Visual Navigation: From Sensor To Modeling II

AAE4203 – Guidance and Navigation

Dr Weisong Wen
Research Assistant Professor
Department of Aeronautical and Aviation Engineering
The Hong Kong Polytechnic University

Week 8, 9th Mar 2022

Outline

> Stereo Depth Estimation and Visual Odometry
  - Stereo based depth estimation (recover the depth)
  - Stereo visual odometry (visual positioning)

> Monocular Visual Odometry
  - Feature Detection and Matching (find same features from consecutive frames)
  - Epipolar Constraint (estimate relative motion-visual positioning)
  - Triangulation (estimate feature depth)

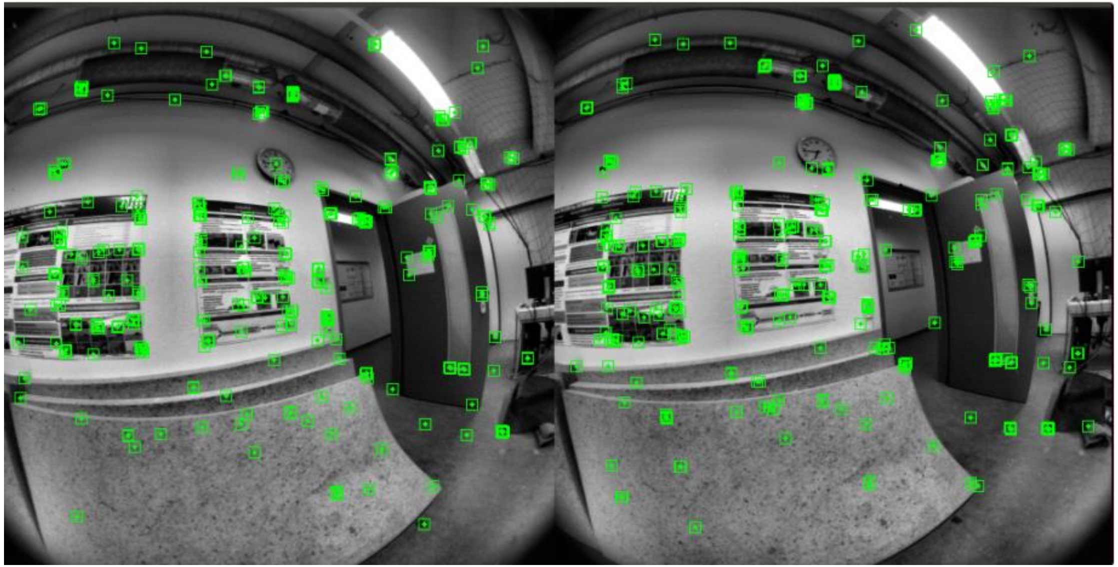
> Preview on Tutorial 2 (Visual Positioning)

> Corrections on the GNSS RTK Jacobian Matrix in Lecture Slide

Stereo Depth Estimation

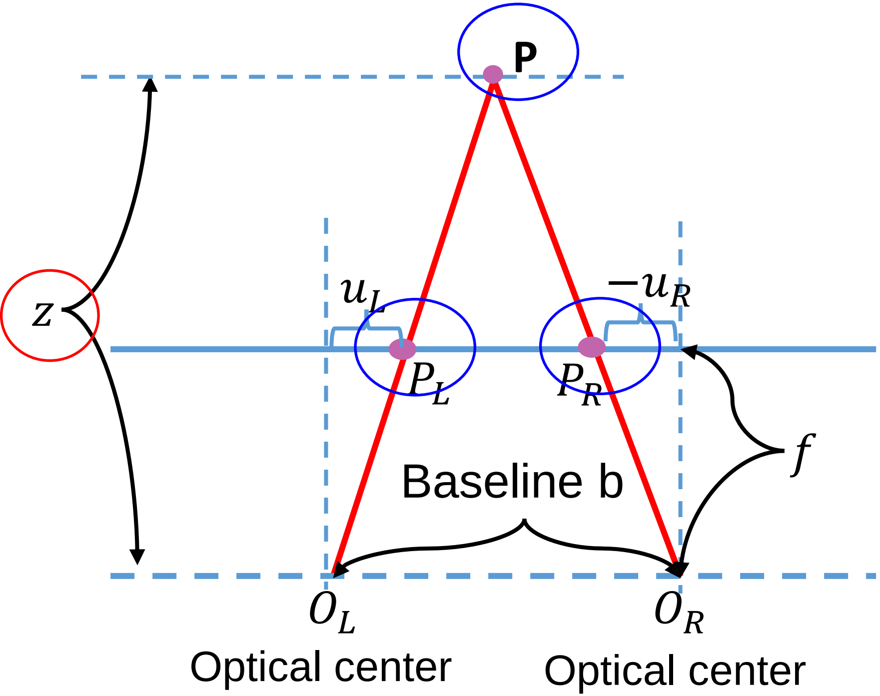
What is stereo camera?

View of stereo camera



Stereo Depth Estimation

What is the **Stereo Model** ?



z: the depth , distance (meter)

f: the focal length (meter)

b: the baseline (meter)

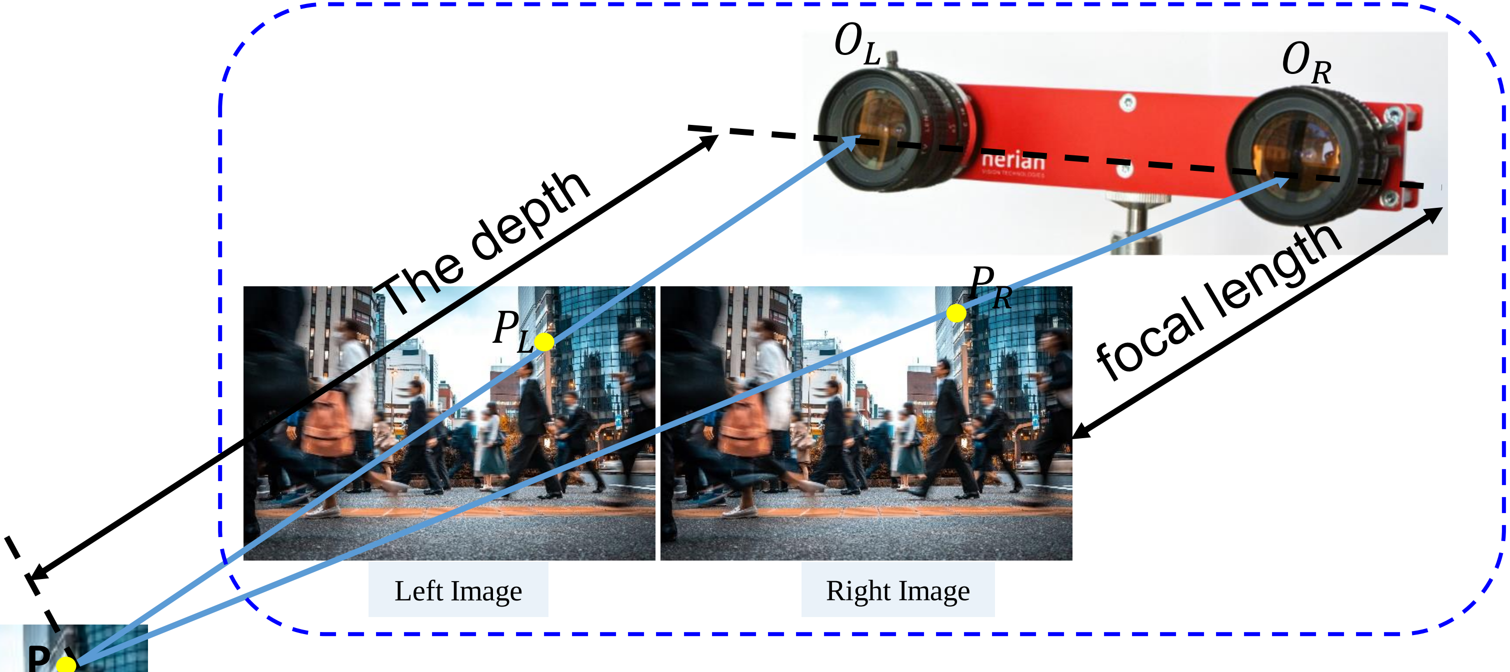
u_L, u_R : the coordinates of point P_L, P_R (pixel)
c: the pixel size of each pixel (m/pixel)

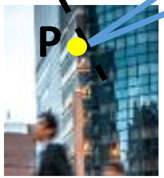
$$\Delta P P_L P_R \sim \Delta P O_L O_R$$

$$\frac{z-f}{z} = \frac{b-u_L+u_R}{b} \Rightarrow z = \frac{fb}{dc}, \quad d = u_L - u_R$$

Parallax in pixel position

Stereo Depth Estimation

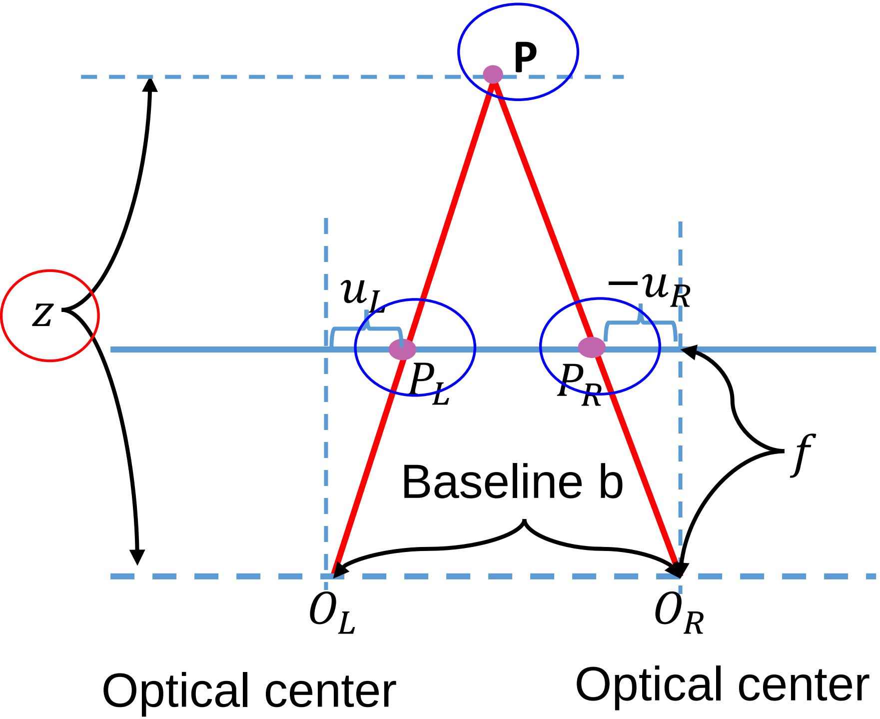




Based on previous model,
the depth can be obtained

Stereo Depth Estimation

What is the **maximum range of the depth estimation**?



$$z = \frac{fb}{d}$$

$$d = u_L - u_R$$

Parallax in pixel position

f: focal length (fixed value for given stereo camera)

$d = u_L - u_R$: maximum value is determined by size of the image

b: baseline we can tune to get maximum range!

Stereo Depth Estimation

Hong Kong UrbanNav: An Open-Sourced Multisensory Dataset for Benchmarking Urban Navigation Algorithm

Li-Ta Hsu*, Weisong Wen, Feng Huang, Guohao Zhang, Hoi-Fung Ng and Yihan Zhong

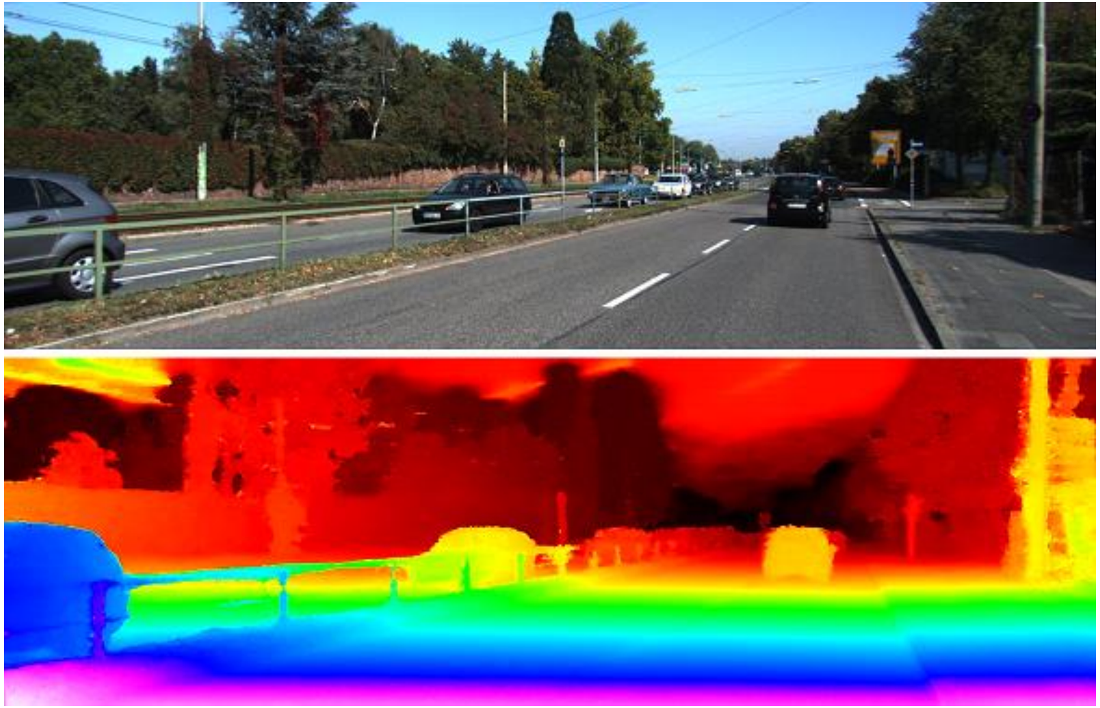
Abstract — **Urban canyon is typical in megacities like Hong Kong and Tokyo. Accurate positioning in urban canyons remains a challenging problem for applications with navigation requirements, such as navigation for pedestrians and unmanned autonomous systems. Specifically, the GNSS positioning can be significantly degraded in urban canyons due to the signal blockage by tall buildings. The visual positioning and light detection and ranging (LiDAR) based odometry is affected by numerous unexpected dynamic objects. Currently, the urban canyon dataset is not easily accessible for many researchers, resulting in the navigation research in the urban canyon is currently still a bottleneck for many innovative applications in urban areas. In addition, the sensors such as LiDAR and fiber optics gyroscope (FOG) are highly costed. To facilitate the research and development of robust, accurate, and precise positioning using multiple sensors in urban canyons, we built a multi-sensory dataset, UrbanNav, collected in diverse challenging urban scenarios in Hong Kong. The dataset provides full-suite sensor data, which includes global navigation satellite system (GNSS), inertial measurement unit (IMU), LiDAR, and cameras. Meanwhile, the ground truth of the positioning is provided using the GNSS real-time kinematic (RTK), and high accuracy integrated inertial system together with the post-processing via forward and backward smoothing. The dataset in its entirety can be found through the Github page https://github.com/IPNL-POLYU/UrbanNavDataset.**

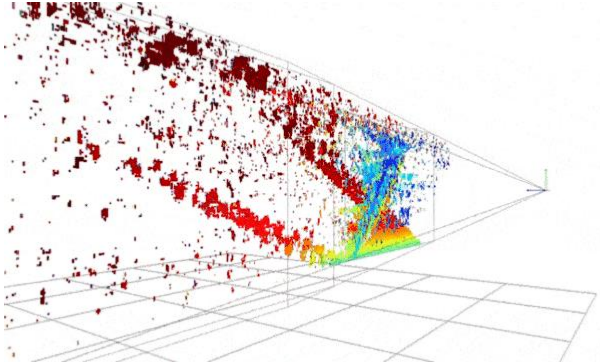
I. Introduction

phenomenon (Hsu, 2018). Numerous existing methods were investigated to mitigate the impacts of those multipath and NLOS receptions, such as 3D mapping aided (3DMA) GNSS (Ng, Zhang, Luo, & Hsu, 2021; Wang, Groves, & Ziebart, 2013, 2015), camera aided GNSS NLOS detection (X. Bai, Wen, & Hsu, 2020; Meguro, Murata, Takiguchi, Amano, & Hashizume, 2009; Wen, Bai, Kan, & Hsu, 2019), and 3D LiDAR aided GNSS NLOS (Wen, Zhang, & Hsu, 2018) or correction (X. Bai et al., 2020; Wen, Zhang, & Hsu, 2019). Unfortunately, the achieved GNSS positioning is still far from enough for fully autonomous systems with centimeter-level accuracy with stingy integrity requirements. How to effectively solve the GNSS positioning problem in urban canyons is still to be explored.

Unreliable odometry estimation in highly dynamic scenarios: Visual/inertial integrated system (VINS) (Qin, Li, & Shen, 2018) can provide low-cost and locally accurate odometry estimation in environments with sufficient features, using camera and IMU measurements. The VINS is characterized by such advantages in size, power assumption, weight, and availability. Many state-of-the-art VINS pipelines have been developed in the past several decades showing outperforming performance, such as the filtering-based methods including MSCKF (Mourikis & Roumeliotis, 2007), ROVIO (Bloesch, Omari, Hutter, & Siegwart, 2015), and Openvins (Geneva, Eckenhoff, Lee, Yang, & Huang, 2020).

Stereo Depth Estimation





What we get using stereo camera?

- Dense point clouds from the single frame images.
- 3D coordinates of the points.

Can we use the depth (dense point clouds) for positioning?

Farther depth with decreased accuracy in depth!

Properties of Several Matrices

Transformation Matrix:

$$\mathbf{T}_{\mathrm{A}}^{\mathrm{B}} = \begin{bmatrix} \mathbf{R}_{\mathrm{A}}^{\mathrm{B}} & \mathbf{t}_{\mathrm{A}}^{\mathrm{B}} \\ 0 & 1 \end{bmatrix}$$

Transform a point $\boldsymbol{p}_1^{\mathrm{A}}$ from coordinate A to coordinate B:

$$\boldsymbol{p}_2^{\mathrm{B}} = \mathbf{R}_{\mathrm{A}}^{\mathrm{B}} \boldsymbol{p}_1^{\mathrm{A}} + \mathbf{t}_{\mathrm{A}}^{\mathrm{B}}$$

Key features of rotation matrix $\mathbf{R}_{\mathrm{A}}^{\mathrm{B}}$ (orthogonal matrix):

$$\mathbf{R}_{\mathrm{A}}^{\mathrm{B}\,\mathrm{T}} = \mathbf{R}_{\mathrm{A}}^{\mathrm{B}\,-1}$$

$$\mathbf{R}_{\mathrm{A}}^{\mathrm{B}\,\mathrm{T}} \mathbf{R}_{\mathrm{A}}^{\mathrm{B}} = \mathbf{I}$$

$\mathbf{x}^{\wedge}$: **Skew-symmetric** Matrix,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\mathbf{x}^{\wedge} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

SVD of $\mathbf{W}$ which is a $m \times n$ matrix:

$$\mathbf{W} = \mathbf{U\Sigma V^T}$$

$\mathbf{U}$: orthogonal matrix ($m \times m$ matrix)

$\mathbf{V}$: orthogonal matrix ($n \times n$ matrix)

$\mathbf{\Sigma}$: orthogonal matrix ($m \times n$ matrix)

SVD: Singular Value Decomposition, tr: Trace

Iterative Closest Point (ICP) Modeling

i, Index of matched point clouds (shortest distance) between two frames

Initial point clouds (n points):
$\mathbf{p}_i = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots\}$

Current point clouds:
$\mathbf{p}_i{}' = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots\}$

Error for points pair ($\mathbf{p}_i$, $\mathbf{p}_i{}'$):
$e_i = \mathbf{p}_i - (\mathbf{R}\mathbf{p}_i{}' + \mathbf{t})$

Mean of point sets:
$\bar{\mathbf{p}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{p}_i)$, $\bar{\mathbf{p}}' = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{p}_i{}')$

Difference of the point *i* to mean:
$\mathbf{q}_i = \mathbf{p}_i - \bar{\mathbf{p}}$, $\mathbf{q}_i{}' = \mathbf{p}_i{}' - \bar{\mathbf{p}}'$

Error Function:

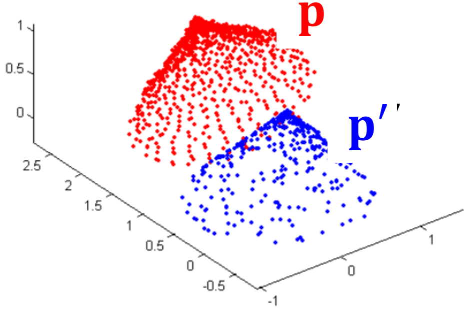
$$\frac{1}{2}\sum_{i=1}^{n}\|\mathbf{p}_i - (\mathbf{R}\mathbf{p}_i{}' + \mathbf{t})\|^2 = \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{p}_i - \mathbf{R}\mathbf{p}_i{}' - \mathbf{t} - \bar{\mathbf{p}} + \mathbf{R}\bar{\mathbf{p}}' + \bar{\mathbf{p}} - \mathbf{R}\bar{\mathbf{p}}'\|^2$$

$$= \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{p}_i - \bar{\mathbf{p}} - \mathbf{R}(\mathbf{p}_i{}' - \bar{\mathbf{p}}')\|^2 + \|\bar{\mathbf{p}} - \mathbf{R}\mathbf{p}_i{}' - \mathbf{t})\|^2$$

$$= \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{q}_i - \mathbf{R}(\mathbf{q}_i{}')\|^2 + \|\bar{\mathbf{p}} - \mathbf{R}\mathbf{p}_i{}' - \mathbf{t})\|^2$$

Related to **R** only based on independent points

Related to **R** and **t** based on mean of points set



Solve $\hat{\mathbf{R}}, \hat{\mathbf{t}}$ via two steps:
Step 1: Solve rotation using Singular Value Decomposition (SVD)

- $\hat{\mathbf{R}} = \min\limits_{\mathbf{R}} \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{q}_i - \mathbf{R}\mathbf{q}_i{}'\|^2$

Step 2: Solve translation based on rotation

- $\hat{\mathbf{t}} = \mathbf{p} - \mathbf{R}\mathbf{p}'$

Steps 1&2 performs iteratively until:

- Change of $\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$ are small enough, or
- Change of the loss is small enough, or
- Number of iterations is large enough.

Iterative Closest Point (ICP) Optimization Using SVD*

Revised Optimization Function:

$$\min_{\mathbf{R},\mathbf{t}} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{p}_i - ((\mathbf{R}\mathbf{p}_i' + \mathbf{t}))\|^2$$

$$= \min_{\mathbf{R},\mathbf{t}} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{p}_i - \mathbf{p} - \mathbf{R}(\mathbf{p}_i' - \mathbf{p}')\|^2 + \|\mathbf{p} - \mathbf{R}\mathbf{p}_i' - \mathbf{t})\|^2$$

Solve $\mathbf{R}^*, \mathbf{t}^*$ via two steps:

Step 1: Solve rotation using Singular Value Decomposition (SVD)

- $\mathbf{R}^* = \min_{\mathbf{R}} \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{q}_i - \mathbf{R}\mathbf{q}_i'\|^2$

Step 2: Solve translation based on rotation

- $\mathbf{t}^* = \mathbf{p} - \mathbf{R}\mathbf{p}'$

SVD of $\mathbf{W}$ which is a 3×3 matrix: $\mathbf{W} = \sum_{i=1}^{n} \mathbf{q}_i'\mathbf{q}_i^T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\mathrm{T}$

Step 1: Solve rotation using Singular Value Decomposition (SVD)

$$\frac{1}{2}\sum_{i=1}^{n}\|\mathbf{q}_i - \mathbf{R}\mathbf{q}_i'\|^2 = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{q}_i^T\mathbf{q}_i + \mathbf{q}_i'^T\mathbf{R}^T\mathbf{R}\mathbf{q}_i' - 2\mathbf{q}_i^T\mathbf{R}\mathbf{q}_i')$$

$$= -\frac{1}{2}\sum_{i=1}^{n}\mathbf{q}_i^T\mathbf{R}\mathbf{q}_i' = \frac{1}{2}\sum_{i=1}^{n} -\mathrm{tr}(\mathbf{R}\mathbf{q}_i'\mathbf{q}_i^T)$$

$$= -\mathrm{tr}(\mathbf{R}\sum_{i=1}^{n}\mathbf{q}_i'\mathbf{q}_i^T)$$

$$= -\mathrm{tr}(\mathbf{R}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\mathrm{T}) = -\mathrm{tr}(\boldsymbol{\Sigma}\mathbf{V}^\mathrm{T}\mathbf{R}\mathbf{U})$$

$\mathbf{V}^\mathrm{T}\mathbf{R}\mathbf{U}$ is an orthogonal matrix and $\mathbf{R}^*$ is obtained when $\mathbf{V}^\mathrm{T}\mathbf{R}\mathbf{U}$ is an identity matrix!

$$\mathbf{R}^* = \mathbf{V}\mathbf{U}^\mathrm{T}$$

$\mathbf{V}^\mathrm{T}, \mathbf{R}, \mathbf{U}$ are all orthogonal matrix

Step 2: Solve translation based on rotation

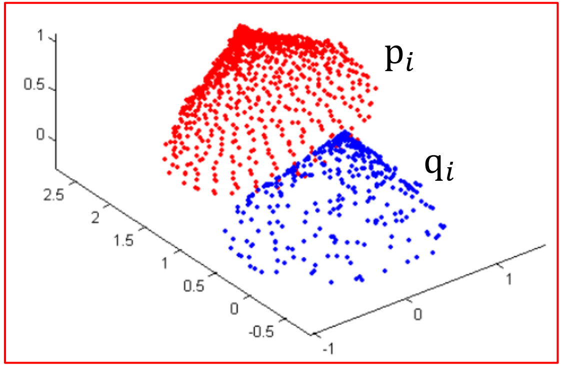
- $\mathbf{t}^* = \mathbf{p} - \mathbf{U}\mathbf{V}^\mathrm{T}\mathbf{p}'$

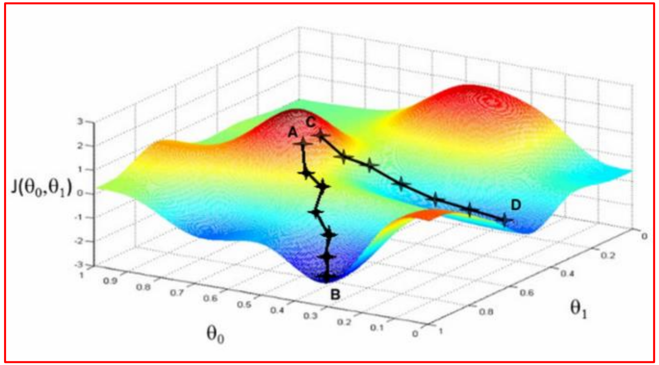
Be noted that the Step 1 and Step 2 performs iteratively until:

- The change of $\mathbf{R}^*$ and $\mathbf{t}^*$ are small enough, or
- The change of the loss is small enough, or
- The number of iterations is large enough, or

SVD: Singular Value Decomposition, tr: Trace

Iterative Closest Point (ICP) Optimization Using SVD*





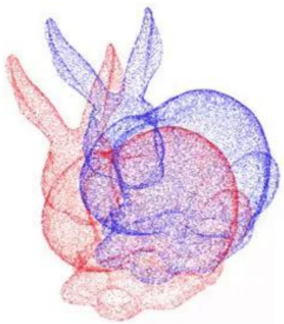
Reference point clouds:

$p_i = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots\}$

Target point clouds:

$q_i = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots\}$

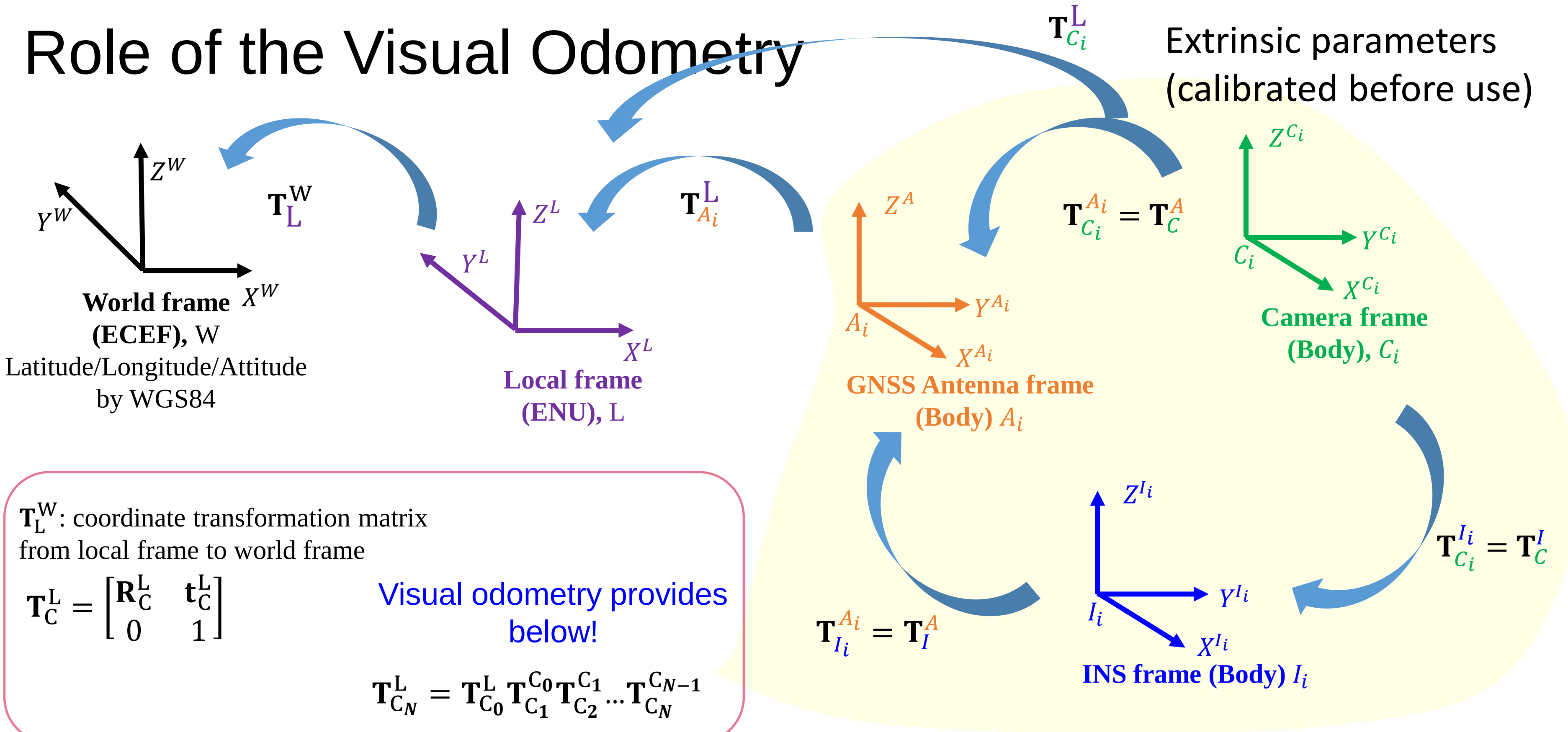
Iteration 0



$$\hat{R}, \hat{\mathbf{t}} = \arg\min_{R,\mathbf{t}} \sum_{i=1}^{N} \|(R\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i\|^2$$

Find the best R,T to map the red point clouds back to the blue point clouds

Role of the Visual Odometry



$\mathbf{T}_{\mathrm{L}}^{\mathrm{W}}$: coordinate transformation matrix from local frame to world frame

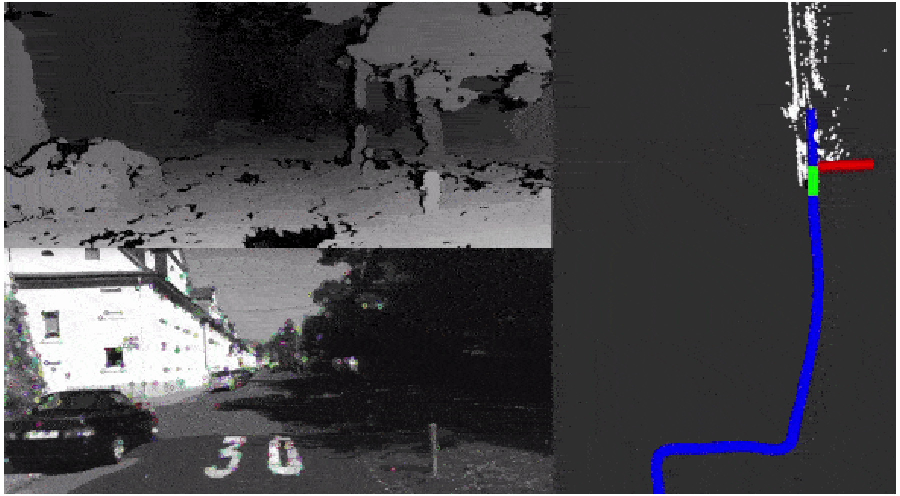
$$\mathbf{T}_{\mathrm{C}}^{\mathrm{L}} = \begin{bmatrix} \mathbf{R}_{\mathrm{C}}^{\mathrm{L}} & \mathbf{t}_{\mathrm{C}}^{\mathrm{L}} \\ 0 & 1 \end{bmatrix}$$

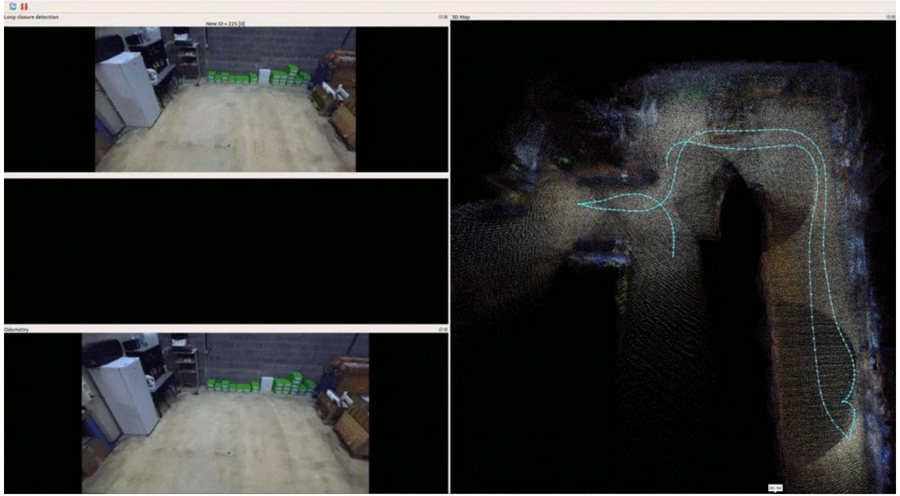
Visual odometry provides below!

$$\mathbf{T}_{\mathrm{C}_N}^{\mathrm{L}} = \mathbf{T}_{\mathrm{C}_0}^{\mathrm{L}} \mathbf{T}_{\mathrm{C}_1}^{\mathrm{C}_0} \mathbf{T}_{\mathrm{C}_2}^{\mathrm{C}_1} \ldots \mathbf{T}_{\mathrm{C}_N}^{\mathrm{C}_{N-1}}$$

Paul Groves (2013), "Principles of GNSS, inertial, and multi-sensor integrated navigation systems" *Artech House* (2nd edition)

Stereo Visual Odometry with Depth Information



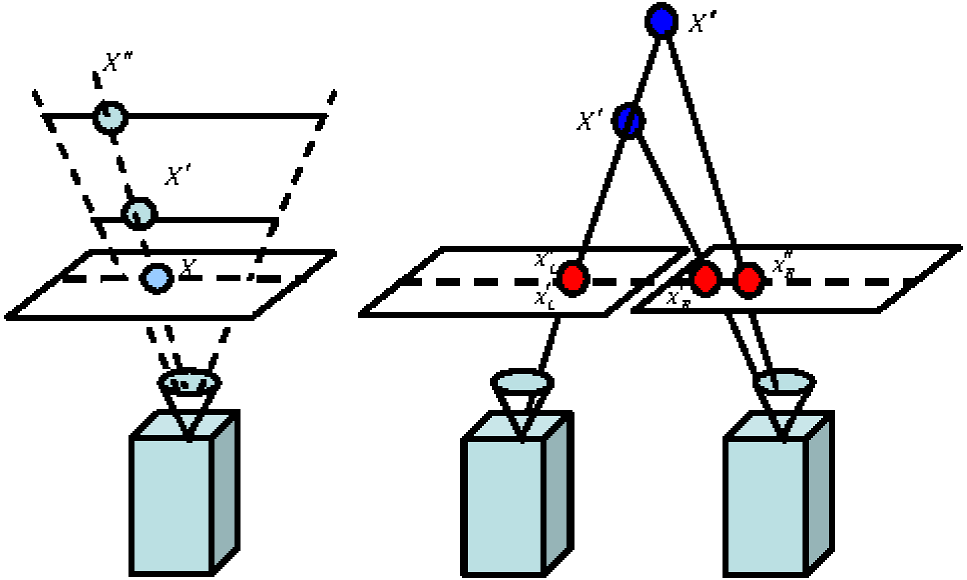


The key drawbacks of stereo visual odometry

- Limited by the range of the depth
- Sensitive to the illumination conditions

Visual Odometry (VO)

> What can we do if we only have a monocular camera?



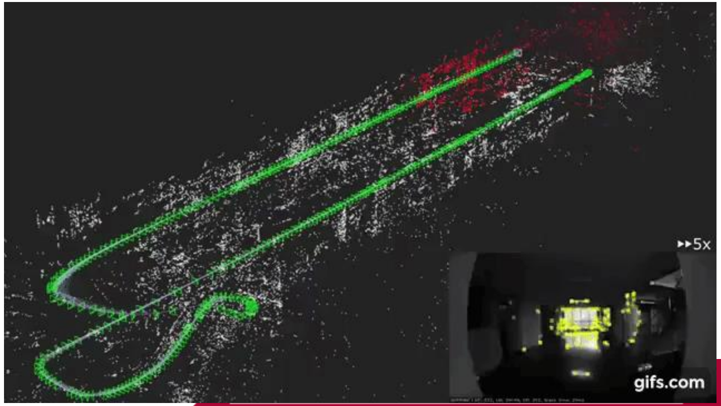
The pipeline of visual odometry with a camera

Find the representative features in an image!

Formulate the difference between stereo and monocular visual positioning!

Camera → Image → Feature Detection → Feature Tracking → Feature Association → Pose Estimation

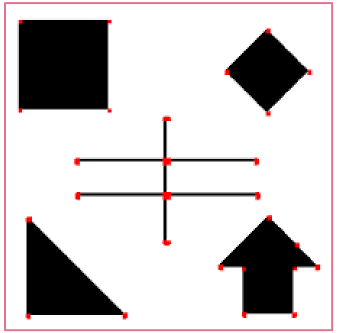
Find the same features in consecutive image!



Feature Detection Using Shi-Tomasi Corner

> What's a good feature? Distinctive! Line is not!

> Consider the image window centered at [*u*, *v*] to produce the grayscale change $E(u,v)$



$$E_p(u_p, v_p) = \sum_{(w_u,w_v)} \left[I(w_u + u_p, w_v + v_p) - I(u_p, v_p)\right]^2$$

Linearize by 1st order Tayler series expansion to use the tools in linear algebra

$$I(w_p + u_p, w_v + v_k) \approx I(u_p, v_p) + uI_{w_u} + vI_{w_v}$$

$I_{w_u} = \frac{\partial I(w_u,w_v)}{\partial w_u}$ $I_{w_v} = \frac{\partial I(w_u,w_v)}{\partial w_v}$

We want corners not lines.

$min(\lambda_1, \lambda_2) > \lambda_{min}$

$$E_p(u_p, v_p) \approx \sum_{(w_u,w_v)} \begin{bmatrix} u_p & v_p \end{bmatrix} \begin{bmatrix} I_{w_u}^2 & I_{w_v}I_{w_v} \\ I_{w_v}I_{w_v} & I_{w_v}^2 \end{bmatrix} \begin{bmatrix} u_p \\ v_p \end{bmatrix}$$

SVD decomposition

$$\mathbf{f} = \mathrm{find}\left(min(\lambda_{p,1}, \lambda_{p,2}) > \lambda_{min}\right)$$
$$p \in \text{pixels in the images}$$

$$\sum_{(w_u,w_v)} \begin{bmatrix} I_{w_u}^2 & I_{w_v}I_{w_v} \\ I_{w_v}I_{w_v} & I_{w_v}^2 \end{bmatrix} = \mathbf{U}^{-1} \begin{bmatrix} \lambda_{p,1} & 0 \\ 0 & \lambda_{p,2} \end{bmatrix} \mathbf{U}$$

Shi, Jianbo. "Good features to track." *1994 Proceedings of IEEE conference on computer vision and pattern recognition.*

Feature Matching via Optical Flow

B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in IJCAI'81, 1981, pp. 674-679

> The **displacement of the feature** is caused by camera movement.

> Three assumptions made:

1. Constant brightness
2. Feature did not move in actual world
3. Spatial consistency

$$I(u,v,t) = I(u+du, v+dv, t+dt)$$

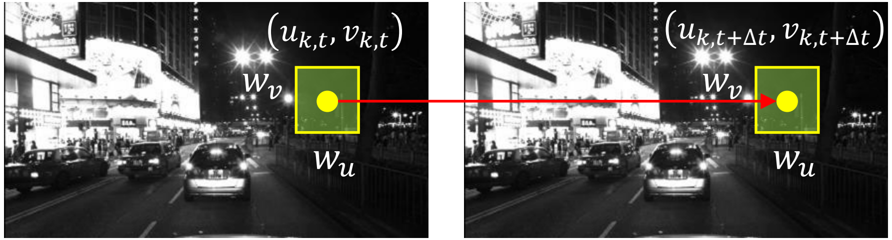
$$I(u+du, v+dv, t+dt) \approx I(u,v,t) + \frac{\partial I}{\partial u}du + \frac{\partial I}{\partial v}dv + \frac{\partial I}{\partial t}dt$$

$$\frac{\partial I}{\partial u}du + \frac{\partial I}{\partial v}dv + \frac{\partial I}{\partial t}dt = 0$$

$$\downarrow$$

$$\frac{\partial I}{\partial u}\frac{u}{dt} + \frac{\partial I}{\partial v}\frac{v}{dt} = -\frac{\partial I}{\partial t}$$

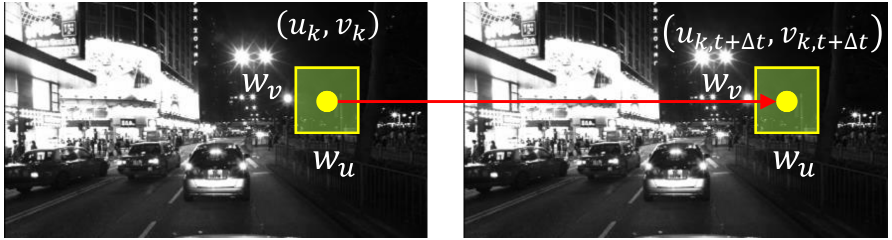
$$I_u \quad u_t \quad I_v \quad v_t \quad I_t$$



$k \in \mathbf{f}$, feature detected

$w_u * w_v$, the neighboring area of feature point

Feature Matching via Optical Flow



Check whether the feature loss, $\varepsilon_k < \text{thres}$

$$\varepsilon_k(\Delta u, \Delta v) = \sum_{x \in w_u} \sum_{y \in w_v} \left(\mathrm{I}_t(x, y) - \mathrm{I}_{t+\Delta t}(x + \Delta u, y + \Delta v)\right)^2$$

$k \in \mathbf{f}$, feature detected

$$\frac{\partial I}{\partial u}\frac{u}{dt} + \frac{\partial I}{\partial v}\frac{v}{dt} = -\frac{\partial I}{\partial t}$$

$$\begin{bmatrix} I_u & I_v \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} -I_t \end{bmatrix}$$

$$\begin{bmatrix} I_{x,1} & I_{y,1} \\ I_{x,2} & I_{y,2} \\ \vdots & \vdots \\ I_{x,(w_u \times w_v)} & I_{y,(w_u \times w_v)} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = -\begin{bmatrix} I_{t,1} \\ I_{t,2} \\ \vdots \\ I_{t,(w_u \times w_v)} \end{bmatrix}$$

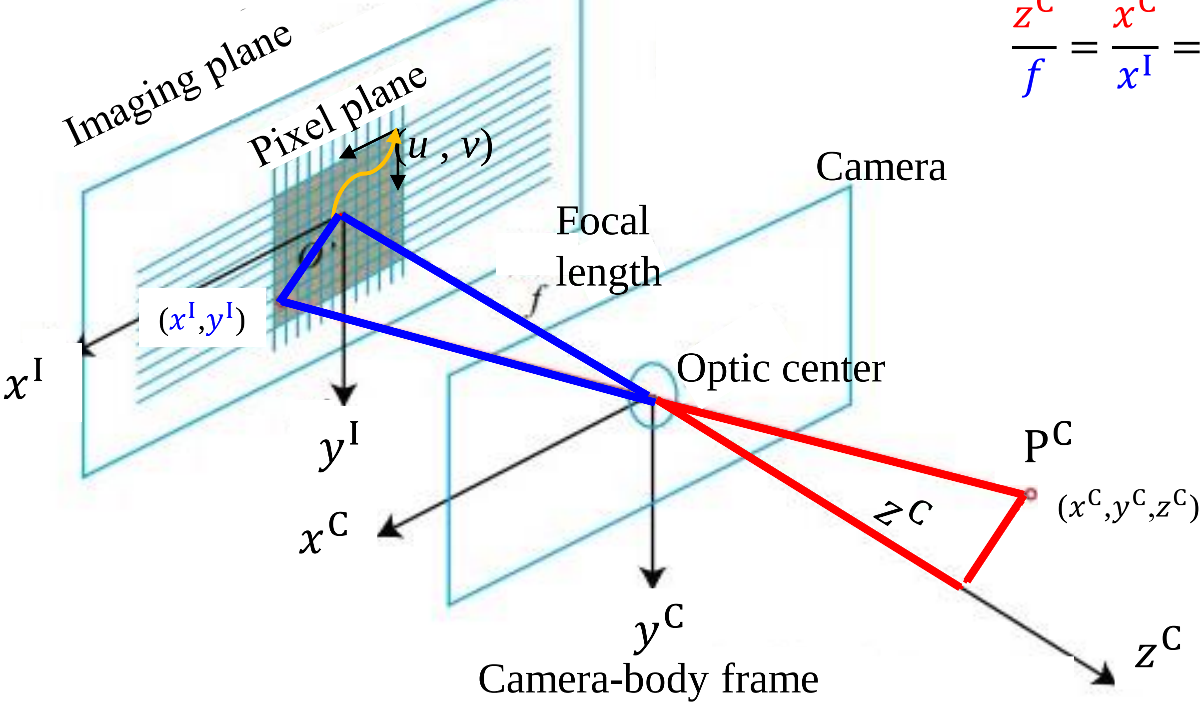
$$\mathbf{x} = \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix}$$

$$\mathbf{H}_k \mathbf{x}_k = -\mathbf{z}_k \qquad \hat{\mathbf{x}}_k = \left(\mathbf{H}_k^{\mathrm{T}} \mathbf{H}_k\right)^{-1} \mathbf{H}_k^{\mathrm{T}} (-\mathbf{z}_k)$$

The goal is to find $(u_{k,t+\Delta t}, v_{k,t+\Delta t})$, where (u_k, v_k) and $(u_{k,t+\Delta t}, v_{k,t+\Delta t})$ are similar, $I(u_{k,t+\Delta t}, v_{k,t+\Delta t}) = I((u_k, v_k) + (\Delta u, \Delta v))$

Model of the camera

Pinhole Model



Camera frame

$$\frac{z^C}{f} = \frac{x^C}{x^I} = \frac{y^C}{y^I}$$

Imaging plane

$$\begin{cases} x^I = f\dfrac{x^C}{z^C} \\ y^I = f\dfrac{y^C}{z^C} \end{cases}$$

Scale and translation

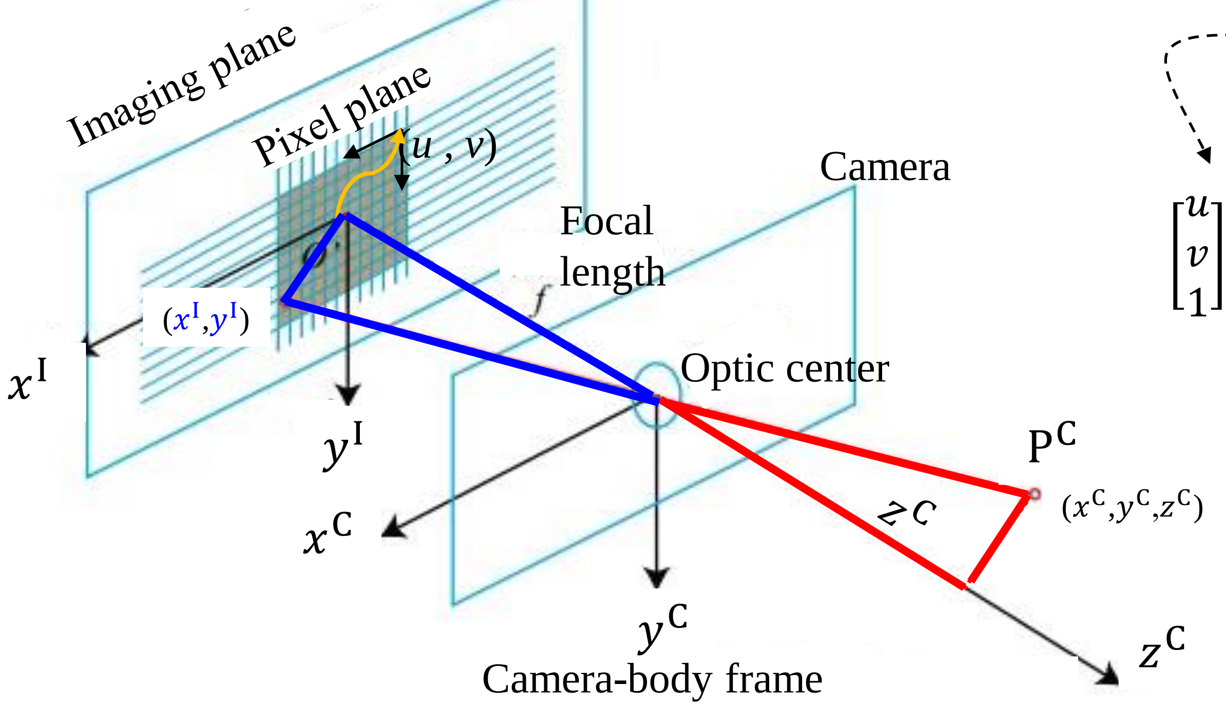
(f_u, f_v) $(\Delta u, \Delta v)$

Pixel plane

$$\begin{cases} u = f_u \dfrac{x^C}{z^C} + \Delta u \\ v = f_v \dfrac{x^C}{z^C} + \Delta v \end{cases}$$

Model of the camera

Pinhole Model



Pixel plane

Scale and translation

$(\mathrm{f}_u, \mathrm{f}_v)$ $(\Delta \mathrm{u}, \Delta \mathrm{v})$

$$\begin{cases} u = \mathrm{f}_u \dfrac{x^{\mathrm{C}}}{z^{\mathrm{C}}} + \Delta \mathrm{u} \\ v = \mathrm{f}_v \dfrac{x^{\mathrm{C}}}{z^{\mathrm{C}}} + \Delta \mathrm{v} \end{cases}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z^{\mathrm{C}}} \begin{bmatrix} \mathrm{f}_u & 0 & \Delta \mathrm{u} \\ 0 & \mathrm{f}_v & \Delta \mathrm{v} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^{\mathrm{C}} \\ y^{\mathrm{C}} \\ z^{\mathrm{C}} \end{bmatrix}$$

$$z^{\mathrm{C}} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{p}^{\mathrm{C}}$$

K:Camera Intrinsic Matrix

s: Depth between Camera & Feature

$$s\mathbf{p}^{\mathrm{I}} = \mathbf{K}\mathbf{p}^{\mathrm{C}}$$

Rotation and Position Representation

Given

- The position of a particle **a** in the body-fixed coordinate as (a_x^B, a_y^B, a_z^B)
- The transformation between between $\mathbf{C}_B$ and $\mathbf{C}_G$ as rotation matrix $\mathbf{R}_B^G$ and translation vector $\mathbf{t}_B^G(x_B^G, y_B^G, z_B^G)$

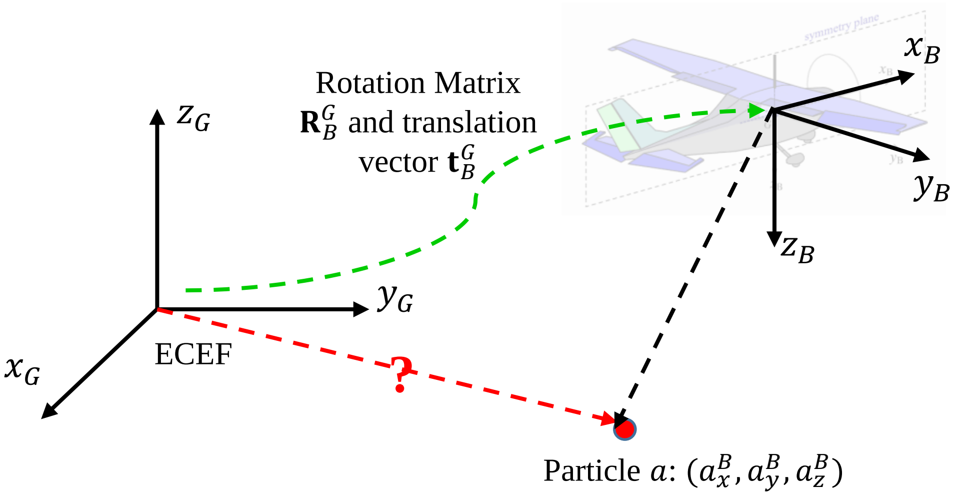
Question:

- Calculate the coordinate of particle **a** in the coordinate $\mathbf{C}_G$.

Considering both the rotation and position, we get

$$\begin{bmatrix} a_x^G \\ a_y^G \\ a_z^G \end{bmatrix} = \begin{bmatrix} e_x^G e_x^B & e_x^G e_y^B & e_x^G e_z^B \\ e_y^G e_x^B & e_y^G e_y^B & e_y^G e_z^B \\ e_z^G e_x^B & e_z^G e_y^B & e_z^G e_z^B \end{bmatrix} \begin{bmatrix} a_x^B \\ a_y^B \\ a_z^B \end{bmatrix} + \begin{bmatrix} x_B^G \\ y_B^G \\ z_B^G \end{bmatrix}$$

Position in $\mathbf{C}_G$ | **Rotation Matrix $\mathbf{R}_B^G$** | **Position in $\mathbf{C}_B$** | **$\mathbf{t}_B^G$, translation between $\mathbf{C}_G$ and $\mathbf{C}_B$**



The $\mathbf{R}_B^G$ represent the orientation and the $\mathbf{t}_B^G$ represents the position of the flight mechanic in the ECEF coordinate system!

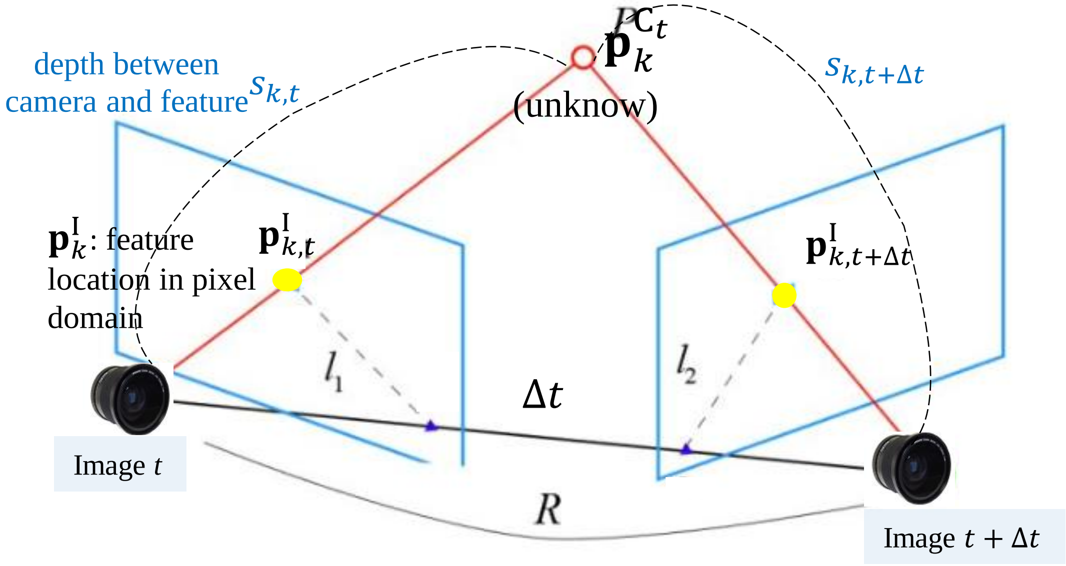
Epipolar Constraint

$$\mathbf{p}^{\mathrm{I}}_{k,t} = \begin{bmatrix} u_k \\ v_k \\ 1 \end{bmatrix}, k \in \boldsymbol{f}$$

$\mathbf{p}^{\mathrm{I}}_{k,t}$ feature location in image plane at time t

$$\mathbf{p}^{\mathrm{C}_t}_{k} = \begin{bmatrix} x^{\mathrm{C}_t}_{k,t} \\ y^{\mathrm{C}_t}_{k,t} \\ z^{\mathrm{C}_t}_{k,t} \end{bmatrix}, k \in \boldsymbol{f}$$

$\mathbf{p}^{\mathrm{C}_t}_{k}$ feature location in camera-body frame at time t



Transformation Matrix $\mathbf{T}^{\mathrm{C}_{t+\Delta t}}_{\mathrm{C}_t} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}$

R : Rotation Matrix
t : Translation Matrix
K: Camera Intrinsic Matrix

Epipolar Constraint is to estimate the relative motion $\mathbf{T}^{\mathrm{C}_{t+\Delta t}}_{\mathrm{C}_t}$ given several feature pairs!

Epipolar Constraint

$$\mathbf{p}_{k,t}^{\mathrm{I}} = \begin{bmatrix} u_k \\ v_k \\ 1 \end{bmatrix}, k \in \boldsymbol{f}$$

$$\mathbf{p}_{k}^{\mathrm{C}_t} = \begin{bmatrix} x_{k,t}^{\mathrm{C}_t} \\ y_{k,t}^{\mathrm{C}_t} \\ z_{k,t}^{\mathrm{C}_t} \end{bmatrix}, k \in \boldsymbol{f}$$

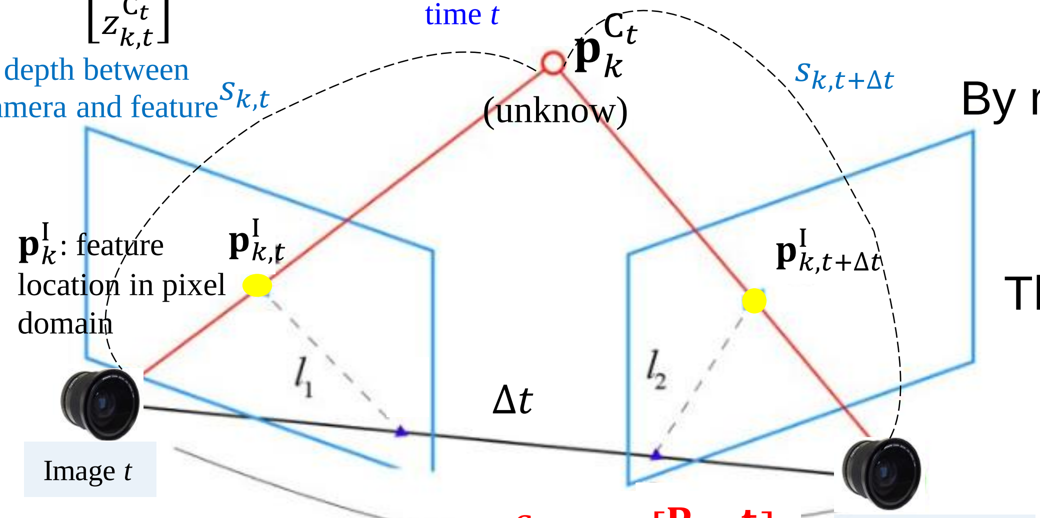
R : Rotation Matrix

t : Translation Matrix

K: Camera Intrinsic Matrix

$\mathbf{p}_k^{\mathrm{C}_t}$ feature location in camera-body frame at time t

depth between camera and feature $s_{k,t}$



Transformation Matrix $\mathbf{T}_{\mathrm{C}_t}^{\mathrm{C}_{t+\Delta t}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}$

By pinhole mode

$$s_{k,t}\mathbf{p}_{k,t}^{\mathrm{I}} = \mathbf{K}\mathbf{p}_{k,t}^{\mathrm{C}_t}$$

By transformation matrix between Δt

$$\mathbf{p}_{k,t+\Delta t}^{\mathrm{C}_{t+\Delta t}} = (\mathbf{R}\mathbf{p}_{k,t}^{\mathrm{C}_t} + \mathbf{t})$$

By measurement from feature tracking

$$\mathbf{p}_{k,t+\Delta t}^{\mathrm{I}} = \mathbf{p}_{k,t}^{\mathrm{I}} + \begin{bmatrix} \dot{u}_k \\ \dot{v}_k \end{bmatrix} \Delta t, k \in \boldsymbol{f}$$

Then, we obtain the model on $t + \Delta t$

$$s_{k,t+\Delta t}\mathbf{p}_{k,t+\Delta t}^{\mathrm{I}} = \mathbf{K}(\mathbf{R}\mathbf{p}_{k,t}^{\mathrm{C}_t} + \mathbf{t})$$

Epipolar Constraint

unknown
measurement
constant
algebra

$$\mathbf{p}_{k,t}^{\mathrm{I}} = \begin{bmatrix} u_k \\ v_k \\ 1 \end{bmatrix}, k \in \boldsymbol{f}$$

$$\mathbf{p}_{k}^{\mathrm{C}_t} = \begin{bmatrix} x_{k,t}^{\mathrm{C}_t} \\ y_{k,t}^{\mathrm{C}_t} \\ z_{k,t}^{\mathrm{C}_t} \end{bmatrix}, k \in \boldsymbol{f}$$

R : Rotation Matrix
t : Translation Matrix
K: Camera Intrinsic Matrix

$\mathbf{p}_k^{\mathrm{C}_t}$ feature location in camera-body frame at time t

depth between camera and feature

$\mathbf{p}_k^{\mathrm{I}}$: feature location in pixel domain

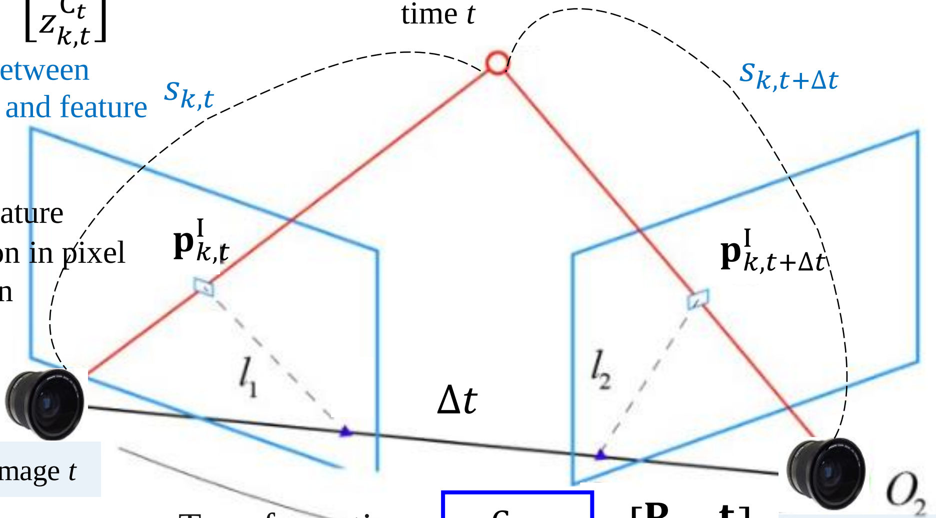


Image t

Image $t + \Delta t$

Transformation Matrix $\boxed{\mathbf{T}_{\mathrm{C}_t}^{\mathrm{C}_{t+\Delta t}}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}$

The model on $t + \Delta t$

$$s_{k,t+\Delta t}\mathbf{p}_{k,t+\Delta t}^{\mathrm{I}} = \mathbf{K}(\mathbf{R}\mathbf{p}_{k,t}^{\mathrm{C}_t} + \mathbf{t})$$

Substitution using linear algebra

$$\boxed{\left(\mathbf{K}^{-1}\mathbf{p}_{k,t+\Delta t}^{\mathrm{I}}\right)^{\mathrm{T}} \mathbf{t}^{\wedge}\mathbf{R}\mathbf{K}^{-1}\mathbf{p}_{k,t}^{\mathrm{I}} = 0}$$

The eight-point algorithm:
Eight pairs of matching feature points between two frames

$$\left[\mathbf{K}^{-1}\mathbf{p}_{k,t+\Delta t}^{\mathrm{I}}\right]^{\mathrm{T}} \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \left[\mathbf{K}^{-1}\mathbf{p}_{k,t}^{\mathrm{I}}\right] = 0$$

$\mathbf{t}^{\wedge}\mathbf{R}$

The rotation and the translation between two frames

Motion Estimation by Epipolar Constraint

$$s_{k,t+\Delta t}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t} = \mathbf{K}(\mathbf{R}\mathbf{p}^{C_t}_{k,t} + \mathbf{t})$$

$$s_{k,t+\Delta t}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t} = \mathbf{K}(\mathbf{R}\mathbf{p}^{C_t}_{k,t} + \mathbf{t}) = \mathbf{K}\mathbf{R}\mathbf{p}^{C_t}_{k,t} + \mathbf{K}\mathbf{t}$$

$$s_{k,t+\Delta t}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t} = \mathbf{K}(\mathbf{R}\mathbf{p}^{C_t}_{k,t} + \mathbf{t}) = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}s_{k,t}\mathbf{p}^{\mathrm{I}}_{k,t} + \mathbf{K}\mathbf{t}$$

$$\mathbf{K}^{-1}s_{k,t+\Delta t}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t} = \mathbf{R}\mathbf{K}^{-1}s_{k,t}\mathbf{p}^{\mathrm{I}}_{k,t} + \mathbf{t}$$

$$s_{k,t+\Delta t}\,\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t} = s_{k,t}\mathbf{R}\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t} + \mathbf{t}$$

$$\mathbf{t}^{\wedge}\,s_{k,t+\Delta t}\,\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t} = \mathbf{t}^{\wedge}s_{k,t}\mathbf{R}\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t} + \mathbf{t}^{\wedge}\mathbf{t}$$

$$\mathbf{t}^{\wedge}\,s_{k,t+\Delta t}\,\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t} = \mathbf{t}^{\wedge}s_{k,t}\mathbf{R}\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t}$$

$$s_{k,t+\Delta t}\,\mathbf{t}^{\wedge}\,\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t} = s_{k,t}\mathbf{t}^{\wedge}\mathbf{R}\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t}$$

$$s_{k,t+\Delta t}\left(\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t}\right)^{\mathrm{T}}\mathbf{t}^{\wedge}\,\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t} = s_{k,t}\left(\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t}\right)^{\mathrm{T}}\mathbf{t}^{\wedge}\mathbf{R}\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t}$$

Given the orthogonal properties:

$$\left(\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t}\right)^{\mathrm{T}}\mathbf{t}^{\wedge}\,\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t} = 0$$

$$\downarrow$$

$$s_{k,t}\left(\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t}\right)^{\mathrm{T}}\mathbf{t}^{\wedge}\mathbf{R}\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t} = 0$$

$$\downarrow$$

$$\left(\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t}\right)^{\mathrm{T}}\mathbf{t}^{\wedge}\mathbf{R}\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t} = 0$$

$$\downarrow$$

The eight-point algorithm:
Eight pairs of matching feature points between two frames

$$\downarrow$$

$$\mathbf{t}^{\wedge}\mathbf{R}$$

Motion Estimation by Epipolar Constraint

unknown
measurement
constant
algebra

The model on $t + \Delta t$

$$s_{k,t+\Delta t}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t} = \mathbf{K}(\mathbf{R}\mathbf{p}^{\mathrm{C}_t}_{k,t} + \mathbf{t})$$

Substitution using linear algebra

$$\left(\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t}\right)^{\mathrm{T}}\mathbf{t}^{\wedge}\mathbf{R}\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t} = 0$$

The eight-point algorithm:
Eight pairs of matching feature points between two frames

$$\left[\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t}\right]^{\mathrm{T}}\begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix}\left[\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t}\right] = 0$$

$\mathbf{t}^{\wedge}\mathbf{R}$

The rotation and the translation between two frames

$\mathbf{T}^{\mathrm{W}}_{\mathrm{L}}$: coordinate transformation matrix from local frame to world frame

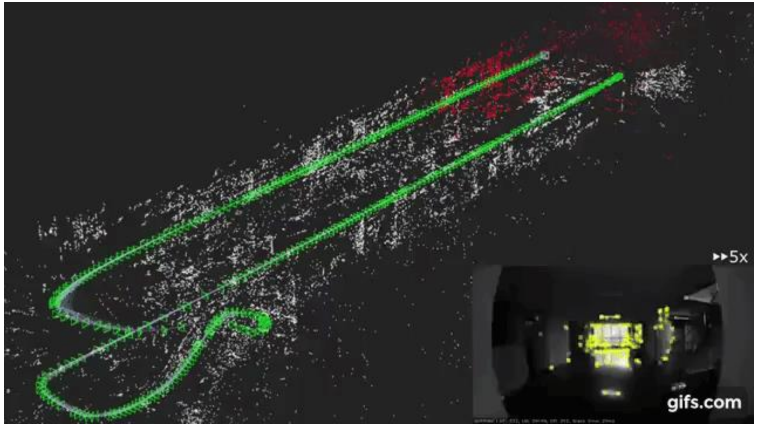
$$\mathbf{T}^{\mathrm{L}}_{\mathrm{C}} = \begin{bmatrix} \mathbf{R}^{\mathrm{L}}_{\mathrm{C}} & \mathbf{t}^{\mathrm{L}}_{\mathrm{C}} \\ 0 & 1 \end{bmatrix}$$

We get the relative motion between two images!

$$\mathbf{T}^{\mathrm{L}}_{\mathrm{C}_N} = \mathbf{T}^{\mathrm{L}}_{\mathrm{C}_0}\mathbf{T}^{\mathrm{C}_0}_{\mathrm{C}_1}\mathbf{T}^{\mathrm{C}_1}_{\mathrm{C}_2}\ldots\mathbf{T}^{\mathrm{C}_{N-1}}_{\mathrm{C}_N}$$

Motion Estimation by Epipolar Constraint

unknown
measurement
constant
algebra



$\mathbf{T}_\mathrm{L}^\mathrm{W}$: coordinate transformation matrix from local frame to world frame

$$\mathbf{T}_\mathrm{C}^\mathrm{L} = \begin{bmatrix} \mathbf{R}_\mathrm{C}^\mathrm{L} & \mathbf{t}_\mathrm{C}^\mathrm{L} \\ 0 & 1 \end{bmatrix}$$

We get the relative motion between two images!

$$\mathbf{T}_{\mathrm{C}_N}^\mathrm{L} = \mathbf{T}_{\mathrm{C}_0}^\mathrm{L}\mathbf{T}_{\mathrm{C}_1}^{\mathrm{C}_0}\mathbf{T}_{\mathrm{C}_2}^{\mathrm{C}_1}\ldots\mathbf{T}_{\mathrm{C}_N}^{\mathrm{C}_{N-1}}$$

The model on $t + \Delta t$

$$s_{k,t+\Delta t}\mathbf{p}_{k,t+\Delta t}^\mathrm{I} = \mathbf{K}(\mathbf{R}\mathbf{p}_{k,t}^{\mathrm{C}_t} + \mathbf{t})$$

Substitution using linear algebra

$$\left(\mathbf{K}^{-1}\mathbf{p}_{k,t+\Delta t}^\mathrm{I}\right)^\mathrm{T}\mathbf{t}^\wedge\mathbf{R}\mathbf{K}^{-1}\mathbf{p}_{k,t}^\mathrm{I} = 0$$

The eight-point algorithm:
Eight pairs of matching feature points between two frames

$$\left[\mathbf{K}^{-1}\mathbf{p}_{k,t+\Delta t}^\mathrm{I}\right]^\mathrm{T}\begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix}\left[\mathbf{K}^{-1}\mathbf{p}_{k,t}^\mathrm{I}\right] = 0$$

$\mathbf{t}^\wedge\mathbf{R}$ The rotation and the translation between two frames

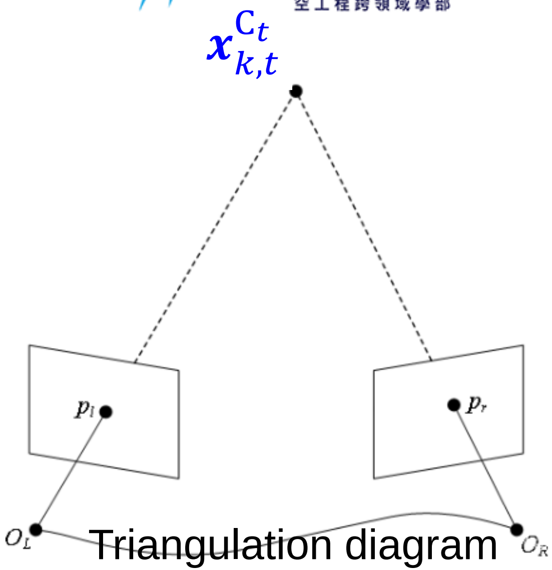
Triangulation

unknown
measurement
constant
algebra
estimated

camera motion $\mathbf{R}, \mathbf{t}$

estimate

the 3D position of feature points



Triangulation diagram

$$\mathbf{K}^{-1} s_{k,t+\Delta t}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t} = \mathbf{R}\mathbf{K}^{-1} s_{k,t}\mathbf{p}^{\mathrm{I}}_{k,t} + \mathbf{t}$$

$$\left(s_{k,t}\mathbf{p}^{\mathrm{I}}_{k,t} = \mathbf{K}\mathbf{p}^{C_t}_{k,t}\right)$$

$\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t+\Delta t}$ denotes $\boldsymbol{x}^{C_t}_{k,t+\Delta t}$

$\mathbf{K}^{-1}\mathbf{p}^{\mathrm{I}}_{k,t}$ denotes $\boldsymbol{x}^{C_t}_{k,t}$

$$s_{k,t+\Delta t}\boldsymbol{x}^{C_t}_{k,t+\Delta t} = \mathbf{R} s_{k,t}\boldsymbol{x}^{C_t}_{k,t} + \mathbf{t}$$

$$s_{k,t+\Delta t}\left(\boldsymbol{x}^{C_t}_{k,t+\Delta t}\right)^{\wedge}\boldsymbol{x}^{C_t}_{k,t+\Delta t} = s_{k,t}\left(\boldsymbol{x}^{C_t}_{k,t+\Delta t}\right)^{\wedge}\mathbf{R}\boldsymbol{x}^{C_t}_{k,t} + \left(\boldsymbol{x}^{C_t}_{k,t+\Delta t}\right)^{\wedge}\mathbf{t}$$

$$\left(\boldsymbol{x}^{C_t}_{k,t+\Delta t}\right)^{\wedge}\boldsymbol{x}^{C_t}_{k,t+\Delta t} = 0$$

Substitution

$s_{k,t}$
$s_{k,t+\Delta t}$ Depth recover

Examples of the Visual Odometry: State-of-the-art

Table I: Summary of the most representative visual (top) and visual-inertial (bottom) systems, in chronological order.

	SLAM or VO	Pixels used	Data association	Estimation	Relocalization	Loop closing	Multi Maps	Mono	Stereo	Mono IMU	Stereo IMU	Fisheye	Accuracy	Robustness	Open source
Mono-SLAM [13], [14]	SLAM	Shi Tomasi	Correlation	EKF	-	-	-	✓	-	-	-	-	Fair	Fair	[15][1]
PTAM [16]–[18]	SLAM	FAST	Pyramid SSD	BA	Thumbnail	-	-	✓	-	-	-	-	Very Good	Fair	[19]
LSD-SLAM [20], [21]	SLAM	Edgelets	Direct	PG	-	FABMAP PG	-	✓	✓	-	-	-	Good	Fair	[22]
SVO [23], [24]	VO	FAST+ Hi.grad.	Direct	Local BA	-	-	-	✓	✓	-	-	✓	Very Good	Very Good	[25][2]
ORB-SLAM2 [2], [3]	SLAM	ORB	Descriptor	Local BA	DBoW2	DBoW2 PG+BA	-	✓	✓	-	-	-	Exc.	Very Good	[26]
DSO [27]–[29]	VO	High grad.	Direct	Local BA	-	-	-	✓	✓	-	-	✓	Fair	Very Good	[30]
DSM [31]	SLAM	High grad.	Direct	Local BA	-	-	-	✓	-	-	-	-	Very Good	Very Good	[32]
MSCKF [33]–[36]	VO	Shi Tomasi	Cross correlation	EKF	-	-	-	✓	-	✓	✓	-	Fair	Very Good	[37][3]
OKVIS [38], [39]	VO	BRISK	Descriptor	Local BA	-	-	-	-	-	✓	✓	✓	Good	Very Good	[40]
ROVIO [41], [42]	VO	Shi Tomasi	Direct	EKF	-	-	-	-	-	✓	✓	✓	Good	Very Good	[43]
ORBSLAM-VI [4]	SLAM	ORB	Descriptor	Local BA	DBoW2	DBoW2 PG+BA	-	✓	-	✓	-	-	Very Good	Very Good	-
VINS-Fusion [7], [44]	VO	Shi Tomasi	KLT	Local BA	DBoW2	DBoW2 PG	✓	-	✓	✓	✓	✓	Good	Exc.	[45]
VI-DSO [46]	VO	High grad.	Direct	Local BA	-	-	-	-	-	✓	-	-	Very Good	Exc.	-
BASALT [47]	VO	FAST	KLT (LSSD)	Local BA	-	ORB BA	-	-	-	-	✓	✓	Very Good	Exc.	[48]
Kimera [8]	VO	Shi Tomasi	KLT	Local BA	-	DBoW2 PG	-	-	-	-	✓	-	Good	Exc.	[49]
ORB-SLAM3 (ours)	SLAM	ORB	Descriptor	Local BA	DBoW2	DBoW2 PG+BA	✓	✓	✓	✓	✓	✓	Exc.	Exc.	[5]

Carlos Campos received an Electronic Engineering degree (mention in Signal Processing) from INP-Toulouse and the Industrial Engineering Bachelor and M.S. degree (mention in Robotics and Computer Vision) from the University of Zaragoza. He is currently working towards the PhD. degree with the I3A Robotics, Perception and Real-Time Group. His research interests include Visual-Inertial Localization and Mapping for AR/VR applications.

Prof. Shaojie Shen | Current Members | Alumni

Prof. Shaojie Shen (沈劭劼)

Ph.D. University of Pennsylvania

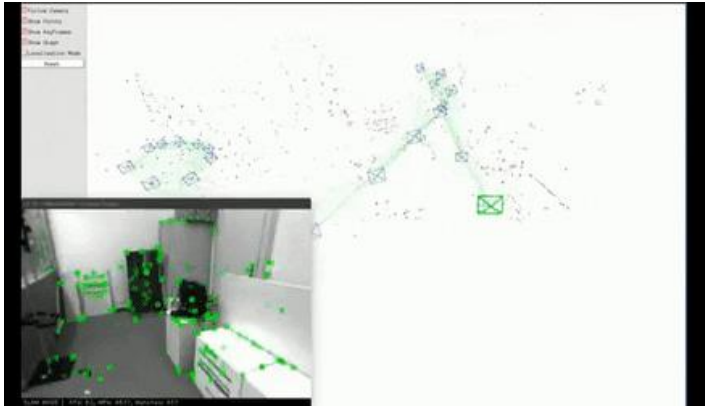
Associate Professor, Dept. of Electronic & Computer Engineering, HKUST
Director, HKUST-DJI Joint Innovation Laboratory

Shaojie Shen received his B.Eng. degree in Electronic Engineering from the Hong Kong University of Science and Technology (HKUST) in 2009. He received his M.S. in Robotics and Ph.D. in Electrical and Systems Engineering in 2011 and 2014, respectively, all from the University of Pennsylvania. He joined the Department of Electronic and Computer Engineering at the HKUST in September 2014 as an Assistant Professor, and is promoted to Associate Professor in July 2020. He is the founding director of the HKUST-DJI Joint Innovation Laboratory (HDJI Lab). His research interests are in the areas of robotics and unmanned aerial vehicles, with focus on state estimation, sensor fusion, localization and mapping, and autonomous navigation in complex environments. He was the regional program chair of SSRR 2017 and program co-chair of SSRR 2015. He is currently serving as associate editor for T-RO and ICRA 2019-2022, and senior editor for IROS 2020-2022.

He and his research team received **Honorable Mention status for the IEEE T-RO Best Paper Award** in 2020 and 2018, and won the **Best Student Paper Award** in IROS 2018, **Best Service Robotics Paper Finalist** in ICRA 2017, **Best Paper Finalist** in ICRA 2011, and **Best Paper Awards** in SSRR 2016 and SSRR 2015. In 2020, Prof Shen receives the **AI 2000 Most Influential Scholar Award Honorable Mention**, and he received this award again in 2021.

Campos, Carlos, Richard Elvira, Juan J. Gómez Rodríguez, José MM Montiel, and Juan D. Tardós. "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam." *IEEE Transactions on Robotics* 37, no. 6 (2021): 1874-1890.

Examples of the Visual Odometry: State-of-the-art



ORB-SLAM

Related Publications:

[ORB-SLAM3] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel and Juan D. Tardós, **ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM**, *IEEE Transactions on Robotics 37(6):1874-1890, Dec. 2021*. PDF.

[IMU-Initialization] Carlos Campos, J. M. M. Montiel and Juan D. Tardós, **Inertial-Only Optimization for Visual-Inertial Initialization**, *ICRA 2020*. PDF

[ORBSLAM-Atlas] Richard Elvira, J. M. M. Montiel and Juan D. Tardós, **ORBSLAM-Atlas: a robust and accurate multi-map system**, *IROS 2019*. PDF.

[ORBSLAM-VI] Raúl Mur-Artal, and Juan D. Tardós, **Visual-inertial monocular SLAM with map reuse**, IEEE Robotics and Automation Letters, vol. 2 no. 2, pp. 796-803, 2017. PDF.

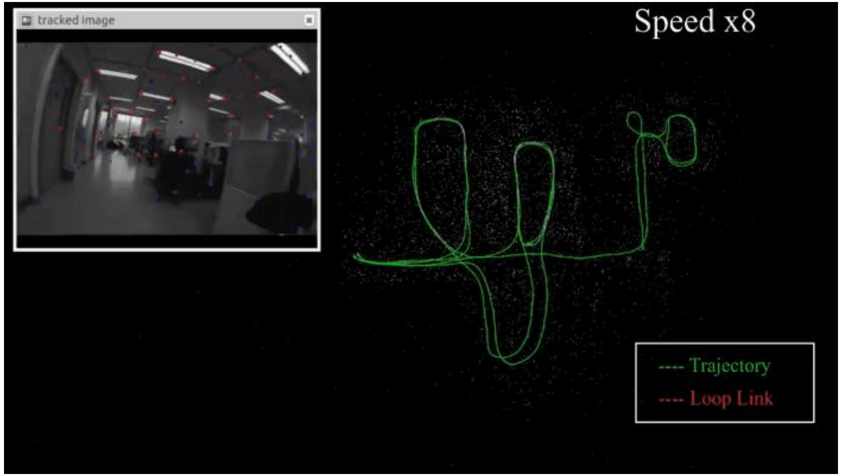
[Stereo and RGB-D] Raúl Mur-Artal and Juan D. Tardós. **ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras**. *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255-1262, 2017. PDF.

[Monocular] Raúl Mur-Artal, José M. M. Montiel and Juan D. Tardós. **ORB-SLAM: A Versatile and Accurate Monocular SLAM System**. *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, 2015. (**2015 IEEE Transactions on Robotics Best Paper Award**). PDF.

[DBoW2 Place Recognition] Dorian Gálvez-López and Juan D. Tardós. **Bags of Binary Words for Fast Place Recognition in Image Sequences**. *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188-1197, 2012. PDF

Campos, Carlos, Richard Elvira, Juan J. Gómez Rodríguez, José MM Montiel, and Juan D. Tardós. "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam." *IEEE Transactions on Robotics* 37, no. 6 (2021): 1874-1890.

Examples of the Visual Odometry: State-of-the-art



VINS-Mono

2018

X. Lyu, H. Gu, J. Zhou, Z. Li, S. Shen and F. Zhang. Simulation and flight experiments of a quadrotor tail-sitter vertical take-off and landing unmanned aerial vehicle with wide flight envelope. *International Journal of Micro Air Vehicles*, 10(4), pp. 303-317, December 2018.

F. Gao, W. Wu, W. Gao and S. Shen. Flying on point clouds: online trajectory generation and autonomous navigation for quadrotor in cluttered environments. *Journal of Field Robotics*, 36(4), pp. 710-733, December 2018. video

S. Chung, A. Paranjape, P. Dames, S. Shen and V. Kumar. A survey on aerial swarm robotics. *IEEE Transactions on Robotics*, 34(4), pp. 837-855, August 2018.

T. Qin, P. Li and S. Shen. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4), pp. 1004-1020, July 2018.

X. Lyu, J. Zhou, H. Gu, Z. Li, S. Shen and F. Zhang. Disturbance observer based hovering control of quadrotor tail-sitter VTOL UAVs using H-infinity synthesis. *IEEE Robotics and Automation Letters*, 3(4), pp. 2910-2917, June 2018. video

2017

T. Liu and S. Shen. Spline-based initialization of monocular visual-inertial state estimators at high altitude. *IEEE Robotics and Automation Letters*, 2(4), pp. 2224-2231, July 2017. video

Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang and S. Shen. Autonomous aerial navigation using monocular visual-inertial fusion. *Journal of Field Robotics*, 35(1), pp. 23-51, July 2017. video

Y. Ling, M. Kuse and S. Shen. Edge alignment-based visual-inertial fusion for tracking of aggressive motions. *Autonomous Robots*, 42(3), pp. 513-528, July 2017. video

K. Qiu, T. Liu and S. Shen. Model-based global localization for aerial robots using edge alignment. *IEEE Robotics and Automation Letters*, 2(3), pp. 1256-1263, January 2017. video

2016

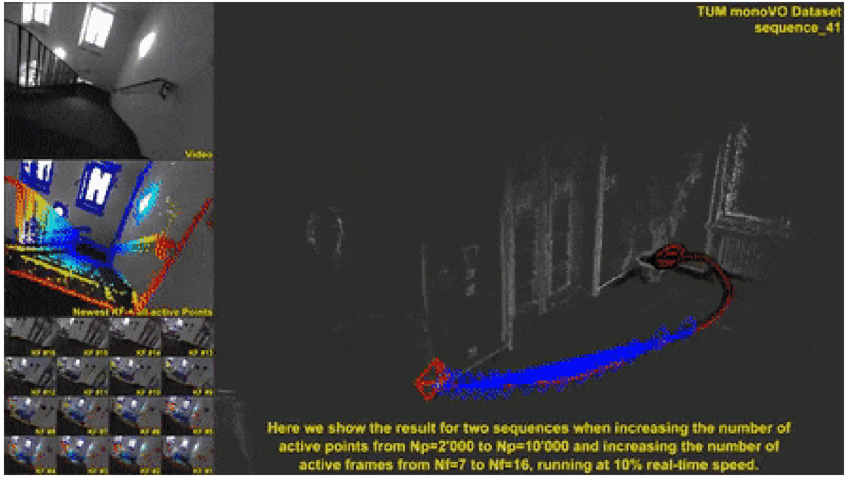
Z. Yang and S. Shen. Monocular visual-inertial state estimation with online initialization and camera-IMU extrinsic calibration. *IEEE Transactions on Automation Science and Engineering*, 14(1), pp. 39-51, April 2016. video

2013

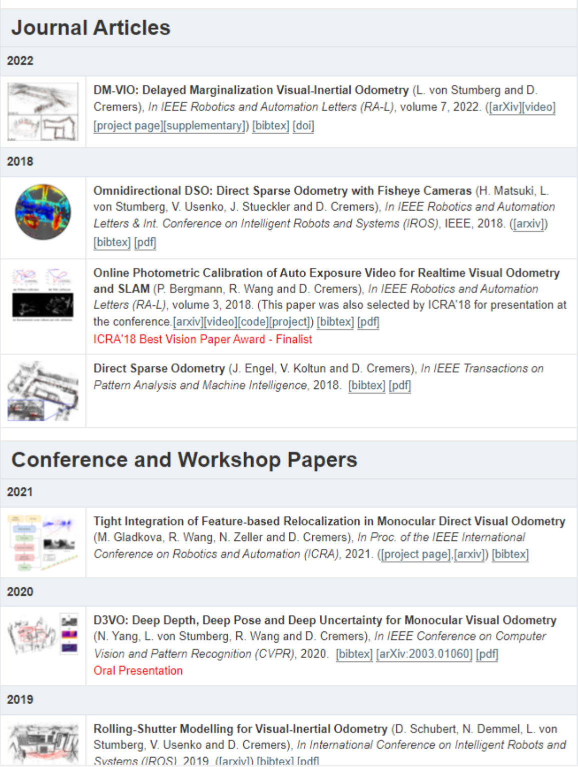
S. Shen, N. Michael and V. Kumar. Obtaining liftoff indoors: autonomous navigation in confined indoor environments. *IEEE Robotics and Automation Magazine*, 20(4), pp. 40-48, December 2013.

Qin, T., Li, P. and Shen, S., 2018. Vins-mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics, 34(4), pp.1004-1020.

Examples of the Visual Odometry: State-of-the-art



DSO: Direct Sparse Odometry



Journal Articles

2022

DM-VIO: Delayed Marginalization Visual-Inertial Odometry (L. von Stumberg and D. Cremers), *In IEEE Robotics and Automation Letters (RA-L)*, volume 7, 2022. ([arXiv][video][project page][supplementary]) [bibtex] [doi]

2018

Omnidirectional DSO: Direct Sparse Odometry with Fisheye Cameras (H. Matsuki, L. von Stumberg, V. Usenko, J. Stueckler and D. Cremers), *In IEEE Robotics and Automation Letters & Int. Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018. ([arxiv]) [bibtex] [pdf]

Online Photometric Calibration of Auto Exposure Video for Realtime Visual Odometry and SLAM (P. Bergmann, R. Wang and D. Cremers), *In IEEE Robotics and Automation Letters (RA-L)*, volume 3, 2018. (This paper was also selected by ICRA'18 for presentation at the conference.[arxiv][video][code][project]) [bibtex] [pdf]
ICRA'18 Best Vision Paper Award - Finalist

Direct Sparse Odometry (J. Engel, V. Koltun and D. Cremers), *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. [bibtex] [pdf]

Conference and Workshop Papers

2021

Tight Integration of Feature-based Relocalization in Monocular Direct Visual Odometry (M. Gladkova, R. Wang, N. Zeller and D. Cremers), *In Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021. ([project page],[arxiv]) [bibtex]

2020

D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry (N. Yang, L. von Stumberg, R. Wang and D. Cremers), *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [bibtex] [arXiv:2003.01060] [pdf]
Oral Presentation

2019

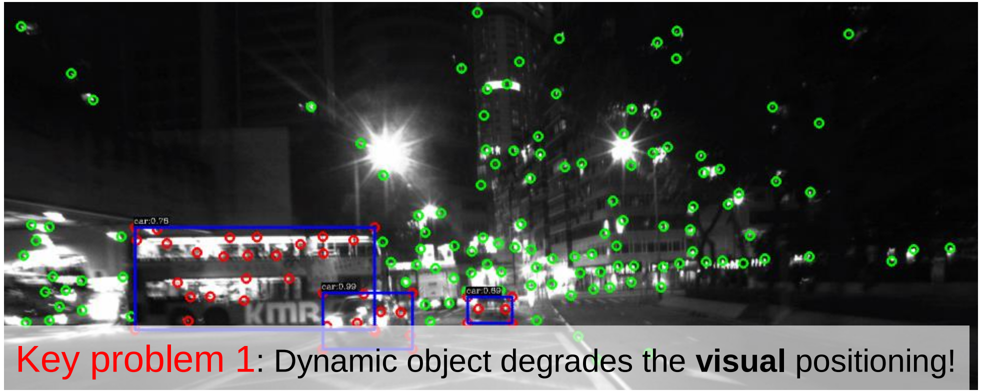
Rolling-Shutter Modelling for Visual-Inertial Odometry (D. Schubert, N. Demmel, L. von Stumberg, V. Usenko and D. Cremers), *In International Conference on Intelligent Robots and Systems (IROS)*, 2019. ([arxiv]) [bibtex] [pdf]

Engel, J., Koltun, V. and Cremers, D., 2017. Direct sparse odometry. IEEE transactions on pattern analysis and machine intelligence, 40(3), pp.611-625.

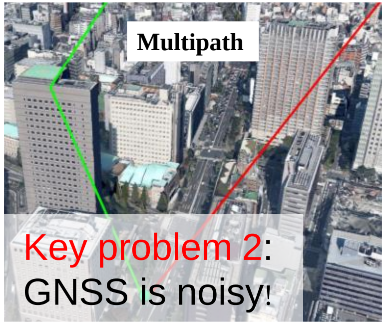
Performance and Challenges of Visual Positioning in Autonomous Driving

Challenges of Visual Positioning in Urban Areas

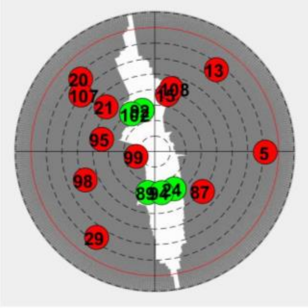
Tall/dense buildings!
Dense traffic!



Key problem 1: Dynamic object degrades the **visual** positioning!



Key problem 2:
GNSS is noisy!



GNSS positioning is challenged due to signal blockage and reflection!

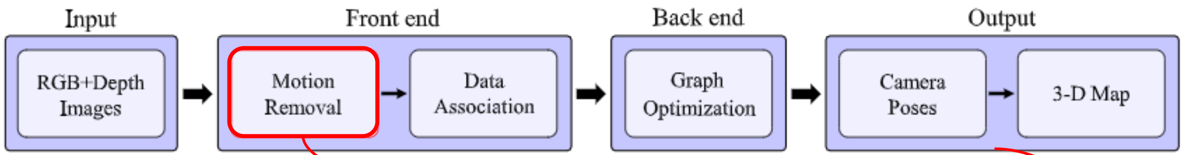
IMU is subject to severe drift in dense traffic scenarios!

Dynamic object degrades the visual positioning!

[1] Sun, Pei, et al. "Scalability in perception for autonomous driving: Waymo open dataset." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

Challenges of Visual Positioning

Principle: identify the features or pixels that are associated with moving objects using **motion tracking**



Pixel-wisely Motion segmentation to find the possible moving objects

Limitations: the algorithm would fail in the scenarios that more than half of pixels come from moving objects, require depth information.

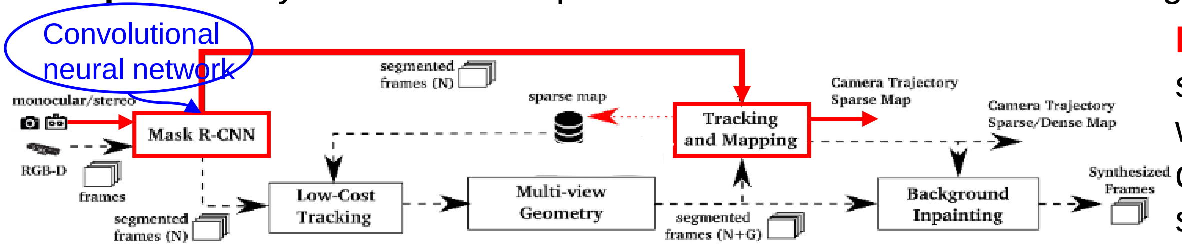
ORB-SLAM

ORB-SLAM with proposed method [1]

[1] Sun Y, Liu M, Meng M Q H. **Motion removal for reliable RGB-D SLAM in dynamic environments**[J]. Robotics and Autonomous Systems, 2018, 108:115-128.

Challenges of Visual Positioning

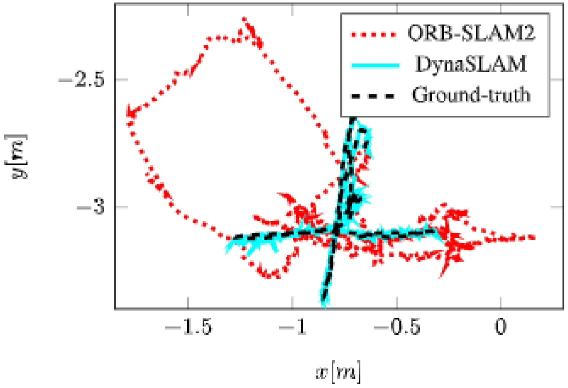
Principle: identify the features or pixels that are associated with moving objects using **deep learning**



Limitations: the proposed R-CNN is a supervised method, the detector model would fail when there are significant differences between training and practical scenes

Input images

Output images with proposed method [2]

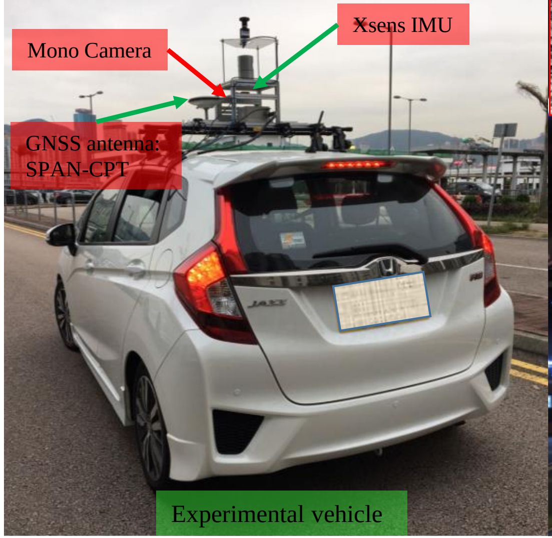


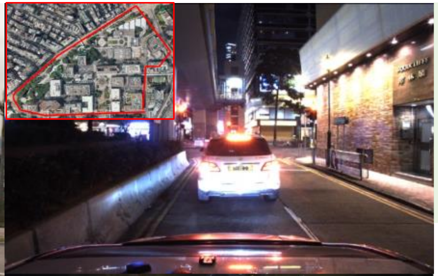
Ground truth and trajectories estimated by **DynaSLAM**[2] and ORB-SLAM2 in the dynamic object dataset

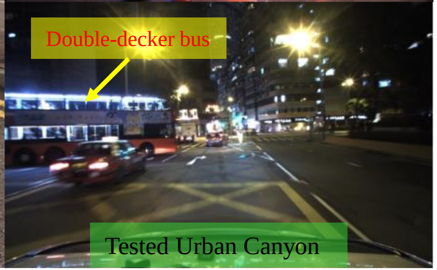
[2] Bescos B, Fácil J M, Civera J, et al. **DynaSLAM**: Tracking, mapping, and inpainting in dynamic scenes[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 4076-4083.

Experimental of Visual Positioning in Hong Kong

Xsens MTi 10 IMU sensor is used to collect raw IMU measurements (200hz), monocular camera is employed to capture raw images (10hz)， NovAtel SPAN-CPT provides the ground truth of positioning







Tsim Sha Tsui East
Trajectory: 1984.448 meters

VINS[1]: positioning from VINS
VINS-R: positioning from VINS with DFP removal
VINS-M: positioning from VINS with DFP remodeling

DFP: dynamic feature point

[1] Qin, Tong, Peiliang Li, and Shaojie Shen. "Vins-mono: A robust and versatile monocular visual-inertial state estimator." *IEEE Transactions on Robotics* 34.4 (2018): 1004-1020.

Preview on Tutorial 2: Tutorial on Visual Positioning

AAE4203 – Guidance and Navigation

Dr Weisong Wen
Research Assistant Professor
Department of Aeronautical and Aviation Engineering
The Hong Kong Polytechnic University

Week 9, 16 March 2022

How to all these unknow parameters?

The process of calculate all these parameters is called camera calibration!

Iterms	param1	param2	param3	param4	param5
Camera Intrinsic-K	f_u	f_v	Δu	Δv	
Lens Distortion	K1	K2	K3	p1	p2

Camera Calibration

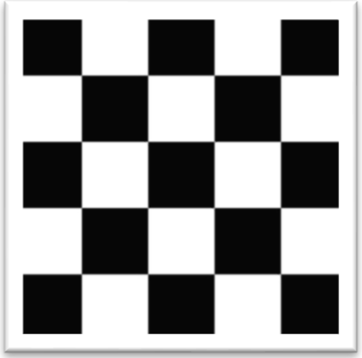
Calibration of camera

Algorithm: Zhang Zhengyou Calibration[1]

Advantages:
The equipment is simple, just a printed checkerboard;
High precision, relative error can be lower than 0.3%;

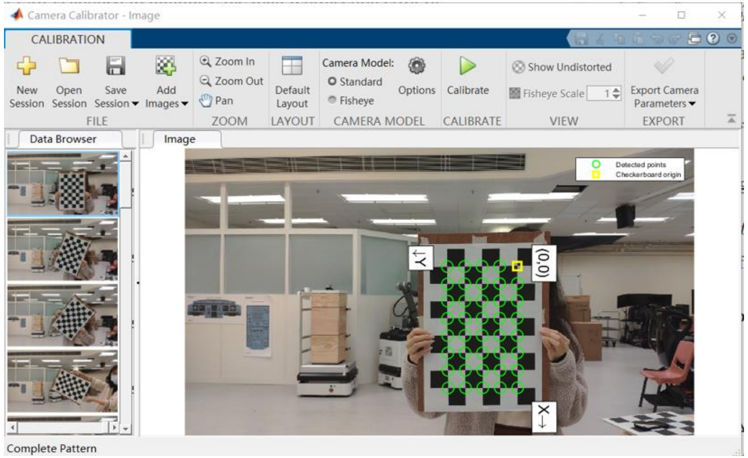
He received the IEEE Helmholtz Time Test Award for "Zhang's Calibration Method“ in 2013

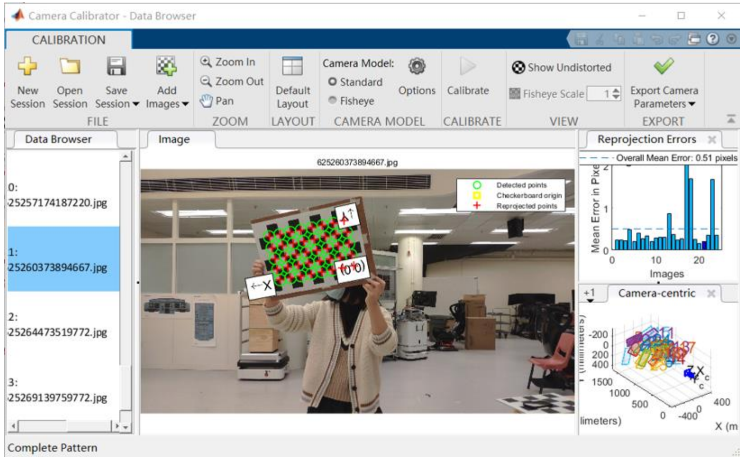
A very famous expert in computer vision and multimedia technology



[1] Zhang, Zhengyou. "A flexible new technique for camera calibration." *IEEE Transactions on pattern analysis and machine intelligence* 22.11 (2000): 1330-1334.

Camera Calibration



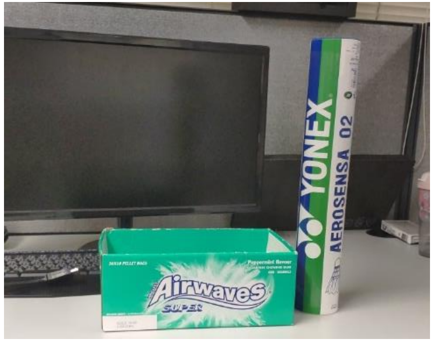


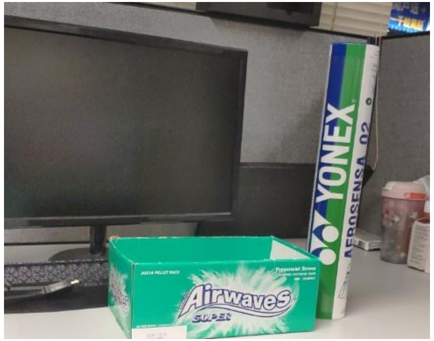
cameraParams | cameraParams.Intrinsics

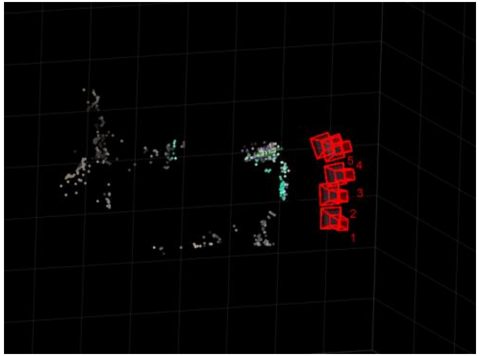
cameraParams.Intrinsics

屬性	值
FocalLength	[1.0054e+03,1.0032e+03]
PrincipalPoint	[654.6171,370.9098]
ImageSize	[720,1280]
RadialDistortion	[0.1889,-0.3517]
TangentialDistortion	[0,0]
Skew	0
IntrinsicMatrix	[1.0054e+03,0,0;0,1.0032e+03,0;654.6171,370.9098,1]

Visual Odometry







GNSS Real-time Kinematic Positioning: Ambiguity Resolution

$\mathbf{p}_{r,t}^{G}$: Float solution of position of GNSS receiver

$\mathbf{a}_t = \Delta\nabla N_{r,t}^{1}, \Delta\nabla N_{r,t}^{2}, \ldots$: Float ambiguity

$\mathbf{Q}_t = \left({\mathbf{G}_t^{G}}^{T}\mathbf{W}_t\mathbf{G}_t^{G}\right)^{-1}$: Covariance matrix

Formulate the Covariance $\mathbf{W}_t$!

$$
\mathbf{G}_t^{G} = \begin{bmatrix}
\frac{p_{t,x}^{G,1} - p_{\mathbf{r},t,x}^{G}}{\left\|\mathbf{p}_t^{G,1} - \mathbf{p}_{r,t}^{G}\right\|} - \frac{p_{t,x}^{G,w} - p_{\mathbf{r},t,x}^{G}}{\left\|\mathbf{p}_t^{G,w} - \mathbf{p}_{r,t}^{G}\right\|} & \frac{p_{t,y}^{G,1} - p_{\mathbf{r},t,y}^{G}}{\left\|\mathbf{p}_t^{G,1} - \mathbf{p}_{r,t}^{G}\right\|} - \frac{p_{t,y}^{G,w} - p_{\mathbf{r},t,y}^{G}}{\left\|\mathbf{p}_t^{G,w} - \mathbf{p}_{r,t}^{G}\right\|} & \frac{p_{t,z}^{G,1} - p_{\mathbf{r},t,z}^{G}}{\left\|\mathbf{p}_t^{G,1} - \mathbf{p}_{r,t}^{G}\right\|} - \frac{p_{t,z}^{G,w} - p_{\mathbf{r},t,z}^{G}}{\left\|\mathbf{p}_t^{G,w} - \mathbf{p}_{r,t}^{G}\right\|} & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{p_{t,x}^{G,m-1} - p_{\mathbf{r},t,x}^{G}}{\left\|\mathbf{p}_t^{G,m-1} - \mathbf{p}_{r,t}^{G}\right\|} - \frac{p_{t,x}^{G,w} - p_{\mathbf{r},t,x}^{G}}{\left\|\mathbf{p}_t^{G,w} - \mathbf{p}_{r,t}^{G}\right\|} & \frac{p_{t,y}^{G,m-1} - p_{\mathbf{r},t,y}^{G}}{\left\|\mathbf{p}_t^{G,m-1} - \mathbf{p}_{r,t}^{G}\right\|} - \frac{p_{t,y}^{G,w} - p_{\mathbf{r},t,y}^{G}}{\left\|\mathbf{p}_t^{G,w} - \mathbf{p}_{r,t}^{G}\right\|} & \frac{p_{t,z}^{G,m-1} - p_{\mathbf{r},t,z}^{G}}{\left\|\mathbf{p}_t^{G,m-1} - \mathbf{p}_{r,t}^{G}\right\|} - \frac{p_{t,z}^{G,w} - p_{\mathbf{r},t,z}^{G}}{\left\|\mathbf{p}_t^{G,w} - \mathbf{p}_{r,t}^{G}\right\|} & 0 & \cdots & 0 \\
\frac{p_{t,x}^{G,1} - p_{\mathbf{r},t,x}^{G}}{\left\|\mathbf{p}_t^{G,1} - \mathbf{p}_{r,t}^{G}\right\|} - \frac{p_{t,x}^{G,w} - p_{\mathbf{r},t,x}^{G}}{\left\|\mathbf{p}_t^{G,w} - \mathbf{p}_{r,t}^{G}\right\|} & \frac{p_{t,y}^{G,1} - p_{\mathbf{r},t,y}^{G}}{\left\|\mathbf{p}_t^{G,1} - \mathbf{p}_{r,t}^{G}\right\|} - \frac{p_{t,y}^{G,w} - p_{\mathbf{r},t,y}^{G}}{\left\|\mathbf{p}_t^{G,w} - \mathbf{p}_{r,t}^{G}\right\|} & \frac{p_{t,z}^{G,1} - p_{\mathbf{r},t,z}^{G}}{\left\|\mathbf{p}_t^{G,1} - \mathbf{p}_{r,t}^{G}\right\|} - \frac{p_{t,z}^{G,w} - p_{\mathbf{r},t,z}^{G}}{\left\|\mathbf{p}_t^{G,w} - \mathbf{p}_{r,t}^{G}\right\|} & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{p_{t,x}^{G,m-1} - p_{\mathbf{r},t,x}^{G}}{\left\|\mathbf{p}_t^{G,m-1} - \mathbf{p}_{r,t}^{G}\right\|} - \frac{p_{t,x}^{G,w} - p_{\mathbf{r},t,x}^{G}}{\left\|\mathbf{p}_t^{G,w} - \mathbf{p}_{r,t}^{G}\right\|} & \frac{p_{t,y}^{G,m-1} - p_{\mathbf{r},t,y}^{G}}{\left\|\mathbf{p}_t^{G,m-1} - \mathbf{p}_{r,t}^{G}\right\|} - \frac{p_{t,y}^{G,w} - p_{\mathbf{r},t,y}^{G}}{\left\|\mathbf{p}_t^{G,w} - \mathbf{p}_{r,t}^{G}\right\|} & \frac{p_{t,z}^{G,m-1} - p_{\mathbf{r},t,z}^{G}}{\left\|\mathbf{p}_t^{G,m-1} - \mathbf{p}_{r,t}^{G}\right\|} - \frac{p_{t,z}^{G,w} - p_{\mathbf{r},t,z}^{G}}{\left\|\mathbf{p}_t^{G,w} - \mathbf{p}_{r,t}^{G}\right\|} & 0 & \cdots & 1
\end{bmatrix}
$$

Q&A

Thank you for your attention ☺

Q&A

Dr. Weisong Wen

If you have any questions or inquiries, please feel free to contact me.

Email: welson.wen@polyu.edu.hk