Loosely Coupled Stereo Inertial Odometry on Low-cost System

HaoChih, LIN[*] Francois, Defay[†]

Abstract — We present a fast and robust stereo visual inertial odometry system which is friendly for low cost sensor and single board computer (SBC). Comparing against other research which uses tightly coupled algorithms or nonlinear optimization to increase accuracy in custom powerful hardware or desktop environment, our system adopts the loosely coupled ESKF to limit the computational complexity in order to fit limited CPU resources and run in real-time on an ARM based SBC. The experiments demonstrates our method could be implemented in both indoor and outdoor scenarios with competitive accuracy. Furthermore, the usage of forward facing stereo cameras also provides the ability of obstacles avoidance. The result are released as an open sourced Robot Operation System (ROS[1]) package.

I INTRODUCTION

In order to make a micro aerial vehicle (MAVs) achieve fully autonomous navigation in GPS denied environment, the one of fundamental challenge is to obtain fast, robust and accurate 3D egomotion estimation with others sensors. Recently, the visual inertial odomerty (VIO) attracts significant attentions in MAV researching field because the algorithm efficiently integrates the rich representation of a scene captured in an image, together with the accurate short-term measurements by IMU. In addition, comparing to heavy and expensive 3D laser scanner or rgbd sensor, the VIO approach could be able to present similar performance with much less value in both price and size aspects which may increase the endurance and maneuverability of MAVs.

In this paper, we propose a loosely coupled stereo inertial odometry based on error state kalman filter (ESKF) algorithm. The main contribution of our method is to limit the computational loading by using the characteristics of loosely coupled architecture which uses the fixed dimension of state space in order to be implemented on low-cost ARM based SBC (Odroid XU4[2]). The algorithm could run at around 100Hz depending on IMU publishing rate and works with forward-facing stereo cameras, allowing for fast flight and removing the need for a second camera for collision avoidance. Apart from this, our approach provides steady metric scale information from disparity matching to meet the power on and go requirement without requiring specific initialization.

[*] Advanced Robotics Co., Ltd, Taiwan. Email: haochihlin93@gmail.com
[†] ISAE-Supaero, France. Email: francois.defay@isae-supaero.fr
[1] http://wiki.ros.org/
[2] http://www.hardkernel.com/main/main.php

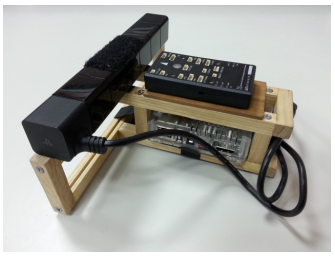

Figure 1: The hardware overview of proposed VIO system composed of ARM based SBC, low cost IMU, and modified PS4-Eye.

II RELATED WORK

The fusion of Inertial Measurement Units (IMU) with a visual sensors has become popular in robotics community. Recent work can be categorized to two approaches: tightly coupled and loosely coupled system. The former, e.g. [1–3], jointly estimates the image features information with IMU data. The latter, [4–6], take the visual odometry(VO) as a independent black box. By means of the characteristics of loosely coupled architecture, we could divide our algorithm into two main sections: VO and ESKF part.

II.1 VO part

There is a vast amount of existing visual algorithms to estimate the locomotion from traditional feature extraction and matching method, for instance PTAM [7] and ORB-SLAM [8], to semi dense approaches, which uses features to locate small patches, then operates directly on pixel intensities. The representative of this method is SVO [9] and LSD-SLAM [10]. In addition, thanks to the significant improvement on CPU/GPU computing power, the current state-of-art method avoid feature detection process instead using all pixel information directly from image stream. The famous achievements are [11, 12]. However, these two approaches generally rely on powerful or custom hardware. Hence, this is the main reason that we choose the traditional features based method for our stereo visual odometry developing.

II.2 ESKF part

The one common way to fuse IMU with odometry or other types of sensor data is EKF (Extended Kalman Filter). Although, recently, some researches adopts other nonlinear optimizer to get more accuracy and robust performance, the computing cost of the method is too larger to be implemented on ARM-based SBC. As a result, in our system, we apply

http://www.imavs.org/

http://www.imavs.org/

ESKF algorithm to overcome the limited CPU issue and also the main drawbacks of traditional EKF architecture. According to the experiments and explanation from [13, 14], ESKF has following remarkable assets: the orientation error-state is minimal to avoid over-parametrization issue, the error-state system is always operating close to the origin to guarantee the linearization validity and the error dynamics are slow, hence the rate of correction could be lower than prediction rate.

The rest of the paper is organized as follows. Section III shows the definition of the notations. Section IV describes the core algorithm of VO and statical way to remove outliers. The architecture of ESKF is fully explained in section V, followed by section VI, which describes the detail of practical implementation. Finally, section VII presents the experimental result and concludes the paper as well as future research directions.

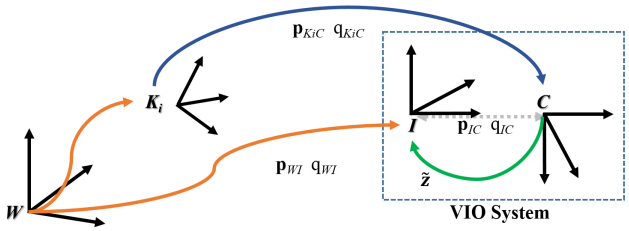

Figure 2: Visualization of the different coordinate frames. Blue lines denote the transformation obtained by VO, red lines stand for the prediction part of ESKF and green lines depict the update part

III NOTATION AND DEFINITIONS

We employ the following notation throughout this work and are illustrated in Figure 2. For the Frame definitions: W is the world frame, F_i is the i-th keyframe frame, I is IMU frame and C is camera frame. A translation vector from frame A to frame B expressed in frame A is denoted as bold $\mathbf{p}_{ab}$. A rotation matrix from A to B is expressed as $\mathbf{R}_b^a \in SO3$ which can also be denoted as a quaternion $\bar{q}_{ab}$. In this paper, we adhere the Hamilton convention [15], which is right-handed and widespread used in robotics, to define a quaternion by $q = [q_w + q_x + q_y + q_z] = [q_w, \mathbf{q}]$. We adopt the notation introduced in [1] to handle the quaternion multiplication as $q_{ac} = q_{ab} \otimes q_{bc}$.

IV STEREO VISUAL ODOMETRY

Take the limits of CPU resources into account, our VO approach adopt ORB [16] for both feature detection and description instead of robust but slow descriptors like SURF or SIFT. We also apply the concept of keyframe to decrease the drift effect. In the other hand, unlike other VO researches [8, 17] emphasis on back-end optimization process, e.g. bundle adjustment or graphic based loop closure, to tackle the

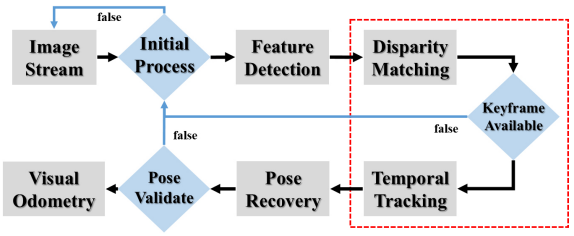

Figure 3: The system overview of proposed VO algorithm.

long-term global localization issue, we are more focusing on the short-term accuracy and robustness because the VO outcome will be used as the local correction inputs for the ESKF based on the previous keyframe pose rather than global correction referred to the initial VO frame. Especially speaking, the continuity of VO does not hugely affect the system performance, hence the back-end optimization process is not mandatory. This decision also descends the demands of CPU resource.

IV.1 Disparity Computing

The system overview of our VO approach is illustrated in Figure 3. Starting from disparity block, the scale metric information can be calculated immediately and continuously from the initial phase due to known camera calibration matrix. The matching method in this block is mainly depends on the brutal force matcher with additional epipolar constraints. Then the succeeded matching pairs will be sent into tracking block. The tracking part is composed of two procedures: the Temporal Features Matching and Pose Recovery.

IV.2 Temporal Features Matching

The state-of-art method to handle the issue is using the circular searching pattern strategy, e.g. [1, 8], which searches similar features pairs from both temporal and disparity correspondences to reduce the outlier probability, however, since the stereo camera in our system is hardware synchronizes and its baseline is very short, the effect is not significant compared to other system configuration. Therefore, our system only relies on left side images to search correspondences between two sets of descriptors. In order to minimize the dissimilarity score between each pair of descriptors p_{ij} and increase the matching robustness, we adopt exhaustive searching strategy, also called brutal force, to find all possible pairs. The main defect of this strategy is the bad correspondences, especially for indoor environments with many repetitive patterns. To solve this issue, we introduced a statistics model to determine which pair is outlier. Every time when the exhaustive searching strategy generates a list of matching pairs, the system will compute the mean μ and standard deviation σ based on Hamming distance ($HD(p_{ij})$) of ORB descriptors. Then the system would decide the outlier by the following equation:

$$if\ HD(p_{ij}) > \mu + a\sigma, \quad then\ p_{ij} \in Outlier \qquad (1)$$

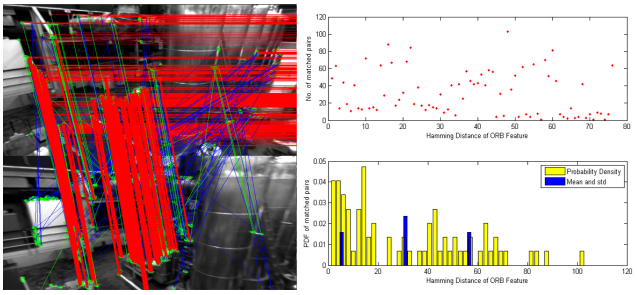

Figure 4: Left: temporal image pair for features tracking matches. Right: histogram to show distribution of matched pair Hamming distance.

where a is a custom defined weighted factor. The outcome of this approach is quite competitive to the circular strategy. Figure 4 demonstrates the performance of our strategy from both raw images and histogram graphic views. The green lines of raw images stands for original brutal force matching result. The blue lines represents the remaining pairs after statistics based outliers removing. The red line is the inlier of PnP RANSAC outputs, which will be explained in the following section.

IV.3 Pose Recovery

There are three types of standard approaches for motion estimation: 2D-2D, 2D-3D and 3D-3D. The first method is based on specified 2D features correspondences in a image pair. The popular related algorithms are 5 and 8 points solution [18]. The main issue of 2D-2D is observability of scale information. The final method also called the point cloud registration which is widely implemented on RGBD or 3D laser scanner since the depth data is highly reliable and accuracy. The popular algorithms include ICP [19] and Bundle Adjustment [20]. None of both methods satisfy our scenario, thus, we choose the 2D-3D approach in our VO system. Currently, there are 3 existing algorithms supported by OpenCV[3] library, they are: Iterative PnP, EPNP and P3P [21, 22]. The main concept of the algorithm is to find the minimal reprojected error by Equation 2 from 3D structure and 2D image correspondences.

$$T_{k-1}^{k} = \underset{T_{k-1}^{k}}{\arg\min} \sum_{i} \left\| q_k^i - \hat{q}_{k-1}^i \right\|^2 \tag{2}$$

where T_{k-1}^{k} is a transformation matrix from k-1 to k, q_k^i is a 2D description vector of i-th feature in k frame. $\hat{q}_{k-1}^i$ is a projected corresponding 2D descriptor from k-1 matched 3D point.

Because of various characteristics among these options, we let the users to select the one satisfied their demands. In addition, to tackle the bad corresponding matching pair generated by previous block, all three algorithms are accompanied by RANSAC process in order to remove outliers.

[3] http://opencv.org/

IV.4 Keyframe Selection and Initialization

The accumulated drift issue is the most common problem among different types of odometry. For the VO system, the most efficient way to decrease the rate of drift accumulation is to introduce the keyframe concept like [1–3, 7–12]. Therefore, we apply the keyframe architecture into our system. The first keyframe will be selected during the initialization process. The VO system will continuously re-initialize until the first keyframe is selected successfully based on the quality of disparity and tracking matching. When the re-initialization occurs, which will be triggered as system detects the lose tracking phenomenon, system will drop current keyframe and try to set up a new one.

IV.5 Practical Aspect

Although our system allows the users to modify parameters to meet their requirements, to get the optimal performance on limited CPU resources SBC, we suggest the resolution of input video stream should not be higher than 320*240. The maximum number of extracted feature points from an image extremely affects the realtime performance.

Moreover, as mentioned above, the primary purpose of our VO system is to provide reliable and accuracy short-term odometry based on the current keyframe pose. Hence, we raise the threshold for tracking matching constraints and re-initialize whole VO system right away when encountering lose tracking scenario. The outcomes largely decrease the CPU loading. The format of output data is a transformation matrix T_{KC}, consists of q_{K_iC} and $\mathbf{p}_{K_iC}$, from latest Keyframe (K_i) to Camera frame (C) expressed in the Keyframe.

V ERROR STATE KALMAN FILTER

The most well-known papers to illustrate the detail of ESKF formula derivation are [13, 14]. This section mainly adheres the concepts shown in both papers, but we replace the JPL's convention with Hamilton's for quaternion definition, in addition, with the benefits of loosely coupled system property, the size of state space and related matrix dimension are all fixed.

In the following subsections, the state of the filter is denoted as a 16-elements state vector $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}_{WI} & \mathbf{v}_{WI} & q_{WI} & \mathbf{b}_a & \mathbf{b}_\omega \end{bmatrix}^T \tag{3}$$

where $\mathbf{p}_{WI}$ is the position of IMU frame's origin (I) in the inertial world frame (W). $\mathbf{v}_{WI}$ is a velocity vector of IMU. q_{WI} describes a rotation from IMU frame to world frame. $\mathbf{b}_a$ and $\mathbf{b}_\omega$ are bias vector of gyro and acceleration respectively. The measurements of the IMU are known to be subject to different error terms, such as a process noise and a bias. Thus, for the real angular velocities $\boldsymbol{\omega}$ and the real accelerations $\mathbf{a}$,

http://www.imavs.org/

http://www.imavs.org/

which is expressed in sensor frame, we have following relations:

$$\boldsymbol{\omega} = \boldsymbol{\omega}_m - \mathbf{b}_\omega - \mathbf{n}_\omega, \quad \mathbf{a} = \mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a \tag{4}$$

where the subscript m denotes the measured value. $\mathbf{n}_a$ and $\mathbf{n}_\omega$ are zero mean white Gaussian noise processes. The bias is non-static and simulated as a random walk process: $\dot{\mathbf{b}}_\omega = \mathbf{n}_\omega$, $\dot{\mathbf{b}}_a = \mathbf{n}_a$.

In order to make the filter more robust and converges faster, we assume the calibrated transformation between IMU frame (I) and Camera frame (C) are fixed. The calibration states are denoted as q_{IC} for rotation from camera to IMU frame and $\mathbf{p}_{IC}$ for translation of camera frame expressed in IMU frame.

Because of limited space and considering the practical implementation issue of numerical integration on embedded system, the following subsections only depicts formulas in discrete time domain. For more information of continuous time model, the reader is referred to [13, 14].

V.1 Prediction

The main concept of the error-state filter is to treat the true state ($\mathbf{x}_t$) as a composition of the nominal state ($\mathbf{x}$: the state is integrated by high-frequency IMU data without considering noise terms and other possible model imperfections) and the error state ($\delta\mathbf{x}$: the state is in charge of collecting all the noise and perturbations). The relation is expressed as: $\mathbf{x}_t = \mathbf{x} \oplus \delta\mathbf{x}$, where the operator $\oplus$ indicates a generic composition. For all vectors in state space are equal to the typical addition symbol (+), except for the quaternion, it is equivalent to the quaternion multiplication symbol ($\otimes$) Additionally, ESKF also continuously predicts a Gaussian estimate of the error-state.

The following differential equations govern the nominal state kinematics:

$$\mathbf{p}_{k+1} \leftarrow \mathbf{p} + \mathbf{v}\Delta t + 0.5(\Omega_{(q)}(\mathbf{a}_m - \mathbf{a}_b) + \mathbf{g})\Delta t^2 \tag{5}$$
$$\mathbf{v}_{k+1} \leftarrow \mathbf{v} + (\Omega_{(q)}(\mathbf{a}_m - \mathbf{a}_b) + \mathbf{g})\Delta t \tag{6}$$
$$q_{k+1} \leftarrow q \otimes q[(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\Delta t] \tag{7}$$
$$\mathbf{a}_b, \boldsymbol{\omega}_b, \mathbf{g} \quad are\ constant \tag{8}$$

where $\Omega_{(q)}$ is the quaternion multiplication matrix of nominal quaternion q and $q[\boldsymbol{\omega}]$ is a quaternion converted by a rotation vector $\boldsymbol{\omega}$. Becasue the rate of prediction depends on IMU publishing rate which is usually higher than 100 Hz and almost constant, we assume the angular velocity over the period Δt is also invariant. Therefore, $q[\boldsymbol{\omega}]$ could be interpreted as the Taylor series of $\boldsymbol{\omega}\Delta t$ by midward zeroth order integration:

$$q_{k+1} \leftarrow q \otimes (1 + \frac{1}{2}\bar{\boldsymbol{\omega}}\Delta t + \frac{1}{2!}(\bar{\boldsymbol{\omega}}\Delta t)^2 + ...) \tag{9}$$

where $\bar{\omega}_k = 0.5(\boldsymbol{\omega}_{k+1} + \boldsymbol{\omega}_k)$. In real-time implementation, the system calculates the Taylor series up to 4th order.

To increases numerical stability and handles the minimal representation of quaternion computing, we defined the error quaternions as $\delta q = q \otimes \hat{q} \approx [1, \frac{1}{2}\delta\boldsymbol{\theta}^T]^T$. Thus, we define a 15-elements error state vector:

$$\delta\mathbf{x} = \begin{bmatrix}\delta\mathbf{p} & \delta\mathbf{v} & \delta\boldsymbol{\theta} & \delta\mathbf{b}_a & \delta\mathbf{b}_\omega\end{bmatrix}^T \tag{10}$$

There are several ways to illustrate the compact form of the error-state kinematics, we modify the form listed in [5] which is expressed as following:

$$\delta\mathbf{x} \leftarrow \mathbf{F}_d(\mathbf{x}, \mathbf{u_m})\delta\mathbf{x} + \mathbf{G}_c\mathbf{Q}_d\mathbf{G}_c^T\mathbf{n} \tag{11}$$

where $\mathbf{u}_m = [\mathbf{b}_a\ \mathbf{b}_\omega]^T$ is the input vector, and $\mathbf{n} = [\mathbf{n}_a\ \mathbf{n}_\omega]^T$ is the noise vector ($\mathbf{n} \sim N\{0, \mathbf{Q}_d\}$). $\mathbf{F}_d \in \mathbb{R}^{15\text{x}15}$ and $\mathbf{Q}_d = diag(\sigma^2_{\boldsymbol{b}_a}\Delta t^2, \sigma^2_{\boldsymbol{b}_\omega}\Delta t^2, \sigma^2_{\boldsymbol{n}_a}\Delta t, \sigma^2_{\boldsymbol{n}_\omega}\Delta t)$ are the discrete state transition matrix and noise covariance matrix respectively. $\mathbf{G}_c = [zeros(3, 12);\ identity(12)]$. To find the ESKF prediction equation, we calculate the expectation of Equation (11) and its state covariance matrix $\mathbf{P}$:

$$\hat{\delta\mathbf{x}} \leftarrow \mathbf{F}_d(\mathbf{x}, \mathbf{u_m})\hat{\delta\mathbf{x}} \tag{12}$$
$$\mathbf{P}_{k+1|k} = \mathbf{F}_d\mathbf{P}_{k|k}\mathbf{F}_d^T + \mathbf{G}_c\mathbf{Q}_d\mathbf{G}_c^T \tag{13}$$

Generally, the error-state $\delta\mathbf{x}$ is set to zero during initialization phase, so Equation (12) always returns zero. For the structure of matrix $\mathbf{F}_d$, we adopt the compact form defined in [realtime metric], with the small angle approximation when $|\boldsymbol{\omega}| \rightarrow \mathbf{0}$. We now proceed the ESKF prediction as follows:

1. Propagate the nominal state variables according to Equation from (4) to (9)
2. Calculate the matrix $\mathbf{F}_d$ and $\mathbf{Q}_d$
3. Compute the state covariance matrix by Equation (13)

V.2 Updates

The update procedure will be triggered whenever a validate visual measurement is generated by the stereo odometry algorithm. As described above, the VO system adopts the keyframe concept to reduce the effect of accumulated drift, hence, the odometry used in update phase represents the transformation from the state at keyframe to the present. Unlike other loosely coupled approaches, e.g. [6], used globe pose of visual measurements to update EKF states, we choose the local odometry to correct the ESKF error states. The main advantage of this approach is the computational delay or losing track scenario will not hugely affect the stability and performance of ESKF.

The measurement model in our case is quite straightforward since the transformation between IMU frame and Camera frame is assumed to be constant and known by means of the system extrinsic calibration. The measurement vector ($\mathbf{z}^{vo}$) is expressed as follows:

$$\mathbf{z}^{vo} = \begin{bmatrix}\mathbf{p}^{vo}_{WI} \\ q^{vo}_{WI}\end{bmatrix} = \begin{bmatrix}\Omega_{(q_{WKi})}(\Omega_{(q_{CI})}\mathbf{p}_{KiC} + \mathbf{p}_{CI}) + \mathbf{p}_{WK_i} + \mathbf{n}_p \\ q_{WK_i} \otimes (q_{CI} \otimes q_{K_iC} \otimes q_{CI}^{-1}) \otimes \delta q_{n_q}\end{bmatrix} \tag{14}$$

where $\mathbf{p}_{WI}^{vo}$ represents the translation from world to imu frame observed by VO system, q_{WI}^{vo} is for rotation. $\mathbf{n}_p$ and $\mathbf{n}_q$ are zero mean, white Gaussian noise of the visual measurement. q_{K_iC} and $\mathbf{p}_{K_iC}$ are the odometry output generated by VO algorithm.

Then we calculate the update residual between propagated state and measurement state:

$$\tilde{\mathbf{z}} = \mathbf{z} \ominus \hat{\mathbf{z}} = \begin{bmatrix} \mathbf{p}_{WI}^{vo} - \mathbf{p}_{WI}^{imu} \\ q_{WI}^{vo} \otimes q_{WI}^{imu\,-1} \end{bmatrix} \tag{15}$$

Then we approximates the $q_{WI}^{vo} \otimes q_{WI}^{imu\,-1} \approx [1, -0.5\tilde{\boldsymbol{\theta}}^T]^T$, and linearise the Equation (15) with respect to the error-state:

$$\tilde{\mathbf{z}} = \mathbf{H}\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{0}_{3x6} & \mathbf{0}_{3x6} \\ \mathbf{0}_{3x6} & \mathbf{I}_{3x3} & \mathbf{0}_{3x6} \end{bmatrix} \tilde{\mathbf{x}} \tag{16}$$

where $\tilde{\mathbf{x}} \in \mathbb{R}^{15\mathrm{x}1}$. Now we can apply the standard EKF update procedure:

1. Compute the residual from Equation (15)
2. Compute the innovation matrix: $\mathbf{S} = \mathbf{HPH}^T + \mathbf{V}$
3. Compute the Kalman gain: $\mathbf{K} = \mathbf{PH}^T\mathbf{S}^{-1}$
4. Compute the correction: $\hat{\tilde{\mathbf{x}}} = \mathbf{K}\tilde{\mathbf{z}}$
5. Compute the state covariance matrix: $\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{KH})\mathbf{P}_{k+1|k}(\mathbf{I} - \mathbf{KH})^T + \mathbf{KVK}^T$

where $\mathbf{V}$ is the covariance matrix of visual measurement Gaussian noise. After obtaining the correction vector $\hat{\tilde{\mathbf{x}}}$, the nominal state of ESKF gets updated with the particular compositions with the correction state: $\mathbf{x} \leftarrow \mathbf{x} \oplus \hat{\tilde{\mathbf{x}}}$.

V.3 Delay Handler

In practical, the computational loading of VO algorithm is much heaver than ESKF core, therefore, it is a common situation that the time-stamp of odometry output is slower than current ESKF state. Hence, we introduce the sliding window method, proposed in [1, 3, 6], to handle the measurement delay issue. The concept of the method is to store the past states in a fixed size buffer. Once the delayed measurement is available, the update procedure will try to find the past state which has most similar time-stamp among buffer, then re-propagate all stored states after the one used for update.

VI IMPLEMENTATION

We take the ideas and concepts of [6, 23] as references to develop the software architecture of our VIO system. The whole package is developed by C++ under ARM based Linux environment. We adopt OpenCV library for image processing functions, especially the PnP solver with RANSAC iteration. For the matrix, vector and quaternion computing, the Eigen library is used for the reasons of numerical efficiency and its cross platform stability. We also employ the Boost library for multi-thread CPU computing for ESKF updates. The system provides the ROS wrapper for easily integration with other ROS packages. The result is fully open source with detail comments on core section in order to help other developer to understand quickly and be able to modify the codes depending on their demands. The package has been released on github link[4]. To get the best performance, the user has to run both camera intrinsic and camera-IMU extrinstic calibration as accurate as possible. The recommended package to execute this nonlinear optimization

The hardware overview is shown in Figure 1. We hacked a PS4-Eye[5], which is a low-cost, hardware synchronized, high resolution and FPS stereo camera, to be able to connect with Odroid XU4 through USB 3.0 port with modified firmware. The accelerometer and gyroscope data is obtained from mpu6050 (a type of low-cost MEMS IMU) which is installed on Pixhawk autopilot micro-controller. The total weight of overall system is lower than 250 grams, and the price of whole system is not higher than 150 dollars.

VII EXPERIMENTS AND CONCLUSION

VII.1 Experimental result

At present, the proposed VIO system has only been tested on hand-held scenario and EuRoC MAV Dataset. For the hand-held experiments, the performance highly depends on the quality of both intrinsic and extrinsic calibration, the complexity of texture and the motion behaviour. The ESKF propagation rate on Odroid XU4 could achieve 100Hz associated to Pixhawk IMU data publishing frequency.

For the MAV dataset, because the resolution of the image stream is too high for Odroid, the testing is executed on the desktop. The system performance is related to the parameter settings of VO algorithm.

VII.2 Conclusion and future direction

We propose a low-cost and lightweight loosely coupled visual inertial odomtery. The result is open source and easily to be implemented on any kinds of unmanned vehicle. Due to the benefits of forward facing stereo camera configuration, the platform which carries our approach could achieve obstacles avoidance without adding extra camera and be able to accomplish power-on-and-go scenario with no needs of particular initialization process.

Currently, the two most time consuming procedures among our VIO system are feature extraction and PnP problem solving. To improve the performance, the former task, from image processing view, could be divided into several cells. Then assigns each CPU thread to handle a subset of cells parallel. For the latter, if we could guarantee the correct rate of inputs feature matching pairs, the number of iteration and criteria of RANSAC procedure would be relaxed, thus, the VIO correction rate will increase.

[4] https://github.com/jim1993/StereoVIO

[5] https://www.playstation.com/en-us/explore/accessories/playstation-camera-ps4/

http://www.imavs.org/

http://www.imavs.org/

REFERENCES

[1] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[2] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 1885–1892, IEEE, 2016.

[3] N. de Palézieux, T. Nägeli, and O. Hilliges, "Duo-vio: Fast, light-weight, stereo inertial odometry," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 2237–2242, IEEE, 2016.

[4] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-imu-based localization: Observability analysis and consistency improvement," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 182–201, 2014.

[5] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 957–964, IEEE, 2012.

[6] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 3923–3929, IEEE, 2013.

[7] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pp. 225–234, IEEE, 2007.

[8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[9] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 15–22, IEEE, 2014.

[10] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*, pp. 834–849, Springer, 2014.

[11] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct slam with stereo cameras," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 1935–1942, IEEE, 2015.

[12] A. Concha, G. Loianno, V. Kumar, and J. Civera, "Visual-inertial direct slam," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 1331–1338, IEEE, 2016.

[13] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, p. 2005, 2005.

[14] S. M. Weiss, *Vision based navigation for micro helicopters*. PhD thesis, Citeseer, 2012.

[15] W. R. Hamilton, *Elements of quaternions*. Longmans, Green, & Company, 1866.

[16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564–2571, IEEE, 2011.

[17] M. Labbe and F. Michaud, "Appearance-based loop closure detection for online large-scale and long-term operation," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 734–745, 2013.

[18] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[19] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP Variants on Real-World Data Sets," *Autonomous Robots*, vol. 34, pp. 133–148, Feb. 2013.

[20] F. Fraundorfer, D. Scaramuzza, and M. Pollefeys, "A constricted bundle adjustment parameterization for relative scale estimation in visual odometry," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 1899–1904, IEEE, 2010.

[21] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[22] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, pp. 155–166, 2009.

[23] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*, Springer, July 2014.