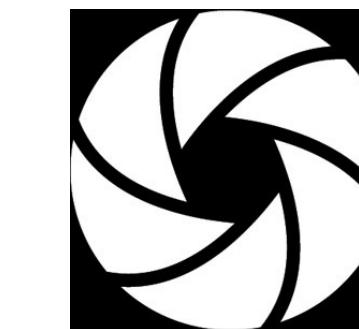
Lentille



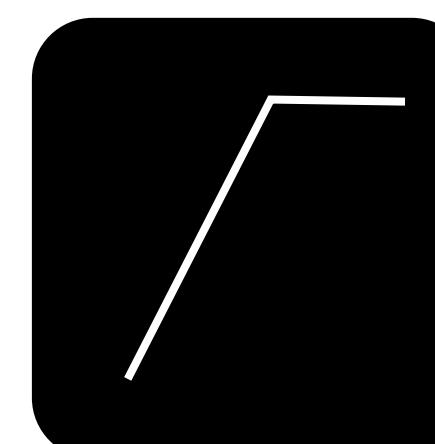
Obturateur



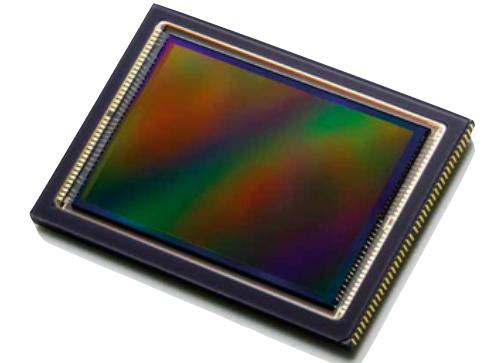
irradiance
au capteur



CCD

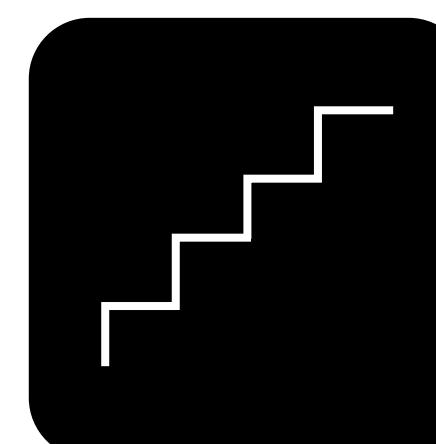


irradiance



$$\text{pixel} \propto \text{radiance} \times \Delta t$$

A/D



voltage



pixel

Relation entre pixels et radiance

- Pour une caméra « linéaire »

$$\text{pixel} \propto \text{radiance} \times \Delta t$$

- Plus l'exposition (Δt) est longue, plus l'image est claire
- Ou encore

$$\text{radiance} \propto \frac{\text{pixel}}{\Delta t}$$

En pratique, les caméras ont également une fonction de réponse non-linéaire...
on y reviendra.

Images à haute plage dynamique

Capturer la radiance



GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Relation entre pixels et radiance

- Pour une caméra « linéaire »

$$\text{pixel} \propto \text{radiance} \times \Delta t$$

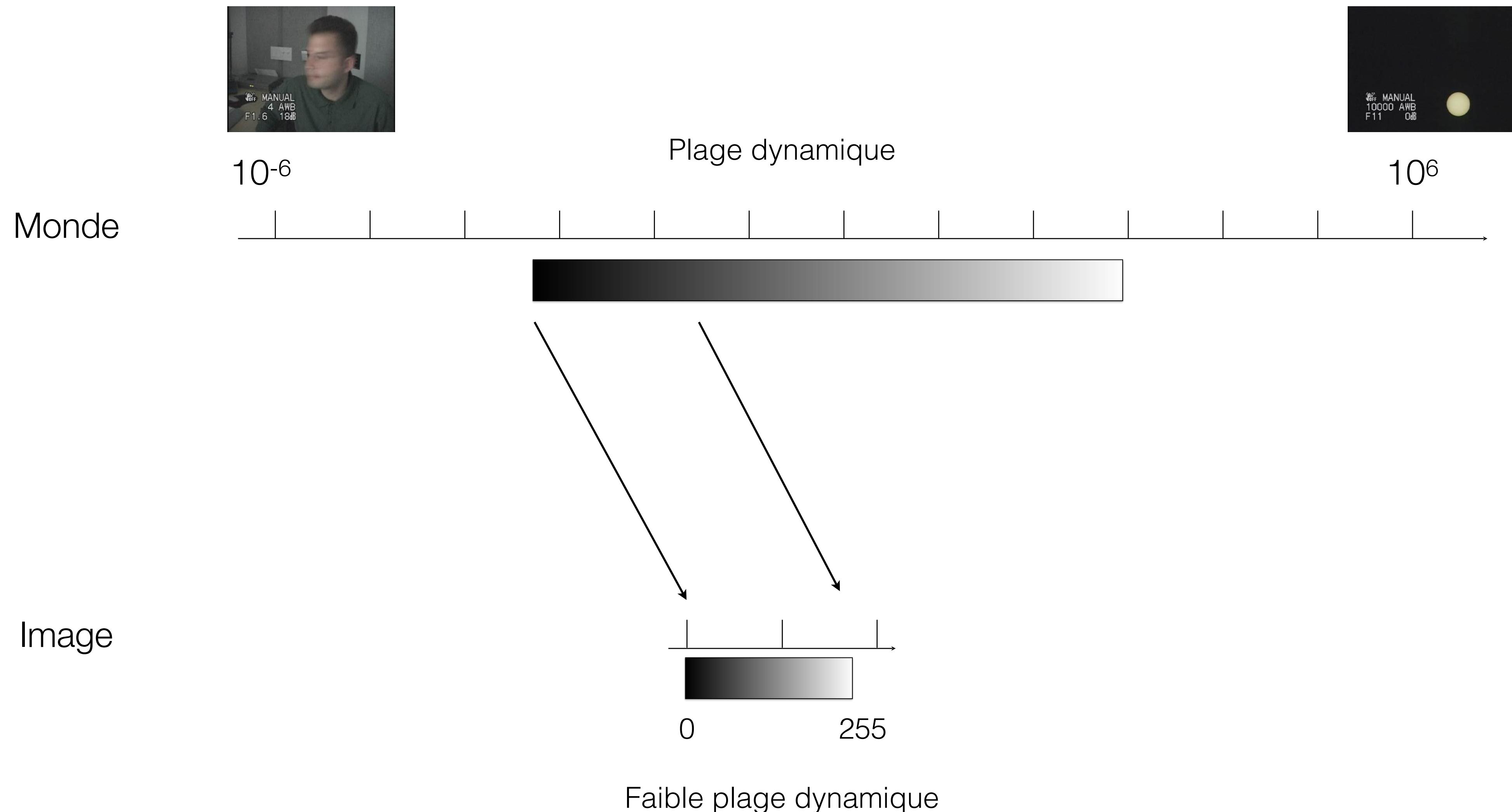
- Plus l'exposition (Δt) est longue, plus l'image est claire
- Ou encore

$$\text{radiance} \propto \frac{\text{pixel}}{\Delta t}$$

En pratique, les caméras ont également une fonction de réponse non-linéaire...
on y reviendra.

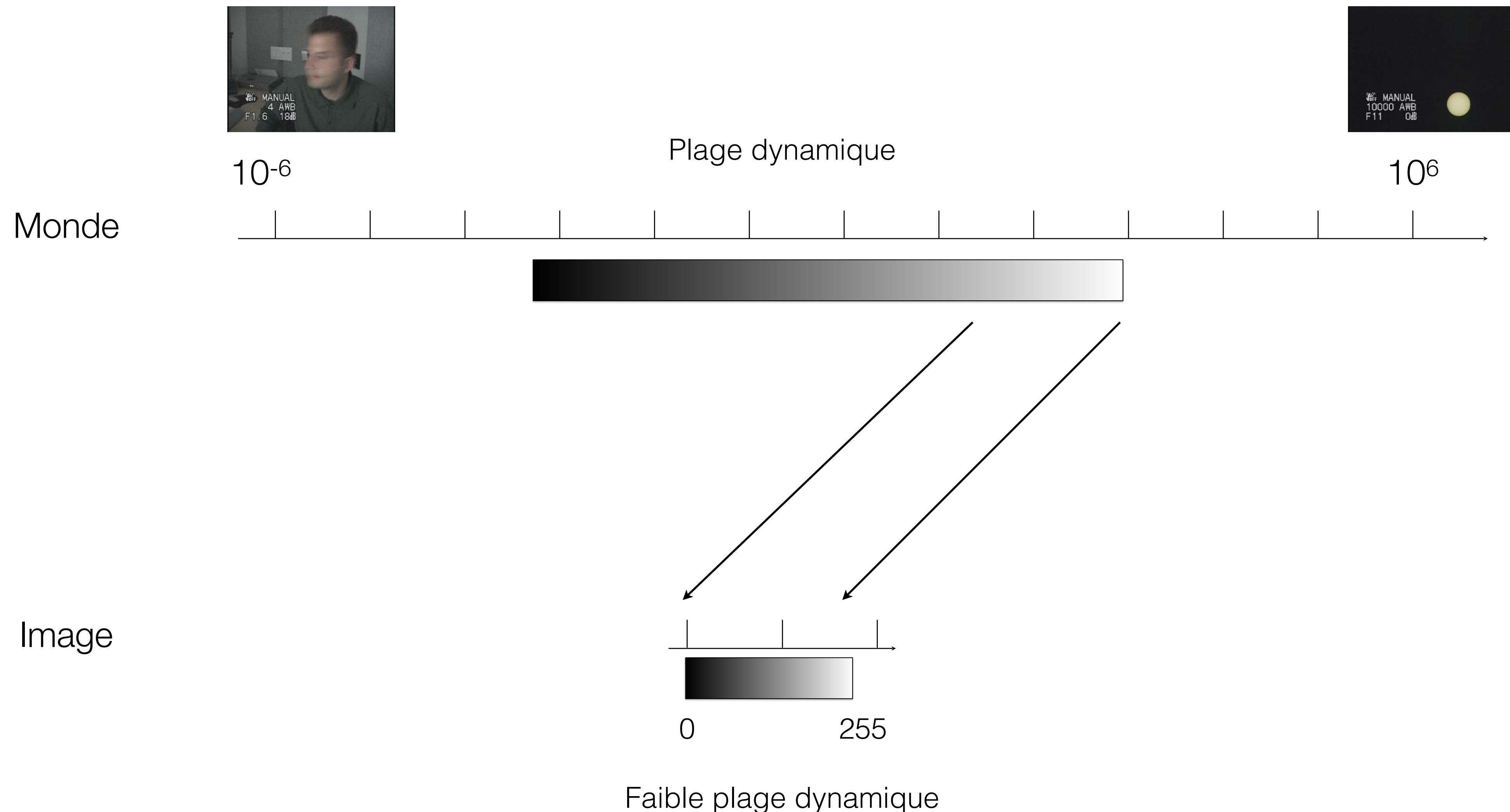
$$\text{radiance} \propto \frac{\text{pixel}}{\Delta t}$$

Exposition longue



$$\text{radiance} \propto \frac{\text{pixel}}{\Delta t}$$

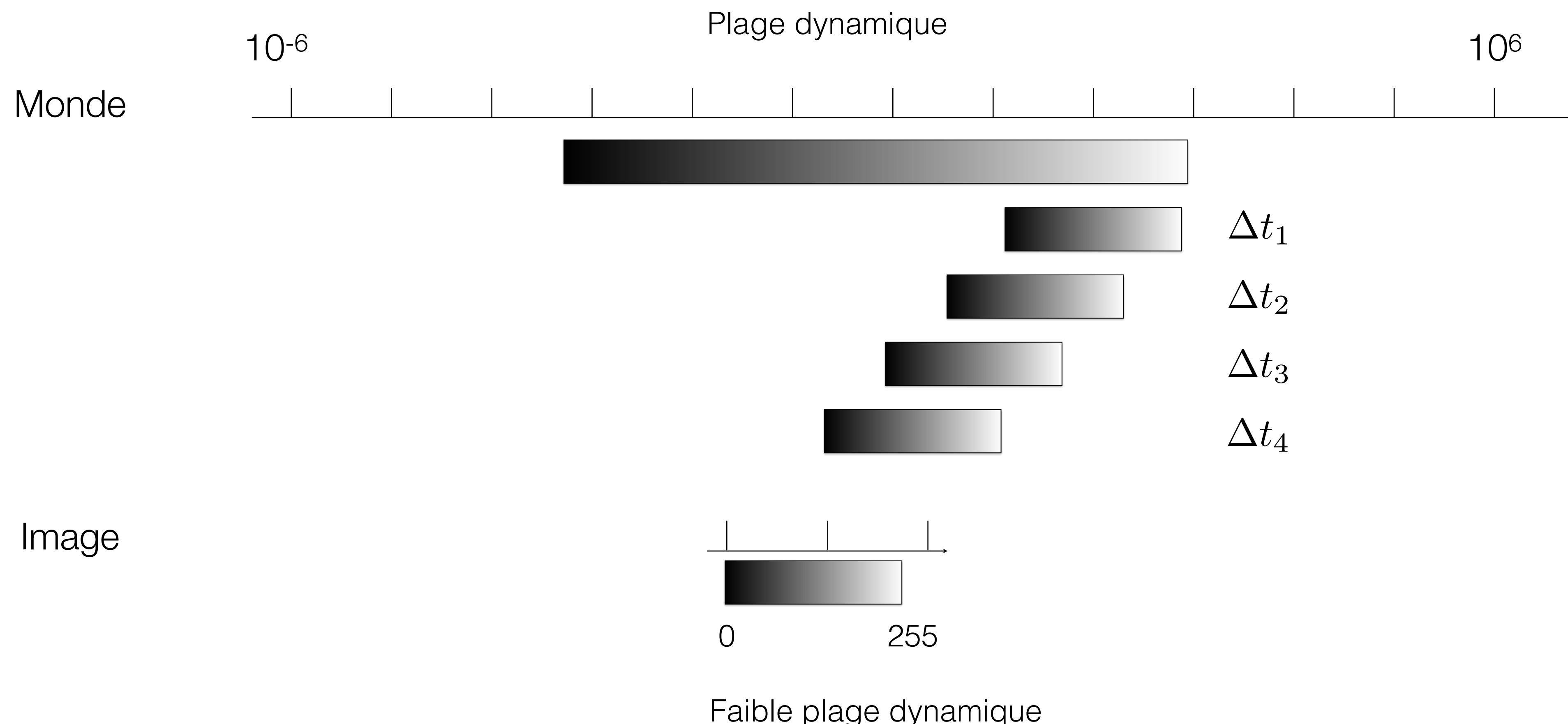
Exposition courte



$$\text{radiance} \propto \frac{\text{pixel}}{\Delta t}$$

Radiance à partir des pixels?

Capturer plusieurs expositions!



$$\text{radiance} \propto \frac{\text{pixel}}{\Delta t}$$

Radiance à partir des pixels?

Capturer plusieurs expositions!



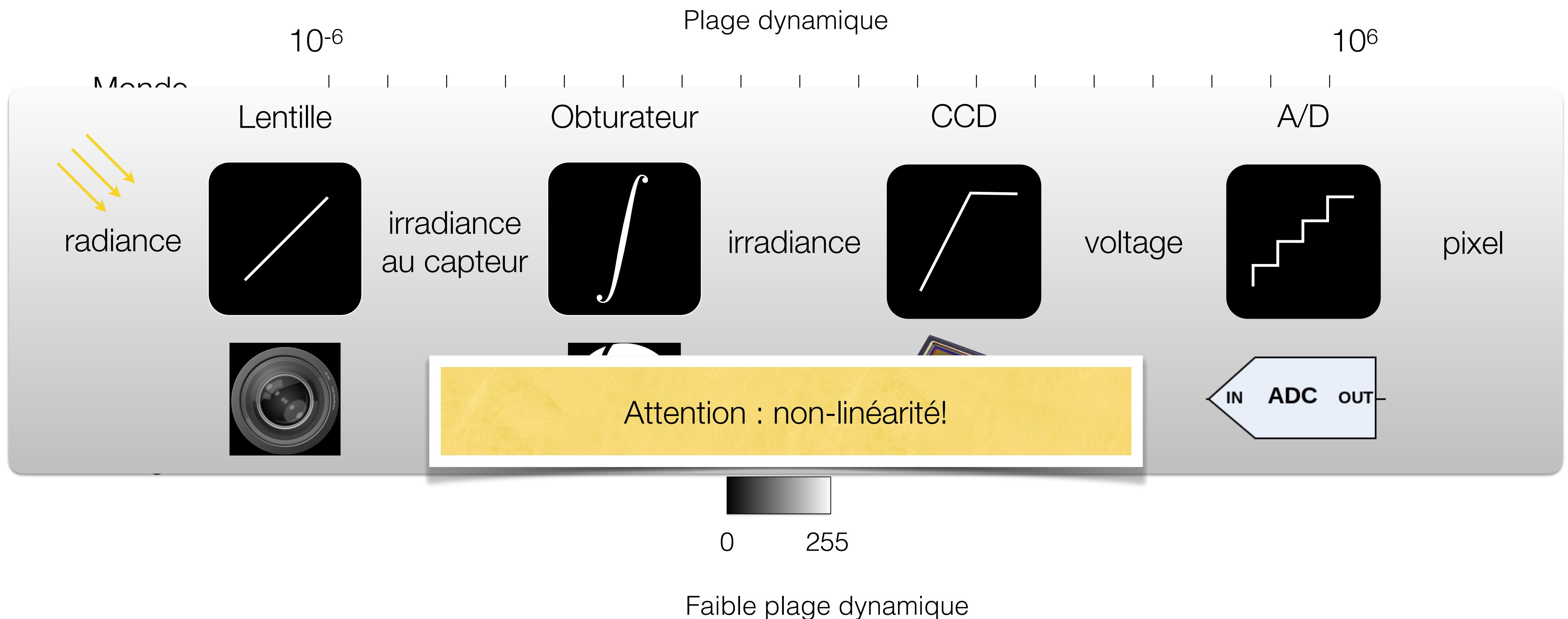
$$I_{\text{radiance}} = \sum_{i=1}^N \frac{I_i}{\Delta t_i}$$



Faible plage dynamique

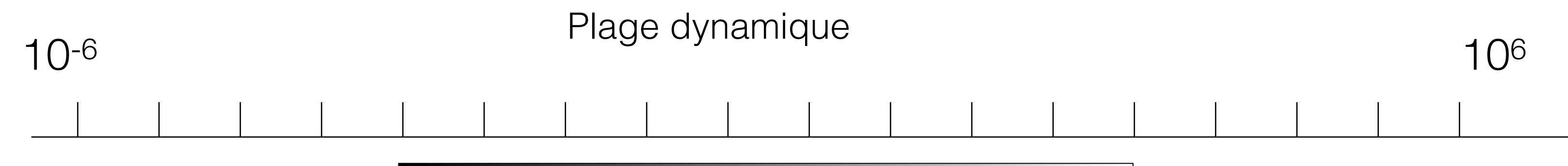
$$\text{radiance} \propto \frac{\text{pixel}}{\Delta t}$$

Radiance à partir des pixels?

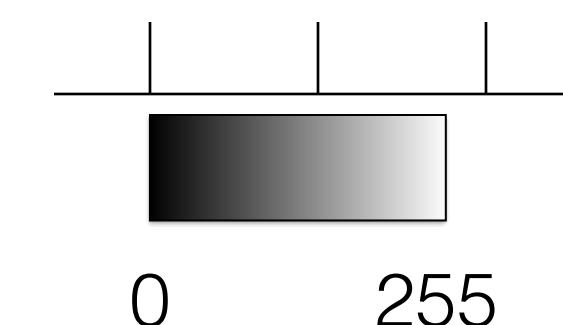


$$\text{radiance} \propto \frac{\text{pixel}}{\Delta t}$$

Radiance à partir des pixels?



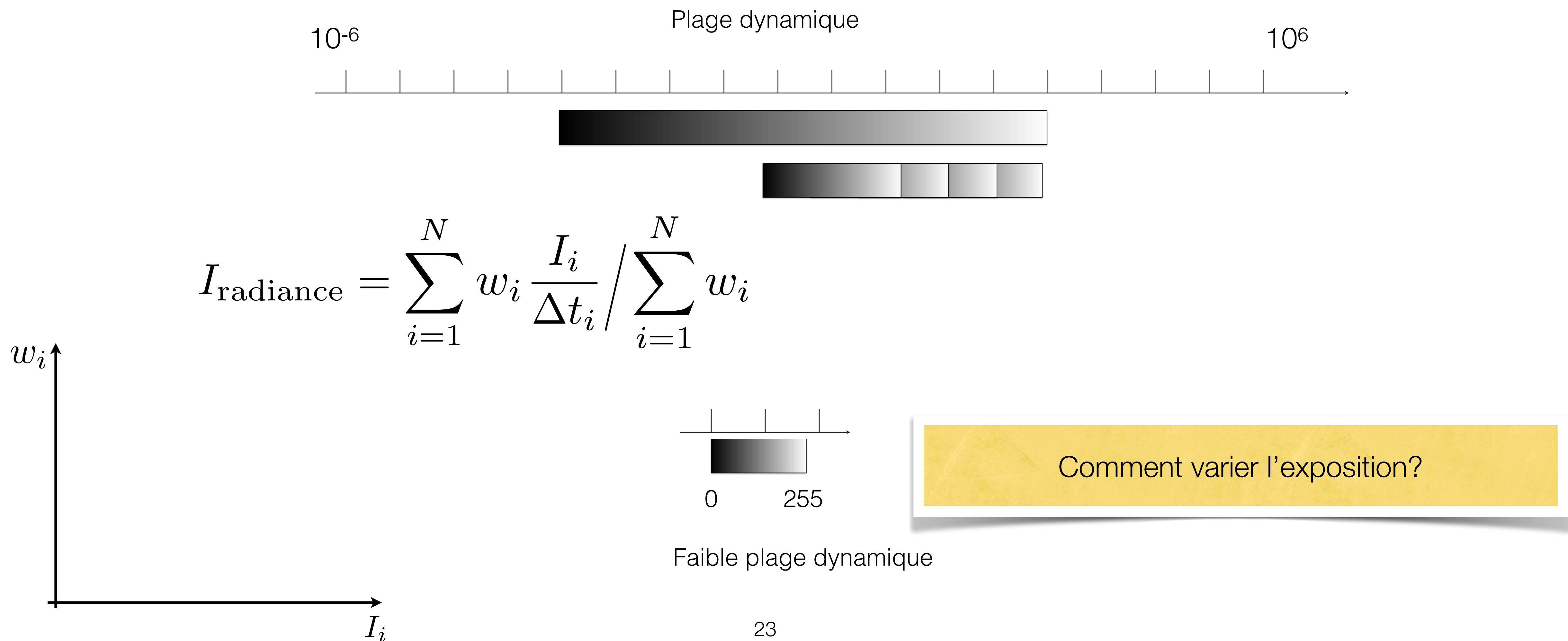
$$I_{\text{radiance}} = \sum_{i=1}^N \frac{I_i}{\Delta t_i}$$



Faible plage dynamique

$$\text{radiance} \propto \frac{\text{pixel}}{\Delta t}$$

Radiance à partir des pixels?



La caméra Exposition

Rappel!



(c) Tomasz Pluciennik

24

GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

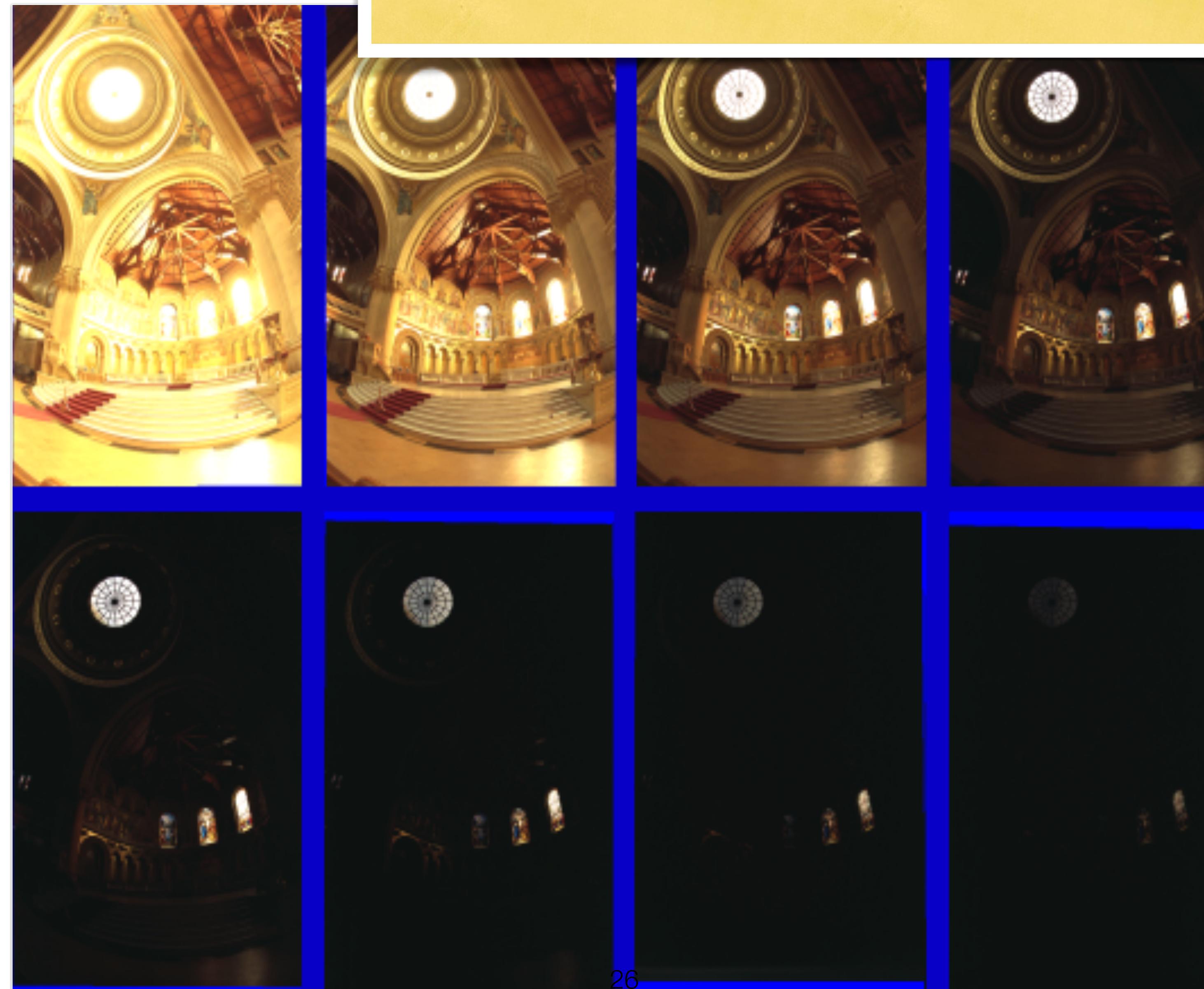
Merci à A. Efros pour (la plupart) des slides!

Varier l'exposition

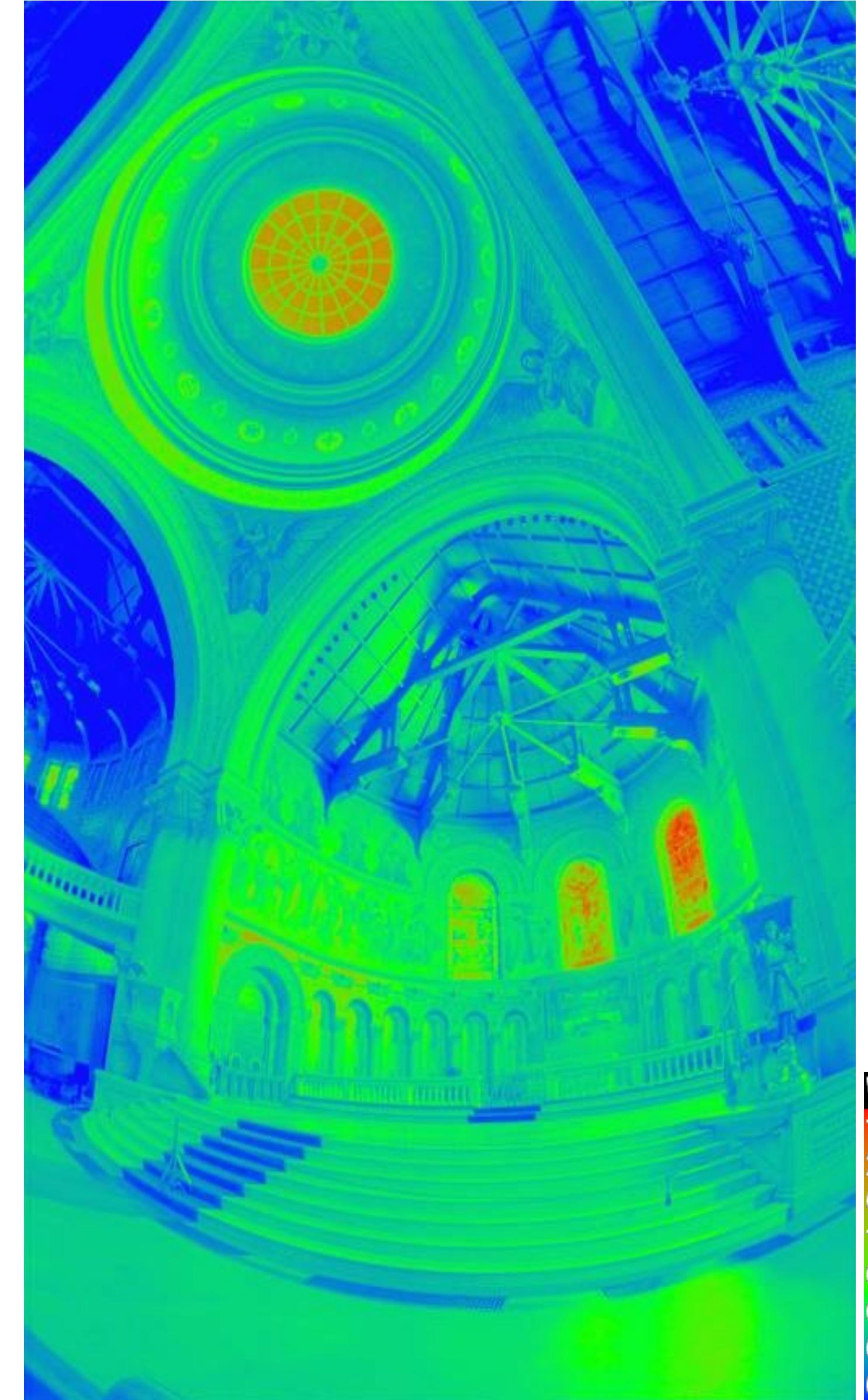
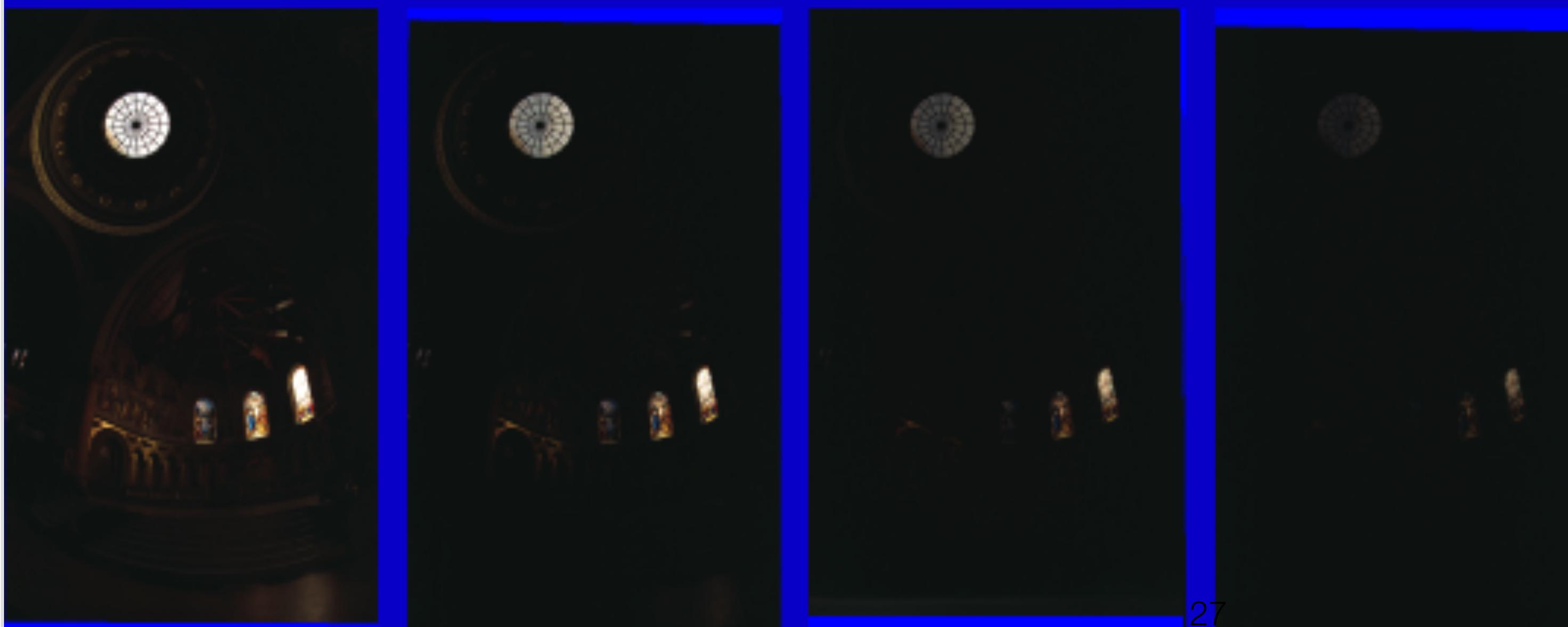
- 3 choix :
 1. Vitesse de l'obturateur (*shutter speed*)
 2. Ouverture (*f-number*)
 3. ISO

Images à plusieurs

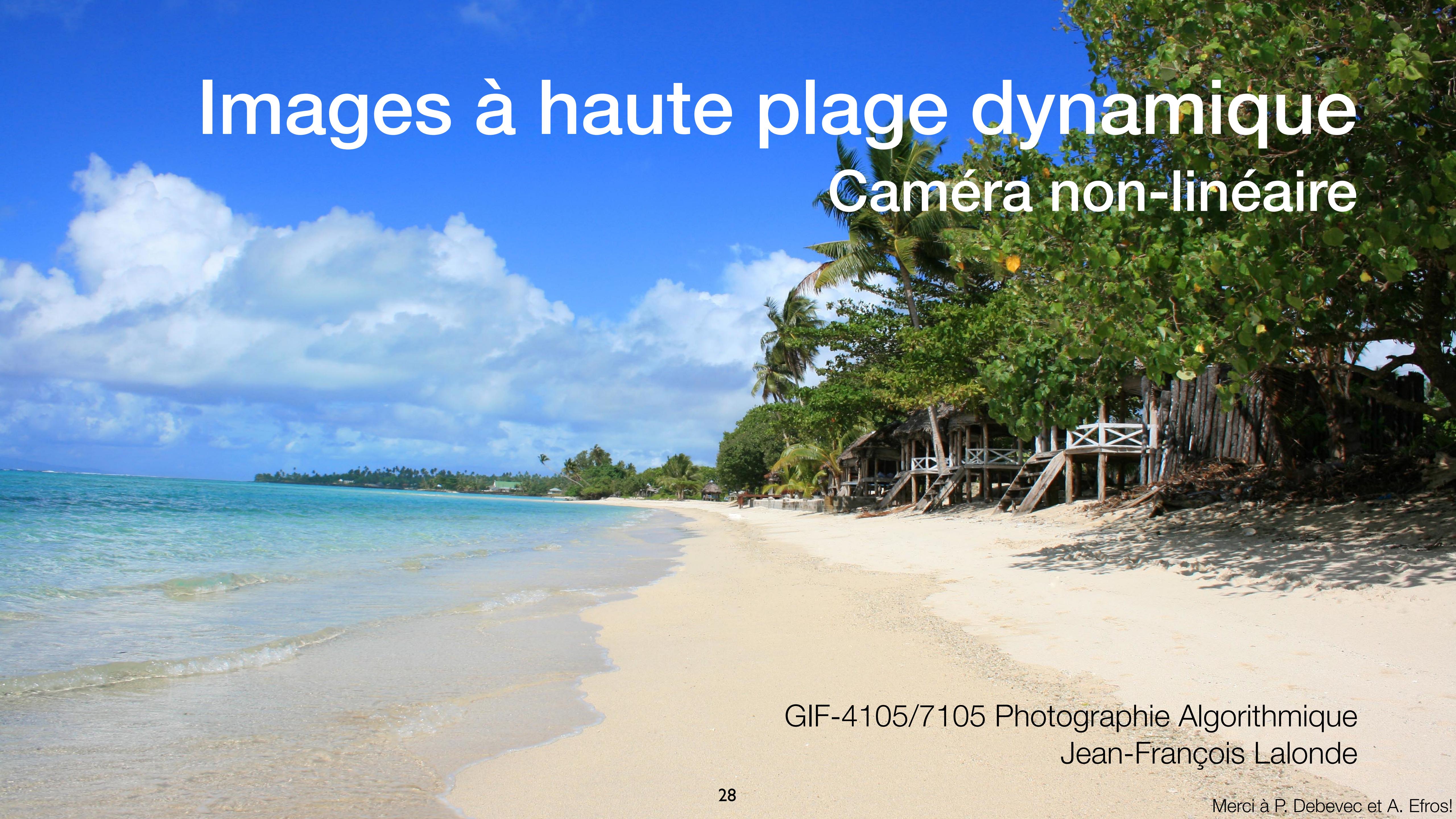
Habituellement, on diminue la lumière d'un facteur 2
ex: $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{15}$, $\frac{1}{30}$, $\frac{1}{60}$, $\frac{1}{125}$, $\frac{1}{250}$, $\frac{1}{500}$, $\frac{1}{1000}$ s



Images à plusieurs expositions



Images à haute plage dynamique Caméra non-linéaire



GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Relation entre pixels et radiance

- Pour une caméra « linéaire »

$$\text{pixel} \propto \text{radiance} >$$

En pratique, les caméras ont également une fonction de réponse non-linéaire...
on y reviendra.

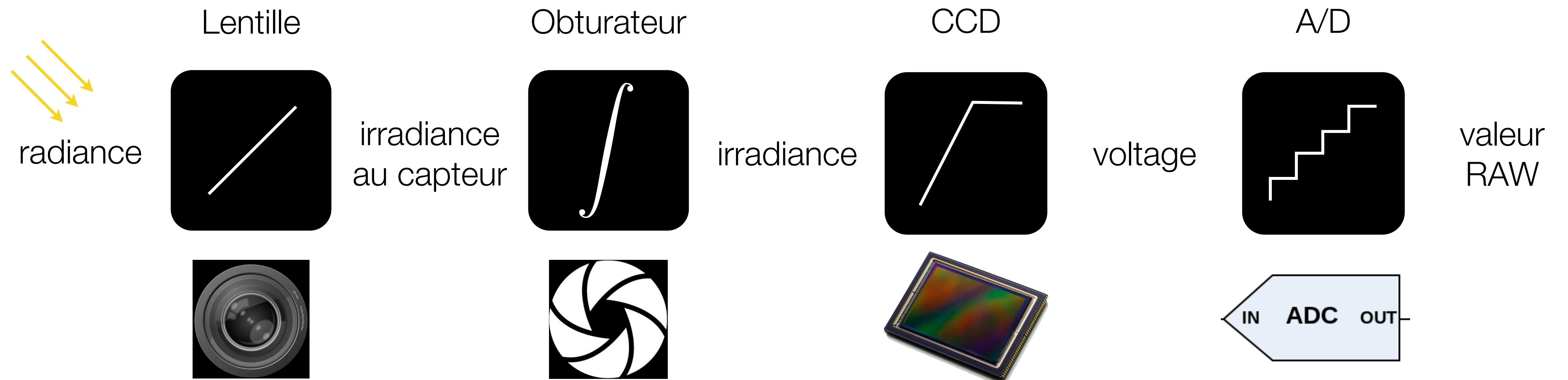
- Plus l'exposition (Δt) est longue, plus l'image est claire
- Ou encore

$$\text{radiance} \propto \frac{\text{pixel}}{\Delta t}$$

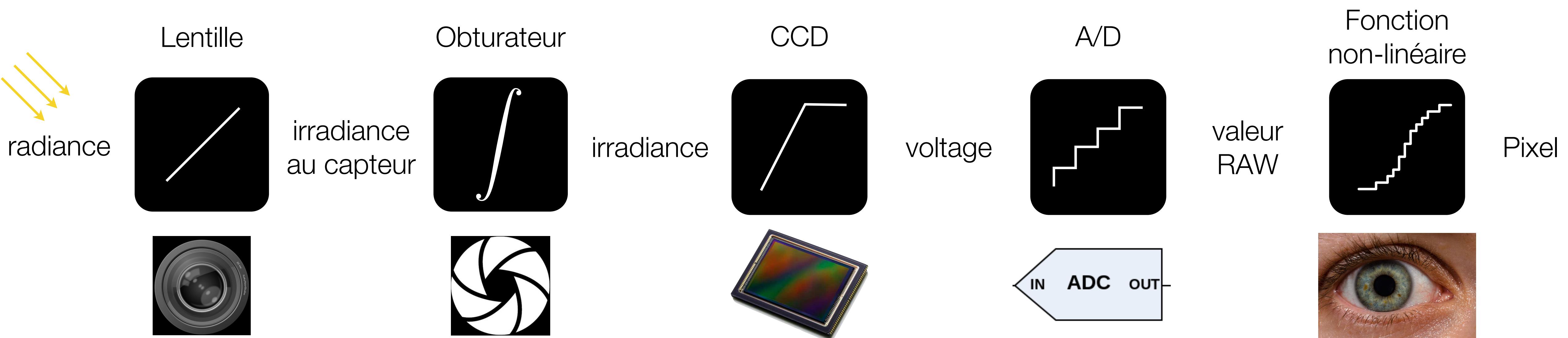
On y revient maintenant!

Rappel!

Dans une caméra...



Dans une caméra...



Dans une caméra...

Pourquoi la fonction non-linéaire?

Laquelle de ces deux images illustre une augmentation **constante** d'intensité?

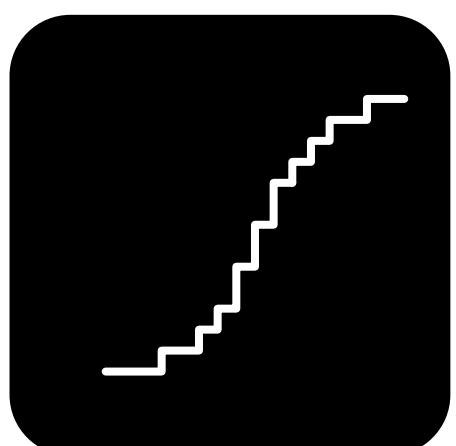
a)



b)



Fonction
non-linéaire



Pixel



Dans une caméra...

Pourquoi la fonction non-linéaire?

Laquelle de ces deux images illustre une augmentation **constante** d'intensité?

a)



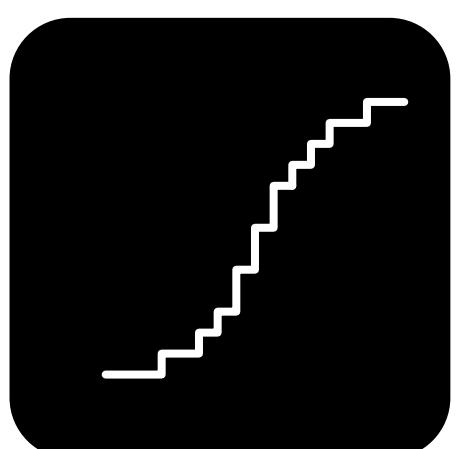
Augmentation constante d'intensité **émise**

b)



Augmentation constante d'intensité **perçue**

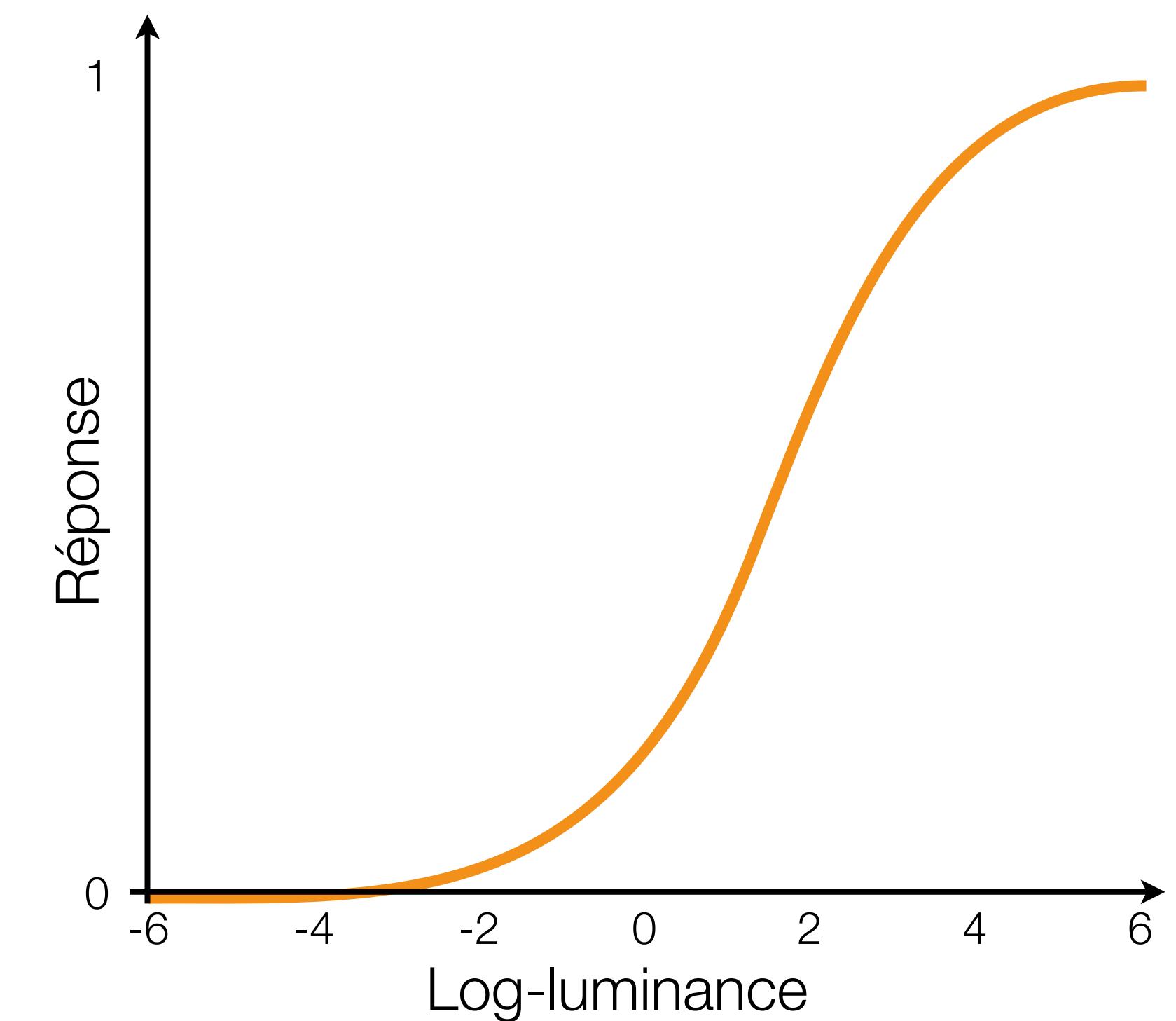
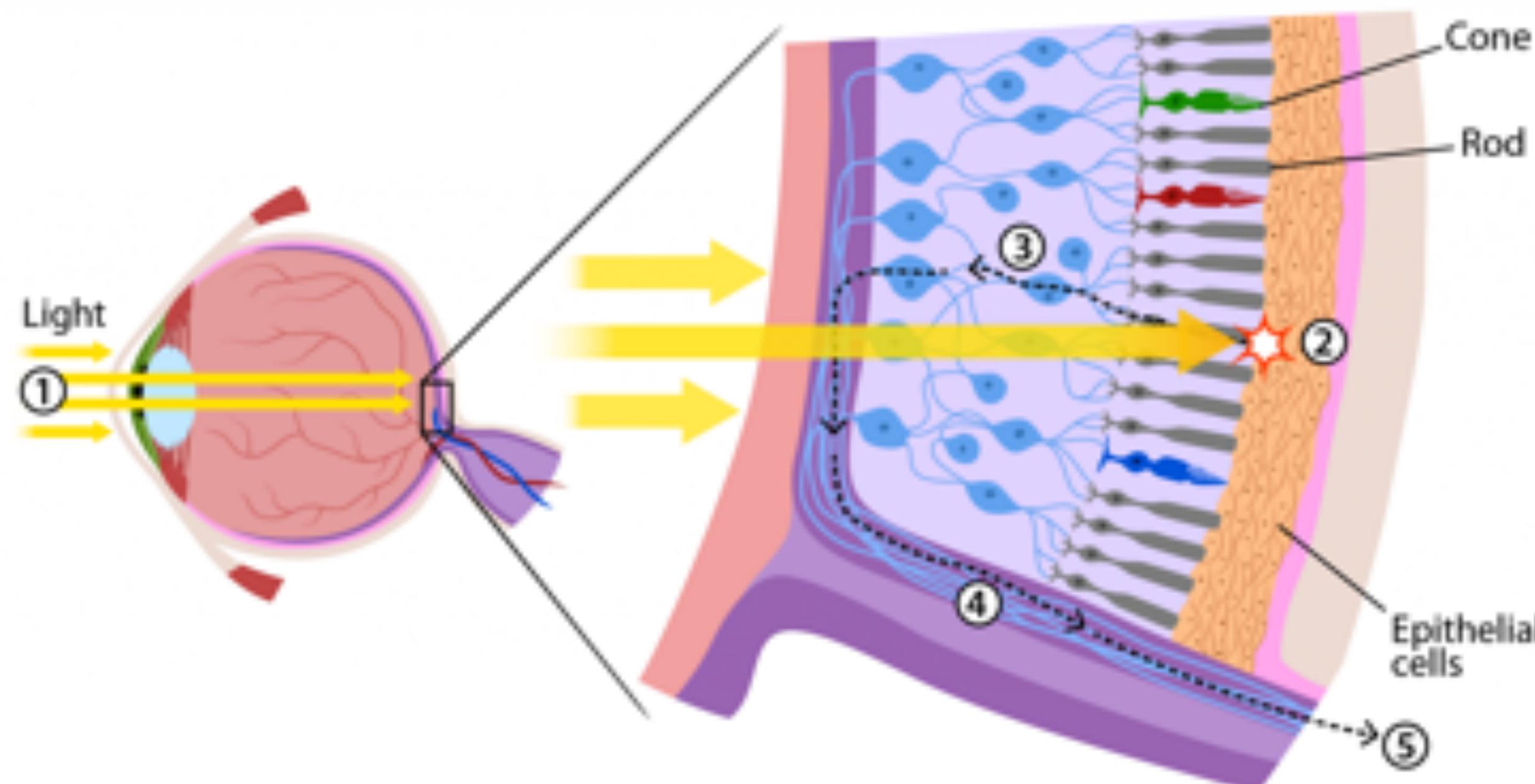
Fonction
non-linéaire



Pixel



Système visuel humain

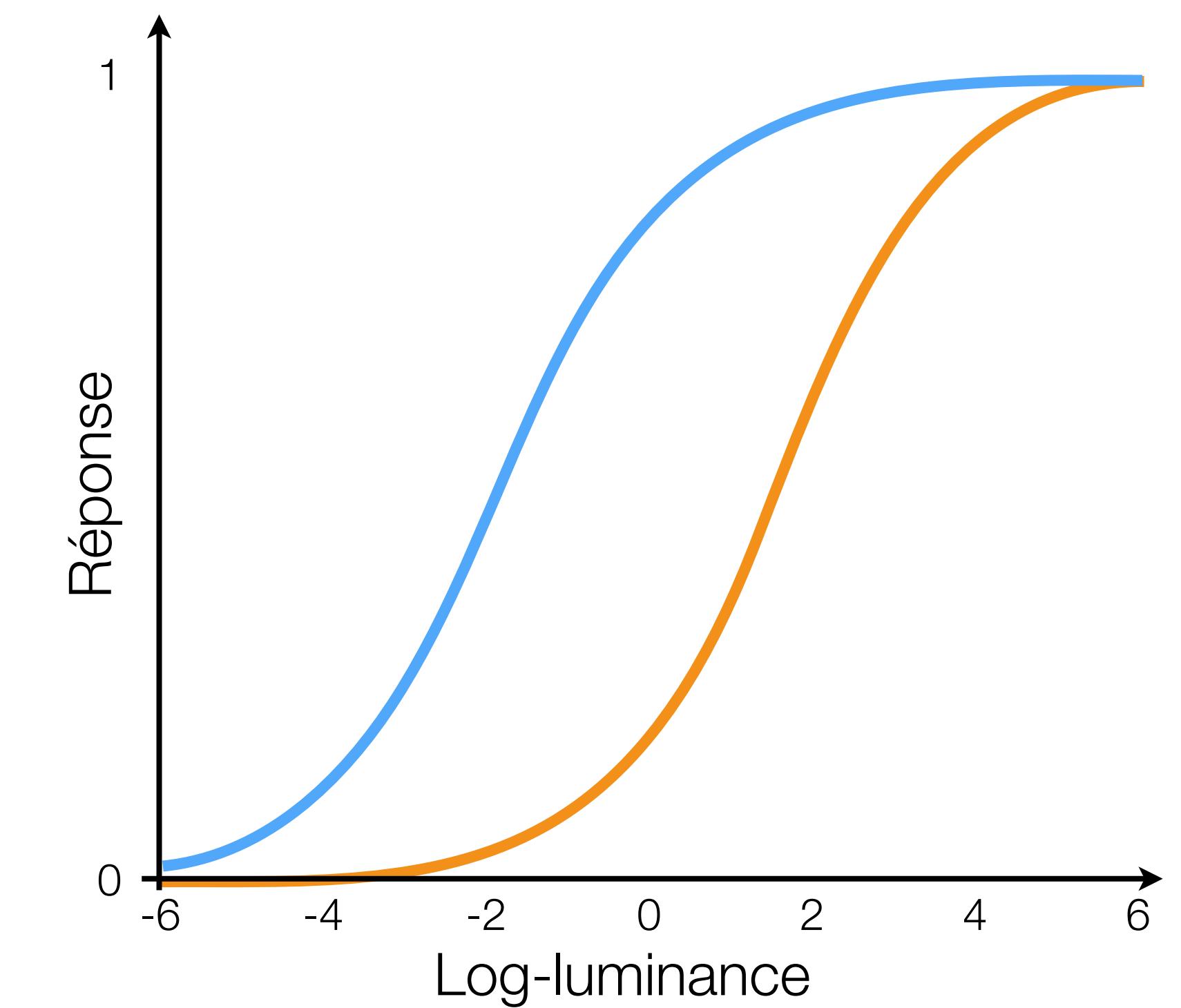


Adaptation : l'oeil n'est pas un photomètre!

Phares allumés, nuit



Phares allumés, jour



Images à haute plage dynamique

Calibrage radiométrique manuel



GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Calibrage

- Géométrique
 - Relation entre les coordonnées en pixel et les points dans le monde
- Photométrique
 - Relation entre les valeurs d'intensité des pixels et la radiance du monde

Dans une caméra...

Rappel!

Pourquoi la fonction non-linéaire?

Laquelle de ces deux images illustre une augmentation **constante** d'intensité?

a)

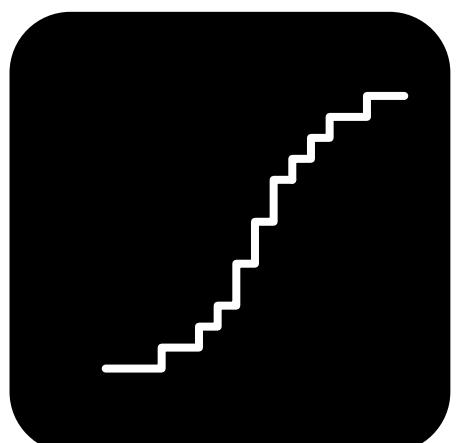


Augmentation constante d'intensité **émise**

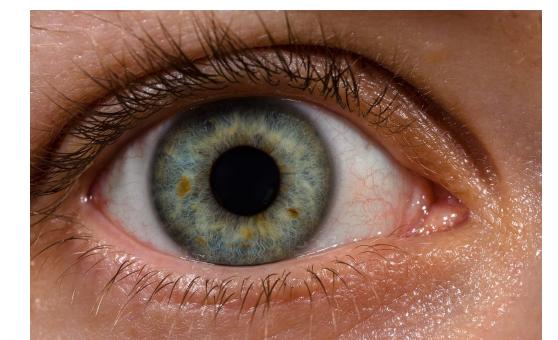
b)



Fonction
non-linéaire



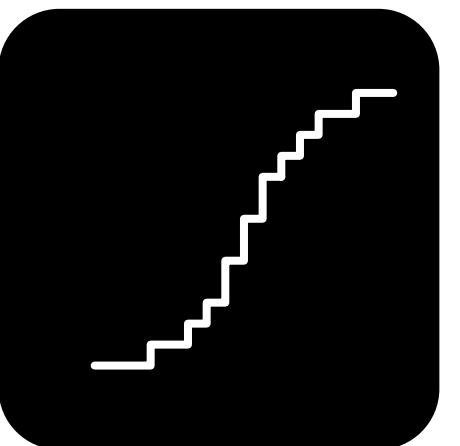
Pixel



Dans une caméra...

- Comment estimer cette fonction?
- Première option : utiliser une cible de calibrage radiométrique

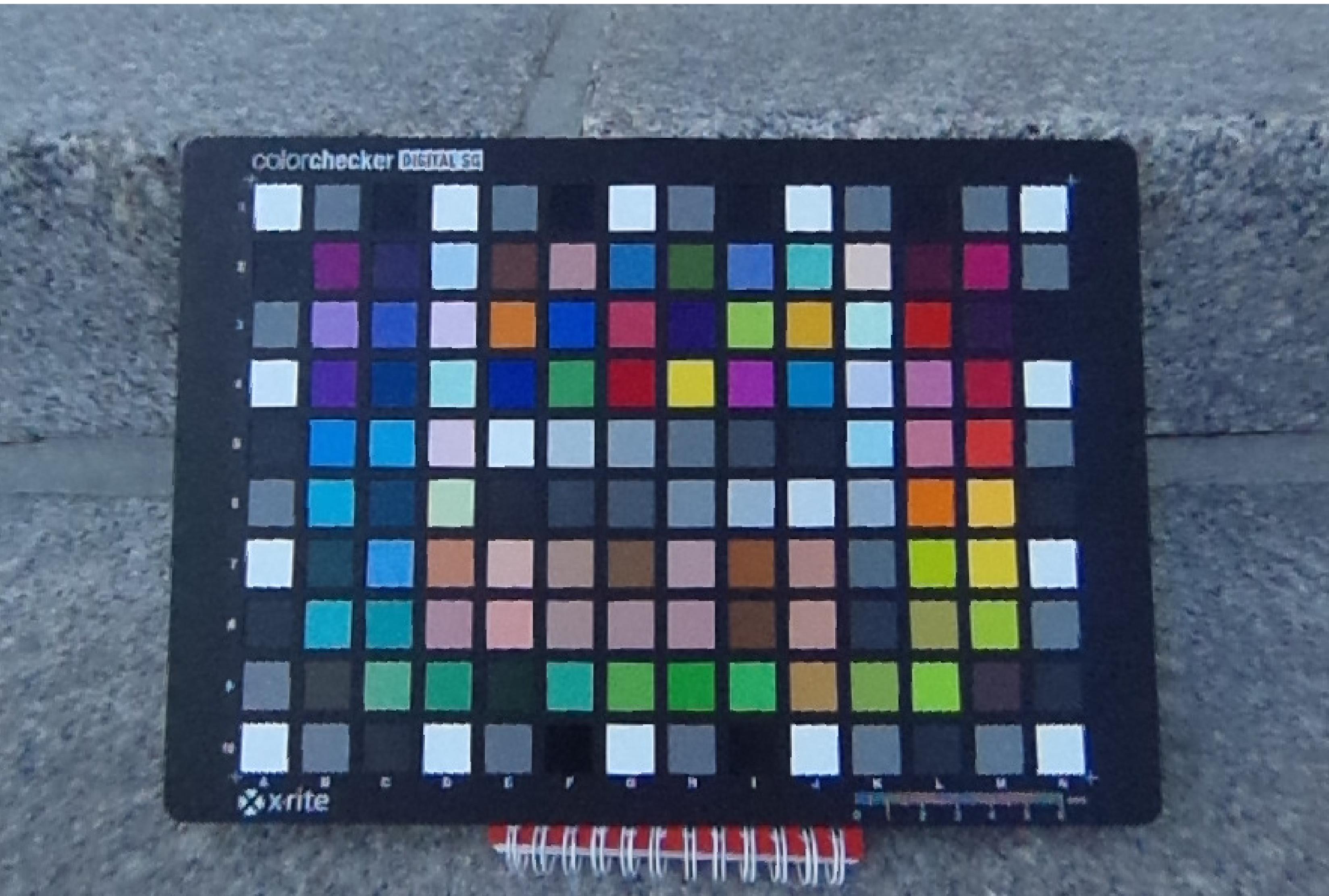
Fonction
non-linéaire



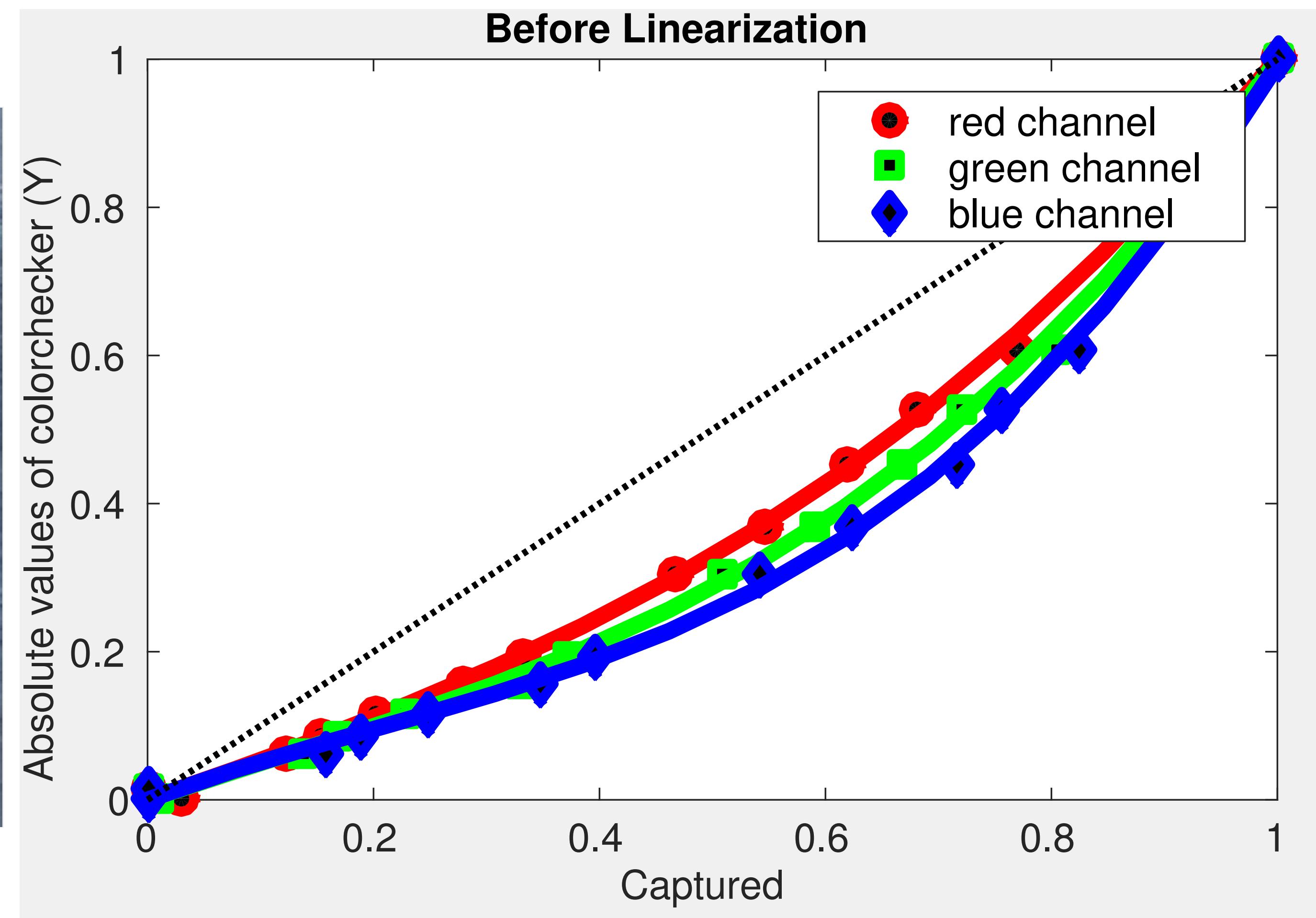
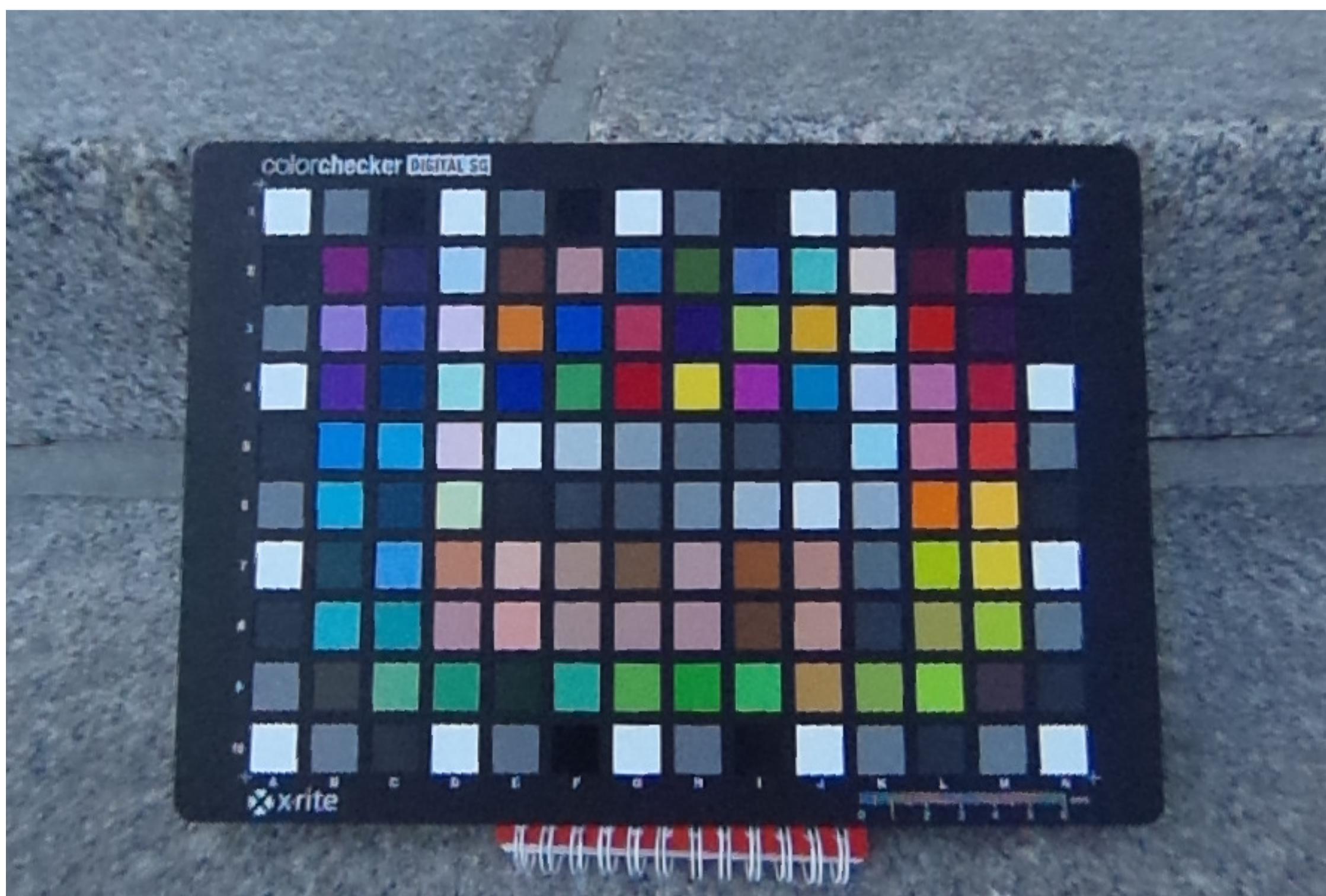
Pixel



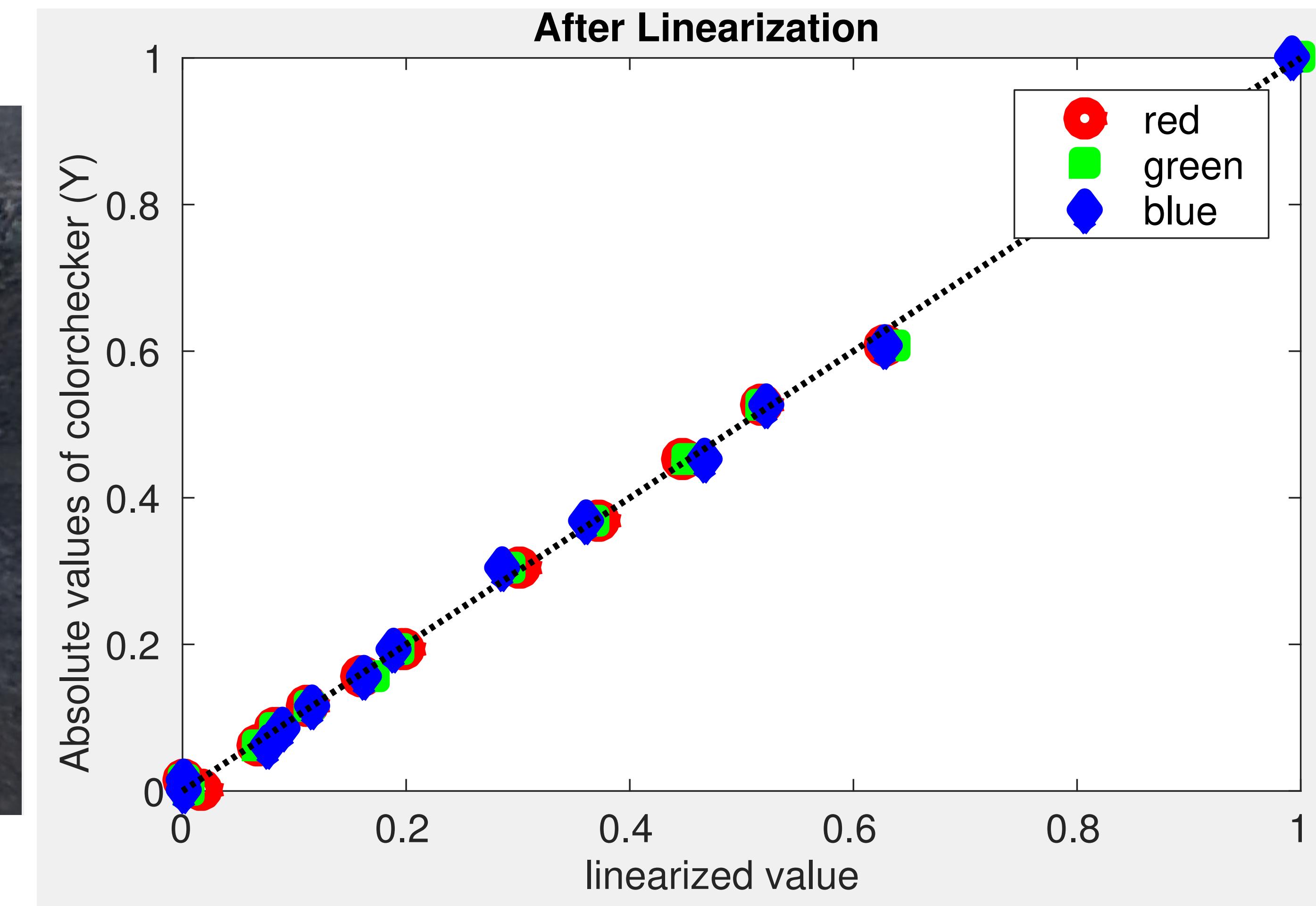
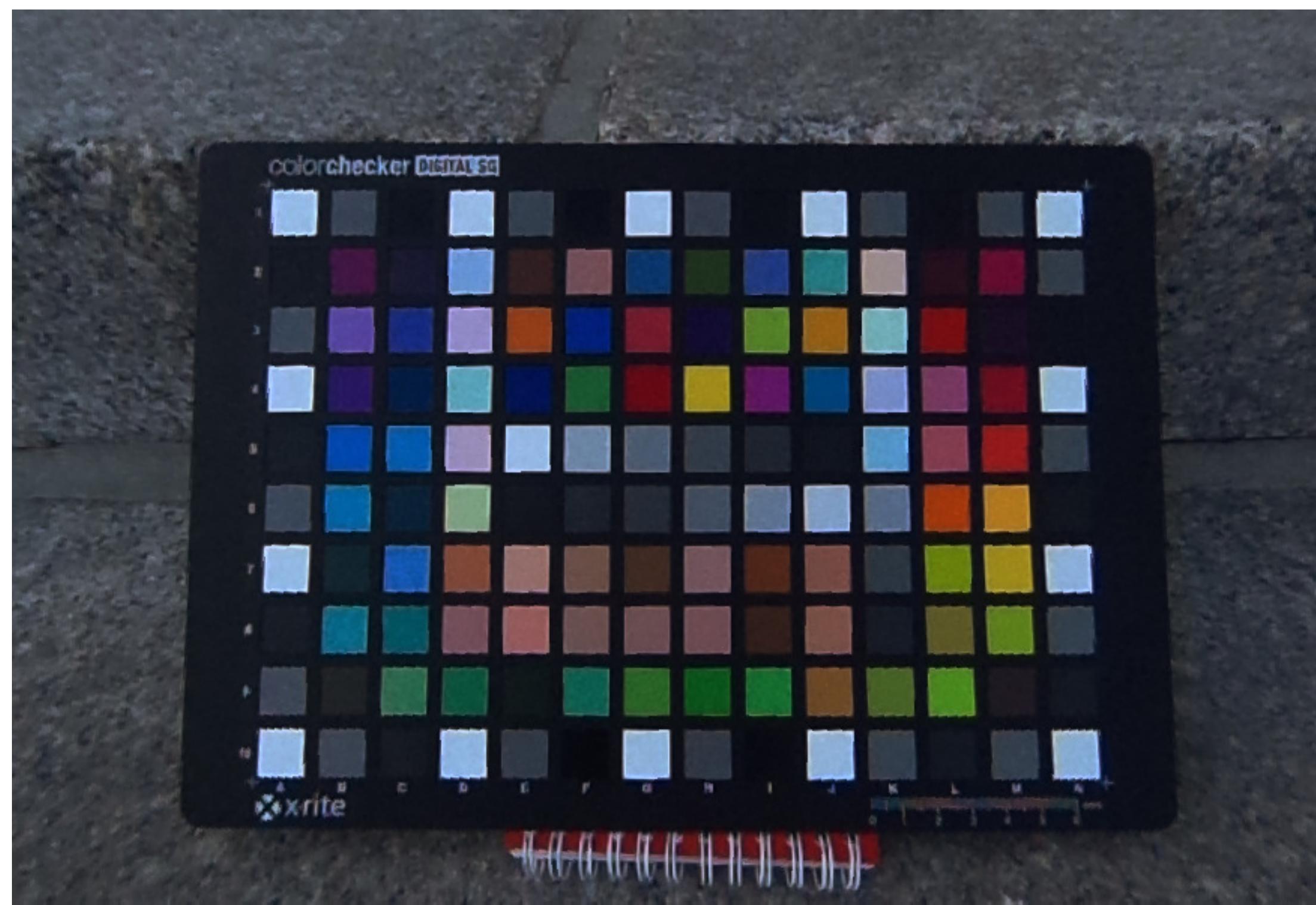
Calibrage radiométrique



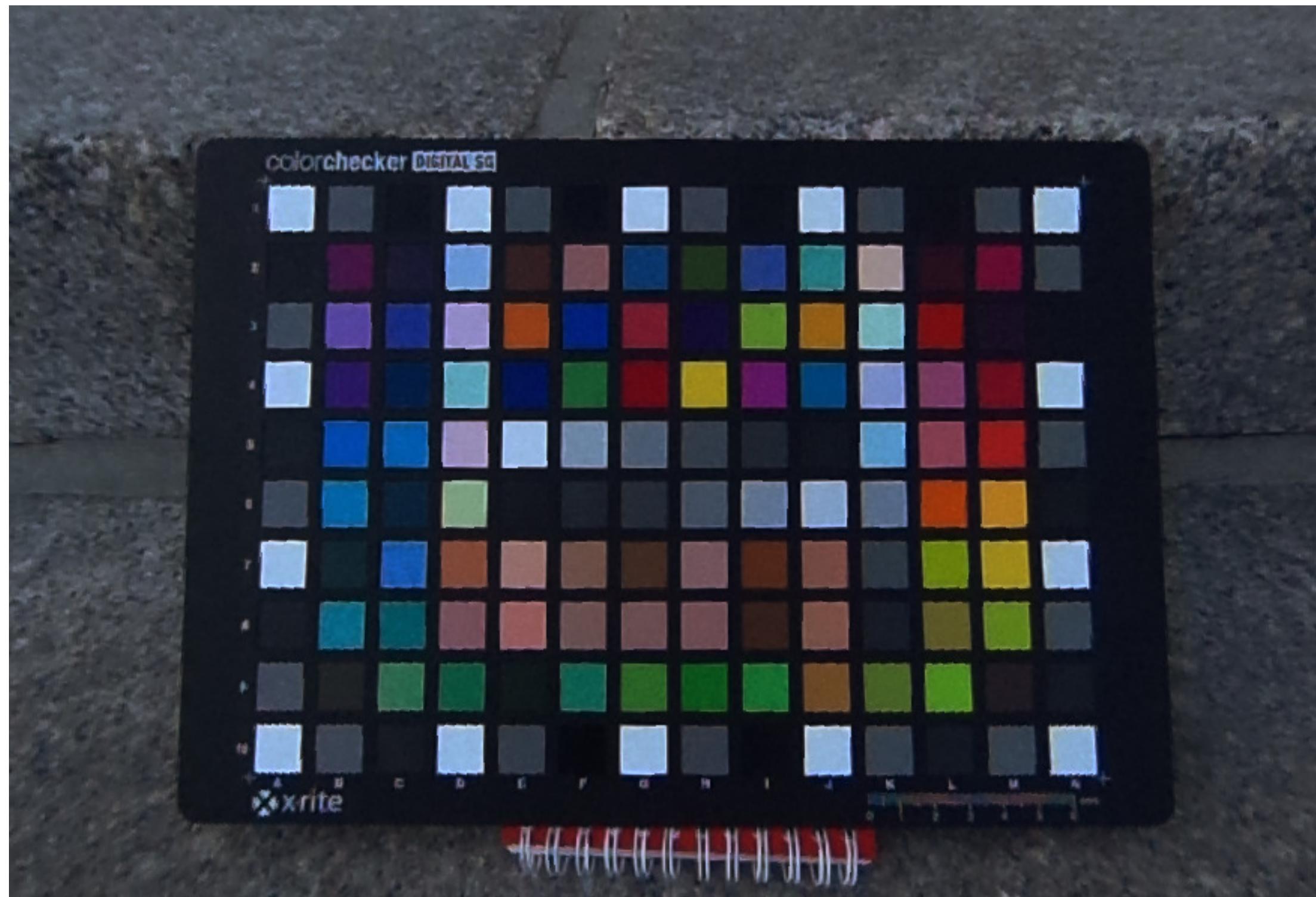
Calibrage radiométrique



Calibrage radiométrique



Calibrage radiométrique



Augmentation constante d'intensité **émise**



PHYSIQUE



Augmentation constante d'intensité **perçue**

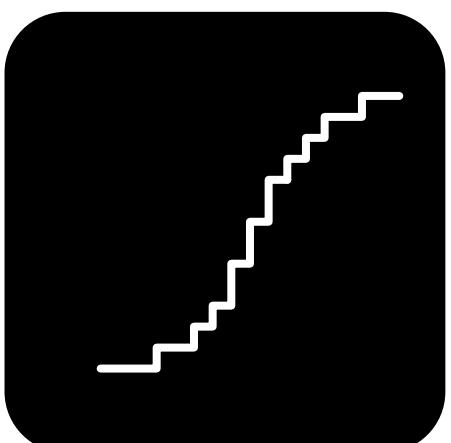


PERCEPTION

Dans une caméra...

- Comment estimer cette fonction?
- Première option
 - utiliser une cible de calibrage radiométrique
- Seconde option
 - l'estimer automatiquement à partir des images
 - article de P. Debevec (SIGGRAPH 1997)

Fonction
non-linéaire



Pixel



Images à haute plage dynamique

Calibrage radiométrique automatique

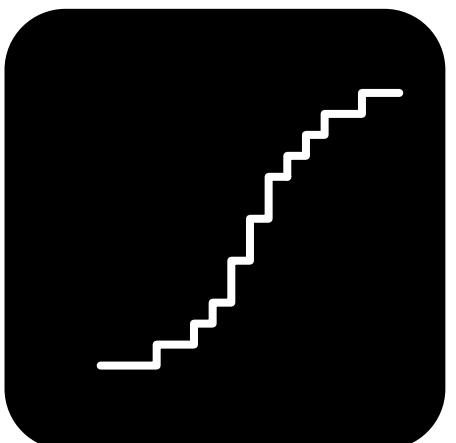


GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Dans une caméra...

- Comment estimer cette fonction?
- Première option
 - utiliser une cible de calibrage radiométrique
- Seconde option
 - l'estimer automatiquement à partir des images
 - article de P. Debevec (SIGGRAPH 1997)

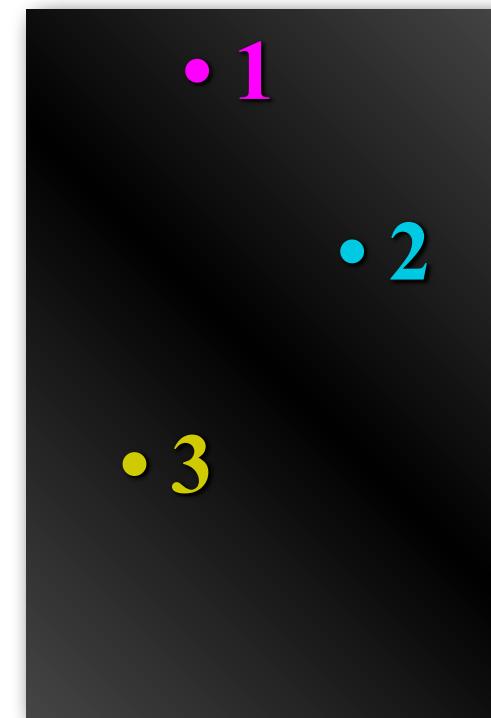
Fonction
non-linéaire



Pixel



Algorithme



$\Delta t = 1/64 \text{ sec}$

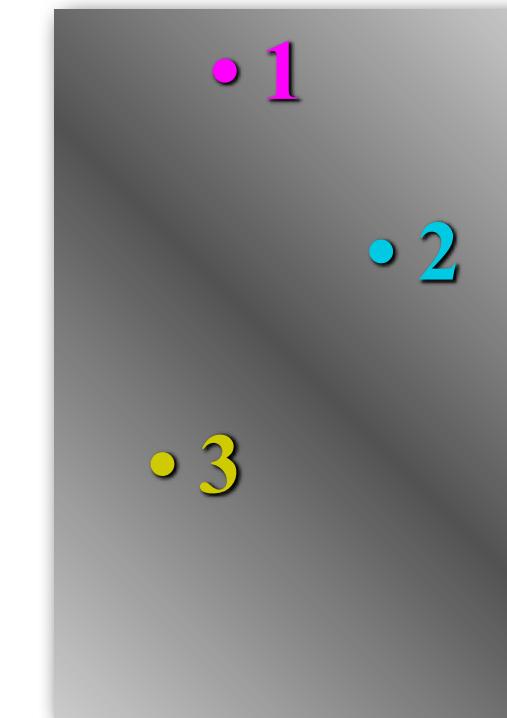


$\Delta t = 1/16 \text{ sec}$

Série d'images



$\Delta t = 1/4 \text{ sec}$



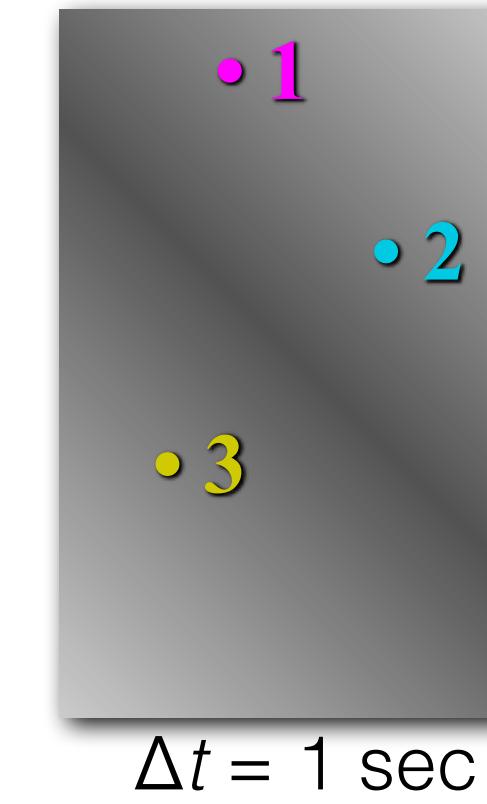
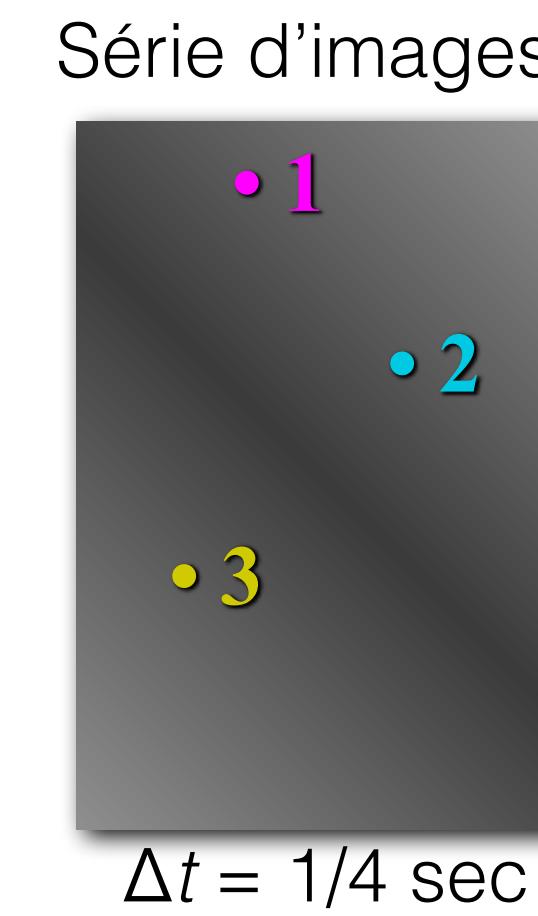
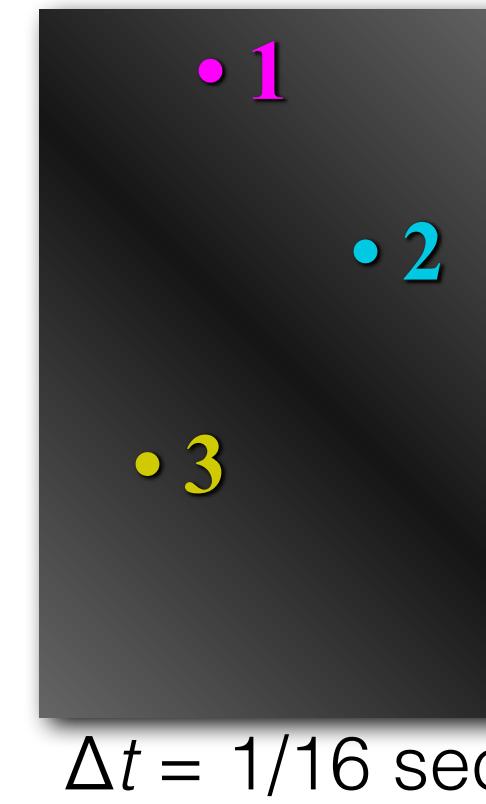
$\Delta t = 1 \text{ sec}$



$\Delta t = 4 \text{ sec}$

$$z = f(\text{radiance} \times \Delta t)$$

Algorithme



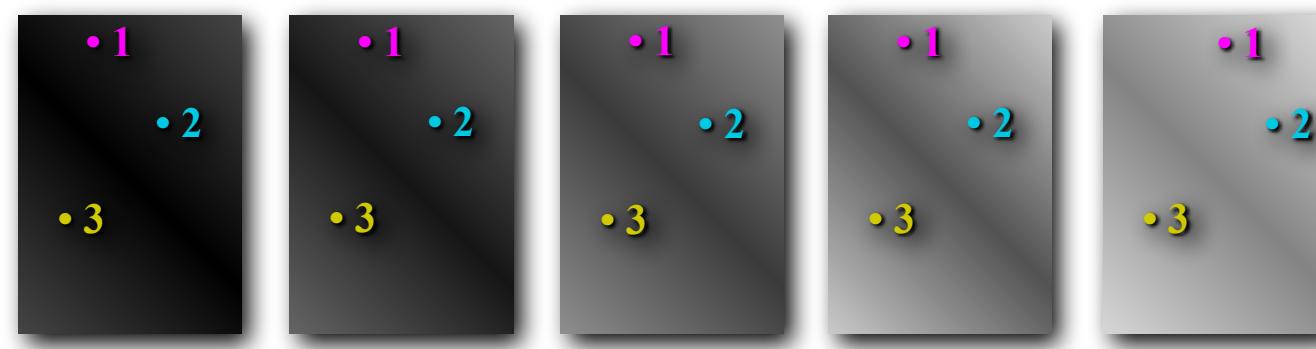
Série d'images

$j = \text{image}$
 $i = \text{pixel}$

$$z_{ij} = f(\text{radiance}_i \times \Delta t_j)$$

$$f^{-1}(z_{ij}) = \text{radiance}_i \times \Delta t_j$$

$$g(z_{ij}) = \log \text{radiance}_i + \log \Delta t_j$$

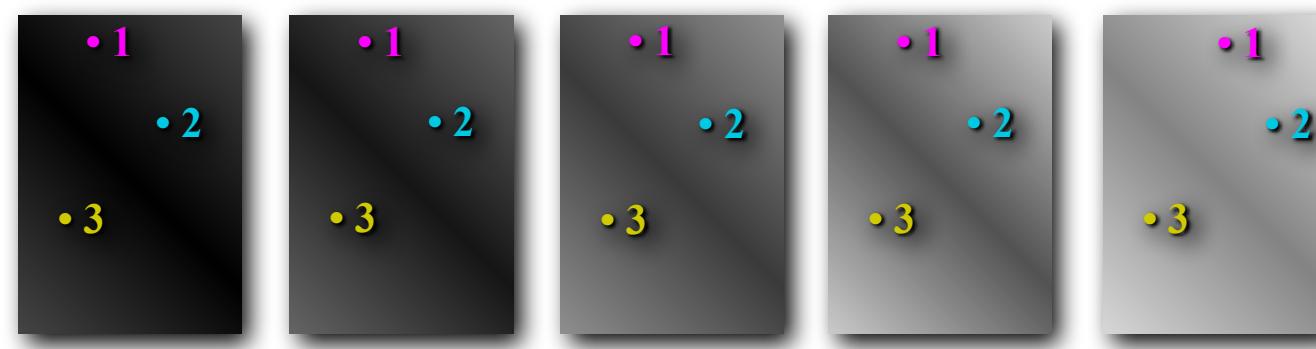


$$g(z_{ij}) = \log \text{radiance}_i + \log \Delta t_j$$

$j = \text{image}$
 $i = \text{pixel}$

Quels sont les connus/inconnus?

Astuce : discréteriser la fonction g()



$$g(z_{ij}) = \log \text{radiance}_i + \log \Delta t_j$$
$$g(z_{ij}) = r_i + t_j$$

$j = \text{image}$
 $i = \text{pixel}$

$$g(0) \quad g(1) \quad g(2) \quad g(3) \quad \dots \quad g(255) \quad r_1 \quad r_2 \quad r_3 \quad \dots \quad r_N$$

A

$$\begin{bmatrix} g(0) \\ g(1) \\ g(2) \\ g(3) \\ \vdots \\ \del{g(255)} \\ r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_N \end{bmatrix} = \begin{bmatrix} \mathbf{b} \end{bmatrix}$$

Comment trouver \mathbf{x}

- Nous avons un système d'équations linéaires de la forme

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

où \mathbf{A} est « sous-constraint »

(car on peut ajouter n'importe quelle constante à \mathbf{x} !)

- En python, on peut le résoudre avec la fonction

```
from scipy import linalg  
x = linalg.lstsq(A, b)
```

qui trouve le \mathbf{x} qui minimise

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$$

Comment trouver \mathbf{x}

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

Autre façon de l'écrire :

$$\arg \min_{g,r} \sum_{i \in \text{pixels}} \sum_{j \in \text{images}} (r_i + t_j - g(z_{ij}))^2$$

Quel est le problème?

Que faire si, par exemple, aucun pixel ne possède la valeur 35?

Comment trouver \mathbf{x}

Contrainte : courbe doit être lisse

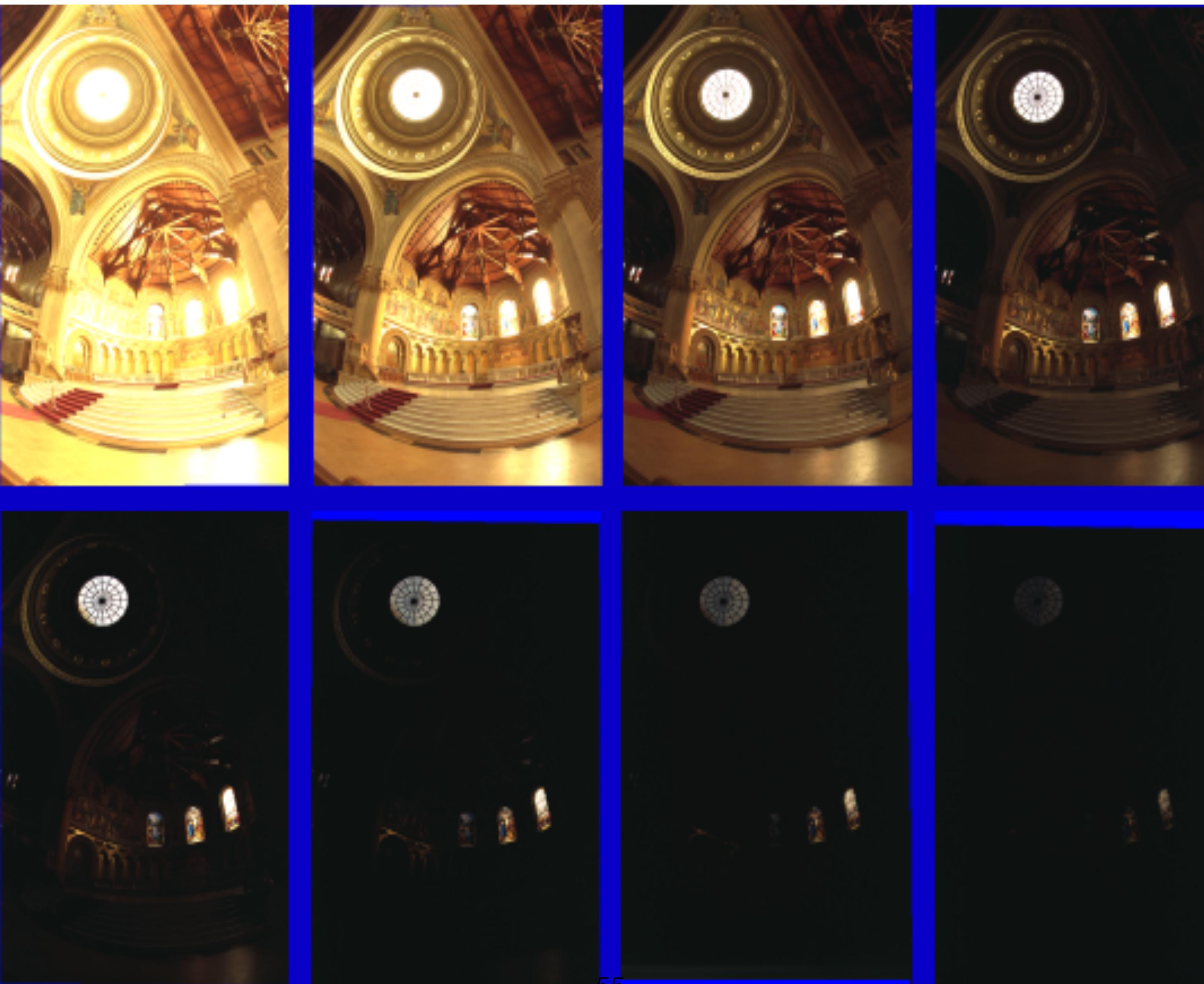
$$\arg \min_{g,r} \sum_{i \in \text{pixels}} \sum_{j \in \text{images}} (r_i + t_j - g(z_{ij}))^2 + \lambda \sum_{z=0}^{255} g''(z)^2$$

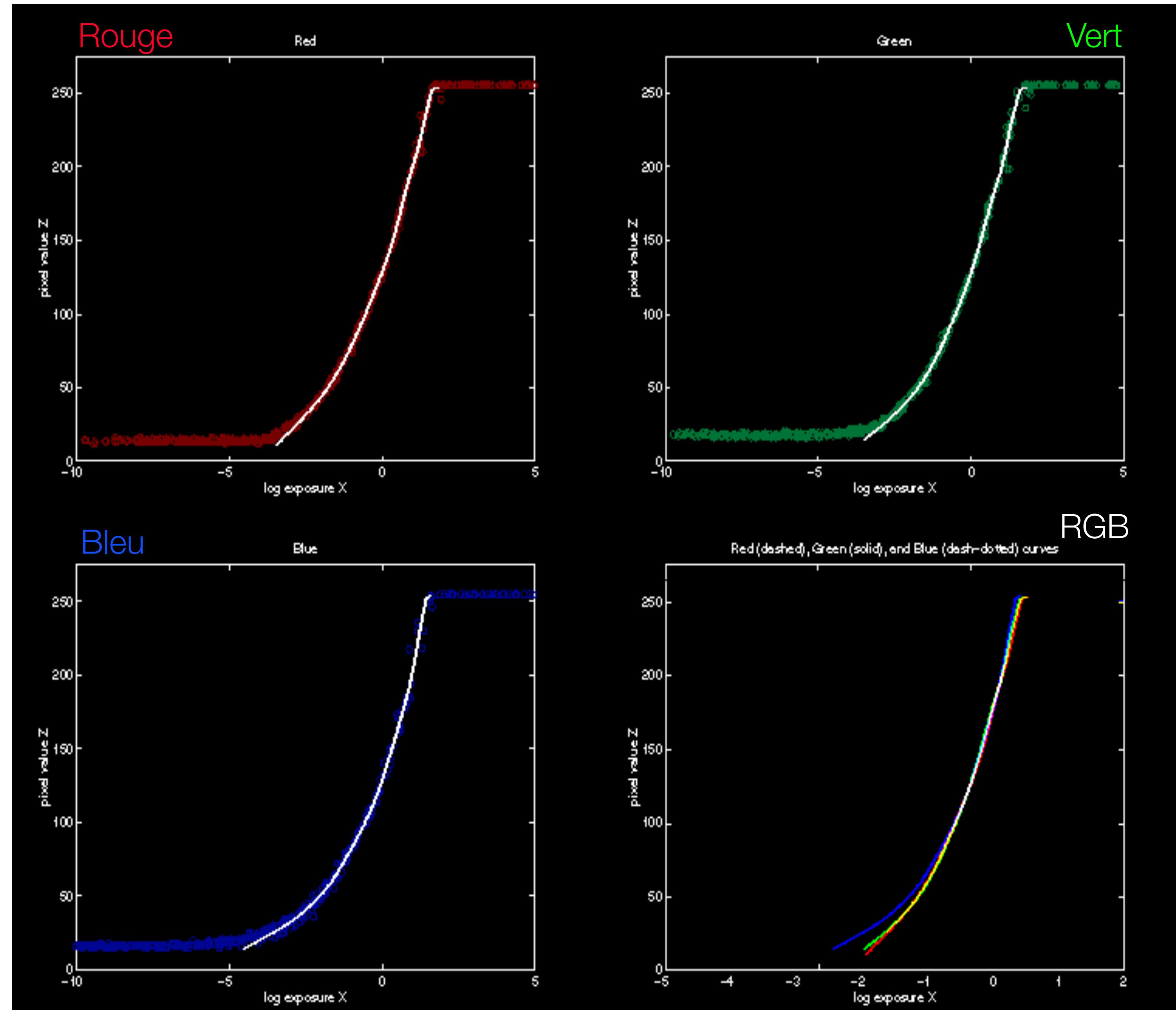
Contrainte : courbe doit être lisse

$$+ \lambda \sum_{z=0}^{255} g''(z)^2$$

$$\begin{bmatrix} g(0) & g(1) & g(2) & g(3) & \cdots & g(255) & r_1 & r_2 & r_3 & \cdots & r_N \end{bmatrix} = \begin{bmatrix} g(0) \\ g(1) \\ g(2) \\ g(3) \\ \vdots \\ g(255) \\ r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_N \end{bmatrix}$$

Résultats: couleur





Math

- Notons la (log-)fonction inverse discrétisée : $g(z)$
- Pour chaque pixel i dans une image j , nous avons :

$$g(z_{ij}) = \log \text{radiance}_i + \log \Delta t_j$$

$$g(z_{ij}) = r_i + t_j$$

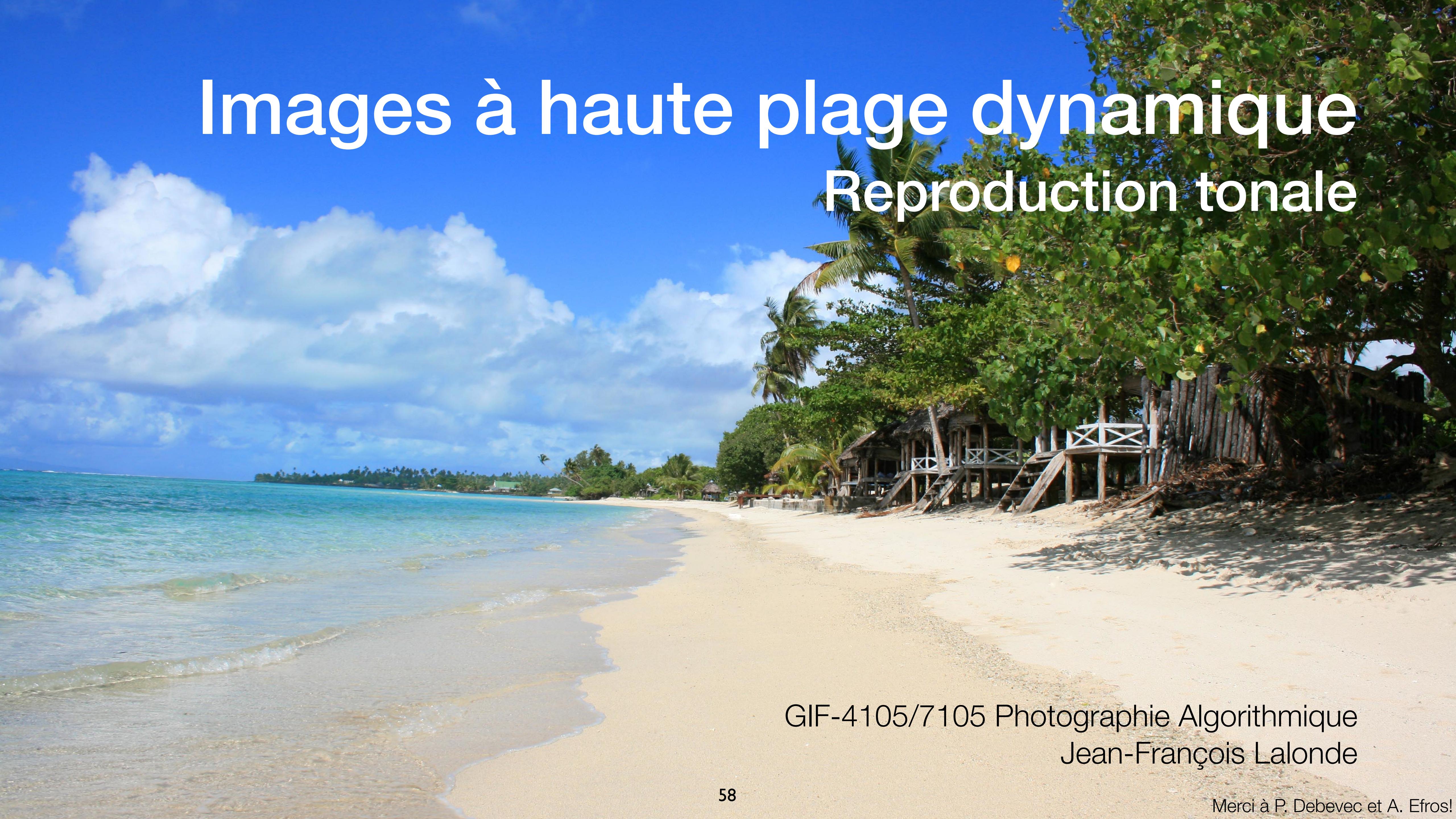
- Système d'équations linéaires sur-contraint :

$$\sum_{i \in \text{pixels}} \sum_{j \in \text{images}} (r_i + t_j - g(z_{ij}))^2 + \lambda \sum_{z=0}^{255} g''(z)^2$$

approximation des données courbe lisse

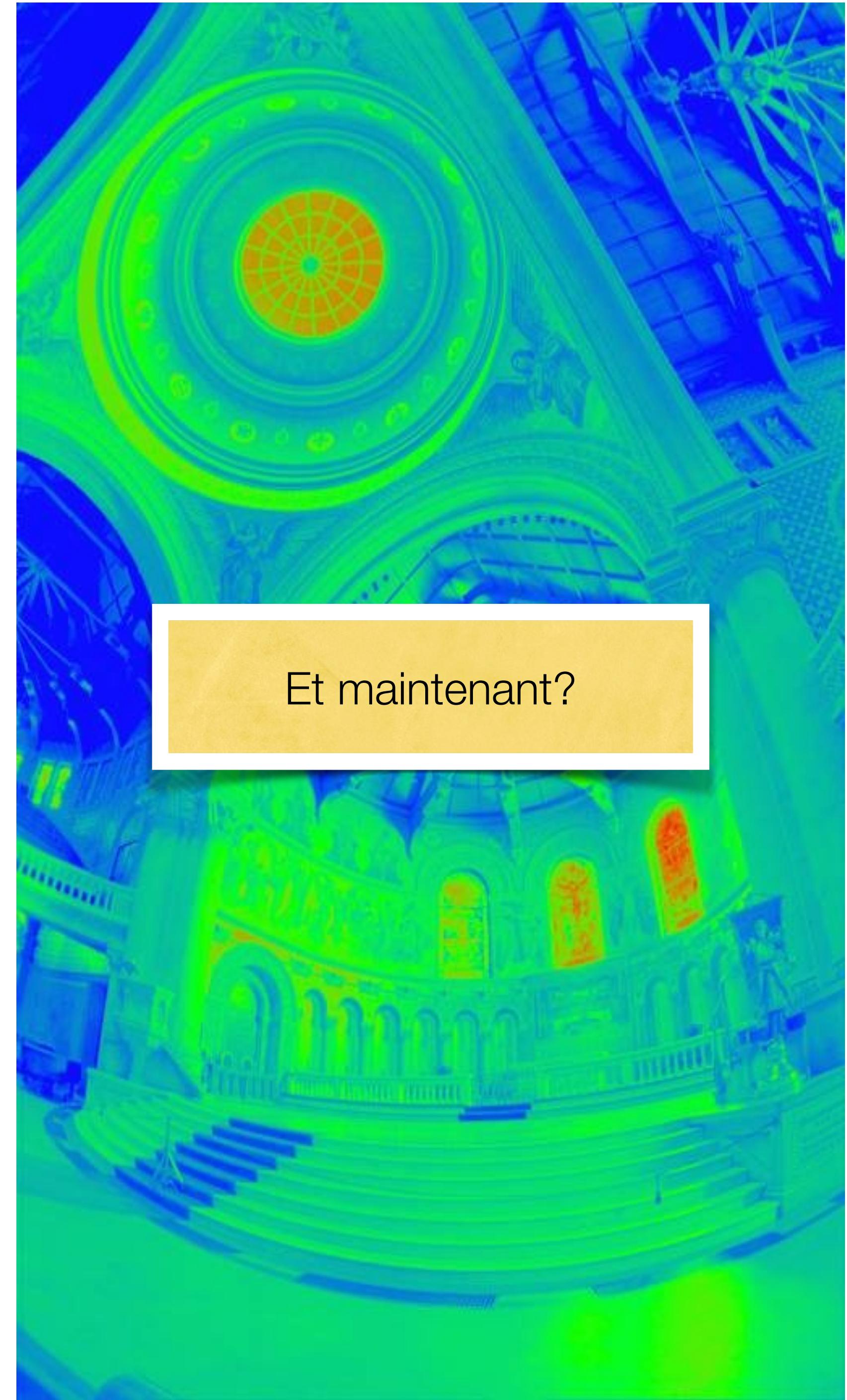
- Permet d'estimer la radiance et la fonction de caméra simultanément!

Images à haute plage dynamique Reproduction tonale



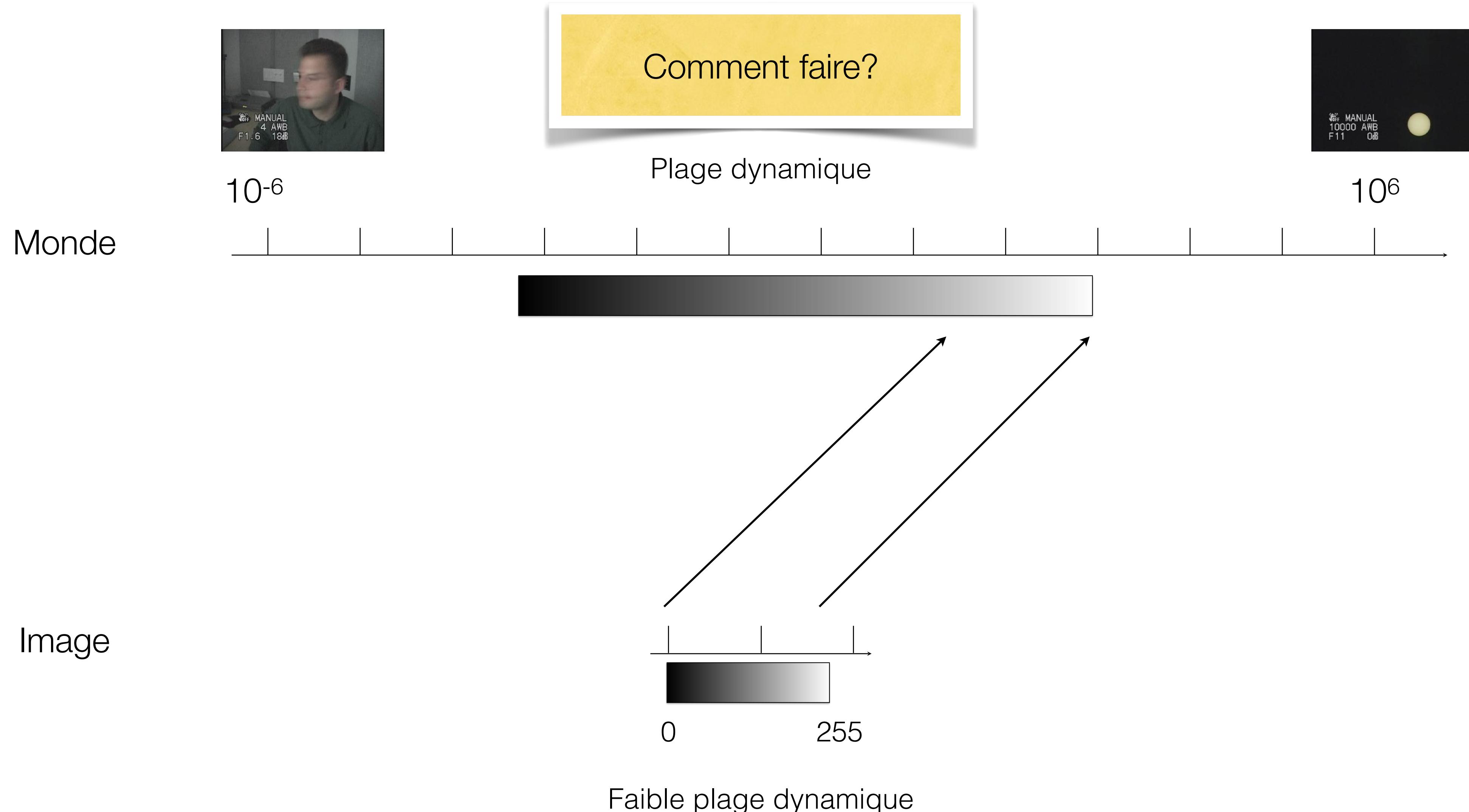
GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Radiance



$$\text{radiance} \propto \frac{\text{pixel}}{\Delta t}$$

Reproduction tonale



Reproduction tonale

- Problème typique :
 - Une caméra (de bonne qualité) peut capturer des valeurs d'intensité de 12 bits, mais nos écrans ne peuvent qu'en afficher 8 !

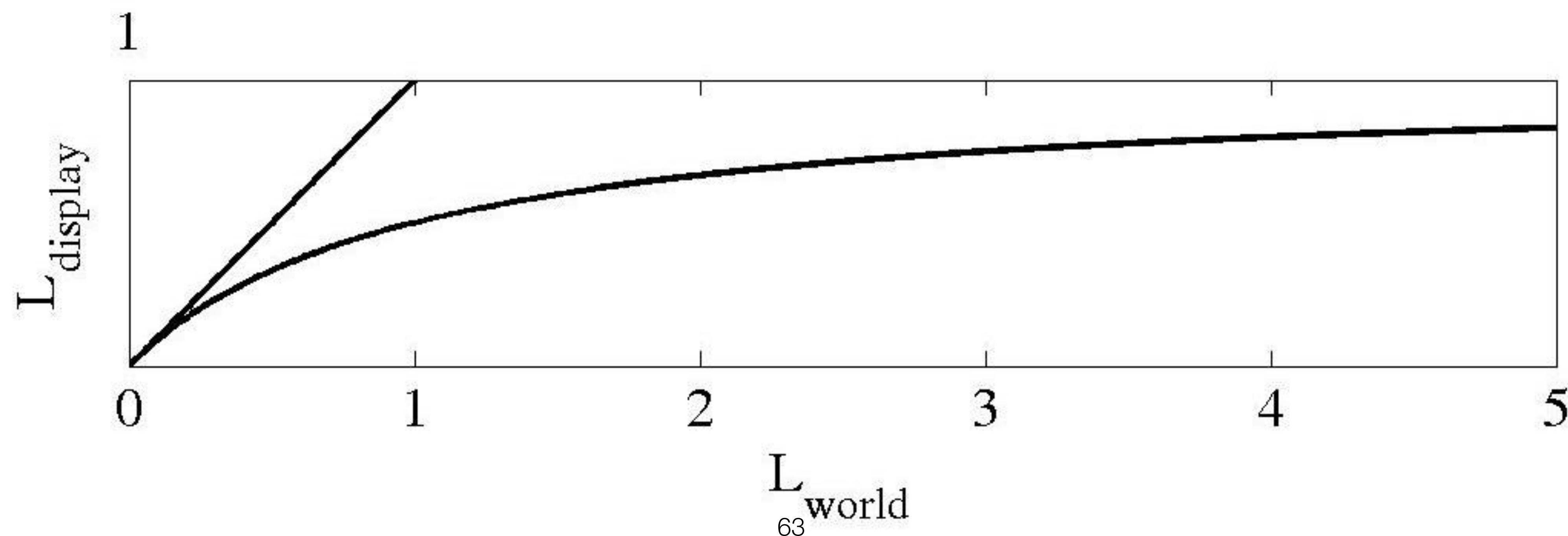
Opérateur global

- Déterminer une courbe qui :
 - Ramène le contenu du signal HDR dans une plage qui convient à un écran ou un projecteur
 - N'augmente pas les parties sombres
- Donc:
 - Asymptote à 255
 - Dérivée = 1 à 0

Idée 1 : opérateur global (Reinhard et al.)

- Solution simple : utiliser une transformée **non-linéaire**

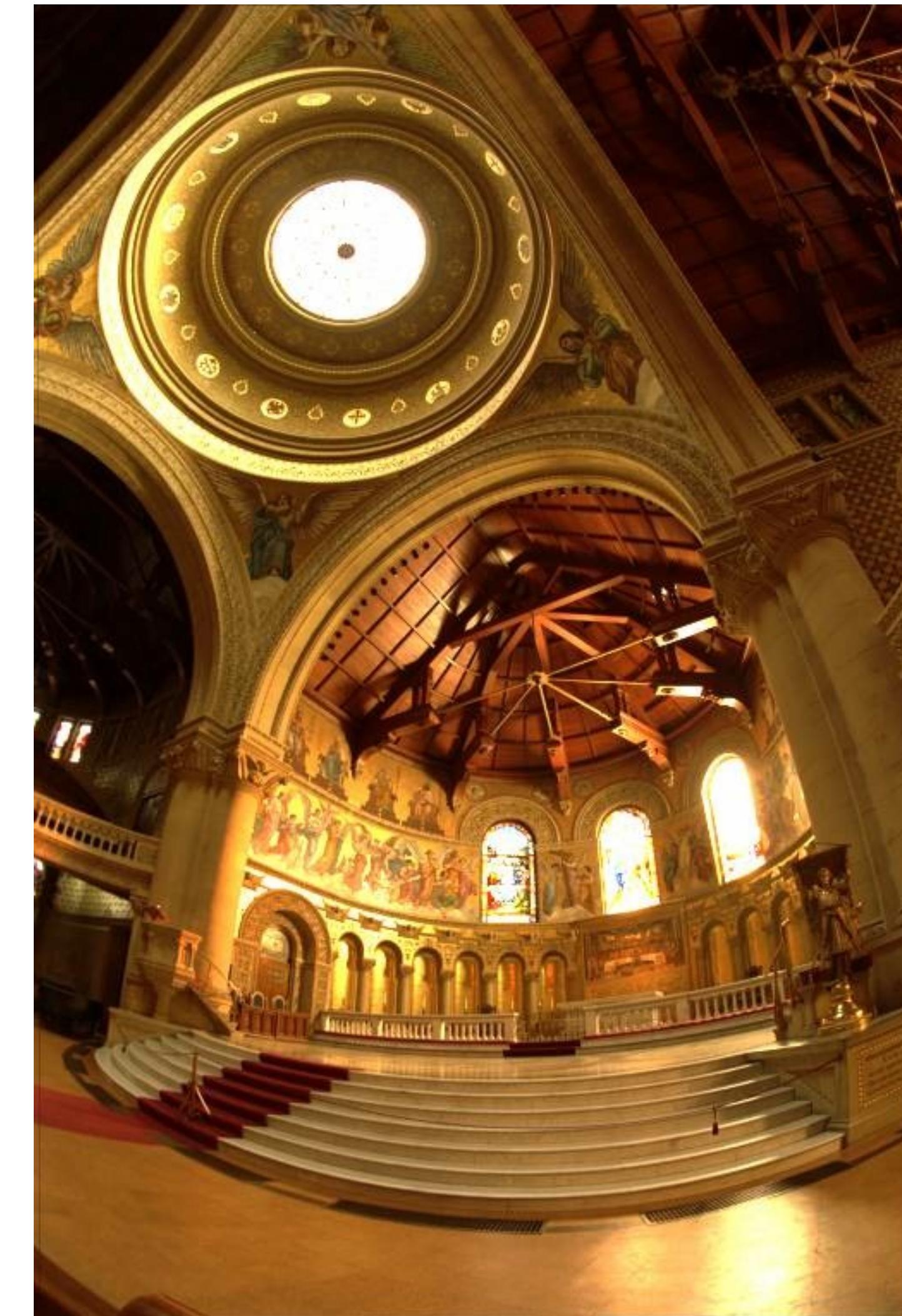
$$L_{display} = \frac{L_{world}}{1 + L_{world}}$$



Non-linéaire



Reinhard

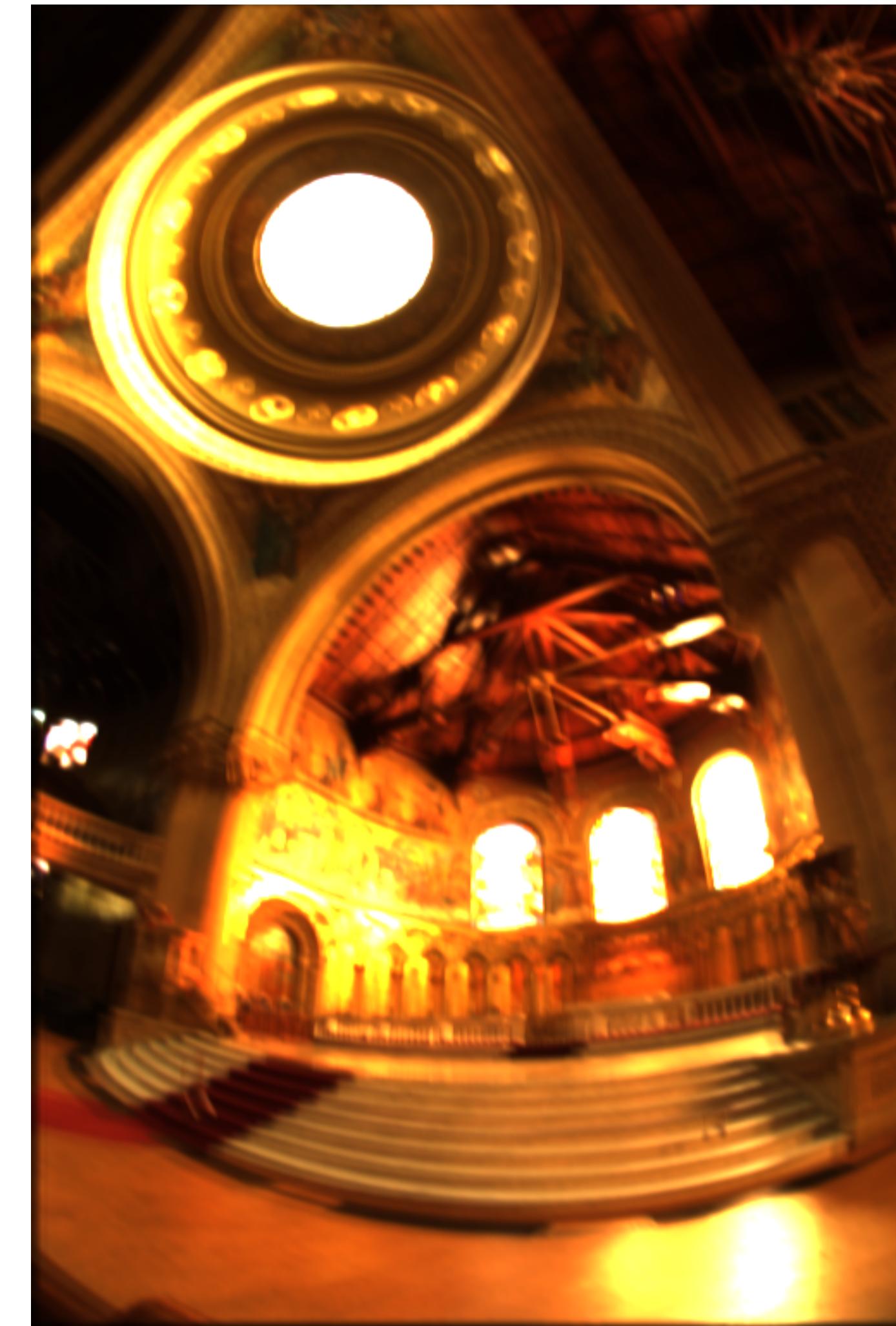


En fonction des pixels
les plus sombres

Idée 2 : utiliser le filtre bilatéral



filtre bilatéral



Filtre bilatéral : deux filtres gaussiens!

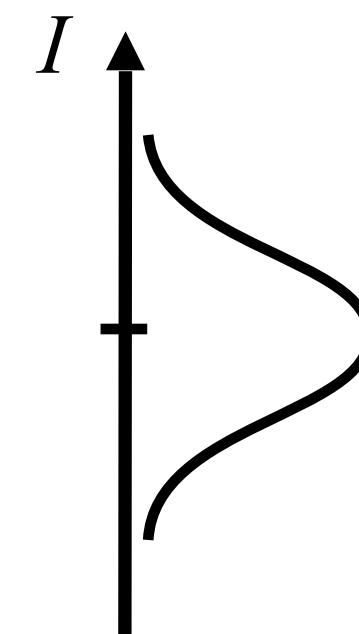
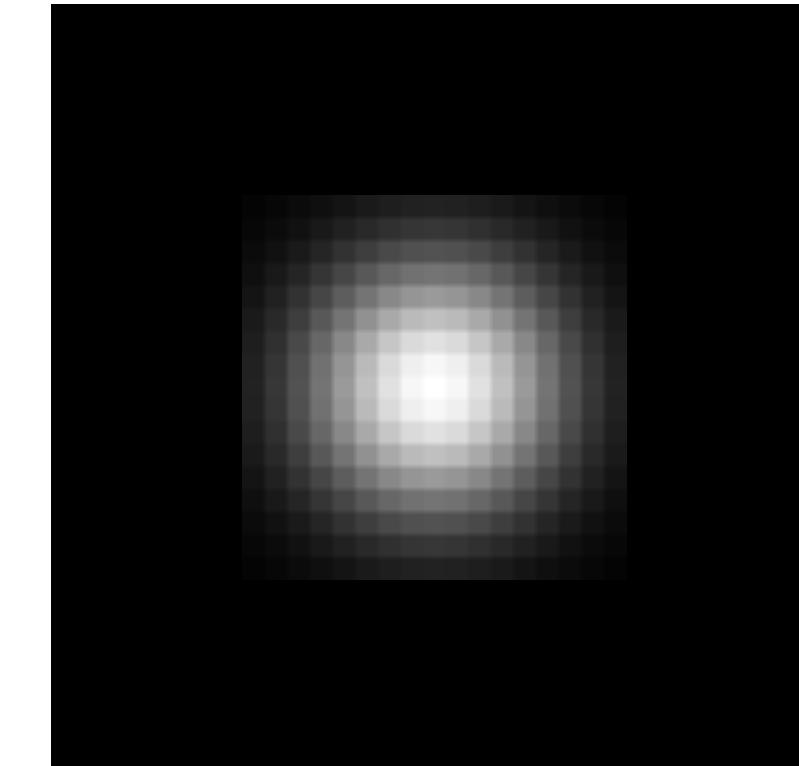
mais pas sur le même domaine!

$$F[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)I_{\mathbf{q}}$$

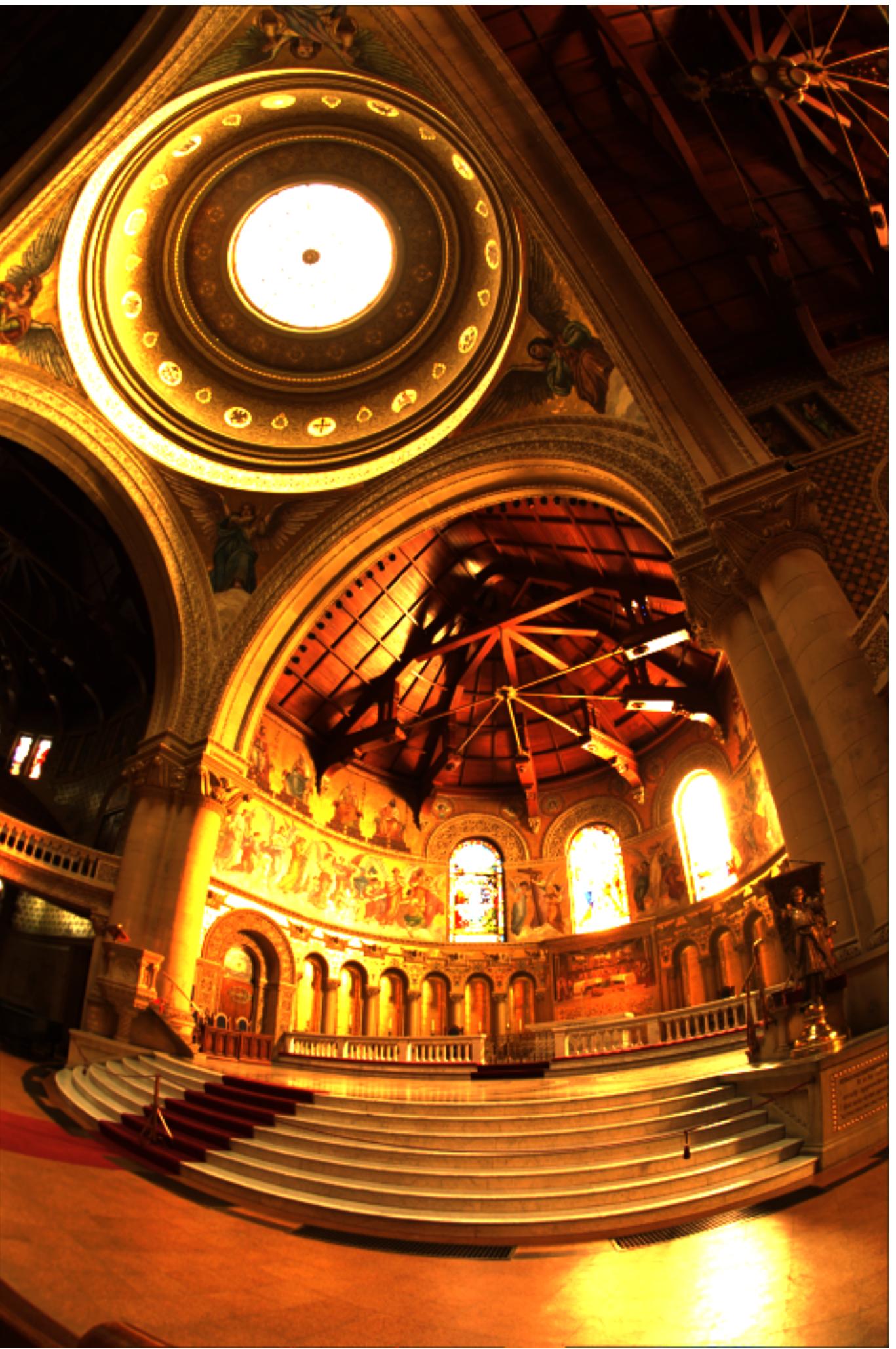
normalisation

pondération spatiale

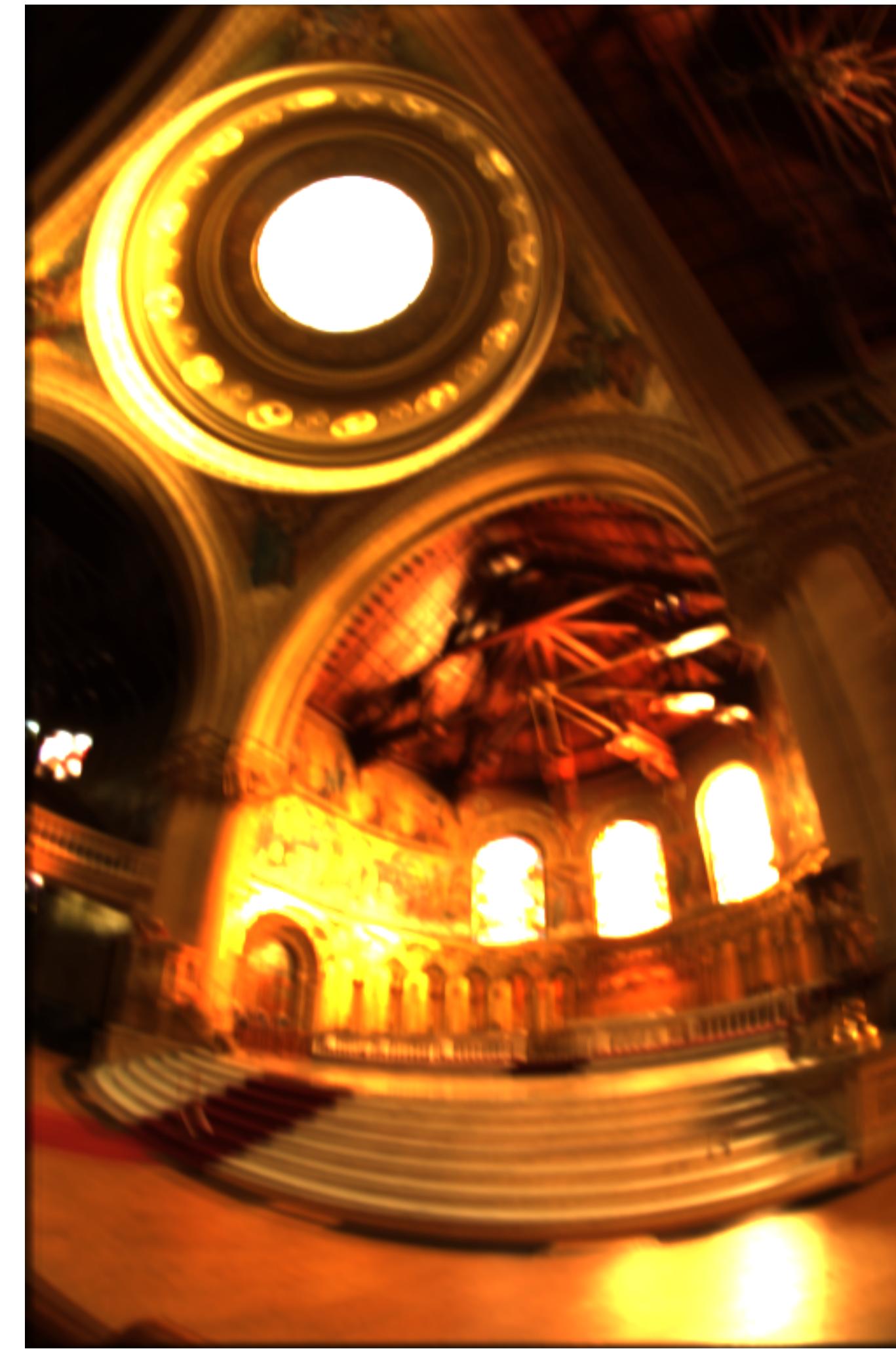
pondération en intensité



Idée 2 : filtre bilatéral



filtre bilatéral



Idée 2 : filtre bilatéral

image originale

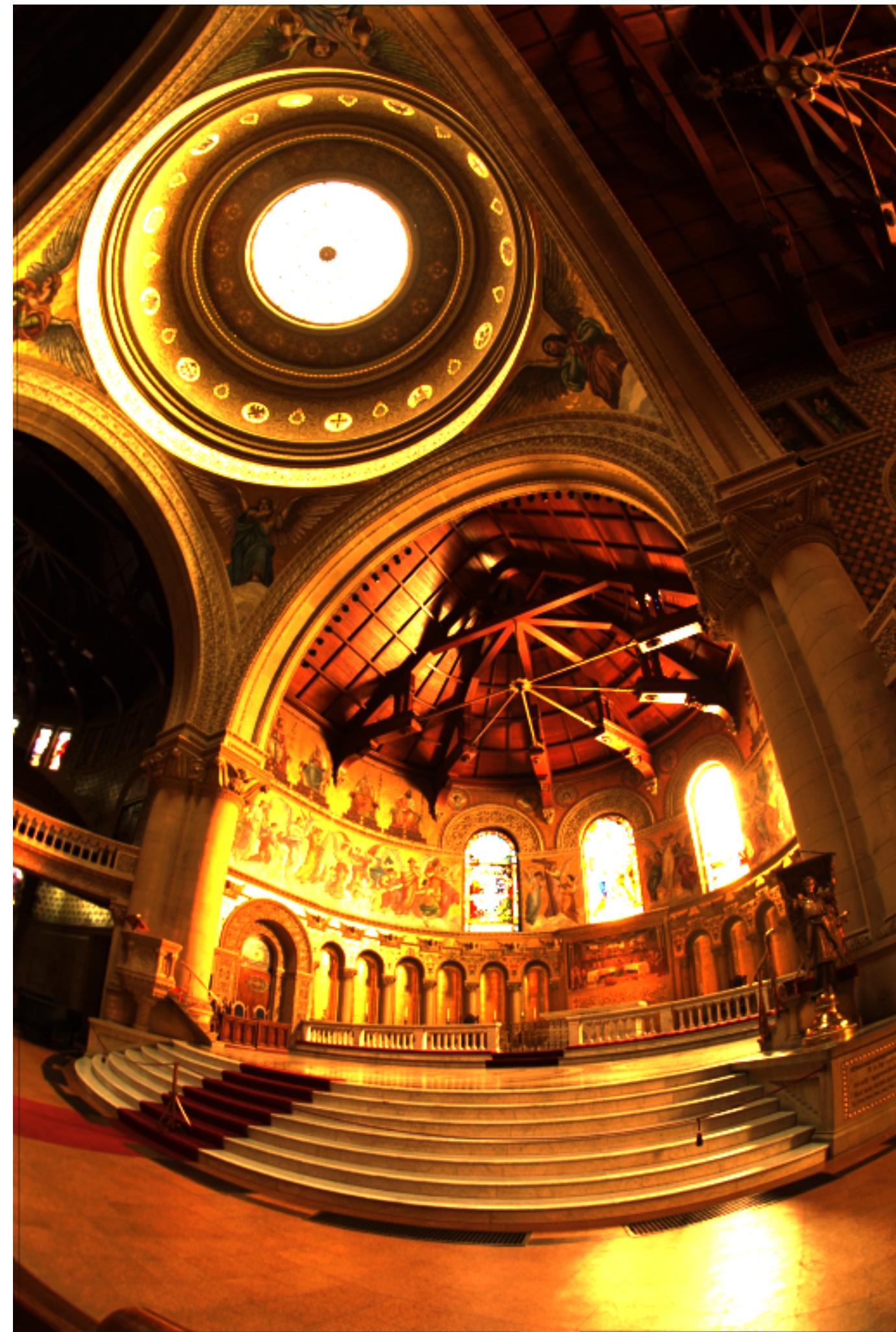
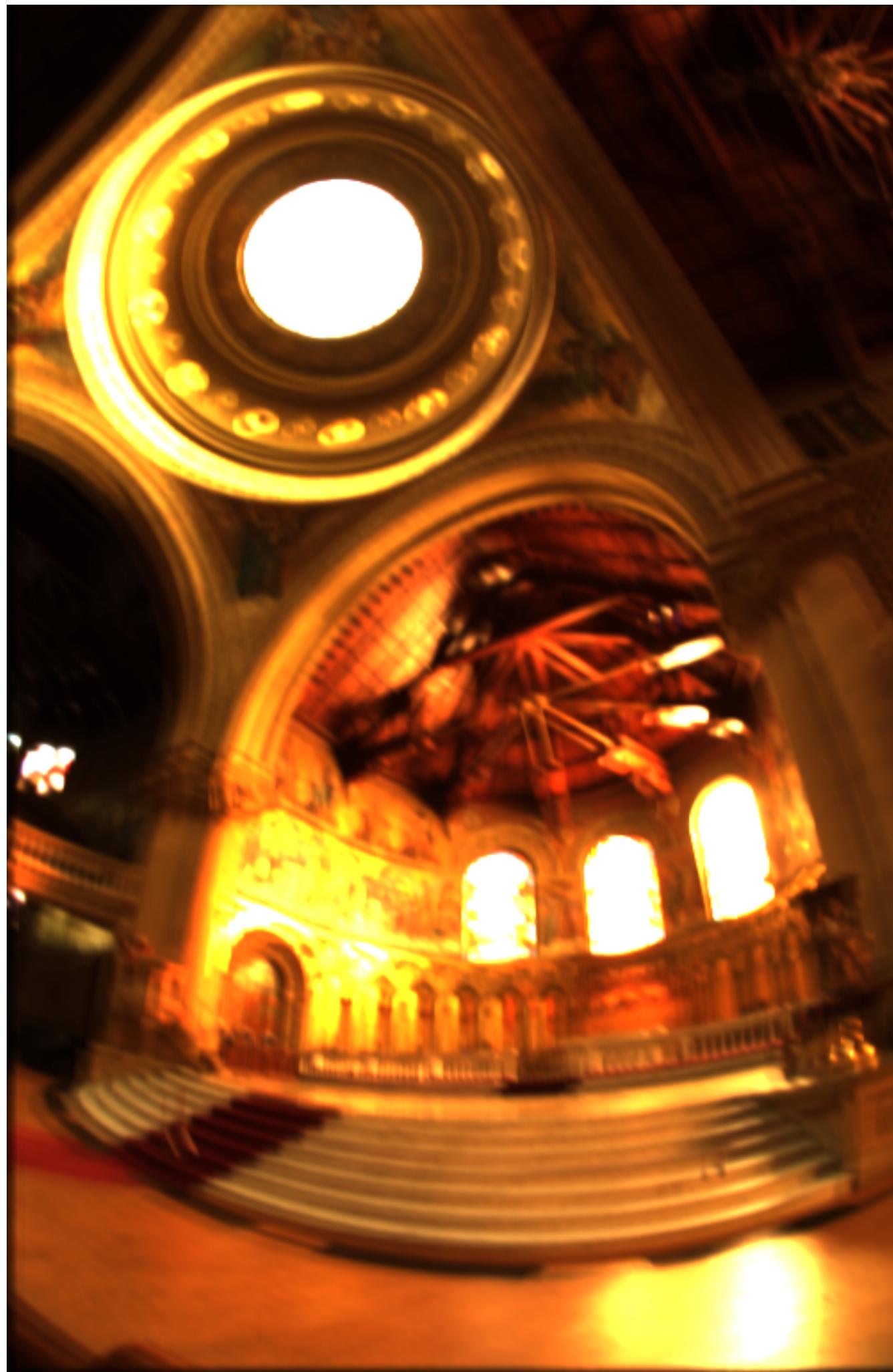


image filtrée (base)



détails = img / base



Idée 2 : filtre bilatéral

image originale

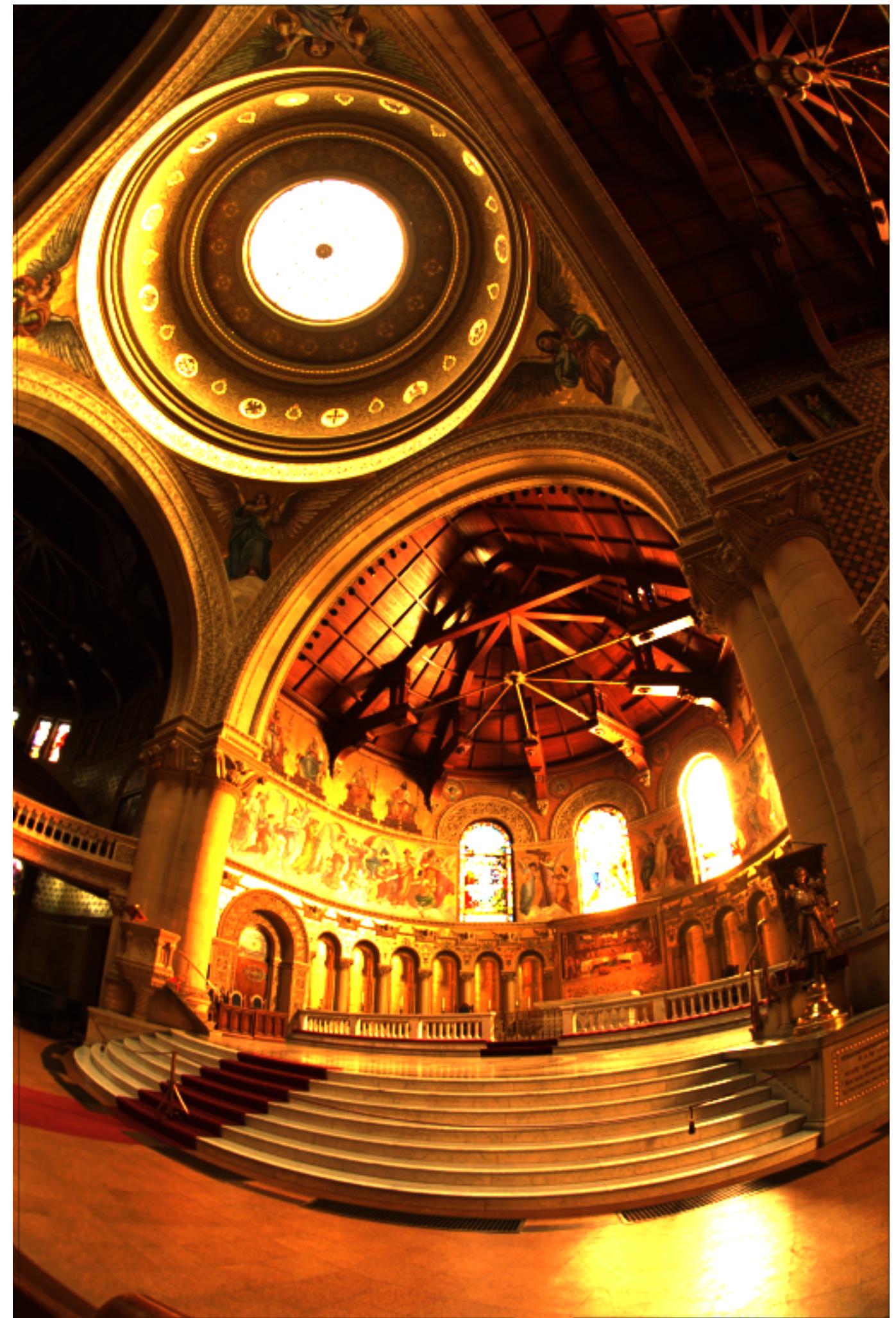
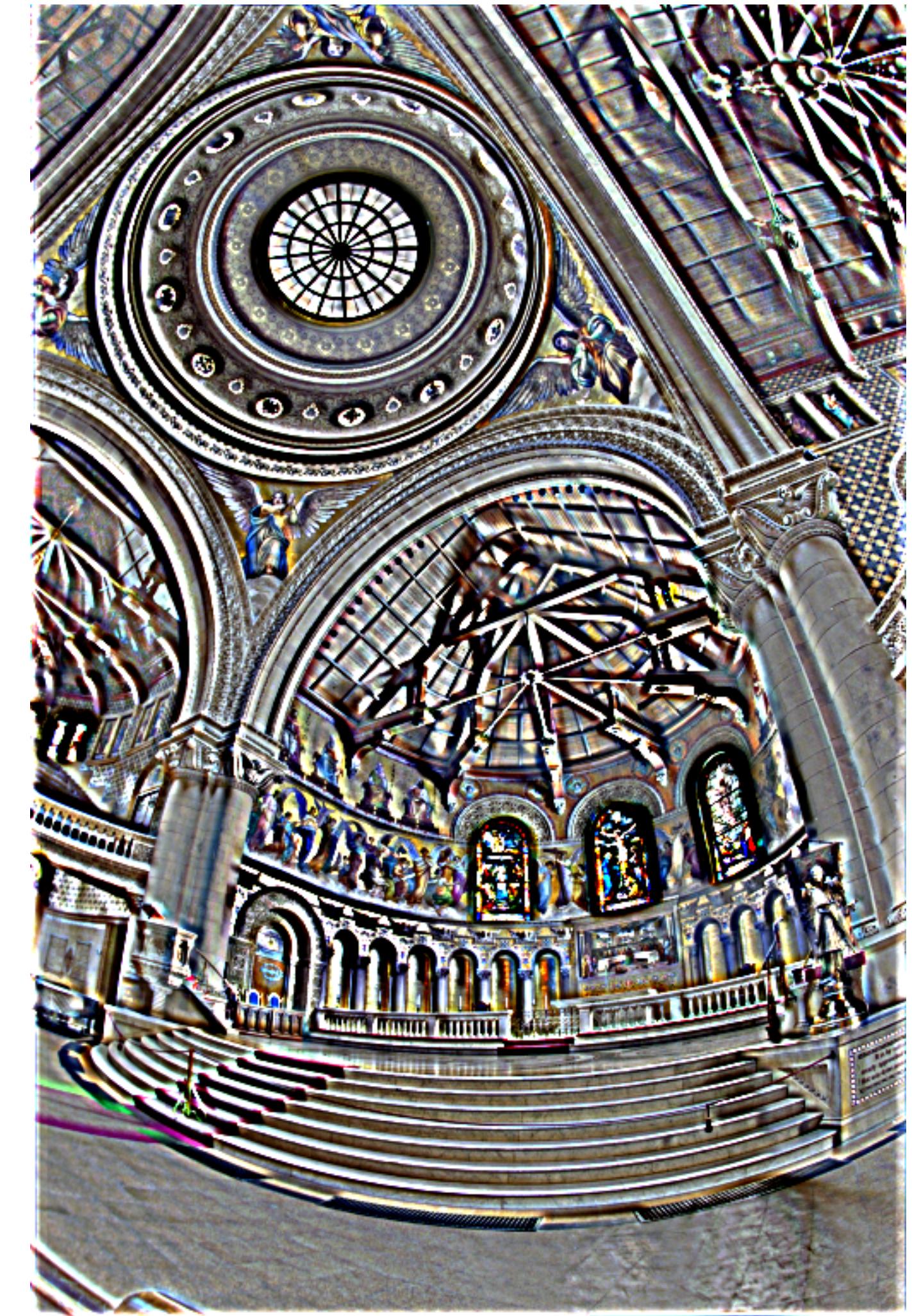


image filtrée (compressée)



détails = img / base



Idée 2 : filtre bilatéral

image originale



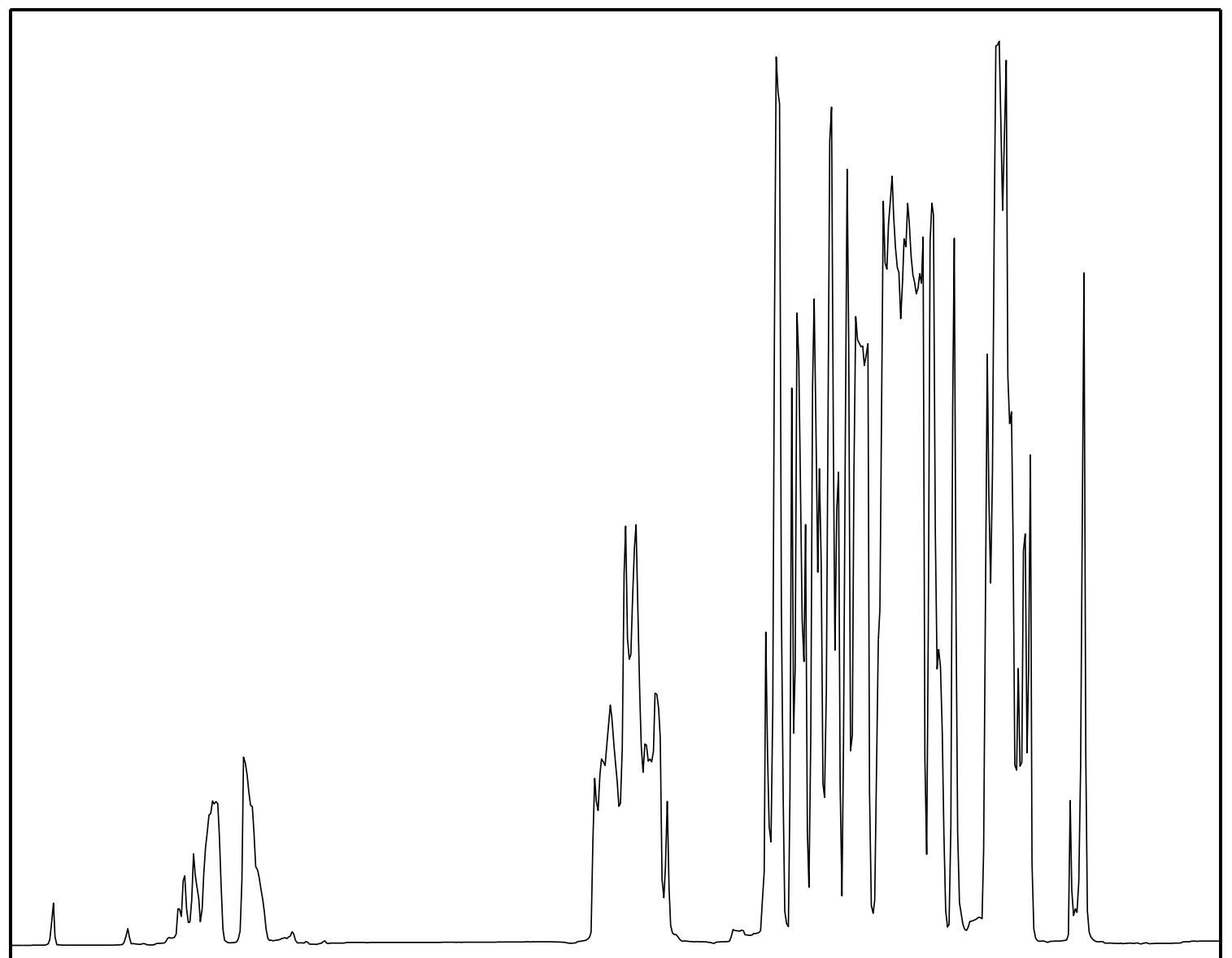
image filtrée (compressée)



détails = img / base

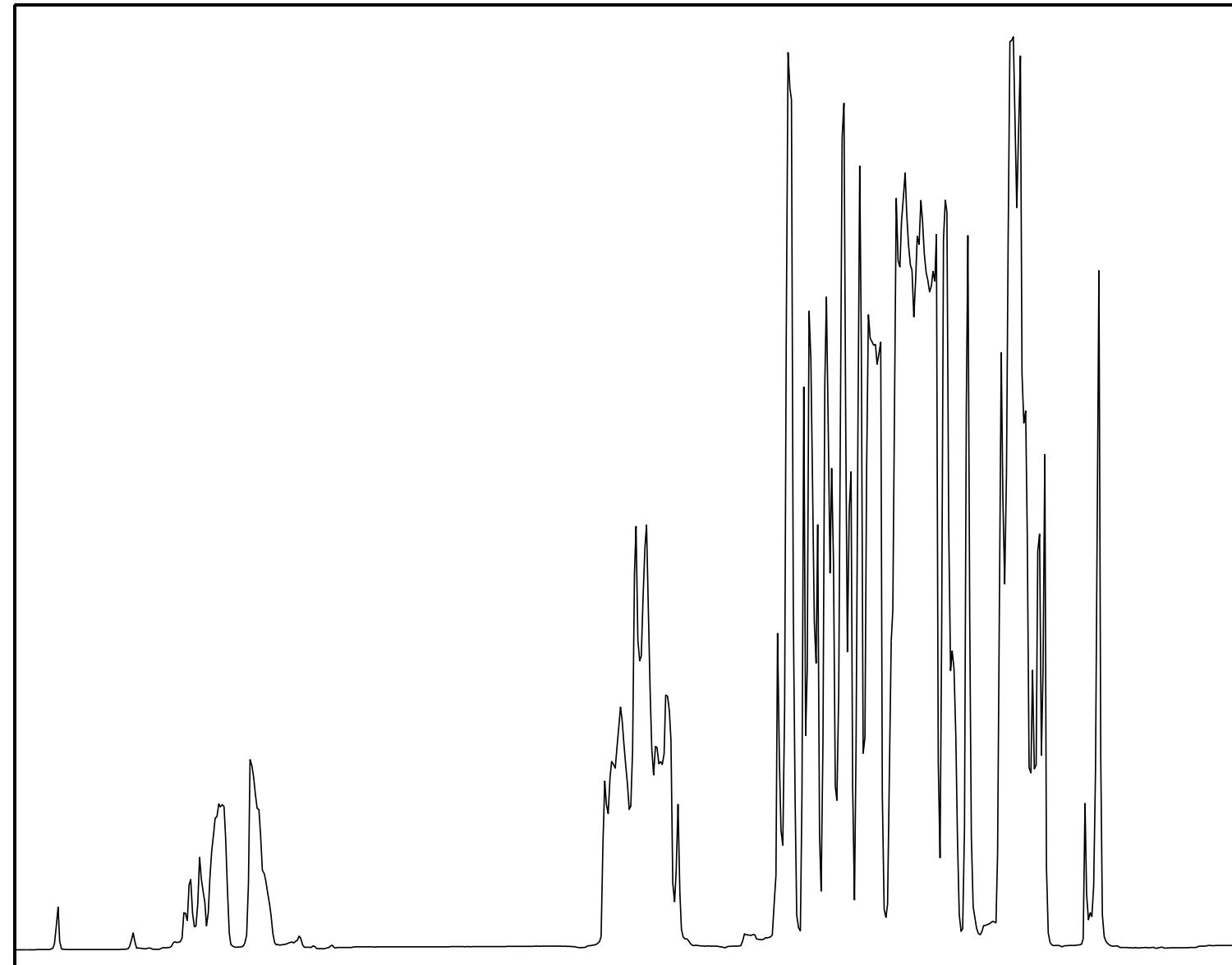


Idée 3 : utiliser les gradients

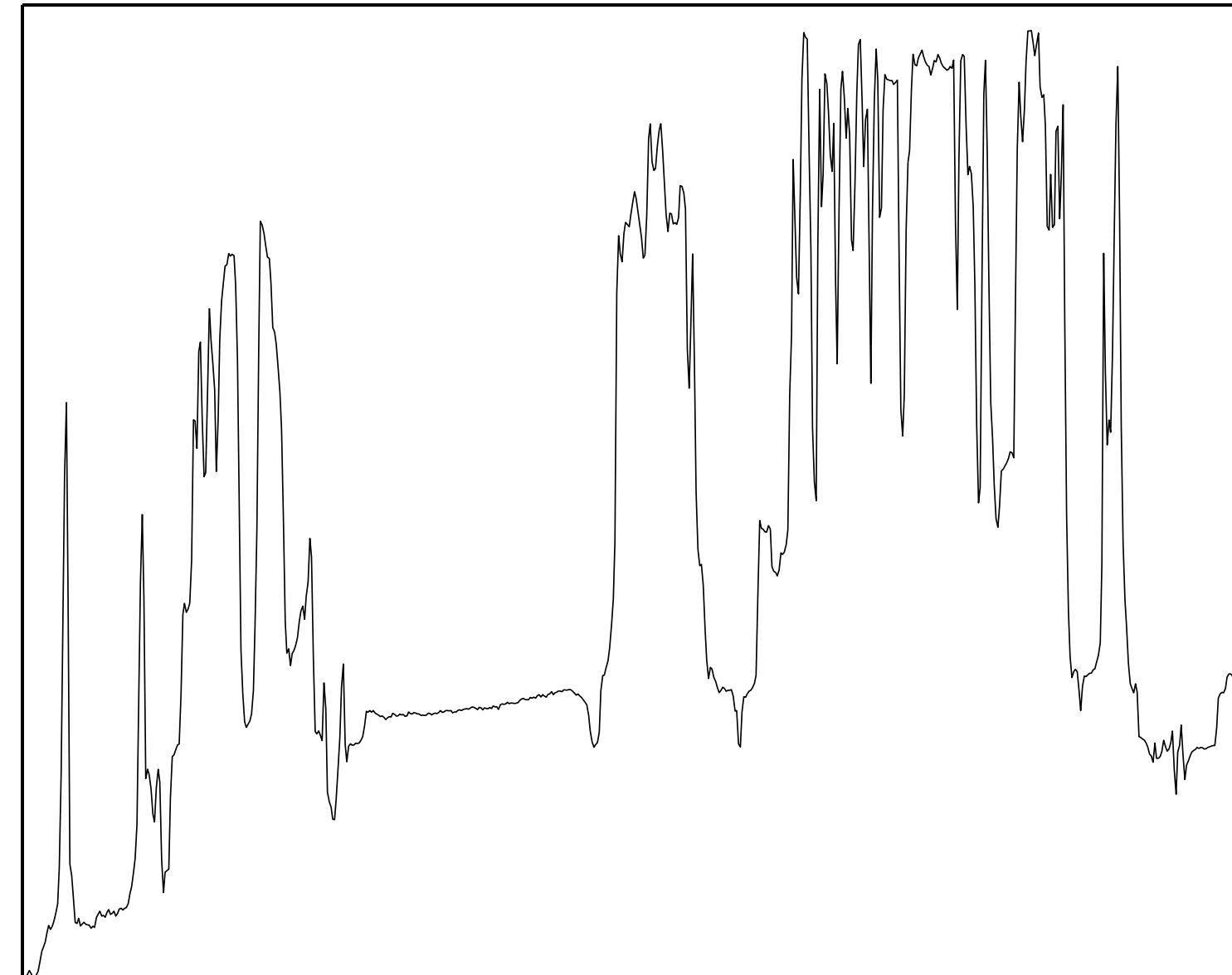


Signal HDR

Idée 3 : utiliser les gradients

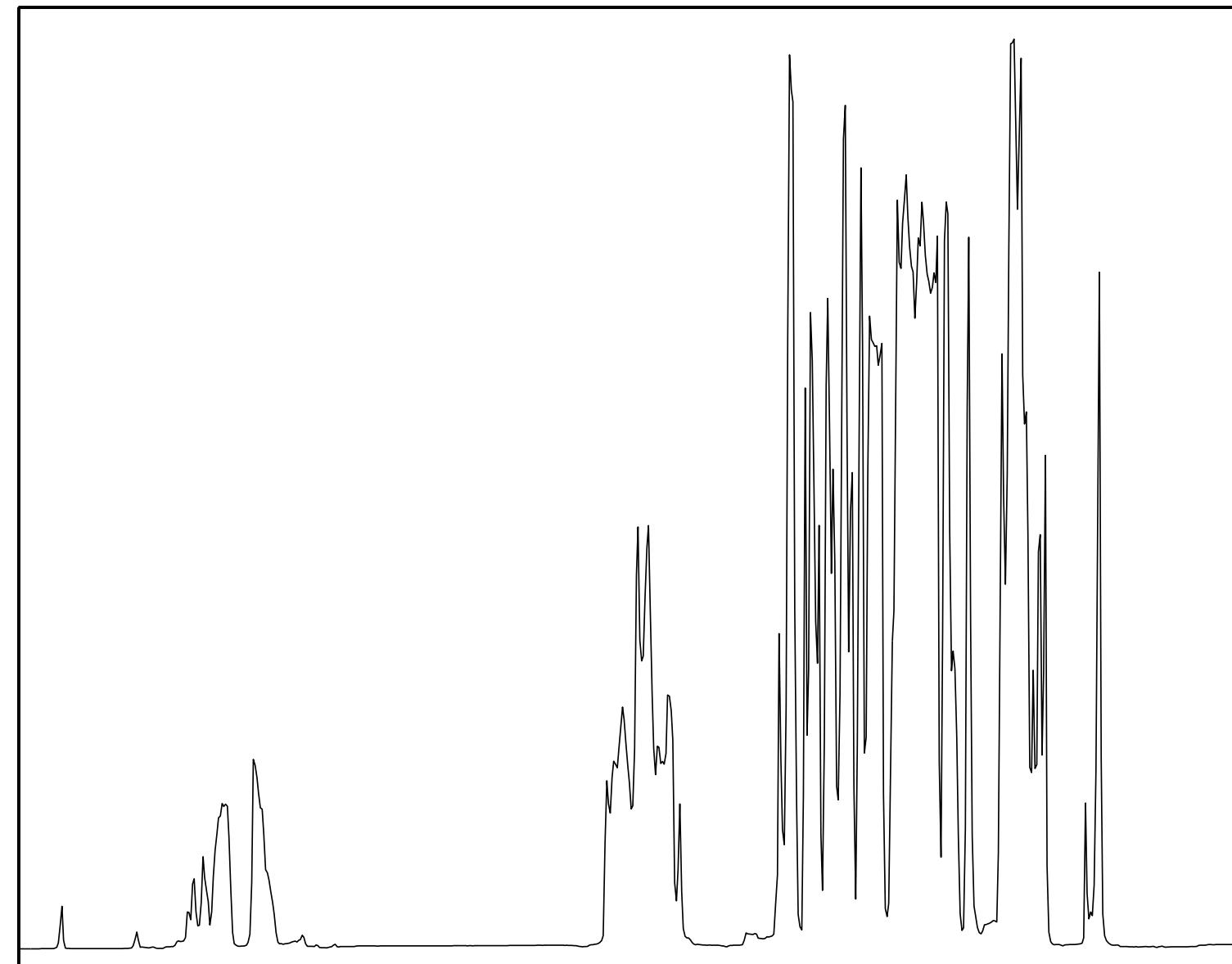


Signal HDR

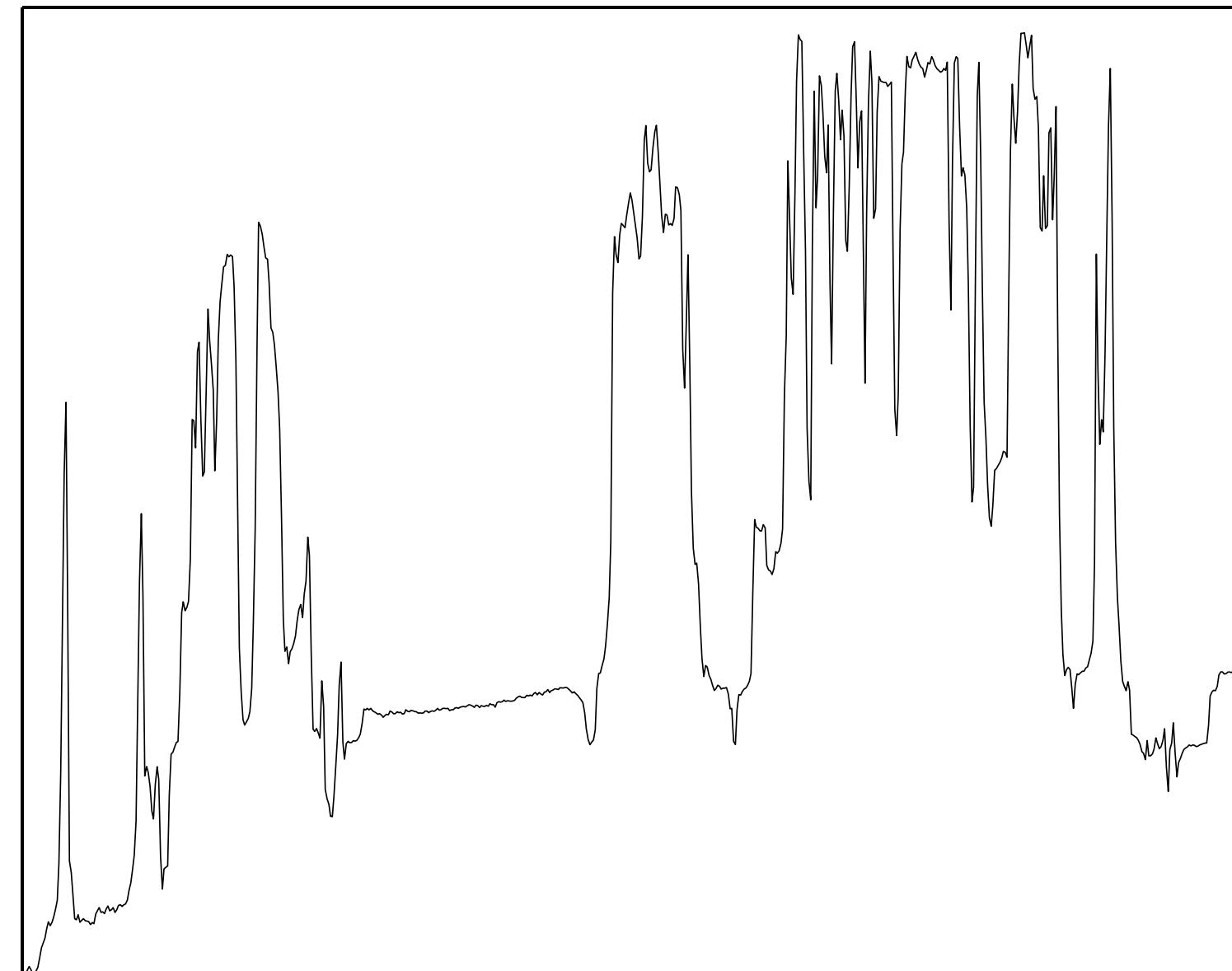


Log-signal HDR

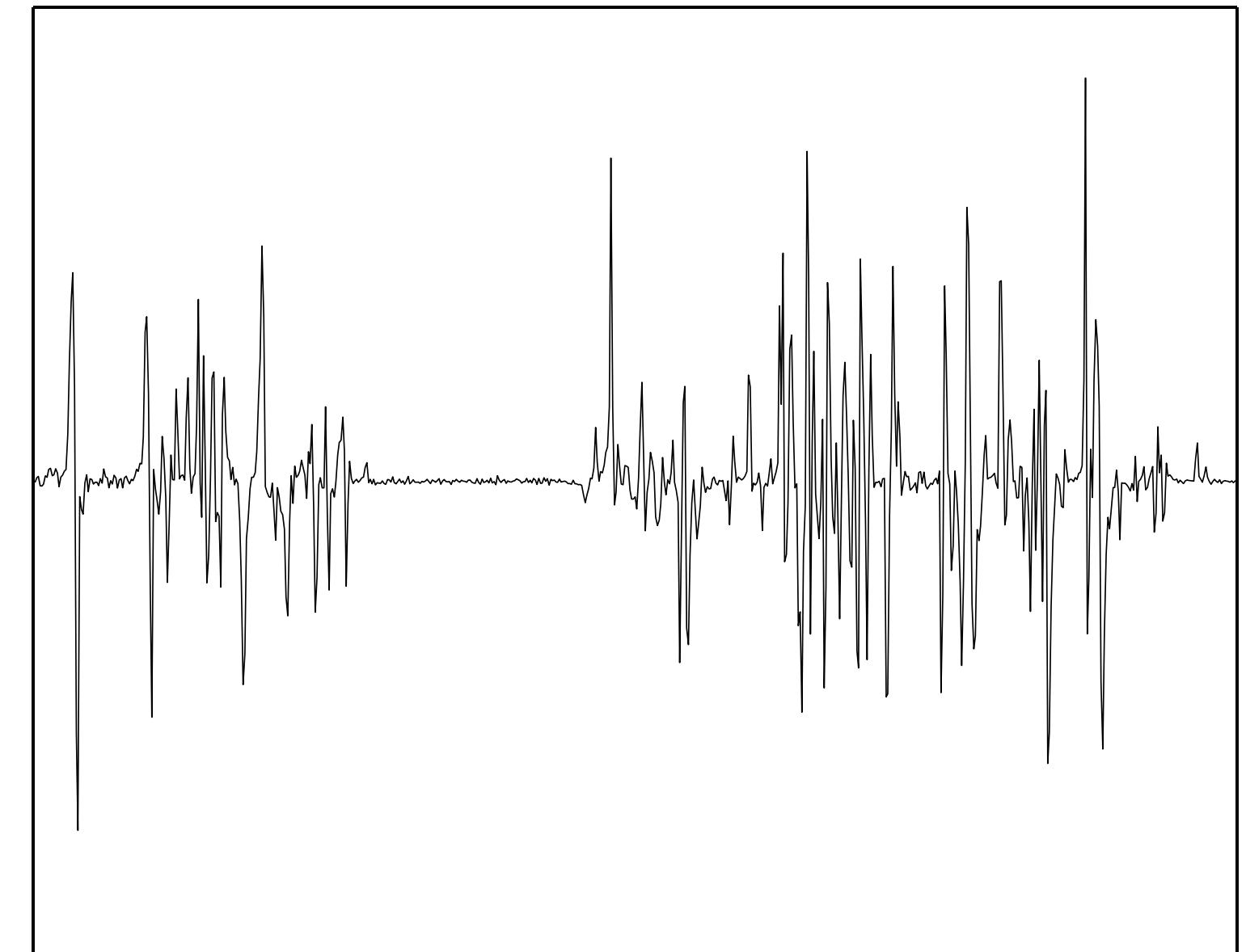
Idée 3 : utiliser les gradients



Signal HDR

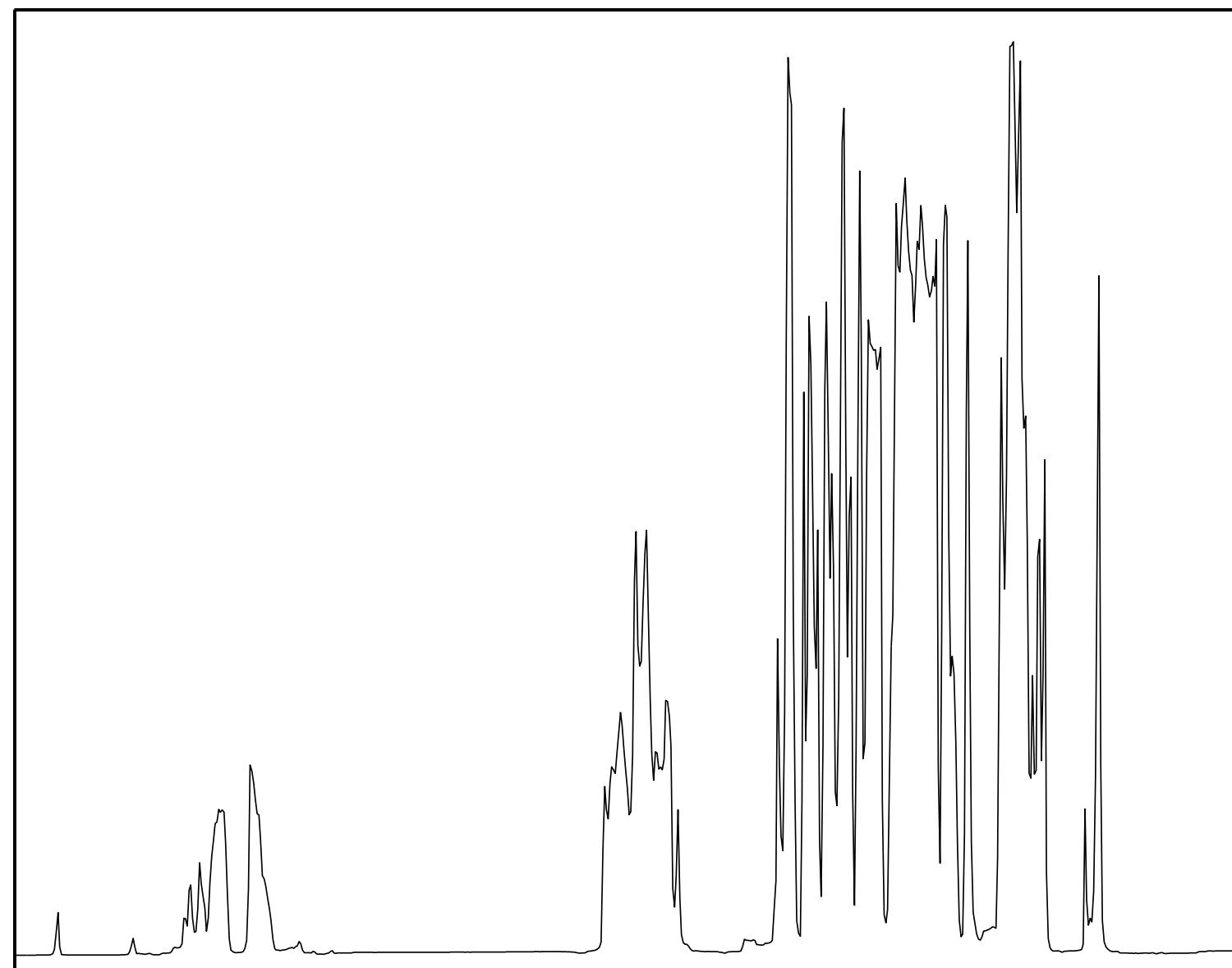


Log-signal HDR

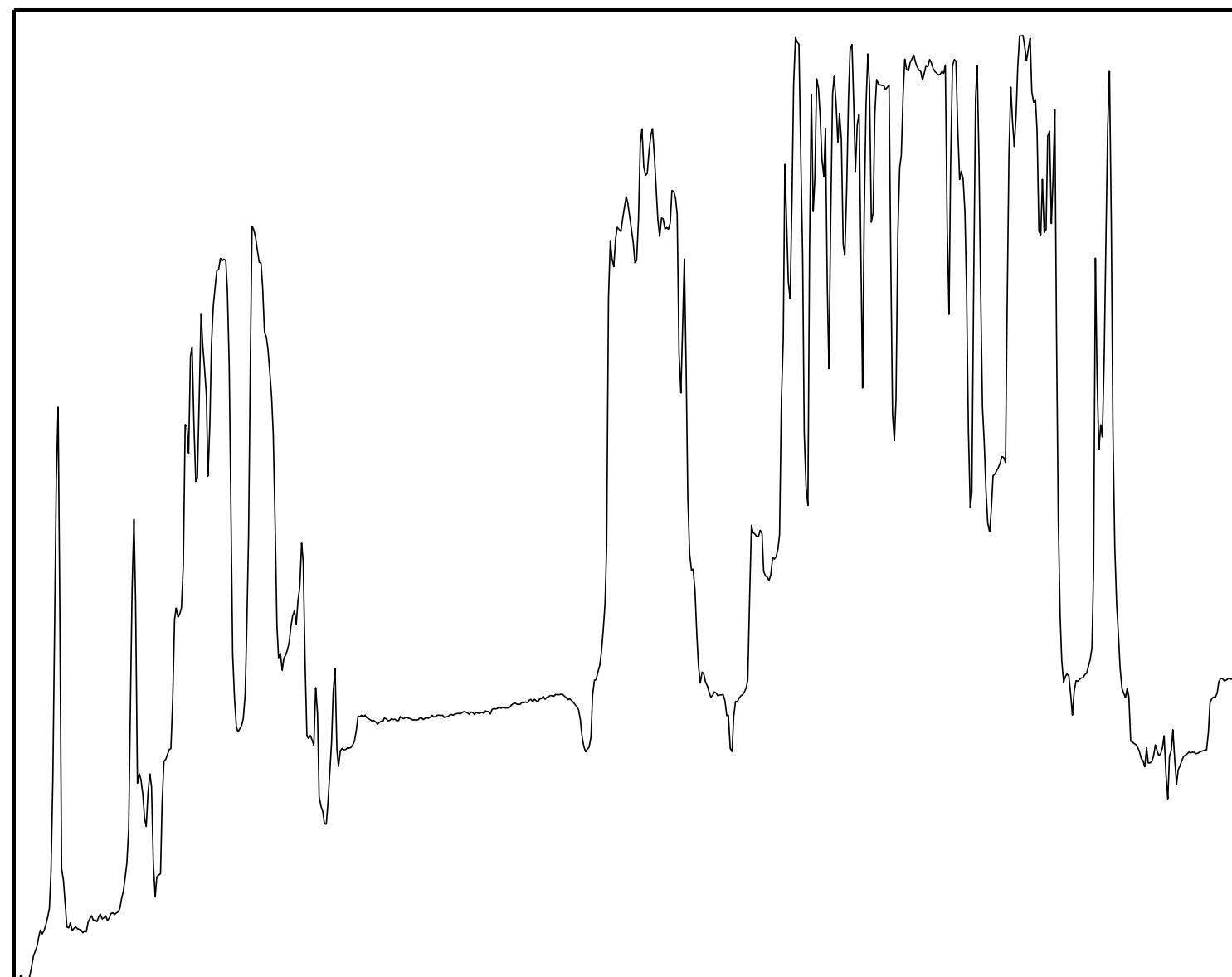


Dérivée du log-signal

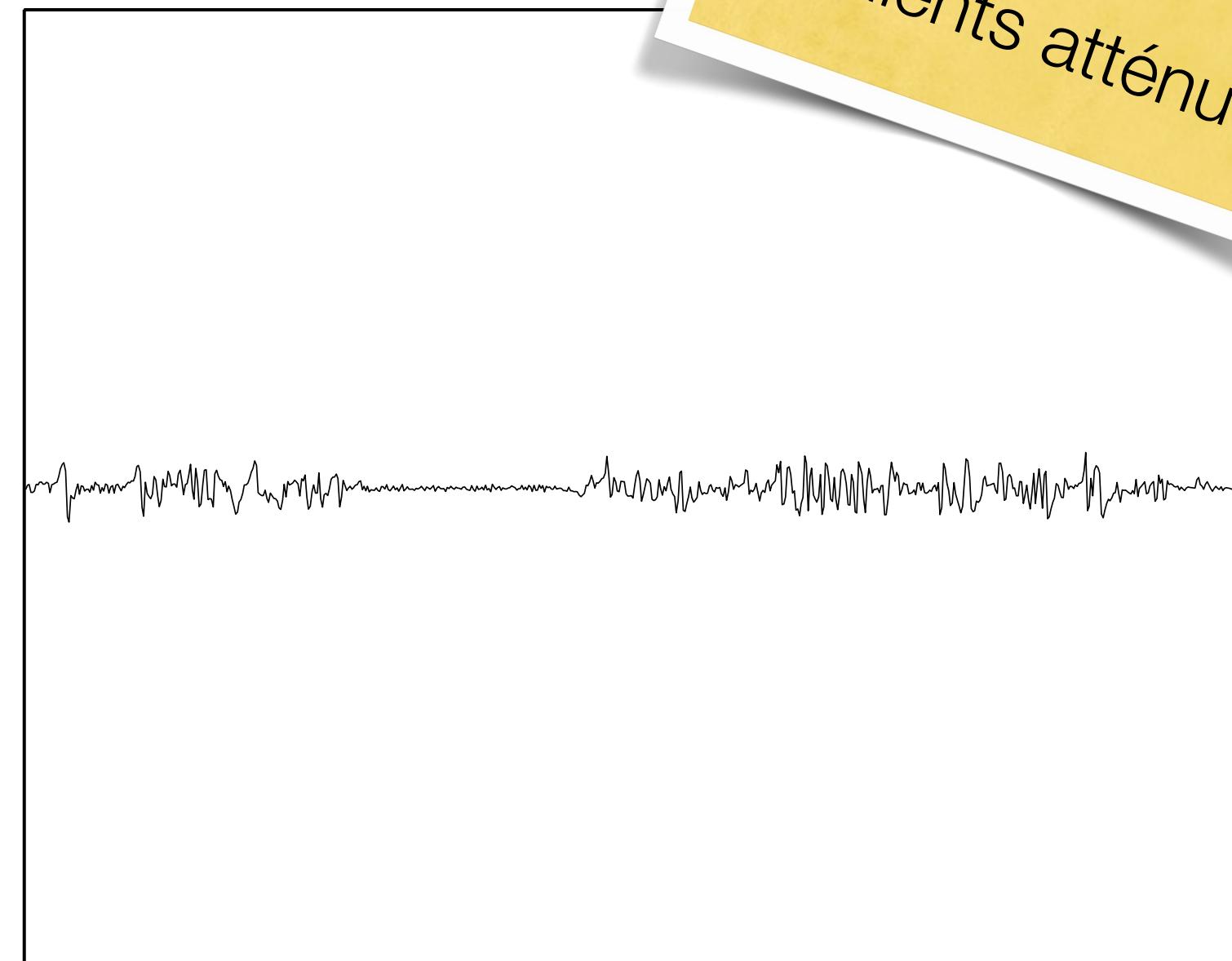
Idée 3 : utiliser les gradients



Signal HDR

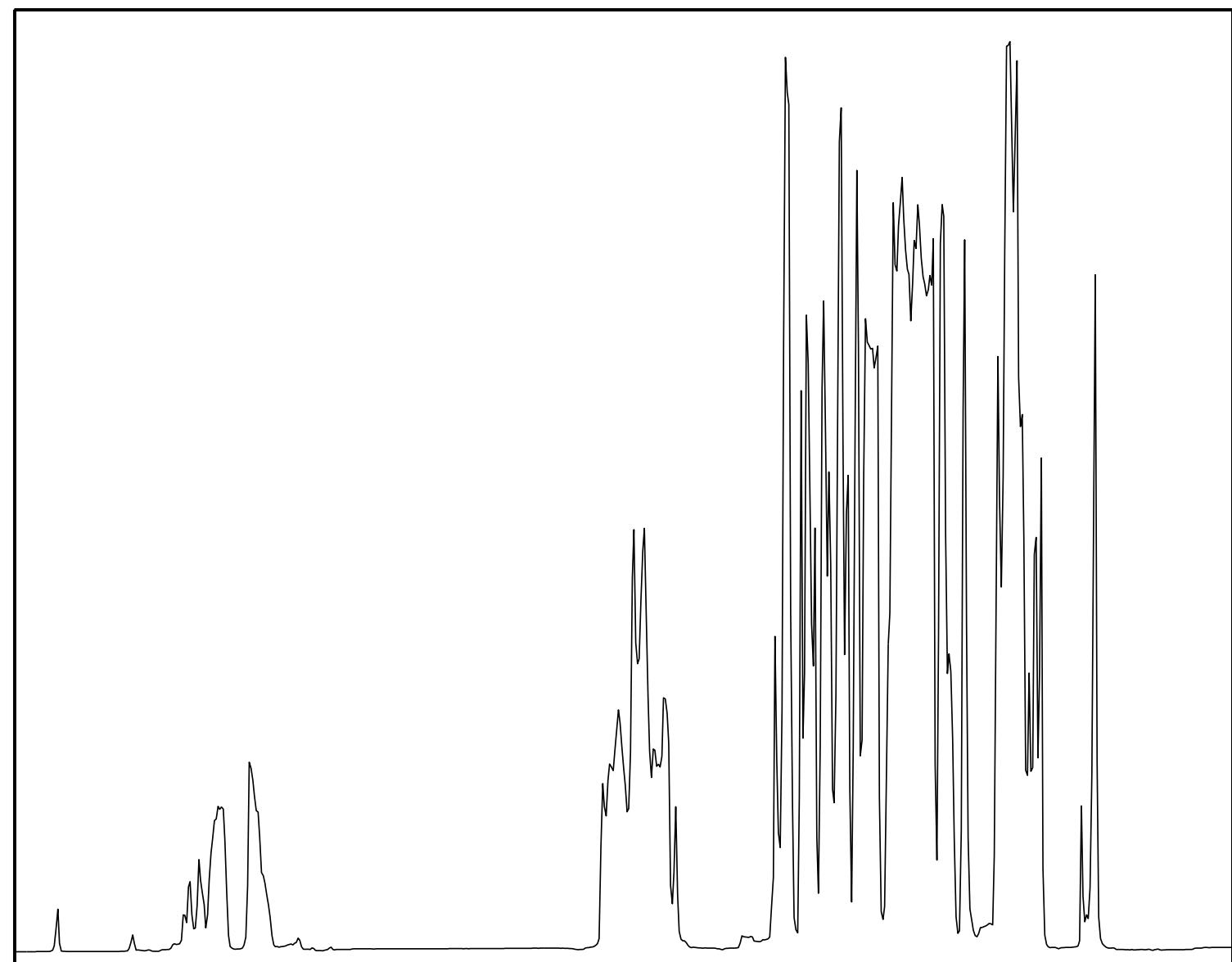


Log-signal HDR

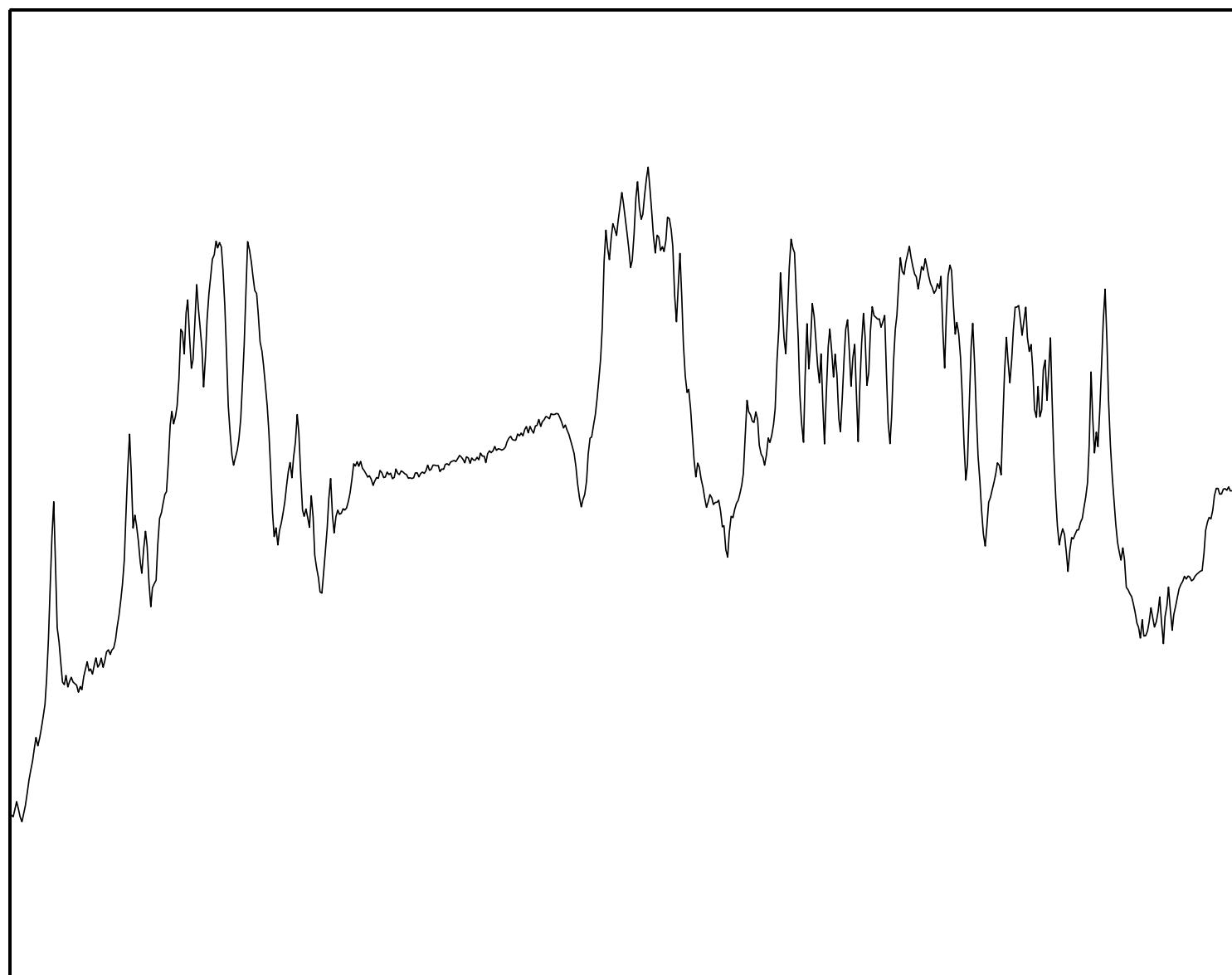


Dérivée du log-signal

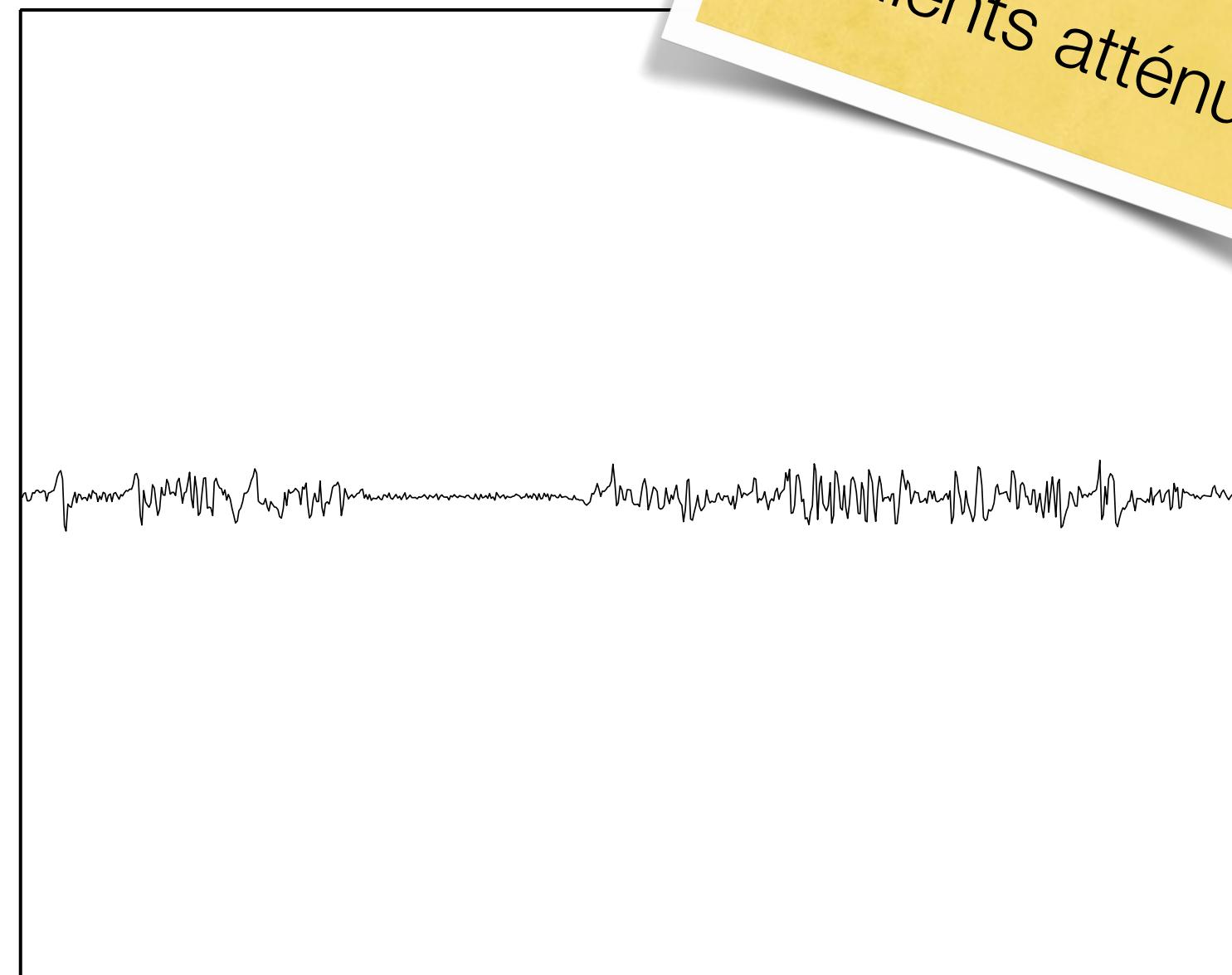
Idée 3 : utiliser les gradients



Signal HDR



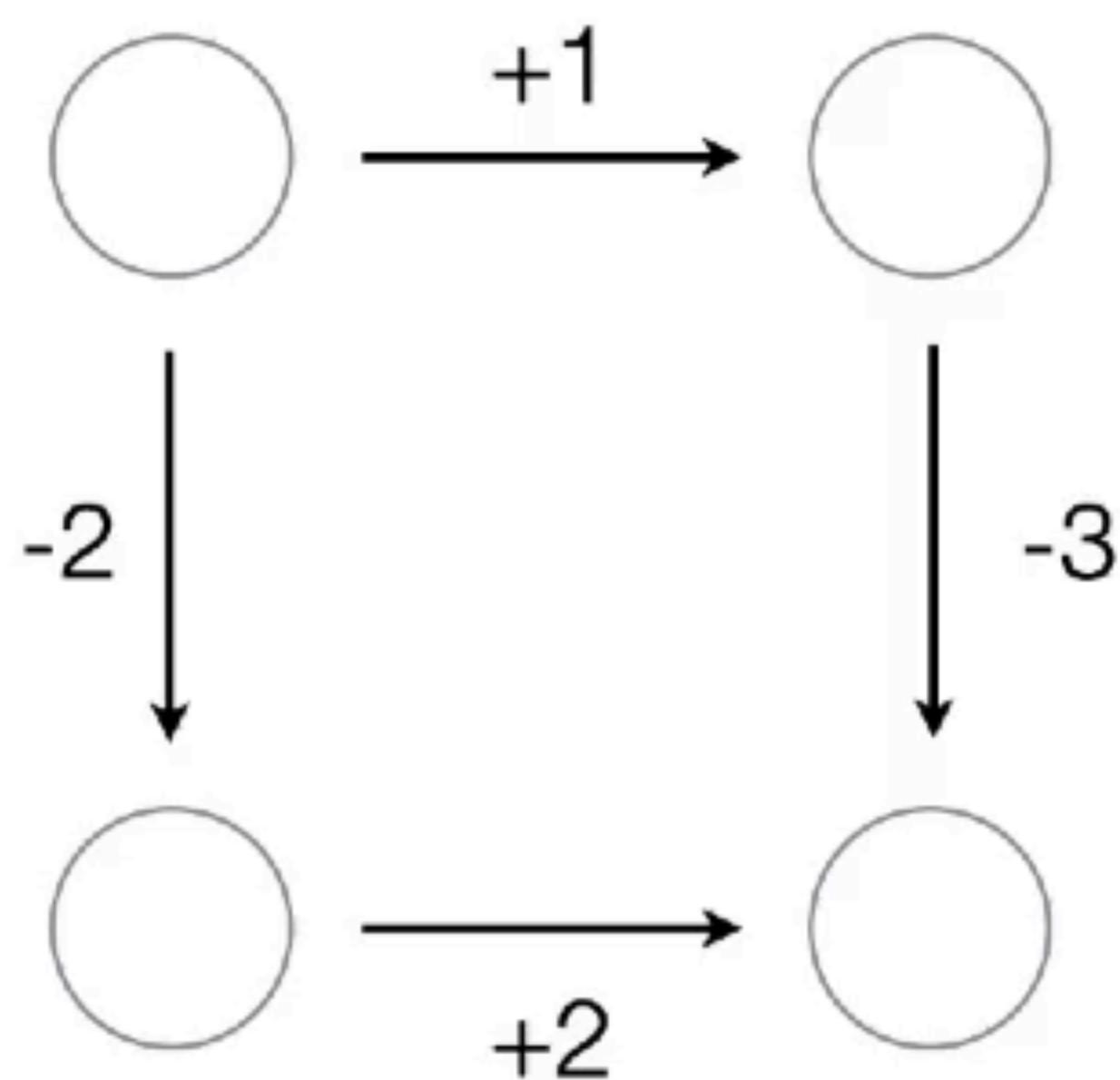
Signal reconstruit!



Dérivée du log-signal

Gradients atténués!

En 2D? Pas si facile...



Perceptuel (Reinhard et al.)



Filtre bilatéral (Durand et Dorsey)



Gradients (Fattal et al.)





Mais n'en mettez pas trop!



Luminance HDR

- Visualiser HDR
- Expérimenter avec opérateurs de reproduction tonale

The screenshot shows the official website for Luminance HDR (aka QtPfsgui) at qtpfsgui.sourceforge.net. The page features a banner image of a beach at sunset. The navigation bar includes links for Blog, About, Download, Documentation, Contacts, Internationalization, and Screenshots. Below the banner, there's a search bar and a "Donate" button with payment method icons. The main content area is titled "Luminance HDR 2.6.0" and discusses the release date (June 9, 2019), developer (Franco Comida), and the introduction of new tone-mapping operators. It also mentions improvements in speed and memory consumption, and adds post-processing options. A sidebar on the right lists "Archives" from June 2019 down to September 2010, and "Categories" including Development, News, Release, and Uncategorized. At the bottom, it mentions Flathub availability and a note about Flatpak.

Not Secure — qtpfsgui.sourceforge.net

Luminance HDR
aka QtPfsgui: a complete open source solution for HDR photography

Blog About Download Documentation Contacts Internationalization Screenshots

« Older Page 1 of 4 Type and Wait to Search

Luminance HDR 2.6.0
Published on June 9, 2019 by Franco Comida in Uncategorized. 0 Comments

Luminance HDR 2.6.0 is out!

After almost two years of development Luminance HDR 2.6.0 is finally out.

This new release introduces new features:

- Four brand new tone-mapping operators: ferwerda, kimkautz, lischinski and vanhateren.
- All tone-mapping operators have been optimized for speed and lower memory consumption (Thanks to Ingo Weyrich).
- Speed up for hdr creation (Also thanks to Ingo Weyrich).
- Added post processing gamma and saturation.
- In HDR Wizard it is now possible to preview the final HDR after applying different fusion settings and before accepting it.
- Other small improvements and bug-fixing as usual.

Minimum requirement for Luminance HDR is a **CPU that supports SSE2**.

You can download Luminance HDR 2.6.0 using one of these links:

[Windows 64 bits]
[macOS 10.12+ (64-bit, Intel)]
[macOS 10.10/11 (64-bit, Intel)]
[Source Code]

If you appreciate Luminance HDR, please consider a donation (there is no minimum amount!):

[Make A Donation](#)

If you want to share with other users your results, you can use our [Flickr group](#) and [Facebook group](#)

Luminance HDR is on Flathub
Published on November 21, 2018 by Franco Comida in Uncategorized. 0 Comments

Luminance HDR is now available on [Flathub](#)

Flathub aims to be the place to get and distribute apps for Linux. It is powered by [Flatpak](#) which allows Flathub apps to run on almost any Linux distribution.

Display a menu

Archives

June 2019
November 2018
September 2017
July 2017
May 2017
April 2017
February 2015
January 2014
April 2013
February 2013
July 2012
May 2012
April 2012
March 2012
January 2012
October 2011
August 2011
June 2011
April 2011
March 2011
January 2011
December 2010
October 2010
September 2010

Categories

Development (8)
News (9)
Release (13)
Uncategorized (6)

Meta

Log in
Entries RSS
Comments RSS
WordPress.org

Mode HDR?



Mode HDR?

