Analysis of water flows videos for surface velocityestimation

Musaab Khalid

ANNÉE 2018

UNIVERSITÉ DE RENNES 1

UNIVERSITE BRETAGNE LOIRE

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Signal, Image, Vision

École doctorale MathSTIC
Présentée par

Musaab Khalid

Préparée à l'unité de recherche HHLY, IRSTEA- Lyon-Villeurbanne
Institut national de recherche en sciences et technologies pour l'environnement et l'agriculture

---

Intitulé de la thèse: Analysis of water flow videos for surface velocity estimation

Thèse soutenue à Villeurbanne le 5 juin 2018

devant le jury composé de :

Eric MARCHAND
Professeur, Université de Rennes 1 / *Président du jury*

Salvatore GRIMALDI
Professeur, Université de Tuscia, Italy / *Rapporteur*

Thomas CORPETTI
Directeur de recherche, CNRS / *Rapporteur*

Magali JODEAU
Ingénieur chercheur expert, EDF LNHE / *Examinatrice*

Etienne MÉMIN
Directeur de recherche, INRIA / *Directeur de thèse*

Lionel PÉNARD
IPEF, IRSTEA Lyon-Villeurbanne / *Encadrant*

Alexandre HAUET
Ingénieur chercheur, EDF DTG / *Invité*

Mais à chaque coucher de soleil je me lève et jette un coup d'œil à l'horizon oriental où la première étoile scintille et je me demande: Comment se fait-il ? ... Qu'est-ce qui m'a amené sur cette planète étrange et dérangeante qu'ils appellent (la Terre) ?!

Dr. Ahmed Khaled Tawfik

Acknowledgments:

I would like to thank Prof. David Fofi and Prof. Fabrice Meriaudeau for introducing me to the field of computer vision. It was a chance for me to be in a close proximity to not just great scientists but also unique individuals. From the same university (Burgundy), I will be forever grateful to Prof. Alamin Mansouri who supervised my master's internship. I've learned alot working with him. Many many things wouldn't have been possible without his assistance and guidance.

Also, I want to thank my Ph.D. supervisor, Dr. Lionel Pénard whose constant support has been of great value, not only in the core science but also with the important aspects of writing articles and giving presentations. His attention to details was quite impressive, averaging that with my own skills, I believe I have improved alot. I particularly appreciated him also standing by my side with all administrative hurdles I had to scramble through. Thank you for that Lionel.

I honestly feel privilged having Dr. Etienne Mémin co-directing my thesis. Even if I was not in the same city as his, every time I'd go there for a week or so, I come back my head full of great ideas. "Science is in the air" is the perfect description of my time in his lab in Rennes. Thank you for the inspiration Etienne.

I also want to thank Prof. Salvatore Grimaldi and Dr. Thomas Corpetti for accepting to review my manuscript. My thanks extend to Prof. Eric Marchand, Dr. Magali Jodeau and Dr. Alexandre Hauet for accepting to take part in my thesis committee.

Much gratitude to my colleagues at Irstea. Espeically Dr. Emeline Perret, whom my stay with at office 167 was a great experience. She was certainly very helpful while I was trying to improve my french but also my understanding of river hydraulics. In addition, of course, to all those fun times we had together. Thank you for that dear Emeline. My thanks extend also to Dr. Jêrome Le Coz, Dr. Sebastien Proust and Dr. Jean-Baptiste Faure for their insightful comments and explications on hydrualics and fluids mechanics. Also to everyone at Irstea whom I might have had the chance to talk to on any subject while taking a coffé break or enjoying a meal.

Finally, I would like to thank EDF-DTG for partially funding my thesis. The same goes to the SCHAPI agency.

To the sole of my beloved grandmother Halima.
To the sole of my beloved father.
To mom, Sally and her kids (Samia, Ghassan and Haneen).

Abstract

This thesis is an application of *computer vision* findings to river velocimetry research. Hydraulic research scientists already use various image processing techniques to process image sequences of rivers. The ultimate goal is to estimate free surface velocity of rivers remotely. As such, many risks related to intrusive river gauging techniques could be avoided. Towards this goal, there are two major issues need be addressed. Firstly, the motion of the river in image space need to be estimated. The second issue is related to how to transform this image velocity to real world velocity. Until recently, image-based velocimetry methods impose many requirements on images and still need considerable amount of field work to be able to estimate rivers velocity with good accuracy. We extend the perimeter of this field by including amateur videos of rivers and we provide better solutions for the aforementioned issues.

We propose a motion estimation model that is based on the so-called *optical flow*, which is a well-developed method for rigid motion estimation in image sequences. Contrary to conventional techniques used before, optical flow formulation is flexible enough to incorporate physics equations that govern rivers motion. Our optical flow is based on the scalar transport equation and is augmented with a weighted diffusion term to compensate for small scale (non-captured) contributions. Additionally, since there is no ground truth data for such type of image sequences, we present a new evaluation method to assess the results. It is based on trajectory reconstruction of few Lagrangian particles of interest and a direct comparison against their manually-reconstructed trajectories. The new motion estimation technique outperformed traditional methods in image space.

Finally, we propose a specialized geometric modelling of river sites that allows complete and accurate passage from 2D velocity to world velocity, under mild assumptions. This modelling considerably reduces the field work needed before to deploy Ground Reference Points (GRPs). We proceed to show the results of two case studies in which world velocity is estimated from raw videos.

Résumé

Introduction

Dans cette thèse, on s'intéresse à l'application du domaine de la vision par ordinateur à la vélocimétrie de surface des rivières. Les hydrauliciens utilisent déjà plusieurs routines de traitement d'images pour traiter des vidéos de rivières. Le but ultime est d'estimer la vitesse surfacique d'un cours d'eau par une méthode sans contact. Cela permet aux chercheurs d'éviter les risques liés au jaugeage intrusif des rivières, notamment en période de crue. Dans ce but, deux enjeux sont à prendre en compte. Tout d'abord, le mouvement apparent de la rivière dans l'espace image doit être estimé. Ensuite, ce mouvement, estimé en pixels par unité de temps, doit être transformé en une vitesse réelle exprimée en mètres par seconde par exemple. Jusqu'à présent, les méthodes de vélocimétrie par images imposent quelques contraintes sur les séquences pour qu'elles soient exploitables (notamment une caméra fixe et le besoin de la présence physique des équipes hydrauliques au site de jaugeage avant ou après l'événement). Dans cette thèse, on vise à élargir ce périmètre en incluant les vidéos prises par des amateurs (c'est à dire de paramètres inconnus, et avec un mouvement potentiel de la caméra) tout en présentant de meilleures solutions pour les enjeux précédemment mentionnés.

Estimation du mouvement

L'estimation du mouvement en traitement d'images est une question de recherche fondamentale dans le domaine de la vision par ordinateur. Beaucoup d'ambiguïtés concernant une scène dynamique sont enlevées si le mouvement des objets formant cette scène est connu. En général, il y a deux approches différentes. La première, dite locale, impose une contrainte de similitude de la vitesse sur tous les points dans une petite fenêtre sur l'image. La deuxième, dite globale, impose les mêmes contraintes sur tous les points de l'image. Les méthodes de vélocimétrie de rivières par images utilisent une approche locale, dite PIV (Particle Image Velocimetry) (Adrian, 1991). L'idée principale c'est de faire une correspondance entre plusieurs fenêtres

définies sur deux images en maximisant un facteur de corrélation entre elles. Cette méthode exige une bonne distribution des traceurs sur la surface de l'eau pour une meilleure corrélation. La taille des fenêtre de corrélation est de plus un paramètre délicat, étant un compromis entre la petite taille nécessaire à un mouvement quasi-identique des pixels, et la grande taille requise pour observer des motifs suffisamment texturés à l'intérieur de cette fenêtre. Les approches globales telles que les méthodes de flot optique ont fait l'objet de recherche active depuis leur apparition en 1981 (Horn and Schunck, 1981). Le flot optique a montré de bonnes performances pour l'estimation de vitesse pour les écoulements. En effet, on peut faire des liens entre la formulation du flot optique qui a été conçu au départ pour le mouvement rigide, avec les équations de la physique des écoulements. Un modèle récent a introduit un terme de diffusion qui compense la perte des informations à petite échelle causée par l'échantillonnage dans la grille des pixels lors de l'acquisition de l'image. Ces informations ne sont pas négligeables dans le cas des écoulements car elles participent au transfert global d'énergie. Ce terme de diffusion était supposé lié à la viscosité turbulente. On propose ici un modèle simple basé sur la hypothèses de mélange de Prandtl (Prandtl, 1925) pour l'estimer, en profitant des valeurs de vitesses hiérarchiques estimées avec un schéma de flot optique multi-résolutions. Le modèle de flot optique final est donc constitué d'un terme d'attache aux données physiques venant de l'équation de transport, augmenté par un terme de diffusion turbulente basé sur un modèle de turbulence et un terme de régularisation conventionnel qui permet d'avoir des couches de mélange naturelles sans introduire des continuités artificielles non réalistes. En l'absence de vérité terrain pour le champ de vitesse dans l'espace image (à la différence des applications de flot optique pour le mouvement rigide et pour le mouvement des écoulements généraux), il est difficile d'évaluer les résultats sur les séquences de rivières. Une solution de substitution est de suivre des particules passives (traceurs) à la surface de l'écoulement de la rivière à partir des champs de vitesses estimés. Pour cela, on utilise une méthode d'intégration Runge-Kutta du 4ème ordre et on compare la trajectoire obtenue avec une trajectoire reconstruite manuellement pour la même particule. Plusieurs séquences avec des différentes caractéris-

tiques ont été utilisées pour la comparaison. De façon générale, le modèle proposé a obtenu de meilleurs résultats que le modèle classique de flot optique et que les méthodes PIV.

Modélisation géométrique

Après avoir estimé le mouvement dans l'espace image, une transformation de cette vitesse en vitesse métrique est requise. Pour cela, certaines connaissances sur les paramètres de la caméra qui a pris la vidéo sont nécessaires. C'est l'étape de calibration de la caméra. Si on possède la caméra (physiquement), il est possible d'estimer ces paramètres en prenant plusieurs vues d'objets de géométrie connue (en général une géométrie plane). En faisant le lien entre ces objets et leurs projections dans les images, on peut facilement arriver à déterminer les paramètres de la caméra. C'est une procédure standard et mature, de nombreuses applications de vision par ordinateur commencent par cette étape. Les méthodes conventionnelles de vélocimétrie rivière par image utilisent des caméras connues ce qui facilite la transformation géométrique et qui permet par la suite à faire ce passage 2D vers 3D. La LSPIV par exemple utilise une caméra fixe et quelques points de contrôle sur la scène pour arriver à enlever les effets de perspective et avoir une nouvelle séquence d'images orthorectifiées pour laquelle le rapport pixel/monde est connu. On peut ainsi estimer une vitesse métrique directement sur les images. Les deux inconvénients de cette approche sont que le déploiement des points de contrôle nécessite un travail terrain important avec plusieurs opérateurs. Deuxièmement, cette étape d'ortho-rectification interpole et change la fonction d'intensité de l'image ce qui peut en dégrader le signal. La fonction de l'intensité de l'image est la seule source d'information pour estimer la vitesse sur les images et il est donc préférable de ne pas la changer. Les vidéos amateurs sont plus complexes à exploiter avec la difficulté liée à une caméra inconnue mais aussi au mouvement de la caméra pendant l'acquisition, qu'il s'agisse d'un mouvement volontaire de l'individu qui prend la vidéo ou involontaire du simple fait que la caméra est tenue à la main. Ces deux problématiques sont bien connues dans le domaine de vision par ordinateur. La

réponse à la première est l'auto-calibration, c'est-à-dire une calibration de la caméra faite seulement avec l'information contenue dans les images. La réponse à la seconde est la stabilisation de la vidéo en estimant le mouvement de la caméra et en l'enlevant de la vidéo. Avec l'utilisation de ces deux techniques, la différence avec les séquences contrôlées est minimisée. La surface libre d'une rivière est en perpétuelle évolution, mais ses mouvements verticaux sont très limités par rapport à sa taille transversale et longitudinale. Ainsi, il est possible de formuler l'hypothèse que la surface d'un écoulement en rivière est plane. Sous cette hypothèse, on peut faire la correspondance plan à plan entre la région de la rivière dans le plan de l'image et le plan de la rivière dans le monde 3D. Les déplacements 2D peuvent ainsi être projetés en 3D, seule l'échelle par rapport au monde 3D réel demeure inconnue. Cette échelle peut être estimée théoriquement à partir des objets reconstruits en 3D dont la taille (ou longueur) est connue dans le monde réel. Toute cette manipulation géométrique fait appel à quelques méthodes de vision par ordinateur, notamment la détection de point d'intérêt et la géométrie épipolaire. Basée sur ces deux méthodes, la stabilisation des vidéos et l'auto-calibration sont réalisées. Pour détecter les points d'intérêt, la seule condition nécessaire est de disposer de suffisamment de régions fixes dans les images. L'auto-calibration quant à elle a besoin du mouvement de la caméra pour pouvoir faire une triangulation 3D. Il y a une forte analogie entre les sites des rivières et les scènes d'intérieur explorées par le domaine de la robotique. En effet, dans les deux cas sont présents des plans dans lesquels la détection de points d'intérêt est ardue. On réduit le problème de reconstruction 3D des rivières à une reconstruction 3D basé sur des plans orthogonaux. Cela permet de simplifier le problème initial et tous les points de l'image qui font partie de la rivière pourront être construits en 3D avec fidélité. On compte ensuite sur le plan vertical pour trouver des objets de taille connue pour estimer le facteur d'échelle dont on a besoin pour passer à la taille réelle du site.

Estimation de la vitesse métrique

La première tâche à faire est de déterminer si une calibration de la caméra est possible. Par exemple si la caméra est inconnue, il faut nécessairement observer un mouvement de la caméra durant la séquence d'images pour que l'autocalibration soit possible. Ensuite, on extrait une séquence d'images consécutives pour faciliter l'estimation par flot optique, mais également et surtout parce que si la camera bouge, la stabilisation des images consécutives est aussi plus aisée. Une reconstruction 3D est donc faite avec une image et le facteur d'échelle est extrait à partir des objets trouvés sur les plans verticaux de la rivière (panneaux, murs, parties de ponts, etc.). Une section en travers est déterminée et les vecteurs du déplacement 3D sur le plan de la rivière sont obtenus en projetant les vecteurs 2D. Le facteur d'échelle est finalement utilisé pour mettre à l'échelle ce déplacement. La vitesse est donc obtenue en fonction de l'intervalle temporel choisi pour la séquence extraite.

Conclusion

Dans cette thèse, nous avons proposé une nouvelle méthode d'estimation de vitesse basée sur l'équation de transport scalaire, avec l'ajout d'un terme de diffusion pour compenser la perte d'information dans le processus d'acquisition d'images. Elle donne sur les séquences testées de meilleurs résultats que le flot optique classique et que la méthode PIV. Le passage des vitesses estimées dans l'espace image aux vitesses métriques du monde réel est assuré par une méthode de reconstruction 3D monoculaire de la surface de l'écoulement, inspirée des techniques de robotique en environnement intérieur. Le développement d'une méthode d'auto-calibration et de stabilisation d'images permet de généraliser l'application des méthodologies proposées à toutes les séquences d'images, notamment celles acquises par une caméra aux paramètres inconnus et en présence de mouvement lors de l'acquisition. Le processus complet depuis la vidéo jusqu'aux vitesses métriques est illustré par différents exemples de séquences d'images.

Contents

List of Figures

Chapter 1

Introduction

We are living in the era where *computer vision* is steadily moving from a theory-only discipline to a wide-range of applications in several scientific domains. Visual data is all over the place and it is being captured by a variety of different devices like smartphones, drones, surveillance cameras or medical imaging equipment. One can now rely on computer vision to inspect goods in industrial machine vision applications, provide vision for autonomous cars or even unlock smartphones using facial recognition.

In this dissertation, we apply computer vision to river velocimetry research. River velocity estimation is an essential part in river gauging process. Why do we gauge rivers anyway ? well, rivers are important for the planet as they participate in the global water cycle. They carry nutrients and are the natural habitat for wild life, providing them food and drink. Some rivers also provide energy and act as a transport medium. As a matter of fact, rivers affect many human activities. We are particularly interested in rivers in motion with noticeable discharge. This also includes seasonal urban floods. Scientists study these natural phenomena to get a deeper understanding of their dynamics and life cycle. The gauging data is important to allow historical data comparisons with current data points. As such, one obtain a tool by which any aberrant behavior could be quickly identified and reported. Urban floods in particular receive a special attention because of their direct (and potentially catastrophic) impact on local communities. These kind of

Figure 1.1: Catastrophic impacts of inundations in France.

hazards are of a repetitive nature and they affect millions of people in areas at risk. Figure (1.1) shows few photos of floods aftermath in France. We need to be able to study these hazards and also predict them. This is not an easy task if we consider the large number of variables involved. Therefore, it is beneficial to be able to collect more data to better fit statistical models. In river gauging, we collect volumetric velocity data (the *discharge*) and water level (the *stage*) on a specific location. Some other measurements regarding water quality or sediments properties might also be collected. The gauging is either performed on hydraulic stations built on the site or using mobile stations instead.

There are several limitations towards the ultimate goal of frequent gauging and more data collection. Scientists can not have hydraulic stations everywhere they would like to and can not be to many sites at the same time at their will, specially on sudden or short period events. In addition to that, some events like floods are very dangerous for the instruments and the

personnel. It requires more preparations and additional safety measures.

In recent years, and in order to overcome the difficulties related to dangerous events, non-intrusive techniques (radar and image-based) gained popularity. These techniques enable us to estimate river free surface velocity without direct interaction with the river itself. The free surface velocity is a valuable piece of information that can be used to estimate the discharge, provided that other information like *rating curves*, *bathymetry,* etc., are available.

Radar devices only provide very sparse data (one point per location/device), many devices might be used at the same time for reliable estimations. The concept of radar-based velocity estimation is based on interpreting the impulse response of an electromagnetic signal emitted on the river free surface at a certain angle. The impulse response is reflected by the rugosity of the free surface with a different frequency that is proportionate to the velocity (the Doppler Effect). An assessment of their performance could be found in (Welber et al., 2016).

Another approach is to use image-based techniques. Remarkable work has been done using image-based techniques in the last two decades (Jodeau et al., 2008, Muste et al., 2014, 2008, Le Coz et al., 2014). Most of them are based on the seminal work of Fujita et al. (1998). The idea behind it is as follows: a camera is mounted somewhere at a certain orientation to capture videos of the river of interest. Some Ground Reference Points (GRPs) with known 3D coordinates have to be deployed manually, either just before the event or in a constant manner. These GRPs help the researcher to geometrically modify the obtained images in order to remove the perspective effect (a process called ortho-rectification). A new set of images is thus obtained with a predefined spatial resolution. The river surface has to be seeded with tracers before the event so that clear motion patterns could be observed. Displacements could then be estimated on these images using correlation-based techniques. The velocity is obtained directly since the time between images is known. Figure (1.2) shows photos of some of these techniques in action.

While both approaches greatly helped avoiding risks, the spatio-temporal limitations are still present. Image-based techniques use cameras mounted in some predefined locations and provide rather sparse measurements in space.

Figure 1.2: Velocity estimation techniques: Intrusive (top), radar-based (bottom left) and image-based (bottom right).

Recently, Dramais et al. (2011) showed the interest of using mobile LSPIV station. While this might elevate the spatial sparsity of measurements, it is still bounded to temporal constraints especially for sudden or far away events. Also, promising results were obtained by exploiting amateur videos of such events in a post-event gauging framework (Pénard et al., 2015, Le Boursicaud et al., 2016).

With the popularity of smartphones equipped with a camera, it turns out that people have the tendency to capture footage of such events and share them on social networks. Searching YouTube for example gives thousands of videos of floods. It is not uncommon to collect data from willing citizens for hydrology research. Lowry and Fienen (2013) for example showcased a system (called CrowdHydrology) that collects the stage measurement from citizens via text messages. A dedicated server then takes into account all the data and display it publicly on the web. It is a very ambitious idea to try to exploit amateur videos for river velocimetry research. This has the potential to provide scientists with a huge new source of data that bypass the aforementioned spatio-temporal constraints. It can even go beyond that to enable gauging events that occurred in the past. This generalization however comes with a cost. The camera used is no longer known and image sequences are generally not stable. Image quality could be degraded too due to compression algorithms used by social media sites.

To the best of our knowledge, there is no scientific research on how to exploit amateur videos for the purpose of river velocimetry, without ever going to the site. A complete and general framework is therefore proposed in this thesis to address this issue while enhancing the existing routines that are commonly applied to this field.

As per any specialized application, the general theory of computer vision might not be applicable out of the box. To this end, we stratify the problem into sub-problems. We briefly review the literature of image-based river velocimetry on the sub-problems. Then, we see what the general theory of computer vision has to offer and eventually we provide enhanced and specialized solutions for river flow image sequences. The thesis is organized as follows: in the next chapter we tackle the problem of motion estimation

Figure 1.3: An image extracted from an amateur video of a flood showing another amateur capturing the same event with a smartphone.

in image sequences. The third chapter will focus on geometry problems, it addresses the question of how to go from 2D image velocity to world metric velocity. In the fourth chapter, we combine concepts from the second and the third chapters to finally estimate a real world velocity starting from raw videos. Two case studies with different conditions are shown. We conclude in chapter five.

Chapter 2

Image Motion Estimation

Motion estimation is one of the fundamental questions in computer vision. A lot of ambiguities involved in understanding a random scene are resolved if the motion of objects in that scene is known. An obvious example could be found in the growing domain of autonomous vehicles. The vehicle needs to be aware of the motion of pedestrians and other vehicles to calculate its next move. We are interested in the motion of rivers free surface. In general, motion estimation of fluids is considered a special case scenario. Indeed, fluids change their shape all the time and they have more complicated motion patterns than rigid objects. We review in this chapter the literature of motion estimation in image sequences. We then proceed to review motion estimation for fluids. Eventually, we derive a method that is specialized to rivers free surface motion and we compare it to conventional motion estimation techniques and other specialized ones. We give more space to the parts that we judged relevant and eventually led to the final mathematical model we propose for rivers.

2.1 Introduction

The motion estimation problem in images is almost always tackled as a correspondence problem. One looks for a feature or a pattern that is invariant under motion and could be uniquely identified throughout an image sequence. Based on this, the correspondence could be successfully established and the displacement is then calculated by taking the location difference of these corresponding features. This is sufficient for applications that do not require real world motion measurements. For example, in intelligent video surveillance applications or even in some video compression algorithms, some tasks are launched only when a motion in image space is detected. The exact velocity (or displacement) here is irrelevant and only the existence of motion is what really matters. In image space, the terms velocity and displacement are used interchangeably implying implicitly that a time unit of 1 is indeed used. However, if a real world motion measurement like the velocity of specific object is needed, then the real time unit is obtained from video frame-rate and is used instead. Some applications might only need the velocity in terms of pixels. In this case, the velocity is readily obtainable if the displacement is known. Other applications might need a real world velocity. In this case, a conversion from motion in image space to motion in metric world units is required. This is exactly the case that we are going to study in this thesis. The sought real world measurement is the velocity of the free surface of rivers or urban floods.

The majority of works found in computer vision literature rely on photometric features correspondence. They are extracted from the intensity (or brightness) function of the image. In general, motion estimation methods are divided into two main classes: global or local. Global methods constitute an energy (or cost) functional based on constraints imposed on the entire image. The minimizer of this functional gives a global solution to all points involved. Local methods, on the other hand, work on image windows of some size (say 15×15 pixels) and try to minimize an error expression based on constraints imposed on these windows individually. Local approaches tend to be more robust to noise in the sense that there is no error propagation between two

different grid points while global approaches are in general more sensitive to it, (Barron et al., 1994). The advantage however provided by global approaches over local ones is that they supply a dense vector field i.e. a vector for every pixel. This is in general a beneficial feature for image sequences of fluids because one can directly observe or extract other relevant fluids patterns like streamlines or vorticity fields. Also, using prior knowledge about the expected vector field, global methods give more plausible results in areas with no motion clues.

2.1.1 Local approach

It is in the early 80s when a very important paper in computer vision was published by Lucas and Kanade (1981) (LK). Even if the title of the paper was about image registration and it didn't mention any thing about "motion estimation", the algorithm proved later to be very effective for that purpose. This is however a logical outcome since image registration and motion estimation are both correspondence problems. LK algorithm idea is to minimize a quadratic error between a template image $T(S)$ and an input image $I(S)$. Some parameterized function $\mathcal{F}(S;\omega)$ tries to transform (or to warp) the input image to resemble the template image where ω is a vector of parameters and S is an image location $S = [x, y]$.

$$\sum_{S} \left[I\left(\mathcal{F}(S;\omega)\right) - T(S) \right]^2$$

In the case of motion estimation, the vector ω is just the 2D displacement vector $[u, v]$. The original and the template images are small windows on the two image frames. Since the relationship between pixel coordinates S and image intensity value at that position is not linear, the above expression is solved in a non-linear fashion.

2.1.2 Global approach

It was about the same year when Horn and Schunck (1981) (HS) published their influential paper on motion estimation. It is one of the most cited pa-

pers in computer vision (12900 citations at the time of this thesis writing). While the term *optical flow* might be equally applicable to any motion estimation algorithm in image sequences, it is widely used to refer to HS-style global algorithm. Horn and Schunck (1981) firstly coined this term in the title of their original paper. Optical flow is defined as the pattern of apparent motion of objects, surfaces and edges in a visual scene caused by the relative motion between an observer and that scene. Most of motion estimation research throughout the years were focused on HS global optical flow. Optical flow belongs to a larger global methods family called the *variational methods*. They are used in many computer vision tasks. Optical flow and image restoration/denoising are the most popular examples. Variational methods are based on a classical optimization technique for cost functions. Their formulation permits a clear and unambiguous modelling without hidden assumptions or variables. In addition, even if images are in discrete space, variational methods formulation is intrinsically continuous, leaving the numerical solution to be determined by the discretization scheme chosen. This gives the advantage of a clearer derivation but leaves the door open for more accurate solutions if better discretization schemes are used. A variational functional typically contains two terms. One term is related to the data, called the data term or the observation term. The second term is called the regularization term. The regularization part plays an important role in the estimation. On the one hand, it helps reducing the solution search space by adding *apriori* information about the expected solution. On the other hand, it has this *fill-in* effect that occurs whenever the data term carries no information or is too noisy. The regularization propagates information to *fill-in* the gaps. The problem is finally formulated as the minimization of an energy function of the type:

$$\int_{\Omega} data\,term + \alpha\, regularization\, dS$$

where Ω is the 2D image domain and α is a strictly positive factor that controls the importance of one term against the other. A simple example to grasp variational formulation is *image denoising* application. One looks for

the *original* denoised image starting from an observation (the noisy image).

$$\int_{\Omega} \|I - f\|^2 + \alpha \|\nabla f\|^2 \, dS \tag{2.1}$$

The data term in the above expression encourages the sought denoised image f to be close to the observed noisy one I. The second term imposes that the denoised image should be smooth by penalizing the gradient of its intensity function. For optical flow, Horn and Schunck (1981) suggested *brightness constancy* (BC) assumption as a feature to establish the correspondence. This is to be the data term for optical flow. It is supposed that points keep their intensity values after moving in a small time interval. This could be written in an equation as:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

The right hand side describes the motion of a point in space and time, the equality affirms that the intensity value is unchanged. This is of course non-linear as these increments could take any value. In an attempt to linearize this model, a Taylor expansion about the point (x, y, t) is developed:

$$I(x, y, t) = I(x, y, t) + I_x \delta x + I_y \delta y + I_t \delta t + \sigma \tag{2.2}$$

where the subscripts denote partial derivatives and σ represents second and higher order terms. After simplification and neglecting σ we end up with:

$$I_x \delta x + I_y \delta y + I_t \delta t = 0$$

dividing both sides by δt gives:

$$I_x \frac{\delta x}{\delta t} + I_y \frac{\delta y}{\delta t} + I_t = 0$$

Let $\omega = [u, v]$ represents the velocity vector field, the above equation can be re-written as:

$$I_x u + I_y v + I_t = 0 \tag{2.3}$$

Equation (2.3) has two unknowns which means it can not be solved without additional information. Only the motion component in the direction of the gradient $\nabla I = [I_x, I_y]$ can be recovered. The motion in the direction parallel to the gradient remains undetermined. This causes a visual phenomenon known as *the aperture problem*, implying that a bigger aperture around a given point is needed to add more constraints. Aperture problem is the reason behind many optic illusions since part of the motion is not determined, Figure(2.1). The worst case scenario occurs in uniform intensity regions where the gradient in both directions vanishes. This means that any value (wrong values included) for u and v would satisfy Equation (2.3).

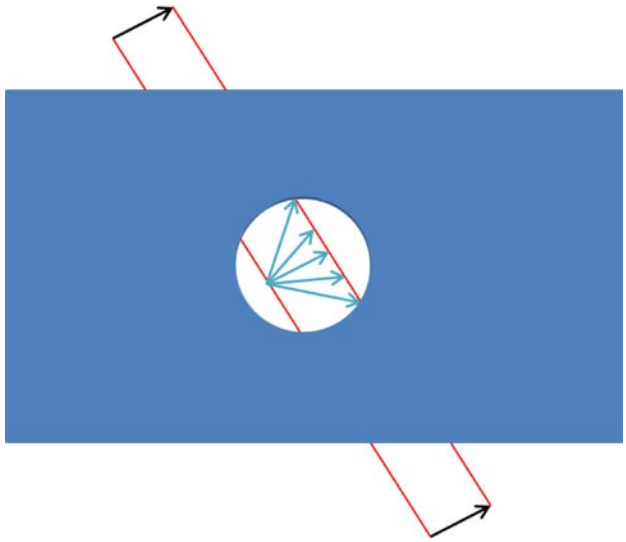

Figure 2.1: An illustration of the aperture problem, any vector inside the aperture could describe the motion of the red line viewed through the aperture. However only the vector parallel to the black vectors outside the aperture is the correct one.

The additional information comes from the regularization part. It helps reducing the ambiguity of the above model by restricting the search space. This is done by adding *apriori* information about the expected vector field and by propagating information to *fill-in* where needed. The regularization proposed by Horn and Schunck consists of assuming a smooth vector field everywhere in the image. This is based on the observation that neighboring

points are likely to belong to the same surface. Points on rigid surfaces move together when the surface moves. It is a plausible assumption in the case of rigid or quasi-rigid objects. The energy then reads:

$$E(u,v) = \iint_{\Omega} \left(I_t + I_x u + I_y v\right)^2 + \alpha \left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right) dxdy \tag{2.4}$$

The data term, encapsulates the main hypothesis about the data which is in this case the brightness constancy (BC). The regularization part tries to minimize the norm of the gradient of the flow field. In other words, neighboring points should have similar velocities. This last idea resembles the implicit assumption made in local methods which assumes that all points in the same window have the same velocity. The regularization here is however more relaxed, points on the same window could have different velocities, if that some how minimizes the energy function. The regularization then tries to propagate information to regions with weak gradient from estimations performed on neighboring regions that might have good gradient signal. The optimality condition of this energy function could be attained using Euler-Lagrange equations (a classical result of calculus of variations). Euler-Lagrange equations for the above model are:

$$\begin{aligned} I_x \left(I_x u + I_y v + I_t\right) - \alpha \Delta u = 0, \\ I_y \left(I_x u + I_y v + I_t\right) - \alpha \Delta v = 0, \end{aligned} \tag{2.5}$$

where Δ is the Laplacian operator. These are Partial Differential Equations (PDEs). For simple models like in Equation (2.4), these equations generally result in a linear system of equations that could be solved using standard linear algebra routines.

2.2 Improved optical flow

Over the years, the model described by Equation(2.4) witnessed many improvements and modifications. We are going to detail only major improvements and describe the derivation of an improved HS model.

One major shortcoming of this model is the linearization of the BC assumption in the form of Taylor expansion, Equation (2.2). This makes the model only valid for small motion magnitudes. This is of course not the case in general and especially in rivers or floods. To cope with large displacements, a coarse-to-fine multi-resolution strategy is introduced by Bergen et al. (1992). The velocity field is divided into an approximated field that is known and a correction field to be estimated. The correction field is supposed to be small and thereby fulfills the small motion assumption. To achieve this, the original image is down-sampled successively to form an *image pyramid* , Figure (2.2). This has direct impact on optical flow since the image is getting smaller at each level and the optical flow "shrinks" accordingly. Starting from the *smallest* pyramid level, a *small* optical flow field is estimated. It is then propagated to the next finer level after applying the necessary scaling via interpolation. This vector field represents the approximated known field for the finer level. The second image of the finer level is then brought *closer* to (or warped towards) the first image using the approximated known field. The estimation on the current pyramid level is performed to recover the small correction field. The final flow field at this level is the addition of these two vector fields. After obtaining a full vector field for the pyramid level at hand, it is then again propagated to the next finer level and the the same process is repeated until the finest level (original image) is reached.

The regularization term imposes a first order smoothness constraint on the entire vector field. While this might be true for points sharing the same surface, it is a strong assumption on objects boundaries (also called discontinuities) inside the image. Scenes are normally composed of many objects that vary in shape, color or depth. The boundaries between these objects and/or with the background represent challenging regions for the model described by equation (2.4). Indeed, these regions would likely to have different motions. Imagine a region in a scene located between two objects boundaries. If the two objects move differently, that part of the image would contain two different motions. The smoothness assumption is then violated. A quadratic penalization (as in the above) tends to average between these competing motions. This will result in a smoothing over regions that basically should not

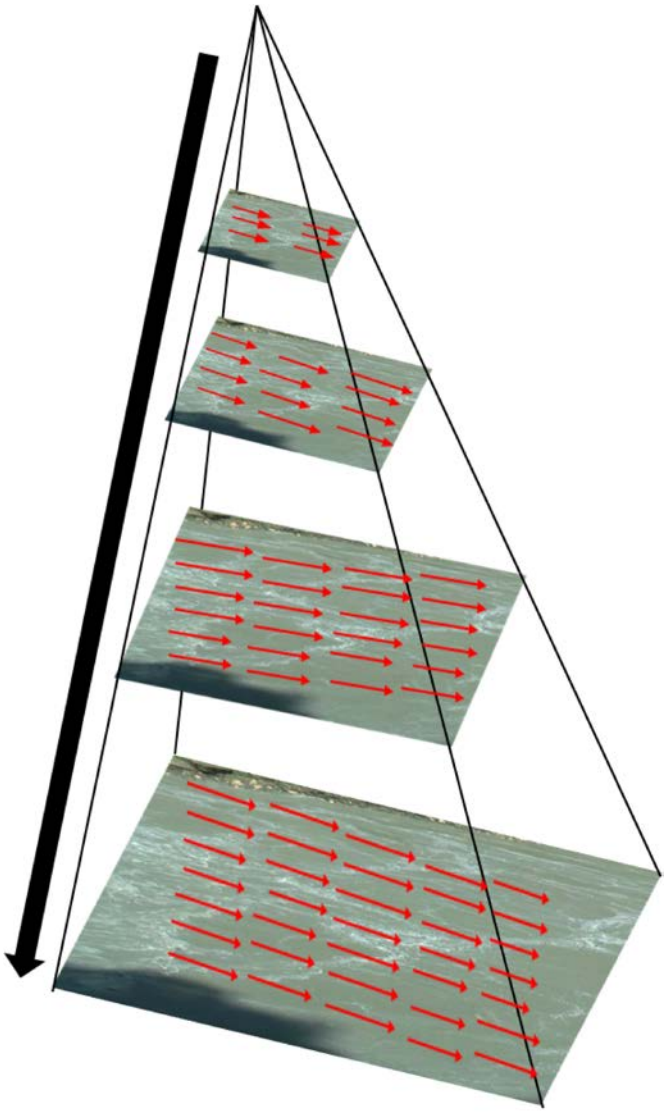

Figure 2.2: The original image is down-sampled to form a multi-resolution image pyramid. The estimation is performed between two image pyramids going downward until the original image.

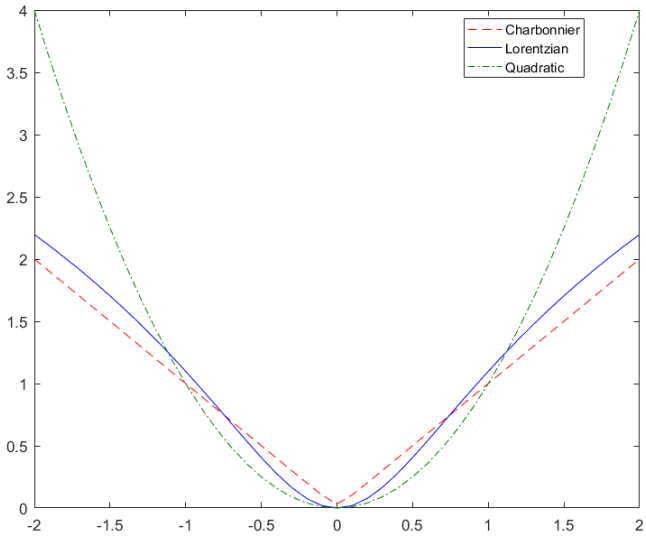

Figure 2.3: Different penalization functions superimposed

be smooth. Consequently, the model will fail to provide accurate estimations in these regions as it will consider one motion and see the other competing motion as an outlier. In that sense, Black and Anandan (1996) suggested to use *robust functions* instead of the usual quadratic function to reduce the effect of outliers. This can also be beneficial to the data term as shown by Mémin and Pérez (1998a). Indeed, the BC assumption is also prone to errors due to reflections, transparency, occlusions and many other factors. We have considered two robust functions throughout the thesis. The Lorentzian $\psi(x,\sigma) = log\left(1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right)$ and the Charbonnier $\psi(x) = \sqrt{x^2 + \epsilon}$ where σ is a parameter to control the shape of the function and ϵ is a constant introduced to ensure the function differentiability, typically chosen to be 0.001. Figure (2.3) shows the shapes of these functions against the quadratic.

One of the key practices that significantly improved the accuracy of optical flow is use of median filtering to reject outliers in the intermediate flow (the flow between image pyramids) during warping steps (when the second image is warped back closer to the first). It has been introduced into pyramidal optical flow computation heuristically at first. It has significantly improved the estimation accuracy. (Sun et al., 2014) showed later on that by using it, the energy functional being minimized is a different energy. This

different energy just tries to regularize over a larger spatial area.

Mémin and Pérez (1998b) and Brox et al. (2004) used the original non-linear model $I(S) = I(S+\partial S)$. This non-linear optical flow modelling allows the integration of large displacements within the framework, unlike the linearized model which only expects small motions. The model using non linear BC and robust functions is:

$$E(u,v) = \iint_{\Omega} \psi\left((I(x+u,y+v,t+1) - I(x,y,t))^2\right) + \psi\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right) dxdy \tag{2.6}$$

where ψ is a robust function. This model is highly non-linear and probably non-convex (if the robust non convex Lorentzian is used). It is either solved iteratively using a multi-grid optimization technique (Mémin and Pérez, 1998b) or a Graduated-Non-Convexity (GNC) technique (Blake and Zisserman, 1987), with two fixed points iterations.

We show in the following a series of *Euler-Lagrange* equations for the non-linear model above:

$$\begin{aligned} &\psi'\left(I_t^2\right).(I_x I_t) - \alpha\nabla\cdot\left(\psi'\left(|\nabla u|^2 + |\nabla v|^2\right)\nabla u\right) = 0, \\ &\psi'\left(I_t^2\right).(I_y I_t) - \alpha\nabla\cdot\left(\psi'\left(|\nabla u|^2 + |\nabla v|^2\right)\nabla v\right) = 0, \end{aligned} \tag{2.7}$$

where ψ' is the derivative of the robust function with respect to its parameters and $\nabla\cdot$ is the divergence operator. Note that we need to rewrite these equations while considering the coarse-to-fine and warping-based strategy. As mentioned earlier, it is supposed that there is a *known* vector field $w = (u, v)$ and a correction field $dw = (du, dv)$ at every pyramid level. The final propagated field (or the final field if we are on original image level) is the addition of these two vector fields. But first let's again study the equations above. They are highly non-linear due to the non-linear terms that come from the non-linear BC or from the derivative of the robust function. Variational methods are typically solved using algorithms like Gauss-Seidel or SOR (Successive Over Relaxation). To that end, a linear system of equations must be obtained. Taking advantage of the coarse-to-fine warping strategy,

a fixed point iteration around w is defined. For more clarity we introduce an iteration variable k so that the sought vector field at each level verifies $w^{k+1} = w^k + dw^{k+1}$.

$$\begin{aligned}\psi'\left(\left(I_t^{k+1}\right)^2\right).\left(I_x I_t^{k+1}\right) - \alpha\nabla\cdot\left(\psi'\left(\left|\nabla u^{k+1}\right|^2 + \left|\nabla v^{k+1}\right|^2\right)\nabla u^{k+1}\right) = 0,\\ \psi'\left(\left(I_t^{k+1}\right)^2\right).\left(I_y I_t^{k+1}\right) - \alpha\nabla\cdot\left(\psi'\left(\left|\nabla u^{k+1}\right|^2 + \left|\nabla v^{k+1}\right|^2\right)\nabla v^{k+1}\right) = 0\end{aligned} \tag{2.8}$$

The following Taylor expansion linearization could be applied:

$$\begin{aligned}I_t^{k+1} &\approx I_t^k + I_x^k du^k + I_y^k dv^k,\\ I_x^{k+1} &\approx I_x^k + I_{xx}^k du^k + I_{xy}^k dv^k,\\ I_y^{k+1} &\approx I_y^k + I_{xy}^k du^k + I_{yy}^k dv^k,\end{aligned} \tag{2.9}$$

where a double subscript means a second derivative quantity. Let:

$$\begin{aligned}(\psi')_D^k &= \psi'\left(\left(I_z^k + I_x^k du^k + I_y^k dv^k\right)^2\right)\\ (\psi')_R^k &= \psi'\left(\left|\nabla\left(u^k + du^k\right)\right|^2 + \left|\nabla\left(v^k + dv^k\right)\right|^2\right)\end{aligned} \tag{2.10}$$

The Euler-Lagrange equations to reflect the introduction of the correction field are:

$$\begin{aligned}(\psi')_D^k\left(I_x^k\left(I_t^k + I_x^k du^k + I_y^k dv^k\right)^2\right) - \alpha\nabla\cdot\left((\psi')_R^k\nabla\left(u^k + du^k\right)\right) = 0,\\ (\psi')_D^k\left(I_y^k\left(I_t^k + I_x^k du^k + I_y^k dv^k\right)^2\right) - \alpha\nabla\cdot\left((\psi')_R^k\nabla\left(v^k + dv^k\right)\right) = 0,\end{aligned} \tag{2.11}$$

It is still however non-linear in the derivative of the robust functions. Another inner fixed point is employed to estimate $(\psi')_D^k$ and $(\psi')_R^k$. This second iteration variable is assigned l. The final *linear* Euler-Lagrange equations are:

$$\begin{aligned}(\psi')_D^{k,l}\left(I_x^k\left(I_t^k + I_x^k du^{k,l+1} + I_y^k dv^{k,l+1}\right)^2\right) - \alpha\nabla\cdot\left((\psi')_R^{k,l}\nabla\left(u^k + du^{k,l+1}\right)\right) = 0,\\ (\psi')_D^{k,l}\left(I_y^k\left(I_t^k + I_x^k du^{k,l+1} + I_y^k dv^{k,l+1}\right)^2\right) - \alpha\nabla\cdot\left((\psi')_R^{k,l}\nabla\left(v^k + dv^{k,l+1}\right)\right) = 0,\end{aligned} \tag{2.12}$$

These are linear equations in du and dv. It is worth noting that the non-linear model, combined with warping and multi-resolution scheme, results in successive linearization steps using Taylor expansion that might better approximate the motion. In contrary to the original HS model where only one global linearization is used.

Algorithm 2.1 Computation of optical flow on a pyramid level

```
for l = 1 to the max number of warping steps do
    Warp the second image towards the first using current u and v.
    Compute the weights ψ'.
    Initialize du , dv to zero.
    for k = 1 to max number of linearization steps do
        Linearize the data term using equation (2.9)
        Solve for du and dv using equation (2.12)
    end for
    update u and v using du and dv
end for
```

One last detail is the use of GNC (Graduated-Non-Convexity) *continuation* method (Blake and Zisserman, 1987). If a non-convex robust function is used, the energy would contain multiple minima and global unique solution is not guaranteed. The idea behind continuation methods is to simply provide a good initialization so that a good minimum is attained. Otherwise a random initialization might get us stuck in a local (usually bad) minimum. Accordingly, the original energy is smoothed to remove small structures that are responsible for local minima. We construct a series of objective functions (depending on the GNC iterations desired) starting with a quadratic (E_Q) (and convex) objective function and moves gradually (and linearly) towards the original robust (E_R) and non-convex problem using:

$$cE_Q(u, v) + (1 - c)E_R(u, v)$$

It is not guaranteed that the favorite scenario as in Figure (2.4) to occur every time but nevertheless it is a good practice that should at least help the optimization to avoid the first (usually very bad) local minima.

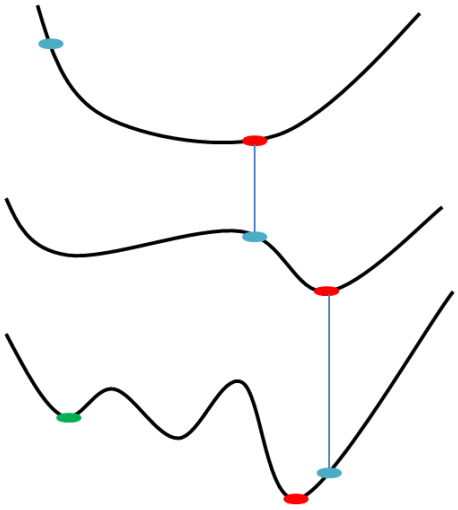

Figure 2.4: continuation methods: points in blue are initialization points, red are the minima using the continuation method and the green is the minimum without the continuation method.

2.3 Preliminary results

We run these motion estimation approaches on a couple of images of a river running from right to left. Figure (2.5) shows the result of original HS plotted on the first image of the sequence. One vector is plotted every 20 pixels in both directions for better visibility. The color code (Baker et al., 2011) is another way to visualize the whole vector field, the color itself represents the direction of the vector and color intensity represents the magnitude. In comparison to the result of the improved HS in Figure (2.6), the improved algorithm recovered more homogeneous and smooth vector field. But most importantly, the algorithm recovered bigger magnitude for almost the entire river area, even those with less information (small intensity gradient). This shows the important role of the multi-resolution coarse-to-fine strategy in helping the algorithm to recover bigger magnitudes. For now however, we only assess these results visually. By observing the original sequence, we notice a turbulent but still translatory motion from right to left. In the improved HS result, we observe continuous motion from right to left with a

magnitude that seems similar to what is observed in the original sequence by the naked eye. The original HS failed to give full continuous motion, one could observe spurious vectors in its color code result, for example points in white means there is no motion observed. Points in magenta appears in many regions on the entire image. This color means there is actually motion from left to right, exactly opposite to the motion observed visually. This is of course an utter outlier. The amount of points with this color gives some indication about the outliers in the vector field. Figure (2.7) shows the result of LK algorithm, there is no color-coded results since we only have a sparse vector field.

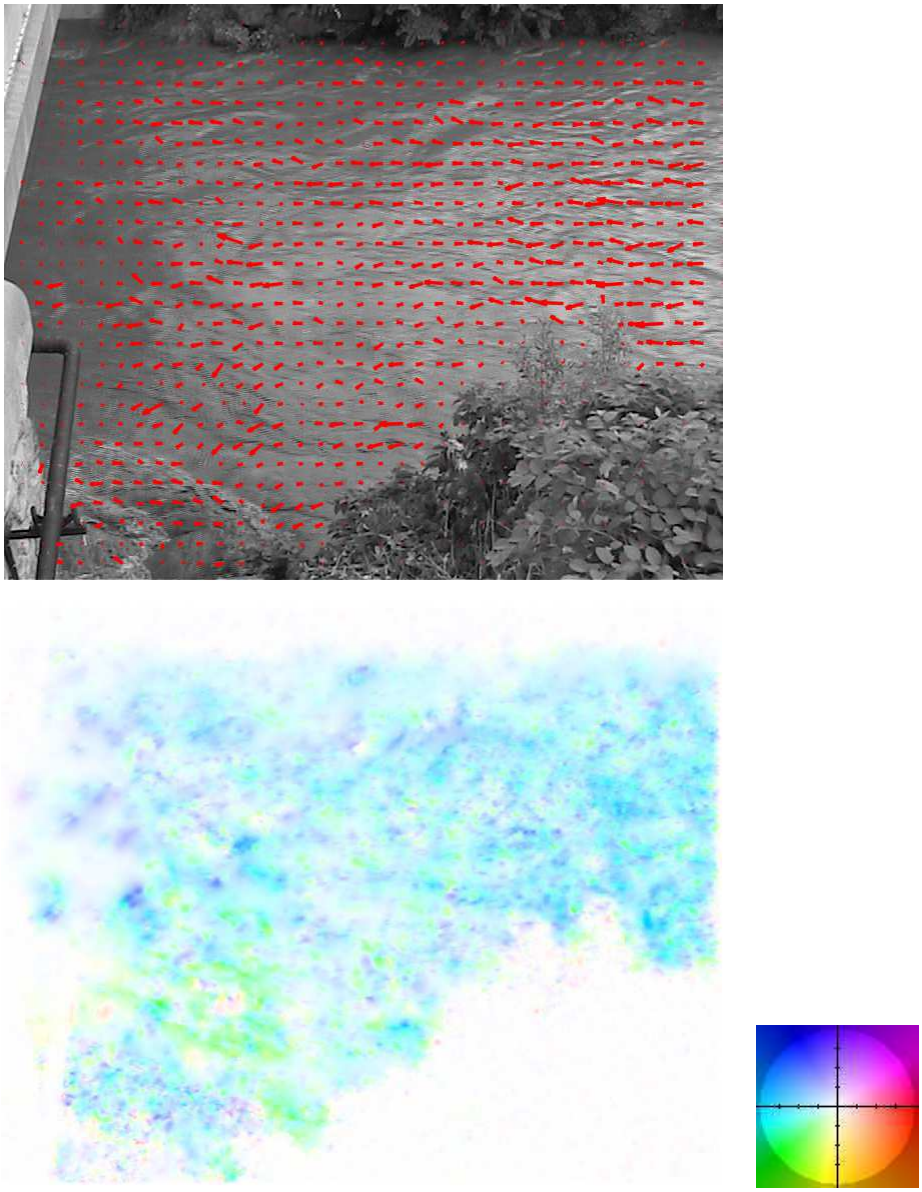

Figure 2.5: Top: HS result super-imposed on the first image, only one vector is plotted for every 20×20 region of pixels for visualization purposes. Bottom: HS result for all pixels in color code format, the color itself represent the direction and color intensity represents the magnitude of the vector.

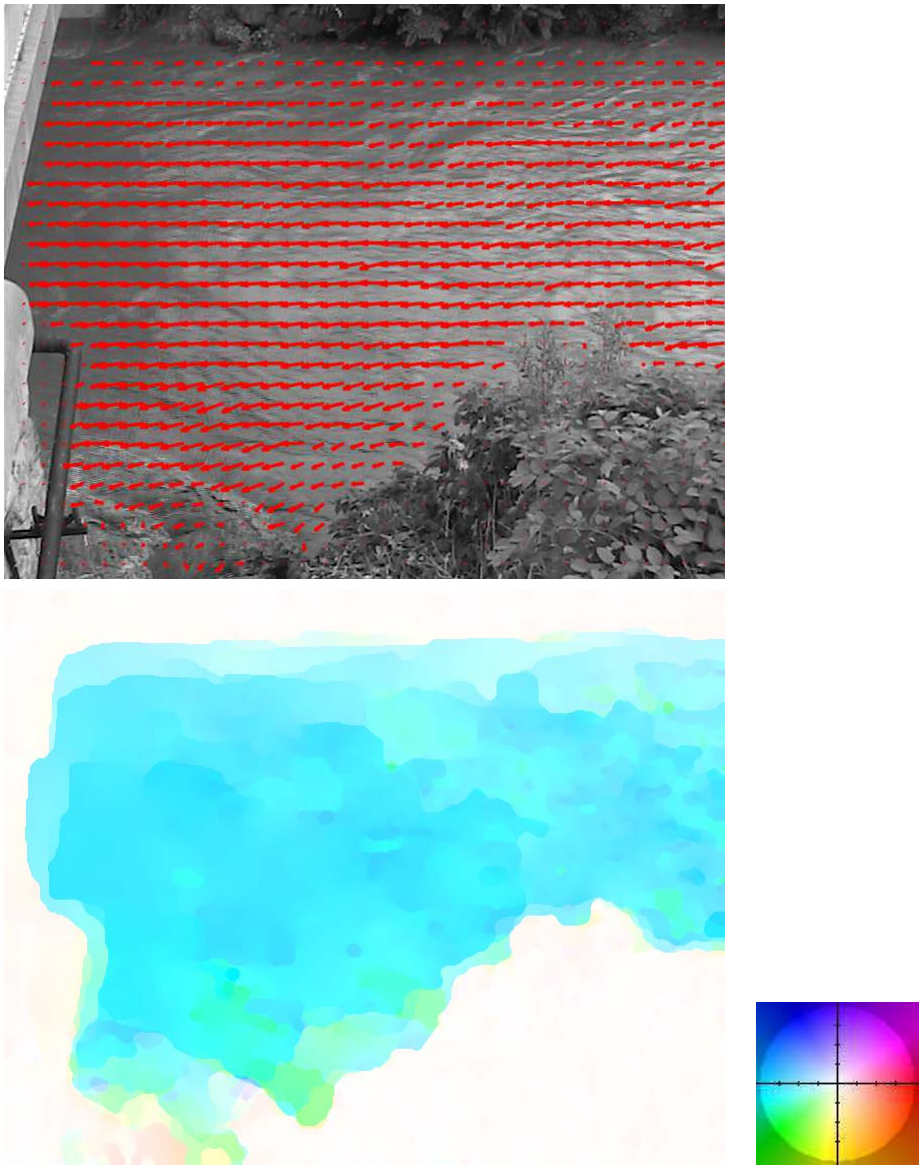

Figure 2.6: Top: Improved HS result super-imposed on the first image, only one vector is plotted for every 20×20 region of pixels for visualization purposes. Bottom: Improved HS result for all pixels in color code format,color itself represent the direction and color intensity represents the magnitude of the vector.

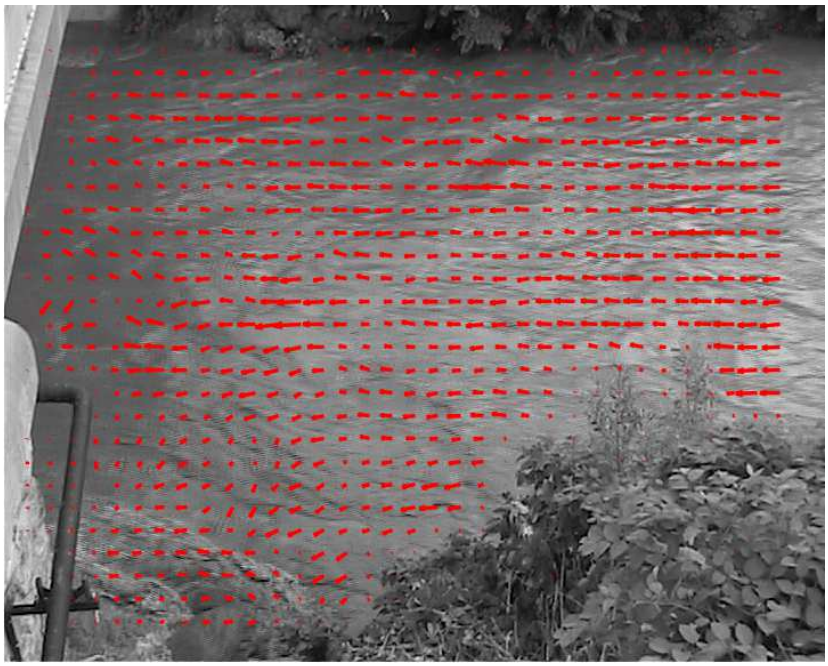

Figure 2.7: LK result with window size of 20×20 pixels.

2.4 Motion estimation for fluids

In the 90s, local methods became very popular in experimental fluid mechanics community with the introduction of Particle Image Velocimetry (PIV) technique (Adrian, 1991). The original technique consists of recording particles moving within a fluid that is being illuminated by a laser sheet. While LK and PIV are both local approaches as the computations are performed over a set of predefined windows, they differ in how they establish the correspondence. In PIV, a window of a certain size (called the *Interrogation Area* IA) is chosen. The goal is to find a window in the second image which maximizes the correlation score computed using the formula in equation (2.13) while assuming translational motion between images. To this end, a bigger size window (called the *Search Area* SA) is determined according to the direction of the fluid. The search is thus conducted in this larger window to fit another window within it with the same size of IA. The normalized correlation is expressed as:

$$R = \frac{\sum_{S \epsilon IA} \left(A_S - \overline{A}_S\right) \left(B_S - \overline{B}_S\right)}{\sqrt{\left[\sum_{S \epsilon IA} \left(A_S - \overline{A}_S\right)^2 \sum_{S \epsilon IA} \left(B_S - \overline{B}_S\right)^2\right]}} \tag{2.13}$$

where A and B are image intensity distributions of two windows in two different images. To compute significant correlation scores, the flow has to be well-seeded with particles. As a consequence, PIV faces difficulties on scalar images in which many areas have low intensity gradients. In many cases, there is a necessary post-processing step to correct or remove spurious vectors, inevitably generated in regions with low intensity gradient. The post-processing step may also consists in interpolating the sparse vector field to generate a denser one. Choosing a suitable window size is a tricky task, one needs bigger window to accumulate more information for better correlation score computation, but then the bigger the window, the more likely it contains complex motions far from the window translational motion assumption. Despite its wide-spread usage specially in controlled lab environments, PIV lacks sound physics foundation, it treats all image sequences equally, regardless of the nature of the object in motion. Furthermore, it only computes sparse estimations, many of them might be discarded in the post-processing phase. In recent years, this PIV trend has found a promising contender to take over image-based velocity estimations in fluids, that is: optical flow. A comprehensive review on the application of optical flow to fluids flows could be found in Heitz et al. (2010). Most of optical flow adaptations to fluids focused on the general case described by Navier-Stokes equations. The applications vary from satellite imagery, transmittance imagery or PIV laboratory experiments using laser sheets. This type of fluids exhibits an important degree of vorticity, Figure (2.8). In river image sequences, this is not necessary the case, rivers exhibit more of a translatory motion than rotational one.

Liu and Shen (2008) formally established the relationship between optical flow and fluid flow based on the perspective projection of the transport equation on the 2D plane. Authors found that optical flow is proportional to the path-averaged velocity field weighted with a relevant field quantity. This gave the physical foundation needed since BC assumption is not based on any physical principle but in fluid images it is rather an image transport

constraint.

A lot of regularization techniques try to deal in some way or another with the vorticity (or curl) and the divergence of the vector field, depending on the fluid observed. Corpetti et al. (2005) showed that from an optimization point of view, the first order gradient regularization tends to penalize equally the vorticity and the divergence norms of the vector field. The regularization part in equation (2.4) could be rewritten as:

$$\begin{aligned} -\alpha u_{xx} - \alpha u_{yy} &= 0, \\ -\alpha v_{xx} - \alpha v_{yy} &= 0, \end{aligned} \tag{2.14}$$

If we compare above expressions to those of a second order div-curl expression $\alpha \left\|u_x + v_y\right\|^2 + \beta \left\|v_x - u_y\right\|^2$, Euler-Lagrange will give:

$$\begin{aligned} -\alpha u_{xx} - \beta u_{yy} - (\alpha - \beta)\, v_{xy} &= 0, \\ -\beta v_{xx} - \alpha v_{yy} - (\alpha - \beta)\, v_{xy} &= 0, \end{aligned} \tag{2.15}$$

which is the same as equation (2.14) if $\alpha = \beta$.

Depending on the fluid observed, more suitable and physics-based regularization for fluids have been suggested. Higher order regularization has been proposed before by Suter (1994) based on the gradient of both divergence and curl of the vector field. These two operators are of particular importance because they are fundamental features of fluids. This model tries to *conserve* these quantities by penalizing the norm of their respected gradients.

$$\int_\Omega \left\|\nabla (\nabla \cdot \omega)\right\|^2 + \left\|\nabla (\nabla \times \omega)\right\|^2 dS$$

This however, leads to a system of four coupled PDEs of order 4 that is difficult to solve numerically. Corpetti et al. (2002) approximated this quantity by introducing auxiliary variables into the regularization term in a way

similar to variational image denoising shown before in equation (2.1).

$$\int_{\Omega} \alpha \left(\left(\|\nabla \cdot \omega - \xi\|^2 + \lambda \|\nabla \xi\|^2 \right) + \mu \left(\|\nabla \times \omega - \zeta\|^2 + \lambda \|\nabla \zeta\|^2 \right) \right)$$

Where μ is a weighting parameter, ξ and ζ are estimates for the curl and the divergence, respectively. The nice thing about this formulation beside the simplicity in approximating the complicated fourth degree PDEs, is that it allows the injection of any prior knowledge of these two quantities. One can see that when the exact divergence and curl are found, the corresponding terms evaluate to zero and the optimization would only rely on the second term (i.e. $\lambda \|\nabla \xi\|^2$) that is responsible of smoothing these quantities. Heitz et al. (2008) suggested to refrain from multi-resolution strategy for PIV applications because the smoothing and down-sampling of images will suppress the particles in the coarsest levels. Instead, authors suggested combining the robustness of local methods with the dense estimation of global approaches, thus replacing the estimations of the multi-resolution scheme with a dense estimation derived from the local correlation method used. Let us remark that local methods are not limited by small-motions-only condition, thus multi-resolution scheme could safely be ignored. Additionally, another term is added to constrain the estimated vector field to be conforming to a physically sound prior based on the equations of Navier-Stokes, it is an adaptation of an earlier work by Heas et al. (2007), the energy has the following form:

$$E = E_{data\,term} + E_{Regularization} + E_c\left(\omega, \omega_R\right) + E_p\left(\omega, \omega_p\right)$$

where the previously described simplified div-curl regularization is used. The term E_c encourages the sought vector field to be consistent with ω_R, the field that is derived from the correlation method. The same applies to E_p where ω_p is a vector that is derived from physics based constraints.

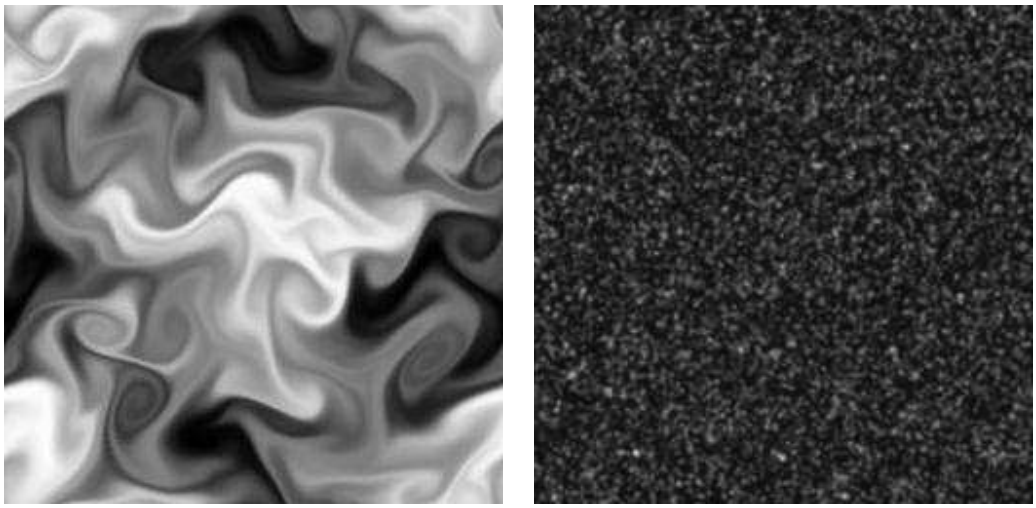

Figure 2.8: Left: Image of scalar incompressible turbulent fluid sequence dataset (Carlier and Wieneke, 2005). Right: PIV image from the same sequence

2.5 Motion estimation for rivers

The first attempts in image-based motion estimation for rivers flows are due to Fujita et al. (1998). Authors suggested to take PIV outside the lab to estimate velocity on rivers image sequences. Their method is called LSPIV (Large-Scale Particle Image Velocimetry) and it relies on a geometric transformation called ortho-rectification that gives direct relationship between pixels and world units. PIV is then performed on these ortho-rectified images to give direct real world velocity. LSPIV became the benchmark for image-based river velocimetry and many case-studies on different conditions or rivers were conducted (Creutin et al., 2003, Jodeau et al., 2008, Dramais et al., 2011, Muste et al., 2008). Let's however highlight that LSPIV by definition inherited all the shortcomings of PIV discussed earlier. Very recently, Tauro et al. (2017) suggested to replace LSPIV with Particle Tracking Velocimetry (PTV) in field applications. The main idea is to directly track particles to provide more accurate results. The PTV result could be improved further by applying trajectory-based filtering in the post-processing phase. However, PTV still produces even more sparse results and it is highly dependent on

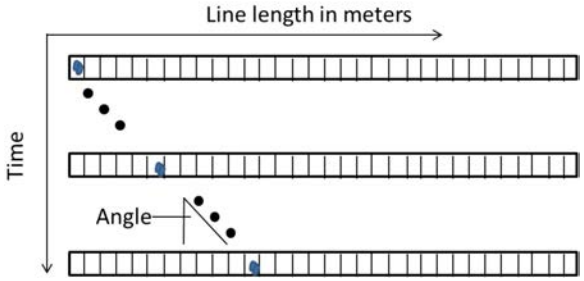

Figure 2.9: A space-time image: the particle in blue changes its position in the line with time which results in a pattern that cross the image at a certain angle.

continuously visible tracers. Fujita et al. (2007) suggested a method based on the so-called space-time images. An image of this type is created by stacking temporal intensity profiles (literally, under each other) of a *searching line* defined in the original image sequence parallel to the main flow. In the ideal case, a clear pattern will emerge crossing the space-time image at an angle Figure (2.9). The velocity is computed based on the angle, time and the metric length of the searching line. Despite being simple and elegant, it uses a strong assumption that the flow field is laminar and hence the tracers stay in a straight line. It is also not clear how the metric length of the searching line is obtained from perspective images. In the following, we propose a physically sound data term and a regularization one to customize optical flow for rivers.

2.5.1 Data term

We derive a method based on optical flow scheme that is customized to rivers flows. We make the assumption that the image intensity function I is related to some passive scalar field concentration C. They are related by a perspective transformation that projects a real world quantity into the observed image plane forming the intensity function I. Let us examine the scalar transport equation:

$$\frac{\partial C}{\partial t} + \nabla \cdot (C\omega) - \frac{1}{Re\, Sc}\Delta C = 0 \tag{2.16}$$

where Re and Sc are the *Reynolds* and *Schmidt* numbers, respectively. The transport equation above links the scalar quantity C to the sought velocity vector field ω. However, in order to be able to extract the velocity, the function C has to contain information till *Kolmogorov* scale (Kolmogorov, 1941). Since C is captured by a camera, the pictured image corresponds indeed to a smooth filtered version of the scalar quantity. It is hence a large-scale representation of the scalar in which the small-scale contributions are omitted. In the case of fluid flows the small-scale effects can not be neglected (Cui et al., 2007). Their action on the large-scale drift component must be modelled. Following Cassisa et al. (2011) and Chen et al. (2015), we propose a LES (Large Eddy Simulation) decomposition of the transport equation to model the small scales contributions in river sequences. The expression $\nabla \cdot (C\omega)$ can be divided into an observed $\nabla \cdot (I\omega)$ and non-observed $(\nabla \cdot (\tau))$ parts:

$$\frac{\partial I}{\partial t} + \nabla \cdot (I\omega) + \nabla \cdot (\tau) - \frac{1}{Re\, Sc}\Delta I = 0 \tag{2.17}$$

The molecular diffusion term $\frac{1}{Re\, Sc}\Delta I$ is usually neglected. We assume that the incompressibility condition for water still holds on the 2D plane (that the divergence of the 2D velocity field is zero). When applying the incompressibility condition and ignoring the molecular diffusion term, equation 2.17 becomes:

$$\frac{\partial I}{\partial t} + \nabla I \cdot \omega + \nabla \cdot (\tau) = 0 \tag{2.18}$$

We see that the BC assumption appears again (in dimensionless form) in addition to the new subgrid term $\nabla \cdot (\tau)$, which means conventional optical flow is consistent with this physics-based derivation for fluids. As suggested by Cassisa et al. (2011) the non-observed term is considered related to turbulent viscosity $\tau = -D_t \nabla I$ where D_t is a turbulent diffusion coefficient. We opted for a simpler model to estimate it using previous velocity estimations within the sequence and/or the pyramid levels of image pair at hand. We feed these estimations to *Prandtl* mixing-length model (Prandtl, 1925). This model uses the stream-wise velocity u and the mixing length l to estimate

the turbulent viscosity $\nu_t = l^2 \left|\frac{du}{dy}\right|$. This quantity is relevant for the computation of $D_t = \frac{\nu_t}{sc_t}$ where sc_t is the *turbulent Schmidt* number which has an empirical value normally determined experimentally. It has been reported that sc_t has widespread values between 0.2 and 3 in general (Tominaga and Stathopoulos, 2007) while others very recently suggested values around 1 to be optimal for water flows (Gualtieri et al., 2017). If we take it to be 1 then D_t is equal to ν_t. The Mixing Length l is defined as the distance traversed by a fluid parcel before it becomes blended in with neighbouring masses. To the best of our knowledge, there is no clear way to predict this value from images. It is taken here to be 1 because the differential optical flow formulation assumes infinitesimal displacements that don't exceed one pixel.

Due to sources or sinks in the fluid, the imaged surface is prone to intermittent changes because of out of plane (depth) motions. Some regions may rise up or sink down during the temporal evolution of the fluid causing changes in the intensity function (Sutton et al., 2008). As a result, the brightness consistency assumption might not hold in many locations. Thus, a robust function is chosen for the data term to reduce the effects of outliers. In practice we mainly use the Lorentzian function (Black and Anandan, 1996) but many others could be used.

2.5.2 Regularization

Regarding the regularization, two main choices were made. Firstly, it can be noticed that usually the river flows of interest do not exhibit strong large-scale eddies motion. Rivers with such eddies have indeed less interest in river velocimetry applications. There might be however many small scale vortices but due to river velocity and/or image acquisition speed, most of them are transported by the large-scale quasi-translational motion. As a result, we opted for a first order gradient penalization like in equation (2.3). As stated earlier, this regularization penalizes the divergence and the curl of the flow field equally. This is a desirable behavior in our case. On the one hand, penalizing the divergence is beneficial to enforce the 2D incompressibility assumed. On the other hand, penalizing a non existent or a weak property

like vorticity will reject solutions with strong vorticity field. For the case of floods image sequences, their strong translatory nature does not promote chaotic motions. In other words, the turbulence does not occur at individual pixels. There is always this notion of groups of pixels that move together and in which case this regularization is optimal. In addition, in uniform intensity areas, the image intensity gradient diminishes and it is the regularization that takes the lead over the data term since the latter depends on image gradient. These uniform areas should then be moving together for this regularization to be relevant. We argue that in uniform intensity image areas, it is unlikely to have very different motions, otherwise, because of water and flow properties including velocity, there would be a mixing phenomenon that would eventually disrupt the uniformity of the intensity function in that specific area. The reverse logic should apply, if the area is uniform in intensity, it is more likely that it contains similar velocity vectors i.e. points move together, in which case this regularization is definitely optimal.

The second choice made is to take a quadratic function instead of robust one for the regularization. The reasoning behind this is: previously robust functions are utilized in the regularization of rigid motion sequences to improve the results on discontinuities. These discontinuities are due to different objects motions and/or depth differences etc. However, there is only one object of interest here which is the river water surface. Though complex, its underlying motion is not discontinuous and there is no reason to introduce penalty terms enforcing the apparition of discontinuities between two adjacent points. The quadratic functions behavior is exactly suitable in this prospect as it tends to average the two different motions in the neighbourhood of interest. Figures (2.10, 2.11 and 2.12) demonstrate this effect. The regularization reads:

$$\int_{\Omega} \alpha \left\| \nabla \omega \right\|^2 dS.$$

The proposed model after combining the two terms is:

$$\int_{\Omega} \psi \left(\left\| \frac{\partial I}{\partial t} + \nabla I \cdot \omega + \nabla \cdot (-D_t \nabla I) \right\|^2 \right) + \alpha \left\| \nabla \omega \right\|^2 dS. \tag{2.19}$$

The final energy after simplifying the above reads:

$$E(\omega) = \int_{\Omega} \psi \left(\left\| \frac{\partial I}{\partial t} + \nabla I \cdot \omega - D_t \Delta I \right\|^2 \right) + \alpha \left\| \nabla \omega \right\|^2 dS, \tag{2.20}$$

which we named SGSD (SubGrid Scale Diffusion).

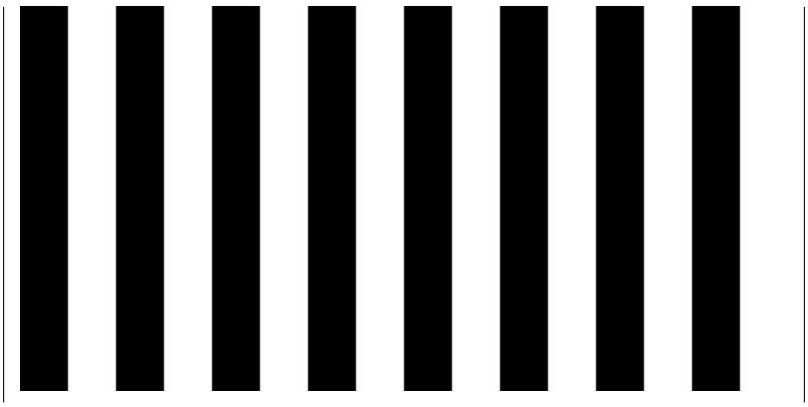

Figure 2.10: Vertical pattern: every black stripe has its width increased to the right in the second image to generate horizontal motion. While white strips has zero motion.

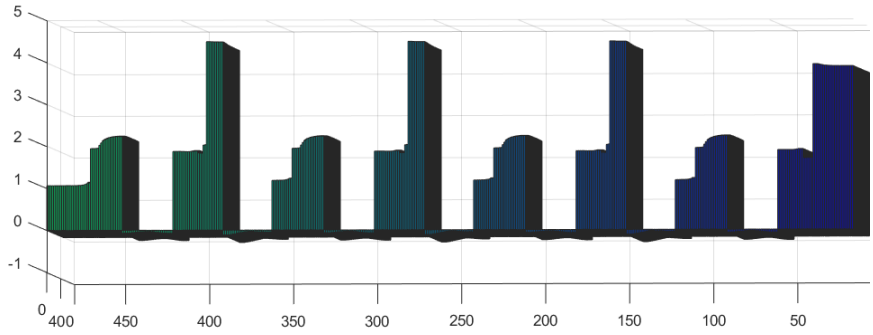

Figure 2.11: Robust estimation, clear motion boundaries obtained.

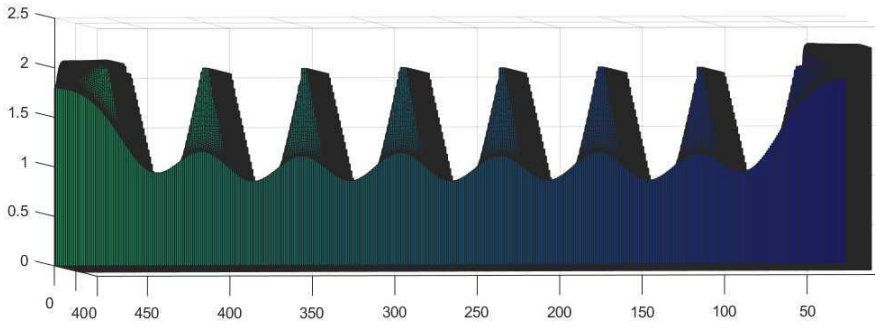

Figure 2.12: Quadratic estimation, the motion on the boundaries is averaged between two different motions.

2.6 Minimization

In practice, the original non-linear displacement model $I(S) = I(S + \partial S)$ is utilized in coarse-to-fine fashion with warping as in (Mémin and Pérez, 1998b, Brox et al., 2004). Not only to allow the estimations of big displacements, but also because the diffusion coefficient D_t depends on the velocity computed on previous coarser pyramid level. The linear BC used earlier is only a Taylor approximation to this model while assuming small motions. The velocity vector field is initialized using conventional optical flow estimation at the coarsest level (this estimator being itself initialized by a zero-valued vector field). The vector field is used to warp the second image of the finer level towards the first image of the same level. Only a small increment is thus sought at this level $\omega^{k+1} = \omega^k + d\omega^{k+1}$, implying that we already know ω^k and we look only for the increment $d\omega^{k+1}$. This is in agreement with the linearization of the non linear displacement model above since $d\omega^{k+1}$ is of small magnitude. The estimated vector field is propagated to the next finer level after applying the necessary scaling.

Equation (2.20) could be minimized using Euler-Lagrange equations as described in section 2.2. However, an Iterative Reweighted Least Squares IRLS approach, previously used by Mémin and Pérez (1998b), is shown to be equivalent to the variational Euler-Lagrange equations (Liu, 2009), yet simpler to derive while working on the discrete equivalent of Equation (2.20):

$$\begin{aligned} E(du, dv) = \sum_S \psi \left(\delta_S^T \left(\mathbf{I}_t + \mathbf{I}_x du + \mathbf{I}_y dv - D_t \left(\triangle \mathbf{I} \right) \right)^2 \right) + \\ \alpha [\left(\delta_S^T F_x \left(u + du \right) \right)^2 + \left(\delta_S^T F_y \left(u + du \right) \right)^2 + \\ \left(\delta_S^T F_x \left(v + dv \right) \right)^2 + \left(\delta_S^T F_y \left(v + dv \right) \right)^2] \end{aligned} \tag{2.21}$$

where δ_S^T is a column vector with all zeros except on the position S. $F_*;_* \epsilon \{x, y\}$ is a derivative filter matrix in the direction of the subscript. u, v, du and dv are all vectorized and $\mathbf{I}_*;\ _* \epsilon \{x, y, t\}$ are all diagonalized, for instance $\mathbf{I}_x = diag\left[I_x\right]$. The goal is to find du and dv that minimizes the gradient

$\left[\frac{dE}{\partial du}; \frac{dE}{\partial dv}\right] = 0$. Using matrix calculus we have:

$$\frac{\partial E}{\partial du} = 2\sum_{X} \psi' \left(\mathbf{I}_x \delta_S \delta_S^T \mathbf{I}_x du + \mathbf{I}_x \delta_S \delta_S^T \left(\mathbf{I}_z + \mathbf{I}_y dv - D_{t,u}\Delta\mathbf{I}\right)\right) + \left(F_x^T \delta_S \delta_S^T F_x + F_y^T \delta_S \delta_S^T F_y\right)(du + u) \quad (2.22)$$

Since $\delta_S \delta_S^T$ corresponds to the identity, some arrangement of the terms will yield:

$$\frac{\partial E}{\partial du} = 2\left(\left(\psi' \mathbf{I}_x^2 + \alpha L\right) du + \psi' \mathbf{I}_x \mathbf{I}_y dv + \psi' \left(\mathbf{I}_x \mathbf{I}_z - \mathbf{I}_x (D_{t,u}\Delta\mathbf{I})\right) + \alpha L u\right) \quad (2.23)$$

where L is a Laplacian filter defined as $F_x^T IDF_x + F_y^T IDF_y$, ID being the identity matrix. The identity here comes from the quadratic function chosen for the regularization. If we use a robust function, we would have a diagonal matrix of the robust weights instead (like in the case of the data term). An equation for dv could be obtained in the same way:

$$\frac{\partial E}{\partial dv} = 2\left(\psi' \mathbf{I}_x \mathbf{I}_y du + \left(\psi' \mathbf{I}_y^2 + \alpha L\right) dv + \psi' \left(\mathbf{I}_y \mathbf{I}_z - \mathbf{I}_y (D_{t,v}\Delta\mathbf{I})\right) + \alpha L v\right) \quad (2.24)$$

The only unknowns beside dv and du are the two turbulent diffusion coefficients, these could now be simply estimated using previous velocity estimates:

$$D_{t,u} = \left|\frac{\partial}{\partial y} u\right|,$$
$$D_{t,v} = \left|\frac{\partial}{\partial x} v\right|. \quad (2.25)$$

IRLS considers the derivative of the robust function as a weight to an ordinary least squares problem. The convergence is achieved when no significant changes are observed in the weights or when a maximum number of iterations is reached. Two fixed point iterations are finally performed. In the inner loop, we solve for du and dv while continuously linearizing the non-linear model and updating the weight ψ' as shown in Algorithm (2.2). We use GNC as described in Section (2.2). The system of linear equations in Equation (2.23) and Equation(2.24) could be organized as:

$$\begin{bmatrix} \psi' \mathbf{I}_x^2 + \alpha L & \psi' \mathbf{I}_x \mathbf{I}_y \\ \psi' \mathbf{I}_x \mathbf{I}_y & \psi' \mathbf{I}_y^2 + \alpha L \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix} = - \begin{bmatrix} \psi' \left(\mathbf{I}_x \mathbf{I}_z - \mathbf{I}_x (D_{t,u} \Delta \mathbf{I}) \right) + \alpha L u \\ \psi' \left(\mathbf{I}_y \mathbf{I}_z - \mathbf{I}_y (D_{t,v} \Delta \mathbf{I}) \right) + \alpha L v \end{bmatrix} \tag{2.26}$$

Algorithm 2.2 Computation of SGSD optical flow on a pyramid level

1: **for** $i = 1$ to the max number of warping steps **do**
2: Compute $D_{t,u}$ and $D_{t,v}$ as in equation (2.27) using current u and v.
3: Warp the second image towards the first using current u and v.
4: Initialize du , dv to zero.
5: **for** $j = 1$ to max number of linearization steps **do**
6: Linearize the data term using equation (2.9).
7: Compute the weight ψ'.
8: Solve for du and dv using equation (2.26).
9: **end for**
10: update u and v using du and dv.
11: **end for**

2.7 Evaluation

One of the key factors for the maturity of optical flow approaches ensues from the availability of ground truth data. Authors compete in benchmarks such as Middlebury (Baker et al., 2011) and MP Sintel (Butler et al., 2012) to evaluate their models. When optical flow was borrowed by experimental fluid dynamics community, similar datasets were required while considering the governing physics equations. A reference simulation based on Navier-Stokes equations is used to generate such images using the simulated (and hence, known) velocity fields (Carlier and Wieneke, 2005). For rivers sequences however, a 3D simulation that respects the physics equations of rivers and then generate 2D image sequences from it is a tedious job and a research question in its own. Rivers 3D simulation is not part of the presented thesis. Unfortunately, 2D simulations are not realistic enough to mimic exactly the changes that happen to the free surface over time. As a result, any such simulation will result in a non-realistic images. In Figure (2.13), we move

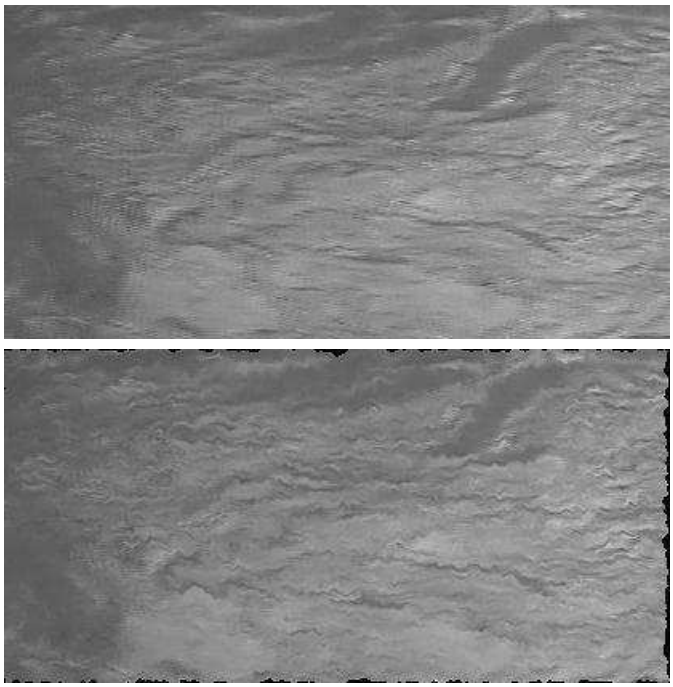

Figure 2.13: Simulated 2D river movements based on a weak-divergence vector field, Top: first image. Bottom: synthesized second image.

pixels between two frames based on a weak divergence vector field. The second frame (bottom) shows some non-realistic zigzag effects.

We show next how we can assess the estimated vector fields and then show results on real world image sequences and on ortho-rectified ones.

2.7.1 Trajectory reconstruction

One way to assess displacement vector fields qualitatively is to check how good it could reconstruct the trajectory of a particle throughout the sequence (Corpetti et al., 2002, Simpson and Gobat, 1994). Suppose we have a sequence of displacement fields $\omega(S, t_0 + n\triangle t)$ where $n\epsilon[0, N-2]$ for $N-1$ vector fields computed on N images and we wish to observe the trajectory of a particle p at position $S(t_0)$ in the first image. Assuming that $\frac{dS(t)}{dt} = \frac{\omega(S(t),t)}{\triangle t}$ we have:

$$\int_{t_0+n\triangle t}^{t_0+(n+1)\triangle t} \frac{dS(t)}{dt} dt = \int_{t_0+n\triangle t}^{t_0+(n+1)\triangle t} \frac{\omega(S(t),t)}{\triangle t} dt$$

These ordinary differential equations ODEs are solved using the classical fourth order Runge-Kutta integration method which gives:

$$S_{n+1} = S_n + 1/6(\omega_1 + 2\omega_2 + 2\omega_3 + \omega_4)$$

where

$$\begin{aligned} \omega_1 &= \omega\left(S_n, n\triangle t\right) \\ \omega_2 &= \omega\left(S_n + \frac{w_1}{2}, t_0 + n\triangle t + \frac{\triangle t}{2}\right) \\ \omega_3 &= \omega\left(S_n + \frac{w_2}{2}, t_0 + n\triangle t + \frac{\triangle t}{2}\right) \\ \omega_4 &= \omega\left(S_n + w_3, t_0 + n\triangle t + \triangle t\right) \end{aligned} \tag{2.27}$$

Spatial and temporal interpolations are used to compute terms between pixels and time steps. The obtained trajectory gives a Lagrangian insight about the estimated vector field. To fully asses the vector field, we manually reconstruct another trajectory of the same point and then we compare the two. In practice we use the tracking software (CellTracker) to manually track particles, Figure(2.14). To improve the accuracy, a zooming function is added to the interface that permits us to select the same point up to few pixels. We use normalized RMSE to measure the difference between trajectories, Equation (2.28).

$$Error(i) = \sqrt{\frac{\left((y(i)_{ref} - y(i))^2 + (x(i)_{ref} - x(i))^2\right)}{\sqrt{\left((y(1)_{ref} - y(end)_{ref})^2 + (x(1)_{ref} - x(end)_{ref})^2\right)}}} \tag{2.28}$$

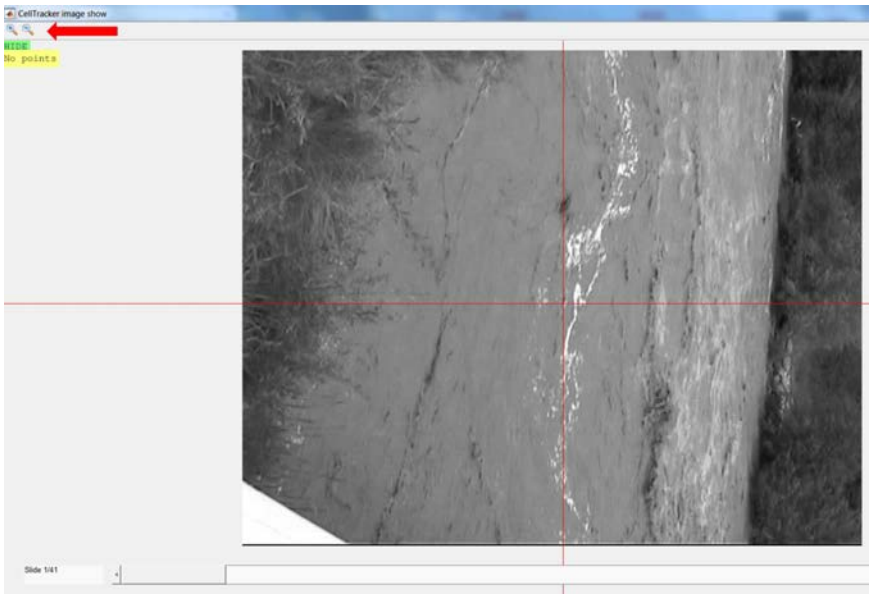

Figure 2.14: CellTracker software interface. We modify it to include a zooming function as indicated by the red arrow.

2.7.2 Results on conventional images

We test the approach in three real world image sequences. The sequences differ in their velocities, image quality, density of tracers (if any) and surface perturbations. The reference trajectories used as ground truth were reconstructed up to few pixels uncertainty. This is because particles change their orientations while moving and their form changes when they are partially (or sometimes fully) occluded by water. This makes it difficult to select the exact same point throughout the sequence. We compare our method SGSD (SubGrid Scale Diffusion) to a modified Horn and Schunck (1981) variation (HS). In fact, (HS) method used for the comparison is exactly the same as SGSD but without the turbulent diffusion term. A PIV-based method Ray (2011) is also compared to the two previous methods.

We use the same parameters for SGSD and the conventional optical flow (HS) to highlight the difference when using the diffusion term. Some of these parameters are general and related to optical flow itself. For instance only two GNC iterations are performed in general. Additional GNC iterations slightly improved the results for sequences with good seeding. Median filtering is an-

other parameter to be considered. In general, it is well-suited for sequences with less turbulence and good seeding. Otherwise if applied to turbulent sequences it might reject turbulent (but maybe correct) vectors and considers them as outliers. However, we found that using it only between GNC iterations (and not between image pyramids) to be beneficial. The weighting factor between the data term and the regularization term is chosen as 0.6 for all sequences. Following Sun (2013), three warping iterations and three linearizations steps per pyramid are used. We found that the Lorentzian robust function works best for the "good sequences". Sequences that suffer in terms of image quality or those which the 2D divergence assumption does not hold, The "Charbonnier" robust function is better than the Lorentzian. One additional parameter is the empirical turbulent Schmidt number Sc_t. We compare the results using different values for this parameter around the value 1 that we chose as default among the reported values (Gualtieri et al., 2017). Considerable changes on the trajectories of some sequences were observed when using an over-weighted or under-weighted diffusion term in noisy sequences.

The color code used is based on Baker et al. (2011) work. As we have seen in Section (2.3), the color itself represents the direction while its intensity represents the magnitude.

The Arc river first sequence

The Arc river, located in the french Alps, is known for its dark color water which enhances the contrast (the gradient) of image intensity when coupled with white tracers. In addition, the image quality itself is good with no changes in scene lighting. The sequence has 41 images with 720×576 resolution. The average displacement in streamwise direction is approximately 11 pixels between two images. In Figure (2.15, top), we plot the mean SGSD dense vector field (with an offset of few pixels for visualization purposes). This gives a quick spatio-temporal idea on the nature of the river at hand. In the bottom of the same figure we show the direction and the magnitude for every pixel in the form of the color code. The spatial coherency of the

estimated vector field is easily observed. Next, we plot the trajectories of different methods along the reference trajectory. In Figure (2.16), it is easy to observe that all methods were generally in agreement with the reference trajectory. This is mainly because of the good gradient signal in this sequence as mentioned earlier, but also because of the good quality of the images. Two zoomed areas around the middle and the end of the trajectory are shown in Figure (2.17). SGSD is almost identical to the reference trajectory, followed by (HS) and then PIV which appears to underestimate the motion magnitude. The zoomed window at the end of the trajectory confirms the results on the zoomed window in the middle (any false positive match in the middle will eventually result in a visible underestimation or overestimation in the window at the end). The normalized distance error in Figure (2.18, top) confirms the results. In Figure (2.18, bottom) we plot the normalized distance error of SGSD using different values for Sc_t. This did not change the results much, there is a maximum error of 0.015 for the case with no diffusion and around 0.05 for other cases with different diffusion coefficients. Still SGSD with $Sc_t = 1$ gives the best results. The dense nature of the vector field enables us to derive other relevant quantities in a straightforward manner. In Figure (2.19,top) the divergence field is shown followed by cross-section trajectories (2.19,bottom).

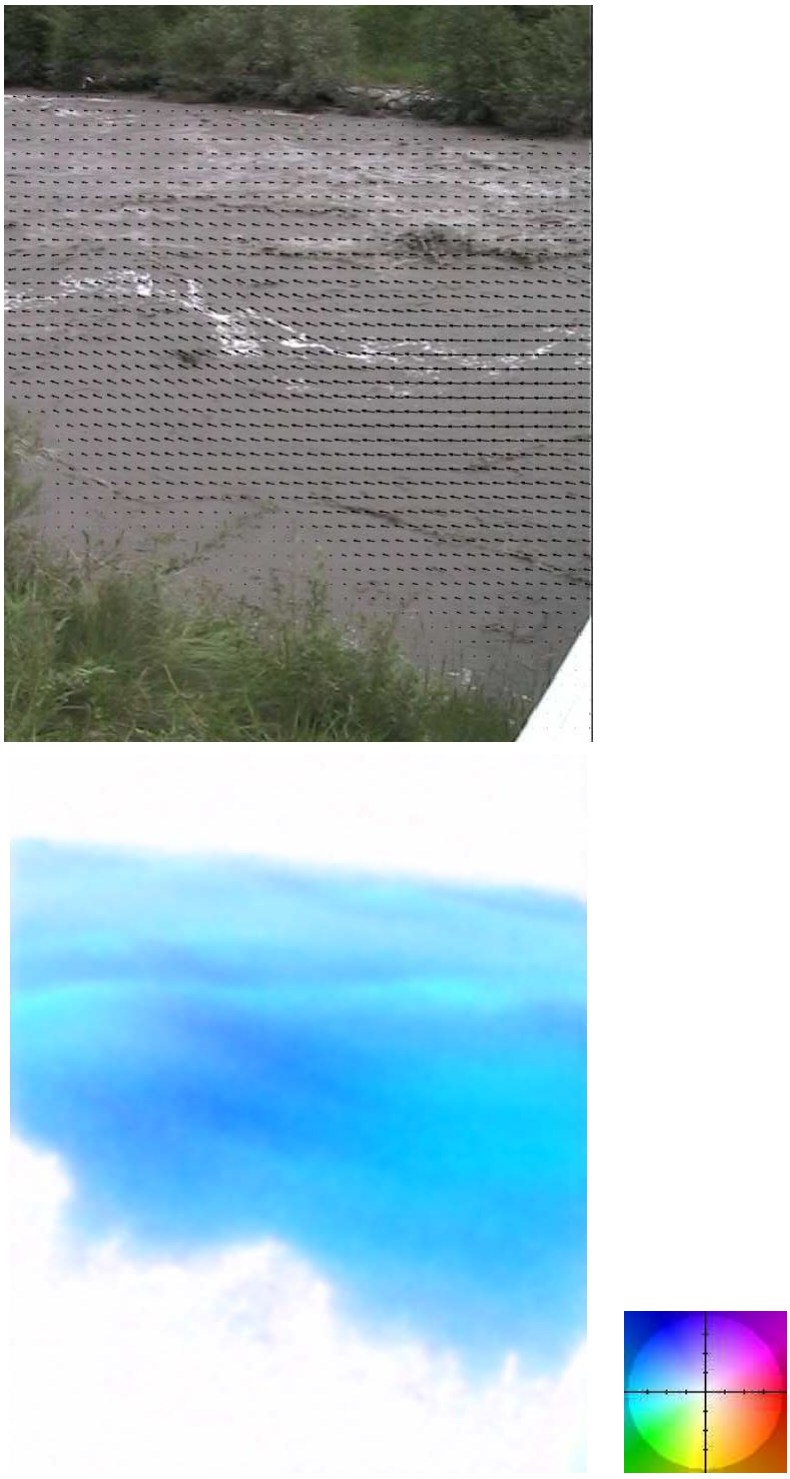

Figure 2.15: Top: SGSD mean field for Arc sequence super-imposed on the first image, only one vector is plotted for every 20×20 region of pixels for visualization purposes. Bottom: SGSD result for all pixels in color code format.

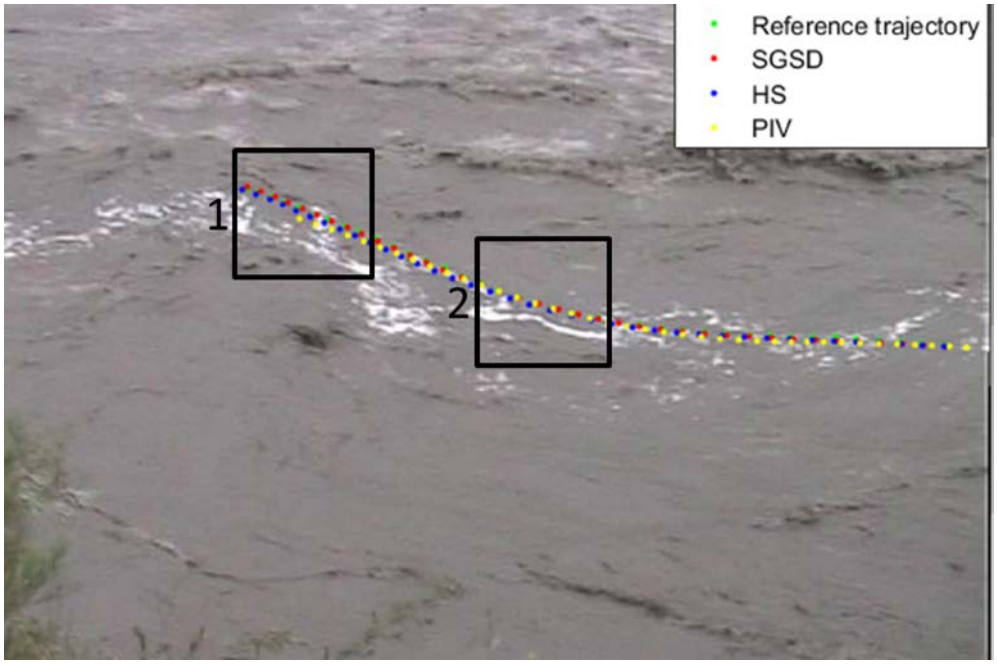

Figure 2.16: The trajectories of different methods superimposed on the first image of Arc river first sequence.

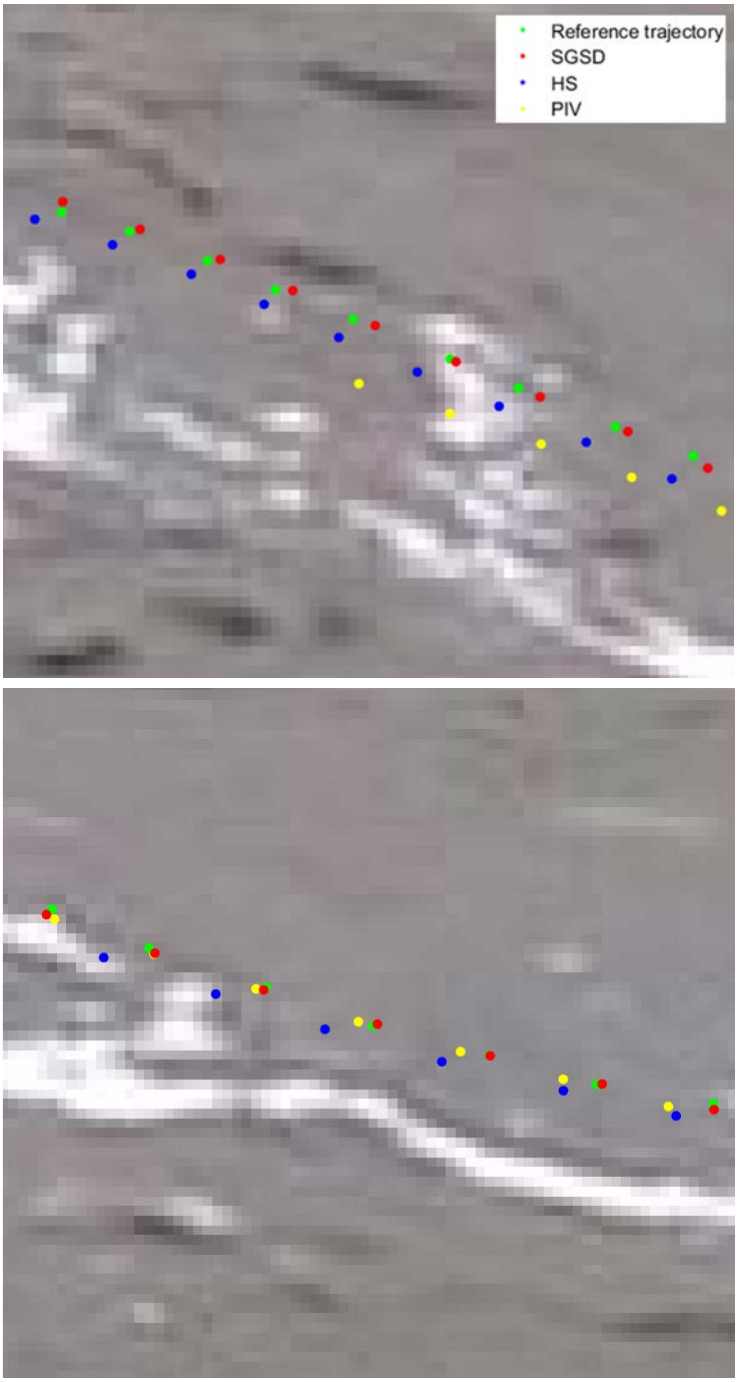

Figure 2.17: Parts of the trajectory seen in Figure (2.16), zoom on window 1 (top). Zoom on window 2 (bottom).

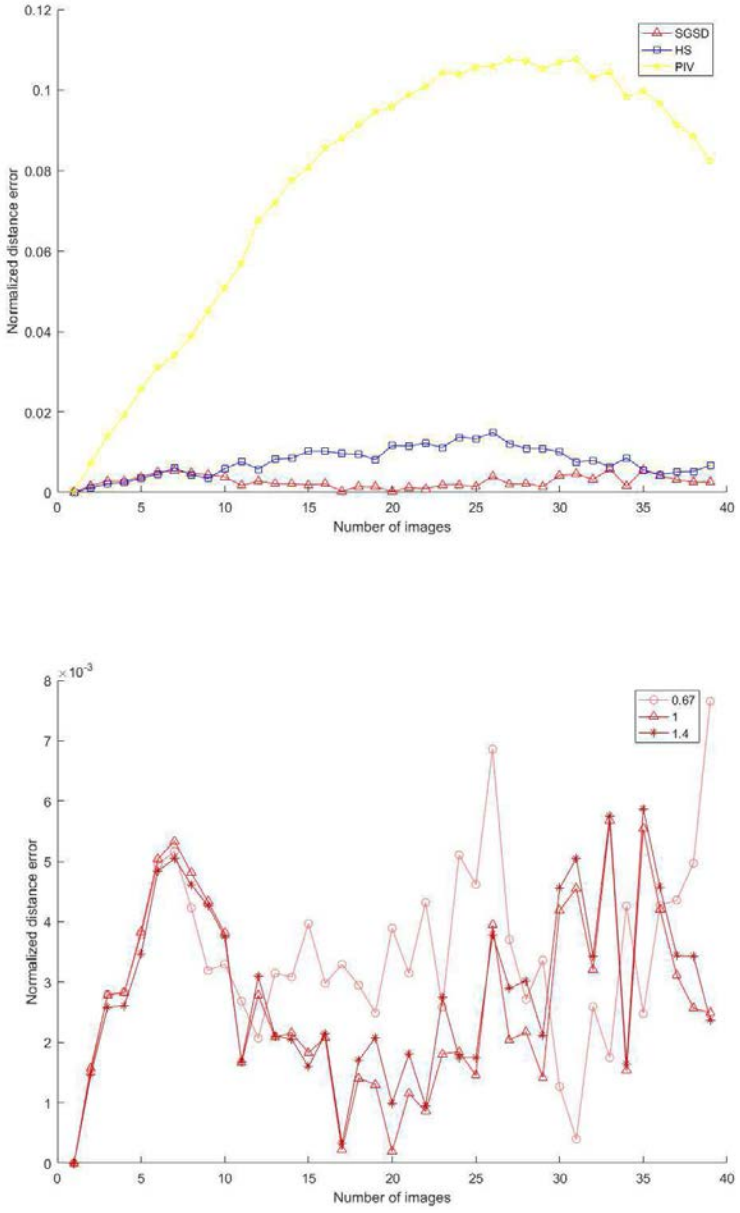

Figure 2.18: Normalized distance error of different methods (top). Normalized distance error of SGSD with different turbulent Schmidt numbers (bottom).

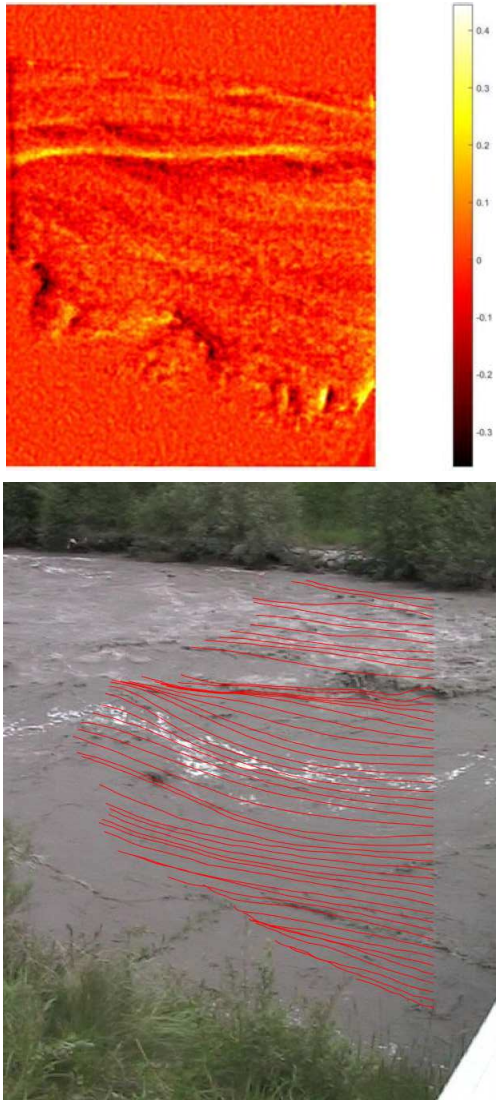

Figure 2.19: Other relevant quantities derived from SGSD for the Arc river first sequence: divergence of the mean vector field (top). Trajectories on a cross-section (bottom).

The Arc river second sequence

In contrast to the first Arc sequence above, the images quality of this sequence is degraded. The size of the tracked particle is very small which made even the manual tracking difficult. The sequence is composed of 55 images with 720×576 resolution. The average displacement between two images is approximately 9 pixels. In Figure (2.20, top), the mean SGSD vector field is superimposed on one of the images. In the bottom of the same figure we show the direction and the magnitude for every pixel in the form of the color code. The spatial coherency of the estimated vector field is easily observed. The color is lighter than the first arc sequence because it is affected by the large outliers in the white triangle at the bottom. One could notice non-zero vectors in the non-fluid area in the bottom left part of the image. These are due to the wind moving the grass and not due to estimation errors. In Figure (2.21, top) we can see that SGSD was able to recover a better vector field in terms of magnitude and direction than PIV. With (HS), both vector fields seem to have the same magnitude, SGSD however was able to recover much better direction. In the same figure, we show the normalized distance error of different methods (middle) and for SGSD using different values for Sc_t (bottom). A larger diffusion coefficient gives better results. However, after 40 images it starts to diverge completely to a poor result with approximately 0.13 normalized error. The default SGSD is more stable throughout the whole sequence and most of the time better than the other two cases with no diffusion or with a small diffusion coefficient. Divergence field and trajectories on an image cross-section are shown in Figure (2.22).

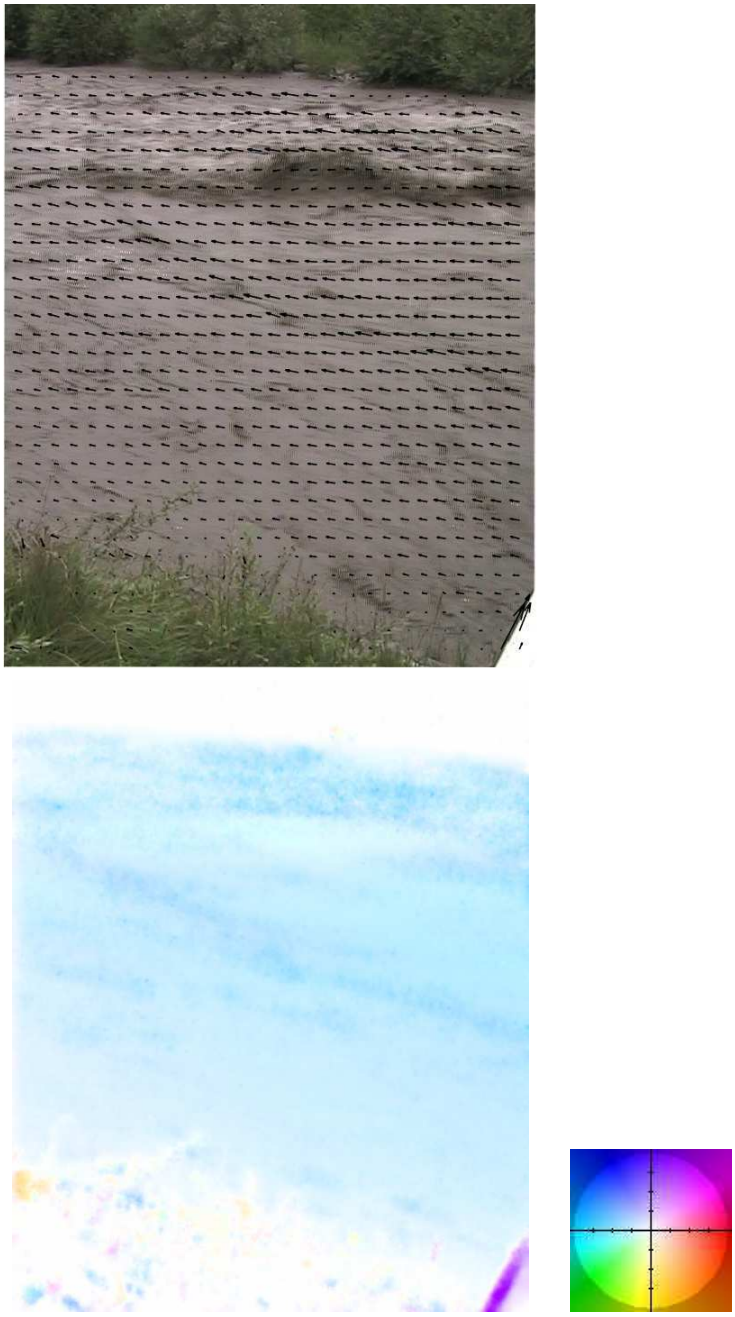

Figure 2.20: Top: SGSD mean field for second Arc sequence super-imposed on the first image, only one vector is plotted for every 20×20 region of pixels for visualization purposes. Bottom: SGSD result for all pixels in color code format.

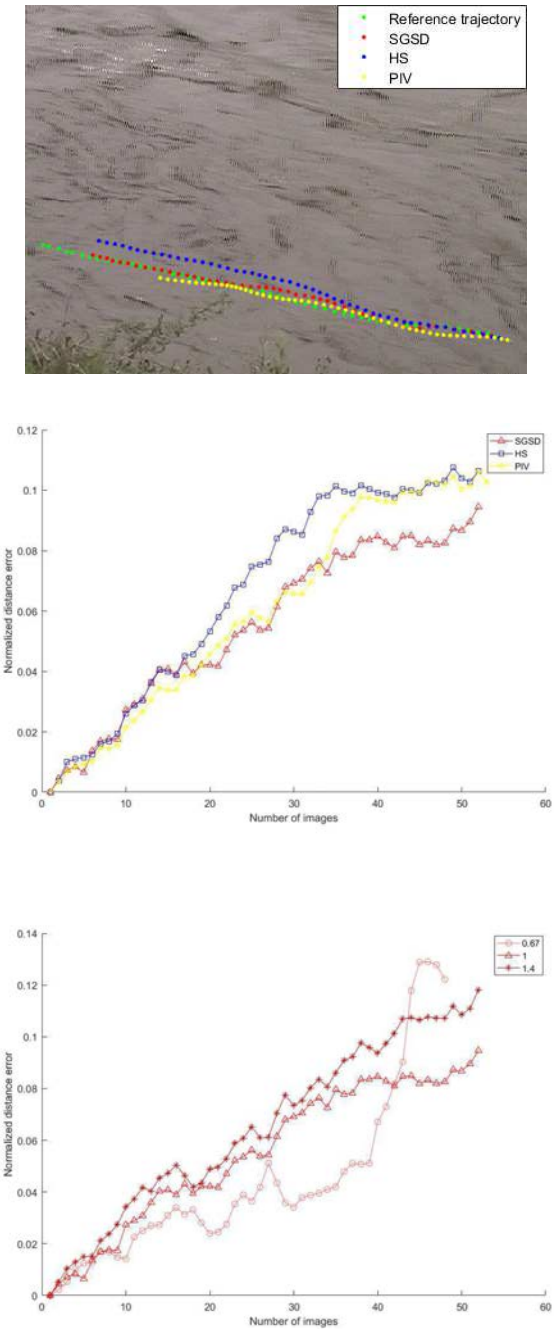

Figure 2.21: The trajectories of different methods superimposed on the first image of the Arc river second sequence (top). Normalized distance error of different methods (middle). Normalized distance error of SGSD with different turbulent Schmidt numbers (bottom)

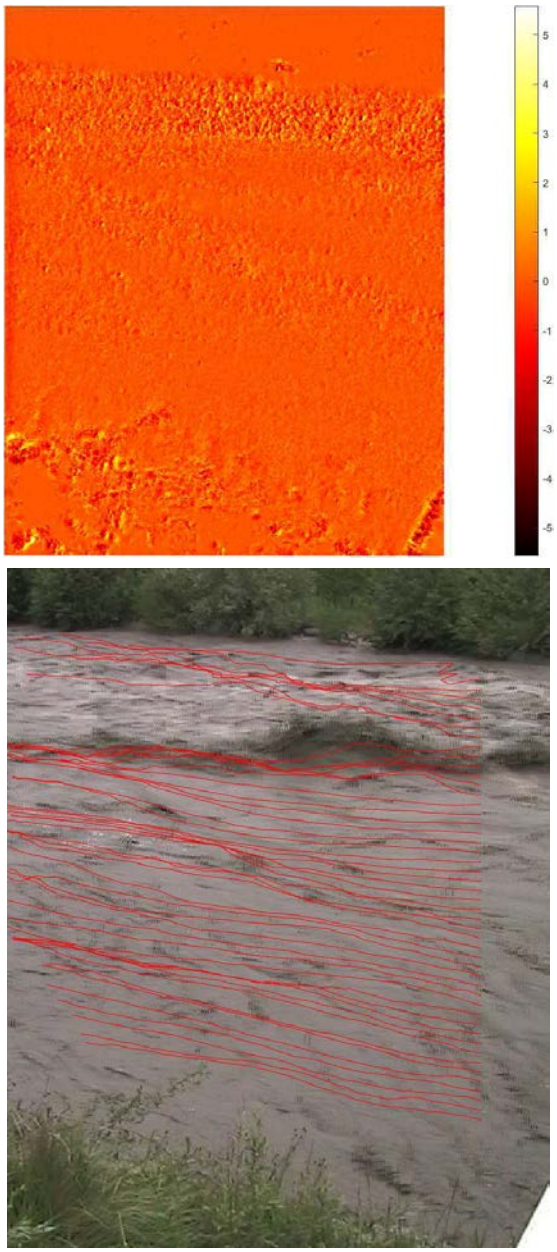

Figure 2.22: Other relevant quantities derived from SGSD for the Arc river second sequence: divergence of the mean vector field (top). Trajectories on a cross-section (bottom).

The Gave De Pau river sequence

The Gave de Pau river, located in the french Pyrenees, is more challenging than the previous cases. The sequence is composed of 49 images with 471×314 resolution. The average displacement in the streamwise direction of approximately 6 pixels. The sequence has many uniform regions with weak gradient signal. Furthermore, the tracked particle is big in size which increases the uncertainty of trajectory reconstruction. This is because many points in the vicinity of the original point resemble each other. In addition, the chosen particle area (and a great deal of other parts) suffered from severe 3D deformations which make the particle moves up and down. This clearly violates the 2D incompressibility condition used to derive the optical flow model. The effect could be seen in the highlighted rectangle of trajectories in Figure (2.25, bottom). In Figure (2.23), we get to have a general idea of the sequence motion patterns by plotting the mean SGSD vector field. The 3D deformations mentioned earlier are readily noticeable when visualizing the dense vector field. SGSD was able to recover a better trajectory compared to other methods, Figure (2.24, top and middle). Testing different Sc_t values other than the default value 1 showed similar performance for all variations until around the 20th image where the default SGSD showed better performance while SGSD with a larger diffusion coefficient gives a trajectory that diverges drastically to a bad local minimum with approximately 0.5 normalized error, Figure (2.24, bottom).

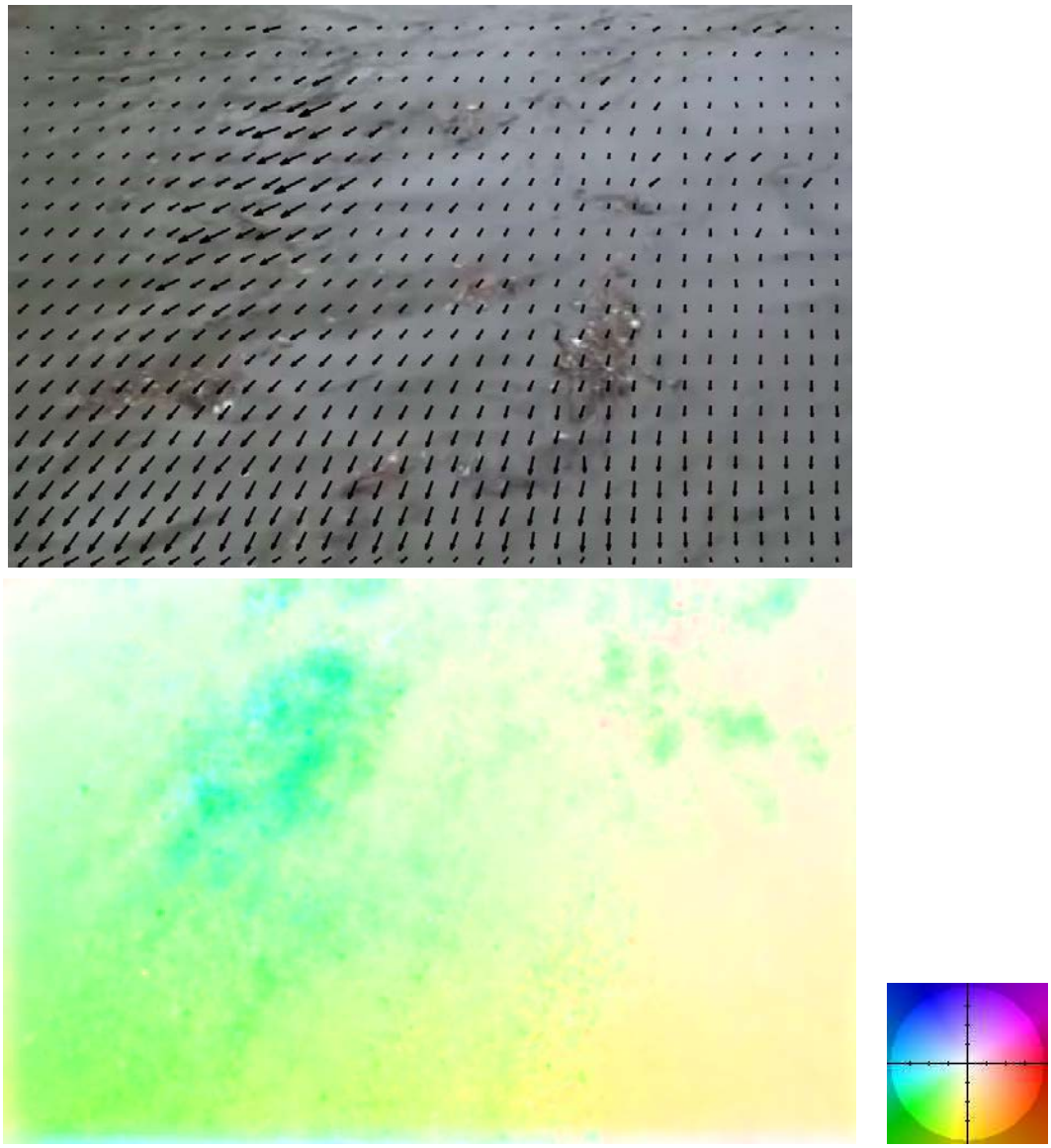

Figure 2.23: Top: SGSD mean field for Gave de Pau sequence super-imposed on the first image, only one vector is plotted for every 20×20 region of pixels for visualization purposes. Bottom: SGSD result for all pixels in color code format.

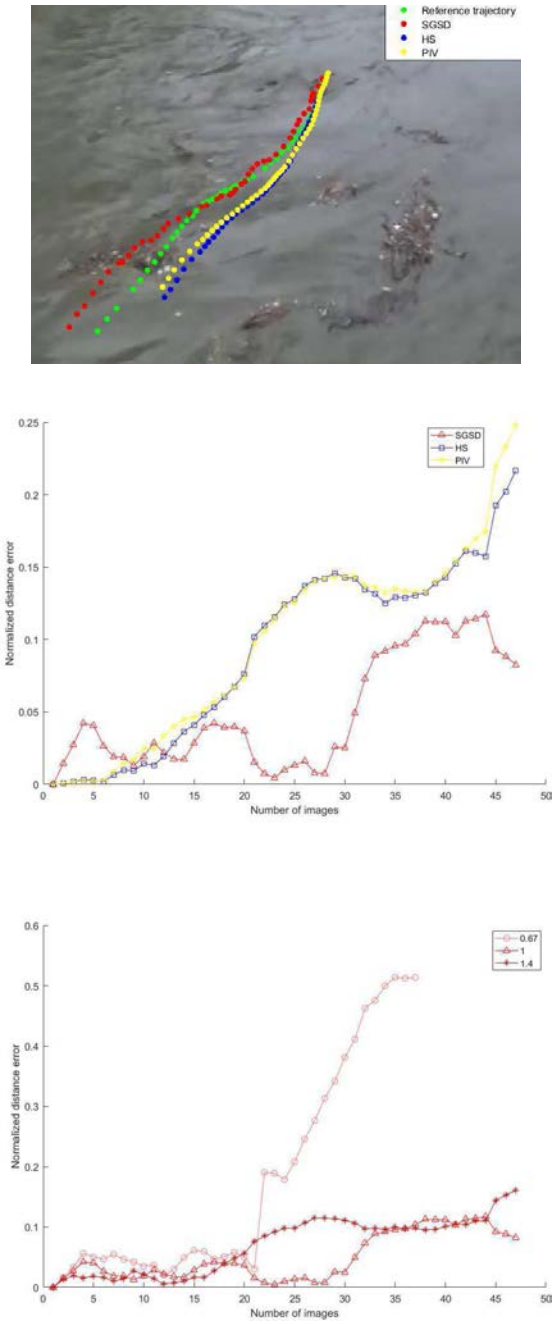

Figure 2.24: The trajectories of different methods superimposed on the first image of Gave de Pau river sequence (top), normalized distance error of different methods (middle) and the normalized distance error of SGSD with different turbulent Schmidt number (bottom).

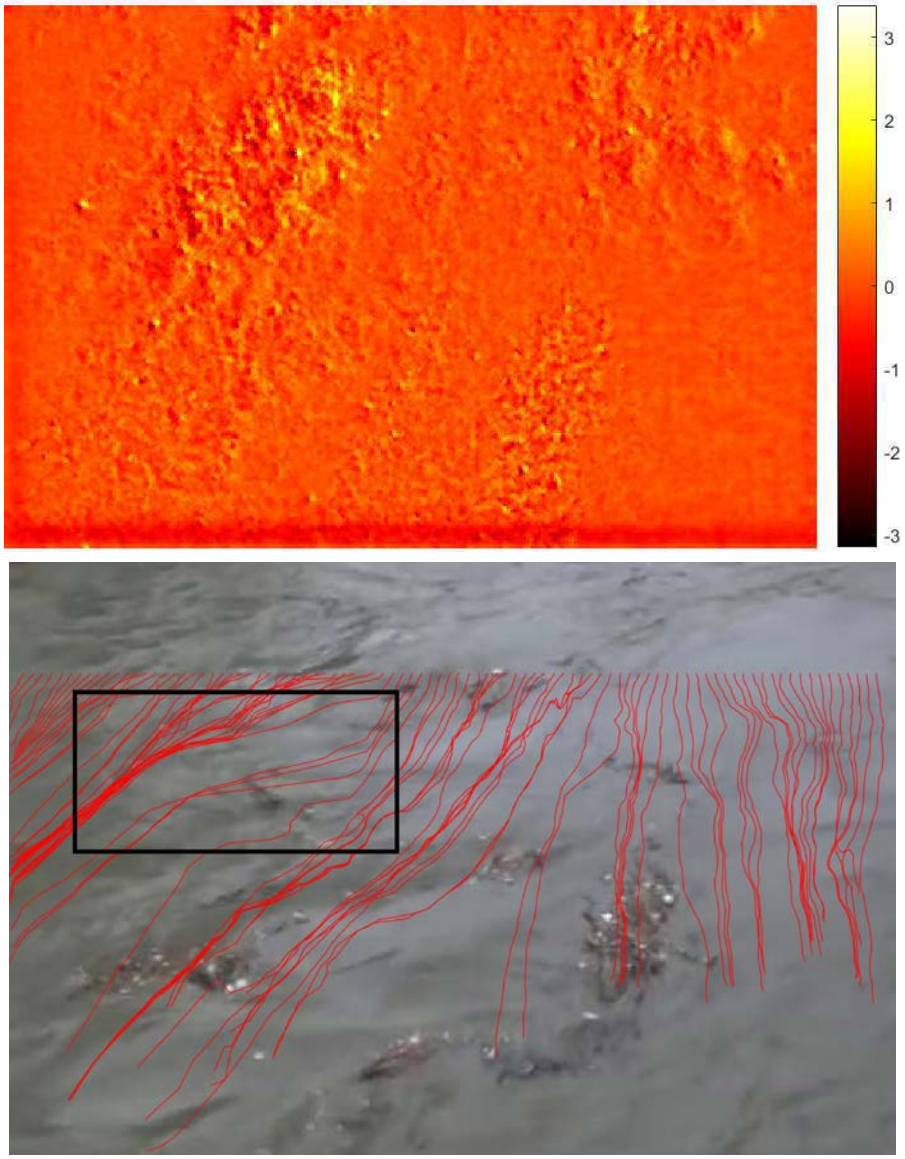

Figure 2.25: Other relevant quantities derived from SGSD for the Gave de Pau sequence: divergence of the mean vector field (top). Trajectories on a cross-section (bottom).

Effects of initialization

We further investigated the results on areas with uniform intensity values and weak gradients. Both PIV and optical flow data term would suffer in this case because they rely on these intensity variations. However, using optical flow, one could still get reasonable estimates since the problem is more constrained by the regularization term. In addition to that, by initializing the vector field using previous estimates in the sequence, optical flow could propagate any prior information about these areas that might have been obtained earlier in the sequence. This can not be demonstrated in a straightforward way since it is not possible to reconstruct a reference trajectory from a uniform intensity location with no particle to track. However, a simple workaround is possible under some assumptions. Consider an image sequence I in which a member image might be described as $I_n, 1 \leqslant n \leqslant N$ where n is image index and N is the total number of images. Let I_1 contains a particle at some location S, when n increases, the particle moves to a different location in other images every time. It follows that we can manually reconstruct the reference trajectory for this particle on location S *only* starting from I_1. We could now run SGSD method on two different set of images. The first one is the original sequence I and the second is on a partial sequence I^p that starts with, say I_5 and finish with I_N. We then proceed to reconstruct two trajectories on image location S starting from I_5 (or I_1^p for the second sequence) based on these estimations. Note that I_5 and I_1^p are the same image in which the location S contains no particle. The difference between the two trajectories is in the estimated vector fields. The first vector field computed on I has some information on the expected solution at the chosen location S since it contained the particle in earlier frames (specifically the first frame). The second one is computed on I^p and is initialized without any prior information. Assuming the river trajectory on S will not vary much in a short amount of time (less than a second), we could compare the estimated trajectories to the reference trajectory. In Figure (2.26), we plot the reference trajectory (in green) on image I_5. We can see that it starts from a uniform location and the particle of interest has already moved to the 5th

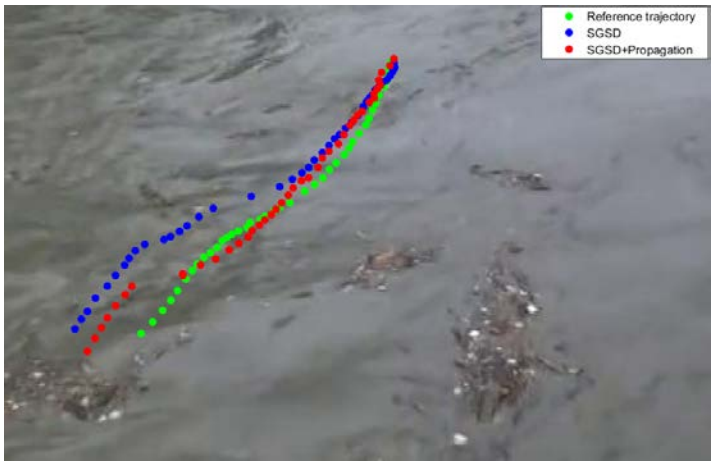

Figure 2.26: Trajectories based on a uniform area.

point of the reference trajectory. The trajectory in red is the one with prior information while the blue is without it (relies only on the regularization). In both cases we see that they are not completely different, the red trajectory being better than the blue with the help of the information propagation process realized via the initialization. We draw the attention to the fact that the process of trajectory reconstruction is very sensitive to outliers. One outlier is enough to make the trajectory to go to a different direction. At the same time, when having a uniform intensity, gradient-based velocity estimation in images is very prone to outliers. Still, we were able to reconstruct similar trajectories to a certain degree, thanks to the regularization term and the prior information propagation.

2.7.3 Results on ortho-rectified images

In river velocimetry, LSPIV method is the benchmark method to estimate free surface velocity. It applies PIV to a grid of points defined in ortho-rectified images. Image ortho-rectification is a way to transform the images to remove the perspective effect so that a direct relationship between image space and the real world could be established. The resulting transformed sequence is used to compute river velocity using image-based techniques like PIV. Unfortunately, the philosophy of LSPIV in comparison to optical flow is

different which makes the comparison difficult. LSPIV relies on estimations obtained on many images and use statistics (the mean mostly) to average the final field in a first step. In a second post-processing step, outliers might be deleted on constraining the magnitude and/or correlation score. The instantaneous velocity field without post-processing is generally noisy because the estimation is carried out locally and if these points happen to be in a uniform area, they would generate outliers. This makes trajectory reconstruction difficult because it uses successive individual estimations and only one outlier is enough to make the whole trajectory divert. In addition, one needs to interpolate between points on the vector field to reconstruct the trajectory. This interpolation is different between the two methods since it is going to be only between adjacent image pixels for optical flow or between grid points (a distance of many pixels) for LSPIV. Optical flow integrates outlier rejection in the process via median filtering and/or regularization. To compare the two methods, we run SGSD on the ortho-rectified sequence and we compute a real world velocity. We then superimpose the same LSPIV grid on SGSD dense field and compare velocity values for grid points in both methods. In Figure (2.27, top), we reconstruct trajectories using SGSD of points happened to be at the grid or very close. The one in the middle follows closely the ground truth. The one in top seems to underestimate the magnitude a bit while the one in the bottom seems to overestimate it. The three trajectories showed perfect direction estimation. In Figure (2.27, bottom), the absolute difference of SGSD and LSPIV estimations on the grid is shown. The difference is close to zero on many points. However, a noticeable difference is observed on the bottom side of the grid. This is not surprising since estimations on image boundaries are always prone to errors, especially that here the flow next to image boundaries is not seeded with particles. This might explains the systematic difference in the PDFs of the two estimations shown in Figure (2.28), even if the overall shape of both functions is similar.

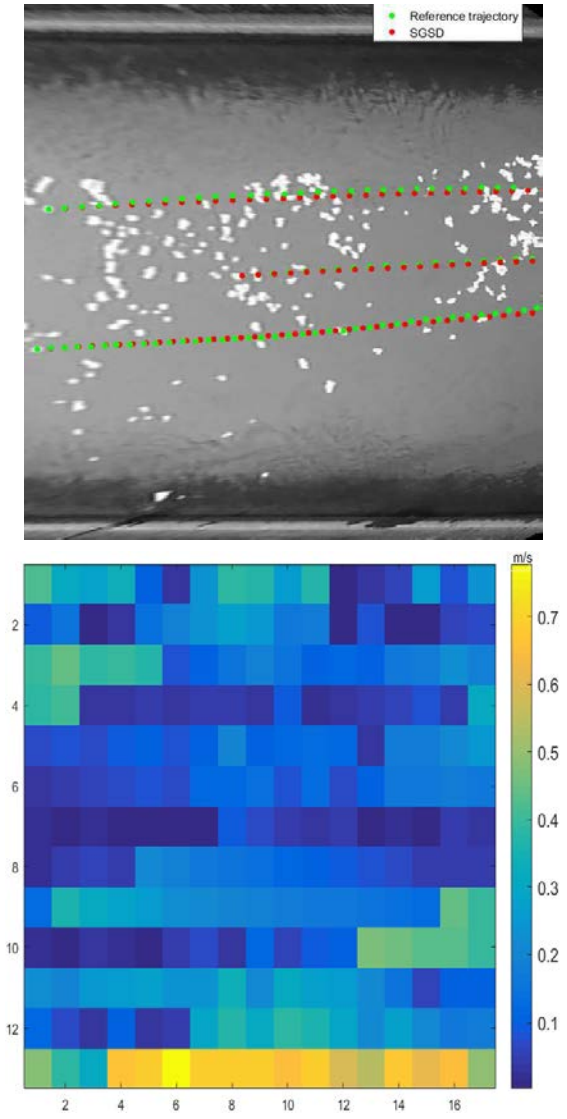

Figure 2.27: SGSD trajectory on the 3 different particles (top). Absolute difference in u component between SGSD and LSPIV (bottom).

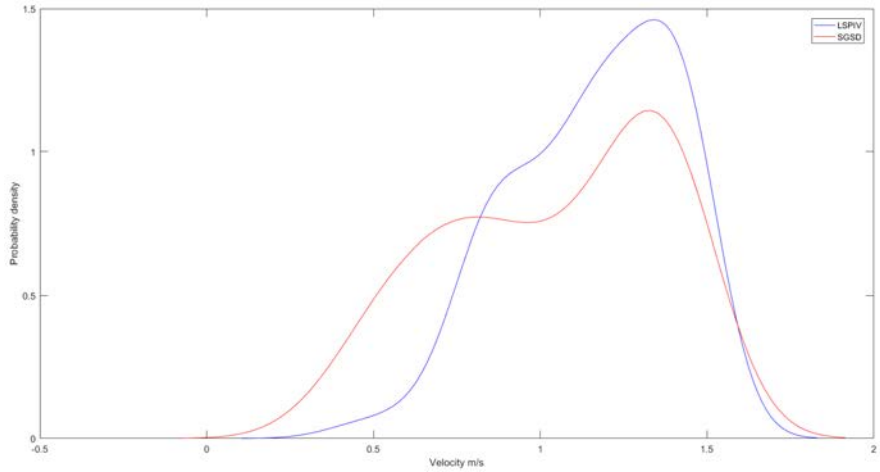

Figure 2.28: A PDF for both LSPIV and SGSD estimations on streamwise direction.

2.8 Conclusion

We presented a novel method for the application of image-based river velocimetry, based on optical flow scheme. We showed that optical flow in its own right is optimal and physically compatible with rivers free surface motions. In the ideal settings as in the first Arc sequence, section (2.7.2), original optical flow provided very good results without any additional terms. Our enhanced model is based on the decomposition of the scalar transport equation which is equivalent to the original optical flow in addition to a new term. This term, considered to be related to turbulent viscosity, models the small scales contributions that are dropped out in the image acquisition phase. We suggest a turbulent diffusion coefficient based on *Prandtl* Mixing Length model. We presented results on different possible cases of rivers with or without tracers, good or bad imaging conditions, high or low surface velocity and high or low surface perturbations. Our method outperformed all other approaches in all image sequences in the recovered magnitude and/or direction. Both assessed visually or statistically via trajectory reconstruction of particles of interest. We draw the attention to the fact that any wrong value at any instance of the estimated vector fields sequence will ruin the part of the trajectory that follows it. It is also sensitive to systematic errors

that accumulate in time. Nevertheless, we have been able to reconstruct trajectories that are similar to the reference trajectory. It is shown that optical flow can still give plausible results in sequences with weak gradient signal via regularization and/or information propagation. It is also shown how the dense estimation of optical flow facilitates the computation of other relevant fine-detail quantities like the divergence for instance. Since the application treated here mainly focuses on river velocimetry, our model is designed to penalize the vorticity in the vector field. Only cross-section trajectories and divergence of the flow field were then derived and shown. However, the variational formulation is flexible and could open the door to more specialized optical flow models for rivers. Indeed, if the vorticity field is of an importance, the model could be easily changed to avoid the penalization of this quantity during the estimation process (Chen et al., 2015). A high resolution vorticity field could then be obtained from the estimated victor field very easily. Using trajectory reconstruction evaluation method feedback, new data terms/regularizers and models for the diffusion coefficients could be tested and assessed.

Chapter 3

Geometric modelling

When capturing a video of a dynamic scene, the 3D motion is projected to a 2D space of successive images. In addition, if the camera was moving during that time, the obtained images contain a mix of different motions. In computer vision, many applications are actually interested in the reverse process i.e. how to go from 2D images back to 3D. We show in this chapter that this is indeed our case.

An image sequence depicting the flow of a river is also a special case scenario. Its particularity comes from the river part of the image which is a blind spot for traditional geometric modelling methods. Not only because it is extremely difficult to detect corresponding points on the river surface but also because this part of the image is constantly moving in a non-rigid fashion. We first investigate what computer vision has to offer and we proceed by showing what suits best the particularities of the application at hand. We investigate many important concepts in computer vision that are related to how cameras do capture the world into images, but also how to exploit these images to infer a 3D world.

3.1 Introduction

In this chapter we talk about the geometric modelling of river sites. This includes the 3D structure of the site in addition to the camera and its movement. These are needed to transform river surface velocity, obtained only in image space as described in the previous chapter, into a meaningful 3D world metric velocity. This 3D world velocity must not be confused with river 3D velocity. The apparatus considered here is an ordinary camera that only sees the water free surface and nothing in the depth direction of the river. Hence, we make the basic assumption that the river free surface is a plane, in which we try to estimate the velocity, Figure (3.1). Let's imagine the ideal situation where an orthographic camera might be taking a video of a perfectly plane river surface orthogonally, i.e. river free surface points vary only in X and Y directions, in camera coordinate system. The 2D image points of river surface obtained in this configuration form a 2D to 2D mapping with their counterparts in the real river surface. Only a scale factor is needed to pass directly from image velocity computed in image space to world velocity. This is the simplest case since the 2D mapping is easy to estimate, provided that the 3D coordinates of some river points along with their 2D projections are known. LSPIV method thus *forces* an ideal situation by ortho-rectifying image sequences. This process produces images with predefined pixel/meter ratio and no perspective effects. One could compute directly world velocities from these images. However, the ortho-rectification process interpolates and possibly changes the image intensity values. It does not seem a good idea to alter the only source of information available for velocity estimation. In addition, the ortho-rectification process needs considerable field work to install Ground Reference Points (GRPs). This severely constrains the number of sites on which such measurements could be performed.

In reality, most river amateur videos do not fall in this ideal case and the camera is generally perspective (not orthographic) and is oriented at an oblique angle to the river. Consequently, in the same camera-centered coordinate system, the 3D points on the plane will have different depth values. This case is more difficult since the mapping is now 3D to 2D, represented

by the perspective projection which is not invertible. There is a loss of information that will hinder the reverse process, i.e. going from 2D back to 3D.

Figure 3.1: Systems of coordinates

The basic relationship between a 3D point with its projected 2D image is illustrated in Figure (3.2). The ratio x/f equals X/Z.

$$x = \frac{fX}{Z}, \; y = \frac{fY}{Z} \tag{3.1}$$

where f is camera focal length.

Let's suppose we have a point $P(X, Y, Z)$ moving in front of a camera, this will create a path relative to the camera in the form $P(t) = (X(t), Y(t), Z(t))$. The derivative of this path is equal to the instantaneous velocity in the 3D world $\omega_{3D} = \left(\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt}\right) = (U, V, W)$. The analogous process to the 2D projected path $p(t) = (x(t), y(t))$ will yield the instantaneous 2D velocity in the image $\omega = \left(\frac{dx}{dt}, \frac{dy}{dt}\right) = (u, v)$. Similar to the set of equations in (3.1) we

can derive:

$$X = \frac{Zx}{f},\ Y = \frac{Zy}{f}$$

differentiating both sides of X equation $w.r.t$ time we get:

$$\frac{dX}{dt} = \left(f\left(Z\frac{dx}{dt} + \frac{dZ}{dt}\right) - \frac{df}{dt}Zx\right)/f^2$$

which will evaluate to

$$U = \frac{Zu + Z' - X}{f}$$

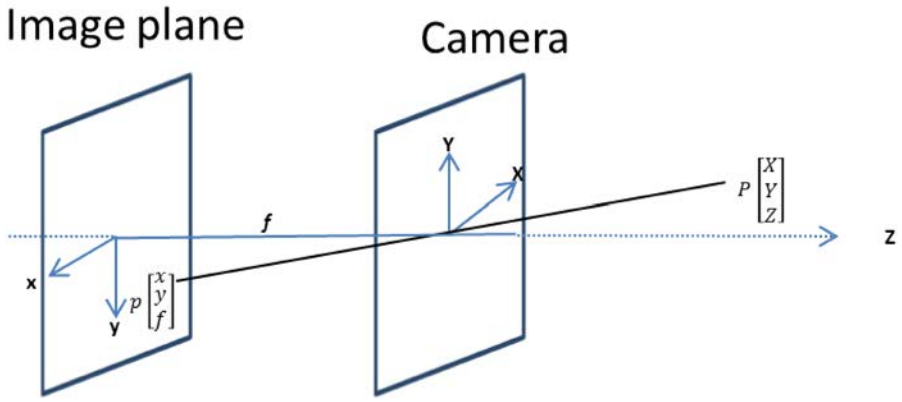

Figure 3.2: Perspective projection

A similar equation for V could be obtained. From the above, it is obvious that the 3D position information of points and focal length of the camera are indispensable to compute world velocity from 2D velocity.

In amateur videos of rivers, we have an individual with an imaging device (smartphone or a camera) capturing the river in motion. We are interested in the general case where the individual happened to witness the event and then felt it is a good idea to capture it on video since it is not an everyday encounter. This rules out individuals with special lenses (fish eye etc.). In this case we expect the camera to not be stable. The motion might come from the natural movement of the individual hand or from an explicit movement of the individual himself while wandering the scene. The resulting images will have motion almost on every pixel due to these motions. Nevertheless,

we can distinguish two regions in the resulting images: fixed image regions where the fixed parts of the scene (buildings, bridges, etc.) are projected to, and a river region. There is a third category we often see in these sequences that is related to vegetation areas. It is easy to decide if it may be considered a fixed region or not by just observing the video. The fixed regions will help us infer the motion of the camera since these objects are fixed in the 3D scene, their 2D image motion is due solely to camera motion. Any estimation of river velocity in image space will be erroneous if the camera motion is not accounted for. Stabilization of these image sequences is needed and the estimation could only now be performed. Obviously, different sequences will have different proportions of the two aforementioned regions. Too many fixed regions is a problem because the object of interest here is the river itself. This happens when the river is too narrow or when the camera is far away. The second case is when no or only small fixed regions are captured. This is even worse than the first case because the camera motion estimation depends on these regions. One advantage of amateur videos however is their good frame-rate (at least 24 frames per second). Consequently, one should expect subtle camera motion between consecutive frames which makes the stabilization easier. Despite camera motion being an issue as far as velocity estimation is concerned, we emphasize that having a moving camera is actually and quite surprisingly an indispensable requirement for sequences taken by unknown amateur cameras. Techniques of *stereo* or *multiple view* geometry are only applicable in scenes with camera motions (or equivalently, fixed camera and scene rigid motion). These techniques are vital for camera parameters estimation. Next section will give some notions about camera calibration and other important concepts related to stereo and multiple view geometry. Then, details on auto-calibration techniques and the framework chosen are given. The last section will give details about general 3D reconstruction and a custom 3D reconstruction approach that suits rivers sites.

3.2 Preliminaries

The purpose of this section is to introduce readers from outside computer vision to concepts and terminology that are frequently used in the field. These concepts will be used in later sections of this chapter. If the reader is familiar with computer vision, he/she could safely skip this section.

3.2.1 Feature-based correspondence

Correspondence between images is a common task in a lot of applications in computer vision. In chapter two, we established a *global* correspondence between two images in the form of optical flow. In this chapter, we rather seek discrete points pairs correspondences to be used in stereo and multiple view geometry context. Whether for triangulation of 3D points, or for the estimation of camera calibration and/or *epipolar geometry* entities, high accuracy correspondence is a must. As the name suggests, the correspondence is based on a sparse set of features that could be identified in images. It is a somewhat mature area of computer vision and many very reliable methods were developed.

Individual pixels do not have much information, a pixel is therefore described within a group of pixels taken around it. The first step is to detect interest point candidates in images. Then, a feature descriptor is created. It is a vector on which we stack information induced from the ensemble of pixels in an image window. This information could simply be the intensity values of pixels but in most cases these are not discriminative enough. The research in this area focuses on what type of information is robust against photometric and geometric changes (rotation, scaling, etc.) that could be used in the feature descriptor vector. The criteria on which the correspondence will be based upon is the distance between these vectors. The smallest the distance, the more likely the two points correspond to each other. Beside the Euclidean distance, there are different types of distances that might be used (Mahalanobis, 1936, Bhattacharyya, 1943). Some distances are more suited than others to some specific cases. One commonly used algorithm is Harris corners (Harris and Stephens, 1988). Corners represent a good feature point

candidates. Imagine a window centered on a corner in an image. Any movement for this window will induce significant change in the intensity profile of that window. This is a desirable property because it facilitates detecting the same point again in other images. This is not the case for example for windows centered on an edge because there will not be significant change in the intensity profile if the window moves along the edge. Consequently, windows located on edges are less distinguishable. The worst case scenario happens when a window is based on a homogeneous area of intensity with no edges nor corners. This is actually a quite frequent situation specially in man-made world. Many objects are of uniform color (walls , cars, etc.). A window located on the center of such objects will hardly has its intensity profile changed if moved. Harris corners algorithm defines a mathematical way to model corner response inside a window in an image. It is invariant to rotation but not to scale. Years after Harris corners algorithm, SIFT (Scale-Invariant Feature Transform) became the benchmark for feature based algorithms (Lowe, 2004). The idea behind it revolves about the scale. For example, if the corner detected in the first image is scaled (zoomed in) in the second image. We need a bigger window to detect the same corner response. The original windows size would most likely only catches one edge of that corner or a different size corner in other images (even if it is the same corner in 3D). To this end, a scale-space approach is used in which the Difference of Gaussians (DoG) is obtained from the image with different standard deviations. The Gaussian is just a blurring algorithm performed by convolving the image with a Gaussian kernel. The DoG detects features at different scales due to the change in the image with different blurring degrees. In other words, we can now find the same corner across the scale and space. The second step in the algorithm is to refine the candidates found based on various criteria. To achieve rotation invariance, the third step consists of creating an orientation histogram with 36 bins covering 360 degrees. This histogram is created based on the magnitude and direction of the gradient computed in a neighbourhood, the size of which depends on the scale chosen for the key-point. The highest peak is then taken as the orientation. This step contributes to the stability of matching. Finally, the feature descriptor

is created based on 16x16 neighbourhood around the point. This is further divided into 16 sub-blocks. For each, 8 bins orientation histogram is created. The feature vector for every point will then contain 128 values. In practice, we used a free and speeded up version of the algorithm called SURF (Speeded-Up Robust Features) (Bay et al., 2008).

Still many outliers may remain, a common practice is to use RANSAC (Random Sample Consensus) which could be seen as an outliers detection algorithm. The idea is simple, the algorithm takes the minimum number of data points to fit a specific model. The remaining data points are evaluated to see if they represent inliers or outliers to this model. This process is repeated and the iteration with most inliers is then taken. In river image sequences, we mainly use feature-based correspondences to model the relationship between the world and/or the different cameras (or the same camera in different locations and time). As a consequence, the correspondence is only established in the fixed regions of the image. Any correspondence on the river surface is not taken into account as it is polluted by the motion of the river which is independent from camera.

Figure 3.3: Top: SURF feature points detection. Bottom: Red-Cyan anaglyph of two images with the correspondence of their respective feature points.

3.2.2 Camera calibration

A camera is said calibrated if certain parameters that regulate the imaging process are known. Let's consider an arbitrary object of interest. It has an arbitrary set of coordinates belonging to world system of coordinates where it can be described in. We denote this system of coordinates with the subscript W. For this object to be projected to a 2D image, an imaging device has to be introduced. The imaging device has its own set of coordinates in the world coordinates system but also forms a new system of coordinates centered at the camera projection center. We denote this one by the subscript C. Now if the object is in the view field of the camera it will be projected into the camera 2D retina. But this is only occurs after being expressed in camera coordinate system. Finally the points in camera retina are transformed into the image in forms of pixels. We talked earlier about projecting a 3D point into the image retina, Equation (3.1), Figure (3.2). We realize now it is only one part of a series of transformation from 3D world coordinates system to 3D camera coordinates system to 2D retina and finally to pixel coordinates. The ideal scenario would be to combine these transformation into one transformation. To that end, we need to express the relationships in Equation (3.1) into matrices. Even if these are non-linear equations, we can still express them with matrices using *homogeneous coordinates*. In homogeneous coordinates, a 3D point is expressed using a 4D vector. Similarly, a 2D point is expressed using a 3D vector. One could get the original vector by dividing all coordinates with the last component of the same vector. Accordingly Equation (3.1) could be rewritten as:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Then, an affine transformation from camera retina coordinates to image pixels has to be performed. In camera retina the origin is set at the center, while in image pixels grid the origin is at the corner. The perspective equa-

tion changes to :

$$x = \frac{fX}{Z} + O_x, y = \frac{fY}{Z} + O_y$$

where O_x and O_y are the offsets needed, both form the so-called *principal point* $pp = (O_x, O_y)$. In addition, there are scale factors in x and y directions of the pixels grid. They determine the form of the pixel, different scale factors will result in non-squared pixels. One last parameter is the skew factor, it has a non-zero value if image axes are not exactly orthogonal.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & O_x & 0 \\ 0 & f_y & O_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

This transformation matrix is the camera intrinsic matrix K, where f_x and f_y are the focal length values in x and y axes for non-squared pixels. However, most recent digital cameras are more or less squared, one can safely assume the same value f for both x and y axes. The skew factor s is often neglected while modelling the camera matrix. There are also non-linear parameters to model the distortions. None of these parameters is considered in the camera model chosen throughout the thesis.

As mentioned above, in order for this transformation to be possible, the object has to be expressed in camera coordinates system. This is achieved by a change of coordinates transformation that collides the origins of camera and world coordinates systems. It is a rigid transformation that consists of a rotation R to align the two set of axes and a translation t to align the two origins, Figure (3.4). These are called the extrinsic parameters. In matrix form and again using homogeneous coordinates it reads:

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

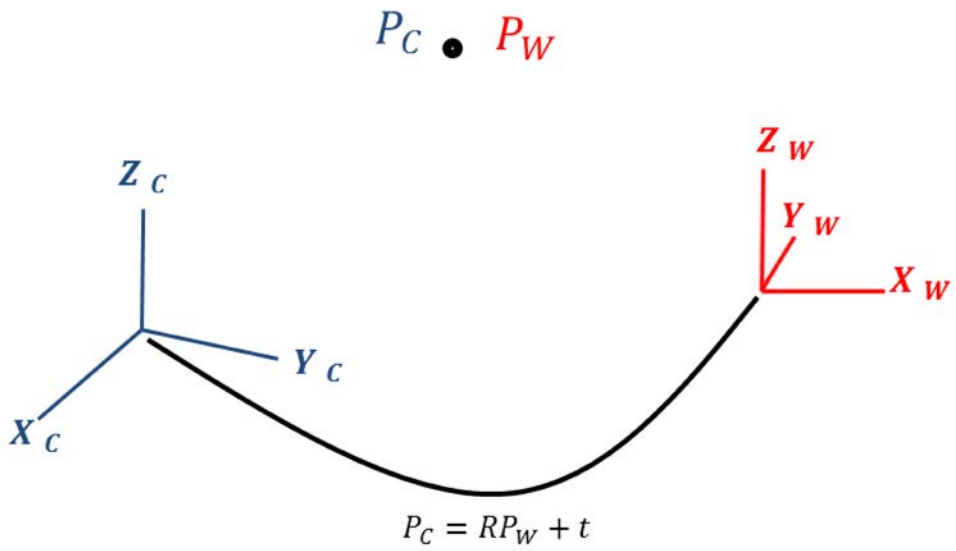

Figure 3.4: Extrinsic parameters, The two origins of the two systems of coordinates are related by a rotation R and translation t.

The knowledge of both intrinsic and extrinsic parameters is what is called camera calibration. Combining all these transformation lead to the well-known equation:

$$p = K\,[R|t]\,P = QP \tag{3.2}$$

Conventional camera calibration process consists of finding a unique Q by tracking known-geometry object in 3D and its projection across images and solve for Q in a non-linear way (Tsai, 1987). Zhang (2000) suggested a simpler way by utilizing a simple planar pattern shown at different orientations. One could simply print the pattern, attache it to a planar object, take few photos and then calibrate with very good accuracy. There is a direct linear estimation method in which there is a known correspondence between 3D points and their 2D projections. A linear system of equation is built around the coefficient of Q (only 11 parameters because Q is defined up to an arbitrary scale). At least 6 point is needed to estimate Q. FUDAA-LSPIV software (Le Coz et al., 2014) use this method to estimate Q exploiting the GRPs. Q could be later decomposed into its intrinsic and extrinsic matrices

using QR decomposition. Note that conventional calibration is not possible when only the image sequence is available (no real physical camera). This led to the rise of Auto-calibration techniques where the camera is calibrated only by using the information contained within the sequence taken by that camera.

3.2.3 Epipolar geometry

Now after we have seen how 3D points are projected in 2D image points, let's start to slowly examine the inverse problem which is going back to 3D. It is a more difficult problem because when a 2D point is back-projected to 3D space, it traces a line in which any point would satisfy the original perspective projection equation. In order to determine the right location of the 3D point, a triangulation technique is used in which the correspondent point in the second image traces a second line in 3D space, the 3D point location is taken as the intersection point of the two lines. This is the core idea of *stereo vision*. The accuracy of the triangulation depends on different factors. First of all, camera parameters have to be accurate since the back-projected rays depend on them. Secondly, a wide *baseline* is preferred. The baseline is the line between the two cameras/views centers. A narrow baseline might cause the back-projected rays to go parallel. Evidently, this is not the best scenario to look for an intersection point. This degenerate case is particularly seen in image sequences with small camera motion. Unfortunately, we find this situation a lot in rivers image sequences. Lastly, the accuracy of the correspondence itself, obviously back-projecting rays of different points will result in bad 3D reconstruction. It is worthy note that, in the narrow baseline case, the correspondence problem is easier than wider baselines as the correspondent point will be in the vicinity of the original point. This shows the difficulty associated with the baseline. One needs wider baseline for better triangulation but then an accurate correspondence (which is the basic building block for the triangulation) is more difficult.

Epipolar Geometry governs the intrinsic relationship between two views. It depends only on camera's internal parameters and relative pose and is to-

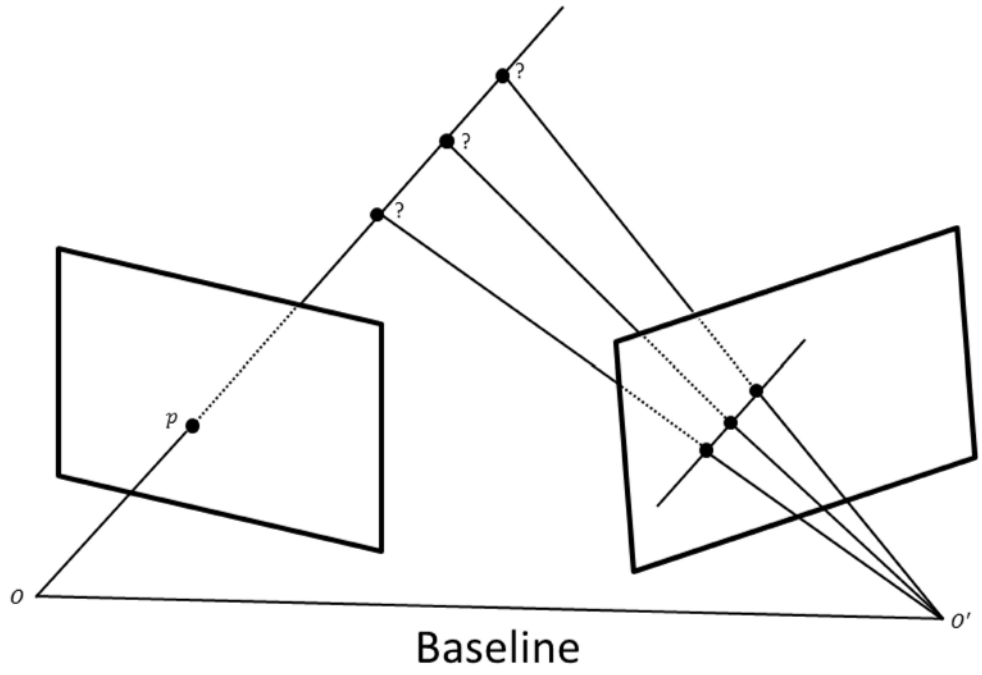

Figure 3.5: Epipolar constraint, p back-projected 3D ray will project to a line in the second image.

tally independent from scene structure. Let's take two correspondent points in two images taken by two cameras. These points are correspondent because they are the 2D projections of the same 3D point. Now if we only have one point and we wish to know the correspondent point in the second image, we need to determine the exact 3D point that projects to these two image points. The one 2D point we have at hand can not determine its true 3D point uniquely, it draws a line in 3D space in which *any* 3D point will project back to it, Figure (3.5). Note that a segment of this infinite line is visible to the second camera and is projected to its image plane. The correspondent 2D point then has to lie on that line in the second image. The 3D point P, camera centers O and O' and the points p and p' all lie in a common 3D plane, Figure (3.6). Let's define some terminology:

- Epipolar plane: A plane that contains the baseline.
- Epipole: The point joining the baseline to image plane.
- Epipolar line: The intersection of an epipolar plane with an image plane (equivalently, it is the image of the back-projected ray from the correspondent point in the other image).

There is a family of epipolar planes all sharing the baseline (and the epipoles) with the difference being in the choice of the 3D point which in its turn changes the epipolar lines. The epipolar lines in every image pass by the corresponding epipole.

In the case of calibrated setup, one could use the K matrices to normalize the coordinates of points (remember every image point is expressed in the coordinate system of its own camera) i.e. $\hat{p} = K^{-1}p$ and $\hat{p}' = K'^{-1}p'$. Additionally we set one of the camera centers to be the origin of the normalized coordinate system. It follows that there is a rigid transformation in form of rotation and translation between systems of coordinates of the two views i.e. $t = O' - O$ and $\hat{p}' = R\hat{p} + t \Leftrightarrow R^T\hat{p}' = \hat{p} + R^T t$ using the fact that R is an orthogonal matrix having its inverse equal to its transpose. The coplanarity mentioned above could now be expressed using R and t because all of the vectors $\hat{p}$, t and $R\hat{p} + t$ are coplanar. It could be expressed as the inner

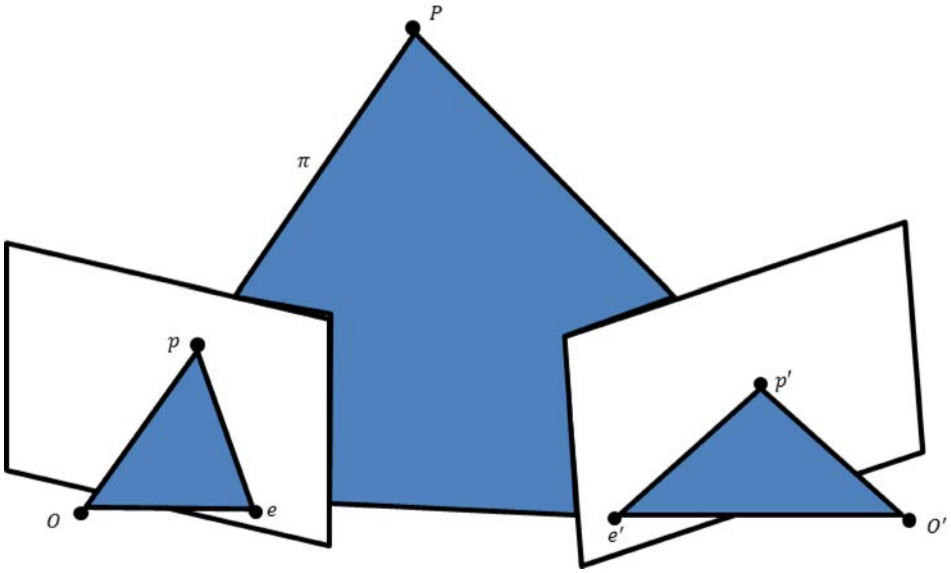

Figure 3.6: The epipolar plane.

product of one vector with the result of the cross product of the two other vectors.

$$\left(R^T \hat{p}'\right)^T (t \times \hat{p}) = 0$$

Now let's introduce a new notation for the cross product.

$$\overrightarrow{c} = \overrightarrow{a} \times \overrightarrow{b} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix}$$

This could be reorganized into

$$\overrightarrow{c} = \underbrace{\begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}}_{[a_\times]} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$$

Using this new notation for cross product the coplanarity constraint becomes

$$\left(R^T \hat{p}'\right)^T [t_\times] \hat{p} = \hat{p}'^T \boldsymbol{\varepsilon} \hat{p} = 0$$

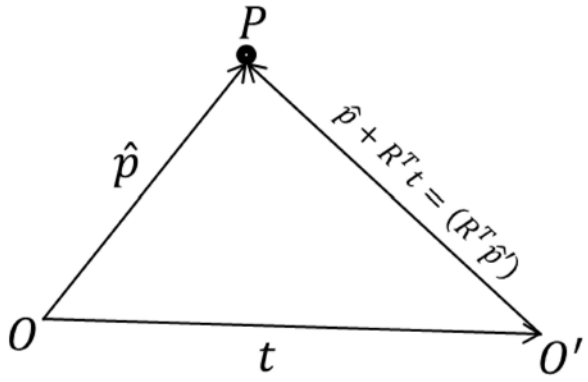

Figure 3.7: The coplanarity between the vectors.

The 3×3 quantity $\boldsymbol{\varepsilon} = R\,[t_\times]$ is called the *Essential Matrix* (Longuet-Higgins, 1981). It defines the epipolar line for points in the first (*resp.*, second) image by $\hat{p}'^T\boldsymbol{\varepsilon}$(*resp.*, $\boldsymbol{\varepsilon}\hat{p}$).

To recover the pose from the essential matrix. A unique solution could be obtained by employing the Chirality constraint (Hartley, 1998). Simply put, there are four solutions for R and t corresponding to $\boldsymbol{\varepsilon}$ and $-\boldsymbol{\varepsilon}$, two solutions each. The Chirality constrains the 3D point to be physically in front of the two cameras, only one of these four configuration verifies that, Figure (3.8). One interesting property of the essential matrix is:
A non zero matrix $\boldsymbol{\varepsilon}$ is an essential matrix if and only if its SVD (Singular Value Decomposition) : $\boldsymbol{\varepsilon} = U\Sigma V^T$satisfies $\Sigma = diag\,[\sigma_1, \sigma_2, \sigma_3]$ with $\sigma_1 = \sigma_2 \neq 0$, and $U, V \,\epsilon\, SO\,(3)$ (Hartley and Zisserman, 2004) result 9.17 (p257).

In the case of an uncalibrated setup, the epipolar constraint is interpreted by another matrix. The epipolar constraint could be written in terms of the unknown normalized coordinates by substituting in $\hat{p}'^T\boldsymbol{\varepsilon}\hat{p} = 0$ as in $p'^T\underbrace{K^{-T}\boldsymbol{\varepsilon}K'^{-1}}_{F}p = 0$. The 3×3 quantity is called the *Fundamental Matrix.*

$$F = K^{-T}\boldsymbol{\varepsilon}K'^{-1} \tag{3.3}$$

Two properties (relevant to our application) for the fundamental matrix are:

- A non zero matrix F is a fundamental matrix if and only if its SVD $F = U\Sigma V^T$, $\Sigma = [\sigma_1, \sigma_2, \sigma_3]$ with $\sigma_1, \sigma_2 \neq 0$ and $\sigma_3 = 0$ and $U, V \,\epsilon\, SO\,(3)$.

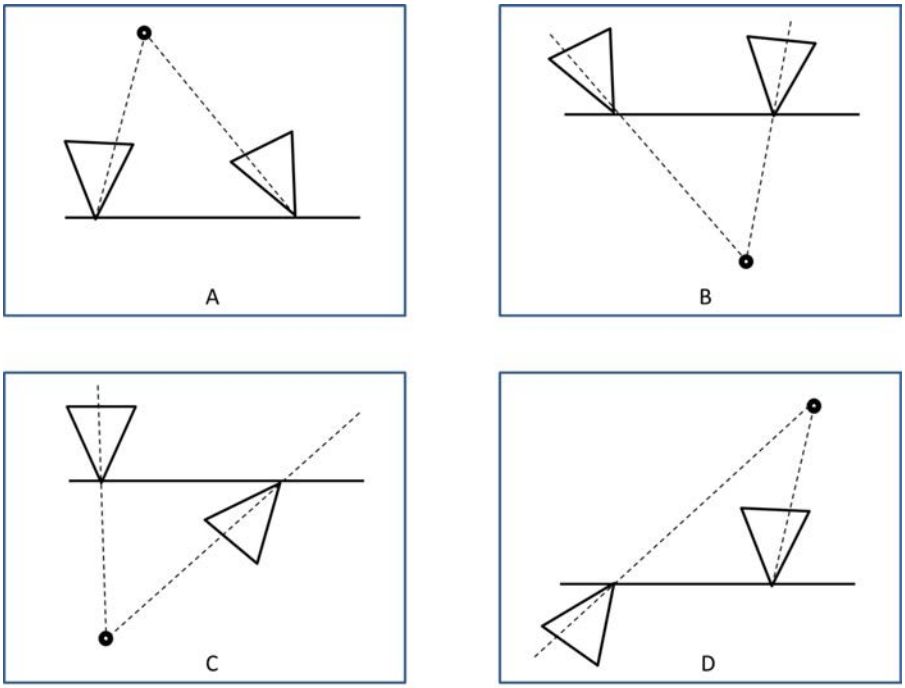

Figure 3.8: Chirality Constraint: four possible solutions, only solution (A) is meaningful.

- The epipolar line $l' = Fp$ contains the epipole e' for every p since all epipolar lines passe by their respective epipole. This means $e'^T(Fp) = (e'^T F)p = 0$. The quantity $e'^T F$ is always zero which means that e' is the left null space of F. Similarly, the quantity Fe is always zero which means e is the right null space of F.

To estimate the fundamental matrix, a set of corresponding image points is needed to build a linear system of equations around the entries of F and then solve in linear least squares fashion.

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = 0 \Longrightarrow \min_F \sum_{i=1}^{N} \left(p_i^T F p_i'\right)^2$$

This is a standard procedure in computer vision and many robust techniques exist (Hartley, 1997a, Huang et al., 2007).

3.3 Auto-calibration

In many cases the physical camera device might not be available. Thus, obtaining images for a particular 3D pattern like how it is done in conventional camera calibration is not possible. This gave rise to the Auto-calibration (or Self-calibration) methods, in which the calibration is achieved only using information encoded within images. This is the case of interest here since the aim is to exploit amateur videos collected from the web.
The literature contains many specialized methods that impose certain conditions on the images or the camera movement for the auto-calibration to work. A classical technique consists of exploiting the essential matrix property mentioned earlier, that the two non-zero singular values of its SVD should be equal. A cost function is thus built that minimizes the difference between the two singular values by using the available fundamental matrices between different views while optimizing over internal calibration matrices using $\varepsilon = K^T F K'$ (Mendonça and Cipolla, 1999). These methods are shown

to generally give only fair intrinsic parameters values at best. It has been reported that sometimes the 3D reconstruction might completely be wrong without an explanation, Fusiello (1999). Authors also showed that the error increases with noise. As such, these methods are not generalized enough for our application . Another family of approaches is homography-based auto-calibration. Planar objects are used for this type of auto-calibration. The planarity reduces the projection to a 3×3 homography matrix (instead of 3×4 projection matrix) between the planar objects and their image projections. This allows to add certain constraints to find K. One promising result is shown in (Herrera et al., 2016) using planes. Their algorithm accuracy is closely comparable to pattern-based calibration. However it requires a planar object that is visible in all images, one that is not uniform in color so that feature points detection is possible. It is difficult to comply to this requirement in the scenes we are dealing with. A big part of the image is not usable for features detection algorithms as it is moving (the river itself). It is hard to find such planar object in the remaining part of the image. However if this is the case. The algorithm showed great results and stability and it is recommended due to its accuracy. We describe next a general framework suggested by (Hartley, 1997b, Sturm et al., 2005) that is based on *Kruppa Equations*. It can be used to auto-calibrate the camera from image sequences of rivers while imposing few reasonable assumptions on these image sequences. Note that we address this problem independently from the particularities of rivers image sequences. It is a general problem in computer vision and any method could be used.

3.3.1 Kruppa Equations

In projective geometry, points at infinity are treated as equally to points located at finite distances. Using homogeneous coordinates, points at infinity have their fourth element equal to zero. Geometric elements at infinity make a great deal to auto-calibration algorithms since Faugeras, Maybank, Luong and Triggs early work (Luong and Faugeras, 1997, Maybank and Faugeras, 1992, Bill Triggs, 1997). New concepts like the Absolute Conic (AC) and

the Absolute Quadric (AQ) have been thus introduced. They are interesting because they exist (and consequently, their respective images also exist) for every image sequence. They serve as virtual calibration patterns. It has been shown that finding the image of the absolute conic (IAC) is equivalent to camera calibration (specifically finding K). The AC is a point conic that resides at infinity. It is defined by a point at infinity P with $[X, Y, Z, 0]$ coordinates where $X^2 + Y^2 + Z^2 = 0$, that is $P^T P = 0$. A point is in a conic if $P^T C P = 0$ where C is a matrix that defines conic parameters. IAC is therefore the perspective projection of points in AC to image plane. Since the last element of P is zero, all translation components (t_i) will eventually evaluate to zero, Equation (3.2). This indicates that this projection is invariant to t. A point on AC would then project into image plane by $p = KRP$. This could be rewritten as $P = R^T K^{-1} p$. Substituting for P in $P^T P = 0$ we have $p^T K^{-T} R R^{-1} K^{-1} p = p^T K^{-T} K^{-1} p = 0$. One can see that this projection is actually invariant to the complete pose of the camera and not only the translation t. This invariance to rotation and translation called the *rigidity constraint*. Image point p is in the image of the absolute conic only if its conic matrix C is represented by $K^{-T} K^{-1}$. The inverse (KK^T) could be factorized using Cholesky factorization such that K is an upper triangular matrix with positive diagonal entries (just like the matrix of the internal camera parameters). One can estimate the inverse KK^T directly using the notion of dual conics. The dual conic of our earlier point conic is a line conic and is defined according to $l^T C^* l = 0$ where l is a line vector and C^* is the adjoint matrix of C . Note that $C^* = C^{-1}$ only for full-rank conic matrices. If the matrix is not full rank, the conic is labeled *degenerate* and the dual conic matrix is not the inverse. Kruppa equations derivation is known to be complicated and somewhat counter-intuitive. The following derivation is due to Hartley (1997b) and it is more intuitive than the original derivation. Even if a different derivation/expression for Kruppa equation is used later on, we start with this one for the sake of simplicity. Let's consider any stereo configuration (i.e. two views related by epipolar geometry), we can change this configuration by multiplying K with a matrix A for the first image AK and $A'K'$ for second image, respectively. Suppose the two transformations

A and A' are chosen in such way that the fundamental matrix between the new projection matrices is:

$$\tilde{F} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

This is a special fundamental matrix that sets the two epipoles to the one origin of the stereo configuration so that the corresponding epipolar lines are identical. Then imagine a plane passing through the camera center and is tangent to the AC. It will project to two epipolar lines which are now tangent to the image of the absolute conic IAC, Figure (3.9). Tangent lines to conics are expressed via the dual conic that verifies $l^T C^* l = 0$ where l is any epipolar line. There are two such planes so we end up with two pairs of epipolar lines. A transforms C^* into a new conic envelope in the first image and the same goes for A' and C'^* in the second image. These are now different conic envelopes even if we assume the same $K = K'$ since transformation matrices A and A' are probably different. Let's denote the new conic envelopes D and D' for the first and the second image, respectively. Let $l = (\lambda, \mu, 0)$, from $l^T D l = 0$, it could be expanded into $\lambda^2 d_{11} + 2\lambda\mu d_{12} + \mu^2 d_{22} = 0$ and similarly for $l^T D' l = 0$ we will obtain $\lambda^2 d'_{11} + 2\lambda\mu d'_{12} + \mu^2 d'_{22} = 0$. Remember that after applying the transformation A and A', these tangent epipolar lines are now identical in the two images which means the two equations are equal (up to scale), leading to:

$$\frac{d_{11}}{d'_{11}} = \frac{d_{12}}{d'_{12}} = \frac{d_{22}}{d'_{22}} \tag{3.4}$$

A point p is on a line l if $l^T p = 0$. This could be rewritten as $l^T A^{-1} A p = 0$, which means that a point p is on l only if the transformed point Ap is on the transformed line $l^T A^{-1}$. From $l^T C^* l = 0$, we could write $(l^T A^{-1})(A C^* A^T)(A^{-T} l) = 0$. Let

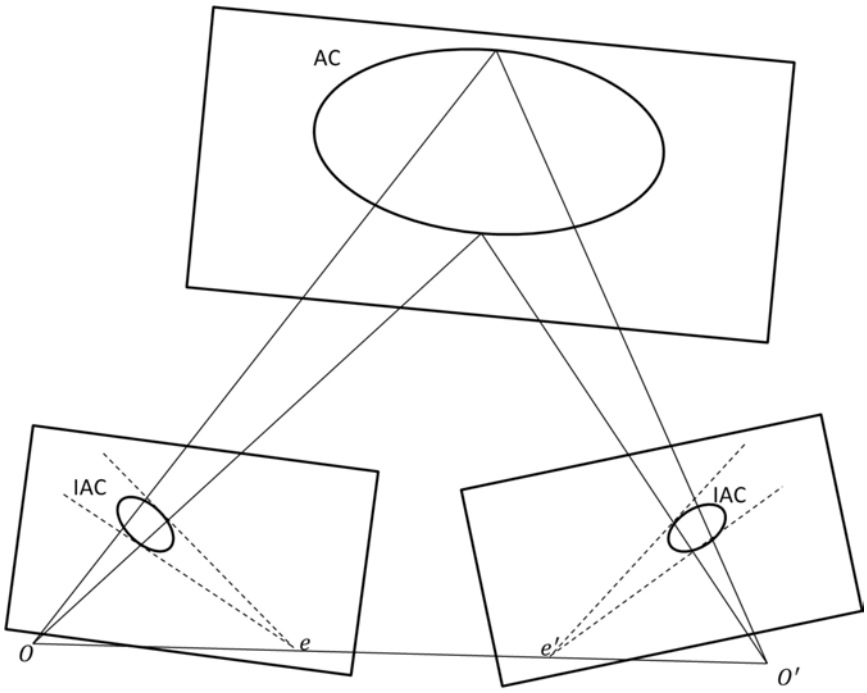

Figure 3.9: Planes tangent to the AC induces epipolar lines that are tangent to the IAC

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix}$$

where a_i is the i-th row of A. From (AC^*A^T), $d_{ij} = a_i^T C^* a_j$ where one subscript denotes a matrix row and two subscripts denote a matrix entry. A similar expression for d'_{ij} is true. Substituting in Equation (3.4) yields:

$$\frac{a_1^T C^* a_1}{a_1'^T C'^* a_1'} = \frac{a_2^T C^* a_2}{a_2'^T C'^* a_2'} = \frac{a_3^T C^* a_3}{a_3'^T C'^* a_3'}$$

These are the so-called Kruppa's equations.

3.3.2 Solving Kruppa equations

Under the assumptions of known aspect ratio (normally 1 for modern cameras) and known principal point (a good approximation would be im-

age center). Sturm et al. (2005) derived a method to estimate the focal length. These assumptions allow the process to move to an intermediate phase between the fundamental matrix (uncalibrated) and the essential matrix (calibrated). From equation (3.3), this new epipolar matrix could be computed by:

$$G \simeq \left(K^T F K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ O_x & O_y & 1 \end{bmatrix} F \begin{bmatrix} 1 & 0 & O_x \\ 0 & 1 & O_y \\ 0 & 0 & 1 \end{bmatrix} \right)$$

where G is a *semi-calibrated fundamental matrix*. The benefit gained here is that with simple change of coordinates to account for the assumed principal point and while assuming fixed and identical focal length and no image skew, the internal calibration matrix K is reduced to $diag(f, f, 1)$ which means:

$$C^* = KK^T = \begin{bmatrix} f^2 & & \\ & f^2 & \\ & & 1 \end{bmatrix}$$

Kruppa equations could also be reinterpreted in terms of the fundamental matrix and the epipole. Appendix (A).

$$GC^*G^T \sim [e']_\times C^* [e']_\times \tag{3.5}$$

The SVD of G might be written as $G = U\Sigma V^T$ where U and V are orthogonal matrices and $\Sigma = [\sigma_1, \sigma_2, \sigma_3]$ is the diagonal matrix of singular values. One might recall from Section (3.2.3) that the second epipole is the left null space of the fundamental matrix. i.e. $e'^T G = 0$. By substituting in Equation (3.5) we get:

$$U\Sigma V^T C^* V \Sigma U^T \sim [u_3]_\times C^* [u_3]_\times$$

since u_3 is the vector of U corresponding to the zero singular value in Σ. Multiplying both sides by U^T from the left and U from the right gives:

$$\Sigma V^T C^* U^T \Sigma V \sim \begin{bmatrix} u_1^T \\ u_2^T \\ u_3^T \end{bmatrix} [u_3]_\times C^* [u_3]_\times \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}$$

an evaluation leads to:

$$\begin{bmatrix} \sigma_1 v_1^T \\ \sigma_2 v_2^T \\ 0 \end{bmatrix} \begin{bmatrix} f^2 & & \\ & f^2 & \\ & & 1 \end{bmatrix} \begin{bmatrix} \sigma_1 v_1 & \sigma_2 v_2 & 0 \end{bmatrix} \sim \begin{bmatrix} u_2^T \\ -u_1^T \\ 0 \end{bmatrix} \begin{bmatrix} f^2 & & \\ & f^2 & \\ & & 1 \end{bmatrix} \begin{bmatrix} u_2 & -u_1 & 0 \end{bmatrix}$$

Even more simplification while taking advantage of unit norm U and V columns and ignoring the third all zero rows:

$$\begin{bmatrix} \sigma_1^2 \left(f^2 + V_{31}^2 \left(1 - f^2\right)\right) & \sigma_1 \sigma_2 V_{31} V_{32} \left(1 - f^2\right) \\ \sigma_1 \sigma_2 V_{31} V_{32} \left(1 - f^2\right) & \sigma_2^2 \left(f^2 + V_{32}^2 \left(1 - f^2\right)\right) \end{bmatrix} \sim \begin{bmatrix} f^2 + U_{32}^2 \left(1 - f^2\right) & -U_{31} U_{32} \left(1 - f^2\right) \\ -U_{31} U_{32} \left(1 - f^2\right) & f^2 + U_{31}^2 \left(1 - f^2\right) \end{bmatrix}$$

This system of symmetric matrices gives rise to 3 quadratic equations in f^2. Two of them however will have the trivial solution $f^2 = 1$. By factoring this out we end up with two linear equations and a quadratic one:

$$f^2 \left(\sigma_1 U_{31} U_{32} \left(1 - V_{31}^2\right) + \sigma_2 V_{31} V_{32} \left(1 - U_{32}^2\right)\right) + U_{32} V_{31} \left(\sigma_1 U_{31} V_{31} + \sigma_2 U_{32} V_{32}\right) = 0 \tag{3.6}$$

$$f^2 \left(\sigma_1 V_{31} V_{32} \left(1 - U_{31}^2\right) + \sigma_2 U_{31} U_{32} \left(1 - V_{32}^2\right)\right) + U_{31} V_{32} \left(\sigma_1 U_{31} V_{32} + \sigma_2 U_{32} V_{32}\right) = 0 \tag{3.7}$$

$$\begin{aligned} f^4 \left(\sigma_1^2 \left(1 - U_{31}^2\right) \left(1 - V_{31}^2\right) - \sigma_2^2 \left(1 - U_{32}^2\right) \left(1 - V_{32}^2\right)\right) + \\ f^2 \left(\sigma_1^2 \left(U_{31}^2 + V_{31}^2 - 2 U_{31}^2 V_{31}^2\right) - \sigma_2^2 \left(U_{32}^2 + V_{32}^2 - 2 U_{32}^2 V_{32}^2\right)\right) + \\ \left(\sigma_1^2 U_{31}^2 V_{31}^2 - \sigma_2^2 U_{32}^2 V_{32}^2\right) = 0 \end{aligned} \tag{3.8}$$

Each type (whether quadratic or linear) has its own degeneracies, that is, a configuration where auto-calibration is impossible. Using all equations, a

plethora of solutions is obtained. In practice we take the peak of a PDF of all solutions as the focal length.

3.3.3 Bundle adjustment

At this point, estimates for camera intrinsic parameters are found. Sturm method is generally stable so a reasonably good value for the focal length is obtained. The principal point is approximated at the image center. To refine the estimation, a standard technique called *bundle adjustment* is used. It is an optimization procedure that simultaneously refines 3D points, camera poses and camera intrinsic parameters, all to minimize a *re-projection error* in image space.

$$\underset{K[R|t]_j, P_i}{min} \sum_{i=1}^{n}\sum_{j=1}^{m} \nu_{ij}\eta\left(\rho\left(K\left[R|t\right]_j, P_i\right), p_{ij}\right)^2$$

where function ρ in the above is the predicted re-projection of a point P_i in the image of a camera characterized by $K\left[R|t\right]_j$, function η measures the distance between this re-projection and the actual point p_{ij} in image space and ν_{ij} is a binary variable represents the visibility of a point P_i by camera j . Bundle adjustment is usually cumbersome and difficult problem to solve due to the large number of free parameters involved, Figure (3.11). Agarwal et al. (2010) developed a non-linear least square framework (Google Ceres) with extensive support for bundle adjustment problems. Significant re-projection error reductions obtained using this framework.

Bundle adjustment is the last step in the so-called *Structure From Motion*. As the name suggests, it is a framework to estimate a 3D structure of a scene from the motion of a camera. It comes to the picture because with Sturm method, we only estimate a focal length and internal camera calibration is completed by adding the principal point. We still need the 3D points and the camera poses. To this end, the first view of the sequence is taken as the reference view which we assign an identity rotation matrix and zero translation vector. Between every two consecutive images we could compute an essential matrix with which we can obtain a relative pose between the two.

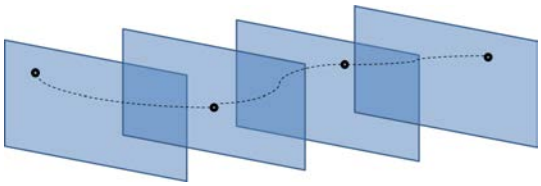

Figure 3.10: Feature point tracked throughout an image sequence

We could then transform it back to the coordinate system of the reference view. To estimate the 3D points, we firstly track the 2D features obtained throughout the whole sequence so that we know the position of each point in every image (this is how the visibility variable ν_{ij} is assigned), Figure (3.10). The triangulation is then performed in multiple view context (contrary to the triangulation performed only between two images in stereo context).

Bundle adjustment is then performed to refine all parameters except for the pose of the first view. It is also found to be beneficial to add lower and upper constraints for the principal point. These steps could be summarized as follows:

- Obtain an initial focal length estimation using Sturm method above.
- Give an initial estimation for the principal point by taking image center.
- Fix the pose of the first camera as the reference coordinate system.
- Add lower and upper constraints to the principal point (usually +- 100).

Results

Bundle adjustment is applied to the auto-calibration of two image sequences. The first sequence Irstea is taken at Irstea Institute building in a configuration that is similar to urban floods sequences. The similarity is in the two different regions in the image. The first is a road that is not perfectly plane (made like this to collect rain water) in which no feature points are detected. The other region is a building on the facing side. The second

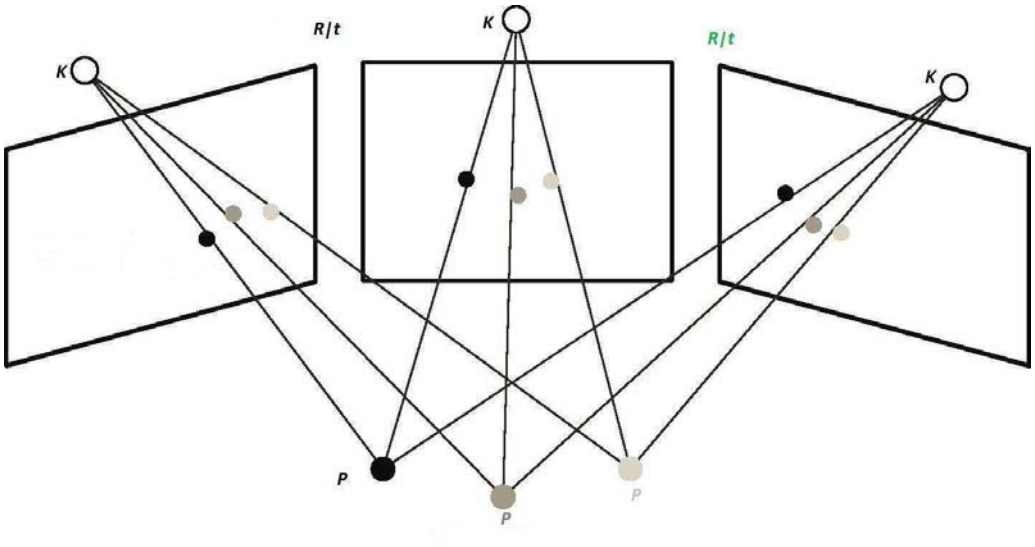

Figure 3.11: Bundle adjustment.

Sequence	Error before BA	Error after BA	Refined f	Refined pp	Image size
Irstea	254.01	12.29	1525.23	(850,656.549)	1920*1080
Aulnay	114.005	3.4694	846.598	(334,245.518)	768*432

Table 3.1: Back-projection error before and after BA in addition to refined intrinsic parameters.

sequence is a real amateur sequence of an urban flood in the Aulnay-Sur-Mauldre city in France. Figure (3.12) shows example images for the two sequences. We measure the re-projection error for the two sequences before and after bundle adjustment. This is shown in Table (3.1).

Figure 3.12: Example images for the sequences used for auto-calibration.

3.4 Image stabilization

The river motion captured by the camera is perturbed by the motion of the camera itself. It is hence necessary to separate these two motions and cancel the latter out. This is called camera stabilization. To this end, the fixed regions in the images are of primordial importance because any motion observed on them is only due to the camera. We take consecutive frames to stabilize. Consecutive frames are easier to establish points correspondence on them but also easier to estimate the velocity by the mean of optical flow as well. The procedure is as follows:

- Take the first two images.
- Establish the correspondence using high accuracy feature points.
- Compute an affine transformation.
- Warp the second image using the estimated transform.

- Take the newly warped image with the third image and repeat the previous steps.

We end up with a stabilized sequence. The stabilization means that the motion of rigid points (or equivalently, the camera motion) has been canceled. The quality of this stabilization could be visualized by observing optical flow (conventional optical flow (HS)) color code result on the stabilized and non-stabilized images, Figure (3.13).

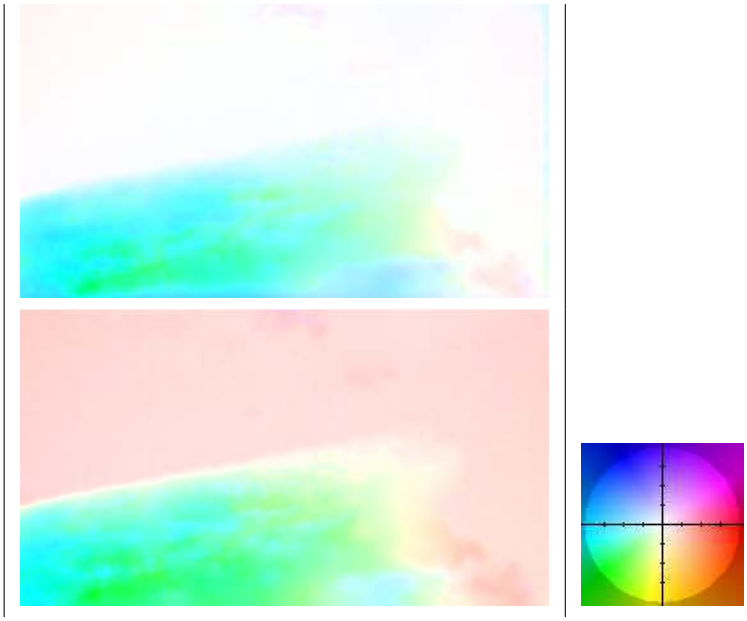

Figure 3.13: Top: Color code of optical flow on a stabilized sequence. Bottom: color code of optical flow on the original non-stabilized sequence.

3.5 3D reconstruction

As it has been indicated earlier, we want to refrain from image ortho-rectification since it interpolates the intensity function of images. This process is not guaranteed to conserve fluids motion patterns. The image intensity function is the only source of information available for velocity computation and we thus want to rely on the original non-degraded data as much as possible. A second and probably more important reason is that ortho-rectification needs installed GRPs *in situ*, necessary for the ortho-rectification to take the

Figure 3.14: GRPs Deployment.

scale factor into account. They are also used to estimate the full projection matrix Q of the camera using the direct linear calibration method. This is particularly a task that could be much enhanced. Indeed, camera calibration is a mature computer vision procedure and the intrinsic parameters could be estimated offline to a high degree of accuracy using planar objects. The extrinsics could then be measured directly *in situ*. In addition, GRPs deployment requires many operators to install and deploy the physical points, Figure (3.14).

Ideally, we would like to be able to back-project displacement vectors computed on the 2D image and compute the final velocity in 3D space. To this end, a 3D reconstruction would solve two problems at once. On the one hand, a 3D reconstructed river plane means that the relationship with its 2D image plane is already known. On the other hand, a dense 3D reconstruction for fixed regions permits in theory to identify objects that could be useful to compute the 3D *similarity factor* (windows, road signs, dam parts, etc.). The similarity factor is analogous to the scale factor in ortho-rectification. Two 3D scenes are equal up to a similarity transformation via this similarity

factor.

We first attempted a Structure From Motion approach since it is already used in the bundle adjustment phase of auto-calibration. But to be able to recognize objects in 3D space, a dense or quasi dense reconstruction is needed. Previously, 3D reconstruction step was introduced only for the purpose of Auto-calibration bundle adjustment step. This rather uses highly accurate but sparse points correspondence. Consequently, only a sparse reconstruction is obtained. One could use another feature detector that gives more (but less accurate) features in order to estimate denser structure and run the bundle adjustment again using previously estimated values as initialization. This second bundle adjustment is meant to correct the 3D structure rather than camera parameters. The result is a 3D point cloud of the fixed regions that we try to recognize objects in them. Unfortunately, amateur sequences tend to have very small baseline. Such condition makes the reconstruction very difficult since the back-projected rays of many points will be quasi-parallel in which case the initial triangulation of many 3D points is doomed to fail. In addition, the more denser set of features, the more it contains outliers that negatively impact the bundle adjustment which is based on least squares estimation framework. It is extremely difficult to reconstruct dense 3D regions to get recognizable 3D objects within them in order to extract a reliable similarity factor. For river plane and 2D to 2D mapping estimation problem, one needs at least 4 non-collinear 3D points to estimate a homography (the 2D mapping matrix). The main assumption of Structure From Motion is a fixed structure with a moving camera. As such, the moving river surface can not be reconstructed especially that it is not exactly a rigid motion. Possible workaround is to detect points on the river-bank joints as they also belong to river surface plane. This is also a challenging task and is not guaranteed to be reliable using automatically detected feature points. Manually selecting corresponding points on banks/river intersections might be the way to proceed. Good homography estimation however needs well-distributed points on the plane and most of these river sequences actually only capture one bank (the amateur is usually standing on the other side). Even if we were able to detect these points, they will only be on one bank

Figure 3.15: Rectified image pair using epipolar geometry, the camera motion causes points to only move on their corresponding epipolar lines which are parallel to x axis in the rectified images.

and probably collinear .

There are some other variational algorithms that produce dense disparity maps, (Alvarez et al., 2002, Slesareva et al., 2007). A disparity map is the image of the difference between corresponding points in two images. It is known that the disparity is proportional to the depth if the two images are rectified (using epipolar geometry). This rectification has the effect of constraining the displacement of points induced by camera motion to only be along the x axis, Figure (3.15). From similar triangles introduced by this rectified setup the depth could be easily computed from:

$$Z = \frac{ft_x}{disparity}$$

Ha et al. (2016) also proposed a method specially tailored for narrow baseline images that produces dense 3D maps. We tested many of these methods and we obtained various results. In all cases the river surface has been always badly estimated. This was to be expected because of the motion of river pixels. But also because they exhibit the same range of intensity values, which makes the correspondence even more ambiguous.

From all the above, we conclude that traditional 3D reconstruction methods for this type of sequences is extremely difficult. A new way of thinking is required to solve the problem while trying to generalize as much as possible. By evaluating the information at hand and the desired output, a new idea has emerged. Let's start by describing the available information we have in

an amateur video of a river. First of all we have the river surface which we considered plane. This plane is however not suitable for the 3D reconstruction as it moves and features detection on it is not reliable. A second piece of information is that this plane doesn't occupy the whole image and some other fixed regions are visible. They have a neighbourhood relationship with river plane in someway or another. The desired output is a 3D reconstruction of river plane with a direct relationship with its 2D image counterpart. In addition to that, a 3D similarity factor is also mandatory to pass from our 3D model measurement to the real world metric measurements. This led us to investigate robotics research literature as the problem is similar to indoor robots navigation. Indeed, the robot is surrounded by planes (floor and walls) which it needs to recognize in 3D in order to be able to navigate in its environment. These planes are similar to river surface plane since it is not easy to detect feature points on them (many indoor walls are uniform in color). Recent work in the context of SLAM (Simultaneous Localization And Mapping) showed good results in indoor environments (Yang et al., 2016b,a). Authors proposed a framework that was capable of continuously estimating the 3D environment throughout a sequence of images obtained while the robot is moving. It only uses single image 3D reconstruction algorithm combined with *machine learning* to refine the 3D structure over time. Authors impose a 3D scene structure composed of a floor with orthogonal walls. In the following, we made the analogy with image sequences of rivers. We particularly talk about amateur videos. The ones taken by scientists for the purpose of river gauging are readily exploitable using this framework. We argue that rivers has to be delimited with banks anyway. It is often possible to find some object perpendicular to the river (dams, walls, etc.). The 3D reconstruction problem is then reduced to a reconstruction of the river surface with a perpendicular object on its boundary. This object is used to compute the similarity factor of the reconstructed model to the real world. Of course, we still need to know the size of the object in real world.

The reconstruction algorithm is derived as follows: A plane could be represented in homogeneous coordinates using a vector $\pi = [\pi_1, \pi_2, \pi_3, \pi_4]^T$ where the first 3 parameters represent the normal of the plane and the 4th

is the distance to the origin. The camera pose is represented by a transformation matrix $[R|t]$ and verifies $P_W = [R|t]\ P_C$, that is, the transformation of points between world and camera systems of coordinates. To deal with planes directly, remember that a point lie on a plane if $\pi^T P = 0$. Multiplying points in by $[R|t]$ moves them to a new plane in the other coordinate system: $\pi_C^T\,[R|t]\ P = \pi_W^T P = 0$. This means $\pi_C^T\,[R|t] = \pi_W^T$. Multiplying both sides by $[R|t]^{-1}$ from the right and transpose gives:

$$\pi_W = [R|t]^{-T}\,\pi_C \tag{3.9}$$

The river plane could now be reconstructed when expressed in camera coordinates system as it is the plane where the ensemble of the back-projected river pixel points in the image intersect at:

$$P_C = \frac{-d_c}{n_C^T\,(K^{-1}p)} K^{-1}p \tag{3.10}$$

where n is the plane normal and d is the distance to the origin. The image is segmented into floor/walls regions, the machine learning in the original SLAM problem is utilized to adjust the segmentation throughout the sequence. In our case however, we build the 3D model based only on one image with manual segmentation. The segmentation defines the segment of the river plane shared with the vertical object(s).

The two starting image points of a river/bank segment could be reconstructed in 3D (P_{C1}, P_{C2}) using previous equations, but these points are also on the perpendicular plane, the normal of which verifies:

$$n_{wall-C} = n_C \times (P_{C2} - P_{C1})$$

The distance to the origin $d_\perp$could be obtained using:

$$d_{wall-C} = n_{wall-C} \cdot P_{C1}$$

The river plane in the world coordinates is set to $\pi_W = [0, 0, -1, 0]^T$. Using this with previous equations we can reconstruct a 3D river/walls model. We still however need to determine the transformation matrix $R|t$. The amateur

is naturally standing above river plane surface and the camera is looking down at an angle. This boils down to one parameter estimation of this angle. The translation vector in homogeneous coordinates is obtained as $t = [0, 0, h, 1]$ where h is the height of the camera from the river plane. Note the the height of the camera only affects the scale of the reconstructed scene which is to be calculated later anyway. A one-axis rotation matrix is defined as follows:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\Theta & sin\Theta \\ 0 & -sin\Theta & cos\Theta \\ 0 & 0 & 0 \end{bmatrix}$$

Note that the world system of coordinates is defined as: X right, Y forward and Z upward. The camera system is defines as: X right Y downward, Z forward. We vary Θ in radian until a good reconstruction is obtained.

There is an Euclidean similarity relationship between the reconstructed scene and the real world. Angles and ratios are conserved. It follows that there is one similarity coefficient that relates the two 3D worlds. Of course to go from the reconstructed scene to the *exact* real world scene, it should be scaled using the similarity coefficient in addition to one or more of rotations, translation and reflection. But for the purpose of this application, only the similarity coefficient is needed. It is easily computed from the ratio of the size/length of the perpendicular object in the 3D model and in the real world.

3.5.1 Results

We show the reconstruction results on both amateur videos and velocimetry videos. As mentioned before, the difference between the two is that the amateur videos contain camera motions and the camera itself is not calibrated. However, for the purpose of 3D reconstruction described above, the camera motion is not an issue since the 3D model is obtained from a single image. The camera motion is only necessary for the Auto-calibration.

Figure 3.16: Claix canal.

Velocimetry videos

We took Claix canal LSPIV example shown in the results section of Chapter (2) where an ortho-rectified images were used. The calibration is obtained through the QR decomposition of the projection matrix computed for LSPIV ortho-rectification process. Figure (3.16) shows the 2D image of the site. Figure (3.17) shows the 3D reconstruction of the site from different angles. One can see that every image point of the river is represented in 3D. The chosen perpendicular object is wall/bank of the canal and Θ gave best results when equals -0.56.

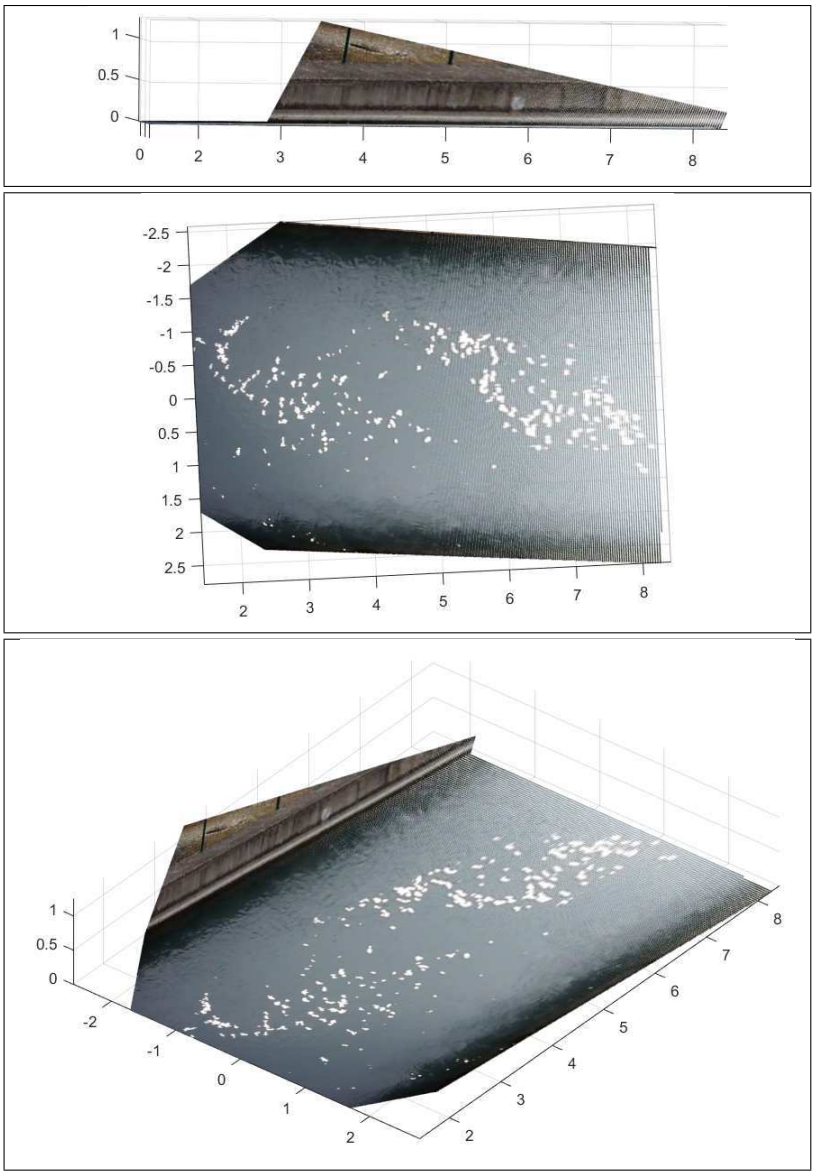

Figure 3.17: Claix canal site from different views, top: side view from water level height, middle: view from above, bottom: view from the front higher than water level.

Amateur videos

We took Aulnay sequence shown before in the Auto-calibration section of this chapter. Conveniently, camera calibration is obtained through the framework described earlier, with bundle adjustment. Figure (3.18) shows the 2D image of the site. Figure (3.19) shows the 3D reconstruction from different angles. The perpendicular object chosen is the small gate of the house and Θ taken to be -0.3.

Figure 3.18: Aulnay-Sur-Mauldre.

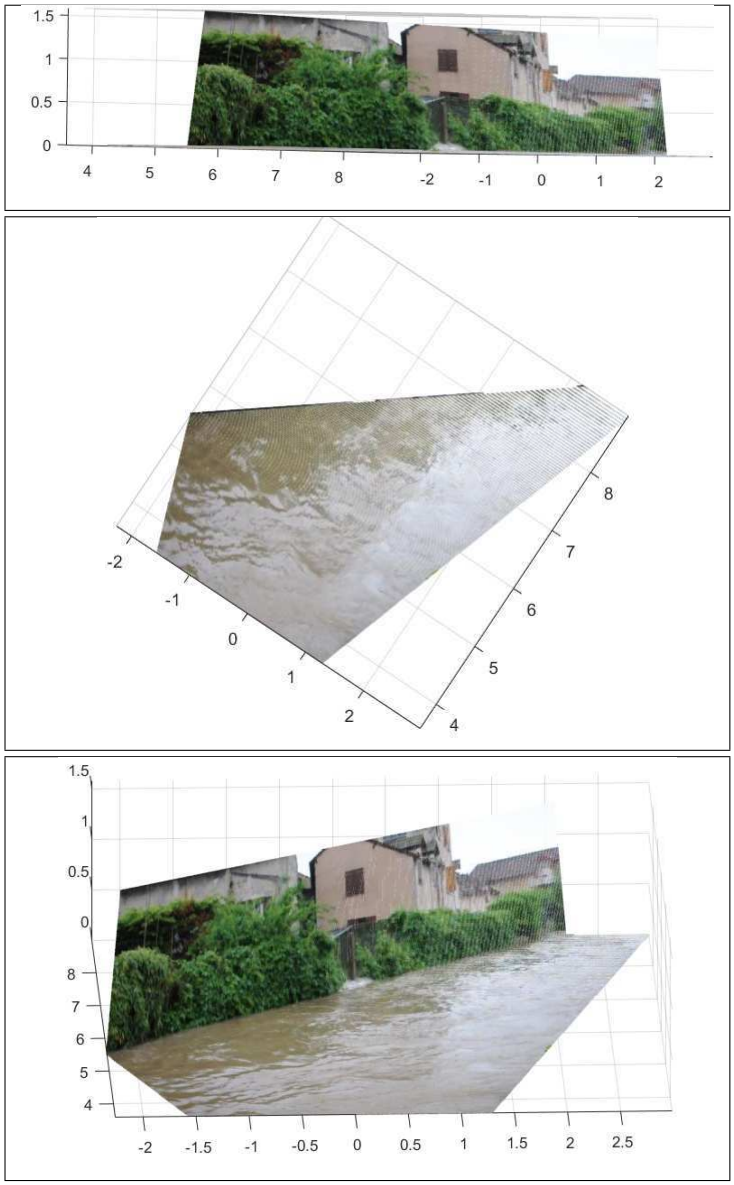

Figure 3.19: Aulnay river site from different views, top: side view from water level height, middle: view from above, bottom: view from the front higher than water level.

3.6 Conclusion

In this chapter we showed that 3D information and focal length are mandatory to go from velocity measurements in image space to useful 3D metric velocity. We have also introduced basic computer vision concepts to readers from outside the domain. These concepts have been used next to treat amateur videos (auto-calibration, stabilization) in order to make them exploitable.

Inspired by indoor robotics research, a 3D reconstruction with high fidelity for river surface points is performed using camera calibration and one image. The reconstruction is facilitated by imposing a river/bank perpendicular planes constraint. This allows a direct passage from 2D to 3D measurements. A 3D similarity coefficient is finally obtained from the vertical planes to scale the estimated 3D measurements. In comparison to traditional methods, our framework allows the integration of amateur videos in addition to videos that were purposefully taken for image-based velocimetry . For the latter, it considerably minimizes the field work needed to deploy GRPs. Indeed, we can now deploy only one point perpendicular to the river surface to be reconstructed later in 3D. Most importantly, we avoid image intensity function deformations as there is no ortho-rectification needed and we work directly on the original non-degraded images.

Chapter 4

Velocity estimation

The problem of image-based metric velocity estimation of rivers has been stratified into two parts in previous chapters. In this chapter, we combine their results to directly go from raw videos of both amateurs and scientists to metric velocity.

4.1 Framework

The starting data is a video of a river with visible fixed regions/banks. Next, we need to determine if camera calibration is possible. If the video is an amateur video, it has to contain camera motion for the application of auto-calibration. Otherwise, an accurate 3D reconstruction is not guaranteed. Figure (4.1) draws a decision tree for the 3D reconstruction of amateur videos. If the video is taken for the purpose of image-based velocimetry, the camera should be calibrated beforehand and the reconstruction is thus straightforward.

One should check for enough fixed regions in the video so that a projective reconstruction (or equivalently, estimating fundamental matrices) is possible. The fixed regions are also used to stabilize consecutive frames later on to be used for image velocity estimation. In the other case where the camera is fixed, stereo and multiple view geometry can not be used. The camera calibration should be obtained by other means. For example, if the physical

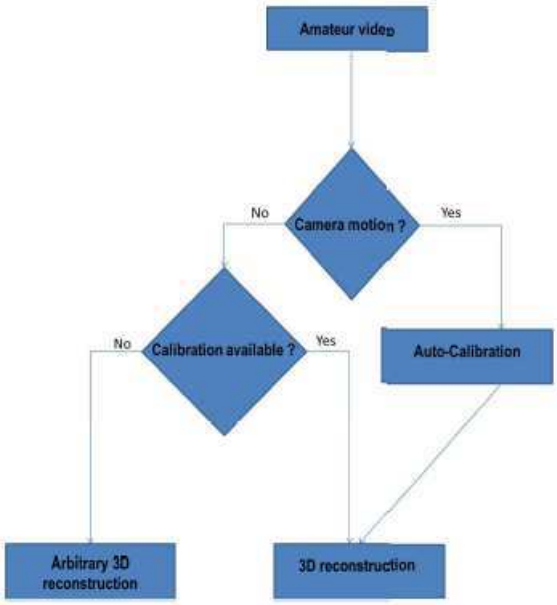

Figure 4.1: Decision tree for 3D reconstruction of amateur videos.

camera is available and under the assumption that the internal calibration is still the same to when the video was taken, then a conventional camera calibration might be performed. In the worst case scenario where non of the above is possible, a 3D reconstruction might be attempted by taking the principal point to be the image center and then empirically assign a unified value for the two focal length components.

Once the data is determined to be exploitable. Two different image sequences are extracted from it. One sequence is used for auto-calibration while the second is used for velocity estimation. Of course, in case of a fixed camera, there will not be a need for the first sequence. A video of one second length of today's cameras has at least 24 frames which facilitate the choices made on image selections. As we have seen in Chapter (3), best scenario for multi view geometry and structure from motion algorithms occurs when we have a wide baseline. However, in most cases the individual is mostly standing on the same point while the camera only moves a bit. Sometimes the video might contain different scenes so that at one instance the images might contain more fixed areas than river areas and vice versa. One should benefit

from the whole video, if we want to extract an image sequence for the purpose of auto-calibration, we should opt for images with more fixed regions, the auto-calibration might then be applied. The second type sequence to extract is the one we wish to compute the velocity on. One should take consecutive images (with the same time step) as they are a lot easier to stabilize. If the sequence contains passive particles, it is a good practice to reconstruct manual trajectories and verify the estimated vector fields. SGSD parameters could be then modified as described in Chapter (2) until satisfying 2D results are obtained. The goal here is to get as much accurate 2D results before back-projecting these vectors to 3D. An image with a clear object that is perpendicular to the river is chosen. This object might be a part of dam or a wall etc., or even a GRP. Constructing a minimal 3D river site is then straightforward as described in Chapter (3). After a good reconstruction is obtained, the perpendicular pattern is used to compute the similarity coefficient. Using the plane equation from the reconstruction, a 2D displacement vector is projected to a 3D path of the form $P(t) = (X(t), Y(t), Z(t))$. The derivative of which is the sought 3D velocity $\omega_{3D} = \left(\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt}\right) = (U, V, 0)$. This is repeated for points along an entire cross-section.

4.2 Case studies

In the following we show the results on two different sequences. For every sequence, we plot the scaled 3D displacement vectors along a cross-section and we show the final metric velocity plot of the same cross-section.

4.2.1 Claix canal

We are now familiar with Claix sequence. It has been used by LSPIV in Chapter (2) and in Chapter (3) the canal is reconstructed in 3D. This time, and without any modifications for the intensity function, we directly use the raw video to get real world velocity. Since the camera is fixed, no stabilization was needed. The similarity coefficient is obtained from the height of the banks. The real metric height is obtained from the positions of

Figure 4.2: GRPs deployed on Claix canal.

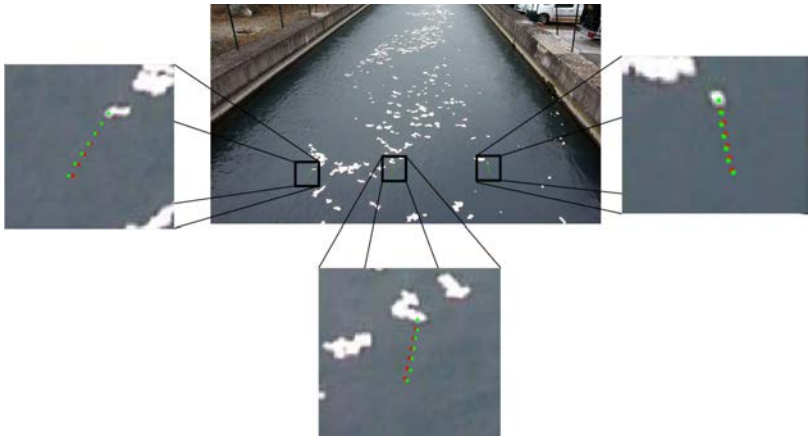

Figure 4.3: Trajectory reconstruction of particles, green: reference trajectory, red: trajectory from the estimated vector field.

GRPs used by LSPIV to ortho-rectify the sequence, Figure (4.2).

We reconstruct trajectories of few particles using the estimated vector field to make sure the 2D estimation is accurate, Figure (4.3).

Now we reconstruct the 3D river site. We choose a cross-section on the canal and we show the scaled displacement vectors, Figure (4.4). We see clearly that the magnitude of the displacement is stronger in the middle of the canal and bottoms out towards the two boundaries. This is a typical behavior in rivers. Figure (4.5) shows the velocity magnitude on the cross section. We compared this magnitude to a measurement provided by an ADCP (Acoustic Doppler Current Provider) instrument at a point in the *middle* of the canal.

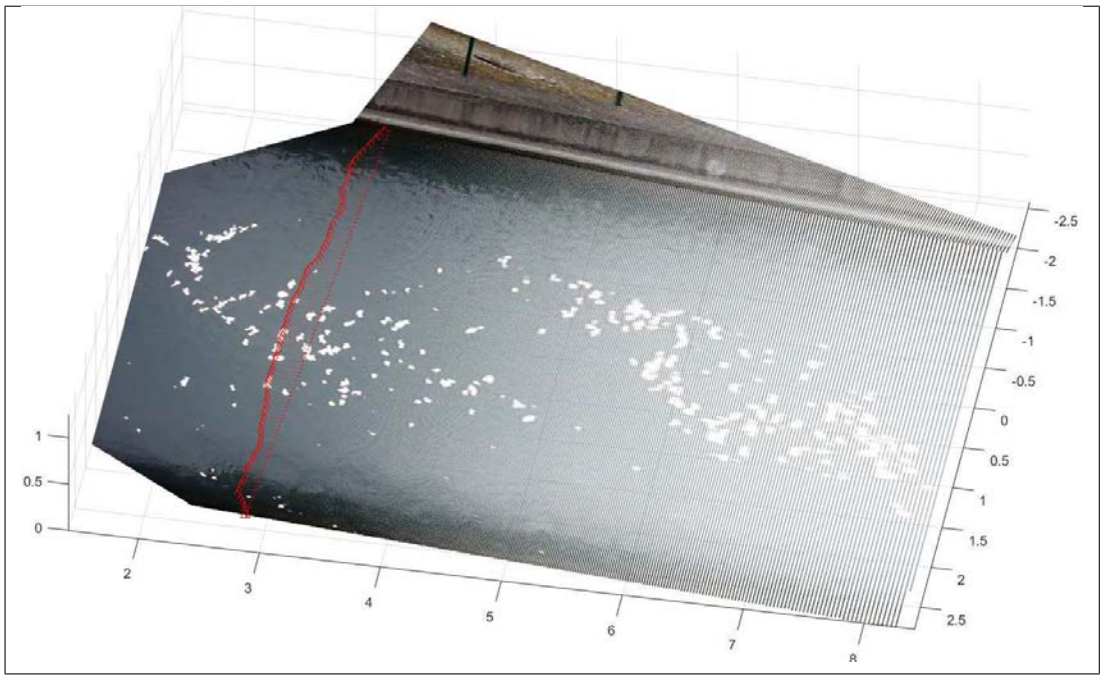

Figure 4.4: 3D displacements on the cross-section.

ADCPs provide velocity values along a vertical profile. In Figure (4.6), 10 ADCP cells measure velocity with the first cell located at 0.4m height from the bed. The last cell is located at 4m height with 0.2m distance from the free surface (represented by the red bar). The free surface velocity is subject to an extrapolation of the this height/velocity graph. We can see that ADCP estimations in agreement with our image-based estimation.

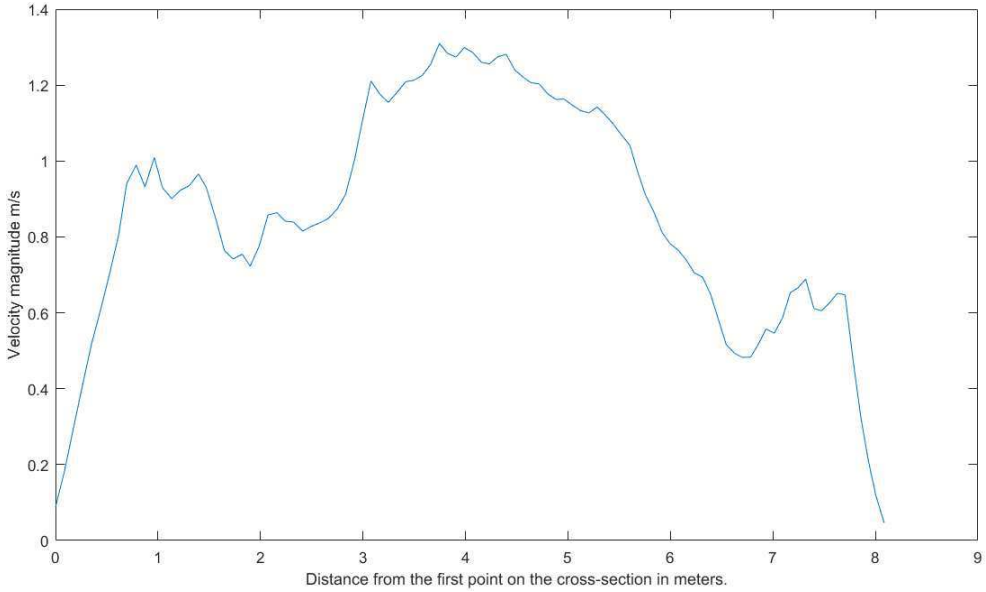

Figure 4.5: Velocity magnitude for points on the cross-section.

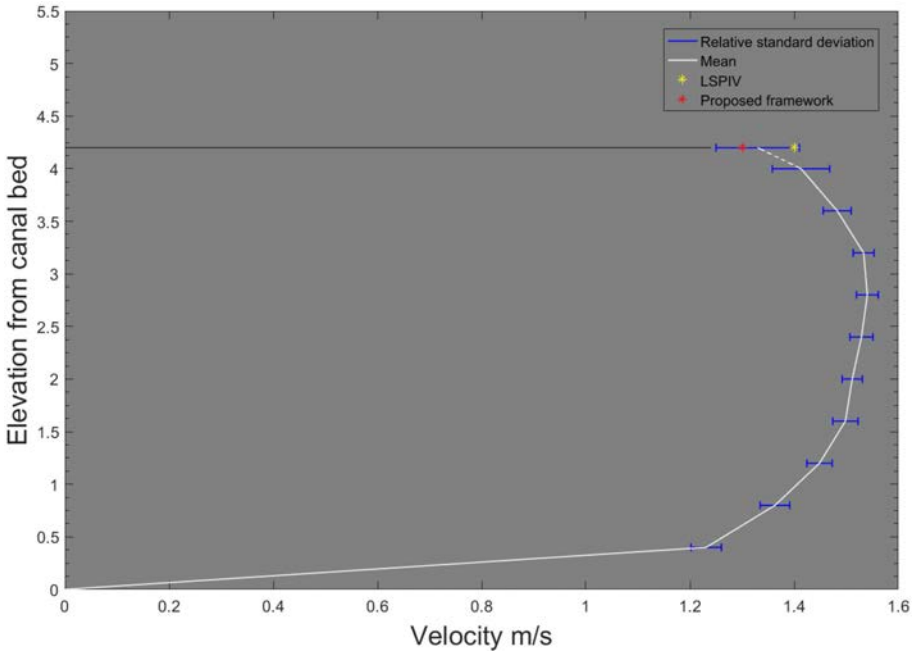

Figure 4.6: Plot of the mean velocity as measured by 10 ADCP cells at different depths, the horizontal bars represent the coefficient variation for every cell accumulated over 97 samples, the horizontal line represents the true height of the free surface.

4.2.2 Aulnay sequence

This is a video of an urban flood shot by a pedestrian in Aulnay-Sur-Mauldre in France. We have also build a 3D model for this site in Chapter (3). The video has both voluntary and involuntary camera motions and needed to be stabilized. We stabilized 6 consecutive frames to be used for motion estimation. We used the perpendicular vertical bar at the bank that supports the small gate to compute the similarity coefficient. We assumed 1.5m for its length. Figure (4.7) shows the 2D displacement vector field. Unfortunately, there was no visible tracers to track. Also, only one bank appears in the video, there is no clear cross-section spanning from one boundary to the other. Single boundary cross-section is taken and the scaled displacements in 3D are shown in Figure (4.8). The distribution of the final metric velocity is shown in Figure (4.9). Note that the assumption of 1.5m for the bar does not affect the reconstruction itself but only its size. Choosing a greater value (2.0m for example) only changes the observed displacement and of course the final velocity, Figure (4.10). Unfortunately there is no other velocity estimates to compare with. Visual comparison against the previous example suggests greater magnitude for this sequence which is what we obtained. The results seem highly plausible and physically consistent.

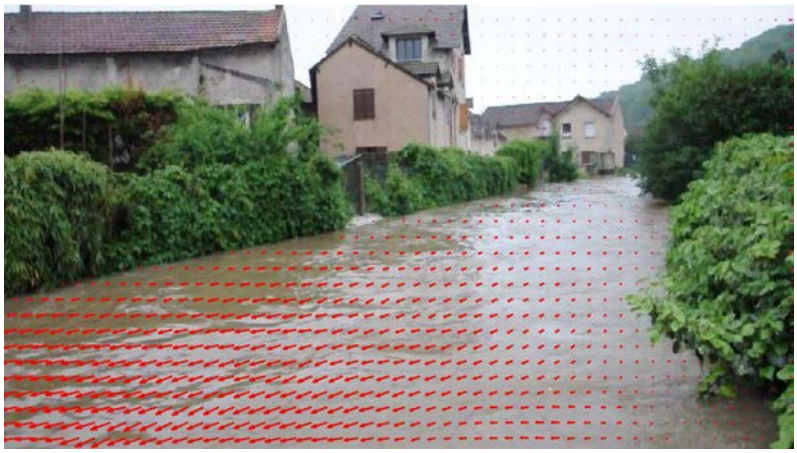

Figure 4.7: Displacement vector field of Aulnay sequence as computed in image space.

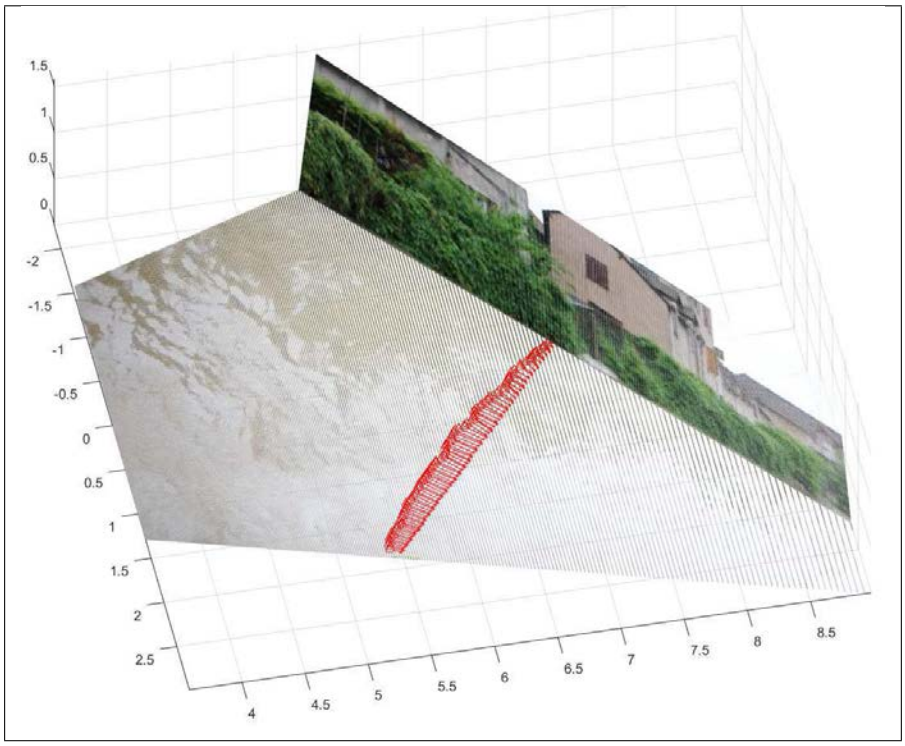

Figure 4.8: 3D displacements on the cross-section.

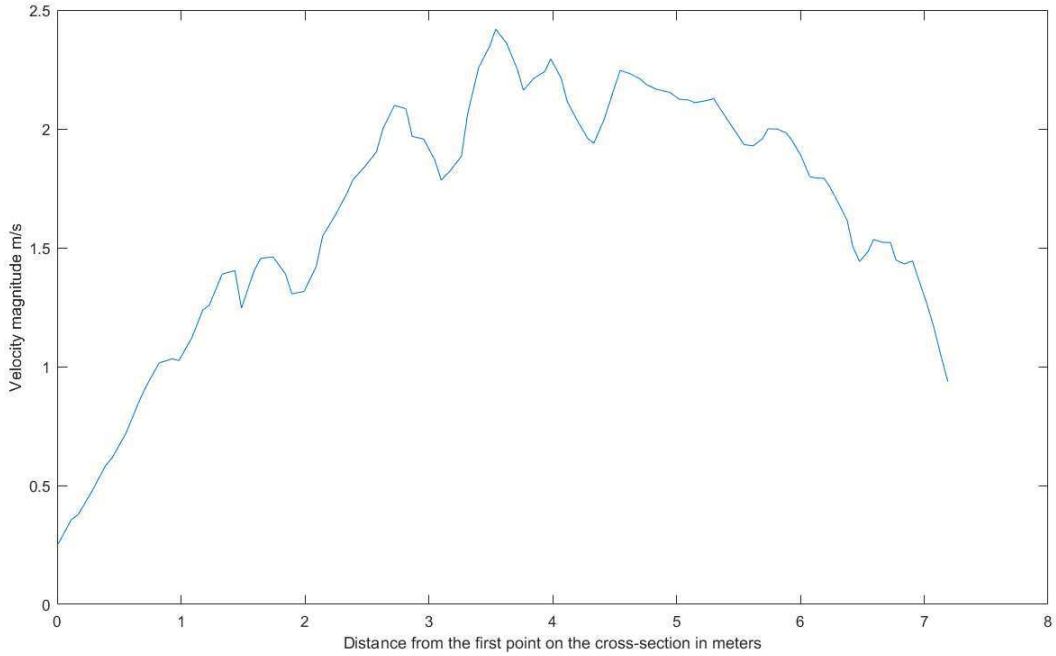

Figure 4.9: Velocity magnitude for the points on the cross-section.

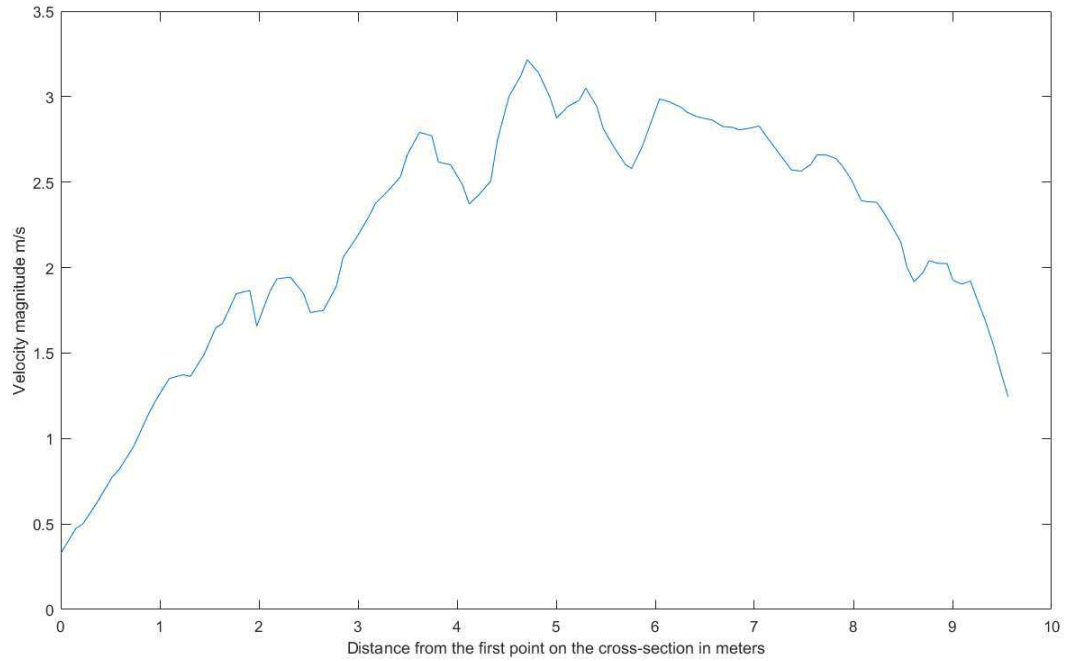

Figure 4.10: Velocity magnitude for the points on the cross-section assuming 2.0m for the vertical bar.

Chapter 5

Summary and future work

In this thesis, we made a first step towards the exploitation of videos taken by amateurs to estimate rivers or urban floods surface velocity. Under mild assumptions, we showed that indeed this data could greatly benefit rivers scientists in their gauging routines. They can even fill the holes in past events data if such videos are found to the location and the time of interest. In addition, better solutions for velocimetry videos (those taken during gauging campaigns) were demonstrated.

In chapter (2), we proposed an optical flow model that is physically consistent with river flows. This model gave by far more interesting results in terms of accuracy, falsefiability, interpretability and flexibility than other methods like the LSPIV. Indeed, the dense nature of the results provided by the model permits further derivation of fine-detail quantities like divergence fields, vorticity fields or streamlines. In addition, it provides instantaneous velocities that are out-of-the-box usable for trajectory reconstruction of particles for the purpose of results verification. This is not the case for LSPIV which rather aim for the time-averaged vector field since the instantaneous fields are generally noisy.

In Chapter (3), we laid the foundations of computer vision that are necessary for the geometrical modelling of videos taken by amateurs. This includes features correspondence, camera model and its calibration, Epipolar Geometry, Auto-calibration and Structure From Motion for Bundle Adjustment.

Parameter	Description
2D image velocity	The image velocity could be verified using trajectory reconstruction.
Camera internal calibration	This could be done offline with high accuracy.
The angle Θ	This could be exactly measured in the field.
GRP deployment	This is reduced to a single GRP deployed perpendicularly to the river surface
Similarity factor	It is exactly known since it is extracted from the single GRP point.

Table 5.1: Description of the proposed framework parameters when applied to videos taken specifically for river velocimetry.

The purpose of the geometrical modelling is to allow the passage from the 2D velocity estimated in images to a meaningful velocity in physical dimensions. We took the time to show that an appealing and somewhat intuitive direction that scientists might take is doomed to fail, namely Structure From Motion for dense reconstruction.

For 2D image vectors to go 3D, two steps are necessary. First, a back-projection transformation that inverse the perspective projection performed by the camera when capturing the motion of the 3D vectors. Second, a universal scale factor to correct the back-projected vectors to their true 3D scale. The latter could be obtained as a similarity coefficient obtained from the ratio of the length of a detected object in the 3D reconstruction with its real length. To this end, we adapted a recent 3D reconstruction method, devised in indoor robotics community after making the analogy with river image sequences. A geometrical constraint of two perpendicular planes is imposed which results in a complete characterization of the river free surface in 3D, in addition to the similarity coefficient obtained from the vertical plane. We draw the attention that the uncertainty of the proposed framework is highly reduced in the case of videos made specifically for velocimetry applications, in comparison to traditional methods. Table (5.1) details these parameters and their descriptions.

Results shown in Chapter (4) are physically plausible and in agreement with those taken by an ADCP device.

Future work

In Chapter (2) we derived a motion estimation model that has a physical meaning and interprets the motion of a river surface as a transport equation with a diffusion term. The diffusion term weight is modelled using a widely-known turbulence model. The parameter α that regulates the smoothness of the vector field has no clear physical meaning and its value is chosen empirically. A study on this parameter might gives a hint on its physical meaning and consequently derive a way to better estimate it. Another track of research could be to investigate optical flow technique in which the turbulence models could be estimated (instead of choosing pre-defined models) as demonstrated by Cai et al. (2017). In the same chapter we talked about the importance of a ground-truth data and how it helped advancing conventional optical flow algorithms. Even if the results based on trajectory reconstruction helped verify the results, a realistic physics-based 3D simulation of rivers to generate ground truth free surface velocity would be interesting for highly accurate motion estimation models for rivers.

In chapter (3), the question of auto-calibration is treated in a general fashion without consideration of river image sequences particularities. It might be interesting to see that if these particularities could be useful to add more constraints to improve the calibration.

For the 3D reconstruction step, we assume the scene has a perpendicular plane to river plane. While this is often the case in urban areas, it is a strong assumption on rivers in the wild. It might be interesting to relax the perpendicularity assumption and allow more angle variations.

Appendix A

Alternative Kruppa equations derivation

Suppose C (*resp.* C') is the image of a conic in a world plane in the first (*resp.* second) image, and that C^* and C'^* are their respective dual line conics. In the epipolar geometry context, C^* and C'^* are defined using only two epipolar lines each. A point conic formed by two distinct lines g and h could be written as $C = gh^T + hg^T$. Points in g satisfy $g^T p = 0$ and are on the conic since they satisfy $p^T C p = (p^T g)(h^T p) + (p^T h)(g^T p) = 0$. Points on the second line h satisfying $h^T p = 0$ also satisfy $p^T C p = 0$. Thus matrix C is symmetric with rank =2. This is a *degenerate* point conic matrix since it is not full rank. The two tangent epipolar lines then give a degenerate point conic of the form $p^T [e]_\times C^* [e]_\times p$ and similarly in the second view $p'^T [e']_\times C'^* [e']_\times p'$.

There is procedure called *transfer via a plane* (Hartley and Zisserman, 2004) (p242) in which two correspondent points transfer via the plane-induced homography $p' = Hp$. Point conics transfer between views according to $C'_p = H^{-T} C_p H^{-1}$. This could be easily verified by substituting in $p^T C p = 0$. Taking the second degenerate point conic and transform it towards the first by using:

$$[e']_\times C'^* [e']_\times = H^{-T} [e]_\times C^* [e]_\times H^{-1}$$

It is also shown that $F = H^{-T} [e]_\times$(Hartley and Zisserman, 2004) (Result 9.1, p243). On substitution in above equation

$$[e']_\times C'^* [e']_\times = FC^* F^T$$

In the special case of the DIAC the plane H is the plane at infinity. Fixed camera parameters means $C'^* = C^*$ which finally leads to Kruppa equations

$$[e']_\times C^* [e']_\times = FC^* F^T$$

Bibliography

Adrian, R., Jan. 1991. Particle-Imaging Techniques for Experimental Fluid-Mechanics. Annual Review of Fluid Mechanics 23 (1), 261–304.

Agarwal, S., Mierle, K., Others, 2010. Ceres Solver.
URL `http://ceres-solver.org`

Alvarez, L., Deriche, R., Sánchez, J., Weickert, J., Mar. 2002. Dense Disparity Map Estimation Respecting Image Discontinuities: A PDE and Scale-Space Based Approach. Journal of Visual Communication and Image Representation 13 (1), 3–21.

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., Szeliski, R., Mar. 2011. A Database and Evaluation Methodology for Optical Flow. International Journal of Computer Vision 92 (1), 1–31.

Barron, J. L., Fleet, D. J., Beauchemin, S. S., 1994. Performance of optical flow techniques. INTERNATIONAL JOURNAL OF COMPUTER VISION 12, 43–77.

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., Jun. 2008. Speeded-up robust features (surf). Comput. Vis. Image Underst. 110 (3), 346–359.

Bergen, J. R., Anandan, P., Hanna, K. J., Hingorani, R., May 1992. Hierarchical Model-Based Motion Estimation. In: Computer Vision — ECCV 92. Springer, Berlin, Heidelberg, pp. 237–252.

Bhattacharyya, A., 1943. On a Measure of Divergence between two Statistical Populations Defined by Their Probability Distributions. Bull. Calcutta Math. Soc. Proc 12, 99.

Bill Triggs, Jun. 1997. Autocalibration and the absolute quadric. International Conference on Computer Vision & Pattern Recognition (CVPR '97), 609–614.

Black, M. J., Anandan, P., Jan. 1996. The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields. Computer Vision and Image Understanding 63 (1), 75–104.

Blake, A., Zisserman, A., 1987. Visual Reconstruction. MIT Press.

Brox, T., Bruhn, A., Papenberg, N., Weickert, J., Jan. 2004. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: Computer Vision - ECCV 2004. Vol. 3024 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 25–36.

Butler, D. J., Wulff, J., Stanley, G. B., Black, M. J., Oct. 2012. A Naturalistic Open Source Movie for Optical Flow Evaluation. In: European Conf. on Computer Vision (ECCV). Part IV, LNCS 7577. Springer-Verlag, pp. 611–625.

Cai, S., Mémin, E., Dérian, P., Xu, C., Nov 2017. Motion Estimation under Location Uncertainty for Turbulent Fluid Flows. Experiments in Fluids 59 (1), 8.

Carlier, J., Wieneke, B., 2005. Report 1 on Production and Diffusion of Fluid Mechanics Images and Data.

Cassisa, C., Simoens, S., Prinet, V., Shao, L., Dec. 2011. Subgrid Scale Formulation of Optical Flow for The Study of Turbulent Flow. Experiments in Fluids 51 (6), 1739–1754.

Chen, X., Zillé, P., Shao, L., Corpetti, T., Jan. 2015. Optical Flow for Incompressible Turbulence Motion Estimation. Experiments in Fluids 56 (1).

Corpetti, T., Heitz, D., Arroyo, G., Mémin, E., Santa-Cruz, A., Oct. 2005. Fluid Experimental Flow Estimation Based on an Optical-Flow Scheme. Experiments in Fluids 40 (1), 80–97.

Corpetti, T., Mémin, E., Pérez, P., 2002. Dense estimation of fluid flows. IEEE Trans. Pattern Anal. Machine Intell 24, 365–380.

Creutin, J. D., Muste, M., Bradley, A. A., Kim, S. C., Kruger, A., Jun. 2003. River gauging using PIV techniques: a proof of concept experiment on the Iowa River. Journal of Hydrology 277 (3–4), 182–194.

Cui, G. X., Xu, C. X., Fang, L., Shao, L., Zhang, Z. S., Jul. 2007. A New Subgrid Eddy-Viscosity Model for Large-Eddy Simulation of Anisotropic Turbulence. Journal of Fluid Mechanics 582, 377–397.

Dramais, G., Le Coz, J., Camenen, B., Hauet, A., Dec. 2011. Advantages of a Mobile LSPIV Method for Measuring Flood Discharges and Improving Stage-Discharge Curves. Journal of Hydro-environment Research 5 (4), 301–312.

Fujita, I., Muste, M., Kruger, A., 1998. Large-Scale Particle Image Velocimetry for Flow Analysis in Hydraulic Engineering Applications. Journal of Hydraulic Research 36 (3), 397–414.

Fujita, I., Watanabe, H., Tsubaki, R., 2007. Development of a non-intrusive and efficient flow monitoring technique: The space-time image velocimetry (stiv). International Journal of River Basin Management 5 (2), 105–114.

Fusiello, A., 1999. The Mendonça and Cipolla Self-calibration Algorithm Experimental Evaluation.

Gualtieri, C., Angeloudis, A., Bombardelli, F., Jha, S., Stoesser, T., Apr. 2017. On the Values for the Turbulent Schmidt Number in Environmental Flows. Fluids 2 (2), 17.

Ha, H., Im, S., Park, J., Jeon, H. G., Kweon, I. S., June 2016. High-quality depth from uncalibrated small motion clip. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5413–5421.

Harris, C., Stephens, M., 1988. A combined corner and edge detector. In: In Proc. of Fourth Alvey Vision Conference. pp. 147–151.

Hartley, R. I., Jun. 1997a. In Defense of the Eight-Point Algorithm. IEEE Trans. Pattern Anal. Mach. Intell. 19 (6), 580–593.

Hartley, R. I., Feb. 1997b. Kruppa's equations derived from the fundamental matrix. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (2), 133–135.

Hartley, R. I., Jan. 1998. Chirality. International Journal of Computer Vision 26 (1), 41–61.

Hartley, R. I., Zisserman, A., 2004. Multiple View Geometry in Computer Vision, 2nd Edition. Cambridge University Press, ISBN: 0521540518.

Heas, P., Memin, E., Papadakis, N., Szantai, A., Dec. 2007. Layered Estimation of Atmospheric Mesoscale Dynamics From Satellite Imagery. IEEE Transactions on Geoscience and Remote Sensing 45 (12), 4087–4104.

Heitz, D., Héas, P., Mémin, E., Carlier, J., Oct. 2008. Dynamic Consistent Correlation-Variational Approach for Robust Optical Flow Estimation. Experiments in Fluids 45 (4), 595–608.

Heitz, D., Mémin, E., Schnörr, C., Mar. 2010. Variational Fluid Flow Measurements from Image Sequences: Synopsis and Perspectives. Experiments in Fluids 48 (3), 369–393.

Herrera, D., Kannala, C. J., Heikkila, J., Mar. 2016. Forget the checkerboard: Practical self-calibration using a planar scene. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1–9.

Horn, B. K. P., Schunck, B. G., Aug. 1981. Determining Optical Flow. Artificial Intelligence 17 (1–3), 185–203.

Huang, J., Lai, S., Cheng, C., 2007. Robust fundamental matrix estimation with accurate outlier detection.

Jodeau, M., Hauet, A., Paquier, A., Coz, J. L., Dramais, G., 2008. Application and Evaluation of ls-piv Technique for the Monitoring of River Surface

Velocities in High Flow Conditions. Flow Measurement and Instrumentation 19 (2), 117 – 127.

Kolmogorov, A., 1941. The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds' Numbers. Akademiia Nauk SSSR Doklady 30, 301–305.

Le Boursicaud, R., Pénard, L., Hauet, A., Thollet, F., Le Coz, J., 2016. Gauging extreme floods on youtube: application of lspiv to home movies for the post-event determination of stream discharges. Hydrological Processes 30 (1), 90–105.

Le Coz, J., Jodeau, M., Hauet, A., Marchand, B., Le Boursicaud, R., 2014. Image-based Velocity and Discharge Measurements in Field and Laboratory River Engineering Studies Using the Free FUDAA-LSPIV Software. In: Proceedings of the International Conference on Fluvial Hydraulics, RIVER FLOW 2014. pp. 1961–1967.

Liu, C., Jun. 2009. Beyond Pixels: Exploring New Representations and Applications for Motion Analysis. Ph.D. thesis, Massachusetts Institute of Technology.

Liu, T., Shen, L., Nov. 2008. Fluid Flow And Optical Flow. Journal of Fluid Mechanics 614, 253–291.

Longuet-Higgins, H. C., Sep. 1981. A computer algorithm for reconstructing a scene from two projections. Nature 293 (5828), 293133a0.

Lowe, D. G., Nov. 2004. Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision 60 (2), 91–110.

Lowry, C. S., Fienen, M. N., 2013. Crowdhydrology: Crowdsourcing hydrologic data and engaging citizen scientists. Groundwater 51 (1), 151–156.

Lucas, B. D., Kanade, T., 1981. An Iterative Image Registration Technique with an Application to Stereo Vision. pp. 674–679.

Luong, Q.-T., Faugeras, O. D., Mar. 1997. Self-Calibration of a Moving Camera from Point Correspondences and Fundamental Matrices. International Journal of Computer Vision 22 (3), 261–289.

Mahalanobis, P. C., Apr. 1936. On the Generalised Distance in Statistics. In: Proceedings National Institute of Science, India. Vol. 2. pp. 49–55.

Maybank, S. J., Faugeras, O. D., Aug. 1992. A theory of self-calibration of a moving camera. Int J Comput Vision 8 (2), 123–151.

Mémin, E., Pérez, P., May 1998a. Dense Estimation and Object-based Segmentation of the Optical Flow with Robust Techniques. IEEE Transactions on Image Processing 7 (5), 703–719.

Mémin, E., Pérez, P., Jan. 1998b. A Multigrid Approach for Hierarchical Motion Estimation. In: Sixth International Conference on Computer Vision, 1998. pp. 933–938.

Mendonça, P. R. S., Cipolla, R., 1999. A Simple Technique for Self-Calibration.

Muste, M., Fujita, I., Hauet, A., Apr. 2008. Large-scale particle image velocimetry for measurements in riverine environments. Water Resour. Res. 44 (4), W00D19.

Muste, M., Hauet, A., Fujita, I., Legout, C., Ho, H.-C., 2014. Capabilities of Large-Scale Particle Image Velocimetry to Characterize Shallow Free-Surface Flows. Advances in Water Resources 70, 160 – 171.

Pénard, L., Le Boursicaud, R., Thollet, F., Lagouy, M., Khalid, M., Le Coz, J., Hauet, A., Jul. 2015. Application of Image-Based River Velocimetry Techniques to Flood Home Videos. In: 2015 IEEE International Geoscience and Remote Sensing Symposium. Milano, Italy.

Prandtl, L., 1925. Z. angew. Math. Mech. 5 (1), 136–139.

Ray, N., Oct. 2011. Computation of Fluid and Particle Motion From a Time-Sequenced Image Pair: A Global Outlier Identification Approach. IEEE Transactions on Image Processing 20 (10), 2925–2936.

Simpson, J. J., Gobat, J. I., May 1994. Robust velocity estimates, stream functions, and simulated lagrangian drifters from sequential spacecraft data. IEEE Transactions on Geoscience and Remote Sensing 32 (3), 479–493.

Slesareva, N., Bühler, T., Hagenburg, K. U., Weickert, J., Bruhn, A., Karni, Z., Seidel, H.-P., Jun. 2007. Robust Variational Reconstruction from Multiple Views. In: Image Analysis. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 173–182.

Sturm, P., Cheng, Z. L., Chen, P. C. Y., Poo, A. N., Jul. 2005. Focal length calibration from two views: method and analysis of singular cases. Computer Vision and Image Understanding 99 (1), 58–95.

Sun, D., 2013. From Pixels to Layers: Joint Motion Estimation and Segmentation. Ph.D. thesis, Brown University.

Sun, D., Roth, S., Black, M., 2014. A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles Behind Them. International Journal of Computer Vision 106 (2), 115–137.

Suter, D., Jun. 1994. Motion Estimation and Vector Splines. In: , 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94. pp. 939–942.

Sutton, M., Yan, J., Tiwari, V., Schreier, H., Orteu, J., 2008. The effect of out-of-plane motion on 2d and 3d digital image correlation measurements. Optics and Lasers in Engineering 46 (10), 746 – 757.

Tauro, F., Piscopia, R., Grimaldi, S., 2017. Streamflow observations from cameras: Large-scale particle image velocimetry or particle tracking velocimetry? Water Resources Research 53 (12), 10374–10394.

Tominaga, Y., Stathopoulos, T., 2007. Turbulent Schmidt Numbers for CFD Analysis with Various Types of Flowfield. Atmospheric Environment 41 (37), 8091 – 8099.

Tsai, R., August 1987. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. IEEE Journal on Robotics and Automation 3 (4), 323–344.

Welber, M., Le Coz, J., Laronne, J., Zolezzi, G., Zamler, D., Dramais, G., Hauet, A., Salvaro, M., 2016. Field Assessment of Noncontact Stream Gauging Using Portable Surface Velocity Radars (SVR). Water Resources Research 52 (2), 1108–1126.

Yang, S., Maturana, D., Scherer, S., May 2016a. Real-time 3d scene layout from a single image using Convolutional Neural Networks. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). pp. 2183–2189.

Yang, S., Song, Y., Kaess, M., Scherer, S., Oct. 2016b. Pop-up SLAM: Semantic monocular plane SLAM for low-texture environments. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1222–1229.

Zhang, Z., Nov 2000. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (11), 1330–1334.