

Quantification du détachement par un
écoulement cisailé de bioparticules adhérant à
un support

Juan Carlos Garcia Hernandez, Nicolas Bonneel

11 janvier 2006

Table des matières

1	Présentation du projet	3
1.1	Présentation de l'expérience	3
1.2	Présentation du sujet	4
2	Méthodes testées	6
2.1	Réduction du bruit	6
2.1.1	Filtre de convolution	6
2.1.2	Filtre NxN	6
2.1.3	Filtre du rang	7
2.2	Rehaussement de contours	8
2.2.1	Gradient de Sobel	8
2.2.2	Gradient de Roberts	8
2.2.3	Laplacien	9
2.2.4	Filtre de Reimer	10
2.2.5	Filtre du rang	10
2.2.6	Techniques morphologiques	11
3	Chaînes de Traitement	11
4	Méthode finalement employée	13
4.1	Traitements	13
4.2	Résultats	17
4.3	Difficultés rencontrées	17
4.3.1	Liées au logiciel	17
4.3.2	Liées au matériel	18
5	Conclusion	19

1 Présentation du projet

Ce sujet est proposé par le département de Génie Biochimique et Alimentaire, et a pour but le comptage de particules dans une image.

1.1 Présentation de l'expérience

Des particules de différents types (cellules de champignons, billes de latex...) sont disposées sur différents types de plaques. Pour une expérience donnée, nous avons donc un type de particule sur un seul type de plaque.

Un courant d'eau est appliqué sur la plaque progressivement, dans une direction donnée, à débit croissant, emportant sur son passage les particules (ou pouvant juste les déplacer). Pour le débit nul et pour différents débits une image des particules est prise au microscope optique. Nous pouvons aussi avoir plusieurs images de différentes zones de la plaque.

Ainsi, sur chaque image nous observons de moins en moins de particules, car celles-ci se détachent de la plaque.

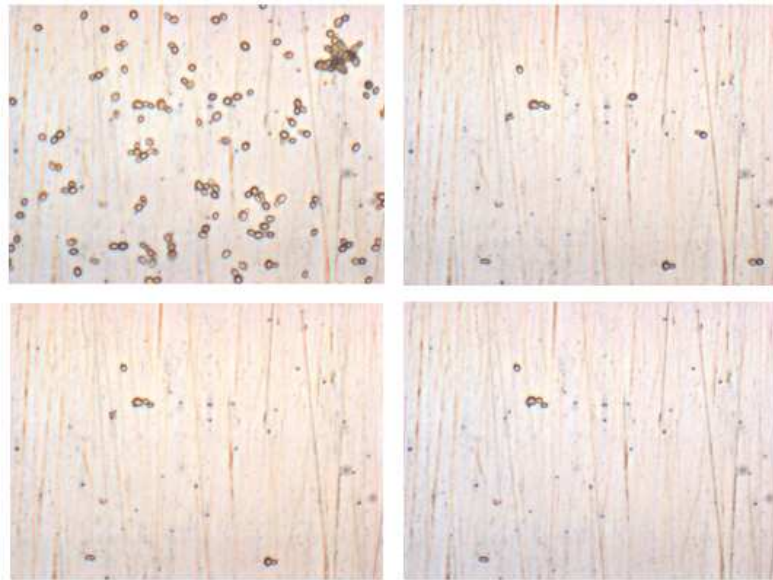


FIG. 1 – Exemple de 4 images : débit nul et 3 débits croissants

Nous n'observons ainsi qu'une partie de la plaque (schéma ci-dessous).

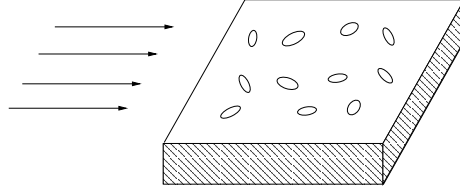


FIG. 2 – Schéma récapitulatif de la manipulation

Cette expérience a pour but d'étudier les forces de cohésion entre les particules et les plaques, et pouvoir ainsi modéliser toutes les forces entrant en jeu sur des particules de cette échelle. Ceci peut permettre, par exemple, de savoir quel débit d'eau il faut utiliser sur une plaque, un tube etc.. pour les nettoyer d'un type de particule. On peut en effet retrouver la force entre une particule et son support à partir du diagramme du nombre de particules sur le nombre de particules initiales : $\frac{N_i}{N_0}$, pour différents débits i par des équations de mécanique des fluides.

1.2 Présentation du sujet

Ayant obtenu différentes images de particules, le but est de les compter. Les particules trop proches les unes des autres gênent le flux qui ne se comporte plus de la même manière. Il faudra donc ne pas compter les particules trop proches.

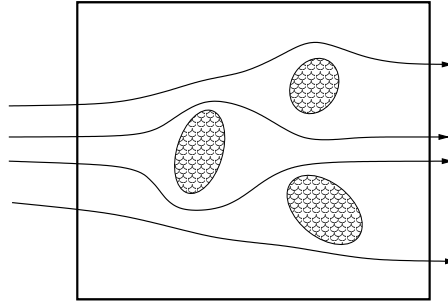


FIG. 3 – Flux perturbé par la présence d'une particule trop proche d'une autre

Par ailleurs, il faudra aussi supprimer les cellules bourgeonnantes (cellules en train de germer) et les particules agglomérées entres-elles.

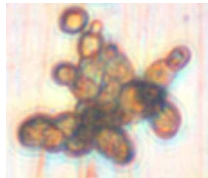


FIG. 4 – Agglomérat de cellules



FIG. 5 – Cellule bourgeonnante

De plus, la plaque pouvant être ponçée, elle peut présenter des rayures et divers bruits. Ceux-ci ne devront pas être pris pour des particules.

Nous avons à notre disposition un logiciel de traitement d'images : Analysis, et nous devons ainsi mettre au point un script (dans un langage similaire au C) chargé d'isoler les particules, ce logiciel étant capable de les compter, et tracer le graphe $\frac{N_i}{N_0}$.

Un premier script donnant des résultats assez moyens permettait déjà un comptage des particules en se basant sur l'image initiale (l'image à débit nul) et en faisant des différences avec cette image. Nous avons dans un premier temps cherché une série de traitement permettant d'enlever les rayures et ainsi de s'affranchir de la passe de soustraction entre images (ce qui peut s'avérer désastreux si le microscope a bougé de plusieurs pixels). De plus, nous voulions nous affranchir de l'étape de seuillage requis par l'ancien script, pour ainsi proposer une chaîne de traitements entièrement automatique. Par ailleurs, l'ancien script comptant le nombre de particules disparues entre chaque images, le cas de particules se déplaçant dans l'image posait un problème que nous aurions pu régler plus facilement en comptant le nombre réel de particules par image.

Manquant de résultats satisfaisant (le langage de script du logiciel Analysis n'étant finalement pas très pratique, nous ne pouvions faire tout ce dont nous avions besoin) nous sommes revenus au script initial Analysis faisant des différences entre images, que l'on a amélioré et auquel nous avons couplé un script Matlab pour ajuster les images si celles-ci étaient décalées.

2 Méthodes testées

Tout d'abord, nous avons remarqué que la localisation des particules repose sur la détection de contours. Il a fallu donc connaître les possibilités offertes par le logiciel de façon à pouvoir concevoir des chaînes de traitement qui rendent la détection de contours plus aisée et indépendante d'un seuillage ou paramétrisation quelconque.

Les chaînes de traitement testées sont constituées des méthodes décrites plus loin. Elles comprennent des méthodes pour rehausser les contours et pour réduire le bruit, ces dernières étant justifiées par la sensibilité au bruit des méthodes de rehaussement et détection de contours. Finalement, une fois les contours accentués on a cherché à utiliser des méthodes morphologiques afin de mieux profiter du traitement préalable.

2.1 Réduction du bruit

2.1.1 Filtre de convolution

En sachant que le bruit constitue une grande partie des hautes fréquences, la méthode la plus naturelle est d'utiliser un filtre passe-bas. Imaging C propose des filtres du type FIR (réponse impulsionnelle finie) dont la définition du masque requiert à chaque fois l'intervention de l'utilisateur. En outre, la taille maximale du masque pour ces filtres étant de 9×9 cela nous a paru insuffisant car on cherchait à éliminer la partie supérieure du spectre sans pour autant anéantir les contours (les contours constituant aussi des hautes fréquences). En effet, le bruit constitue la toute dernière partie du spectre et donc on souhaite que le filtre de change rapidement sur une zone très réduite, cela impose à la fonction de transfert du filtre d'être très raide. La raideur étant liée directement à la taille du filtre par le principe d'Heisenberg, un filtre avec un masque 9×9 n'est pas adapté à nos besoins compte tenu de la taille des images.

On peut aussi considérer des filtres FIR passe-haut afin de renforcer les détails d'une image, en sachant que ces derniers sont constitués par les contours.

2.1.2 Filtre $N \times N$

Il s'agit du filtre de la moyenne dont le masque est une matrice carrée remplie de 1. Dans le but de conserver l'information, chaque élément de la matrice doit être divisée par le nombre d'éléments de cette matrice. Ce filtre sert à réduire les bruits statistiques, c'est-à-dire, il minimise l'apport d'une variable aléatoire qui aurait pu s'ajouter à l'image. Ce type de bruit est inhérent au processus d'acquisition.

Le filtre $N \times N$ est par construction un filtre qui lisse les données, on a donc fait particulièrement attention à son utilisation afin de ne pas adoucir excessivement les contours avec un filtre de taille démesurée. On remarque aussi, d'après la valeur absolue de sa fonction de transfert, que ce filtre génère des oscillations dans le spectre du signal initial. Ceci peut s'avérer un inconvénient lorsqu'on souhaite avoir une prépondérance des fréquences associées aux contours dans le spectre de l'image.

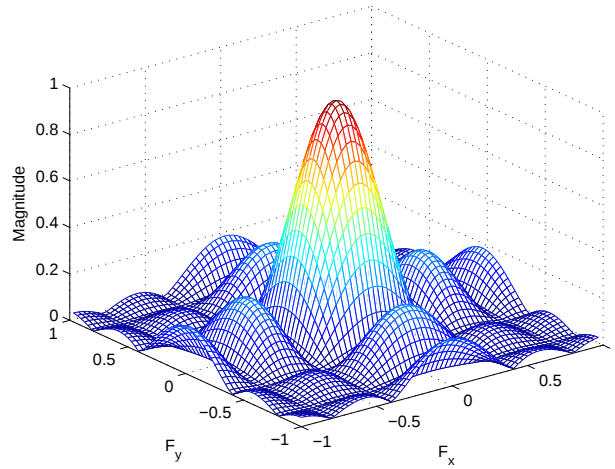


FIG. 6 – Fonction de transfert du filtre $N \times N$

2.1.3 Filtre du rang

Celui-ci est un filtre, appelé ainsi par abus de langage, qui permet de réduire le bruit statistique aussi bien que les bruits locaux. Pour le filtre de rang $\alpha \in]0, 1[$ la valeur d'un pixel est déterminée par le quantil d'ordre α sur un voisinage du pixel. Lorsque $\alpha = 0.5$, il s'agit du filtre de la *mediane*. Il est peu sensible aux valeurs aberrantes contrairement aux filtres $N \times N$ et FIR.

Ce filtre étant non linéaire, on ne peut pas calculer sa transformée de Fourier pour connaître son action sur le spectre d'une image. Cependant il est possible de voir comment il agit sur les contours et le comparer à un filtre $N \times N$.

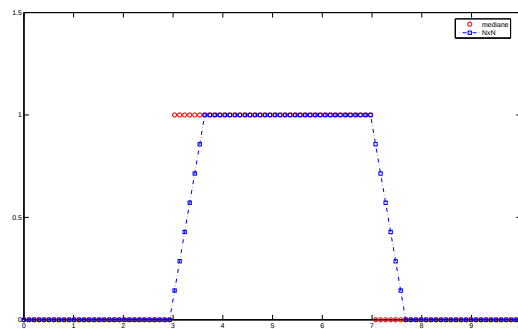


FIG. 7 – Résultat du filtrage de la fonction indicatrice de $[3,7]$ par un filtre de la médiane et un filtre $N \times N$ de taille 7

Il est évident que le filtre de la médiane se comporte d'une meilleure façon par rapport aux contours, car, à l'opposé du filtre $N \times N$, il n'adoucit pas les contours.

2.2 Rehaussement de contours

2.2.1 Gradient de Sobel

C'est un opérateur qui permet d'estimer localement la norme du gradient sur une image. Il amplifie les zones de fortes variations locales correspondant aux contours. Il est formé par deux noyaux de convolution dont l'expression est :

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Si l'on note I^x et I^y le résultat de la convolution de l'image (I) par les noyaux G_x et G_y , on a une estimation de la norme du gradient au pixel (i, j) donnée par :

$$\|\nabla I\|_{i,j} = \sqrt{I_{i,j}^{x\,2} + I_{i,j}^{y\,2}}$$

C'est ainsi qu'est défini l'opérateur de Sobel. Cependant la quantité précédente est souvent approchée par la formule suivante qui est plus rapide à calculer :

$$\|\nabla I\|_{i,j} = |I_{i,j}^x| + |I_{i,j}^y|$$

On s'aperçoit que les noyaux sont liés par une rotation de $\frac{\pi}{2}$ de l'espace de définition des noyaux. Pour pouvoir mieux expliquer son fonctionnement, on remarque qu'on a l'égalité suivante :

$$G_x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = h_1 \cdot h_2^T$$

Ceci montre que ce filtre est séparable, c'est-à-dire que son noyau est à variables séparées. Filtrer une image par le noyau G_x équivaut donc à appliquer la succession de deux filtres :

- filtrer les colonnes, ie. suivant y, par un filtre dont le noyau est $h_1 = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}^T$ qui est un intégrateur, donc un filtre passe bas.
- filtrer les lignes, ie. suivant x, en utilisant le noyau est $h_2 = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}^T$, ce filtre est une approximation centrée de la dérivée d'après la formule : $f'(x) = \frac{f(x+h) - f(x-h)}{2 \cdot h} + o(h)$. On détecte donc les contours verticaux.

Pour G_y , on a les propriétés équivalentes. Ainsi, l'opérateur de Sobel détecte les contours suivant deux directions principales, à savoir horizontales et verticales. De plus, le filtrage passe-bas réalisé par chaque noyau lui confère une bonne stabilité par rapport aux bruits. Ces propriétés sont retrouvées lorsqu'on regarde la réponse impulsionnelle de chaque noyau.

2.2.2 Gradient de Roberts

C'est un filtre pour détecter les contours dont l'intérêt principal est sa rapidité. Il est semblable à l'opérateur de Sobel, il est constitué par les noyaux suivants :

$$H_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad H_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

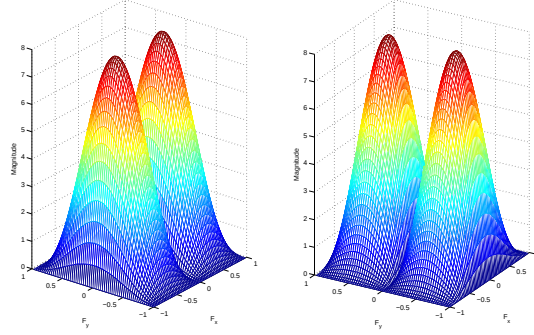


FIG. 8 – Réponse impulsionnelle des filtres associés aux noyaux G_x et G_y

La convolution de l'image (I) par le noyau H_i constitue l'approximation de la dérivée de I suivant l'orthogonal à l' i -ème diagonale principale et donc on détecte les contours suivant l' i -ème diagonale. En effet on a :

$$f(x + h_x, y + h_y) - f(x, y) = \partial_1 f(x, y)h_x + \partial_2 f(x, y)h_y + o(\|h\|)$$

Dans le cas de H_1 on a $h = [1 \ -1]$ et pour H_2 on a $h = [-1 \ -1]$. Ce filtre n'est constitué que par une approximation de la dérivée suivant une direction, il est donc très sensible au bruit. De plus il produit de faibles valeurs sur les contours, ceci est dû au fait que l'approximation de la dérivée n'est que de premier ordre, contrairement à l'opérateur de Sobel qui en fait une de second ordre. On identifie ces propriétés dans les graphes des réponses impulsionnelles :

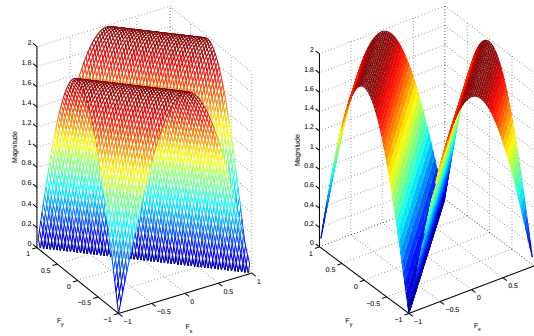


FIG. 9 – Réponse impulsionnelle des filtres associés aux noyaux H_1 et H_2

2.2.3 Laplacien

Une frontière est un lieu de forte variation. Les contours d'une image constituent les endroits où le gradient est maximal. L'idée derrière cette méthode est la recherche des points où la dérivée seconde s'annule, c'est à dire là où le gradient est extrême. Cependant pour une image, on a quatre dérivées secondes : $\frac{\partial^2 f}{\partial x^2}$,

$\frac{\partial^2 f}{\partial x \partial y}$, $\frac{\partial^2 f}{\partial y \partial x}$ et $\frac{\partial^2 f}{\partial y^2}$ (trois si la fonction est $\mathcal{C}^2$). Pour s'affranchir de ce problème nous avons recours au Laplacien qui fait la somme des deux dérivées secondes principales. Son noyau est obtenu à partir de la formule :

$$f(x+h) - 2f(x) + f(x-h) = f''(x)h^2 + o(h^2)$$

Ainsi, par addition de la dérivée seconde suivant x et y , on a le noyau suivant :

$$H = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Ce filtre a deux principaux inconvénients, le premier est que le laplacien peut être nul (au seuil près) dans des endroits où il n'y a pas de contours francs, plus exactement là où le gradient présente des variations faibles. Le deuxième est qu'il est très sensible au bruit (cf figure 10), il est donc utilisé à la suite d'un filtrage avec un filtre passe-bas, typiquement un filtre gaussien après quoi on obtient un comportement plus homogène au niveau des hautes fréquences (cf. figure 10).

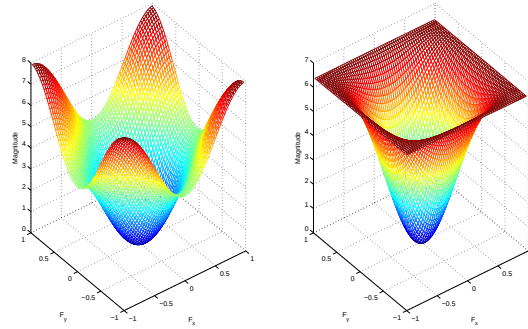


FIG. 10 – Réponse impulsionnelle du noyau de Laplace seul et celui convolé avec un noyau gaussien. On remarque l'homogénéité du deuxième suivant les hautes fréquences

2.2.4 Filtre de Reimer

Ce filtre nous permet de calculer une image en équidensité de deuxième ordre à partir de l'image originale. Dans une image en équidensité, les niveaux de gris qui se trouvent dans un intervalle de niveaux de gris dans l'original ne sont représentés que par un seul niveau de gris. Si dans l'image en équidensité, seules les limites entre les aires de mêmes couleurs sont restituées, on parle d'une image en équidensité de deuxième ordre.

Le filtre Reimer rend visibles des écarts de niveaux de gris constants que l'oeil n'arrive plus à discerner.

2.2.5 Filtre du rang

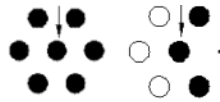
Lorsque les valeurs de α sont extrêmes ce filtre permet d'accentuer les contours tout en gardant sa stabilité vis-à-vis des valeurs aberrantes. Pour ce

filtre, il est conseillé d'utiliser un voisinage petit par rapport aux contours qu'on souhait faire ressortir. En effet, ce filtre fait disparaître des contours dont la taille est strictement inférieur à la taille du voisinage pris par le filtre multiplié par α .

2.2.6 Techniques morphologiques

Lorsqu'on réalise une accentuation de contours, il peut arriver que l'on n'ait pas des contour fermés, ce qui rend plus difficile le travail d'Analysis lors de la détection des particules. On fait donc recours aux techniques morphologiques qui s'appliquent sur des images binaires (ou niveaux de gris) et permettent de mettre en évidence des propriétés particulières tels que les contours.

Ces techniques sont basées sur l'utilisation d'un noyau appelé "élément structurant", noté E , qui est une configuration élémentaire à rechercher dans l'image, ci-dessous deux exemples de noyaux :



Avec :

- pixel valant 1
- pixel valant 0
- ↓ pixel d'origine ou pixel dont on détermine la valeur
- pixel indifférent

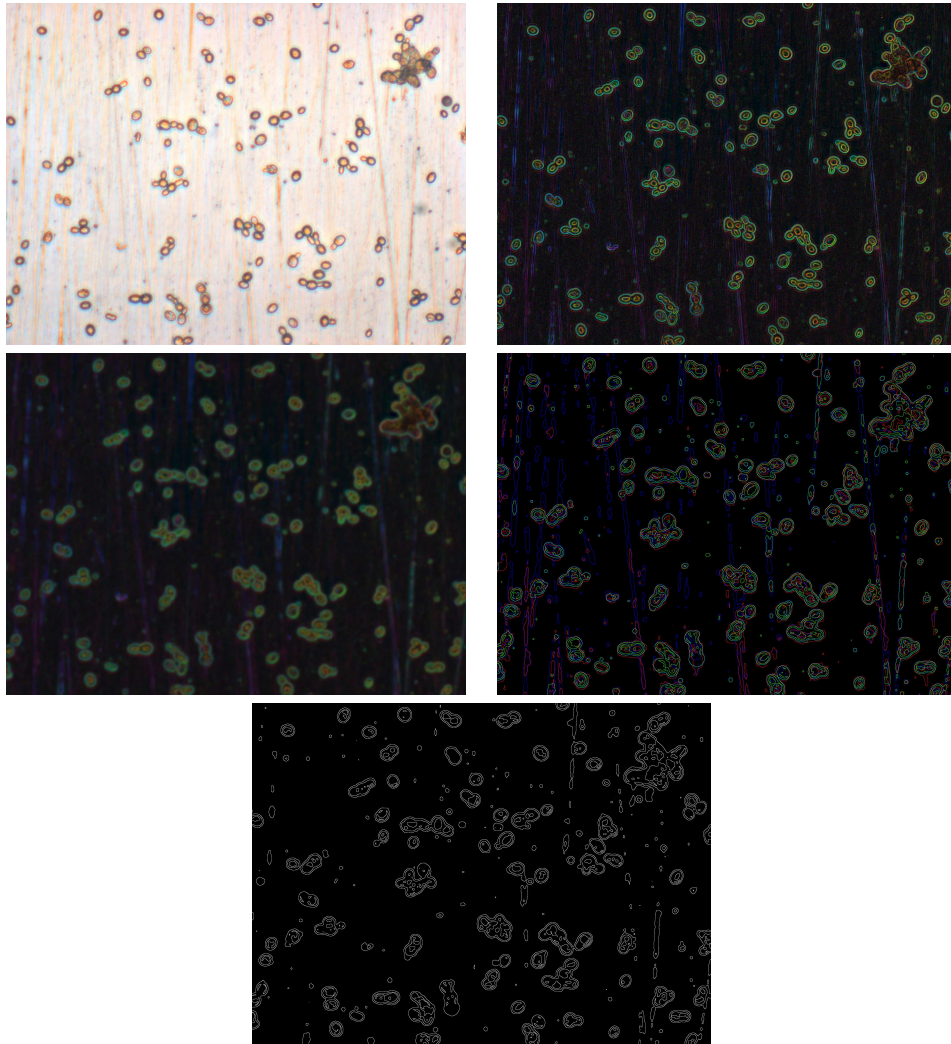
On note E' le symétrique d' E par rapport au pixel d'origine. Ainsi, on peut définir plusieurs opérations morphologiques :

- *Dilatation* : C'est la convolution booléenne par E , c'est à dire, la valeur du pixel d'origine est 1 s'il existe un pixel de E valant 1 ainsi que son correspondant sur l'image. Ce filtre permet généralement d'agrandir les structures d'une image en comblant les trous.
- *Érosion* : La valeur du pixel d'origine est 1 si tous les pixels valant 1 en E ont un correspondant sur l'image dont la valeur est 1. Ce filtre permet de réduire les structures présentes dans l'image en intensifiant les trous.
- *Ouverture* Cette opération a pour but d'isoler les surfaces présentes dans l'image en supprimant les surfaces trop petites mais en conservant l'aire des surfaces présentes et lisse les contours. Il s'agit d'une érosion par E suivie d'une dilatation par E'
- *Fermeture* Cette opération a pour but de fermer les contours disjoints et combler les trous suffisamment petits dans les surfaces et lisser les contours. C'est une dilatation par E suivie d'une érosion par E'

Les opérations dont on se servira le plus sont l'ouverture et la fermeture afin d'isoler et de rendre connexes les particules.

3 Chaînes de Traitement

Plusieurs bancs de filtres ont été testés, les résultats étant très différents, on donne ici les chaînes de traitement qui présentent les meilleures performances. Le démarche étant de rehausser les contours, lisser l'image et finalement d'effectuer le passage en binaire.



TAB. 1 – Chaîne : originale, Sobel, lissage, Reimer et séparation de l'intensité

4 Méthode finalement employée

4.1 Traitements

Nous avons finalement repris le script initial. Celui-ci avait un algorithme simple :

```
s = Demander une valeur de seuillage
Pour les image i de 2 à N
    Diff = image[i] - image[1]
    Gris = Convertir_en_niveaux_de_gris(Diff)
    Bin = Binariser(Gris, s)
    Ferm = Fermeture_Morphologique(Bin)
    Detecter_les_particules
Fin Pour
```

L'opération de différence est une opération très délicate, car si aucune particule ne s'est détaché mais que le microscope a légèrement bougé entre deux images, la différence ne sera pas nulle, et la fermeture morphologique qui s'ensuite risque de nous créer de nouvelles particules qui ne devraient pas exister. De même, lorsque le flux n'est pas suffisant pour emporter complètement la particule, mais juste suffisant pour l'avoir déplacé de quelques pixels, cette différence entre les deux images nous créera une nouvelle particule au nouvel emplacement de la particule qui s'est déplacée.

Nous avons donc créé un script Matlab cherchant un maximum de corrélation entre deux images dont l'une est translatée à chaque fois d'un vecteur (dx, dy) (pour chaque image, l'image courante et une image de référence) dans une petite zone pour ainsi recadrer les images décalées avant d'utiliser le script d'Analysis. Pour cela nous gardons la translation rendant la norme de la différence entre l'image translatée et l'image de référence minimale. Nous ne pouvons calculer qu'une approximation de cette norme en prenant un nombre restreint de pixels de l'image, cette opération devant rester rapide et les images étant relativement grandes (env. 2000*2000 pixels).

Nous sauvons ainsi les images décalées dans des fichiers .TIF, en faisant attention à ne conserver que la zone commune à toutes les images, c'est à dire en supprimant les bords noirs (sinon l'opération de différence serait désastreuse). Ce script permet aussi de normaliser les noms de fichiers qui ont souvent des dénominations variables, en une série de fichier du type : "nom de la manipulation + '_' + numéro de l'image.TIFF".

Nous avons résolu le problème du bruit statique important dans les images (rayures sur la plaque etc...) en utilisant une image initiale qui soit entièrement vide de particules et ne donnant ainsi qu'une image de la plaque, qui sera considérée comme la dernière image de la série. Cela résoud aussi le problème de la première image qui n'était pas comptée dans le script original, et donc, il ne fournissait que le nombre de particules qui se détachaient, sans jamais connaître finalement le nombre réel de particules initial (qui pouvait cependant être retrouvé si à l'image finale il n'y avait plus du tout de particules, ce qui n'était pas le cas).

Nous avons ensuite opéré à chaque étape à la différence entre l'image courante et l'image précédente et non l'image initiale. Ceci permet d'éviter les erreurs cumulées (baisse progressive de la luminosité de la pièce par exemple).

Nous avons cherché à homogénéiser l'intensité des images en utilisant la fonction de correction d'éclairage d'Analysis, mais celle-ci donnait des résultats non exploitables (images beaucoup trop contrastées, mettant en relief les rayures de la plaque et donc ne facilitant pas les traitements).

De plus, nous avons ajouté une ouverture morphologique sur l'image en niveaux de gris qui permet déjà de supprimer les petites surfaces assimilables à du bruit.

Une fois cette chaîne de traitements accomplie nous obtenons une image en noir et blanc (sans niveaux de gris). Analysis est alors capable de compter les particules présentes sur l'image et de fournir des propriétés de chaque particules (Aire, périmètre, ellipticité, sphéricité, facteur de forme...).

Nous nous servons ainsi de l'aire de chaque particule pour déterminer s'il s'agit d'un agglomérat de particules ou d'un bruit grâce à un seuil haut et un seuil bas (les particules trop grosses étant considérées comme plusieurs particules aggrégées et les particules trop petites étant considérées comme des bruits). Nous nous servons aussi du facteur de forme des particules (dans le schéma : $\frac{b}{a}$, le plus grand diamètre sur le plus petit diamètre de l'ellipse circonscrite à la particule) pour déterminer si une cellule est bourgeonnante ou si nous avons affaire à une rayure.

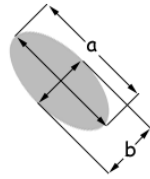


FIG. 11 – Facteur de forme

Ainsi, les particules ayant un facteur de forme au dessus d'un certain seuil seront considérées comme étant une rayure (une rayure n'est pas ronde... elle est plutôt elliptique) ou comme une cellule bourgeonnante (deux cellules qui se joignent auront une ellipse circonscrite de facteur de forme supérieur à 1 alors qu'une cellule unique est plutôt sphérique donc aura un facteur de forme proche de 1).

Nous utilisons aussi le centre de gravité de chaque particule par rapport à celui de la particule la plus proche dans l'image de référence. Si la distance entre ces deux centres est trop importante, il s'agit d'un bruit qui n'a pas été détecté par les seuils précédents ou que la particule s'est déplacée de façon trop importante (nous la supprimerons dans ce cas rare) ou encore d'un agglomérat de particules se désagrégeant particule par particule (en effet, dans ce cas la distance de la particule dont nous venons de détecter le départ à la particule la plus proche de taille raisonnable dans l'image initiale sera importante). Cette méthode permet de conserver les particules qui se sont déplacées de quelques pixels d'une image sur l'autre si ce déplacement n'est pas trop important.

De même nous calculons la distance à la deuxième particule la plus proche dans l'image de référence (la première étant censée être la particule elle-même) nous permettant ainsi d'éliminer les particules trop proches les unes des autres.

et perturbant ainsi l'écoulement. Aussi, nous calculons le diamètre moyen de toutes les particules de la première image vérifiant les critères de facteur de forme et d'aire), et nous demandons le nombre de diamètres moyen entre deux centres de particules comme critère de distance.

Nous cumulon ainsi toutes les particules respectant tous ces critères et pouvons ainsi tracer la courbe attendue $\frac{N_i}{N_0}$.

Nous obtenons ainsi l'algorithme suivant :

```
// Phase de recadrage (prétraitement Matlab)
Pour chaque image i
  Pour x = -dx à dx
    Pour y = -dy à dy
      d = norm(Image[1](dx:lx:Nx-dx, dy:ly:Ny-dy) - Image[i](dx:lx:Nx-dx, dy:ly:Ny-dy) )
      Si d < d_min
        X_optimum = x
        Y_optimum = y
        d_min = d
      Fin Si
    Fin Pour
  Fin Pour

  Sauver Image[i]( (dx:end-dx) + X_optimum, (dy:end-dy) + Y_optimum )
Fin Pour

// Phase de traitement (sous Analysis) :
s = Demander une valeur de seuillage
Pour les image i de 2 à N
  Diff = image[i] - image[i-1]
  Gris = Convertir_en_niveaux_de_gris(Diff)
  Ouv = Ouverture_Morphologique(Gris)
  Bin = Binariser(Ouv, s)
  Ferm = Fermeture_Morphologique(Bin)
  Detecter_les_particules
Fin Pour

// Phase de seuillage (sous Analysis) :
d_moyen = Calculer le diamètre moyen de particules valides de l'image 1
Pour les image traitées i de 1 à N-1
  Pour chaque particule P de l'image i
    Eliminer si diamètre < seuil1
    Eliminer si diamètre > seuil2
    Eliminer si facteur de forme < seuil3

    distance_min1 = 1E10
  Pour chaque particule P' de l'image 1
    Si la particule P' verifie les critères(seuil1, seuil2, seuil3)
      d = distance(P,P')
      Si d < distance_min1
```

```

        distance_min2 = distance_min1
        distance_min1 = d
    Fin Si
Fin Si
Fin Pour

Eliminer si distance_min1 > seuil4
Eliminer si distance_min2 < seuil5*d_moyen

Fin Pour

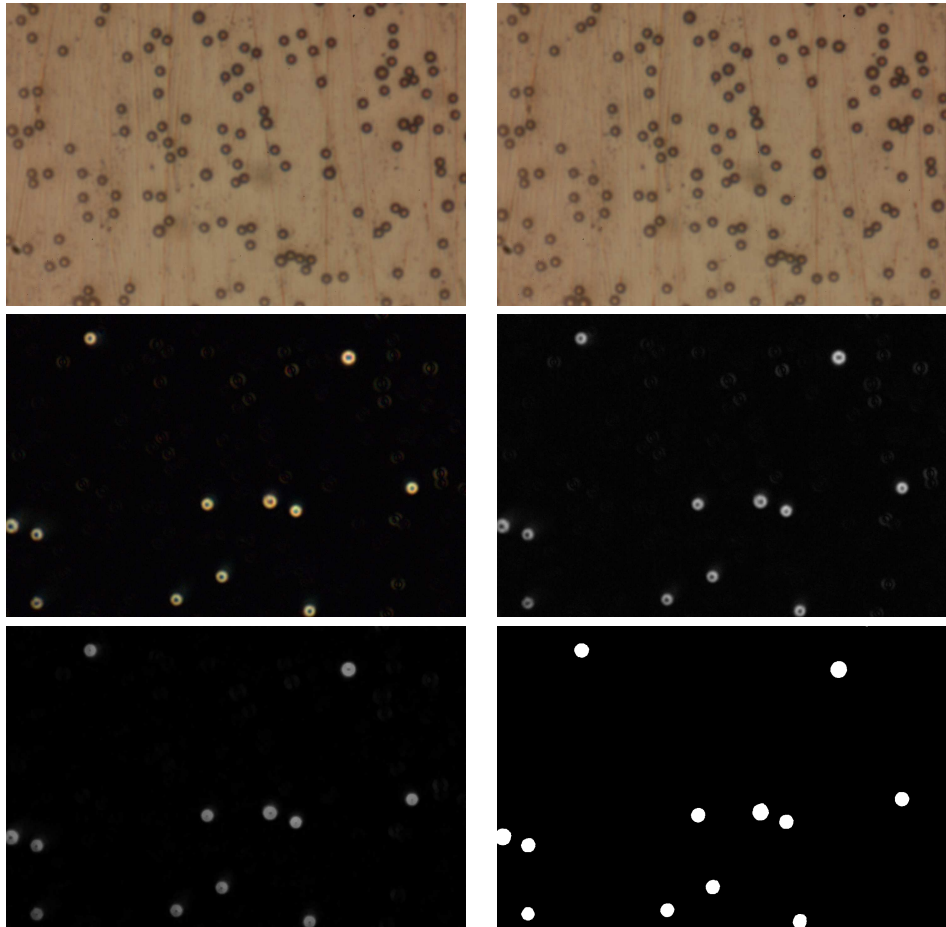
Si i = 1
    N0 = nombre de particules restantes
    N = N0
Sinon
    N = N0 - nombre de particules restantes
Fin Si

X(i) = i
Y(i) = N/N0
Fin Pour

Tracer de (X(i), Y(i))

```

Pour conserver une intervention minimale de l'opérateur, les seuils 1 à 4 sont fixés (mais facilement modifiables dans le programme si le besoin en était pour une application très particulière).



TAB. 2 – Algorithme final : images consecutives, difference, séparation d'intensité, ouverture morphologique et seuillage-fermeture

4.2 Résultats

Nous observons que les résultats obtenus sont conformes au comptage manuel. Il est à noter que le comptage manuel n'est pas un nombre de particules théoriques mais juste une indication des résultats que nous devrions trouver, le comptage manuel étant soumis lui aussi à des erreurs d'appréciation (des distances entre particules, des particules faiblement bourgeonnantes...).

4.3 Difficultés rencontrées

4.3.1 Liées au logiciel

Le logiciel Analysis s'est révélé être peu flexible : Ainsi il nous a été impossible par exemple de traduire une image. La fonction standard de translation affichait impérativement une boîte de dialogue à chaque translation (donc peu pratique lorsque la translation est utilisée de manière itérative, comme dans notre programme matlab)... puis nous avons essayé de traduire en utilisant

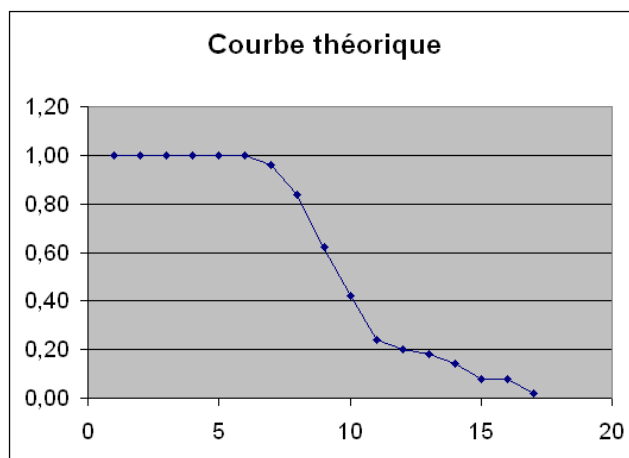


FIG. 12 – Evolution du nombre effectif de particules

des filtres de convolution (sic!) mais cette fonction affichait toujours des boîtes de dialogue à répétition. Le codage “en dur” d’une fonction de translation s’est avérée tellement peu pratique que nous avons alors dû avoir recours à Matlab pour coder la fonction de recadrage des images.

De plus, certaines fonctions sont implémentées et ne sont pas du tout documentées (nous avons appris leur existence grâce à l’enregistreur automatique de macros), et comme ce logiciel semble peu répandu, aucune explication complémentaire n’a pu être trouvée sur Internet.

Certaines fonction prenaient en paramètre des “fichiers” de paramètres, fichiers qui n’étaient pas trouvables sur le disque dur une fois créés, donc il est impossible de distribuer une macro faisant appel à de tels fichiers de configuration.

4.3.2 Liées au matériel

Nous avons été ralentis au début du projet car nous devions disposer d’un ordinateur possédant Analysis, sous Windows, et dont nous ne pouvions avoir accès à distance (Analysis ne fonctionnant pas à distance pour des raisons de sécurité). Nous avons alors disposé d’un ordinateur du GMM qui était extrêmement lent, à savoir qu’il pouvait falloir une dizaine de minutes pour opérer à un banc de filtres pour une expérience (il faut maintenant moins d’une minute pour la même série d’opérations sur un ordinateur que nous avons pu avoir après plusieurs semaines au GBA).

Par ailleurs, il n’existe pas de normalisation des noms de fichiers au sein du laboratoire ce qui se révèle peu pratique lorsque des séries d’images sont nomées de manière très différentes (nom de la manipulation + numéro de zone + numéro d’image + autres numéros de zones.TIFF pour certains, juste nom de la manipulation + numéro d’image pour d’autres...).

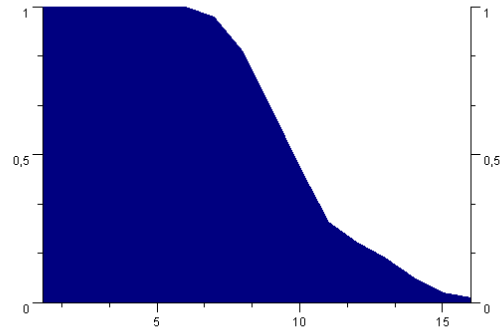


FIG. 13 – Evolution du nombre de particules donné par le script developpé

5 Conclusion

Ce projet, bien que résolu avec un logiciel adapté mais peu maniable, donne des résultats très satisfaisant dans les cas que nous avons rencontrés. Il nécessite toujours une intervention de l'utilisateur (un seuillage, ainsi que l'entrée d'un nombre de diamètres moyen séparant deux particules) mais celui-ci est préférable pour conserver une chaine de traitements qui fonctionnera dans tous les cas de figure.