An Embedded Solution to Visual Mapping for Consumer Drones

Guyue Zhou and Zexiang Li
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology
Hong Kong, P.R.China
{*zhouguyue,eezxli*}*@ust.hk*

Ang Liu, Kang Yang and Tao Wang
Dajiang Innovations Technology Co., Ltd.
Shenzhen, P.R.China
{*leon.liu, kang.yang, frank.wang*}*@dji.com*

***Abstract*—In this paper, we propose a real-time visual mapping scheme which can be implemented on a low-cost embedded system for consumer-level ratio control (RC) drones. In our work, a 3-dimensional occupancy grid map is obtained based on an estimated trajectory from data fusion of multiple on-board sensors, composed of two downward-facing cameras, two forward-facing cameras, a GPS receiver, a magnetic compass and an inertial measurement unit (IMU) with 3-axis accelerometers and gyroscopes. Taking the advantages of the low-cost FPGA and ARM NEON intrinsics, we run our visual odometry and mapping algorithms at 10Hz on board. Meanwhile, we also present a hierarchical multi-sensor fusion algorithm to provide a robust trajectory for mapping usage. Finally, we verify the feasibility of our approaches and serval potential applications with experimental results in complex indoor/outdoor environments.**

I. Introduction

Recently, the micro multi-rotors become quite popular because of their small dimension, low cost, low power consumption and high maneuverability. All these characteristics make multi-rotors be adapted to a growing list of commercial applications, including aerial photography, aerial inspection, entertainment and recreation. As a result, it is an irresistable trend to develop more intelligent control systems on the low-cost hardware platform with regard to safety concerns and consumer requirements.

With a fire-new perspective observing the surroundings, the community of RC drones becomes a newly emerging force in the market of consumer electronics. Specifically, Parrot AR.Drone [1] is a very successful product which can provide on-board first-person view (FPV) and the hovering function based on a combination of ultrasonic and optical flow sensors in indoor environment. Meanwhile, in the broader outdoor cases, DJI Phantom series [2] is holding the dominant position in this segment of the market. It owes much to DJI's powerful wireless communication technique and mechanically stabilized camera on board. However, none of existing consumer-level drones can take intelligent actions due to the trade-off between computing power and cost.

Technically, local mapping is a critical technique for high-level intelligent applications hence many researchers demonstrated this ability using different sensor configurations. In 1991, Leonard *et al.*[3] proposed the famous SLAM (simultaneous localization and mapping) problem which coupled the mapping and localization problems together. Recently, there were a numbers of successful examples of implementing real-time SLAM algorithm for micro aerial vehicle (MAV) applications. For instance, Bachrach *et al.* [4] used laser-based SLAM to achieve a straight-forward 2-dimensional grid map and Shen *et al.*[5] generated a 3-dimensional sparse feature map using a simplified visual SLAM, which optimized the map, rotation and translation separately and used a linear solution to a modified re-projection function for translation estimation based on a constant depth (in two successive frames) assumption. On the other hand, another milestone work was proposed by Fraundorfer *et al.*[6] to solve stereo mapping and visual odometry independently. But their system was too complex since there were totally three processing units, including an ARM Cortex M4 based optical flow module [7], an Intel 1.86GHz dual core processor and an off-board ground station.

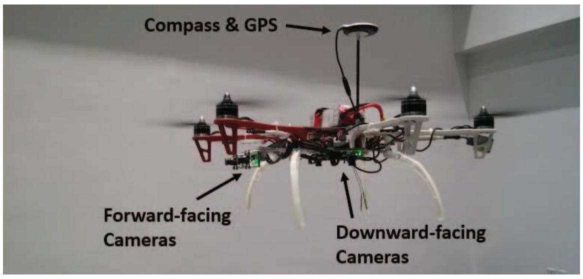

Figure 1. The hex-copter platform (DJI F550) with the on-board sensing suite consisted of an 200Hz IMU (DJI A2), two 10Hz downward-facing cameras (18cm baseline), two 10Hz forward-facing cameras (12cm baseline), an integrated GPS and compass module (DJI A2) and a low-cost processing unit.

Considering the limited computational power for a low-cost embedded system, we would like to track the separated mapping and localization framework, causing the multi-sensor fusion becomes significant to both localization precision and mapping performance. In previous work, Schleicher *et al.* [8] proposed a GPS-assisted visual SLAM which could hardly provide a smooth positional sequence in real time.

Shen *et al.* [9] used an UKF-based fusion to generate quite a smooth estimation result but their work was not applicable for modular systems.

The main objective of this paper is to propose a low-cost embedded solution to visual mapping based on multi-sensor fusion. We build upon our previous work [10] and therefore assume the existence of an embedded on-board visual odometer. Then the stereo-based depth map can be directly applied on the flight trajectory generated by fusing inertial readings, visual odometry and GPS signals hierarchically.

The remainder of the paper is organized as follows. Section II introduces the hardware architecture and software framework of the system design. Section III explains the detailed methods used for multi-sensor data fusion. Section IV describes our experiments and results.

II. System Design

In this section, the proposed embedded solution to a real-time visual mapping system for consumer drones will be described in detail in two parts respectively: the sensing part and the processing part.

A. Sensor Configurations

All popular on-board sensors for MAV applications are listed in Table.I for selection. In the first place, LIDARs is not a favorable choice for a high cost and active cameras is excluded as well for its disability in the sunlight.

Additionally, the classical assembly for RC drones, consisted of the IMU, the GPS and the magnetic compass, should be reserved for the low-level attitude stabilization and the alignment between global and body frames. In our system, the DJI A2 (including a flight control unit, an IMU, a GPS receiver and a magnetic compass) will be used to simplify both our hardware design and low-level software development.

Lastly, cameras is obviously a better choice for mapping tasks than sonars. A pre-calibrated stereo rig is required to determine the metric scale. Therefore, we decide to install two stereo camera pairs (forward-looking and downward-looking respectively) to fulfill both the odometry and mapping requirements.

B. Hardware and Software Co-design

1) System Overview: This work is in fact an updated version of our previous work [10] therefore a low-cost Altera SoC FPGA (with ARM hardcores inside) continues to be selected as the main processor with the following advantages.

- Be capable to fit various interfaces of multiple sensors;
- Be capable to handle massive and repetitive operations (e.g. low-level image processing) with low-cost and high performance;
- Be capable to support module-based time sharing multiplex (TSM);
- Be with lower cost than mini-computers (e.g. Intel 1.86GHz dual core processor).

Table I
Popular On-board Sensors for MAV Applications

Sensor	Cost	Acc- uracy	FOV	In- door	Out- door	Frame
GPS	Low	Low	N/A	×	✓	Global
Compass	Low	Low	N/A	×	✓	Global
IMU	Low	Low	N/A	✓	✓	Body
Sonar	Low	Low	Small	✓	✓	Body
LIDAR	High	High	Large	✓	✓	Body
Camera	Low	High	Media	✓	✓	Body
Active Camera	High	High	Media	✓	×	Body

The intuition of our solution is to implement standard pixel-level algorithms (e.g. anti-distortion, rectification, feature detection and description, dense stereo matching, etc.) on FPGA and to run flexible feature-level algorithms (e.g. feature tracking or matching, outlier rejection, motion estimation, occupancy grid mapping, etc.) on ARM, making it possible to run complicated computer vision algorithms on low-cost and computationally constrained embedded systems. For this purpose, it is critical to follow the criteria below during the algorithm design.

- Select algorithms which are based on integer or fixed-point calculation for FPGA;
- Select algorithms which use less on-chip buffer memory for FPGA;
- Select algorithms which use lower bandwidth of randomly accessing the external memory for FPGA;
- Divide multi-stage algorithms into pipelined modules in FPGA;
- Apply TSM to multiple cameras on FPGA;
- Select algorithms which can be accelerated with the single instruction multiple data (SIMD) technique for ARM;
- Avoid using high-dimensional float-point iterative optimizations on both FPGA and low-cost ARM.

Fig.2 shows the hardware architecture and the data flow of the proposed system. The selected algorithms are pointed out in Table.II. Furthermore, a summary of the visual mapping software framework is illustrated in Fig.3.

2) Hardware and Data Flows: For the first step (red data flow in Fig.2), image data are read from four camera sensors and are stored in external DDR memory. Then (blue data flow in Fig.2), for each stereo pair, the image pixels are pre-processed (un-distorted and rectified), and then pushed into feature extraction (including FAST and BRIEF) module and block matching module in parallel. Please notice that there exist data exchange between on-chip memory and external DDR in this part but it is not marked in Fig.2 because the

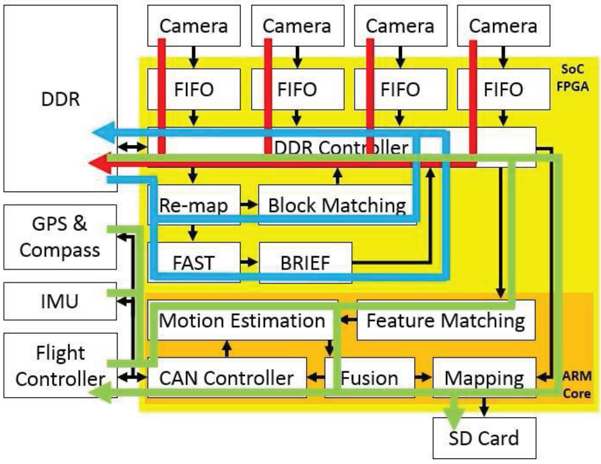

Figure 2. The system-level hardware architecture and data flows: there are totally 3 operating steps marked in red, blue and green colors in turn.

Table II
ALGORITHMS FOR EMBEDDED IMPLEMENTATION

Function	Algorithm	Features
Anti-distortion, rectification	Re-map	Look-up table
Feature detection	FAST [11]	Integer-based small buffer
Feature description	BRIEF [12]	Binary-based median buffer
Feature association	Match	Bit-opertation
Dense depth map	Block matching	Integer-based small buffer
Motion estimation	LONSC [10]	Simple and fast
Visual mapping	Octomap [13]	Simple and fast

DDR bandwidth consumption is quite small. The obtained feature information and dense depth map will go back to DDR for buffering. Interested readers can refer to [10] for some module-level details for FPGA implementation. Finally, the ARM hardcore will convert the collected data from multiple sensors to visual mapping results recorded in a SD card according to the green data flow in Fig. 2.

3) Software and Implementation: It is clear that the software and hardware designs are highly coupled in our proposed system. It means that, for example listed in Table.II, the feature descriptor will have a significant influence on not only the FPGA hardware implementation but also the feature tracking method in ARM. BRIEF [12] is a perfect choice for its simple implementation and feasibility of bit-operation based matching, which is well suited to ARM NEON intrinsics, which are high-level instructions based on the SIMD technique. In particular, some of the implementation tricks are introduced as below.

- The Calculation of the Hamming distance between two 32-byte descriptors is done using a look-up table indexed with the XOR result of each pair of bytes and such kind of algorithm structure can be greatly accelerated with the SIMD technique since there is no dependency among different pairs;
- Store the look-up table in the fastest cache because it will be accessed very frequently;
- Address the pointer only once for each descriptor to avoid the repeated calculations during addressing.

In our experience, by refining a straight-forward implementation, the BRIEF-based feature matching algorithm will be 5-6 times faster.

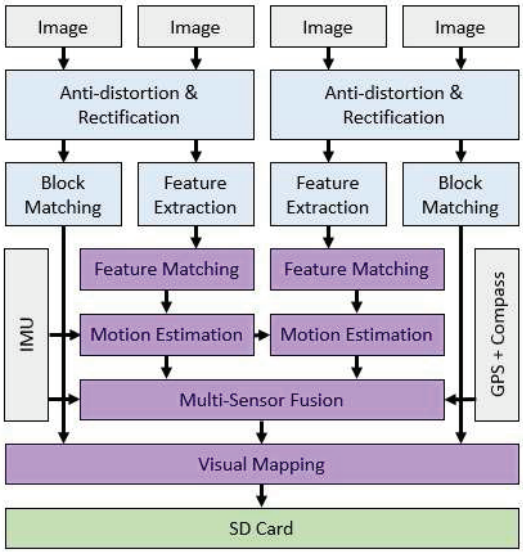

Figure 3. The software framework for visual mapping (loosely coupled fusion for visual odometers): gray and green modules stand for input and output devices accordingly and blue and purple modules are implemented using Verilog HDL and C++ respectively.

III. MULTI-SENSOR FUSION

Based on the ready-to-use filtered rotation (embedded in IMU), the rest of our multi-sensor fusion tasks can be decomposed into three steps. Firstly, IMU data is used to speed up the computation of visual odometer and to filter with the visual odometer outputs. Secondly, we need to combine the results from forward-facing visual measurement with the downward-facing one. Thirdly, we will consider the fusion between local frame (based on body measurements) and global frame. Since the first problem has been discussed in [10], we will focus on the system calibration and the other two fusion problems in this section.

A. Calibration

Initially, lens distortion parameters, camera intrinsic parameters and stereo extrinsic parameters are calibrated using traditional methods (OpenCV) [14]. Then Daniilidis' method [15] is employed for calibrating IMU-camera extrinsic parameters, which is a typical hand-eye calibration problem.

Meanwhile, the extrinsic parameters between two stereo pairs can be solved using the same hand-eye calibration method based on the estimated motion sequences from two visual odometers. Therefore, we simply refine the IMU-camera extrinsic parameters based on the more reliable bias-free vision-based results.

B. Fusion between Visual Odometers

To fuse the measurements from different visual views, both loosely coupled and tightly coupled methods will be introduced in this subsection. Please notice that only translational motion is considered since the rotational motion in body frame is directly measurable using IMU.

1) Loosely coupled fusion: It is as shown in Fig.3 that two visual odometers are run independently and the outputs are denoted by $\{\boldsymbol{R}_{\Delta c_0}, \boldsymbol{t}_{\Delta c_0}\}$ and $\{\boldsymbol{R}_{\Delta c_1}, \boldsymbol{t}_{\Delta c_1}\}$ where Δ stands for the differential motion between two successive sampling time points and all of them are represented in the corresponding camera-centered local frames. Furthermore, the average depth d_i and the count of inliers m_i for each stereo pair are provided to set up a complementary filtering scheme as below.

$$K = \begin{cases} 0 & \text{if } m_1 \leq m_h \text{ and } m_0 > m_h \\ 1 & \text{if } m_0 \leq m_h \text{ and } m_1 > m_h \\ \frac{d_0}{d_0+d_1} & \text{if } m_{0,1} \leq m_h \\ \frac{d_0}{d_0+d_1} & \text{if } m_{0,1} > m_h \text{ and } |d_0 - d_1| \leq d_h \\ \frac{m_1}{m_0+m_1} & \text{if } m_{0,1} > m_h \text{ and } |d_0 - d_1| < d_h \end{cases} \tag{1}$$

where m_h is a threshold to judge if the visual odometor is valid, d_h is a threshold of the gapped distance between observed objects in different views.

$$\boldsymbol{t}^{c_0}_{\Delta b} = \tilde{\boldsymbol{R}}_{bc_0}\boldsymbol{t}_{\Delta c_0} + (\boldsymbol{I}_{3\times 3} - \boldsymbol{R}_{\Delta b})\tilde{\boldsymbol{t}}_{bc_0} \tag{2}$$

$$\boldsymbol{t}^{c_1}_{\Delta b} = \tilde{\boldsymbol{R}}_{bc_1}\boldsymbol{t}_{\Delta c_1} + (\boldsymbol{I}_{3\times 3} - \boldsymbol{R}_{\Delta b})\tilde{\boldsymbol{t}}_{bc_1} \tag{3}$$

$$\boldsymbol{t}_{\Delta b} = (1-K)\boldsymbol{t}^{c_0}_{\Delta b} + K\boldsymbol{t}^{c_1}_{\Delta b} \tag{4}$$

Here $\boldsymbol{t}_{\Delta b}$ is the desired translation in body frame fusing multiple visual measurements.

2) Tightly coupled fusion: In the tightly coupled scheme, there are two stereo views denoted by two camera-centered frames c_0 and c_1 and we need to project every observed 2D-3D correspondence $c_i^{c_1} = \{\boldsymbol{p}_i^{c_1}, \boldsymbol{u}_i^{c_1}\}$, which are used for motion estimation [10], in c_1 to the other frame c_0.

$$\boldsymbol{p}_i^{c_0} = \boldsymbol{R}_{c_0c_1}\boldsymbol{p}_i^{c_1} + \boldsymbol{t}_{c_0c_1} \tag{5}$$

$$\boldsymbol{u}_i^{c_0} = \frac{\boldsymbol{R}_{c_0c_1}\boldsymbol{u}_i^{c_1} + \boldsymbol{t}_{c_0c_1}}{[\boldsymbol{R}_{c_0c_1}\boldsymbol{u}_i^{c_1} + \boldsymbol{t}_{c_0c_1}]_z} \tag{6}$$

where $[\boldsymbol{v}]_z$ indicates the third element for a 3×1 vector $\boldsymbol{v}$.

Next, a unified motion estimation is applied on the visual features in different views but two important tricks are required. One we appoint different re-projection thresholds to improve the estimating robustness to the extrinsic calibration parameters. Also, leading in a hysteretic unit is highly recommended to avoid the repeated sudden changes in the marginal conditions.

C. Fusion between Body and Global Frames

In this section, we try to align the sensing results collected from various sensors. In particular, some of the sensors measure the relative motion in the body frame and the others measure the absolute motion in the global frame. The former is usually rigidly connected with the vehicle body for the local navigation and exploration but suffers the long-term accumulated errors. In contrast, the latter is free of drift but can hardly be always available and accurate enough.

To bridge the gap between the body and global frames, we only need to focus on the yaw alignment according to our visual mapping target because of the following reasons.

- The measurement of pitch and roll from IMU is free of drift due to the observable gravity direction;
- The mapping performance is greatly affect by the rotational mismatches instead of translational offsets;
- The yaw error will significantly magnify the integrated positional error.

Next we will discuss the detailed methods. Noted that here we ignore the altitude information in the analysis of the planar cases. For the global frame, we select good GPS signals (with more than 5 satellites) periodically and represent them using $\boldsymbol{p}_i^g$ in the North-East frame. Meanwhile, we record the corresponding positional measurement $\boldsymbol{p}_i^b$ in the body frame. The mismatched yaw angle θ and scale factor s (due to the integrated positional drift) are the only 2 degrees of freedom to be aligned based on n correspondences $\{\boldsymbol{p}_i^b, \boldsymbol{p}_i^g\}$ between body and global frames. It can be modeled as the following nonlinear least square problem.

$$\{\theta, s\} = \arg\min_{\theta, s} \sum_{i=1}^{n} \left\| \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \boldsymbol{p}_i^b - s\boldsymbol{p}_i^g \right\|_2^2 \tag{7}$$

The famous Levenberg-Marquardt algorithm [16] is employed to solve this non-linear least square problem with an initial guess $\theta_0 = 0$ and

$$s_0 = \frac{2}{n(n+1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{\left\|\boldsymbol{p}_i^b - \boldsymbol{p}_j^b\right\|_2}{\left\|\boldsymbol{p}_i^g - \boldsymbol{p}_j^g\right\|_2} \tag{8}$$

where n is the number of local/global correspondences.

IV. Experimental Results

A. Experiment Configurations

Besides an IMU, a precise GPS positioning system and an autopilot board, we have a forward-facing CMOS global shutter camera pair with a 12cm baseline and another downward-facing pair with a 18cm baseline amounted on a hex-copter. These cameras capture images with QVGA (320×240) resolution and transfer them to a processing board based on a low-end SoC FPGA (with two 800MHz ARM Cortex hardcores inside). The cameras' frame-rates are set at 10Hz whereas the IMU provides measurements at 200Hz.

The experiment environment includes two indoor/outdoor transition areas. We firstly conduct two experiments: (1) performance evaluation of our fusion algorithm between inertial-assisted visual odometers on real data, and (2) real time visual mapping in complex environment. Then we demonstrate some potential applications based on our proposed system.

B. Fusion between Visual Odometers

In this part, we make a comparison experiment to see if loosely coupled method could work better than tightly coupled method or not. Since it is very hard to find a ground truth reference in such an indoor/outdoor transition environment, we instead evaluate their hovering performances in 3 conditions: (1) both stereo pairs can capture textured objects with different observing distances, (2) both stereo pairs can capture similar-distanced textured objects, and (3) only downward-facing stereo pair can capture textured objects.

Initially, we only use the downward-facing cameras to hover the MAV and then we calculate the hovering error using different fusion methods as shown in Fig.4. In the first condition, the MAV faces to a wall to ensure the forward-facing cameras have better observation causing the fusion results (blue and green curves) are quite far away from reference curve (red) in the first figure. In the other two conditions, fusion results are very close to or equivalent to the downward-facing visual odometer since the downward-facing stereo pair becomes the significant/dominating component in fusion.

After verifying our intuitions, we try to use different fusion methods to provide positional feedback control for hovering. The expectation and standard deviation of the hovering errors using single stereo pair, loosely coupled algorithm and tightly coupled algorithm are respectively $\boldsymbol{m}_{e_1} = 0.185$m, $\boldsymbol{\sigma}_{e_1} = 0.094$m, $\boldsymbol{m}_{e_2} = 0.106$m, $\boldsymbol{\sigma}_{e_2} = 0.073$m and $\boldsymbol{m}_{e_3} = 0.118$ m, $\boldsymbol{\sigma}_{e_3} = 0.076$m, implying the more stereo pair really helps.

C. Real-time Visual Mapping

This experiment considers the performance of the on board visual mapping system in a scenic spot environment which contains a mixture of indoor and outdoor conditions. During the flight, the MAV is commanded to take off outside a courtyard, then flies through the opened door and landed inside the courtyard. The frequency of both localization and mapping systems is set at 10Hz. The total duration of this experiment is around 100 seconds and the average flying speed is approximately 0.5m/s (the maximum speed is around 1.4m/s).

As shown in Fig.5, we can see that these GPS landmarks greatly improve the performance of the inertial-assisted visual odometers in term of eliminating accumulated errors. The crosses are the sampled points for the 1Hz fusion (based on the most recent 5 correspondences) between the body and global frames.

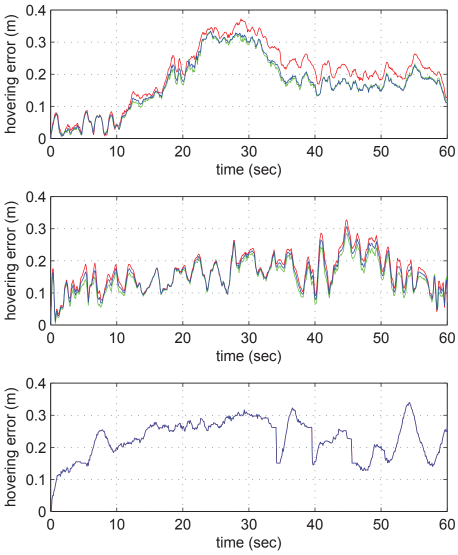

Figure 4. Hovering errors (red: downward-facing visual odometer, green: loosely-coupled fusion, blue: tightly-coupled fusion) in different conditions.

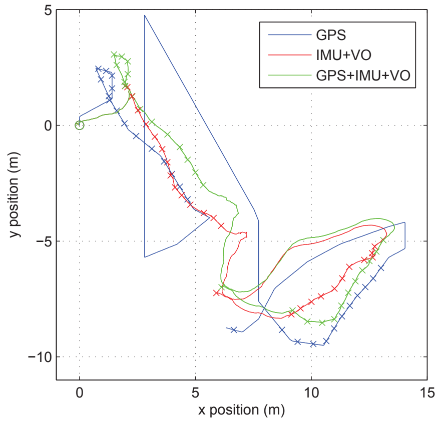

Figure 5. The comparison among GPS-based, inertial/vision-based and fusion-based trajectories: the crosses represent the sampled points for local/global fusion.

In the end, the visualization of the estimated trajectory and mapping results can be found in Fig .6 and Table.III explains the averaged runtime of every module on board.

D. Potential Applications for Consumer Drones

In this part, two potential applications are introduced to demonstrate how the consumer drones can benefit from our

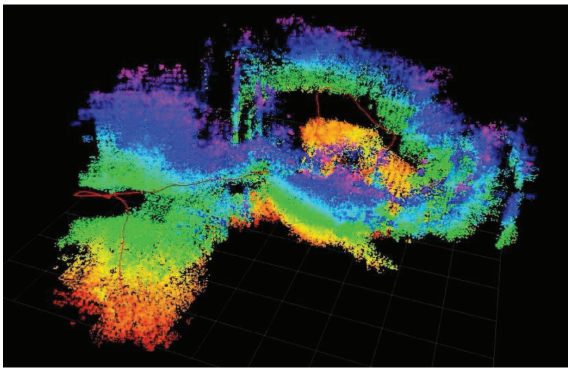

Figure 6. The visualization of 3D mapping results (the color-coded point cloud: colors transit from red to blue while the altitude is increased) and the estimated vehicle trajectory (the red curve)

Table III
RUNTIME OF THE PROPOSED MODULES

Module	FPGA Runtime	ARM Runtime
Re-mapping	14.24ms	N/A
FAST + BRIEF	19.12ms	N/A
Block Matching	33.42ms	N/A
Feature Matching	N/A	17.21ms
Motion Estimation	N/A	16.46ms
Data Fusion	N/A	8.37ms
Mapping	N/A	24.25ms
Total	66.78ms	66.29ms

low-cost mapping system. However, the robust implementations can be considered as our future work.

Active Safety: The RC drones usually crash due to beginners' unintentional misoperations, thereby motivating an active safety system on board. To solve this problem, visual mapping is a good choice to provide a large field of view under cost and dimension constraints.

Autonomous Homing: The RC drones usually go out of control because of the radio signal break, causing the autonomous homing function is necessary for commercial drones. With the current GPS-based solution, there exist great hidden dangers since it can not sense local dynamic environments. In this study, we just implement a simple zigzag path planning to demonstrate the feasibility of detecting and avoiding obstacles with our low-cost visual mapping solution.

V. CONCLUSION

In this paper, we present an embedded solution to visual mapping based on the multi-sensor fusion results. The entire computation is processed on board and is capable to be implemented using a low-cost hardware platform which is suitable to consumer drones. Several experiments have been carried out to demonstrate that both of our proposed data fusion scheme and our proposed mapping method work well in complex environments.

REFERENCES

[1] P.-J. Bristeau, F. Callou, D. Vissiere, and N. Petit. The navigation and control technology inside the ar.drone micro uav. In *18th IFAC World Congress*, pages 1477–1484, 2011.

[2] A. Fitzpatrick. Finally, a drone you can own. *TimeTech: http://techland.time.com/2014/01/30/dji-phantom-vision-quadcopter-drone-review/*, Jan. 30, 2014.

[3] J. Leonard and H. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, pages 1442–1447 vol.3, Nov 1991.

[4] A. Bachrach, A. de Winter, R. He, G. Hemann, S. Prentice, and N. Roy. Range - robust autonomous navigation in gps-denied environments. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1096–1097, May 2010.

[5] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar. Vision-based state estimation for autonomous rotorcraft mavs in complex environments. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1758–1764, May 2013.

[6] F. Fraundorfer, L. Heng, D. Honegger, G. Lee, L. Meier, P. Tanskanen, and M. Pollefeys. Vision-based autonomous mapping and exploration using a quadrotor mav. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4557–4564, Oct 2012.

[7] 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*, pages 1736–1741, May 2013.

[8] D. Schleicher, L. Bergasa, M. Ocana, R. Barea, and E. Lopez. Real-time hierarchical gps aided visual slam on urban environments. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 4381–4386, May 2009.

[9] S. Shen and N. Michael. State estimation for indoor and outdoor operation with a micro-aerial vehicle. In *The 13th International Symposium on Experimental Robotics*, pages 273–288, 2012.

[10] G. Zhou, J. Ye, W. Ren, T. Wang, and Z. Li. On-board inertial-assisted visual odometer on an embedded system. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014.

[11] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[12] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010*, pages 778–792. Springer, 2010.

[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34:189–206, 2013.

[14] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

[15] K. Daniilidis. Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research*, 18(3):286–298, 1999.

[16] P. Gill and W. Murray. Algorithms for the solution of the nonlinear least-squares problem. *SIAM Journal on Numerical Analysis*, 15(5):977–992, 1978.