Automatic extraction of moving objects from UAV-borne monocular images using multi-view geometric constraints

M. Kimura*, R. Shibasaki, X. Shao, and M. Nagai
University of Tokyo, Tokyo, Japan

ABSTRACT

This paper proposes a method to detect dynamic objects in the images obtained by a small UAV. Two geometric constraints in multi-view images are used to classify each of the extracted feature-points as static or dynamic. The first constraint is the epipolar constraint which requires static points to lie on the corresponding epipolar lines in the subsequent image. The second constraint, named as flow-vector bound constraint here, restricts the motion of static points along the epipolar lines. In addition, the pose of the UAV-borne camera, which is required when applying these constraints, is estimated by using a vision-based SLAM method, PTAM. The proposed method fully exploits the characteristics of UAV-borne images and achieves satisfactory results. The algorithms were tested with a small quadrotor platform in a real-world scene and successfully detected features extracted from multiple pedestrians.

1 INTRODUCTION

Detection of dynamic objects from images has been widely studied in computer vision research for many applications, such as traffic supervision, robot navigation, and crowd surveillance. This paper primarily focuses on moving object detection from the images obtained by small unmanned aerial vehicles (UAVs). It is not easy to detect dynamic objects from moving cameras since there are two motions involved: the motion of moving objects and the motion of the camera itself. Dynamic object detection from small UAVs is especially challenging because of the characteristics of these vehicles, such as continuous unrestricted pose variation and bad vibrations. To address these characteristics, new approaches are needed.

Jung *et al.* applied a probabilistic approach to detect moving objects from a mobile robot using a single camera in outdoor environments[1]. The changes in the images caused by camera motion is compensated using corresponding feature sets and outlier detection, and the positions of moving objects are estimated using an adaptive particle filter and EM algorithm. Their algorithms were also tested with unmanned helicopter. Rodriguez *et al.* developed a real-time method to detect and track moving objects from UAVs using a single camera[2]. The main concept proposed in their work is to create an artificial optical flow field using the camera motion between two subsequent images. They compare this artificial flow with the real optical flow directly calculated from the images to detect features that belong to dynamic objects. Siam *et al.* proposed a automatic multiple moving target detection and tracking framework that executes in real-time and is suitable for UAV imagery[3]. Their framework is based on image feature processing and projective geometry, homography. The outlier image features, which violate homography, are computed with least meadian square estimation and clustered spatially as dynamic objects. These dynamic objects are tracked using Kalman filtering while persistency check is carried out to remove false detections.

These earlier studies[1, 2, 3] for moving object detection from moving platforms including UAVs focused on how to discriminate the changes in image sequences caused by dynamic objects from the ones caused by the camera motion. In other words, the motion of the platform is considered as a disadvantage for moving object detection in these approaches and their performances are thought to be best when the platform is not moving. In contrast, we propose an approach for moving object detection utilizing the motion of UAVs. As mentioned below, our approach uses multi-view geometric so that it can detect moving objects from UAV-borne images in real-time. There are some preceding studies using multi-view geometric constraints for moving object detection from moving platforms. Some of these studies are introduced below and we clarify the contribution and novelty of our approach.

Takeda *et al.* proposed a method to detect moving obstacles using the residual error calculated in the process of FOE (Focus Of Expansion) estimation[4]. At first in their method, the dense optical-flow field is extracted from sequence of dynamic images captured by a camera fixed on a moving platform. Next, the FOE is estimated in local image regions. An image region corresponding to the block is added with the residual error. This process is repeated by sliding and adding for the local region while changing the size of the local region. Finally, regions which have high residual error values are detected as candidate regions of moving obstacles. Experiments using ground-vehicles show that the method works

*Email address: motoki@iis.u-tokyo.ac.jp

well in a real outdoor scene. However, this method assumes the pure translation as the platform motion and the rotational motion is not assumed. Since small UAVs has unrestricted pose variation, thier method is not appropriate for UAV-borne images. Kang *et al.* developed an approach to detect and track independently moving regions in a 3D scene captured by a moving camera in the presence of the strong parallax[5]. Each of detected moving pixels are classified into independently moving regions or parallax regions by analyzing two geometric constraints, the commonly used epipolar constraint and the structure consistency constraint. Experiment results using airborne images show that their approach can successfully detect and track independently moving objects in a 3D scene despite of the strong parallax in the images. However, their approach is complex and unsuitable for the real-time process. Kundu *et al.* proposed a similar method to ours to detect moving object from image-sequences obtained by a robot on the ground[6]. Their approach uses two geometric constraints but it does not assume the rotational motion in one of the constraints. Besides, it requires other types of sensors, such as wheel encoders, to estimate the camera motion.

In this paper, we present an automatic method using multi-view geometric constraints to detect moving objects in UAV-borne images. This method fully exploits the unrestricted and continuous pose variation of UAVs and is appropriate for the small UAVs whose motion is unstable. The method can detect moving objects in real-time using an ordinal laptop computer and does not need sensors other than a monocular camera.

2 METHODOLOGY OVERVIEW

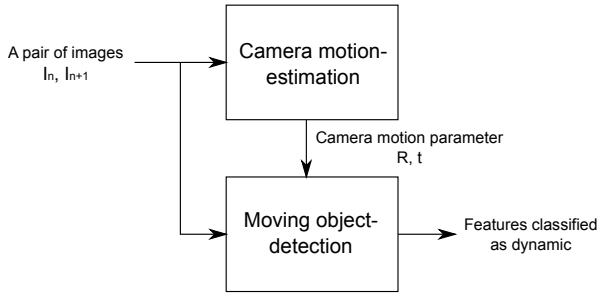

Figure 1: The overview of the proposed method

An approach proposed in this paper consists of two major parts shown in Figure 1. The first part is the one to estimate the camera motion equipped with the UAV. Using two images and a vision-based SLAM method, this part estimates the relative motion of the camera in a 3D scene between the two frames. This relative motion is the one so called camera extrinsic parameter, which includes the translation vector t and the rotational matrix R. These parameters are used in the second part, which is the one for moving object detection. It uses geometric constraints in image sequences calculated from these parameters to detect feature-points which belong to independently moving objects in a 3D scene. The camera motion estimation part is detailed in section 3 and the moving object detection part is in section 4.

3 CAMERA MOTION ESTIMATION

In our approach, the camera extrinsic parameter between a pair of frames is estimated by a vision-based SLAM, PTAM (Parallel Tracking And Mapping)[7]. PTAM is a robust and real-time key-frame based SLAM mehod and have been applied to some vision based navigations for small UAVs[8, 9].

Using the image I_n captured at time-index n, PTAM calculates the camera pose z_n in a reference frame at n:

$$z_n = \begin{bmatrix} r_n \\ q_n \end{bmatrix}, \tag{1}$$

where r_n is the position and q_n is the the quaternion which describes the attitude of the camera. The camera extrinsic parameters or the relative motion between two captured images I_n and I_{n+1} can be calculated as being

$$t_{n:n+1} = -R\left(q_{n+1}\right) \times \left(r_{n+1} - r_n\right), \tag{2}$$

$$R_{n:n+1} = R\left(q_n^* \otimes q_{n+1}\right), \tag{3}$$

where $R\left(q\right)$ is a directional cosine matrix (DCM) defined by the quaternion q and q^* is the conjugation of the quaternion q. The symbol $\otimes$ represents quaternion products.

In the motion detection part, the calculated extrinsic parameters $t_{n:n+1}$ and $R_{n:n+1}$ are used to detect the moving objects in the images I_n and I_{n+1}.

4 MOVING OBJECT DETECTION

Figure 2 represents the process of the moving object detection part in our approach. An upper half of Figure 2 is the process of the moving object detection and lower half are the images which are the results of each step in the process. The details of each step in the process are explaied below in this section.

4.1 Feature extraction and tracking

In the feature extraction step, sparse Kanade-Lucas-Tomasi (KLT) features[10] are extracted from images I_n and I_{n+1} captured at time indexes n and $n+1$. Next, in the feature tracking step, each of features extracted from the images are tracked by KLT feature tracker. Let p_n^i and p_{n+1}^i be the positions of the ith identical 3D-scene point X^i in images I_n and I_{n+1}, which are obtained by feature extraction and tracking steps. KLT features extracted from the image I_n are represented as red points in the left image in Figure 2 and tracking result of each feature between the images I_n and I_{n+1} is represented as green lines in the second image from the left in Figure 2.

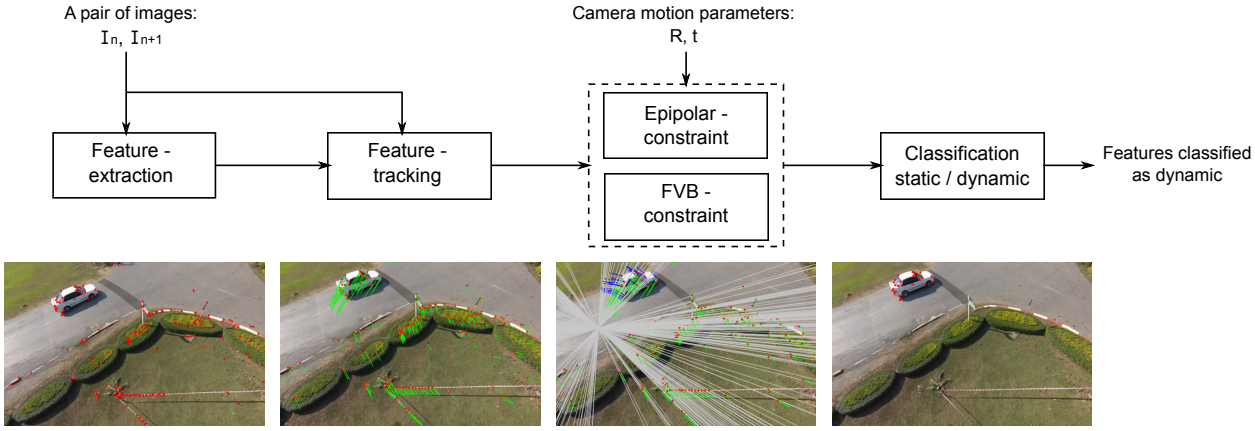

Figure 2: The process of moving object detection and result images of each step

Then, two geometric constraints are evaluated at each of features to classify each of them as static or dynamic. The first constraint we use is the epipolar constraint and the second constraint is the one called flow-vector bound (FVB) constraint in this paper. To calculate each constraint and detect dynamic feature points, the fundamental matrix between the images I_n and I_{n+1} is used. The fundamental matrix $F_{n,n+1}$ between the pair of images are defined as

$$F_{n,n+1} = K^{-T}[t_{n:n+1}]_{\times} R_{n:n+1} K^{-1}, \tag{4}$$

where K is the intrinsic matrix of the camera and $R_{n:n+1}$, $t_{n:n+1}$ is the rotation and translation of the camera between two views, which are given by the camera motion estimation part. The details of the constraints we use and how to evaluate them at each of features are explained in the rest of this section.

4.2 *Epilolar constraint*

The epipolar constraint is represented by $p_{n+1}^{iT} l_{n+1} = 0$, where l_{n+1}^i is the epipolar line in the image I_{n+1} corresponding to the feature p_n^i. The epipolar line l_{n+1}^i is given by: $l_{n+1}^i = F_{n,n+1} p_n^i$. This equation means that features which extracted from static point in a 3D scene to lie on the corresponding epipolar lines in the subsequent image. However, if a point is not static in a 3D scene, the feature p_{n+1}^i may be off the corresponding epipolar line l_{n+1}^i and the perpendicular distance from the feature to the epipolar line, $h_{epi,n+1}^i$ is not zero as shown in the left figure of Figure 3.

If the coefficients of the line l_{n+1}^i are normalized, the perpendicular distance in the image I_{n+1} is given by $h_{epi,n+1} = |l_{n+1}^i \cdot p_{n+1}^i|$. Similarly, the perpendicular distance in the image I_n is given by $h_{epi,n} = |l_n^i \cdot p_n^i|$. If the value of $h_{epi,n}$ or $h_{epi,n+1}$ is far from zero, it is more likely to be an image of the moving point.

The evaluation step of the epipolar constraint is shown in the second image from the right in Figure 2. The white lines represent the epipolar lines and the blue lines represent the distance from features to the corresponding epipolar lines.

4.3 *Flow-vector bound (FVB) constraint*

When the camera does not move, the epipolar line cannnot be defined. Besides, when the degenerate motion arises, moving points cannot be detected with the epipolar constraint since the features move along the epipolar lines even though they belong to dynamic points in a 3D scene as shown in the right figure of Figure 3.

We use the flow-vector bound constraint as the second constraint to detect moving points correctly during degenerate motions. Assuming the pin-hole camera model, we get the equation which describes the feature movement in the images:

$$p_{n+1}^i - K R_{n:n+1} K^{-1} p_n^i = \frac{1}{z} K t_{n:n+1}, \tag{5}$$

where z is the depth of a static 3D point corresponding to the features p_n^i and p_{n+1}^i. If we set z_{max} and z_{min} as the upper and lower bound on z, we then find image displacement bounds along the epipolar line, d_{min} and d_{max}, corresponding to z_{max} and z_{min} using Equation 5. If the image displacement $d^i = |p_{n+1}^i - K R_{n:n+1} K^{-1} p_n^i|$ does not lie between d_{min} and d_{max}, it is more likely to be a dynamic point.

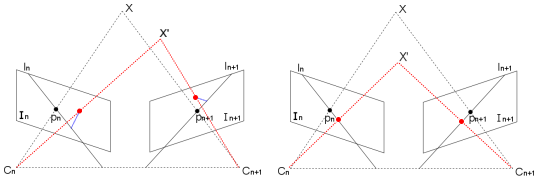

Figure 3: LEFT: a point X in a 3D scene moves non-degenerately hence its image point p does not lie on the corresponding epipolar line. RIGHT: The point X moves degenerately in the epipolar plane. Hence, despite moving, its image point p lies on the corresponding epipolar line.

4.4 *Probabilistic model for the classification*

As mentioned, we denote by p_n^i the ith feature p^i in the image I_n. The corresponding feature in I_{n+1} is denoted by p_{n+1}^i. The probability of p^i being stationary is defined as

$$P(p^i = static) = f_{EP} \times f_{FV}, \tag{6}$$

where f_{EP} and f_{FV} are defined as

$$f_{EP} = e^{-\alpha(|p_n^i \cdot l_n^i| + |p_{n+1}^i \cdot l_{n+1}^i|)}, \tag{7}$$

$$f_{FV} = \left\{ 1 + \left(\frac{d^i - d_{mean}}{d_{range}} \right)^{\beta} \right\}^{-1}, \tag{8}$$

where $d_{mean} = \frac{d_{max}+d_{min}}{2}$, $d_{range} = \frac{d_{max}-d_{min}}{2}$.
α and β are smoothing factors. If the probability is below the threshold, the feature p^i is classified as a dynamic point.

The values of these parameters and threshold need to be adjusted because the optimal values depend on the situation, such as the flight height, the velocity of the moving target, the image resolution, etc. We adjusted these values using particular image sequences captured from the UAV before the experiments.

The features classified as dynamic are shown as red points in the right image in Figure 2. The features which belong to the moving car are correctly detected.

5 Experimental result

The algorithms we propose were tested in a real-world scene with a quadroter-type UAV, AR.Drone2.0, shown in Figure 4.

Figure 4: The UAV used for the experiment (AR.Drone 2.0)

The UAV flew at the height of 10 meters over a crossing where some pedestrians walked. The results of the experiment were shown in Figure 5.

Exracted KLT features in the first image are shown in the topmost image in Figure 5. At this stage, features are extracted from both dynamic objects (pedestrians) and static objects. In the second image from the top in Figure 5, red points represent the features in the second image and green lines show the result of KLT tracking between the first image and the second image. The features extracted from both dynamic and static objects move in the image. In the third image from the top in Figure 5, each white line is the epipolar line of the corresponding feature and each blue line represents the perpendicular line from feature to the corresponding epipolar line. Note that features extracted from pedestrians at right part of the image move vertically to the epipolar line, but features extracted from pedestrians at left part of the image move along the epipolar line. In the bottommost image in Figure 5, red points represent the points which classified as dynamic by our method. Only the features which belong to pedestrians are detected. Although some features extracted from pedestrians move along the epipolar line as can be seen in the third image from the top in Figure 5, those features are also detected by flow-vector bound constraint.

The proposed method was implemented using OpenCV library and could be run at maximally 15 fps on ordinal laptop computer (Intel Core i5-2540M, 2.6GHz, 4GB RAM). Computational resources are mainly consumed for the camera localization (PTAM).

6 Conclusion

We proposed a real-time method to detect the moving points from UAV-borne images using multi-view geometric constraints. The propose method makes the best use of the characteristics of small UAVs, such as their great mobility and pose variation. The algorithm was tested in a real-world scene, a crossing where some pedestrians walk, and the points which belong to dynamic objects were succesfully detected.

As future works, we will develop a clustering method for grouping a set of moving points to a moving object and a robust tracking method to know the behavior of each moving object. We will also challenge the sensor fusion to estimate the camera motion. It improves the speed of our algorithm since vision-based localization is computationally expensive, and it enables us to estimate the motion of the camera robustly even in dynamic environments and texture-poor environments where the performance of PTAM is poor.

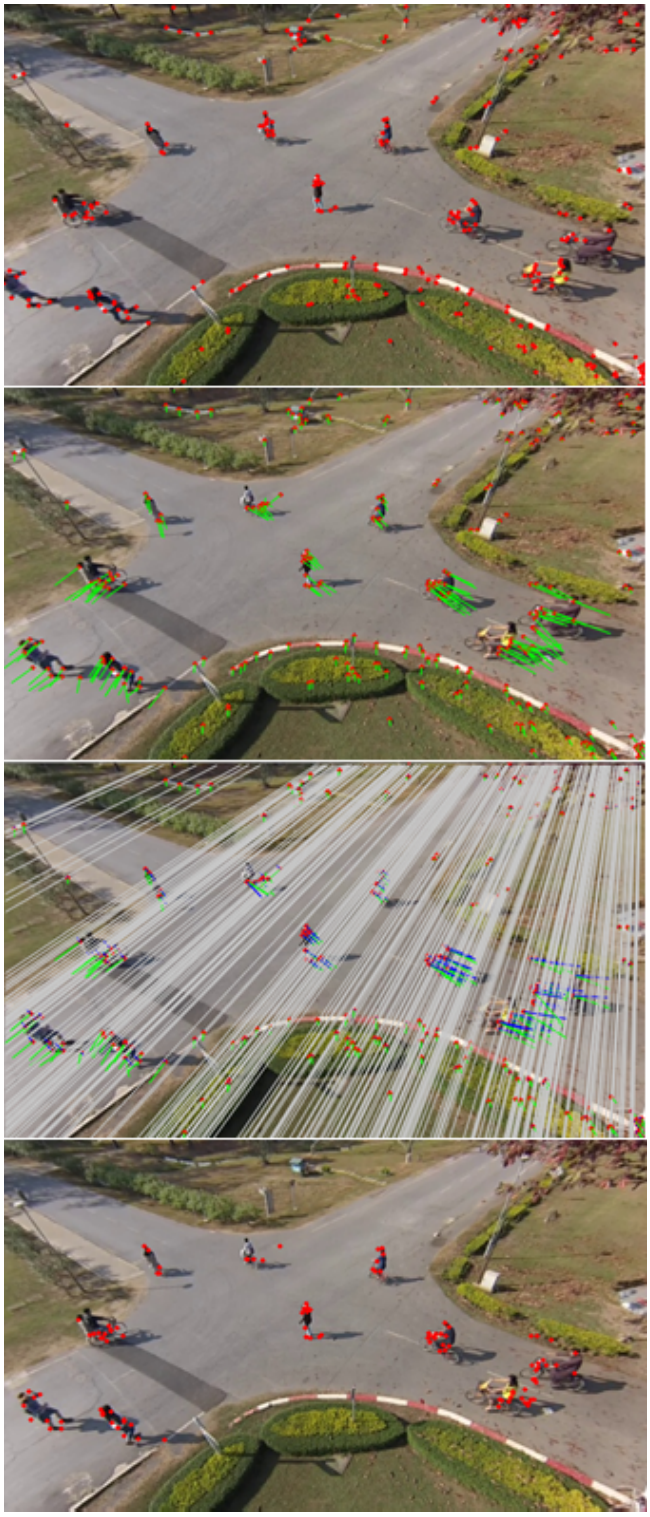

Figure 5: Topmost: Extracted features in the first image (red), The second from the top: Extracted features in the second image (red) and the result of feature tracking (green), The third from the top: Epipolar lines (white) and perpendicular lines from features to the corresponding epipolar lines (blue), Bottommost: Detected moving points (red)

REFERENCES

[1] B. Jung and G. S. Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *International Conference on Intelligent Autonomous Systems*, pp. 980–987, 2004.

[2] G. R. Rodriguez-Canosa, S. Thomas, J. del Cerro, A. Barrientos, and B. MacDonald. A real-time method to detect and track moving objects (DATMO) from unmanned aerial vehicles (UAVs) using a single camera. *Journal of Remote Sensing*, Vol. 4, No. 4, pp. 1090–1111, 2012.

[3] M. Siam and M.ElHelw. Robust autonomous visual detection and tracking of moving targets in UAV imagery. *11th IEEE International Conference on Signal Processing (ICSP)*, Vol. 2, pp. 1060–1066, 2012.

[4] N. Takeda, M. Watanabe, and K. Onoguchi. Moving obstacle detection using residual error of FOE estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vol. 3, pp. 1642–1647, 1996.

[5] J. Kang, I. Cohen, G. Medioni, and C. Yuan. Detection and tracking of moving objects from a moving platform in presence of strong parallax. In *10th IEEE International Conference on Computer Vision (ICCV)*, Vol. 1, pp. 10–17, 2005.

[6] A. Kundu, K. Krishna, and J. Sivaswamy. Moving object detection by multi-view geometric techniques from a single camera mounted robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4306–4312, 2009.

[7] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, 2007.

[8] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time Onboard Visual-Inertial State Estimation and Self-Calibration of MAVs in Unknown Environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 957-964, 2012.

[9] J. Engel, J. Sturm, and D. Cremers. Camera-based navigation of a low-cost quadrocopter, In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2815-2821, 2012.

[10] D. B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *the 7th International Joint Conference on Articial Intelligence*, Vol. 81, pp. 674–697, 1981.