Proceedings of the 17th International Conference on Pattern Recognition (ICPR2004), Cambridge, UK, August 23-26, 2004, IEEE-CS Press, pp. 634-637.

Text Extraction from Web Images Based on A Split-and-Merge Segmentation Method Using Colour Perception

D. Karatzas and A. Antonacopoulos

Pattern Recognition and Image Analysis (PRImA) Group,
Department of Computer Science, University of Liverpool, Liverpool L69 3BX, United Kingdom
http://www.csc.liv.ac.uk/~prima

Abstract

This paper describes a complete approach to the segmentation and extraction of text from Web images for subsequent recognition, to ultimately achieve both effective indexing and presentation by non-visual means (e.g., audio). The method described here (the first in the authors' systematic approach to exploit human colour perception) enables the extraction of text in complex situations such as in the presence of varying colour (characters and background). More precisely, in addition to using structural features, the segmentation follows a split-and-merge strategy based on the Hue-Lightness-Saturation (HLS) representation of colour as a first approximation of an anthropocentric expression of the differences in chromaticity and lightness. Character-like components are then extracted as forming textlines in a number of orientations and along curves.

1 Introduction

Web document designers frequently create text in image form (headers, titles, banners etc.) on Web pages, as an attempt to overcome the stylistic limitations of HTML. This text, however, has a potentially high semantic value in terms of indexing and ranking (for search engine query results) the corresponding Web pages. As current search engine technology does not allow for text extraction and recognition in images (see [1] for a list of indexing and ranking criteria for different search engines), the text in image form is ignored.

Not being able to access the text embedded in images can be a serious matter since, according to a study carried out by the authors [2], of the total number of words visible on a Web page, 17% are in image form (most often semantically important text). Worse still, 76% of these words in image form do not appear elsewhere in the encoded (e.g. ASCII or UNICODE) text. These results agree with earlier findings [3] and clearly indicate an alarming situation that does not seem to be improving.

Another significant goal is to obtain a uniform representation (e.g. UNICODE) of all visible text on a Web page. This uniform representation can be used by a number of applications such as voice browsing [4] and automated content analysis [5] for viewing on small screen devices such as PDAs and mobile (cell) phones.

There has been a provision for specifying the text included in images, in the form of ALT tags in HTML. However, the same study mentioned earlier [2], assessing the impact and consequences of text contained in images, indicates that the ALT tag strategy is not effective. It was found that the textual description (ALT tags) of 56% of images on Web pages was incomplete, wrong or did not exist at all.

It can be seen from the above that there is a significant need for methods to extract and recognise the text in images on Web pages. However, this is a challenging problem for the following reasons. First, these (sometimes complex) colour images tend to be of low resolution (usually just 72 dpi) and the font-size used for text is very small (about 5pt–7pt). Such conditions clearly pose a challenge to traditional OCR, which works with 300dpi images (mostly bilevel) and character sizes of usually 10pt or larger. Moreover, images on Web pages tend to have various artefacts due to colour quantization and lossy compression [6].

Previous attempts to extract text from Web images mainly assume that the characters are of uniform (or almost uniform) colour, work with a relatively small number of colours (reducing the original colours if necessary) and restrict all their operations in the RGB colour space [7][8][9].

This paper proposes a complete approach to extract characters of non-uniform colour and in more complex situations (e.g., see Fig. 1). It argues that the RGB colour space representation is not suited to the extraction of text from Web images and adopts a segmentation method based on analysing differences in chromaticity and lightness that are closer to how humans perceive distinct objects. This is the authors' first approach among a number of alternatives in their on-going pursuit of different ways to address this problem by exploiting human colour perception.

The whole approach comprises two main stages: *segmentation* and *text extraction*. The aim of the

segmentation method is to partition an image into disjoint regions, in such a way that pixels belonging to character-like components are separated from those of the background. The text extraction stage that follows it classifies the segmented regions as text/non-text.

The text segmentation method is described in the following section, while the text extraction (textline identification) method that follows segmentation is presented in Section 3. Results are presented for both the segmentation and the text extraction method and discussed in Section 4.

Figure 1. Sample Web images, containing (gradient) text over a multicoloured background. Originally reproduced in colour.

2 Split-and-Merge segmentation

In this step, character-like components are identified as distinct regions with separate chromaticity and/or lightness by first performing a layer decomposition of the image as a result of histogram analysis of Hue and Lightness in the HLS colour space. The HLS colour space is chosen since the factors that enable humans to perform (chromatic) colour differentiation are mainly the wavelength separation between colours (expressed by Hue differences), the colour purity of the colours involved (expressed by Saturation) and the perceived luminance of the colours involved (expressed by Lightness). Moreover, biological information available for Wavelength, Colour Purity and Lightness discrimination is used in connection to the HLS image data to direct the way mergers occur during the component aggregation stage.

The first operation performed by the method is a conversion of the RGB data stored in the image file into the HLS representation. Following this, the Split-and-Merge method performs segmentation in three steps:

Pre-processing. The image is split in two layers, one containing the *chromatic* pixels (i.e. those for which a dominant wavelength can be identified, such as red, green, blue, purple etc.) and a second containing the *achromatic* (black, white and shades of grey) ones. To perform this separation, biological information on the amount of pure Hue needed to be added to white before the Hue becomes detectable is used [10][11].

Splitting stage. The subsequent splitting process attempts to identify areas of similar (as humans perceive it) colour in the image. For the pixels of the *achromatic* layer the histogram of *Lightness* is computed, and peaks are identified. These peaks are analysed and certain pairs of (adjacent) peaks are combined if the Lightness values spanned by the peaks are deemed to be perceived as 'similar' by a human observer. Lightness value similarity in this case is defined based on the results of experiments designed and conducted by the authors, which determined the least noticeable (by humans) lightness differences. These results broadly agree with the biological information available about least noticeable luminance differences [11]. For each peak identified (after all groupings have taken place), the pixels in the image that have Lightness values under the peak are exported to a separate sub-layer.

In a similar manner, the histogram of the Hue values is computed for the pixels of the chromatic layer and peaks are identified. Two adjacent peaks are combined here if the Hue values spanned by the peaks are deemed to be perceived as 'similar' by a human observer. Similarity here is defined based on biological information available for wavelength discrimination [11]. The chromatic layer is thus split into sub-layers of different Hues (each layer containing the range of hues under each of the final peaks).

For each of the sub-layers produced, the Lightness histogram is then computed, peaks are identified and the peak analysis process is repeated. Peaks are suitably combined and new image sub-layers are created for pixels with Lightness values in the ranges under each of the final peaks. The splitting process can be terminated early if only one peak can be identified in the histogram analysed and, therefore, splitting cannot produce more than one sub-layer. Following this process, a tree of layers is produced, where the original image is the root of the tree and the layers produced are the nodes.

Merging stage. After the splitting process is finished, a bottom-up merging process takes place. Connected components are first identified in each of the bottom (leaf) layers. The neighbouring pixels (in the original image) of each connected component are then examined, and if similar to the colour of the component, they are flagged as a potential extension for it. The similarity measure depends on the type of layer the analysis is performed in. If the layer in question is the result of Hue histogram analysis, then Hue (wavelength) discrimination data is used to assess if a viewer is able to differentiate between the Hue of the component and the Hue of the neighbouring pixels. Similarly, if the layer in question was produced by splitting based on the Lightness histogram, Lightness discrimination data is used. At the end of this

process, connected components have been identified in each of the bottom layers, along with their potential extensions (referred to as *vexed areas* in the following).

Starting with the bottom layers, the overlapping of pairs of components (and their vexed areas) is computed and, if greater than a specified threshold, the two components are merged into a new component (with a new vexed area). After this process finishes at the bottom layers, the resulting components are copied one level up, and their vexed areas are refined according to the type of the layer they are copied into (taking into account either Hue or Lightness discrimination data). Then the same process of component aggregation based on overlapping is performed and the process continues, working its way towards the root of the tree. The merging process stops when the layer corresponding to the original image is reached. At that point, the desired result will be that characters in the image are described by connected components not containing parts of the background.

3 Text Extraction

The vast number and the variety of connected components produced by segmenting Web images, hinder any feature-based attempt to classify connected components as character/non-character. Instead of classifying individual components as representing either characters or parts of the background, the proposed method aims at identifying groups of collinear components as potential text lines. The rationale for the text-line based character identification is that a set of similar components arranged as a potential textline will most probably correspond to text.

It should be pointed out at this point that the characters in any Web image are not necessarily placed along straight lines. The method can cope with curved textlines but there is a trade-off to be kept in mind between the maximum curvature allowed and detection accuracy.

3.1 Text Line Extraction

The first step of the classification method proposed is to group the connected components produced by the segmentation process according to their size. The size metric used to group the connected components is the length of the bounding box diagonal, since it is orientation-independent. The range of each size-group was defined based on an average diagonal value as follows. The minimum and maximum diagonals were measured for different fonts (Arial and Times typefaces, normal, bold, italics) and various sizes (6 to 36pt), and the factor (f) was computed so that given an average diagonal value (D_{av}), the size thresholds to include a component to the size group are $D_{min} = D_{av}/f$ and $D_{max} = D_{av} \cdot f$. The average value obtained for f is *1.46*.

For each of the components belonging to a size group, the centre of gravity is calculated. A Hough transform is performed on the positions of the components in the Web image (the computed centres of gravity). The cell (or cells) of the accumulator array with the maximum count is identified and the respective components extracted as a possible text line. The Hough transform is repeated for the remaining components, and a second line is identified. The process continues until no cell exists with a count of more than three.

Fig. 2. A curved text line would be identified as a number of shorter straight lines.

Three is the minimum number of components required for a line to be identified. The rationale behind this decision is twofold. First, geometrically at least three points are needed to be able to assess the co-linearity between them. Statistically, three points would be just enough to give an indication, but not to define with certainty that there is a (text) line there. Nevertheless, there are cases where single words are found alone in a text line, and words comprising three (or even less) characters are very common. The second, and probably most important reason is that we need to be able to address cases where text is not actually written on straight lines. By setting a low threshold (number of collinear points), even if words were written along a curve, straight lines of three characters would be possible to identify (Fig. 2).

3.2 Assessment of Lines

Two mechanisms were devised for assessing the lines extracted by the previous operations. The first mechanism examines the distances between successive components and produces a higher confidence value if the components have equal distances between them.

The second mechanism devised for assessing the lines, uses the projection profile of the components along the direction of the line identified, and examines whether this projection is structurally similar to the projection profile expected from a real textline.

4 Results

In order to evaluate the segmentation and component classification methods described here, a dataset of images collected from a variety of representative web pages was used. The dataset comprises 115 images, of varying size, colour content, and spatial resolution, all of which contain some text.

The images in the dataset were grouped into four categories according to the colour combinations used. Category A holds 14 images that contain multicolour characters over multicolour background. Category B contains 15 images that have multicolour characters over single-colour background. Category C has 37 images with single-colour characters over multicolour background. Finally, Category D holds 49 images with single-colour characters rendered over single-coloured background.

The segmentation method was evaluated based on all the images contained in the dataset. The evaluation of the method was performed by visual inspection (since no precise ground-truth is available, or easy to construct). Each character contained in the images of the dataset was characterised as *identified*, *merged*, *broken* or *missed*. Identified characters are those that are described by a single component. Broken ones, are the characters described by more than one component, as long as each of those components contain only pixels of the character in question (not any background pixels). If two or more characters are described by a single component, yet no part of the background is merged in the same component, then they are characterised as merged. Finally, missed are the characters for which no component or combination of components exists that describes them completely without containing pixels of the background.

Table 1. Split-and-Merge method results.

Category	Identified	Merged	Broken	Missed
All	69.65%	8.15%	14.63%	7.56%
A	55.83%	0.00%	29.13%	15.05%
B	51.92%	18.46%	25.77%	3.85%
C	75.82%	6.87%	9.16%	8.15%
D	74.24%	8.01%	11.64%	6.11%

The results for the Split-and-Merge segmentation method are shown in Table 1. The method's performance rises substantially when it comes to the relatively more straightforward categories C and D, containing single-colour characters.

To evaluate the text extraction process on its own, without accumulating errors from the segmentation processes, only a subset of the images (31 in total) was used, for which the segmentation processes were able to correctly identify 100% of the characters. All four categories defined before are represented in this set of images.

For this set of images, the text extraction process was run and a number of lines exported and assessed as either text or non-text ones. The total number of components classified as characters were then counted, as well as the number of them that actually represented characters. Measures for recall and precision were calculated based on Eq. 1, were C is the set of connected components that correspond to characters and I is the set of connected components identified as characters by the text extraction process.

$$P = \frac{|C \cap I|}{|I|} \qquad R = \frac{|C \cap I|}{|C|} \qquad \textbf{Eq. 1}$$

The text extraction method achieves 53.4% precision and 88% recall.

References

[1] Search Engine Watch, http://searchenginewatch.com

[2] A. Antonacopoulos, D. Karatzas and J. Ortiz Lopez, "Accessing Textual Information Embedded in Internet Images", *Proceedings of SPIE Internet Imaging II*, San Jose, USA, January 24-26, 2001, pp.198-205.

[3] J. Zhou and D. Lopresti, "Extracting Text from WWW Images", Proceedings of the 4th International Conference on Document Analysis and Recognition (ICDAR'97), Ulm, Germany, August, 1997

[4] M.K. Brown, "Web Page Analysis for Voice Browsing", *Proceedings of the 1st International Workshop on Web Document Analysis (WDA'2001)*, Seattle, USA, September 2001, pp. 59–61.

[5] G. Penn, J. Hu, H. Luo and R. McDonald, "Flexible Web Document Analysis for Delivery to Narrow-Bandwidth Devices", *Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR'01),* Seattle, USA, September 2001, pp. 1074–1078.

[6] D. Lopresti and J. Zhou, "Document Analysis and the World Wide Web", Proceedings of the Workshop on Document Analysis Systems, Marven, Pennsylvania, October 1996, pp. 417–424.

[7] A. Antonacopoulos and F. Delporte, "Automated Interpretation of Visual Representations: Extracting textual Information from WWW Images", Visual Representations and Interpretations, R. Paton and I. Neilson (eds.), Springer, London, 1999.

[8] A. D. Lopresti and J. Zhou, "Locating and Recognizing Text in WWW Images," *Information Retrieval*, vol. 2, pp. 177-206, 2000.

[9] A.K. Jain and B. Yu, "Automatic Text Location in Images and Video Frames", *Pattern Recognition*, vol. 31, no. 12, 1998, pp. 2055–2076.

[10] G. Murch, "Color Displays and Color Science," in *Color and the Computer*, J. H. Durrett, Ed. Orlando, Florida: Academic Press INC., 1987, pp. 1-25.

[11] G. Wyszecki and W.S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd ed. New York: John Wiley & sons, 2000.