

Google Map Aided Visual Navigation for UAVs in GPS-denied Environment

Mo Shan^{1*}, Fei Wang¹, Feng Lin¹, Zhi Gao¹, Ya Z. Tang¹, Ben M. Chen²

Abstract—Since GPS based UAV navigation lacks security and reliability, we propose a framework for UAV navigation in GPS-denied environment, based on Google Map. Geo-referenced navigation provides drift-free localization and does not require loop closures. The UAV position is initialized via correlation, which is simple and efficient. We then use optical flow to predict its position in subsequent frames. During pose tracking, we obtain inter-frame translation either by structure from motion or homography decomposition, and we use HOG features for registration on Google Map. We employ particle filter to conduct a coarse to fine search to localize the UAV. Offline test using aerial images collected by our quadrotor platform shows promising results as our approach eliminates the drift in dead-reckoning, and the small localization error indicates the superiority of our approach as a supplement to GPS.

I. INTRODUCTION

Navigation of unmanned aerial vehicles (UAVs) often relies on Global Positioning System (GPS), which lacks security and reliability [1]. Pose estimation via the fusion of inertial measurement unit (IMU) and optical flow (OF) suffers from the drift. Simultaneous localization and mapping (SLAM) minimizes the drift but requires loop closures. To address these drawbacks, we propose a geo-referenced localization framework, which is reliable, drift-free, and does not require loop closures.

We leverage on image registration to provide an absolute position in Google Map. Although its accessibility is appealing, and some relevant works have been reported, this task is quite demanding. Variation in scale, orientation, and illumination poses a great challenge to register the image captured by the onboard camera to the map. Furthermore, the scene changes between the onboard frame and the map is obvious when using Google Map as reference, which is not updated constantly.

To address these challenges, we use image registration by relying on gradient patterns, which are the most reliable feature for multi-modal matching [2]. Histograms of Oriented Gradients (HOG) [3] has proved to be highly successful for pedestrian detection, which incorporates fine scale gradient orientations and spatial binning. Furthermore, its local normalization mitigates illumination variation. Focusing on gradients makes HOG invariant for different modalities, since the gradient patterns remain similar and the correlation between HOG features is high. To expedite the matching

process, we employ particle filter (PF) to avoid sliding window search. For efficiency, the search is confined around the UAV location predicted by OF.

Since our approach combines HOG, OF and PF, we coin the terms and name it HOP. In short, our contributions are summarized as follows. Firstly, we present a simple yet effective navigation framework, which relies on correlation for initialization, HOG features to describe the images and PF to reduce the amount of comparisons. Secondly, we propose an OF based approach to compute inter-frame motion. To the best of our knowledge, this is the first time that low resolution Google Map and HOG are used for UAV navigation.

The rest of the paper is organized as follows: Section II presents a literature review; the detailed implementation of HOP is illustrated in Section III; Section IV contains experiments and analysis; following it are the conclusion and future research directions.

II. RELATED WORK

Various methods have been developed for UAV navigation to deal with GPS disruption. These works rely on different geographic information, including Geographic Information System (GIS), Google Earth, and Google Street View.

GIS data and its vector layers have been used to estimate UAV position in early works [4], [5], [6] on geo-referencing. In [6], GIS data in the form of Ordnance Survey (OS) layers are divided into overlapping tiles, and inertial navigation system (INS) estimates which tile the UAV is located. The aerial image is rotated, scaled, and classified to form feature codes, which is compared with the OS data. Nevertheless, the training data may be inadequate to reflect the spectral signatures of all classes, especially under the presence of varying light conditions. Moreover, the classification, cross correlation and distance calculation are time consuming.

Images obtained via Google Earth have also been deployed for geo-referencing [7], [8]. In [7] the authors combine visual odometry and image registration to augment the navigation system. The Kalman filter is adopted to fuse the vision system with the INS. To compensate for the drift, an image registration technique realized by edge matching is developed. The registration is robust to change in scale, rotation and illumination to a certain extend. However, during the whole flight there are few successful matches. Therefore this method may not be suitable for long range flights. Another approach [8] that relies on Google Earth images involves classification of the scene. UAV images are segmented into superpixels and then classified as grass, asphalt and house. Circular regions are selected to construct

¹Temasek Laboratories, National University of Singapore, Singapore

²Department of Electrical & Computer Engineering, National University of Singapore, Singapore

*Email: shanmo@nus.edu.sg

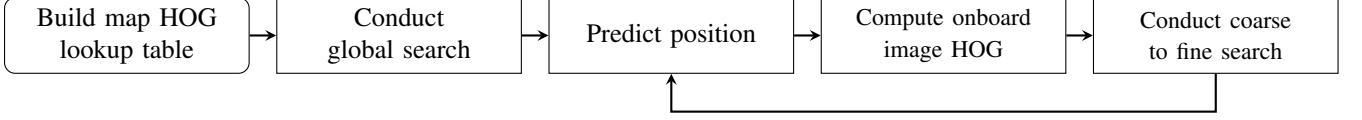


Fig. 1: The HOG features for the map are computed offline. During onboard processing, we use global search to initialize the UAV position. Then for each frame, we track the pose by position prediction and image registration.

the class histograms, which are rotation invariant. However, discarding rotation gives rise to the classification uncertainty. Consequently sometimes the drift in position estimation is not successfully removed. Moreover, the matching accuracy is poor in large homogeneous regions. In addition, the training requires labeling the reference map manually.

Recently, several approaches [9], [10] using Google Street View images for UAV localization have emerged. In [9], the aerial images are searched in the Google Street View database. To tackle with large viewpoint change, artificial views of the aerial images are generated and then Scale Invariant Feature Transform (SIFT) features are extracted. These features are compared against those extracted from ground images to find nearest neighbors. The outliers are removed via a histogram voting scheme and the good matches are verified by the Virtual Line Descriptor. Nevertheless, SIFT requires intensive computation and this approach does not use motion dynamics.

To summarize, previous works either report low accuracy or require additional information provided by GIS and classification. In other words, the matching accuracy depends on the complexity of the required input. Furthermore, these methods do not make full use of the motion dynamics retrieved from onboard sensors. We propose a method that achieves relatively high accuracy, while only relying on simple input as reference. Our proposed approach mainly differs from the aforementioned ones in the following ways:

- 1) The easily accessible Google Map provides the geometric information for navigation, requiring less memory consumption compared with GIS and Google Street View.
- 2) The onboard sensors are utilized to obtain the rotation and scale of the frames, as well as the inter-frame translation.
- 3) It matches multi-modal images without feature detection and description. Instead, HOG is used holistically for image description and PF is employed to avoid sliding window search.

III. GEO-REFERENCED NAVIGATION

In this section the visual navigation framework will be investigated. An flowchart of HOP is given in Fig. 1, which provides an overview. A lookup table is built offline, and then correlation based global search is performed for initialization. Position prediction is performed next to restrain the search. After the onboard image is described by HOG, the particles are drawn in a coarse grid. If the match is reliable, then the particles will vote for the UAV location. Otherwise, fine search is conducted. After the search, it predicts the

position for the next frame. If the search fails, the OF position is retained as UAV location unless the motion is too large, in which case the UAV will re-initialize itself.

A. Global localization

After taking off, the UAV location is searched in the entire map for initialization. To avoid sliding window search, which is quite time consuming, we adopt the correlation filter with a high frame rate proposed in [11]. According to the Convolution Theorem, correlation is equivalent to multiplication in Fourier domain. Thus, we exploit the Fast Fourier Transform and expedite the correlation as in Eq. 1, where F is the 2D Fourier transform of the input image, H is the transform of the filter, \odot denotes element wise multiplication and $*$ indicates complex conjugate. Because no training is available for detection, we correlate the current frame and the map. As a result, transforming G into the spatial domain gives a confidence map of the location.

$$G = F \odot H^* \quad (1)$$

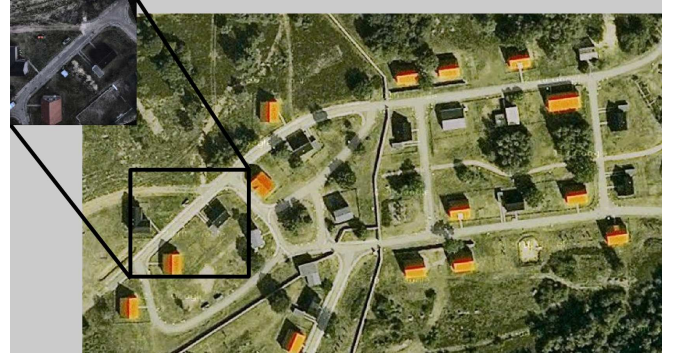


Fig. 2: Onboard image at taking off position, and its corresponding rectangular region in the map.

Take the onboard image displayed in Fig. 2 for example, it corresponds to the rectangular region on the map. Its confidence map is shown in Fig. 3, from which it is evident that the correct location of the onboard image possesses the highest confidence. In addition, the processing time of correlation is merely 0.036 s.

Apart from initialization, global search could also be conducted for lost recovery. When the match is unreliable and OF prediction is used, the onboard image may not be in the search region due to large motion. When this happens the UAV is often lost. Hence UAV should ascend to observe a broader region and search globally to re-initialize itself.

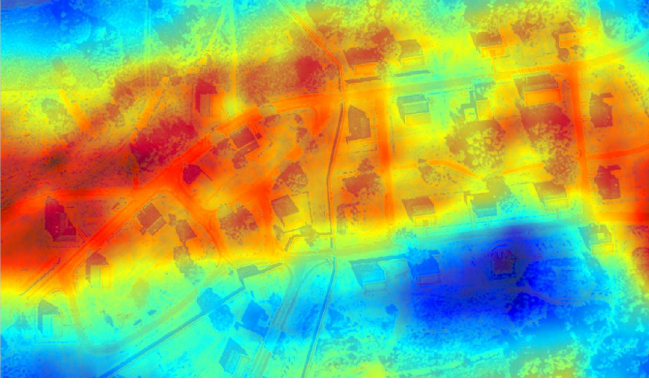


Fig. 3: The confidence map of the frame. Red indicates high confidence while blue indicates low confidence. The black area represents the highest confidence, which suggests that the UAV is at taking off position. Best viewed in color.

In case global localization fails, we need to label the position manually. Although manual labeling may be more reliable, we still propose an autonomous global localization algorithm to make our framework complete.

B. Pose tracking

After initialization, the UAV position will be tracked based on local image registration. In this section, OF based motion estimation as well as HOG and PF based image registration will be introduced.

1) *Position prediction*: To narrow down the search, we make a rough guess and confine the matching around the predicted position by estimating the inter-frame motion. To obtain the motion, the points to be tracked are selected based on [12], and iterative Lucas-Kanade method with pyramids [13] is used to construct the OF fields. The inter-frame translation could be derived from two approaches based on [14], [15] respectively, both relying on supplementary information from the onboard avionic system and assuming the ground plane is flat.

Structure from Motion: For an interest point P , its coordinates (x, y) in the camera frame and 3-D position $(X, Y, Z)^T$ are related by $x = f \cdot \frac{X}{Z}, y = f \cdot \frac{Y}{Z}$ according to projective projection, where f is the focal length. Taking derivatives on both sides, we have $\dot{x} = v_x = f(\frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2}), \dot{y} = v_y = f(\frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2})$ where v_x, v_y are the OF.

Let the camera motion be expressed as a translation, $\mathbf{T} = (T_x, T_y, T_z)^T$ and a rotation, $\Omega = (\omega_x, \omega_y, \omega_z)^T$, the velocity of the feature point \mathbf{V} is defined by $\mathbf{V} = -\mathbf{T} - \Omega \times \mathbf{P}$, whose explicit form is (2).

$$\begin{aligned}\dot{X} &= -T_x - \omega_y Z + \omega_z Y \\ \dot{Y} &= -T_y - \omega_z X + \omega_x Z \\ \dot{Z} &= -T_z - \omega_x Y + \omega_y X\end{aligned}\quad (2)$$

Therefore, v_x, v_y are related to the motion by (3).

$$\begin{aligned}v_x - (-\omega_y f + \omega_z y + \frac{\omega_x xy}{f} - \frac{\omega_y x^2}{f}) &= \frac{T_z x - T_x f}{Z} \\ v_y - (-\omega_x f + \omega_z x + \frac{\omega_y xy}{f} - \frac{\omega_x y^2}{f}) &= \frac{T_z y - T_y f}{Z}\end{aligned}\quad (3)$$

The rotational motion $(\omega_x, \omega_y, \omega_z)$ are read from IMU and feature depth Z is obtained by the barometer. Hence the terms on the left hand side are measurable. A linear equation set could be formulated for many feature points and the translation could be determined.

Homography decomposition: Homography describes the relationship between co-planar feature points in two images, from which the motion dynamics could be derived according to (4).

$$\mathbf{H} = \mathbf{R} + \frac{1}{h}\mathbf{T}\mathbf{N}^T \quad (4)$$

The \mathbf{R} and \mathbf{T} are the inter-frame rotation and translation, \mathbf{N} is the normal vector of the ground plane, and h is the altitude. $\mathbf{R}, \mathbf{N}, h$ are obtained from the onboard sensors and \mathbf{T} can be calculated as (5).

$$\mathbf{T} = h(\mathbf{H} - \mathbf{R})\mathbf{N} \quad (5)$$

2) *Image descriptor*: To choose the appropriate image descriptor, we use images typically encountered when drawing particles. Those at taking off position are given in Fig. 4. The first image is the onboard image, while the second one is its counterpart in the map. Two subimages, whose locations are close to the ground truth location, are extracted from the map as outliers. The image descriptor should be able to maximize the relative difference between the ground truth and the outliers.



Fig. 4: Images for comparison. First one is the onboard image, and the second one is its ground truth match. The third and fourth images are subimages located close to the ground truth.

	Image 1	Image 2	Image 3
MI	1	1.253	1.208
CC	1	1.328	1.320
LSS	1	1.055	1.025
HOG	1	1.829	2.428

TABLE I: Comparing the image descriptors, HOG features give 1.829 and 2.428 for outliers, which are the largest distances.

We choose four commonly used approaches for registering images of different modality, namely Mutual information (MI) used in [16], Sample correlation coefficient (CC), Local

Self Similarity (LSS) descriptors proposed in [17] and HOG features. The performance of different image descriptors are demonstrated in Table I, where the second, third, fourth images in Fig. 4 are assigned the numbers 1, 2, 3. The correlation values are transformed to distance by $d = 1 - \text{correlation}$, and the values are normalized using the ground truth distance. MI is calculated based on Eq. 6, where $H(X), H(Y)$ are marginal entropies, $H(X, Y)$ is the joint entropy. CC is computed according to Eq. 7, where x_i, y_i are measurements in X, Y . LSS descriptors are extracted from the images at 5 pixels apart, and the distance is computed by L_2 norm. For HOG, the value is obtained based on the correlation coefficient of the HOG features extracted from the images.

$$I(X; Y) = H(X) + H(Y) - H(X, Y) \quad (6)$$

$$r_{xy} = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2 \sum_{i=1}^N (y_i - \bar{y})^2}} \quad (7)$$

From the comparison results, it is obvious that HOG is superior to other image descriptors, since MI and CC could not differentiate the outliers as good as HOG. In addition, LSS is the worst, barely recognizing the outliers.

HOG is widely used for pedestrian detection due to its robustness, which renders it suitable as a image descriptor. To construct HOG, 1D point derivative masks are convolved with the image to get the gradients. Gradient histograms are constructed in cells and blocks. A cell contains 8×8 pixels whereas a block is consisted of 2×2 cells. The gradient orientations are divided into 9 bins and every pixel in the cell votes for two adjacent bins weighted by gradient magnitude. Additionally, bilinear interpolation is used to compute the contribution of a pixel to the cells containing it, wherein the importance of pixels follows Gaussian distribution with respect to the block center. The histograms are normalized locally to compensate for illumination variance. Clipped L_2 norm normalization scheme is performed to the histogram of every block. Because the blocks are overlapped, every cell contributes to multiple blocks, significantly improving the performance of HOG. Eventually the histograms are vectorized to form a 1D feature.

Unlike object detection, no training is available in navigation. Therefore, we use HOG in a holistic manner as an image descriptor to encode the gradient information in multi-modal images.

The HOG glyph [18] is visualized in Fig. 5. It is evident that the gradient patterns remain similar even though the onboard image undergoes photometric variations compared with the map. In particular, the structures of road and house are clearly preserved.

During offline preparation phase, a lookup table is constructed to store the HOG features extracted at every pixel in the map. In this way, the HOG features for the map are

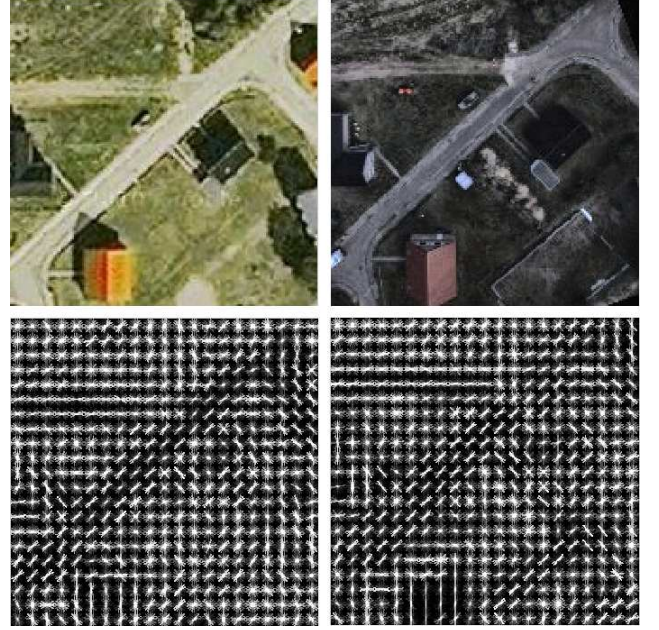


Fig. 5: Visualization of HOG histograms. First row: subimage of reference map and onboard image. Second row: HOG glyph. The gradient patterns for houses and roads are quite similar in HOG glyph.

retrieved from the table when registering images online to save computation time.

3) *Similarity metrics*: Several metrics are compared to compute the similarity of HOG descriptors. Since HOG descriptors are essentially histograms, we measure their similarity using Correlation, Chi-Square, Intersection, and Bhattacharyya distance. To choose the optimal metric, we also use images presented in Fig. 4.

	Image 1	Image 2	Image 3
Correlation	1	1.829	2.428
Chi-Square	1	1.810	2.009
Intersection	1	0.954	0.970
Bhattacharyya	1	1.260	1.248

TABLE II: Comparing similarity metrics, correlation is the best for differentiating the outliers, since the distances of 1.829, 2.428 are the largest.

The comparison of similarity metrics is summarized in Table II. Correlation and Intersection measures similarity while Chi-Square and Bhattacharyya measures distance. Since distance is used in PF, we transform similarity values to distance by $d = 1 - \text{correlation}$. The distance values are then normalized with respect to the ground truth value.

It turns out that all metrics are able to produce minimum distance for the matched pair except for Intersection. Moreover, Correlation differentiates outlier better than Chi-Square and Bhattacharyya as the distance of outliers are larger. Therefore, Correlation is chosen as the similarity metric for HOG descriptors in our experiment.

4) *Confined localization*: Comparison of HOG features is time consuming. Therefore, the traditional sliding window approach seems unfit as it demands more computational resources. Inspired by the tracking algorithms, we employ PF as in [19] to estimate the true position. Furthermore, in order to reduce the number of particles, we confine our search in the vicinity of the predicted position, adopting a coarse to fine procedure.

Particle filter: There are N particles, and for each particle p , its properties include $\{x, y, H_x, H_y, w\}$, where (x, y) specify the top left pixel of the particle, (H_x, H_y) is the size of the subimage covered by the particle and w is the weight. The (x, y) is generated around the predicted position, while (H_x, H_y) equals to the size of the onboard image.

The optimal estimation of the posterior is the mean state of the particles. Suppose each p predicts a location l , then the estimated state is computed in Eq. 8.

$$E(l) = \sum_{i=1}^N w_i l_i \quad (8)$$

Based on the predicted state (x_p, y_p) of where the UAV could be in the next frame, we calculate the likelihood that UAV location (x_c, y_c) is actually at this location. After the particles are drawn, the subimages of the map located at the particles are compared with the current frame. To estimate the likelihood, we use Gaussian distribution to normalize these distance values based on Eq. 9, where d is the distance between the two images under comparison, σ is the standard deviation, \hat{w} is then normalized based on the sum of all weights to ensure that w is in the range $[0, 1]$.

$$\hat{w} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-d^2}{2\sigma^2}\right) \quad (9)$$

We do not use a dynamical model here to propagate the particles. Instead, we initialize the particles in every frame using OF estimation, similar to [20], for the UAV moves erratically due to wind and the camera is not stabilized. Furthermore, we have to conduct coarse to fine search and the particle number changes from frame to frame. Moreover, the particles will concentrate on those that have large weights when using motion model after several iterations. The particles whose weights are trivial will be discarded, causing degeneracy. Re-initialization works like resampling in addressing this issue.

Coarse to fine search: At the beginning, the particles are drawn around the taking off position. Subsequently, OF provides translation between consecutive frames, and the predicted position is updated by accumulating the translation prior to image registration.

Around the predicted position, the search is conducted from coarse level to fine level to reduce the computational burden, similar to the coarse-to-fine procedure described in [21]. For the coarse search, N particles are drawn randomly in a rectangular area, whose width and height are both s_c , with a large search interval Δ_c . The fine search, on the other hand, is carried out in a smaller area with size s_f and search

interval Δ_f . Different from [21], HOP relies mainly on coarse search which is often quite accurate. If the minimum distance of coarse search is larger than a threshold τ_d , then the match is considered invalid. Only when coarse search fails to produce valid match do we conduct fine search. Fine search still centers at the predicted position and the coarse search result is discarded.

When the minimum distances in both coarse and fine search are above the threshold τ_d , indicating that image registration result is unreliable, the predicted position by OF is retained as the current location. If the motion is large, UAV conducts global search for re-initialization.

IV. EXPERIMENT

To evaluate the performance of HOP, we use the aerial images we have collected, which are displayed in the video accompanying the paper. The platform used to collect these images will be described, and then the pre-processing procedure as well as the parameter setting are elaborated. Next, we run experiments to analyse the effect of OF, compare two outlier rejection schemes, and compare HOP with OF which is used as baseline.

A. Setup

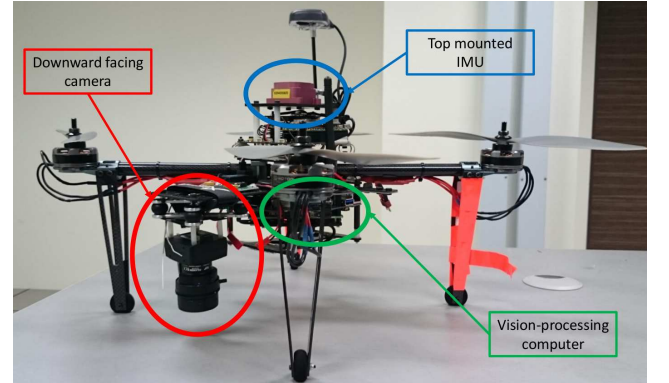


Fig. 6: Quadrotor platform used for image capture, whose onboard sensors include IMU, Mastermind, and a camera.

In order to test the performance of the proposed algorithm, data is collected onboard using a quadrotor platform shown in Fig. 6, whose dimension is 35 cm in height and 86 cm in diagonal width with a maximum take-off weight of 3 kg. Its onboard sensors include an IG-500N attitude and heading reference system (AHRS) from SBG Systems and a downward-facing PointGrey Chameleon camera. An Ascending Technologies Mastermind computer is used to decode and log data in a 5 Hz rate. The flight test is carried out at Oostdorp⁺, the Netherlands, and the onboard images and IMU data collected are the actual fly-off data for the final round of the 2014 International Micro Air Vehicle Competition (IMAV 2014). The quadrotor flies at about 80 m above the ground and sweeps overhead the whole Oostdorp village. The speed is about 2 m/s and the total flight duration is about 3 min.

⁺Google Map location [52.142815, 5.843196]

B. Preprocessing

The reference with size $w \times h$ is the Google Map of Oostdorp village, which corresponds roughly to a 300×150 m region. The resolution of the map is low, for about 3.15 pixels represent 1 m. The onboard frames are undistorted and then pre-processed to have the same orientation and scale as the reference map. First, the onboard frame is rotated by the yaw angle. Second, the frame is scaled to 3.15 pixels/m. The frames are cropped from the center with the same size $s_i \times s_i$.

C. Parameters

The most important parameters in HOP are N and s_c . More N increases the accuracy of the weighted center but demands more computational resources. Likewise, larger s_c ensures the matching is robust to jitter while smaller s_c reduces the time consumed. Hence, we trade off the robustness and efficiency when determining those parametric values. Regarding the sensitivity of HOP to these parameters, it is found that N should be larger than 40 to have sufficient particles to make a valid estimation. Meanwhile, s_c should be larger than 35 to account for the inaccuracy arises from OF.

In our experiment, the varied parameters used are set as follows. During preprocessing, $w \times h = 850 \times 500$, and $s_i = 180$. We use the HOG in OpenCV with cell size 32×32 , block size 64×64 , block stride 32×32 . For coarse to fine search, we set $N = 50$, $s_c = 40$, $\Delta_c = 4$, $s_f = 20$, $\Delta_f = 1$, $\sigma = 0.01$, $\tau_d = 0.75$.

D. Results

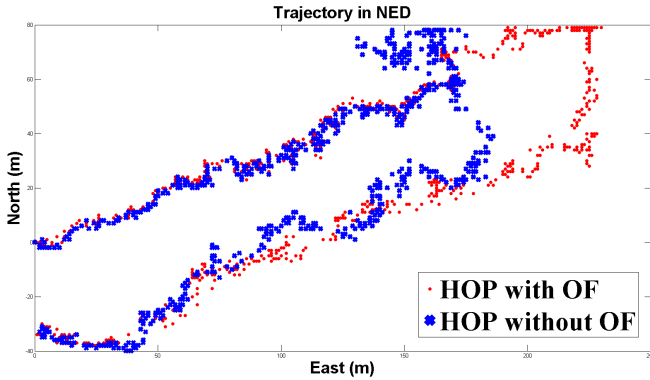


Fig. 7: Comparison of HOP and HOP without OF. Red dots represent HOP, while blue crosses represent HOP without OF. Without OF, HOP fails when there is large motion or unreliable match, while using OF handles those issues effectively. Best viewed in color.

1) *Effect of prediction:* We compare the effect of OF based position prediction as shown in Fig. 7. UAV localization fails without OF, especially when the motion is large. The localization is only resumed when the UAV revisits a previously seen place. By contrast, OF overcomes the large motion by moving the search region to the predicted position.

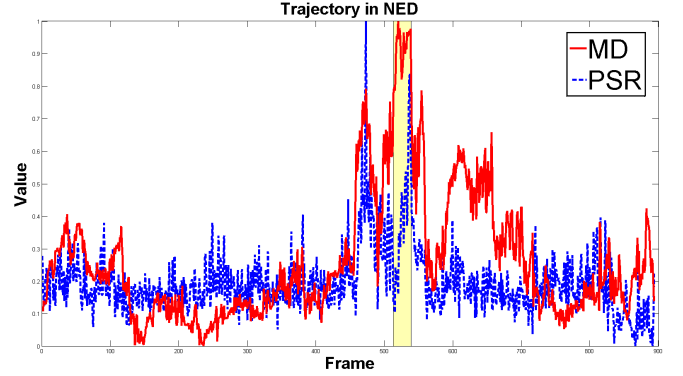


Fig. 8: Comparison of outlier rejection methods. Red line depicts MD, whereas blue line depicts PSR. The yellow area highlighted corresponds to the frames with large illumination change, when the match becomes unreliable. MD is significantly higher for unreliable match in comparison to PSR. Best viewed in color.

2) *Outlier rejection:* We compare two methods to reject outliers, namely minimum distance (MD) as well as Peak to Sidelobe Ratio (PSR) defined in [11]. PSR is computed according to Eq. 10, where d_{min} is the minimum distance, and μ, σ are the mean and standard deviation of the distance for all particles in the search region, excluding a circle with 5 pixels radius around the minimum position.

$$\Theta = \frac{d_{min} - \mu}{\sigma} \quad (10)$$

The solid line in Fig. 8 is MD while the dashed line is PSR. Both MD and PSR are normalized to the range $[0, 1]$. MD peaks within the highlighted interval and attains large values, when the match becomes unreliable due to significant illumination change (refer to video). In contrast, PSR remains oscillating in that region. MD outperforms PSR in the sense that it indicates when the match is incorrect.

3) *Path analysis:* The red line in Fig. 9 depicts the GPS ground truth, the brown line indicates the localisation from OF alone as a baseline. The green dots represent the HOP output and the blue crosses are outliers. The sequence is challenging for image registration for three reasons. Firstly, the Google Map is not up to date, and the trees and buildings are missing in some region. Secondly, the map image only has low resolution, which may reduce the amount of visible gradient patterns. Moreover, the scene undergoes large illumination change (refer to video).

The dead-reckoning of OF gives poor results as the drift accumulates over time. On the other hand, the green dots follow the GPS closely, which corroborates the effectiveness of HOG based image registration. Furthermore, OF deals with unreliable match effectively as well when there is obvious illumination change. In comparison to ground truth, the root mean square error (RMSE) of HOP is 6.773 m. The errors are quite small compared with a 3 m RMSE for GPS, which indicates the accuracy of HOP. Admittedly, the

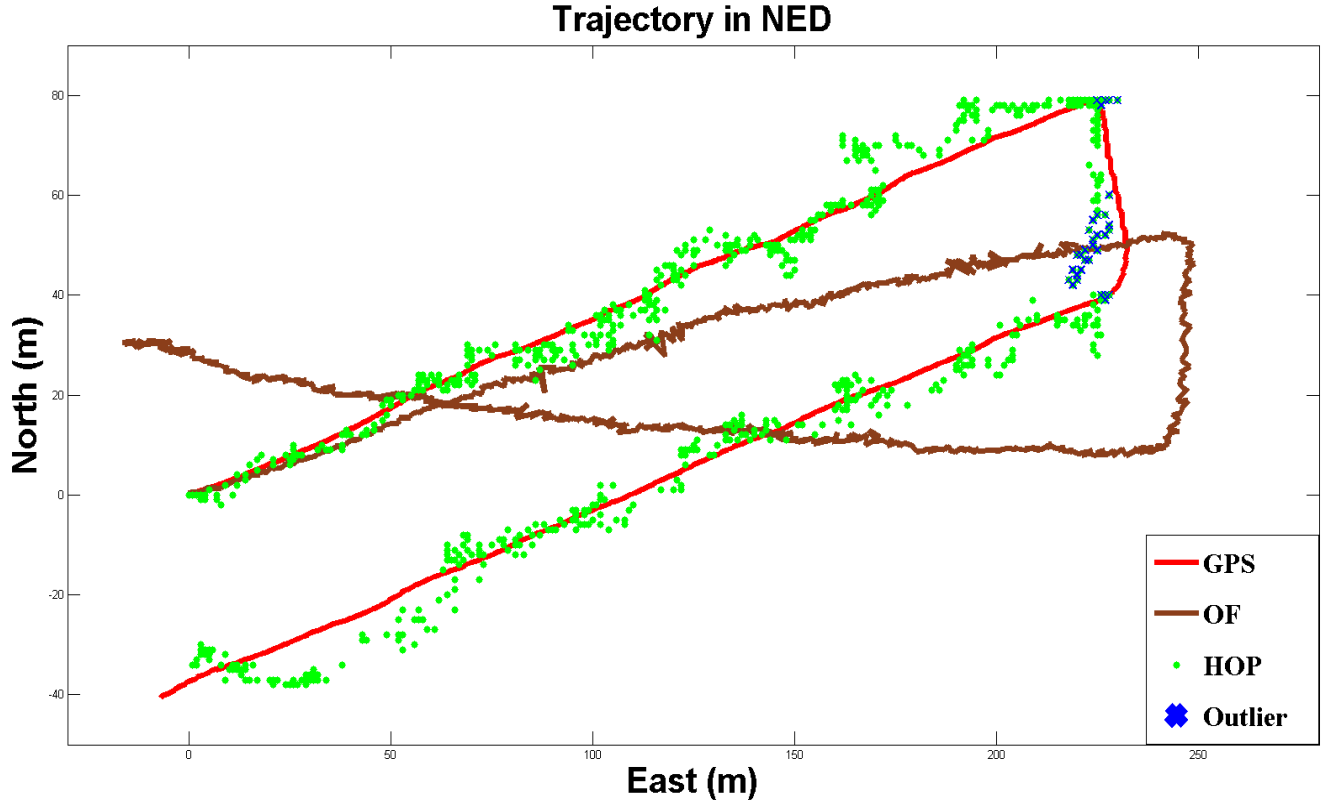


Fig. 9: Path analysis comparing GPS (red line), OF (brown line), and HOP (green dots for reliable matches and blue crosses for outliers where OF is used). HOP performs markedly better than the OF baseline. Best viewed in color.

oscillation of HOP is sometimes significant, mainly due to wind and jitter of UAV. A gimbal could be used to mitigate the oscillation.

Image registration failure constitutes 7% where OF prediction is retained. The outliers mainly concentrate at two regions, where either there are few gradient patterns in the scene or has significant illumination change.

In this flight data, the velocity of UAV is slow and thus lost recovery is not necessary. In addition, HOP still works when the frame rate is increased by taking one in every two frames, doubling the UAV velocity. Nevertheless, when we triple the UAV velocity, large motion occurs and lost recovery is needed.

E. Speed

HOP is implemented in C++ using OpenCV. It is not optimized for efficiency and runs at 15.625 Hz on average for each frame on a Intel i7 3.40 GHz processor. The current update rate of HOP is sufficient for the position measurement, since its output will be fused with onboard INS at 50 Hz [15]. Practically, the resulting trajectory is smooth as long as HOP is faster than 10 Hz.

V. CONCLUSIONS

This paper presents the first study of localization using HOG features in GPS-denied environment by registering aerial images and Google Map. The experiment using flight data shows that HOP could supplement GPS since its error is

comparatively small. As the dataset is limited, our approach constitutes an initial benchmark, and we will make the onboard images and the reference map publicly available for research and comparison purposes.

For future research directions, we will install a gimbal to stabilize the camera against wind and vibration, and a thermal camera for navigation at night. Subsequently, we will perform more evaluation on challenging environments, including day and night conditions. We will also use HOP to give state feedback to an actual system.

ACKNOWLEDGMENT

The authors would like to thank the members of NUS UAV Research Group for their kind support.

REFERENCES

- [1] A. Viswanathan, B. F R Esteves Pires, and D. Huber, "Vision-based robot localization by ground to satellite matching in gps-denied situations," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 192–198.
- [2] G. Pang and U. Neumann, "The gixel array descriptor (gad) for multimodal image matching," in *WACV*, 2013, pp. 497–504.
- [3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.
- [4] C.-F. Zhu, S.-X. Li, H.-X. Chang, and J.-X. Zhang, "Matching road networks extracted from aerial images to gis data," in *Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on*, vol. 2. IEEE, 2009, pp. 63–66.

- [5] D.-Y. Gu, C.-F. Zhu, J. Guo, S.-X. Li, and H.-X. Chang, "Vision-aided uav navigation using gis data," in *Vehicular Electronics and Safety (ICVES), 2010 IEEE International Conference on*. IEEE, 2010, pp. 78–82.
- [6] T. Patterson, S. McClean, P. Morrow, and G. Parr, "Utilizing geographic information system data for unmanned aerial vehicle position estimation," in *2011 Canadian Conference on Computer and Robot Vision (CRV)*. IEEE, 2011, pp. 8–15.
- [7] G. Conte and P. Doherty, "An integrated uav navigation system based on aerial image matching," in *Aerospace Conference, 2008 IEEE*. IEEE, 2008, pp. 1–10.
- [8] F. Lindsten, J. Callmer, H. Ohlsson, D. Tornqvist, T. Schon, and F. Gustafsson, "Geo-referencing for uav navigation using environmental classification," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1420–1425.
- [9] A. L. Majdik, Y. Albers-Schoenberg, and D. Scaramuzza, "Mav urban localization from google street view data," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3979–3986.
- [10] C. Le Barz, N. Thome, M. Cord, S. Herbin, and M. Sanfourche, "Global robot ego-localization combining image retrieval and hmm-based filtering," in *6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, 2014, p. 6.
- [11] D. S. Bolme, J. R. Beveridge, B. Draper, Y. M. Lui *et al.*, "Visual object tracking using adaptive correlation filters," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2544–2550.
- [12] J. Shi and C. Tomasi, "Good features to track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1994, pp. 593–600.
- [13] J. yves Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," *Intel Corporation, Microprocessor Research Labs*, 2000.
- [14] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, "An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1736–1741.
- [15] S. Zhao, F. Lin, K. Peng, B. M. Chen, and T. H. Lee, "Homography-based vision-aided inertial navigation of uavs in unknown environments," in *Proc. 2012 AIAA Guidance, Navigation, and Control Conference*, 2012.
- [16] W. M. Wells, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis, "Multi-modal volume registration by maximization of mutual information," *Medical image analysis*, vol. 1, no. 1, pp. 35–51, 1996.
- [17] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.
- [18] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "Hoggles: Visualizing object detection features," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–8.
- [19] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and vision computing*, vol. 21, no. 1, pp. 99–110, 2003.
- [20] A. Yao, D. Uebbersax, J. Gall, and L. Van Gool, "Tracking people in broadcast sports," in *Pattern Recognition*. Springer, 2010, pp. 151–161.
- [21] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 10, pp. 2002–2015, 2014.