RUGGED WATER LEVEL AND FLOW METER

S.R. Dulani
(130139L)

N.G.Y. Dulanjani
(130140G)

H.A.D.A Perera
(130430V)

J.A.L.P Jayakody
(130683X)

Bachelor of the Science of Engineering

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

December 2017

RUGGED WATER LEVEL AND FLOW METER

Samarawickrama Ruchira Dulani
(130139L)

Niminna Gamage Yamuna Dulanjani
(130140G)

Halthota Arachchige Dulaj Ayeshan Perera
(130430V)

Jayakody Arachchige Lahiru Pradeep Jayakody
(130683X)

Thesis submitted in partial fulfillment of the requirements for the degree Bachelor of the Science of Engineering in Computer Science and Engineering

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

December 2017

DECLARATION

We, the project group FlowVision hereby declare that except where specified reference is made to the work of others, the project "Rugged Water Level and Flow Meter" is our own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgement.

Signatures of the candidates:

…….
S. Ruchira Dulani (130139L)

…….
N.G. Yamuna Dulanjani (130140G)

…….
H.A. Dulaj Perera (130430V)

…….
J.A. Lahiru Jayakody (130683X)

Supervisors:

……..
(Signature and Date)
Dr. H.M.N. Dilum Bandara

……..
(Signature and Date)

Dr. Chathura De Silva

ABSTRACT

Urban flooding is on the rise due to unplanned developments, inadequate drainage facilities, and climate change. Such floods have serious implications to human life and property. To predict urban flooding, estimating real-time water levels and flow rates of underground drainage systems is essential. As water levels and flow rates can reach higher values and could include debris, it is difficult to use conventional immersive sensors to estimate flow rate. Moreover, number of installed rugged water level and flow meters in a selected area should be sufficiently high to improve spatial and temporal accuracy of the data. However, such dense deployments are costly as flow meters currently in the market are relatively expensive. Furthermore, these sensors may be deployed in areas without direct connectivity to the power grid and Internet. Hence, they must be self-powered and need relatively large communication range. Therefore, there is still a need to develop non-immersive, low cost flow and water level meter.

We address this problem by designing and developing a rugged, non-immersive, and low-cost flow meter, which can take the measurements under higher flow rate conditions. Computer vision techniques have chosen, as vision-based flow measurement is inherently non-intrusive and can be economically developed using commodity digital cameras and low-end processors. The system consists of two sperate hardware devices, where one device is kept inside the manhole while other is kept outside of the drainage. The separation of the system into two devices is mainly due to complexity of GSM communication inside a closed drainage. We further modularized the system components to use as a general platform for vision-based flow meters that can be adopt to the new algorithms and devices. The system measures the surface flow velocity with a maximum absolute error of $0.189\ ms^{-1}$, when water level is 2 m.

Keywords: Digital Image Processing, Flow measurement, Flow meter, Particle Image Velocimetry

ACKNOWLEDGEMENT

First and foremost, we would like to express our sincere gratitude to our project supervisors, Dr. H.M.N. Dilum Bandara for the valuable guidance and dedicated involvement at every step throughout the process and Dr. Chathura De Silva for the valuable advices and the direction given to us regarding the project.

In addition, we would like to thank Lt. Col. Dr. Chandana Gamage, Mr. Nalin Karunasinghe and Mrs. Hashini Senaratne for the support, encouragement and insightful comments.

Last but not least, we would like to express our greatest gratitude to the Department of Computer Science and Engineering, University of Moratuwa for providing the support for us to successfully finish the project.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

CMOS	Complementary metal–oxide–semiconductor
CPU	Central Processing Unit
CSI	Camera Serial Interface
CTE	Circuit-Terminating Equipment
FFT	Fast Fourier Transform
FIR	Far infrared
GSM	Global System for Mobile Communications
IP	Internet Protocol
LED	Light Emitting Diode
LSPIV	Large Scale Particle Image Velocimetry
PCB	Printed Circuit Board
PSF	Point Spread Function
PSU	Power Supply Unit
PTV	Particle Tracking Velocimetry
SBC	Single Board Computer
SSH	Secure Shell
STI	Space-Time Image
STIV	Space-Time Image Velocimetry
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus

LIST OF APPENDICES

1 INTRODUCTION

1.1 Background

Urban flooding is on the rise due to unplanned developments, inadequate drainage facilities, and climate change. Such floods in urban areas have been identified as disastrous not only to the urban environment but also has serious implications to human life and property. The World Bank has introduced several possible solutions to manage urban flooding [1]. According to their cost-effective matrix, installing early warning systems is a low cost, high impact solution.

1.2 Motivation

To predict urban flooding, estimating water level and flow rate of underground drainage systems in real time is essential. However, estimating and communicating such data are difficult during potential flooding. For example, as water levels and flow rates can reach higher values and could include debris, it is difficult to use conventional immersive sensors. Moreover, density of the number of installed rugged water level and flow meters in a selected area should be sufficiently high to improve spatial and temporal accuracy of the data. Thus, water flow sensors must be economical enough to facilitate large-scale deployments. However, flow meters currently in market are relatively expensive. Furthermore, these sensors may be deployed in areas without direct connectivity to the power grid and Internet. Hence, they must be self-powered and need relatively large communication range. Therefore, there is still a need to develop non-immersive, low cost flow and water level meter.

Apart from the concerned situation, flow velocity measurement is important in hydrology as well. Recent researches show that vision-based flow velocity measurement which is called image velocimetry has become an active research area [2], [3], [4]. New methodologies and improvements have proposed along with experiments. Most of them focused on extraction of river flow information non-intrusively specially in flood conditions. Reasons to converge the flow measurement towards digital image processing can be summarized as:

- Inherent non-intrusiveness of image processing preserves applicability in various flow conditions such as flood flow with debris.
- More human sensible information can be obtained from images rather than from numerical sensor data.
- Increasing power of computers and widespread use of digital cameras have made the infrastructure for image velocimetry.

Although experiments and researches are conducted on the image velocimetry, the field has not still become a common option among practical implementations for general use. But this is widely used for laboratory level tests. Therefore, it will be further useful to develop an extendable flow measuring instrument. Modularity and extendibility of the instrument is important as new algorithms and improvements on image velocimetry are emerging day by day. Therefore, development of a vision-based flow measuring platform will be a contribution to bring image velocimetry for general use.

1.3 Problem Description

1.3.1 Problem Statement

Design and develop a rugged, non-immersive, and low-cost flow meter, which is capable of taking the measurements under higher flow rate conditions. The proposed device should be deployable in manholes of underground drainages. Therefore, the power supply and data transmission methods should be designed accordingly. Thus, the problem that this project attempts to address can be formulated as follows:

How to develop a non-immersive flow rate and water level meter for real-time measurements of a drainage flow?

1.3.2 Objectives

Objectives of this project can be stated as follows:

- To conduct a comprehensive literature study to identify related products, their features, and limitations.
- Identify suitable technologies such as image processing techniques and Ultrasonic

for water flow rate and level estimation.

- To design and develop a flow meter which is capable of
    - Estimating water level and flow rate in underground drainage systems.
    - Transmitting real-time measurements.
    - Installing inside manholes of drainages.
    - Using as a general platform which can adopt to new algorithms and technologies.

1.3.3 Contribution

Contributions of this project include the following:

- IoT enabled, non-immersive water level and flow rate meter for real-time measurements of a drainage flow
- General platform for vision-based flow meters
- New methodology for measuring velocity of fast moving objects using popular CMOS color cameras.

1.4 Outline

This report is organized as follows. Chapter 2 discusses the related work about existing products, their features, and limitations. We further discuss different algorithms and technologies, which can be adopted to the identified problem. System design which combines hardware design and software design is presented in Chapter 3. Chapter 4 describes the implementation of the hardware design and the software design. Performance evaluation is presented in Chapter 5. Chapter 6 will conclude the report with problems encountered, challenges, and future work.

2 Literature Review

This chapter describes the background information and the existing literature related to the research problem. Since the basic requirement of flow meter is measuring flow discharge, first section describes the existing methodologies for the flow discharge measurement. Velocity measurement is a necessary step to measure the flow discharge. Depend on the requirement of non-contact flow meter, existing methodologies and existing products for those methodologies discuss in the second section. Digital image processing is the technique used for velocity calculation in the flow meter. Next chapter explain about the digital image processing for velocity measurement under three main techniques. Since the water level measurement also a requirement of designed flow meter, Non-contact water level measurement methodologies are described in a separate section. At the end of the literature review comparison of those method is included.

2.1 Flow Discharge Measurement

There are several methodologies to measure flow discharge rate of drainages and reservoirs. The three main methods are Acoustic Doppler Current Profiler (ADCP), index-velocity, and velocity-area [5]. Most of the commercially available flow meters are based on velocity-area method which is an ISO standard for estimating flow discharge in a channel (see Figure 2.1) [1]. According to the standard, the method requires velocity measurements in several different points on the flow area and the water level, while the geometry of the cross-section of the pipe being known. Therefore, flow meters have multiple sensors to take velocity and length measurements.

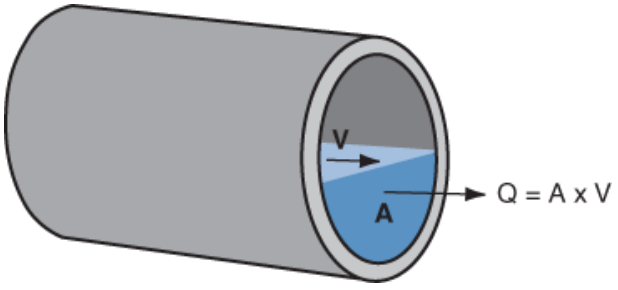


Figure 2.1 Calculation of flow discharge using velocity-area method.

Source: [6].

In most of the situations, it is capable of measuring only the surface flow velocity due to the site constraints or inherent limitations of measuring method. In such a situation, mean velocity needs to be derived for calculation of flow discharge using velocity-area method. Entropy method and its several extensions derived from the experiments done by Farina et al. [7], [2] can be used to calculate the mean velocity.

2.2 Non-contact Flow Velocity Measurement

2.2.1 Mechanical methods

Researches have come up with different mechanical methods to calculate flow rate. Kavanagh et al. [8] showed that rotating drums can be used to measure flow rate. The quality of estimation of the surface velocity obtained through usage of a measuring wheel placed in direct contact with the drum.

Rotameters are the most widely used type of Variable-Area (VA) flow meter. The falling and rising action of a float in a tapered tube provides a measure of flow rate in rotameters. Rotameters are known as gravity-type flowmeters because they are based on the opposition between the downward force of gravity and the upward force of the flowing fluid. The density of the flow will directly affect this method since this is based on a float. Clear flow needed to be go through the rotameter because float should not be blocked by any debris in the flow. When the flow is constant, the float stays in one position that can be related to the volumetric flow rate. That position is indicated on a graduated scale. FL-10 and FL-2000 [9] are two of commercially available products by omega engineering which are using rotameters to measure flow rate.

Spring piston flow meters are another mechanical approach used to measure flow rate in industrial level. This is also a variable-area flow meter. Instead of a float which is controlled by gravity in rotameter, there is a piston inside the component. So that this can be laid horizontally. This method need a clear and opaque fluid. Accuracy is bit higher than the rotameters. There are commercially available spring piston flow meters as FL-8100A and FL-8300A by Omega Engineering [10].

Turbine flowmeters use the mechanical energy of the fluid to rotate a “pinwheel” (rotor) in the flow stream. Blades on the rotor are angled to transform energy from the flow stream into rotational energy. The rotor shaft spins on bearings with the flow. When the fluid moves faster, the rotor spins proportionally faster. Turbine flowmeters now constitute 7% of the world market. Flow should be clear debris free because if the rotating wheel got stuck due debris flow rate cannot be measured. FTB-1300 [11] is a product which use turbine to measure flow rate by omega engineering. Daniel series 1500 liquid turbine flow meter is another commercially available product.

Pitot tubes or differential pressure sensor technology used widely in the industry to measure flow rate. Differential pressure flowmeters use Bernoulli’s equation to measure the flow of fluid in a pipe. These flowmeters introduce a constriction in the pipe that creates a pressure drop across the flowmeter. When the flow increases, more pressure drop is created. Impulse piping routes the upstream and downstream pressures of the flowmeter to the transmitter that measures the differential pressure to determine the fluid flow. FPT-3000 [12]is an existing product which uses differential pressure to calculating flow rate.

Coriolis method is another technique to measure flow velocity. In a Coriolis meter, material to be measured passes through one or more oscillating tubes. Oscillation affected by the rate which masses flow within the tube and it measures as a phase shift. This phase shift quantity is proportional to the mass flow. Since it output directly proportional to the mass flow rate it eliminates the complex calculation which needed in traditional flow meters. Coriolis meter is used as a standard for checking other flow meters because it has a good accuracy like ±0.15% [13]. For liquid flows, it is essential to make sure that flowmeter is completely full of liquid. With that constraint this method is more suitable for situations where pipe is fully filled and hard to implement in partially filled pipes. Krohne OPTIMASS 7000 [14] is a commercially available Coriolis Mass Flow Meter which have the capability to measure velocity, concentration, density and volume.

2.2.2 Ultrasonic methods

Researched have found out ultrasonic is an important method which can be used to measure flow rate and water level. They have tried up various methods using ultrasonic to measure flow rate. One of them is using transit time of the ultrasonic waves. The flow velocity determination method is based on the ultrasonic reflection principle. The ultrasonic transmitter is placed on the top of the pipe and receiver is also placed as same level as the transmitter considering the angle of the ultrasonic waves comes from the transmitter and reflected from the bottom of the pipe. This is a non-contact method used to measure flow rate. Rajita and Mandal [15] have reviewed present progress of ultrasonic transit time method. In that review they mainly focused on improvement of accuracy of velocity measurement and no implementation improvements suggested. FDT500 [16] is one of the commercially available flow meters which use ultrasonic transit time.

There is another different implementation of ultrasonic transit time method. In this implementation ultrasonic transmitter is placed on the top of the pipe and receiver is placed on the bottom of the pipe which can directly receive ultrasonic waves. So that the transmitter and receiver need to be fixed inside of the pipe which will contact the fluid flow. Saikia and Joshi [17] along with Joshi et al. [18] have worked on cheap solutions which are less disturbance to the flow. In those solutions they have designed a thin transducer which can be fixed inside walls of channels. However, the method is still hard to implement in partially filled pipes.

Ultrasonic Doppler method solves the problem of partially filled pipes. Akinin et al. [19] implemented and validated that the method can be used to measure the rate of a blood flow. The work by Dong et al. [20] presented a way of measuring the average flow velocity in continuous oil-water two-phase flow. Moreover, both these studies have shown the hardware needed for the method is simple and cost economical. However, this method requires some intrusive particles in the fluid. Blood vessels and oil drops are present in above scenarios. Those particles support to measure the flow rate by reflecting the incident wave back. FD-400 series [21] is an existing ultrasonic Doppler product in the industry.

Use of cross correlation technique of ultrasonic transit time is another approach for measure flow rate using ultrasonic waves. Velmurugan and Rajalakshmy [22] came up with a new system consists of ultrasonic sensors, changing the natural pressure, stochastic fluctuations of velocity and density into two signals with a real-time correlator and a delay time and obtaining the delay time from the signals and calculating the average rate of flow of the fluid.

2.2.3 Electromagnetic methods

Electromagnetism also can be used for measuring the flow rate. When electrically conducting material moves through magnetic field, a force is generated. If that conducting material is a fluid, then this principle can be used to measure the flow rate. Electromagnetic flow meters have been in use for several decades and now it has been grown up to the variants of standard practices [23]. Lorentz Force Velocimetry (LFV) is a contactless technique for the measurement of liquid metal flow rates [24]. Dubovikova et al. [25] presented a method for measure velocity and flow rate of liquid metal by using time-of-flight Lorentz force velocimetry. In that technique they use two flow meters and two permanent magnets and vortex are moving through those magnetic fields. Velocity will be calculated using obtained force signals and cross correlation method. This method has eliminated several problems in traditional electromagnetic flow meters but still fluid should have electrical conductivity. Some existing electromagnetic flow meters have capability of measuring flow rate when pipe is partially filled. Toshiba Magmeters LF series [26] provide electromagnetic flow meters with capabilities like measuring partially filled pipes LF502, harsh and nasty industrial applications LF510.

2.2.4 Surface velocity radar method

Surface Velocity Radars (SVR) is a sophisticated surface flow velocity measuring technique. SVR uses Doppler Effect in radars to measure the velocity in transparent or semi-transparent fluid flows. Costa et al. [27] conducted an experiment to make a completely non-contact open channel discharge measurement. They proof that it is possible to measure the actual discharge of the river within the standard accuracy using Doppler radar surface velocity measuring technique. Welber et al. [28]evaluated existing Doppler radar surface velocity measuring techniques and introduced a method for stream

gauging using portable surface velocity radars. They used midsection velocity area method to compute the discharge and velocity data are obtained using SVR. Flo-Dar 4000 LR [29] is an existing product which is implemented using Digital Doppler Radar velocity sensing technology with ultrasonic pulse echo depth sensing. This flow meter is highly accurate under both open channel and submerged conditions. When it becomes submerged no longer capable of measuring velocity using radar and it uses an electromagnetic sensor to measure the velocity. Finally, radar-based velocity results are unreliable when the return signal strength is too weak, typically when the free surface of the water is too smooth. Compared to other discussed methods SVR method is expensive to implement.

2.2.5 Laser Doppler velocimetry

Laser Doppler Velocimetry (LDV) is an important technique used to measure the flow velocity accurately. Specialty of the LDV is it produces absolute, linear velocity measurement without pre-calibration. If the fluid is transparent or semi-transparent, flow rate can then be determined by integrating the measured velocity profiles over the pipe cross-section. For opaque fluids, surface flow velocity can be calculated if the fluid is light reflecting. In this technique, a coherent laser beam is emitted from the transmitting optics and it is split into two beams using a prism. The paths of these beams are made to cross at the measurement location inside the transparent fluid or on the surface of the flow. Then scattered light is converged using receiving optics. Calculation of Doppler shift of converged light pattern gives the absolute fluid velocity. Ünsal et al. [30] have successfully used LDV for instantaneous mass flowrate measurements in 2006. After, Morita et al. [31] have developed a laser Doppler velocimeter with reduced physical size in the same year. At present, there are commercially available industrial flow meters that uses LDV for velocity measurement. ISCO Laser Flow [32] is an advanced product that uses LDV along with pressure and ultrasonic sensors for water level measurement in both partially filled and fully filled situations. Further, patented product miniLDV is an LDV based miniature velocimeter developed by Measurement Science Enterprise Inc. [33]. These products can address concerned problem. But, those sensors need multiple extensions or supportive products from same supplier to deploy as a working system. Then it becomes a very expensive solution, limiting its ability to be used in drainages.

2.2.6 Vision-based velocimetry

Surface flow velocity measurement along, it is possible to estimate the flow discharge. It facilitates to implement flow monitoring systems by observing the surface particle movement of flows using electromagnetic waves such as radio waves, microwaves, infrared and visible light. Among different electromagnetic waves, application of visible light or infrared for flow measurement has become increasingly common due to following reasons [2], [5].

- Diversity and quantity of physical information obtained from visible light or infrared images are greater than that obtained from other types of electromagnetic waves.
- Increasing power of computers and widespread use of digital cameras have made applications based on visible light or infrared is easier to develop than using low frequency techniques like radar.
- Image velocimetry is more user friendly because of the use of video, instead of transducer output such as voltage signals.

Yang and Kang [34]conducted a recent research about crowd-based image velocimetry and they have chosen to focus their research on an image-based method, because they think image-based velocity measurement methods have become an important research direction. It also proves that vision based velocimetry has a good future.

There are two main approaches for vision based velocimetry as particle image velocimetry (PIV) and Particle Tracking Velocimetry (PTV). PIV method is an Eulerian approach which models the fluid motion as vector field. In that method, fluid is added sufficiently small particles which are called tracer particles. Motion of these particles are used to calculate the velocity field of the fluid flow. PTV method is a Lagrangian approach which models the flow as set of particles having individual physical properties. In that method, tracer particles are individually tracked, and velocities of each particle is calculated.

In the considering drainage flow monitoring situation, high density of visible tracer particles cannot be expected. Main advantage of using PTV over PIV for sewer flow

monitoring is PTV can be applied to a flow having fewer number of particles. But PIV needs to have large number of particles distributed over flow volume.

In addition to those two common approaches, spatio-temporal images and blurred images also can be used for velocity vector construction of moving objects. Those approaches are described separately in following section along with corresponding related works.

2.3 Digital Image Processing for Velocimetry

2.3.1 Velocity extraction from Tracer Motion

The motion of the liquid flow is determined by the motion of the tracer particles. According to the density of the particles, there are two basic modes of operation. If the particle density is very low, it is referred as low image density PIV or PTV (Particle Tracking Velocimetry). If the particle density is high, it is called high image density PIV which is usually referred as PIV. In PTV the motion of a single particles is tracked between sequential images while PIV provides velocity for a set of particles in a selected interrogation region. One of the important facts about this vision based approaches is that its simplicity. Using videos and images instead of transducer outputs such as signals have made the technique more user friendly [1].

Particle Image Velocimetry

Particle image velocimetry (PIV) is an optical technique that is based on series of images which is one of the mainly used non-contact type surface flow measurement techniques. Some types of flows such as sewer flows contain particles that may float on the flow surface. In the context of vision based flow measurement, they are known as tracer particles. But most of the flows does not contain enough particles. In such a situation the fluid is seeded with tracer particles. Then the motion of the seeding particles is used to calculate the flow velocity.

Calculation of the flow velocity using tracer particles are carried out under the following three assumptions:

- Tracer particles follow exactly the fluid motion

- Tracer particles are distributed homogeneously
- Uniform displacement of tracer particles within the interrogation region.

A typical PIV system consists of three basic components which are strongly interrelated to each other. They are the light source, tracer particles, and the image recorder [2], [35]. Next, we discuss each of these in detail.

Light source – Illumination is very important in image recording. It improves the quality of the images. To capture clear images of the flow where the particles can be seen clearly, the illumination source plays an important role. For outdoor setups mostly where LSPIV is used, natural light (sun light) is used as the light source. For indoor experimental setup Laser, Halogen and sodium-vapor lamps are used to provide a strong illumination. These lamps are positioned to give a uniform illumination to the specific area and to avoid light reflection on the surface [2], [4]. In the case study conducted by Nguyen et al. [36] they have used low power and water-resistant infrared LED illumination devices as their light source. Because in contrast to visible light, infrared has the benefit of not attracting insects. In the case study about LSPIV conducted by Hauet et al. [37] , they have identified strong or poor illumination that might occur when only natural light is present, as a pervasive problem that affects the flow visualization. Further they have mentioned that the image quality is degraded by glare and shadow on the water surface.

Tracer particles – The movement of the water surface is clearly identified only if it contains visible elements moving with the flow. If tracer particles are not naturally present on flow surface or not well distributed, additional particles should be added. That increases the efficiency of resolving velocity on the entire image. Tracer particles should be smaller in size and lighter than the density of the flowing fluid, so that the particles can float on the flow surface [2].

Image recorder – Capturing the images of the surface flow is one of the basic and essential processes in PIV technique. The type of the camera depends on the type of experimental situation. Different types of cameras have been used in different case studies.

SelvaBalan, N. Sharma and G. Deshpande [2] have used a high speed digital camera having a CMOS image sensor. CMOS sensor is highly used than the CCD sensors because of its characteristics like size, price, function and power dissipation. [38].

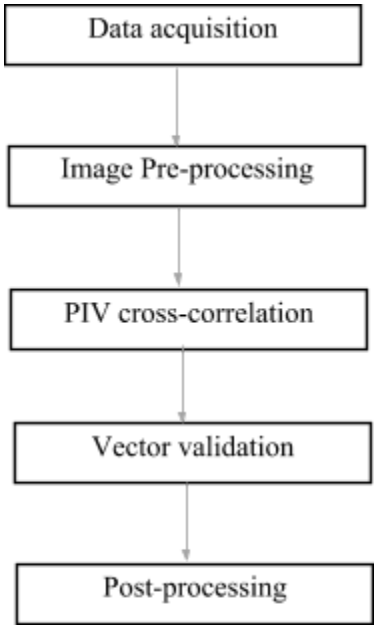
In the case study conducted by Nguyen et al. [36], they proposed using of a POE (Power over Ethernet) IP cameras to capture the flow surface. The main advantage of using that is, both power and data can be transmitted through the same ethernet cable and have standard connectors as well.

Generic PIV program

The flow diagram for the generic PIV technique is shown in Figure 2.2. Next, we discuss each of the process steps.

Data Acquisition

Data acquisition in PIV consists of seeding, illumination, imaging and registration. In the development of HydroPix system by Jeanbourquin et al. [39], AXIS 221 network cameras were chosen as they adapt the color depth of the images. They also adapt the acquisition frame rate according to the existing lighting condition. They are PoE IP cameras that enable both power and data to be transmitted through the same cable. They proposed using infrared LED for illumination. With correctly placed LEDs a 25fps frame rate was achieved.



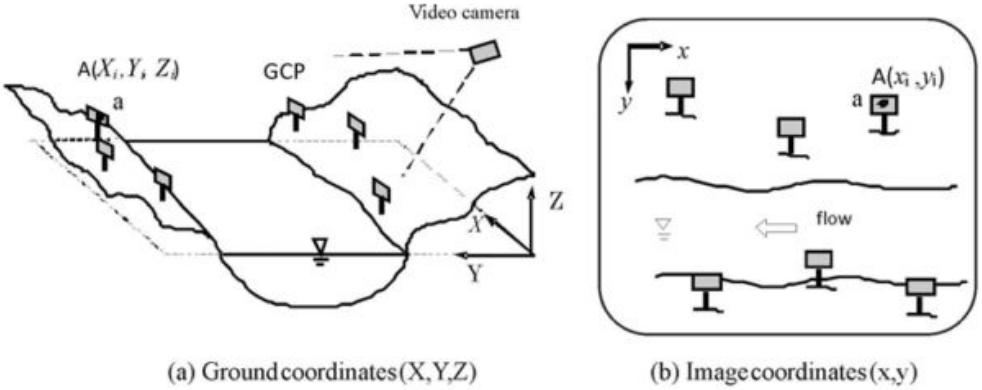
In most of the case studies regarding the large-scale PIV applications such as flow velocity measurement in rivers, the images have been taken from an oblique angle to the free surface plane. The images taken from an oblique angle do not give accurate results. So, the images acquired from the camera should be rectified by an appropriate image transformation scheme [2], [40]. This process is known as image registration or orthorectification. Muste et al. showed how a photogrammetric relation is applied to obtain ortho-rectified images. The relationship between the real world 3-D coordinate (X, Y, and Z) and the 2-D image coordinates (x and y) is shown in Figure 2.3. The coordinate mapping relationships between two images is given by following equation [40]. The mapping coefficients A1 to C3 can be determined by the least square method using the known ground corner points coordinates.

$$x = \frac{A_1X+A_2Y+A_3Z+A_4}{C_1X+C_2Y+C_3Z+1}\ ,\ y = \frac{B_1X+B_2Y+B_3Z+B_4}{C_1X+C_2Y+C_3Z+1} \tag{2.1}$$

Image Pre-processing

Image pre-processing includes work such as reducing non-uniformity of illumination, reflections and subtracting the background of the images. Most of the time the acquired images have a low Contrast-to-Noise Ratio (CNR). CNR value can be calculated by using below given following equation.

$$CNR = \frac{|m_{wall}-m_{water}|}{S_n} \tag{2.2}$$



(a) Ground coordinates (X,Y,Z) (b) Image coordinates (x,y)

m_{wall} and m_{water} are mean pixel intensities of the wall and the water respectively. S_n is the standard deviation of the noise. Noise due to low illumination, high video compression rate, and presence of tiny water particles in air resulted in such low-quality images [36].

The case study by Sharma et al. [3] discussed about applying a Gaussian smoothing to all video frames to attenuate the noise caused due to light reflection, data compression, etc. To estimate the background of the images, a temporal median filter is applied. Then this background is subtracted from each frame to obtain a clear visibility of moving particles. Then an adaptive threshold is used to convert the enhanced directional gradient image to a binary image [39], [4]. Figure 2.4 shows images of the original frame (a), background estimation (b) and binary image with possible particles respectively (c).

a) Original image b) Background estimation c) Binary image

PIV cross-correlation

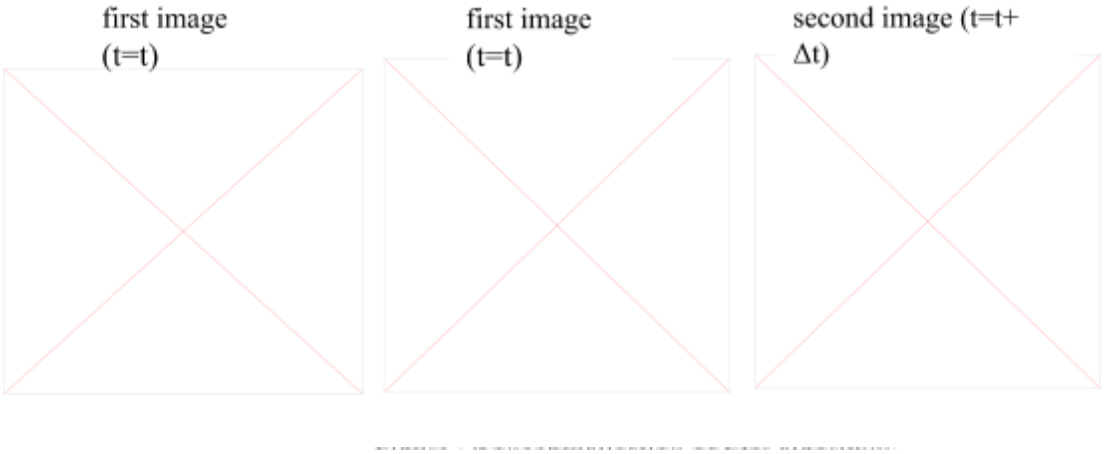
Historically two PIV methods have been developed. At first an auto-correlation method was found and then cross-correlation method. Auto-correlation method needs images to be doubly exposed while the cross-correlation method required images to be singly exposed. In both methods after determining the correlation peak, the displacement information can be obtained. The differences between the auto-correlation and cross correlation methods can be listed as follows (Table 2.1)

Table 2.1 Comparison between auto-correlation and cross correlation

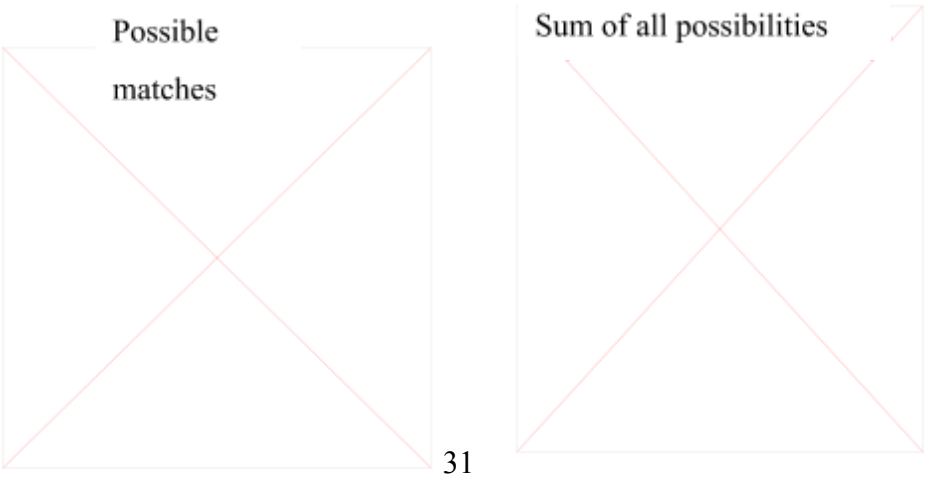
Auto-correlation method	**Cross-correlation method**
No need to transfer data within Δt	Fast data transfer

Directional ambiguity of displacement	No directional ambiguity
Can't detect small displacement	Can detect small displacements

Due to the above facts the mostly preferred method in PIV using presently is cross-correlation with singly exposed images. First the frame to be evaluated is divided into interrogation areas (IA) such that each area contains some number of particles. Then one interrogation area is selected as the Region of Interest (ROI) and cross-correlation procedure is applied between the selected area from the first frame and same size window of larger searching area of the second frame [3].



Because there is no directional ambiguity in cross-correlation, the direction of the flow can be easily calculated with more accuracy. When using the cross-correlation technique, one particle can be matched with number of candidates. The process is repeated for all the particles. Wrong combinations(matches) will lead to noise. However, the true displacement will have the highest correlation value and it will dominate as shown in Figure 2.6.



The cross-correlation coefficient R_{ab} is defined as:

$$R_{ab} = \frac{\sum_{x=1}^{MX}\sum_{y=1}^{MY}\{(a_{xy}-\overline{a_{xy}})(b_{xy}-\overline{b_{xy}})\}}{\{\sum_{x=1}^{MX}\sum_{y=1}^{MY}(a_{xy}-\overline{a_{xy}})^2 \sum_{x=1}^{MX}\sum_{y=1}^{MY}(b_{xy}-\overline{b_{xy}})^2\}^{\frac{1}{2}}} \quad (2.3)$$

Where *MX* and *MY* are the sizes of interrogation areas, and a_{xy} and b_{xy} are the distribution of gray level intensities in the two interrogation areas separated by the time interval Δt. The overbar indicates the mean value of the intensity for the interrogation area.

The pixel-level peak locations produced by the cross-correlation coefficient might not be exact. Nitika Sharma et al. [35] proposed to find sub-pixel positions from the cross-correlation coefficients. They have applied a 2-D analytical function to the correlation surface around the peak.

Vector validation

In PIV technique it is necessary to analyze the data and remove erroneous vectors. This process is named as vector validation. Due to uneven illumination and bad effect invalid velocity vectors can be occurred.

Common vector validation methods are based on comparison of the inspected vector with its immediate neighborhood. Vector Different Test and Median Test are some vector validation methods based on the neighborhood. In those method inspect the vector with its neighborhood vectors and delete the vectors which not satisfied the threshold. Neighborhood based methods are unable to detect false vectors when clustered and likely to mistakenly invalidate correct vectors.

Berghe [41] showed that there are another possible two methods, Velocity Limits and Standard Deviation Filter for vector validation. In velocity limit method, velocity limiters are applied to the data by selecting maximum and minimum horizontal and vertical velocities. Then vectors exceeding the threshold are categorized as outliers. Standard

Deviation Filter is a simple mathematical method. This method notes the number of times standard deviation of a vector which may differ from the mean value. These two methods are used for vector validation by Sharma et al. [35] in measuring non-intrusive water surface velocity using Spatial Cross Correlation technique.

Masullo et al. [42] proposed an adaptive method for outlier detection with the aim of been more robust in validation process if outlier clusters are present. The proposed method emulates the process of outlier detection in human vision. For each vector, the neighborhood is automatically enlarged until at least half the enclosed vectors are coherent. Although this method is more adaptive still applies the concept of median normalized thresholds. This algorithm capable of reducing the degree of over-detection.

Post-processing

Post-processing steps are applied after the vector validation. Interpolation, smoothing and calibration are happened in post-processing.

Interpolation and smoothing are applied to give the results a more natural and fluent look. After removing erroneous vectors in vector validation stage, there are some missing vectors which needs to be interpolated. Wang et al. [43] mentioned some methods to interpolate error vector in Digital Practical Image Velocimetry. Using the continuity equation of flow velocity for interpolation is a one such method. In this method the Interpolation precision is high. But the calculation cost is also high. Kriging method is another method to interpolate velocity vector. Kriging method is based on variation of function space analysis based on the limited value of regional variables within the region to conduct an unbiased optimal estimation method [44].

Calibration is a mandatory post-processing step in PIV technique. After the analysis of the images, the results are in the unit *pixels per frame*. After the analysis of the images, the results are in the unit *pixels per frame*. Therefore, these units need to be calibrated using an actual distance. Calibration is done using the time difference between two images (frame rate is used) and a reference point with known distance present in the images. This distance on the image in pixels can be related then to a real distance. This distance in

pixels can be related then to a real distance as given in following equation [35]

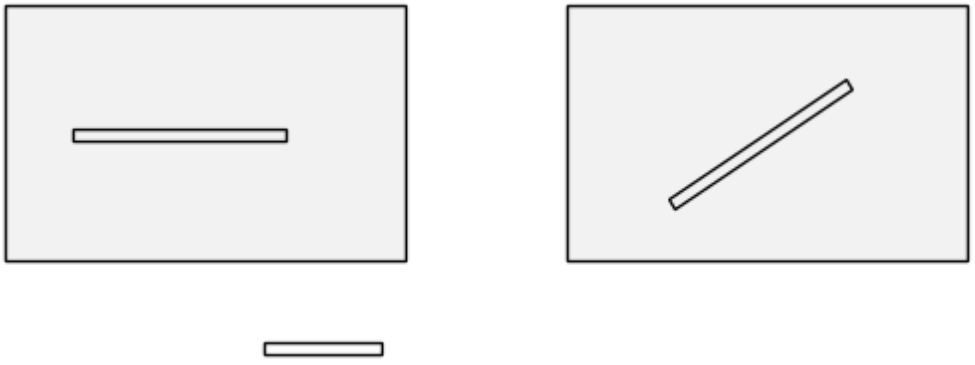
$$v = \frac{x\,pixels}{y\,frame} = \frac{pixel}{frame} \times \frac{Frame}{second} \times \frac{meter}{second} \quad (2.4)$$

2.3.2 Space-Time Image Velocimetry

The space-time image velocimetry (STIV) technique is an image analysis technique for measuring stream wise velocity distributions efficiently.

Space-Time Image

In STIV technique, space-time image (STI) is used to measure the flow velocity. Space-time image is generated using set of consecutive frames extracted from a video recording. To generate a STI, first a search line should be selected from the captured images. Search line is a line segment which is parallel to the flow direction. Figure 2.7 shows how to choose a search line from an image with respect to the flow direction.



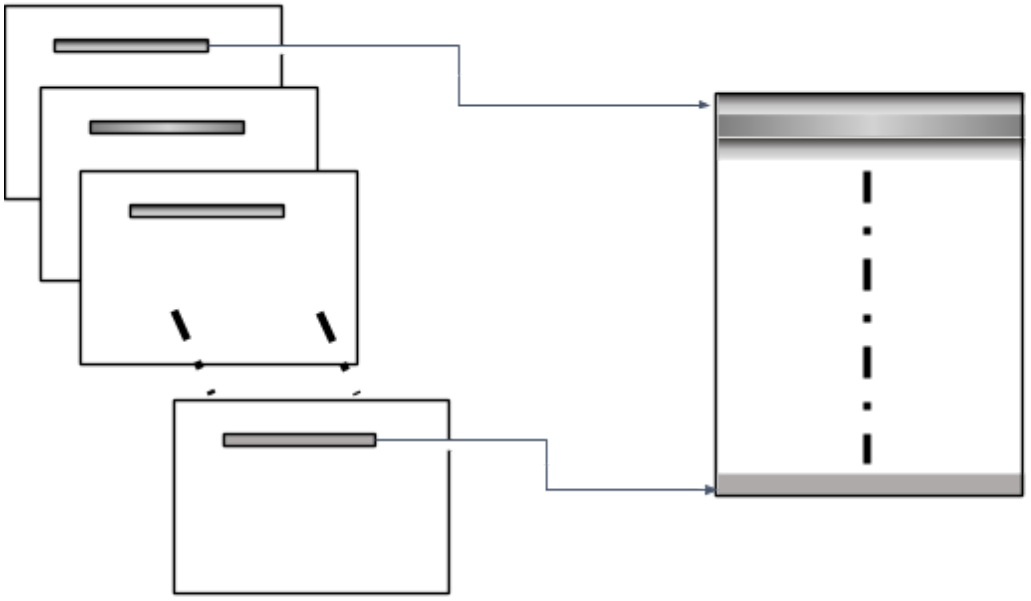
- flow direction
- selected search line

Image 1

Image 2

After selecting a search line, that line segment should be extracted from each image frame. A new space-time image will be generated by keeping the search line of each frame vertically one after the other. Figure 2.8 shows the construction of space-time image using consecutive frames of a video which has a flow direction that is parallel to the horizontal line of the image frame.

Horizontal length of the space-time image is equal to the length of the line segment which is selected as the search line and the vertical length vary with the number of consecutive frames obtained (N) to generate the STI.



t=0

t= Δt

t= 2 Δt

t= (N-1) Δt

Image frames

Space -Time Image

x – length of search line

N

Orientation Analysis of STI

In STIV main orientation angle of the space-time image will be used for measuring the flow velocity. Average orientation angle is considered as the main orientation angle of the pattern. Fujita et al. [45] showed a method to identify the orientation angle. They have used gradient tensor method to obtain the average orientation angle. In that method, the STI is divided into small segments and then the local gradient for each segment is calculated as indicated in Figure 2.9. They have calculated the local orientation angle φ

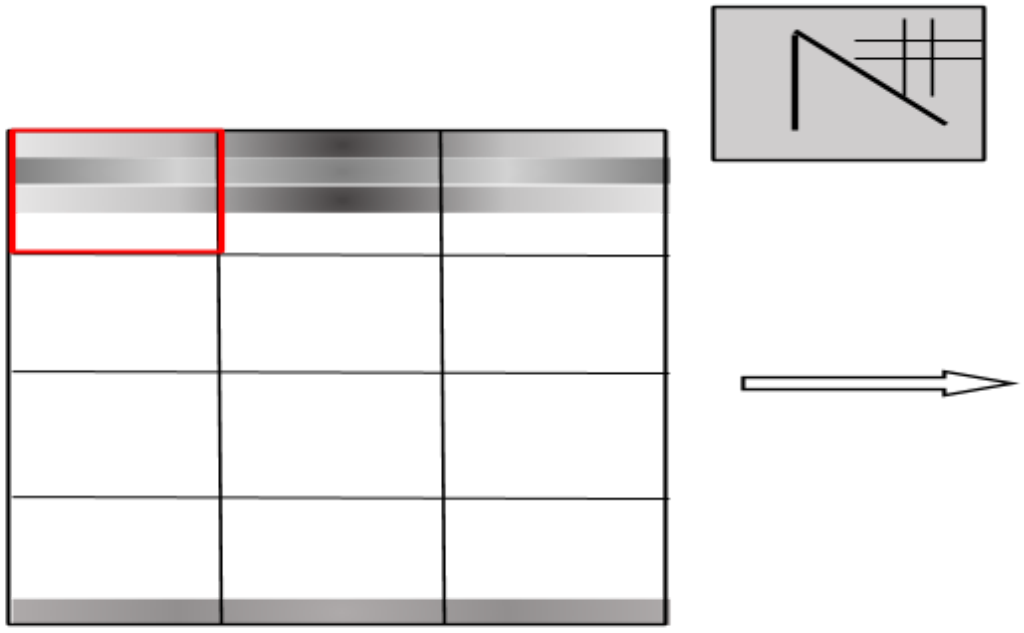
using following relations.

$$tan\ 2\phi = \frac{2J_{xt}}{J_{tt} - J_{xx}} \tag{2.5}$$

Where,

$$J_{xx} = \int_A \frac{\partial g}{\partial x}\frac{\partial g}{\partial x} dxdx \ , \ J_{xt} = \int_A \frac{\partial g}{\partial x}\frac{\partial g}{\partial t} dxdt \ , \ J_{tt} = \int_A \frac{\partial g}{\partial t}\frac{\partial g}{\partial t} dtdt$$

A - local area



x´

t´

g (x´, t´)

φ

space-time image (STI)

orientations for each local segment

$g(x,t)$ - grey level intensity on STI

When calculating the mean value of orientation angle for each small segment, coherency value (C) is considered as a weighting function. The coherency is a measure of image

pattern coherence and it is calculated using the following equation for local area.

$$C = \frac{\sqrt{(J_{tt} - J_{xx})^2 + 4J_{xt}^2}}{J_{tt} + J_{xx}} \tag{2.6}$$

Clear and reliable directional information for orientation angles can be picked by referring to the coherence value. Coherence value will be one for ideal local orientation and zero for an isotropic gray value pattern. Using the coherence value and the local orientation angle, mean orientation angle $\overline{\varphi}$ is calculated using the following equation.

$$\overline{\varphi} = \frac{\int \varphi \, C(\varphi) \, d\varphi}{\int C(\varphi) \, d\varphi} \tag{2.7}$$

Relationship between orientation angle and velocity information

In STIV technique, velocity information is derived based on the assumption that the brightness distribution of river surface image is convected with the surface velocity. Image orientation of the space-time image for a searching line set parallel to the main flow would indicate the velocity information. Since the length and time scales of the STI are given, mean velocity ($\overline{v}$) along the searching line segment can be calculated using following equation.

$$\overline{v} = \frac{x_0}{t_0} tan \, \overline{\varphi} \tag{2.8}$$

x_0 - unit length scale along the search line (m/pixel)

t_0 - unit time scale of the time axis

Use of STIV

Fujita et al. [45] proposed STIV as a novel image analysis technique for measuring the river surface flow. They compared the STIV with the existing LSPIV technique and showed that this technique is more robust to noise images than LSPIV. The other feature of STIV is its efficiency in the image analysis speed. CPU load is smaller than LSPIV.

In the case study of measuring discharge of snowmelt flood conducted by Ichiro Fujita

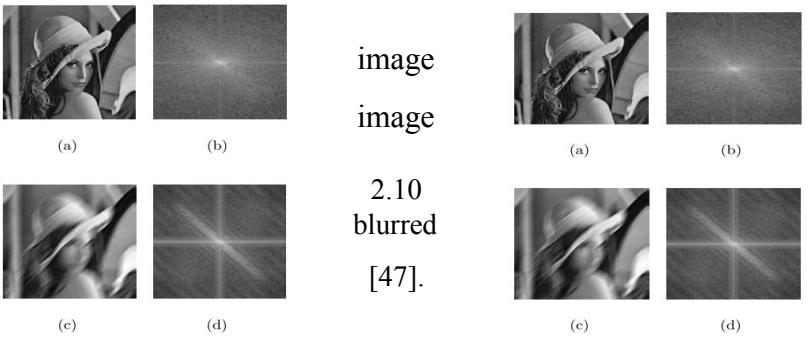
[46], STIV technique is used to measure the surface velocity. This case study is conducted based on the Uono River on Honshu Island, Japan. He presented that discharge measurements of snowmelt flood during the night can be conducted without any difficulty using an Far-Infrared (FIR) camera and the STIV technique. He proposed FIR camera for establish a real-time measurement system because other conventional cameras are difficult to use with little surrounding light.

This technique is not suitable for measuring recirculating flows which varies its direction with time. To obtain reliable velocity information, there should appear some traceable image pattern or surface texture moving with the surface flow.

2.3.3 Blur parameter based velocimetry

Motion blur

Blurred images can be formulated due to various causes such as atmospheric turbulence, defocusing of the lens, aberration in the optical systems, relative motion between the imaging system and the original scene. Motion blur is one of the most common causes of image degradation which is caused by the relative motion between the camera and the scene during the exposure time. Shutter speed of the camera one of the parameter that one can control without blurring the image the camera captures.



Original image and Blurred image Figure 2.10 Original and blurred images. Source: [47].

In general, blurred image can be approximately described by following equation

$$g(x,y) = f(x,y) \star h(x,y) + n(x,y) \qquad (2.9)$$

Where, * represent the convolution operator, $g(x,y)$ is the Observed image (Blurred image), $f(x,y)$ is the- Original image, $h(x,y)$ is the Blur kernel (point spread function),

and $n(x,y)$ is Noise function.

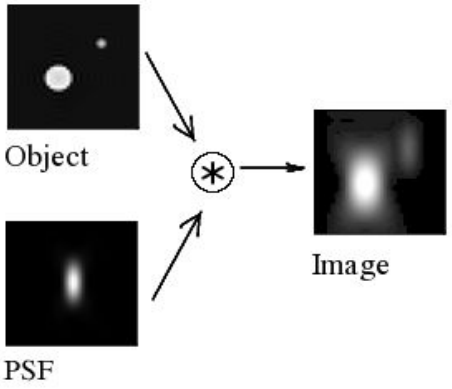


Figure 2.11 Blur image formation.

Source: [48].

The point spread function (PSF) describes the response of an imaging system to a point source or point object. In spatial domain PSF describes the degree to which an optical system blurs (spreads) a point of light. The motion blur PSF is characterized by two parameters, namely blur direction and blur length.

Assuming a linear motion blur, the PSF can be formulated using the motion blur angle (θ) and the motion blur length (d) using the following equation:

$$h(x,y) = \{\tfrac{1}{d} \; if \; 0 \leq |x| \leq d \; cos\theta \; ; \; y \; = \; d \; sin\theta \; 0 \; otherwise \qquad (2.10)$$

Assuming motion is along the horizontal direction i.e. φ = 0◦,

$$h(x,y) = \{\tfrac{1}{d} \; if \; 0 \leq |x| \leq d \; ; \; y \; = \; 0 \; 0 \; otherwise \qquad (2.11)$$

Using $sin\, x \; = \; \frac{1}{2j}[e^{jx} - e^{-jx}]$ on the Fourier transform of 2.11 one gets

$$H(k) = \; sinc(\pi kd).e^{-j\pi kd} \qquad (2.12)$$

Identification of blur parameters

1. Determination of PSF parameters in the spectral domain
2. Identify PSF parameters in the power cepstrum of the image by inspecting the negative peak
3. Power spectrum is averaged for reducing noise along the lines parallel to the expected minima and then the resulting one-dimensional spectral function is examined to detect the minima
4. Blur direction by rotating the coordinate system by an angle φ (varying from 0 to 180◦) and then by computing its 1-D spectrum and inspecting the peaks and valleys in it.
5. The blur direction is identified using Hough transform to detect the orientation of line in the log magnitude spectrum of the blurred image. The blur length is found by rotating the binarized spectrum of the blurred image in the estimated direction then by collapsing the 2-D spectrum into 1-D spectrum and finally by taking the inverse Fourier transform and finding the first negative value.

Almost all researchers have identified the blur parameters analyzing the Fourier transform of the blurred image. Some researchers have applied various filters identify parameters while some practiced different algorithms to detect blur parameters.

In Figure 2.12, image (a) and (b) shows the original image and its Fourier transform spectrum while image (c) and (d) shows the blurred image and its Fourier transform spectrum. Figure 2.13 represent the line pattern of a magnitude spectrum of a blurred image. Angle θ corresponds to the blur direction of the image and the measurement *d* in Figure 2.13 is inversely proportional to the blur length of the image. Researchers have found out many algorithms and methods to identify blur direction (angle θ) the and blur length (d). Most of the researchers have used the analysis of magnitude spectrum to identify blur parameters. Following two sections describes some methods they have presented in literature to identify blur parameters.

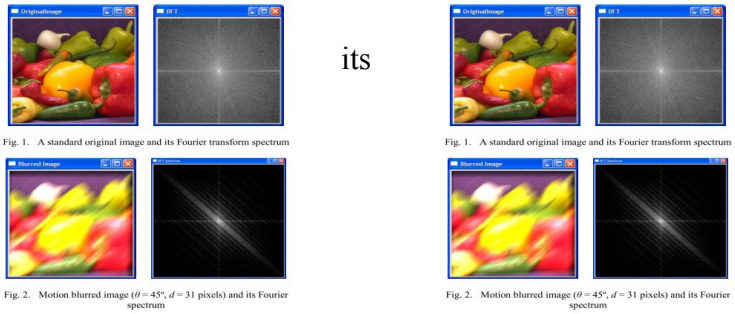
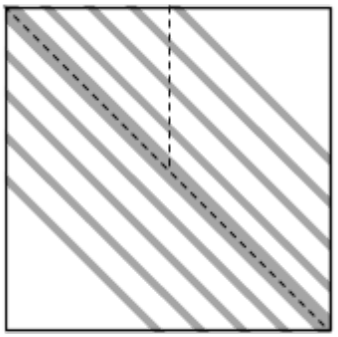


Figure 2.12 Images with its magnitude spectrum.

(c)

(d)



d

θ

Figure 2.13 Line pattern of a magnitude spectrum of a blurred image.

Blur direction identification

Lokhande et al [47] showed a method to identify blur direction based on the observation that the spectrum of non-blurred original image is isotropic, whereas that of motion blurred image is anisotropic. When observing the spectrum image, anisotropy in the spectrum introduced by the motion blur is in the direction, perpendicular to the motion direction. Therefore, to determine the blur direction, the spectrum is treated as an image and the Hough transform is then used to detect the orientation of the line in the spectrum. Some researches use Radon transform method to detect the orientation.

Soe et al [49] have represented an algorithm to detect the blur direction by analyzing the Fourier spectrum of the blurred image. They have identified the brighter and thicker pixel intensities in the central strip of the image (d) in Figure 2.12, the Fourier transform spectrum of the blurred image. Most of the brightest pixels of the spectrum are accumulated along the central stripe. Taking the advantage of this phenomenon, they have developed an algorithm which only finds the brightest pixel's orientation in each row of the spectrum matrix. The angles of those brightest pixels according to the origin of the coordinate system are calculated, the number of the same angle values are counted and then collected in an accumulation matrix. Then the algorithm finds the maximum angle

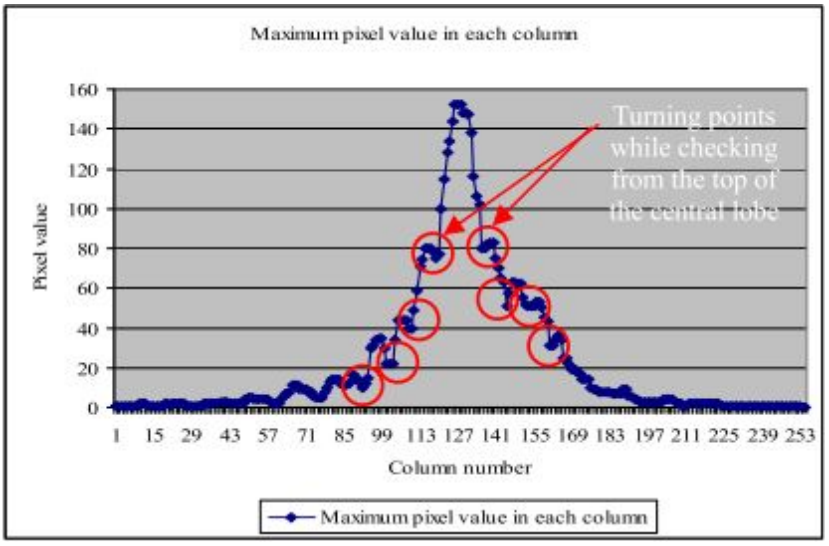
value from the accumulation matrix and that value is taken as the orientation of the spectrum stripes.

Blur length identification

Lokhande et al [47] showed the Fourier transform of observed image is equal to the multiplication of Fourier transform of PSF ($h(x,y)$) and original image. First need to need to take the Fourier transform of the blurred image. Then calculate the log spectrum of the previous image and convert it to binary. Rotate the binary spectrum in the direction opposite to the blur direction. Collapse the 2-D data into 1-D data by taking the average along the columns. Take the Inverse Fourier Transform of 1-D data obtained previously and locate the first negative value in the real part, which corresponds to the blur length.

Mostly suggested methods for identify blur length are Cepstrum method, radon transform method and rotation of coordinate axis method. Apart from using above methods, Soe et al [49] suggested two new approaches to detect blur length. The Fourie spectrum of the blurred image is first rotated counterclockwise according to the inclination angle in lines in image (d) in Figure 2.12. Then the maximum pixel value in each column of the image is checked vertically from top to bottom of the image. Then plot those pixel values against column number as showed in the Figure 2.14.

Calculate the average lobe width by summing up widths of all the considered lobes and then diving it with the total number of lobes considered. Blur length can be derived by dividing the width of the image by the average lobe width. To reduce the errors in matrix rotation in above example Soe et al. [49] came up with another algorithm to identify blur length without rotating the spectrum matrix.



2.4 Non-contact Water Level Measurement

Different methods are proven to be used to measure water level of a flow. Use of pressure, floats, ultrasonic waves, radar and image processing are mainly focused methods which used to measure water level [41].

Pressure is a technique used to measure water level with high measuring speed and it is easy to install but the precision is affected by the density of water. Floats is another approach taken to measure water level with very low power consumption. Ultrasonic transit time is a non-contact type which can be used to measure water level. Radar is a non-contact method with high measuring accuracy. Radar is equipped with the highest precision according to the methods discussed above. The main disadvantage in radar is, it is expensive to commercially use.

Above discussed methods cannot be changed with environment changes. So even after the installation and calibration regular visual inspections of the site are required identifying environmental changes. Moreover, irregular measurement results cannot be investigated comprehensively since a continuous observation of the site is impossible. One solution is the use of surveillance-cameras, which provides reliable picture quality. And in combination with advanced image processing algorithms the water level can be detected reliably and accurately [42].

Image processing techniques are used to measure water level. A research done by Kim et al. [43] used a scaled ruler fixed in the drainage to process the water level by an image taken from a camera. To overcome the difficulty of reflection of the water. The proposed method accumulates the degree of light reflectance variation between the current image and the previous images and constructs a histogram using them. In this histogram, the boundary between the ruler and water surface is appeared as the sharpest drop point.

Even though it seems image processing techniques are better than ultrasonic, radar, pressure etc. methods when considering the installation environment which is differ from previously discussed environments, image processing is not seeming to be the best

solution. According to problem statement, flow meter is required to install in manholes which are closed channel and with less lighting inside the drainage. So that it is hard to install a ruler inside the manhole and then the ruler need to have a separate lighting to get a reading from the ruler.

2.5 Comparison of Flow Measurement Methods and Products

Products and methods can be mainly classified into two categories based on placement of sensor. Contact type products needs to be placed in the flowing fluid and non-contact type measures remotely. Though contact type techniques are simple to implement and inexpensive, it can only be used for a debris free fluid flow. Because, debris and intrusive objects will get entangled in submerged parts of the meter. Moreover, there are methods which can be used to take measurements in partially filled pipes. Drainages do not have fixed water levels. Therefore, ability to take measurements in partially filled pipes is a must for a method.

Table 2.2 summarizes comparison of methods and products. Capability and cost are compared under following notations.

Capabilities:

- P - Capable of measuring velocity when pipe is Partially filled
- F - Capable of measuring velocity when pipe is Fully filled
- WL - Capable of measuring water level

Intrusiveness:

- C - Contact
- N - Non-contact

Cost:

- L - below $1000
- M - $1000 to $3000
- H - above $300

Table 2.2 Comparison of flow measurement methods and products.

Methods		**Intrusiveness**	**Capabilities**	**Cost**	**Constrains**	**Commercial products**	**Cites**
Mechanical	Rotameter	C	F	L	Vertical closed tube Clear fluid Depend on density	FL-10 FL-2000	[8] [9] [11] [12] [50]
	Spring Piston Flowmeters	C	F	L	Clear and Opaque Fluids Closed tube	FL-8100A and FL-8300A	
	Liquid Turbine Flowmeters	C	F	L	Clean water Closed tube	FTB-1300series Daniel series 1500	
	Pitot Tubes	C	F	L	Closed tube	FPT-3000	
Electromagnetic		N	PF	M	Conductive liquid	FMG70B Toshiba Magmeters series (LF150 LF450 LF510 LF502)	[24] [25] [26]
Coriolis mass		C	F	M	Closed tube	Krohne OPTIMASS 7000	[13] [14]
Ultrasonic gauge	Transit time	N	FWL	M	Two mountings needed Fully filled pipes	FDT500 FDT -30 series	[10] [16] [22]
	Doppler	N	PWL	M	Solid particles or air bubbles needed	FD -400 series	
	Cross correlation	N	WL	-	Continuous monitoring and frequency adjustment is needed to eliminate phase error		
Image Processing		N	PWL	H	Need tracer particles, Need illumination sources for closed channels	HydroPix Monitoring	[4] [5] [39] [51]

Radar	N	P	H	Inaccurate for very slow flows	Flo-Dar 4000 Lr RAVEN-EYE	[27] [28] [29]
Laser Doppler	N	P F	H	Transparent or semi-transparent flows needed	LaserFlow ISCO LaserFlow Flow Meter	[31] [33]

3 SYSTEM DESIGN

The system is a combination of hardware and software design. Hardware design consist of two sperate hardware devices where one is kept inside the manhole and the other is kept outside of the drainage. In section 3.1 describes the design of hardware. Software design consist of several modules such as algorithm module, camera module, communicator module and debugger module. In section 3.2 describes about those software modules. Algorithm module consists of several algorithm designs which can calculate flow velocity. Last section of this chapter describes the design of flow velocity calculation algorithms.

3.1 Hardware Design

High-level overview of the proposed system is illustrated in Figure 3.1. The system consists of two sperate hardware devices. One device which is called *Head* is kept inside the manhole while other device called *Tail* is kept outside of the drainage. The separation of the system into two devices is mainly due to low GSM signal strength inside a closed drainage. Having Tail device outside of the drainage also makes easy debugging possible without opening manhole.

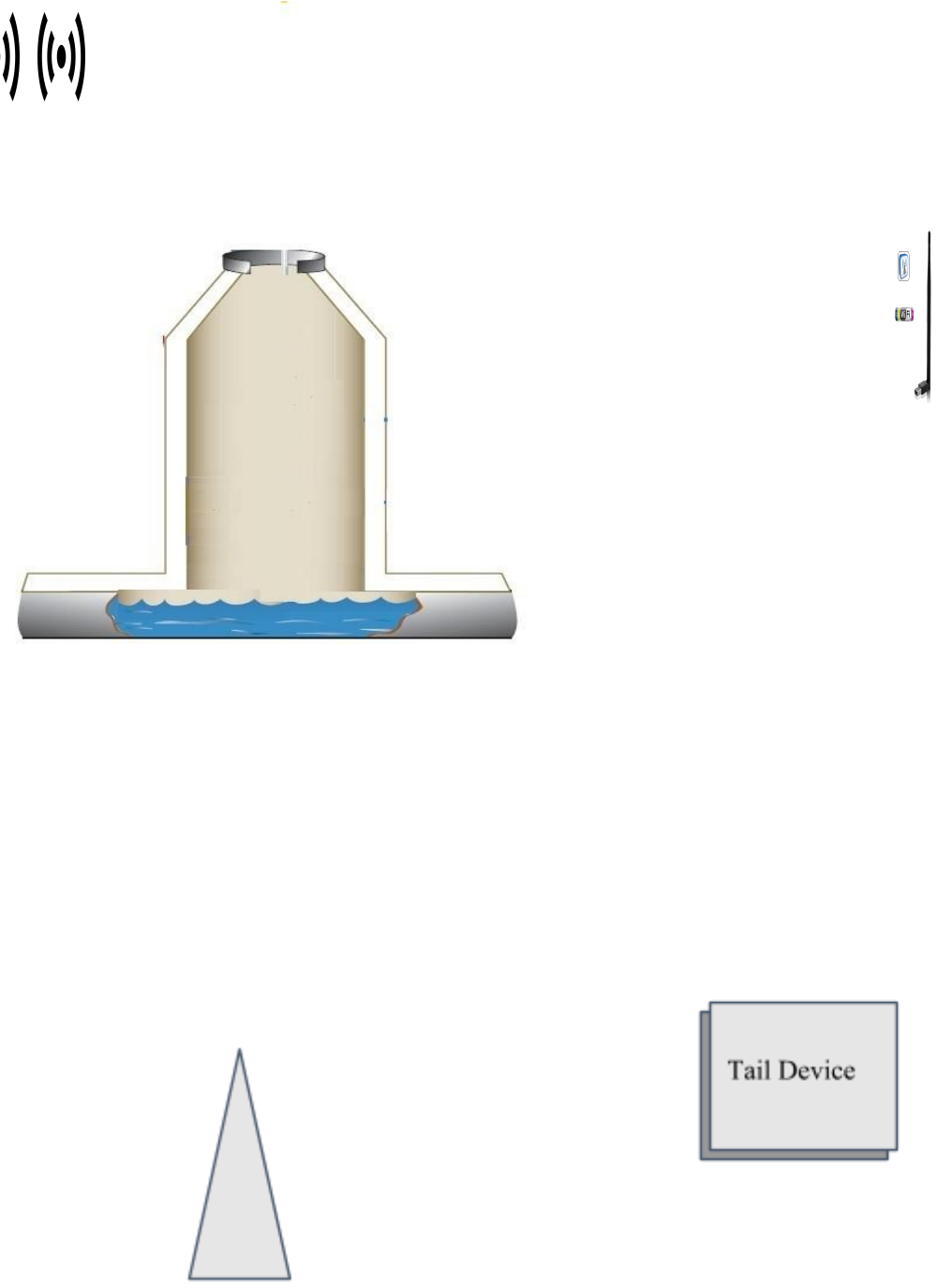


Figure 3.1 Deployment of the flow meter as two seperate devices.

Camera with Light source and Distance sensor must have to be in the Head to obtain data from the drainage. GSM modem needs to be placed outside to obtain good GSM signal

strength. Therefore, drainage sensors and GSM modem hard to be placed in a same device. That constrain fixes the placement of drainage sensors in the Head and modem in the Tail.

Main design considerations are placement of Single Board Computer (SBC) and Power Supply Unit (PSU). PSU should be along with the SBC to reduce the line drop in power transmission. If SBC is placing away from the camera, protocol conversion has to be done as Head and Tail will have more than 5 m distance in the deployment environment. Table 3.1 shows maximum data transmission distances for several possible communication protocols. According to the summery, Ethernet is the best possible option among other protocols. However, it introduces additional overhead of protocol conversion if the camera is not a IP camera. Moreover, IP cameras will not give adequate framerate for the requirement. Therefore, system is designed as both SBC and Camera are placed in Head to avoid communication overheads for video capturing.

Table 3.1 Communication protocols and limitations.

Protocol	Max Data Rate	Max Distance (m)
Camera Serial Interface (CSI) - 2	2 Gbps	1.5 m
USB - 2.0	480 Mbps	5 m
Ethernet - 1000BASE-T	1000 Mbps	100 m
RS-232	160 kbps	15 m
RS-485	10 Mbps	120 m

Head and Tail devices needs to communicate for sending measurements over the GSM network. Power should be transmitted from Tail to head and additionally state information may be transmitted from the head to tail for debugging purpose. These communications should be over the wires as power line is also among them. Having a multi-wire single cable in between Head and Tail with standard sockets and connectors will make maintenance and installment easy. Figure 3.2 depicts basic components in Head and Tail.

Figure 3.3 shows the architectural design of Head device. Power connections are omitted for the simplicity. In the design, all the hardware modules are considered as abstract entities which can be implemented according to the concerned situation. Following section describes the purpose of each module and implementation considerations.

Single Board Computer (SBC) and Camera

This is a computer which can be embedded in the flow meter. This should support for the runtime environment which is needed by the flow meter software. Camera model and protocol between camera and SBC depends on the supported camera interface by the SBC. Camera serial interface will be used, if Raspberry Pi or Banana Pi is used as the SBC. SBC should support at least two simultaneous UART communications; one for GSM modem and other for support chip. Design allows to have four wire debug interfaces. It allows to have many common protocols like JTAG, SPI, or Ethernet. Ethernet is preferred as it works well for far distance and it allows to have user friendly debugging such as SSH or VNC.

Distance Sensor

Distance sensor is responsible for measuring water level. It should provide stream of readings to the SBC via any two-wire protocol.

DCE Interface

Data Circuit-terminating Equipment (DCE) interface is responsible of converting UART signals of SBC to a suitable signal which can be transmitted over long distance. This module may be optional, if Head and Tail device are placed in distance less than 4 meters. Otherwise, this may convert UART to RS232 or RS485 depending on the distance.

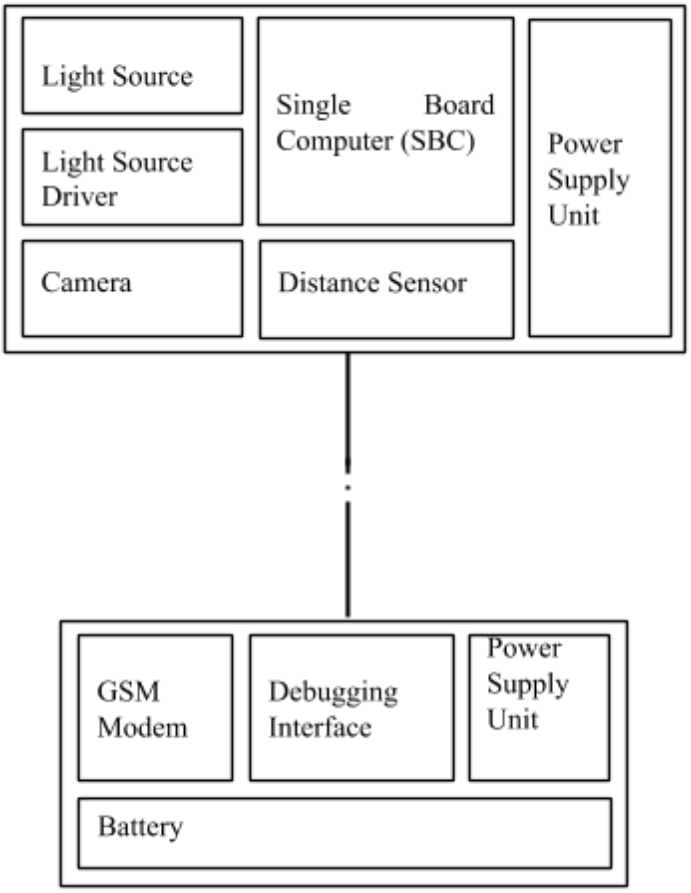
Tilt Sensor

This module is necessary for the image prospective corrections. In a dynamic and previously unknown environment, relying on perspective correction based on static environment features is hard. This module gives 2-D orientation of the camera by producing two angles with respect to the vertical.

Support Chip and On-board sensors

The flow meter head needs to be tightly sealed to have resistance for bad weather conditions. In the sealed environment humidity and temperature should be kept controlled. In the case of internal air reaching the dew point will hinder the camera view by accumulating dew. Further, design gives flexibility to add additional sensors to the I^2C bus

as needed. Support chip accepts configuration commands from SBC and it handles enclosure and lighting system related events. The design supports up to three LED lights.



Head Device

Tail Device

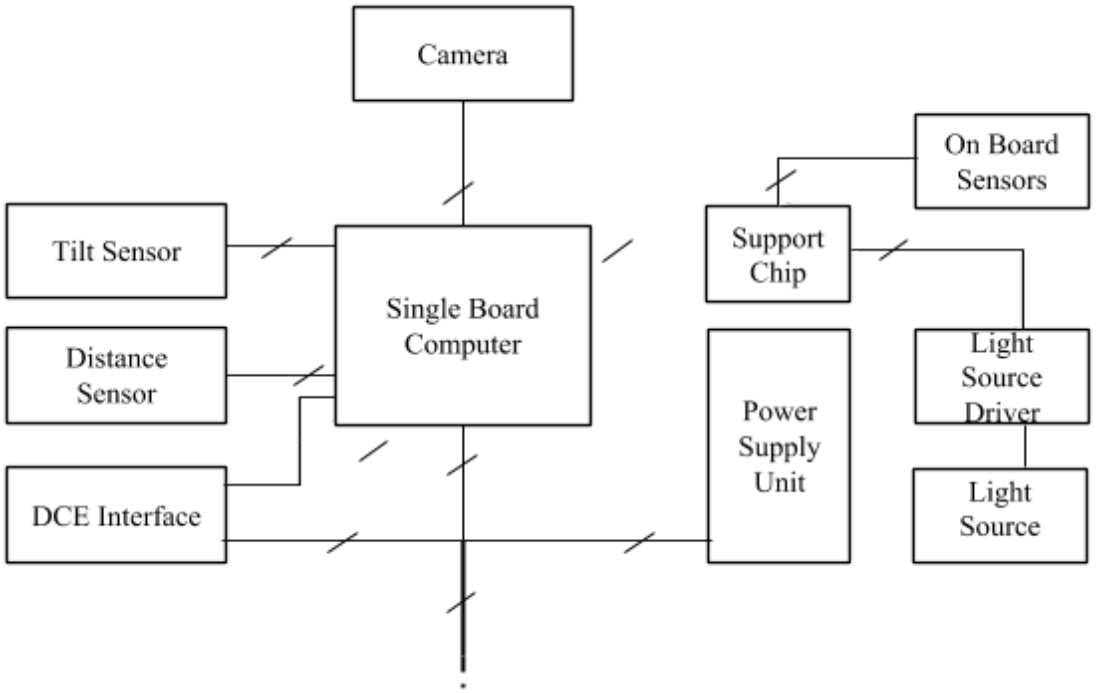
Wired connection

Power Supply Unit (PSU)

PSU of the Head needs to receive power from the PSU of the Tail. The design enables power transmission at a higher voltage. Transmission voltage and necessity of power boost conversion can be determined during the implementation depending on the Head and Tail separation distance. However, step-down conversion has to be done in the PSU. Table 3.2 shows the typical efficiencies of different power convertors. According to the efficiency Buck convertor is the best choice for step-down conversion. It has less than 3% power

loss. To decide to have a power boost before the transmission, power loss during the transmission under low voltage should be greater than 3%.

Figure 3.4 shows architectural design of Tail device. Main two functions of this device are to distribute power and to provide GSM communication. Additionally, it provides an interface to debug the flow meter.



CSI

15

I^2C

2

2

UART

2

UART

2

I^2C

2

3

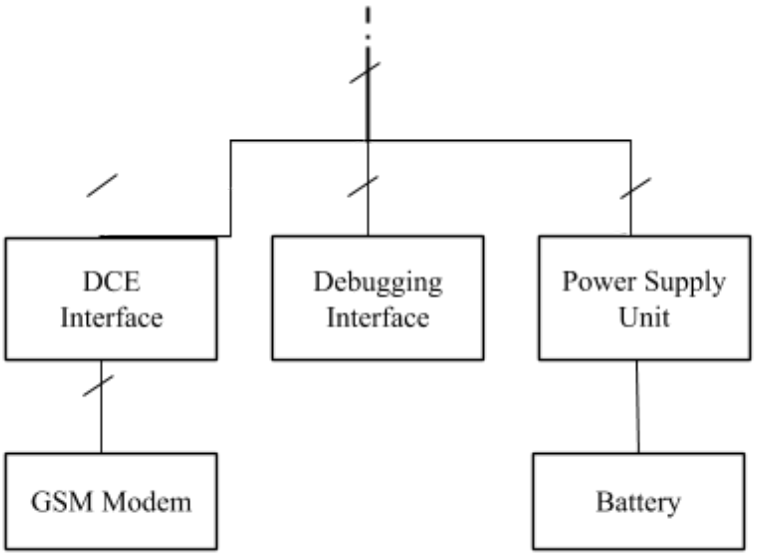
2

4

2

8

Figure 3.3 Architectural design of Head device.



8

2

4

2

2

UART

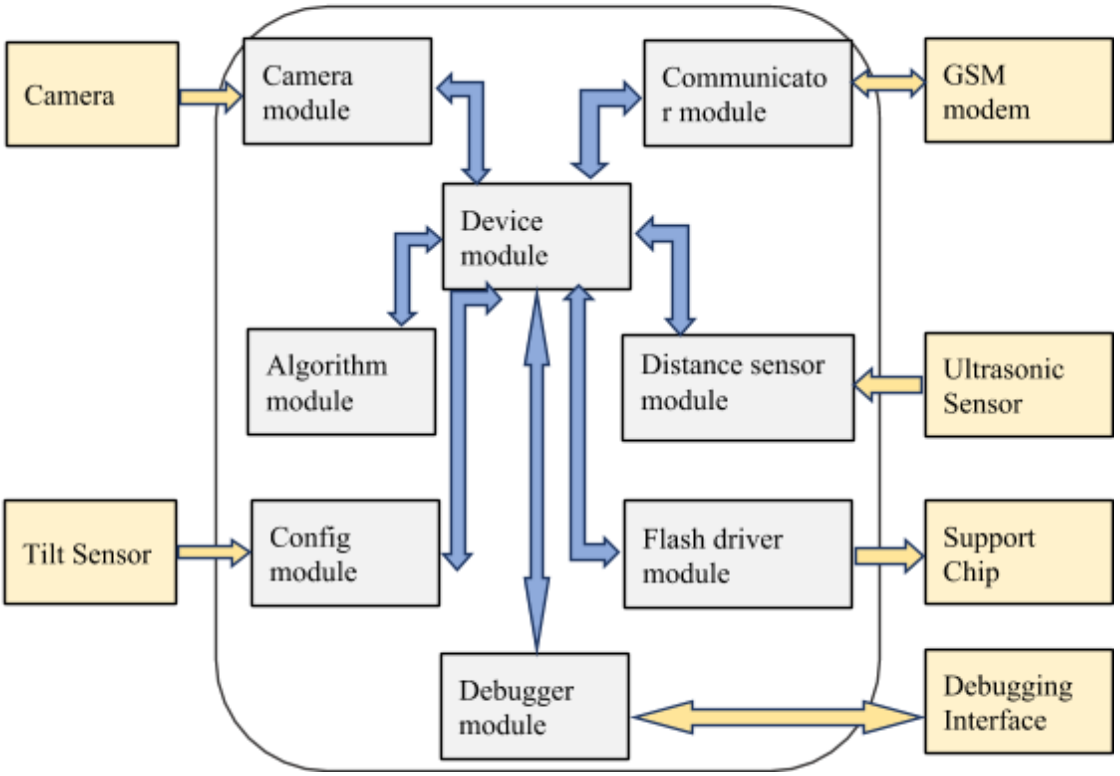
Table 3.2 Typical efficiencies of different power convertors.

Regulator	Efficiency
Linear regulator	49%
Buck / Boost convertor	> 97%
Buck convertor + Linear regulator	76%
Resonant-mode convertor	> 95%

3.2 Software Design

The software system in the Head device consists of several modules such as algorithm, camera, communicator, and debugger modules. Figure 3.5 shows how these software modules are interconnected with each other and their connection with the underlying

hardware devices. The software solution act as a framework to the overall system because all the software components are modularized and can be replaced. All the main modules in the system is connected to other modules through a unique single object from Device module. The software is designed to reduce the coupling between modules by assigning related tasks to the respective models. Following sections present a brief description about the main software modules in the system.



Single Board Computer

3.3 Algorithms

Algorithm module consists of several algorithm designs that can calculate the flow velocity. PIV algorithm [35] is one such design to calculate the velocity. However, it has the drawback of detecting false matches, which decreases the accuracy of the system. By extending the PIV algorithm we introduce a new algorithm called PIV Three-Frames algorithm, as it is more robust due to removal of false matching present in generic PIV. STIV [45] technique is another algorithm which can be used to calculate velocity of

turbulent high-density flows. These algorithms depend on the frame rate. Frame rate of a camera is mainly limited due to the data preparation and transmission bandwidth limitations. Therefore, we propose a new algorithm called PIV Color Channels algorithm that does not depend on the frame rate to calculate the velocity. PIV Three-Frames algorithm is the most suitable algorithm for previously unknown environment where floating particle colors, sizes may vary. Next four sections describe the design of these algorithms.

3.3.1 PIV Algorithm

Template matching is a technique in digital image processing to find small parts of an image which match a template image. This concept is used to match features present in consecutive frames. Template matching happens between two consecutive frames. Template is selected from the *n*-th frame and swept and matched on the *n+1*-th frame as shown in Figure 3.6.

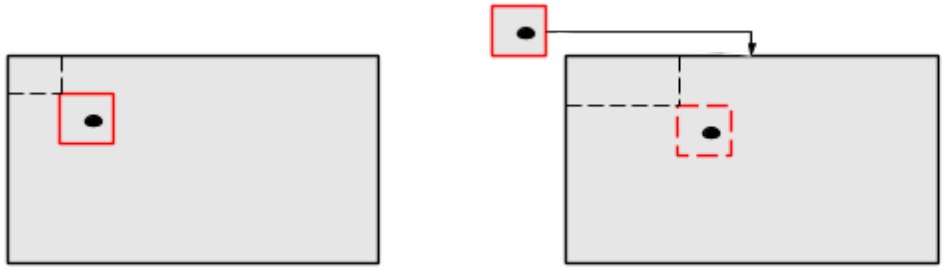


Figure 3.6 Template matching using two frames.

Figure 3.7 shows the basic flow of calculating velocity using template matching. First step is to apply pre-filters to remove noise. Next, template matching step selects a template and matches it with immediate next frame. Last step is validation, where matching results are validated to remove false matches.

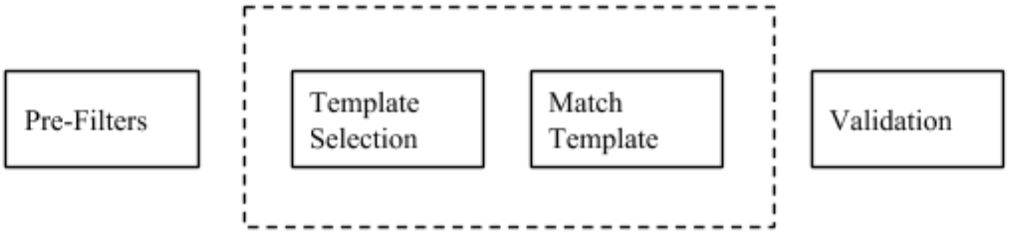


Figure 3.7 Velocity calculation using template matching for two frames.

Cross-Correlation procedure is applied here to match the selected area. Matching area will dominate with the highest correlation value. Selected template coordinates and matched coordinates returns after cross-correlation procedure are used to calculate the displacement within two frames. When using the cross-correlation technique, one particle can be matched with several candidates. To remove those false positives some validation techniques should be applied after getting the matching coordinates.

Accuracy of matching depends on the techniques used as pre-filters, validations, and template selection. Algorithm will change according to how the template is selected and matching process. How the two frames are selected as template for matching is different in PIV three-frame algorithm and PIV color channel algorithm. Those techniques are described in next sections.

3.3.2 PIV Three-Frames Algorithm

In this method three consecutive frames are used to calculate the velocity. Same flow in Figure 3.7 is extended with new steps for third frame matching as shown in Figure 3.8.

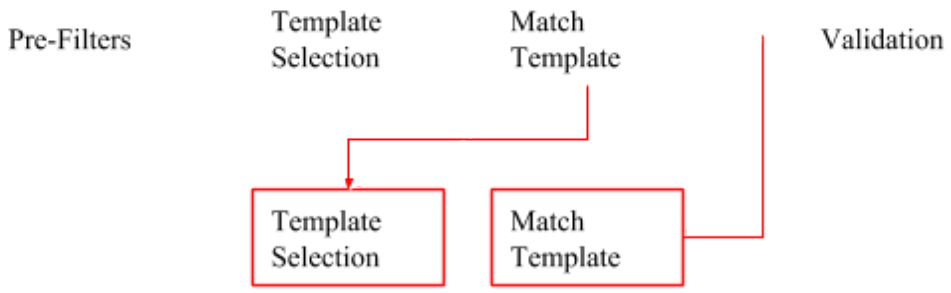


Figure 3.8 Velocity calculation using template matching for three frames.

In this algorithm template matching is applied two times before the validation state. First template matching happens between the frames at time t and $t+ \Delta t$. Second matching involves frames at $t+ \Delta t$ and $t+ 2\Delta t$. Figure 3.9 shows how the three frames are used in the template selection and template matching process.

Figure 3.9 Template matching using three frames.

Not like in basic PIV algorithm, here in this algorithm first the template matching is done between the first two frames and the matched area from the second image is selected as the template (ROI) for the next matching between the second and third frames. It is implemented in a way that helps to reduce the false positives. Rather than selecting a template and matching that template on next two frames, it changes the template to the matched region in second matching.

Pre-Filters

Before processing some pre-filters are applied to emphasize the features present in frames. We used background subtractor and morphological filters to identify the foreground and to remove the noise.

Background Subtractor Filter

Background subtractor is used to extract the foreground from static background. In PIV algorithm region of interest (i.e., template) should contains moving objects as features to match on next frames. Background subtractor is applied to extract the information about tracer particles floating with the water. Background subtractor generates a foreground mask which is a binary image containing the pixels belonging to moving objects in the scene. Foreground mask is generated by performing a subtraction between current frame and the background model and applying a threshold as showed in Figure 3.10.

Background initialization process and background updating to adapt possible changes in the scene are the two stages of background modeling. In background modeling, number of last frames which are used to generate the background model is an important parameter. That parameter value will decide the number of last frames that affect the background model.

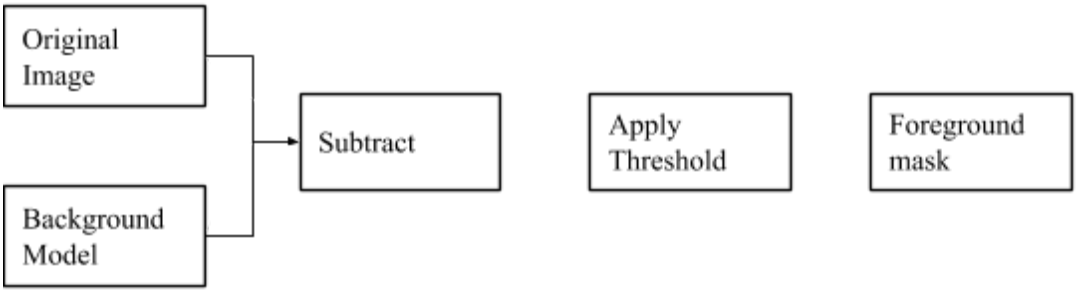


Figure 3.10 Construction of foreground mask.

Threshold: one of core parameters in background subtracting process. Threshold is applied to the absolute values got from subtracting the background model from the original image. Values greater than the threshold are classified as foreground and a binary image is generated for the foreground mask.

Gaussian Mixture-based Background/Foreground Segmentation Algorithm is one of the algorithms used in background subtraction. In this algorithm, Gaussian probabilistic density function is fitted into the number of last frames (*n* frames) is generated for each pixel. It is modeled as a mixture of adaptive Gaussians. It uses a mixture which is appropriate for each pixel because it provides better adaptability to varying scenes due to illumination changes.

Morphological Filters

Morphological filters are applied to the foreground masked frame, which is received after applying background subtractor. Aim of applying these filters is to remove noise present in the foreground mask. These filters will remove small bright spots and connect small dark cracks present. Erosion and Dilation morphological operations are typically applied, where:

- Erosion – Morphological erosion sets a pixel at (*i, j*) to the minimum over all pixels in the neighborhood centered at (*i, j*).
- Dilation – Morphological dilation sets a pixel at *(i, j)* to the maximum over all pixels in the neighborhood centered at *(i, j)*.

Erosion operation is first applied to the image as it removes the light regions that are

smaller than the actual particles. Some amount of noise presented will be removed and actual foreground is retained. Dilation operation is applied next as it increases the thickness of presented foreground. It will emphasize the features in the frame.

Template Selection

Before going to template matching process, we need to select a good template which includes enough tracer particles to be tracked. Regions with good features need to be identified. Clustering is used to identify the region of interests.

Density Based Clustering

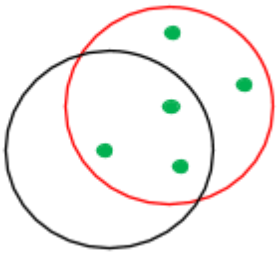
In density based clustering clusters are defined as areas of higher density than the remainder of data set. It is used to cluster the features in the foreground mask binary image. As shown in Figure 3.11 foreground mask can be clustered to extract features. For clustering Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm is used. Core points are the points form clusters together with all points that are reachable from it. In this algorithm clusters are defined based on two parameters, namely:

Figure 3.11 Application of density based clustering.

1. Maximum radius of the neighborhood from a core point (Eps).
2. Minimum number of points within the maximum radius (MinPts).

Figure 3.12 shows an illustration of those two parameters. A low MinPts means it will build more clusters from noise. So, the value of MinPts should be a considerable value which decrease the effect of noise.

Only the white pixels, which are pixels relevant to the foreground in foreground mask, are used for the clustering. Background is not considered. After identifying clusters template area should be selected in a way it covers best features present. Best features mean the largest clusters from the identified clusters set. After finding largest cluster, template boundaries should be selected to cover the cluster with an offset. In Figure 3.13 shows that how the template boundaries are calculated based on the best cluster points.



p

q

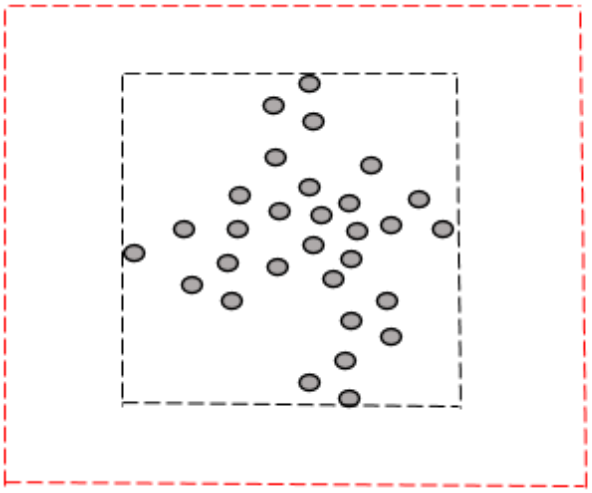
Eps = Ɛ

MinPts = 5

Ɛ

Ɛ

core point



cluster boundaries

offset 1

offset 2

template boundaries

Figure 3.13 Template selection with respect to the best cluster.

Validation

In PIV three frames algorithm four pixels distance values are obtained after the template matching process is carried out on three consecutive frames. They are the x, y coordinate pixel distances calculated by template matching between first and second frames and x, y pixel distance values calculated by applying template matching between second and third frame. In Figure 3.14 shows the illustration of x, y pixel distances according to the matchings happened between three frames.

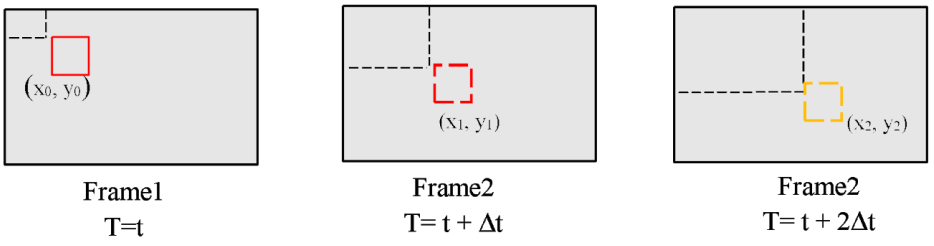


Figure 3.14 Pixel distance calculation.

first_x_difference = x1 – x0

first_y_difference = y1 – y0

second_x_difference = x2 – x1

second_y_difference = y2 – y1

x_difference = | first_x_difference - second_x_difference |

y_difference = | first_y_difference - second_y_difference |

In the validation phase the value difference between *first_x_difference* and *second_x_difference* (*x_difference*) is calculated. Then a threshold value is used to filter the pixel distances of correctly matched templates. If the *x_difference* value is less than the threshold value, that is considered as a correct pixel distance value. The same procedure is applied to y-coordinates. All the *x_difference* and *y_difference* values that are higher than the threshold value are neglected.

The next important process is the calculation of direction vectors for those correctly classified pixel distance values. If the direction of the flow in both the matches are same

that match is accepted as a correct match and the velocity value is calculated for that pixel distance value.

3.3.3 PIV Color Channels Algorithm

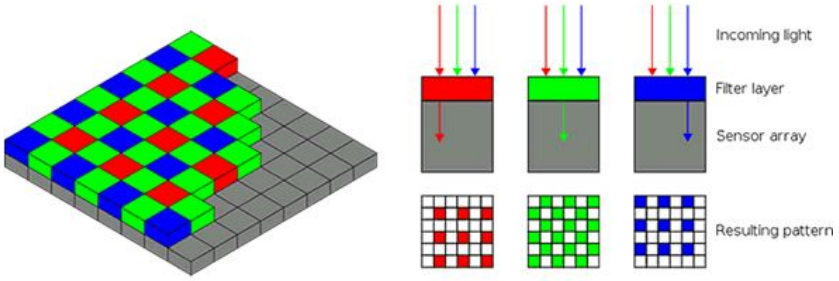


Figure 3.15 Bayer arrangement of color filters on the pixel array of an image sensor.

This algorithm implements new method which is based on Bayer filter of CMOS color cameras. Bayer filter is a Color Filter Array (CFA) which is used to store RGB color details separately in homogeneous image sensor elements. In normal single frame single exposed PIV implementation, two consecutive frames are needed to compute cross-correlation matrix. Reliably measurable maximum velocity is mainly limited by the minimum time delay between two consecutive frames. That delay is inversely proportional to the frame rate of the camera. Therefore, camera should be capable of capturing images at higher framerate for measuring higher velocities using PIV algorithms. Frame rate of a camera is mainly limited due to the data preparation and transmission bandwidth limitations.

This data transmission overhead can be avoided by doubly exposing a single frame to have two overlapped time shifted images in a same frame. However, it makes analysis of the image erroneous since features can be hidden in a case of lack of particles and present of lengthy particles. In the new methodology, Bayer filter is used to separate two overlapped images from a color image. To make the separation possible, light conditions should be controlled.

Bayer filter is a visible light frequency pass filter arrangement (see Figure 3.15 and 3.16). If we illuminate the imaging object with green light, pixels under the green filter only will

be exposed. Ideally it should cause to have an image with black red and blue channels and object should be appeared in green channel of the image. This does not happen in a real scenario. Pure red lights which emits narrow band of frequencies are not available for object lighting purpose. Filter layers in the Bayer filter also does not perfectly pass a narrow beam (see Figure 3.16). We identified two special features of red and blue light combination:

1. Relative responses of Bayer filter for red and blue colors are considerably independent (see Figure 3.16).
2. Red and blue colors channels have equal resolutions while green has twice as others (see Figure 3.15).

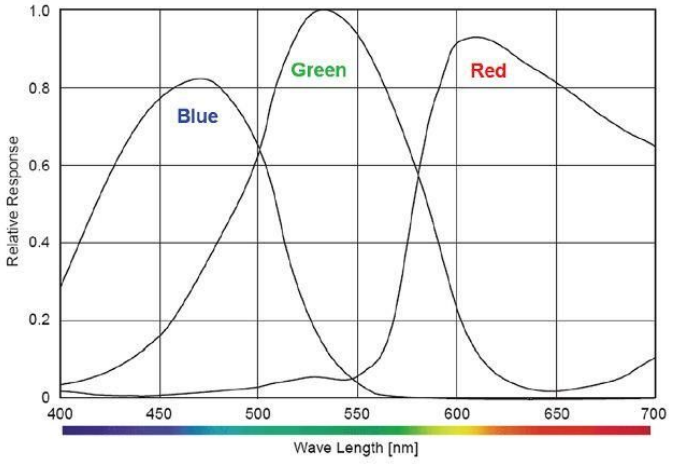


Figure 3.16 Relative responses of RGB color channels.

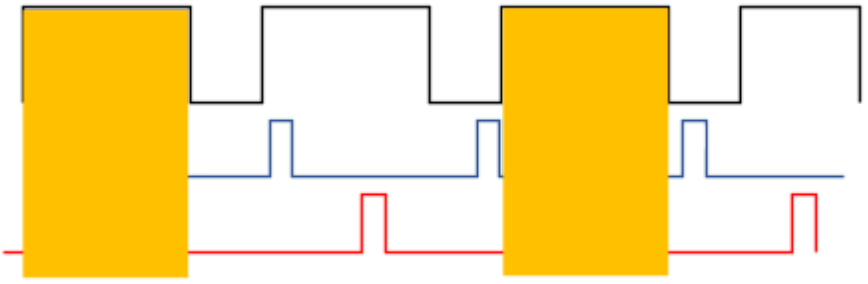
Image Acquisition

In our method, red and blue flashers are flashed one after another with a time delay (see Figure 3.17). Commonly used CMOS cameras have electronic rolling shutter. Syncing of shutter with flashers is needed to get all frames correctly double exposed. However, this feature is not available in most digital cameras. Therefore, incorrectly exposed or non-exposed frames needs to be rejected before processing. It is done by comparing intensities of red and blue colors within a sequence of images. When these two light pulses are exposed into a single frame, it contains overlapped two images, one in red color, one in blue color. Then red and blue color channels are separated to have two grayscale images. After cross-correlation technique which we called *Point Around Comparing* is used to

identify shifted distance of obtained images.

Point Around Comparing

As the first step corners are detected in both grayscale images to determine good features for tracking. Then image having highest features is chosen as the first image. For each corner point in the first image, search area around the point is chosen and best matching position is calculated by searching in the corresponding area of second image. After that mode of matching distances for each point are calculated using a histogram.



Shutter

Blue flash

Red flash

Double exposed

Single exposed

Figure 3.17 Timing diagram of flash pulses and shutter openings.

Figure 3.18 Red and Blue channel separation.

3.3.4 STIV Algorithm

STIVAlgorithm class is implemented to calculate

the velocity using Space-Time Image Velocimetry. Figure 3.19 shows the process of direction calculation using this algorithm. Space-time image is calculated using set of consecutive frames as described in Section 2.3.2. In Figure 3.20 shows a space-time image constructed using the implemented space-time image building algorithm.

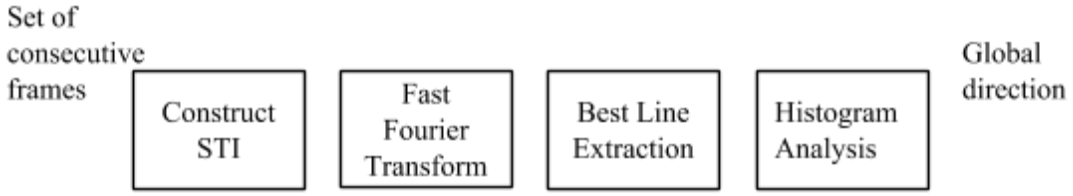
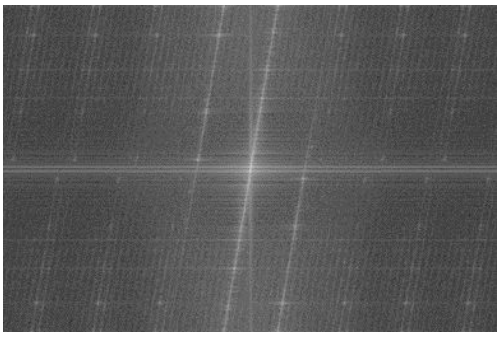


Figure 3.19 Process of direction calculation using STIV.

In STIV Technique velocity information are derived using the orientation angle of the STI. Orientation angle of STI should be extracted from STI. Structure tensor method [45] is widely used for derived the orientation angle, but it will not give the priority for strong features when calculating the local orientation. Therefore, a new technique is implemented using Fast Fourier Transform to identify the global orientation angle.

Fast Fourier Transform will transform an image from its spatial domain to its frequency domain. As digital images are discrete, Fourier Transform also needs to be of a discrete type. Therefore, Discrete Fourier Transform will be applied in this context. The DFT has become a mainstay of numerical computing in part because of a very fast algorithm for computing it, called the Fast Fourier Transform (FFT). The complex result of Fourier Transform is transformed to magnitude and apply logarithmic scale to amplify the small differences. After applying logarithmic scale small differences and large differences comes to the same level. In Figure 3.21 shows the resulting image after Transforming STI showed

in Figure 3.20 to Desecrate Fourier Transform and applying logarithmic scale.



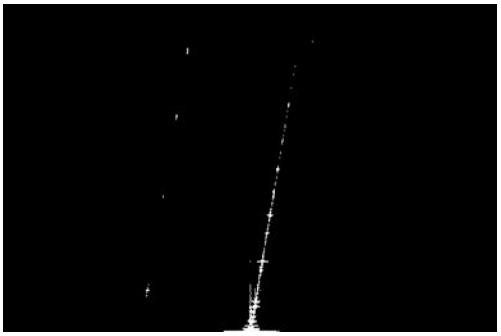
High changing values are displayed as white points and small changing values are black. Because the logarithm scale is applied, a gray-scale image which changing the pixels values from white to black according to the changing values will be generated. White straight lines appeared in the magnitude spectrum will give the orientation angle.

Best Line Extraction

Orientation angle should be calculated using the magnitude spectrum. Best line present in the magnitude spectrum will be used to calculate the orientation angle. For that purpose, best line should be extracted from the magnitude spectrum. Following steps are executed to extract the best line:

1. Create a new gray-scale image (filtered spectrum) with the same size of magnitude spectrum by initiating all pixel values to zero (completely black image will have generated).
2. Find the pixel positions from magnitude spectrum which have the white values (white pixels) in each row.
3. Assign white to those position in filtered spectrum.

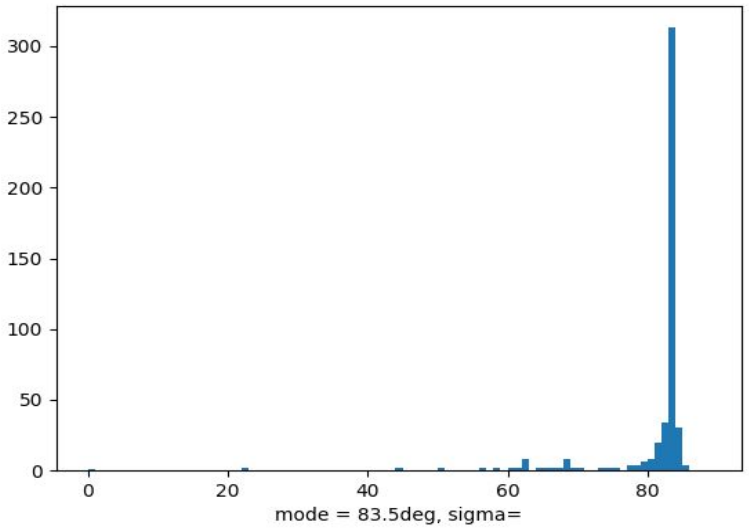
Figure 3.22 shows the filtered spectrum with the best line extraction which is generated using the magnitude spectrum as shown in Figure 3.21.



Histogram Analysis

To find the global direction, orientation of the best line should be calculated. Histogram analysis is used to identify the best line orientation as the global direction. White pixel positions in filtered spectrum will used to plot the histogram. Before plotting the histogram, pixel position coordinates should be converted to coordinates in first quadrant. Coordinates are converted by taking magnitude with respect to the center position coordinate of the image. Then all the white pixel positions will represent the coordinates in first quadrant. The angle made by each point will be calculated using converted coordinates. Calculated angles are represented from a histogram. Figure 3.23 shows the histogram plotted based on to the angles calculated using Figure 3.22 filtered spectrum.

Mode of the histogram can be considered as the orientation of best line others can be discarded as outliers. According to that mode values extract from histogram analysis can be used as the global orientation of the space-time image. That angle can be used to calculate the flow velocity using the STIV velocity calculation equation in 2.8


300
250
200
150
100
50
0
0
20
40
60
80
mode = 83.5deg, sigma=

4 SYSTEM IMPLEMENTATION

This chapter describes implementation of the flow meter. First four sections describe the implementation of Algorithm, Camera, Communication and Distance measurement modules with UML class diagrams respectively. Complete class diagram can be found in Appendix I. Section 4.5 describes how the debugging and initial configuration is facilitated in the system. It includes debugger module implementation along with web application implementation for monitoring internal state and setting environmental parameters. Circuit board manufacturing procedure is explained in Section 4.6. Module integration is discussed in the last section of this chapter.

4.1 Algorithm Module

Algorithm module is implemented as an abstract class while enforcing the flow of the data to be processed. Algorithm module is capable of analyzing the retrieved frame using an algorithm and returns a value for velocity in pixels per second. Methods named *getName*, *receiveFrame*, *update*, and *calulatePixelsPerSecond* needed to be implemented in every algorithm inherited by the abstract algorithm, which is responsible to return name of the defined algorithm, how the frame is received from the camera module, analyze and process the frame to identify and calculate velocity components, and calculate velocity in the pixels per second respectively. We implemented PIV Three Frames, PIV Color Channels, and STIV algorithms. In Figure 4.1 shows the class hierarchy of algorithm module. Another feature in algorithm module is, it uses a debugger module. Debugger module is used to debug the algorithms showing the inner states of the frames to the visualization attribute. Methods *getVsualization* and *getState,* which is inherited from the *Debuggable* interface includes the common behaviors of the algorithms. *getVisualization* method should return the content assign to *visualization* attribute. The content assigned to visualization attribute may differ from algorithm to algorithm. So, the algorithm is responsible for what is assigned to the visualization attribute and debugger is responsible for visualizing that. Internal states of the algorithm should be return through *getState* method. Algorithm type and the pixel per second value are the internal state of the algorithm. So, *getState* method returns the values assigned for *algorithm_name* and

pixels_per_second. Following sections describes the details and technologies used in implemented algorithms, PIV Three Frames, PIV Color Channels, and STIV Algorithms.

4.1.1 PIV Algorithm

PIVAlgorithm class supports basic functionalities support by the Particle Image Velocimetry technique, which calculates the velocity using Template Matching Technique. In the process of PIV algorithm, first it adds pre-filters to the image frames. Method *processPreFilters* is implemented in subclasses according to the context of the algorithm. *Update* is a method inherited from the *Algorithm* super class, which needs to be implemented in sub-classes. Most important method in *PIVAlgorithm* is *matchTemplate,* which is responsible for comparing two consecutive frames and return the matched coordinates of the second frame compared to the template in the first frame. This was implemented using OpenCV library [52] feature named matchTemplate.

To find a proper template in first frame *findGoodTemplate* method was used. Template was selected considering the white count in foreground mask image. To improve the precision of template selection method cluster was implemented. DBSCAN algorithm implemented in Python sklearn.cluster module is used to identify the clusters present in foreground mask image. Parameter values for match with the context are Eps = 3 (optimal epsilon value) and MinPts = 10. In the final step of the PIV process is validation, which varies according to the used method. Therfore, the validation is implemented in sub-classes accordingly.

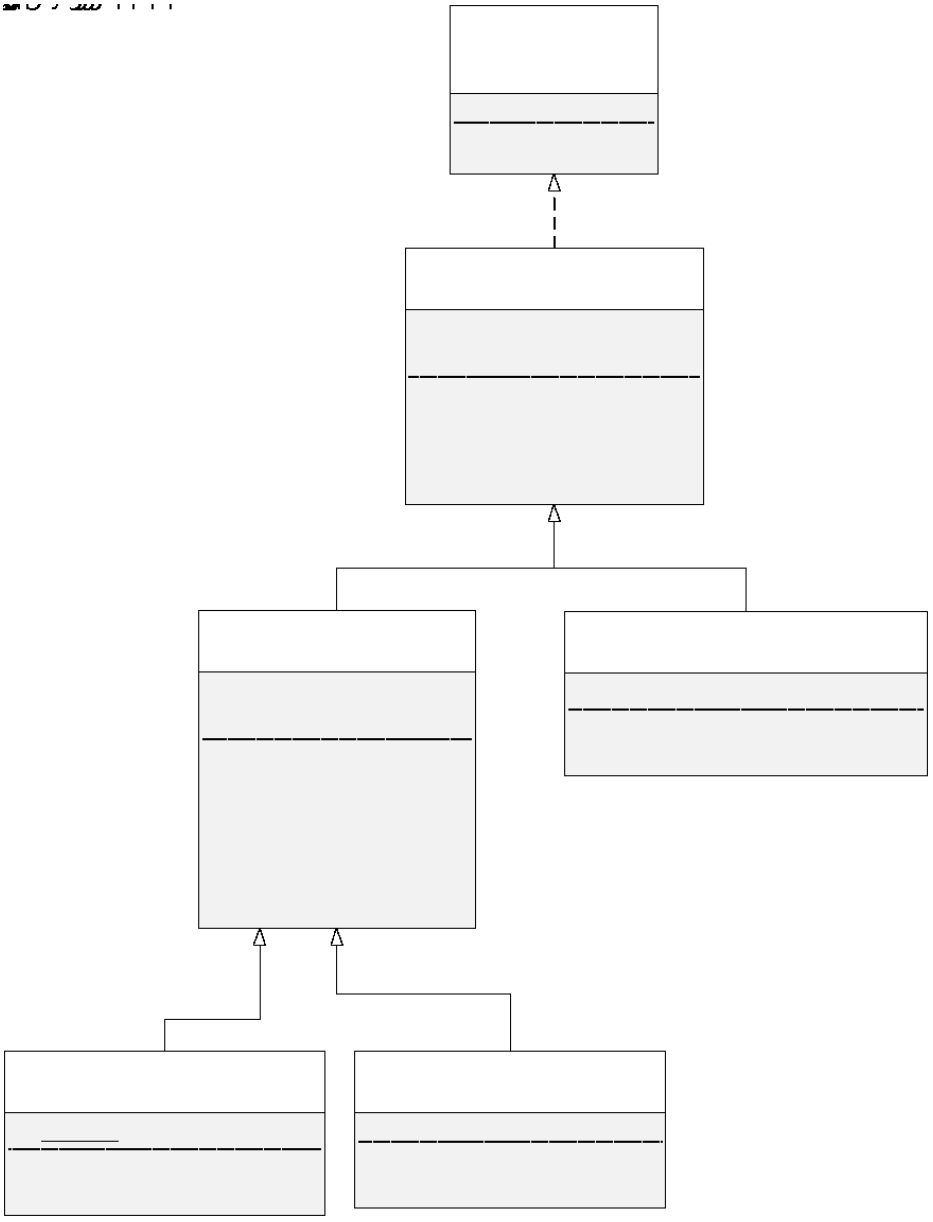


Figure 4.1 Implementations of algorithm module.

4.1.2 PIV Three Frames Algorithm

PIVThreeFramesAlgorithm class calculates the velocity using three consecutive frames. Since its followed Template matching technique, it is implemented by extending *PIVAlgorithm* class. In this algorithm, template matching is applied two times. The matching between first and second frames are done as implemented in *matchTemplate* method in *PIVAlgorithm* class. The matching between second and third frames happens in a different way as implemented in *findSecondTemplate* method in *PIVThreeFrames* class.

In this method, first matching coordinates are used to calculate the template for the second matching.

processPreFilters method defined in parent class is overridden here to apply background subtractor and morphological filters. Algorithms implemented in OpenCV library are used for background subtraction. Gaussian Mixture-based Background/Foreground Segmentation Algorithm is applied. We set history (i.e., number of last frames which used to generate the background model) to 20 and threshold to 10.

Morphological operations also applied in *processPreFilter* method. Morphological operations implemented in Python *scikit-image* module is used to apply morphological operation.

findDirection method is implemented to apply the validation. First matching displacement and second matching displacement is calculated here to check whether those two values are consistent. A threshold value is applied to filter the pixel distances of correctly matched templates.

4.1.3 PIV Color Channels Algorithm

PIVColorChannels class was implemented using a new method which is based on Bayer filter of CMOS color cameras. In this method template matching can be done on a single frame that is obtained by color separating doubly exposed frames. To identify an optimal template in one channel, use *goodFeaturesToTrack* method in OpenCV library [53]. That method will result a set of good points in the frame. That enables identifying good templates to match between two color separated channels. All the good points are matched in two frames and calculated the pixel distance. Based on those calculated distances, it plots a histogram to identify the mode to reduce the effect from the outliers.

Two get double exposed frames camera shutter and two flashers needed to be synced. Two flashers are driven using a IRF540 MOSFET [54] and switching signal is generated by a Atmega238p [55] micro controller. Raspberry Pi camera [56] is used for the image acquisition as it has the capability of adjusting the shutter speed which is essential to get

sync shutter and the flashers. Lighting conditions are controlled with the use of shutter speed, frame rate and ISO value in the Raspberry Pi camera. In flashers, illumination intensity of red light is higher than the blue light. Voltage given to the two flashers are controlled to balance the illumination intensities in a captured frame. Red flasher is operated in low voltage than in blue flasher to get an equal exposure in double exposed frame.

4.1.4 STIV Algorithm

STIVAlgorithm class is implemented to calculate the velocity using the orientation angle of space-time image using Space-Time Image Velocimetry. Since this technique is different from the PIV implementation and supports the basic functionalities of an algorithm, it is inherited from the Algorithm base class. When receiving a frame, that frame should be used to construct the space-time image. Therefore, the *receiveFrame* method is implemented to build the STI when a frame is received.

buildSTI is the method which builds the space-time image. STI is calculated using a set of consecutive frames as described in Section 2.3.2. currently received frame and most recently received frame are used to implement the STI. For the implementation of the magnitude spectrum, FFT models built in python *NumPy* fundamental module [57] are used.

getFilteredSpectrum method is implemented to extract best line from the magnitude spectrum and to build the filtered spectrum. In Histogram analysis stage, histogram implemented in *NumPy* is used to find the mode. That mode value is the global orientation calculated in STIV Technique. Each time of that calculation that value will be stored in *global_direction* attribute. That orientation can be accessed through *getDirection* method.

4.2 Camera Module

Image capturing of the drainage pipes is one of the core requirement in this system. All the analysis and processing are done on captured frames to calculate the flow velocity in the drainages. There can be various camera types plugged into the system to be used to capture

frames. Implementation have been done for Raspberry Pi camera, web camera, and from video camera. Figure 4.2 shows the overview of the implementation of the different camera types implemented.

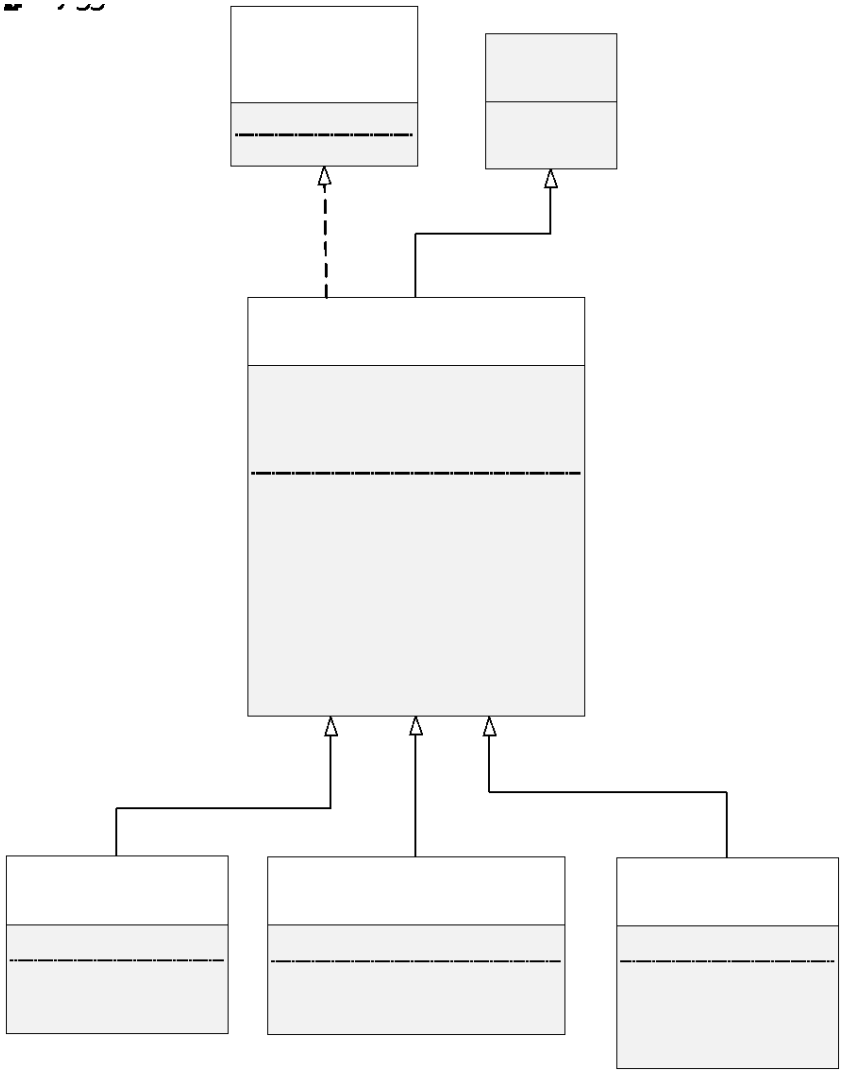


Figure 4.2 Implementations of camera module.

4.2.1 Web camera

WebCamera is a concrete implementation using abstract class Camera. This class was implemented to handle the operations in a web camera for image capturing, storing, and retrieving frames. Because *WebCamera* is an inherited class from abstract camera class it needs to implement the methods process and release. Image acquisition is done in the process method. Video capturing is done using a OpenCV library interface [58] for Python. After image acquisition frames were resized into standard resolution where the system operates. Captured frames are stored in a queue of frames which can be retrieved

using the *getFrame* method in camera module. After capturing process finished or when it came out of the loop then the *release* method will evoke to release the camera.

Web cameras could change the shutter speed according to the situation. However, that does not seem to be useful in this system because the shutter speed cannot be set manually. So, it is one of the disadvantage in web cameras. In web cameras all the processing after image acquisition is done in a built-in processor inside the camera which lead to give a low frame rate. To have non-blurred images in the drainages high frame rate is essential. Maximum achievable frame rate of a normal we camera limited to 30 frames per second. Web camera image data transmission is done using a USB cable. It supports only for law distances, if the distance is high the image data will not transmit properly.

4.2.2 Raspberry Pi camera

Among other implementations of the camera module, Raspberry Pi camera [56] seems the best suit for the addressed problem. *RPiCamera* is another concrete implementation of the camera. In the *process* method in the camera module, the amount of light that should be received by the capturing image, will be manually adjusted by the code considering the existing light conditions.

Raspberry Pi camera itself supply the facilities to manually adjust frame rate, ISO, shutter speed, etc. *RPiCamera* class has implemented using *picamera* module for image capturing, *numpy* library for store images in numpy arrays and OpenCV library to other processing in the frame. Pi camera modules have a discrete set of sensor modes that they can use to output data to the GPU.

Table 4.2 show the different sensor modes the *picamera* module supports and their limitations. According to the requirements, sensor mode 7 has been selected to meet the optimal performance using Raspberry Pi camera. *Picamera* module gives a method to capture frame sequence that can be configured according to the requirement. In capture sequence method in what format frames should be saved can be given as an argument. Another feature in *picamera* module gives is video port function. When assigning *video port* true will increase the frame rate because in *still port* uses noise reduction algorithms

in frames which will lead to decrease the performance of the image capturing.

Table 4.1 Sensor modes in Raspberry Pi camera.

#	Resolution	Aspect Ratio	Framerates	Video	Image	FoV	Binning
1	1920x1080	16:9	1 < fps <= 30	x		Partial	None
2	2592x1944	4:3	1 < fps <= 15	x	x	Full	None
3	2592x1944	4:3	1/6 <= fps <= 1	x	x	Full	None
4	1296x972	4:3	1 < fps <= 42	x		Full	2x2
5	1296x730	16:9	1 < fps <= 49	x		Full	2x2
6	640x480	4:3	42 < fps <= 60	x		Full	4x4
7	640x480	4:3	60 < fps <= 90	x		Full	4x4

Source: [59]

Our *RPiCamera* implementation supports changing the shutter speed of the camera with respect to the water level of the drainage flow to control the light intensity of the drainage that affects the image quality of the capturing image.

4.2.3 From video camera

FromVideoCamera is the implementation which uses a saved video from the memory to capture frames rather than real time capturing. Camera can be used to record a video and save in the memory according to the requirements within a given time frame. From those videos, frames need to be captured and kept in the queue used in the implementations. The path to the video should be given at the initiation of an object from *FromVideoCamera.* Frame rate of the video needed to be mapped with the frame rate of the camera which used to get the video before processing.

4.3 Communicator Module

This module is responsible of transmitting calculated measurements to an IOT server using a GSM modem. In a case of change of GSM modem, relevant communicator module should be implemented. In this implementation the AWSIoT [60] is used as the IoT gateway and the MQTT protocol is used for this IoT communication. The communicator runs in its own thread. Figure 4.3 shows the implementation of communicator module.

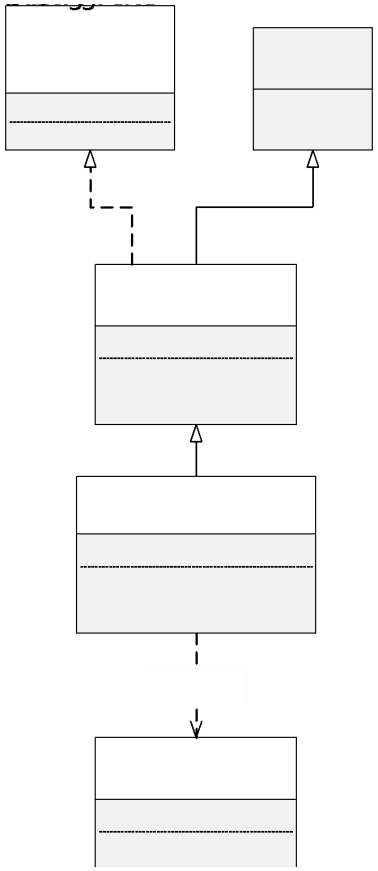


Figure 4.3 Implementations of communicator module.

4.4 Distance Measurement

The distance is estimated using an ultrasonic sensor. We used JSN SR04 [61] sensor as it is waterproof. The sensor can accurately measure up to a minimum distance of 5cm and a maximum of 2.8m with a least count of 0.5 cm.

The module includes a transmitter, receiver, and a control circuit. First a trigger pulse is sent to the signal pin of the sensor. Then the sensor module automatically sends eight 40 KHz signal and detect whether a pulse signal comes back (echo pulse). Then the distance is calculated using the time difference and the speed of wind. In this implementation when the distance measurement is needed, the sensor is automatically driven by the Raspberry pi device and the measurement values are obtained.

4.5 Debugger Module & Configuration Interface

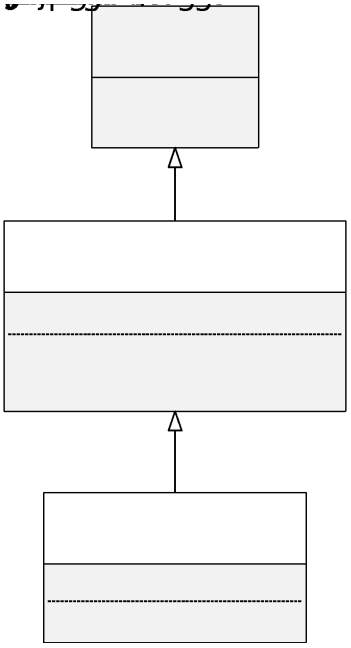
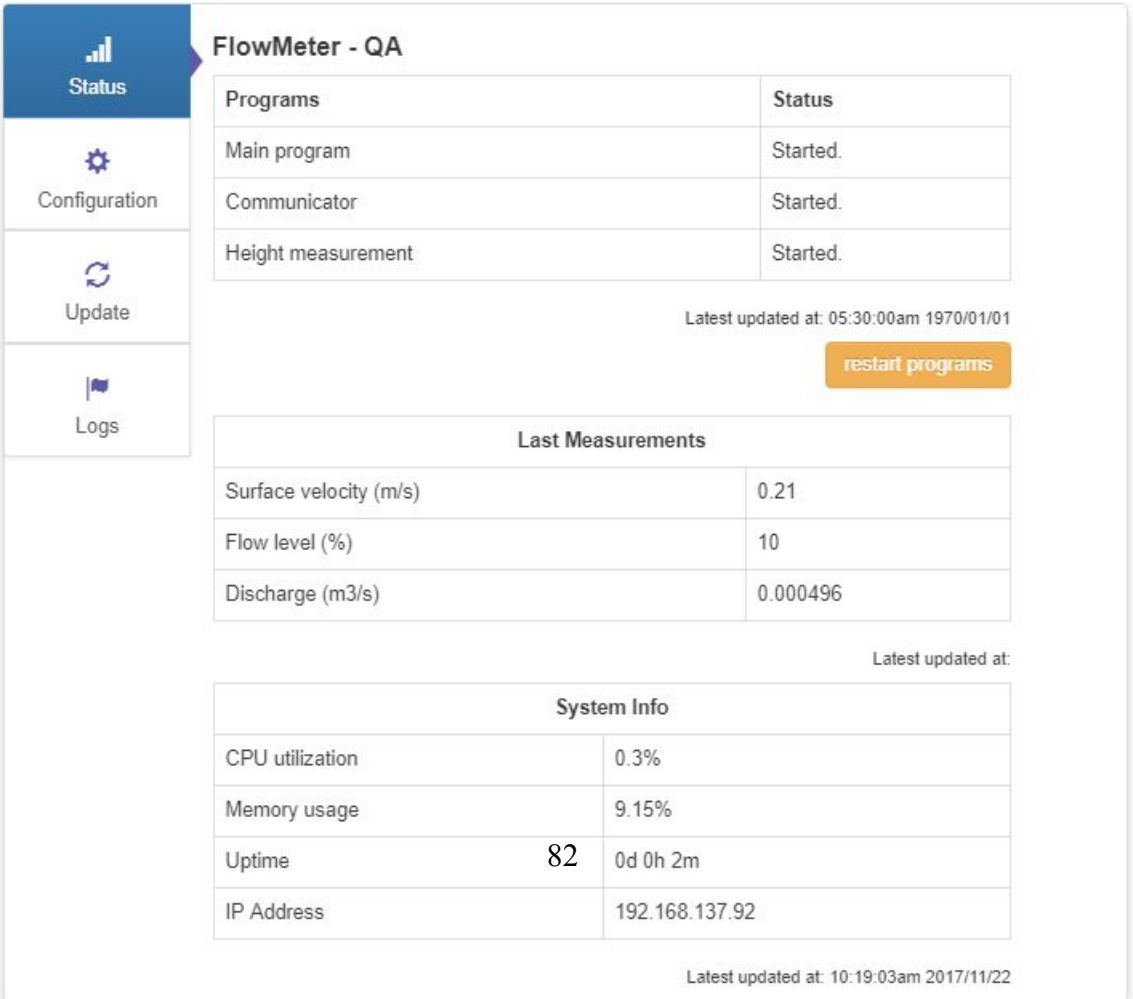


Figure 4.4 Implementations of debugger module.

Debugger module is used for debugging purposes of the system. All the debugging related works are assigned to another thread. Figure 4.4 shows the overview of debugger module. Debugger abstract implication must be extended and should implement the run method according to the debugging purpose. Concrete *getStateObject* method is responsible for returning the current state of the camera including camera type, frame rate, resolution, etc., and the state of the algorithm including algorithm type and the velocity in pixels per second.

sing the data that are being taken through the *getStateObject* method in debugger module, a web application is provided to the user. Figure 4.5 shows a screenshot of the web application. Using the status section of the application, the user will get the information about the main three programs (main program, communicator and height measurement), latest measurement values and system properties. Using the restart program button shown in Figure 4.5, the above said programs can be restarted.


Status
Configuration
Update
Logs
FlowMeter - QA
Programs
Status
Main program
Started.
Communicator
Started.
Height measurement
Started.
Latest updated at: 05:30:00am 1970/01/01
restart programs
Last Measurements
Surface velocity (m/s)
0.21
Flow level (%)
10
Discharge (m3/s)
0.000496
Latest updated at:
System Info
CPU utilization
0.3%
Memory usage
9.15%
Uptime
0d 0h 2m
IP Address
192.168.137.92
Latest updated at: 10:19:03am 2017/11/22

Using the configuration section, a user is able do the initial configuration of the device. Also, the drainage parameters and location details can be updated here. Using this web application, a user can view the system logs and it supports the software package update functionality as well. Configuration and Logs user interfaces are attached in Appendix III.

4.6 Circuit Boards Manufacturing

The system consists of various replicable components. Printed circuit boards are necessary for proper connectivity between hardware modules. It also minimizes cohesiveness between components and reduces human faults while installation. While designing the PCB commonly using sockets and connectors used as much as possible to keep the extendibility. Figure 4.5 and 4.6 shows the SBC related schematic diagram of Head PCB. *VCC* and *BUCK_IN* nets corresponds to regulated 5v power line and unregulated power receiving line respectively. Table 4.2 summaries the connector types used for ports.

Table 4.2 Summary of the connector types used for ports.

Port (s)	Interface
P1, P10	RJ45 socket
P11, P5	USB type A female
P7	IDE socket
P3	4-way square

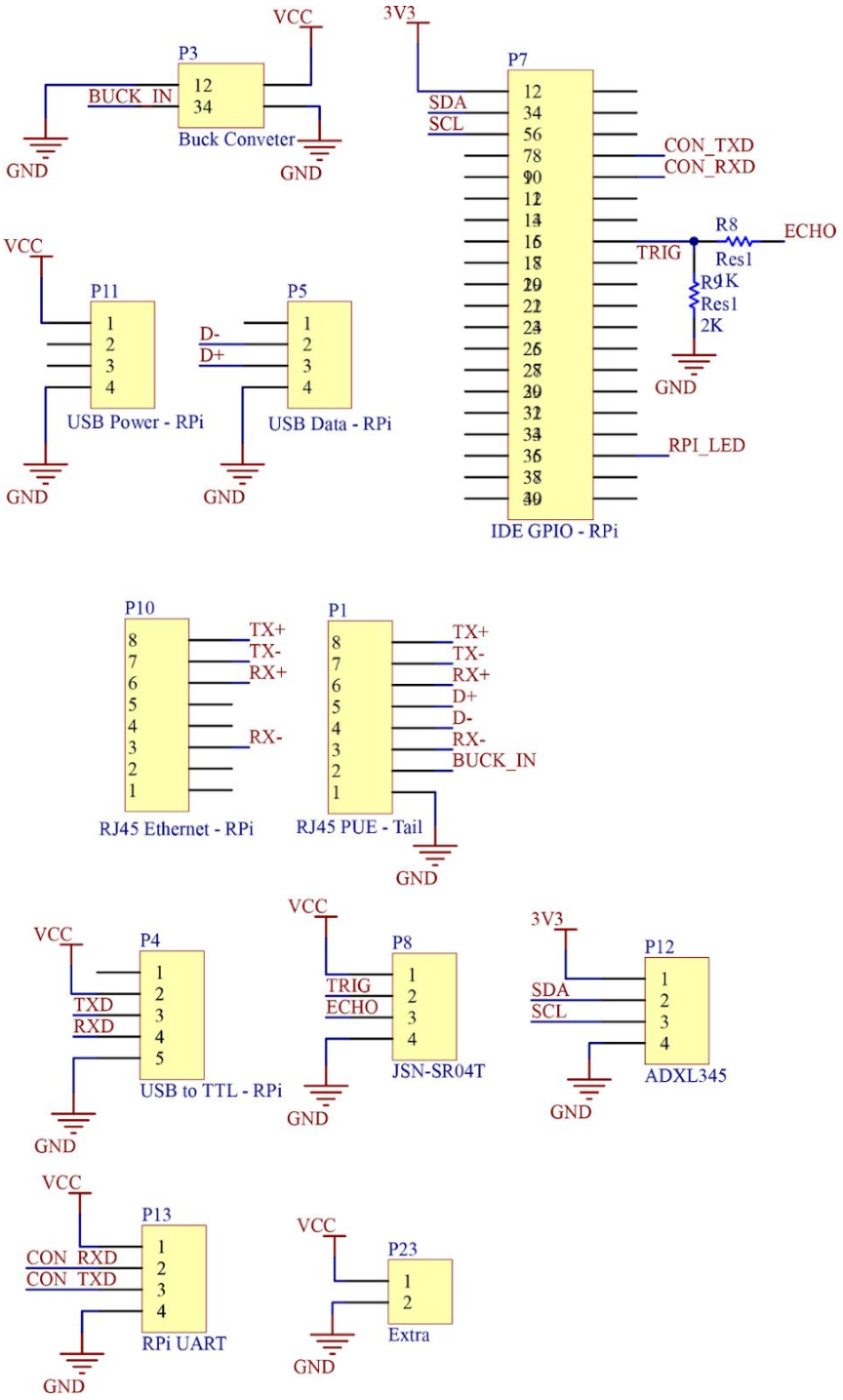


Figure 4.6 Schematic diagram of Head PCB – Region 1.

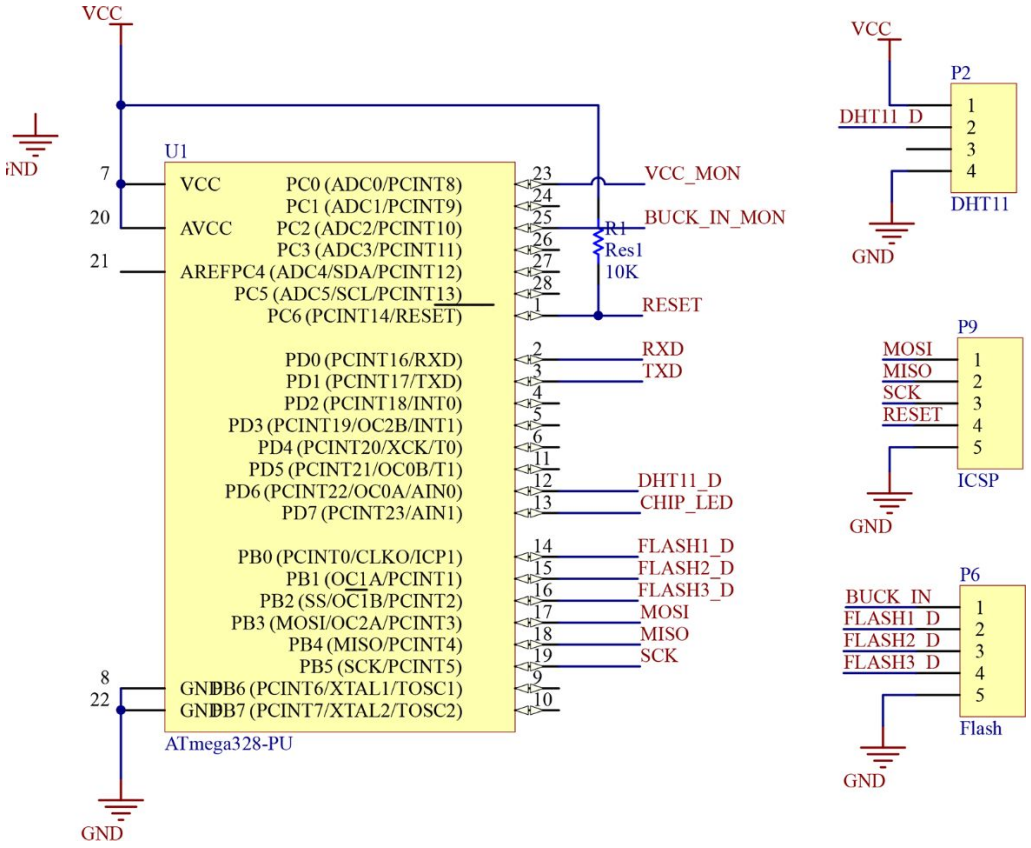


Figure 4.7 Schematic diagram of Head PCB – Region 2.

Figure 4.5 and 4.6 shows the schematic diagram of on-board chip related components in Head PCB board. P2 can be used as I2C bus to connect on board sensors. P9 is the ICSP port for the on-board chip. That port used for downloading bootloader to the chip which is not a regularly used. Lighting system connects to P6 and BUCK_IN net is the unregulated power receiving line. Figure 4.7 is the schematic diagram for the Tail PCB. P4 and P6 are optional sockets those will be used in a situation where power transmission needs a boost conversion and battery charging mechanism exist.

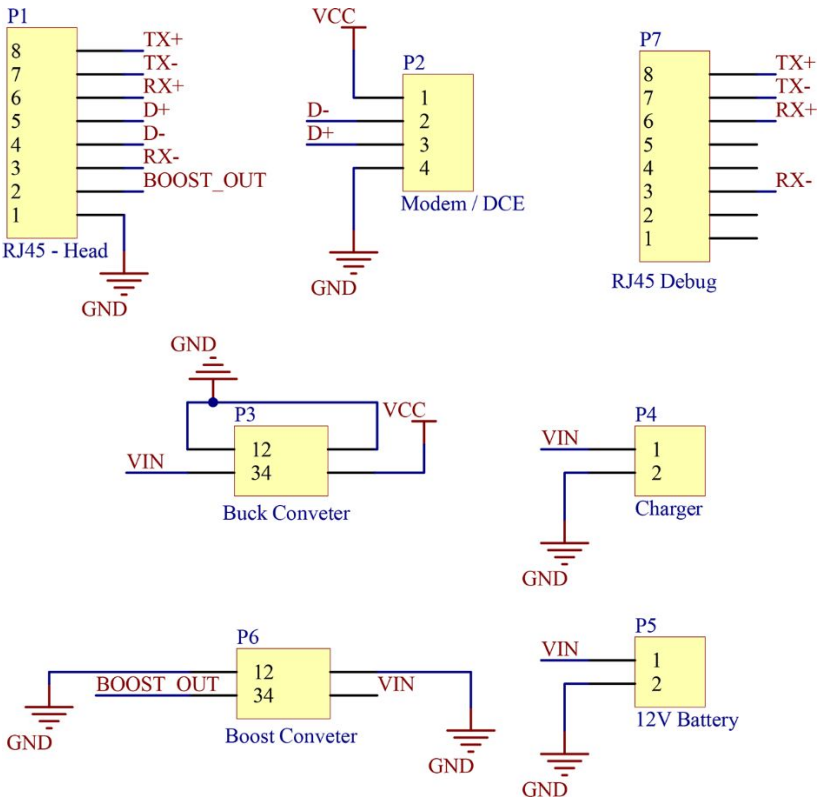


Figure 4.8 Schematic diagram of Tail PCB.

Two through hole PCBs were manufactured for the Head and Tail devices. We used photoengraving method which is a subtractive PCB manufacturing process. First component placing, and routing was done using schematic diagram. Altium Designer 16 [62] was used as PCB design software since it is widely used in industrial applications.

First PCB footprints of uncommon sockets were drawn according to the datasheet drawing. We had to draw footprints of RJ45 socket and USB type A socket. Other footprints were available as libraries. After, component placing was done manually to have the best physical arrangement. USB interfaces placed near to the edge of the board. Power receiving connector and Raspberry Pi power supply port kept near to reduce path length. Further, placing was adjusted while routing; to have the optimum routing configuration with lesser number of jumper wires. Path widths and path clearances were main routing considerations.

Figure 4.9

Routing rules configuration.

Minimum width restricted to 15 mils and power paths carrying more current drawn with 70 mils. Other lines were drawn at around 2 mils. Clearances are followed as in the Table 4.4. All the distances are in mils. Minimum clearance was set to 20 mils to avoid manufacturing errors. Ground net polygon was added to reduce cross talk between adjacent paths. It also reduces electrical noise.

Table 4.3 Minimum clearances between shapes.

	Track	**Pad**	**Poly**
Track	20		
Pad	20	20	
Poly	40	40	40

Procedure followed to get circuit board printed is as follows:

1. First, mirrored bottom layer (see Figure 4.9 blue area) was printed using a laser printer on a transparent sticker.
2. A copper clad board was cleaned using a metal polish.
3. A dry photo resistive film was placed on the top of copper clad board and heated until the photoresist glued to the copper board.
4. After printed sticker was put on the top of all those layers.
5. Then it was kept under ultraviolet light source for 20 minutes. Here printed sticker act as a mask to avoid exposing necessary areas to ultraviolet. As we used a positive photoresist, ultraviolet exposed area becomes soluble.
6. Sodium Hydroxide aqueous solution was used for developing the film.
7. Later PCB was etched using a Ferric Chloride solution.
8. Finally, photoresist was removed by applying Turpentine.

Then printed circuit board was drilled carefully using an electronic PCB drilling machine. After placing and soldering the components, a Polyurethane layer was applied as a conformal coating. Top layer (see Figure 4.9 red area) of the PCB used to define jumper wires. Intermediate states of head circuit board in manufacturing process can be seen in the Appendix II.

Ground Net

Polygon
Vcc Net Solid Region

Figure 4.10 Head PCB layout.

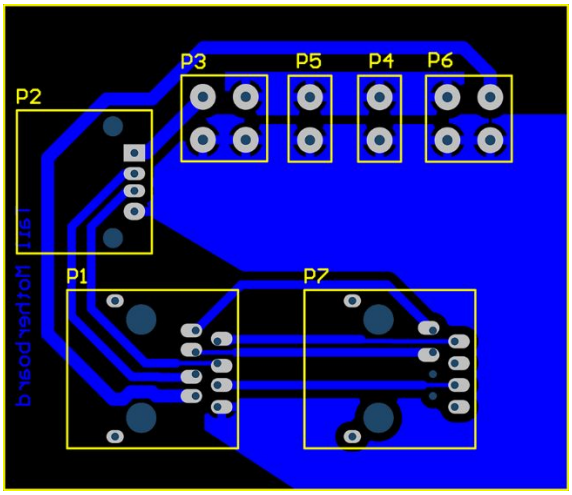


Figure 4.11 Tail PCB layout.

4.7 System Integration

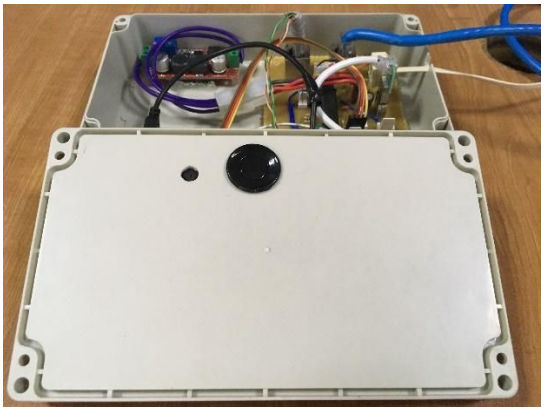
Finally, we integrated all the software modules and hardware components to have the final product. In the software implementation, three separate processes are used.

1. Device process: Handles retrieval of frames from camera.
2. Communicator process: Transmits measurements via modem.
3. Sensor process: Measures and prepare water level.

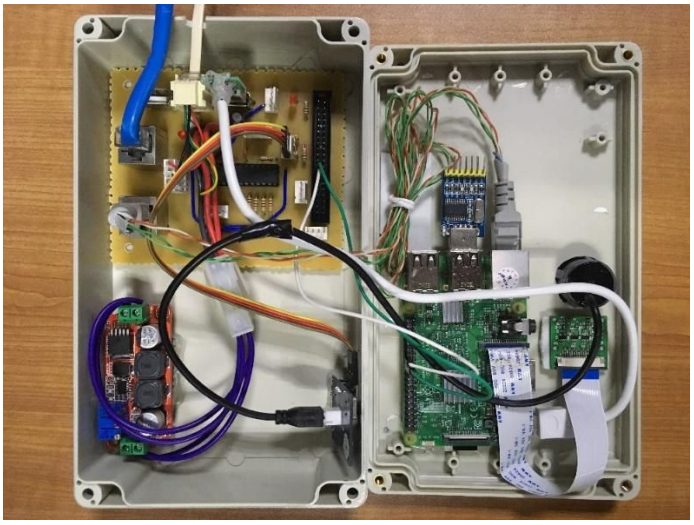
Main reason for using processes is Global Interpreter Lock (GIL) in python. It doesn't allow for threads in one process to execute concurrently. Since I/O operations are carried out by communicator module and distance sensor, CPU utilization will not be efficient if all operations are done in same process. These processes should be started after every restart of the device. This was done using a python script which can start three these processes and it registers itself as a cron job for running in every minute. In addition to these tasks, this script helps to configuration web application by serializing device state in to a file.

As hardware considerations, a separate enclosure is used for lighting system (see Figure 4.11) since in the production environment light will be a commercially off-the-shelf LED

flood light which can't be put with other components, mainly due to the heat dissipation. Component arrangement and inside view of the head device is in the Figure 4.12.



Light source
Head device
Camera opening
Ultrasonic sensor



Head Board
Step down converter
Ultrasonic sensor
Raspberry Pi camera
Raspberry Pi board
CAT6 cable – To Tail
RJ11 cable – To light

board
source

Figure 4.14 Component arrangement in Head

device.

5 PERFORMANCE EVALUATION

Reliability of the measurements of the system is based on the values we get from the algorithm and from the distance sensor. Algorithm module is responsible for give an accurate result for flow velocity and the sensor is responsible for give an accurate height measurement. In this chapter, first we evaluate our algorithm and conclude that how reliable the measurements we get for velocity. Next, section 5.2 concludes the accuracy of height measurements we get from ultrasonic sensor. At last we conclude the accuracy of pixels to real world coordinates mapping function.

5.1 Algorithm

The algorithm used in the device has evaluated considering many factors affecting the output velocity. Particle density is one factor that affect the accuracy of the algorithm. We have validated the algorithm using low, average and high density of particles in the flow. Then we have tested the algorithm by changing the velocity of the flow. Calculated pixel distance from the algorithm is validated by measured pixel distance from the Amped FIVE software [63]. Details about the algorithm validations are in below sections.

5.1.1 Particle density vs. discharge

PIV three-frame algorithm was chosen as the optimal algorithm among other algorithms to match with the given requirement. In PIV algorithms velocity is calculated by matching template matching. To identify a template and match that template between three frames, we need to have particles in the flow. Therefore, we evaluated the PIV three-frames algorithm with low, average, and high density of particles.

While maintaining all the other factors in the flow at same level, we changed the particle density of the flow and evaluated the PIV three frame algorithm. Flow velocity is maintained at same level by ensuring the pressure head of the water flow. Water was cycled through a PVC pipe with 2-inch diameter (see Figure 5.1) the water from a motor at constant speed. To circulate the water flow DC water pump is used. To evaluate the algorithm, we took three videos with different particle densities. While getting videos we calculated the flow discharge without using the algorithm to compare with the values given

from the algorithm. To measure the flow discharge we calculate the time taken for fill up a 1L bottle of water. That measured discharge is indicated in Figure 5.1, 5.2, and 5.3 in dotted line. Figure 5.1 shows the graphs by plotting the flow discharge for each frame classified into the particle density.

Raspberry Pi

Ultrasonic sensor

Camera

Movable trolley

Water container

Water pump

Figure 5.1 Video recording aperture with water circulation.

Figure 5.1 shows the flow discharge captured in each frame when there is low density of the particles in the flow. In all the three graphs shown related to the comparison of the particle density, two dashed line indicate the average discharge calculated from the algorithms for over 2000 frames taken from the Raspberry Pi camera and the measured discharge taken by measuring the time taken to fill up a 1L bottle. Figure 5.2 is related to the average particle density in the flow of the pipe. Comparing those three graphs, average discharge and the measured discharge is closer in average particle density graph. When there is low density in particles, template matching seems not to be optimal since to track particles in three frames will not be successful at every time. When particle density is high, it is not very accurate to use PIV three-frame algorithm, as it has many particles in the flow it may match to same particle and might give erroneous output.

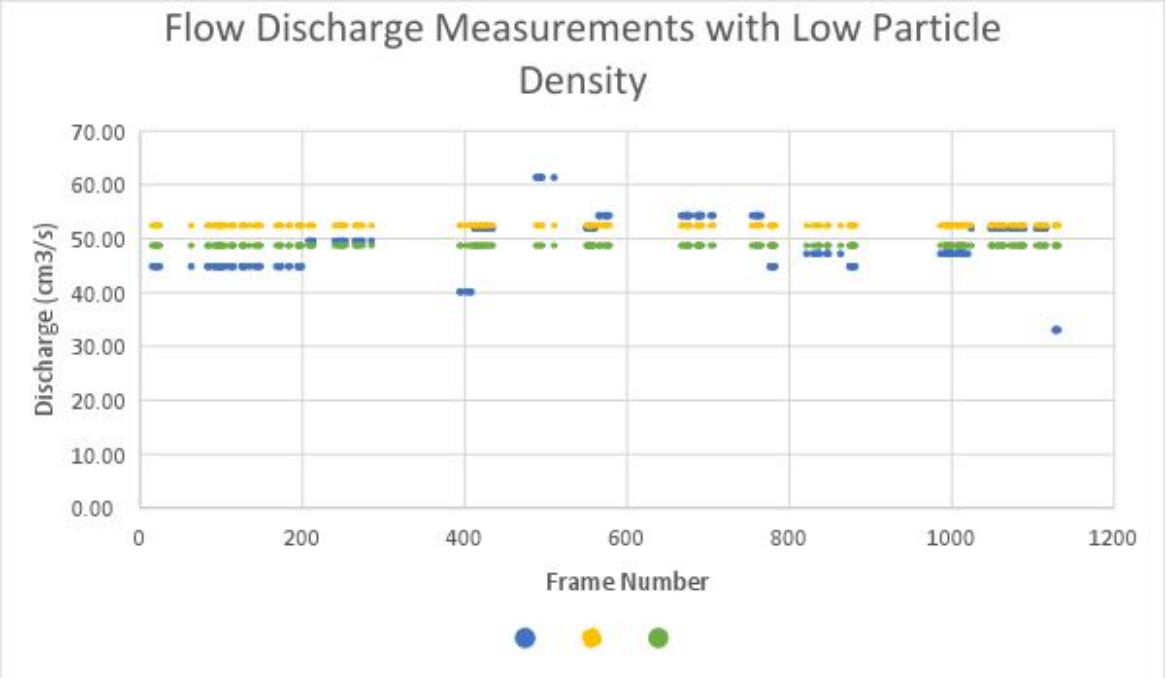


Figure 5.2 Flow discharge with low particle density.

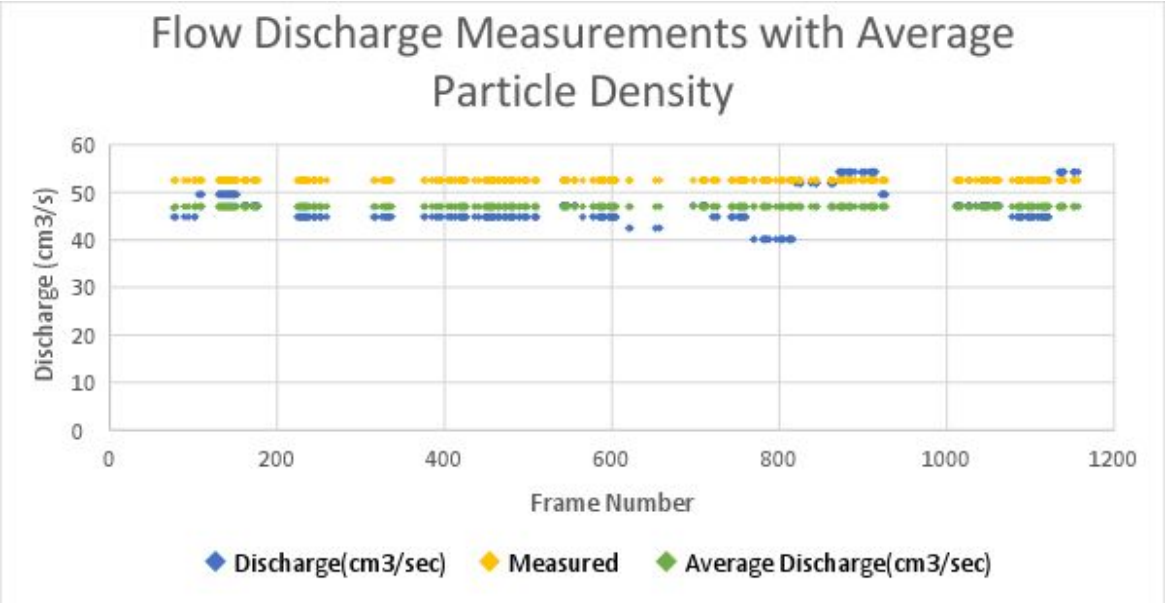


Figure 5.3 Flow discharge with average particle density.

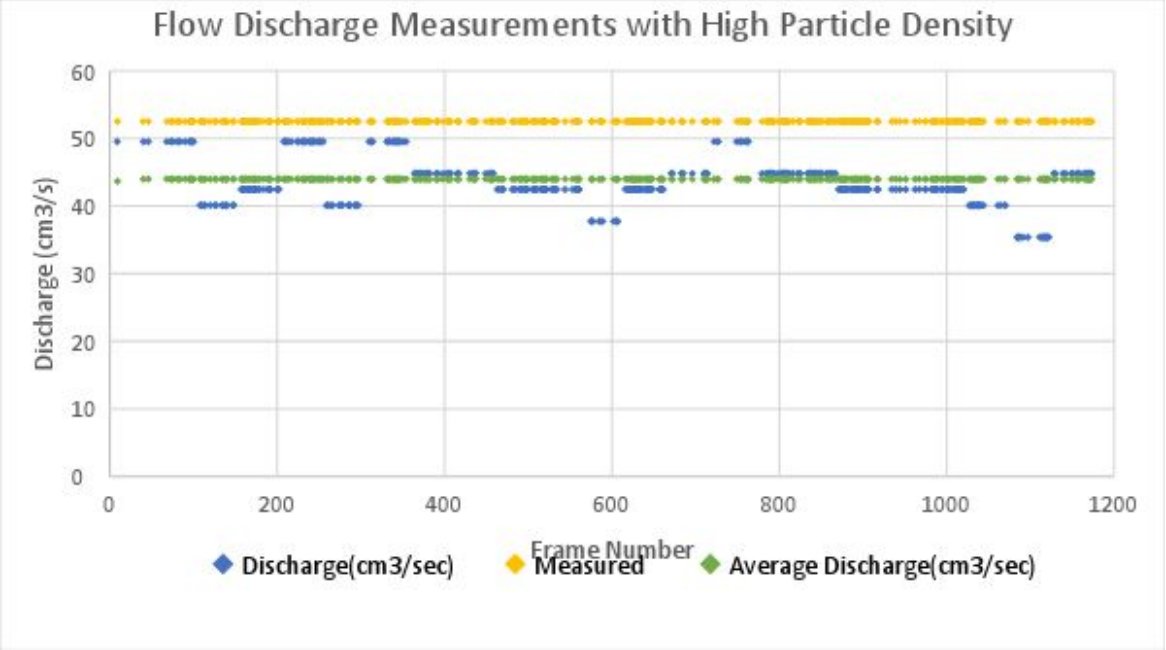


Figure 5.4 Flow discharge with high particle density.

5.1.2 Flow velocity vs. discharge

We evaluate the PIV three frame algorithm with the variation of flow velocity. Algorithm is evaluated in three situations with low, average and high velocity. Three videos are recorded while maintaining other factors at same level. Same apparatuses in Figure 5.1 is used here to evaluate the algorithm with the velocity variation. In this testing particle density is maintained in same level throughout the three situations. We changed the

velocity to low, average and high level by changing the pumping head of the water pump in the water circulation system. Low velocity is gained by keeping the pumping head of the water pump in a below level than the recorded flow level. Average velocity is the level that pumping head as same as the recorded flow level. High velocity mapped with the pumping head which is higher than the flow level.

Figure 5.4, 5.5, and 5.6 show the graphs by plotting the flow discharge for each frame classified into the velocity level. According to the velocity change, discharge was measured in both by manually and by algorithm. Manual process is same as mentioned in Section 5.1.1. That measured discharge is indicated in Figure 5.4, 5.5, and 5.6 with a dotted line.

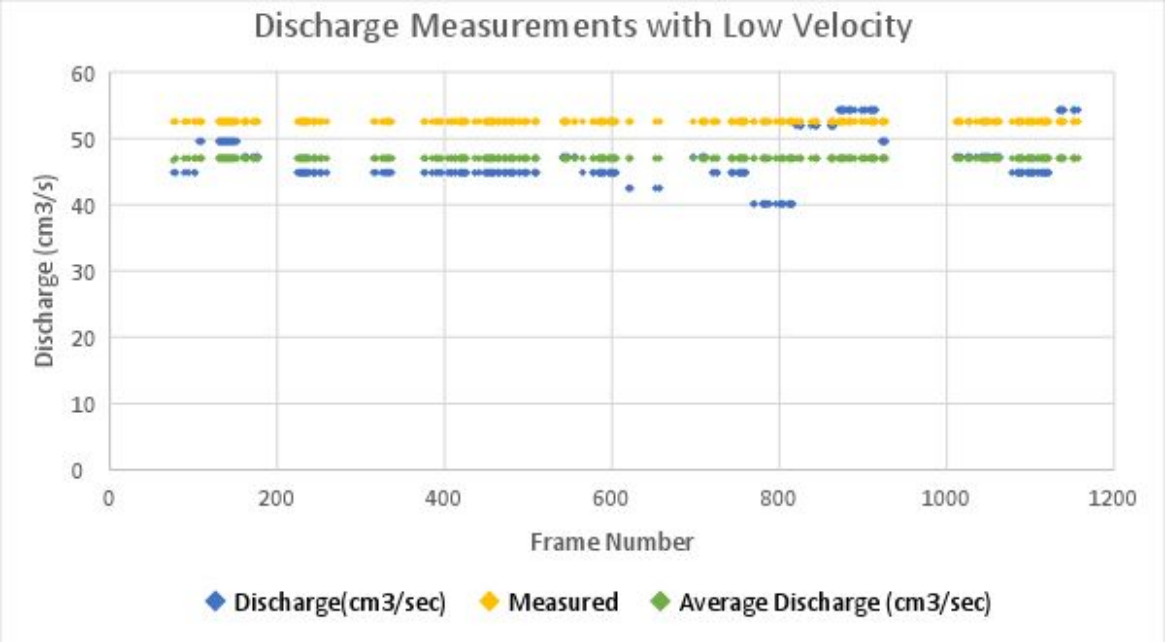


Figure 5.5 Discharge measurement with low velocity.

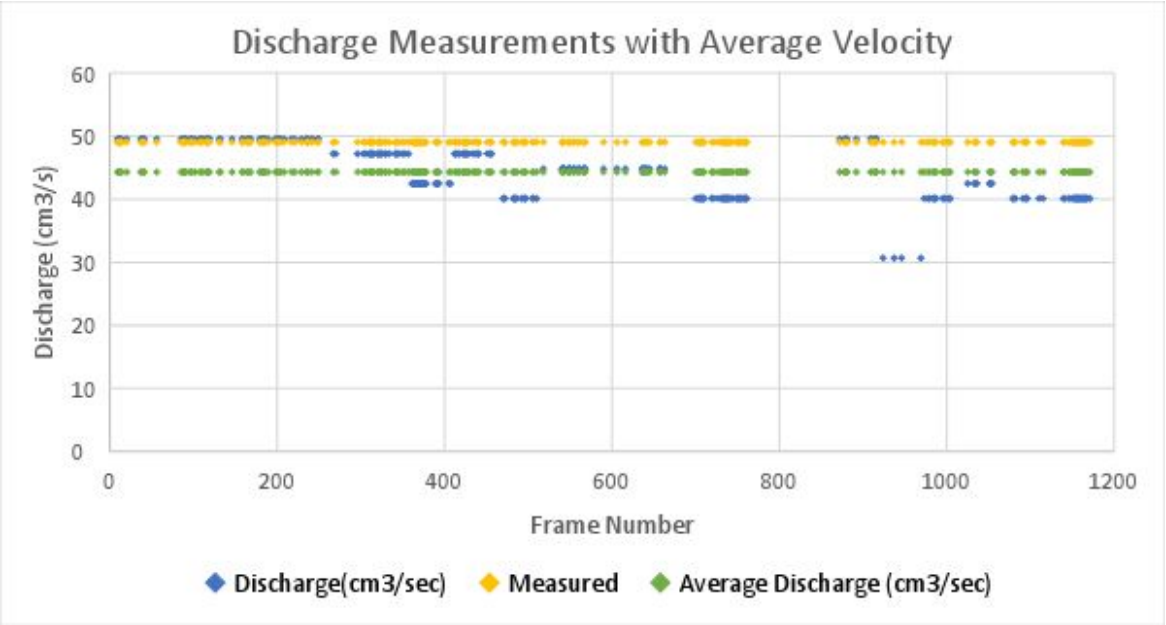


Figure 5.6 Discharge measurement with average velocity.

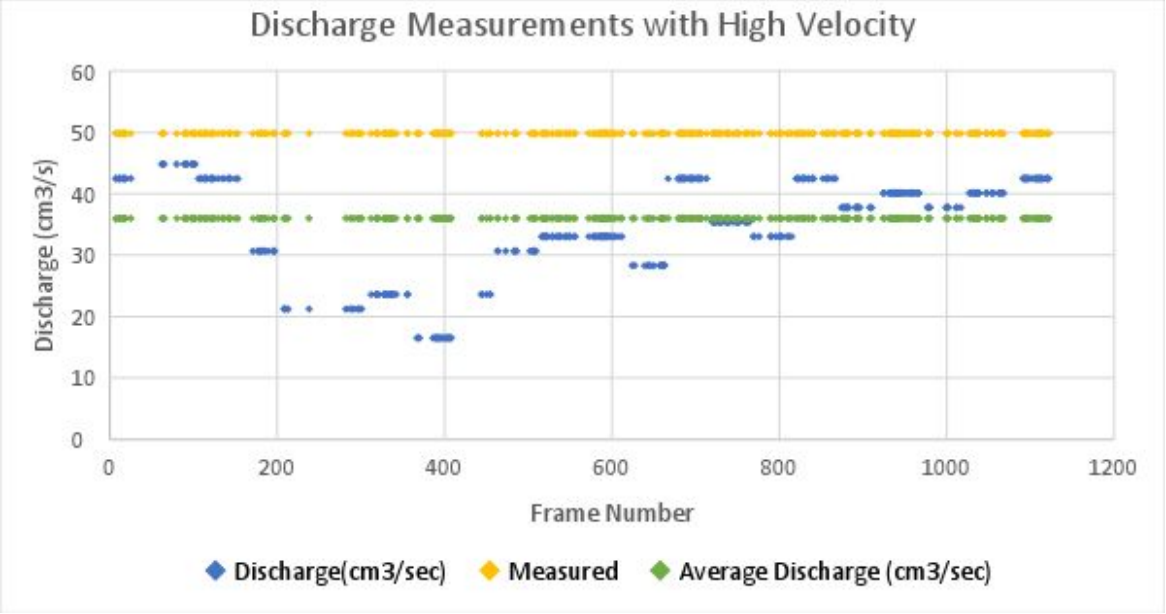


Figure 5.7 Discharge measurement with high velocity.

5.1.3 Validation of pixel distances

In PIV three frame algorithm, three consecutive frames are used for matching purpose and two matching cycles are happened in a process. After the matching, x and y pixel displacement is calculated using the average of two matchings in three consecutive frames. We implemented some validation process to uplift the accuracy of algorithm. We

evaluated how that validation process affect to uplift the accuracy of the algorithm. We used a recorded video of the flow with average level of particle density. In Figure 5.7 plotted the results given by the PIV three frame algorithm without applying the validation we implemented. According to the plotting there is a significant variation of x, y displacement. There are many false matching return from the template matching. Those false matching should be removed to uplift the accuracy.

We implemented a validation process to uplift the accuracy of PIV three frame algorithm. False matching must remove to uplift the accuracy. We implemented a direction filter which analyze the direction of the particle moved. In analyze the direction of first matching and direction of second matching. If the same particle is tracked from the templating matching algorithm, direction should be consistent with the flow. If two matchings give the same direction, we can categorize it as true matching. That process reduces the false matchings to some level. We increased the accuracy more by applying another validation. Although both matchings give the same direction there is a possibility that it still can be a false matching We validated the matching results in a way that can identified the false matchings happened for same direction. In Figure 5.8 shows the results we get after applying validation process for the same recorded video which used in Figure 5.7. When comparing those two plotting graphs it is clearly shown that our PIV three frame algorithm can remove the false matchings return from the template matching. In our product reliability is the main requirement so if we get a result it must more accurate. Rather than getting all true matching results with falser matching results, it is important to get some results which include true matchings without false matchings. So, in this situation we calculate the recall and precision for our validation function using following equations.

$$Recall = \frac{Retrieved\ points}{Relevant\ points} \times 100\% \quad (5.1)$$

$$Precision = \frac{Relevant\ Points}{Retrieved\ points} \times 100\% \quad (5.2)$$

We calculated the recall by checking how many relevant points in Figure 5.7 are retrieved to the Figure 5.8 by our validation process. It concludes that there is a 98% precision for

the validation process and 78% of recall.

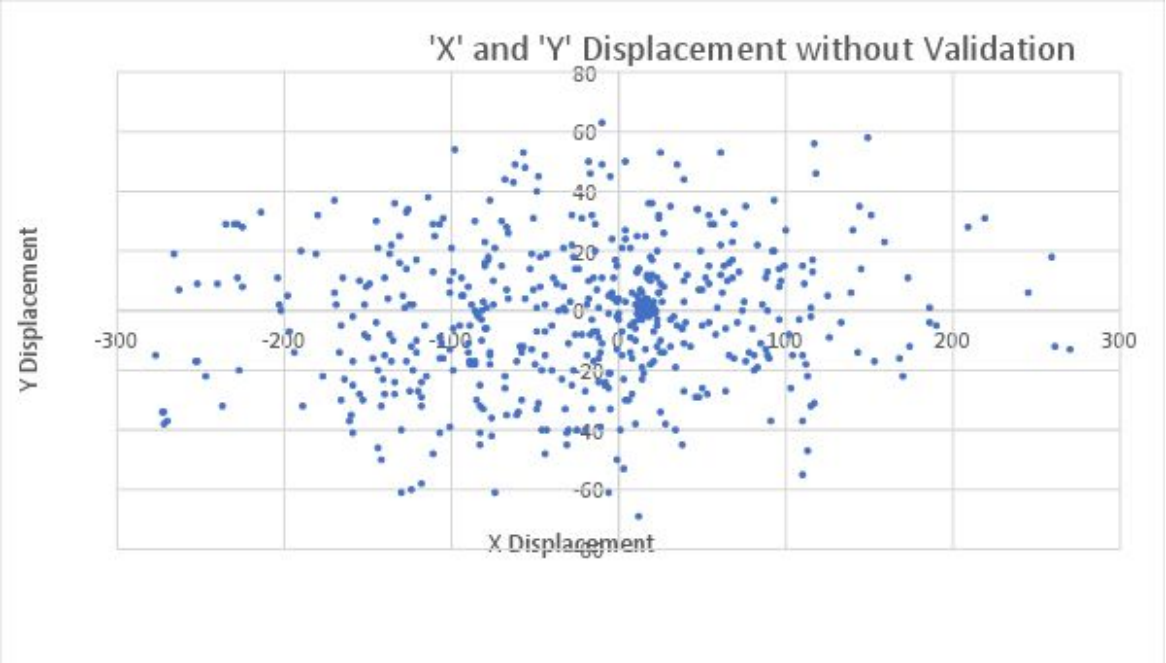


Figure 5.8 X and Y displacement results without validation.

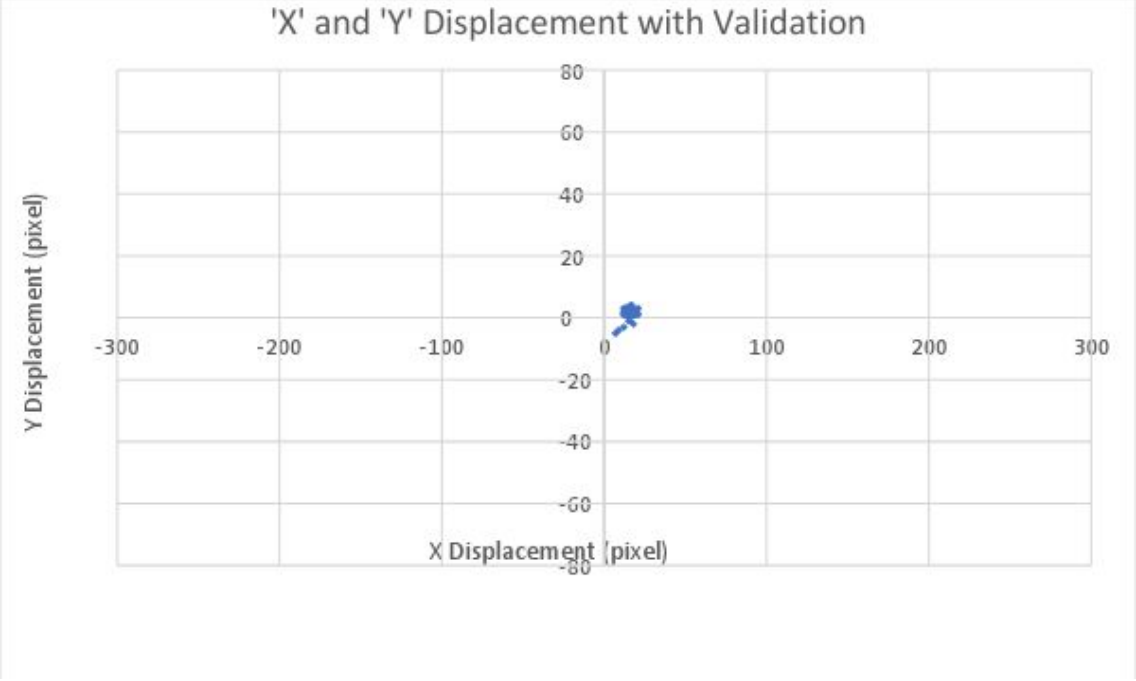


Figure 5.9 X and Y displacement results with validation.

5.1.4 Evaluation of pixel displacement

Velocity is calculated from the algorithm by measuring the distance moved from one frame to next consecutive frame with respect to a template selected in the first frame. In

the algorithm we calculate the distance moved between two frames in pixels. Therefore, the calculated pixel distance is validated by measuring the pixel distance moved from Amped FIVE software which is a leading forensic image and video enhancement software. Figure 5.9 shows the graph plotted displacement against frame number. Blue color dots indicate the calculated value given from the algorithm and Orange color dots represents the values measured from the Amped FIVE software.

We calculated the error in each pixel by following equation.

$$Error\ \% = \frac{|Algorithmic\ value - Measured\ value|}{Measured\ value} \times 100 \qquad (5.3)$$

We get an average error percentage by averaging all the error percentages for the given data set. With considered measurements (mean = 15.9 pixels, standard deviation = 1.02), we can conclude that calculating pixel distance is calculated with a 94.5% accuracy (100 – Error %) from the algorithm.

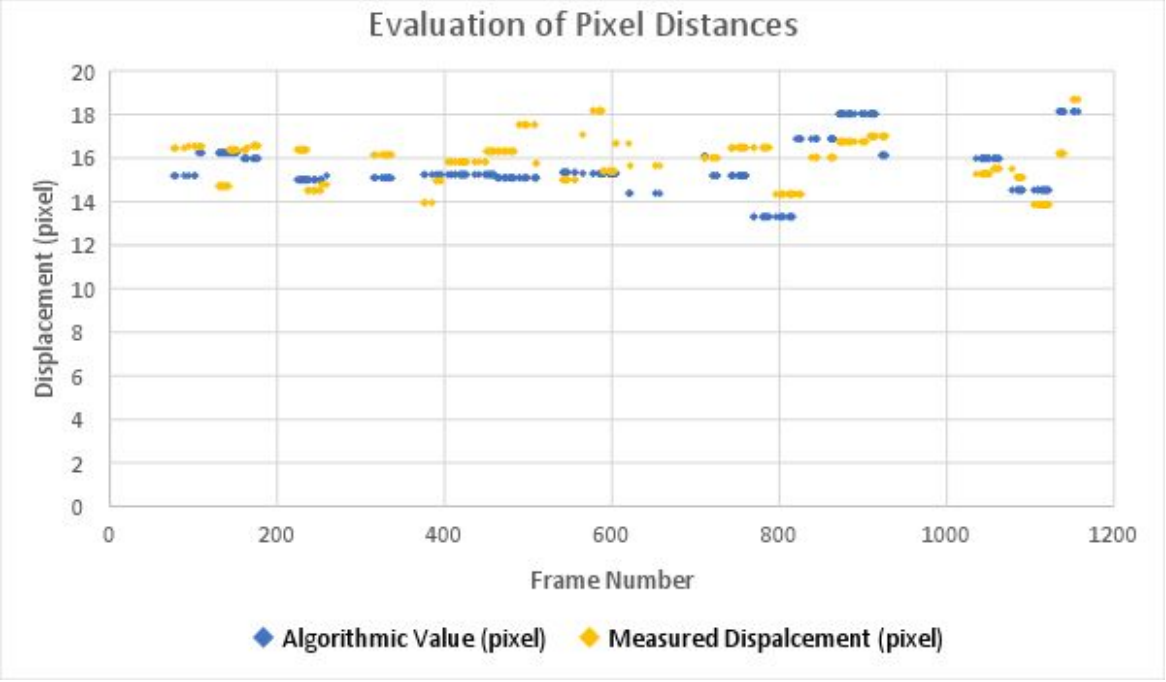


Figure 5.10 Pixel displacement measured from PIV three frame.

5.2 Height calculation

In the device, the height of the water level is calculated using an ultrasonic sensor.

Evaluation of the accuracy of this method is done using two experiments. First the distance to a solid object is measured and then the tests are conducted by measuring the height of an actual water surface.

5.2.1 Measuring the distance to a solid surface

A rectangular shaped solid board of size 31 cm × 22 cm is placed as shown in Figure 5.1 and a perpendicular line of 3m is drawn. The drawn line is marked with the distance values measured using a measurement tape taking the said board as the origin.

Then the sensor readings are recorded for different known distances up to 3m. The sensor gave possible output values up to 152 cm with the above said apparatus. But for values greater than 152 cm, the sensor readings were not accurate. Then the solid surface was replaced with a new rectangular shaped solid surface of size 100cm x 50cm. Then again, the sensor readings were taken from 150 cm to 300 cm and the possible values were returned by the sensor. Graph in Figure 5.12 shows the values obtained from the sensor, plotted against the actual distance values.

Figure 5.12 Distance measurement from a solid surface.

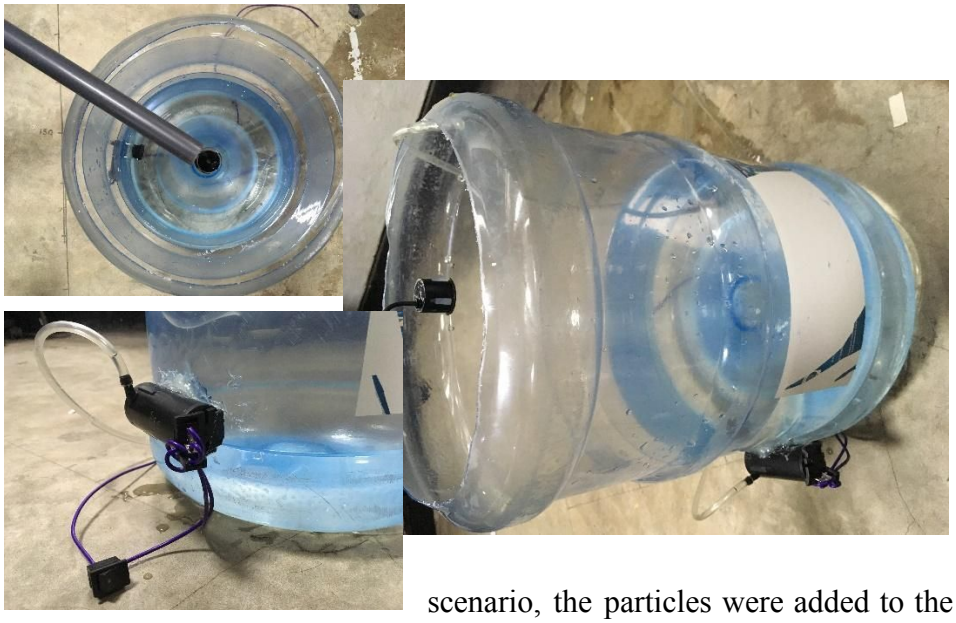
The actual distance values measured using the measurement tape is shown from the green color dashed line and the values obtain by the sensor are shown in yellow color solid line. The red color line shows the systematic error

distribution. The measurements are taken in the range of 0 to 300 cm. The experimented minimum and maximum distance values from the sensor are 19 cm and 295 cm respectively. According to the above statistics, there is a systematic error in the ultrasonic sensor readings. The error as a percentage of full scale is 0.97%. A systematic error is an error having a non-zero mean. That effect cannot be reduced by averaging the observation values. The reason for this error is the offset in the sensor reading for the minimum possible measurement value of 21 cm. Then a calibration was done by reducing that offset value from all the sensor readings.

5.2.2 Measuring the distance to a water surface

The apparatus is set as shown in the Figure 5.3. A cylindrical shaped bucket is used to fill water. The height of the water level in the bucket is changed and the sensor readings are recorded. At the same time the actual distance value is recorded using a measurement tape. The dimensions of the water bucket used in this experiment are as follows.

- Height of the bucket – 28 cm
- Diameter of the bucket – 25 cm



..

To make the environment more similar and appropriate to the real scenario, the particles were added to the water surface. Then the sensor reading values and actual distance values are plotted in a graph as shown in Figure 5.4. It also shows the error (value difference between actual value and sensor value) values.

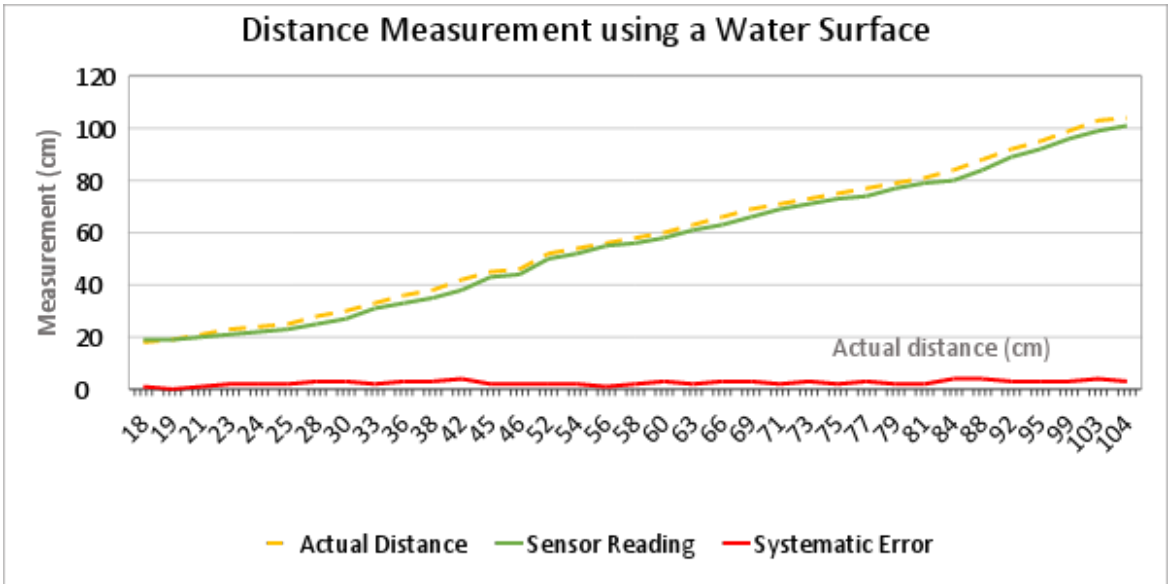


Figure 5.14 Distance measurement from a water surface.

The measurements are taken in the range of 10 to 100 cm as in the above section. Similar to the above experiment, the systematic error was there in this experiment as well. The error as a percentage of full scale is 0.87%. Therefore, the calibration was done by reducing that offset value from all the sensor readings.

5.3 Equations

5.3.1 Pixels to real world convention

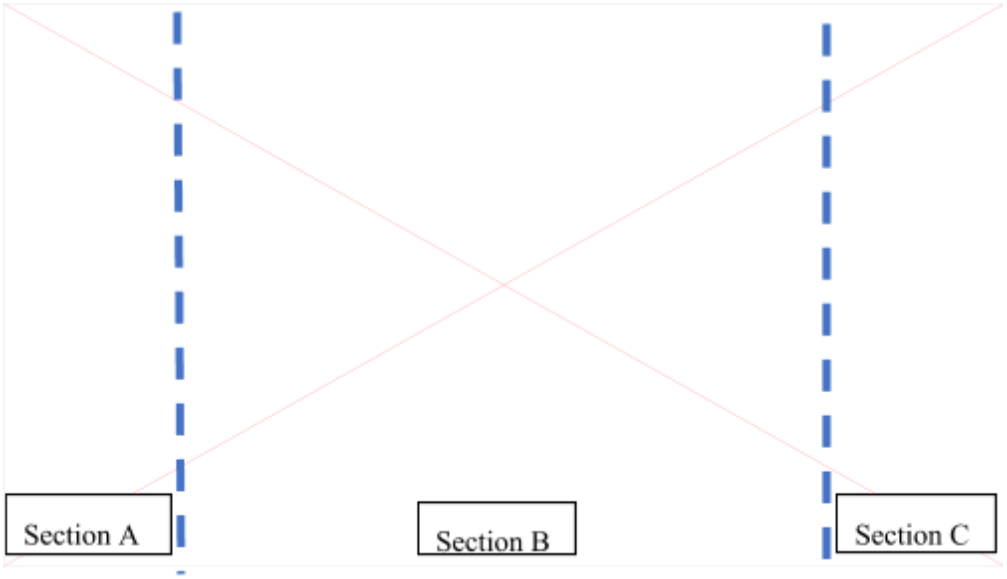
In PIV three algorithm we calculate the pixel distance moved from one frame to another. That pixel distance should be match with the standard distance measuring units. For that we use the following formula.

$$d = \frac{2h\ tan(\frac{\theta}{2})}{w} \times d_p \qquad (5.4)$$

Where, d is the distance, h is the height of the camera from the water flow, θ represents the camera angle, w is the width of the frame taken and d_p represents the pixel distance.

To validate the above mapping equation, we have taken an image of a square ruled page (See Figure 5.15) from the Raspberry Pi camera that we used in the device with the same

environment. In that image we measured the pixel distance in a small square width.



We measured the pixel distance in three sections of the frame. Blue color dotted lines in Figure 5.15 separates the frame in to approximately three sections as section A, section B and section C. Section B is the middle section where the Raspberry Pi camera is focused on. In that area, the distance is mapped to 0.56 cm.

$$d = \frac{2\times 0.16\times tan(\frac{54}{2})}{1366} \times 47 = 0.56\ cm$$

Measured distance of a small square width by a ruler says 0.6 cm. The error percentage is 6% in that section. Section A and C represents the corners of the frames. In those corner points, pixel distance is averaged to 44 pixels. That pixel measurement is mapped to 0.525cm in real distance. When the objects are moving away from the camera, error percentage has increased up to 12.5%.

Finally, absolute error of the velocity measurement can be calculated by considering all measurement errors, as in equation 5.5.

$$Velocity\ error(m/s) = [\frac{(\pm E_P)}{w} \times 2tan(\frac{\theta}{2}) \times (h \pm E_H) \pm E_C] \times \gamma$$

$$= \pm[\frac{(1.06)}{640} \times 2tan(\frac{54}{2}) \times (h + 0.01) + (4\times 10^{-4})] \times 50$$

$$= \pm [8.44\times10^{-2}h + 8.44\times10^{-4} + (2\times10^{-2})]$$

$$\approx \pm [0.0844\ h + 0.02]$$

(5.5)

Where, E_P = pixel distance error, E_H = height error, E_C = camera lens error, γ = frame rate. Moreover, when water level is 2 m, velocity error can be stated as $\pm 0.189\ ms^{-1}$ from the equation 5.5.

6 SUMMARY

We developed a device which could be used to measure the water flow rate and the water level of underground drainages. It is developed using a vision-based and ultrasonic techniques. The product is non-immersive and can be mounted inside the manholes of drainages. This device has two basic functionalities. They are measuring the water-flow level and calculating the flow velocity. Water level is calculated using a waterproof ultrasonic sensor. The flow velocity calculation is carried out using an extended version of an image processing technique called Particle Image Velocimetry. In velocity calculation, a Raspberry Pi camera is used to capture the image frames of the flow surface and then they are processed to calculate the flow velocity. However, this device is built in a way such that it can be used as a general platform, which can adopt to new algorithms and technologies.

This device could be mounted on underground drainages. Because the underground drainages are almost completely dark, to capture images from the camera it needs a lighting source. Therefore, A device called Head along with a light source will be mounted inside the manhole. To get power and have Internet connectivity to the Head device, another part of the system called Tail is mounted outside of the manhole. That will also facilitate debugging and configuration of the Head device which mounted inside the manhole. To obtain the best accuracy device should be mounted on a place parallel to the drainage flow, while having the top view the flow. Currently embedded algorithm is relying on floating particles of the drainage flow. This device will not be suitable for clear liquid flow with no particles. However, if there are average particle density in the flow, this device will ensure the accuracy of the velocity measurement of the flow. Because the device is IoT enabled, it supports MQTT gateways AWSIoT and Thingspeak.

Testing of the system should be done in a real-world drainage system to evaluate the performance in an actual setting. Because it was difficult to test our product in a real drainage, we test our product using an apparatus which is made by us for emulating the drainage environment. Some tests were conducted using the drainage flow videos that were recorded by us in several locations in University premises and Katubedda area.

Reliability of the velocity measurement depends on the calculation in the algorithm module and the accuracy of the distance sensor. Therefore, we evaluated the accuracy of the algorithm and the sensor readings. We evaluated the measurements of the system with different particle densities and different velocity levels. Then we evaluated the algorithm matching process. It showed that our validation process can provide high recall. Accuracy of the height measurements also tested using two experiments. First the distance to a solid object is measured and then the tests are conducted by measuring the height of an actual water surface. It concluded that height measurements error is insignificant the maximum absolute error of velocity measurement is 0.189 m/s, when water level is 2 m.

Finally, we can conclude that we have studied relevant literature and techniques for velocimetry. After that we implemented and evaluated those techniques to use in our concerned scenario. Because of that we could identify possible improvements and inherent weaknesses of each image techniques. Further, we introduced a methodology called Color Channels PIV, which has unique benefits in velocimetry domain. Finally, we implemented a hardware platform along with a device software to achieve the given requirements. Moreover, the platform is made common and customizable for using it as a framework for vision-based flow meters. However, we faced problems while achieving our goal. Some of them were overcame and some were not. Those considerations and suggesting improvements are explained in following two sections.

6.1 Problems and Challenges

6.1.1 Requirement of testing environment

Testing is essential during the whole implementation process of a product development. Time to time when changes are done very frequently, there should be a proper testing environment to conduct the needed tests. That environment should be easily reproducible. Without testing and having the feedbacks about an implemented section, it is very difficult to proceed with the implementation. However, we did not have a proper testing environment to conduct our tests. Most of the tests were done using the videos that were recorded by mobile phones and web camera. Also, it is very difficult to take the full system components outside to the conduct tests.

Not having a clear idea about the actual production environment is another big challenge faced by the team. Without having a good understanding about the real view of the drainage, size and shape, it is very difficult to decide on some of the features when implementing the system.

6.1.2 Requirement of calibrated measuring instruments

One of the main challenges we faced during the project is, we did not find any measuring instrument that can measure the flow velocity. A calibrated product in the market is needed to test the accuracy of our velocity calculation algorithm. We had to build our own testing environment and equipment to test the accuracy of the product.

In the first phase a rotating bed driven by two server motors were made. A texture patterns drawn polythene sheet was rotated using that apparatus and by placing the camera on top of that the velocity of the texture patterns were calculated. At the same time the time taken to rotate one round by those patterns were calculated using a stopwatch.

6.1.3 Image processing technique varies- not independent

To decide on the most specific algorithm, there should have a proper understanding about the production environment. The ability of using different image processing techniques vary with minimum and maximum speed of the flow and availability of tracking particles.

Also, the device contains a head motherboard that will be placed inside the manhole and a tail board which will be place outside the manhole. To decide on the proper communication protocol an idea about the distance between the head board and the tail board need to be known.

6.1.4 Height of the manhole

Having an idea about the height of the manhole is very important to decide on the appropriate ultrasonic sensor. Because different types of sensors support different distance ranges. When changing the sensor, it affects the complexity of the algorithm as well. The way of mounting the device also depends on the height of the manhole.

Height of the manhole affects the positioning of the camera. Also, the interior shape of the manhole affects the camera positioning. Because the camera has a specific camera angle. It should only capture the flow surface without including the walls of the manholes.

6.1.5 Computational power and High-power requirement

The computational power is a vital factor that should be considered in the development of this device. Because the core of our implementation is image processing, the implemented algorithms require high computational power. That is an inherent problem. Any of the image processing based solution require high computational power. When the computational power increases, the power requirement also increases.

Another important component of our device is the illumination source. Sufficient illumination should be provided to the flow surface so that the particles can be clearly captured by the camera. When the depth of the drainage increases the power (intensity) of the light source should be increased. When high power light source is used, it increases the overall power requirement of the system by a considerable amount. When the power consumption increases the running time of the batteries reduces and need frequent charging. Therefore, the device installing areas should be decided based on these factors. Areas where electricity is available will be more convenient as the charging of the batteries can be done easily.

6.2 Future Work

6.2.1 Facilitating long-range communication

Communication protocols and limitations are discussed in Chapter 3. Comparison of communication protocols in Table 3.1 clearly shows that RS-485 is the best protocol for the communication between Head device and Tail device. Because, it has long working distance and high immunity for noise in communication lines compared to RS-323. Further, RS-485 circuitry is economical to be implemented using MAX485 IC. Because modem necessarily need not to send messages to the Head board, RS-485 can be implemented in half-duplex mode. Then it only needs two lines and it further supports

twisted pair as the physical medium. It nicely fits for existing system design and only additional RS-485 to TTL converters are needed.

6.2.2 Rechargeable battery system

Implemented system should be powered by external 12VDC power supply and system does not include a power supply system. As a completed flow measuring system, a power supply should be included to match required power demand. This can be done by integrating a rechargeable battery, battery charger and charging mechanism. Charging mechanism may be an alternate current supply or set of solar panels. This design can badly affect to the economical nature of the implemented system. Because, batteries and solar panels are expensive items. Therefore, power supply should be designed by targeting specific installation site and after correctly identifying necessity of backup battery hours.

6.2.3 Making enclosure water resistance

Since head device is installed inside the drainage there is a possibility of contacting water. In flooding conditions water level become higher and may be rise to the level which device is mounted. If this happens device may contact water and circuit will be damaged. It is essential to design an enclosure which is water resistance to protect the circuit and other non-water resistance components of the device. When designing the water resistance enclosure, sockets and connectors also needs to be specially design for make them water resistance.

REFERENCES

[1] A.K. Jha and R.B.J. Lamond, Cities and Flooding, A Guide to Integrated Urban Flood Risk Management for the 21st Century, World Bank Publications, 2012.

[2] M. Selvabalan, N. Sharma, and G. Deshpande, "Surface Water Velocity Measurement Using Video Processing: A Survey," in Proc. *2014 Intel. Conf. on Electronics and Communication System (ICECS -2014),* 2014.

[3] N. Sharma, A. Naik, and M. S. Balan, "Non-intrusive Water Surface Velocity Measurement Using Spatial Cross Correlation Technique".

[4] S. Gharahjeh and I. Aydin, "Application of video imagery techniques for low cost measurement of water surface velocity in open channels," *Flow Measurement and Instrumentation,* vol. 51, pp. 79-94, 2016.

[5] M. Muste, W. Kim and J.M. Fulford, "Developments in hydrometric technology : new and emerging instruments for mapping river Title hydrodynamics," *WMO Bulletin,* vol. 57, pp. 163-169, 2008.

[6] OPENCHANNELFLOW, "Methods of Measuring Flows in Open Channel," OPENCHANNELFLOW, [Online]. Available: https://www.openchannelflow.com/blog/methods-of-measuring-flows-in-open-channels. [Accessed 09 11 2017].

[7] S. Farina, S. Alvisi, M. Franchini, and T. Moramarco, "Three methods for estimating the entropy parameter M based on a decreasing number of velocity measurements in a river cross-section," *Entropy,* vol. 16, no. 5, pp. 2512-2529, 2014.

[8] R.C. Kavanagh and A. Jones, "New Low-Cost Technique for Accurate Surface Velocity Measurement of a Rotating Drum," *IEEE Transactions on Instrumentation and Measurement,* vol. 53, no. 2, pp. 416-422, 2004.

[9] O. Engineering, "Acrylic Flowmeters," OMEGA Engineering, [Online]. Available: http://www.omega.com/pptst/FL2000.html. [Accessed 10 11 2017].

[10] O. Engineering, "Large Capacity In-line Flowmeters," OMEGA Engineering, [Online]. Available: http://www.omega.com/pptst/FL8100A_8300A.html. [Accessed 10 11 2017].

[11] O. Engineering, "Economical Liquid Turbine Flowmeters," OMEGA Engineering, [Online]. Available: http://www.omega.com/pptst/FTB1300_SERIES.html. [Accessed 10 11 2017].

[12] O. Engineering, "Economical Pilot Tube Flow Sensors," OMEGA Engineering, [Online]. Available: http://www.omega.com/pptst/FPT-3000.html. [Accessed 20 11 2017].

[13] CEP, "Coriolis: The direct approach to mass flow measurement," American Institute of Chemical Engineers, [Online]. Available: http://www2.emersonprocess.com/siteadmincenter/PM%20Micro%20Motion%20Documents/Direct-Approach-Mass-Flow-Measurement-AR-001694.pdf. [Accessed 19 11 2017].

[14] INSTRUMART, "Krohne OPTIMASS 7000 Coriolis Mass Flow Meter," Total Temperature Instrumentation, [Online]. Available: https://www.instrumart.com/products/33948/krohne-optimass-7000-coriolis-mass-flow-meter. [Accessed 19 11 2017].

[15] G. Rajita and N. Mandal, "Review on Transit time Ultrasonic Flowmeter," *in Proc. 2nd Intl. Conf. on Control, Instrumentation, Energy & Communication (CIEC), 2016,* pp. 88-92, 2016.

[16] O. Engineering, "In-Line Ultrasonic Flow Meter," OMEGA Engineering, [Online]. Available: http://www.omega.com/pptst/FDT500.html. [Accessed 19 1 2017].

[17] M. Saikia and S.G. Joshi, "Miniature , Low Cost , High Efficiency Transducers for Use in Ultrasonic Flow Meters," *Master's Theses,* no. 222, pp. 978-981, 2013.

[18] S. Joshi, "Efficient mode conversion transducers for use in ultrasonic flow meters," in *IEEE International Ultrasonic Symposium Proceedings,*, 2009.

[19] A. Akinin, J. Yang, A. Williams, A. Lee, P. Pourhoseini, A. Fronek, and G. Cauwenberghs, "Continuous wave ultrasonic doppler tonometry," *IEEE 2014 Biomedical Circuits and Systems Conference, BioCAS 2014 - Proceedings,* pp. 328-331, 2014.

[20] X. Dong, C. Tan, and F. Dong, "Water Continuous Oil-Water Flow Velocity Measurement Based on Continuous Waves Ultrasonic Doppler Method," in *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*,

2015.

[21] O. Engineering, "Ultrasonic doppler flow meter," Omega Engineering, [Online]. Available: http://www.omega.com/pptst/FD-400.html. [Accessed 12 June 2017].

[22] R. Velmurugan and P. Rajalakshmy, "Ultrasonic Flowmeter using Cross-Correlation Technique," *Intl. Journal of Computer Applications,* vol. 66, no. 10, pp. 19-22, 2013.

[23] J.A. Shercliff, The theory of electromagnetic flow-measurement, Cambridge: Cambridge University Press, 1962.

[24] A. Vire, B. Knaepen, and A. Thess, "Lorentz force velocimetry based on time-of-flight measurements," *Physics of Fluids,* vol. 22, no. 12, 2010.

[25] N. Dubovikova, C. Karcher and Y. Kolesnikov, "Velocity and flow rate measurement of liquid metal by contactless electromagnetic Lorentz force technique," *IOP Conference Series: Materials Science and Engineering,* vol. 143, 2016.

[26] C. TECH, "Toshiba Industrial Systems," Toshiba, [Online]. Available: http://www.clrwtr.com/Toshiba-Flowmeters.html. [Accessed 2017 November 06].

[27] J.E. Costa, K.R. Spicer, R.T. Cheng, F.P. Haeni, N.B. Melcher, E.M. Thurman, W.J. Plant, and W.C. Keller, "Measuring stream discharge by non-contact methods: A proof-of-concept experiment," United States Geological Survey, 2000.

[28] W. Matilde, J. Laronne, M. Salvaro, and G. Dramais, "Field assessment of noncontact stream gauging using portable surface velocity radars," *Water Resources Research,* vol. 52, no. 2, pp. 1108-1126, 2016.

[29] "Flo-Dar 4000 LR," Flowline, [Online]. Available: http://www.enviromontel.co.uk/flo-dar-4000-lr-flow-monitor-none-intrusive/. [Accessed 4 March 2017].

[30] B. Unsal, D. Trimis, and F. Durst, "Instantaneous mass flowrate measurements through fuel injection nozzles," *Intl. Journal of Engine Research,* vol. 7, no. 5, pp. 371-380, 2006.

[31] N. Morita, H. Nogami, E. Higurashi, and T. Ito, "Development of a Built-In Micro-Laser Doppler Velocimeter," *Journal of Microelectromechanical Systems,* vol. 25, no. 2, pp. 380-387, 2016.

[32] "ISCO MCERTS LaserFlow non contact Flowmeter," RS Hydro, [Online]. Available: http://www.rshydro.co.uk/flow-meters/open-channel-flowmeter/open-channel-flow-meters-stream-river-partially-filled/isco-laserflow-meter/. [Accessed 3 May 2017].

[33] M.D.G. Mory and F. Taugwalder, "Miniature optical sensor". United States Patent 6,654, 25 November 2013.

[34] Y.Y. Yang and S.C. Kang, "Crowd-based velocimetry for surface flows," *Advanced Engineering Informatics,* vol. 32, pp. 275-286, 2017.

[35] N. Sharma, A.A. Naik and M.S. Balan, "Non-intrusive Water Surface Velocity Measurement Using Spatial Cross Correlation Technique," *2014 International Conference on Information Technology,* vol. 3, no. 2, pp. 356-361, 2014.

[36] L.S. Nguyen, B. Schaeli, D. Sage, S. Kayal, D. Jeanbourquin, D.A. Barry, and L. Rossi, "Vision-based system for the control and measurement of wastewater flow rate in sewer systems," *Water Science and Technology,* vol. 60, no. 9, pp. 2281-2289, 2009.

[37] A. Hauet, A. Kruger, W. Krajewski, A. Bradley, M. Muste, J.D. Creutin, and M. Wilson, "Experimental system for real-time discharge estimation using an image-based method," p. 105–110, 2008.

[38] Z. Yu-xin, . X. Zheng-ping, G. Wen-qi and L. Yu, "A Real-time Two-dimensional Correlation Speed Measurement Based on Image," in Proc. *Intl. Conf. on Computer, Mechatronics, Control and Electronic Engineering (CMMCE)*, 2010, pp. 13-16.

[39] D. Jeanbourquin, D. Sage, L. Nguyen, B. Schaeli, S. Kayal, D.A. Barry, and L. Rossi, "Flow measurements in sewers based on image analysis: Automatic flow velocity algorithm," *Water Science and Technology,* vol. 64, no. 5, pp. 1108-1114, 2011.

[40] M. Muste, I. Fujita and A. Hauet, "Large-scale particle image velocimetry for measurements in riverine environments," *WATER RESOURCES RESEARCH, VOL. 44, W00D19, doi:10.1029/2008WR006950, 2008,* vol. 44, pp. 1-14, 2008.

[41] T. V. Berghe, "Image processing for a LSPIV application on a river," 2013.

[42] A. Masullo and R. Theunissen, "Adaptive vector validation in image velocimetry to minimise the influence of outlier clusters," *Experiments in Fluids,* vol. 57, no. 3, pp.

1-21, 2016.

[43] P.W.P. Wang, B.L.B. Li, G.Y.G. Yu, and D.S.D. Sun, "Analysis and processing of the river model test velocity data based on Digital Particle Image Velocimetry," in Proc. *3rd Intl. Congress Image and Signal Processing (CISP)*, 2010, vol. 4, pp. 1808-1812.

[44] Z. Huaien and H. Shengxiang, "Research on spatial data interpolation based on Kriging interpolation," *Engineering of Surveying and Mapping,* pp. 5-9, 2007.

[45] I. Fujita, H. Watanabe, and R. Tsubaki, "Development of a non-intrusive and efficient flow monitoring technique: The space-time image velocimetry (STIV)," *International Journal of River Basin Management,* vol. 5, no. 2, pp. 105-114, 2007.

[46] I. Fujita, "Discharge Measurements of Snowmelt Flood by Space-Time Image Velocimetry during the Night Using Far-Infrared Camera," *Water,* vol. 9, no. 4, p. 269, 2014.

[47] R.A.K.G.P. Lokhande, "Identification of parameters and restoration of motion blurred images," 2006.

[48] Dicklyon, "Point Spread Function," wikimedia, 06 01 2007. [Online]. Available: https://en.wikipedia.org/wiki/Point_spread_function. [Accessed 11 11 2017].

[49] A.K. Soe, "A simple PSF parameters estimation method for the de-blurring of linear motion blurred images using wiener filter in OpenCV," in Prpc. *Intl. Conf. on Systems and Informatics (ICSAI)*, 2012.

[50] O. Engineeering, "Rotameter for Liquid Flow," OMEGA Engineeering, [Online]. Available: http://www.omega.com/pptst/FL-10.html. [Accessed 10 11 2017].

[51] G.M. Hvasta, E. Kolemen and A. Fisher, "Application of IR imaging for free-surface velocity measurement in liquid-metal systems," *Review of Scientific Instruments,* vol. 88, no. 1, 2017.

[52] OpenCV, "Template Matching," open cv dev team, [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html. [Accessed 17 November 2017].

[53] OpenCV, "Feature Detection," opencv dev team, [Online]. Available: https://docs.opencv.org/3.0-beta/modules/imgproc/doc/feature_detection.html. [Accessed 11 November 2017].

[54] Alldatasheet.com, "Electronic Components Datasheet Search," Alldatasheet, [Online]. Available: http://www.alldatasheet.com/view.jsp?Searchword=IRF540. [Accessed 12 November 2017].

[55] MICROCHIP, "ATmega328P," Microchip Technology, [Online]. Available: http://www.microchip.com/wwwproducts/en/ATmega328P. [Accessed 11 November 2017].

[56] R.P. Foundation, "CAMERA MODULE," Raspberry Pi Foundation, [Online]. Available: https://www.raspberrypi.org/documentation/hardware/camera/. [Accessed 12 November 2017].

[57] N. Developers, "NumPy," NumPy developers, [Online]. Available: http://www.numpy.org/. [Accessed 11 November 2017].

[58] OpenCV, "Open Source Computer Vision Library," [Online]. Available: https://opencv.org/. [Accessed 11 November 2017].

[59] "Raspberry Pi camera sensor modes," [Online]. Available: http://picamera.readthedocs.io/en/release-1.13/fov.html#camera-modes. [Accessed 12 July 2017].

[60] A.W. Services, "Internet of Things," AWS, [Online]. Available: https://aws.amazon.com/iot/. [Accessed 17 11 2017].

[61] "Module range finder," Janakitshop, [Online]. Available: https://www.jahankitshop.com/market/d/8946. [Accessed 21 November 2017].

[62] "Altium Designer 16 Is Now Available," Altium, [Online]. Available: https://resources.altium.com/pcb-design-blog/altium-designer-16-is-now-available. [Accessed 11 November 2017].

[63] "AMPED Software," AMPED Software, [Online]. Available: https://ampedsoftware.com/five. [Accessed 10 November 2017].

[64] E. Edge, "Fluid Flow Velocity Profiles," ENGINEERS EDGE, [Online]. Available: https://www.engineersedge.com/fluid_flow/flow_velocity_profiles.htm. [Accessed 19 11 2017].

[65] "Point spread function," [Online]. Available: https://en.wikipedia.org/wiki/Point_spread_function. [Accessed October 2017].

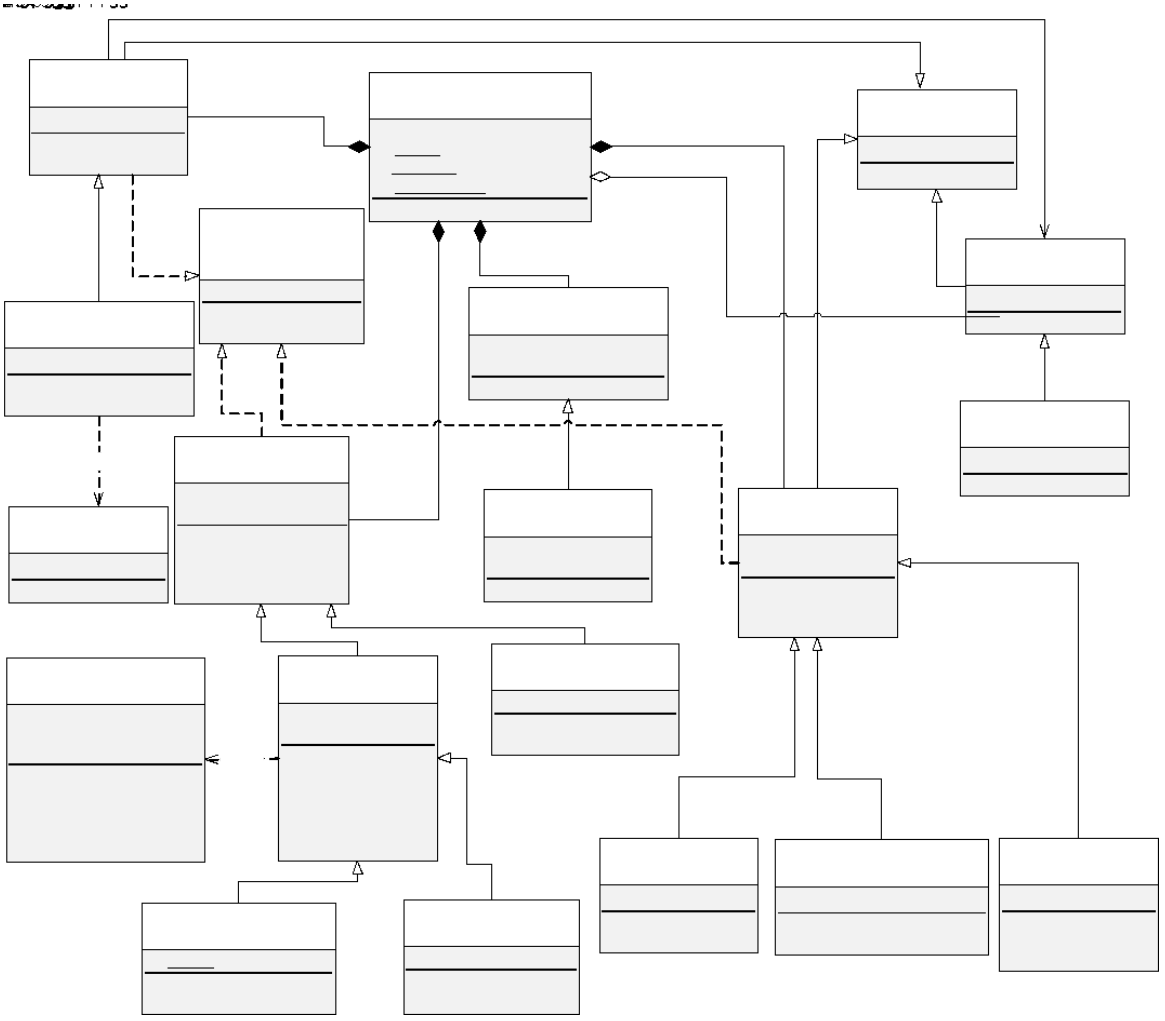
[66] OpenCV, "How to Use Background Subtraction Methods," 2016. [Online].

Available: https://docs.opencv.org/3.2.0/d1/dc5/tutorial_background_subtraction.html. [Accessed 2017 November 11].

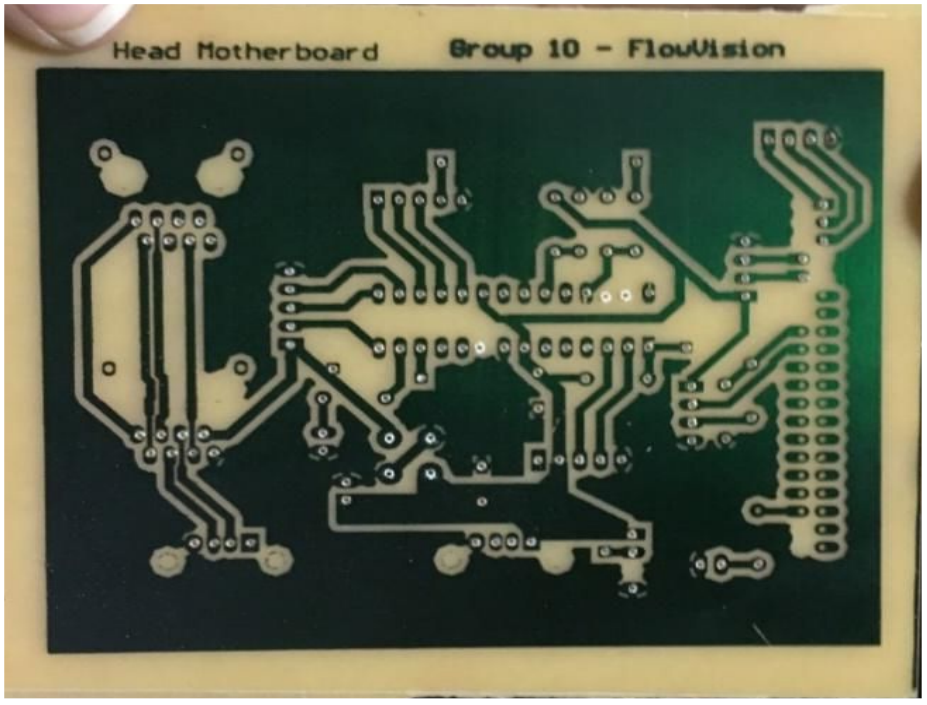
[67] OpenCV, "Background Subtraction," [Online]. Available: https://docs.opencv.org/3.3.0/db/d5c/tutorial_py_bg_subtraction.html. [Accessed 2017 November 11].

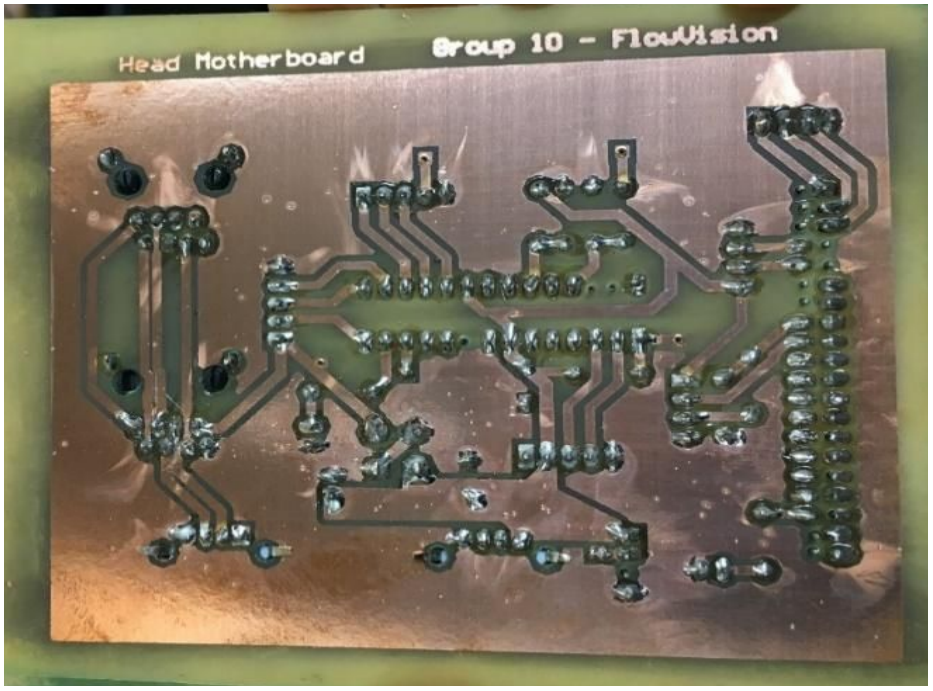
[68] R.P. FOUNDATION, "CAMERA MODULE," [Online]. Available: https://www.raspberrypi.org/documentation/hardware/camera/. [Accessed 19 11 2017].

Appendix I - Class Diagram with main software modules



Appendix II - Etching the head device circuit on the pre-sensitized board





Appendix III - Initial configuration & recent log views in debug web application