Visual-Inertial Odometry with Point and Line Features

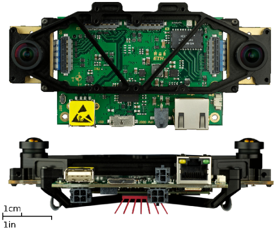
Yulin Yang, Patrick Geneva, Kevin Eckenhoff and Guoquan Huang

November 15, 2019

University of Delaware, Newark DE, USA

Introduction

- Visual-inertial navigation is widely used for 6DOF estimation.
- Rich amount of geometrical information available in man-made environment can be exploited.
- Points and lines are most commonly seen and can both be utilized for robust and accurate pose estimation.



Contribution

- A tightly-coupled monocular visual-inertial navigation system which leverages point&lines and is evaluated on real-world datasets.
- Discuss two 3D line feature triangulation algorithms.
- Degenerate motion identification for 3D line triangulation based on line segment measurements.
- Performance evaluation for 3 line representations in a visual SLAM scenario.

Related Work

- Kottas et al. [1] proposed to used lines in VIO with quaternion representation and performed line obs analysis.
- Guo et al. [2] utilized SLAM line (free&structural) within an visual-inertial SLAM system.
  - No degenerate motion analysis.
  - Only use only quaternion to represent line orientation.

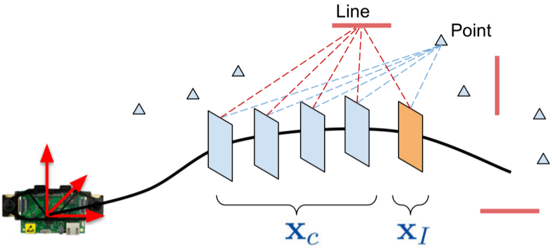
- Yu et al. [3] proposed two point inverse depth parameterization and applied line feature for rolling shutter.
  - No degenerate motion analysis.
  - No comparisons of line triangulation algorithms.

[1] Kottas and Roumeliotis, "Efficient and consistent vision-aided inertial navigation using line observations"

[2] Guo et al., "Large-scale cooperative 3d visual-inertial mapping in a manhattan world"

[3] Yu and Mourikis, "Vision-aided inertial navigation with line features and a rolling-shutter camera"

Problem Formulation



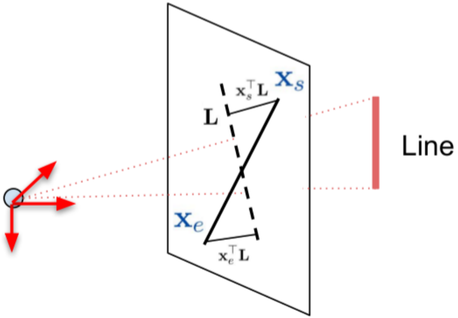
- State parameters: $\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^\top & \mathbf{x}_{calib}^\top & t_d & \mathbf{x}_c^\top \end{bmatrix}^\top$
- Perform online spatial and temporal calibration for both point&line features.
- Limit state vector size through MSCKF feature null-space operation.

Line Measurement

- Camera line measurement:

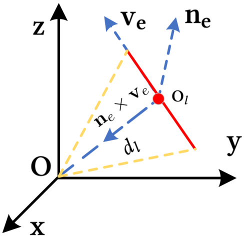
$$\mathbf{z}_l = \begin{bmatrix} \frac{\mathbf{x}_s^\top \mathbf{l}}{\sqrt{l_1^2+l_2^2}} & \frac{\mathbf{x}_e^\top \mathbf{l}}{\sqrt{l_1^2+l_2^2}} \end{bmatrix}^\top \quad (1)$$

- Start endpoint: $\mathbf{x}_s = [u_s, v_s, 1]^\top$
- End endpoint: $\mathbf{x}_e = [u_e, v_e, 1]^\top$
- Projected 2D image line: $\mathbf{L}$



Line Representation

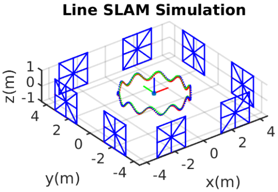
- The norm of plane formed by line with orig $\mathbf{O}$: $\mathbf{n}_e$.
- Line direction: $\mathbf{v}_e$
- Line distance to orig $\mathbf{O}$: d_l .
- Orientation form $\mathbf{R}(\bar{q}) = [\mathbf{n}_e, \mathbf{v}_e, \mathbf{n}_e \times \mathbf{v}_e]$.

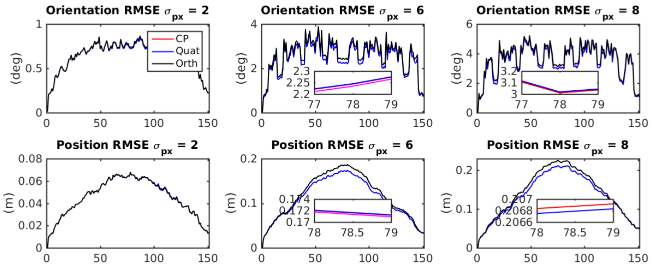


Model #	Line	Error states
Orthonormal	$\mathbf{n}_l$, $\mathbf{v}_l$	$\delta\boldsymbol{\theta}_l$, $\delta\phi_l$
Quaternion	d_l, $\bar{q}_l$ with $\mathbf{R}(\bar{q}_l) = [\mathbf{n}_e, \mathbf{v}_e, \mathbf{n}_e \times \mathbf{v}_e]$	$\delta\boldsymbol{\theta}_l$, $\tilde{d}_l$
Closest Point (CP)	$\mathbf{p}_l = d_l\bar{q}_l$	$\mathbf{p}_l = \hat{\mathbf{p}}_l + \tilde{\mathbf{p}}_l$

Simulation: Line Representation Comparisons

- Visual SLAM environment.
- Orthogonal, Quat and Closest Point (CP) line are compared.
- Finding: Quat and CP perform equally better than Ortho.

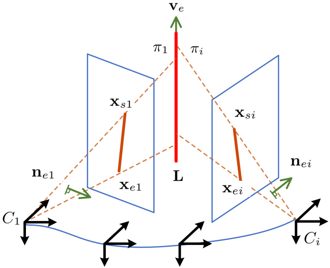




Line Triangulation - Overview

Given series of line segment observations from different clones
$\Rightarrow$ Estimate of geometric elements ($\mathbf{n}_e$, $\mathbf{v}_e$ and d_l) of the 3D line.

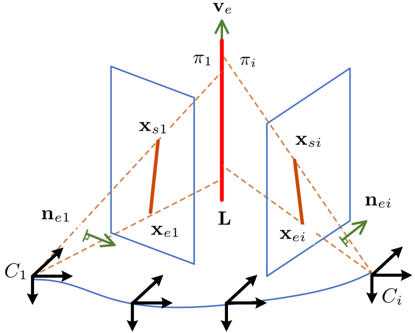
- Algorithm A: based on orthogonality to formulate linear system to recover $\mathbf{n}_e$, $\mathbf{v}_e$ and d_l respectively.
- Algorithm B: based on two intersecting planes to formulate Plücker matrix which contains 3D line parameters.



Algorithm A:

- Recover $\mathbf{n}_e$:
  $^{C_1}\mathbf{n}_e = \frac{\mathbf{x}_{s1} \times \mathbf{x}_{e1}}{\|\mathbf{x}_{s1} \times \mathbf{x}_{e1}\|}$
- Recover $\mathbf{v}_e$:

$$\begin{bmatrix} \vdots \\ {}^{C_i}\mathbf{n}_{ei}^{\top}{}^{C_1}_{C_i}\mathbf{R}^{\top} \\ \vdots \end{bmatrix} {}^{C_1}\mathbf{v}_{e1} = \mathbf{0}$$



- Recover the line distance d_l:

$$\begin{bmatrix} \vdots \\ \mathbf{b}_i^{\top}{}^{C_1}\mathbf{n}_{e1} \\ \vdots \end{bmatrix} {}^{C_1}d_l = \begin{bmatrix} \vdots \\ \mathbf{b}_i^{\top}\lfloor {}^{C_1}\mathbf{p}_{C_i} \rfloor {}^{C_1}_{C_i}\mathbf{R}\,{}^{C_i}\mathbf{v}_{ei} \\ \vdots \end{bmatrix}$$

where $\mathbf{b}_i = \lfloor {}^{C_1}\mathbf{v}_{e1} \rfloor {}^{C_1}_{C_i}\mathbf{R}\,{}^{C_i}\mathbf{n}_{ei}$.

Algorithm B:

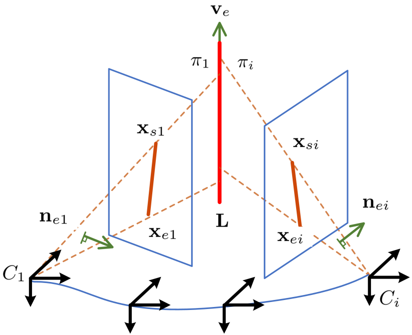
- The Plücker matrix:

$$\mathbf{L}^* = \pi_1 \pi_i^\top - \pi_i \pi_1^\top = \begin{bmatrix} \lfloor {}^{C_1}\mathbf{v}_{e1}^{(i)} \rfloor & {}^{C_1}d_l^{(i)} \, {}^{C_1}\mathbf{n}_{e1}^{(i)} \\ -{}^{C_1}d_l^{(i)} ({}^{C_1}\mathbf{n}_{e1}^{(i)})^\top & 0 \end{bmatrix}$$

- Recover geometry:

$${}^{C_1}\mathbf{n}_{e1} = \sum_{i=2}^{m} {}^{C_1}\mathbf{n}_{e1}^{(i)} \Big/ \left\| \sum_{i=2}^{m} {}^{C_1}\mathbf{n}_{e1}^{(i)} \right\|$$

$${}^{C_1}\mathbf{v}_{e1} = \sum_{i=2}^{m} {}^{C_1}\mathbf{v}_{e1}^{(i)} \Big/ \left\| \sum_{i=2}^{m} {}^{C_1}\mathbf{v}_{e1}^{(i)} \right\|$$

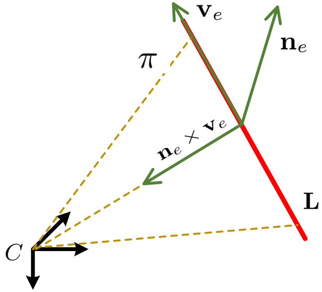


Degenerate Motion

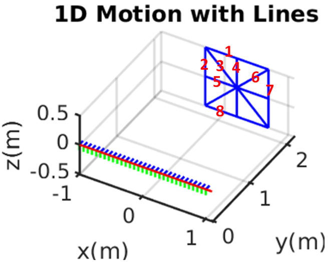
- Degenerate motions cause the line parameters to become unobservable, hence should be avoided.
- Any combined degenerate motions are also degenerate (i.e. all motions within the plane π will be degenerate).

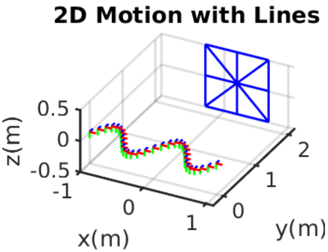
Table 1: Degenerate Motion Summary

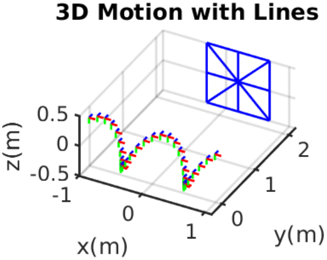
Motion	Solvable	Unsolvable
Along line direction: $\mathbf{v}_e$	$\mathbf{n}_e$	$\mathbf{v}_e$ and d
Toward line: $\mathbf{v}_e \times \mathbf{n}_e$	$\mathbf{n}_e$	$\mathbf{v}_e$ and d
Pure rotation	$\mathbf{n}_e$	$\mathbf{v}_e$ and d
Perpendicular to plane: $\mathbf{n}_e$	$\mathbf{n}_e$,$\mathbf{v}_e$ and d	-
Random motion	$\mathbf{n}_e$,$\mathbf{v}_e$ and d	-



- Simulate 3 motions with 8 lines.
- Algorithm A&B are implemented.
- Lines 1,5&8 are degenerate in 1D motion.

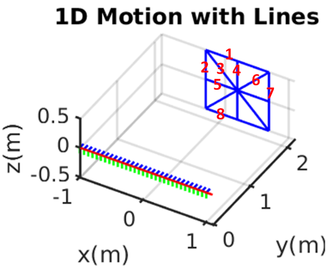


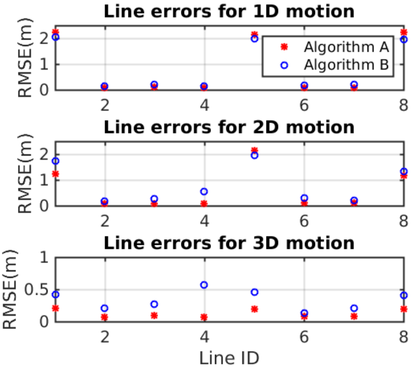




Simulation: Triangulation Algorithm and Degenerate Motion

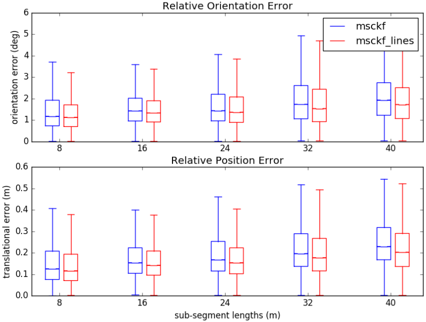
- Algorithm A, which takes advantage of the geometrical orthogonality, is better than B.
- Algorithm A w/ 3D motion performs best.

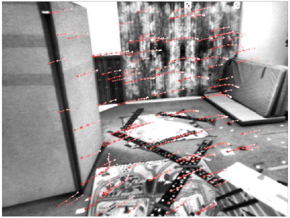


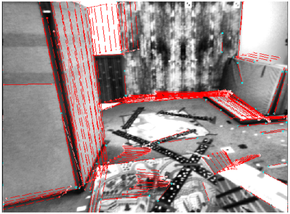


Real-World Experiments: EuRoc Datasets Results

- Adding lines improves the overall accuracy in most cases.
- Line quality is affected by environment structure, mono camera motion and visual tracking.

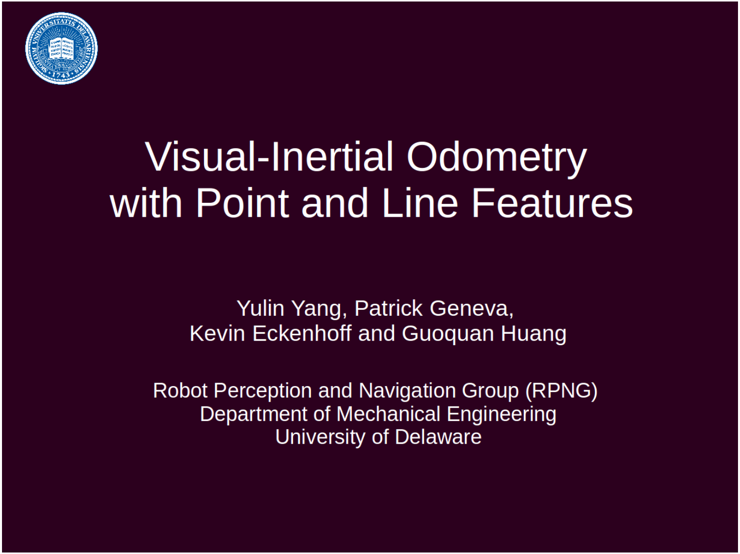






RPNG

System Demonstration



Visual Inertial Odometry with Point and Line Features

- A tightly-coupled mono-VIO with point&line features.
- Analysis different line triangulation algorithms and their degenerate motion.
- Compared line representations and showed Quat. and CP outperformed Ortho for larger pixel noises.
- Future work: Leverage SLAM line features (free&structural line), evaluate inverse depth line representations.

Y. Yang, P. Geneva, K. Eckenhoff, G. Huang
{yuyang,pgeneva, keck, ghuang}@udel.edu