Fluid Motion Analysis

Using OpenCV and Python

S/19/435

Introduction

In fluid mechanics, understanding velocity distribution and shear stress is crucial for predicting flow behavior, especially in industrial processes, environmental studies, and engineering applications.

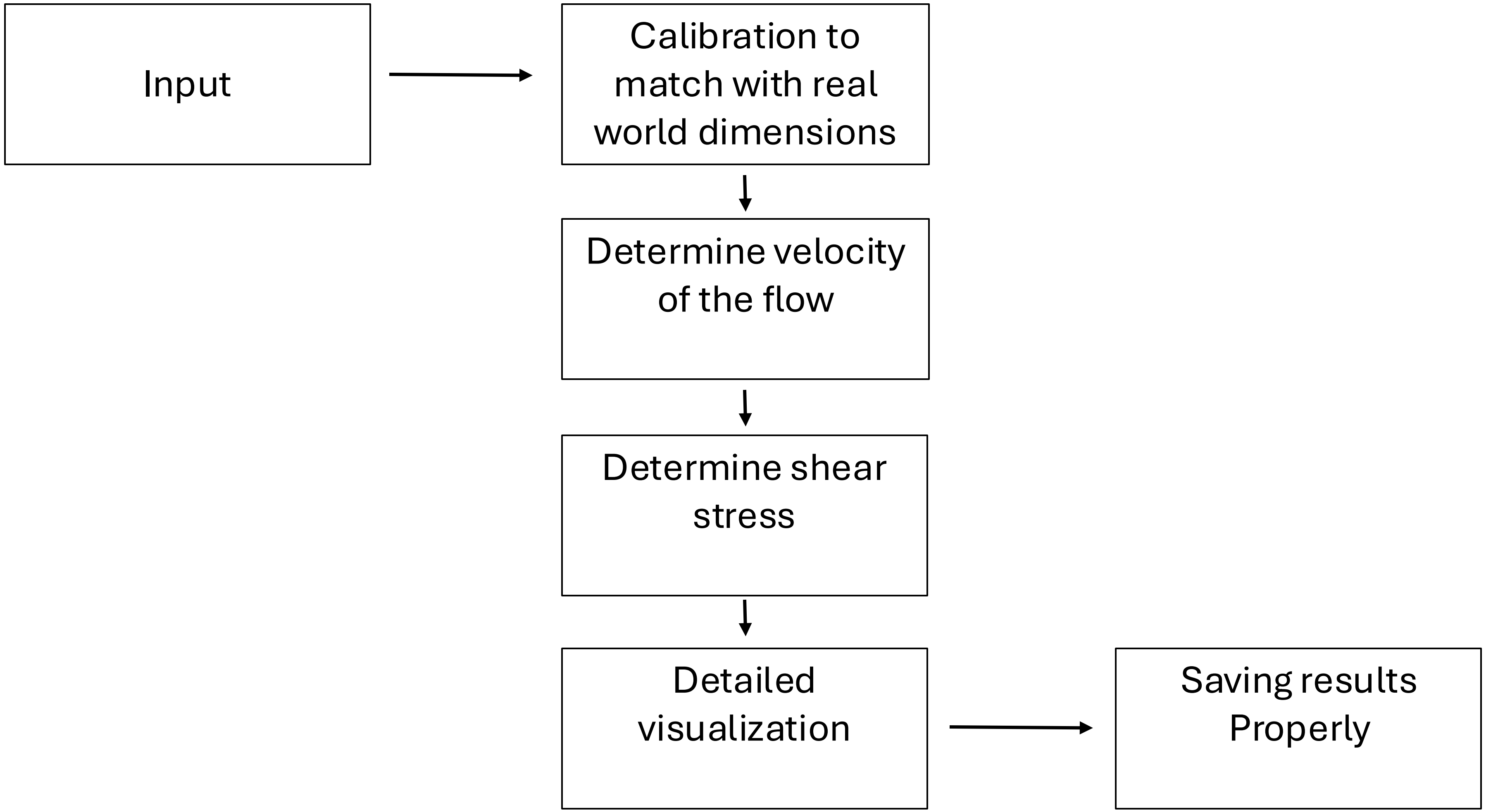
Addresses two main problems

- Velocity Calculation
- Shear Stress Calculation

Objectives

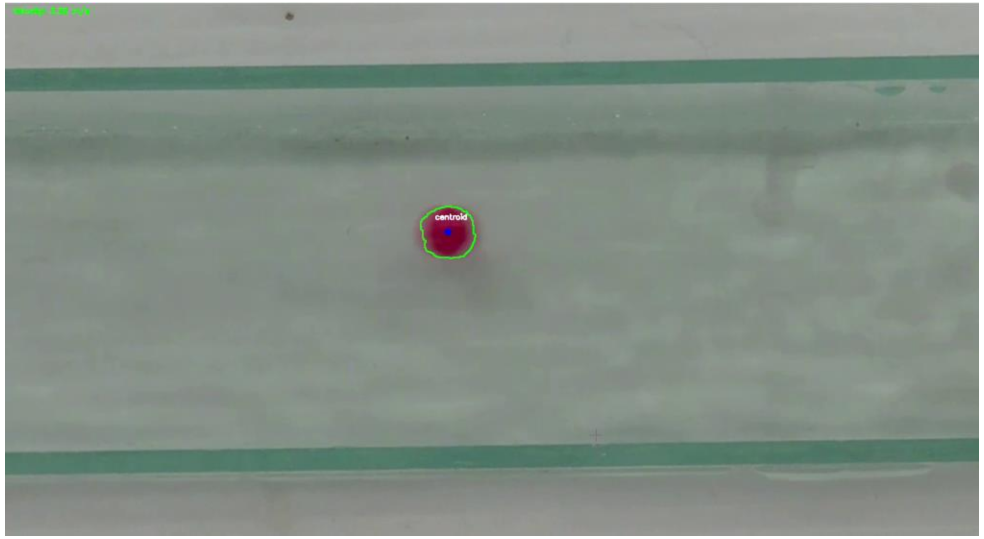
- tracking particle velocity
- calculating shear stress
- visualizing turbulent flows

System Workflow



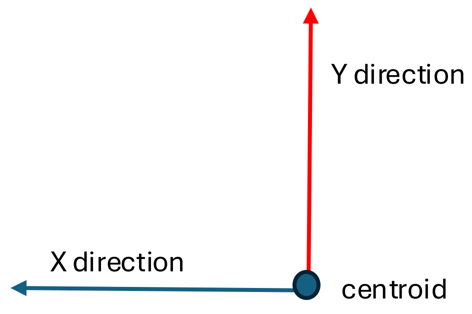
How to determine the Velocity ?

Take the floating particle position by using contour centroids at each frame.



$$\boxed{\text{Velocity}} = \left(\boxed{\begin{array}{c}\text{Current}\\ \text{frame}\\ \text{position}\end{array}} - \boxed{\begin{array}{c}\text{Previous}\\ \text{Frame}\\ \text{position}\end{array}} \right) \times \boxed{\text{fps}}$$

$$\boxed{\begin{array}{c}\text{Magnitude}\\ \text{of the}\\ \text{Velocity}\end{array}} = \left(\boxed{\begin{array}{c}\text{Velocity}\\ \text{Of}\\ \text{X direction}\end{array}}^{2} + \boxed{\begin{array}{c}\text{Velocity}\\ \text{Of}\\ \text{Y direction}\end{array}}^{2} \right)^{1/2}$$



How to determine the shear stress?

$T = \mu \ (\Delta y/\Delta v)$

Where:

- T = Shear stress (Pa)
- μ\muμ = Dynamic viscosity of the fluid (Pa·s)
- Δv\Delta vΔv = Change in velocity between fluid layers (m/s)
- Δy\Delta yΔy = Distance between layers (m)

```
du_dy = (velocity_magnitude[i] - velocity_magnitude[i - 1]) / (1 / pixels_per_cm_y)
shear = viscosity * du_dy
```

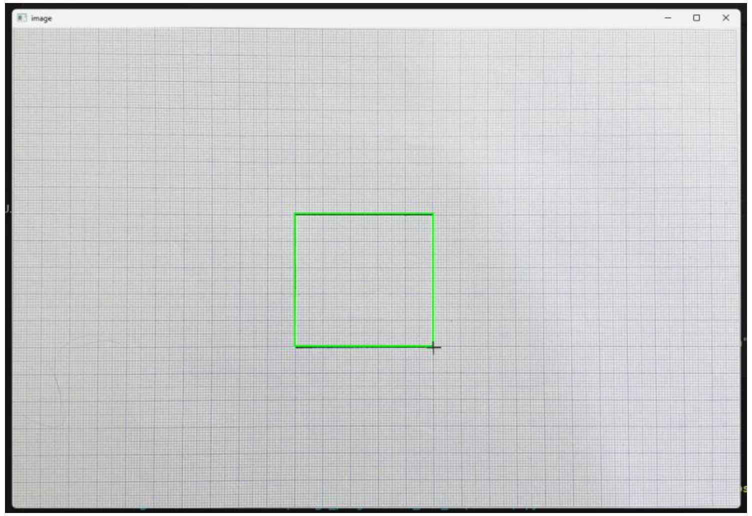
How to relate to real-world dimensions?

Purpose of Calibration:

Convert pixel measurements from video frames into real-world units.

Calibration Method:

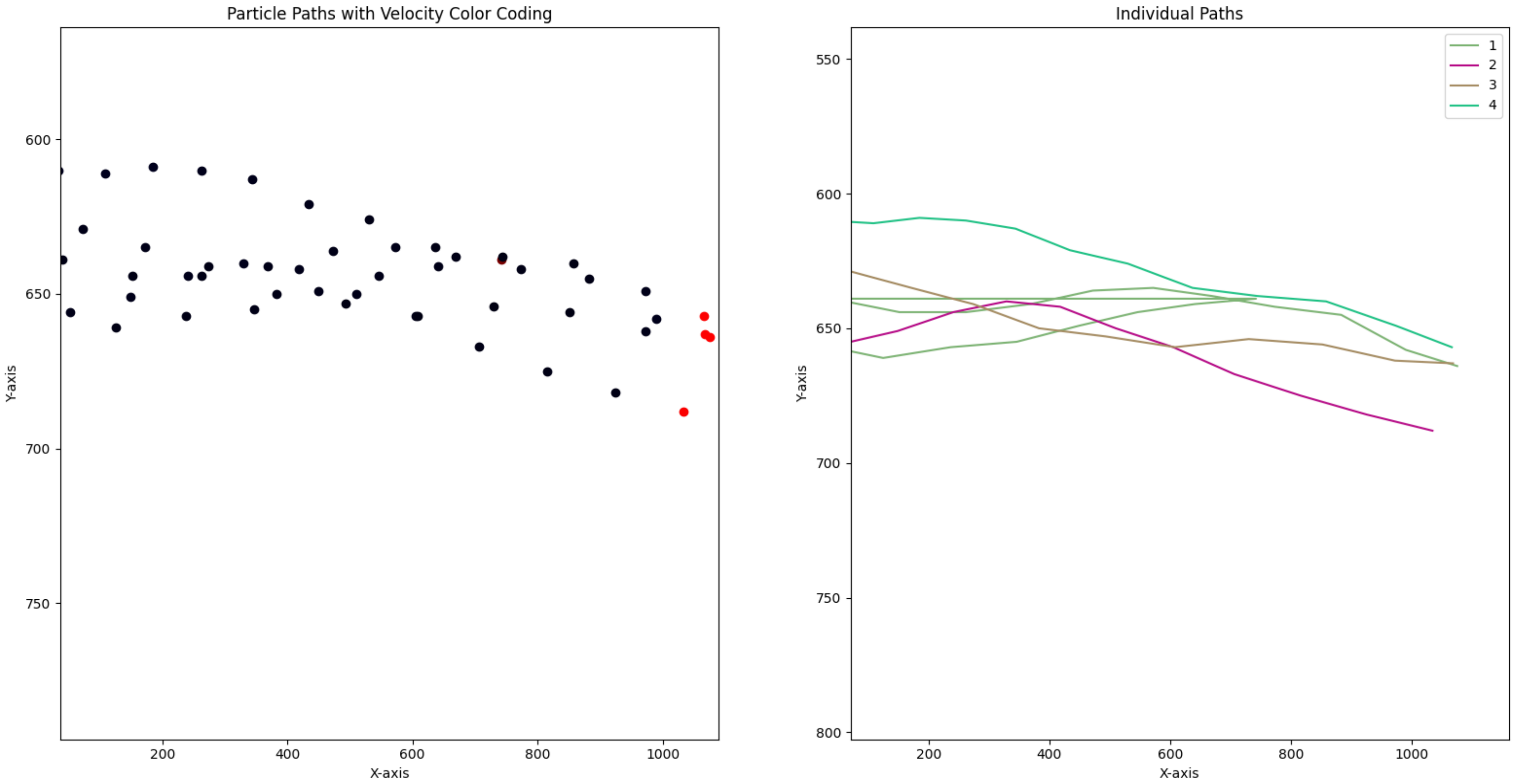
- **Load Calibration Image**:
  - A known reference object of a specific size is used (e.g., a square with known dimensions).
- **Manual Selection**:
  - User selects two points on the calibration image to define the known object's size.



```
def calculate_conversion_factor(ix, iy, x, y):
    width_pixels = abs(x - ix)
    height_pixels = abs(y - iy)
    return width_pixels / KNOWN_WIDTH_CM, height_pixels / KNOWN_HEIGHT_CM
```

```
Pixels per cm (Width): 22.6
Pixels per cm (Height): 20.6
```

Results Analysis and Visualization



Results Analysis and Visualization

Fluid ID	Date-Time	Avg Velocity (m/s)	Avg Shear Stress (Pa)
fluid_20240915_190601	2024-09-15 19:06:26	2.96	-0.07
fluid_20240915_190647	2024-09-15 19:07:28	1.40	-1.72

Watch a sample video demonstrating how the program works.

Potential further improvements

Challenges with Current Implementation:

- The program may struggle to track multiple moving particles accurately.
- Particle overlap or occlusion leads to errors in velocity and shear stress calculations.

Potential Solutions:

- Implement **blob detection**
- Use **Kalman filters** for tracking particles
- **Machine learning** approaches for robust object detection and tracking.

Thank you!