



Rapport : Traitement des données audio-visuelles

Adam Rochdi

Année: 2024/2025

Sommaire

1	Préambule	3
2	TP4	4
2.1	Introduction	4
2.2	Rappel du sujet	4
2.2.1	Approche initiale	4
2.2.2	Detection par champ de Markov	4
2.2.3	Détection par processus ponctuel marqué	5
2.3	Pour aller plus loin	5
2.3.1	Modélisation plus réaliste des objets (forme elliptique)	5
2.3.2	Segmentation et binarisation de l'image	6
2.3.3	Utilisation de YOLO	7
2.3.4	Comparaison	8
2.4	Conclusion	8
3	TP5	9
3.1	Introduction	9
3.2	Rappel du sujet	9
3.2.1	Débruitage par variation totale	9
3.2.2	Inpainting par diffusion	9
3.3	Pour aller plus loin	10
3.3.1	Inpainting par rapiéçage	10
3.3.2	Ajout d'un ordre de priorité	11
3.4	Conclusion	11
4	TP7	12
4.1	Introduction	12
4.2	Rappel du sujet	12
4.2.1	Collage naïf	12
4.2.2	Collage par équation de Poisson	12
4.2.3	Décoloration partielle sans segmentation	13
4.3	Pour aller plus loin	13
4.3.1	Autre technique de photomontage	13
4.4	Conclusion	14
5	TP8	15
5.1	Introduction	15
5.2	Rappel du sujet	15
5.2.1	Décomposition par filtrage fréquentiel	15
5.2.2	Décomposition par variation totale (modèle ROF)	16
5.3	Pour aller plus loin	16
5.3.1	Modèle TV-Hilbert	16
5.3.2	Transfert de style neuronal (Gatys et al.)	17
5.4	Conclusion	18
6	TP11	20
6.1	Introduction	20
6.2	Rappel du sujet	20
6.2.1	Détection des pics spectraux	20

6.2.2	Appariement de pics	20
6.2.3	Indexation	20
6.2.4	Reconnaissance simplifiée	20
6.3	Pour aller plus loin	21
6.3.1	Reconnaissance musicale avancée	21
6.3.2	Comparaison avec une approche par apprentissage profond	21
6.4	Conclusion	22
7	Conclusion générale	23
8	Bibliographie	23

1 Préambule

Avant d'entrer dans le détail des travaux pratiques, je tiens à exprimer ma reconnaissance envers les professeurs **Jean-Denis Durou** et **Jean Mélou**. Tout au long de cette UE, et en particulier lors des TP, nous avons ressenti un véritable respect pour notre capacité à apprendre, expérimenter, et réfléchir comme de futurs ingénieurs — voire comme des jeunes chercheurs.

Ce regard porté sur nous, exigeant mais bienveillant, a profondément influencé ma manière d'aborder ces travaux. Il m'a personnellement motivé à m'investir pleinement, à aller plus loin que ce qui était demandé, et à explorer avec curiosité des pistes parfois complexes mais passionnantes. C'est dans cet esprit que ce rapport a été rédigé.

Je tiens également à souligner la pertinence des sujets abordés, qui couvrent un large spectre des applications modernes du traitement d'image et du signal. Cette diversité m'a offert une vision panoramique du domaine et m'a encouragés à établir des ponts entre différentes disciplines.

2 TP4

2.1 Introduction

Ce TP vise à détecter et compter automatiquement les flamants roses dans une image aérienne, dans un contexte de suivi écologique. La tâche est difficile en raison de la petite taille des individus, de leur teinte claire sur un fond complexe, et du besoin de précision dans le comptage. Nous utilisons une méthode probabiliste basée sur des modèles de type champ de Markov et processus ponctuel marqué, qui permet de localiser les individus en se basant uniquement sur l'intensité lumineuse et la répartition spatiale des objets.

2.2 Rappel du sujet

2.2.1 Approche initiale

Le script exercice_0 propose une méthode simple : positionner aléatoirement N disques dans l'image et chercher à maximiser la somme des niveaux de gris moyens à l'intérieur de ces disques. Bien qu'intuitive, cette approche présente plusieurs limites. Elle n'impose aucune contrainte de non-recouvrement, ce qui peut conduire à placer plusieurs disques sur un même flamant. De plus, des zones très claires sans flamants peuvent être sélectionnées, tandis que des individus moins visibles peuvent être ignorés. Cette méthode n'est donc pas adaptée à un comptage fiable, ce qui justifie le recours à un modèle plus rigoureux.

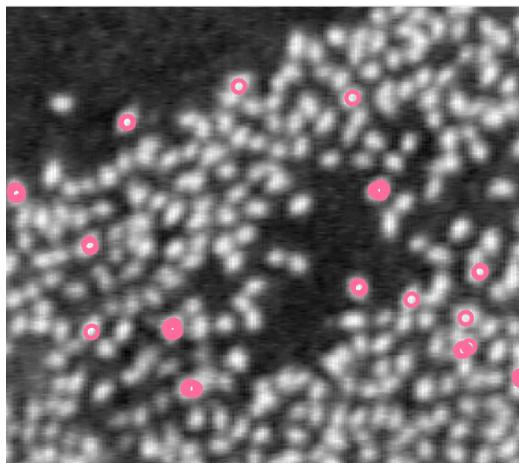


Figure 1: Detection naïve

2.2.2 Détection par champ de Markov

Nous formulons la détection de flamants roses comme un problème de Maximisation a posteriori (MAP) sous contrainte de non-chevauchement. Chaque flamant est modélisé de façon simplifiée par un disque de rayon R , et une énergie $U(c)$ est définie pour une configuration c de N disques. Cette énergie combine un terme d'attache aux données (basé sur la luminance moyenne à l'intérieur de chaque disque, qui favorise les disques positionnés sur des zones claires) et un terme de régularisation (pénalité β si deux disques sont trop proches, empêchant le recouvrement des cibles). Avec un paramètre de séparation très grand ($\beta \rightarrow +\infty$), minimiser $U(c)$ revient à sélectionner N positions correspondant aux N taches claires les plus marquées, tout en assurant une distance minimale entre disques.

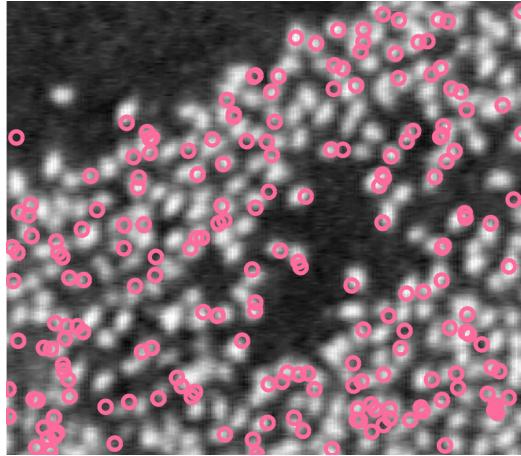


Figure 2: Detection par champ de Markov

2.2.3 Détection par processus ponctuel marqué

Nous étendons ensuite le modèle via un processus ponctuel marqué (PPM), qui autorise un nombre variable N d'objets en introduisant des mécanismes aléatoires de naissance et de mort de disques. L'énergie $U(c)$ est redéfinie de manière similaire (somme des énergies individuelles et pénalités de recouvrement) mais avec une fonction d'attache aux données modifiée. En effet, pour qu'un disque bien positionné (sur un flamant réel) soit favorisé et qu'un disque mal positionné soit découragé, on utilise une fonction sigmoïde pour l'énergie individuelle. Celle-ci décroît quand l'intensité moyenne du disque augmente et produit des valeurs négatives pour les disques correspondant à de vraies cibles et positives pour les faux positifs, respectant ainsi les contraintes du modèle.

2.3 Pour aller plus loin

2.3.1 Modélisation plus réaliste des objets (forme elliptique)

Au départ, nous avons modélisé chaque flamant rose par un disque. Cela fonctionnait raisonnablement bien, mais dès les premières visualisations, une évidence s'imposait : les flamants sont bien plus allongés que ronds. Nous avons alors fait évoluer le modèle vers une représentation elliptique, avec un demi-grand axe $a = R$ et un demi-petit axe $b = 0,75R$. Ce changement, simple en apparence, a permis de mieux épouser la forme des flamants (ou des bateaux, dans les images ultérieures), tout en limitant le recouvrement entre objets. L'ajout d'un angle θ d'orientation a aussi apporté plus de souplesse, permettant de détecter des objets inclinés de manière plus réaliste. La figure suivante illustre cette amélioration.

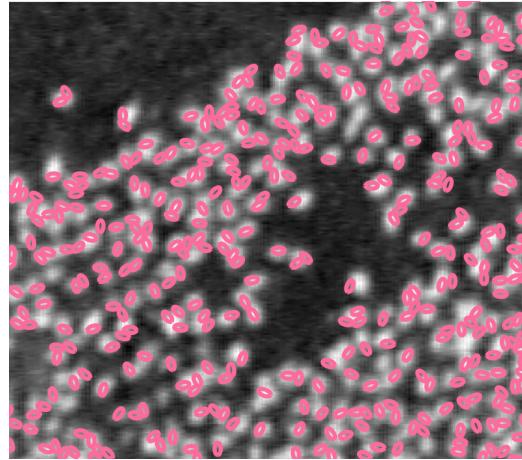


Figure 3: Forme elliptique

2.3.2 Segmentation et binarisation de l'image

Malgré cette amélioration morphologique, je constatais encore des erreurs : certains objets étaient mal détectés, voire ignorés, notamment dans des scènes plus complexes comme une marina pleine de bateaux.



Figure 4: Marina pleine de bateaux

C'est là qu'une idée m'est venue : et si je « préparais » l'image avant même de détecter ?

J'ai alors expérimenté une segmentation automatique par `kmeans`, afin de séparer les zones claires (souvent les objets) du fond sombre. Une fois cette segmentation obtenue, je l'ai convertie en image binaire avec `I = mat2gray(I)` ; pour simplifier l'analyse.

Résultat : l'algorithme probabiliste se concentre désormais uniquement sur les zones utiles. Cela accélère la convergence mais n'améliore pas beaucoup l'exactitude de la détection.

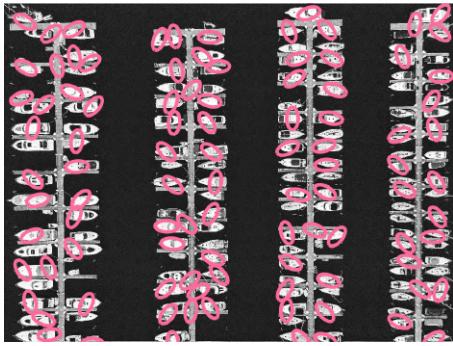


Figure 5: Image non binarisée

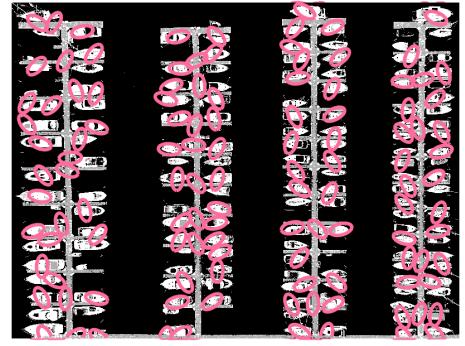


Figure 6: Image binarisée

2.3.3 Utilisation de YOLO

Le modèle probabiliste repose fortement sur la luminosité ce qui me pose un problème, surtout si les objets sont sombres, ou partiellement cachés. C'est à ce moment-là que je me suis dit : « *Et si j'utilisais du deep learning ?* »

J'ai alors utilisé l'approche YOLO (*You Only Look Once*), un modèle de détection d'objets puissant basé sur l'apprentissage supervisé.

J'ai commencé par constituer un petit jeu de données : quelques images de bateaux vus du ciel, que j'ai annotées à l'aide de la plateforme **Roboflow**. Chaque bateau a été entouré par une boîte englobante, comme illustré ci-dessous :

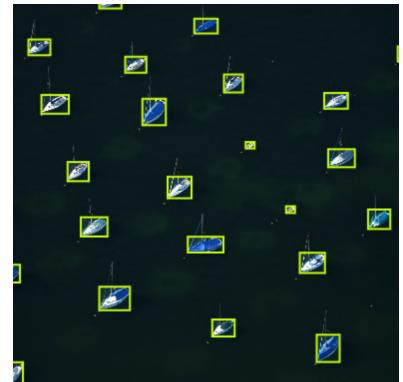


Figure 7: Extraits du jeu d'entraînement annoté avec Roboflow

Une fois les annotations prêtes, j'ai entraîné un modèle YOLOv11 avec ce dataset. Après 50 époques, le modèle a commencé à bien détecter les bateaux, même dans des zones denses ou partiellement recouvertes.

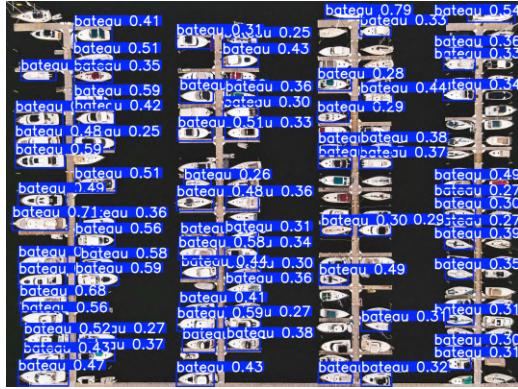


Figure 8: Détection automatique des bateaux par YOLOv8

2.3.4 Comparaison

On constate que l'approche par deep learning surpasse clairement l'approche probabiliste en termes de robustesse et de précision. Toutefois, chaque méthode présente ses propres avantages et limites.

Approche Probabiliste

- + Ne nécessite pas une base de données annotée
- + Efficace pour le comptage d'objets bien contrastés et peu nombreux
- + Interprétable : on peut visualiser l'énergie et comprendre les choix
- Moins adaptée aux scènes complexes ou bruitées
- N'est pas toujours précise sur des objets collés ou partiellement visibles
- Implémentation parfois compliquée des objets complexes (ellipse, recouvrement, etc.)

Approche par Deep Learning

- + Robuste aux variations de forme, couleur, fond, et éclairage
- + Détecte les objets même partiellement visibles ou superposés
- + Très efficace une fois entraînée (rapide, généralisable)
- Nécessite une base de données annotée (Pas évidente à trouver)
- Besoin de ressources de calcul (GPU pour l'entraînement)
- Moins interprétable : la décision dépend des couches internes du réseau

2.4 Conclusion

En conclusion, l'approche probabiliste permet de résoudre des problèmes de détection simples sans avoir besoin de données annotées, ce qui la rend intéressante dans un cadre exploratoire. Cependant, dès que les scènes deviennent plus complexes, l'approche par deep learning, bien que plus exigeante en ressources et en données, offre des résultats nettement plus robustes et généralisables. Selon les contraintes du projet (données disponibles, puissance de calcul, précision attendue), l'une ou l'autre méthode peut être privilégiée.

3 TP5

3.1 Introduction

Ce TP porte sur la restauration d'images, et plus spécifiquement sur la technique d'inpainting. L'objectif est de remplir des zones manquantes ou dégradées d'une image de manière visuellement cohérente. Le cas d'étude choisi est celui de la suppression d'un randonneur dans une image naturelle, en utilisant des méthodes de rapiéçage (*patch-based inpainting*). Cette approche se distingue des techniques classiques par diffusion et permet une meilleure reproduction des textures.

3.2 Rappel du sujet

3.2.1 Débruitage par variation totale

Au premier temps, l'objectif était de restaurer une image bruitée en minimisant une énergie contenant deux termes : un terme d'attache aux données, et un terme de régularisation favorisant la variation totale. Contrairement à la régularisation quadratique, la variation totale permet de mieux préserver les contours.



Figure 9: Débruitage d'image

3.2.2 Inpainting par diffusion

Dans ce second exercice, l'objectif est de remplir des zones manquantes dans une image, en supposant qu'un masque binaire D identifie les pixels dégradés. Contrairement au débruitage, le terme d'attache aux données ne s'applique que sur les pixels fiables $\Omega \setminus D$, tandis que la régularisation est étendue à toute l'image.

$$E_{\text{Inpainting}}(u) = \frac{1}{2} \iint_{\Omega \setminus D} [u(x, y) - u_0(x, y)]^2 dx dy + \lambda \iint_{\Omega} \sqrt{|\nabla u(x, y)|^2 + \varepsilon} dx dy \quad (1)$$

Cette méthode fonctionne bien pour des défauts fins (comme des rayures), mais peine à reconstituer des textures riches ou complexes. Elle est donc moins adaptée à la suppression d'objets entiers.



Figure 10: Inpating par variation totale

3.3 Pour aller plus loin

3.3.1 Inpainting par rapiéçage

Dans cette dernière partie, j'ai approfondi l'inpainting par rapiéçage. Contrairement à l'inpainting par diffusion, cette approche s'appuie sur la redondance de l'image pour reconstruire visuellement les zones manquantes avec plus de réalisme.

L'algorithme procède en choisissant aléatoirement un pixel sur la frontière du masque, puis en recherchant dans une fenêtre locale le patch le plus similaire au voisinage connu. Une fois ce patch trouvé, ses valeurs sont copiées dans la zone à compléter, ce qui permet de faire progresser le remplissage. Le masque est mis à jour et l'opération se répète jusqu'à ce que toute la zone soit comblée.

Le résultat obtenu est visuellement plus naturel que celui de l'exercice 2, en particulier pour les textures complexes comme la végétation. Cependant, cette méthode reste coûteuse en temps de calcul, et sensible au choix du patch. Des extensions comme l'introduction d'un ordre de priorité (Criminisi et al.) pourraient améliorer la cohérence structurelle du remplissage.



Figure 11: Inpating par rapiéçage

3.3.2 Ajout d'un ordre de priorité

Pour améliorer le réalisme du remplissage, j'ai modifié l'algorithme initial en introduisant un ordre de priorité inspiré de la méthode de Criminisi. Plutôt que de tirer un pixel au hasard sur la frontière, je sélectionne en priorité ceux dont le voisinage contient le plus de pixels connus. Cela permet de propager l'information de manière plus structurée, en reconstruisant d'abord les bords, puis en progressant vers les zones internes. Cette stratégie améliore significativement la cohérence des contours et évite les artefacts dans les zones texturées.



Figure 12: Ajout d'un ordre de priorité ; On observe une amélioration de qualité d'inpating

3.4 Conclusion

Ce TP a permis de comparer plusieurs approches de restauration, de la diffusion par variation totale à l'inpainting par rapiéçage. Si la diffusion donne de bons résultats pour les défauts fins, elle échoue dès que les textures deviennent complexes. L'inpainting par rapiéçage, surtout avec l'ajout d'un ordre de priorité, s'est révélé bien plus efficace visuellement, notamment dans les scènes contenant des éléments naturels comme des arbres. Ce choix stratégique permet de reconstruire les contours et la texture de manière plus cohérente, offrant un rendu final bien plus convaincant.

4 TP7

4.1 Introduction

L'objectif de ce TP est de modifier une image de manière réaliste, en intégrant des éléments provenant d'une autre image. Pour cela, différentes méthodes ont été expérimentées, allant de la plus simple à des approches plus élaborées faisant appel à des outils mathématiques issus de la physique et du traitement de signal.

4.2 Rappel du sujet

4.2.1 Collage naïf

L'approche naïve consiste à remplacer directement les pixels d'une zone de l'image cible par ceux d'une zone sélectionnée dans l'image source, après ajustement par une transformation affine. Le résultat est souvent peu réaliste : les bords sont visibles, les couleurs mal intégrées, et l'effet de collage est évident. Cette méthode est toutefois rapide à mettre en œuvre, ce qui en fait un bon point de départ pour expérimenter le photomontage.



Figure 13: Collage naïf

4.2.2 Collage par équation de Poisson

Pour obtenir un rendu plus naturel, nous avons implémenté une méthode fondée sur la résolution d'une équation de Poisson. Elle repose sur la construction d'un champ de gradient g qui combine

- le gradient de l'image cible à l'extérieur de la région insérée ;
- le gradient de l'image source à l'intérieur de cette région.

Resultat du photomontage

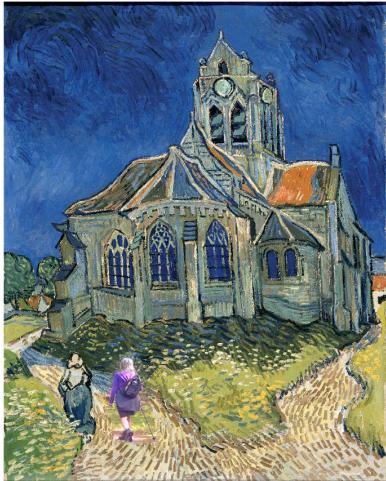


Figure 14: VanGogh



Figure 15: Yamal contre l'équipe de France

4.2.3 Décoloration partielle sans segmentation

Dans cette application, nous exploitons la capacité du modèle Poisson à fusionner deux images, même sans segmentation fine. Nous considérons ici une seule image, convertie dans l'espace LAB :

- l'image originale (canaux complets) joue le rôle de source ;
- la cible est construite en ne conservant que le canal de luminance L .

Nous appliquons ensuite la méthode de collage dans l'espace LAB, en conservant uniquement la chrominance dans une région sélectionnée, ce qui produit un effet de « couleur partielle » très esthétique.

Resultat du photomontage



Figure 16: Décoloration partielle sans segmentation précise

4.3 Pour aller plus loin

4.3.1 Autre technique de photomontage

En m'inspirant de l'article de Pérez, Gangnet et Blake (2003), j'ai implémenté une autre technique de photomontage illustrée dans la figure 8 de leur papier. Cette fois, il ne s'agissait plus

seulement d'incruster un objet, mais de transférer son relief et ses ombres dans un nouveau contexte.

Le Poisson Image Editing consiste à reconstruire une région d'image à partir d'un champ de gradients, en imposant une continuité parfaite avec les bords. Plutôt que de copier les pixels, on copie les variations locales (les gradients) de la source, puis on résout une équation de Poisson pour reconstituer les couleurs de manière fluide. Cette approche permet d'insérer, transformer ou fusionner des contenus visuellement de façon naturelle, sans coutures apparentes.

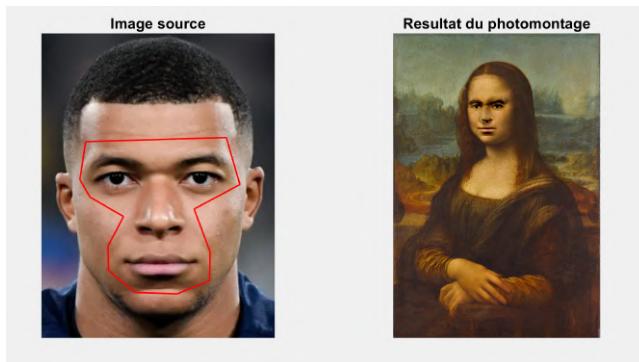


Figure 17: Photomontage réaliste par fusion de gradients

4.4 Conclusion

Ce TP a permis de découvrir la puissance des méthodes de photomontage guidées par les gradients. Si l'approche naïve montre rapidement ses limites visuelles, la résolution de l'équation de Poisson permet d'obtenir des incrustations beaucoup plus naturelles, en assurant une continuité d'intensité et de texture. Nous avons également exploré des applications originales comme la décoloration partielle ou l'insertion d'objets complexes, démontrant la flexibilité de ce cadre mathématique. Au-delà du simple collage, ces techniques ouvrent la voie à des manipulations d'image plus fines, alliant rigueur numérique et potentiel créatif.

5 TP8

5.1 Introduction

Ce TP porte sur la décomposition d'une image en deux composantes complémentaires : la structure (basses fréquences) et la texture (hautes fréquences). J'ai choisi, pour faire la liaison avec l'UE de **Traitement d'image** de détourner légèrement l'objectif initial : au lieu de simplement analyser la structure et la texture d'une même image, j'utilise la structure d'une première image et la texture d'une seconde pour effectuer un transfert de texture. Autrement dit, je reconstruis une image synthétique en combinant la forme générale d'une image source avec les motifs texturaux d'une autre.



Figure 18: Structure



Figure 19: Texture

5.2 Rappel du sujet

5.2.1 Décomposition par filtrage fréquentiel

Dans ce premier exercice, l'objectif est de séparer les basses et hautes fréquences d'une image à l'aide de la Transformée de Fourier discrète. La structure correspond aux basses fréquences (formes larges et douces), tandis que la texture correspond aux hautes fréquences (détails fins et motifs rapides). Pour cela, nous avons appliqué un filtre passe-bas doux sur le spectre de l'image.

Le spectre a été pondéré selon la formule :

$$\Phi(f_x, f_y) = \frac{1}{1 + \frac{f_x^2 + f_y^2}{\eta}},$$

où η est un paramètre de contrôle de la coupure fréquentielle. Cette pondération permet une séparation progressive (et non abrupte) entre les fréquences, ce qui évite des artefacts visuels comme des bords nets ou des oscillations (effet de Gibbs).

Le résultat donne une image de structure douce (contenant les grandes zones homogènes) et une image de texture (qui concentre les détails fins).



Figure 20: Structure



Figure 21: Texture



Figure 22: Fusion

5.2.2 Décomposition par variation totale (modèle ROF)

Dans ce second exercice, la séparation structure/texture repose sur un modèle variationnel proposé par Rudin, Osher et Fatemi (ROF). Ce modèle minimise une énergie composée de deux termes :

- un terme d'attache aux données, qui garantit que l'image restaurée reste proche de l'image originale ;
- un terme de régularisation, basé sur la variation totale, qui favorise les images « douces », à contours nets mais sans bruit ni textures fines.

Mathématiquement, l'énergie minimisée est donnée par :

$$E_{\text{ROF}}(u) = \frac{1}{2} \|u - u_0\|^2 + \lambda \int_{\Omega} \sqrt{|\nabla u|^2 + \varepsilon} dx dy$$

où u_0 est l'image d'origine, u la structure recherchée, λ le poids de régularisation, et ε un petit terme pour éviter les divisions par zéro.



Figure 23: Structure

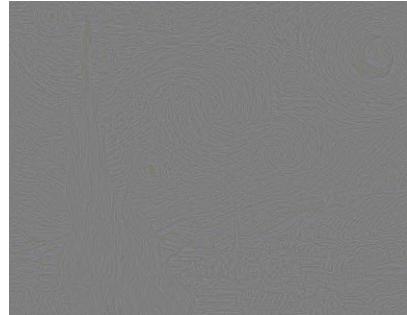


Figure 24: Texture



Figure 25: Fusion

Comparé à la méthode fréquentielle, le modèle ROF offre une séparation plus géométrique de la structure : les formes sont nettes et débruitées, mais certaines textures fines sont également éliminées.

5.3 Pour aller plus loin

5.3.1 Modèle TV-Hilbert

Pour aller au-delà de la simple séparation fréquentielle (exercice 1) ou variationnelle spatiale (exercice 2), j'ai exploré un modèle plus complet : le modèle TV-Hilbert. Ce dernier combine les avantages de la Transformée de Fourier et de la variation totale dans une formulation unifiée.

L'idée centrale est d'imposer une similarité entre les basses fréquences des images u (structure) et u_0 (image originale), tout en régularisant la structure avec une variation totale, comme dans le modèle ROF. Le terme de fidélité est formulé dans le domaine fréquentiel :

$$E(u) = \frac{1}{2} \left\| \mathcal{F}^{-1} \{ \Phi(f_x, f_y) \cdot [\mathcal{F}(u) - \mathcal{F}(u_0)] \} \right\|^2 + \mu \int_{\Omega} \sqrt{|\nabla u|^2 + \varepsilon} dx dy$$

où $\Phi(f_x, f_y)$ est un filtre passe-bas centré, μ est un poids de régularisation, et ε permet d'assurer la différentiabilité du terme TV. Cette formulation vise à ne conserver que les basses fréquences partagées avec u_0 , tout en supprimant les hautes fréquences non cohérentes, i.e. la texture. L'optimisation est réalisée par un schéma de descente de gradient, avec mise à jour itérative. :



Figure 26: Structure



Figure 27: Texture



Figure 28: Fusion)

Par rapport aux modèles précédents, le modèle TV-Hilbert permet un contrôle plus précis des fréquences conservées via le filtre Φ , tout en gardant une régularisation géométrique forte. Il constitue une approche hybride intéressante entre le filtrage fréquentiel et le traitement variationnel.

5.3.2 Transfert de style neuronal (Gatys et al.)

Pour approfondir l'idée de séparation entre structure et texture, et pour faire le lien avec l'UE de **Traitement d'image**, j'ai également expérimenté un algorithme très influent proposé par Gatys et al. (2016), appelé *Neural Style Transfer*. Contrairement aux approches précédentes qui reposent sur la Transformée de Fourier ou des modèles variationnels, cette méthode exploite les représentations hiérarchiques apprises par un réseau de neurones convolutif (CNN), typiquement VGG-19.

L'algorithme de transfert de style présenté par Gatys et al. repose sur l'utilisation d'un réseau de neurones convolutionnel pré-entraîné (comme VGG-19). L'idée principale est de générer une nouvelle image qui combine :

- le **contenu** d'une image (représenté par les activations d'un haut niveau du réseau pour l'image source),
- et le **style** d'une autre image (capturé par les corrélations entre les activations de couches convolutionnelles, i.e. les matrices de Gram).

La nouvelle image est obtenue par descente de gradient, en minimisant une fonction de perte qui mesure la distance aux représentations de contenu et de style. Cela permet d'obtenir un résultat visuellement impressionnant : une image qui « ressemble » à une peinture célèbre tout en conservant la structure de la photo initiale.

Concrètement, l'image de style et l'image de contenu sont analysées par le réseau pour extraire leurs caractéristiques respectives. Une image initiale aléatoire est ensuite optimisée

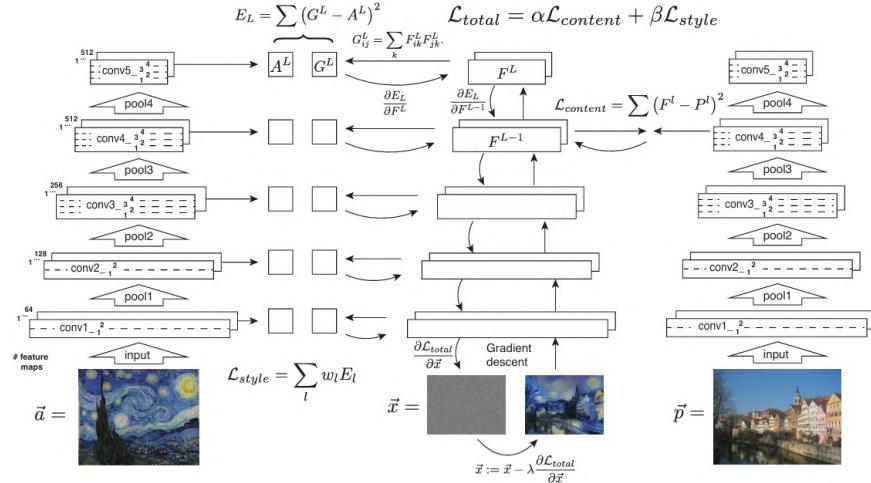


Figure 29: Réseau de Gatys

pour réduire à la fois la **perte de contenu** (différence entre les activations haut niveau de l'image générée et celles de l'image de contenu) et la **perte de style** (différence entre les matrices de Gram de l'image générée et de l'image de style). Le gradient de cette perte totale est rétropropagé pour ajuster les pixels de l'image générée jusqu'à convergence.



Figure 30: Fusion par style neuronal

Remarque : MATLAB propose déjà une implémentation fonctionnelle de cette méthode via sa *Deep Learning Toolbox*. J'ai ainsi simplement utilisé le script de démonstration fourni dans la documentation, sans en coder moi-même les détails.

5.4 Conclusion

Ce TP m'a permis d'explorer différentes méthodes de séparation structure/texture, du filtrage fréquentiel aux modèles variationnels avancés comme ROF ou TV-Hilbert. En allant plus loin, j'ai choisi de détourner l'objectif initial pour créer une synthèse d'image combinant la structure d'une photo et la texture d'une autre, dans un esprit proche du transfert de style.

L'approche classique (par fusion fréquentielle) offre un bon contrôle sur les composantes spatiales, mais reste limitée dans la qualité perçue. En intégrant l'approche de Gatys et al.,

j'ai pu constater la puissance des réseaux convolutionnels à modéliser séparément le contenu et le style, ouvrant la voie à des résultats plus esthétiques et nuancés. Cela montre l'intérêt de combiner méthodes analytiques classiques et techniques d'apprentissage profond pour des applications créatives en traitement d'image.

6 TP11

6.1 Introduction

Ce TP explore la problématique de la reconnaissance musicale à partir de courts extraits sonores, en s'inspirant de l'algorithme utilisé par l'application Shazam. Le but est de générer une « empreinte » sonore robuste, compacte et facilement comparable, pour identifier un morceau en quelques secondes.

6.2 Rappel du sujet

6.2.1 Détection des pics spectraux

Le signal est transformé en spectrogramme via une TFCT (fenêtre de Hann, taille 512, recouvrement 256). Un pic spectral est un maximum local dans une fenêtre (η_t, η_f) autour d'un point, au-dessus d'un seuil $\varepsilon = 1$ dB. Ces pics, en général situés dans les basses fréquences, forment la base de l'empreinte.

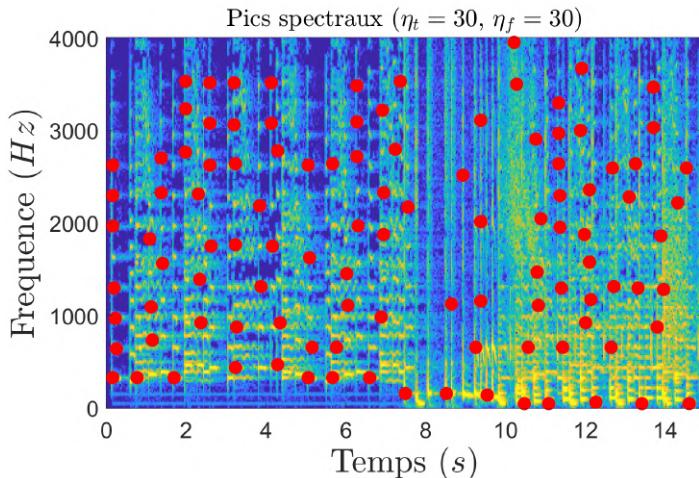


Figure 31: Pics spectraux extraits dans un morceau de musique

6.2.2 Appariement de pics

Chaque pic est associé à ses $nv = 5$ voisins les plus proches selon une contrainte temporelle δ_t et fréquentielle δ_f (toutes deux fixées à 90). Cela génère des paires (k_i, k_j, m_i, m_j) qui capturent une signature temporelle et fréquentielle stable du morceau.

6.2.3 Indexation

Les paires sont indexées par le triplet $(k_i, k_j, m_j - m_i)$ sur 32 bits. Chaque entrée de la base contient en plus le temps m_i et le numéro du morceau. Cela permet une recherche rapide des correspondances.

6.2.4 Reconnaissance simplifiée

L'extrait est transformé en paires de pics, puis comparé à la base. Le morceau contenant le plus de correspondances est sélectionné. Cette méthode simple atteint un taux de reconnaissance d'environ 91–92%, robuste jusqu'à un SNR de 10.

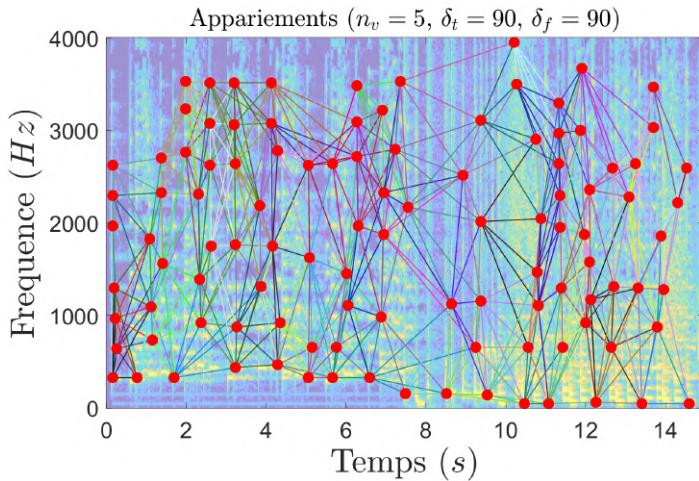


Figure 32: Pics spectraux extraits dans un morceau de musique

6.3 Pour aller plus loin

6.3.1 Reconnaissance musicale avancée

L'algorithme de reconnaissance simplifiée repose uniquement sur le comptage brut des correspondances entre empreintes. Pourtant, deux morceaux peuvent partager des transitions similaires (ex. $(k_i, k_j, \Delta t)$), sans pour autant correspondre à l'extrait. D'où l'idée d'introduire une contrainte de cohérence temporelle : si un extrait provient du temps t dans un morceau, alors tous ses pics devraient apparaître autour de $m_i + t$ dans la base.

Pour cela, nous avons modifié la fonction de recherche afin de stocker non seulement le numéro du morceau mais aussi le décalage temporel $d = m_i^{(\text{base})} - m_i^{(\text{extrait})}$ pour chaque paire. Cela permet d'identifier le morceau **et** l'instant probable où commence l'extrait.

Les résultats sont plus précis : au lieu de simplement voir quel morceau revient le plus souvent, on repère la plus forte accumulation temporelle cohérente dans un morceau donné.

1198
 Le morceau "Cosmo Sheldrake-Wriggle.ogg" a été correctement reconnu avec un décalage temporel de 12.70 secondes !
 >>

Ce raffinement permet de mieux discriminer entre morceaux similaires et d'améliorer la reconnaissance même dans les cas ambigus. Le taux de reconnaissance est passé de 91% à environ 98%, avec une stabilité renforcée face aux bruitages ou chevauchements musicaux.

6.3.2 Comparaison avec une approche par apprentissage profond

L'approche mise en œuvre dans ce TP repose sur une modélisation explicite des caractéristiques acoustiques stables (pics spectraux, appariements, indexation). Une alternative moderne consisterait à remplacer cette ingénierie par un réseau de neurones convolutif (CNN), entraîné directement sur des spectrogrammes pour apprendre à extraire des empreintes discriminantes.

Un tel réseau recevrait en entrée une image du sonagramme (ou un patch temporel) et produirait en sortie un vecteur compact, appelé *embedding*, censé être unique pour chaque morceau. La reconnaissance serait alors effectuée via une simple mesure de distance (cosine, euclidienne...) entre l'empreinte de l'extrait et celles contenues dans la base.

Avantages d'une telle approche :

- Capacité à capturer des motifs complexes, même en présence de bruit, de réverbération ou de variations harmoniques.

- Généralisation possible à d'autres tâches (identification d'instruments, transcription musicale, classification de genres...).
- Aucune hypothèse explicite sur les pics ou la structure fréquentielle.

Inconvénients :

- Besoin d'une base de données annotée massive pour entraîner le CNN (milliers d'heures de musique).
- Nécessité de GPU pour l'entraînement, voire pour l'inférence rapide.
- Moins interprétable : il est difficile d'expliquer pourquoi deux extraits sont jugés similaires.

Pourquoi Shazam reste une référence ? L'approche basée sur les pics spectraux reste inégalée en termes d'efficacité à grande échelle :

- Les identifiants binaires sont compacts et indexables en base de données.
- Les appariements tiennent en quelques dizaines de bits.
- La recherche dans la base se fait en temps quasi constant (via des tables de hachage).

6.4 Conclusion

Ainsi, bien que les réseaux de neurones offrent des performances remarquables dans des contextes supervisés, l'approche de Shazam incarne une solution robuste, légère, et rapide, particulièrement bien adaptée à la recherche à grande échelle avec des ressources limitées.

7 Conclusion générale

Dans ce rapport, nous avons exploré plusieurs techniques de traitement d'image et de signal, allant des méthodes classiques aux approches modernes basées sur le deep learning. Chaque TP a été l'occasion de découvrir une méthode différente et de comprendre ses points forts et ses limites.

En détection d'objets, les méthodes probabilistes sont faciles à interpréter mais moins performantes que YOLO sur des scènes complexes. En restauration d'images, le rapiéçage avec priorité a donné de très bons résultats pour des textures naturelles. Pour le photomontage, la résolution d'une équation de Poisson permet une fusion très fluide entre deux images. En décomposition d'image, nous avons comparé plusieurs approches, avant de tester le transfert de style neuronal, qui produit des résultats visuellement impressionnants. Enfin, pour la reconnaissance audio, l'algorithme de Shazam reste très efficace grâce à ses empreintes spectrales.

Ces expériences montrent que les méthodes classiques restent utiles et solides, mais que les réseaux de neurones permettent d'aller plus loin. L'avenir du traitement d'image semble se trouver dans l'association intelligente de ces deux mondes.

8 Bibliographie

References

- [1] Ultralytics. *YOLO - You Only Look Once*. <https://github.com/ultralytics.ultralytics>
- [2] MathWorks. *Neural Style Transfer Using Deep Learning*. <https://fr.mathworks.com/help/images/neural-style-transfer-using-deep-learning.html>
- [3] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge. *Image Style Transfer Using Convolutional Neural Networks*. CVPR 2016. <https://arxiv.org/abs/1508.06576>
- [4] Patrick Pérez, Michel Gangnet, Andrew Blake. *Poisson Image Editing*. ACM Transactions on Graphics (SIGGRAPH 2003). <https://doi.org/10.1145/882262.882269>