Validation of GPS by Ground Scanning Radar Generated with an on-board wide-angle FMCW radar for General Aviation

Maas, J.B.; Stefanovici, V.; van Gent, R.N.H.W.; Hoekstra, J.M.

Publication date
2020

Document Version
Final published version

Published in
ICRAT 2020

Validation of GPS by Ground Scanning Radar

Generated with an on-board wide-angle FMCW radar for General Aviation

J. Maas
TU Delft
Faculty of Aerospace Engineering
Control and Simulation
Delft, Netherlands
j.b.maas@tudelft.nl

V. Stefanovici
TU Delft
Faculty of Aerospace Engineering
Control and Simulation
Delft, Netherlands

R. van Gent
Selfly BV
Soest, Netherlands

J. Hoekstra
TU Delft
Faculty of Aerospace Engineering
Control and Simulation
Delft, Netherlands

***Abstract*—Many pilots in General Aviation use electronic add-ons aids in flight, which rely on satellite navigation information. This navigation information is often a single point of failure which is undesirable since the pilot relies on the information.**

This paper presents the results of research whether a novel mobile radar station can be used to validate the navigation results from the GPS. The radar transmits signals to the ground, and compares the locations of the reflections to a digital map such as Google maps.

A test flight was performed with a radar system on board. Fifteen different methods for processing the images were investigated, and it was found that Ridge Operators and Entropy Detection are good methods to extract similar features in Google and radar images. These algorithms were always successful in picking the single correct GPS coordinate out of a pool of 300 false ones within *150m* of the correct answer, except when the aircraft was making a turn and the radar was pointed to the sky.

It is concluded that a ground-scanning radar on board can be used to validate the results of a GPS, provided that the radar can observe recognizable features that can be compared to a digital map. The type of image processing used to extract the data is crucial for the application.

Radar; Navigation; GPS; Validation; Direction of Arrival; FMCW; Radar Sensing; Situation Awareness

I. INTRODUCTION

The most important instrument for pilots flying under Visual Flight Rules (VFR) is eyesight. Nevertheless, many different kinds of aides are used in flight. Example functions of these add-ons are collision warnings, navigation assistance and weather forecasts.

Many of these add-ons rely on satellite navigation, such as GPS. But this is often used as a single point of failure, and faults in navigation data can mislead a pilot to taking wrong decisions. This can cause unsafe situations, such as airspace infringements.

In recent years, the availability and quality of microwave sensors and modern sensing techniques have improved significantly. This is partially caused by the demand for self-driving cars and the situation awareness that they need to have. Newly developed microwave sensing hardware can be brought on board of an aircraft, where it may empower new applications for aviation.

A test flight with such hardware was performed (seen in Figure 1), and it was found that structures on the ground were recognizable in the radar output, by the naked eye. This raises the question: is it possible to double-check the GPS results with the data from the ground?

Figure 1: The test aircraft with the radar in the wingtip

In this paper, the possibilities of this manner of navigation are explored. The hardware use in the experiment is described in section II, where it is explained how ground images are created. The test flight performed for the experiment is introduced in section III. Different image processing algorithms for improving the raw results are presented in section IV, and the method for assessing the performance of these algorithms is presented in section VI. The results are

presented in section VI, after which a discussion and conclusion follow in sections VII and VIII.

II. Ground Scanning Hardware

Frequency-Modulated Continuous Wave (FMCW) Radar systems emit microwave signals and receive the reflections of those signals on the surroundings. A comparison of the outgoing and incoming signals yields information about range (*R)* and Doppler velocity (V_R) of the reflecting surface [1], as

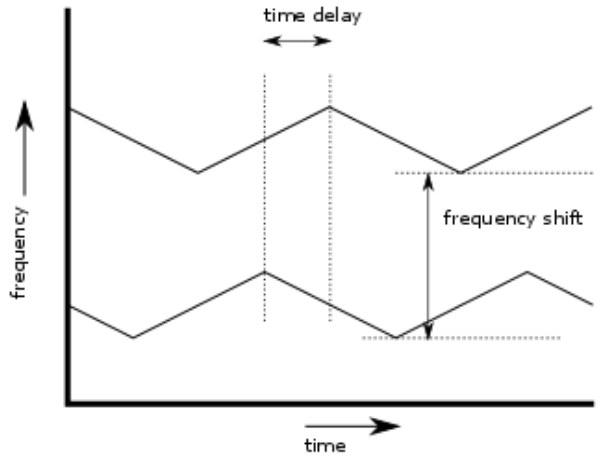

Figure 2: FMCW radar principle to find range (time delay) and Doppler (frequency shift) information by comparing an outgoing and incoming signal

is pictured in Figure 2.

FMCW radars have the properties to be lightweight, cheap and have a low power consumption. This makes them suitable for mobile applications, such as using them on small aircraft. A side-looking radar was mounted on the wingtip of a Socata TB-10, as illustrated in Figure 1 and Figure 3. A radar such as this prototype can weigh 10kgs, be under €5000 and use 50W power.

When an antenna with a wide aperture is used, multiple reflections are recorded simultaneously, and summed together in one returning signal. Fourier analysis helps differentiate different reflections from one another, provided that the reflection sources have unique values for R and V_R.

Direction Of arrival (DoA) estimation is possible for returned radio signals, when they are recorded by multiple adjacent antennas [2]. The phase difference between the incoming signals can be used to extract the incidence angles of the reflections. Together with the range information of a signal, this yields the position of a reflection in 3D.

In this test, a radar system was used with a range of maximum 5*000m*, and a Nyquist velocity of about *80m/s*. For this application, the Nyquist velocity can be considered the maximum possible velocity.

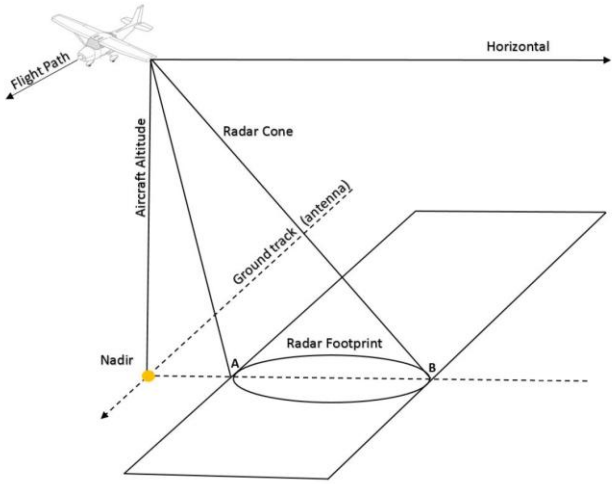

Figure 3: Side looking FMCW concept

A. *Differences with SAR*

Although the images in Figure 1 and Figure 3 may imply that the hardware in this research is an off-the-shelf Synthetic Aperture Radar (SAR) [3], it should be noted that this is not the case.

The radar in this experiment has a wide aperture, being able to observe a large area instantaneously. A SAR has a very narrow aperture, and the width of an image is synthetically

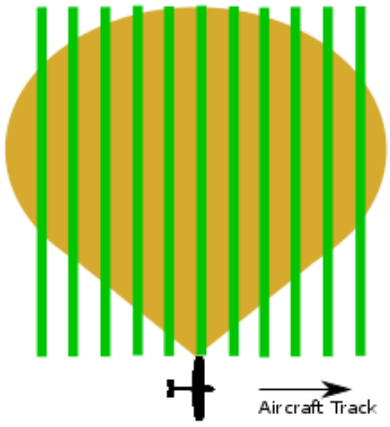

Figure 4: Differences in imagery for a SAR (green) and side looking FMCW radar (orange)

created by the movement of the aircraft. A SAR combines many consecutive measurements into one image, which may be disturbed by attitude changes of the platform. Generating a SAR image takes time, but a FMCW image can be generated instantaneously. SAR images are usually generated by post-processing the data, whereas the results of the FMCW radar can be generated live. These differences are illustrated in Figure 4.

III. TEST FLIGHT

A local flight with the ground scanning radar on board of the aircraft from Figure 1 was performed. The flight took place at Teuge airport (EHTE) in the Netherlands, and consisted of two circuits with a touch-and-go in the middle. The aircraft from Figure 1 was used as the platform. The flight was logged with GPS, and the track is plotted in Figure 5.

Figure 5: Test Flight Track, with the runway and taxiways on the north side

During this flight, the radar was continuously broadcasting and receiving radio signals. When the returns were filtered, they were processed and their locations were projected onto a flat surface. An example screenshot from the flight is shown in Figure 6.

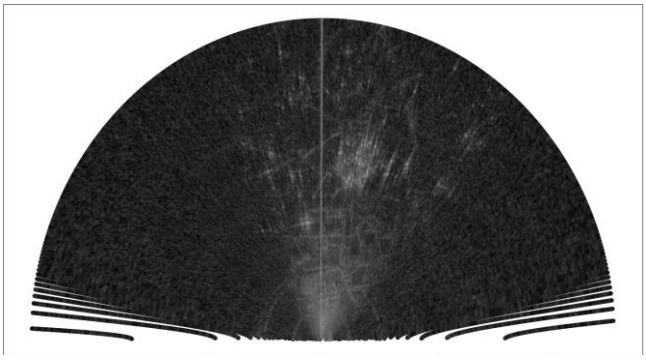

Figure 6: Semicircular scatter map around the aircraft with max range *5km*

In Figure 6, a greyscale image of the local ground surface beneath the aircraft can be found. The aircraft was located at the center of the semicircle in the image. The radar was mounted in the left wing, therefore in this image the aircraft was flying to the right.

The brightness in the greyscale image is dependent on the strength of the received signal. It can be observed that many well-reflecting sources are within sight of the radar equipment. The radar, with an aperture of about 60 degrees in horizontal and vertical direction, does not receive signals from everywhere: the sides of the semicircle are dark since no signal is perceived from that direction.

It was noted that the reflections from Figure 6 appear to form shapes and structures on the ground. These shapes may correspond to buildings, roads, rivers or trees on the ground. To evaluate this statement, the GPS track was used to compare the radar map with a Google map of the same place, same attitude. The result of this is seen in Figure 7.

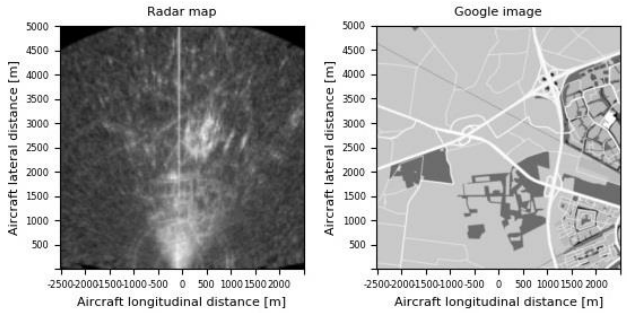

Figure 7: Radar map results compared to Google map

In Figure 7 a comparison is made between the radar map and the Google map. It is seen that the map from Google contains detailed information on roads locations and tree areas, but some of this information also appears to be present in the radar map. The location of the big highway appears to be visible as well in the radar map.

Even though it is difficult for a human to compare the radar and Google maps to one another, it does seem that this is within the capabilities of modern image recognition algorithms. If the right strategy for pre-processing is found, an experiment can be performed to evaluate the comparison algorithm. In section IV, 15 different image processing algorithms are introduced, which will be subjected to the experiment presented in section V.

IV. Image Processing Algorithms

In this section, different algorithms are presented that should make comparisons between radar maps and maps from Google as effective as possible. The selection of algorithms is based on a literature study, as well as simple testing in code. The presence of noise and the quality of the radar images excludes the possibility of using point detection algorithms. As such, image transformation algorithms that are robust to noise are desired.

As seen in Figure 6, a bright area near the center of the semicircle occurs, where the reflections of the radar are stronger than far away. This is a consequence of the shorter range to the aircraft and a higher reflection coefficient, as the ground is more perpendicular to the signal. It was tried to compensate this in an analytic way, but a homogeneous image was never achieved. This means that the solution in recognizing the structures lies in local intensity changes, instead of global variables.

Many of the final algorithms consist of two steps. The first step is a way to map the results of the greyscale to a domain $0 \leq x \leq 1$, and the second step is the way of locally finding the regions of interest. The steps are listed below.

A. *Scaling Steps*

- **Gamma Correction** – Also known as Power Law Transform. This function transforms the input image pixel-wise as a power of gamma, after scaling each pixel to the domain [4].
- **Thresholding** - The creation of a new binary containing the pixel positions of all intensities that are above a certain threshold.
- **Local Histogram Equalization** – A method which modifies intensities in pixels, to evenly stretch out the entire intensity range, reducing any non-linearity within pixel intensities [5].

B. *Transformation Steps*

- **Contour Finding** – A curve joining all of the continuous points (along the borders) that have the same intensity and/or color. The contours are useful for shape analysis and object detection [6].
- **Ridge Operators** – Algorithm thar relies on the eigenvalues of the Hessian matrix, calculated from intensities within the image in order to detect ridge structures where the intensity changes horizontally, but not along the structure [7].
- **Straight Line Hough Transform** – A common algorithm that assigns pixels to the existence of a line that meets width, length and direction properties [8].
- **Blob Detection** – Find a group of pixels that share some common property, for example colour or intensity [9]. In this experiment intensity is used. Since this method is about areas instead of lines in the image, it will not be compared to a Google map, but to the ESA Sentinel 1 database, which contains satellite reflectivity data. This database has more information about areas, but less about lines in the map, as can be seen in the comparison in Figure 8.

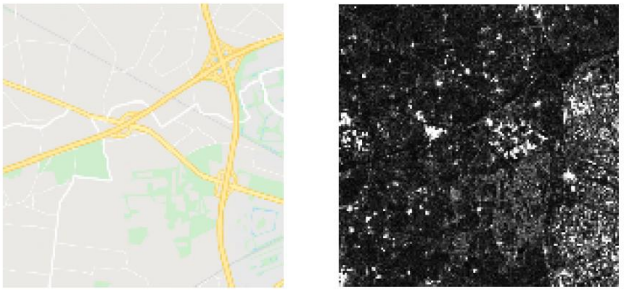

Figure 8: Comparison of Google map and Sentinel 1 map

- **Entropy Detection** – The entropy filter is capable of detecting slight variations in the local grey level distribution. It is used to determine regions in the image where many of these small changes are present [10].
- **Watershed** – A marker-controlled Watershed is an image transformation algorithm that interacts with a greyscale image and considers the image as a topographic surface, calculating the gradient of a high energy regions to low energy regions [11].
- **CGAN-CRF** – Unsupervised learning algorithm based on hierarchical Conditional Generative Adversarial Nets (CGAN) and Conditional Random Fields (CRF) Geo Land sensing – categorizing each pixel in satellite images into a category such that we can track the land cover of each area [12].
- **CAE-TVL** – A pre-trained Convolutional Autoencoder with Total Variation Loss (CAE-TVL) for satellite image segmentation as well as generic images [13].

With these different algorithms available, 15 methods are selected for testing in the experiment. These are found in Table 1.

TABLE 1: IMAGE PROCESSING ALGORITHM SELECTIONS

Method	***Transformation 1***	***Transformation 2***
1	Gamma Correction	None
2	Gamma Correction	Blob Detection
3	Gamma Correction	Thresholding
4	Thresholding	Contour Finding
5	Thresholding	Entropy Detection
6	Thresholding	Hough Line Transform
7	Thresholding	Blob Detection
8	Histogram Equalization	Ridge Operators
9	Histogram Equalization	Entropy Detection
10	Histogram Equalization	Watershed Marker
11	Histogram Equalization	Contour Finding
12	None	CGAN-CRF
13	None	CAE-TVL
14	Histogram Equalization	CGAN-CRF
15	Histogram Equalization	CAE-TVL

V. EXPERIMENT SETUP

In order to compare a radar map and a Google map to one another, the Structural Similarity Index Measure (SSIM) is used to quantify the likeliness of the two images. The SSIM is chosen because it takes into account differences in geometry of detected points, even with a strong presence of noise. The SSIM takes into account the luminance, contrast and structure of regions within the image.

The goal of using the radar data is to improve possibilities of navigation, and therefore the radar map will not only be compared to the local Google map based on the aircraft GPS, but also to 300 false alternatives, which are generated by altering the GPS data. An alternative GPS location lays at a distance of *0-150m* from the true GPS location, and a heading disturbance of maximum *15degrees* is used. This is illustrated in Figure 9 and Figure 10.

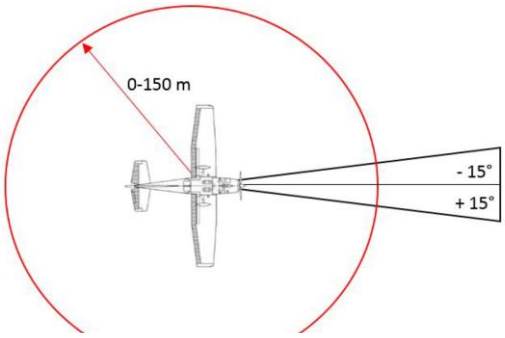

Figure 9: Generated GPS faults for alternative locations

In Figure 10 the red circle from Figure 9 is visible and drawn to scale. It can be seen that even with a maximum of *150m* range difference, the two images will still be similar, given the total range of *5km*. The task for the computer will be to distinguish the differences between 300 similar images, and to find the correct Google map which corresponds to the radar map.

For algorithms 2 and 7, the comparison will not be made to a Google map but to the Sentinel 1 database, as discussed in IV.B. This only changes the image for comparison, all other steps are identical.

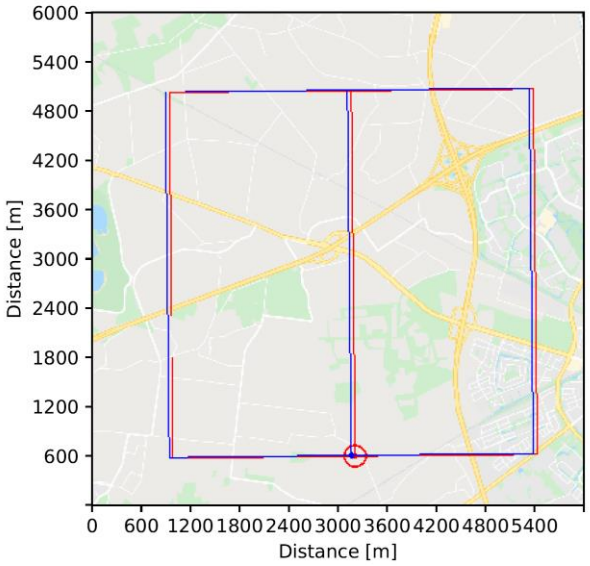

Figure 10: The original Google map and an alternative map with GPS errors

A. *Dependent Variables*

Three dependent parameters are selected to evaluate the performance of the algorithm. For all 300 images, the SSIM is computed. The Google map with the highest SSIM is selected to be the best match for the given radar map. The GPS coordinates of this generated Google map are then compared to the true GPS coordinates of the measurement. The heading and position errors are noted down, and used as the first and second independent variables.

The third and final dependent parameter is the number of false positives. Out of the 300 false alternatives, it is noted how many score a better SSIM than the original Google map. This number is expressed as a percentage, and should be as low as possible.

B. *Test points*

The experiment is conducted in two steps. First, four test points are selected by hand. They were chosen on different parts of the flight for which clear reflections were to be seen in the radar output. The point from Figure 7 is one of them, the other three are similar.

The radar results from these four points are compared to all 15 algorithms from Table 1. The three algorithms that perform best under these 'easy' conditions will then be subjected to a further test.

In the second test the best algorithms are subjected to 20 radar data points. These points selected randomly, on condition that the aircraft is not performing a take-off or landing. In Figure 11 the locations of the selected points are shown.

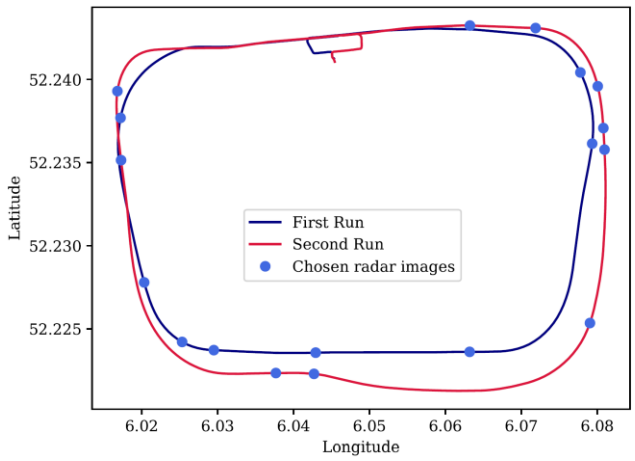

Figure 11: The randomly selected points for the detailed experiment

VI. Results

The numerical results of the first test are shown in Table 2. In this table, the results for the four data points are grouped together, and the average is taken.

It is seen that methods 8 and 9 yield superior results. For all four experiments, these algorithms were successful in picking the correct GPS image on the hand of the radar map out of a pool of 300 alternative GPS maps. Out of the other algorithms, number 5 was chosen to complement the set of three, predominantly based on its low number of false positives.

After experiment A, the second experiment was run, with only the three best methods and 20 selected points. The distance results of this experiment are shown in Figure 12, and all results are numerically summarized in Table 3.

Table 2: Experiment 1 results for 15 methods

Method	**Average results over 4 data points**		
	Heading deviation [deg]	***Position error [m]***	***False Positives***
1 Gamma	7	108	53%
2 Gamma Blob	6	68	52%
3 Gamma Thresh.	3	75	50%
4 Thresh. Countour	11	46	39%
5 Thresh. Entropy	**5**	**24**	**23%**
6 Thresh. Hough Line	4	53	67%
7 Thresh. Blob	8	35	28%
8 Histogram Ridge	**0**	**0**	**0%**
9 Histogram Entropy	**0**	**0**	**0%**
10 Histogram Watersh.	6	119	38%
11 Histogram Contour	2	37	89%
12 CGAN	14	126	55%
13 CAE	12	124	53%
14 Histogram CGAN	12	52	45%
15 Histogram CAE	6	112	35%

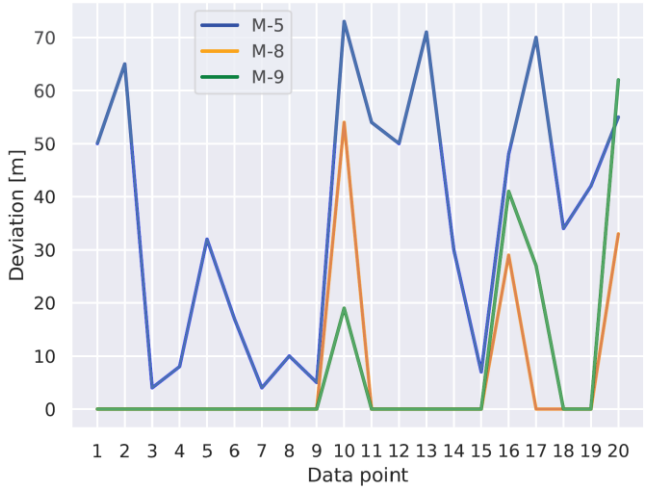

Figure 12: position error for the 3 best methods for all 20 data points

Table 3: Experiment 2 results for the best 3 methods

Method			
	Heading deviation [deg]	***Position error [m]***	***False Positives***
5	5	45	42%
8	1	1	14%
9	2	5	9%

It is seen that also for the second experiment, methods 8 an 9 are performing well over all. For the majority of the data points, the algorithms manage to find the exact GPS location out of the 300 false alternatives. For method 8, 3 points were not related to the correct GPS position, but a position dozens of meters away. For method 9, 4 points differed from the true GPS. In the next section, these results are discussed in detail.

VII. Discussion

In this section, the results from the two experiments are discussed in order. First are the results of experiment 1.

A. Experiment 1

The results for all methods are discussed briefly. The selected methods for experiment 2 are discussed at the end of the list.

1) Only Basic Operation

For **method 1** only a scaling step was applied on the images. The number of false positives for this method was 53%. This is close to what would be expected if maps were evaluated by a randomizer. In that case the number of false positives would lay around 50%. This highlights the case that special operations are necessary in order to compare the images to each other. Also, a simple operation such as thresholding is not sufficient for the second step, as seen with **method 3.**

2) Blob detection

Methods 2 and 7 use Blob detection. This method is based on the assumption that buildings leave dense spots of illumination in the image. For Blob detection, the comparison is made to the Sentinel 1 database (Figure 8), as the reflectivity information in that database is shaped much like blobs, and therefore suitable for the algorithm. Blob detection reduces a spot of illumination to a circle on the image – apparently the presence of noise disturbed the circles too much for a decent comparison to be made.

3) Contour Finding

Methods 4 and 11 use Contour finding. It appeared that the noise present in the images distorted the resulting contours by such an extent that it was impossible to compare them to the contours based on Google maps.

4) Hough Line Transform

In **method 6** it was attempted to connect different visible speckles on the radar map, to form roads. The presence of the noise speckles caused that many different lines could be drawn through the images, which could all be roads in reality. It is possible to tune the Hough Line Transform algorithm with multiple settings, but these attempts were not successful.

5) Watershed Marker

The Watershed Marker was used in **method 10**. This algorithm segments the image in multiple regions, based on greyscale results. The method works poorly, since noise is not at all filtered.

6) Computer learning algorithms

Methods 12, 13, 14 and 15 are based on pre-trained neural network algorithms, with or without scaling before that. All the false positive percentages lay around 50%. 50% is the expected result for a randomizer, so it is concluded that none of the attempts for implementing AI were successful.

7) The Successful Algorithms

Two algorithms are found to be successful in comparing the radar data to Google maps data. These are **Ridge Operators** for **method 8** and **Entropy Detection** for **methods 5 and 9**.

The method of **Ridge Operators** is used to find edges in the image. In order to do this, the local Hessian Matrix of the image is computed for each pixel, as well as the corresponding eigenvalues. If the eigenvalues are strong in one particular direction, then there must be a line running in that direction. This method is not hindered by noise is used in the medical industry to detect blood vessels in noisy images.

Entropy Detection borrows from the physical term ‘entropy’, in the sense that it is a method that quantifies how much entropy is present locally in an image. Regions with much contrast between pixels have higher entropy. More homogeneous regions have low entropy. It appears that this method has the same regions highlighted in the Google and radar images.

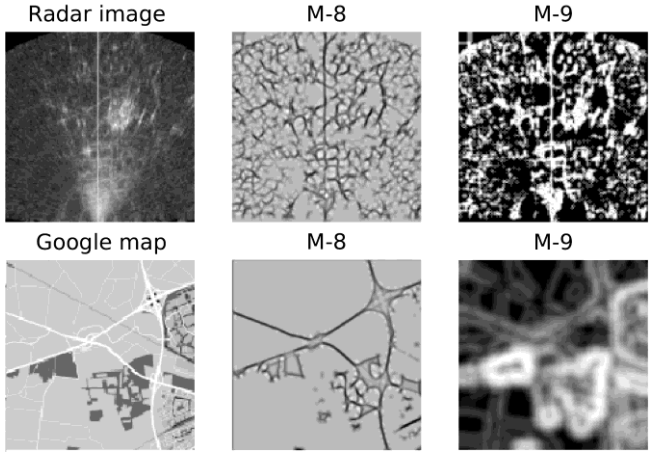

Figure 13: Ridge Operators (M8) and Entropy Detection (M9) results

A comparison of Ridge Operator results and Entropy Detection (only for method 9) is shown in Figure 13. It is seen that the algorithm is applied both on the radar image and on the google map. The two results 'M8' are compared to each other using the SSIM, and the two results 'M9' are compared to each other using the SSIM as well.

B. Experiment 2

It is found that from the three algorithms selected for the test, **method 8** has the best results of all. In the second experiment, where the algorithms are fed 20 random in-flight situations, the algorithm could find the aircraft position for 17 situations. The algorithm for **method 9** is also a good candidate for being put in practice: it found the correct location 16 times.

In Figure 12 it was observed that three data points yield bad results for all methods. These points are 10, 16 and 20. It was investigated if there was a reason behind the collective failure. It was found that these data points are taken when the aircraft was making a right turn. Because of this the left wing went up, and the radar was pointed to the sky. Ground reflections are still received because of the high aperture of the radar, but reflections are difficult to distinguish. This caused trouble for the algorithms. This may be compensated by putting a second radar on the right wing, which would point downward. But because only right turns are made in this flight, this cannot be verified in this experiment.

When the algorithms were off, the error was always in the order of magnitude of tens of meters. If the navigation algorithm is performed at reasonable frequency, an outlier filter can help to differentiate the inconsistent results from the correct ones.

The differences in the two algorithms are too small to yield a definitive answer to the question which one is better. A next step for testing can be to asses the algorithms in a more open setting, where the task is not to find the best answer out of a discrete set, but to find the best matching location on a map. Such an optimization problem for finding the best match can be solved with extra code, such as a Newtonian solution or a Genetic Algorithm.

VIII. CONCLUSION

In this paper, the results are presented of an experiment with an on board wide-angle FMCW radar that received reflections from the ground during flight. It was noted that visually, structures on the ground could be identified in the reflections, and it was tested whether computer vision could also distinguish these reflections.

Fifteen different methods were tested. First, they were subjected to a simple experiment, and the best three algorithms were put to the test for an extensive comparison. It was found that a method based on Ridge Operators and one based on Entropy Detection performed exceptionally well. The Ridge Operator algorithm was slightly better, but no definitive verdict can be given based on this study alone.

Both of the algorithms were able to validate the location of GPS, even when provided with 300 similar alternatives that were close around the GPS location. Errors were found when the aircraft was making a turn, thereby aiming the radar away from the ground.

It is concluded that observations of ground reflections from airborne radars can be used to validate the functioning of a GPS, as long as the radar can observe features on the ground that can be correlated to that on a map of the environment. The use of processing algorithms to extract features from the radar data is crucial for this application.

REFERENCES

[1] A. G. Stove, "Linear FMCW radar techniques", IEEE Proceedings for Radar and Signal Processing, pp 343-350, 1992

[2] A. Paulraj, T. Kailath, "Direction of arrival estimation by eigenstructure methods with unknown sensor gain and phase", International Conference on Acoustics, Speech, and Signal Processing, 1985

[3] D. Nitti, F. Bovenga, M. Chiaradia, M. Grecco, G. Pinelli, "Feasibility of using synthetic aperture radar to aid UAV navigation", Sensors, vol. 15 issue 8, 2015, pp 13334-18359

[4] E. Reinhard, W.Heidrich, et.al. "High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting" Morgan Kaufmann. P.82 ISBN 9780080957111 (2010)

[5] D. Ketcham, R.Lowe, J.Weber, "Image enhancement techniques for cockpit displays" technical report, Hughes Aircraft 1974

[6] W. Lorensen, E. Cline, "Marching cubes: a high resolution 3d surface construction algorithm", Computer Graphics (SIGGRAPH 87 Proceedings), July 1987, pp 163-170

[7] C. Ng, M. Yap, N. Costen, B. Li, "Automatic Wrinkle Detection Using Hybrid Hessian Filter" Asian Conference on Computer Vision (pp 609-622)

[8] C. Galamhos, J. Matas, J. Kittler, "Progressive probabilistic Hough transform for line detection" IEEE Computer Society Conference on Computer Vision and Pattern Recogniition, 1999

[9] L. Nretzner, T. Lindeberg, "Feature Tracking with automatic selection of spatial scales" Computer Vision and Image Understanding, 1998

[10] S. Van der Walt et. al. "scikit-image: image processing in Python", PeerJ 2:e453 https://doi.org/10.7717/peerj.453

[11] L. Shafarenko, M. Petrou, J. Kittler, "Automatic Watershed Segmentation of Randomly Textured Color Images", IEEE Transactions on Image Processing, vol6, no. 11, November 1997

[12] L. Matikainen et al., "Object-based analysis of multispectral airborne laser scanner data for land cover classification and map updating", ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 128, 2017

[13] C. Wang, B. yang, Y. Liao, "Unsupervised image segmentation using convolutional autoencoder with total variation regularization as preprocessing" IEEE International Conference on Acoustics, Speech and Signal Processing, 2017, pp 1877-1881