# Visual Odometry (VO)

CSC2541, Feb 9th, 2016

Presented by Patrick McGarey

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

# Primer on Odometry

- What is Odometry?
  - The Greeks invented it… "Route Measure"
  - Estimating change in position over time
- What is Visual Odometry?
  - Estimating the motion of a camera in real time using sequential images (i.e., egomotion)
  - The idea was first introduced for planetary rovers operating on Mars – Moravec 1980

Pathfinder landing, 1997

# Primer on Visual Odometry

- Camera Types
  - Passive
    - *Monocular*
    - *Stereo*
    - *Omnidirectional*
  - Active
    - *Lidar*
    - *Time-of-flight*
    - *RGB-Depth*
  - Hybrid
    - *Uses multiple sensors*

UEye Camera


Point Grey Stereo Cam

Bubl Omnicam

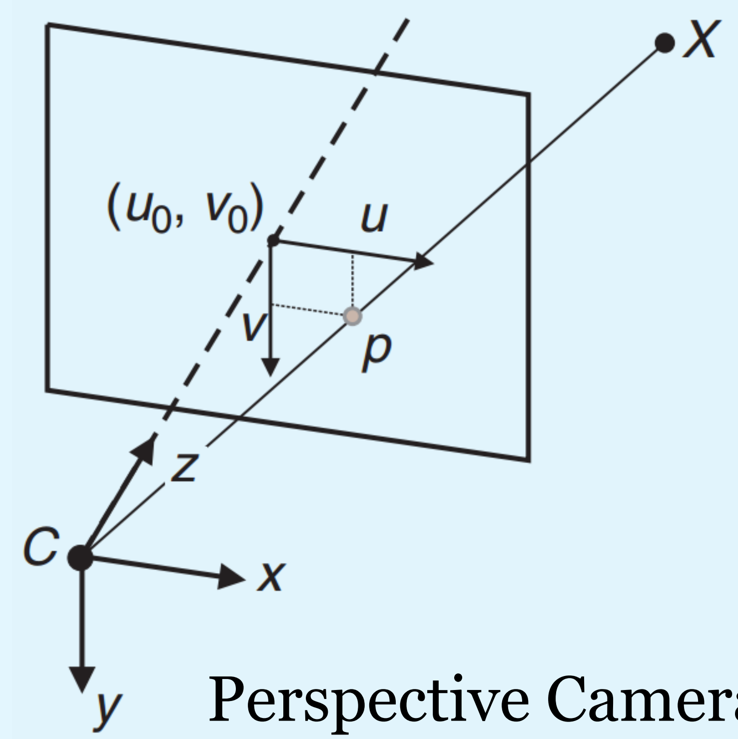

Velodyne Lidar

Mesa TOF Cam

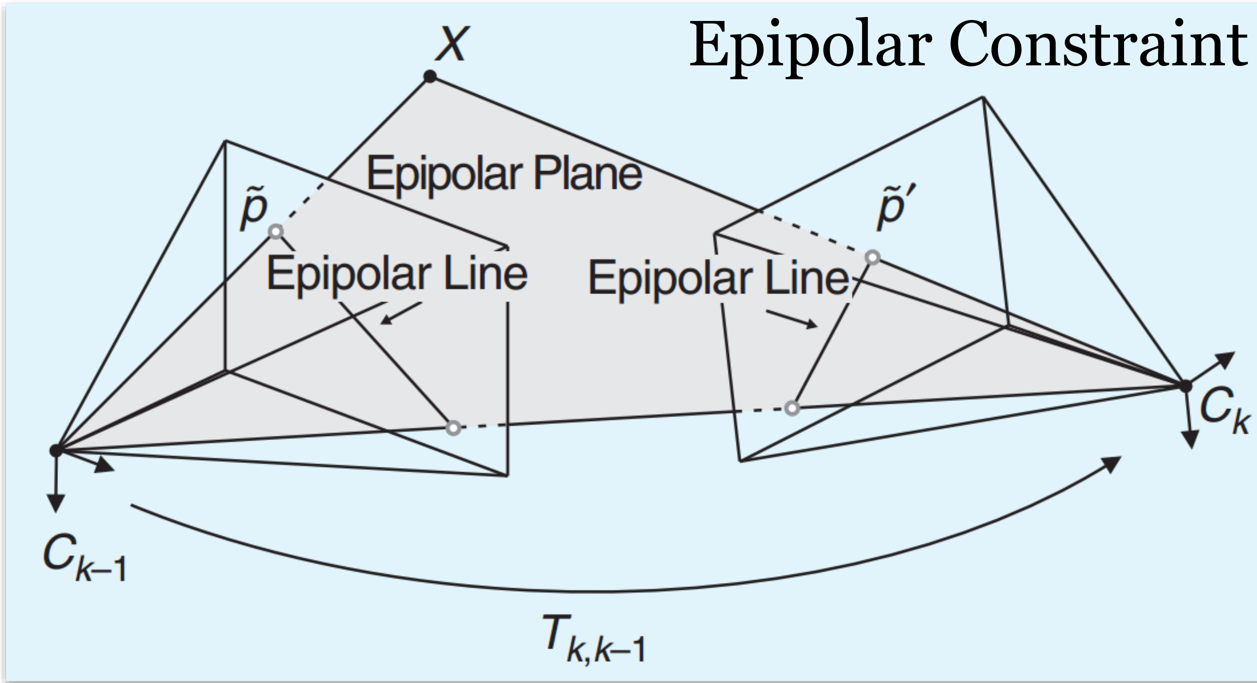
Kinect RGB-D

# Primer on Visual Odometry

- Monocular Visual Odometry
  - A single camera = angle sensor
  - Motion scale is unobservable (it must be synthesized)
  - Best used in hybrid methods
- Stereo Visual Odometry
  - Solves the scale problem
  - Feature depth between images
  - Degenerates to the monocular case if only distant features are used




Perspective Camera Model




Epipolar Constraint

Images from Scaramuzza and Fraundorfer, 2011

# Primer on Visual Odometry

- Monocular Visual Odometry
  - A single camera = angle sensor
  - Motion scale is unobservable (it must be synthesized)
  - Best used in hybrid methods
- Stereo Visual Odometry
  - Solves the scale problem
  - Feature depth between images
  - Degenerates to the monocular case if only distant features are used

VO Problem Illustration

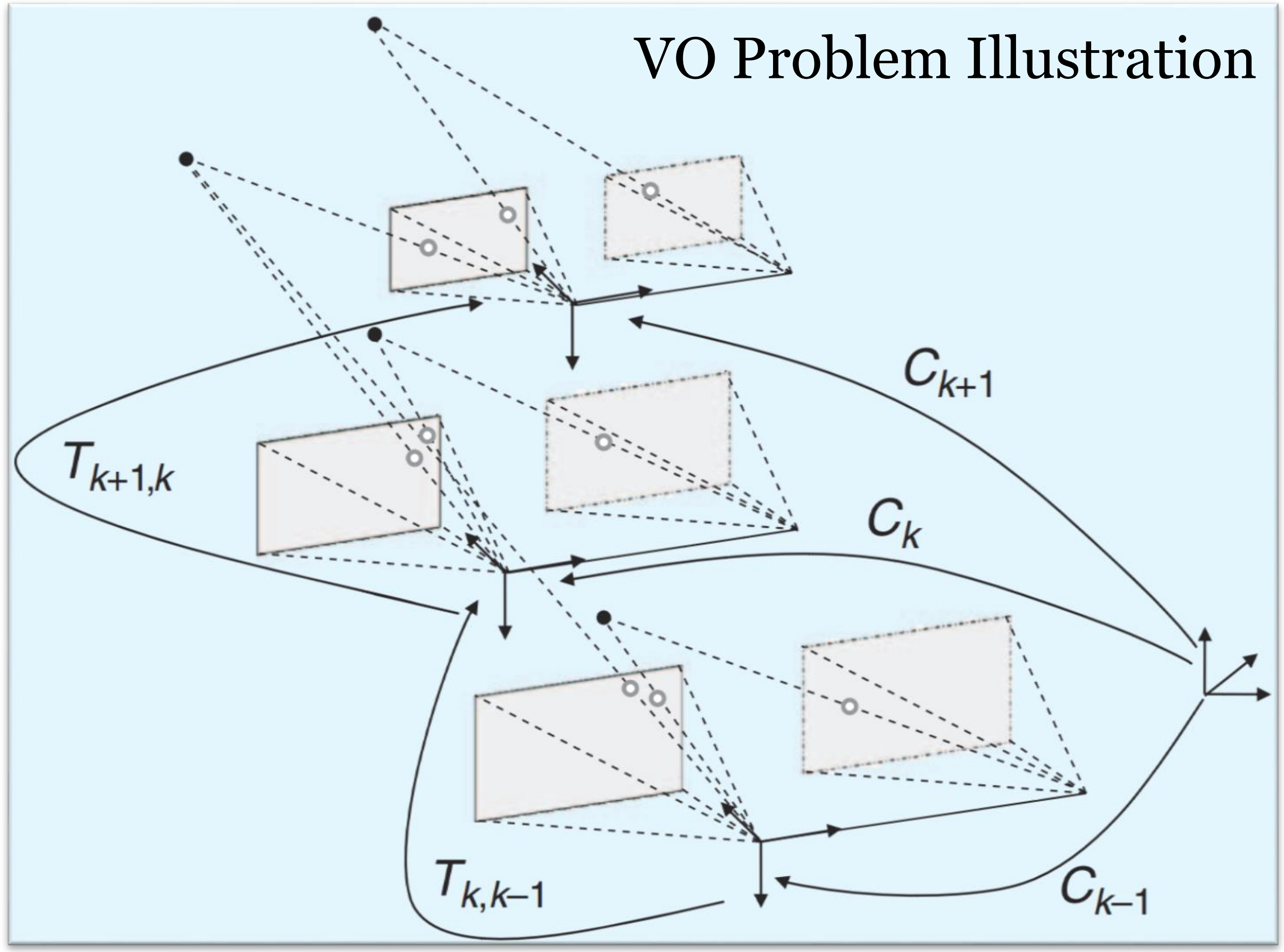



Image from Scaramuzza and Fraundorfer, 2011

# Primer on Visual Odometry

- Monocular Visual Odometry
  - A single camera = angle sensor
  - Motion scale is unobservable (it must be synthesized)
  - Best used in hybrid methods
- Stereo Visual Odometry
  - Solves the scale problem
  - Feature depth between images
  - Degenerates to the monocular case if only distant features are used

VO Pipeline

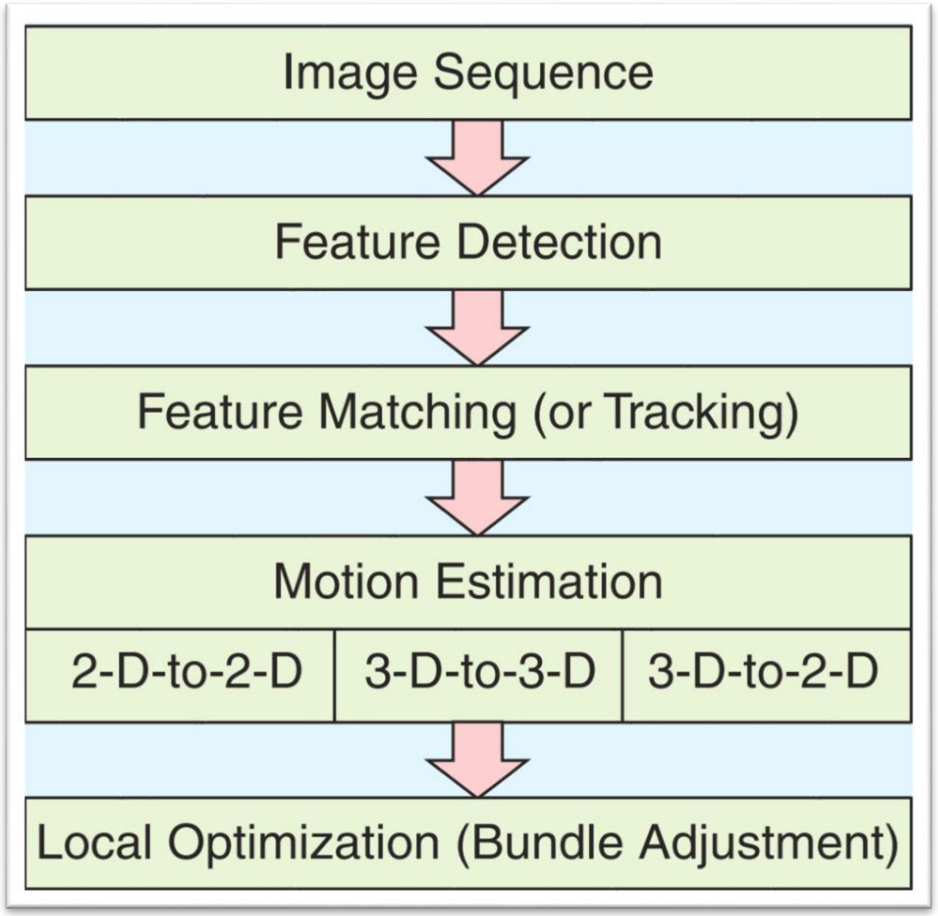



Image from Scaramuzza and Fraundorfer, 2011

# Primer on Visual Odometry

## Algorithm 1. VO from 2-D-to-2-D correspondences.

1) Capture new frame $I_k$
2) Extract and match features between $I_{k-1}$ and $I_k$
3) Compute essential matrix for image pair $I_{k-1}, I_k$
4) Decompose essential matrix into $R_k$ and $t_k$, and form $T_k$
5) Compute relative scale and rescale $t_k$ accordingly
6) Concatenate transformation by computing $C_k = C_{k-1} T_k$
7) Repeat from 1).

Image from Scaramuzza and Fraundorfer, 2011

- Essential Matrix is computed from features correspondences using epipolar constraint
- The matrix has an unknown scale factor (problem with Monocular)
- Commonly solved with the Nister Five-Point Algorithm [2003] (solves using SVD)

# Primer on Visual Odometry

**Algorithm 2. VO from 3-D-to-3-D correspondences.**

1) Capture two stereo image pairs $I_{l,k-1}, I_{r,k-1}$ and $I_{l,k}, I_{r,k}$
2) Extract and match features between $I_{l,k-1}$ and $I_{l,k}$
3) Triangulate matched features for each stereo pair
4) Compute $T_k$ from 3-D features $X_{k-1}$ and $X_k$
5) Concatenate transformation by computing $C_k = C_{k-1} T_k$
6) Repeat from 1).

Image from Scaramuzza and Fraundorfer, 2011

# Use Cases for Visual Odometry

- Motion estimation for vehicles
  - Driver assistance
  - Autonomy
- Point-Cloud Mapping
  - VO to estimate the motion of a lidar during the collection of a single scan
  - Reduce rolling shutter effect
- Challenges
  - Robustness to lighting conditions
  - Lack of features / non-overlapping images
  - Without loop closure the estimate still drifts

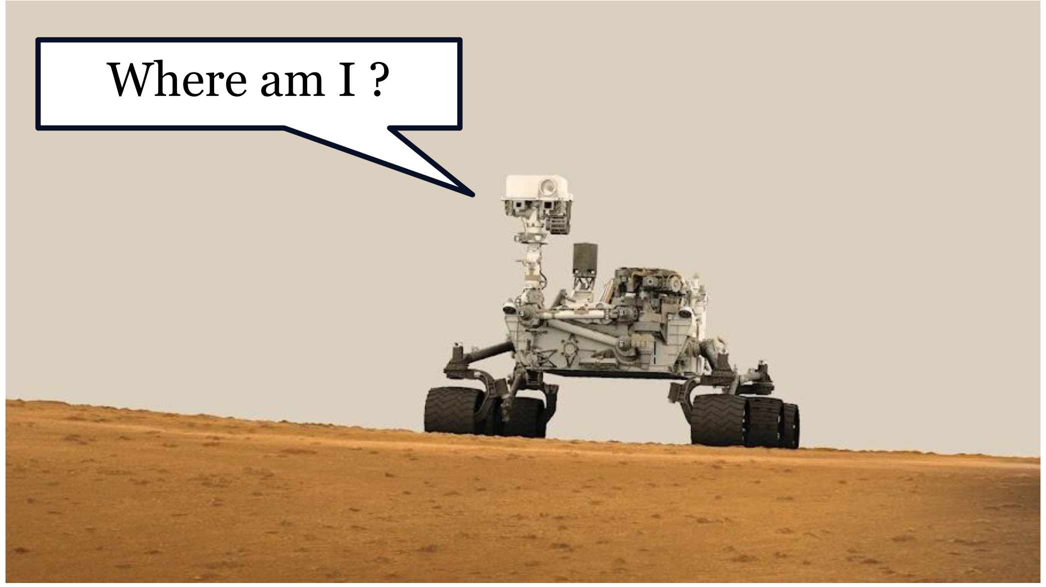



Image from NASA/JPL

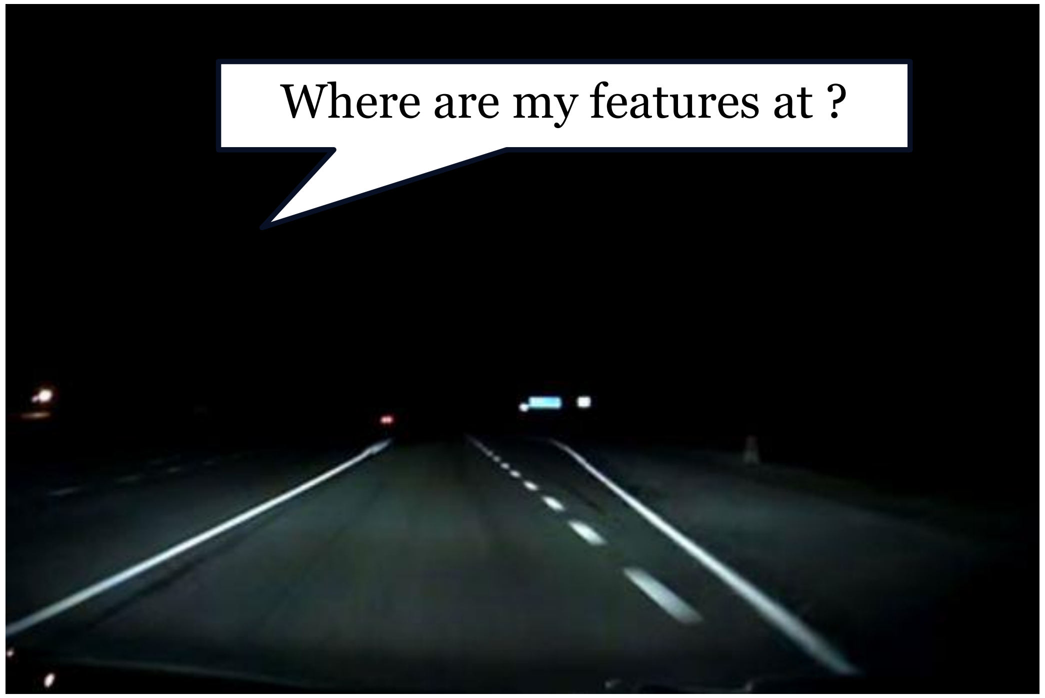



Image from mirror.co.uk

Paper Review

# Visual-Lidar Odometry and Mapping:
## Low-drift, Robust, and Fast (Robotics Institute, CMU)

Paper by: Ji Zhang and Sanjiv Singh, ICRA 2015

*Presented by Patrick McGarey*

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

# General Problems

- Localization and mapping in degenerate environments is challenging
- e.g., lack of discrete/unique features, or poor lighting

Images from Zhang and Singh, 2015

# Problems with VO & LO

- VO requires moderate-good lighting conditions
- VO fails when features are limited
- Lidar Odometry (LO) is limited because of motion distortion
- LO inherently involves many variables (the world is represented by points)
- LO depends on scan matching, which fails in degenerate environments

Image from swinburne.edu.my

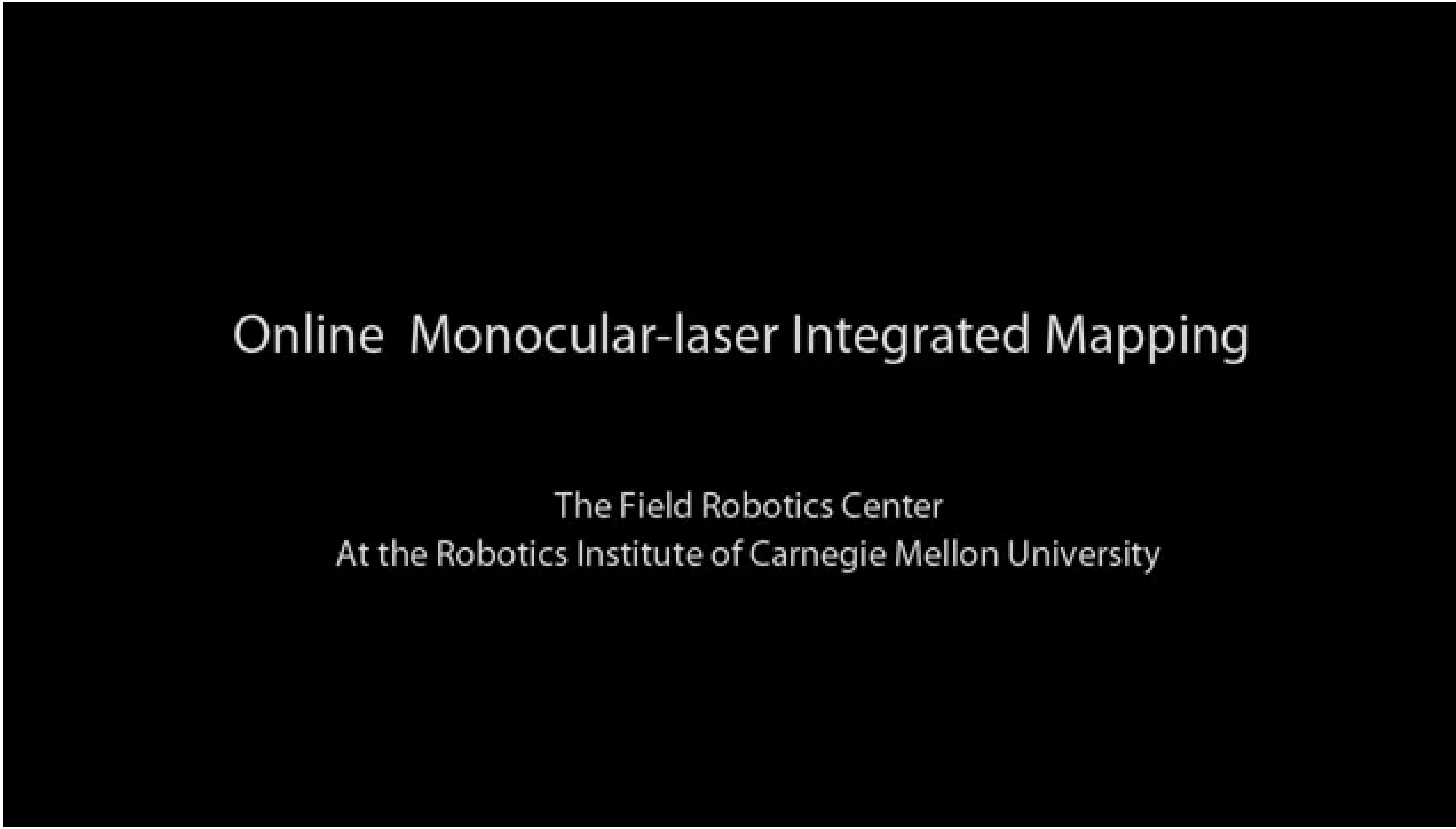

Image from IEEE Spectrum

# Motivation

- Estimate 6-DOF motion of a camera/lidar rig
- Collect spatially accurate 3D point cloud of the environment
- Exploit complementary strengths and weaknesses of a monocular camera and lidar
- Be robust to (1) aggressive motion and (2) intermittent feature dropouts


Image from Zhang and Singh, 2015

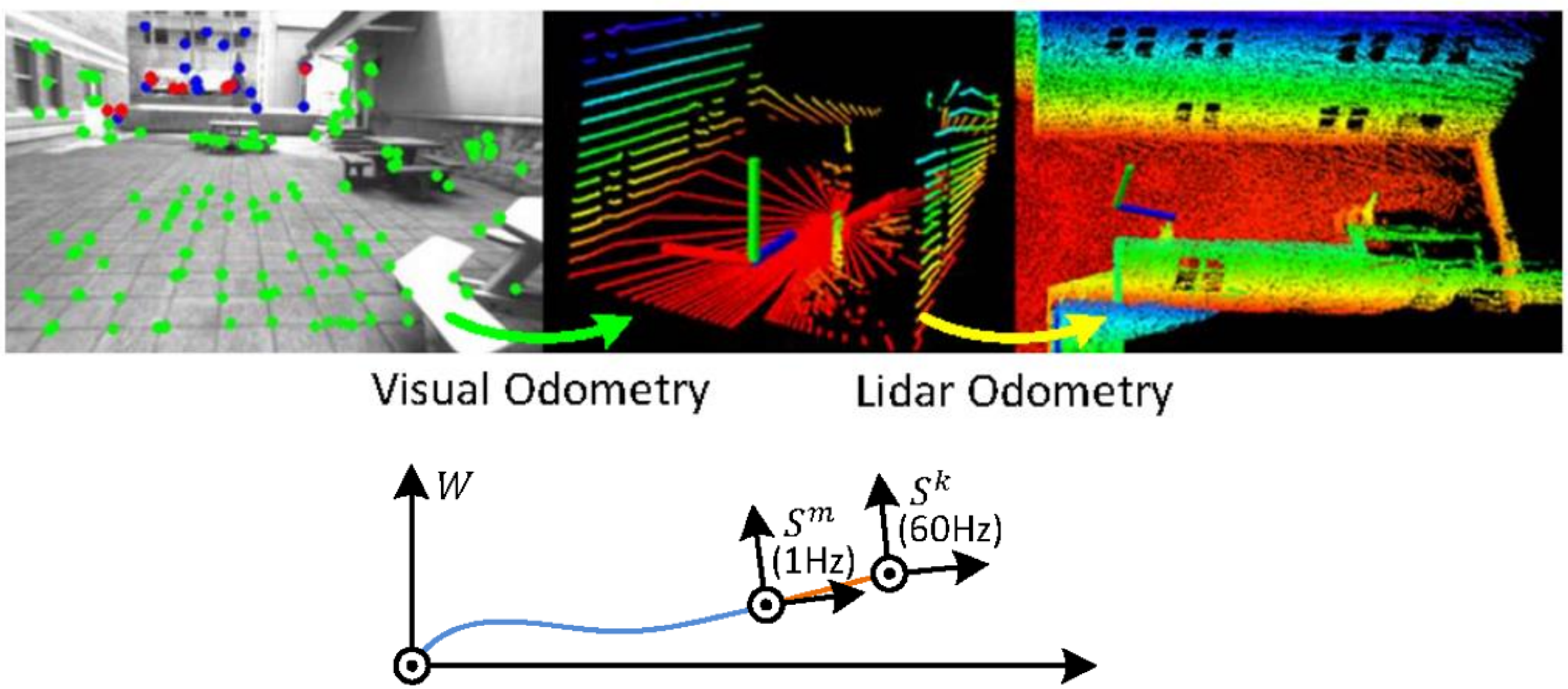
# Preview




Video from Zhang and Singh, 2015

# Method V-LOAM

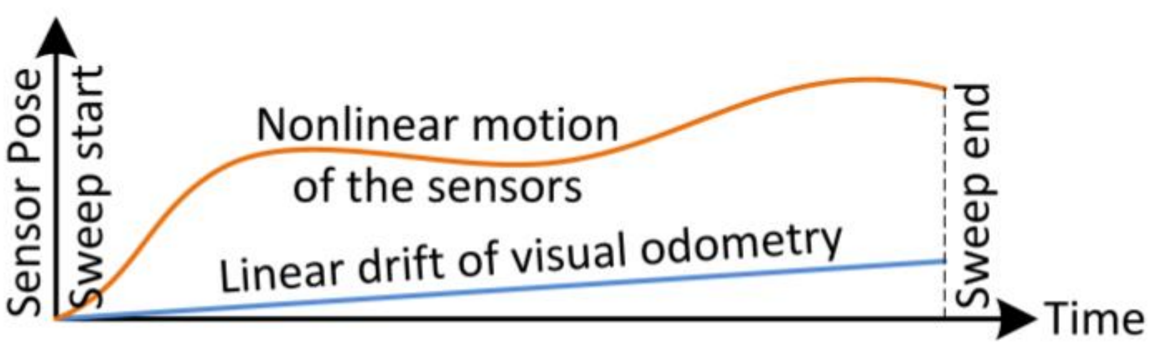
- Use VO at  high frame-rate (60Hz) to estimate motion
- Use LO at low frequency (1Hz) to refine motion estimate
  - Uses course point cloud from last iteration to provide depth to features
- Use improved VO motion to undistort point cloud
- Merge current point cloud into the  global map




Images from Zhang and Singh, 2015

# Method – Visual Odometry (VO)

- Course lidar data is used to add depth to high-rate images
  - Camera motion is approximated as linear for the short distances between images
- 3 Types of features are generated, those with…
  - no depth, depth from lidar, and depth from triangulation (i.e., SfM)
- Solve equation with 6 unknowns using least squares
- Output high-frequency transform



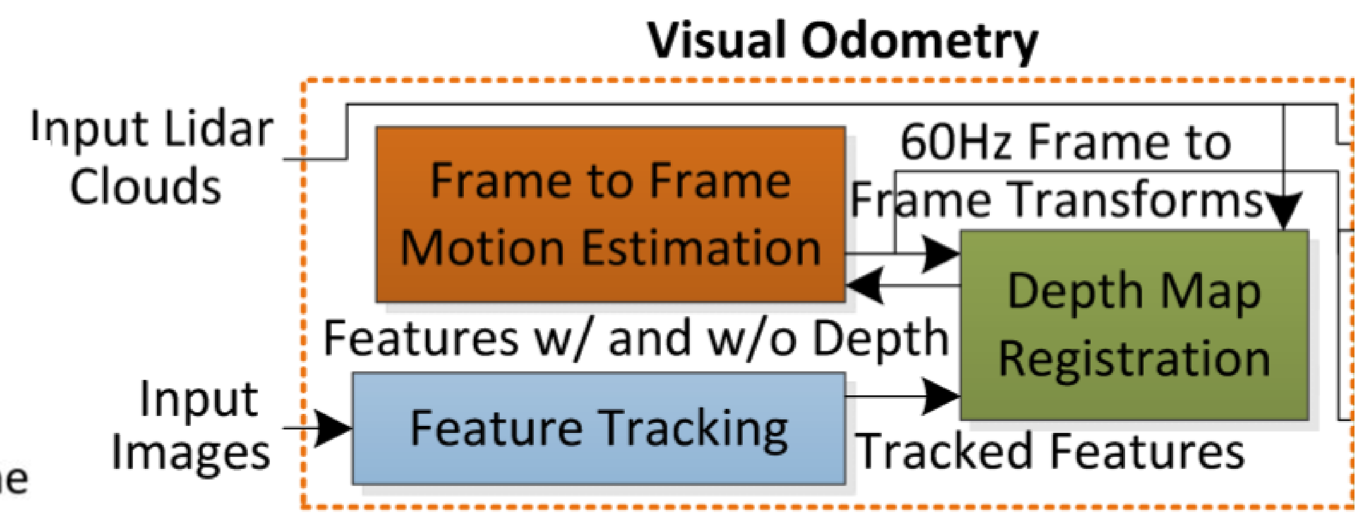
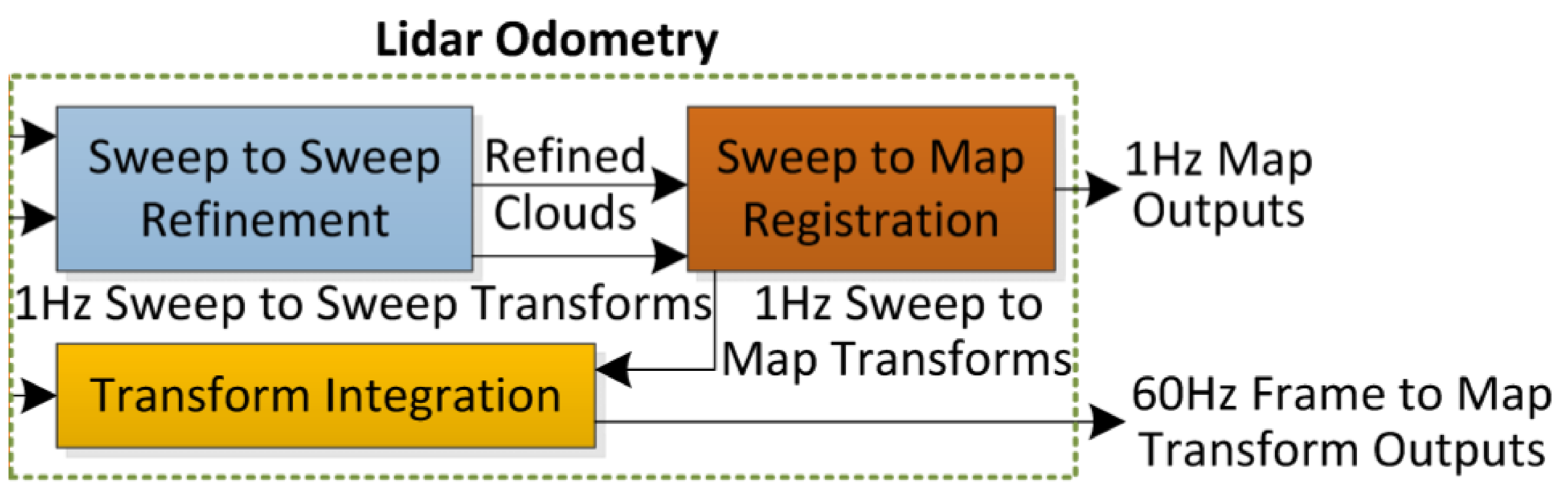




Image from Zhang and Singh, 2015

# Method – Lidar Odometry (LO)

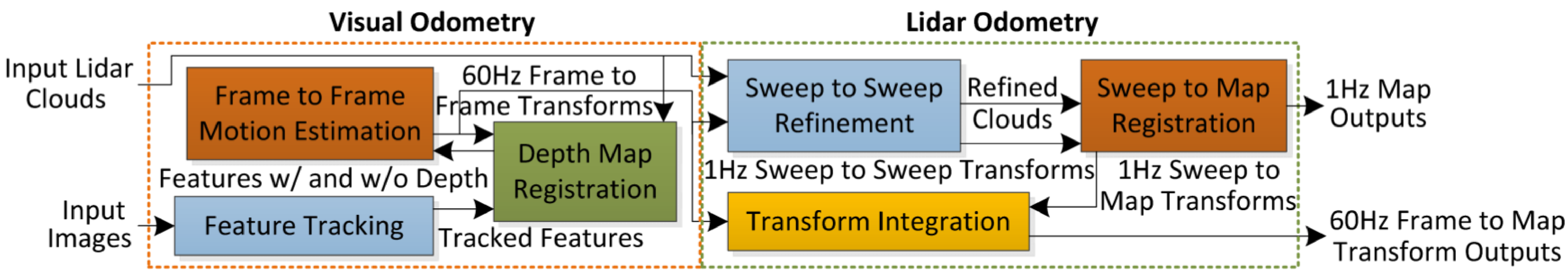
Steps

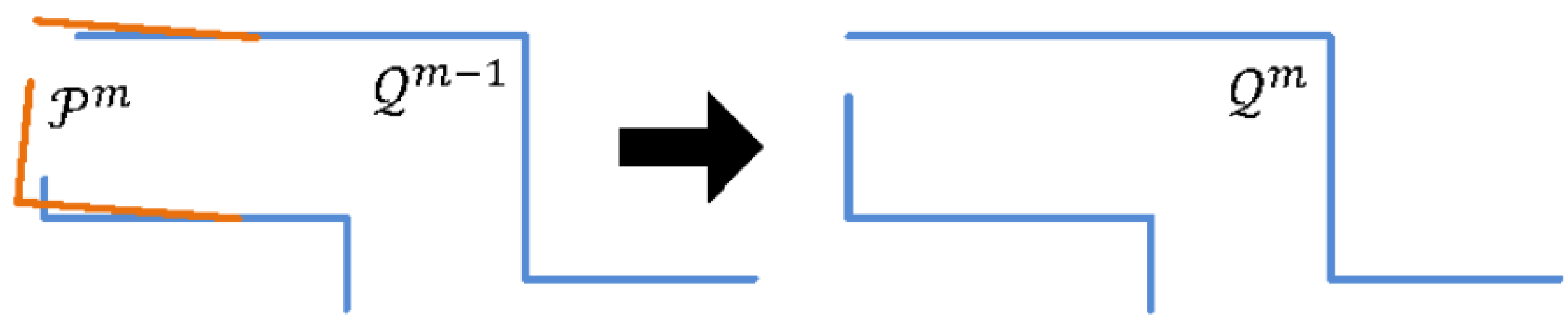
- 1) Sweep-to-Sweep: match 2 consecutive point clouds using ICP*
  - Removes some distortion, *(ICP = Iterative Closest Point algorithm)
- 2) Sweep-to-Map: to the map add undistorted point cloud
  - Along with the cloud, provide low-rate sensor poses
- 3) Using input from VO, output high-rate transforms and update map
  - Removes more distortion




Images from Zhang and Singh, 2015

# Method – Overall Pipeline VO+LO



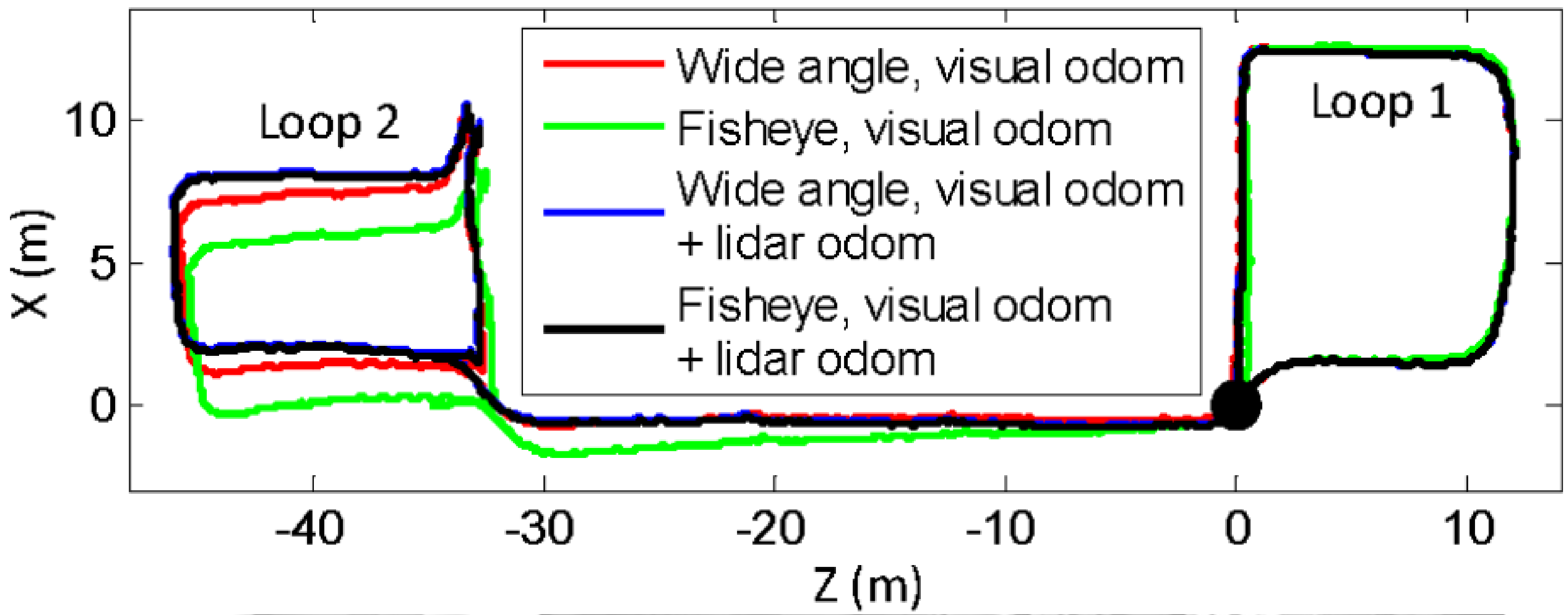



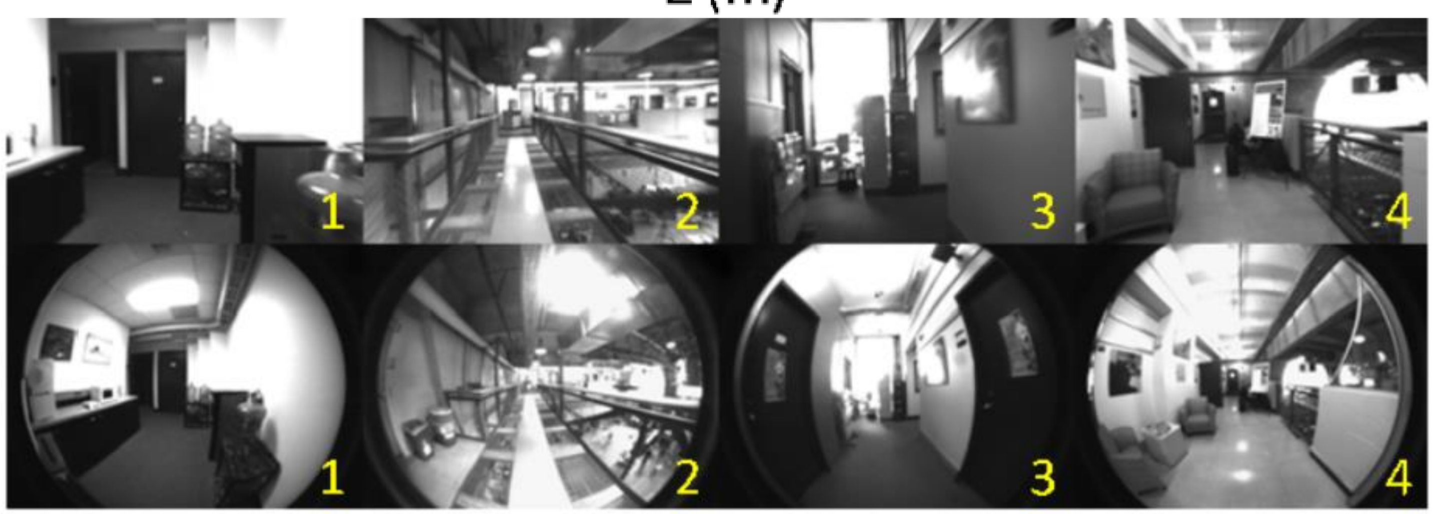

Images from Zhang and Singh, 2015

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

# Results – Wide Angle vs. Fisheye Lenses

- Confirm Low Drift
- Fisheye drifts faster than wide angle for lack of features
- Incorporating LO results in equal performance

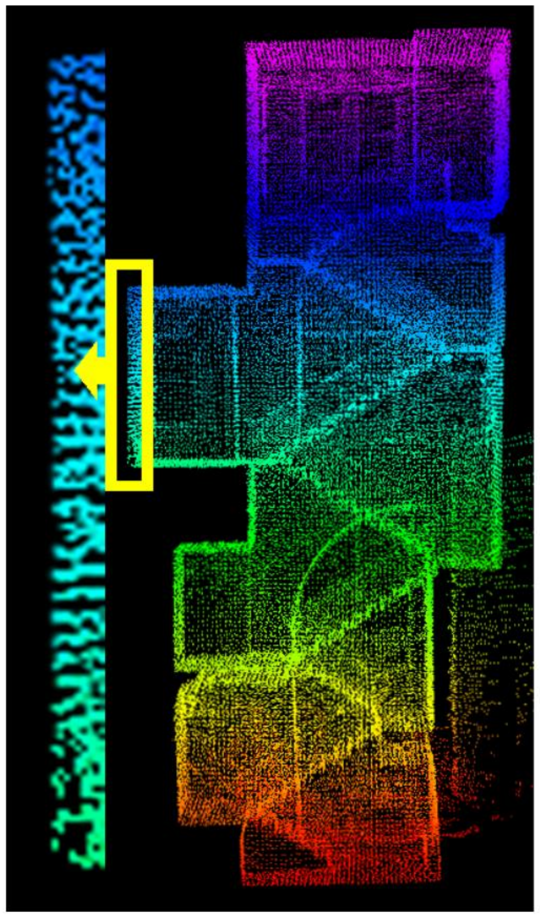


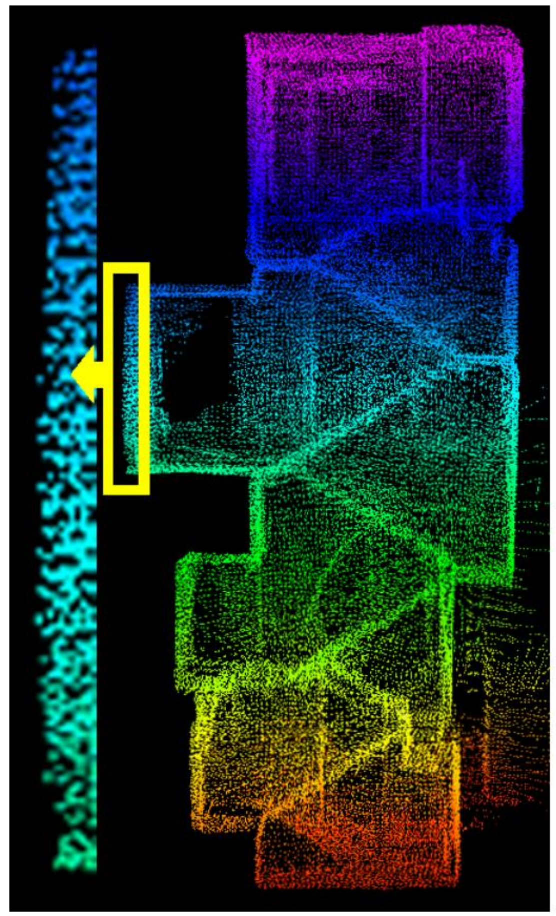

Images from Zhang and Singh, 2015

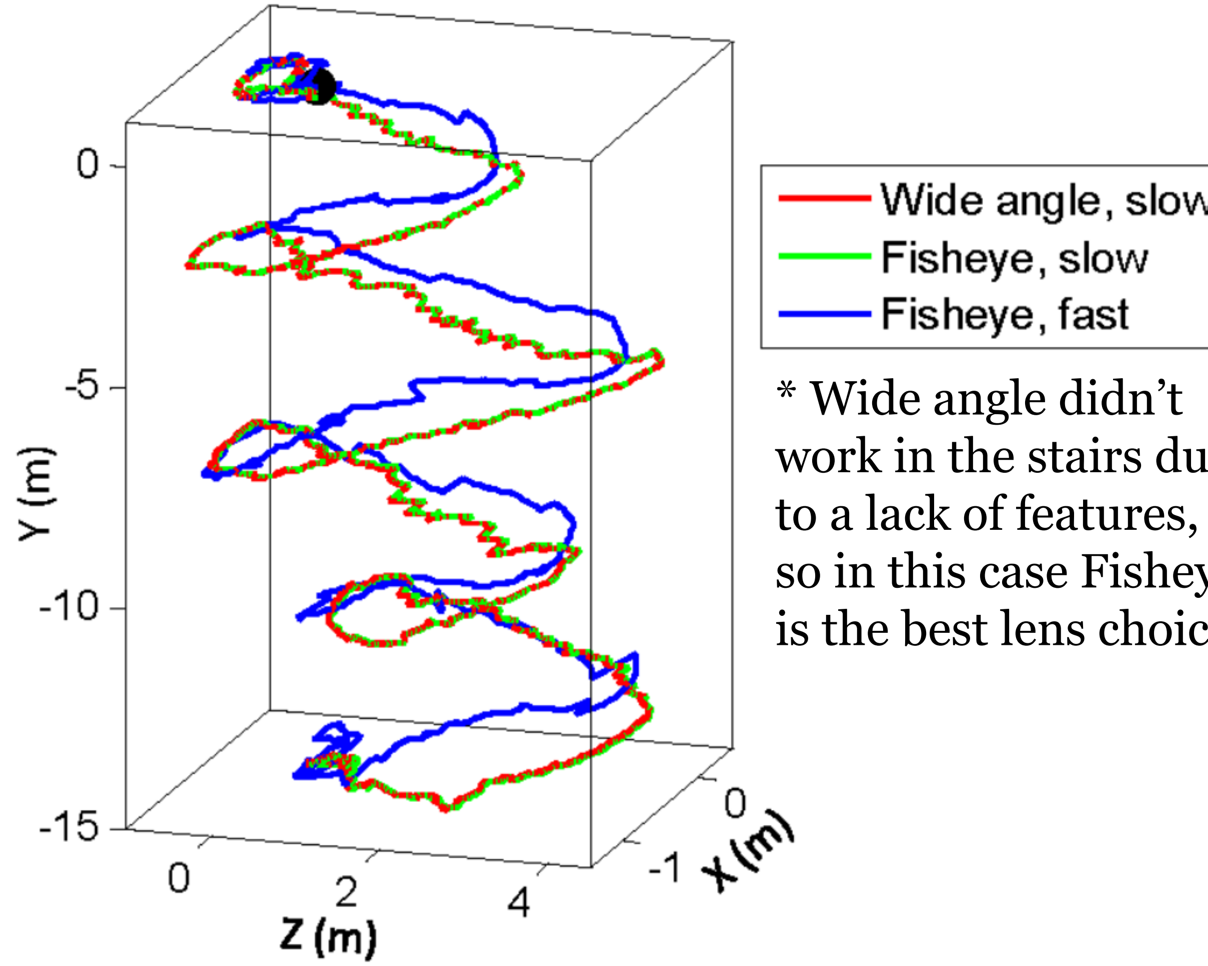
# Results – Fast (aggressive) vs. Slow Motion

Slow

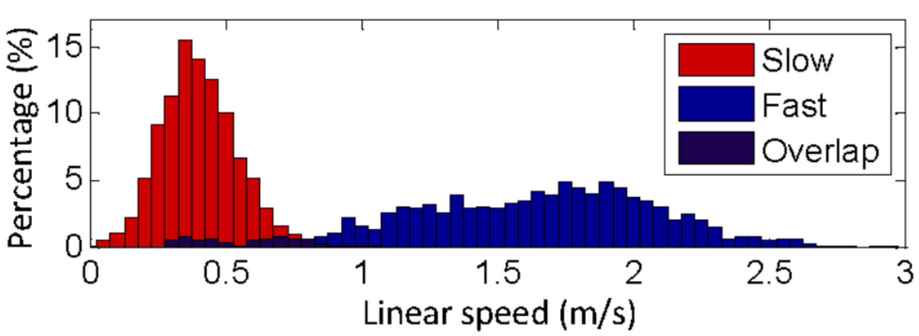


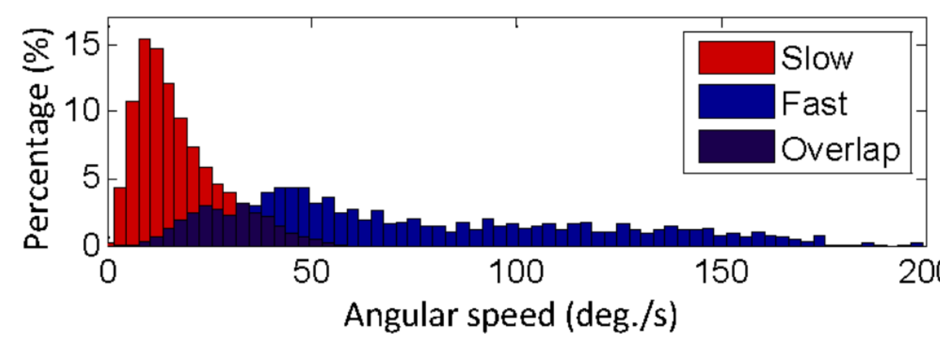
Fast






* Wide angle didn't work in the stairs due to a lack of features, so in this case Fisheye is the best lens choice

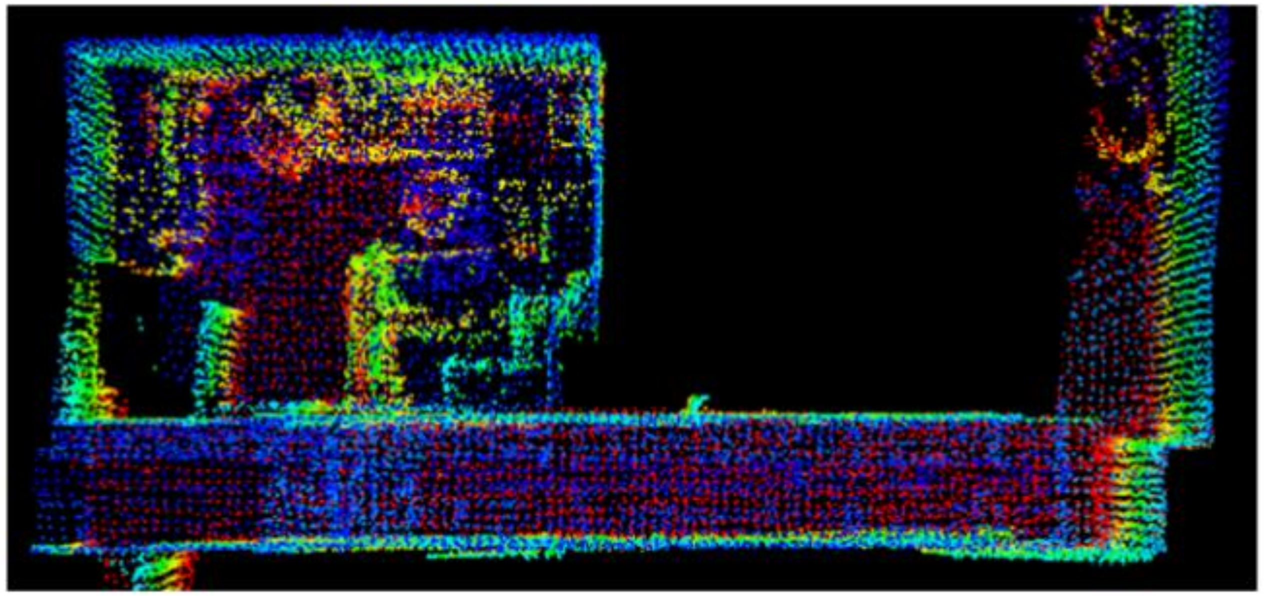


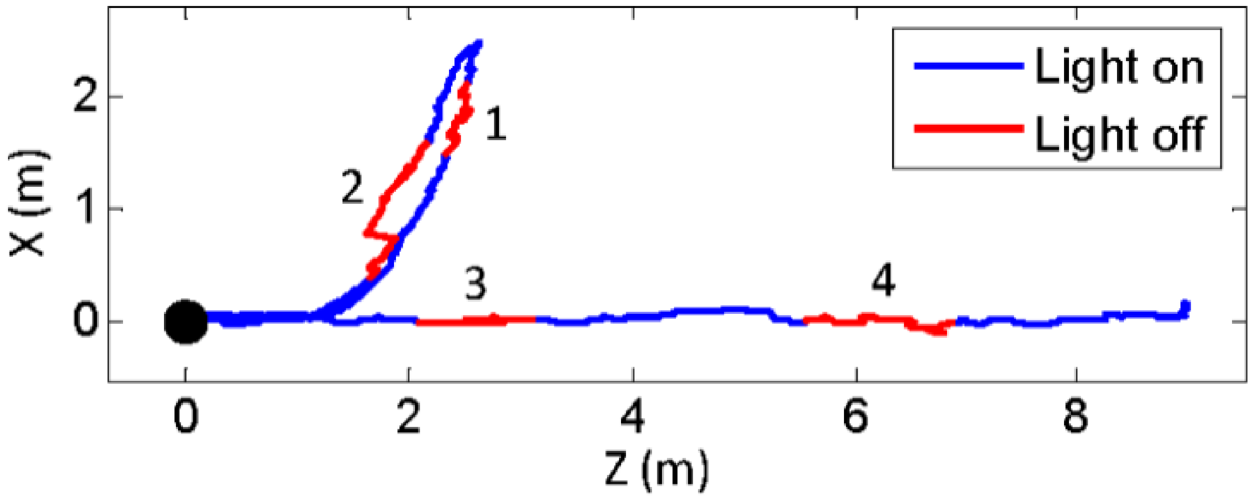



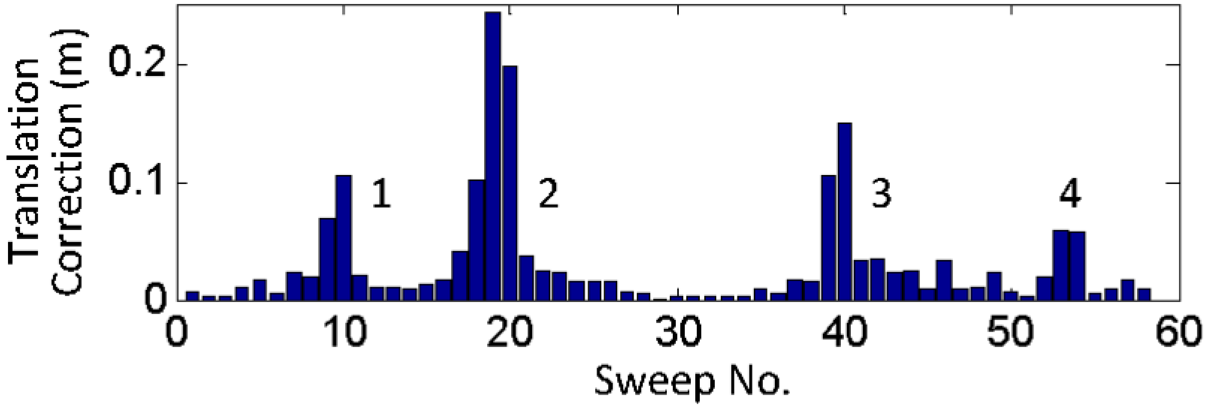

Images from Zhang and Singh, 2015

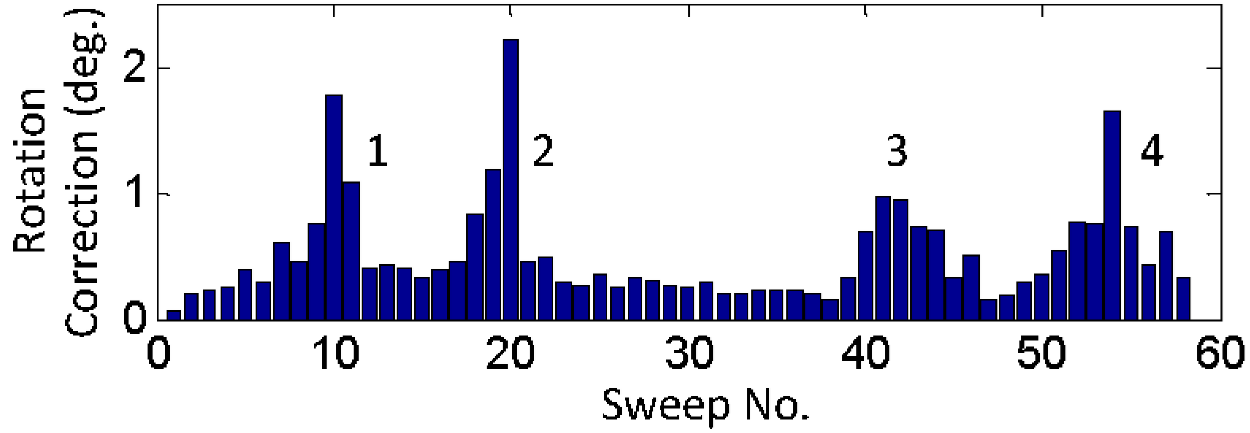
# Results – Light vs. Dark

- Robustness to feature dropouts
- Turn on and off the lights (for 2 seconds)





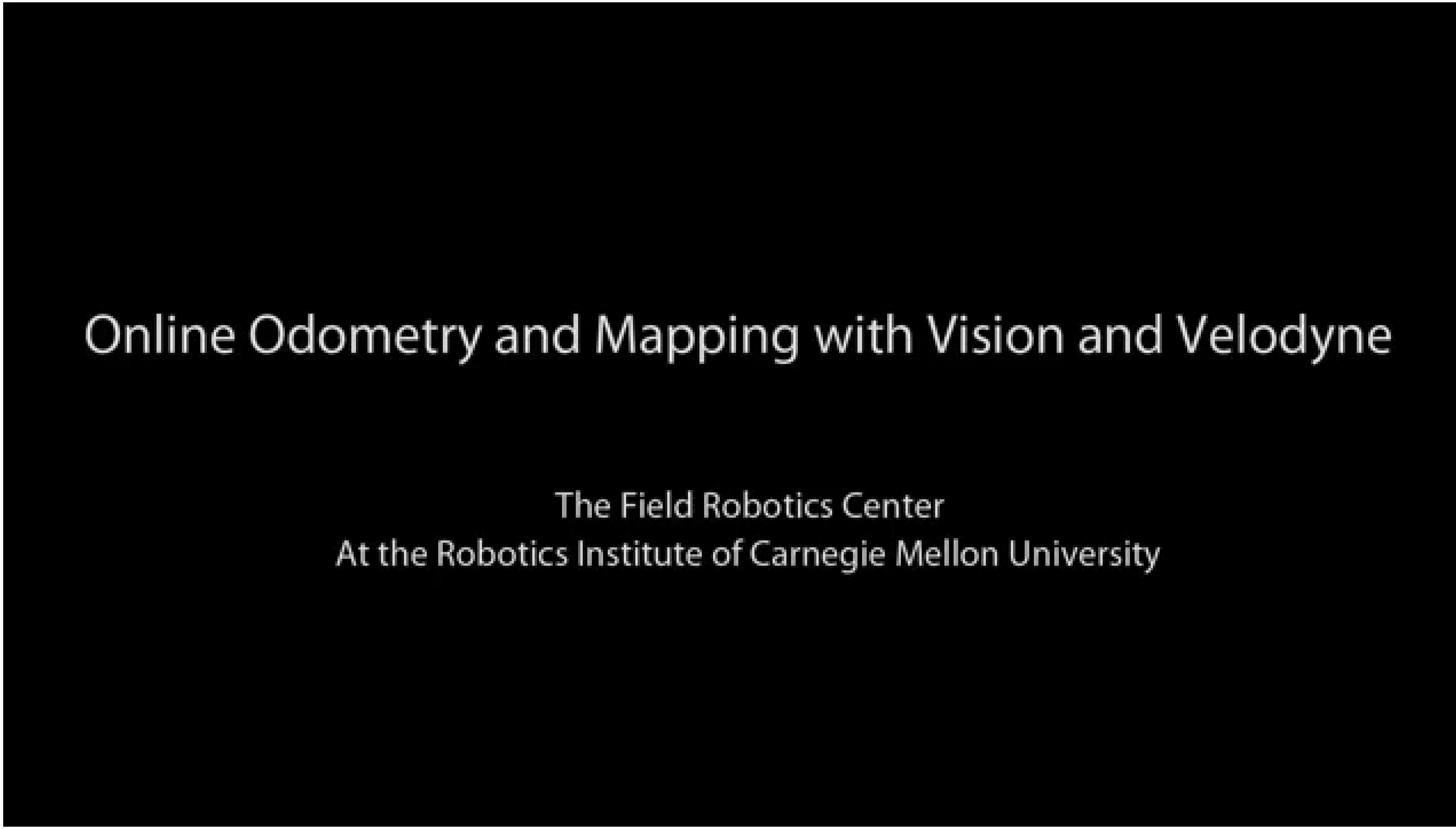

Images from Zhang and Singh, 2015

# Results – KITTI Benchmark

| | Method | Setting | Code | Translation | Rotation | Runtime | Environment |
|---|---|---|---|---|---|---|---|
| 1 | V-LOAM | | | **0.68 %** | **0.0015 [deg/m]** | 0.1 s | 4 cores @ 2.5 Ghz (C/C++) |
| J. Zhang and S. Singh: Visual-lidar Odometry and Mapping: Low- rift, Robust, and Fast. IEEE International Conference on Robotics and Automation(ICRA) 2015. | | | | | | | |
| 2 | LOAM | | | 0.88 % | 0.0022 [deg/m] | 1.0 s | 2 cores @ 2.5 Ghz (C/C++) |
| J. Zhang and S. Singh: LOAM: Lidar Odometry and Mapping in Real-time. Robotics: Science and Systems Conference (RSS) 2014. | | | | | | | |
| 3 | SOFT | | | 0.88 % | 0.0022 [deg/m] | 0.1 s | 2 cores @ 2.5 Ghz (C/C++) |
| I. Cvišić and I. Petrović: Stereo odometry based on careful feature selection and tracking. European Conference on Mobile Robots (ECMR) 2015. | | | | | | | |

Images from Zhang and Singh, 2015

# Updated Results




Video from Zhang and Singh, 2015

Paper Review

# Stereoscan: Dense 3D Reconstruction in Real-Time

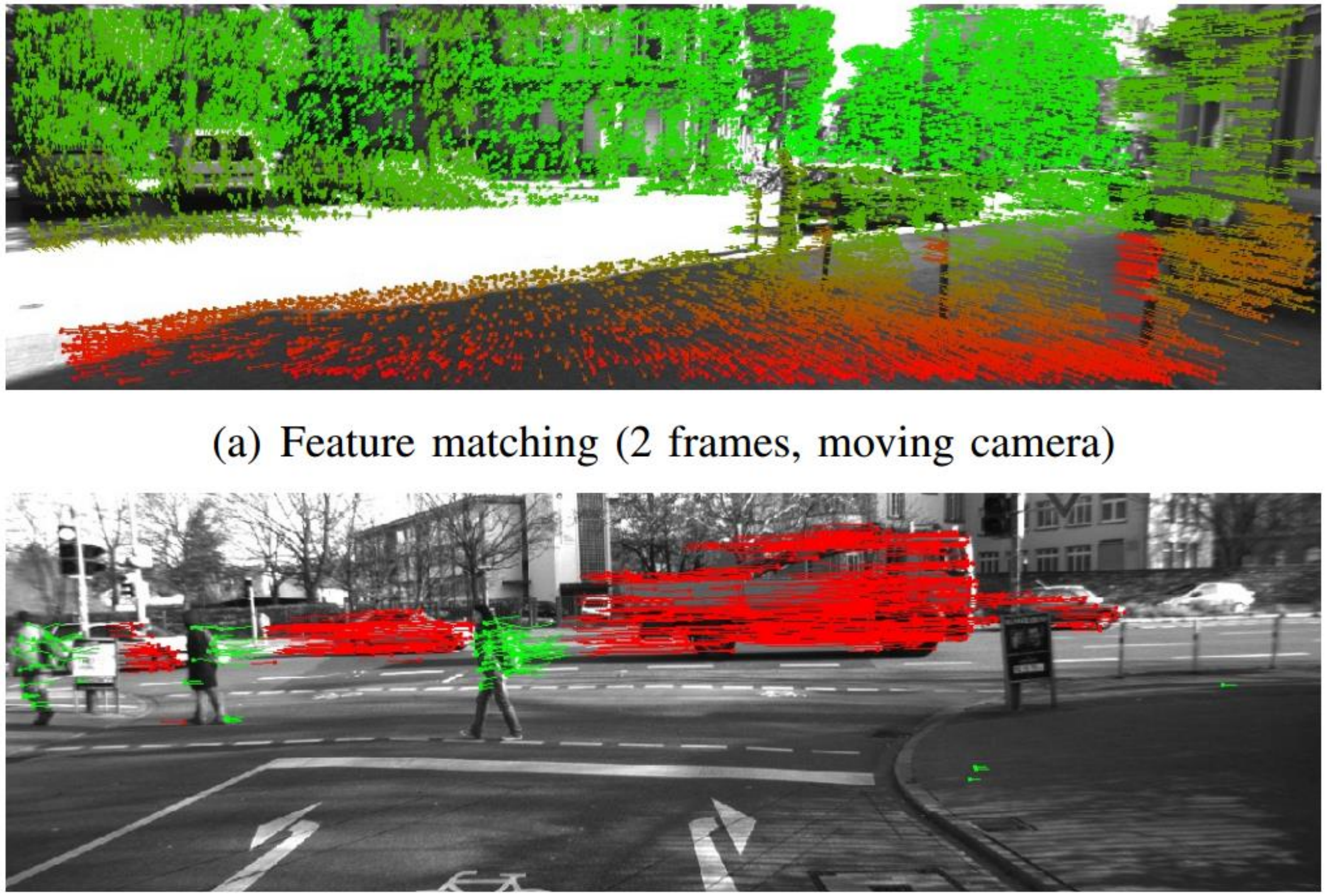
Paper by: Geiger et al., IVS 2011

*Presented by Patrick McGarey*

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

# Problems

- The accuracy of real-time stereo VO is a function of image resolution, meaning resolution is typically reduced to achieve faster performance
- Current stereo VO pipelines are computationally expensive, implying that mobile and embedded systems must be fully dedicated to the task



(a) Feature matching (2 frames, moving camera)

(b) Feature tracking (5 frames, static camera)

Image from Geiger et al., 2011

# Method

- Perform 3D reconstruction in real-time by leveraging
  - Sparse features for efficient stereo matching
  - Multi-view reconstruction (recast 3D points from last frame into scene)

H.S. in Carbonite, Star Wars

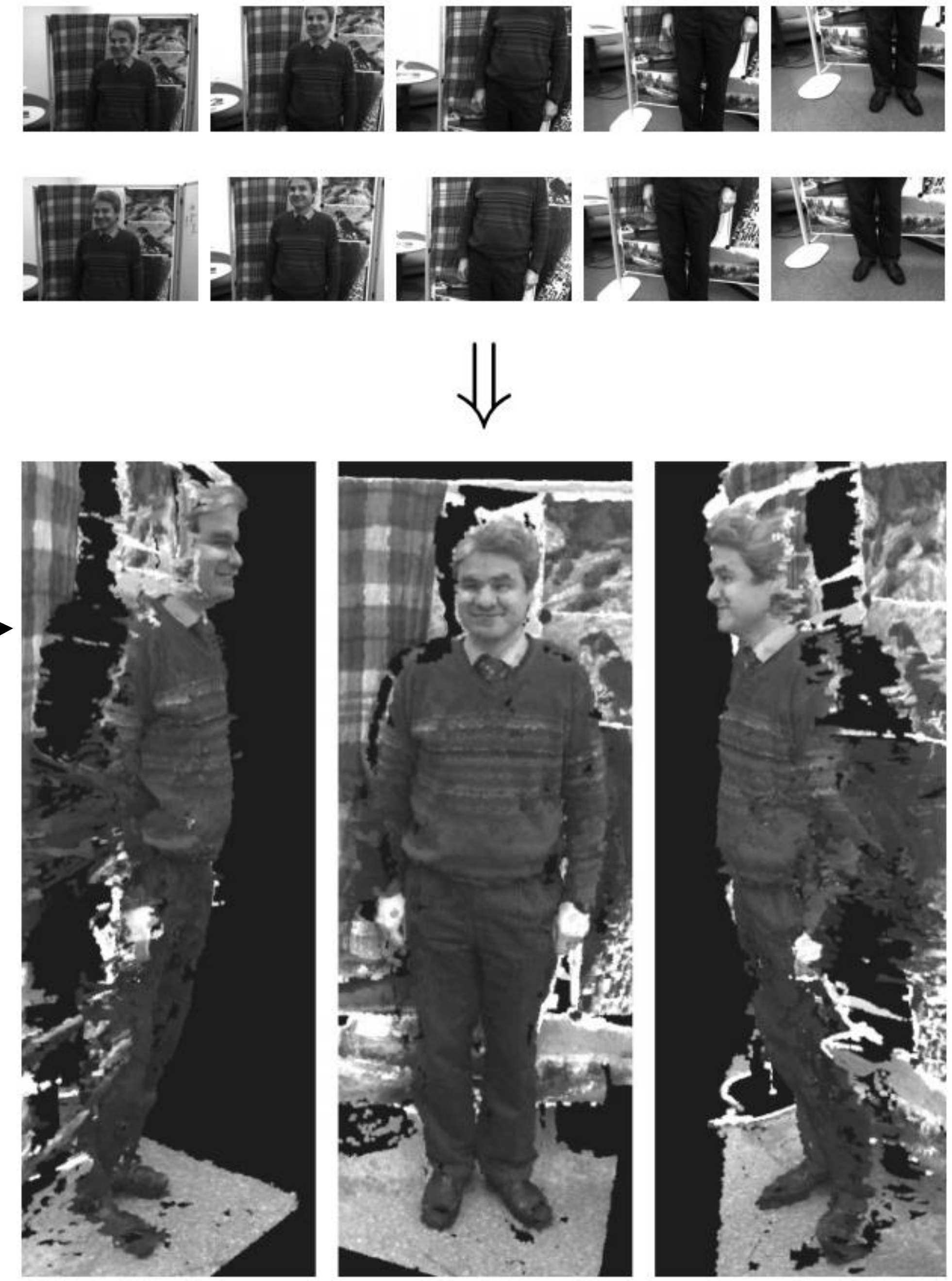
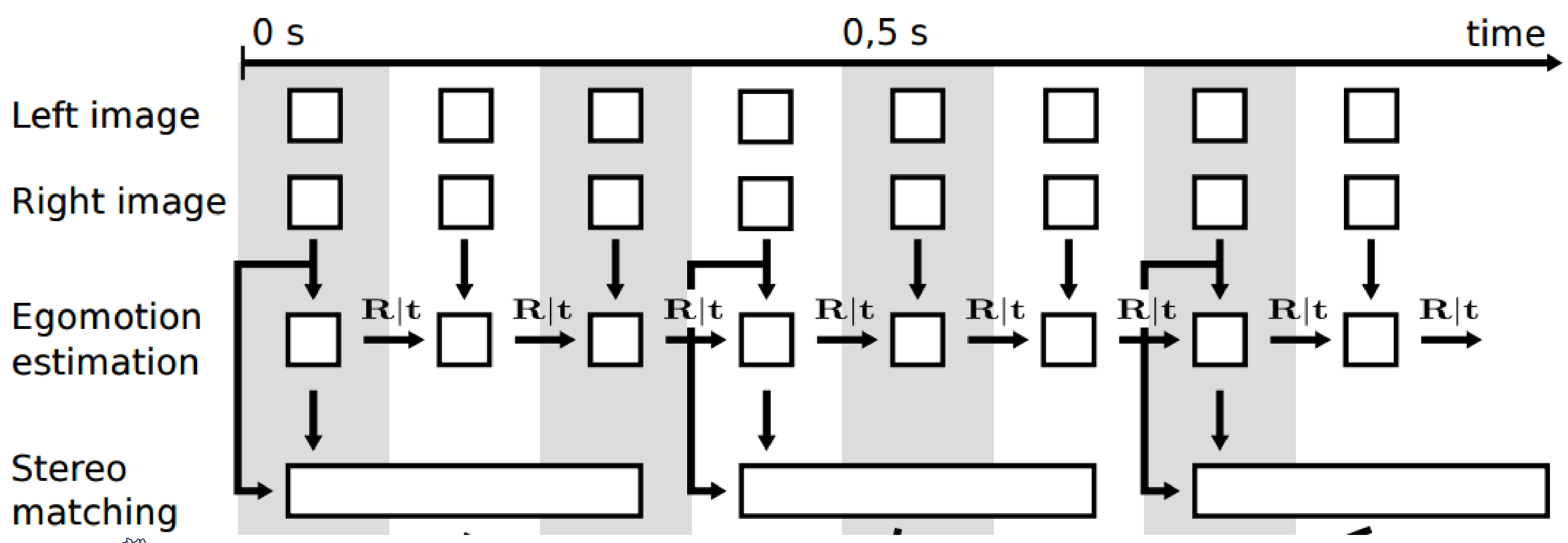
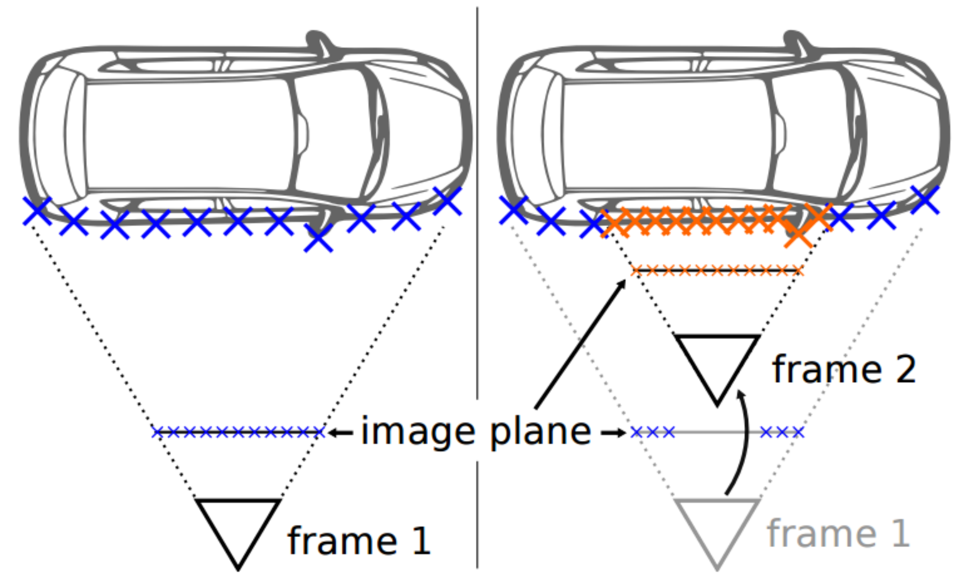


Image from Geiger et al., 2011

# Method: Pipeline

- 1) Feature Matching: only a subset of features detected are used to match in reduced search windows. (other CV tricks are employed)
- 2) Egomotion Estimation: minimize projection errors using EKF
- 3) Stereo Matching: uses ELAS method
  - Efficient Large Scale Stereo Mapping, Geiger et al., 2010
- 4) 3D Reconstruction: cast prior 3D points into current frame and take the mean pose of the combined 3D point and a new point on the image
  (they do this to create consistent point clouds from large amounts of data)



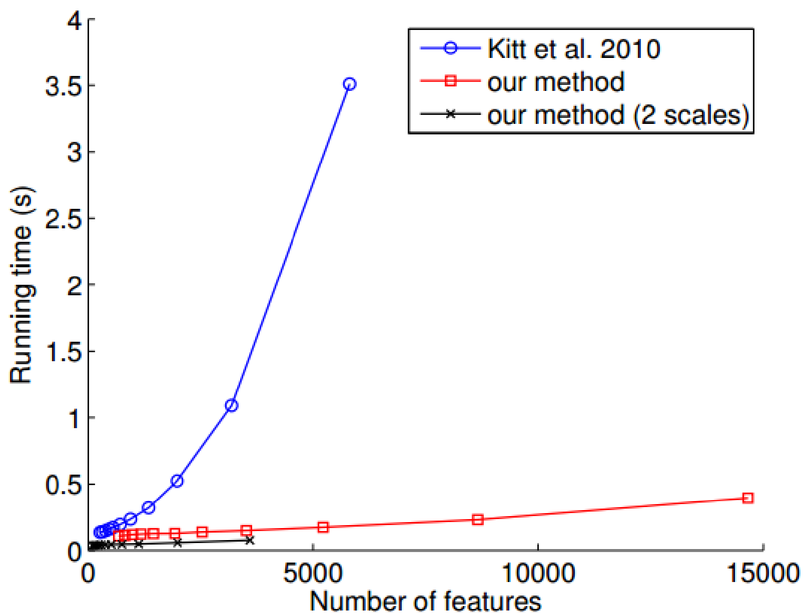




Images from Geiger et al., 2011

# Results

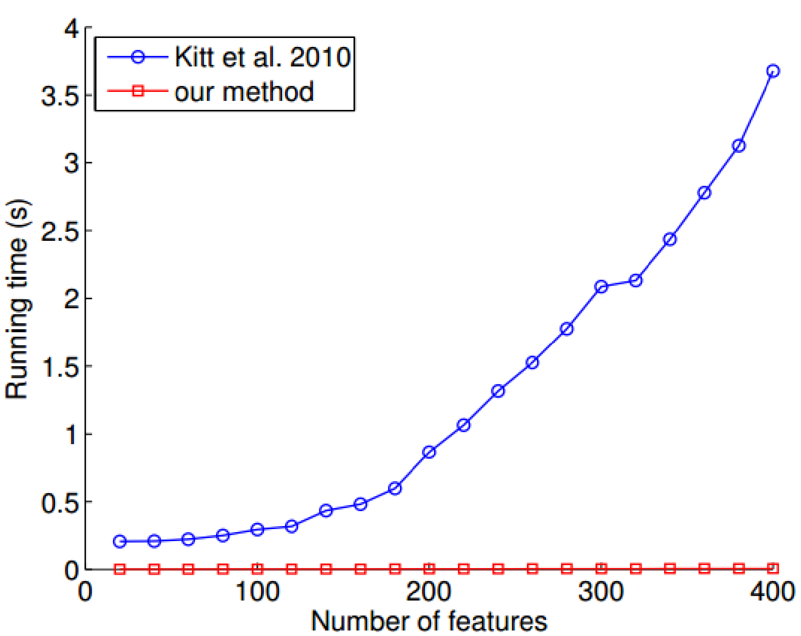
## Computational Time

Kitt et al. required inverting large matrices that grew linearly with features




| Stage | Time |
|---|---|
| Filter | 6.0 ms |
| NMS | 12 ms |
| Matching 1 | 2.8 ms |
| Matching 2 | 10.7 ms |
| Refinement | 5.1 ms |
| Total time | 36.6 ms |

(a) Feature matching




| Stage | Time |
|---|---|
| RANSAC | 3.8 ms |
| Refinement | 0.4 ms |
| Kalman filter | 0.1 ms |
| Total time | 4.3 ms |

(b) Visual odometry

## Odometry Result

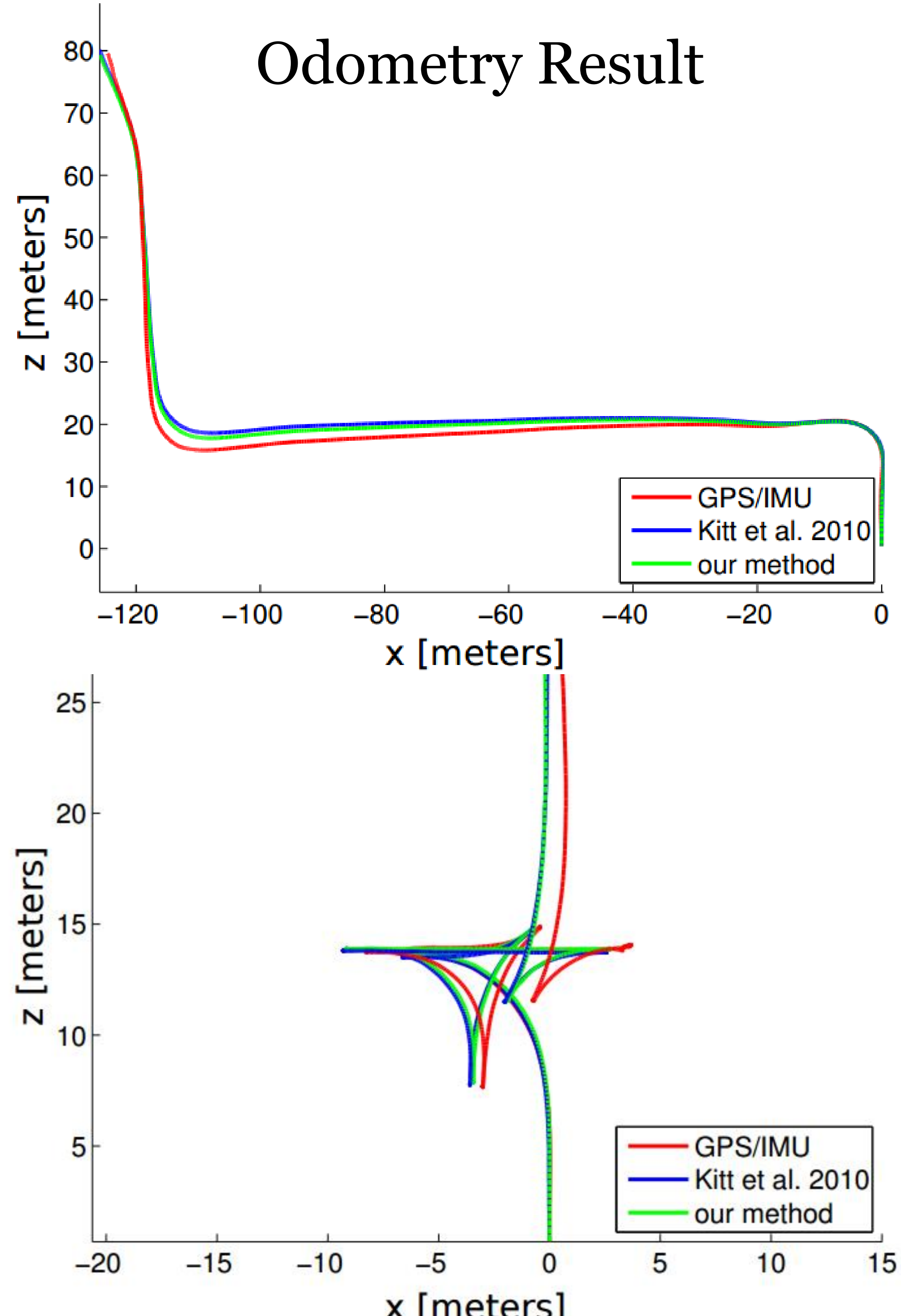



Image from Geiger et al., 2011

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

Paper Review

# Real-Time Stereo Visual Odometry for Autonomous Ground Vehicles

Paper by: Howard, IROS 2008

*Presented by Patrick McGarey*

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

# Problems

- Inertial sensors are prone to drift
- Wheel odometry is unreliable in 'off-road' terrain

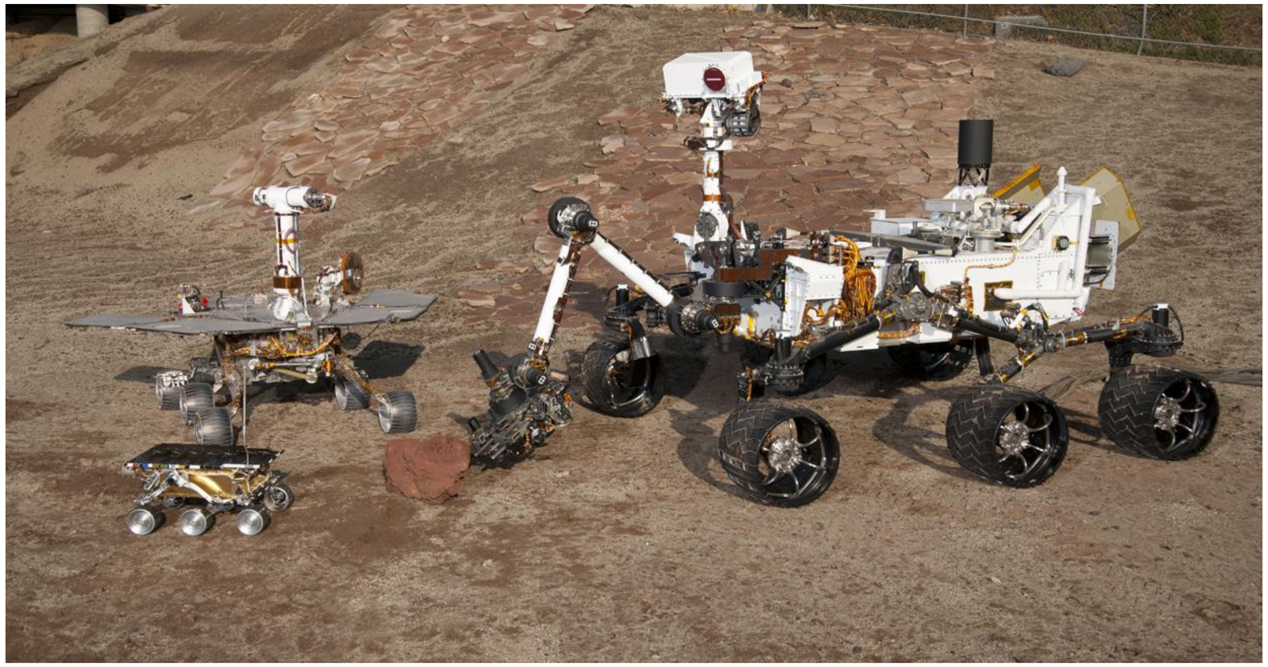


Image from NASA/JPL

- *Historical Context*
  - *At the time (2007-2008), Stereo VO was still a newer topic of investigation*
  - *Inlier detection was very slow*
  - *Stereo navigation had recently been used on the MER Rovers*
  - *The Curiosity rover was about to be launched*

# Method

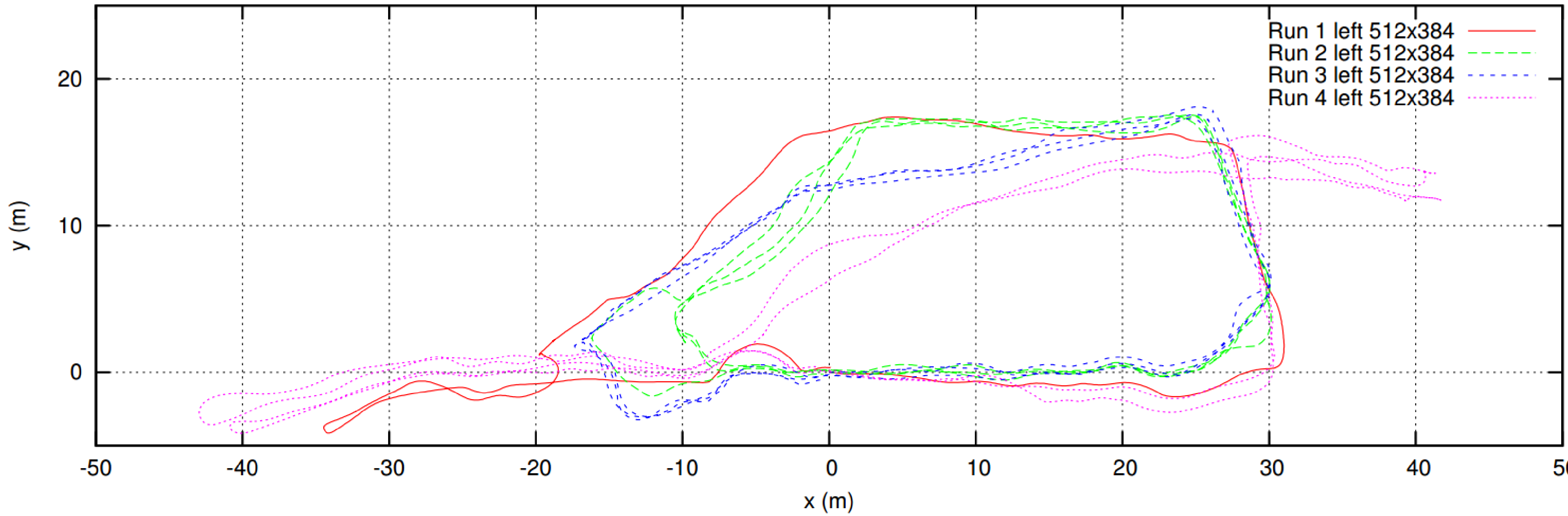
- Construct a VO pipeline that
  - 1) Does not make assumptions of camera motion (no priors)
  - 2) Works on dense disparity images in real time (prior methods were slow)


DARPA LAGR robot, 2008

# Method: Pipeline

- 1) Inputs: rectified, pre-filtered images
- 2) Feature Detection: detect corner features, assign 3D vals from disparity
  - Used Harris corners and FAST features
- 3) Score Matrix: using sum of absolute differences generate feature scores
  - Low score indicates match
- 4) Match Features: use local minima from score to generate features
  - Improves on state-of-the-art computation : from cubic to squared complexity
- 5) Find Inliers: inforce rigid world constraint to reject unlikely features
- 6) Estimate Motion: minimize the reprojection error and output motion

# Results (VO vs. Wheel Odometry)




| | | | | | | | VisOdom | | Encoders | |
|---|---|---|---|---|---|---|---|---|---|---|
| Run | Src. | Size | Frames | Dist. | Fail | Time | 2D RMS err. | 3D RMS err. | 2D RMS err. | 3D RMS err. |
| 1 | left | 256x192 | 2334 | 166m | 10 | 4.8ms | 0.462m (0.28%) | 0.484m (0.29%) | 2.575m (1.55%) | 3.356m (2.02%) |
| 2 | left | 256x192 | 3936 | 335m | 5 | 7.8ms | 0.960m (0.29%) | 1.501m (0.45%) | 7.196m (2.15%) | 8.415m (2.51%) |
| 3 | left | 256x192 | 3467 | 360m | 13 | 7.1ms | 1.323m (0.37%) | 1.742m (0.48%) | 12.053m (3.35%) | 13.689m (3.81%) |
| 4 | left | 256x192 | 4114 | 406m | 39 | 7.4ms | 2.015m (0.50%) | 3.038m (0.75%) | 16.349m (4.03%) | 17.906m (4.41%) |
| 1 | right | 256x192 | 2548 | 166m | 1 | 7.2ms | 1.358m (0.82%) | 1.654m (0.99%) | 2.575m (1.55%) | 3.356m (2.02%) |
| 2 | right | 256x192 | 4067 | 335m | 0 | 7.0ms | 1.436m (0.43%) | 1.559m (0.47%) | 7.196m (2.15%) | 8.415m (2.51%) |
| 3 | right | 256x192 | 3681 | 360m | 5 | 8.6ms | 0.738m (0.21%) | 1.088m (0.30%) | 12.053m (3.35%) | 13.689m (3.81%) |
| 4 | right | 256x192 | 4235 | 406m | 11 | 8.5ms | 1.154m (0.28%) | 1.642m (0.40%) | 16.349m (4.03%) | 17.906m (4.41%) |
| 1 | left | 512x384 | 2334 | 166m | 0 | 17.7ms | 0.145m (0.09%) | 0.434m (0.26%) | 2.575m (1.55%) | 3.356m (2.02%) |
| 2 | left | 512x384 | 3936 | 335m | 3 | 20.9ms | 0.317m (0.09%) | 0.758m (0.23%) | 7.196m (2.15%) | 8.415m (2.51%) |
| 3 | left | 512x384 | 3467 | 360m | 5 | 16.1ms | 0.630m (0.18%) | 1.013m (0.28%) | 12.053m (3.35%) | 13.689m (3.81%) |
| 4 | left | 512x384 | 4114 | 406m | 13 | 22.8ms | 0.965m (0.24%) | 1.364m (0.34%) | 16.349m (4.03%) | 17.906m (4.41%) |
| 1 | right | 512x384 | 2548 | 166m | 1 | 19.0ms | 0.249m (0.15%) | 1.485m (0.89%) | 2.575m (1.55%) | 3.356m (2.02%) |
| 2 | right | 512x384 | 4067 | 335m | 1 | 14.6ms | 0.437m (0.13%) | 0.736m (0.22%) | 7.196m (2.15%) | 8.415m (2.51%) |
| 3 | right | 512x384 | 3681 | 360m | 0 | 17.2ms | 0.531m (0.15%) | 0.601m (0.17%) | 12.053m (3.35%) | 13.689m (3.81%) |
| 4 | right | 512x384 | 4235 | 406m | 8 | 13.1ms | 0.534m (0.13%) | 1.033m (0.25%) | 16.349m (4.03%) | 17.906m (4.41%) |

Image from Howard, 2008

# Supplemental Slide

# Primer on Visual Odometry

**Algorithm 3. VO from 3-D-to-2-D Correspondences.**

1) Do only once:
1.1) Capture two frames $I_{k-2}, I_{k-1}$
1.2) Extract and match features between them
1.3) Triangulate features from $I_{k-2}, I_{k-1}$
2) Do at each iteration:
2.1) Capture new frame $I_k$
2.2) Extract features and match with previous frame $I_{k-1}$
2.3) Compute camera pose (PnP) from 3-D-to-2-D matches
2.4) Triangulate all new feature matches between $I_k$ and $I_{k-1}$
2.5) Iterate from 2.1).

Image from Scaramuzza and Fraundorfer, 2011

# References


- Scaramuzza, Davide, and Friedrich Fraundorfer. "Visual odometry [tutorial]." Robotics & Automation Magazine, IEEE 18.4 (2011): 80-92.
- Zhang, Ji, and Sanjiv Singh. "Visual-lidar odometry and mapping: Low-drift, robust, and fast." Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015.
- Geiger, Andreas, Julius Ziegler, and Christoph Stiller. "Stereoscan: Dense 3d reconstruction in real-time." Intelligent Vehicles Symposium (IV), 2011 IEEE. IEEE, 2011.
- Howard, Andrew. "Real-time stereo visual odometry for autonomous ground vehicles." Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. IEEE, 2008.