Optical Flow For Clustering CO Measurements From Low Cost Sensors

Student: Mari Jananashvili, Master (1st year), Applied Mathematics, Ivane Javakhishvili Tbilisi State University

Supervisor: Ramaz Botchorishvili, Kutaisi International University

GGSB 2020: Student Presentations at Kutaisi International University, Virtual Event, November 27, 2020

Outline

- Low Cost And High Precision Sensors Problem
- Optical Flow
- Lucas-Kanade Method
- Optical Flow Framework For Sensor Measurements
- Clustering

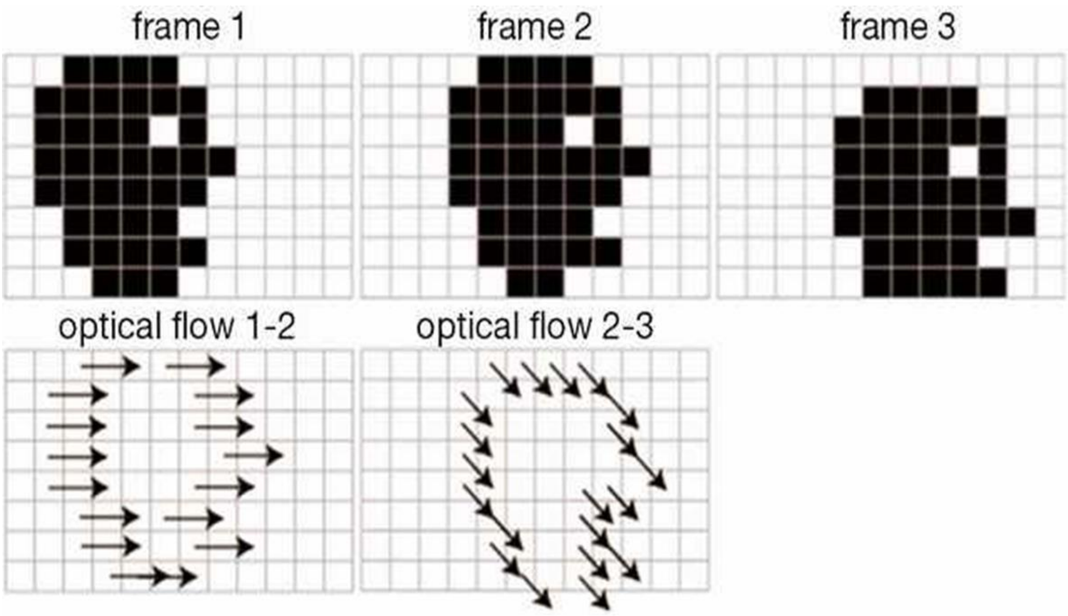
Low Cost And High Precision Sensors Problem

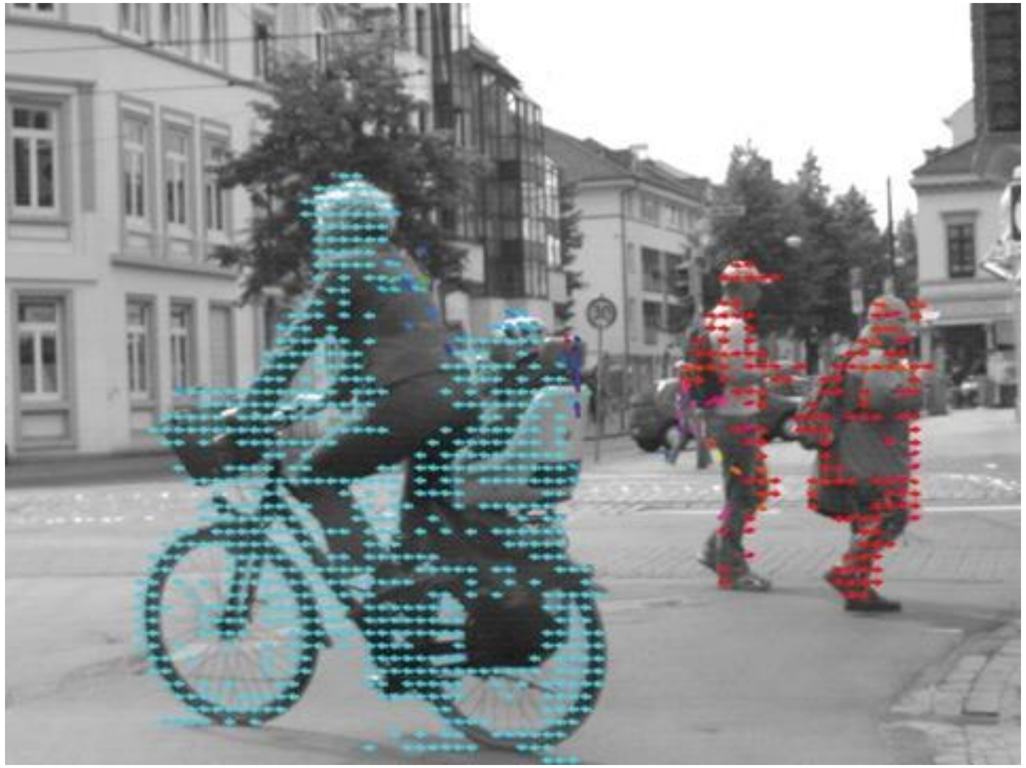
- Low Cost Sensors
  - Pros: Cheap, cost fraction of high precision sensor
  - Cons: Low accuracy of measurements
- High Precision Sensors:
  - Pros: High accuracy of measurements
  - Cons: Expensive

- ❑ Question 1: How to improve low cost measurements and compare to high precision sensors?
- ❑ Question 2: Is it possible to use Optical Flow for processing the data?

What is Optical Flow ?

- The pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene





https://themagicalworldofsakie.wordpress.com/2018/07/22/a-bit-of-optical-flow-background/

Optical Flow Equation

$$I_x u + I_y v + I_t = 0$$

I_x, I_y, I_t - Partial derivatives

(u, v) - Optical flow vectors

Lucas-Kanade method

- Flow vector (u,v) is **constant** and satisfy:

$$A\vec{u} = b \qquad A = \begin{pmatrix} I_x(q_1) & I_y(q_1) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{pmatrix} \qquad \vec{u} = \begin{pmatrix} u \\ v \end{pmatrix} \qquad b = \begin{pmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{pmatrix}$$

- $q_1, q_2, \ldots, q_n$ Sensor Measurements

- $I_x(q_i), I_y(q_i), I_t(q_i)$ partial derivatives, evaluated at the point at the q_i current time.

Lucas-Kanade method

- Minimize $\left\| A\vec{u} - b \right\|^2$

$$A\vec{u} = b$$

$$\Downarrow$$

$$A^T A\vec{u} = A^T b$$

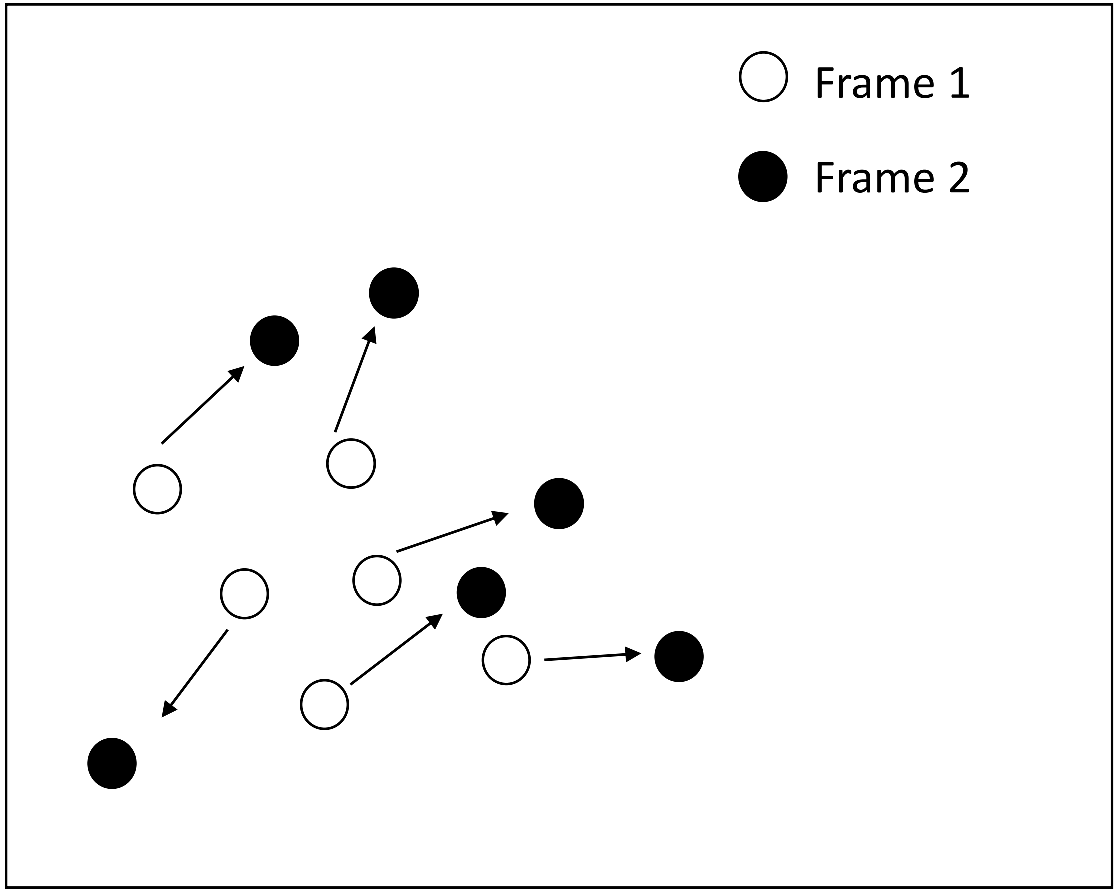
$$\Downarrow$$

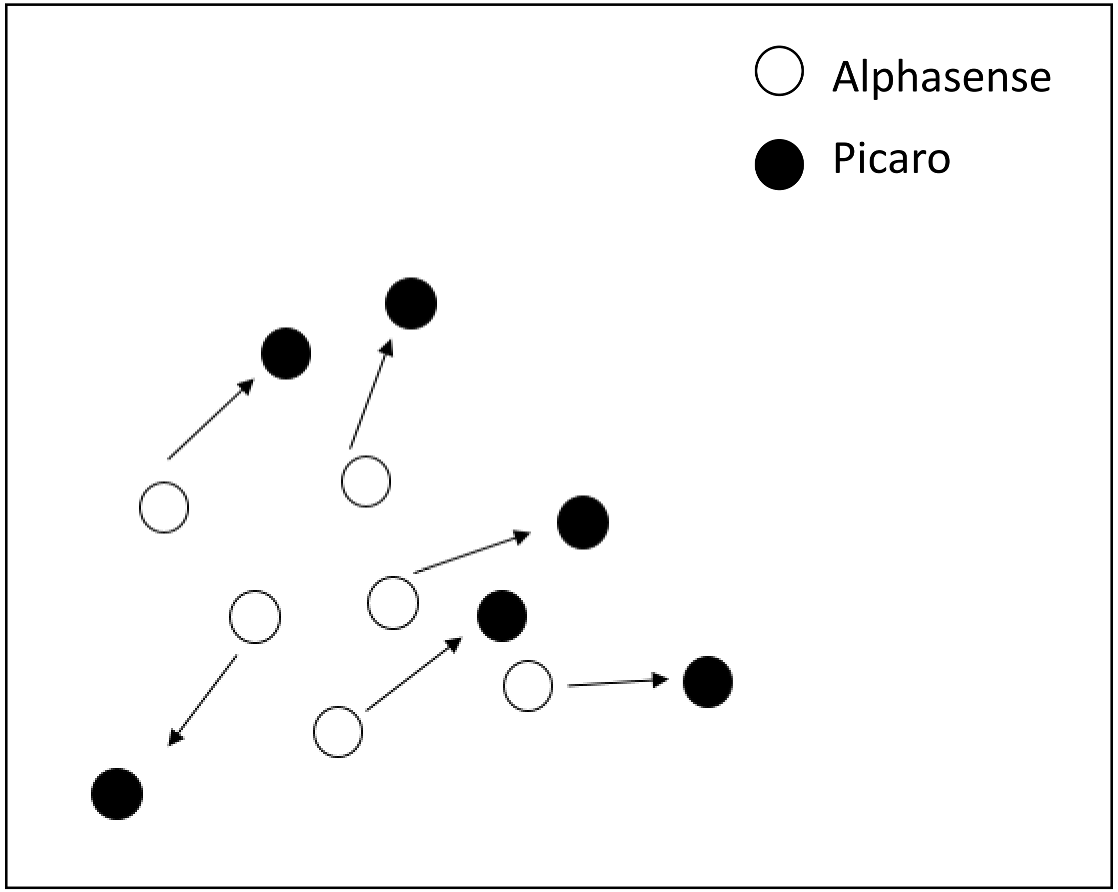
$$\vec{u} = (A^T A)^{-1} A^T b$$

The method of **least squares**

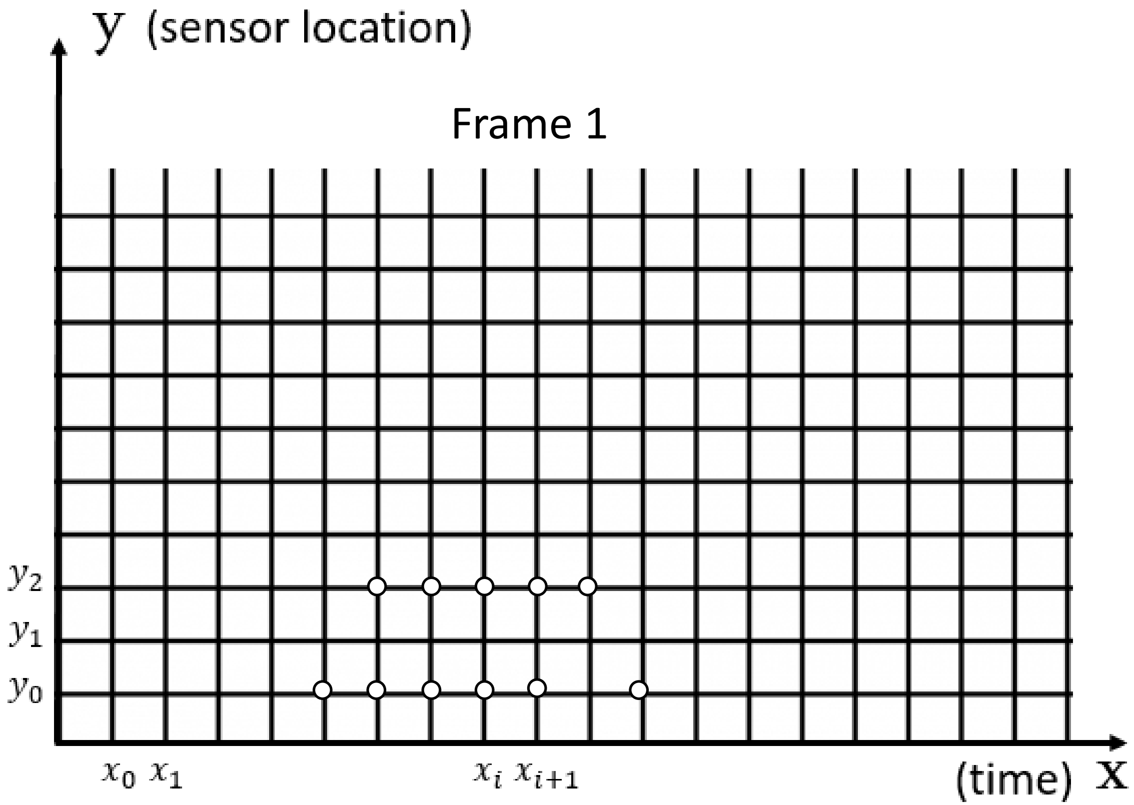
$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} I_x(q_i)^2 & \sum_{i=1}^{n} I_x(q_i) I_y(q_i) \\ \sum_{i=1}^{n} I_y(q_i) I_x(q_i) & \sum_{i=1}^{n} I_y(q_i)^2 \end{pmatrix}^{-1} \begin{pmatrix} -\sum_{i=1}^{n} I_x(q_i) I_t(q_i) \\ -\sum_{i=1}^{n} I_y(q_i) I_t(q_i) \end{pmatrix}$$

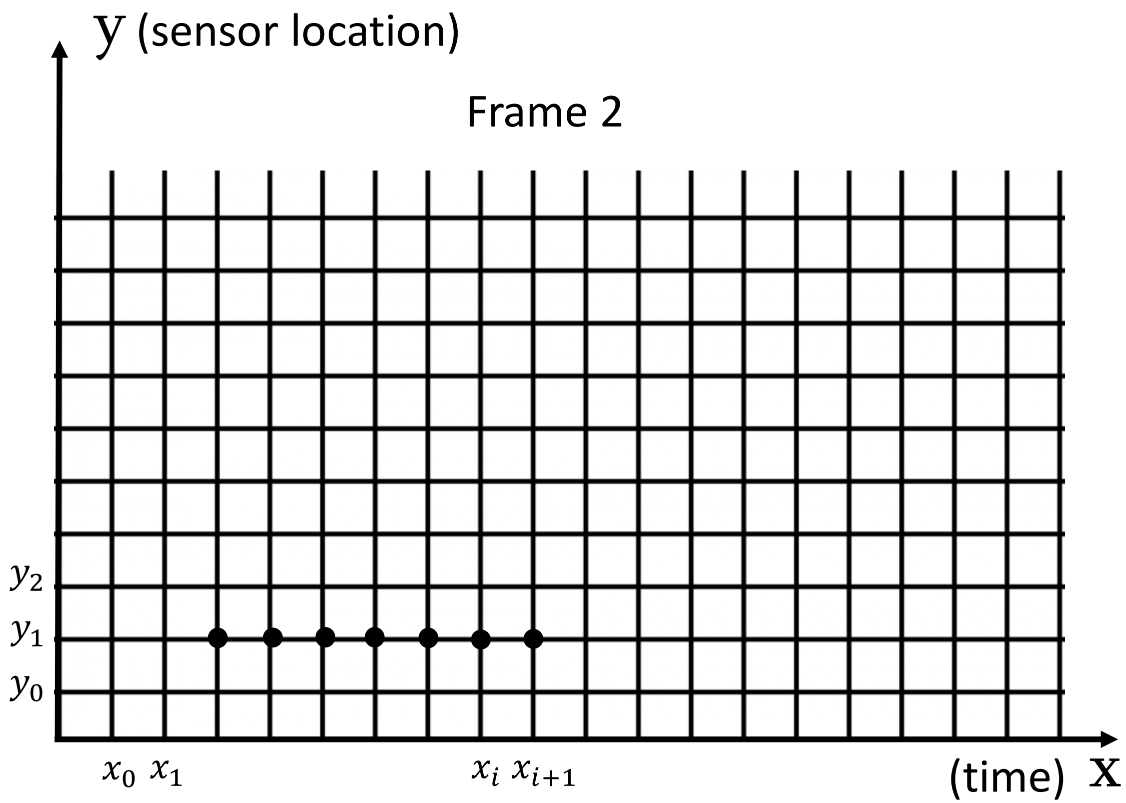
Optical Flow Framework For Sensor Measurements





Discretization





- ● $P - Picaro\ measurement$
- ○ $S - Alphasense\ measurement$

$x_i - Moment\ of\ time$

$y_0, y_2 - Alphasense\ location$

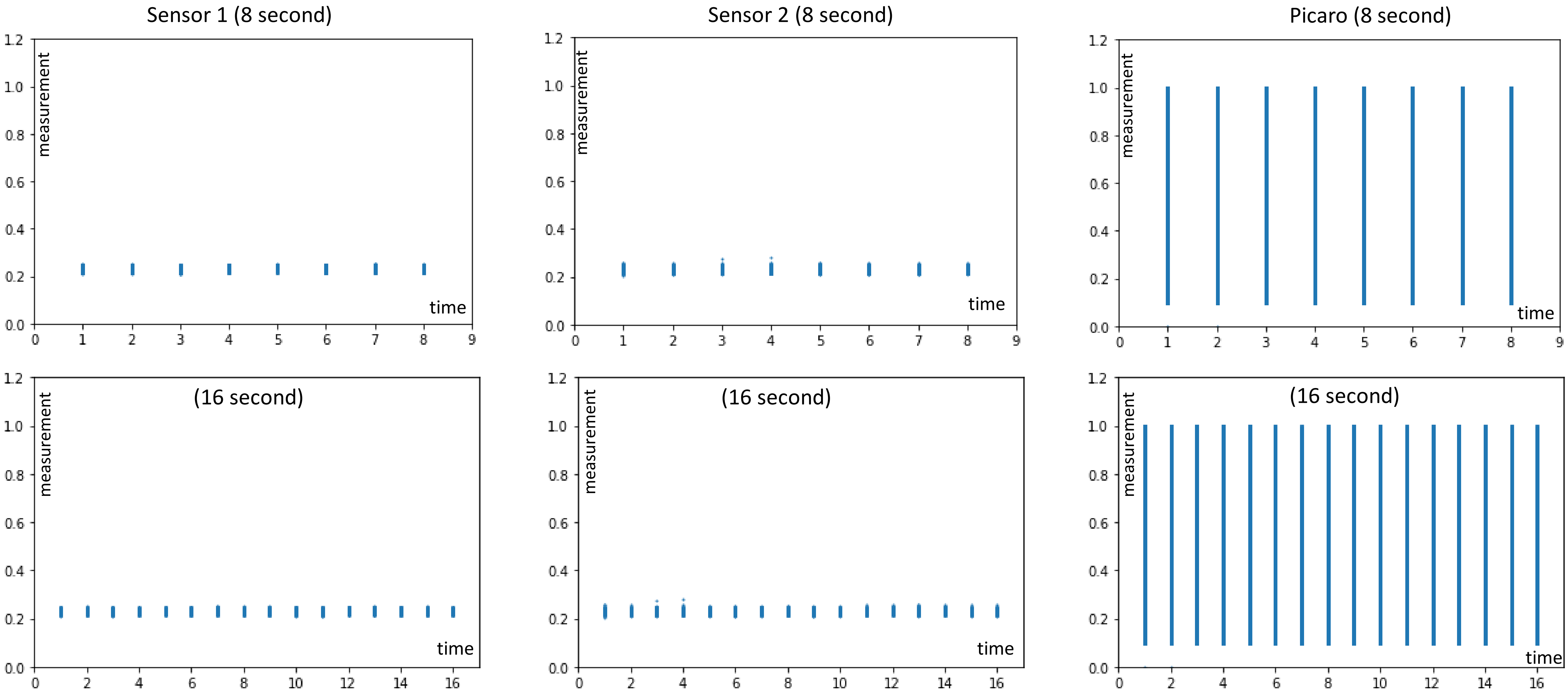
$y_1 - Picaro\ location$

$$I_x = \frac{1}{2h}\left(S_{i+1,0} - S_{i,0} + S_{i+1,2} - S_{i,2}\right)$$

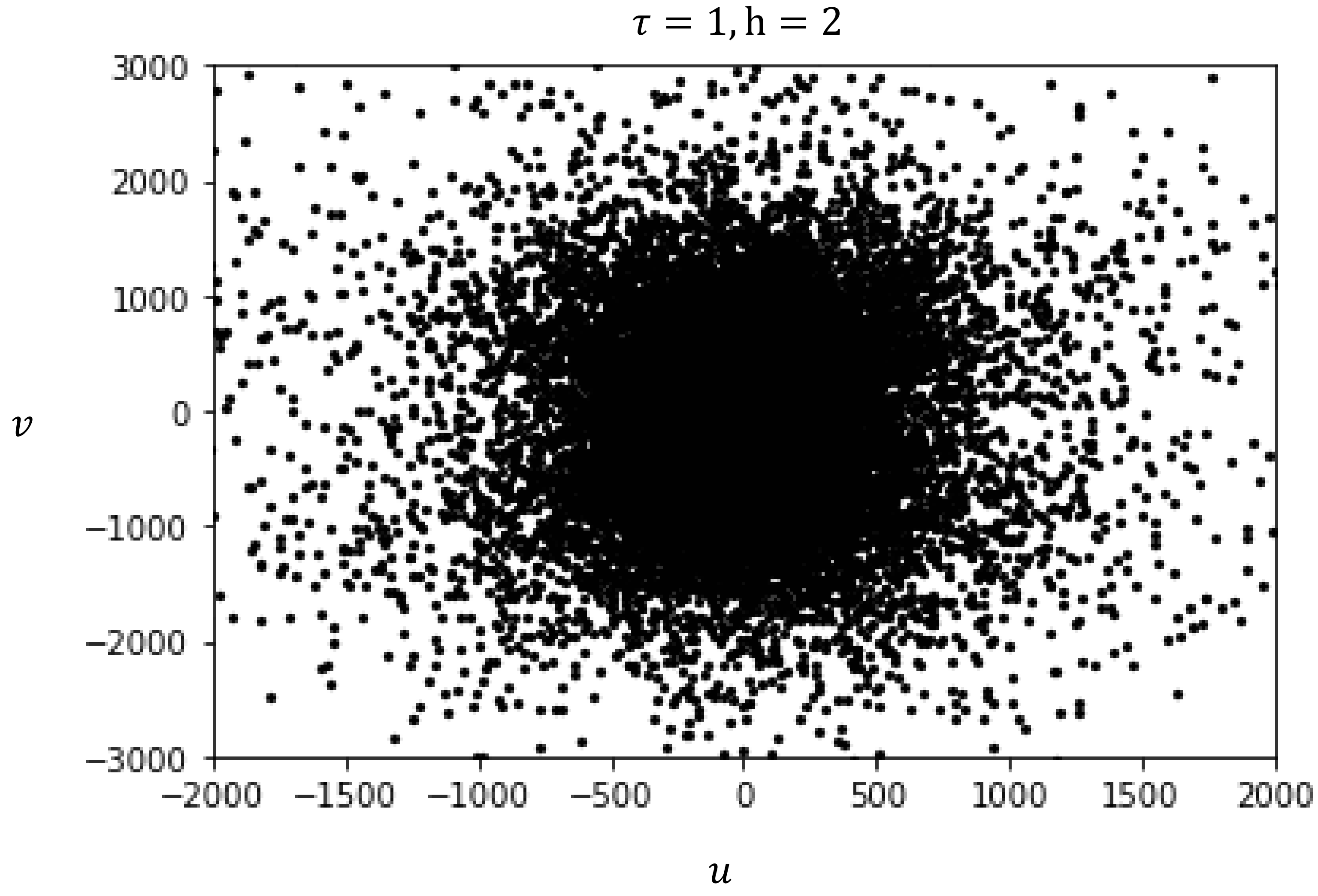
$$I_y = \frac{1}{2h}\left(S_{i+1,2} - S_{i+1,0} + S_{i,2} - S_{i,0}\right)$$

$$I_t = \frac{1}{2\tau}(P_i + P_{i+1}) - \frac{1}{4\tau}\left(S_{i,0} + S_{i,2} + S_{i+1,0} + S_{i+1,2}\right)$$

- Sensor Measurements from SMART|AtmoSim-Lab, shared by Dr. Giorgi Jibuti
- Data Processing and cleaning by Marta Melia (Received as a row data)

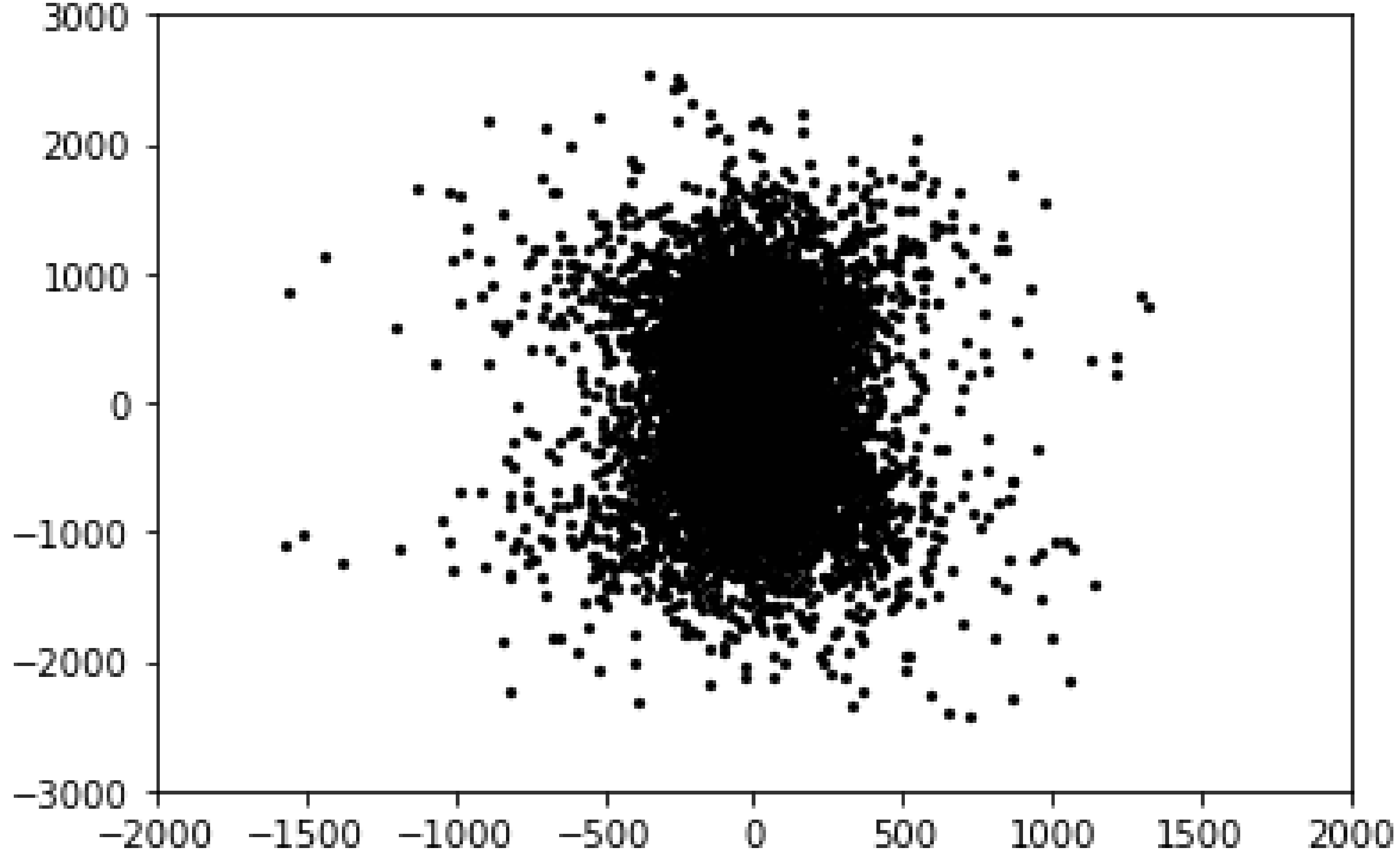


(u,v) points in every 8 second



(u,v) points in every 16 second

$\tau = 1, \mathrm{h} = 2$

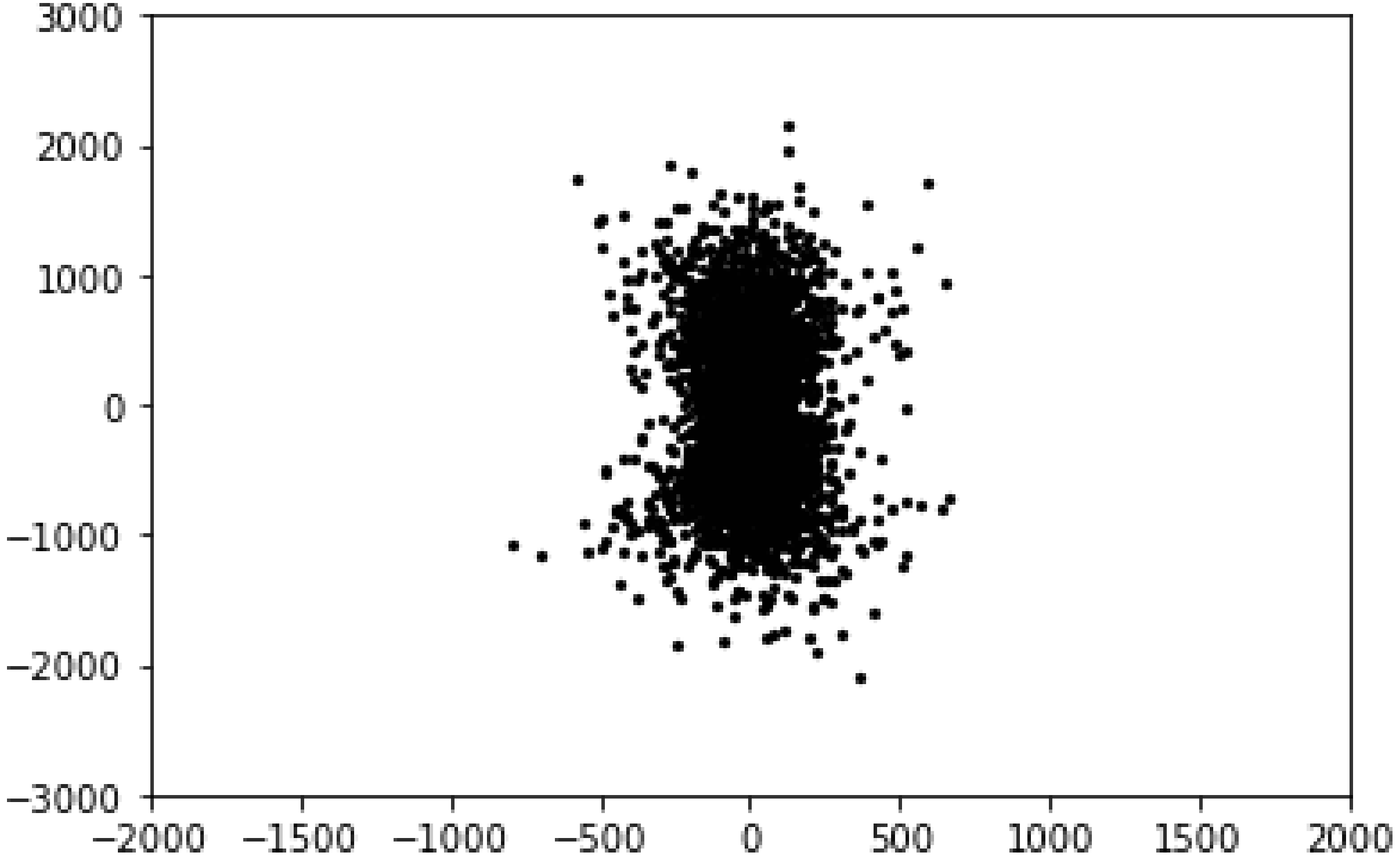


v

u

(u,v) points in every 30 second

$\tau = 1, \mathrm{h} = 2$

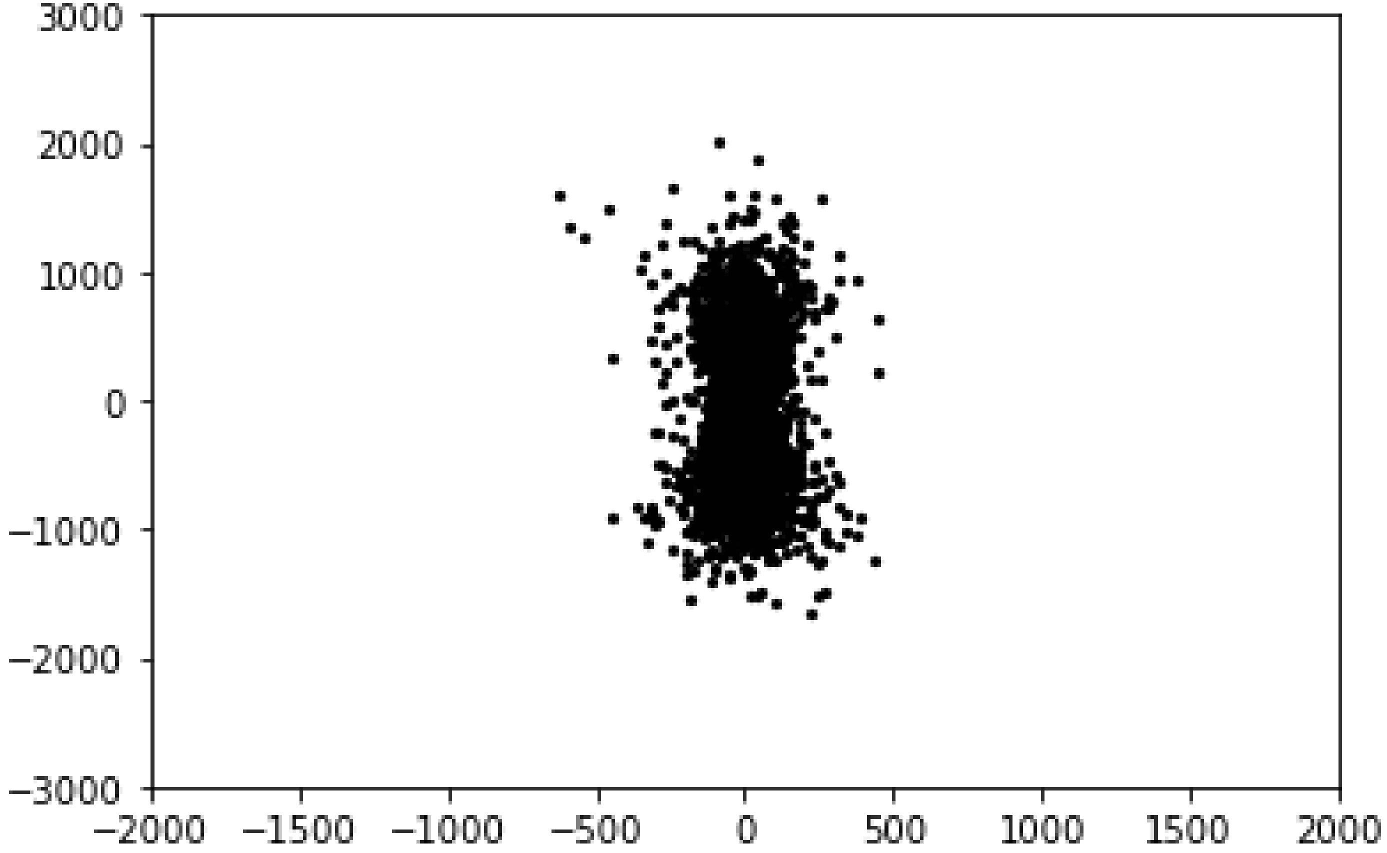


v

u

(u,v) points in every 60 second

$\tau = 1, \mathrm{h} = 2$



v

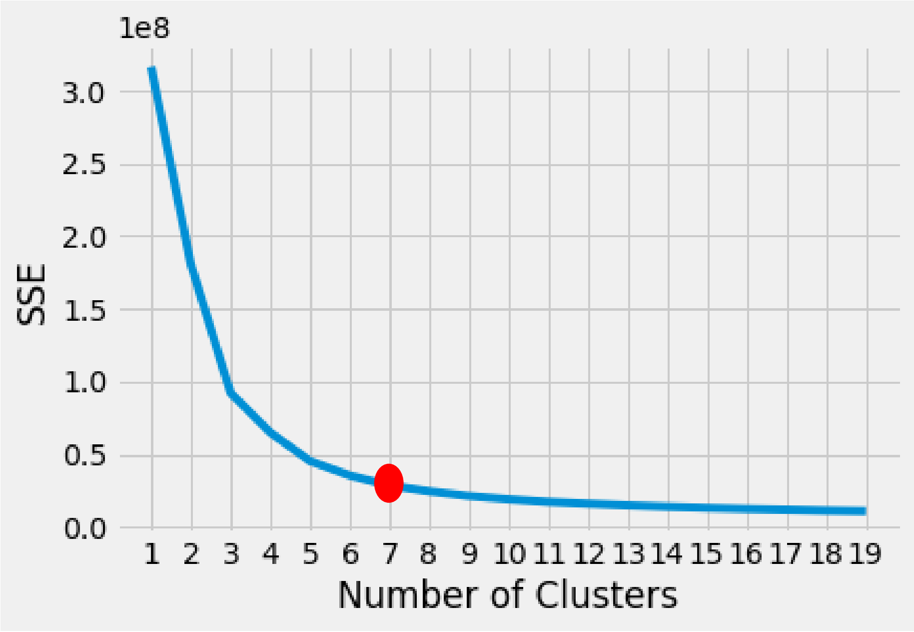
u

K-means algorithm- clustering

```
Specify the number k of clusters to assign.
Randomly initialize k centroids.
repeat
    expectation: Assign each point to its closest centroid.
    maximization: Compute the new centroid (mean) of each cluster.
until The centroid positions do not change.
```

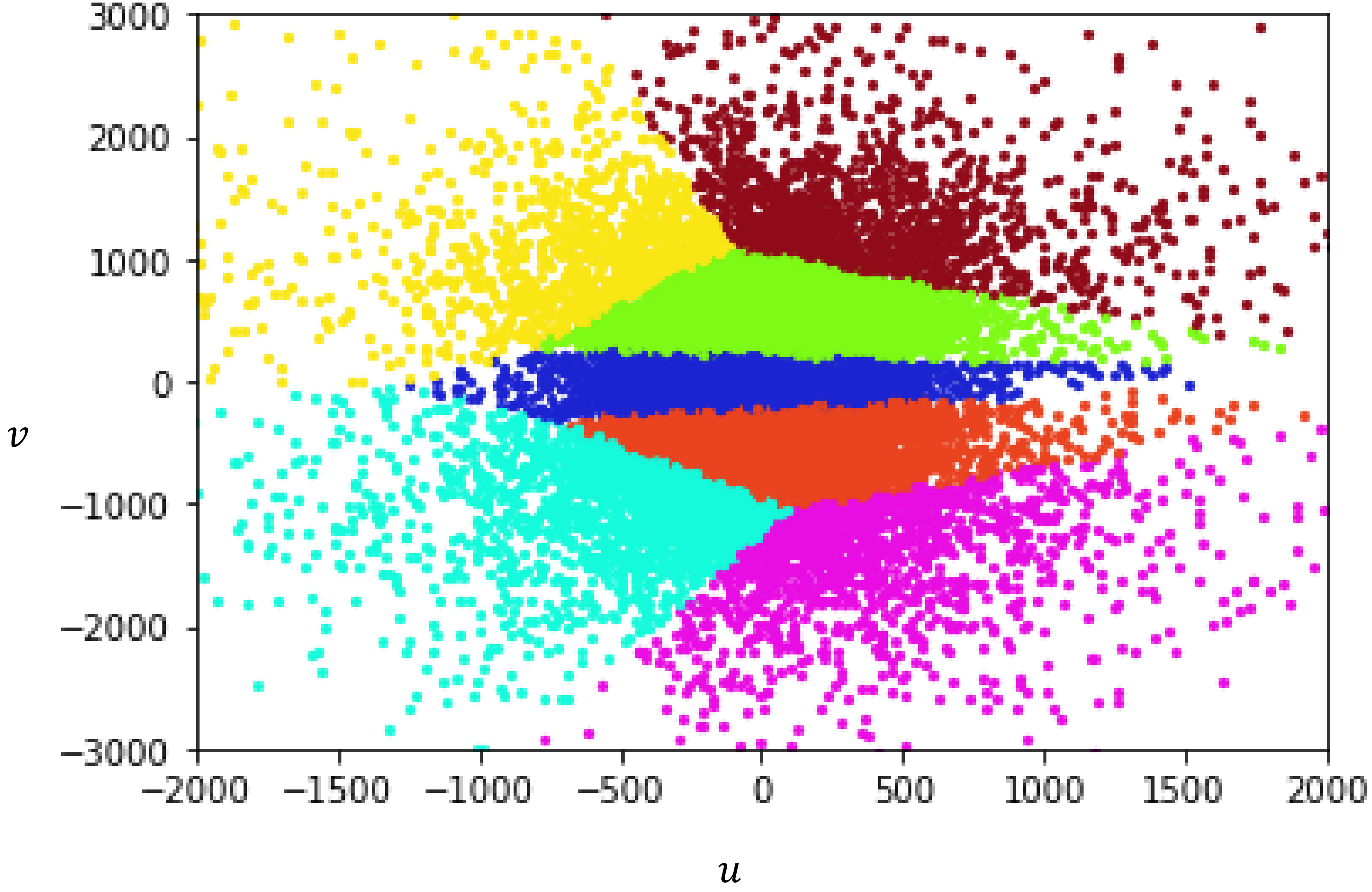
Elbow method: run several *k*-means, increment k with each iteration, and record the **SSE**

SSE is defined as the sum of the squared distance between centroid and each member of the cluster.

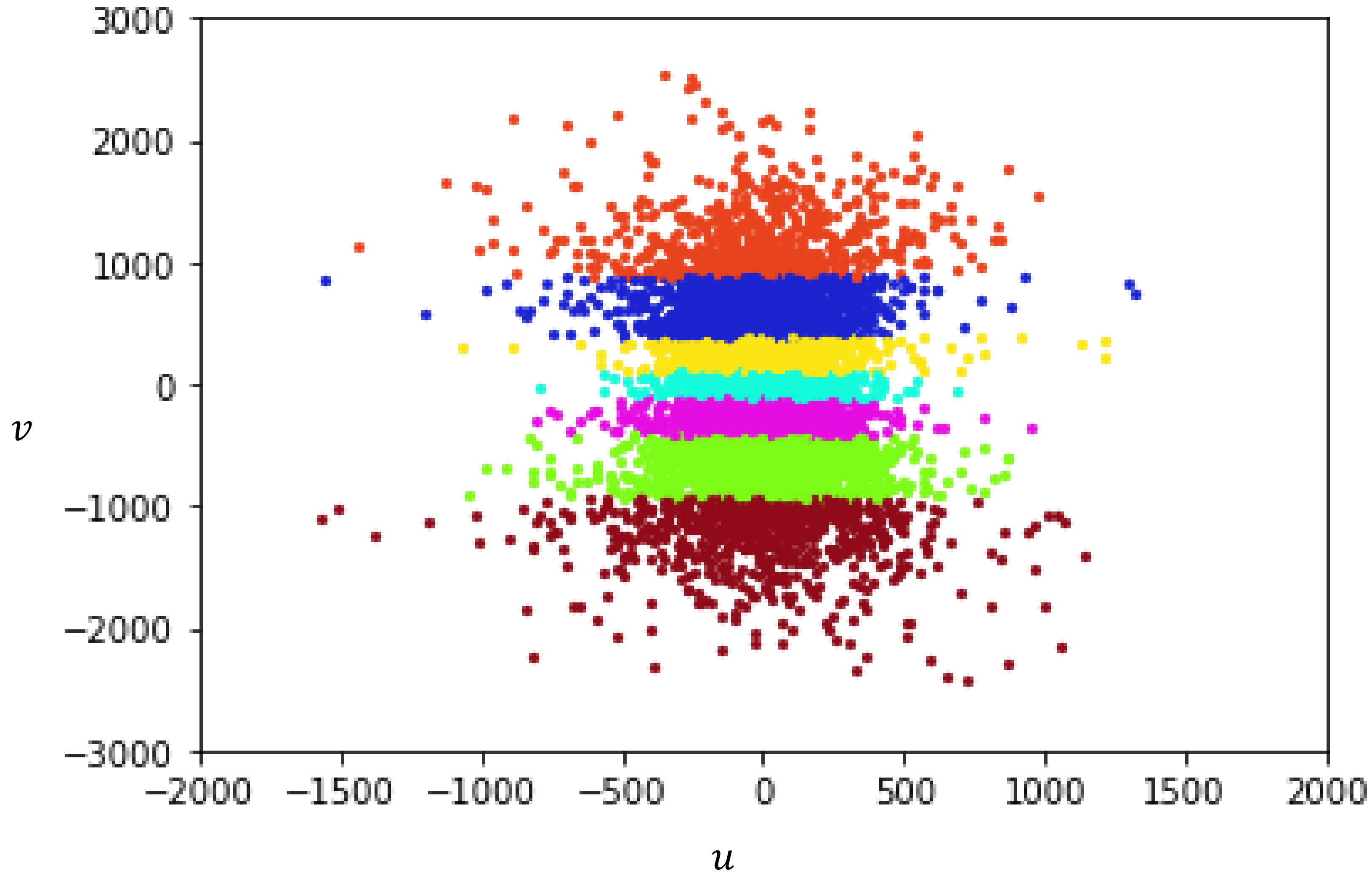


https://realpython.com/k-means-clustering-python/

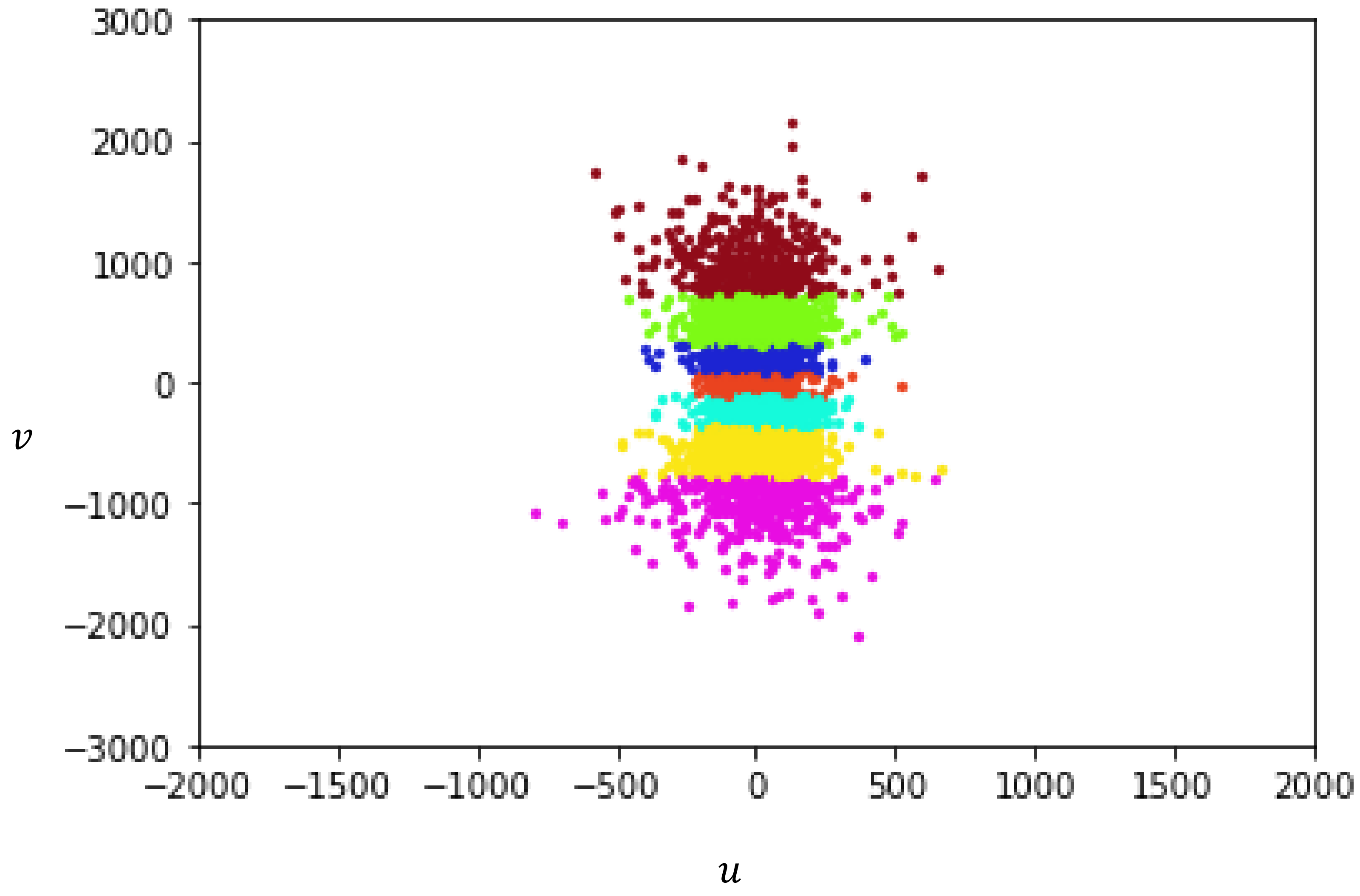
Clustering results (8 second)



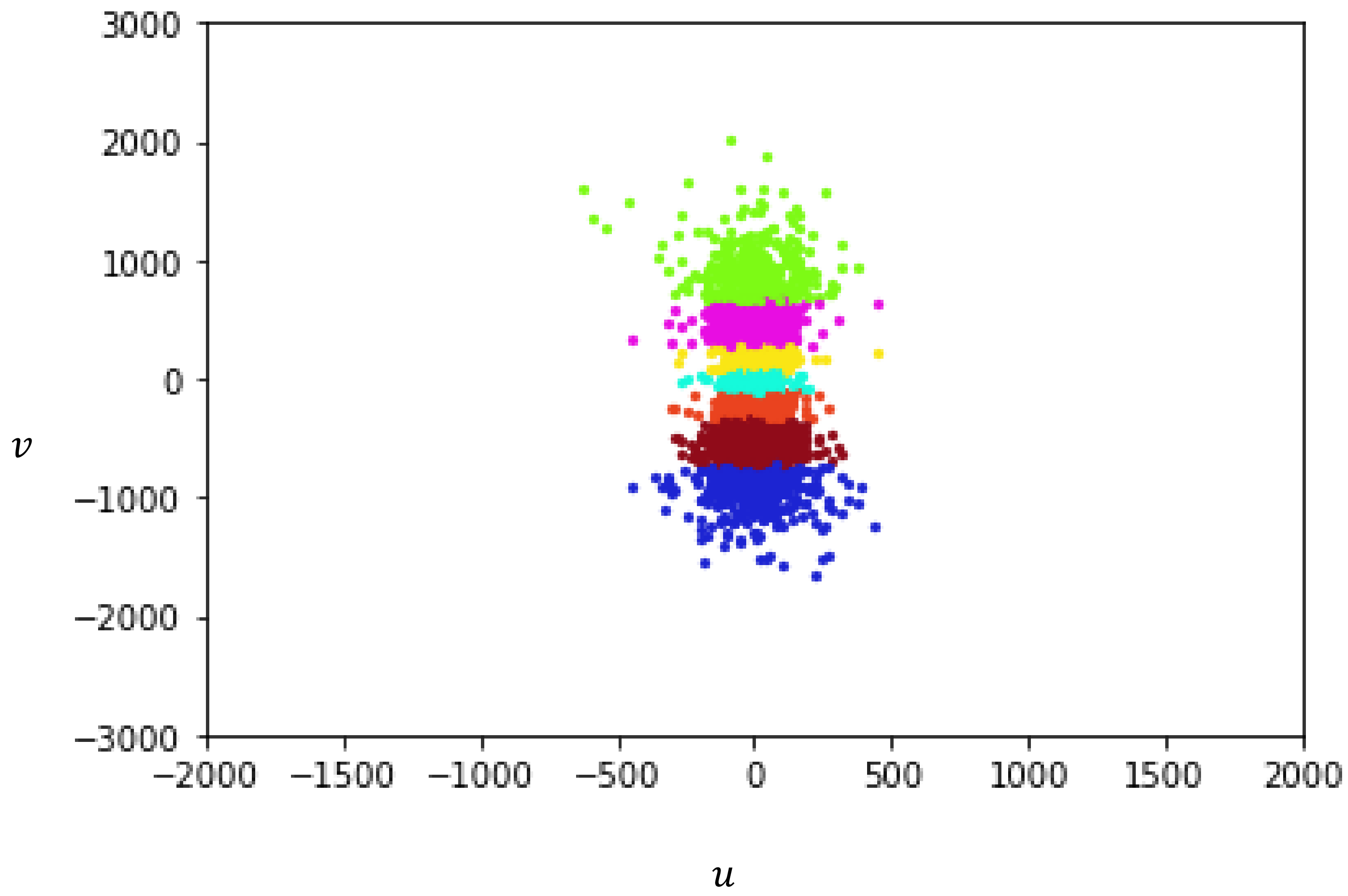
Clustering results (16 second)



Clustering results (30 second)



Clustering results (60 second)



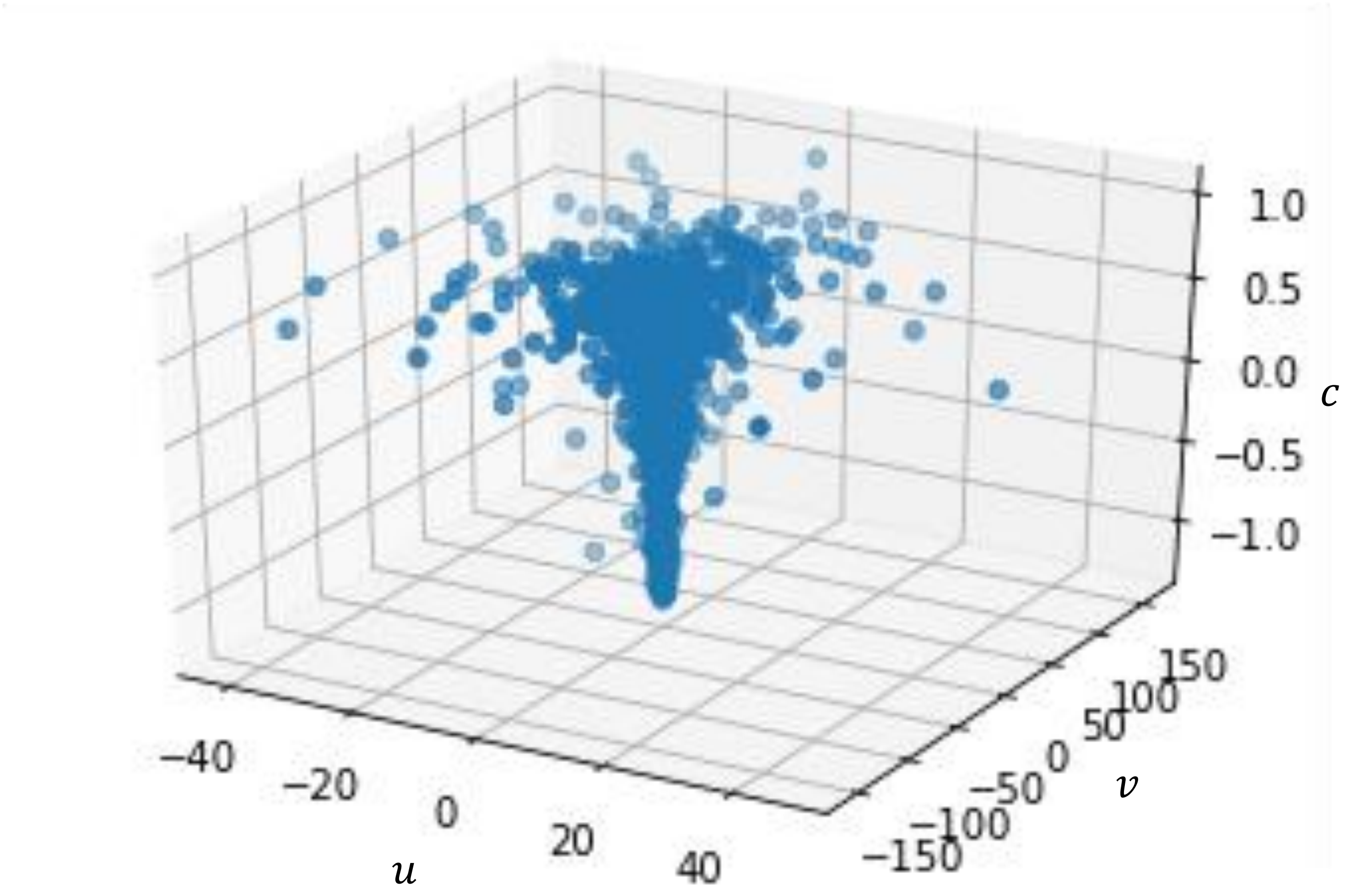
Optical flow equation with right hand side

$$I_x u + I_y v + I_t = Ic$$

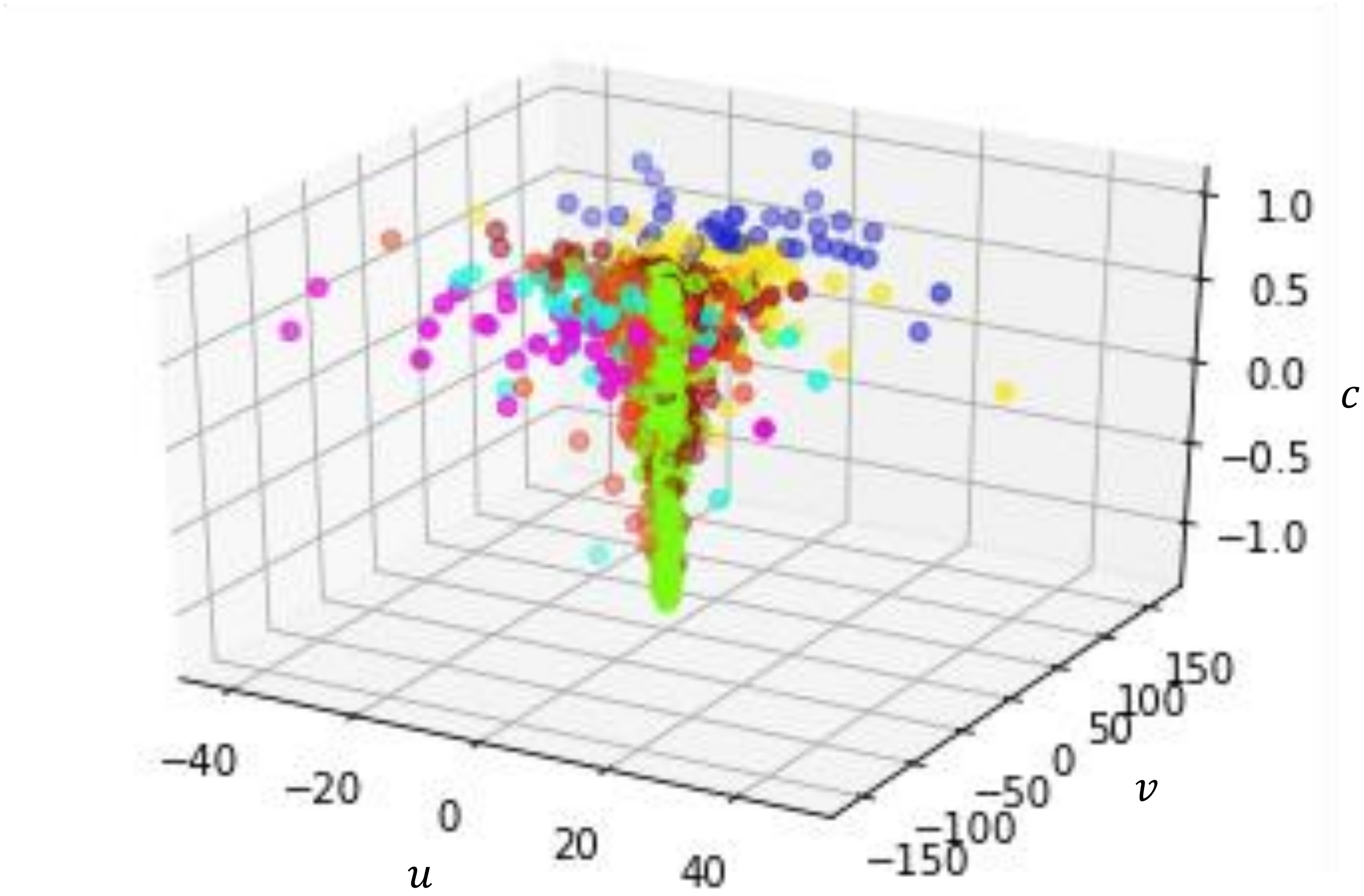
$$I_x = \frac{1}{2h}\left(S_{i+1,0} - S_{i,0} + S_{i+1,2} - S_{i,2}\right)$$
$$I_y = \frac{1}{2h}\left(S_{i+1,2} - S_{i+1,0} + S_{i,2} - S_{i,0}\right)$$
$$I_t = \frac{1}{2\tau}(P_i + P_{i+1}) - \frac{1}{4\tau}\left(S_{i,0} + S_{i,2} + S_{i+1,0} + S_{i+1,2}\right)$$
$$I = \frac{\tau}{2}(P_i + P_{i+1})$$

https://www.researchgate.net/publication/3193253_Computing_optical_flow_with_physical_models_of_brightness_variation

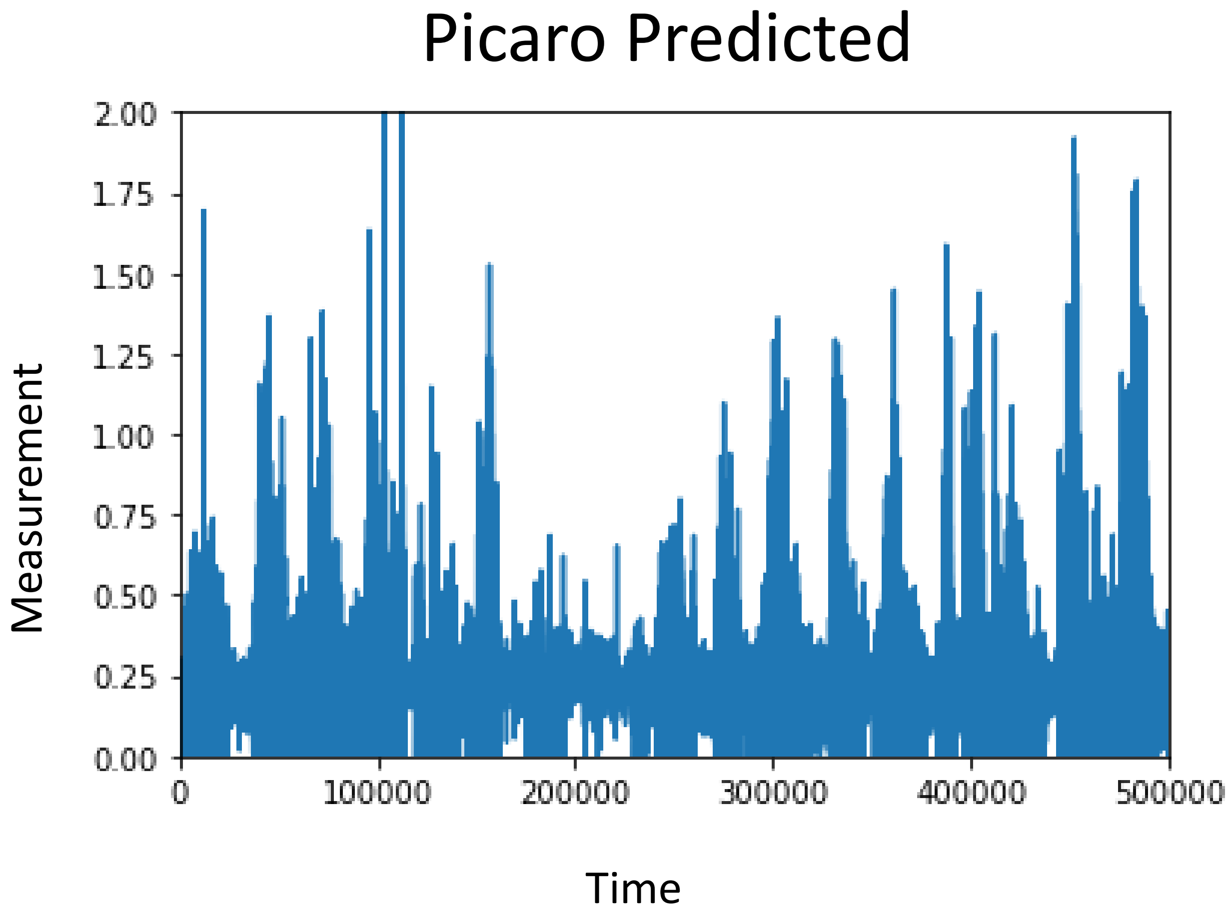
(u,v,c) points in every 60 second)

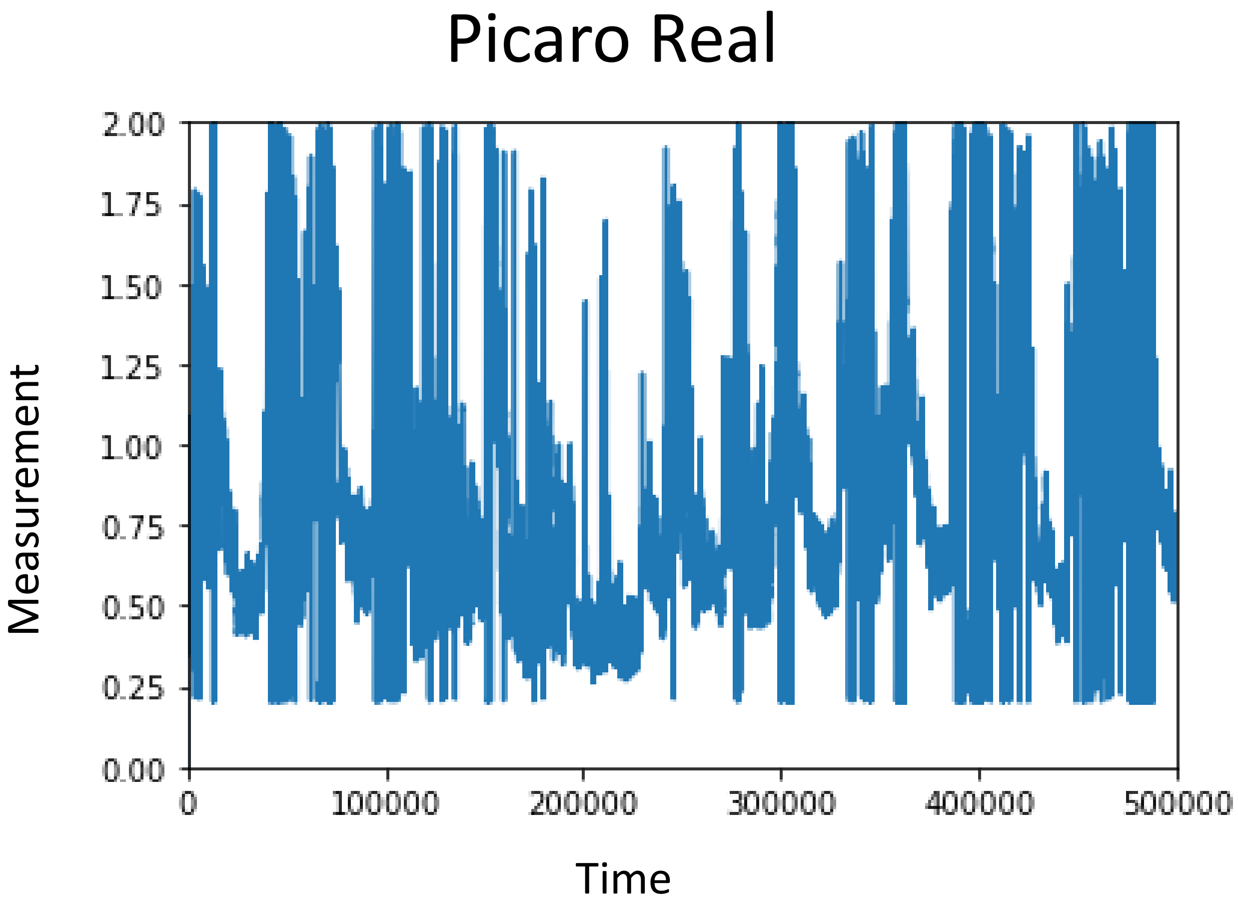


Clustering results (60 second)



Picaro predicted/real results in 3D (60 second)





$$Absolute\ error = |Predicted - Real|$$

89% of points are below absolute error=0.1
94% of points are below absolute error=0.2

Thank You!