Postprint of Improved Optical Flow Matching Algorithm Integrating IMU for Motion Blur Removal

Authors: Luan Xiaozhen, Wei Guoliang, Cai Jie

Date: 2022-05-10T11:22:58Z

Abstract

To further improve the accuracy and speed of optical flow matching in visual SLAM, this paper proposes an improved optical flow matching algorithm that integrates Inertial Measurement Unit (IMU) information to remove motion blur. The algorithm first employs a point spread function computed from IMU motion data to eliminate motion blur, thereby enhancing the feature point matching rate. Subsequently, building upon the LK (Lucas-Kanade) optical flow framework, it incorporates gradient error and utilizes the L1 norm of image gradients as a regularization term to model sparse noise, constructing a cost function. Furthermore, IMU-predicted feature point positions are utilized as initial values for the algorithm, and the BB (Barzilai-Borwein) step size is introduced to enhance the conventional Gauss-Newton algorithm, thereby improving computational efficiency. Experimental results demonstrate that, through inter-frame comparisons, the proposed algorithm achieves superior efficiency and accuracy compared to the LK optical flow method. Moreover, integration of the proposed algorithm into the VINS-Mono framework and evaluation on the EuRoC dataset reveals that it enhances the localization accuracy and robustness of the original framework.

Full Text

Preamble

Feature Tracking Algorithm for Removing Motion Blur by Improving Optical Flow Incorporating IMU

Luan Xiaozhen[1], Wei Guoliang[2]†, Cai Jie[1]
([1]College of Science; [2]Business School, University of Shanghai for Science & Technology, Shanghai 200093, China)

Abstract: To further improve the accuracy and efficiency of feature point matching in visual SLAM, this paper proposes a novel feature point matching algorithm that fuses inertial measurement unit (IMU) data to remove motion blur. The algorithm first utilizes IMU motion information to compute a point spread function for motion blur removal, thereby improving feature point matching rates. Building upon the Lucas-Kanade (LK) optical flow method, we introduce gradient error and employ the L1 norm of image gradients as a regularization term to model sparse noise, constructing a robust cost function. IMU-predicted feature point positions serve as initial values for the algorithm, and the original Gauss-Newton method is enhanced with BB (Barzilar-Borwein) step sizes to improve computational speed. Experimental results demonstrate that the proposed algorithm outperforms the LK optical flow method in both efficiency and accuracy when comparing consecutive frames. Integration into the VINS-Mono framework and evaluation on the EuRoC dataset shows that the algorithm improves the localization accuracy and robustness of the original framework.

Keywords: optical flow; visual odometry; motion blur; multi-sensor fusion

0 Introduction

In recent years, Visual Simultaneous Localization and Mapping (V-SLAM) has played a crucial role in robotics applications such as logistics distribution, Virtual Reality (VR), and smart home systems. Although monocular cameras are low-cost and convenient, they cannot estimate environmental scale reliably. Consequently, researchers increasingly favor multi-sensor fusion techniques to address this scale unobservability issue in V-SLAM.

Inertial Measurement Units (IMUs) measure angular velocity and acceleration independently of external environmental conditions, making the metric scale observable for monocular vision while providing robust and accurate inter-frame motion estimation. This has led to the rapid development of Visual-Inertial SLAM (VI-SLAM) technologies. VI-SLAM frameworks primarily consist of front-end odometry, back-end optimization, loop closure detection, and mapping. As the initial step, front-end odometry is particularly critical, requiring both IMU pre-integration and feature point extraction and matching from camera images to compute robot motion between consecutive frames and construct local 3D positions of feature points. Mismatches can lead to inaccurate estimations, making feature point extraction and matching a vital component of front-end odometry.

Feature point extraction and matching methods fall into three categories: feature-based methods, optical flow methods, and direct methods. Feature-based methods involve substantial computation for keypoint detection, descriptor calculation, and matching. Optical flow and direct methods address these limitations; optical flow methods compute only keypoints without descriptors, directly minimizing photometric error through iterative matching based on the brightness constancy assumption. This paper focuses on optical flow methods.

Two primary challenges exist in optical flow methods. First, when objects move rapidly, relative motion between the scene and camera during exposure causes pixel overlap and image blur, known as motion blur. Optical flow methods strongly depend on photometric stability and image continuity; when these assumptions are violated, tracking points may be lost or, worse, incorrect features may be tracked. Second, in the non-linear optimization process for minimizing photometric error, the optimal solution is highly dependent on initial values. Without proper initialization, global search becomes necessary, increasing computational cost and mismatch risk. Additionally, traditional Gauss-Newton methods use fixed step sizes or line search algorithms. Overly large step sizes converge quickly initially but may overshoot the optimal solution in final stages, while overly small step sizes slow convergence. This paper addresses both motion blur removal and the selection of initial values and step sizes for non-linear optimization.

To tackle these issues, we propose an improved optical flow matching algorithm that fuses IMU data to remove motion blur. The algorithm leverages IMU motion information to compute a point spread function for blur removal and predicts feature point positions to provide initial values for optical flow tracking, thereby enhancing feature extraction and matching performance. We integrate this algorithm into a complete VI-SLAM framework.

Regarding feature extraction and tracking in VI-SLAM frameworks, VINS-Mono by Tong Qin et al. uses optical flow for keypoint extraction and multi-level pyramid LK tracking. Although the framework fuses IMU and visual data, it does not incorporate IMU information into keypoint extraction and tracking. Qi Guan et al. proposed the IFPT feature point matching algorithm that fuses feature-based methods with IMU by predicting next-frame keypoint positions from IMU data and performing local descriptor search. OKVIS (Open Keyframe-based Visual-Inertial SLAM) uses IMU-predicted poses to filter visible feature points in the current frame, removing untrackable points to save computational time. Both IFPT and OKVIS fuse feature-based methods with IMU, using pre-integration theory to predict possible positions of features in new images. The difference lies in that IFPT performs local search after obtaining predicted positions, while OKVIS filters out invisible points before global brute-force matching.

Our proposed algorithm differs from prior work in several aspects. Algorithmically: (a) Previous work directly set translation information, whereas our algorithm fully utilizes IMU data by incorporating accelerometer measurements, making IMU-predicted feature positions more accurate; (b) Prior work did not fully exploit velocity information, while our algorithm uses IMU pre-integration to compute point spread functions, adding motion blur restoration to reduce matching failures; (c) We incorporate BB variable step sizes in the non-linear optimization for adaptive step size adjustment. Framework-wise: Previous work treated bias as part of the state vector for optimization and predicted camera poses even with inaccurate bias before initialization, causing error propagation that is difficult to eliminate without loop closure. Our algorithm performs IMU prediction after system initialization when bias has been estimated, reducing accumulated error.

---

1 System Framework

This algorithm builds upon the front-end of the VINS-Mono framework. The original VINS-Mono front-end uses Shi-Tomasi corner detection and LK optical flow tracking, which suffers from significant matching errors under large motion and illumination changes.

To address these issues and improve localization accuracy, we enhance the VINS-Mono front-end as follows: Before initialization, traditional optical flow extracts feature points. After initialization, IMU accelerometer and gyroscope biases obtained from pre-integration are used to fuse predicted velocity, position, and rotation information sent to the front-end. If velocity exceeds a threshold, indicating rapid motion, motion information is used for non-blind image restoration. If restoration is completed or velocity is below the threshold (indicating small motion), image restoration is skipped for real-time performance. IMU-predicted position and rotation are then fused with visual information by projecting restored points onto the next frame, providing initial values for the Gauss-Newton algorithm in optical flow tracking to determine local search positions. Figure 1 illustrates the integrated algorithm flow.

---

2 Pre-Integration Model for Predicting Feature Point Positions

This section defines commonly used notation. Let $\mathbf{z}$ denote sensor measurements, $\mathbf{R}$ rotation matrices, $\mathbf{q}$ quaternion representations of rotation, and $\mathbf{T}$ pose transformation matrices comprising rotation and translation. $\mathcal{I}$ represents the IMU coordinate system, $\mathcal{C}$ the camera coordinate system. The first camera frame serves as the reference coordinate system with rotation as identity matrix $\mathbf{I}$ and zero translation vector. After initialization, the extrinsic rotation $\mathbf{R}_{\mathcal{C}}^{\mathcal{I}}$ and translation $\mathbf{t}_{\mathcal{C}}^{\mathcal{I}}$ from camera to IMU coordinates, gyroscope bias $\mathbf{b}_g$, accelerometer bias $\mathbf{b}_a$, and gravity vector $\mathbf{g}_0$ in the reference frame are known.

2.1 Solving Robot Position, Velocity, and Orientation via Pre-Integration

Considering only white noise and random walk noise for gyroscope and accelerometer biases, the IMU model is:

$$
\begin{aligned}
\hat{\omega}_t &= \omega_t + \mathbf{b}_g + \mathbf{n}_g \\
\hat{\mathbf{a}}_t &= \mathbf{R}_{\mathcal{W}}^{\mathcal{I}_t}(\mathbf{a}_t - \mathbf{g}_0) + \mathbf{b}_a + \mathbf{n}_a \\
\dot{\mathbf{b}}_g &= \mathbf{n}_{bg} \\
\dot{\mathbf{b}}_a &= \mathbf{n}_{ba}
\end{aligned}
$$

where ω_t is true angular velocity, $\hat{\omega}_t$ measured angular velocity, $\mathbf{a}_t$ true acceleration, $\hat{\mathbf{a}}_t$ measured acceleration, and $\mathbf{n}_{bg}, \mathbf{n}_{ba}$ are random walk noises following Gaussian distributions.

In the camera coordinate system, taking time derivatives of robot position, velocity, and orientation at time t yields:

$$
\begin{aligned}
\dot{\mathbf{p}}_t^{\mathcal{C}_0} &= \mathbf{v}_t^{\mathcal{C}_0} \\
\dot{\mathbf{v}}_t^{\mathcal{C}_0} &= \mathbf{R}_t^{\mathcal{C}_0}\hat{\mathbf{a}}_t - \mathbf{g}_0 \\
\dot{\mathbf{q}}_t^{\mathcal{C}_0} &= \frac{1}{2}\mathbf{q}_t^{\mathcal{C}_0} \otimes \begin{bmatrix} 0 \\ \hat{\omega}_t \end{bmatrix}
\end{aligned}
$$

where $\mathbf{p}_t^{\mathcal{C}_0}$ is translation from first camera frame to IMU frame at time t, $\mathbf{v}_t^{\mathcal{C}_0}$ velocity in camera coordinates, and $\mathbf{q}_t^{\mathcal{C}_0}$ rotation from IMU frame to first camera frame as quaternion.

Integrating these equations gives position, velocity, and orientation at time t:

$$
\begin{aligned}
\mathbf{p}_{t+\Delta t}^{\mathcal{C}_0} &= \mathbf{p}_t^{\mathcal{C}_0} + \mathbf{v}_t^{\mathcal{C}_0}\Delta t + \frac{1}{2}\mathbf{R}_t^{\mathcal{C}_0}\hat{\mathbf{a}}_t\Delta t^2 - \frac{1}{2}\mathbf{g}_0\Delta t^2 \\
\mathbf{v}_{t+\Delta t}^{\mathcal{C}_0} &= \mathbf{v}_t^{\mathcal{C}_0} + \mathbf{R}_t^{\mathcal{C}_0}\hat{\mathbf{a}}_t\Delta t - \mathbf{g}_0\Delta t \\
\mathbf{q}_{t+\Delta t}^{\mathcal{C}_0} &= \mathbf{q}_t^{\mathcal{C}_0} \otimes \begin{bmatrix} 0 \\ \hat{\omega}_t\Delta t \end{bmatrix}
\end{aligned}
$$

Using midpoint integration for acceleration $\hat{\mathbf{a}}_t$ and angular velocity $\hat{\omega}_t$:

$$
\begin{aligned}
\hat{\mathbf{a}}_t &= \frac{1}{2}(\hat{\mathbf{a}}_t + \hat{\mathbf{a}}_{t+\Delta t}) - \mathbf{R}_{\mathcal{C}_t}^{\mathcal{I}_t}\mathbf{g}_0 \\
\hat{\omega}_t &= \frac{1}{2}(\hat{\omega}_t + \hat{\omega}_{t+\Delta t})
\end{aligned}
$$

During pose optimization, velocities and rotation matrices change, requiring re-integration and significant computation. Pre-integration is therefore introduced:

$$\mathbf{R}_{\mathcal{C}_i}^{\mathcal{C}_j} = \mathbf{R}_{\mathcal{C}_0}^{\mathcal{C}_i}\mathbf{R}_{\mathcal{C}_0}^{\mathcal{C}_j}$$
$$\mathbf{p}_{\mathcal{C}_i}^{\mathcal{C}_j} = \mathbf{R}_{\mathcal{C}_0}^{\mathcal{C}_i}(\mathbf{p}_{\mathcal{C}_0}^{\mathcal{C}_j} - \mathbf{p}_{\mathcal{C}_0}^{\mathcal{C}_i})$$
$$\mathbf{v}_{\mathcal{C}_i}^{\mathcal{C}_j} = \mathbf{R}_{\mathcal{C}_0}^{\mathcal{C}_i}(\mathbf{v}_{\mathcal{C}_0}^{\mathcal{C}_j} - \mathbf{v}_{\mathcal{C}_0}^{\mathcal{C}_i} - \mathbf{g}_0\Delta t_{ij})$$

where $\otimes$ denotes quaternion multiplication. Substituting into the acceleration update yields the discrete kinematic model with pre-integration:

$$\mathbf{p}_{j+1} = \mathbf{p}_j + \mathbf{v}_j\Delta t + \frac{1}{2}\mathbf{R}_j\hat{\mathbf{a}}_j\Delta t^2 - \frac{1}{2}\mathbf{g}_0\Delta t^2$$
$$\mathbf{v}_{j+1} = \mathbf{v}_j + \mathbf{R}_j\hat{\mathbf{a}}_j\Delta t - \mathbf{g}_0\Delta t$$
$$\mathbf{q}_{j+1} = \mathbf{q}_j \otimes \begin{bmatrix} 0 \\ \hat{\omega}_j\Delta t \end{bmatrix}$$

2.2 Predicting Pose and Feature Point Positions

Using results from Section 2.1, we compute the transformation matrix between frames i and j in camera coordinates:

$$\mathbf{T}_{\mathcal{C}_i}^{\mathcal{C}_j} = \begin{bmatrix} \mathbf{R}_{\mathcal{C}_i}^{\mathcal{C}_j} & \mathbf{t}_{\mathcal{C}_i}^{\mathcal{C}_j} \\ \mathbf{0} & 1 \end{bmatrix}$$

where the rotation matrix and translation vector are:

$$\mathbf{R}_{\mathcal{C}_i}^{\mathcal{C}_j} = \mathbf{R}_{\mathcal{J}}^{\mathcal{C}}\mathbf{R}_{\mathcal{C}_0}^{\mathcal{J}_i}\mathbf{R}_{\mathcal{C}_0}^{\mathcal{J}_j}\mathbf{R}_{\mathcal{C}}^{\mathcal{J}}$$
$$\mathbf{t}_{\mathcal{C}_i}^{\mathcal{C}_j} = \mathbf{R}_{\mathcal{J}}^{\mathcal{C}}(\mathbf{t}_{\mathcal{C}_0}^{\mathcal{J}_j} - \mathbf{t}_{\mathcal{C}_0}^{\mathcal{J}_i}) + \mathbf{t}_{\mathcal{J}}^{\mathcal{C}} - \mathbf{R}_{\mathcal{C}_i}^{\mathcal{C}_j}\mathbf{t}_{\mathcal{J}}^{\mathcal{C}}$$

Mapping normalized coordinates $\mathbf{p}_i$ from frame i to frame j yields:

$$\mathbf{p}_j^* = \mathbf{R}_{\mathcal{C}_i}^{\mathcal{C}_j}\mathbf{p}_i + \mathbf{t}_{\mathcal{C}_i}^{\mathcal{C}_j}$$

where $\mathbf{p}_i$ is the homogeneous coordinate vector of the feature point, and $\mathbf{p}_j^*$ is the IMU-predicted normalized coordinate in frame j.

---

3 Motion Blur Removal and Improved Optical Flow

The motion blur restoration model describes a sharp image $\mathbf{f}(x,y)$ degraded by a blur function plus noise:

$$\mathbf{g}(x,y) = \mathbf{f}(x,y) * \mathbf{h}(x,y) + \mathbf{n}(x,y)$$

where $\mathbf{h}(x, y)$ is the blur operator or point spread function, $*$ denotes convolution, and $\mathbf{n}(x, y)$ is additive noise. Restoration methods are classified as blind or non-blind based on prior motion knowledge. Since IMU provides acceleration and angular velocity to determine motion, we focus on non-blind restoration.

3.1 Non-Blind Motion Blur Restoration with IMU Prior

Given known timestamps and time interval Δt_{ij} between frames, robot velocity is obtained from Eq. (7), and normalized coordinates $\mathbf{p}_i$, $\mathbf{p}_j$ from Eq. (11). Motion blur direction angle θ and length L are:

$$\theta = \arctan\left(\frac{v_j - v_i}{u_j - u_i}\right)$$
$$L = \|\mathbf{v}\| \times \Delta t_{ij}$$

where $\mathbf{v}$ is robot velocity. The point spread function dimensions are:

$$b_h = L \times \cos\theta$$
$$b_w = L \times \sin\theta$$

The point spread function $\mathbf{h}(x, y)$ is defined implicitly as:

$$\mathbf{h}(x, y) = \begin{cases} \frac{1}{b_h \times b_w} & \text{for } 0 \le x \le b_w - 1, 0 \le y \le b_h - 1 \\ 0 & \text{otherwise} \end{cases}$$

The blur restoration model is:

$$\mathbf{f}(x, y) = \arg\min_{\mathbf{f}} \|\mathbf{g}(x, y) - \mathbf{f}(x, y) * \mathbf{h}(x, y)\|^2$$

where the fidelity term ensures consistency between the sharp image convolved with the point spread function and the blurred image. This can be solved efficiently via Fast Fourier Transform.

3.2 Improved Optical Flow Optimization for Feature Matching

Building upon the LK optical flow' s brightness constancy assumption, we first introduce gradient constancy:

$$\nabla \mathbf{I}_1(x, y) = \nabla \mathbf{I}_2(x + \Delta x, y + \Delta y)$$

where $\mathbf{I}_1(x, y)$ and $\mathbf{I}_2(x, y)$ are grayscale values in reference and current frames, and ∇ denotes image gradient. To prevent overfitting, we use the L1 norm of the second frame' s gradient to model sparse noise, constructing the regularization term. The final model is:

$$\begin{aligned}\arg\min_{\Delta x, \Delta y} &\|\mathbf{I}_1(x,y) - \mathbf{I}_2(x+\Delta x, y+\Delta y)\|^2 \\ &+ \alpha\|\nabla\mathbf{I}_1(x,y) - \nabla\mathbf{I}_2(x+\Delta x, y+\Delta y)\|^2 \\ &+ \beta\|\nabla\mathbf{I}_2(x+\Delta x, y+\Delta y)\|_1\end{aligned}$$

where α and β are weighting coefficients for gradient and sparse noise terms.

The IMU-predicted feature position from Section 2.2 serves as the initial value for Gauss-Newton optimization. We enhance the fixed-step Gauss-Newton method with BB variable step sizes. Taylor expanding the non-linear function $\mathbf{F}(\mathbf{X})$ to second order:

$$\mathbf{F}(\mathbf{X}) \approx \mathbf{F}(\mathbf{X}_k) + \nabla\mathbf{F}(\mathbf{X}_k)(\mathbf{X} - \mathbf{X}_k) + \frac{1}{2}(\mathbf{X} - \mathbf{X}_k)^T\mathbf{H}(\mathbf{X}_k)(\mathbf{X} - \mathbf{X}_k)$$

The optimization direction is:

$$\mathbf{d}_k = -\mathbf{H}^{-1}(\mathbf{X}_k)\nabla\mathbf{F}(\mathbf{X}_k)$$

For a general quadratic optimization problem $\mathbf{F}(\mathbf{X}) = \frac{1}{2}\mathbf{X}^T\mathbf{A}\mathbf{X} + \mathbf{b}^T\mathbf{X}$, the gradient at step k is $\mathbf{g}_k = \mathbf{A}\mathbf{X}_k + \mathbf{b}$. The BB step size is derived from the secant equation:

$$\mathbf{S}_{k-1} = \mathbf{X}_k - \mathbf{X}_{k-1}, \quad \mathbf{Y}_{k-1} = \nabla\mathbf{F}(\mathbf{X}_k) - \nabla\mathbf{F}(\mathbf{X}_{k-1})$$

The step size at iteration k is:

$$\delta_k = \frac{\mathbf{S}_{k-1}^T\mathbf{Y}_{k-1}}{\|\mathbf{Y}_{k-1}\|^2}$$

This yields the update direction:

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \delta_k\mathbf{H}^{-1}(\mathbf{X}_k)\nabla\mathbf{F}(\mathbf{X}_k)$$

The BB step size depends only on the difference between consecutive variable states and gradient information, enabling adaptive adjustment based on the loss function.

4 Experimental Results and Analysis

All experiments were conducted on a uniform platform: AMD Ryzen 5 3500U with Radeon Vega Gfx quad-core processor, 12GB RAM, 2.10GHz base frequency, Ubuntu 18.04 OS.

We first evaluate the standalone algorithm performance by comparing matching time and accuracy between consecutive frames against LK optical flow and ORB feature detection in VINS-Mono. To further verify accuracy and system stability, we integrate the algorithm into VINS-Mono and compare localization performance against VINS-Mono, PL-VIO, VINS-Fusion monocular-IMU (Mono-IMU), and VINS-Fusion stereo-IMU (Stereo-IMU).

The dataset used is the EuRoC MAV (Micro Aerial Vehicle) visual-inertial dataset collected at ETH Zurich, featuring a machine hall and two ordinary rooms. To validate effectiveness across scenarios, we use nine sequences of varying difficulty: MH_02_easy, MH_03_medium, MH_04_difficult, V1_01_easy, V1_02_medium, V1_03_difficult, V2_01_easy, V2_02_medium, V2_03_difficult.

4.1 Inter-Frame Algorithm Runtime Comparison

As our algorithm belongs to the optical flow family, we control for feature extraction using identical Shi-Tomasi corner detection (OpenCV' s `cv::goodFeaturesToTrack` with parameters: max corners 500, quality level 0.01, min distance 30). Averaged over 1000 iterations, our algorithm runs in 1.361 ms compared to 2.892 ms for OpenCV LK optical flow, representing a 52.939% speed improvement (Figure 2).

When comparing with feature-based methods, descriptor computation time must be included. Averaged over 1000 iterations, our algorithm requires 8.382 ms total, while OpenCV ORB extraction and matching takes 35.110 ms, achieving a 76.125% speed improvement (Figure 3).

4.2 Inter-Frame Matching Accuracy Evaluation

For correctly matched feature points $\mathbf{p}_i$ and $\mathbf{p}_j$ between frames, the epipolar constraint should satisfy $\mathbf{p}_j^T \mathbf{F} \mathbf{p}_i = 0$. Due to measurement errors, we evaluate the least-squares solution and define a matching score:

$$\text{score} = \frac{1}{1 + \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_j^T \mathbf{F} \mathbf{p}_i\|_\Sigma^2}$$

where smaller reprojection error yields higher scores. As shown in Table 1, our algorithm improves accuracy by 53.295% over LK optical flow, validating the improved optimization model (Eq. 17) that introduces gradient error and sparse noise regularization. Compared to ORB, accuracy decreases by 13.554% but runtime improves by 76.125%, an acceptable trade-off for real-time applications.

Figures 4-6 show matching results on MH_02_easy sequence. Without initial values (defaulting to identity rotation and zero translation), high motion blur causes significant mismatches (Figure 4). With IMU-predicted initial values enabling local search, mismatches are largely eliminated. OpenCV LK still exhibits errors (e.g., points #2, 11, 20, 21, 22, 27, 33, 39, 49), while our algorithm achieves robust matching (Figures 5-6).

4.3 System Localization Accuracy Evaluation

Integrating our algorithm into VINS-Mono front-end, we compare against state-of-the-art methods using the EVO evaluation tool. Following prior work, we use Absolute Trajectory Error (ATE) RMSE and mean error as metrics, averaging over 20 runs (Table 2). Our algorithm outperforms PL-VIO and matches VINS-Fusion performance, though VINS-Fusion occasionally diverges significantly on V2_03_difficult. Notably, MH_03_medium and V2_01_easy show degraded performance due to larger average IMU bias errors that violate our algorithm' s assumption of precise initialization. Without good initial values, the optimizer degrades to standard Gauss-Newton, increasing search time and reducing accuracy.

On other sequences, performance improves, most notably on V2_03_difficult (fast motion, severe blur) where RMSE decreases by 18.066% and mean error by 28.902%. Figures 7-10 demonstrate superior robustness during camera shake, with our algorithm maintaining tighter trajectory consistency and lower error statistics compared to VINS-Mono.

---

5 Conclusion

This paper proposes a novel algorithm that fuses visual and IMU data for motion blur removal and improved optical flow feature matching. The algorithm fully exploits IMU motion information to restore blurred images and provide initial values for iterative matching, reducing mismatch probability. As a generic module, it can be transplanted to other VI-SLAM frameworks.

Validation across three difficulty levels demonstrates superior efficiency and accuracy over OpenCV' s optical flow implementation. Integration into the state-of-the-art VINS-Mono framework improves pose estimation accuracy. Future work will investigate advanced deblurring with regularization terms to remove artifacts and further enhance matching precision.

References

[1] Durrant Whyte H, Bailey T. Simultaneous localization and mapping: part I [J]. IEEE Robotics & Automation Magazine, 2006, 13 (2): 99–110.

[2] Pradeep V, Rhemann C, Izadi S, et al. MonoFusion: Real-time 3D reconstruction of small scenes with a single web camera [C]// Proc of IEEE International Symposium on Mixed and Augmented Reality. Australia: IEEE, 2013: 83–88.

[3] 郭金辉, 陈秀万, 王媛. 视觉惯性 SLAM 研究进展 [J]. 火力与指挥控制, 2021, 46 (1): 1-8. (Guo JinHui, Chen Xiuwan, Wang Yuan. Research progress of visual inertia slam [J]. Fire control and command control, 2021, 46 (1): 1-8.)

[4] 胡立华, 左威健, 张继福. 采用逆近邻与影响空间的图像特征误匹配剔除方法 [J]. 计算机辅助设计与图形学学报, 2022, 34 (3): 449-458. (Hu LiHua, Zuo Weijian, Zhang Jifu. The image feature mismatch elimination method of inverse nearest neighbor and influence space is adopted [J]. Journal of Computer-Aided Design and Computer Graphics, 2022, 34 (3): 449-458.)

[5] 付煜, 郑爽, 别桐等. 融合点线特征的视觉惯性 SLAM 算法 [J]. 计算机应用研究: 2022, 39 (2): 1-8. (Fu Yu, Zheng Shuang, Bie Tong, et al. Visual inertial SLAM algorithm with fused point and line features [J]. Application Research of Computers: 2022, 39 (2): 1-8.)

[6] Triggs B, Mclauchlan P F, Hartley R I, et al. Bundle adjustment-A Modern Synthesis [C]// Proc of International Workshop on Vision Algorithms. Heidelberg: Springer, 1999: 298–372.

[7] 伞红军, 王汪林, 陈久朋等. 改进误匹配剔除的 SLAM 算法 [J]. 软件导刊, 2021, 20 (08): 99-104. (San Hongjun, Wang Wanglin, Chen Jiupeng, et al. Improved SLAM algorithm for error matching elimination [J]. Software Guide, 2021, 20 (08): 99-104.)

[8] Lowe D G. Distinctive image features from scale-invariant keypoints [J]. International Journal of Computer Vision, 2004, 60 (2): 91–110.

[9] Rosten E, Drummond T. Machine learning for high-speed corner detection [C]// Proc of European conference on computer vision. Berlin: Heidelberg, 2006: 430-443.

[10] Lucas B D, Kanade T. An iterative image registration technique with an application to stereo vision [C]// Proc of Imaging Understanding Workshop, 1981.

[11] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM [C]// Proc of European conference on computer vision. Cham: [s. n.], 2014: 834-849.

[12] Qin Tong, Li Peiliang, Shen Shaojie. VINS-Mono: A robust and versatile monocular visual-inertial state estimator [J]. IEEE Trans on Robotics, 2018, 34 (4): 1004–1020.

[13] Guan Qi, Wei Guoliang, Wang Licheng, et al. A novel feature points tracking algorithm in terms of IMU-aided Information Fusion [J]. IEEE Trans on Industrial Informatics, 2021, 17 (8): 5304–5313.

[14] Leutenegger S, Lynen S, Bosse M, et al. Keyframe-based visual–inertial odometry using nonlinear optimization [J]. The International Journal of Robotics Research, 2015, 34 (3): 314-334.

[15] Mourikis Anastasios I, Roumeliotis S I. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation [J]. IEEE International Conference on Robotics and Automation, 2007: 3565-3572.

[16] Davison A J. Active search for real-time vision [C]// Proc of Tenth IEEE International Conference on Computer Vision Volume 1. [S. l.]: IEEE, 2005: 66-73.

[17] Shi J, Tomasi C. Good features to track [C]// Proc of IEEE conference on computer vision and pattern recognition. [S. l.]: IEEE, 1994: 593-600.

[18] Fergus R, Singh B, Hertzmann A, et al. Removing camera shake from a single photograph [M]// [S. l.]: ACM SIGGRAPH, 2006: 787-794.

[19] Chen Liang, Fang Faming, Wang Tingting, et al. Blind image deblurring with local maximum gradient prior [C]// Proc of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 1742-1750.

[20] Mur-Artal R, Tardós J D. ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras [J]. IEEE Trans on Robotics, 2017, 33 (5): 1255-1262.

[21] Qin Tong, Shen Shaojie. Robust initialization of monocular visual-inertial estimation on aerial robots [C]// Proc of IEEE/RSJ International Conference on Intelligent Robots and Systems. Vancouver: IEEE, 2017: 5303-5310.

[22] Lupton T, Sukkarieh S. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions [J]. IEEE Trans on Robotics, 2012, 28 (1): 61–76.

[23] Shen Shaojie, Michael N, Kumar V. Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs [C]// Proc of IEEE International Conference on Robotics and Automation. [S. l.]: IEEE, 2015: 5303-5310.

[24] Chen Liang, Fang Faming, Lei Shen, et al. Enhanced sparse model for blind deblurring [C]// Proc of European Conference on Computer Vision. [S. l.]: Springer, 2020: 631-646.

[25] He Yijia, Zhao Ji, Guo Yue, et al. PL-VIO: tightly-coupled monocular visual-inertial odometry using point and line features [J]. Sensors, 2018, 18 (4): 1159.

[26] Qin Tong, Pan Jie, Cao Shaozu, et al. A general optimization-based framework for local odometry estimation with multiple sensors [J/OL]. (2019-01-11) [2022-04-11]. http://arxiv.org/abs/1901.03638.

[27] Grupp M. EVO: Python package for the evaluation of odometry and SLAM [J/OL]. (2017-01-01) [2022-04-11]. https://github.com/MichaelGrupp/evo.

Note: Figure translations are in progress. See original paper for figures.