

BERTHET Vincent  
QUERO Benoît  
22/12/19



## Compte rendu : TP ES

### HDR



Enseignant : Olivier Le Meur

<b>Introduction</b>	<b>2</b>
<b>Création d'une image LDR à partir de différentes expositions</b>	<b>3</b>
Algorithme de Mertens	3
Algorithme de fusion reposant sur l'intensité des pixels et le gradient global	5
<b>Algorithmes de Tone-Mapping</b>	<b>6</b>
Algorithmes python cv2	6
Algorithmes de Tone-Mapping simples et globaux	7
<b>Conclusion</b>	<b>7</b>

## Introduction

Durant le module effets spéciaux, nous avons découvert différentes techniques liés aux images High Dynamic Range.

Ce TP sur le HDR nous a permis de manipuler l'API d'OpenCV via python pour générer des images HDR, LDR à partir de différentes expositions et à partir d'images HDR (Tone-Mapping).

# 1. Création d'une image LDR à partir de différentes expositions

Une image HDR permet d'obtenir une dynamique supérieure à 14 stops, cette image est codée en flottant. Une image LDR est quant à elle codée sur 24 bits par pixels ce qui par conséquence réduit la dynamique de l'image (moins de 10 stops). Cependant une image LDR est visualisable sur un écran classique contrairement à une HDR (non tonemapped)

La fusion d'exposition permet d'obtenir une image avec une plus grande gamme dynamique (dynamic range) en fusionnant des images représentant la même chose mais ayant une exposition différente. L'algorithme de Mertens permet d'obtenir une image à plus grande dynamique sans passer au préalable par une image HDR.

## a. Algorithme de Mertens

L'algorithme de Mertens fonctionne en fusionnant les images de différentes expositions en pondérant les pixels de celles-ci.

L'approche naïve consiste à calculer pour chaque image une carte de poids à partir du contraste, de la saturation et de la "bonne exposition" (favorisant les pixels avec une intensité moyenne) et de la multiplier avec l'image associée. Ce résultat n'est pas très bon puisque de nombreux artefacts apparaissent.



*Naive Fusion*

Mertens propose d'ajouter au calcul l'utilisation d'une pyramide laplacienne pour les images de différentes expositions et d'une pyramide gaussienne pour les cartes de poids. En multipliant chaque couche de la pyramide d'images avec chaque couche de la pyramide des poids, en additionnant les différents résultats, puis en reconstruisant la pyramide, on obtient un résultat réaliste ressemblant à une image HDR.



*Exposure time = 1/800*



*Exposure time = 1/125*



*Exposure time = 1/40*

*Images sources*



*Exposure Fusion (Mertens)*

## b. Algorithme de fusion reposant sur l'intensité des pixels et le gradient global

Cette méthode de fusion repose sur l'intensité des pixels et le gradient global. Le calcul des poids est différent de l'algorithme de Mertens. On calcule ici deux cartes de poids :

1 - au lieu de favoriser les pixels ayant une intensité moyenne (uniquement), le calcul de la "bonne exposition" prend aussi en compte l'intensité globale de l'image. Si certains pixels ont une intensité éloignée de la moyenne de l'image, ils sont favorisés. Cela permet de garder les pixels sombres des images à exposition longue (moyenne des pixels élevée) et les pixels clairs des images à exposition courte (moyenne des pixels basse).

2 - au lieu de ne prendre que les zones avec forts contrastes locaux, cette méthode donne plus de poids aux pixels où l'histogramme cumulé varie lentement. Autrement dit, la valeur de ces pixels sont relativement rares dans l'image, ce qui signifie que les pixels autour d'eux varient beaucoup.

La carte de poids finale est obtenue après multiplication (chaque carte peut être pondérée par une puissance) des deux cartes, puis filtrer.



*Fusion reposant sur l'intensité des pixels et le gradient global*

Pour notre exemple, cette algorithme semble moins efficace que celui de Mertens. Il y a énormément de contraste au niveau du ciel : il est bleu dans les zones homogènes et blanc autour des arbres et du bâtiment, ce qui n'est pas très réaliste. Le reste de l'image est plus clair mais semble moins contrasté.

## 2. Algorithmes de Tone-Mapping

### a. Algorithmes python cv2

- [Tonemap Drago](#) : Algorithme globale basé sur un adaptive logarithmic mapping
- [Tonemap Durand](#) : Algorithme basé sur un filtre bilatéral
- [Tonemap Mantiuk](#) : Algorithme basé sur des pyramide gaussienne
- [Tonemap Reinhard](#) : Algorithme globale

Les paramètres optimaux ont été définis expérimentalement. (image en plus grande sur le [GitHub](#))



Drago  
gamma=1.0 saturation=1.0 lum=500



Durand  
gamma=1.0 saturation=1.0 lum=700



Mantiuk  
gamma=1.0 saturation=1.0 lum=900



Reinhard  
gamma=1.0 lightAdaptation=0.75 lum=500

## b. Algorithmes de Tone-Mapping simples et globaux



Opérateur Gamma  
gamma=1



Opérateur Logarithmique



Opérateur exponentiel

## Conclusion

Ce TP nous a permis de découvrir différentes fonctions python et OpenCV concernant le HDR. Nous avons testé et comparé plusieurs méthodes de fusion d'expositions comme l'algorithme de Mertens. Ces algorithmes permettent de créer une image LDR ressemblant à du HDR sans passer par du Tone-Mapping. Nous avons ensuite pris en main différents algorithmes de Tone-Mapping permettant donc cette fois de générer une image LDR à partir d'une image HDR.

Source : <https://github.com/RealVincentBerthet/HDR>

