Feature-Based SLAM Algorithm for Small Scale UAV with Nadir View

Danilo Avola[1], Luigi Cinque[1], Alessio Fagioli[1], Gian Luca Foresti[2], Cristiano Massaroni[1], and Daniele Pannone[1](✉)

[1] Department of Computer Science, Sapienza University,
Via Salaria 113, 00198 Rome, Italy
{avola,cinque,fagioli,massaroni,pannone}@di.uniroma1.it

[2] Department of Mathematics, Computer Science and Physics, University of Udine,
Via Delle Scienze 206, 33100 Udine, Italy
gianluca.foresti@uniud.it

Abstract. Small-scale Unmanned Aerial Vehicles (UAVs) have recently been used in several application areas, including search and rescue operations, precision agriculture, and environmental monitoring. Telemetry data, acquired by GPSs, plays a key role in supporting activities in areas like those just reported. In particular, this data is often used for the real-time computation of UAVs paths and heights, which are basic prerequisites for many tasks. In some cases, however, the GPS sensors can lose their satellite connection, thus making the telemetry data acquisition impossible. This paper presents a feature-based Simultaneous Localisation and Mapping (SLAM) algorithm for small-scale UAVs with nadir view. The proposed algorithm allows to know the travelled route as well as the flight height by using both a calibration step and visual features extracted from the acquired images. Due to the novelty of the proposed algorithm no comparisons with other methods are reported. Anyway, extensive experiments on the recently released UAV Mosaicking and Change Detection (UMCD) dataset have shown the effectiveness and robustness of the proposed algorithm. The latter and the dataset can be used as baseline for future research in this application area.

Keywords: UAVs · SLAM algorithm · Mosaicking · A-KAZE

1 Introduction

Nowadays, robots and small-scale UAVs are used in several fields, such as SAR [1,2], environment monitoring and inspection [3–6] and precision agriculture [7,8] due to their low cost and easiness of deployment. These devices are equipped with different sensors, such as gyroscope, accelerometer, compass, and GPS, thus allowing to know the state of the UAV (e.g., breakdown occurrences, travel speed, etc.) with a very high precision. During the execution of a task, it may happen that the UAV loses the connection with the satellites, thus making it

E. Ricci et al. (Eds.): ICIAP 2019, LNCS 11752, pp. 457–467, 2019.
https://doi.org/10.1007/978-3-030-30645-8_42

impossible to retrieve data such as the flight height, or to use the GPS coordinates to know the overflown route. In robotics and computer vision, the most common approaches to determine both position and orientation of the robot are Simultaneous Localization and Mapping (SLAM) [9] and Visual Odometry (VO) [10–12]. The simplest sensor that can be used in performing SLAM and VO is the RGB camera. There are two main categories of works using a RGB camera: methods using monoscopic cameras [13–18] or methods based on stereo camera [19–21]. The major difference between stereo and monoscopic cameras is that the former allows to feel the distance from objects within the scene, as if it had a third dimension. Human vision is stereoscopic by nature due to the binocular view and to our brain, which is capable of synthesizing an image with stereoscopic depth. It is important to notice that a stereoscopic camera must have at least two sensors to produce a stereoscopic image or video, while the monoscopic camera setup is typically composed by a single camera with a 360° mirror. Whenever the scene-to-stereo camera distance is much larger than the stereo baseline, stereo VO can be degraded to the monocular case, and stereo vision becomes ineffectual [22]. Authors in [11] present the first real-time large-scale VO with a monocular camera based on a feature tracking approach and random sample consensus (RANSAC) for outlier rejection. The new upcoming camera pose is computed through 3D to 2D camera-pose estimation. The work in [16] leverages a monocular VO algorithm for feature points tracking on the world ground plane surrounding the vehicle, rather than a traditional tracking approach applied on the perspective camera image coordinates. Two real-time methods for simultaneous localization and mapping with a freely-moving monocular camera, are proposed by the LSD-SLAM [17] and ORB-SLAM [18] algorithms. In [13], the FAST corners and optical flow are used to perform a motion estimation task and, subsequently, a mapping thread is executed through a depth filter formalized as a Bayesian estimation problem. Other works, such as [14,15], propose a robust framework which makes direct use of the pixel intensity, without exploiting a feature extraction step.

In this paper, a feature-based SLAM algorithm for small-scale UAVs with a nadir view is proposed. In detail, a first calibration step is performed to know the ratio between pixels/meters and flight height. Then, during flights, keypoints extracted from the video stream are exploited to know if the flight height changed, while the center of mass of the frames are used for route estimation. Exhaustive experiments performed on the recently released UMCD dataset highlight the robustness and the reliability of the proposed approach. To the best of our knowledge, there are no works in literature that estimates both the trajectory and the flight height of a UAV. Hence, no comparisons with other SLAM algorithm are provided, and the obtained results are meant to be considered as a baseline for future works.

The remainder of the paper is structured as follows. In Sect. 2, the proposed method is described in detail. In Sect. 3, the performed experiments and the obtained results are discussed. Finally, Sect. 4 concludes the paper.

2 Proposed Method

The proposed method main idea is to exploit keypoint matching between two consecutive video frames, received from the UAV, in order to determine the flight height, while the frames center of mass is used to determine the overflown route. A necessary condition for the algorithm correctness, is that features must always be matched between frames, otherwise it is not possible to estimate the correct flight height.

2.1 System Calibration

In order to find the relation between the spatial resolution of the RGB sensor and the flight height, a calibration step is required. To perform this calibration, a marker of known dimensions (e.g., 1×1 m) is placed on the ground and it is then acquired at a known height (e.g., 10 m), through the UAV sensors. During this process, the GPS sensor of the UAV is used to know the exact height. Markers have been chosen for this step due to their robustness and easiness of recognition within the observed environment [23,24]. With this procedure, it is possible to compute the pixels/meters ratio needed to initialize the system. In Fig. 1, the marker detected during the calibration step is depicted.

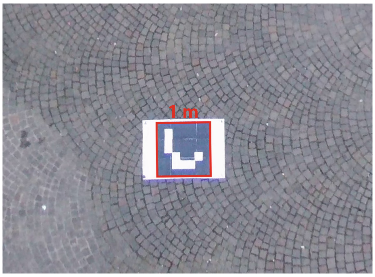


Fig. 1. System calibration step example. By knowing both the UAV height and the marker size, it is possible to estimate the pixels/meters ratio which is a requirement for the algorithm.

In case information about the camera focal length f is not available, this calibration step also allows its estimation. Let us consider the height h of the UAV during the calibration step, the marker to have a real size of w meters and a pixel size at height h of p pixels. Then, f can be computed as follows:

$$f = \frac{(h \times p)}{w} \tag{1}$$

By knowing how to compute the focal length, the height estimation step can be performed with any kind of sensor, as shown in the next Section.

2.2 Flight Height Estimation

For the flight height estimation, keypoints extracted from the video stream frames are used. The A-KAZE [25,26] feature extractor is adopted due to its performance, allowing for a faster feature extraction with respect to SIFT, SURF, and ORB [27]. In more details, features are extracted and matched between two consecutive frames f_{t-1} and f_t, creating two sets of keypoints K_{t-1}, K_t and a set of matches Θ_t. Subsequently, the affine transformation matrix is computed from these matches, and is used to determine the scale changes between keypoints. More thoroughly, if we have an incremental scale change it means that a zoom in operation is performed, so the UAV is lowering the flight height. On the contrary, if we have a decremental scale change it means that a zoom out operation is performed, so the UAV is increasing the flight height. In the flight height estimation, two goals are pursued:

- To filter the identified matches and exclude keypoints belonging to the foreground component (i.e., dynamic elements within the scene) during the drone movement estimation, in order to avoid moving objects negatively influencing the height estimation;
- To estimate all altitude variations.

The first goal is obtained using the set of matches Θ_t, and the homography matrix H_t that maps the coordinates of a keypoint $k \in K_{t-1}$ into the coordinates of a keypoint $\hat{k} \in K_t$. The matrix H_t is computed applying the RANdom SAmple Consensus (RANSAC) algorithm [28] on the matches contained in Θ_t. The reprojection error in H_t can be minimized through the use of the Levenberg-Marquardt optimization [29]. To find the keypoints belonging to the moving objects present in the scene, the following check is performed for each match $(k, \hat{k}) \in \Theta_t$:

$$\gamma = \begin{cases} 1 \text{ if } \sqrt{(k-\hat{k})^2} - \sqrt{(k-(Hk))^2} \geq \rho \\ 0 \text{ otherwise.} \end{cases} \tag{2}$$

where ρ is a tolerance applied on the difference between the estimated distance obtained by homography and the estimated distance obtained by Θ_t. If ρ has a low value, then a large number of keypoints found in the background result as static and, consequently, many false positives can occur for the background keypoint estimation. Instead, if ρ has a high value, the estimation of the keypoint movements is less restrictive, but a large number of false negatives can occur. According to [30], the value of ρ has been fixed to 2.0. In more details, if $\gamma = 0$, then the keypoint $\hat{k}$ is a background keypoint, otherwise $\hat{k}$ is a foreground keypoint. Finally, all background keypoint matches are used to compose a new filtered set of matches called $\hat{\Theta}_t$.

In order to achieve the second goal, an affine transformation matrix A is computed. Given three pairs of matches (k_a, k_b), (k_c, k_d), and $(k_e, k_f) \in \hat{\Theta}_t$ with k_a, k_c, $k_e \in K_{t-1}$ and k_b, k_d, $k_f \in K_t$, the A matrix can be calculated as follows:

$$A = \begin{bmatrix} \lambda_x & 0 & \tau_x \\ 0 & \lambda_y & \tau_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x_{k_b} & x_{k_d} & x_{k_f} \\ y_{k_b} & y_{k_d} & y_{k_f} \end{bmatrix} \begin{bmatrix} x_{k_a} & x_{k_c} & x_{k_e} \\ y_{k_a} & y_{k_c} & y_{k_e} \\ 1 & 1 & 1 \end{bmatrix}^{-1} \tag{3}$$

The translations on the x and y axes are indicated by the τ_x and τ_y, respectively. Drone altitude variations are estimated using the λ_x and λ_y, representing the scale variation on the x and y axes. Once λ_x, λ_y values are computed, we can multiply them by the original pixels/meters ratio to determine the UAV flight height variation. Notice that, altitude changes cause zoom-in (or zoom-out) operations in the frames acquired by the drone and, in those cases, we obtain $\lambda_x = \lambda_y$. Also recall that in order to know the altitude variation, there must always be a match between two consecutive frames, so that it is possible to estimate the transformation matrix and the λ_x, λ_y values. Otherwise, it is unfeasible to correctly estimate the variation.

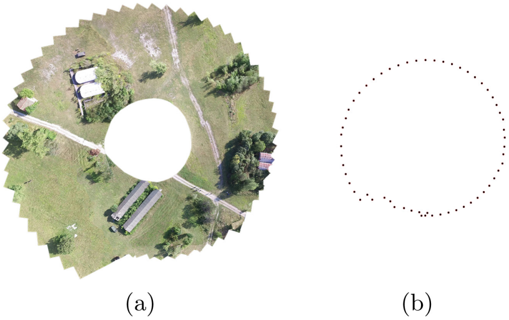


Fig. 2. Example of route estimated with the proposed method. In 2(a), the mosaic of the overflown area is shown, while in 2(b) the route estimates through frames center of mass is depicted.

By using Eq. 1, it is possible to estimate the flight height h' through the triangle similarity:

$$w' = \lambda \times w \tag{4}$$

$$h' = \frac{w' \times f}{p} \tag{5}$$

where λ can be either λ_x or λ_y.

2.3 Route Estimation

Concerning the UAV route estimation, centers of mass from the received video stream frames are used. In order to know where the new center of mass must be positioned with respect to the others, a reference coordinate system must be used. In the proposed method, we use the mosaic of the area overflown by the UAV as reference for the centers of mass. By following the steps shown in [31], a mosaic is built incrementally and in real-time in the following way:

1. Frame Correction: In this step, the radial and tangent distortions are removed (if needed) from the received frame. To perform this step, a matrix containing the calibration values of the camera is required, and it is computed by using well-known methods [32];
2. Feature Extraction and Matching: In this step, keypoints are extracted from the current video frame and the partial mosaic is built up to the previous algorithm iteration. Then, features are matched together and a similarity transformation matrix is generated;
3. Frame Transformation: The similarity transformation matrix generated at the previous step is used to scale, rotate, and translate the received frame in order to align it with the partial mosaic;
4. Stitching: The last step consists in merging together the frame and the partial mosaic seamlessly, using some well-known techniques such as the multiband blending [33].

For each new received frame, the coordinates of all the centers of mass are recomputed. This is due to the fact that when a new frame is added to the partial mosaic, space within the latter must be allocated for the new frame. This operation is performed by appropriately translating the partial mosaic, as well as the centers of mass of the frames composing it, in the new mosaic image. Notice that the centers of mass can be associated with the real GPS coordinates of the UAV frame acquisition. In this way, it is possible to map the estimated route to the real world. In Fig. 2, an example of mosaic and the corresponding estimated route is shown. To summarize, Algorithm 1 shows all the performed steps for both route and flight height estimation.

3 Experiments

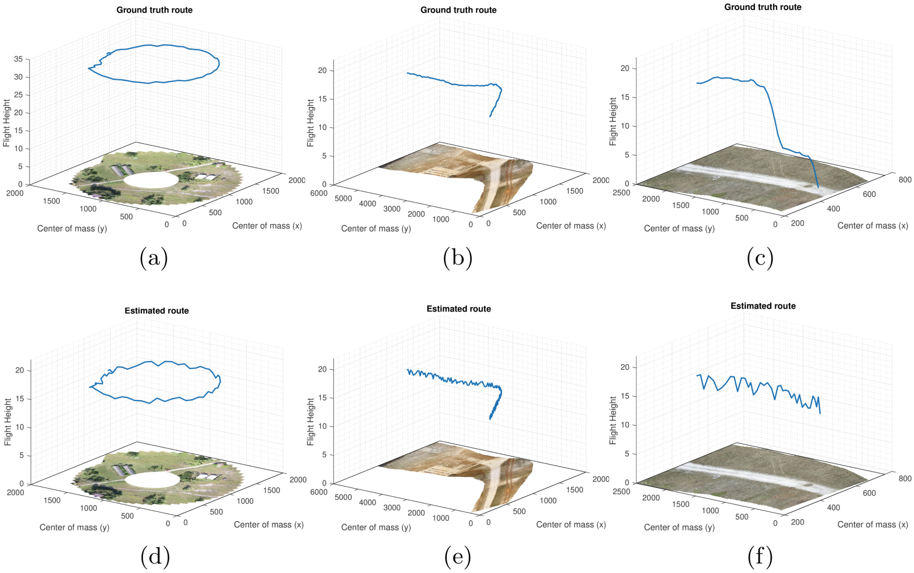
In this section, the results obtained in the performed experiments are reported.

3.1 Dataset

In our experiments, the recently released UMCD dataset [34] is used. The latter provides 50 geo-referenced aerial videos that can be used for mosaicking and change detection tasks at very low altitudes. The authors provide, together with the videos and the GPS coordinates, a basic mosaicking algorithm that has been used in our experiments as ground truth. In addition to the dataset, we have acquired 12 new videos. The latter have been acquired by following the same protocol of the used dataset, in order to have homogeneous testing data. Moreover, the same drone used for building the UMCD dataset, i.e., the DJI Phantom 3, has been used. Since within the dataset there are no videos with a ground marker, the calibration step for those videos has been performed by using

Algorithm 1. Steps performed by the proposed algorithm to estimate UAV route and flight height.

```
Require: Video stream sent by UAV
Ensure: Estimated flight height and route overflown by the UAV
Estimate the pixels/meters ratio through calibration step
while UAV sends video data or it is impossible to match features do
    Extract features from two consecutive frames
    Match the features to create both the $\Theta_t$ set and $A$ matrix
    Compare $A$ matrix with the matrix generated at the previous iteration in order
    to determine the scale changes
    Compute the center of mass for the received frame.
    Use $A$ matrix to align the new received frame with the mosaic generated up to
    the current iteration
    Translate the centers of mass of the frames in the correct position by using the
    $A$ matrix
    Stitch the new frame to the mosaic generated up to the current iteration
end while
```



(a) (b) (c)

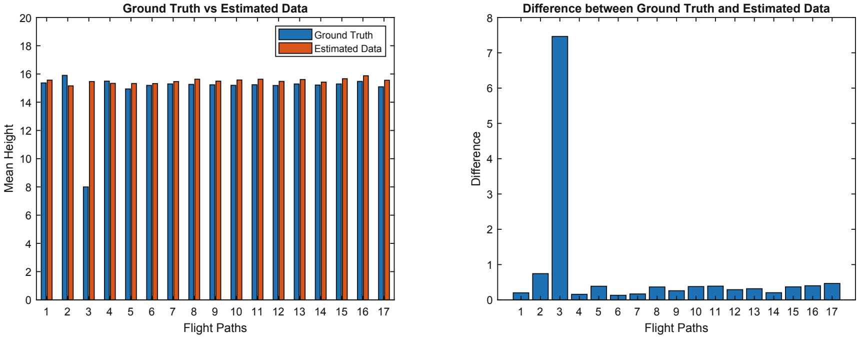
(d) (e) (f)

Fig. 3. Example of paths used for testing the proposed method. In 3(a), (b) and (c), the ground truth data is reported, while in 3(d), (e) and (f) the estimated data is shown. In the x and y axes the centers of mass coordinates are represented, while in the z axis the flight height is shown.

the change detection procedure. This is possible due to the fact that the authors also provide the real size of the objects, in conjunction with the videos. Finally, through the given GPS file, it is possible to know the UAV flight height when in proximity of an object, allowing to compute the pixels/meters ratio needed for the calibration.

3.2 Qualitative Results

For each test, a mosaic of the overflown area has been built to extract the center of mass of each frame, and to estimate the flight route. Since the proposed method relies on the mosaicking algorithm, whenever the mosaic generation failed only a partial route and flight height estimation was given as a result. In Fig. 3, some experimental results are shown. The Figs. 3(a), (b) and (c) show the ground truth for both flight height and route, while the Figs. 3(d), (e) and (f) present the results obtained with the proposed method. In detail, Figs. 3(a) and (d) depict the route of an area of our own acquisitions, while the other figures show two paths provided in the UMCD dataset. As presumed, the results obtained with the proposed algorithm reflect, approximately, the ground truth data. Despite the estimated route and the ground truth route being almost similar, we have more variations on the estimated flight height. This is due to the features matching problem being sensible to outliers, as well as features mismatches. While for Figs. 3(a), (d) and (b), (e) the estimated height and the ground truth height are similar, this is not true for Figs. 3(c) and (f). This is due to the fact that in this specific path the GPS sensor fails in acquiring data, highlighting the potentialities of the proposed algorithm.



(a) (b)

Fig. 4. (a) Comparison between raw data (blue bars), and estimated data (orange bars), and (b) difference between raw and estimated data. (Color figure online)

3.3 Quantitative Results

In Fig. 4, the ground truth and estimated data, together with their difference, is reported. As shown in both Figs. 4(a) and (b), the results obtained with the proposed method are very close to the raw data obtained through the sensors. From Fig. 4(a) it is possible to notice that, in average, the estimated data is slightly overestimated with respect to the ground truth. An exception regards the third flight path, which corresponds to the example shown in Fig. 3(c). In this case, we have a higher distance since the UAV lost the GPS signal during the experiments.

Concerning the execution time, the proposed method strongly depends on the mosaicking algorithm since both keypoints and centers of mass are computed during the process. This means that using new generation hardware and optimizing the algorithm for multicore CPUs or GPUs allows to reach real-time performances.

4 Conclusion

In this paper, a feature-based SLAM algorithm for small-scale UAV with a nadir view is presented. The proposed method exploits a state-of-the-art mosaicking algorithm to estimate the UAV flight route, while image features in conjunction with an affine transformation are used to estimate the flight height. Experimental results are performed on our aerial acquisitions and on the recently released UMCD dataset, showing the effectiveness of the proposed approach.

Acknowledgement. This work was supported in part by the MIUR under grant "Departments of Excellence 2018-2022" of the Department of Computer Science of Sapienza University.

References

1. Qi, J., et al.: Search and rescue rotary-wing uav and its application to the lushan ms 7.0 earthquake. J. Field Robot. **33**(3), 290–321 (2016)
2. Ho, Y.-H., Chen, Y.-R., Chen, L.-J.: Krypto: assisting search and rescue operations using wi-fi signal with UAV. In: Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use, DroNet 2015, pp. 3–8. ACM, New York (2015)
3. Meng, X., Wang, W., Leong, B.: SkyStitch: a cooperative multi-UAV-based real-time video surveillance system with stitching. In: Proceedings of the 23rd ACM International Conference on Multimedia, MM 2015, pp. 261–270. ACM, New York (2015)
4. Avola, D., Foresti, G.L., Martinel, N., Micheloni, C., Pannone, D., Piciarelli, C.: Aerial video surveillance system for small-scale UAV environment monitoring. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6 (2017)
5. Avola, D., Cinque, L., Foresti, G.L., Marini, M.R., Pannone, D.: A rover-based system for searching encrypted targets in unknown environments. In: Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM, INSTICC, pp. 254–261. SciTePress (2018)
6. Piciarelli, C., Avola, D., Pannone, D., Foresti, G.L.: A vision-based system for internal pipeline inspection. IEEE Transact. Industr. Inf. **15**(6), 3289–3299 (2019)
7. Faiçal, B.S., et al.: An adaptive approach for UAV-based pesticide spraying in dynamic environments. Comput. Electron. Agric. **138**, 210–223 (2017)
8. Tokekar, P., Hook, J.V., Mulla, D., Isler, V.: Sensor planning for a symbiotic uav and ugv system for precision agriculture. IEEE Trans. Rob. **32**(6), 1498–1511 (2016)

9. Davison, A.J., Murray, D.W.: Simultaneous localization and map-building using active vision. IEEE Transact. Pattern Anal. Mach. Intell. **24**(7), 865–880 (2002)
10. Aqel, M.O.A., Marhaban, M.H., Saripan, M.I., Ismail, N.B.: Review of visual odometry: types, approaches, challenges, and applications. SpringerPlus **5**(1), 1897 (2016)
11. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2004, CVPR 2004, vol. 1, pp. I-652–I-659, June 2004
12. Shan, M., et al.: A brief survey of visual odometry for micro aerial vehicles. In: 42nd Annual Conference of the IEEE Industrial Electronics Society 2016, IECON 2016, pp. 6049–6054 (2016)
13. Faessler, M., Fontana, F., Forster, C., Mueggler, E., Pizzoli, M., Scaramuzza, D.: Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle. J. Field Robot. **33**(4), 431–450 (2016)
14. Bloesch, M., Omari, S., Hutter, M., Siegwart, R.: Robust visual inertial odometry using a direct EKF-based approach. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 298–304, September 2015
15. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: fast semi-direct monocular visual odometry. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 15–22, May 2014
16. Hamme, D.V., Goeman, W., Veelaert, P., Philips, W.: Robust monocular visual odometry for road vehicles using uncertain perspective projection. EURASIP J. Image Video Process. **1**, 2015 (2015)
17. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_54
18. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-Slam: a versatile and accurate monocular slam system. IEEE Transact. Robot. **31**(5), 1147–1163 (2015)
19. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visual-inertial odometry using nonlinear optimization. Int. J. Robot. Res. **34**(3), 314–334 (2015)
20. Witt, J., Weltin, U.: Robust stereo visual odometry using iterative closest multiple lines. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4164–4171, November 2013
21. Heng, L., Choi, B.: Semi-direct visual odometry for a fisheye-stereo camera. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4077–4084, October 2016
22. Scaramuzza, D., Fraundorfer, F.: Visual odometry [tutorial]. IEEE Robot. Autom. Mag. **18**(4), 80–92 (2011)
23. Avola, D., Cinque, L., Foresti, G.L., Mercuri, C., Pannone, D.: A practical framework for the development of augmented reality applications by using ArUco markers. In: Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods, pp. 645–654 (2016)
24. Bergamasco, F., Albarelli, A., Cosmo, L., Rodolà, E., Torsello, A.: An accurate and robust artificial marker based on cyclic codes. IEEE Transact. Pattern Anal. Mach. Intell. **38**(12), 2359–2373 (2016)
25. Alcantarilla, P.F., Nuevo, J., Bartoli, A.: Fast explicit diffusion for accelerated features in nonlinear scale spaces. In: British Machine Vision Conference (BMVC) (2013)

26. Avola, D., Cinque, L., Foresti, G.L., Martinel, N., Pannone, D., Piciarelli, C.: Low-level feature detectors and descriptors for smart image and video analysis: a comparative study. In: Kwaśnicka, H., Jain, L.C. (eds.) Bridging the Semantic Gap in Image and Video Analysis. ISRL, vol. 145, pp. 7–29. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73891-8_2
27. Avola, D., Cinque, L., Foresti, G.L., Massaroni, C., Pannone, D.: A keypoint-based method for background modeling and foreground detection using a PTZ camera. Pattern Recogn. Lett. **96**, 96–105 (2017)
28. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
29. Marquardt and Donald: An algorithm for least-squares estimation of nonlinear parameters. SIAM J. Appl. Math. **11**(2), 431–441 (1963)
30. Avola, D., Bernardi, M., Cinque, L., Foresti, G.L., Massaroni, C.: Combining keypoint clustering and neural background subtraction for real-time moving object detection by PTZ cameras. In: Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM, INSTICC, pp. 638–645. SciTePress (2018)
31. Avola, D., Foresti, G.L., Martinel, N., Micheloni, C., Pannone, D., Piciarelli, C.: Real-time incremental and geo-referenced mosaicking by small-scale UAVs. In: Battiato, S., Gallo, G., Schettini, R., Stanco, F. (eds.) ICIAP 2017. LNCS, vol. 10484, pp. 694–705. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68560-1_62
32. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transact. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)
33. Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. Int. J. Comput. Vis. **74**(1), 59–73 (2007)
34. Avola, D., Cinque, L., Foresti, G.L., Martinel, N., Pannone, D., Piciarelli, C.: A UAV video dataset for mosaicking and change detection from low-altitude flights. IEEE Transact. Syst. Man Cybern. Syst. **PP**(99), 1–11 (2018)