



Rapport  
Traitement des données audio-visuelles

SARHANE ABDELMOUHAIMEN

Département Sciences du Numérique - Deuxième année  
2022-2023

# Contents

<b>1 TP3 - Classification bayésienne</b>	<b>3</b>
1.1 Introduction à la classification bayésienne . . . . .	3
1.2 Segmentation supervisée . . . . .	3
1.3 Extension à la classification non supervisée . . . . .	5
1.4 Application à d'autres images et pistes d'amélioration . . . . .	5
1.5 Segmentation par regroupement de pixels . . . . .	7
<b>2 TP6 – Contours Actifs</b>	<b>9</b>
2.1 Introduction aux Contours Actifs . . . . .	9
2.2 Étude de l'Énergie Externe . . . . .	9
2.3 Implémentation du Contour Actif . . . . .	10
2.4 Diffusion vers les Contours . . . . .	11
2.4.1 Résultats de la Diffusion vers les Contours . . . . .	11
2.4.2 Analyse des Résultats . . . . .	12
2.5 Conclusion . . . . .	13
<b>3 TP7 : photomontage par collage</b>	<b>14</b>
3.1 Introduction . . . . .	14
3.2 Principe du photomontage par collage . . . . .	14
3.3 Implémentation . . . . .	15
3.4 Résultats . . . . .	15
3.5 Conclusion et perspectives . . . . .	16
<b>4 TP11 – Reconnaissance Musicale</b>	<b>18</b>
4.1 Introduction . . . . .	18
4.2 Calcul des Pics Spectraux . . . . .	18
4.3 Appariement des Pics Spectraux . . . . .	18
4.4 Indexation des Paires de Pics Spectraux . . . . .	19
4.5 Reconnaissance Musicale Simplifiée . . . . .	19
4.6 Reconnaissance Musicale Avancée . . . . .	19
4.7 Conclusion . . . . .	20
<b>5 TP12 – Séparation de Sources</b>	<b>21</b>
5.1 Introduction . . . . .	21
5.2 Séparation Harmonique/Percussive . . . . .	21
5.3 Décomposition d'un Sonagramme par NMF . . . . .	24
5.3.1 Implémentation de la Fonction NMF . . . . .	25
5.4 Méthodes par Apprentissage Profond . . . . .	26
5.4.1 Séparation de sources avec U-Net . . . . .	26
5.5 Conclusion . . . . .	27

# 1 TP3 - Classification bayésienne

## 1.1 Introduction à la classification bayésienne

La classification bayésienne est une méthode probabiliste permettant de segmenter une image en niveaux de gris  $x = (x_s)_{s \in S}$  en  $N$  classes. On suppose que les classes suivent des distributions gaussiennes de moyennes  $\mu_1, \dots, \mu_N$  et d'écart-types  $\sigma_1, \dots, \sigma_N$ . L'objectif est de trouver la configuration  $\hat{k} = (\hat{k}_s)_{s \in S}$  qui maximise la probabilité a posteriori  $p(K = k | X = x)$ . D'après le théorème de Bayes, on a :

$$p(K = k | X = x) \propto p(X = x | K = k)p(K = k) \quad (1)$$

En supposant l'indépendance des données, la vraisemblance s'écrit :

$$p(X = x | K = k) = \prod_{s \in S} \frac{1}{\sigma_{k_s} \sqrt{2\pi}} \exp - \frac{(x_s - \mu_{k_s})^2}{2\sigma_{k_s}^2} \quad (2)$$

La probabilité a priori de la configuration  $k$  est donnée par le modèle de Potts :

$$p(K = k) \propto \exp \left( -\beta \sum_{s,t \in C_2} [1 - \delta(k_s, k_t)] \right) \quad (3)$$

où  $C_2$  contient les paires  $s, t$  de pixels voisins. Finalement, on cherche à minimiser l'énergie  $U(k)$  :

$$U(k) = \frac{1}{2} \sum_{s \in S} \left[ \ln \sigma_{k_s}^2 + \frac{(x_s - \mu_{k_s})^2}{\sigma_{k_s}^2} \right] + \beta \sum_{s,t \in C_2} [1 - \delta(k_s, k_t)] \quad (4)$$

Pour cela, nous utilisons l'algorithme du recuit simulé qui fait décroître un paramètre  $T$  (la température) à chaque itération.

## 1.2 Segmentation supervisée

Nous avons implémenté en Matlab les différentes étapes de la classification bayésienne supervisée. La fonction `estimation_loi_normale` estime la moyenne et la variance de chaque classe à partir d'un échantillon sélectionné par l'utilisateur. La fonction `attache_aux_donnees` calcule, pour chaque pixel  $s$  et chaque classe  $k$ , le terme d'attache aux données. Enfin, la fonction `recuit_simule` implémente l'algorithme du recuit simulé.

Nous avons testé notre implémentation sur l'image (Figure 1). Les résultats montrent que la classification bayésienne supervisée permet d'obtenir une segmentation de bonne qualité (Figure 2) lorsque le nombre de classes est adéquat et que les échantillons sont bien sélectionnés.

La classification avec le maximum de vraisemblance présente un taux de pixels correctement classés d'environ 94% à cause de la présence de bruit, car elle ne prend pas en compte la position des pixels par rapport au voisinage (Figure 2a). Ce problème est résolu avec le maximum a priori, qui intègre cette information et améliore ainsi la qualité de la segmentation donnant ainsi un taux de pixels correctement classés égal à 99,86% (Figure 2b) avec  $T_0 = 1$ ,  $\alpha = 1$ ,  $\beta = 2$  et 50 itérations. En revanche, si le nombre de classes est incorrect ou si les échantillons ne sont pas représentatifs, la segmentation peut être de mauvaise qualité. De

plus, si la température initiale est nulle, l'énergie est forcée à décroître à chaque itération, ce qui peut mener à un minimum local.

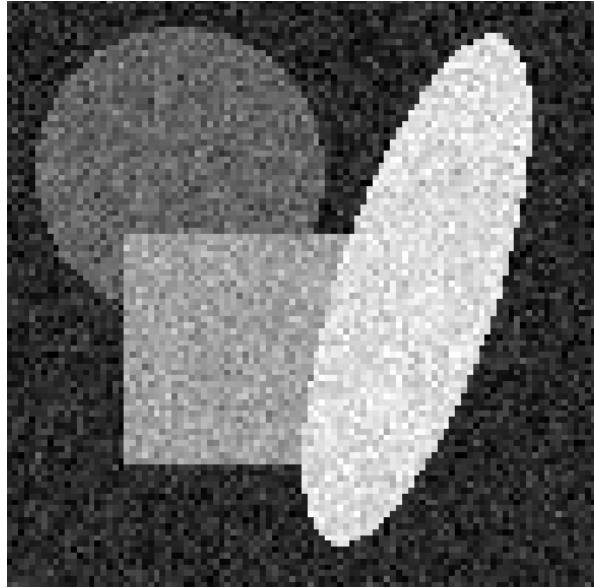
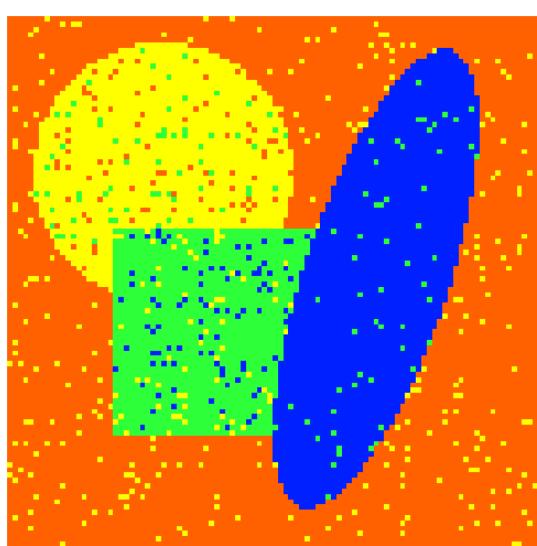
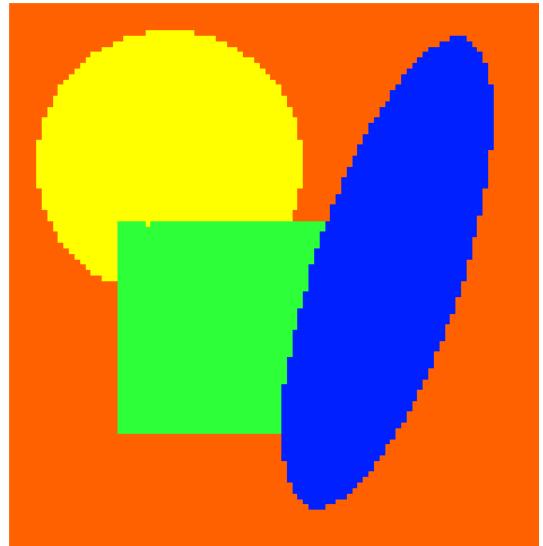


Figure 1: Image utilisée pour tester la classification bayésienne.



(a) Classification avec le maximum de vraisemblance



(b) Classification avec le maximum a priori avec  $T_0 = 1$ ,  $\alpha = 1$ ,  $\beta = 2$  et 50 itérations

Figure 2: Résultat de la segmentation supervisée avec 4 classes.

### 1.3 Extension à la classification non supervisée

Pour éviter la sélection manuelle des échantillons, nous avons implémenté une méthode de classification non supervisée qui estime les paramètres des  $N$  classes en cherchant un mélange de  $N$  gaussiennes coïncidant avec l'histogramme  $f(x)$  de l'image :

$$f(x) = \sum_{i=1}^N \frac{p_i}{\sigma_i \sqrt{2\pi}} \exp -\frac{(x - \mu_i)^2}{2\sigma_i^2}, \quad x \in 1, \dots, 255 \quad (5)$$

L'estimation des paramètres revient à résoudre un problème en moindres carrés linéaire vis-à-vis des poids  $p_i$ , mais non linéaire vis-à-vis des moyennes  $\mu_i$  et des écarts-types  $\sigma_i$ . Nous avons implémenté cette estimation en minimisant l'argument par tirages aléatoires. La fonction `estimation_poids` résout la partie linéaire du problème, c'est-à-dire l'estimation des poids  $(p_i)_{i \in E}$ . Les résultats montrent que la classification non supervisée permet d'atteindre un pourcentage de bonnes classifications 99,91% comparable à celui de la méthode supervisée (Figure 3), mais de manière entièrement automatique. En revanche, l'estimation des paramètres par tirages aléatoires est beaucoup plus lente.

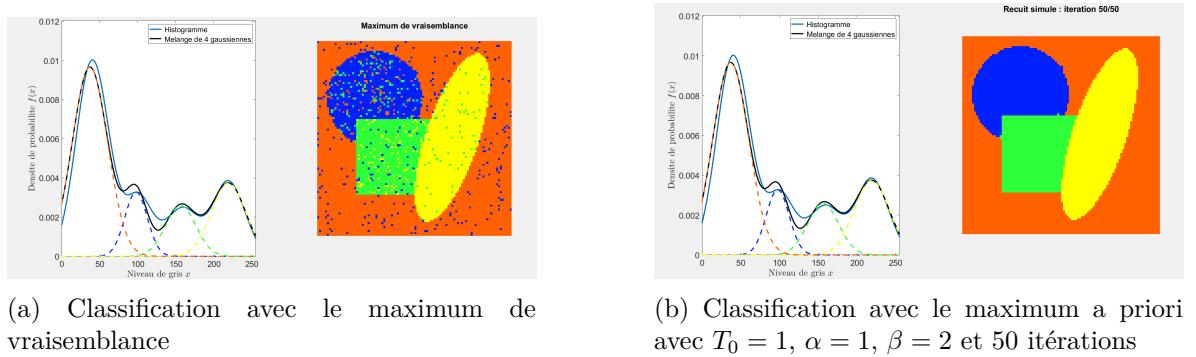
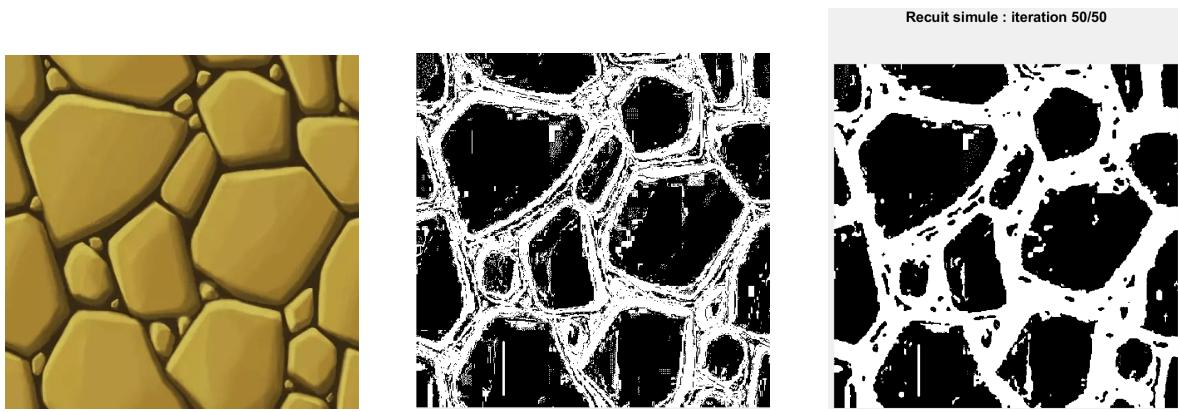


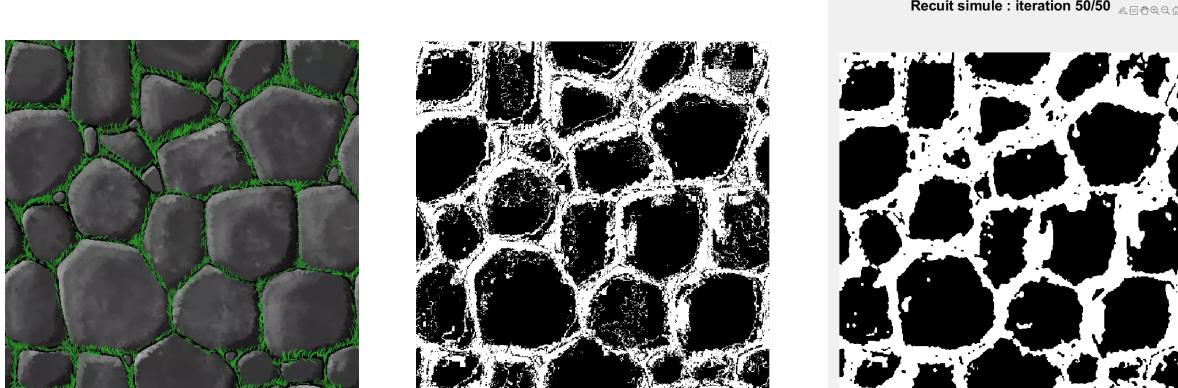
Figure 3: Résultat de la segmentation supervisé.

### 1.4 Application à d'autres images et pistes d'amélioration

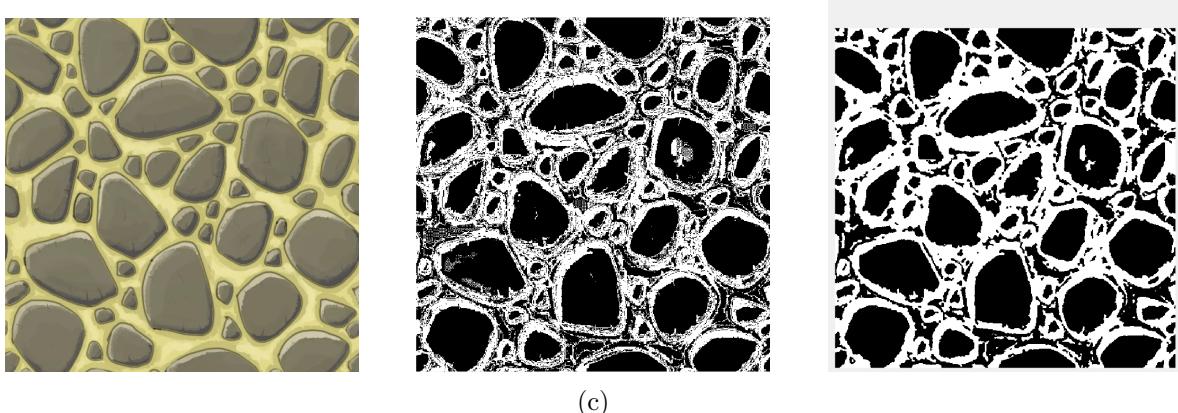
Nous avons également testé notre implémentation sur d'autres images en niveaux de gris (Figure 4). Les résultats confirment l'efficacité de la binarisation des images présentant deux classes bien distinctes. Cependant, la méthode montre ses limites lorsque les classes se chevauchent ou lorsque l'image présente des textures complexes.



(a) Image Originale, Binarisation avec le maximum de vraisemblance et Binarisation avec le maximum a priori avec  $T_0 = 1$ ,  $\alpha = 0.8$ ,  $\beta = 2$  et 50 itérations



(b)



(c)

Figure 4: Exemples d'images en niveaux de gris utilisées pour tester la classification bayésienne.

Plusieurs pistes d'amélioration sont envisageables :

- segmentation par regroupement de pixels (Voir 1.5)
- Optimiser l'estimation des paramètres dans le cas non supervisé, par exemple en utilisant l'algorithme EM.

- Combiner la classification bayésienne avec d'autres méthodes de segmentation, comme le SLIC ou les contours actifs.

### 1.5 Segmentation par regroupement de pixels

En nous inspirant de la méthode proposée dans l'exercice 3 du TP2, nous avons implémenté une segmentation par regroupement de pixels (clustering) en choisissant comme caractéristiques d'un pixel son niveau de gris et sa position dans l'image. Dans un premier temps, nous avons appliqué la méthode des k-moyennes (k-means) en utilisant comme caractéristiques pour chaque pixel son niveau de gris et sa position (coordonnées  $i$  et  $j$ ) dans l'image. Les résultats obtenus avec cette approche sont mitigés, avec un score de bonne classification de seulement 64% (Figure 5). Visuellement, on constate que le clustering n'a pas réussi à bien segmenter l'image.

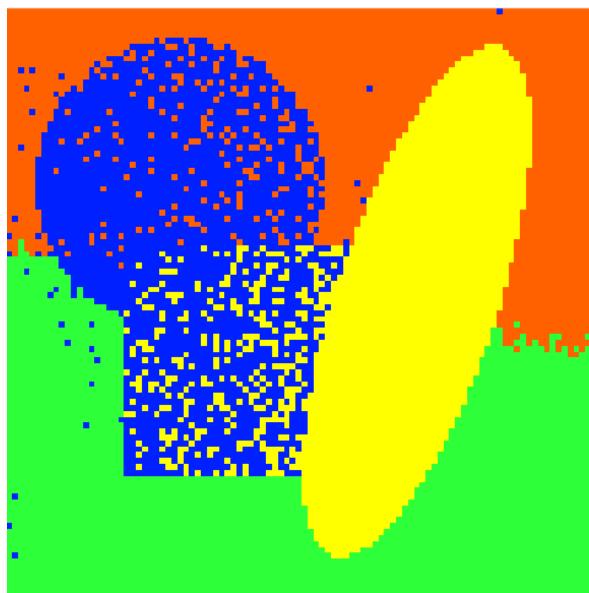


Figure 5: Résultat de la segmentation par k-means avec les caractéristiques de niveau de gris et de position.

Pour améliorer ces résultats, nous avons décidé de prendre en compte le voisinage de chaque pixel, en nous inspirant des méthodes utilisées dans les deux premiers exercices. Ainsi, nous avons choisi comme caractéristiques pour chaque pixel son niveau de gris ainsi que la moyenne des niveaux de gris de ses 8 voisins. Cette approche a permis d'obtenir un score de bonne classification de 98%, ce qui est nettement supérieur au score précédent. La Figure 6 montre le résultat de cette segmentation, où l'on peut constater visuellement la nette amélioration par rapport à la méthode précédente.



Figure 6: Résultat de la segmentation par k-means avec le niveau de gris et la moyenne du voisinage.

## 2 TP6 – Contours Actifs

### 2.1 Introduction aux Contours Actifs

Les contours actifs, également appelés snakes, sont des modèles de courbes déformables utilisés pour segmenter les objets visibles dans une image. Ces modèles sont très utilisés, notamment en imagerie médicale pour la segmentation de tumeurs.

Un contour actif est représenté par une courbe paramétrique  $P(s) = [x(s), y(s)]^T$  où  $s \in [0, 1]$  est une abscisse curviligne. Pour assurer que la courbe est fermée, on impose  $P(1) = P(0)$ . L'objectif est d'obtenir une courbe qui adhère aux contours d'une image de niveau de gris  $u$ , qui soit aussi courte que possible et qui oscille le moins possible. Ces propriétés sont modélisées par l'énergie interne et l'énergie externe.

L'énergie interne est définie comme suit :

$$E_{\text{int}}(P(s)) = \frac{\alpha}{2} \|\mathbf{P}'(s)\|^2 + \frac{\beta}{2} \|\mathbf{P}''(s)\|^2 \quad (6)$$

où  $\alpha > 0$  et  $\beta > 0$  sont des paramètres choisis par l'utilisateur. Quant à l'adéquation aux contours, elle est modélisée par l'énergie externe qui pénalise les faibles gradients du niveau de gris  $u$  :

$$E_{\text{ext}}(P(s)) = -\|\nabla u(P(s))\|^2 \quad (7)$$

L'équation d'Euler-Lagrange associée à l'énergie totale est :

$$\int_0^1 \left( E_{\text{ext}}(P(s)) + \frac{\alpha}{2} \|\mathbf{P}'(s)\|^2 + \frac{\beta}{2} \|\mathbf{P}''(s)\|^2 \right) ds \quad (8)$$

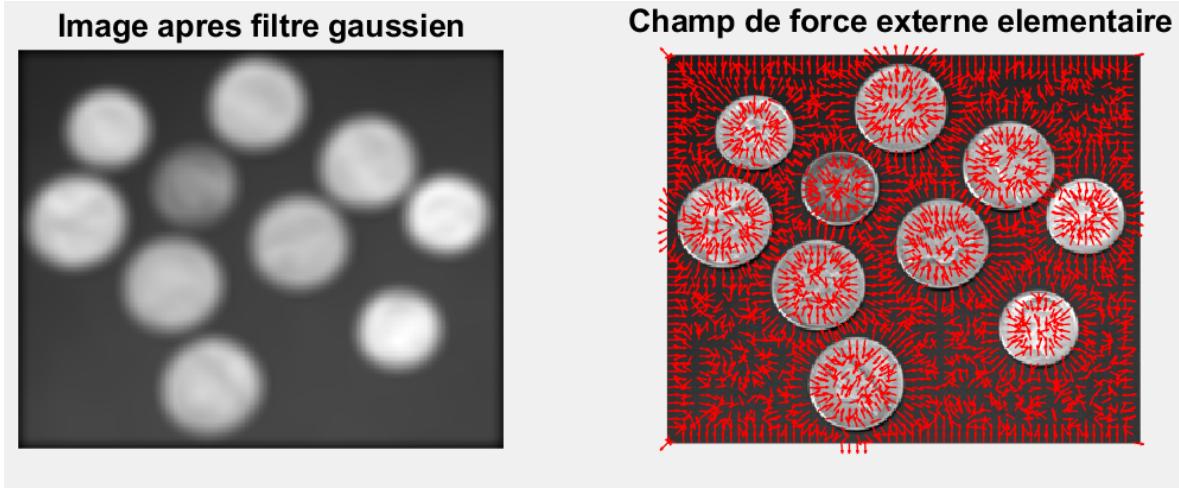
### 2.2 Étude de l'Énergie Externe

L'objectif de cette section est d'étudier l'énergie externe et son champ de force. Pour attirer davantage le contour actif vers les contours de l'image, nous appliquons un filtre de convolution gaussien à l'image avant de calculer son gradient :

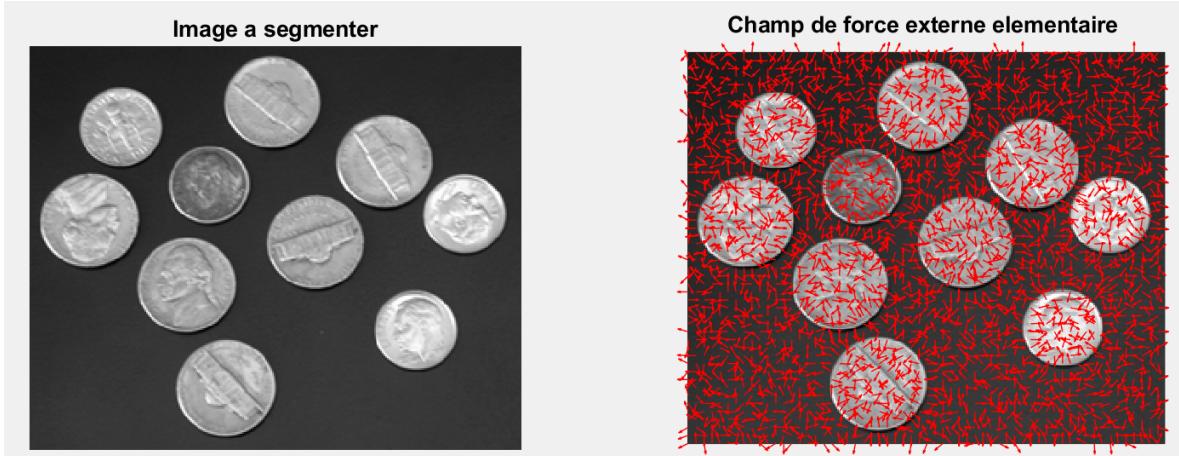
$$E_{\text{ext}}(P(s)) = -\|\nabla(G_{\sigma,T} * u)(P(s))\|^2 \quad (9)$$

où  $G_{\sigma,T}$  est un noyau gaussien d'écart-type  $\sigma$  et de taille  $T \times T$ .

Les figures suivantes montrent l'image après l'application du filtre gaussien et le champ de force externe élémentaire :



(a) Image après filtre gaussien



(b) Champ de force externe élémentaire

Figure 7: Étude de l'énergie externe avec filtre gaussien

### 2.3 Implémentation du Contour Actif

Pour implémenter le contour actif, nous devons calculer la matrice  $A$ , initialiser le contour actif, et faire évoluer le contour actif selon l'itération :

$$\begin{cases} x_{k+1} = Ax_k + B_x(x_k, y_k) \\ y_{k+1} = Ay_k + B_y(x_k, y_k) \end{cases} \quad (10)$$

La matrice  $A$  est définie par :

$$A = I + \gamma(\alpha D_2 - \beta D_2^T D_2) \quad (11)$$

où  $I$  est la matrice identité et  $D_2$  est l'approximation discrète de la dérivée seconde.

Les résultats de l'implémentation sur l'image coins.png sont illustrés ci-dessous :

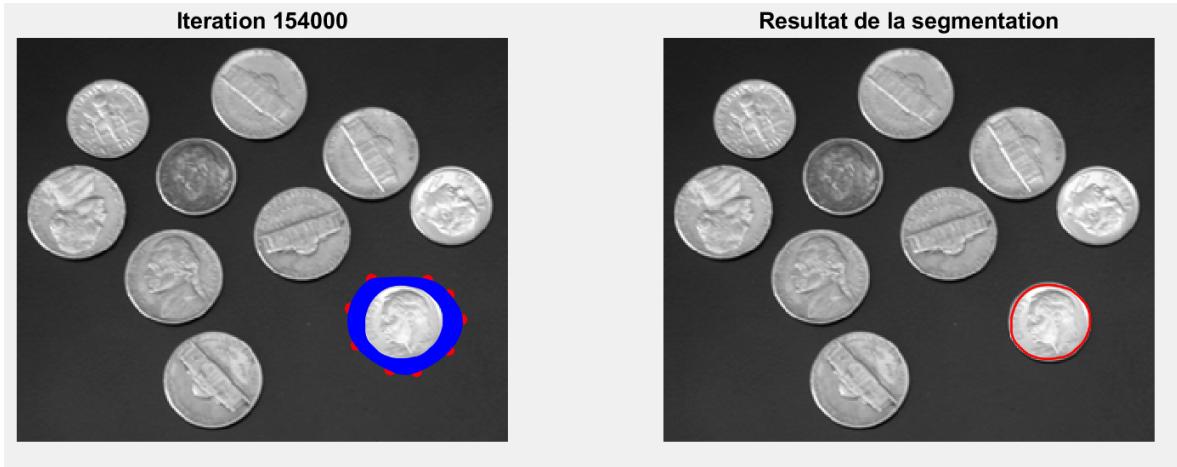


Figure 8: Implémentation du contour actif

## 2.4 Diffusion vers les Contours

Pour améliorer les résultats, nous utilisons le modèle de force externe de diffusion vers les contours (GVF, Gradient Vector Flow). Les équations d'Euler-Lagrange associées sont :

$$\begin{cases} \|\nabla E_{\text{ext}}^0(x, y)\|^2(F_x(x, y) - F_x^0(x, y)) - \mu_{\text{GVF}}\Delta F_x(x, y) = 0 \\ \|\nabla E_{\text{ext}}^0(x, y)\|^2(F_y(x, y) - F_y^0(x, y)) - \mu_{\text{GVF}}\Delta F_y(x, y) = 0 \end{cases} \quad (12)$$

Les résultats montrent que le champ de force GVF améliore significativement la segmentation :

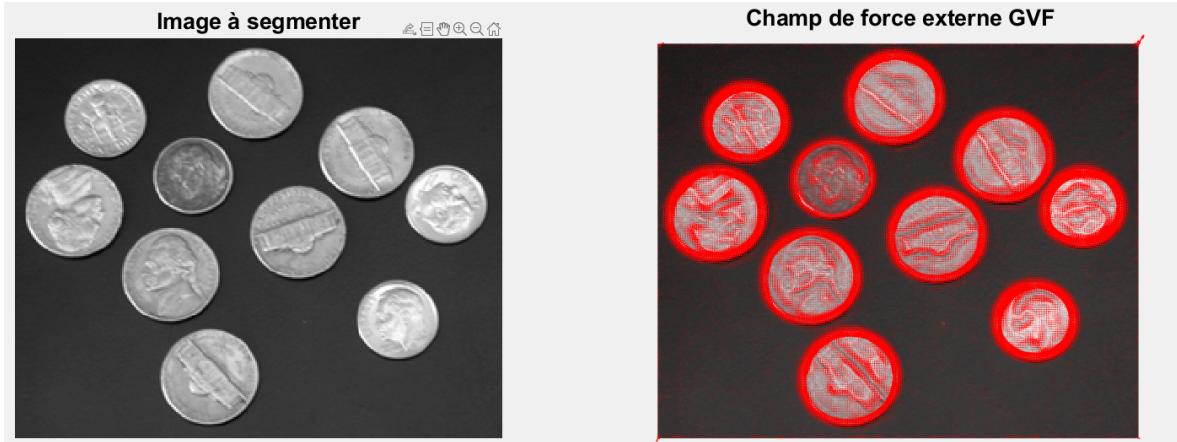
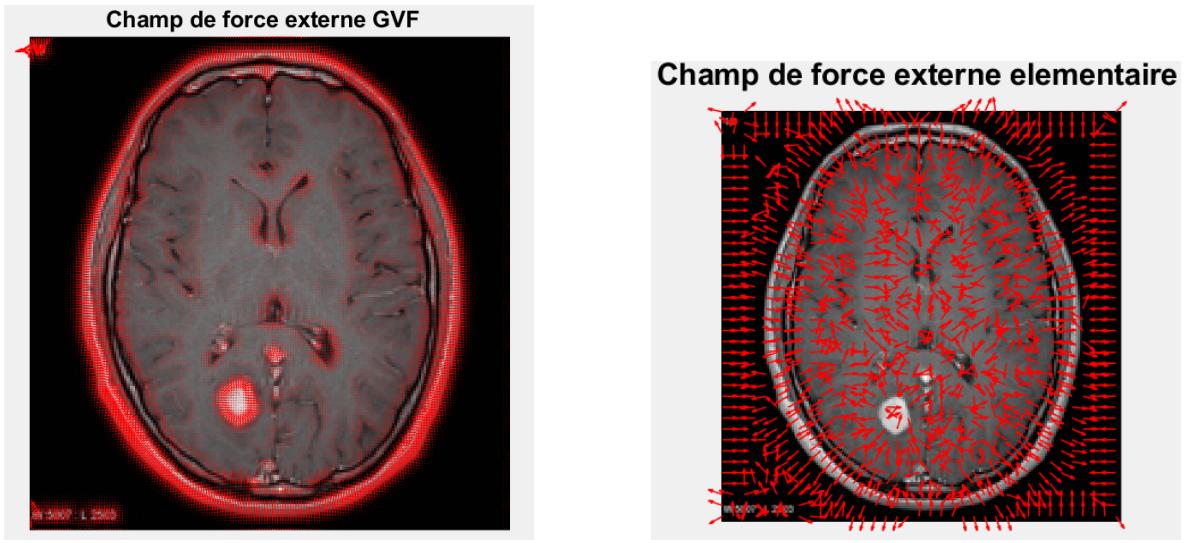


Figure 9: Diffusion vers les contours avec GVF

### 2.4.1 Résultats de la Diffusion vers les Contours

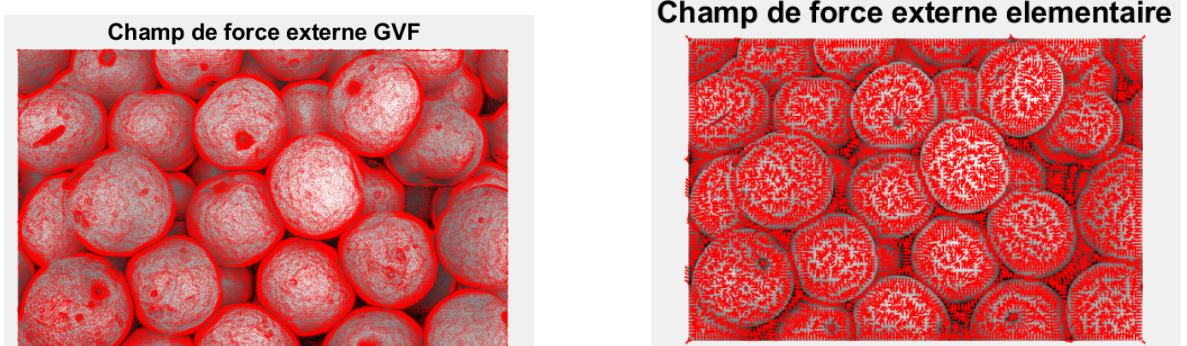
Les résultats montrent que le champ de force GVF améliore significativement la compréhension de l'image:



(a) Champ de force externe GVF - IRM

(b) Champ de force externe élémentaire - IRM

Figure 10: Comparaison des champs de force pour l'image IRM



(a) Champ de force externe GVF - Pears

(b) Champ de force externe élémentaire - Pears

Figure 11: Comparaison des champs de force pour l'image Pears

#### 2.4.2 Analyse des Résultats

L'analyse des champs de force montre que la diffusion vers les contours (GVF) permet d'obtenir des champs de force plus lisses et mieux orientés vers les contours réels des objets dans l'image.

Pour l'image IRM, le champ de force GVF (Figure 10a) montre une meilleure détection des contours internes et externes de la tumeur comparé au champ de force élémentaire (Figure 10b), qui présente des forces plus dispersées et moins ciblées.

Pour l'image Pears, le champ de force GVF (Figure 11a) permet une meilleure capture des formes circulaires des poires, tandis que le champ de force élémentaire (Figure 11b) montre des forces moins organisées et moins efficaces pour suivre les contours arrondis.

## 2.5 Conclusion

Dans ce TP, nous avons implémenté et testé l'algorithme des contours actifs pour la détection de contours dans les images. En appliquant un filtre gaussien et en utilisant le modèle de diffusion vers les contours GVF, nous avons amélioré la précision et la robustesse de la segmentation. Ces techniques sont puissantes et flexibles, applicables à une grande variété de problèmes en traitement d'images.

### 3 TP7 : photomontage par collage

#### 3.1 Introduction

L'objectif de ce TP est d'explorer différentes techniques de photomontage pour incruster une zone d'une image source dans une image cible de façon réaliste. Nous avons implémenté et comparé deux approches principales:

- Le collage naïf qui remplace directement les pixels de l'image cible par ceux de l'image source
- Le collage par résolution d'une équation de Poisson qui assure une meilleure continuité entre les images

Nous avons également testé une application originale permettant de décolorer partiellement une image sans avoir besoin de la segmenter précisément.

#### 3.2 Principe du photomontage par collage

Le photomontage par collage consiste à remplacer une zone d'une image cible  $c$  par des données issues d'une image source  $s$ . On définit pour cela:

- Un polygone  $p$  sélectionné dans l'image source  $s$
- Un rectangle  $r$  sélectionné dans l'image cible  $c$
- Une transformation affine  $t : s \rightarrow c$  telle que  $t(p) = r$  où  $e$  est le rectangle englobant de  $p$

L'approche naïve consiste à définir l'image résultat  $u$  telle que:

$$u(x, y) = \begin{cases} c(x, y) & \text{si } (x, y) \notin t(p) \\ s(t^{-1}(x, y)) & \text{si } (x, y) \in t(p) \end{cases}$$

Cependant, cette approche produit des résultats peu réalistes à cause des discontinuités aux bords de la zone collée (Figure 12). Pour obtenir un meilleur résultat, nous définissons un



Figure 12: Résultat du collage naïf. Des discontinuités apparaissent aux bords de la zone incrustée.

champ vectoriel  $g$  tel que:

$$g(x, y) = \begin{cases} \nabla c(x, y) & \text{si } (x, y) \in r \setminus t(p) \\ \nabla s(x, y) & \text{si } (x, y) \in t(p) \end{cases}$$

Nous cherchons ensuite l'image  $u$  minimisant l'écart avec ce champ vectoriel au sens des moindres carrés:

$$\min_{u: \mathbb{R}^2 \rightarrow \mathbb{R}} \int \int_{(x,y) \in r} |\nabla u(x, y) - g(x, y)|^2 dx dy$$

D'après l'équation d'Euler-Lagrange,  $u$  est solution de l'équation de Poisson suivante:

$$\Delta u(x, y) = \nabla \cdot g(x, y)$$

avec la condition aux limites  $u = c$  sur le bord de  $r$ .

### 3.3 Implémentation

Nous avons implémenté le collage par résolution de l'équation de Poisson de la façon suivante:

- Conversion des images en doubles
- Calcul de l'opérateur Laplacien discret  $A$  de taille  $N \times N$  où  $N$  est le nombre de pixels dans  $r$
- Modification des lignes de  $A$  et du second membre pour imposer la condition aux limites  $u = c$  sur le bord de  $r$
- Résolution du système linéaire  $Au^k = b^k$  pour chaque canal de couleur  $k \in R, V, B$
- Redimensionnement des résultats  $u^k$  à la taille de l'image

Un point clé est d'imposer la condition aux limites en modifiant la matrice  $A$  et le second membre  $b^k$ . Pour cela, nous avons utilisé la fonction `sparse` de Matlab afin de mettre à zéro toutes les lignes correspondant aux pixels du bord sauf sur la diagonale où nous avons mis des 1.

### 3.4 Résultats

La figure 13 montre un exemple de photomontage réalisé avec la méthode du collage par résolution de Poisson. L'incrustation de l'orque dans la montagne ou la randonneuse dans le tableau sont beaucoup plus naturelle qu'avec l'approche naïve (figure 12).

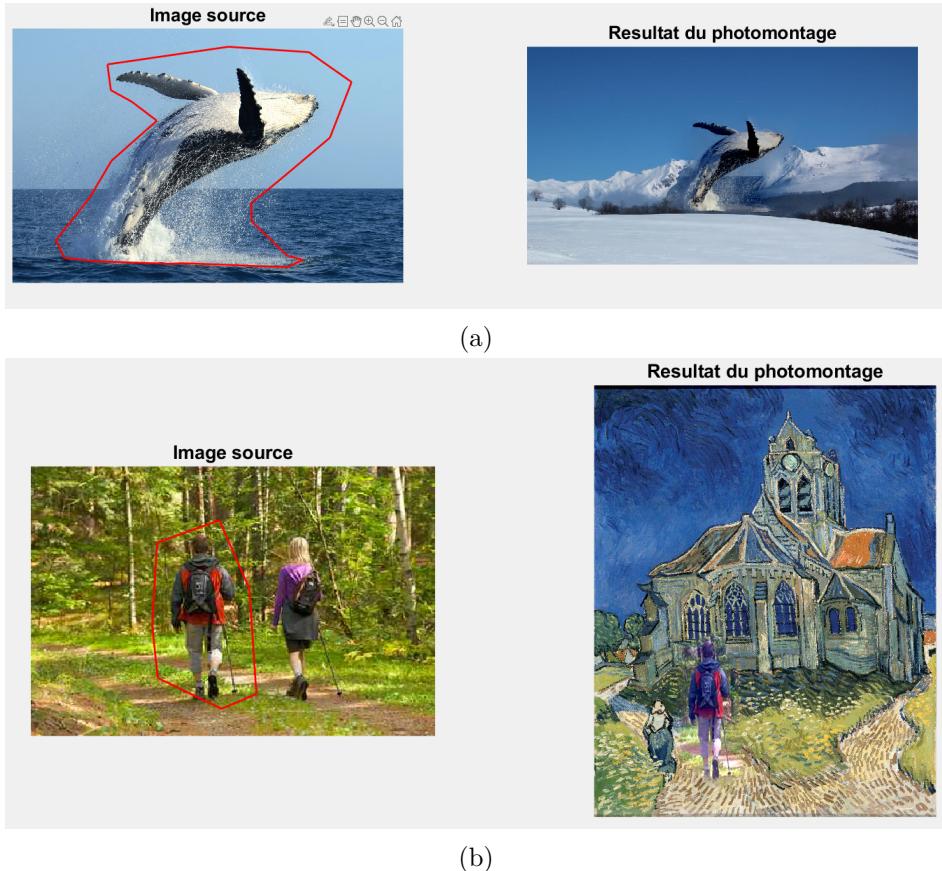


Figure 13: Photomontage réalisé avec la méthode du collage par résolution de Poisson.

Nous avons également testé une application originale de cette technique permettant de décolorer partiellement une image (figure 14). Pour cela, nous avons utilisé:

- L'image originale comme source  $s$
- Le canal de luminance  $L$  de l'image LAB comme cible  $c$

Il est intéressant de noter que cette approche ne nécessite pas de segmenter précisément la partie à garder en couleur. Une sélection grossière avec un polygone suffit.

### 3.5 Conclusion et perspectives

Dans ce TP, nous avons vu comment réaliser des photomontages de façon réaliste en utilisant une technique de collage par résolution d'une équation de Poisson. Cette approche permet d'éviter les discontinuités visibles de l'approche naïve en assurant une meilleure continuité du champ de gradient. Nous avons implémenté la méthode sous Matlab en posant un problème de Poisson avec conditions aux limites que nous avons résolu par une méthode directe après discrétisation. Nous avons montré deux applications :

- L'incrustation réaliste d'un objet extrait d'une image dans une autre image
- La décoloration partielle d'une image sans nécessiter de segmentation fine



Figure 14: Exemple de décoloration partielle d'une image obtenue sans segmentation fine, en utilisant le canal de luminance L de l'image LAB comme cible.

Pour aller plus loin, il serait intéressant de s'inspirer d'autres techniques de photomontage présentées dans l'article de Pérez, Gangnet et Blake datant de 2003, par exemple :

- **Échange de caractéristiques** : Le clonage sans couture permet de remplacer facilement certaines caractéristiques d'un objet par d'autres caractéristiques alternatives, comme le remplacement de textures.
- **Transfert monochrome** : Dans certains cas, le transfert de couleur intégral peut ne pas être souhaitable, notamment pour le transfert de texture. Convertir l'image source en monochrome avant le clonage sans couture peut résoudre ce problème.
- **Insertion d'objets avec des trous** : Le clonage sans couture classique peut être inefficace pour les objets avec des trous, mais une approche mixte de clonage sans couture basée sur une sélection lâche peut être efficace.
- **Insertion d'objets transparents** : Le clonage sans couture mixte facilite le transfert d'objets partiellement transparents, en utilisant un mélange non linéaire de champs de gradients pour sélectionner la structure la plus pertinente entre l'image source et la destination.
- **Édition de sélection** : Cette section détaille diverses techniques, telles que le lissage de texture, les changements d'éclairage sélectifs, les modifications de couleur de l'arrière-plan ou du premier plan, et le carrelage sans couture. Ces effets reposent sur des modifications non linéaires de la carte de gradient de l'image d'origine dans une région sélectionnée.

## 4 TP11 – Reconnaissance Musicale

### 4.1 Introduction

La reconnaissance musicale nécessite de définir une "empreinte" sonore qui caractérise chaque enregistrement musical sans ambiguïté. Cette empreinte doit être spécifique, robuste aux transformations simples, de taille réduite, facile à calculer et à comparer. Shazam utilise les pics spectraux (maxima locaux du spectrogramme) pour créer une empreinte sonore propre à chaque enregistrement musical.

### 4.2 Calcul des Pics Spectraux

L'idée est d'utiliser les pics spectraux pour obtenir une empreinte sonore. Un point  $(m_0, k_0)$  est considéré comme un pic spectral si, pour tous  $m \in [m_0 - \frac{\eta_t}{2}, m_0 + \frac{\eta_t}{2}]$  et  $k \in [k_0 - \frac{\eta_f}{2}, k_0 + \frac{\eta_f}{2}]$ , on a :

$$S_{dB}(m_0, k_0) \geq \max\{S_{dB}(m, k), \epsilon\} \quad (13)$$

où  $\eta_t$  et  $\eta_f$  définissent la fenêtre de recherche, et  $\epsilon$  est un seuil.

La figure 15 montre les pics spectraux détectés dans un morceau de musique.

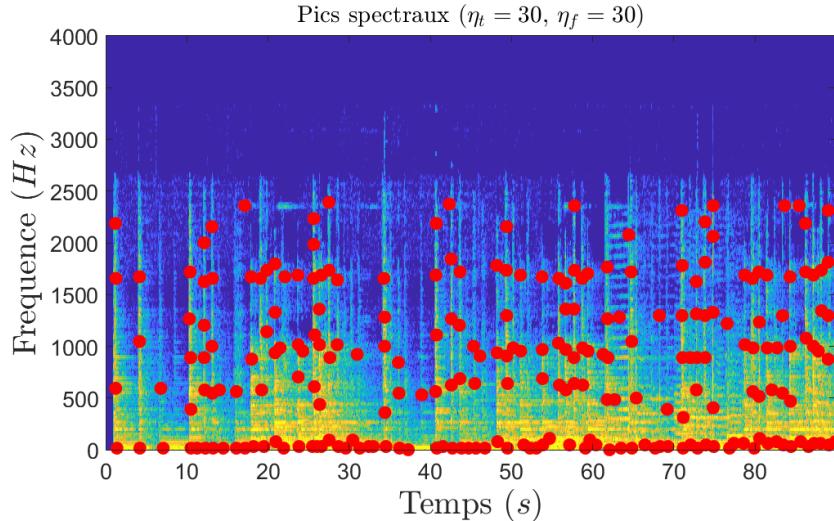


Figure 15: Pics spectraux détectés dans les 15 premières secondes d'un morceau de musique ( $\eta_t = 30$ ,  $\eta_f = 30$ ).

### 4.3 Appariement des Pics Spectraux

Les pics spectraux obtenus doivent être appariés pour permettre une bonne indexation. Chaque pic est apparié avec les  $n_v$  pics les plus proches temporellement et fréquentiellement, formant ainsi des paires de pics spectraux voisins.

Les critères pour l'appariement sont les suivants :

- $0 < m_j - m_i \leq \delta_t$
- $|k_i - k_j| \leq \delta_f$

La figure 16 montre les appariements des pics spectraux pour un extrait musical.

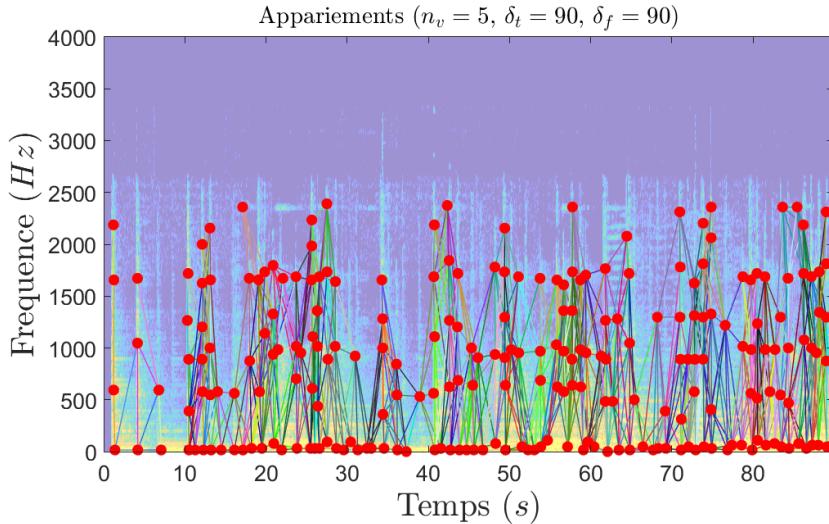


Figure 16: Appariements des pics spectraux ( $n_v = 5, \delta_t = 90, \delta_f = 90$ ).

#### 4.4 Indexation des Paires de Pics Spectraux

Chaque paire  $(k_i, k_j, m_i, m_j)$  est indexée par le triplet  $(k_i, k_j, m_j - m_i)$ . Cette indexation permet une recherche rapide et efficace dans la base de données.

La table 1 montre la structure de la base de données utilisée pour la reconnaissance musicale.

$(k_i, k_j, m_j - m_i)$	$m_i$	num
(23, 45, 52)	6	1
:	:	:
(174, 154, 84)	498	73

Table 1: Structure de la base de données utilisée pour la reconnaissance musicale.

#### 4.5 Reconnaissance Musicale Simplifiée

Pour reconnaître un extrait, on calcule son empreinte et on la compare avec celles de la base de données. Chaque paire de pics spectraux de l'extrait est comparée aux paires indexées dans la base pour identifier le morceau présentant le plus grand nombre de correspondances. La méthode simplifiée atteint un taux de reconnaissance de 91%. En testant la robustesse du système, nous constatons que le pourcentage de bonnes reconnaissances reste constant jusqu'à un SNR de 10. Pour des valeurs de SNR supérieures, la reconnaissance diminue significativement, atteignant 80% à un SNR de 13.

#### 4.6 Reconnaissance Musicale Avancée

L'algorithme de reconnaissance musicale simplifié ne tient pas compte de la cohérence temporelle entre les instants d'apparition des paires de pics dans l'extrait et dans le morceau complet.

Pour améliorer la reconnaissance, nous devons imposer cette cohérence.

Chaque paire de pics  $((m_i, k_i), (m_j, k_j))$  de l'extrait est recherchée dans la base de données, mais nous enregistrons également la différence temporelle entre leur apparition dans le morceau complet et dans l'extrait. Cette différence temporelle nous aide à identifier l'origine potentielle de l'extrait par rapport au début du morceau intégral.

Pour un morceau reconnu, toutes les paires doivent théoriquement avoir une différence temporelle égale. Cependant, nous observons des différences entre les correspondances. Cela s'explique par le fait que les identifiants des paires sont utilisés pour chercher les moments où une fréquence  $k_i$  passe à une fréquence  $k_j$  en  $m_j - m_i$  secondes, ce qui peut se produire à plusieurs instants dans le morceau. Il faut donc prendre le délai minimum.

En mode discret, pour des signaux échantillonnés à la fréquence  $f_{ech}$ , avec un décalage  $D$  entre deux positions successives de la fenêtre, on peut retrouver le délai en secondes avec la formule :

$$\text{tcoef}(m) := \frac{mD}{f_{ech}} \quad (14)$$

#### 4.7 Conclusion

Dans ce TP, nous avons implémenté et testé des algorithmes de reconnaissance musicale basés sur les pics spectraux et leurs appariements. En intégrant la méthode de recherche avancée prenant en compte la cohérence temporelle, nous avons amélioré la précision et la robustesse de la reconnaissance musicale, atteignant un taux de reconnaissance de 98%. Ces techniques peuvent être étendues et adaptées pour améliorer la scalabilité dans grands problèmes de reconnaissance de motifs dans les données.

## 5 TP12 – Séparation de Sources

### 5.1 Introduction

La séparation de sources musicales consiste à isoler  $N$  sources à partir de  $M$  enregistrements. Cette tâche est essentielle pour des applications telles que le re-mixage, le débruitage, la transcription musicale, le changement de tempo ou de hauteur, et l'analyse des paroles. Cependant, la séparation de sources reste un défi en raison des corrélations complexes entre les sources et des effets non linéaires lors du mixage.

### 5.2 Séparation Harmonique/Percussive

La séparation harmonique/percussive est une méthode simple pour séparer les composantes harmoniques (voix, instruments mélodiques) des composantes percussives (batterie) d'un morceau de musique. Cette méthode repose sur les observations suivantes :

- Le contenu spectral d'un son harmonique varie peu au cours du temps, apparaissant comme des lignes horizontales dans le sonogramme.
- Un son percussif est bref mais riche en fréquences, apparaissant comme des lignes verticales dans le sonogramme.

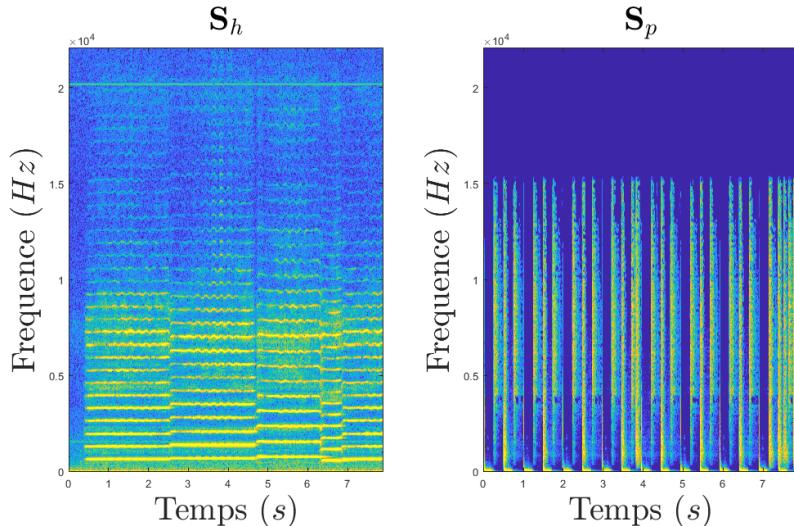


Figure 17: Sonogramme  $S$  d'un morceau de violon et de batterie.

En mélangeant ces deux sons, le sonogramme suivant est obtenu :

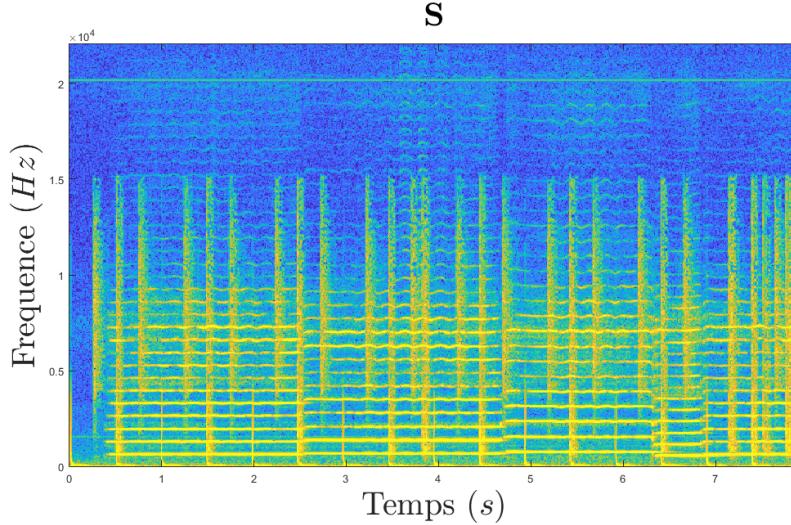


Figure 18: Sonogramme mélangé avant la décomposition par NMF.

Pour séparer ces composantes, nous appliquons un filtrage médian horizontal pour renforcer les composantes harmoniques et un filtrage médian vertical pour les composantes percussives. Cela donne le sonogramme suivant :

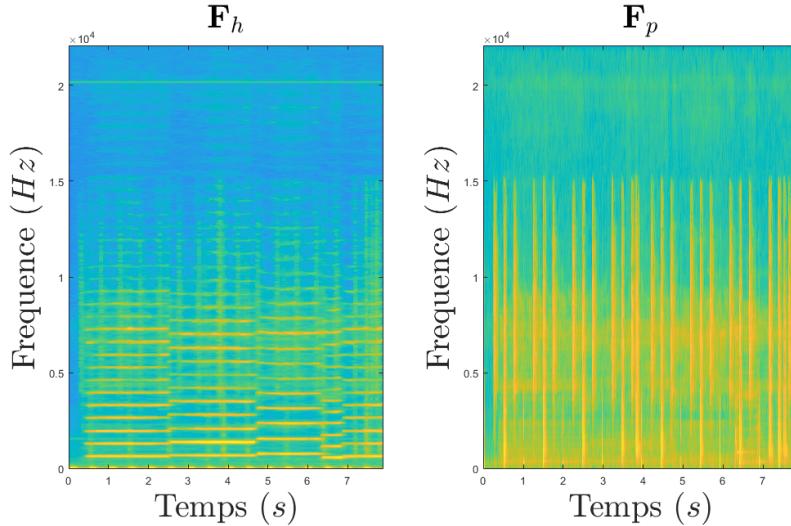


Figure 19: Filtrage médian du sonogramme mélangé.

Il est ensuite possible de créer des masques correspondant aux parties harmoniques et percussives. Pour cela, on peut créer deux masques binaires, un élément du sonogramme étant soit harmonique soit percussif :

$$M_h = (F_h \geq F_p); \quad M_p = (F_h < F_p)$$

mais on peut également créer des masques « doux » ( $\odot$  désigne la division élément par élément) :

$$M_h = \frac{F_h}{F_h + F_p}; \quad M_p = \frac{F_p}{F_h + F_p}$$

En appliquant cette paire de masques à la TFCT  $Y$  du signal d'origine ( $\odot$  désigne le produit élément par élément), on obtient bien une décomposition  $Y = \hat{Y}_h + \hat{Y}_p$ , puisque  $M_h$  et  $M_p$  sont complémentaires :

$$\hat{Y}_h = M_h \odot Y; \quad \hat{Y}_p = M_p \odot Y$$

Le résultat de cette décomposition peut être écouté en inversant la TFCT.

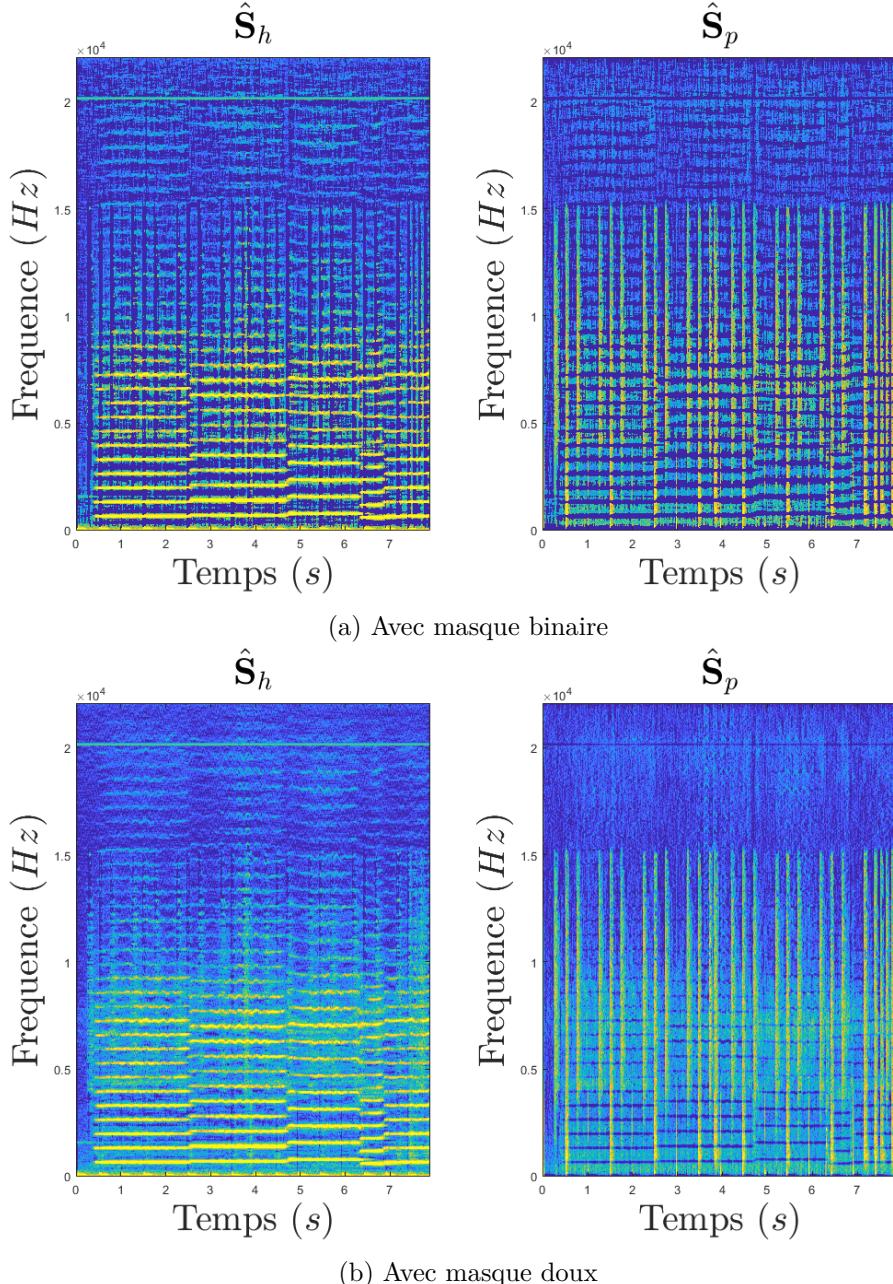


Figure 20: Filtrage médian du sonogramme

Les masques obtenus permettent de créer une décomposition  $Y = \hat{Y}_h + \hat{Y}_p$ , où  $\hat{Y}_h$  est la partie harmonique et  $\hat{Y}_p$  est la partie percussive.

- Partie harmonique avec masque doux
- Partie percussive avec masque doux
- Partie harmonique avec masque binaire
- Partie percussive avec masque binaire

Bien que les deux types de masques (doux et binaires) permettent de séparer les composantes harmoniques et percussives, il est difficile d'entendre une différence significative entre les deux méthodes. Cela s'explique par le fait que les masques binaires et doux partagent une similarité fondamentale dans leur approche : ils isolent les mêmes régions du spectrogramme mais de manière légèrement différente.

Les masques binaires attribuent chaque élément du spectrogramme exclusivement à la composante harmonique ou percussive. Les masques doux, en revanche, attribuent une probabilité d'appartenance à chaque composante. Cependant, pour les signaux audio utilisés ici, cette différence n'est pas perceptible à l'oreille humaine.

### 5.3 Décomposition d'un Sonagramme par NMF

La factorisation en matrices non négatives (NMF) vise à approcher une matrice  $S$  par le produit de deux matrices  $D$  et  $A$  à coefficients non négatifs :

$$S \approx DA \quad (15)$$

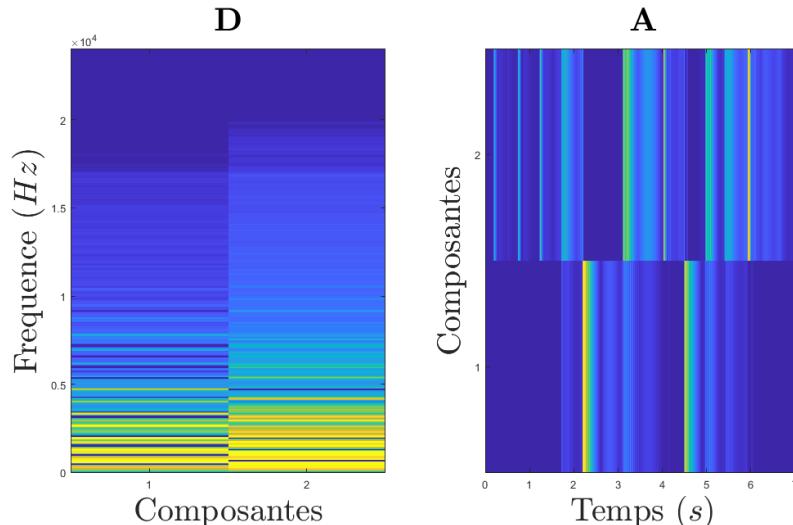


Figure 21: Décomposition NMF du sonogramme avec  $R = 2$ .

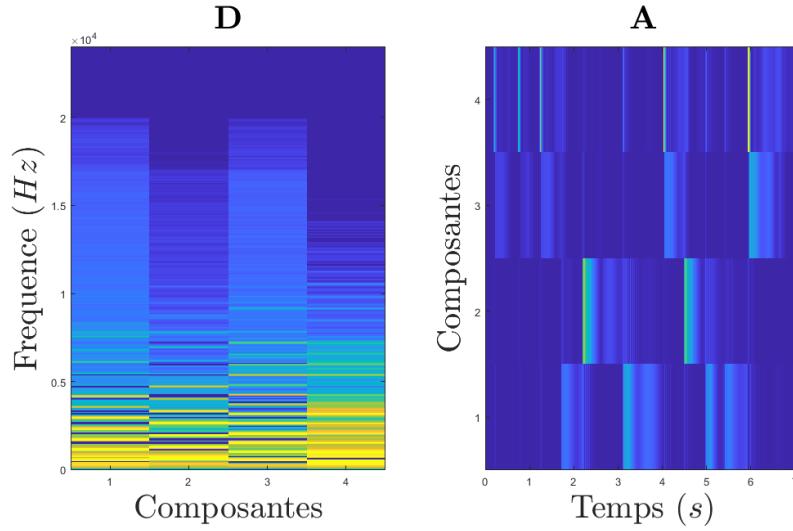


Figure 22: Décomposition NMF du sonogramme avec  $R = 4$ .

La méthode NMF permet de séparer les composantes harmoniques et percussives en utilisant les vecteurs de base de  $D$  et les activations temporelles de  $A$ .

### 5.3.1 Implémentation de la Fonction NMF

Nous avons implémenté la fonction `nmf` pour réaliser cette décomposition, en initialisant  $D$  et  $A$  avec des valeurs positives aléatoires. Les masques  $M_r$  sont créés pour chaque composante séparée :

$$M_r = \frac{D(:, r)A(r, :)}{DA} \quad (16)$$

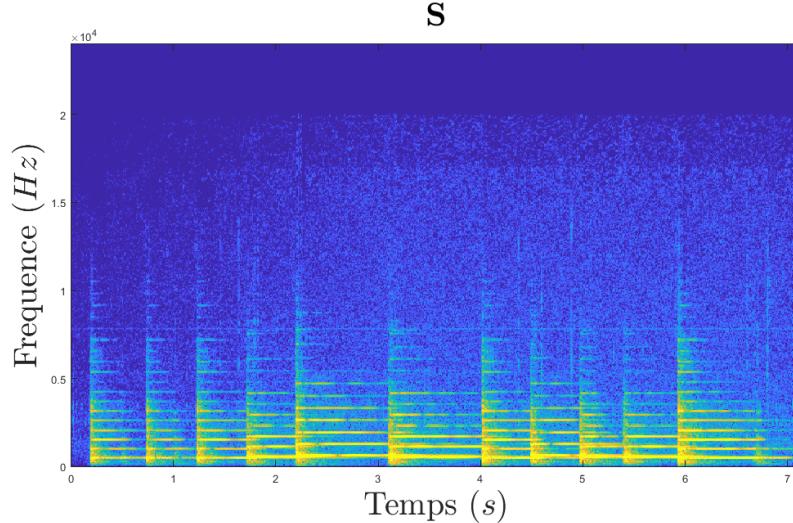


Figure 23: Sonogramme mélangé avant la décomposition par NMF.

Les résultats de cette décomposition montrent une séparation efficace des composantes

harmoniques et percussives.

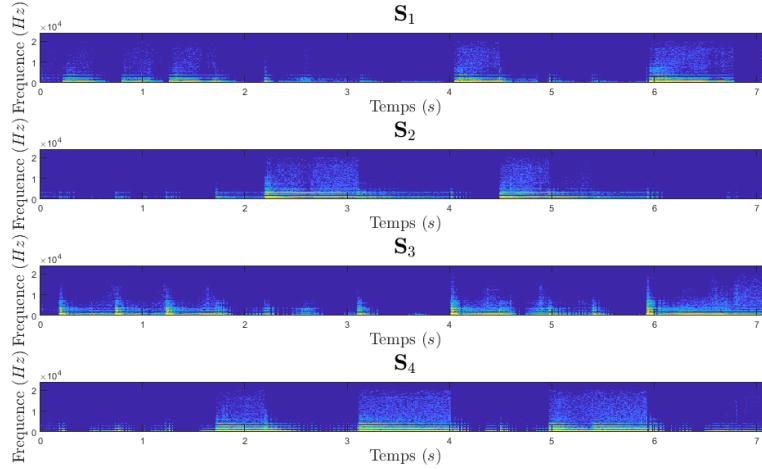


Figure 24: Sonogrammes des composantes séparées pour  $R = 4$ .

En utilisant les vecteurs de base  $D$  et les activations temporelles  $A$ , nous avons pu isoler différentes composantes du sonogramme initial, ce qui permet une analyse plus fine des éléments constitutifs du signal audio.

## 5.4 Méthodes par Apprentissage Profond

Les méthodes par apprentissage profond, telles que les architectures U-Net, sont actuellement l'état de l'art pour la séparation de sources musicales. Ces réseaux utilisent des convolutions 2D pour capturer les caractéristiques complexes des signaux musicaux.

### 5.4.1 Séparation de sources avec U-Net

Nous nous concentrerons ici sur la séparation du vocal et de la musique de fond, une tâche plus simple que la séparation en quatre sources distinctes.

Le réseau de neurones U-Net est utilisé pour cette tâche, car il est bien adapté aux problèmes de séparation de sources grâce à sa capacité à capturer à la fois des caractéristiques locales et globales du signal. La structure du U-Net est la suivante :

- **Conv2D** : Filtres = 16, Stride = 2, Kernel = 5
- **Conv2DTranspose** : Filtres = 256, Stride = 2, Kernel = 5

Les résultats obtenus montrent une séparation efficace des voix et de la musique de fond. Les fichiers audio suivants présentent les résultats de la séparation :

- [Voix d'origine](#)
- [Musique de fond d'origine](#)
- [Voix prédictes](#)

- [Musique de fond prédictive](#)

Les audios ci-dessus montrent que le modèle U-Net a bien réussi à séparer les composantes vocales et musicales. Les fichiers originaux et prédictifs peuvent être comparés pour évaluer la qualité de la séparation.

## 5.5 Conclusion

Ce TP a permis d'explorer diverses techniques de séparation de sources musicales, de la séparation harmonique/percussive aux méthodes de factorisation en matrices non négatives et aux approches par apprentissage profond. Les résultats montrent que chaque méthode a ses avantages et ses limites, et qu'une combinaison de techniques pourrait offrir des performances optimales.