

ECE 445  
SENIOR DESIGN LABORATORY  
FINAL REPORT

---

# Particle Image Velocimetry

---

Team #8

HANFEI YAO (hanfeiy3@illinois.edu)  
YIHUI CHEN (yuhuic3@illinois.edu)  
YUEMING YUAN (yy28@illinois.edu)  
SIYUAN QIAN (siyuanq4@illinois.edu)

Sponsor: Timothy Lee

TA: Lijin Ma

May 7, 2024

## **Abstract**

Particle Image Velocimetry (PIV) is essential for understanding fluid dynamics, but existing equipment is complex and costly for lower-grade students. To address this, here comes a low-cost, portable device demonstrating PIV principles in a simplified manner. It comprises a Flowing Channel Subsystem for air circulation with tracer particles, an Illumination Subsystem for clear visualization, and an Image Acquisition Subsystem for precise velocity calculations. A user-friendly Interactive Graphical User Interface facilitates parameter adjustments and real-time velocity field visualization, while the Data Visualization Subsystem allows efficient fluid velocity evaluation. This solution meets educational needs for real-time fluid velocity measurements and visualization, fostering student interest in fluid dynamics.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem . . . . .	1
1.2	Solution . . . . .	1
1.3	Visual Aid . . . . .	2
1.4	High-level requirements list . . . . .	2
1.5	Block Diagram . . . . .	2
<b>2</b>	<b>Design</b>	<b>4</b>
2.1	Physical Design . . . . .	4
2.2	Mechanical System . . . . .	4
2.2.1	Aluminum profile bracket . . . . .	4
2.2.2	Wooden box . . . . .	5
2.3	Flowing System . . . . .	6
2.4	Illumination System . . . . .	7
2.4.1	Description . . . . .	7
2.4.2	Design Alternatives . . . . .	7
2.4.3	Subsystem Diagrams and Schematics . . . . .	8
2.5	Image Acquisition System . . . . .	8
2.5.1	Description . . . . .	8
2.5.2	Design Alternatives . . . . .	9
2.5.3	Subsystem Diagrams and Schematics . . . . .	9
2.6	Interactive Graphical User Interface System . . . . .	9
2.6.1	Description . . . . .	9
2.6.2	Design Alternatives . . . . .	11
2.6.3	Subsystem Diagram . . . . .	11
2.7	Image Processing and Data Visualization System . . . . .	12
2.7.1	Description . . . . .	12
2.7.2	Design Alternative . . . . .	13
2.8	Tolerance Analysis . . . . .	13
<b>3</b>	<b>Verification</b>	<b>16</b>
3.1	Mechanical System . . . . .	16
3.2	Flowing Channel Analysis . . . . .	16
3.3	Interactive GUI System . . . . .	17
3.4	Image Processing and Visualization System . . . . .	18
<b>4</b>	<b>Cost and Schedule</b>	<b>20</b>
4.1	Cost . . . . .	20
4.1.1	Labor Cost . . . . .	20
4.1.2	Parts Cost . . . . .	20
4.1.3	Total Cost . . . . .	21
4.2	Schedule . . . . .	21

<b>5</b>	<b>Conclusion</b>	<b>23</b>
5.1	Accomplishments . . . . .	23
5.2	Uncertainties . . . . .	23
5.3	Future Work . . . . .	24
5.4	Ethical Considerations . . . . .	24
5.4.1	Ethics . . . . .	24
5.4.2	Safety . . . . .	25
	<b>References</b>	<b>26</b>
	<b>Appendix A Requirements and Verification</b>	<b>27</b>

# 1 Introduction

## 1.1 Problem

Understanding how fluids move is crucial for many scientific and engineering applications. Nowadays, Particle Image Velocimetry (PIV) has evolved to be the dominant method for velocimetry in experimental fluid mechanics and has contributed to many advances in our understanding of turbulent and complex flows. However, we found out that the PIV equipment for experimental use was not straightforward enough for lower-grade students to understand the basic principle, also it had some restrictions on the environment and a relatively high cost. As a result, we plan to design a low-cost, easy-maintaining, and portable device that can demonstrate how PIV works in a simple way without many restrictions on the environment. Also there are some interactive functions in the device so it can give a deep impression on students and raise their interests in fluid dynamic study.

## 1.2 Solution

This is a device that can detect the state of particle motion in a flowing channel and present it intuitively. It includes some distinct subsystems to realize relatively accurate and real-time measurements. Within the design, the Flowing Channel Subsystem ensures continuous air circulation through a sophisticated channel system driven by a blower and uses tiny particles to be the tracer objects. The Illumination Subsystem incorporates a laser source and optical components, such as lenses and mirrors, to illuminate a certain area in the flowing channel, in order to gain clear images of particles. The Image Acquisition Subsystem, equipped with a digital camera, captures and aligns particle motion images and transmits them to an Image Processing System for precise velocity calculations, data analysis as well as data visualization. The Interactive Graphical User Interface Subsystem offers a user-friendly platform with a control panel for parameter adjustments, a display for image acquisition preview, and real-time velocity field visualizations. The Data Visualization Subsystem enables efficient fluid velocity evaluation and automated plot generation. This solution caters to educational scenarios that require real-time and reasonably accurate fluid velocity measurements and visualization, with flexible requirements for specific components.

### 1.3 Visual Aid

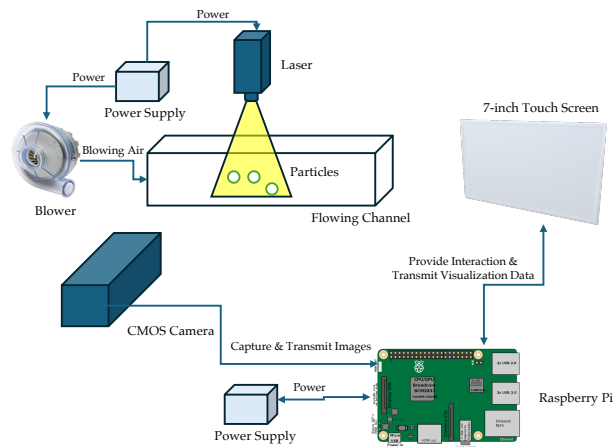


Figure 1: Visual Aid

### 1.4 High-level requirements list

- The flowing channel should hold different small particles (e.g., tiny Styrofoam balls, Quartz sands or water droplets) to flow through and show the movement of them clearly.
- In the motion images, the length of the residual shadow produced by particles should be no more than four times the diameter of them to ensure that the motion of the particles can be well captured.
- The system should offer an interactive experience with a user-friendly interface that includes real-time visualizations, e.g., particle images, particle velocities and calculated flow rate of fluids.
- The instructor can control the device manually and demonstrate how the PIV device works directly or choose to control the device like how the users do. Under the instruction and supervision, students should be able to control the PIV demonstration, e.g., toggling on or off the measurement and changing the flow rate of fluids, with a real-time graphical user interface (GUI).

### 1.5 Block Diagram

The block diagram of the device is shown in Fig. 2. The device includes a laser source and a CMOS camera to perform the estimation of the fluid velocity. A user-friendly instantaneous visual feedback on particle images, velocities, and calculated flow rates is provided on a 7-inch touchscreen driven by a Raspberry Pi. The device offers the flexibility to fine-tune a variety of parameters that directly impact the measurement of fluid flow rates, which helps illustrate the principles of fluid dynamics in a hands-on experience.

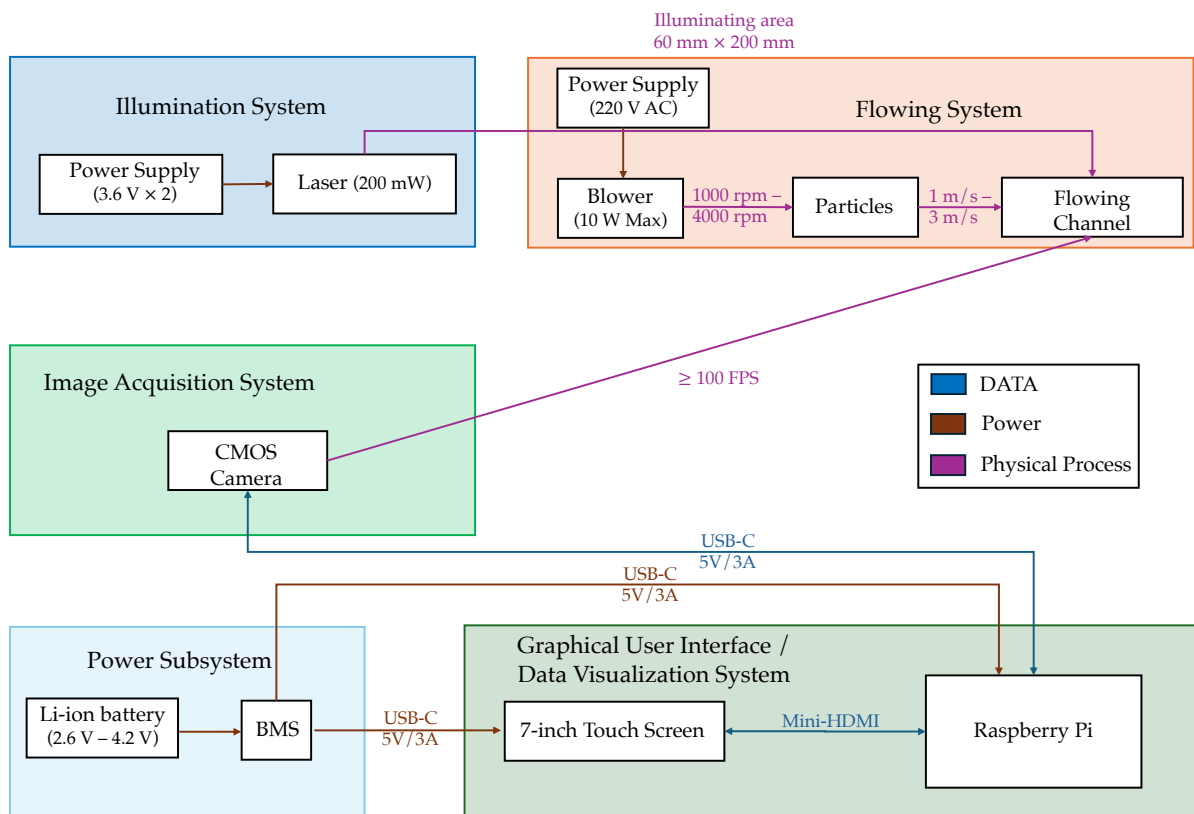


Figure 2: Block Diagram

## 2 Design

### 2.1 Physical Design

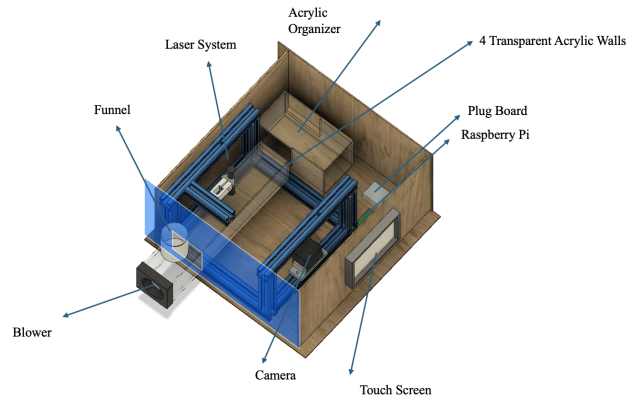


Figure 3: Design Overview

The Design Overview is shown in Figure 3. Our design consists of the following parts: the transparent channel in the figure is our flowing system where its body is a closed channel connected to a blower with a maximum power of 20 W at one end, and at the other end is a small acrylic organizer for easier particle recycling, and the body of the channel is covered by four transparent acrylic panels in the front, at the back, at the bottom, and on the top. At the outside of the channel is a mechanical support structure made of aluminum profiles, which are connected using a corner connector. The high-speed camera, with an accuracy of 480 fps, is attached to the aluminum profiles using a support piece. The laser pointer on the top can present a laser sheet, which is also attached to the aluminum profile. On the outside of the aluminum profile there is a box made of wood to ensure an overall dark environment, one of the sides has a cut in the lower left corner to allow for a portion of the tubing to protrude out of the wooden box, thus facilitating particle addition and the operation of the blower, with particles being added through a larger funnel. Another side also has a cut in it to facilitate the attachment of the display to the wooden board using bolts and nuts. The remaining two sides are complete boards with no cuts. The bottom is also a full board. To make it easier to adjust the laser pointer and camera position, the top is not secured to the board but is covered with a wooden board to ensure that the channel is overall in a dark environment. The Raspberry Pi is attached to the display by wires and placed on the bottom board. Inside the wooden box, there is an integrated plug board for connecting all devices that need to be powered, which is connected to the external power supply via a switch.

### 2.2 Mechanical System

#### 2.2.1 Aluminum profile bracket

The frame used to connect the camera to the laser system is made of aluminum profiles, a material with many benefits: strong, inexpensive, and easy to install, a material well



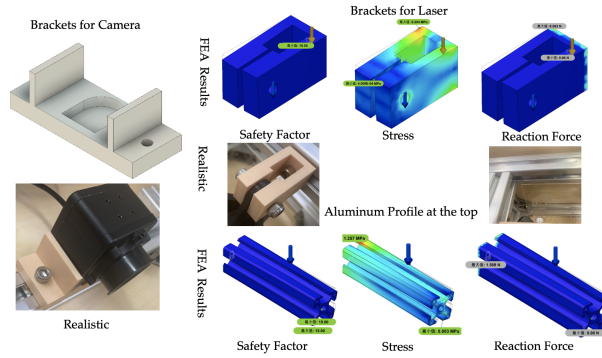


Figure 4: Brackets and FEA Analysis

suited for support structures. The overall frame is made entirely of 3030 aluminum profiles, which are connected to each other by means of T-nuts and corner connectors. The bottom part consists of two 30cm and two 24cm profiles. The camera is attached to one of the 24cm profiles by a bracket, which can be moved to facilitate the focusing of the camera. The length of the four vertical profiles is 15cm, a height that makes it easy to adjust the position of the laser pointer and also easier to place the particles. The top profile is 12mm long, and within this limited length, the laser pointer can be adjusted to the center of the channel. There are also two 3D printed brackets on the whole structure, one connecting the camera to the aluminum profile and the other connecting the laser pointer to the aluminum profile, where the latter is one of the few parts of the whole mechanical system that has a significant force, so it needs to be subjected to a simple FEA analysis. The laser pointer weighs about 180 grams or 1.764 N. We prepare a 10-times tolerance, or 17.64 N, to be added at the laser pointer's location, and the final factor of safety is much greater than 15, indicating that such a design is safe. Figure 4 presents the design of the brackets and a simple FEA analysis.

## 2.2.2 Wooden box



Figure 5: Outer and inner parts of the wooden box

This wooden box serves as a crucial component for creating a controlled environment for the main channel, enabling optimal particle movement visible through the camera. Its design, depicted in the accompanying image, features an opening on one side, positioned at the lower left corner, facilitating easy access for adding particles and operating the

blower. Another opening on the opposite side allows for securing the display to the planks using bolts and nuts.

Two sides and the bottom consist of complete boards with no cuts, ensuring structural integrity. The top is covered with a large wooden board, ensuring the channel remains enclosed in darkness. This setup enables easy adjustment of the laser pointer and camera.

Inside, a Raspberry Pi is wire-connected to the display and placed on the bottom plank. An integrated plug board facilitates connection to all powered devices, linked to an external power supply via a switch for convenient operation.

### 2.3 Flowing System

Transparent container that allows fluid to flow through. The container will be a hollow acrylic channel. The length of the channel is 40 cm, and the speed of the blower is set to about 2 m/s, which means that the time for the particles to pass through the whole channel is about 0.2 s and a scientific camera with a frame rate of 480 frames per second is used, which means that the spacing of the same particles on the two neighboring frames is about 0.4 cm. The reason for setting the cross-section as a rectangle is to make it easier for the whole channel to be assembled, since two of the sides of the channel are wrapped in nylon mesh, a circular cross-section is less suitable.

There is a manually operated 20 W Max blower that ensures that the air and particles can circulate through the flowing channel and the flow rate of the air can be measured and controlled. The particles (which will be discussed in the following paragraphs) can be put in the flowing channel before we turn on the blower. The Flowing System must ensure that a relatively closed environment that other air disturbances in the environment will not affect the flow. In the system, the particles can flow at a speed of 1 m/s - 3 m/s.

For the particle selection, Jingsong [1] has provided some standards: particles should have a close-sphere shape in order to give a homogeneous image at any orientation. Particles should also be distributed evenly in the field so that the same accuracy can be obtained throughout the whole image area. This also guarantees that the two correlated images will not see a dramatic change in particle density. Such a sudden change in image density can introduce a large number of unmatched image pairs, and hence, introduce measurement "noise" to the result. So the shape should be as round as possible and the size distribution should be as uniform as possible and there should be a sufficiently high light scattering efficiency. Also, the tracer particles should have a volume fraction of less than  $10^6$  [2] to eliminate the effect it would bring to the flow.

So some of the choices of the materials are listed as follows (several tests will be done to find which one is the most suitable), which are Styrofoam balls, quartz sands, and water droplets, the following table shows the relevant parameters and advantages of the three particles.

Particles	Size (Diameter)	Advantages	Disadvantages
-----------	-----------------	------------	---------------

Styrofoam balls	2-4 mm	High strength, light weight and durable	The size is too big for a good demonstration
Quartz sands	0.3-0.8 mm	The size fits better and it's cheaper	Heavy weight, not easy to blow
Water droplets	0.1-0.2 mm	Cheap, have a better particle generation with a sprayer	Too small to catch for the camera

Table 1: Particle Choice

We will use such an analog device to conduct experiments for the following three tests.

After experimentation, different particles require different kinds of blowers. Styrofoam balls are light enough that a slow blower will blow them up, while a fast blower will cause them to go too fast and make them difficult to capture on camera. Quartz sand is denser and requires a fast blower to blow it up. So to easily test several different particles, it is necessary to design different channels for two different-speed blowers, so that when replacing the test particles, it is only necessary to replace the entire channel together. Figure 6 shows the two different channels.



Figure 6: Channels for different blowers

## 2.4 Illumination System

### 2.4.1 Description

The illumination system consists of a laser source and optical lens combinations, it should illuminate the smaller particles (Quartz sands and water droplets) when necessary. A laser source that provides a thin laser sheet to illuminate the flowing area, it should be position-adjustable and can synchronize with the image acquisition system. The system contains an optical system which could include lenses, mirrors, and filters to create a well-lit area for capture high-quality images which are used to do the further process. The laser requires a power of 200 mW and two 3.6 V rechargeable batteries are used to provide power.

### 2.4.2 Design Alternatives

For now, the illumination system can illuminate the particles well and make the image clear enough for further image processing. But the laser is a relatively separate part and it

has to be turned on and off manually before the demonstration starts, which can expose people to lasers for long periods which could cause damage, so a better solution is adding a power control unit and connecting it the Raspberry Pi, so the demonstrator can turn on and turn off the laser by the touch screen. The newest action taken is that the laser pointer is changed to a laser module which is connected to the Raspberry Pi's power supply, so the laser and the Raspberry PI can be turned on and off simultaneously, but for long-term convenience, a power control unit is necessary.

### 2.4.3 Subsystem Diagrams and Schematics

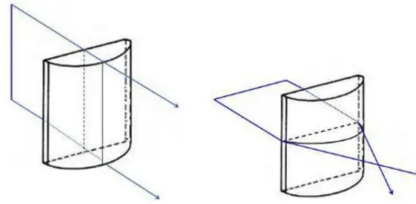


Figure 7: 3D Ray diagrams of the semi-cylindrical lens



Figure 8: The Laser



Figure 9: The semi-cylindrical lens

## 2.5 Image Acquisition System

### 2.5.1 Description

The image acquisition system is used to capture the image for the flowing particles and send the images to Raspberry Pi for further information processing. A camera is used to capture images of the particles flowing in the fluid, the captured images are going to be sent to the Raspberry Pi via a USB cable. The camera should have high frame rates, high sensitivity, short and accurate inter-frame time, and sometimes high resolution. A global shutter camera is preferred. The frame rate of the Camera should be larger than 100 fps. The camera needs 5 V / 3 A power input via USB-C cable and transmit information via USB 2.0 cable.

## 2.5.2 Design Alternatives

For now, the image acquisition system can capture high-quality images that provide relatively satisfying visualization and measurement results, but when doing experiments, there is a strong reflection at the bottom of the flowing channel. A physical way to deal with this problem is to add a filter to the camera to mitigate reflections. The latest action we take is using algorithms to reduce the effect of the reflection. Another small issue that is worth mentioning is the focus problem, our camera is a fixed-focus camera, so every time the position of the camera is changed, the focus may need to change manually, an auto-focus camera can be used for convenience.

## 2.5.3 Subsystem Diagrams and Schematics

Resolution	MJPEG frame rate
640 x 360	270 fps
640 x 480	480 fps
800 x 600	190 fps
1024 x 768	190 fps
1280 x 720	270 fps
1280 x 1024	190 fps

Table 2: 1/2

Sensor size	1/3"
Pixel size	2.2 $\mu\text{m}$ x 2.2 $\mu\text{m}$
Module Interface	USB 2.0 High Speed (4 pin - 2.0 mm)
Working voltage	5 V
Focus mode	Adjustable, manual focus
Shutter Type	Global shutter

Table 3: Main parameters of the camera

## 2.6 Interactive Graphical User Interface System

### 2.6.1 Description

The Interactive Graphical User Interface (GUI) System features a 7-inch touchscreen that displays customized GUI software for an enhanced user experience, as shown in Fig-

ure 10. The GUI software processes user commands, interacts with other subsystems, and offers visual feedback to the user. The GUI software supports real-time camera control and real-time display of data visualization, as shown in Figure 11, which is essential for enabling precise fine-tuning of the CMOS camera and communicating information to the user.

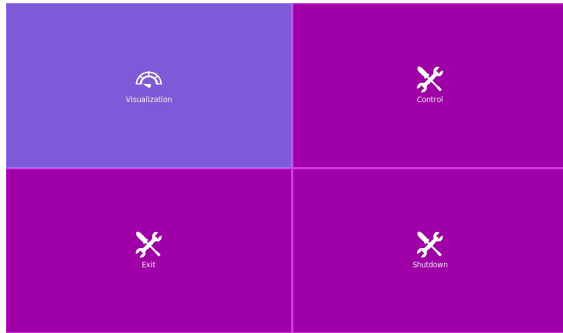


Figure 10: Top-level interface of the GUI software.

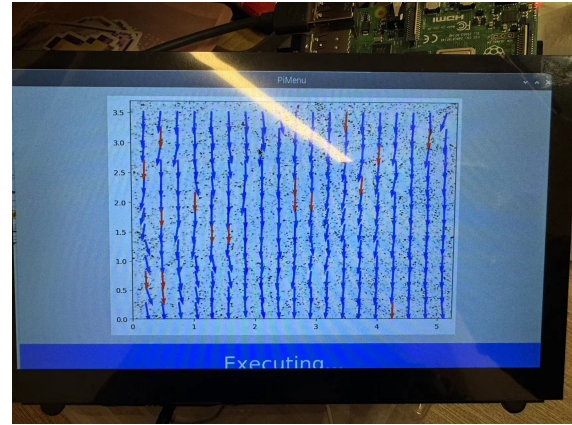


Figure 11: Example integration of data visualization into the GUI software.

Back	
MJPEG	YUYV
640x480 @ 480 fps	640x480 @ 30 fps
800x600 @ 190 fps	800x600 @ 15 fps
1024x768 @ 190 fps	1024x768 @ 10 fps
1280x960 @ 190 fps	1280x960 @ 5 fps
640x360 @ 270 fps	640x360 @ 30 fps
1280x720 @ 270 fps (default)	1280x720 @ 10 fps
1280x1024 @ 190 fps	1280x1024 @ 5 fps

Figure 12: Camera mode selection functionality of the GUI software.



Figure 13: GUI interface for fine-tuning camera parameters.

The interactive GUI Subsystem is connected to the Data Visualization Subsystem and the Image Acquisition Subsystem. As shown in Algorithm 1, users can toggle and adjust the data visualization software, which is a part of the Data Visualization Subsystem, and manipulate the settings of the CMOS camera, which is a part of the Image Acquisition Subsystem. Adjustments in camera parameters [3], such as contrast, brightness, hue, and saturation, are received with button widgets and passed to the Image Acquisition Subsystem as OpenCV API calls [4], which influences the accuracy of the fluid velocity interpolated by the Data Visualization Subsystem.

Power supply for hardware components in this subsystem comes through a 5V/3A USB-C connection from the Power Subsystem. The power supply ensures that the screen and

the Raspberry Pi can maintain responsiveness of the GUI software. The Raspberry Pi sends graphical data to the touchscreen via a mini-HDMI connection, which supports the display of high-resolution (1024 × 600) graphics and ensures that the visual output is clear and vibrant, and sends control signals via a micro USB cable.

### 2.6.2 Design Alternatives

It is potentially useful to be able to fine-tune the frame rate (FPS) of the CMOS camera. However, in the current system, the frame rate of the CMOS camera used cannot be fine-tuned as it is fixed for each specific mode (YUYV, MJPG) and resolution pair available. This limitation is due to the hardware specifications of the CMOS camera and the its driver provided by the manufacturer, which preset the maximum achievable frame rate for each mode-resolution combination. Manually setting the frame rate with OpenCV function calls (`VideoCapture.set`) causes the frame rate drops to zero. Consequently, users must select from the predefined mode-resolution combinations available, which restricts the flexibility of camera control over frame rates. Current GUI implementation supports camera mode selection, which enables a coarse control over the frame rate. Such mode selection functionality appears to be sufficient for enabling expected computational effects.

During the implementation phase of the GUI software, computational software from the Data Visualization Subsystem blocked the GUI updates, leading to unresponsive GUI behavior during data visualization. One potential solution would include calling data visualization routines with separate threads. However, this strategy does not work since the computational software uses Matplotlib [5], which has thread-safety issues when called from separate threads. This issue can be resolved by calling data visualization routines in a separate process.

### 2.6.3 Subsystem Diagram

A diagram of the Interactive GUI Subsystem and its interaction with other subsystems is shown in Figure 14.

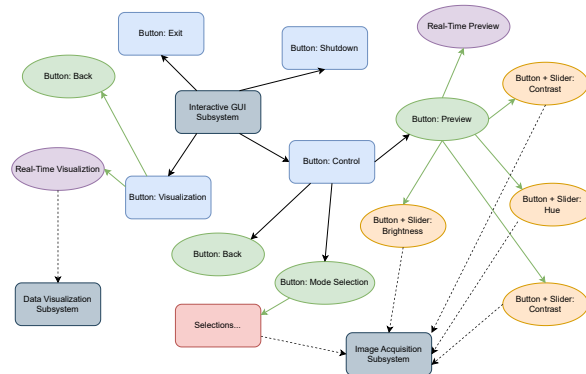


Figure 14: A diagram of the Interactive GUI Subsystem.

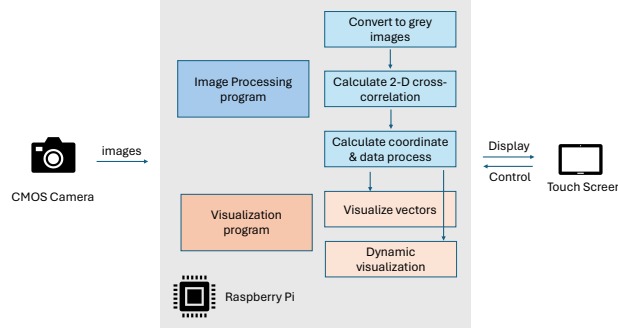


Figure 15: The block diagram of the image processing and visualization subsystem.

## 2.7 Image Processing and Data Visualization System

### 2.7.1 Description

The system first processes the images and calculates basic information, including the velocity magnitude and the angular deviation. The captured images of the particles are transferred to the Raspberry Pi for further processing and visualization. Here, we use Python to do the image processing and use the library `OpenPIV` to calculate PIV vectors.

The system block diagram is shown in Fig. 15. The software system consists of the following steps:

1. Process the image. Retrieve the photos from CMOS Camera, select the target area in the original photos and convert the images into greyscale. Then compress the pixels in the images by 10x.
2. Calculate a 2-D cross-correlation of the placement of the particles (white pixels) on the processed images.
3. Process the data. Remove the outlier ones (signal-to-noise ratio  $\geq 1.5$ ) and replace by an estimated one. Calculate the visualization parameters, including output coordinates.
4. Complete the visualization. Draw the vector image using `Matplotlib`.

We will also provide multiple visualizations of the calculated PIV. The basic figure, static visualization, will show the velocity vectors in different parts of the fluid. For better illustration, we will use multiple colors to display the vectors more intuitively. We show the sample images and visualization in the verification section. We will also provide an dynamic visualization. Combining the data measured in successive images, the system will generate a continuous changing visualization to show the dynamic variations. The FPS of the visualization mainly depends on the performance restriction of the Raspberry Pi.

These visualizations will also be done in Python on Raspberry Pi, and will be shown on the touch screen to interact with.



## 2.7.2 Design Alternative

The main issue we faced in this subsystem is how to improve the accuracy of the PIV calculation.

In the first test on our real images, the PIV vectors is near random. The reason is the random noise in our images is too large for an accurate correlation computation. Besides, it require more than 5 seconds to process two images and generate one visualization result, which is too slow for the flowing process.

To solve these problems, we first compress the image to smaller pixels. Since the PIV is calculated on pixels, it will reduce the influence of random noise. Also, it reduce the computation time to 0.5 second on the Raspberry Pi.

In order to have a better result, we remove the PIV vectors if the signal-to-noise-ratio is larger than 1.5, and use the other vectors to estimate them.

The main code flow is as below:

```
# Some code here... Load camera output figures and set image format
# Some code here... Set configurations

compressed_fig1_path = compress.compress_image(fig1_path)
compressed_fig2_path = compress.compress_image(fig2_path)

u0, v0, sig2noise = pyprocess.extended_search_area_piv(frame_a, frame_b,
                                                    window_size=winsize, overlap=overlap,
                                                    dt=fps, search_area_size=searchsize,
                                                    sig2noise_method='peak2peak')

flags = validation.sig2noise_val(sig2noise, threshold = 1.5)
u2, v2 = filters.replace_outliers(u0, v0, flags, method='localmean', max_iter=
                                3, kernel_size=3)

x, y = pyprocess.get_coordinates(...)
x, y, u3, v3 = scaling.uniform(...)
x, y, u3, v3 = tools.transform_coordinates(x, y, u3, v3)

tools.display_vector_field(...)
```

## 2.8 Tolerance Analysis

The velocity is calculated by the following formula:

$$V = \frac{\Delta X}{\Delta t}$$
$$\Delta X = \frac{D_p A}{R}$$

where  $\Delta t$  is the time interval between successive images.  $D_p$  is the average displacement

of particles between successive images, measured in pixels;  $A$  is the size of the area being imaged;  $R$  is the resolution of the digital camera, measured in pixels.

The main error may come from the following sources: 1. Error in  $\Delta t$ . We require the digital camera to take successive photos in a short time. If it happens to have a delay in taking photos, the actual  $\Delta t$  will be different from the expected one. 2. Error in  $\Delta X$ . This error includes errors in measuring the displacement between the particles ( $D_p$ ) and the error in measuring the size of the imaged area ( $A$ ).

To achieve our expected precision of the measurements (within 20% error), we need to ensure:

1. The inter-frame time accuracy of the camera. Assume the displacement measurement is accurate and we use a camera of 60 fps, namely  $\Delta t = \frac{1}{60} \approx 0.01667$  s. The tolerated inter-frame time interval is between  $\frac{1}{1 \pm 0.2} \Delta t \approx 0.01389$  s – 0.02083 s.
2. The accuracy of the displacement measurement. Assume we use a camera of 60 fps and its inter-frame time is exactly 0.01667 s. We consider two cases here, the minimal estimated velocity 10 cm/s and the maximum estimated velocity 30 cm/s. When the velocity is 10 cm/s, we need to ensure  $\Delta X < 0.2 * V * \Delta t = 0.0333$  cm in two successive photos. Similarly, when the velocity is 30 cm/s, we need to ensure  $\Delta X < 0.1$  cm.

---

**Algorithm 1** Interactive GUI

---

```
1: procedure INITIALIZECAMERA
2:   Set camera resolution and frame rate based on selected mode
3:   If camera initialization fails, print error
4: end procedure
5: procedure STARTPREVIEW
6:   Initialize camera settings and GUI components for preview
7:   while true do
8:     Capture frame from camera
9:     Update GUI with the captured frame
10:    Save frame if debugging is enabled
11:    Delay briefly
12:   end while
13: end procedure
14: procedure STARTVISUALIZATION
15:   Setup GUI components for visualization
16:   Initialize camera
17:   Capture and process images continuously
18: end procedure
19: procedure CAPTUREANDPROCESSIMAGES
20:   while true do
21:     Capture two consecutive images
22:     Process images with data visualization routine piv_plot
23:     Display results on GUI
24:     Delay briefly
25:   end while
26: end procedure
27: procedure ADJUSTCAMERASETTINGS
28:   Adjust camera setting based on inputs received from button widgets
29:   Update GUI progress bar to reflect changes
30: end procedure
31: procedure RESTOREDEFAULTS
32:   for each setting in default settings do
33:     Restore setting to its default value
34:     Update GUI progress bar
35:   end for
36: end procedure
37: procedure MAIN
38:   Setup main GUI window
39:   Create instance of PiMenu
40:   Run main event loop
41: end procedure
```

---

## 3 Verification

### 3.1 Mechanical System

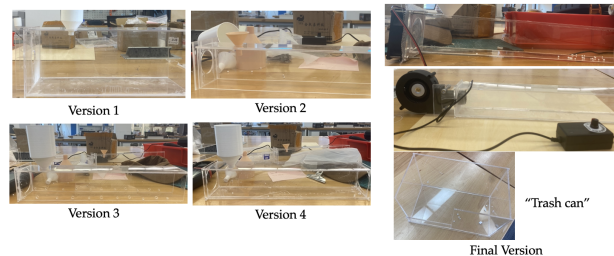


Figure 16: Iteration of the channel design

As shown above in Figure 16, our channel iterated through five major design versions throughout the project. Each deprecated version had a specific reason for failure. This section will briefly discuss the shortcomings of each discarded version. Version 1 was the largest of the pipelines, and this design resulted in a large amount of useless bulk that would cause the entire system to be too large, so the primary issue was to reduce the size of the channel to make it as small as possible. Version 2 is simply a smaller version of version 1, one of the main problems with this version is that it does not recover the particles well, once blown out the particles will fly to various places in the experimental environment, so version 3 utilizes several threaded holes in the wall of the tube and a screen is mounted through these holes and surrounds the entire tube. The mesh is also available in different sizes, with a 0.5mm diameter nylon mesh being used for larger particles such as the Styrofoam ball, and a 0.05mm diameter nylon mesh being used for smaller particles such as quartz sand. The problem with version 3 is that the bottom of the screen is not necessary, because after testing found that the blower is strong enough under the circumstances that the particles will not continue to stay on the bottom of the plate, so in the bottom of the bolts and nuts to install the screen seems redundant, so the version 4 directly to the bottom of the piece of the acrylic plate. The problem with version 4 is that although the particles can be recycled, it is not convenient to do so, and the filter needs to be disassembled every time it is used, which is troublesome, so in version 5, which is also the final version, we used to open up one end of the channel, and put a "trash can" made of acrylic plate at the other end, so that the particles blown out of the channel can be recycled. This way, the particles blown out from the channel can be organized by the "trash can", which makes it easier to clean.

### 3.2 Flowing Channel Analysis

Several simple CFD analysis are tried to learn more about the airflow in the channel in order to design a flowing channel with a relatively stable and consistent flow for better measurement. CFD analysis for all five version of the flowing channel are shown below in Figure 17 to 21:

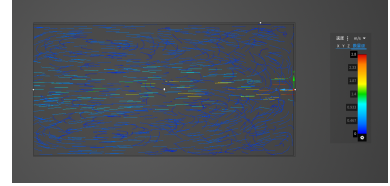
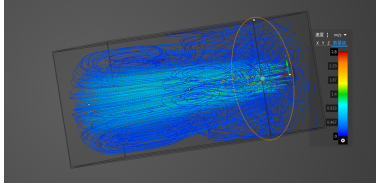


Figure 17: CFD results for version 1

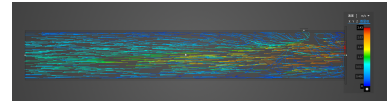
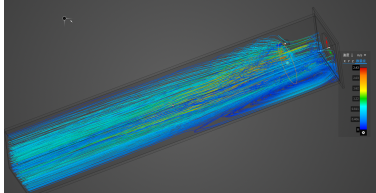


Figure 18: CFD results for version 2

For every pair of the CFD results, the first image shows the streamline around the cross-section of the inlet of the small particles, and the second image shows the overall velocity vector field.

To design a stable and consistent airflow channel, the streamline and the vector field should be settled and stable. So from the CFD results, we can see that with the iteration, the streamline and the vector field are more settled. For the latest version of our channel, it seems that the performance under the higher speed is not satisfying enough, we may upgrade it to fit the high-speed flow better.

### 3.3 Interactive GUI System

Test Item	Details
<b>GUI Response Time Verification</b> (Modular Testing)	<ul style="list-style-type: none"> <li>• Procedure: Measure input response time using a software timer (<code>time.time()</code>) from the moment of touch to the action being executed on the GUI.</li> <li>• Tolerance: The GUI is updated within 50 ms.</li> <li>• Result: The GUI finishes updates in 10 ms.</li> </ul>
<b>GUI Command Execution Error Rate</b> (Modular Testing)	<ul style="list-style-type: none"> <li>• Procedure: Randomly issue different kinds of commands 100 times. Record number of errors.</li> <li>• Tolerance: Error rate must be below 0.1%.</li> <li>• Result: No GUI error noticed in the latest version.</li> </ul>

Test Item	Details
<p align="center"><b>GUI Navigation Efficiency</b> (Modular Testing)</p>	<ul style="list-style-type: none"> <li>• Procedure: Conduct user navigation tests to ensure that any primary function is within 3 clicks from the main screen.</li> <li>• Tolerance: None.</li> <li>• Result: All functionalities are within 3 clicks from the main screen.</li> </ul>
<p align="center"><b>Integration with Data Visualization Subsystem</b> (System Testing)</p>	<ul style="list-style-type: none"> <li>• Procedure: Initiate and terminate data visualization repeatedly. See if the GUI can control the data visualization smoothly.</li> <li>• Tolerance: Error rate must be below 0.1%.</li> <li>• Result: The data visualization always starts normally in the latest version.</li> </ul>
<p align="center"><b>Integration with Image Acquisition Subsystem</b> (System Testing)</p>	<ul style="list-style-type: none"> <li>• Procedure: See if the GUI can adjust the CMOS camera effectively while updating widgets, such as progress bars, to reflect the changes.</li> <li>• Tolerance: Error rate must be below 0.1%.</li> <li>• Result: No adjustment failure noticed during testing.</li> </ul>

### 3.4 Image Processing and Visualization System

We tested our algorithm and implementation on multiple particles and velocities. Fig. 22 shows two successive images taken by our system that is used for visualization. Fig. 23 show the visualization of our software on these two images.

Fig. 23 shows the signal-to-noise ratio distribution on this sample image in the correlation calculation. Noted that if the signal-to-noise ratio is larger than 1.5, we will remove this vector from the visualization result. In 23, the blue vectors are actually calculated PIV vectors, and the red vectors are estimated using their neighbors' value.

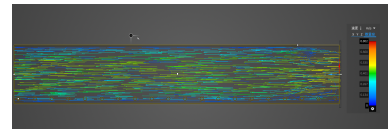
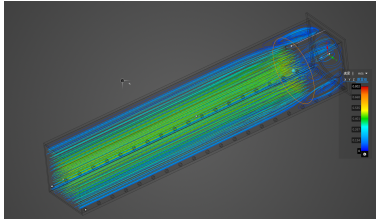


Figure 19: CFD results for version 3

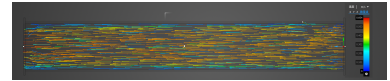
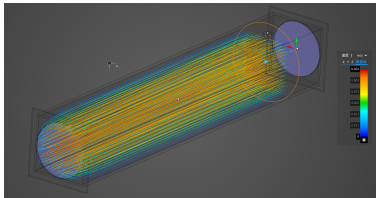


Figure 20: CFD results for version 4

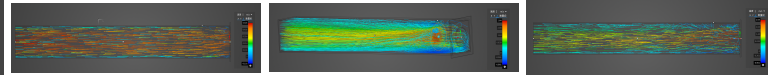
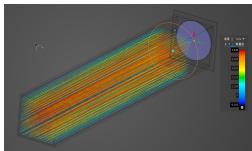


Figure 21: CFD results for version 5



Figure 22: Two successive images on styrofoam balls taken by the camera, FPS = 480.

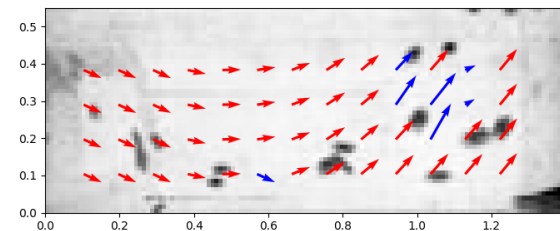
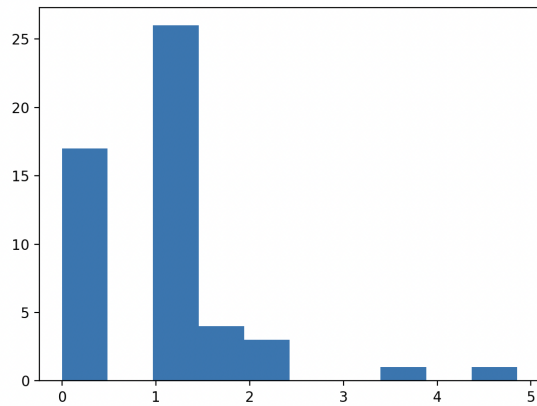


Figure 23: The visualization of the above two images.

## 4 Cost and Schedule

### 4.1 Cost

#### 4.1.1 Labor Cost

We assume that each person in our group works at least 10 hours a week on our project. Given that a Graduate Research Assistant at UIUC typically earns around \$40 per working hour, the total labor cost for our team will be

$$4\text{people} \times \$40\text{perhour} \times 10\text{hoursperweek} \times 10\text{weeks} = \$16000$$

#### 4.1.2 Parts Cost

Parts	Manufacturer	Cost
Camera	Zhonganweishi	¥ 898
Styrofoam Balls	Xinkang Industry	¥ 12
Laser	Qilan	¥ 99
Aluminum profiles and fittings	Zhongda Aluminum Industry	¥ 58
Aluminum Profile Accessories	“Twenty Years Old” Accessories Factory	¥ 24.8
Faster Blower	Yizhan firefight Industry	¥ 37
Slower Blower	Yunjia Industry	¥ 16.8
Touch Screen	Makebit	¥ 189
Raspberry Pi Package	Raspberry Pi Foundation	¥ 408
Nylon Mesh	Tonglewangye flagship store	¥ 39.12
Nuts and screws	Ending Industry	¥ 21.35
Acrylic Glue	Extraordinary power Industry	¥ 19.8
Folders	Pengyi	¥ 13.7
Anti-Laser Glasses	Pleasing Optics Industry	¥ 9.61
Sprayer	Hzschehope	¥ 18.31
One kilogram of quartz sand	Runjia Environmental Protection	¥ 6.5



<b>Total</b>		<b>¥ 1871.99</b>
--------------	--	------------------

Table 5: Parts Cost List

### 4.1.3 Total Cost

¥ 115654 + ¥ 1871.99 = ¥ 117525.99

## 4.2 Schedule

Week	Hanfei Yao	Yihui Chen	Yueming Yuan	Siyuan Qian
3/18-3/24	Explore mechanical algorithms for PIV system	Check the simple and low-cost implementation of PIV	Explore the image processing and OpenPIV usage	Explore low-cost single-board computer and touchscreen options
3/25-3/31	Build a small model to test different particles	Selection for our camera and laser	Write Python script for the basic components and test on laptop	Consider GUI functionalities and order Raspberry Pi and touchscreen
4/1-4/7	Build and test the overall simple model	Basic function testing with the particles and phone camera	Install required software on Raspberry Pi and do unit tests on Raspberry Pi	Initialize Raspberry Pi OS and test touchscreen integration
4/8-4/14	Build and test the optic system with Yihui	Test and adjust the laser and camera	Implement the static visualization	Implement and test basic functionalities of the GUI
4/15-4/21	Build parts for the real mechanical system	Test and adjust the laser and camera improve the degree of synergy	Verify the static visualization and implement the dynamic visualization	Implement GUI functionality for user fine-tuning of the CMOS camera

4/22-4/28	Assemble the parts and test the whole system	Assemble the system, improve image quality physically by adjust the mechanical design	Complete the implementations and add GUI control	Perform final software adjustments and test functionalities on the touchscreen with Yueming
4/29-5/5	Test the whole system	Test the whole system	Test the whole system	Test the whole system
5/6-5/12	Prepare for the demo	Prepare for the demo	Prepare for the demo	Prepare for the demo
5/13-5/19	Prepare for the demo	Prepare for the demo	Prepare for the demo	Prepare for the demo

Table 6: Schedule

## 5 Conclusion

### 5.1 Accomplishments

The final function of the product includes the following:

1. **Particle Velocity Detection:** Realized by continuous air circulation driven by a sophisticated channel system and a blower, utilizing tiny particles as tracer objects to detect particle movement in the flow channel. Different particles (e.g. Styrofoam balls, quartz sand, water droplets) can be used as tracer particles.
2. **Illumination and Image Acquisition:** A wooden box provides a dark environment and a laser source is used to illuminate a specific area in the flow channel to obtain a clear image of the particles, and a digital camera is equipped to capture and align the image of the particle motion and transmit it to an image processing system for accurate velocity calculations, data analysis and data visualization. We introduced multiple processing strategy and tuned the parameters to reduce the noise in the images and increase the accuracy.
3. **User-Friendly Interaction Platform:** A user-friendly platform offers a control panel for parameter adjustments, display of image acquisition previews, and real-time velocity field visualization. It also realizes the automatic generation of velocity vector diagrams, making fluid velocity data easier to understand and analyze. It can be flexibly adjusted and customized to meet different teaching or experimental needs. Multiple wires are centralized in one plug board to ensure convenient power control.

### 5.2 Uncertainties

Our experimental results for water droplets are not satisfying. The average diameter of the water droplets is around  $10\ \mu\text{m}$  -  $15\ \mu\text{m}$ . The speed of the droplets is around  $0.5\ \text{m/s}$  -  $2.5\ \text{m/s}$ , the minimum frame rate for getting high quality images to do math analysis and visualization is 270 fps which have a resolution of  $1280 * 720$ , under the actual area of  $200\ \text{mm} * 60\ \text{mm}$ , the pixel size of the camera is  $2.2\ \mu\text{m} * 2.2\ \mu\text{m}$ , the sensor size is  $1/3''$ ,  $\frac{2.2}{1/3 * 1000} < \frac{200}{1280}$ , so the resolution is too small to capture the motion of the small water droplet. A higher resolution camera is needed for better results on tiny water droplets.

The noise in the images still influence the PIV calculation. Due to the restriction of particle size and the camera, the noises are inevitable in this system. As shown in Fig. ??, 35 calculated correlation values our of 52 in the image has a larger noise than the signal (signal-to-noise ratio  $> 1$ ). Since we remove the outliers in the images and replaced by estimated values, the visualization may show a wrong flow on the whole picture when the image is not clear or the number of particles in one image is too small.

### 5.3 Future Work

Several upgrades and revises in subsystems are expected in the future improvements:

- **Physical System.** For the wooden box, a hinge could be added to open the top roof of the wooden box for easy operation.
- **Power Control System.** A general power system could be added in the design. In the expected power system, there is a switch installed in the outside of the wooden box to turn on the Raspberry Pi, and the power of the illumination system can be controlled by Raspberry Pi via the touch screen.
- **Flowing System.** A PP hollow board could be used to stabilize the airflow and make it relatively uniform.
- **Image Acquisition System.** If more adequate funding is available, a higher frame rate and higher resolution camera is expected for more accurate speed measurement for smaller particles or particles at a relatively high speed.
- **Image processing.** To further improve the accuracy of the PIV detection, we may introduce more methods to process the image and reduce the influence of the noise. We can possibly work on elaborating the background in the images.
- **GUI.** One can implement more sophisticated performance optimization of the GUI software, which eliminates GUI lagging during image processing. In addition, a separate GUI mode could support batch processing of captured images, instead of displaying data visualization frame-by-frame. After all images have been processed, the GUI software can display a batch of data visualization plots completely without lagging.
- **Material Selection.** When doing the demonstration, the acrylic flowing channel could have electrostatics, which make it hard to clean the remaining Styrofoam balls, so the building material can be changed to the Anti Static Acrylic or ESD Acrylic Sheets. Another reasonable method is change the Styrofoam ball to other materials less affected by the electrostatics.

### 5.4 Ethical Considerations

In this project, we consider following ethics and safety concerns:

#### 5.4.1 Ethics

- **Privacy.** The IEEE and ACM[6] codes also require engineers to respect the public's privacy. Since our design includes a voice control system, we need to ensure that the information we obtain will not be disclosed for other uses.
- **Educational impact.** Since our project will be applied to education, especially to demonstrate PIV to children, we need to ensure the system has a positive educational impact, reducing confusion and avoiding any harm to the users. For this rea-

son, we plan to create materials that explain the principles of the PIV measurement and operation manual of the control system in an age-appropriate manner.

### 5.4.2 Safety

The IEEE[7] code emphasizes the importance of prioritizing safety and health in engineering project, "to hold paramount the safety, health, and welfare of the public." This is strongly relevant to the most of the system in our design.

- **Laser Safety.** Safety notes in the LD-PS/5[8] states that The laser must only be used when integrated into a system that does not allow laser radiation to exit the system. Any eye and skin exposure to the light must be strictly prevented. Never operate the device without the cylindrical divergent lens. Never point the laser beam at humans animals or flammable materials. Fire, serious injury, or death might result from this action. Never adjust the laser while it is turned on. Safety goggles are suggested for use while operating.
- **PM Safety.** When the blower is blowing air, it can carry some particulate matter. The United States Environmental Protection Agency[9] declares that Exposure to such particles can affect both your lungs and your heart. Numerous scientific studies have linked particle pollution exposure to a variety of problems, including: premature death in people with heart or lung disease; nonfatal heart attacks; irregular heartbeat; aggravated asthma; decreased lung function; increased respiratory symptoms, such as irritation of the airways, coughing or difficulty breathing. People with heart or lung diseases, children, and older adults are the most likely to be affected by particle pollution exposure.
- **Blower Safety.** The operating blower could work at a speed of 20000 RPM. The OSHA training[10] states that using a compressed air blower presents potential hazards that can do serious harm to the user, as well as to other people working in close proximity. Flying chips, dust, and particles can be sent flying through the air at a high rate of speed and strike someone, causing cuts and abrasions to their skin or an eye. And if a compressed air blower is activated when placed directly against or near the skin or other body parts of a person, the high-pressure stream of air can actually pierce the person's skin, inject air or chemicals into their bloodstream, rupture an eardrum, or permanently damage an eyeball.

## References

- [1] J. Liu. "DEVELOPMENT AND ANALYSIS OF A LOW SPEED PARTICLE IMAGE VELOCIMETRY SYSTEM." (1998), [Online]. Available: [https://trace.tennessee.edu/cgi/viewcontent.cgi?article=11834&context=utk\\_gradthes](https://trace.tennessee.edu/cgi/viewcontent.cgi?article=11834&context=utk_gradthes) (visited on 03/06/2024).
- [2] R. Adrian. "particle imaging techniques for experimental fluid mechanics." (1991), [Online]. Available: <https://www.annualreviews.org/content/journals/10.1146/annurev.fl.23.010191.001401> (visited on 03/06/2024).
- [3] Raspberry Pi, *Plug and play raspberry pi usb webcam*, Accessed: 2024-03-27, 2023. [Online]. Available: <https://www.raspberrypi.com/tutorials/plugin-and-play-raspberry-pi-usb-webcam/>.
- [4] OpenCV, *cv::VideoCapture Class Reference*, Accessed: 2024-04-14, 2024. [Online]. Available: [https://docs.opencv.org/4.x/d8/dfe/classcv\\_1\\_1VideoCapture.html](https://docs.opencv.org/4.x/d8/dfe/classcv_1_1VideoCapture.html).
- [5] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.
- [6] ACM. "ACM Code of Ethics." (2018), [Online]. Available: <https://www.acm.org/code-of-ethics> (visited on 03/06/2024).
- [7] IEEE. "IEEE Code of Ethics." (2016), [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html> (visited on 03/06/2024).
- [8] Optolution. "LD-PS/5: OEM laser module for Particle Image Velocimetry (PIV)." (None), [Online]. Available: [https://optolution.com/media/ld-ps\\_datasheet.pdf](https://optolution.com/media/ld-ps_datasheet.pdf) (visited on 03/06/2024).
- [9] U. S. E. P. Agency. "Particulate Matter (PM) Pollution." (2023), [Online]. Available: <https://www.epa.gov/pm-pollution/health-and-environmental-effects-particulate-matter-pm> (visited on 03/06/2024).
- [10] O. training. "Controlling Hazards Associated with Compressed Air Blowers." (2016), [Online]. Available: [https://oshatraining.com/wp-content/uploads/OSHA\\_Training\\_Toolbox\\_Talk\\_-\\_Controlling\\_Hazards\\_Associated\\_with\\_Compressed\\_Air\\_Blowers.pdf](https://oshatraining.com/wp-content/uploads/OSHA_Training_Toolbox_Talk_-_Controlling_Hazards_Associated_with_Compressed_Air_Blowers.pdf) (visited on 03/06/2024).

## Appendix A Requirements and Verification

Requirements	Verification	Status (Y/N)
The particles can flow at a speed of 1 m/s - 3 m/s	We can use a high-speed camera to take successive pictures of the ball in motion so that we can calculate the instantaneous speed of the ball and thus determine whether the ball speed can reach the required speed.	Y
The inner space of the channel is closed and the small ball will not be blown out.	Turn on the blower continuously for a few minutes and observe if any of the small balls are blown out of the channel.	Y
The channel should have 6 cm * 6 cm cross section and 40 cm in length.	First, the dimensions are determined in the modeling software, and second, the actual dimensions are determined using precision measuring equipment such as vernier calipers.	Y

Table 7: Requirements and Verification for Flowing System

Requirements	Verification	Status (Y/N)
High resolution and high frame rate	Write a code to access the camera for the frame rate and resolution information, it should have around 190 fps at the resolution of 1280*1024 and 270 fps at the resolution of 1280 * 720.	Y
Synchronize with the illumination system	Check whether the input image is clear enough to do the data visualization.	Y
Work well with Raspberry Pi	Connect to the Raspberry Pi and check whether the image can be loaded in the Raspberry Pi, do the data visualization and show the result to the GUI surface.	Y

Table 8: Requirements and Verification for Image Acquisition System

Requirements	Verification	Status (Y/N)
--------------	--------------	--------------

The laser should be position-adjustable.	Check whether the laser source can illuminate the flowing channel in different angles.	Y
Synchronize with the image acquisition system.	Check whether the input image is clear enough to do the data visualization.	Y
Illuminate the flowing channel and particles well	Open the blower and turn on the laser, check whether the we can see the particles in the output of the camera.	Y

Table 9: Requirements and Verification for Illumination System

Requirements	Verification	Status (Y/N)
GUI response must appear on the touch-screen within 50 ms after the user input is received at the device driver level.	Measure input response time using a software timer from the moment of touch to the action being reflected on the GUI.	Y
GUI software must have an error rate of less than 0.1% for command execution.	Use a test script to execute a large number of commands and track the success rate over the total number executed.	Y
GUI must allow users to navigate to any primary function within 3 clicks from the home screen.	Conduct a test where users are observed navigating to different functionalities from the home screen. Record the number of clicks and verify that the primary functionalities are accessible within 3 clicks.	Y

Table 10: Requirements and Verification for Interactive User Interface System

Requirements	Verification	Status (Y/N)
--------------	--------------	--------------



<p>The error of calculated velocity magnitude at an single point is less than 20% and the angular deviation is less than 20 degrees compared with the designed one.</p>	<p>Compare the calculated result we get from the Python program on Raspberry Pi with the expected velocity we have in the flowing system.</p>	<p>Y</p>
<p>The static visualization shows an accurate vector (as defined above) in more than 75% areas of the picture.</p>	<p>Compare the visualized vectors with the theoretical result.</p>	<p>Y</p>
<p>The dynamic visualization displays the fluid within 3 seconds delays.</p>	<p>Test the visualization system for 30s, and compare the dynamic visualization and the recorded video of the actual fluid.</p>	<p>Y</p>

Table 11: Requirements and Verification for Data Visualization System