Colour Map Classification for Archive Documents

J. He and Andy C. Downton

Department of Electronic Systems Engineering, University of Essex, UK
{jhe,acd}@essex.ac.uk

Abstract. This paper presents our work on color map classification for archive documents. An approach is proposed which is very similar to the way that humans perceive document colors, by using the diversity quantization algorithm applied within the HSV colour space. Template color maps are manually registered for sample document images; images from batches to be classified are then mapped onto the closest template color map using a fuzzy color classification algorithm. The results of testing this approach on batches of archive index cards from the UK Natural History Museum were very encouraging. We tested over 400 biological specimen index cards, and achieved more than 98% correct color classification.

1 Introduction

Archive document image conversion for online digital libraries is a major application area of interest both for cultural purposes and where historical data need to be compared (e.g. for biodiversity or climate change monitoring). We have developed a prototype user-assisted Archive Document Image Analysis System [1], which automatically classifies the physical text contents of structured archive documents into a user-defined logical structure. The current system achieves about 93% correct classification of archive document text fields, but its performance is limited by the simple grey-level binarization algorithm used to separate foreground text from background card texture. While such algorithms are effective on clean modern printed documents, archive documents often show significant variations in background texture and colour due to ageing and fading, and foreground text may be written, typed, stamped or printed with several colours in a single document. Color rather than grey-scale segmentation of archive documents not only allows more accurate segmentation of the document content into independent colour planes, but may also help in labelling each plane (e.g. as containing a species name, annotation, or validation stamp) for subsequent OCR and database entry. On the one hand, this improves the quality of the segmented texture; on the other, it reduces the complexity of subsequent analysis by partially pre-segmenting the different text fields of the document.

To implement color segmentation, it is first necessary to obtain color maps that are representative of most of the archive document colors in a batch of documents. However, both background and foreground colors on archive documents vary significantly. The variation can be classified into two aspects, intra-class color variation, where the color (e.g. red) of a document component varies because of variable illumination or age-related fading; and inter-class color variation, where different images may have different numbers of colored document components due to varying document content.

S. Marinai and A. Dengel (Eds.): DAS 2004, LNCS 3163, pp. 241–251, 2004.

If too many colors are identified for a batch of archive documents, or an improper color map is assigned to an image, this will increase both classification errors and computational complexity. Therefore, we propose a method that uses HSV color space modelled by fuzzy logic [2] [3] to represent colors, and the diversity color quantisation algorithm [4] to detect the proper color map of an image.

The paper is organised as follows: Section 2 introduces modeling the HSV color space with fuzzy logic. Section 3 describes the color map template registration process. Section 4 describes how the color map for a sample document is identified by using the diversity color quantisation algorithm. Section 5 explains the classification process used for mapping the detected quantised color map onto the correct registered template. Section 6 describes our evaluation of the colour segmentation process and the results obtained. Section 7 draws conclusions.

2 HSV Color Space Modelling with Fuzzy Logic [5]

The HSV model is well-suited for mapping a linguistic representation of color, because of its similarities to the way humans tend to perceive color (e.g. grouping close colors together). It defines color space in terms of three parameters, hue (H), saturation (S) and value (V) [6]. Hue describes the basic color (e.g. red, blue or green) in terms of its angular position on a 'color wheel' from 0 to 360 degrees. Saturation describes the purity of the color, ranging between 0 and 100%. The higher the value of saturation, the purer is the color; and as saturation decreases, the color turns to grey and eventually white. Value, often called intensity, describes the brightness of the color, and also ranges between 0 and 100%. With decreasing intensity, the color turns to black.

2.1 Modeling with Fuzzy Logic

In archive documents, small amounts of color dominate the image foreground. Therefore, it is sufficient to represent document colors by dividing the color space into relatively big regions. First of all, we convert the 3-dimensional model into 2-dimensions (HS color space) by simply dropping the V coordinate, except for black representation. This has two advantages. Firstly, the illumination distortion effects, (variation of brightness) caused by the document scanning process or simply by colors fading with age, are reduced because they mainly affect the intensity value. Secondly, the 2-dimensional color model is much easier to visualise. The conversion is carried out by dividing the intensity range 0-100 into two regions, "LOW" and "HIGH" as shown in Figure 1. Any color *C(h,s,v)* whose intensity value falls in the range "LOW" is considered as the linguistic color *black* (*C(h,s,v) = black*), otherwise it is in HS color space (in other words, its intensity value is dropped *C(h,s) = C(h,s,v)*).

In 2-dimensions, HS color space can then be modelled, where hue (H) is equally divided into 12 regions as shown in figure 2 and each region corresponds to a linguistic color (e.g. red). Saturation (S) is divided into 3 regions, "LOW", "MEDIUM" and "HIGH" as shown in figure 3. Any color *C(h,s)* with saturation "LOW" in HS space is considered to be the linguistic color *white* (*C(h,s) = white*); a color with saturation "MEDIUM" is considered to be *desaturated*; and a color with saturation "HIGH" is considered as *pure*.

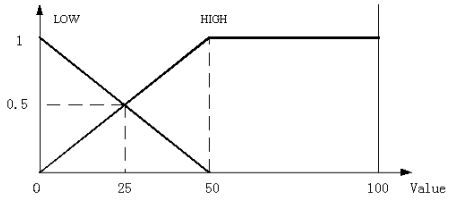


Fig. 1. Membership function for value (V)

Fig. 2. Membership function for hue (H)

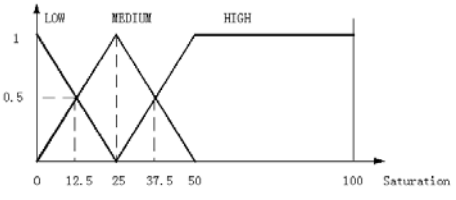


Fig. 3. Membership function for saturation (S)

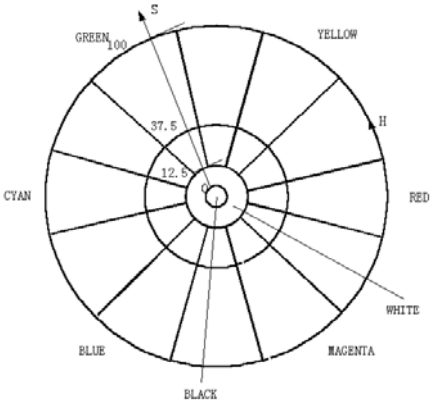


Fig. 4. HS color palette

The HSV color space, therefore, can be represented by a 2-dimensional HS-based color palette as shown in figure 4. The palette comprises four circles, each inside the other with the same centroid. The region inside the most central circle represents the color *black*. The ring region outside the central circle represents the color *white*. The remaining two ring regions are equally divided into 12 radial sections. Sections in the outermost ring region represent pure colors (e.g. red), while sections in the intermediate ring region represent desaturated colors (e.g. grey red).

2.2 Color Reasoning

Colors in HSV space are classified into specific regions on the defined palette by using a set of "IF-THEN" rules. First of all, we assign weights for each fuzzy set. For Hue, "RED" weighs 1, "YELLOW" weighs 3, and so on, round to 12; for Saturation, "LOW" weighs -1, "MEDIUM" weighs 1 and "HIGH" weighs 13. Regardless of the values of Hue and Saturation, if the Intensity is classified as "LOW", the color must be "BLACK". Therefore, the weight assigned to Intensity "LOW" must override others when combining the HSV parameters; this is achieved by selecting weights of 0 for "LOW" and 1 for "HIGH" in the representation of Intensity. Then, the product operator is applied to calculate the overall weight $C_w(h, s, v) = C_w(h).C_w(s).C_w(v)$. Each region on the palette has a specific value of $C_w(h, s, v)$ as shown in figure 5.

Fig. 5. Weight assignation on the fuzzy HSV palette

Table 1. Partial Color Reasoning

Hue	Weight	Saturation	Weight	Value	Weight	Color	Weight
RED	1	HIGH	13	HIGH	1	RED	13
RED	1	LOW	-1	LOW	0	BLACK	0
RED	1	MEDIUM	1	HIGH	1	GREY RED	1
RED	1	LOW	-1	HIGH	1	WHITE	-1(<0)
YELLOW	3	HIGH	13	HIGH	1	YELLOW	39
YELLOW	3	HIGH	13	LOW	0	BLACK	0
YELLOW	3	MEDIUM	1	HIGH	1	GREY YELLOW	3
YELLOW	3	LOW	-1	HIGH	1	WHITE	-3(<0)

For the BLACK region, the value of $C_w(h, s, v)$ is always 0; for the WHITE region, the value of $C_w(h, s, v)$ is always negative; for other regions, the value of $C_w(h, s, v)$ is unique. The required "IF-THEN" rules for color reasoning can then be systematically derived from the weights on the color wheel, e.g. if hue $C(h)$ is "RED" and saturation $C(s)$ is "HIGH" and value $C(v)$ is "HIGH", then the color $C(h, s, v)$ is "RED". If hue $C(h)$ is "RED" and saturation $C(s)$ is "HIGH" and value $C(v)$ is "LOW", then the color $C(h, s, v)$ is "BLACK". Table 1 shows some examples of color reasoning. In sections 3 and 4, reasoned colors are used to represent the image color map.

3 Color Map Registration

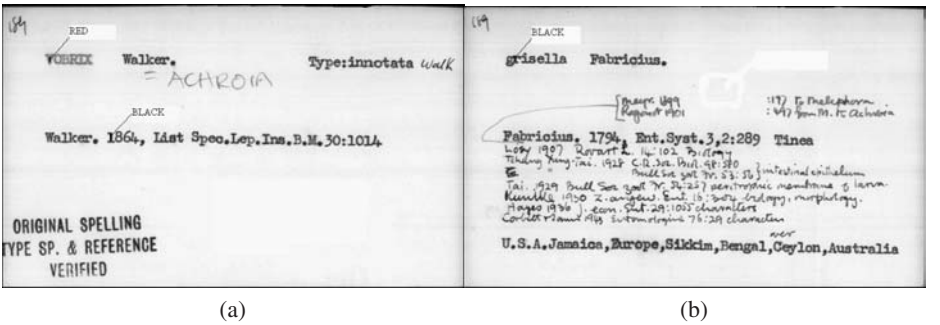
Sample color patches are selected from representative sample images (figure 6(a)(b)) by rubber-banding suitable areas using a user interface. As background color varies significantly image by image in archive documents, an exclusive representation for background is difficult to determine. Therefore, only the foreground color pixels of the selected sample patches (representing handwriting, typescript or stamps) are useful in constructing candidate template color maps. The marking mechanism also links the selected foreground color areas to relevant document analysis functions (e.g. text layout analysis, stamp detection and elimination).

Because some foreground areas are thin or narrow, the corresponding foreground samples will probably contain background as well as foreground pixels, and therefore require further processing to distinguish which pixels represent the distinctive foreground. Each pixel's color is represented using 3 16-bit values in RGB color space, so the color histogram of each patch is initially very flat, with individual pixels representing subtly different colors. This makes it difficult to find a single representative color value to represent the whole patch. Therefore, we quantize each pixel's color in each patch linearly into 5 x 5 x 5 = 125 unique colors by reducing the original 16 bit color representation to only 5 bits per color. This has the effect of clustering similar colors into a single common color value. Then, we convert the quantized colors into HSV space [7], and select the modal color as the representative color for that patch. The representative color for each patch defines the template image foreground color maps (figure 7(a)(b)) (which are represented by the fuzzy HSV color space modeled in Section 2), and are used as a mask for segmentation of images' foreground into color planes (each of which is subsequently processed using one or more linked document analysis functions). In the examples of figure 6(a) and (b), (a) contains two different foreground colors, "RED" and "BLACK", while (b) only has one foreground color, "BLACK" (which includes both typed data fields and handwritten annotations).

4 Color Map Detection

Each input image in the batch from which figure 6(a) and (b) have been selected as templates is first quantized linearly into 5 x 5 x 5 = 125 unique colors (as for the color map registration) to reduce the number of colors represented by its pixels. Any remaining colors which have less than 100 pixels are then discarded, as such small amounts of colour are assumed to represent noise. The remaining colors are then further quantized down to six distinct colors to minimise computational requirements using the diversity algorithm. This generates the color maps shown in figures 8(a) and (b).

The diversity algorithm runs a histogram on the entire input image, picks the color with the highest pixel count (normally the background color), and then iteratively finds five additional colors in the unpicked list that are furthest in HSV Euclidean distance



(a) (b)

Fig. 6. Biological specimen index card (a), with two foreground colors (b) with one foreground color

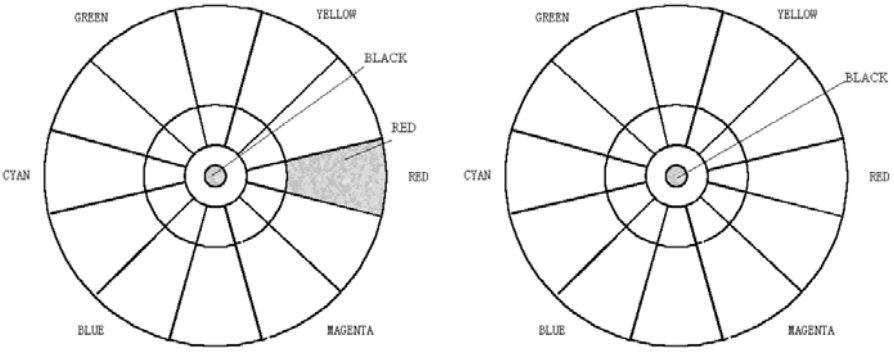


Fig. 7. Registered color map (a) with two foreground colors, (b) with one foreground color

[7]) from already picked colors. All remaining pixels are then added to the nearest color cluster. Other quantization algorithms such as Median Cut [8], Sequential Scalar [9] and Lloyd [10] can perform a similar color clustering function, but the Diversity algorithm is the most suitable here, because it identifies the potential background color by picking the color with the highest pixel count as its starting point. Although the Popularity algorithm [8] has a similar capability to select the background color, it may miss distinct foreground colors with small numbers of pixels when the number of quantized colors is limited. The way the Diversity algorithm picks colors is similar to humans observing colors on documents, where foreground colors (e.g. text) which are distinctively different from each other stand out, whereas similar colors tend to be perceived as part of the same document component.

Evaluation of our archive documents has shown that no more than 4 logically distinct colors are found on a single image. Therefore, 6 quantized image colors should be enough to include all distinct foreground and background colors. In general however, the number of colors in the quantized color map $CMAP_q$ should be designed always to be more than that in the registered $CMAP_r$. The next section shows how to map the sample image's quantized color map onto each of the template colour maps, and hence determine which template the sample document image matches.

5 Mapping

In this section, the sample image color map $CMAP_q$ generated by the diversity algorithm is matched against each registered colour map $CMAP_r$ using three steps, background mapping, foreground mapping and color map refinement.

5.1 Background Mapping

Even though no background color is recorded in $CMAP_r$, it must still be identified from $CMAP_q$, since the background color will eventually be used for color segmentation. The diversity quantization algorithm provides an easy way to identify it, as it is

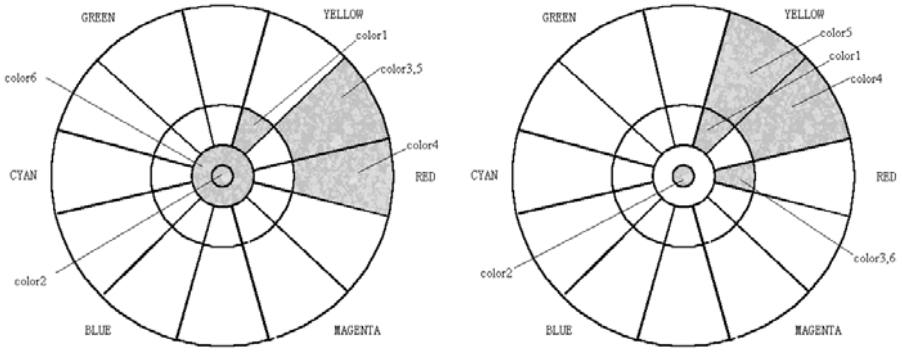


Fig. 8. Quantized color map (a) and (b) with six colors

always labelled first due to its dominant contribution to the overall pixel count. Therefore, the background color can immediately be identified, and the foreground colors are chosen from the remaining 5 unidentified colors.

5.2 Foreground Mapping

The foreground color mapping is straightforward. Each color in $CMAP_r$ is compared one by one with those in $CMAP_q$. Each match scores 1 (if a color in $CMAP_r$ matches twice or more in $CMAP_q$, it still scores 1). If the total score is equal to the number of colors in $CMAP_r$, $CMAP_q$ is considered to have the same color map as $CMAP_r$. If $CMAP_q$ can't be mapped onto any registered $CMAP_r$, the sample image is rejected. If $CMAP_q$ can be mapped onto more than one $CMAP_r$, the one with the highest score is chosen. For instance, color map figure 8(a) can be mapped onto both color map figure 7(a) and (b). The former scores 2 and the latter 1. Therefore, the former is chosen. The foreground colors are therefore identified in $CMAP_q$. However, two more complex cases need to be considered. First, there are some cases where unidentified colors remain in $CMAP_q$ after the foreground mapping process. Conversely, more than one color may appear within the same color class (e.g. two colors both classified as "BLACK") in $CMAP_q$. In both cases, it is necessary to decide which unmatched colors are useful and should be kept, and which of them are useless and should be discarded.

5.3 Color Map Refinement

The purpose of the refinement is to keep a reasonable number of colors (apart from those identified foreground colors which are already matched) in $CMAP_q$ for later color segmentation. The presence of too many colors may reduce the cohesion of the texture on each segmented color plane and increase the cost of computation. The absence of enough colors may result in undesirable segmentation (e.g. part of the background may be lost). Therefore, three rules are defined to refine the color map $CMAP_q$.

Rule 1: if more than one unidentified color or more than one identified foreground color have the same color class, these colours are merged.
Rule 2: if any unidentified color is next to an identified color (either background or foreground) on the defined palette, it is merged with the adjacent colour.
Rule 3: after applying Rule 1 and 2, if any unidentified color is still present, it is marked as background.
Figures 9(a)and (b) show the final color maps after refinement. In the final stage of color segmentation, even though each input image may be segmented into a different total number of color planes (e.g. more than one background plane may appear like figure 9(b)), the number of foreground planes will always be identical to that of the relevant $CMAP_r$. Additional background planes have no impact on subsequent analysis, as they are eventually discarded.

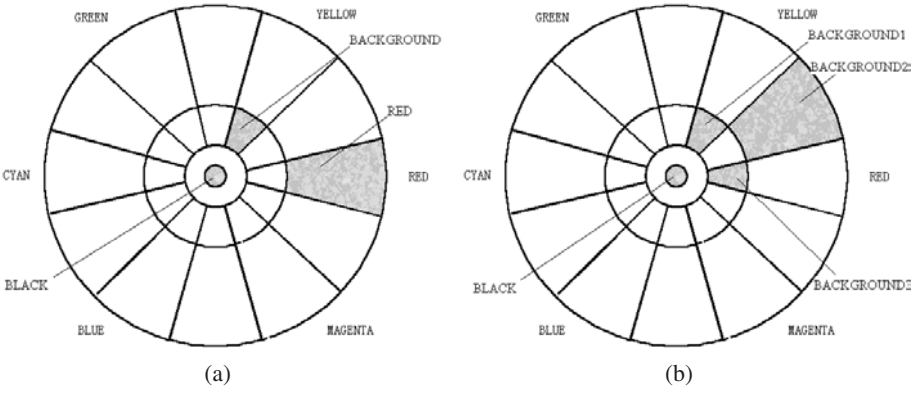
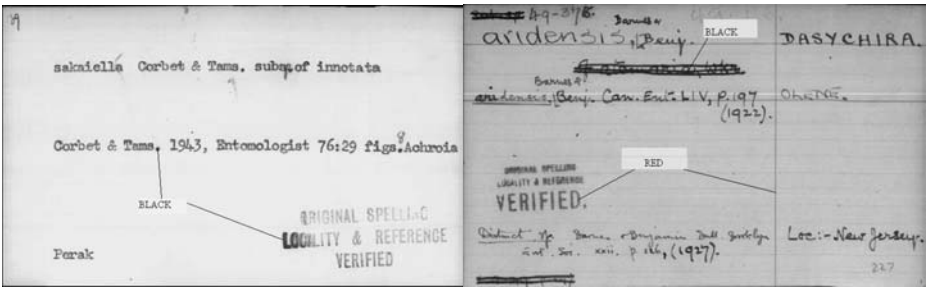


Fig. 9. Refined color map (a) with three colors, (b) with four colors

6 Evaluation and Result

6.1 Evaluation

Evaluation of the colour segmentation algorithm was carried out on a set of 400 sample biological archive index cards from the UK Natural History Museum, for which two color maps are registered as shown in figure 7(a)and (b). None of the testing cards was found to contain more than 3 foreground colors. 187 of them are similar to Figure 10 (a) (Type I), and have only one foreground color "BLACK". 213 of them are similar to Figure 10(b) (Type II), and have two foreground colors "BLACK" and "RED". The background colors of samples of both types vary significantly caused by color fading due to age and the use of different cards over time. Their foreground colors also vary because of fading and illumination variation. The texture with color "RED" includes machine-typed text (genus names), handwritten annotations, lines and stamps. The texture with color "BLACK" includes machine-typed text (species names and other data fields), handwritten annotations and stamps.



(a) (b)

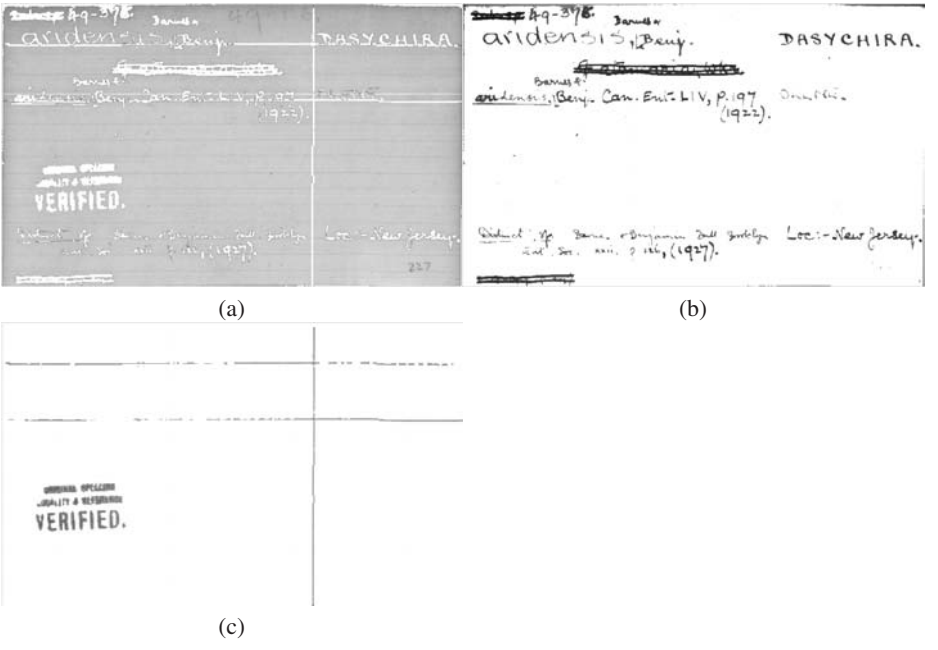
Fig. 10. Biological specimen index card (a), with one foreground color "BLACK" (b) with two foreground colors "BLACK" and "RED"

Table 2. Result of evaluation

Image Type	Color Map A (Fig. 7(a))	Color Map B (Fig. 7(b))	Rejection
Type I	5/178(2.8%)	173/178(97.2%)	0%
Type II	211/213(99.1%)	2/213(0.9%)	0%
Overall correct rate		98.2%	

6.2 Results

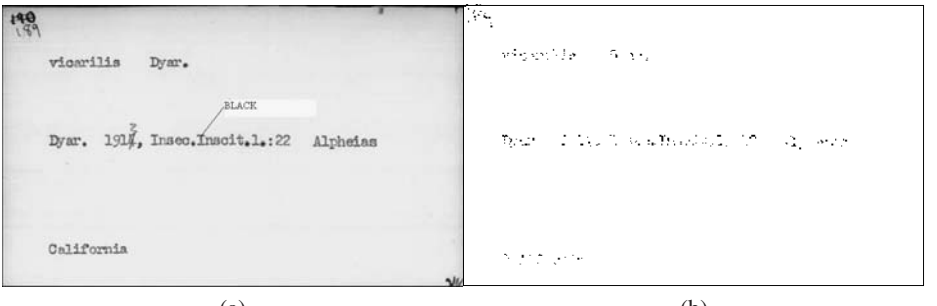
The results shown in Table 2 are very encouraging. 97.2% of Type I images have been assigned the correct Type I color map, while 99.1% of Type II images have been assigned the correct Type II color map. Figure 11 shows the background and two foreground color planes resulting from correct classification of the Type II sample image shown in Figure 10(b). The small number of Type I images (figure 12(a)) with a wrongly assigned Type II color map result from unexpected widely distributed "RED" pixels (figure 12(b)). These noise pixels may be caused by mixture of the foreground and background colors at the time of typing. The two classification errors for Type II (figure 13) images were caused by insufficient "RED" pixels in the image; as a result, the small number of "RED" pixels (caused by there being only a few "RED" characters with very thin strokes) were discarded during the initial color quantization stage. In both cases, the errors could be eliminated by refining the thresholding of the color map detection algorithm, which currently discards any color with less than 100 pixels. The required additional processing would be to set a higher discard threshold, but then to evaluate the positional variance of pixels within each quantized color below this threshold, and only retain colours with a small position variance, corresponding to correlated foreground print rather than distributed noise. In the case of the Type 1 images wrongly classified as Type II above, the "RED" pixels would then be below the increased discard threshold level, but are evenly distributed over the image, so would be discarded. In the case of the Type II images wrongly classified as Type I, the small number of "RED" pixels (previously discarded) would now be retained as a legitimate colour because they would be spatially closely correlated.



(a) (b)

(c)

Fig. 11. Successful segmented color planes (a) background (b) foreground1 (text in BLACK ink) (c) foreground2 (stamp and line in RED ink)



(a) (b)

Fig. 12. Biological specimen index card (a) with one foreground color "BLACK"; (b) distributed "RED" pixels (enlarged) causing erroneous classification

7 Conclusion

In this paper, the use of diversity quantization of the HSV color space has been proposed to classify colors on archive document images, so as to overcome color variation over different images in both the foreground and background colours. The diversity color quantisation algorithm combined with a fuzzy color classification algorithm is applied to detect a primary color map for each sample image. The primary color map is compared with those of pre-registered color templates to determine a correctly-labeled color

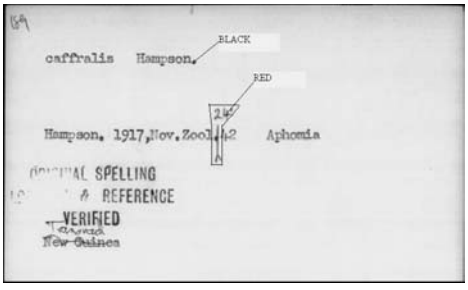


Fig. 13. Biological specimen index card with two foreground colors "BLACK" and "RED", which only represent a small notation

map for subsequent color segmentation and other document analysis. Initial results of experimental evaluation are very encouraging, with over 98% of test images assigned the correct color map. Small refinements to the algorithm are proposed to resolve remaining color classification errors.

References

1. J.He, A.C.Downton: User-assisted archive document image analysis for digital library construction. Proc.ICDAR'03 7th Int.Conf.on Document Analysis and Recognition **Vol.1** (2003) 498–502
2. L.A.Zadeh, J.Kacprayk: Fuzzy Logic for the Management of Uncertainty. John-Wiley, New York (1992)
3. D.Dubois, H.Prade, R.Y.Yager: Fuzzy Sets for Intelligent Systems. Morgan Kaufmann Publishers, San Mateo (1993)
4. J.Bradley: XV.Interactive Image Display for the X Window System. Version 3.10a 1053 Floyd Terrace. Bryn Mawr, PA 19010, USA. (1994)
5. L.Hildebrand, B.Reusch: Fuzzy Color Processing in: Fuzzy Techniques in Image Processing. Physica-Verlag (2000)
6. A.H.Watt: Fundamentals of Three-dimensional Computer Graphics. Addison-Wesley,Wokingham (1989)
7. J.D.Foley, A.V.Dam, S.K.Feiner, J.F.Hughes: Computer graphics: principles and pratice. (2nd ed. in C). Addison-Wesley, (1996)
8. P.Heckbert: Color image quantization for frame buffer display. Comput.Graph., **Vol.16** (1982) 297–307
9. R.Balasubramanian, C.A.Bouman, J.P.Allebach: Sequential scalar quantization of vector: An analysis. IEEE Trans. Image Processing **Vol.4** (1995) 1282–1295
10. S.P.Lloyd: Least squares quantization in pcm. IEEE Trans. Inform. Theory **IT-28** (1982) 129–137