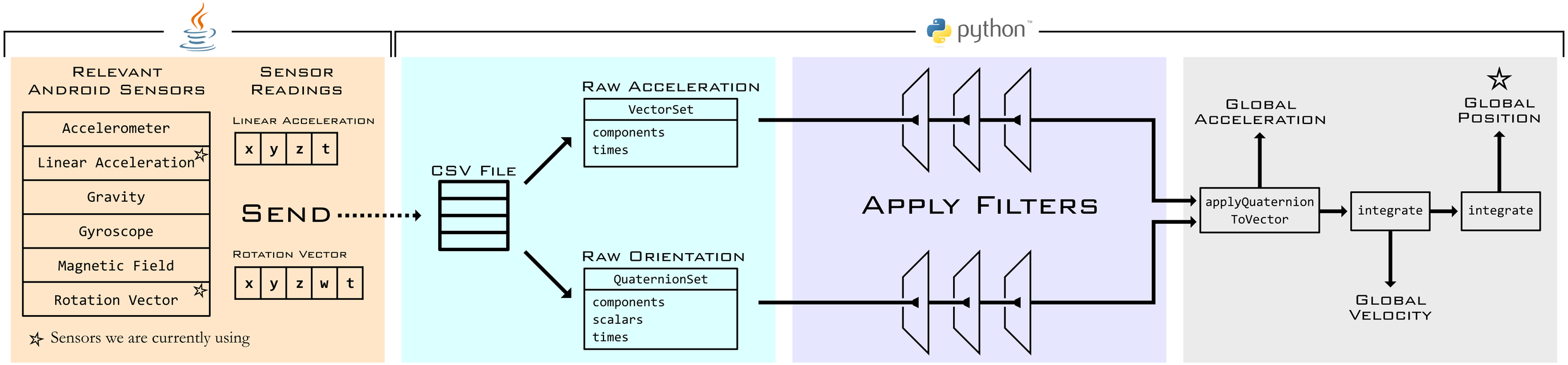
Inertial Navigation with Android

Lewis Collum, Mahesh K. Banavar

Clarkson University — defy convention™

COSINE LAB — COMMUNICATIONS SIGNAL PROCESSING NETWORKING



1. Sensing

Linear Acceleration:
Measures the acceleration force in m/s^2 that is applied to the device on all three physical axes (x, y, and z), *excluding the force of gravity*.

Rotation Vector:
Measures the orientation of the device, provided in *quaternion* form (x, y, z, and w).

2. Sending

Android app sends the sensor data as a CSV file, via email.

Room for Improvement (future):

Data will be sent real-time with Bluetooth.

OpenGL can then be used for visualizing real-time position of the phone.

3. Organizing

Data is sliced from the CSV file and stored in data structures, *VectorSet and QuaternionSet*.

Reusability:
These custom data structures inherit from NumPy's "ndarray".

This allows them to be seamlessly used with algorithms from packages including NumPy, SciPy, etc.

4. Filtering

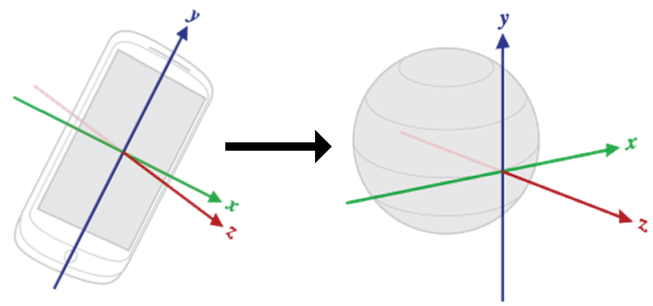
Moving Average:
Splits data into segments, then averages data within each segment. This also helps synchronize data.

Low-Pass:
Helps smooth spikes and noise.

High-Pass:
Helps reduce low-frequency effects, e.g, gravity.

5. Fusing

Global Acceleration:
Orientation must be applied to the local acceleration (along the phone's axes) to obtain global acceleration (along the world's axes).



☆ **Global Position (OBJECTIVE):**
Calculated by integrating global acceleration twice.

How Orientation Works

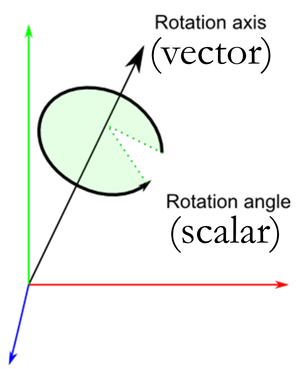
4D representation of a 3D rotation

Axis-Angle:

Visually intuitive, algebraically not.

Cannot apply the rotation to a vector.

Must convert to quaternion.



Quaternions:

Algebraically intuitive, visually not.

Easily interpolated.

Easy to apply to a vector (e.g. acceleration).

Check out our current projects at...

COSINE.CLARKSON.EDU