Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Real-Time Drift-Free Path Recording and Replaying Based On Virtual Image Matching For MAVs

Ruihang Miao, Peilin Liu, Fei Wen, Zheng Gong, Wuyang Xue, Rendong Ying

***Abstract*—This paper presents an efficient method to achieve real-time drift-free path replaying for Micro Aerial Vehicle (MAV). The drift-free path replaying method is proposed for precise autonomous path following in GPS-denied environments. In this paper, we present solutions on how to effectively select recording images during flying and drive the MAV to fly automatically to target position. During the path replaying, a strategy named triggered virtual image matching is used for reducing the computational cost and eliminating the drift. Time of image selection is reduced to negligible due to above strategy. Moreover, we use optical flow method for features matching between current image and the virtual image which is precise and faster. Experimental results show that our approach can achieve autonomous returning and precise path following in indoor scene and outdoor scene.**

***Index Terms*—Vision-based navigation, Path following, Micro aerial vehicle (MAV)**

I. Introduction

Autonomous path following is a fundamental problem in navigation of micro aerial vehicles (MAVs). When dedicated positioning infrastructure is unavailable, such as in GPS-denied indoor environments or GPS-unreliable urban environments with tall buildings, precise path following only based on vision is a challenging problem. With vision only sensors, a potential solution is to use optical flow or simultaneous localization and mapping (SLAM) based methods for position estimation [1]–[3], and navigate along a pre-recorded path based on the estimated position.

Accurate local relative localization is of critically important for reliably following an appointed path. As visual odometry and SLAM often suffer from accumulated localization error, it would lead to deviation of the flight path from the desired one. Meanwhile, this deviation usually increases as the travel distance increases. As a consequence, the MAV may be navigated to a position far away from the desired target. Although SLAM is able to provide accurate localization results when there exist loop closures in the route, it depends on how the MAV flies and may not always be satisfied. Moreover, most SLAM algorithms are computationally expensive, which hinders their use in light-weighted MAVs with limited computational resources.

To achieve precise path following in the condition that only inaccurate position information is available, this work proposes a novel two-phase solution, namely path recording and replaying approach. The new approach mainly consists of an active path record module, a virtual image matching module and a triggered position correction module. It is efficient and enables real-time running on light-weighted MAVs. Besides, it is drift-free since the drift is corrected by virtual image matching in time. In order to record the path as intact as possible, some factors includes the sample rate, orientation, curvature of path are concerned. The proposed path replaying method does not give a precise global position but the precise relative position between MAV and the inherent feature along the path. During computing the relative position, we use an asynchronous method since image process is complicated and time-consuming. Furthermore, a nonlinear control method of MAV is used to reduce the trajectory error that will make our path replaying system work better.

In this paper we propose an novel path recording and replaying approach that can work without global localization device for MAVs. The main contributions of this paper are:

1) An active path recording method which can recover the path with a negligible error.
2) A fast and accurate image matching method based on virtual frame.
3) A real-time trajectory correcting method which combines trajectory control errors and image matching.

The rest of this paper is organized as follows. Related work is discussed in Section II. In Section III, the problem statement is defined. In Section IV, we present our implementation of path recording and replaying method based on image matching in detail. Subsequently, we show the results of simulation experiments and real world experiments in Section V. Lastly, a brief conclusion about our work is given in section VI.

II. Related work

Until now, many works have been done to improve the autonomous navigation of MAV such as reducing computational cost, using multi-sensors fusion and so on. For reducing the running time on SLAM, the parallel tracking and mapping method which chose a camera as the main sensor was proposed in [4]. And the approach of navigating a high speed MAV on limited onboard computer is proposed in [5]. The camera on MAVs sometimes can be placed looking downward as in [6], [7], or can be placed looking forward as in [8]. Moreover, S. Yang *et al.* presented a method for visual SLAM that combines the downward camera and the forward camera together in [9]. Normally, the downward camera provides high frequency velocity of the MAV while the forward camera provides the position of the MAV. Consequentially, combining the downward camera and the forward camera to calculate

The authors are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: mrhcat@sjtu.edu.cn; liupeilin@sjtu.edu.cn; wenfei@sjtu.edu.cn; gongzheng@sjtu.edu.cn; icecreamxwy@sjtu.edu.cn; rdying@sjtu.edu.cn;).

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

the position provides an ideal method for precise trajectory tracking without considering the computation cost.

More recently, vision-based solutions become the dominant solution for autonomous navigation of MAVs. S. A. Scherer *et al.* demonstrated a drift-corrected tracking and mapping method for autonomous MAVs in [10]. With the help of an RGBD camera, the accumulated drift during the long time running can be corrected. This paper also proposed a method that fuses the information about efficiently detected ground planes with visual odometer, which contributes to its drift correction in attitude and altitude. Inspired by this paper, we fuse stereo camera information and optical flow information to correct the position and the attitude. Moreover, G. Loianno *et al.* built an autonomous navigation system for MAVs by using smartphone in [11]. In this work, all the computation, include sensing and control, runs on the smartphone. This paper shows how the MAVs can achieve autonomous flight in indoor buildings by control commands sent by smartphone.

In addition, many works about image-based navigation, one kind of vision-based navigation methods, have been proposed in [12]–[15]. Though these methods are proposed for ground robots within 2-D plane, they can be extended into 3-D space. The work in [16] extends the "funnel lane" concept in [12] for path following of MAV. The passive localization approaches are used in the aforementioned image-based navigation methods except [15]. In [15], an active vision-based localization and navigation approach is proposed by G. Luca *et al.* In this work, the active strategy can disambiguate the location from possible hypotheses. Inspired by this work, we proposed the triggered position correction method which corrects the location by combining the information of control and image matching.

III. Problem Statement

We consider that one 3-D path $\boldsymbol{r}(t)$ is given by an inaccurate map G when solving the MAVs' path following problem. The map G will provide a series of frames which contain N_m images $\{\mathcal{I}_k\}_{k=1,\dots,N_m}$, $N_m \times n$ 2-D points $\{\boldsymbol{x}_{kl}\}_{k=1,\dots,N_m,l=1,\dots,n}$ on image and correspond 3-D map points $\{\boldsymbol{X}_{kl}\}_{k=1,\dots,N_m,l=1,\dots,n}$. The objective is to design a method that drives MAVs fly as the given path and reach to target position without the global locating device. Actually, there are three paths during solving the problem. The three paths are the given (or recording) path $\boldsymbol{r}_{rcd}(t)$ via map, the following path $\boldsymbol{r}_{flw}(t)$ calculated by MAVs and the physics path $\boldsymbol{r}_{real}(t)$ in real world. However, the three paths are usually not the same because there are always localization errors both in the recorded map and current MAVs' localization module.

IV. Methodology

Fig. 1 shows the system overview of proposed approach. The data of IMU and optical flow is input of pose estimation module and image data is directly sent to image matching module. Trajectory module sends control error to image matching module since it contains path generator and position control.

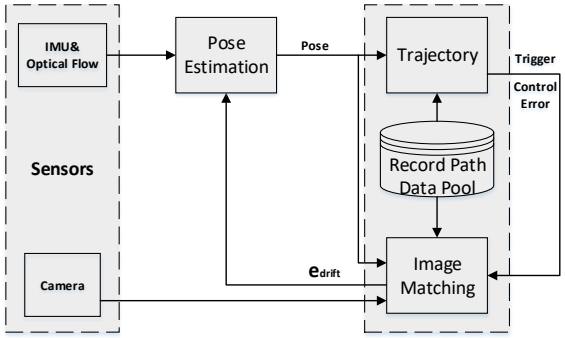


Fig. 1. System overview.

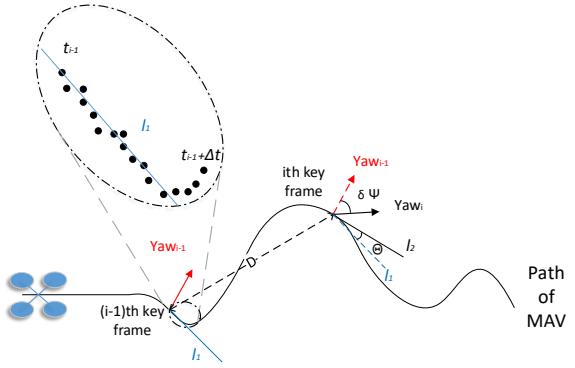


Fig. 2. Parameters used during recording path: distance D between two neighbouring recording position, included angle $\delta\psi$ between two neighbouring recording yaw angle, angle Θ which equals to curvature K of the path at the recording point.

A. Active path recording

An active path recording method is proposed for recovering the path accurately and eliminating the drift. The recording data should contain 3-D points so that we can calculate the relative pose through forward-looking camera. Thus, stereo camera are used as the data collecting device.

We define a $keyfram$ which contains the useful information including 3-D position $\boldsymbol{X}_W$ of recording device, attitude quaternion $\boldsymbol{q}$ of recording device, arrival time t and the depth information extracted from stereo images. There are 4 path recording rules described as following,

- Limit the maximum distance D_{max} between two neighbouring recording position.
- Limit the maximum included angle $\delta\psi_{max}$ between two neighbouring recording yaw angle calculated through attitude quaternion $\boldsymbol{q}$.
- Limit the maximum included angle Θ_{max} between two tangent line of neighbouring recording points on the flying path.
- Limit the minimum the number of feature points N_{min} in stereo image.

As shown in Fig. 2, the maximum distance D_{max} between two neighbouring recording position is decided by the average drift velocity $\boldsymbol{v}_{drift}$, MAVs' velocity $\boldsymbol{v}$ and the maximum tolerable trajectory error E_{max}. The relationship between the four parameters is given by

$$D_{max} = \frac{||\boldsymbol{v}||}{||\boldsymbol{v}_{drift}||} E_{max}. \tag{1}$$

The $\delta\psi_{max}$ is the included angle between two neighbouring recording yaw angle. It is decided by the overlap of camera view. This means $\delta\psi_{max}$ is related to the viewing angle of camera Φ. Normally, we calculate $\delta\psi_{max}$ by

$$\delta\psi_{max} = \frac{\Phi}{4}. \tag{2}$$

For Θ, the included angle between tangent lines, we need to calculate the tangent line of the trajectory. However, it is difficult to directly calculate the precise tangent of one discrete curve. Here, we chose to use an approximation of tangent line. Denoting the trajectory as $\boldsymbol{r}(t)$, the arrival time of the i-th $keyframe$ as t_i and an interval time of sampling as Δt. The points set O_1 and O_2 are collected during the time t_{i-1} to $t_{i-1} + \Delta t$ and during the time $t_i - \Delta t$ to t_i respectively. The collecting points are come from pose estimation module at 50 Hz. The direction of the tangent lines, denoted by $\boldsymbol{k}_{l1}$ and $\boldsymbol{k}_{l2}$, can be calculated through least square method. Thus, the approximate included angle Θ of the two tangent line could derived easily from $\boldsymbol{k}_{l1}$ and $\boldsymbol{k}_{l2}$ as

$$\Theta = cos^{-1}(\frac{\boldsymbol{k}_{l1}\boldsymbol{k}_{l2}}{||\boldsymbol{k}_{l1}||||\boldsymbol{k}_{l2}||}). \tag{3}$$

For more precisely recovering the path, we assume that we can recover a circle by cubic spline interpolation method. According to [17], the max error of cubic spline interpolation is given by

$$\max_{a<b<c} |f(x) - S(x)| \leq \frac{5}{384} |f^{(4)}(x) h^4|, \tag{4}$$

where the $f(x)$ denotes origin curve, $S(x)$ denotes the spline interpolation curve and h denotes the maximum interval. For a unit circle, the max error must be less than $\frac{5}{384}h^4$. In our case, the maximum interval h is decided by inequality as

$$h \leq 2 \sin(\frac{\Theta_{max}}{2}). \tag{5}$$

Thus, we take $\Theta_{max} = \pi/8$ which makes max relative error be less than 3.10×10^{-4}.

The last rule is about the minimum feature points changed with the environment. We have to ensure the feature points are enough during flying so that the MAVs can calculate relate pose precisely. Normally, N_{min} is setting as 50.

For example, the maximum distance D_{max} is setting as 0.6m according to the (1) when the drift velocity is $5m/min$. We can control the maximum drift distance within 0.1m during flying to the next recording position in our MAV system when the MAV flies at 0.5m/s. And the $\delta\psi_{max}$ is decided by the viewing angle of camera and the size of flying space. In our test space with a size of $8m \times 5m \times 3m$, the $\delta\psi_{max}$ is setting as 20 degree. And Θ_{max} is setting as $\frac{\pi}{8}$ because we can almost recover a semicircle under this value.

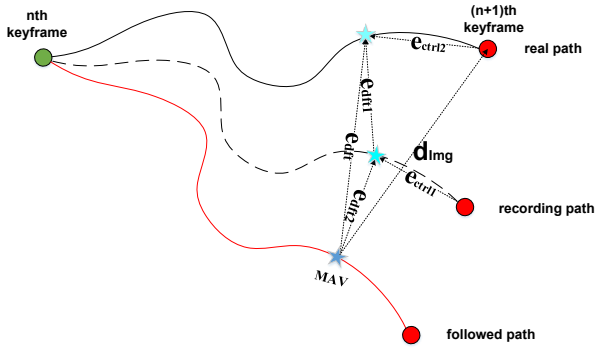


Fig. 3. The relationship between the three paths: real path, recording path and followed path.

B. Path drift correction

The recording path generated by the active path recording approach has a drift error $\boldsymbol{e}_{dft1}(t)$ that can not be eliminated. When MAVs receive the record path and try to follow this path, there is another drift error $\boldsymbol{e}_{dft2}(t)$ caused by the drift of pose estimation. Compare with the real path, We have

$$\begin{aligned} \boldsymbol{r}_{rcd}(t) &= \boldsymbol{r}_{real}(t) + \boldsymbol{e}_{dft1}(t), \\ \boldsymbol{r}_{flw}(t) &= \boldsymbol{r}_{real}(t) + \boldsymbol{e}_{dft}(t), \\ \boldsymbol{e}_{dft}(t) &= \boldsymbol{e}_{dft1}(t) + \boldsymbol{e}_{dft2}(t). \end{aligned} \tag{6}$$

Image matching and path correcting are executed when the trajectory planner sends a trigger signal. The trigger signal is sent out when the trajectory planner thinks MAV reaches one record $keyframe$. The relationship between three paths are as shown in Fig. 3.

Now, we would like to calculate $\boldsymbol{e}_{dft}$ via control error and image matching as following

$$\boldsymbol{e}_{dft} = \boldsymbol{e}_{ctrl2} + \boldsymbol{d}_{Img}. \tag{7}$$

First, the trajectory planner should know the control error $\boldsymbol{e}_{ctrl1}$ because trajectory planner controls MAVs to track trajectory according to record path. Second, $\boldsymbol{d}_{Img}$ can be calculated from the image matching which will be introduced in next section. We note that (7) use $\boldsymbol{e}_{ctrl2}$ rather than $\boldsymbol{e}_{ctrl1}$ and the fact that $\boldsymbol{e}_{ctrl1}$ is an approximate vector to $\boldsymbol{e}_{ctrl1}$. So the path correct could be expressed as

$$\boldsymbol{e}_{dft} = \boldsymbol{e}_{ctrl1} + \boldsymbol{d}_{Img} + (\boldsymbol{e}_{ctrl2} - \boldsymbol{e}_{ctrl1}). \tag{8}$$

Note that the $\boldsymbol{e}_{ctrl2} - \boldsymbol{e}_{ctrl1}$ is depended on average drift velocity $\boldsymbol{v}_{dft1}$ and the error of trajectory control $\boldsymbol{e}_{ctrl1}$. Normally $\boldsymbol{e}_{ctrl2} - \boldsymbol{e}_{ctrl1}$ is small enough that (8) can be written as

$$\boldsymbol{e}_{dft} \approx \boldsymbol{e}_{ctrl1} + \boldsymbol{d}_{Img}. \tag{9}$$

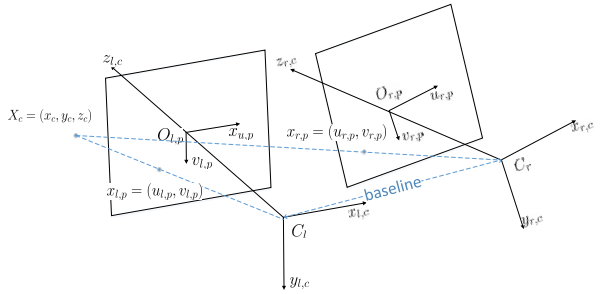


Fig. 4. Configurations of the stereo camera frames and their image planes are depicted.

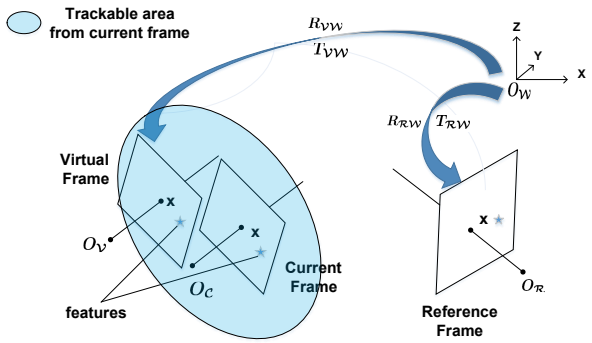


Fig. 5. The relationship between virtual frame, reference frame and current frame. Virtual frame is created from reference frame for matching enough points.

C. Virtual Image matching

In our system, the coordinates is established as in Fig. 4. Denoting world frame as $\mathcal{W}$ and left camera coordinate frame as $\mathcal{C}$. The pose of left camera in world frame can be expressed with a 3×4 matrix, denoted by $\boldsymbol{P}_{\mathcal{WC}} \in SE(3)$. The pose matrix can be decomposed as

$$\boldsymbol{P}_{\mathcal{WC}} = [\boldsymbol{R}_{\mathcal{WC}} \quad \boldsymbol{T}_{\mathcal{WC}}], \tag{10}$$

where $\boldsymbol{R}_{\mathcal{WC}}$ denotes the rotation information and $\boldsymbol{T}_{\mathcal{WC}}$ denotes the translation information.

The proposed virtual image matching contains three steps : 1) to prepare the $keyframe$, 2) to transform the reference frame into virtual frame, 3) to match current frame with reference frame. Each $keyframe$ should store the following data,

- the feature points set V_i extra from left image;
- 3-D position $\boldsymbol{X}$ corresponding the feature points;
- the pose $\boldsymbol{P}$ of left camera;
- the arrival time t of the MAV when recording the position;
- the average depth d_{avg} of the 3-D points calculated from stereo camera.

When preparing the $keyframe$, the feature points and 3-D position of them need to be calculated. For processing more easily, the origin point of the stereo camera is set as the center of left camera. Then, FAST corners [18] or Shi-Tomasi corners [19] are detected in left image. The detected corners are denoted by $\boldsymbol{x}_l$. Subsequently, Lucas-Kanade algorithm in [20] is applied to find the matching points $\boldsymbol{x}_r$ on right image. Thus, 3-D position of the corresponding 3-D points can be extracted.

Considering calibrated images, the projection equation is given by

$$\hat{\boldsymbol{x}} = \pi(\boldsymbol{P}_{\mathcal{WC}}, \hat{\boldsymbol{X}}) = sK\boldsymbol{P}_{\mathcal{WC}}\hat{\boldsymbol{X}}, \tag{11}$$

where K denotes the camera intrinsics matrix, s denotes a scale for translating projection result to homogeneous coordinates, the $\hat{\boldsymbol{x}} = [x, y, 1]^T$ denotes the homogeneous coordinates of point $\boldsymbol{x} = [x, y]^T$ on image planes and $\hat{\boldsymbol{X}} = [X, Y, Z, 1]^T$ denotes the homogeneous coordinates of 3-D point $\boldsymbol{X} = [X, Y, Z]^T$ in real world. In calibrated stereo camera system, the coordinates of camera centers C_l and C_r are known and $\boldsymbol{x}_l, \boldsymbol{x}_r$ are calculated as above processing. Thus, there are two equations as (11) for $\boldsymbol{x}_l$, $\boldsymbol{x}_r$ respectively. Then, the unknown 3-D points can be reconstructed by a triangulation function.

After the 3-D position $\boldsymbol{X}$ of each point is calculated, we can calculate the average depth d_{avg} which is used in producing virtual frame later. In following derivation, we only concern on left camera and calculate its relative pose with certain reference $keyframe$. Denoting the relative pose as $\boldsymbol{P}_i \in SE(3)$, where the subscript i represents that the reference frame is the i-th recording $keyframe$. $\boldsymbol{P}_i$ gives the relationship between the pose of camera in i-th recording $keyframe$ and the current pose of camera. The relationship can be expressed as

$$\boldsymbol{P}_{\mathcal{WC}}(t) = \boldsymbol{P}_i\boldsymbol{P}_{\mathcal{WC},i}, \tag{12}$$

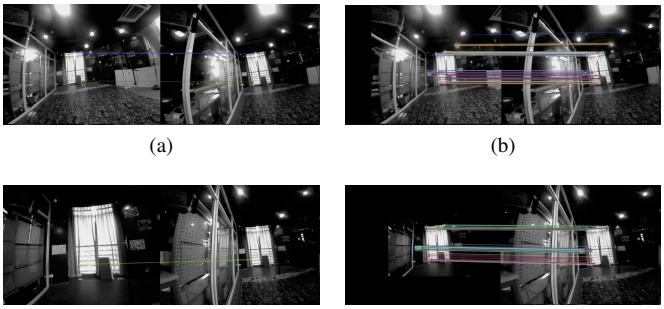
where $\boldsymbol{P}_{\mathcal{WC},i}$ denotes the pose of camera in i-th recording $keyframe$ and $\boldsymbol{P}_{\mathcal{WC}}(t)$ denotes the current pose of camera.

Lucas-Kanade algorithm is used to find enough matching points. It is known that Lucas-Kanade algorithm only works well when the images are close enough. Thus, for robustly image matching, we create a virtual frame transformed from the reference $keyframe$. The relationship between virtual frame, reference frame and current frame is shown in Fig. 5.

During the MAV flies between two neighbouring recording $keyframes$, an accurate pose is acquired through the pose estimation of MAV after fusing the drift correction at last recording $keyframe$. Denoting this accurate pose as $\boldsymbol{P}_{\mathcal{VW}}$, rotation matrix as $\boldsymbol{R}_{\mathcal{VW}}$ and translation matrix as $\boldsymbol{T}_{\mathcal{VW}}$. $\boldsymbol{R}_{\mathcal{VW}}$ and $\boldsymbol{T}_{\mathcal{VW}}$ can be acquired from $\boldsymbol{P}_{\mathcal{VW}}$. The recording pose, denoted by $\boldsymbol{P}_{\mathcal{RW}}$, can be acquired from reference $keyframe$. The subscript "$\mathcal{VW}$" represent from world coordinate to virtual frame and subscript "$\mathcal{RW}$" represent from world coordinate to reference frame.

For each 2-D point on image, there is a corresponding 3-D point in real world. According to perspective geometry, the relationship between 2-D points and 3-D points in virtual frame and reference frame can be written as

$$\hat{\boldsymbol{x}}_{\mathcal{V}} = s_{\mathcal{V}}\boldsymbol{K}[\boldsymbol{R}_{\mathcal{VW}} \quad \boldsymbol{T}_{\mathcal{VW}}]\hat{\boldsymbol{X}}_{\mathcal{W}}, \tag{13}$$



(a) (b) (c) (d)

Fig. 6. Comparison of result of 2-D points matching with descriptor matching and our method. (a), (c): Result of 2-D point matching by matching descriptors. (b), (d): Result of our 2-D point matching method.

$$\hat{\boldsymbol{x}}_{\mathcal{R}} = s_{\mathcal{R}} \boldsymbol{K}'[\boldsymbol{R}_{\mathcal{R}\mathcal{W}} \quad \boldsymbol{T}_{\mathcal{R}\mathcal{W}}]\hat{\boldsymbol{X}}_{\mathcal{W}}, \tag{14}$$

where $\boldsymbol{K}$, $\boldsymbol{K}'$ denote the intrinsics matrix of MAV's camera and the intrinsics matrix of data collecting device's camera respectively, $\hat{\boldsymbol{x}}_{\mathcal{V}}$, $\hat{\boldsymbol{x}}_{\mathcal{R}}$ denote the homogeneous coordinates of 2-D point in virtual frame and reference frame respectively and $\hat{\boldsymbol{X}}_{\mathcal{W}}$ denotes the homogeneous coordinates of the corresponding 3-D point. The relationship between $\hat{\boldsymbol{x}}_{\mathcal{V}}$ and $\hat{\boldsymbol{x}}_{\mathcal{R}}$ can be derived through (13) and (14). The relationship is given by

$$\begin{aligned} \hat{\boldsymbol{x}}_{\mathcal{V}} =& \frac{s_{\mathcal{V}}}{s_{\mathcal{R}}} K \boldsymbol{R}_{\mathcal{V}\mathcal{W}} \boldsymbol{R}_{\mathcal{R}\mathcal{W}}^T K'^{-1} \hat{\boldsymbol{x}}_{\mathcal{R}} \\ &+ s_{\mathcal{V}} K(\boldsymbol{T}_{\mathcal{V}\mathcal{W}} - \boldsymbol{R}_{\mathcal{V}\mathcal{W}} \boldsymbol{R}_{\mathcal{R}\mathcal{W}}^T \boldsymbol{T}_{\mathcal{R}\mathcal{W}}), \\ s_{\mathcal{V}} =& \frac{1}{Z_{\mathcal{V}}}, \\ s_{\mathcal{R}} =& \frac{1}{Z_{\mathcal{R}}}, \end{aligned} \tag{15}$$

where the $Z_{\mathcal{V}}$ and $Z_{\mathcal{R}}$ denote the depth of this 3-D point in reference frame and virtual frame respectively. Normally, $Z_{\mathcal{V}}$ is approximate to $Z_{\mathcal{R}}$. Because the depth of each point can hardly acquired, we use the average depth calculated by selected points to take place of $Z_{\mathcal{V}}$ and $Z_{\mathcal{R}}$. At last, the relationship between $\hat{\boldsymbol{x}}_{\mathcal{V}}$ and $\hat{\boldsymbol{x}}_{\mathcal{R}}$ is rewritten as

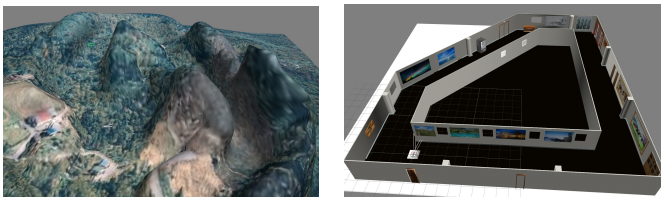
$$\hat{\boldsymbol{x}}_{\mathcal{V}} = K \boldsymbol{R}_{\mathcal{V}\mathcal{W}} \boldsymbol{R}_{\mathcal{R}\mathcal{W}}^T K'^{-1} \hat{\boldsymbol{x}}_{\mathcal{R}} + \frac{K(\boldsymbol{T}_{\mathcal{V}\mathcal{W}} - \boldsymbol{R}_{\mathcal{V}\mathcal{W}} \boldsymbol{R}_{\mathcal{R}\mathcal{W}}^T \boldsymbol{T}_{\mathcal{R}\mathcal{W}})}{d_{avg}}. \tag{16}$$

Thus, one to one relationship between 2-D points on reference frame and virtual frame is given. The virtual image can be created by applying (16) on each point on the reference frame. Later Lucas-Kanade algorithm is used to find the matching points $\boldsymbol{x}_{\mathcal{C}}$ in current frame.

After calculating the matching points, the homography matrix and the fundamental matrix with an efficient fitting algorithms [21] are used to remove outliers. Comparison results of points matching with descriptor matching and virtual image matching is given in Fig. 6.

After the correspondences of 3-D points and 2-D image points are given, an optimization method is used to reconstruct the relative pose of the camera at time t as

$$\boldsymbol{P}_i = \arg\min_{\boldsymbol{P}} (\sum_{\mathcal{I}} \rho(\boldsymbol{x}_{\mathcal{C}}, \pi(\boldsymbol{P}\boldsymbol{P}_{\mathcal{W}\mathcal{C}}, \boldsymbol{X}))), \tag{17}$$



(a) (b)

Fig. 7. Two simulation scene in gazebo. (a): valley scene with a size about $50m \times 20m \times 10m$. (b): corridor scene with a size about $10m \times 10m \times 2m$.

where the function ρ denotes the square of Euclidean distance, $\mathcal{I} = \{(\boldsymbol{x}_{\mathcal{C}}, \boldsymbol{X}) | \boldsymbol{x}_{\mathcal{C}} \leftrightarrow \boldsymbol{X}, \boldsymbol{x}_{\mathcal{C}} \in V_i\}$ denotes the correspondence set of 2-D image points and 3-D points and V_i denotes all matching feature points in i-th $keyframe$.

D. Trajectory generation

In trajectory planning module, the position of MAV and yaw angle of MAV are required in each recording point. Trajectory is split into n pieces and is only given the start and end point for each piece. We take all recording points as inputs of trajectory planning. Obviously, it is easy to expressed the trajectory as a piecewise polynomial functions in n time intervals. And since the continuous position, continuous velocity and continuous acceleration are required, the trajectory is regarded as minimum acceleration trajectory. That means our trajectory is the solution of the optimization as

$$\min_{\boldsymbol{r}_i(t), \psi_i(t)} \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} L(\boldsymbol{r}_i(t), \dot{\boldsymbol{r}}_i(t), \ddot{\boldsymbol{r}}_i(t), \psi_i(t), \dot{\psi}_i(t), \ddot{\psi}_i(t)) dt. \tag{18}$$

The solution of the optimization is given by the Euler-Lagrange equation with $L(t) = \ddot{\boldsymbol{r}}_i(t)^2 + \ddot{\psi}_i(t)^2$. And the result of Euler-Lagrange equation shows that cubic spline is a good choice for the path generating.

V. Simulation and Experimental Results

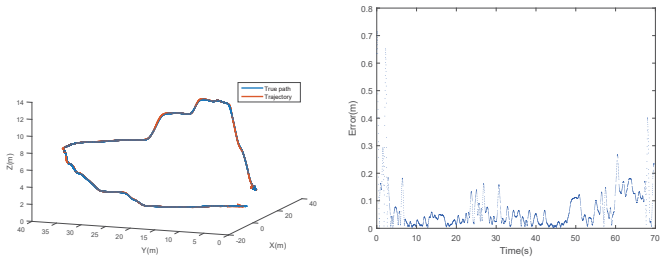
A. Simulation experiments

Simulation are carried out in Gazebo[1] with two scenes: valley and corridor. We add a man-made horizontal drift with a drift velocity at 5m/min. That means if there is no drift correction, MAV will have a collision with the peak or wall. The scenes are as shown in Fig. 7.

Position estimation and controller both runs at 50 Hz. We will run at different flying velocity to analysis the performance of our approach. The parameters used for simulation is given in Table I,

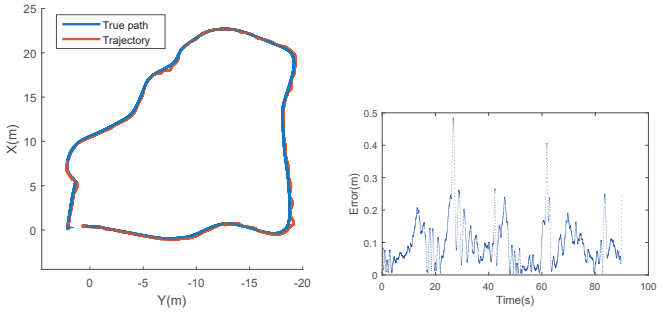
The simulation result in valley scene is as shown in Fig. 8. The average velocity of MAV is set to $0.9m/s$ in experiment and the MAV flies around $70s$ with 275 $keyframes$. The MAV is put at a position where is about $0.7m$ far away from the start point of the trajectory. Thus, there is a big error when MAV starts to follow the trajectory. The errors are limited under $0.3m$ once the MAV catches up the target position for the first time.

[1] http://gazebosim.org/



(a) (b)

Fig. 8. The results of simulation in valley scene with average velocity equal to $0.9m/s$. (a): The trajectory (red) and the true path (blue). (b): The performance of the following result, path error is almost less than 0.2m.



(a) (b)

Fig. 9. The result of simulation in corridor scene with average velocity equal to $0.5m/s$. (a): The trajectory (red) and the true path (blue). (b): The performance of the following result, path error is almost less than 0.3m.

simulation result in corridor scene is as shown in Fig. 9. The average velocity is set to $0.5m/s$ and MAV flies with 178 $keyframes$. The distance between wall and MAV is so close that the matching is difficult when MAV turns right or turns left. Thus, most of the big errors are appeared at corners. However, the error will be fixed quickly by our approach. The average trajectory error is $9.6cm$.

The simulation results show that the MAV is able to follow the recording path within an appointed time. And the trajectory errors are within the required tolerance in both indoor and outdoor environments. Through analysis variation of error over time, the value of error appears periodically. This is caused by the fact that the pose correcting has only been carried out while MAV reach one point that has recording $keyframe$.

Then we try different velocity to analyze the performance in valley scene. The result is shown in Fig. 10. The max average trajectory error is 0.23 m while the velocity is 2.0 m/s. The result shows that the error increases with velocity

TABLE I
PARAMETERS USED FOR SIMULATION.

Parameter	*Value*
D_{max}	0.6m
$\delta\psi_{max}$	$\frac{\pi}{6}rad$
Θ_{max}	$\frac{\pi}{8}rad$
N_{min}	50
v_{drift}	$6m/min$
v_{avg}	0.5-2.5m/s

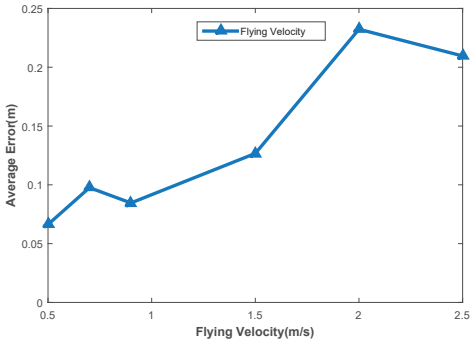
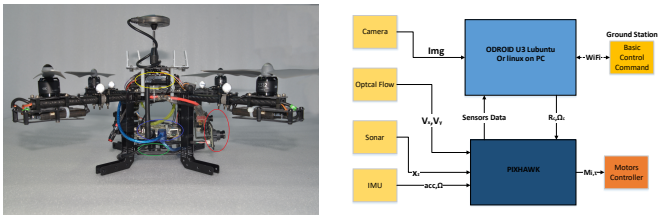


Fig. 10. The relationship between average flying velocity and average trajectory error.



(a) (b)

Fig. 11. Our MAV platform. (a): Our MAV platform, with PX4FMU (yellow ellipse), onboard computer (blue ellipse), stereo camera (red ellipse) mounted looking forward and PX4FLOW (green ellipse) mounted looking downward. (b): System hardware structure.

increases. This is because the image matching would become more difficult when MAV moves faster. The result also shows that our path following method is robust. Even a fairly large velocity is set, MAV can still be able to follow the recording path successfully.

B. Experiment with Ground Truth

The path replaying experiment in real world is taken on our MAV platform. On our MAV platform, one stereo camera mounted looking forward and one PX4FLOW camera mounted looking downward. Our MAV platform is shown in Fig. 11. The flight management unit we used is the PIXHAWK. Only the inner loop control is used for controlling the attitude and controlling the throttle of the MAV. While the position control, pose estimation and image processing are running on a onboard computer or a laptop. All of the control processing and image processing are running on-board while the start, record and return command are sent from ground station. For processing each image within 0.1s, images from stereo camera are converted to 320×240 pixels and the number of feature points are limited in 1000 for each frame. In this experiment, VICON motion capture system [22] is used as the ground truth.

The path of the MAV is set as a $2m \times 2m$ square in a room. The experiment is split into two stage. The first stage is that the MAV flies along the square path. At the same time, path recording module records the valid information. The second stage is that MAV follows the recorded path or auto return as the recording path with its own pose estimation module.

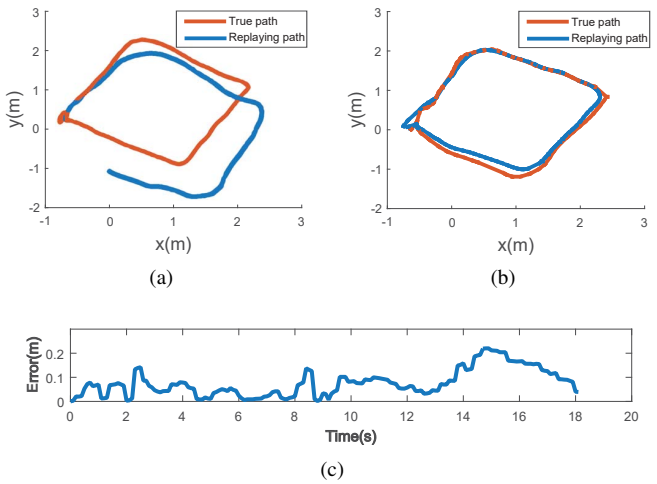


Fig. 12. Results of real-time drift-free path replaying experiment. (a): the result without drift correction under VICON system. (b): the result with drift correction under VICON system. (c): Performance data of path replaying.

For comparison, another replaying experiment without drift correction is taken. And the results are shown as Fig. 12.

The result of path replaying without drift correction is shown in Fig. 12(a). There is about 1m drift in X direction and about 1m drift in Y direction in this experiment when the path replaying is over. The result of path replaying with drift correction is shown in Fig. 12(b). The result shows MAV is able to fly along the recorded path and the following error would not increase with time.

VI. Conclusion

In this paper, we propose a path recording and replaying approach for solving real-time path following problem without global localization device on MAVs. The active path recording method considers several factors that it can recover the path within a small error. And triggered path replaying method gives MAV a real-time pose correction for navigating without a precise global localization. The whole path recording and replaying system is verified in both broad and narrow environments. This approach shows how MAVs use the stereo camera as their eyes to navigate themselves without a precise global localization. In future work, obstacle avoidance and estimating the trend of drift will be concerned for better performance.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 61871265 and 61903246.

References

[1] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, "An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1736–1741.

[2] S. Yang, S. A. Scherer, and A. Zell, "Visual slam for autonomous mavs with dual cameras," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5227–5232.

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

[4] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *IEEE and Acm International Symposium on Mixed and Augmented Reality*, 2008.

[5] S. Liu, M. Watterson, S. Tang, and V. Kumar, "High speed navigation for quadrotors with limited onboard sensing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1484–1491.

[6] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation*, 2014.

[7] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a vtol uav on a moving platform using image-based visual servoing," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 971–976.

[8] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor mav," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 4557–4564.

[9] S. Yang, S. A. Scherer, and A. Zell, "Visual slam for autonomous mavs with dual cameras," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 5227–5232.

[10] S. A. Scherer, S. Yang, and A. Zell, "Dctam: Drift-corrected tracking and mapping for autonomous micro aerial vehicles," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2015, pp. 1094–1101.

[11] G. Loianno, Y. Mulgaonkar, C. Brunner, D. Ahuja, A. Ramanandan, M. Chari, S. Diaz, and V. Kumar, "Smartphones power flying robots," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 1256–1263.

[12] Z. Chen and S. T. Birchfield, "Qualitative vision-based path following," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 749–754, 2009.

[13] J. Courbon, Y. Mezouar, and P. Martinet, "Indoor navigation of a non-holonomic mobile robot using a visual memory," *Autonomous Robots*, vol. 25, no. 3, pp. 253–266, 2008.

[14] A. Diosi, S. Segvic, A. Remazeilles, and F. Chaumette, "Experimental evaluation of autonomous driving based on visual memory and image-based visual servoing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 870–883, 2011.

[15] G. L. Mariottini and S. I. Roumeliotis, "Active vision-based robot localization and navigation in a visual memory," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 6192–6198.

[16] T. Nguyen, G. K. Mann, and R. G. Gosine, "Vision-based qualitative path-following control of quadrotor aerial vehicle," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2014, pp. 412–417.

[17] C. A. Hall and W. W. Meyer, "Optimal error bounds for cubic spline interpolation," *Journal of Approximation Theory*, vol. 16, no. 2, pp. 105–122, 1976.

[18] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, Oct 2005, pp. 1508–1515 Vol. 2.

[19] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2002.

[20] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International journal of computer vision*, vol. 56, no. 3, pp. 221–255, 2004.

[21] F. Wen, R. Ying, Z. Gong, and P. Liu, "Efficient algorithms for maximum consensus robust fitting," *IEEE Transactions on Robotics*, 2019.

[22] "Vicon motion systems ltd." https://www.vicon.com/.