Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR2003), Edinburgh, UK, August 2003, IEEE Computer Society Press, pp. 131-136.

Two Approaches for Text Segmentation in Web Images

D. Karatzas and A. Antonacopoulos

PRImA Group, Department of Computer Science, University of Liverpool,
Peach Street, Liverpool L69 7ZF, United Kingdom
http://www.csc.liv.ac.uk/~prima

Abstract

There is a significant need to recognise the text in images on web pages, both for effective indexing and for presentation by non-visual means (e.g., audio). This paper presents and compares two novel methods for the segmentation of characters for subsequent extraction and recognition. The novelty of both approaches is the combination of (different in each case) topological features of characters with an anthropocentric perspective of colour perception— in preference to RGB space analysis. Both approaches enable the extraction of text in complex situations such as in the presence of varying colour and texture (characters and background).

1 Introduction

Web images play a crucial role in bringing visual impact to an otherwise plain text medium. Web page designers regularly create page headers and titles as well as other semantically important textual entities in image form. However, using current technology it is not possible to analyse the text embedded in images on Web pages. This is a significant problem for a number of reasons.

First, search engine crawlers [1] are not able to use the text in images for indexing. More often than not, this text contains key index terms, which in the majority of cases do not appear elsewhere on the page (main text)[2][3]. Second, this inability to access important key terms may hinder the effective ranking of search results.

Another important issue is that with the increasing use of multimedia content on the Web there is no uniform representation of the content of a page in a form that can be analysed in an automated way. Text remains the only medium that can be readily analysed, converted to voice etc. It can be argued that it is desirable, given the richness of text expressions, to obtain a textual representation (natural language) of the content of Web pages. The recognition of text in images is a step towards achieving this representation.

For completeness, it should be mentioned that HTML provides for an alternative text description of an image (using the ALT tag) but this is not used by all search engines either for indexing or for ranking [4]. Furthermore, a recent survey by the authors indicates that a significant proportion of ALT tag descriptions (56%) cannot be relied upon as they are either incorrect or do not exist [3].

Images on the Web differ in many aspects from real scenes and other document images. Especially the images that this paper is concerned with (banners, headers, illustrations etc.), are optimised for viewing on a monitor screen.

Web images have to obey file-size constraints, as downloading speed directly depends on the data volume. As a result, these images tend to be of low resolution (just good enough for display) and tend to be compressed (JPEG standard). More specifically to the images containing text, the following can be observed: The resolution is usually just 72 dpi, and various artefacts are present due to colour quantization and lossy compression. Moreover, the font-size used for text is very small (about 5pt–7pt) [2]. These conditions clearly pose a challenge to traditional OCR, which works with 300dpi images (mostly bilevel) and character sizes of usually 10pt or larger.

There is a small number of existing approaches addressing the problem of character segmentation and extraction. These methods produce good results for relatively simple images, but fail when more complex images are encountered for the following reasons. These approaches mostly deal with a very small number of colours (they do not work on full-colour – e.g., JPEG – images). They also assume a practically constant and uniform colour for text [5, 6] and fail when this is not the case. In practice, there are many situations where gradient or multicolour text is present (see Fig 1). The situation where dithered colours are present (especially in GIF images) has received some attention [7, 8] but such colours can only be found in a relative small number of Web images. Furthermore, the background may also be complex (in terms of colour) so that the assumption that it is the largest area of (almost) uniform colour in the image [9] does not necessarily hold. It should be noted that methods that deal with the extraction of text from video (a different field with different image characteristics) also make similar assumptions about the uniformity of the colour of the text (especially caption/credits text).

This paper proposes two methods to extract characters of non-uniform colour and in more complex situations. It argues that the RGB colour space representation is not suited to the extraction of text from Web images and adopts approaches based on analysing differences in chromaticity and lightness that are closer to how humans perceive distinct objects.

In the following section, the text extraction methods are described, each one in a separate subsection. Results are presented, and the two methods are compared and discussed in Section 3, while the paper is concluded by Section 4.

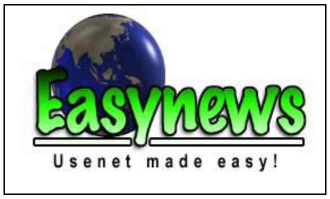

Figure 1. Sample Web Page image, containing gradient text over a (partly) multicoloured background. Originally reproduced in colour.

2 Text segmentation methods

Humans perceive colour differences with certain bias and distinguish colours based on chromaticity and lightness (different from the commonly used (R+G+B)/3) [12].

The premise of both approaches described in this paper is that a method for text extraction in these circumstances should perform based on the analysis of colour differences as humans perceive them and not necessarily as expressed in the RGB space.

The first approach, in addition to using structural features, 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.

The second approach performs a bottom-up aggregation of colour connected-components based on a fuzzy *propinquity* measure that utilises two features: a colour distance in the perceptually uniform L*a*b* colour space and a feature expressing topological properties of character strokes. Each of the approaches is concisely described next. Earlier more detailed (albeit preliminary) individual accounts of the methods are in [13, 14]. Here an updated description of the methods is given primarily to fulfil the main task of this paper which is the presentation of their results and their comparison.

2.1 Split-and-Merge approach

In the first approach, character-like components are identified as distinct regions with separate chromaticity and/or lightness 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, explained in detail below.

2.1.1. Pre-processing. After converting the image data (RGB) to the HLS representation, a pre-processing step is performed, where 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. The importance of this step lies in the fact that any process that involves Hue values will fail if applied to achromatic pixels, since the Hue for those pixels is either very unreliable or undefined (by default set to zero).

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].

2.1.2. 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. The peaks identified are analysed and certain peaks are combined in the following way. For every pair of adjacent peaks, the range of Lightness values spanned by both peaks is examined. If the Lightness value at the left minimum of the left peak is similar to the Lightness value at the right minimum of the right peak the two peaks are combined. Two colours are considered similar (in terms of Lightness) if a human being cannot differentiate between the two. Similarity here is defined based on the results of in-house experiments which determined the least noticeable lightness differences (by humans). These results broadly agree with the biological information available about least noticeable luminance differences [11]. For each peak identified (after all combinations have taken place), the pixels in the image that have Lightness values under the peak are exported to a different sub-layer.

In a similar manner, the histogram of the Hues is computed for the pixels of the chromatic layer and peaks are identified.. Two adjacent peaks are combined here if the Hue value at the left minimum of the left peak is similar to the Hue value at the right minimum of the right peak. 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.

2.1.3. 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.

2.2 Fuzzy approach

The second segmentation method developed operates in a bottom-up way, initially identifying connected components of uniform (as perceived by humans) colour inside a given image, and progressively (selectively) merging them into larger components – ultimately representing the characters in the image. Component aggregation is directed here by a metric of ‘closeness and similarity’ between components called *Propinquity*, which is defined and evaluated within the framework of a fuzzy inference system.

2.2.1. Initial component identification. In order to identify connected components the colour similarity between pixels here is measured with the help of a colour distance metric, which expresses – in essence – whether a human would consider two colours similar or not. It should be noted that colour discrimination in this method is approached in a distinctly different way, not focusing on separate colour properties (e.g. Hue, Lightness and Saturation), but on a single metric for colour similarity.

At this point it should be pointed out that the HLS and RGB colour spaces lack a straightforward measurement method for perceived colour difference. This is due to the fact that colours having equal (Euclidean) distances in those colour spaces may not necessarily be perceived by humans as being equally dissimilar. To address this problem the perceptually uniform $L^*a^*b^*$ colour space is used here. Furthermore, by definition, the Euclidean distance between two colours in the $L^*a^*b^*$ colour system corresponds to the perceived colour difference between the colours.

Neighbouring pixels of similar colour (with difference less than an experimentally derived threshold $\Delta E^*=20$) are grouped into (colour) connected components.

2.2.2. The fuzzy inference system. Having identified the initial set of connected components and in order for the component aggregation process to start, the *Propinquity* between each pair of components must be calculated. The Propinquity is defined as the single output of a fuzzy inference system, which combines two inputs: a *Colour*

Distance metric, and a metric expressing the topological relationship between two components (called the *Connections Ratio*).

The Colour Distance metric used as an input for the fuzzy inference system, is the Euclidean distance in the $L^*a^*b^*$ colour space between the average colours of the components involved. Four fuzzy sets have been defined for the Colour Distance input, which express different levels of colour dissimilarity: *Insignificant*, *Small*, *Medium* and *Large*. Components having a Colour Distance in the Insignificant or Small fuzzy sets are considered similar enough to be able to be merged. For components with a Colour Distance in the Medium set the Colour Distance provides no certain indication (as to whether they can be merged or not – both outcomes are possible depending on the Connections Ratio input). Finally, components with Colour Distances in the Large set are never considered for merging.

The second input, the *Connections Ratio*, expresses the topological relationship between two components. A *Connection* is defined here as a link between a pixel and any one of its 8-neighbours, each pixel thus having 8 connections. A connection is called *internal* when both the pixel in question and the neighbouring one belong to the same component, and *external* when the neighbouring pixel belongs to another component. Given any two components *a* and *b*, the Connections Ratio, denoted as $CR_{a,b}$ is defined as :

$$CR_{a,b} = \frac{Ce_{a,b}}{\min(Ce_a, Ce_b)}$$

where $Ce_{a,b}$ is the number of (external) connections of component *a* to pixels of component *b*, and Ce_a and Ce_b refer to the total number of external connections (to all neighbouring components) of each of the components *a* and *b* respectively.

In practical terms, as the Connections Ratio expresses the extent to which components neighbour each other, it can be used as an indication of whether components form parts of continuous character strokes or not. Five fuzzy sets are defined to express the different situations. Components that partially neighbour, will have a Connections Ratio value in the middle range (covered by two fuzzy sets: *Medium Low* and *Medium High*). Large Connections Ratio values (in the *High* fuzzy set) indicate components extensively neighbouring (up to one including the other), which in most of the cases are not parts of the same character and should not be merged. Pairs of components loosely neighbouring have values in the range represented by the *Low* set. Finally, the *Zero* set is defined for components whose Connections Ratio is zero (disjoint components). These are components that do not neighbour at all and, therefore, cannot be merged.

In a similar manner to the two inputs, a number of fuzzy sets and the appropriate membership functions are defined for the Propinquity output. A total of seven fuzzy sets are defined for the output. A *Zero* fuzzy set is defined to facilitate the rejection of certain cases where components should not be merged. On the opposite end, the *Definite* fuzzy set is defined so that cases where components should definitely be merged are awarded a high Propinquity value. The Propinquity output is defined so that a value of 0.5 will be the threshold above which two components should be considered for a merger, while values below 0.5 indicate that two components should not be merged. A *Medium* fuzzy set is defined to cover the middle range of Propinquity values (0.4 to 0.6) and is used to indicate cases where it is not certain whether two components should be merged or not. The rest of the fuzzy sets provide for Propinquity values to be associated with different degrees of confidence on whether two components should be merged or not.

The combination of the two inputs into the single Propinquity output is achieved with the use of a set of rules. These rules favour small Colour Distance values, and Connections Ratio values in the middle range (for reasons explained above). Furthermore, appropriate rules are introduced which ensure that a zero propinquity value (expressed by the *Zero* fuzzy set) is assigned to pairs of components that do not neighbour (Connections Ratio values equal to zero – *Zero* fuzzy set) or are of completely dissimilar colour (high Colour Distance values – *Large* fuzzy set). The surface of Figure 2 shows the mapping from Colour Distance and Connections Ratio inputs to the Propinquity output.

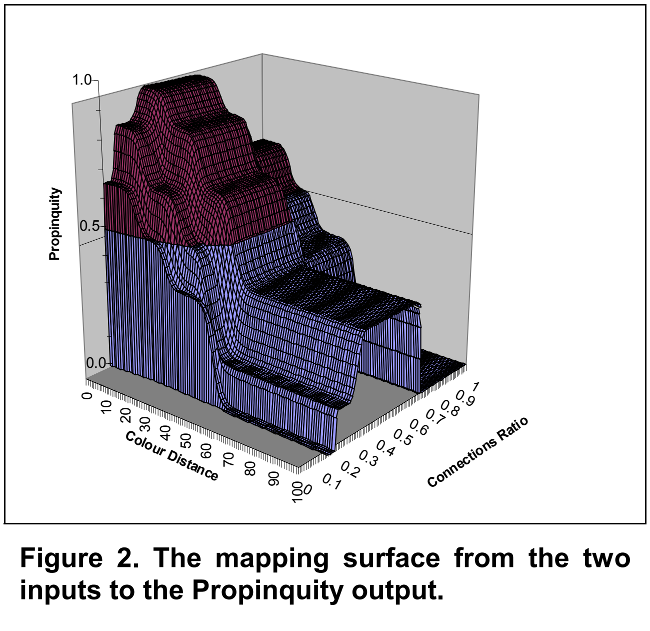

Figure 2. The mapping surface from the two inputs to the Propinquity output.

2.2.3. Component aggregation. At this stage, based on the propinquity value computed for every possible pair of components, a sorted list of possible mergers is created. As long as the Propinquity value associated with the first merger in the list is higher than the threshold of 0.5, the merger in question is performed, and the list is updated accordingly. The component aggregation process finishes when no pair of components exist with a Propinquity value above the specified threshold. Similarly to the split-and-merge method (Section 2.1), the desired result at the end of the process will be that characters in the image are described by connected components not containing parts of the background.

3 Results and discussion

In order to evaluate the methods described here, a dataset of images collected from a variety of Web pages was used. The dataset used comprises 115 images, of varying size, colour complexity and text content, representative of the way image text is used in Web pages. The images in the dataset were grouped into four categories according to the colour combinations (text/background) present. Category A holds 14 images that contain multicoloured characters over multicoloured background. Category B contains 15 images that have multicoloured characters over monochrome background. Category C has 37 images with monochrome characters over multicoloured background. Finally, category D holds 49 images with monochrome characters over monochrome background. Whether the characters (or background) is monochrome or multicoloured is assessed by visual inspection. The distribution of images in the four categories reflects the occurrence of images in Web Pages.

The aim of the segmentation process is to partition the image in question into disjoint regions, in such a way that the text is separated from the background. Furthermore, ideally, characters should not be split into sub-regions or be merged with other characters. Since ground truth information is not available for the images of the dataset (a significant project on its own), the evaluation of the segmentation methods was performed by visual inspection of the final segmentation results. For each character contained in the image, the observer decides whether it has been either identified as a single component (correctly identified), split in multiple components, merged with one or more other characters, or missed (not correctly separated from the background). This assessment can be subjective since the borders of the characters are not precisely defined in most of the cases (due to anti-aliasing, artefacts introduced by compression etc).

The correct identification results for each category of the dataset are summarised in Figure 3. The Split-and-Merge segmentation method was able to correctly identify 55.83% of the characters in category A, 51.92% in category B, 75.82% in category C, and 74.24% in category D. The above results reflect the increasing difficulty in categories where the text is multicoloured. The Fuzzy segmentation method was able to correctly identify 59.22% of the characters in category A, 69.23% in category B, 70.67% in category C, and 71.62% in category D. Overall, both methods achieve a percentage of correctly identified characters that is approximately 69%.

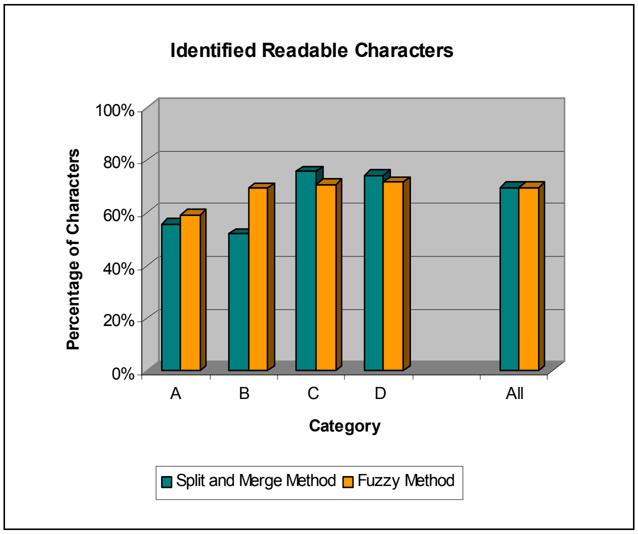

Figure 3. Character identification performance comparison for the two segmentation methods.

The Fuzzy segmentation method appears to cope better with images of the first two categories (multicoloured characters) than the Split-and-Merge method. On the other hand, the Split-and-Merge method gives better results for images of categories C and D (monochrome characters).

In terms of efficiency, the Fuzzy segmentation method tends to outperform the Split-and-Merge one as the latter has to work across a number of images (the different sub-layers). Furthermore, the Fuzzy segmentation method more-often-than-not deals with a smaller number of larger components during merging, whereas the Split-and-Merge method tends to produce a large number of smaller components prior to merging

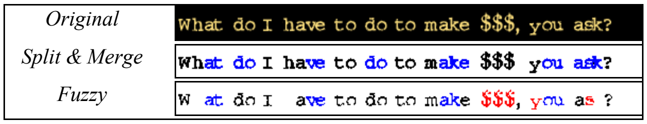
Original
Split & Merge
Fuzzy
What do I have to do to make $$$, you ask?
What do I have to do to make $$$ you ask?
W at do I ave to do to make $$$, you as ?

Figure 4. Example of image with multicoloured text over monochrome background.

Overall, the Split-and-Merge method exhibits a tendency to merge medium-sized components, in contrast to the Fuzzy method, which is more conservative. Although in many cases this tendency results in better final segmentations (e.g. Figure 4), there are cases (e.g. Figure 5) where unwanted mergers between characters take place. In the examples shown here, identified characters are shown in red colour, split characters in red and merged characters in blue.

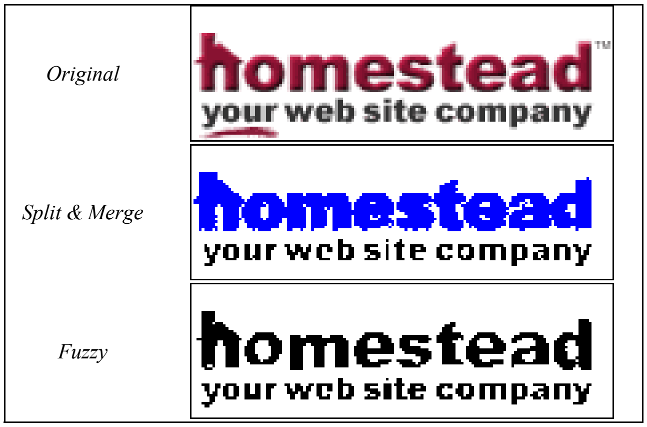

Figure 5. Example of image with multicoloured text over monochrome background.

On the other hand, the Fuzzy segmentation method seems to deal better with small characters than the Split-and-Merge method. This fact manifests itself in two ways. First, the final segmentation produced by the Fuzzy method is generally much "cleaner" in the sense that not many small components (noise) are left, rather they are merged into larger components. The second and most important fact is that small characters (having width or height less than 4-6 pixels) are much better segmented by the Fuzzy method as can be seen in the example of Figure 6.

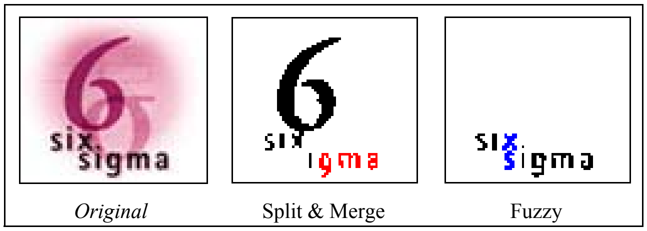

Figure 6. Example of image containing small characters over multicoloured background.

4 Conclusion

This paper presented and compared two methods for the segmentation of text embedded in images on Web pages, which was argued is a significant problem to be solved. One of the key characteristics of both methods is the emulation (approached from different angles in each case) of the way humans perceive differences in colour.

Given the difficulty presented by the nature of the images in question, both methods can be said to perform well, the final effectiveness of course will be judged by the target application domain. Work continues on optimising the methods' performance while the recognition of the text will be the next open problem with a significant number of hurdles to overcome.

References

[1] D. Amor, *The E-Business (R)evolution*, Prentice Hall, 1999.

[2] 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.

[3] 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 2001, Vol. 4311, pp. 198–205.

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

[5] 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

[6] 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.

[7] J. Zhou, D. Lopresti, and T. Tasdizen, "Finding Text in Color Images," proceedings of the IS&T/SPIE Symposium on Electronic Imaging, San Jose, California, pp. 130-140, 1998.

[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.

[12] R.W.G. Hunt, *Measuring Colour*, John Wiley & Sons, 1987.

[13] A. Antonacopoulos and D. Karatzas, "An Anthropocentric Approach to Text Extraction from WWW Images", *Proceedings of 4th IAPR International Workshop on*

Document Analysis Systems (DAS2000), Rio de Janeiro, Brazil, December 2000, pp. 515–525.

[14] A. Antonacopoulos and D. Karatzas, “Fuzzy Segmentation of Characters in Web Images Based on Human Colour Perception”, in the book *Document Analysis Systems V*, D. Lopresti, J. Hu and R. Kashi (Eds.), Springer, LNCS 2423, pp. 295–306.