

École polytechnique de Louvain

Mesures de débit en rivière à partir de techniques d'imagerie

Développement d'une application

Auteurs: **Edouard DE GRAND RY, Charlotte EVERAERT**

Promoteur: **Sandra SOARES-FRAZÃO**

Lecteurs: **Olivier CARLIER, Paul FISETTE, François JONARD**

Année académique 2020–2021

Master [120] : ingénieur civil des constructions

Résumé

Les techniques d'imagerie sont utilisées depuis quelques années pour mesurer le débit des cours d'eau. Elles sont avantageuses de par leur rapidité, leur facilité de mise en oeuvre et leur faible coût. Elles sont également plus adaptées à la mesure de débit en périodes de crues et d'étiages par rapport aux techniques de mesures intrusives qui nécessitent un contact avec l'eau.

Dans ce contexte, une application a été développée lors d'un précédent mémoire. Elle permet d'obtenir le débit d'un cours d'eau à partir d'une vidéo et de la bathymétrie. Cette application, contrairement à certains logiciels existants, prend en charge le processus complet jusqu'à l'obtention du résultat final.

L'objectif de notre travail a été de valider cette application, de vérifier son fonctionnement, les résultats obtenus et d'identifier les limites de celle-ci. Pour cela, nous avons réalisé un ensemble de tests sur une série de vidéos, nous permettant de détecter certaines erreurs de fonctionnement. Cela nous a amené à apporter les corrections et améliorations nécessaires.

Enfin, une phase de validation a été mise en oeuvre afin de démontrer, sur base d'un grand nombre de tests, la stabilité et la précision des résultats obtenus par l'application suite aux corrections apportées. Cette phase a également permis de déterminer l'influence de chaque étape de l'application dans le résultat final.

Maintenant que chaque partie de l'application a été vérifiée, il serait intéressant, dans les pistes futures de ce travail, d'implémenter de nouvelles méthodes afin de rendre l'application plus complète et compétitive et de la rendre accessible en open-source.

Abstract

Imaging techniques are used since several years to measure river discharge. They are beneficial because they are fast, easy to use and low cost. Moreover, they are more adapted to discharge measurements in case of floods and drought. This is not the case with invasive methods, which always need a contact with water.

In this context, an application was developed in a previous master thesis. It allows to obtain river discharge from a video and bathymetry. Unlike some existing software's, this application takes in charge the whole process until final result obtention.

The aim of our work was to validate this application, to check its working, the obtained results and to identify its limits. Then, we performed a set of tests on a serie of videos, allowing us to detect several functioning mistakes. This leads us to bring necessary corrections and improvements.

Finally, a validation step was carried out to demonstrate result stability and precision after corrections. This phase allows to determine the influence of each step of the application in the final result.

Now that each application step was checked, it would be interesting to develop new methods in future works to make application more completed and competitive and to make it open-source reachable.

Remerciements

Nous tenons tout d'abord à remercier Madame Soares Frazão, notre promotrice, pour son suivi et ses conseils tout au long de cette année.

Merci également à Messieurs Carlier, Fisette et Jonard d'avoir accepté de faire partie de notre jury.

Merci à Loïc et Emmanuel de nous avoir aidé à réaliser notre set de vidéos sur le terrain.

Enfin, nous tenons à remercier nos proches pour leur soutien tout au long de ce mémoire.

"La théorie c'est quand on sait tout et que rien ne fonctionne. La pratique c'est quand tout fonctionne et que personne ne sait vraiment pourquoi. Ici, nous avons réuni théorie et pratique : rien ne fonctionne... et personne ne sait pourquoi!"
- Albert Einstein

Table des matières

Résumé	i
Remerciements	i
Introduction	1
I État de l’art	3
1 Techniques actuelles de mesures de débit	4
1 Mesures intrusives	4
1.1 Outils de mesure de vitesses	4
1.2 Utilisation des données mesurées	6
1.3 Limites des techniques intrusives	7
2 Mesures non intrusives	7
2.1 PIV	7
2.2 LSPIV	8
2 Logiciels de mesures de débit actuels	10
1 Fudaa-LSPIV	10
2 PIVlab	12
3 Discharge	13
4 RIVeR	16
5 Réflexions sur les logiciels existants	18
3 Concepts utiles à l’application	20
1 Traitement d’images	20
1.1 Stabilisation	20
1.2 Calibration	21
1.3 Orthorectification	24
1.4 Filtres	27
2 Calcul de la hauteur d’eau	32
2.1 Méthode d’Hellebaut	33
2.2 Méthode image	36
2.3 Méthode angle et hauteur	37
3 Analyse PIV	39
3.1 Définition	39
3.2 OpenPIV	40
4 Calcul du débit	45

II	Analyse de la première version de l'application	49
4	Structure de l'application	50
1	Structure générale	50
1.1	Dossiers internes	50
1.2	Flowchart	51
2	Chargement de la bathymétrie	51
3	Chargement de la vidéo	54
4	Stabilisation	55
5	Calibration	55
6	Orthorectification	57
7	Détermination du niveau d'eau	58
8	Région d'intérêt et masques	60
9	Filtres	61
10	Analyse PIV	62
11	Section d'intérêt	63
12	Détermination du débit	65
5	Tests effectués sur l'application	66
1	Vidéo globe challenge	66
2	Méthodologie	68
2.1	Utilisation du set Vidéo Globe Challenge	68
2.2	Paramètres à faire varier	69
2.3	Types de test	69
6	Problèmes détectés et améliorations possibles suite aux tests effectués	71
1	Problèmes détectés	71
1.1	Range	71
1.2	Step	71
1.3	Analyse PIV	72
1.4	Mask	74
1.5	Facteur d'échelle	75
1.6	Erreurs de code	75
2	Améliorations possibles	76
2.1	Manque de robustesse générale	76
2.2	Version Beta	76
2.3	Méthode Image	76
3	Conclusion	77
III	Nouveaux développements	78
7	Corrections des problèmes détectés	79
1	Correction du range	79
2	Correction de l'analyse PIV	79
2.1	Influence du step	79
2.2	Analyse des fonctions d'OpenPIV	80
3	Correction de la méthode Angle et Hauteur	81
3.1	Cas du drone centré par rapport à la rivière	81
3.2	Cas du drone décalé par rapport au centre de la rivière	82
4	Correction de l'application du masque	83
5	Correction du facteur d'échelle	83

5.1	Avec orthorectification	83
5.2	Sans orthorectification	84
8	Améliorations	85
1	Développement d'un aspect "user-friendly"	85
2	Fichiers en local	88
3	Nouvelles options	88
3.1	Méthode de calibration	88
3.2	Méthode pour déterminer le niveau d'eau	89
3.3	Choix des filtres	89
3.4	Sauvegarde des résultats dans fichier texte	93
3.5	Version sous Graphical Processing Unit	93
3.6	Multi-pass	94
9	Validation de l'application	97
1	Vidéos utilisées pour les tests	97
1.1	Cas limites du Vidéo Globe Challenge	97
1.2	Nouveau set de vidéos	99
2	Tests de validation	101
2.1	Influence de l'ordinateur de l'utilisateur	101
2.2	Influence de la méthode de validation de l'analyse PIV	102
2.3	Influence de la présence des traceurs	104
2.4	Limites dues à la forme de la bathymétrie	106
2.5	Influence de l'étape d'orthorectification	110
2.6	Influence du ROI	112
2.7	Influence de la section d'intérêt sans orthorectification	114
2.8	Précision générale	115
3	Conclusions de validation	115
10	Résultat final	117
1	État des lieux	117
2	Pistes d'amélioration	119
	Conclusion	120
	Table des figures	122
	Liste des tableaux	127
	Bibliographie	128
A	Tests de validation	132
1	Influence de la méthode de validation de l'analyse PIV	132
2	Limites dues à la forme de la bathymétrie	134
3	Influence de l'étape d'orthorectification	137
4	Influence du ROI	139
5	Influence de la section d'intérêt sans orthorectification	142
6	Précision générale	145

Introduction

La connaissance du débit d'un cours d'eau est indispensable dans de nombreux domaines tels que la conception d'ouvrages hydrauliques, la navigation ou la distribution d'eau. Ces données permettent également de prédire les événements extrêmes tels que les périodes de crues et d'étiages.

Les techniques de mesures standards utilisées depuis toujours nécessitent un contact entre la surface d'eau et l'appareil de mesures : ces techniques sont dites « intrusives ». Ce contact direct est contraignant par plusieurs aspects : le coût, le temps de mesure et l'inadéquation des techniques aux événements extrêmes. Leur champ d'application est donc relativement restreint. En effet, les débits en période de crues et d'étiages sont quasiment impossibles à mesurer au moyen des techniques intrusives alors qu'ils sont les plus utiles.

Pour cette raison, il est essentiel d'investiguer de nouvelles techniques réduisant au maximum le contact avec l'eau : les mesures non intrusives. Celles-ci sont basées sur la détermination des vitesses de surface d'un cours d'eau à partir d'images. Connaissant la bathymétrie du cours d'eau, l'aire mouillée peut être calculée et finalement le débit peut être déduit. Ces techniques apparaissent de plus en plus pertinentes au regard de la démocratisation des drones, permettant alors d'explorer les sites hydrauliques inaccessibles à l'homme. De plus, les mesures sans contact sont plus pratiques, plus rapides et plus adaptées aux situations extrêmes qui sont de plus en plus fréquentes au vu de la situation climatique.

Dans ce contexte, une application a été développée l'année dernière par Simon Renard, étudiant de l'UCL, dans le cadre de son mémoire et sur base de précédents mémoires traitant des techniques non-intrusives. Celle-ci est écrite dans le langage Python et l'interface graphique utilisée pour créer l'application est Kivy (bibliothèque de Python). L'application permet de déterminer le débit d'une rivière sur base d'une vidéo et présente deux enjeux majeurs : la détermination des vitesses et du niveau d'eau. Les vitesses des particules présentes à la surface d'un cours d'eau sont obtenues par la technique «Large Scale Particule Image Velocimetry». Son algorithme calcule les déplacements de ces particules entre deux images et connaissant l'intervalle de temps entre celles-ci, les vitesses peuvent être déduites. Si la bathymétrie d'une section du cours d'eau est connue de l'utilisateur, le débit à travers l'aire mouillée peut alors être calculé. Pour calculer le niveau d'eau, plusieurs méthodes ont été développées et seront présentées au cours de ce travail.

L'application développée présente l'avantage d'être auto-suffisante puisqu'elle ne nécessite pas l'intervention d'autres logiciels dans le processus d'obtention du débit. Les étapes de pré-processing et de post-processing y sont implémentées, permettant respectivement le traitement des images et des données de vitesses calculées.

L'objectif de notre travail a été d'améliorer l'application reçue selon deux axes : simplifier son usage et assurer la fiabilité et la stabilité des résultats générés. Pour cela, notre travail s'est divisé en trois parties. Dans un premier temps, nous avons réalisé une série de tests sur un set de vidéos filmées dans des conditions variables. Cette étape a permis d'une part, de nous familiariser avec le code de l'application et d'autre part, de mettre en évidence un certain nombre de problèmes

et d'aspects manquants. Cela nous a amené, dans un second temps, à apporter les corrections et améliorations nécessaires pour atteindre les objectifs fixés. Dans ce cadre, de nouvelles méthodes ont été ajoutées. Enfin, une phase de validation a permis d'assurer la stabilité des résultats obtenus au moyen d'un ensemble de tests sur une série de vidéos.

Première partie

État de l'art

Chapitre 1

Techniques actuelles de mesures de débit

Parmi les techniques utilisées pour mesurer le débit, deux catégories se distinguent : les mesures intrusives et les mesures non intrusives. Dans ce chapitre, ces deux techniques seront présentées ainsi que leur mise en oeuvre et leurs avantages et limites.

1 Mesures intrusives

Les techniques de mesures intrusives sont les techniques qui nécessitent un contact direct entre l'instrument de mesure et le cours d'eau. Comme notre travail s'est concentré principalement sur la détermination des vitesses, seules les techniques de mesure de vitesses seront présentées. Il existe également des techniques de mesures de contact pour la bathymétrie¹, la hauteur d'eau et le débit. Pour en savoir plus sur ces techniques, d'autres mémoires traitant du sujet peuvent être consultés.

1.1 Outils de mesure de vitesses

Capteur à moulinet

Celui-ci porte également le nom de courantomètre mécanique. A l'intérieur de l'appareil, un signal est généré et produit la révolution d'une hélice. Le nombre de tours de cette hélice (déterminé par un compteur) sur une certaine période de temps permet de calculer la vitesse angulaire due à la vitesse de l'eau. La vitesse du mouvement angulaire est alors transformée en vitesse linéaire. La qualité des résultats de vitesses obtenues dépend donc de l'inertie du rotor [1]. Le moulinet peut également se présenter sous forme d'une masse en forme d'ogive équipée d'un moulinet (voir figure 1.2). Cette forme est appelée "saumon" [2].

Le moulinet permet de réaliser des mesures de vitesses instantanées en un point de la rivière. Ces vitesses pourront être utilisées pour calculer le débit au moyen des méthodes mean et mid expliquées dans la sous-section 1.2 de ce chapitre.

Cette méthode de mesure de vitesse a l'avantage d'être peu chère et applicable à grande profondeur. Les cours d'eau où le moulinet à hélice peut être utilisé varient des canaux aux rivières en passant par les réseaux d'eaux et les canalisations. De plus, le courantomètre est petit et léger donc facilement transportable.

1. La bathymétrie est l'ensemble des coordonnées (x, y, z) des points qui constituent une section d'un cours d'eau.



FIGURE 1.1 – Capteur à moulinet [3]



FIGURE 1.2 – Saumon d'une station de mesure [4]

Cependant, étant donné que la mesure de vitesse se fait sous eau, l'appareil s'abîme rapidement et n'est donc pas durable. Les mesures peuvent donc être erronées du à l'usage de l'appareil. De plus, le courantomètre ne reconnaît pas le sens du courant et risque donc de commettre des erreurs quand des courants de retour se produisent. Cette technique de mesure peut également prendre beaucoup de temps.

Courantomètre électromagnétique

Le courantomètre électromagnétique utilise une sonde qui génère un champ magnétique. D'après la loi de Faraday, un conducteur en mouvement dans un champ magnétique génère une différence de tension qui varie linéairement avec la vitesse du conducteur. Les conducteurs sont ici les particules d'eau chargées qui bougent selon le courant. Celles-ci vont donc générer une force électromagnétique qui pourra être mesurée par des électrodes présents dans la sonde. Les résultats obtenus dépendent donc de la forme de la sonde et de la position des électrodes dans celle-ci [1].



FIGURE 1.3 – Courantomètre électromagnétique [5]

Cette méthode a l'avantage de pouvoir détecter un champ de vitesse relativement large (entre $0,1$ et $6 \left[\frac{m}{s}\right]$) [6] et capte également les changements de sens du courant. De plus, les vitesses mesurées sont directement fournies sans passer par le nombre de rotations comme c'était le cas avec le moulinet.

Cependant, étant donné que l'appareil produit un champ magnétique, son utilisation doit restée éloignée des objets métalliques. De plus, tout comme le courantomètre mécanique, son utilisation prend beaucoup de temps.

Vélocimètre acoustique

Les appareils acoustiques utilisés pour la mesure de vitesse sont aussi appelés Acoustic Doppler Velocimeter (ADV) ou Acoustic Doppler Current Profiler (ADCP). Ces appareils émettent une onde acoustique dans l'eau. Celle-ci est réfléchiée par les particules du cours d'eau et analysée selon principe de Doppler.



FIGURE 1.4 – ADCP [7]

L'avantage de cette méthode est sa rapidité.

Cependant, les vitesses ne peuvent être mesurées que sur une partie de la profondeur d'eau. En effet, à cause de la réflexion des ondes sur le fond du cours d'eau, les vitesses de la partie inférieure de la profondeur (environ six pour cents de la hauteur) ne sont pas mesurables [2]. De plus, comme l'appareil est légèrement enfoncé dans l'eau, les vitesses de la zone sous la surface d'eau ne sont pas non plus mesurables.

Méthode des flotteurs

Dans cette méthode, des traceurs sont dispersés sur la surface du cours d'eau. Le temps mis par les particules pour parcourir une distance déterminée est mesuré. A partir de ces données, les vitesses de surface sont obtenues et ensuite transformées en vitesses moyennes sur la profondeur grâce à un certain coefficient. Pour obtenir des résultats corrects, il faut que les particules soient bien réparties sur toute la largeur du cours d'eau [2]. Cette méthode est généralement utilisée si aucune des méthodes citées précédemment n'est utilisable.

1.2 Utilisation des données mesurées

A partir des mesures de vitesses obtenues par les appareils présentés ci-dessus, le débit à travers l'aire mouillée peut être obtenu via deux méthodes différentes : la méthode *mean* et la méthode *mid*.

Ces méthodes requièrent une subdivision de la largeur en un certain nombre de sous-sections. Sur la verticale de chaque sous-section, la vitesse doit être mesurée sur plusieurs points, correspondant à certains pourcentages de la hauteur de cette verticale. Ces mesures permettent de déterminer le profil de vitesse vertical sur chaque tranche. Le débit total est calculé comme la somme des débits obtenus sur chaque tranche verticale de la section.

La différence entre ces deux méthodes se trouve dans les hypothèses utilisées. La méthode *mid* suppose que le débit des tranches proches des rives est toujours nul. En effet, dans ces sous-sections, le débit est souvent faible et peut être approximé à zéro [8]. Bien que la méthode *mean* ne se base pas sur la même hypothèse, les deux méthodes donnent toutes deux des résultats semblables.

1.3 Limites des techniques intrusives

Le contact obligatoire avec l'eau de ces techniques présente malheureusement certains inconvénients, notamment pour les mesures de vitesses en période de crue et d'étiage. Or, les données de débit durant ces périodes particulières sont indispensables, que ce soit pour le design d'ouvrages hydrauliques ou pour la prédiction de crue. Les principaux inconvénients des mesures intrusives sont les suivants.

Tout d'abord, les méthodes intrusives ne sont pas toujours évidentes à mettre en oeuvre lors de crues car les appareils de mesure n'y sont pas forcément adaptés. En effet, lorsque les cours d'eau atteignent des débits plus importants, le transport de sédiments devient également plus important. Le matériel de mesure risque donc d'être endommagé.

De même, les appareils de mesures intrusifs ne sont généralement pas adaptés pour la mesure de faibles débits. Les mesures obtenues lors des périodes d'étiages ne sont donc pas toujours fiables.

Enfin, les mesures de contact nécessitent, dans la plupart des cas, une intervention humaine. Dans le cas où la rivière est inaccessible, les outils avec contact sont tout simplement inutilisables. En cas de crues, les débits peuvent augmenter très rapidement. Les mesures deviennent alors dangereuses pour l'homme. De plus, étant donné qu'en période de crue le niveau d'eau varie très vite, la mesure doit être réalisée rapidement ce qui n'est pas toujours le cas avec les mesures intrusives.

Nous constatons donc que le champ d'application des mesures avec contact est relativement limité et n'est pas adapté à toutes les situations.

2 Mesures non intrusives

A l'inverse, les techniques non intrusives ne nécessitent aucun contact entre l'appareil de mesure et l'eau. Il s'agit donc de l'analyse d'images de l'écoulement sur base desquelles, les vitesses et ensuite le débit peuvent être déduits. L'analyse sans contact permet une mesure à la fois plus pratique à réaliser mais aussi plus sécuritaire pour l'appareil et la personne effectuant les mesures. De plus, lorsque le site est difficile d'accès pour l'homme, le drone peut quant-à-lui facilement y accéder.

2.1 PIV

L'analyse PIV (*Particule Image Velocimetry*) est une technique qui permet de calculer un champ de vitesses instantanées 2D d'une surface à partir d'images consécutives d'un cours d'eau. Les images sont analysées par paires. Le cours d'eau en question est encensé de particules dont les déplacements sont suivis via la technique de corrélation entre les paires d'images successives. Comme les déplacements sont connus et que le temps entre deux images l'est également, les vitesses peuvent être déduites.

$$\vec{V} = \frac{\Delta \vec{X}}{\Delta t} \quad (1.1)$$

A partir de ces vitesses, connaissant la bathymétrie de rivière ainsi que le coefficient liant les vitesses de surface aux vitesses moyennes de la section, le débit du cours d'eau peut être obtenu.

Cette technique de mesure n'apparaît qu'à la fin des années 70 et était appelée LSV (*laser speckle velocimetry*) par Meynart. Ce n'est qu'à partir de 1984 que le terme PIV commence à être utilisé suite aux travaux d'Adrian et de Pickering et Halliwell qui mettent en avant l'importance

des particules [9]. Son développement a été assez discret dans un premier temps avant d’accélérer ces dernières années avec d’une part, les progrès constants de l’informatique et l’arrivée de caméras à haute résolution et d’autre part, les besoins des laboratoires souhaitant connaître rapidement l’ensemble d’un champ de vitesse. C’est donc bien une technique qui est exclusivement utilisée en laboratoire.

Cette méthode se décompose en deux étapes : l’acquisition et l’analyse des images obtenues. Le but de l’acquisition est d’obtenir des images des particules se déplaçant dans le fluide dont nous souhaitons étudier le champ de vitesse. Il faut alors créer un plan lumineux dans l’écoulement et la caméra sera placée perpendiculairement à ce plan. Rapidement, le laser s’impose comme source lumineuse idéale pour cette opération à la fois pour sa puissance et parce qu’il permet de mieux maîtriser les caractéristiques du plan lumineux, en particulier sa divergence. Les images obtenues sont ensuite analysées soit par **autocorrélation**, soit par **intercorrélation**. Pour être utilisée, l’intercorrélation doit être associée à une caméra spéciale permettant de sauvegarder rapidement deux enregistrements. Comme de telles caméras ne sont réellement disponibles que depuis quelques années, les premiers travaux concernant l’analyse PIV ont surtout utilisé la méthode de l’autocorrélation [10].

Le principe de l’autocorrélation est de comparer une groupe des particules dans deux images consécutives, séparées d’un Δt connu. Cette comparaison s’effectue en évaluant la fonction d’autocorrélation (présentée à la section 3.2 du chapitre 3) et permet de déterminer le Δx entre les deux images, comme représenté à la figure 1.5. De cette façon, la vitesse peut être déduite.

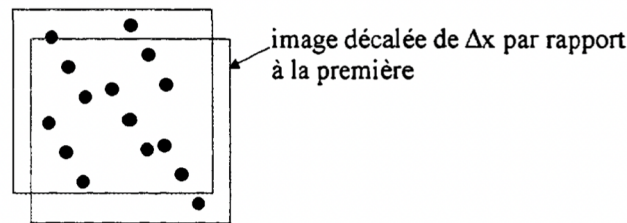


FIGURE 1.5 – Principe de l’autocorrélation [10]

2.2 LSPIV

Alors que la méthode PIV est une méthode précise développée pour une utilisation en laboratoire, l’idée de l’utiliser dans des conditions moins contrôlées et sur des objets plus grands, telle qu’une rivière par exemple, s’est rapidement fait ressentir. La méthode LSPIV (*Large Scale Particle Image Velocimetry*) correspond à ce développement de la PIV à grande échelle. Bien entendu, des ajustements sont nécessaires. Par exemple, l’ensemencement de la rivière doit être fait avec des particules de taille plus importante pour qu’elles puissent être détectées malgré le champ de vision agrandi. Par contre, ces traceurs ne sont plus nécessaires dans le cas de rivière à grand débit où les remous de l’eau peuvent être suivis. Des rectifications géométriques telles qu’une calibration de la caméra ou une orthorectification des images deviennent également nécessaires.

D’après Jodeau et Bercovitz (2017), la méthode LSPIV se décompose en quatre étapes :

1. l’**acquisition** d’une séquence d’images dont la différence de temps, Δt , est connu entre chacune d’elles.
2. la rectification géométrique de ces images. Le but est de corriger les distorsions dues aux effets de perspectives. C’est l’étape d’**orthorectification**.

3. Le **calcul** en tant que tel des déplacements, Δx entre chaque image, basé sur une analyse statistique de corrélation. C'est l'étape clef de l'analyse.
4. L'analyse **post-processing** des données de vitesses obtenues. Il s'agit de moyennes effectuées sur les vitesses et de l'estimation du débit.

Cette technique permet d'obtenir les mouvements de l'eau entre chaque paire d'images et donc la vitesse de surface de l'écoulement. Une moyenne sur l'ensemble des paires étudiées de la vidéo peut être faite et connaissant la bathymétrie au préalable, le débit peut être calculé. Elle a déjà été utilisée pour calculer les champs de vitesses dans différents types de canaux et dans différentes conditions et configurations et donnait déjà de très bons résultats. Les travaux de Bradley et al. en 2002, Hauet et al. en 2008 et Jodeau et al. en 2008 peuvent en attester [11]. C'est pour cette raison qu'elle a été implémentée dans l'application l'année dernière par Simon Renard dans le cadre de son mémoire.

Chapitre 2

Logiciels de mesures de débit actuels

Dans cette section, plusieurs logiciels pouvant être utilisés pour déterminer le débit de cours d'eau à partir de techniques d'imagerie seront présentés. Pour chacun d'eux, la démarche d'utilisation ainsi que les avantages et inconvénients seront présentés.

1 Fudaa-LSPIV

Le logiciel Fudaa-LSPIV est un logiciel distribué gratuitement et librement. Il a été créé par la société d'ingénierie informatique DeltaCAD en collaboration avec EDF et Irstea. Basé sur des travaux scientifiques précédents, il est conçu spécifiquement pour la technique LSPIV. Celui-ci se présente sous la forme d'une interface graphique "user-friendly" en Java qui fait appel à des solveurs Fortran [12].

Démarche

Afin de déterminer le débit à partir de Fudaa-LSPIV, il faut réaliser les étapes suivantes [13] :

1. **Importer** une série d'images ou une vidéo du cours d'eau. Les particules naturelles ou ajoutées doivent y être visibles pour pouvoir être tracées. Il est également possible de charger la vidéo sur un certain intervalle pour réduire le temps de l'analyse.
2. **Stabilisation** de la vidéo si celle-ci présente des mouvements de caméra.
3. **Orthorectification** de l'image. Trois cas de figure se présentent (voir figure 2.1) :
 - Mise à l'échelle : lorsque la vidéo est déjà filmée perpendiculairement au plan d'eau.
 - Orthorectification 2D : lorsque la vidéo est filmée à partir de la berge et que les points de référence sont situés dans le même plan que le plan d'eau.
 - Orthorectification 3D : lorsque la vidéo est filmée à partir de la berge et que les points de référence utilisés sont situés à différentes hauteurs.

Après orthorectification, l'utilisateur doit fournir le niveau d'eau en mètre.

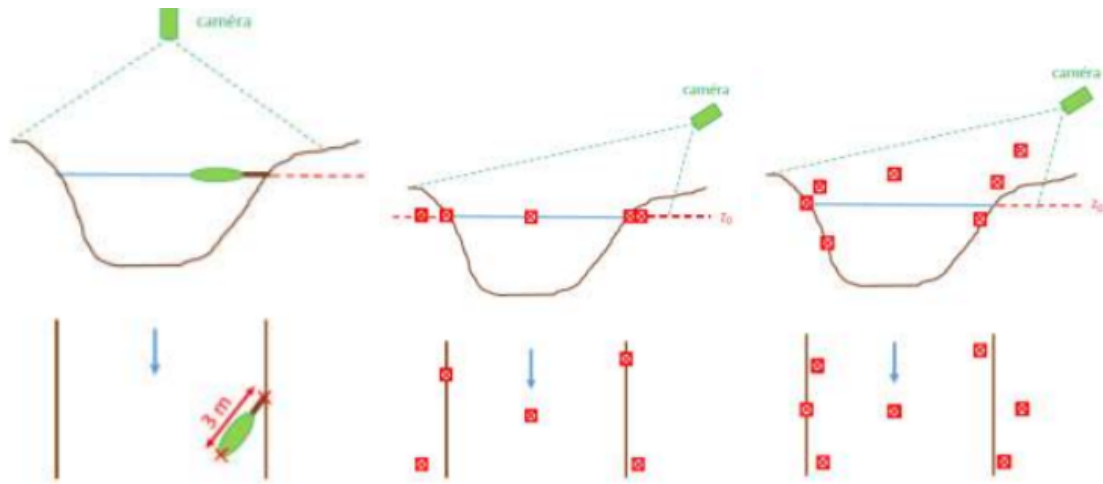


FIGURE 2.1 – 3 cas différents peuvent se présenter à l'orthorectification [13]

4. **Analyse LSPIV** réalisée sur base des images orthorectifiées. Les déplacements des particules entre deux images sont calculés. Connaissant l'intervalle de temps entre ces deux images, les vitesses peuvent être déduites. La méthode de corrélation croisée telle que présentée à la section 3.2 du chapitre 3 est utilisée. Chaque image est divisée en grille sur laquelle les calculs sont effectués de manière itérative. Le champ de vitesses calculé peut ensuite être affiché à l'écran.
5. **Post-processing** : cette étape permet un filtrage des résultats des vitesses et l'élimination des valeurs improbables. Le filtrage peut être effectué de deux manières :
 - Soit il est basé sur un intervalle entre un minimum et un maximum tous deux indiqués par l'utilisateur. Si les vitesses sont comprises dans cet intervalle, celles-ci seront validées.
 - Soit il se base sur une valeur de corrélation acceptable. Les vecteurs filtrés peuvent également être affichés.
 Ensuite, ces vecteurs peuvent être moyennés et affichés. L'affichage s'accompagne d'une échelle des couleurs qui permet de classifier les vecteurs par intensité (voir figure 2.2).
6. Importer un ou plusieurs profils de **bathymétrie** soit via les différentes coordonnées qui composent ceux-ci soit via un fichier texte contenant les coordonnées sous un certain format.
7. Choix du **coefficient** liant la vitesse de surface à la vitesse moyenne sur la profondeur.
8. **Calcul du débit** sur la section dont la bathymétrie a été précisée par l'utilisateur. La section est alors divisée en un certain nombre de pas, choisi par l'utilisateur. Sur chaque pas, le débit est calculé à partir des vitesses les plus proches et de l'aire de la sous-section. Le débit total équivaut à la somme des débits sur les différentes sous-sections. Si plusieurs sections ont été renseignées, le débit traversant chaque section est calculé de la même manière.

Avantages et inconvénients

Le principal avantage de Fudaa-LSPIV est que l'utilisateur peut réaliser toute la démarche permettant de calculer les vitesses et le débit dans le même logiciel, sans intervention d'un logiciel externe. De plus, étant donné qu'une stabilisation et une orthorectification de l'image peuvent être réalisées, les vidéos filmées lors de crues par un public amateur pourront être traitées pour être exploitées par le logiciel.

Cependant, l'inconvénient de Fudaa-LSPIV est que les résultats calculés dépendent fortement de la démarche et des paramètres choisis par l'utilisateur. Il n'est pas toujours facile, pour un utilisateur quelconque, de comprendre l'influence de ces paramètres.

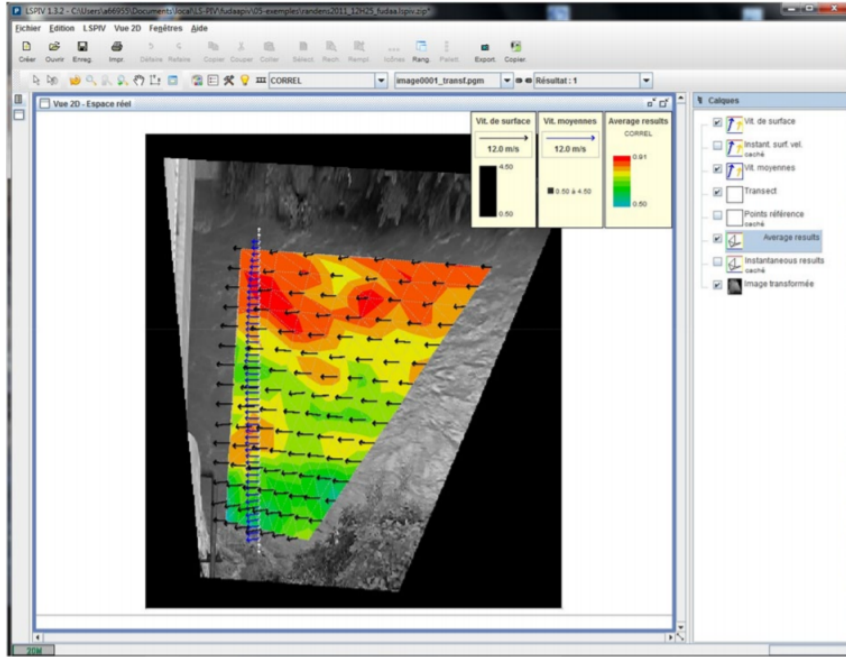


FIGURE 2.2 – Affichage des vitesses calculées lors de l'analyse PIV [12]

2 PIVlab

PIVlab est une routine d'analyse PIV dans Matlab qui permet d'appliquer la technique PIV pour obtenir les vitesses moyennes de surface en analysant les images par paires. Elle a été développée en 2010 dans le cadre de la thèse d'un chercheur biologiste allemand, William Thielicke, également docteur en dynamique des fluides et biomimétique [14]. Au départ, le logiciel a été créé pour analyser l'aérodynamique des vols d'oiseaux. Puis, il a été adapté et son champ d'application s'est élargi à l'analyse de la déformation du sable et à la technique LS-PIV. Il permet également de calculer et d'exporter des données relatives à un modèle d'écoulement.

PIVlab a l'avantage d'avoir une interface graphique qui rend son utilisation plus visuelle pour l'utilisateur [15]. De plus, de nouvelles fonctionnalités sont régulièrement ajoutées.

Démarche

L'analyse PIV est divisée en trois étapes [16] :

1. Pré-processing :

- Importation des images
- L'utilisateur choisit une région d'intérêt sur laquelle l'analyse PIV sera réalisée et plusieurs masques pour exclure les zones sans vitesse. Le choix d'une région d'intérêt et des masques permet de réduire la zone sur laquelle les calculs des déplacements sont réalisés et donc de diminuer le temps de calcul.
- Il est possible d'appliquer un ou plusieurs des quatre filtres disponibles (CLAHE, Intensity High Pass, Intensity Capping, Wiener2 denoise) qui accentuent le contraste entre les particules et le fond de la rivière. L'effet et le fonctionnement de chaque filtre seront détaillés dans la section 1.4 du chapitre 3.
- Il est possible de réaliser la calibration de la caméra.

2. **Analyse LSPIV.** Elle est basée sur un algorithme de corrélation. L'utilisateur a le choix entre une analyse "single pass" et une analyse "multi-pass".

3. **Post-processing.** Cette phase comprend la validation, l'interpolation et le lissage des vecteurs vitesses. L'utilisateur a le choix entre différentes méthodes pour la validation des vecteurs vitesses : une validation géométrique sur base du rang de vitesses acceptées ou une validation basée sur la médiane.

Les données relatives au calcul des vitesses de surface peuvent ensuite être exportées sous différents formats (fichier texte, fichier Matlab, etc).

Avantages et inconvénients

L'avantage de la routine PIVlab est qu'elle permet un calcul relativement rapide et précis des vitesses de surface à partir d'une vidéo. Elle possède également une interface graphique qui rend son utilisation beaucoup plus abordable.

L'inconvénient de cette routine est qu'elle ne permet malheureusement aucune stabilisation ni aucune orthorectification. Les images fournies doivent donc être stabilisées et orthorectifiées au préalable avec un autre logiciel. De plus, pour déterminer l'aire mouillée et le débit, le travail du logiciel doit être complété par une analyse sur Excel. Dans le processus de détermination du débit, PIVlab sert donc uniquement à calculer les vitesses de surface sans implémentation du pre-processing et du post-processing.

3 Discharge

Discharge est une application pour android accessible via Google Play Store qui permet de déterminer rapidement les vitesses et le débit d'une rivière à partir d'une vidéo. L'application a été développée par le Docteur Beat Lüthi au sein de l'entreprise Phototrack AG.

Pour pouvoir utiliser l'application, il est nécessaire de créer un compte personnel où seront stockées les informations relatives aux différents sites créés. Ce compte est lié à une plateforme en ligne prévue par les concepteurs [17].

La première étape est la création du site où l'utilisateur se trouve. Le nom et la localisation (sous forme de coordonnées géographiques) de chaque site doivent être précisés. Chaque site sera ensuite répertorié sur la plateforme.

Démarche

1. Choisir un profil type pour la **bathymétrie** de la rivière. L'utilisateur a l'opportunité de choisir entre quatre types de section et doit fournir les informations relatives au type de section choisie [18].
 - Section rectangulaire : largeur et profondeur
 - Section trapézoïdale : largeur, profondeur et largeur de fond
 - Section parabolique : largeur et profondeur
 - Section libre : fichier à charger au préalable sur la plateforme contenant les coordonnées (y, z) des points de la section. Il faut d'abord charger toutes les coordonnées selon y séparées par des virgules puis de la même manière toutes les coordonnées selon z.

Le terme profondeur désigne la profondeur par rapport au point le plus bas de la section.

2. Fournir une valeur pour la **rugosité** de la rivière qui correspond au coefficient de Strickler k_{st} . Ce coefficient est utilisé pour calculer la vitesse moyenne v .

$$v = k_{st} \cdot r^{2/3} \cdot l_E^{1/2} \quad (2.1)$$

avec r le rayon hydraulique¹ et l_E le gradient hydraulique. Typiquement, les surfaces lisses auront un coefficient compris entre 50 et 100 alors que les surfaces rugueuses auront un coefficient compris entre 25 et 50. Afin que l'utilisateur puisse choisir la valeur de rugosité la plus adaptée à la réalité, un tableau des coefficients en fonction des différents cours d'eau est présent dans l'onglet information de l'application.

3. **Orthorectification** : l'utilisateur doit placer quatre balises dans le même plan horizontal, avec deux balises sur chaque rive. Cela implique donc que les berges de cette partie de la rivière soient accessibles à l'homme. Les distances entre ces balises ainsi que les distances entre chaque balise et la rive sur lesquelles elles se trouvent doivent être fournies par l'utilisateur (voir figure 2.3).



FIGURE 2.3 – Chacun des quatre points de contrôle est placé dans une des quatre zones [18]

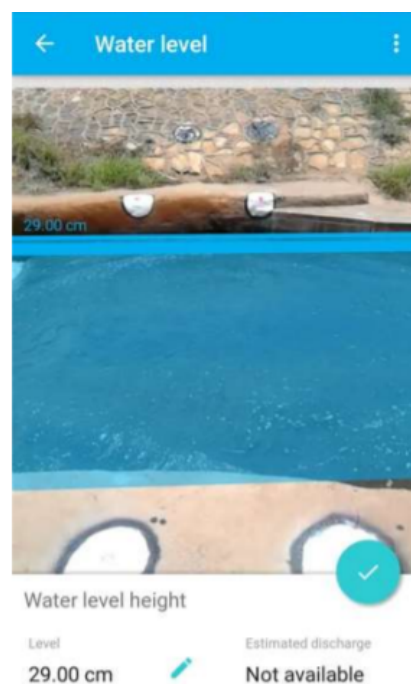


FIGURE 2.4 – Sélection de la zone bleue qui permet de déterminer le niveau d'eau [18]

4. La **résolution** de la vidéo qui sera prise peut également être modifiée entre 1080p et 720p. Avec un signal suffisamment robuste, une résolution plus basse peut-être essayée sans diminuer la précision des résultats [18].
5. **Filmer** la portion du cours d'eau où les quatre balises doivent être visibles à tout moment. L'écran, affichant la vidéo, est divisé en quatre parties, chacune devant contenir graphiquement une balise. Il faut également veiller à ce qu'il y ait un angle compris entre 20 et 70 degrés entre l'horizontale et la ligne d'eau de la berge opposée. De plus, l'écoulement doit contenir des particules suffisamment visibles pour être tracées lors de l'analyse PIV. La vidéo dure cinq secondes et sera ensuite enregistrée. Une fois la vidéo enregistrée, les quatre zones apparaissent une à une et il est demandé à l'utilisateur de situer plus précisément la position des balises.
6. Une zone bleue s'affiche alors sur la première image de la vidéo. L'utilisateur peut ajuster ses limites pour qu'elle couvre toute la surface du cours d'eau visible sur la caméra (voir figure 2.4). Il est possible de zoomer pour ajuster plus précisément la position de la zone bleue. Cette zone permet à l'algorithme de **déterminer la hauteur d'eau**. Celle-ci peut également être entrée manuellement.

1. Le rayon hydraulique est défini comme le rapport entre l'aire de la section et le périmètre de celle-ci.

7. Enfin, trois options se présentent.

- La vidéo filmée peut être supprimée si elle ne convient pas.
- La vidéo peut être stockée sur la plateforme en ligne sans réaliser l'analyse de vitesse.
- Les vitesses des particules de la vidéo ainsi que le débit sont calculés sur base de la vidéo (voir figure 2.5). Les vitesses sont calculées à partir de la technique Surface Structure Imaging Velocimetry (SSIV). Cette méthode est semblable à une analyse LSPIV mais se base sur les particules présentes naturellement dans le cours d'eau. Cela nécessite que ces particules soient relativement visibles. Pour cela, une phase de preprocessing où les reflets de l'eau ainsi que les ombres temporaires sur le cours d'eau sont supprimés peut être réalisée, rendant cette technique plus robuste [19].

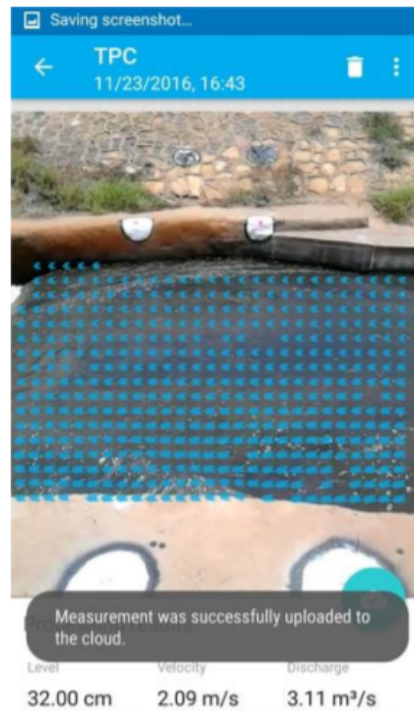


FIGURE 2.5 – Affichage des vecteurs vitesses calculés [18]

Les résultats peuvent ensuite être chargés sur la plateforme web.

Avantages et inconvénients

L'avantage de cette application est qu'elle permet de déterminer les vitesses très rapidement. En effet, les résultats des vitesses calculées et de débit peuvent être obtenus en à peu près deux minutes. De plus, cette application est simple d'utilisation et facile d'accès. Elle peut être utilisée par tout public.

Cependant, l'utilisation de Discharge, même si elle est présentée comme rapide et simple, comporte un certain nombre de contraintes qui ne sont pas toujours simples à mettre en oeuvre. En effet, la vidéo doit être prise sous un certain angle par rapport à l'horizontale, le site doit être accessible pour pouvoir poser les balises et celles-ci doivent être dans le même plan horizontal. Ces conditions dépendent des sites visités et ne sont pas forcément applicables. Il n'est malheureusement pas possible de réaliser l'analyse d'une vidéo filmée au préalable. L'analyse ne peut donc se faire qu'une seule fois, sur le terrain. En outre, les particules présentes naturellement dans le cours d'eau doivent être suffisamment visibles pour être tracées car l'algorithme de la SSIV est utilisé pour la détermination des vitesses.

4 RIVeR

RIVeR est une application gratuite qui permet de calculer le débit d'une rivière à partir d'images de celle-ci. Elle utilise deux techniques non intrusives d'analyse d'images : Particule Image Velocimetry (PIV) et Particule Tracking Velocimetry (PTV). La technique PIV est basée sur le suivi des déplacements d'un groupe de particules, suivant une analyse statistique. La technique PTV suit quant à elle les déplacements des particules individuellement. Les déplacements des particules issus de ces deux techniques permettront de déterminer les vitesses de surface de l'écoulement. RIVeR est un logiciel gratuit mais il est compilé dans l'environnement de Matlab qui doit être installé au préalable [20].

Démarche

1. Charger une **vidéo de l'écoulement**. L'utilisateur peut choisir l'intervalle de la vidéo qu'il souhaite analyser. Généralement, le passage choisi est celui où les particules sont les plus visibles. Il est également possible de choisir le nombre d'images par seconde à analyser pour réduire le temps de calcul [21]. Les images extraites peuvent également être grisées afin d'obtenir un meilleur contraste. L'utilisateur peut choisir de corriger les distorsions induites par la caméra.
2. **Analyse PIV** : Celle-ci fait appel à PIVlab pour une analyse PIV ou à PTVlab pour une analyse PTV. Voici un bref résumé des étapes de l'analyse :
 - Choix d'une région d'intérêt et de masques : cette étape n'est pas obligatoire mais permet de réduire considérablement le temps de calcul.
 - Application de filtres pour accentuer le contraste entre les particules à tracer et le fond de la rivière (voir figure 2.6). Deux filtres sont disponibles : CLAHE et passe-haut (voir section 1.4 du chapitre 3 pour plus d'informations).
 - Analyse de vitesse "single pass" ou "multi-pass".
 - Validation des vecteurs vitesse en choisissant les vitesses limites (voir figure 2.7).
 - Calcul des vitesses moyennes sur toutes les images.



FIGURE 2.6 – Région d'intérêt (zone noire) et masques (zones rouges) choisi, après application des filtres [21]

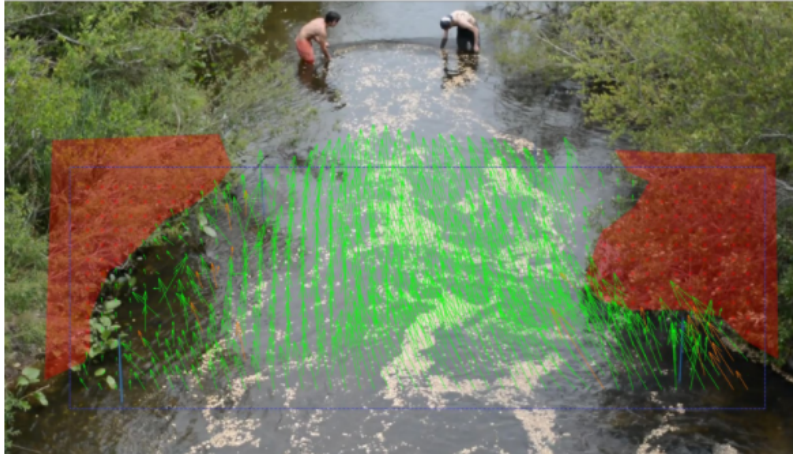


FIGURE 2.7 – Vecteurs vitesse après analyse PIV et validation [21]

3. **Orthorectification** des images. Celle-ci doit être réalisée dans le cas où la vidéo est prise de la berge. Quatre points, appelés points de contrôle, doivent être placés sur la vidéo (voir figure 2.8). Les longueurs entre ces points peuvent être fournies soit par un fichier excel soit manuellement. Ces points ne doivent pas non plus être alignés.

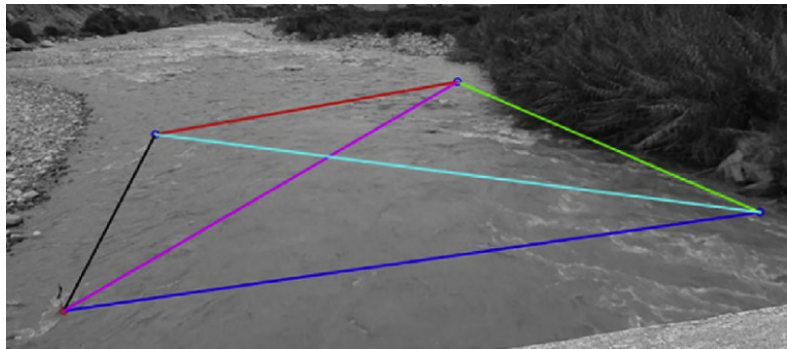


FIGURE 2.8 – Points de contrôles [21]

Ensuite, l'utilisateur sélectionne une nouvelle région d'intérêt (différente de celle sélectionnée auparavant) correspondant à la région à orthorectifier [21] (voir figure 2.9).



FIGURE 2.9 – Région d'intérêt sélectionnée pour l'orthorectification [21]

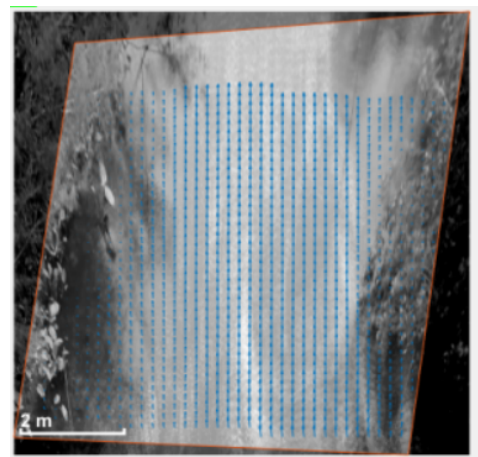


FIGURE 2.10 – Vecteurs vitesse de la région d'intérêt après orthorectification [21]

4. Le **débit** est calculé à partir des vitesses moyennes. Le niveau d'eau ainsi que la **bathymétrie**, sous la forme d'un fichier excel, doivent être importés. L'aire mouillée peut alors être déduite et elle sera subdivisée en n sous-sections de largeurs égales. L'utilisateur choisit ensuite la section, correspondant à la bathymétrie, à partir de laquelle le débit sera calculé. Celui-ci est ensuite multiplié par le coefficient de surface, posé à "1" par défaut mais dont la valeur peut être modifiée selon le site du cours d'eau [20].

Avantages et inconvénients

L'un des principaux avantages de cette application est qu'elle permet une orthorectification des images et des vecteurs vitesse. Cela élargit donc le champ des vidéos exploitables par les techniques non intrusives.

Cependant, l'inconvénient du logiciel RIVeR est qu'il fait appel à PIVlab pour l'analyse PIV et nécessite donc l'installation de Matlab.

5 Réflexions sur les logiciels existants

Il est important de noter que chaque logiciel existant présente certains avantages apportant une réelle valeur ajoutée mais aussi certains inconvénients qui compliquent l'utilisation de ceux-ci. Dans l'idéal, l'application sur laquelle nous avons travaillé devrait s'inspirer de ces avantages et être utilisable dans un maximum de situations différentes tout en présentant le moins d'inconvénients possibles.

A partir de l'état des lieux des logiciels existants réalisé aux sections précédentes, plusieurs grandes lignes directrices ont été définies et ont permis de donner un fil conducteur à notre travail et à nos objectifs pour l'application.

Le premier objectif souhaité est qu'elle ne nécessite pas l'intervention d'autres logiciels. Celle-ci devrait être **auto-suffisante** et donc munie d'outils permettant le travail de pré-processing (i.e. le traitement d'images) et le travail de post-processing, après l'analyse PIV. L'utilisateur ne devrait donc pas avoir à extraire les données de vitesses de l'application pour les traiter et obtenir un débit. Le but de l'application est que celle-ci prépare les images à l'analyse PIV et fournisse, elle-même, le débit du cours d'eau. Cet objectif permet un gain de temps et simplifie le processus d'obtention de débit en évitant à l'utilisateur de jongler entre plusieurs logiciels pas forcément adaptés les uns aux autres.

Ensuite, l'application doit être **user-friendly**. En effet, dans certains logiciels, l'utilisateur peut modifier beaucoup de paramètres qui ont chacun une forte influence sur le débit obtenu. Le résultat est donc fortement dépendant du choix de ces paramètres. Or, il n'est pas toujours simple, pour un utilisateur non aguerri, de comprendre le sens physique qui se cache derrière chaque paramètre ni l'influence de ceux-ci. L'idéal serait d'avoir une application où peu de paramètres modifiables interviennent et qui permettrait tout de même d'obtenir un débit proche du débit réel. Pour les paramètres restants, ceux-ci pourraient être initialisés à une valeur standard exemple tout en restant modifiables. Une explication de chaque paramètre et de son influence sur le débit pourrait également être fournie pour une utilisation plus adéquate. Il serait important d'avoir une application avec une interface permettant à l'utilisateur de pouvoir effectuer le processus d'obtention du débit sans devoir intervenir manuellement dans le code.

Enfin, dans certains logiciels, la vidéo à fournir pour l'analyse PIV doit être filmée en direct. L'analyse sur une même vidéo ne peut donc être réalisée qu'une seule fois. De plus, la vidéo doit être prise dans certaines conditions pour pouvoir être exploitée correctement par la suite.

Ces deux conditions restreignent le champ l'utilisation de ces logiciels. Il serait donc intéressant d'avoir une application qui n'imposerait **pas de condition spécifique sur la vidéo** à fournir et qui pourrait l'analyser autant de fois que l'utilisateur le souhaite.

Cette réflexion basée sur les différents logiciels nous guidera dans notre travail et dans les objectifs que l'application devra remplir.

Chapitre 3

Concepts utiles à l'application

Les différents concepts détaillés dans ce chapitre sont utilisés et implémentés dans l'application. Dans la première section, les concepts liés au traitement d'images appliqués à la vidéo en prévision de l'analyse de PIV seront traités. Dans les sections qui suivent, les concepts de calcul et les méthodes choisies à différentes étapes de l'application seront présentées.

1 Traitement d'images

1.1 Stabilisation

Le principe de stabilisation d'images a pour objectif de supprimer les mouvements non désirés générés par la caméra. En effet, si la caméra n'est pas maintenue par un stabilisateur lors de la prise d'images, celle-ci bouge (qu'elle soit maintenue par une personne ou bien par un drone). La stabilisation permet d'améliorer la netteté des images en supprimant l'effet de flou dû aux tremblements. Ceux-ci peuvent avoir un effet important sur l'apparence visuelle des images et donc sur les résultats de l'analyse PIV basée sur ces images. Il était donc obligatoire que l'application soit pourvue d'une étape réalisant la stabilisation des vidéos.

Une fonction `stabilization()` a donc été créée pour stabiliser les vidéos. Celle ci prend deux arguments en entrée :

- `filename` : la vidéo entrée par l'utilisateur.
- `newname` : le nouveau nom donné à la vidéo quand elle sera stabilisée, c'est-à-dire l'ancien nom de la vidéo suivi de l'expression *stabVersion*.

Lorsque l'utilisateur choisit de réaliser la stabilisation, la première image de la vidéo apparaît. Il est demandé à l'utilisateur de sélectionner quatre zones sur cette image. Ces régions doivent présenter certaines caractéristiques qui seront repérées sur cette première image et tracées dans les autres images de la vidéo par des traceurs d'un algorithme de la librairie OpenCV. Ces régions peuvent être des GCP ou bien des points fixes des berges. L'algorithme utilisé ici est un traceur KCF (Kernerlized Correlation Filters). Cet algorithme parcourt chaque image de la vidéo et cherche le centre de chaque zone sélectionnée puis enregistre les coordonnées de ceux-ci [22]. Ensuite, la fonction `cv2.getPerspectiveTransform()` est utilisée. Celle-ci effectue une transformation de perspective entre les coordonnées des centres trouvées pour chaque image de la vidéo et les coordonnées des centres des zones de la première image.

Dès lors, il faut remarquer que l'algorithme n'envisage pas le cas où une des zones sélectionnées n'apparaît plus sur une des images extraites de la vidéo (e.g. zoom dans une vidéo). En effet, elle ne pourra plus repérer le centre de cette zone [22]. Cependant, il n'est pas courant que ce genre de cas se produise. De plus, cet algorithme a l'avantage d'être rapide tout en maintenant une

certaine précision grâce à certaines propriétés mathématiques utilisées dans l'implémentation du traceur KCF.

1.2 Calibration

Certains types de caméra induisent des distorsions importantes dans les images. Étant donné que l'analyse PIV se base sur ces images, les distorsions peuvent avoir un impact non négligeable sur les résultats de débit obtenus avec l'application.

Le but de la calibration de la caméra est donc de corriger les distorsions induites grâce aux différents paramètres de la caméra. Cette étape permet également de lier précisément les points 2D de la projection de la caméra (i.e. pixel) aux points 3D correspondant du monde réel (i.e. mètre) [23].

Distorsions

Les distorsions causées par la caméra engendrent des déformations dans les images. Il existe deux types de déformations : la distorsion radiale et la distorsion tangentielle [24]. Ces distorsions sont caractérisées par des coefficients de distorsion qui seront définis plus tard dans la section 1.2 du chapitre 3.

La distorsion radiale est un phénomène qui fait apparaître les lignes droites comme des lignes incurvées (voir figure 3.1). Cette distorsion a un effet non-linéaire, c'est-à-dire que son effet est plus important sur les bords de l'image [25]. L'exemple le plus flagrant est celui des judas situés sur les portes d'entrée. La distorsion radiale peut être positive ou négative (voir figure 3.2).

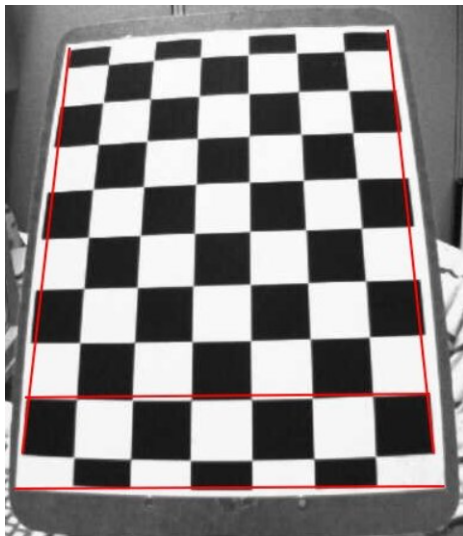


FIGURE 3.1 – Effet d'une distorsion radiale [24]

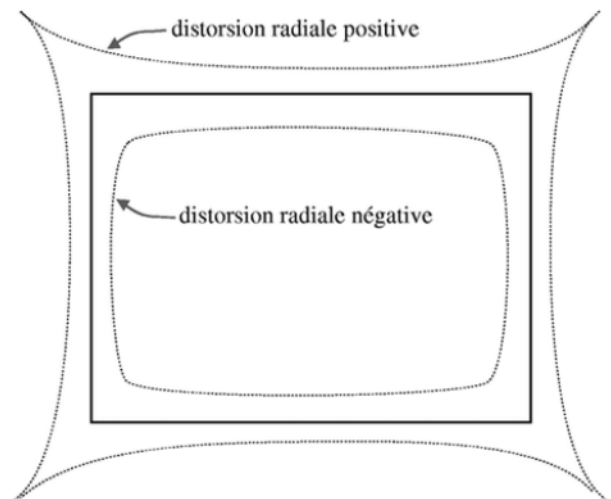


FIGURE 3.2 – Distorsion radiale positive et négative [26]

La distorsion tangentielle a pour effet de faire apparaître certaines régions de la photo plus proches qu'en réalité. Il s'agit d'une rotation autour du centre de l'image [25] causée par le fait que le plan de la lentille de la caméra et le plan de l'image ne sont pas exactement parallèles (voir figure 3.3). Cette distorsion a un effet beaucoup moins important que la distorsion radiale [25].

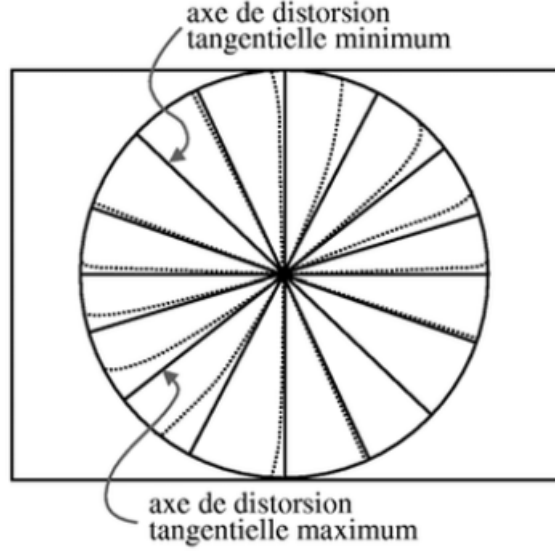


FIGURE 3.3 – Principe de la distorsion tangentielle [26]

Détermination des paramètres de calibration

Il existe deux types de paramètres [23] :

- Paramètres extrinsèques : Ces paramètres sont liés à l'orientation de la caméra dans le monde réel. Ils permettent la transition entre le monde réel et la caméra.
- Paramètres intrinsèques : Ces paramètres sont spécifiquement liés à la caméra. E.g. : distance focale, coordonnées du centre optique, coefficients de distorsion. Ils permettent la projection de la caméra sur l'image deux dimensions.

La transformation entre les coordonnées des points 3D (X_w, Y_w, Z_w) et leur projection plan 2D (x, y) est régie par l'équation suivante [23].

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = P \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad (3.1)$$

avec P la matrice de projection contenant les paramètres intrinsèques et extrinsèques.

$$P = \underbrace{K}_{\text{Matrice intrinsèque}} \cdot \underbrace{[R | t]}_{\text{Matrice extrinsèque}} \quad (3.2)$$

La matrice intrinsèque est une matrice 3x3 :

$$K = \begin{pmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (3.3)$$

avec f_x, f_y les distances focales selon x et y , γ l'angle entre les axes et (c_x, c_y) les coordonnées du centre optique en pixel.

La matrice extrinsèque est composée d'une matrice de rotation R et d'un vecteur de translation t [23].

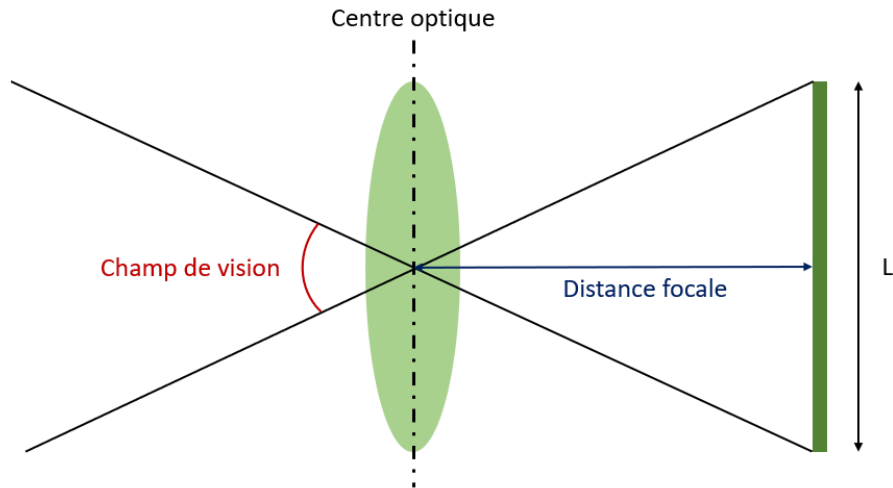


FIGURE 3.4 – Définition de la distance focale et du champ de vision

Pour déterminer les paramètres de calibration de la caméra, il est donc nécessaire de fournir les coordonnées connues des points 3D d'un objet du monde réel et 2D de l'image de cet objet. Dans le code, les points 3D seront appelés **object points** et les points 2D du plan **image points**.

Pour réaliser la calibration, une fonction `calibrateCamera()` a été écrite. Celle-ci utilise des fonctions de la librairie OpenCV. Cette librairie fournit plusieurs fonctions permettant d'utiliser trois modèles différents : le damier noir et blanc, le cercle symétrique et le cercle asymétrique [27]. Le modèle choisi est celui du damier noir et blanc (voir figure 3.1). Il faut ensuite fournir au minimum dix images (ici une vidéo) de ce damier prises sous différents angles et à différentes distances de celui-ci. Ces images permettront de déduire les points objets (3D) ainsi que les points images (2D) relatifs à ce damier.

Les points 3D sont définis par les coins des différents carrés du damier. Cependant, les points extrêmes du damier, soit ceux situés sur les bords, seront exclus. Le repère choisi est tel que l'axe Z est perpendiculaire au damier qui, fixé verticalement, contient les axes X et Y. Ainsi, la coordonnée selon Z vaut 0 pour chaque point du damier. Cela a un impact via l'équation de transformation 3.1 où la composante selon Z_w des points 3D ne sera pas prise en compte. Pour le reste, étant donné que les dimensions du damier ainsi que la taille des carrés sont connues, les coordonnées selon X et Y des points 3D peuvent être simplement déduites.

Pour trouver les points 2D, les coins du damier vont être détectés sur les images grâce à la fonction `cv2.findChessboardCorners()`. Cette fonction nécessite plusieurs arguments en entrée : l'image du damier et un vecteur contenant les dimensions du damier.

Ensuite, la fonction `cv2.calibrateCamera()` est utilisée pour déterminer les paramètres de calibration à partir des points objets et images trouvés juste avant. Ainsi, les vecteurs de rotation et de translation ainsi que la matrice de la caméra et les coefficients de distorsions sont désormais connus.

Le champ de vision (i.e. Field Of View), représentant l'angle de vue, est défini à partir des distances focales¹. Ce champ de vision peut être décliné selon trois axes : horizontal, vertical et diagonal. Les trois composantes sont calculées sur base des équations 3.4 et des formules trigonométriques [28].

1. La distance focale est la distance comprise entre le foyer d'un système optique et son centre optique (voir figure 3.4).

$$\begin{cases} FOV_x = 2\arctan \frac{L}{2 \cdot f_x} \\ FOV_y = 2\arctan \frac{H}{2 \cdot f_y} \\ FOV_{diag} = 2\arctan \frac{\sqrt{L^2 + H^2}}{2 \cdot f} \end{cases} \quad (3.4)$$

où H et L sont respectivement la hauteur et la longueur de l'image. f_x , f_y et f sont les distances focales selon les différents axes et sont issues de la matrice intrinsèque. f , la focale selon le champ de vision diagonal, est ici définie comme étant égale à f_x mais aurait pu être égale à f_y puisque f_x et f_y sont égales.

Enfin, la dernière étape de la fonction créée est le calcul de l'erreur commise lors de la détermination des paramètres. Pour cela, la fonction `cv2.projectPoints()` est utilisée pour calculer, au départ des paramètres trouvés, les points 2D. Ensuite, la distance entre ces points et les premiers points détectés sur le damier est calculé via la fonction `cv2.norm()`. L'erreur est simplement quantifiée comme la somme de ces distances. Dans le cas idéal, l'erreur doit être nulle. Cependant, si celle-ci est proche de zéro, les paramètres calculés seront tout à fait acceptables.

A partir de chaque image, de nouveaux points objet et image ainsi qu'une nouvelle équation sont créés. Plus le nombre d'images est important, plus le modèle d'équations devient précis [28]. C'est pour cette raison qu'un minimum de dix images est requis. Cependant, il est possible que l'algorithme ne détecte pas les coins du damier sur une ou plusieurs images. Pour pallier à ce problème, il est demandé à l'utilisateur de fournir une vidéo du damier selon plusieurs angles et à différentes distances ainsi que le nombre d'images qu'il veut d'extraire de la vidéo. Il est conseillé d'en extraire une vingtaine.

Processus de calibration

Une fois les différents paramètres trouvés, il faut enlever les distorsions induites par la caméra dans la vidéo. Une fonction `applyCalibrationOnVideoFrames()` a été créée. Celle-ci utilise la fonction `cv2.undistort()` de la librairie OpenCV avec les paramètres trouvés précédemment (la matrice contenant les distances focales et le centre optique ainsi que les coefficients de distorsions) pour supprimer les effets des différentes distorsions.

En sortie, la fonction créée fournit la vidéo où les distorsions ont été supprimées. Cette vidéo est alors renommée de son ancien nom avec la mention *Undistorted* ajoutée. Afin de rendre cette étape plus rapide, la calibration est réalisée seulement sur l'intervalle de la vidéo choisi par l'utilisateur. La nouvelle vidéo peut donc être plus courte.

Le processus de calibration est nécessaire pour appliquer la méthode "Angle et hauteur" qui permet la détermination du niveau d'eau. En effet, cette méthode utilise les différents paramètres de la caméra déterminés à la section précédente.

1.3 Orthorectification

Lors du développement de l'application, Simon Renard avait choisi de permettre à l'utilisateur de pouvoir l'utiliser sur une vidéo filmée de la berge. C'est un choix tout à fait logique, que nous soutenons. En effet, il est parfois plus facile de prendre une vidéo tout simplement de la berge que d'utiliser un drone qui prendra une vidéo parfaitement à la verticale. Seulement, cette prise de vue engendre des distorsions importantes dont il faut tenir compte et qu'il faut corriger. C'est là qu'intervient l'orthorectification.

L'orthorectification est une correction géométrique des images qui a pour but de les présenter comme si elles avaient été acquises depuis la verticale [29]. Elle correspond donc à une transformation géométrique qui effectue le lien entre le plan de l'image et le monde réel. Comme la partie de l'image que nous souhaitons corriger correspond à la surface d'eau dans le monde réel, on peut considérer que cette surface est un plan. Une transformation plane est donc suffisante. Il en existe plusieurs types tels que la translation, la rotation, l'effet d'échelle, le rapport d'aspect et le cisaillement [30]. Utiliser ces cinq transformations correspond à une transformation affine qui possède six degrés de liberté. Si on y ajoute la distorsion, on obtient une transformation homographique [28] dont la forme générale est l'équation 3.5, avec (X_p, Y_p) les coordonnées dans le plan de l'image, (X_r, Y_r) les coordonnées dans le plan du monde réel et la matrice H appelée matrice d'homographie. De plus, c'est une matrice qui a la propriété d'être inversible.

$$\begin{pmatrix} X_p \\ Y_p \\ 1 \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} X_r \\ Y_r \\ 1 \end{pmatrix} \quad (3.5)$$

$$\begin{cases} X_p = \frac{H_{11}X_r + H_{12}Y_r + H_{13}}{H_{31}X_r + H_{32}Y_r + H_{33}} \\ Y_p = \frac{H_{21}X_r + H_{22}Y_r + H_{23}}{H_{31}X_r + H_{32}Y_r + H_{33}} \end{cases} \quad (3.6)$$

La matrice d'homographie H possède neuf entrées inconnues, mais a-t-elle bien neuf degrés de liberté? Non, car si l'équation matricielle est réécrite comme le système d'équation 3.6, on note que l'on peut multiplier chaque H_{ij} par un coefficient non nul k sans changer les équations [31]. Le terme H_{33} peut ainsi être fixé à 1. Le système possède donc bien huit degrés de liberté et devient l'équation 3.7. Il suffit alors de connaître les coordonnées de quatre points dans le monde réel ainsi que leur correspondant sur l'image pour obtenir la matrice d'homographie H [32].

$$\begin{pmatrix} X_p \\ Y_p \\ 1 \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & 1 \end{pmatrix} \begin{pmatrix} X_r \\ Y_r \\ 1 \end{pmatrix} \quad (3.7)$$

Ces quatre points de références, généralement appelés Ground Control Points (GCPs), doivent être placés sur le même plan que celui formé par le plan d'eau et être répartis aux quatre extrémités de la surface étudiée (voir figure 3.7 à la page 27). Il peut également s'agir d'éléments naturels, tant qu'ils sont parfaitement reconnaissables et que leurs positions sont connues. Dans l'application, les six distances renseignées ainsi que l'ordre des quatre GCPs doivent respecter la convention représentée à la figure 3.5.

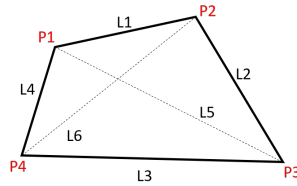


FIGURE 3.5 – Convention de distance utilisée dans l'application

$$\begin{cases} L_1^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 \\ L_2^2 = (x_3 - x_2)^2 + (y_3 - y_2)^2 \\ L_3^2 = (x_4 - x_3)^2 + (y_4 - y_3)^2 \\ L_4^2 = (x_4 - x_1)^2 + (y_4 - y_1)^2 \\ L_5^2 = (x_3 - x_1)^2 + (y_3 - y_1)^2 \\ L_6^2 = (x_4 - x_2)^2 + (y_4 - y_2)^2 \end{cases} \quad (3.8)$$

Les coordonnées de ces quatre GCPs peuvent être facilement retrouvées à partir des six distances qui les séparent et qui sont renseignées par l'utilisateur. Dès lors, nous obtenons six équations, dont une qui peut être déduite à partir des autres, pour retrouver les huit coordonnées inconnues. Il y a donc trois degrés de liberté qu'il faut fixer pour résoudre le système. L'astuce choisie dans l'application est de fixer le point 4 à (0,0) et de considérer que le point 1 est aligné verticalement avec celui-ci. En d'autres termes, on fixe x_1 à 0.

Les coordonnées des quatre points de références sont donc connues dans le monde réel, exprimées en mètres, ainsi que dans le plan de l'image, exprimé en pixel. Il reste à exprimer le jeu de coordonnées du monde réel en pixels en les multipliant par un facteur d'échelle, nommé `scaling_factor` dans l'application. Finalement, le package OpenCV permet d'utiliser la fonction `cv2.findHomography()` qui prend en arguments les deux jeux de coordonnées (issus de l'image et du monde réel) et renvoie la matrice d'homographie H en sortie. Seulement, cv2 ne travaille pas avec des coordonnées négatives et son origine est située dans le coin supérieur gauche de l'image. Les quelques lignes de code ci-dessous (figure 3.6) permettent de retravailler les coordonnées pour qu'elles correspondent au système de coordonnées utilisé par cv2.

```

if min(X) < 0:
    X = X - min(X)
if min(Y) < 0:
    Y = Y - min(Y)
Y = abs(Y - max(Y))

```

FIGURE 3.6 – lignes 29 à 33 de `orthorectificationComputation()`

Maintenant que la matrice d'homographie est calculée, il y a deux façons de l'utiliser. En utilisant son inverse, nous pourrions soit orthorectifier chaque image de la vidéo, effectuant ainsi l'analyse LSPIV sur base d'images orthorectifiées soit orthorectifier les vecteurs vitesses. Selon la résolution de la caméra, une image peut contenir des millions de pixels alors qu'on ne compte que quelques milliers de vecteurs vitesses à la sortie de l'analyse LSPIV. Il sera donc beaucoup plus rapide d'utiliser l'inverse de la matrice d'homographie pour corriger les vecteurs vitesses calculés sur bases des images déformées. Selon l'article de Antoine Patalano, Carlos Marcelo Garcia et Andrés Rodríguez, ce raisonnement est également utilisé dans l'application RIVeR [33]. La première image de la vidéo après orthorectification sera tout de même conservée pour une question d'affichage.

Scaling factor

Comme expliqué précédemment, le scaling factor est une valeur fixe qui peut être choisie arbitrairement dont les unités sont des pixels par mètre ($[\frac{px}{m}]$). Comme son nom l'indique, il s'agit bien d'un facteur d'échelle qui fait le lien entre les pixels sur l'image et les mètres du monde réel. Il est également important de préciser que ce lien se fait entre l'image redressée par l'étape d'orthorectification et le plan d'eau réel. Il est donc bien unique, peu importe la position de l'image que l'on considère puisque les déformations ont été corrigées. La seule contrainte qui lui est imposée est d'ordre graphique. En effet, s'il est trop grand, les quatre GCPs ne seront pas tous visibles dans la fenêtre de l'application. Par contre, s'il est trop petit les GCPs se confondront dû à un écart en pixel trop faible. Le compromis implémenté dans l'application est de faire une moyenne des rapports des six distances en pixels avec les six distances correspondantes renseignées en mètres.

Ce facteur d'échelle est ensuite réutilisé tout au long de l'application :

1. Il permet de déterminer le niveau d'eau en recoupant la largeur d'eau avec la bathymétrie, c'est le rôle de la méthode image décrite à la section 2.2 du chapitre 3 (p. 36).
2. Il permet le changement d'unité des vecteurs vitesses initialement exprimés en $[\frac{m}{s}]$.

Exemple

Les figures 3.7 et 3.8 sont un exemple de l'utilisation de l'orthorectification dans l'application. On peut également comparer la figure 3.8, qui est l'image obtenue suite à l'orthorectification, avec la figure 3.9, qui est une image prise avec un drone de façon parfaitement orthogonale au plan d'eau.



FIGURE 3.7 – Image initiale

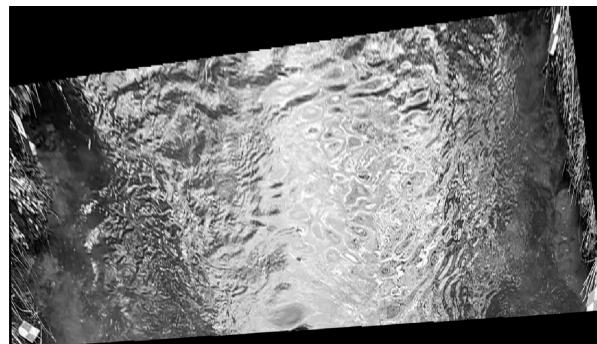


FIGURE 3.8 – Image orthorectifiée

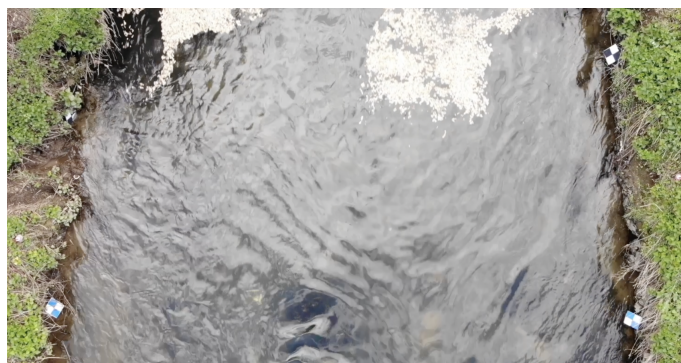


FIGURE 3.9 – Vue orthogonale

1.4 Filtres

L'application de filtre(s) a pour but de contraster les particules dispersées dans le cours d'eau par rapport au fond de la rivière avant de procéder à l'analyse PIV. Leurs déplacements pourront alors être mieux détectés par l'algorithme de l'analyse PIV. Les résultats de cette analyse seront donc plus précis.

Pour appliquer le filtre choisi sur une image de la vidéo, une fonction `filtersImg()` a été créée. Celle-ci prend en argument une image, le filtre choisi ainsi que les paramètres liés à ce filtre et retourne l'image filtrée. Cependant, l'utilisateur a la possibilité de choisir d'appliquer plusieurs filtres sur la même vidéo. Dans ce cas, les filtres sont stockés dans une liste. Pour appliquer ces différents filtres sur toute la vidéo, la fonction `applyFilterOnVideoFrames()` a été écrite. Celle-ci permet d'appliquer les différents filtres choisis sur la vidéo entière ou sur l'intervalle de temps choisi dans le cas où l'utilisateur l'a précisé auparavant. Cette fonction retourne la vidéo filtrée qui sera alors utilisée pour l'analyse PIV.

Les différents filtres implémentés dans l'application sont détaillés dans les sections suivantes. Comme annoncé précédemment, chaque filtre possède des paramètres initialisés au départ à des valeurs par défaut. L'utilisateur aura la possibilité de modifier ceux-ci en fonction du contraste qu'il souhaite voir apparaître.

CLAHE

Le filtre CLAHE soit Contrast Limited Adaptive Histogram Equalization, est basé, comme son nom l'indique sur le principe d'égalisation d'un histogramme. Cette méthode est principalement utilisée sur des images présentant un contraste très faible. Par exemple, sur certaines images, l'avant plan et l'arrière plan sont tous les deux dans les mêmes tons. Pour ces images, l'intensité des pixels est confinée dans une seule région au lieu de s'étendre dans tout le spectre disponible. Afin de mieux répartir les intensités des pixels, l'histogramme sera étiré de manière à couvrir un plus large champ de valeurs, ce qui augmentera le contraste des images [34].

La librairie OpenCV offre une implémentation de ce filtre via la fonction `cv2.createCLAHE()` [34]. L'algorithme utilise une fonction qui étend l'intensité des pixels restreints dans une zone à tout le spectre selon le principe d'égalisation d'un histogramme. Cependant, comme expliqué précédemment, ce principe est uniquement adapté pour les images dont l'ensemble des pixels reste dans les mêmes tons. En cas de contraste important sur l'image, l'algorithme n'est pas capable d'étendre les valeurs d'intensités sur un plus large spectre. Par exemple, les zones claires initialement deviendront surexposées.

Pour pallier à ce problème, une méthode "adaptative" a été ajoutée dans le filtre. Celle-ci divise la photo en une grille de plusieurs carreaux dont les intensités des pixels sont proches. L'égalisation de l'histogramme peut ainsi être réalisée sans problème de contraste. La taille de toutes ces carreaux (appelée "tile grid size" dans l'application) est un paramètre du filtre.

Toutefois, il est courant que les différentes régions présentent du bruit² qui va alors être amplifié par cette méthode de division. Une dernière méthode a donc été ajoutée au filtre : la limitation de contraste. Toutes les valeurs de l'histogramme d'intensité seront limitées en dessous d'une certaine valeur de pixel, appelée "clip limit" dans l'application, fixée par défaut (à 2 initialement) mais que l'utilisateur a la possibilité de changer. Si les pixels présentent une valeur supérieure à cette limite, ils seront déplacés pour compléter la distribution uniforme de l'histogramme avant que le processus d'égalisation n'ait lieu.



FIGURE 3.10 – Image originale



FIGURE 3.11 – Application d'un filtre CLAHE

2. Le bruit visuel est un parasite qui dégrade la qualité de l'image.

Intensity capping

Certains écoulements possèdent des points plus clairs qui influencent l'analyse statistique réalisée lors de l'analyse PIV et diminuent la précision des vecteurs vitesses calculés [16]. L'objectif de ce filtre est de rejeter ces points d'intensité trop élevée par rapport à une certaine valeur [35]. Pour cela, un plafond est appliqué à tous les pixels. Si leur intensité est supérieure à ce plafond, celle-ci sera réduite à la valeur de la limite. Dans l'application, la valeur de cette limite est définie initialement comme deux fois l'écart-type ajouté à la médiane [28]. Cependant, si le résultat affiché n'est pas convaincant, l'utilisateur a la possibilité de modifier cette limite.

Ce filtre est particulièrement adapté aux situations où l'écoulement n'est pas illuminé uniformément. Après l'application du filtre, les traceurs contribueront de manière plus uniforme à l'analyse statistique. De plus, il a l'avantage de ne modifier que très peu l'intensité des pixels de l'image limitant les conséquences néfastes d'une modification de l'image [36].



FIGURE 3.12 – Image originale



FIGURE 3.13 – Application d'un filtre Intensity capping (paramètre fixé à 190)

Intensity high-pass

Le filtre passe-haut a pour rôle, comme son nom l'indique, de laisser passer les hautes fréquences tout en bloquant les fréquences basses. Les fréquences sont considérées comme basses lorsqu'elles sont inférieures à une certaine fréquence de coupure. Étant donné que les fréquences hautes correspondent en grande majorité aux particules du cours d'eau, cela a pour objectif d'uniformiser l'arrière-plan du cours d'eau et de mettre en évidence ces particules [16]. L'image sera ainsi plus nette après application du filtre passe-haut.

Le filtre passe-haut a été implémenté en trois étapes dans l'application [28]. Tout d'abord, un filtre passe-bas est appliqué à l'image colorée en gris auparavant via la fonction `cv2.GaussianBlur()`. Le filtre passe-bas utilise un noyau gaussien de convolution. Cet algorithme recalcule la valeur de l'intensité des pixels à partir de sa propre valeur et des valeurs des intensités

des pixels voisins. Les intensités des pixels voisins contribuent au calcul via des coefficients de pondération qui sont issus du noyau gaussien.

Ensuite, on soustrait l'image soumise au filtre passe bas à l'image grisée via la fonction `cv2.subtract()`. En effet, un filtre passe-haut n'est jamais que l'inverse d'un filtre passe-bas.

Enfin, chaque pixel est multiplié par un facteur d'amplification qui permet de restaurer le contraste atténué lors de la soustraction, à l'étape précédente.

Les paramètres du filtre passe-haut sont le facteur d'amplification et les paramètres relatifs au noyau gaussien.



FIGURE 3.14 – Image originale

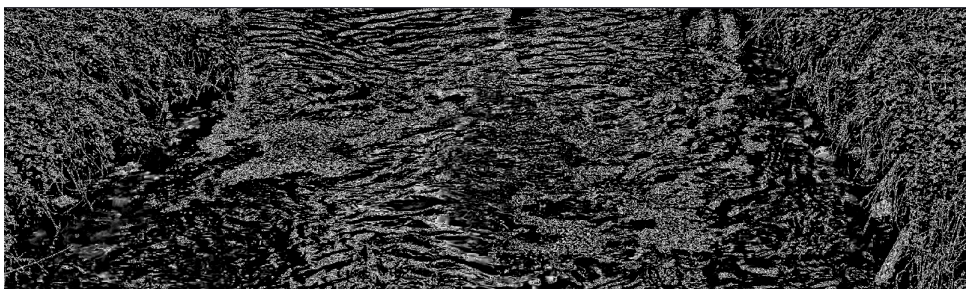


FIGURE 3.15 – Application d'un filtre Intensity high-pass (paramètre fixé à 10)

Mean Denoising

L'arrière plan d'une image présente souvent du bruit qui a tendance à biaiser les résultats de l'analyse PIV. Dans l'application, la fonction `cv2.fastNlMeansDenoising()` est utilisée. Celle-ci applique un algorithme qui permet de supprimer le bruit sur une image colorisée dans les tons gris. Le principe consiste à remplacer un pixel par la moyenne des zones similaires à ce pixel [37]. Cependant, les régions similaires ne sont pas toujours proches les unes des autres et doivent donc être recherchées dans un périmètre assez large de l'image. C'est pour cette raison que l'application de ce filtre prend un certain temps en comparaison aux autres filtres. Cependant, il présente de bons résultats.

Les paramètres de ce filtre sont sa puissance ainsi que la taille de la fenêtre et la taille de la fenêtre de recherche.



FIGURE 3.16 – Image originale



FIGURE 3.17 – Application d'un filtre mean denoising avec les paramètres par défaut

Contrast/Brightness

Ce filtre a pour objectif de corriger le contraste et la luminosité qui sont parfois accentués lors de la surexposition à la lumière. Pour cela, l'application fait appel à la fonction `cv2.convertScaleAbs()`. Chaque élément de l'image est mis à l'échelle, pris en valeur absolue et convertit en corrigeant contraste et luminosité grâce à l'équation suivante :

$$g(i, j) = \alpha \cdot f(i, j) + \beta \quad (3.9)$$

Après application, il est probable que l'intensité de certains pixels ne soit pas comprise entre 0 et 255. Dans ce cas, ces valeurs sont arrondies à la valeur la plus proche de cette intervalle.

Les paramètres sont les coefficients α (correction du contraste) et β (correction de la luminosité).



FIGURE 3.18 – Image originale



FIGURE 3.19 – Application d’un filtre Contrast/Brightness (paramètres α et β fixés à 1.3 et 1)

Gamma Correction

Cette méthode corrige la luminosité d’une image en transformant les pixels vers de nouvelles valeurs au moyen de la relation suivante :

$$O = 255 \cdot \left(\frac{I}{255} \right)^\gamma \quad (3.10)$$

avec I l’intensité initiale des pixels [38]. A nouveau, si l’intensité des pixels n’est pas comprise entre 0 et 255, ces valeurs sont arrondies à la valeur la plus proche de cette intervalle.

L’utilisateur a la possibilité de modifier la valeur du paramètre γ . Si $\gamma > 1$, l’histogramme sera décalé vers la gauche et l’intensité des pixels sera plus foncée. Si $\gamma < 1$, l’histogramme sera décalé vers la droite et l’intensité des pixels sera plus claire.



FIGURE 3.20 – Image originale



FIGURE 3.21 – Application d’un filtre Gamma avec $\gamma = 2$

2 Calcul de la hauteur d’eau

Trois méthodes ont été implémentées dans l’application pour déterminer le niveau d’eau si celui-ci n’est pas connu de l’utilisateur.

1. Méthode adaptée d'Hellebaut
2. Méthode image
3. Méthode angle et hauteur

La première se base sur la photogrammétrie et les deux suivantes sont basées sur la recherche de la largeur d'eau.

2.1 Méthode d'Hellebaut

Cette méthode a été déterminée par François-Xavier Hellebaut dans son mémoire. Elle consiste à déduire le niveau d'eau à partir de l'utilisation de la photogrammétrie [39]. Le principe de la méthode conçue par Hellebaut a été repris et simplifié dans l'application pour déterminer le niveau d'eau.

Photogrammétrie

La photogrammétrie peut être définie comme suit [40] :

La photogrammétrie est l'art d'obtenir des informations fiables sur les propriétés de surfaces ou d'objets sans contact physique avec ceux-ci et l'art de mesurer et d'analyser ces informations.

En résumé, les informations et propriétés relatives à un objet peuvent être déterminées à partir de photos de celui-ci. Dans le contexte de ce mémoire, la photogrammétrie sera utilisée pour reconstruire un modèle de la section de la rivière.

La photogrammétrie est apparue peu après la photographie. Selon le nombre et le type de photos, différentes sortes de photogrammétries peuvent être utilisées. Dans ce travail, les images digitales seront utilisées ainsi que la technique "multi-images". Cette technique est aussi appelée "Structure from Motion" (SfM).

Structure from motion

Cette technique a été développée il y a maintenant une trentaine d'années [41]. Elle permet de déterminer les points en 3D qui constituent un objet à partir de photos 2D superposées de cet objet (voir figure 3.22). Pour que le résultat produit par la SfM soit représentatif de la réalité, les photos doivent être nombreuses et couvrant le plus grand angle de vue possible. Plus il y aura de superpositions entre les zones photographiées, plus le modèle sera précis.

Le modèle 3D produit par cette technique est appelé "Digital elevation model" (i.e. DEM). Dans l'application, c'est celui-ci qui doit être fourni par l'utilisateur lorsqu'il choisit la méthode d'Hellebaut pour déterminer le niveau d'eau. Cette technique a émergé rapidement ces dernières années. Elle doit son succès au faible coût technique engendré par la modélisation menant toute de même à une haute résolution dans la topographie obtenue. Dans certaines conditions, les résultats obtenus sont équivalents en terme de qualité à un modèle construit à partir de techniques de détection réelles. Grâce à cette technique, les chercheurs peuvent reconstruire un modèle de manière démocratique lorsqu'ils ont besoin de données en trois dimensions [41]. De plus, les résultats sont obtenus rapidement.

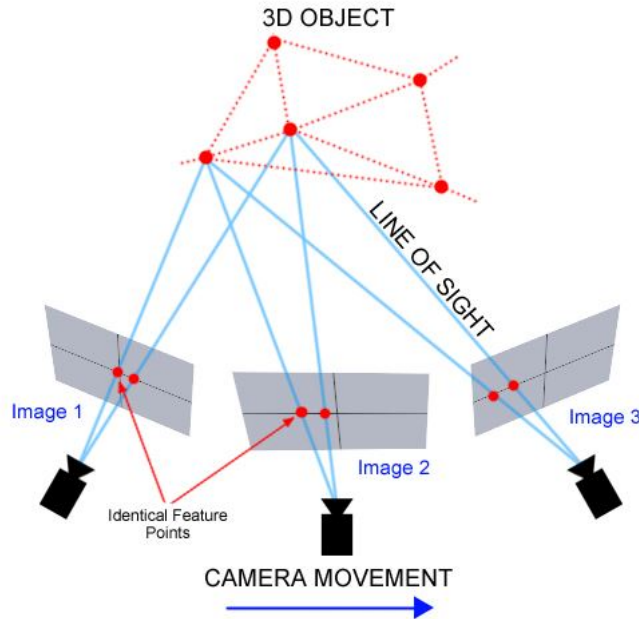


FIGURE 3.22 – Illustration de la technique SfM [42]

Pour pouvoir utiliser la technique SfM, il faut connaître les positions et orientations exactes de la caméra qui prend ces photos. Les paramètres intrinsèques et extrinsèques sont déterminés par une calibration faite par la SfM à partir des photos prises.

D'un point de vue logiciel, la technique SfM est implémentée dans plusieurs de ceux-ci. Le logiciel Agisoft Photoscan, utilisé dans les mémoires précédents, permet de générer un DEM à partir de la technique SfM. Il possède une interface user friendly qui simplifie son utilisation mais rend plus difficile l'accès aux informations précises. Le logiciel peut être comparé à une "boîte noire".

Travail d'Hellebaut

Le travail d'Hellebaut peut être divisé en différentes étapes [39] :

- Il a utilisé la photogrammétrie pour déterminer la topographie de la rivière à partir d'une série de photos de la berge où l'eau de la rivière était visible.
- Il a ensuite comparé la bathymétrie de la rivière avec la topographie obtenue par photogrammétrie et s'est rendu compte que le point de divergence entre ces deux courbes correspondait au niveau d'eau (voir figure 3.23). En effet, la technique SfM modélise correctement la berge jusqu'à l'apparition de la surface de l'eau qui vient biaiser les résultats à cause de la réflexion à la surface de celle-ci.

Hellebaut a ensuite déterminé deux paramètres, qu'il a fixés empiriquement, pour aider à détecter le point de décrochage. Le premier est appelé "Count" et représente le nombre de points. Le second est appelé "Difference" et correspond au seuil représentant une différence de hauteur.

Il a ensuite imposé deux conditions :

- La dérivée de la courbe de Difference doit être positive si le Count des points va de gauche à droite.
- Les différences sur les points du Count doivent être plus grandes que le paramètre Difference.

Enfin, après plusieurs tests en laboratoire, Hellebaut a pu tirer plusieurs constats [39] :

- Il est plus facile de ne modéliser qu'une seule des deux rives. En effet, lorsqu'il tente d'avoir un modèle où les deux rives sont présentes, cela nécessite que les deux rives soient visibles sur les photos à entrer pour créer ce modèle. Lorsque la rivière est large, cela peut rapidement devenir complexe voir impossible.
- Il remarque aussi que la végétation sur la berge peut poser problème dans la détermination de la topographie. Il ajoute donc certains paramètres pour s'adapter à cette situation.
- A la comparaison des deux courbes, la détection du point de décrochage est plus efficace quand on part du centre de la rivière vers berge.

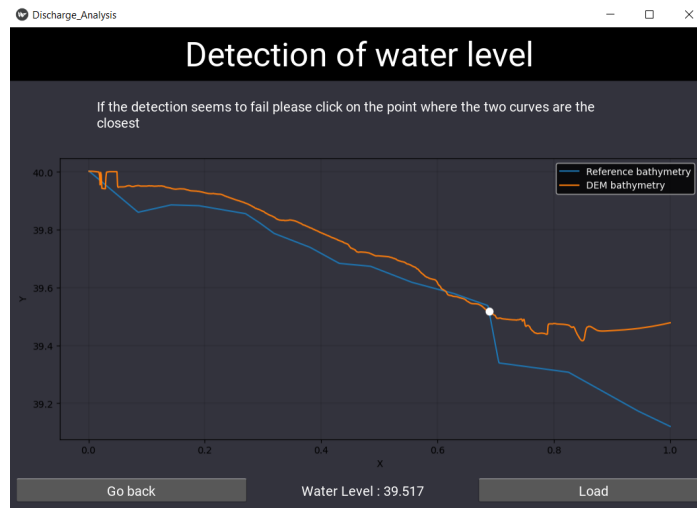


FIGURE 3.23 – Comparaison entre la bathymétrie et le topographie issue du DEM. Le point blanc est le point de décrochage.

Adaptation de la méthode pour l'application

Dans l'application, la méthode d'Hellebaut détaillée ci-dessus a été adaptée. Elle est simplement implémentée comme la détection du point où les courbes du modèle digital d'élévation (DEM) et de la bathymétrie sont les plus proches. Cette première détection a pour but de donner une idée générale de la région de localisation de ce point. Si celle-ci ne correspond pas au point précis, l'utilisateur aura alors la possibilité de déplacer le point [28].

L'algorithme a été implémenté dans l'application de la manière suivante. Tout d'abord, en chaque point, pour la même abscisse x , la valeur absolue de la différence entre le DEM et la bathymétrie $y(x)$. Les minima de ces différences sont ensuite stockés.

$$D(x) = |DEM(x) - y(x)| \quad (3.11)$$

Cependant, la courbe issue du DEM présente beaucoup d'oscillations (voir figure 3.23). Il y aura donc plusieurs minima et l'algorithme aura pour objectif de choisir le bon. Pour cela, plusieurs étapes ont été implémentées [28].

- Tous les minima entre les différences sont détectés en se déplaçant du centre de la rivière vers la berge. Étant donné que le DEM issu de la photogrammétrie reflète quasiment le relief de la berge si l'on ne considère que la partie où il n'y pas d'eau, les différences par rapport à la bathymétrie seront faibles. Les minima présents sur cette partie du DEM auront donc de faibles valeurs. La partie du DEM contenant la berge sera plus importante que celle où l'eau est présente car il n'est pas nécessaire de comparer le DEM à la section entière.

- La moyenne des minima est ensuite calculée. Celle-ci servira de seuil. Sa valeur risque d’être relativement basse puisque la partie contenant la berge (où les valeurs de minima sont faibles) est plus importante que la partie où l’eau se trouve et où les minima seront plus élevés.
- Chaque minimum est alors comparé au seuil calculé. Le premier minimum détecté en dessous de cette moyenne est considéré comme le point de décrochage.

Il y a donc deux conditions pour que cet algorithme détecte correctement le point de décrochage. Tout d’abord, les courbes du DEM doivent être relativement saccadées afin de présenter un nombre de minima suffisant pour couvrir toute la largeur de la berge. La seconde condition est que le DEM soit relativement proche de la bathymétrie sur la partie de la section sans eau. Cela permet d’assurer que les minima soient suffisamment faibles pour que le seuil (donc la moyenne des minima) ait une valeur basse.

Si l’algorithme échoue et que le résultat ne semble pas réaliste à l’utilisateur, celui-ci a la possibilité de sélectionner le point de décrochage tout en ayant une idée de l’endroit où celui-ci risque de se trouver.

Cependant, il existe plusieurs inconvénients à cette méthode. Tout d’abord, elle nécessite des points de contrôle (Ground Control Point i.e. GCP) pour exprimer le DEM dans le même repère que la bathymétrie afin de permettre une comparaison. De plus, la méthode nécessite un DEM, donc l’utilisation d’un autre logiciel et un certain nombre de photos pour produire celui-ci.

2.2 Méthode image

Pour utiliser cette méthode, il faut qu’une orthorectification ait été réalisée au préalable. Ensuite, sur l’image orthorectifiée, l’utilisateur sélectionne deux points (un sur chaque berge). Ces deux points correspondent aux limites de la largeur de la rivière. La distance en pixels entre ces deux points est calculée et transformée en mètres grâce au facteur d’échelle calculé durant l’étape d’orthorectification. Cette largeur est ensuite importée dans la bathymétrie chargée lors de la première étape grâce à la fonction `findWaterLevel()` créée à cet effet.

Cette fonction prend comme argument les coordonnées des points de la bathymétrie ainsi que la largeur calculée en mètres. L’algorithme fonctionne de la manière suivante. Partant des points les plus hauts de la bathymétrie vers le bas, la distance horizontale entre les points extrêmes est calculée. Cette distance est comparée à la largeur calculée. Si celle-ci correspond, cela signifie que l’intersection entre la largeur et la bathymétrie se trouve à cet endroit. Le niveau d’eau est donc aisément déduit grâce à cette intersection (voir figure 3.24).

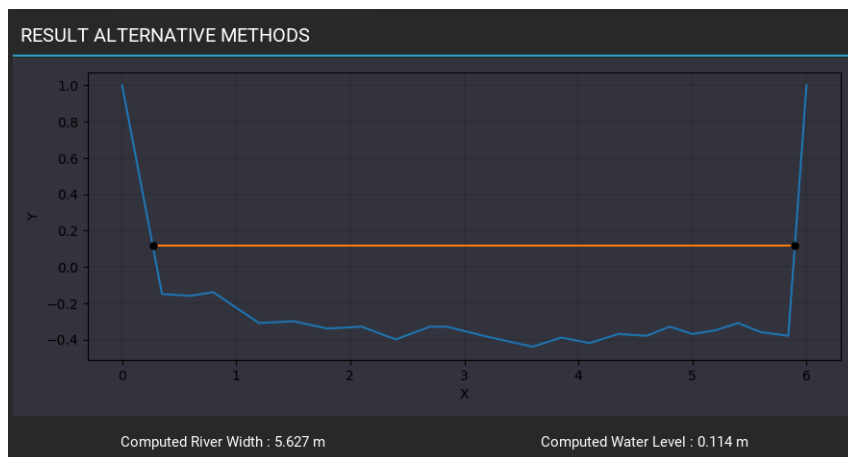


FIGURE 3.24 – Recherche du niveau d’eau dans la section

Cependant, cette méthode ne fonctionne pas dans tous les cas. En effet, si la section est rectangulaire, c'est-à-dire que les berges sont complètement verticales, ou que les berges ne sont que très peu inclinées par rapport à la verticale, la largeur varie très peu pour une hauteur d'eau très différente. De plus, dans le cas où la bathymétrie présente une berge plus haute que l'autre, l'algorithme sera dans l'incapacité de calculer une distance horizontale entre les points les plus hauts, puisque, pour la même ordonnée, il n'y a qu'un seul point. Ces deux cas de bathymétrie ne peuvent donc pas être pris en compte par une méthode basée sur la largeur.

2.3 Méthode angle et hauteur

Pour utiliser la méthode angle et hauteur et fournir les données utiles à celle-ci, il est préférable d'utiliser un drone. L'utilisateur doit entrer les éléments suivants : deux photos de l'écoulement, prises sur le même axe vertical mais à différentes altitudes, la distance entre ces deux positions ainsi que le champ de vision (FOV). Pour connaître ce dernier, une calibration de la caméra doit être réalisée au préalable. Sur base des distances focales, le champ de vision est déduit grâce aux formules détaillées dans la section 1.2 de ce chapitre relative à la calibration (voir équations 3.4).

Le principe de cette méthode se base sur la trigonométrie pour trouver la largeur d'eau. Le niveau est alors déduit à partir de la largeur calculée et de la bathymétrie, de manière identique à la méthode image.

Deux cas ont été implémentés dans l'application. Le premier est celui où le drone est placé à la moitié de la largeur de la rivière. Cette position apporte une contrainte supplémentaire sur la position du drone mais a l'avantage d'être calculée plus simplement. Le second cas est celui où le drone n'est pas centré par rapport à la largeur de la rivière. En réalité, il est difficile de placer le drone exactement au centre. La réalité sera donc plus proche de ce cas. Cependant, les équations deviennent plus complexes et la résolution moins facile. Dans l'application, les deux cas sont envisagés permettant d'avoir une solution de repli si le second cas ne fonctionne pas.

Drone centré

Cette configuration est illustrée à la figure 3.25.

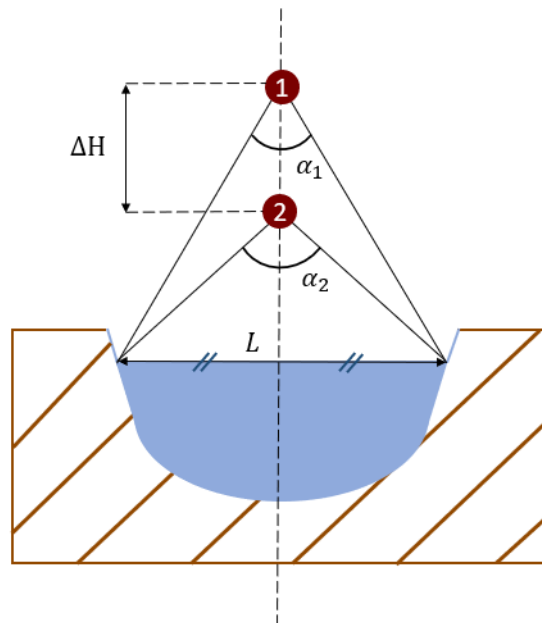


FIGURE 3.25 – Cas du drone positionné au centre de la largeur

Tout d'abord, les deux angles α_1 et α_2 peuvent être déduits au moyen de la formule suivante :

$$\alpha = FOV \cdot \frac{\#Pixel_L}{\#Pixel_{img}} \quad (3.12)$$

où $\#Pixel_L$ et $\#Pixel_{img}$ désignent respectivement la largeur de la rivière en pixels et la taille de l'image en pixels. Cette formule est correcte tant que l'on fait une approximation de degré 1 de son développement en série de Taylor, ce qui introduit la condition $|\frac{L}{L_{tot}} \tan(\frac{FOV}{2})| \leq 1$. Une correction a donc été effectuée et sera expliquée dans la section 3 du chapitre 7.

La largeur de la rivière en pixels, utilisée dans la formule 3.12, est déterminée par la distance entre les deux points placés sur chaque rive de la rivière sur l'image par l'utilisateur. La taille de l'image en pixels est quand-à-elle propre aux propriétés de l'image.

La seconde étape est de déterminer la largeur réelle. Selon la géométrie exprimée à la figure 3.25, les deux angles α peuvent être exprimés en fonction de cette largeur.

$$\begin{cases} \tan(\frac{\alpha_1}{2}) = \frac{L/2}{H + \Delta H} \\ \tan(\frac{\alpha_2}{2}) = \frac{L/2}{H} \end{cases} \quad (3.13)$$

Les seules inconnues sont H, la hauteur à laquelle la première photo a été prise et L la largeur d'eau de la rivière. Ces deux inconnues peuvent être isolées dans ces deux équations. La largeur en mètre est ainsi déduite et permet de déterminer le niveau d'eau à partir de la bathymétrie.

Drône décalé du centre

Dans le cas où le drône est décentré par rapport à la largeur de la rivière (voir figure 3.26, les équations diffèrent.

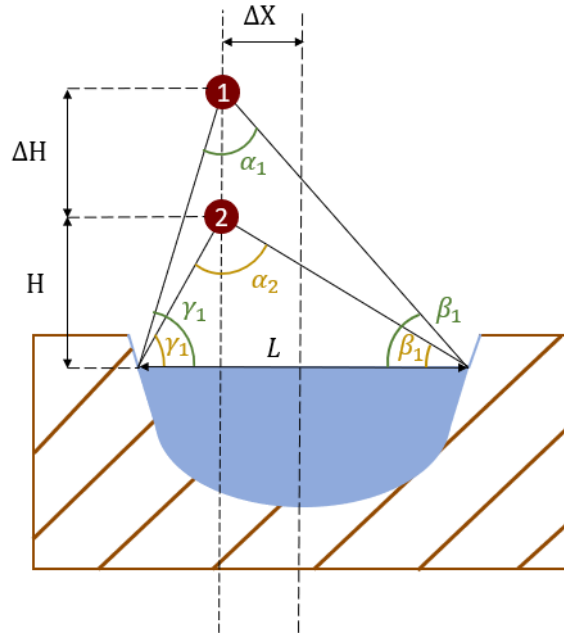


FIGURE 3.26 – Cas du drone décalé par rapport au centre de la rivière

Les angles α_1 et α_2 sont calculés à partir des formules 3.12 du cas centré. Or, dans le cas non centré, ces formules ne sont pas valables. Elles ont également été corrigées comme expliqué dans la section 3 du chapitre 7.

Les système d'équations relatifs à cette configuration sont plus complexes. Les inconnues que l'on souhaite connaître sont toujours la hauteur H et L la largeur de la rivière. Les indices 1 et 2

dans les équations représentent respectivement les indices relatifs à la photo prise à plus haute altitude et à la photo prise à plus basse altitude. La valeur de ΔX correspond au décalage par rapport à la position centrale de la largeur.

$$\begin{cases} \tan(\gamma_1) = \frac{H+\Delta H}{L/2-\Delta X} \\ \tan(\beta_1) = \frac{H+\Delta H}{L/2+\Delta X} \\ \alpha_1 + \beta_1 + \gamma_1 = \pi \end{cases} \quad (3.14)$$

$$\begin{cases} \tan(\gamma_2) = \frac{H}{L/2-\Delta X} \\ \tan(\beta_2) = \frac{H}{L/2+\Delta X} \\ \alpha_2 + \beta_2 + \gamma_2 = \pi \end{cases} \quad (3.15)$$

Les inconnues sont les mêmes que dans le cas précédent avec en plus la variable ΔX , soit le décalage horizontal entre la position du drône et le centre de la rivière. Celle-ci peut être déterminée grâce au rapport $\frac{\Delta X}{L}$ exprimé en mètres et en pixels sur les deux images. Ce rapport est identique qu'il soit exprimé en mètres ou en pixels.

$$\frac{\Delta X}{L} = \frac{\Delta X_{pix,1}}{L_{pix,1}} = \frac{\Delta X_{pix,2}}{L_{pix,2}} \quad (3.16)$$

Les indices 1 et 2 désignent toujours respectivement les données relatives à la photo prise au point le plus éloigné de la surface de l'eau et au point le plus proche de l'eau.

Le décalage en pixels ΔX_{pix} peut être calculé comme la distance horizontale entre le centre de l'image et le centre de la rivière. La largeur de la rivière en pixel peut, quant à elle, être calculée à partir de la distance entre les deux points posés par l'utilisateur. On constate dès lors que le décalage relatif en pixel $\frac{\Delta X_{pix}}{L_{pix}}$ peut être déduit pour chacune des deux images. Comme ce décalage relatif est identique en pixels et en mètres, le décalage ΔX peut être déduit.

En introduisant alors cette donnée dans les systèmes d'équations 3.14 et 3.15, la largeur d'eau réelle peut être déduite. Le niveau est alors calculé de la même façon que dans la méthode image expliquée à la section précédente.

3 Analyse PIV

3.1 Définition

Comme présenté au début au point 2.1 (p. 7), l'analyse PIV (Particule Image Velocimetry) est une technique utilisée dans l'application qui permet de déterminer le champ de vitesses à la surface du plan d'eau étudié. Cette technique, qui ne nécessite aucun contact avec la surface de l'eau, se base sur une séquence d'images analysées par paires. Entre deux images, les déplacements des particules sont déterminés. Connaissant l'intervalle de temps entre elles, les vitesses peuvent ainsi être calculées (équation 3.17).

$$\vec{V} = \frac{\Delta \vec{X}}{\Delta t} \quad (3.17)$$

Particules

Le principe de l'analyse PIV est basé sur le suivi de groupes de particules. Si le débit de la rivière est élevé et génère des remous, contrastant avec la surface lisse d'une eau calme, ces remous seront utilisés pour l'analyse. Dans le cas contraire, des particules appelées traceurs devront être semées dans le cours d'eau au moment de la prise de vidéo. De cette façon, le contraste de l'eau est augmenté artificiellement.

Il est démontré que la taille des particules a un impact sur les résultats obtenus [43]. Un compromis doit donc être trouvé. Elles doivent être suffisamment petites pour suivre le cours d'eau et suffisamment grandes pour être reconnaissables. Il est également recommandé d'avoir des particules de même taille [28]. Une discussion de l'influence des traceurs sur les résultats obtenus sera présentée à la section 2.3 du chapitre 9.

Pour intensifier le contraste des particules, plusieurs filtres présentés à la section 1.4 (p. 27) peuvent également être appliqués.

Détermination de la zone à analyser

L'analyse PIV est une opération dont le coût en temps est élevé. Pour le réduire, une solution très simple est de diminuer la taille de la zone d'analyse. Au lieu de la réaliser sur l'ensemble de l'image, une région d'intérêt, appelée ROI dans l'application, est définie. Seuls les déplacements des particules de cette zone seront alors calculés.

Delta t

Pour calculer les vitesses (en $\frac{m}{s}$), il est nécessaire de connaître l'intervalle de temps (en secondes) entre les paires d'images. Celui-ci est déduit à partir du nombre d'images par seconde de la vidéo, exprimé en *frame per second*.

$$dt = \frac{1}{\frac{frame}{s}} = [s] \quad (3.18)$$

3.2 OpenPIV

L'analyse PIV est le coeur de l'application. Elle permet de déterminer le déplacement Δx d'un groupe de particules entre deux images consécutives et connaissant l'intervalle de temps Δt , la vitesse instantanée peut être calculée. Pour ce faire, le package OpenPIV de python est utilisé. Il est important de préciser que ce package est encore en version *Beta*, c'est-à-dire qu'il est toujours en cours de développement et de correction. Les changements majeurs comme, par exemple, un changement de type d'arguments doivent être suivis pour éviter tout problème ultérieur.

Aire d'interrogation et Aire de recherche

Pour comprendre comment le déplacement d'un groupe de particules est déterminé entre chaque paire d'images, il faut définir les concepts de **grille d'interrogation**, d'**Aire d'interrogation (AI)** et d'**Aire de recherche (AR)**.

- Lors de l'analyse de chaque paire d'images, une grille carrée est virtuellement créée sur l'image. Elle est appelée **grille d'interrogation** et est représentée en traits discontinus verts à la figure 3.27. Les déplacements, et donc plus tard les vitesses, sont évalués pour chaque noeud formé aux intersections de cette grille. La taille des côtés des carrés, appelée *overlap* dans l'application, est exprimée en pixels et est généralement fixée à 50% de la longueur d'un côté de l'aire d'interrogation.
- L'**aire d'interrogation (AI)** est une surface carrée de côté M, centrée sur le noeud de la grille d'interrogation et représentée en trait continu rouge aux figures 3.27 et 3.28. Elle représente la surface d'intérêt dont la meilleure correspondance va être recherchée dans l'image B. C'est donc bien le déplacement du groupe de particules présentes dans cette surface qui est déterminé.

- Finalement, l'**aire de recherche (AR)** est une surface carrée de côté N , centrée sur le noeud de la grille d'interrogation et représentée en trait discontinu rouge sur la figure 3.28. Cette surface de l'image représente la zone dans laquelle la meilleure correspondance à l'AI va être recherchée dans l'image B. L'AR permet de réduire le temps de calcul. En effet, comme l'intervalle de temps Δt entre les deux images est très faible, les particules dans l'AI de l'image A devraient être dans une zone relativement proche de l'image B. On s'attend donc à un déplacement inférieur à $(N+M)/2$. Par conséquent, la taille de l'AI et de l'AR devra varier en fonction du cours d'eau étudié et du Δt entre les images [28].

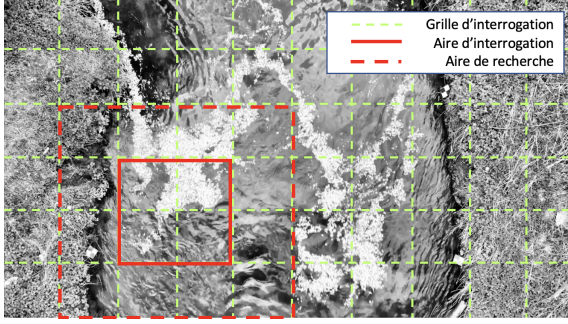


FIGURE 3.27 – Image A

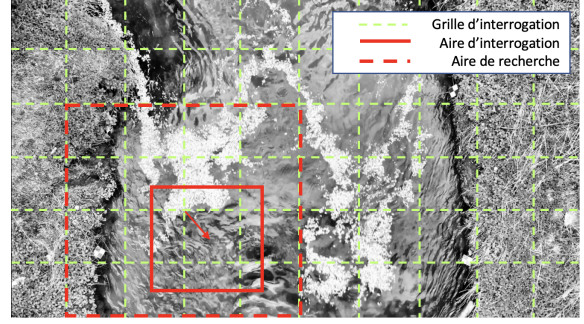


FIGURE 3.28 – Image B

Corrélation croisée

Maintenant que les bases ont été définies, il reste à présenter la méthode implémentée pour trouver la nouvelle position dans l'image B de l'AI initiale de l'image A. C'est la méthode de la corrélation croisée qui est utilisée dans l'application et que nous allons détailler ci dessous. Pour chaque noeud de la grille d'interrogation, une partie de l'image A, correspondant à l'aire d'interrogation centrée en ce noeud, est isolée et nommée I . Une partie de l'image B, correspondant à l'aire de recherche centrée en ce même noeud, est également isolée et nommée I' . La matrice de corrélation R entre ces deux aires isolées est ensuite calculée en utilisant l'équation 3.20 [44].

$$R_s = (I \otimes I')(s) = \sum_i I(x'_i - s)I'(x'_i) \quad (3.19)$$

$$\Leftrightarrow R_{k,l} = \frac{\sum_{i=1}^M \sum_{j=1}^M [(I_{i,j} - \bar{I})(I'_{i,j} - \bar{I}')]}{\left[\sum_{i=1}^M \sum_{j=1}^M (I_{i,j} - \bar{I})^2 \cdot \sum_{i=1}^M \sum_{j=1}^M (I'_{i,j} - \bar{I}')^2 \right]^{\frac{1}{2}}} \quad (3.20)$$

De façon plus intuitive, chaque entrée de la matrice de corrélation correspond au coefficient de corrélation calculé entre l'aire de recherche I' et l'aire d'interrogation I déplacée de $s = (k, l)$ comme représenté à la figure 3.29. Ce coefficient peut être interprété comme un indicateur de ressemblance des deux parties d'image isolées. Plus il est élevé, plus les particules de l'AI coïncident avec celles de l'AR. Après avoir complété la matrice de corrélation en testant tous les déplacements s possibles, la position du maximum de corrélation, appelé pic de corrélation, permet d'estimer le déplacement de l'image A vers l'image B. Ce pic est souvent représenté dans la littérature à l'aide d'une surface 3D comme illustrée à la figure 3.30.

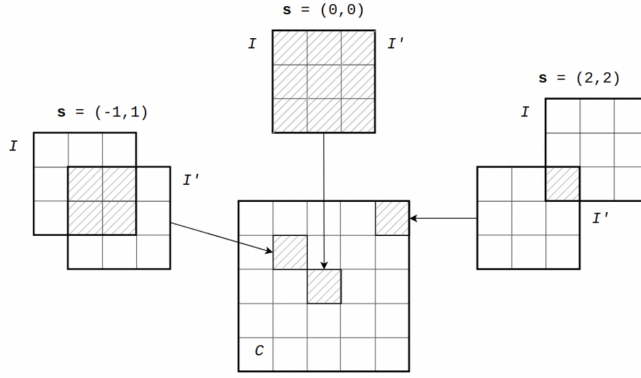


FIGURE 3.29 – Application de la méthode de corrélation croisée [44]

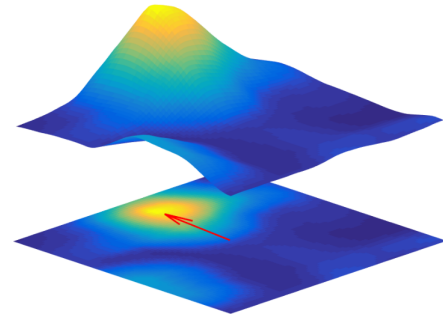


FIGURE 3.30 – Représentation du pic de corrélation [45]

Cette méthode permet donc de déterminer le déplacement d'un noeud de la grille d'interrogation. On répétera cette opération pour chacun des noeuds afin de couvrir toute l'image. Finalement, on divise les déplacements calculés par l'intervalle de temps Δt qui sépare les deux images. Nous obtenons donc une répartition spatiale de vitesses exprimées en pixels par seconde.

Dans l'application, la fonction du package OpenPIV `pyprocess.extended_search_area_piv()` calcule les vecteurs vitesses entre deux images. Elle prend en argument, entre autres, la taille de l'aire d'interrogation (M), appelée `window_size`, et la taille de l'aire de recherche (N), appelée `search_area`. Alors que le choix de la taille de l'aire d'interrogation est laissé à l'utilisateur, celle de l'aire de recherche est fixée et est égale à celle de l'aire d'interrogation. On a donc $M = N$ et la taille des matrices de corrélation sera $2M \times 2M$.

Fast Fourier Transform

Le principal point faible de la corrélation croisée est son temps de calcul. En effet, la matrice de corrélation étant de la taille $(2M)^2$, il y a autant de coefficients de corrélation à calculer, et ce pour chaque noeud de la grille d'interrogation. La taille de cette grille varie quant à elle en fonction de la taille du ROI choisi et de l'*overlap* choisi par l'utilisateur. Le temps de calcul de cette étape oscille entre $\mathcal{O}(M^2)$ et $\mathcal{O}(M^4)$ [28]. Pour rappel, tout ceci permet d'obtenir les déplacements entre deux images et il ne faut pas perdre de vue que cet algorithme est répété autant de fois qu'il y a de paire d'images à analyser.

Pour remédier à ce problème, il est possible d'accélérer l'étape de la corrélation croisée en l'appliquant dans le domaine fréquentiel, grâce à l'utilisation de la transformée de Fourier. En effet, le théorème de corrélation croisée, qui s'énonce sous la forme de l'équation 3.21 [44], montre que le calcul de corrélation 2D est de l'ordre de $\mathcal{O}(2M^2 \log_2 M)$ [46].

$$A \otimes B = \mathcal{F}^{-1}\{\mathcal{A}^* \cdot \mathcal{B}\} \quad (3.21)$$

C'est cette amélioration du calcul de la matrice de corrélation en utilisant la transformée de Fourier qui est implémentée dans la fonction `pyprocess.extended_search_area_piv()`, native d'OpenPIV. Une synthèse de la méthode de la corrélation croisée utilisant la transformée de Fourier rapide est représentée à la figure 3.31.

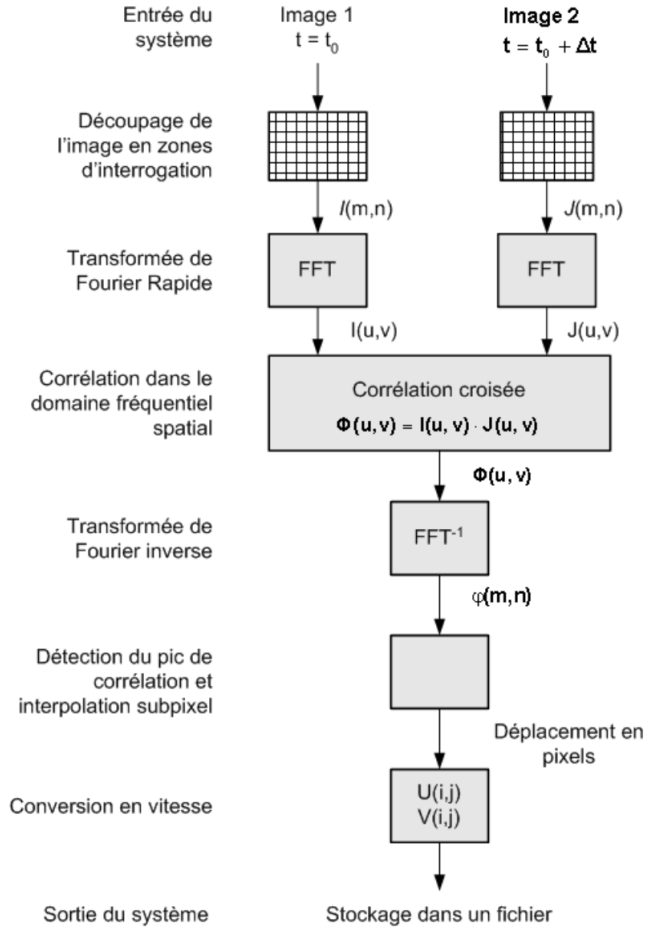


FIGURE 3.31 – Synthèse de la méthode de la corrélation croisée [47]

Post-processing

Après avoir effectué l'analyse PIV, le travail n'est pas tout à fait fini. L'étape de post-processing permet de supprimer ou remplacer les valeurs extrêmes calculées. Elle est composée de la validation des données suivie du remplacement des données extrêmes. Les calculs des vitesses moyennes et du débit en font également partie.

Validation

Le champ de vitesses obtenu à la fin de l'analyse PIV peut contenir des vecteurs vitesses erronés. Afin d'exclure ces vecteurs, une phase de validation des résultats est nécessaire.

La librairie OpenPIV fournit cinq fonctions permettant d'appliquer différentes méthodes de validation. Ces méthodes utilisent des critères différents pour considérer un vecteur comme erroné [48].

1. La validation utilisée dans l'application se base sur la matrice de corrélation. Elle utilise la fonction `validation.sig2noise_val()` qui prend en argument les vecteurs vitesses u et v ainsi qu'une valeur seuil dont la valeur est choisie par l'utilisateur. La validation se base sur les ratios de corrélation croisée issus de l'analyse d'images par paires. Si le ratio relatif à une vitesse est inférieur au seuil indiqué par l'utilisateur, ce vecteur vitesse est considéré comme une valeur extrême.
2. La validation géométrique peut être utilisée via la fonction `global_val()`. Cette fonction prend en arguments les composantes u et v des vecteurs vitesse calculés ainsi qu'une valeur

seuil pour chaque composante. Les valeurs seuils se présentent sous la forme (u_{min}, u_{max}) et (v_{min}, v_{max}) . La composante est considérée comme invalide si elle n'est pas comprise dans l'intervalle défini par la valeur seuil. Si une des deux composantes est invalide, le vecteur est considéré comme erroné.

3. La fonction `global_std()` réalise une validation géométrique des vitesses sur base d'une valeur seuil et de l'écart type. La fonction prend en arguments les composantes u et v des vecteurs vitesse ainsi qu'une valeur seuil dont la valeur par défaut vaut 3. Si la longueur d'au moins une composante du vecteur est plus grande que la valeur seuil multipliée par l'écart type de cette composante, le vecteur est considéré comme erroné.
4. Une autre validation géométrique basée, cette fois, sur la valeur médiane est réalisable via la fonction `local_median_val()`. Cette fonction prend en arguments les composantes u et v des vecteurs vitesses ainsi qu'une valeur seuil pour chaque composante. Un vecteur est considéré comme erroné si l'écart entre une de ses composantes et la médiane des composantes des vecteurs vitesse est supérieur au seuil indiqué en argument.
5. Une validation plus globale peut être effectuée via la fonction `typical_value()`. Elle prend en arguments les composantes u et v des vitesses ainsi que le tableau relatif à la corrélation issu de l'analyse d'images par paires et certains paramètres appelés settings déjà définis par OpenPIV. Cette méthode applique les fonctions de validation `global_val()`, `global_std()` et `local_median_val()` expliquée précédemment.

Quand le vecteur vitesse est considéré comme erroné, il est remplacé par la valeur `NaN`, quelle que soit la fonction de validation utilisée. En sortie, ces fonctions renvoient toujours les vecteurs vitesse u et v corrigés où les valeurs erronées sont remplacées par la valeur `NaN` ainsi qu'un masque. Celui-ci est un tableau de valeurs booléennes dont les valeurs égales à `True` correspondent aux valeurs remplacées.

Remplacement des données extrêmes

Pour compléter l'étape de validation, la fonction `filters.replace_outliers()` remplace toutes les valeurs `NaN` par une valeur moyenne. Cette valeur est calculée localement sur base des vecteurs vitesses avoisinants celui à remplacer. La figure 3.32 illustre le principe du remplacement des données extrêmes. Les zones vides sur la première image représentent les valeurs `NaN` issues de la validation. Une analyse détaillée de son fonctionnement a été réalisée durant la correction de l'application, présentée à la section 2.2 au chapitre 7.

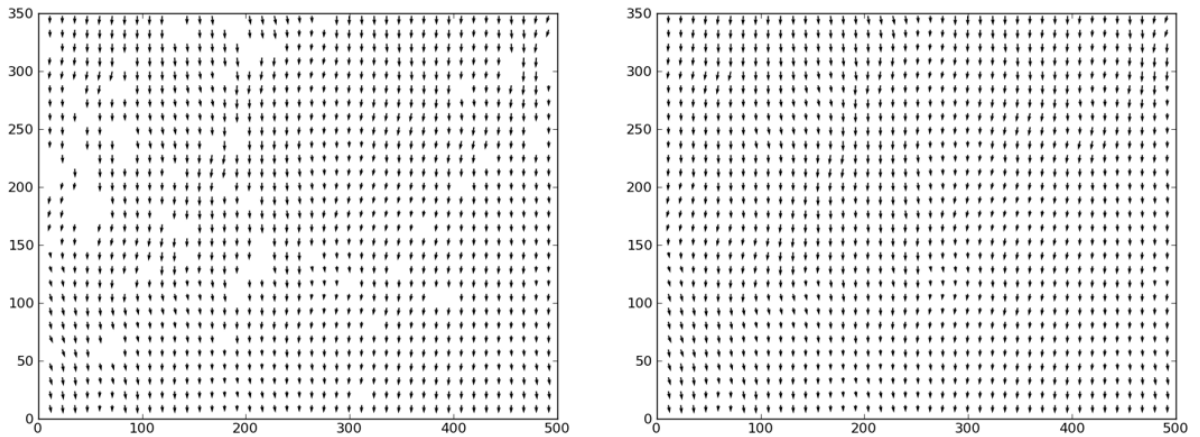


FIGURE 3.32 – Vecteurs vitesses avant et après remplacement des valeurs aberrantes [49]

Calcul des vecteurs vitesses moyens et calcul du débit

En fonction des paramètres choisis par l'utilisateur, le nombre de frame par seconde de la vidéo à analyser est déterminé et les vitesses sont calculées entre chaque doublon d'image successive. A la fin de l'analyse, nous obtenons pour chaque noeud de la grille d'interrogation un nombre de vecteurs vitesse équivalent au nombre de doublon d'images successives. Ainsi, si une vidéo est analysée sur un intervalle de temps d'une seconde et possède 30 frames sur cette seconde, on obtiendra 29 vecteurs vitesse pour chaque noeud.

La fonction `computeMeanVelocity()` a donc été créée et calcule simplement la moyenne de tous les vecteurs vitesse d'un même noeud. On obtient donc le champ de vitesses moyen à la surface de l'eau qui sera ensuite utilisé pour calculer le débit.

4 Calcul du débit

Plusieurs étapes sont encore nécessaires pour calculer le débit :

- La section sur laquelle est calculée le débit est choisie par l'utilisateur. Sur cette section, les vecteurs moyens de vitesse les plus proches sont isolés et le changement d'unité de $[\frac{px}{s}]$ en $[\frac{m}{s}]$ est effectué en appliquant le facteur d'échelle. De plus, si une orthorectification a été nécessaire, les vecteurs vitesses isolés seront corrigés à l'aide de la matrice d'homographie présentée à la section 1.3 de ce chapitre. Les vecteurs vitesses orthorectifiés et exprimés en $[\frac{m}{s}]$ sont alors affichés sur la section dans l'image qui apparaît. Si la section choisie n'est pas parfaitement horizontale, une rotation de l'image sera effectuée afin que cette condition soit respectée et que seules les composantes de vitesse verticales soient prises en compte pour le calcul de débit.
- Cette section d'intérêt est subdivisée en n sous-sections sur lesquelles le calcul du débit sera effectué, connaissant les vitesses de surface et en appliquant un coefficient liant la vitesse de surface au profil de vitesse moyenne de l'écoulement.

Facteur d'échelle

Les vecteurs vitesses issus de l'analyse PIV et isolé sur la section choisie sont exprimés en $[\frac{px}{s}]$. Elles sont alors converties en $[\frac{m}{s}]$ grâce au facteur d'échelle. Ce facteur d'échelle est calculé différemment selon qu'une orthorectification a été réalisée ou non. Dans le premier cas, la section 1.3 décrit comment il est déterminé.

Si aucune orthorectification n'a été nécessaire, le *scaling factor* se calcule sur base de la section d'intérêt sur laquelle le débit est calculé. Concrètement, la hauteur d'eau et donc la largeur d'eau est connue et est exprimée en mètre. La longueur de la section d'intérêt étant exprimée en pixels, le *scaling factor* est simplement exprimé comme le rapport de la largeur d'eau sur la longueur de la section d'intérêt et ses unités s'expriment toujours en $[\frac{px}{m}]$.

Lien entre vitesse de surface et vitesse moyenne sur la profondeur

Comme expliqué dans la section 3.2 (p. 40), les vecteurs vitesses moyens issus de l'analyse PIV représentent les vitesses de surface de l'écoulement. Cependant, pour calculer le débit d'une section, il est nécessaire de connaître la composante de vitesse normale à la section choisie. Les vitesses de surface doivent donc être converties en vitesses moyennes sur la profondeur. Pour cela, le coefficient de vitesse α qui lie ces deux vitesses sera utilisé (voir figure 3.33 et équation 3.22).

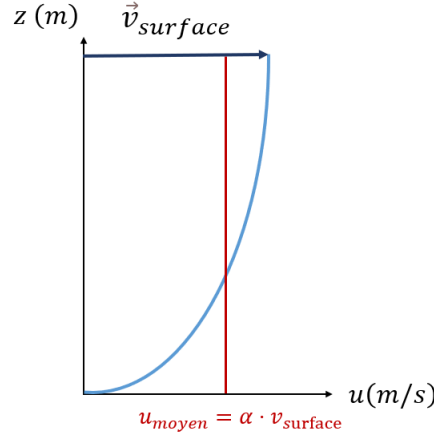


FIGURE 3.33 – Illustration du coefficient alpha

$$\alpha = \frac{\text{Vitesse moyenne sur la profondeur}}{\text{Vitesse de surface}} \quad (3.22)$$

Dans l'application, le coefficient est supposé constant quelle que soit la sous section traitée.

La valeur du coefficient α peut être déterminée expérimentalement ou théoriquement. Cependant, ce coefficient est encore une inconnue. Son importance est cruciale pour les techniques non intrusives puisque leur enjeu est de déterminer le débit à partir d'informations sur la surface de l'eau. Différents travaux s'intéressent plus en détail à ce coefficient afin de déterminer quels sont les paramètres qui l'influencent.

Dans une recherche menée en 2000 par Costa *et al.*, les vitesses de surface ont été mesurées en utilisant un radar doppler à micro-ondes sur une section de rivière divisée en 25 sous sections. Dans l'expérience, le coefficient liant la vitesse de surface à la vitesse moyenne sur la profondeur a été associé à une valeur de 0,85. Cette valeur donne des résultats précis prouvant, par la même occasion que les techniques sans contact utilisées dans des conditions idéales donnent des résultats aussi précis que les techniques intrusives conventionnelles [50].

En 2005, Polatel et ses confrères ont réalisé une série d'expériences démontrant l'influence de la rugosité du lit sur le coefficient α . Dans ces expériences, les vitesses de surface ont été mesurées au moyen de la technique LSPIV et les vitesses moyennes sur la profondeur ont été mesurées au moyen de la méthode Laser Doppler Velocimetry (LDV). Le coefficient α , supposé ici constant sur toute la section de la rivière, a pu être déduit expérimentalement. La valeur du coefficient α peut également être déduite mathématiquement si la fonction de distribution des vitesses est connue [51]. Les expériences menées ont démontré l'influence non négligeable de la rugosité du lit ainsi que de la hauteur d'eau sur le coefficient α (voir figure 3.34). Celui-ci semble augmenter lorsque la hauteur d'eau est plus élevée.

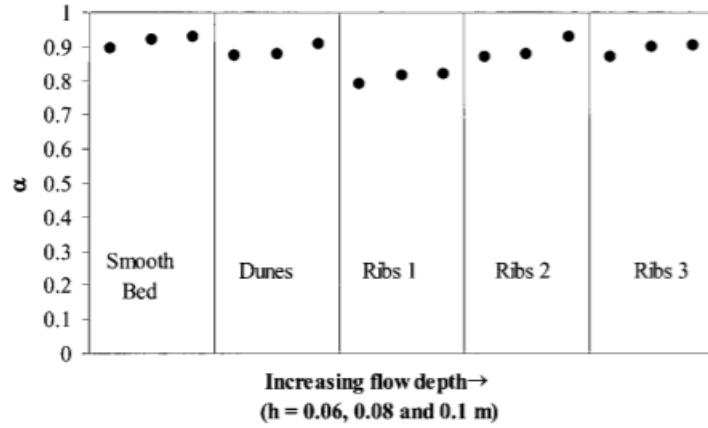


FIGURE 3.34 – Variation de l'index de vitesse α pour trois valeurs de hauteur d'eau dans cinq milieux différents. Malgré le changement de milieu, l'index de vitesse reste compris entre 0,789 et 0,928 [51]

En 2006, Hauet, durant son travail, fait plusieurs constatations concernant le coefficient alpha. Il exprime ce coefficient selon trois manières différentes : la loi logarithmique (voir équation 3.23), la loi puissance (voir équation 3.24) et les valeurs expérimentales.

$$\alpha = \frac{D_i}{D_i - z_0} - \ln \left(\frac{D_i}{z_0} \right) \quad (3.23)$$

où D_i représente la profondeur d'eau et z_0 la rugosité.

$$\alpha = \frac{m}{m + 1} \quad (3.24)$$

avec $m = L_0 - 1$ et $L_0 = \ln \left(\frac{D_i}{z_0} \right)$

Les lois théoriques logarithmique et de puissance sont ici utilisées car elles correspondent aux profils de vitesse verticaux observés en laboratoire.

Hauet compare ensuite les résultats obtenus sur un large échantillon couvrant aussi bien des cours d'eau de faibles débits que ceux de débits importants [2]. A partir de cette comparaison, il constate que les valeurs moyennes du coefficient obtenues par les trois méthodes sont proches. De plus, il remarque que le coefficient α augmente avec le rayon hydraulique R_h de la section et est impacté par la rugosité relative $\frac{R_h}{d}$. Le coefficient est également influencé par le vent qui augmente les vitesses de surface. Enfin, la distribution des vitesses dépend de la rugosité du lit. Or cette donnée est relativement difficile à mesurer. Si le profil correspond à une loi logarithmique, la valeur de 0,85 peut être utilisée. Hauet recommande finalement d'utiliser une valeur de 0,9 pour un canal lisse en béton et une valeur de 0,8 pour une rivière naturelle à faible profondeur³.

En 2016, une recherche effectuée par Welber *et al.* utilise la technique Surface Velocity radar (SVR) sur un large échantillon de rivières. Les profils de vitesse et les débits obtenus par radar sont comparés avec ceux obtenus avec les techniques intrusives (ici, ADCP⁴).

Le coefficient de vitesse α a également été exprimé selon la relation logarithmique (voir équation 3.23). La valeur théorique et la valeur expérimentale ont été comparées. Lorsque le coefficient est supposé uniforme sur toute la section, quelle que soit la sous section, la valeur du coefficient est stable et proche de la valeur théorique.

Selon Welber, la valeur de 0,85 est valable pour des valeurs de rugosité assez variables. Cependant, lorsqu'elle est utilisée, il y a un risque d'obtenir des débits trop faibles pour des

3. Une rivière à faible profondeur désigne ici une rivière dont la hauteur d'eau est inférieure à deux mètres.

4. Acoustic Doppler Current Profiler. Pour plus d'informations, la section 1.1 du chapitre 1 sur les mesures intrusives peut être consultée.

canaux lisses et un risque de surestimation pour des rivières naturelles. Il est donc préférable de calibrer la valeur du coefficient selon le site étudié. A propos de la technique SVR, l'article conclut qu'elle permet d'obtenir les débits avec une précision de 10 % si la rugosité du site est intermédiaire [52].

En résumé, le coefficient α dépend non seulement de paramètres géométriques et hydrauliques tels que le rayon hydraulique et le niveau d'eau mais aussi de la présence de végétation, de la rugosité, du vent ou encore de la présence d'un affluent. Dans l'application, ce coefficient dépend uniquement de la profondeur de la rivière et est implémenté de la manière suivante selon l'article de Hauet [2] :

$$\alpha = \begin{cases} 0.8 & \text{si } Water\ Level \leq 2m \\ 0.9 & \text{sinon} \end{cases} \quad (3.25)$$

Cette valeur calculée peut également être modifiée par l'utilisateur. Sans indication de valeur pour le coefficient de la rivière analysée, la valeur générale de 0,85 est conseillée et permet d'obtenir un débit avec une erreur relativement faible.

Calcul du débit

Le débit total sur la section Q_{tot} est alors calculé comme la somme des débits Q_i traversant chaque sous section.

$$Q_{tot} = \sum_{i=1}^n Q_i = \sum_{i=1}^n V_i \cdot A_i \left[\frac{m^3}{s} \right] \quad (3.26)$$

avec A_i et V_i respectivement l'aire mouillée et la vitesse de chaque sous section et n le nombre de sous sections.

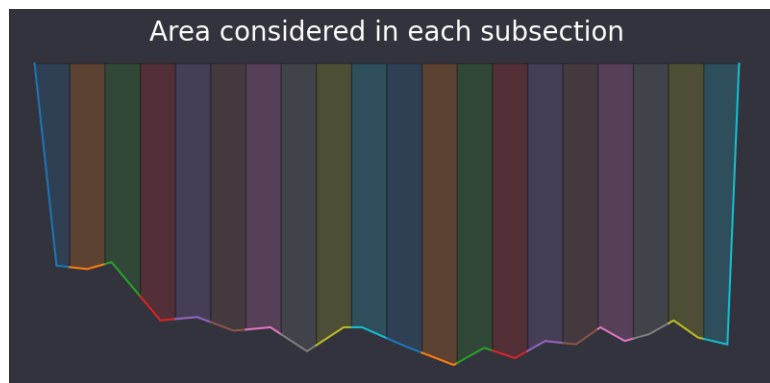


FIGURE 3.35 – Subdivision de l'aire mouillée pour le calcul du débit

Sur chaque sous-section, il faut connaître la vitesse et l'aire. La vitesse a été déduite à l'étape du choix du nombre de sous sections. Les seules inconnues qui restent pour calculer le débit sont donc les aires mouillées des sous sections. Celles-ci sont calculées au moyen de la fonction `computeAreaSection()`. Cette fonction utilise un algorithme d'intersection pour déterminer la courbe complète de bathymétrie puis fait appel à la règle trapézoïdale pour déduire l'aire.

$$A_i = \frac{H_i + H_{i+1}}{2} \cdot L_i \left[m^2 \right] \quad (3.27)$$

avec L_i la largeur de chaque sous section.

Deuxième partie

Analyse de la première version de l'application

Chapitre 4

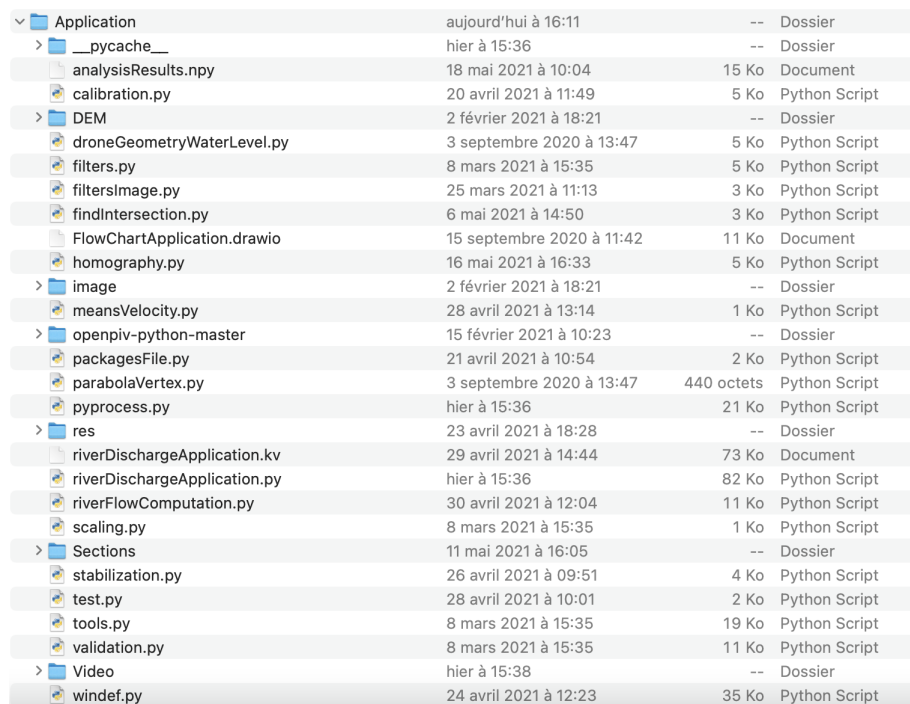
Structure de l'application

La structure générale présentée ci-dessous est la structure de l'application telle que nous l'avons reçue. Durant notre travail, bien que nous ayons corrigé certaines étapes et apporté de nouvelles options, nous avons gardé cette structure. La structure générale de l'application présentée dans ce chapitre reste valide après les modifications réalisées lors de notre travail.

1 Structure générale

1.1 Dossiers internes

L'application est écrite en python et exécutée à l'aide de Spyder. Tous les fichiers nécessaires à son fonctionnement se trouvent dans le dossier *Application* dont la structure est affichée à la figure 4.1. On y retrouve les fichiers *.py* qui constituent les différentes parties de l'application ainsi que les dossiers *image*, *Sections* et *Video* où l'on retrouve respectivement les images utilisées dans l'application, les bathymétries enregistrées et les vidéos des différents cours d'eau. La construction de l'application impose de respecter cette structure si l'on souhaite l'utiliser correctement.



Application	aujourd'hui à 16:11	--	Dossier
__pycache__	hier à 15:36	--	Dossier
analysisResults.npy	18 mai 2021 à 10:04	15 Ko	Document
calibration.py	20 avril 2021 à 11:49	5 Ko	Python Script
DEM	2 février 2021 à 18:21	--	Dossier
droneGeometryWaterLevel.py	3 septembre 2020 à 13:47	5 Ko	Python Script
filters.py	8 mars 2021 à 15:35	5 Ko	Python Script
filtersImage.py	25 mars 2021 à 11:13	3 Ko	Python Script
findIntersection.py	6 mai 2021 à 14:50	3 Ko	Python Script
FlowChartApplication.drawio	15 septembre 2020 à 11:42	11 Ko	Document
homography.py	16 mai 2021 à 16:33	5 Ko	Python Script
image	2 février 2021 à 18:21	--	Dossier
meansVelocity.py	28 avril 2021 à 13:14	1 Ko	Python Script
openpiv-python-master	15 février 2021 à 10:23	--	Dossier
packagesFile.py	21 avril 2021 à 10:54	2 Ko	Python Script
parabolaVertex.py	3 septembre 2020 à 13:47	440 octets	Python Script
pyprocess.py	hier à 15:36	21 Ko	Python Script
res	23 avril 2021 à 18:28	--	Dossier
riverDischargeApplication.kv	29 avril 2021 à 14:44	73 Ko	Document
riverDischargeApplication.py	hier à 15:36	82 Ko	Python Script
riverFlowComputation.py	30 avril 2021 à 12:04	11 Ko	Python Script
scaling.py	8 mars 2021 à 15:35	1 Ko	Python Script
Sections	11 mai 2021 à 16:05	--	Dossier
stabilization.py	26 avril 2021 à 09:51	4 Ko	Python Script
test.py	28 avril 2021 à 10:01	2 Ko	Python Script
tools.py	8 mars 2021 à 15:35	19 Ko	Python Script
validation.py	8 mars 2021 à 15:35	11 Ko	Python Script
Video	hier à 15:38	--	Dossier
windef.py	24 avril 2021 à 12:23	35 Ko	Python Script

FIGURE 4.1 – Structure des documents de l'application

1.2 Flowchart

La structure de l'application est synthétisée à l'aide du flowchart représenté à la figure 4.2. Nous y retrouvons les trois grandes parties : pré-processing, analyse PIV et post-processing. Il permet également de mettre en avant les étapes facultatives. Les sections suivantes présentent en détails chacune de ces étapes.

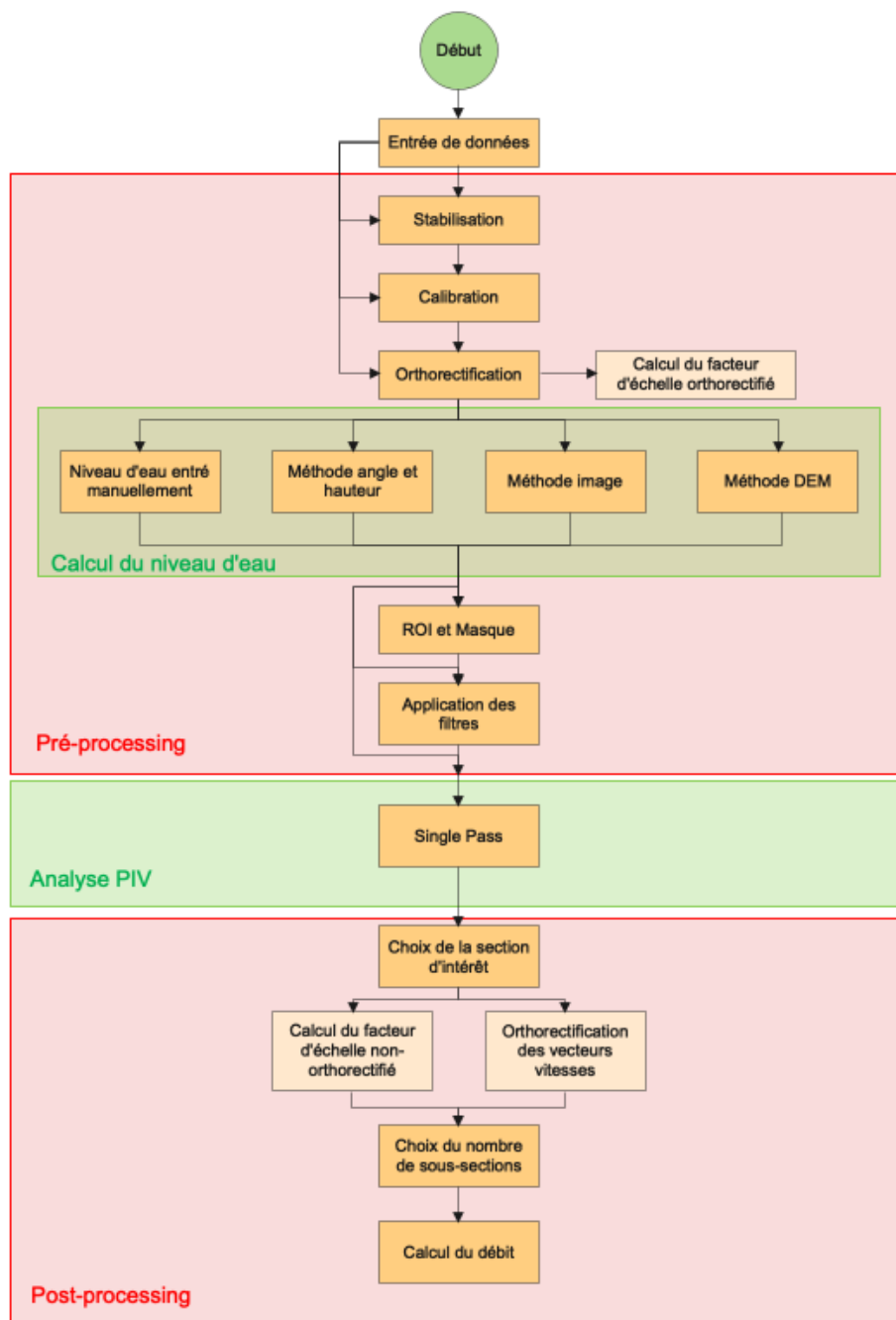


FIGURE 4.2 – Flowchart initial

2 Chargement de la bathymétrie

Cette application fait l'hypothèse que la bathymétrie est connue de l'utilisateur. Celle-ci doit donc être introduite dans l'application. Cela peut être fait de deux manières différentes (voir figure 4.3).

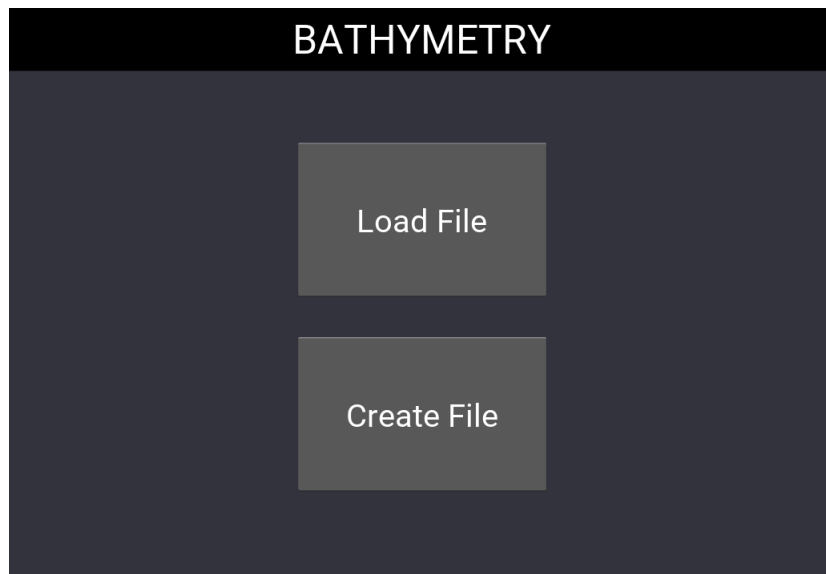


FIGURE 4.3 – Choix de la méthode pour le chargement de la bathymétrie

1. **Load File** : l'utilisateur peut charger un fichier `.txt` existant contenant les coordonnées des points qui forment la bathymétrie (voir figure 4.4). L'abscisse et l'ordonnée de chaque point doivent être séparées par une virgule et sont exprimées en mètres. Une fois que l'utilisateur a chargé le fichier, la bathymétrie définie par les points du fichier s'affiche dans une nouvelle fenêtre et peut être chargée (voir figure 4.5).

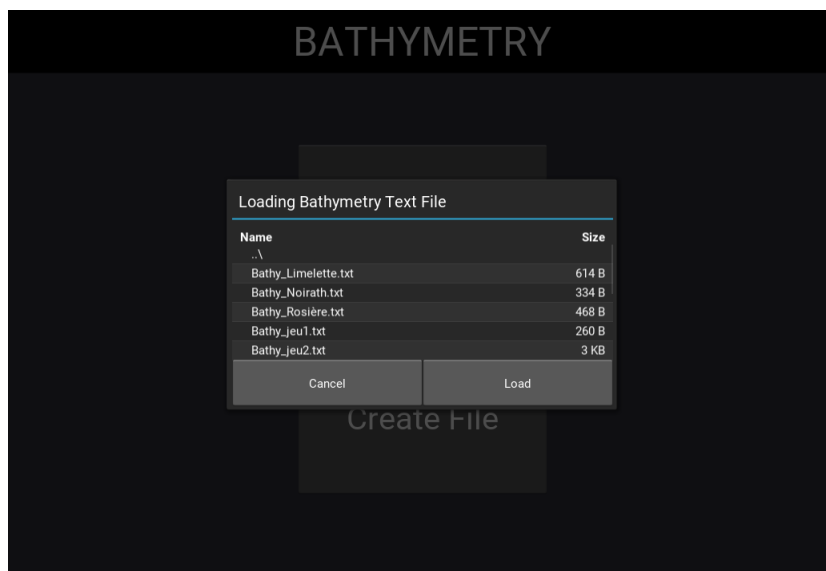


FIGURE 4.4 – Choix du fichier contenant la bathymétrie

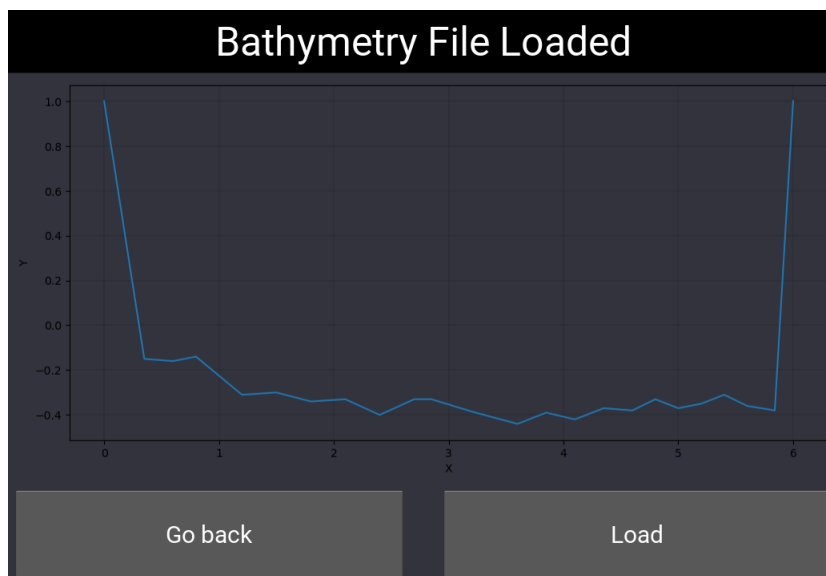


FIGURE 4.5 – Affichage de la bathymétrie choisie

2. **Create File** : l'utilisateur peut créer lui-même un fichier `.txt` dans lequel les données de la bathymétrie seront encodées. Pour cela, il doit sélectionner une forme de section correspondant au cours d'eau à analyser : rectangulaire, trapézoïdale, parabolique ou libre. Pour chaque forme, il est nécessaire de fournir le nom du fichier, les coordonnées d'un point de référence (en rouge sur la figure 4.6) et les dimensions caractéristiques de la section (e.g. pour un rectangle, largeur et hauteur). Pour la section libre, l'utilisateur doit fournir une série de coordonnées de points qui définissent celle-ci. Ces données sont ensuite enregistrées dans un fichier placé dans le dossier "Sections" de l'application. Ce fichier pourra être réutilisé sans devoir être recréé.

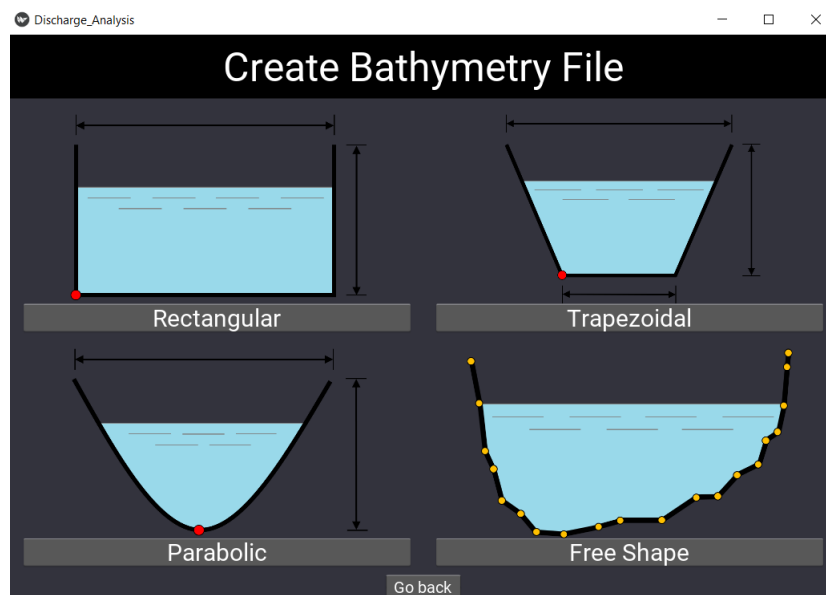


FIGURE 4.6 – Fenêtre permettant à l'utilisateur de choisir le profil de bathymétrie représentatif

Lors de cette étape, le fichier de bathymétrie est enregistré comme variable globale de l'application afin de pouvoir être réutilisé plus tard dans le processus de détermination du débit.

3 Chargement de la vidéo

Cette étape permet de charger la vidéo de l'écoulement (voir figure 4.7). A nouveau, il faut veiller à bien placer cette vidéo dans le dossier "Video" de l'application.

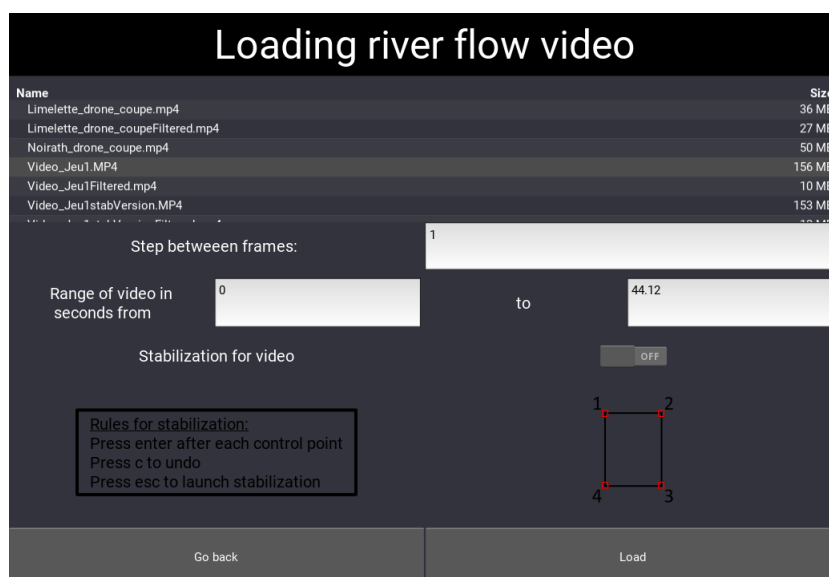


FIGURE 4.7 – Fenêtre permettant à l'utilisateur de choisir une vidéo

L'utilisateur a la possibilité de choisir un **step**. Celui-ci permet d'accélérer le processus de détermination du débit en analysant seulement les images espacées de ce step. Attention, l'utilisateur doit avoir placé le fichier au préalable dans le dossier "Sections" de l'application. Par défaut, le step a une valeur de "1" ce qui signifie que toutes les images de la vidéo sont analysées.

L'utilisateur peut également choisir un intervalle de la vidéo pour l'analyse, appelé **range**. Cela permet de ne pas analyser la vidéo entière. Par exemple, dans un cas où le cours d'eau est ensemencé par des particules qui ne sont pas visibles sur toute la durée de la vidéo, il peut être intéressant de ne garder que le passage où elles sont visibles pour l'analyse PIV. Sélectionner une partie de la vidéo permet également d'accélérer cette analyse. Par défaut, le range initial de la vidéo est la vidéo entière soit de 0 à la durée totale de la vidéo.

L'utilisateur a aussi la possibilité de choisir s'il souhaite ou non réaliser une stabilisation de la vidéo et finalement de charger les informations qu'il a sélectionnées. Si la stabilisation a été cochée, la fenêtre de stabilisation s'ouvre (voir section 4 du chapitre 3) et si non, l'application passe simplement à l'étape de calibration.

Les variables globales enregistrées sont le chemin de chargement de la vidéo, le step et le range sous forme de vecteur.

4 Stabilisation

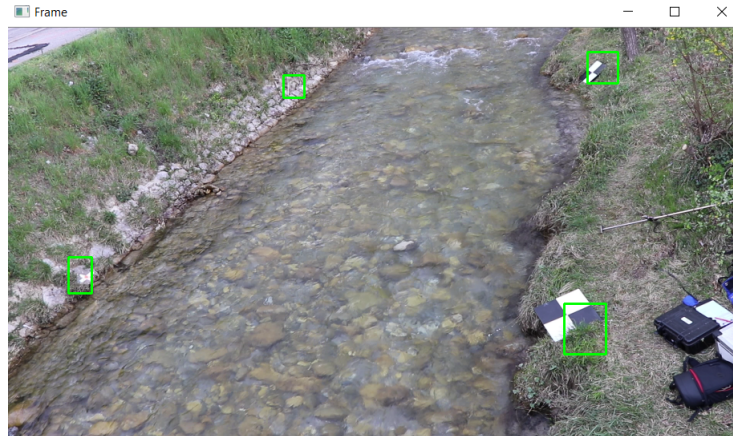


FIGURE 4.8 – Fenêtre qui apparaît lors de la stabilisation

La stabilisation permet de corriger les mouvements de la caméra. Si la vidéo a été filmée par une personne, il est impossible que la caméra soit restée parfaitement statique tout au long de la vidéo. De même lorsque la vidéo est filmée par un drone, des mouvements peuvent être induits. Cette étape permet donc de corriger ces mouvements non désirés afin qu'ils ne faussent pas les résultats.

Pour cela, il est demandé à l'utilisateur de sélectionner quatre points fixes sur la première image de la vidéo. Après chaque sélection, l'utilisateur doit presser la touche "enter" et à la fin la touche "escape" pour lancer la stabilisation.

La vidéo stabilisée sera alors enregistrée dans le dossier "Video" afin de pouvoir être réutilisée lors d'une prochaine analyse.

5 Calibration

Cette étape n'est pas obligatoire mais elle permet de corriger les distorsions occasionnées par la caméra et d'obtenir les paramètres intrinsèques et extrinsèques de la caméra (voir Sections 1.2 du chapitre 3).

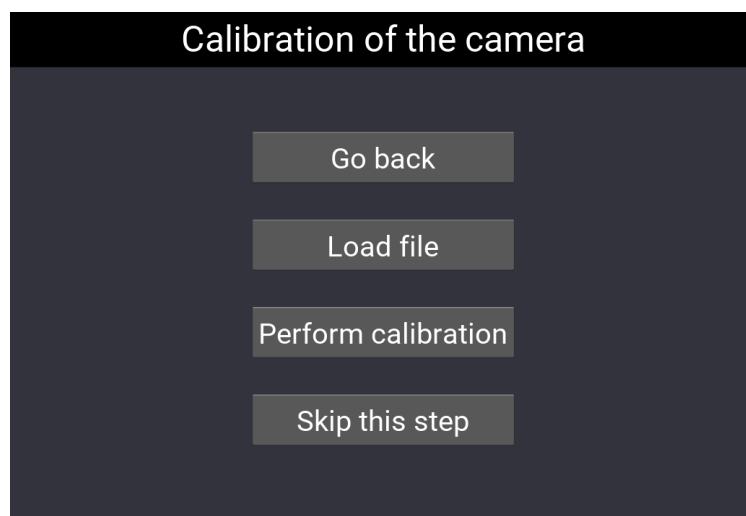


FIGURE 4.9 – Choix de la méthode de calibration

Pour cela, il existe deux méthodes (voir figure 4.9) :

1. **Load File** : Si une calibration de cette caméra a déjà été réalisée dans une analyse précédente, l'utilisateur doit simplement sélectionner le fichier `.npz` correspondant (voir figure 4.10). Celui-ci doit se trouver dans le dossier "Application".

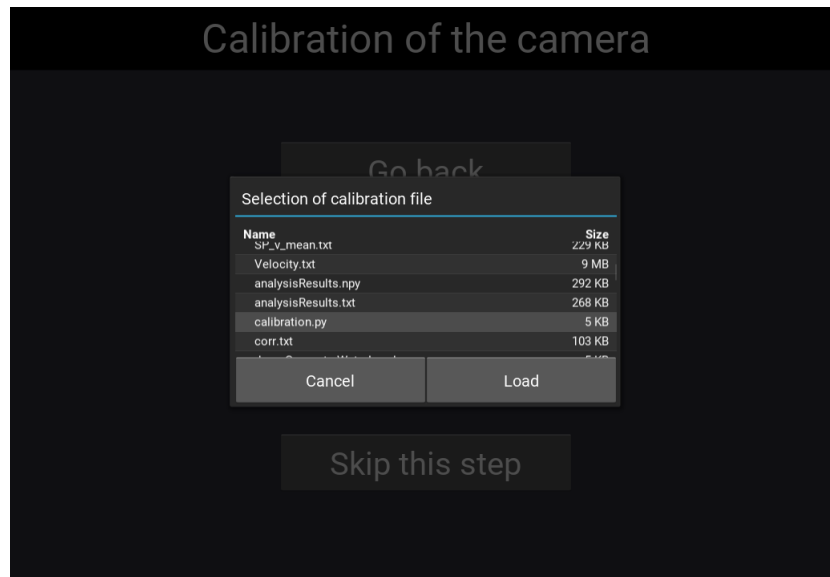


FIGURE 4.10 – Chargement du fichier contenant les paramètres de calibration

2. **Perform Calibration** : Une fenêtre apparaît avec un certain nombre de paramètres à fournir (voir figure 4.11). Tout d'abord, l'utilisateur doit fournir une vidéo d'un damier, prise selon plusieurs angles et vues. C'est cette vidéo qui permettra de réaliser la calibration. Ensuite, un nom de fichier, le nombre d'images extraites ainsi que la taille des carrés et les dimensions du damier doivent être encodés.

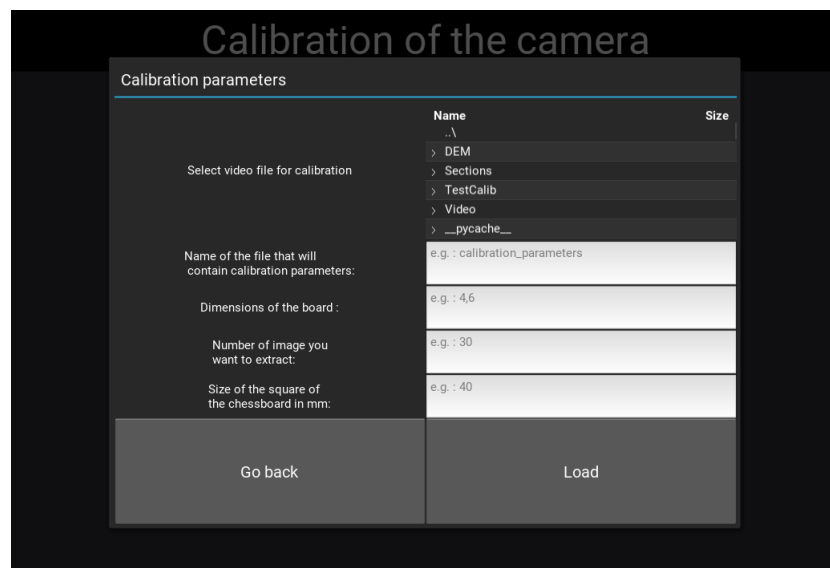


FIGURE 4.11 – Chargement de la vidéo et des caractéristiques du damier

A la fin de ces deux étapes, lorsque l'utilisateur choisit de charger, la vidéo est calibrée et enregistrée dans le dossier "Video" afin de pouvoir être réutilisée. De plus, un fichier `calibration.npz` contenant les paramètres de calibration est enregistré dans les variables globales de l'application.

Comme cette étape n'est pas obligatoire, l'utilisateur a l'occasion de la passer grâce à l'option **Skip this step**.

6 Orthorectification

Cette étape est utile lorsque la vidéo n'a pas été prise avec un angle de nonante degrés par rapport à la surface de la rivière. Dans ce cas, une orthorectification peut être effectuée permettant de redresser la vidéo comme si elle avait été prise perpendiculairement à la surface, comme expliqué à la section 1.3 du chapitre 3.

Une fenêtre apparaît affichant la première image de la vidéo (voir figure 4.12)

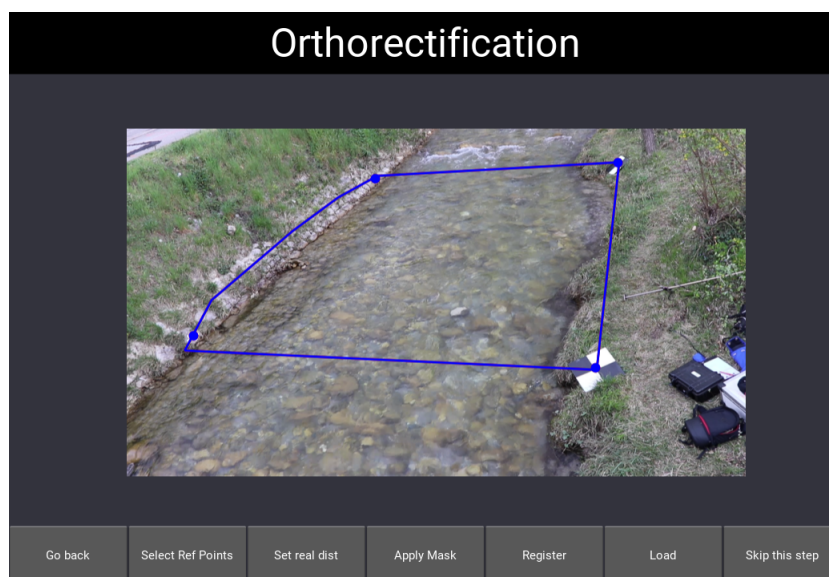


FIGURE 4.12 – Fenêtre affichée à l'orthorectification

Lors de cette étape, l'utilisateur choisit d'abord de placer les quatre points correspondant aux GCPs sur l'image qui apparaît.

L'utilisateur entre ensuite les longueurs, en mètres, entre ces différents points (voir figure 4.13). Pour cela, une fenêtre d'encodage s'ouvre et l'utilisateur peut encoder les longueurs, séparées d'un espace et d'une virgule. Attention, la numérotation de celles-ci correspond à l'ordre dans lequel les points ont été placés.

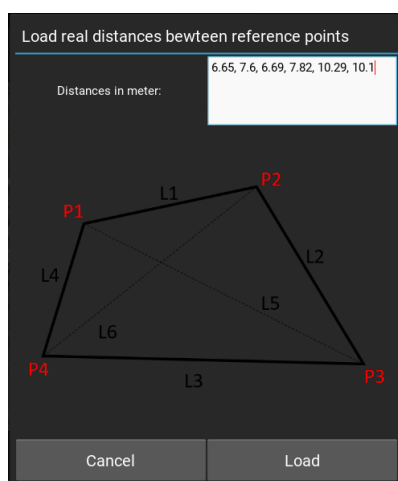


FIGURE 4.13 – Chargement des longueurs entre les points de contrôle

Enfin, l'utilisateur sélectionne un masque, c'est-à-dire une région de l'image qui sera orthorectifiée. L'application de ce masque permet d'accélérer le processus d'orthorectification en éliminant les régions qui n'ont pas d'intérêt à être orthorectifiées (e.g. les berges).

Lorsque les différents paramètres sont chargés, l'utilisateur passe à l'étape suivante. Les paramètres d'orthorectification (tels que les points de contrôle, le masque, etc) sont alors enregistrés comme variables globales de l'application. A nouveau, il n'est pas obligatoire de réaliser une orthorectification. L'utilisateur a donc l'opportunité de la passer si elle n'est pas nécessaire.

7 Détermination du niveau d'eau

La détermination du niveau d'eau est un des enjeux majeurs de cette application. Pour connaître celui-ci, plusieurs méthodes ont été implémentées (voir figure 4.14).

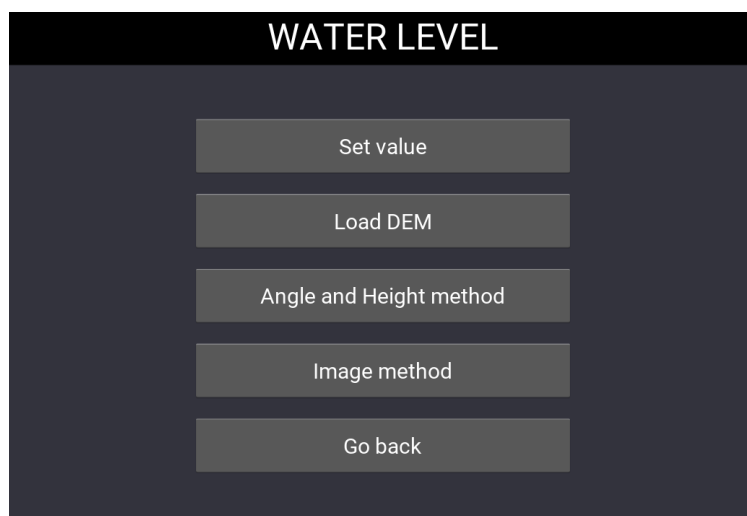


FIGURE 4.14 – Choix de la méthode pour déterminer le niveau d'eau

1. **Set value** : Une fenêtre s'affiche demandant à l'utilisateur de fournir le niveau d'eau en mètres (voir figure 4.15). Attention, le niveau d'eau renseigné par l'utilisateur doit être exprimé dans le même repère que celui défini par la bathymétrie.

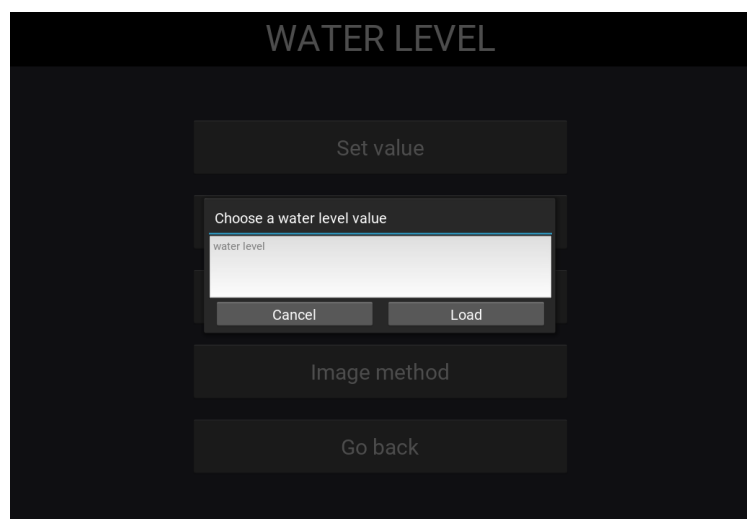


FIGURE 4.15 – Fenêtre qui permet de charger le DEM

2. **Load DEM** : L'utilisateur devra charger un fichier **.tiff** contenant le DEM (i.e. Digital Elevation Model) du cours d'eau. Il devra également fournir les limites (x, y) du domaine modélisé par ce DEM (en mètres), la précision du DEM (en $\left[\frac{mm}{px}\right]$) et la distance à laquelle le DEM doit être comparé à la bathymétrie (en mètres). Une fois ces informations chargées, une fenêtre s'ouvre affichant la bathymétrie et le DEM sur le même graphe. Par défaut, le programme a sélectionné le point où le DEM et la bathymétrie sont les plus proches et ce point correspond à la hauteur d'eau. Si la détermination de ce point (et donc du niveau d'eau) semble incorrecte, l'utilisateur a la possibilité de déplacer ce point et donc d'obtenir une nouvelle hauteur d'eau.

Name	Size
..\	
Modèle_12Mars_V2.tif	26 MB
currentDEM.txt	20 KB

Domain of DEM: e.g. : -1,2 -1,2

Precision of DEM in mm/pixel: e.g. : 1

Distance up to compare in m: e.g. : 1

Buttons: Cancel, Load

FIGURE 4.16 – Fenêtre qui permet de charger le DEM

3. **Angle and height method** : Pour cette méthode, il est obligatoire d'avoir réalisé une calibration de la vidéo. L'utilisateur doit fournir deux images de la rivière prises à deux altitudes différentes ainsi que la hauteur ΔH (en mètres) entre celles-ci (voir figure 4.17). Une fois qu'il a chargé ces informations, une première fenêtre s'ouvre contenant la première image sur laquelle l'utilisateur sélectionne deux points correspondant à la largeur de la rivière. Une seconde fenêtre s'ouvre ensuite où l'utilisateur est à nouveau invité à placer ces deux points. Grâce aux règles trigonométriques, la largeur de la rivière peut être calculée et celle-ci est détectée dans la bathymétrie pour en déduire le niveau d'eau.

Select image at the lower altitude		Select image at the higher altitude	
Name	Size	Name	Size
..\		..\	
distances.png	17 KB	distances.png	17 KB
imgCloseFromVideo.png	3 MB	imgCloseFromVideo.png	3 MB
imgFarFromVideo.png	3 MB	imgFarFromVideo.png	3 MB
reperes.png	1 KB	reperes.png	1 KB

Delta of height between the two pictures in meter: e.g. : 1

Buttons: Go back, LOAD

FIGURE 4.17 – Fenêtre de la méthode Angle et Hauteur

4. **Image Method** : Pour cette méthode, il est obligatoire de réaliser l'étape de l'orthorectification au préalable. L'utilisateur va placer un point sur chaque rive, la distance entre ceux-ci correspondant à la largeur de la rivière (voir figure 4.18). Cette largeur est ensuite détectée dans la bathymétrie et la profondeur d'eau en est déduite.

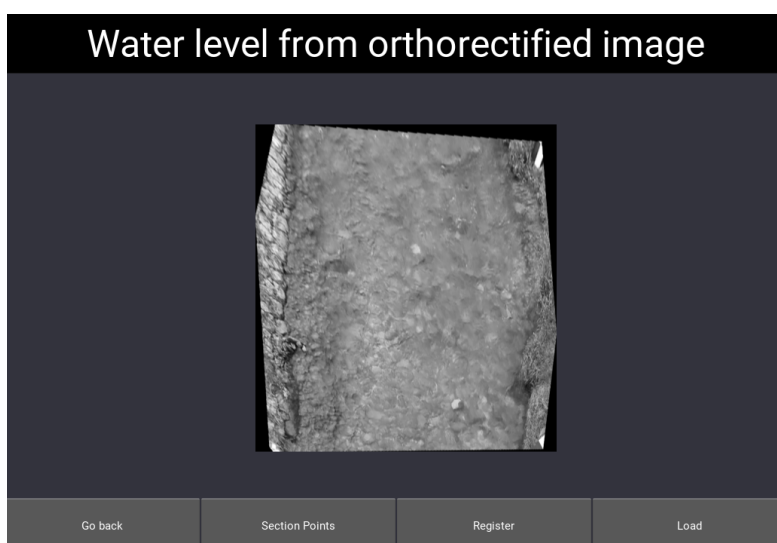


FIGURE 4.18 – Fenêtre de la méthode Image

Après avoir choisi une méthode parmi celles présentées ci-dessus, le niveau d'eau en mètres est enregistré en tant que variable globale de l'application.

8 Région d'intérêt et masques

Durant cette étape, l'utilisateur doit sélectionner, sur la première image de la vidéo (voir figure 4.19), une région d'intérêt, c'est-à-dire une région où il souhaite que l'analyse PIV soit réalisée. Une fois celle-ci enregistrée, l'utilisateur sélectionne, dans cette région d'intérêt, un ou plusieurs masques, c'est-à-dire des régions où les vitesses ne seront pas calculées. Ensuite, il enregistre. Souvent, l'utilisateur sélectionnera le masque à l'endroit des berges car celles-ci ne doivent pas être analysées pour déterminer le débit.

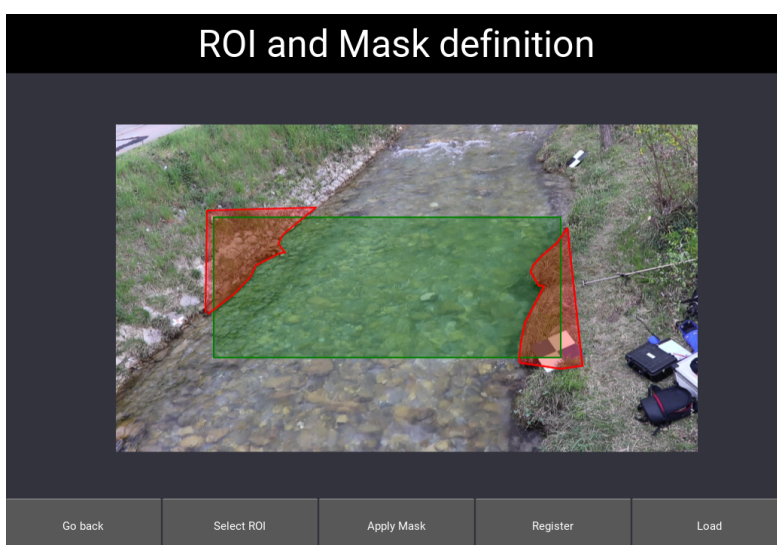


FIGURE 4.19 – Fenêtre affichée à l'étape du ROI et du masque

La sélection de la région d'intérêt et des masques permet d'accélérer le processus d'analyse en réduisant la surface sur laquelle celle-ci est réalisée et en éliminant les parties dont l'analyse est inutile.

Les points qui constituent la région d'intérêt et les masques sont stockés sous forme de tableau en tant que variables globales de l'application. De même, l'image où la région d'intérêt et les masques sont appliqués est enregistrée parmi les variables.

9 Filtres

L'application de filtres sur la vidéo permet d'accentuer le contraste entre les particules et le fond. Celles-ci sont alors plus faciles à tracer lors de l'analyse de vitesse. L'application d'un ou plusieurs filtres est donc laissée à l'appréciation de l'utilisateur (voir figure 4.20). Il peut également choisir de n'en appliquer aucun et passer à l'étape suivante.

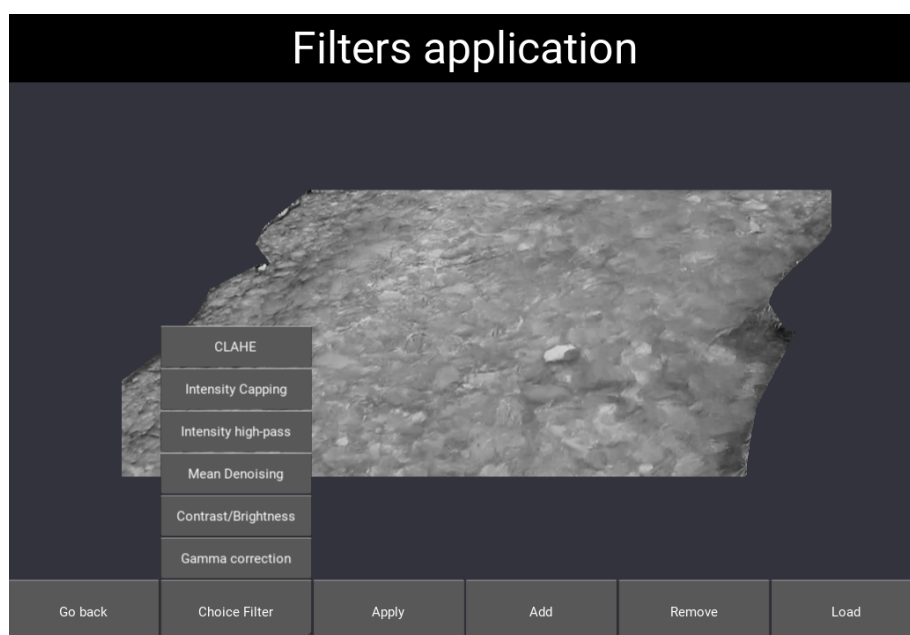


FIGURE 4.20 – Fenêtre permettant la sélection des filtres

L'utilisateur a le choix entre six filtres différents :

1. CLAHE
2. Intensity Capping
3. Intensity high-pass
4. Mean Denoising
5. Contrast/Brightness
6. Gamma Correction

L'effet de l'application de chaque filtre est détaillé dans la section 1.4 (27). Pour chaque filtre, l'utilisateur peut changer les paramètres proposés par défaut pour modifier le résultat. Le filtre est appliqué sur la première image de la vidéo (uniquement dans la région d'intérêt choisie et avec le masque appliqué) afin de donner un visuel. Si le contraste est suffisamment marqué, l'utilisateur peut l'ajouter à la liste des filtres à appliquer. Si le résultat n'est pas convaincant, le filtre peut être effacé de la liste.

Enfin, lors du chargement, les filtres sélectionnés sont appliqués à la vidéo qui est ensuite stockée dans le dossier "Video" de l'application pour éventuellement être réutilisée lors d'une

autre analyse. Les filtres ainsi que les paramètres associés à ceux-ci sont stockés comme variables globales de l'application.

10 Analyse PIV

Cette étape permet de déterminer les vitesses de surface du cours d'eau. La vidéo analysée est celle qui a été filtrée et l'analyse est réalisée sur la région d'intérêt dont les parties masquées sont retirées. Le range de la vidéo sur lequel l'analyse est appliquée est celui choisi par l'utilisateur lors du chargement de la vidéo. Plus celui-ci est court, plus l'analyse sera rapide. L'analyse prend également en compte le step choisi par l'utilisateur. Par exemple, si celui-ci choisit un step de 3, l'analyse de vitesse ne tiendra compte que d'une image sur 3. Ce processus permet d'accélérer l'analyse et d'améliorer la précision. Une fenêtre s'affiche (voir figure 4.21), permettant à l'utilisateur de modifier les paramètres Interrogation Window Size, Overlap et SNR ratio (voir section 3.2 du chapitre 3).

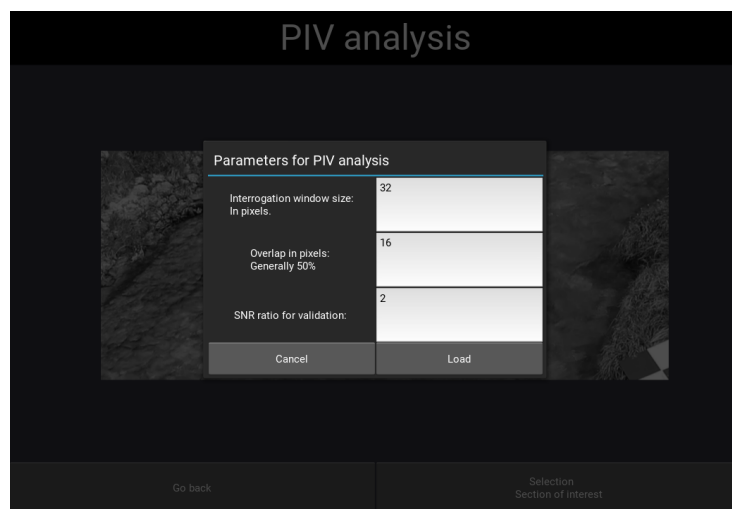


FIGURE 4.21 – Choix des paramètres de l'analyse PIV

Une fois ces trois paramètres validés, l'analyse de vitesse commence. Une fenêtre affichant une barre de chargement s'affiche afin d'informer l'utilisateur de l'avancée de l'analyse car celle-ci peut prendre du temps (voir figure 4.22).

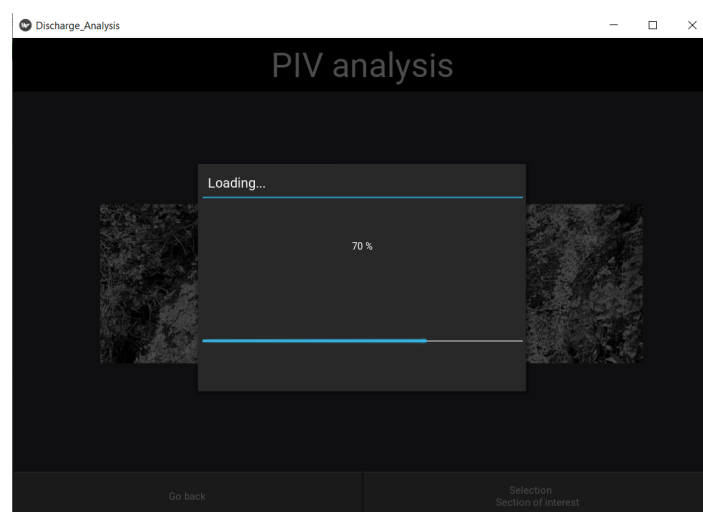


FIGURE 4.22 – Chargement de l'analyse de vitesses

Dès que l'analyse est terminée, la première image de la vidéo analysée s'affiche avec les vecteurs vitesses moyens relatifs à chaque point (voir figure 4.23).

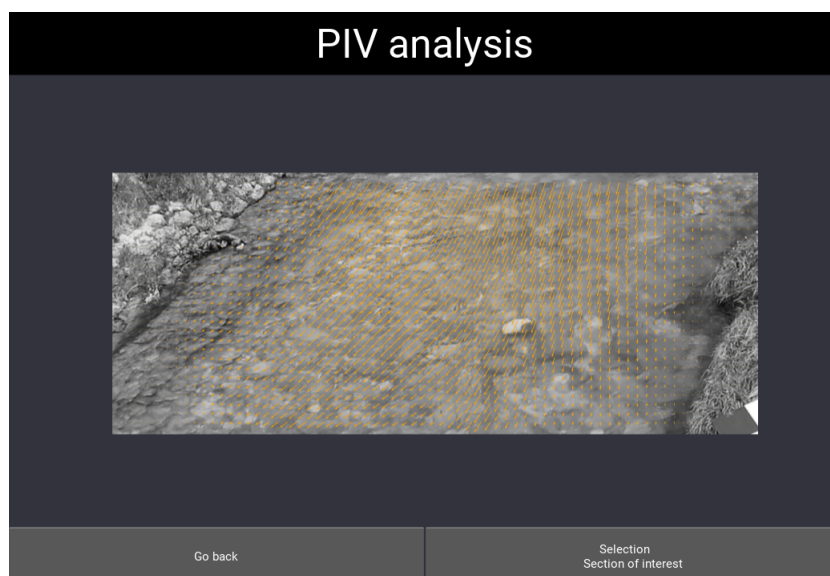


FIGURE 4.23 – Affichage des vecteurs vitesse moyens calculés

Les vitesses sont sauvegardées dans un fichier `velocity.npy`. Les vecteurs vitesse sont enregistrés en tant que variables globales de l'application.

11 Section d'intérêt

La première image de la vidéo s'affiche (voir figure 4.24). Si l'utilisateur a réalisé une orthorectification, il s'agit de l'image orthorectifiée. Sinon, il s'agit simplement de la première image de la vidéo.

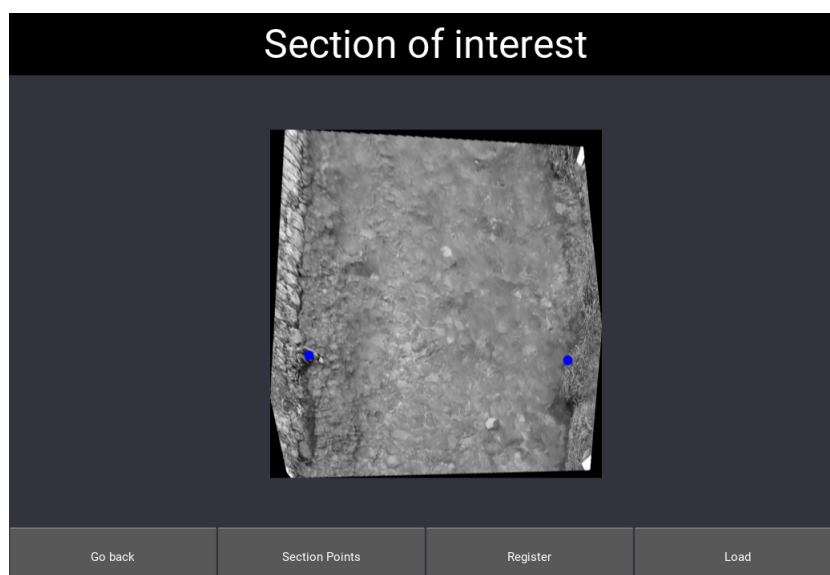


FIGURE 4.24 – Fenêtre où l'utilisateur choisit la section d'intérêt

Il est demandé dans cette étape de sélectionner, sur l'image affichée, la section sur laquelle le calcul de débit sera réalisé. L'utilisateur choisit deux points face-à-face de part et d'autre de la

rivière. Ensuite, il les enregistre.

Pour obtenir le débit le plus proche possible de la réalité, il faudra veiller à choisir la section d'intérêt le plus proche possible de la section sur laquelle la bathymétrie a été mesurée.

Une fenêtre s'ouvre ensuite afin que l'utilisateur puisse fournir le nombre de sous sections désiré (voir figure 4.25). Celui-ci est utilisé dans le calcul du débit, la section totale sera ainsi divisée en autant de sous sections. Sur chacune de celles-ci, un débit sera calculé.

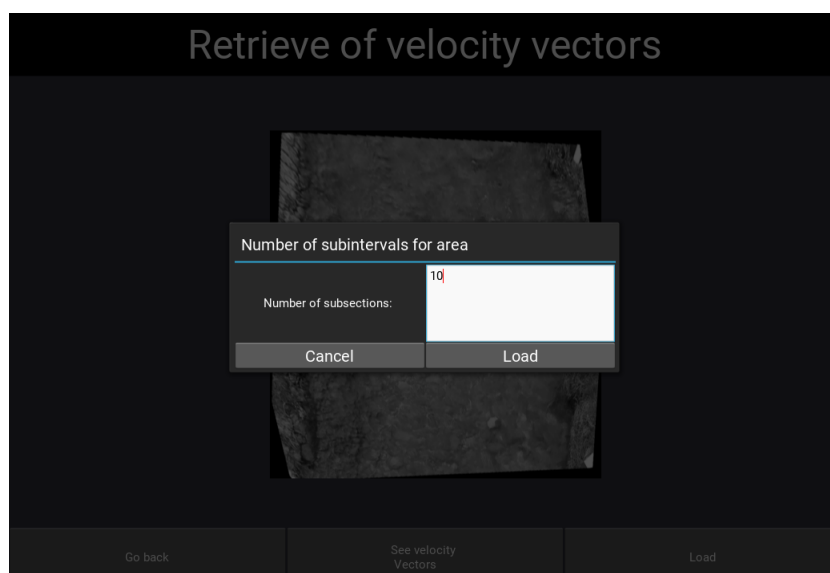


FIGURE 4.25 – Choix du nombre de sous sections

La largeur est alors affichée sur la première image de la vidéo. Si celle-ci n'est pas parfaitement horizontale, l'image subit une rotation afin que la largeur soit horizontale et que seules les composantes de vitesse verticales soient prises en compte pour le calcul de débit.

Ensuite, l'utilisateur peut afficher les vecteurs vitesses moyens de surface de la section choisie afin d'afficher profil de vitesse (voir figure 4.26).

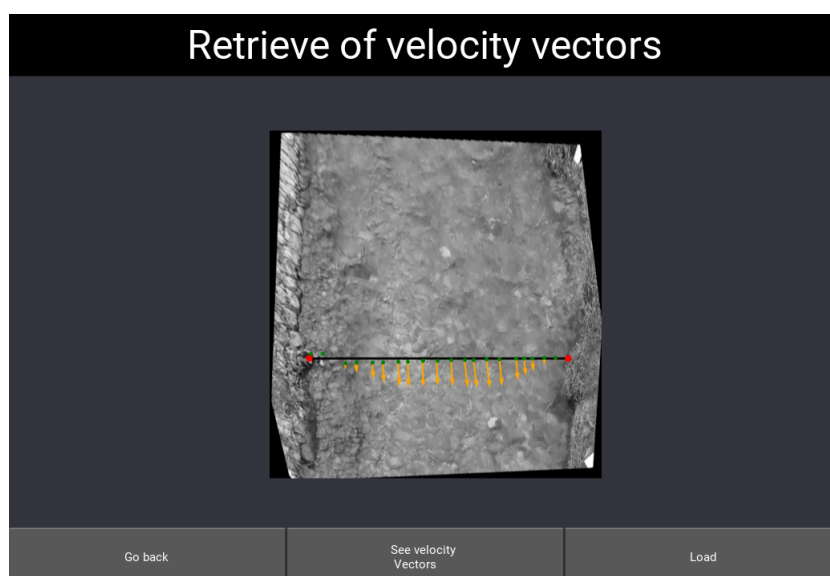


FIGURE 4.26 – Affichage des vecteurs vitesses perpendiculaires à la section choisie

12 Détermination du débit

Il s'agit de la dernière étape. Une fenêtre apparaît pour que l'utilisateur puisse entrer le coefficient liant la vitesse moyenne de surface à la vitesse moyenne de profondeur (voir figure 4.27). Par défaut, celui-ci a une valeur de 0,8 ou 0,9 dépendant de la profondeur d'eau déterminée auparavant. Toutefois, sans plus d'informations sur celui-ci, il est conseillé d'utiliser la valeur de 0,85 comme discuté dans la section 4 du chapitre 3.

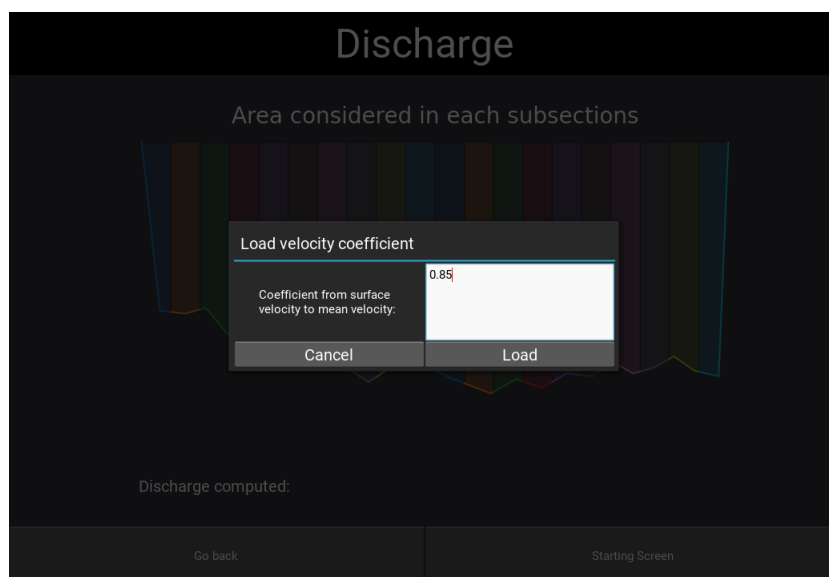


FIGURE 4.27 – Chargement du coefficient α

Le débit calculé est alors affiché en $\left[\frac{m^3}{s}\right]$ ainsi que la section subdivisée (voir figure 4.28).

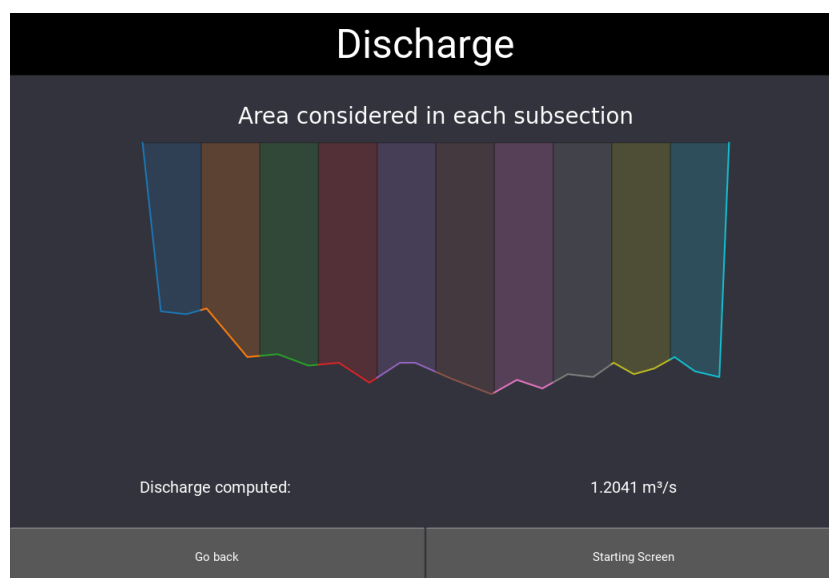


FIGURE 4.28 – Fenêtre affichant le débit

Les données concernant les vitesses, la hauteur d'eau et le débit calculé sont enregistrées dans un fichier `analysisResults.npy`.

L'utilisateur a alors la possibilité de retourner à l'écran de départ pour effectuer une nouvelle analyse.

Chapitre 5

Tests effectués sur l'application

Afin de se familiariser avec le code de l'application, une série de tests ont été effectués à l'aide d'un set de vidéos de référence. Ces tests ont également permis de nous aiguiller dans notre tâche de vérification des étapes de l'application.

Au fur et à mesure des tests effectués, différents problèmes dans le fonctionnement de l'application et dans les résultats obtenus ont été constatés. Ceux-ci seront détaillés dans le prochain chapitre.

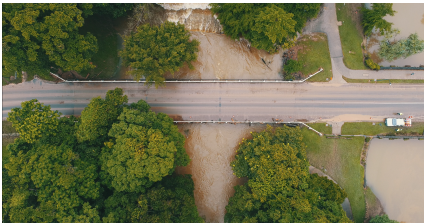
1 Vidéo globe challenge

Les vidéos de référence mises à notre disposition sont issues du Vidéo Globe challenge. Il s'agit d'un challenge auxquels ont participé les étudiants du précédant TFE sur la mesure de débit en rivière. L'objectif de ce challenge était de déterminer les débits des cours d'eau dont la bathymétrie et une vidéo de l'écoulement étaient fournis. L'avantage d'utiliser ces vidéos pour une prise en main est que les débits réels des différents cours d'eau sont fournis. Ils pourront donc être comparés aux débits obtenus par l'application.

Les vidéos de ce challenge sont filmées dans des conditions très variables (voir tableau 1) qui nécessitent ou non l'application de certaines étapes. De plus, les différentes informations fournies relatives aux vidéos ne sont pas toujours identiques. Par exemple, certaines vidéos sont filmées perpendiculairement au plan d'eau et ne nécessitent pas d'orthorectification.

Ces diversités de cas, illustrées dans le tableau 1, ont permis, lors des tests sur l'application, de déceler les cas limites auxquels l'application n'est pas adaptée mais aussi les erreurs présentes dans celle-ci et les points à améliorer.

Cependant, par manque de données, certaines parties de l'application n'ont pas pu être testées telles que la calibration et les méthodes d'Hellebaut et Angle et Hauteur.

Vidéo	Conditions
1 	Tronçon droit Vidéo filmée de la berge Particules ajoutées
2 	Tronçon droit Vidéo filmée perpendiculairement au plan d'eau Niveau d'eau donné
3 	Vidéo prise dans un méandre Vidéo filmée de la berge Les GCPs ne sont pas dans le même plan Mouvements de la caméra
4 	Tronçon droit Vidéo filmée perpendiculairement au plan d'eau Niveau d'eau donné Végétation abondante qui couvre la rivière
5 	Tronçon droit Vidéo filmée de la berge

6 	Tronçon droit Vidéo filmée d'un pont Les GCPs ne sont pas dans le même plan Mouvements de la caméra
7 	Tronçon droit Vidéo filmée perpendiculairement au plan d'eau Niveau d'eau donné
8 	Tronçon droit Vidéo filmée de la berge Mouvements de la caméra Zoom dans la vidéo

2 Méthodologie

La méthodologie appliquée lors de la phase de test peut être résumée en trois points principaux. Tout d'abord, les limites du set de vidéo ont été déterminées. En effet, il était important de savoir quelles vidéos étaient les plus adaptées pour tester chaque partie de l'application. Ensuite, la liste des paramètres à faire varier a été dressée. Les premiers tests étaient essentiels pour avoir une première idée de l'influence de ces paramètres. Enfin, nous avons défini deux types de tests que nous avons ensuite utilisés pour identifier les corrections à apporter et les améliorations possibles.

2.1 Utilisation du set Vidéo Globe Challenge

Calibration

Comme les données des caméras utilisées pour les vidéos du *Vidéo Globe Challenge* n'étaient pas disponibles, aucun test n'a pu être effectué sur la partie calibration de l'application. De plus, comme la méthode *Angle et hauteur* nécessite une calibration préalable, elle n'a pas non plus pu être testée.

Orthorectification

Un grand nombre de vidéos du set de tests sont filmées de la berge et nous ont donc permis de tester l'étape d'orthorectification. Parmi celles qui sont utilisables, se trouvent les vidéos 1, 3, 5, 6 et 8. Mais comme l'étape d'orthorectification implémentée dans l'application est une

transformation géométrique 2D, les vidéos 3 et 6 sont à exclure car leur *Groud Control Points* ne sont pas dans un même plan et ne sont pas au niveau de l'eau.

Détermination du niveau d'eau

Les données disponibles avec les vidéos nous ont permis de tester deux méthodes de détermination du niveau d'eau : le **set value** lorsque la hauteur d'eau était fournie et l'**image method** applicable pour les vidéos dont l'orthorectification 2D peut être réalisée.

2.2 Paramètres à faire varier

Pour obtenir un débit semblable au débit réel, il est possible de faire varier plusieurs paramètres et de constater l'effet de ce changement sur les résultats des vitesses et du débit. Les différents paramètres qui peuvent être modifiés sont les suivants :

- Range et step : Pour accélérer l'analyse de vitesse, le rang de la vidéo ainsi que le step choisi peuvent être changés. Par exemple, pour un même rang de la vidéo, augmenter le step et regarder son influence est une démarche intéressante.
- Les différents filtres accentuent l'influence des particules dans le calcul des vitesses.
- Le nombre de sous sections influence le calcul du débit.
- Le coefficient α fait le lien entre les vitesses de surface et les vitesses moyennes de l'écoulement.

2.3 Types de test

Parmi les tests réalisés, deux types de tests peuvent être distingués. Ceux-ci ne présentent pas les mêmes objectifs et sont détaillés dans les sections ci-dessous.

Tests identiques

Dans un premier temps, avant que les différents problèmes n'aient été détectés, une série de tests identiques ont été réalisés sur les vidéos. Les conditions de ces tests étaient exactement les mêmes afin de pouvoir comparer les débits et de regarder les écarts présents dans les résultats.

Sur la figure 5.1 se trouvent un exemple de résultats obtenus à partir de tests réalisés sur le jeu 1.

Set de tests Jeu 1 avant modification - Q = 1,26 m³/s									
i	Stab	Filtres	Range	Step	ALPHA	nbe subsections area	L[m]	WL [m]	Q [m³/s]
1	Non	CLAHE	TOUT	3	0,85	20	5,526	-0,128	0,4031
2	Non	CLAHE	TOUT	3	0,85	20	5,818	0,569	1,7846
3	Non	CLAHE	TOUT	3	0,85	20	5,629	0,119	0,8331
4	Non	CLAHE	TOUT	3	0,85	20	5,839	0,62	1,8137
5	Non	CLAHE	TOUT	3	0,85	20	5,654	0,178	1,0323
6	Non	CLAHE	TOUT	3	0,85	20	5,783	0,484	1,8147
7	Non	CLAHE	TOUT	3	0,85	20	5,692	0,27	1,1814
8	Non	CLAHE	TOUT	3	0,85	20	5,706	0,303	1,1296
9	Non	CLAHE	TOUT	3	0,85	20	5,747	0,4	1,3469
10	Non	CLAHE	TOUT	3	0,85	20	5,553	-0,061	0,5988
11	Non	CLAHE	TOUT	3	0,85	20	5,662	0,196	1,0249
12	Non	CLAHE	TOUT	3	0,85	20	5,759	0,428	1,7078
13	Non	CLAHE	TOUT	3	0,85	20	5,66	0,194	1,0819
14	Non	CLAHE	TOUT	3	0,85	20	5,62	0,099	0,9126
15	Non	CLAHE	TOUT	3	0,85	20	5,694	0,273	1,2828
16	Non	CLAHE	TOUT	3	0,85	20	5,726	0,351	1,2225
17	Non	CLAHE	TOUT	3	0,85	20	5,646	0,159	1,0509
18	Non	CLAHE	TOUT	3	0,85	20	5,719	0,333	0,8737
19	Non	CLAHE	TOUT	3	0,85	20	5,724	0,345	1,3335
20	Non	CLAHE	TOUT	3	0,85	20	5,744	0,394	1,5278

FIGURE 5.1 – Exemple de tests identiques réalisés sur le jeu 1

Un certain écart entre les débits obtenus et le débit réel est observé. Étant donné que cet écart est assez important, celui-ci ne peut pas être dû à la seule erreur de l'utilisateur. Il faudra donc approfondir les recherches sur ce point.

Tests variables

Dans un second temps, après les premières corrections et une fois que les résultats obtenus dans les mêmes conditions étaient semblables, des tests dans différentes conditions ont été réalisés. Ceux-ci avaient pour objectif de déterminer l'influence des différents paramètres modifiables dans l'obtention du débit. A nouveau, ces tests ont permis de détecter l'influence anormale de certains paramètres et d'effectuer des corrections dans ce sens.

Chapitre 6

Problèmes détectés et améliorations possibles suite aux tests effectués

Suite aux tests effectués et décrits précédemment, certaines anomalies ont été détectées. De plus, grâce à l'utilisation intensive de l'application, certaines pistes d'amélioration ont été remarquées.

1 Problèmes détectés

1.1 Range

Pour rappel, le range est l'option qui permet à l'utilisateur de sélectionner un intervalle de la vidéo à analyser. Bien que cette fonction soit simple, elle n'en est pas moins très importante. En effet, l'analyse d'une vidéo complète de trente secondes peut prendre jusqu'à une heure de calcul. De plus, si l'utilisation des traceurs est nécessaire, il est primordial de pouvoir isoler les secondes de vidéo où ils sont les plus visibles.

Malheureusement, cette fonctionnalité ne semblait pas être correctement implémentée à l'étape de la calibration. En effet, lorsque le range choisi ne commençait pas à 0 (soit `Range[0] ≠ 0`) et que l'utilisateur choisissait de calibrer la vidéo, l'analyse PIV ne pouvait être réalisée sur la vidéo calibrée. L'application restait alors en pause avant que la barre de chargement n'ait pu se lancer.

La même erreur a été constatée lors de l'application des filtres. Peu importe le filtre choisi, l'analyse PIV ne se lançait pas sur la vidéo filtrée dont le `Range[0]` était différent de 0.

1.2 Step

Le step est une fonctionnalité qui apparaît lors du chargement de la vidéo. Initialement égal à 1, l'utilisateur a la possibilité de modifier sa valeur. Par exemple, si la valeur ne vaut plus un mais cinq, cela signifie que le programme n'analysera plus toutes les images de la vidéo mais seulement une image sur cinq. Cela permet d'accélérer le processus d'analyse de vitesse.

Sur la vidéo 2 du set, nous avons réalisé une série de tests identiques. Puis en faisant varier les paramètres du step et du filtre, nous avons noté une influence anormale du step choisi par l'utilisateur (voir Figure 6.1).

Set de tests - Jeu 2						
Test	SF	Stab	Filtres	Range	Step	Q [m³/s]
26	23,23	oui	Aucun	Tout	3	226,6221
27	23,22	oui	Aucun	Tout	3	234,0873
28	23,46	oui	Aucun	Tout	3	212,3123
29	23,47	oui	Aucun	Tout	3	219,2818
30	22,746	oui	CLAHE	Tout	3	235,4806
31	22,96	oui	CLAHE	Tout	3	227,2626
32	23,0321	oui	CLAHE	Tout	3	222,5761
33	23	oui	CLAHE	Tout	3	206,7823
34	22,4	oui	IC	Tout	3	249,1167
35	23,06	oui	IC	Tout	3	214,5551
36	23,249	oui	IC	Tout	3	207,1324
37	23,318	oui	IC	Tout	3	218,5942
38	23,488	oui	IC	Tout	3	204,0039
39	23,2029	oui	IC	Tout	3	218,1507
40	23,305	oui	CLAHE	Tout	5	313,798
41	23,19	oui	CLAHE	Tout	5	307,6504
42	23,22	oui	CLAHE	Tout	5	350,3793

FIGURE 6.1 – Erreur de step sur le Jeu 2

Nous remarquons qu'avec un step de 3, le débit était toujours compris entre 200 et 250 $[\frac{m^3}{s}]$. Lorsque l'utilisateur changeait de step et faisait varier celui-ci à 5 (voir les trois derniers tests sur la figure 6.1), le débit augmentait considérablement et était alors situé aux alentours de 300 $[\frac{m^3}{s}]$. L'influence de ce step était anormalement importante. En effet, même si on analyse qu'une seule image sur cinq au lieu d'une image sur trois, le débit ne doit pas varier autant. Il y avait donc un manquement au niveau de l'implémentation du step.

1.3 Analyse PIV

Comme expliqué précédemment, à l'issue de cette étape, la première image de la vidéo de l'écoulement est affichée ainsi que les vecteurs vitesses résultant de l'analyse (voir figure 4.23). La répartition des flèches attendue est donc uniforme, sans zone où aucune flèche n'est dessinée. Or, sur la plupart des vidéos où l'analyse PIV était réalisée, une zone "vide" sans vitesse apparaissait (voir figures 6.2, 6.3 et 6.4).

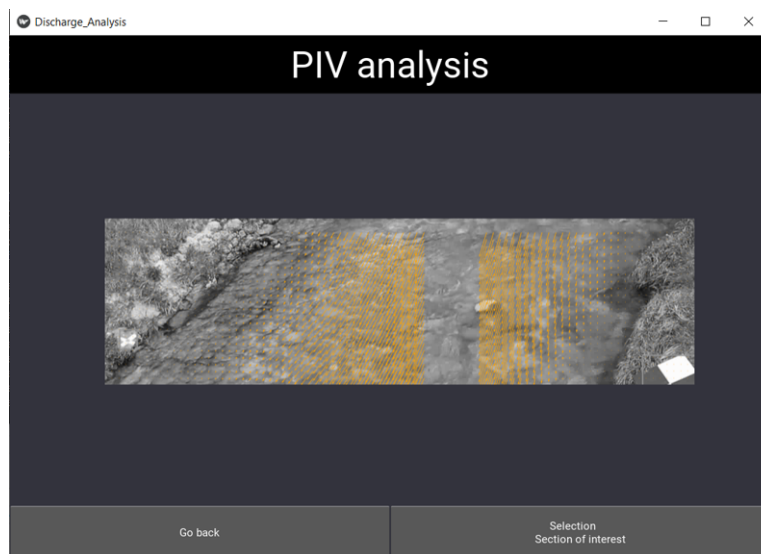


FIGURE 6.2 – Illustration du problème de vitesse dans le jeu 1

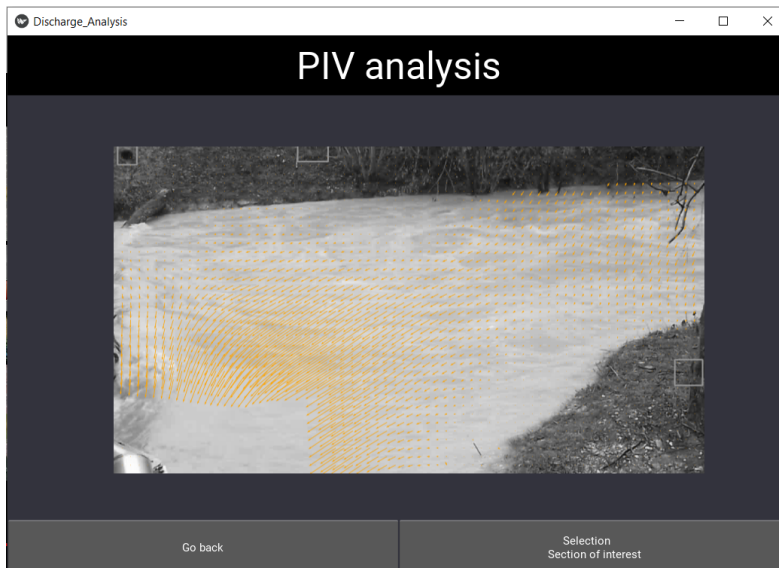


FIGURE 6.3 – Illustration du problème de vitesse dans le jeu 3

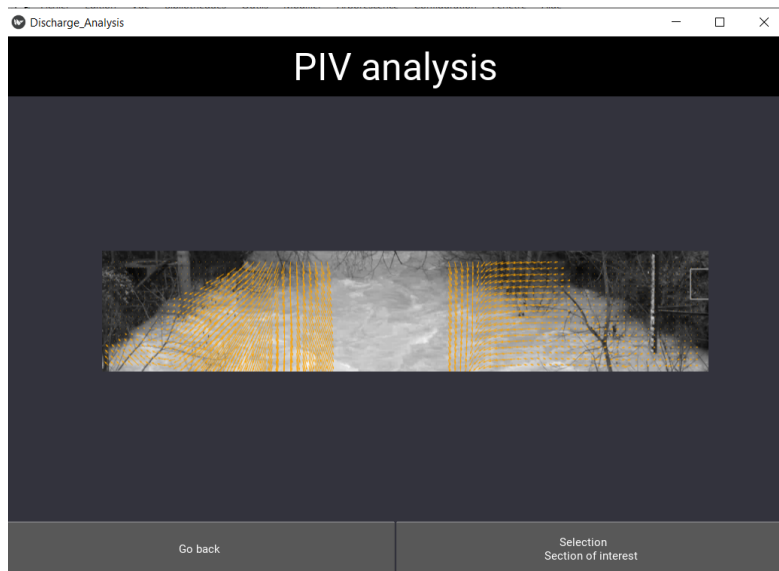


FIGURE 6.4 – Illustration du problème de vitesse dans le jeu 6

Dès lors, l’affichage des vecteurs vitesses orthorectifiés n’était pas non plus correct puisqu’il se base sur le calcul des vitesses. Le profil de vitesse 2D sur la surface de l’eau devrait normalement suivre une répartition telle que les vitesses près des berges (donc sur le lit mineur) sont plus faibles que les vitesses au centre (sur le lit majeur) comme représenté sur la figure 6.5.

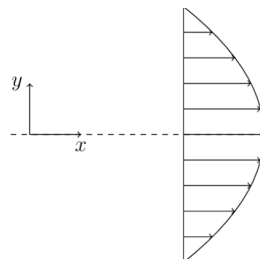


FIGURE 6.5 – Profil de vitesse à la surface attendu d’un écoulement

Hors, suite à l'analyse de vitesse réalisée, le profil de répartition de vitesse obtenu ne ressemblait pas à celui attendu (voir figures 6.6).

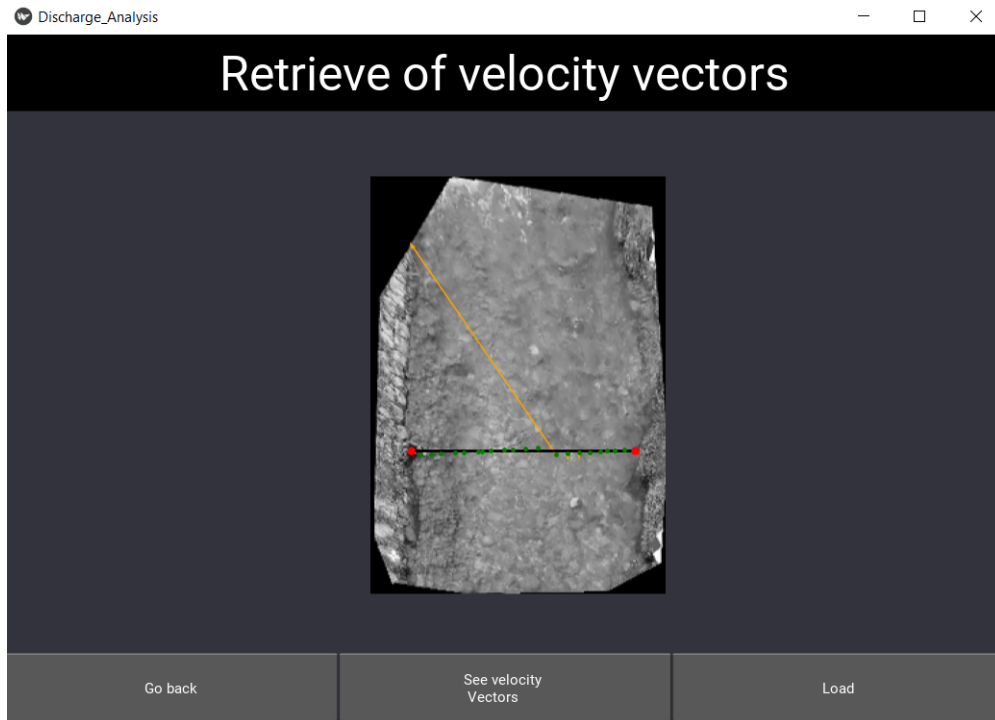


FIGURE 6.6 – Profil de vitesse - Jeu 1

De plus, lors de l'analyse des débits, nous avons remarqué que ceux-ci étaient très variables. Malgré l'application de conditions relativement identiques sur la vidéo, les résultats variaient tout de même énormément. La variabilité constatée paraît trop importante pour être causée par l'erreur de l'utilisateur.

1.4 Mask

Pour accélérer l'analyse PIV, une région d'intérêt rectangulaire peut être fixée, comme présenté aux sections 3.1 du chapitre 3 et 8 du chapitre 4. Un masque, appelé *mask* dans l'application, peut également être appliqué à cette région d'intérêt pour y cacher des éléments non intéressants pour l'analyse PIV telles que les berges par exemple.

Durant nos premiers tests, des zones de part et d'autre de l'écoulement où les vecteurs vitesses étaient manquants apparaissaient régulièrement. Le problème est particulièrement marquant sur le résultat présenté à la figure 6.7. Nous avons rapidement émis l'hypothèse que ce problème pouvait venir de l'implémentation du masque mais la localisation de celui-ci dans le code n'a pas été aussi simple.

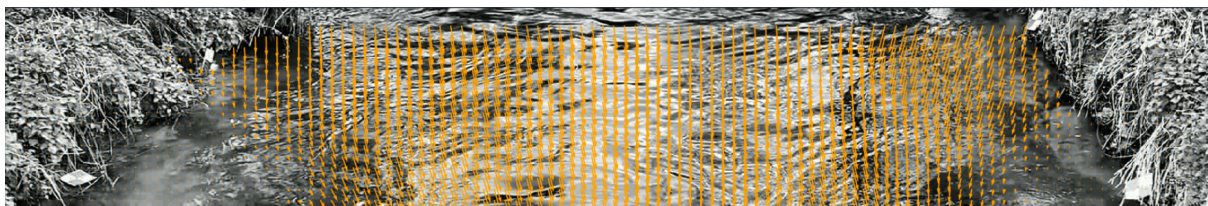


FIGURE 6.7 – Erreur causée par l'implémentation du masque

1.5 Facteur d'échelle

Le facteur d'échelle permet le passage des vitesses calculées en pixels par seconde vers des vitesses en mètres par seconde. Il est utilisé à deux endroits dans le code : lors de l'orthorectification et lors du calcul du débit. Pour calculer celui-ci, il est nécessaire de connaître pour une même distance, sa longueur en pixel et sa longueur en mètre afin d'en calculer le rapport. Le facteur d'échelle est calculé différemment selon le cas où l'utilisateur réalise une orthorectification.

Dans le cas de l'orthorectification, celui-ci était fixé à 30, correspondant à une certaine vidéo exemple. Or, comme expliqué dans la section 1.3 du chapitre 3, ce facteur ne peut pas être fixé complètement arbitrairement à cause de la contrainte d'affichage.

Dans le cas sans orthorectification, le facteur d'échelle n'était simplement pas défini et conduisait à l'arrêt soudain de l'application.

Sur ce point, le code n'était donc pas complet et demandait d'être approfondi.

1.6 Erreurs de code

Boutons back

Une erreur au niveau de certains boutons "back" qui permettent le retour à l'étape précédente en cas d'erreur de l'utilisateur a été détectée. L'étape réalisée auparavant ne s'effaçait pas correctement de l'application et rendait l'usage de ces boutons impossible.

Barre de chargement : implémentation du step

Lors de l'utilisation d'un step, la barre de chargement de l'analyse PIV n'atteignait jamais les cent pourcents. En effet, lorsqu'un step de trois était appliqué, la barre de chargement n'allait pas au delà de 33 pourcents soit cent divisé par trois.

Affichage de la largeur

Lors de l'étape de l'affichage de la section d'intérêt et des vecteurs de vitesse, l'affichage ne se faisait pas correctement. La largeur s'affichait dans une zone en dehors de l'image, loin de là où l'utilisateur l'avait placée initialement. De plus, les vecteurs vitesses n'étaient pas placés sur cette largeur. Cette erreur ne se produisait que dans le cas où aucune orthorectification n'avait été réalisée. Une illustration de ce problème est affichée à la figure 6.8.

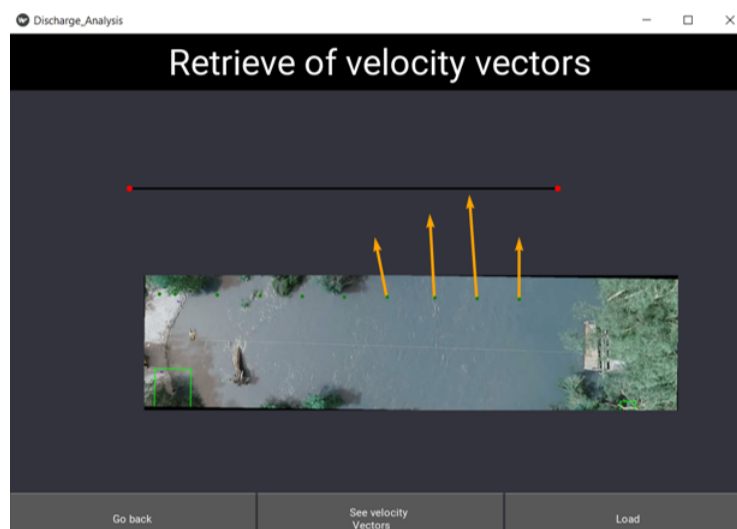


FIGURE 6.8 – Affichage de la largeur et des vitesses sur le jeu 2

2 Améliorations possibles

2.1 Manque de robustesse générale

L'une des premières difficultés que nous avons rencontrées lors de l'utilisation de l'application est le manque de robustesse général de celle-ci. En effet, lorsqu'un utilisateur utilise une application, il arrive couramment qu'il agisse trop rapidement ou ne fasse pas spécifiquement attention à réaliser correctement les différentes étapes. Dans ce cas, l'application n'anticipait pas les problèmes tels que le manque de données ou la mauvaise réalisation des diverses étapes. Elle se fermait donc anormalement ou restait en pause à la moindre information manquante. Ce manque d'anticipation rendait vite l'utilisation de l'application pénible et difficile pour un utilisateur non habitué ou non aguerri.

Par exemple, lors du chargement du fichier contenant la bathymétrie, il arrive assez facilement que l'utilisateur clique sur le bouton "Load" sans avoir sélectionné de fichier, par erreur d'inattention. Hors, le programme doit avoir reçu ce fichier pour lancer le chargement. Celui-ci se fermait donc inopinément indiquant une erreur.

2.2 Version Beta

Comme expliqué à la section 3.2 du chapitre 3, la librairie OpenPIV utilisée dans l'application est une version *Beta*. Il s'agit d'une version encore en développement qui peut être l'objet de certaines modifications. Ce détail a toute son importance pour le développement futur de l'application, étant donné la perte de temps qu'il peut occasionner si le développeur n'en est pas averti. Une solution à ce problème serait d'enregistrer les fichiers d'OpenPIV en local parmi les autres fichiers de l'application. De cette façon, le risque qu'une mise à jour empêche le bon fonctionnement de l'application pour son développement et son utilisation future peut être évité. Bien entendu, il sera toujours possible de mettre la version actuellement enregistrée à jour. Cette mise à jour devra alors se faire manuellement et être accompagnée d'une révision du code pour éviter tout problème.

2.3 Méthode Image

Lors de l'utilisation de la méthode image, l'utilisateur est invité à sélectionner deux points correspondant à la largeur. Cette largeur est ensuite détectée dans la section de la rivière pour en déduire le niveau d'eau. Cependant, lorsque la largeur choisie par l'utilisateur n'est pas détectable dans la bathymétrie, le programme se mettait à l'arrêt. L'utilisateur n'avait donc pas la possibilité de progresser vers les autres étapes, de sélectionner d'autres points ou même de revenir en arrière (voir Figure 6.9).

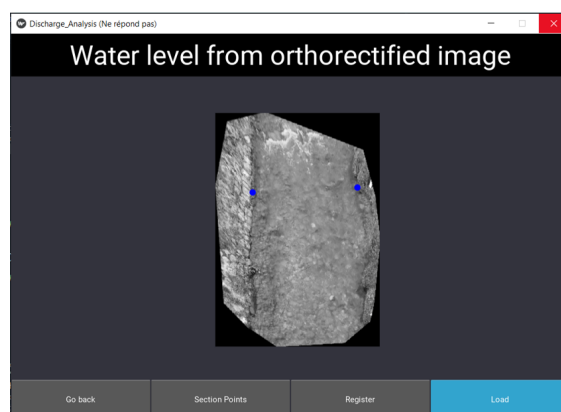


FIGURE 6.9 – Méthode image : largeur non détectée dans la section

De plus, lors de l'exploitation du Jeu 1, la méthode image a été utilisée. Hors, la bathymétrie du jeu 1 est particulière puisque les berges sont très raides (voir figure 6.10).

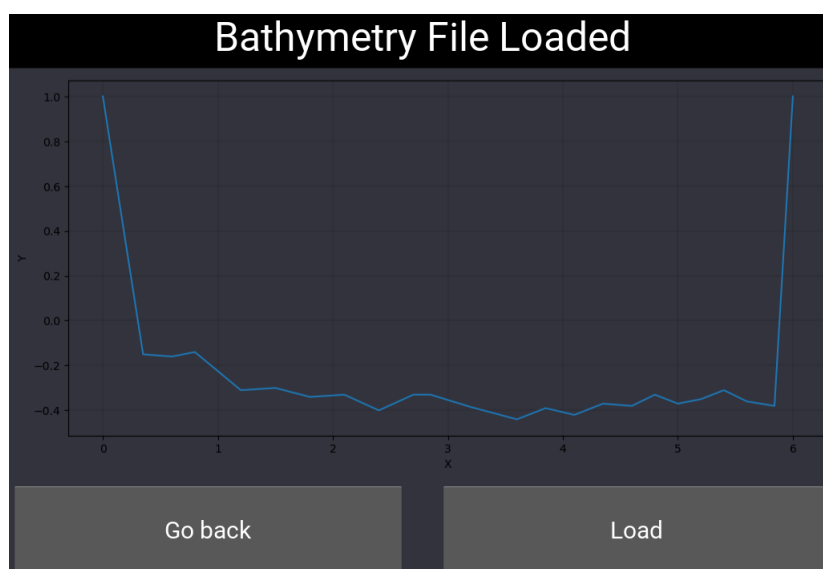


FIGURE 6.10 – Bathymétrie du jeu 1

Dès lors, à chaque sélection, même si les largeurs choisies sont très proches, une petite différence peut induire une grande différence de niveau d'eau et donc de débit. Hors, comme la sélection est manuelle, il est impossible d'avoir toujours exactement la même largeur. Nous observons donc que dans le cas des bathymétries raides, il y a une limite à l'utilisation de cette méthode.

3 Conclusion

Le travail réalisé durant le premier quadrimestre a été une étape clé pour la suite de notre travail. Il nous a permis de découvrir le code et d'établir notre méthodologie de travail. C'était également l'occasion de nous familiariser avec les nombreux concepts utilisés dans l'application. Sur cette base, de nombreux tests ont été réalisés, mettant en évidence les erreurs et des potentielles pistes d'amélioration. C'est donc cette étape qui a défini le travail de correction et d'amélioration à réaliser au second quadrimestre ainsi que les objectifs à remplir. Ceux-ci s'articulent autour de deux axes principaux.

Le premier objectif était de rendre l'application **user-friendly**, c'est-à-dire plus robuste et plus facile à utiliser. Celle-ci devrait pouvoir anticiper les erreurs de distraction et le manque de données en avertissant l'utilisateur. De plus, les corrections devront être réalisées de manière à ce que chaque option présente dans l'application fonctionne correctement, sans anomalie ni erreur d'affichage.

Le second d'objectif était d'assurer la stabilité et la précision des résultats obtenus. Les variations entre les résultats obtenus, causées par l'utilisateur, devraient donc être négligeables.

Troisième partie

Nouveaux développements

Chapitre 7

Corrections des problèmes détectés

1 Correction du range

Comme expliqué précédemment, le range ne semblait pas être correctement implémenté au niveau de l'analyse PIV et ne pouvait pas être utilisé pour tous les cas. Dans la situation où l'utilisateur choisissait de réaliser une calibration de la caméra ou dans le cas où un filtre était appliqué, l'analyse PIV ne se lançait pas. Un travail d'analyse a donc été réalisé dans ces deux étapes pour déterminer la source de l'erreur.

Le point commun de ces deux étapes est que celles-ci modifient la vidéo seulement sur le range choisi par l'utilisateur. Une fois que ces étapes ont été réalisées, la vidéo modifiée est réenregistrée sur la durée du range sous un nouveau nom.

Hors, lors de l'analyse PIV, seules les images comprises dans le range initial sont analysées. Cependant, comme la durée de la vidéo n'est plus égale à la durée initiale mais vaut maintenant la durée du range, un problème apparaît.

Pour illustrer plus concrètement ce problème, un exemple issu du jeu 1 est détaillé ci-dessous. Celle-ci a une durée de 44 secondes. Comme cette vidéo comporte des particules ajoutées entre 30 et 40 secondes, l'utilisateur peut choisir d'analyser la vidéo sur le range de 36 à 38 secondes. S'il décide ensuite d'appliquer un filtre sur la vidéo appelée initialement *Video1*, celle-ci sera renommée *Video1Filtered* et durera seulement 2 secondes puisque le range choisi va de 36 à 38 secondes. Cependant, l'analyse PIV est prévue pour débiter à 36 secondes alors que la vidéo a désormais une durée de 2 secondes. Dans l'impossibilité de trouver l'indice de début, l'analyse PIV ne pouvait donc pas se lancer.

Une correction de l'indice de début d'analyse PIV a donc été réalisée pour éviter que ce problème ne se reproduise dans le cas où une calibration ou l'application de filtre est choisie.

2 Correction de l'analyse PIV

2.1 Influence du step

Lors du travail d'analyse sur le set de vidéos reçu, une erreur concernant l'implémentation du step avait été pointée. En effet, une augmentation anormale de l'ordre de grandeur des débits obtenus se produisait lorsque le step était modifié (voir figure 6.1).

L'erreur se situait au niveau du calcul du Δt , c'est-à-dire de l'intervalle de temps compris entre deux images successives. Il avait été calculé de cette manière :

$$\Delta t = \frac{1}{fps} \quad (7.1)$$

avec *fps* le nombre d'images par seconde.

L'équation 7.1 ne tient alors pas compte du *step* éventuellement choisi par l'utilisateur et influence donc anormalement le résultat. L'implémentation de cet intervalle a donc été corrigée pour tenir compte du *step*.

$$\Delta t = \frac{1}{fps} \cdot step \quad (7.2)$$

2.2 Analyse des fonctions d'OpenPIV

Le travail d'analyse mené sur les vidéos reçues a montré une variabilité importante et manque de fiabilité dans les résultats obtenus après l'analyse PIV malgré le fait que les tests aient été réalisés dans les mêmes conditions. De plus, sur certaines vidéos, les résultats de l'analyse de vitesse présentaient des zones vides sans vitesse (voir figures 6.2 et 6.3). Or, comme expliqué dans l'état de l'art, plusieurs fonctions de la librairie d'OpenPIV sont utilisées pour réaliser l'analyse PIV. Ces fonctions peuvent être comparées à des boîtes noires prenant en entrée certains paramètres. Elles ont donc été analysées en détail afin de comprendre le fonctionnement de chacune ainsi que l'influence de la modification de leurs paramètres.

Après avoir calculé les vitesses selon les deux axes *u* et *v*, une validation de ceux-ci est réalisée. Si les valeurs ne sont pas validées, celles-ci sont considérées comme des valeurs aberrantes et sont assignées à la valeur *NaN*. Elles doivent donc être remplacées. La fonction `replace_outliers()` de la librairie OpenPIV [48] intervient alors. Celle-ci sera détaillée ci-dessous car la modification d'un de ses paramètres a permis de résoudre les zones sans vitesse. Cette fonction prend en entrée les arguments suivants :

- *u* et *v*, les vecteurs vitesse issus de la validation
- `method="localmean"`, la méthode utilisée pour remplacer les valeurs aberrantes
- `max_iter`, le nombre d'itérations
- `kernel_size`, la taille du kernel

Cette fonction fait appel à un algorithme itératif d'implantation d'image. Le principe de la méthode *localmean* est de remplacer les valeurs extrêmes par une moyenne pondérée des valeurs des vitesses aux alentours si celles-ci sont valides. Pour obtenir le poids *p* de chaque vecteur alentour dans la moyenne, la méthode *localmean* fait appel à la taille du kernel (voir équation 7.3).

$$p = \frac{1}{(2 \cdot kernel_size + 1)^2 - 1} \quad (7.3)$$

Si les éléments voisins ne sont eux-mêmes pas validés, il faudra plusieurs itérations. La première permettra de corriger les vecteurs voisins. Les suivantes répartiront l'information en partant du bord jusqu'au centre. Cette fonction retourne les composantes *u* et *v* des vecteurs vitesses corrigés (c'est-à-dire avec les valeurs extrêmes remplacées) sous forme d'un tableau à deux dimensions.

En modifiant la taille du kernel, initialement égale à 2, vers 3, les vecteurs vitesses qui présentaient auparavant systématiquement des zones vides, apparaissaient correctement (voir figure 7.1). Cela peut s'expliquer par le fait que si le kernel n'est pas assez grand, les itérations ne vont pas jusqu'au point souhaité et des zones sans vitesse apparaissent alors.

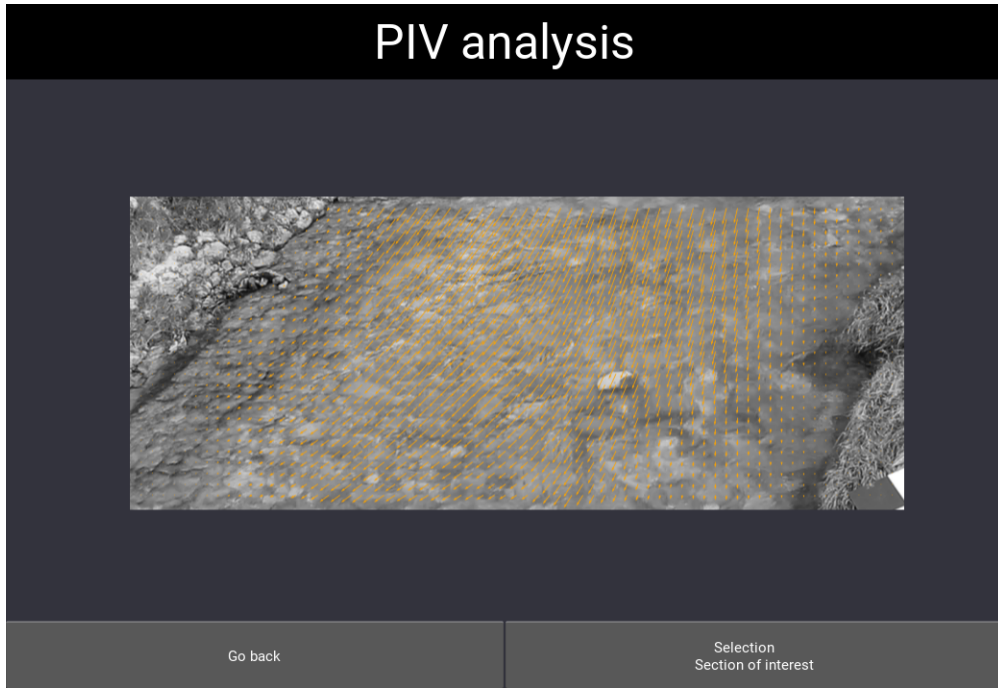


FIGURE 7.1 – Vecteurs vitesses calculés et affichés correctement

3 Correction de la méthode Angle et Hauteur

Comme expliqué dans la section 2.3 du chapitre 3, des erreurs dans les formules utilisées pour le calcul de l'angle α ont été détectées. Les équations utilisées ensuite, permettant la détermination de la hauteur H en mètres et de la largeur de la rivière L en mètres sont correctes.

3.1 Cas du drone centré par rapport à la rivière

La configuration du drone centré est illustrée à la figure 7.2. Dans ce cas, la formule implémentée par Simon Renard était simplifiée en faisant une approximation de degré 1 de son développement en série de Taylor, ce qui introduisait la condition $|\frac{L}{L_{tot}} \tan(\frac{FOV}{2})| \leq 1$.

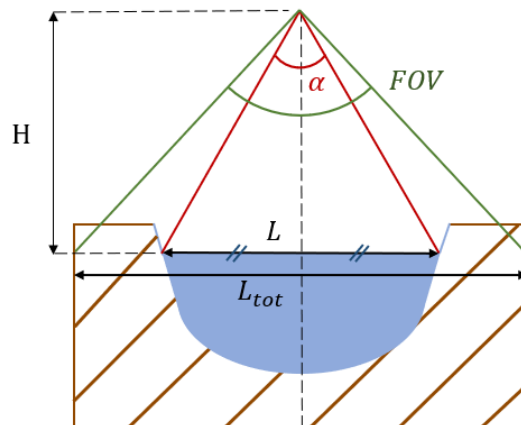


FIGURE 7.2 – Cas du drone centré

En réalité, il n'est pas beaucoup plus compliqué de déterminer l'angle α sans restriction sur

l'angle de vue. Par la trigonométrie, le système suivant peut être obtenu où les seules inconnues sont la hauteur H entre le plan d'eau et la caméra et l'angle α . Les longueurs L et L_{tot} sont respectivement les longueurs en pixels de la rivière et de l'image.

$$\begin{cases} \tan\left(\frac{FOV}{2}\right) = \frac{L_{tot}/2}{H} \\ \tan\left(\frac{\alpha}{2}\right) = \frac{L/2}{H} \end{cases} \quad (7.4)$$

$$\Rightarrow \alpha = 2 \cdot \arctan\left(\frac{L}{L_{tot}} \cdot \tan\left(\frac{FOV}{2}\right)\right) \quad (7.5)$$

Cette équation sera utilisée de la même manière pour déterminer les deux angles α_1 et α_2 (voir figure 3.25).

3.2 Cas du drone décalé par rapport au centre de la rivière

La configuration du drone décalé d'un écart ΔX par rapport au centre de la rivière est illustrée à la figure 7.3.

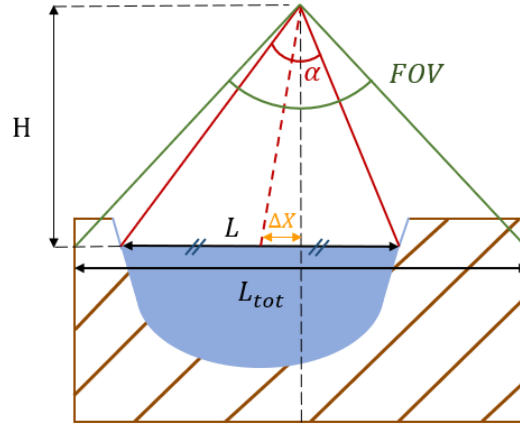


FIGURE 7.3 – Cas du drone décalé par rapport au centre de la rivière

Dans ce cas, la formule utilisée pour le calcul de l'angle α était incorrecte. Les corrections effectuées sont basées sur l'équation 7.6 déterminée à partir de la trigonométrie. Les inconnues sont, à nouveau, la hauteur H et l'angle α .

$$\begin{cases} \tan\left(\frac{FOV}{2}\right) = \frac{L_{tot}/2}{H} \\ \tan\left(\frac{\alpha_1}{2}\right) = \frac{L/2 - \Delta X}{H} \\ \tan\left(\frac{\alpha_2}{2}\right) = \frac{L/2 + \Delta X}{H} \end{cases} \quad (7.6)$$

$$\Rightarrow \alpha = \arctan\left(2 \cdot \frac{L/2 - \Delta X}{L_{tot}} \tan\left(\frac{FOV}{2}\right)\right) + \arctan\left(\frac{L/2 + \Delta X}{L_{tot}} \tan\left(\frac{FOV}{2}\right)\right) \quad (7.7)$$

Les deux angles α_1 et α_2 de la configuration de la figure 3.26 sont désormais calculés à partir de cette formule.

4 Correction de l'application du masque

Malgré la correction des zones sans vitesse créées par un mauvais choix de paramètre dans la fonction `replace_outliers()` au niveau de l'analyse PIV, il restait dans la plupart des vidéos deux zones sans vitesse, en particulier pour les vidéos nécessitant une orthorectification. Alors qu'une erreur au niveau de l'application du masque était fortement envisageable, détecter celle-ci a été fastidieux. Il s'est finalement avéré que l'erreur provenait de la fonction `computeMeanVelocity()`. Cette fonction permet de calculer les vecteurs vitesses moyens pour chaque noeud de la grille d'interrogation. Un mauvais changement de système de coordonnées entraînait une symétrie orthogonale du masque par rapport à l'horizontale. C'est l'application de ce masque retourné qui supprimait les vecteurs vitesses dans les coins inférieurs (voir figure 7.4).

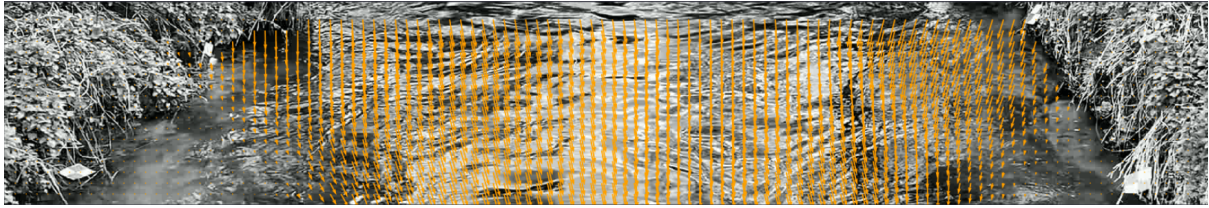


FIGURE 7.4 – Erreur causée par l'implémentation du masque

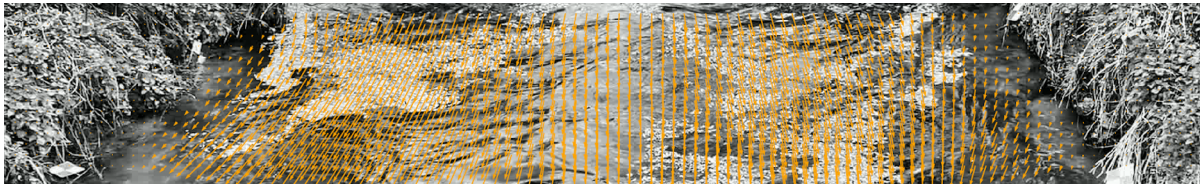


FIGURE 7.5 – Résultat après correction

5 Correction du facteur d'échelle

Pour rappel, le facteur d'échelle permet la conversion entre l'unité *pixel* et l'unité *mètre* et est calculé différemment si une orthorectification a été réalisé ou non.

$$Scaling\ Factor = \frac{L_{pixel}}{L_{mètre}} \quad (7.8)$$

5.1 Avec orthorectification

Dans le cas d'une orthorectification, nous avons expliqué, à la section 1.3 du chapitre 3, que le facteur d'échelle pouvait être choisi "quasiment" arbitrairement. La seule contrainte concerne l'affichage de l'image orthorectifiée dans l'application. Si le facteur d'échelle est trop petit, l'image corrigée risque d'être trop petite. En effet, si nous considérons le cas extrême d'un facteur d'échelle égal à 1, les quatre points de références écartés de quelques mètres dans le monde réel seront représentés écartés de quelques pixels dans le plan de l'image. De la même façon, si le facteur d'échelle est trop grand, l'image ne pourra pas être entièrement affichée.

Ce facteur était initialement fixé à 30 dans le code. Pour que sa valeur soit adaptée pour des géométries de différents ordres de grandeur, nous avons décidé de le calculer comme la moyenne des rapports des six distances en pixels et des six distances réelles entre les quatre points de références.

5.2 Sans orthorectification

Le principe de l'équation 7.8 est également utilisé dans ce cas-ci. A l'étape *Section of interest*, lorsque l'utilisateur choisit une section sur laquelle le débit sera calculé, il sélectionne deux points (un sur chaque rive) définissant la largeur de la rivière en pixels. Pour obtenir la largeur en mètres, une fonction `findLength()` a été définie. Elle prend comme arguments le fichier contenant la bathymétrie ainsi que le niveau d'eau. Ce dernier est connu par l'utilisation d'une des trois méthodes *Set Value*, *Adapted Hellebaut* ou *Angle and Height*. La fonction `findLength()` va détecter, dans la bathymétrie, la largeur correspondant au niveau d'eau.

Enfin, le facteur d'échelle est calculé via un simple rapport entre la longueur en pixels et la longueur en mètres. Bien entendu, ce facteur va varier selon l'endroit où l'utilisateur placera les points.

Chapitre 8

Améliorations

1 Développement d'un aspect "user-friendly"

Lors du premier volet de notre travail, nous avons remarqué que l'application était très peu robuste. Cela signifie qu'elle est assez sensible aux erreurs couramment effectuées par un utilisateur lorsqu'il utilise l'application trop vite ou qu'il appuie accidentellement sur un bouton avant d'avoir enregistré les paramètres nécessaires. Dans ce cas, avant modifications, l'application tournait en boucle ou s'arrêtait brusquement.

Notre but a été de rendre l'application plus facile d'utilisation et donc de développer un aspect "user-friendly". Pour cela, nous avons anticipé les erreurs qu'un utilisateur pressé ou non connaisseur pourrait aisément commettre.

Chargement des différents fichiers

Lors de plusieurs étapes, l'utilisateur peut facilement faire l'erreur de cliquer sur "Load" sans avoir sélectionné de fichier. Si tel est le cas, une fenêtre s'ouvre pour en avertir l'utilisateur et lui permettre de sélectionner à nouveau un fichier (voir figure 8.1). Cette fenêtre est utilisée lors du chargement du fichier contenant la bathymétrie, la vidéo de l'écoulement, les paramètres de la calibration, la vidéo du damier pour la calibration et le DEM.

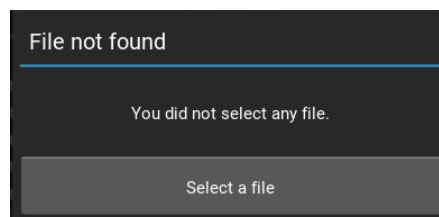


FIGURE 8.1 – Popup qui apparaît si aucun fichier n'est sélectionné

Chargement des différentes valeurs

De même, tout au long du processus pour trouver le débit, il est demandé à l'utilisateur de fournir certains paramètres ou valeurs. Sans ceux-ci, le programme ne peut pas fonctionner. Une fenêtre s'ouvre donc pour signaler à l'utilisateur qu'il doit fournir le(s) paramètre(s) requis (voir figure 8.2).

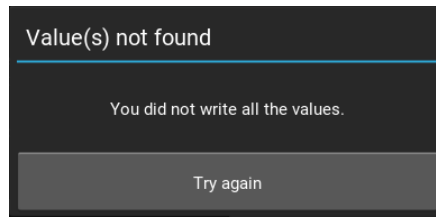


FIGURE 8.2 – Popup qui apparaît si aucune valeur n'est fournie

Orthorectification

Lors de l'orthorectification, il est nécessaire de sélectionner quatre points, de fournir les distances entre ceux-ci ainsi que de sélectionner la région de la vidéo où l'orthorectification sera utilisée. Si ces trois paramètres ne sont pas fournis, une fenêtre apparaît et donne deux possibilités : sélectionner correctement les paramètres ou passer cette étape (voir figure 8.3).

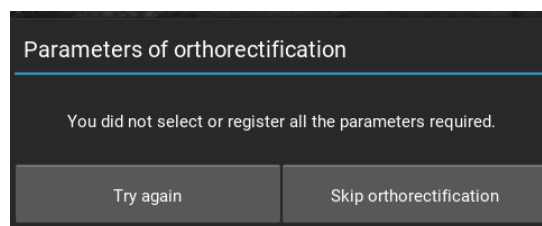


FIGURE 8.3 – Popup qui apparaît si tous les paramètres pour l'orthorectification ne sont pas correctement sélectionnés

Méthode Image

Pour utiliser cette méthode, il est obligatoire de réaliser, au préalable, l'orthorectification. Une fenêtre a donc été ajoutée pour prévenir l'utilisateur et lui laisser l'opportunité de retourner à l'étape "Orthorectification" (voir figure 8.4).

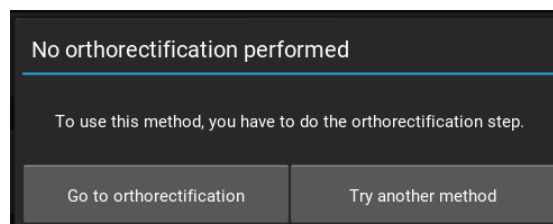


FIGURE 8.4 – Popup qui apparaît si l'utilisateur choisit la méthode "Image" sans avoir réalisé d'orthorectification

De plus, cette méthode a pour objectif de retrouver le niveau d'eau à partir de la bathymétrie de la rivière et de la largeur. Cette largeur est déduite grâce à deux points choisis par l'utilisateur. Pour cela, une fonction `findWaterLevel()` a été définie. Le problème était que si cette fonction ne détectait pas de largeur correspondant à la distance entre les deux points dans la section, le programme ne proposait aucune alternative. Une nouvelle fenêtre a donc été créée pour avertir l'utilisateur qu'aucune largeur correcte n'a été détectée dans la section et lui donner la possibilité de choisir de nouveaux points (voir figure 8.5).

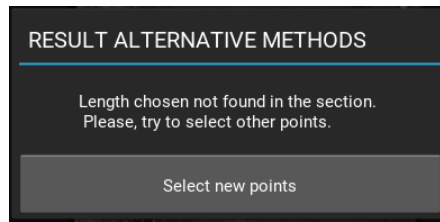


FIGURE 8.5 – Popup qui apparaît si la largeur choisie n'est pas détectée dans la section

Méthode Angle et Hauteur

A nouveau, pour utiliser cette méthode, il est nécessaire de réaliser une calibration. Dans le cas où celle-ci n'a pas été faite, une fenêtre apparaît pour en informer l'utilisateur et lui donner l'opportunité d'en réaliser une (voir figure 8.6).

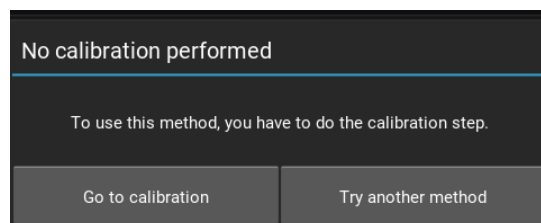


FIGURE 8.6 – Popup qui apparaît si l'utilisateur choisit la methode "Angle et hauteur" sans avoir réalisé de calibration

De même, si la largeur trouvée grâce à cette méthode n'est pas détectée dans la section, la fenêtre de la figure 8.5 apparaît pour donner à l'utilisateur la possibilité de sélectionner une nouvelle largeur.

Application des filtres

Une fenêtre a été ajoutée lors de l'application des filtres. Si l'utilisateur choisit d'ajouter un filtre sans l'avoir sélectionné au préalable, la fenêtre de la figure 8.7 apparaît.

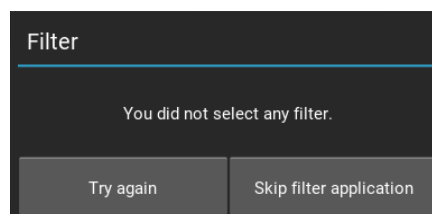


FIGURE 8.7 – Fenêtre avertissant que le filtre n'a pas été sélectionné

De plus, lors de l'application de certains filtres, si ceux-ci ne sont pas adaptés à la vidéo, l'effet n'est pas toujours bien visible. Une fenêtre a donc été ajoutée pour confirmer à l'utilisateur que le filtre choisi a été correctement appliqué (voir figure 8.8).

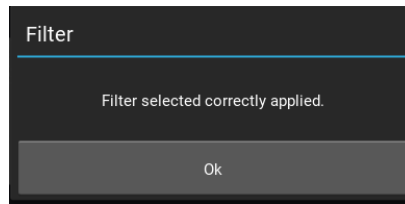


FIGURE 8.8 – Fenêtre confirmant l'application du filtre

"Close", "Back" et "Starting screen"

Pour permettre à l'utilisateur de fermer correctement l'application, un bouton "Close" a été ajouté sur la dernière fenêtre.

Une vérification du bouton "Starting Screen" (qui permet de réaliser à nouveau le cycle pour trouver le débit) ainsi que des boutons "Back" (qui permettent, à chaque étape, de revenir à l'étape précédente) a également été réalisée. Désormais, lorsque l'utilisateur choisit de revenir à l'étape précédente, les images s'effacent correctement, permettant de réaliser de nouveau l'étape souhaitée.

2 Fichiers en local

Dans le chapitre 6 sur les pistes d'amélioration, nous avons eu l'idée de mettre les fichiers d'OpenPIV en local (voir section 2.2 du chapitre 6). Les fichiers *pyprocess.py*, *windef.py* et *tools.py* ont donc été enregistrés dans le dossier "Application" et des modifications du fichier *packagesFile.py* ont également été nécessaires.

3 Nouvelles options

3.1 Méthode de calibration

La méthode de calibration de la caméra implémentée dans l'application est basée sur une vidéo d'un damier qui permet d'obtenir les paramètres de la caméra. Or, cette vidéo ne peut être obtenue si l'accès à la caméra n'est pas possible. Il est donc intéressant de permettre à l'utilisateur d'encoder les paramètres de la caméra (tels que la distance focale et les coordonnées du centre optique) afin que la calibration puisse être réalisée (voir figure 8.9).

Calibration parameters	
Name of the file that will contain calibration parameters:	e.g. : calibration_parameters
Coefficients of distortion :	e.g. : 4,6,8,12
c_x (optical center):	e.g. : 0
c_y (optical center):	e.g. : 0
Horizontal focal distance :	e.g. : 200
Vertical focal distance :	e.g. : 200
Diagonal focal distance :	e.g. : 200
Go back Load	

FIGURE 8.9 – Fenêtre d'encodage des paramètres de calibration

3.2 Méthode pour déterminer le niveau d'eau

Comme expliqué dans la section 2.2 du chapitre 3, les méthodes *Image* et *Angle et Hauteur* alternatives sont basées sur la détermination de la largeur et sur la recherche de cette largeur dans la section. La méthode ajoutée demande à l'utilisateur de fournir la largeur en mètres (voir figure 8.10). Cette largeur est ensuite détectée dans la bathymétrie de la même façon que les méthodes alternatives.

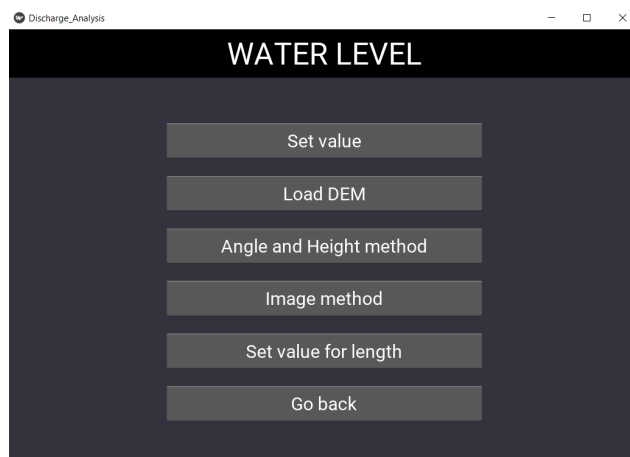


FIGURE 8.10 – Méthode ajoutée pour encodé la largeur d'eau

Cette méthode a l'avantage de fournir une largeur plus précise que celle calculée par la méthode *Image* qui est soumise à l'erreur de l'utilisateur. De plus, fournir la largeur a un avantage par rapport à la méthode *Set value* où l'utilisateur fournit le niveau d'eau. En effet, il est plus simple de mesurer une largeur que le niveau d'eau.

3.3 Choix des filtres

Étant donné que le choix entre les filtres proposés n'est pas forcément aisé, une aide à l'utilisateur a été ajoutée. Celle-ci consiste en un bouton *information* ajouté et permettant d'obtenir des informations additionnelles sur le filtre choisi (voir figure 8.11). Les informations relatives au filtre comprennent la façon dont le filtre agit sur l'intensité des pixels ainsi que les défauts types d'une vidéo que ce filtre permet de corriger.

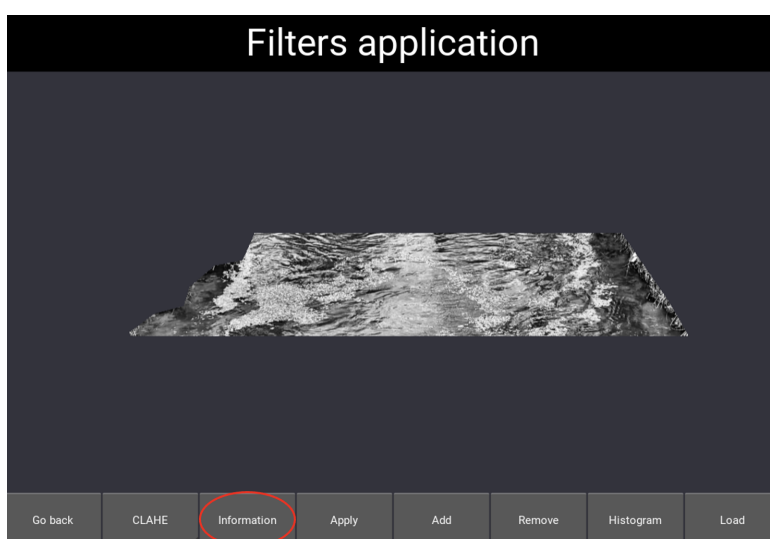


FIGURE 8.11 – Aide à l'utilisateur pour le choix d'un filtre

Sur base de nombreux tests, nous avons pu déterminer quels filtres permettent d'obtenir les meilleurs résultats pour chaque vidéo disponible. La figure 8.12 reprend la synthèse des tests effectués. Nous avons ensuite tenté de déterminer les situations adaptées à chaque filtre. Ce sont ces informations qui ont été ajoutées dans l'application.

	Sans Filtre	CLAHE	Intensity capping	Intensity high-pass	mean denoising	Gamma	Contrast/brightness
terrain_limelette_video1_camera1.MOV		X					
Limelette_drone_coupe.mp4		X		X			
terrain_noirhat_video1_camera1.MOV		X					
terrain_noirhat_video1_camera2.MOV						X	
Jeu1	X						
Jeu7		X				X	
terrain_rosière_video1_coupe		X					

FIGURE 8.12 – Filtres les plus adaptés à chaque vidéo

Clahe

Comme le filtre *CLAHE* agit sur l'histogramme d'intensité des pixels, celui-ci peut être affiché afin d'indiquer à l'utilisateur si ce filtre doit être utilisé (voir figure 8.13). Si l'histogramme d'intensités est confiné dans une région réduite, le filtre va étendre celui-ci afin de que le contraste entre les particules et le fond de la rivière soit accentué.



FIGURE 8.13 – Histogramme des intensités des pixels avant et après l'application du filtre CLAHE

Nous remarquons que l'histogramme, après l'application du filtre CLAHE, est effectivement plus étendu.

Intensity Capping

Le filtre *intensity capping* aura pour effet de réduire les pics d'intensité. Il devrait donc être utilisé sur des vidéos surexposées. Il y aura alors un effet d'uniformisation des pics d'intensité. Il est difficile d'estimer de façon expérimentale les caractéristiques de vidéo pour lesquelles ce filtre serait le plus adapté. En effet, aucune des vidéos à notre disposition ne correspond à son utilisation.

Par contre, il est important de noter que le risque d'une perte de contraste est élevé avec ce filtre.

Intensity high-pass

Suite aux tests, nous avons observé que le filtre *intensity high-pass* était relativement bien adapté à la vidéo filmée à Limelette, à l'aide d'un drone. Nous observons sur les figures 8.14 et 8.15 l'influence du filtre, en particulier dans les zones 1 et 2.

- Dans la zone 1, en rouge, les particules ne sont pas présentes mais le filtre permet de toute de même suivre l'eau correctement.

- Dans la zone 2, en bleue, les particules sont plus contrastées et pourront donc être mieux suivies.

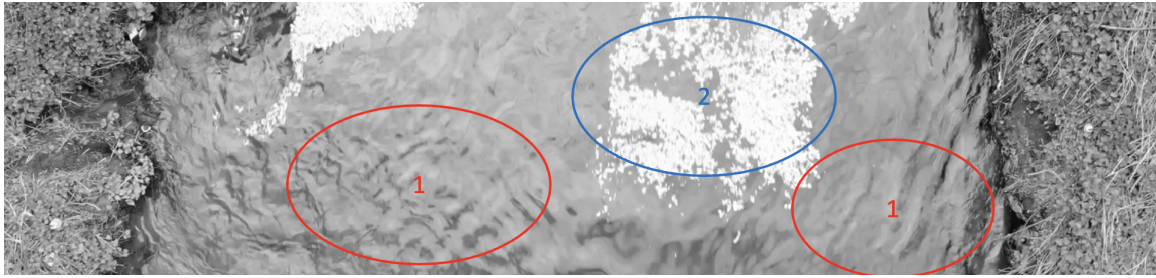


FIGURE 8.14 – Limelette avant l'application du filtre High-pass

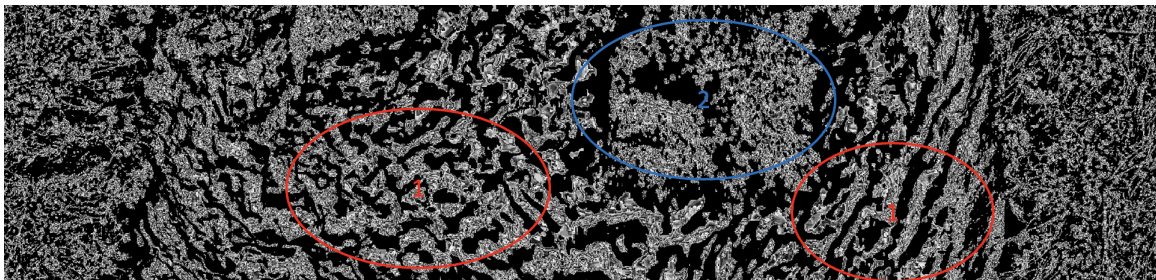


FIGURE 8.15 – Limelette après l'application du filtre High-pass (facteur d'amplification égale à 200)

Il peut donc être intéressant d'utiliser ce filtre lorsque les traceurs ne sont pas répartis de manière uniforme et que les parties sans traceurs forment une surface d'eau opaque. Cette dernière condition est importante. En effet, si l'eau est transparente, comme sur la vidéo du jeu 1 (voir figure 8.16), le filtre ne fera pas la distinction entre le fond de la rivière et la surface d'eau. Ainsi, après avoir appliqué le filtre sur le jeu 1, un grand nombre de particules visibles sur la figure 8.17 représentent en réalité des pierres du fond.



FIGURE 8.16 – Jeu1 avant l'application du filtre High-pass

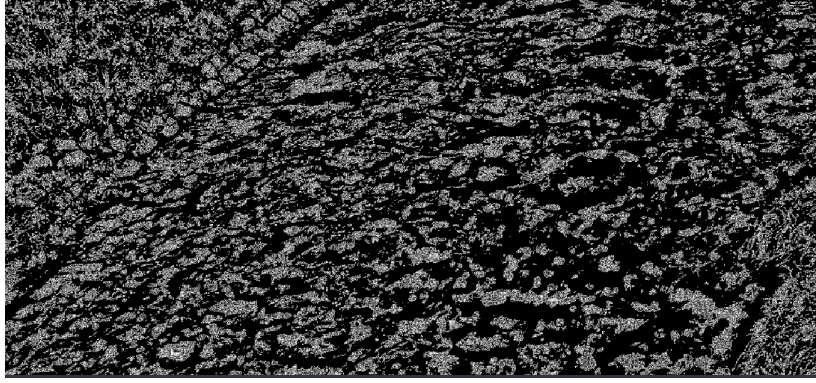


FIGURE 8.17 – Jeu1 après l’application du filtre High-pass (facteur d’amplification égal à 200)

Mean denoising

Le filtre Mean Denoising n’a pas été testé car il ne permet pas de corriger le contraste de l’image. En effet, le filtre a pour but de corriger les parasites et le bruit virtuel parfois présent dans certaines vidéos. Or, dans le set de vidéos dont nous disposons, aucune d’entre elles ne présentait ce type de défaut. En outre, l’application de ce filtre prend beaucoup de temps.

Gamma

Le filtre *gamma* fonctionne particulièrement bien pour la deuxième vidéo filmée à Noirhat et pour le jeu 7. En comparant ces deux vidéos, nous pouvons conclure que ce filtre est utile dans le cas d’une vidéo plutôt claire et présentant un minimum de contraste. L’application de celui-ci va alors augmenter le contraste des particules ou des remous à la surface de l’eau, comme par exemple dans le jeu 7.

Par contre, il faut éviter de l’utiliser pour des vidéos présentant des trous dans la répartition des particules, comme par exemple à Rosières, représenté aux figures 8.18 et 8.19. Ces trous sans contraste vont être assombris et donc l’erreur créée sur la vitesse de l’écoulement à ces endroits va augmenter.

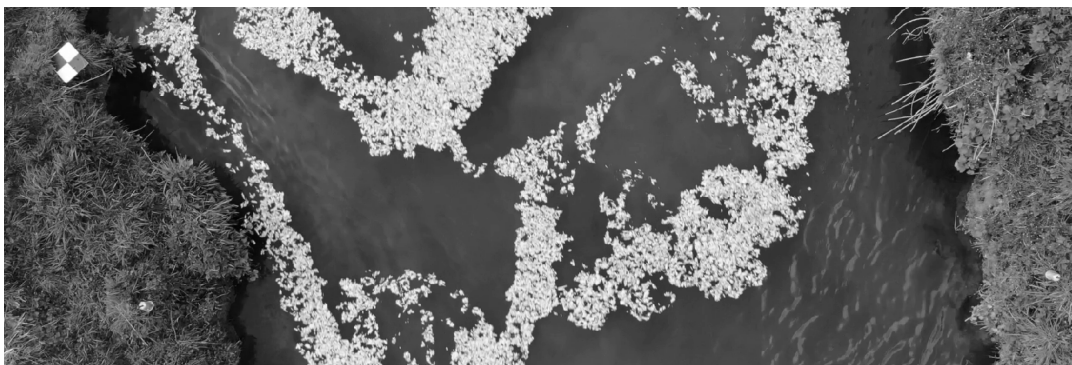


FIGURE 8.18 – Rosières avant l’application du filtre gamma

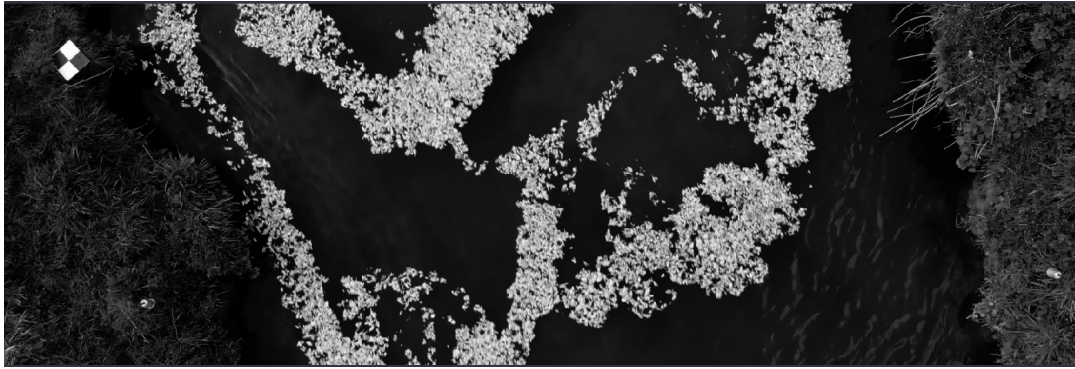


FIGURE 8.19 – Rosières après l'application du filtre gamma (paramètre égal à 2)

Contrast/brightness

Le filtre contrast-brightness possède deux paramètres différents :

- α qui influence le contraste. Si α augmente, le contraste augmente et inversement.
- β qui influence la luminosité. Si β augmente, la luminosité augmente.

Si α et β sont tous deux égaux à 1, l'image n'est pas modifiée. Si seul le paramètre α est modifié, le filtre contrast/brightness équivaut à un filtre gamma. Modifier les deux paramètres pour obtenir un contraste suffisant est complexe. Nous recommandons donc plus plutôt d'utiliser le filtre gamma.

Sur les vidéos testées, il n'y avait pas de résultat obtenu avec ce filtre qui se démarquait spécifiquement. Les vidéos dont nous disposons n'étaient donc pas suffisantes pour pouvoir observer l'effet exact du filtre.

3.4 Sauvegarde des résultats dans fichier texte

Lors de notre travail concernant l'analyse PIV, il a été utile de connaître les vitesses de surfaces calculées à cette étape. Une impression de ces données dans un fichier texte, appelé *Velocity.txt* a été implémentée permettant à l'utilisateur, s'il le souhaite, de connaître ces données de vitesse. Ce fichier peut être exporté en Excel.

Les données générales telles que les vitesses moyennes u et v , le niveau d'eau, le coefficient α utilisé ainsi que le débit calculé après une analyse sont également sauvegardées dans un fichier texte appelé *analysisResults.txt* qui se placera dans le dossier "Application".

3.5 Version sous Graphical Processing Unit

La version d'OpenPIV implémentée actuellement fonctionne sous CPU¹ (Central Processing Unit). Cette version de l'analyse de vitesse peut prendre jusqu'à quarante minutes pour des vidéos d'une durée de plus de trente secondes sans step. Réduire le temps mis par cette étape pourrait donc constituer un avantage si l'utilisateur a la volonté d'analyser une vidéo complète.

Il existe une version d'OpenPIV sous GPU²(Graphical Processing Unit). Lorsque celle-ci est implémentée, le temps de calcul mis par l'analyse PIV peut être réduit. Cependant, cette version nécessite l'installation de plusieurs packages : PyCuda et ScikitCuda. Pour installer PyCuda, il faut une carte graphique NVIDIA, ce qui n'était le cas pour aucun de nos deux ordinateurs. Des travaux futurs pourraient donc investir cette piste pour accélérer l'analyse PIV. Il est à noter que la version d'OpenPIV sous GPU est une version pré-beta et peut donc présenter des erreurs

1. Le CPU correspond au processeur principal de l'ordinateur.

2. Le GPU correspond au processeur utilisé pour le calcul graphique. Ici, on fait référence à un GPU additionnel.

et des changements réguliers [53]. En attendant que cette version puisse être implémentée dans l'application, le step et le range sont de bonnes alternatives pour accélérer l'analyse PIV.

3.6 Multi-pass

Principe d'une analyse PIV multi-pass

Comme expliqué dans la section 3 du chapitre 3, l'analyse PIV implémentée initialement est une analyse "single pass" utilisant une transformée de Fourier. L'inconvénient de cette méthode est que la taille de l'aire de recherche est fixe. Si celle-ci est trop grande, il y a aura un risque de perte de précision mais si elle est trop petite il y aura cette fois un risque de perdre d'informations.

Il devient donc intéressant, pour obtenir des résultats de vitesses plus précis, de faire intervenir une analyse multi-pass. L'avantage de cette analyse est que la taille de l'aire de recherche est adaptée au fur et à mesure des passages. Au premier passage, l'aire de recherche est supposée relativement grande (voir figure 8.20). Les résultats obtenus sont utilisés pour décaler la position de l'aire de recherche lors des passages suivants et limiter ainsi les pertes d'informations [16]. La taille de l'aire de recherche est diminuée au second passage (voir figure 8.21) et les vecteurs vitesses du premier passage sont interpolés sur l'aire plus petite du second passage. Ce sont donc ces vecteurs interpolés qui participent à la corrélation du deuxième passage [54]. Les passages successifs suivants se déroulent de la même manière. Entre chaque passage, les données sont lissées et les données manquantes sont interpolées. Les vecteurs vitesses seront donc raffinés jusqu'à obtenir une résolution spatiale maximale. Il est important de remarquer que le nombre de passages doit être choisi comme un juste milieu entre un temps de calcul raisonnable et la précision recherchée.

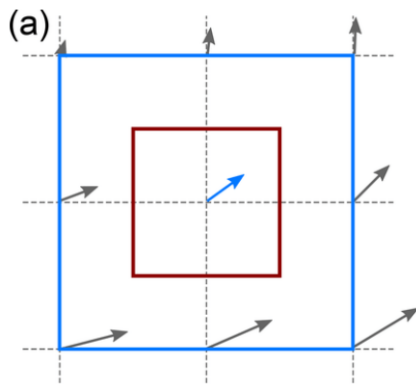


FIGURE 8.20 – Premier passage : aire de recherche (en bleu) avec un chevauchement de 50 % et vecteurs vitesses calculés sur les bords et au centre [54].

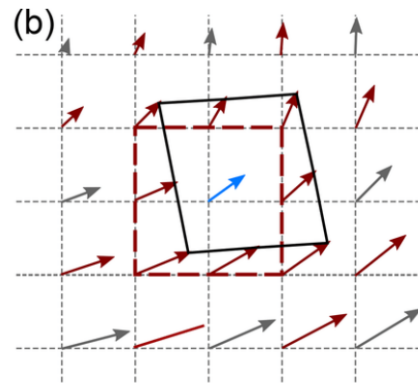


FIGURE 8.21 – Second passage : la taille de l'aire de recherche (en rouge) a été réduite et les vecteurs issus du premier passage (gris et bleus) ont été interpolés [54].

L'analyse multi-pass tient également compte de la potentielle déformation de l'aire de recherche (voir figure 8.22). Les déplacements issus du premier passage sont connus sur neuf points : un point central et huit points sur les bords, calculés à partir du chevauchement des aires d'interrogation. Ces données permettent de connaître les déplacements sur chaque pixel de l'aire d'interrogation via une interpolation bilinéaire. L'aire d'interrogation est ensuite déformée selon les déplacements du premier passage. La première aire d'interrogation et la seconde aire déformée sont alors corrélées [16].

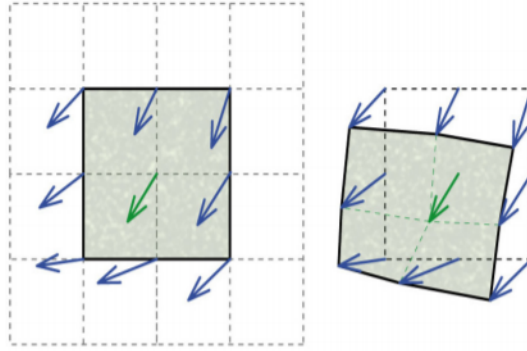


FIGURE 8.22 – Déformation de l'aire de recherche [16]

Analyse multi-pass dans l'application

Une analyse PIV multi-pass a été développée dans la librairie OpenPIV. Au cours de notre travail, nous avons tenté de l'implémenter dans l'application afin d'offrir à l'utilisateur deux types d'analyse PIV (voir figure 8.23).

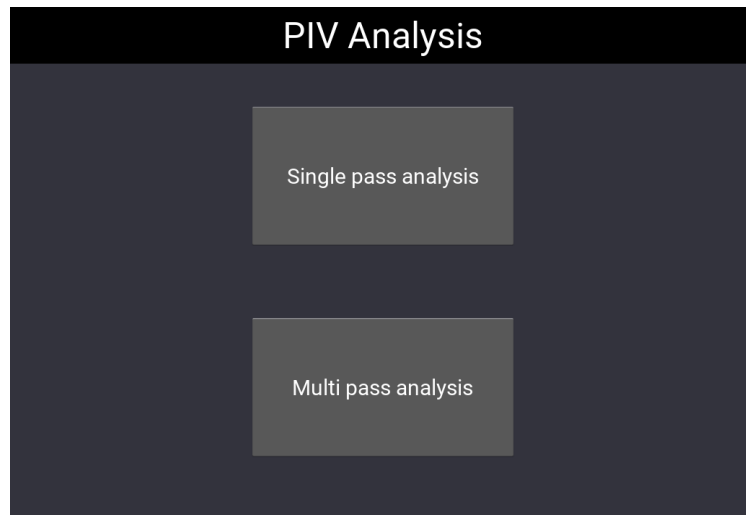


FIGURE 8.23 – Choix du type d'analyse PIV dans l'application

L'analyse multi-pass que nous avons implémentée comporte actuellement trois passages. Le premier passage est calculé sur base d'une aire de recherche de 64×64 pixels. Ensuite, lors du second et du troisième passage, l'aire de recherche est réduite à 32×32 pixels puis à 16×16 pixels.

Cependant, après une série de tests réalisés sur une même vidéo dans des conditions semblables, nous avons constaté que l'analyse multi-pass implémentée présente des résultats de débit très divergents (voir figure 8.24). Malgré une étude détaillée du code et un grand nombre de modifications réalisées, nous n'avons pas pu identifier la cause de ce manque de régularité des résultats. De plus, la documentation sur les fonctions de l'analyse multi-pass d'Openpiv est relativement rare et difficile à trouver. Cette étape pourrait donc être encore améliorée lors de futurs travaux afin de permettre à l'utilisateur d'obtenir des résultats fiables et stables via une autre option d'analyse PIV.

Set de tests Jeu 2 - Q = 107.945 m ³ /s						
Test	Stabilisation	Filtres	Range	Step	SF	Q [m ³ /s]
1	Oui	CB	0-2	3	22,15	55,0652
2	Oui	CB	0-2	3	22,06	28,865
3	Oui	CB	0-2	3	22	36,6779
4	Oui	CB	0-2	3	21,99	43,1141
5	Oui	CB	0-2	3	22,17	45,8504
6	Oui	CB	0-2	3	22,61	35,7754
7	Oui	CB	0-2	3	22,42	40,8392
8	Oui	CB	0-2	3	22,56	55,4311
9	Oui	CB	0-2	3	22,42	26,8335
10	Oui	CB	0-2	3	22,64	36,6674
1	Oui	CB	0-2	3	22,85	38,9117
2	Oui	CB	0-2	3	22,45	43,9259
3	Oui	CB	0-2	3	22,29	31,8
4	Oui	CB	0-2	3	21,96	40,074
5	Oui	CB	0-2	3	22,18	41,6787
6	Oui	CB	0-2	3	22,17	36,12
7	Oui	CB	0-2	3	22,29	45,49
8	Oui	CB	0-2	3	22,71	31,44
9	Oui	CB	0-2	3	22,26	29,73
10	Oui	CB	0-2	3	21,91	38,044

FIGURE 8.24 – Analyse multi-pass sur le jeu 2

Chapitre 9

Validation de l'application

Dans ce dernier chapitre, nous avons effectué une batterie de tests sur l'application corrigée. Le but de cette phase de validation est d'obtenir une première estimation de l'erreur produite à chaque étape. Les conclusions pourront alors être utiles pour les développements futurs.

1 Vidéos utilisées pour les tests

1.1 Cas limites du Vidéo Globe Challenge

Pour réaliser la phase de tests, toutes les vidéos issues du Vidéo Globe Challenge n'ont pas pu être utilisées. Cette section présentera les vidéos inutilisables ainsi que les limites de l'application sur celles-ci.

Jeu 3

Les données du jeu 3 ne permettent pas de réaliser une orthorectification. Cependant, si la méthode *Image* avait pu être utilisée sur le jeu 3, la sélection de la largeur aurait été compliquée. En effet, la vidéo est filmée dans un méandre et ne permet pas de choisir une largeur représentative (voir figure 9.1).



FIGURE 9.1 – Géométrie du tronçon du jeu 3

Jeu 4

La rivière du jeu 4 est filmée par un drone qui survole celle-ci et est couverte de beaucoup de végétations et d'un pont qui la surplombe (voir figure 9.2).

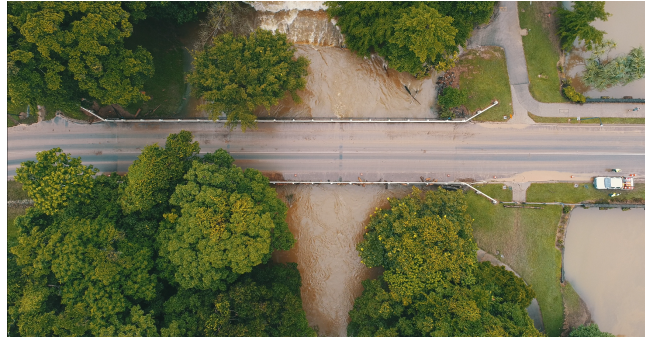


FIGURE 9.2 – Tronçon du jeu 4

Les résultats obtenus lors de l'analyse du jeu 4 étaient tous largement inférieurs au débit réel de la rivière, $Q = 139,588[\frac{m^3}{s}]$ (voir figure 9.3).

Set de tests Jeu 4 - $Q = 139.588 \text{ m}^3/\text{s}$								
Test	Stab	Filtres	Range	Step	n	alpha	SF	Q [m^3/s]
1	oui	Sans	Tout	1	20	0,85	17,11	88,1508
1.1	oui	Sans	Tout	1	10	0,85	17,11	89,0089
2	oui	Sans	Tout	2	20	0,85	17,73	67,4927
3	oui	Sans	Tout	3	20	0,85	17,55	58,4098
4	oui	CLAHE	Tout	3	20	0,85	16,83	58,4039
5	oui	CLAHE	Tout	3	20	0,91	18,64	54,3748
6	oui	CLAHE	Tout	3	20	0,91	18,62	46,379
7	oui	CB	Tout	3	20	0,91	19,33	32,1177
8	oui	CB	Tout	1	20	0,91	18,18	75,557
9	oui	IC + CLAHE	Tout	1	20	0,91	19,5	49,1457

FIGURE 9.3 – Résultats des tests sur le jeu 4

Jeu 8

La vidéo du jeu 8 présente un zoom dans les premières secondes de la vidéo (voir figures 9.4 et 9.5).

Dès lors, sur les premières images de la vidéo, les points de contrôle ne sont pas tous visibles. Il est donc impossible de les sélectionner lors de l'orthorectification, réalisée sur la première image de la vidéo.



FIGURE 9.4 – Premières secondes : tous les points de contrôle ne sont pas visibles



FIGURE 9.5 – Tous les points de contrôle sont maintenant visibles

Pour pouvoir analyser cette vidéo, les premières secondes la vidéo ont donc été coupées afin que les quatre points de contrôle soient visibles sur la première image de la vidéo.

1.2 Nouveau set de vidéos

Afin de pouvoir tester complètement toutes les fonctionnalités de l'application et de pouvoir comparer plusieurs méthodes, il était nécessaire de disposer d'un nouveau set de vidéos complet.

Les vidéos ont été réalisées sur trois sites différents : Rosières (Lasne), Limelette (Dyle) et Noirhat (Dyle) (voir figure 9.6). Ces sets de vidéos comprennent plusieurs films, pris de différentes positions, avec ou sans copeaux ajoutés. Les données de largeur, hauteur d'eau et débit réels ont été mesurées sur le terrain.

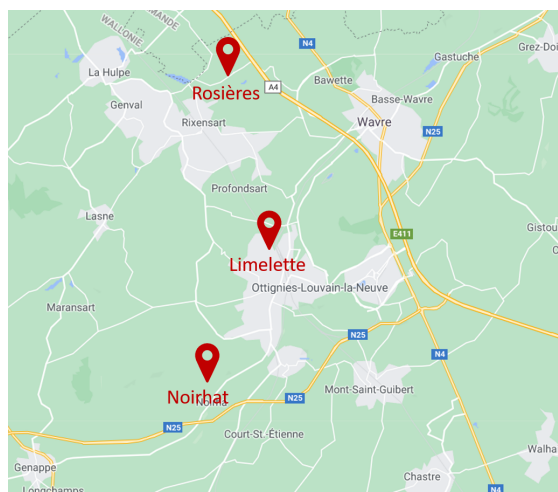


FIGURE 9.6 – Localisation des trois sites visités

Il est important de noter que ces trois rivières sont des rivières à faibles débits.

Site	Hauteur d'eau [m]	Q [m^3/s]
Limelette	0,478	1,403
Noirhat	0,155	0,203
Rosières	0,64	0,664

Noirhat

A Noirhat, trois vidéos ont été filmées selon trois angles différents. Celles-ci ont été filmées au même moment, sur une rivière où des traceurs ont été ajoutés.



FIGURE 9.7 – Vidéo filmée depuis la berge



FIGURE 9.8 – Vidéo filmée depuis le pont

Les deux vidéos présentes aux figures 9.7 et 9.8 ont été filmées depuis la berge et le pont. Elles nécessitent donc une orthorectification.



FIGURE 9.9 – Vidéo filmée perpendiculairement à la surface d'eau

La vidéo de la figure 9.9 n'a malheureusement pas pu être exploitée pour plusieurs raisons. Tout d'abord, les particules sont trop nombreuses et forment des masses qui ne permettent plus de les distinguer. De plus, la vidéo a été filmée par un drone avec une lentille qui n'était pas adaptée à la luminosité présente à ce moment ce qui amplifie le problème causé par les masses de particules. Les filtres présents dans l'application ont été testés pour tenter de corriger le problème de luminosité mais n'étaient pas suffisants pour rendre la vidéo exploitable.

Limelette

Sur le site de Limelette, deux vidéos ont été tournées, avec et sans copeaux ajoutés à la rivière. La vidéo de la figure 9.10 a été filmée du pont et nécessite une orthorectification alors que la seconde est filmée perpendiculairement au plan d'eau et ne doit donc pas être orthorectifiée.



FIGURE 9.10 – Vidéo filmée depuis le pont

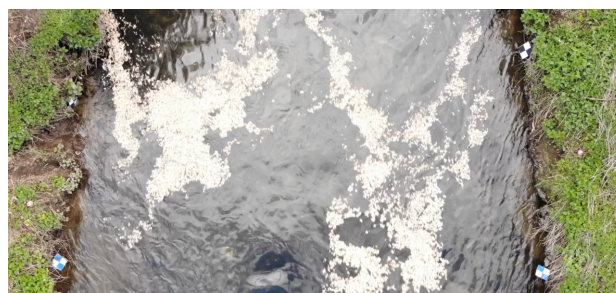


FIGURE 9.11 – Vidéo filmée perpendiculairement à la surface d'eau

Ces deux vidéos ont été utilisées pour la phase de validation. Il est à noter que sur ce site, les berges sont très raides. Il était donc impossible de filmer depuis la berge sans perdre une trop grande partie de la vue directe sur la rivière (voir figure 9.12).



FIGURE 9.12 – Berges sur le site de Limelette

Rosières

Sur le site de Rosières, deux vidéos de la rivière ont été filmées, avec et sans particules. La première vidéo est filmée depuis la berge et nécessite une orthorectification (voir figure 9.13). Cependant, comme le quatrième point de contrôle n'est pas visible et que les GCPs ne sont pas correctement placés sur le plan d'eau, l'orthorectification n'a pas pu être réalisée et cette vidéo n'a pas pu être exploitée. La seconde vidéo a été filmée perpendiculairement au plan d'eau et a pu être utilisée pour la phase de validation (voir figure 9.14).



FIGURE 9.13 – Vidéo filmée depuis la berge à Rosières



FIGURE 9.14 – Vidéo filmée perpendiculairement au plan d'eau

2 Tests de validation

2.1 Influence de l'ordinateur de l'utilisateur

Au travers de notre première série de tests, nous avons voulu nous assurer que l'ordinateur n'influençait pas les résultats obtenus.

Pour ce faire, nous avons commencé par bloquer un maximum de paramètres. Ainsi pour une même vidéo, le range, le step, la hauteur d'eau, le ROI, le masque, le filtre, le nombre de sous section et le coefficient α ont tous été fixés. Après avoir effectué les 5 premiers tests, il est apparu qu'il fallait absolument bloquer toute influence de l'utilisateur pour obtenir des débits identiques. L'orthorectification a donc ensuite également été bloquée. Les résultats des quinze tests réalisés sont disponibles aux figures 9.15 et 9.16.

n°	Nom video	Range	Step	Stab.	Calib.	Ortho.	Hauteur d'eau	ROI/mask	Filtre	Section d'intérêt	scaling factor	nbre sous-sec.	α	Débit	Débit attendu
1	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI	0,478	FIXE	CLAE	FIXE	150,5233	20	0,85	1,2233	1,403
2	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI	0,478	FIXE	CLAE	FIXE	149,9697	20	0,85	1,2284	1,403
3	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI	0,478	FIXE	CLAE	FIXE	150,6822	20	0,85	1,1479	1,403
4	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI	0,478	FIXE	CLAE	FIXE	150,3565	20	0,85	1,1768	1,403
5	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI	0,478	FIXE	CLAE	FIXE	150,6819	20	0,85	1,2202	1,403
6	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI/FIXE	0,478	FIXE	CLAE	FIXE	149,820306	20	0,85	1,2509	1,403
7	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI/FIXE	0,478	FIXE	CLAE	FIXE	149,820306	20	0,85	1,2509	1,403
8	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI/FIXE	0,478	FIXE	CLAE	FIXE	149,820306	20	0,85	1,2509	1,403
9	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI/FIXE	0,478	FIXE	CLAE	FIXE	149,820306	20	0,85	1,2509	1,403
10	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI/FIXE	0,478	FIXE	CLAE	FIXE	149,820306	20	0,85	1,2509	1,403
11	Limelette_drone_coupe.mp4	3->6	3	NON	NON	NON	0,478	FIXE	CLAE	FIXE	1,4152	30	0,85	1,4152	1,403
12	Limelette_drone_coupe.mp4	3->6	3	NON	NON	NON	0,478	FIXE	CLAE	FIXE	1,4152	30	0,85	1,4152	1,403
13	Limelette_drone_coupe.mp4	3->6	3	NON	NON	NON	0,478	FIXE	CLAE	FIXE	1,4152	30	0,85	1,4152	1,403
14	Limelette_drone_coupe.mp4	3->6	3	NON	NON	NON	0,478	FIXE	CLAE	FIXE	1,4152	30	0,85	1,4152	1,403
15	Limelette_drone_coupe.mp4	3->6	3	NON	NON	NON	0,478	FIXE	CLAE	FIXE	1,4152	30	0,85	1,4152	1,403

FIGURE 9.15 – Tests effectués avec l'ordinateur 1

n°	Nom video	Range	Step	Stab.	Calib.	Ortho.	Hauteur d'eau	ROI/mask	Filtre	Section d'intérêt	scaling factor	nbre sous-sec.	α	Débit	Débit attendu
1	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI	0,478	FIXE	CLAE	FIXE	149,78264	20	0,85	1,2524	1,403
2	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI	0,478	FIXE	CLAE	FIXE	150,206519	20	0,85	1,2424	1,403
3	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI	0,478	FIXE	CLAE	FIXE	150,142497	20	0,85	1,2314	1,403
4	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI	0,478	FIXE	CLAE	FIXE	148,943423	20	0,85	1,2717	1,403
5	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI	0,478	FIXE	CLAE	FIXE	150,184292	20	0,85	1,2453	1,403
6	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI/FIXE	0,478	FIXE	CLAE	FIXE	149,820306	20	0,85	1,2509	1,403
7	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI/FIXE	0,478	FIXE	CLAE	FIXE	149,820306	20	0,85	1,2509	1,403
8	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI/FIXE	0,478	FIXE	CLAE	FIXE	149,820306	20	0,85	1,2509	1,403
9	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI/FIXE	0,478	FIXE	CLAE	FIXE	149,820306	20	0,85	1,2509	1,403
10	terrain_limelette_video1_camera1.MOV	11->14	3	NON	NON	OUI/FIXE	0,478	FIXE	CLAE	FIXE	149,820306	20	0,85	1,2509	1,403
11	Limelette_drone_coupe.mp4	3->6	3	NON	NON	NON	0,478	FIXE	CLAE	FIXE	151,787082	30	0,85	1,4152	1,403
12	Limelette_drone_coupe.mp4	3->6	3	NON	NON	NON	0,478	FIXE	CLAE	FIXE	151,787082	30	0,85	1,4152	1,403
13	Limelette_drone_coupe.mp4	3->6	3	NON	NON	NON	0,478	FIXE	CLAE	FIXE	151,787082	30	0,85	1,4152	1,403
14	Limelette_drone_coupe.mp4	3->6	3	NON	NON	NON	0,478	FIXE	CLAE	FIXE	151,787082	30	0,85	1,4152	1,403
15	Limelette_drone_coupe.mp4	3->6	3	NON	NON	NON	0,478	FIXE	CLAE	FIXE	151,787082	30	0,85	1,4152	1,403

FIGURE 9.16 – Tests effectués avec l'ordinateur 2

A la vue de ces résultats, nous pouvons tirer deux conclusions :

- Ce premier test nous a montré l'importance de bloquer tous les paramètres lorsque l'on souhaite étudier l'influence d'un élément précis. Cette observation a été très utile pour les séries de tests suivantes.
- Sur base des tests 6 à 15, nous pouvons confirmer que l'ordinateur n'a aucune influence sur le débit calculé. Si des différences apparaissent, elles seront dues à l'utilisateur ou à la variation d'un paramètre.

2.2 Influence de la méthode de validation de l'analyse PIV

A plusieurs reprises dans notre travail, nous avons expliqué que, lors du post-processing de l'analyse PIV, les vecteurs vitesses étaient validés. Plusieurs méthodes de validation sont possibles. Elles ont été présentées à la section 3.2 du chapitre 3. Il en existe cinq : `sig2noise_val`, `global_val`, `global_std`, `local_median_val` et `typical_value`. Le but de cette deuxième série de tests était de déterminer quelles méthodes donnent les meilleurs résultats.

Pour ce faire, plusieurs séries de tests ont été réalisés sur quatre vidéos différentes. Pour chacune d'entre elles, tous les paramètres ont été bloqués, nous permettant ainsi de réaliser exactement le même test pour chaque méthode de validation. C'est ce que nous appellerons les *paramètres fixes*. L'opération est ensuite répétée à trois reprises et nous obtenons un tableau de résultats de forme identique à celui représenté à la figure 9.18. Un exemple du tableau des paramètres associés à ces résultats se trouve à la figure 9.17. La différence entre les trois *paramètres fixes* se situe au niveau de l'orthorectification, du ROI, du masque et de la section d'intérêt. La variation des résultats est donc entièrement due à l'utilisateur.

SERIE A - Limelette	
nom video	terrain_limelette_vide o1_camera1.MOV
range	11->14
Step	3
Stabilisation	NON
Calibration	NON
Orthorectification	OUI
Hauteur d'eau	0,478
Filtre	CLAHE
Scaling factor	FIXE
ROI / Mask	FIXE
Section d'intérêt	FIXE
nbre de sous section	20
Alpha	0,85
Débit attendu	1,403

	Arg. 1	Arg. 1	Arg. 1	Arg. 1	Arg. 1	Paramètres FIXE A.1	Paramètres FIXE A.2	Paramètres FIXE A.3
Sig2noise	u0	v0	sig2noise	threshold=SNR	-	Débit	Débit	Débit
Median_val	u0	v0	3	3	size=1	1,2442	1,1411	1,2605
global_std	u0	v0	threshold = 3	-	-	1,0901	0,9621	1,171
global_val	u0	v0	(-150,150)	(-150,150)	-	1,0468	0,8303	1,0835
typical	u0	v0	sig2noise	settings	-	1,0397	0,8279	1,0885
						0,6017	0,6998	0,5662

FIGURE 9.18 – Tests série A

FIGURE 9.17 – Paramètres fixes A

Nous pouvons alors rassembler tous ces résultats pour obtenir le tableau de la figure 9.19.

	SERIE A - Limelette			SERIE B - Noirath			SERIE C - Limelette			SERIE D - Noirath		
	Paramètres FIXE A.1	Paramètres FIXE A.2	Paramètres FIXE A.3	Paramètres FIXE B.1	Paramètres FIXE B.2	Paramètres FIXE B.3	Paramètres FIXE C.1	Paramètres FIXE C.2	Paramètres FIXE C.3	Paramètres FIXE D.1	Paramètres FIXE D.2	Paramètres FIXE D.3
	Débit	Débit	Débit	Débit	Débit	Débit	Débit	Débit	Débit	Débit	Débit	Débit
Sig2noise	1,2442	1,1411	1,2605	0,2065	0,2014	0,2038	1,4152	1,4113	1,3645	0,2069	0,2037	0,2072
Median_val	1,0901	0,9621	1,171	0,1959	0,1979	0,1952	1,0495	0,9831	0,9605	0,1784	0,1829	0,1755
global_std	1,0468	0,8303	1,0835	0,1878	0,1883	0,1897	0,7477	0,6883	0,7033	0,1547	0,1581	0,1432
global_val	1,0397	0,8279	1,0885	0,1863	0,1885	0,1892	0,7464	0,6918	0,6874	0,0696	0,0891	0,0551
typical	0,6017	0,6998	0,5662	0,0178	0,0204	0,0194	0,0389	0,026	0,0359	0,0017	0,0033	0,0019

FIGURE 9.19 – Débits calculés sur 4 vidéos

Au travers de ces premiers résultats, une tendance se dessine déjà. En effet, sachant que le débit réel à Limelette est de 1,403 $[\frac{m^3}{s}]$ alors qu'à Noirhat, le débit est de 0,203 $[\frac{m^3}{s}]$, la méthode `sig2noise_val` donne de meilleurs résultats dans la plupart des cas. Bien que l'observation des résultats obtenus à partir des vidéos A, C et D semble univoque, la série de tests B réalisés à partir d'une vidéo de Noirhat n'est pas aussi catégorique. En effet, les résultats de la méthode `Median_val` ainsi que, mais dans une moindre mesure, ceux des méthodes `global_std` et `global_val` donnent également des débits acceptables. Il est donc légitime de se demander si les conditions des paramètres fixes sont représentatives. C'est pourquoi des tests supplémentaires ont été réalisés sur les séries de tests A, B et D. Cette fois, les paramètres fixes restent identiques, seuls les filtres et la présence des traceurs sont modifiés.

	tests	Paramètres variables		Paramètres FIXE A.3				
		Range	Filtre	sig2noise	Median_val	global_std	global_val	typical
SERIE A - LIMELETTE	1	11->14	CLAHE	1,2605	1,171	1,0835	1,0885	0,5662
	2	24->27	CLAHE	0,9693	0,6185	0,6583	0,6532	0,4019
	3	11->13	SANS FILTRE	1,2365	1,1672	1,0799	1,0854	0,5811
	4	11->13	Gamma - 2	1,2488	1,1884	1,1147	1,1081	0,5702
	5	11->13	High pass - 10	1,2473	1,1838	1,098	1,1095	0,5598

FIGURE 9.20 – Tests supplémentaires sur la série A

	tests	Paramètres variables		Paramètres FIXE B.3				
		Range	Filtre	sig2noise	Median_val	global_std	global_val	typical
SERIE B - Noirath	1	6->8	CLAHE	0,2038	0,1952	0,1897	0,1892	0,0194
	2	20->22	CLAHE	0,0621	0,0543	0,0368	0,037	0,0062
	3	6->8	SANS FILTRE	0,2002	0,189	0,1828	0,1815	0,0167
	4	6->8	Gamma - 2	0,2014	0,1888	0,1829	0,1765	0,0148
	5	6->8	High pass - 10	0,1817	0,1866	0,1725	0,1765	0,0148

FIGURE 9.21 – Tests supplémentaires sur la série B

	tests	Paramètres variables		Parametres FIXE D.3				
		Range	Filtre	sig2noise	Median_val	global_std	global_val	typical
SERIE D - Noirath	1	6->8	CLAHE	0,2072	0,1755	0,1432	0,0551	0,0019
	2	20->22	CLAHE	0,0672	0,0552	0,063	0,0507	0,0032
	3	6->8	SANS FILTRE	0,2073	0,1846	0,1447	0,0545	0,0019
	4	6->8	Gamma - 2	0,2095	0,1882	0,1426	0,053	0,0016
	5	6->8	High pass - 10	0,1968	0,1734	0,1415	0,0557	0,0025

FIGURE 9.22 – Tests supplémentaires sur la série D

Peu importe le filtre utilisé, nous observons à nouveau de meilleurs résultats lorsque la méthode de validation `sig2noise_val()` est utilisée. Nous pouvons donc confirmer notre première intuition et c'est cette méthode qui restera implémentée dans l'application.

2.3 Influence de la présence des traceurs

L'objectif de ce test est de déterminer l'importance des traceurs ajoutés dans une rivière sur le débit obtenu. En effet, comme l'ajout de traceurs est assez contraignant, il convient de se demander si ceux-ci sont réellement utiles dans l'obtention d'un débit correct et si oui, quelle précision ils ajoutent aux résultats.

Les tests que nous avons effectués ont été réalisés sur les vidéos de Limelette, Noirhat et le jeu 1. Cette partie de la validation a été divisée en deux phases :

1. Étant donné que le test vise à déterminer l'importance des traceurs, le contraste avec le fond de la rivière est important. Plusieurs filtres ont été testés sur chaque vidéo et les résultats ont été comparés afin de choisir le filtre le plus performant.
2. Pour le filtre choisi, une comparaison de la vidéo avec et sans traceur a été réalisée sur cinq tests différents. Pour chaque test entre la partie avec et sans traceurs, tous les paramètres excepté ont été fixés. Cependant, entre les tests, ces paramètres sont modifiés pour montrer l'influence réelle des particules (voir figure 9.26).

Limelette



FIGURE 9.23 – Cas de Limelette sans particule ajoutée



FIGURE 9.24 – Cas de Limelette avec particules

Dans cette vidéo, les résultats obtenus sur la partie sans particules étaient meilleurs sans filtre que ceux obtenus avec l'application des différents filtres (voir figure 9.25).

Q = 1,403 m³/s	Paramètres fixes 1			
	SANS FILTRE	CLAHE	Gamma - 2	High pass - 10
avec	1,2999	1,2924	1,2999	1,3159
sans	1,1167	1,0899	1,0805	1,0781

FIGURE 9.25 – Tests des différents filtres sur Limelette

Q = 1,403 m³/s	SANS FILTRE	
	Avec	sans
Paramètres fixes 1	1,2999	1,1167
Paramètres fixes 2	1,2398	0,9834
Paramètres fixes 3	1,1534	1,0481
Paramètres fixes 4	1,194	1,0126
Paramètres fixes 5	1,2413	0,9951

FIGURE 9.26 – Tests sur Limelette avec les paramètres variables sans filtre

Noirhat

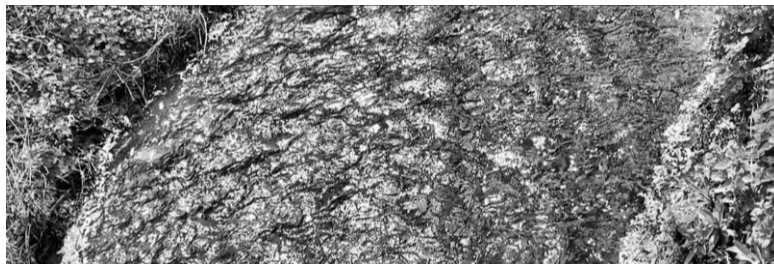


FIGURE 9.27 – Cas de Noirath sans particule ajoutée



FIGURE 9.28 – Cas de Noirhat avec particules

Dans ce cas-ci, le filtre permettant de s'approcher au plus près du débit réel est le filtre CLAHE (voir figure 9.29).

Q = 0,203 m³/s	Paramètres fixes 1				
	CLAHE	SANS FILTRE	GAMMA - 2	HIGH PASS - 10	Contrast - 0.5 - 100
avec	0,2072	0,2073	0,2095	0,1968	0,1422
sans	0,0672	0,078	0,0633	0,0564	0,0775

FIGURE 9.29 – Tests des différents filtres sur Noirhat

Étant donné que les résultats obtenus sans particule sont clairement inférieurs au débit réel, les tests avec paramètres variables n'ont pas été réalisés car ils n'auraient pas permis d'observation supplémentaire.

Jeu 1



FIGURE 9.30 – Jeu 1 sans particule ajoutée

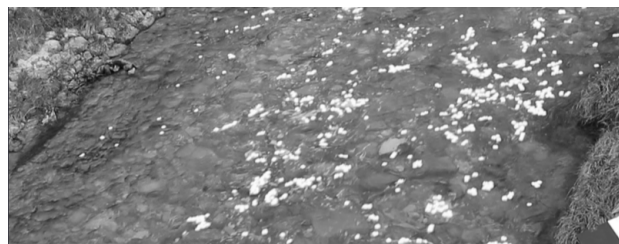


FIGURE 9.31 – Jeu 1 avec particules

Dans ce cas-ci, le débit le plus proche de la réalité était obtenu sans filtre (voir figure 9.32).

Q = 1,26 m ³ /s	Paramètres fixes 1					
	CLAHE	CAPPING	Sans filtre	HIGH PASS - 10	Contrast - 0.5 - 100	GAMMA
avec	0,6266	1,0778	1,2028	0,4118	1,5319	1,3361
sans	0,0081	0,0062	0,0105	0,0058	0,0058	0,0086

FIGURE 9.32 – Tests des différents filtres sur le Jeu 1

A nouveau, dans ce cas-ci, les résultats sans particule sont très éloignés du débit réel et ce peu importe le filtre appliqué. Les tests avec paramètres variables n'ont donc pas été réalisés.

Conclusions

Les conclusions de ces tests sont univoques. Pour un cours d'eau de faible débit, il est obligatoire d'ajouter des traceurs. Pour les cours d'eau à débit plus important, nous n'avons malheureusement pas pu réaliser ce test par manque d'une vidéo présentant une situation avec et sans traceurs.

2.4 Limites dues à la forme de la bathymétrie

Comme vu précédemment, la méthode image permet de calculer la hauteur d'eau sur base de la bathymétrie et de deux points, un sur chaque berge, renseignés par l'utilisateur (voir détails à la section 2.2 du chapitre 3). Avant toute phase de test, nous pouvons déjà affirmer que cette méthode a ses limites. Plus la pente des berges sera raide, plus une très légère variation de largeur d'eau entraînera une forte variation de la hauteur d'eau. Elle n'est d'ailleurs pas utilisable pour des bathymétries ayant des pentes de berges proches de la verticale, c'est-à-dire des bathymétries proches des canaux. Cette limite n'est pas très contraignante car il est souvent facile de mesurer la hauteur d'eau directement sur le terrain dans le cas d'une section en canal. Nous sommes donc en droit de nous poser la question suivante : dans le cas d'une rivière naturelle, pour quelle pente de berge la méthode image permet d'obtenir des débits satisfaisants ? C'est à cette question que nous tentons de répondre par ce test.

Tests effectués

Pour cette étude de la pente de la bathymétrie, des séries de 20 tests ont été réalisées sur base de 3 vidéos correspondant à 3 lieux différents et donc 3 bathymétries différentes. Rappelons également qu'une orthorectification est nécessaire pour utiliser la méthode *image*. Par conséquent, seules des vidéos prises de la berge ont été utilisées. Finalement, tous les paramètres différents de la hauteur d'eau ont été bloqués. Contrairement aux autres séries de tests de ce chapitre, nous comparerons les résultats obtenus avec le débit obtenu en utilisant la hauteur d'eau réelle et non pas le débit réel de la rivière.

1. La première série de tests a été réalisée à **Limelette**. La hauteur d'eau réelle est égale à 0,478 [m] ce qui équivaut à un débit calculé de 1,2413 [$\frac{m^3}{s}$] et à une largeur d'eau de 7,436 [m]. Les figures 9.33 et 9.34 représentent respectivement la variation de la hauteur d'eau en fonction de la largeur d'eau et la variation du débit en fonction de la hauteur d'eau.

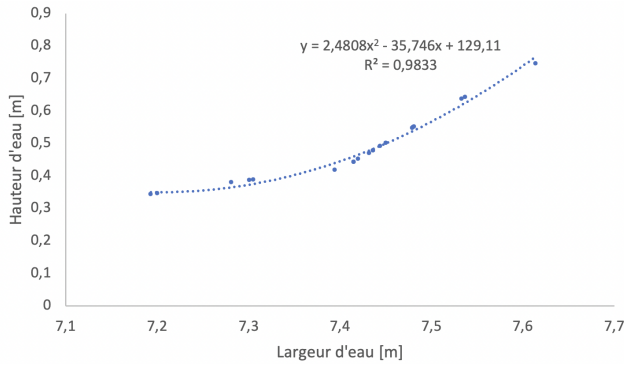


FIGURE 9.33 – Limelette - Hauteur d'eau en fonction de la largeur d'eau

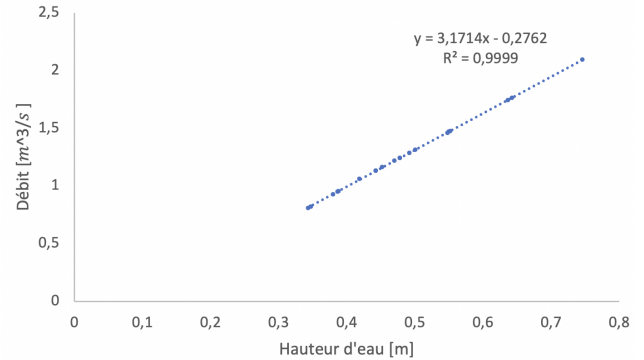


FIGURE 9.34 – Limelette - Débit en fonction de la hauteur d'eau

2. La deuxième série de tests a été réalisée à **Noirhat**. La hauteur d'eau réelle est égale à 0,155 [m] ce qui équivaut à un débit calculé de 1,2413 [$\frac{m^3}{s}$] et une largeur d'eau de 3,45 [m]. Les figures 9.35 et 9.36 représentent respectivement la variation de la hauteur d'eau en fonction de la largeur d'eau et la variation du débit en fonction de la hauteur d'eau.

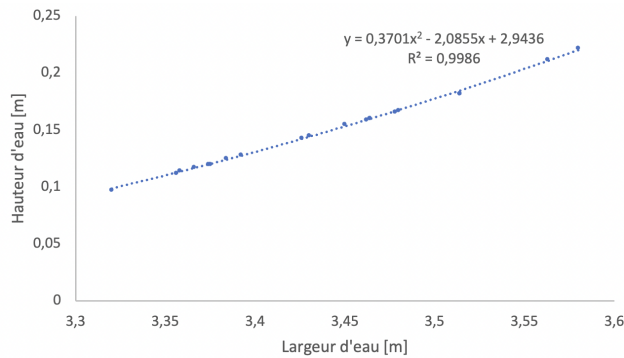


FIGURE 9.35 – Noirhat - Hauteur d'eau en fonction de la largeur d'eau

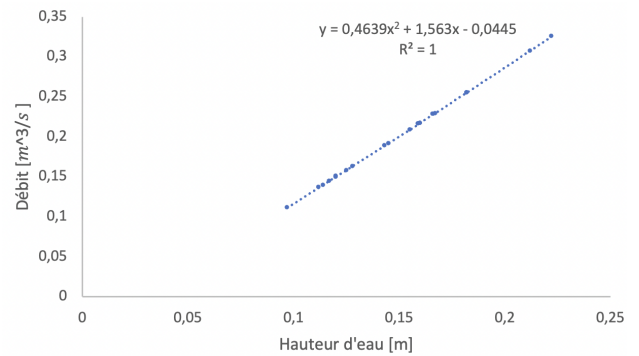


FIGURE 9.36 – Noirhat - Débit en fonction de la hauteur d'eau

3. La dernière série de tests a été réalisée à partir de la première vidéo du Vidéo Globe Challenge, c'est-à-dire à partir du **Jeu 1**. La hauteur d'eau réelle est égale à 0,155 [m] ce qui équivaut à un débit calculé de 1,2413 [$\frac{m^3}{s}$] et une largeur d'eau de 5,754 [m]. Les figures 9.37 et 9.38 représentent respectivement la variation de la hauteur d'eau en fonction de la largeur d'eau et la variation du débit en fonction de la hauteur d'eau.

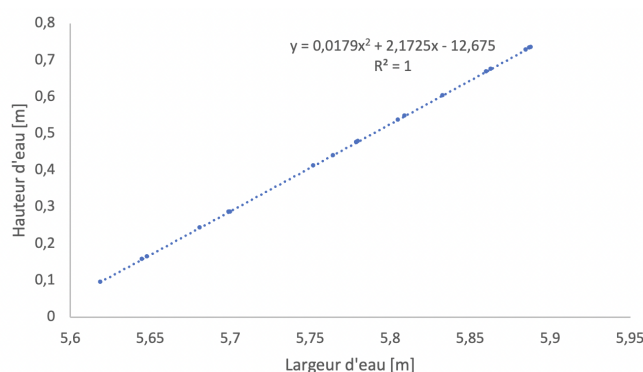


FIGURE 9.37 – Noirhat - Hauteur d'eau en fonction de la largeur d'eau

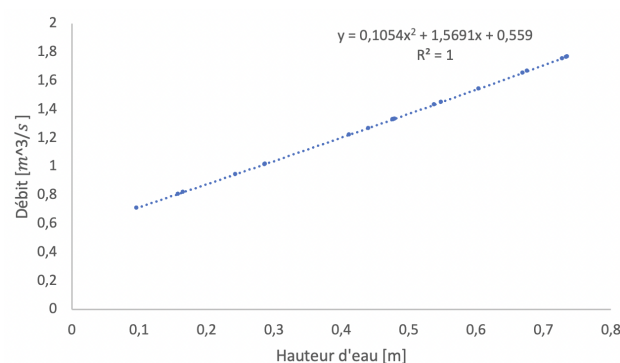


FIGURE 9.38 – Noirhat - Débit en fonction de la hauteur d'eau

Première analyse

Dans un premier temps, nous nous sommes intéressés aux valeurs extrêmes obtenues. Comme nous pouvons le voir au tableau 9.1, l'erreur causée par l'utilisateur ne dépasse pas les 4 % de la largeur d'eau attendue. Cette erreur est propre à la méthode *image* et ne pourra pas être diminuée. De plus, nous remarquons que le pourcentage d'erreur sur le calcul de la hauteur d'eau et du débit est beaucoup plus important. Ces résultats ne sont pas étonnants, étant donné que les trois bathymétries correspondantes possèdent des pentes de berges relativement raides. Les pentes moyennes des berges sont données à titre indicatif au tableau 9.2.

	Limelette		Noirhat		Jeu 1	
	Valeur	Erreur	Valeur	Erreur	Valeur	Erreur
L_{min}	7,193 [m]	3 [%]	3,32 [m]	4 [%]	5,619 [m]	3 [%]
L_{max}	7,614 [m]	2 [%]	3,58 [m]	4 [%]	5,888 [m]	2 [%]
h_{min}	0,343 [m]	28 [%]	0,097 [m]	37 [%]	0,096 [m]	78 [%]
h_{max}	0,746 [m]	56 [%]	0,222 [m]	43 [%]	0,735 [m]	67 [%]
Q_{min}	0,808 [$\frac{m^3}{s}$]	35 [%]	0,1115 [m]	47 [%]	0,7107 [m]	44 [%]
Q_{max}	2,091 [$\frac{m^3}{s}$]	68 [%]	0,3256 [m]	56 [%]	1,7696 [m]	39 [%]

TABLE 9.1 – Valeurs extrêmes

	Pente moyenne
Limelette	260 [%]
Noirhat	89 [%]
Jeu 1	596 [%]

TABLE 9.2 – Pente moyenne des bathymétrie

Le cas de Noirhat peut interpellier car sa bathymétrie possède une pente moyenne plus faible que les deux autres lieux mais les erreurs sur la hauteur d'eau calculée et sur le débit restent du même ordre de grandeur. Deux facteurs peuvent expliquer ces résultats :

1. Le pourcentage d'erreur sur la sélection de la largeur d'eau est plus importante à Noirhat qu'aux autres endroits. C'est une erreur entièrement causée par l'utilisateur.
2. Alors que le débit à Limelette et sur le Jeu 1 est du même ordre de grandeur (autour de $1,3 [\frac{m^3}{s}]$), celui à Noirhat est six fois moins important ($0,2 [\frac{m^3}{s}]$). Le débit étant beaucoup plus faible, les erreurs sont amplifiées.

Nous pouvons également noter que le pourcentage d'erreur sur le calcul de la hauteur d'eau dans le cas du jeu 1 est particulièrement élevé. Ceci s'explique par la pente des berges de sa bathymétrie qui sont particulièrement raides.

Finalement, comme le pourcentage d'erreur est amplifié surtout lors du calcul de la hauteur d'eau à partir la largeur d'eau, le problème peut être simplifié. Nous tentons maintenant de déterminer la pente moyenne des berges, appelée γ dans la suite, qui garantit un pourcentage d'erreur acceptable sur la hauteur d'eau calculée, sachant que 4 % d'erreur sur la largeur d'eau ne peuvent pas être évités.

Détermination de l'angle théorique

Un choix d'erreur acceptable sur le calcul de la hauteur d'eau doit donc être fait. Nous avons donc fixé le pourcentage d'erreur acceptable sur la hauteur d'eau à 5%. Le problème est synthétisé à la figure 9.39 et l'angle γ est simplement calculé avec l'équation 9.1.

$$\tan(\gamma) = \frac{\Delta h}{\Delta L} = \frac{0,05 \cdot h_{réelle}}{0,04 \cdot L_{attendue}} \quad (9.1)$$

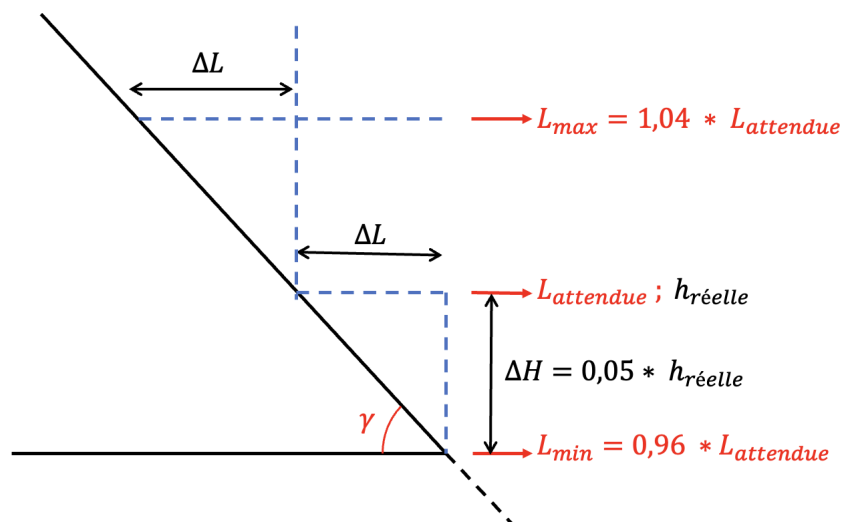


FIGURE 9.39 – Détermination de l'angle γ acceptable

Sur base de nos trois séries de tests, nous calculons γ , exprimé en degrés et en pourcentages de pente (voir tableau 9.3).

	$h_{réelle}$ [m]	$L_{attendue}$ [m]	γ [°]	γ [%]
Limelette	0,478	7,44	4,6	8
Noirhat	0,155	3,35	3,2	5,6
Jeu 1	0,44	5,76	5,2	9

TABLE 9.3 – Angles γ acceptables

Conclusion

En première approximation, nous remarquons qu'il faudrait limiter l'utilisation de cette méthode à des bathymétries ayant des pentes de berges inférieures à 10 % par rapport à l'horizontale et même inférieures à 5 % dans le cas de débits très faibles. Une telle condition paraît

fort restrictive et diminue grandement l'intérêt de la méthode.

Heureusement, la méthode redevient intéressante si elle est utilisée de façon statistique. En effet, en comparant le débit attendu au débit moyen calculé sur les 20 tests respectifs de chaque lieu, l'erreur calculée ne dépasse pas les 4 % dans nos exemples (voir 9.4).

	$Q_{attendu} [\frac{m^3}{s}]$	$Q_{moyen,20} [\frac{m^3}{s}]$	Erreur [%]
Limelette	1,2413	1,261	2
Noirhat	0,2089	0,2	4
Jeu 1	1,2697	1,309	3

TABLE 9.4 – Débits moyens calculés sur 20 tests

Bien entendu, une analyse plus poussée sur un plus grand échantillon de tests devrait être réalisée pour confirmer cette première analyse.

2.5 Influence de l'étape d'orthorectification

Influence de l'orthorectification

L'orthorectification correspondant à une correction géométrique supplémentaire, nous pouvons nous attendre à ce que l'erreur sur le débit soit plus importante. Ce test a donc pour objectif de déterminer cette erreur.

Pour cela, deux vidéos réalisées à Limelette, filmées au même moment avec des conditions de luminosité identiques, ont été utilisées. La première vidéo a été filmée depuis le pont qui surplombe la rivière et nécessite donc une orthorectification (voir figure 9.40).

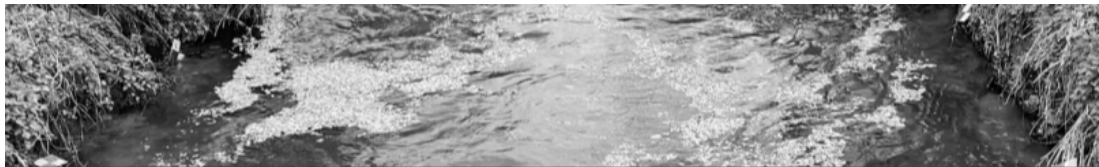


FIGURE 9.40 – Vidéo filmée depuis le pont

La seconde vidéo a été filmée par un drone perpendiculairement à la surface d'eau et ne nécessite donc pas d'orthorectification.



FIGURE 9.41 – Vidéo filmée perpendiculairement à la surface d'eau

Sur chaque vidéo, 20 tests ont été réalisés. Le nombre 20 a été choisi car il permet de donner une valeur statistique valable et donc d'obtenir une moyenne représentative. Entre ces 20 tests réalisés, les paramètres ont été fixés.

Q = 1,403 m ³ /s	
Résultats Caméra 1	
Moyenne	1,1398
Erreur	19%

FIGURE 9.42 – Résultats de la vidéo qui nécessite une orthorectification

Q = 1,403 m ³ /s	
Résultats Caméra 2	
Moyenne	1,3883
Erreur	1%

FIGURE 9.43 – Résultats de la vidéo qui ne nécessite pas d'orthorectification

Les résultats obtenus montrent que la vidéo qui ne nécessite pas d'orthorectification fournit un débit moyen très précis alors que celle qui a été orthorectifiée est beaucoup moins précise (voir figures 9.42 et 9.43).

Cela est probablement dû au fait que lorsqu'une orthorectification est réalisée, l'influence de l'utilisateur est beaucoup plus importante. Les résultats varient donc plus. En effet, le choix des points de contrôle et du masque ne sera jamais totalement identique, faisant varier le facteur d'échelle ainsi que la section d'intérêt. Ces variabilités induites par l'orthorectification seront plus importantes que dans le cas non orthorectifié, où seuls la section d'intérêt et le facteur d'échelle seront impactés par l'erreur de l'utilisateur.

Nous pouvons donc conclure qu'une vidéo filmée perpendiculairement au plan d'eau produit en moyenne des résultats très précis. Si l'utilisateur a la possibilité de filmer la vidéo selon cet angle, il pourra s'attendre à des résultats très proches de la réalité. Cependant, les résultats produits par une vidéo qui nécessite une orthorectification restent, eux aussi, proches de la réalité.

Influence de l'angle de la caméra sur l'orthorectification

L'objectif de ce test est de déterminer l'influence de l'angle de vue de la caméra sur l'orthorectification et le débit obtenu. Il est important de savoir si l'application est capable de faire face à des vidéos filmées de tout angle ou si un angle donne des résultats sensiblement meilleurs que l'autre.

Pour cela, nous avons filmé la rivière de Noirhat à partir de deux caméras positionnées comme indiqué à la figure 9.44.

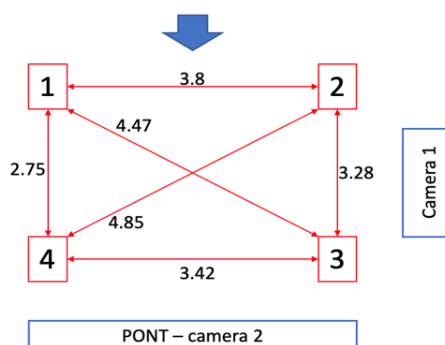


FIGURE 9.44 – Position des deux caméras sur le site de Noirhat

Ces deux caméras ont filmé la rivière en même temps ; les conditions de luminosité ainsi que les traceurs sont donc identiques. La caméra 1 filmait la rivière de côté, avec une partie de la végétation qui couvre une petite portion de la rivière (voir figure 9.45). La caméra 2 était quant-à-elle filmée depuis le pont surplombant la rivière (voir figure 9.46). Les conclusions de ce

test donneront donc une indication mais seront insuffisantes pour tirer une conclusion générale sur un angle idéal.



FIGURE 9.45 – Angle de vue de la caméra 1



FIGURE 9.46 – Angle de vue de la caméra 2

Sur chacune des deux vidéos, 20 tests aux conditions identiques ont été réalisés. Tous les paramètres ont été fixés excepté ceux relatifs à l'orthorectification.

Q = 0,203 m ³ /s	
Résultats Caméra 1	
Moyenne	0,2110
Erreur	4%

FIGURE 9.47 – Résultats de la caméra 1

Q = 0,203 m ³ /s	
Résultats Caméra 2	
Moyenne	0,2042
Erreur	1%

FIGURE 9.48 – Résultats de la caméra 2

Les résultats montrent une erreur de la moyenne légèrement supérieure de la caméra 1 par rapport à la caméra 2 (voir figures 9.47 et 9.48). Pour rappel, ces tests sont insuffisants pour conclure à une meilleure performance de la caméra 2. De plus, les résultats de la caméra 1 sont également très proches de la valeur de débit attendue et montrent que l'application est capable de déterminer un débit correct selon les deux angles testés.

Les résultats légèrement moins précis de la caméra 1 peuvent s'expliquer par la végétation présente sur la berge de droite qui cache une partie de la rivière et empêche donc de détecter certains vecteurs vitesse. La caméra 2, placée dans l'axe de la rivière, ne présente pas ce problème. Une expérience avec des tests plus complets et prenant en compte plus d'angles différents pourrait compléter ces observations.

2.6 Influence du ROI

L'objectif de ce test est de déterminer l'influence du choix du ROI et du masque sur le débit. Cela permettra de quantifier l'erreur apportée par l'utilisateur lors de cette étape et de déterminer si celle-ci peut être négligée.

Les tests de cette partie ont été effectués sur trois vidéos différentes : le Jeu 1, Limelette (filmée avec un drone) et Noirhat (voir figures 9.49, 9.51 et 9.50).



FIGURE 9.49 – Jeu 1 - $Q = 1,26 \text{ m}^3/\text{s}$



FIGURE 9.50 – Noirhat - $Q = 0,203 \text{ m}^3/\text{s}$



FIGURE 9.51 – Limelette - $Q = 1,403 \text{ m}^3/\text{s}$

Sur chaque vidéo, 20 tests ont été réalisés pour obtenir une valeur moyenne représentative. Entre ces tests, tous les paramètres variables ont été fixés excepté le ROI et le masque, qui variaient à chaque test. Sur la vidéo de Limelette filmée perpendiculairement à l'écoulement, la section d'intérêt n'a pas pu être fixée étant donnée qu'elle est sélectionnée sur le ROI.

	Jeu 1 $Q = 1,26 \text{ m}^3/\text{s}$		Limelette $Q = 1,403 \text{ m}^3/\text{s}$		Noirhat $Q = 0,203 \text{ m}^3/\text{s}$	
	Q [m^3/s]	Erreur	Q [m^3/s]	Erreur	Q [m^3/s]	Erreur
Moyenne	1,3075	4%	1,3517	4%	0,2054	1%
Minimum	1,1448	10,06%	1,2853	9,16%	0,2002	1,38%
Maximum	1,4646	14%	1,4252	2%	0,2129	5%

FIGURE 9.52 – Résultats des tests sur l'influence du ROI

Nous remarquons que pour les trois vidéos, la valeur moyenne des tests est très précise car l'erreur maximale est de 4%. Cette faible erreur est probablement due au fait que le ROI influence le résultat de l'analyse PIV car il modifie le repère dans lequel les centres des aires d'interrogation sont définis. C'est la position de ces points qui va influencer le calcul des vitesses. Cependant, pour les valeurs extrêmes, une erreur plus importante apparaît pour le cas du Jeu 1. Cela est probablement dû au fait que le nombre de traceurs ajoutés est beaucoup moins important que sur les deux autres cours d'eau (voir figure 9.49). Ceux-ci risquent alors de ne pas couvrir toute la section d'intérêt choisie.

Il est possible que, dans certains cas, l'utilisateur trouve exceptionnellement cette valeur extrême. Il est conseillé, étant donné la précision de la moyenne des résultats, d'effectuer plusieurs fois le test en changeant la région d'intérêt et de réaliser une moyenne des résultats obtenus.

2.7 Influence de la section d'intérêt sans orthorectification

L'objectif de ce test est de déterminer l'influence du choix de la section d'intérêt sur le débit dans le cas où la vidéo ne nécessite pas d'orthorectification. Le facteur d'échelle calculé dans ce cas dépend directement de cette section choisie, qui aura donc une influence encore plus importante (voir section 5 du chapitre 7).

Les tests de cette partie ont été réalisés sur trois vidéos différentes, toutes trois filmées perpendiculairement au plan d'eau (voir figures 9.53, 9.54 et 9.55).



FIGURE 9.53 – Limelette - $Q = 1,403 \text{ m}^3/\text{s}$



FIGURE 9.54 – Rosières - $Q = 0,664 \text{ m}^3/\text{s}$



FIGURE 9.55 – Jeu 7 - $Q = 655 \text{ m}^3/\text{s}$

Sur chaque vidéo, 20 tests ont été réalisés pour que la moyenne soit représentative. Entre ces 20 tests, les paramètres ont été fixés afin que seule la section d'intérêt soit source de variations.

	Limelette $Q = 1,403 \text{ m}^3/\text{s}$		Rosières $Q = 0,664 \text{ m}^3/\text{s}$		Jeu 7 $Q = 655 \text{ m}^3/\text{s}$	
	Q [m^3/s]	Erreur	Q [m^3/s]	Erreur	Q [m^3/s]	Erreur
Moyenne	1,3834	1%	0,5754	13%	682,2360	4%
Minimum	1,3287	5,59%	0,5043	31,67%	645,4216	1,5%
Maximum	1,4340	2%	0,6469	3%	719,0526	9%

FIGURE 9.56 – Résultats des tests sur l'influence de la section d'intérêt

Sur la figure 9.56, nous constatons que les valeurs moyennes sont assez proches des valeurs obtenues excepté les résultats de Rosières. L'influence de la section d'intérêt semble donc faible sur base des valeurs moyennes. Cependant, il arrive que les valeurs extrêmes de ce test soient assez éloignées de la valeur de débit attendu (voir figure 9.56). Les valeurs moyennes sont donc plus proches des débits attendu que les valeurs extrêmes. C'est pour cette raison qu'il est conseillé à l'utilisateur d'effectuer plusieurs tests en changeant la section d'intérêt à l'aide du bouton *back* et de calculer une moyenne des débits obtenus.

Les résultats moyens moins précis obtenus sur la vidéo de Rosières peuvent s'expliquer par plusieurs points. Tout d'abord, la section d'intérêt correspondant à la largeur mesurée n'est pas évidente à déterminer étant donné la variabilité de la largeur d'eau (voir figure 9.54). De plus, la distribution des traceurs n'est pas suffisamment homogène. Si la section d'intérêt est choisie à un endroit légèrement différent, il risque d'y avoir une différence importante au niveau de la couverture de la largeur par les particules. Enfin, la surface d'eau est très calme et ne présente quasiment pas de mouvement en comparaison des autres vidéos utilisées, ce qui peut également influencer la variabilité des résultats.

2.8 Précision générale

Jusqu'à présent, nous avons étudié l'erreur commise à chaque étape de l'application de façon individuelle. Le but de cette dernière série de test est d'estimer sa précision générale.

Une série de 20 tests, tous différents, a donc été réalisée sur base de 8 vidéos différentes. Le résumé des résultats obtenus est donné à la figure 9.57. L'erreur correspond au pourcentage de différence entre le débit calculé et le débit attendu.

Test	Vidéo	Filtre	Erreur
1	terrain_limelette_video1_camera1.MOV	CLAHE	11%
2	Limelette_drone_coupe.mp4	CLAHE	1%
3	terrain_rosière_video1_coupe	SANS FILTRE	10%
4	Jeu7	CLAHE	1%
5	terrain_noirhat_video1_camera1.mp4	CLAHE	2%
6	terrain_limelette_video1_camera1.MOV	CLAHE	16%
7	Limelette_drone_coupe.mp4	CLAHE	0%
8	terrain_noirhat_video1_camera2.mp4	Gamma - 2	0%
9	video_jeu_1.MP4	SANS FILTRE	1%
10	terrain_noirhat_video1_camera2.mp4	CLAHE	1%
11	terrain_noirhat_video1_camera2.mp4	Gamma-2	2%
12	video_jeu_1.MP4	Gamma - 3	7%
13	Limelette_drone_coupe.mp4	Gamma - 2	2%
14	terrain_limelette_video1_camera1.MOV	CLAHE	12%
15	terrain_noirhat_video1_camera1.mp4	Gamma - 2	8%
16	terrain_noirhat_video1_camera2.mp4	CLAHE	6%
17	Jeu7	SANS FILTRE	15%
18	Limelette_drone_coupe.mp4	CLAHE	1%
19	GOPRO3225_Trim.mp4	SANS FILTRE	3%
20	terrain_noirhat_video1_camera1.mp4	SANS FILTRE	9%

FIGURE 9.57 – Tests généraux

Nous observons que dans la majorité des cas (80%), l'erreur est inférieure ou égale à 10%. Nous pouvons également en déduire que 55% des cas sont inférieurs ou égale à 3% d'erreur et un cas sur quatre se situe même à seulement 1% d'erreur.

Sur base de tests uniques, l'application corrigée affiche une précision tout à faire acceptable et même prometteuse. En effet, dans le point précédent, nous avons expliqué qu'une utilisation statistique de l'application permettrait de réduire fortement les erreurs dues à l'utilisateur. Il reste donc encore une marge d'amélioration.

3 Conclusions de validation

Suite à la phase de tests, plusieurs conclusions générales peuvent être déduites.

Tout d'abord, l'influence de l'utilisateur est assez faible si nous comparons la moyenne des résultats obtenus avec le débit réel. Le même phénomène a été constaté lors de l'utilisation de la méthode *image* qui nécessite également l'intervention de l'utilisateur pour déterminer le niveau d'eau. Cependant, certains tests peuvent produire des valeurs extrêmes alors que les

valeurs moyennes sont, quant à elles, quasiment semblables aux valeurs réelles. Nous conseillons donc toujours à l'utilisateur de réaliser plusieurs tests sur la vidéo qu'il analyse et de prendre la moyenne des résultats obtenus.

Ensuite, nous constatons que sur des cours d'eau à faible débit, l'ajout de traceurs, bien qu'il ne soit pas toujours évident, est obligatoire pour obtenir des résultats cohérents. Pour des débits plus importants, nous n'avons pas pu réaliser ce même test mais il serait intéressant de faire cette expérience afin de déterminer si l'ajout de traceurs améliore le résultat dans ce cas là.

De plus, l'étude sur l'orthorectification a permis de déterminer que celle-ci a une légère influence sur le débit obtenu, puisqu'elle fait intervenir l'utilisateur. Les résultats obtenus sur une vidéo filmée directement perpendiculairement au plan d'eau étaient, quant à eux, plus stables et plus proches de la valeur réelle. Il est également important de préciser que la vidéo peut être filmée de tout angle, tant que la végétation et la berge ne cachent pas la surface d'eau.

Enfin, de manière plus générale, il est conseillé à l'utilisateur de bien analyser la vidéo au préalable pour obtenir un débit proche de la valeur réelle. En effet, avant d'entrer les différentes informations dans l'application, il est important de choisir l'intervalle de la vidéo où les particules sont bien visibles et contrastées avec le fond de la rivière. Ensuite, il faut choisir un ROI correspondant à l'endroit où les particules passent dans l'intervalle de temps choisi et comprenant la section d'intérêt. Il est également à noter que la section d'intérêt doit correspondre à la section où la bathymétrie a été mesurée.

Pour terminer ces conclusions générales, nous rappelons qu'il s'agit là de premiers tests et que ceux-ci ne sont pas toujours complets. Ils demanderaient donc à être approfondis afin de confirmer certaines conclusions ci-dessus.

Chapitre 10

Résultat final

1 État des lieux

L'objectif de ce travail était d'améliorer l'application reçue et de la tester sur un ensemble de vidéos. A la fin de ce travail, nous pouvons affirmer qu'elle remplit désormais correctement plusieurs critères.

Tout d'abord, celle-ci est indépendante car elle reprend les étapes de pré et post-processing ainsi que quatre méthodes permettant de déterminer le niveau d'eau si celui-ci n'est pas connu. De cette manière, l'utilisateur ne devra pas forcément faire appel à d'autres logiciels ce qui est parfois le cas avec d'autres applications de calcul de débit.

Ensuite, la vidéo utilisée pour la détermination du débit peut être chargée plusieurs fois et peut être filmée de n'importe quel point de vue. Comme les étapes de traitement d'images (stabilisation, calibration, orthorectification et application des filtres) sont implémentées dans l'application, la vidéo pourra être traitée afin d'être correctement exploitée lors de l'analyse PIV même si elle présente des défauts. L'application peut donc traiter toutes sortes de vidéos sans restriction particulière. De plus, les vidéos traitées sont enregistrées afin de pouvoir être réutilisées dans un test futur. Dans les étapes de pré-processing, l'application propose également six filtres, permettant d'accentuer le contraste entre les particules et la rivière de différentes manières. Une aide permettant à l'utilisateur de choisir le filtre adéquat selon la vidéo a été ajoutée.

L'analyse PIV sur une vidéo entière prend un certain temps. Certaines méthodes telles que le range, le step et le ROI ont alors été implémentées pour réduire le temps de calcul.

Enfin, l'application est désormais user-friendly par la présence de plusieurs points :

- Les fenêtres de prévention lorsque l'utilisateur n'a pas importé correctement les données ou fichiers attendus.
- Les boutons *back* qui permettent de retourner à l'étape précédente sans devoir recommencer le processus complet.
- L'exportation des résultats de vitesses et de débit en fichier texte pour qu'ils puissent être traités si besoin.
- La possibilité de sélectionner certains éléments directement sur l'image affichée.
- La simplicité d'utilisation.

Cependant, bien que l'application puisse traiter un large champ de vidéos, elle possède tout de même plusieurs limites. Comme l'analyse PIV se base sur des images, il est essentiel que les conditions de luminosité soient suffisantes pour que les images soient exploitables. Il faut également veiller à ce que la végétation présente sur les berges ne cache pas la surface d'eau. De

plus, l'orthorectification implémentée dans l'application est une orthorectification 2D, c'est-à-dire que les quatre GCPs doivent être placés dans le même plan que le plan d'eau.

Finalement, les nouvelles options de l'application ont été ajoutées au flowchart initial, présenté à la section 1.2 du chapitre 4. Le flowchart final de l'application est donc représenté à la figure 10.1.

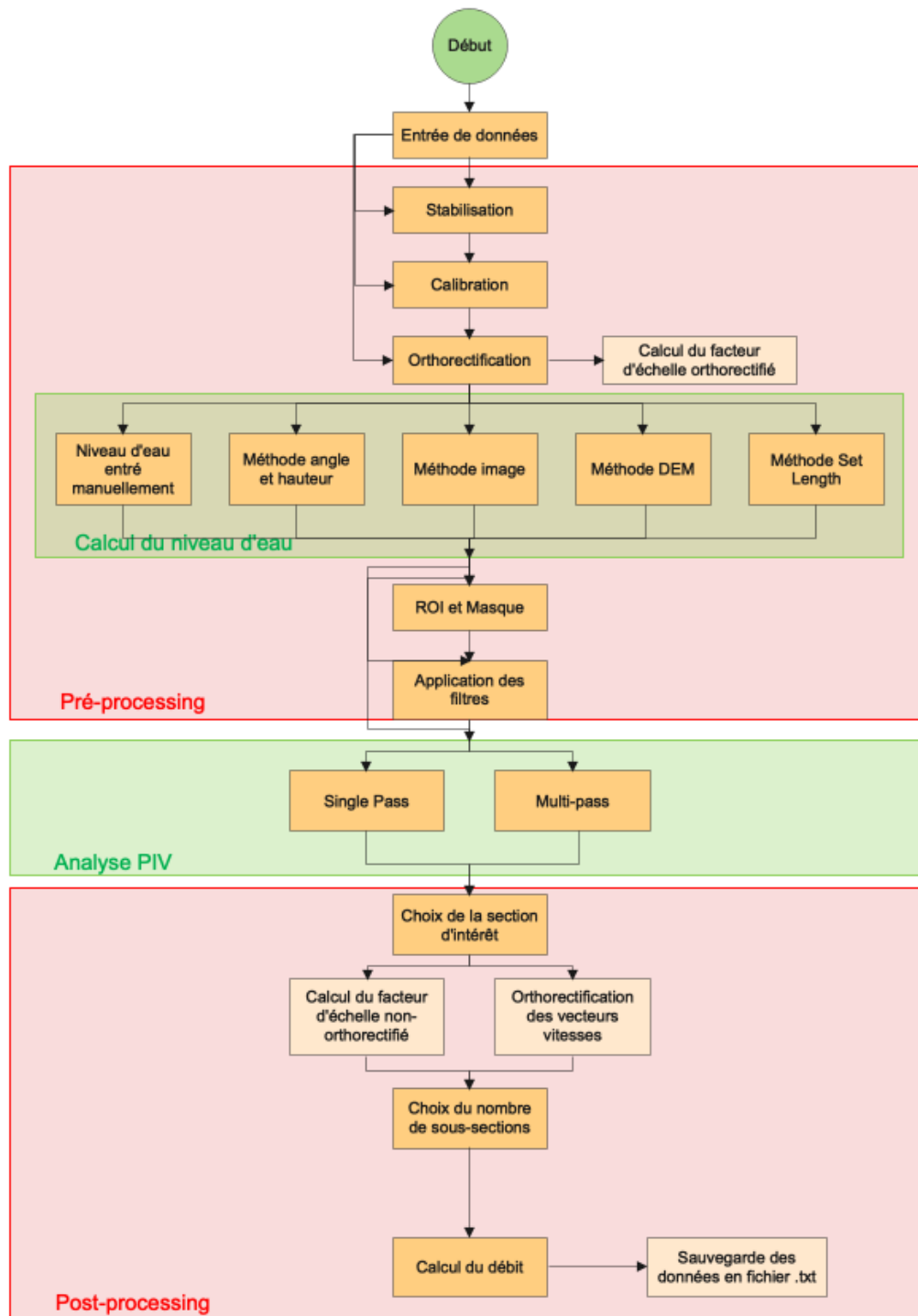


FIGURE 10.1 – Flowchart final

2 Pistes d'amélioration

Notre travail s'est principalement concentré sur la correction de l'analyse PIV et d'une partie des étapes de l'application. Nous n'avons malheureusement pas eu le temps de tester certaines méthodes en profondeur. Par exemple, la méthode *Angle et hauteur* ainsi que la calibration n'ont pas été vérifiées et testées de manière approfondie.

Lors d'une validation complémentaire, il serait intéressant de tester l'influence du nombre de sous sections afin, par exemple, de déterminer quel nombre est le plus adapté à un type de cours d'eau. Il pourrait aussi être utile de tester l'influence du coefficient α sur le débit obtenu.

Lors de notre travail sur l'analyse PIV, nous avons tenté d'implémenter une analyse multi-pass afin d'augmenter la précision des vecteurs vitesse calculés. Cependant, celle-ci n'est pas encore aboutie puisque les résultats qu'elle fournit ne sont pas stables et sont très éloignés des débits attendus. Il serait donc intéressant de poursuivre le travail entamé afin de proposer à l'utilisateur une seconde méthode pour l'analyse PIV.

Finalement, lors de nos tests de validation, nous avons souvent observé quelques valeurs extrêmes fort différentes du débit attendu mais la valeur moyenne calculée sur plusieurs tests était toujours très précise. Une des pistes d'amélioration majeure de l'application serait donc de trouver une méthode qui permettrait de calculer automatiquement cette moyenne. Prenons par exemple le cas de la méthode image, où l'erreur de l'utilisateur est de l'ordre de 4% lorsque celui-ci sélectionne deux points permettant de retrouver la largeur d'eau. Nous pourrions imaginer ne pas demander seulement 2 points, et donc une largeur à l'utilisateur, mais plusieurs couples de points. A partir de l'ensemble de ces couples de points, une largeur moyenne serait calculée. Le même principe pourrait être appliqué à l'étape de la sélection de la section d'intérêt. La difficulté est de trouver une méthode qui n'alourdit pas pour autant l'utilisation de l'application.

Conclusion

Depuis quelques années, les techniques d'imagerie sont de plus en plus utilisées pour la mesure de débit. En effet, celles-ci présentent notamment l'avantage d'être adaptées aux cours d'eau en période de crue.

Dans ce contexte, une application a été créée en 2020 par un étudiant de l'UCL dans le cadre de son mémoire. A l'issue de son travail, une première version de l'application reprenant plusieurs concepts des techniques non intrusives était disponible. Cependant, cette version n'avait alors été testée que sur une seule vidéo et demandait donc à être appliquée à un plus large échantillon de vidéos.

Lors de notre recherche théorique, nous nous sommes renseignés sur les différents logiciels de mesures de débit existants. Nous avons alors comparé l'application reçue avec ces logiciels et cela nous a permis de définir plusieurs objectifs.

Dans un premier temps nous avons testé l'application reçue sur un set de vidéos. Celui-ci comportait des vidéos filmées dans des conditions très variées : angles de prise de vue différents, mouvements de caméra, variation du débit, etc. Le débit de ces différents cours d'eau était fourni et permettait une comparaison avec le débit calculé par l'application. Grâce à ces premiers tests, nous avons pu identifier les limites de l'application et détecter certaines erreurs et incohérences dans le code. Nous avons également noté quelques améliorations à apporter pour remplir les objectifs définis au début du travail.

Dans un second temps, nous avons corrigé les problèmes détectés sur base des tests effectués auparavant. Une fois que chaque partie de l'application était corrigée, nous avons pu implémenter de nouvelles fonctionnalités.

Enfin, une fois toutes les modifications effectuées, nous avons réalisé une phase de validation des résultats. Pour cela, nous nous sommes rendus sur le terrain afin d'obtenir un nouveau set de vidéos plus complet. Une série de tests a été réalisée à partir de ces vidéos sur les différentes parties de l'application. Afin d'étudier l'influence de chaque paramètre, nous avons choisi de bloquer tous les paramètres provoquant une modification du résultat final excepté le paramètre étudié. Il est ressorti de ces tests que les résultats moyens, après les modifications et corrections effectuées, étaient très proches des débits réels. Certaines valeurs extrêmes pouvaient cependant apparaître. Pour éviter ce cas de figure, l'utilisateur peut répéter plusieurs fois le même test et utiliser la valeur moyenne de ceux-ci. Ces tests ont également permis de mettre en évidence la nécessité de l'ajout de traceurs pour un cours d'eau de faible débit. De plus, nous avons remarqué que l'erreur causée par l'utilisateur lorsqu'il choisit les paramètres était relativement faible. Cependant, en éliminant totalement cette erreur, l'application serait beaucoup plus fiable. Il serait donc intéressant d'investiguer cela lors d'un travail futur.

A l'issue de ce travail, l'application que nous présentons fournit désormais des résultats stables et précis. De plus, chaque partie a été minutieusement vérifiée et fonctionne correctement. Bien

qu'il reste certaines fonctionnalités à compléter, ce travail a permis de consolider en profondeur l'application qui pourra ainsi contribuer de manière fiable à la mesure de débit à partir de technique d'imagerie.

Table des figures

1.1	Capteur à moulinet [3]	5
1.2	Saumon d'une station de mesure [4]	5
1.3	Courantomètre électromagnétique [5]	5
1.4	ADCP [7]	6
1.5	Principe de l'autocorrelation [10]	8
2.1	3 cas différents peuvent se présenter à l'orthorectification [13]	11
2.2	Affichage des vitesses calculées lors de l'analyse PIV [12]	12
2.3	Chacun des quatre points de contrôle est placé dans une des quatre zones [18] . .	14
2.4	Sélection de la zone bleue qui permet de déterminer le niveau d'eau [18]	14
2.5	Affichage des vecteurs vitesses calculés [18]	15
2.6	Région d'intérêt (zone noire) et masques (zones rouges) choisi, après application des filtres [21]	16
2.7	Vecteurs vitesse après analyse PIV et validation [21]	17
2.8	Points de contrôles [21]	17
2.9	Région d'intérêt sélectionnée pour l'orthorectification [21]	17
2.10	Vecteurs vitesse de la région d'intérêt après orthorectification [21]	17
3.1	Effet d'une distorsion radiale [24]	21
3.2	Distorsion radiale positive et négative [26]	21
3.3	Principe de la distorsion tangentielle [26]	22
3.4	Définition de la distance focale et du champ de vision	23
3.5	Convention de distance utilisée dans l'application	25
3.6	lignes 29 à 33 de orthorectificationComputation()	26
3.7	Image initiale	27
3.8	Image orthorectifiée	27
3.9	Vue orthogonale	27
3.10	Image originale	28
3.11	Application d'un filtre CLAHE	28
3.12	Image originale	29
3.13	Application d'un filtre Intensity capping (paramètre fixé à 190)	29
3.14	Image originale	30
3.15	Application d'un filtre Intensity high-pass (paramètre fixé à 10)	30
3.16	Image originale	31
3.17	Application d'un filtre mean denoising avec les paramètres par défaut	31
3.18	Image originale	31
3.19	Application d'un filtre Contrast/Brightness (paramètres α et β fixés à 1.3 et 1) .	32
3.20	Image originale	32
3.21	Application d'un filtre Gamma avec $\gamma = 2$	32
3.22	Illustration de la technique SfM [42]	34

3.23	Comparaison entre la bathymétrie et le topographie issue du DEM. Le point blanc est le point de décrochage.	35
3.24	Recherche du niveau d'eau dans la section	36
3.25	Cas du drone positionné au centre de la largeur	37
3.26	Cas du drone décalé par rapport au centre de la rivière	38
3.27	Image A	41
3.28	Image B	41
3.29	Application de la méthode de corrélation croisée [44]	42
3.30	Représentation du pic de corrélation [45]	42
3.31	Synthèse de la méthode de la corrélation croisée [47]	43
3.32	Vecteurs vitesses avant et après remplacement des valeurs aberrantes [49]	44
3.33	Illustration du coefficient alpha	46
3.34	Variation de l'index de vitesse α pour trois valeurs de hauteur d'eau dans cinq milieux différents. Malgré le changement de milieu, l'index de vitesse reste compris entre 0,789 et 0,928 [51]	47
3.35	Subdivision de l'aire mouillée pour le calcul du débit	48
4.1	Structure des documents de l'application	50
4.2	Flowchart initial	51
4.3	Choix de la méthode pour le chargement de la bathymétrie	52
4.4	Choix du fichier contenant la bathymétrie	52
4.5	Affichage de la bathymétrie choisie	53
4.6	Fenêtre permettant à l'utilisateur de choisir le profil de bathymétrie représentatif	53
4.7	Fenêtre permettant à l'utilisateur de choisir une vidéo	54
4.8	Fenêtre qui apparaît lors de la stabilisation	55
4.9	Choix de la méthode de calibration	55
4.10	Chargement du fichier contenant les paramètres de calibration	56
4.11	Chargement de la vidéo et des caractéristiques du damier	56
4.12	Fenêtre affichée à l'orthorectification	57
4.13	Chargement des longueurs entre les points de contrôle	57
4.14	Choix de la méthode pour déterminer le niveau d'eau	58
4.15	Fenêtre qui permet de charger le DEM	58
4.16	Fenêtre qui permet de charger le DEM	59
4.17	Fenêtre de la méthode Angle et Hauteur	59
4.18	Fenêtre de la méthode Image	60
4.19	Fenêtre affichée à l'étape du ROI et du masque	60
4.20	Fenêtre permettant la sélection des filtres	61
4.21	Choix des paramètres de l'analyse PIV	62
4.22	Chargement de l'analyse de vitesses	62
4.23	Affichage des vecteurs vitesse moyens calculés	63
4.24	Fenêtre où l'utilisateur choisit la section d'intérêt	63
4.25	Choix du nombre de sous sections	64
4.26	Affichage des vecteurs vitesses perpendiculaires à la section choisie	64
4.27	Chargement du coefficient α	65
4.28	Fenêtre affichant le débit	65
5.1	Exemple de tests identiques réalisés sur le jeu 1	69
6.1	Erreur de step sur le Jeu 2	72
6.2	Illustration du problème de vitesse dans le jeu 1	72
6.3	Illustration du problème de vitesse dans le jeu 3	73
6.4	Illustration du problème de vitesse dans le jeu 6	73

6.5	Profil de vitesse à la surface attendu d'un écoulement	73
6.6	Profil de vitesse - Jeu 1	74
6.7	Erreur causée par l'implémentation du masque	74
6.8	Affichage de la largeur et des vitesses sur le jeu 2	75
6.9	Méthode image : largeur non détectée dans la section	76
6.10	Bathymétrie du jeu 1	77
7.1	Vecteurs vitesses calculés et affichés correctement	81
7.2	Cas du drone centré	81
7.3	Cas du drone décalé par rapport au centre de la rivière	82
7.4	Erreur causée par l'implémentation du masque	83
7.5	Résultat après correction	83
8.1	Popup qui apparaît si aucun fichier n'est sélectionné	85
8.2	Popup qui apparaît si aucune valeur n'est fournie	86
8.3	Popup qui apparaît si tous les paramètres pour l'orthorectification ne sont pas correctement sélectionnés	86
8.4	Popup qui apparaît si l'utilisateur choisit la méthode "Image" sans avoir réalisé d'orthorectification	86
8.5	Popup qui apparaît si la largeur choisie n'est pas détectée dans la section	87
8.6	Popup qui apparaît si l'utilisateur choisit la methode "Angle et hauteur" sans avoir réalisé de calibration	87
8.7	Fenêtre avertissant que le filtre n'a pas été sélectionné	87
8.8	Fenêtre confirmant l'application du filtre	88
8.9	Fenêtre d'encodage des paramètres de calibration	88
8.10	Méthode ajoutée pour encodé la largeur d'eau	89
8.11	Aide à l'utilisateur pour le choix d'un filtre	89
8.12	Filtres les plus adaptés à chaque vidéo	90
8.13	Histogramme des intensités des pixels avant et après l'application du filtre CLAHE	90
8.14	Limelette avant l'application du filtre High-pass	91
8.15	Limelette après l'application du filtre High-pass (facteur d'amplification égale à 200)	91
8.16	Jeul avant l'application du filtre High-pass	91
8.17	Jeul après l'application du filtre High-pass (facteur d'amplification égal à 200) .	92
8.18	Rosières avant l'application du filtre gamma	92
8.19	Rosières après l'application du filtre gamma (paramètre égal à 2)	93
8.20	Premier passage : aire de recherche (en bleu) avec un chevauchement de 50 % et vecteurs vitesses calculés sur les bords et au centre [54].	94
8.21	Second passage : la taille de l'aire de recherche (en rouge) a été réduite et les vecteurs issus du premier passage (gris et bleus) ont été interpolés [54].	94
8.22	Déformation de l'aire de recherche [16]	95
8.23	Choix du type d'analyse PIV dans l'application	95
8.24	Analyse multi-pass sur le jeu 2	96
9.1	Géométrie du tronçon du jeu 3	97
9.2	Tronçon du jeu 4	98
9.3	Résultats des tests sur le jeu 4	98
9.4	Premières secondes : tous les points de contrôle ne sont pas visibles	98
9.5	Tous les points de contrôle sont maintenant visibles	98
9.6	Localisation des trois sites visités	99
9.7	Vidéo filmée depuis la berge	99
9.8	Vidéo filmée depuis le pont	99
9.9	Vidéo filmée perpendiculairement à la surface d'eau	100

9.10	Vidéo filmée depuis le pont	100
9.11	Vidéo filmée perpendiculairement à la surface d'eau	100
9.12	Berges sur le site de Limelette	101
9.13	Vidéo filmée depuis la berge à Rosières	101
9.14	Vidéo filmée perpendiculairement au plan d'eau	101
9.15	Tests effectués avec l'ordinateur 1	102
9.16	Tests effectués avec l'ordinateur 2	102
9.17	Paramètres fixes A	103
9.18	Tests série A	103
9.19	Débits calculés sur 4 vidéos	103
9.20	Tests supplémentaires sur la série A	103
9.21	Tests supplémentaires sur la série B	103
9.22	Tests supplémentaires sur la série D	104
9.23	Cas de Limelette sans particule ajoutée	104
9.24	Cas de Limelette avec particules	104
9.25	Tests des différents filtres sur Limelette	105
9.26	Tests sur Limelette avec les paramètres variables sans filtre	105
9.27	Cas de Noirath sans particule ajoutée	105
9.28	Cas de Noirhat avec particules	105
9.29	Tests des différents filtres sur Noirhat	105
9.30	Jeu 1 sans particule ajoutée	106
9.31	Jeu 1 avec particules	106
9.32	Tests des différents filtres sur le Jeu 1	106
9.33	Limelette - Hauteur d'eau en fonction de la largeur d'eau	107
9.34	Limelette - Débit en fonction de la hauteur d'eau	107
9.35	Noirhat - Hauteur d'eau en fonction de la largeur d'eau	107
9.36	Noirhat - Débit en fonction de la hauteur d'eau	107
9.37	Noirhat - Hauteur d'eau en fonction de la largeur d'eau	108
9.38	Noirhat - Débit en fonction de la hauteur d'eau	108
9.39	Détermination de l'angle γ acceptable	109
9.40	Vidéo filmée depuis le pont	110
9.41	Vidéo filmée perpendiculairement à la surface d'eau	110
9.42	Résultats de la vidéo qui nécessite une orthorectification	111
9.43	Résultats de la vidéo qui ne nécessite pas d'orthorectification	111
9.44	Position des deux caméras sur le site de Noirhat	111
9.45	Angle de vue de la caméra 1	112
9.46	Angle de vue de la caméra 2	112
9.47	Résultats de la caméra 1	112
9.48	Résultats de la caméra 2	112
9.49	Jeu 1 - $Q = 1,26 \text{ m}^3/s$	113
9.50	Noirhat - $Q = 0,203 \text{ m}^3/s$	113
9.51	Limelette - $Q = 1,403 \text{ m}^3/s$	113
9.52	Résultats des tests sur l'influence du ROI	113
9.53	Limelette - $Q = 1,403 \text{ m}^3/s$	114
9.54	Rosières - $Q = 0,664 \text{ m}^3/s$	114
9.55	Jeu 7 - $Q = 655 \text{ m}^3/s$	114
9.56	Résultats des tests sur l'influence de la section d'intérêt	114
9.57	Tests généraux	115
10.1	Flowchart final	118
A.1	Tests sur la méthode de validation sur Limelette	132

A.2	Tests sur la méthode de validation sur Noirhat	132
A.3	Tests sur la méthode de validation sur Limelette	133
A.4	Tests sur la méthode de validation sur Noirhat	133
A.5	Tests de la méthode image sur Limelette	134
A.6	Tests de la méthode image sur Noirhat	135
A.7	Tests de la méthode image sur le Jeu 1	136
A.8	Tests sur l'influence de l'orthorectification	137
A.9	Tests sur l'influence de l'angle de prise de vue sur l'orthorectification	138
A.10	Tests sur l'influence du ROI et du masque sur le jeu 1	139
A.11	Tests sur l'influence du ROI et du masque sur Limelette	140
A.12	Tests sur l'influence du ROI et du masque sur Noirhat	141
A.13	Tests sur l'influence de la section d'intérêt à Limelette	142
A.14	Tests sur l'influence de la section d'intérêt à Rosières	143
A.15	Tests sur l'influence de la section d'intérêt sur le jeu 7	144
A.16	Tests aléatoires sur l'application	145

Liste des tableaux

9.1	Valeurs extrêmes	108
9.2	Pente moyenne des bathymétrie	108
9.3	Angles γ acceptables	109
9.4	Débits moyens calculés sur 20 tests	110

Bibliographie

- [1] Kirk G. Thibodeaux JANICE M. FULFORD et William R. KAEHRLE. « Comparison of current meters used for stream gaging ». In : (), p. 10. DOI : 10.1051/lhb/2014024ff.ffhal-01118478.
- [2] Daubagnan L. HAUET A. Morlot T. « Velocity profile and depth-averaged to surface velocity in natural streams : A review over a large sample of rivers ». In : *E3S Web Conf.* 40 (2018), p. 6-15. DOI : 10.1051/e3sconf/20184006015.
- [3] ANONYME. *Mesures d'écoulements dans les cours d'eau*. sur https://nanopdf.com/download/mesures-decoulements-dans-les-cours-deau_pdf#. consulté le 07/05/2021.
- [4] ANONYME. *Le suivi limnimétrique*. sur <https://davar.gouv.nc/ressource-en-eau-mesures-et-etudes-reseau-de-mesures/le-suivi-limnimetrique>. consulté le 07/05/2021.
- [5] ANONYME. *Courantomètre à induction magnétique OTT MF pro*. sur <https://www.ott.com/fr-fr/produits/le-debit-deau-70/ott-mf-pro-177/>. consulté le 07/05/2021.
- [6] ANONYME. *Mesures de débit*. sur http://wikhydro.developpement-durable.gouv.fr/index.php/Mesures_de_d%C3%A9bit#:~:text=Le%20courantom%C3%A8tre%20%C3%A9lectromagn%C3%A9tique%20permet%20%C3%A9galement,%C3%A0%20la%20vitesse%20d'%C3%A9coulement.. consulté le 05/05/2021.
- [7] ANONYME. *HydroSurveyor ADCP Rental*. sur <https://www.fondriest.com/hydrosurveyor-adcp-rental.htm>. consulté le 07/05/2021.
- [8] Rajesh SHRESTHA et Slobodan SIMONOVIC. « A Fuzzy Set Theory Based Methodology for Analysis of Uncertainties in Stage-Discharge Measurements and Rating Curve ». In : (jan. 2009).
- [9] Ronald J. ADRIAN. *Twenty years of particle image velocimetry*. sur https://www.researchgate.net/publication/226240853_Twenty_years_of_particle_image_velocimetry. consulté le 11/05/2021.
- [10] LAPORTA A. et MARECHAL J.P. *La vélocimétrie par images de particules - Principes et premiers résultats*. sur https://inis.iaea.org/collection/NCLCollectionStore/_Public/29/015/29015338. consulté le 10/05/2021.
- [11] Yvan Bercovity et al. MAGALI JODEAU. *Laboratory and field LSPIV measurements of flow velocities using Fudaa-LSPIV a free user-friendly software*. sur https://www.researchgate.net/publication/315684488_Laboratory_and_field_lsipiv_measurements_of_flow_velocities_using_Fudaa-LSPIV_a_free_user-friendly_software. consulté le 10/05/2021.
- [12] Le Coz J. Marchand B. Die Moran A. et al. HAUET A. Jodeau M. « Application de la méthode LSPIV pour la mesure de champs de vitesse et de débits de crue sur modèle réduit et en rivière. » In : *La Houille Blanche - Revue internationale de l'eau* 3 (2014), p. 16-22. DOI : 10.1051/lhb/2014024ff.ffhal-01118478.

- [13] Bodart G. et AL. « Manuel d'utilisation du logiciel Fudaa-LSPIV Version 1.7.3 ». In : (2020).
- [14] Thielicke W. *PIVlab*. sur <http://william.thielicke.org/>. consulté le 01/04/2021.
- [15] Thielicke W. *PIVlab - particle image velocimetry (PIV) tool with GUI*. sur <https://nl.mathworks.com/matlabcentral/fileexchange/27659-pivlab-particle-image-velocimetry-piv-tool-with-gui>. consulté le 01/04/2021. 2014.
- [16] Thielicke William. et AL. « PIVlab – Towards User-friendly, Affordable and Accurate Digital Particle Image Velocimetry in MATLAB ». In : *Journal of Open Research Software* 2 (oct. 2014). DOI : 10.5334/jors.bl.
- [17] Phototrack AG. *Discharge, platform for professional surface flow measurements*. sur <https://discharge.ch/>. consulté le 05/04/2021.
- [18] Phototrack AG. *Discharge, user manual*. sur https://app.discharge.ch/res/discharge_manual_20170131.pdf. consulté le 05/04/2021.
- [19] João LEITÃO et al. « Urban overland runoff velocity measurement with consumer-grade surveillance cameras and surface structure image velocimetry ». In : *Journal of Hydrology* 565 (sept. 2018). DOI : 10.1016/j.jhydro.2018.09.001.
- [20] ANONYME. *RIVeR Tutorial*. sur <https://riverdischarge.blogspot.com/p/tutorial.html>. consulté le 12/04/2021.
- [21] ANONYME. *How to process a video with RIVeR Software*. sur <https://my.usgs.gov/confluence/display/SurfBoard/How+to+process+a+video+with+RIVeR+Software>. consulté le 12/04/2021.
- [22] Mallick S. *Object Tracking using OpenCV (C++/Python)*. sur <https://learnopencv.com/object-tracking-using-opencv-cpp-python/>. consulté le 07/03/2021.
- [23] Sadekar K. et AL. *Camera Calibration using OpenCV*. sur <https://learnopencv.com/camera-calibration-using-opencv/>. consulté le 09/03/2021.
- [24] ANONYME. *Camera calibration*. sur https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html. consulté le 10/03/2021.
- [25] Dong D. « Correction de la distorsion radiale : une approche non-linéaire. » In : ().
- [26] Vincent NOZICK. « Video-Based Rendering and Virtual Reality ». In : (juin 2006).
- [27] ANONYME. *Camera calibration With OpenCV*. sur https://docs.opencv.org/3.4/d4/d94/tutorial_camera_calibration.html. consulté le 10/03/2021.
- [28] Renard S. « Development of an application for river discharge measurement ». In : (2019-2020).
- [29] Olivier HAGOLLE. *L'ortho-rectification, comment ça marche ?* sur <https://labo.obs-mip.fr/multitemp/lortho-rectification-comment-ca-marche/>. consulté le 2/04/2021.
- [30] Larry ZITNICK. *Geometric Transformations*. sur <https://fddocuments.in/document/geometric-transformations-cse-p-576-larry-zitnick-larryzmicrosoftcom.html>. consulté le 4/04/2021.
- [31] Robert COLLINS. *Lecture 16 : Planar Homographies*. sur <http://www.cse.psu.edu/~rtc12/CSE486/lecture16.pdf>. consulté le 4/04/2021.
- [32] Sanja FIDLER. *Matching Planar Objects In New Viewpoints ... And Much More - via Homography*. sur <http://www.cs.toronto.edu/~fidler/slides/2017/CSC420/lecture9.pdf>. consulté le 4/04/2021.

- [33] Patalano et AL. « Rectification of Image Velocity Results (RIVeR) : A simple and user-friendly toolbox for large scale water surface Particle Image Velocimetry (PIV) and Particle Tracking Velocimetry (PTV) ». In : *Computers and Geosciences* 109 (déc. 2017), p. 323-330. DOI : 10.1016/j.cageo.2017.07.009.
- [34] ANONYME. *Open CV Histograms - 2 : Histogram Equalization*. sur https://docs.opencv.org/master/d5/daf/tutorial_py_histogram_equalization.html. consulté le 14/03/2021.
- [35] Shavit U. et AL. *Intensity Capping : a simple method to improve cross-correlation PIV results*. sur <https://link.springer.com/article/10.1007/s00348-006-0233-7>. consulté le 15/03/2021.
- [36] sur <https://www.coursehero.com/file/p25h1o1/Intensity-capping-The-DPIV-method-assumes-that-all-particles-within-an/>. consulté le 15/03/2021.
- [37] Buades A. et AL. *Non-Local Means Denoising*. sur http://www.ipol.im/pub/art/2011/bcm_nlm/. consulté le 15/03/2021.
- [38] ANONYME. *OpenCV : Operations on arrays*. sur https://docs.opencv.org/3.4/d2/de8/group__core__array.html#ga3460e9c9f37b563ab9dd550c4d8c4e7d. consulté le 16/03/2021.
- [39] Hellebaut F.X. « Mesure du niveau d'eau par photogrammétrie à partir d'un drone. » In : (2017-2018).
- [40] Schenk T. « Introduction to Photogrammetry ». In : *Department of Civil and Environmental Engineering and Geodetic Science The Ohio State University* (2005), p. 3.
- [41] Carrivick J.L. SMITH M.W. « Structure from motion photogrammetry in physical geography ». In : 40 (2016), p. 247-275. DOI : 10.1177/0309133315615805.
- [42] Austin MASON. *Making 3D Models with Photogrammetry*. sur <https://thehaskinssociety.wildapricot.org/photogrammetry>. consulté le 24/04/2021.
- [43] T. Hadad R. GRUKA. « Effects of particle size, concentration and surface coating on turbulent flow properties obtained using PIV/PTV ». In : (2012).
- [44] Jérôme LEMELIN. *Vélocimétrie par images de particules*. sur <https://polrismer.blog/2020/05/08/velocimetrie-par-images-de-particules/>. consulté le 15/05/2021.
- [45] Tristan BIA HAROLD SOMERS. *Mesure du champ de vitesses superficielles en rivière à partir d'un drone*. sur <https://dial.uclouvain.be/memoire/ucl/en/object/thesis%3A14838>. consulté le 2/04/2021.
- [46] Rob J.M. BASTIAANS. *Cross-correlation PIV; theory, implementation and accuracy*. sur https://www.researchgate.net/publication/241849539_Cross-correlation_PIV_theory_implementation_and_accuracy. consulté le 15/05/2021.
- [47] Loic Jouan DE KERVENOAEL. *Validation des mesures par vélocimétrie par image de particules dans une conduite de section rectangulaire*. sur <https://corpus.ulaval.ca/jspui/bitstream/20.500.11794/20570/1/25982>. consulté le 15/05/2021.
- [48] Liberzon A. *OpenPIV-python*. sur <https://github.com/OpenPIV/openpiv-python>. consulté le 27/04/2021.
- [49] Liberzon A. *OpenPIV 0.0.1 documentation : how to process an image pair*. sur <http://alexlib.github.io/openpiv-python/tutorial.html>. consulté le 27/04/2021.
- [50] Cheng R.T. Haeni F.P. Melcher N.B. Thurman E.M. Plant W.J. et Keller W.C COSTA J.E. Spicer K.R. « Measuring stream discharge by non-contact methods : a proof-of-concept experiment. » In : (2000), p. 553-556.

- [51] Polatel C. « Indexing by free surface velocity - a prospect for remote discharge estimation. » In : (2005), p. 1453-1454.
- [52] Laronne J. Zolezzi G. Zamler D. et al WELBER M. Le Coz J. « Field assessment of noncontact stream gauging using portable surface velocity radars (SVR). » In : (2017), p. 1108-1126.
- [53] Liberzon A. *OpenPIV Python version with GPU support*. sur <https://github.com/OpenPIV/openpiv-python-gpu>. consulté le 13/05/2021.
- [54] David BOUTELIER, Christoph SCHRANK et Klaus REGENAUER-LIEB. « 2-D finite displacements and strain from particle imaging velocimetry (PIV) analysis of tectonic analogue models with TecPIV ». In : *Solid Earth* 10 (juil. 2019), p. 1123-1139. DOI : 10.5194/se-10-1123-2019.

Annexe A

Tests de validation

1 Influence de la méthode de validation de l'analyse PIV

A) Limelette									
nom video	terrain_limelette_vide o1_camera1.MOV								
range	11->14								
Step	3								
Stabilisation	NON								
Calibration	NON								
Orthorectification	OUI/FIXE								
Hauteur d'eau	0,478								
Filtre	CLAHE								
Scaling factor	FIXE								
ROI / Mask	FIXE								
Section d'intérêt	FIXE								
nbre de sous section	20								
Alpha	0,85								
Débit attendu	1,403								
	Arg. 1	Arg. 1	Arg. 1	Arg. 1	Arg. 1	Parametres FIXE A.1	Parametres FIXE A.2	Parametres FIXE A.3	
Sig2noise	u0	v0	sig2noise	threshold=SNR	-	Débit	Débit	Débit	
Median_val	u0	v0	3	3	size=1	1,2442	1,1411	1,2605	
global_std	u0	v0	threshold = 3	-	-	1,0901	0,9621	1,171	
global_val	u0	v0	(-150,150)	(-150,150)	-	1,0468	0,8303	1,0835	
typical	u0	v0	sig2noise	settings	-	1,0397	0,8279	1,0885	
						0,6017	0,6998	0,5662	

FIGURE A.1 – Tests sur la méthode de validation sur Limelette

B) Noirhat									
nom video	terrain_noirhat_video1 _camera2.mp4								
range	7->10								
Step	3								
Stabilisation	OUI								
Calibration	NON								
Orthorectification	OUI								
Hauteur d'eau	0,155								
Filtre	Gamma - 2								
Scaling factor	Fixé à xxx								
ROI / Mask	FIXE								
Section d'intérêt	FIXE								
nbre de sous section	20								
Alpha	0,85								
Débit attendu	0,203								
	Arg. 1	Arg. 1	Arg. 1	Arg. 1	Arg. 1	Parametres FIXE B.4	Parametres FIXE B.5	Parametres FIXE B.6	
Sig2noise	u0	v0	sig2noise	threshold=SNR	-	Débit	Débit	Débit	
Median_val	u0	v0	3	3	size=1	0,1856	0,1787	0,1951	
global_std	u0	v0	threshold = 3	-	-	0,1867	0,1785	0,196	
global_val	u0	v0	(-150,150)	(-150,150)	-	0,1778	0,1778	0,1901	
typical	u0	v0	sig2noise	settings	-	0,1773	0,177	0,1876	
						0,0128	0,0118	0,013	

FIGURE A.2 – Tests sur la méthode de validation sur Noirhat

C) Limelette

nom video	Limelette_drone_coupe.mp4
range	3->6
Step	3
Stabilisation	NON
Calibration	NON
Orthorectification	NON
Hauteur d'eau	0,478
Filtre	CLAHE
Scaling factor	Fixé à xxx
ROI / Mask	FIXE
Section d'intérêt	FIXE
nbre de sous section	30
Alpha	0,85

	Arg. 1	Arg. 1	Arg. 1	Arg. 1	Arg. 1	Paramètres FIXE C.1	Paramètres FIXE C.2	Paramètres FIXE C.3
						Débit	Débit	Débit
Sig2noise	u0	v0	sig2noise	threshold=SNR	-	1,4152	1,4113	1,3645
Median_val	u0	v0	3	3	size=1	1,0495	0,9831	0,9605
global_std	u0	v0	threshold = 3	-	-	0,7477	0,6883	0,7033
global_val	u0	v0	(-150,150)	(-150,150)	-	0,7464	0,6918	0,6874
typical	u0	v0	sig2noise	settings	-	0,0389	0,026	0,0359

FIGURE A.3 – Tests sur la méthode de validation sur Limelette

D) Noirath

nom video	terrain_noirhat_video1_camera1.mp4
range	6->8
Step	3
Stabilisation	NON
Calibration	NON
Orthorectification	OUI
Hauteur d'eau	0,155
Filtre	CLAHE
Scaling factor	FIXE
ROI / Mask	FIXE
Section d'intérêt	FIXE
nbre de sous section	20
Alpha	0,85
Débit attendu	0,203

	Arg. 1	Arg. 1	Arg. 1	Arg. 1	Arg. 1	Paramètres FIXE D.1	Paramètres FIXE D.2	Paramètres FIXE D.3
						Débit	Débit	Débit
Sig2noise	u0	v0	sig2noise	threshold=SNR	-	0,2069	0,2037	0,2072
Median_val	u0	v0	3	3	size=1	0,1784	0,1829	0,1755
global_std	u0	v0	threshold = 3	-	-	0,1547	0,1581	0,1432
global_val	u0	v0	(-150,150)	(-150,150)	-	0,0696	0,0891	0,0551
typical	u0	v0	sig2noise	settings	-	0,0017	0,0033	0,0019

FIGURE A.4 – Tests sur la méthode de validation sur Noirhat

2 Limites dues à la forme de la bathymétrie

A) Limelette			
nom video	terrain_limelette_video1_camera1.MOV	Débit attendu	1,403 [m ³ /s]
range	11->13	Hauteur d'eau attendue	0,478 [m]
Step	3	Largeur d'eau attendue	7,45 [m]
Stabilisation	NON		
Calibration	NON		
Orthorectification	OUI/FIXE		
Hauteur d'eau	Image Method		
Filtre	SANS FILTRE		
Scaling factor	FIXE/		
ROI / Mask	FIXE		
nbre de sous section	20		
Alpha	0,85		

Test n°	Largeur d'eau	Hauteur d'eau	Débit
1	7,436	0,478	1,2413
2	7,45	0,501	1,313
3	7,281	0,38	0,9265
4	7,432	0,47	1,2162
5	7,394	0,419	1,0588
6	7,301	0,387	0,9491
7	7,479	0,548	1,4594
8	7,614	0,746	2,0912
9	7,305	0,388	0,9537
10	7,415	0,443	1,1314
11	7,193	0,343	0,8079
12	7,436	0,478	1,2416
13	7,533	0,637	1,7399
14	7,444	0,492	1,2842
15	7,415	0,443	1,1314
16	7,2	0,347	0,8208
17	7,48	0,55	1,468
18	7,481	0,552	1,4723
19	7,537	0,643	1,7617
20	7,42	0,452	1,161

FIGURE A.5 – Tests de la méthode image sur Limelette

B) Noirhat

nom video	terrain_noirhat_video1_camera1.MOV
range	6->8
Step	3
Stabilisation	NON
Calibration	NON
Orthorectification	OUI/FIXE
Hauteur d'eau	Image Method
Filtre	Gamma - 2
Scaling factor	FIXE/
ROI / Mask	FIXE
nbre de sous section	20
Alpha	0,85

Débit attendu	0,203 [m ³ /s]
Hauteur d'eau attendue	0,155 [m]
Largeur d'eau attendue	3,58 [m]

Test n°	Largeur d'eau	Hauteur d'eau	Débit
1	3,4499	0,155	0,2089
2	3,58	0,222	0,3256
3	3,464	0,16	0,2172
4	3,356	0,112	0,1366
5	3,514	0,182	0,2554
6	3,358	0,114	0,1392
7	3,426	0,143	0,1887
8	3,384	0,125	0,1577
9	3,374	0,12	0,1497
10	3,48	0,167	0,2292
11	3,32	0,097	0,1115
12	3,43	0,145	0,1914
13	3,392	0,128	0,1629
14	3,464	0,16	0,2172
15	3,366	0,117	0,1445
16	3,563	0,212	0,3072
17	3,514	0,182	0,2554
18	3,478	0,166	0,2281
19	3,375	0,12	0,1506
20	3,462	0,159	0,2163

FIGURE A.6 – Tests de la méthode image sur Noirhat

c) Jeu 1

nom video	video_jeu_1.MP4
range	29->32
Step	3
Stabilisation	NON
Calibration	NON
Orthorectification	OUI/FIXE
Hauteur d'eau	Image Method
Filtre	SANS FILTRE
Scaling factor	FIXE
ROI / Mask	FIXE
nbre de sous section	20
Alpha	0,85

Débit attendu	1,26 [m ³ /s]
Hauteur d'eau attendue	0,44 [m]
Largeur d'eau attendue	- [m]

Test n°	Largeur d'eau	Hauteur d'eau	Débit
1	5,7645	0,44	1,2697
2	5,863	0,676	1,668
3	5,885	0,728	1,7571
4	5,779	0,476	1,3294
5	5,619	0,096	0,7107
6	5,648	0,165	0,8208
7	5,833	0,604	1,5452
8	5,78	0,479	1,3342
9	5,779	0,477	1,3318
10	5,805	0,538	1,4332
11	5,699	0,286	1,0161
12	5,645	0,158	0,8093
13	5,887	0,734	1,7671
14	5,752	0,412	1,2241
15	5,888	0,735	1,7696
16	5,779	0,477	1,3318
17	5,86	0,669	1,6557
18	5,809	0,548	1,4502
19	5,7	0,287	1,0184
20	5,681	0,243	0,946

FIGURE A.7 – Tests de la méthode image sur le Jeu 1

3 Influence de l'étape d'orthorectification

Limelette	
Avec orthorectification	
nom video	terrain_limelette_video1_camera1.MOV
range	11->12
Step	3
Stabilisation	NON
Calibration	NON
Orthorectification	OUI
Hauteur d'eau	0,478
Filtre	CLAHE
ROI / Mask	FIXE
Section d'intérêt	Variable
nbre de sous sections	20
Alpha	0,85
Sans orthorectification	
nom video	Limelette_drone_coupe.mp4
range	3->6
Step	3
Stabilisation	NON
Calibration	NON
Orthorectification	NON
Hauteur d'eau	0,478
Filtre	CLAHE
ROI / Mask	FIXE
Section d'intérêt	Variable
nbre de sous sections	20
Alpha	0,85

Test n°	Débit
1	1,2108
2	1,209
3	1,1196
4	1,1333
5	1,0955
6	1,0878
7	1,134
8	1,106
9	1,0841
10	1,2074
11	1,1366
12	1,087
13	1,1292
14	1,1381
15	1,1775
16	1,1933
17	1,1469
18	1,1414
19	1,1317
20	1,1277
Moyenne	1,139845
Erreur	19%

Test n°	Débit
1	1,4082
2	1,3717
3	1,355
4	1,3793
5	1,3961
6	1,4379
7	1,3899
8	1,3715
9	1,3404
10	1,384
11	1,4298
12	1,3796
13	1,3422
14	1,397
15	1,378
16	1,4116
17	1,3909
18	1,4085
19	1,4144
20	1,3799
Moyenne	1,388295
Erreur	1%

FIGURE A.8 – Tests sur l'influence de l'orthorectification

Noirhat			
Angle 1		Angle 2	
nom video	terrain_noirhat_video1_camera1.MOV	nom video	terrain_noirhat_video1_camera2.MOV
range	6->8	range	7->10
Step	3	Step	3
Stabilisation	NON	Stabilisation	NON
Calibration	NON	Calibration	NON
Orthorectification	OUI	Orthorectification	OUI
Hauteur d'eau	0,155	Hauteur d'eau	0,155
ROI / Mask	FIXE	ROI / Mask	FIXE
Filtre	CLAHE	Filtre	CLAHE
Section d'intérêt	Libre	Section d'intérêt	Libre
nbre de sous sections	20	nbre de sous sections	20
Alpha	0,85	Alpha	0,85

Test n°	Débit
1	0,2041
2	0,2055
3	0,209
4	0,2168
5	0,2069
6	0,2107
7	0,217
8	0,21
9	0,2208
10	0,2077
11	0,21
12	0,2118
13	0,2062
14	0,217
15	0,2084
16	0,2087
17	0,2207
18	0,208
19	0,2029
20	0,2183
Moyenne	0,2110
Erreur	4%

Test n°	Débit
1	0,2037
2	0,2016
3	0,2042
4	0,2055
5	0,2057
6	0,2056
7	0,2068
8	0,2033
9	0,2003
10	0,202
11	0,2069
12	0,207
13	0,2039
14	0,2053
15	0,2034
16	0,2018
17	0,2081
18	0,1987
19	0,2042
20	0,205
Moyenne	0,2042
Erreur	1%

FIGURE A.9 – Tests sur l'influence de l'angle de prise de vue sur l'orthorectification

4 Influence du ROI

A) Jeu 1

nom video	video_jeu_1.MP4
range	29->32
Step	3
Stabilisation	NON
Calibration	NON
Orthorectification	OUI
Hauteur d'eau	0,44
Filtre	Sans Filtre
Scaling factor	FIXE
ROI / Mask	Variable
Section d'intérêt	FIXE
nbre de sous sections	20
Alpha	0,85

Débit attendu	1,26	[m^3/s]
Hauteur d'eau attendue	0,44	[m]
Largeur d'eau attendue	-	[m]

Moyenne	Erreur produite
1,30752	4%

Test	Débit min	Erreur produite
7	1,1448	10,06%

Test	Débit max	Erreur produite
10	1,4646	14%

Test n°	Débit
1	1,2734
2	1,3536
3	1,1992
4	1,3715
5	1,2723
6	1,2211
7	1,1448
8	1,3143
9	1,3817
10	1,4646
11	1,2969
12	1,3319
13	1,355
14	1,2841
15	1,4575
16	1,3668
17	1,1696
18	1,3077
19	1,225
20	1,3594

FIGURE A.10 – Tests sur l'influence du ROI et du masque sur le jeu 1

B) Limelette		
nom video	Limelette_drone_coupe.mp4	
range	3->6	
Step	3	
Stabilisation	NON	
Calibration	NON	
Orthorectification	NON	
Hauteur d'eau	0,478	
Filtre	CLAHE	
Scaling factor	Variable	
ROI / Mask	Variable	
Section d'intérêt	Variable	
nbre de sous section	20	
Alpha	0,85	

Débit attendu	1,403	[m ³ /s]
Hauteur d'eau attendue	0,478	[m]
Largeur d'eau attendue	7,45	[m]

Test n°	Débit
1	1,4082
2	1,3596
3	1,2853
4	1,3159
5	1,3686
6	1,3066
7	1,362
8	1,3501
9	1,3779
10	1,3063
11	1,4252
12	1,3454
13	1,3417
14	1,3455
15	1,3057
16	1,3707
17	1,3833
18	1,3438
19	1,3772
20	1,3546

Moyenne	Erreur produite
1,35168	4%

Test	Débit min	Erreur produite
3	1,2853	9,16%

Test	Débit max	Erreur produite
11	1,4252	2%

FIGURE A.11 – Tests sur l'influence du ROI et du masque sur Limelette

C) Noirhat		
nom video	terrain_noirhat_video1_camera1,MOV	
range	6->8	
Step	3	
Stabilisation	NON	
Calibration	NON	
Orthorectification	OUI	
Hauteur d'eau	0,155	
Filtre	Gamma - 2	
Scaling factor	FIXE	
ROI / Mask	Variable	
Section d'intérêt	FIXE	
nbre de sous section	20	
Alpha	0,85	

Débit attendu	0,203	[m ³ /s]
Hauteur d'eau attendue	0,155	[m]
Largeur d'eau attendue	3,58	[m]

Test n°	Débit
1	0,2032
2	0,2067
3	0,2009
4	0,2028
5	0,2023
6	0,2129
7	0,2114
8	0,2005
9	0,2024
10	0,2042
11	0,2057
12	0,2081
13	0,2102
14	0,2028
15	0,206
16	0,2025
17	0,2002
18	0,2095
19	0,2044
20	0,2112

Moyenne	Erreur produite
0,205395	1,38%

Test	Débit min	Erreur produite
17	0,2002	5%

Test	Débit max	Erreur produite
6	0,2129	1%

FIGURE A.12 – Tests sur l'influence du ROI et du masque sur Noirhat

5 Influence de la section d'intérêt sans orthorectification

A) Limelette		
nom video	Limelette_drone_coupe.mp4	
range	3->6	
Step	3	
Stabilisation	NON	
Calibration	NON	
Orthorectification	NON	
Hauteur d'eau	0,478	
Filtre	CLAHE	
Scaling factor	Variable	
ROI / Mask	FIXE	
nbre de sous sections	20	
Alpha	0,85	
Débit attendu	1,403	[m ³ /s]
Hauteur d'eau attendue	0,478	[m]
Largeur d'eau attendue	7,45	[m]
	Moyenne	Erreur produite
	1,383405	1%
Test	Débit min	Erreur produite
16	1,3287	5,59%
Test	Débit max	Erreur produite
1	1,434	2%
Test n°	Scaling factor	Débit
1	151,1863	1,434
2	154,02254	1,3529
3	151,182337	1,3915
4	150,6629	1,3985
5	151,441586	1,4192
6	151,438278	1,3915
7	151,181619	1,3999
8	148,595869	1,4228
9	150,1473145	1,4284
10	151,7044091	1,3424
11	152,732159	1,3361
12	153,505689	1,3584
13	153,506342	1,376
14	152,2188226	1,421
15	150,9353643	1,378
16	154,79809	1,3287
17	154,79809	1,3311
18	150,923196	1,3947
19	153,2497144	1,3799
20	153,250531	1,3831

FIGURE A.13 – Tests sur l'influence de la section d'intérêt à Limelette

B) Rosières

nom video	terrain_rosière_video1_coupe
range	16->18
Step	3
Stabilisation	NON
Calibration	NON
Orthorectification	NON
Hauteur d'eau	0,64
Filtre	Sans filtre
Scaling factor	Variable
ROI / Mask	FIXE
nbre de sous sections	20
Alpha	0,92

Débit attendu	0,664	[m^3/s]
Hauteur d'eau attendue	0,64	[m]
Largeur d'eau attendue	5,48	[m]

	Moyenne	Erreur produite
	0,575375	13%

Test	Débit min	Erreur produite
7	0,5043	31,67%

Test	Débit max	Erreur produite
5	0,6469	3%

Test n°	Scaling factor	Débit
1	213,4845	0,6042
2	218,988705	0,5501
3	216,236095	0,5596
4	216,558572	0,5784
5	205,39279	0,6469
6	220,784196	0,6011
7	222,169488	0,5043
8	224,637679	0,5883
9	222,769922	0,5701
10	220,972414	0,5709
11	222,958254	0,5412
12	220,582891	0,5533
13	220,519383	0,5797
14	218,963443	0,5397
15	223,367805	0,551
16	221,420669	0,5832
17	213,98078	0,5981
18	215,0354288	0,615
19	214,779887	0,5884
20	216,3376	0,584

FIGURE A.14 – Tests sur l'influence de la section d'intérêt à Rosières

C) Jeu 7		
nom video	Jeu7	
range	0->2	
Step	3	
Stabilisation	NON	
Calibration	NON	
Orthorectification	NON	
Hauteur d'eau	0	
Filtre	CLAHE	
Scaling factor	Variable	
ROI / Mask	FIXE	
nbre de sous sections	40	
Alpha	0,95	

Débit attendu	655,087367 [m ³ /s]	
	Moyenne	Erreur produite
	682,236	4%

Test	Débit min	Erreur produite
19	645,4216	1,5%

Test	Débit max	Erreur produite
15	719,0526	9%

Test n°	SF	Débit
1	20,71	678,0794
2	20,79	659,0512
3	20,53	664,7285
4	20,71	676,6418
5	20,68	683,8241
6	20,65	700,9478
7	20,76	688,5133
8	20,65	685,9442
9	20,87	676,7918
10	20,81	675,1744
11	20,65	711,0854
12	20,63	680,485
13	20,76	679,1889
14	20,73	685,6017
15	20,65	719,0526
16	20,68	692,7381
17	20,76	686,6146
18	20,52	674,5301
19	20,81	645,4216
20	20,73	680,3055

FIGURE A.15 – Tests sur l'influence de la section d'intérêt sur le jeu 7

6 Précision générale

7) Précision générale de l'application

Tests	Vidéo	Q_attendu	Range	Step	Stab	Calib	Ortho	L [m]	WL [m]	Filtre(s)	SF	nb subsection	alpha	Q [m³/s]	Erreur
1	terrain_limelette_video1_camera1.MOV	1,403	11->14	3	NON	NON	OUI	7,436	0,478	CLAHE	150,2315	20	0,85	1,2605	11%
2	Limelette_drone_coupe.mp4	1,403	3->6	3	NON	NON	NON	-	0,478	CLAHE	151,441586	20	0,85	1,4192	1%
3	terrain_rosière_video1_coupe	0,664	16->18	3	NON	NON	NON	-	0,64	SANS FILTRE	213,4845	20	0,92	0,6042	10%
4	Jeu7	655,087367	0->2	3	NON	NON	NON	-	0	CLAHE	20,79	40	0,95	659,0512	1%
5	terrain_noirhat_video1_camera1.mp4	0,203	6->8	3	NON	NON	OUI	3,4499	0,155	CLAHE	309,8599	20	0,85	0,2072	2%
6	terrain_limelette_video1_camera1.MOV	1,403	11->12	3	NON	NON	OUI	7,436	0,478	CLAHE	150,26	20	0,85	1,2108	16%
7	Limelette_drone_coupe.mp4	1,403	3->6	3	NON	NON	NON	-	0,478	CLAHE	154,62	20	0,85	1,3961	0%
8	terrain_noirhat_video1_camera2.mp4	0,203	7->10	3	OUI	NON	OUI	3,4499	0,155	Gamma - 2	205,54408	20	0,85	0,2038	0%
9	video_jeu_1.MP4	1,26	29->32	3	NON	NON	OUI	5,7645	0,44	SANS FILTRE	119,0451	20	0,85	1,2697	1%
10	terrain_noirhat_video1_camera2.mp4	0,203	7->10	3	OUI	NON	OUI	3,4499	0,155	CLAHE	205,15	20	0,85	0,2018	1%
11	terrain_noirhat_video1_camera2.mp4	0,203	7->10	3	OUI	NON	OUI	3,4499	0,155	Gamma-2	204,533708	20	0,85	0,2065	2%
12	video_jeu_1.MP4	1,26	29->32	2	NON	NON	OUI	5,7645	0,44	Gamma - 3	119,28809	30	0,85	1,1776	7%
13	Limelette_drone_coupe.mp4	1,403	3->6	1	NON	NON	NON	-	0,478	Gamma - 2	154,315979	30	0,85	1,3774	2%
14	terrain_limelette_video1_camera1.MOV	1,403	11->13	3	NON	NON	OUI	7,436	0,478	CLAHE	151,2432	30	0,85	1,2502	12%
15	terrain_noirhat_video1_camera1.mp4	0,203	6->8	2	NON	NON	OUI	3,4499	0,155	Gamma - 2	310,734577	30	0,85	0,2198	8%
16	terrain_noirhat_video1_camera2.mp4	0,203	7->10	1	OUI	NON	OUI	3,4499	0,155	CLAHE	205,354204	30	0,85	0,2165	6%
17	Jeu7	655,087367	0->2	1	NON	NON	NON	-	0	SANS FILTRE	20,2326	50	0,85	570,4706	15%
18	Limelette_drone_coupe.mp4	1,403	3->6	2	NON	NON	NON	-	0,478	CLAHE	156,2286	30	0,85	1,3864	1%
19	GOPRO3225_Trim.mp4	1,39	0->2	1	NON	NON	OUI	-	40	SANS FILTRE	88,72077	10	0,85	1,3533	3%
20	terrain_noirhat_video1_camera1.mp4	0,203	6->8	1	NON	NON	OUI	3,4499	0,155	SANS FILTRE	311,327211	20	0,85	0,2222	9%

FIGURE A.16 – Tests aléatoires sur l'application

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl