



Master STS Informatique ISIDIS  
Ingénierie des Systèmes Informatiques Distribués

## Projet de synthèse

---

# Etude et développement d'une webcam HDR avec correction de couleurs

---

*Sous la direction de :*

M. Fabien TEYTAUD Maître de conférence  
Directeur d'études Master 1 ISIDIS  
M. Julien DEHOS Maître de conférence  
M. Pierre-Alexandre HEBERT Maître de conférence

*Auteur :*  
Eric SAILLY  
Master 1 ISIDIS

Version 1.0 du  
4 juin 2017



# Table des matières

<b>Remerciements</b>	<b>2</b>
<b>Résumé</b>	<b>3</b>
<b>1 Introduction / Contexte</b>	<b>4</b>
1.1 Présentation du Projet . . . . .	4
1.2 HDR définition . . . . .	4
1.3 Problématique . . . . .	5
1.4 Besoins, Spécifications des demandes et environnement de développement . . . . .	6
1.5 Etat du produit . . . . .	6
<b>2 Méthodes de génération HDR</b>	<b>7</b>
2.1 Principes . . . . .	7
2.1.1 Calibrage . . . . .	7
2.1.2 Fusion / Reconstruction . . . . .	9
2.1.3 Tone Mapping . . . . .	9
<b>3 Développements</b>	<b>12</b>
3.1 Environnement . . . . .	12
3.2 Adaptations obligatoires pour passer en vidéo . . . . .	12
3.2.1 Problématique du flux vidéo . . . . .	12
3.3 Produit développé . . . . .	13
3.3.1 Génération des images d'exposition différentes . . . . .	13
3.3.2 Génération des statistiques avec le programme de test . . . . .	14
3.3.3 Développement du Programme de vidéo HDR . . . . .	14
3.3.4 Ajout du correction automatique de luminosité et adaptation des temps d'exposition à l'environnement . . . . .	15
3.4 principales difficultés rencontrées . . . . .	15
3.4.1 Difficulté de synchronisation des images et du temps d'exposition . . . . .	15
3.4.2 Matériel utilisé . . . . .	16

<b>Table des matières</b>	<b>iii</b>
<b>4 Résultats</b>	<b>17</b>
4.1 Calibrages . . . . .	17
4.2 Temps de traitement d'une image et nombre d'images par seconde . . . . .	17
4.3 résultats visuels . . . . .	17
<b>Conclusion</b>	<b>26</b>

# Table des figures

1.1	Oeuvre surréaliste . . . . .	5
2.1	Old Saint Paul 6 prises de vue identiques . . . . .	7
2.2	Old Saint Paul HDR reconstituée . . . . .	8
2.3	Images pour Calibrage . . . . .	9
2.4	Calibrage Debevec . . . . .	10
2.5	Calibrage Robertson . . . . .	10
3.1	communication des flux . . . . .	16
4.1	Temps moyen de reconstruction . . . . .	18
4.2	Temps moyen de mappage . . . . .	18
4.3	Temps moyen de traitement . . . . .	19
4.4	Images flux vidéo Bright Debevec . . . . .	20
4.5	Images flux vidéo Dark Debevec . . . . .	21
4.6	Images flux vidéo Bright Robertson . . . . .	22
4.7	Images flux vidéo Dark Robertson . . . . .	23
4.8	Images flux vidéo Bright Mertens . . . . .	24
4.9	Images flux vidéo Dark Mertens . . . . .	25

# Liste des sigles et acronymes

**HDR** *High Hynamic Range ou Imagerie à grande gamme dynamique*

**bpp** *Bits par pixel*

**fps** frame per second ou images par seconde



# Remerciements

Je remercie mes encadrants, Messieurs Julien Dehos, Fabien Teytaud, Pierre-Alexandre Herbert, pour leur implication dans le projet et leur assistance, Sébastien Adamiak professeur vacataire d'anglais ULCO pour la relecture.

Je remercie également mes camarades de promo pour l'intégration au sein du groupe et les aides diverses apportées tout au long de cette année et particulièrement sur le projet.

# Résumé

Ce document traite de la vidéo HDR, particulièrement du développement d'un logiciel de vidéo en HDR. C'est une première étape du développement qui est de choisir la meilleure méthode de calcul pour un rendu fluide et une image de bonne qualité répondant à la dynamique de l'HDR. Il explore trois techniques de calcul et reconstruction : Debevec, Robertson et Mertens.

# Chapitre 1

## Introduction / Contexte

### 1.1 Présentation du Projet

Une image est une mesure de la lumière au niveau d'un capteur (œil, camera). Les capteurs de caméra ont une dynamique (plage d'énergie lumineuse traitée par le capteur) moins large que l'œil humain. Ce qui explique une différence de rendu final : des zones très sombres ou surexposées apparaissent, alors que l'œil les aurait traitées.

L'objectif principal est d'étudier l'influence de la reconstruction HDR pour implémenter un logiciel de webcam l'utilisant. La première étape de ce projet est de réaliser la capture des images vidéos affin d'effectuer le traitement.

### 1.2 HDR définition

Dans la nature, l'œil humain est capable de distinguer d'infimes nuances d'ombres et de lumières. Cette capacité à discerner l'écart entre les zones claires et foncées est nommée "plage dynamique". Infiniment plus performants que les capteurs présents dans nos appareils numériques, nos yeux transmettent des informations pas toujours exploitables lors de la capture et de la restitution de l'image. Alors, pour pallier à ce manque, les photographes ont trouvé la solution : la photo HDR (High Dynamic Range ou Image à grande gamme dynamique). Cette technique permet ainsi de surpasser et d'optimiser au maximum les capacités de l'appareil photo, en mettant en avant cette grande plage dynamique lumineuse perçue par l'homme.

Donc, la véritable HDR ne peut pas être obtenu avec un seul cliché. Un travail après traitement doit avoir lieu, pour redonner toute l'intensité et la luminosité à la photographie. Cette technique valorisera au maximum vos clichés en faisant apparaître chaque détail et nuance dissimulés dans une unique photo. Grâce à ce type de photographie, on se rapproche au plus près de la vision humaine et des nuances vues par l'œil.

Avec le HDR, le rendu peut être réaliste (comme vous la percevez, naturel), ou impressionniste, (c'est à dire presque surréaliste, semblable à une peinture ou un dessin avec beaucoup de texture)<sup>1.1</sup>. Mais dans les deux cas, le résultat reste tout simplement bluffant et la qualité de l'image incomparable.



FIGURE 1.1 – oeuvre surrealisme d'Igor Morski.  
<http://www.wikilinks.fr/wp-content/uploads/2012/07/oeuvre-surrealisme-Igor-Morski-3.jpg>

### 1.3 Problématique

Le traitement d'images par ordinateur impose des limites. Pour la capture, comme pour la restitution, les matériels standard ne sont capables de gérer une plage aussi étendue que l'œil humain. Sauf exception, les pixels sont traités en 8 bits par pixels (bpp) soit 256 niveaux de luminosité dans le cas d'une image en niveau de gris. L'œil est capable d'en distinguer beaucoup plus. Les 256 nuances de gris affichées ne lui suffisent pas pour une bonne dynamique.

Dans le cas d'une image en couleur, cette limitation est aussi présente, car le traitement se fait sur les trois couleurs primaires (que sont le Rouge, le Vert, et le Bleu) codées chacun sur 256 niveaux de luminosité soit  $256^3 = 16.7$  Millions.

Avec seulement 256 niveaux de luminosité, il y a des cas où des sources lumineuses apparaissent surexposées ou saturées, tandis que les scènes sombres sont simplement capturées en noir.

Il existe actuellement des capteurs qui capturent plus de 8 bpp<sup>1</sup>. Mais, la méthode la plus courante pour créer des images à haute dynamique est d'utiliser une caméra 8 bpp et de prendre des images avec différentes valeurs d'exposition [DM].

Cette méthode a plusieurs inconvénients :

- Imposse un trépied (le capteur doit être statique).
- Utilisation d'au moins trois images d'exposition faible, moyenne et haute garantissant la dynamique.

C'est un problème pour la vidéo, le flux continu doit afficher 25 à 30 images par seconde pour être fluide. Utiliser plusieurs captures, avec des temps de pose important, va diminuer le nombre d'images par seconde en dessous de 5 images par seconde voir 1.

Pour le développement, il faudra trouver une solution pour fournir un minimum d'image par

1. Bit per pixel ou bit par point

seconde ou l'anglicisme fps<sup>2</sup>

## 1.4 Besoins, Spécifications des demandes et environnement de développement

Développement à réaliser en C++ sous linux en utilisant les bibliothèques :

- OpenCV pour le traitement d'images.
- LibUVC pour la gestion des périphériques USB vidéo.

## 1.5 Etat du produit

Pour démarrer, une première version de capture vidéo fonctionnant, avec 2 images alternativement sur-exposées et sous-exposées, est fournie par M. Julien Dehos sur son compte GitHub <https://github.com/juliendehos/hdrcam>.

---

2. Frame per second ou images par seconde

# Chapitre 2

## Méthodes de génération HDR

### 2.1 Principles

Le principe le plus courant est de prendre plusieurs photos avec des temps de pose différent (au minimum 3) et d'effectuer des traitement sur l'ensemble des images. Un exemple de la chapelle Old Saint Paul et sa représentation HDR par Dean S. Pemberton. 2.1 2.2

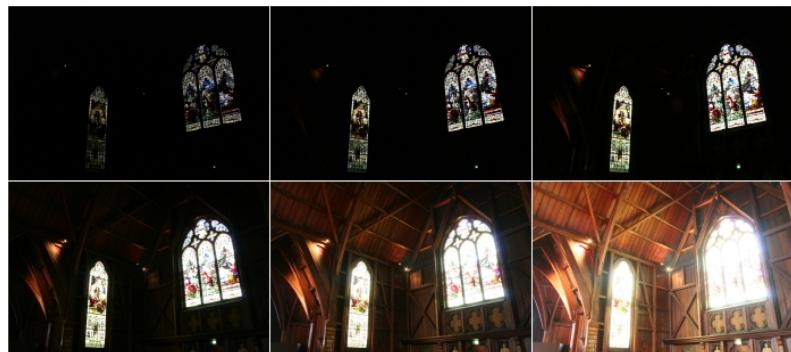


FIGURE 2.1 – Old Saint Paul 6 prises de vue identiques avec des temps d'exposition : 1/40" 1/10"  
1/2" 1" 6" 25".

Elles servent de base pour le Calibrage de la caméra et la reconstruction de l'image HDR 2.2

Les étapes à suivre sont :

- calibrage de la caméra
- la fusion des images
- la correction de la tonalité

#### 2.1.1 Calibrage

La représentation informatique d'une image est nécessairement discrète, alors que l'image est de nature continue. Certains capteurs effectuent une discréttisation : c'est le cas des appareils photos numériques. Par contre, les appareils photos argentiques n'effectuent pas cette quantification.

Pour un ordinateur, une image est un ensemble de pixels. Un pixel, c'est un élément de l'image. Un pixel possède une valeur qui peut être un scalaire représentant un niveau de gris, un vecteur représentant une couleur ou tout autre chose.



FIGURE 2.2 – version HDR reconstituée à partir des images 2.1

Les images dites en “noir et blanc” sont composées de pixels noirs ou blancs (deux valeurs possibles). Les images en niveaux de gris sont composées de pixels de valeurs scalaires représentant la luminosité. En général, les valeurs sont entières entre 0 (le noir) et 255 (le blanc).

Les images couleurs sont composées de pixels dont les valeurs sont soit scalaires (on parle alors de couleurs indexées car la valeur représente un numéro de couleur) soit vectorielles (les couleurs sont représentées selon trois composantes voire plus).

Pour augmenter artificiellement la dynamique, on utilise plusieurs images. Il devient utile de générer une table de correspondance des couleurs/luminosités par valeur de pixel et les temps d’exposition.

Il existe plusieurs méthodes reconnues :Debevec et Robertson. Elles créent une table de réponses d’intensité par rapport aux valeurs de pixels en entrée.

### Calibrage Debevec

La méthode de calcul version Debevec utilise uniquement quelques points pris au hasard dans la matrice ou fixés par l’utilisateur et fournit une courbe linéaire de réponse de luminosité comme l’indique la figure 2.4.

Comme expliqué dans l’article [DM], l’algorithme prend quelques pixels dans l’image. Il calcule ensuite la réponse correspondante de ces pixels. Qu’il transforme en une fonction discrète établie sur l’intervalle [0-255] pour respecter les 8 bpp. Vu les résultats obtenus par cette méthode, Debevec et Malik ont décidé de ne pas traiter tous les points mais de générer une fonction continue tout en pondérant les extrêmes.

Selon les auteurs, cette méthode donne de bons résultats. Surtout s’il existe des zones très ou sous-exposées.

### Calibrage Robertson

La méthode de calcul version Robertson utilise tous les points de la matrice pour calculer le vecteur des réponses, ce qui donne alors une courbe avec des variations importantes, non monotone,



FIGURE 2.3 – Jeux d’images prises pour calibrage. Exposition 1/1000" 2/1000" 1/100" 1/30" 1/20" 1/10". Cliché personnel

comme l’indique la figure 2.5.

Le traitement de Robertson est similaire avecDebevec. La grosse différence est que le traitement est effectué sur l’intégralité des pixels de l’image. Le résultat n’est dans ce cas plus continu.

### 2.1.2 Fusion / Reconstruction

Après le Calibrage de la réponse caméra, qui peut être fait une fois par lancement du logiciel, il faut effectuer une fusion des différentes prises de vues. Cette étape va ajouter une plus large dynamique de l’intensité lumineuse, ce qui va avoir pour conséquence directe de changer le poids de chaque pixels qui dépassent les 8 bpp.

Pour la reconstruction des images, il existe aussi trois méthodes Debevec, Robertson et Mertens. Elles calculent la valeur de chaque pixel de l’image reconstituée, en se basant sur la fonction de Calibrage vue précédemment.

### 2.1.3 Tone Mapping

La reconstruction de l’image génère des valeurs de luminosité trop grandes pour être traitées par les moniteurs vidéo. De plus, les caméras et les moniteurs ne travaillent pas avec les mêmes valeurs. Il convient alors de faire une correction de tonalité.

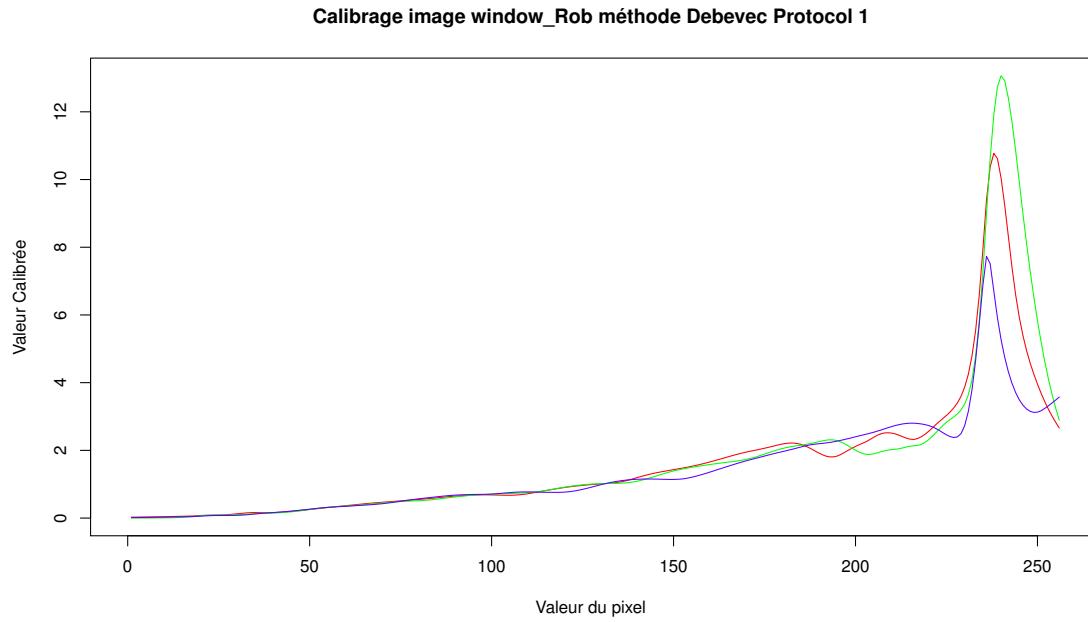


FIGURE 2.4 – Courbe de réponses suite au calibrage avec la méthode Debevec et 6 images d'exposition 1/1000" 1/30" 1/500" 1/20" 1/100" 1/10" on remarque la monotonie de la courbe avec une distorsion important dans les extrêmes. Jeux d'images utilisées pour le calibrage figure2.3

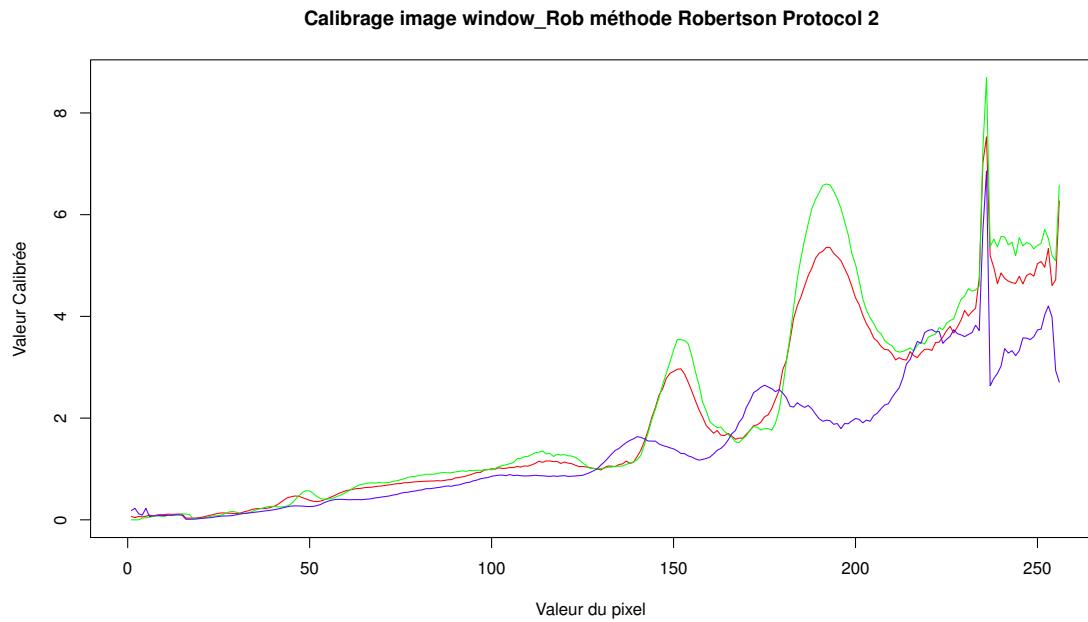


FIGURE 2.5 – Courbe de réponses suite au calibrage avec la méthode Robertson et 6 images d'exposition 1/1000" 1/30" 1/500" 1/20" 1/100" 1/10".  
On remarque les variations importantes et comme avec Debevec distorsion importante dans les extrêmes.Jeux d'images utilisées pour le calibrage figure2.3

Les paramètres de correction sont la valeur du gamma, de la saturation et du contraste. Le plus important étant la gestion du gamma, car elle corrige fortement l'éclairage de l'image.

Seule la méthode de reconstruction de Mertens, qui utilise le gamma, ne nécessite pas de refaire une modification de la tonalité.

# Chapitre 3

## Développements

### 3.1 Environnement

Le logiciel est développé pour Linux en C++ 14.

Les périphériques d'entrée utilisés pour le développement et les test sont :

- 1 Hercules HD Twist P/N 5090310
- 2 Logitech Webcam Pro 9000 for Business P/N960-00562

#### OpenCV3

La librairie utilisée pour les traitements d'images est OpenCV 3.2.0. Elle est téléchargeable sur <https://github.com/opencv/opencv/archive/3.2.0.zip> et la documentation sur <http://docs.opencv.org/3.2.0/>

#### Libuv

Pour la librairie Libuv, elle est téléchargeable sur <https://github.com/ktossell/libuv>. La documentation est incluse dans le code.

Dans notre cas, la librairie est intégrée au projet hdrcam développé par M. Julien Dehos qui sert de départ au projet.

### 3.2 Adaptations obligatoires pour passer en vidéo

La principale modification des méthodes de développement pour la vidéos HDR est qu'il faut plusieurs images avec des temps d'exposition différents. Dans la pratique, quand le flux vidéos est lancé, il fournit des images avec un temps d'exposition fixe et continu.

Le passage des paramètres aux périphériques n'est pas le même pour tous les constructeurs et il est très peu documenté.

#### 3.2.1 Problématique du flux vidéo

Le changement du temps d'exposition est possible, mais occasionne des temps de latence entre le paramétrage du périphérique et la réponse réelle de celui-ci.

### 3.3 Produit développé

Le développement s'est passé en plusieurs phases :

- 1 Un programme capable de générer plusieurs images avec des temps d'exposition différents HdrCAm\_Vidéo.
- 2 Un programme test pour générer des statistiques HdrCam\_Test.
- 3 Modification du programme principal une fonction pour effectuer le Calibrage de la caméra et une boucle dans le main qui fournit un flux HDR de base.
- 4 Ajout d'une méthode de correction de luminosité automatique.

#### 3.3.1 Génération des images d'exposition différentes

La prise en main du logiciel HDRcam générant 2 images (une claire et une sombre) est passée par les étapes :

- 1 Installation de la bibliothèque openCV3.2. suivie de Libuv.
- 2 Test de lancement avec la caméra intégrée au portable. Aucun résultat, erreur d'accès au périphérique. Après discussion avec M. Dehos, le problème est double. La caméra du portable n'est pas compatible avec le logiciel et il doit être lancé en mode administrateur car il doit accéder aux fichiers système.
- 3 mise en place d'un vecteur de matrice et d'un index pour passer de 2 à 3 images.

Début des dysfonctionnements, le programme doit fournir 3 images placées dans des conteneurs spécifiques un clair, un moyen et un sombre. Mais elles arrivent totalement désynchronisées. Avec aucune garantie que l'image traitée soit la bonne.

La capture pour le calibrage n'est pas identique à celle pour le flux vidéo. En effet, le calibrage, pour être le plus performant possible, lors du calcul de l'image HDR, nécessite un nombre d'images important et il ne doit être fait qu'une seule fois. Le développement de deux modes différents était évident. Il y a donc une fonction de capture standard 1 associée à la boucle principale 2 pour le flux vidéo et une fonction de capture calibrage 3 associée à la boucle de calibrage 4 pour le calibrage.

---

#### Algorithm 1 Fonction de capture standard

---

```

if Capture ok then
    if nbFrame > 0 then
        Bloque les process
        copy Frame => images[index]
        Capture = ko
        nbFrame = 0
    else
        incrémente NbFrames
    end if
end if

```

---

---

**Algorithm 2** Boucle dans le programme principal pour acquisition vidéo

---

```

index = 0 | CompteurCapture = nbImages
while critère d'arrêt non renseigné do
    Bloque les process
    if index != CompteurCapture then
        change le temps d'exposition par exposition[index]
        Wait pendant exposition[index] + Temps de latence
        capture = OK
        CompteurCapture = index
    end if
end while

```

---



---

**Algorithm 3** Fonction de capture Pour calibrage

---

```

T = TempsEcoulé
if Capture ok et T > temps minimum et T < temps maximum then
    Bloque les process
    copy Frame => images[index]
    Capture = ko
end if
restart TempsEcoulé

```

---



---

**Algorithm 4** Boucle dans le programme principal pour Calibrage

---

```

index = 0 | CompteurCapture = 0
while CompteurCapture < nbImages do
    Bloque les process
    if index != CompteurCapture then
        change le temps d'exposition par exposition[index]
        Wait pendant exposition[index] + Temps de latence
        capture = OK
        CompteurCapture = index
    end if
end while

```

---

### 3.3.2 Génération des statistiques avec le programme de test

Pour effectuer les tests de qualification, une version simplifiée du programme a été développée. elle génère des fichiers d'export pour la réponse du calibrage, pour les statistiques et les prises de vues associées. Ces données ont servi au chapitre 4.

### 3.3.3 Développement du Programme de vidéo HDR

Après la réalisation de la synchronisation des flux, une nouvelle fonctionnalité est apparue, la reconstruction HDR. Pour continuer les tests, le programme demande en paramètres la méthode à utiliser pour le calcul de reconstruction :

- 0 Mertens
- 1 Debevec

2 Robertson.

Une fois lancé, il commence par calibrer la réponse caméra. Une solution aurait été de charger un fichier de réponse pour l'initialisation. Mais, le calibrage dépend beaucoup de la configuration, car l'environnement lumineux impacte les résultats, comme précisé précédemment.

Le programme affiche les images de calibrage pour contrôle de synchronisation. Pour supprimer cet affichage il faut appuyer sur la touche [SUPPR] du pavé numérique lorsque les images sont à l'écran. Alors, le programme n'affiche plus que celles du flux vidéo HDR.

Le principe retenu est l'affichage d'une image HDR pour chaque groupe d'images, soit toutes les 4 images dans notre cas.

### 3.3.4 Ajout du correction automatique de luminosité et adaptation des temps d'exposition à l'environnement

Pour la modification automatique de la correction de luminosité, le programme calcule la valeur moyenne de l'image en ne tenant compte que du canal vert qui est le plus sensible pour l'œil humain. Et applique une correction du gamma pour que la valeur moyenne soit environs 50% comme le montre algorithme 5.

---

#### Algorithm 5 Fonction d'adaptation correction de luminosité

---

```

if nombre d'images depuis dernier traitement > valeur de déclenchement then
    imageEnGris = extractionCanal(Vert)
    laMoyenne = moyenne(imageEnGris)
    sauveGamma = gamma
    gamma = 0.5f / (laMoyenne / ancienneValeurGamma )
    if gamma n'est pas valide then
        gamma = sauveGamma
    end if
    change le paramètre gamma de la fonction de mapping par gamma
    compteur d'images = 0
end if
compteur d'images + 1

```

---

## 3.4 principales difficultés rencontrées

### 3.4.1 Difficulté de synchronisation des images et du temps d'exposition

Les webcams utilisées ne sont pas adaptées pour changer le temps d'exposition à chaque image. De plus, le paramétrage de l'exposition est indépendant du flux vidéo qui n'en transmet pas la valeur de l'image fournie. Pour compliquer un peu plus, le flux envoie en continu. Si la fonction de gestion de la capture est occupée, l'image est perdue. Elle ne sera jamais traitée comme indiquée dans le diagramme de séquence 3.1. Le programme ne peut dans ce cas avoir la certitude du positionnement de l'image dans le conteneur. Comme le calcul est fortement dépendant des valeurs d'exposition l'image fournie ne peut être correcte.

Pour corriger le dysfonctionnement, j'ai positionné un compteur de temps dans la fonction de capture. Résultat des temps de traitement trop longs avec un index d'affectation des images qui devient inopérant.

La solution m'a été inspiré par M. Dehos, mettre un drapeau qui stoppe le traitement des images au niveau de la fonction de capture et une temporisation du programme principal après chaque changement de paramétrage. Mieux mais encore des images incorrectes.

La dernière solution trouvée et l'ajout au drapeau un compteur qui empêche de traiter la première image qui arrive. La conséquence est un augmentation du temps de traitement d'un cycle complet mais une stabilité dues images obtenues.

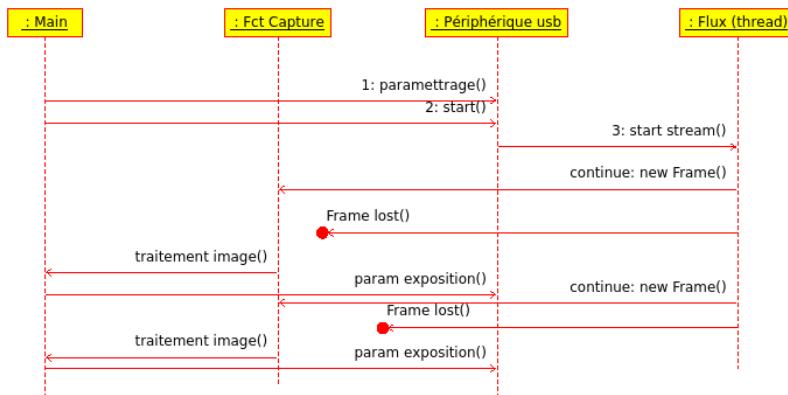


FIGURE 3.1 – On constate sur l'image : la difficulté de synchronisation des flux.

En effet, il n'y a pas de lien entre le programme principal et le flux vidéo. Le flux vidéo envoie en continu des images, sans en donner les caractéristiques. D'où la difficulté de positionner l'image reçue dans le bon conteneur.

### 3.4.2 Matériel utilisé

Pour ce qui est du matériel, je me suis aperçu tardivement que la procédure de paramétrage ne fonctionne pas de la même manière pour tous les périphériques. Le passage de la valeur d'exposition n'est pas le même d'une caméra à l'autre. Très peu documenté, il devient compliqué de fournir les bonnes valeurs aux fonctions de calcul. Pour le moment, je me suis concentré sur la seule caméra dont le temps d'exposition correspond à la librairie libuvc limitant donc le programme à un périphérique spécifique.

# Chapitre 4

## Résultats

### 4.1 Calibrages

Chaque méthode de calibrage donne un résultat totalement différent. Par conséquent, il est impossible de mixer les deux méthodes. Debevec va fournir une courbe monotone comme indiqué dans la figure 2.4. Robertson lui donnera une courbe avec beaucoup de variations.

### 4.2 Temps de traitement d'une image et nombre d'images par seconde

Le temps de traitement d'une image dépend de la méthode utilisée. Si on ne tient pas compte du calibrage qui ne doit être fait qu'une fois par lancement, voire une fois par périphérique de capture.

Les résultats sont :

- 1 Pour la reconstruction, bien qu'il reste dans des valeurs acceptables, Mertens est moins performant que les autres. Figure 4.1
- 2 Pour le mappage de l'image, on remarque l'absence de Mertens qui n'en a pas besoin et une non-linéarité des valeurs certainement due au fait qu'il n'y a qu'un seul test avec 5 images. Figure 4.2
- 3 Pour le résultat final, si le calibrage n'est pas pris en compte, Mertens reste le plus lent. Plus il y a d'images à fusionner, plus le traitement prend du temps. Pour debevec et Robertson, le nombre d'images n'impacte que très peu le temps de traitement. C'est uniquement le temps de chargement des images en mémoire qui justifie l'augmentation de temps entre 4 et 6 images. Figure 4.3

### 4.3 résultats visuels

La même scène a été filmée avec les 3 méthodes et un paramétrage pour environnement lumineux et sombre. Cela démontre les différences de rendu et l'importance de bien paramétrier les temps d'exposition.

Le paramétrage sombre n'est pas adapté, car c'est une scène fortement lumineuse et avec très peu de zones d'ombre.

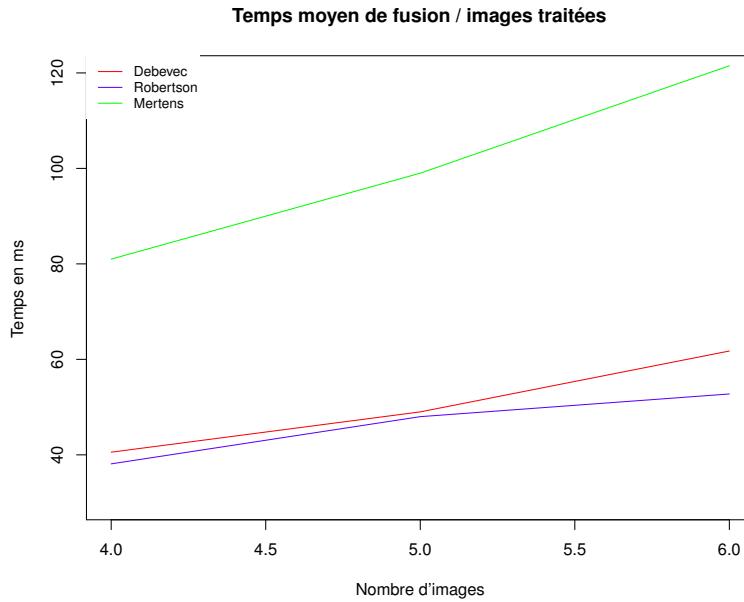


FIGURE 4.1 – Temps moyen de reconstruction de l'image.

On remarque un temps de traitement plus important pour Mertens contre Debevec et Robertson.

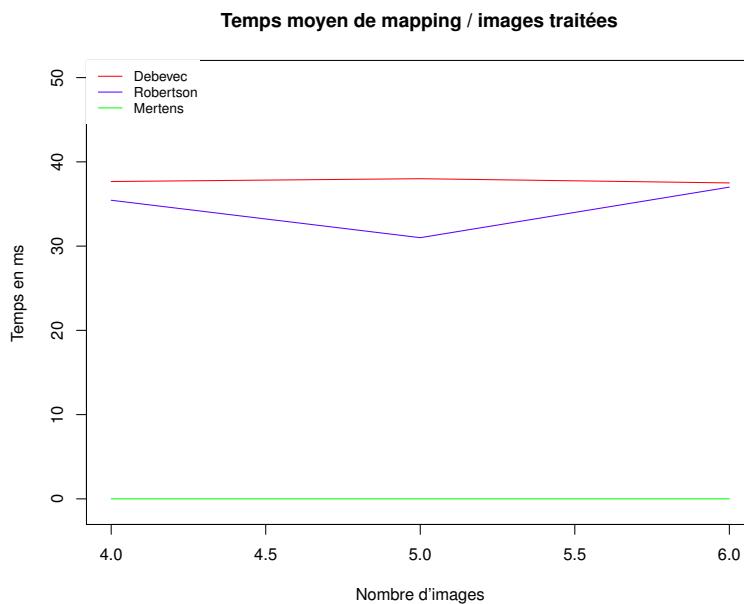


FIGURE 4.2 – Temps moyen de mappage de l'image.

Contrairement à la figure 4.1 Mertens est supérieure aux autres méthodes.

#### Rendu méthode Debevec :

- 1 Lumineux, le rendu des couleurs est proche de la réalité pour le rouge et le bleu, le jaune par contre tire légèrement sur le vert anis. Le contre jour n'a pas d'impact sur les détails. Ils apparaissent correctement et distinctement, que ce soit sur les zones fortement exposées

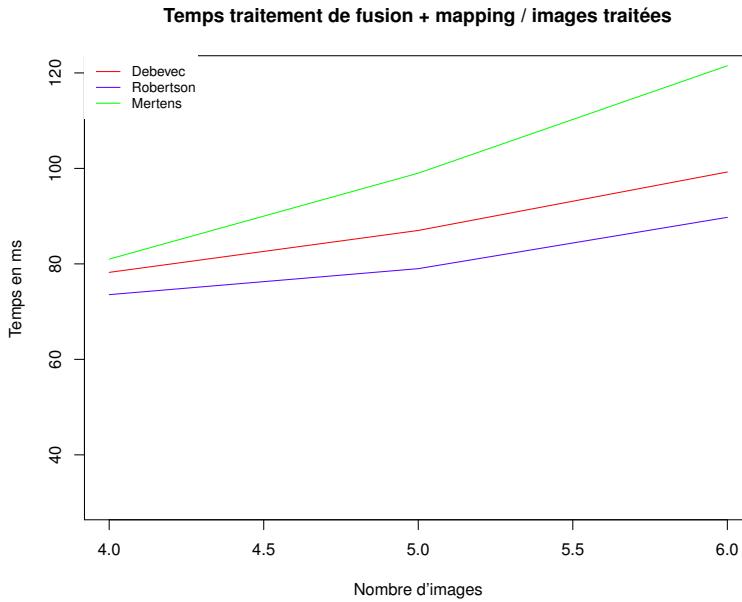


FIGURE 4.3 – Temps moyen de traitement d’images.

Dans cette figure, on remarque les vrais temps de traitement. En effet, Mertens n’effectue pas de mappage, car il est inclus dans la reconstruction. Il reste néanmoins légèrement (5 à 20%) plus gourmand en temps de traitement, même si pour lui il n’y a pas besoin d’effectuer un Calibrage du périphérique.

(extérieur) ou sous-exposées (intérieur). Le seul défaut gênant est la trame qui apparaît dans le halo de la fenêtre 4.4.

- 2 Sombre, le rendu des couleurs est moins proche de la réalité que le paramétrage lumineux, le jaune est remplacé par un vert très intense. Le contre-jour a un impact important sur les détails, qui ont presque disparu à l’extérieur. Il n’y a pas d’impact sur les zones sous-exposées (intérieur). La trame est plus présente dans le halo de la fenêtre 4.5.

#### Rendu méthode Robertson :

- 1 Lumineux, comme avecDebevec, le rendu des couleurs est proche de la réalité, comme le jaune qui tire aussi sur le vert anis. Le contre jour impacte les détails. Ils disparaissent presque totalement sur les zones fortement exposées (extérieur) ou sous-exposées (intérieur). La trame qui apparaît dans le halo de la fenêtre est plus prononcée que pour Debevec 4.6.
- 2 Sombre, le rendu des couleurs est moins proche de la réalité que le paramétrage lumineux, le jaune est remplacé par un vert très intense. Un flou s’est invité autour des objets en contre-jour et les détails extérieurs ne sont plus visibles. La trame est présente sur l’intégralité de l’image 4.7.

#### Rendu méthode Mertens :

- 1 Lumineux, le rendu est parfait, juste le jaune qui tire un peu sur le vert anis. Les détails intérieur et extérieur sont visibles. C’est la plus réussie de tous les tests. 4.8.
- 2 Sombre, même si le paramétrage ne s’y prête pas, le rendu est correct. Il manque juste un peu de contraste, car l’image apparaît un peu fade. 4.9.

Pour le rendu, Mertens sort gagnant. Il faut quand même préciser que les test ont été réalisés

en environnement fortement lumineux. Avec un environnement sombre les résultats auraient été moins flagrant et Robertson peuvent en avoir des bluffants.



FIGURE 4.4 – Images vidéo prises avec l'exposition pour environnement lumineux 1/1000" 3/1000 1/50" 1/30".

Et la reconstruction HDR Debevec + correction de luminosité. Cliché personnel



FIGURE 4.5 – Images vidéo prises avec l'exposition pour environnement sombre 1/100" 1/5" 1/2"  
1"

Et la reconstruction HDR Debevec + correction de luminosité. Cliché personnel



FIGURE 4.6 – Images vidéo prises avec l'exposition pour environnement lumineux 1/1000" 3/1000 1/50" 1/30".

Et la reconstruction HDR Robertson + correction de luminosité. Cliché personnel



FIGURE 4.7 – Images vidéo prises avec l'exposition pour environnement sombre 1/100" 1/5" 1/2"  
1"

Et la reconstruction HDR Robertson + correction de luminosité. Cliché personnel



FIGURE 4.8 – Images vidéo prises avec l'exposition pour environnement lumineux 1/1000" 3/1000 1/50" 1/30".

Et la reconstruction HDR Mertens + correction de luminosité. Cliché personnel



FIGURE 4.9 – Images vidéo prises avec l'exposition pour environnement sombre 1/100" 1/5" 1/2" 1"

Et la reconstruction HDR Mertens + correction de luminosité. Cliché personnel

# Conclusion

Du point de vue technique ou culturel, c'est un projet intéressant qui m'a apporté une nouvelle vision du traitement des photos et de la vidéo. C'était un défi complexe et intéressant. Les difficultés rencontrées avec le matériel et la résolution de la synchronisation m'ont contraint à ne mettre que la première pierre au projet. Néanmoins, le programme est capable de faire de la vidéo HDR avec un temps de latence important et limité pour le moment à un seul périphérique vidéo : Hercules HD Twist.

Il reste beaucoup de pistes à explorer :

- la réduction du nombre d'images pour le flux vidéo,
- une meilleur répartition des temps d'exposition voire une automatisation,
- ajout d'une fonction de recadrage de l'image pour éviter le flou de bougé et l'apparition de "fantomes" sur la vidéo.

# Bibliographie

- [DM] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*.
- [opea] opencv.org. <http://docs.opencv.org/3.2.0/>.
- [opeb] opencv.org. <https://github.com/opencv/opencv/archive/3.2.0.zip>.
- [PH] Photo-Hdr.com. [www.photo-hdr.com](http://www.photo-hdr.com).