Gourmand Nicolas

Thomas Allan

Rapport Projet Outils Mathématiques

**Sommaire:**

1. **Transformée de Fourier directe discrète pour une image**
2. **Transformée de Fourier rapide discrète 1D**
3. **Transformée de Fourier rapide discrète 2D**
4. **Transformée de Fourier inverse rapide 1D**
5. **Transformée de Fourier inverse rapide 2D**
6. **Détecteur de contours**

**Transformée de Fourier directe discrète pour une image**

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquementUne image contenant texte, Police, ligne, algèbre

Description générée automatiquementL’algorithme commence par prendre une image. Si cette image est en couleurs ( RGB ), alors on la convertit en niveaux de gris. On récupère ensuite les dimensions de l’image avec M et N, où M est le nombre de lignes et N le nombre de colonnes. On initialise ensuite une matrice *dft\_result* de zéros de la même taille que l’image, puis on calcule manuellement la dft 2d :

Il y a ici 4 boucles for imbriquées : les deux premières itèrent sur chaque fréquences ( u et v ) de la matrice de sortie ( *dft\_result* ) = fréquences analysées dans le spectre de sortie (coefficients DFT), alors que les deux autres itèrent sur les coordonnées spatiales de l’image ( x et y ).

Pour chaque (u,v), on fait la somme pondérée des intensités des pixels de l’image à différentes positions spatiales.

On calcule ensuite la valeur DFT correspondante en sommant de produit de l’intensité de chaque pixel dans l’image avec le terme exponentiel complexe évalué à cette fréquence particulière. Image(x,y) correspond à l’intensité du pixel à la position (x,y).  
L’exponentielle correspond à la contribution de chaque pixel de l’image de base à une fréquence(u,v) dans le domaine fréquentiel. Précisons que dans la formule utilisée, on soustrait 1 à u,x,v et y, car Matlab utilise un système de coordonnées commençant à 1, et non 0 : il faut donc enlever 1 pour être à 0.

Le terme exponentiel complexe représente une sinusoïde dans le domaine spatial, qui oscille à la fréquence déterminée par (u,v) lors du déplacement sur l’image.

e(ix) est le nombre complexe qui décrit un point du cercle unité dans le plan complexe. Quand x est un multiple de 2PI, e(ix) vaut 1. On multiplie par -i pour effectuer un décalage de phase de 90 ° dans le sens trigonométrique dans le plan complexe. C’est important, car cela aide à représenter les informations de phase de la sinusoïde.

Une image contenant Police, symbole, texte, ligne

Description générée automatiquementCeci détermine la fréquence de la sinusoïde. En variant u et v, différentes fréquences en x et y sont échantillonnées dans l’image. On divise par M et N pour normaliser les fréquences pour les adapter à la plage de dimension de l’image ( = mise à l’échelle ) : garantit que les fréquences calculées se trouvent dans la plage de fréquence valide pour la taille d’image donnée.

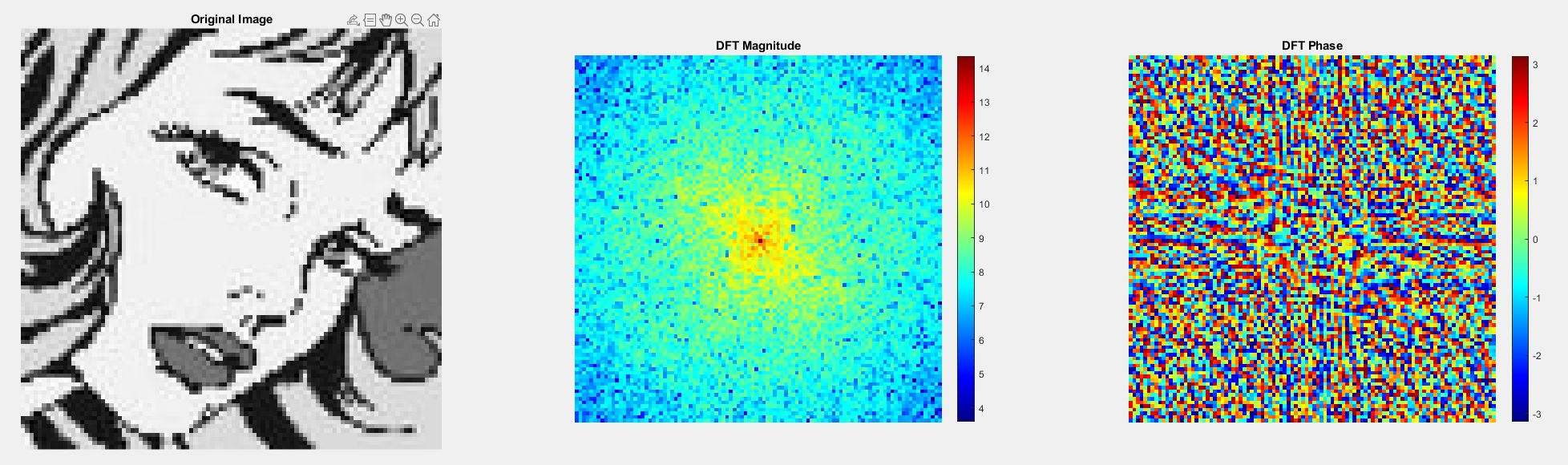
L’expression complexe représente la nature sinusoïdale des composantes de fréquence et aide à décomposer l’image en ses composantes de fréquence constitutives, permettant la transformation du domaine spatial ( image ) vers le domaine fréquentiel (spectre DFT ).

Une fois les calculs effectués, on décale les composants de valeur 0 au centre, pour une meilleure visualisation par la suite.

On calcule ensuite la magnitude du résultat, en prenant la valeur absolue ( module ) des composants DFT. Le spectre d’amplitude représente l’amplitude ou la force de chaque composante de fréquence dans l’image. Cela montre quelles fréquences sont dominantes et leur force respective.

Dans un spectre de magnitude non centré ( si nous n’avions pas fait le centrage juste avant ), la composante de coordonnée (0,0) représente l’intensité moyenne , ou correspond au niveau de luminosité global dans l’image, positionné à la fréquence (0,0). Les coins correspondent aux fréquences les plus basses dans les directions horizontales/verticales. Ils contiennent des informations relatives aux variations à grande échelle, bords ou motifs sur l’ensemble de l’image. Quand on se rapproche du centre, les fréquences augmentent et on rencontre des infos de fréquence plus élevées => informations sur des détails plus fins, textures et changements rapides dans l’image, tel que des textures fines, des petits bords ou du bruit. Nous utilisons dans nos algorithmes colormap(gca,’jet’), ce qui affiche les faibles fréquences en bleu et les hautes fréquences en rouge.

On calcule ensuite la phase du résultat, en utilisant angle(), qui calcule l’angle de phase des nombres complexes du résultat de la DFT. Le spectre de phase représente les informations de phase associées à chaque composante de fréquence. Cela donne des infos sur les déphasages des différentes fréquences de l’image. Il indique la relation de phase entre différentes parties de l’image correspondant à différentes fréquences.Ensuite, on fait une transformation logarithmique : on met log( 1 + magnitude\_spectrum ) dans magnitude\_spectrum. Cela compresse la plage de valeur et améliore le contraste dans la visualisation. Cela étale les valeurs des magnitudes inférieures tout en contrôlant l’effet des valeurs plus élevées, rendant la visualisation plus adaptée à l’œil humain. On affiche ensuite l’image de base, puis la magnitude et enfin la phase (résultats de la DFT ) :



**Transformée de Fourier rapide discrète 1D**

On commence par créer des variables/fréquences qui nous seront utiles :  
fs est la fréquence d’échantillonnage ( Hz ) | t est le vecteur temporel qui va de 0 -> 1s avec un pas déterminé par fs | f1 et f2 sont les fréquences de 2 ondes sinusoïdales ( Hz ) | x est un signal composé de f1 et f2 ( ici une somme de 2 sinusoïdes ).

On commence par trouver la puissance de 2 la plus proche de la longueur du signal x, puis on complète le signal x avec des 0 pour que sa longueur soit celle de la puissance de 2 trouvée. On fait cela car on va utiliser la stratégie diviser pour régner : l'algorithme divise le calcul en divisant la séquence d'entrée en sous-séquences plus petites. Lorsque la taille d'entrée est une puissance de 2, cela permet à l'algorithme de diviser récursivement la séquence en deux jusqu'à ce qu'elle atteigne des séquences de longueur 1, optimisant ainsi le processus.



On calcule ensuite les fréquences correspondantes au bins de la fft. N\_fft est la taille du signal après complétion par 0. (0 :N\_fft-1) génère des indices pour les bins fft de 0->N\_fft-1, multiplié par fs/N\_fft pour mettre a l’échelle ces indices pour représenter les fréquences réelles en Hz. Chaque index de fréquence résultant représente un groupe de fréquence spécifique dans la sortie fft. Ils contiennent des informations sur l’amplitude et la phase des composantes de fréquence correspondantes présentes dans le signal temporel d’origine. Ils aideront à reconstruire le spectre de fréquence du signal.

On va jusqu’à N\_fft-1 car cela garantit que les indices de fréquence correspondent au nombre de bins fft ( N\_fft ) tout en couvrant toute la plage de fréquence qui va de 0 jusqu’à la fréquence de Nyquist ( non incluse ) = fs/2, qui est la fréquence maximale représentable dans le signal discret. On appelle ensuite la fonction pour calculer la fft 1d :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquementCet algorithme utilise l’algorithme de Cooley-Tukey utilisant la méthode radix-2 decimation in time (DIT), en utilisant la récursivité.

On commence par vérifier que le signal x a une taille qui est une puissance de 2 en utilisant un bitwise ( opération and pour vérifier que la longueur et la longueur-1 sont différents de 0, ce qui veut dire que la longueur n’est pas une puissance de 2.

Ensuite, si la longueur du signal vaut 1, on retourne directement le signal d’entrée.

Sinon, on est dans le cas récursid ( longueur N > 1 ) : on commence par séparer les échantillons d’index pairs et impairs en x\_even et x\_odd. Puis on calcule récursivement la fft 1d pour les parties paires et impaires en s’appelant elle-même pour chaque partie séparément. On calcule ensuite le facteur Twiddle, correspondant au fft pour l’étape en cours.

Une image contenant Police, texte, typographie, ligne

Description générée automatiquementFormule du facteur Twiddle, avec N le nombre d’échantillons, k l’index de la bin actuelle dans l’étape fft et n l’index dans la bin.

* Nombres complexes pour effectuer des déphasages sur les composantes fréquentielles d’entrée pendant le calcul de la fft.

Il s’agit de coefficients appliqués aux opérations papillons à chaque étape de calcul fft : ils mettent en œuvre efficacement les déphasages nécessaires pour combiner les résultats fft de sous-problèmes plus petits ( opérations papillons ) en résultats fft plus grands.

* Déphasage : ils contrôlent le degré de déphasage des composants sinusoïdaux quand ils sont combinés avec d’autres composants pendant le processus de la fft.
* Multiplication complexe : la multiplication des éléments du signal d'entrée par des facteurs de rotation fait tourner ou décale efficacement les valeurs complexes des bins FFT, contribuant à l'alignement de phase et à l'agrégation des composantes de fréquence.



Ici, N est la longueur de la partie se faisant traiter. (0 :N/2-1) génère les indices (0->N/2-1) pour les facteurs twiddle correspondants au bins fft de l’étape courante.

L’expression exp() fait le calcul en utilisant la formule d’euler :

* -2j\*PI\*(0 :N/2-1)/N représente les angles de phase en radian pour les facteurs twiddle . /N met à l’échelle ces indices en fonction de la longueur de la partie actuelle du signal N.
* On multiplie par -2PI, ce qui fait un déphasage dans le plan complexe, pour aider la recombinaison des composantes fréquentielles.

On les calcule pour N/2 éléments car il y a une symétrie et une périodicité des propriétés de l’algorithme fft.

On recombine ensuite les résultats en utilisant les opérations papillons :

On fait l’ajout et le retrait des résultats des 2 calculs d’une manière spécifique, en utilisant des facteurs twiddle.

* On ajoute à la partie paire x\_even la partie impaire x\_odd multipliée par le facteur twiddle : ce qui permet d’aligner les fréquences de manière constructive -> signaux d’amplitude plus élevées quand ils sont en phase.
* On soustrait à la partie paire x\_even la partie impaire x\_odd multipliée par le facteur twiddle : ce qui permet d’aligner les fréquences de manière destructive -> annule/réduis l’amplitude des signaux en cas de déphasage.

Cela permet de réduire les signaux/fréquences qui ne sont pas en phase pour garder ceux en phase.

On multiplie élément par élément avec x-odd, pour s’assurer de l’alignement correct et de la combinaison correcte des composantes de fréquence pendant le calcul.

Le code ci-dessus concatène les deux parties ( addition et soustraction ) ;

Pour finir, on stocke le résultat dans le tableau X.

On affiche ensuite le signal original par rapport au temps t, puis on affiche la magnitude (valeur absolue) du résultat de la fft par rapport au vecteur fréquence frequencies. Il s’agit d’informations ds le domaine fréquentiel obtenues à partir de la fft, montrant les amplitudes des différentes composantes fréquentielles présentes dans le signal.

Une image contenant texte, ligne, diagramme, capture d’écran

Description générée automatiquement

**Transformée de Fourier rapide discrète 2D**

Dans cet algorithme, on prend une image en paramètre, on la transforme en niveaux de gris si elle est en couleur.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquementOn appelle ensuite la fonction pour calculer la fft 2d :

On récupère la taille de l’image ( comme les algorithmes précédents ), puis on crée une matrice de taille de la puissance de 2 la plus proche des tailles récupérées, complétées avec des 0, pour les mêmes raisons que précédemment.

On applique ensuite la fft 1d que nous avons construite sur les lignes et les colonnes de notre image.

Le .’ pour les colonnes permet de transposer le vecteur colonne en un vecteur ligne, pour le donner à l’algorithme fft 1d. X contient les résultats de la fft 2d de l’image de base et correspond à la représentation dans le domaine fréquentiel de l’image de base.  
On effectue ensuite les mêmes opérations que pour les autre algorithmes : c’est-à-dire calculer et Une image contenant Caractère coloré, capture d’écran, art, motif

Description générée automatiquementafficher la magnitude et la phase.

**Transformée de Fourier inverse rapide 1D**

Le but de cet algorithme est de reconstruire un signal à partir du résultat de la fft1d.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquementOn utilise les formules suivantes :

On utilise la même méthode qu’au début de la fft 1d : on crée un vecteur temps…

Puis on assemble un signal x.

Une image contenant texte, capture d’écran, Police

Description générée automatiquementComme on peut le voir, les 2 calculs sont presque les mêmes. La seule différence est que pour la directe on fait exp( -2i … ), alors que pour l’inverse on fait exp(2i … ).

La raison est la direction de la transformation du domaine fréquentiel.

Dans le calcul direct, l’exponentielle est utilisée pour transformer le signal du domaine temporel vers le domaine fréquentiel => décalage de -90 degrés pour les composantes de fréquence ( décalage vers la droite dans le plan complexe ). Pour l’inverse, l’exponentielle est utilisée pour reconvertir le domaine fréquentiel vers le domaine temporel =>déphasage de +90 degrés ( décalage vers la gauche dans le plan complexe ).

Les deux opérations sont opposées, et donc le signe est aussi opposé.

Une image contenant texte, capture d’écran, ligne, Police

Description générée automatiquementPuis, de la même manière, on affiche le signal de base, le résulat de la fft 1d, puis le résultat de la fft 1d inverse :

**Transformée de Fourier inverse rapide 2D**

Le principe de cet algorithme est le même que la transformée de Fourier rapide inverse 1d, mais cette fois sur une image.

Comme d’habitude, on charge une image et on la transforme en niveaux de gris si nécessaire.

On calcule ensuite la fft 2d ( celle vue précédemment ) , puis on calcule la ifft 2d :

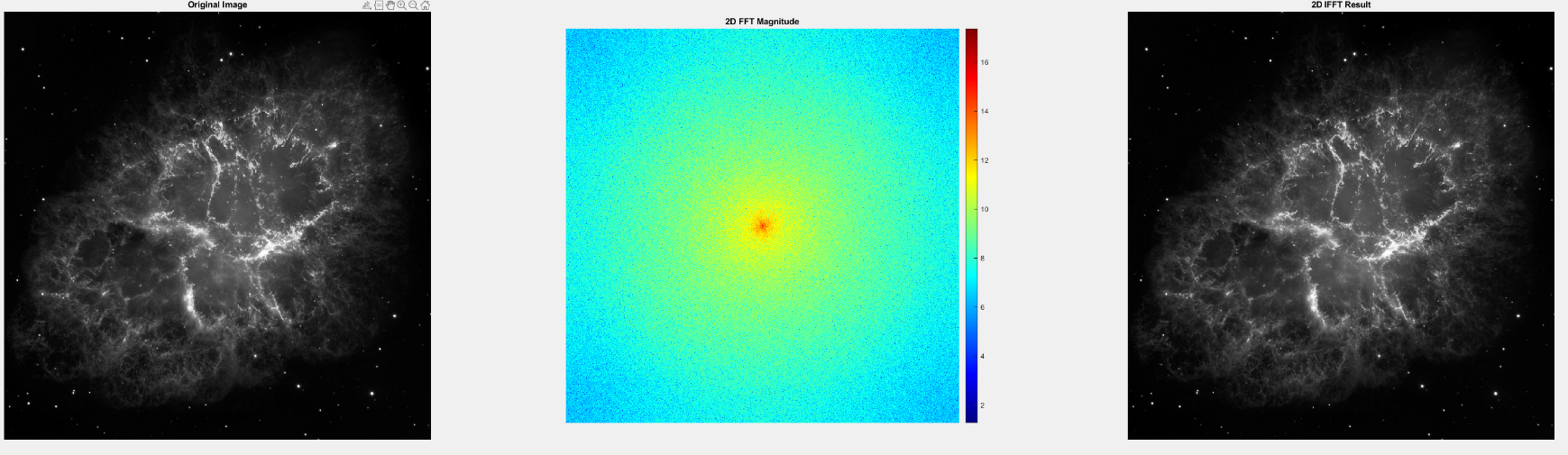
Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquementOn commence par calculer le conjugué complexe de la matrice d’entrée => pour chaque nombre complexe du résultat de la fft 2d, on calcule son conjugué, inversant le signe de la partie imaginaire.

On calcule ensuite la fft 2d du conjugué complexe du résultat de la 1ere fft 2d => cela fait l’opération inverse de la fft classique, visant à récupérer l’image originale à partir de sa représentation dans le domaine fréquentiel.

On récupère ensuite la taille du résultat pour faire une mise à l’échelle du résultat ifft en divisant chaque élément par le nombre total d’élément, ce qui garantit que l’amplitude de l’image résultante reste cohérente avec l’image d’origine.

On ne renvoie que la partie réelle du résultat pour éliminer les composantes imaginaires, car une image ne contient que des valeurs réelles.

Comme d’habitude, on affiche l’image de base, la magnitude et l’image reconstruite :

**Détecteur de contours**

Nous avons décidé de « concevoir » un détecteur de contours, utilisant les algorithmes vus précédemment.

On commence par prendre une image et la transformer en niveaux de gris si nécessaire. On récupère sa taille. Puis on calcule la fft 2d pour cette image, puis on affiche la magnitude de ce résultat.

On applique ensuite un filtre Laplacien sur le résultat

* Élément central (8) : L'élément central de la matrice a la valeur la plus élevée (8), indiquant le poids attribué au pixel lui-même dans la réponse Laplacienne. Il contribue le plus au calcul et met l'accent sur l'intensité des pixels au centre de la région 3x3.
* Autres éléments (-1) : Les autres éléments de la matrice ont une valeur de -1. Ces éléments représentent les contributions des pixels voisins et influencent la réponse Laplacienne en considérant les différences entre le pixel central et ses voisins. Ils aident à détecter les changements rapides ou les contours de l'image.

La matrice a un pic central entouré de poids négatifs. La combinaison de la valeur centrale positive élevée et des valeurs négatives environnantes permet de mettre en valeur les régions où l'intensité des pixels change rapidement, indiquant les bords de l'image.

Cette matrice améliore ou met en évidence les contours et les régions de changement rapide d'intensité au sein de l'image, fournissant une réponse qui met l'accent sur ces caractéristiques pour des tâches de traitement ultérieur ou de détection de contours.

Les valeurs positives de la réponse Laplacienne indiquent une augmentation de l'intensité (du clair au foncé ou du foncé au clair), et les valeurs négatives indiquent une diminution de l'intensité.

La réponse du filtre Laplacien est au plus haut au niveau des bords ou des régions avec un taux de changement élevé dans l'intensité des pixels, ce qui le rend adapté aux tâches de détection de contours.

Le filtre Laplacien est sensible au bruit car il amplifie les composantes haute fréquence, y compris le bruit présent dans l'image.

Ici, nous appliquons la matrice Laplacienne dans le domaine fréquentiel. Cette application est utilisée pour mettre en évidence les bords ou les régions de changement rapide d'intensité en filtrant les composantes fréquentielles de l'image.

Une fois le filtre créé, on le complète par des zéros pour correspondre à la taille de l’image, ce qui garantit que la matrice de filtre a la même taille que l'image pour une multiplication correcte dans le domaine fréquentiel.

La fonction fft2d est appliquée à la matrice de filtre Laplacien à remplissage nul.

La FFT 2d résultante du filtre Laplacien est multipliée élément par élément avec le résultat FFT 2d de l'image originale (fft2\_result), appliquant efficacement le filtre Laplacien dans le domaine fréquentiel. Le filtre Laplacien est initialement représenté dans le domaine spatial (sous forme de matrice). Pour effectuer un filtrage dans le domaine fréquentiel, la matrice du filtre Laplacien est transformée dans le domaine fréquentiel à l'aide de la fonction fft2d, qui calcule vraisemblablement la transformée de Fourier rapide (FFT) 2D du filtre.

On affiche ensuite la magnitude centrée du résultat de ce filtre, puis cette même magnitude, mais décentrée.

On utilise ensuite la fonction ifft 2d pour retrouver une image dans le domaine spatial.

Après avoir effectué la FFT inverse, la valeur absolue du résultat (ifft2\_result) est utilisée pour gérer toutes les composantes à valeurs complexes qui pourraient survenir en raison d'opérations numériques.

Nous appliquons ensuite un seuil (abs(ifft2\_result) > 30) pour créer une image binaire (binary\_result), où les pixels d'intensité supérieure à 30 sont définis sur 1 (blanc) et les autres sont définis sur 0 (noir). Le seuillage est utilisé pour créer une représentation binaire de l’image, qui met en évidence les caractéristiques ou les régions spécifiques en fonction des niveaux d'intensité. Il permet de mettre en valeur ou d'isoler les zones d'intérêt, telles que les bords, les contours ou les régions présentant des caractéristiques d'intensité particulières.

L'ajustement de la valeur seuil (30 dans ce cas) a un impact sur l'image binaire résultante, affectant la sensibilité de la détection des contours ou de l'extraction des caractéristiques.

Une image contenant texte, capture d’écran, ciel, graphisme

Description générée automatiquementL'image binaire finale (binary\_result) représentant les bords ou régions d'intérêt détectés en fonction de l'opération de filtrage est ensuite affichée : imshow(binary\_result, []) affiche l'image binaire sans mettre à l'échelle la plage d'affichage. Les valeurs supérieures au seuil apparaissent sous forme de zones blanches, indiquant les contours ou caractéristiques identifiés, tandis que le noir représente d'autres zones.

Une image contenant texte, capture d’écran, graphisme, art

Description générée automatiquement