

Uncalibrated downward-looking UAV visual compass based on clustered point features

Ying LIU¹, Yu ZHANG^{1*}, Ping LI¹ & Bin XU²

¹State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China;

²School of Automation, Northwestern Polytechnical University, Xi'an 710072, China

Received 30 August 2018/Revised 17 October 2018/Accepted 28 December 2018/Published online 24 April 2019

Citation Liu Y, Zhang Y, Li P, et al. Uncalibrated downward-looking UAV visual compass based on clustered point features. *Sci China Inf Sci*, 2019, 62(9): 199202, <https://doi.org/10.1007/s11432-018-9748-1>

Dear editor,

Unmanned aerial vehicles (UAVs) have generated significant research interest in recent years [1–3], in particular to obtain more accurate orientation estimations. A visual compass (VC) does this by using image features only, which avoids magnetic field interferences commonly experienced when using traditional magnetic compasses.

Most VC algorithms employ omnidirectional or panoramic cameras with a wide field of view. For instance, Refs. [4, 5] rely on a calibrated catadioptric camera and traditional image features. Ref. [6] uses pixel intensity over the entire image. The problem with these cameras is that they are not only challenging to install and to use for computation [4] but will fail easily in dynamic situations [7]. Another method uses pinhole cameras to obtain images similar to omnidirectional views but has more requirements on employment. For example, Ref. [8] uses a purely rotating camera and Ref. [9] uses multi-cameras in unknown dynamic environments. However, the horizontal view is of limited use for capturing suitable image features.

We devised a novel VC algorithm that applies image sequences taken by a downward-looking low-cost uncalibrated monocular camera, and a cluster-based method to obtain real point features from oriented FAST and rotated BRIEF (ORB) keypoints. Then the yaw angle of the camera can be computed. Outdoor experiments verified the suitability of the algorithm for UAVs.

* Corresponding author (email: zhangyu80@zju.edu.cn)

Contributions. Relative to the conventional VC algorithms cited, our new method offers the following advantages. (a) In terms of computation and memory use, it has a much more economical approach towards defining point features from matched cluster centers, which not only decreases the required number of keypoints but also reduces the influences of individual points. (b) It enhances robustness towards camera jitter by computing yaw angles using line vectors defined from cluster centers. (c) It employs a new camera installation method that could maximize image information. (d) This uncalibrated algorithm and the convenient plug-and-play sensor assist the optimized operation of the UAV navigation system, particularly in hostile environments.

Statement. A UAV hovering in real situation with lots of jitters and little translations may invalidate the plane hypothesis. And most lines in 3D space transfer to different planes. This is problematic, as a homography matrix can only be used when the plane hypothesis is valid, and essential matrix cannot be used for a purely rotating camera. While the changes in inclination angle of these 3D lines can always fully represent the rotated yaw angle of the camera, which means depending on geometry, our method can eliminate the impacts of translation and installation offsets of the camera.

Real point features. First, we extract a large number m of ORB keypoints, unevenly distributed in the raw image of current frame, to form initial

keypoints set $D_K^c = \{k_1^c, k_2^c, \dots, k_m^c\}$, where the keypoint positions are denoted as $k_i^c = (x_i^c, y_i^c)$, $i = 1, 2, \dots, m$ in pixel coordinate frame and the superscript c represents current frame (k represents keyframe). Then a density-based spatial clustering of applications with noise (DBSCAN) algorithm [10] is handled so that most ORB keypoints are sorted into n clusters. Hence, neglecting sparse noise points, an ORB keypoint cluster can be represented by a single point (center):

$$\begin{aligned} C_1^c &= \{k_{11}^c, k_{12}^c, \dots, k_{1m_1}^c\}, \\ C_2^c &= \{k_{21}^c, k_{22}^c, \dots, k_{2m_2}^c\}, \\ &\vdots \\ C_n^c &= \{k_{n1}^c, k_{n2}^c, \dots, k_{nm_n}^c\}, \end{aligned} \quad (1)$$

where $\{k_{11}^c, k_{12}^c, \dots, k_{nm_n}^c\} \subset D_K^c$, ignoring the noise points in D_K^c .

Then a set of cluster centers can be easily constructed using point positions in each cluster, denoted as $D_C^c = \{c_1^c, c_2^c, \dots, c_n^c\}$, where

$$\begin{aligned} c_1^c &= \frac{k_{11}^c + k_{12}^c + \dots + k_{1m_1}^c}{m_1}, \\ c_2^c &= \frac{k_{21}^c + k_{22}^c + \dots + k_{2m_2}^c}{m_2}, \\ &\vdots \\ c_n^c &= \frac{k_{n1}^c + k_{n2}^c + \dots + k_{nm_n}^c}{m_n}. \end{aligned} \quad (2)$$

Thus, hundreds of keypoints are reduced to tens of more robust cluster centers that are much less sensitive to individual keypoints. Afterwards, ORB descriptors are computed for these cluster centers. Each frame is matched to a keyframe initially by brute force to reduce cumulative errors. If c_i^c matches to c_j^k , the two clusters C_i^c and C_j^k are matched accordingly. The following two criteria are applied to remove erroneous matches in the initial matching results:

$$\begin{aligned} N_{cp}^c - N_{cp}^k &\leq T_{\Delta cs}, \\ d(c_i^c, c_j^k) &= \sqrt{(x_i^c - x_j^k)^2 + (y_i^c - y_j^k)^2} \leq T_{\Delta cd}, \end{aligned} \quad (3)$$

where N_{cp}^c is the size of the matched cluster in current frame (described by the number of cluster points), N_{cp}^k is the size of the matched cluster in keyframe, $T_{\Delta cs}$ is the maximum size difference for each matched cluster, $d(c_i^c, c_j^k)$ is the distance between c_i^c and c_j^k , and $T_{\Delta cd}$ is the maximum distance between matched cluster centers.

Cluster-center pairs satisfying both matching conditions are denoted $D_C^{c*} = \{c_1^{c*}, c_2^{c*}, \dots, c_{N_c}^{c*}\}$

and $D_C^{k*} = \{c_1^{k*}, c_2^{k*}, \dots, c_{N_c}^{k*}\}$, where the superscript $*$ represents a good match between current frame and keyframe (same as the following), and N_c is the number of well-matched points. Improper matches are regarded as outliers and discarded. Finally, D_C^{c*} represents the real point features in current image. The proposed method is outlined in Figure 1 and the algorithm is described in Appendix A.

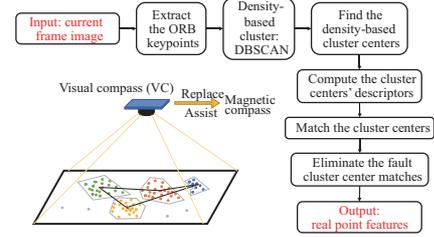


Figure 1 (Color online) Extraction of real point features.

Line vectors constructed from real point features. Writing D_C^{c*} and D_C^{k*} to denote the real point features in current frame and keyframe respectively. Two points in D_C^{c*} (together with the corresponding points in D_C^{k*}) then serve as start and end points to construct two sets of matched line vectors, which are matched accordingly and denoted as D_V^{c*} and D_V^{k*} :

$$\begin{aligned} D_V^{c*} &= \{v_1^{c*}, v_2^{c*}, \dots, v_{N_v}^{c*}\} \\ &= \{(x_{v_1}^{c*}, y_{v_1}^{c*}), (x_{v_2}^{c*}, y_{v_2}^{c*}), \dots, (x_{v_{N_v}}^{c*}, y_{v_{N_v}}^{c*})\}, \\ D_V^{k*} &= \{v_1^{k*}, v_2^{k*}, \dots, v_{N_v}^{k*}\} \\ &= \{(x_{v_1}^{k*}, y_{v_1}^{k*}), (x_{v_2}^{k*}, y_{v_2}^{k*}), \dots, (x_{v_{N_v}}^{k*}, y_{v_{N_v}}^{k*})\}, \end{aligned} \quad N_v = N_c - 1,$$

where N_v is the number of line vectors.

Two sets of inclination angles $D_{\theta_v}^{c*} = \{\theta_{v_1}^{c*}, \theta_{v_2}^{c*}, \dots, \theta_{v_{N_v}}^{c*}\}$ in current frame and $D_{\theta_v}^{k*} = \{\theta_{v_1}^{k*}, \theta_{v_2}^{k*}, \dots, \theta_{v_{N_v}}^{k*}\}$ in keyframe can be obtained from D_V^{c*} and D_V^{k*} :

$$\begin{aligned} \theta_{vi}^{c*} &= \arctan \frac{y_{vi}^{c*} - y_{vj}^{c*}}{x_{vi}^{c*} - x_{vj}^{c*}}, \\ \theta_{vi}^{k*} &= \arctan \frac{y_{vi}^{k*} - y_{vj}^{k*}}{x_{vi}^{k*} - x_{vj}^{k*}}, \end{aligned} \quad (4)$$

$$i = 1, 2, \dots, N_v - 1, j = 2, 3, \dots, N_v, \\ i \neq j, N_v > T_{N_i}.$$

Then a set of yaw angle increments is derived from the differences between the inclination angles of each matched line vector. To enhance robustness and reduce the influences of mismatching, a median filter is used to obtain the yaw angle increment $\Delta\Psi$ of the frame relative to keyframe:

$$\Delta\Psi = \text{median}(D_{\theta_v}^{c*} - D_{\theta_v}^{k*}). \quad (5)$$

Subsequently, the final output $\Delta\Psi_c$ can be retrieved as follows:

$$\Psi_c^1 = \Psi_k^1 + \Delta\Psi, \Psi_c = \Psi_1 + \Psi_c^1, \quad (6)$$

where Ψ_1 is the absolute yaw angle of the first frame provided by an extra sensor, Ψ_k^1 and Ψ_c^1 are the yaw angles of keyframe and current frame relative to Ψ_1 , Ψ_c is the absolute yaw angle of current frame.

Updating keyframes too frequently can significantly increase cumulative errors, especially in dramatically dynamic environments in which mismatching might result in inaccuracies or even failures. Therefore, it is advisable to update keyframe only when either of these conditions is satisfied

$$\begin{cases} N_f \geq T_{kN}, & \text{(i)} \\ N_P \leq T_{pN}, N_l \leq T_{Nl}, & \text{(ii)} \end{cases} \quad (7)$$

where N_f is the number of interval frames between two keyframes, T_{kN} is the maximum number of interval frames between two keyframes, and T_{pN} is the minimum number of point features.

Experiments. A UAV was equipped with a downward-looking monocular camera and a PX4 flight controller (see Appendix B) that provided accurate attitude estimations by fusing three low-cost inertial measurement units (IMUs), global positioning system (GPS), compass, and barometer. Outdoor experiments were conducted to compare the proposed VC algorithm with the fusion algorithm in PX4 and the algorithm using essential matrix. The results of these experiments are shown in Appendix B.

In most situations, the VC algorithm yielded a similar accuracy with the PX4 algorithm even when the visual scene included dynamic objects; however, it did not always work well with the algorithm of essential matrix. Our VC uses only an unaligned, uncalibrated, low-cost monocular camera, while the fusion algorithm in PX4 requires various sensors that mostly need calibration. The use of a GPS also results in a loss of autonomy. Additionally, the essential-matrix-based algorithm requires calibration parameters and cannot be used for a UAV undergoing pure rotation. Despite the continual vibrations of the UAV during the experiments, our VC algorithm responded robustly to interferences.

Conclusion and future work. We presented a novel VC algorithm to estimate the yaw angle of an uncalibrated downward-looking camera mounted on UAVs. Outdoor experiments highlighted some of its attractive features: it is uncalibrated, robust

to noise and illumination changes, resilient to interference from magnetic fields and visual scenes, capable of real-time operation, and versatile. It can assist existing navigation systems. Whereas the essential-matrix-based method is slightly more imprecise and cannot be used when dealing with pure rotations. Future work will focus on developing it to estimate 3D attitude by fusing other sensors.

Remark. In addition to the VC, UAVs require other sensors (e.g., a magnetic compass) to correct inevitable long-term drifts globally.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61673341, 61573091, 61622308, 61873206), National Key R&D Program of China (Grant No. 2016YFD0200701-3), and Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (Grant No. ICT1800421).

Supporting information Appendixes A and B. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

- Wang K L, Ke Y J, Chen B M. Autonomous reconfigurable hybrid tail-sitter UAV U-Lion. *Sci China Inf Sci*, 2017, 60: 033201
- Peng K M, Lin F, Chen B M. Online schedule for autonomy of multiple unmanned aerial vehicles. *Sci China Inf Sci*, 2017, 60: 072203
- Li P, Yu X, Peng X Y, et al. Fault-tolerant cooperative control for multiple UAVs based on sliding mode techniques. *Sci China Inf Sci*, 2017, 60: 070204
- Mariottini G L, Scheggi S, Morbidi F, et al. An accurate and robust visual-compass algorithm for robot-mounted omnidirectional cameras. *Robot Auton Syst*, 2012, 60: 1179–1190
- Bermudez-Cameo J, Puig L, Guerrero J J. Hypercatadioptric line images for 3D orientation and image rectification. *Robot Auton Syst*, 2012, 60: 755–768
- Labrosse F. The visual compass: performance and limitations of an appearance-based method. *J Field Robot*, 2006, 23: 913–941
- Sturm J, Visser A. An appearance-based visual compass for mobile robots. *Robot Auton Syst*, 2009, 57: 536–545
- Montiel J M M, Davison A J. A visual compass based on SLAM. In: *Proceedings of IEEE International Conference on Robotics and Automation*, 2006. 1917–1922
- Sabnis A, Vachhani L. A multiple camera based visual compass for a mobile robot in dynamic environment. In: *Proceedings of IEEE International Conference on Control Applications*, 2013. 611–616
- Ester M. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996. 226–231