École polytechnique de Louvain

Development of an application for river discharge measurement

Author: **Simon Renard**
Supervisor: **Sandra Soares Frazao**
Readers: **Bruno Dehez, Olivier Carlier d'Odeigne, Paul Fisette**
Academic year 2019–2020
Master [120] in Mechanical Engineering

Abstract

River flow measurements are not recent and are required in many different fields. Moreover, since the river flow fluctuates over time it must be measured regularly. Those fluctuations have been amplified with the climate change during the last years. However, the most widespread techniques are often intrusive, expensive and their deployment can be complicated especially in extreme conditions like floods. Therefore, the interest in imaging techniques which address these issues has been growing in the last years. Each technique is focused on one aspect of the discharge analysis. Thus, they are often implemented in different software which makes the analysis more tedious to perform.

The aim of this work is to develop a stand-alone application implementing imaging techniques which computes the discharge of a river. The application performs the analysis based on a video sequence of the river which needs to be seeded (e.g. with wood shavings) and the knowledge of its cross-section.

During the development of the application the techniques studied in the previous Master Theses will be presented and modified. New methods for the water level measurement will also be developed. Finally, several image processing techniques will be implemented allowing to fix potential distortions in the images or to enhance their quality before the analysis. At the end of this document the results obtained by the application developed in the frame of this work and out-of-the-shelf image-based discharge computation software will be compared.

The results are encouraging since both provide similar results on a real river example. It is still necessary to benchmark the application using a larger set of examples and comparing the different implemented water level estimation methods. Some areas for improvement are also presented in the end regarding the velocity analysis and the features that could be interesting to implement.

Acknowledgements

I would like to express my gratitude to my supervisor Sandra Soares Frazao for giving me the opportunity to work on this really interesting and challenging subject. Thank you also for the time you spend to help me with your comments, remarks and suggestions along this year.

I am also grateful to the Discharge application developing team, more precisely to Salvador Peña, for answering my questions on their application.

Last but no least I would like to thank my friends and my family for their constant support and advice, especially my uncle for his careful reading.

Contents

Chapter 1

Introduction

River flow measurements are required when dealing with assessment and management of water resources (e.g. irrigation supplies, drinking water, navigation) and while designing water-related structures (e.g. bridges, sewage treatment, urban drainage schemes). These measurements are useful to prevent or warn in case of flood. In addition, such data are critical in hydropower plants to assess the available hydroelectric potential. Nuclear power plants located next to a river use it to condense the low-pressure steam by running a large amount of water through the condensers in a single pass and discharging it back into the river a few degrees warmer. River discharge measurements are important here to compute the heat exchanges capacity and prevent high raise in the river temperature to not endanger the environment.

It is important to measure river flow on a regular basis. The latter fluctuates, mirroring the complex hydroclimatic processes and the riverbed evolution. Global warming is a new concern and its impact is uneven on river flow patterns. Recording data is thus the key to identify, quantify and interpret hydrology trends.

In hydrology, the discharge is defined as the volume of water passing through a cross-section of the river per unit of time. This means that in practice the discharge cannot be measured directly because it relies on the knowledge of several features: cross-section geometry, water level and velocity distribution across studied section. This also limits the precision achievable because of the accumulation of uncertainties. Another challenge is to infer the mean value of the velocity along each vertical of the river's cross-section. Theoretically this mean value is the integration of the velocity profile over the vertical. In practice the data available is either the surface velocity or the velocity of n-points along each vertical.

To measure the features mentioned above many techniques and devices have been developed over the years allowing to instrument the river. Some of them must be partially or totally submerged. For example, the different current meters to measure the velocity. This immersion can be a problem in the case of flood where rubble is carried away by the flow and can harm the device. Some devices require a bridge or an above-water installation. For example, water level measurement using radar or acoustic waves. Such structures are also required when the river is too deep and does not allow the operator to directly perform the measurements.

These techniques and structures are limited by their cost and their difficulty of deployment in the environment being studied. They are fixed and located on a river portion limiting their study zone. This limitation makes observation of the evolution of river flow due to the river tributaries more tedious and not portable.

All these reasons justify the interest of developing non-intrusive, cheaper, and more easily deployed techniques. In the continuity of the previous Master Theses, the present study will focus on imagery-based techniques because this approach meets these requirements.

The usage of drones for example offers an affordable means to make measurements everywhere and even in difficult conditions such as during flooding where it guarantees both the operator and device safety. Drone's portability makes measurements on different sections of river easier (which is interesting in the case of measuring impact of tributaries flow impact) as well as for different rivers. All this while taking advantage of the swift developments of drones or UAVs (Unmanned Aerial Vehicles) during the recent years: lower price, reduced size, increased stability, better battery, etc.

Over the years, others Master Theses have been done on two imagery-based techniques that this work will also use:

– **Large-Scale Particle Image Velocimetry (LSPIV):** this technique is used for computing the surface velocity of the river. Its basic principle is simple: some particles are thrown into the river then from a series of images of the flow (e.g. frames of a video), the displacement of these particles is computed. Knowing the time span between each image, a surface velocity is inferred.

– **Photogrammetry:** this technique allows to reconstruct a three-dimensional model on base of pictures of an object. It can be combined with the river's cross-section to retrieve the water level.

However, the methods that they have studied or developed are scattered since each

Master Thesis has been mainly focused on one aspect of the analysis. To benefit from their work, one must seek the different program files or software used and try to put the different results together making the process more tedious. This is where the application will help by gathering all the work that has been achieved so far in one place.

The aim of this work is to develop an application applying imaging techniques and computing the discharge on the base of a known river cross-section.

The design goals for this application are the following:

- Work stand-alone. It should provide the requested results without dependency on other applications
- Be user friendly. To achieve this goal, it proposes a sensible set of default configuration parameters at each processing step. Experienced user can fine tune and modify those parameters
- Gather the work of previous Master Theses.

The application computes the discharge using a description of the river cross-section and a video sequence of the river surface previously seeded with particles such as wood, e.g. as can be captured by a drone-borne camera.

The application proposes different steps to process those data and several methods for some of them. Those steps are described in the Figure 1.1.

After the input step, the video frames can be modified thanks to some image processing. The latter aim to correct several sources of distortions in the frames, e.g. stabilize the frames of a video taken by a drone.

For the water level computation three different methods are available. The one called Adapted Hellebaut's method is the adaptation made of the technique combining the photogrammetry and the cross-section introduced earlier. The others two methods (Angle and Height method, Image method) are new ones. They are both based on the following principle: combined the real river width and the cross-section curve to deduce the water level.

Then a selection and enhancement of the frames is proposed before the velocity analysis which is performed using the LSPIV technique.

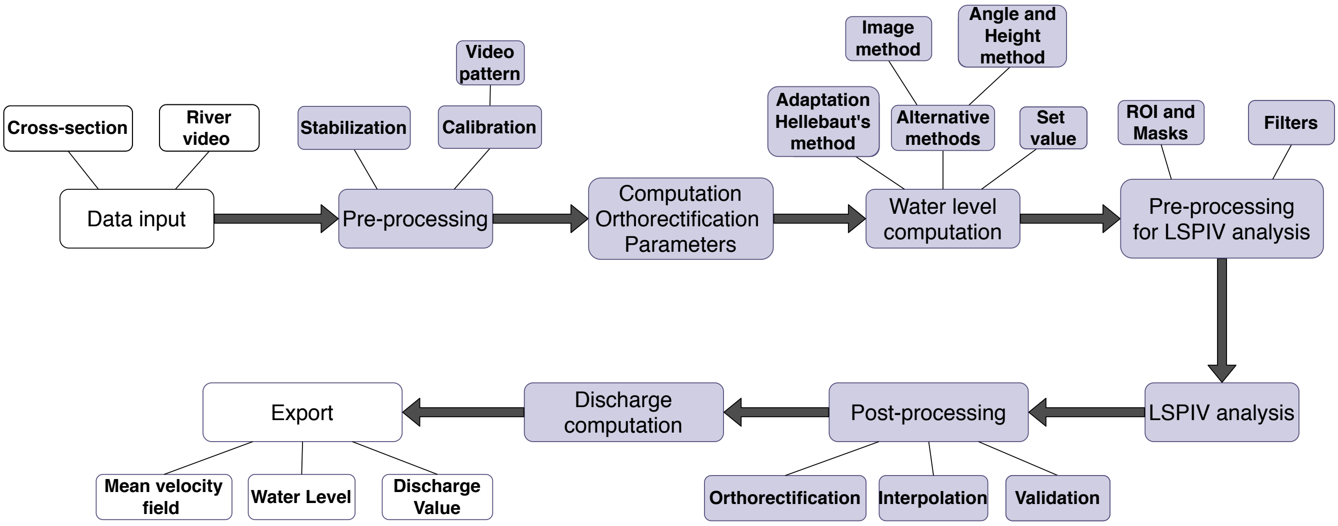


Figure 1.1: Workflow of the new developed application.

Finally, the discharge is computed and the discharge, water level and velocity field are saved in a file.

During the development of this software application:

- New methods have been developed
- Some methods have been modified and enhanced
- Processes on images have been implemented to correct possible distortions
- Execution speed of the stabilization method has been improved using new implementation
- Methods have been re-implemented in a more modern and unified programming language

This report is organized as followed:

- **Part I, Preliminary steps in the development of the application:** In the first part all the foundations for the application development are covered. Understand the problematic through a review of the literature on the subject. Get inspired by existing solutions for the features to implement. Finally choose the tools for the development of the application.

- **Chapter 2, State of the art:** Description of the different elements involved in discharge measurement. For each element, an overview of the different techniques to measure it will be done. Their functioning as well as their advantages and disadvantages will be listed. The techniques are presented in the following order, first the non-imaging then the imaging techniques.

- **Chapter 3, Choice of tools and integrated development environment (IDE):** The different tools used to develop the application as well as the switch from MATLAB to Python will be covered in this chapter.

- **Chapter 4, Existing application and tools:** There are tools, applications that address the same problematic. The functioning of these and how they have inspired the present work will be discussed.

- **Part II, Content of the new developed application:** In the second part all features that the application offers are covered. This include the different methods required for the discharge computation along with other options that enhance the results.

- **Chapter 5, Work on water level methods:** The modification made on the Hellebaut's method to make it more automatic will be described. Two new methods aiming to ease the water level measurement will also be proposed.

- **Chapter 6, PIV software/packages used for the velocity analysis:** The tool that has been used in the previous Master Theses and the one used in this work, to perform a PIV analysis will be described. A comparison of their features will also be given.

- **Chapter 7, Image processing:** In the application, the images will go through a series of processes that will be covered in this chapter. It encompasses the calibration of the camera to perform lens distortion correction, application of filters, orthorectification from perspective correction and stabilization for images extract from movie taken by a drone.

- **Chapter 8, Discharge Computation:** This chapter describes the combination of all required measurements to provide a discharge value. In summary it explains how the surface velocity is used, how the wetted-area is computed and how the velocity vectors on the section of interest are retrieved.

- **Part III, Overview of the new developed application:** In the last part the instructions to use the application are given by going through the different screens displayed at each step of the analysis. An evaluation of the application is done and possibilities for improvement are explored.

- **Chapter 9, Use of the new developed application:** This chapter covers how to use the application. This includes the different options proposed at each step and the explanation of the different choices and the inputs required as well as their format. At the end of the chapter an example on a real river using the application will be done.

- **Chapter 10, Evaluation and room for improvement:** In this part a review on the application will be given with the highlighting of its strong and weak points. The possibilities to improve it will also be discussed.

Part I

Preliminary steps in the development of the application

Chapter 2

State of the art

Previous Master Theses were focused on imaging techniques to compute the discharge features (cross-section geometry, water level and velocity of the flow) and this work is no exception. To better understand this choice and what have been done, this chapter will be divided in two sections.

In the first section several non-imaging techniques will be assessed and compared according to their precision and applicability in different situations. The techniques will be grouped according to the feature of the discharge they measure.

The second section will be dedicated to the imaging techniques used/developed in previous Master Theses that will be included in this work. The measurement problem of the bathymetry will also be discussed. The theory behind these techniques will be explained along with the add-ons of the previous works. It will be a general description of what already existed. However, the way they have been implemented in the application will not be covered here but in the second part of the report.

2.1 Popular non-imaging techniques:

2.1.1 Bathymetry survey

Bathymetry is the measurement of the water depth at several places in any water bodies [1]. Then from the gathering of measurements a bathymetric map is created to ease the study of water bodies floor. To perform bathymetry surveys many techniques are available. The following non exhaustive list summarizes the most common ones:

▷ Acoustic depth measurements

Principle

A transducer generates acoustic pulses, send them to the waterway bottom and measures the elapsed time that it takes to bounce back to the transducer [2]. Then the depth can be easily retrieved if the velocity of sound propagation in the water column is known. Using this principle while changing the transducer position, a bathymetric map can be produced. The principle is illustrated in the Figure 2.1. The position of each measurement is recorded by a differential Global Positioning system (dGPS) onboard. The principle of dGPS will be covered later.

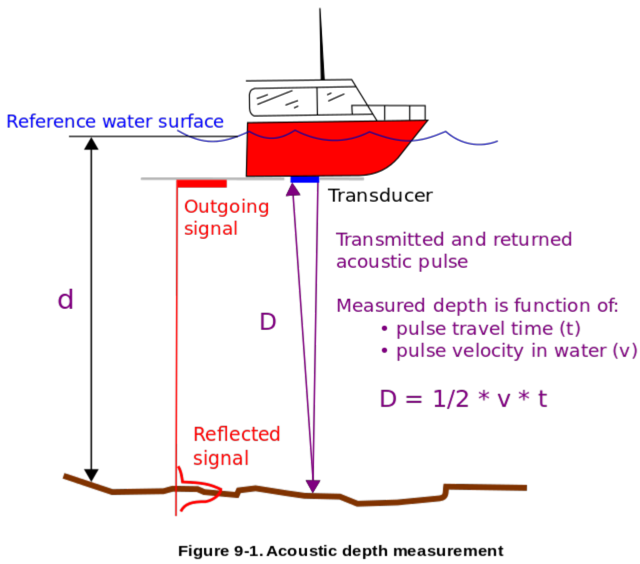


Figure 2.1: Acoustic depth measurement principle. Figure from [3]

.

The velocity of sound in water is dependent on multiple factors such as water temperature and salinity. Since these properties can strongly vary with depth in a river, the velocity is not constant in the water column. In shallow water to address this problem a mean velocity of sound is usually assumed. This mean velocity can be computed either by calibration (known depth compared to computed depth) or directly using a velocity probe at different depth.

Another factor to consider is the acoustic frequency which is proportional to the attenuation of signal in water. The higher the frequency is the higher the attenuation will be, meaning lower penetration in the water body. However high frequency affords more detailed bottom definition. The optimum transducer frequency is so highly project dependent [4].

The transducer is usually placed on a ship's hull. To track its positions a Global Position System (GPS) signal is required along with an Inertial Navigation System (INS) to be able to correct the movement of the ship along each axis (roll, pitch, way, and azimuth) and its speed variations.

The simplest system available is the Single Beam Echo Sounder (SBES) where only the depth directly below the ship is found. This can take some time when the area to be covered is large. Another system is the Multi Beam Echo Sounder (MBES). Its principle is basically the same as the SBES except that it is a set of pulses in a fan shape that are sent. So, the depth measured is not only the one directly below the ship but also on either side. For example, the ship James Clark Ross has 288 beams and can map a swath width of about 4 times the water depth [5].

Advantages

Development in acoustic electronics in the 1970's and 80's allowed precise control of the phase and amplitude of the acoustic signal allowing high resolution results. Today the development is on the increase of digital processing power making recording and computing simpler. Software for data acquisition and satellite positioning systems (e.g. GPS) have also made it more efficient.

Drawbacks

Main drawbacks of this technique are the time and cost associated with measurements which are taken from a ship in deep water or a small vessel in shallow waters. Ship time is costly and operating in shallow water is more tedious due to the decrease of the swath width requiring more time to cover the whole area. Moreover, to avoid biasing the results by the ship or vessel movement they must be equipped to track their position and orientation. This technique is not suited for small non-navigable rivers. This is an issue since these rivers can be responsible of many floods of small scale but where the damages are important.

Scope of application

The MBES is principally used in deep water surveys such as seabed studies since in shallow water the swath width decreases losing the advantage of using a MBES [6]. Despite the MBES or new techniques developments the SBES is still nowadays the most widely used depth measurement technique for surveying river. Echo-sounder are also used in fisheries surveys to access fish density [7].

▷ **Airborne Laser System**

Principle

It consists of a laser scanning system, a global positioning system (GPS) to measure the aircraft position and an inertial measurement unit (IMU) which gives the roll, pitch, and yaw of the aircraft [8]. The technique is called LIDAR for Light Detection And Ranging. This system emits laser pulses at two frequencies blue or green and infrared and records the reflection on both the water surface and the waterbody floor. Indeed, the infrared will bounce off the water surface back to the sensor while the blue or green light penetrates through the water before getting reflected on the waterbody floor. Then the water depth is estimated from the elapsed time between the two reflections. An illustration of this principle is shown on Figure 2.2. The position of each measurement is again recorded using a dGPS onboard allowing to produce a bathymetric map of the region studied.

Advantages

In shallow waters the laser systems are efficient mainly due to their high surveying speed and their swath width, which is independent of water depth. In comparison with the MBES where the surveying speed is low, and the swath width proportionate to the water depth. Moreover, they have the benefit of avoiding navigation hazard using airborne systems and can covered places where boats cannot access. Costal surveys using LIDAR allows measuring both land and water under one approach.

Drawbacks

The propagation of light is influenced by the presence of material suspended in the water. Indeed, blue, or green light undergoes attenuation due to the absorption and scattering when encounters particles. That is why LIDAR system have a limited maximum depth at which it can work. A typical range in which it can operated in optimal conditions (no suspended particles) is from 1.5 to 60 meter [8]. LIDAR sensors are mounted on mobile platforms such as airplanes which requires instrumentation to determine their position and orientation. Compare to other methods operational cost is substantially higher. Therefore nowadays LIDAR sensors on drone are developed.

Scope of application

This is mainly used to produce high resolution bathymetric map of the seafloor and topographic features rapidly.

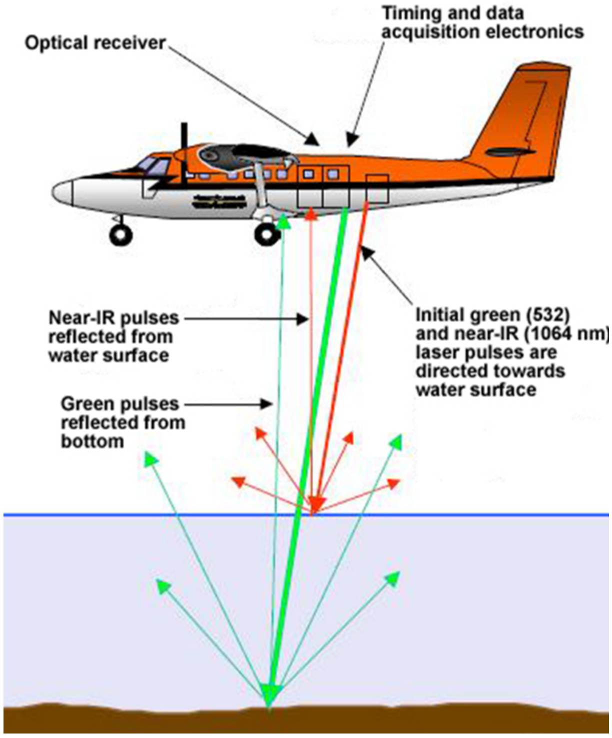


Figure 2.2: Principle of LIDAR hydrographic surveying. Figure from [9]
.

▷ <u>**Differential Global Positioning System (DGPS)**</u>

Principle

This technique is made for shallow rivers since it implies that the operator cross on foot the river during the bathymetry survey. The riverbanks must also be measured which requires that they can be easily accessible. During this survey, a base station and a mobile one will be used, as well as an antenna to communicate between them. A schema of this set up is shown on the Figure 2.3. This set-up is part of the dGPS requirements whose principle which be

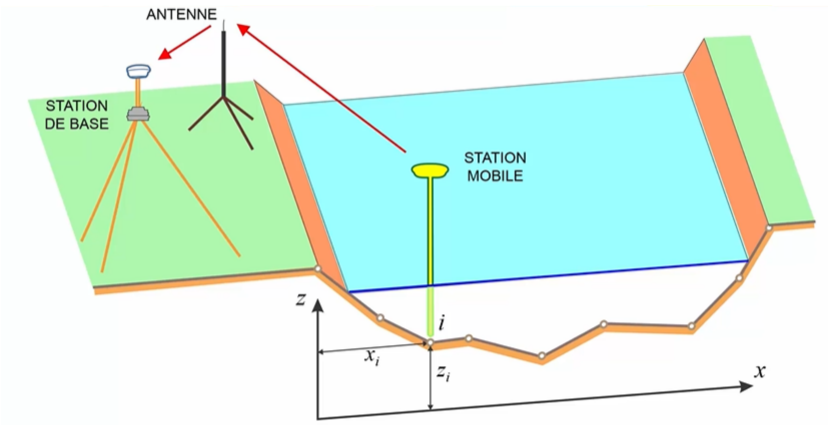


Figure 2.3: Principle of DGPS hydrographic surveying. Figure from [10]

.

covered later. The mobile station is composed of a GPS which is mounted on top of a rod whose bottom part must be in contact with the ground during the measurements. This way the GPS is not underwater during the survey part in the water. Then knowing the length of the rod, the altitude of the rod tip can be measured by tracking the GPS position. In addition to the altitude of the different points the GPS gives their latitude and longitude. These latter are particularly interesting when bathymetric map must be generated. Since the measurements takes time there are generally only taken along one cross section. A measuring tape or marked rope is also used to keep track at which distances along the river cross-sections the measurements are made. It is deployed between the two shores and perpendicularly to the cross-section studied. Then measurements are made each x cm by the operator thanks to the rope marks or to the measuring tape. Some dGPS as the Leica 1200 [11] which have been used in previous Master Theses come with a keyboard, touch screen and intuitive interface making the recording of data easier.

DGPS general principle

DGPS is an enhancement to the Global Positioning System (GPS) which provides improved location accuracy. DGPS uses a fixed based station of known position, then for each satellite a correction signal can be sent to the GPS receiver [12]. Indeed, the real distance between the base station and satellites is known and can be compared to the measured distance. This gives a correction signal that can be sent to the GPS receiver which will apply it when measuring. The receiver needs to be close to the base station since this correction works because both signals travel through the same parts of the atmosphere. Real Time Kinematic (RTK) is a type of DGPS which enhanced the transmission of correction signals allowing centimetric precision [13].

Advantages

The cost is low and the device user-friendly. The portability is also convenient since one device can be used for multiple rivers.

Drawbacks

The measurements are tedious to make if a dense bathymetric map is wanted. During the recording of the depths the operator can disturb the bathymetry by walking on the waterbody floor and induce errors by pushing down to hard the device in the floor.

Scope of application

It is mainly used in shallow rivers where the operators can step into the river and cross it on foot. This assumes that the river is accessible. It can also be used to referenced ground control points placed on the shore.

2.1.2 Water level measurement

▷ **Staff Level gauges**

Principle

It is basically a ruler which is put permanently along the shore or on structure present in the river, an example is shown on the Figure 2.4.

Advantages

It can provide a quick, visual indication to the operator on what is the water level of the river. It is cheap and easy to install.

Figure 2.4: Example of staff level gauges. Figure from [14]
.

Drawbacks

River's surface is rarely completely still but have some amount of surface ripple or turbulence making the reading more difficult. To make the measurement easier for the operator the staff gauges should be located where the water is relatively turbulence free. Since the ruler is in the water rubbles can harm it and in case of flood it might be completely submerged.

Scope of application

This is often used in rivers due to its low cost and combine with other more accurate techniques to check if the results are similar. This can also be used inside a tank to observe its liquid level.

▷ **Ultrasonic and Radar level sensors**

Principle

These two different techniques use the same principle, pulses are aimed to the water surface and the time that it takes to get to the water surface, bounces off and get back to the sensor is recorded. Then from it the distance to the water surface is computed and finally the water level. The only difference is the type of pulses emitted which are either ultrasonic sound pulses or microwave ones. Their principle is shown in the Figure. 2.5

Advantages

Being out of the water these techniques offer a better durability since they are not affected by the water and rubbles that can be vehicle by it. The radar is less sensitive to the temperature and the surface conditions than the ultrasonic technique providing more accuracy in the results [15].

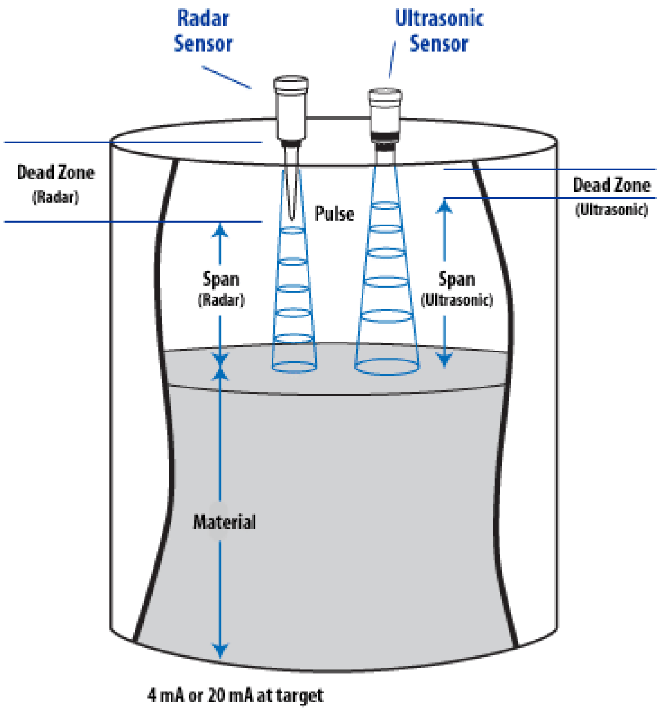


Figure 2.5: Principle of radar and ultrasonic sensors for level measurements. Figure from [16]

.

Drawbacks

Since the sensor needs to be above the water it is often mounted on a bridge or on an above water structure. These structures must allow the sensors to be placed away from the shore to avoid edge effects [17]. The ultrasonic sensor is sensitive to the surface conditions such as the presence of foam, some debris, or even turbulence. This affects the accuracy this sensor can achieved. Environment conditions such as high temperature or humidity have also an impact on the results for the ultrasonic sensor.

Scope of application

Being non-contact sensors, they are especially useful in rivers where floods can occur. Or in tank containing liquid highly corrosive or hot (more for the radar sensor).

▷ **Encoder**

Principle

Here an instrument which mechanically couples the position of the water surface to an electrical rotary encoder is used [18]. The link between the two is a wire which is connected to a water surface float at one end and a counterweight on the other end. Then the encoder gives an outcome that relates to the position of the water surface. A typical installation is shown on the Figure.2.6.

Advantages

In can work even in flood conditions. Being one of the earliest automated equipment used to measure and record water level, many stilling wells have already been built. This technique also provides an excellent linearity and accuracy in the measurements.

Drawbacks

To work this needs to be installed inside a stilling well which requires high installation costs compared to the other types of water level instruments [20].

Scope of application

Any river where such an installation is possible. The same principle can also be applying in tank.

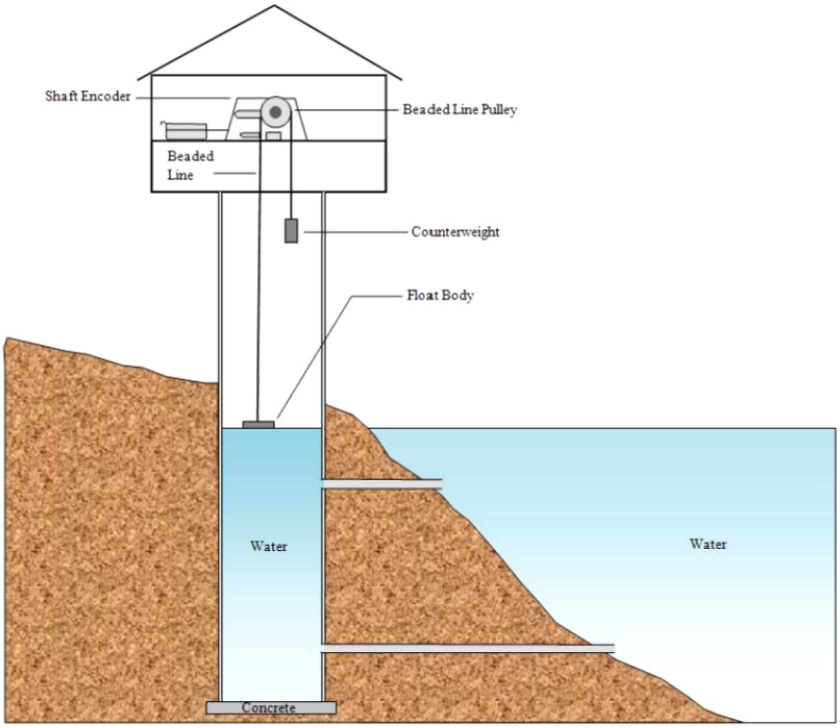


Figure 2.6: Installation for float and stilling well installation. Figure from [19]
.

▷ **Gas bubbler pressure transducer**

Principle

A pressure sensor is coupled to the weight of water through a gas-filled tube [21]. This is illustrated on the Figure.2.7 The underwater end of the tube is fastened at a reference depth while the gas is provided by a gas bottle or a pump. The pressure sensor measures the pressure of the gas needed to displace the liquid in the orifice. Then the depth of water above the orifice can be deduced from it.

Advantages

Compare to the encoder case where stilling wells were needed the installation is cheaper. Another advantage is that the pressure sensor is not directly in contact with the water preventing wear and damage due to debris.

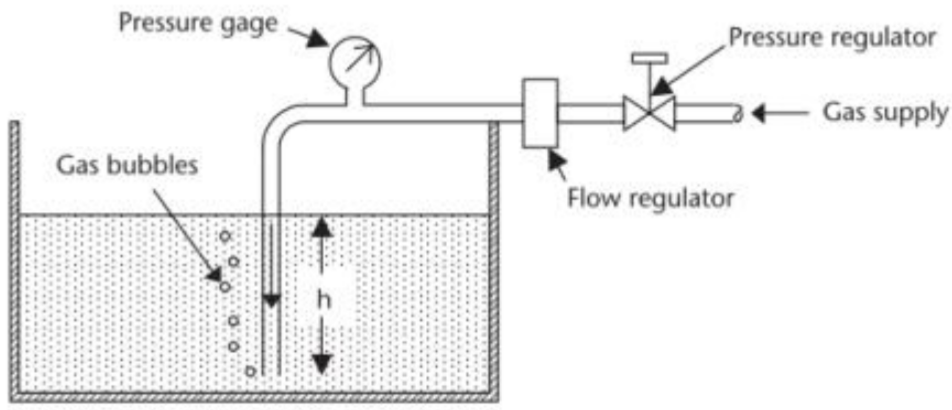


Figure 2.7: Bubbler device for measuring liquid level. Figure from [22]
.

Drawbacks

If there is a change in the liquid density the device must be re-calibrated. It requires pressure tank or other pressures sources.

Scope of application

Bubbler systems are suited for accurate liquid level measurement which explains that they are used in industrial process systems.

2.1.3 Velocity profile measurement

To get the whole velocity profile of a cross-section in the river the measurements are generally made on several verticals along the cross-section. On each vertical the velocity is measured at a finite number of points. The number of points used varies following the river studied. Depending on the number of points used several methods exist [23] to deduce the resulting velocity profile. Then from this profile a mean value of the velocity is computed to find the discharge.

▷ **Mechanical current meter**

Principle

These current meters use the relationship between the angular speed of a rotating part, typically a propeller or an anemometer, and the streamflow velocity [24]. The two current meters just cited are represented on the Figure.

2.8. The current meter body is usually mounted on a wading rod. The velocity at a certain point of the river is then computed from the number of propeller revolutions over a certain amount of time. Since the current meter measures velocity at a single point the process needs to be repeated several times on the cross-section studied. The cross-section is divided in n verticals where velocity is computed. On each vertical three measurements are often recommended to provide a better representation of the velocity distribution [23].

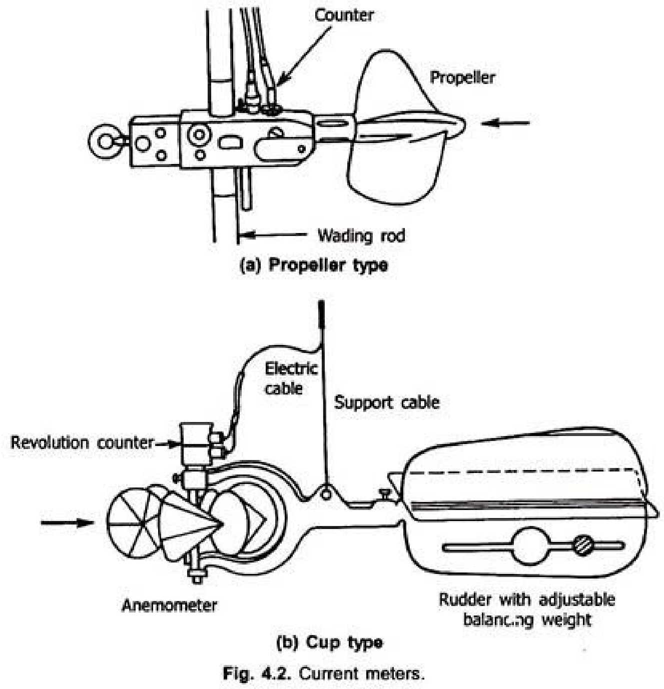


Figure 2.8: Illustration of two types of current meters. Figure from [25]

.

Advantages

They are low-cost devices easily deployed in river.

Drawbacks

Since it needs to be under water to work the durability is affected as well as risks of damage by some debris. This type of current meter has no sense of direction of the flow which may induce errors when backflows happen.

Scope of application

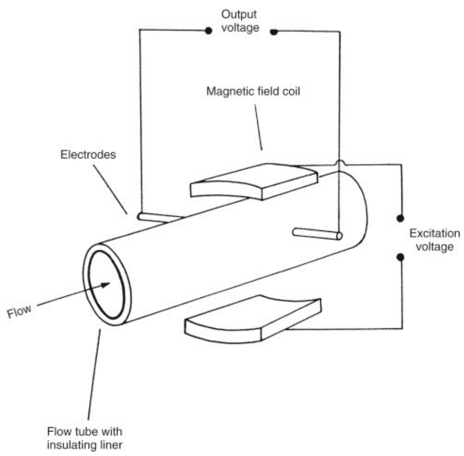
Still nowadays mechanical current meters are the most universally used in rivers, or in pipe flow.

▷ <u>Electromagnetic current meter</u>

Principle

The working principle is based on the Faraday's law of electromagnetic induction. This law states that a conductor (here water) when passing through a magnetic field produces a voltage linearly proportional to the water flow velocity. The Figure 2.9 illustrated this current meter.

(a) Electromagnetic current meter



(b) Schematic of electromagnetic current meter

Figure 2.9: Electromagnetic current meter representations. Figures from [26] and [27]

Advantages

One advantage over the mechanical current meters is that this one offers a direct reading of velocity, no need to compute the number of revolutions [28]. It also allows measurement of crossflow since it is directional. They can measure rapid changes in flow.

Drawbacks

They are not as reliable as the mechanical current meters in rivers and cannot operates near metallic object which limit their use.

Scope of application:

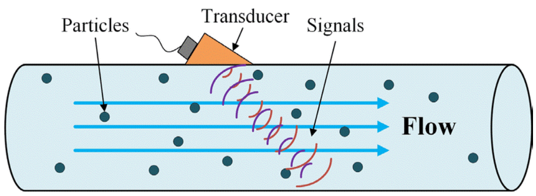
These meters are specially designed for conductive fluids like water, acids, caustic liquids, and slurries [29]. They are more suited for measuring flows through pipes because they are not in contact with the fluid allowing them to have a service life of 75 years.

▷ Doppler velocity sensor

Principle

A sound wave of known frequency is sent into the water, this latter is reflected by a moving particle in the water, either small sediments or air bubbles and the echo frequency is measured at the sensor. Then the Doppler principle is used to deduce the water velocity from this delta of frequency. The name of the device using this technique is the Acoustic Doppler current profilers (ADCP). Doppler instrument is represented on Figure 2.10.

(a) Doppler current meter



(b) Schematic of doppler current meter

Figure 2.10: Doppler current meter representations. Figures from [30] and [31]

Advantages

It can determine the velocity at different places along a vertical which helps to provide more detailed velocity profile. As the previous ones it can be mounted on a wading rod but also on a boat or into a small watercraft allowing to perform measurement along the cross-section more rapidly [32]. It can measure both velocity magnitude and directions.

Drawbacks

These sensors have a range of depth limited from 1.5m to 50m range of depth because of design considerations of acoustic frequency, size, processing technique employed, etc [33]. In case of very clear water, as in the tropics, the pings may not hit enough particles to produce reliable data.

Scope of application

ADCP is involved in surveys of sea, rivers, and canals. When it is anchored to the seafloor, it can measure the water velocity all the way up to the surface and when mounted horizontally on bridge pilings in river, it can measure the velocity profile from shore to shore.

2.2 Imagery based technique

Now that the most common non-imaging techniques have been seen, the imaging techniques used in previous Master Theses and in this work will be covered. Their principle will be explained along with the add-ons developed in the previous Master Theses.

2.2.1 Photogrammetry

Definition

According to Wolf and Dewitt [34], photogrammetry is "the art, science, and technology of obtaining reliable information about physical objects and the environment through processes of recording, measuring, and interpreting images." Another way of putting it is given by Luhmann [35], who said that "photogrammetry encompasses methods of image measurement and interpretation in order to derive the shape and location of an object from one or more photographs of that object."

The aim of the photogrammetry is to retrieve a three-dimensional reconstruction of an object from pictures. The results of a photogrammetric analysis may be in digital form (coordinates, geometric models) or in graphical form (images, plans, maps) [35]. It is typically use in topography studies.

Types of photogrammetry

Photogrammetry came up shortly after the photography itself. At that time only photochemical images were used. Over time, other types have emerged such as photoelectric images (digital photography) and more recently laser scanner images which have a distance information associated with every picture element [36]. In this work the photogrammetry will be performed using digital photography.

Apart from the images there are multiple ways to categorized photogrammetry [35]. Based on camera position and object distance, the number of measurement images, the method of recording and processing, etc.

Based on the number of measurement images, three categories can be distinguished:

- Single image
- Stereoscopic
- Multi-images

This work uses a multi-images technique called "Structure from Motion "(SfM). SfM has the advantage of being a low-cost photogrammetric method which allows high resolution topographic reconstruction [37]. It can produce digital elevation models (DEMs) that have an accuracy like lidar or better [38]. SfM technique was developed in the 1990s, in the computer vision community.

Principle of SfM

In all photogrammetry categories a three-dimensional structure is inferred from a series of overlapping, offset images. These images can be taken using multiple cameras or one moving camera. This construction of a 3D structure, referred to as triangulation [39] requires the complete knowledge of the cameras, their relative positions, and orientations as well as their settings (e.g focal length) during the recording of the images. The main difference between the SfM method and the others is the determination of these parameters, also known as internal and external camera parameters. SfM takes advantage of the multiple images taken at different point of view of the object to perform a self-calibration which finds the value of these parameters. The other techniques require the presence of ground control points (GCPs) in the images to compute it. Instead with SfM these are solved by automatic identification of matching features in multiple images. This is represented on the Figure 2.11. This approach requires a set of images with a high degree of overlap and a wide range of different camera positions. However, compared to the other techniques of photogrammetry which uses GCPs, SfM lacks the scale and orientation provided by them. To convert the relative 3D points coordinates given by the SfM technique to absolute real world coordinates a small number of GCPS with known coordinates can be used.

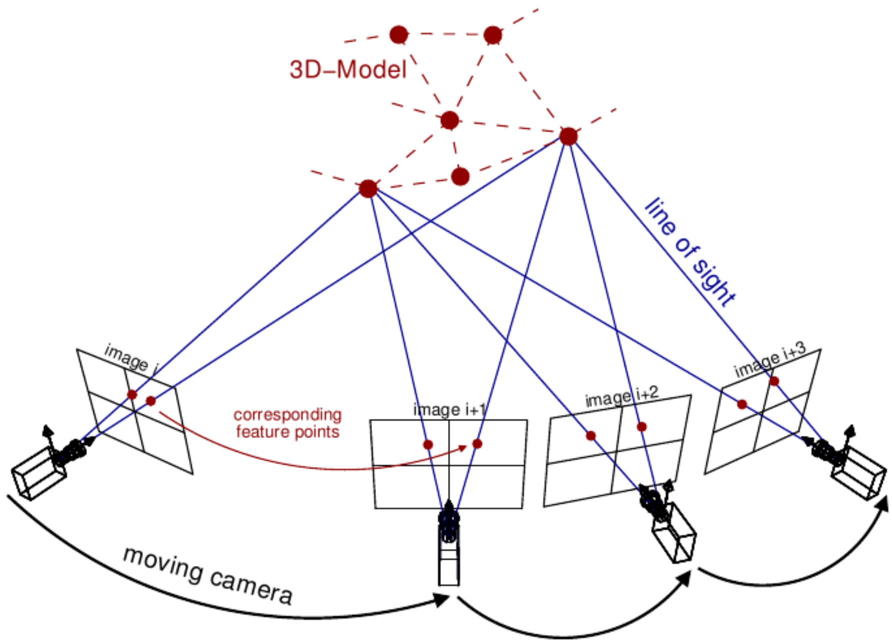


Figure 2.11: Structure from Motion (SfM) photogrammetric principle. Figure from [40]

.

Sofwtare choice and use

Software implementing photogrammetry are numerous. They can be distinguish based on the photogrammetry method they implement, ease of use, price, etc.

Agisoft PhotoScan, a commercial software package has been used in the previous Master Theses. It offers an implementation of the SfM method, where both image alignment and 3D model reconstruction are fully automated. Moreover, it provides a user-friendly graphical user interface.

An analysis using this software follows the following those mains stages [41]:

1. The first stage is to supply photographs of the desired object to the software. There is a possibility to apply a mask that will limited the region of interest (ROI) on the images which will be a time saver in the following steps
2. Then there is a photographs alignment step. SfM uses the supplied photographs and tracking algorithms to identify and match the features discovered on photographs and monitor their movements. Moreover, it finds the position of the camera for each picture and refines camera calibration parameters. The result is a sparse point cloud and a set of camera positions are formed.
3. The sparse point cloud is generated in an arbitrary coordinate system. Photoscan can transform the model into the absolute coordinate systems if a minimum three GCPs or camera coordinates has been recorded. The user needs to manually identify the GCPs within the imported photographs and provide the corresponding xyz coordinates. Then an optimization transformation is available to reduce the difference between the modeled point clouds and the supplied coordinates.
4. The next stage is generating dense point cloud. The software uses the estimated camera positions, pictures, and a multiple view stereo reconstruction to produce this dense point cloud.
5. After a surface is generated. It consists of a surface reconstruction using 3D polygonal mesh model to represent the object surface based on the dense point cloud. This mesh can be textured to better visual rendering.
6. The final stage is the creation of a digital elevation model (DEM). This represents the topography of the studied object. A color code is used to represent the difference in the altitude of each computed point. This is the file that will be exported and used for later computations.

For a more detailed explanation of the photogrammetry process or to see how the pictures are taken on field the Master Thesis of Hellebaut can be consulted [42].

The main disadvantage of this software is that it is a commercial one. This aspect has two main drawbacks. The first one is the cost it brings. The second is that the algorithms are not transparent. This kind of software acts as a black box where only input and output are visible. In case of errors or fails the reasons are difficult to establish due to this fact. That is why this year another Master Thesis was focused on the search and test of an open-source alternative.

2.2.2 Bathymetry measurement problem

Photogrammetry was initially developed to measure topography, however, in the presence of water the analysis is getting much harder to perform due to the change of medium air/water and the visibility. Some studies have been done on the subject [43] where a refraction correction has been applied and seems promising in clear water systems. However, for deep river or river where water surface is not transparent it is impossible to use images for computing the bathymetry.
Since it cannot be measured by images in general, in this application no part is dedicated to its computation, instead the bathymetry of the river is supposed to be known in advance. The user will give it to the application. The choice of the method used to compute the bathymetry is up to the user.

2.2.3 Water level measurement

Even if the bathymetry cannot directly be measured by the photogrammetry technique the latter can be useful for measuring the water level. Indeed, Hellebaut in his Master Thesis [42] has developed a method that can measure the water level using photogrammetry measurements. This method is presented below.

Hellebaut's method

It has already been established that measuring bathymetry using photogrammetry can only be done under specific conditions like transparent water and shallow river. So, photogrammetry cannot be used to measure the bathymetry in general. However, Hellebaut in his Master Thesis proposed a method including photogrammetry that can measure water level.

Principle

Firstly, he uses the photogrammetry based on images of one riverbank including a small portion of the river itself. This allows him to retrieve the topography of this part of the river. He knows that the topography measured is right until water appears. Indeed, measurements are biased by the reflection and refraction that occurred at the surface. Knowing the bathymetry of the river (including the surroundings of it) he can compare the topography computed with SfM method with it. Normally the two will be similar until water appears, so by computing the place where the two curves starts to differ the water level can be retrieved. This is illustrated on the Figure 2.12 where the errors caused by the water presence grows with the water depths.

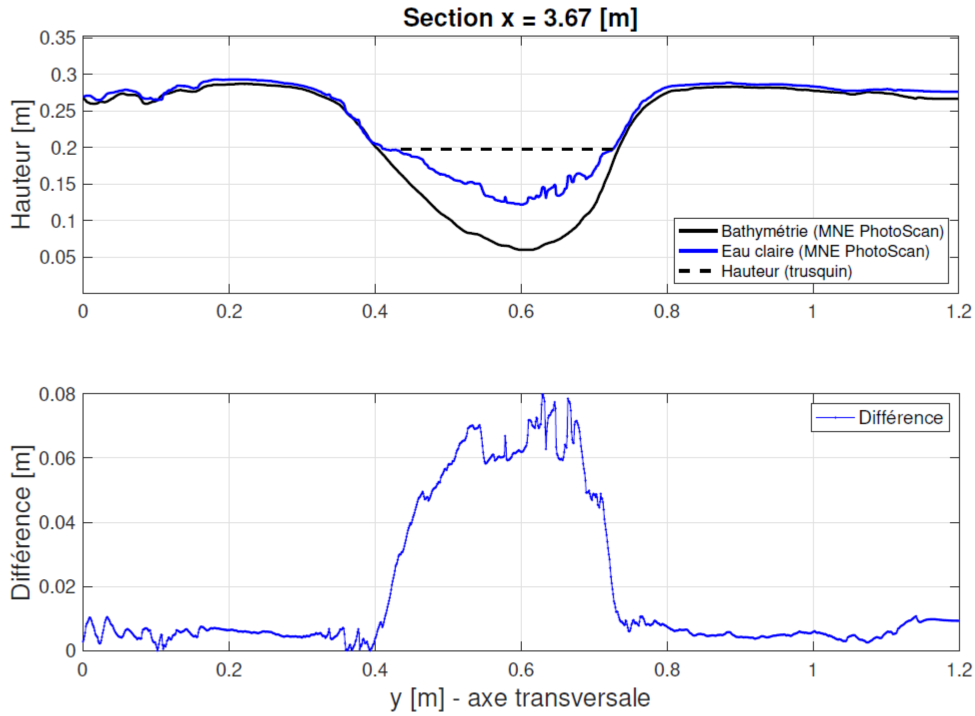


Figure 2.12: Measurements for water level on scale model. Figure from [42]
.

Hellebaut's algorithm has several parameters helping to detect the divergence point. The first one is named Count and represent a certain number of points while the second called Difference is a threshold representing a difference of heights in meters. From these parameters he infers two conditions. Firstly, the derivative of the curve of the difference needs to be positive (resp. negative) on Count consecutive points when the graph is covered from left to right (resp. right to left). The second condition is that the difference observed on theses Count points needs to be greater than the Difference parameter. Moreover, the graph of the difference is smoothed using a moving average to ease the computation. The parameters are fixed empirically.

Hellebaut has first tested its method in laboratory on a scale model of a river with transparent water where he modelized the river including the 2 shores using the SfM software. Then during field measurements on real rivers, he realized that only one riverbank can be modelized. Indeed, to model the river including the shores they must be present on the images which is difficult to achieve in practice especially for large river and small angle of view camera. Moreover, the SfM method struggle to find features to track on the water surface since all surface points can look similar. In addition, during the real experiment he noticed that the vegetation was problematic.

The vegetation problem made him add more parameters while the problem of the shores was resolved by focusing on one shore. An example of the curves he obtained on one shore are presented in Figure 5.1. He also noticed that when comparing the two curves, the method was working better when going from the center of the river to the shore. Indeed, the error made on the shoreline can mislead the algorithm. The implementation with the added parameters will not be described here since the idea is the same as before. For more details, the Master Thesis of Hellebaut can be accessed [42].

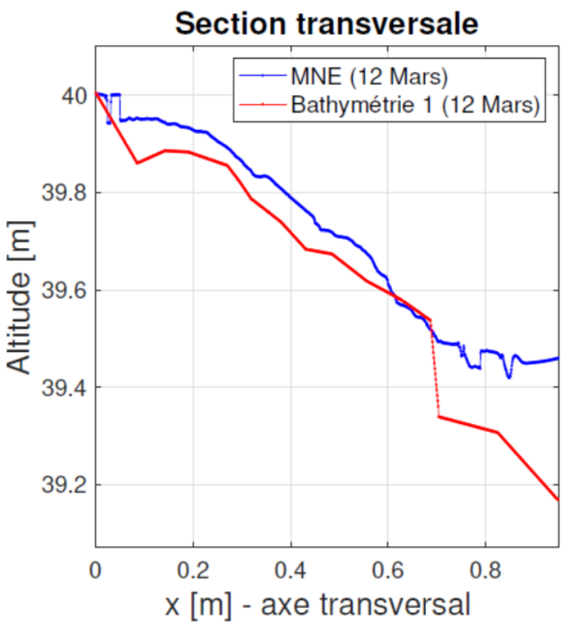


Figure 2.13: Measurements for water level on field. Figure from [42]

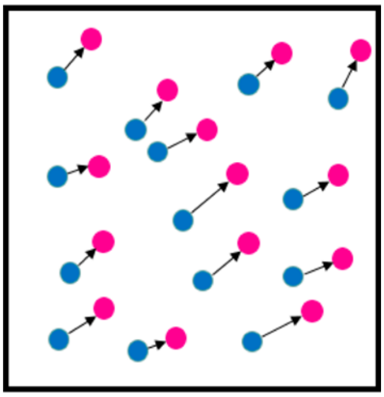
.

2.2.4 Velocity analysis

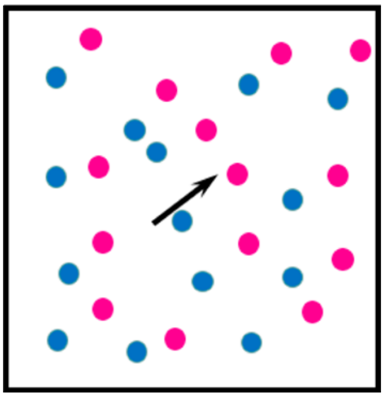
Particle Image Velocimetry (PIV)

The particle Image Velocimetry (PIV) is a non-intrusive optical technique which aims to compute a two-dimensional velocity field of a flow from consecutive images of it. The images are usually taken from a video of the flow. The basic tenet is to find and compute the displacement of tracer particles between two images. To derive the velocity field using PIV analysis a procedure must be followed.

A similar method is the Particle Tracking Velocimetry (PTV) where the displacement of each particles is tracked while in PIV analysis rather uses a statistical approach to find the most likely displacement of a group of particles. These two approaches are illustrated on the Figure 2.14.



(a) PTV measurement



(b) PIV measurement

Figure 2.14: Schematics of the PTV and PIV methods where the blue dots are the initial position and the red dots the final. Figures from [44]

Seeding of the flow

Seeding of the flow is the first step before any recording. Since the surface of the flow is usually homogeneous and lacking distinct patterns, the seeding step provides particles that can be track later by algorithms to follow adequately the fluid motion. Some studies have been made without seeding but only show positive results where the flow was turbulent producing patterns that can be tracked.

The size of the seeding particles has an impact on the results [45]. A small size is preferred to improve flow tracking while increasing the size improves light scattering and so a high signal to noise ratio. A compromise between the two must be made. Using particles of the same size is also recommended to avoid situations where smaller particles are considered as background noise in images. The spatial concentration of particles should be sufficiently high to capture details in flow field. The particles need to have a refractive index which is higher than the medium which is studied. Particles with spherical shapes are often used since the drag on that particles will be minimized. Regarding their density it must be as closed as possible to the fluid one to ensure buoyancy and the ability of particles to follow the flow as the fluid would do. The nature of the particles is also important since they must be chemically inert and not harmful to the environment. Of course, during a laboratory experiment or in river different types of particles would be allowed. The particles and their characteristics are determined by the experiment, the conditions under which it is carried out, etc. In river case wood shavings are often used. That is what has been used in the previous Master Theses as well.

The number of particles in the flow will influence the quality of the result. More particles result in better signal to noise ratio. If a method using one particle such as PTV gives an error E then one method using N particles as PIV reduces the error to E/sqrt(N) [46]. The concentration should not be too high otherwise the particles cannot be distinguished anymore [47]. Studies [48] have shown that considering groups of 10 particles leads to improvements but beyond that the reduction in the error becomes marginal.

Recording of the flow

Once the flow is seeded, images must be taken. The time between images is also important for the analysis to infer the velocity. So, a camera is the easiest way to do it since the time between two images can be easily inferred from the frames per second (fps) characteristic of it.

Filters

As said earlier the principle of PIV is to track the seeding particles. To do so they must be easily distinguished to the medium that carry them. Once the images of the flow are taken one way to enhance the difference between the particles and the medium is by the applications of filters. However, since luminosity conditions, medium and particles change from one experiment to another, different filters must be applied. Here are some problems that can arise, and which should be addressed by application of filters: light reflections, low contrast, image noise.

Interrogation ans search area

Once the images are recorded and filtered while it is necessary, the displacement analysis can take place. For this analysis, images are studied in pairs, frame A and B will be used to denote them. Before covering the process used to compute the displacement the concept of interrogation and search area must be covered.

In PIV, to compute the most likely displacement of a group of particles, a grid is applied on frame A, this grid can be seen on the Figure 2.15 in green dashed lines. Once the grid is defined a displacement vector will be computed for each nodes of this grid. Centered on these nodes an interrogation window also called interrogation area (IA) is defined. Each IA contains a group of particles which defines a "fingerprint" which can be found in both frames. That assumes that the displacement within the IA is uniform. This assumption limits the size chosen for IA since large ones will result in a loss of information about the flow. On the other hand, small IA have higher risks to not have any particles in it. An example of IA is represented on the Figure 2.15 in white solid line.

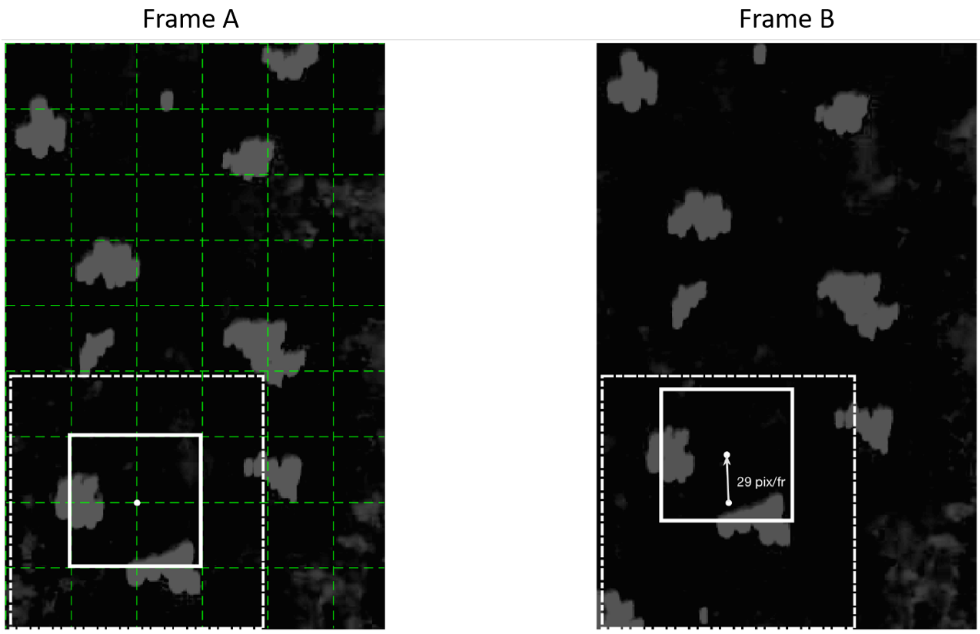


Figure 2.15: Example of displacement computation where the grid defining the center of the IA is represented in green dashed lines, the IA in solid white line, the SA in white dashed line. Figure from [49]

.

Once the fingerprint of the interrogation window of frame A is known the next step is to retrieve its position in frame B. Since the time elapsed between frame A and B is short the group of particles cannot have move very far from its original position. So instead of searching in the all frame B the new position of the fingerprint, a search area (SA) is defined around the original position of the IA, speeding up the process. This SA can be seen on the Figure 2.15 in while dash line. If the size of IA and SA are respectively MxM pixels and NxN pixels, then this assumes that the displacement between frames A and B will not be greater than (N-M)/2. Thus, the size of the IA and SA will vary depending of the flow, the delta of time between two frames as well as the particle concentration in the flow.

Direct cross-correlation (DCC)

The direct cross correlation (DCC) will be used to retrieve the new position of the IA of frame A in the SA of frame B. First a correlation matrix is computed between the IA of frame A and all possible new emplacements of this IA in the SA of frame B. The way the correlation matrix is built is represented on the Figure 2.16.

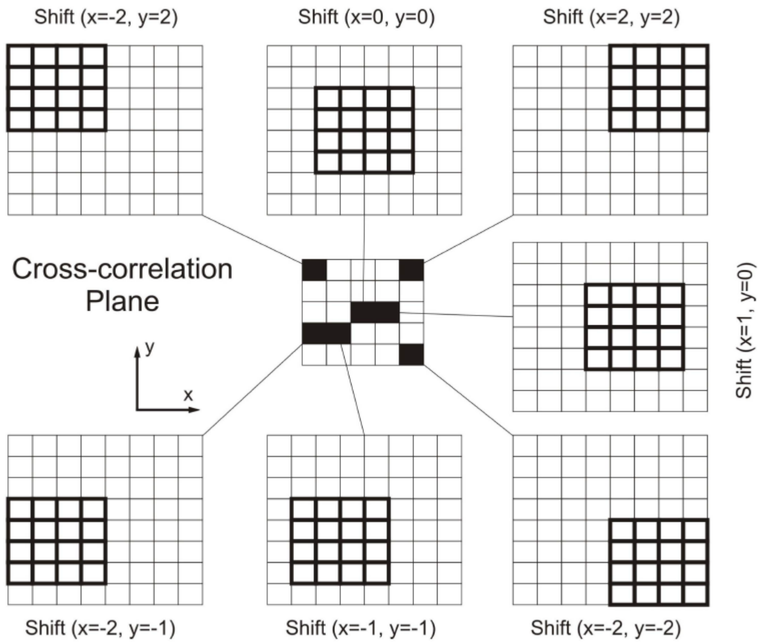


Figure 2.16: Cross-correlation plane from the cross-correlation function using a interrogation window 4x4 in size. Figure from [50]

.

In this matrix each value is the coefficient of correlation between the IA and a possible emplacement of this IA in the SA. The formula used is the classical formula of correlation coefficient but adapted to the case of matrices comparison. This correlation matrix will be square and with a size of N-M+1. This correlation map will present a peak at the most probable location where the IA of frame A is in the SA of frame B. Then the displacement is simply found by taking the distance between the center of the IA and the peak of correlation which corresponds at the maximum of the correlation matrix. This is illustrated on the Figure 2.17. It will give a displacement in pixel that can be convert in meter using a simple calibration where an object of known size is on the images. Finally, the velocity is found by dividing the displacement by the delta of time between the 2 frames.

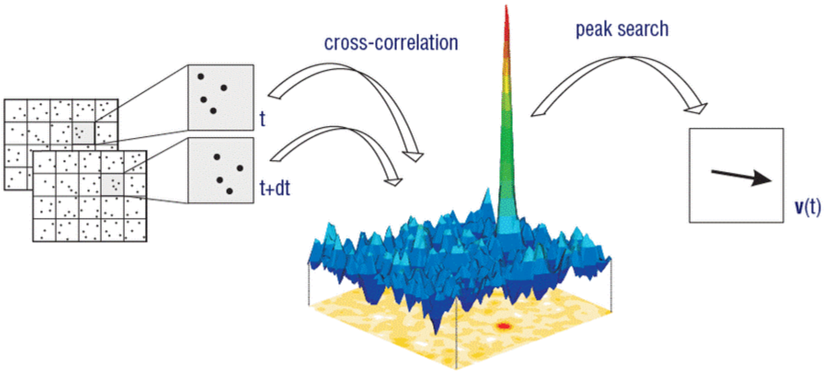


Figure 2.17: Cross-correlation technique typically used in PIV. Figure from [51]

.

Fast Fourier Transform (FFT)

The major issue with the DCC is the time it takes to process. Indeed, to compute the correlation matrix, $(N - M + 1)^2$ correlation coefficient are calculated and this must be repeated for each of the MxM IA. The computational time is lying between $\mathcal{O}(M^2)$ and $\mathcal{O}(M^4)$ [52]. Keep in mind that this process must be repeated for each pair of frames. One way to improve it is to compute the correlation matrix in the frequency domain using the Fast Fourier Transform (FFT). This takes advantage of the correlation theorem [53] which states that the cross-correlation of two functions is equivalent to a complex conjugate multiplication of their Fourier transforms. This approach uses interrogations areas of identical size which induce more noise in the correlation matrix that can be reduce using multi-pass as it will be seen later. The process is represented in the Figure 2.18. Since only one area is used for the search, the displacement cannot be greater than M/2 in general. So generally, a bigger IA is preferred since it is recommended that the displacement stays lower than one quarter of the IA size [54]. Performing the correlation matrix computation using FFT allows to have a computational time of $\mathcal{O}(2M^2 log_2 M)$ [52].

Multi-pass PIV

The issue of the FFT method is no longer the computation time but the spatial resolution. Indeed, since there is no longer a SA, a larger IA is needed to avoid loss particles between the two frames. However, choosing larger IA results in a

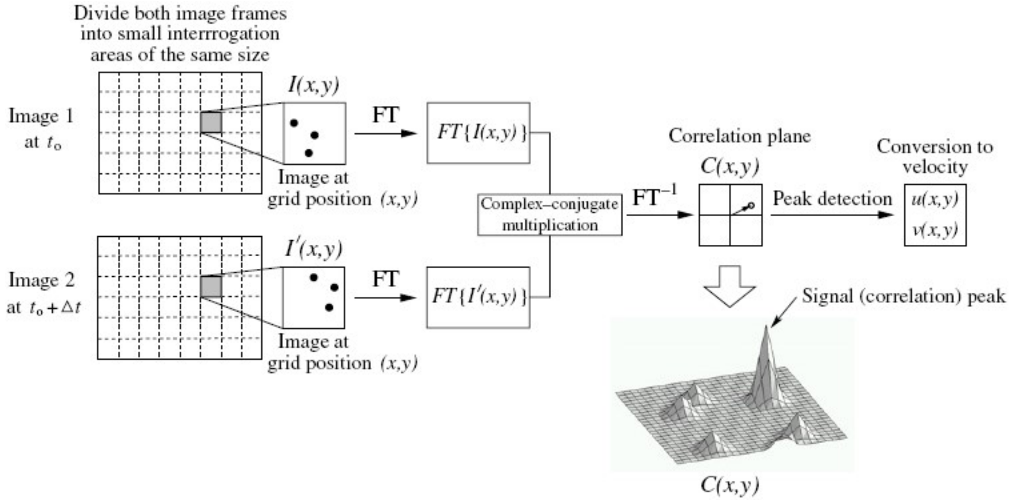


Figure 2.18: Cross correlation algorithm using FFT. Figure from [55]
.

loss of information about the flow. To counteract this a multi-pass analysis is recommended where the size of the IA is progressively reduced using the previous result to adjust the position of the IA in the new pass. The idea is to start with a large IA in which the new emplacement of the particles is likely to be found but since it means losing information about the flow it is not accurate. Thus using this position of the particles, a smaller IA can be defined and shift such as it englobes the new emplacement and be more accurate. The process can be repeated if wanted. The number of passes is a compromise between improved accuracy and computational time. A typical compromise setting is three passes starting with a 64x64 then 32x32 and finally a 16x16 pixels IA [56].

Subpixel interpolation

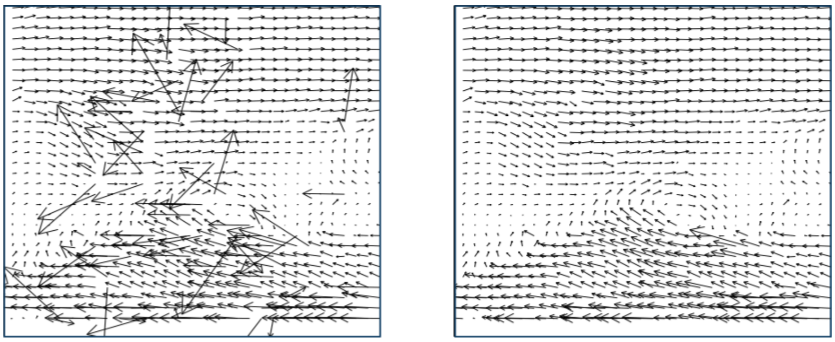
The resolution obtained using the peak of correlation is a pixel accuracy. Sometimes when the ratio of pixels per meter is too low a subpixel precision may be useful. One way to achieve it is by using three-point estimator [57].

Delta of time choice

The delta of time needs to be such that the displacement between two frames is sufficient but not too high regarding the IA size [58].

Vectors validation and interpolation

The velocity field obtained at the end of the process may contain spurious or bad vectors. On the Figure 2.19 an example of the vectors with or without a validation step. Some errors are easy to detect as they differ drastically in magnitude and/or direction from neighboring vectors. However even if they are easily detected by humans it is harder to design computer algorithms to do it. These errors may be due to correlations occurring between unmatched particle pairs, non-uniform particle distribution or variation in image intensity [59]. In case of large displacement vector some particles may be lost in the area studied reducing the particle concentration which introduces a bias to select smaller displacement. Another source of error is the presence of high velocity gradients across the IA. These latter spread the peak in the correlation plane over a larger area reducing its height and increasing the selection of a random correlation peak.



(a) Vectors before validation step (b) Vectors after validation step

Figure 2.19: Importance of a validation step to avoid spurious vectors. Figures from [46]

.

To improve the results a common strategy is to spot the spurious vectors, get rid of them and replace their value using interpolation. To detect which vectors need to be removed the following techniques are often used:

- A threshold can be applied on the velocity vector components. This threshold can be computed in several ways. One method is to plot velocities values with the velocity components as axes. The major part of the values will lie at one region of the graph. The outliers are the points far away from this region and can be discarded easily for example by selecting the region of high concentration with a rectangle. The Figure 2.20 represents this selection. This is a visual thresholding. Another thresholding is a statistical one where each velocity vector is compared to the global mean. First the mean μ and standard deviation σ are computed. Then vectors that are outside the interval $[\mu - \sigma.n, \mu + \sigma.n]$, where n can be chosen, are discarded. This method does not work when the flow follows an irregular velocity field.

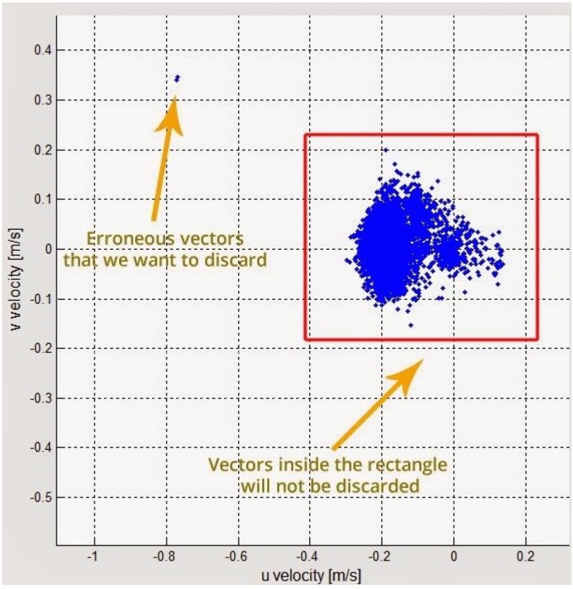


Figure 2.20: One vector validation method. Figure from [60]

.

- The validation can also be made using only the neighboring vectors. Here, a vector is considered valid is it close to the amplitude of its neighbors. This method assumes that the flows is relatively homogeneous. However, it fails when a lot of wrong vectors are grouped.

- Another way to decide which vectors are reliable is to use the signal to noise ratio (SNR). The validation is not made on the vectors itself but using the correlation matrix of the PIV analysis. There are multiples ways to measure it, one is to take the ratio between the highest correlation peak to the second tallest peak.This is illustrated on the Figure 2.21. Another measure is to take the ratio with the highest correlation peak to the mean value of the correlation matrix. Using the SNR as a measure of the quality of the results is more reliable than using the vectors directly. Indeed, the correlation plane reflects the combined effects of various sources of error influencing the particles displacement [61].

Once spurious vectors are removed, to keep a certain continuity in the field an interpolation is performed. A 2D spline interpolation has been proved to be the

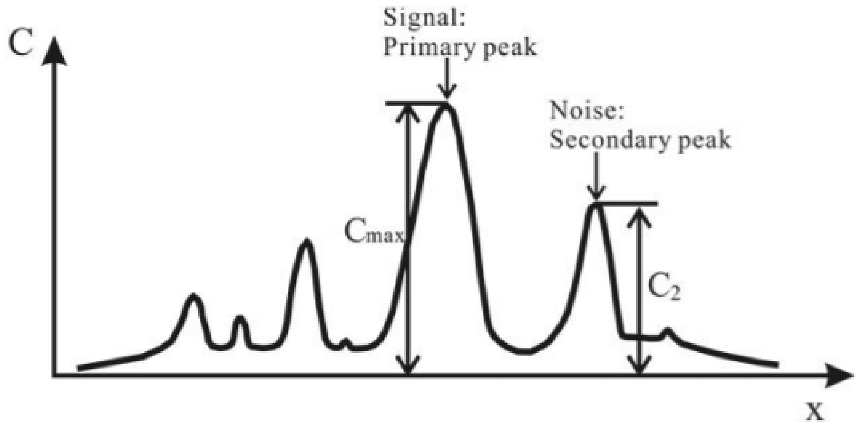


Figure 2.21: Illustration of the primary and second peak of the correlation, often used for the signal to noise ratio value. Figure from [61]

.

best interpolation method when the missing vector are below 5% [56]. In general cases, it is the boundary value solver, which solves partial differential equations on the velocity field, that shows the best overall accuracy.

Means of velocity vector field

To get accurate results the PIV analysis is not made on two frames only but on many pairs. Then the average velocity field is computed considering all the results obtained. This is an important step because seeding is not always homogeneous and so averaging allows to stabilize and correct the results.

Large Scale Particle Image Velocimetry (LSPIV)

The PIV method has been used in laboratory for more than 30 years [62]. The LSPIV on the other hand has emerged recently. It has been developed to adapt PIV to field measurement such as natural rivers, channels, or large physical models. The principle is essentially the same, the main difference between the LSPIV and PIV analysis are the scales of the surface flow of interest. This adaptation brings some challenges such as the variable illumination and the limited accessibility to orthogonal camera views [63]. Moreover, since the camera will have to be at a higher height to capture the entire flow the resolution of the camera should be high enough to capture the seeding particles. The time step between two frames should also be raised since the displacement must be sufficient to give a reliable speed.

The seeding is more difficult to do specially to get a homogeneous repartition when dealing with large flow. To get reference distances some ground control points (GRPs) are placed on the same plane of the flow.

Bia and Somers have developed in their Master Thesis some enhancement of the implementation of the PIV method that they have been using. They remark that using the mean of correlation matrix instead of velocity vector can result in must reliable results especially when particle concentration is low [49]. They also used the signal to noise ratio (SNR) as a weight when doing the mean of the correlation matrix, giving more weight to the most reliable correlation matrix.

Chapter 3

Choice of tools and Integrated Development Environment (IDE)

Originally, it was intended to develop the application using MATLAB. The principal reason was that the others Master Theses works have been done using MATLAB. The latter is great for matrix and array mathematics, but the development of application is limited in terms of design compared to others exiting tools.

During the early stages of the project, the decision was taken - in agreement with the supervisor of this Master Thesis - to use Python instead. The main reason for this change is that MATLAB imposes design constraints (e.g., application very slow to save and run, tedious to customize widgets) that were limiting in the frame of this work and the desire to use a more generally spread environment. Since Python is only a programming language an Integrated Development Environment (IDE), libraries and tools must be chosen to develop the application. The IDE chosen is Spyder and the tool to develop the application is the cross-platform python framework Kivy. The comparison between MATLAB and python as well as the choice of Spyder and Kivy will be covered in the following sections. The library OpenCV will also be mentioned as a substitute to some toolboxes of MATLAB.

3.1 Programming language

3.1.1 MATLAB

The MATLAB software used in the previous Master Theses is a commercial software developed by MathWorks that combines a desktop environment with a programming language that expresses matrix and array mathematics directly [64]. Previous Master Theses have used it to perform the velocity analysis and write functions implementing the methods they have developed.

This software was used because it was taught in the bachelor of the faculty and used in many courses. Students needed to purchase a license, but it was affordable thanks to the faculty. This software came out in 1984 [65] and has developed a lot since. Nowadays it offers many toolboxes in different fields like Artificial Intelligence, Image Processing, Control Systems, etc. It is used by many universities worldwide. The reasons for this success are the simplicity of MATALB as well as the active community and easy access to documentation.

3.1.2 Python

Python is object oriented, interpreted and a high-level programming language [66]. Python's code is easy to read and understand thanks to its syntax. It is free, has a strong library and is supported by numerous operating systems such as Android, iOS, Windows, GNU/Linux. The community has been growing tremendously these last year's making it one of the most used language worldwide in both industrial and academic fields.

Nowadays the faculty teach Python instead of MATLAB, one reason being that Python is free and more used. Considering the growing interest for Python and that new students will learn it the choice has been made to use Python for the programming part of this work.

This change of programming language requires rewriting the code developed in the previous Master Theses. Since some functions and toolboxes provided by MATLAB have not their counterpart in Python the rewriting may differ from the original functions. These differences will be highlighted when the methods will be covered.

3.2 Choice of the python IDE, libraries and framework

Spyder has been chosen as IDE. 'Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts' [67]. It looks like the IDE of MATLAB. Indeed, it has an editor, an interactive console, a variable explorer, a debugger, and a help feature to consult the documentation. The easiest way to download it is as part of the Anaconda distribution and use the conda package and environment manager to keep it up to date. The transition from MATLAB IDE to Spyder is made easier due to their similarity.

MATLAB toolboxes are compensated in Python with the numerous libraries available. A library that will be often used in this work is the OpenCV one. OpenCV is a well-known library mainly aimed for computer vision. It is written in C++ but there are bindings with python. It provides many functions related to image processing that will be used in this work. Another library that has been helpful is the NumPy one. NumPy is a powerful scientific and numerical computation library providing Python with a similar computational power as found in C or Fortran.

The Kivy framework has been chosen for rapid user interface and graphical application development. "Kivy is an open-source Python library for rapid development of application that make use of innovative user interfaces, such as multi-touch apps" [68]. Kivy is quickly installed being a Python library. It runs on Linux, Windows, OS X, Android, and Raspberry Pi with the advantage that the same code can be run on all the supported platforms. This allows the developer to build an app and use it across all devices easily.

Kivy provides a wide range of widgets. It also has a specific language called kV language that can be used to easily design custom widgets. This language is simple to learn thanks to its clear syntax. The graphics engine is built over OpenGL ES 2 a modern and fast graphics pipeline.

Kivy was first released in 2011. So, the community is not particularly large for the moment, but it has a well-documented API. A drawback of Kivy might be the kV language which needs to be mixed with python code to access complex applications. However, it is easy to learn it and powerful to custom both widgets appearance and behavior. Kivy provides some useful features as the automatic widget resizing based on the window's size.

Compared to MATLAB, Kivy provides widgets whose behavior and appearance can be more easily controlled and customized.

Chapter 4

Existing application and tools

Before developing the application, some researches have been done to find existing applications or tools that also try to simplify the computation of discharge from images. The idea was to draw some inspirations on how they implement it to ease the development of the present application. Two mains sources of inspiration will be covered in the following, which are the Discharge application and the RIVeR toolbox.

4.1 Discharge Application

Discharge is an application for smartphone which can be download on Google Play Store. It is developed by the company Photrack AG that was founded in January 2012 by Dr. Beat Lüthi [69]. On the description of the application they claim that in less than 2 minutes you can obtain measurement results of the water level, the average flow velocity, and the discharge of the channel or river studied [70]. The results can be stored in a database which can be managed using their web platform.

4.1.1 Overview

An overview of the application will be given and a few steps of the usage scenario will be detailed further. For a more detailed overview their manual user can be consulted [71].

Set-up of the site

The first step is to create a new site. After the creation of a site the application can store it so that the next time the user can skip this step. During the creation of a site the user needs to give a name and a location for it. The location can be automatically filled if the location service is active on the smartphone. Then the user must choose the cross-section profile of the river. Four choices are presented: a rectangular shape, trapezoidal, parabolic, or free. The latter must be done on the web platform. Once the user has chosen the shape, he must provide the different distances/points that can defined it. For example, the rectangular shape requires the width and depth values. The trapezoidal and parabolic shapes follow the same principle. For the free shape case a file must be loaded on the web platform. This file must contains coma-separated values of the cross-section points and following the convention (y,y,y,z,z,z). This means that all y values of the points then all z values of the points must be given. The y and z values are the coordinates following the reference shown on the Figure 4.1.

Once the cross-section is defined the user must specify a roughness coefficient value corresponding to the Manning-Strickler coefficient [72]. To help the user to provide a suitable value, a table containing typical values for different cases is provided by the application. The application requires this coefficient to derive a roughness length, used according to the approach developed by Absi in his paper [73]. The Discharge application uses the same approach and computes the velocity profile using a mixing length model [74].

Finally, four markers (same as GCPs) must be placed, two markers on each bank. Once placed the user must provide the distances between the markers on the same shore as well as the diagonals lengths. The distances from each marker to shore must also be provided. The Figure 4.1 illustrated the set-up and the order of the markers. Before going to the recording part, he must also provide the resolution of the camera used 720p or 1080p.

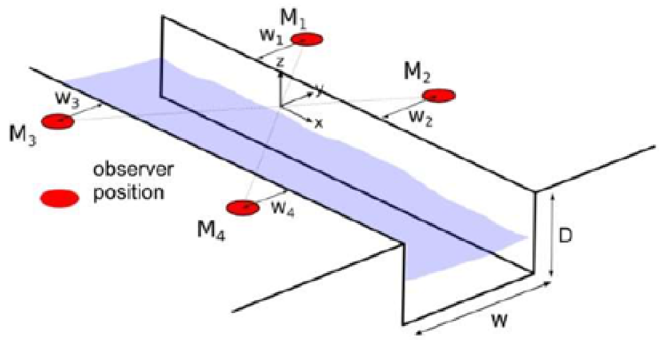


Figure 4.1: Markers position and order. Figure from [71]

Measuring process

The next step is to record the flow with the smartphone's camera. The user must ensure that the four markers are visible as well as the two shores. Moreover, the waterline on the opposite side must be seen at an angle that is between 20-70 degrees (measured from the horizontal) [71]. To have a quick idea of what that implies, if the smartphone is held at 1.5 meters just above the water surface on the shore, using this range of angles limits the width of the river to be between 0.55 and 4.12 meters. Then by pressing the camera icon the recording starts and lasts 5 seconds. The user is assumed to hold the camera steady during this process. After the recording, the user must indicate the location of the four markers on the image. Then a blue line will be drawn on the river and the user must move it to the shore. Pinching allows to zoom to make it easier and more accurate. This procedure allows to compute the water level.

Water level

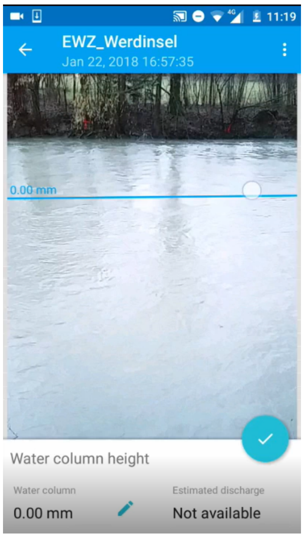
The way those input data - particularly the blue line positionning - are processed to compute the water level remains unclear. That is why a contact with the developing team of the application has been made where more details about this method was asked. In their response they explained that the level 0 is defined at the bottom of the channel or river and represented by the blue line on the Figure 4.2. Then as the user moves it up the value shown is the water column height from the deepest point. Once the user has dragged the line to the shore, the pixels coordinates of the line is computed. Then by looking to the intersection of those coordinates with the cross-section and transforming that to metric units they can retrieve the water level. They computation is possible since the camera is calibrated using the markers and the cross-section is known. In Figure 4.2 a schematic is also represented and can help to understand the principle where the top part is a cross-section view and the bottom one an eye-bird view.

Velocity Analysis

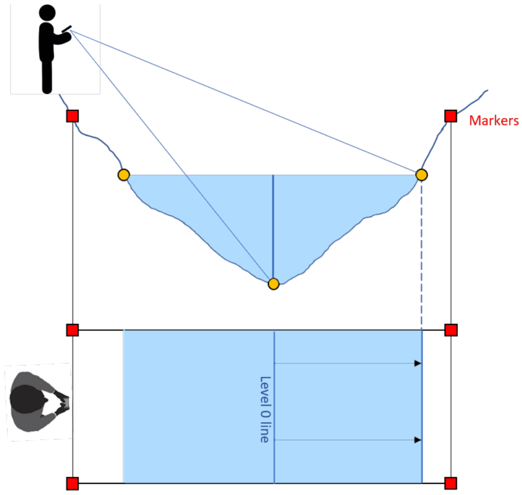
After the computation of the water level the users has three possibilities:

1. Delete the measurements.
2. Proceed to velocity and discharge computation.
3. Directly retrieve the discharge from a rating curve.

This last option assumes that a rating curve for the site already existed. Most likely the user will choose to process the video for the velocity analysis. In that



(a) Water level screen in Discharge. Figure from [71]



(b) Principle of the water level measurement in Discharge using cross and top schematic views

Figure 4.2: Illustrations of the principle of the water level measurement in Discharge application.

case, the velocity vectors are rendered as well as the calculated discharge. The velocity analysis is performed using a Surface Structure Imaging Velocimetry (SSIV) technique [74] which is essentially the same technique as an LSPIV analysis but without seeding particles. This requires that enough visible structures must be present on the water surface.

4.1.2 Evaluation of Discharge

In the paper of Carrel [74] an evaluation of the Discharge application is conducted. Their experiments are limited to a laboratory flume under controlled conditions with flow rates of $20 - 120\,\frac{L}{s}$. The set-up is represented on the Figure 4.3. The measurements were made using a smartphone, either hand-held or placed on a tripod, to evaluate whether the use of a tripod provides a significant accuracy gain. Their conclusion was that in the hand-held case 87% of the data show an accuracy of $\pm 15\%$ whereas for the tripod case 100% lies within the same accuracy range. They planned to test it under larger discharges rates in the order of $1 - 10\,\frac{m^3}{s}$.

In the Master Thesis of Asselberghs and Peeters [75], they have tested the Discharge application on different river sites. Their feedback was that it is more

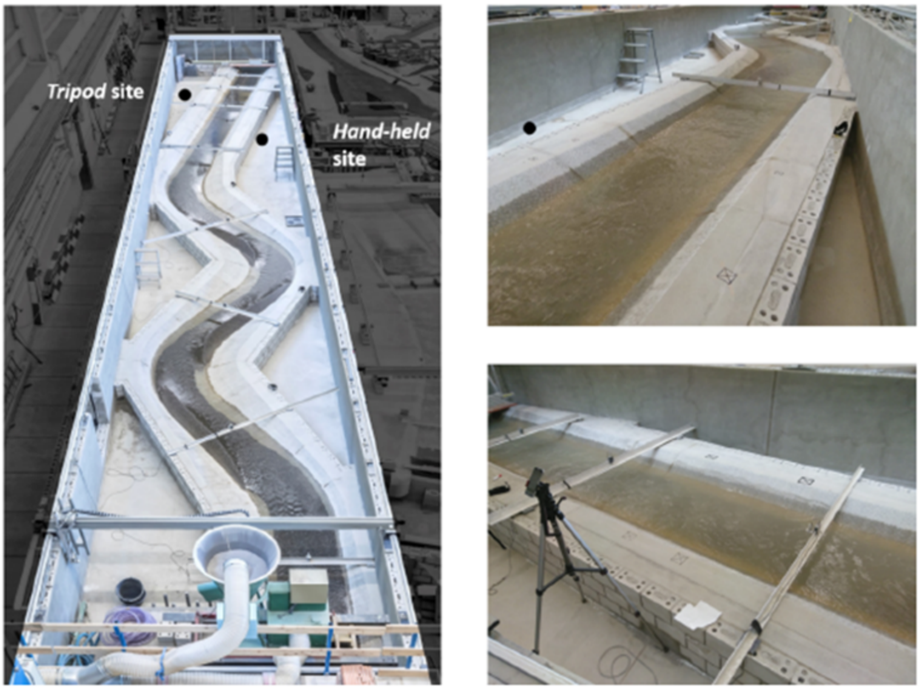


Figure 4.3: Flume use in the experiment of Carrel to evaluate Discharge application. Figure from [74]

difficult to handle than expected. Indeed, on four sites the analysis was only possible one time over three attempts on each three sites and was impossible to conduct on one site, maybe because of the viewpoint being too high. From these measurements they managed to obtain one estimation of the discharge on three sites. They compared these values to a reference one and the result of a LSPIV analysis. While the results of the LSPIV analysis were consistent the Discharge ones varied a lot. On one site the error was 50.2% while on another 1.8%. The main reason is that Discharge is more sensible to the conditions in which the video is taken.

Advantages

The advantages are the simplicity and the low-cost characteristic of the Discharge application making it a useful tool for fast measuring of discharges. The simplicity of operation and the availability of smartphones make it accessible to the untrained citizens as well as experts.

Drawbacks

Discharge has several drawbacks. The water surface must have enough pattern (e.g. turbulent patches, waves, leaves, bubbles) to perform the velocity analysis since no seedings particles are used. The accessibility of the site and the viewpoint constraints are limiting its use. As specified in the user manual, the measurement position must not be far away from the river, the water surface should expand over at least 1/3 of the image height [71]. The four markers should also be approximately in a horizontal plan which may not be always possible. This is the same assumption made in the orthorectification step in the application.

Inspirations

In the application developed in this work the step where the user must enter the bathymetry data has been inspired by the Discharge application. Indeed, in the application the user will also have the choices between typical bathymetry curves and having the possibility to enter free-form curve coordinates.

4.2 RIVeR software

During the recording of a flow, the captured images can be oblique for example when they are taken from the shore. This induces a deformation in the images which impacts the results of the LSPIV analysis. Most software performing image-based velocity analysis do not include tools allowing to rectify this deformation for example PIVlab. That is why RIVeR has been developed as an add-on to software like PIVlab for producing an efficient large-scale analysis system. RIVeR is also a standalone application. It has been developed at the National University of Cordoba Argentina, in 2013. This is a free application built in MATLAB.

4.2.1 Overview

RIVeR requires the use of an image processing software such as PIVlab or PTVlab to perform the velocity analysis. The image extraction and all the other steps leading to the discharge can be done in RIVeR. To better understand how the 2 parts are combined, the flowchart of the overall process of Large-Scale analysis is shown on the Figure 4.4. The blue steps are the ones done with RIVeR while the black one is performed using an image processing software. The main steps are described below.

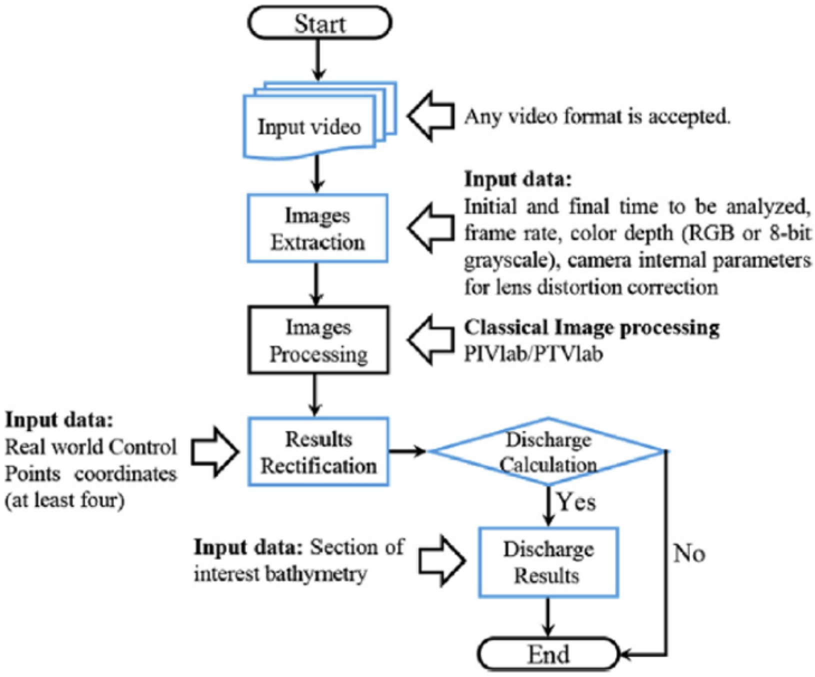


Figure 4.4: RIVeR overall process. Figure from [63]

Image extraction

The user must first load the video of the flow. An advantage is that most existing video formats are accepted. Then several options are proposed before extracting the images. The user can choose a step between the frames if he wants to skip some. The default value is one meaning all frames are chosen. Another parameter is the range in seconds of the video that will be extracted. Finally, the user may select the option that converts the image in grayscale and/or the option that corrects the lens distortions. If the latter is selected it will propose some camera models to the user. Then using the intrinsic parameters of the camera model chosen, the lens distortions will be corrected.

Results rectification

This step will not be described here. Indeed, the orthorectification step of RIVeR has inspired the one implemented in the application developed in this work. The principle is similar than the one covered in the image processing chapter 7. To have more information, the Patalano article can be consulted [63].

Discharge calculation

The mean velocity used is the one computed with the image processing software. The user selects the cross-section where he wants the discharge evaluation. The corresponding velocity profile is extracted from the velocity field and shown on the Figure 4.5. The user must provide an Excel file which contains the bathymetry of the section. The water level must also be included as an additional input. Then the cross-section is divided by N adjacent verticals at an equal distance d. The discharge is computed using the following formula:

$$Q_s = \sum_{i=1}^{N-1} d \left(h_i + \frac{h_{i+1}}{2}\right)\left(V_{s_i} + \frac{V_{s_i+1}}{2}\right)$$

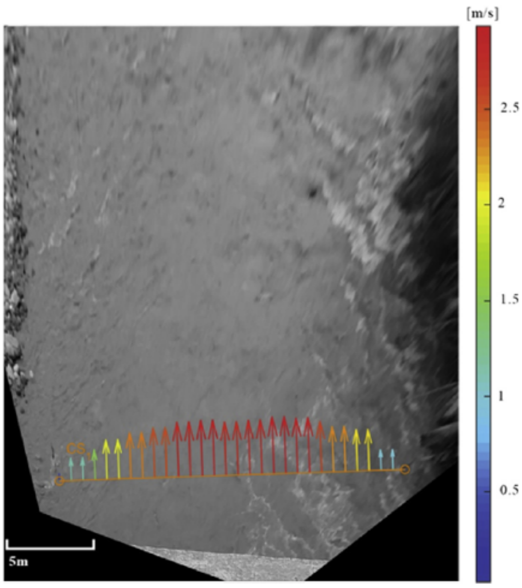


Figure 4.5: Surface velocity retrieved from a selected cross-section. Figure from [63]

Where hi and h_{i+1} denotes the depths of two adjacent verticals and V_{s_i} and $V_{s_{i+1}}$ the normal to transect surface velocity component over two verticals. To correspond to the real discharge, it must be multiplied by the velocity index. A value of $\alpha = 0.85$ is set by default but the user can choose a more suitable one for its data if he wants.

$$Q = \alpha\, Q_s$$

RIVeR offers more features, some of them are presented here [76] along with examples.

Inspirations

The image extraction step of RIVeR has inspired the one implemented in the application developed in this work for example the range of time or the step taken between the frames. The main inspiration is the orthorectification part explained in the article of Patalano [63] which has been implemented in the application.

Part II

Content of the new developed application

Chapter 5

Work on water level methods

In the application three methods are implemented to retrieve the water level:

- One adapted from Hellebaut's method detecting the divergence point using a minima-based method.
- Two based on a mapping of the measured river width on the bathymetric curve.

In the first method the user is offered the possibility to correct the imprecision of the results by manually adjusting the position of the divergence points. The functioning of these three methods is covered in the following sections.

5.1 Adaptation of Hellebaut's method

5.1.1 Problem with the Hellebaut's method

In the State-of-the-art chapter, the Hellebaut's method for measuring the water level was introduced. This method was first tested in laboratory experiments and requires the settings of two parameters (Count and Difference) to work. Then, this method was tested on real life measurements where it failed because of the presence of vegetation on the river's banks which misled the computation. Thus Hellebaut had added more parameters to circumvent this issue. Along with the new parameters the user must also take a guess on where the divergence point might occur and provide a window size around this point to limit the search.

Hellebaut's method requires a lot of parameters to solve a problem whose resolution is visually easy. To illustrate this, look at the Figure 5.1 where two curves, the digital elevation model (DEM, MNE in french) and the bathymetry, are shown. On this figure you can easily spot the point where these curves diverge.

Thus, instead of an algorithm with numerous parameters, a simpler algorithm will be used in this application with the possibility to adjust the result in case of failure. The idea is to propose a first guess to the user, if this guess seems right the user goes to the next step if not, he has the possibility to adjust the solution by clicking on the graph where he thinks the solution is.

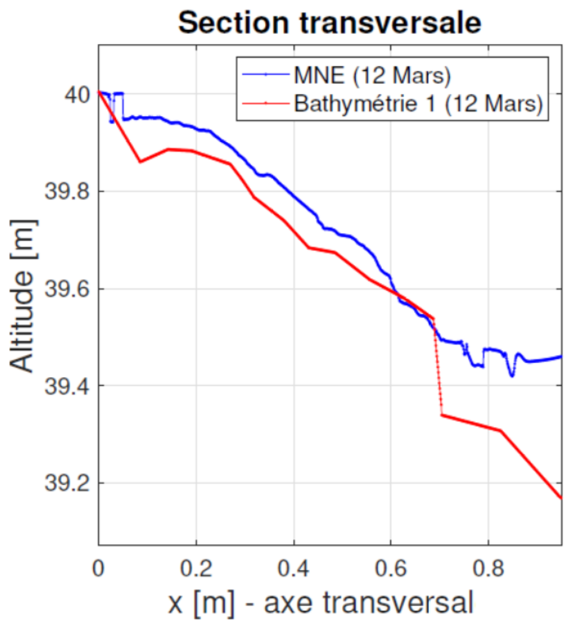


Figure 5.1: Measurements for water level on field. Figure from [42]

.

5.1.2 Minima based-algorithm

Principle

The first step is to take the absolute value of the difference between the two curves of the Figure 5.1: $D(x) = |\text{MNE(x)} - \text{Bath(x)}|$. In the example displayed, the divergence point is where the two curves intersect which corresponds to a minimum on the D(x) curve. Therefore, working on the minimum seems a good idea. More generally an observation is that the DEM curve presents many little oscillations. This will result in the presence of minima spread all over the D(x) curve. The algorithm will have to select the right minimum. Another aspect is that the algorithm must be kept simple.

The fact that the curves are closer before the divergence point can be due the absence of vegetation near the water or maybe the fact that the depth of water is low near the shore leading to similar results.

The algorithm that has been implemented is the following:

1. All the minima on the D(x) curve are found using the function `signal.argrelextrema` of the SciPy Python library. The minima are the points whose direct neighbors have higher values.

2. Their average value is computed.

3. All the minima are covered starting from the river to the bank and the first minimum below the average value is returned.

In the application the points of the D(x) curve are assessed to determine which bank is shown and change the direction in which the minima are covered accordingly.

This algorithm relies on the following observations. The part of the curves corresponding to the bank are similar. They are not identical because the Structure from Motion (SfM) photogrammetry technique is not 100% accurate. Thus, the minima present in this part should have a low value on the D(x) curve. On the other hand the photogrammetry is lost on the river part (see chapter 2) resulting in greater error. This means that the minima on this part will have a higher value on the D(x) curve. To find the divergence point as suggested by Hellebaut [42] the two curves only need to be compared using a small portion of the river's cross-section. This means that the bank part is larger than the river part. The minima are also more numerous in the bank part where their values are low. Thus, taking the average value of all minima should result in a low value threshold. This threshold is then used to find the first "low" minima starting from the river to the bank.

This should give the divergence point if the data respect the following assumptions:

- The curves are not smooth such that their absolute difference present a sufficient amount of minima spread all over the x range.

- The minima present in the river part have higher values than those on the bank part and their proportion should be such that the threshold computed is below all minima of the river part.

On the example used here the resulting D(x) curve as well as the minima computed are displayed on the Figure 5.2. In this case since the assumptions are respected the algorithm works.

The goal is not to produce a robust method that will cover all the configurations but just a simple method allowing a first guess. This approach can be done thanks to the correction step that will be explained now.

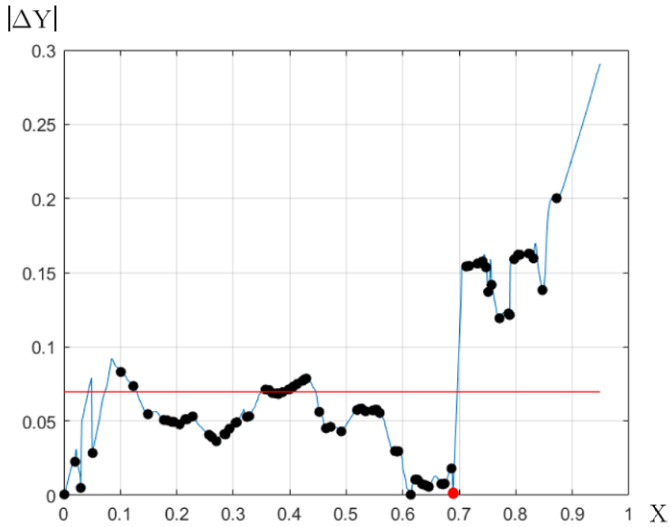


Figure 5.2: Difference curve on which the principle of the minima algorithm is shown. The black points represent the minima, the red line their average value and the red point the minimum selected by the algorithm.

Correction step

Once the guess of where the divergence point is computed, the application will display it on the graph along with the two curves of bathymetry as you can see on the Figure 5.3. Then depending on whether the user find this guess correct or not, he can either go to the next step or click on the graph where he thinks the divergence point is. The application will then store this information before going to the next step.

5.2 Alternative methods

The method of Hellebaut can be simplified as shown in the previous section. However even simplified it has still some drawbacks. Indeed, Hellebaut's method uses the SfM technique and requires adding GCPs to generate the DEM in the same coordinates system used by the bathymetry model. SfM also requires many pictures from different points of view of the place that need to be modelized. That is why a

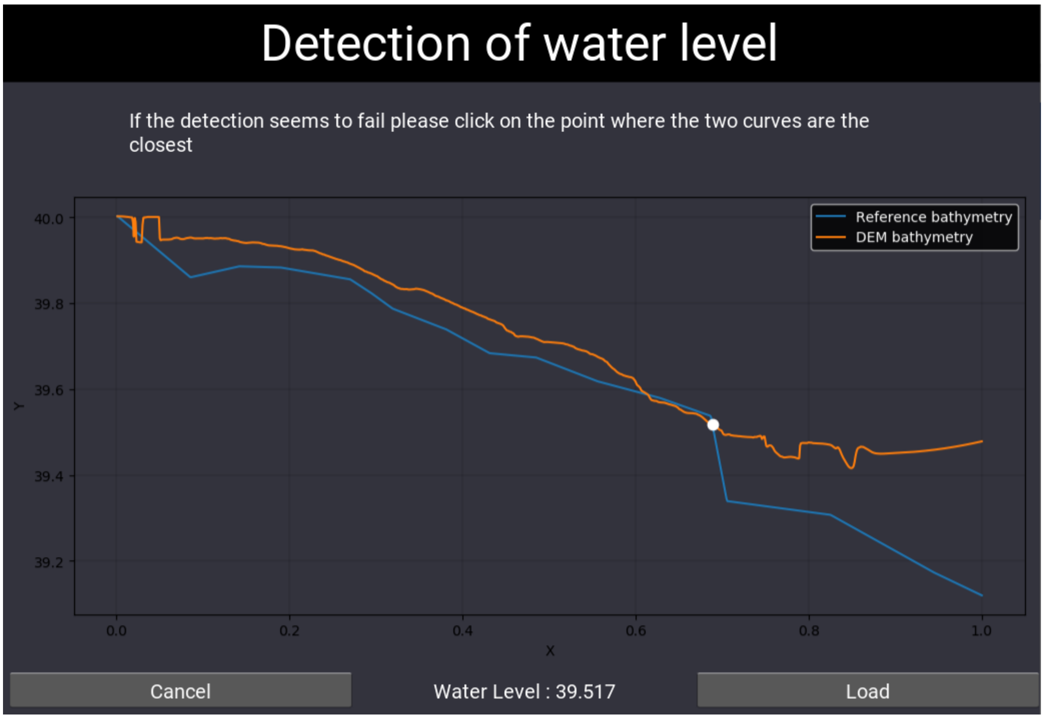


Figure 5.3: Correction step proposed in the application to the user.

method where less images are needed and where GCPs can be avoided would be nice. Thus, the scaling factor between pixels to meters must be computed without GCPs.

In this context two alternative methods of the Hellebaut's one have been developed: Angle and Height method and Image method. They are both based on the following principle: combine the real river width and the cross-section curve to deduce the water level. They differ on the acquisition of the real river width: the first one uses two images taken from different altitudes, the second one use orthorectification. Thus to use the Image method an orthorectifiation step (see Chapter 7) must be performed earlier because this method takes advantage of the mapping done in this step. In this case we lose the advantage of getting rid of the GCPs since they are required for the orthorectification but the computation is more straightforward. These two methods are covered in the following sections. The influence of errors measurements in the Angle and Height methods will also be presented.

5.2.1 Angle and Height method

Required data

The method takes the following inputs:

- Two images of the river, each taken from above at a different altitude but on the same vertical axis
- The difference of altitude in meter between the two images
- The field of view (FOV) of the camera

The two images can be taken from a drone which can be easily placed above the river and change altitude. To measure the difference in altitude between these images the drone must have a GPS system. The FOV describes the angular extent of a given scene taken by a camera, it can be measured horizontally, vertically, or diagonally [77]. The computation of the FOV is done in the calibration part (see Chapter 7) since this step allows to measure the focal value. Once the focal length is known the following equation can be used to compute the FOV:

$$FOV_x = 2 \arctan \frac{W}{2\,f_x} \qquad FOV_y = 2 \arctan \frac{H}{2\,f_y} \qquad FOV_{diag} = 2 \arctan \frac{\sqrt{W^2 + H^2}}{2\,f}$$

Where W and H refer to the width and height of the image. The FOV_x and FOV_y are the horizontal and vertical component of the FOV. The reasons two focal length f_x and f_y are used is because the individual pixels on a typical low-cost imager are rectangular rather than square [78]. The focal length f_x for example is the product of the physical focal length F and the size $s_x = \frac{W}{w}$ of the individual imager elements. These focal lengths are also used since F and s_x and s_y cannot be measured directly. In the formula of the diagonal FOV the f will be here taken arbitrarily as equal to f_x since f_x and f_y are supposed to be almost the same.

This equation comes from the trigonometric relation that can be deduced from the schematic shown on the Figure 5.4. These components of the FOV are computed because in the method the value used must correspond to the axis of the section of the river that may not be horizontal in the image. The way which one will be used is determined is by looking at the angle of the section of the river on the image: below 30° FOV_x is taken, above 60° FOV_y and between the FOV_{diag}. The angle can be computed thanks to the points given by the user as it will be mentioned later.

The formula used for the computation of the FOV and the formula that will be developed in the following assume a rectilinear pinhole camera model where straight lines in the real world are mapped to straight lines in the image generated by the camera. However, most real optical systems will introduce some undesirable effects, rendering the assumption of the pinhole camera model inaccurate [79]. This is also why a calibration step is useful to correct these effects before doing the analysis.

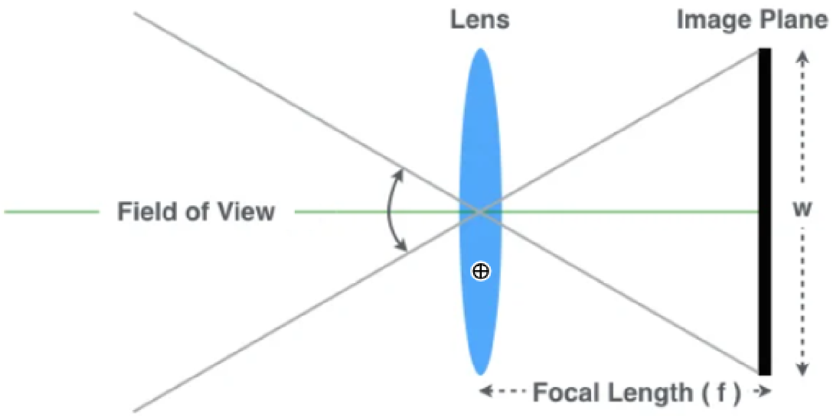


Figure 5.4: Relationship between Focal Length and Field of View. Figure from [80]

Principle

The idea of this method is to retrieve the width of the river occupied by water. Then combined it with the section profile to find the water level value. Two implementations of this method are possible, one where the drone is considered centered in the middle of the river and another more general which considers the potential shift of the drone. In the application the two cases are used. The first one has the advantage to be easily computed while the second one is more accurate but requires a numerical solution. The computation is done using the two and if the second one has failed it returns the first solution. In the following the two implementations will be covered.

Center drone case

In this case the configuration is the one depicted on the Figure 5.5. The alpha angles represent the angles of view of the river width at the two heights. They are computed using the following formula:

$$\alpha = FOV\,\frac{\#\mathrm{Pix}_L}{\#\mathrm{Pix}_{img}}$$

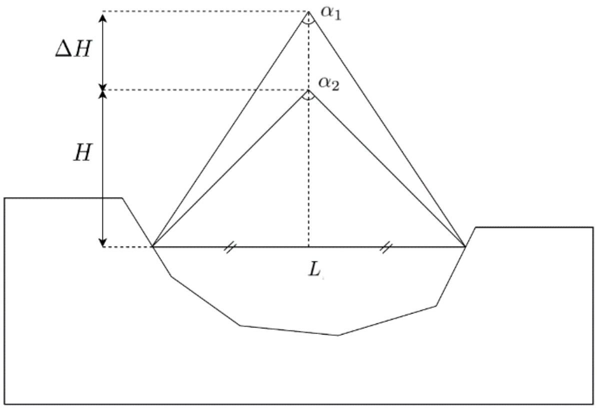


Figure 5.5: Center-drone configuration.

Their computation requires the value of the FOV, the size of the river section and image in pixels which are called Pix_L and Pix_{img} in the formula. The Pix_L value is found thanks to the user which will have to select two points that delimits the section of the river. From these points the angle of the section and so the FOV used can also be determined. The $\#\mathrm{Pix}_{img}$ is simply the width/height/diagonal size of the image that can be easily retrieved having the image. The size that will be selected follows the same principle that the FOV.

The implication of the user for the section on the image could be avoided in the future by using an algorithm that automatically detects it. However, this requires either detection algorithms or machine learning program adapted for rivers. This is not the subject of this Master Thesis so for the moment a user intervention is still needed.

This process just described is repeated for the two images which allows the computation of α_1 and α_2. Once those are known along with the difference in altitude between the two image denotes by ΔH the following system can be solved:

$$\tan\left(\frac{\alpha_1}{2}\right) = \frac{L/2}{H + \Delta H} \tag{5.1}$$

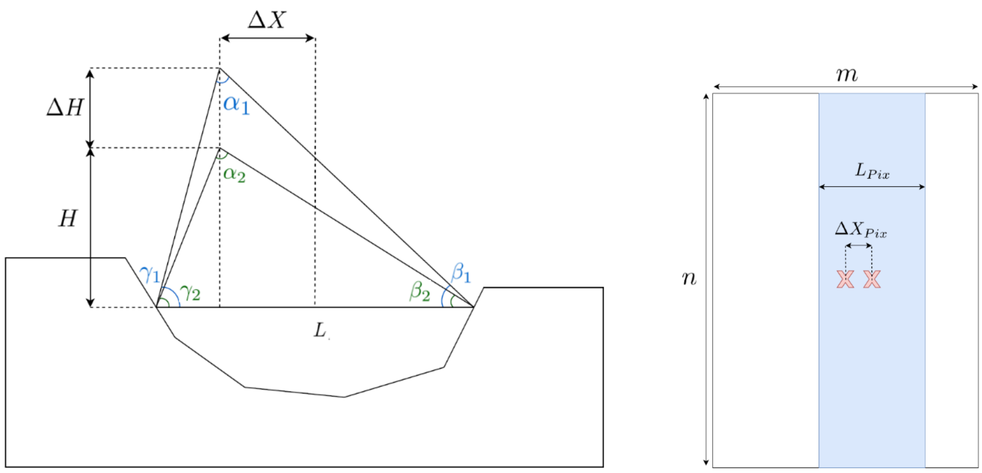
$$\tan\left(\frac{\alpha_2}{2}\right) = \frac{L/2}{H} \tag{5.2}$$

Where the unknowns are the river width L and the height H of the drone during the take of the first image.

Off-center drone case

In practice the drone will most likely be off-center as represented on Figure 5.6 where ΔX denotes the shift from the center and is positive to the left and negative otherwise. The system of equations to solve becomes the following:

$$\begin{aligned}
\tan(\gamma_1) &= \frac{H + \Delta H}{\frac{L}{2} - \Delta X} & \tan(\gamma_2) &= \frac{H}{\frac{L}{2} - \Delta X} \\
\tan(\beta_1) &= \frac{H + \Delta H}{\frac{L}{2} + \Delta X} & \tan(\beta_2) &= \frac{H}{\frac{L}{2} + \Delta X} \\
\alpha_1 + \gamma_1 + \beta_1 &= \pi & \alpha_2 + \gamma_2 + \beta_2 &= \pi
\end{aligned}$$



(a) Off-center drone configuration step

(b) ΔX value in pixels computation

Figure 5.6: Off-center drone case.

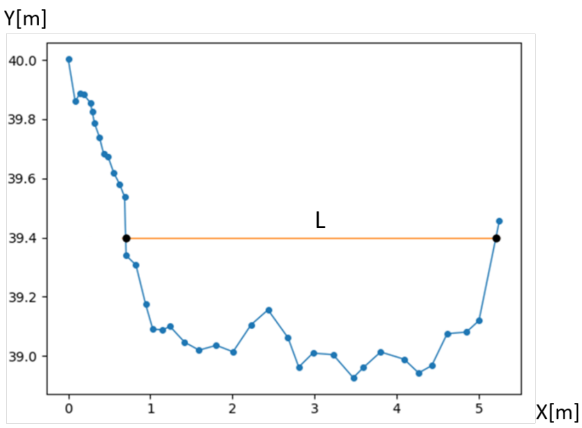
All the needed inputs are the same as before with an additional one, the ΔX. It cannot directly be known but the equations can be arranged such as only the relative shift $\frac{\Delta X}{L}$ appears which can be computed using the following equality:

$$\frac{\Delta X}{L} = \frac{\Delta X_{Pix,1}}{L_{Pix,1}} = \frac{\Delta X_{Pix,2}}{L_{Pix,2}}$$

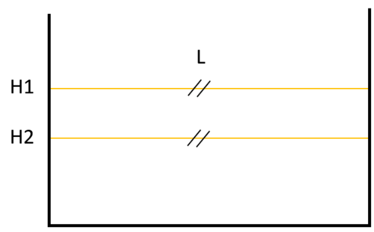
This equality can be written since the relative shift is the same if it is express in term of pixels length or real ones. The relative shift value is simply computed using the pixel lengths ratio since the shift in terms of pixel is simply the difference between the center of the image and the center of the river occupied by water which are both known. This is shown on the Figure 5.6. The average value of the ratio computed for both images is taken.

Water level retrieving

Once the river width is found the water level must still be retrieved. The principle is the following, starting from the top of the bathymetry curve a line is moved down and the intersection points with the curve is computed. When the distance between the two intersection points is equal to the width found earlier the algorithm stops and returns the corresponding height. However, the algorithm will not work in some bathymetry configurations. For example, it fails in the case of a rectangular bathymetry where all heights correspond to the same length. This is represented on the Figure 5.7. Since the measurements can lead to the wrong width value this method may fail. In the example display if the width found was greater than the height of the rightmost point of the cross-section the algorithm would have failed since no height has the width L between two intersection points above this point. Ideally the bathymetry should vary slowly such that small errors have less impacted on the water level value. Tests on real river should be done in the future to see if this method can lead to acceptable results in field conditions.



(a) Free cross-section case



(b) Rectangle cross-section case

Figure 5.7: Retrieve of the water level based on the length L occupied by water.

5.2.2 Sources of errors and expression of uncertainty of the measurements

This method requires different measurements where errors can occur and accumulate. The first one is the assumption that the optical axis of the camera is vertical for both images, this can be difficult to maintain depending on the weather conditions. This could be solved with the use of an internal gyroscope but limitations on load capacity and power supply make their use impracticable. An article has investigated this problem [81] and proposes a solution based on a pattern placed on the ground to detect the inclination of the images. Solutions exist to counteract this issue but there are not implemented here.

Another assumption is that the river section chosen by the user is where the bathymetry curve has been computed. To ensure that feature can be put on the image near the river's section where the bathymetry has been computed. The other sources of errors are the GPS measurements and the user intervention to give the pixel width by clicking on the images.

No experiment has been conducted on real river so good values for the height of the drone and the ΔH cannot be drawn from measurements. However, it is possible to get an estimate of the uncertainty on a measurement using the error propagation law [82]. This will be done on the equation 5.2 which can be written $L = 2\,H\,\tan\left(\frac{\alpha}{2}\right)$. The aim is to see the dependency of the uncertainty of H and α on L. In the case where the quantity depends on two variables the error propagation law is the following:

$$\frac{\delta f}{|f|} = \frac{1}{|f|}\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 (\delta x)^2 + \left(\frac{\partial f}{\partial y}\right)^2 (\delta y)^2 + 2\,\frac{\partial f}{\partial x}\,\frac{\partial f}{\partial y}\,\delta x\,\delta y}$$

The last term of this equation is zero if the two variables x and y are uncorrelated. In our case α and H are correlated so the last term must be kept. In this case the expression becomes:

$$\frac{\delta L}{|L|} = \frac{1}{|L|} \sqrt{\left(\frac{\partial L}{\partial H}\right)^2 (\delta H)^2 + \left(\frac{\partial L}{\partial \alpha}\right)^2 (\delta \alpha)^2 + 2 \frac{\partial L}{\partial H} \frac{\partial L}{\partial \alpha} \delta H \, \delta \alpha}$$

$$\frac{\delta L}{|L|} = \sqrt{\frac{1}{H^2} (\delta H)^2 + \frac{1}{4 \left[\sin\left(\frac{\alpha}{2}\right) \cos\left(\frac{\alpha}{2}\right)\right]^2} (\delta \alpha)^2 + 2 \frac{1}{H \cos\left(\frac{\alpha}{2}\right) \sin\left(\frac{\alpha}{2}\right)} \delta H \, \delta \alpha}$$

This formula suggest to use high values of the height if we only look at the terms in $(\delta H)^2$ and δH. However, using high values of H means that the α values will be smaller since it represents the angle of view of the object, here the river, on the image. A compromise must be found between the two.

Similar experiment [83] has been done where the goal was to measure the height of a drone using an object of known size on the image. During the tests of his method, the author has measured the distance to a sign and shown on graphs, such as the one displayed on the Figure 5.8, the relation between the pixel size of the sign and the distance from it. The shape of the curve suggests that a compromise must be found between the two variables size. If the pixel size is large it means that uncertainty on it will have low impact of height value but the uncertainty on height will have a large impact in that zone. If the height is large the uncertainty regarding the height will have small impact on the pixel size but the pixel size uncertainty will have a large impact. This compromise will depend on the quality of the measurement for example the precision of the GPS for the height.

5.2.3 Image method

When an orthotorectification step has been performed, see Chapter 7 the width of the river in meters can be easily retrieved. Indeed, during this step, a mapping between pixels coordinates and real-world ones is done. This mapping is done thanks to 4 GCPs whose distances from each other are known in meters. The real river width is thus simply deduced from the value in pixels on the orthorectified image. Once the real river width is known the water level is retrieved using the same procedure presented in the Angle and Height method.

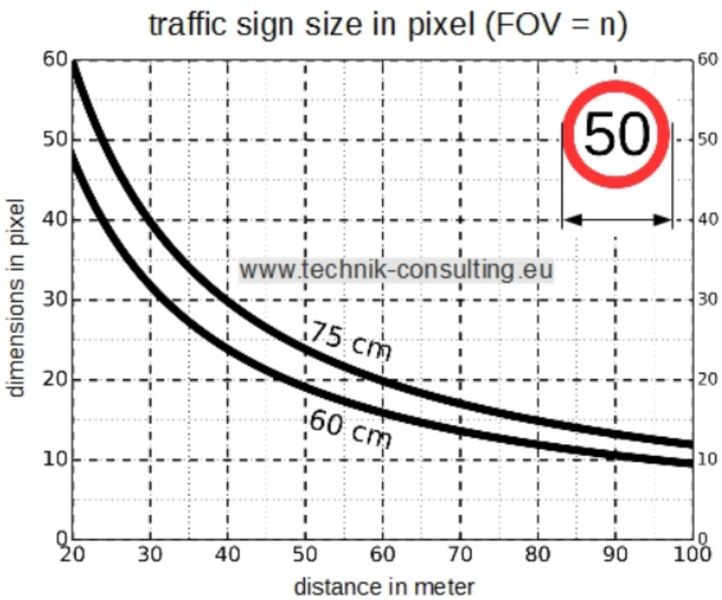


Figure 5.8: Evolution of the size in pixel and distance from an sign with a radius of 60 and 75cm.

Chapter 6

PIV software/packages used for the velocity analysis

In the State-of-the-art chapter, a method called PIV was presented, allowing to measure the velocity field on the base of flow images. During a PIV analysis several algorithms are involved and all of them have an impact on the results. Choosing the most adequate algorithms and implementing them can be quite tedious. That is why several software, packages, toolboxes have been developed over the years to ease the PIV analysis. In the following sections two software allowing to compute it will be presented and compared the last one being used in this application.

6.1 PIVlab

PIVlab is a MATLAB toolbox which uses the PIV method to calculates velocities in image data (movies, pictures). It was created by William Thielicke in 2010. He is a German biologist who has a doctorate in biomimetics/fluid dynamics/flapping flight. At first, he created it to study the flapping flight but today the field of applications has grown. For example, William Thielicke says [56] that it can be used in flow visualization within cells, echocardiographic velocimetry of the human heart and deformation of sand and gravel.

The Master Thesis of Bia and Somers was focused on the PIV analysis applied on river. Instead of implemented themselves the method they have searched for existing tools to perform a PIV analysis. During their search they have found PIVlab. Since they already had a MATLAB license, they had free access to it compared to other commercial software. Moreover, since the code was available and customizable, they could easily enhance it according to their needs. William Thielicke has also created a forum on his site where users can help each other and on which he is very active. New versions of the software came up regularly which keeps it up to date. For all these reasons PIVlab was chosen by Bia and Somers for their work. Another advantage is the features provided. They are all listed on the website dedicated to PIVlab [84], in the following the main ones will be described.

To have a description on how to conduct an analysis using PIVlab with the different options available at each step, the Master Thesis of Bia and Somers [49] can be accessed as well as this tutorial video [85], realized by William Thielicke himself. It is not covered here because it is already done in another Master Thesis and PIVlab is not the tool used in this work. To compare the two a general overview of how PIVlab works is sufficient, this can be found on the workflow represented on the Figure 6.1. On this workflow you can see that in addition to the image evaluation which is the core of the PIV method several stages are proposed such as image pre-processing, post-processing, and data exploration.

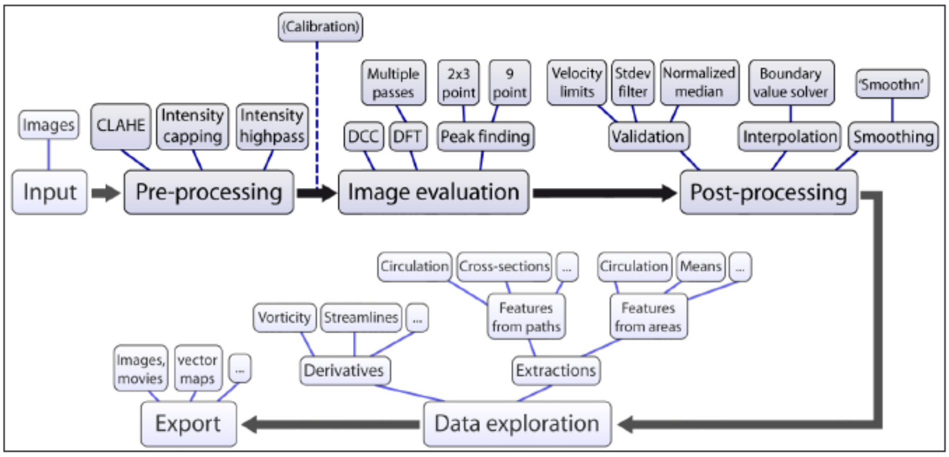


Figure 6.1: Overview of the workflow and the implemented features in PIVlab. Figure from [56].

Before going to these stages let us start with the global feature of PIVlab: it is a user-friendly and GUI based PIV tool which makes the analysis easier to conduct. This enables other features like the selection of a region of interest (ROI) or the application of masks after images loading. Then there is the pre-processing part where the user can apply filters on images to help the analysis later. The following filters are available: CLAHE, High-pass, Intensity capping and Wiener2 denoise. They will be addressed in the image processing chapter.

In the image evaluation the user has the option of performing a multiple pass Discrete Fourier Transform (DFT).

After the computation of the velocity field, in the post-processing part there is a validation vector step where different methods can be used. A sub-pixel estimator, the boundary value solver, is also available.
Once the analysis is done the user can extract the features, data he wants and export them under different formats.

6.2 OpenPIV

Non-commercial software which implement the PIV method are not numerous. Among them PIVlab and OpenPIV are often recommended. In this work OpenPIV has been chosen over PIVlab for the main reason that it exists in Python. Moreover, in Python OpenPIV can be used simply by importing some modules which make it easier to integrate in the application.

The Python version of OpenPIV will now be described. Regarding the installation of OpenPIV or to have a more detailed explanation their documentation can be consulted [86]. The first step it to load the different modules of OpenPIV. The next step is to read the image files to transform them into NumPy arrays. NumPy is one of the Python libraries used (see Chapter 3). Then a call to the `extended_search_area_piv()` function is made to process images by pairs. In addition of a pair of images this function requires several parameters. For example, the search and interrogation windows size and the overlap between adjacent windows in pixels. The user must also provide the delta of time dt between the two frames in seconds. The final parameter offers the choice among two metrics for the returned signal-to-noise ratio: peak2mean and peak2average. This function returns three arrays containing respectively the velocity u, v, and the signal to noise ratio named sig2noise obtained from each cross-correlation function. The latter will be used later in the validation of vector step.

The module called pyprocess of OpenPIV has a similar function just called `piv()` which takes almost the same inputs as the previous one. The only differences are the correlation method which needs to be precise here and the possibility to choose a subpixel estimator. The user has the choice for correlation method between the DCC or the FFT one. The latter is the value by default. The different subpixel estimators proposed are the following: centroid, gaussian and parabolic.

Once the velocity has been computed the coordinates of the center of the interrogation windows are retrieved using the function `get_coordinates()`. This function needs the window size, the overlap, and the image size. Knowing the velocity and the coordinates of the nodes where these appears, the results can then be displayed using the quiver library.

In terms of post-processing OpenPIV provides a function called `validation.sig2noisechar_val()`. It takes the velocity arrays u and v as well as the sig2noise ratio array. The user sets the threshold parameter to a certain value under which the vector will be considered as spurious. The function returns the velocity arrays u,v which now contains some NaN elements and a mask which is a boolean array where elements corresponding to invalid vectors have been replace by NaN.

Finally to replace these spurious vectors a function `filters.replace_outliers()` is used. which takes as input the arrays of velocities u,v containing the NaN elements. This function implements an iterative image inpainting algorithm with a specified kernel where the number of iterations as well as the kernel size must be provided. There is only one method available for the moment which is called 'localmean' and as it indicates replaces the missing vectors by a local mean of the valid surroundings ones.

Once the analysis is done the results can be saved in a text file using the `save()` function from the tools module. The result can also be displayed using the `display_vector_field()` which reads the text file containing the results. Some others function like the `scaling.uniform()` one allows to get velocity and coordinates in m/s or m instead of pixel/s and pixel using a scaling factor given by the user.

6.3 Comparison

In the OpenPIV documentation written the 03/01/2020 the authors list the things that OpenPIV currently needs. The implementation of advanced processing algorithms, good documentation, flow field filtering and validation functions were cited. Regarding the documentation it is incomplete since some section titles are presented with no content for the moment. Some tutorials are available, but they stick to simple cases. For example, a multipass seems to be possible with OpenPIV but there is only a short script tutorial about it which is not clear, and no explanation can be found in the documentation on that. The authors of OpenPIV are present in different forums but there is not one forum dedicated to it, making the information sparse and not always easy to find.

Unlike PIVlab there is no preprocessing step where filters can be applied. For the images, no mask or ROI can be selected although this might come when OpenPIV will have a GUI in Python. Regarding the post-processing the validation vector step is only based on the signal ratio no other methods are proposed unlike in PIVlab. Regarding the features that can be extracted and the format in which the data is saved PIVlab proposed a larger choice than OpenPIV. Multipass is implemented in PIVlab and easily done.

OpenPIV exists in three different languages MATLAB, C++ and Python. This explains why the updates of the different version take some time. However, the community keeps growing and the code being open source the number of contributors increases also. Moreover, the Python OpenPIV version seems to be the one that will undergo the major development in the next years due to the growing interest to the language.

Despite the absence of some features, OpenPIV will be used since it offers the basic tools to perform a PIV analysis in Python. The principal missing features are those in the pre-processing step (e.g., filters, selection of ROI, masks). To address this problem and have these features in the application, the functions of the OpenCV library will be used. Moreover, the features of PIVlab are a source of inspiration for those that will be implemented in the application.

Chapter 7

Image Processing

7.1 Calibration

Cameras can cause different types of distortion which introduce changes in the geometry of the image. Working with image-based techniques, these changes in the geometry can have a significant impact on the quality of the result. The goal of the calibration process is to find the parameters that will enable the correction of the distortion in the images.

Among these parameters there is the focal length which is used to compute the FOV of the camera needed in one of the alternative methods called Angle and Height method used for measuring the water level (see Chapter 5). This is computed along with the calibration since the formulas developed for this method are based on the pinhole camera model which is inaccurate if the image present distortion effects.

To perform the calibration and correct the distortions since the OpenCV library provides some functions dedicated to it [87] they will be used in the application.

7.1.1 Types of distortions

The two major kind of distortion are the radial and tangential ones. The radial distortion makes straight lines look curved and it arises because of the shape of the lens [88]. The effect is larger for the points farther from the center of the image. Two types of radial distortion can occur, a positive one called Barrel distortion and a negative one called Pincushion distortion. They are both represented on the Figure 7.1.

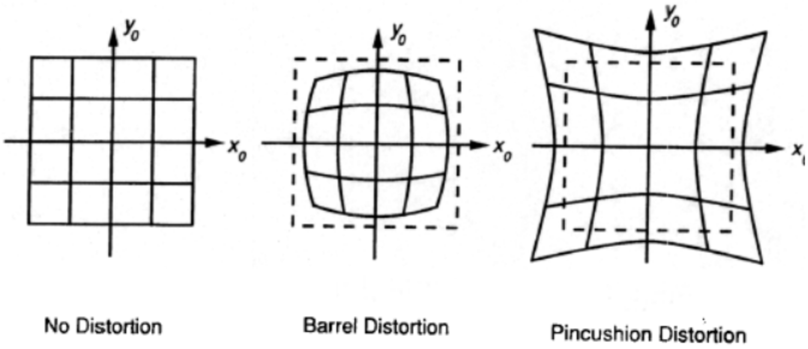
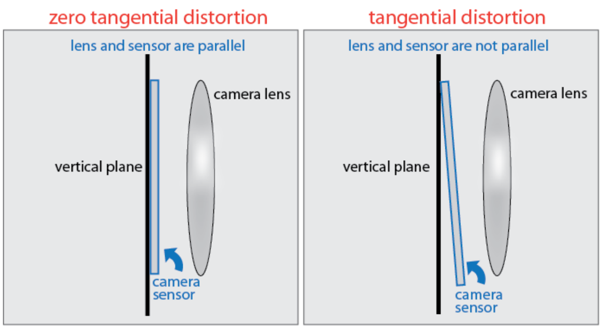


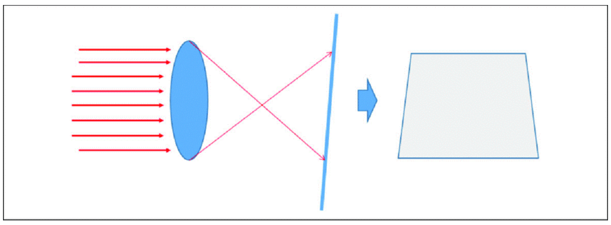
Figure 7.1: Types of radial distortions. Figure from [89]

.

The tangential distortion arises from the assembly of the camera. Indeed, it occurs when the lens and the image plane are not parallel as shown on the Figure 7.2. Thus, some areas look nearer or farther than they really are. The radial distortion is more common than the tangential distortion and introduces more changes on the geometry of the image. That is why most calibration procedure focus on the correction of the radial one but here the correction will address the two types.



(a) Tangential distortion cause



(b) Effect of tangential distortion

Figure 7.2: Tangential distortion. Figure from [90] and [91]

7.1.2 Calibration process

To undistort the images the intrinsic and extrinsic properties of a camera are required. The intrinsic parameters are specific to a camera and include the focal length (fx,fy) and the optical center (cx,cy). Extrinsic parameters designate a rotation and translation which take the world coordinates system to the camera coordinates system. So, the extrinsic parameters allow to go from the real world to the camera and finally the intrinsic to the image plane as represented on the Figure 7.3. The matrix formulation is the following:

$$s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{C} \begin{pmatrix} X_w \\ Y_W \\ Z_w \\ 1 \end{pmatrix} \tag{7.1}$$

$$\mathbf{C} = \mathbf{K}\,[\mathbf{R}\,\mathbf{t}] \tag{7.2}$$

In the equation 7.1, s is the scaling factor, **C** the camera matrix, the coordinates on the left hand side are the image points and those on the right hand side the real world ones. In the equation 7.2 **R** is the rotation matrix, **t** the translation vector and **K** the intrinsic matrix.

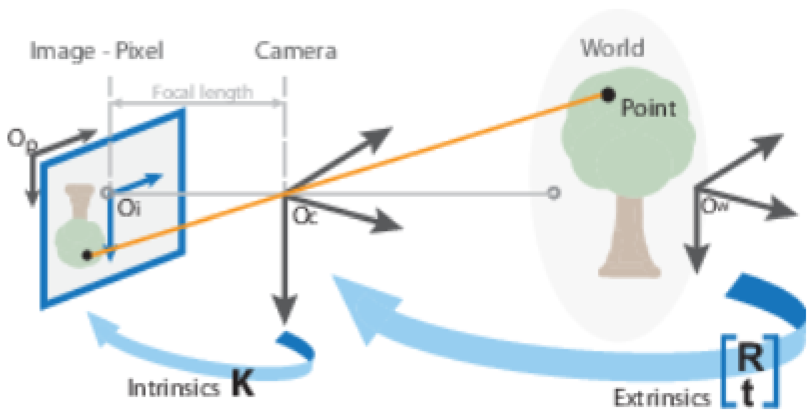


Figure 7.3: Intrinsic and extrinsic parameters usefulness. Figure from [90]

.

To retrieve the intrinsic and extrinsic parameters a set of known 3D points (X_w, Y_w, Z_w) and their corresponding pixel location (x, y) on the images must be used. To provide such data a well-defined pattern of known dimensions is photographed at many different orientations. It is recommended to use at least 10 test images to achieve better results [87]. A recurring pattern is the checkerboard one.

It arises frequently in computer vision because of its geometry which is well-suited for algorithmic detection. This work will be no exception and will choose it as well.

The checkerboard pattern will be fixed to a wall and the 3D points will be the corners of the squares except those at the borders. The coordinates system is the one depicted in the Figure 7.4. The Z axis is chosen perpendicular to the wall therefore, all points on the checkerboard are on the XY plane and $Z_w = 0$. Thus, the Z components in equation 7.1 can be ignored and the Camera Matrix C is simplified into an homography matrix, the explanation is given in the paper of Zhang [92]. The remaining coordinates are (X_w, Y_w) which can be easily defined with respect to the origin chosen. Indeed, all points on the checkerboard are equally spaced and the dimensions of the squares are known.

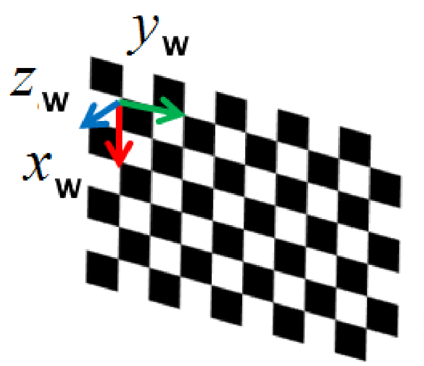


Figure 7.4: Coordinates system of the real world points. Figure from [93]

.

Regarding the 2D coordinates in the image they can be easily retrieved using algorithmic detection. OpenCV provides a detection function `cv2.findChessboardCorners()` that finds the checkerboard corners coordinates on images. The parameters of the function are the image, the dimensions of the checkerboard pattern used and the size of the squares composing it. For example, in the test shown on the Figure 7.5 a 7x5 checkerboard with square of 40mm was used.

The algorithm which finds the corners of the chessboard might failed on some images. In the application to ensure enough valid images are taken for the calibration, the input is a video of the checkerboard pattern where the camera is moved from different distances and angles. The idea is to cover the maximum pattern's positions on the image. Then the user states the number of images n, he wants to use for the calibration. The program will divide the video frames in n parts and in each of them will take the first image for which the detection algorithm has worked. This ensures that only valid images are taken. On the Figure 7.5, 2 images used for the calibration test are shown where the corners found by OpenCV are highlighted.

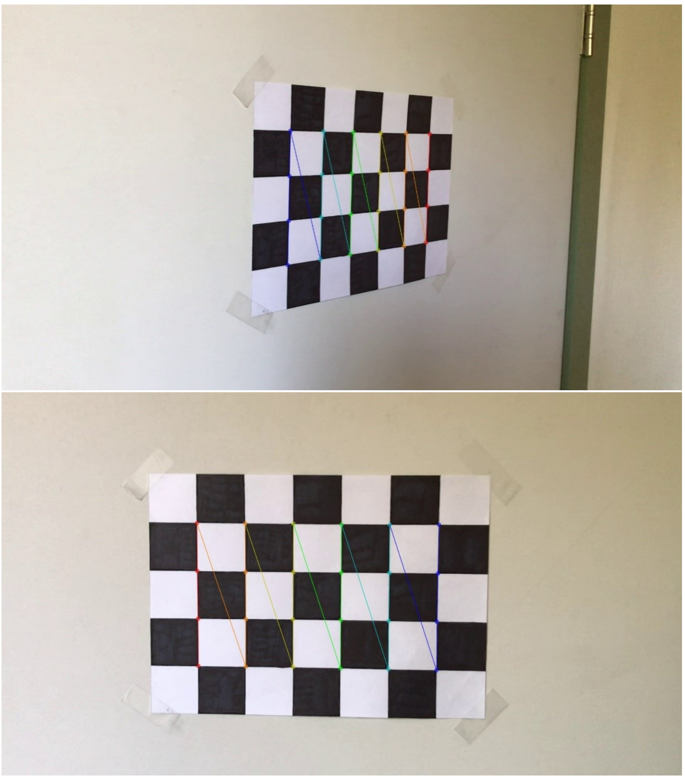


Figure 7.5: Images used for testing the detection algorithm.

Once the real world and images points coordinates are known another OpenCV function `cv2.calibrateCamera()` is called. This function returns the intrinsic, extrinsic parameters, the camera matrix, and the distortion coefficients. Finally using the `cv2.undistort()` function with the camera matrix and the distortion coefficients, the images can be undistort.

To have an idea of how just the parameters found are, a reprojection error can be computed. The closer it is to 0 the more accurate the parameters are. In the calibration test the mean value of the reprojection error was 0.185 pixel.

In addition to the correction of the image, the FOV can be deduced from the dimensions of the image (H,W) and from the focal length computed during the calibration.

The equations giving the horizontal, vertical and diagonal FOV are the following:

$$FOV_x = 2 \arctan \frac{W}{2\,f_x} \qquad FOV_y = 2 \arctan \frac{H}{2\,f_y} \qquad FOV_{diag} = 2 \arctan \frac{\sqrt{W^2 + H^2}}{2\,f}$$

These formulas were covered in the chapter 5. OpenCV provides a function that uses the same equations [94].

7.2 Filters

Filters are often applied before PIV analysis to enhance the distinction in images between the particles and the fluid, water in our case. This will help the algorithm to find the displacement of the particles leading to a better accuracy of the result. In the PIVlab software four filters are available: CLAHE, High-pass, Intensity capping and Wiener2 denoise. However, in OpenPIV there is no preprocessing tools for the moment.

Since it can lead to significant improvement, a filter step has been implemented in the application. The first filters that were implemented were inspired by the PIVlab ones except the Wiener2 denoise whose equivalent in python has not been found. Then other filters have been added to widen the possibilities for the user. Each filter in the application address a specific issue that might occurs in the image.

To ease the process, each filter has default settings which are either based on the images features or are generic values. The user can adjust these settings. The filters and their principle will be covered in the following.

7.2.1 Intensity High-Pass

The particles have generally a higher pixel intensity value than the surrounding fluid. A high-pass filter conserves the high frequency and suppresses the low frequency information in the images. In our case this will emphasize the particles presence in the image and unify the background. This process is also called sharpening. The way it is done in the application is to derive a high-pass filter from a low pass one (high-pass = image - low-pass). This is done on grayscale images and in the following way:

1. A Gaussian low-pass filter is applied to get a blurred image
2. This image is subtracted to the original one which has for effect to unify the background.
3. All pixels are multiplied by a factor to enhance the contrast

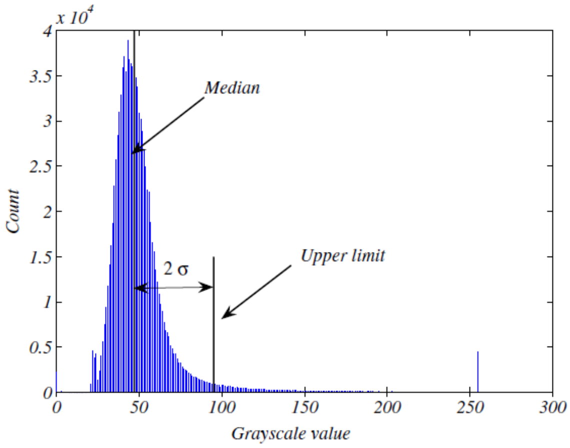
The last step is necessary since the particle's intensity has also decreased during the second step. The factor value is set to the value $\frac{255}{max(pixels_{intensity})}$ [49]. The user can change this value as well as the ones proposed for the kernel of the Gaussian filter. The different steps are shown on the Figure 7.6.

Figure 7.6: Intensity High-Pass filter. Figure from [49]

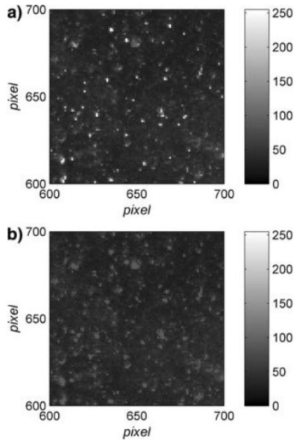
.

7.2.2 Intensity Capping

In the images some spots are characterized by a grayscale intensity much greater than the mean intensity of the image. The main cause is the intense scattering of the seed particles. The presence of these bright spots can degrade the PIV results [95]. Indeed, bright spots will contribute more to the correlation signal which may bias the resulting velocity vector. An easy way to counteract this is by applying an intensity capping filter. An upper limit of the grayscale intensity is imposed and all pixels which exceeds this threshold are replaced by the limit value. The upper limit is set by default to the median value plus two times the standard deviation, but again the user can change it. This filter has also the advantage of impacting only a small subset of pixels limiting the potential negative impact of image modifications. The principle is illustrated on the Figure 7.7.



(a) Principle of the Intensity capping filter.



(b) Effect of the Intensity capping filter, (a) before, (b) after.

Figure 7.7: Intensity Capping filter. Figures from [95]

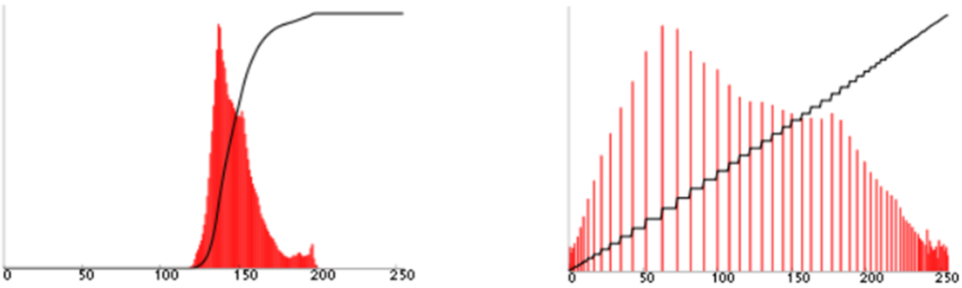
7.2.3 CLAHE

CLAHE stands for Contrast Limited Adaptive Histogram Equalization. The basic idea of histogram equalization is to take advantage of the full spectrum on the pixel intensity values. Indeed, some images have pixels intensities that are confined to a certain range of values, an easy way to see it, is by plotting a histogram. The fact that they are confined decreases the contrast within the image. That is what histogram equalization is about using a transformation function that maps the original pixels values to the full spectrum. An example can be seen in the Figure 7.8.

Histogram equalization works when pixels values are confined to one region. However, it will fail when histogram covers a large region of the spectrum like when very bright and dark pixels are both present for example tracer seeds and water. It fails since it considered the global contrast of the image. Adaptative histogram equalization (AHE) was developed to prevent this effect. The principle is the following, the image is divided in small blocks called tiles and on each block a histogram equalization is performed. If noise is present in a small block it will be amplified using this procedure. To solve this a contrast limiting is applied which gives the CLAHE algorithm. It limits the amplification by clipping the histogram at a value called clip limit. The pixels above this value are clipped and distributed uniformly to the other bins of the histogram before the equalization. Finally, to remove the borders between tiles an interpolation can be applied. A comparison between AHE and CLAHE filters can be seen in Figure 7.9. OpenCV has a function to perform

CLAHE algorithm which was used in the application. The tile grid and the clip limit are set to the value by default of OpenCV, but they can be changed by the user.

(a) Image before and after the application of an Histogram Equalization filter.



(b) Effect of the filter on the pixels histograms in red and cumulative histogram in black.

Figure 7.8: Histogram equalization filter. Figures from [96]

Figure 7.9: From left to right, original image, after AHE filter and after CLAHE filter. Figure from [96]

.

7.2.4 Brightness and contrast adjustment

If the images are underexposed the brightness and the contrast can be adjusted to correct them. The function used is the following:

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

Where alpha controls the contrast, beta the brightness and f and g are two matrices representing the pixel intensity values of the input and output image respectively. The pixel intensities which falls outside the range $[0,255]$ after the application of the function are set to the nearest valid value. On the Figure 7.10 an example of image which has been corrected is shown. The user can adjust the alpha and beta value if the ones by default are not suitable.

Figure 7.10: Application of a brightness and contrast filter with $\alpha = 1.3$ and $\beta = 40$. Figure from [97]

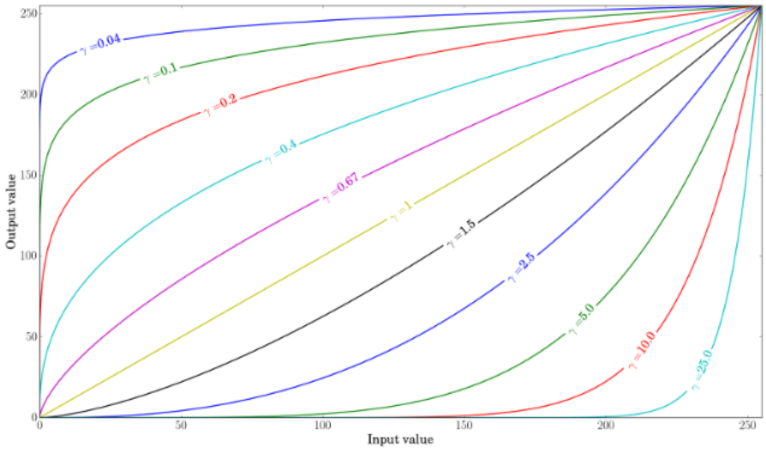
.

7.2.5 Gamma correction

The gamma correction is used to correct the brightness of an image but using a nonlinear transformation. The following function is mapping the pixels to their new values:

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

Where I is the original pixel intensity. Again, if pixel intensities fall outside the grayscale, they are set to the nearest valid one. The case where $\gamma < 1$ will make the dark regions brighter, resulting in a shift of the histogram to the right, while $\gamma > 1$ will do the opposite. The correction of the same example as before but using a gamma correction is represented on the Figure 7.11. The user can play with the gamma value.



(a) Graph of different values of γ

(b) Application of a gamma correction filter with $\gamma = 0.4$.

Figure 7.11: Gamma correction Filter. Figures from [97]

7.2.6 Non-local means filter

The background noise can also impact the result of a PIV analysis. One method that provides very good result but takes more time to perform is the non-local means denoising. The noise is generally considered to be a random variable with zero mean. The idea of the non-local mean denoising is to consider small windows in the images and find similar region in the image. Then the average of all the windows is taken and used to replace the pixels value in these regions. OpenCV has a function which implements this algorithm called `fastNlMeansDenoising` which works on grayscale images. The parameters are the filter strength, windows size, and search windows size. An example is displayed on the Figure 7.12.

Figure 7.12: Application of a non-local means filter. Figure from [98]

.

7.3 Orthorectification

The orthorectification is a geometrical correction of images that aims at presenting them in a bird's eye view [99]. Sometimes the easiest way to take pictures or videos of a river is from the shoreline. However, some distortion is induced due to the camera angle making the velocity vector computed in PIV analysis spurious. Therefore, an orthorectification step may be handy to correct them. This step in the application make it usable even if a drone is not used. The user can skip this step if the images does not need an orthorectification.

The main goal of the orthorectification is to correct the velocity vectors giving by the PIV analyis. Depending on the camera resolution an image may contain millions of pixels. Velocity vectors on the other hand are computed on a grid defined on the image, limiting the number of vectors to a few thousand. Performing a correction on images or velocity vectors is equivalent but doing this correction of the velocity vector is more interesting. Indeed, the computational time of the process will be higher if images are rectified [63]. Another application of the orthorectification results is to compute the water level has explained in the Chapter 5 with the Image method.

The orthorectification process can be done by finding a transformation between the points coordinates in the image plane and the real world. In the real world the only thing that we are interested in is the surface of the river. The latter can be considered as plane. If some features are placed on the same level that the river's surface the transformation to perform the orthorectification becomes a planar transformation. Several planar transformations existed [100]: translation, rotation, scale, aspect ratio, shear, distortions. The affine transformation includes the first 5 and possess 6 degrees of freedom (dof) while the homography transformation includes all 6 and has 8 dof so this latter has been chosen. The transformation between one plane to another using the homography matrix is done by the following expression:

$$s \begin{pmatrix} X_p \\ Y_p \\ 1 \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & 1 \end{pmatrix} \begin{pmatrix} X_r \\ Y_r \\ 1 \end{pmatrix} \tag{7.3}$$

Where the coordinates on the left-hand side are the image ones and on the right-hand side are the real world ones. If the terms H_{31} and H_{32} are zero the transformation becomes an affine transformation. Generally, an affine transformation is a decent approximation of the homography one. This can be observed by looking at the values of these terms in the homography matrix when it is applies on an example.

To compute an homography matrix 4 or more points must be used [101]. In the application four Ground Control Points (GCPs) are used, 2 on each shore and placed as close as possible to the water level so that they share the same plane as the water. They should be placed such that the section of interest of the river is inside the shape formed by these GCPs. A GCP is a feature that you can clearly identify in the image so it can be a rock, a piece of wood, anything that can be easily recognized. The six real distances between them should be measured. It can be done using a GPS or a measuring tape.

The pixels coordinates of these are easily retrieved since the user is asked to click on the four GCPs present in the image. To compute the real world coordinates the six distances, given in the order shown on the Figure 7.13, are used to solve the following system:

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

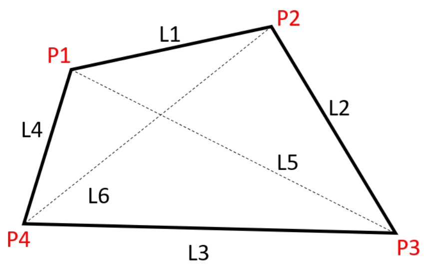
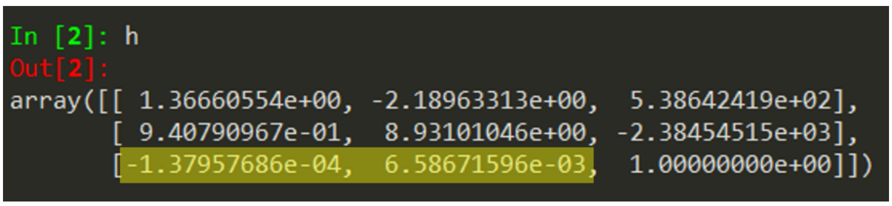


Figure 7.13: Order of the GCPs and distances.

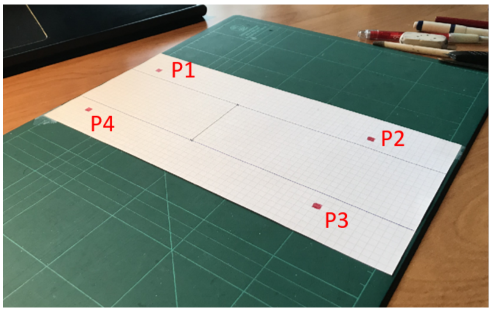
There are 4 GCPs so 8 unknowns and only 6 equations but the coordinates of one point can be set making it solvable. Here the point 4 is chosen to be at (0,0). However, there is still an unknown since one equation can be deduced from the others. To solve this, we also consider that the point 1 and 4 are vertically align. Once the system is solved the real-world coordinates are known as well as the pixels ones so the homogaphy matrix can be computed. The calculation is done using the OpenCV function `cv2.findHomography()`. It takes the two different coordinates of the GCPs as input (images ones and real world ones). Testing the method on a real example find on the RIVeR toolbox site [102] the homography matrix on Figure 7.14 was found. The values for the H_{31} and H_{32} are close to 0 which means that an affine transformation would have been a good approximation in this case.

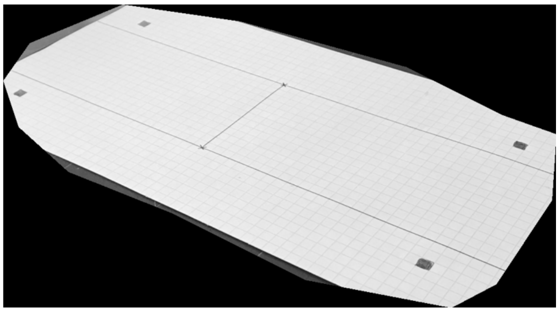


```
In [2]: h
Out[2]:
array([[ 1.36660554e+00, -2.18963313e+00,  5.38642419e+02],
       [ 9.40790967e-01,  8.93101046e+00, -2.38454515e+03],
       [-1.37957686e-04,  6.58671596e-03,  1.00000000e+00]])
```

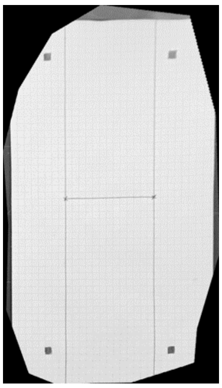
Figure 7.14: Homography matrix found with RIVeR tutorial data.

Using the inverse of the homography matrix the first image of the video of the flow can be transformed in an orthorectified one where the user can later select the section of the river where the discharge computation will be performed. A test of the orthorectification process is shown on the Figure 7.15. This will be the only image on which the orthorectification will be performed. Then the inverse of the homography matrix will be used on the velocity vectors as explain in the beginning to correct their values that have been biased by the perspective of the image. The fact that the image is orthorectified also allows to directly convert the pixels size into meters. Indeed, the orthorectified image is known in meters times a scaling factor so knowing the size of the image in pixels the ratio pixels/meter can be deduced. This is why the water level can also be computed using the orthorectification as explained in the Chapter 5.





(a) Selection of the 4 GCPs and mask steps.



(b) Orthorectified image.

Figure 7.15: Orthorectification image process test.

7.4 Stabilization

Since images can be taken from a video made by a drone a stabilization step is required. The drone movements can induce velocities in images where there was none or adding to the existing ones influencing the results of the PIV analysis. In their Master Thesis Bia and Somers already looked in this problem and performed some tests in real situations to quantify the error induced. To correct it they have written a stabilization method using MATLAB functions. Since the programming language is in Python which does not have all the equivalent functions a new implementation has been written.

The principle is the following: the user chooses four regions on the first frame on the video which present some features that can be tracked, for example 4 GCPs. Then using functions of OpenCV the features in these four regions are tracked in each frame of the video. The selection of the original position of these 4 GCPs in the first frame is shown on the Figure 7.16. OpenCV has several tracking algorithms here the one chosen is a KCF Tracker which stands for Kernelized Correlation Filters. It has the advantage to be fast while maintaining a good accuracy. It does not handle full occlusion well which means that if on one frame one GCP is no longer visible the algorithm may fail. However, this situation will rarely occur. When the regions are tracked in the frames the coordinates of the center of each region is recorded. The stabilization is then done by doing a perspective transformation from the coordinates of the region on the frame n to the coordinates of the first frame.

To compare this method to the MATLAB one developed by Bia and Somers a stabilization on one of their videos has been performed. The original video was 21.56s long, the python stabilized video was 21.7s long since an average frame rate has been chosen but the MATLAB video was only 17.5s long. Regarding the computational time the python method took 2min to process while the MATLAB one took more than 20min.

Figure 7.16: First frame of a video that needs stabilization. The stabilization is performed using the four green rectangles.

Chapter 8

Discharge computation

The discharge computation step in the application is based on the following data - obtained in preliminary steps as described in the previous chapters:

- The cross-section of the river in real world coordinates
- The water level in real world coordinates
- The mean surface velocity field of the river in pixels/s

The method applied to compute the discharge is the following:

1. Define the river's section of interest
2. Determine the image scale (pixels/meters)
3. Convert the stream surface velocity to a depth-average velocity
4. Subdivide the river section in areas where discharge is evaluated based on the local depth-average velocity
5. Compute the total discharge

8.1 Section of interest

The user will select a section of the river where he wants to perform the analysis. He does it by simply selecting two points on the first frame of the video of the river. He must choose the section where the cross-section of the river loaded at the beginning is still valid and where the velocity vectors have more chances to be well computed. For example, where the seeding particles covers almost the entire section on the frames of the video.

He also chooses the number n of subdivisions he wants to make on that section. This will determine the number of velocity vectors retrieve as well. Then the algorithm selects the closest velocity vector in each subdivisions of the section chosen.

8.2 Pixels to meters conversion

The velocity in pixels needs to be converted in meters before the discharge computation. A common way to do this is to have an object of known real dimensions on the image then measuring its size in pixels and deducing the scaling factor between both. In the application this conversion is done in two different ways depending on whether an orthorectification has been performed.

With orthorectification

During the orthorectification a mapping from image to real world coordinates is done using distances between the four GCPs, which are known in both pixel and meter coordinates. Since the velocity vectors are orthorectified after the velocity analysis they are already in meter per second. Here the conversion is direct.

Without orthorectification

The image scale is computed from the knowledge of the river width in real coordinates compared to the pixel one. The latter is known thanks to the section of interest selection.

The real river width can be obtained in 3 different ways:

- From the water level provided by the user. It is used together with the bathymetric curve. The latter is supposed to be known in real coordinates.
- **Angle and Height method:** Here it is the delta in height between the 2 images given in meter that allows to retrieve both the water level and the river width in meter.
- **Adapted Hellebaut's method:** The DEM which is put in the same coordinates systems as the cross-section curve allows to retrieve the water level in meter by looking at the intersection of both curves. Again, using this information and the bathymetric curve the river width in meters can be computed.

8.3 Surface velocity to depth-averaged velocity

During the Large-Scale Particle Image Velocimetry analysis, the surface velocity is computed between each pair of frames in the video of the flow. These vectors are computed on the nodes of a grid defined on the frames (see Chapter 2). To get a resulting field of velocity vectors more reliable a time average of these vectors is taken. This means that for a node (i,j) the average value of all velocity vectors at this node is computed. Once this is done for all frames and all the nodes a mean velocity vector field is obtained. Moreover, in the discharge analysis the component of the velocity that is interesting is the normal to the section studied. So, in the following when the surface velocity is mentioned it refers to the average normal component of the velocity in time.

The surface velocity is known but to be able to compute the discharge the depth-averaged velocity is required. So, assumptions to pass from one to the other must be made. A convenient way to link these two velocities is with a velocity index α which can be expressed as followed:

$$\alpha = \frac{\text{depth-average velocity}}{\text{surface velocity}}$$

Several studies have tried to quantify the value of this coefficient and its link to some features (e.g. water depth, bed roughness) to get a formula allowing its computation. One of these studies was conducted by Hauet [103] on a large sample of gauging realized with a current meter. Using these data, he computes α with the three following different methods, fitting a log-law, a power-law and using the measured surface velocity. According to the published data, the mean values obtained by the three different methods were similar and present few dispersions.

The paper also shows that α increases with the water depth, as α increase with the hydraulic radius Rh. The hydraulic radius is the ratio between the cross-sectional area and the wetted perimeter. Regarding the bed roughness the paper does not draw any conclusion except that for artificial concrete channels the α value is near 0.9 while for natural rivers the value is closer to 0.8. The paper concludes by saying that for water depth less than 2 meters a value of α = 0.8 should be used and for greater water depth α=0.9 is more appropriate. The accuracy of the results will rely on which conditions the analysis has been performed. Indeed, geometry section, wind effect, presence of vegetation, or secondary flow can affect the velocity profile [104].

Another study [105], has explored the drone driven LSPIV method and compared the results to the ones obtained with a classical method, the current meters. In it they used the stabilization method developed by Bia and Somers in their Master Thesis. They show that the surface velocity was always overestimated by the PIV analysis and so needed the application of a correction factor. They compute an optimal value based on their experiments data using an error optimization procedure leading them to a value of 0.86. Then they applied the usual velocity index to retrieve the mean velocity.

This correction factor may come from a problem in the Bia and Somers stabilization method. Indeed, when comparing their method to the one implemented in the application, an observation was that their stabilized video was shorter in time than the original one: 17.5 seconds instead of 21.56 seconds. Since the frame rate is now higher the dt will be lower leading to a greater velocity field. This observation has been made on a different video than the one used in the paper.

In the application there is only a velocity index whose default value follows the rule present in the Hauet article [103]: 0.8 if water level < 2m, 0.9 otherwise. The user can change this value. The correction factor is not implemented since it might come from an error in the stabilization method of Bia and Somers. Future work can investigate this lead to see if a correction factor should be included.

8.4 Bathymetry Subdivision area

Since the number n of subdivisions is given by the user and a way to compute the mean velocity on each subdivision is explained above the only remaining unknown is the area of each subdivision. To compute it, the trapezoidal rule will be used. The points on the bathymetry curve corresponding to the subdivisions of the area can be computed using a simple intersection algorithm. Then with these points and the points of the bathymetry curve the trapezoidal rule can be applied on each subsection. The trapezoidal rule is used because of its simplicity.

8.5 Computation

On each subsection the closest vector of velocity is retrieved. Then for each subsection the discharge is computed as the product between the area and the mean velocity in this area. The total discharge is the sum of these subsection's discharge. The Figure 8.1 illustrates this principle.

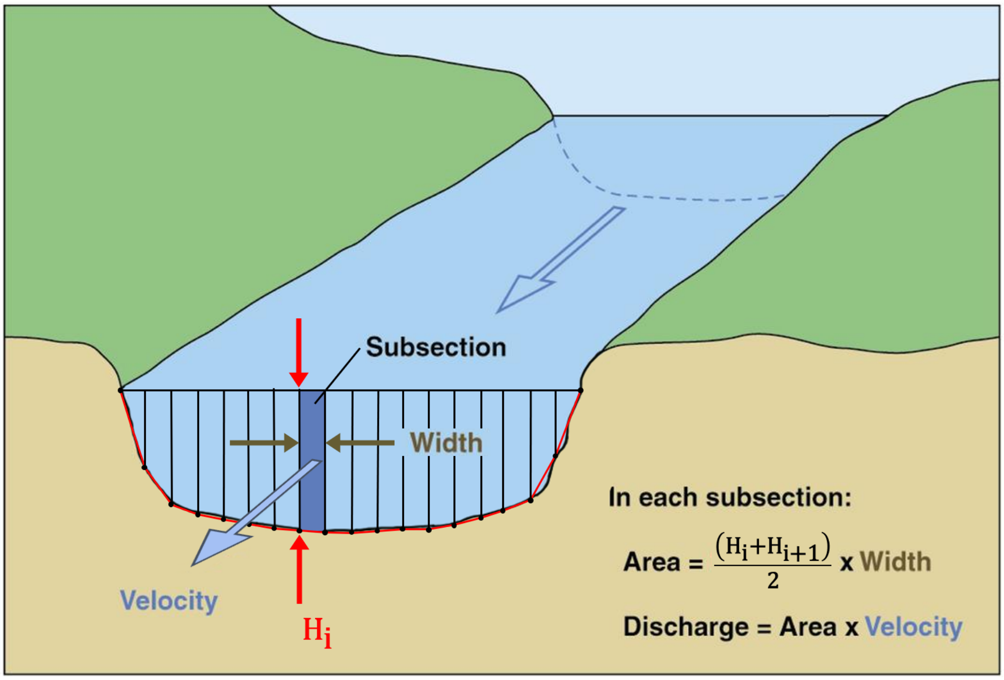


Figure 8.1: Computation of the discharge based on trapezoidal subarea. Figure adapted from [106].

Part III

Overview of the new developed application

Chapter 9

Use of the new developed application

The previous chapters have covered some techniques, choices, influences that have helped to build this application. This chapter will not describe any new technique but rather how the user will go through the application. This includes the different choices that are available, the parameters that can be tuned, the input and output data, etc.

This chapter will be organized in three parts:

1. Description of the global structure of the application with the help of a flowchart.
2. Detailed explanation of the contents and purpose of the different screens. The screens will be grouped by the step they belong to in the analysis. Each group will be described using the following pattern:
   - Aim of the step
   - Description of the different choices available at this step
   - For each choice presentation of the different screens involved: user inputs, buttons use
3. Real example where the application is tested and compared to the results of RIVeR and PIVlab.

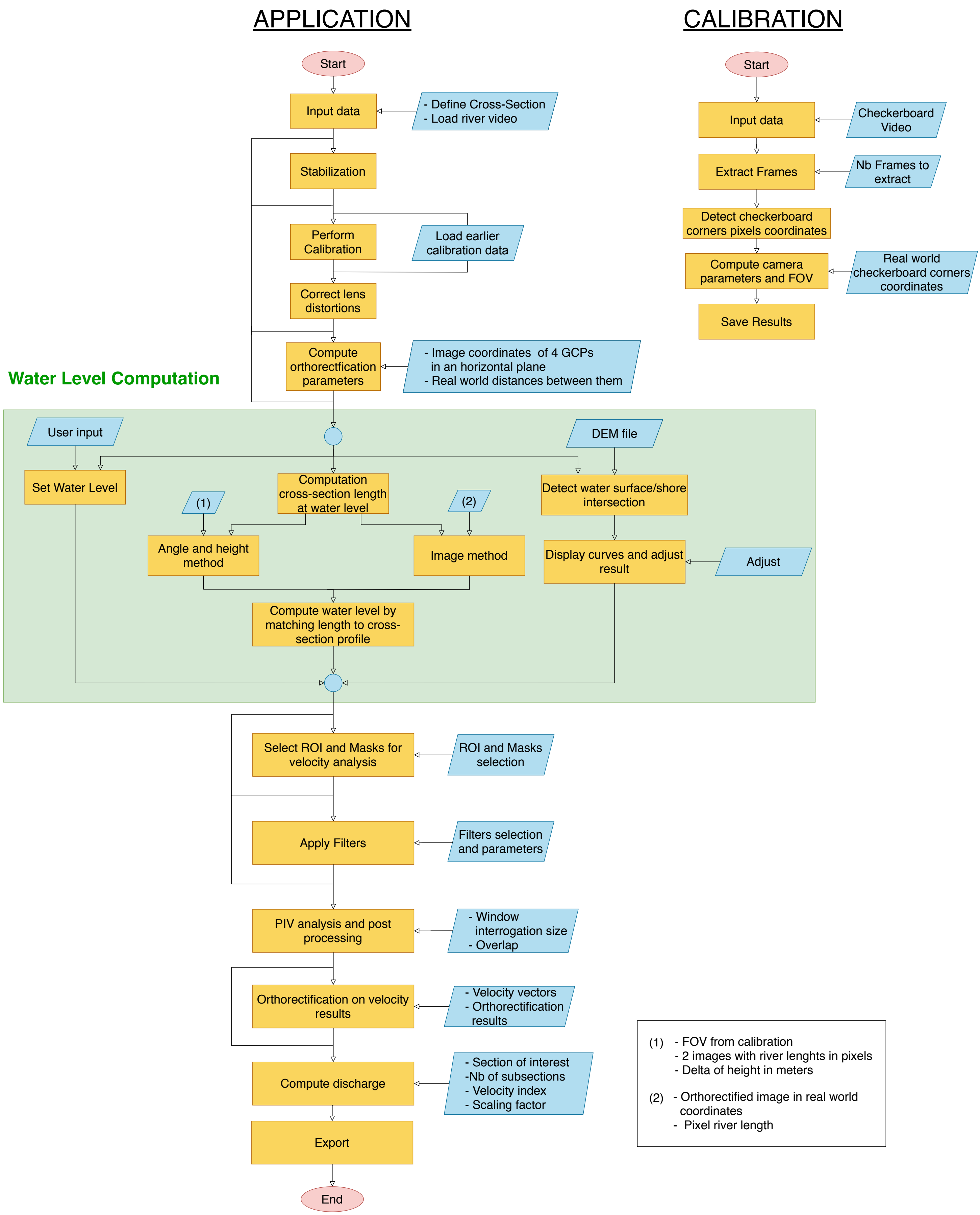


Figure 9.1: Flowchart of the application and calibration.

9.1 Application design

On the Figure 9.1 the flowchart of the application is shown. It depicts the process to get a river's discharge where inputs data are analyzed, optimized through the different steps. Some steps propose several methods to get the same result. The user chooses one according to the river configuration, data quality and availability. Some steps are optional such as the correction of the image or the choice of a region of interest.

9.1.1 Input data

The first part concerns the loading of the input data. This includes the definition of the cross-section where the discharge is computed and the video of the seeded river.

9.1.2 Correction process

The raw video images can be of insufficient quality for direct use in the application. For example, the following cases would lead to spurious results:

- The camera is moving during the recording (inducing non relevant velocities)
- The lens of the camera induce some distortions in the image (distances in image altered)
- The video is not taken along a vertical axis above the water surface (distances in image altered)

To broaden the use of the application and allow these configurations three image improvement optional steps are proposed:

- **Stabilization:** correct the camera movement. It takes the video as input then stabilized the frames by projecting them on reference points on the first frame.
- **Calibration:** correct the lens distortions and compute the FOV. If the calibration was not already performed, the user will have to go through the process depicts on the calibration flowchart also shown on the Figure 9.1. The inputs are a checkerboard pattern video as well as the characteristics of this pattern. The output will be a file where the camera parameters and the FOV are saved. However, if the calibration has been done in a previous analysis the user just loads the file containing the results of this calibration. In both cases this leads to the correction of the lens distortions using the calibration results.

- **Orthorectification:** correct the perspective distortion. It requires the use of GCPs and real distances between them in meters. At this stage of the application only the parameters allowing to perform the orthorectification are computed and stored in the application.

9.1.3 Water level computation

The next process is about the water level computation. The simplest choice is the left one on the flowchart where the value is directly set by the user. Then in the center there are two methods based on the two following steps:

- Compute the cross-section length at water level
- Compute water level by matching this length to cross-section profile

The two methods differ on the first step. The Angle and Height method uses:

- 2 images taken along a vertical axis above the water level surface at different heights
- The delta of height between these 2 images
- Cross-section length at water level in pixels on the 2 images
- The FOV from the calibration step

While the Image method uses:

- Orthorectified image using the orthorectification parameters
- Cross-section length at water level in pixels on the image

Finally, at the right of the flowchart is the adapted Hellebaut's method which requires a DEM of the shore. This method is based on the detection of the water surface/shore intersection. After the computation, a step of adjustment by the user is also proposed.

9.1.4 Surface velocity computation and preliminary steps

Once the water level computation is done two process are available before the velocity analysis. These two processes are:

- Selection of ROI and masks

- Application of filters

The first one is to avoid unnecessary computation and focus on the interesting part of the frames. The second is there to enhance the image in terms of contrast, luminosity, etc. This will help the PIV analysis.

Then the PIV analysis is performed where the user can set the size of the interrogation windows, the overlap between them and the Signal to Noise Ratio (SNR). The SNR is used as a threshold in the validation of the vectors. Once the velocity field on each pair of frames has been computed the mean velocity field is computed. Then if the orthorectification has been performed earlier the velocity vectors are orthorectified.

9.1.5 Discharge Computation and export

Finally, the discharge computation is done. It needs the following inputs:

- Section of interest
- Number of subsections
- Velocity index
- Scaling factor

All inputs are given by the user except the scaling factor which is computed by the application in different ways depending of the choices of the user as explained in the chapter8. Finally, after the computation of the discharge the following features are saved in a 'npy' file:

- Mean velocity fields
- Water level
- Discharge value

9.2 Screen to screen explanation

9.2.1 Bathymetry

The first step is dedicated to the bathymetry definition which is one of the mandatory inputs of the application. The different screens encountered in the bathymetry step of the application are shown on the Figure 9.2. The first one gives two choices to the user:

– Select an existing file

– Create a file

According to his choice he ends up with one of the two screens below the first one on the Figure 9.2.

For the first choice the selected file must be a text file with each row being of the following format: x,y where x is the distance across the river and y the elevation of the point. The values must be in meters and the file must contain the coordinates of a single cross-section.

The other choice is to create a file. The user will get four propositions as it is shown on the right of the Figure 9.2. These propositions correspond to four possible cross-sections shapes: rectangular, trapezoidal, parabolic, and free. Once the user has decided, he will end up with a screen similar to the one displayed on the Figure 9.2 where the trapezoidal choice is shown. On these screens he must provide the parameters values that characterize the selected shape as well as a reference point coordinates illustrated in red in the schematic. Then the coordinates points of the cross-section are computed and stored in a text file which has the name given by the user. Since the result is saved it can be used later with the selection file option.

If the user has chosen to load a bathymetry file it will be displayed on the screen as shown on the left part of the Figure 9.2 to allow the user to check if he has selected the correct one.

9.2.2 Loading of the video

The second required input is the video of the flow that will be used for the velocity analysis. The corresponding screen is display on the Figure 9.3. The user may decide to limit the analysis to every Nth frame - skipping intermediary frames to improve speed and accuracy. By default, all frames are analyzed (N=1). He can also choose the part of the video that is relevant for the analysis by specifying a range of time. To avoid the selection of a time outside the range the maximum value is set to the duration of the video. Then the user must say if he wants or not perform a stabilization of the video. The instructions to perform the stabilization are provided at the bottom of the screen to guide the user. This includes the order of the control points that will be used and the command to do/undo the selection and to launch the stabilization. If the user uses the stabilization the first frame of the video will be displayed, and the user just need to select four regions that will be tracked.

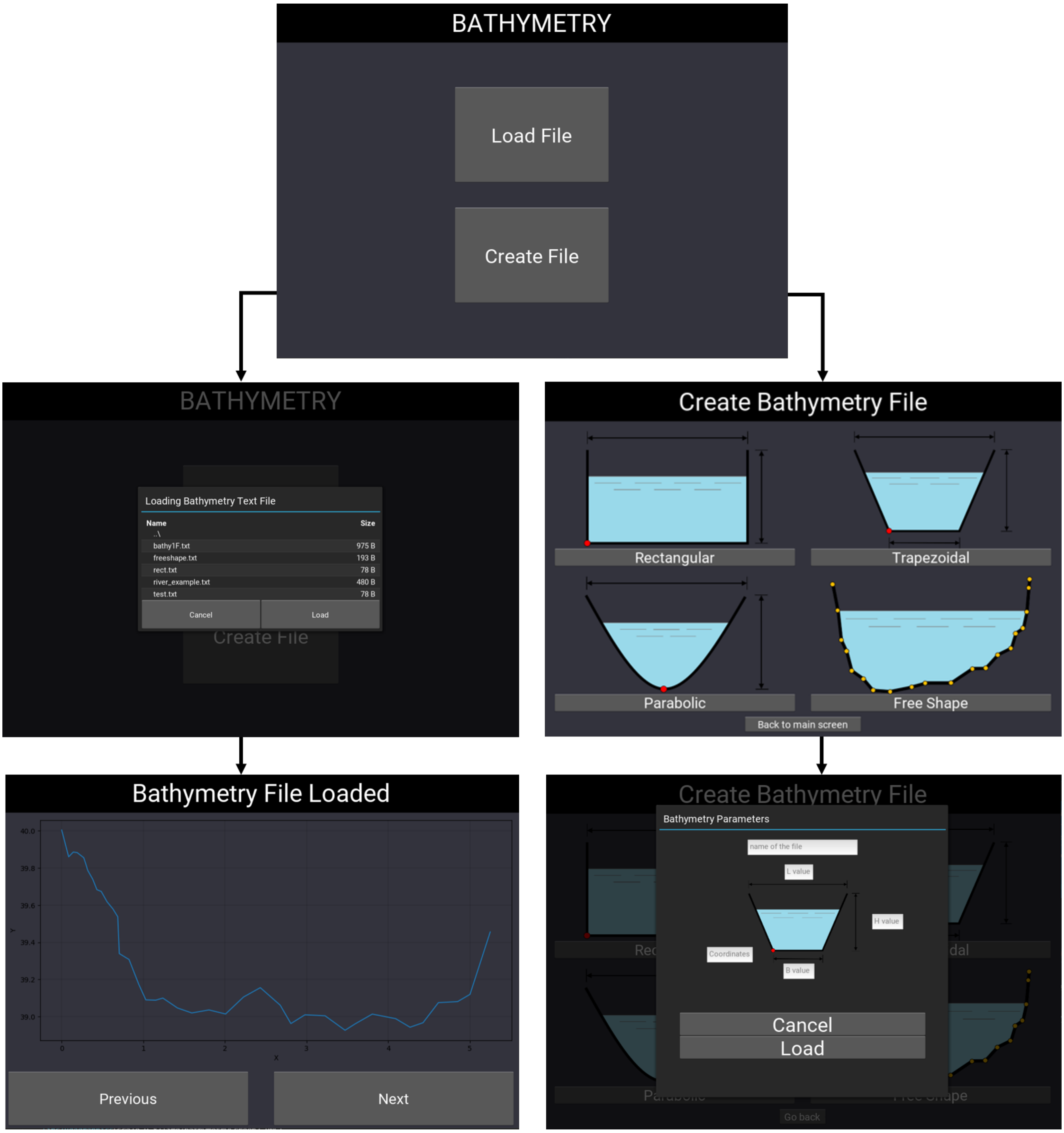


Figure 9.2: Screens of the cross-section definition part. The left screens are dedicated to the Load option, the right ones to the Create option.

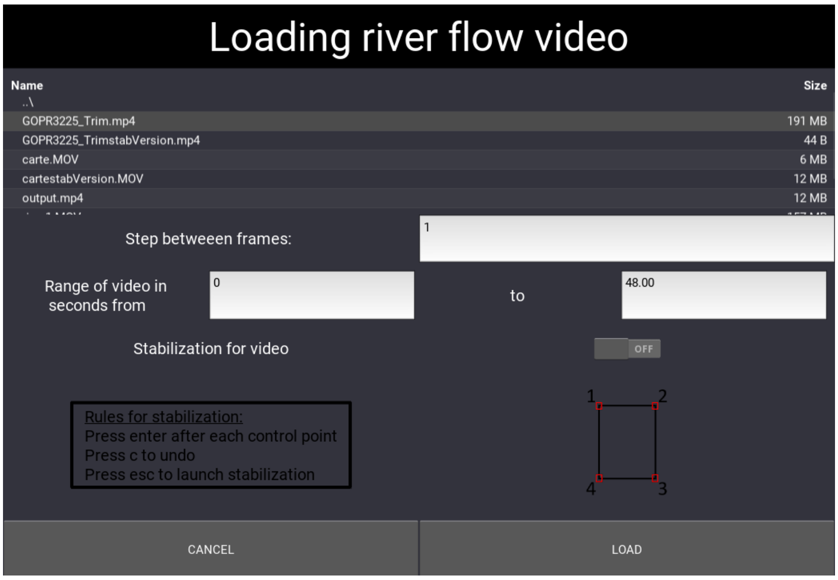


Figure 9.3: Input video screen.

9.2.3 Calibration

The next screen is dedicated to the calibration of the camera. The calibration allows to compute the Field of view of the camera as well as the intrinsic and extrinsic parameters of the camera. The camera parameters allow to undistort the images and the FOV is useful in the Angle and Height water level method.

In this step the user gets three choices:

- Load a file with the calibration parameters.
- Perform the calibration.
- Skip this step

The first option assumes that the calibration of the camera has already been done before. So, the user can simply load the file where the results were stored. The third option sends him directly to the next step. The second option is the case where the camera must be calibrated. In this case the user gets the screen shown on the Figure 9.4. The flowchart of the calibration is also shown on the Figure 9.1. The required inputs for the calibration step are the following:

- Video file of a checkerboard pattern
- Number of frames to use

– Name of the save file

– Checkerboard dimensions

– Size of the square of the checkerboard in mm

The number of frames is the number that will be extracted from the video and on which the calibration will be based, a recommended number is 10.

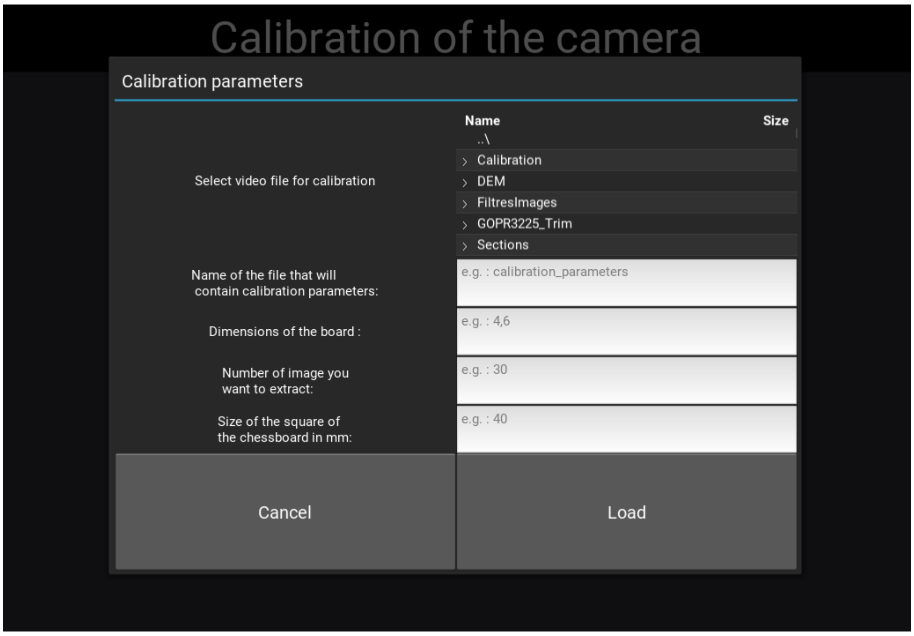


Figure 9.4: Calibration screen.

9.2.4 Orthorectification

The Orthorectification screen can be seen in the Figure 9.5. As for the calibration step, the orthorectification can be skipped by the user.

The aim of this step is to compute the parameters that characterize the orthorectification such as the homography matrix. These parameters will be used in the Image method for the water level computation if it is chosen by the user and in the velocity analysis to orthorectified the velocity vectors.

The different inputs for this step are managed by the three different buttons:

- Select Ref Points
- Set real Dist
- Apply mask

The first one will provide the pixels coordinates of the GCPs by asking the user to click on the four GCPs present on the image. Once the selection is done the user must push the 'Register' button to validate it. The second button allows to enter the real distances in meters between these GCPs. Pressing the button opens the popup shown on the Figure 9.5 where the user enters the data which must follows the order shown on the schematic.

Finally, the last button allows the user to apply a mask which will disregard everything outside. The user closes the mask by doing a right click. The selected region will be orthorectified and use later for the selection of the section of interest where the discharge computation will be made. This mask helps to only keep the river part of the image and avoid the images filled with shore surroundings. An example of mask and GCPs is illustrated on the Figure 9.5.

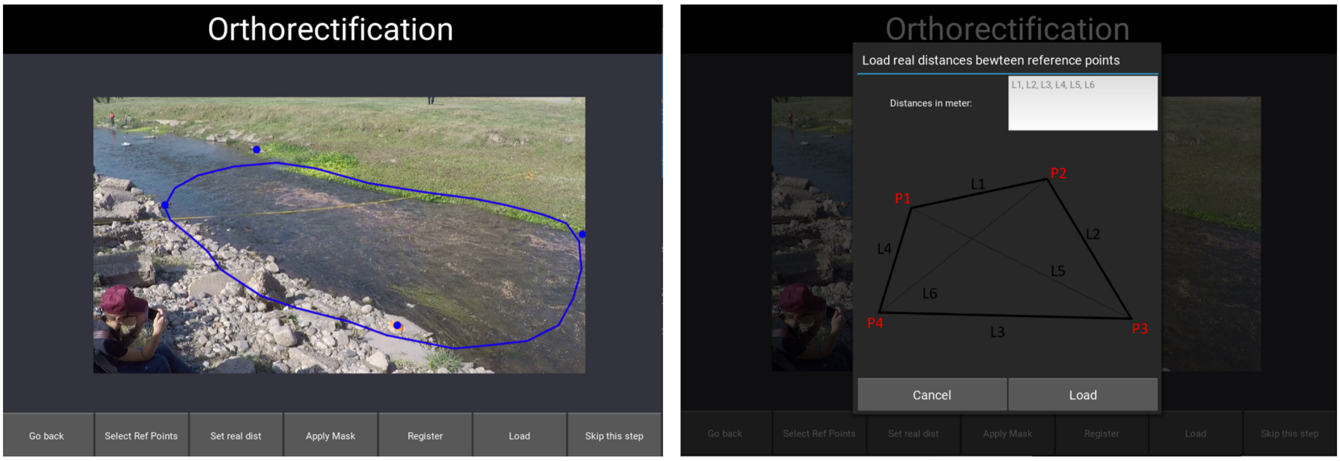


Figure 9.5: Orthorectification screen: Reference points and mask example on the right and real distances settings popup on the left.

9.2.5 Water level

The next step is the determination of the water level. The flowchart (see Figure 9.1) can give some insights about their organization. In the application screen (see Figure 9.6) four options are available which will be covered in the next subsections.

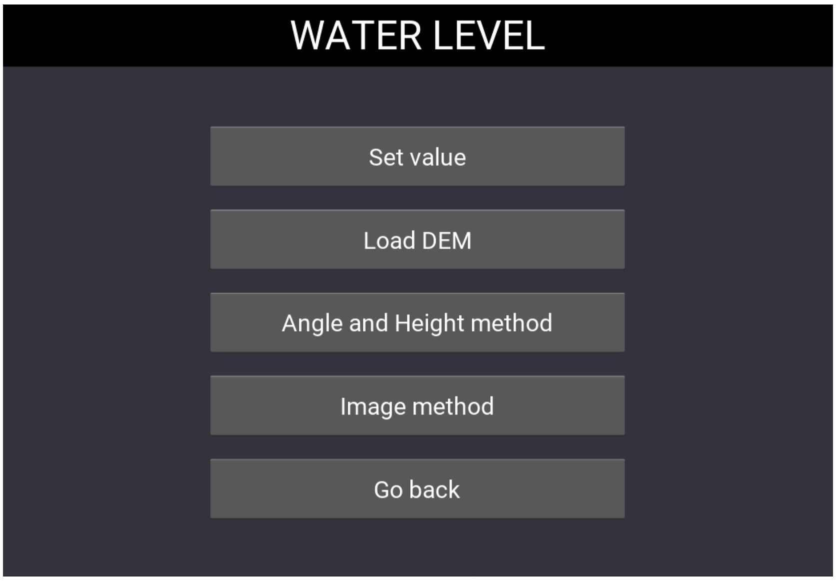


Figure 9.6: Choice of water level method screen.

Set value

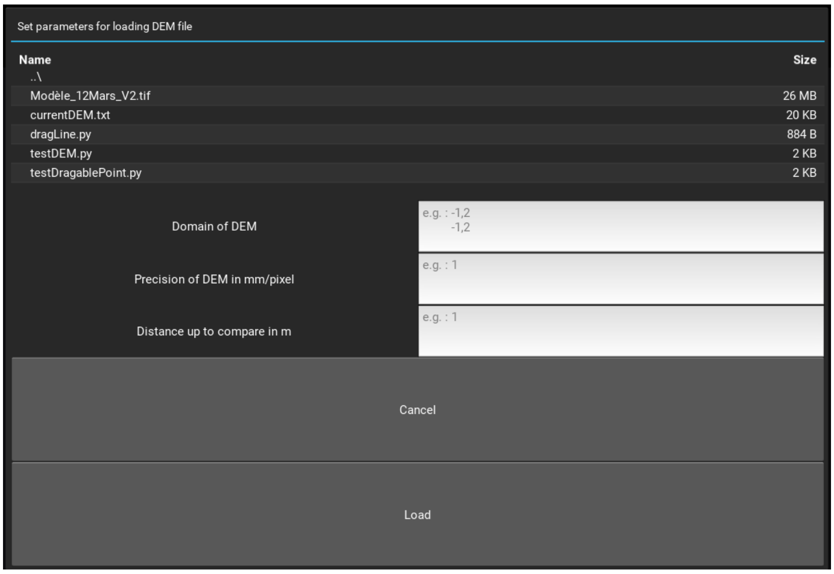
The first one is simply to set the value of the water level. The user must be aware that in the application the water level value corresponds to the elevation of that level not the depth. The elevation must be taken in the same reference as the bathymetry data given earlier.

Load DEM

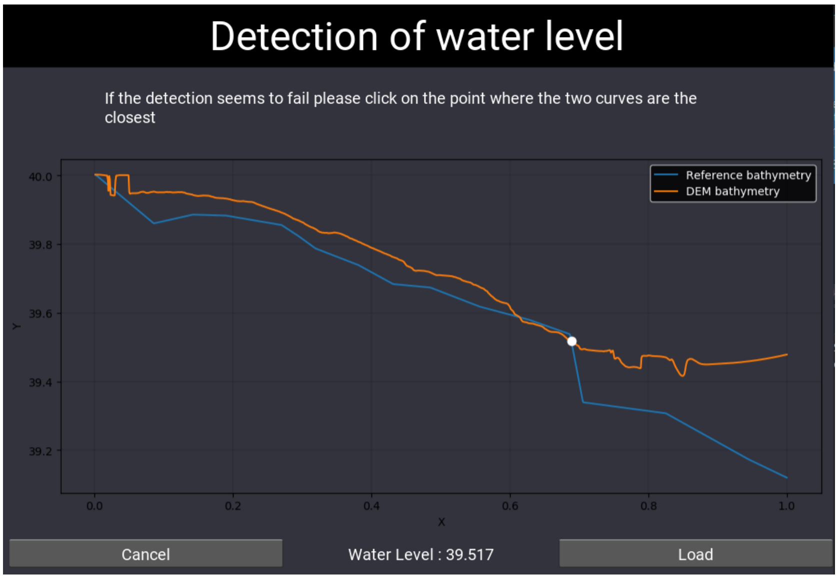
The second option is to use the modified Hellebaut's method. This method uses the cross-section and a Digital Elevation Model (DEM) of the shore computed with the SfM photogrammetry technique. The only unknown is the DEM. By choosing this option the user will go to the screen shown on the Figure 9.7 where he will have to give the following information:

- Selects a tiff file containing the DEM
- Boundaries of the region modeled by the DEM in meters
- Precision of the DEM in mm/Pixel
- Distance upon which the DEM should be compare to the real bathymetry

The second input is there to have the coordinates of the region model by the DEM which must be in the same reference as the bathymetry file. The boundaries need be provided in x and y directions and in meters. The x coordinate is the one across the river while the second one is along the shore. The x range is given in the first row separated by a comma then the y range is given on the second row also separated by a comma as indicated to the user in the application.



(a) Modified Hellebaut's method parameters.



(b) Modified Hellebaut's method results display and adjustment screen.

Figure 9.7: Modified Hellebaut's method screens.

The last input is there to limit the search near the shore which is the interesting part for the method.

Once all the parameters are provided the result is displayed on the screen of the Figure 9.7. On this screen the user has the possibility to adjust the result by clicking on the graph if the computation has failed. This way even if the computation fails the user can rectify it easily.

Angle Height method

This method and the following one are both based on the following principle: compute the cross-section length at water level in meter and with the bathymetry curve deduce the water level. They differ in the way the cross-section length at water level is computed.

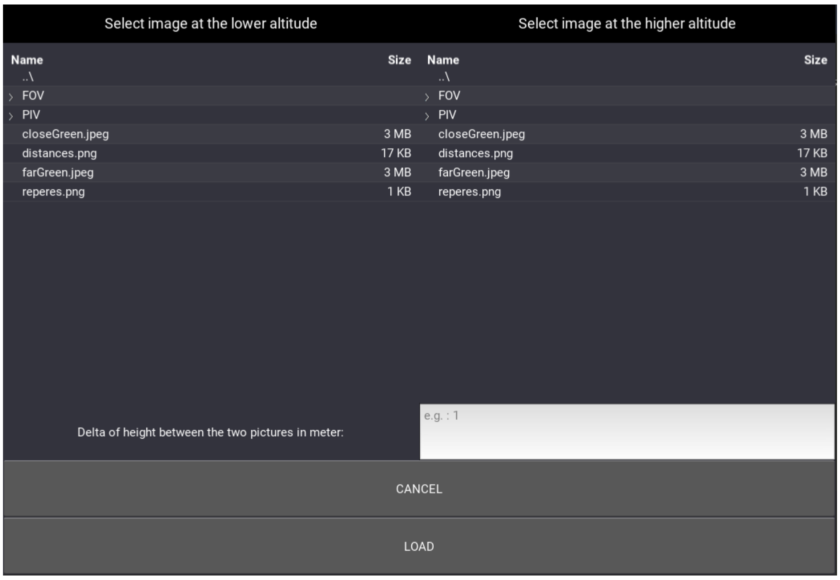


Figure 9.8: Angle and Height method parameters screen.

If the Angle and Height method is chosen the user will get the screen presented on the Figure 9.8.
This method required the following input:

- Two images taken by a drone at different heights along a vertical axis
- The ΔH in meters between the two images
- The FOV from the calibration step

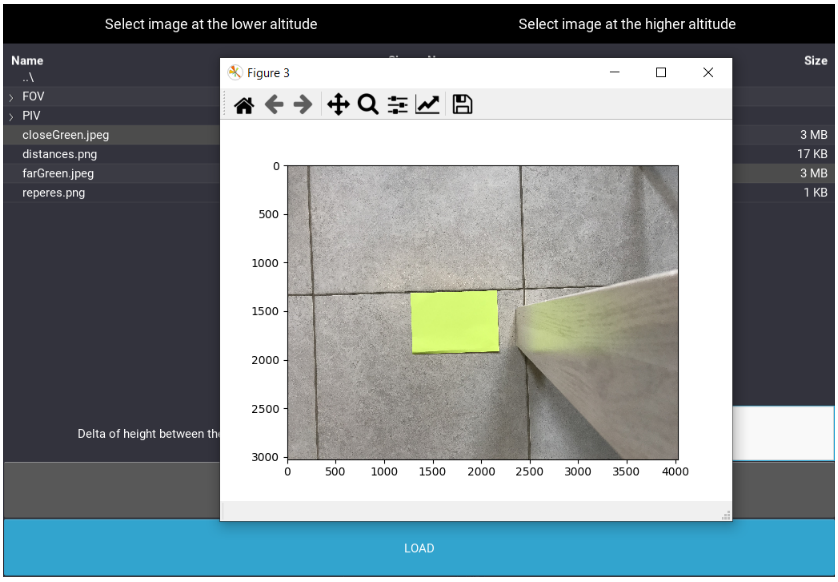


Figure 9.9: Angle and Height method images pixel coordinates input screen.

– Cross-section length at water level in pixels on the two images

In the application the user first selects the two images files and gives the ΔH value. Then after pressing the load button, the two images will be displayed one after the other and the user will have to click on leftmost and rightmost point of the cross-section length at water level.

These inputs allow the computation of the cross-section length at water level in meters which leads to the water level. These two values are displayed to the user as shown on the Figure 9.11.

Image method

This method requires that the orthorectification parameters have been computed earlier. Here the user gets the screen displayed on the Figure 9.10 where the image displayed is an orthorectified one. Then the only input required is the pixels cross-section length at water level given by the user which simply clicks on two points. Then the conversion of the length in meters is done. From here the computation is the same that in the previous method. Finally the results will be displayed in the same way as before (see Figure 9.11).

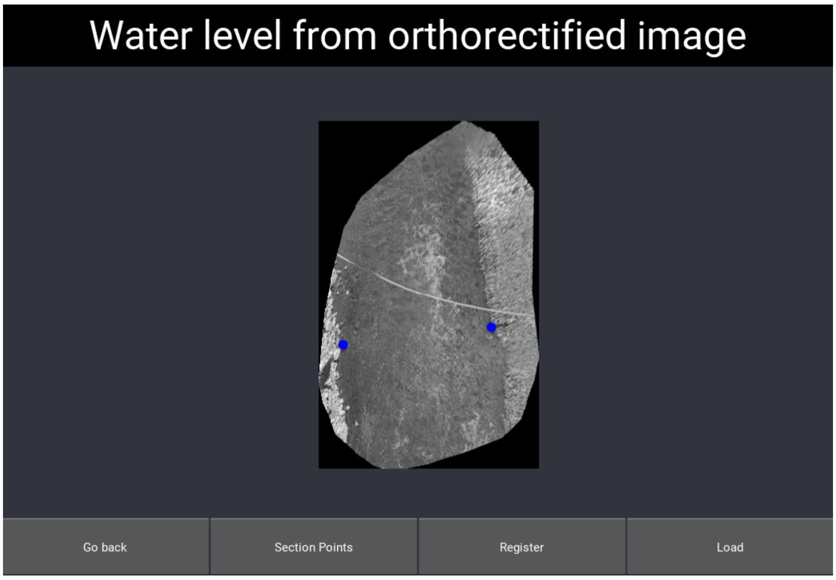


Figure 9.10: Image method screen. Blue points are the user input which determines the section width in pixel.

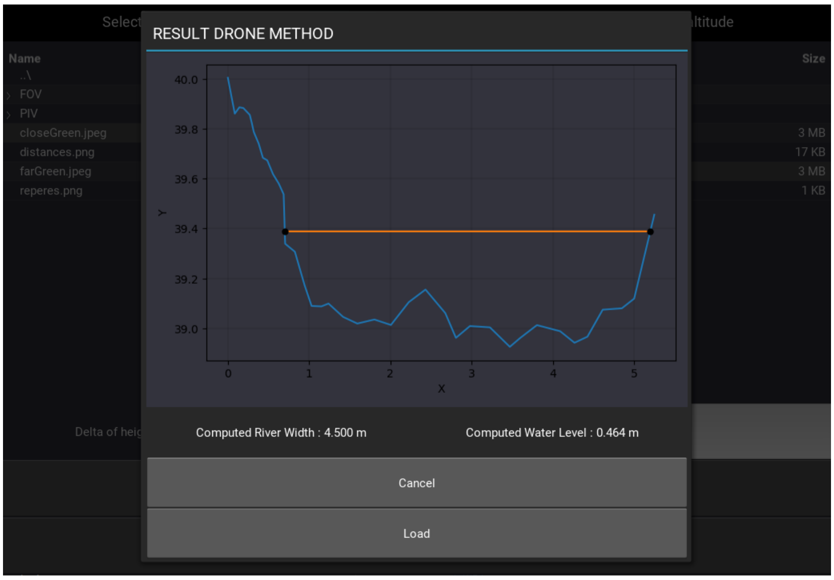


Figure 9.11: Screen of the results of the Angle and Height method and Image method

9.2.6 Region of interest and masks

The next part is dedicated to the selection of a region of interest (ROI) and the application of masks of the video frames. The aim of this step is to select the interesting part of the frames where the velocity analysis will be performed. Keeping only this part will speed up the process by avoiding unnecessary computation. This step can be skipped by the user.

An example of ROI and masks selection is displayed on the Figure 9.12. To perform this step and obtain a similar result the user has the two following possibilities:

1. Selection of a rectangular ROI
2. Exclusion of the unwanted regions in the ROI or the image using masks

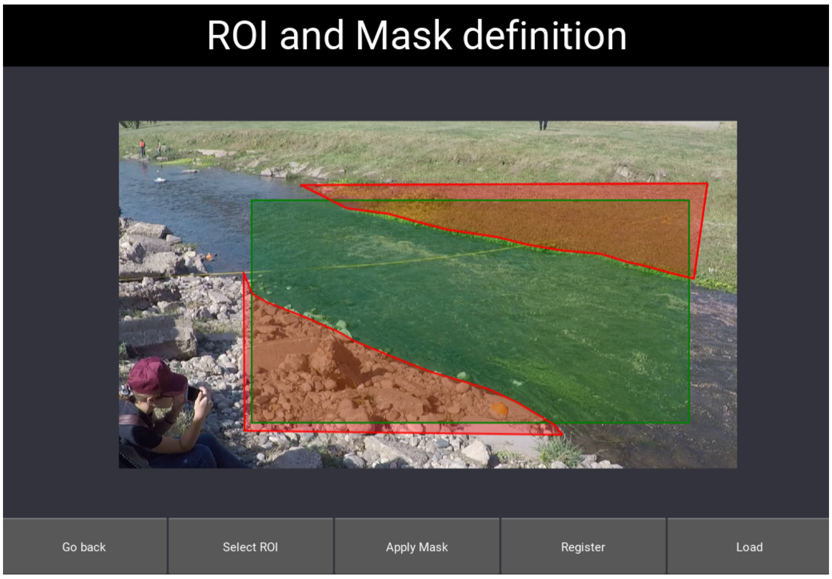


Figure 9.12: ROI and masks example. The green rectangle is the ROI and the red shapes are the masks.

For the first one the user pushes the 'Select ROI' button then clicks on the screen where the top left point of the rectangle will be and click again but where the bottom right point should be. After the first click to help the user a rectangle will be displayed where the second point is the mouse position. If the user is satisfied, he needs to press the 'Register' button to validate his ROI choice. The user does not have to select a ROI, he can decide to keep the all image.

To discard the parts of the ROI or image that may not be relevant for the velocity analysis, the user can apply some masks. To apply one mask the user pushes the button 'Apply mask' then clicks around the zone that needs to be discarded, the contour is drawn as clicks are made. To close the contour, just right-click. To validate a mask the button 'register' must be pressed. Everything inside the mask will be discarded. Several masks may be needed in which case the user just repeated the process. The user can also just select a ROI without applying any mask.

9.2.7 Filters

Before the velocity analysis a filter step, which can be skipped by the user, is also proposed to help the velocity analysis. This step aims to apply filters on images to help the PIV analysis by removing noise, adding contrast, etc. The images on which the filters will be applied are either the original ones of the video or the ones resulting of the previous step. This latter being the ROI selected without the masks. The filters are not proposed before the ROI and masks step because the parts disregarded can have a non-wanted influence on the applied filter.

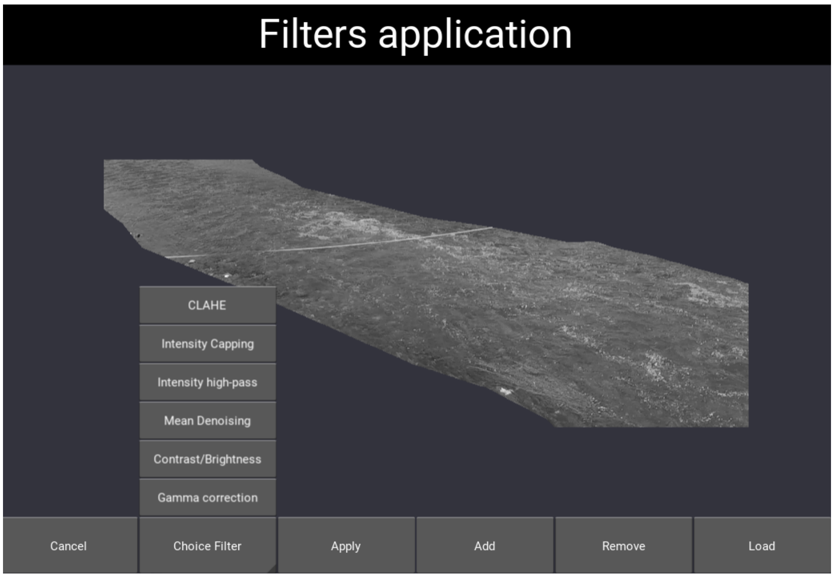


Figure 9.13: Filter selection screen.

The image displayed on the screen, see Figure 9.13, is the first image that will be used in the velocity analysis. It is there to see the effect of each filter on the image and choose the more appropriate ones. On the Figure 9.13 you can also see the different filters that the user can applied which appears when the user press the 'Choice filter' button. These are the ones presented in the chapter 7. The other buttons involve in the filters process are the following:

– Apply

– Add

– Remove

The first one is just to see the effect of the filter on the image, the second one is to register it and the third is to remove the most recent added filter. This allows the user to test several combinations of filters to see which one works the best on the image. For each filter, the user will have the possibility to tune the filter parameters or used the one set by default which are either common values or values computed using the image features. An example of these parameters settings is shown on the Figure 9.14.

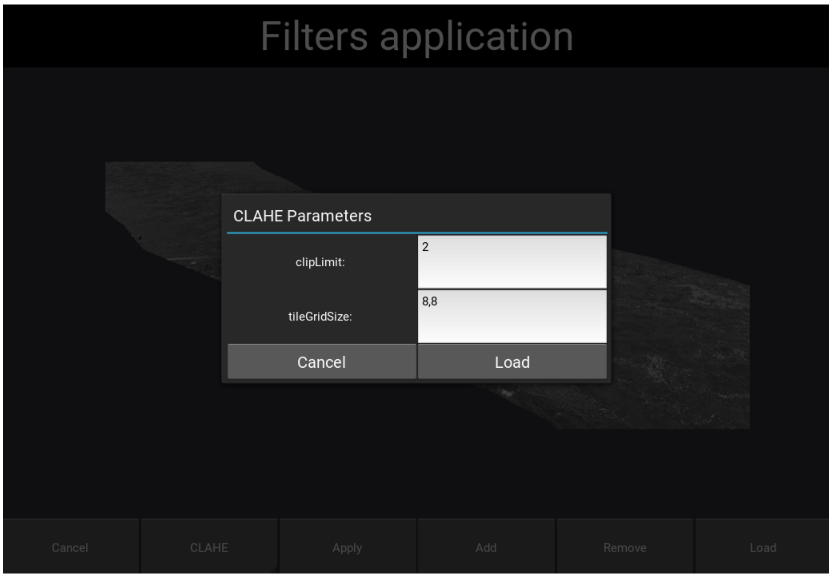


Figure 9.14: CLAHE filter parameters screen.

9.2.8 PIV analysis

The next step is the computation of the velocity field by the PIV technique using the FFT transform. The user will have a popup window as shown on the Figure 9.15 in which the parameters require for the PIV analysis are asked. Theses parameters are the interrogation window size, the overlap between them and the SNR used for the validation step. Default values are proposed but the user can change them.

Since the PIV analysis can take some time to process a loading bar that you can see on the Figure 9.16 will be displayed to show the advancement of the analysis to the user. The validation and interpolation are also performed here. The validation is based on the value of the SNR given earlier while the interpolation is done using a local mean method. Once the analysis is finished the resulting mean velocity vectors are displayed on the first image used for the analysis as shown in Figure 9.17.

In this last screen the button 'Selection of section of interest' will take the user to the next step.

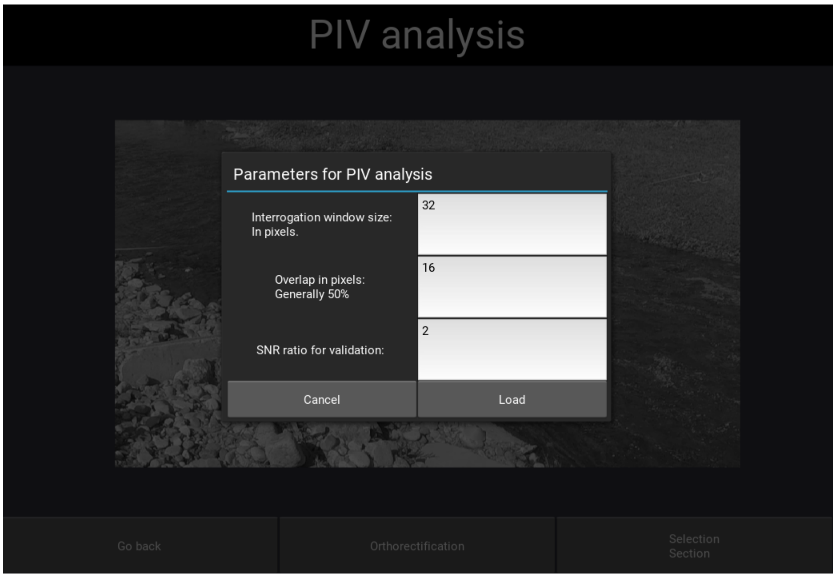


Figure 9.15: PIV parameters screen.

9.2.9 Section of interest

Once the velocity analysis is done the next step is to select the section where the discharge analysis will be performed. This section of interest is the part of the section occupied by water. The user must push the 'Section points' button before the selection then the 'Register' button to store it. An example is displayed on the Figure 9.18. The user must select the section where the velocity vectors have more chances to be well computed and where the cross-section profile loaded at the beginning of the application is still valid.

The image on which the user selects the section is the orthorectified one when an orthorectification has been performed. In the other case it is simply the first frame of the video of the river.

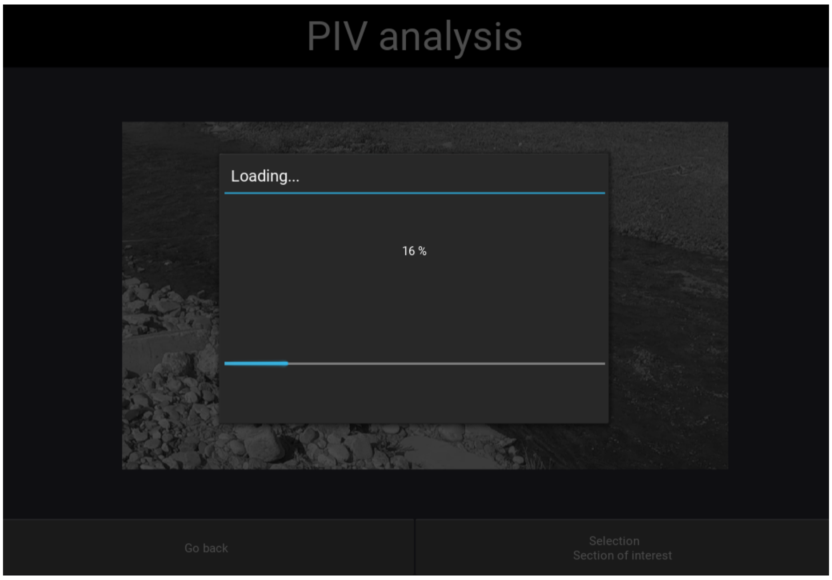


Figure 9.16: PIV screen loading bar.

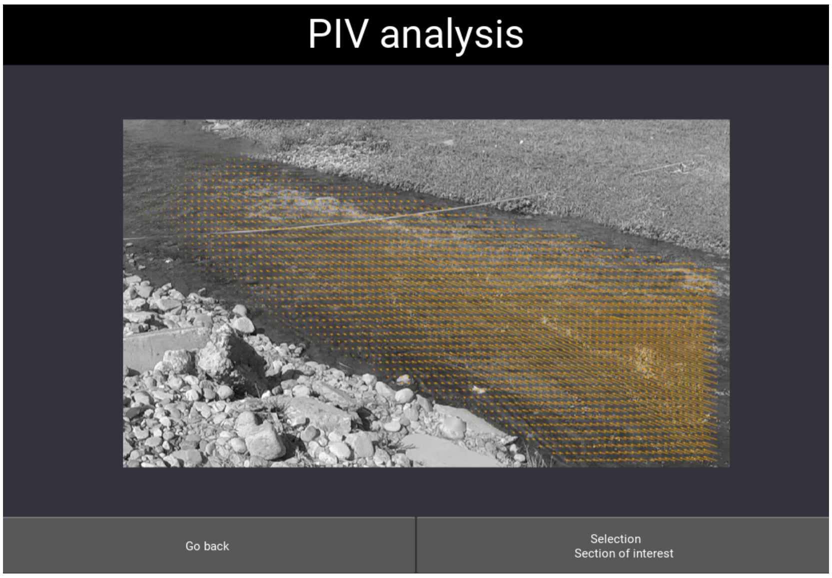


Figure 9.17: PIV screen velocity vectors.

9.2.10 Retrieving of the velocity vectors

Once the section has been selected the user will end up with the screen of the Figure 9.20. If the section points of the previous step are not aligned horizontally a rotation is performed on the image to make it horizontal. The velocity vectors are also rotated. The goal is to put the section of interest horizontal such that in the discharge computation only the vertical component of the velocity is used.

On this screen the user must provide the number of subdivisions of the section he wants to use. This number will also determine the number of velocity vectors retrieved on this section. Then by clicking on the 'see velocity vectors' button the velocity vectors retrieved on that section will be displayed as shown on the Figure 9.19.

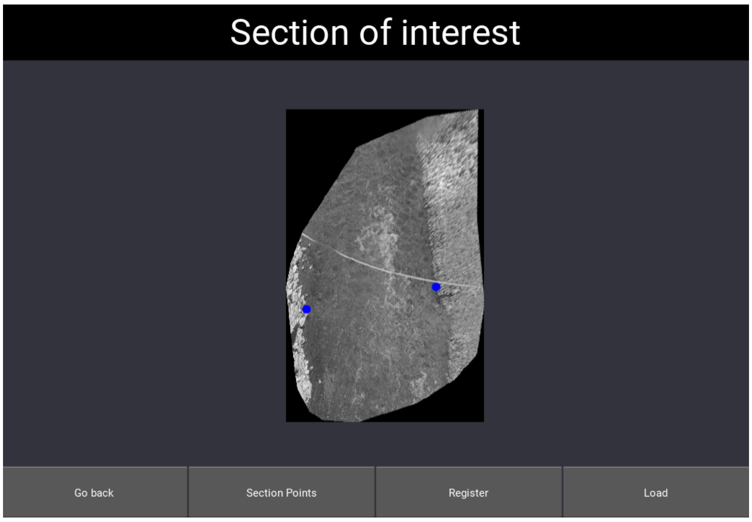


Figure 9.18: Section of interest selection screen. Blue dots are an example of section chosen by the user.

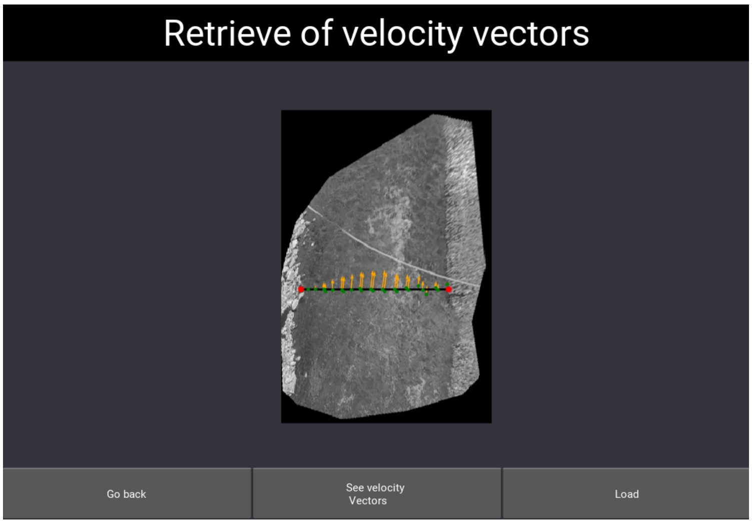


Figure 9.19: Retrieve of velocity vectors of the section of interest.

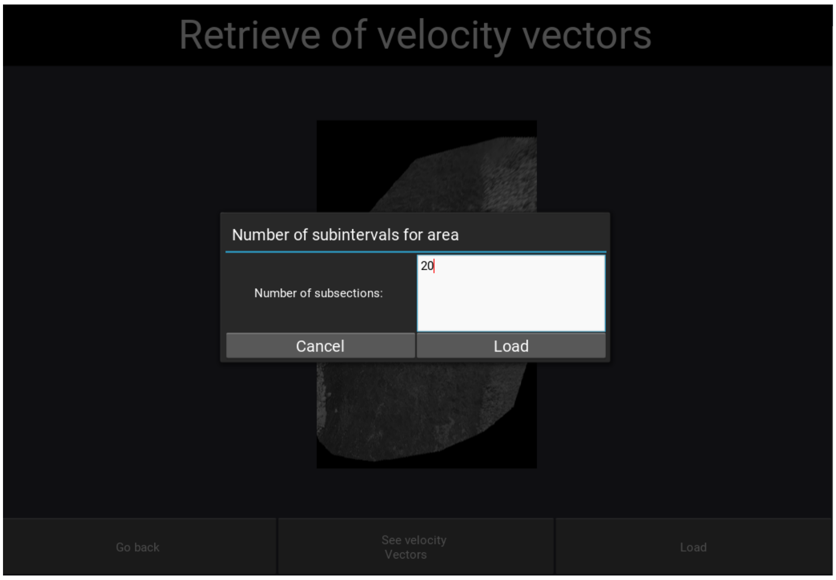


Figure 9.20: Nb of subsection popup.

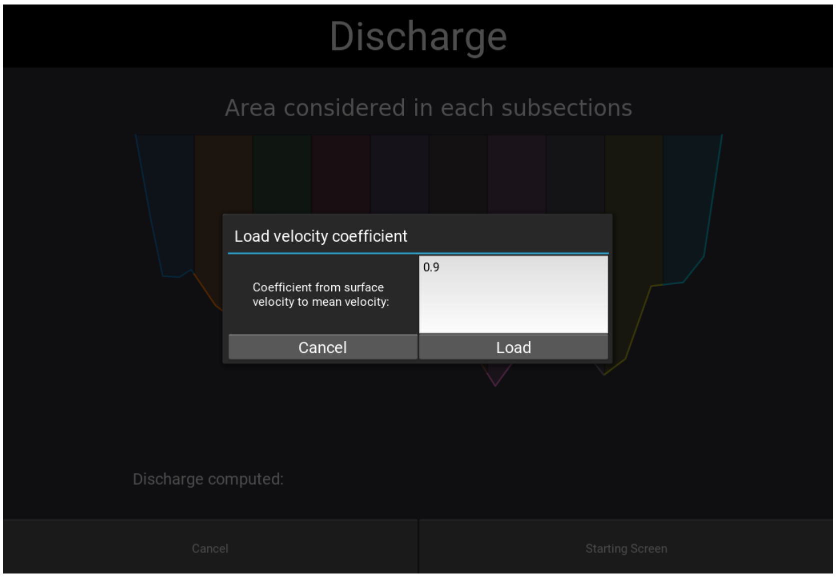


Figure 9.21: Discharge screen. Index of velocity popup.

9.2.11 Discharge

The final step is the computation of the discharge. The first step is to go from velocity in pixels to meter per seconds by applying a scaling factor. The way it is computed is explained in the chapter 8. Then to switch from the surface velocity to the mean velocity the user must provide the velocity index. In the application a default value is provided based on water level value (see chapter 8).

For the discharge computation the velocity component in the flow direction is used. The area of each subdivision is also computed using the trapezoidal rule. These parameters help to find the discharge value which is displayed to the user as shown in the Figure 9.21. The area of the subdivisions of the section is also shown.

9.3 Use on a real example

RIVeR is an application that has been presented in the Chapter 4 which works with PIVlab to provide the computation of a discharge. To evaluate the application developed in this Master Thesis a comparison between the result obtain by them and this application has been made on an example. This example was provided on the website of RIVeR in the tutorial done by Antoine Patalano. The data on which the computation have been done here comes from this tutorial. For this example, they have recorded the flow of the Suquia River in Argentina from the shore. The different steps taken in the application for this example will be explained below.

The following steps were taken:

1. The cross-section was first converted from an Excel file to a text file following the convention supported by the application then loaded. The graph of this cross-section can be seen on the Figure 9.22.
2. The video of the flow was loaded and the range [0,15] seconds chosen to match their choice.
3. The calibration step was skipped
4. The orthorectification was performed using the different distances provided, these are shown on the Figure 9.22.
5. The water level was set to the same value used in RIVeR. It corresponds here to the maximum point of the cross-section points given.
6. The selection of the region of interest and the masks is represented on the Figure 9.22.
7. No filters were applied.
8. The velocity analysis was performed with the default settings. The resulting field is shown on the Figure 9.22.

9. The selection of the section of interest was done close to the center of the image.
10. The number of subsections chosen was set to 30. The velocity vectors on this section of interest are shown on the Figure 9.22.
11. The discharge computation was done with the same velocity index $\alpha = 0.85$ and a value of $1.46\,m^3/s$ was computed (see Figure 9.22).

The application gives a value of $1.46\,m^3/s$ while they obtain $1.39\,m^3/s$ using RIVeR and PIVlab. This shows that the developed application is sufficiently robust, flexible, and accurate enough to provide results in accordance to other similar state-of-the-art programs, based on the same input data. Further work is required to benchmark the application using a wider range of real-world cases.

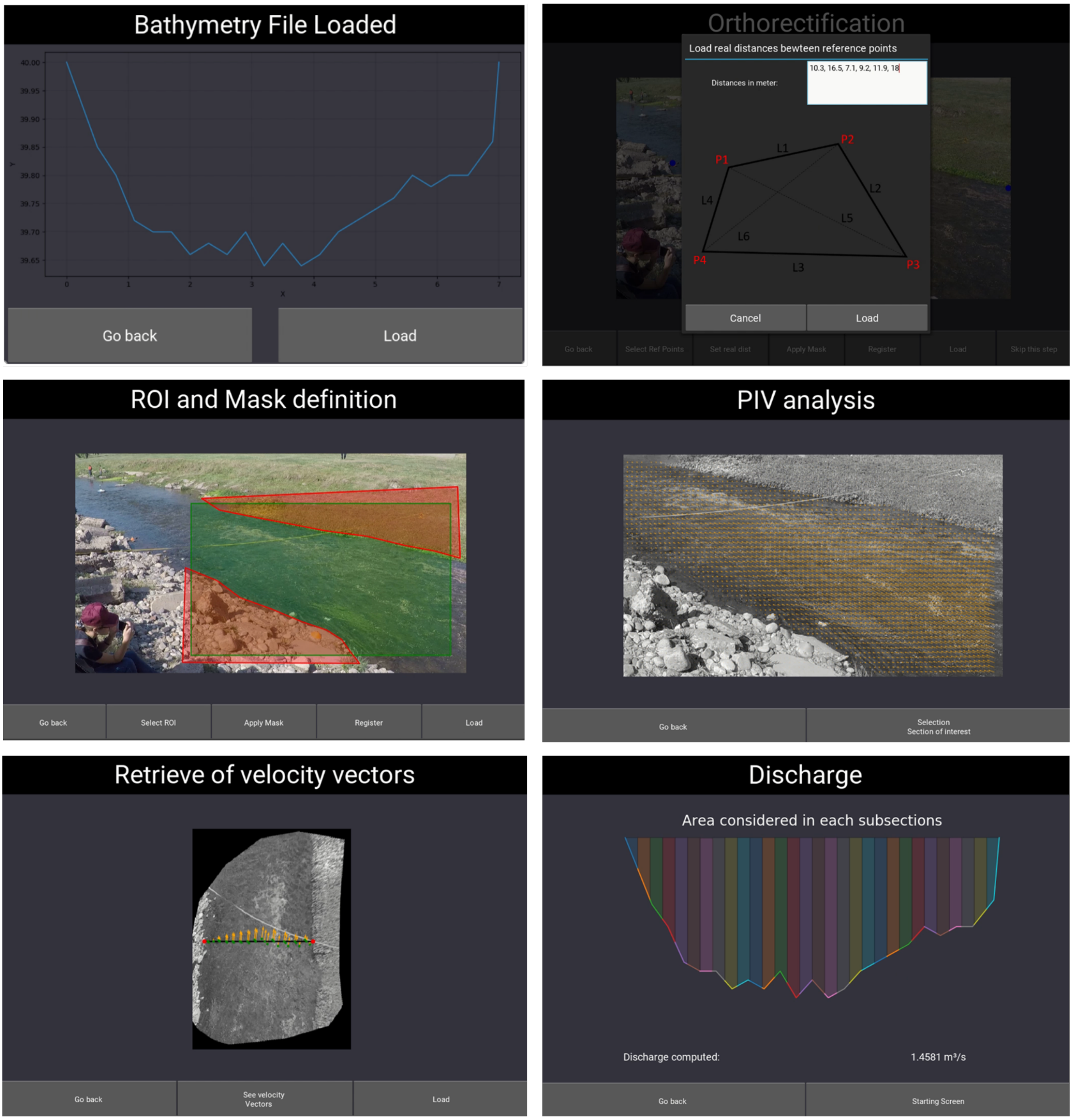


Figure 9.22: Screens of the application during the discharge analysis on the RIVeR example.

Chapter 10

Evaluation and room for improvement

The aim of the application is achieved which is performed a discharge computation providing results in accordance with similar state-of-the-art programs as illustrated in the last chapter. The application is a stand-alone one where features from the pre-processing step to the post-processing one are implemented with in addition several methods for the water level method. This latter is rare in the similar existing programs where the value must either be given or computed using a unique method.

Further work is required for several reasons. Test the application in more real-life examples. Broaden its scope with additional features. Improve its performances by reducing the computational time. To highlight these points and show the application abilities this chapter will be divided in three parts. The first one is a feedback on the ease of use of the application while the second and third parts cover its strong points and flaws.

10.1 Usage Feedback

The application was developed with usability in mind. The objective was reached thanks to:

- Clear captions on buttons, screens, and dialogs
- Detailed instructions added on some screens
- Logical sequence of the different steps
- Visual aid in selection steps by drawing the shapes during the selection (e.g. ROI and Masks)

- Unlimited attempts for the selection steps since only validate when Register button is pushed
- Progress bars display during lengthy processing
- Reactive display when resizing windows
- Possibility to going back at each step

These features make the application easy to use. However, a validation of the input data could not be implemented in the available time frame. Inconsistent data input can still result in an abortion of the process. This is certainly a candidate for improvement.

10.2 Strong points

Several features regarding the image processing are implemented in the application:

- Selection of a region of interest (ROI)
- Application of masks
- Six filters available
- Orthorectification
- Stabilization
- Correction of distortion using a calibration step

The first two features allow to speed up the process since it reduces the size of the images used for the velocity analysis. The filters can enhance the velocity analysis. In the application they can be combined and tested on the images before being added. The orthorectification step allows the images to be taken other than vertically and as mentioned by the RIVeR software authors [63] it is lacking in most of the PIV applications. A stabilization method is also available which is required when working with video taken from a drone. This stabilization method has also the advantage to be faster than the one developed by Bia and Somers in their Master Thesis. Moreover, a possible error in their implementation may have been spotted which could explain the need of a correction factor on the surface velocity values. This need to be more investigate in future work. Finally, a calibration step is implemented which allows to correct the distortion than a camera may induce in the image.

In addition to these features the application offers several methods for the water level computation. Indeed, there are three methods for computing the water level, a fourth choice is to simply set the value. One method is a new implementation of the Hellebaut's method where the user has the possibility to directly correct the result at the end of the computation. The other methods are based on retrieving the water level from the river width. One using two images taken at different height by a drone and the other using the orthorectification step. The principle of the method has been tested using simple examples as measuring sheet of paper width. However, no measurement on river has been done to check it feasibility in real condition. This should be assessed in future work. Regarding the bathymetry the application offers the possibility to create new files rapidly that can be reuse later.

10.3 Flaws

As mentioned earlier there is no error management in the application. This means that if the input given by the user is in the wrong format or leading to an error in the computation, the application will crash. The accepted input data are limited, not all video formats are accepted neither all file format. For example, the bathymetry file must be a text file and DEM file must be a tiff one.

For the moment, the bathymetry file can contain only one cross-section geometry. Indeed, the analysis is focused on the computation of the discharge on one cross-section while it could be interesting to compute it on several sections to take an average value for example.

In the application the PIV analysis is done in one pass using the FFT algorithm. It could be interesting to add a multi-pass option since OpenPIV seems to have implement it. The velocity analysis done using OpenPIV can take some time. The OpenPIV version used was the Python implementation using the Central Processing Unit (CPU). The authors of OpenPIV have developed a Graphics Processing Unit (GPU) accelerated version. This version was not used in this work since it was still under development. In the future it could be interesting to test it to see the gain in computational time.

The validation step of the velocity vectors can also be enhanced. Indeed, currently the validation is based on the SNR ratio, other methods could be implemented and proposed to the user.

Finally, regarding the exportation of the results the velocity means vectors and their position as well as the water level and the discharge value are saved in a 'npy' file. Having the possibility to save graphs of these vectors, and other data treatment could be nice. Having a database where all results are stored and that can be managed online could ease the accessibility to the data and gather it in one place. Such database could also contain reference value of the discharge, water level and velocity to assess the precision of the application more easily.

Chapter 11

Conclusion

The interest in non-intrusive techniques to measure discharge's river has grown these last years. Their cost, deployment capability, rapidity of the measurements are the reasons that explained this interest. Especially with the development of Unmanned aerial vehicle (UAV) which add even more possibilities for measurements. One technique in particular has emerged, the Large-Scale Particle Image Velocimetry (LSPIV) adapted from the PIV technique that was used in laboratory for a long time. The LSPIV which is quite new (early 2000s) takes advantage of all the research that has already be done on the PIV technique but faced new challenges to transpose it for field measurement. Several software/tools implement this PIV analysis, but few are adapted to LSPIV measurements. For example, the commercial software DaVis [107] is dedicated to laboratory studies. This lack of software adapted to the LSPIV technique and the techniques studied or developed in the previous Master Theses is what have motivated the development of an application.

To start, an overview of the existing techniques for computing the discharge of a river has been done. In this overview the techniques were separated according to which feature they measured among the cross-section of the river, the water level, and the velocity profile. A distinction between the imaging techniques and the others was made to highlight the difference between the two in terms of cost, applicability, etc. During this overview, the methods developed or used in the previous Master Thesis have been assessed. More precisely, the LSPIV technique that Bia and Somers studied as well as the photogrammetry technique called Structure from Motion (SfM) used by Hellebaut.

Before the development of the application some choices have been made regarding the tools, libraries, and integrated development environment (IDE) used. This choice has leads to the Pyhton language, Spyder IDE, Kivy framework and NumPy and OpenCV libraries. To get some inspirations of features useful to implement some tools that already existed for LSPIV analysis such as Discharge or RIVeR have been analyzed. Several interesting features of these applications haev been implemented in the present application.

Several processes to correct distortions in images have also been implemented. A calibration step for the lens distortion induced in the image. The orthorectification of the images when they are taken from the shore is also implemented. Finally, if the image is taken from drone, a stabilization method has been developed as well. This stabilization method has been compared to the one Bia and Somers have developed. Their method was found slower than the one developed here and not conserving the duration of the video.

In the water level part of the application three methods have been implemented. The first one is an adaptation of the Hellebaut's method to get rid of parameters as much as possible. This adaptation also takes advantage of the visual aspect of the problem by offering a correction step to the user after the water level calculation by the method. The other two methods, Angle and Height method and Image method are new ones which are based on the same principle. They are simpler in terms of data measurements and avoid the used of commercial software which was used for the photogrammetry part of Hellebaut's method. The principle of this method has been tested on small examples where it worked. Experiments on real river should be done in future work to assess its precision in real conditions.

Regarding the PIV analysis the OpenPIV python module was chosen since it can easily be integrated to the application. To enhance the results several filters have been implemented in the application. They allow the user to counteract the different lighting conditions that can affected the image used for the analysis. The selection of a region of interest in the image and the application of masks also ease the computation of the velocity and speed up the process.

The application was tested on a real example and compared to other software results. The fact that the results were similar shows that the application can conduct a discharge computation from start to end and provide results in agreement with state-of-the-art software.

The LSPIV seduced more and more people with its cost, rapidity, and deployment capability. Nowadays few applications have implemented the LSPIV technique and proposed a complete analysis with pre and post processing steps. In this context the application developed here has aiming to provide as many features as possible while gathering the previous years' work. Of course, this application is not yet finished as its reliability can be enhanced, the features more numerous, and analysis to evaluate its accuracy conducted. Nevertheless, this is a first step in the development of an application which offers a solution to the growing need of LSPIV measurements.

Bibliography

[1] National Geographic. Bathymetry. Available at `https://www.nationalgeographic.org/encyclopedia/bathymetry/`, lastchecked: 10-10-2019.

[2] Brooke Mason. Depth Determination. Available at `https://iho.int/uploads/user/pubs/cb/c-13/english/C_13_Chapter_3_December2010.pdf`, lastchecked: 10-13-2019.

[3] Unknown. Diagram showing the basic principle of echo sounding. Available at `https://www.wikiwand.com/en/Echo_sounding`, lastchecked: 10-13-2019.

[4] Karl B. Jeffers. Hydrograhic manual, page 86. 1960.

[5] British Antarctic Survey. Multibeam echosounder. Available at `https://www.bas.ac.uk/polar-operations/sites-and-facilities/facility/rrs-james-clark-ross/multibeam-echosounder-2/`, lastchecked: 10-15-2019.

[6] Jr. Heidi M. Dierssen, Albert E. Theberge. Bathymetry: Assessment. Encyclopedia of Natural Resources, 09 2016.

[7] R.B. Mitson K.A. Johannesson. Fisheries Acoustics - A Practical Manual for Aquatic Biomass Estimation, page Chapter 3. 1983. Available at `http://www.fao.org/3/x5818e/x5818e04.htm`, lastchecked: 11-02-2019.

[8] Alvarinho J. Luis1 Shridhar D. Jawak1, Somashekhar S. Vadlamani2. A synoptic review on deriving bathymetry information using remote sensing technologies: Models, methods and comparisons. Advances in Remote Sensing, 2015.

[9] LaRocque and West. Principles of LIDAR Hydrographic Surveying. Available at `shorturl.at/ipuHV`, lastchecked: 11-05-2019.

[10] Unknown. DGPS. Available on EDX platform in course "Hydraulique fluviale 1 - Écoulements à surface libre", lastchecked: 11-08-2019.

[11] Leica geosystems. Leica GPS1200+ Series High performance GNSS System. Available at https://w3.leica-geosystems.com/downloads123/zz/gps/general/brochures/GPS1200_brochure_en.pdf, lastchecked: 11-10-2019.

[12] Racelogic. How Does DGPS (Differential GPS) Work? Available at https://racelogic.support/01VBOX_Automotive/01General_Information/Knowledge_Base/How_Does_DGPS_(Differential_GPS)_Work%3F, lastchecked: 11-10-2019.

[13] Service public de Wallonie (SPW). Le réseau WALCORS. Available at https://gnss.wallonie.be/walcors/reseau-walcors.html, lastchecked: 11-10-2019.

[14] Unknown. Staff gauge. Available at https://stevenswater.com/wp-content/uploads/staff_gage_hq.jpg, lastchecked: 11-20-2019.

[15] Unknown. Ultrasonic Vs Radar Level Measurement Techniques. Available at http://lidarradar.com/info/ultrasonic-vs-radar-level-measurement-techniques, lastchecked: 11-21-2019.

[16] Unknown. Radar and Ultrasonic sensors. Available at shorturl.at/ekwS2, lastchecked: 11-21-2019.

[17] LALLEMENT Christian. Hydrometry: measuring the flow rate of a river, why and how? Available at https://www.encyclopedie-environnement.org/en/water/hydrometry-measuring-flow-river-why-how/, lastchecked: 11-22-2019.

[18] Unknown. Water Level Instruments. Available at https://niwa.co.nz/our-services/instruments/instrumentsystems/products/water-level-instruments, lastchecked: 11-25-2019.

[19] Department of Irrigation and Drainage. DID manual, Hydrology and Water Resources, volume 4, page 131. 2009.

[20] Unknown. Shaft Encoder. Available at http://www.indiawrm.org/HIS/index.html?_toc420067979.htm, lastchecked: 11-25-2019.

[21] Unknown. Bubbler. Available at http://www.indiawrm.org/HIS/_toc420067982.htm, lastchecked: 11-25-2019.

[22] Unknown. Bubbler device for measuring liquid level. Available at https://www.slideserve.com/curry/lecture-4-level-measurement, lastchecked: 11-25-2019.

[23] W.Boiten. Hydrometry, pages 78–87. 2008.

[24] G. Perzyna. Current meter streamflow measurement by wading, pages 18–20. 2016.

[25] R. Bhagat. How to Measure Flow of Water in Open Channels? Available at https://www.engineeringenotes.com/water-engineering-2/open-channels/how-to-measure-flow-of-water-in-open-channels-water-engineering/44684, lastchecked: 12-03-2019.

[26] OTT HydroMet. Water flow meters. Available at https://www.ott.com/products/water-flow-3/ott-mf-pro-water-flow-meter-968/, lastchecked: 12-05-2019.

[27] Reza Langar Alan S. Morris. Measurement and Instrumentation. 2016.

[28] Unknown. Current Meters. Available at shorturl.at/yJTX8, lastchecked: 12-05-2019.

[29] Cadillac Meter. Ultrasonic Vs. Magnetic Flow Meters. Available at http://cadillacmeter.com/ultrasonic-vs-magnetic-flow-meters/, lastchecked: 12-08-2019.

[30] Teledyne RD. Teledyne RDI Workhorse Sentinel ADCP. Available at https://seatronics-group.com/metocean/teledyne-rdi-workhorse-sentinel-adcp/, lastchecked: 12-08-2019.

[31] Blessing Masasi. Doppler ultrasonic arrangement. Available at https://www.researchgate.net/figure/Doppler-ultrasonic-arrangement_fig1_318472016, lastchecked: 12-08-2019.

[32] USGS. How Streamflow is Measured. Available at https://www.usgs.gov/special-topic/water-science-school/science/how-streamflow-measured?qt-science_center_objects=0#qt-science_center_objects, lastchecked: 12-12-2019.

[33] Woods Hole Oceanographic Institution. Acoustic Doppler Current Profiler (ADCP). Available at https://www.whoi.edu/know-your-ocean/ocean-topics/tools-technology/acoustics/acoustic-doppler-current-profiler-adcp/, lastchecked: 12-12-2019.

[34] Benjamin E. Wilkinson Paul R. Wolf, Bon A. Dewitt. Elements of Photogrammetry, page 1. 02 2014.

[35] Stephen Kyle Thomas Luhmann, Stuart Robson and Jan Boehm. Close-Range Photogrammetry and 3D Imaging, pages 2–6. 09 2013.

[36] Karl Kraus. Photogrammetry, Geometry from Images and Laser Scans, pages 1–2. 1997.

[37] N.F. Glasser M.J. Hambrey J.M. Reynolds M.J. Westoby, J. Brasington. 'structure-from-motion' photogrammetry: A low-cost, effective tool for geoscience applications, geomorphology. 179:300–314, 04 2012.

[38] J. T. Dietrich B. C. Courville J. L. Jensen Fonstad, M. A. and P. E. Carbonneau. Topographic structure from motion: A new development in photogrammetric measurement. Earth surface processes and landforms, 38:421–430, 04 2013.

[39] Maarten Vergauwen Theo Moons and Luc Van Gool. 3D reconstruction from multiple images, pages 7–9. 07 2008.

[40] Theia vision. Incremental SfM Pipeline. Available at `http://theia-sfm.org/sfm.html`, lastchecked: 11-05-2019.

[41] Agisoft LLC. Agisoft PhotoScan User Manual. Available at `https://www.agisoft.com/pdf/photoscan-pro_1_4_en.pdf`, lastchecked: 11-05-2019.

[42] François-Xavier Hellebaut. Mesure du niveau d'eau par photogrammétrie à partir d'un drone. 2017-2018.

[43] Amy S. Woodget James T. Dietrich. Bathymetric structure from motion: Change detection and technical improvements. Research Gate, 04 2018.

[44] Bo Beltoft Watz Fahrettin Gökhan Ergin and Nicolai Fog Gade-Nielsen. A review of planar piv systems and image processing tools for lab-on-chip microfluidics. Sensors, 07 2018.

[45] R. Gurka T. Hadad. Effects of particle size, concentration and surface coating on turbulent flow properties obtained using piv/ptv. Elsevier, 2012.

[46] Ken Kiger. PIV Basics: Correlation. Available at `http://www.civil.ist.utl.pt/~ruif/SUMMER_SCHOOL/presentations/PIV_basics_correlation_final.pdf`, lastchecked: 11-20-2019.

[47] Katarína RATKOVSKÁ. Particle Image Velocimetry. Available at `http://home.zcu.cz/~rcermak/opvk_htt/VY_01_02.pdf`, lastchecked: 01-20-2020.

[48] F.T. Nieuwstadt. Flow visualization and Image Analysis, page 232. 1992.

[49] Harold SOMERS Tristan BIA. Mesure du champ de vitesses superficielles en rivière à partir d'un drone. 2017-2018.

[50] Carlos León Juan F. Reinoso and Jesús Mataix. Estimating horizontal displacement between dems by means of particle image velocimetry techniques. Remote Sens., 2015.

[51] Bernhard Wieneke. Piv uncertainty quantification and beyond. 12 2017.

[52] R. J. M. Bastiaans. Cross-correlation piv; theory, implementation and accuracy. pages 8–9, 01 2000.

[53] J. S. Bendat and A. G. Piersol. Nieuwstadt. Random Data, Analysis and Measurement Procedures. 1986.

[54] W. Thielicke. The flapping flight of birds: Analysis and application. 2014.

[55] H Eizenhöfer R Higham J M Buick, J A Cosgrove and R Royles. Characterization of the bubble cluster and velocity field in the focal region of a lithotripter. New Journal of Physics, 9, 06 2007.

[56] William Thielicke1 and Eize J. Stamhuis2. Pivlab – towards user-friendly, affordable and accurate digital particle image velocimetry in matlab. open research software, 2014.

[57] Kobayashi T. Okamoto K. Hu H., Saga T. and Taniguchi N. Evaluation of the cross correlation method by using piv standard images. Visualization, 1:87–94, 1998.

[58] Dimitri Bonnet. Vélocimétrie par images de particules (PIV) pour le diagnostic d'écoulements dans la boucle d'air automobile. pages 40–41, 2012.

[59] Douglas P. Hart. Vélocimétrie par images de particules (PIV) pour le diagnostic d'écoulements dans la boucle d'air automobile. Available at `http://web.mit.edu/dphart/www/PIV_ERROR2.PDF`, lastchecked: 01-28-2020.

[60] William Thielicke. PIVlab tutorial. Available at `https://pivlab.blogspot.com/p/blog-page_19.html`, lastchecked: 02-01-2020.

[61] John J. Charonko2 Zhenyu Xue1 and Pavlos P. Vlachos. Signal-to-noise ratio, error and uncertainty of piv measurement.

[62] Ronald Adrian. Scattering particle characteristics and their effect on pulsed laser measurements of fluid flow: Speckle velocimetry vs. particle image velocimetry. Applied optics, 23:1690–1, 06 1984.

[63] AndrésRodríguez Antoine Patalano, Carlos Marcelo García. Rectification of image velocity results (river): A simple and user-friendly toolbox for large scale water surface particle image velocimetry (piv) and particle tracking velocimetry (ptv). Computers Geosciences, 109:323–330, 12 2017.

[64] MathWorks. Matlab. Available at `https://nl.mathworks.com/products/matlab.html?s_tid=hp_ff_p_matlab`, lastchecked: 07-01-2020.

[65] MathWorks. A Brief History of MATLAB. Available at `https://nl.mathworks.com/company/newsletters/articles/a-brief-history-of-matlab.html`, lastchecked: 07-01-2020.

[66] Shrikant Srivastava. 7 Kinds of Applications That Can be Developed Using Python. Available at `https://appinventiv.com/blog/types-of-apps-developed-using-python/`, lastchecked: 07-01-2020.

[67] Spyder. Spyder overview. Available at `https://www.spyder-ide.org`, lastchecked: 07-02-2020.

[68] Kivy. Kivy. Available at `https://kivy.org/#home`, lastchecked: 07-03-2020.

[69] Phototrack AG. About us. Available at `http://www.photrack.ch/about_us.html`, lastchecked: 06-25-2020.

[70] Phototrack AG. Discharge. Available at `https://play.google.com/store/apps/details?id=ch.photrack.discharge&hl=fr`, lastchecked: 06-26-2020.

[71] Phototrack AG. Discharge manual. Available at `https://app.discharge.ch/res/discharge_manual_20170131.pdf`, lastchecked: 06-26-2020.

[72] Gérard DEGOUTTE. TRAITÉ D'HYDRAULIQUE À SURFACE LIBRE. Available at `https://www.hydrologie.org/MISC/hydr/degoutte/degoutte1.pdf`, lastchecked: 06-27-2020.

[73] Rafik Absi. A roughness and time dependent mixing length equation. Doboku Gakkai Ronbunshuu B, 62:437–446, 01 2006.

[74] Salvador Peña-Haro, Beat Lüthi, Martin Deter, and Maxence Carrel. Evaluation of the dischargeapp: a smartphone application for discharge measurements. 02 2019.

[75] Brieuc PEETERS Paul ASSELBERGHS. Mesure du niveau d'eau et de la vitesse de l'écoulement par imagerie en laboratoire et en rivière naturelle. 2018-2019.

[76] Larry R. Bohman. How to process a video with RIVeR Software. Available at https://my.usgs.gov/confluence/display/SurfBoard/How+to+process+a+video+with+RIVeR+Software, lastchecked: 10-12-2020.

[77] Intel. D400 Series Product Family. Available at https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf, lastchecked: 07-05-2020.

[78] Adrian Kaehler Gary Bradski. Learning OpenCV: Computer Vision with the OpenCV Library, page 373. 2008.

[79] Ciarán Hughes, Patrick Denny, Edward Jones, and Martin Glavin. Accuracy of fish-eye lens models. Applied optics, 49:3338–47, 06 2010.

[80] Satya Mallick. Approximate Focal Length for Webcams and Cell Phone Cameras. Available at https://www.learnopencv.com/approximate-focal-length-for-webcams-and-cell-phone-cameras/, lastchecked: 07-07-2020.

[81] Jin et al Zhang. Measurement of unmanned aerial vehicle attitude angles based on a single captured image. Sensors (Basel, Switzerland), 18, 08 2018.

[82] University of North Carolina. Measurements and Error Analysis. Available at https://www.webassign.net/question_assets/unccolphysmechl1/measurements/manual.html, lastchecked: 07-10-2020.

[83] TECHNISCHE BERATUNG INGENIEURBÜRO. Drone altimeter by image analysis. Available at https://www.technik-consulting.eu/en/problem_solving/drone_altimeter.html, lastchecked: 07-04-2020.

[84] William Thielicke. What is PIVlab? Available at https://pivlab.blogspot.com/p/what-is-pivlab.html, lastchecked: 07-07-2020.

[85] William Thielicke. PIVlab. Available at https://www.youtube.com/watch?v=Sp3Ounq07Qc, lastchecked: 03-12-2020.

[86] OpenPiv. OpenPIV: a python package for PIV image analysis. Available at https://openpiv.readthedocs.io/en/latest/, lastchecked: 04-02-2020.

[87] OpenCV. Camera Calibration. Available at https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html, lastchecked: 05-06-2020.

[88] Athanasia Louloudi. Evaluation of a bisensor system for 3d modeling of indoor environments. 09 2010.

[89] Unknown. Radial Distortion Correction. Available at http://www.baspsoftware.org/radcor_files/hs100.htm, lastchecked: 05-02-2020.

[90] MathWorks. Camera Intrinsic. Available at https://nl.mathworks.com/help/vision/ref/cameraintrinsics.html, lastchecked: 05-02-2020.

[91] Hyunchul Lee and Okkyung Choi. An efficient parameter update method of 360-degree vr image model. International Journal of Engineering Business Management, 11, 04 2019.

[92] Zhengyou Zhang. A flexible new technique for camera calibration. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22:1330 – 1334, 12 2000.

[93] unkown. Extrinsic Calibration. Available at https://stackoverflow.com/questions/32851702/extrinsic-calibration-with-cvsolvepnp, lastchecked: 07-12-2020.

[94] OpenCV. Calibration. Available at https://github.com/opencv/opencv/blob/2.4/modules/calib3d/src/calibration.cpp#L1778, lastchecked: 05-20-2020.

[95] Jonah V. Steinbuck Uri Shavit, Ryan J. Lowe. Intensity capping: a simple method to improve cross-correlation piv results. Exp Fluids, 42:225–240, 2007.

[96] OpenCV. Histogram. Available at https://docs.opencv.org/master/d5/daf/tutorial_py_histogram_equalization.html, lastchecked: 05-21-2020.

[97] Visem. Correct an underexposed image. Available at https://docs.opencv.org/3.4/d3/dc1/tutorial_basic_linear_transform.html, lastchecked: 05-22-2020.

[98] OpenCV. Image Denoising. Available at https://docs.opencv.org/3.4/d5/d69/tutorial_py_non_local_means.html, lastchecked: 06-01-2020.

[99] Olivier Hagolle. The orthorectification : how it works. Available at https://labo.obs-mip.fr/multitemp/the-orthorectification-how-it-works/, lastchecked: 06-03-2020.

[100] Larry Zitnick. Geometric Transformations. Available at `https://courses.cs.washington.edu/courses/csep576/11sp/pdf/Transformations.pdf`, lastchecked: 07-12-2020.

[101] Sanja Fidler. Matching Planar Objects In New Viewpoints And Much More via Homography. Available at `http://www.cs.toronto.edu/~fidler/slides/2017/CSC420/lecture9.pdf`, lastchecked: 07-13-2020.

[102] RIVeR. Tutorials. Available at `http://riverdischarge.blogspot.com/p/tutorials.html`, lastchecked: 06-05-2020.

[103] Alexandre Hauet, Thomas Morlot, and Léa Daubagnan. Velocity profile and depth-averaged to surface velocity in natural streams: A review over alarge sample of rivers. E3S Web of Conferences, 40:06015, 01 2018.

[104] Gartner J.W. Mason R.R. Costa J.E. Plant W.J. Spicer K.R. Haeni F.P. Melcher N.B. Keller W.C. Hayes K. Cheng, R.T. Evaluating a radar-based, non contact streamflow measurement system in the san joaquin river at vernalis, california. menlo park, california. 2004.

[105] FERDINAND SCHOUTHEETE, OLIVIER D'ODEIGNE, and Sandra Soares-Frazão. Drone-driven surface velocity measurements in natural rivers. pages 1137–1144, 09 2019.

[106] USGS. How Streamflow is Measured. Available at `https://www.usgs.gov/special-topic/water-science-school/science/how-streamflow-measured?qt-science_center_objects=0#qt-science_center_objects`, lastchecked: 06-04-2020.

[107] LaVision. DaVis – Software Solution for Intelligent Imaging. Available at `https://www.lavision.de/en/products/davis-software/`, lastchecked: 06-28-2020.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**