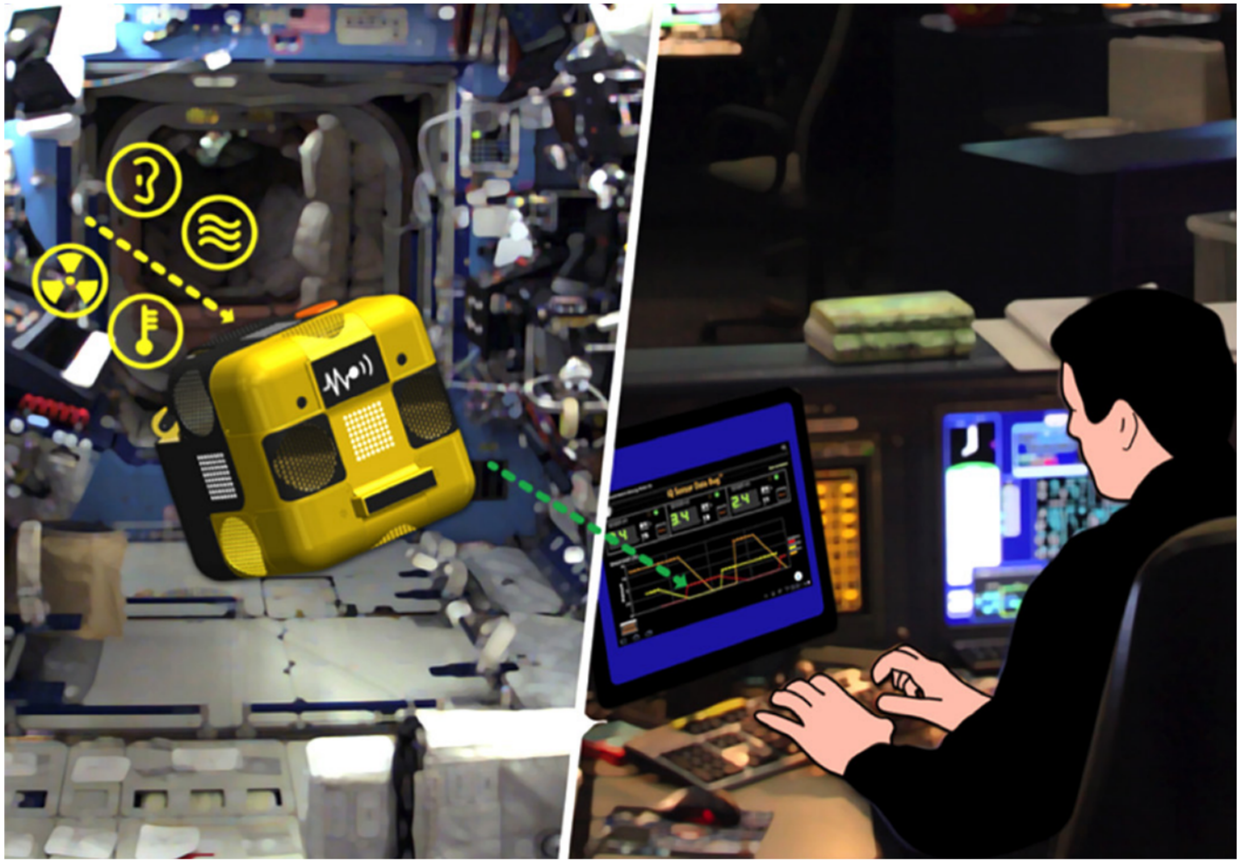
Astrobee Localization

Brian Coltin[1], Jesse Fusco[2], Zack Moratto[1], Oleg Alexandrov[1], and Robert Nakamura[2]

[1] SGT Inc. [2] NASA Ames

IROS 2016

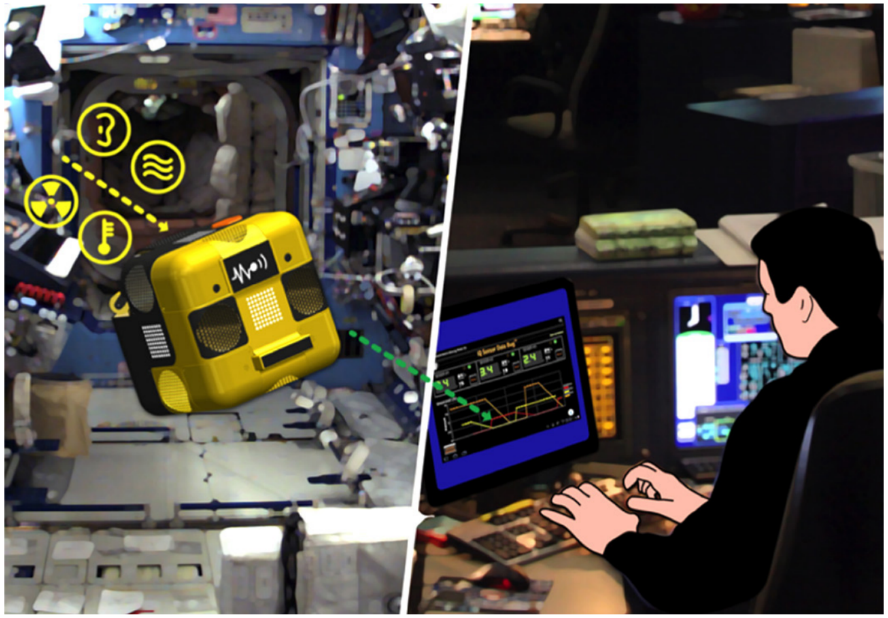


Spheres Robots

- Currently one of the most popular payloads on ISS
- Used to conduct experiments in microgravity
- Use gas thrusters and rechargeable batteries, must be upmassed
- Limited to 2m cube due to localization by triangulation from fixed ultrasonic beacons

Astrobee

- Replacement for Spheres
  - Movement by electric fans (no compressed gas)
  - Rechargeable batteries
  - Greatly increased computation
  - **Localize anywhere with no additional infrastructure**
- Set for completion by end of 2017

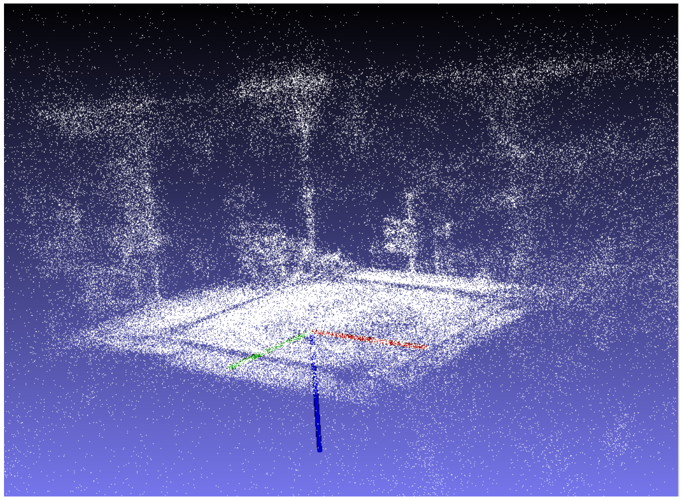


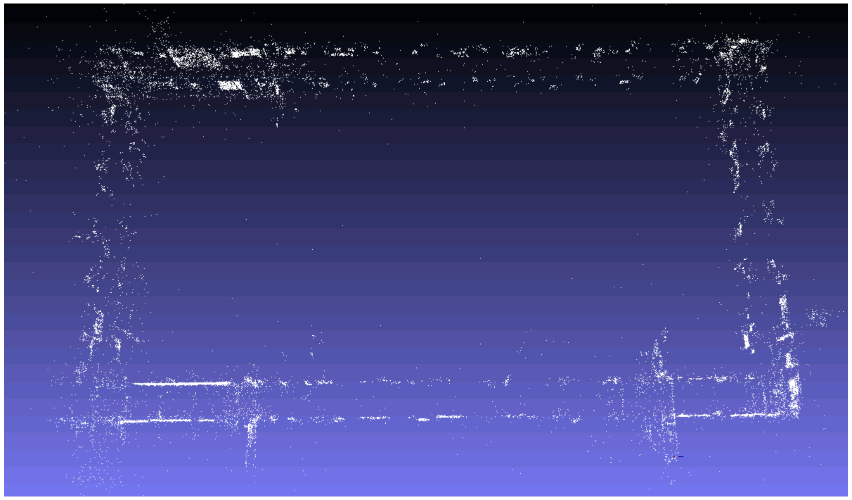
Overview

- Mapping
- Visual Features for Localization
- Position Tracking
- Results

Offline Sparse Mapping

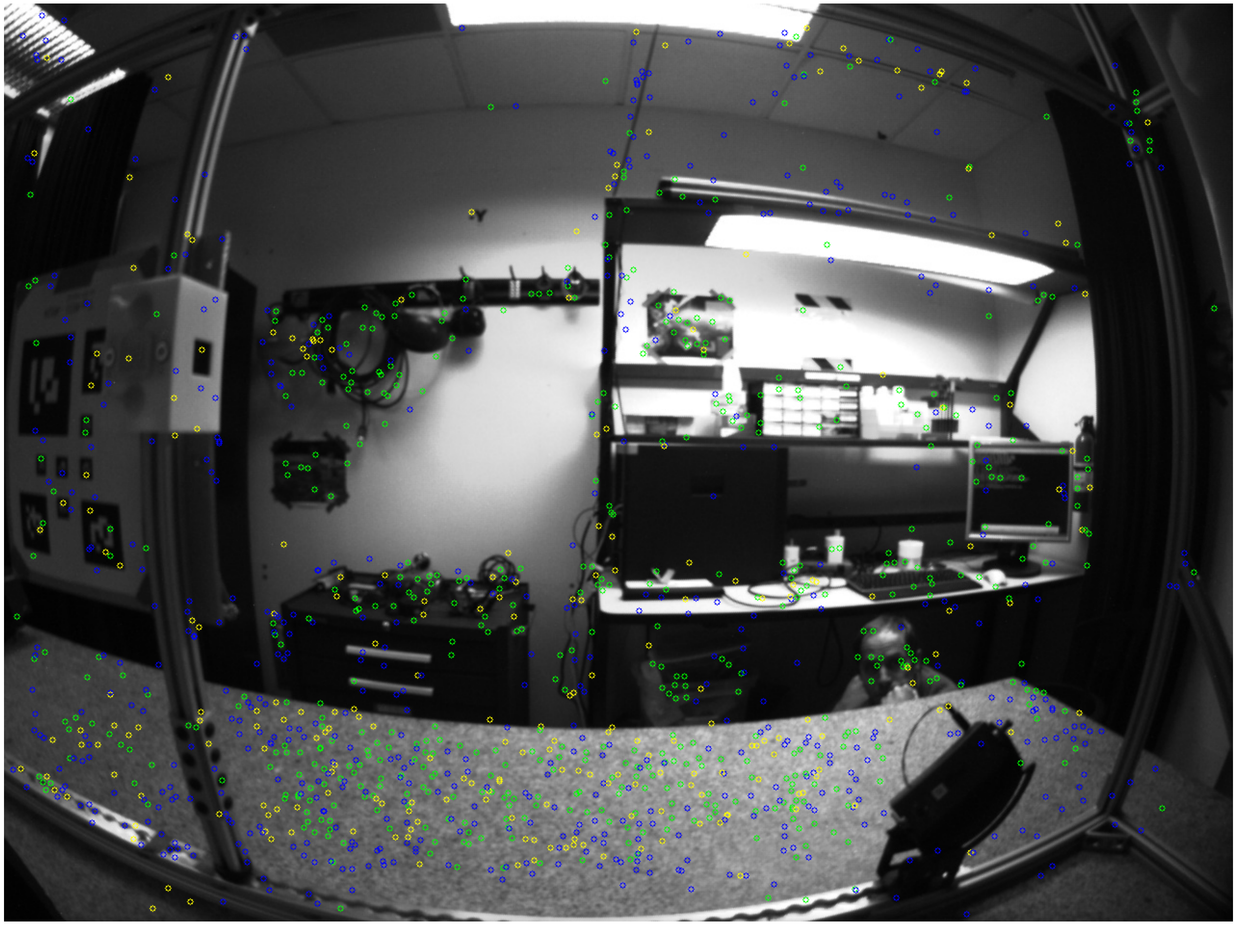
- We build a map offline from images to localize the robot against using interest points
- The map used for localization consists of:
  - Interest point descriptors
  - Interest point 3D location estimates





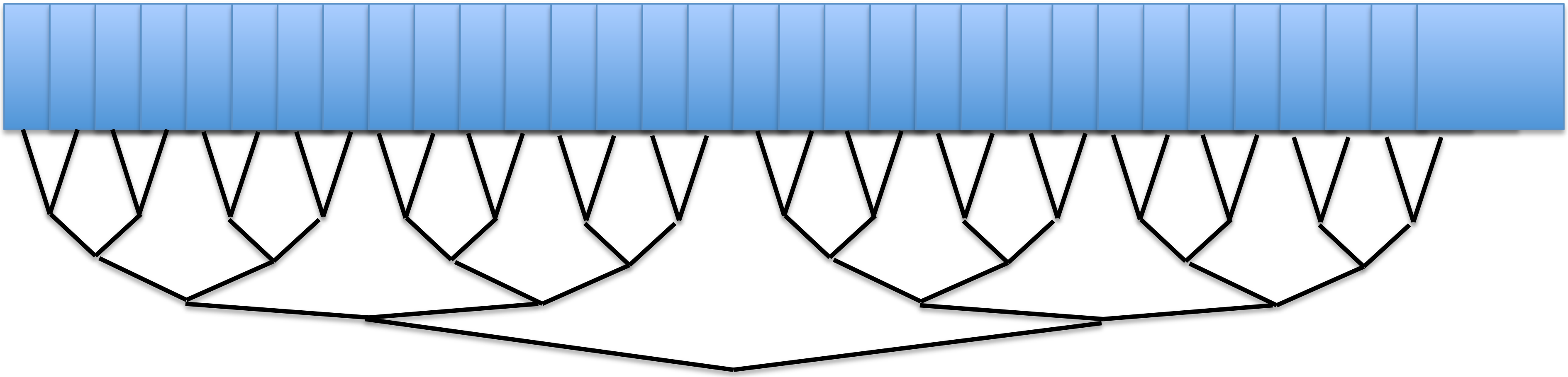
Map Building Steps

1. **Collect images** from different perspectives.
2. **Detect SURF features** on all images.
3. **Match features** between image pairs.



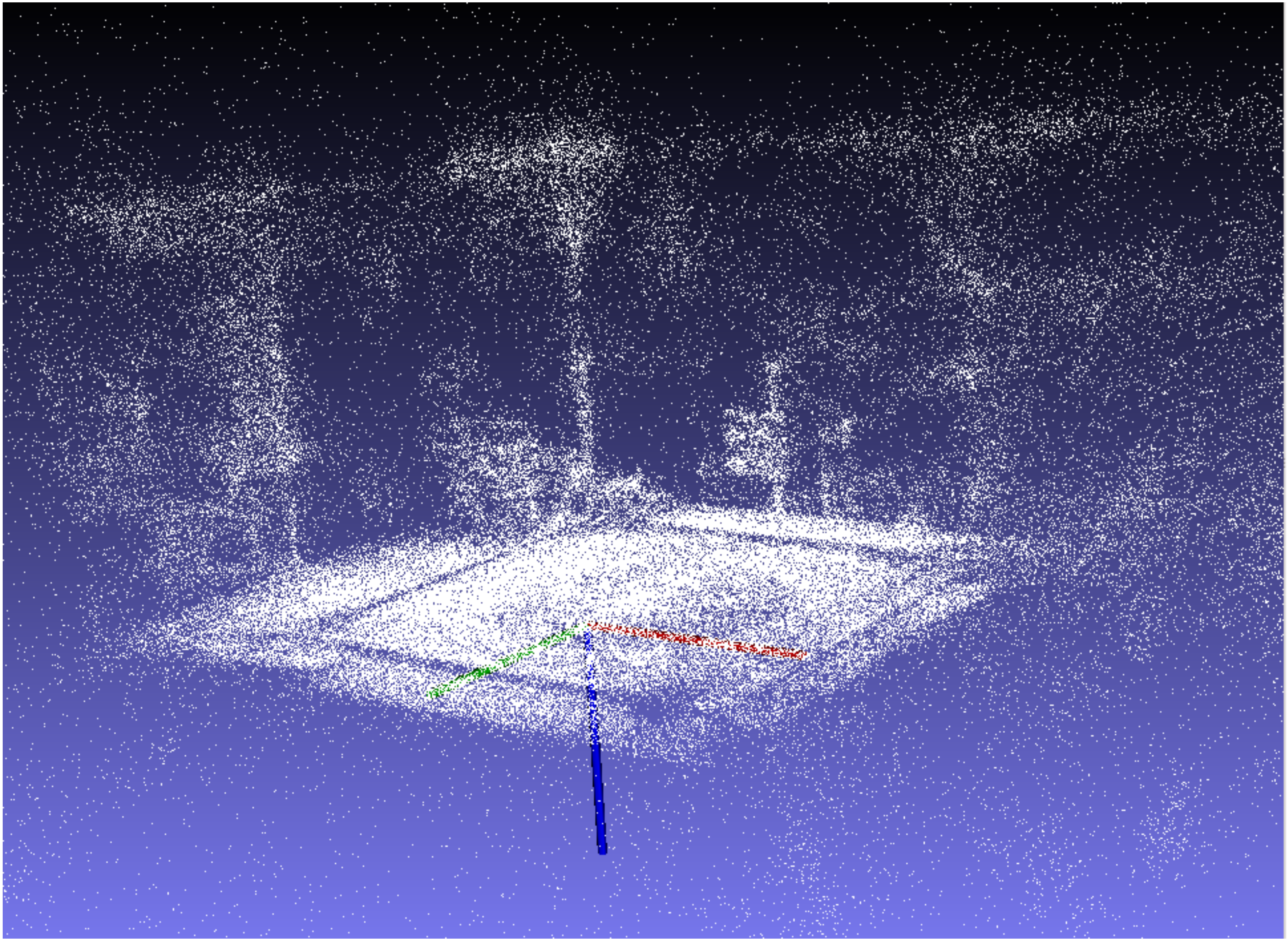
Map Building Steps

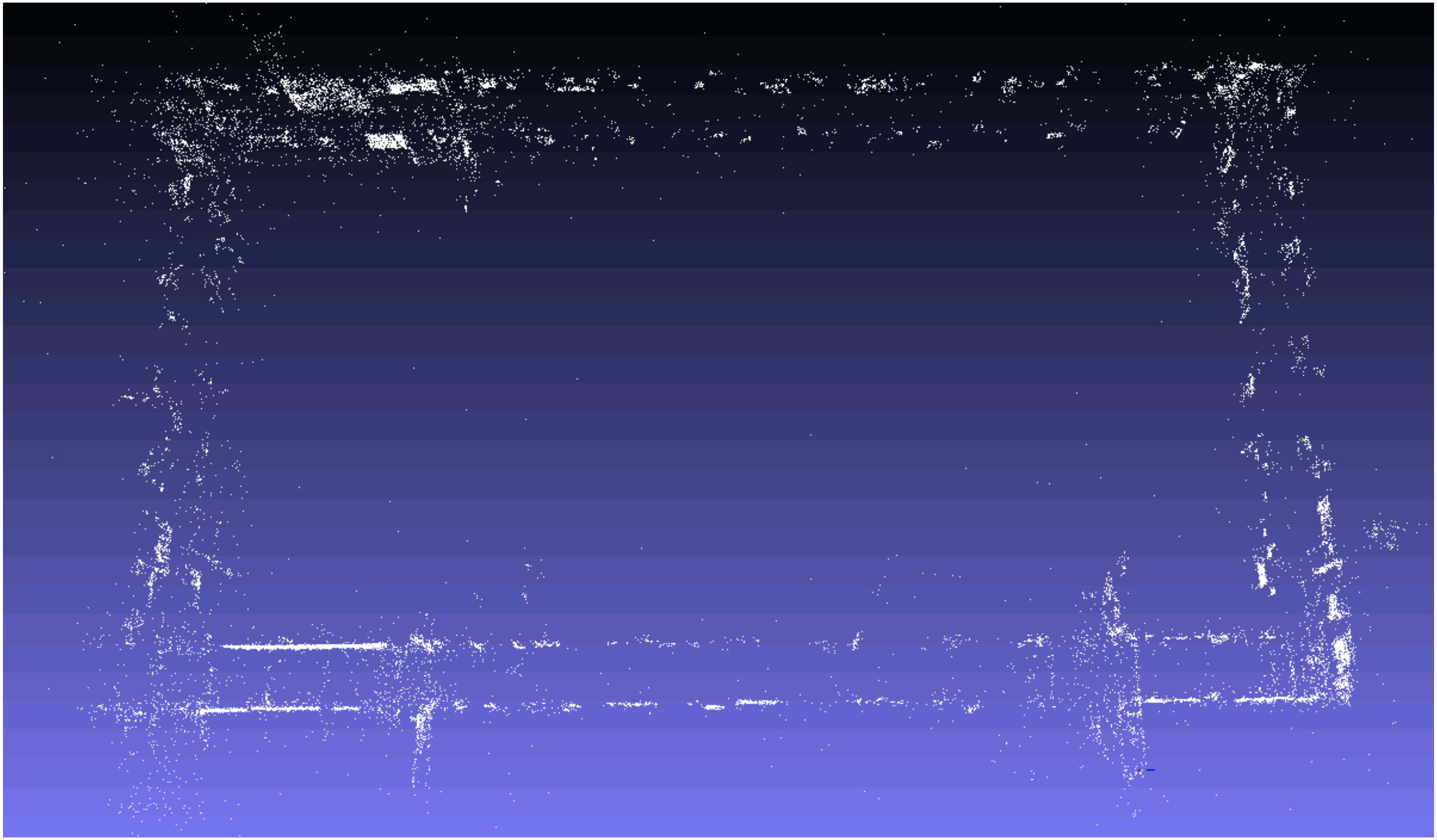
4. **Form tracks** of features that are seen in multiple images, triangulate initial camera pose and interest point location estimates.
5. Simultaneously solve for 3D positions of tracked features and camera poses with **incremental bundle adjustment** on images in a tree structure.



Map Building Steps

6. Adjust map with **global bundle adjustment**, remove outlier interest points.
7. **Rebuild** map with known camera poses and faster BRISK features.
8. **Register** to fixed coordinate frame by manually selecting known locations on images.
9. Build **bag of words database** to quickly search for similar images.

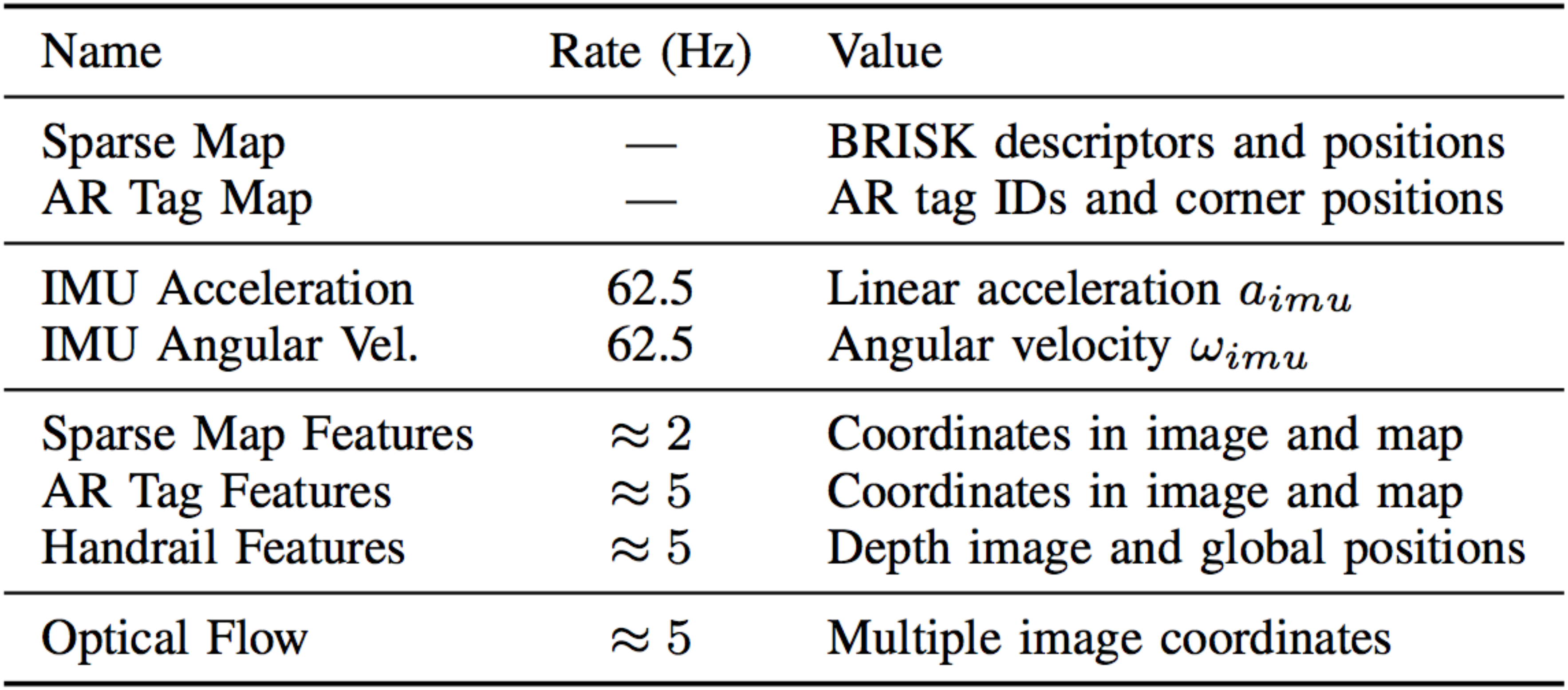




Visual Features for Localization

- Sparse Map Features:
  1. Detect BRISK features.
  2. Find similar images with bag of words database.
  3. Use RANSAC with P3P to find set of inlier matches.
  4. Send inlier pixel coordinates and 3D positions to EKF.
- AR Tags
  - Corners sent to EKF just like sparse map features.
- Handrails
  - Detected with time of flight depth sensor, points passed to EKF along with handrail axis.
- Optical Flow Features
  - Lucas Kanade with Good Features to Track.
  - Backwards tracking to remove outliers.

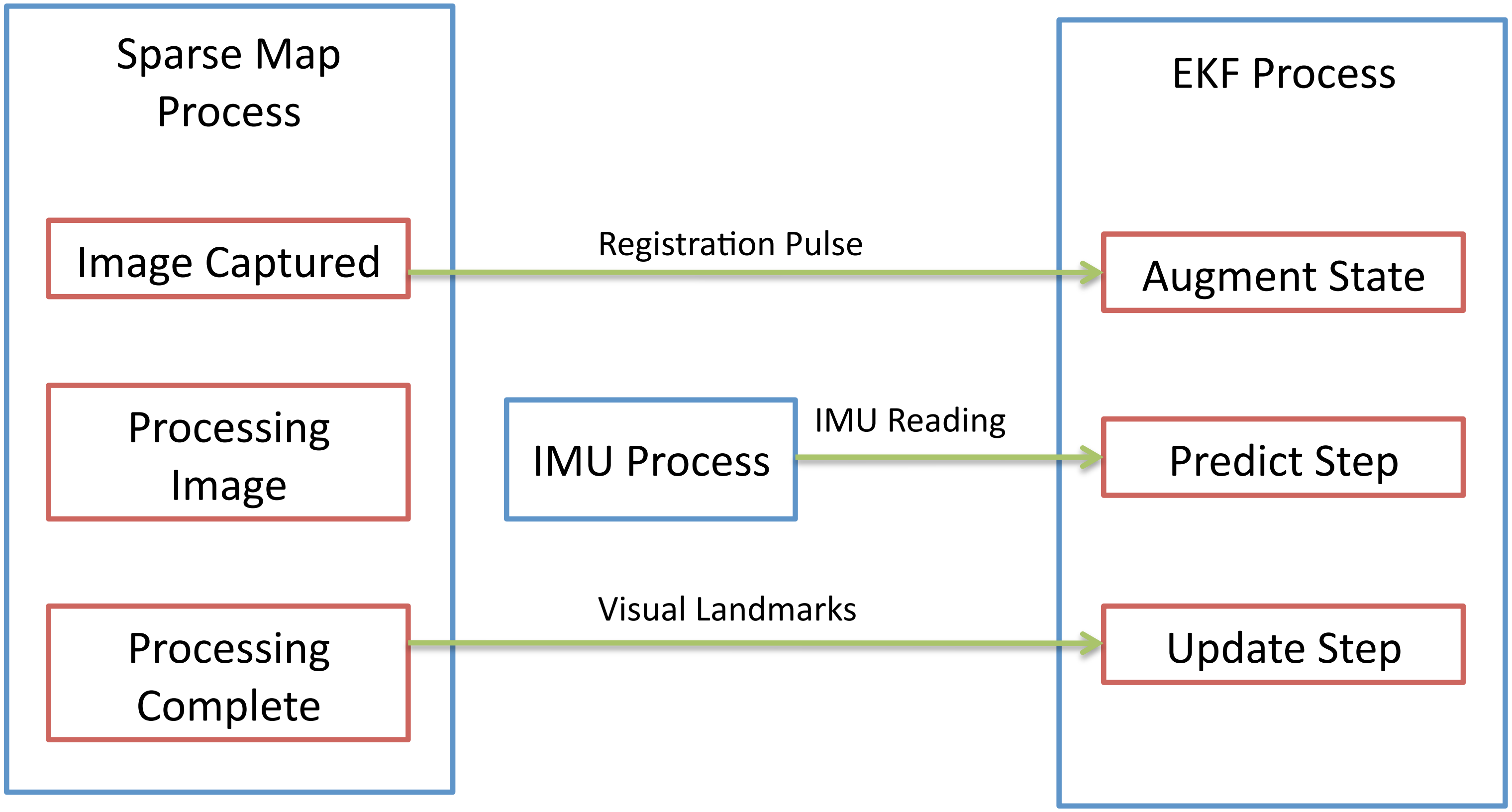
EKF Inputs



Name	Rate (Hz)	Value
Sparse Map	—	BRISK descriptors and positions
AR Tag Map	—	AR tag IDs and corner positions
IMU Acceleration	62.5	Linear acceleration a_{imu}
IMU Angular Vel.	62.5	Angular velocity ω_{imu}
Sparse Map Features	≈ 2	Coordinates in image and map
AR Tag Features	≈ 5	Coordinates in image and map
Handrail Features	≈ 5	Depth image and global positions
Optical Flow	≈ 5	Multiple image coordinates

EKF Position Tracking

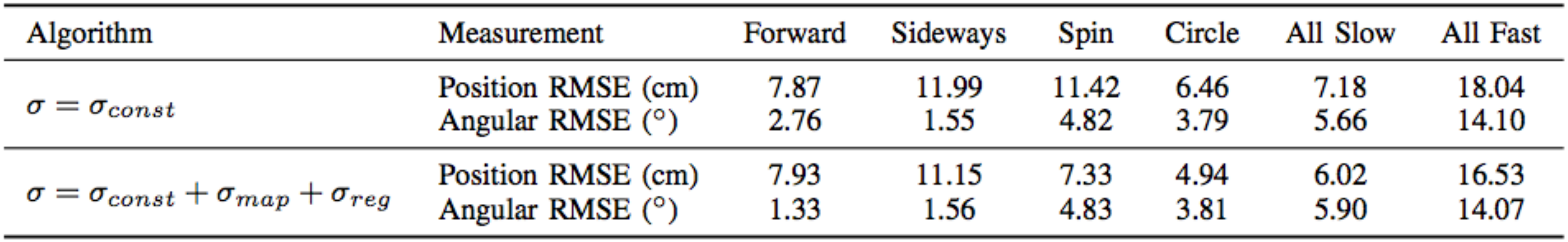
- Approach based on augmented error state extended Kalman filter by Mourikis, Trawny, Roumeliotis, et.al.
- Key idea: account for delay in image processing



EKF Tips and Tricks

- Account for errors in:
  - Map
  - Registration pulse timing
- Automatic IMU bias initialization
- (Re)initialization from RANSAC on visual landmarks
- Precise IMU position knowledge is key for centrifugal and Euler accelerations
- Many computational challenges, especially for optical flow

Selected Results



Algorithm	Measurement	Forward	Sideways	Spin	Circle	All Slow	All Fast
$\sigma = \sigma_{const}$	Position RMSE (cm)	7.87	11.99	11.42	6.46	7.18	18.04
	Angular RMSE (°)	2.76	1.55	4.82	3.79	5.66	14.10
$\sigma = \sigma_{const} + \sigma_{map} + \sigma_{reg}$	Position RMSE (cm)	7.93	11.15	7.33	4.94	6.02	16.53
	Angular RMSE (°)	1.33	1.56	4.83	3.81	5.90	14.07

- Pushed robot by hand and recorded logs
- Compared with and without additional error source modeling

[redacted]