



HAL
open science

Suivi de mouvements dans une séquence d'images IRM

O. Brossard

► **To cite this version:**

O. Brossard. Suivi de mouvements dans une séquence d'images IRM. Sciences de l'environnement. 2015. hal-02601966

HAL Id: hal-02601966

<https://hal.inrae.fr/hal-02601966v1>

Submitted on 16 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



ESIR
Université de Rennes 1
Campus de Beaulieu
35042 RENNES Cédex



IRSTEA Rennes
17 Avenue de Cucillé
35000 RENNES
Unité TERE

Suivi de mouvements dans une séquence d'images IRM

Sous la responsabilité de :
IRSTEA : Guylaine COLLEWET
ESIR : Régine LE BOUQUIN

Odran BROSSARD
Ingénierie pour la Santé
Deuxième année

Stage du 01/06/2015 au 18/09/2015

Remerciements

Je tiens tout d'abord à remercier le centre de recherche IRSTEA Rennes et plus particulièrement l'équipe IRM-food pour l'accueil qui m'a été réservé et les bonnes conditions de travail dans lesquelles j'ai évolué.

Pour leur soutien ainsi que leur présence je remercie Amélie AUTRET, Stagiaire en informatique pour IRM-food, ainsi que Julien PICAUD, doctorant en IRM quantitative de matière grasse, avec qui j'ai partagé un bureau ainsi que d'agréables moments de convivialité.

Enfin, je remercie ma tutrice de stage Guylaine COLLEWET, chercheur en traitement d'images, pour la supervision de mon travail ainsi que pour l'aide apportée qui ont contribué à la réussite de ce stage.

Sommaire

Introduction.....	5
I. Cadre du stage.....	6
1. L'IRSTEA.....	6
2. Le centre de Rennes.....	6
3. Le logiciel Scilab.....	7
4. La visualisation et l'analyse d'images avec ImageJ.....	7
5. La création de graphiques avec Gnuplot.....	8
II. Estimation du flot optique.....	8
1. Définition.....	8
2. Approche retenue.....	9
3. Représentation du résultat.....	9
III. Évaluation de la méthode.....	10
1. Création d'images de test.....	10
2. Critères d'évaluation.....	13
3. Test sur le paramètre alpha.....	13
4. Test sur le nombre de microcapsules.....	14
5. Test sur l'influence de l'ampleur des déplacements.....	17
IV. Optimisation du résultat.....	18
1. Vers une amélioration du rendu final.....	18
2. Définition des paramètres et implémentation.....	19
3. Application à des séquences entières.....	22
Conclusion.....	24
Bibliographie.....	25
Annexes.....	26

Introduction

L'imagerie par résonance magnétique (IRM) est une technique d'imagerie classiquement utilisée dans le domaine médical. Ses potentialités en font également un outil de choix pour d'autres applications. C'est le cas de la caractérisation des produits agroalimentaires, thème de recherche de l'équipe IRM-food d'IRSTEA au sein de laquelle s'est déroulé ce stage.

En termes de traitement d'images, une des problématiques étudiées est la quantification de mouvements dans une séquence d'images. Ces mouvements sont liés à l'évolution du produit au cours du temps lors d'un procédé de transformation. C'est par exemple le cas d'un pâton de boulangerie (pâte à pain) lors de la pousse. L'objectif de cette étude par IRM est la compréhension des différents mécanismes mis en jeu lors de l'expansion d'un pâton par la visualisation et la quantification des mouvements de matière. Afin d'atteindre cet objectif, des images IRM de pâtons ont été acquises avec une fréquence de l'ordre de la minute. Il s'agit alors d'analyser les séquences d'images afin d'obtenir des cartographies de déplacement de la matière.

Parmi les méthodes répondant à ce problème, les techniques d'estimation du flot optique sont particulièrement intéressantes. Elles permettent de prendre en compte les phénomènes physiques spatio-temporels sous-jacents à la formation de l'image. Elles sont basées sur la minimisation d'une fonction de coût, comprenant à la fois un terme d'attache aux données lié à la modélisation du phénomène et un terme de régularisation permettant d'obtenir une solution dense et spatialement cohérente.

L'objectif de ce stage est la mise au point d'une méthode de quantification de transfert de matière par estimation du flot optique. Le travail s'est basé sur l'utilisation d'une fonction de calcul de flot optique déjà existante. Une validation expérimentale a été menée sur une base d'images réelles déjà acquises. Nous avons évalué en particulier la sensibilité de la méthode aux différents paramètres de réglage.

I. Cadre du stage

1. L'IRSTEA

L'Institut national de Recherche en Sciences et Technologie pour l'Environnement et l'Agriculture (IRSTEA) est un organisme de recherche spécialisé en sciences et technologies pour l'environnement. Il fonde sa stratégie sur une double compétence dans les domaines de l'eau et des agro procédés.

À l'IRSTEA, les approches scientifiques permettent d'étudier les écosystèmes complexes, l'eau, les territoires et la biodiversité et leurs interrelations avec les activités humaines. Elles visent à offrir des solutions concrètes aux questions environnementales et sont fondées sur un partenariat fort avec les industriels et les collectivités territoriales.

Pluridisciplinarité, partenariat, solutions concrètes, sont toutes caractéristiques d'IRSTEA. Ses chercheurs relèvent au quotidien le défi de la compréhension du changement global et de l'appui aux politiques publiques pour la mise en œuvre d'un développement durable écoresponsable.

L'IRSTEA comporte 9 centres en France métropolitaine ainsi qu'un centre en Martinique.

2. Le centre de Rennes

Le centre de Rennes de l'IRSTEA est implanté sur le site de Rennes Beauregard où il regroupe une soixantaine d'agents permanents dont 30 ingénieurs et chercheurs.

En sus de cet effectif propre, il accueille en permanence une vingtaine de doctorants, post-doctorants et ingénieurs-chercheurs étrangers et sur projet chaque année une vingtaine de stagiaires de l'enseignement supérieur.

Ses locaux de 4800 m² abritent des halls permettant des expérimentations sur des pilotes, des prototypes et plusieurs laboratoires.

Son budget personnel, hors personnel permanent, est voisin de 2 millions d'euros.

Il est constitué de deux Unités de Recherche (U.R.) :

- U.R. GERE : Gestion environnementale et traitement biologique des déchets
Responsable : Fabrice BELINE
- U.R. TERE : Technologie des équipements agro-alimentaires
Responsable : François MARIETTE

Les travaux sont menés à la fois en laboratoire et « sur le terrain », où les équipes interviennent tant dans des sites naturels que sur des installations de traitement des ordures

ménagères ou des déjections animales ou encore dans des usines de transformation alimentaire.

La pertinence et la qualité des travaux découlent également de l'insertion des équipes dans un réseau dense de partenariats tant scientifiques qu'institutionnels ou industriels.

L'unité Technologie des équipements agroalimentaires se consacre à l'amélioration et à la maîtrise de la qualité des matières premières et des produits agroalimentaires lors de leur transformation et de leur conservation. Elle développe des connaissances et des méthodologies en mécanique des fluides (aérodynamique) et pour la caractérisation non invasive des produits et procédés (RMN/IRM).

Au sein du thème de recherche ACTA, les recherches portent sur la maîtrise des écoulements et des transferts de masse et de chaleur pour préserver l'hygiène des aliments, maîtrise appliquée aux procédés frigorifiques et ultra propres. L'objectif est de préserver les aliments des conditions physiques de température et de pollution sur et autour des aliments et des chaînes de production.

Le deuxième thème de recherche, IRMfood, a pour objectif d'apporter au secteur agroalimentaire une expertise s'appuyant sur le développement d'outils non invasifs tels que la résonance magnétique nucléaire et l'imagerie de résonance magnétique, pour mieux connaître les mécanismes de mise en place des structures internes et les conséquences sur la qualité et la sûreté des bio-produits. Les filières d'intervention de ce thème de recherche sont la filière laitière, la filière des produits carnés et la filière boulangerie, viennoiserie et pâtisserie. IRM-food est la composante agroalimentaire de la plate-forme Rennaise d'imagerie et de spectroscopie multi modalités PRISM.

3. Le logiciel Scilab

Afin de mener à bien la mission qui m'a été donnée, un logiciel a été mis à ma disposition. Il s'agit de Scilab, un équivalent libre de Matlab avec un langage de programmation similaire. Il fournit un puissant environnement de développement pour les applications scientifiques et de l'ingénierie [7].

L'avantage d'utiliser ce logiciel est que non seulement il est gratuit, mais on peut également trouver une page internet contenant toutes les équivalences des fonctions entre Matlab et Scilab [8].

En revanche, les limites de Scilab interviennent quand on aura besoin de faire de la visualisation des images. En effet, on est vite limité en informations ainsi qu'en élaboration de graphiques. C'est pourquoi j'ai utilisé deux logiciels en compléments : ImageJ et Gnuplot.

4. La visualisation et l'analyse d'images avec ImageJ

ImageJ est un logiciel libre développé par le National Institutes of Health [4]. Il est dédié au traitement d'images. Ses nombreuses fonctionnalités très complètes permettent de faire exactement ce que l'on veut.

En effet, ImageJ peut par exemple lire de très nombreux formats d'images différents (DICOM qui est le format des images IRM, raw, PNG, etc.). Il permet également de faire des piles d'images (« images 3D »). Il y a également une multitude d'outils d'analyse qui sont à notre disposition : opérations, segmentation, histogramme, profils, mesures, etc.

5. La création de graphiques avec Gnuplot

Une fois que nous aurons calculé le flot optique entre deux images, nous verrons plus tard dans le rapport que nous avons choisi de représenter le résultat sous forme de champ de vecteurs. Gnuplot sera le logiciel nous permettant de créer ces graphes résultats.

Gnuplot est un logiciel libre permettant de tracer des courbes 2D ou 3D [1]. Il fonctionne par l'intermédiaire de commandes simples. Il offre la possibilité de tracer des fichiers de points. Différentes options sont possibles comme les styles de tracés, l'intervalle de définition d'une fonction, ou encore l'application d'une palette de couleurs permettant de faciliter la visualisation du résultat.

II. Estimation du flot optique

Dans cette partie, on s'attachera à détailler le choix de la méthode de calcul de flot optique retenue, ainsi que son implémentation, de manière à comprendre comment seront générés les résultats par la suite.

1. Définition

Le flot optique est un champ de vitesse mesuré à partir des variations de la luminance [6]. De façon plus concrète, considérons I_1 et I_2 , deux images consécutives d'une même scène prises respectivement aux instants t et $t + 1$. Le flot optique entre I_1 et I_2 est représenté par deux matrices, que l'on notera u et v , l'une codant pour le déplacement horizontal (u) et l'autre pour le déplacement vertical (v) en pixels.

De nombreuses approches existent pour calculer ce déplacement. Pour la plupart, elles reposent sur l'hypothèse de conservation de l'intensité lumineuse entre deux images successives qui s'écrit :

$$I_1(i, j) - I_2(i + u, j + v) = 0$$

Cette hypothèse est insuffisante pour déterminer le flot car on ne dispose que d'une équation pour deux inconnues. Le problème est dit « mal posé ». De nombreuses solutions

ont été proposées pour résoudre ce problème et peuvent être classées en trois catégories : il existe des solutions basées sur une représentation fréquentielle des images, des méthodes basées sur la corrélation et enfin les approches variationnelles.

2. Approche retenue

Dans le cadre de ce stage, nous avons utilisé le code Scilab développé par Julien MARZAT [6]. Ce dernier a fait cette fonction durant son stage de fin d'études portant sur l'estimation du flot optique, à l'INRIA, pendant l'année 2008. Il est à noter que notre cadre applicatif diffère du sien : nous cherchons ici à comprendre les mouvements complexes d'un pâtton de pain alors que L. MARZAT cherchait à calculer la perception de l'environnement pour des véhicules totalement automatisés.

La méthode implémentée est de type variationnelle. Elle a été proposée par LUCAS et KANADE [5]. Pour résoudre le problème « mal posé », ils ont ajouté à l'hypothèse de conservation de la luminance une hypothèse de constance du flot sur un voisinage, encore appelé patch. De plus, afin d'apporter de la robustesse à la méthode, une technique dite de régularisation spatiale est utilisée. Le poids de cette régularisation sur la solution est réglée par un scalaire, alpha.

L'implémentation est de type pyramidale, c'est-à-dire que l'image est sous-échantillonnée à différents niveaux. Si on note N le nombre de sous-échantillonnages, le calcul du flot optique est tout d'abord réalisé au niveau de sous-échantillonnage N . Le flot optique sert d'initialisation au calcul sur le niveau de sous-échantillonnage $N - 1$ et ainsi de suite jusqu'au niveau de sous-échantillonnage 1 qui correspond à l'image initiale.

Enfin, l'implémentation est itérative : l'algorithme déplace une des deux images suivant le flot optique calculé. Le flot optique est de nouveau estimé et ajouté au flot précédent, ce qui permet de raffiner la solution. Cela se fait sur un nombre fixé d'itérations.

Ainsi, l'algorithme est régi par 4 paramètres : la taille du patch, le paramètre de régularisation alpha, le nombre de niveaux de sous-échantillonnage et enfin le nombre d'itérations. D'après la validation de ces paramètres effectuée par J. MARZAT, nous avons utilisé 5 itérations avec 4 niveaux de sous-échantillonnage. La taille du patch est variable (5, 7 ou 9 pixels) selon la taille des images utilisées ainsi que la taille de l'objet étudié dans ces images. Pour le paramètre alpha, nous avons effectué la validation nous-mêmes, les résultats seront présentés plus loin dans ce document.

Dans la suite de ce rapport, chaque mention d'un calcul de flot optique entre deux images ou entre toutes les images d'une même séquence fera référence à cette méthode de calcul.

3. Représentation du résultat

Comme nous l'avons vu précédemment, notre résultat est un ensemble de deux matrices. La

première, u , représente le déplacement horizontal, alors que la seconde, v , représente le déplacement vertical. Là encore, nous avons à notre disposition une fonction développée par J. MARZAT. Cette fonction prenait en entrée les deux matrices de déplacements u et v . Cette fonction nous donne en sortie une image colorée où la couleur correspond à une orientation du vecteur représenté mais aussi sa norme. L'orientation et la norme du vecteur représenté est calculé à partir des matrices u (composante horizontale) et v (composante verticale) et est représentée par le champ de vecteurs que l'on retrouve ci-dessous. Voici la correspondance entre ces deux éléments :

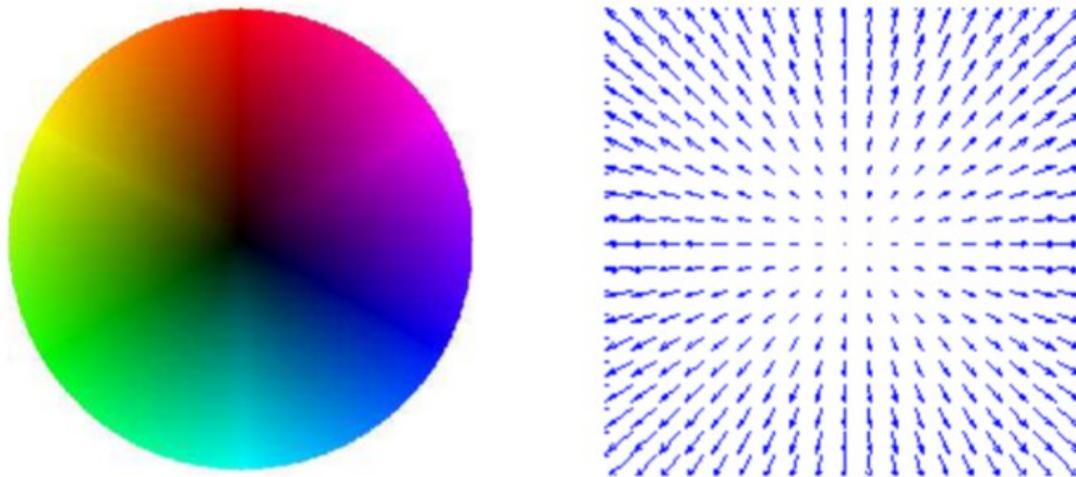


Illustration 1: Code couleur de représentation du résultat

Cette représentation nous donnera donc une image colorée en sortie, et il faut nécessairement avoir le code couleur sous les yeux pour pouvoir interpréter les résultats. L'avantage de ce type de représentation est qu'il va nous permettre directement de voir des zones entières de l'image ayant tendance à avoir un mouvement similaire. Le principal inconvénient en revanche reste le manque de précision. Comme on le voit sur la figure ci-dessus, il est difficile par exemple de faire la différence entre deux nuances de vert, alors qu'elles n'auront ni la même orientation, ni la même norme. Nous verrons par la suite de ce rapport que nous avons changé de représentation finale.

III. Évaluation de la méthode

Une fois l'algorithme clairement défini, il nous faut évaluer cette méthode afin de connaître la marge d'erreur. Nous allons ainsi pouvoir observer les réglages précis à utiliser pour un fonctionnement optimal de l'algorithme.

1. Création d'images de test

Pour pouvoir réaliser nos tests, nous allons avoir besoin d'images de référence, notées I1 et I2. Nous aurons également besoin de connaître le flot optique entre ces deux images, qui sera appelé flot réel, et noté (u_r, v_r) .

Voici les images que nous avons choisi d'utiliser comme images de référence :

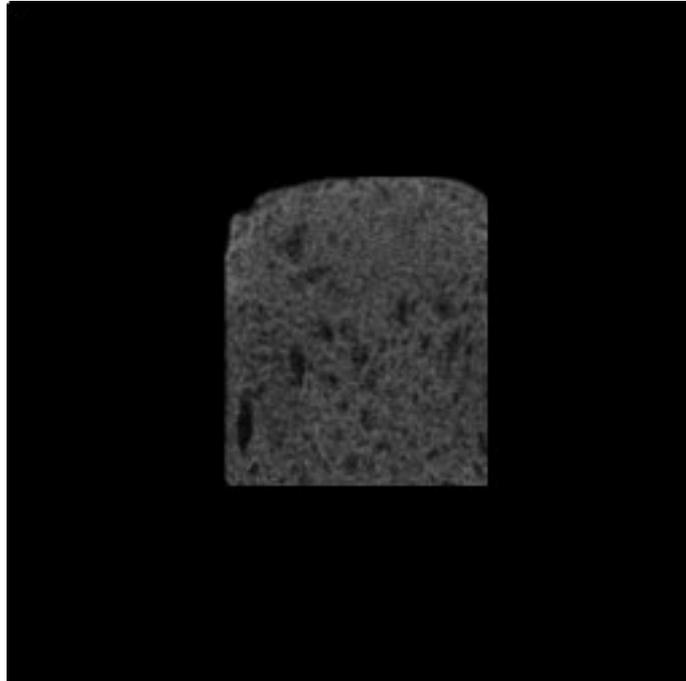


Illustration 2: Image initiale I1 à l'instant t

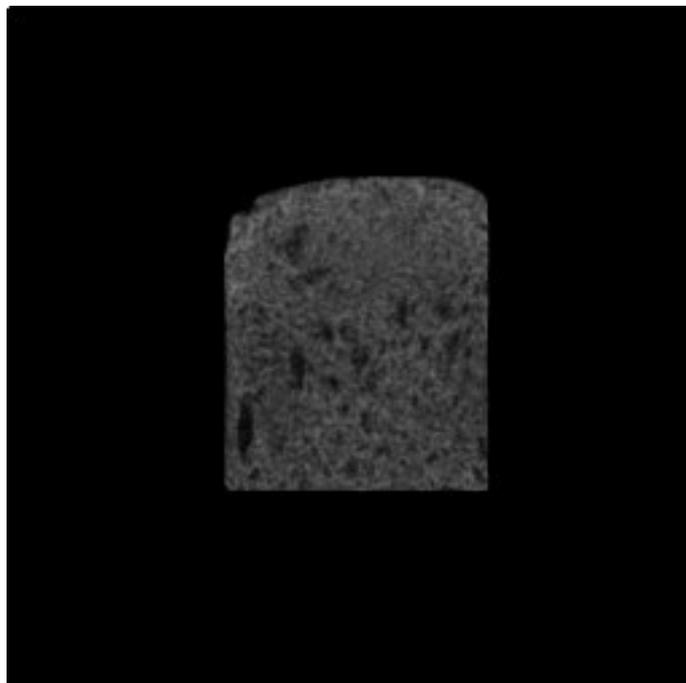


Illustration 3: Image initiale I2 à l'instant t + 1

Si l'on appelle $I1'$ et $I2'$ nos deux images de synthèse, on commence par définir $I1'$ comme étant identique à $I1$. Pour calculer $I2'$, nous allons appliquer le flot optique (u_r, v_r) à l'image $I1'$. $I2'$ sera donc la transformée de $I1'$ par le flot réel (u_r, v_r) .

Voici le schéma résumant ce processus de création d'images de synthèse :

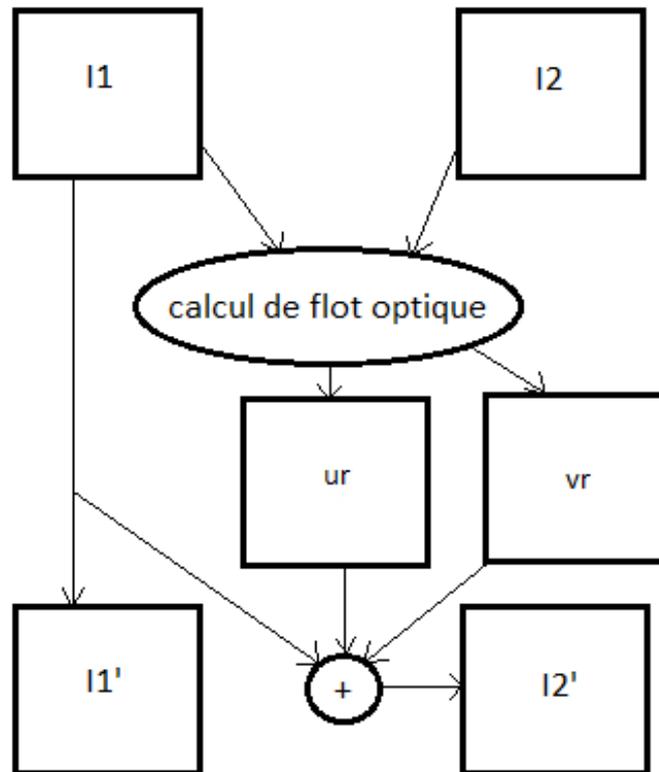


Illustration 4: Principe de la création d'images de synthèse

À noter que dans ce schéma, l'opérateur « + » fait référence à une opération de transformation, et non pas à une opération d'addition.

Pour calculer $I2'$, la transformée de $I1'$ par (u_r, v_r) , nous partons du constat que les déplacements calculés (u_r, v_r) ne sont pas des valeurs entières. Ainsi, on ne peut pas déplacer aisément les pixels, nous devons utiliser une astuce de calcul. La valeur du déplacement étant non entière, on cherche les pixels voisins les plus proches du pixel d'arrivée idéal et on leur affecte une proportion du pixel initial dépendant de la valeur du déplacement. Cela nous permet donc de calculer l'image finale en fonction de l'image initiale et de cette pondération.

Par exemple, pour un pixel de $I1'$ à la position (i, j) d'intensité k et ayant un déplacement horizontal de 3,47 pixels, on calcule de cette façon : dans l'image $I2'$, au pixel à la position $(i + 3, j)$ on affecte la valeur $0,47 * I1'(i, j)$. Le reste $(0,53 * I1'(i, j))$ sera donc affecté au pixel à la position $((i + 3) + 1, j)$.

2. Critères d'évaluation

Maintenant que nous avons nos deux images de synthèse I1' et I2', nous allons calculer le flot optique entre ces deux images, que nous notons (u_c, v_c) , le flot calculé. L'évaluation de nos résultats repose principalement sur deux notions essentielles : la norme des vecteurs calculés ainsi que leur angle. On souhaite évaluer l'écart entre (u_r, v_r) et (u_c, v_c) . On va donc définir deux critères d'évaluation, qui seront l'erreur angulaire ainsi que l'erreur en norme.

Voici les formules que nous avons choisi d'utiliser pour effectuer ces calculs :

$$err_{ang} = \frac{1}{N \cdot M} \cdot \sum_{i=1}^N \sum_{j=1}^M \arccos\left(\frac{u_r(i, j)u_c(i, j) + v_r(i, j)v_c(i, j) + 1}{\sqrt{(u_r^2(i, j) + v_r^2(i, j) + 1)(u_c^2(i, j) + v_c^2(i, j) + 1)}}\right) ,$$

$$err_{norm} = \frac{1}{N \cdot M} \cdot \sum_{i=1}^N \sum_{j=1}^M \sqrt{((u_r(i, j) - u_c(i, j))^2 + (v_r(i, j) - v_c(i, j))^2)} ,$$

en considérant (u_c, v_c) le flot calculé, (u_r, v_r) le flot réel, N le nombre de lignes des matrices u_r, v_r, u_c, v_c et M leur nombre de colonnes.

On va donc enregistrer les résultats de ces opérations pour pouvoir aisément faire des comparaisons. Nous avons choisi d'utiliser le logiciel Excel (version 2010) de Microsoft, car il permet de façon simple de stocker des données dans un tableau, puis de les utiliser pour générer des graphiques ensuite.

3. Test sur le paramètre alpha

Le paramètre alpha de l'algorithme est le coefficient de régularisation. Nous savons donc que sa valeur doit être proche de 0. Pour cela, nous avons choisi de tester comme valeurs de ce paramètre : $1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}$.

Afin de tester ce paramètre, nous avons calculé 9 flots optiques différents (les paramètres n'étaient pas les mêmes) entre nos deux images initiales (nous avons donc 9 (u_r, v_r)). Cela nous a permis d'obtenir nos couples d'images de synthèse. Ensuite, nous avons calculé pour chaque couple d'images de synthèse le flot optique entre elles, et ce pour les différentes valeurs de alpha à tester.

Ainsi, on obtient donc 9 couple (u_c, v_c) pour chacun des 9 (u_r, v_r) . Après collecte des données, nous les avons retranscrites dans un fichier Excel afin de générer les graphes suivants :

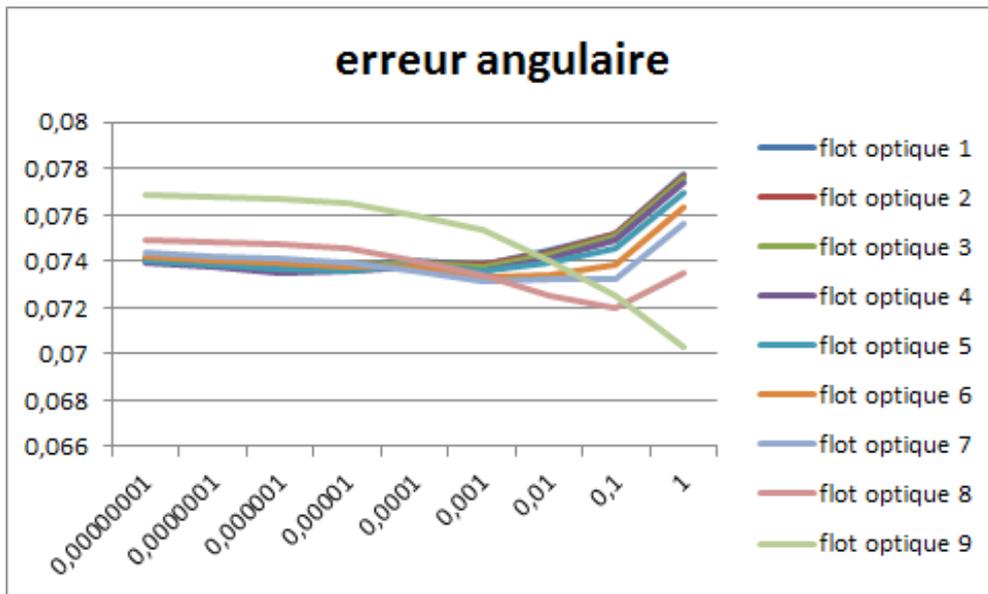


Illustration 5: Erreur angulaire en fonction du paramètre alpha pour les 9 flots optiques considérés

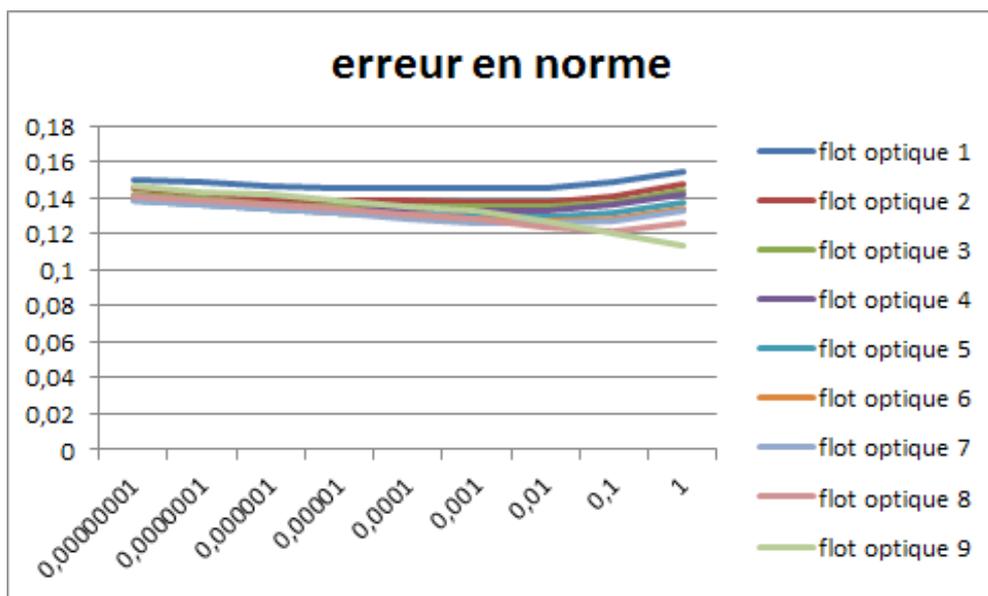


Illustration 6: Erreur en norme en fonction du paramètre alpha pour les 9 flots optiques considérés

C'est à partir de ces deux graphiques que l'on a pu déterminer une valeur optimale pour le paramètre alpha. La valeur ainsi sélectionnée est donc 10^{-2} , c'est ce que l'on utilisera donc pour la suite du travail.

4. Test sur le nombre de microcapsules

Avant que soit mise en place la solution que j'ai testée utilisant le flot optique, il était déjà possible d'observer la poussée des pâtons de pain. Cette méthode a été développée par Tiphaine LUCAS, chercheur de l'équipe IRM-Food d'IRSTEA. Cela consiste en l'utilisation des microcapsules contenant un liquide différenciable de la pâte à pain lors de l'observation du pâton sous image IRM. Ces microcapsules sont placées à des endroits stratégiques permettant de visualiser l'expansion de ces endroits. En effet, en prenant des images aux instants t et $t + \Delta t$, on pouvait observer où était la microcapsule sur les deux images, ce qui donnait un aperçu du déplacement. Voici un exemple d'image avec des microcapsules :

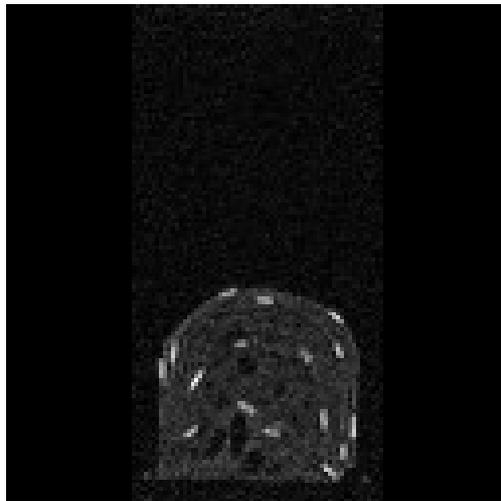


Illustration 7: Exemple d'image de pâton avec des microcapsules

Insérer les microcapsules dans le pâton est relativement long, en revanche, nous avons voulu vérifier s'il était utile d'en mettre quand même dans la mesure où cela peut éventuellement créer des conditions favorables pour l'utilisation de l'algorithme. L'hypothèse que nous avons faite est que le fait d'avoir des points remarquables dans l'image pourrait aider à calculer le déplacement de façon fiable.

Pour tester cette hypothèse, nous avons donc rajouté manuellement des microcapsules aux images initiales. Il a été déterminé que ces microcapsules devaient respecter au mieux la taille et la luminance des vraies microcapsules rajoutées physiquement dans le pâton. C'est pourquoi nous avons convenu que ces microcapsules seraient de taille $2 * 2$ (en pixels) et auraient une valeur (en niveaux de gris) de 180.

Pour tester l'influence de la présence de microcapsules, nous avons décidé d'en ajouter de 0 à 9 à l'image initiale I_1 . Cela nous a donné I_1' . Ensuite, nous avons calculé I_2' de la même

façon que précédemment, soit en effectuant la transformée de I1' par le flot optique réel (u_r, v_r). Nous avons calculé le flot optique calculé (u_c, v_c) entre I1' et I2'. Puis, nous avons calculé les deux erreurs (erreur angulaire et erreur en norme) entre le flot optique calculé et le flot optique réel. Puis nous avons réitéré ces étapes pour chaque nombre de microcapsules à insérer. Voici les graphiques faits sous Excel que nous avons obtenus en résultats :

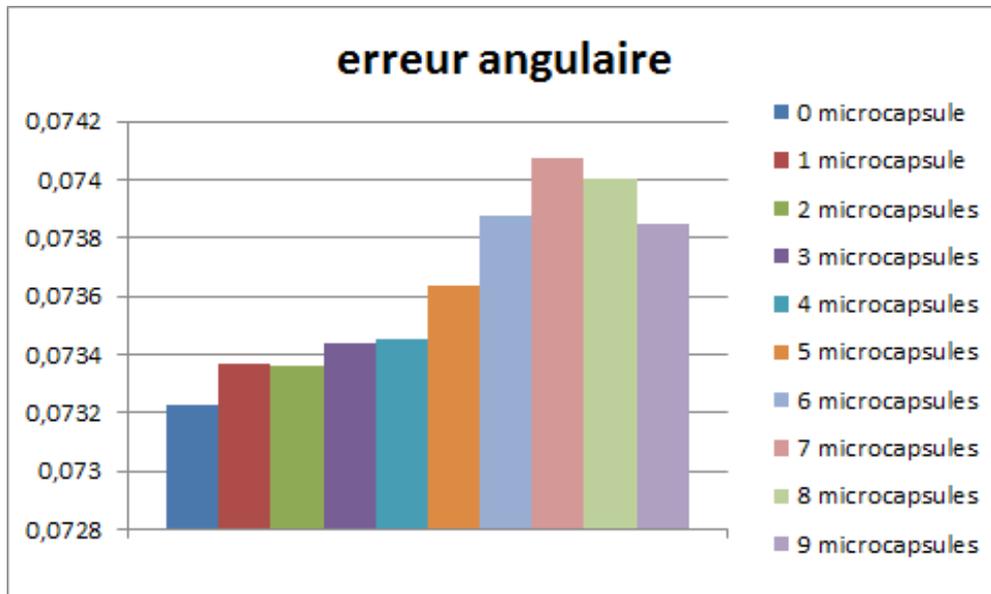


Illustration 8: Erreur angulaire en fonction du nombre de microcapsules

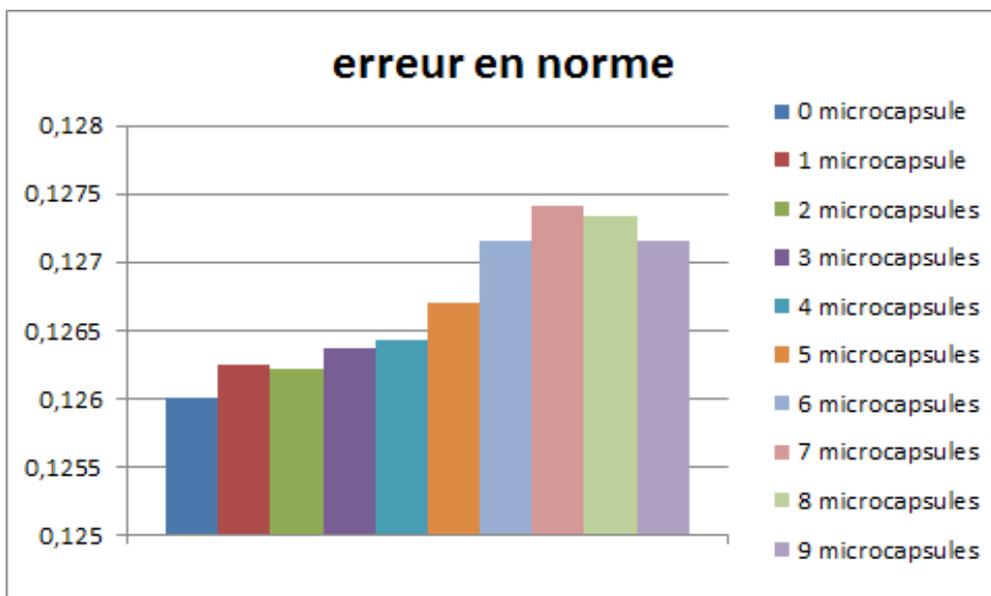


Illustration 9: Erreur en norme en fonction du nombre de microcapsules

Comme le montrent ces graphiques, la valeur de l'erreur semble augmenter en même temps que le nombre de microcapsules. Il est vrai qu'à partir de 8 microcapsules, l'erreur commence à diminuer, donc il pourrait être bon de voir ce qu'il en est pour plus de 10 microcapsules. Cependant, en pratique les microcapsules prennent du temps à être placées, donc on va choisir de garder le nombre de microcapsules assurant les erreurs les plus faibles dans ces graphiques. Nous en concluons donc que nous n'utiliserons pas de microcapsules pour la suite de ce travail.

5. Test sur l'influence de l'ampleur des déplacements

L'expansion d'un pignon de pain n'est pas homogène au cours du temps. En effet, si le déplacement peut sembler peu perceptible à l'œil nu entre deux images consécutives, il se peut qu'entre deux autres images il soit flagrant. L'intérêt de cette partie va donc être d'observer l'influence de l'ampleur du déplacement sur les erreurs.

Pour cela, nous allons procéder de façon simple, c'est-à-dire que nous allons multiplier le flot réel (u_r, v_r) entre les deux images initiales I1 et I2 par un coefficient k . Cela va donc nous donner un nouveau flot $(k * u_r, k * v_r)$. L'image de synthèse I1' est une nouvelle fois égale à I1. L'image de synthèse I2' est la transformée de I1' par ce nouveau flot $(k * u_r, k * v_r)$.

En pratique, nous n'allons pas pouvoir tester avec de grandes valeurs de k , puisque le pignon devient plus grand que la taille de l'image. Le calcul de flot optique entre I1' et I2' serait alors biaisé. C'est pourquoi nous avons choisi de faire varier k de 1 à 4. En calculant une nouvelle fois l'erreur entre (u_c, v_c) le flot calculé et $(k * u_r, k * v_r)$ le flot réel, nous obtenons ces graphiques de résultats :

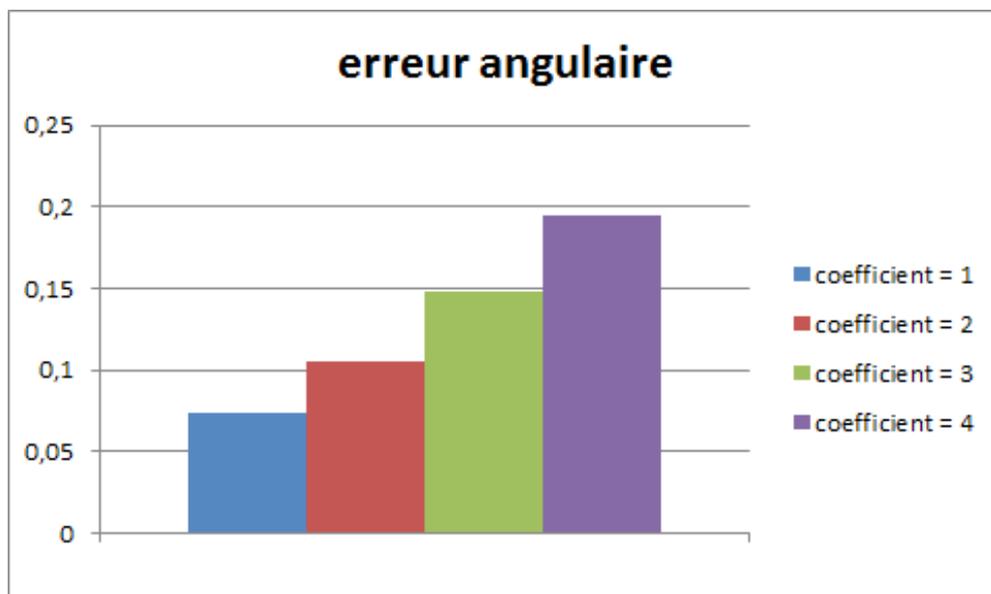


Illustration 10: Influence de l'ampleur des déplacements : erreur angulaire

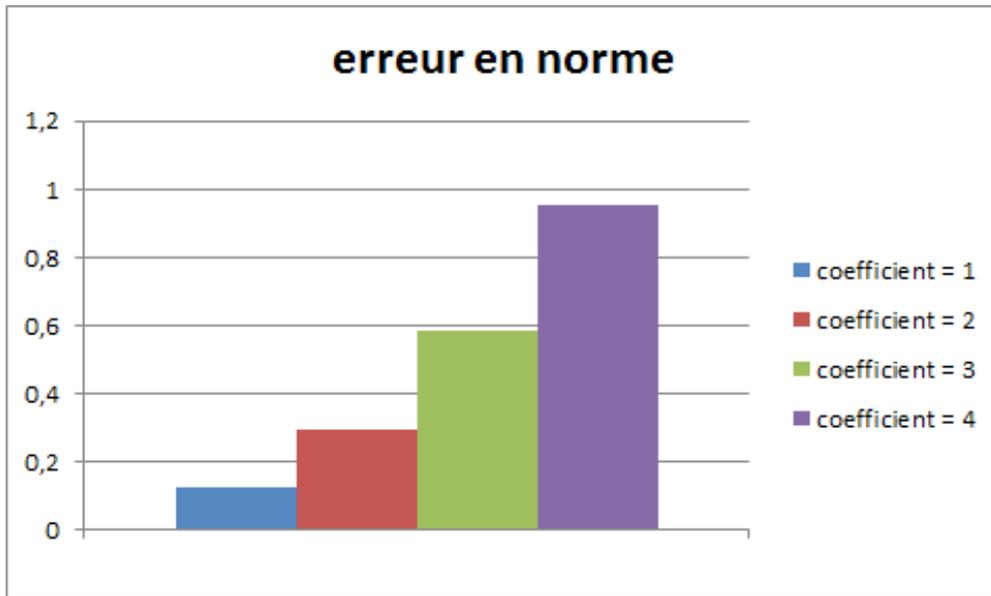


Illustration 11: Influence de l'ampleur des déplacements : erreur en norme

Cette fois-ci, le résultat est net et l'interprétation immédiate : plus le flot optique contient des déplacements importants, plus les erreurs sont élevées. Nous savons donc maintenant que notre algorithme est plus fiable pour des déplacements de petites envergures. Cette conclusion peut amener à de nouveaux changements, par exemple au fait de diminuer le temps d'acquisition sur l'IRM pour éviter de traiter des déplacements trop importants.

IV. Optimisation du résultat

Maintenant que nous avons testé quelques-unes des caractéristiques principales de notre algorithme, nous allons chercher à avoir un rendu graphique rapidement exploitable.

1. Vers une amélioration du rendu final

Jusqu'ici, nous n'avons pas eu recours à la visualisation du résultat, ce dernier étant une image utilisant le flot optique (u, v) calculé. Comme détaillé dans le point II.4., la représentation se fait grâce à un code couleur. Cependant, l'inconvénient majeur de cette méthode est qu'on doit constamment avoir le code couleur à portée de main pour pouvoir interpréter. Un autre inconvénient est que cette méthode nous délivre des images intégralement colorée, ce qui fait que si jamais on a une information sur une petite zone, elle peut se retrouver noyée dans une grande zone de couleur et on pourra pas interpréter correctement le résultat.

Il a donc fallu réfléchir à un nouveau moyen de représenter notre flot (u, v) . Nous avons utilisé le logiciel libre Gnuplot (qui a été présenté en I.5.). Ce logiciel est spécialisé dans le tracé de courbes ainsi que de champs de vecteurs. C'est cette dernière option qui nous intéresse particulièrement. En effet, à titre de rappel, lorsque l'on parle du flot optique (u, v) , la matrice u correspond au déplacement horizontal et la matrice v correspond au déplacement vertical. Donc pour chaque couple de coordonnées (x, y) on a bien $u(x, y)$ qui correspond à la valeur des abscisses d'un vecteur et $v(x, y)$ qui correspond à la valeur des ordonnées.

Nous avons convenu de sauvegarder notre résultat au format PNG (Portable Network Graphics), puisque c'est un format facile à ouvrir, même sans l'aide de ImageJ si besoin.

2. Définition des paramètres et implémentation

Pour fonctionner correctement, le logiciel Gnuplot va prendre en entrée un fichier de données. Il va donc falloir créer ce fichier. Ce fichier va regrouper 4 informations : l'indice des abscisses x , l'indice des ordonnées y , la valeur du déplacement horizontal $u(x, y)$ et la valeur du déplacement vertical $v(x, y)$. À chaque nouveau quatuor d'informations, on revient à la ligne, ainsi chaque ligne du notre fichier de données va coder pour un pixel. L'annexe II présente une capture d'écran d'une partie de ce fichier.

Il est important de noter que la majorité des déplacements, dans notre cas d'étude, sont en moyenne de 2 pixels. Ainsi, nous allons multiplier nos flots optiques par un coefficient k (valeur arbitraire : 4) afin de pouvoir mieux les visualiser. Sinon les vecteurs représentant les déplacements n'auraient pas une norme suffisamment grande pour pouvoir les interpréter.

Les images utilisées pour le calcul du flot optique sont une nouvelle fois les images de référence que l'on peut retrouver en illustration 2 et illustration 3 de ce document. Si l'on regarde ce que cela nous donne pour le moment, voici le résultat que nous trouvons en sortie de calcul d'un flot optique :

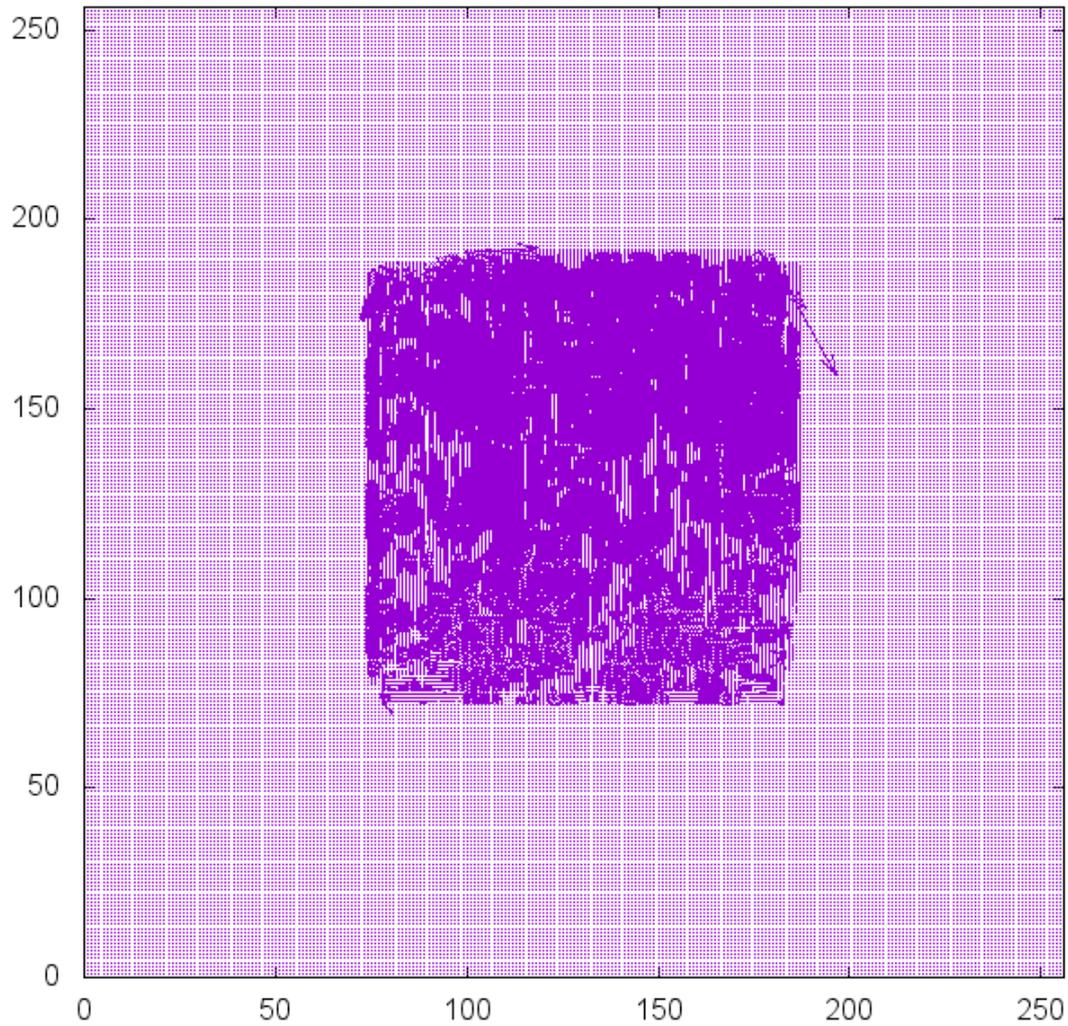


Illustration 12: Première version de la représentation graphique implémentée avec Gnuplot

Comme nous le montre cette image, il y a tellement de vecteurs de représentés que l'information est noyée. Nous allons donc chercher à pallier ce problème. Pour cela, une fonctionnalité très intéressante du logiciel Gnuplot réside dans l'attribut `every` [3]. Il permet de limiter le nombre de vecteurs que l'on souhaite afficher sur notre graphe pour ne pas le surcharger. Ainsi, nous allons choisir de définir un rang de sélection des vecteurs à afficher. On peut par exemple afficher toutes les 4 colonnes ainsi que toutes les 4 lignes. Cela donne ce résultat :

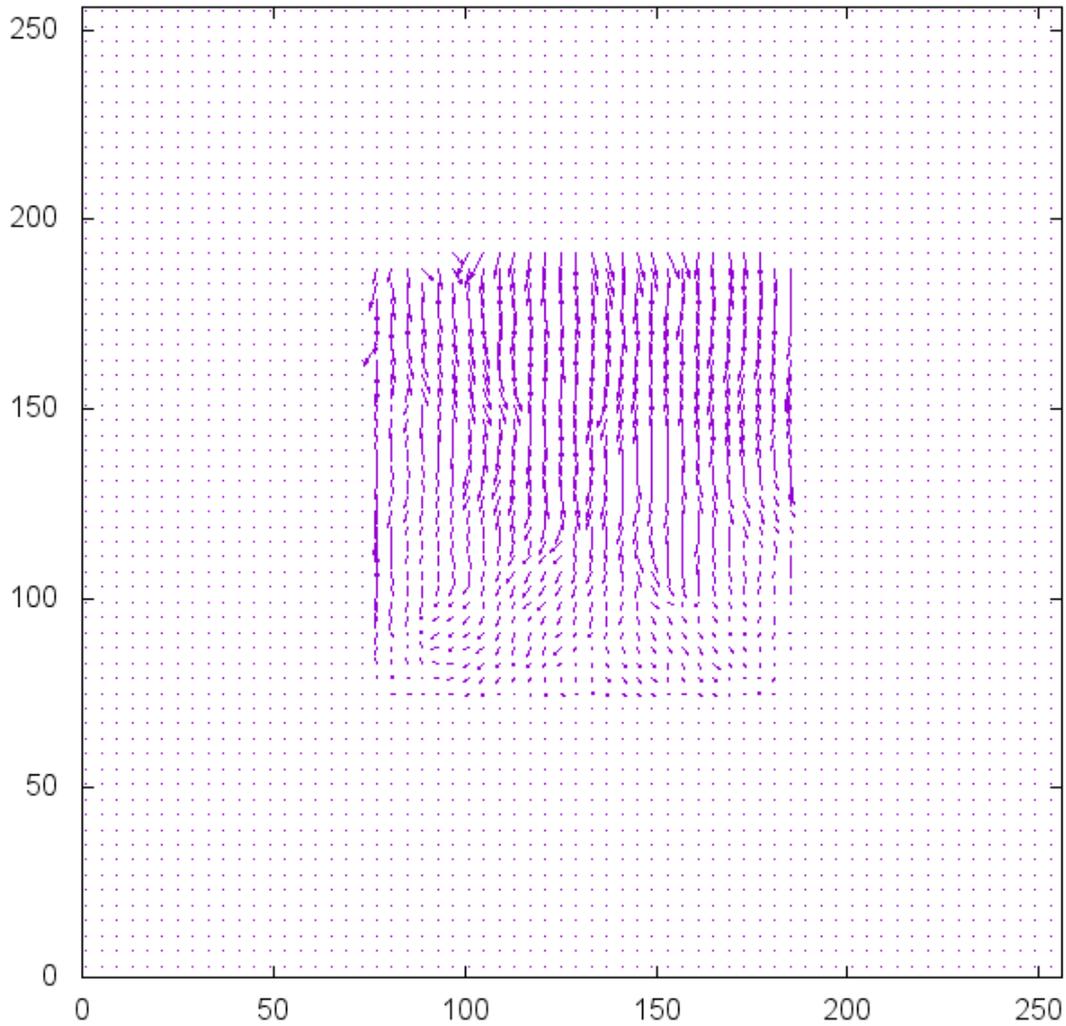


Illustration 13: Deuxième version de la représentation graphique implémentée sous Gnuplot

Maintenant, nous avons un résultat exploitable, mais il subsiste un dernier inconvénient : bien que l'orientation des vecteurs soit immédiatement visible, leur norme, quant à elle, manque de précision. C'est pourquoi nous avons convenu d'utiliser une palette de couleurs. Cette dernière va nous permettre de colorer les vecteurs selon un code couleur bien précis. Nous avons décidé donc que le paramètre qui coderait pour la couleur serait la norme. Ainsi, voici ce que l'on obtient comme rendu à ce stade :

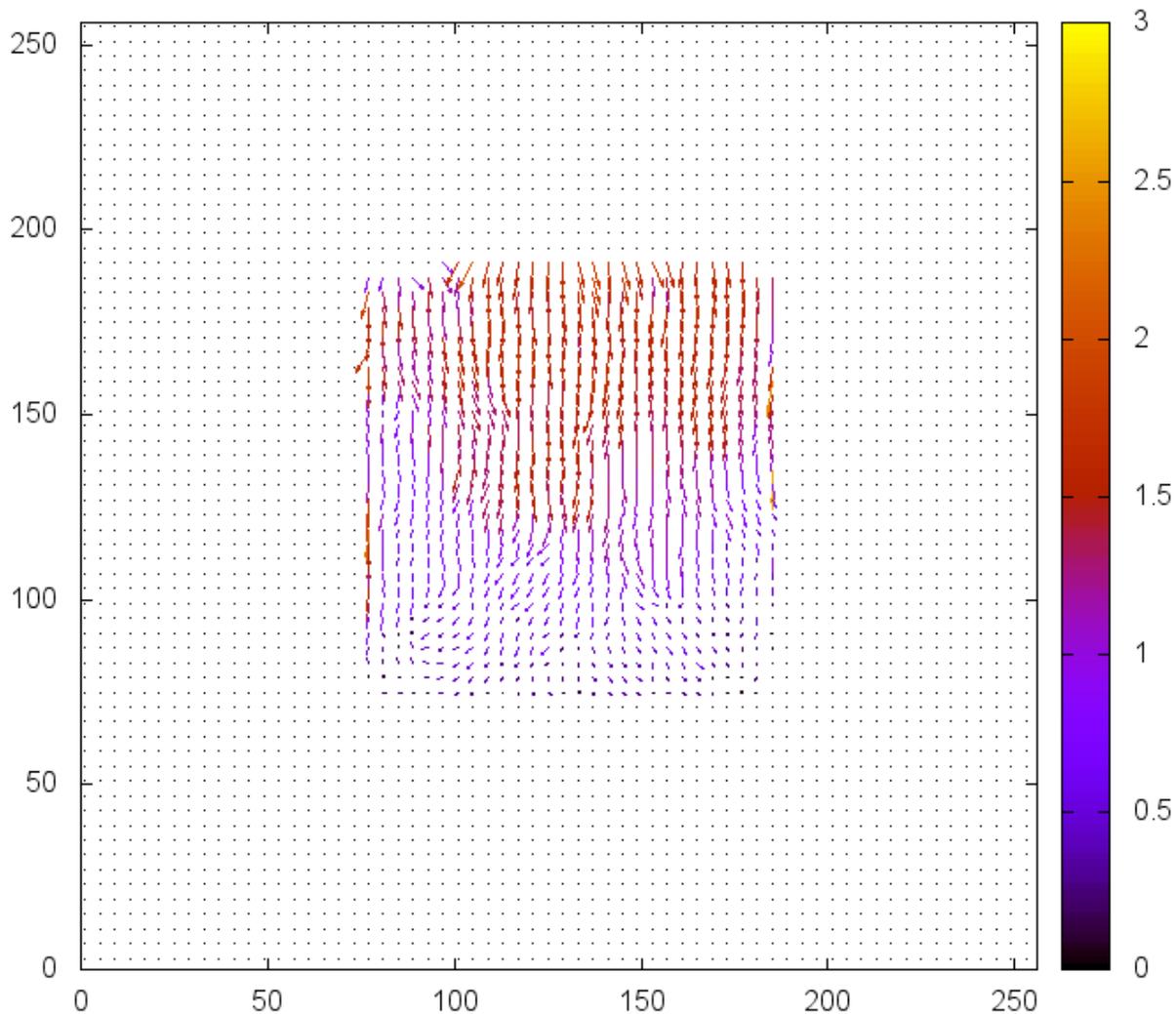


Illustration 14: Troisième version de la représentation graphique implémentée sous Gnuplot

Comparé aux deux versions précédentes du résultat obtenu, celle-ci apparaît comme étant la plus exploitable. Nous utiliserons donc cette représentation graphique du résultat par la suite.

3. Application à des séquences entières

Maintenant que nous avons vu que le rendu graphique est correct, nous allons chercher à étendre cette méthode pour pouvoir l'appliquer à des séquences entières d'images. Ce que l'on appelle séquence sera par exemple une suite d'images au format DICOM (.dcm). Afin de simplifier, nous allons séparer le calcul du flot optique des images consécutives et la création des graphes résultats. Cela permettra entre autres de pouvoir modifier les options des graphes sans avoir à recalculer le flot optique à chaque fois. L'intérêt réside dans le temps d'exécution : le calcul du flot optique est long alors que la production des graphes ne l'est pas.

Ainsi, nous avons un premier programme Scilab qui charge toute la séquence d'images, et qui crée les flots optiques des images consécutives, puis les enregistre dans des fichiers.

Ensuite, nous avons un second programme Scilab qui charge ces flots, et qui construit les graphes correspondant, grâce à Gnuplot. Pour cela, ce script Scilab crée tout d'abord le fichier de données présenté en IV.2. (une ébauche de ce fichier est disponible en annexe II). Ensuite, il crée le fichier de commandes Gnuplot, contenant toutes les commandes disponibles (le détail de ce fichier est disponible en annexe III). Enfin, il est possible de lancer Gnuplot depuis Scilab grâce à des commandes DOS. C'est ce que l'on va utiliser pour lancer Gnuplot et lui indiquer de lancer notre fichier de commandes. Ainsi cela va créer notre graphe de sortie. Pour pouvoir l'appliquer à une séquence entière d'images, il suffit de réitérer ces opérations pour chaque nouveau flot optique calculé. Le détail des résultats sur une séquence de 3 images est disponible en annexe IV.

Conclusion

Les principales difficultés rencontrées au cours de ce projet ont été de deux sortes. Tout d'abord, le temps de calcul du flot optique entre deux images pouvait être très long. Même en optimisant les paramètres (taille du patch, taille de la fenêtre de calcul, nombre d'itérations) cela pouvait aller jusqu'à 15 minutes. Ce problème a été pris en charge très rapidement, puisque l'on m'a permis de changer d'ordinateur dès les premiers jours de mon stage, afin d'avoir un ordinateur plus rapide et plus puissant en terme de calcul.

L'autre type de difficulté rencontrée a été pour l'utilisation du logiciel Gnuplot, utilisés pour la création de graphes. En effet, il dispose d'une documentation relativement mince, et il m'a fallu environ une semaine pour maîtriser la création des graphes ainsi que les différentes options possibles.

Une fois ces difficultés surmontées, le programme de calcul de flot optique a été clairement défini et implémenté sur Scilab. Cela nous a permis de mener des tests sur les valeurs des paramètres du programme, qui ont été concluants puisque nous avons pu déterminer les paramètres optimums pour notre algorithme.

En définitive, l'objectif initial a bien été atteint : de l'étude bibliographique jusqu'à la mise en place du programme Scilab. Nous avons même eu l'occasion d'aller encore plus en avant en optimisant le rendu final. Ainsi, la clarté de nos résultats reste toujours un des principaux atouts de ce programme.

Bibliographie

- [1] ENSI Bourges. GNUPLOT, logiciel de tracé de fonctions et de graphismes de données.
enseignement.ensi-bourges.fr/cours/GNUPLOT/document/gnuplot.html
- [2] GASTEBOIS, S (2009). Application du Flot Optique au suivi de la pousse d'un pàton par IRM, stagiaire UR1 - CEMAGREF (40 pages)
- [3] KIELBASIEWICZ, N. Aide-mémoire Gnuplot 4.0, 20 juin 2008.
perso.ensta-paristech.fr/~kielbasi/docs/gnuplot.pdf
- [4] LOSSON, M. Introduction à ImageJ.
master-ivi.univ-lille1.fr/fichiers/Cours/lo_imageJ_intro.pdf
- [5] LUCAS, B.D., KANADE, T. An Iterative Image Registration Technique with an Application to Stereo Vision In IJCAI81, pages 674 – 679, 1981.
- [6] MARZAT, J (2008). Estimation temps réel du Flot Optique, stagiaire ENSEM – INRIA (79 pages).
- [7] Scilab Enterprises. À propos de Scilab. www.scilab.org/fr/scilab/about
- [8] Scilab Enterprises. Matlab-Scilab equivalents.
https://help.scilab.org/doc/5.3.3/en_US/section_36184e52ee88ad558380be4e92d3de21.html

Annexes

I – Table des illustrations.....	27
II – Fichier de données pour Gnuplot.....	28
III – Fichier de commandes pour Gnuplot.....	29
IV – Résultat du traitement sur une séquence d'images.....	30

I – Table des illustrations

Illustration 1: Code couleur de représentation du résultat.....	10
Illustration 2: Image initiale I1 à l'instant t.....	11
Illustration 3: Image initiale I2 à l'instant $t + 1$	11
Illustration 4: Principe de la création d'images de synthèse.....	12
Illustration 5: Erreur angulaire en fonction du paramètre alpha pour les 9 flots optiques considérés.....	14
Illustration 6: Erreur en norme en fonction du paramètre alpha pour les 9 flots optiques considérés.....	14
Illustration 7: Exemple d'image de pâton avec des microcapsules.....	15
Illustration 8: Erreur angulaire en fonction du nombre de microcapsules.....	16
Illustration 9: Erreur en norme en fonction du nombre de microcapsules.....	16
Illustration 10: Influence de l'ampleur des déplacements : erreur angulaire.....	17
Illustration 11: Influence de l'ampleur des déplacements : erreur en norme.....	18
Illustration 12: Première version de la représentation graphique implémentée avec Gnuplot	20
Illustration 13: Deuxième version de la représentation graphique implémentée sous Gnuplot	21
Illustration 14: Troisième version de la représentation graphique implémentée sous Gnuplot	22

II – Fichier de données pour Gnuplot

```
35.000000 45.000000 -0.416022 0.182943
35.000000 46.000000 -0.429162 0.194672
35.000000 47.000000 -0.435234 0.203688
35.000000 48.000000 -0.434237 0.209992
35.000000 49.000000 -0.426173 0.213584
35.000000 50.000000 -0.418140 0.213608
35.000000 51.000000 -0.410140 0.210065
35.000000 52.000000 -0.402173 0.202954
35.000000 53.000000 -0.394237 0.192275
35.000000 54.000000 -0.386369 0.177773
35.000000 55.000000 -0.378569 0.159446
35.000000 56.000000 -0.370836 0.137295
35.000000 57.000000 -0.363170 0.111319
35.000000 58.000000 -0.355375 0.076191
35.000000 59.000000 -0.347450 0.031908
35.000000 60.000000 -0.339394 -0.021528
35.000000 61.000000 -0.331209 -0.084118
35.000000 62.000000 -0.322678 -0.155418
35.000000 63.000000 -0.313801 -0.235427
35.000000 64.000000 -0.304578 -0.324146
35.000000 65.000000 -0.295010 -0.421575
35.000000 66.000000 -0.286253 -0.512562
35.000000 67.000000 -0.278309 -0.597107
35.000000 68.000000 -0.271177 -0.675211
35.000000 69.000000 -0.264858 -0.746872
35.000000 70.000000 -0.259375 -0.812366
35.000000 71.000000 -0.254730 -0.871694
35.000000 72.000000 -0.250922 -0.924855
35.000000 73.000000 -0.247951 -0.971850
35.000000 74.000000 -0.244949 -1.016177
35.000000 75.000000 -0.241917 -1.057837
35.000000 76.000000 -0.238853 -1.096830
35.000000 77.000000 -0.235759 -1.133156
35.000000 78.000000 -0.232652 -1.166671
35.000000 79.000000 -0.229530 -1.197374
35.000000 80.000000 -0.226396 -1.225266
```

À titre de rappel, les valeurs de ce fichier représentent : x , y , $u(x, y)$, $v(x, y)$, avec

x : indice des abscisses,

y : indice des ordonnées,

$u(x, y)$: déplacement horizontal en (x, y) ,

$v(x, y)$: déplacement vertical en (x, y) .

III – Fichier de commandes pour Gnuplot

```
reset
set terminal png size 800, 600
set output PATH_OUTPUT

set size ratio -1
set xr[0 : 256]
set yr[0 : 256]
k = 4
plot PATH_FICHER_DONNEES ev 4:4 u 1:(256.000000-$2):(k*$3):(-k*$4):(sqrt(($3)**2+
($4)**2)) w vec lc palette notitle
```

Ceci est un exemple de script Gnuplot que l'on écrit à partir de Scilab, ce qui permet de modifier aisément les variables PATH_OUTPUT et PATH_FICHER_DONNEES afin de rendre cette partie la plus générique possible.

IV – Résultat du traitement sur une séquence d'images

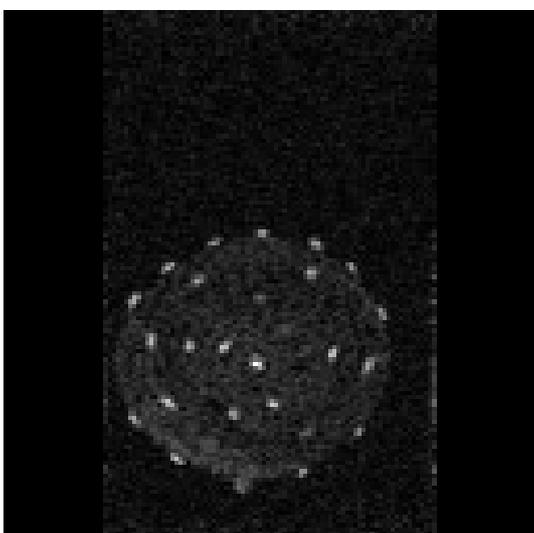
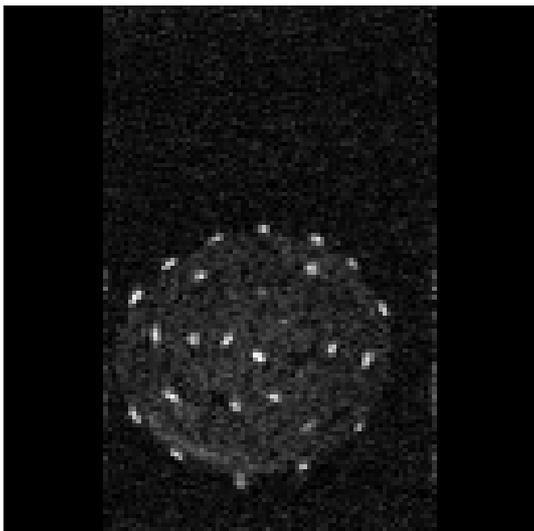
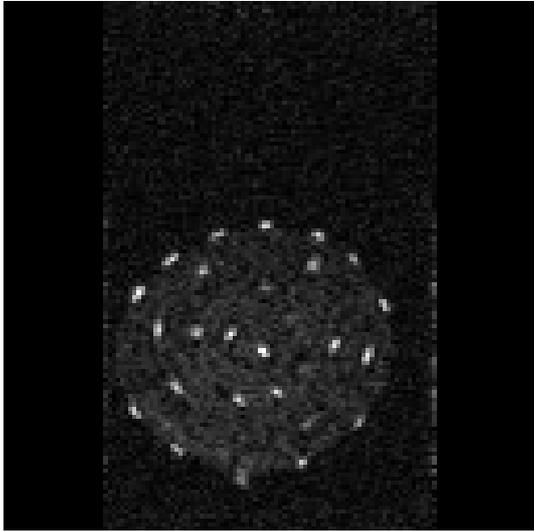


Illustration 16: Séquence de 3 images consécutives aux instants t , $t + 1$ et $t + 2$

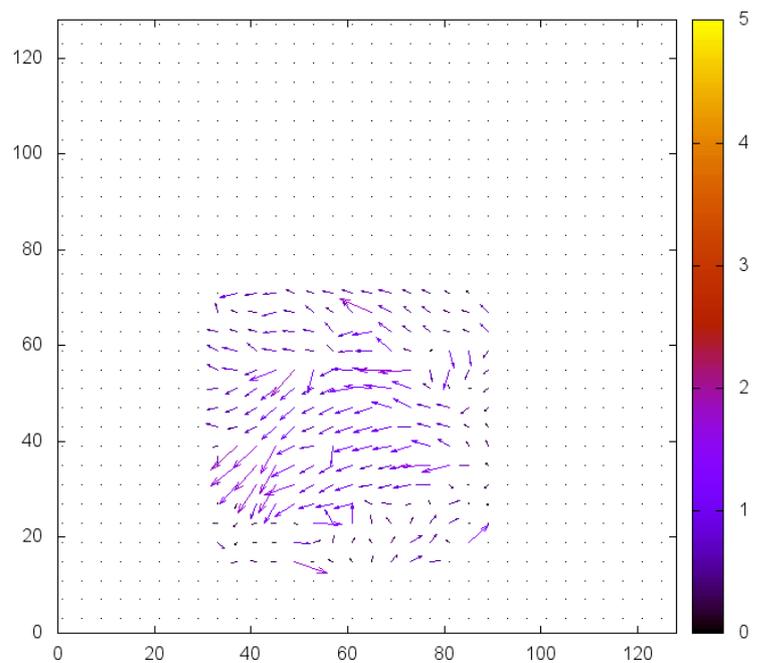
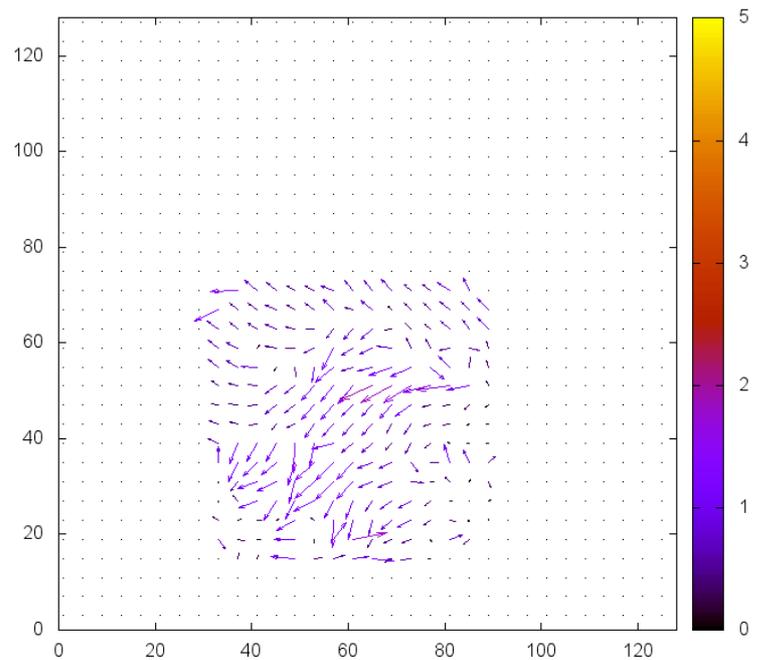


Illustration 15: Résultat du traitement pour chaque couple d'images