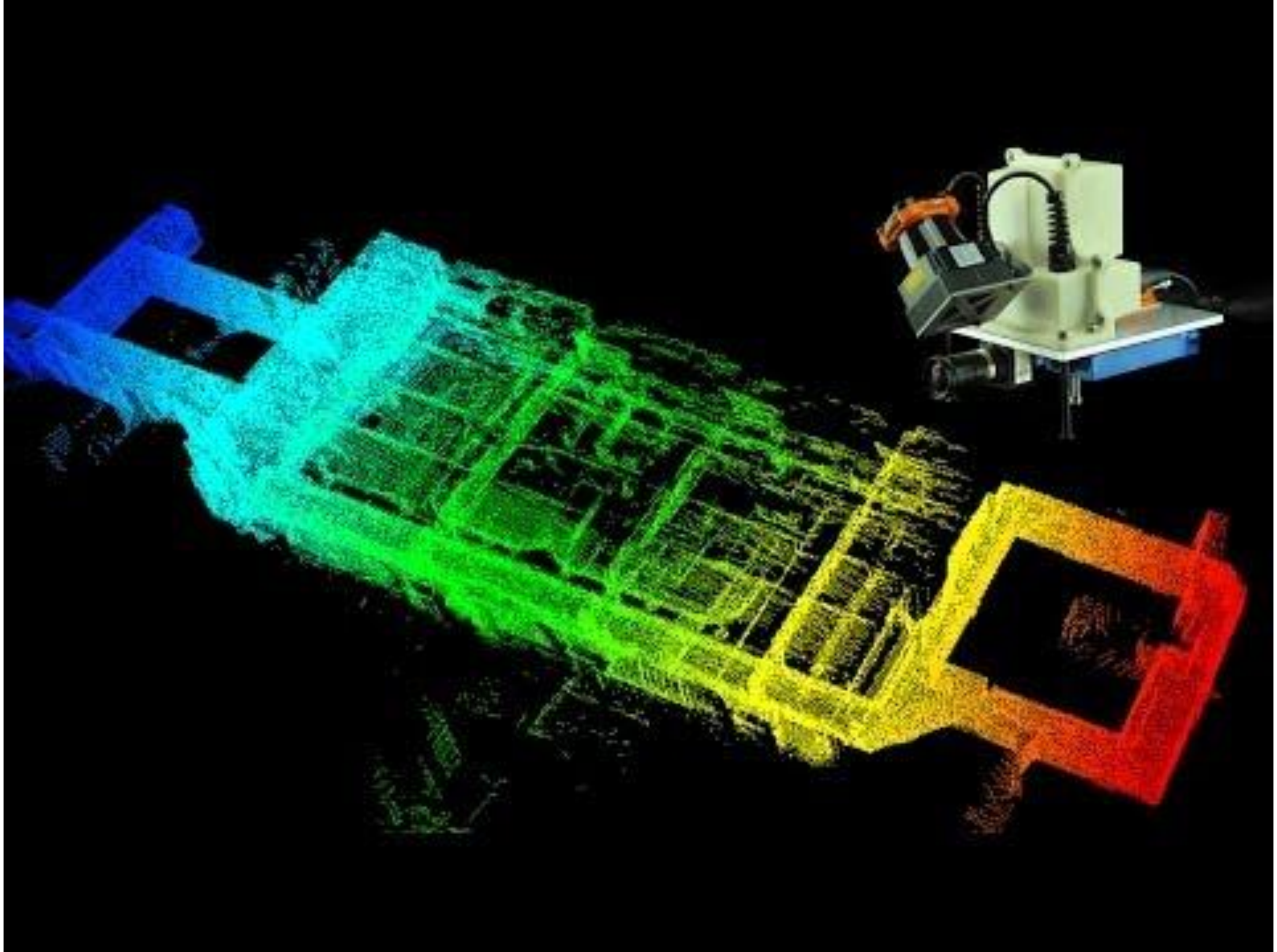
VLOAM - Visual-Lidar Odometry And Mapping

16-833: Robot Localization and Mapping

Spring 2021

Srinivasan Vijayarangan

Ji Zhang and Sanjiv Singh - Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast, International Conference on Robotics and Automation (ICRA), May 26-30, 2015



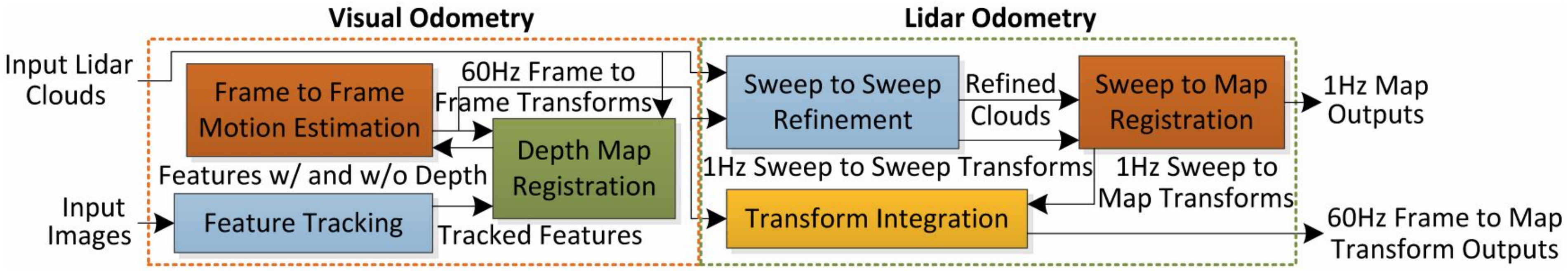
VLOAM Overview

- Related work
- Software system
- Visual odometry
- Lidar odometry
- Experiments

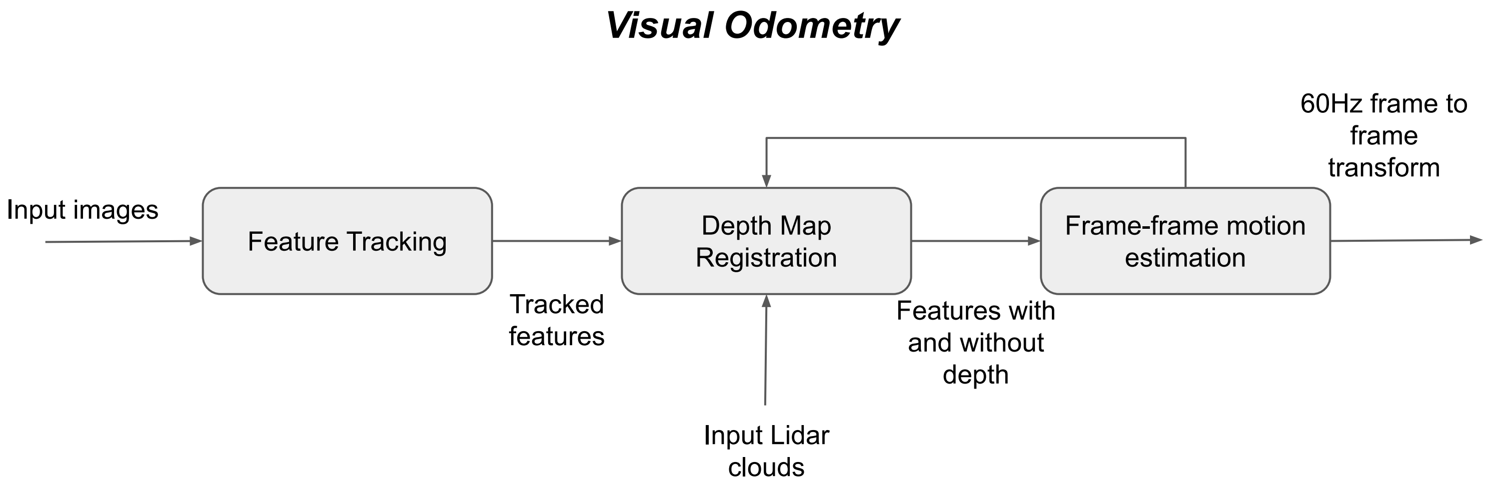
Related Work

- Depth for image pixels
  - Similar to [Engelhard et al., 2011] and [Whelan et al., 2013] using visual images with additional depth
- Motion recovery
  - Different from [Scherer et al., 2012] and [Droeschel et al., 2014] where multiple cameras are employed
- State estimation and map building
  - Different from [Bosse and Zlot, 2012] and [Zlot and Bosse, 2014] who showed state estimation with 3D lidars alone
- Visual and lidar odometry performed separately
  - VO from [Zhang et al., 2014] and LO from [Zhang and Singh 2014]

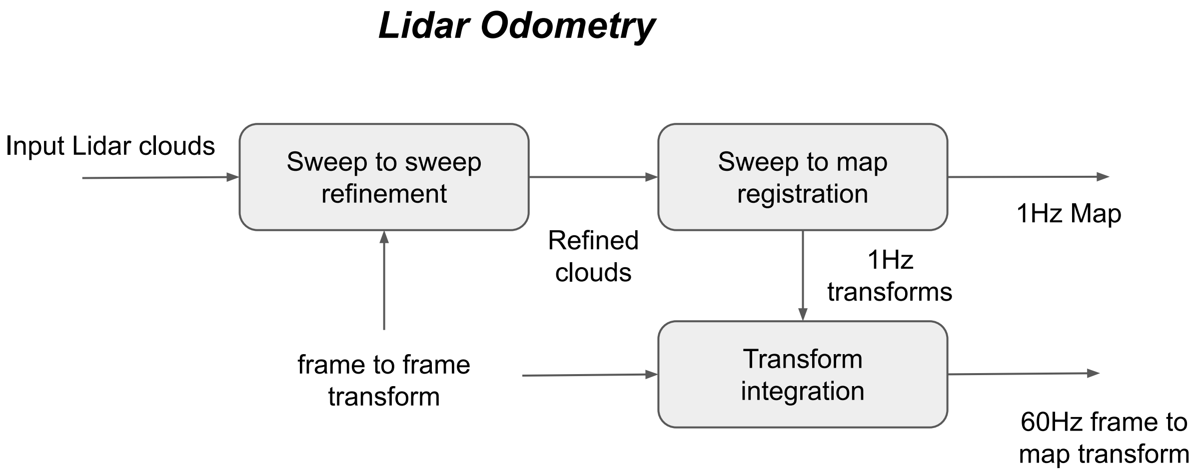
Software System



Software System



Software System



Visual Odometry

Visual Odometry

- Feature Tracking/Extraction
  - Use any feature descriptor to extract features

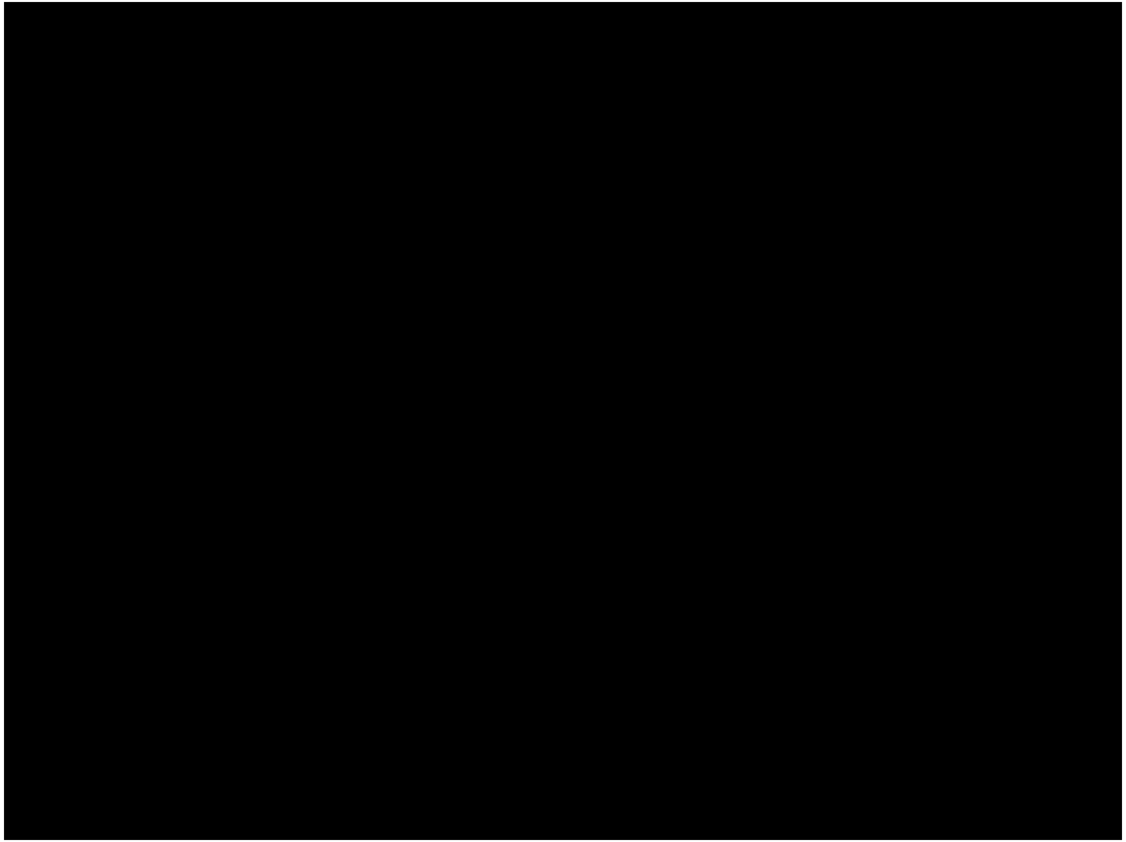
- Depth Map Registration
  - Use a KD-tree to store the 3d points
  - Find 3 closest points
  - Form a planar patch
  - Project to the patch to get distance

J. Zhang, M. Kaess, and S. Singh, “Real-time depth enhanced monocular odometry,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sept. 2014.

Motion Estimation

Problem:

Compute camera motion between two consecutive frames using features with known and unknown depths



Motion Estimation

Known distance

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

Unknown distance

$$ {}^{S}\boldsymbol{X}_i^k = [{}^{S}x_i^k,\ {}^{S}y_i^k,\ {}^{S}z_i^k]^T \qquad {}^{S}\bar{\boldsymbol{X}}_i^k = [{}^{S}\bar{x}_i^k,\ {}^{S}\bar{y}_i^k,\ {}^{S}\bar{z}_i^k]^T $$

$$ ||{}^{S}\bar{\boldsymbol{X}}_i^k|| = 1 $$

$$ {}^{S}\mathbf{X}_\mathbf{i}^\mathbf{k} = {}^{\mathbf{S}}\mathbf{d}_\mathbf{i}^{\mathbf{k}\,\mathbf{S}}\bar{\mathbf{X}}_\mathbf{i}^\mathbf{k} $$

Eliminate the distance term

$$ {}^{S}d_i^{k\,S}\bar{\mathbf{X}}_i^k = \mathbf{R}^{\mathbf{S}}\mathbf{X}_\mathbf{i}^{\mathbf{k}-\mathbf{1}} + \mathbf{T} $$

Motion Estimation

Known distance

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

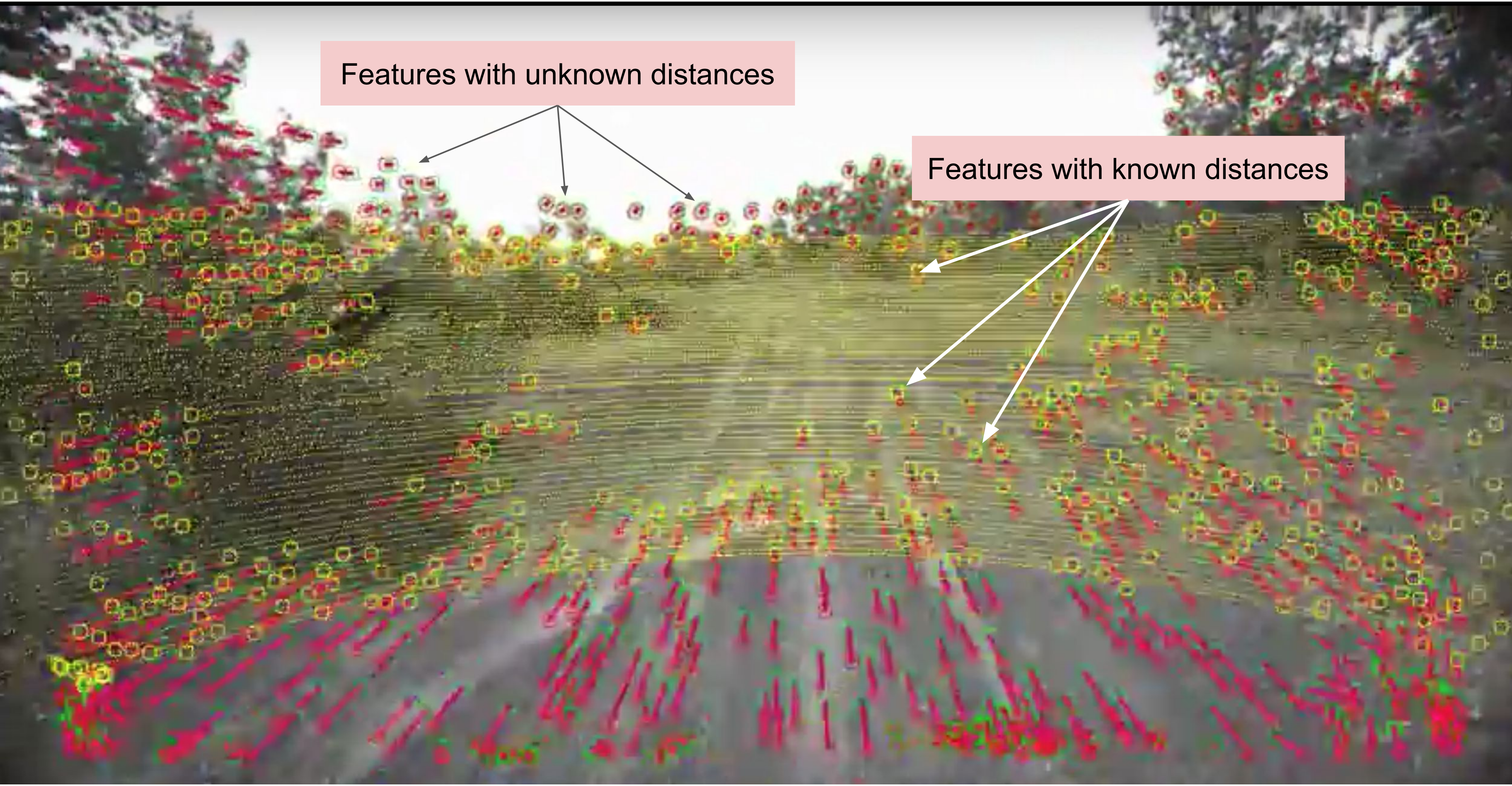
Unknown distance

$$ {}^{S}d_i^k\, {}^{S}\bar{\mathbf{X}}_i^k = \mathbf{R}^{\mathbf{S}}\mathbf{X}_{\mathbf{i}}^{\mathbf{k-1}} + \mathbf{T} $$

Combining first and second row with third row,

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

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


Features with unknown distances
Features with known distances

Motion Estimation

$$ {}^{S}\boldsymbol{X}_i^{k-1} = {}^{S}d_i^{k-1}\,{}^{S}\bar{\boldsymbol{X}}_i^{k-1} \qquad\qquad {}^{S}\boldsymbol{X}_i^{k} = {}^{S}d_i^{k}\,{}^{S}\bar{\boldsymbol{X}}_i^{k} $$

Eliminate the distance terms

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

Optimization

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

Unknowns

- Known distances - 2 equations
- Unknown distances - 1 equation

$$ \theta = [\theta_x,\ \theta_y,\ \theta_z]^T $$

$$ \mathbf{R} = e^{\hat{\theta}} = \mathbf{I} + \frac{\hat{\theta}}{||\theta||}\sin||\theta|| + \frac{\hat{\theta}^2}{||\theta||^2}(1 - \cos||\theta||), $$

$$ \boldsymbol{f}([\boldsymbol{T};\ \theta]) = \epsilon, $$

$$ \mathbf{J} = \partial f / \partial[\boldsymbol{T};\ \theta]. $$

$$ [\boldsymbol{T};\ \theta]^T \leftarrow [\boldsymbol{T};\ \theta]^T - (\mathbf{J}^T\mathbf{J} + \lambda \mathrm{diag}(\mathbf{J}^{\mathrm{T}}\mathbf{J}))^{-1}\mathbf{J}^{\mathrm{T}}\epsilon. $$

J. Zhang, M. Kaess, and S. Singh, “Real-time depth enhanced monocular odometry,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sept. 2014.

Visual Odometry

$$(\boldsymbol{R}_1 - \bar{x}_i^k \boldsymbol{R}_3)\boldsymbol{X}_i^{k-1} + T_1 - \bar{x}_i^k T_3 = 0, \tag{3}$$

$$(\boldsymbol{R}_2 - \bar{y}_i^k \boldsymbol{R}_3)\boldsymbol{X}_i^{k-1} + T_2 - \bar{y}_i^k T_3 = 0, \tag{4}$$

$$[-\bar{y}_i^k T_3 + T_2,\ \bar{x}_i^k T_3 - T_1,\ -\bar{x}_i^k T_2 + \bar{y}_i^k T_1]\mathbf{R}\bar{\boldsymbol{X}}_i^{k-1} = 0. \tag{6}$$

$$\mathbf{R} = e^{\hat{\theta}} = \mathbf{I} + \frac{\hat{\theta}}{||\theta||}\sin||\theta|| + \frac{\hat{\theta}^2}{||\theta||^2}(1 - \cos||\theta||), \tag{7}$$

$$\boldsymbol{f}([\boldsymbol{T};\ \theta]) = \epsilon, \tag{8}$$

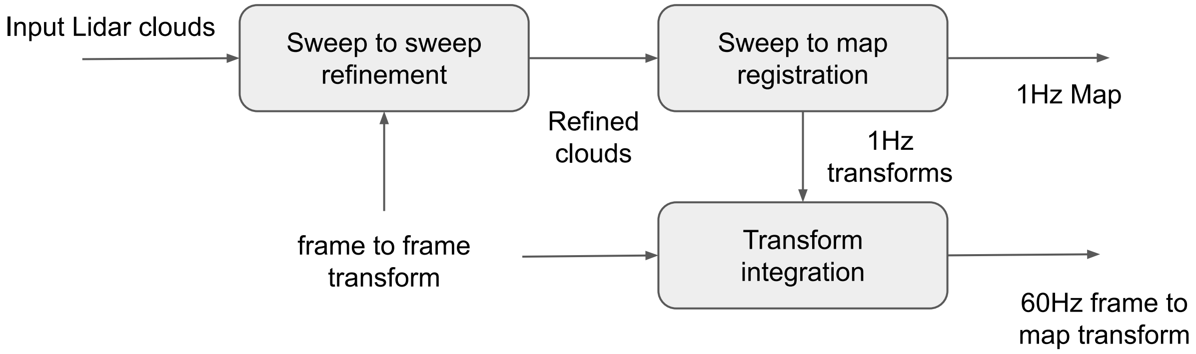
$$[\boldsymbol{T};\ \theta]^T \leftarrow [\boldsymbol{T};\ \theta]^T - (\mathbf{J}^T\mathbf{J} + \lambda \mathrm{diag}(\mathbf{J}^{\mathrm{T}}\mathbf{J}))^{-1}\mathbf{J}^{\mathrm{T}}\epsilon. \tag{9}$$

J. Zhang, M. Kaess, and S. Singh, “Real-time depth enhanced monocular odometry,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sept. 2014.

Algorithm 1: Frame to Frame Motion Estimation

input : $\bar{\boldsymbol{X}}_i^k$, $\boldsymbol{X}_i^{k-1}$ or $\bar{\boldsymbol{X}}_i^{k-1}$, $i \in \mathcal{I}$
output : θ, $\boldsymbol{T}$
begin
    $\theta, \boldsymbol{T} \leftarrow \mathbf{0}$;
    for a number of iterations **do**
        for each $i \in \mathcal{I}$ **do**
            if i is depth associated **then**
                Derive (3)-(4) using $\bar{\boldsymbol{X}}_i^k$ and $\boldsymbol{X}_i^{k-1}$, substitute (7) into (3)-(4) to obtain two equations, stack the equations into (8);
            end
            else
                Derive (6) using $\bar{\boldsymbol{X}}_i^k$ and $\bar{\boldsymbol{X}}_i^{k-1}$, substitute (7) into (6) to obtain one equation, stack the equation into (8);
            end
            Compute a bisquare weight for each feature based on the residuals in (3)-(4) or (6);
            Update θ, $\boldsymbol{T}$ for one iteration based on (9);
        end
        if the nonlinear optimization converges **then**
            Break;
        end
    end
    Return θ, $\boldsymbol{T}$;
end

Lidar Odometry



Lidar Odometry

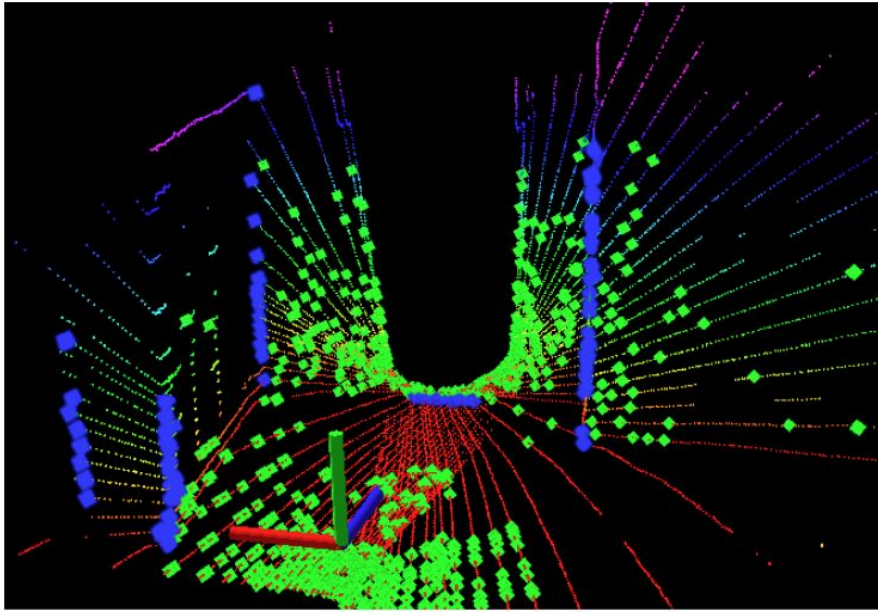
Smoothness term
(or curvature)

$$c = \frac{1}{|\mathcal{S}| \cdot ||\boldsymbol{X}^L_{(k,i)}||} \left\| \sum_{j \in \mathcal{S}, j \neq i} (\boldsymbol{X}^L_{(k,i)} - \boldsymbol{X}^L_{(k,j)}) \right\|.$$

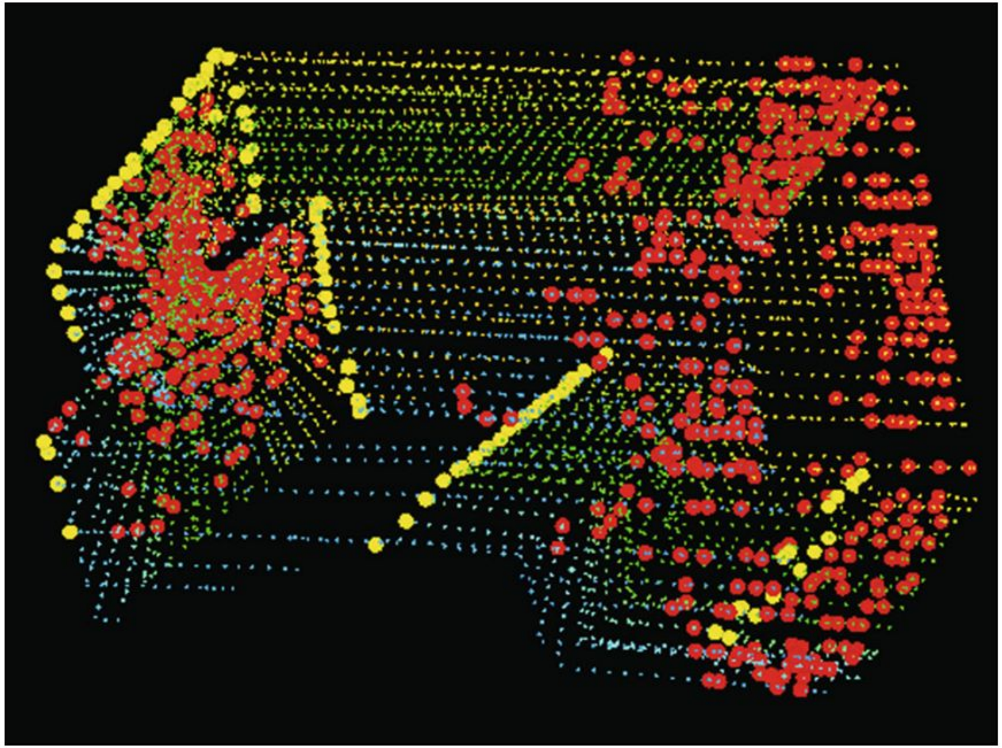
$\mathcal{S}$ contains half of its points on each side of i and 0.25 deg intervals between two points

J. Zhang and S. Singh, “LOAM: Lidar odometry and mapping in realtime,” in Robotics: Science and Systems Conference (RSS), July 2014.

Lidar Odometry



Edge (blue) and planar (green) points



Edge (yellow) and planar (red) points

Lidar Odometry

Smoothness term
(or curvature)

$$c = \frac{1}{|\mathcal{S}| \cdot ||\boldsymbol{X}^L_{(k,i)}||} \Big\| \sum_{j \in \mathcal{S}, j \neq i} (\boldsymbol{X}^L_{(k,i)} - \boldsymbol{X}^L_{(k,j)}) \Big\|.$$

$\mathcal{S}$ contains half of its points on each side of i and 0.25 deg intervals between two points

Edge points

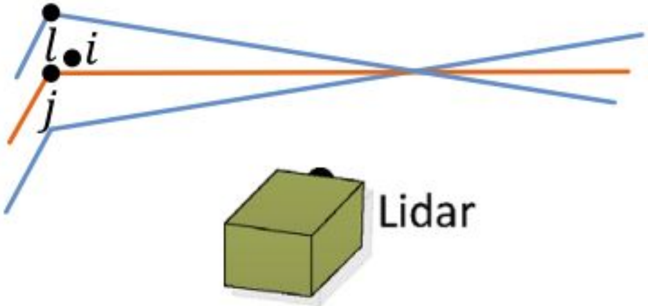
- Find two closest point in edge set
- Construct a line
- Find point to line distance

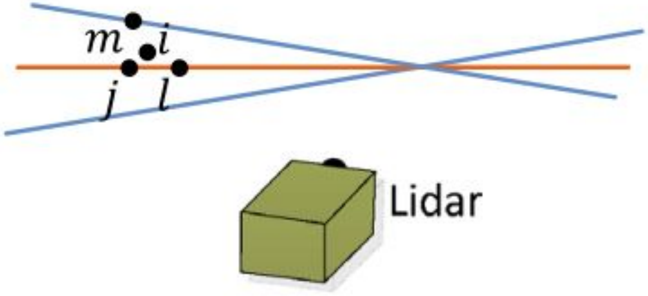
Planar points

- Find three closest point in planar set
- Construct a patch
- Find point to plane distance

J. Zhang and S. Singh, “LOAM: Lidar odometry and mapping in realtime,” in Robotics: Science and Systems Conference (RSS), July 2014.

Lidar Odometry





Edge points

- Find two closest point in edge set
- Construct a line
- Find point to line distance

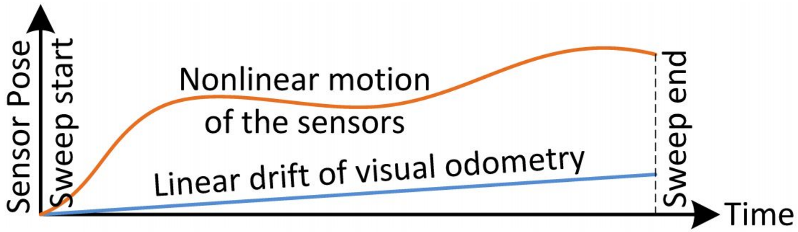
Planar points

- Find three closest point in planar set
- Construct a patch
- Find point to plane distance

J. Zhang and S. Singh, “LOAM: Lidar odometry and mapping in realtime,” in Robotics: Science and Systems Conference (RSS), July 2014.

Sweep to Sweep Refinement

- Removes point cloud distortion
- Models as linear drift



$$\underbrace{\boldsymbol{T}'_i = \boldsymbol{T}'}_{\text{6x1 vector describing the linear drift}} \underbrace{(t_i - t^m)}_{\text{Time between the start of sweep to the current point}} / \underbrace{(t^{m+1} - t^m)}_{\text{Total time between current and previous sweeps}}.$$

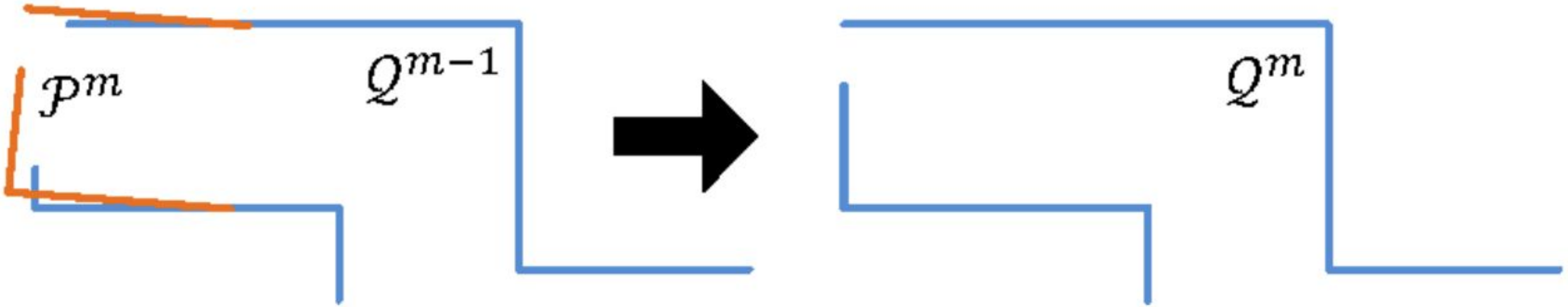
Sweep to Sweep Refinement

$$\boldsymbol{T}'_i = \boldsymbol{T}' \underbrace{(t_i - t^m)}_{\text{Time between the start of sweep to the current point}} / \underbrace{(t^{m+1} - t^m)}_{\text{Total time between current and previous sweeps}}.$$

6x1 vector describing the linear drift

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

Sweep to Map Registration



Transform Integration

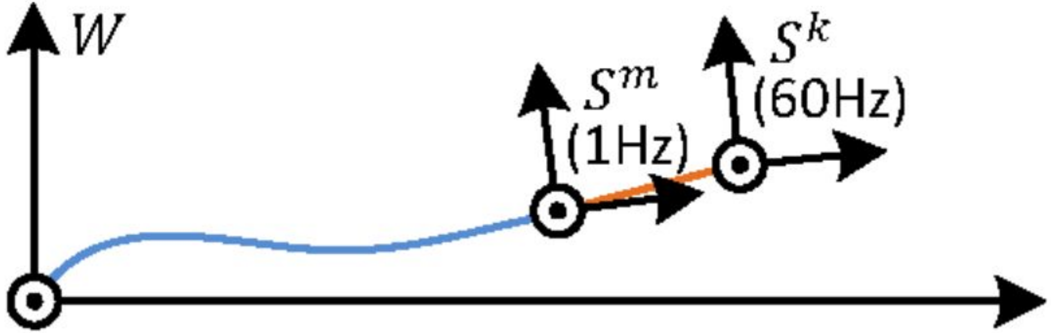


Fig. 7. Illustration of transform integration. The blue segment represents transforms published by the lidar odometry at a low frequency, regarding sensor poses in the world coordinate system $\{W\}$. The orange segment represents transforms published by the visual odometry at a high frequency containing frame to frame motion. The two transforms are integrated to generate high frequency sensor pose outputs at the image frame rate.

Experiments and Results

Datasets

- KITTI
- Custom

Tests

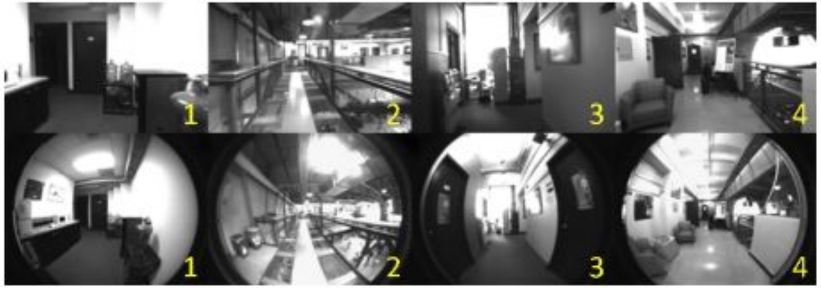
- Accuracy
- Robustness

Accuracy Tests

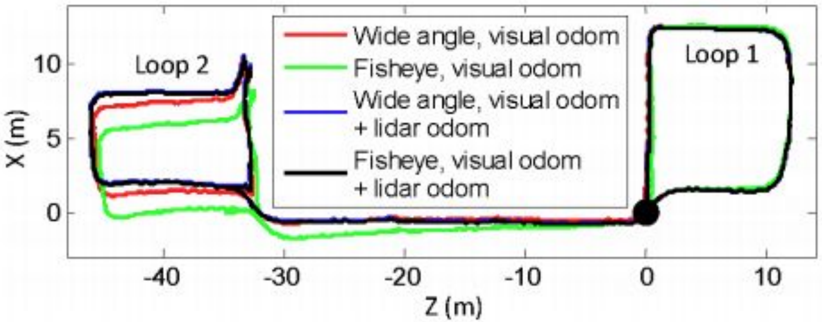
RELATIVE POSITION ERRORS IN ACCURACY TESTS

W: WIDE-ANGLE, F: FISHEYE, V: VISUAL ODOM (1ST SECTION IN FIG. 2), VL: VISUAL ODOM + LIDAR ODOM (BOTH SECTIONS IN FIG. 2).

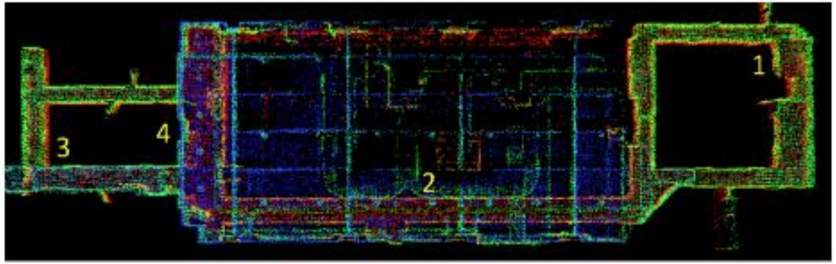
Test No.	Dist.	Relative Position Error			
		W-V	F-V	W-VL	F-VL
Test 1 (Loop 1)	49m	**1.1%**	1.8%	**0.31%**	**0.31%**
Test 1 (Loop 2)	47m	**1.0%**	2.1%	**0.37%**	**0.37%**
Test 2	186m	**1.3%**	2.7%	**0.63%**	0.64%
Test 3	538m	**1.4%**	3.1%	**0.71%**	0.73%


1
2
3
4
1
2
3
4

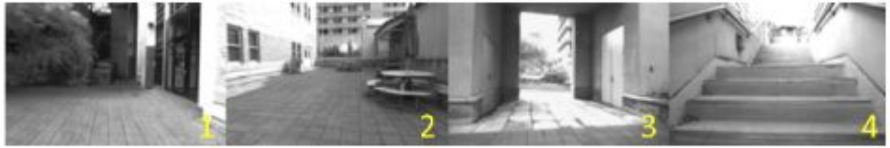
(a)


Loop 2
Loop 1
Wide angle, visual odom
Fisheye, visual odom
Wide angle, visual odom + lidar odom
Fisheye, visual odom + lidar odom
X (m)
Z (m)
10
5
0
-40
-30
-20
-10
0
10

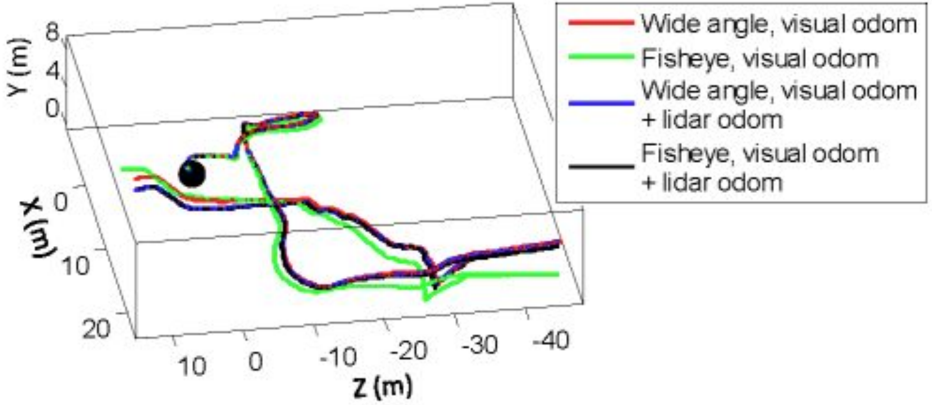
(b)


1
2
3
4

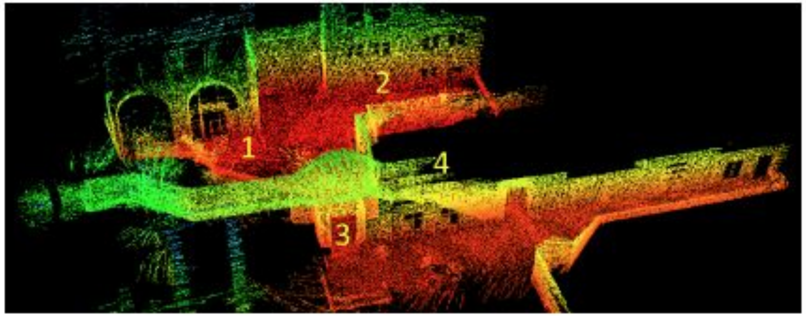
(c)


1
2
3
4

(a)


Wide angle, visual odom
Fisheye, visual odom
Wide angle, visual odom + lidar odom
Fisheye, visual odom + lidar odom
Y (m)
8
4
0
X (m)
0
10
20
Z (m)
10
0
-10
-20
-30
-40

(b)

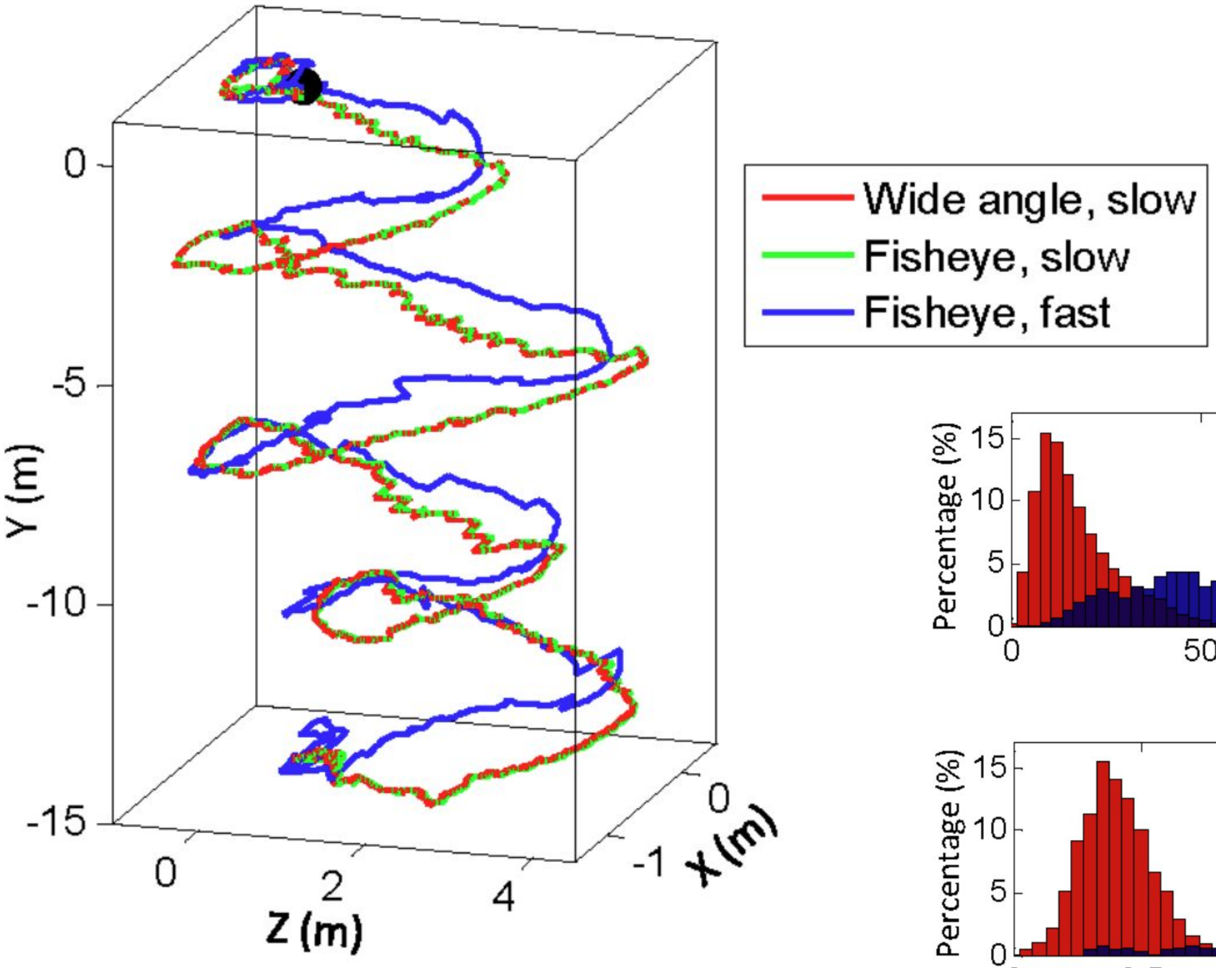

1
2
3
4

(c)

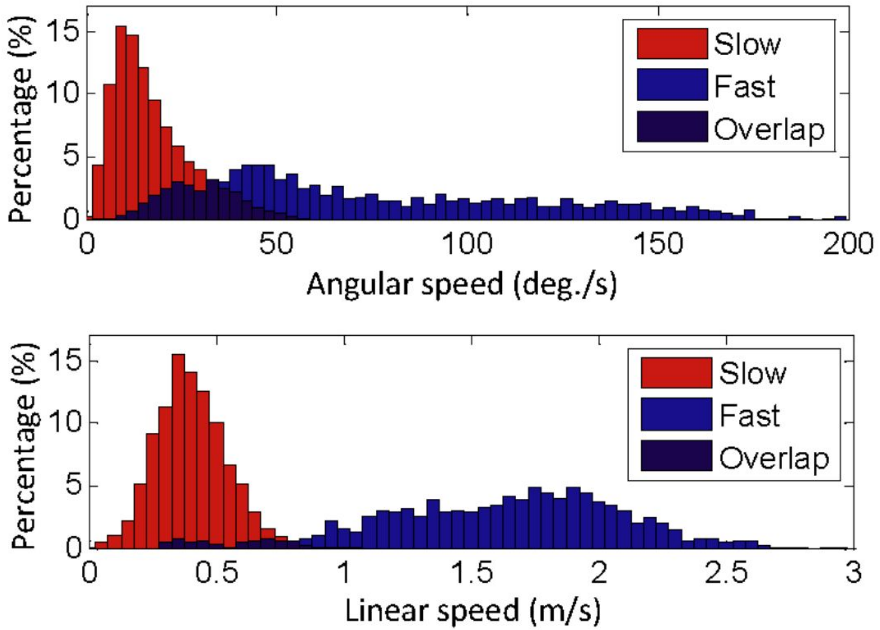
Robustness Tests

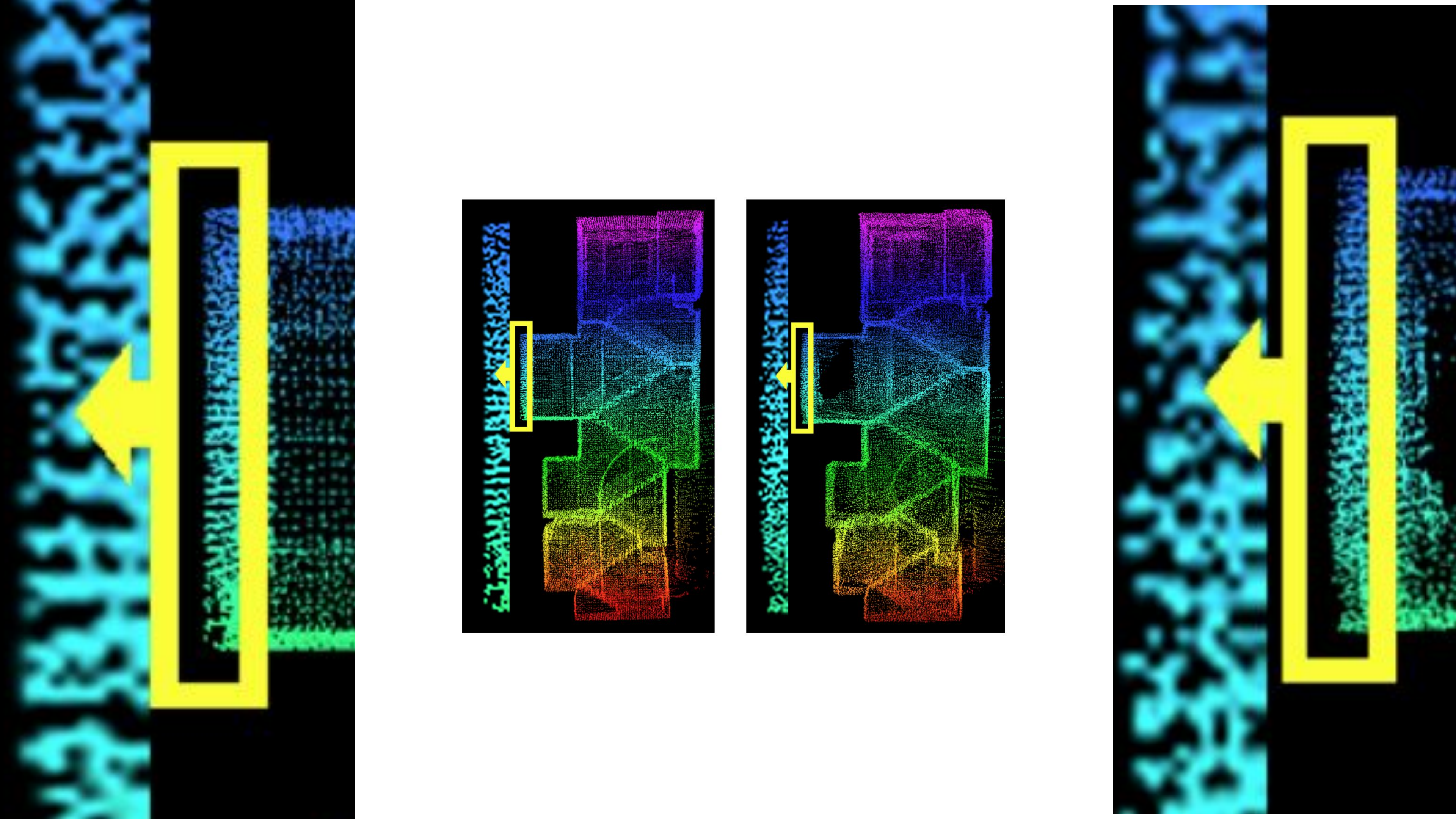
RELATIVE POSITION ERRORS IN FAST MOTION TESTS
W: WIDE-ANGLE, FI: FISHEYE, S: SLOW, FA: FAST.

Test No.	Dist.	Relative Position Error			
		W-S	Fi-S	W-Fa	Fi-Fa
Test 4	66m	**0.67%**	0.68%	Failed	**1.3%**
Test 5	54m	**0.27%**	0.28%	Failed	**0.39%**


Wide angle, slow
Fisheye, slow
Fisheye, fast
Y (m)
0
-5
-10
-15
Z (m)
0
2
4
X (m)
0
-1

(a)


Percentage (%)
Slow
Fast
Overlap
Angular speed (deg./s)
0
50
100
150
200
0
5
10
15
Percentage (%)
Slow
Fast
Overlap
Linear speed (m/s)
0
0.5
1
1.5
2
2.5
3



Questions?