Online Digital Image Stabilization for an Unmanned Aerial Vehicle (UAV)

Wahyu Rahmaniar[1]* and Amalia Eka Rakhmania[2]
[1]Department of Electrical Engineering, National Central University, Taoyuan, Taiwan
[2]Department of Electrical Engineering, State Polytechnic of Malang, Malang, Indonesia
Email: [1]wahyu.rahmaniar@gmail.com
*Corresponding Author

***Abstract*— The Unmanned Aerial Vehicle (UAV) video system uses a portable camera mounted on the robot to monitor scene activities. In general, UAVs have very little stabilization equipment, so getting good and stable images of UAVs in real-time is still a challenge. This paper presents a novel framework for digital image stabilization for online and real-time applications using a UAV. This idea aims to solve the problem of unwanted vibration and motion when recording video using a UAV. The proposed method is based on dense optical flow to select features representing the displacement of two consecutive frames. K-means clustering is used to find the cluster of the motion vector field that has the largest members. The centroid of the largest cluster was chosen to estimate the rigid transform motion that handles rotation and translation. Then, the trajectory is compensated using the Kalman filter. The experimental results show that the proposed method is suitable for online video stabilization and achieves an average computation time performance of 47.5 frames per second (fps).**

Keywords—K-means, Kalman filter, image stabilization, optical flow, UAV.

I. Introduction

Recently, mobile robots are increasingly needed to help in everyday human life such as humanitarian assistance [1], services industry [2], environmental monitoring [3], and security [4]. The Unmanned Aerial Vehicle (UAV) often use for surveillance [5], target tracking [6], navigation [7], and localization [8] due to its flexibility of moving in any area.

The aerial image sequence is obtained from a portable camera set on an Unmanned Aerial Vehicle (UAV) to monitor activity [5]. Since, in general, UAVs have very little stabilization equipment, they can be seen as unstable platforms with different intense, unexpected vibrations. As such, images taken using a UAV are of a lower quality than those captured from a stationary camera. Thus, target or landmark recognition often fails due to loss of focus and blurriness caused by the vibrations in the image. Besides, the challenge of this work is to find an effective solution to stabilize the sequence of images for online applications

Several approaches have been proposed to solve the problem of image stabilization from video. There are two types of image stabilization techniques for UAVs: physical stabilizer and digital stabilizer. Physical stabilizer, which include Optical Image Stabilization (OIS) [9]-[10] and Electric Image Stabilization (EIS) [11]-[12], are equipped with several sensors (*e.g.*, gyro sensor, hall sensor, and CCD image sensor) and actuators (*e.g.*, stepping motor, piezo-actuator, and servo motor) so they cost more than digital stabilizer and also add to the weight of the UAV. The OIS method manipulates lens movement using motion sensors [13] and active optical system [14] that causes several typical magnet circuit phenomena such as saturation, hysteresis, and lack of linearity. These phenomena make it difficult to control the system. The EIS method uses motion sensors and mechanical devices to compensate for camera movement so that image blurriness and vibration can be reduced. But, the stabilization rate becomes poor at high vibration frequency [15] and needs to be improved by digital image stabilization (DIS).

A DIS method generally consists of camera motion estimation and compensation that have feedforward control and system compactness [16]-[17]. Conventional techniques for DIS usually use feature points [18] such as sparse optical flow [19]-[20] and select some useful point features using Shi-Tomasi corner detection [21]. This method selects multiple features in the previous frame and tracks similar features in the current frame, then calculates the optical flow in the region of interest (ROI). Motion estimates can be computed with the Kalman filter [22], but this involves only a few points that may not represent the global movement of two consecutive images. Another challenge in DIS is online computation, subject to a constraint on the number of frames per second (fps). In the online application, the accumulative global motion (AGM) [23]-[24] may tend to fail when videos are running with long online duration due to accumulated motion estimation error.

Several DIS approaches have been proposed to stabilize the sequence of images taken by UAVs [20], [25] that processes the entire series of images before stabilizing each image. As such, these methods cannot be used for online applications that require direct processing. The methods in [18], [22], [26], and [27] performs image stabilization online and in real-time but not for the camera on the UAV, so these methods are not sufficient for handling images on a UAV which has more unexpected motion and vibration.

To solve the problem of online and real-time image stabilization in a UAV, a new framework is proposed in this paper. The digital system uses dense optical flow to

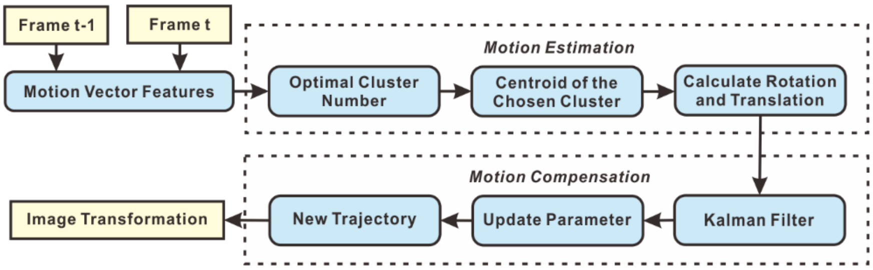


Fig. 1. System overview

determine the motion vector field between two consecutive frames then uses K-means clustering to classify the motion estimation. A Kalman filter compensates for the trajectory.

The remainder of the paper is organized as follows. Section II introduces the overview system of the proposed method. Section III presents the results and discussion. Finally, conclusions are drawn in Section IV.

II. The Proposed Method

A. System Overview

The framework of the proposed aerial image stabilization (AIS) is shown in Fig. 1. In the first step, the proposed method extracts the motion vectors field from two successive consecutive frames, which are used as a feature to estimate the motion model parameters. The proposed AIS adopts the K-means algorithm [28] as unsupervised learning to find global motion on the frame based on the centroid of the cluster with the largest member. An optimal cluster number was found based on the gap statistics to avoid outliers in the clustering [29]. Then, the motion estimation is determined from the selected cluster centroid.

Fig. 2 shows the system overview of the proposed method. Fig. 2(a) illustrates the distribution of the motion vectors field in the image. The red color represents a substantial region that does not contain any objects and has a zero motion vector value. The blue color represents dynamic objects or pixels that have a higher displacement value than other regions. The green color represents static objects or pixels that have a lower displacement value than the blue region. In this case, the green color represents camera movement, because the displacement of the static object is caused by the movement of the camera. Fig. 2(b) illustrates the clustering of motion vectors that have a value higher than zero. The cluster with the largest member (green pixels) is selected, and its centroid value is used to estimate the image motion. Fig. 2(c) shows the smoothed trajectory obtained from the motion compensation of each frame.

The proposed motion compensation also aims to avoid large gaps in motion estimation caused by unwanted UAV movements. If the estimated motion of the previous frame and the current frame are too different, the image transformation will shift too far from the original image plane.

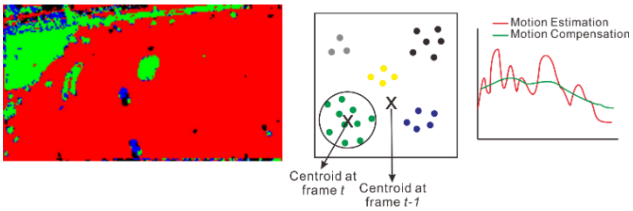


Fig. 2. System overview: (a) Motion vector distribution in the image. (b) Clusters of motion vectors. (c) Motion compensation to smooth the trajectory.

B. Motion Vectors Distribution

Farneback optical flow [30] is used to estimate the field of motion vectors of the dense optical flow in each frame. This optical flow method is based on polynomial expansions to compute each grid point in the image as a local neighborhood. Polynomial expansion transforms based on translation and rotation from two consecutive frames derived using coefficients. The point of vector displacement is estimated in a two-dimensional function.

For every two consecutive frames, the previous and current frames are defined as $f(t-1)$ and $f(t)$, respectively, at time *t*. Then, the image size is reduced by 50% of the original size and the color is changed into a grey-scale. Let $f'(t-1)$ and $f'(t)$ denotes the new image of $f(t-1)$ and $f(t)$, respectively.

Given the pixel position *J* with the pixel's neighbourhood *M*(*J*), the local coordinate system at $f'(t-1)$ can be approximated by

$$f_p(t-1) = p^T A(t-1)p + b^T(t-1)p + c(t-1) \tag{1}$$

where $p \in M(J)$, $A(t-1)$ is a symmetric matrix, $b(t-1)$ is a vector, and $c(t-1)$ is a scalar. Assuming the image brightness is constant, the local coordinate system at $f'(t)$ can be obtained by

$$f_p(t) = p^T A(t)p + b^T(t)p + c(t) \tag{2}$$

A new signal is constructed by $f_p(t) = f_{(p-d(t))}(t-1)$, then the new local coordinate system of the two input images can be approximated by

$$f_p(t) = p^T A(t-1)p + \left(b(t-1) - 2A(t-1)d(t)\right)^T p \\ +d^T(t)A(t-1)d(t) - b^T(t-1)d(t) + c(t) \tag{3}$$

where d is a global displacement, and by equating the coefficients in (2) and (3), the vector displacement can be calculated by

$$b(t) = b(t-1) - 2A(t-1)d(t) \tag{4}$$

where $A(t) = A(t-1)$, so that total displacement can be solved by

$$d(t) = -\frac{1}{2}A^{-1}(t)\left(b(t) - b(t-1)\right) \tag{5}$$

The initial estimation of the displacement vector is set as (0,0). Thus, the ROI of the two frames has the same path.

C. Motion Vectors Clustering

The motion vector matrix is calculated for every two consecutive frames of the image sequences in the next step. The initial number of clusters is estimated by $N = \sqrt{n/2}$ where n represents the total number of the displacement matrices in (5). In this step, the K-means algorithm ends with several empty clusters results. So that the size of the cluster validity based on statistical hypothesis testing is used to obtain the optimal number of N clusters.

The gap statistics compare the similarity in clusters for a dataset with the expected cluster similarities of the data points resulting from the null-model reference distribution at each N value. The sum of the distance of all points from the cluster r (C_r) taken in pairs for i and i', return the number of points in the cluster to the centroid distance by

$$D_r = \sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ij} - x_{i'j})^2 \tag{6}$$

where x is the data of C_r, $i = 1, 2, ..., n$ is the points of every cluster, and $j = 1, 2, ..., m$ is the total cluster in the iteration.

The size of the dispersion as the sum of squares in the cluster collected can be computed by

$$W_N = \sum_{r=1}^{N} \frac{D_r}{2\,|\,C_r\,|} \tag{7}$$

Then, the gap statistics can be computed as a $log(W_N)$ ratio to the expectation as

$$Gap_N = E_n\left\{\log(W_N)\right\} - \log(W_N) \tag{8}$$

where E_n is the expectation of a new reference dataset with a uniformly generated maximum entropy of the same boundary and size n from the original dataset. The optimal number of clusters is selected from the N at the highest associated gap as

$$Gap_N \geq Gap_{N-1} \tag{9}$$

After obtaining the optimal number of clusters in (9), each cluster member is searched using the K-means algorithm. The centroid of the cluster with the largest member is selected, as shown in Algorithm 1.

```
Algorithm 1: Algorithm for centroid selection

Input: Set of motion vectors $x_1, \ldots, x_m \in d(t)$
Initial: Cluster centroids $\mu_1, \ldots, \mu_N$ are chosen randomly

//Predict N cluster centroids and label $c_i$ for each data point
FOR $n$ = 1 to ITERATION
    1. For every i: $c_i = arg \min_j \|x_i - \mu_j\|^2$
    2. For every j: $\mu_j = \frac{\sum_{i=1}^{m} 1\{c_i=j\}x_i}{\sum_{i=1}^{m} 1\{c_i=j\}}$
END

IF $c_i > 0$

//Select the centroid from a cluster with the biggest members
    1. $N_{best} = arg \max_j \sum_{i=1}^{m}\{c_i = j\}$
    2. $\mu_{best} = \frac{\sum_{i=1}^{m} 1\{c_i=N_{best}\}x_i}{\sum_{i=1}^{m} 1\{c_i=N_{best}\}}$
END

Output: Centroid from the cluster with the largest members
```

The centroid selected from the $f'(t-1)$ and $f'(t)$ are denoted as $\left(\mu_x(t-1), \mu_y(t-1)\right)$ and $\left(\mu_x(t), \mu_y(t)\right)$, respectively. Then, the value of the translation $\left[T_x(t), T_y(t)\right]$ and the rotation $\theta(t)$ can be computed by

$$T_x(t) = \mu_x(t) - \mu_x(t-1) \tag{10}$$

$$T_y(t) = \mu_y(t) - \mu_y(t-1) \tag{11}$$

$$\theta(t) = \tan^{-1}\left(T_x(t)/T_y(t)\right) \times \left(180/\pi\right) \tag{12}$$

D. Motion Compensation

Unwanted UAV movement causes the motion estimation to fail because the scene of images of $f'(t-1)$ and $f'(t)$ are very different and makes a massive displacement of the trajectory. Therefore, to avoid a rough transformation image, a Kalman filter [5] is used to smooth the trajectory and generate a new transformation set for each frame.

The Kalman filter consists of two steps: prediction and update. The prediction step estimates the trajectory at $f'(t-1)$ as $\hat{s}(t) = \left[\hat{T}_x(t), \hat{T}_y(t), \hat{\theta}(t)\right]$ by $\hat{s}(t) = s(t-1)$, and the error covariance is compensated by $\hat{e}(t) = e(t-1) + \alpha$ where α is the noise covariance of the process. The Kalman gain can be obtained by

$$K(t) = \frac{\hat{e}(t)}{\hat{e}(t) + \beta} \tag{13}$$

where β is the noise covariance of the measurement. Then, the measurement correction step compensates the trajectory state by

$$s(t)=\hat{s}(t)+K(t)\big(Z(t)-\hat{s}(t)\big) \tag{14}$$

where $s(t)=\left[\tilde{T}_x(t),\tilde{T}_y(t),\tilde{\theta}(t)\right]$ and $Z(t)$ are the actual measurement of the trajectory and can be computed by

$$Z(t)=\sum_{\tau=1}^{t-1}\Big[\big(\bar{T}_x(\tau)+T_x(t)\big),\big(\bar{T}_x(\tau)+T_x(t)\big),\big(\bar{\theta}(\tau)+\theta(t)\big)\Big] \tag{15}$$

where $Z(t)=\left[Z_x(t),Z_y(t),Z_\theta(t)\right]$, and $\bar{T}_x(\tau)$, $\bar{T}_y(\tau)$, and $\bar{\theta}(\tau)$ are the accumulated trajectory from the first frame to the current frame. Then, the new trajectory can be obtained by

$$\begin{aligned}\left[\bar{T}_x(t),\bar{T}_y(t),\bar{\theta}(t)\right]&=\left[T_x(t),T_y(t),\theta(t)\right]\\&+\left[\Delta_x(\tau),\Delta_y(\tau),\Delta_\theta(\tau)\right]\end{aligned} \tag{16}$$

where the trajectory differential variables are calculated by

$$\Delta_x(t)=\tilde{T}_x(t)-Z_x(t) \tag{17}$$

$$\Delta_y(t)=\tilde{T}_y(t)-Z_y(t) \tag{18}$$

$$\Delta_\theta(t)=\tilde{\theta}(t)-Z_\theta(t) \tag{19}$$

The scale factor can be computed by

$$\alpha(t)=\frac{\cos\bar{\theta}(t)}{\cos\left(\tan^{-1}\left(\dfrac{\sin\bar{\theta}(t)}{\cos\bar{\theta}(t)}\right)\right)(t)} \tag{20}$$

Finally, a new image $\bar{f}(t)$ is obtained by transforming $f(t)$ using the new trajectory in (16)

$$\bar{f}(t)=f(t)\begin{bmatrix}\alpha(t)\cos\bar{\theta}(t) & -\alpha(t)\sin\bar{\theta}(t)\\ \alpha(t)\sin\bar{\theta}(t) & \alpha(t)\cos\bar{\theta}(t)\end{bmatrix}+\begin{bmatrix}\bar{T}_x(t)\\ \bar{T}_y(t)\end{bmatrix} \tag{21}$$

III. Results and Discussion

The performance of the proposed method was evaluated using three types of aerial video sequences obtained from the UCF aerial action dataset (https://www.crcv.ucf.edu) and an online video from the UAV. The “Park” video has the fewest objects than others, and the displacement is smaller than the "Street” video. The “Street" video has the most massive displacement and larger objects than other videos. The “Car” video had the smallest displacement but contained the most vibration compared to other videos. For online shooting, we use a UAV as shown in Fig. 3 and a wireless camera with a resolution of 640×480. The UAV sends images to the ground station (PC), and the system runs a stabilization algorithm.

Experiments were carried out using a 2.30 GHz CPU with 8 GB RAM. The experimental results are summarized in Table I concerning computation time performance. The proposed method achieves an average computation time of 47.5 fps.

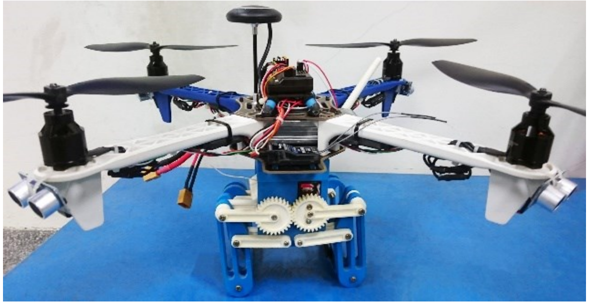


Fig. 3. The UAV aircraft.

TABLE I. PROPOSED METHOD PERFORMANCE

Video Name	Resolution	FPS
Park	960×540	42
Street	852×480	48
Car	640×480	55
Online video	640×480	45
Average		47.5

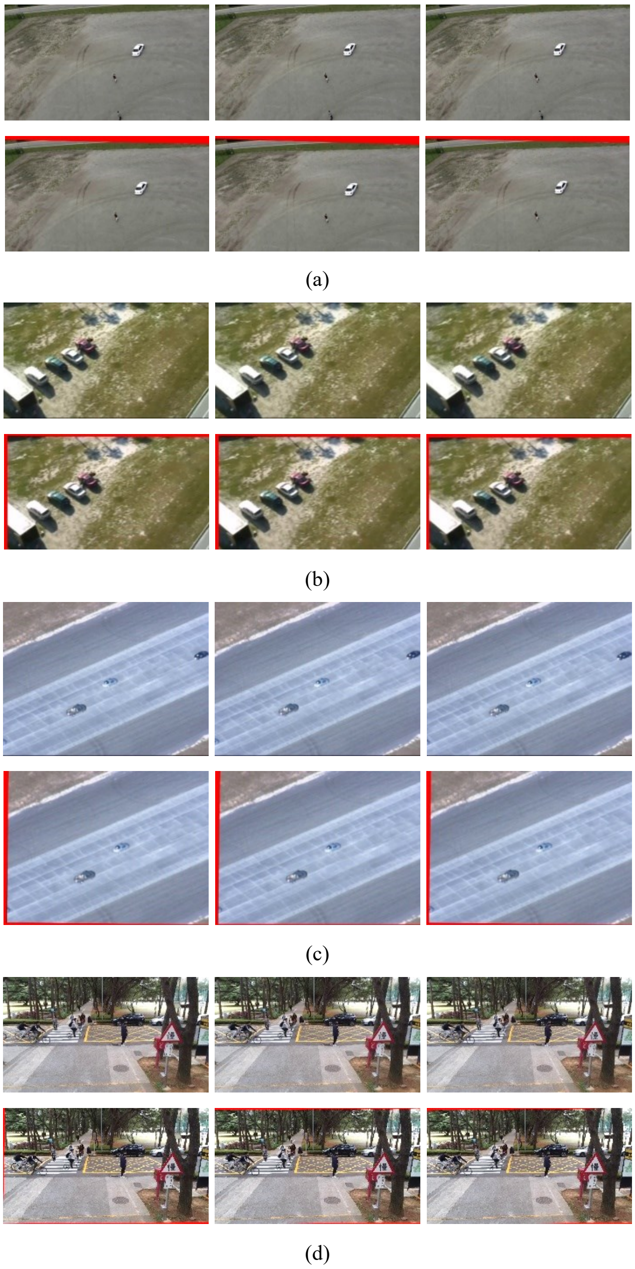
Table II summarizes the comparison based on the average time with the existing methods in [18], [22], [26], and [27] methods were chosen because they perform frame-by-frame stabilization of digital images for real-time applications and give attention to the computation time for processing.

TABLE II. PERFORMANCE COMPARISON

Method	Resolution	FPS
[18]	720×576	25
[22]	640×480	25
[26]	542×496	30
[27]	720×480	30
Proposed	960×540	42

Fig. 4 shows the comparison of the original image sequence (top) and the image stabilization results (bottom). The red area shows the shift of pixels in the image due to translation and rotation.

Fig. 5 shows the comparison of motion estimation and compensation. The results indicate that the system effectively removes jitter from the original image sequences with proper translation and rotation transformations. The proposed method can also stabilize images that have a large enough shake. The result of motion compensation for translation and rotation makes the stabilized image look natural and coherent. The stabilization results show the “Park” and “Street” videos have a smoother motion with less displacement and the “Car” video has a vibration reduction. The online video has a little transformation from the original image due to the less shaking of the UAV.



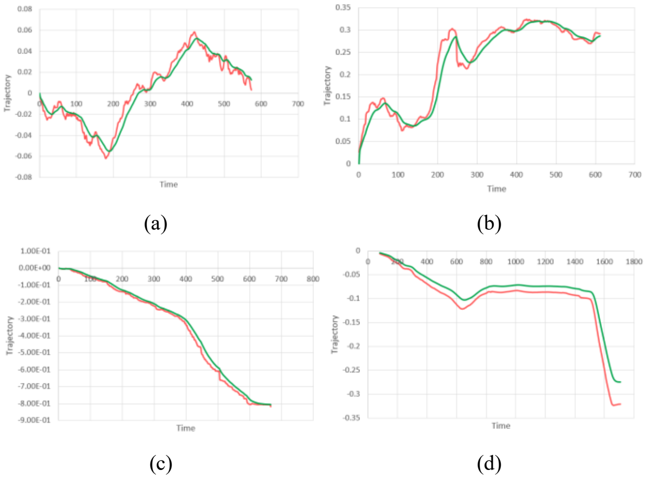
(a)

(b)

(c)

(d)

Fig. 4. Original (top) and stabilized (bottom) images. (a) Park. (b) Street. (c) Car. (d) Online video.



(a)

(b)

(c)

(d)

Fig. 5. Motion estimation (red) and compensation (green). (a) Park. (b) Street. (c) Car. (d) Online video.

IV. Conclusions

A new method for stabilizing the sequence of images captured by a UAV has been presented in this paper. The main contribution of the proposed method is to perform online image stabilization on a UAV with fast computation time and find accurate estimation of image motion based on the most frequent motion vector value of two consecutive images. The precise motion estimation of this method can handle the rough transformations of images caused by UAV motion and vibration. Comparing the results on various image sequences, the proposed method could be a potential online application for image stabilization using UAV. We believe that a vision-based system using a UAV can have better results with online image stabilization due to better image quality.

Acknowledgment

The authors would like to thank Professor Wen-June Wang at National Central University, Taiwan, for his advice on making this paper better.

References

[1] Z. Zheng, Y. Ma, H. Zheng, Y. Gu, and M. Lin, “Industrial part localization and grasping using a robotic arm guided by 2D monocular vision,” Ind. Rob., vol. 45, no. 6, pp. 794–804, 2018.

[2] B. N. Vahrenkamp and T. Asfour, “Simultaneous grasp and motion planning: Humanoid robot ARMAR-III,” IEEE Robotics and Automation Magazine, vol. 19, no. 2, pp. 43–57, 2012.

[3] W. Rahmaniar and A. Wicaksono, “Design and implementation of a mobile robot for carbon monoxide monitoring,” J. Robot. Control, vol. 2, no. 1, 2021.

[4] M. Dunbabin and L. Marques, “Robots for environmental monitoring: Significant advancements and applications,” IEEE Robot. Autom. Mag., vol. 19, no. 1, pp. 24–39, 2012.

[5] W. Rahmaniar, W. Wang, and H. Chen, “Real-time detection and recognition of multiple moving objects for aerial surveillance,” Electronics, vol. 8, no. 12, pp. 1373-1390, 2019.

[6] W. C. Hu, C. H. Chen, T. Y. Chen, D. Y. Huang, and Z. C. Wu, “Moving object detection and tracking from video captured by moving camera,” J. Vis. Commun. Image Represent., vol. 30, pp. 164–180, 2015.

[7] Y. Lu, Z. Xue, G.-S. Xia, and L. Zhang, “A survey on vision-based UAV navigation,” Geo-spatial Inf. Sci., vol. 21, no. 1, pp. 21–32, 2018.

[8] S. Minaeian, J. Liu, and Y.-J. Son, “Vision-based target detection and localization via a team of cooperative UAV and UGVs,” IEEE Trans. Syst. Man, Cybern. Syst., vol. 46, no. 7, pp. 1005–1016, Jul. 2016.

[9] D. H. Yeom, “Optical image stabilizer for digital photographing apparatus,” IEEE Trans. Consum. Electron., vol. 55, no. 3, pp. 1028–1031, 2009.

[10] P. Pournazari, R. Nagamune, and M. Chiao, “A concept of a magnetically-actuated optical image stabilizer for mobile applications,” IEEE Trans. Consum. Electron., vol. 60, no. 1, pp. 10–17, 2014.

[11] M. Hashimoto, T. Kuno, and H. Sugiura, “A new image-stabilizing method by transferring electric charges,” in Proc. of Digest of Technical Papers - IEEE International Conference on Consumer Electronics, 2007, pp. 1-2.

[12] C. Song, H. Zhao, W. Jing, and H. Zhu, “Robust video stabilization based on particle filtering with weighted feature points,” IEEE Trans. Consum. Electron., vol. 58, no. 2, pp. 570–577, 2012.

[13] J. H. Woo, J. H. Yoon, Y. J. Hur, N. C. Park, Y. P. Park, and K. S. Park, “Optimal design of a ferromagnetic yoke for reducing crosstalk in optical image stabilization actuators,” IEEE Transactions on Magnetics, vol. 47, no. 10, pp. 4298–4301, 2011.

[14] C. Kim, M. G. Song, Y. Kim, N. C. Park, K. S. Park, and Y. P. Park, “Design of a new triple electro-magnetic optical image stabilization actuator to compensate for hand trembling,” Microsyst. Technol., vol. 18, no. 9–10, pp. 1323–1334, 2012.

[15] T. Kinugasa, N. Yamamoto, H. Komatsu, S. Takase, and T. Imaide, “Electronic image stabilizer for video camera use,” IEEE Trans.

Consum. Electron., vol. 36, no. 3, pp. 520–525, 1990.

[16] Y. Zhang and M. Xie, “Robust digital image stabilization technique for car camera,” Inf. Technol. J., vol. 10, no. 2, pp. 335–347, 2011.

[17] L. Kejriwal and I. Singh, “A hybrid filtering approach of digital video stabilization for UAV using kalman and low pass filter,” Procedia Comput. Sci., vol. 93, no. September, pp. 359–366, 2016.

[18] J. Li, T. Xu, and K. Zhang, “Real-time feature-based video stabilization on FPGA,” IEEE Trans. Circuits Syst. Video Technol., vol. 27, no. 4, pp. 907–919, 2017.

[19] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in Proc. of International Joint Conference on Artificial Intelligence, 1981, pp. 674--679.

[20] A. Lim, B. Ramesh, Y. Yang, C. Xiang, Z. Gao, and F. Lin, “Real-time optical flow-based video stabilization for unmanned aerial vehicles,” J. Real-Time Image Process., pp. 1–11, 2017.

[21] C. Feichtenhofer and A. Pinz, “Good features to track,” in Proc. of IEEE International Conference on Computer Vision, 1994, pp. 246–253.

[22] Y. G. Ryu and M. J. Chung, “Robust online digital image stabilization based on point-feature trajectory without accumulative global motion estimation,” IEEE Signal Process. Lett., vol. 19, no. 4, pp. 223–226, 2012.

[23] C. Wang, J. Kim, K. Byun, J. Ni, and S. Ko, “Robust digital image stabilization using the Kalman filter,” IEEE Trans. Consum. Electron., vol. 55, no. 1, pp. 6–14, 2009.

[24] X. Wang, X. He, Q. Teng, and M. Gao, “Digital image stabilization based on harmony filter,” J. Softw., vol. 9, no. 4, pp. 913–920, 2014.

[25] W. G. Aguilar and C. Angulo, “Real-time model-based video stabilization for microaerial vehicles,” Neural Process. Lett., vol. 43, no. 2, pp. 459–477, 2016.

[26] T. Yabuki and Y. Yamaguchi, “Real-time video stabilization on an FPGA,” in Proc. of 2013 IEEE International Conference on Smart Structures and Systems, 2013, pp. 114–119.

[27] S. Kumar, H. Azartash, M. Biswas, and T. Nguyen, “Real-time affine global motion estimation using phase correlation and its application for digital image stabilization.,” IEEE Trans. image Process., vol. 20, no. 12, pp. 3406–18, 2011.

[28] G. Liu, T. Wang, L. Yu, Y. Li, and J. Gao, “The improved research on K-means clustering algorithm in initial values,” in Proc. of Int. Conf. Mechatron. Sci. Electr. Eng. Comput., 2013, pp. 2124–2127, 2013.

[29] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the number of clusters in a data set via the gap statistic,” J. R. Stat. Soc. Ser. B Stat. Methodol., vol. 63, no. 2, pp. 411–423, 2001.

[30] G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in Proc. of Scandinavian Conference on Image Analysis, 2003, pp. 363–370.