



Article Marker-Based Multi-Sensor Fusion Indoor Localization System for Micro Air Vehicles

Boyang Xing¹, Quanmin Zhu², Feng Pan^{1,3} and Xiaoxue Feng^{1,*}

- ¹ School of Automation, Beijing Institute of Technology, Beijing 100081, China; golaced@163.com (B.X.); panfeng@bit.edu.cn (F.P.)
- ² Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England, Bristol BS16 1QY, UK; quan.zhu@uwe.ac.uk
- ³ Kunming-BIT Industry Technology Research Institute INC, Kunming 650106, China
- * Correspondence: fengxiaoxue@bit.edu.cn; Tel.: +86-010-6891-3261

Received: 3 April 2018; Accepted: 22 May 2018; Published: 25 May 2018



Abstract: A novel multi-sensor fusion indoor localization algorithm based on ArUco marker is designed in this paper. The proposed ArUco mapping algorithm can build and correct the map of markers online with Grubbs criterion and K-mean clustering, which avoids the map distortion due to lack of correction. Based on the conception of multi-sensor information fusion, the federated Kalman filter is utilized to synthesize the multi-source information from markers, optical flow, ultrasonic and the inertial sensor, which can obtain a continuous localization result and effectively reduce the position drift due to the long-term loss of markers in pure marker localization. The proposed algorithm can be easily implemented in a hardware of one Raspberry Pi Zero and two STM32 micro controllers produced by STMicroelectronics (Geneva, Switzerland). Thus, a small-size and low-cost marker-based localization system is presented. The experimental results show that the speed estimation result of the proposed system is better than Px4flow, and it has the centimeter accuracy of mapping and positioning. The presented system not only gives satisfying localization precision, but also has the potential to expand other sensors (such as visual odometry, ultra wideband (UWB) beacon and lidar) to further improve the localization performance. The proposed system can be reliably employed in Micro Aerial Vehicle (MAV) visual localization and robotics control.

Keywords: indoor localization; ArUco marker; federated filter; Micro Aerial Vehicle

1. Introduction

Global position system (GPS) is the main localization system of Micro Aerial Vehicle (MAV) in an outdoor environment, which has the advantages of low-cost and high-precision [1]. By combining GPS with the inertial measurement unit (IMU), the GPS/IMU localization system can obtain the meter-level accuracy [2]. However, GPS cannot be used in the indoor environment [3] because of the signal shading, obstacles and environmental disturbances. Moreover, the localization precision and reliability for the indoor localization system are much more essential. Generally, the indoor localization system can be divided into non-cooperative localization technology and cooperative localization technology based on whether alternative sensors are introduced to provide additional information.

In the non-cooperative localization type, simultaneous localization and mapping (SLAM) is the most important technology, which includes lidar SLAM(L-SLAM) and visual SLAM (V-SLAM). The L-SLAM technology is very mature on unmanned ground vehicle (UGV), which has the advantages of high speed and reliability since the lidar can measure the precise distance and angle of the obstacles. Furthermore, the filters such as extended Kalman filter (EKF) [4] or particle filter (PF) [5] are used to obtain the accurate localization. L-SLAM has been applied in MAV system in recent years gradually. In literature [6], the reliable L-SLAM is realized on MAV system based on miniature 3D-lidar, and the proposed system can locate the MAV in unknown environments without GPS. Some similar systems are designed based on low-cost 2D-lidar since it is much cheaper and lighter. However, its low refresh frequency results in the matching error during the attitude change and high speed flight. V-SLAM is based on mono camera, RGB-D camera and stereo camera, which uses the graph optimization algorithm to minimize the projection error of key-points. Several V-SLAM systems have been proposed in recent years such as Semi-direct visual odometry (SVO) [7] and oriented FAST and rotated brief SLAM (ORB-SLAM) [8]. These two strategies have been applied in many MAV systems, which can realize precise localization based on the high-performance image process unit. However, the robustness issue is an inevitable problem of these pure vision-based methods, due to the fact that image tracking easily fails in the environments with poor visual features. Therefore, research through fusing V-SLAM with other sensors to achieve robust robot navigation is highly demanded.

In the cooperative localization type, base station, motion capture system and 2D markers are utilized as additional sensors. The requirement for on-board sensor and processor is much lower than V-SLAM which makes it suitable for the localization in a particular area. Traditional base station cooperative localization technology uses several signal stations to locate the beacon attached on the target via wireless fidelity (WiFi), ZigBee or radio frequency identification which is very susceptible to the signal multi-path interference, and can only provide meter localization accuracy [9]. Ultra wideband (UWB) is a short-range radio technology, which has been used for indoor localization in recent years. In contrast to ZigBee or WiFi, the localization via UWB is done with the time of flight methodology so it has good anti-interference capability and centimetre localization accuracy. In reference [10], the location information of UWB is introduced in the GPS/IMU navigation system which enhances the reliability of the integrated navigation in GPS inapplicable environment. However, UWB is still affected by the metal shielding and absorption. Meanwhile, the cost of it is much higher than other base station cooperative localization technology. Motion capture cooperative localization technology has high reliability and localization accuracy, which is widely used by movie-making, computer vision and control. For example, The Vicon MX motion capture system is a popular high end commercial solution for the development of autonomous aerial robotic [11]. However, the high cost of the motion capture system limits it only in adaptive industrial or laboratory environments; 2D markers cooperative localization technology is typically introduced in an environment where localization and navigation are needed for robots using low cost machine vision. It can provide accurate six-degrees-of-freedom information of the camera. Meanwhile, it is more reliable and faster than V-SLAM since the markers can be easily identified from a wider range of viewpoints. Therefore, 2D marker array can be used in featureless indoor environments (such as laboratory or corridors) to build a large-scale and cost-effective localization system. For example, the Sky-Trax system uses hundreds of 2D markers to locate the position of materials and equipments inside warehouses and factory [12].

In Sky-Trax, the 2D marker array is regularly arranged so it is easy to artificially calibrate the position of each marker, but in some cases the markers are irregular, which makes accurate calibration very difficult. Therefore, the automatic mapping of the large-scale 2D marker array is very important in practical application. A straightforward way is to implement the SLAM framework as in the literature [13]. In literature [14,15], the authors use odometer/IMU to estimate the robot's trajectory. Once the camera trajectory is accurately obtained, marker locations are obtained by triangulation. In literature [16] a graph is used to describe the geometric relationship of each marker, which is updated dynamically. Whenever a pair of markers are seen in a frame, their relative position is updated and if it is better than the previous one, it is replaced. For localization, they only consider for localization one marker from all visible ones. However, using all visible markers at the same time can lead to a better localization result. Literature [17] designed a simple marker-based localization system, and the markers are added into the map gradually through the Tf package in robot operate system (ROS) [18]. Although the system can realize real-time online map building, it has a serious problem of mapping distortion due to lack of correction. Literature [19] presents a monocular visual-inertial

EKF-SLAM system based on 2D markers. Since only 2D markers and IMU are used, drift may occur if the markers are lost for a long time. Moreover, it requires increasing computational resources to handle covariance matrices that are expanded as more markers are added. Literature [20] proposes a localization method considering all visible markers and handling the mapping as a variant of the sparse bundle adjustment problem. The proposed system can generate an accurate map, but the algorithm runs off-line and requires a large amount of computation.

In this paper, a multi-sensor fusion indoor localization system (MMFL) based on ArUco marker is proposed. The proposed ArUco Mapping algorithm can build and correct the map online with Grubbs criterion and K-mean clustering algorithm, which avoids the map distortion due to lack of correction. In order to realize the localization in marker undetectable situation, the federated Kalman filter with adaptive information coefficient is utilized to synthesize multi-source information from marker, optical flow, ultrasonic and the inertial sensor. Thus, the MMFL system can effectively reduce the position drift due to the long-term loss of markers. The presented system not only gives satisfying localization performance, but also has the potential to expand other localization sensors (such as visual odometry, UWB, and lidar) to further improve the localization performance. The on-board MMFL system can be easily realized on a low cost and low power consumption hardware platform including a Raspberry Pi Zero and two STM32 micro controllers produced by STMicroelectronics. The experimental results show that the speed estimation result is better than that of the Px4flow sensor [21], and the proposed system has the centimeter level errors in positioning and mapping to the manual calibration map.

The rest of this paper is organized as follows. Section 2 introduces the core modules in the proposed MMFL system, including the mapping module and speed estimation module described in Section 2.1, the attitude estimation module described in Section 2.2, and the multi-sensor fusion module described in Section 2.3. Section 3 introduces the hardware design and experimental results of the MMFL system. The experiment results of the proposed mapping algorithm is stated in Section 3.1, the experiment result of the multi-sensor fusion algorithm is stated in Section 3.2, the comparison experiment with ORB-SLAM2 is stated in Section 3.3 and the experiments of comparing the hovering performance with Px4flow sensor is stated in Section 3.4.

2. Multi-Sensor Fusion Indoor Localization System Based on ArUco Marker

Generally, the ArUco marker can provide precise localization information with less CPU consumption. However, the tilting and shaking of MAV may cause localization errors, discontinuous positioning, even miss detecting since the camera is mounted on it. Figure 1 shows the cases of abnormal detection caused by shaking and reflecting light, in which most markers can not be detected in these situations and the vertices of the identified markers also have a position error. Therefore, it is straightforward to consider introducing additional sensors to get a satisfying localization result. Thus, a multi-sensor fusion structure combining markers, IMU, ultrasonic and optical flow by the federated Kalman filter is proposed in order to decrease the effect of strong illumination and low quality images.

The flow chart of the MMFL system proposed in this paper is shown in Figure 2. The sensor layer is composed of a variety of heterogenous sensors including markers, IMU, ultrasonic and camera. It is possible to further improve the performance by adding other sensors (such as UWB beacon, barometer or lidar). In the sensor registration layer, the proposed ArUco Mapping module uses an online mapping algorithm to realize the camera localization, the attitude estimation of MAV is based on the EKF [22], and the pyramid Lucas-Kanade (PLK) optical flow algorithm is utilized to estimate the translational speed of camera. The information fusion layer is realized by a federated Kalman filter (FK) [23], which contains two sub-filters (SF) and one main filter (MF). The camera position and body acceleration are input SF-I to estimate the position and speed of MAV. The translational speed estimated by optical flow, the height measured by ultrasonic and the body acceleration are input SF-II to estimate the SF with the same statement. Finally, the MF synthesizes two SFs' results for an optimal estimation.



Figure 1. Abnormal Detection Caused by Camera Shake and Reflection.



Figure 2. Flow Chart of Multi-Heterogeneous Sensors Layer.

2.1. Visual Algorithms in Sensor Registration Layer

ArUco Mapping: containing 1024 different modified Hamming codes, the ArUco marker [24] is capable of providing detection, recognition and six-degree-of-freedom (6Dof) information of camera, which is used to give the precise position of MAV in this paper. During location of a large area, a precise map containing the poses of markers is required. The direct way to build this map is to manually calibrate each marker. Obviously, manual mapping is extremely difficult and impossible for an irregular marker array, which is time-consuming and error-prone when calibrating hundreds of markers. A more appropriate way is to automatically build the map with the visual information. Based on this idea, the proposed ArUco Mapping algorithm is described as below.

Define the marker coordinate as *m*, whose origin locates at the center of marker, the *y*-axis points in the forward direction, the *z*-axis goes through the center and the *x*-axis points to the right direction. Define the map coordinate as *n*, which is coincided with the coordinate of the first detected marker. The state vectors of Marker-*i* in coordinate *n* include the center position $\mathbf{p}_i^n = (mx_i^n, my_i^n, mz_i^n)$ and orientation γ_i^n in radian. For the markers are set on the flat floor in this paper so the mz_i^n of all markers are assumed to be zero. In addition, the camera's position and orientation calculated by Marker-*i* in coordinate *m* is $\mathbf{p}_i^m = (x_i^m, y_i^m, z_i^m)$ and γ_i^m , the camera's position and orientation calculated by Marker-*i* in coordinate *n* is $\mathbf{P}_i^n = (x_i^n, y_i^n, z_i^n), \gamma_i^n$.

$$\mathbf{Y}_i^n = \boldsymbol{\gamma}_i^n + \boldsymbol{\gamma}_i^m \tag{1}$$

$$\mathbf{P}_{i}^{n} = \mathbf{p}_{i}^{n} + \mathbf{R}(\gamma_{i}^{n})\mathbf{p}_{i}^{m}$$
⁽²⁾

with
$$\mathbf{R}(\gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
.

It is assumed that the position and orientation of the MAV in map coordinate calculated by each marker are assumed to be same, thus the \mathbf{P}^n and \mathbf{Y}^n calculated by two markers should be same in this case. The newly detected Marker-*j*'s position \mathbf{p}_j^n and orientation γ_j^n in coordinate *n* can be calculated by Equations (3)–(6) based on the position and orientation of Marker-*i*.

$$Y_{i,j}^n = \gamma_i^n + \gamma_i^m = \gamma_j^n + \gamma_j^m$$
(3)

$$\gamma_j^n = \gamma_i^n + \gamma_i^m - \gamma_j^m \tag{4}$$

$$\mathbf{P}_{i,j}^{n} = \mathbf{p}_{i}^{n} + \mathbf{R}(\gamma_{i}^{n})\mathbf{p}_{i}^{m} = \mathbf{p}_{j}^{n} + \mathbf{R}(\gamma_{j}^{n})\mathbf{p}_{j}^{m}$$
(5)

$$\mathbf{p}_{j}^{n} = \mathbf{p}_{i}^{n} + \mathbf{R}(\gamma_{i}^{n})\mathbf{p}_{i}^{m} - \mathbf{R}(\gamma_{j}^{n})\mathbf{p}_{j}^{m}$$
(6)

The presented mapping algorithm uses the first detected marker to initialize the map *M* then the newly detected marker is added into the map gradually. In addition, the position and orientation of each marker is calculated and restored in its own point cloud ζ . The Grubbs criterion [25] is used to remove the outliers from point cloud when its size is bigger than *S*, then the *K*-mean cluster [26] is adopted to find the center of *K* clusters in the point cloud. The most likely center **P**^{*}, Y^{*} synthesize each clusters' center with the corresponding weight, which is determined by the degree of data concentration. **P**^{*}, Y^{*} are used to correct the markers' position **Pc** and orientation **Yc** in map *M* based on step β . After each map correction, a point will be removed from its point cloud if the pose error is bigger than *R* and a marker will be "fixed" if the pose error between two revisions is smaller than *F*. Finally, the ArUco SDK is used to calculate the camera's position **Pc** and orientation **Yc** based on the map. The pseudo-code of ArUco Mapping (AM) is given in the following Algorithm 1.

Pyramid Lucas-Kanade Optical Flow: optical flow is the velocity vector caused by camera movement, which is used to estimate the speed of MAV in this paper. The traditional method for measuring optical flow is the Lucas-Kanade method [27], which assumes that the image displacement between two adjacent frames is small. Therefore, the disadvantage of this method is that the matched pixel candidates cannot be found in a small neighborhood during fast motion. A common improvement of the LK method is to calculate the optical flow with pyramids, which is known as the PLK method [28]. The PLK method compresses the size of the image on different pyramid levels, so that the optical flow caused by fast motion can be calculated on the top level. For this reason, the PLK method is able to estimate the optical flow during fast motion. The PLK method at level *L* and \mathbf{g}^L is the predicted optical flow based on the upper level. Thus the updated optical flow of level *L* is $\mathbf{k}^L = \mathbf{g}^L + \mathbf{d}^L$, and the predicted optical flow of level *L* is $\mathbf{g}^{L+1} = 2 \times \mathbf{k}^L$. Through repeating this method to the lowest level of the pyramids, the final optical flow can be calculated.

Algorithm 1 Pseudo-Code of the ArUco Mapping Algorithm

Input: M: current map, ζ: point cloud, Π: set of detected marks, S: size of point cloud, β: correction step, L: number of markers in map, N: number of detected markers, K: number of cluster, R: culling threshold of point cloud, F: marker fixed threshold
 Output: Pm: markers' position in map, Ym: markers' orientation in map, Pc: camera's position, Yc: camera's orientation;

1: Map initialize: initialize the map coordinate by the firstly detected marker 2: if $\dot{N} > 0$ then 3: **for** i = 0 to N in Π **do** 4: if $\Pi(i)$ not in *M* then 5: for j = 1 to N do 6: if $\Pi(j)$ in *M* and i! = j then 7: *Add the newly detected Marker in to map:* Calculate \mathbf{P}_{i}^{n} and γ_{i}^{n} by Equations (3)–(6) and initialize $\zeta(i)$; 8: end if 9: end for 10: else for j = 1 to N do 11: 12: if $\Pi(j)$ in M and i! = j then 13. **Point cloud sampling:** Calculate \mathbf{P}_{i}^{n} and γ_{i}^{n} by Equations (3)–(6) and add it into $\zeta(i)$; 14: end if 15: end for 16: 17. *Remove outliers:* $\zeta^*(i) = \text{Grubbs}(\zeta(i));$ 18: 19: Find the center of the point cloud: The center of each cluster: $C_n^p, C_n^Y(n = 1, 2, ..., K) = K\text{-means}(\zeta^*(i), K);$ 20: 21: for n = 1 to K do 22: Calculate the lumpiness δ_n^p , δ_n^p of C_n^p , C_n^p , based on the average distance between all points and the centers; 23: end for Find the maximum value ℓ^p , ℓ^Y of lumpiness δ_n^p , δ_n^Y (n = 1, 2, ..., K); 24: Calculate the final center and orientation of the point cloud $\mathbf{P}^*(i) = \sum_{n=1}^{K} \frac{\ell^p / \delta_n^p}{\sum_{m=1}^{K} \ell^p / \delta_m^p} \mathbf{C}_n^p, \mathbf{Y}^*(i) = \sum_{n=1}^{K} \frac{\ell^Y / \delta_n^N}{\sum_{m=1}^{K} \ell^Y / \delta_m^N} \mathbf{C}_n^p;$ 25: 26: 27: *Map correction:* Use $\mathbf{P}^*(i)$, $\mathbf{Y}^*(i)$ to modify $\mathbf{Pm}(i)$, $\mathbf{Ym}(i)$ with step β ; 28: 29: Point cloud clipping: 30: for k = 1 to all points in $\zeta(i)$ do 31: if $\|\zeta(i)_k - \mathbf{Pm}(i)\|_{L^2} > R$ then 32: Remove this point from $\zeta(i)$; 33: end if 34: end for 35: 36: Fix marker: 37: if $\|\mathbf{P}^*(i) - \mathbf{Pm}(i)\|_{L2} < F$ then 38: Fix this marker in map and do not correct it later; 39: end if 40: end if 41: end for 42: Calculate Camera position: 43: for i = 1 to all Π in M do 44: 45. Calculate \mathbf{P}_{i}^{n} and Y_{i}^{n} by Equations (1) and (2); 46: end for 47: calculate the average value of \mathbf{P}_i^n (i = 1, 2, ..., N) as $\mathbf{Pc} = (P_x^c, P_y^c, P_z^c)$, calculate the average value of Y_i^n (i = 1, 2, ..., N) as Yc 48: end if

Since the camera faces the ground, the rotation of pitch, roll will generate the rotating optical flow, which can be compensated with IMU to reflect the real translational speed. Define the body coordinate as *b*, which is a non-inertial coordinate system associated with the vehicle's center of gravity. The *x*-axis points in the forward direction, the *z*-axis down through the vehicle and the *y*-axis completes the right-hand coordinate system. Set d_x , d_y as the average of the final optical flow calculated by the PLK method, the compensated optical flow Dx, Dy can be calculated as below.

$$D_x = d_x - w_x^b W_x / \mu \tag{7}$$

$$D_y = d_y - w_y^b W_y / \mu \tag{8}$$

where w_x^b and w_y^b are the angular speeds measured by the gyroscope. W_x is the number of image rows, W_y is the number of image columns, μ is the field of view and the translational speed of camera is given as below:

$$v_x = 2hD_x/W_x \tan(\mu/2) \tag{9}$$

$$v_y = 2hD_y/W_y \tan(\mu/2) \tag{10}$$

where h is the distance to the ground level given by the ultrasonic sensor.

2.2. Attitude Estimation Module in Sensor Registration Layer

Figure 3 shows the flow chart of the attitude estimation module in sensor registration layer, and an IMU includes 3-axis gyroscope, 3-axis acceleration meter and 3-axis magneto meter is used here. $\mathbf{w}^{b}(k) = [w_{x}^{b}(k), w_{y}^{b}(k), w_{z}^{b}(k)]$ is the angular speed measured by gyroscope, $\mathbf{a}^{b}(k) = [a_{x}^{b}(k), a_{y}^{b}(k), a_{z}^{b}(k)]$ is the body acceleration measured by acceleration meter, $\mathbf{m}^{b}(k) = [m_{x}^{b}(k), m_{y}^{b}(k), m_{z}^{b}(k)]$ is magnetic field measured by magnetometer, $\boldsymbol{\xi}^{n}(k) = [\hat{\phi}^{n}(k), \hat{\theta}^{n}(k), \hat{\psi}^{n}(k), \mathbf{a}^{n}(k)]^{T}$ is the estimated attitude and acceleration in map coordinate *n*.



Figure 3. Flow Chart of Attitude Estimation Module.

State Model: the attitude estimation is based on the quaternion [29] and EKF. The state vector $\mathbf{x}(k) = [\mathbf{q}(k), \mathbf{b}(k)]^T = [q_0(k), q_1(k), q_2(k), q_3(k), b_x(k), b_y(k), b_z(k)]^T$ consists of two parts: quaternion and gyro bias, and the system function is given below:

$$\mathbf{x}(k) = f(\mathbf{x}(k-1), k-1) = \begin{bmatrix} (\mathbf{I} + (T/2)\mathbf{\Omega}_w(k-1))\mathbf{q}(k-1) \\ \mathbf{b}(k-1) \end{bmatrix} + \sigma(k-1)$$
(11)

where
$$\mathbf{\Omega}_{w}(k-1) = \begin{bmatrix} 0 & -w_{x}^{b}(k) + b_{x}(k-1) & -w_{y}^{b}(k) + b_{y}(k-1) & -w_{z}^{b}(k) + b_{z}(k-1) \\ w_{x}^{b}(k) - b_{x}(k-1) & 0 & w_{z}^{b}(k) - b_{z}(k-1) & -w_{y}^{b}(k) + b_{y}(k-1) \\ w_{y}^{b}(k) - b_{y}(k-1) & -w_{z}^{b}(k) + b_{z}(k-1) & 0 & w_{x}^{b}(k) - b_{x}(k-1) \\ w_{z}^{b}(k) - b_{z}(k-1) & w_{y}^{b}(k) - b_{y}(k-1) & -w_{x}^{b}(k) + b_{x}(k-1) & 0 \end{bmatrix}$$

T is the sample period, **I** is the identity matrix of appropriate dimension, $\sigma(k)$ is gaussian white noise of system and $E[\sigma(k-1)\sigma(k-1)^T] = Q(k-1)$.

Measurement Model: the measurement $\mathbf{z}(k) = [a_x^b(k), a_y^b(k), a_z^b(k), \psi^n(k) - b_{\psi}(k)]^T$ includes the body acceleration and yaw in map coordinate, $\psi^n(k)$ is the yaw in earth coordinate and it can be calculated by the measurement of magneto meter:

$$\psi^n(k) = \arctan(m_y^n(k)/m_x^n(k)) \tag{12}$$

$$m_x^n(k) = m_x^b(k)\cos\hat{\theta}(k) + m_y^b(k)\sin\hat{\phi}(k)\sin\hat{\theta}(k) - m_z^b\cos\hat{\phi}(k)\sin\hat{\theta}(k)$$
(13)

$$m_y^n(k) = m_y^b(k) \cos\hat{\phi}(k) + m_z^b(k) \sin\hat{\phi}(k)$$
(14)

In addition, there is a bias between map coordinate and earth coordinate since the map coordinate is initialized by the first detected marker. Define this bias as $b_{\psi}(k)$ and it can be calculated as below:

$$b_{\psi}(k) = \begin{cases} \psi^{n}(k) - Yc(k) & \text{If any markers are detected in the image} \\ b_{\psi}(k-1) & \text{Otherwise} \end{cases}$$

Therefore, the measurement function is given below:

$$\mathbf{z}(k) = h(\mathbf{x}(k), k) + \mathbf{v}(k) = \begin{bmatrix} -2g(q_1(k)q_3(k) - q_0(k)q_2(k)) \\ -2g(q_2(k)q_3(k) + q_0(k)q_1(k)) \\ -g(q_0(k)^2 - q_1(k)^2 - q_2(k)^2 + q_3(k)^2) \\ \arctan(-\frac{2(q_1(k)q_2(k) - q_0(k)q_3(k))}{q_0(k)^2 - q_1(k)^2 + q_2(k)^2 - q_3(k)^2}) \end{bmatrix} + \mathbf{v}(k)$$
(15)

where $\mathbf{v}(k)$ is the measurement noise and $E[\mathbf{v}(k)\mathbf{v}(k)^T] = R(k)$.

Attitude Estimation: the extended Kalman filter equations of the attitude estimation module are given below.

$$\hat{\mathbf{x}}(k,k-1) = f(\hat{\mathbf{x}}(k-1),k-1)$$
(16)

$$\mathbf{P}(k,k-1) = \nabla f_{\mathbf{x}(k-1)} \mathbf{P}(k-1) \nabla f_{\mathbf{x}(k-1)}^T + \mathbf{Q}(k)$$
(17)

$$\mathbf{K}(k) = \mathbf{P}(k,k-1)\nabla h_{\mathbf{x}(k-1)}^{T}(\nabla h_{\mathbf{x}(k-1)}\mathbf{P}(k,k-1)\nabla h_{\mathbf{x}(k-1)}^{T} + \mathbf{R}(k))^{-1}$$
(18)

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k, k-1) + \mathbf{K}(k)[\mathbf{z}(k) - h(\hat{\mathbf{x}}(k, k-1), k)]$$
(19)

$$\mathbf{P}(k) = (\mathbf{I} - \mathbf{K}(k)\nabla h_{\mathbf{x}(k-1)})\mathbf{P}(k, k-1)$$
(20)

where $\mathbf{P}(k)$ is the covariance matrix and $\mathbf{K}(k)$ is the Kalman gain matrix, $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ are the system and measurement covariance matrices respectively. As the appropriate matrices $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ cannot be chosen based on classical theories, they are usually tuned experimentally by a trial-and-error method. The Jacobi matrix of the nonlinear system function and measurement function are given below.

$$\nabla f_{\mathbf{x}(k)} = \begin{bmatrix} 1 & -T\hat{w}_{x}(k)/2 & -T\hat{w}_{y}(k)/2 & -T\hat{w}_{z}(k)/2 & Tq_{1}(k)/2 & Tq_{2}(k/2 & Tq_{3}(k)/2 \\ T\hat{w}_{x}(k)/2 & 1 & T\hat{w}_{z}(k)/2 & -T\hat{w}_{y}(k)/2 & -Tq_{0}(k)/2 & Tq_{3}(k)/2 & -Tq_{2}(k)/2 \\ T\hat{w}_{y}(k)/2 & -T\hat{w}_{z}(k)/2 & 1 & T\hat{w}_{x}(k)/2 & -Tq_{0}(k)/2 & Tq_{1}(k)/2 \\ T\hat{w}_{z}(k)/2 & T\hat{w}_{y}(k)/2 & -T\hat{w}_{x}(k)/2 & 1 & Tq_{2}(k)/2 & -Tq_{0}(k)/2 & -Tq_{0}(k)/2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$
(21)
$$\nabla h_{\mathbf{x}(k)} = \begin{bmatrix} 2gq_{2}(k) & -2gq_{3}(k) & 2gq_{0}(k) & -2gq_{1}(k) & 0 & 0 & 0 \\ -2gq_{1}(k) & -2gq_{0}(k) & -2gq_{3}(k) & -2gq_{2}(k) & 0 & 0 & 0 \\ -2gq_{0}(k) & 2gq_{1}(k) & 2gq_{2}(k) & 2gq_{3}(k) & 0 & 0 & 0 \\ \frac{2q_{3}(k)S_{1}+4q_{0}(k)S_{2}}{S_{1}^{2}+4S_{2}^{2}} & \frac{-2q_{2}(k)S_{1}-4q_{1}(k)S_{2}}{S_{1}^{2}+4S_{2}^{2}} & \frac{2q_{0}(k)S_{1}-4q_{3}(k)S_{2}}{S_{1}^{2}+4S_{2}^{2}} & 0 & 0 & 0 \end{bmatrix}$$
(22)

Sensors 2018, 18, 1706

where $\hat{\mathbf{w}}(k) = \begin{bmatrix} \hat{w}_x(k) \\ \hat{w}_y(k) \\ \hat{w}_z(k) \end{bmatrix} = \begin{bmatrix} w_x^b(k) - b_x(k) \\ w_y^b(k) - b_y(k) \\ w_z^b(k) - b_z(k) \end{bmatrix}$ is the angular velocity after compensated the gryoscope-drift, $S_1 = q_0(k)^2 - q_1(k)^2 + q_2(k)^2 - q_3(k)^2$, $S_2 = q_1(k)q_2(k) - q_0(k)q_3(k)$ and g is the gravity.

Normalization: the quaternion needs to be normalized at the end of each filter cycle. So the final output of the filter is given as below:

$$q_i(k) = q_i(k) / \sqrt{q_0(k)^2 + q_1(k)^2 + q_2(k)^2 + q_3(k)^2}, (i = 0, 1, 2, 3)$$
(23)

$$\boldsymbol{\xi}^{n}(k) = \begin{bmatrix} \hat{\theta}^{n}(k) \\ \hat{\varphi}^{n}(k) \\ \hat{\psi}^{n}(k) \\ \mathbf{a}^{n}(k) \end{bmatrix} = \begin{bmatrix} \arcsin(2(q_{2}(k)q_{3}(k) + q_{0}(k)q_{1}(k))) \\ \arctan(\frac{2(q_{1}(k)q_{3}(k) - q_{0}(k)q_{2}(k))}{q_{0}(k)^{2} - q_{1}(k)^{2} - q_{2}(k)^{2} + q_{3}(k)^{2}}) \\ \arctan(-\frac{2(q_{1}(k)q_{2}(k) - q_{0}(k)q_{3}(k))}{q_{0}(k)^{2} - q_{1}(k)^{2} + q_{2}(k)^{2} - q_{3}(k)^{2}}) \\ \end{bmatrix}$$
(24)

where

$$\mathbf{C}_{b}^{n} = \begin{bmatrix} q_{0}(k)^{2} + q_{1}(k)^{2} - q_{2}(k)^{2} - q_{3}(k)^{2} & 2(q_{1}(k)q_{2}(k) - q_{0}(k)q_{3}(k)) & 2(q_{1}(k)q_{3}(k) + q_{0}(k)q_{2}(k)) \\ 2(q_{1}(k)q_{2}(k) + q_{0}(k)q_{3}(k)) & q_{0}(k)^{2} - q_{1}(k)^{2} + q_{2}(k)^{2} - q_{3}(k)^{2} & 2(q_{2}(k)q_{3}(k) + q_{0}(k)q_{1}(k)) \\ 2(q_{1}(k)q_{3}(k) - q_{0}(k)q_{2}(k)) & 2(q_{2}(k)q_{3}(k) - q_{0}(k)q_{1}(k)) & q_{0}(k)^{2} - q_{1}(k)^{2} - q_{2}(k)^{2} + q_{3}(k)^{2} \end{bmatrix}$$
(25)

2.3. Information Fusion of Multi-Heterogeneous Sensors

The information fusion layer is designed to fuse the information received from the upper layers. A federated filter is used here based on the advantage of fusing heterogeneous sensors data, which allows the sensor layer to measure the same states with different sensors (for example ultrasonic and ArUco Mapping can both give the height of MAV). Additionally, it is easy to expand other sensors to further improve the localization precision and provide a good redundancy. The federated filter is comprised of one main filter (MF) and several sub-filters (SF), which is fault tolerant and flexible. The state and covariance matrix of each sub-filter are sent to the MF and the MF fuses the two estimations to get the optimal result.

State Model: A distributed system with controlled input and noise is given as follows:

$$\mathbf{X}_{g}(k) = \mathbf{A}\mathbf{X}_{g}(k-1) + \mathbf{B}\mathbf{U}(k-1) + \mathbf{\Gamma}\mathbf{W}(k-1)$$
(26)

$$\mathbf{A} = \begin{bmatrix} \mathbf{F} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{F} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{F} \end{bmatrix}$$
(27)

$$\mathbf{B} = \begin{bmatrix} \mathbf{G} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} & \mathbf{G} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{G} \end{bmatrix}$$
(28)

$$\mathbf{U}(k-1) = \mathbf{a}^n(k-1) \tag{29}$$

with $\mathbf{F} = \begin{bmatrix} 1 & T & -T^2/2 \\ 0 & T & -T \\ 0 & 0 & 1 \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} T^2/2 \\ T \\ 0 \end{bmatrix}$, where T is the sample period, A is

the state-transition matrix, **B** is the controlled input matrix, $\mathbf{0}_{3\times3}$ is the zero matrix with the appropriate dimension, $\mathbf{U}(k-1)$ is the controlled input, $\mathbf{\Gamma}$ is the noise drive matrix, $\mathbf{W}(k-1)$ is the gaussian white noise and $E[\mathbf{W}(k-1)\mathbf{W}(k-1)^T] = \mathbf{Q}(k-1)$. The state vector

matrix is $\mathbf{R}_2(k)$.

 $\mathbf{X}_{g}(k) = [x(k), \dot{x}(k), \tau_{x}(k), y(k), \dot{y}(k), \tau_{y}(k), z(k), \dot{z}(k), \tau_{z}(k)]^{T}$ includes position, speed, and acceleration bias in the map coordinate.

Measurement Model: The measurement of SF-I is the camera position $Pc(\mathbf{k}) = (P_x^c(k), P_y^c(k), P_z^c(k))$, and the measurement function is:

$$\mathbf{z}_{1}(k) = \begin{bmatrix} P_{x}^{c}(k) & 0 & 0 & P_{y}^{c}(k) & 0 & 0 & P_{z}^{c}(k) & 0 & 0 \end{bmatrix}^{T} = \mathbf{H}_{1}\mathbf{X}_{1}(k) + \mathbf{n}_{1}(k)$$
(30)

where $\mathbf{H}_1 = \text{diag}(1, 0, 0, 1, 0, 0, 1, 0, 0)$, $\mathbf{X}_1(k)$ is the state vector of SF-I, $\mathbf{n}_1(k)$ is the measurement noise for marker, and its covariance matrix is $\mathbf{R}_1(k)$.

The measurement of SF-II contains the speed estimated by PLK and the height given by ultrasonic. The measurement function is:

$$\mathbf{z}_{2}(k) = \begin{bmatrix} 0 & V_{\mathbf{x}}(k) & 0 & 0 & V_{y}(k) & 0 & \frac{h(k)}{\cos\hat{\varphi}^{n}(k)\cos\hat{\theta}^{n}(k)} & 0 & 0 \end{bmatrix}^{T} = \mathbf{H}_{2}\mathbf{X}_{2}(k) + \mathbf{n}_{2}(k)$$
(31)

where $\begin{bmatrix} V_{\mathbf{x}}(k) \\ V_{\mathbf{y}}(k) \end{bmatrix} = \begin{bmatrix} \cos \hat{\psi}^n(k) & \sin \hat{\psi}^n(k) \\ -\sin \hat{\psi}^n(k) & \cos \hat{\psi}^n(k) \end{bmatrix} \begin{bmatrix} v_{\mathbf{x}}(k) \\ v_{\mathbf{y}}(k) \end{bmatrix}$, $\mathbf{H}_2 = \operatorname{diag}(0, 1, 0, 0, 1, 0, 1, 0, 0)$, $\mathbf{X}_2(k)$ is the state vector of SF-II, $\mathbf{n}_2(k)$ is the measure noise vector for PLK and ultrasonic, and its covariance

Information Fusion: the standard Kalman filter algorithm is used in the presented MMFL system since the state and measurement functions are both linear, and the federated Kalman filter algorithm [23] is given as below:

(1) The MF initializes $\hat{\mathbf{X}}_g(k-1)$, $\mathbf{P}_g(k-1)$ and $\mathbf{Q}_g(k-1)$, then uses the positive coefficient $\beta_i(k-1)$ to distribute the information as follows:

$$\hat{\mathbf{X}}_i(k-1) = \hat{\mathbf{X}}_g(k-1) \tag{32}$$

$$\mathbf{P}_{i}(k-1) = \beta_{i}^{-1}(k-1)\mathbf{P}_{g}(k-1)$$
(33)

$$\mathbf{Q}_{i}(k-1) = \beta_{i}^{-1}(k-1)\mathbf{Q}_{g}(k-1)$$
(34)

where $\hat{\mathbf{X}}_i(k)(i = 1, 2)$ represents the estimated state vector of each sub-filter, $\mathbf{P}_i(k-1)$ and $\mathbf{Q}_i(k-1)(i = 1, 2)$ represent the corresponding estimated state covariance and process noise covariance, respectively. The coefficient is subject to $\beta_i(k-1) > 0$ and $\sum \beta_i(k-1) = 1$.

According to Equations (32)–(34) the federated filter allocates the global estimation to each SF based on $\beta(k-1)$, and a smaller information coefficient means less use of the global estimation. For example, the global estimation is average allocated between each SF when $\beta_i(k-1) = 0.5(i = 1, 2)$ [30]. To make the filter more adaptive, the presented algorithm adopts the variance matrix's Frobeniuse norm of each SF as the distribution coefficient since the fixed information coefficients can not reflect the accuracy change of SFs. Therefore, a better accurate SF can allocate a larger information coefficient and have a great contribution to the global estimation [31].

$$\beta_i(k) = \frac{(\|\mathbf{P}_i(k-1)\|_F)^{-1}}{(\|\mathbf{P}_1(k-1)\|_F)^{-1} + (\|\mathbf{P}_2(k-1)\|_F)^{-1}}$$

where $\|\boldsymbol{\chi}\|_F = \sqrt{\sum diag(\boldsymbol{\chi}^T \boldsymbol{\chi})}.$

(2) The state prediction of SF-I and SF-II is implemented independently.

$$\hat{\mathbf{X}}_i(k,k-1) = \mathbf{A}\hat{\mathbf{X}}_i(k-1) + \mathbf{B}\mathbf{U}(k-1)$$
(35)

$$\hat{\mathbf{P}}_i(k,k-1) = \mathbf{A}\mathbf{P}_i(k-1)\mathbf{A}^T + \mathbf{Q}_i(k-1)$$
(36)

$$\mathbf{K}_{i}(k) = \hat{\mathbf{P}}_{i}(k,k-1)\mathbf{H}_{i}^{T}(\mathbf{H}_{i}\hat{\mathbf{P}}_{i}(k,k-1)\mathbf{H}_{i}^{T} + \mathbf{R}_{i}(k))^{-1}$$
(37)

$$\hat{\mathbf{X}}_i(k) = \hat{\mathbf{X}}_i(k, k-1) + \mathbf{K}_i(k)(\mathbf{z}_i(k) - \mathbf{H}_i\hat{\mathbf{X}}_i(k, k-1))$$
(38)

$$\mathbf{P}_{i}(k) = (\mathbf{I} - \mathbf{K}_{i}(k)\mathbf{H}_{i})\hat{\mathbf{P}}_{i}(k, k-1)$$
(39)

(4) The MF fuses the estimations of the two SFs to obtain the optimal result.

$$\mathbf{P}_{g}(k) = (\mathbf{P}_{1}^{-1}(k) + \mathbf{P}_{2}^{-1}(k))^{-1}$$
(40)

$$\hat{\mathbf{X}}_{g}(k) = \mathbf{P}_{g}(k)(\mathbf{P}_{1}^{-1}(k)\hat{\mathbf{X}}_{1}(k) + \mathbf{P}_{2}^{-1}(k)\hat{\mathbf{X}}_{2}(k))$$
(41)

$$\mathbf{Q}_{g}(k) = (\mathbf{Q}_{1}^{-1}(k) + \mathbf{Q}_{2}^{-1}(k))^{-1}$$
(42)

A special case is that the markers cannot be detected when MAV is taking off, landing or flying at a low height. To make the filter work properly in those situations, the local estimation of SF-I will be blocked to MF. Thus, the global estimation is $\hat{\mathbf{X}}_g(k) = \hat{\mathbf{X}}_2(k)$, and the measurement prediction of SF-I is $\hat{\mathbf{z}}_1(k) = \mathbf{H}_1 \hat{\mathbf{X}}_g(k)$. Besides, the measurement delay to IMU can be handled by the ring-buffer scheme proposed in [32] for each SF.

3. Experiment Result and Analysis

In order to verify the proposed MMFL system, a 4×4 m experiment site is established in this paper. It contains 100 ArUco markers of 18×18 cm in a regular layout with increasing number. The marker array's center row spacing is 38 cm and the center column spacing is 38.5 cm. The experimental site is shown in Figure 4.



Figure 4. Experiment Site.

In the proposed MMFL system, one Raspberry Pi Zero and two STM32 micro controllers produced by STMicroelectronics are used to build the on-board module, which is smaller than 8 × 3 cm and lighter than 50 g. The detailed hardware structure is shown in Figure 5. In the proposed on-board module, the Raspberry Pi Zero captures the image from the Pi camera and implements ArUco Mapping at the frequency of 10 Hz. The STM32F4 captures the image from the OV5640 camera and implements speed estimation via PLK at the frequency of 20 Hz. All sensor acquisition (including one MPU9250 inertial sensor and one US-100 ultrasonic) and data fusion are completed in STM32F1 at the frequency of 100 Hz. The STM32F4 and Pi Zero are connected by serial port, and the Mavlink [33] protocol is adopted to guarantee the data transmission. In the software of STM32F1, each sensor has a flag for detecting whether its data is received and the sub-filter uses the measurement to update its state only if its flag is true. The commands and desired way points of MAV are transmitted from the ground station via 2.4 G radio and the real-time 6Dof information of MAV can be observed on the ground station.



Figure 5. On-board Module Structure Diagram.

3.1. Experiment Result of the ArUco Mapping Module

The experiment in this section is designed to test the performance of the ArUco Mapping (AM). Comparison experiments between the proposed ArUco Mapping in this paper and the mapping systems proposed in reference [18] (noted as Method 1), reference [20] (noted as Method 2) are performed. All the three candidate systems use the same video source as input. Besides, the ArUco SDK is used to calculate the camera position with the different maps generated by the three candidate mapping systems. The trajectory calculated by the manually calibrated map is adopted as the "benchmark" to evaluate the performance of the three candidate systems, since the marker array is regularly arranged and the calibrated error can be omitted. The comparison result is shown in Figure 6.

It can be seen from Figure 6 that the map of the experimental site can be successfully established by Method 2 and the proposed AM system. The mapping result of Method 1 has the catastrophic mistake due to the lack of correction in markers' pose. Compared with the manual calibrated map, Method 2 has the outstanding mapping performance, however it is running off-line, which takes 10 s to build and optimize the map (executing on IntelRAtom i5-4300U (1.9 GHz) CPU with Ubuntu operating system). Furthermore, two performance indexes are introduced to evaluate the performance of the three candidate mapping systems: estimate accuracy of the marker poses, and estimate accuracy of the camera position. The first one can be evaluated by calculating the Absolute Corner Error (ACE), which computes the mean error between the estimated marker corner locations and the manual calibrated ones. The estimate accuracy of the camera position can be obtained by calculating the Absolute Trajectory Error (ATE), which computes the mean error between the mean error between the estimated camera position and the benchmark one. The ACE and ATE comparison results of the three candidate methods are shown in Table 1. Although the mapping result of the proposed AM system has a little deviation and distortion compared with Method 2, but the map is built online and the computation is much less than Method 2.

The local enlarged results of markers 17–20, 27–30 and 37–40 in Figure 6d are given in Figure 7 for clear view. In Figure 8, the pink dotted square is the real position of each marker, the green bold square is the first center of each marker, the pink cross is the scattered point cloud of each marker and the black solid square is the final mapping result. It can be seen that most of the markers' positions are closer to the real position rather than its first center, so it can be concluded that the proposed algorithm is effective.

Table 1. Performance Comparison of the Three Candidate Systems.

Index	AM Proposed in This Paper	Method 1	Method 2
ACE	0.047 m	0.36 m	0.035 m
ATE	0.085 m	0.45 m	0.061 m



Figure 6. Result of Comparison Experiments. (a) Manual Calibrated Map; (b) Map of Method 2; (c) Map of Method 1; (d) Map of the Proposed System.



Figure 7. Local Enlarged Results of Figure 6d.



Figure 8. Mapping Error of Experiment Site. (a) Mapping Error; (b) Positioning Error.

In Figure 8a, the thin lines in the background is the corner location error of each marker, and the bold line with circle mark is the absolute corner error of the AM system which decreases with the correction times and converges to 5 cm after 12 corrections. It can be concluded that the proposed algorithm can build the map effectively and avoid the map distortion caused by the abnormal detection. Besides, the positioning error of the camera trajectory is shown in Figure 8b. The maximum error between camera estimation is 10 cm compared with the benchmark, which is satisfactory for the low cost indoor localization application. A video of the online mapping process can be seen at https://youtu.be/j6ZTGQrOdO0.

3.2. Experiment Results of Multi-Sensor Information Fusion

The proposed system utilizes the EKF algorithm to estimate the attitude. Figure 9 shows the attitude estimation result during one hovering experiment. The solid line in Figure 9 is the result of the proposed MMFL system and the dotted line is the calculated result based on the manual calibrated map. When the marker array is regularly arranged, the benchmark attitude is the average of all detected markers' results with the manual calibrated map. It is clear that the result of the solid line is smoother than that of the dotted line, the small gap in Yaw is caused by the orientation error of the first detected marker and the map distortion during online mapping. In addition, the proposed system assumes that all the markers are located on the ground, thus the errors of Pitch and Roll are avoided via the static calibration. Figure 10 is the speed fusion result. The blue point line with cross is the speed measured by Px4flow. The red point line with square is the speed measured by the PLK method. The solid line is the speed estimated by the proposed MMFL system. As we can see, the speed estimation of the MMFL system is smoother and the noise of solid line is much smaller than the others. In Figure 11a, the red point line with square is the camera position calculated by the manual calibration map, the black solid line with cross is the position calculated by ArUco Mapping and the solid line is the estimation result of the MMFL system. As we can see, the result of the MMFL system is considered to be a good estimation. Meanwhile, it can be seen that the interfered ultrasonic data (dotted line) does not affect the final estimation result of the MMFL system, which shows the federated filter has good anti-interference ability. Figure 11b gives the position fusion results of the MMFL system in the long-term loss of markers. The visual location information of markers is lost after the arrowhead. As can be observed from Figure 11b, the position fusion results in X-Axis and Y-Axis have the negligible drifting, for the optical flow can still give the speed measurement. Meanwhile, the estimation of Z-Axis also can be corrected with the measurement of ultrasonic after the marker lost.



Figure 9. Attitude Estimation Result.



Figure 10. Speed Fusion Result.



Figure 11. Position Fusion Result. (a) Without Marker Loss; (b) Marker Loss After the Arrowhead.

3.3. Comparison Experiment with ORB-SLAM2 System

In this section, the location performance of the proposed MMFL system is compared with the ORB-SLAM2 [34], where the benchmark trajectory is provided by the localization system including four UWB base stations. The 20 markers are distributed sparsely and randomly in the 5×3 m region which is shown in Figure 12a,b. The MMFL system is used directly to map the experiment area, and the video recorded by the Pi camera is provided to ORB-SLAM2 for 3D reconstruction in monocular mode. The 3D reconstruction result of ORB-SLAM2 is given in Figure 12c and the estimated camera trajectory of the two candidate systems are given in Figure 12d.



Figure 12. Results of Comparison Experiments with ORB-SLAM2. (a) The Front View of Experimental Site; (b) The Back View of Experimental Site; (c) 3D Reconstruction Result of ORB-SLAM2; (d) Estimated Camera Trajectory of Two Systems.

As can be observed form Figure 12d, the proposed MMFL system can successfully build the map of randomly and sparsely distributed markers in featureless outdoor environment. The localization result of ORB-SLAM2 is discontinuous, which is affected by the poor feature and sparse markers. The localization performance of the MMFL system is still satisfactory compared with the benchmark provided by UWB. In addition, the data refresh rate of the ORB-SLAM2 is 17 Hz (use the same hardware platform as the Method 2 mentioned in Section 3.1) which is much lower than the proposed MMFL system (100 Hz). Compared with the results in Section 3.1, it can be concluded that a dense marker array will provide higher accuracy, which is much more suitable for MAV localization with high precision requirement.

3.4. Experiment Results of Hovering and Flying

In order to test the location performance of the proposed MMFL system, the Px4flow sensor is chosen here for comparison. Firstly, a hover experiment with the Px4flow and the proposed MMFL system is implemented. The speed measurements is used as the feedback only since the Px4flow

can only give the speed measurement of MAV, and the same auto disturbances rejection controller (ADRC) [35] is used in this experiment. The trajectory of one minute hovering is given in Figure 13. In Figure 13, the dotted line is trajectory with Px4flow sensor as speed feedback, which has an obvious drift. The solid line is the trajectory with the MMFL system as speed feedback, which concentrates in the circle of the 18 cm.



Figure 13. Hover Trajectory of the Proposed System and Px4Flow.

Next, two autonomous flight missions are tested. Firstly, the MAV is commanded to follow an expected trajectory, and a video of round-trip in four way points can be seen at https://youtu. be/5BbyORJeGgk. In addition, the experiment result of following a circle trajectory is shown in Figure 14. In Figure 14, the dotted line is the expected circle trajectory and the square is the original camera position calculated by ArUco Mapping, which is discontinuous. The solid line is the trajectory estimated by the MMFL system, which is smooth and can still work even if the marker is not available to be detected. It is shown that the MAV is capable of flying autonomously and following the expected trajectory based on the proposed system, and a more appropriate control parameter may provide a better result. The video of following the circle trajectory can be seen at https://youtu.be/68FB3HK49-w.



Figure 14. Trajectory of Autonomous Flight.

4. Conclusions

A multi-sensor fusion indoor location system based on ArUco marker is proposed and designed in this paper. The ArUco Mapping algorithm uses the visual information from the markers on the ground (at unknown positions) to build a map automatically. The proposed multi-sensor fusion framework utilizes the federated Kalman filter to synthesize the multi-source information from markers, inertial sensor, ultrasonic and optical flow, which can obtain continuous localization and avoid the problem of dead zone in pure marker localization. The experiment results show that our system is capable of realizing MAV localization in a 4×4 m marker array with centimeter level positioning and mapping accuracy. However, the proposed system is sensitive to the accumulated error in ArUco Mapping. This limitation brings the requirement of introducing the map optimize and closure loop algorithm, which is the subject to be studied in the future work.

Author Contributions: B.X., F.P. and Q.Z. conceived the idea of the paper; B.X. designed and performed the experiments, and wrote the manuscript; F.P. and X.F. reviewed the manuscript.

Funding: This work has been supported by the National Natural Science Foundation (NNSF) of China under Grant 61603040.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Ardaens, J.; Damico, S.; Cropp, A. GPS-based relative navigation for the Proba-3 formation flying mission. *Acta Astronaut.* **2013**, *91*, 341–355. [CrossRef]
- 2. Zhao, L.; Qiu, H.; Feng, Y. Analysis of a robust Kalman filter in loosely coupled GPS/INS navigation system. *Measurement* **2016**, *80*, 138–147.
- 3. Global Position System. Available online: https://www.gps.gov/systems/gps/performance/accuracy/ (accessed on 2 January 2017).
- 4. Sola, J.; Vidal-Calleja, T.; Civera, J.; Montiel, J. Impact of Landmark Parametrization on Monocular EKF-SLAM with Points and Lines. *Int. J. Comput. Vis.* **2012**, *97*, 339–368. [CrossRef]
- 5. Montemerlo, M.; Thrun, S.; Roller, D.; Wegbreit, B. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *Int. Jt. Conf. Artif. Intell.* **2003**, *133*, 1151–1156.
- Song, Y.; Xian, B.; Zhang, Y.; Jiang, X.; Zhang, X. Towards autonomous control of quadrotor unmanned aerial vehicles in a GPS-denied urban area via laser ranger finder. *Opt. Int. J. Light Electron. Opt.* 2015, 126, 3877–3882. [CrossRef]
- 7. Agarwal, S.; Lazarus, S.B.; Savvaris, A. Monocular Vision Based Navigation and Localisation in Indoor Environments. *IFAC Proc. Vol.* **2012**, *45*, 97–102. [CrossRef]
- 8. Gutierrez-Gomez, D.; Mayol-Cuevas, W.; Guerrero, J.J. Dense RGB-D visual odometry using inverse depth. *Robot. Auton. Syst.* **2016**, *75*, 571–583. [CrossRef]
- Yang, Z.; Zhou, Z.; Liu, Y. From RSSI to CSI: Indoor localization via channel response. ACM Comput. Surv. 2003, 46, 25. [CrossRef]
- 10. Li, Z.; Chang, G.; Gao, J.; Wang, J.; Hernandez, A. GPS/UWB/MEMS-IMU tightly coupled navigation with improved robust Kalman filter. *Adv. Space Res.* **2016**, *58*, 2424–2434. [CrossRef]
- 11. Vicon Motion Capture. Available online: https://www.vicon.com (accessed on 11 January 2017).
- 12. Godil, A.; Tsai, R.E.; Hong, H. *Ground Truth Systems for Object Recognition and Tracking*; National Institute of Standards and Technology (NIST): Gaithersburg, MA, USA, 2013.
- 13. AprilSlam. Available online: https://github.com/ProjectArtemis/aprilslam (accessed on 2 January 2017).
- 14. Lim, H.; Lee, Y.S. Real-time single camera SLAM using fiducial markers. In Proceedings of the ICCAS-SISE 2009, Fukuoka, Japan, 17–21 August 2009; pp. 177–182.
- Klopschitz, M.; Schmalstieg, D. Automatic Reconstruction of Wide-Area Fiducial Marker Models. In Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality IEEE Computer Society, Nara, Japan, 13–16 November 2007; pp. 1–4.

- 16. Shaya, K.; Mavrinac, A.; Herrera, J.L.A.; Chen, X. A Self-localization System with Global Error Reduction and Online Map-Building. *Capabilities* **2012**, *7508*, 13–22.
- 17. Bacik, J.; Durovsky, F.; Fedor, P.; Perdukova, D. Autonomous flying with quadrocopter using fuzzy control and ArUco markers. *Intell. Serv. Robot.* **2017**, *10*, 1861–2776. [CrossRef]
- 18. Robot Operating System. Available online: http://www.ros.org/ (accessed on 11 January 2017).
- Sanchez-Lopez, J.L.; Arellano-Quintana, V.; Tognon, M.; Campoy, P.; Franchi, A. Visual Marker based Multi-Sensor Fusion State Estimation. In Proceedings of the 20th IFAC World Congress, Toulouse, France, 13–16 July 2017; 6p.
- 20. Muoz-Salinas, R.; Medina-Carnicer, R.; Marín-Jimenez, M.J.; Yeguas-Bolivar, E. Mapping and Localization from Planar Markers. *Pattern Recognit.* **2017**, *73*, 158–171. [CrossRef]
- 21. Honegger, D.; Meier, L.; Tanskanen, P.; Pollefeys, M. An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 1736–1741.
- 22. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* 2004, 92, 401–422. [CrossRef]
- 23. Carlson, N.A. Federated square root filter for decentralized parallel processors. *IEEE Trans. Aerosp. Electron. Syst.* **1990**, *26*, 517–525. [CrossRef]
- 24. Garrido-Jurado, S.; Muoz-Salinas, R.; Madrid-Cuevas, F.J.; Marin-Jimenez, M.F. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognit.* **2014**, 47, 2280–2292. [CrossRef]
- 25. Grubbs, F.E. Sample Criteria for Testing Outlying Observations. Ann. Math. Stat. 1950, 21, 27–58. [CrossRef]
- 26. Forgy, E.W. Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. *Biometrics* **1965**, *21*, 41–52.
- 27. Lucas, B.D.; Kanade, T. An iterative image registration technique with an application to stereo vision. In Proceedings of the International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, 24–28 August 1971; pp. 674–679.
- 28. Bouguet, J.-Y. Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm. *OpenCV Doc.* **1999**, *22*, 363–381.
- 29. Hamilton, W.R. On Quaternions, or on a New System of Imaginaries in Algebra. Philos. Mag. 1848, 33, 58-60.
- 30. Zhang, Y.; Cai, Y.; Tang, C.K. A Study of Information Sharing Factor Based on Federated Kalman Filter. *Comput. Simul.* **2009**, *26*, 358–361.
- 31. Qui, K.; Wei, R.X.; Zhang, Z.L. Study of an adaptive and fault-tolerant federated filter algorithm based on RBF. *Control Decis.* **2004**, *19*, 1420–1424.
- Lynen, S.; Achtelik, M.W.; Weiss, S.; Chli, M. A robust and modular multi-sensor fusion approach applied to MAV navigation. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 3923–3929.
- 33. Mavlink. Available online: http://qgroundcontrol.org/mavlink/start (accessed on 2 January 2017).
- 34. Raul, M.A.; Juan, D.T. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262.
- 35. Balajiwale, S.; Arya, H.; Joshi, A. Study of Performance of ADRC for Longitudinal Control of MAV. *IFAC PapersOnLine* **2016**, *49*, 585–590. [CrossRef]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).