THE UNIVERSITY OF ARIZONA
University Libraries
UA CAMPUS REPOSITORY

Using Image Processing and Pattern Recognition in Images from Head-Up Display

Item type	text; Proceedings
Authors	Guarino de Vasconcelos, Luiz Eduardo; Kusomoto, André Yoshimi; Leite, Nelson Paiva Oliveira
Publisher	International Foundation for Telemetering
Journal	International Telemetering Conference Proceedings
Rights	Copyright © held by the author; distribution rights International Foundation for Telemetering
Downloaded	8-Mar-2018 08:58:09
Link to item	http://hdl.handle.net/10150/579665

Using Image Processing and Pattern Recognition in Images From Head-Up Display

Msc. Luiz Eduardo Guarino de Vasconcelos;
Eng. André Yoshimi Kusumoto;
Dr. Nelson Paiva Oliveira Leite.
DCTA - IPEV
Instituto de Pesquisa e Ensaios em Voo
Pça Marechal Eduardo Gomes nº50 (CTA-IPEV)
São José dos Campos, SP, BRAZIL, 12.228-901
cti-i@ipev.cta.br, cti-p@ipev.cta.br, epd@ipev.cta.br.

ABSTRACT

Images frames have always been used as information source for the Flight Test Campaigns (FTC). During the flight tests, the images displayed on the Head-Up Display (HUD) could be stored for later analysis. HUD images presents aircraft data provided by its avionics system. For a simplified Flight Test Instrumentation (FTI), where data accuracy is not a big issue, HUD images could become the primary information source. However in this case data analysis is executed manually, frame by frame for information extraction (e.g. Aircraft position parameters: Latitude; Longitude and Altitude). In approximately one hour of flight test about 36,000 frames are generated using standard-definition television format, therefore data extraction becomes complex, time consuming and prone to failures. To improve efficiency and effectiveness for this FTC, the Instituto de Pesquisas e Ensaios em Voo (IPEV - Flight Test and Research Institute) with Instituto Tecnológico de Aeronáutica (ITA - Aeronautical Technology Institute) developed an image processing application with pattern recognition using the correlation process to extract information from different positions on the images of the HUD. Preliminary test and evaluation carried out by 2012 using HUD images of the jet fighter EMBRAER A1. The test results demonstrate satisfactory performance for this tool.

KEY WORDS

Flight Test, Head-Up Display, Image Processing, Pattern Recognition

1. INTRODUCTION

The execution of experimental Flight Test Campaign (FTC) provides all information required for the aircraft and/or its systems development, operation, certification and qualification. Typically the execution of a FTC encompasses three major systems as follows: Flight Test Instrumentation System (FTI) that is basically a measurement system; Real-Time Telemetry Link (RTL); and Ground Telemetry System (GTS). Images frames have always been used as information source for several FTC (e.g. external store separation).

For tactical operations the Head-Up Display (HUD) is a key element to provide essential information for the pilot during the navigation and weapon enforcement flight regimen. This system is composed by an Electronic Unit (EU) and a pilot display unit (PDU) where the computed information is displayed. The HUD main role is to minimize the pilot operational workload.

This system projects navigation and weapon information in a user-defined symbolic form. Various FTC (e.g. weapon aiming) requires the analysis of HUD images. Therefore a simple non-intrusive FTI data gathering technique for such information is to record the HUD video images frames and to embed such information into the FTI measurement set for post mission data analysis. The major constraint of this architecture is the fact that data analysis is typically qualitative, where image frames are manually observed and analysed for information extraction. This process is very ineffective because in one flight test hour it 36,000 frames are generated at a standard-definition television format (i.e. 704 x 480 pixels of resolution x 30 frames/s). As result the data analysis process becomes complex, time consuming and suitable to failures.

To improve efficiency and effectiveness for this application, the Instituto de Pesquisas e Ensaios em Voo (IPEV - Flight Test and Research Institute) with Instituto Tecnológico de Aeronáutica (ITA - Aeronautical Technology Institute) developed an image processing application with pattern recognition using the correlation process to extract information from different positions on the images of the HUD.

The remainder of this paper is organized as follows. Section 2 describes image recognition overview. Section 3 describes in detail the architecture of the proposed system. Experimental results are reported in Section 4. Conclusions are drawn in the end of paper.

2. IMAGE RECOGNITION OVERVIEW

Head-Up Display images contain important semantic information such as feet, altitude, elevation, azimuth, time, etc. While for printed documents, optical character recognition (OCR) systems have already reached high recognition rates, and are widely commercialized, recognition of characters in others context is still the subject of active research. In fact, this task is a challenging problem because of low resolution, complex background, non-uniform lightning or blurring effects.

Most of the state of the art text image recognition methods are based on template matching which is also the case of most available commercial OCR systems mainly designed for printed text. This is an old principle that was proposed for OCR in 1929 by Tausheck. It reflects the technology at that time, which used optical and mechanical template matching. Light passed through mechanical masks is captured by a photo-detector and is scanned mechanically.

When an exact match occurs, light fails to reach the detector and so the machine recognizes the characters printed on the paper. Nowadays, the idea of template matching is still used but with more sophisticated techniques. A database of models is created and matching is performed based on a distance measure. The models are generally composed of specific features that depend on the properties of the pattern to be recognized.

Chen at al [1] proposed a method based on character side profiles, in videos. First, a database is constructed with left, right, top and bottom side-profiles of sample characters. Then the candidate characters are recognized by matching their side-profiles against the database. This method requires of course a ‘cleaned’ binary text image.

Therefore, they apply various pre-processing techniques before the recognition step, namely: shot boundary detection, edge-based text segmentation, multiple frame integration, gray-scale filtering, entropy-based thresholding, and noise removal using Line Adjacency Graphs (LAGs).

Another template matching method was proposed by Kopf et al. [2]. They have chosen to analyse the contour of a character and derive features extracted from the Curvature Scale Space (CSS). This technique which is based on the idea of curve evolution requires also binary text images.

Yokobayashi et al [3, 4] proposed two systems for character recognition in natural scene images. Both of them rely on two steps: the first one is the binarization step and the second one is the recognition step based on an improved version of Global Affine Transformation (GAT) technique for grayscale character images.

Once a binary image is obtained, an improved GAT correlation method [5] is applied for recognition. This is a matching measure between the binary character image and a template image. As templates, the authors use a single font set of binary images of alphabets and numerals, which explains the need for the previously mentioned binarization step.

In [4], the authors proposed a binarization method based on three steps, as follows:

1. Colour vectors of all pixels in an input image are projected onto different arbitrarily chosen axis;
2. The maximum between-class are separately computed by setting an optimal threshold according to the Otsu's binarization technique [6];
3. The axis that gives the largest between-class separability and the corresponding threshold for binarization of the input image is selected.

Then, the method decide which class corresponds to characters or background according to the ratio of black pixels to white ones on the border of the binarized image. As in their previous work [3], once the binary image is obtained, an improved GAT correlation method is applied for recognition.

It should be noticed that in all the works mentioned above, there is a need for a pre-processing steps (i.e. binarization) and for finding optimal tuneable parameters.

Agility in the acquisition, processing and transmitting information, it becomes a powerful tool that enables quick decision-making. It is no longer desirable but becomes essential in critical systems. [7].

Therefore we propose the use of an automatic recognition scheme for HUD images, applying a pre-processing like binarization, using tuneable parameters for each type of background image, a database to characters and a simple correlation processes for matching de correct character.

3. APPLICATION ARCHITECTURE

For the development of this application a HUD embedded in EMBRAER A1 aircraft is used. This HUD recorded into the camera's buffer all images of the test flight. Then recorded video is downloaded to the computer through its USB (Universal Serial Bus) interface. Therefore the gathered information would be processed by the developed application.

The development of this tool required customized solutions to overcome the following main challenges:

- The scenario could change very rapidly which can result in significant change of lighting (Figure 01).
- The images have different patterns of texts position.
- Process images to obtain better accuracy even in frames not cleaned (e.g. with blurring, overlapping objects).
- Transitions between images produced by the HUD generate blurring areas.

The details of algorithms for image processing and obtaining results are discussed below.

3.1. APPLICATION ALGORITHM

Although there are different patterns of images produced by HUD, all images have text in preset positions. Thus, the adopted strategy was to identify these positions so that we could make cutouts in image. In Figure 02, we can observe a pattern image with texts in pre-defined positions (eg azimuth, elevation, distance).

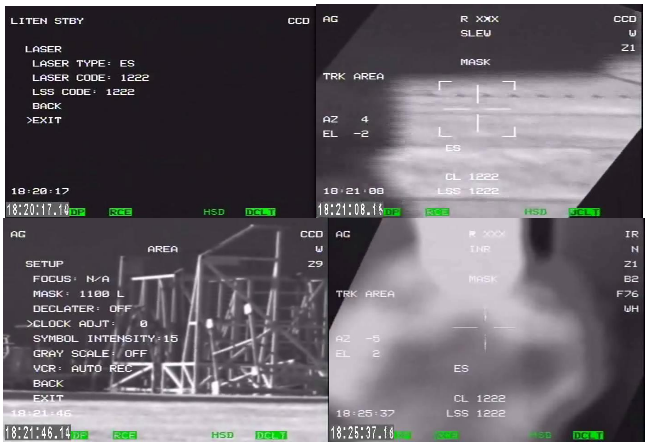


Figure 01 - Different scenarios with varying brightness and background

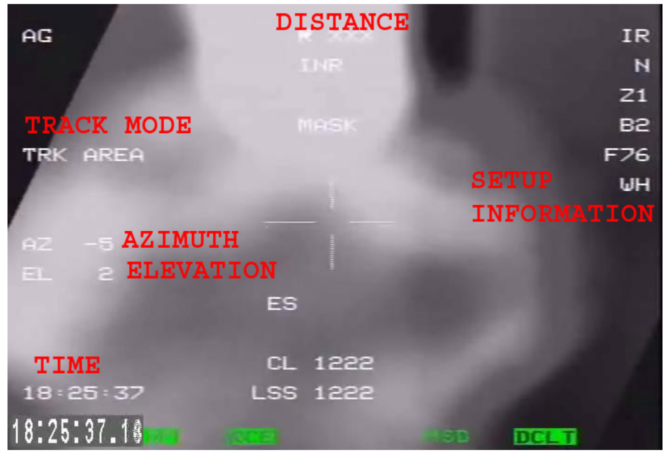


Figure 02 - Image with texts in defined positions

For the different image patterns, it was realized that there are two types of fonts printed in the images. One type is used by most of the information (e.g. azimuth, elevation). The second type of font is used to display the clock in the lower left corner of the image (Figure 02). After identifying these sources, a database of templates was created for each character (Figure 03). This database is used in processing each image. Each image can be represented by a two-dimensional array, therefore an algorithm that computes the correlation coefficient between two matrices was designed. This coefficient yields results between zero and one. Values close to one indicate that the matrices are similar. Values near zero indicate that the matrices are different.

The application was implemented using a known image processing techniques [8] that uses the following steps:

1. The video produced by HUD is converted to JPEG images;
2. The template for each character is loaded from the database;
3. The HUD images are loaded in the application;
4. Each image is processed using the following steps:
   a. Binarization [9];
   b. Adjustments are applied to the image depending on the gray level, background and histogram of the original image;
   c. Extraction of the preset positions of the image (text) using crop;
   d. Extraction of images that represent the characters in each preset position (Remove pepper noises using mathematical morphology [10]. Detection of image regions using properties of the image. At the end, the image is cropped);

   e. Applying the correlation algorithm (i.e. “***r***” in eq. 01) between the template and the extracted image (character). If correlation coefficient is greater than 0.5 (50% of correlation), the character is recognized. Otherwise, the image is discarded; and
   f. The results are stored in an array structured (Figure 04) with all the image information.
5. At the end of the process, it's possible to view the results of the array.

Figure 03 – Partial database of templates for each character

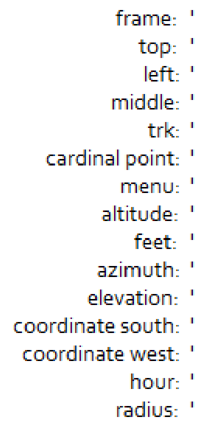


Figure 04 – Example of array structured with image information after processing

After binarization, each image is an array with values ranging between zero and one. Figure 05 depicts an example of image representation in matrix format. For this application, the template used for each character has 24 x 42 pixels. This template has the character size extracted from the original image displayed at the HUD.

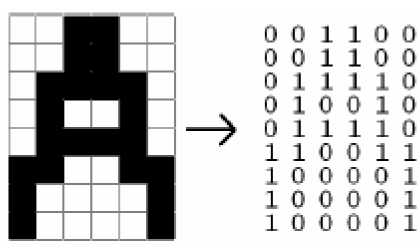


Figure 05 – Example of character ‘A’ represented by matrix format.

The sequence of Figure 06 shows the character “***S***” in different formats. In addition to have significant differences in pixels, the character "***S***" may be confused with the number five. To solve this problem, we applied a learning process based on context. For example, the character “***S***” should not be identified in areas which may appear only numbers. Other alike restrictions were implemented (e.g. character “***Z***” and the number seven character “***O***” and number zero).

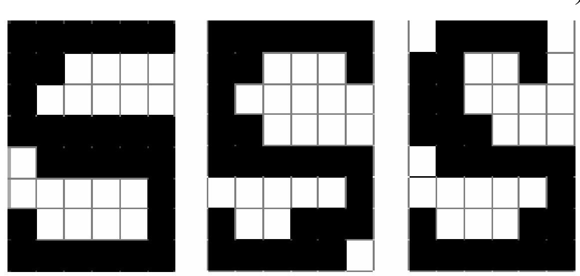


Figure 06 – Examples of caracter “***S***” in different images

The determination of the correlation coefficient ("r") is obtained by:

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}} \qquad \textbf{eq. 01}$$

Where:

- ***r*** is the correlation coefficient between zero and one.
- ***A*** is the matrix (character) of the database
- ***B*** is the matrix (character) to be matched
- $\bar{A}$ is average of matrix elements in A
- $\bar{B}$ is average of matrix elements in B

4. APPLICATION EVALUATION

Preliminary test and evaluation of this application was executed with HUD images produced by EMBRAER A1 (i.e. AM-X) aircraft, during the Brazilian Flight Test Course (CEV) carried out by the 2012 class students. In addition, the application was evaluated with more than 1,000 frames and more than 40,000 characters in the frames.

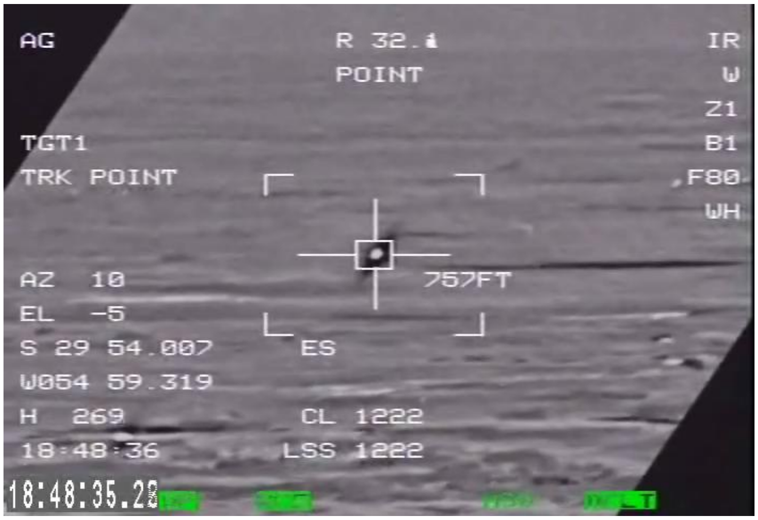
The main application was developed under MatLab® environment and tested with Intel®Pentium IV Core™ 2 Duo CPU T5800 2.00 GHz notebook with Ethernet Gigabit Local Area Network (LAN) interface, 4 Gb RAM and Microsoft Windows 7 Professional.

Initially the algorithm for recognition characters was setup to discard the image if the correlation coefficient ("*r*") is less than 0.5. After evaluation, it was realized that the algorithm can validate 80.4% of characters in the frames. This result occurs because some characters cannot be recognized in some transitions of frames produced by HUD. Figure 07 depicts such problem that is highlighted in red rectangles. In this case, the coordinate south and altitude (H) parameters are discarded. The others information in images are processed normally.

Figure 07 – Transitions of frames

The smaller the area of the image to be processed, the faster is processing. Thus, small templates were produced to the characters and used as a database. However the results of applying such simplification resulted in unacceptable performance. Therefore, the optimum performance was achieved with 24 x 42 pixels templates.

In Figure 08 depicts an example of a single frame image processed by the application and the results. The usage of this application resulted in satisfactory performance.



frame:	'36996.jpg'
top:	'point'
left:	'ag'
middle:	''
trk:	'trkpoint'
cardinal point:	'w'
menu:	''
altitude:	'269'
feet:	'757'
azimuth:	'10'
elevation:	'-5'
coordinate south:	'29.54.007'
coordinate west:	'054.59.319'
hour:	'18:48:36'
radius:	'32.1'

Figure 08 – Image processed by the application. In the left is the original image and the right is the array structured after the processing.

CONCLUSION

Usage of image processing and pattern recognition in images from HUD produces results that can increase flight tests efficiency. This paper proposed an automatic recognition system for text image recognition, based on a specific correlation algorithm. The application allows reduction of processing time in post-mission operations for data extraction from HUD images.

As future work:

1. Different strategies for pattern recognition (e.g. Neural networks) should be evaluated;
2. It is possible to use parallel processing techniques to improve image processing efficiency; and
3. Different setup schemes for post processing to improve the accuracy index should be experimented.

ACKNOWLEGMENT

We wish to thank the partial support given by ITA and IPEV, specially the 2012 class Flight Test Course Students, for supporting the measurement and the HUD flight tests campaigns.

Also we like to thank FINEP under agreement 01.13.0518.00 that funded the development of this tool and the presentation trip.

REFERENCES

[1] CHEN, T.; GHOSH, D. and RANGANATH, S.; Video-text extraction and recognition. TENCON 2004, IEEE Region 10 Conference, 1:319–322, November 2004.

[2] GARCIA, C. and DELAKIS, M.; Convolutional face finder: A neural architecture for fast and robust face detection. IEEE Transactions on pattern analysis and machine intelligence, 26(11), November 2004.

[3] YOKOBAYASHI, M. and WAKAHARA, T.; Segmentation and recognition of characters in scene images using selective binarization in color space and gat correlation. Eighth International Conference on Document Analysis and Recognition ICDAR’05, 1:167–171, 29 Aug.-1 Sept. 2005.

[4] YOKOBAYASHI, M. and WAKAHARA, T.; Binarization and recognition of degraded characters using a maximum separability axis in color space and gat correlation. 18th International Conference on Pattern Recognition ICPR 2006, 2:885–888, 20-24 Aug. 2006.

[5] WAKAHARA, T.; KIMURA, Y. and TOMONO, A.; Affine-invariant recognition of gray-scale characters using global affine transformation correlation. IEEE Transaction on pattern analysis and machine intelligence, PAMI-23:384–395, 20-24 Aug. 2001.

[6] OTSU, N.; A threshold selection method from gray-level histogram. SMC-9, 1979.

[7] SOMMERVILLE, I.; Software Engineering, 8ª edition, Person Education, 2007.

[8] GONZALEZ, R.C.; WOODS, R.E.; Digital Image Processing, Prentice-Hall, Inc., 2002.

[9] KOVASZNAY, L. S. G.; JOSEPH, H. M.; Image Processing. Proceedings IRE, vol. 43, pp. 560-570, 1955.

[10] SERRA, J.; Image Analysis and Mathematical Morphology, vol. 2, Academic Press, New York, 1988.