DOI 10.5162/ettc2020/5.2

Displays Validation Based on Image Processing

Israel López Herreros
Airbus Defence and Space, John Lennon S/N 28906 Getafe-Madrid, Spain
israel.lopez@airbus.com

Abstract:

Aircraft manufacturers integrate a lot of equipment from different suppliers and on some occasion this equipment are like black boxes, the only easy way to check the data is through special monitoring displays supplied with the equipment. On other occasions a real aircraft display needs to be validated. These displays can be very complex, big amount of symbols and very dynamics. The validation of these displays can be a very tedious work if it is done by hand.

This document presents a methodology to automate the validation of the aircraft displays based on image processing to extract numbers, texts and symbols. These displays are not completely standards in the market and they have not been created taking into account these image processing techniques, so it is almost impossible to develop a generic tool. A specific tool will be developed for each display but they will share common stages from the specific pipeline.

There is a lot of literature about image processing but what it is really important is the definition of the pipeline in order to have a robust and efficient data acquisition system based on image processing. This document will give an overview of the different stages and some pipelines applied in real use cases in Flight Test Airbus Defence and Space.

Key words: Image processing, computer vision, displays validation.

1. Acronyms, Abbreviations and Symbols

AC:	Aircraft
CSV:	Comma-Separated Values

2. Introduction

Aircraft manufacturers integrate a lot of equipment from different suppliers and on some occasion this equipment are like black boxes, the only easy way to check the data is through special monitoring displays supplied with the equipment. On other occasions a real aircraft display needs to be validated. These displays can be very complex, big amount of symbols and very dynamics. The validation of these displays can be a very tedious work if it is done by hand.

This document presents a methodology to automate the validation of the aircraft displays based on image processing to extract numbers, texts and symbols (see Fig. 1).

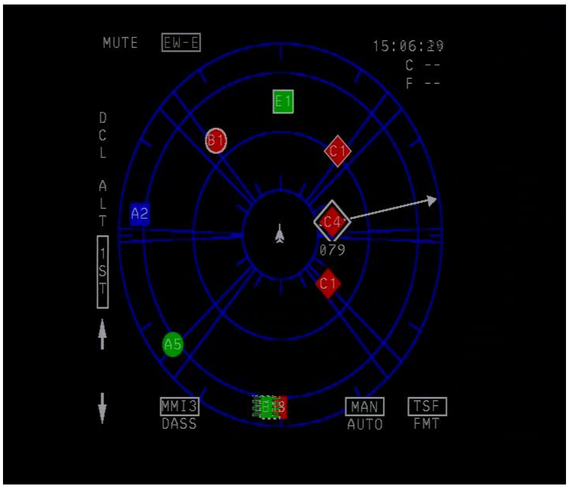

Fig. 1. Image of the aircraft display used for the prototype.

3. The Challenge

The methodology presented in this document will demonstrate the capacity to extract the screen position, color, text and shape of the symbols presented in a real AC display.

This display is not completely standard and it has not been created taking into account the image processing techniques applied at a later phase.

Relative to this data acquisition phase several difficulties have been found and they will be described below.

The symbols have not a unique identifier. It makes it difficult to track two symbols with the same identifier in a similar screen position or when the symbols disappear for a while and they appear again in different positions (see Fig. 2)

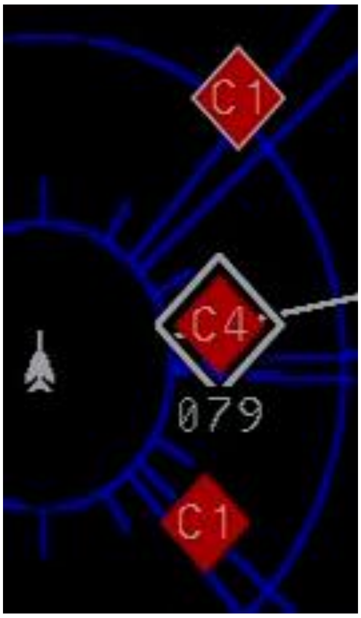

Fig. 2. Non-unique identifier.

There are rendering problems in some frames. The origin of this problem is unknown but some hypothesis could be related to the graphics hardware or the codecs used for the recording video (see Fig. 3).

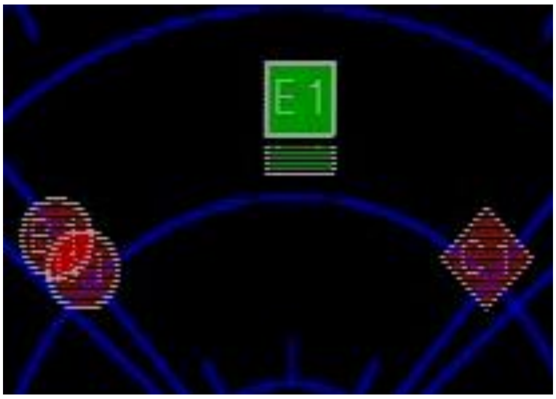

Fig. 3. Image with rendering problems.

There are different types of symbols overlapping that difficult the symbol recognition: Arrows that cross a symbol and several symbols overlapped (see Fig 4).

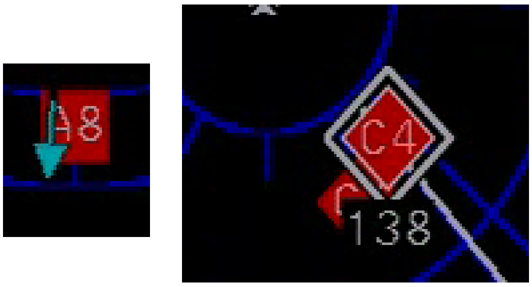

Fig. 4. Image with overlapping issues.

The blue color is reused for static background and some symbols. The colors in general are not completely pure, they have an interval of colors, i.e. when the blue symbols are analyzed different blue colors are found. These issues make the color recognition phase difficult (see Fig. 5).

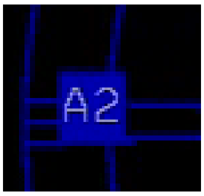

Fig. 5. Blue color is reused in the display.

The objective is to minimize false positive with the introduction of high restrictions to avoid recognize positive symbols when they are not true, e.g. recognize C1 when it is not really C1.

Other objective is to reduce false negatives using the accuracy level, e.g. if a false negative symbol as BB is recognized it may be rejected because its accuracy is low.

4. Prototype

To demonstrated the viability of this project a low cost prototype to extract data using image processing techniques from videos has been developed.

Searching in Google it can be seen that there are a lot of strategies to recognize text and numbers. Different algorithm evaluations based on Neural Network as the Tesseract [1] ocr library has been tested but all of them were very general and the accuracy wasn't enough for the requirements.

The philosophy applied in this prototype is to develop a specific tool, versatile, with well differentiated phases in order to have a general strategy as much as possible but without compromising the accuracy.

OpenCV [2] has been selected as the computer vision library. The library is cross-platform and free for use under the open-source BSD license.

Python [3] has been chosen for rapid prototyping, departmental experience and for its good integration with OpenCV.

The pipeline chosen to solve the data extraction is described below:

First phase: Segmentation by color. The original image and the chosen color are the inputs of this phase. Due to the fact that the colors are not pure an interval has to be taking into account. The output of this phase is the image filtered by the chosen color (see Fig. 6).

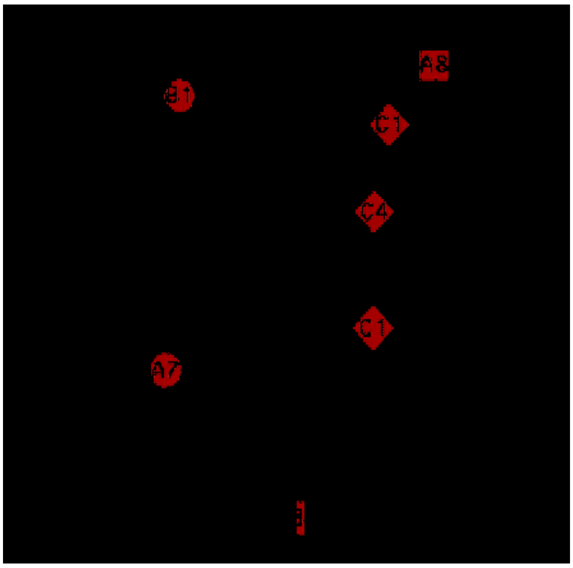

Fig. 6. Filtered image using red colors.

Second phase: Thresholding. The filtered image and the original image are the inputs of this phase. Two binary images [4] are generated, one for contours recognition from the filtered image by color (see Fig. 7.). This binary image is enhanced with a morphology close step [5] (see Fig. 8).

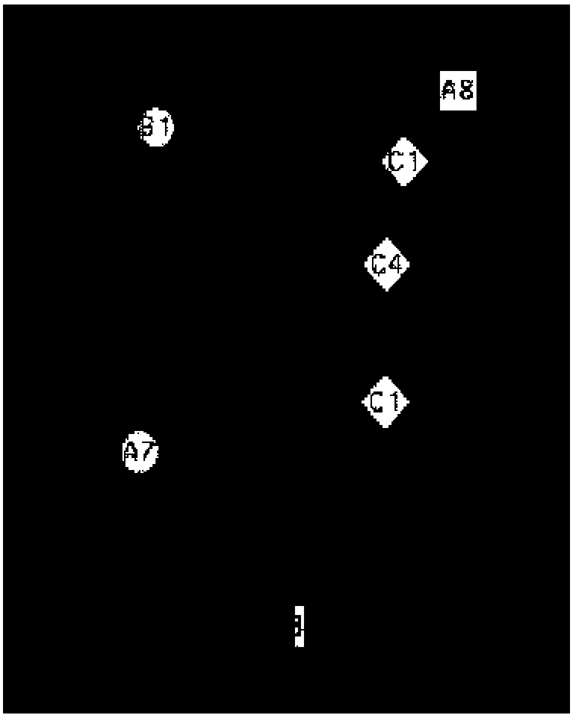

Fig. 7. Binary image for contours recognition.

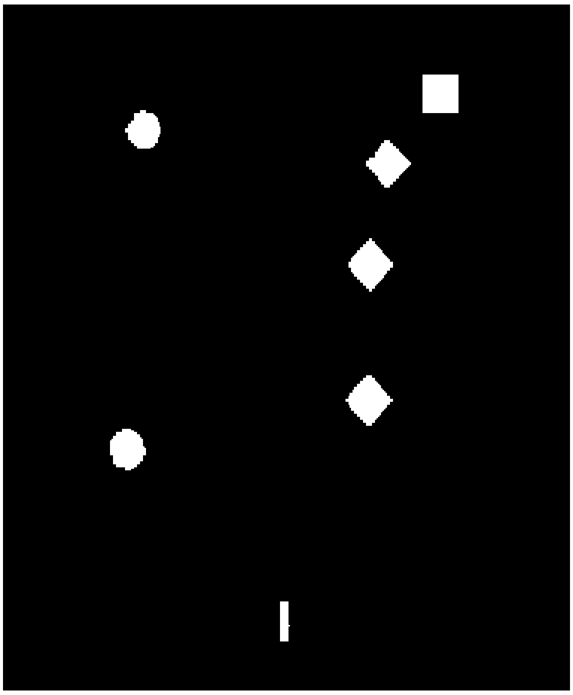

Fig. 8. Morphology close image for contours recognition.

A second binary image is generated from the original image for text and number recognition (see Fig. 9).

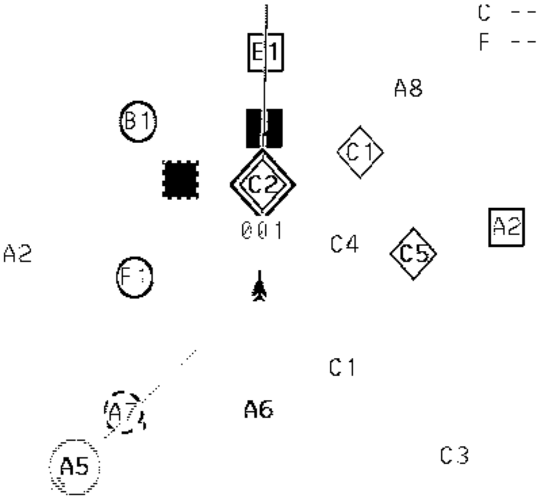

Fig. 9. Binary image for text/number recognition.

Third phase: Contours recognition. The morphology close image is the input of this phase. The contours found are filtered by the area calculated from the moments [6] of these contours. The contours are smoothed out with the convex hull [7] (see Fig. 10).

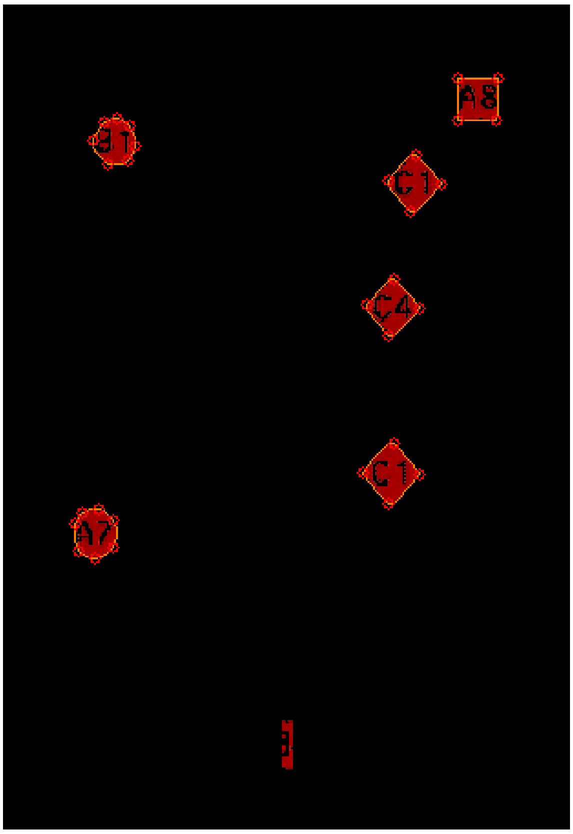

Fig. 10. Image with recognized contours.

Fourth phase: Shape recognition. The contours are the inputs of this phase. These contours are analyzed using geometry properties to identify if the contours correspond to circles, squares or diamonds. The outputs of this phase are the different shapes with the corresponding bounding boxes (see Fig. 11).

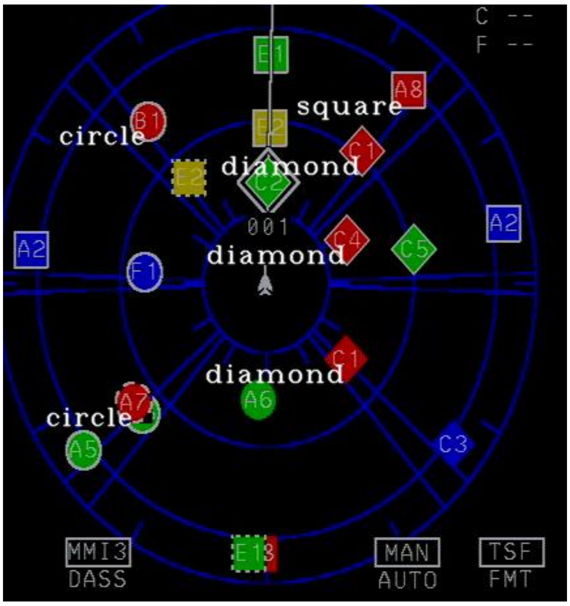

Fig. 11. Image with recognized shapes.

Fifth phase: Text/number recognition. The binary image (see Fig.9) and the bounding boxes are the inputs of this phase. A training dataset of all characters from the binary images is used in this phase (see Fig. 12).

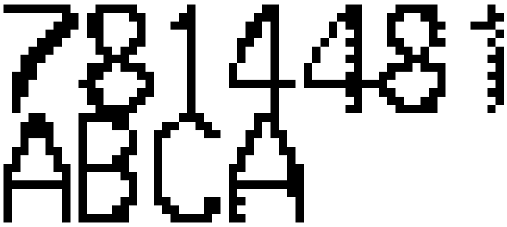

Fig. 12. Example of training dataset.

Each digit from the boundary box is isolated based on geometry properties. For each digit a correlation algorithm is used against the dataset to find the characters that match with best accuracy. The outputs of this phase are the different recognized texts with their corresponding accuracies (see Fig. 13).

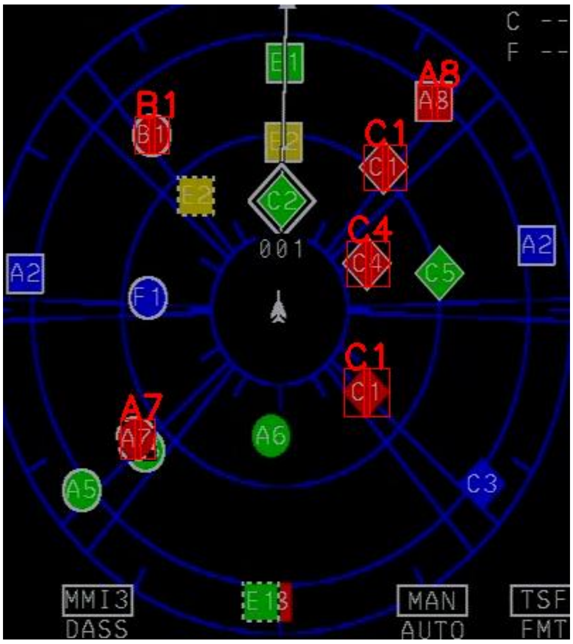

Fig. 13. Recognized texts in the image.

Sixth phase: Update symbol information in the corresponding series. The actual and the historical information are used to better identify the right series to add the new symbol information. The non-unique identifier makes this operation more difficult and some assumptions have to be taken. In our case a maximum displacement based on system information of 25 pixels is used to consider that the symbol belongs to the series. If the displacement is bigger than 25 pixels or it disappears during one frame a new series is created with this symbol information.

5. Results

For each color and text symbol a csv file is generated with the following time series information: id image, accuracy, x screen position, y screen position, shape and color.

A time series is drawn with each component position. Each color represents a different series due to the restrictions explained in chapter 4, sixth phase (see Fig. 14).

Reviewing the time series is easy to detect system malfunctions. A specific event detector could be implemented to check these kinds of malfunctions.

A time series drawing can show different symbols at the same time due to the non-unique identifier (see. Fig. 15).

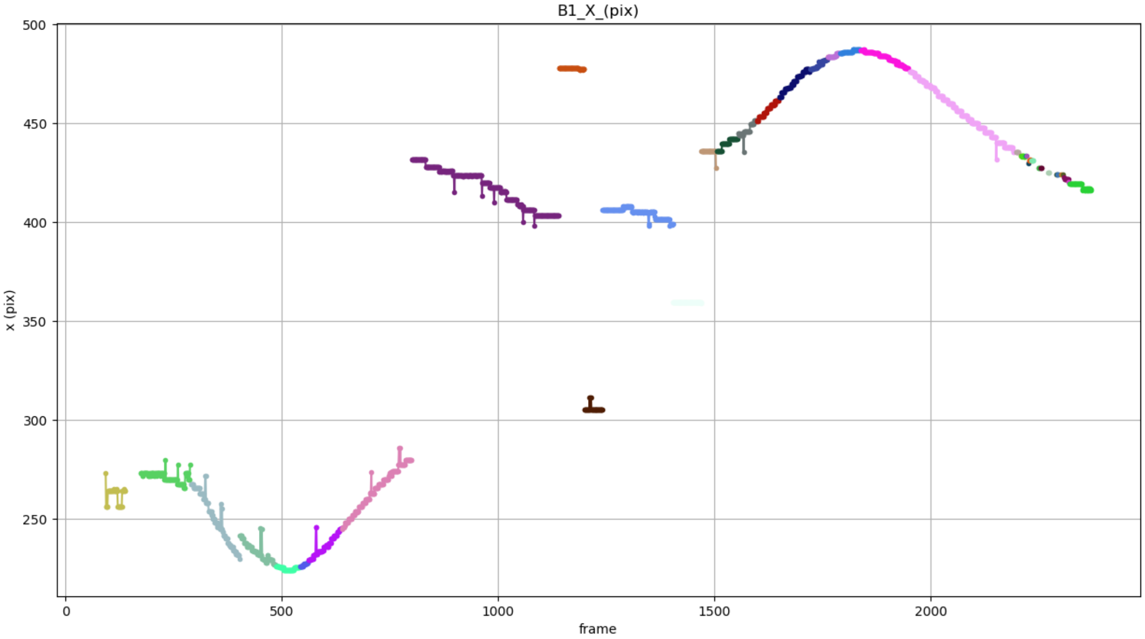

Fig. 14. Red B1 symbol x screen position.

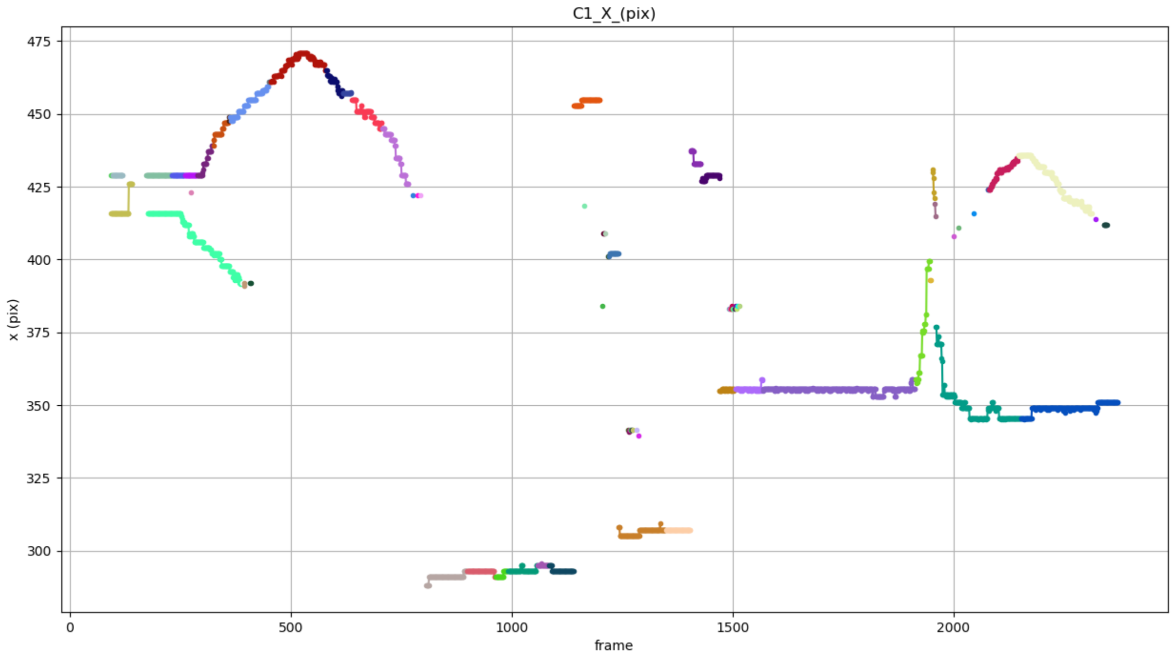

Fig. 15. Red C1 symbol x screen position.

A time series with the symbol accuracy is drawn. Each color represents a different series due to the restrictions explained in chapter 4, sixth phase (see Fig. 16). This data is very interesting to filter symbol data with low accuracy. False positive symbols identifications will disappear in this way, i.e. below 0.7 B1 symbols will be filtered.

A final figure with the overall symbol accuracy is drawn (see Fig. 17). This boxplot is a method for graphically drawing groups of numerical data through their quartiles.

The accuracy by symbol is represented. It is a very useful figure to improve the accuracy data of the overall recognition. In our video example the only true symbols are B1, A7, C1, A8 and C4, so if the data is filtered by an accuracy level of 0.6 all false positives and false negatives are rejected.

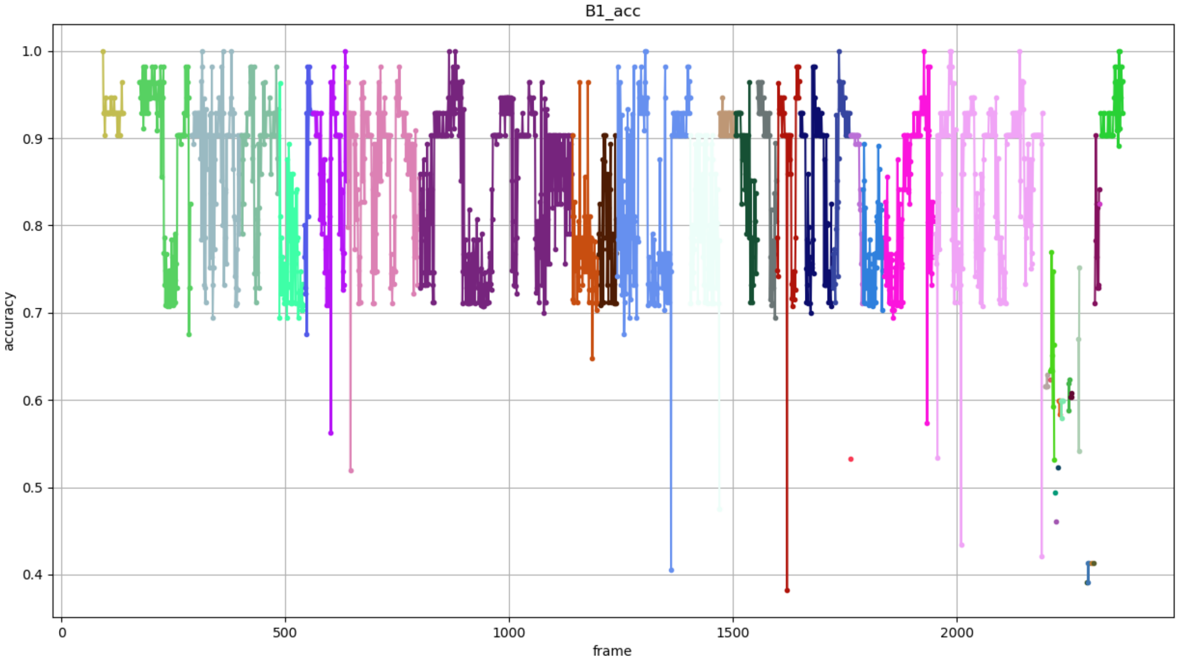

Fig. 16. Red B1 symbol accuracy.

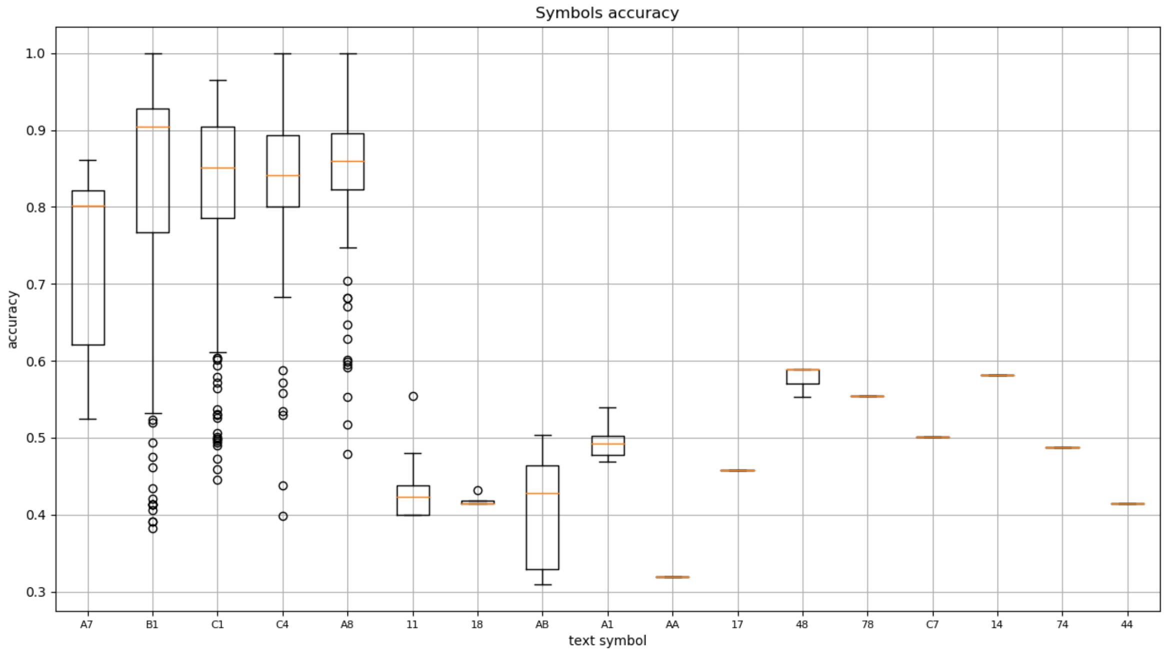

Fig. 17. Overall symbol accuracy.

6. Recommendations

In order to facilitate these kinds of displays validations several recommendations could be taking into account.

Visual data acquisition is a complex problem, therefore it is important to have several testing videos to increase robustness

It could be very useful to design displays trying to reduce all the challenges seen in chapter 3.

7. Conclusions

Aircraft manufacturers integrate a lot of equipment from different suppliers and on some occasion this equipment are like black boxes, the only easy way to check the data is through special monitoring displays supplied with the equipment. On other occasions a real aircraft display needs to be validated. These displays can be very complex, big amount of symbols and very dynamics. The validation of these displays can be a very tedious work if it is done by hand.

The methodology presented in this document demonstrates the capacity to extract the screen position, color, text and shape of the symbols presented in a real AC display with great accuracy. This pipeline is in some way specific to the display but it shares common phases to other AC displays.

This methodology provides very useful information in order to validate the system behind the display or to validate the display itself.

It can be concluded that displays validation based on image processing is a valid methodology.

References

[1] Tesseract

https://github.com/tesseract-ocr/tesseract

[2] OpenCV

https://en.wikipedia.org/wiki/OpenCV

[3] Python

https://en.wikipedia.org/wiki/Python_(programming_language)

[4] Thresholding

https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html

[5] Morphological Transformations

https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html

[6] Image Moments

https://docs.opencv.org/3.4/d0/d49/tutorial_moments.html

[7] Convex Hull

https://docs.opencv.org/3.4/d7/d1d/tutorial_hull.html