CS691: Introduction to Aerial Robotics

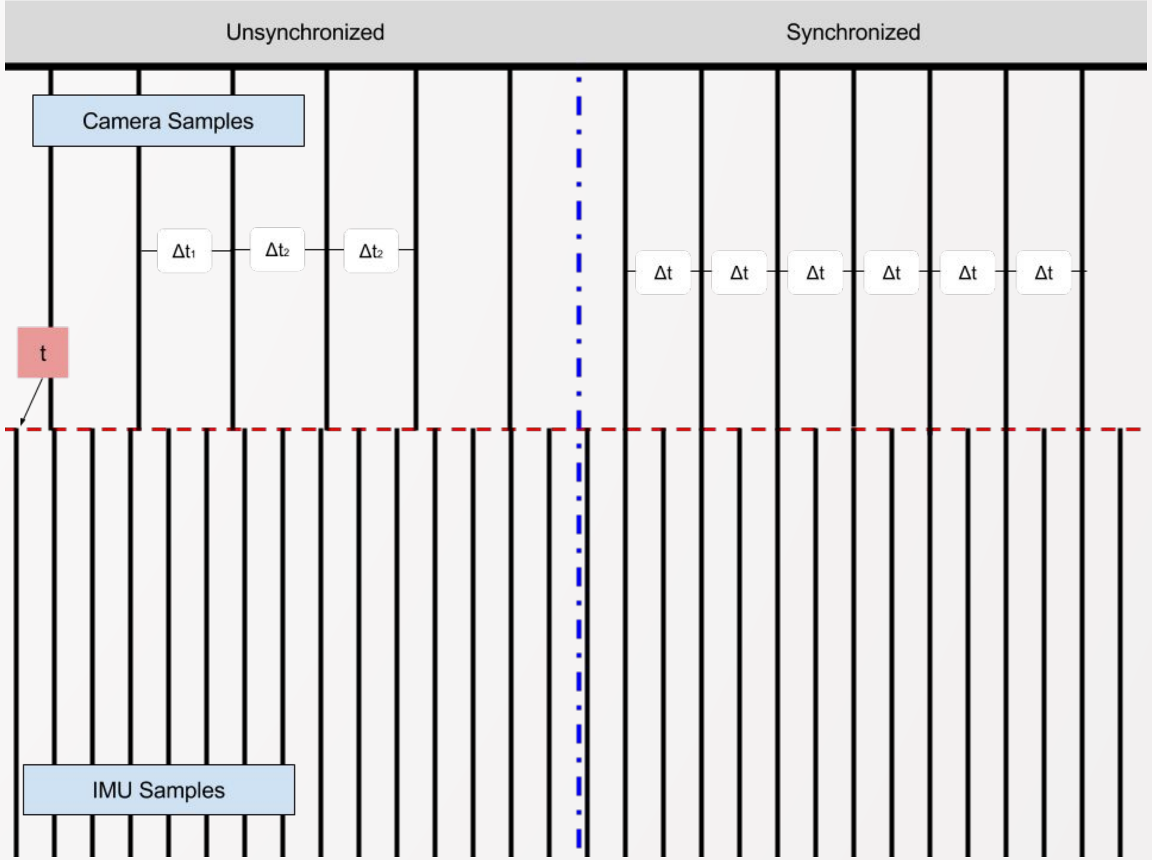
Hardware IMU & Camera Synchronization

Team: NSYNC

Motivation and Problem Description

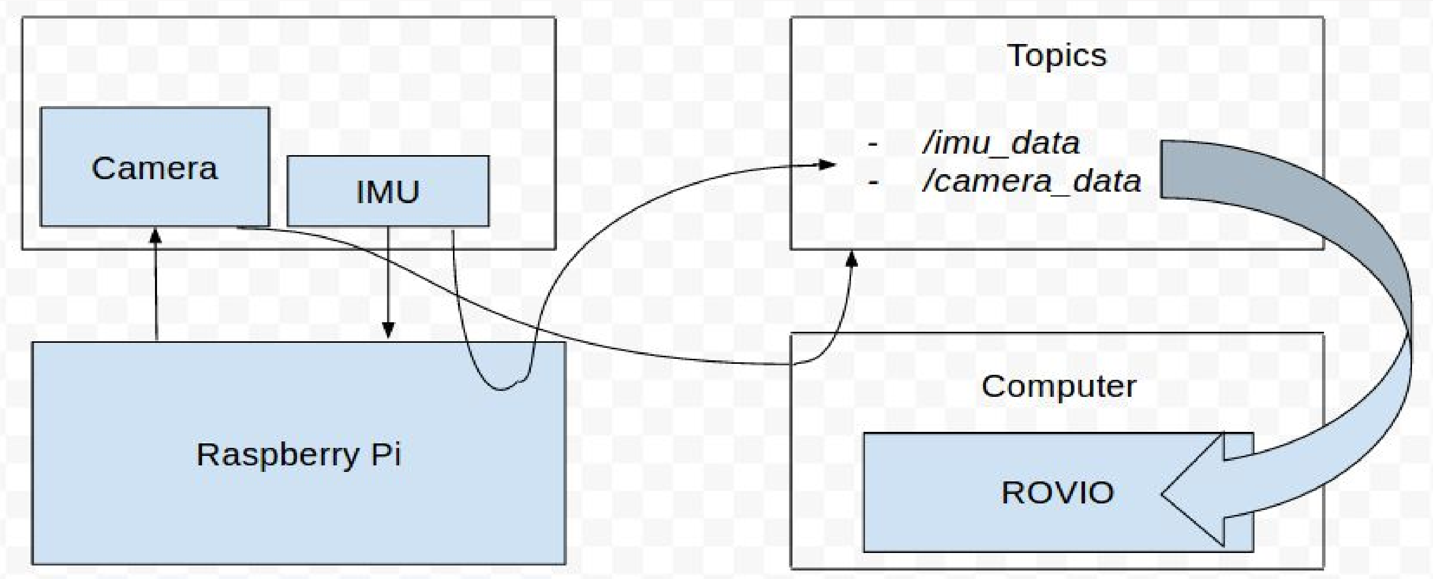
- Unsynchronized Camera and IMU data produces poor localization results.
- Provide a cost effective method for reliable camera & IMU synchronization.
- Currently, synchronized camera & IMU systems are very expensive.
- Our System:

PS eye:	$5.00
UM7 IMU:	$124.00
RaspberryPi:	$30.00



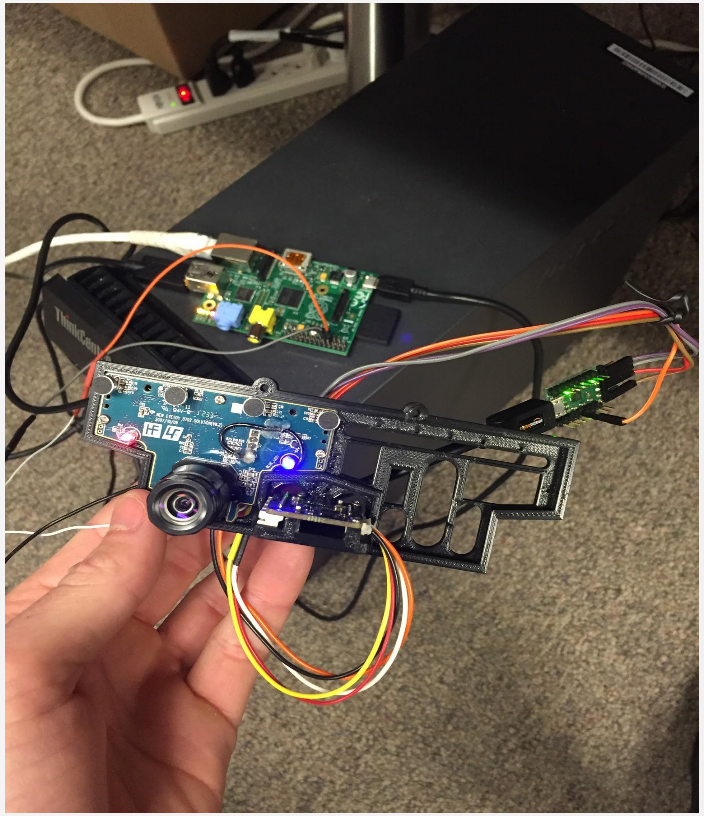
Proposed Approach

- Trigger the camera when IMU data received
  - Triggers camera at 60 Hz
  - Once every 4th IMU data reception.
- Store the time-stamp of the triggering IMU data
  - Store all subsequent IMU time-stamps until camera frame is received
- Calculate an estimated frame time-stamp
- Publish the image with the computed time stamp
  - Rovio uses more accurate frame time-stamp.



System Description

- Raspberry Pi, PS Eye, CHR UM7, ROS, ROS pkgs: um7 (modified), imucam_sync (built), rovio (ethz)
- Proper workflow:
  - UM7 and PS Eye capture frame and inertial data
  - Pi runs um7 driver and uses GPIO to trigger camera at 60 Hz
  - imucam_sync estimates “exact” time of frame data
  - publishes frames with adjusted headers
  - Rovio performs slam on frame and inertial data.



Results

- Successfully got the system communicating and running on a Raspberry Pi.
    - Receiving and publishing IMU data at approx 240 Hz
    - Triggering frame capturing at 60 Hz
- Calculate frame capture time estimate and publish image information
- Under construction:
    - Proper integration into rovio
    - Tuning rovio for optimal results