Active contour based optical character recognition for automated scene understanding

Joanna Isabelle Olszewska

PII: S0925-2312(15)00213-1
DOI: http://dx.doi.org/10.1016/j.neucom.2014.12.089
Reference: NEUCOM15180

To appear in: Neurocomputing

Received date: 1 June 2014
Revised date: 5 November 2014
Accepted date: 1 December 2014

Cite this article as: Joanna Isabelle Olszewska, Active contour based optical character recognition for automated scene understanding, Neurocomputing, http://dx.doi.org/10.1016/j.neucom.2014.12.089

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting galley proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
Active Contour Based Optical Character Recognition for Automated Scene Understanding

Joanna Isabelle Olszewska
School of Computing and Technology, University of Gloucestershire
The Park, Cheltenham, GL50 2RH, United Kingdom
jolszewska@glos.ac.uk

Abstract
In this paper, we present a new optical character recognition (OCR) approach which allows real-time, automatic extraction and recognition of digits in images and videos. Our method relies on active contours in order to robustly extract optical characters from real-world visual scenes. The detected character recognition is based on template matching. Our developed system has shown excellent results when applied to the automated identification of team players’ numbers in sports datasets and has outperformed state-of-the-art methods.

Key words: Pattern Recognition, Information Fusion, Fast Segmentation, Scene-Text Localization and Extraction, Active Contours, Snake Evolution, Chromaticity, Online Optical Character Recognition, Sport Events, Image and Video Datasets, Automated Scene Understanding

1. Introduction
Scene text extraction in images and videos is of prime importance for automated scene understanding and video content analysis [1]. Besides the perceptual content of the images and videos analyzed with vision-based techniques like object detection [2], object recognition [3], or tracking [4], the semantic content of the images and videos provides inestimable information in terms of scene objects and their relations [5], [6], useful for applications such as automated comments production and media archiving.

In particular, sport events’ refereeing and analysis rely on the identification of team players based on the recognition of the numbers on their uniform rather than on their face recognition. Indeed, face detection techniques [7] are not intrinsically adapted to identify a player whose back is turned to the camera, in which case his face is poorly or not visible at all.

Automatic extraction of the scene text involves the use of an optical character recognition (OCR) system to detect and recognize the textual characters from a given image.

In this work, we focus on the automatic player identification in images of any type for sport scene analysis, based on the detection and recognition of the player’s jersey number, and therefore, on the development of a full, efficient OCR system for this purpose.

OCR major phases are (i) character extraction and (ii) character recognition. In the first step, the system localizes and extracts the character by detecting its geometrical features like edges [8] or color features [9], or both [10]. In the second step, character recognition is usually performed by matching [11] or by using classifiers, e.g. AdaBoost [12]. However, these existing OCR systems are mainly applied to recognize license plate numbers or handwritten characters, whereas player number recognition presents additional challenges. Indeed, the foreground, i.e. the character, could be highly skewed with respect to the camera, or the background, i.e. the jersey, could be folded so that part of the number could be hidden. Moreover, sport images are often blurred, since cameras or players or both are quickly moving.

Most of the existing OCR approaches developed to identify numbers on team player’s uniform exploit the temporal redundancy of a character across several frames and thus are limited to analyze only videos [13], [14], [15], [16], [17], [18] and not suited for tasks such as still image dataset retrieval. Other works [19] use both facial and textual cues, but present a low computational speed.

Hence, variations of scene text due to differences in size, style, font, color, orientation, and alignment, as
well as foreground motion/distortion and background noise make the problem of automatic identification of team players' numbers very challenging [20].

In this paper, we firstly propose to automatically extract characters from images based on both their local and global properties in order to obtain a reliable digit detection. For this purpose, we adopt two different segmentation approaches run in parallel by our OCR system. Most scene text in sport events are designed to be easily read, thereby resulting in large contrast between foreground and background as well as strong edges at the boundaries of the scene text. Thus, our first detection process relies on pixel chromaticity/achromaticity properties. The other detection method is based on active contours which are curves evolving towards the scene text boundaries, driven by forces based on image properties. This approach allows us to develop a full OCR system which has two distinct phases, i.e., the detection and the recognition, rather than a reduced OCR system directly matching shape/contour templates as in [21], [22], [23]. Moreover, our active contours provide more information than the methods based on sole edge maps [8]. Indeed, edge maps usually present not connected edges and then provide only sparse boundaries of a character to extract, whereas active contours allow to delineate the boundaries of the entire targeted digit, and thus extract it as a whole.

Next, we use a digit template to recognize separately the characters extracted by our chromaticity-based technique and active contour method, respectively. Then, we fuse the outputs obtained through this recognition process, following a new merging strategy combining the ideas of weighted vote and majority vote.

The resulting OCR system robustly deals with scene digits detection and recognition, while it is computationally effective. No temporal redundancy assumption is made in our method, which is thus valid not only for video frames, but also for still images such as those contained in sport datasets or on Internet.

In our approach, players could be identified even in back profile, since our OCR system detects and recognize characters which could be anywhere on the team player's clothes. Hence, the contributions of this paper are:

- the use of the active contour evolution in the optical character detection process combined at a late stage, with the chromaticity-based segmentation in order to extract optical characters robustly;
- the new merging strategy itself which takes advantage of majority vote and weighted vote approaches;
- the development of a powerful OCR system based on this innovative fusion for the character detection with the template matching-based technique for the fast character recognition, for the automated identification of digits in online images and videos.

The paper is structured as follows. In Section 2, we present our optical character recognition approach for computationally effective and reliable extraction and identification of digits in real-world images. In Section 3, we report the excellent performance of our method compared to state-of-the art approaches, tested in context of soccer players' image database. Conclusions are drawn up in Section 4.
2. Character Recognition and Identification System

In this section, we present our optical character recognition approach (Fig. 1) for the reliable identification of soccer player’s numbers present in real-world images and videos. Firstly, the studied image is segmented by both chromaticity-based approach and active contour approach, as explained in Section 2.1. Then, the extracted character is recognized by means of template matching described in Section 2.2.

2.1. Character Extraction

Character extraction consists here in image segmentation and character detection. On one hand, the image is binarized based on chromaticity properties of the foreground and background pixels as described in Section 2.1.1. Next, the characters’ inner boundary tracing algorithm is applied in order to extract the numbers as presented in Section 2.1.2. On the other hand, active contours are processed and then, they delineate the boundaries of the character under investigation as explained in Section 2.1.3. Hence, this later approach gives a feedback on the first processed extraction, leading to a more robust character detection.

2.1.1. Image Segmentation

Let us consider a color image $I$, where $M$ and $N$ are its width and height, respectively. The first step to extract numbers or foregrounds of this still image is to separate them from their background. In fact, in football, players’ number color is chosen by the football league to be in contrast with players’ kit (shirt and sweater), in order to allow visibility of the number in diverse conditions. The study of [24] has found that this contrast is the most important in the hue, saturation and value (HSV) color space when looking at the saturation of the number pixels and the jersey pixels. Consequently, the image $I$ could be segmented based on the low and high saturated pixels, i.e. objects’ achromatic and chromatic colors, respectively, leading to a binary image $I_B$. In particular, a color pixel under investigation $P = [P_h, P_s, P_v]$ is considered as achromatic if its saturation ($P_s$) is below the saturation threshold ($Y_S$) or if its intensity ($P_v$) is below intensity threshold ($Y_V$). If the pixel saturation and intensity are above these thresholds, then it is considered as chromatic.

The segmentation is initialized by defining the mean color vector of the jersey $J = [J_h, J_s, J_v]$ and the mean color vector for the number $N = [N_h, N_s, N_v]$, based on provided image samples. Next, the image $I$ is processed depending if the number color is chromatic or

### Algorithm 1 Achromatic-color Number & Achromatic-color Jersey

```latex
\text{if} \ ((N_S < Y_S \text{ or } N_V < Y_V) \text{ and } (J_S < Y_S \text{ or } J_V < Y_V)) \text{ then} \\
\quad \text{if} \ J_V > Y_{\text{thresh}} \text{ then} \\
\qquad \text{for all } P \text{ do} \\
\qquad\quad \text{if} \ P_V < Y_{\text{thresh}} \text{ then} \\
\qquad\qquad I_B(P) = 0 \quad \text{\(\triangleright\) set pixel as black} \\
\qquad\quad \text{else} \\
\qquad\qquad I_B(P) = 1 \quad \text{\(\triangleright\) set pixel as white} \\
\qquad\quad \text{end if} \\
\qquad \text{end for} \\
\quad \text{else} \\
\quad\quad \text{for all } P \text{ do} \\
\quad\quad\quad \text{if} \ P_V < Y_{\text{thresh}} \text{ then} \\
\quad\quad\qquad I_B(P) = 1 \quad \text{\(\triangleright\) set pixel as white} \\
\quad\quad\quad \text{else} \\
\quad\quad\qquad I_B(P) = 0 \quad \text{\(\triangleright\) set pixel as black} \\
\quad\quad \text{end if} \\
\quad \text{end for} \\
\text{end if} \\
\text{return } I_B
```

### Algorithm 2 Achromatic-color Number & Chromatic-color Jersey

```latex
\text{if} \ ((N_S < Y_S \text{ or } N_V < Y_V) \text{ and } (J_S > Y_S \text{ and } J_V > Y_V)) \text{ then} \\
\text{for all } P \text{ do} \\
\quad \text{if} \ ((P_S < Y_S) \text{ and } (P_V < Y_V)) \text{ then} \\
\qquad I_B(P) = 0 \quad \text{\(\triangleright\) set pixel as black} \\
\quad \text{else} \\
\qquad \text{if} \ (h_{\text{diff}}(J_H, P_H) < H_{\text{thresh}}) \text{ then} \\
\qquad\quad I_B(P) = 1 \quad \text{\(\triangleright\) set pixel as white} \\
\qquad \text{else} \\
\qquad\quad I_B(P) = 0 \quad \text{\(\triangleright\) set pixel as black} \\
\quad \text{end if} \\
\text{end for} \\
\text{return } I_B
```


Algorithm 3 Chromatic-color Number & Achromatic-color Jersey

if \((J_s < Y_s \text{ or } J_V < Y_V) \text{ and } (N_s > Y_s \text{ and } N_V > Y_V)\) then
    if \(J_V > V_{\text{threh}}\) then
        for all \(P\) do
            if \((P_s < Y_s \text{ and } P_V < Y_V)\) then
                if \(P_V < V_{\text{threh}}\) then
                    \(I_B(P) = 0\) \(\triangleright \) set pixel as black
                else
                    \(I_B(P) = 1\) \(\triangleright \) set pixel as white
                end if
            else
                \(I_B(P) = 0\) \(\triangleright \) set pixel as black
            end if
        end for
    else
        for all \(P\) do
            if \((P_s < Y_s \text{ and } P_V < Y_V)\) then
                if \(P_V > V_{\text{threh}}\) then
                    \(I_B(P) = 0\) \(\triangleright \) set pixel as black
                else
                    \(I_B(P) = 1\) \(\triangleright \) set pixel as white
                end if
            else
                \(I_B(P) = 0\) \(\triangleright \) set pixel as black
            end if
        end for
    end if
end if
return \(I_B\)

Algorithm 4 Chromatic-color Number & Chromatic-color Jersey

if \((N_s > Y_s \text{ and } N_V < Y_V) \text{ and } (J_s > Y_s \text{ and } J_V > Y_V)\) then
    for all \(P\) do
        if \((h_{\text{diff}}(N_H, P_H) < H_{\text{threh}})\) then
            \(I_B(P) = 0\) \(\triangleright \) set pixel as black
        else
            \(I_B(P) = 1\) \(\triangleright \) set pixel as white
        end if
    end for
end if
return \(I_B\)

achromatic and if the jersey color is chromatic or achromatic, leading to four cases, i.e. to four Algorithms 1-4. The segmentation is based on the hue threshold \(H_{\text{threh}}\) and the hue difference in the case of a chromatic-color jersey, whereas the intensity difference and the intensity threshold \(V_{\text{threh}}\) are used in the case of an achromatic-color jersey [24]. In the case where the number has an achromatic color and the jersey color is chromatic (Algorithm 2), the hue difference \(h_{\text{diff}}\) is defined as follows:

\[
h_{\text{diff}}(J_H, P_H) = \begin{cases} 
\Delta(J_H, P_H) & \text{if } \Delta(J_H, P_H) < 180^\circ, \\
360^\circ - \Delta(J_H, P_H) & \text{otherwise}, 
\end{cases}
\]

(1)

with

\[\Delta(J_H, P_H) = |J_H - P_H| . \]

(2)

When both the jersey and the number have chromatic colors, the image is segmented as described in Algorithm 4, using the hue difference \(h_{\text{diff}}\) defined as follows:

\[
h_{\text{diff}}(N_H, P_H) = \begin{cases} 
\Delta(N_H, P_H) & \text{if } \Delta(N_H, P_H) < 180^\circ, \\
360^\circ - \Delta(N_H, P_H) & \text{otherwise}, 
\end{cases}
\]

(3)

with

\[\Delta(N_H, P_H) = |N_H - P_H| . \]

(4)

2.1.2. Character Detection

In the binarized image \(I_B\) computed by the process explained in Section 2.1.1, jerseys appear as white objects, while numbers as black ones. Based on that fact, tracing internal boundaries of these objects is an efficient method for number region localization and extraction. For this purpose, we have adapted the Boundary Tracing approach [25]. Hence, our process presented in Algorithm 5 initiates by tracing all the boundaries \(B_i\) within the segmented binary image, and then, in relation to the specific area aspect ratio \(F\) characterizing the number region, the boundaries are filtered, in order to select only those containing numbers. Once this process is completed, the binary image \(I_B\) is cropped and the cropped image \(I_C\) is transferred to the recognition stage which then identifies the numbers as detailed in Section 2.2.

This section has presented the single digit case. The identification of two-digit numbers is as follows. If two cropped images are of the same size and are in adjacent bounding rectangles, they are flagged as forming a two-digit number [26].


Algorithm 5 Boundary Tracing

\begin{enumerate}
\item Find boundaries \( B = \{ B_i \} \) of all objects
\item for all \( B_i \) do
  \begin{enumerate}
  \item if \( B_i \) is black object then
    \begin{enumerate}
    \item if \( B_i \) dimensions = \( F \) dimensions then
      \begin{enumerate}
      \item \( x_1 = \min(B_i[1]) \)
      \item \( y_1 = \min(B_i[2]) \)
      \item \( x_2 = \max(B_i[1]) \)
      \item \( y_2 = \max(B_i[2]) \)
      \item \( I_C = I_B[x_1 : x_2][y_1 : y_2] \)
    \end{enumerate}
    \item else
      \begin{enumerate}
      \item Ignore boundary
      \item Go the next boundary
      \end{enumerate}
    \end{enumerate}
  \end{enumerate}
  end if
end for
\item return \( I_C \)
\end{enumerate}

2.1.3. Active Contours

In this work, multi-feature vector flow active contours are used to provide another character segmentation in order to have a feedback on the results computed in Sections 2.1.1-2.1.2.

Indeed, multi-feature active contour is a parametric planar deformable curve \( \mathbf{C}(s) = [\mathbf{C}_x(s), \mathbf{C}_y(s)] \), with \( 0 \leq s \leq 1 \), which evolves from an initial position to object boundaries with the use of the MFVF field \( \mathbf{E}(x,y) = [\xi_x(x,y), \xi_y(x,y)] \) [27].

In this framework, the convergence of the curve is guided by internal and external forces, which are involved in a gradient descent process. The internal forces constrain the active contour shape, in the way to ensure regularity and smoothness of the curvature. MFVF external force regroups all the selected features in one original bidirectional force, enabling the active contour to reach its final accurate position, even in complex situations.

Formally, the deformable curve \( \mathbf{C}(s,t) \) is modeled itself by a B-spline paradigm in order to be computationally efficient, and must satisfy the following dynamic equations,

\begin{align*}
\mathbf{C}_x(s,t) &= \alpha \mathbf{C}''_x(s,t) - \beta \mathbf{C}_x''(s,t) + \xi_x(x,y) \quad (5) \\
\mathbf{C}_y(s,t) &= \alpha \mathbf{C}''_y(s,t) - \beta \mathbf{C}_y''(s,t) + \xi_y(x,y) \quad (6)
\end{align*}

where \( \mathbf{C}_x'' \), \( \mathbf{C}_y'' \), \( \mathbf{C}_x''' \), \( \mathbf{C}_y''' \), respectively, are the second and fourth-order derivatives with respect to the parameter \( s \) of the curve, \( \alpha \) is the curvature rigidity coefficient, and \( \beta \) is the curvature elasticity coefficient.

The active contour, found by solving (5) and (6), could be, in practice, roughly initialized from a distance of the target, as the MFVF force offers a large capture range. This obtained fast multi-feature active contour owns high-deformation capabilities that are well suited for tracking non-rigid objects whose shapes change markedly. Indeed, tracking with the multi-feature active contour could be performed by minimizing the associated energy functional, for each corresponding feature, in each frame.

Moreover, this computed curve enables precise foreground segmentation, without any kind of assumption about the object appearance [4].

2.2. Character Recognition

For the recognition of the characters extracted either with the chromaticity-based technique or active contour method, we have adopted the template matching approach. Indeed, this pattern classification method is well suited in the identification of small regions [28], which is the case in our application.

The basis of template matching is that a processed image is compared to each of the images stored within a template. In many instances, the extracted number region has smaller or larger dimensions compared to the template dimensions, or has not the same orientation. Thus, the extracted number image has first to be rotated and rescaled to fit the template orientation and size, respectively.

Next, the correlation coefficients \( \rho_{CH} \) and \( \rho_{AC} \) between a template image \( t \) and the processed image containing the extracted character by means of chromaticity-based segmentation or active contour approach, respectively, are computed as follows:

\[
\rho_{CH} = \frac{\sum_m \sum_n (T_{mn} - \bar{T})(S_{mn} - \bar{S})}{\sqrt{\sum_m \sum_n (T_{mn} - \bar{T})^2} \sqrt{\sum_m \sum_n (S_{mn} - \bar{S})^2}}, \quad (7)
\]

where \( T_{mn} \) are the values of the pixels of the template image with an \( m \times n \) size and a mean \( \bar{T} \); \( S_{mn} \) are the values of the pixels of the rescaled, rotated, chromaticity-segmented image with a mean \( \bar{S} \);

and

\[
\rho_{AC} = \frac{\sum_m \sum_n (T_{mn} - \bar{T})(X_{mn} - \bar{X})}{\sqrt{\sum_m \sum_n (T_{mn} - \bar{T})^2} \sqrt{\sum_m \sum_n (X_{mn} - \bar{X})^2}}, \quad (8)
\]

where \( T_{mn} \) are the values of the pixels of the template image with an \( m \times n \) size and a mean \( \bar{T} \); \( X_{mn} \) are the values of the pixels of the rescaled, rotated, binarized image with the extracted shape, with a mean \( \bar{X} \).
When the structure of a processed image is greatly similar to the structure of one of the template images, the correlation coefficient value is high.

For each template image $i$, the global correlation coefficient $r_i$ is defined as follows:

$$r_i = \frac{\rho_{CHi} + \rho_{ACi}}{C_E}, \quad \text{with} \quad i \in [0, 9], \quad (9)$$

where $C_E$ is the number of character extraction methods involved in the process. In our work, $C_E$ is equal to 2, since we used the chromaticity-based technique and the active contour approach.

Then, the number is identified if the following condition is met:

$$\exists i \in [0, 9] \max(r_i) \wedge (1 - \rho_{CHi}) < \epsilon_i \vee (1 - \rho_{ACi}) < \epsilon_i), \quad (10)$$

with $\epsilon_i$, a constant value.

The digit is thus recognized as being the one corresponding to the $i$th image of the template when the sum of the correlation coefficients of the two character extraction methods is maximum and when at least one of the correlation coefficient of the two character extraction methods is very high, as per Eq. (10). If the solution of Eq. (10) is empty or not unique, the digit is then flagged as unrecognized.

To recognize two-digit numbers, single numbers flagged as constituting a two-digit number in Section 2.1 are processed individually by matching each of them with the template. The two-digit number is then formed and identified based on that information.

We can notice that the use of the template matching technique is well suited for our system of automatic number recognition of soccer players. On one hand, template matching is particularly fast when used in context of our system, because it requires only the recognition of numerical characters, contrary to a wider range of alphanumerical characters as in other applications, such as license plate recognition (LPR). Indeed, our template stores in total only 10 images of one-digit numbers (0 to 9). Hence, the matching is performed against a maximum of ten stored images, in order to recognize the extracted character, which is computationally very efficient. Moreover, the scale sensitivity of the template matching technique is used in our work as an advantage, since smaller dimensions of the template dimensions lead to a faster matching. On the other hand, the recognition rate obtained by our implementation of this method in our system is much higher than those presented in the literature as discussed in Section 3.

3. Discussion

To validate our method, we have carried out experiments which consist in automatically recognizing numbers from the soccer players’ jerseys within a database containing data images with soccer-related content, as such illustrated in Fig. 2.

For this purpose, our system has been applied on a dataset containing 4500 football images whose average resolution is of 230x330 pixels and which were captured in outdoor environment. This database owns challenges of quantity, pose and scale variations of the players. Moreover, the colors of the teams’ uniforms have various colors and the fonts on the players’ jerseys could vary strongly.

All the experiments have been run on a computer with an Intel(R) Core(TM)2 Duo 2.53 GHz processor, 4 Gb RAM, and using our OCR software implemented with MatLab. Our system is able to support different types of image formats such as jpeg, tif, bmp, and png.

In order to assess the performance of our OCR system, we use the following criteria:

$$\text{extraction rate} = \frac{CL \times 100}{TT}, \quad (11)$$

$$\text{recognition rate} = \frac{CR \times 100}{TT}, \quad (12)$$

with $CL$, the number of correctly localized characters, $CR$, the number of correctly recognized characters, and $TT$, the total number of tested characters.

Some examples of the results of our OCR system are presented in Fig. 2. These samples present difficult situations such as variability of the jerseys’ colors, i.e. different pixels’ chromaticity properties of the foregrounds and the backgrounds; numbers’ changing characteristics, i.e. different characters’ geometrical and spatial properties; scale effects such as zoom out or close-up.

We can observe that using our approach, characters are correctly extracted and correctly recognized, despite their geometrical and chromatical differences. Hence, our OCR system is robust towards changes in numbers and colors of the foregrounds and the backgrounds as well as towards variations of fonts, size, and orientation of the characters. Moreover, the system is robust even in case the chromatic detection provides a sparse result such as in Fig. 2(f), because of the effect of the feedback provided by the active contours as displayed in Fig. 2(e).

In Table 1, we have reported the extraction and recognition rates of our OCR method against the rates achieved by approaches using chromatic/achromatic...
Figure 2: Examples of results obtained with our OCR system. First column: input image. Second column: chromaticity-based segmentation. Third column: active contour-based segmentation. Fourth column: recognized character.

segmentation (C/A Segm.) or template matching (TM) MSERE + TM [19], C/A Segm. + CL [24], (C/A Segm. + TM) [1].

We can see in Table 1 that our OCR method relying on the active contour feedback into the OCR process which combines chromatic/achromatic segmentation and matching-based recognition outperforms the state-of-art approaches for soccer player’s number identification. In particular, we can notice than the extraction rate is improved when using the active contour as feedback for the chromatic/achromatic segmentation instead of using C/A segm. alone. Our OCR method outperforms also other state-of-the-art techniques such as maximally stable extremal region extraction. On the other hand, we can observe the positive effect of our active contour based automatic feedback approach on the recognition rate compared to other classification methods.

From Table 1, we can conclude that the incorporation of the active contours through our fusion process
increases the robustness of the OCR system, and provide both high detection and recognition rates.

For all the dataset, the average computational speed of our combined OCR method is in the range of few seconds, and thus, our developed system could be used in context of online visual scene analysis.

4. Conclusion

Scene text detection in online data such as images and videos is a challenging topic we have studied in this work. For this purpose, we have developed a new OCR approach relying on two detection approaches, namely, chromaticity-based segmentation and active contour method, to robustly extract the textual characters. Template matching is used each time for the character recognition step. The fusion of the resulting information is performed at a later stage through an original process based on weighted vote and majority vote. Our OCR system is more performant than the state-of-the-art ones in both extraction and recognition of soccer players’ numbers. Furthermore, our OCR approach is well suited for the automatic retrieval and analysis of online, visual data about team sports.

References


