

Rapport CST : *Détection de bruit dans les  
images de synthèses stéréoscopiques par  
méthodes d'apprentissage automatiques*

Jérôme BUISINE

July 2019



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Présentation de la thématique . . . . .	9
1.2	Contexte de la thèse . . . . .	10
1.3	Problématique abordée . . . . .	10
<b>2</b>	<b>Bibliographie</b>	<b>11</b>
2.1	Bibliographie . . . . .	11
<b>3</b>	<b>Travaux réalisés</b>	<b>13</b>
3.1	La base de données d'images . . . . .	13
3.2	L'approche décomposition SVD . . . . .	18
3.2.1	Description de la méthode . . . . .	18
3.2.2	Sélection des attributs . . . . .	18
3.2.3	Paramètres et résultats . . . . .	24
3.3	Etude des échantillons de rendu . . . . .	25
3.4	Les approches futures . . . . .	26
<b>4</b>	<b>Enseignements</b>	<b>27</b>
<b>5</b>	<b>Formation</b>	<b>29</b>
5.1	Conclusion . . . . .	29
	<b>Appendices</b>	<b>31</b>
<b>A</b>	<b>Transformation <math>L^*a^*b</math></b>	<b>33</b>
<b>B</b>	<b>Mean Subtracted Contrast Normalized (MSCN)</b>	<b>35</b>
<b>C</b>	<b>Modèles utilisés</b>	<b>37</b>
C.1	M1 : SVM classique . . . . .	37
C.2	M2 : Voting classifier avec 3 estimateurs . . . . .	37
C.3	M3 : Voting classifier avec 5 estimateurs . . . . .	37



# Table des figures

1.1	Intéraction du modèle de détection de bruit lors du rendu d'image	10
3.1	Découpage de l'image d'une scène pour traitement . . . . .	14
3.2	Aperçu des images de références de la base d'images . . . . .	15
3.3	Images de la scène Appart1opt02 à différents moments de l'étape de rendu . . . . .	16
3.4	Extraction de seuil perceptif pour une zone de la scène . . . . .	16
3.5	Processus de génération des bases d'apprentissage, de validation et de test . . . . .	17
3.6	Décomposition en valeurs singulières . . . . .	18
3.7	Aperçu des valeurs singulières sur la scène Appart1opt02 . . . . .	19
3.8	Aperçu des valeurs singulières de la zone 3 de la scène Appart1opt02	20
3.9	Première approche de sélection des composantes du vecteur . . . . .	22
3.10	Schéma récapitulatif du traitement de l'image . . . . .	22



# Liste des tableaux

3.1	Information sur la base d'images de synthèse . . . . .	14
-----	--	----



# Chapitre 1

## Introduction

### 1.1 Présentation de la thématique

Les moteurs de rendus d'illuminations globales sont utilisées de nos jours dans le but de générer des images photo-réalistes. La représentation d'une scène modélisée en 3D fournie en entrée au moteur de rendu permet l'obtention d'une image en sortie de celui-ci. Il définit la valeur de chaque pixel de l'image suivant les propriétés physiques des objets qui constituent la scène. Plusieurs algorithmes de rendus ont été développés durant ces dernières années dans le but d'approximer l'image finale souhaitée. Chaque méthode possède ses propres avantages et inconvénients, parfois coûteuse en temps ou pas assez précise si le temps est en manquant. Ces méthodes de rendus se basent toutes sur une méthode d'approximation Monte-Carlo (suivant la loi des grands nombres) dans le but d'approximer l'équation de rendu [?] et chaque valeur de pixel de l'image par le biais d'une moyenne empirique de l'approximation des valeurs des pixels durant le rendu.

Dans l'objectif d'évaluer la qualité d'une image, des approches et métriques de calculs ont été développées et proposées dans la littérature. Ces métriques sont parfois spécifiques pour des distortions d'images telles que le flou gaussien, JPEG2000, compression JPEG bruitée, ringing. Généralement ciblé sur du bruit additif. Globalement, les approches publiées sont appliquées pour la mesure de qualité d'une image dite naturelle et très peu pour mesurer la qualité d'une image photo-réaliste. En effet, lors du rendu de l'image, un bruit peut être perceptible par l'humain. Il ne s'agit ici en aucun cas d'un bruit additif mais plutôt un bruit dit « impulsif », dû au fait de la complexité d'approximation des valeurs finales de pixels dans une zone de la scène provoquant une erreur numérique perceptible humainement.

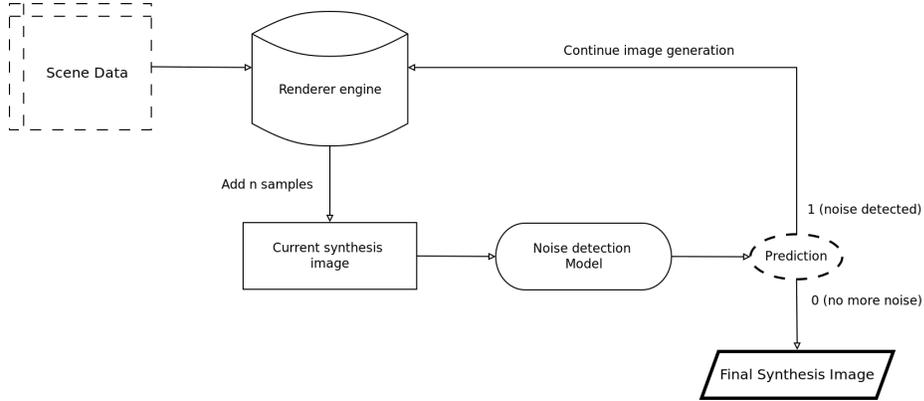


FIGURE 1.1 – Représentation de l’interaction du modèle de détection souhaité de bruit lors du rendu d’une image de synthèse à partir des données de la scène 3D

## 1.2 Contexte de la thèse

Le sujet de thèse présenté ici s’intitule « Détection de bruit dans les images stéréoscopiques par méthodes d’apprentissage automatiques ». La thèse s’incruste dans un projet ANR donc l’objectif final serait de pouvoir concevoir un moteur rendu mixte mêlant rendu temps réel (une image calculée en moins de 30 ms) et photo-réalisme. C’est à dire cibler la partie à améliorer par rapport au regard de l’utilisateur dans un support de rendu stéréoscopique. L’objectif de la thèse est double, à la fois détecter le bruit, le quantifier à partir de l’image uniquement et non les informations de la scène (si possible) puis proposer une méthode de filtrage/débruitage (post-traitement) permettant d’améliorer son aperçu. Tout en gardant pour objectif que cela puisse être réalisé en temps réel.

## 1.3 Problématique abordée

Comme évoqué précédemment, l’objectif de la thèse est double pouvoir à la fois détecter le bruit dans une image de synthèse issue d’un moteur de rendu et mais également proposer une méthode de réduction de ce bruit perçu (denoising method).

La Fig. 1.1 détaille davantage la manière dont le problème est abordé et surtout elle permet de cibler la partie sur laquelle la problématique de la thèse s’incruste. Nous avons donc le modèle de détection de bruit (modèle de classification binaire) qui permettrait de prédire si le moteur de rendu a encore la nécessité d’améliorer l’image ou non. On pourrait bien entendu ici, mettre en place une étape de post-traitement pour amélioration de l’image avant évaluation du bruit contenu dans celle-ci.

## Chapitre 2

# Bibliographie

### 2.1 Bibliographie

Découper ici la bibliographie en sous sections :

- Algorithmes de rendus
- Image quality metrics
- Approches Deep learning pures
- Application détection de bruits images de synthèse
- SVD



## Chapitre 3

# Travaux réalisés

### 3.1 La base de données d'images

Cette partie sera consacrée à la présentation des différents travaux réalisés dans le cadre de la première année de thèse. La majorité des travaux ne donnant pas de résultats unanime, chaque approche sera donc détaillée pour mettre en avant ses avantages, ses inconvénients, les raisons pour lesquelles l'approche n'est pas forcément concluante pour enfin s'ouvrir à de nouveaux axes et méthodes d'applications.

Pour l'ensemble des travaux réalisés durant cette première année, l'image de taille  $800 \times 800$  a été découpée en 16 zones de  $200 \times 200$  (voir Fig. 3.1). C'est ensuite une zone qui est traitée pour extraction des informations caractérisant le bruit présent dans le celui-ci. La première partie de la thèse porte donc sur cette étape du processus de classification.

Une base de données d'images a été développée, elle contient 9 scènes ainsi que les seuils perceptifs prelevés lors d'expérience pour chaque zone d'une scène. Chaque sujet de l'expérience devait partir de l'image la plus bruitée augmenter la qualité (nombre d'échantillons) de chaque zone pour se rapprocher au maximum de l'image de référence (image obtenue avec le plus grand nombre d'échantillons). Dans le cadre de ses premières expériences, uniquement les scènes issues du moteur de rendu Maxwell ont été sélectionnées (voir tableau 3.1 et Fig. 3.2). La Fig. 3.3 propose un aperçu du bruit perceptible lors du processus rendu d'une image de synthèse, ici sur la scène `Appart1opt02`.

La Fig. 3.4 présente la manière dont chaque zone des différentes images (nombre d'échantillons) sont labelisés.

Le processus de génération des données utilisées pour création des bases d'apprentissage, de validation et de test pour le modèle mathématique choisi est détaillé dans la Fig. 3.5. Ce processus vient de l'hypothèse que chaque zone est traitée indépendamment de la scène. Plutôt que d'utiliser 3 scènes en apprentissage et une en test, la sélection de  $z$  zones pour former la base d'apprentissage

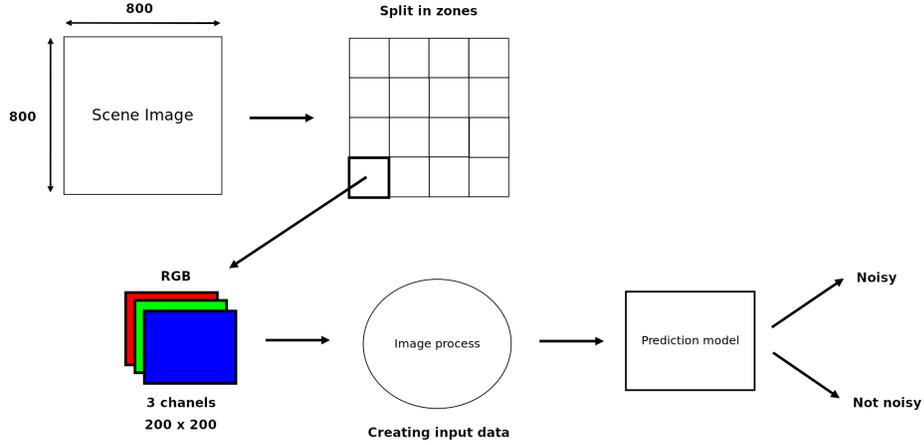


FIGURE 3.1 – Découpage de l’image d’une scène pour extraction des caractéristiques des zones de celle-ci

ID	Name	Indices	Step	Images
A	Appart02	20 → 900	10	91
D	Cuisine01	20 → 1200	10	121
G	SdbCentre	20 → 950	10	96
H	SdbDroite	20 → 950	10	96

TABLE 3.1 – Information sur la base d’images de synthèse

sont réalisées aléatoirement avec  $z \in [4, 6, 8, 10, 12]$ . Les zones restantes seront alors utilisées pour les bases de validation et de test du modèle.

Afin de mesurer l’importance de la séparabilité des données à partir du seuil, 3 manières de construire les données finales ont été abordées :

- **all** : consistant à prendre l’ensemble des données.
- **center** : consistant à ne prendre que les images à échantillons proche de 150 du seuil estimé de la zone.
- **split** : les données non sélectionnées par la seconde (données fortement extérieur au seuil).

Enfin, la comparaison des performances de modèles se fera via le score AUC - ROC (Area Under The Curve Receiver Operating Characteristics) sur la base de test. Il permet de mesurer la performance d’un modèle pour un problème de classification, plus particulièrement à quel point il arrive à bien séparer les classes du problème.



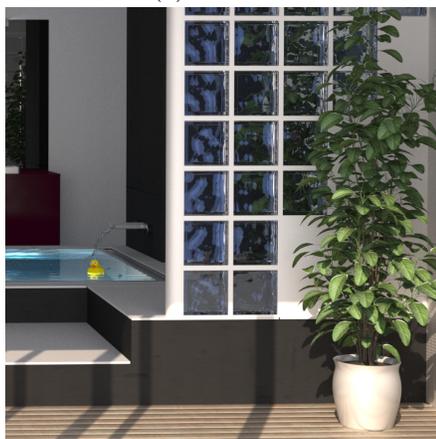
(a) Apart1opt02



(b) Cuisine01



(c) SdbCentre



(d) SdbDroite

FIGURE 3.2 – Aperçu des images de références de la base d'images

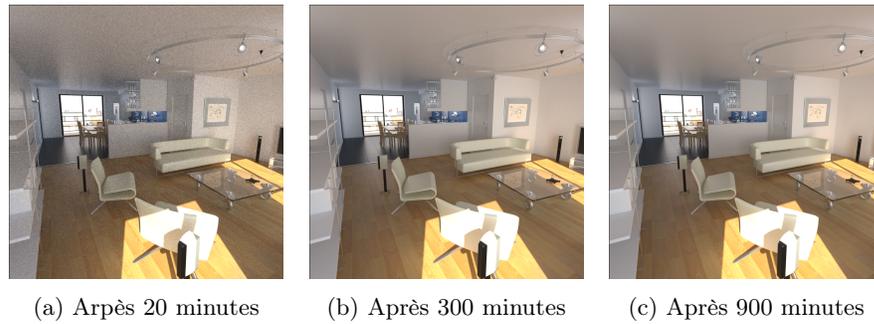


FIGURE 3.3 – Images de la scène *Appart1opt02* à différents moments de l'étape de rendu. lors des premières minutes on peut remarquer un bruit perceptible. Il s'agit d'une approximation de pixels non précise obtenue lors du rendu.

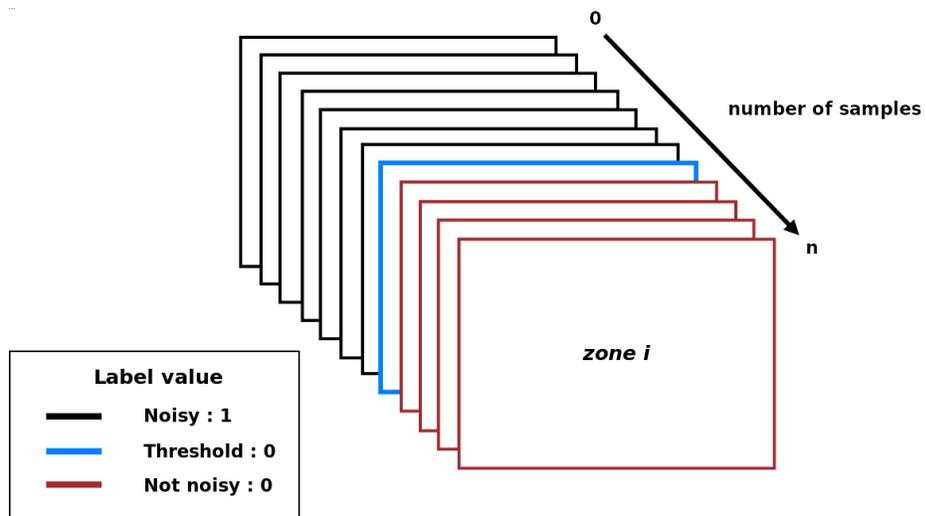


FIGURE 3.4 – Le seuil perceptif est identifié par la moyenne des scores obtenus sur les sujets de l'expérience. Les images situées avant ce seuil perceptif sont considérées comme bruitées, celles situées après, comme non bruitées.

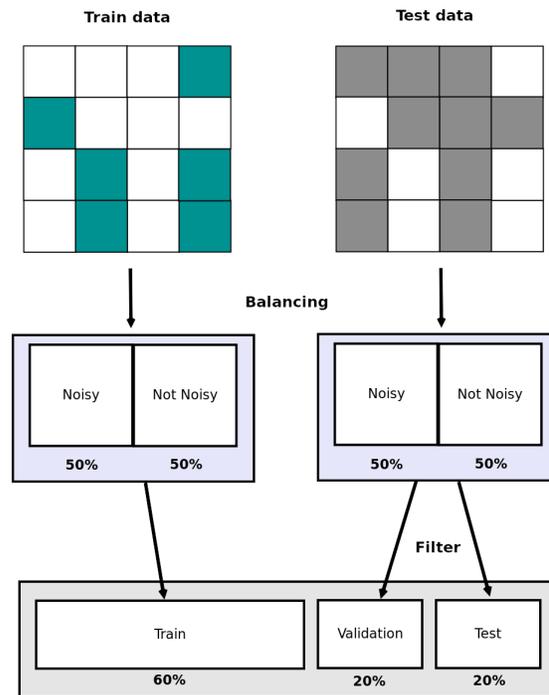


FIGURE 3.5 – Dans cet exemple, 6 zones ont été sélectionnées pour la base d'apprentissage. Un équilibrage des classes est réalisé pour ne pas biaiser le modèle lors de l'apprentissage puis la répartition des différentes bases est effectuée.

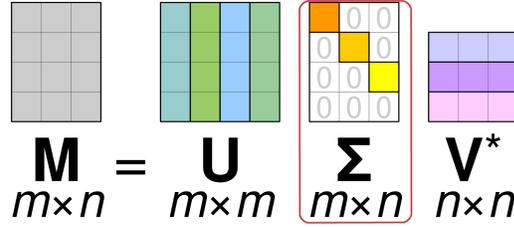


FIGURE 3.6 – Méthode de factorisation utilisée pour réduction de dimension de l'image : décomposition en valeurs singulières

## 3.2 L'approche décomposition SVD

### 3.2.1 Description de la méthode

La décomposition en valeurs singulières (SVD) est une méthode de factorisation de matrice. L'utiliser dans le cadre de la réduction d'informations pour le traitement de l'image pourrait être intéressant. La Fig 3.6 propose un aperçu de la factorisation proposée par la méthode.

La méthode propose donc une factorisation de la matrice de la manière suivante :

$$M = U \times \Sigma \times V^*$$

C'est le vecteur de valeurs singulières ( $\Sigma$ ) auquel nous nous sommes principalement intéressé. En effet, la méthode de décomposition appliquée à une image réduite au canal de luminance  $L$  de la transformation  $L^*a*b$  (voir annexe A) ou la transformation MSCN [?] (voir annexe B) qui permet d'obtenir le vecteur de valeurs singulières  $\Sigma$ .

Pour une image de  $200 \times 200$  d'une zone nous obtenons donc un vecteur de valeurs singulières de taille 200. C'est ce vecteur que nous allons traiter par la suite.

Pourquoi s'être intéressé à la décomposition SVD ? Nous avons pu observer que suivant le nombre d'échantillons utilisés pour générer une image d'un scène, le vecteur de valeurs singulières semblaient significativement bien distinguer le niveau de bruit jusqu'à l'image de référence (voir Fig. 3.7). La Fig. 3.8 propose également un aperçu des valeurs singulières mais ici sur une zone précise de la scène, la zone 3.

### 3.2.2 Sélection des attributs

Nous avons donc pour chaque image traitée, un vecteur de 200 valeurs à traiter. Dans cette section nous allons présenter les différentes manières (transformations de l'image) d'obtenir ce vecteur ainsi que les différentes approches de sélection des composantes de celui-ci.

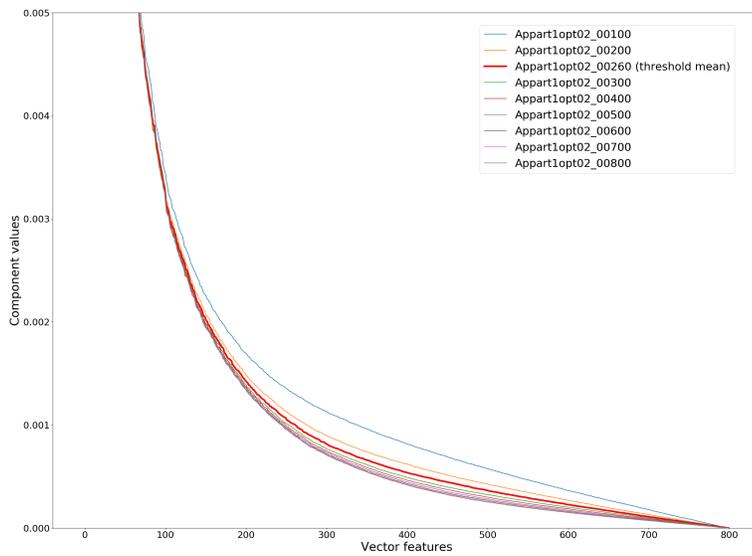


FIGURE 3.7 – Aperçu du vecteur de valeurs singulières des images de la scène Appart1opt02 à différents niveaux de bruits. Ici le vecteur possède 800 valeurs car la décomposition a été appliquée sur l'ensemble de l'image et non une zone spécifique. À noter que le vecteur est obtenu à partir du canal  $\mathbf{L}$  de l'image et qu'il a été ici normalisé.

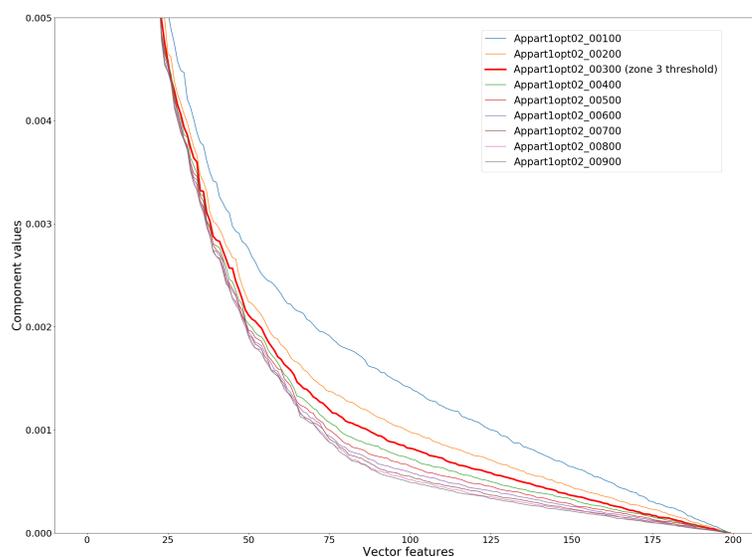


FIGURE 3.8 – Aperçu du vecteur de valeurs singulières des sous-images de la zone 3 de la scène Appart1opt02 à différents niveaux de bruits. Ici le vecteur possède 200 valeurs car la décomposition a été appliquée sur une zone spécifique. À noter que le vecteur est obtenu à partir du canal **L** de la sous-image et qu'il a été ici normalisé.

### Liste des attributs extrais

La liste des transformations ci-dessous appliquées à l'image sont utilisées pour obtention du vecteur de valeurs singulières de 200 valeurs via la décomposition SVD.

- **lab** : utilisation de la luminance de l'image.
- **mfcn** : utilisation de la matrice MSCN.
- **low\_bits\_2** : utilisation uniquement des 2 bits de poids faibles de la matrice de luminance (un octet de représentation).
- **low\_bits\_3** : utilisation uniquement des 3 bits de poids faibles de la matrice de luminance.
- **low\_bits\_4** : utilisation uniquement des 4 bits de poids faibles de la matrice de luminance.
- **low\_bits\_5** : utilisation uniquement des 5 bits de poids faibles de la matrice de luminance.
- **low\_bits\_6** : utilisation uniquement des 6 bits de poids faibles de la matrice de luminance.
- **low\_bits\_4\_shifted\_2** : utilisation uniquement des 4 bits de centraux de la matrice de luminance.
- **ica\_diff** : utilisation d'une matrice calculée à partir de la différence entre la matrice de luminance et une image reconstruite à partir des 50 composantes principales à partir de la méthode « Independent Component Analysis » (ICA).
- **svd\_trunc\_diff** : utilisation d'une matrice calculée à partir de la différence entre la matrice de luminance et une image reconstruite à partir des 30 composantes principales à partir de la méthode de décomposition SVD.
- **ipca\_diff** : utilisation d'une matrice calculée à partir de la différence entre la matrice de luminance et une image reconstruite à partir des 20 composantes principales à partir de la méthode de « Incremental Principal Components Analysis » (IPCA).
- **svd\_reconstruct** : utilisation d'une image reconstruite à partir des 110 dernières composantes de la décomposition SVD.

### Approche dites naïve de sélection d'attributs

La première approche de sélection d'attributs consiste à choisir  $n$  composantes du vecteur de valeurs singulières avec  $n \in [4, 8, 16, 26, 32, 40]$  à une position  $P$  avec  $|P| = 5$  (voir Fig. 3.9). Cette approche peut paraître naïve mais elle permet déjà d'essayer un certains nombres de paramètres.

La Fig. 3.10 permet un aperçu récapitulatif de la manière dont l'image est traitée pour la sélection d'attributs d'entrée au modèle d'apprentissage.

### Autres approches de sélection des attributs

Étant donné l'approche naïve de sélection d'attributs et le fait que les valeurs des composantes d'une scène à une autre différent, la capacité d'un modèle

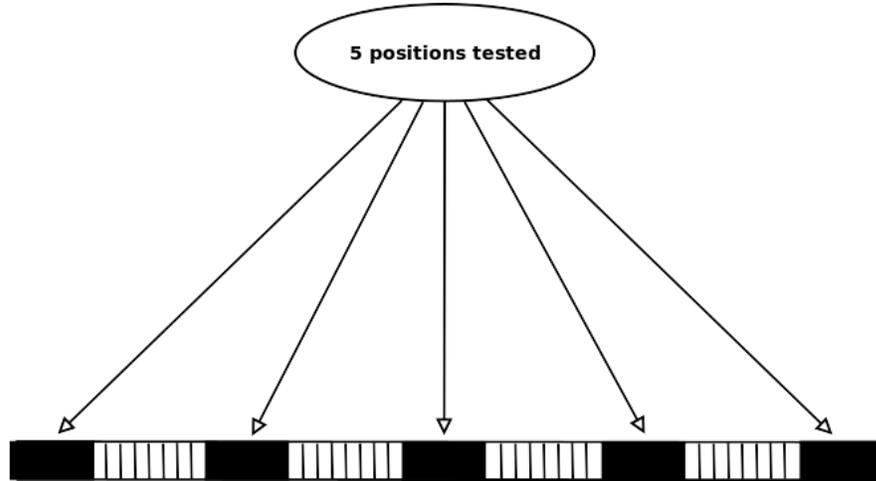


FIGURE 3.9 – Sélection d’attributs sur différentes zones du vecteur de valeurs singulières. La position est choisie comme telle, au début, au quart, au centre ; au trois quarts et en fin de vecteur. Si l’on décide de prendre  $n = 20$  composantes à partir de la position  $P$  centrale, l’intervalle sélectionné sera  $[90, 120[$

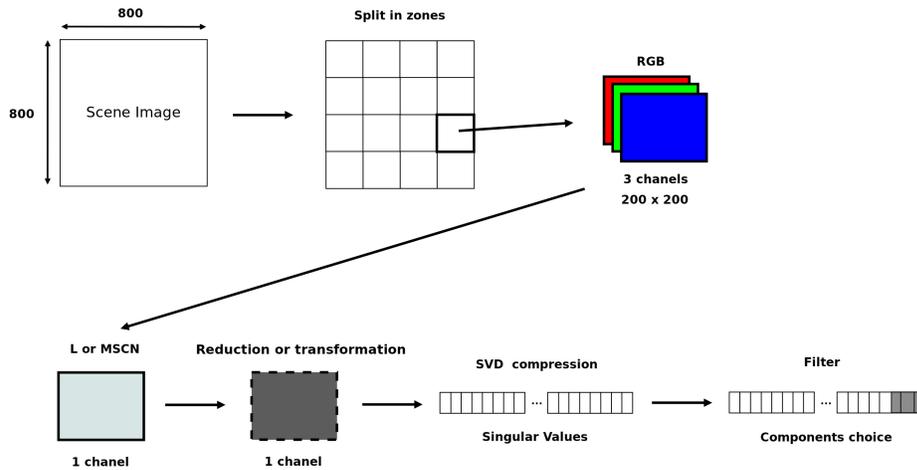


FIGURE 3.10 – Schéma récapitulatif de traitement de l’image dans l’objectif de sélectionner les meilleurs attributs pour entrée au modèle

de classifier correctement une image peut être impactée. Les listes ci-dessous énumèrent les approches exploitées dans le cadre de la thèse.

Sur l'ensemble des données :

- **highest\_corr\_sv** : récupération des composantes les plus corrélées entre-elles (score de corrélation) sur l'ensemble des données disponibles
- **lowest\_corr\_sv** : récupération des composantes les moins corrélées entre-elles (score de corrélation) sur l'ensemble des données disponibles

Sur l'image seule :

- **sv\_std\_filters** : génération de 4 images supplémentaires à l'image de niveau de luminance en entrée, à partir des filtres « wiener » (fenêtres kernel [3, 3] et [5, 5]) et « median » (fenêtres kernel [3, 3] et [5, 5]).
- **wave\_sv\_std\_filters** : génération de 5 images supplémentaires à l'image de niveau de luminance en entrée, à partir des filtres « wiener » (fenêtres kernel [3, 3] et [5, 5]), « median » (fenêtres kernel [3, 3] et [5, 5]) et « wavelet ».
- **sv\_std\_filters\_full** : génération de 13 images supplémentaires à l'image de niveau de luminance en entrée, à partir des filtres « wiener » (fenêtres kernel [3, 3] et [5, 5]), « median » (fenêtres kernel [3, 3] et [5, 5]), « mean » (2 kernels), « gaussian blur » (6 kernels) et « wavelet ».

La décomposition est ensuite appliquée à chacune des images disponibles dans la liste des images. Ce qui donne une matrice de taille  $l \times 200 \times 200$  avec  $l$  le nombre d'images dans la liste. Pour chacune des composantes, les valeurs qui lui sont trouvées ( $l$  valeurs) sont normalisées suivant cette composante et la valeur de l'écart-type est calculée. Les  $n$  plus grandes (highest) ou petites (lowest) valeurs d'écart-type permettent de cibler les  $n$  composantes à sélectionner (sélection dynamique pour l'image).

Sur l'image seule :

- **sv\_entropy\_std\_filters** : les mêmes filtres que « sv\_std\_filters\_full » soit, 13 images générées supplémentaires sont utilisées. La contribution à l'entropie de chaque composante du vecteur de valeur singulières est calculée. C'est ensuite la valeur de l'écart-type de ces contributions d'entropie qui sont utilisées comme valeurs de sélection des composantes.

Pourquoi s'intéresser à un traitement d'image seul pour sélection des attributs? Étant donné les résultats sur une approche de sélection d'interval, il semblerait que la sélection des attributs soient fortement liée à la scène elle-même (structure de la scène). Une approche proposant dynamiquement de choisir les composantes pour une scène (ici l'image directement) serait plus judicieuse et permettrait d'avoir une méthode générique qu'importe les images de nouvelles scènes à traiter. Une étude plus approfondie permettrait de mettre en avant pour chaque scène les composantes les plus utilisées (répartition/distribution de sélection des composantes).

### 3.2.3 Paramètres et résultats

Avant de mettre en avant les résultats issus de la décomposition SVD, il est important de rappeler l'architecture des modèles utilisés, les manières de normaliser les données ainsi que le récapitulatif de l'ensemble des paramètres.

#### Modèles

L'annexe C présente ces différents modèles. En voici la liste :

- **M1** : Support Vector Machine
- **M2** : Ensemble model (3 sub-models)
- **M3** : Ensemble model v2 (5 sub-models)

Ces ensemble de modèles sont des « voting classifier » avec le principe de le vote équitable et réglementé sur les élections « douces ».

#### Normalisation des données

Au niveau du traitement des données, nous avons opter sur trois manières de fournir les données en entrée aux modèles :

- **svd** : sans normalisation
- **svdn** : le sous-vector est normalisé avec ces propres valeurs
- **svdne** : le sous-vector est normalisé en utilisant les valeurs minimale et maximale des sous-vecteurs ou composantes de l'ensemble du dataset.

#### Les différents paramètres

#### Résultats

### **3.3 Etude des échantillons de rendu**

### 3.4 Les approches futures

## Chapitre 4

# Enseignements

- **L1 Mathématiques** : Fondements de l'algorithmique (19.38 Eq TD)
- **L2 Informatique** : Algorithmique avancée en C++ (12 Eq TD)
- **DUT 1<sup>re</sup> année Informatique** : Bases de la POO (28 Eq TD)



## Chapitre 5

# Formation

### 5.1 Conclusion



# Appendices



## Annexe A

# Transformation L\*a\*b

L\*a\*b est obtenu en utilisant la transformation XYZ de la matrice d'image (espace CIE XYZ). Les équations (A.1), (A.2), (A.3) détaillent la manière dont chaque canal est calculé.

$$L = 116f_y - 16 \quad (\text{A.1})$$

$$a = 500(f_x - f_y) \quad (\text{A.2})$$

$$b = 200(f_y - f_z) \quad (\text{A.3})$$

où

$$f_x = \begin{cases} \sqrt[3]{x_r} & \text{if } x_r > \epsilon \\ \frac{\kappa x_r + 16}{116} & \text{otherwise} \end{cases} \quad (\text{A.4})$$

$$f_y = \begin{cases} \sqrt[3]{y_r} & \text{if } y_r > \epsilon \\ \frac{\kappa y_r + 16}{116} & \text{otherwise} \end{cases} \quad (\text{A.5})$$

$$f_z = \begin{cases} \sqrt[3]{z_r} & \text{if } z_r > \epsilon \\ \frac{\kappa z_r + 16}{116} & \text{otherwise} \end{cases} \quad (\text{A.6})$$

$$x_r = \frac{X}{X_r}, \quad y_r = \frac{Y}{Y_r}, \quad z_r = \frac{Z}{Z_r}$$

$$\epsilon = \begin{cases} 0.008856 & \text{Actual CIE standard} \\ 216/24389 & \text{Intent of the CIE standard} \end{cases} \quad (\text{A.7})$$

$$\kappa = \begin{cases} 903.3 & \text{Actual CIE standard} \\ 24389/27 & \text{Intent of the CIE standard} \end{cases} \quad (\text{A.8})$$



## Annexe B

# Mean Subtracted Contrast Normalized (MSCN)

Pour construire la matrice MSCN, il faut premièrement convertir l'image RGB en une image de niveau de gris. La matrice MSCN va permettre d'extraire les informations naturelles de structures de la scène (Natural Scene Structure : NSS) de cette image en niveau de gris. Une opération est appliquée An operation is applied à la luminance de l'image  $I(i, j)$  pour obtenir :

$$\hat{I}(i, j) = \frac{I(i, j) - \mu(i, j)}{\sigma(i, j) + C} \quad (\text{B.1})$$

où  $i \in 1, 2 \dots M, j \in 1, 2 \dots N$  sont les indices spatiaux,  $M, N$  sont la hauteur et la largeur de l'image respectivement,  $C$  est une constante, de valeur 1 pour prévenir des instabilité et où

$$\mu(i, j) = \sum_{k=-K}^K \sum_{l=-L}^L w_{k,l} I_{k,l}(i, j) \quad (\text{B.2})$$

$$\sigma(i, j) = \sqrt{\sum_{k=-K}^K \sum_{l=-L}^L w_{k,l} (I_{k,l}(i, j) - \mu(i, j))^2} \quad (\text{B.3})$$

36 ANNEXE B. MEAN SUBTRACTED CONTRAST NORMALIZED (MSCN)

# Annexe C

## Modèles utilisés

### C.1 M1 : SVM classique

Nous utilisons le fameux modèle SVM (Support Vector Machine) pour prédire le bruit et non les images bruyantes. SVM est principalement utilisé et fonctionne très bien pour les problèmes NR-IQA (No Reference Image Quality Assessment) avec ces paramètres :

- Kernel : *rbf*
- $C \in \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$
- $\gamma \in \{0.001, 0.01, 0.1, 1, 5, 10, 100\}$
- $cv = 10$  (paramètre de validation croisée pour les données d'entraînement, ce qui augmente la précision et évite le surapprentissage)

Ensuite, tous ces paramètres sont testés à l'aide du principe de recherche par grille et le meilleur modèle est obtenu.

### C.2 M2 : Voting classifier avec 3 estimateurs

Nous proposons également un modèle Voting classifier (ensemble de modèles) qui est composé de 3 sous modèles :

- **SVM** avec un kernel RBF comme présenté précédemment.
- **Random Forest** avec 100 estimateurs
- **Logistic Regression** avec le noyau *liblinear* et le paramètre *ovr* pour la classification

### C.3 M3 : Voting classifier avec 5 estimateurs

Ce nouveau modèle est aussi un Voting classifier et est composé de 5 sous modèles :

- **SVM** avec le RBF Kernel comme présenté précédemment

- **Random Forest** avec 100 estimateurs
- **Logistic Regression** avec le noyau *liblinear* et le paramètre *ovr* pour la classification
- **KNeighbors Classifier** avec seulement deux voisins (bruité, non bruité)
- **Gradient Boosting Classifier** avec 100 estimateurs, une fonction de perte réglée sur *déviante* et *pas d'apprentissage* de 1.0