Sensor fusion for vision based localization: an overview

Dr. Sara susca
JPL, California Institute of Technology
Gainesville, Nov 15th, 2011

Outline

- Motivation
  - Why navigation?
  - GPS and inertial sensors fusion
  - Examples of GPS-denied navigation environments
- Vision
  - Building blocks
  - Features extraction
  - Feature matching
  - Visual odometry and ransac
  - Mapping algorithms
- Applications
  - LS3
  - Quadrotor
- Conclusions

Motivation

Navigation vs Localization

- In this talk they are interchangeable
- They both answer to a very primitive question: where am I?
- Who wants to know?
  - Commercial transportation systems (planes, ships, etc)
  - Personal transportation systems (cars, trucks, etc)
  - Robots (UGV, UAVs, …even Rumba!)
  - People (soldiers, first responders, parents obsessing over kids!, etc)
  - Assets

Inertial Sensors

- Inertial Measurement Unit
  - *Oldest sensor* used for navigation (compass is considered an aid)
  - Measures acceleration and attitude rate
  - Biased and noisy
  - High frequency (>500Hz), necessary for control of fast dynamics vehicles
  - Position errors grow cubically with time! The IMU has been often "aided"

	Size	Weight	Bias	Cost
tactical grade IMU	~3"x3"	~1kg	~1deg/hour	~10K$
IMU on chip	~penny	~10gr	~1000deg/hr	~500$

As size decrease, the biases increase → larger position error

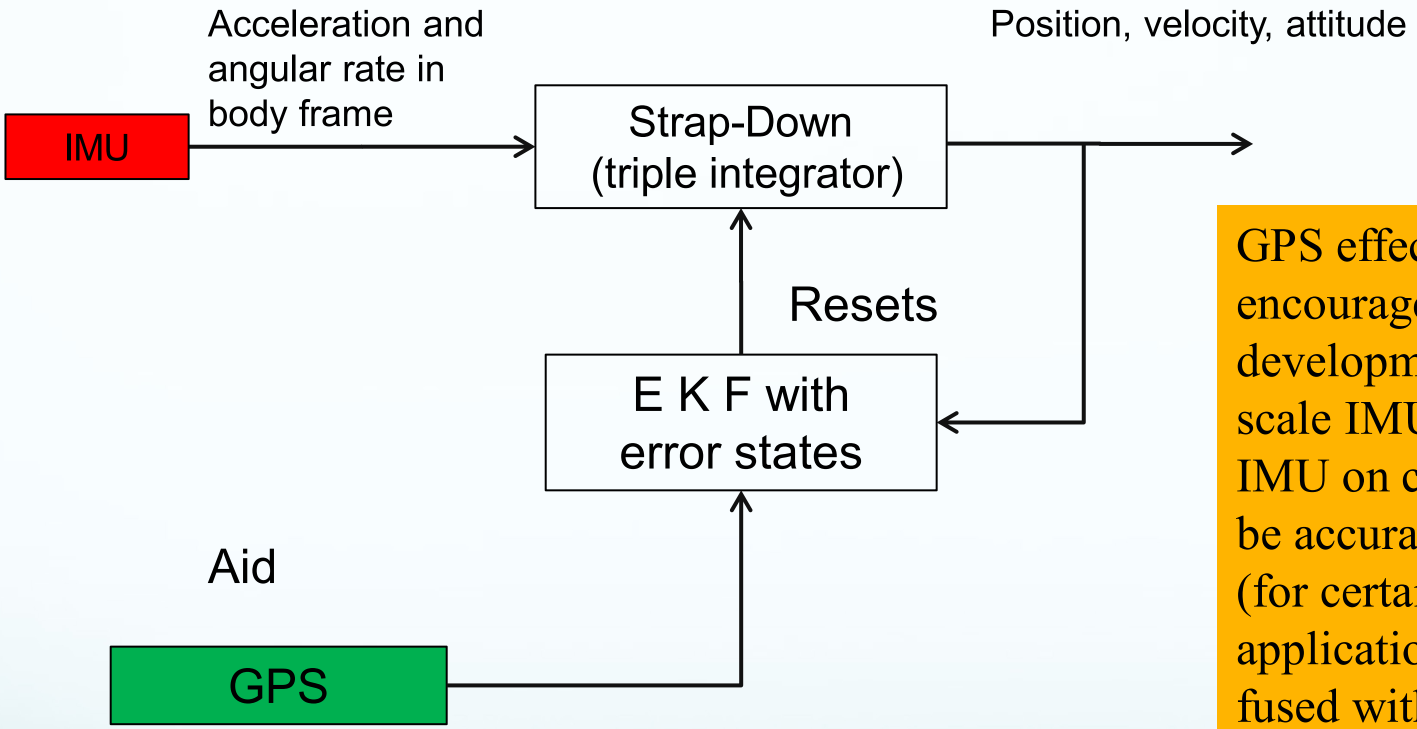
GPS: a Gift from the DOD

- Global Positioning System
  - Range measurement to satellite
  - Bounded error in absolute position
  - Frequency 4-1Hz

GPS and IMU are "complementary":
- GPS isn't biased and has low frequency information
- IMU is biased and has high frequency information

By fusing them we get best of both!

(Traditional) Sensor Fusion Filter

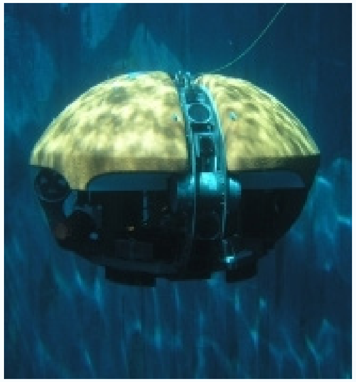


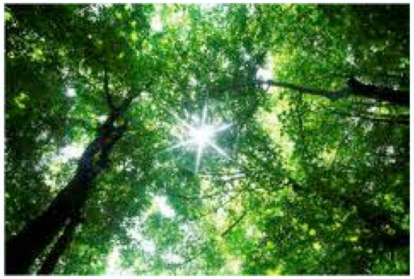
GPS effectively encouraged development of small scale IMUs. Even an IMU on chip can still be accurate enough (for certain applications) when fused with GPS!

Because of the causality and least-square nature of the filter, any erroneous measurement can irreparably compromise the filter integrity.

GPS-denied Environments

- We are spoiled!
- GPS-denied environments are not that uncommon
  - Underwater
  - Mines
  - Tree canopies
  - Urban canyon (I bet you know about this when dealing with your Garmin)





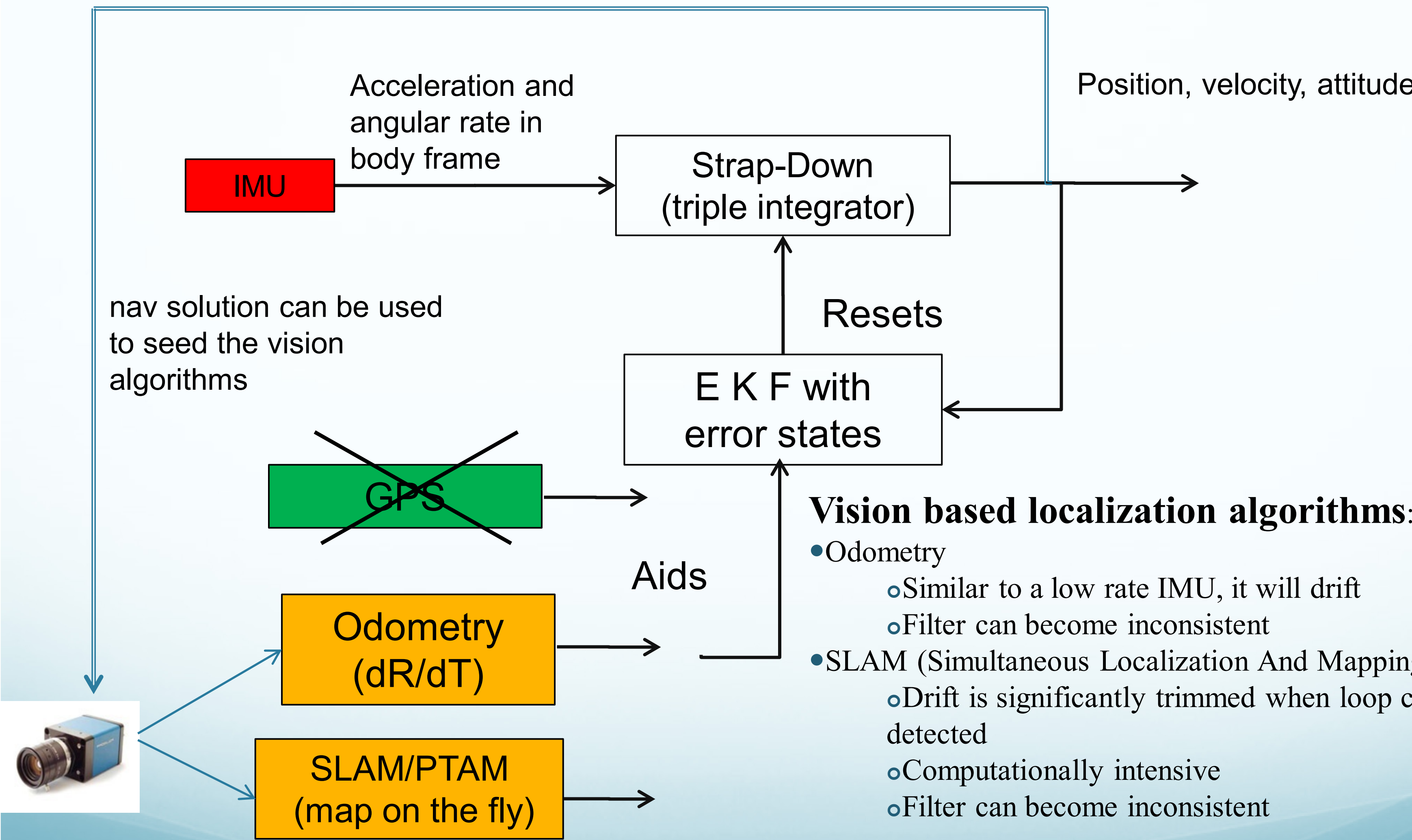
What to do?

Vision

Why Vision?

- It is a passive sensor
  - Low power, weight, and cost
- Data/cost ratio is high
  - Navigation, obstacle avoidance, recognition
- Image processing can be computationally expensive
- Some limitation on applications
  - night time operations, texture free environment

Fusion: an example



Vision based localization algorithms:

- Odometry
  - Similar to a low rate IMU, it will drift
  - Filter can become inconsistent
- SLAM (Simultaneous Localization And Mapping)
  - Drift is significantly trimmed when loop closure is detected
  - Computationally intensive
  - Filter can become inconsistent

Building Blocks

- From images we can
  - Build a map of the world and navigate in it (SLAM)
  - Compute the frame to frame motion (visual odometry)
- In both cases there are some common steps
  - Feature extraction
    - (at different scale, might involve the whole image)
  - Feature matching (Left to Right, T1 to T2)
  - False matches (outliers) rejection

- Challenge: outliers are very common and often there are more outliers than inliers.

Feature Extraction

- Point features in 2D (monocular and stereo)
  - SIFT/SURF, and Harris corners
    - SIFT and SURF features have a 128-elements vector as descriptors
    - Harris corners do not have a descriptor, they are less robust to changes in light, scale, and rotation but computational load is reduced
  - Whole image: template tracking

Find features that are unique and easily distinguishable is key

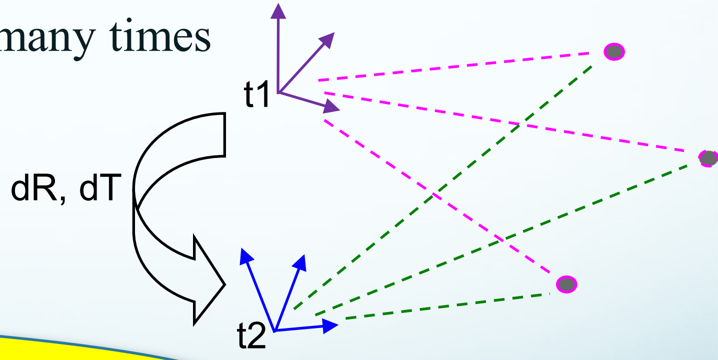
Feature Matching

- Points features
  - SIFT/SURF : All-to-All distance of descriptors is computed, the pairs with the smallest distance (unique enough) are considered matches. *False matches do occur*.
  - Harris corners:
    - KLT tracker, it involves solving a minimization problem and it is computationally expensive. *False matches do occur.*
    - SumOfDifferences, least computationally expensive, more prone to false matches
- Template tracking
  - Best approach for small images and feature-challenged environments

Matching is computational expensive
False matches do occur
Could be driven by IMU prediction

Visual Odometry and Outliers Rejection

- Motion can be approximated as an incremental rotation and translation
  - If the matched features are static, their new locations in the new image have to be coherent with the platform motion (dR and dT)
- Matched features can be used to compute dR and dT which is often argmin of a non convex minimization problem: how do we get a good initial condition?
- RanSaC: Random Sample Consensus
  - Hypothesis generation and testing
  - Solve for dR and dT using a small set of features, many times
  - Pick the one that agrees with most of the features



- Refinement
- VO is a measurement as well as an outlier rejection method

SLAM vs PTAM

- SLAM (Simultaneous Localization And Mapping)
  - Stand alone or aid to larger navigation filter
  - **(Kalman) Filter** at its core, solves for position of robots and of map (position of feature points) that it sees as it moves along
  - Map becomes large and not manageable, lots of ink spilled to improve its efficiency
  - Drift is eliminated as the robot returns to previously visited location: loop closure.
- PTAM (Parallel Tracking And Mapping)
  - Stand alone or aid to larger navigation filter
  - **Bundle adjustment method** at its core (non-convex optimization), position of features across multiple frames are the unknowns
  - Developed for monocular systems and has two threads
  - Mapping thread
    - solves for features location (map) using bundle adjustment
  - Filter thread
    - fuses gyros and features location provided by the mapping thread
    - provides 6DOF position updated between mapping threads cycles

Applications

One of each …

- Legged Squad Support System (LS3, next generation Big Dog)
  - Fusion of stereo vision odometry and IMU
- Quadrotor
  - Fusion of monocular PTAM and IMU

Legged Squad Support System (LS3)

Perception for High-Speed Legged Vehicles

NATIONAL ROBOTICS ENGINEERING CENTER NREC

JPL

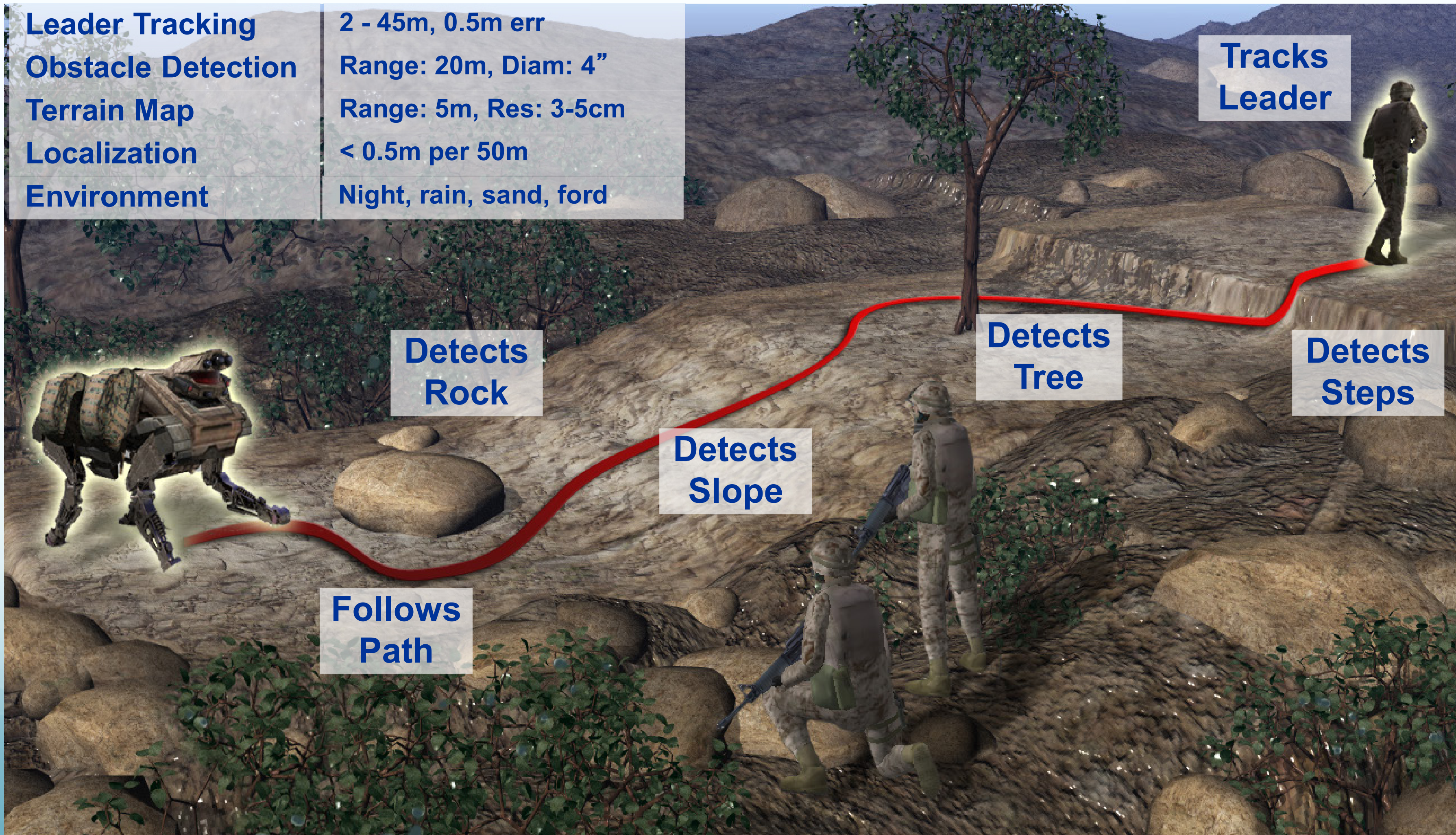
Boston Dynamics

LS3 Overview

- **Objective:**
  - A robot that goes anywhere a dismount soldier goes, carrying 400lbs, covering 20 miles, and lasting 24 hours
- **Team:**
  - Boston Dynamics, Bell Helicopter, Woodward HRT, AAI Corporation, CMU/NREC, JPL
- **Sponsor:**
  - DARPA (TTO), U.S. Marine Corp Warfighting Lab (MCWL)

Perception Objectives

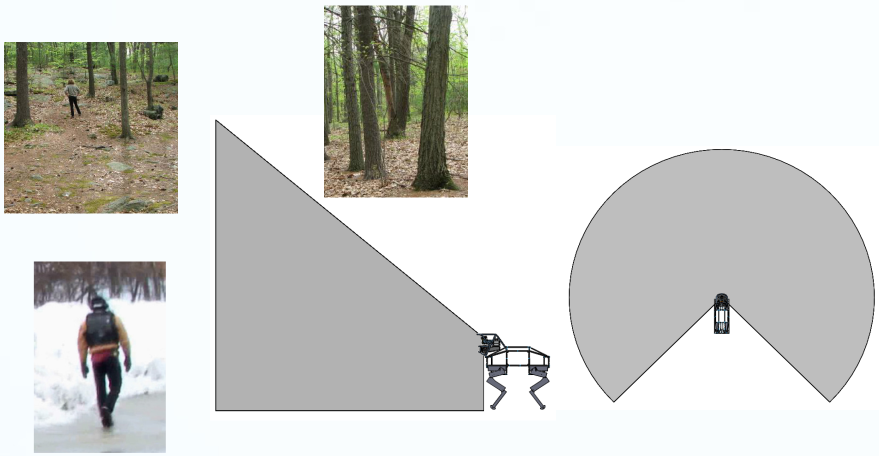
- Perception sensors are located in the head



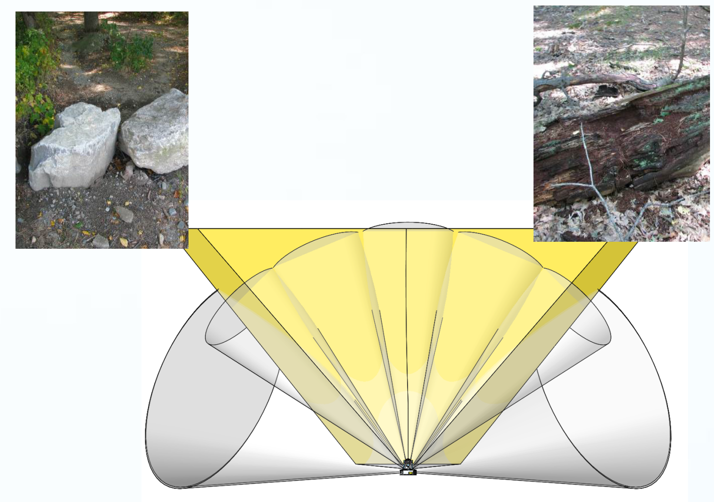
LS3 in action

- LS3 follows leader or navigates its own path.
- Prototype unifies all core perception, behavior, and robot control.

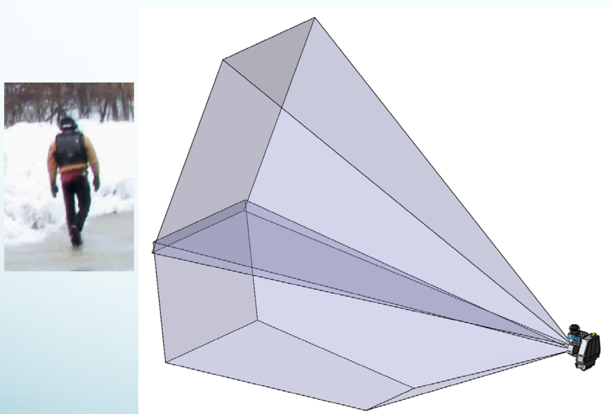
Visual Sensor Field of View



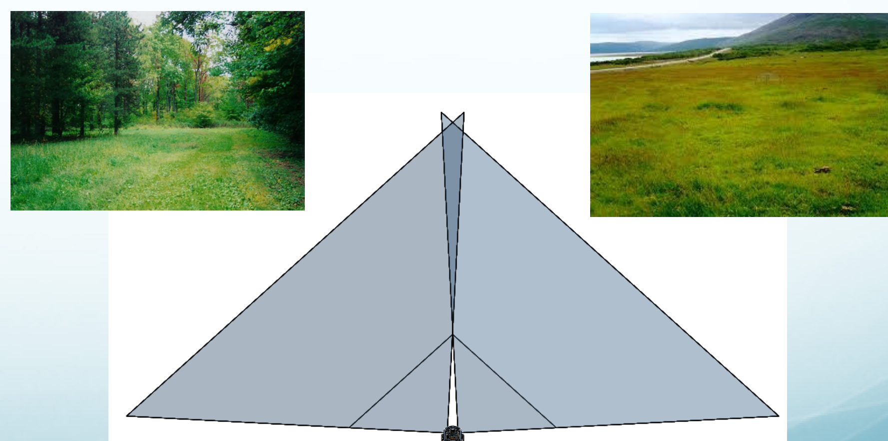
LIDAR: 5 meters shown, 50m range, 270˚ HFOV



Stereo Camera and Illuminators: 5 meters, 97˚ HFOV



NIR Cameras: 50m range
30˚ HFOV, 30˚ up, 14˚ down VFOV,
panned 200˚



Color Cameras: 5 meters shown, 180˚ HFOV

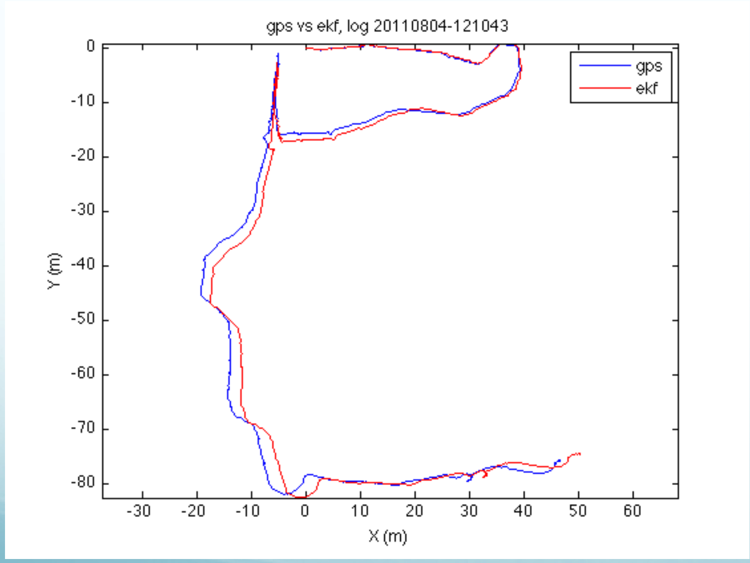
Localization Filter Overview

- Asynchronous EKF filter
  - Error states: position, velocity, attitude, and IMU biases
  - Prediction step: IMU (tactical grade) integration at 600Hz
  - Correction step: triggered when a measurement is available
- Measurements (autonomously triggered)
  - ZUPT: when stationary to prevent attitude drift
  - Visual odometry in nominal conditions, leg odometry when Visual odometry fails
- "Gates"
  - Hard filter reset when engine is off
  - Residual tests on visual odometry and leg odometry
  - Fall detection to further shield the filter from erroneous leg odometry measurements

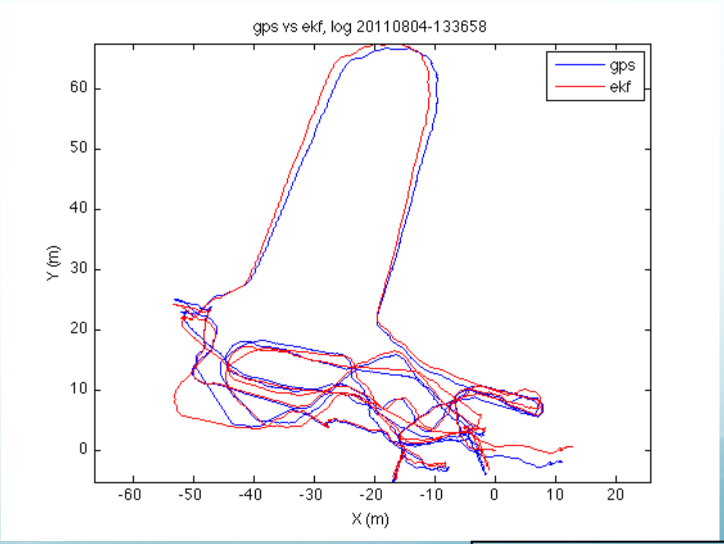
Localization Performance

- Ground truth provided by differential GPS or surveyed markers

Description	Run time	Distance	x-y % error*	z max error
(a) Asphalt/grass	350s	150m	1%	0.2m
(b) Asphalt/grass	1200s	300m	1%	4m
(c) Asphalt/grass with falls	1700s	800m	0.5%	1m
(d) Asphalt	350s	150m	<1%	0.1m
(e) Asphalt	270s	100m	1%	0.2m
(f) Asphalt	650s	400m	<1%	0.5m



(b)



(c)

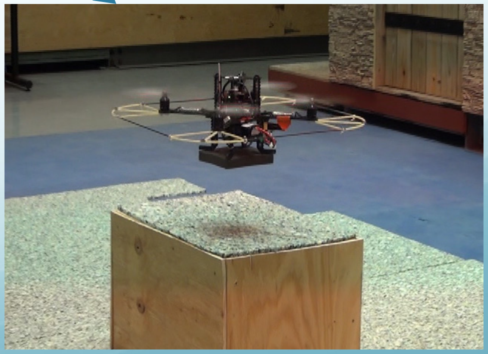
* = percent of distance traveled at steady state

Micro Autonomous Systems & Technology

MAST Overview

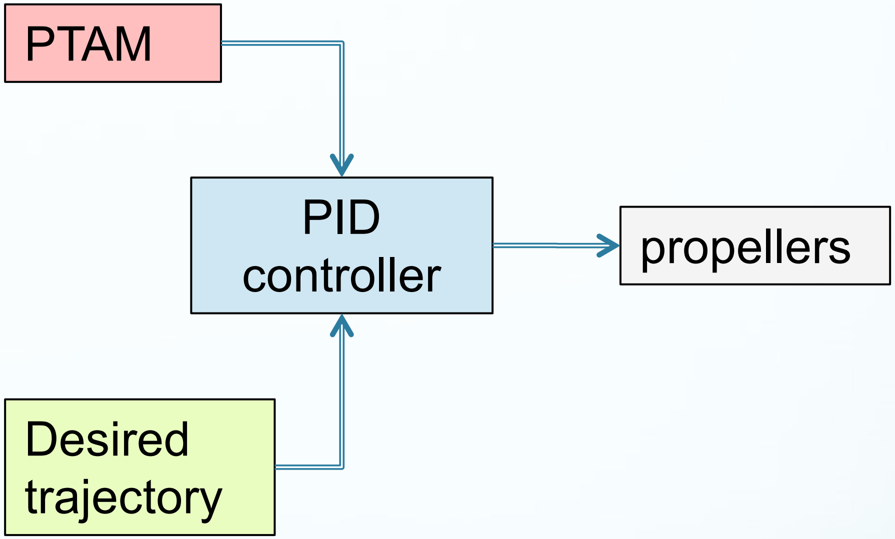
MAST: Micro Autonomous Systems & Technology

- **Objective:**
  - To develop autonomous, multifunctional, collaborative ensembles of agile, mobile microsystems to enhance tactical situational awareness in urban and complex terrain for small unit operations.
- **Team:**
  - BAE Systems, CalTech, GTech, JPL , Harvard University, MIT , North Carolina Agricultural & Technical State University , UC Berkeley , U of Maryland , U of Michigan, U of New Mexico , U of Pennsylvania
- **Sponsor:**
  - ARL (Air force Research Laboratory)

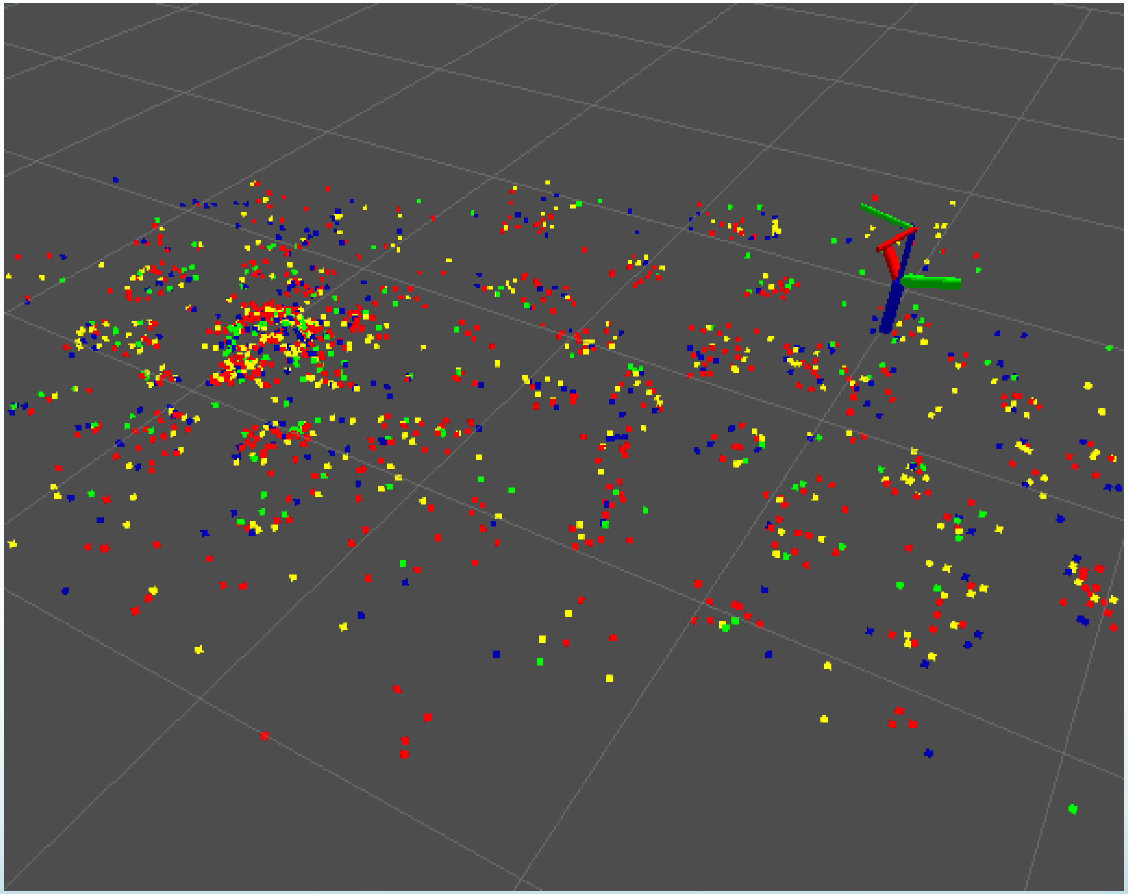


Autonomous Landing with PTAM

- Fast dynamics vehicle
- Exploration, detection of landing platform and landing, all fully autonomous, **running on-board in real time**
- PTAM locates the vehicle within a pre-built local map (~10Hz)
- Autonomous navigation uses 6DOF PTAM pose as input to the PID to control the vehicle position



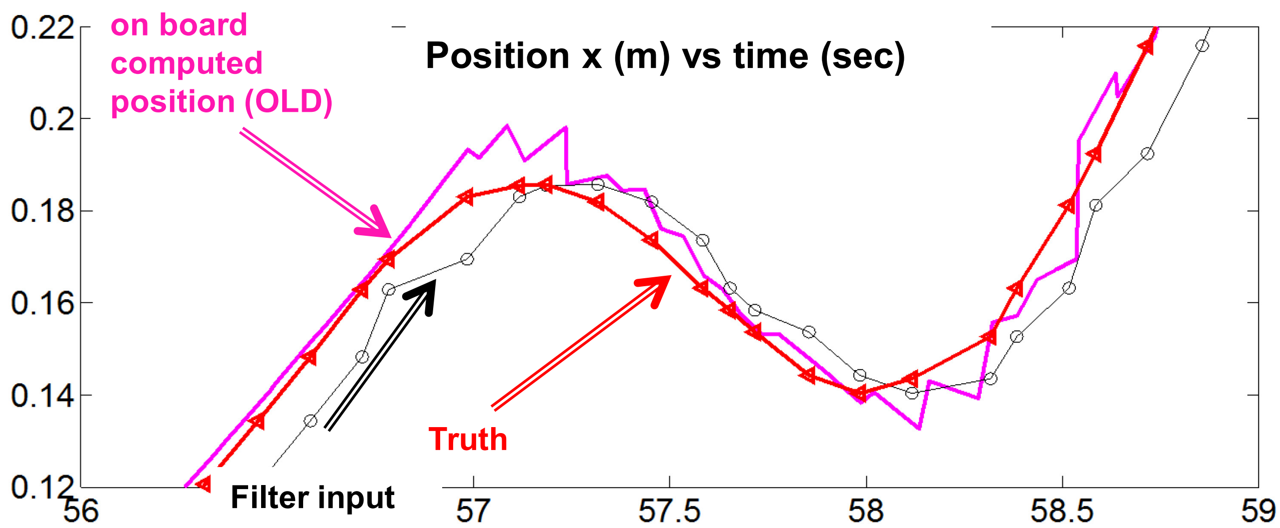
Results



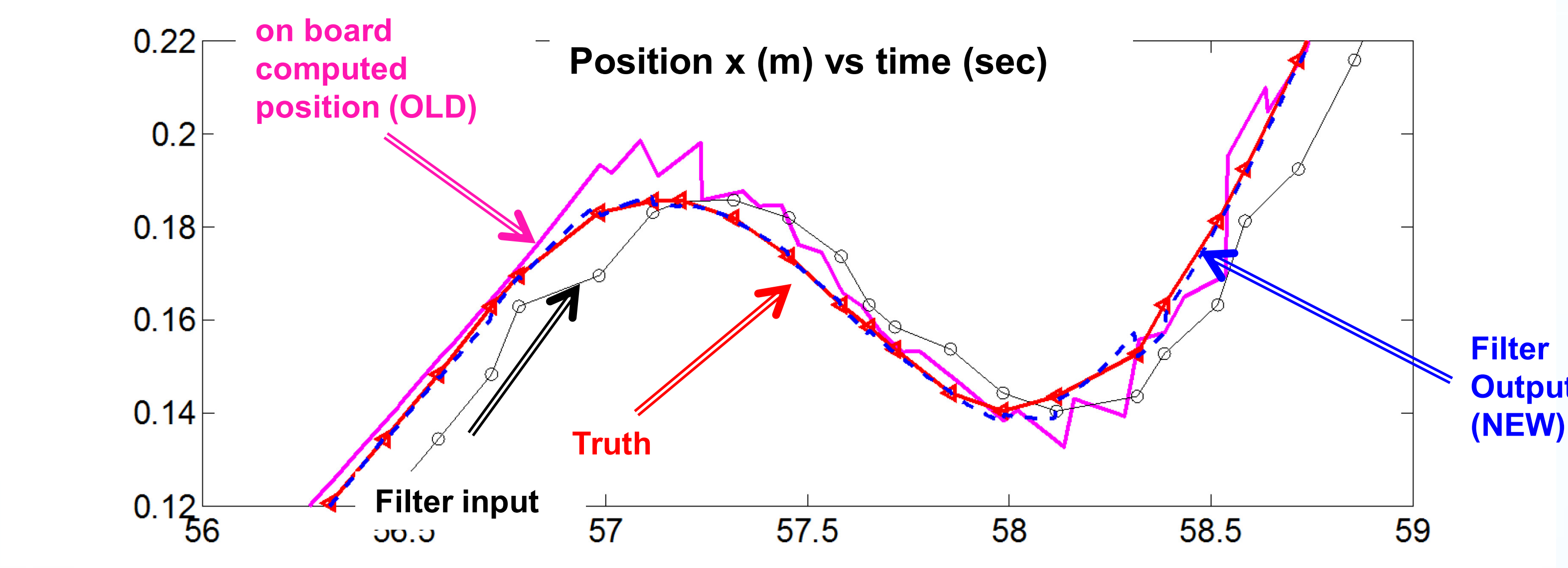
A Sloppy Controller?

- Symptom:
  - the controller seems “sloppy” even though PID gains were tuned (… a lot!)
  - ~15cm amplitude oscillation around target point
- Cause:
  - PTAM is computationally expensive and is available with ~0.1sec of latency
- Solution:
  - Fusion of on board inertial sensor and PTAM!

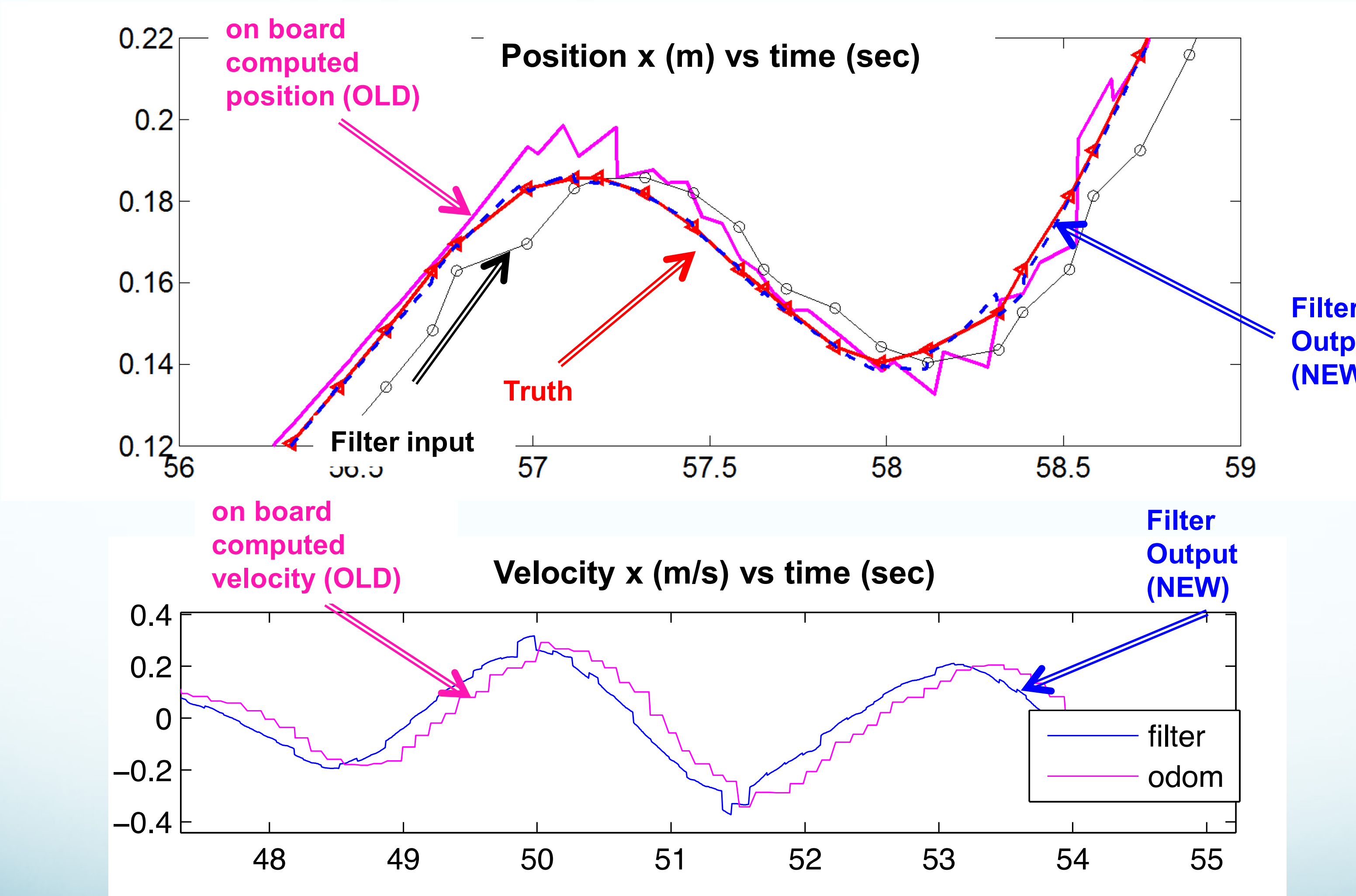
Sensor Fusion Filter Performance



Sensor Fusion Filter Performance



Sensor Fusion Filter Performance



The delay is eliminated!

On board implementation is in progress

Conclusions

- Navigation has a wide range of applications
- GPS and inertial sensor complementarity has been exploited in the the fusion algorithms
- Vision offers an appealing aid to inertial sensors in GPS-denied environments
- Applications
  - LS3: vision and IMU achieve state of the art performance in visual odometry
  - Quadrotor: IMU mitigates latency of vision