Carnegie Mellon University

Embedded LS-PIV for Measuring Stream Flows

Ashish Dwivedi, Jun Taguchi, Justin Nguyen, Kyle Liang

18747 - Fall 2020

12/10/2020

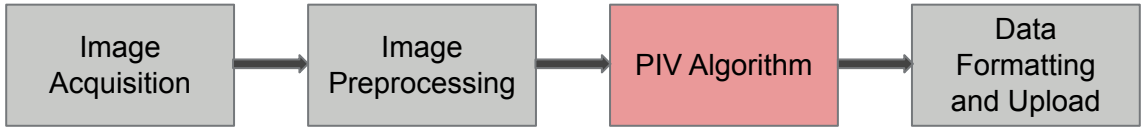
Problem Statement

The USGS measures stream flows since they are “critical for long-term tracking and modeling/forecasting to ensure that Federal water priorities and responsibilities can be met” [1]. PIV is a widely used algorithm used for measuring stream flows using optical data, but can be computationally intensive, so it is performed as a post-processing step after data collection[2]. Currently no open-source systems exist which can perform PIV in real-time on the remotely deployed sensors which collect the data.

1. https://www.usgs.gov/special-topic/water-science-school/science/how-streamflow-measured
2. Legleiter, C.J.; Kinzel, P.J. Inferring Surface Flow Velocities in Sediment-Laden Alaskan Rivers from Optical Image Sequences Acquired from a Helicopter. *Remote Sens.* **2020**, *12*, 1282.

Stakeholder Use Cases

- Bridge mounted PIV sensor



- UAS mounted

Components of the pipeline our project is specifically scoped for

Carnegie Mellon University

Stakeholder Requirements

- Ported PIV algorithm for stream flow measurement to embedded device for real time in-situ measurements
  - Measurement updates faster than current standard: 15min
  - Ported algorithm accuracy matches PIVLab and ground truth measurements
  - Lifetime of 3 months to a year

Our Project Goals

- Create an embedded version of the PIV algorithm targeted to run in real-time on embedded systems
  - Target the CommonSense Platform
  - Benchmark accuracy against PIVLab implementation
  - Benchmarks accuracy against ground-truths computed previously by USGS
- Optimize algorithm & software architecture:
  - Energy use
  - Computational time
  - Application specific requirements
- Clearly define energy budget and interfaces to our CommonSense code and start early system design for final end product

Project Design Use Cases

PIV algorithm on CommonSense can be viewed as solely compute MPU

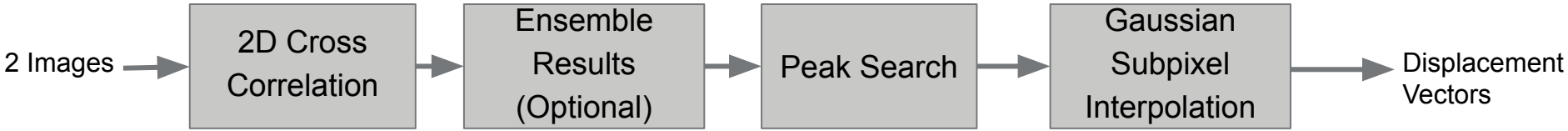
- User deploys battery powered sensor on bridge mount scenario
  - Energy budget for several configurations for long term, low measurement rate deployment
  - Several operating modes for type of measurements, measurement rate, etc.
- User deploys battery powered sensor on UAS scenario
  - Energy budget for several configurations for short term, high measurement rate deployment

System Architecture

- CommonSense Board
  - LoRa Daughter Board
- Camera
- Hardware requirements are light

The scope of our project is focusing solely on the PIV algorithm optimized for CommonSense.

Algorithm Architecture: Components of PIV



Ensemble - Nothing fancy. Just average of 2D cross correlation matrices for multiple pair of images (ensemble length 16 means 32 images in total, 2 taken in pair to produce a total of 16 2D Cross correlation matrix)

Ensembling **rejects noise**.

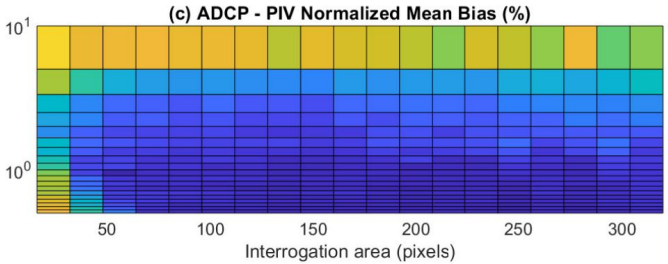
Carnegie Mellon University

Algorithm Architecture: Components of PIV

- 2D Cross Correlation
  - Outputs a 2D matrix of window correspondence values at different window shifts
- Peak Search
  - Distance from image center to highest peak gives velocity vector
- Ensemble Method
  - Filters noise of a single window area with cross-correlated windows
- Gaussian Subpixel Interpolation
  - Find velocity vectors between resolution of pixels

Evaluation: Accuracy (Interrogation Area)

- One tradeoff between direct and FFT cross-correlation is interrogation area constraints
- Minimum FFT size that has reasonable results is 64x64
- Larger window sizes (128x128) are preferred
- CommonSense is memory constrained



PIV Parameter optimization performed by Carl Legleiter and Paul Kinzel, showed that a 50px interrogation window (window side length) was the minimum size that gave reliable results

Carnegie Mellon University

Evaluation: Hardware Limitations affecting Accuracy

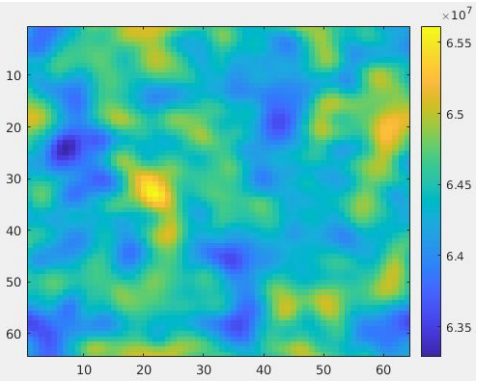
- Images stored as uint8_t in flash memory
- 2D FFTs do not have direct real-real transformation, a complex (float) intermediate representation is required
- Images stored as floats in ram
  - 2x 64x64 windows: 32kB
  - 2x 64x64 temporary storage for imaginary values: 32kB
  - Theoretical minimum: 64kB
  - Can store additional windows in RAM for quicker ensemble correlation
- 128px Interrogation Area memory usage
  - 4x 128x128 float arrays: 262kB minimum to perform 2D FFT
  - Will **NOT** fit in CommonSense's (ATSAMD51P20A) 256kB RAM without extra work

Evaluation: Hardware Limitations affecting Accuracy

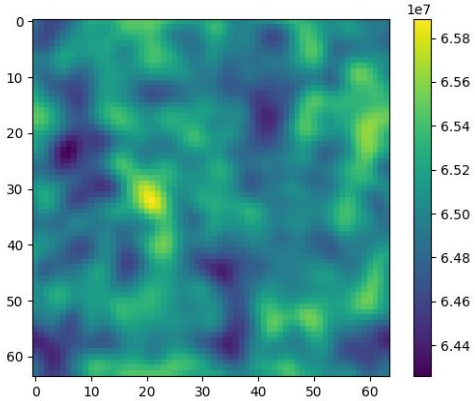
- 262kB for 2D FFT larger than CommonSense's 256kB RAM
- Solutions:
  - Use Q15 fixed-point floats or 16-bit half-precision floats
    - *Could affect accuracy of result*
  - Split Radix FFT
    - *More complex implementation, still need to store some of the results in flash memory*
  - Store parts of FFT result in offboard memory
    - *Most optimal solution if 128px Interrogation Areas desired*

Evaluation: Accuracy

- Our cross-correlation and windowing implementation matches that of PIVLab (Matlab)



PIVLab (Matlab) cross-correlation matrix for window 2610 between images 1 and 2 of the Alaska dataset

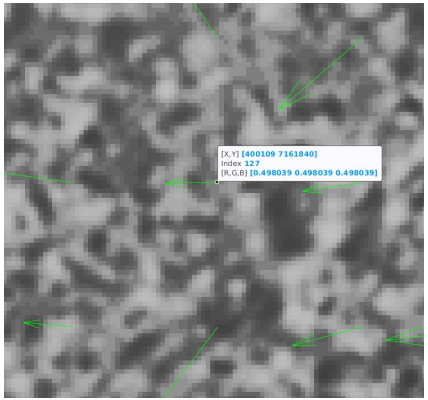


Our implementation of cross-correlation matrix for window 2610 between images 1 and 2 of the Alaska dataset

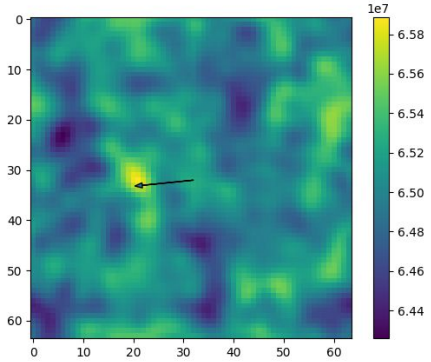
Carnegie Mellon University

Evaluation: Accuracy

- Our cross-correlation and windowing implementation matches that of PIVLab (Matlab)
- Our peak search implementation matches that of PIVLab
- The implementation of Gaussian Subpixel Interpolation varies very slightly within the subpixel region



PIVLab displacement vector plotted on the window from image 1 of the Alaska dataset
Magnitude 1.6834(m/s) Direction: (-1, -0.0108)

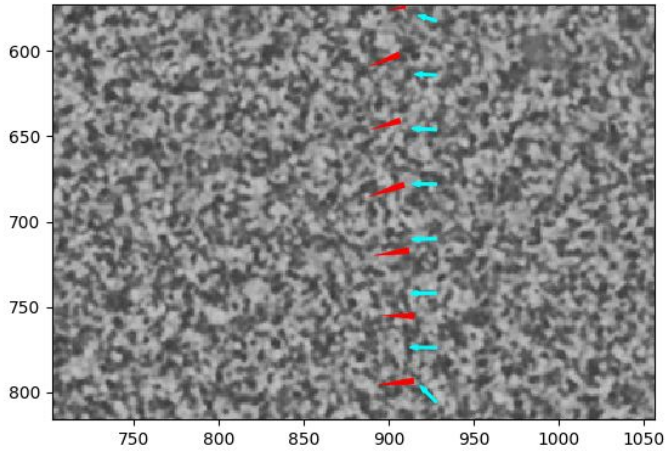


Our implementation of cross-correlation matrix for window 2610 between images 1 and 2 of the Alaska dataset
Magnitude: 10.273 (PIXELS/s = 1.54m/s) Direction: (-1, 0.1007)
Note the change in sign, the MatLab output is reoriented 180deg such that North is up, opposite of our implementation

Carnegie Mellon University

Evaluation: Acoustic Doppler Current Profiler (ADCP)

- Our algorithm was designed based off of PIVLab and Benjamin Pelc's Pivelocimetry
- Comparing the results from the Alaska Dataset against the ground truth measurements
  - Results are quite close



Comparing the accuracy of our embedded PIV implementation (cyan) against the ground truth values (red).

Carnegie Mellon University

Evaluation: Time Complexity

- PIVLab is able to compute PIV on a pair of images (4272 windows) in 1.23 seconds
  - Vectorized FFT approach allows the cross correlation for each window to be performed in parallel
- Our direct cross-correlation implementation followed the **O(N^4)** scaling and proved unfeasible for the application and hardware
  - 13min 30s for a single 64x64 pixels window
- Our FFT cross-correlation implementation is feasible for the realtime nature of the application
  - 0.2s for a single 64x64 window
  - Still not feasible to perform PIV across the entire image, not even including ensembling yet...

Time Complexity Optimization

- Application use case have several givens which can be leveraged:
  - For static mounting: areas of interesting flows usually will not move over time
  - The location of the sensor relative to the stream is known ahead of time
  - Surface flows are much smoother at the center of the channel compared to near the banks
  - Only surface flow rate near the center of the channel is needed to compute the discharge rate

A Hydrologic Technician manually measures streamflow at USGS streamgaging station 13063000 in Idaho.
Credit Marshall Williams USGS

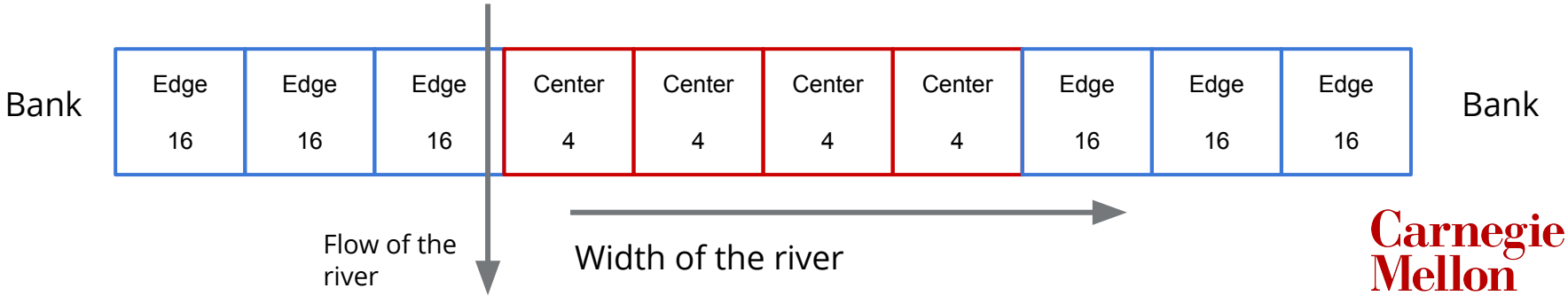
Time Complexity Optimization Approach

- Number of images in ensemble correlation is parameter for tradeoff between PIV accuracy and PIV speed
- Our approach:
  - Include more images in the ensemble correlation for windows close to the bank
  - Fewer images in the ensemble correlation for windows near the center of the channel
  - Perform PIV only for a small strip of the full image
  - Have an option for non-square PIV windows

PIV Area (Length and Breadth in pixels)		Total # Ensemble pairs	Total time consumption (in sec)
1580	64	224	44.94261333
1000	64	140	28.08913333
64	64	12	2.40764
1580	128	432	86.67504
1000	128	280	56.17826667
64	128	24	4.81528

Velocity Field Measurements: Graded PIV

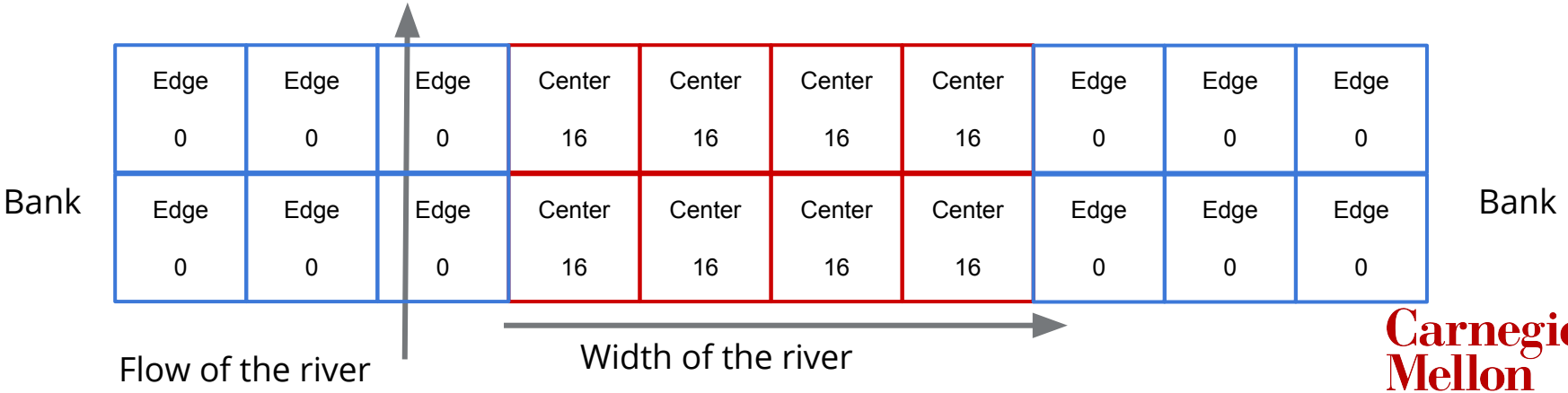
- Velocity field measurements require performing PIV across a wider area of the image
- Due to speed of CommonSense processor only feasible to get velocity measurement of a narrow cross section of stream at a time (a few windows wide)
- Taking advantage of the localized turbulent flows - Graded PIV approach:
  - More turbulent flows near banks, larger ensemble size (e.g. 16)
  - Smoother flow at center of stream, smaller ensemble size (e.g. 4)
  - Flows from bank to center should transition smoothly, apply a gaussian kernel
- A scaled down version of this is as shown below:



Carnegie Mellon University

Discharge Measurement PIV

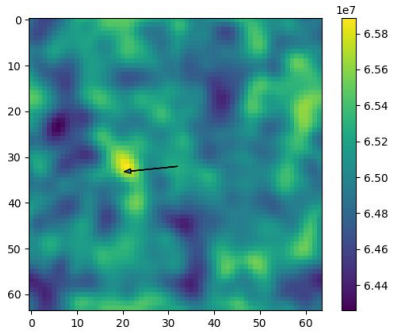
- For the discharge measurement, we only need to focus on the central squares
  - E.g. 256*256 pixels in the middle of the window of interest
  - Giving us in total 16 windows of 64*64
- Do obtain precise results, the ensemble size for *all* windows should be high
  - Experimentally determined 12-16
- Faster and more power efficient than velocity measurement across whole width



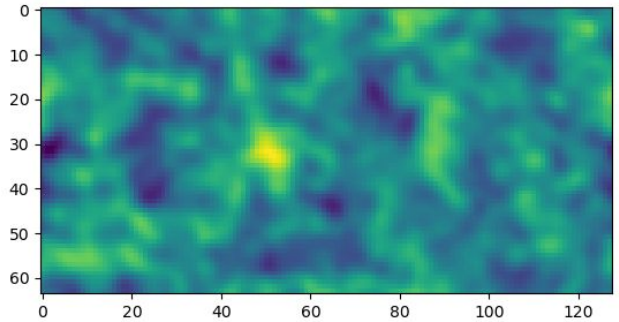
Carnegie Mellon University

Rectangular PIV Window

- Implemented, analysis in progress
- Stream will always* flow in one direction
  - “Tracked” particles will tend to flow in that direction
  - Yields better cross-correlation results
- Rectangular PIV allows for fewer computations to be performed:
- Increasing stride length thereby reducing number of windows required (FFT is still O(nlogn))



Square-Window Cross-correlation matrix for window 2610 for reference.

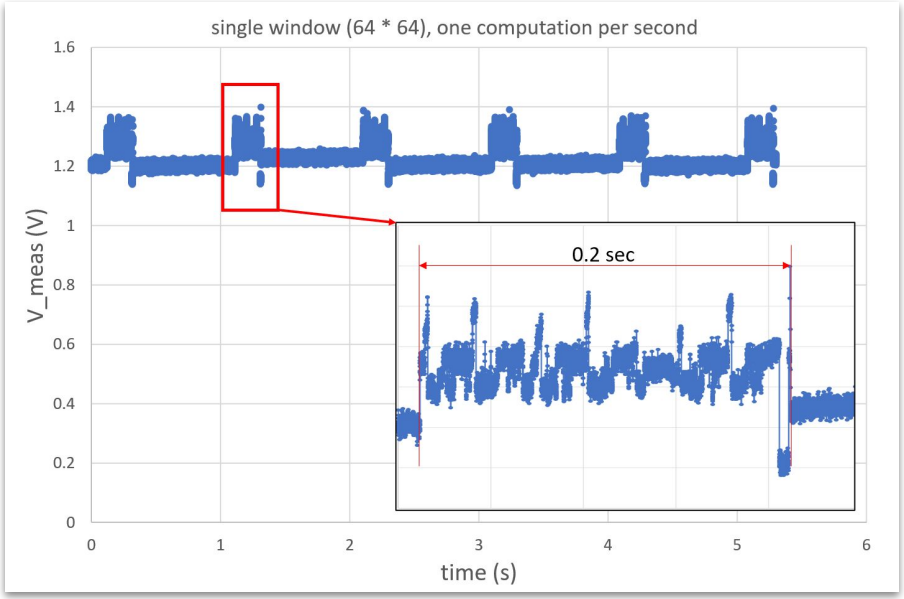


Rectangular cross correlation matrix centered at the position for square-window 2610.

Carnegie Mellon University

Evaluation: Energy for processing images

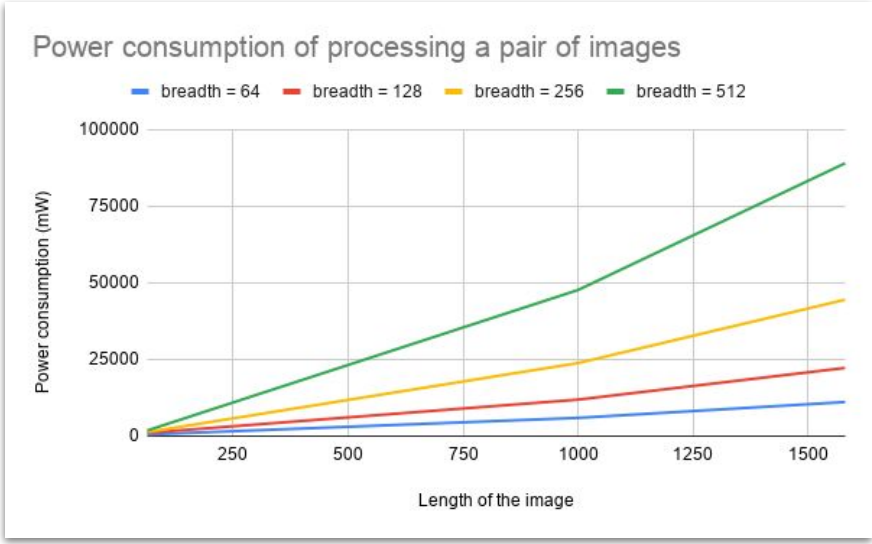
- 0.2 sec for single 64*64 window (one part of the image)
- V_meas_ave = 1.27 (V) => I = 50.8 (mA), V_load = 3.25 (V), P = 165 (mW)



Carnegie Mellon University

Evaluation: Energy Usage

- 0.532 years if we transmit the result 8 times / hour
- Result = 128 bytes (**vel_x, vel_y, start_x, start_y, peak_correlation_value, timestamp**)
- Have not included power consumption of camera

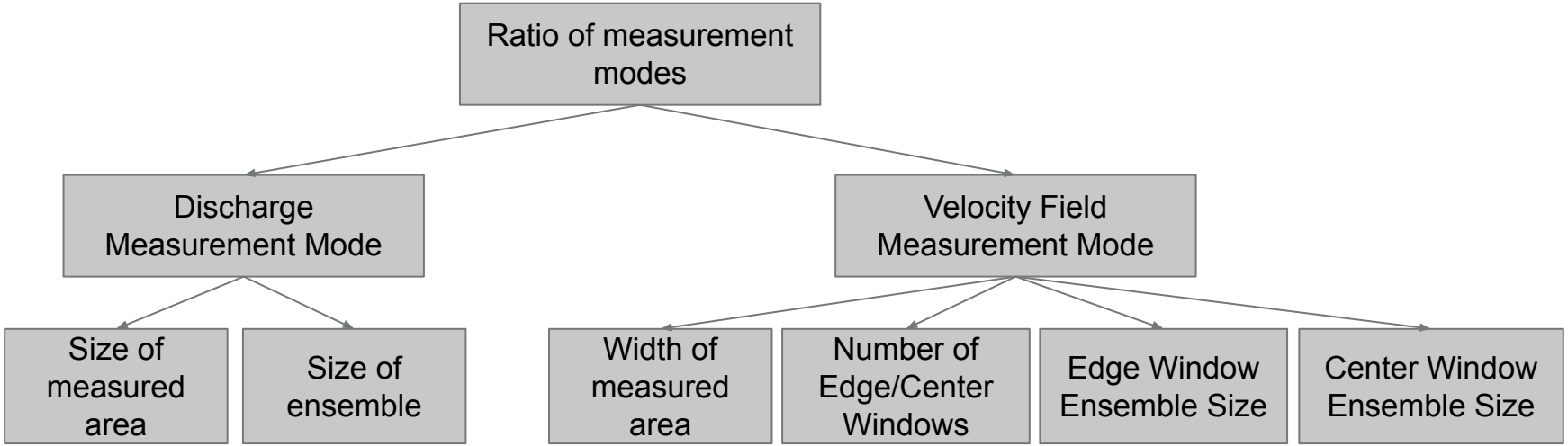


Power consumption if transmit 8 times / hour
(1580 * 64, battery = 12000mAh)

		val	% ON	mA
Processor	Computing (mA)	50.808	10.07%	5.1181642
	standby (uA)	18	0.00%	0
	hibernate(uA)	4	89.93%	0.00359
Radio	Receive Mode (mA)	10.8	0.01%	0.0012
Lora Module RFM95C	Transmit Mode (mA)	29	0.08%	0.0218080
	Sleep Mode (uA)	0.2	99.91%	0.0001998
			total	5.1449620
			years of life	0.5325066

Carnegie Mellon University

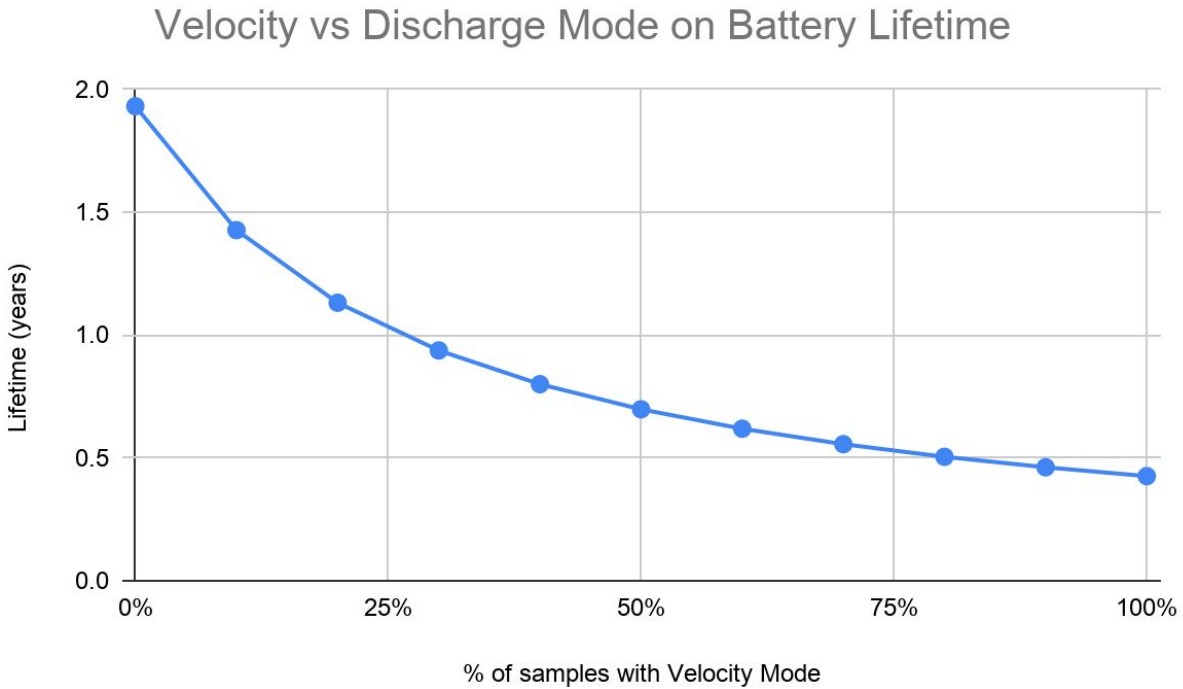
Optimization Parameters



Comparison of Modes Affecting Battery Life

10 samples an hour
(1580, 64) image size
Velocity Mode
Center Windows: 8
Center Ensembles: 4
Edge Windows: 8
Edge Ensembles: 12

Discharge Mode
Area: 256x256
Ensembles: 12



Carnegie Mellon University

Proposed Hardware Design

- Expanded RAM **OR** Low Power FPGA Daughterboard for CommonSense

Carnegie Mellon University

Future steps

- Design algorithm to be robust to additional scenarios
  - Focus on daytime, and optical images
- Stakeholder stretch goals
  - Continuous monitoring application
  - Image stabilization
  - Pre processing and tag geological info at the device
- Implement CommonSense features
  - QSPI Flash
  - Integrate with camera system & LoRa radio

Carnegie Mellon University

Lessons Learned

- Time and Space complexity are large considerations for IoT applications (Spatial->FFT)
  - 13 min -> 0.25 seconds, a 3000x speedup
  - 64x64 is the minimum for accuracy (and it shows)
- This is a potential scenario where the constraints make flash memory computation feasible
- Using hardware-accelerated computation saves time and energy

Acknowledgements

We would like to thank Carl Legleiter, Paul Kinzel, and Matt Marineau for this opportunity and their help.

We would also like to thank Bob, Eve, and Reese for their insight and support with CommonSense.

Carnegie Mellon University

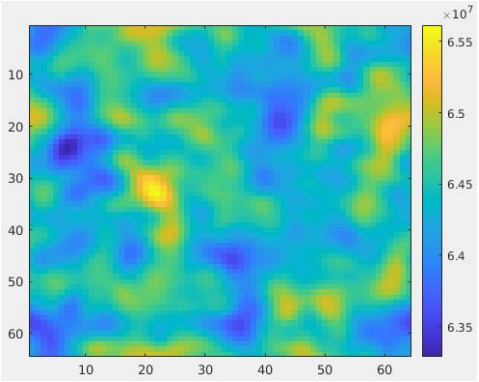
References

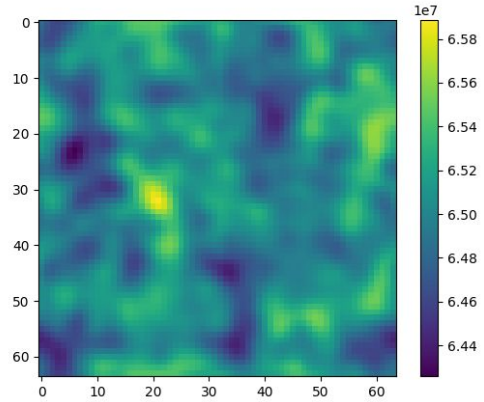
1. Kinzel, P.J. and Legleiter, C.J., 2019. sUAS-based remote sensing of river discharge using thermal particle image velocimetry and bathymetric lidar. Remote Sensing, 11(19), p.2317.
2. E. Delnoij, J. Westerweel, N.G. Deen, J.A.M. Kuipers, W.P.M. van Swaaij, Ensemble correlation PIV applied to bubble plumes rising in a bubble column, Chemical Engineering Science, Volume 54, Issue 21, 1999, Pages 5159-5171 https://doi.org/10.1016/S0009-2509(99)00233-X
3. Legleiter, C.J.; Kinzel, P.J. Inferring Surface Flow Velocities in Sediment-Laden Alaskan Rivers from Optical Image Sequences Acquired from a Helicopter. Remote Sens. 2020, 12, 1282.
4. Thielicke, W. and Stamhuis, E.J. (2014): PIVlab – Towards User-friendly, Affordable and Accurate Digital Particle Image Velocimetry in MATLAB. Journal of Open Research Software 2(1):e30, DOI: http://dx.doi.org/10.5334/jors.bl
5. Thielicke, W. (2014): The Flapping Flight of Birds - Analysis and Application. Phd thesis, Rijksuniversiteit Groningen. http://irs.ub.rug.nl/ppn/382783069
6. Santiago, J., Wereley, S., Meinhart, C. et al. A particle image velocimetry system for microfluidics. Experiments in Fluids 25, 316–319 (1998). https://doi.org/10.1007/s003480050235
7. Delnoij, E., Westerweel, J., Deen, N. G., Kuipers, J. A. M., & van Swaaij, W. P. M. (1999). Ensemble correlation PIV applied to bubble plumes rising in a bubble column. Chemical engineering science, 54(21), 5159-5171. https://doi.org/10.1016/S0009-2509(99)00233-X
8. *Particle Image Velocimetry*.(2015), benjaminpelc. 11/05/20. Web. https://github.com/benjaminpelc/pivelocimetry
9. Willert, Christian. (2008). Adaptive PIV processing based on ensemble correlation.
10. Dabiri, D. Digital particle image thermometry/velocimetry: a review. *Exp Fluids* **46,** 191–241 (2009). https://doi.org/10.1007/s00348-008-0590-5
11. Westerweel, J., Dabiri, D. & Gharib, M. The effect of a discrete window offset on the accuracy of cross-correlation analysis of digital PIV recordings. *Experiments in Fluids* **23,** 20–28 (1997). https://doi.org/10.1007/s003480050082

Carnegie Mellon University

Thank you

Any questions?





		val	% ON	mA
Processor	Computing (mA)	50.808	10.07%	5.1181642
	standby (uA)	18	0.00%	0
	hibernate(uA)	4	89.93%	0.00359
Radio	Receive Mode (mA)	10.8	0.01%	0.0012
Lora Module RFM95C	Transmit Mode (mA)	29	0.08%	0.0218080
	Sleep Mode (uA)	0.2	99.91%	0.0001998
			total	5.1449620
			years of life	0.5325066

Carnegie Mellon University

Extra slides

Direct v.s. FFT Cross-Correlation

- Direct approach advantages over FFT
  - Smaller space complexity (no extra buffers required)
  - Aliasing is not a problem

- FFT Aliasing
  - CommonSense is so space-constrained that zero-padding the image prior to FFT is not feasible
  - CommonSense version does not zero-pad but implementation seems to matche MatLab