in the book: "Web Document Analysis: Challenges and Opportunities", A. Antonacopoulos and J. Hu (Eds.), Series in Machine Perception and Artificial Intelligence, World Scientific Publishing Co., 2003, pp. 203-221

CHAPTER 11

A FUZZY APPROACH TO TEXT SEGMENTATION IN WEB IMAGES BASED ON HUMAN COLOUR PERCEPTION

A. Antonacopoulos and D. Karatzas

PRImA (Pattern Recognition and Image Analysis) Group,
Department of Computer Science, University of Liverpool,
Liverpool, L69 7ZF, United Kingdom
E-mail:{aa, karatzas}@csc.liv.ac.uk
URL: http://www.csc.liv.ac.uk/~prima

This chapter describes a new approach for the segmentation of text in images on Web pages. In the same spirit as the authors' previous work on this subject, this approach attempts to model the ability of humans to differentiate between colours. In this case, pixels of similar colour are first grouped using a colour distance defined in a perceptually uniform colour space (as opposed to the commonly used RGB). The resulting colour connected components are then grouped to form larger (character-like) regions with the aid of a propinquity measure, which is the output of a fuzzy inference system. This measure expresses the likelihood for merging two components based on two features. The first feature is the colour distance between the components, in the $L^*a^*b^*$ colour space. The second feature expresses the topological relationship of two components. The results of the method indicate a better performance than previous methods devised by the authors and possibly better (a direct comparison is not really possible due to the differences in application domain characteristics between this and previous methods) performance to other existing methods.

1. Introduction

In a typical Web document, there are significant discrepancies between the text appearing in the *view* of the document (what *actually appears* in the browser window) and the text contained in the *code* of the document (the file containing markup language tags, program instructions and various types of text). A major (and very frequently occurring) discrepancy is that some of the visible text in the view of the document is actually embedded in images. In such cases, there is no direct correspondence between the code (an instruction to display a given image) and the text contained in that image. The human reader, of course, can read all

the text on the screen (document view), whether this text exists in the code or not. From this point on, visible text that is contained in the code will be referred to as *encoded text*, while text that is embedded in images will be referred to as *image text*.

The difference between encoded text and image text can be seen by contrasting the example of Fig. 1, where both encoded and image text is shown, and that of Fig. 2, where the image text is missing (note that there is also no alternative text for the images—see below).

The above discrepancy between the code and the view representations of a Web document is potentially very significant. The origins of the problem are twofold. First, Web document designers create image text as a way of overcoming the limitations of the markup language used in the code. Second, due to limitations of current technology, image text is not accessible to any automated process or analysis performed on the document. Both of these interrelated issues are examined next in more detail.

Image text is created for two main reasons. The first is one of necessity as the markup language (HTML in this case) cannot adequately display textual entities such as mathematical equations, text in diagrams and charts etc. The second and main reason is that document creators wish to add impact to certain textual entities such as titles, headings, buttons etc. The effects applied to the text and its background (e.g., complex colour combinations, unusual fonts, image as background etc.) are such that cannot be expressed in the markup language.

Not having all the visible text in the code of the document means that a proportion of the text seen by the human reader (image text) is not available for any automated analysis. Such analysis includes essential processes, fundamental to the modus operandi of the Web, such as automated indexing by search engines. Currently, as search engine technology does not allow for text extraction and recognition in images (see the Search Engine Watch website[1] for a list of indexing and ranking criteria for different search engines), the image text is ignored. Moreover, the problem of indexing is compounded by the fact that it is precisely the semantically important text (titles, headings etc.) that is most often required to make a visual impact and, therefore, represented as image text.

The lack of a uniform representation of the text impacts negatively on several other possibilities for exploiting the Web. If all the visible text was available as encoded text, it would be possible to perform accurate voice browsing,[2] for instance. One could listen to the Web document read to them instead of having to look at a monitor. Such a possibility will enable browsing in the car or via the telephone and also will benefit visually impaired people.

Fig. 1. A Web Page with images shown.

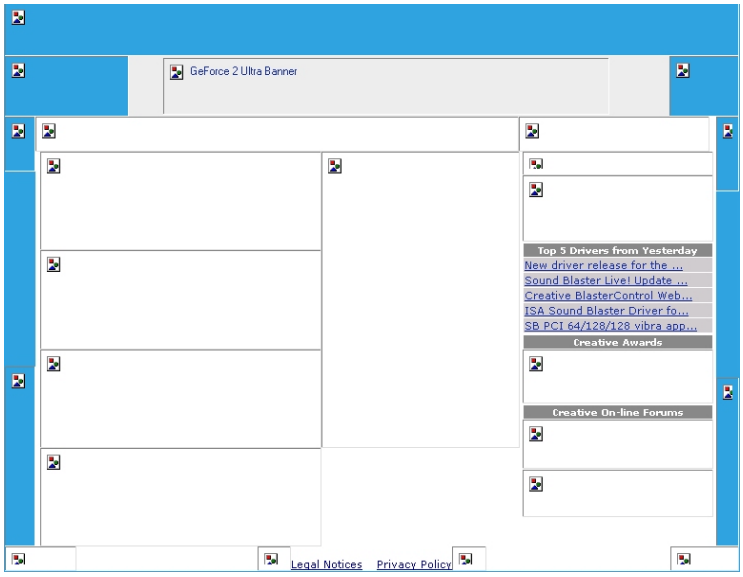

Fig. 2. The same Web Page without the images.

Another major application area is the analysis of the content of a Web document for filtering, summarisation and display (also referred to as *adaptive content delivery*) on small form-factor devices such as PDAs and mobile phones.[3]

It should be noted that there has been a provision for specifying an encoded version of the text embedded in images, in the form of ALT tags in HTML. However, a study conducted by the authors,[4] 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. This can be a serious matter since, 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 text. These results agree with earlier findings[5] and clearly indicate an alarming trend.

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 unique and challenging problem. One would immediately attempt to draw on the similarities between the segmentation/recognition of characters in web images and established (albeit far from a solved problem) technologies such as those employed by traditional OCR. It could even be argued that as image text is created and displayed electronically (no digitisation artefacts) traditional OCR would have one less obstacle. Nevertheless, there are significant issues that make the analysis of image text a significantly more difficult problem than that faced by traditional OCR. The most important of these issues are examined next.

Perhaps the single most important realisation is that image text is created with the goal of minimizing its transmission/rendering time while being good enough to view on a monitor screen. The implications are quite significant in terms of quality. First, the (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, image text tends to have various artefacts that are not encountered in the analysis of traditional document images. One of the problems affecting most the process of differentiating the foreground from the background is the presence of anti-aliasing. The resulting smooth transition from background to foreground colours produces characters with poorly defined edges, in contrast to the characters in typical document images. Another significant problem is the loss of colour information and ‘blocky’ appearance of regions resulting from

compressing the images with a lossy scheme (e.g. JPEG). Furthermore the colour of individual characters or areas of the background is rarely uniform due to either intentional effects (e.g. gradient colour) or colour quantization artefacts (originating from the software producing the image text).[6] Finally, there is a pronounced difference in creativity expressed through image text (rather than in traditional documents) in the form of a very wide variety of fonts, colour combinations, complex backgrounds, 3D effects on characters and so on.

It should be mentioned that text in Web images is of different nature than text in video, for instance. In principle, although methods attempting to extract text from video (e.g., Li *et al.*[7]) could be applied to a subset of Web images, they make restricting assumptions about the nature of embedded text (e.g., colour uniformity). As such assumptions are, more often than not, invalid for text in Web images, such methods are not directly discussed here.

The significant variety (in every aspect) of Web images containing text has led most researchers to limit their methods to subsets of Web images that conform to certain assumptions. For instance, Zhou and Lopresti[8] only deal with 8-bit palletised images, where text appears in uniform colour. Antonacopoulos and Delporte[9] deal with 24-bit images as well, but also require text to be of uniform colour. Jain and Yu[10] require that the background is of uniform colour, in addition to being the largest component of the image. Moreover, previous attempts to extract text from Web images work with a relatively small number of colours and restrict all their operations in the RGB colour space.

A novel method that is based on information on the way humans perceive colour differences has recently been proposed by the authors.[11] That method works on full colour images and uses different colour spaces in order to approximate the way humans perceive colour. It comprises the splitting of the image into layers of similar colour by means of histogram analysis and the merging of the resulting components using criteria drawn from human colour discrimination observations.

This paper describes a new method for segmenting character regions in Web images. In contrast to the authors' previous method,[11] it is a bottom-up approach. This is an alternative method devised in an attempt to emulate even closer the way humans differentiate between text and background regions. Information on the ability of humans to discriminate between colours is used throughout the process. Pixels of similar colour (as humans see it) are merged into components and a fuzzy inference mechanism that uses a 'propinquity' measure is devised to group components into larger character-like regions.

The colour segmentation method and each of its constituent operations are examined in the next section and its subsections. Experimental results are presented and discussed in the subsequent section, concluding the paper.

2. Colour Segmentation Method

The basic assumption of this paper is that, in contrast to other objects in general scenes, text in image form can always be easily separated (visually, by humans) from the background. It can be argued that this assumption holds true for all text, even more so for text intended to make an impact on the reader. The colour of the text in Web images and its visual separation from the background are chosen by the designer (consciously or subconsciously) according to how humans perceive it to 'stand out'.

To emulate human colour differentiation, a colour distance measure is defined in an alternative colour space, rather than in the 'standard' RGB space (a key argument in this and previous work of the authors is that the perceptual non-uniformity of the RGB colour space makes it less attractive to work in – see below). The distance measure is used first to identify colour connected components and then, combined with a new topological feature (using a fuzzy inference system), it is used to aggregate components into larger entities (characters).

Each of the processes of the system is described in a separate subsection below. First, the colour measure is described in the context of colour spaces and human colour perception. The connected components labelling process using this colour distance is described next. The two features (colour distance and a measure of spatial proximity) from which the new 'propinquity' measure is derived are presented in Section 2.3. Finally, the fuzzy inference system that computes the propinquity measure is the subject of Section 2.4 before the description of the last stage of colour connected component aggregation (Section 2.5).

2.1. *Colour Distance*

To model human colour perception in the form of a colour distance measure, requires an examination of the different colour spaces in terms of their perceptual uniformity. The *RGB* colour system, which is by far the most frequently used system in image analysis applications, lacks a straightforward measurement method for *perceived* colour difference. This is due to the fact that colours having equal distances in the *RGB* colour space may not necessarily be perceived

by humans as having equal distances.[a] A more suitable colour system would be one that exhibits perceptual uniformity. The CIE (Commission Internationale de l'Eclairage) has standardised two colour systems ($L^*a^*b^*$ and $L^*u^*v^*$) based upon the CIE *XYZ* colour system.[12,13] These colour systems offer a significant improvement over the perceptual non-uniformity of *XYZ*[14] and are a more appropriate choice to use in that aspect than *RGB* (which is also perceptually non-uniform, as mentioned before).

The measure used to express the perceived colour distance in the current implementation of this method is the Euclidean distance in the $L^*a^*b^*$ colour space ($L^*u^*v^*$ has also been tried, and gives similar results). In order to convert from the *RGB* to the $L^*a^*b^*$ colour space, an intermediate conversion to *XYZ* is necessary. This does not, at first, appear to be a straightforward task, since the *RGB* colour system is by definition hardware-dependent, resulting in the same *RGB*-coded colour being reproduced on each system slightly differently (based on the specific hardware parameters). On the other hand, the *XYZ* colour system is based directly on characteristics of human vision (the spectral composition of the *XYZ* components corresponds to the colour matching characteristics of human vision) and therefore designed to be totally hardware-independent. In reality, the vast majority of monitors conform to certain specifications, set out by the standard *ITU-R recommendation BT.709*,[15] so the conversion suggested by *Rec.709* can be safely used and is the one used for this method. The conversion from *XYZ* to $L^*a^*b^*$ is straightforward and well documented.

2.2. *Colour Connected-Component Identification*

Colour connected-component labelling is performed in order to identify components of similar colour. These components will form the basis for the subsequent aggregation process (see Section 2.5). It should be noted that although the aggregation process that follows would still work with pixels rather than connected components as input, using connected components significantly reduces the number of mergers and subsequently the computational load of the whole process.

The idea behind this pre-processing step is to group pixels into components, if and only if a human being cannot discriminate between their colours. The rationale at this stage is to avoid wrong groupings of pixels as— this is true for

[a] For example, assume that two colours have RGB (Euclidean) distance d. Humans find it more difficult to differentiate between the two colours if they both lie in the green band than if the two colours lie in the red-orange band (with the distance remaining d in both cases). This is because humans are more sensitive to the red-orange wavelengths than they are to the green ones.

all bottom-up techniques— early errors have potentially significant impact on the final results.

The identification of colour connected-components is performed using a one-pass segmentation algorithm adapted from a previously proposed algorithm used for binary images.[16] For each pixel, the colour distance to its adjoining (if any) connected components is computed and the pixel is assigned to the component with which the colour distance has the smallest value. If the pixel in question has a distance greater than a threshold to all its neighbouring connected components, a new component is created from that pixel.

The threshold below which two colours are considered similar was experimentally determined and set to *20* in the current implementation. In fact, it was determined as the maximum threshold for which no character was merged with any part of the background. It should be noted, since the images in the training data set include cases containing text very similar to the surrounding background in terms of hue, lightness or saturation, this threshold is believed to be appropriate for the vast majority of text in Web images. Finally, the chosen threshold is small enough to conform to the opening statement that only colours that cannot be differentiated by humans should be grouped together.

2.3. *Propinquity Features*

The subsequent aggregation of the connected components produced by the initial labelling process into larger components is based on a fuzzy inference system (see next section) that outputs a *propinquity* measure. This measure expresses how close two components are in terms of colour and topology.

The propinquity measure defined here is based on two features: a colour similarity measure and a measure expressing the degree of 'connectivity' between two components. The colour distance measure described above (Section 2.1) is used to assess whether two components have perceptually different colours or not.

The degree of connectivity between two components is expressed by the *connections ratio* feature. 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 can be either *internal* (i.e., both the pixel in question and the neighbour belong to the same component) or *external* (i.e. the neighbour is a pixel of another component). Figure 3 illustrates the external and internal connections of a given component to its neighbouring components.

Given any two components *a* and *b*, the connections ratio, denoted as $CR_{a,b}$, is defined as

$$CR_{a,b} = \frac{Ce_{a,b}}{\min(Ce_a, Ce_b)} \qquad \textit{Eq. 1}$$

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 components *a* and *b*, respectively. The connections ratio is therefore the number of connections between the two components, divided by the total number of external connections of the component with the smaller boundary (it follows that $Ce_{a,b} = Ce_{b,a}$). The connections ratio ranges from *0 – 1*.

In terms of practical significance, the connections ratio is far more descriptive of the topological relationship between two components than other spatial distance measures (e.g., the Euclidean distance between their centroids). A low connections ratio indicates loosely linked components, a medium value indicates components connected only at one side, and a high connections ratio indicates that one component is almost included in the other. Moreover, the connections ratio provides a direct indication of whether two components are neighbouring or not in the first place, since it will equal zero if the components are disjoint.

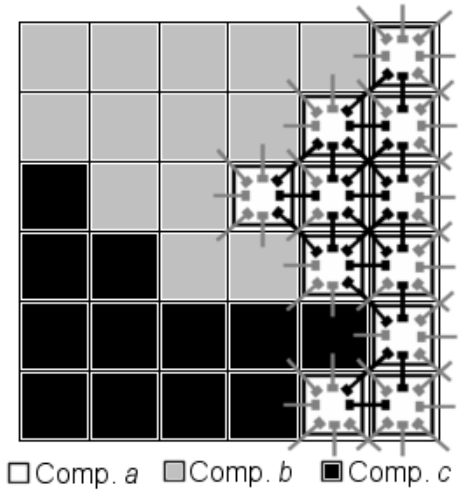

Fig. 3. Illustration of connected-components and their connections.

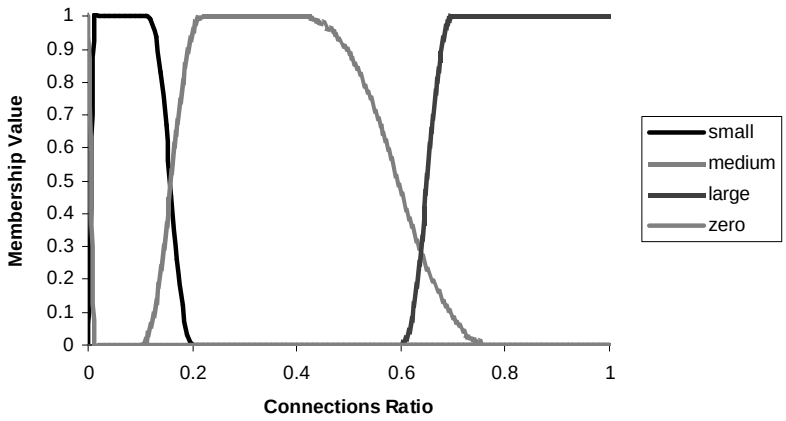

Fig. 4. Membership functions for the connections ratio input.

2.4. *Fuzzy Inference*

A fuzzy inference system has been designed to combine the two features described above into a single value indicating the degree to which two components can be merged to form a larger one. The $L^*a^*b^*$ colour distance and the connections ratio described in the previous sections form the input to the fuzzy inference system. The output, called the *propinquity* between the two participating components, is a value ranging between zero and one, representing how close the two components are in terms of their colour and topology in the image. Each of the inputs and the output are defined using a number of fuzzy sets and corresponding membership functions, described below. In the fuzzy inference system, the relationship between the two inputs and the output is defined with a set of rules, also explained below.

The rationale in defining the fuzzy sets and function for the connections ratio input to the fuzzy inference system is the following. The components that should be combined are those that correspond to parts of characters. Due to the fact that characters consist of continuous strokes, the components in question should only partially touch each other (i.e. one should not be contained in the other nor they should be disjoint). For this reason, a membership function is defined on a *medium* fuzzy set ranging between *0.15* and *0.65*. It is considered advantageous for two components to have a connections ratio that falls in that range in order to

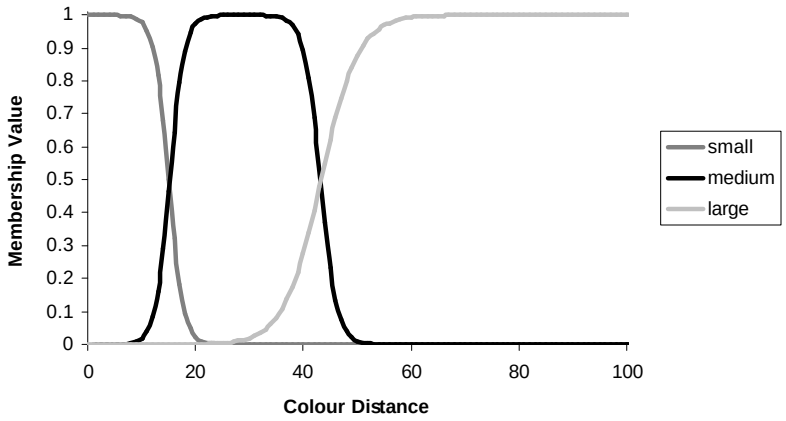

Fig. 5. Membership functions for the colour distance input.

combine them. This fact is enforced by the rules comprising the fuzzy inference system, which favour a connections ratio in the *medium* region, rather than one in the *small* or *large* regions (ranging between *0* and *0.15*, and *0.65* and *1*, respectively). Furthermore, a membership function called *zero* is defined, in order to facilitate the different handling of components that do not touch at all, and should not be considered for merging. The fuzzy sets and membership functions defined for the connections ratio input can be seen in Fig. 4.

There are three fuzzy sets and corresponding membership functions defined for the $L^*a^*b^*$ colour distance input, namely *small*, *medium* and *large* (see Fig. 5). The *small* membership function is defined on a fuzzy set ranging between *0* and *15*. Colours having an $L^*a^*b^*$ distance less than *15* cannot be discriminated by humans, therefore a colour distance falling in the *small* range is being favoured by the rules of the fuzzy inference system. In contrast, a membership function has been defined on a *large* fuzzy set for colour distances above *43*. Components having a colour distance in that range are considered as the most inappropriate candidates to be merged. The middle range, described by the *medium* fuzzy set and membership function, is where there is no high confidence about whether two components should be merged or not. In that case, the rules of the system give more credence to the connections ratio input. The thresholds of *15* and *43* were experimentally determined, as the ones that minimise the number of wrong mergers.

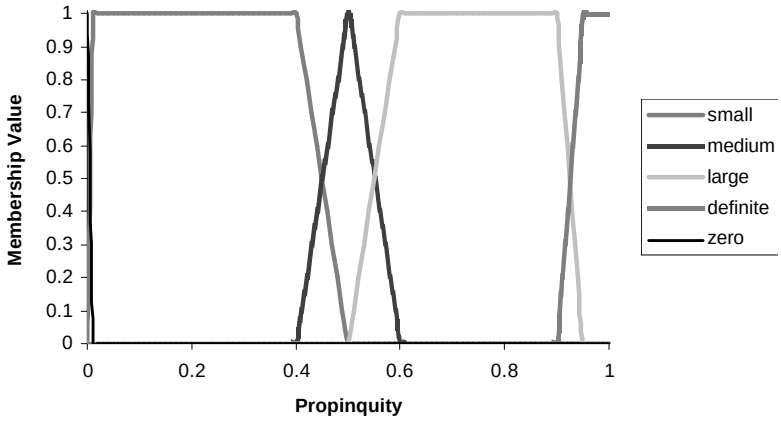

Fig. 6. Membership functions for the propinquity output.

The single output of the fuzzy inference system, the propinquity, is defined using five fuzzy sets and corresponding membership functions (see Fig. 6). There are two membership functions defined on fuzzy sets at the edges of the possible output values range, namely *zero* and *definite*, and three membership functions defined on fuzzy sets in middle range: *small*, *medium* and *large*. This group of membership functions allows for a high degree of flexibility in defining the rules of the system, while it captures all the possible output cases.

The rules mapping the two inputs (connections ratio and colour distance) to the output (propinquity) of the fuzzy inference system are shown in Fig. 7. The rules embody the observations on the connections ratio and colour distance between components belonging to characters, as explained above.

The first rule handles the case when two components are not neighbouring (connections ratio is *zero*) and, therefore, should not be considered for a merger.

In all other cases, if the colour distance is *small*, the propinquity is always set to be above *medium* (*large* or *definite*, depending on the value of the connections ratio input). For *medium* connections ratio values, the propinquity is set higher than for *small* or *large* connections ratio values. This directly relates to the observations mentioned before: components that correspond to parts of character strokes (determined by the fact that they are partially neighbouring) should be favoured during the component aggregation process.

If **Connections Ratio** is *Zero*		then **Propinquity** is *Zero*
If **Connections Ratio** is *Small*	and **Colour Distance** is *Small*	then **Propinquity** is *Large*
If **Connections Ratio** is *Small*	and **Colour Distance** is *Medium*	then **Propinquity** is *Medium*
If **Connections Ratio** is *Small*	and **Colour Distance** is *Large*	then **Propinquity** is *Zero*
If **Connections Ratio** is *Medium*	and **Colour Distance** is *Small*	then **Propinquity** is *Definite*
If **Connections Ratio** is *Medium*	and **Colour Distance** is *Medium*	then **Propinquity** is *Large*
If **Connections Ratio** is *Medium*	and **Colour Distance** is *Large*	then **Propinquity** is *Small*
If **Connections Ratio** is *Large*	and **Colour Distance** is *Small*	then **Propinquity** is *Large*
If **Connections Ratio** is *Large*	and **Colour Distance** is *Medium*	then **Propinquity** is *Medium*
If **Connections Ratio** is *Large*	and **Colour Distance** is *Large*	then **Propinquity** is *Zero*

Fig. 7. The rules of the fuzzy inference system.

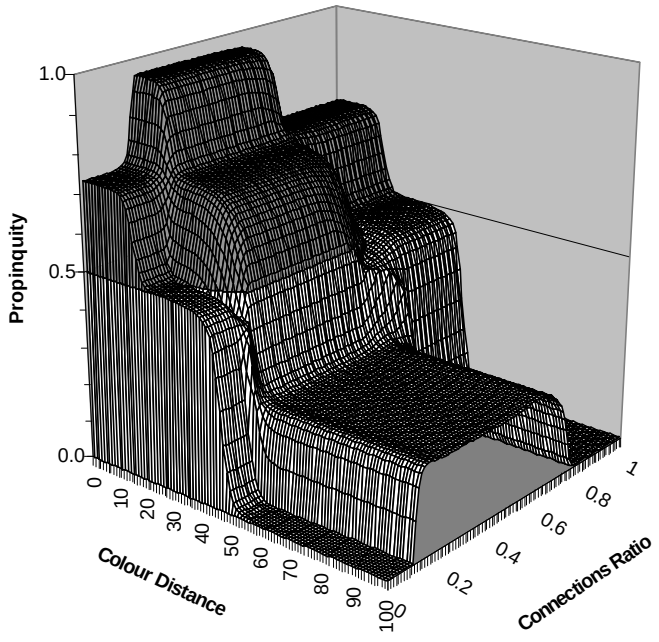

Fig. 8. The surface depicting the relationship between the two inputs and the propinquity output of the fuzzy inference system.

In a similar manner, if the colour distance is *medium* or *large*, the propinquity is set to *medium* or below (i.e., *medium*, *small* or *zero*, depending again on the value of the connections ratio input).

The fuzzy inference surface, illustrating the relationship defined by the rules of the system between the two inputs and the propinquity output can be seen in Fig. 8. The fuzzy inference system is designed in such a way, that a propinquity value of *0.5* can be used as the threshold in deciding whether two components should be considered for merging or not. Eventually, all pairs of components having a propinquity value above *0.5* (at the time of consideration) will be merged. Nevertheless, the exact value of propinquity plays an important role during the component aggregation phase, since it dictates the order in which mergers should take place, as will be seen next.

2.5. *Colour Component Aggregation*

The merging algorithm considers pairs of connected components, and based on the propinquity output of the fuzzy inference system, combines them or not. All components produced by the initial colour connected components identification process are considered.

For each connected component, the propinquity to each of the neighbouring components is computed, and if it is greater than a set threshold, a possible merger is identified. A sorted list of all possible mergers is maintained, based on the computed propinquity value. The algorithm proceeds to merge the components with the largest propinquity value, and updates the list after each merger, including possible mergers between the newly created component and its neighbours. Only the necessary propinquity values are recalculated after each merger, keeping the number of computations to a minimum. The process continues in an iterative manner, as long as there are merger candidates in the sorted list having propinquity greater that the threshold. The threshold for propinquity is set (as a direct result of the design of the membership functions) to be *0.5*.

3. Results and Discussion

The colour segmentation method was evaluated using a variety of images collected from different Websites. The test set comprises 124 images, which are divided into four categories: (a) Multicoloured text over multicoloured background (24 images), (b) Multicoloured text over single-coloured background (15 images), (c) Single-coloured text over multicoloured background (30 images) and (d) Single-coloured text over single-coloured background (55 images). This

distribution reflects the occurrence of images on Web documents. The number of colours in the images ranges from two to several thousand and the bits per pixel are in the range from 8 to 24. A width of four pixels was defined as the minimum for any character to be considered readable.

The evaluation of the segmentation method was performed by visual inspection. This assessment can be subjective for the following reasons. First, the borders of the characters are not precisely defined in most of the cases (due to anti-aliasing or other artefacts e.g. artefacts caused by compression). Second, no other information is available about which pixel belongs to a character and which to the background (no ground truth information is available for Web images). For this reason, in cases where it is not clear whether a character-like component contains any pixel of the background or not, the evaluator decides on the outcome based on whether by seeing the component on its own he/she can understand the character or not. The foundation for this is that even if a few pixels have been misclassified, as long as the overall shape can still be recognised, the character would be identifiable by OCR software.

The following rules apply regarding the characterisation of the results. Each character contained in the image is characterised as identified, partially identified or missed. Identified characters are those that are described by a single component. Partially identified 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 only one component (thus merged together), yet no part of the background is merged in the same component, then they are also characterised as partially identified. Finally, missed are the characters for which no component or combination of components exists that describes them completely without containing pixels of the background as well.

The algorithm was tested with images of each of the four categories. In category (a) 223 out of 420 readable characters (53.10%) were correctly identified, 79 characters (18.57%) were partially identified and 119 characters (28.33%) were missed. In addition, out of the 487 non-readable characters of this category, the method was able to identify 245 and partially identify 129. In category (b) the method correctly identified 284 out of 419 characters (67.78%) while 88 (21.00%) were partially identified and 47 (11.22%) missed. There were no non-readable characters in this category. In category (c) 443 (72.74%) out of 609 readable characters were identified, 115 (18.88%) partially identified and 51 (8.37%) missed. In this category, the method was also able to identify 130 and partially identify 186 out of 388 non-readable characters. Finally, in category (d) 572 (73.71%) out of 776 readable characters were identified, 197 (25.39%)

partially identified and 7 (0.9%) missed. In addition, 127 out of 227 non-readable characters were identified and 53 partially identified.

The method presented here, compares favourably to the previous (Split-and-Merge) method of the authors.[11] Two key differences can be identified. First, the present method performs considerably better on images in categories (a) and (b) (containing multicoloured characters) than the Split-and-Merge approach. The second difference between the two methods is the processing time, with the current method running in a fraction of the time required by the Split and Merge one.

The results mentioned above reflect the increasing difficulty in categories where the text and/or the background are multi-coloured. Some of the difficulties encountered in special cases can be seen in figures 9 to 13. For each image, the original is shown along with an image of the final segmentation and an image of the segmented characters. Characters shown in black colour denote correctly identified ones, whereas the ones in red are partially identified characters.

A typical problem encountered during the segmentation of Web images containing text, is the existence of small characters. The smaller a character is, the more its appearance is influenced by anti-aliasing or compression artefacts. An example of an image containing small characters is shown in Fig. 9. Although most of the characters in this example are correctly segmented, a number of them are broken into more than one connected component, mainly due to the presence of anti-aliasing artefacts in the original image.

Characters in gradient colour can also be difficult to segment, especially when the gradient effect is extensive and tends towards the colour of the background. In such cases, the border of the characters is indistinct, and the segmentation method may produce false results. An example of such a case is shown in Fig. 10. In Fig. 12, a further example of characters in gradient colour is given. Although in this case the gradient effect of the characters is extensive (over a large range of colours), the method performs considerably better, since the characters do not blend with the background.

Due to the significant variety of image text, a number of other problems can be identified. Such cases can be, for instance, the existence of semi-transparent characters (e.g., see Fig. 13), or characters that are originally —in the original image— merged, split, or overlapping (e.g., see Fig. 11). However, in most of these situations the segmentation method is able to perform quite well.

In conclusion, a new approach for the segmentation of characters in images on Web pages is described. The method is an attempt to emulate the ability of humans to differentiate between colours. A propinquity measure produced by a fuzzy inference system is used to express the likelihood for merging two

components, based on topological and colour similarity features. The results of the method indicate a better performance than the previous method devised by the authors and comparable performance to other existing methods. Further work is concentrating on the possibilities to enhance the propinquity measure by employing more features and in the further refinement of the fuzzy inference system.

Acknowledgement

The authors would like to express their gratitude to Hewlett-Packard for their substantial equipment donation in support of this project.

References

1. Search Engine Watch, http://www.searchenginewatch.com
2. M.K. Brown, S.C. Glinski and B.C. Schmult, "Web Page Analysis for Voice Browsing", *Proceedings of the 1st International Workshop on Web Document Analysis (WDA'2001)*, Seattle, USA, September 2001 (ISBN: 0-9541148-0-9) and also at `http://www.csc.liv.ac.uk/~wda2001`, pp. 59–61.
3. 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.
4. 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.
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. D. Lopresti and J. Zhou, "Document Analysis and the World Wide Web", *Proceedings of the 2nd IAPR Workshop on Document Analysis Systems (DAS'96)*, Marven, Pennsylvania, October 1996, pp. 417–424.
7. H. Li; D. Doermann and O. Kia, "Automatic text detection and tracking in digital video", *IEEE Transactions on Image Processing*, vol. 9, issue 1, Jan. 2000, pp. 147–156.
8. D. Lopresti and J. Zhou, "Locating and Recognizing Text in WWW Images", *Information Retrieval*, **2** (2/3), May 2000, pp. 177–206.
9. 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.
10. A.K. Jain and B. Yu, "Automatic Text Location in Images and Video Frames", *Pattern Recognition*, vol 31, no. 12, 1998, pp.2055–2076.
11. A. Antonacopoulos and D. Karatzas "An Anthropocentric Approach to Text Extraction from WWW Images", *Proceedings of the 4th IAPR Workshop on*

Document Analysis Systems (DAS'2000), Rio de Janeiro, Brazil, December 2000, pp. 515–526.
12. R. C. Carter and E. C. Carter, "CIE L*u*v* Color-Difference Equations for Self-Luminous Displays," *Color Research and Applications*, vol. 8, 1983, pp. 252–253.
13. K. McLaren, "The development of CIE 1976 (L*a*b*) Uniform Colour Space and Colour-diference Formlua," *Journal of the Society of Dyers and Colourists*, vol. 92, 1976, pp. 338–341.
14. G. Wyszecki and W. S. Stiles, *Color Science - Concepts and Methods, Quantitative Data and Formulae*. 2nd ed. John Wiley, New York, 2000.
15. *Basic Parameter Values for the HDTV Standard for the Studio and for International Programme Exchange*, ITU-R Recommendation BT.709 [formerly CCIR Rec.709] Geneva, Switzerland: ITU 1990.
16. A. Antonacopoulos, "Page Segmentation Using the Description of the Background", *Computer Vision and Image Understanding*, vol. 70, 1998, pp. 350-369.

Original Image Final Segmentation Identified Characters

Fig. 9. An image containing very small characters.

Original Image Final Segmentation Identified Characters

Fig. 10. An image with gradient characters over uniformly coloured background.

Original Image Final Segmentation Identified Characters

Fig. 11. An image with overlapping characters.

Original Image Final Segmentation Identified Characters

Fig. 12. An image containing gradient characters.

Original Image Final Segmentation Identified Characters

Fig. 13. An image with multicoloured (semi-transparent) characters over photographic background.