IMPROVED FOURIER MELLIN INVARIANT FOR ROBUST ROTATION ESTIMATION WITH OMNI-CAMERAS

Qingwen Xu[1] Arturo Gomez Chavez[2] Heiko Bülow[2] Andreas Birk[2] Sören Schwertfeger[1]

[1]School of Information Science Technology of ShanghaiTech University

[2]Robotics Group, Computer Science & Electrical Engineering, Jacobs University Bremen, Germany

MOTIVATION

- Omni-directional cameras have large field of view (FOV) with low resolution
- iFMI method achieves better performance than feature-based method on difficult environments (e.g. foggy, featureless)
- iFMI method is only applicable on 2D motion and pin-hole images

CONTRIBUTIONS

- Using the iFMI method to estimate motion between omni-images; hence
- Allowing 3D registration instead of 2D for iFMI;
- Proposing a motion model based on sub-image patches to compensate for omni-images non-linear distortions;
- Providing baseline comparisons against commonly used registration feature-based methods.

MOTION MODEL

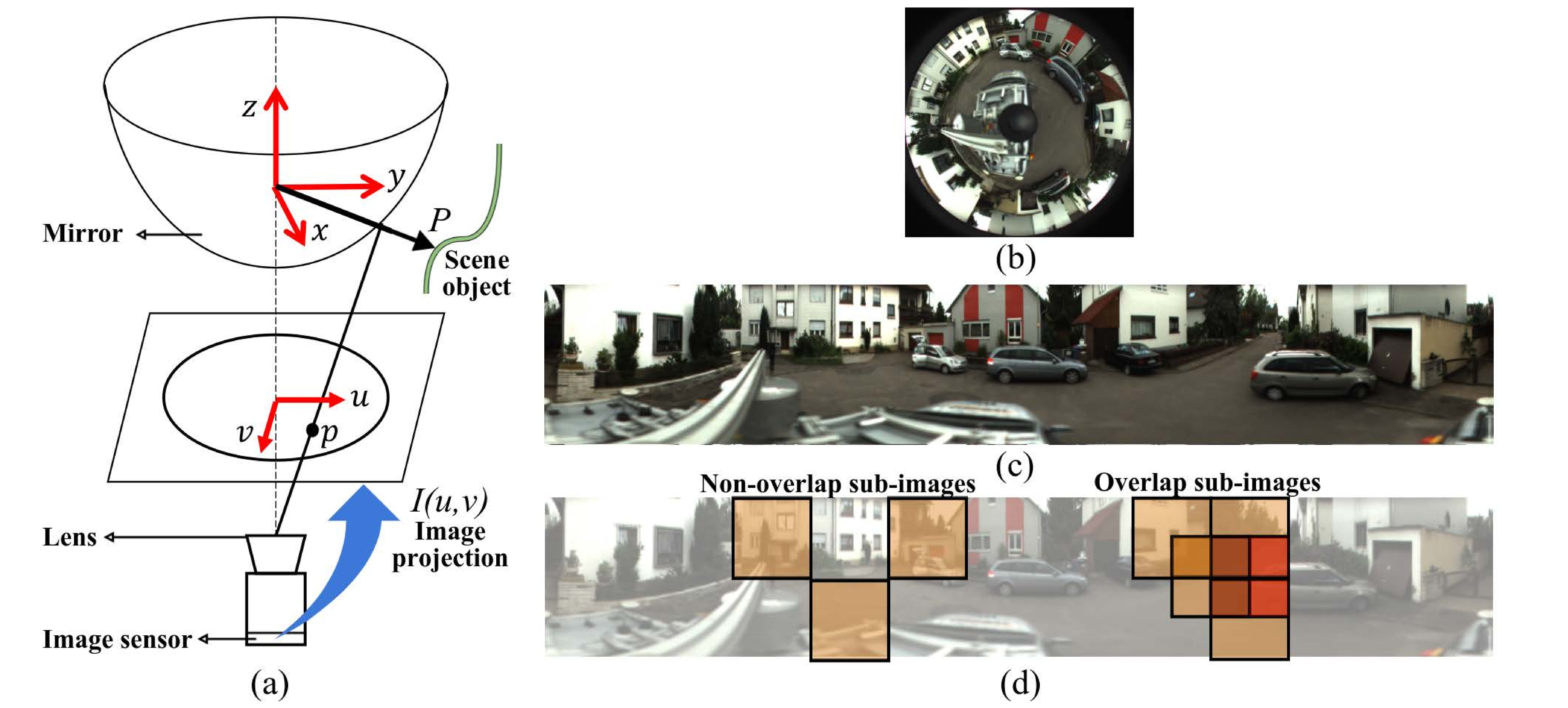

Fig.1 (a) Catadioptric Omni-directional Camera Model [1]. (b) Omnidirectional image (c) Panorama image obtained from omni-image. (d) Non-overlapping and overlapping sub-image extraction.

iFMI Method

- Rotation is the same in time and frequency domain
- Scaling between two images in frequency domain can be described as

$$|G(a_i^1)| = \sigma^{-2}|G(\sigma^{-1}a_i^2)| \quad (1)$$

- Eq. (1) can be expressed as Eq. (2) with iFMI descriptor

$$V_{G(a_i^1)} = \sigma^{-2}e^{-j2\pi r(\theta,s)}V_{G(a_i^2)} \quad (2)$$

IMPLEMENTATION

Algorithm: Proposed iFMI-based rotation estimation

1: Input: Omni images I_o^1, I_o^2;
Noise filter thresholds th_{pr}, th_{pnr}

2: Obtain panorama images I_p^1, I_p^2 of size $W \times H$ by cartesian-to-polar transformation

3: Extract sub-image set $\mathbb{A}^1, \mathbb{A}^2$ from I_p^1 and I_p^2

4: **for** all sub-images $a_i^1 \in \mathbb{A}^1$, $a_i^2 \in \mathbb{A}^2$ **do**

5: Compute relative motion
m_i = iFMI$(a_i^1, a_i^2, th_{pr}, th_{pnr}) = [s, \theta, t_x, t_y]^T$

6: Select pixel $p_{a_i}^1 = (c_x + \delta, c_y + \delta)$, where $\delta > 0$

7: Find motion pixel pair $F_i = (p_{a_i}^1, p_{a_i}^2)$

8: Convert F_i to omni-image coordinates
polar-to-Cartesian(F_i)

9: Find camera ray pair $(P_i^1, P_i^2) = \pi^{-1}(F_i)$

10: Add (P_i^1, P_i^2) to correspondences set $\mathbb{S}$

11: **end for**

12: Transformation T_2^1= STEWENIUS-5-Points($\mathbb{S}$)

13: Output: T_2^1

RESULTS

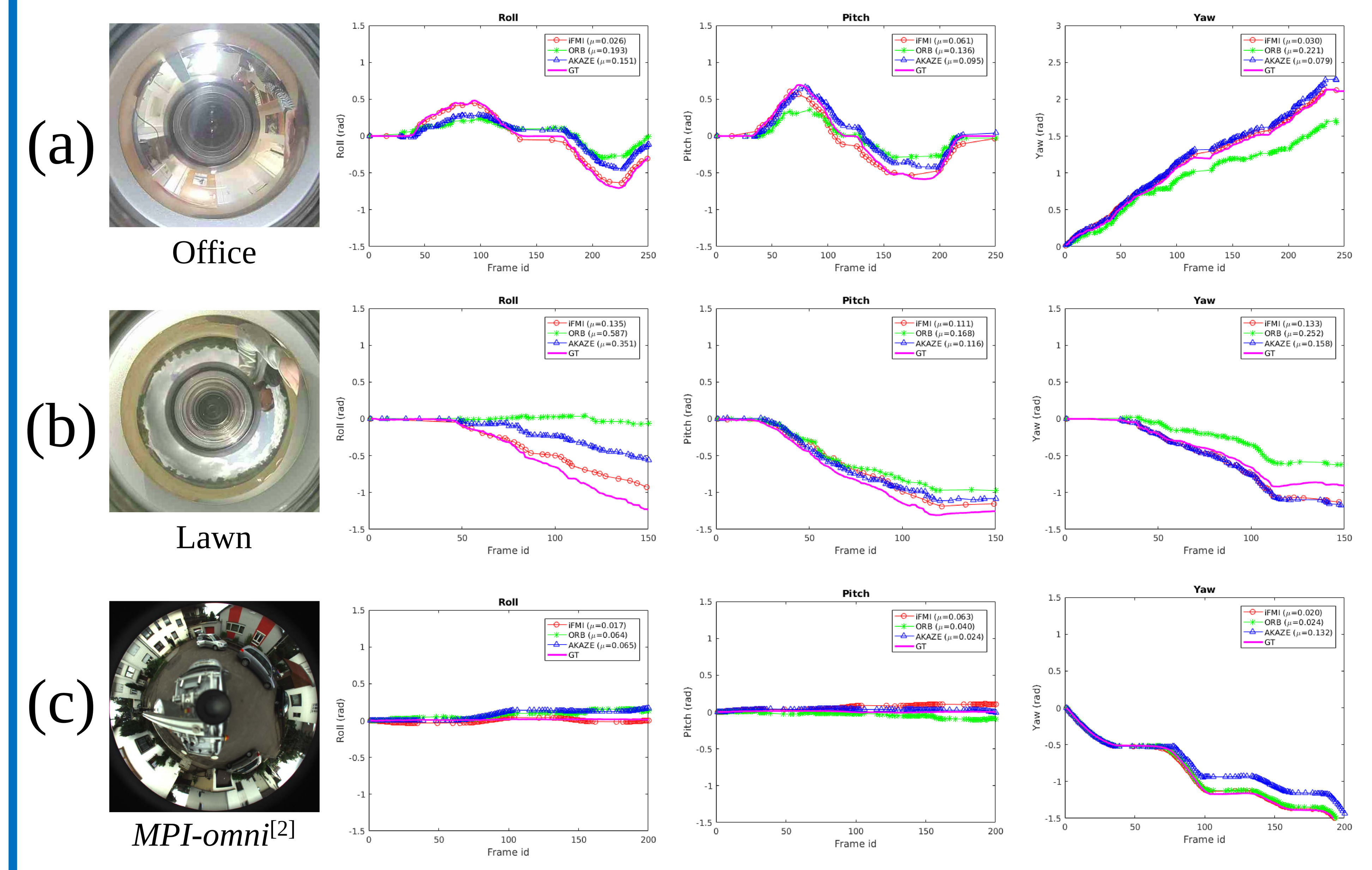

Fig.2 Orientation estimation on different datasets and absolute trajectory root mean square error μ

	iFMI(Ours)	ORB	AKAZE
Roll ε[rad]	**0.058** ± 0.056	0.166 ± 0.078	0.128 ± 0.044
Pitch ε[rad]	**0.107** ± 0.053	0.300 ± 0.337	0.183 ± 0.204
Yaw ε[rad]	**0.075** ± 0.080	0.191 ± 0.139	0.153 ± 0.058
$\mu(\epsilon)$ [rad]	**0.080** ± 0.025	0.219 ± 0.071	0.155 ± 0.028
$\mu(t)$ [s]	0.12	**0.11**	0.75

Table.1 Average orientation root mean square error (RMSE) and computation time per frame from all datasets samples

[1] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in Computer Vision Systems, 2006 ICVS'06. IEEE International Conference on. IEEE, 2006, pp. 45–45.

[2] Miriam Schnbein and Andreas Geiger, "Omnidirectional 3d reconstruction in augmented manhattan worlds," in International Conference on Intelligent Robots and Systems (IROS), 2014