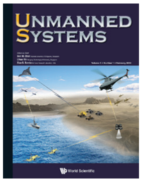
Unnamed Systems, Vol. 8, No. 2 (2020) 179–190
© World Scientific Publishing Company
DOI: 10.1142/S2301385020500119

UNMANNED SYSTEMS
https://www.worldscientific.com/worldscinet/us



Loosely-Coupled Ultra-wideband-Aided Scale Correction for Monocular Visual Odometry

Thien Hoang Nguyen*, Thien-Minh Nguyen†, Muqing Cao‡, Lihua Xie§

School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore 639798, Singapore

In this paper, we propose a method to address the problem of scale uncertainty in monocular visual odometry (VO), which includes *scale ambiguity* and *scale drift*, using distance measurements from a single ultra-wideband (UWB) anchor. A variant of Levenberg–Marquardt (LM) nonlinear least squares regression method is proposed to rectify unscaled position data from monocular odometry with 1D point-to-point distance measurements. As a loosely-coupled approach, our method is flexible in that each input block can be replaced with one's preferred choices for monocular odometry/SLAM algorithm and UWB sensor. Furthermore, we do not require the location of the UWB anchor as prior knowledge and will estimate both scale and anchor location simultaneously. However, it is noted that a good initial guess for anchor position can result in more accurate scale estimation. The performance of our method is compared with state-of-the-art on both public datasets and real-life experiments.

Keywords: Visual-based navigation; sensor fusion; localization.

1. Introduction

1.1. *Motivation*

In recent years, vision-based localization methods, such as visual odometry (VO) or simultaneous localization and mapping (SLAM), have become an integral part of robotics research. While a wide variety of sensors have been studied in various VO and SLAM systems, e.g. stereo camera, infrared (IR) and thermal cameras, laser scanner LiDAR, etc., traditional monocular camera systems are still attracting great interests from the community due to its high flexibility and easy integration with many mobile platforms. Detailed comparison of monocular, stereo and multi-camera VO pipelines [1] shows that among these approaches, monocular setup would be the most preferred solution for many real-world applications, in which size and weight constraints would restrict the kind of sensors and computational resources that can be carried by the robot [2, 3]. As illustrated in Fig. 1, a monocular camera shows clear advantages over stereo camera in the aforementioned properties.

In exchange for the flexibility and low demand on computational resources, two particular challenges need to be addressed when using a monocular VO/SLAM system. The first one is scale ambiguity, whereby from a sequence of images provided by one camera, one can only obtain the relative positions between different points in the environment that have been scaled up or down by an unknown factor. For monocular systems, scale ambiguity is inherent since depth information of 3D scene is lost when projected onto 2D frame. All estimates of camera positions and a 3D representation of the environment are therefore calculated "up to a scale". The second challenge, as a consequence, is "scale drift" [4], whereby the scales have to be adjusted in different regions of the map. In principal, initial scale estimate in monocular odometry can be corrected by adjusting the values of parameters. Even then, scale can change arbitrarily from run-to-run even with the same algorithm, thus this is not a viable solution especially for fast-deployable platform like Micro Aerial Vehicle (MAV).

Received 12 August 2019; Accepted 14 January 2020; Published 17 March 2020. This paper was recommended for publication in its revised form by editorial board member, Qin Yuan Ren.

Email Addresses: *e180071@e.ntu.edu.sg, †thienminh.nguyen@ntu.edu.sg, ‡caom0006@e.ntu.edu.sg, §elhxie@e.ntu.edu.sg

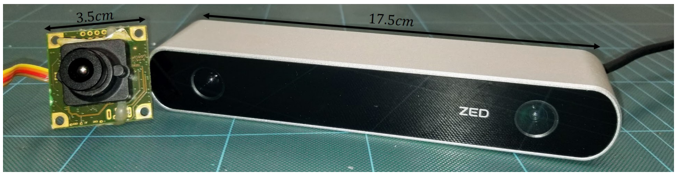


Fig. 1. Size comparison between a monocular camera (left) and a stereo camera (right) typically employed for VO/VSLAM. For small and micro-size robotic platforms, monocular camera alleviates the complexity of mechanical and software design.

Table 1. Advantages and disadvantages of different VO algorithms [1] in terms of hardware (HW) and software (SW) complexity, along with robustness and feature distance.

	Monocular	Stereo	Multi-camera
HW complexity	Low	Medium	Low
SW complexity	Medium	Low–Medium	High
Robustness	Low–Medium	Medium	High
Feature distance	High	Medium–High	High

To overcome these challenges, this work aims to leverage ultra-wideband (UWB) ranging measurement to establish a relationship between monocular VO and real-world metric-scaled distance. From that not only true metric scale estimate can be obtained, but also the scale drift would be corrected since scale correction is performed continuously along the MAV's trajectory. Among various wireless technologies such as ZigBee, Bluetooth or Wifi, UWB is chosen thanks to its properties of strong multi-path resistance and accurate ranging measurement in indoor, GPS-denied and cluttered environment [5–8].

1.2. *Related works*

1.2.1. *Metric scale correction methods*

Several methods have been proposed to recover metric scales for monocular VO in the literature. Through bundle-adjustment in the back-end task [9, 10], one can find an optimal solution for the whole trajectory, thus correcting the scales at the expense of high computational cost. However, this method only works upon detection of loop closure, thus the front-end odometry still suffers from lack of metric scale. Other approaches often introduce at least one additional sensor that provides or comes with some metric-scaled measurements to complement the monocular system. In stereo [11] or multi-camera [12] setups, since fixed baseline length is provided, depth estimates can be directly computed and metric scale can thus be obtained. The performance, however, is highly dependent on the accuracy of intrinsic and extrinsic calibrated parameters. A popular approach is to incorporate sensors that can provide range measurements such as 1D LiDAR [13], 2D LiDAR [14], 3D LiDAR [15], RGB-D camera [16], ultrasonic altimeter, radar etc., each of which effectively provides supporting distance measurements in different dimensions. By exploiting the right properties in each configuration, the scale can be accurately recovered. Considering that each sensor comes with its own limitation, the application scenario might be restricted after the fusion of odometry and range data [13, 17]. When 1D LiDAR is used, the camera is assumed to face a flat surface. Most 2D and 3D LiDAR sensors are relatively heavy and power hungry, require high computational power, which is not available on a lightweight MAV. Depth resolution and range of RGB-D camera is often limited and may only be applicable in indoors or small scale scenes. Through utilizing IMU measurements, visual-inertial odometry (VIO) methods achieve state-of-the-art performance in estimating both ego-motion and map coordinates in true scale. Many approaches [18–20] demonstrate very high accuracy and robustness in a wide range of environments. However, coupling of IMU and camera, especially in a tightly-coupled fusion scheme, often requires carefully supervised initial states and accurate multi-sensor calibration (including intrinsic and extrinsic calibration parameters, IMU biases characterization, time-synchronization between sensors), so that convergence towards inaccurate local minima [21] or even divergence can be avoided.

Other approaches to recover scale include learning the depth of environment or dimensions of objects [22–24] or applying an adaptive control strategy [25]. Depth learning-based methods [22, 23] are applicable in a wide range of scenarios, but require large amount of data and high computing power platforms for training and deployment. If a predefined target with known dimensions is exploited as *a priori* information for initialization [24], once the robot moves away from the original scene, the scale drift problem will not be addressed. An interesting solution is controlling the robot's movements to recover metric scale through observations on control gains [25], in which neither additional sensors nor high computation resources are needed. However, the robot will have to spend its limited battery on completing this task and observation is essential to know when the scale drift error becomes too large to trigger the procedure again.

1.2.2. *UWB-based and UWB-aided localization*

UWB-based localization has the capability to overcome the shortcomings of vision-based methods in reflective or featureless environments [26] while being robust to multipath and nonline-of-sight (NLOS) effects [6, 27]. However, full 3D localization using only UWB sensors requires a setup of at least four UWB anchors, which might not be practical in cluttered, dynamic environments like industrial facilities and warehouses, and also not most cost-effective setup for commercial applications. Even when an inadequate number of UWB anchors is available, other methods of localization can still employ UWB data to improve estimation accuracy, such as VO [26, 28], LiDAR-based [29] or RGB-D-based [30]. UWB distance measurements between robots in a formation can be utilized for cooperative relative localization and control problem in [31]. The use of single UWB anchor placed at an arbitrarily unknown position was studied in [32, 33] for a distance-based docking problem of MAVs without the need of visual information.

In [34], an optimization-based approach was proposed to perform scale and orientation correction of different trajectories with 1D UWB distance measurements between points on those trajectories. However, this method only considers movement in 2D case and requires complete trajectory to reach a desirable solution, thus all experiments were performed offline. In comparison, by leveraging only one UWB sensor and anchor, our method would also provide a setup that requires the least number of sensors while still be able to work in real-time, update the scale estimates continuously and estimate the anchor's position simultaneously.

1.3. *Main contributions*

In this work, we propose a loosely-coupled approach to fuse 1D point-to-point distance measurements from the MAV to a fixed UWB anchor to correct the scale problems of monocular VO. The main contributions of this work include:

- A variant of Levenberg–Marquardt (LM) method to fuse the unscaled position from monocular odometry with point-to-point distance measurements to estimate metric scale.
- The location of the static UWB anchor is not required to be known. As such, scales and anchor's position will be estimated simultaneously during the operation.
- User can choose the specific monocular odometry/SLAM algorithm and type of UWB sensor depending on one's preference, which ensures flexibility for our approach.
- In comparison to other range-based localization approaches, our method offers a simpler and more cost-effective implementation in many applications where only one anchor is possible to be setup in the environment.

1.4. *Organization of the paper*

The remaining of this paper is organized as follows: we first lay out the basic definitions, and then formulate a nonlinear least squares regression problem in Sec. 2, a modified LM Algorithm is proposed to solve the nonlinear least squares regression problem in Sec. 3. We then implement our algorithm and present our experimental results in Sec. 4. Finally, we conclude our paper in Sec. 5.

2. Problem Formulation

2.1. *Basic definitions*

In reference to Fig. 2, let us denote $\{C\}$, $\{W\}$ as the camera and VO/SLAM coordinate frames, respectively. Here, we assume that $\{W\}$ coincides with $\{C\}$ when the VO/SLAM process is initialized, i.e. the origin and orientation of the world frame are chosen as those of the first VO/SLAM calculation. This is a common practice by most VO/SLAM systems developed and reported in the literature.

At time instance k, the unscaled position $\mathbf{p}_k^S \in \mathbb{R}^3$ is obtained from the monocular VO/SLAM system. We assume that the camera and UWB sensor are attached on the same rigid body, where the translational offset between UWB antenna and camera is negligible. Therefore, the nearest associated range measurement $d_k \in \mathbb{R}^+$ from UWB anchor to UWB sensor on the robot is directly referred as distance from UWB anchor to camera. The true position of the camera and the UWB anchor in the $\{W\}$ frame are denoted as $\mathbf{p}_k = [p_k^x,\ p_k^y,\ p_k^z]^\top \in \mathbb{R}^3$ and $\mathbf{p}_a = [p_a^x,\ p_a^y,\ p_a^z]^\top \in \mathbb{R}^3$, respectively. A block diagram of the proposed system is

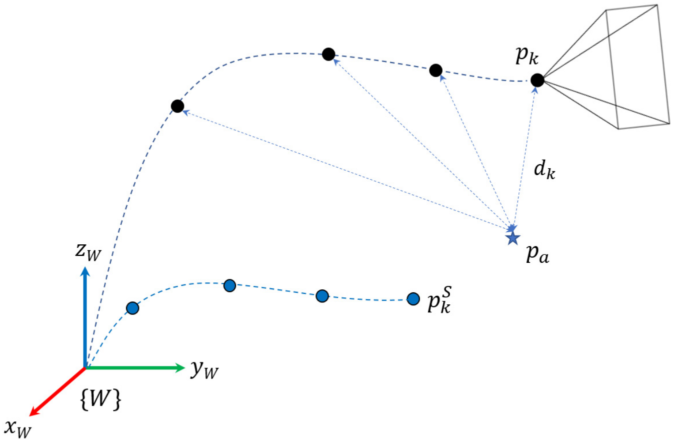


Fig. 2. System overview: Over a sliding window of recent position and range data, the proposed system will perform a continuous optimization to estimate the scales and anchor position. $\{W\}$ is the VO/SLAM coordinate frame. $\mathbf{p}_k^S$ is the unscaled position data from monocular VO/SLAM system, $\mathbf{p}_k$ is the true camera position, and $\mathbf{p}_a$ is the unknown anchor position. d_k is ranging measurement obtained from UWB sensors.

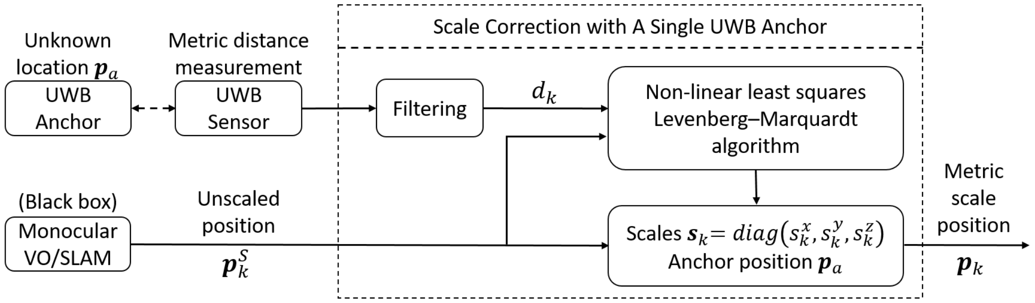

Unknown location $\boldsymbol{p}_a$
UWB Anchor
Metric distance measurement
UWB Sensor
Scale Correction with A Single UWB Anchor
Filtering
d_k
Non-linear least squares Levenberg–Marquardt algorithm
(Black box)
Monocular VO/SLAM
Unscaled position
$\boldsymbol{p}_k^S$
Scales $\boldsymbol{s}_k = diag(s_k^x, s_k^y, s_k^z)$
Anchor position $\boldsymbol{p}_a$
Metric scale position
$\boldsymbol{p}_k$

Fig. 3. Block diagram of our method. ORB-SLAM2 (Mono) and Humatics' P440 UWB sensors were used for experiments. A standard 1D median filter with an order of 5 is employed to reject noise and smooth out the raw data of the UWB. The median filter is chosen for its robustness against outliers.

illustrated in Fig. 3. The monocular VO/SLAM is regarded as "black box", i.e. only its position output is used as input for our system.

To address the issues related to the scale in VO/VSLAM, we propose to find the scales $\boldsymbol{s}_k = \mathrm{diag}(s_k^x, s_k^y, s_k^z)$ such that $\mathbf{p}_k$ can be recovered by applying the formula

$$\mathbf{p}_k = |\,\boldsymbol{s}_k\,|\,\mathbf{p}_k^S. \tag{1}$$

Note that the absolute operator is applied element-wise and is used as a way to keep sign of scales positive without employing any explicit bound constraints. This will be further explained in Sec. 3.4.

The range measurement d_k is obtained by multiplication of light speed c and time of flight, which is measured from the time stamps T_{M1}^{Rx} and T_{M0}^{Tx} of when the UWB ranging signal is sent and received. Taking into account the processing time delay σ_k of the UWB sensors, we have

$$\begin{aligned} d_k &= c\frac{T_{M1}^{Rx} - T_{M0}^{Tx} - \sigma_k}{2} + \eta_k \\ &= \|\mathbf{p}_a - \mathbf{p}_k\| + \eta_k \\ &= \|\mathbf{p}_a - |\,\boldsymbol{s}_k|\mathbf{p}_k^S\| + \eta_k, \end{aligned} \tag{2}$$

where $\| * \|$ denotes Euclidean norm of the argument vector, and $\eta_k \sim \mathcal{N}(0, \Omega_k)$ is the assumed zero mean Gaussian noise [26].

2.2. *Nonlinear least squares regression*

We propose a method based on continuous optimization over a sliding window $\mathcal{N}_m$ composed of odometry positions and range measurements. Given a set of m samples $\mathcal{N}_m = \{(\mathbf{p}_i^S \top, d_i)\}_{i=k-m}^{i=k}$, our aim is to find the vector of parameters

$$\boldsymbol{\beta}_k = [s_k^x, s_k^y, s_k^z, p_{ak}^x, p_{ak}^y, p_{ak}^z]^\top, \tag{3}$$

where k is the time index of the latest data added to the window that minimizes the sum of squares of range errors

$$E_k^r = \sum_{i=k-m}^{k} r_i{}^2(\boldsymbol{\beta}_k), \tag{4}$$

where the residuals r_i are defined as

$$r_i = d_i{}^2 - \|\mathbf{p}_{ak} - |\,\boldsymbol{s}_k|\,\mathbf{p}_i^S\|^2 = y_i{}^2 - f(\boldsymbol{\beta}_k)^2. \tag{5}$$

Wit the cost function (4), the optimal values for $\mathbf{p}_{ak}$ and $\boldsymbol{s}_k$ can be obtained through the minimization of

$$\boldsymbol{\beta}_k^* = \arg\min_{\boldsymbol{\beta}_k} E_k^r. \tag{6}$$

The optimization algorithm is described in detail in Sec. 3.

Remark 2.1. The following formulas for the residuals r_i have been validated over the same datasets:

$$\begin{aligned} r_i &= d_i - \|\mathbf{p}_{ak} - |\boldsymbol{s}_k|\,\mathbf{p}_i^S\|, \\ r_i &= d_i{}^2 - \|\mathbf{p}_{ak} - |\boldsymbol{s}_k|\,\mathbf{p}_i^S\|^2, \end{aligned}$$

none of which showed significant and consistent improvements compare to the other. However, the calculation of the Jacobian matrix from the former involves a square root in the denominator, which might lead to the division by zero issue. Thus, the latter was selected for this work.

Remark 2.2. When no prior information for the anchor position is available, in [34], an optimization scheme is outlined where one can obtain the scale of the trajectory and relative anchor position. On the other hand, this method requires that a trajectory which contains diverse points in the space is available, which renders it unsuitable as an online solution but adequate to obtain initial estimates. Therefore one can apply this method for the first run, then utilize the outputs as initial guesses for the proposed approach in the subsequent runs.

2.3. *Existence of solution*

For UWB sensors that is based on the time-of-flight technology, which is the type of sensor used in our experiments, the measured distance d_k will be the shortest when there is direct point-to-point path between sensors. Taking multipath and nonline-of-sight effects into account, the actual measurements would tend to increase compare to ideal cases due to propagation delay. From this insight, at time instance k, we have the following inequality:

$$\begin{aligned} d_k &\geq \|\mathbf{p}_{ak} - \boldsymbol{s}_k \mathbf{p}_k^S\| \\ &\geq \left\| \begin{bmatrix} p_{ak}^x \\ p_{ak}^y \\ p_{ak}^z \end{bmatrix} - \begin{bmatrix} s_k^x & 0 & 0 \\ 0 & s_k^y & 0 \\ 0 & 0 & s_k^z \end{bmatrix} \begin{bmatrix} p_k^{Sx} \\ p_k^{Sy} \\ p_k^{Sz} \end{bmatrix} \right\| \\ &\geq \left\| \begin{bmatrix} -p_k^{Sx} & 0 & 0 & 1 & 0 & 0 \\ 0 & -p_k^{Sy} & 0 & 0 & 1 & 0 \\ 0 & 0 & -p_k^{Sz} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_k^x \\ s_k^y \\ s_k^z \\ p_{ak}^x \\ p_{ak}^y \\ p_{ak}^z \end{bmatrix} \right\| \\ &\geq \|A_k \boldsymbol{\beta}_k\|, \end{aligned} \tag{7}$$

which leads to

$$\boldsymbol{\beta}_k{}^\top A_k{}^\top A_k \boldsymbol{\beta}_k \leq d_k{}^2. \tag{8}$$

Since $A_k{}^\top A_k$ is positive semi-definite, the problem can be viewed as finding a solution within the intersection of convex regions created by each range measurement constraint. It is clear that with d_k being positive, a feasible solution can always be found since the point $\boldsymbol{\beta}_k = 0$ is contained in each of the convex regions. In [34], it is shown that the optimization problem of similar structure will have a convex hull defined by the convex constraints, and the global optima must lie on the convex hull. Thus, with sufficient sliding window size and step size, a globally optimal result can be found for the proposed problem.

3. Main Algorithm

In this section, we will first present the basic construction of a LM solution to the optimization problem in (6). After which, modification techniques will be proposed to improve the solution.

3.1. *Recursive LM algorithm*

In this section we describe our variant of LM algorithm [35] to solve (6). At time instance k, starting with an initial guess $\boldsymbol{\beta}_k^{(0)}$, the parameter vector $\boldsymbol{\beta}_k$ is refined after each iteration l by a shift vector $\Delta\boldsymbol{\beta}$ and learning rate $\boldsymbol{\gamma}$:

$$\boldsymbol{\beta}_k^{(l+1)} = \boldsymbol{\beta}_k^{(l)} + \boldsymbol{\gamma}\Delta\boldsymbol{\beta}, \tag{9}$$

where $\Delta\boldsymbol{\beta}$ can be found via solving the equation:

$$(\mathbf{J}^\top\mathbf{J} + \boldsymbol{\lambda}\mathbf{I})\Delta\boldsymbol{\beta} = \mathbf{J}^\top[\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}_{\boldsymbol{k}})]. \tag{10}$$

In the above, $\mathbf{y}$ and $\mathbf{f}(\boldsymbol{\beta}_{\boldsymbol{k}})$ are vectors with ith component d_i and $\|\mathbf{p}_{ak} - |\boldsymbol{s}_k|\, \mathbf{p}_i^S\|^2$, respectively, $\mathbf{I}$ is the identity matrix, $\mathbf{J} = [J_{ij}]$ with $\mathbf{J}_{ij} = -\partial r_i / \partial \beta_j$ is the Jacobian matrix and $\boldsymbol{\lambda}$ is a nonnegative damping *matrix*, adjusted at each iteration l. The proposed method is summarized in Algorithm 3.1, with details of our modifications as follows:

Algorithm 3.1. Scale correction in VO with UWB anchor

```
Parameters: $\mathbf{p}_{ak}^0, \boldsymbol{s}_0, \gamma_0, \tau, \zeta, m, L, \rho, \lambda_0, \nu, \epsilon$.
Initialization: $\boldsymbol{\beta}_0 = (\boldsymbol{s}_0, \mathbf{p}_{ak}^0)$. $k = 0$. $\mathcal{N}_m$ is initialized with first odometry and range data $(\mathbf{p}_0^S, d_0)$ as all entries.
Input: odometry data $\tilde{\mathbf{p}}^S$, range data $\tilde{d}$.
if $\big\| |\boldsymbol{s}_k|\, \tilde{\mathbf{p}}^S - |\boldsymbol{s}_k|\, \mathbf{p}_m^S \big\| > \rho$ then
    Update sliding window: Remove first data entry in $\mathcal{N}_m$, add $(\tilde{\mathbf{p}}^S, \tilde{d})$ as last entry.
    $l := 1$
    do
        Update $r_i$ in (5) and the Jacobian.
        Calculate shift vector $\boldsymbol{\Delta\beta}$ with (10).
        $\boldsymbol{\beta}_k := \boldsymbol{\beta}_k + \boldsymbol{\gamma}\boldsymbol{\Delta\beta}$.
        for $i := 1$ to 6 step 1 do
            if $\left|\frac{\Delta\beta_j}{\beta_j}\right| < \epsilon$ then
                $\boldsymbol{\lambda}_{jj} := \boldsymbol{\lambda}_{jj}/\nu$
            else
                $\boldsymbol{\lambda}_{jj} := \boldsymbol{\lambda}_{jj} * \nu$
            end if
        end for
        Update $E_k^{r\,(l)}$
        $l := l + 1$
    while $l \leq L$ and $\left|(E_k^{r\,(l-1)} - E_k^{r\,(l)})/E_k^{r\,(l-1)}\right| < \zeta$
end if
$\boldsymbol{\beta}_{k+1} := \boldsymbol{\beta}_k$
$k := k + 1$
Output: $\mathbf{p}_k := |\boldsymbol{s}_k|\, \tilde{\mathbf{p}}^S$
```

3.2. *Damping*

Various approaches have been put forward to efficiently adjust damping term λ in the original LM algorithm. By changing the value of λ from large to small, one can modify the algorithm's behavior between the gradient descent and Gauss–Newton methods. However, in real-time application when data arrives one by one in a stream, the received data up until the current time might not provide enough information to perform optimization over parameters of *all* axes. For example, if at first the camera only moves along x axis in $\{W\}$, we only have new data corresponds to x-axis but λ gets updated as a damping term for *all* axes. Consequently, new algorithm's behavior will be applied to y- and z- axis as well. In such scenarios, the algorithm might perform well on one axis but yield unsatisfactory results on others. To overcome this, we propose the following two modifications to adjust damping term λ for individual parameters:

(1) Change λ to damping *matrix*

$$\boldsymbol{\lambda} = \mathrm{diag}(\lambda_{s_k^x}, \lambda_{s_k^y}, \lambda_{s_k^z}, \lambda_{p_a^x}, \lambda_{p_a^y}, \lambda_{p_a^z}), \tag{11}$$

which consists of six damping terms corresponding to six parameters defined in (3). Each $\boldsymbol{\lambda}_{jj}$ is initialized with large value λ_0 so that updates are carried out with small steps in steepest-descent direction at the beginning.

(2) At the end of every iteration l, each diagonal entry $\boldsymbol{\lambda}_{jj}$ is decreased by a factor $\nu > 1$ if

$$\left|\frac{\Delta\beta_j}{\beta_j}\right| < \epsilon, \quad j = 1, \ldots, 6, \tag{12}$$

otherwise $\boldsymbol{\lambda}_{jj}$ is increased by ν. Here ϵ is the level of refinement, e.g. $\epsilon = 0.001$ is equivalent to specifying 0.1% precision refinement.

Remark 3.1. It has been suggested that for many optimization problems, the identity matrix $\mathbf{I}$ can be changed to a diagonal matrix consisting of the diagonal elements of $\mathbf{J}^\top\mathbf{J}$ which represent the relative scaling of the parameters, thus making the solution *scale invariant*, that is if the parameters were to be replaced by a factor of μ, i.e. $\vec{\boldsymbol{\beta}}_k = \mu\boldsymbol{\beta}_k$, the cost function values would not change. However, this behavior is counter productive in our problem since only one set of values for $\boldsymbol{s}$ and $\mathbf{p_a}$ should be the right solution. Thus, in this work, the identity matrix remains unchanged.

3.3. *Dynamic learning rate*

The time-varying learning rate

$$\boldsymbol{\gamma} = \mathrm{diag}(1, 1, 1, \gamma(k), \gamma(k), \gamma(k)), \tag{13}$$

only applies to anchor's position $\mathbf{p}_{ak}$ to incorporate the knowledge that anchor's position is fixed. Starting with an initial value γ_0, $\gamma(k)$ decreases gradually as the number of run times k increases

$$\gamma(k) = \gamma_0 e^{-(k/\tau)}, \tag{14}$$

where τ is a time constant. Initially, $\mathbf{p}_{ak}$ should be estimated with large gradient and as $\mathbf{p}_{ak}$ becomes more precisely determined, only fine tuning with smaller and smaller gradient is required. Thus, the estimate of $\mathbf{p}_a$ will benefit from a good initial guess $\mathbf{p}_a^0$ since the trajectory might not encapsulate movement along all axes.

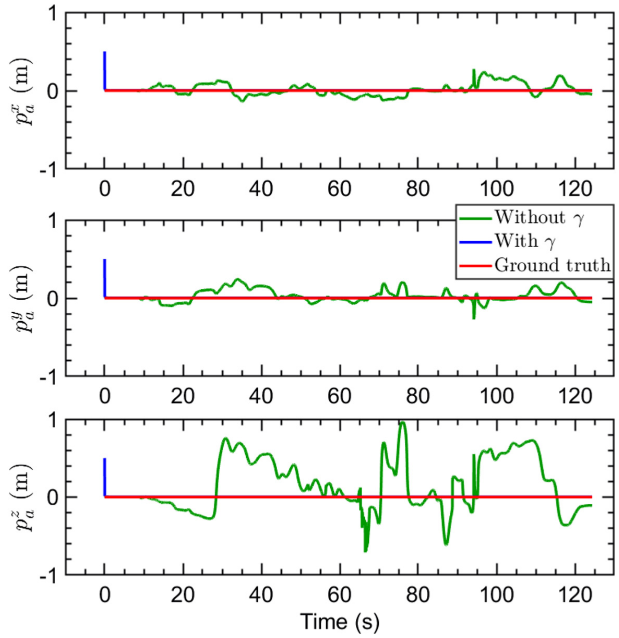


Fig. 4. Effectiveness of $\boldsymbol{\gamma}$ in fixing the anchor position when the same initial guess of $(0.5, 0.5, 0.5)$ are applied. Without $\boldsymbol{\gamma}$, $\mathbf{p}_{ak}$ will be estimated continuously. When $\boldsymbol{\gamma}$ is used, $\mathbf{p}_{ak}$ will stay at a final value after a certain period. The experiments are performed on MH_03 dataset, as described in Sec. 4.2.

As depicted in Fig. 4, when $\boldsymbol{\gamma}$ is employed, the values of $\mathbf{p}_a$ will converge to true position after a number of run times and stay at that position afterwards, hence improving the accuracy of scale estimation.

3.4. *Sign-bounded cost function*

All distance measurements would be unchanged if the trajectory and the position of the anchor are mirrored on one plane or axis of $\{W\}$. In such cases, signs of scale estimates $\boldsymbol{s}$ can be flipped but the cost function value will still be the same, resulting in local minimas. To address this issue, one can keep the sign of $\boldsymbol{s}$ positive or fix the position of anchor as known parameters, with the former being the only viable option as the anchor's position is unknown. Since the theory of LM algorithm does not define a way to handle explicit bound constraints, we formulate the relationship between scale-corrected position and odometry output as (1), so that final position output never changes sign.

Remark 3.2. It should be noted that in order to keep $\boldsymbol{s}$ positive, another approach could be to use the square of $\boldsymbol{s}_k$, that is $\mathbf{p}_k = \boldsymbol{s}_k{}^2\mathbf{p}_k^S$. However, this technique makes the gradient zero whenever $\boldsymbol{s}_k$ becomes zero, which might happen when position data goes near zero, and the iterations will not move from that point.

3.5. *Start and stop conditions*

The data stored in the sliding window might not contain statistically diverse information that contributes to the optimization. For instance, if the robot stops intermittently along the trajectory, the sliding window will eventually contain similar data points (pose and distance measurements) and the estimation would not improve no matter what optimization scheme is used. To maintain a spatial spread for the sliding window, a condition is checked when new odometry data $\tilde{\mathbf{p}}^S$ arrives. Only if the new position data is outside a pre-defined diameter ρ from the last point in sliding window $\mathcal{N}_m$, will it be added into the window

$$\| |\boldsymbol{s_k}| \tilde{\mathbf{p}}^S - |\boldsymbol{s_k}| \mathbf{p}^S_m \| > \rho. \tag{15}$$

Otherwise, $\tilde{\mathbf{p}}^S$ is discarded and the previous estimations are carried on until the next data is received. Once started, the algorithm iteratively repeats the steps as described in Algorithm 3.1 until the number of iterations l has exceeded a pre-defined limit L or convergence criterion is met, which is defined as

$$\left| \frac{E^r_k(l-1) - E^r_k(l)}{E^r_k(l-1)} \right| < \zeta, \tag{16}$$

where $E^r_k(l)$ is the sum of squares errors at data sample k and iteration l, ζ is a numerical constant, i.e. $\zeta = 0.0001$.

4. Experimental Results

In this section, we present experimental results on EuRoC datasets [36] and real-life experiments, using ORB-SLAM2 [16] as the monocular VO module and various state-of-the-art methods for comparison.

4.1. *Evaluation method*

On each dataset and experiment, scales are estimated with the proposed method and scale-corrected position is obtained by (1). Scale-corrected and ground truth trajectories are aligned by applying the method in [37], then absolute translation error (RMSE) is calculated over the aligned trajectories. Position error is also validated using ATE and RPE [38]. Let $\Delta x_i, \Delta y_i, \Delta z_i$ be the difference between estimated and ground truth position on x-, y- and z-axis, respectively. Translation error on each axis and the absolute translation error is

$$\mathrm{RMSE}_x = \left(\frac{1}{N} \sum_{i=0}^{N-1} \|\Delta x_i\| \right)^{1/2},$$

$$\mathrm{RMSE}_y = \left(\frac{1}{N} \sum_{i=0}^{N-1} \|\Delta y_i\| \right)^{1/2},$$

$$\mathrm{RMSE}_z = \left(\frac{1}{N} \sum_{i=0}^{N-1} \|\Delta z_i\| \right)^{1/2},$$

$$\mathrm{RMSE}_{\mathrm{pos}} = \left[\frac{1}{3} \left(\mathrm{RMSE}_x{}^2 + \mathrm{RMSE}_y{}^2 + \mathrm{RMSE}_z{}^2 \right) \right]^{1/2}.$$

Scale estimates are compared with ground truth scale to demonstrate our method's capability to track scale. At time k, given unscaled positions from monocular odometry $\{\mathbf{p}^S_i\}^k_{i=1}$ and ground truth positions $\{\mathbf{p}^W_i\}^k_{i=1}$, ground truth scale $\boldsymbol{s}^W_k$ is calculated through Umeyama alignment [37] on each axis, using all data available up to time k. Umeyama alignment is a quick and simple algorithm to estimate scale $\boldsymbol{s}^W_k$, rotation $\mathbf{R}_k$ and translation t_k of a set of points to match corresponding points by finding the similarity transformation $S = \{\boldsymbol{s}^W_k, \mathbf{R}_k, t_k\}$ that satisfies

$$S = \underset{\{\boldsymbol{s}^W_k, \mathbf{R}_k, t_k\}}{\arg\min} \sum_{i=1}^{k} \|\mathbf{p}^W_i - (\boldsymbol{s}^W_k \mathbf{R}_k \mathbf{p}^S_i + t_k)\|^2. \tag{17}$$

The scale error e^s_k is then computed as L^1-norm between $|\boldsymbol{s}_k|$ and true scale $\boldsymbol{s}^W_k$, which is

$$e^s_k = \||\boldsymbol{s}_k| - \boldsymbol{s}^W_k\|_1. \tag{18}$$

4.2. *Public datasets*

Experiments were performed on a laptop with Intel Core i7-8750H CPU. Since 1D UWB distance measurement is not available in any of the datasets, it is calculated from ground truth positions in the aligned trajectory, assuming the anchor is located at the origin. Initial values for anchor's position and scales are fixed at $\mathbf{p}^0_{ak} = (0.5, 0.5, 0.5)$ and $\boldsymbol{s}_0 = (1, 1, 1)$, respectively, to demonstrate the performance for the case where we have a rough guess for the anchor position, but no prior knowledge of how well-scaled the input odometry is. The algorithm generally performs better with increasing size of sliding window, but in our evaluations, we fixed the size of sliding window to 500 data points to maintain consistency between experiments. The other parameters were tuned to achieve the best results on each dataset. Figures 5 and 6 are examples of the final trajectory and position output for MH_03 dataset. Most datasets start with a period when the MAV is static, thus scale estimates remain unchanged from the beginning until there are initial movements. Figure 7 illustrates the scale estimates in MH_03 dataset. Scale estimates only change from 10 s onwards when the MAV starts moving, quickly approaches true values, oscillates for a short period before stabilizing with little change thereafter. Since movements in the

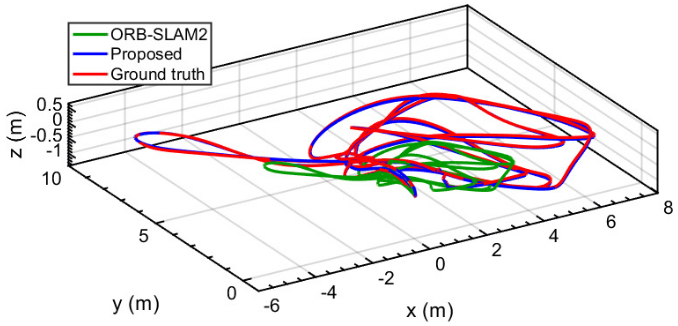


Fig. 5. 3D trajectory view in MH_03 dataset: ORB-SLAM2 Mono (green), scale-corrected (blue) and ground truth (red).

dataset starts at around $t = 10\,\text{s}$, scale estimates moves from initial guesses at this time as well. All scale estimates quick converges to their true value. In this dataset, ground truth scales retain the value to the end of the experiment. In general, this might not be the case and scale value might change depending on the depth of the scene. Figure 8 shows the value of scale error e_k^s calculated over all datasets, with final values in brackets. All datasets begin with the MAV in static position and e_k^s remain constant, followed by initial movements where e_k^s reduces sharply. Eventually, e_k^s approaches zero in all datasets, demonstrating that the proposed method can estimate the correct metric scale over time.

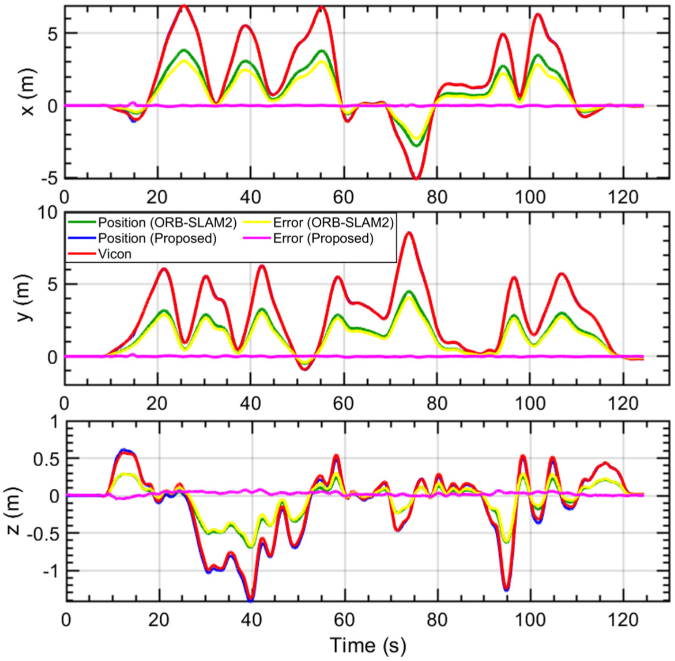


Fig. 6. Position output and error in MH_03 dataset: ORB-SLAM2 Mono (green for position, yellow for error), scale-corrected (blue for position, magenta for error) and ground truth position (red). It can be seen that the position estimate converges to and follows ground truth quite closely after corrected by our method.

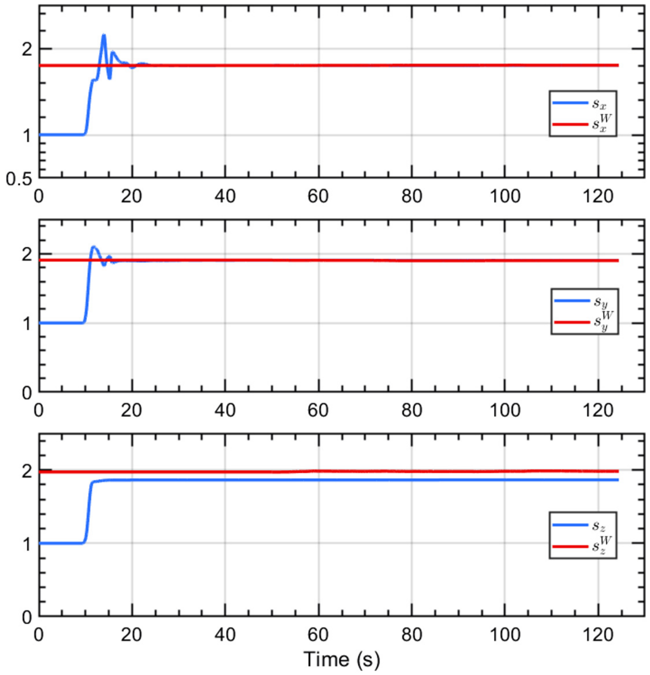


Fig. 7. Estimated scale (s_x, s_y and s_z) and ground truth scale (s_x^W, s_y^W and s_z^W) on each axis in MH_03 dataset. Since the range of motion in the z-axis is much smaller than the other axes, as can be seen in Fig. 5 to be less than 1.5 m compared to 5–8 m in x and y, the position error in the z-axis was also of small value. Hence the scale estimate for z-axis was not updated over most of the course of the experiment. This would not occur if the range of motion in z-axis was of higher magnitude or a larger sliding window is employed at the cost of higher computation time.

After alignment with ground truth, RMSEs are calculated and results are reported in Table 2. We compare RMSE results of the proposed method with state-of-the-art VIO methods, extracted from [39]. In all datasets, the proposed

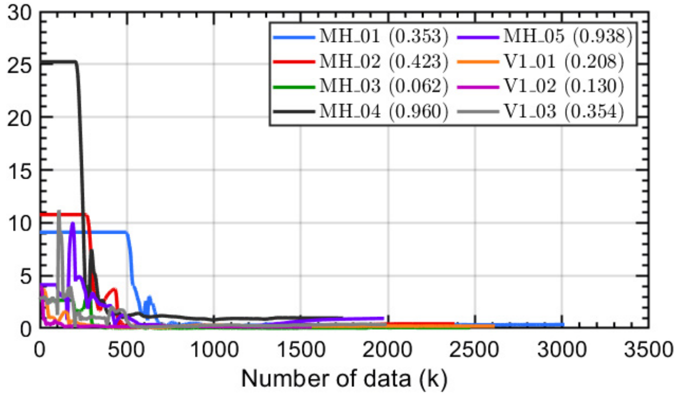


Fig. 8. Scale errors e_k^s in experiments with EuRoC datasets. Final error values are shown in brackets.

Table 2. EuRoC datasets: Comparison of translational errors (RMSE) in meter. The best results are highlighted in **bold**. Our algorithm can achieve the same level of accuracy in most datasets, with an exception of MH_05 dataset where the ground truth data is partially occluded, thus affecting both our estimation and RMSE calculation.

	RMSE (m)			
Sequence	Proposed	OKVIS	ROVIO	VINS-Mono
MH_01	0.21	**0.16**	0.21	0.27
MH_02	0.20	0.22	0.25	**0.12**
MH_03	**0.13**	0.24	0.25	**0.13**
MH_04	0.40	0.34	0.49	**0.23**
MH_05	0.72	0.47	0.52	**0.35**
V1_01	0.10	0.09	0.10	**0.07**
V1_02	0.11	0.20	**0.10**	**0.10**
V1_03	**0.13**	0.24	0.14	**0.13**

method can provide accurate scale-corrected position and the position of the anchor simultaneously. Furthermore, the proposed method greatly alleviate the mechanical complexity and computational resources for any robotic platform that already employs UWB sensor for relative localization and docking tasks.

4.3. *Real-life experiments*

The proposed system was further validated with real-time experiments, with the results summarized in Table 3. Our setup consists of a MYNT EYE S sensor,[a] a stereo camera with IMU synchronization, UWB ranging sensor and anchor are Humantics' P440[b] and an Intel NUC i7[c] onboard computer mounted on a hexacopter frame (Fig. 9). The system was implemented in Robot Operating System[d] (ROS), the most widely used open-source robotics software framework. Ground truth was recorded with a Vicon[e] system. Each experiment has different trajectories but the same anchor position relative to take-off point in world coordinate frame. Only images from left camera were used for

Table 3. Real-life experiments: Comparison of translational error (RMSE_{pos}), absolute trajectory error (ATE) and relative pose error (RPE) between the proposed approach and ORB-SLAM2 Stereo (ORB-S for short). The best results are highlighted in **bold**. Experimental results show that our method can achieve the same level of accuracy as state-of-the-art method.

	RMSE_{pos} (m)		ATE (m)		RPE (m)	
Sequence	Ours	ORB-S	Ours	ORB-S	Ours	ORB-S
SC01	0.15	**0.10**	0.15	**0.14**	**0.21**	0.22
SC02	**0.14**	0.16	0.22	**0.16**	0.32	**0.25**
SC03	**0.06**	0.12	**0.10**	0.20	**0.20**	0.32
SC04	**0.09**	0.12	**0.08**	0.16	**0.14**	0.21
SC05	0.15	**0.13**	**0.14**	0.17	**0.22**	0.26

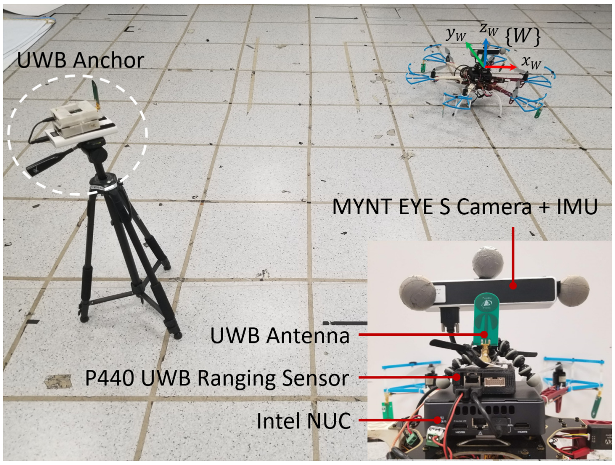


Fig. 9. Experimental setup consists of one hexacopter platform equipped with a stereo camera and UWB ranging sensor.

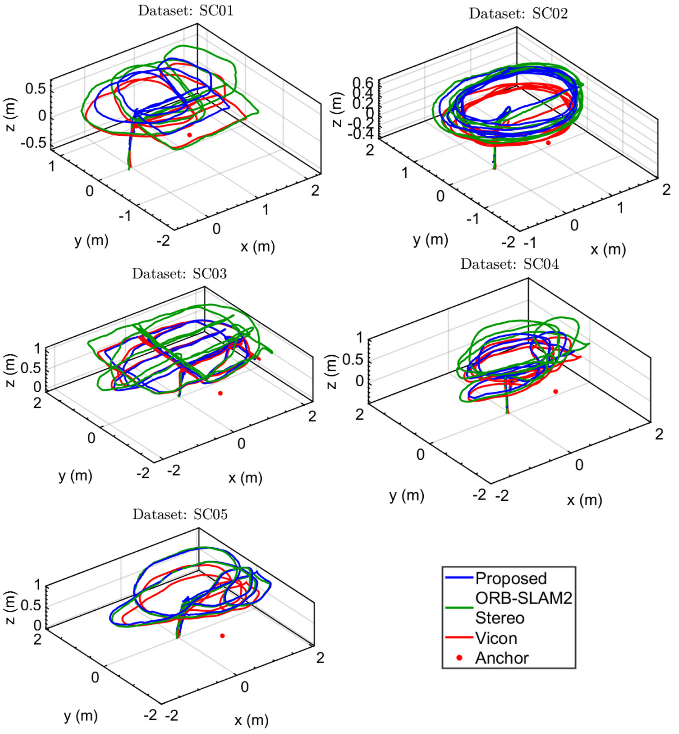


Fig. 10. 3D trajectory results in our real-life experiments.

[a] https://www.mynteye.com/products/mynt-eye-stereo-camera.
[b] https://www.humatics.com/products/scholar/.
[c] https://simplynuc.com/7i7dnke-kit/.

[d] https://www.ros.org/
[e] https://www.vicon.com/

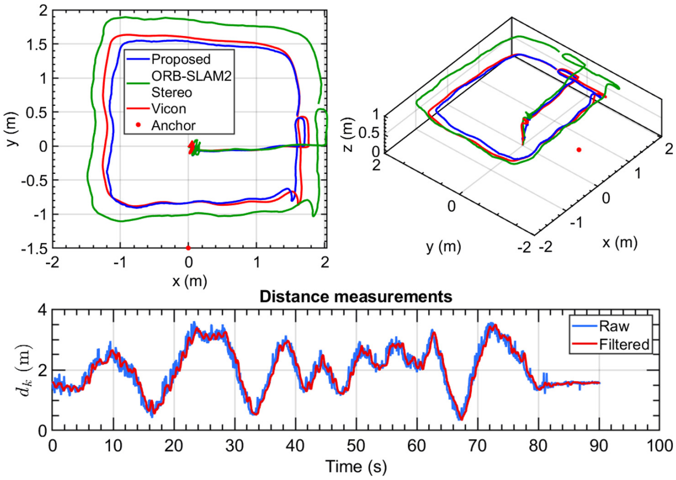


Fig. 11. 3D trajectory view (top) and UWB distance measurements (bottom) in SC03 experiment. For illustration purpose, only the first 30 s of the trajectory is showed here, while the complete trajectory can be viewed in Fig. 10. A smooth filter is applied on raw UWB distance measurements before being fused in the proposed scheme. It can be clearly seen that the loosely-couple method can effectively correct ORB-SLAM2 Mono data to metric scale.

ORB-SLAM2 Mono (without loop closure) and UWB measurements were processed to reject potential outliers and a smoothing filter was applied to the data before fusion. For comparison, we recorded the data and ran against ORB-SLAM2 Stereo. Initial guess for scale is again identity $\boldsymbol{s}_0 = I$ and other parameters were tuned selectively. An overview of all trajectories in the experiments is depicted in Fig. 10 and a closer look on one experiment can be viewed in Figs. 11 and Fig. 12. Video of our real-time experiments can be found at: https://youtu.be/pShkJHs1ZAc.

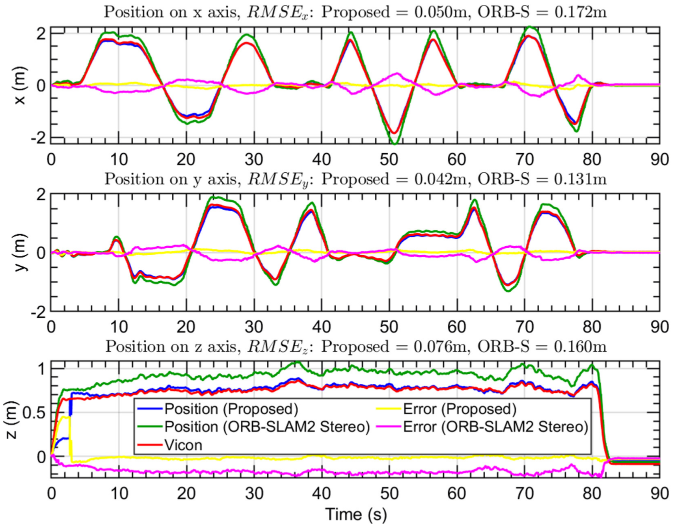


Fig. 12. The scale-corrected positions on each axis in SC03 experiment. Sliding windows for real-time experiments contain 100 data points which corresponds to 3.3 seconds of data. At $t = 3.3$ s, the first batch of data is collected, the proposed method starts and scale is corrected for the first time, which results in a sudden jump in odometry, most evident in the z-axis.

With the image stream running at 60 fps and UWB data coming at 40 Hz, the proposed method can process data at 50 Hz while ORB-SLAM2 Stereo runs at 20 Hz. The advantages of the proposed method over ORB-SLAM2 Stereo are twofold. First, by only using a monocular camera, the mechanical complexity of hardware design would be greatly alleviated. Second, there is a significant improvement in processing speed compared to current state-of-the-art method.

5. Conclusions

In this work, a loosely-coupled fusion scheme that utilizes both unscaled position data from monocular VO and UWB ranging measurements to perform estimation of metric scale and anchor position simultaneously, while continuously correct the scale drift is proposed. As a loosely-coupled approach, each input module can be replaced with user's preferences. Moreover, our method does not require that the exact location of the UWB anchor position be known. The system was validated with public datasets and real experiments. Results showed that we can achieve accuracy that is on par with state-of-the-art stereo VO method.

However, the proposed method suffers from the inaccuracy of the monocular odometry and would benefit from good approximation of anchor location in scenarios where movements along certain axis are limited. For future works, translational offsets between UWB sensor and camera would also be taken into account, which would not only generalize the system better, but also allow for more flexibility in mechanical setup on a variety of robotic platforms. Furthermore, we plan to further investigate the incorporation of UWB measurements by employing a tightly-coupled fusion approach.

References

[1] K. Mohta, M. Watterson, Y. Mulgaonkar, S. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico *et al.*, Fast, autonomous flight in gps-denied and cluttered environments, *J. Field Robot.* **35**(1) (2018) 101–120.

[2] S. Saeedi, C. Thibault, M. Trentini and H. Li, 3d mapping for autonomous quadrotor aircraft, *Unmanned Syst.* **5**(3) (2017) 181–196.

[3] C. Liu, S. D. Prior and J. P. Scanlan, Design and implementation of a low cost mini quadrotor for vision based maneuvers in gps denied environments, *Unmanned Syst.* **4**(3) (2016) 185–196.

[4] H. Strasdat, J. Montiel and A. J. Davison, Scale drift-aware large scale monocular slam, *Robot. Sci. Syst. VI* **2** (2010).

[5] K. Guo, Z. Qiu, C. Miao, A. H. Zaini, C.-L. Chen, W. Meng and L. Xie, Ultra-wideband-based localization for quadcopter navigation, *Unmanned Syst.* **4**(1) (2016) 23–34.

[6] T. M. Nguyen, A. H. Zaini, K. Guo and L. Xie, An ultra-wideband-based multi-uav localization system in gps-denied environments, *2016 Int. Micro Air Vehicles Conf.* (2016), pp. 56–61. http://www.imavs.org/tag/imav2016/#paperkey_43

[7] T.-M. Nguyen, A. H. Zaini, C. Wang, K. Guo and L. Xie, Robust target-relative localization with ultra-wideband ranging and communication, *2018 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, 2018), pp. 2312–2319.

[8] T.-M. Nguyen, T. H. Nguyen, M. Cao, Z. Qiu and L. Xie, Integrated uwb-vision approach for autonomous docking of uavs in gps-denied environments, *2019 Int. Conf. Robotics and Automation (ICRA)* (IEEE, 2019), pp. 9603–9609.

[9] M. J. M. M. Mur-Artal, Raúl and J. D. Tardós, ORB-SLAM: A versatile and accurate monocular SLAM system, *IEEE Trans. Robot.* **31**(5) (2015) 1147–1163.

[10] G. Dubbelman and B. Browning, Cop-slam: Closed-form online pose-chain optimization for visual slam, *IEEE Trans. Robot.* **31**(5) (2015) 1194–1213.

[11] J. Engel, J. Stückler and D. Cremers, Large-scale direct slam with stereo cameras, *2015 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)* (IEEE, 2015), pp. 1935–1942.

[12] P. Liu, M. Geppert, L. Heng, T. Sattler, A. Geiger and M. Pollefeys, Towards robust visual odometry with a multi-camera system, *2018 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 1154–1161.

[13] Z. Zhang, R. Zhao, E. Liu, K. Yan and Y. Ma, Scale estimation and correction of the monocular simultaneous localization and mapping (slam) based on fusion of 1d laser range finder and vision data, *Sensors* **18**(6) (2018) 1948.

[14] Q. Lv, J. Ma, G. Wang and H. Lin, Absolute scale estimation of orb-slam algorithm based on laser ranging, *2016 35th Chinese Control Conf. (CCC)* (IEEE, 2016), pp. 10279–10283.

[15] T. Caselitz, B. Steder, M. Ruhnke and W. Burgard, Monocular camera localization in 3d lidar maps, *2016 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2016), pp. 1926–1931.

[16] R. Mur-Artal and J. D. Tardós, ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras, *IEEE Trans. Robot.* **33**(5) (2017) 1255–1262.

[17] R. Giubilato, S. Chiodini, M. Pertile and S. Debei, Scale correct monocular visual odometry using a lidar altimeter, *2018 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 3694–3700.

[18] T. Qin, P. Li and S. Shen, Vins-mono: A robust and versatile monocular visual-inertial state estimator, *IEEE Trans. Robot.* **34**(4) (2018) 1004–1020.

[19] S. Lynen, T. Sattler, M. Bosse, J. A. Hesch, M. Pollefeys and R. Siegwart, Get out of my lab: Large-scale, real-time visual-inertial localization, *Proceedings of Robotics: Science and Systems* (Rome, Italy, 2015). http://www.roboticsproceedings.org/rss11/p37.html.

[20] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart and P. Furgale, Keyframe-based visual–inertial odometry using nonlinear optimization, *Int. J. Robot. Res.* **34**(3) (2015) 314–334.

[21] J. Kaiser, A. Martinelli, F. Fontana and D. Scaramuzza, Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation, *IEEE Robot. Autom. Lett.* **2**(1) (2017) 18–25.

[22] K. Tateno, F. Tombari, I. Laina and N. Navab, Cnn-slam: Real-time dense monocular slam with learned depth prediction, in *Proc. IEEE Conf. Computer Vision Pattern Recognit. (CVPR)*, Vol. 2 (IEEE, 2017), pp. 6565–6574.

[23] S. Wang, R. Clark, H. Wen and N. Trigoni, Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks, *2017 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, 2017), pp. 2043–2050.

[24] D. Eberli, D. Scaramuzza, S. Weiss and R. Siegwart, Vision based position control for mavs using one single circular landmark, *J. Intell. Robot. Syst.* **61**(1–4) (2011) 495–512.

[25] S. H. Lee and G. de Croon, Stability-based scale estimation for monocular slam, *IEEE Robot. Autom. Lett.* **3**(2) (2018) 780–787.

[26] C. Wang, H. Zhang, T.-M. Nguyen and L. Xie, Ultra-wideband aided fast localization and mapping system, *2017 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2017), pp. 1602–1609.

[27] T. H. Nguyen, M. Cao, T.-M. Nguyen and L. Xie, Post-mission autonomous return and precision landing of uav, *2018 15th Int. Conf. Control, Automation, Robotics and Vision (ICARCV)* (IEEE, 2018), pp. 1747–1752.

[28] J. Tiemann, A. Ramsey and C. Wietfeld, Enhanced uav indoor navigation through slam-augmented uwb localization, *2018 IEEE Int. Conf. Communications Workshops (ICC Workshops)* (IEEE, 2018), pp. 1–6.

[29] Y. Song, M. Guan, W. P. Tay, C. L. Law and C. Wen, Uwb/lidar fusion for cooperative range-only slam, *2019 Int. Conf. Robotics and Automation (ICRA)* (IEEE, 2019), pp. 6568–6574.

[30] F. J. Perez-Grau, F. Caballero, L. Merino and A. Viguria, Multi-modal mapping and localization of unmanned aerial robots based on ultra-wideband and rgb-d sensing, *2017 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2017), pp. 3495–3502.

[31] T.-M. Nguyen, Z. Qiu, T. H. Nguyen, M. Cao and L. Xie, Distance-based cooperative relative localization for leader-following control of mavs, *IEEE Robot. Autom. Lett.* **4**(4) (2019) 3641–3648.

[32] T.-M. Nguyen, Z. Qiu, M. Cao, T. H. Nguyen and L. Xie, An integrated localization-navigation scheme for distance-based docking of uavs, *2018 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 5245–5250.

[33] T.-M. Nguyen, Z. Qiu, M. Cao, T. H. Nguyen and L. Xie, Single landmark distance-based navigation, *IEEE Trans. Control Syst. Technol.* **(early access)** (2019).

[34] A. Shariati, K. Mohta and C. J. Taylor, Recovering relative orientation and scale from visual odometry and ranging radio measurements, *2016 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2016), pp. 3627–3633.

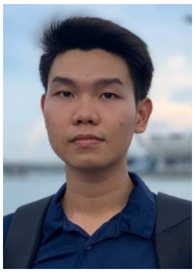
[35] K. Levenberg, A method for the solution of certain non-linear problems in least squares, *Q. Appl. Math.* **2**(2) (1944) 164–168.

[36] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik and R. Siegwart, The euroc micro aerial vehicle datasets, *Int. J. Robot. Res.* **35**(10) (2016) 1157–1163.

[37] S. Umeyama, Least-squares estimation of transformation parameters between two point patterns, *IEEE Trans. Pattern Anal. Mach. Intell.* (4) (1991) 376–380.

[38] J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, A benchmark for the evaluation of rgb-d slam systems, in *2012 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (IEEE, 2012), pp. 573–580.

[39] J. Delmerico and D. Scaramuzza, A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots, in *2018 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, 2018), pp. 2502–2509.



Thien Hoang Nguyen received his Bachelor of Engineering (Honors) in Electrical and Electronic Engineering from Ho Chi Minh City University of Technology, Vietnam in 2017. From 2017 to 2018, he was a Project Officer at the Nanyang Technological University, Singapore. He is currently working towards the Ph.D. degree with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His current research interests include visual-based perception and navigation for autonomous mobile robots.

Thien-Minh Nguyen received his Bachelor of Engineering (Honors) in Electrical and Electronic Engineering from Ho Chi Minh City University of Technology in 2014. He is currently working towards his Ph.D. degree at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include robot navigation, learning and adaptive systems, multi-robot systems, unmanned micro-aerial vehicles.

Muqing Cao received his Bachelor of Engineering (Honors) in Aerospace Engineering from Nanyang Technological University, Singapore. He is currently working towards the Ph.D. degree with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His current research interests include path planning and control for unmanned aerial vehicles.

Lihua Xie received the B.E. and M.E. degrees in Electrical Engineering from Nanjing University of Science and Technology in 1983 and 1986, respectively, and the Ph.D. degree in Electrical Engineering from the University of Newcastle, Australia, in 1992. Since 1992, he has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, where he is currently a Professor and Director, Delta-NTU Corporate Laboratory for Cyber-Physical Systems. He served as the Head of Division of Control and Instrumentation from July 2011 to June 2014. He held teaching appointments in the Department of Automatic Control, Nanjing University of Science and Technology from 1986 to 1989.

His research interests include robust control and estimation, networked control systems, multi agent networks, localization and unmanned systems. He is an Editor-in-Chief for Unmanned Systems and an Associate Editor for IEEE Transactions on Network Control Systems. He has served as an editor of IET Book Series in Control and an Associate Editor of a number of journals including IEEE Transactions on Automatic Control, Automatica, IEEE Transactions on Control Systems Technology, and IEEE Transactions on Circuits and Systems II. He is an elected member of Board of Governors, IEEE Control System Society (Jan 2016–Dec 2018). He is a Fellow of IEEE and a Fellow of IFAC.