StructVIO: Visual-Inertial Odometry with Structural Regularity of Man-Made Environments

Danping Zou

VALSE online seminar
2019年7月10日

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Visual SLAM

- Visual SLAM techniques have been widely applied to unmanned vehicles.

Fisheye camera

Stereo camera

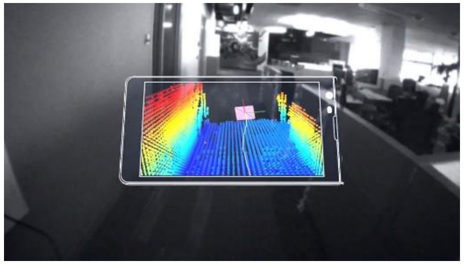
Stereo camera

Visual SLAM

- Augmented reality (AR) (Hololens Glass，Project Tango Tablet)

Hololens uses four cameras for visual SLAM

Tango use one fisheye camera for visual SLAM



Visual SLAM

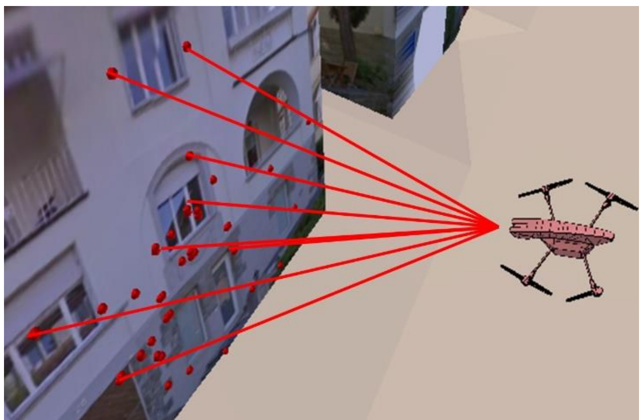
- Operation system on cellphones
  - Google and Apple integrate visual SLAM into their OS (iOS, Android).

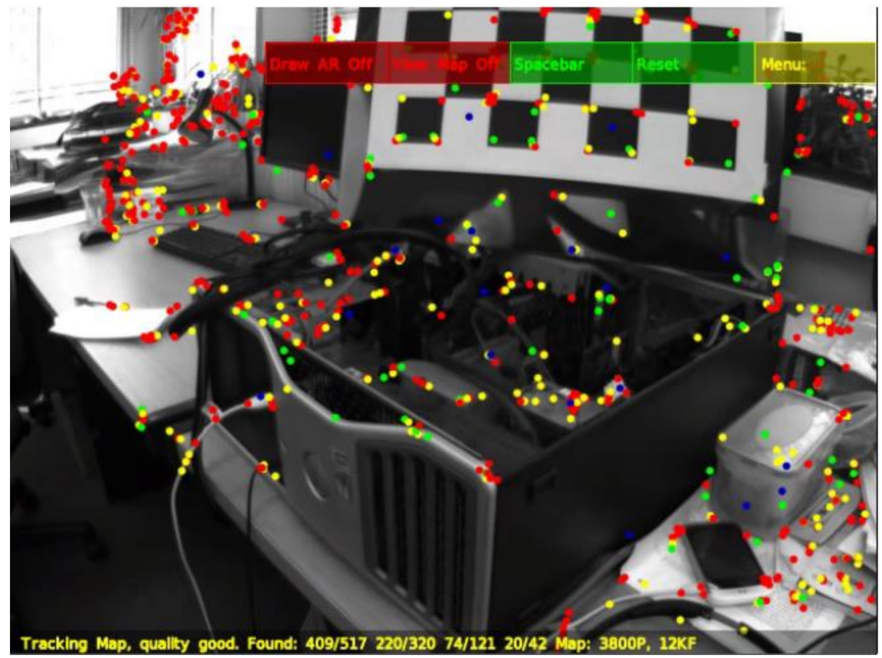
Visual SLAM

- A lot of algorithms have been proposed for visual SLAM in the past 15 years.
  - MonoSLAM (2003), StructSLAM(2014)
  - PTAM(2007), ORB-SLAM(2015)
  - SVO(2014), LSD-SLAM(2014), DSO(2016)
- Pure visual SLAM system is not robust in practical applications.
- Visual-inertial systems become predominant for real applications.
  - MSCKF (2007), ROVIO (2009)
  - OKVIS (2015), VINS(2017), ICE-BA(2018)

Features in v/vi-SLAM systems

- Most visual-SLAM or visual-inertial systems choose points as the landmarks.





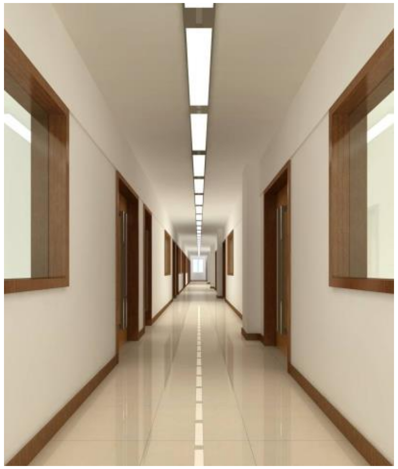
Features in v/vi-SLAM systems

- Man made environments exhibit **strong regularity on geometry**.

Natural scenes

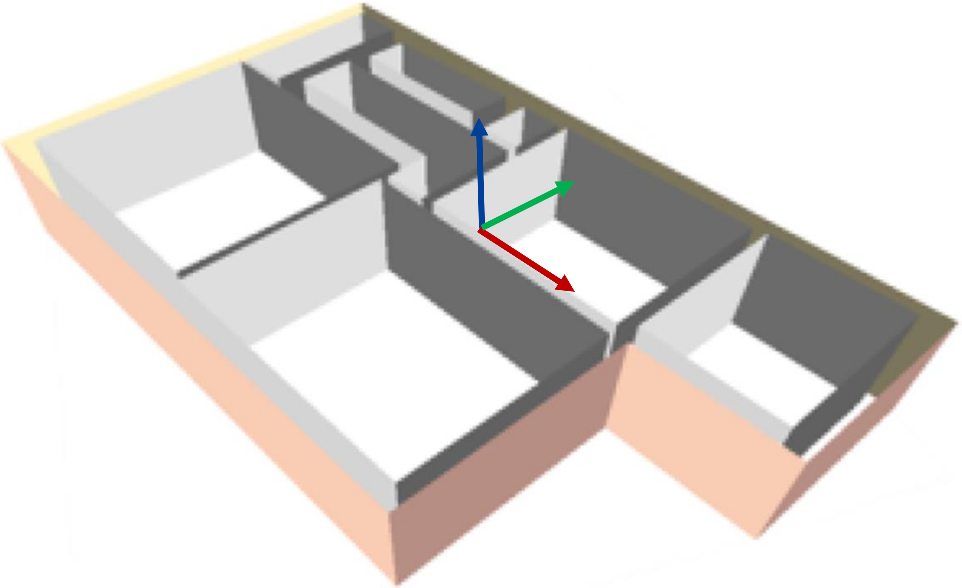
Street

Underground parking



Indoor

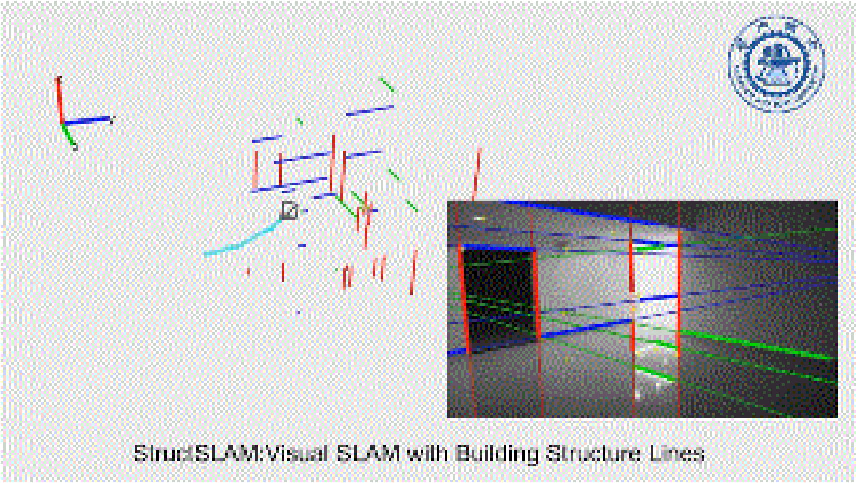
Structural regularity - Manhattan word



1. Rich of line features
2. Three known directions (x, y, z)

Visual SLAM with Manhattan world model

- StructSLAM (Presented VALSE online seminar, 2016, 30th,Mar)
  - Point + structural lines (lines aligned with x, y, z directions)
  - The direction of lines improves the observability of camera orientation



Zhou, Huizhong, Danping, Zou, et al. "StructSLAM: Visual SLAM with building structure lines." *Vehicular Technology, IEEE Transactions on* 64.4 (2015): 1364-1375. - Special session for indoor localization

Real word is full of diversity

- A lot of man made environments can not be well described by Manhattan world model.
- Oblique/curvy structures.

StructVIO

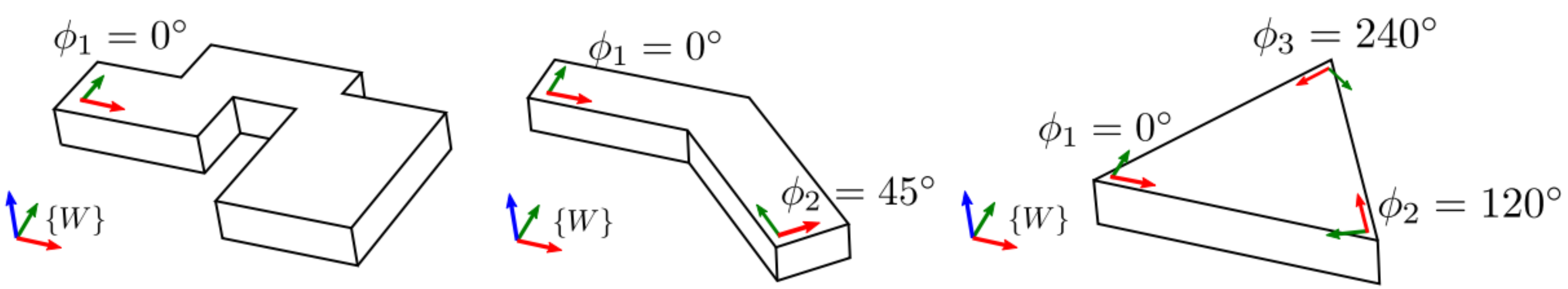
- A novel visual-inertial odometry method is presented
  - Use **Atlanta world** model to better describe irregular scenes.
  - Made **several improvements** to existing VIO approach.
  - A VIO dataset that can be used evaluate different methods.

Zou, Danping, et al. "StructVIO: Visual-inertial Odometry with Structural Regularity of Man-made Environments." IEEE Trans. on Robotics, 2019

Executable, tools & dataset : http://drone.sjtu.edu.cn/dpzou/project/structvio.html

Key idea – Atlanta world model

- We can approximate an irregular world by **a group of local Manhattan worlds.**
- Each one of them can be represented by a heading direction : ϕ.



One Manhattan world   Two Manhattan worlds   Three Manhattan worlds

Key idea – Atlanta world model

- Locally, the world is a Manhattan world. We can still use
  - Three directions
  - Structural line features
- to improve the performance of the VIO system.

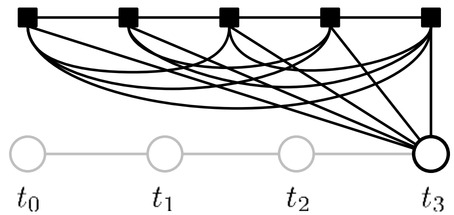
Three directions → X,Y directions – Render the Yaw angle observable (locally)

Three directions → Z direction – Render the gravity direction observable

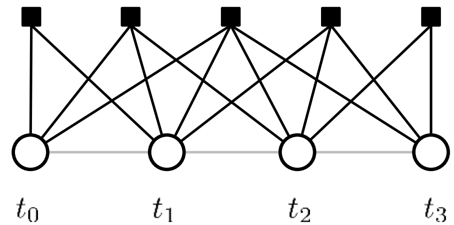
Line features → A good complementary to point features in texture-less scenes.

The framework of StructVIO

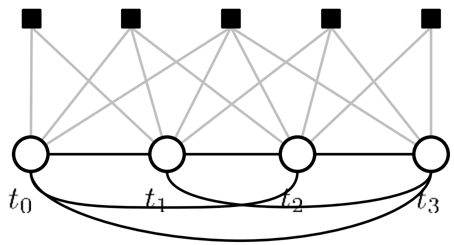
- We adopt the multi-state EKF filter based framework.
- Comparing with classic EKF filter
  - Much faster since the features are not included in the state vector.
- Comparing with key-frame optimization
  - Short feature trajectories are fully explored.
  - State update using a single feature trajectory.
  - Efficient but without losing much accuracy.



Classic EKF filter



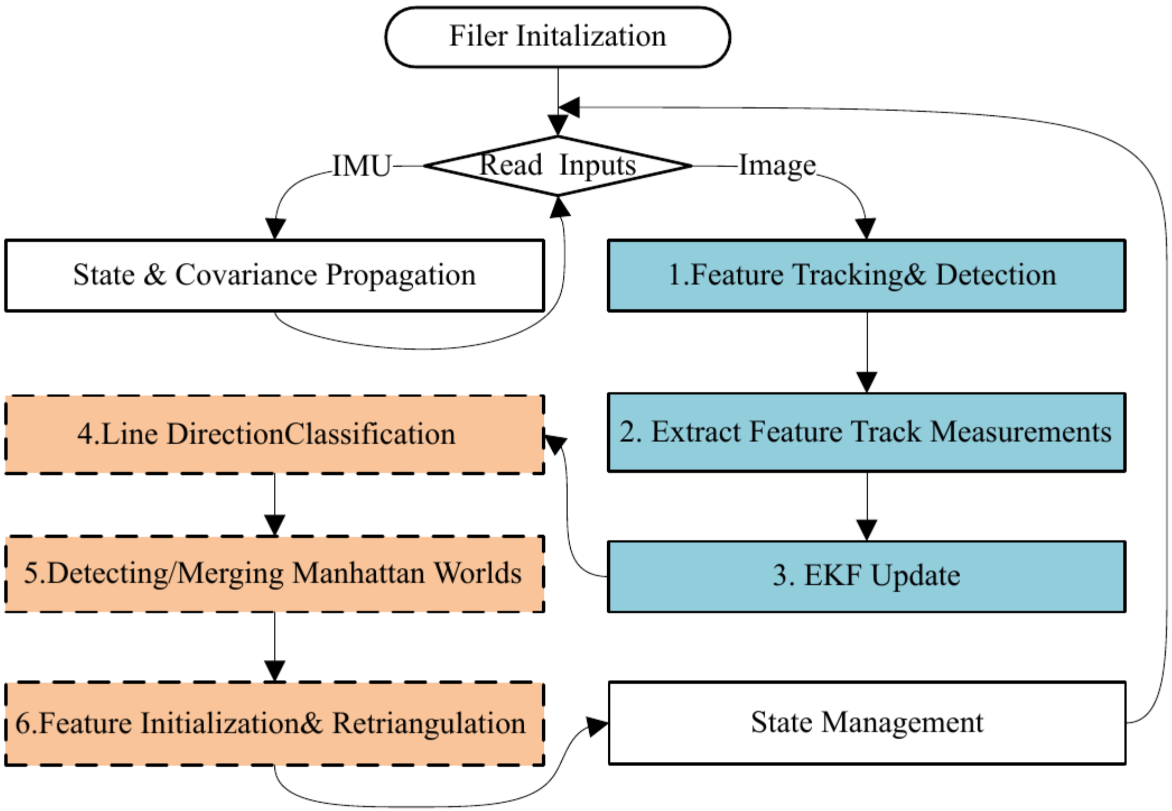
Key-frame optimization



Multi-state EKF filter

The framework of StructVIO

- The pipeline of StructVIO is as the following:



State definition of StructVIO

- The state vector consists of the *current IMU state*, *historical IMU poses, calibration parameters*, and the *heading directions* of local Manhattan worlds

$$x_k = [x_{I_k}, {}^{I}_{C}\tau, \boxed{\phi_1, \cdots \phi_N}, \boxed{{}^{W}_{I_1}\tau, \cdots {}^{W}_{I_M}\tau}]$$

$$x_{I_k} = [{}^{W}_{I_k}\tau, {}^{W}v_{I_k}, b_{g_k}, b_{a_k}]$$

Current IMU state

Camera-IMU calibration

Manhattan worlds

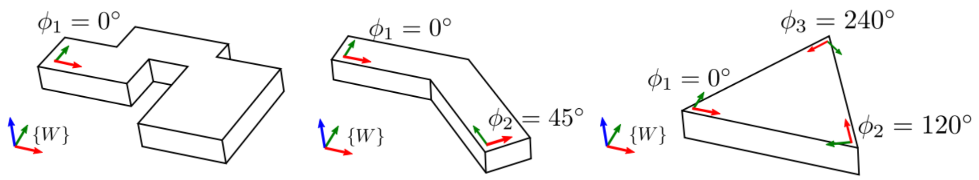
Historical IMU poses

StructVIO – Technical details

- Inside of the filter
  - Parameterization
  - Measurement equation
- Outside of the filter
  - Structural line related tasks:
    - Line detection & tracking
    - Classification of structural lines
    - initialization & triangulation
    - Handling long feature tracks
  - Manhattan world :
    - Detection
    - Merging
- Other details
  - Outlier rejection

Coordinate frames

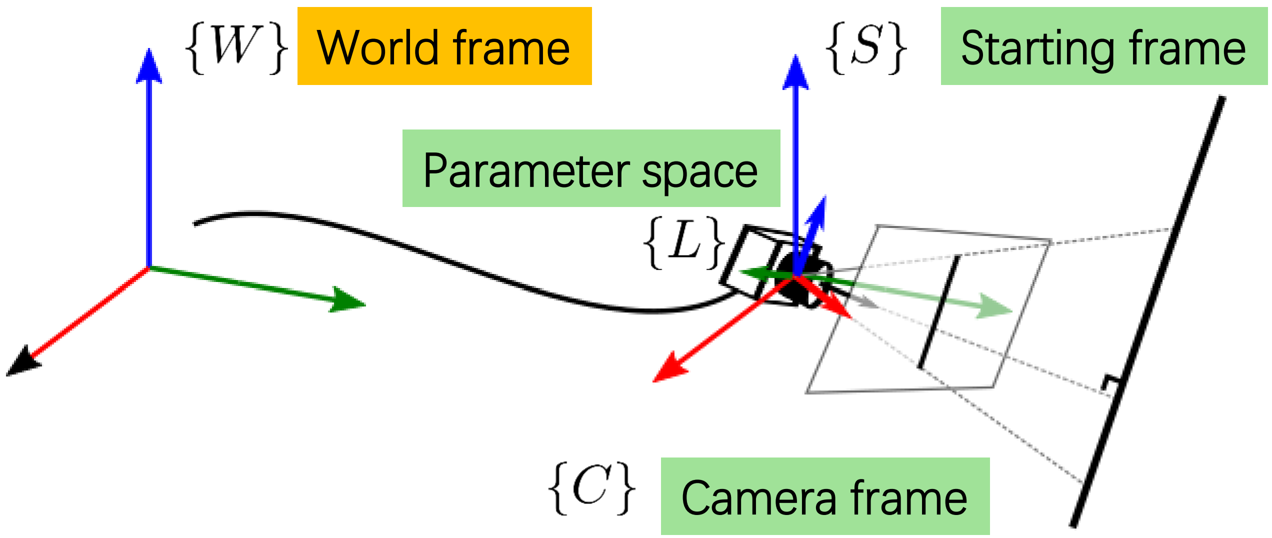
- World frame : $\{W\}$
  - Z axis aligned with gravity
  - Starting point as the origin



- Local Manhattan frame: $\phi_i \in [0, \pi/2)\,, i = 1, \ldots, N$
- Camera frame : $\{C\}$
  - Z axis aligned with the optical axis toward the viewing direction.
  - X, Y axes aligned with x,y axes of the image
- Starting frame: $\{S\}$ - Moving Manhattan frame
  - The origin is located at the camera center.
  - Three axes aligned with those of local Manhattan frame

Representation of a structural line

- We use a **camera-centric** representation.



- Parameter space : $\{L\}$ - use for line representation

Structural line parameter space

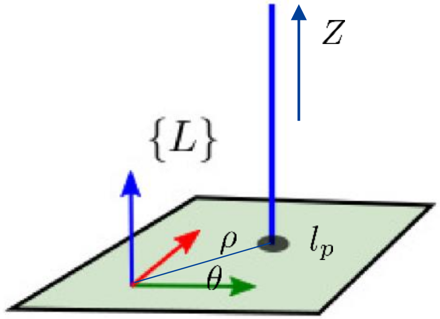
- In parameter space $\{L\}$, a structural line can be represented by a point and a vertical direction.

$$l_p \sim (a, b, 0)^{\mathrm{T}} \quad Z = (0, 0, 1)^{\mathrm{T}}$$

- To achieve better linearization, the intersection point can be represented using inverse-depth approach. We have

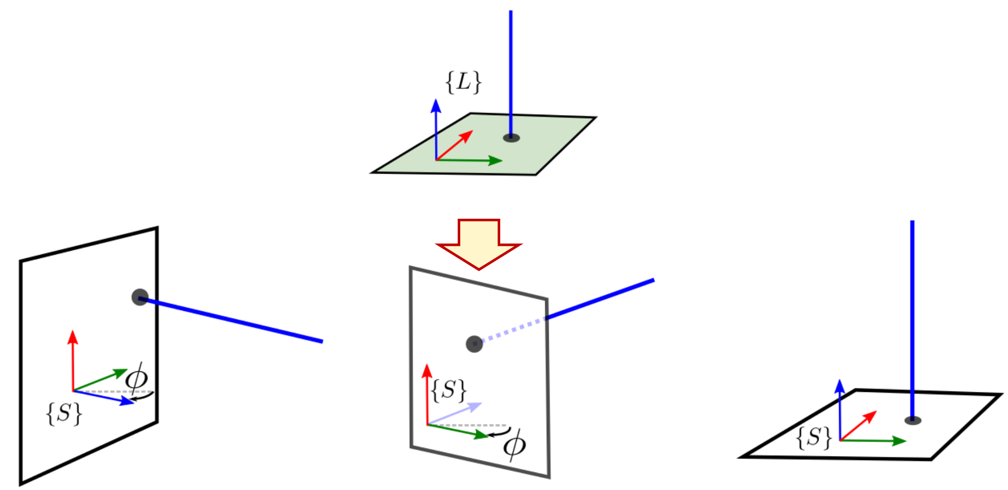
$$\boxed{l_p \sim (\theta, \rho, 0)^{\mathrm{T}}}$$

$$\theta = \operatorname{atan2}(b, a)$$
$$\rho = 1/\sqrt{a^2 + b^2}$$

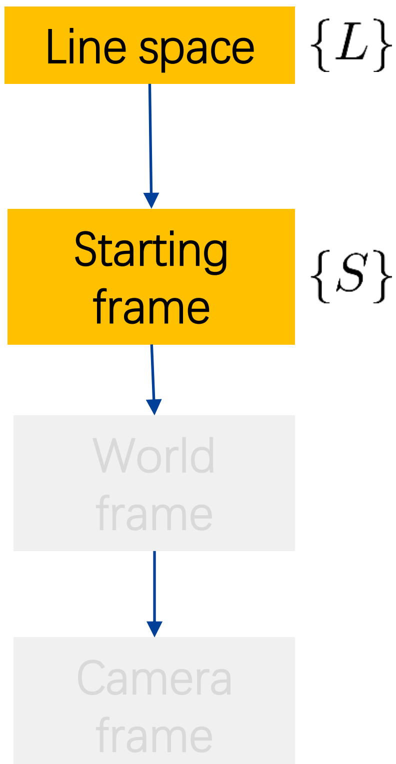


Line space -> Starting frame

- The structural line can be transformed into three axes of the starting frame $\{S\}$ by the rotation ${}_{L}^{S}R$.



$${}_{L}^{S}R = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \qquad {}_{L}^{S}R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \qquad {}_{L}^{S}R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



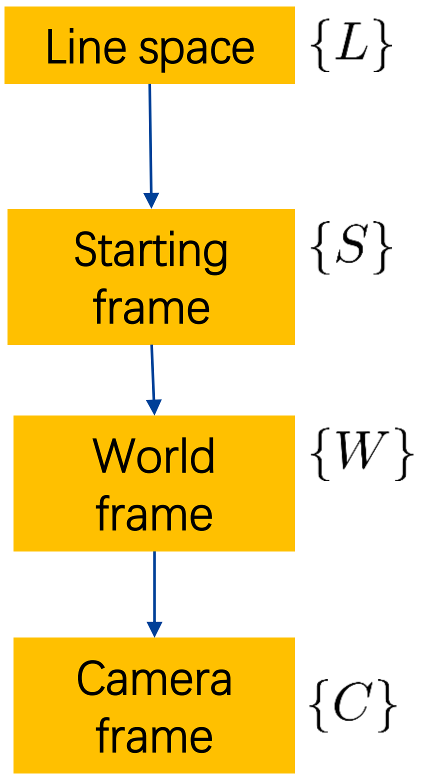
Starting frame -> World frame

- The structural line can be further transformed into the world frame by using the heading direction (ϕ_i) of the local Manhattan world.

$$
{}^{W}_{S}R(\phi_i) = \begin{bmatrix} \cos(\phi_i) & \sin(\phi_i) & 0 \\ -\sin(\phi_i) & \cos(\phi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

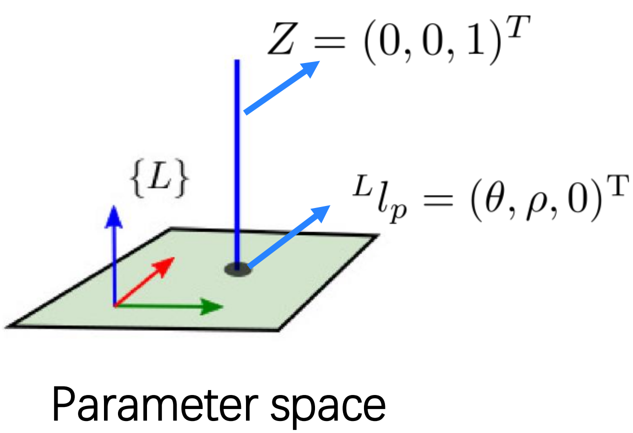
- The structural line is then transformed to the current camera frame by.

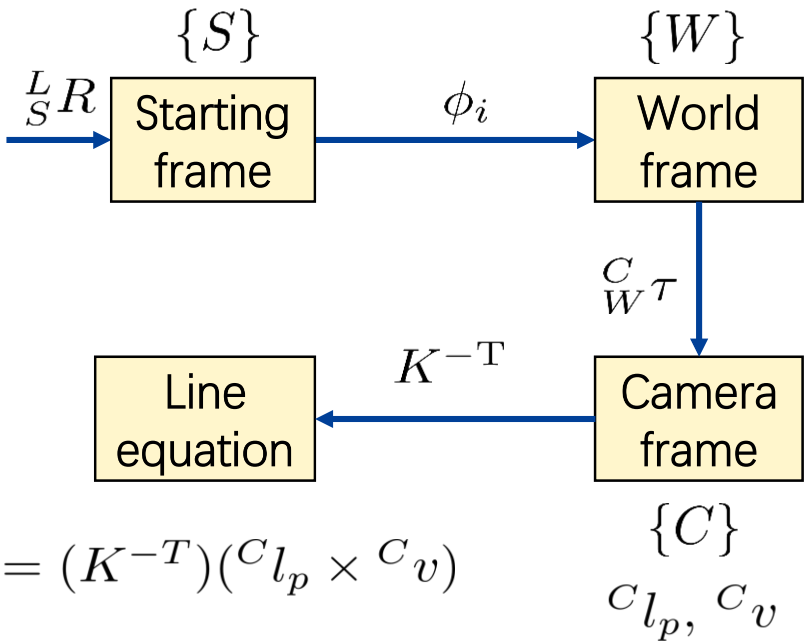
$$
{}^{C}_{W}\tau = ({}^{C}_{W}R, {}^{C}p_W)
$$



Line projection on the image

- Apply the transformations to both the point l_p and the vertical direction Z





$${}^{im}l = (K^{-T})({}^{C}l_p \times {}^{C}v)$$

Line projection on the image

- Line projection can be written as the following functions

$$ {}^{im}l = \Pi(l, \phi_i, {}^S_L R, {}^W_C \tau) $$

$$ {}^{im}l = \Pi(l, \phi_i, {}^S_L R, {}^I_C \tau, {}^W_I \tau) $$ (unknown camera-IMU calibration ${}^I_C\tau$)

, where ${}^S_L R$ are known constants after line direction classification.

- Hence we further write

$$ {}^{im}l = \Pi(l, \phi_i, {}^W_C \tau) $$

$$ {}^{im}l = \Pi(l, \phi_i, {}^I_C \tau, {}^W_C \tau) $$

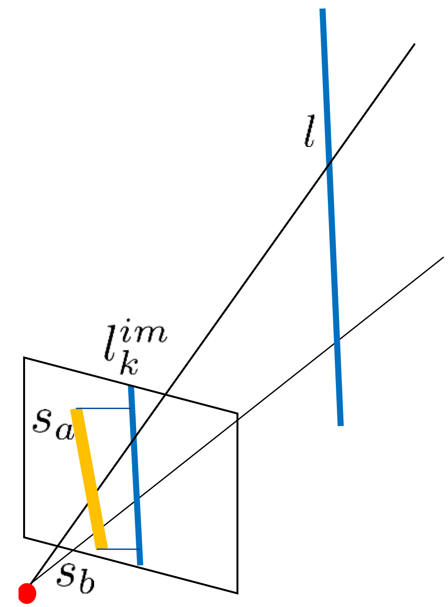
- We can use the above functions to derive the measurement equations.

Measurement equations

- Measurement equation by re-projection errors
  - The line projection at time k is given by:

$$ {}^{im}l_k = \Pi(l, \phi_i, {}^{I}_{C}\tau, {}^{W}_{C}\tau) $$

  - The line segment detected on the image is denoted by : $s_a \leftrightarrow s_b$
  - Hence the re-projection error can be computed as the signed distances between the line projection and the two end points:



$$ r_k = D(s_a, s_b, {}^{im}l) = D(s_a, s_b, \Pi(l, \phi_i, {}^{I}_{C}\tau, {}^{W}_{I_k}\tau)) $$

Measurement equations

- After local linearization, we have

$$r_k = h_0 + J_l \delta l + J_\phi \delta\phi + J_{IC} \delta\, {}^I_C\tau + J_{WI_k} \delta\, {}^W_{I_k}\tau$$

- By stacking all observations from time <u>1</u> to time <u>*M*</u>

$$\begin{aligned}
r_1 &= h_0 + J_l \delta l + J_\phi \delta\phi + J_{IC} \delta\, {}^I_C\tau + J_{WI_1} \delta\, {}^W_{I_1}\tau \\
&\cdots \quad \cdots \\
r_k &= h_0 + J_l \delta l + J_\phi \delta\phi + J_{IC} \delta\, {}^I_C\tau + J_{WI_k} \delta\, {}^W_{I_k}\tau \\
&\cdots \quad \cdots \\
r_M &= h_0 + J_l \delta l + J_\phi \delta\phi + J_{IC} \delta\, {}^I_C\tau + J_{WI_M} \delta\, {}^W_{I_M}\tau
\end{aligned}$$

$$\Downarrow$$

$$z = H_l \underbrace{\boxed{\delta l}}_{\text{Line parameters}} + H_\phi \underbrace{\boxed{\delta\phi}}_{\text{Heading of Manhattan world}} + H_{CI} \underbrace{\boxed{\delta\, {}^I_C\tau}}_{\text{Camera-IMU calibration}} + H_{WI} \underbrace{\boxed{[\delta\, {}^W_{I_1}\tau \cdots \delta\, {}^W_{I_M}\tau]}}_{\text{Historical IMU poses}}$$

Measurement equations

- Project the residual to the left null space of $\boldsymbol{H}_{\boldsymbol{l}}$, we can get rid of the line parameters:

$$z = H_l \delta l + H_\phi \delta \phi + H_{CI} \delta\, {}^I_C\tau + H_{WI} [\delta^W_{I_1}\tau \cdots \delta^W_{I_M}\tau]$$

$$\Downarrow$$

$$z^{(0)} = H^{(0)}_\phi \delta\phi + H^{(0)}_{CI} \delta\, {}^C x_I + H^{(0)}_{WI} [\delta^W_{I_1}\tau \cdots \delta^W_{I_M}\tau],$$

- The measurement equation involves
  - 1. Heading direction of the local Manhattan world
  - 2. IMU-camera relative pose
  - 3. Historical IMU poses

Outside of the filter

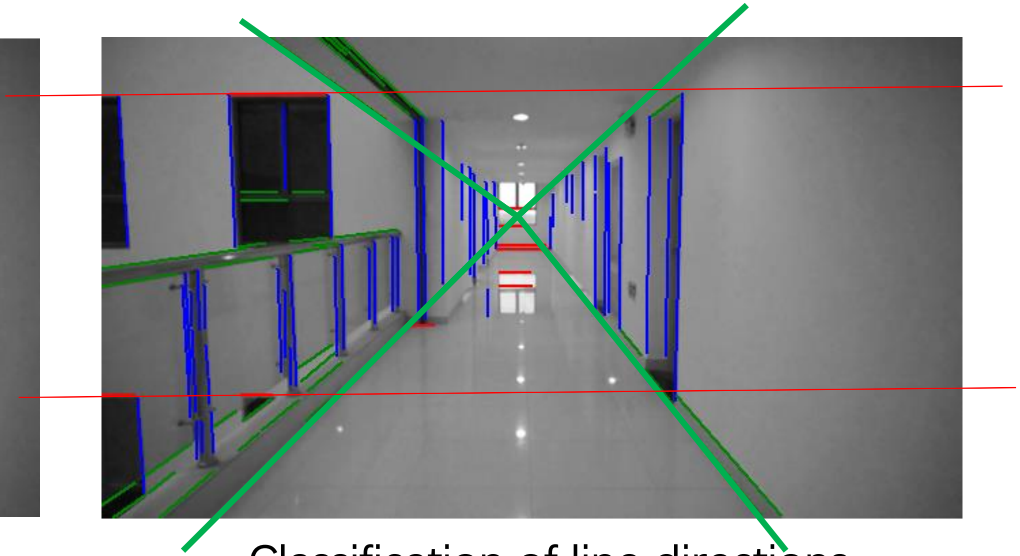
- Structural line related tasks:
    - Line detection & classification of structural lines
    - initialization & triangulation
    - Line tracking
    - Handling long feature tracks

Outside of the filter

- Structural line related tasks:
  - Line detection & classification of structural lines

For a line segment $s_a \leftrightarrow s_b$ find its **Manhattan world** ϕ_i and **its direction** (X,Y,orZ)

Detection of line segments



Classification of line directions (X,Y,or Z) and identify the Manhattan world $\boldsymbol{\phi_i}$

Outside of the filter

- Structural line related tasks:
  - Initialization

    1. Longer line segment first
    2. Establish the starting frame (in which Manhattan world it lies)
    3. Use the middle point $\boldsymbol{m}$ of the line segment for initialization

For horizontal lines (aligned with X, Y axes of a certain Manhattan frame)

Camera frame → World frame → Starting frame (Local Manhattan) → Line parameter space

$$ {}^{L}m = {}^{L}_{S}R \; {}^{S}_{W}R(\phi_i) \; {}^{W}_{C}R \; K^{-1} m $$

Outside of the filter

- Structural line related tasks:
  - Initialization

    1. Longer line segment first
    2. Establish the starting frame (in which Manhattan world it lies)
    3. Use the middle point $\boldsymbol{m}$ of the line segment for initialization

For vertical lines (aligned with Z axis of any Manhattan frames)

Camera frame → World frame → A dummy Manhattan world $\phi_0 = 0$ → Line parameter space

$$
{}^{L}m = {}^{W}_{C}R \, K^{-1} m
$$

Outside of the filter

- We can write the initialization process as a function:

$$l_0 = \begin{bmatrix} \theta_0 \\ \rho_0 \end{bmatrix} = \begin{bmatrix} \Pi^{-1}(s, \phi_i, {}^W_C R) \\ \rho_0 \end{bmatrix}$$

Initial parameters

s : line segment
ϕ_i: Local Manhattan frame
${}^W_C R$: Current camera orientation
ρ_0: Initial inverse depth

$$\Sigma_0 = \begin{bmatrix} \sigma^2_{\theta_0} & 0 \\ 0 & \sigma^2_{\rho_0} \end{bmatrix}$$

Initial covariance

$\sigma^2_{\theta_0}$: small value to account line detection error (2-4 pixels)
$\sigma^2_{\rho_0}$: uncertainty of inverse depth (5 by default)

Outside of the filter

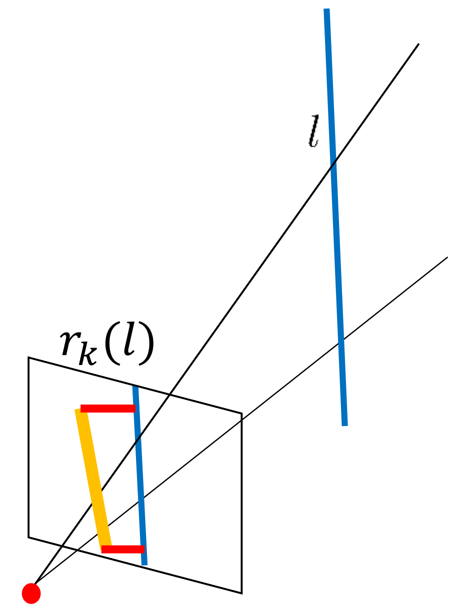
- Line triangulation with prior Knowledge

Line projection error

Prior knowledge

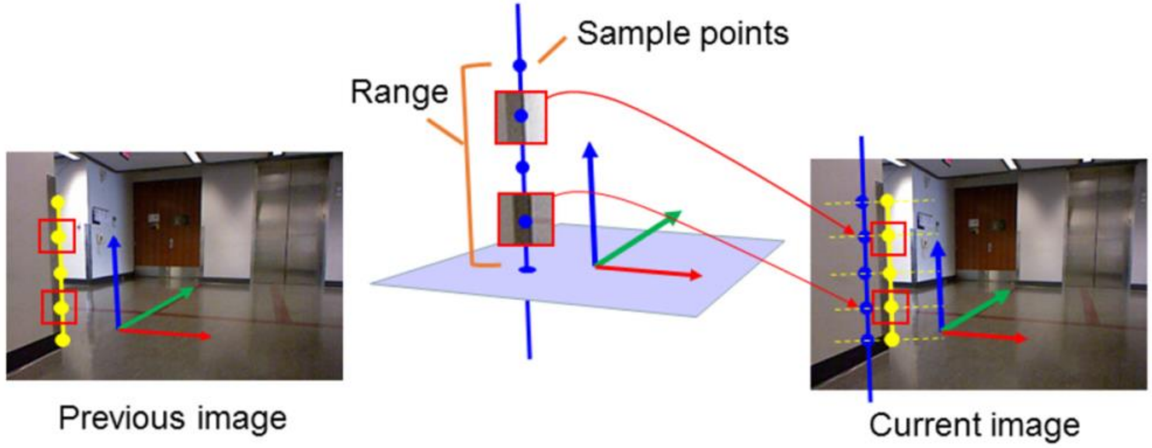
$$\arg\min_{l=(\theta,\rho)^T} \boxed{\sum_{k\in\mathcal{V}} r_k^2(l)/\sigma_{im}^2} + \boxed{(l-l_0)^T\Sigma_0^{-1}(l-l_0)}$$

l_0 : Prior line parameters
$r_k(l)$: line projection error
$\mathcal{V}$: set of visible views



Outside of the filter

- Line tracking
  - 1. Sample **several points** on the line
  - 2. Project those points onto the image, searching corresponding points perpendicular to the line projection.
  - 3. Use the small patches around those points as the descriptor



Outside of the filter

- Handling long feature tracks

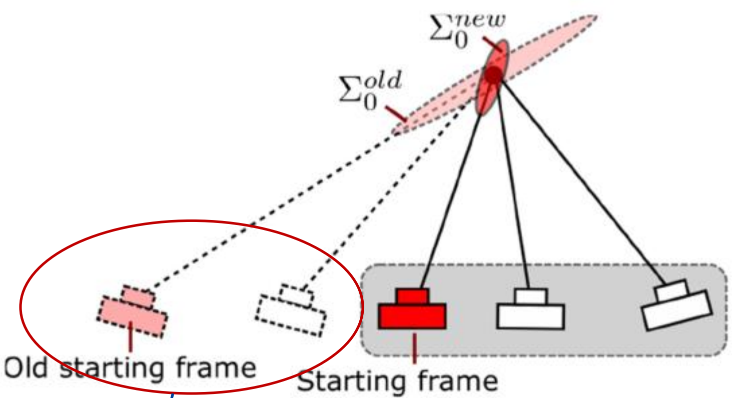
Step1 – Absorb dropped measurements into priori information :

$$\underset{l=(\theta,\rho)^T}{\arg\min} \sum_{k\in\mathcal{D}} r_k^2(l)/\sigma_{im}^2 + (l - l_0^{\mathrm{old}})^T(\Sigma_0^{\mathrm{old}})^{-1}(l - l_0^{\mathrm{old}})$$

$$\Lambda \delta l = Y$$

Normal equation in the last Gauss-Newton iteration

$$\Sigma_0^{\mathrm{new}} \leftarrow \Lambda^{-1}$$



Dropped views $\{\mathcal{D}\}$

Step2 – Change the starting frame $S \rightarrow S'$

$$\boxed{\begin{array}{l} l' \leftarrow {}^{S'}_{S}\mathcal{T}(l), \\ \Sigma' \leftarrow J\Sigma J^T, \end{array}}\boxed{\begin{array}{l} l'_0 \leftarrow {}^{S'}_{S}\mathcal{T}(l_0) \\ \Sigma'_0 \leftarrow J\Sigma_0 J^T \end{array}}$$

Current estimate

Prior information

Outside of the filter

- Manhattan world detection :
  - 1. starts once vertical lines are identified
  - 2. compute the horizontal line $l_\infty = K^{-T}\ {}^{C}_{W}R[0,0,1]^T$
  - 3. run 1-line RANSAC to detect one of the two horizontal directions (X or Y)
    - Randomly select one line, extended it to intersect l_∞ to get a vanishing point v_x
    - Compute the other vanishing point v_y
    - Check the consistent line segments aligned with v_x or v_y
    - Repeat the aforementioned steps
  - It is **a possible Manhattan world** if the maximum consensus set contains sufficient inliers.

Outside of the filter

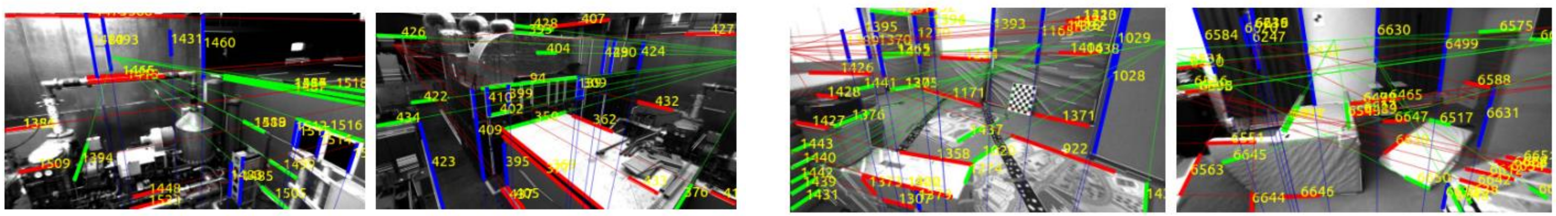
- Manhattan world merging :
  - The heading direction of two Manhattan worlds could be very close.

$$|\phi_i - \phi_j| < \Delta\phi$$

  - We merge them by removing the newly detected one and update the information of related structural lines

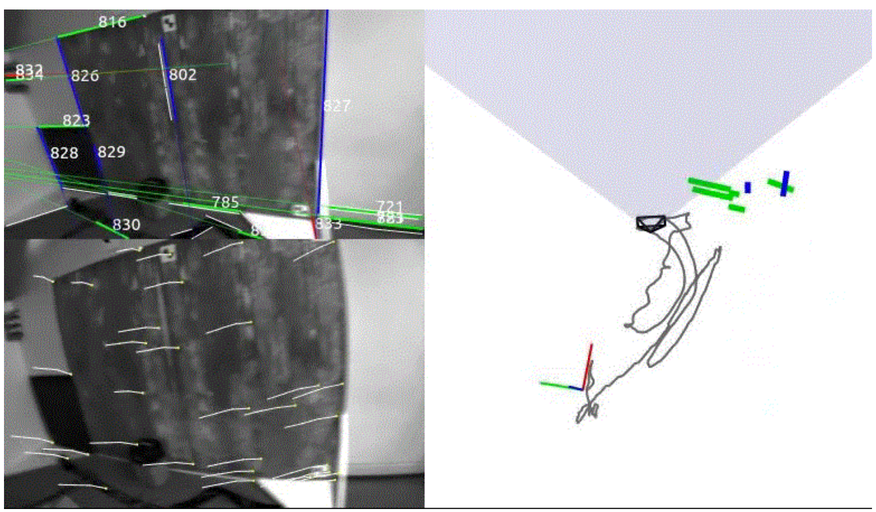
Results

- Benchmark tests on Euroc dataset

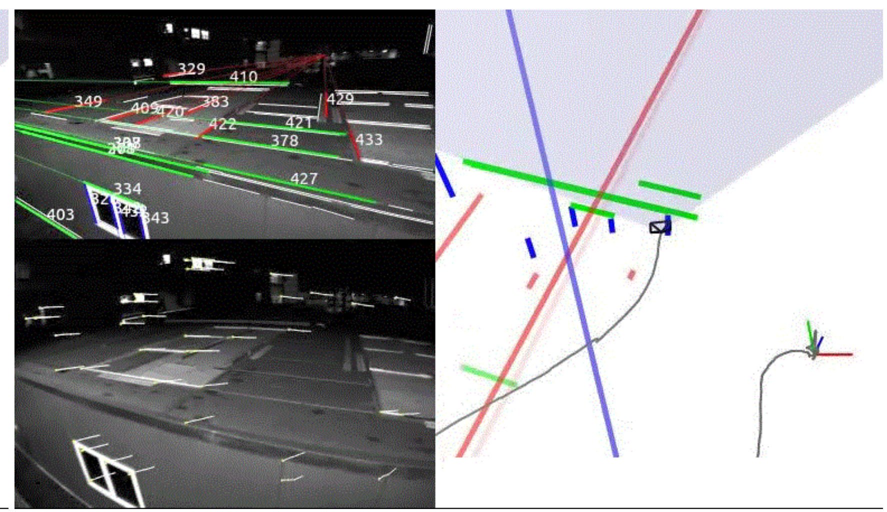


(a) Machine hall

(b) Vicon room



V2_03_difficult



MH_05_difficult

Results

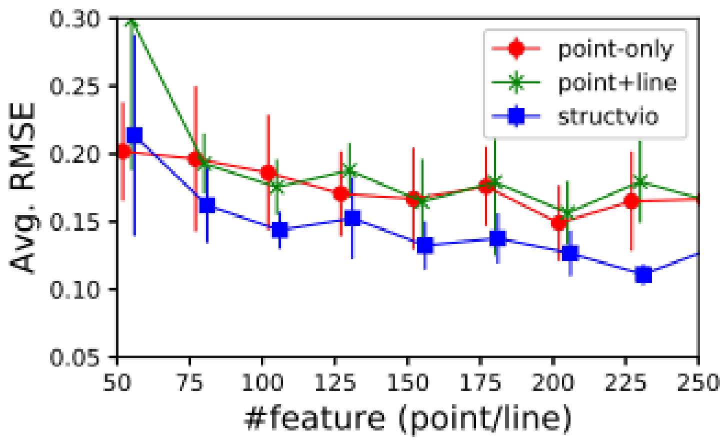
- Euroc dataset

Dataset	OKVIS[5]		VINS[6](w/o loop)		StructVIO	
	RMSE	Max.	RMSE	Max.	RMSE	Max.
MH_01_easy	0.308	0.597	**0.157**[2]	0.349	**0.079**[1]	0.251
MH_02_easy	0.407	0.811	**0.181**[2]	0.533	**0.145**[1]	0.267
MH_03_medium	0.241	0.411	**0.196**[2]	0.450	**0.103**[1]	0.271
MH_04_difficult	0.363	0.641	**0.345**[2]	0.475	**0.130**[1]	0.286
MH_05_difficult	0.439	0.751	**0.303**[2]	0.434	**0.182**[1]	0.358
V1_01_easy	**0.076**[2]	0.224	0.090	0.201	**0.060**[1]	0.180
V1_02_medium	0.141	0.254	**0.098**[1]	0.334	**0.130**[2]	0.260
V1_03_difficult	0.240	0.492	**0.183**[2]	0.376	**0.090**[1]	0.263
V2_01_easy	0.134	0.308	**0.080**[2]	0.232	**0.045**[1]	0.140
V2_02_medium	0.187	0.407	**0.149**[2]	0.379	**0.066**[1]	0.157
V2_03_difficult	**0.255**[2]	0.606	0.268	0.627	**0.110**[1]	0.231

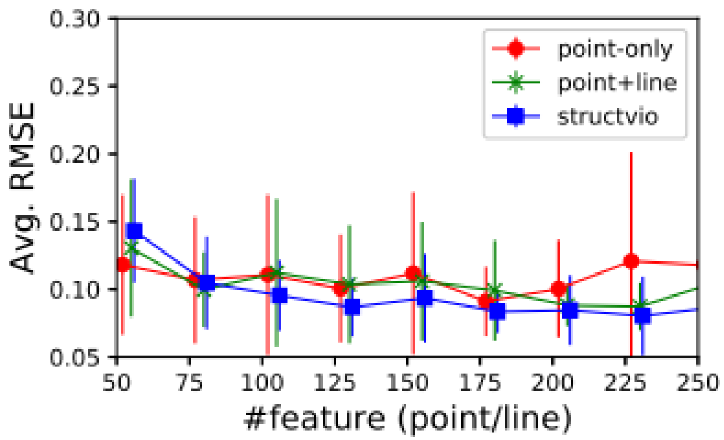
RMSE-Rooted Mean Squared Error

Results

- Euroc datasets
  - StructVIO performs better in Machine hall, since it exhibit stronger structural regularity.



(a) Machine hall

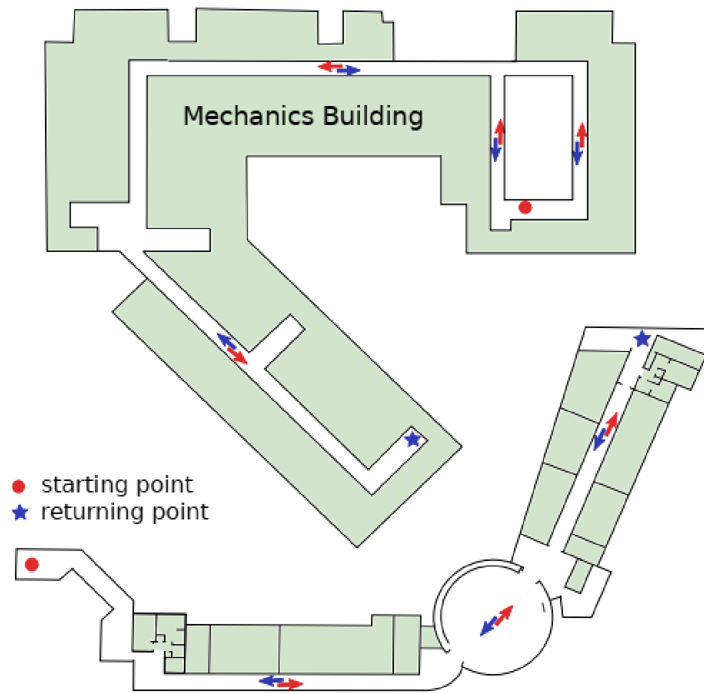


(b) Vicon room

Results

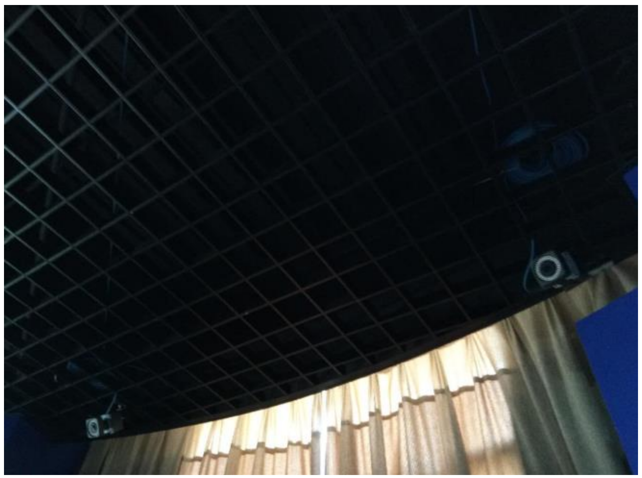
- Visual-inertial data collected by Google Tango Tablet (16 test sequences)
- Different buildings in SJTU campus.
- Indoor/Outdoor, Large illumination changes, 5~10 minutes walking

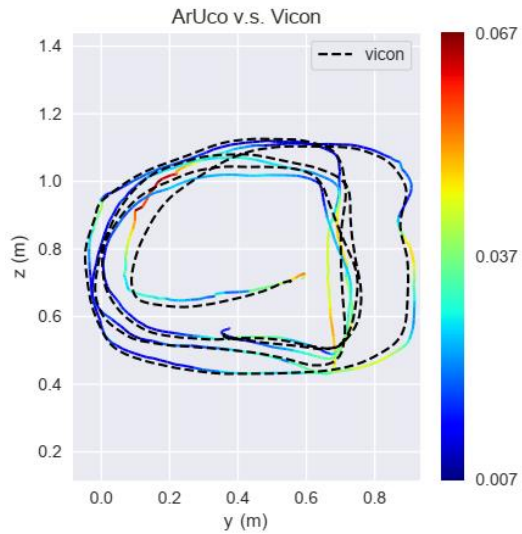
Project Tango



Results

- Ground truth data were collected by either Vicon or ArUco code.





Starting segment: s

Ending segment: e

Align the starting segments : $T^* = \arg\min_{T} \sum_{t \in s} (\|T(p^t) - g_s^t\|^2)$

Compute the ending segment's RMSE and Max errors:

$$RMSE = \sqrt{\frac{1}{|e|} \sum_{t \in e} \|(T(p^t) - g_e^t\|^2)} \qquad Max. = \max |T(p^t) - g_e^t|$$

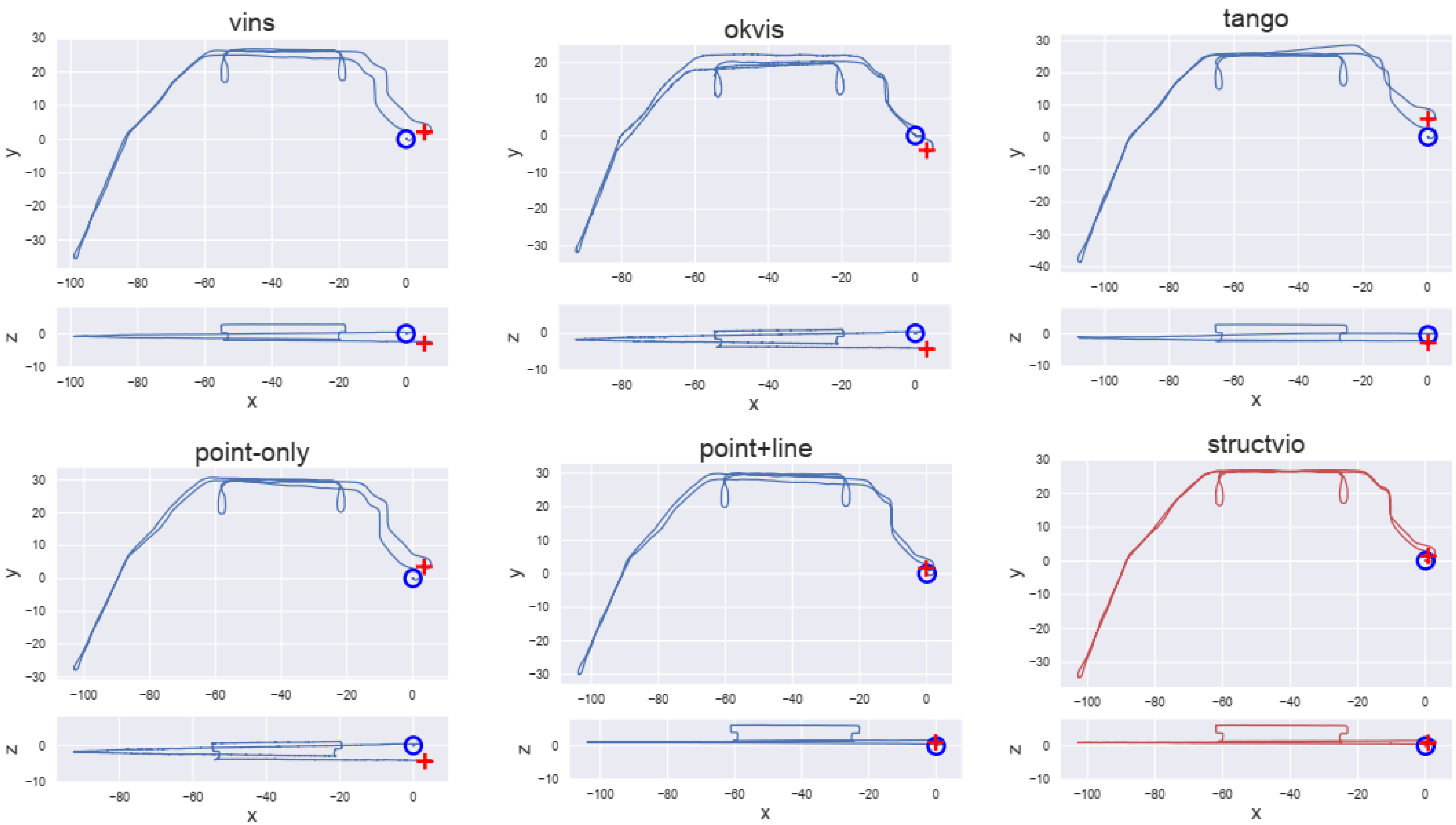
Results

- Methods： OKVIS, VINS, Project Tango, Point-only, Point-line, StructVIO

Seq. Name	Traveling Dist. [m]	OKVIS[5]		VINS[6](w/o loop)		Project Tango		Point-only		Point+Line		StructVIO	
		RMSE	Max.	RMSE	Max.	RMSE	Max.	RMSE	Max.	RMSE	Max.	RMSE	Max
Soft-01	315.167	6.702	9.619	4.861	6.688	5.715	8.181	**2.153**[2]	2.728	2.262	2.842	**1.931**[1]	2.437
Soft-02	438.198	4.623	6.713	2.713	4.086	4.238	6.226	3.905	5.243	**1.468**[2]	2.026	**1.429**[1]	1.984
Soft-03	347.966	**4.505**[2]	6.223	7.270	9.832	167.825	228.630	6.515	8.119	8.618	10.790	**0.325**[1]	1.020
Soft-04	400.356	3.993	5.784	28.667	75.479	2.453	3.544	**1.550**[1]	2.028	4.051	5.262	**1.722**[2]	2.241
Mech-01	340.578	3.627	4.745	2.452	3.260	**1.948**[2]	2.726	3.298	3.961	4.323	5.181	**0.909**[1]	1.165
Mech-02	388.548	3.079	4.195	3.570	4.754	**1.596**[2]	2.217	1.663	2.108	2.317	2.927	**0.779**[1]	1.022
Mech-03	317.974	3.875	5.324	4.682	9.113	4.220	5.781	**2.384**[2]	3.020	4.193	5.272	**1.161**[1]	1.532
Mech-04	650.430	-	-	3.002	8.592	1.915	5.808	1.785	4.663	**1.425**[2]	3.729	**0.742**[1]	1.940
MicroA-01	257.586	2.485	3.382	**0.654**[2]	1.148	45.599	61.058	2.849	3.505	2.189	2.721	**0.642**[1]	1.225
MicroA-02	190.203	3.428	5.186	14.222	57.172	**1.145**[1]	1.692	1.964	2.514	**1.723**[2]	2.207	2.089	2.661
MicroA-03	388.730	0.078	0.779	**1.800**[1]	2.578	4.400	6.253	3.824	5.169	3.072	4.232	**1.884**[2]	2.892
MicroA-04	237.856	6.136	8.532	**0.994**[2]	1.765	55.200	75.318	2.056	2.897	2.406	2.879	**0.350**[1]	0.448
MicroB-01	338.962	2.898	4.025	**1.856**[2]	2.944	38.197	50.572	7.084	8.576	7.337	8.913	**1.477**[1]	1.902
MicroB-02	306.316	2.240	3.490	**1.030**[2]	2.431	5.660	8.652	2.521	3.714	3.197	4.610	**0.470**[1]	0.799
MicroB-03	485.291	-	-	2.132	3.368	**2.009**[2]	2.960	6.490	8.978	4.507	6.301	**0.445**[1]	0.675
MicroB-04	357.251	4.064	6.481	**1.332**[2]	2.068	13.962	22.028	5.078	7.713	1.977	3.074	**0.473**[1]	0.777
Mean Drift Err.(%)		1.078%		1.410%		6.180%		0.957%		**0.956%**[2]		**0.292%**[1]	
Median Drift Err.(%)		0.781%		**0.538%**[2]		0.900%		0.559%		0.570%		**0.176%**[1]	

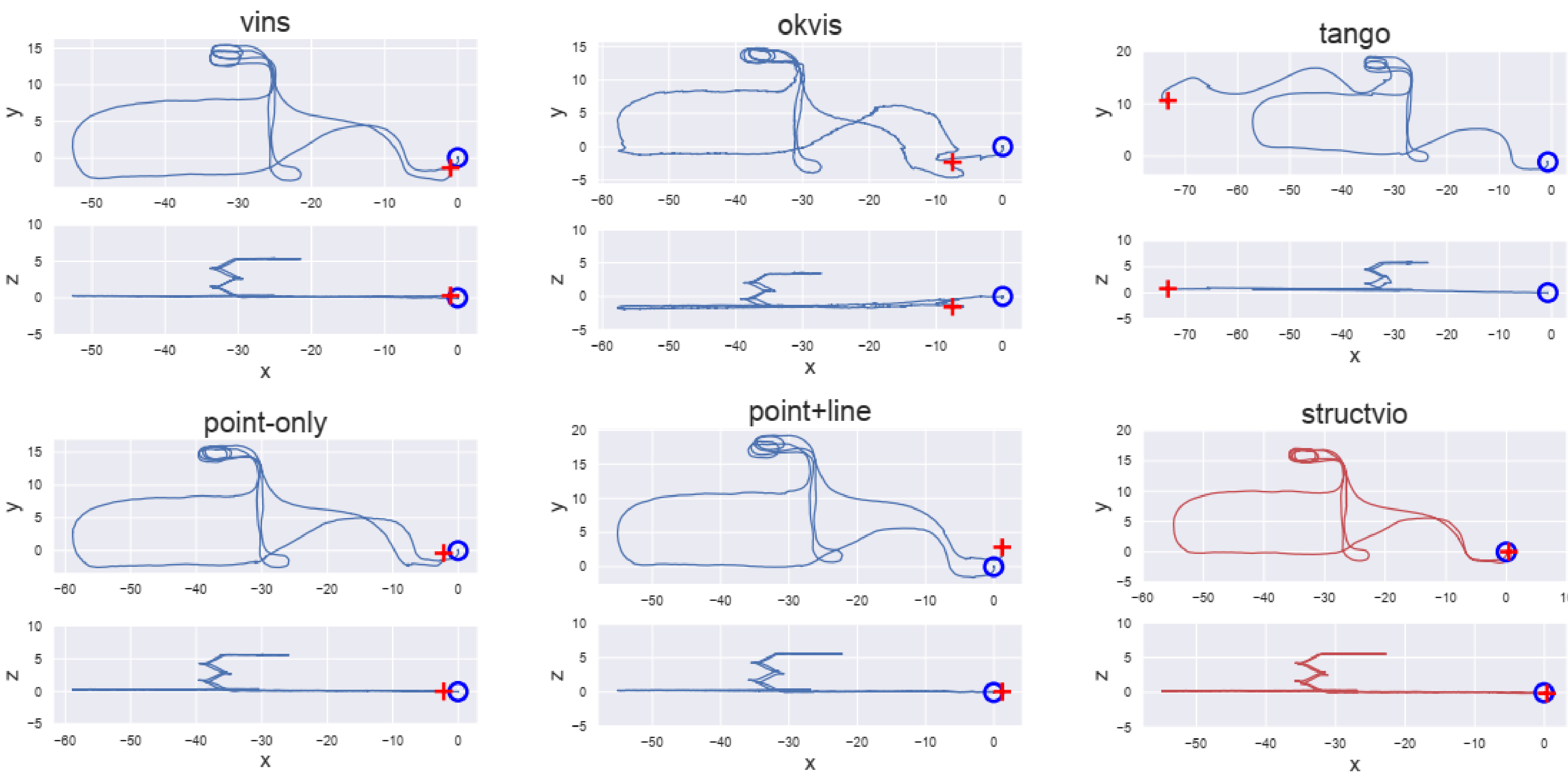
Results

- Software building (Soft-02)



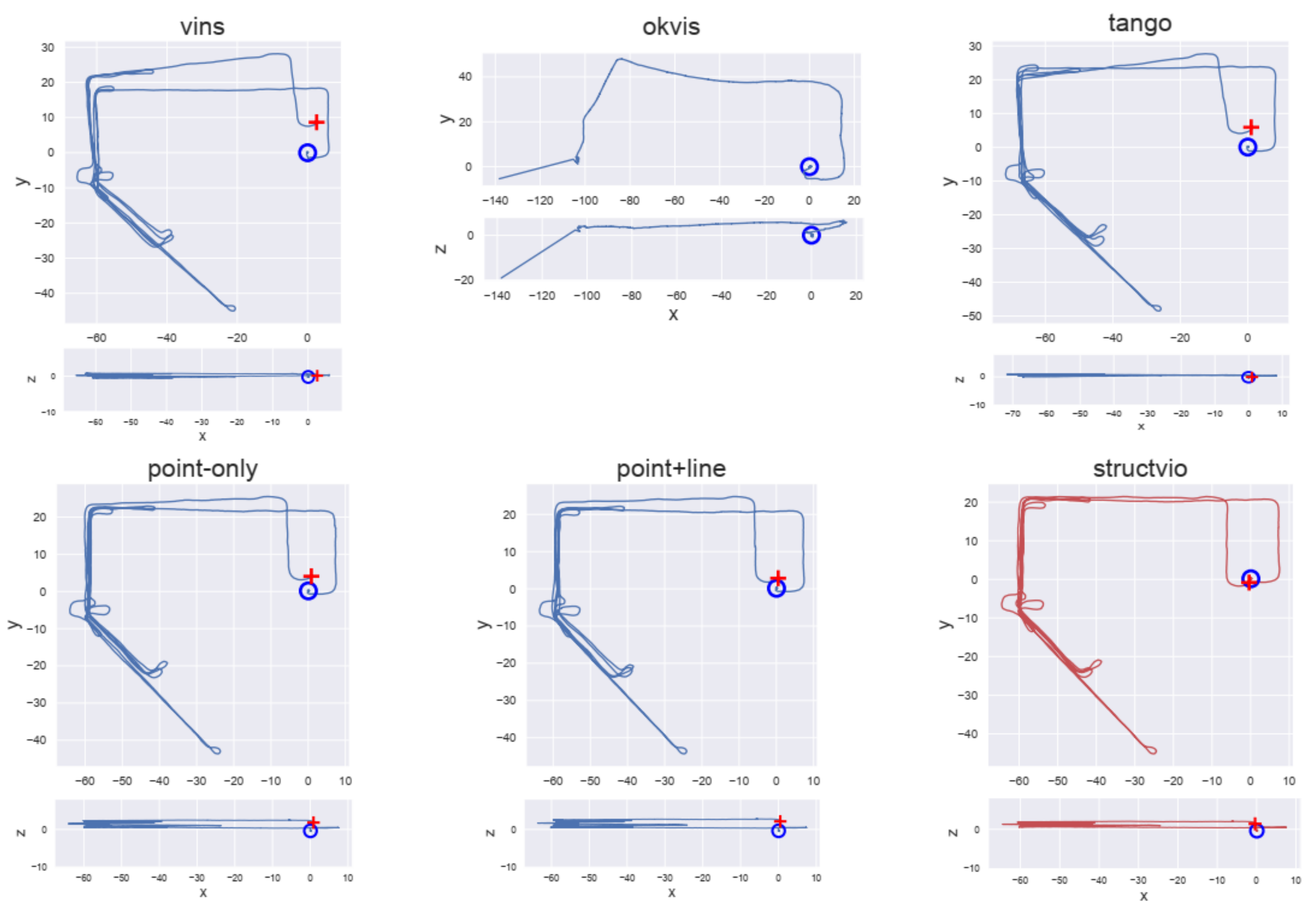
Results

- Micro Electronic Engineering Building (MicroA-04)



Results

- Mechanical Engineering Building (Mech-04)

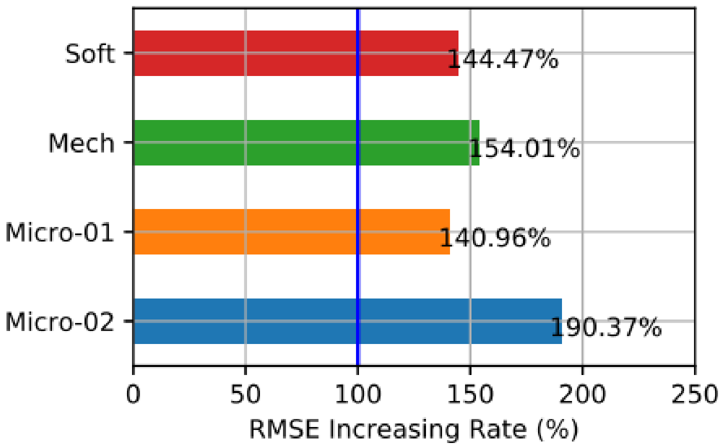


Results

- Other tests

Seq. Name	Atlata world		Manhattan world	
	RMSE	Max.	RMSE	Max.
Mech-01	0.909	1.165	1.144	1.524
Mech-02	0.779	1.022	1.286	1.061
Mech-03	1.161	1.532	2.029	1.211
Mech-04	0.742	1.940	1.822	2.193
Soft-01	1.931	2.437	2.896	2.397
Soft-02	1.429	1.984	3.092	4.149
Soft-03	0.325	1.020	3.352	4.236
Soft-04	1.722	2.241	3.178	4.120

Atlanta world vs Manhattan world



Without dealing with dropped measurements for long feature tracks

- Software, tools, and dataset:

http://drone.sjtu.edu.cn/dpzou/project/structvio.html

Software usage

- Supports the following modes
  - Point/Line/Structural line-only
  - Point+line
  - Point+Structural line

```
-l <0|1|2>,  --line_type <0|1|2>
  Type of lines used: 0-structlines, 1-general lines, 2-both

-f <0|1|2>,  --feature_type <0|1|2>
  Type of features used: 0 - point, 1 - line, 2 - both
```

```
./structvio -i ./Soft-01 -n Soft-01 -r Soft-01-res -c structvio_data.yaml
```

- Script for evaluation (revised from evo)

```
vio_eva.py -r <path to the result file - 'state.txt'> -d <path to the folder of data - e.g. 'Mech-01'>
```


謝
謝