

Numeral recognition for quality control of surgical sachets

Ernest Valveny, Antonio López
Computer Vision Center - Dept. Informàtica
Universitat Autònoma de Barcelona
Edifici O - Campus UAB,
08193 Bellaterra, SPAIN
{ernest,antonio}@cvc.uab.es

Abstract

In this paper we describe an application of OCR techniques to quality control in industrial production. The purpose of the system is to verify the correct printing of numerical information in sachets with surgical material. Numerals are printed on an aluminium surface covered by a transparent plastic film, which can produce shadows or reflections in the image. The main difficulties for character recognition arise from low acquisition resolution, noise, heavy or light printing, and different printing patterns. The system must perform with an error rate lower than 0.1% and with the minimal computation time. Thus, we have decided to use well-known and simple algorithms, adding to them some refinements which take advantage of specific domain-knowledge, making them more robust and reliable. The system is currently working with real production complying with the required specifications.

1 Introduction

Many industrial processes require OCR capabilities to test the quality of the final product. Some special fields, such as expiry-date, product identification, etc. are often printed somewhere on the product. These fields must be verified in order to check the correct product labelling.

It is clear that OCR is a very mature field, with many commercial systems performing well in controlled environments, such as printed paper documents. However, there are still some challenging applications where general-purpose OCR systems can not be applied to because of noise, low resolution, degradation, complex background, handwriting, etc. [2, 3]. Industrial processes are usually one such kind of applications. Because of irregular printing surface, low printing quality, etc. characters may appear with irregular shapes, can be broken, joined, distorted or with other kinds of degradation which make recognition more difficult.

Industrial processes must achieve really high performance indices with recognition rates very close to 100%. One simple recognition error can be prohibited in some applications. In addition, computation time must be very low to comply with production speed. The combination of both factors has led us to make use of well-known, simple and fast methods for image processing, character segmentation and recognition. However, we have added to them some ad-hoc refinements and improvements based on domain-specific knowledge, which permit to achieve the desired robustness in the recognition algorithms.

In the industrial process described in this paper, The final product are sachets containing surgical material which pass through a computer vision system performing several quality controls. One of these controls is the verification of the correct printing of three numerical fields: product reference, batch number and expiry-date. Difficulties for character recognition arise from four different factors:

- Characters are printed on an irregular surface made of aluminium and covered by a plastic film. Then, characters may appear distorted and shadows and reflections can be produced when illuminating the surface.
- Low image resolution. As many quality controls must be done on different regions of the sachet, a single image covering the whole area of the sachet, not only the printed area, must be acquired. This fact sets image resolution for character recognition around 175 dpi.
- Computation time. All quality controls must be performed at most in 0.4s. to comply with production line speed. As character recognition is only one of multiple controls computation time must be at most 0.08s. for processing all characters.
- Variability in printing. Actual thickness and shape of characters depend on some printer parameters, which are not always adjusted to the same values.

As we have argued before, we will use simple and fast algorithms. Character segmentation is based on thinning and connected component analysis [1] while character recognition is based on zoning methods [6]. Thus, we will concentrate on describing the refinements that we have introduced to these well-known methods in order to achieve robustness and performance requirements. In section 2, we will give a general overview of the whole system and its requirements. Then, in section 3 we will explain the algorithms employed at three levels: image processing, character segmentation and character recognition. Section 4 discusses results of the application to real production and, finally in section 5 we draw the conclusions from this work.

2 System overview

In figure 1(a) we can see an image of a correct sachet. The sachet is made of aluminium and is placed inside an external plastic film. The surgical material is located inside the sachet and must be kept sterile all along the whole process. A computer vision system is placed in the production line to perform a set of quality controls. There are fourteen different controls carried out by the computer vision system.

Figure 1 shows some examples of incorrect sachets, which should be rejected by due to different reasons. Some of the controls have to do with compliance to product specifications, such as sachet measures or distance between external and internal sachet edges; another group has to do with correct sachet placement inside the plastic film; others with correct sterilization of the surgical material and, finally there is a group of controls concerning the correctness of printing. Among them, we will focus on the verification of reference number, batch number and expiry date, printed at the center of the sachet - figure 1(a) - .These fields are used to identify the product inside the sachet. Just after passing through the computer vision system, sachets are placed together in boxes. Thus, it is very important to test that all sachets are well labelled.

Available time to test all these controls is 400ms. Thus, algorithms must be very simple and fast but, at the same time, they must be robust enough as the system is expected to have a global error rate not higher than 0.5% for all controls. In addition, two different kinds of lighting were needed for different controls. Then, we have decided to place two different acquisition systems, each one with its own lighting systems and its own computer to perform the controls. One of the acquisition systems is based on directional lighting to enhance surface irregularities, while the other one is based on diffused lighting in order to get an homogeneous lighting and reduce shadows. Printing verification has been associated to the last lighting system. As many controls concern the whole area of the sachet, the resolution of the image is only around 175dpi.



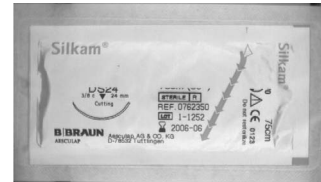
(a)



(b)



(c)



(d)

Figure 1. Examples of different sachets. (a) Correct sachet. (b) Thread outside the sachet. (c) Folded sachet. (d) Missing printing

3 Character recognition

Character verification requires three different kinds of processing: first, the whole image must be deskewed and numeral fields must be located and binarized. Then, numerals must be segmented and finally, they can be recognized. In this section, we describe the algorithms we have developed for these three levels of processing.

3.1 Image processing

Skew correction is the first operation carried out after image acquisition and before any further process is applied to it. Sachets will not always be in horizontal position. They are usually a little bit rotated due to the movement along the production line, but also due to the movement of the sachet inside the plastic film. This rotation must be corrected to



Figure 2. Skew correction. Original image and edge detection.

achieve a good performance in all quality controls, not only printing verification. It is a very critical operation because an error in skew correction can lead to errors in all quality controls.

Thus, to detect the skew of the sachet we combine two different sources of information: the bottom edge of the aluminium sachet and the horizontal lines printed around the word LOT which precedes the batch number. The reason for that is the great importance of accurate skew correction. We have found images where either bottom edge or horizontal lines surrounding LOT can not be well identified due to noise. Then, we have decided to compute both of them and combine them to get the skew angle.

To find the bottom edge of the sachet we compute the horizontal gradient to the region of interest, we binarize it and we analyze the largest connected components to distinguish between the edge of the sachet and the edge of the plastic film. To detect the horizontal lines around the word LOT, we have used the horizontal top hat operation [5]. After applying it, we must look for a pair of connected components aligned in the x-direction and having the right size. If both the bottom edge and the horizontal lines can be detected, we take the angle computed from the bottom edge because it is longer and more reliable. In figure 2 we can see the result of applying the vertical gradient and the top hat operation.

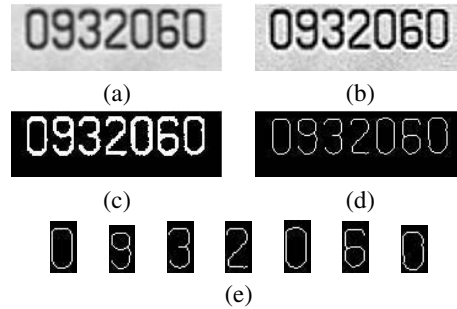


Figure 3. Segmentation of a numeral string. (a) Original image. (b) Sharpened image. (c) Binary image. (d) Thinned image. (e) Segmented numerals.

After skew correction, numeral fields must be located. They will not always be in the same absolute image location because the sachet moves inside the plastic film along its way through the production line, and because printing location depends on some printer parameters which are not always adjusted to the same values. Then, the area where numeral fields are supposed to be is found through the location of the words REF and LOT preceding the reference number and batch number and the location of the symbol preceding the expiry date. These symbols can be found in the image by simple template matching. Their location and the size of the numerals define the area where each field must be located.

Finally, as the algorithms for segmentation and recognition work with binary images, we must binarize the image. As there can be shadows, and lighting conditions may change slightly, we can not use a fixed threshold value. We select an optimal threshold value for each image assuming that the image histogram has two peaks, one corresponding to the numerals and the other one to the background, and finding the value which minimizes the misclassification probability [4].

3.2 Character segmentation

Segmentation is performed after binarization and thinning, based on the analysis of connected components [1]. Due to noise and shadows, numbers can touch after binarization. Then, any analysis based on connected components or projection profiles will yield to a lot of segmentation errors. Thus, we have decided to thin the binarized image. This way, characters are separated and the resulting connected components should correspond to numerals - figure 3 -.

However, there can still be some touching or broken nu-

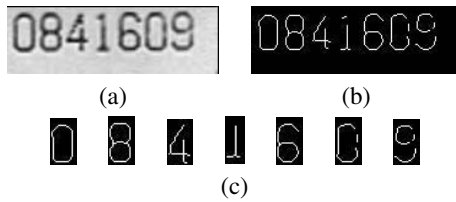


Figure 4. Segmentation of broken numerals. (a) Original image. (b) Thinned image. (c) Segmented numerals.

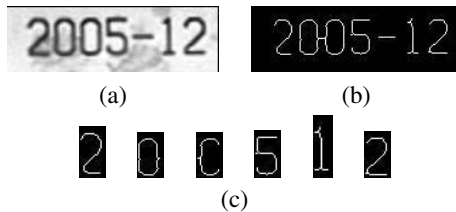


Figure 5. Segmentation of touching numerals. (a) Original image. (b) Thinned image. (c) Segmented numerals.

numerals - figures 4, 5 and 6-, which would lead to segmentation errors. To avoid them we take advantage of the knowledge of the number of numerals in each field and of the width of each numeral. Thus, we apply a post-process to split the components too wide and to join the components too small and close from each other.

3.3 Character recognition

The algorithm for numeral verification is based in the well-known technique of zoning [6]. Each numeral is divided in a set of regions, and some feature is computed for each region. The resulting feature vector is taken as the

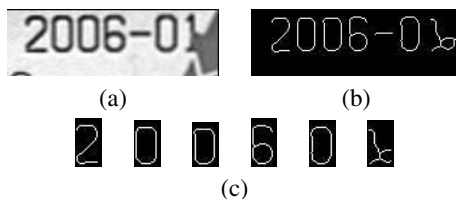


Figure 6. Segmentation of numerals joined to the arrow. (a) Original image. (b) Thinned image. (c) Segmented numerals.

representation of the numeral. The zoning method we have implemented has the following properties:

- The size of each region is not constant. Each image is divided into 5 rows and three columns. However, the size of all rows and all columns is not always the same. Each row and column has its own size in order to locate them in the most discriminant areas of the image.
- The feature computed for each region is a measure of the significance of that region inside the whole numeral. For each region, the number of white pixels is computed. If this number exceed some percentage of the total number of white pixels in the image, the feature value is set to 1. Otherwise, it is set to 0. For the region located at the center of the numeral, the feature value is multiplied by 2.
- Three feature values are added to the feature vector, combining information of various regions. These added values help to distinguish among similar numbers. They measure the global significance of the given combination of regions inside the whole numeral. These selected regions correspond to the upper area, the upper-left area and the down-right area of the numeral.

The feature vector - figure 7 - is used to classify the image into one of the classes corresponding to each numeral. Classification is based on a distance measure between the feature vector of the image and the feature vector of each numeral model. If $I = (i_1, \dots, i_n)$ is the image vector and $M = (m_1, \dots, m_n)$ is the model vector, the distance d is defined by taking the absolute value of the differences among feature values:

$$d = \sum_{j=1}^{j=n} |i_j - m_j| \quad (1)$$

Each image is associated to the class with minimal distance. If there are several classes sharing the minimal distance, we take advantage of the fact that we are carrying out a verification process, i.e, we know which numeral should be at every position. Then, if the correct numeral is among the set of candidates, it is assigned to the image, but with a flag signaling that it is an ambiguous classification. If there are too many ambiguous numerals the numeral string will be rejected. It is to be noted that we have defined three different models for numeral 1 and eight models for numeral 4, to be able to handle variability in printing of these numerals.

After classification of each numeral, the whole string (reference number, batch number or expiry date) must be accepted or rejected depending on the number of numerals misclassified or ambiguous. We will reject one string if there are more than one misclassified numeral or more than



1	1	1
1	0	1
1	2	1
0	0	1
1	1	1
2.0	0.5	0.0

Added features

Figure 7. Feature vector of a numeral. (a) Numeral division. (b) Feature vector.

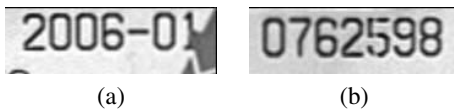


Figure 8. Difficult cases well recognized for the system. (a) Overlapped numerals. (b) Broken numerals.

two ambiguous numerals (these parameters can be changed by the operators). This way, we can accept one string even if there is some error due to noise or similarity between numerals. On the other hand, we do not risk of accepting incorrect strings because different reference numbers will have at least three different numerals in their strings.

4 Results

Currently, the system is working, processing an average of 6000 sachets per hour, sixteen hours per day, complying with constraints about recognition rates and computing time. To measure the real performance of the system, we have analyzed one-year real production. In this time, the system has processed 4.109.713 sachets. The percentage of rejected sachets due to printing errors has been 0.22%. This rate includes three kind of sachets: incorrect sachets rejected by the system, correct sachets rejected due to recognition errors and incorrect sachets due to errors in printing of the name of the product (which are included in the same class as errors in printing of digits). As we do not have statistics about the real rate of incorrect sachets, we cannot provide an exact measure of the system performance, but we can say that it is higher than 99.9% in real production.

In table 1 we show the result of numeral recognition for 200 correct sachets, i.e 3600 numerals. All sachets have been accepted by the system although there are some re-

N. of numerals	Error	Ambiguous	Accept.	Ambiguous + Accept.
3600	0.22%	0.08%	99.69%	99.77%

Table 1. Recognition rates for 200 sachets.

jected numerals, but not more than one rejected numeral for each string. Even in cases with broken or touching numerals - figure 8 - the system performs well.

To test the performance of the system with incorrect sachets, we have randomly taken 25 images of sachets with reference number 0312096, and we have tried to verify them with reference number 0812366. There are only three different numerals, which is the minimal difference between two reference numbers. Moreover different numerals are 3 and 8, 0 and 3, and 9 and 6, some of which could be easily confused. The result has been that all sachets have been rejected and that all incorrect numbers have been identified.

5 Conclusions

We have described an OCR system which is currently performing well in an industrial process for quality control. The system should meet two simple, but difficult requirements. It must work well in all situations and it must be fast enough to adapt to the production line speed. Thus, we should use very simple algorithms.

We have decided to use a variation of well-known methods (connected component analysis for segmentation and zoning for recognition). We have shown how, taking advantage of domain-specific knowledge (numeral width, number of numerals in each string, number of different numerals between two reference numbers, different shapes for each numeral), we can make these algorithms robust enough to perform well in this specific scenario. The final error rate is about 0.1% with real production.

References

- [1] R. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Transactions on PAMI*, 18(7):690-706, 1996.
- [2] S. Mori, H. Nishida, and H. Yamada. *Optical Character Recognition*. John Wiley and sons, 1999.
- [3] S. Rice, G. Nagy, and T. Nartker. *Optical Character Recognition: An illustrated guide to the frontier*. Kluwer Academic Publishers, 1999.
- [4] A. Rosenfeld and A. Kak. *Digital Picture Processing*. Academic Press, San Diego, California, 1982.
- [5] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.
- [6] O. Trier, J. A.K., and T. Taxt. Feature extraction methods for character recognition - a survey. *Pattern Recognition*, 29(4):641-662, 1996.