Feature Article:

DOI. No. 10.1109/MAES.2018.170075

Implementation and Performance Evaluation of Optical Flow Navigation System Under Specific Conditions for a Flying Robot

***Ali Aminzadeh, M.A. Amiri Atashgah,* University of Tehran, Tehran, Iran**

INTRODUCTION

Honeybees land on a flower [1], [2], estimate traveled distance [3], and avoid obstacles [4] merely based on optical flow (OF) reckonings. Like OF reckoning, OF navigation used in flying robots uses biological and computer vision activities to estimate distance and avoid obstacles. Currently, most flying robots use Global Navigation Satellite System (GNSS)/inertial navigation system (INS) integration for navigation/positioning purposes. Navigation is defined as estimation/determination of the position, velocity, and orientation of a vehicle. If the position alone is considered, it is called positioning/localization data and is one input for guidance and control systems in a flight computer [5]. Accuracy and safety during an autonomous flight rely heavily on the positioning data. However, under indoor and in some outdoor environments similar to dense forests and caves, GNSS signals are not accessible or dependable [6]. In environments like this, without any supplemental sensing devices, flying robots are routinely incapable of sensing and collision avoidance. For this reason, the use of such systems is not appropriate for indoor and outdoor spaces where GNSS signals are not reliable/available.

Typically, GNSS measurement accuracy is only a few meters while a vision-based navigation system can provide measurement accuracy of a few centimeters [7]. Inertial navigation systems are often used where GNSS is unavailable; however, due to the accumulated errors in an inertial navigation system and the extensive dependence of a flying robot on accurate position and velocity data, we are proposing to use a low-cost vision sensor for integration with a navigation system for a long-term operation. In fact, localization in a long-term operation is essential for every autonomous robot and many methods have been suggested for such a situation [8].

In this article, we test an online optical navigation system (ONS) onboard a flying robot under different flight conditions, under desirable and undesirable illumination conditions, and on indoor and outdoor missions. To this end, initially utilizing an offline toolset, major claimed methods such as Lucas-Kanade, Horn-Shunk, and Block Matching are adopted for onboard implementation and further improvements in terms of precision and stability. For both outdoor and indoor experiments, scientific evidence of flight path is provided based on ground truth data created with the help of Google Earth by choosing four waypoints around a specific location in Kooye Daneshgah and the Guidance and Navigation Laboratory of the University of Tehran, respectively.

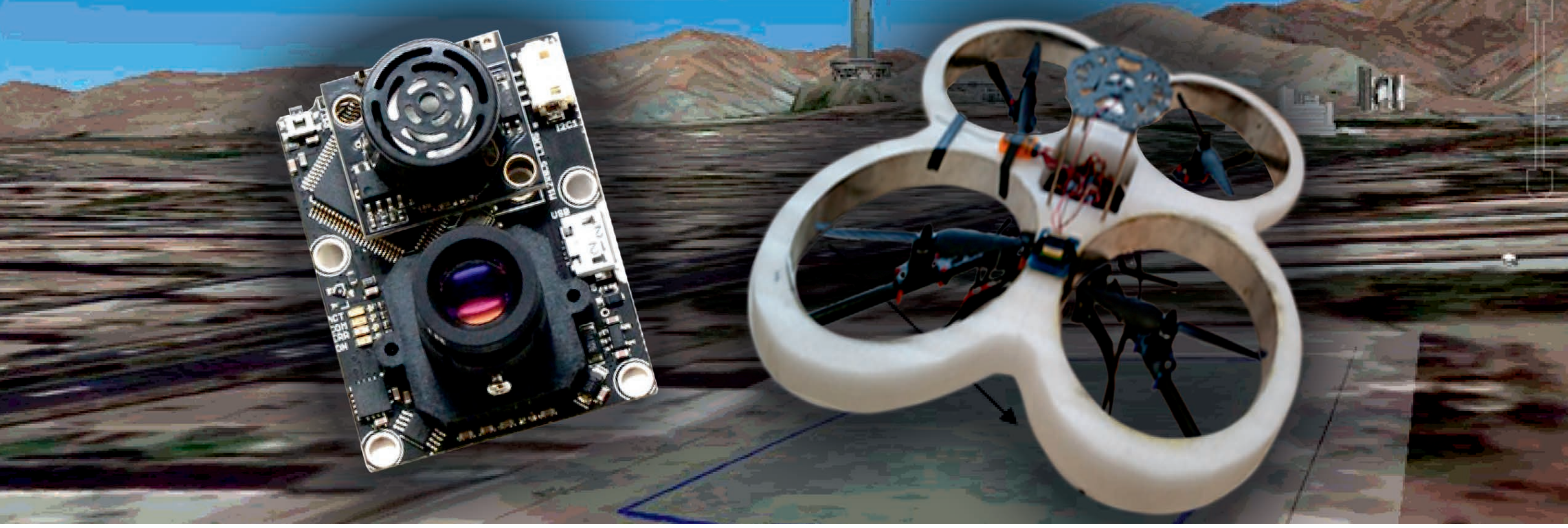
It is worth mentioning that in many real conditions, such as indoor operation with low-cost sensors, positioning cannot be reliable. In many environments, a camera is an ideal sensor for positioning for a long-term mission. A camera can provide worthy information from the environment around the robot to estimate the position and orientation of the vehicle. In recent years, numerous research efforts have been done in the field of vision-based autonomous navigation. OF is one of the most common methods in the vision-based navigation of vehicles. There are several methods for OF based estimation of position and velocity, which fall into four general categories: (1) differential or gradient methods, (2) correlation and block matching (BM) schemes, (3) energy and phase-based approaches, and (4) sensor-based methods [9]. For more, [10] introduces the two basic pixel-based and block-based motion estimation approaches.

RELATED WORKS

In this section, we review the existing methods for OF reckonings and cite some examples of the applications of ONSs for flying robots. As per the indications in [11], various types of schemes for OF calculations have been developed, such as differential methods (e.g., Lucas-Kanade (LK) [12] and Horn-Shunk (HS) [13]), feature-based methods (e.g., Scale-Invariant Feature Transform (SIFT) [14]), and interpolation techniques [15]. Despite all advances in ONS methods, there are still some difficulties in achieving accurate and efficient real-time OF estimations, especially in aerial vehicles. In the following, we refer to some research on the application of OF-based reckonings in aerial visual navigation. Kendoul et al. [9] presented an optical autopilot for unmanned aerial vehicles (UAVs) for autonomous localization and mapping in which a real-time vision algorithm was designed to estimate flow field using a low-resolution onboard camera. According to [9], based on several real-

Authors' current address: Department of New Sciences and Technologies, University of Tehran, P.O.B. 14395-1561 Tehran, Iran, E-mail: (atashgah@ut.ac.ir).

Manuscript received April 17, 2017, revised September 21, 2017, January 22, 2018, and ready for publication March 14, 2018.
Review handled by M. Braasch.
0885/8985/18/$26.00 © 2018 IEEE



time indoor and outdoor tests, the proposed vision system is robust and efficient for the operations which are almost independent of the environment and camera motion. Conroy et al. [16] presented an implementation of an insect-inspired gradient-based OF algorithm for autonomous quadrotor navigation in which local proximity information is extracted through spatial decomposition of the instantaneous OF. Accordingly, the presented method is beneficial in terms of computational speed and simplicity. Prashanth et al. [17] studied a gradient-based OF scheme to examine navigation and obstacle avoidance ability of a quadrotor; there the HS method was implemented on an Open Multimedia Applications Platform -3530 onboard processor. In [18], the speed of the vehicles in traffic was determined by implementing a semireal-time vehicle tracking algorithm on traffic control videos. The results show that LK is more accurate than the HS method. Furthermore, in [19], an OF based control system is implemented on a quadrotor to enable a fully autonomous flight. In that paper, LK is used to estimate a two-dimensional (2D) position and altitude estimation, by fusion of ultrasonic, infrared, inertial, and pressure sensor data. Correspondingly, Yang [20] presented a real-time implementation of LK method on a ground vehicle to estimate its side velocity and position.

Even though many types of research have been conducted in the field of optical navigation, one cannot find any proper criteria to evaluate the performance of the methods. [21] and [22] suggested two special techniques to compare different OF methods that are based on the angle between estimated and corrected flow vector and also an absolute difference between estimated and corrected flow vector, respectively. Moreover, many different types of OF schemes are compared in [23] and [24], based on the two suggested methods.

MATHEMATICS AND FORMULATION OF ONS

PIXEL-BASED MOTION ESTIMATIONS

In a gradient-based optical flow method, the estimation of image pixel displacement via local derivatives through sequences of images is performed. Initially, spatial-temporal intensity derivatives are measured and then the image velocity from the motion constraint equation is computed. If I (x, y, t) is the center pixel in a $n \times n$ neighborhood system and moves by δx and δy, in a specified time δt toward I (x + δx, y + δy, t + δt), assuming a constant intensity for small δt, we have:

$$I(x,y,t) = I(x+\delta x, y+\delta y, t+\delta t). \quad (1)$$

By applying first order, Taylor series expansion about I(x, y, t), (1) gives:

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + \text{HOT}. \quad (2)$$

Ignoring the higher order terms (HOT) of (2) and dividing it by δt, the following relation appears:

$$\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t} = 0 \quad (3)$$

in which $\partial I/\partial x$, $\partial I/\partial y$, and $\partial I/\partial t$ are intensity derivatives and v_x and v_y are image velocity of OF [25]. This equation is called the Motion Constraint Equation. To solve the aforementioned equation, several methods such as LK [12] and HS [13] have been introduced in the literature.

LK METHOD

In the LK algorithm, it is assumed that the intensity of a $n \times n$ neighborhood of the desired pixel is constant. Hence the Motion Constraint Equation will be solved for all pixels in the featured neighborhood.

$$\begin{aligned}
&\frac{\partial I(q_1)}{\partial x}v_x + \frac{\partial I(q_1)}{\partial y}v_y + \frac{\partial I(q_1)}{\partial t} = 0\\
&\frac{\partial I(q_2)}{\partial x}v_x + \frac{\partial I(q_2)}{\partial y}v_y + \frac{\partial I(q_2)}{\partial t} = 0\\
&\qquad\qquad .\\
&\qquad\qquad .\\
&\frac{\partial I(q_n)}{\partial x}v_x + \frac{\partial I(q_n)}{\partial y}v_y + \frac{\partial I(q_n)}{\partial t} = 0
\end{aligned} \quad (4)$$

The system of equations in (4) can be transcribed in a matrix arrangement to be solved through the least squares method [25], as noted below:

$$v=\begin{bmatrix} v_x \\ v_y \end{bmatrix} b=\begin{bmatrix} -\frac{\partial I(q_1)}{\partial t} \\ . \\ . \\ . \\ -\frac{\partial I(q_n)}{\partial t} \end{bmatrix} A=\begin{bmatrix} \frac{\partial I(q_1)}{\partial x} & \frac{\partial I(q_1)}{\partial y} \\ . & . \\ . & . \\ . & . \\ \frac{\partial I(q_n)}{\partial x} & \frac{\partial I(q_n)}{\partial y} \end{bmatrix} \tag{5}$$

$$v=(A^TA)^{-1}A^Tb \tag{6}$$

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix}=\begin{bmatrix} \sum_{i=1}^{n} I_x(q_i)^2 & \sum_{i=1}^{n} I_x(q_i)I_y(q_i) \\ \sum_{i=1}^{n} I_x(q_i)I_y(q_i) & \sum_{i=1}^{n} I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^{n} I_x(q_i)I_t(q_i) \\ -\sum_{i=1}^{n} I_y(q_i)I_t(q_i) \end{bmatrix} \tag{7}$$

HS METHOD

In addition to the gradient constraint which presented in (3), the global smoothness term is considered in the HS algorithm as a minimization criterion (cost function):

$$E=\iint\left[\left(I_xv_x + I_yv_y + I_t\right)^2+\alpha^2\left(\left\|\nabla v_x\right\|^2+\left\|\nabla v_y\right\|^2\right)\right]dx\,dy \tag{8}$$

where the magnitude of the constant value α indicates the relative significance and impact of the flow smoothness term in comparison to the gradient constraint [13]. By minimizing the cost function in (8), the image velocity v = (v_x, v_y) will be attainable through iterative numerical methods such as the Gauss-Seidel that solve appropriately the Euler-Lagrange equations [25], as follows:

$$v_x^{k+1}=\overline{v}_x^k-\frac{I_x\left(I_x\overline{v}_x^k+I_y\overline{v}_y^k+I_t\right)}{\alpha^2+I_x^2+I_y^2} \tag{9}$$

$$v_y^{k+1}=\overline{v}_y^k-\frac{I_y\left(I_x\overline{v}_x^{kk}+I_y\overline{v}_y^k+I_t\right)}{\alpha^2+I_x^2+I_y^2} \tag{10}$$

where $\overline{v}_x^k,\overline{v}_y^k$ stand for a weighted average of v_x, v_y in a neighborhood of a center pixel located at (*x, y*), and estimated using the following finite difference relations:

$$\overline{v}_{xi,j,k}^k=\frac{1}{6}\left[v_{xi-1,\ j,\ k}+v_{xi,j+1,k}+v_{xi+1,j,k}+v_{xi,j-1,k}\right] \\ +\frac{1}{12}\left[v_{xi-1,j-1,k}+v_{xi-1,j+1,k}+v_{xi+1,j+1,k}+v_{xi+1,j-1,k}\right] \tag{11}$$

$$\overline{v}_{yi,j,k}^k=\frac{1}{6}\left[v_{yi-1,j,k}+v_{yi,j+1,k}+v_{yi+1,j,k}+v_{yi,j-1,k}\right] \\ +\frac{1}{12}\left[v_{yi-1,j-1,k}+v_{yi-1,j+1,k}+v_{yi+1,j+1,k}+v_{yi+1,j-1,k}\right] \tag{12}$$

BM-BASED MOTION ESTIMATION

In the BM method, each frame is divided into distinct blocks of *b* × *b* pixels; then for each block, the best-matched block from the previous frame is associated and the motion vector for that block is obtained. Dividing each frame of size m × n into square blocks of size *b* × *b*, the search operation in the previous frame will be done for each block in the current frame. In fact, through a matching algorithm, the best-matched block is acquired, which means the motion vector for this block is calculated. The motion vector is the displacement from the current block to the best-matched block in the previous frame. In order to enhance runtime efficiency of the algorithm, a window of pixels is defined to restrict the search area (Figure 1).

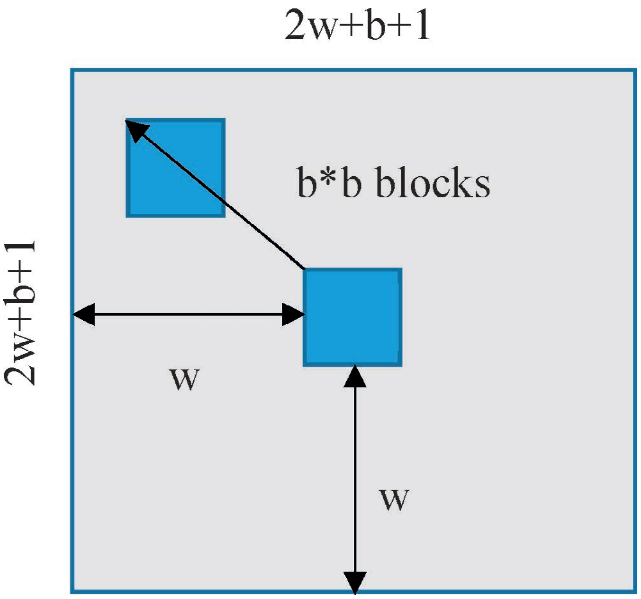


Figure 1.
Search window in current frame for BM, and the related reference frame.

In Figure 1, we define a search window with the size of ±w pixels in both vertical and horizontal directions. Accordingly, there is $(2w + 1)^2$ number of possible displacements for the block. The displacements are determined by the matching algorithms presented in the following section. Consequently, the matching method, the block, and search window sizes are three important elements in a block-based motion estimation. In lieu of a real-time and on-line implementation, we have exploited the Sum of Absolute Difference (SAD) matching method. Moreover, as illustrated in Figure 1, a *16* × *16*-block size which can grow by ±7 pixels in horizontal and vertical directions provides a flexible search window. In the following, the reasons for the decisions made are presented. Moreover, the matching algorithm is also used to compare the block of the current frame with the previous one in the selected search window. Depending on the matching algorithm, a cost function can be defined. The block that minimizes the cost function is the best matched block. The most common matching methods used for a block-based motion estimation are Mean Square Error (MSE) and SAD [10]. The related formulation is given in the following equations:

$$MSE\left(w_x,w_y\right)=\frac{1}{N\times N}\sum_{i=x}^{x+N-1}\sum_{j=y}^{y+N-1}\left[I_t(i,j)-I_{t-1}\left(i+w_x,j+w_y\right)\right]^2 \\ (v_x,v_y)=\min_{-w\le w_x.w_y\le +w} MSE(w_x,w_y) \tag{13}$$

$$\mathrm{SAD}\left(w_x, w_y\right) = \sum_{i=x}^{x+N-1}\sum_{j=y}^{y+N-1}\left|I_t\left(i,j\right) - I_{t-1}\left(i+w_x, j+w_y\right)\right|$$

$$(v_x, v_y) = \overset{\min}{_{-w \le w_x . w_y \le +w}} \mathrm{SAD}(w_x, w_y) \qquad (14)$$

in which I_t and I_{t-1} are the intensity functions of the current and the previous frames, respectively. According to the assumption of a ±w pixels displacement both in horizontal and vertical directions, the proposed cost function is computed for every $(2w + 1)^2$ possible candidate block to find the best matched block. Since, in MSE, the number of operations is proportional to the square of the block size, it is more efficient to exploit the SAD in real-time applications. In fact, possibly, when a more powerful processor is attainable, the MSE can be adopted for further enhancement of the precision of the matching process. Furthermore, the BM precision depends on the block size; the smaller the block size, the better estimation. However, very small block size increases computational complexity. In this article we use a 16 × 16 block size. The search range is defined by the maximum allowed displacement in both horizontal and vertical directions (Figure 1). Search window size has a great impact on computational complexity and estimation precision. A small search window degrades results of estimation for a rapid motion; on the other hand, a large search window affects computational complexity and runtime. Accordingly, a seven-pixel window is suggested for situations in which the robot encounters relatively high displacements.

VELOCITY DETERMINATION

To transform the pixel-based motion field into a metric velocity, we use the PM, which is widely used in the literature [26]. Essentially, the PM projects a three-dimensional (3D) point in the real-world onto a 2D image plane in the camera frame. According to the PM, projected pixel coordinates of a 3D point (P = [X Y Z]^T), resolved in the camera frame, are calculated by the following equation:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} X \\ Y \end{bmatrix} \qquad (15)$$

where Z-axis is the optical axis of the camera, f is the focal length, and the origin of the camera frame is the center of projection. A visual illustration of the presented concepts is exposed in Figure 2. As a result, pixel velocity originates from the derivative of (15) as noted below:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \frac{f}{Z^2}\begin{bmatrix} ZV_x - XV_z \\ ZV_y - YV_z \end{bmatrix} \qquad (16)$$

where V_x, V_y, and V_z are the velocity components in *x-*, *y-*, and z-directions of point P respectively. v_x and v_y are also the image velocity components and Z is the distance between the feature point on the ground and the center of projection of the camera [27]. Since the camera has both translational and rotational motions, the velocity of point P on the 2D image plane in the camera frame in terms of translational velocity (T) and rotational velocity (ω), is obtained by:

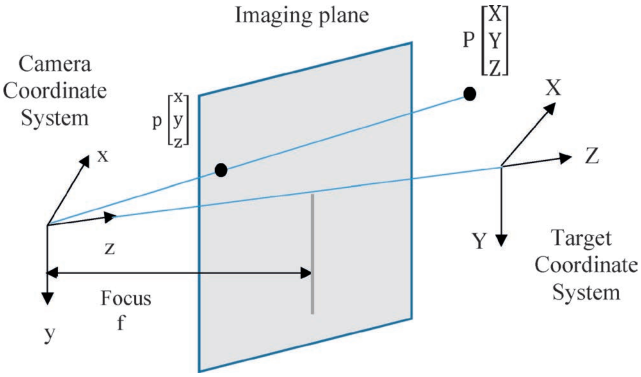

Imaging plane
Camera
Coordinate
System
$p\begin{bmatrix}x\\y\\z\end{bmatrix}$
$P\begin{bmatrix}X\\Y\\Z\end{bmatrix}$
x
z
y
Focus
f
X
Z
Y
Target
Coordinate
System

Figure 2.
Projection of a 3D point on a 2D plane in the Pinhole model.

$$V = -T - \omega \times P \qquad (17)$$

Substituting (17) in (16), the equations of motion field, in case of assuming constant flight altitude, are derived as [28]:

$$v_x = \frac{-T_x f}{Z} - \omega_y f + \omega_z y \qquad (18)$$

$$v_y = \frac{-T_y f}{Z} + \omega_x f - \omega_z x \qquad (19)$$

in which T_x and T_y are translational velocity components of the camera, and ω_x, ω_y, and ω_z are angular ones, respectively.

EXPERIMENTAL SETUP

Towards practical implementation of the suggested visual navigation system, an aerial testbed has been prepared. To this end, we used the Tarot IRON MAN 650 frame, with a 650 mm motor to motor span, which weighs 476 grams. To accomplish an autonomous flight, we adopted a *PIXHAWK* autopilot, including an open source package, designed and built at ETH University. The indicated flight controller involves an inertial navigation system, consisting of a barometer and a trio of accelerometers, gyroscopes, and magnetometers. This microcomputer is established on a 168 MHz Cortex M4F central processing unit (CPU) with 2 MB flash memory and 264 KB random-access memory (Figure 3). Subsequently, to implement the suggested optical navigation system, we adopted a *Px4flow* sensor which includes open source C++ codes, embedded open hardware, and a metric optical camera [28]. The prescribed vision sensor is connected to the autopilot via I2C port. The camera has a maximum resolution of 752 × 480 pixels and renders the optical data at 400 Hz. Unlike many available sensors, it also works indoors and in low outdoor light conditions without an illumination LED. The vision sensor contains a 168 MHz Cortex M4F CPU, 752 × 480 MT9V034 image sensor, and an L3GD20 3D Gyro. Performance of this gyro is guaranteed by 0.15 *deg/*

min bias drift in *x*- and *y*-directions and 0.4 *deg/min* bias drift in the z-direction. Moreover, the update rate is 780 Hz and the maximum measurable angular rate is 500 °/*s*.

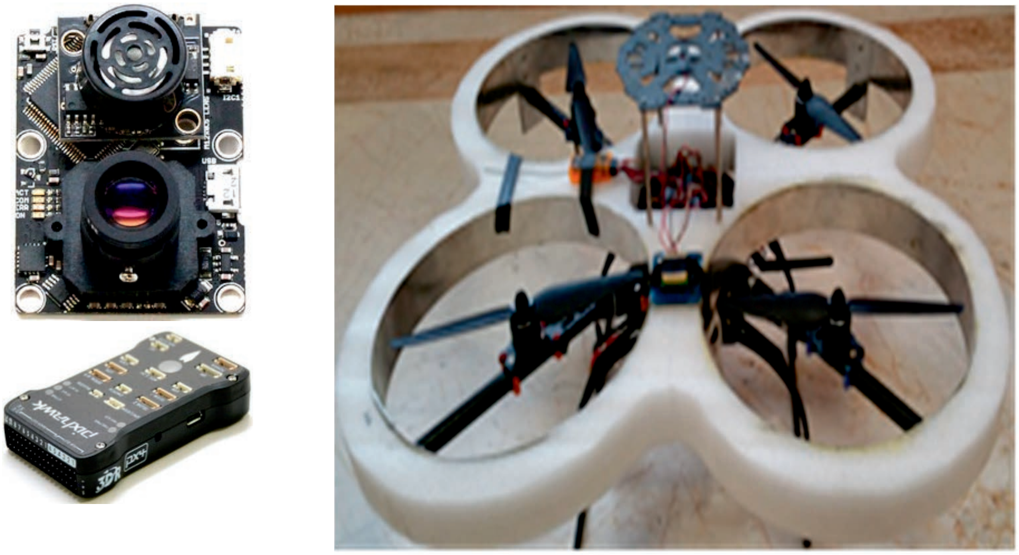


Figure 3.
Px4flow sensor, Pixhawk autopilot, and the developed vehicle.

In order to implement a real-time BM, as explained previously, we first divided each frame into blocks with dimensions of 8 x 8 pixels and also adopted the related search windows with dimensions of ±4 pixels. Subsequently, we used the Harris Corner Detector (HCD) [29] to detect salient points and examined the suitability of pixel locations for OF reckonings. To indicate the best match for the detected salient points in the previous step, the SAD for the adopted block in the current frame and every block in the previous frame is computed. Then the motion vector (u, v), which stands for the displacement of a block in two consecutive frames, in *x*- and *y*- directions (Figure 1) is determined. It should be noted that the precision in vision-based navigation systems depends heavily on the quality of images captured by the vision sensor [30]. At this phase, to show the effects of the illumination condition and image quality on the performance of the optical navigation system, we defined a quality parameter/factor. This factor is calculated based on the number of pixel locations which are suitable for flow estimation as noted below:

$$Q = \frac{255\times j}{k\times k} \tag{20}$$

in which *Q* is the image quality and *j* is the number of tiles to check. If there is any suitable pixel location, based on HCD, the corner detector uses minimum eigenvalues [29] that act as threshold values for the quality level of a corner. If a corner is below the quality threshold value, that corner is not tracked. In (20), *k* indicates the number of corners which are tracked. In fact, this parameter shows the effect of illumination on the precision of optical flow calculations and is quantified from 0 to 255. Zero value is acquired when there is no suitable pixel location under the worst illumination condition. On the other hand, the quality factor reaches 255 when all pixel locations in the search window are suitable for OF calculations under a perfect illumination condition.

TEST RESULTS

As stated before, in order to evaluate the performance of the mentioned three optical navigation methods (LK, HS, and BM), an off-line implementation was conducted on captured images of the vision sensor mounted on an in-house-built quadrotor. Since the quality of the images and the degree of the precision for navigation purposes depend on the vehicle's orientation during the flight [5], in this research we tested the proposed optical navigation system at almost constant altitudes.

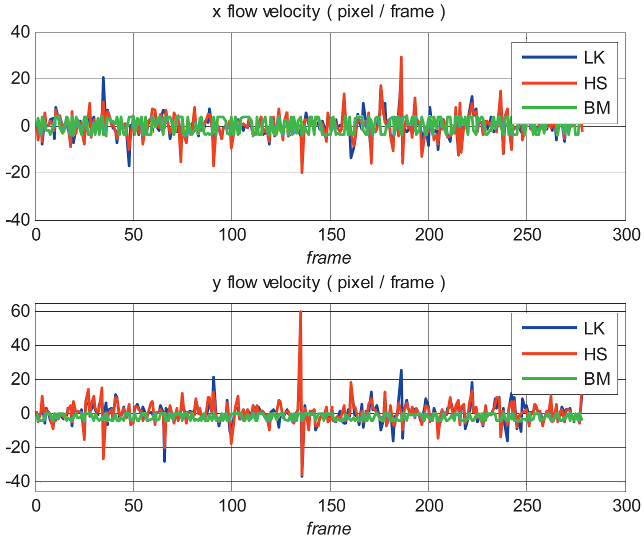


Figure 4.
Components of pixel-based velocity estimated by LK, HS, and BM methods.

Based on the test scenarios, the vehicle was flown over the *Kooye Danesgah* area at the University of Tehran Campus. Both *x* and *y* components of the pixel velocity are calculated through LK, HS, and BM as explained previously. In each frame, every pixel's velocity is estimated and then the average of the values is assumed as the image velocity (Figure 4). As far as this research is concerned, we compared the precision of the three methods in terms of standard deviation of pixel velocity, estimated at each pixel location of the considered frame (Figure 5). Since low dispersion is associated with high precision, the lower the data dispersion, the more precise the estimation. Subsequently, as illustrated in Figure 5, since the dispersion of pixel

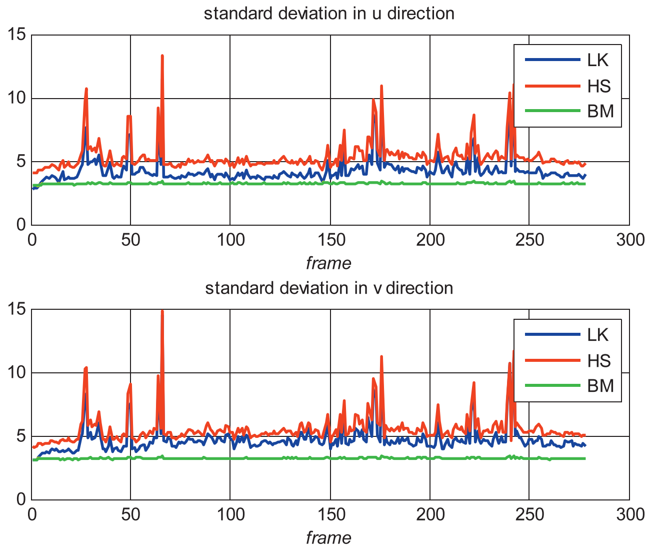


Figure 5.
Comparison between dispersion of pixel velocities of each frame in LK, HS, and BM methods.

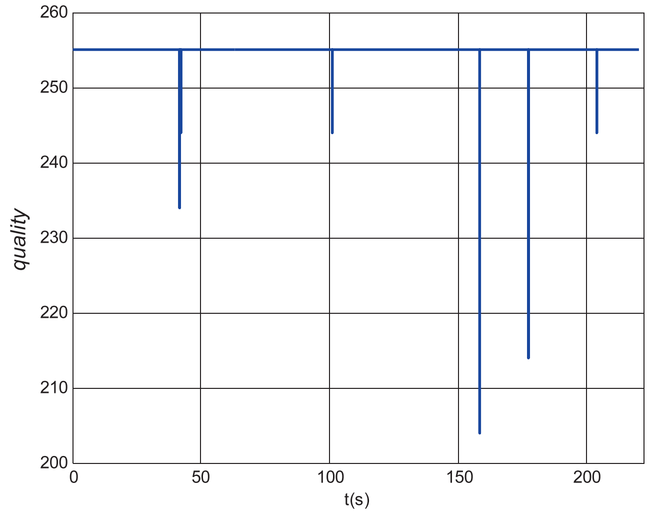


Figure 6.
Quality of images captured by vision sensor over outdoor flight under desired illumination condition.

velocities obtained from the BM method is lower than the that from the two other methods, the BM method is more accurate than the others. Hence, we implemented a real-time BM navigation system on our in-house quadrotor using the Px4flow sensor.

As explained previously, in real-time implementations, suitable pixel locations for OF calculations are detected through the HCD, and the illumination condition is evaluated through the quality parameter. We implemented BM in real-time for an outdoor flight over *Kooye Daneshgah* under a desired illumination condition where the quality parameter is almost constant and perfect (Figure 6). Pixel velocity obtained from OF (Figure 7a) and onboard estimated angular velocity (Figure 7b), is transmitted to the ground station through telemetry. In addition, replacing the sensor values in (18) and (19), *x* and *y* components of metric velocity of the aerial robot are obtained (Figure 7c). Scientific evidence of the flight path is provided

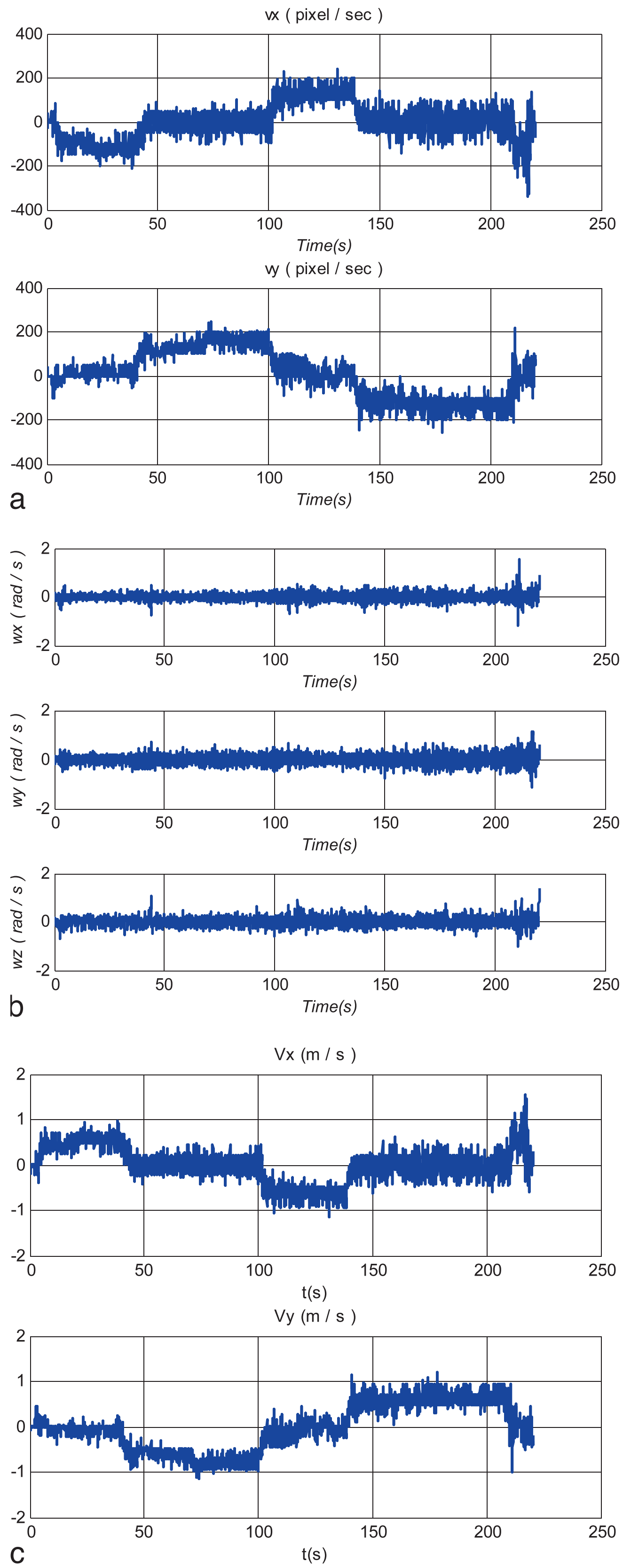


Figure 7.
(a) Components of pixel velocity over outdoor flight under desired illumination condition. (b) Components of angular velocity sensed by gyroscope. (c) Components of the calculated metric velocity of the quadrotor.

based on ground truth data created with the help of Google Earth by choosing four waypoints around a specific location in *Kooye Daneshgah*, as shown in Figure 8a. For better understanding and verification of the results, the estimation outcomes are illustrated in a 3D plot, together with an augmented graphical view with the aid of Google Earth (Figure 8b).

In order to demonstrate the effects of lighting on the OF predictions, we prepared an additional examination in the same place. Unlike the former test, under an adverse illumination condition, the image quality turned to be undesirable (Figure 9).

As previously explained, the first test was done under desired illumination conditions to ensure that captured images by Px4flow sensor be suitable for OF calculations in almost all the moments of the flight test. Afterwards, the pixel velocity of the vehicle, the metric velocity, and position are calculated as well. In the new test, as can be seen in Figure 9, from seconds 10 to 65, there is almost no any suitable image for OF calculations. In other words, through applying HCD, there is not any suitable pixel location for pixel velocity computations. Accordingly, pixel velocity is estimated nearly zero through this period of time and at the succeeding moments of the test.

In the following, Figure 10 exhibits that the velocity estimations suffer from negative spikes in the *y*-direction and symmetrical ones in *x* components. As uncertainty increases under adverse lighting conditions, statistical characteristics of the predictions, such as standard deviation of pixel velocity, confront negative impacts, prominently from unpredicted degradations of the images. Consequently, low image quality due to improper illumination conditions causes an increase in gross errors in the velocity predictions (Figure 10). Accordingly, this discrepancy appears in position results, in comparison with the ground-truth data provided with the help of Google Earth that are acquired by choosing four waypoints around a specific location in *Kooye Daneshgah,* and in the flight path under desired illumination conditions, as well (Figure 11a).

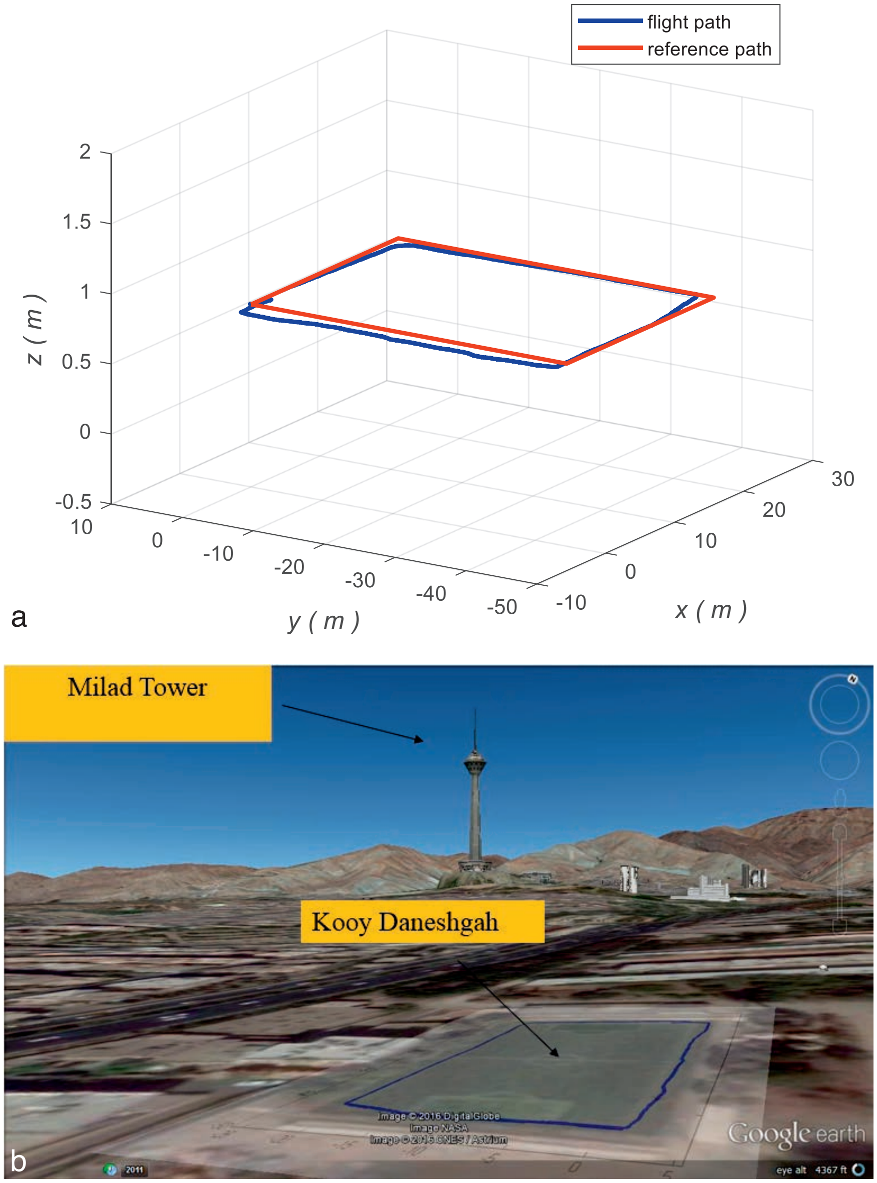


Figure 8.
(a) The 3D traveled path. (b) Augmented graphical observation on Google Earth.

Finally, to evaluate the robustness of the presented method, an indoor flight test was performed to evaluate the ONS under different illumination conditions (Figure 11b). Analogous to the outdoor tests in adverse illumination conditions, shown in Figure 11b, an appropriate image for OF reckoning can hardly be found during all of the flight time. Hence, pixel velocity is estimated practically zero in this period of time as well. Due to the adverse illumination conditions, increasing of the gross errors on the velocity (Figure 11c) and position estimations (Figure 11d) is inevitable in indoor as well as outdoor conditions. Similar to the outdoor flight test, scientific evidence of flight path is provided based on ground truth data created with the help of Google Earth by choosing four waypoints around a specific location in *Guidance and Navigation Laboratory of University of Tehran*, as shown in Figure 11d.

CONCLUSIONS

In this article, we first implemented three off-line OF algorithms (LK, HS, and BM), in order to compare them in terms of precision and stability of velocity and position estimations. For the predictions, during the OF process, the image velocity is assumed as the average of all pixel velocities of the considered frame. According to data dispersion criteria in every frame, the BM-based OF is more accurate

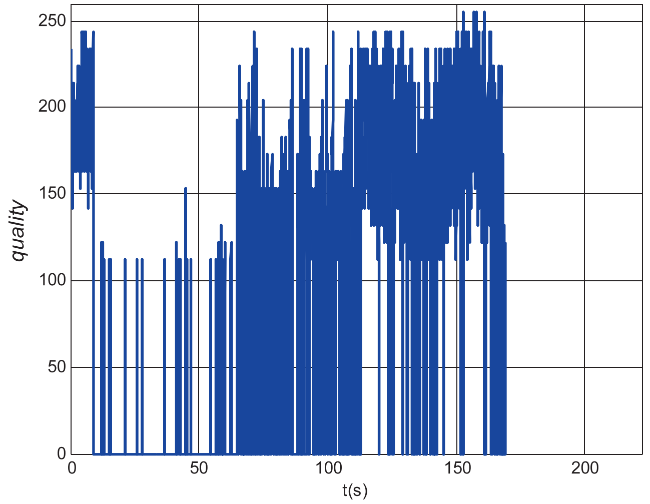


Figure 9.
Image quality (see (20)) under adverse lighting conditions.

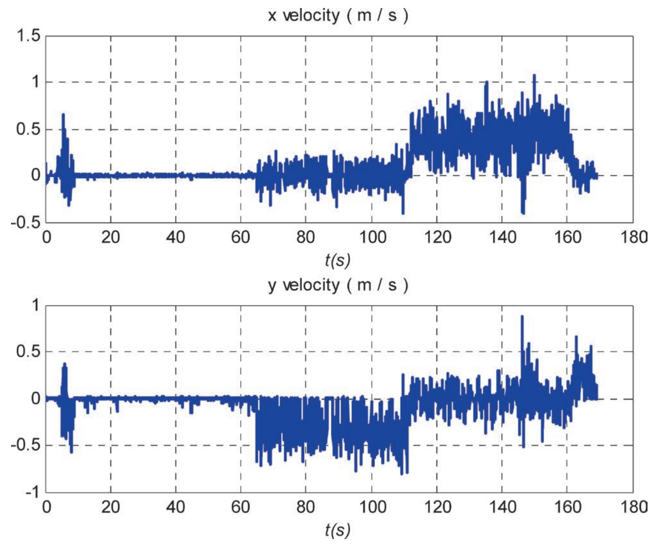


Figure 10.
Predicted velocity error under adverse lighting conditions.

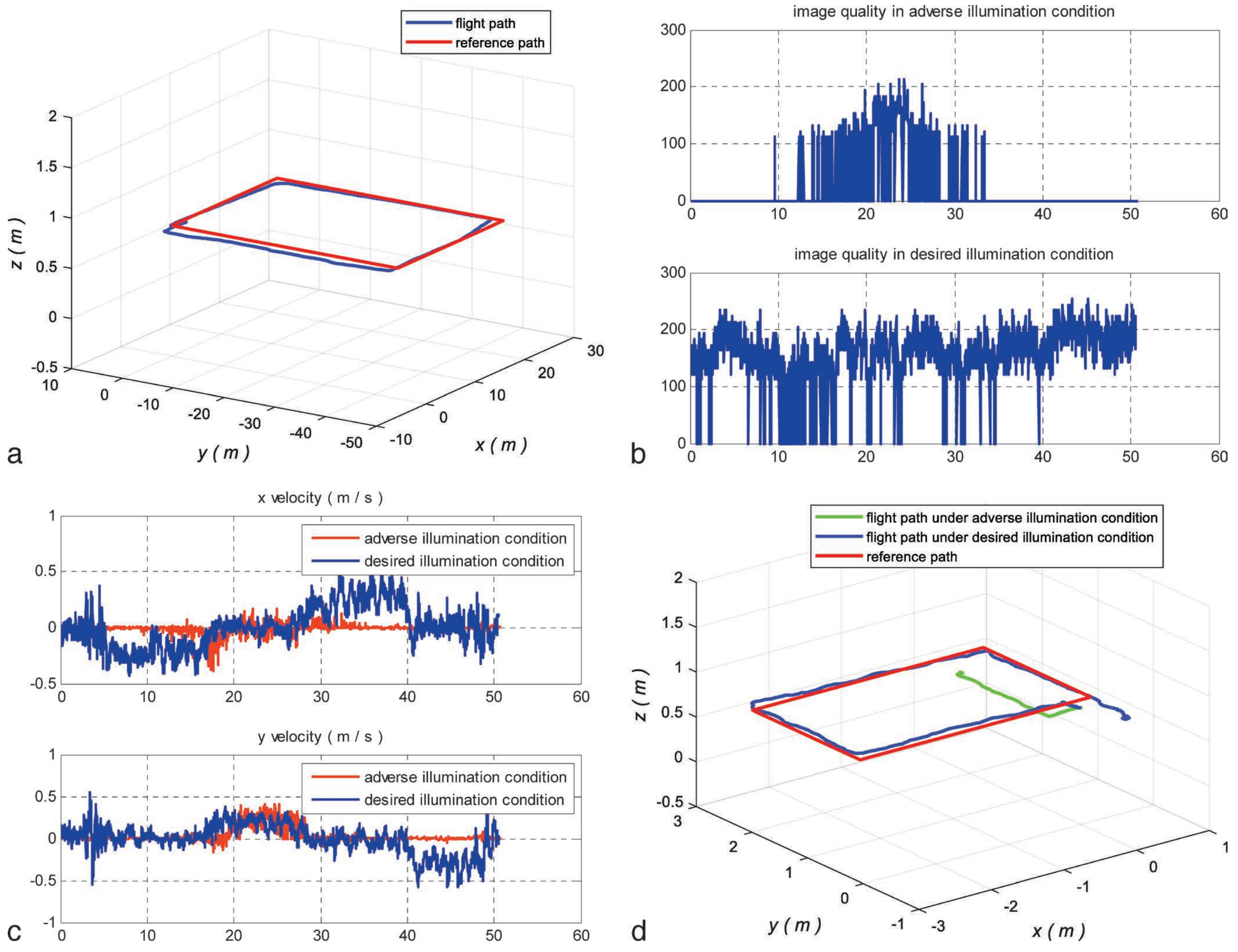


Figure 11.
(a) Predicted position error under adverse lighting conditions in comparison with reference path and the flight path under desired illumination condition. (b) Image quality factor (see (20)) under adverse and desirable optical conditions of indoor flight test. (c) Velocity components under adverse and desirable optical conditions for indoor test. (d) 3D position predictions under adverse and desirable optical condition for indoor test.

than the other ones. For this reason, that one was implemented in real-time using the OF sensor. Successively, indoor and outdoor flight-tests under different illumination conditions were accomplished for investigation of the sensitivity of the OF velocity estimations to the illumination conditions. It was exhibited that the precision of the ONS is remarkably dependent on lighting conditions. Considering those demonstrated inabilities of an ONS, future works will focus on the fusion of the OF data with a GPS/INS integrated navigation system to enhance the precision and stability of the estimations. Lastly, for further enhancement of the aerial robot guidance, navigation, and control system, a vision based control task such as OF altitude hold, obstacle avoidance, and a formation control system can be realized and incorporated into the current framework. ◆

REFERENCES

[1] Chao, H., Cao, Y., and Chen, Y. Autopilots for small unmanned aerial vehicles: A survey. *International Journal of Control, Automation and Systems,* Vol. 8, 1 (2010), 36–44.

[2] Chahl, J. S., Srinivasan, M. V., and Zhang, S.-W. Landing strategies in honeybees and applications to uninhabited airborne vehicles. *The International Journal of Robotics Research,* Vol. 23, 2 (2004), 101–110.

[3] Esch, H., and Burns, J. Distance estimation by foraging honeybees. *The Journal of Experimental Biology,* Vol. 199, 1 (1996), 155–162.

[4] Srinivasan, M. V. Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics. *Physiological Reviews,* Vol. 91, 2 (2011), 413–460.

[5] Amiri Atashgah, M. A., and Malaek, S. M. B. A simulation environment for path and image generation in an aerial single-camera vision system. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 225, no. 5 (2011): 541-558.

[6] He, R., Prentice, S., and Roy, N. Planning in information space for a quadrotor helicopter in a GPS-denied environment. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA 2008)*, 2008, 1814–1820.

[7] Hightower, J., and Borriello, G. Location systems for ubiquitous computing. *Computer,* 8 (2001), 57–66.

[8] Thrun, S., Burgard, W., and Fox, D. *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[9] Kendoul, F., Fantoni, I., and Nonami, K. Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles. *Robotics and Autonomous Systems,* Vol. 57, 6 (2009), 591–602.

[10] Metkar, S., and Talbar, S. Performance evaluation of block matching algorithms for video coding. In *Motion Estimation Techniques for Digital Video Coding*. Springer, 2013, 13–31.

[11] Chao, H., Gu, Y., and Napolitano, M. A survey of optical flow techniques for robotics navigation applications. *Journal of Intelligent & Robotic Systems,* Vol. 73, 1–4 (2014), 361–372.

[12] Lucas, B. D., and Kanade, T. An iterative image registration technique with an application to stereo vision. In *Proceedings of the IJCAI*, 1981, Vol. 81, 674–679.

[13] Horn, B. K., and Schunck, B. Determining optical flow-a retrospective. *Artificial Intelligence,* Vol. 59, (1994), 81–87.

[14] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision,* Vol. 60, 2 (2004), 91–110.

[15] Srinivasan, M. V. An image-interpolation technique for the computation of optic flow and egomotion. *Biological Cybernetics,* Vol. 71, 5 (1994), 401–415.

[16] Conroy, J., Gremillion, G., Ranganathan, B., and Humbert, J. S. Implementation of wide-field integration of optic flow for autonomous quadrotor navigation. *Autonomous Robots,* Vol. 27, 3 (2009), 189–198.

[17] Prashanth, K., Shankpal, P., Nagaraja, B., Kadambi, G. R., and Shankapal, S. Real time obstacle avoidance and navigation of a quadrotor MAV using optical flow algorithms. *SASTECH Journal,* Vol. 12, 1 (2013), 31–35.

[18] Shukla, D., and Patel, E. Speed determination of moving vehicles using Lucas-Kanade Algorithm. *International Journal of Computer Applications Technology and Research,* Vol. 2, 1 (2013), 32–36.

[19] Gageik, N., Strohmeier, M., and Montenegro, S. An autonomous UAV with an optical flow sensor for positioning and navigation. *International Journal of Advanced Robotic Systems,* Vol. 10, 2013.

[20] Yang, J.-S. Estimation of vehicle's lateral position via the Lucas-Kanade optical flow method. *WSEAS Transactions on Systems*, Vol. 8, 11 (2012), 349–363.

[21] Barron, J. L., Fleet, D. J., and Beauchemin, S. S. Performance of optical flow techniques. *International Journal of Computer Vision,* Vol. 12, 1 (1994), 43–77.

[22] Otte, M., and Nagel, H.-H. Optical flow estimation: Advances and comparisons. In *Computer Vision—ECCV'94*. Springer, 1994, pp. 49–60.

[23] Khalil, F. F., and Payeur, P. Optical flow techniques in biomimetic UAV vision. In *Proceedings of the 2005 International Workshop on Robotic Sensors: Robotic and Sensor Environments*, 2005, 14–19.

[24] McCarthy, C., and Bames, N. Performance of optical flow techniques for indoor navigation with a mobile robot. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA'04)*, 2004, Vol. 5, 5093–5098.

[25] Barron, J. L., and Thacker, N. A. Tutorial: Computing 2D and 3D optical flow. Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, 2005.

[26] Kannala, J., and Brandt, S. S. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 28, 8 (2006), 1335–1340.

[27] Chhaniyara, S., Bunnun, P., Seneviratne, L., and Althoefer, K. Optical flow algorithm for velocity estimation of ground vehicles: A feasibility study. *International Journal on Smart Sensing and Intelligent Systems,* Vol. 1, 1 (2008), 246–268.

[28] Honegger, D., Meier, L., Tanskanen, P., and 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 (ICRA)*, 2013, 1736–1741.

[29] Harris, C., and Stephens, M. A combined corner and edge detector. *Alvey Vision Conference,* Manchester, UK, 1998, Vol. 15, 10.5244.

[30] Atashgah, M.A.A., and Malaek, S.M.B. Prediction of aerial-image motion blurs due to the flying vehicle dynamics and camera characteristics in a virtual environment. In *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 227, 7 (2013), 1055–1067.