Vision-Based Text Segmentation System for Generic Display Units

José Carlos Castillo, María T. López, and Antonio Fernández-Caballero

Universidad de Castilla-La Mancha, Departamento de Sistemas Informáticos
& Instituto de Investigación en Informática de Albacete
Campus Universitario s/n, 02071-Albacete, Spain
`caballer@dsi.uclm.es`

Abstract. The increasing use of display units in avionics motivate the need for vision-based text recognition systems to assist humans. The system for generic displays proposed in this paper includes some of the usual text recognition steps, namely localization, extraction and enhancement, and optical character recognition. The proposal has been fully developed and tested on a multi-display simulator. The commercial OCR module from Matrox Imaging Library has been used to validate the textual displays segmentation proposal.

1 Introduction

There is an increasing use of displays in avionics. A very recent study [9] has investigated how vibration affects the reading performance and visual fatigue in an identification task of numeric characters shown on a visual display terminal. It was found that under vibration, the two display factors - font size and number of digits - significantly affect the human reaction time and accuracy of the numeric character identification task. Moreover, the vibrations in aircraft are mainly vertical and cause reading errors when the pilots read the instruments [2]. Therefore, automated vision-based systems seem to be good assistants to the human. Optical character recognition (OCR) is one of the most studied applications of automatic pattern recognition.

The text recognition problem can be divided into the following sub-problems: (i) detection, (ii) localization, (iii) tracking, (iv) extraction and enhancement, and, (v) recognition (OCR)[6]. Text detection refers to the determination of the presence of text in a given frame. Text localization is the process of determining the localization of text in the image and generating bounding boxes around the text. Text tracking is performed to reduce the processing time for text localization and to maintain the integrity of position across adjacent frames. Although the precise localization of text in an image can be indicated by bounding boxes, the text still needs to be segmented from the background to facilitate its recognition. This means that the extracted text image has to be converted to a binary image and enhanced before it is fed into an OCR engine. Text extraction is the stage where the text components are segmented from the background. Thereafter, the extracted text images can be transformed into plain text using OCR technology.

In this paper, we introduce a vision-based text segmentation system to assist humans in reading avionics displays. In these kinds of displays, be it of type CRT (cathode ray tube), LCD (liquid crystal display) or TFT-LCD (thin film transistor-liquid crystal display), the characters use to be placed at fixed positions. Therefore, our solution establishes a set of bitmaps - also called cells - in the display, in accordance with the number of rows and columns that the display is able to generate.

2   Usual Problems in Vision-Based Text Segmentation

Text segmentation strategies can be classified into two main categories: (1) difference based (or top-down) and (2) similarity based (or bottom-up) methods. The first method is based on the difference in contrast between the foreground and background, for example, the fixed thresholding method [13], global and local thresholding method [3], Niblack's method [20], and the improved Niblack method [22]. Indeed, thresholding algorithms have been used for over forty years for the extraction of objects from background [12]. The effectiveness of these approaches depends on the bi-modality of image histogram. This unfortunately is not always the case for real world images and as a result, the histogram-based image binarization techniques are not very effective. Thus, in general, these methods are simple and fast; however, they tend to fail when the foreground and background are similar. Alternative methods have been proposed in the literature to alleviate this problem, such as clustering-based methods [10,7], object attribute-based [11,14] neural networks-based binarization [21]. In [5] a binarization method for document images of text on watermarked background is presented using hidden Markov models (HMM). Alternatively, the similarity-based method clusters pixels with similar intensities. For example, Lienhart [8] used the split and merge algorithm, and Wang et al. [18] used a method in which edge detection, watershed transform, and clustering were combined. However, these methods are unstable because they exploit many intuitive rules for the text shape.

A big problem to be faced with vision-based text segmentation is camera calibration. Indeed, lens distortion is one of the main factors affecting camera calibration. A typical camera calibration algorithm uses one-to-one correspondence between the 3-D and 2-D control points of a camera [4,17]. The most used calibration models are based on Tsai's model [17] for a set of coplanar points or on the direct linear transformation (DLT) method originally reported by Abdel-Aziz and Karara [1]. Camera calibration techniques considering the lens distortion have long been studied. Utilized was the known motion of the camera [15] or the feature correspondences of a few images [16]. More recently, a new model of camera lens distortion has been presented [19]. The lens distortion is governed by the coefficients of radial distortion and a transform from ideal image plane to real sensor array plane. The transform is determined by two angular parameters describing the pose of the real sensor array plane with respect to the ideal image plane and two linear parameters locating the real sensor array with respect to the optical axis.

3   OCR for Generic Display Units

In this section a proposal for optical character recognition in generic displays is presented. In many cases, this kind of displays is only used for the presentation of alphanumeric characters. Also, in the majority of cases, the characters are placed in predefined fixed positions of the display. Therefore, our solution has to recognize the characters in a pre-defined set of cells (or bitmaps) of the display. Each bitmap, $B(i,j)$, contains a single character, $ch(i,j)$, where (i,j) is the co-ordinate of the cells row and column. The number of rows, N_r, and columns, N_c, of bitmaps being able to be generated on a given display defines the maximum number of recognizable characters, $N_r \times N_c$.

Generic display means that the system proposed has to recognize characters in any type of display used in avionics. For this reason, the approach adjusts the system to the dimensions of the display by definition. As the displays are prepared to be easily read by the pilots, it is assumed that the contrast between the background and the character is high enough.

Now, different steps followed to face the challenges appeared during bitmap localization, character extraction and enhancement, and optical character recognition phases are described in detail. Remember that the objective is to accurately recognize the ASCII values of the characters, $ch(i,j)$, contained in bitmaps $B(i,j)$.

3.1   Image Calibration

One of the greatest difficulties for an optimal segmentation in fixed positions of a textual display is the calculation of the exact starting and ending positions of each bitmap, (x_{init}, y_{init}) and (x_{end}, y_{end}), respectively, in the coordinate system (x, y) of the display. This it is an important challenge, as important screen deformations appear due to the camera lens used for the display acquisition process. These deformations consist of a *"ballooning"* of the image, trimmed in the point to which the camera focuses. For this reason, it is essential to initially perform a calibration of the image. Let us remind, once again, that the segmentation in this type of displays is essentially based in an efficient bitmaps localization. It is absolutely mandatory to scan any captured image with no swelling up, row by row, or column by column, to obtain the precise position of each bitmap $(B(i,j))$. On the contrary, pixels of a given row or column might belong to an adjacent bitmap.

In order to solve this problem, a "dots grid", $G_{dots}(x,y)$, is used as a pattern (see Fig. 1a). Each grid dot corresponds to the central pixel of a bitmap (or cell) $B(i,j)$ of the display. Once the grid points have been captured by the camera, the image ballooning and each dot deviation with respect to the others may be studied (see Fig. 1b).

Thanks to this information, and by applying the piecewise linear interpolation calibration method [4,17], any input image, $I(x,y)$, is *"de-ballooned"*. Thus, this swelling up is eliminated, providing a resulting new image $I_P(x,y)$. The centers of the dots are used to perform the calculations necessary to regenerate the original
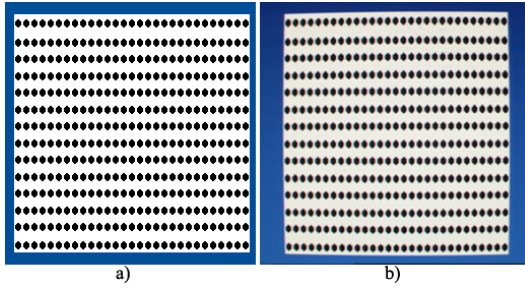
Fig. 1. (a) Optimal dots grid. (b) Captured dots grid.

rectangular form of the input image. In addition, the average, $\overline{G_{dots}(x,y)}$, of a certain number n_C of captured dots grids is used as input to the calibration method to augment the precision of the process.

3.2   Bitmap Localization

After *calibration*, the algorithms for *bitmap localization* are started. This phase is in charge of obtaining the most accurate localization of all bitmaps present in the calibrated image $I_P(x,y)$. In other words, the algorithm obtains, for each bitmap $B(i,j)$ its initial and final pixels' exact positions, (x_{init}, y_{init}) and (x_{end}, y_{end}), respectively. From the previous positions, also the bitmap's height, $B_h(i,j)$, and width, $B_w(i,j)$ are calculated.

For performing the precise bitmap localization, another template (or pattern) is built up. This template consists of a "bitmaps grid" (see Fig. 2a), that is to say, a grid establishing the limits (borders) of each bitmap. The process consists in capturing this "bitmaps grid", $G_{cells}(x,y)$, which, obviously, also appears convex after camera capture (see Fig. 2b). Again, a mean template image, $\overline{G_{cells}(x,y)}$, is formed by merging a determined number n_C of bitmaps grids captures. This process is driven to reduce noise that appears when using a single capture.

On the resulting average image, $\overline{G_{cells}(x,y)}$, a series of image enhancement techniques are applied. In first place, a binarization takes place to clearly separate
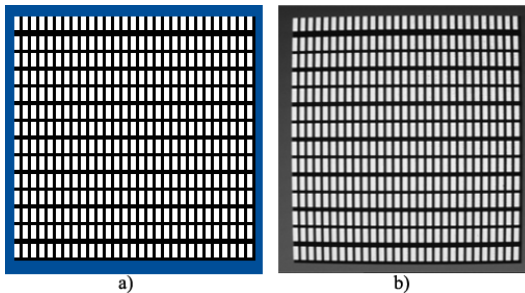
Fig. 2. a) Optimal bitmaps grid. (b) Captured bitmaps grid.
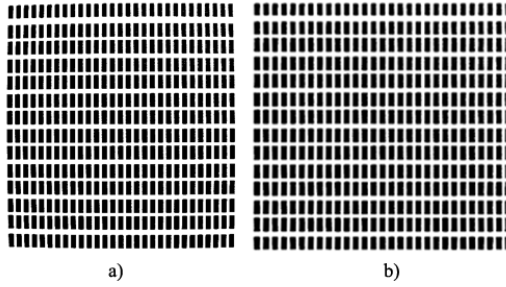
Fig. 3. (a) Binarized bitmaps grid. (b) Binarized and calibrated bitmaps grid.

the background from the foreground (see Fig. 3a). The binarization is performed as shown in formula (1).

$$BG_{cells}(x, y) = \begin{cases} 0, \text{if } \overline{G_{cells}(x, y)} \leq 135 \\ 255, \text{otherwise} \end{cases} \tag{1}$$

Next, the calibration algorithm is applied to the bitmaps grid (see Fig. 3b), similarly to the calibration performed on the dots grid, in order to correct the distortion caused by the camera lens.

Once the template has been calibrated, it is now the time to perform little refinements on the bitmaps. For this purpose, an object search algorithm is used in the captured image. It is necessary to eliminate possible spots that do not represent bitmap zones. For this, a filter to eliminate too small or too big "objects" is applied. Then, the generated "objects" are analyzed. It is verified that the total number of "objects" corresponds with the total number of bitmaps in the display (that is to say, in the template). If this is the case, the resulting "objects" are sorted from left to right and from top to bottom.

This way the initial and final pixels, (x_{init}, y_{init}) and (x_{end}, y_{end}), of each bitmap $B(i, j)$ have been calculated. This information provides the size of each bitmap; the height is gotten as $B_h(i, j) = y_{end} - y_{init} + 1$ and the width is obtained as $B_w(i, j) = x_{end} - x_{init} + 1$. Finally, the overall information of all bitmaps is also obtained. The mean size of the bitmaps is calculated through obtaining the mean height, $\overline{B_h}$, and the mean width, $\overline{B_w}$. This information is crucial to establish the mean size in pixels, $B_{sz} = \overline{B_h} \times \overline{B_w}$, which uses to be a fundamental parameter of an OCR to recognize the characters within the bitmaps.

While the position of the camera or the display type do not change during the segmentation process, the calibration and localization remain for all the searches in bitmaps. Nonetheless, some problems may arise during these phases. For instance, the camera may not be correctly adjusted. In this case, the processing of the cells fails irremediably. Some cells may appear united due to a sub-exposure (iris too much closed) or a de-focusing (see Fig. 4), or they disappear due to an over-exposure (iris too much open). Then, the localization function is unable to position the bitmaps appropriately, and, hence, to get their sizes. So, it is
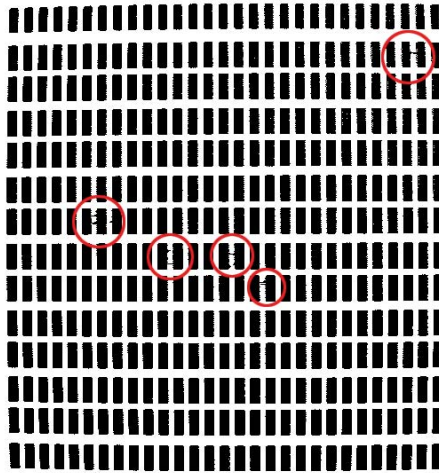
Fig. 4. Captured bitmaps grid after binarization in case of de-focusing

necessary to correctly adjust the camera lens and to repeat the complete process of calibrating the image and locating the bitmaps if any trouble occurs.

3.3   Bitmap Enhancement

This section introduces the enhancements introduced on the layout of each bitmap, $B(i,j)$. The image processing technique turns now in efficiently recognizing the ASCII character $ch(i,j)$ contained in a given bitmap. For it, we will work on the whole image, $I_P(x,y)$, as well as on each particular bitmap, $B(i,j)$. The process is based in eliminating deformations produced during the capture process (by using the values calculated during the calibration process) and in enhancing the visual quality of each bitmap, in consistence with its exact position within the display.

The 5×5 enhancement spatial mask shown in equation (2) is applied to differentiate the characters much more from the background (see Fig. 5). As you may observe in column *Enhanced Cell*, this filter enhances the characters respect to the appearance in column *Calibrated Cell*.

$$BR(i,j) = BG(i,j) \circ \begin{vmatrix} 1 & -2 & 3 & -2 & 1 \\ -2 & 3 & 5 & 3 & -2 \\ 3 & 5 & 9 & 5 & 3 \\ -2 & 3 & 5 & 3 & -2 \\ 1 & -2 & 3 & -2 & 1 \end{vmatrix} \qquad (2)$$

Next, a 2×2 erosion filter, as shown in equation (3) is applied, to limit the thickness of the character (see Fig. 5). The previously applied 5×5 enhancement filter unfortunately introduces an undesired effect of blurring the character

Calibrated Cell	Binarized Cell (without filters)	Enhanced Cell	Eroded Cell	Binarized Cell (with filters)
A	A	A	A	A
E	E	E	E	E
F	F	F	F	F
a	a	a	a	a
b	b	b	b	b
2	2	2	2	2
)	)	)	)	)
?	?	?	?	?

Fig. 5. Result of filtering the cells

borders. This effect is now corrected by means of the erosion filter, obtaining a better defined shape, as you may appreciate in column *Eroded Cell* of Fig. 5.

$$BE_{x,y}(i,j) = \min_{(x',y') \in [0..1,0..1]} BR_{x+x',y+y'}(i,j) \tag{3}$$

Now, a new binarization process is launched to leave the background in white color and the foreground (the character) in black color. This way, the analysis performed by a typical OCR is more reliable (see Fig. 5). When comparing the columns related to binarizations, with and without filters, you may observe that after applying the filters the characters are better defined, with finer outlines. All this is desirable for a better perception by the OCR.

Another necessary step for enhancing the segmentation consists in adding some margin at the four sides of the character, $ch(i,j)$. This way, the character does not touch the borders of the bitmap, as this usually reduces the hit ratio of the OCR. Hence, the pixels around the bitmap are eliminated (a rectangle of 1 pixel) to reduce the noise, and two rows and columns are added around the bitmap $BE(i,j)$.

Once the character has been binarized and the bitmap size has been augmented, isolated pixels are eliminated within the bitmap. The objective is to have the more regular characters. The pixels elimination algorithm follows a 4-connected criteria for erasing pixels that do not have 2 neighbors at least.

3.4 Optical Character Recognition

Finally, after all the enhancements performed, the bitmap is processed by the OCR to obtain the character. In our particular case, we have used the commercial

OCR module from Matrox Imaging Library (MIL). One of the principal param-
eter of this OCR - also of other commercial OCRs - is the size of the character
within the bitmap. Our experience has taken us to run the OCR with three
different sizes:

- Firstly, the character size is set to the mean size of all the display's bitmaps,
  $B_{sz} = \overline{B_h} \times \overline{B_w}$.
- Secondly, the character size is augmented in 1 pixel in height and width
  respect to the mean size of the display's bitmaps, namely, $\overline{B_h}+1$ and $\overline{B_w}+1$,
  respectively.
- Lastly, the character size is set to the exact height and width calculated for
  the concrete bitmap, that is, $B_w(i,j)$ and $B_h(i,j)$.

Obviously, the hit percentage obtained for each call is studied, and the recog-
nition result is the character with the highest matching score.

4   Data and Results

This section shows the results of the implementation of our algorithms. The
tests performed have demonstrated the capabilities of the system in relation to
the optical character recognition task. In order to get the necessary displays for
performing the tests, a simulator has been developed. The simulator is generic,
enabling to configure the characteristics of any kind of display, CRT, LCD, and

Table 1. Hit percentage for all ASCII characters

Char Code	% Hits	Char Code	% Hits	Char Code	% Hits	Char Code	% Hits
33	10	57	94	81	35	105	99
34	100	58	100	82	77	106	81
35	100	59	100	83	100	107	100
36	100	60	100	84	71	108	86
37	95	61	100	85	52	109	87
38	84	62	100	86	99	110	99
39	100	63	13	87	99	111	83
40	100	64	67	88	100	112	100
41	100	65	94	89	100	113	100
42	100	66	86	90	78	114	100
43	100	67	53	91	77	115	84
44	89	68	67	92	100	116	100
45	100	69	40	93	60	117	87
46	100	70	73	94	99	118	100
47	100	71	71	95	81	119	92
48	92	72	98	96	100	120	99
49	100	73	68	97	90	121	99
50	73	74	69	98	86	122	89
51	95	75	99	99	88	123	100
52	100	76	66	100	88	124	100
53	76	77	98	101	83	125	94
54	83	78	95	102	98	126	97
55	83	79	30	103	94		
56	52	80	78	104	100		

TFT-LCD. Due to the generality of the simulator, the size of a simulated display (rows and columns) may be easily modified for generating a wide range of displays.

Due to limitation in space, in this article we only offer the results of testing the character segmentation on a complete set of ASCII characters (from character code 33 to 126). The mean results of the recognition may be observed on Table 1, where the mean hit percentage overcomes an 86%, throwing a hit of 100% for 32 different characters, and a hit greater than an 80% for 71 different characters. There are only two characters offering a very poor hit percentage, namely, ASCII characters 33 and 66, corresponding to ? and ! symbols, respectively. This is a problem of the commercial OCR, as the library handles very badly the characters that present unconnected elements (formed by more than one shape).

5   Conclusions

A vision-based text segmentation system able to assist humans has been described in this paper. The proposed system for generic displays includes some of the usual text recognition steps, namely localization, extraction and enhancement, and optical character recognition. In avionics displays the characters use to be placed at fixed positions. Therefore, our solution establishes a set of bitmaps in the display, in accordance with the number of rows and columns that the display is able to generate. The proposal has been tested on a multi-display simulator and a commercial OCR system, throwing good initial results.

As future work, we are engaged in introducing some learning algorithms related to the type and size of the character sets in order to enhance the classification of the optical character recognizer.

Acknowledgements

References

1. Abdel-Aziz, Y.I., Karara, H.M.: Direct linear transformation into object space coordinates in close-range photogrammetry. In: Proceedings of the Symposium on Close-Range Photogrametry, pp. 1–18 (1971)
2. Andersson, P., von Hofsten, C.: Readability of vertically vibrating aircraft displays. Displays 20, 23–30 (1999)
3. Chang, F., Chen, G.C., Lin, C.C., Lin, W.H.: Caption analysis and recognition for building video indexing system. Multimedia Systems 10(4), 344–355 (2005)
4. Faugeras, O.: Three-dimensional computer vision: A geometric viewpoint. MIT Press, Cambridge (1993)
5. Huang, S., Ahmadi, M., Sid-Ahmed, M.A.: A hidden Markov model-based character extraction method. Pattern Recognition (2008), doi:10.1016/j.patcog.2008.03.004

6. Jung, K., Kim, K.I., Jain, A.K.: Text information extraction in images and video: A survey. Pattern Recognition 37, 977–997 (2004)
7. Kittler, J., Illingworth, J.: Minimum error thresholding. Pattern Recognition 19, 41–47 (1986)
8. Lienhart, R.: Automatic text recognition in digital videos. In: Proceedings SPIE, Image and Video Processing IV, pp. 2666–2675 (1996)
9. Lin, C.J., Hsieh, Y.-H., Chen, H.-C., Chen, J.C.: Visual performance and fatigue in reading vibrating numeric displays. Displays (2008), doi:10.1016/j.displa.2007.12.004
10. Otsu, N.: A threshold selection method from gray-level histogram. IEEE Transactions on Systems, Man, and Cybernetics 9, 62–66 (1979)
11. Pikaz, A., Averbuch, A.: Digital image thresholding based on topological stable state. Pattern Recognition 29, 829–843 (1996)
12. Prewitt, J.M.S., Mendelsohn, M.L.: The analysis of cell images. Annals of the New York Academy of Sciences 128(3), 1035–1053 (1965)
13. Sato, T., Kanade, T., Hughes, E.K., Smith, M.A., Satoh, S.: Video OCR: indexing digital news libraries by recognition of superimposed caption. ACM Multimedia Systems Special Issue on Video Libraries 7(5), 385–395 (1998)
14. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging 13(1), 146–165 (2004)
15. Stein, G.P.: Accurate internal camera calibration using rotation with analysis of sources of error. In: Proceedings of the Fifth International Conference on Computer Vision, p. 230 (1995)
16. Stein, G.P.: Lens distortion calibration using point correspondences. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 602–608 (1997)
17. Tsai, R.Y.: A versatile camera calibration technique for high accuracy 3-d maching vision metrology using off-the-shelf TV cameras and lenses. IEEE Journal of Robotics & Automation 3, 323–344 (1987)
18. Wang, K., Kangas, J.A., Li, W.: Character segmentation of color images from digital camera. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 210–214 (2001)
19. Wang, J., Shi, F., Zhang, J., Liu, Y.: A new calibration model of camera lens distortion. Pattern Recognition 41(2), 607–615 (2008)
20. Wolf, C., Jolion, J.: Extraction and recognition of artificial text in multimedia documents. Pattern Analysis and Applications 6, 309–326 (2003)
21. Yan, H., Wu, J.: Character and line extraction from color map images using a multi-layer neural network. Pattern Recognition Letters 15, 97–103 (1994)
22. Zhu, K., Qi, F., Jiang, R., Xu, L., Kimachi, M., Wu, Y., Aizawa, T.: Using adaboost to detect and segment characters from natural scenes. In: Proceedings of the International Workshop on Camera-based Document Analysis and Recognition, pp. 52–58 (2005)