Visual-Lidar Odometry and Mapping: Low-drift, Robust, and Fast

Ji Zhang and Sanjiv Singh
&& Yu Zhang @ iMorpheus.ai

Friday 08/12/2017
12:00PM (GMT+8)

无人驾驶技术交流群

Abstract

- State-of-art odometry methods & comparison
- V-LOAM method
- Results
- Q & A

iMorpheus.ai

Single-sensor Odometry

- Visual odometry weakness:
  - requires moderate lighting condition
  - requires sufficient distinct visual features
- Lidar odometry weakness:
  - moving Lidar involves motion distortion (low scan rate)
  - scan matching fails due to planar area

RGB-D visual odometry

- reference [8]-[12]
- possibly wasting large areas in visual images without depth coverage

iMorpheus.ai

Motion distortion by Lidar

- Scherer [14]: estimate motion by stereo camera + IMU
- Droeschel [15]: multi-camera + scan matching
- VLOAM: only one camera, model drift as linear motion

iMorpheus.ai

3D Lidar only

- Tong [16]: motion modeled with constant velocity and Gaussian processes; however, requires dense point cloud

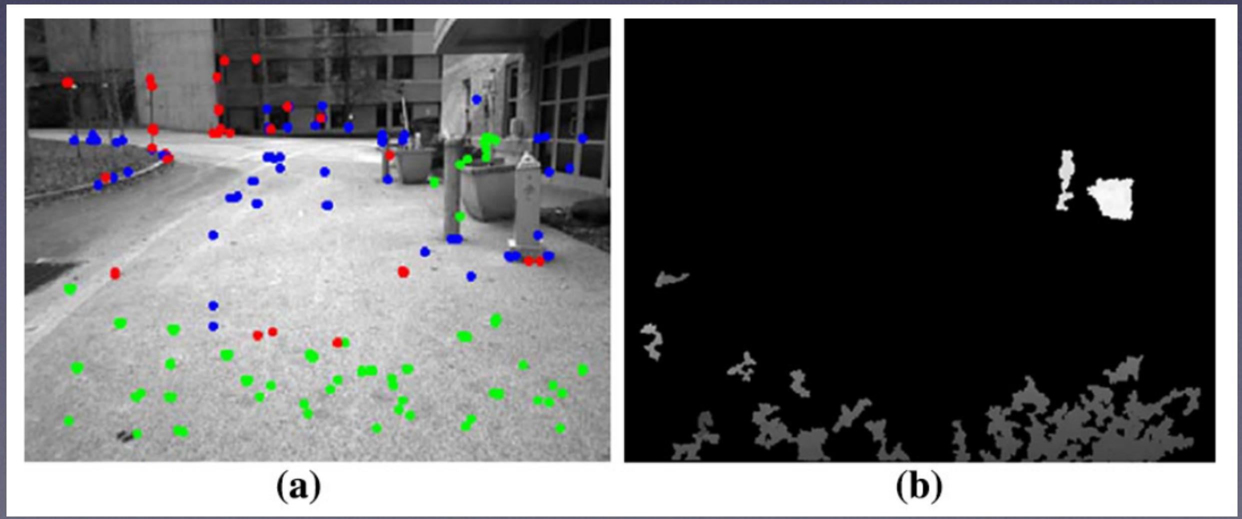
- Bosse and Lot [17], [18]: batch optimization processing segmented data; offline method

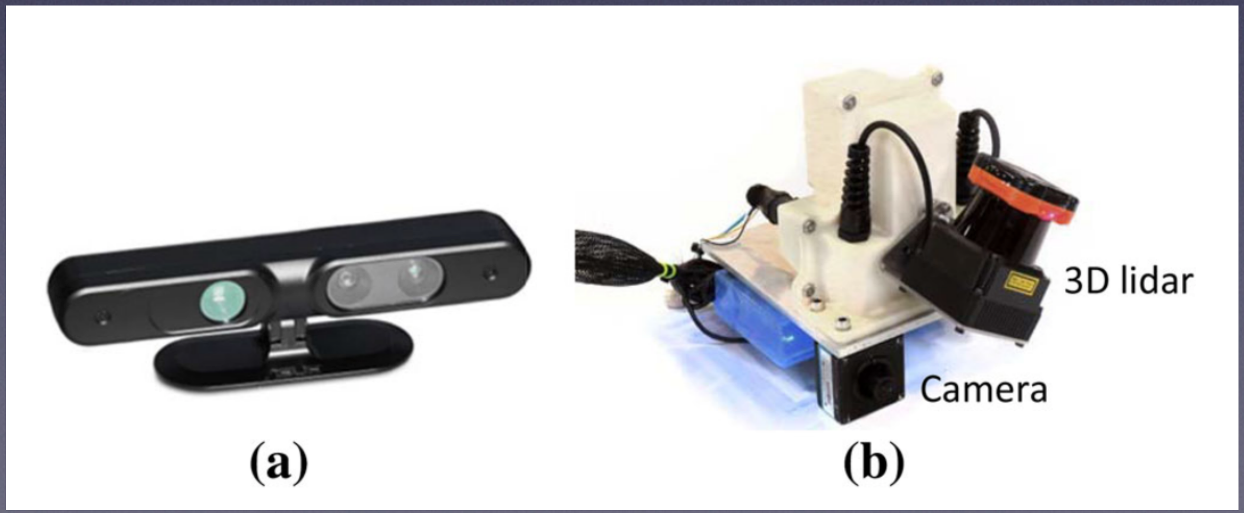
iMorpheus.ai

Visual odometry - DEMO

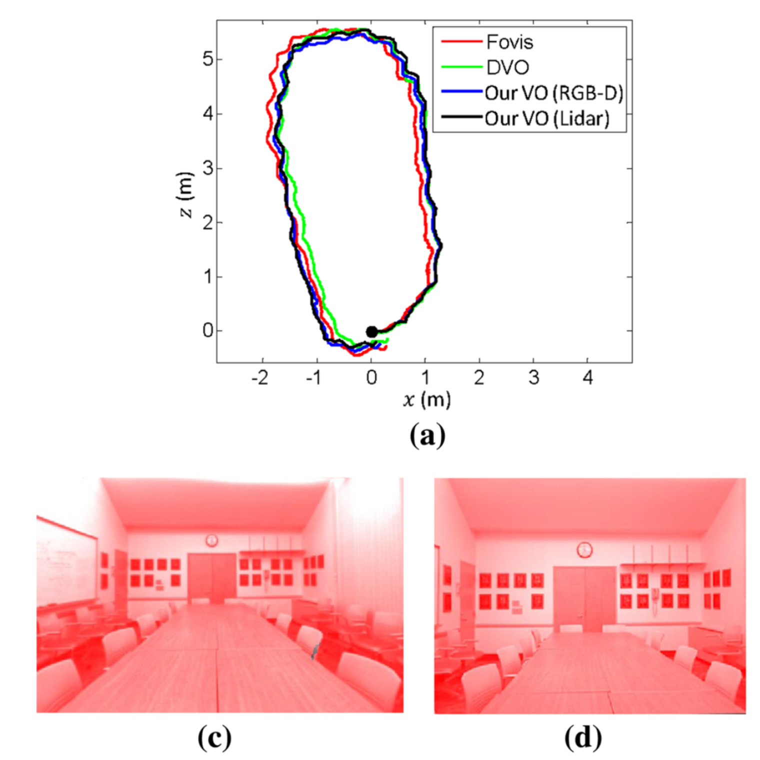
- [19] J. Zhang, M. Kaess, and S. Singh. Real-time Depth Enhanced Monocular Odometry. Sept. 2014.

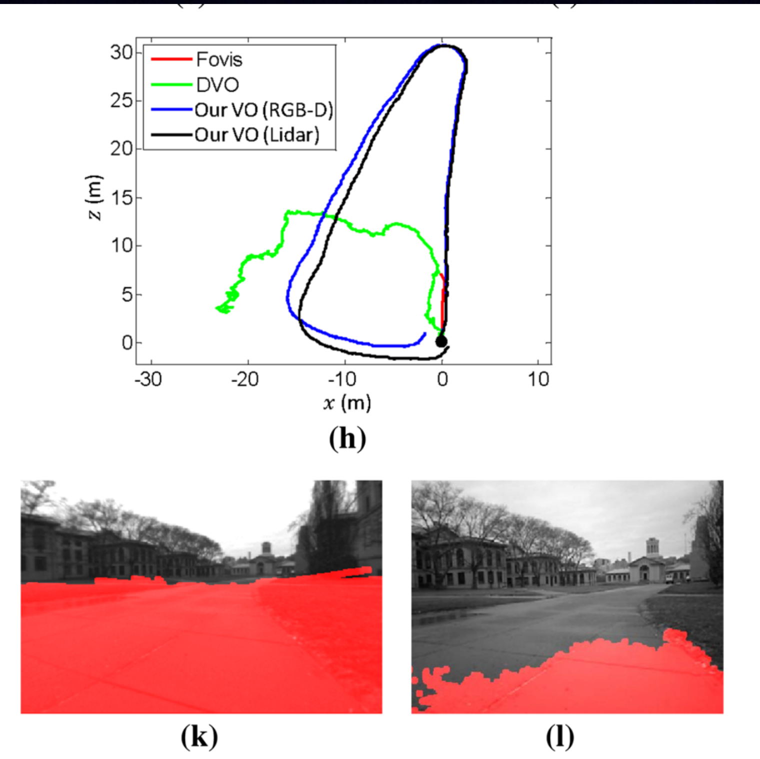
- Depth from: RGB-D cameras or 3D Lidar





DEMO results

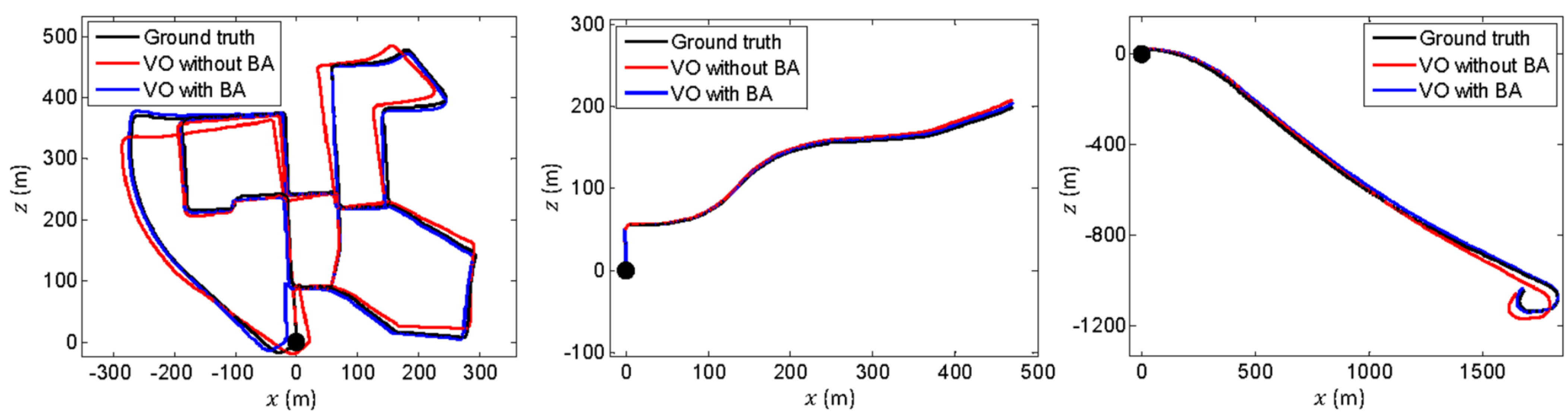




Envir-onment	Dist-ance	Relative position error			
		Fovis	DVO	Our VO (RGB-D)	Our VO (Lidar)
Room	16m	2.72%	1.87%	2.14%	2.06%
Lobby	56m	5.56%	8.36%	1.84%	1.79%
Road	87m	13.04%	13.60%	1.53%	0.79%
Lawn	86m	9.97%	32.07%	3.72%	1.73%

iMorpheus.ai

DEMO results



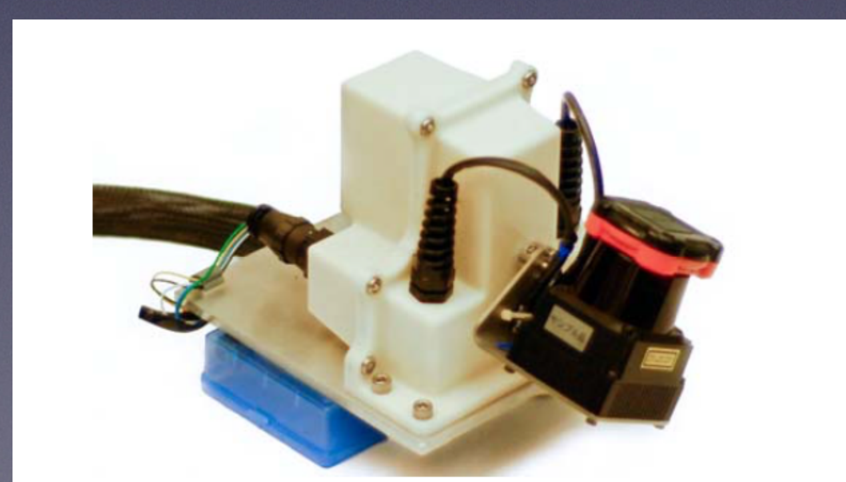
(a) (b) (c)

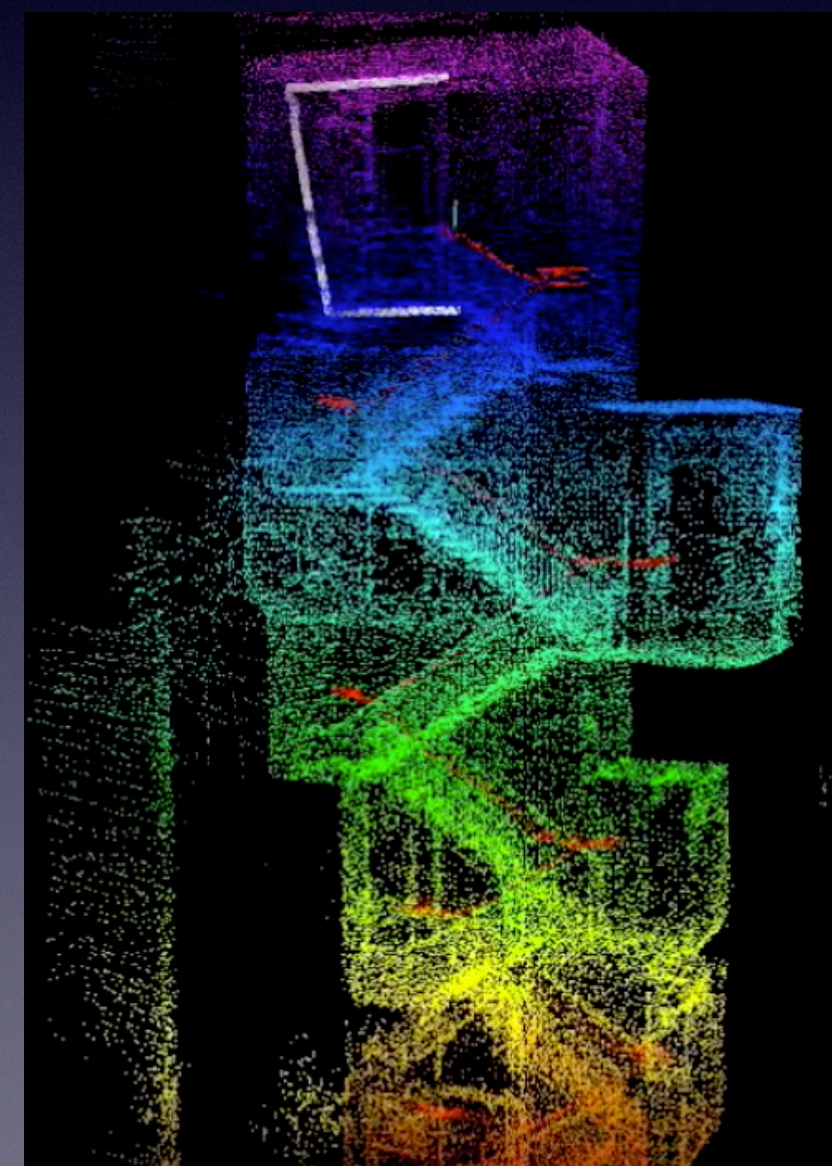
Data no.	Configuration		Mean relative position error
	Distance	Environment	
0	3714m	Urban	1.05%
1	4268m	Highway	1.87%
2	5075m	Urban + Country	0.93%
3	563m	Country	0.99%
4	397m	Country	1.23%
5	2223m	Urban	1.04%
6	1239m	Urban	0.96%
7	695m	Urban	1.16%
8	3225m	Urban + Country	1.24%
9	1717m	Urban + Country	1.17%
10	919m	Urban + Country	1.14%

iMorpheus.ai

Lidar odometry - LOAM

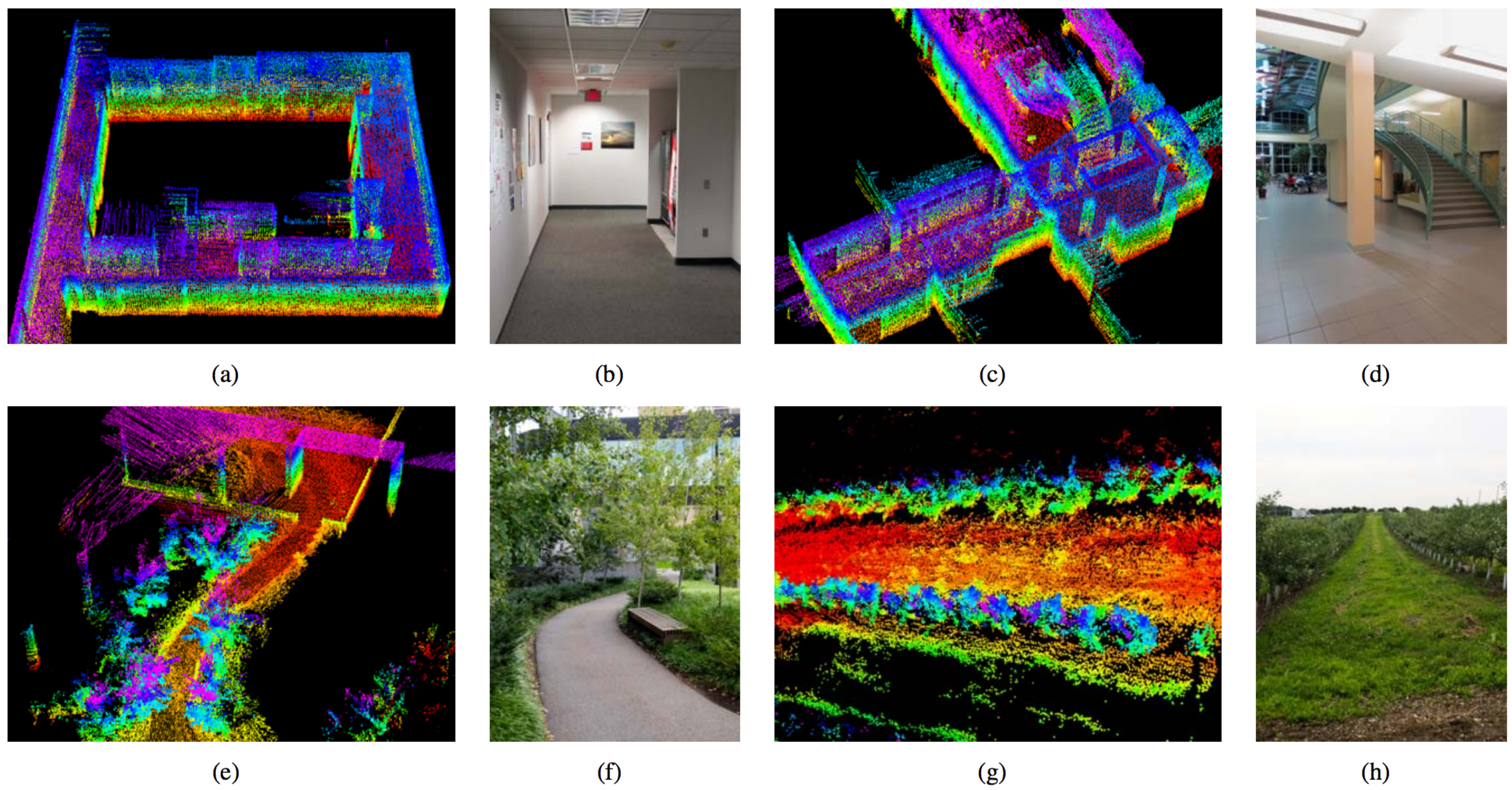
- [20] J. Zhang and S. Singh. LOAM: Lidar Odometry and Mapping in Real-time. July 2014
- requires smooth motion
- relies on IMU to compensate high frequency motion





iMorpheus.ai

LOAM results



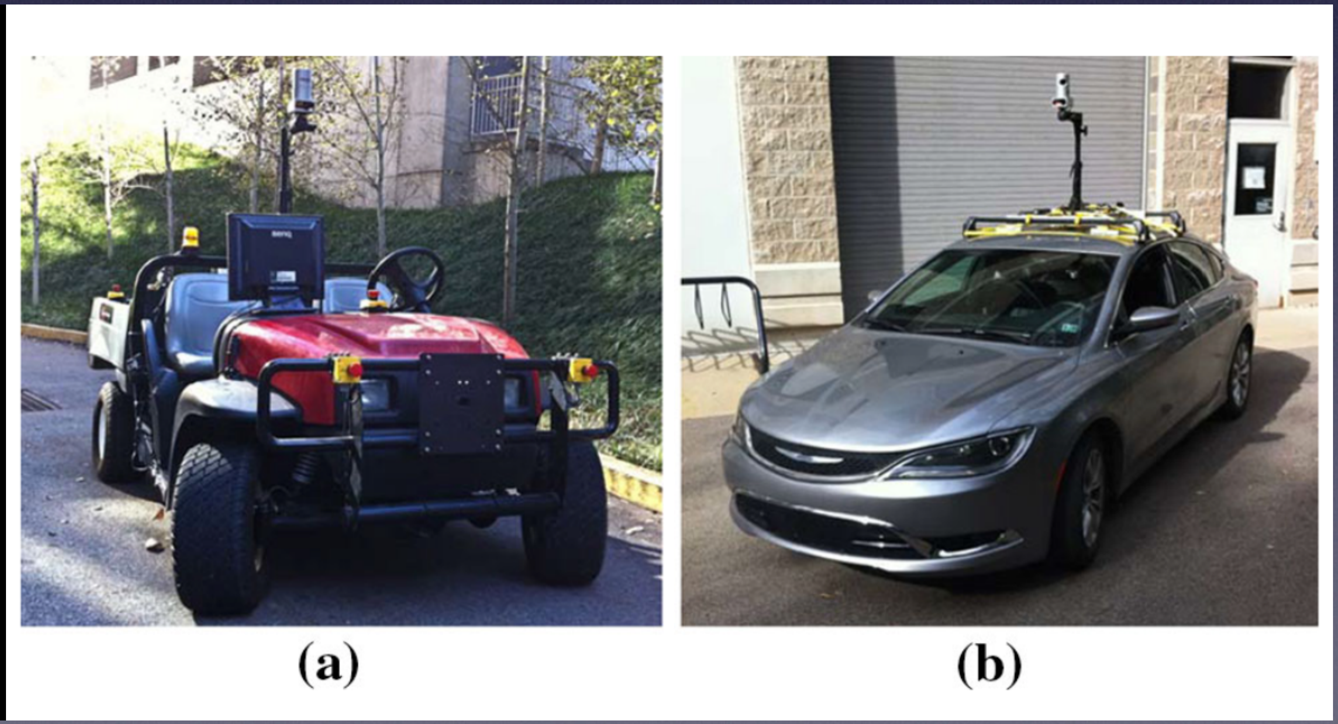
MOTION ESTIMATION ERRORS WITH/WITHOUT USING IMU.

Environment	Distance	Error		
		IMU	Ours	Ours+IMU
Corridor	32m	16.7%	2.1%	0.9%
Lobby	27m	11.7%	1.7%	1.3%
Vegetated road	43m	13.7%	4.4%	2.6%
Orchard	51m	11.4%	3.7%	2.1%

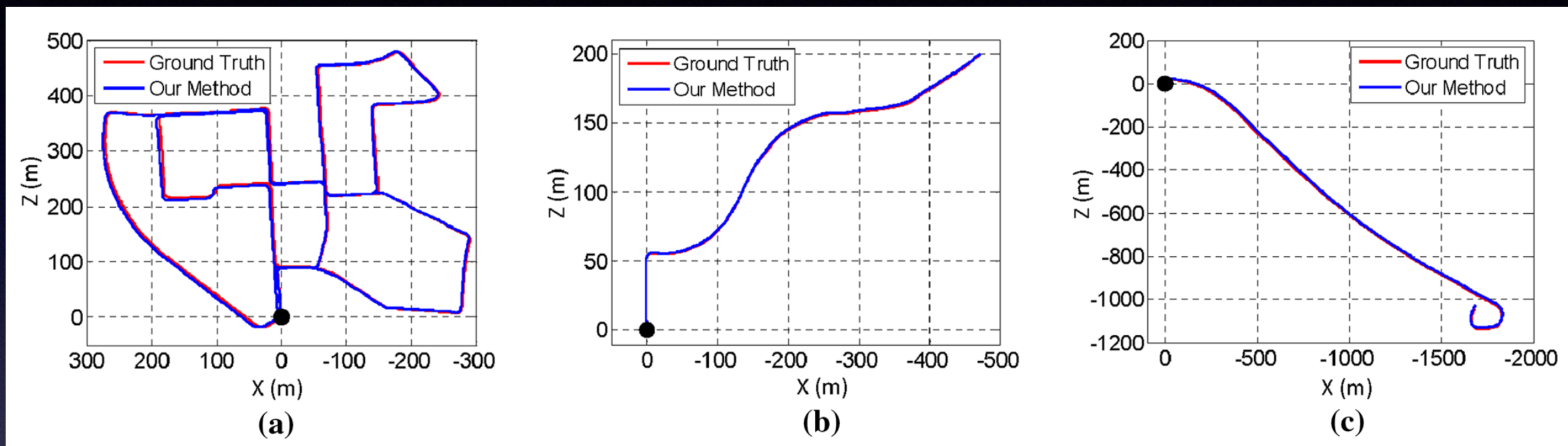
iMorpheus.ai

LOAM - Velodyne

- J. Zhang and S. Singh. Low-drift and Real-time Lidar Odometry and Mapping. Autonomous Robots. vol. 41, no. 2, pp. 401–416, 2017. (PDF) (Video)



LOAM KITTI (vs DEMO)

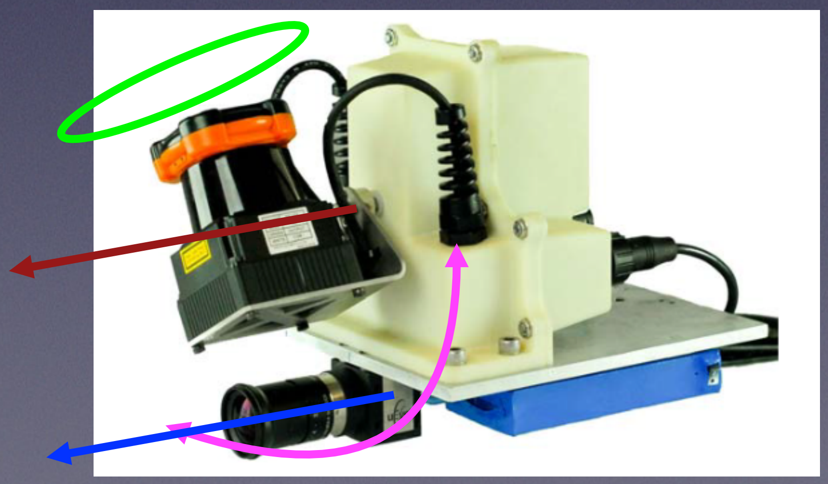


Data no.	Configuration		Mean relative position error (%)
	Distance (m)	Environment	
0	3714	Urban	0.78
1	4268	Highway	1.43
2	5075	Urban + Country	0.92
3	563	Country	0.86
4	397	Country	0.71
5	2223	Urban	0.57
6	1239	Urban	0.65
7	695	Urban	0.63
8	3225	Urban + Country	1.12
9	1717	Urban + Country	0.77
10	919	Urban + Country	0.79

Data no.	Configuration		Mean relative position error
	Distance	Environment	
0	3714m	Urban	1.05%
1	4268m	Highway	1.87%
2	5075m	Urban + Country	0.93%
3	563m	Country	0.99%
4	397m	Country	1.23%
5	2223m	Urban	1.04%
6	1239m	Urban	0.96%
7	695m	Urban	1.16%
8	3225m	Urban + Country	1.24%
9	1717m	Urban + Country	1.17%
10	919m	Urban + Country	1.14%

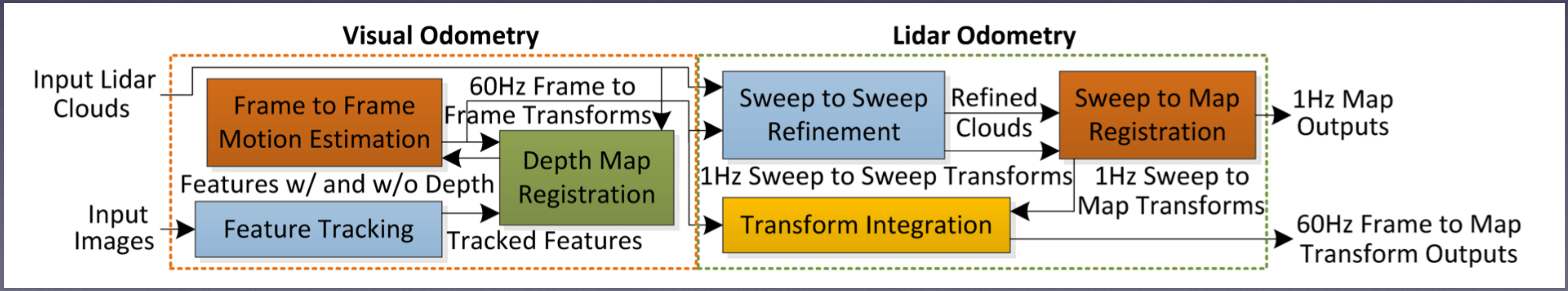
V-LOAM

- online method for ego-motion estimation
- visual odometry + register point cloud by Lidar
- improves robustness against
  - aggressive motion
  - lack of optical texture

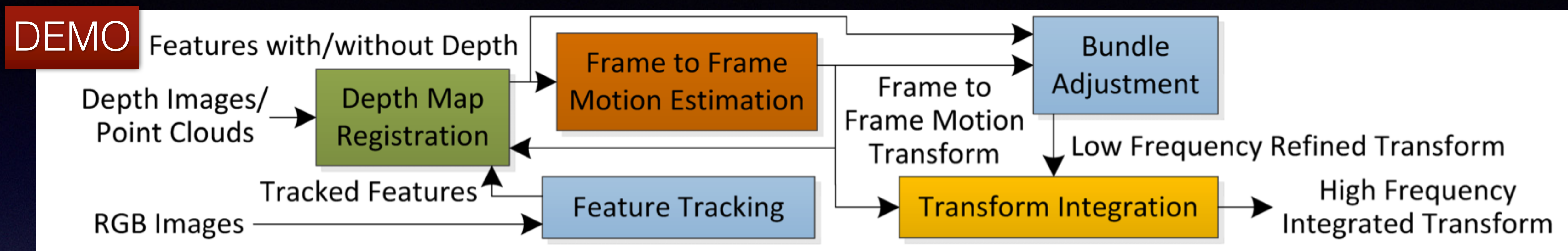


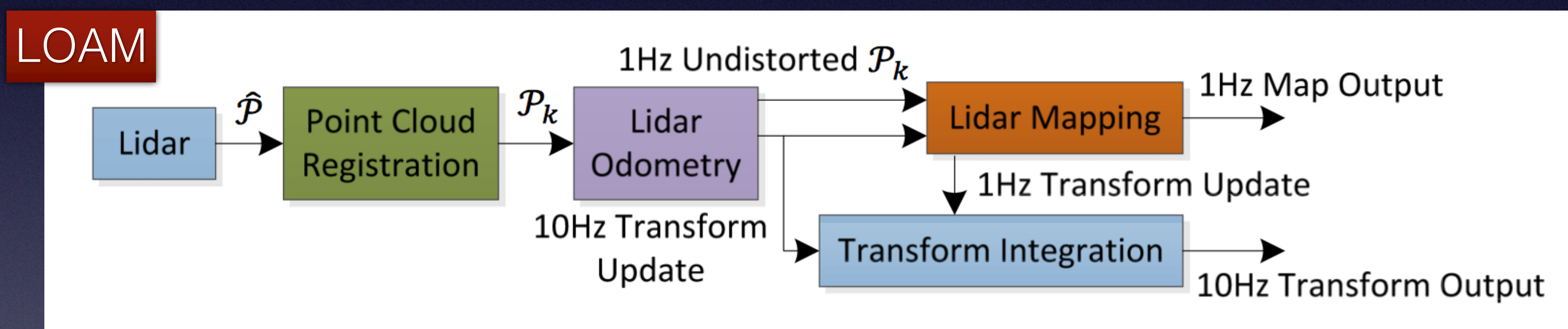
2-step process

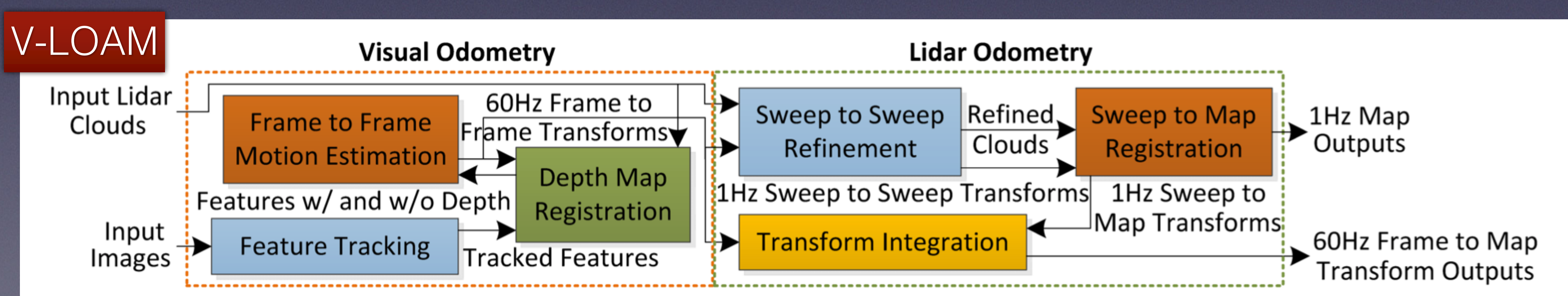
- Visual odometry @60Hz to estimate motion
- Lidar odometry @1Hz to refine motion estimates and remove distortion in point cloud for mapping



Based on DEMO & LOAM







Coordinate System

- Sensor Coordinate System {S}

The x-axis points to the left, the y-axis points upward, and the z-axis points forward coinciding with the camera principal axis.

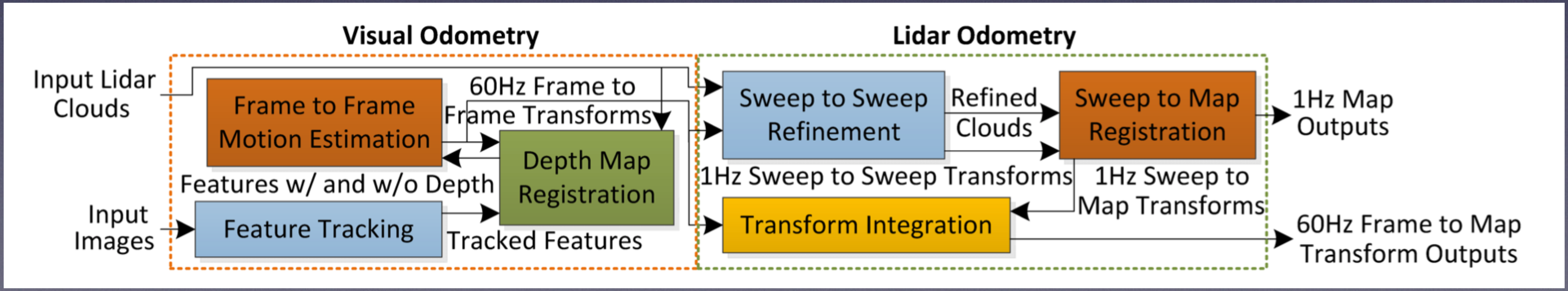
The x-axis points to the left, the y-axis points upward, and the z-axis points forward coinciding with the camera principal axis.

- World Coordinate System {W}

- **Problem**: Given visual images and lidar could perceived in {S}, determine poses of {S} w.r.t. {W} and build a map of the traversed environment in {W}

Visual Odometry

- **Feature Tracking**: extracts and matches visual features btw consecutive frames
- **Depth Map Registration**: registers lidar clouds on a local depth map, associates depth to the visual features
- **Frame to Frame Motion Estimation**: computes motion estimates from visual features



Motion Estimation

- Feature coordinate

$$ {}^S\boldsymbol{X}_i^k = [{}^S x_i^k,\ {}^S y_i^k,\ {}^S z_i^k]^T $$

- Unknown distance, normalized

$$ {}^S\bar{\boldsymbol{X}}_i^k = [{}^S\bar{x}_i^k,\ {}^S\bar{y}_i^k,\ {}^S\bar{z}_i^k]^T, \text{ where } ||{}^S\bar{\boldsymbol{X}}_i^k|| = 1 $$

- Motion Formulated 6-DOF

$$ {}^S\boldsymbol{X}_i^k = \mathbf{R}\ {}^S\boldsymbol{X}_i^{k-1} + \boldsymbol{T}. \qquad (1) $$

- Feature with known distance

$$ ({}^S\bar{z}_i^k \boldsymbol{R}_1 - {}^S\bar{x}_i^k \boldsymbol{R}_3)\,{}^S\boldsymbol{X}_i^{k-1} + {}^S\bar{z}_i^k T_1 - {}^S\bar{x}_i^k T_3 = 0, \qquad (2) $$

$$ ({}^S\bar{z}_i^k \boldsymbol{R}_2 - {}^S\bar{y}_i^k \boldsymbol{R}_3)\,{}^S\boldsymbol{X}_i^{k-1} + {}^S\bar{z}_i^k T_2 - {}^S\bar{y}_i^k T_3 = 0. \qquad (3) $$

- Feature with unknown distance

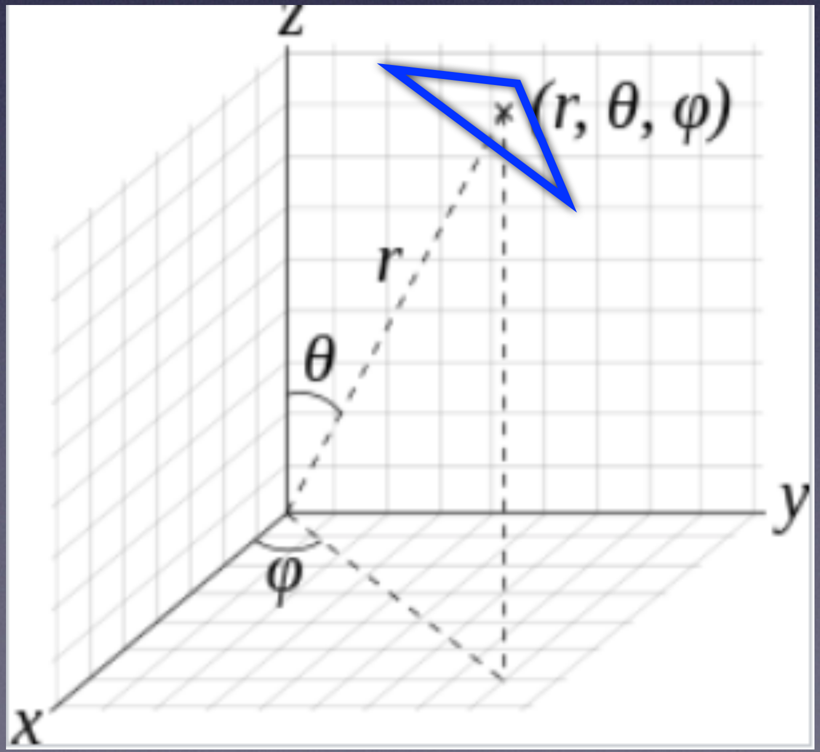
$$ \begin{bmatrix} -{}^S\bar{y}_i^k T_3 + {}^S\bar{z}_i^k T_2 \\ {}^S\bar{x}_i^k T_3 - {}^S\bar{z}_i^k T_1 \\ -{}^S\bar{x}_i^k T_2 + {}^S\bar{y}_i^k T_1 \end{bmatrix} \mathbf{R}\ {}^S\bar{\boldsymbol{X}}_i^{k-1} = 0. \qquad (4) $$

- Solve by Levenberg-Marquardt method

iMorpheus.ai

Depth Map Registration

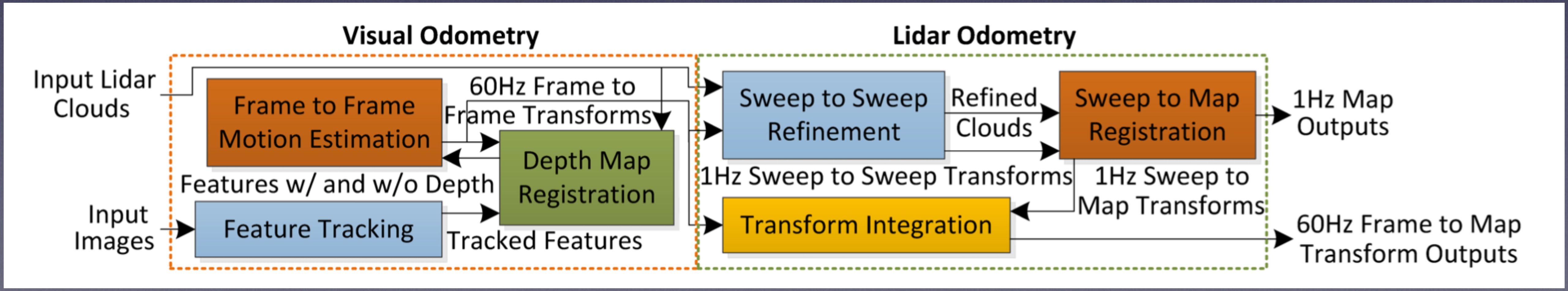
- New points added, old points aged
- Downsized, to constant point density
- Projected to last image transformed
- Points by spherical coordinates
- Stored in 2D KD-tree by 2 angular coordinates
- Associate depth to feature by projecting it to planar patch of 3 closest* points
- Triangulate tracked features w/o distance



Lidar Odometry

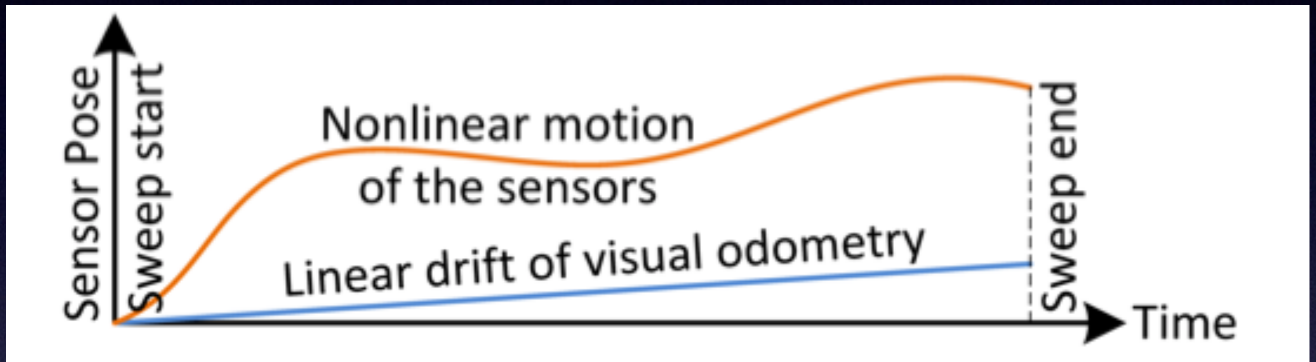
- **Sweep to Sweep Refinement:** matches point clouds between sweeps to refine motion estimates

- **Sweep to Map Registration:** matches and registers point clouds on the map



Sweep to Sweep Refinement

- Visual Odometry drift modeled as linear motion within a sweep (1s)



$$\boldsymbol{T}'_i = \boldsymbol{T}'(t_i - t^m)/(t^{m+1} - t^m). \quad (5)$$

- Lidar cloud extract geometric features and matching
    - Edge point: find two closest edge points forms an edge
    - Planar point: find three closest planar points forms a local planar patch

$$f({}^S\boldsymbol{X}_i^m, \boldsymbol{T}'_i) = d_i, \quad (6)$$

- minimize total distances of all edge & planar points

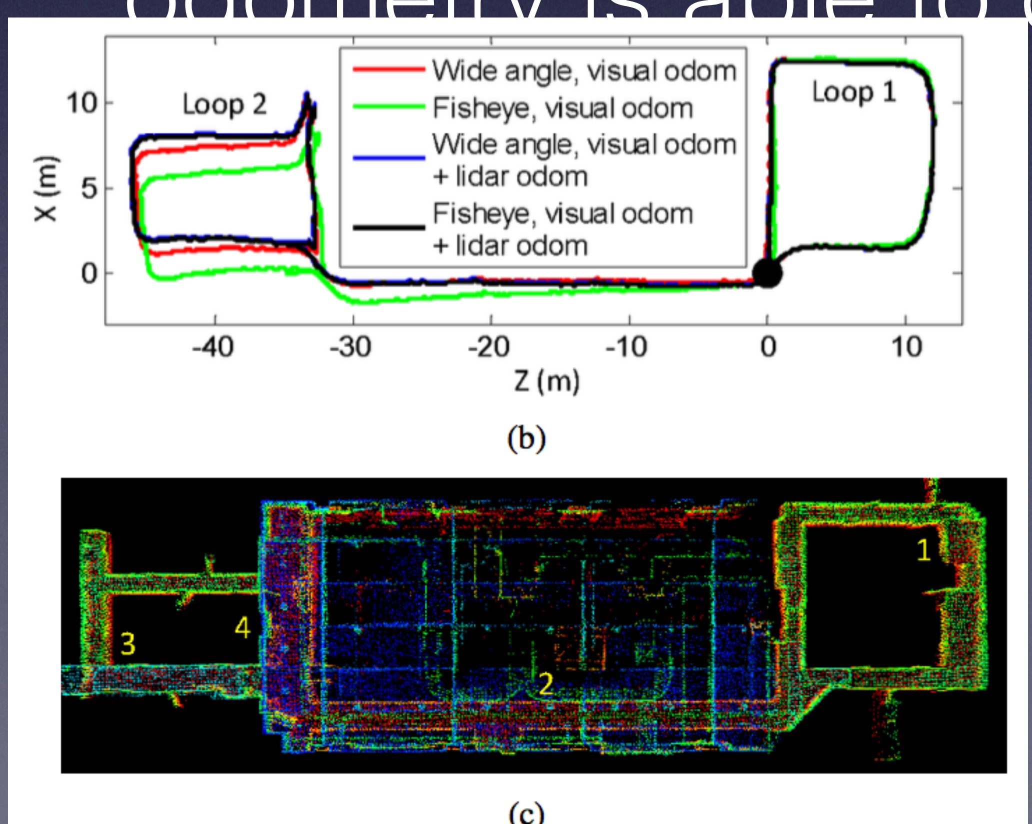
Sweep to Map Registration

- Merge the previous map with current point cloud to build new map
- scan matching via an iterative closest point method
- publish sensor pose transform in world coordinate system {W}

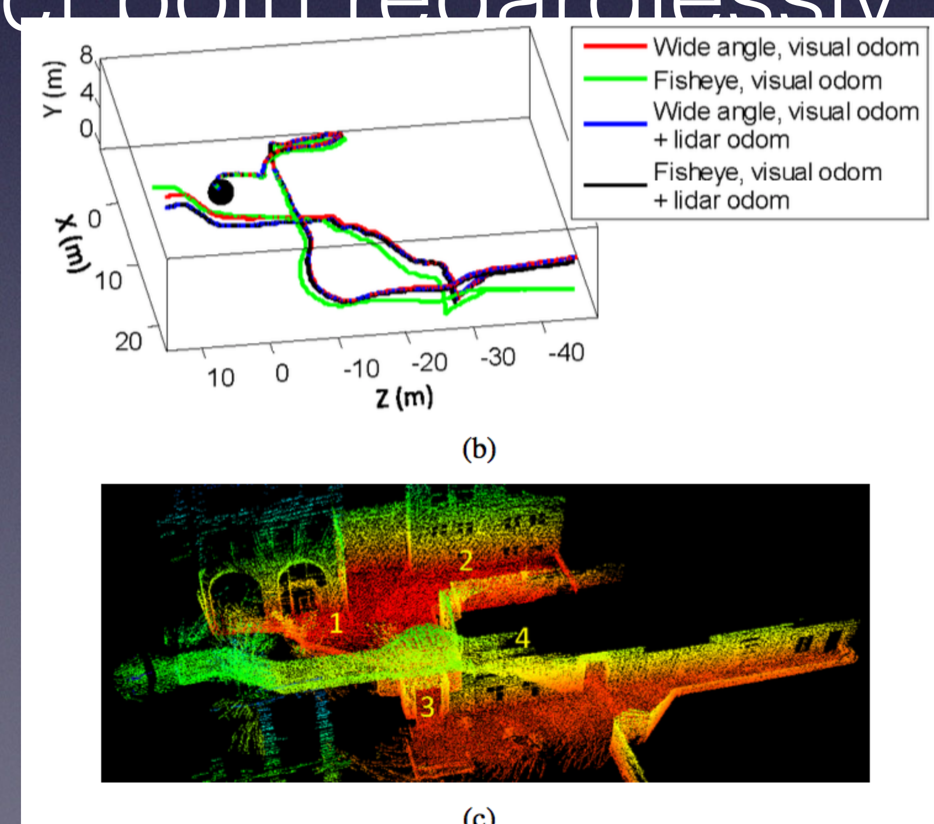
iMorpheus.ai

Experiment - Accuracy

- Wide angel / FishEye camera + 3D lidar
- Fisheye camera results in faster drift, but lidar odometry is able to correct both regardlessly



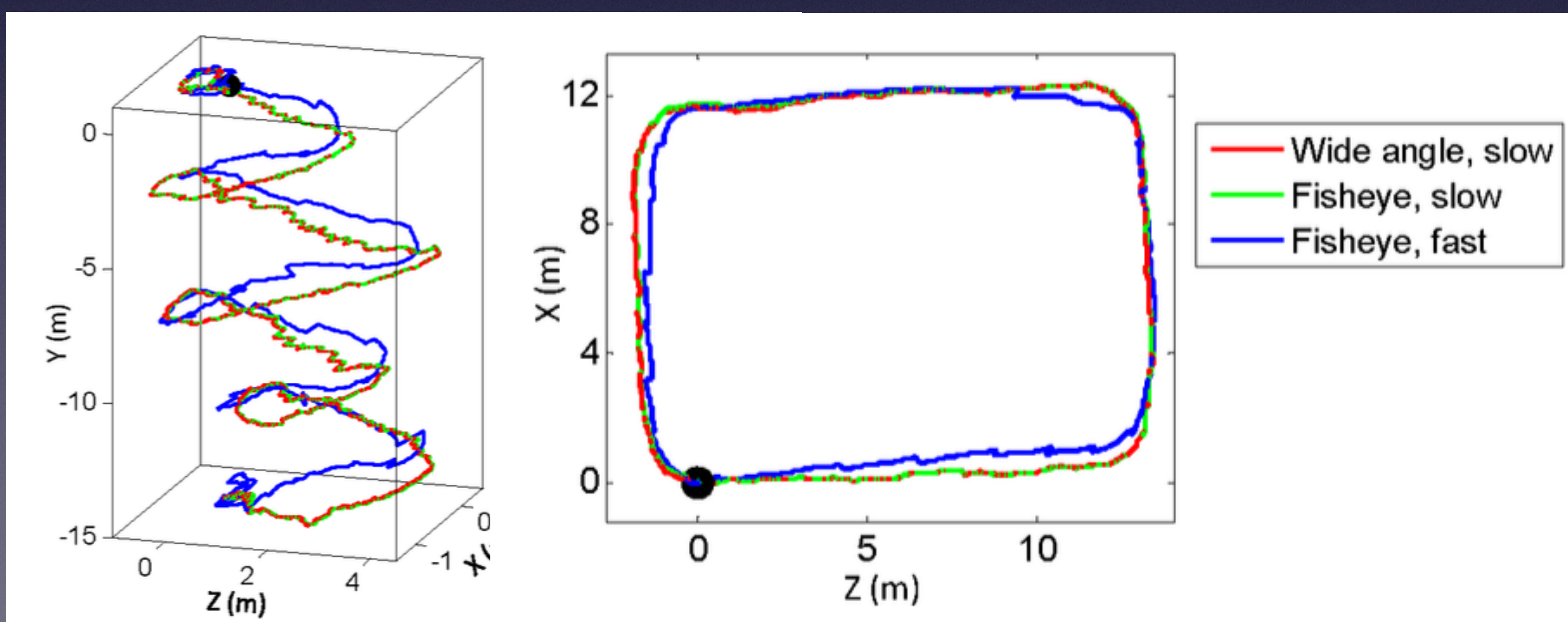
(b)
(c)



(b)
(c)

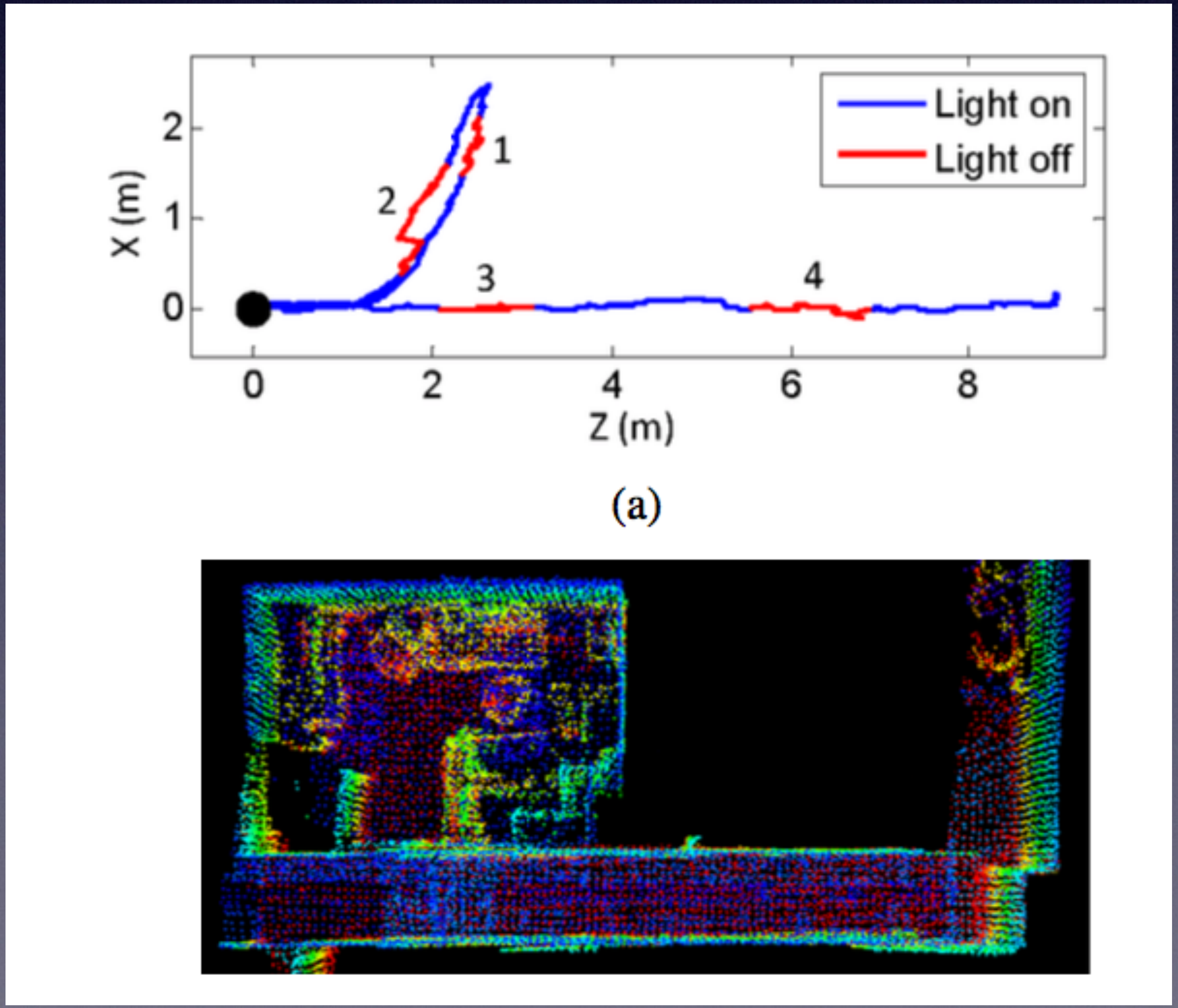
Experiment - Robustness

- Wide angel camera in fast motion: visual feature loose tracking



Experiment - others

- Can handle temporary light outage
- KITTI benchmark
    - avg %0.75 drift



(a)

iMorpheus.ai

iMorpheus Journal Club

Next Friday, 22/12/2017 12:00PM GMT+8

SLAM中简并环境的处理

关键词：visual odometry, LiDAR odometry, degeneracy, state space

Website : **http://imorpheus.ai**
Email Address : **live@imorpheus.ai**

iMorpheus.ai

无人驾驶技术交流群