Mono camera-based GPS spoofing detection for aerial vehicles

Peter Petro [*] **Peter Bauer** [**]

[*] *Budapest University of Technology and Economics, Budapest, Hungary*
[**] *Systems and Control Laboratory, Institute for Computer Science and Control (HUN-REN SZTAKI), HUN-REN, Budapest, Hungary*
(e-mail: bauer.peter@sztaki.hun-ren.hu)

Abstract: This paper presents mono camera-based GPS spoofing detection for aerial vehicles utilizing only image information besides the initial orientation of the vehicle. Orientation information is propagated and motion direction is estimated solely from the mono camera images through the estimation of the essential matrix. Histograms of Oriented Displacements and their correlation are considered to detect spoofing considering GPS and image-based data. Straight and turning simulated flight trajectories of a fixed wing research drone with different turbulence levels are compared to evaluate the method. The results are promising with timely detection of every spoofing scenario and without false alarm. The exploration of real flight data is the topic of future development.

Keywords: GPS spoofing detection, mono camera relative motion, UAV

1. INTRODUCTION

With the increasing use of unmanned aerial vehicles (UAVs) the threat of GPS spoofing becomes more and more imminent Kerns et al. (2014), Sorbelli et al. (2020), Khan et al. (2021), Talaei Khoei et al. (2022) , Septentrio (2023). Meng et al. (2021) categorizes the possible spoofing attacks into 1. forwarding spoofing, 2. generative spoofing and 3. track tracking spoofing. The second one is implementable and complex enough to be a real threat for UAVs modifying their flight trajectory while the autopilot 'thinks' that it guides the UAV on the originally targeted trajectory. Talaei Khoei et al. (2022) gives a good overview about the possible spoofing detection methods such as external UAV characteristics-based (utilizing IMU measurements e. g. Feng et al. (2017), Jiang et al. (2022)), artificial intelligence methods (e. g. Jiang et al. (2022)) and vision-based (e. g. Qiao et al. (2017), Xue et al. (2020), Varshosaz et al. (2020)). Additional methods can be GPS receiver-based detection Schmidt et al. (2020), statistical methods Meng et al. (2021) and the application of a companion drone Hiba et al. (2023).

The vision-based techniques apply satellite imagery matching Xue et al. (2020) which is resource intensive, mono camera and IMU-based method Qiao et al. (2017) where the inclusion of the IMU and the solution for scale ambiguity is not really discussed and stereo camera-based solutions Varshosaz et al. (2020). The current article targets to achieve and possibly improve the results of Varshosaz et al. (2020) based on a mono camera instead of the stereo making system requirements lower. Besides implementing a mono camera-based method another contribution is the comparison of the Histograms of Oriented Displacements (HODs) through their correlation coefficient instead of angle or taxicab distances. This is a normalized and possibly more meaningful measure for which threshold selection is easier. Another difference from the referenced work is the application of simulated flight trajectories and synthetic image sequences instead of real data but in close to real conditions.

The structure of the paper is as follows. Section 2 introduces the considered flight trajectories which even make it possible to explore turbulence effects on GPS spoofing detection. Section 3 introduces the method applied for spoofing detection while Section 4 presents tuning and detection results. Finally, Section 5 concludes the paper.

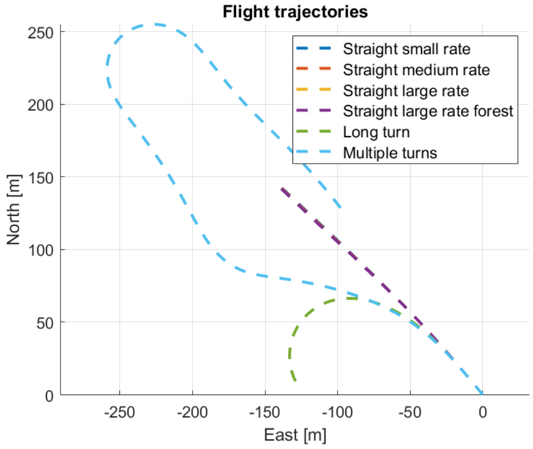

Fig. 1. The flight trajectories

2. FLIGHT TRAJECTORIES

Test trajectories for GPS spoofing detection were generated with the Matlab simulation of the Sindy aircraft

(SZTAKI (2014)) including a waypoint tracking controller. First, straight trajectories with different turbulence levels (small, medium, large) were generated resulting in different angular rates and slightly different trajectories through the flight. The large turbulence flight was repeated with vegetation only background to challenge feature detection and tracking (see Fig. 3, referenced as 'forest case'). Then a long left turn and a trajectory with multiple turns were generated to simulate GPS spoofing (see Fig. 1). Note that in Fig. 1 the four different straight trajectories cover each other. The GPS spoofing detection should not differentiate the straight trajectories from each other despite the different noises but should differentiate long turn and multiple turn ones from each other and from the straight ones. Thus the generated trajectory set is satisfactory to check noise tolerance and the basic detection capabilities of the method. To better evaluate detection, gradually deflected trajectories should be considered but this will be the topic of future development.

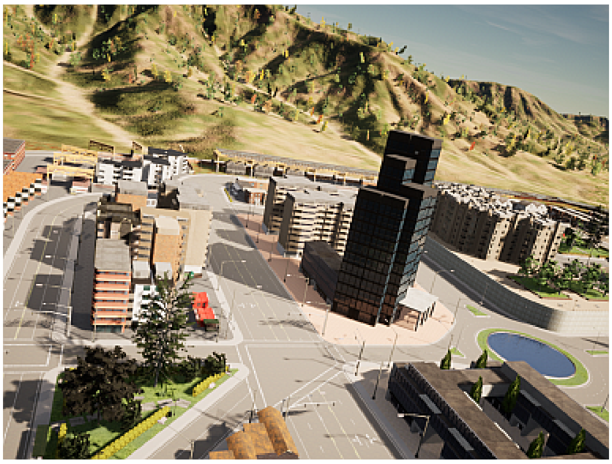

Fig. 2. Example city plus vegetation camera image from Unreal-Carla

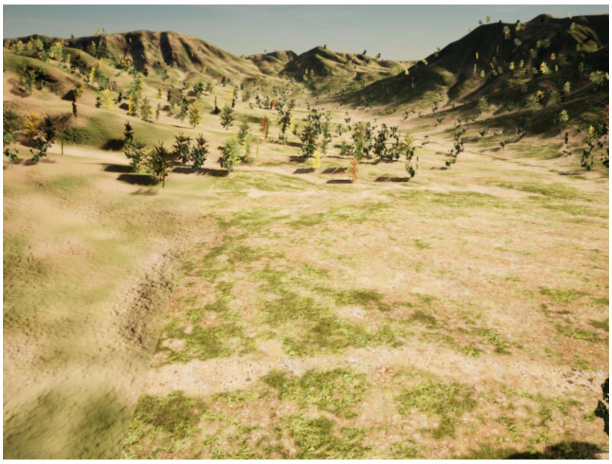

Fig. 3. Example vegetation camera image from Unreal-Carla (forest case)

All of the trajectories were sampled with $\Delta t = 0.02s$ saving aircraft position and orientation. This data was exported to Unreal-Carla where sequences of images were generated considering a downward tilted HD camera (1280 × 960 pixels with $f = 1108.5$). A city landscape surrounded by hills and vegetation was considered as the scene (see Fig.s 2 and 3) and images were generated with $0.02s$ sampling (50 fps). Unreal- Carla environment was selected as it became a kind of standard for virtual scenario generation in the last years.

3. GPS SPOOFING DETECTION FROM CAMERA IMAGES

Throughout the work the considered coordinate systems are the North-East-Down (NED) earth, the X_B, Y_B, Z_B body and X_C, Y_C, Z_C camera systems see Fig.s 4 and 5. From now on a vector v^{S1} means that its coordinates are represented in $S1$ system, a rotation matrix T_{S1S2} means a rotation from $S2$ to $S1$ system. The position of the camera in the body system is $r_C^B = [x_b \; y_b \; z_b]^T$ and the rotation matrix from body to camera system is $T_{CB}(\phi_c, \theta_c, \psi_c)$ defined by Z-Y-X Euler angle rotations. The rotation from earth to body is defined similarly as $T_{BE}(\phi, \theta, \psi)$. Note that T_{CB} also includes an axis swap (see Fig. 4). In the generation of the Unreal-Carla images $r_C^B = [1 \; 0 \; 0]^T \; m$ camera position and $\theta_c = -20°$ (down) camera angle were considered.

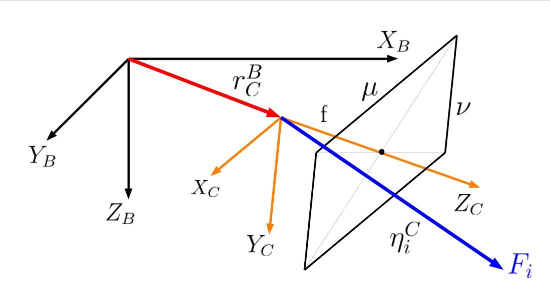

Fig. 4. The applied coordinate systems

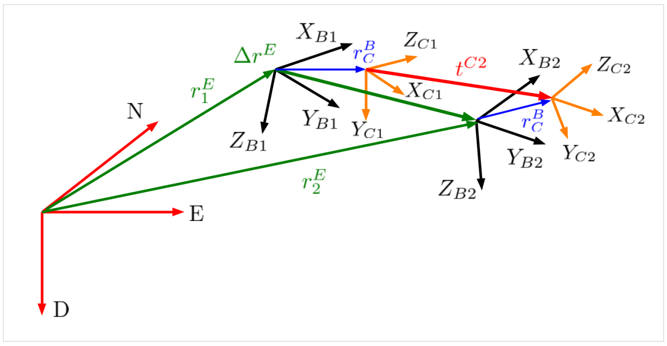

Fig. 5. One step body and camera motion and rotation

GPS spoofing can be detected based on the camera images if one can reconstruct at least the motion direction in NED system and compare it to the direction of motion from the GPS measurements. A one step motion and rotation of UAV body and so the attached camera is visualized in Fig. 5. Here $\Delta r^E = r_2^E - r_1^E$ is the translation in NED system which direction should be estimated from the camera images. However, considering the essential matrix and its decoupling to translation and rotation (see e. g. Hartley and Zisserman (2003)) the direction of motion between the two camera frames $\overline{t^{C2}}$ (from now on $\overline{(.)}$ will denote a normalized unit vector) and the relative rotation T_{C1C2} can be obtained. Considering the calculation of $t^E = T_{EC2}t^{C2}$ based on Fig. 5 results in:

$$\begin{aligned} t^E_{C1} =& r^E_1 + T_{EB1}r^B_C, \;\; t^E_{C2} = r^E_2 + T_{EB2}r^B_C \\ t^E =& t^E_{C2} - t^E_{C1} = \Delta r^E + (T_{EB2} - T_{EB1})r^B_C \\ t^E =& T_{EB1}T_{BC}T_{C1C2}t^{C2} = \\ & \Delta r^E + T_{EB1}(T_{BC}T_{C1C2}T_{CB} - I)r^B_C \end{aligned} \tag{1}$$

Here indexes 1 and 2 mean first and second position of body or camera system (see Fig. 5). (1) shows that besides the GPS measured motion of the body system Δr^E an additional term appears from the non CG-centered (CG means center of gravity) camera system and rotation. As one can only obtain $\overline{t^E_I}$ from $\overline{t^{C2}}$ as shown in (2) this term can not be removed from the image measurement (I stands for image). However, it can be added to the GPS measurement Δr^E making a fair comparison between the motion directions as shown by (3) substituting t^E from (1) (G stands for GPS).

$$\overline{t^E_I} = T_{EB1}T_{BC}T_{C1C2}\overline{t^{C2}} \tag{2}$$

$$\overline{t^E_I} \sim \overline{t^E_G} = \frac{t^E}{\|t^E\|} \tag{3}$$

(2) and (3) show that direction of motion estimation requires the body to earth rotation matrix of the first system at every time besides the constant camera to body (T_{BC}) and the estimated T_{C1C2} matrices. As the Euler angles are usually estimated based on IMU-GPS integration (see e. g. Gebre-Egziabher and Gleason (2009)) they can be corrupt in case of GPS spoofing that's why their application should be avoided if possible. Assuming that at the beginning of the mission the GPS signals are healthy the body to earth rotational matrix can be propagated solely from image data and an initial T_{EB1} value as follows:

$$T_{EB2} = T_{EB1}T_{BC}T_{C1C2}T_{CB} \rightarrow T_{EB1} = T_{EB2} \tag{4}$$

After the first frame this propagation is applied instead of the simulated Euler angles to prove the feasibility of this concept.

To find and track the features for essential matrix calculation the OpenCV functions *goodFeaturesToTrack* (Shi-Tomasi corner detector with maximum 30 corners, 0.5 quality level, 50 minimum distance and 5 blocksize) and *calcOpticalFlowPyrLK* (sparse optical flow with iterative Lucas-Kanade method applying pyramids with 15×15 window size, 7 max. level, 10 count and 0.03 epsilon) were applied downsampling the $50fps$ data with $\Delta t = 0.1s$ meaning $10fps$ which is a realistic value for onboard image processing with state of the art hardware. The Lucas-Kanade method was applied both forward (frame 1 to 2) and backward (frame 2 to 1) to filter out the false feature pairs. Only the features found also backward on frame 1 are preserved. The *findEssentialMat* (with RANSAC method, 0.999 prob. and 0.5 threshold) and *recoverPose* OpenCV functions are applied to get T_{C2C1} and t^{C2}. The selected OpenCV functions are a kind of industry standard for the given problems.

After calculating the motion directions from images ($\overline{t^E_I}$) and GPS data ($\overline{t^E_G}$) the HODs from Varshosaz et al. (2020) are considered calculating only the heading directions in the horizontal plane. The HOD calculation (for both vectors) is implemented as follows:

$$\begin{aligned} & BI = \frac{360°}{8} = 45° \\ & BN = 0 : BI : 360 - BI \;\;\; bins \\ & \psi = atan2\left(\frac{t^E(2)}{t^E(1)}\right)/\pi \cdot 180° \;\; \rightarrow \;\; [0, 360°) \\ & 1stBN = floor\left(\frac{\psi}{BI}\right) \\ & 2ndBN = (1stBN + 1 \;\, or \;\, 1) \\ & 1stM = 1 - \frac{(\psi - 1stBN \cdot BI)}{BI} \\ & 2ndM = \frac{(\psi - 1stBN \cdot BI)}{BI} \end{aligned} \tag{5}$$

Eight bins (BN) are created among which the heading directions are divided. Modifications relative to Varshosaz et al. (2020) are the conversion of heading to the range 0 to 360 degrees, the handling of the case if the $1stBN$ bin index is the last in the bin sequence ($2ndBN$ should be the first) and the omission of the length of vectors from the measures ($1stM$, $2ndM$) as $\overline{t^E_I}$ and $\overline{t^E_G}$ are both unit vectors. The measures are cumulatively added to the selected (through the indexes $1stBN$ and $2ndBN$) bin counters to create the histogram.

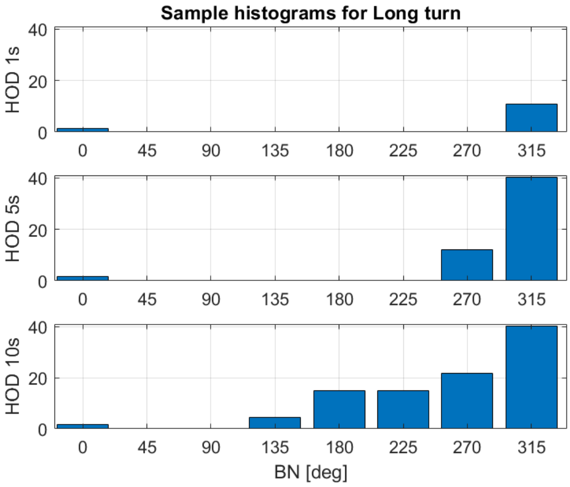

Fig. 6. Sample HODs from different phases of long turn flight

Sample HOD diagrams from different flight phases in the long turn maneuver are plotted in Fig. 6 (obtained from GPS data). The figure shows that when the flight starts at $\psi = -40°$ it is closest to the 315° bin so its measure is the dominant together with some minor portion in the

0° = 360° bin which is the second closest. Upon turning the heading enters more and more bins into decreasing heading direction so their measures are also increased. Note that between 5s and 10s there is no increase in the 315° bin value as it is left before.

Based-on the HOD for image and GPS data the spoofing can be detected if the motion directions become different. Contrary to Varshosaz et al. (2020) neither angle distances (HOD_AD) nor taxicab distances (HOD_TD) are applied rather the correlation coefficient (Wikipedia (2023)) of the HODs is calculated as it should well characterize similarity or dissimilarity and threshold selection is easier because it is normalized to $[-1, 1]$ independent of the considered data.

After establishing the method for GPS spoofing detection its tuning and evaluation is done considering the flight data sets presented in Section 2.

4. TUNING AND RESULTS

Tuning of the GPS spoofing detection means the pairwise comparison of all possible trajectories calculating the HOD correlations and noting their minimum values in Table 1. Note that correlations are started to be calculated after the first ten samples. In the table the spoofed cases (different compared trajectories) are denoted by boldface measures. Note that the small, medium, large and large forest scenarios cover the same straight trajectory so they are not spoofed the difference is in turbulence parameters and vegetation background instead of the city (forest case). Based on these values the decision threshold can be easily selected if the non-spoofed minimum values differ well from the spoofed ones. The results show that the minimum non-spoofed measure is 0.999 while the spoofed cases can usually decrease to around 0.8 values. Fig.s 7 and 8 show that the non-spoofed sections give correlation measures very close to 1 while upon the occurrence of trajectory deviation the measure starts to decrease as expected (see Fig. 8). Finally, a 0.997 threshold was selected below which GPS spoofing is declared. It is worth mentioning that such values very close to 1 can only occur because of the ideal simulated flight data for real flight data sets these measures are expected to degrade.

Table 1. Minimum HOD correlation values (boldface values are for spoofed trajectories)

Trajectories	Small	Medium	Large	Forest	Long	Multiple
Small	0.999	0.999	0.999	0.999	**0.825**	**0.78**
Medium	0.999	0.999	0.999	0.999	**0.825**	**0.78**
Large	0.999	0.999	0.999	0.999	**0.825**	**0.78**
Forest	0.999	0.999	0.999	0.999	**0.825**	**0.78**
Long	**0.825**	**0.866**	**0.834**	**0.81**	0.999	**0.88**
Multiple	**0.79**	**0.79**	**0.8**	**0.78**	**0.87**	0.999

Evaluation of the decisions was done running the algorithm again for every trajectory pair and noting the decision. There was no false decision as expected from Table 1 despite some small dissimilarity between small and large turbulence rate trajectories shown in Fig. 9. In the figure (such as in Fig. 10) the estimated motion directions are plotted at the given GPS positions. For the spoofed trajectory the image-based directions cover the GPS-based ones showing the good performance of image-based motion direction estimation. Considering the image result-based propagation of the body to earth rotational transformation (4) this is a really good result without any drift in the directions along time.

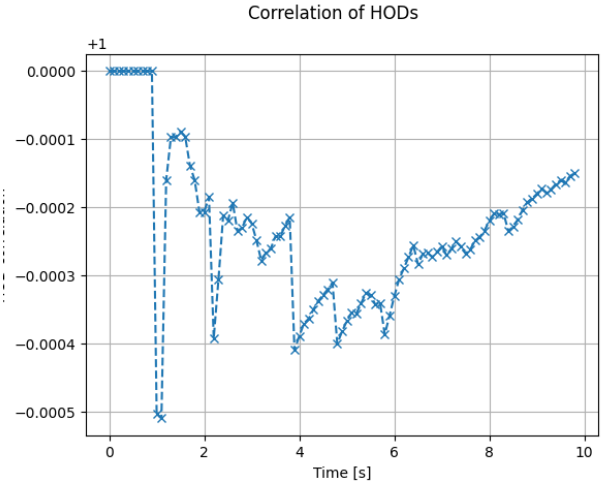

Fig. 7. HOD correlation between small rate and large rate forest straight scenarios (non-spoofed). Note that the minimum value is $1 - 5 \cdot 10^{-4}$!

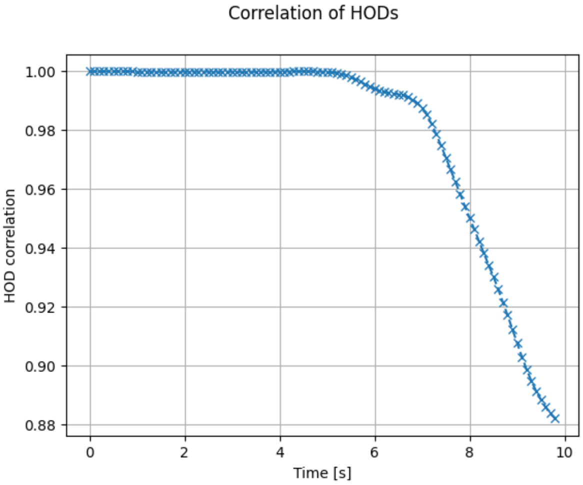

Fig. 8. HOD correlation between long turn and multiple turn scenarios (spoofed)

Table 2. Detection results

Detection measures	Small	Medium	Large	Forest	Long	Multiple
Long time [s]	0.9	0.9	0.9	0.9	-	1.0
Long distance [m]	5.86	5.86	5.9	5.9	-	9.84
Multiple time [s]	0.7	0.7	0.7	0.7	0.3	-
Multiple distance [m]	4.75	4.75	4.77	4.77	3.6	-

In case of different trajectories all simulated spoofing occurrences were successfully detected. The time for spoofing detection was calculated measuring the time between spoofing alarm and the time when the trajectories are 2m apart from each other (stated to be the beginning of spoofing). Detection times are summarized in Table 2 together with the distance between the trajectories when spoofing is declared. The detection times are 0.3 to 1.0 seconds being much lower than with the image-based method presented in Qiao et al. (2017) (5s). The distances at detection are about 5-6m in most of the cases. What is interesting is the difference between long-multiple and multiple-long pairs. In the first case image-based directions of the long turn case are compared to GPS track of the multiple turns and vice versa and this causes a difference in the detection time. Comparing the results to Varshosaz et al. (2020) this method can similarly only detect the change in the direction of motion no velocity change along the trajectory can be detected (as motion vectors are normalized). However, here the HOD correlation measure does not go back to 1 after the turn contrary to SEDCP, HOD_AD and HOD_TD in the reference. This is because here the global flight direction is estimated and evaluated contrary to the local flight direction in the measurement window in Varshosaz et al. (2020).

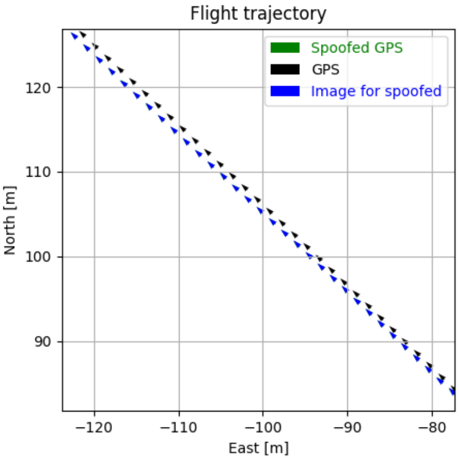

Fig. 9. Tracks of small rate and large rate forest straight scenarios

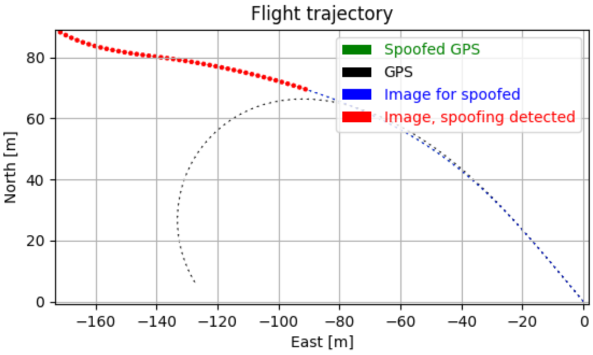

Fig. 10. Tracks of long turn and multiple turns scenarios

As a comparison the HOD_TD measures presented in Varshosaz et al. (2020) are also calculated and shown in Fig.s 11 and 12. The figures show that threshold selection is not as easy as for the correlation because this measure continuously increases even in the non-spoofed case (Fig. 11).

Finally, Fig. 10 shows the instance of HOD detection (by considering the HOD correlation) by coloring the further trajectory with red dots.

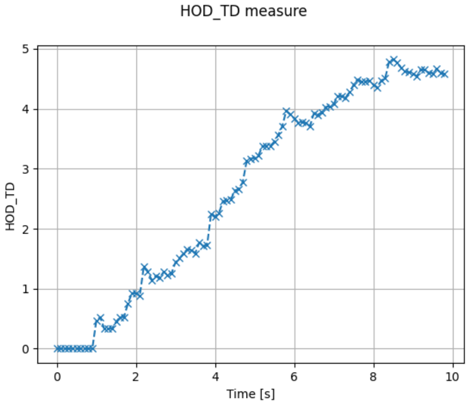

Fig. 11. HOD_TD measure between small rate and large rate forest straight scenarios

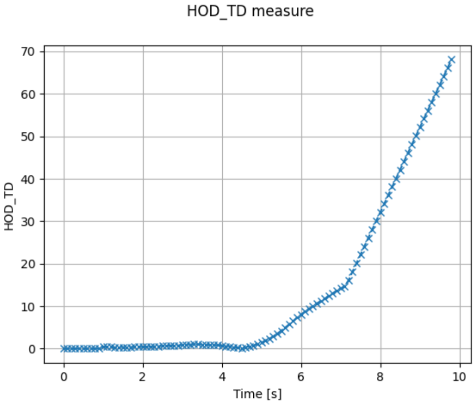

Fig. 12. HOD_TD measure between long turn and multiple turns scenarios

5. CONCLUSION

The paper presents mono camera-based GPS spoofing detection utilizing the direction of motion and rotation information from the essential matrix. Straight and turning flight trajectories are generated with the dynamic simulation of the Sindy test aircraft considering also different turbulence levels. Synthetic images are generated in Unreal-Carla considering city and rural scenarios. After presenting the coordinate systems and the description of UAV motion from images and GPS coordinates image processing and the estimation of the essential matrix are briefly discussed. The global orientation of the UAV is propagated starting from a known rotation matrix and utilizing the relative rotations of the image frames. Finally, the motion direction of the aircraft in the horizontal North-East frame can be obtained both from images and GPS data. To detect GPS spoofing the headings from the motion directions are calculated and registered in Histograms of Oriented Displacement (HODs) based on the literature. However, contrary to the literature these are global heading angles instead of local ones in a moving window. The introduced measure for spoofing detection is the correlation coefficient of GPS and image-based HODs as it is normalized and so threshold selection is easy. After tuning the HOD correlation threshold the performance of the algorithm is evaluated considering all possible pairs of the simulated flight trajectories. Neither false alarm nor missed detection resulted having about 0.3-1.0s detection time (from start of spoofing) and about 5-6m distance between the normal and spoofed trajectories when spoofing is detected. The HOD taxicab distance measure from the literature is also examined showing a continuous increase even for non-spoofed trajectories which makes threshold selection problematic. Future directions can be the extension to 3D GPS spoofing detection, the utilization of IMU data to estimate aircraft velocity and so detect spoofing of the flight velocity and the application of the methods on real flight data and camera images which is missing from the current article.

ACKNOWLEDGEMENTS

Part of the research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems. (RRF-2.3.1-21-2022-00002).

Project no. TKP2021-NVA-01 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme.

The authors gratefully acknowledge the help of Antal Hiba (Computational Optical Sensing and Processing Laboratory, SZTAKI, HUN-REN) with the generation of the Unreal-Carla image series based-on our Sindy simulated flight trajectories.

REFERENCES

Feng, Z., Guan, N., Lv, M., Liu, W., Deng, Q., Liu, X., and Yi, W. (2017). Efficient drone hijacking detection using onboard motion sensors. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, 1414–1419. doi:10.23919/DATE.2017.7927214.

Gebre-Egziabher, D. and Gleason, S. (2009). *GNSS Applications and Methods.* Artech House, Inc.

Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in computer vision.* Cambridge University Press.

Hiba, A., Kortvelyesi, V., Kiskaroly, A., Bhoite, O., David, P., and Majdik, A. (2023). Indoor vehicle-in-the-loop simulation of unmanned micro aerial vehicle with artificial companion. In *2023 International Conference on Unmanned Aircraft Systems (ICUAS).*

Jiang, P., Wu, H., and Xin, C. (2022). Deeppose: Detecting gps spoofing attack via deep recurrent neural network. *Digital Communications and Networks*, 8(5), 791–803. doi:https://doi.org/10.1016/j.dcan.2021.09.006. URL https://www.sciencedirect.com/science/article/pii/S2352864821000663.

Kerns, A.J., Shepard, D.P., Bhatti, J.A., and Humphreys, T.E. (2014). Unmanned aircraft capture and control via gps spoofing. *Journal of Field Robotics*, 31(4), 617–636. doi:https://doi.org/10.1002/rob.21513. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21513.

Khan, S.Z., Mohsin, M., and Iqbal, W. (2021). On gps spoofing of aerial platforms: a review of threats, challenges, methodologies, and future research directions. *PeerJ. Computer science*, 7, e507.

Meng, L., Yang, L., Ren, S., Tang, G., Zhang, L., Yang, F., and Yang, W. (2021). An approach of linear regression-based uav gps spoofing detection. *Wireless Communications and Mobile Computing*, 2021, 5517500. doi:10.1155/2021/5517500. URL https://doi.org/10.1155/2021/5517500.

Qiao, Y., Zhang, Y., and Du, X. (2017). A vision-based gps-spoofing detection method for small uavs. In *2017 13th International Conference on Computational Intelligence and Security (CIS)*, 312–316. doi:10.1109/CIS.2017.00074.

Schmidt, E., Gatsis, N., and Akopian, D. (2020). A gps spoofing detection and classification correlator-based technique using the lasso. *IEEE Transactions on Aerospace and Electronic Systems*, 56(6), 4224–4237. doi:10.1109/TAES.2020.2990149.

Septentrio (2023). Gnss spoofing. Technical report, Septentrio.

Sorbelli, F.B., Conti, M., Pinotti, C.M., and Rigoni, G. (2020). Uavs path deviation attacks: Survey and research challenges. In *2020 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*, 1–6. doi:10.1109/SECONWorkshops50264.2020.9149780.

SZTAKI (2014). Sindy test aircraft. URL http://uav.sztaki.hu/sindy/home.html.

Talaei Khoei, T., Ismail, S., and Kaabouch, N. (2022). Dynamic selection techniques for detecting gps spoofing attacks on uavs. *Sensors*, 22(2). doi:10.3390/s22020662. URL https://www.mdpi.com/1424-8220/22/2/662.

Varshosaz, M., Afary, A., Mojaradi, B., Saadatseresht, M., and Ghanbari Parmehr, E. (2020). Spoofing detection of civilian uavs using visual odometry. *ISPRS International Journal of Geo-Information*, 9(1). doi:10.3390/ijgi9010006. URL https://www.mdpi.com/2220-9964/9/1/6.

Wikipedia (2023). Pearson correlation coefficient. URL https://en.wikipedia.org/wiki/Pearson_correlation_coefficient.

Xue, N., Niu, L., Hong, X., Li, Z., Hoffaeller, L., and Poepper, C. (2020). Deepsim: Gps spoofing detection on uavs using satellite imagery matching. In *Annual Computer Security Applications Conference*, ACSAC '20, 304–319. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3427228.3427254. URL https://doi.org/10.1145/3427228.3427254.