Aircraft Navigation on Taxiways: Evaluation of Line Detection Algorithms Proposed for Automotive Applications

Claire Meymandi-Nejad, Esteban Perrotin, Ariane Herbulot, Michel Devy

Aircraft navigation on taxiways: evaluation of line detection algorithms proposed for automotive applications

Claire Meymandi-Nejad
LAAS-CNRS, Univ. de Toulouse, CNRS, INSA, Toulouse, France
cmeymand@laas.fr

Esteban Perrotin
LAAS-CNRS, Univ. de Toulouse, CNRS, Toulouse, France
Airbus, Toulouse, France
eperrotin@laas.fr

Ariane Herbulot
LAAS-CNRS, Univ. de Toulouse, CNRS, UPS, Toulouse, France
herbulot@laas.fr

Michel Devy
LAAS-CNRS, Univ. de Toulouse, CNRS, Toulouse, France
devy@laas.fr

ABSTRACT

While working on aircraft navigation on taxiways, the line detection is one of the main challenging problems to be solved. This subject has been widely studied in the literature in the automotive field. In this paper, we propose a comparison of three line detection algorithms based on methods validated in the automotive field but transposed in aeronautics where this subject has not been widely addressed. Some problematics appear: the tarmac environment differs from the usual road model and the camera's position impacts the visibility on the image. The first method presented here uses a particle filter while the second one is based on the Hough transform. In the second method, we perform a color-based detection and introduce a method to compute the reference color, using technical specifications for airport markings. The last method is the LaneNet neural network. Criteria such as the precision or the max range of the detection are computed and exploited to discuss the algorithms relevance. The comparison is performed on both simulated images (from a product of the OKTAL-SE company) and real ones (from Airbus Operations S.A.S.).

Keywords

line detection, feature extraction, particle filter, Hough transform, aircraft navigation

1. INTRODUCTION

Lane detection plays a significant role in driver assistance systems. We can note several use cases: axis keeping, vehicle position estimation, lane departure warning, road modelling, and automatic lane following. There are two main types of geometric methods commonly used for lane detection: feature-based and model-based methods. Another possibility is to use neural network based methods.

In feature-based algorithms, the most used feature is the contrast variation between marking/road or road/grass. This assumes that the contrast between the object and its environment is obvious. In the case of the contrast between the line marking and the road, the ground lines are designed to have more reflectivity than the road. That is why most of the studies ([14], [7], [2], [4]) try to detect line markings for lane detection applications, either by using color models, contrast or orientation of surfaces. The low computational complexity of those algorithms is an advantage but they are sensitive to shadows or occlusions. Model-based methods are more robust but require more assumptions on the road model (number of lines in the image, expected width and curvature, ...). Those algorithms are more consuming in terms of computational resources.

Neural network based methods have been developed for lane detection in the automotive field, based on machine learning and convolutive neural networks. In those studies, we can separate the networks in two specific fields: lane ([11]) and line ([6]) detections, where the first's objective is to define the usable area for the vehicle whereas the second's is specific to the detection of road markings. The neural network must learn how to extract important information from images in order to recognize lines. This method requires a larger dataset.

[10] reviews a large number of geometric line detection methods in the automotive fields and offers a good comparison in terms of advantages and drawbacks. Fewer studies concern the scene interpretation in aeronautics such as [15] or [1]. The distinction between the automotive and aeronautic fields is important because they differ in terms of line models and varying environments. Those specifications complicate the transfer of algorithms from one domain to the other.

One example is the region of interest (ROI) where lines must be detected in images. In the aeronautic field, images are provided by different cameras where the ROI is not always easy to define. Figure 1 presents the two cameras used for this study. The image from the fin camera is largely occupied by the sky and the plane. The area of detection is limited and far from the sensor (for a Airbus A380, which is our model for the simulation, lines to be detected are at least 70 meters from the camera). Hence, the number of pixels containing useful information is greatly reduced.

Another important difference is that, in the automotive field, the line detection is most of the time based on the assumption that the marking to be detected is white. This assumption enables simplifications such as using a grayscale or luminance image: it allows to focus on the edge detection. In the taxiway areas, the line to be detected is yellow so the line appearance is not as easy to exploit. As white lines also exist on the tarmac and should not be tracked, the

chrominance information is important. The method also needs to be robust to the noise (as the plane fuselage is visible in some images and could be painted in yellow).

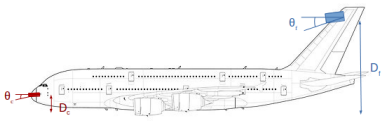


Figure 1: Camera positions in the aircraft fin and in the cockpit

This article aims to discuss line detection in the airport areas context. The discussion will focus on three main points: a comparison between three methods based respectively on the particle filter (PF), the Hough transform (HT) and the LaneNet neural network (LNN); the usefulness of the inverse perspective mapping (IPM) preprocessing and the performance of the three methods under degraded weather conditions. The algorithms will be compared on either simulated images or both simulated and real images.

This paper is organized as follows: in Section 2, we present first the IPM preprocessing, applied or not to every image before detecting lines, and then, the three selected methods. Section 3 details the criteria used for the comparison. The results are summed up and discussed in Section 4 before the conclusion in Section 5.

2. METHODS

Before performing one of the three methods to be compared, several preprocessing functions are automatically applied to our images such as distortion correction, white balance, calibration or homographic transformation. However, we decided to separate two cases of study; on one hand we use the resulting image as an input for the line detection algorithms, on the other hand, we apply an IPM transformation as another preprocessing.

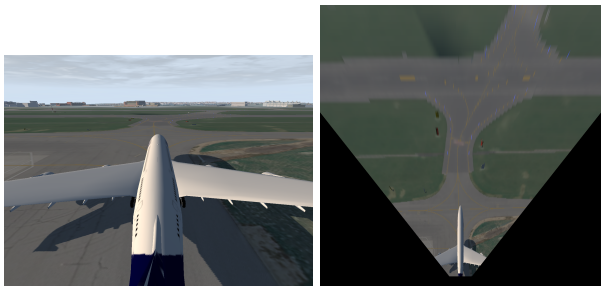
The two geometric methods use a color selection function for the line detection. When working on line detection, the reference color can easily vary from one location to another, or from the scene lighting. Some authors ([8], [4]) propose to work in other color spaces, such as HSV, Lab or YCbCr, that have the particularity to differentiate lightness information from chromatic information. Other authors ([12] and [5]) try to use a local adaptative threshold in order to be robust to shadowing or lightness variation. However, all of these methods still require to define the desired color which is chosen arbitrary.

We decided to implement these functions as parts of the methods instead of considering them as a preprocessing, because methods do not share the same color assumption. The particle filter method is based on [9] which includes the color assumption as part of the whole process, while the color assumption used for our implementation of the Hough Transform method is specific to our application and based on technical specifications defined in the xyz color space. The Hough transform takes as an input a binary image obtained from features extraction. Since we want to use the color as a feature, we have to define a threshold (i.e. a subset from the color space) to obtain the binary image. In contrast, the particle filter requires a weight map representing the probability of pixels to be on a line. The color extraction is done by computing the distance between the pixels' color and a reference color, different for the two methods.

The output of the three algorithms is a list of clusters. A cluster is a part of the image considered as a line (or a marking). It can be composed of all the consecutive pixels selected as part of the line (when using the particle filter or the LaneNet algorithm) or the two line extremities (when using the Hough transform).

2.1 Preprocessing: The IPM transform

We do not detail the IPM transform but it can be found in [9]. The result of the IPM transform can be observed in Figure 2. We decided to study the result of the three line detection algorithms with and without the IPM transform preprocessing because one of our assumption is that the IPM transform corrects the perspective effects on the lines which could simplify the detection. Moreover, the IPM transform creates additional synthetic information. It can be considered as a noise addition and degrading for the image. However, distant parts of a line are represented by few pixels in the raw image. The IPM transform will increase the number of pixels representing the line. Our second assumption is that this phenomenon, while adding noise to the image, should ameliorate the line detection on distant areas.



(a) (b)

Figure 2: (a) Original image, (b) IPM transform

2.2 First Method: Particle Filter

The first method we proposed for this comparison is an algorithm using the particle filter principle and presented in [9]. The philosophy of this method is to consider a pixel as a particle and a line of the image as an instant t. The particle filter is launched from the bottom line to the top line of the image. For each instant t, a number of particles are generated and a weight is assigned to them. The weight given by the weight map computed beforehand represents an observation whereas the pixel is a possible explanation of the hidden state of the system. The observation helps determine which pixel is prone to be a description of a line in the image. The weight map contains information from color and edges. A prediction of the particle positions on the next line (at time $t+1$) is made following a transition probability density to determine. In this implementation, the state is assumed to spread up vertically in the image but other cases can be thought upon such as in [16].

2.3 Second Method: Hough Transform

The Hough transform is based on the extraction of the color and edges information. The color extraction is dependent on

the definition of the reference color which is more difficult to express than the edges. In our approach, we proposed to use a more specific color definition.

In aeronautics, airport signs and markings are well defined by the ICAO recommendation [3]. From these recommendations, the color of airport lines are defined by a subset of the xyz color space. With that definition, it is possible to select a threshold to extract most of the lines from the image. However, images used for this comparison are defined in the sRGB space (a sub-space of the xyz color space). The yellow definition provided by ICAO is defined for the full xyz space. For our implementation we applied an empirical modification of this definition.

The first step of this method is to apply a chromatic thresholding using this yellow definition. A binary image is produced and the edge extraction method applied is the morphological gradient which is more adapted in this case than the Sobel or Canny filters usually found in the literature ([17], [13]). The line detection is performed by a Hough transform of the resulting image and the clustering of the results are provided by a hierarchical clustering algorithm such as presented in [2].

2.4 Third Method: Neural network approach

CNNs (Convolutional Neural Network) provide the best performances for computer vision applications including lane detection. They obtain the best results on most of the automotive benchmarks. However, the nature of airport areas is quite different from automotive ones and could degrade CNNs performances.

We chose to use the LaneNet CNN ([11]). It is a CNN architecture composed of three networks, one encoder and two decoders. The first decoder is used to provide a binary image containing pixels with potential line information. The second decoder is used to construct a pixel embeddings map that will be used to cluster pixels from the binary image in different lanes. Since our comparison is based on pixel comparison, (see Section 3), we only require the binary image.

Because of the low number of images in our dataset, we could not train this network again but test the already pretrained version of this network from TuSimple. This comparison is to argue about the possibility of using algorithms from the automotive field directly for aeronautics.

3. CRITERIA

In order to compare those three algorithms, we decided to base our analysis on multiple criteria that can be divided in categories, depending on the added knowledge value:

1. Based on detection range
   - **Maximum range of detection**
     One of the objectives in aeronautics is to have a wide detection field, because the aircrafts speed implies that decisions have to be taken as soon as possible. This value is given in pixels, computed on the overall image and represents the highest line in the image where pixels are detected.
2. Based on detection precision
   - **Mean of the clusters' pixels' distances to the ground truth**
     This criterion is the main one to quantify the algorithm precision. It computes, for each cluster, the distance between the proposed pixels to represent a marking and the ground truth.
   - **Recall**
     This criterion is used to analyze the pertinence of an algorithm, dividing the number of true positives by the total number of detections.

The aim of this article is to compare methods and analyze their relevance on real images. We are not concerned by the computation time because we are not working on the final implementation of the algorithm (in GPU or FPGA).

4. RESULTS AND DISCUSSION

Multiple datasets are available in the automotive field for the benchmark of algorithms (e.g. KITTI, Oxford Robotcar or TuSimple data sets). In aeronautics, images acquired by cameras embedded on aircrafts are more difficult to recover. Thanks to a joint project with AIRBUS, we can use the OKTAL-SE simulator to build synthetic images on airport areas. The three methods have been compared mostly on simulated images but also from few real images acquired by cameras placed at different spots in the aircraft, such as shown in Figure 1.

The aim of this comparison is to discuss three points. The first is to compare the results of the three methods on several images, varying the environment and image quality. The second point is to question the use of an IPM transform as a preprocessing and its contribution to the improvement of the algorithms performances. The third point of this study is to challenge the three algorithms by providing images with degraded meteorological conditions.

4.1 Performance of the three methods

Figure 3 presents the different precision criteria computed from the results obtained from both real and simulated images without IPM preprocessing nor degraded weather conditions. Figures 5(a) and 5(b) present the precision criteria with IPM processing and Figures 5(c) and 5(d) present the precision criteria with degraded weather conditions for images provided by the cockpit and fin cameras. Figure 4 presents the maximum range of detection for each use case.

The particle filter method recall is mostly higher than the recall of the other methods. This value should be compared with the mean distance between the detection and the ground truth. The particle filter has a higher mean distance because it returns a bigger set of points in each cluster. It increases the recall after the convergence of the clusters around the patterns to detect but also impact the precision. On contrary, the Hough Transform method returns few points, which are mostly well detected but a few false detections can lower significantly the recall of the method. LaneNet performances seem bad at first view. However, these performances have to be put in opposition with our annotation system. Because of its nature, the network tries to extrapolate more lines to describe a whole lane. Our aim is to detect only lines and the addition of lines from the LaneNet algorithm to recreate a lane strongly impacts the recall and precision.

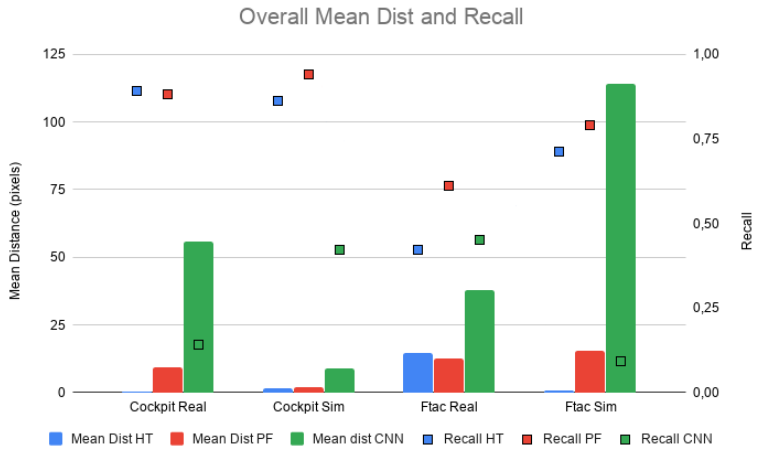


Figure 3: Mean distance to the ground truth and recall performances for real and simulated images from cockpit and fin cameras. Results in blue for Hough Transform, red for the Particle Filter and green for LaneNet.

4.2 Impact of the IPM transform on the performances

The IPM transformation requires a good estimation of the camera position on the aircraft, which we have not been provided with for the real images. We evaluate the interest of the IPM transform for synthetic images, acquired from simulated fin and cockpit cameras with a resolution of 1280pX960p and an horizontal field of view of 80 degrees. We also provide results for the real images, with an approximation of the position and field of view of the camera in real images but it should be noted that precision in this information can greatly impact the result of the IPM transform and the performances.

Figure 5 gives an overview of the impact of the IPM and the effect of degraded weather conditions on the results. The use of the IPM transform does not increase massively the recall results from the three methods. For LaneNet it considerably degrades the results, probably because of the nature of its training (TuSimple). It is predictable as the IPM transform adds noise to the image when trying to reconstruct an information not present in the initial image.

While this preprocess increases the detection range, it also decreases the precision and recall of each methods. It is due to the interpolations used to commute from the initial view to the bird's eye view. A pixel chosen in the IPM view is computed from two interpolations. When compared to the ground truth in the initial view, it is assumed that the precision of the detection will be impacted. The results of the IPM show that this preprocess can benefit the detection when the position of the camera is well known. However, while applied to the real images, with an approximated position of the camera, we noticed that the IPM transform is more likely to degrade the detection.

We observed an impact of the IPM on simulated images on the long range detection. Results from the top part of the images have been greatly improved. On contrary, the IPM impact is not relevant on this part of the images from the cockpit camera. Thoses pixels are far from the airplane and they are not distinguishable for the two cameras. The IPM transform adds noise, increasing the detection range for those pixels.

4.3 Impact of the degraded weather conditions

For the first two methods, we can observe a drop in maximum detection range. This was predictable as these methods strongly rely on color assumptions that are impacted by variations of the visibility in the environment. However, LaneNet does not seem to be strongly affected, in terms of results only. Qualitatively, the algorithm detects less lines and it virtually increases the precision (see Figure 6).

4.4 Qualitative comparison

Figure 6 is based on Figure 2(a) and offers a comparison of line detection of the same scene in two different weather conditions (either normal conditions or dusk and some fog). Figure 6(b) represents the detection for clear weather. We can see that the Particle Filter and Hough Transform methods detect lines homogeneously in the image. LaneNet encounters difficulties with this image as the pattern of the lines is specific to aeronautics and is not found in the automotive field. It performs at the top of the image where we can found two parallel lines, a mostly found configuration in automotive datasets. The detection in Figure 6(c) is sparse. The lines detections are reduced as the contrast between the line and the tarmac is more difficult to observe.

In Figure 7, we selected a fin camera image and a cockpit camera to present a qualitative comparison of the algorithm where the results of the particle filter are represented in red, the results of the Hough transform are represented in blue and the results of LaneNet are represented in green.

We observe that the three methods offer a satisfactory performance on real images. However, the particle filter is subject to false detection on far range pixels due to the color assumption. The Hough transform method does not perform as well as the other methods on far range detection for fin camera images. LaneNet recreate lines based on its training. It enables the algorithm to be resistant to occlusions but degrades virtually the precision.

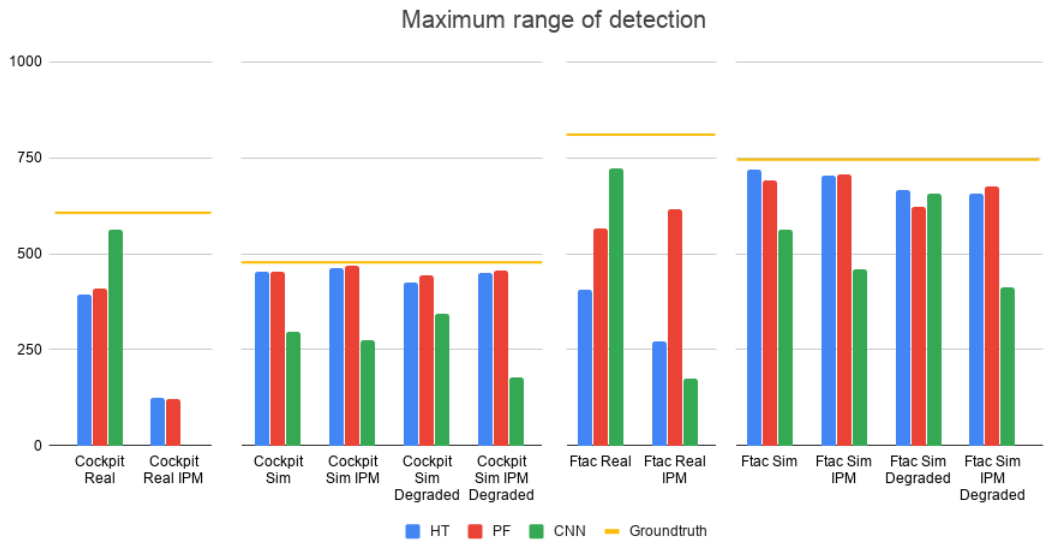


Figure 4: Maximum range detection for all situations (blue for Hough Transform, red for Particle Filter and green for LaneNet)

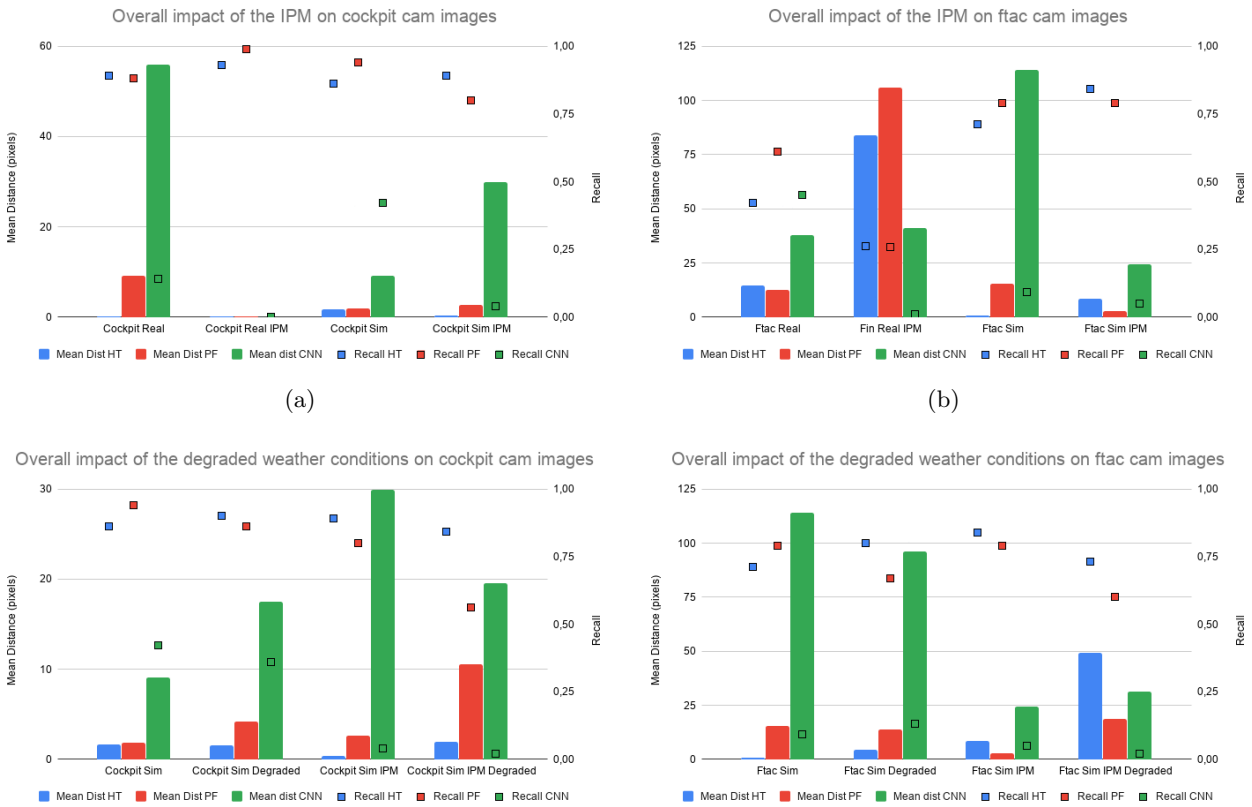
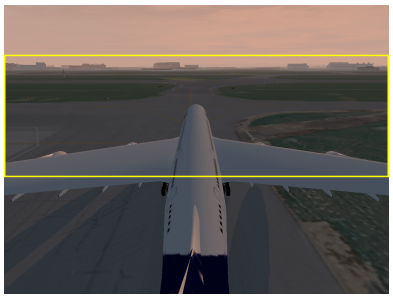
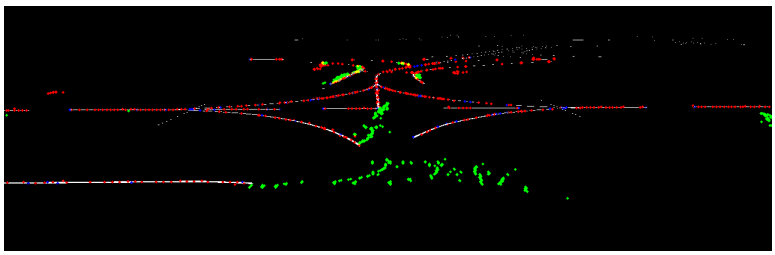


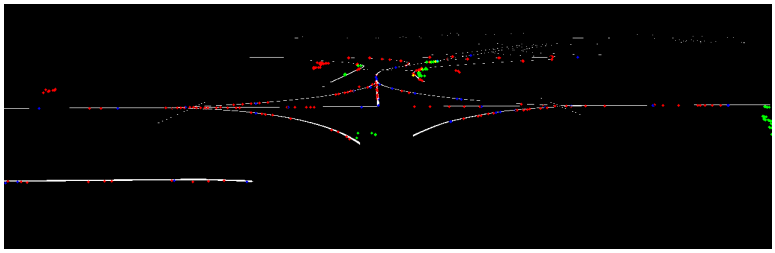
Figure 5: Impact of the IPM on the performances for (a) cockpit and (b) fin camera images. Impact of degraded weather conditions for (c) cockpit and (d) fin camera images.



(a)

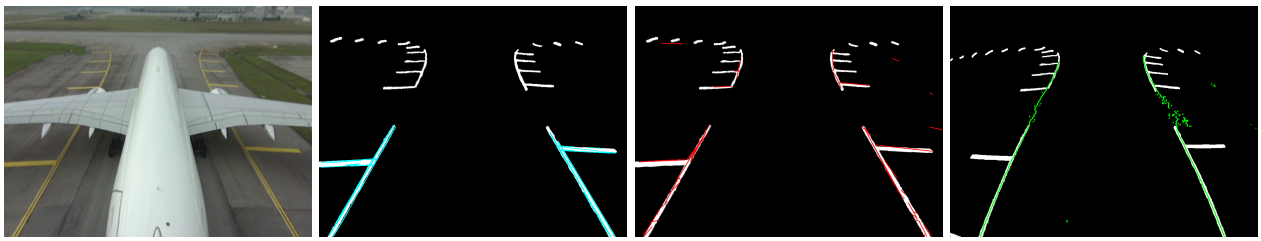


(b)



(c)

Figure 6: (a) Degraded conditions. Results of yellow area (blue for Hough Transform, red for the Particle Filter and green for LaneNet) for (b) clear weather and (c) degraded weather



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7: Real images from (a) a fin camera and (e) a cockpit camera. Ground truth and results (b)(f) for Hough Transform, (c)(g) for the Particle Filter and (d)(h) for LaneNet.

5. CONCLUSIONS

In this paper, we described three algorithms used in the literature for line detection. We compared those algorithms on simulated and real images, obtained by mounting cameras on several positions in an aircraft. We also studied the impact of performing an IPM transformation on the input image before running the algorithms and the impact of degraded weather conditions. The results presented above have been discussed based on average values of the detection on several images.

For the particle filter method, we observe that results on simulated images are satisfactory but this evaluation must be mitigated on real images. A number of thresholds have to be fixed in this method which implies difficulties to adapt for real images or degraded weather conditions. The Hough Transform method is based on a strong color assumption. It is easy to fix on simulation but harder to fix in real images, as many other computer vision applications. LaneNet provides unsatisfactory results as expected. The change of field from automotive to aeronautics prevents the network from detecting a good number of lines because of the differences in line patterns. The IPM Transform requires a precise setup. It can improve detection range in simulated images but minor changes in parameters on real images can heavily degrade the results. Lastly, the methods still give good results at a near range but the degraded weather conditions have a great impact on middle and far range detections.

In future studies, we will work on the color assumption. Combining ICAO information and a method to perform an automatic detection of a reference color in the image should improve results on real images. We also plan to implement a new algorithm for the particle filter based on an idea presented in [16] where the particle can be expressed as a window of pixels instead of a simple pixel. Combined with a better appreciation of the line color, it should reduce the false detections and improve the precision. We plan to increase the number of test images as it will be beneficial for further CNN-based studies. It will enable us to retrain the LaneNet network to adapt to the aeronautic context. We are currently working on an estimation of the angular variation of the aircraft position in regards to the centerline of the tarmac. It can be added as another criterion to select the preferred line detection algorithm and it could be used to perform a lane departure estimation algorithm.

6. ACKNOWLEDGEMENTS

We would like to thank the OKTAL-SE company for providing the simulation tool, used to build synthetic images. C. Meymandi Nejad is supported by the project COCOTIER funded by the Direction Générale de l'Aviation Civile depending on the French government and E. Perrotin is supported by a grant from AIRBUS OPERATIONS. We would also like to thank the AIRBUS OPERATIONS company for providing real images from test planes.

7. REFERENCES

[1] T. Beda, O. Koukol, and O. Pokorny. Methods and systems for providing taxiway stop bar information to an aircrew, 2017.

[2] R. N. Hota, S. Syed, S. Bandyopadhyay, and P. R. Krishna. A simple and efficient lane detection using clustering and weighted regression. In *COMAD*, 2009.

[3] ICAO. *AERODROMES: aerodromes design and operations*. 2018.

[4] M. Kazemi and Y. Baleghi. L*a*b* color model based road lane detection in autonomous vehicles. 2017.

[5] A. Kheyrollahi and T. P. Breckon. Automatic real-time road marking recognition using a feature driven approach. *Machine Vision and Applications*, 2010.

[6] X. Li, J. Li, X. Hu, and J. Yang. Line-cnn: End-to-end traffic line detection with line proposal unit. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[7] C. Lipski, B. Scholz, K. Berger, C. Linz, T. Stich, and M. Magnor. A fast and robust approach to lane marking detection and lane tracking. *IEEE Southwest Symposium on Image Analysis and Interpretation*, 2008.

[8] A. Mammeri, A. Boukerche, and Z. Tang. A real-time lane marking localization, tracking and communication system. *Computer Communications*, 2016.

[9] C. Meymandi-Nejad, S. E. Kaddaoui, M. Devy, and A. Herbulot. Lane detection and scene interpretation by particle filter in airport areas. In *International Conference on Computer Vision Theory and Applications*, 2019.

[10] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane. A review of recent advances in lane detection and departure warning system. *Pattern Recognition*, 2017.

[11] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool. Towards end-to-end lane detection: an instance segmentation approach. *IEEE Intelligent Vehicles Symposium*, 2018.

[12] A. Saudi, J. Teo, Mohd Hanafi Ahmad Hijazi, and J. Sulaiman. Fast lane detection with Randomized Hough Transform. In *International Symposium on Information Technology*. IEEE, 2008.

[13] S. Srivastava, M. Lumb, and R. Singal. Improved lane detection using hybrid median filter and modified hough transform. *International Journal of Advanced Research inComputer Science and Software Engineering*, 2014.

[14] T.-Y. Sun, S.-J. Tsai, and V. Chan. Hsi color model based lane-marking detection. *IEEE Intelligent Transportation Systems Conference*, 2006.

[15] K. Theuma and D. Z. Mangion. An image processing algorithm for ground navigation of aircraft. In *Advances in Aerospace Guidance, Navigation and Control*, 2015.

[16] A. Tran and A. Manzanera. A versatile object tracking algorithm combining Particle Filter and Generalised Hough Transform. In *International Conference on Image Processing Theory, Tools and Applications*. IEEE, 2015.

[17] H. Xu and H. Li. Study on a robust approach of lane departure warning algorithm. *International Conference on Signal Processing Systems*, 2010.