Use of Colour for Hand-Filled Form Analysis and Recognition

Nasser Sherkat[1], Tony Allen, Wing Seong Wong
Intelligent Systems, School of Computing and Informatics
The Nottingham Trent University
Burton Street, Nottingham, NG1 4BU, U.K.
Tel: 0115-8486032 Fax: 0115-9486518
{Nasser.Sherkat, Tony.Allen, Wing.Wong}@ntu.ac.uk

ABSTRACT

Colour information in form analysis is currently under utilised. As technology has advanced and computing costs have reduced, the processing of forms in colour has now become practicable. This paper describes a novel colour-based approach to the extraction of filled data from colour form images.

Images are first quantised to reduce the colour complexity and data is extracted by examining the colour characteristics of the images. The improved performance of the proposed method has been verified by comparing the processing time, recognition rate, extraction precision and recall rate to that of an equivalent black and white system.

Keywords: Form Analysis, Form Processing, Layout Analysis, Colour Reduction, Form Extraction.

[1] Corresponding author

1. INTRODUCTION

A form is a special type of document that is used to capture data and information. Once captured, this data must then be extracted and processed in order to fulfil the purposes for which the data was acquired. Traditionally, most of this data has to be manually 'keyed' into the computer system before it could be processed. This is especially so for form documents that are filled-in with cursive handwriting. Unfortunately, this manual capturing process is both tedious and prone to errors. The process also requires many staff hours and can be very costly. According to [1], it costs 7-8 cents to process a single payment (one bill and one Cheque) in most utility companies in Canada. In the United States, the cost of manually capturing data from a form has been estimated to be about $2.50 per form [2]. Taking count of the amount of payments or forms that need to be processed every year, the costs will be considerable. Another problem associated with the manual capture of data from forms is that, even after data entry process, the original forms need to be kept for legal or audit purposes. Without the automatic indexing of form images on the basis of their entered data, subsequent inspection of these documents can be a labour intensive process. Automatic form processing and data capture, with its potential to rationalise the situation, has thus become a major research area.

Ideally, an automated form processing system should be able to capture all of the data from a form. Unfortunately, this is not the case and most of today's form-processing systems still require the need for human intervention to verify and

correct errors produced by the system. Effective form design has been shown to reduce the automated data capture errors [3] but in many cases re-design of existing forms is difficult and in most cases uneconomical. Moreover, even if the form can be redesigned, many of today's automated form-processing systems still fail to reliably process hand-filled data forms; especially those filled with cursive writing. This is due, principally, to the poor performance of the handwriting recognizer(s). These restrictions thus limit the potential of current form processing solutions and point to the need for a better and more reliable system to realize the full potential of office automation.

There are two fundamental approaches for extracting the filled-in data from form images – Model based and Model-less. In a model-based approach, the filled-in data is extracted by using a reference template created from a blank form image. This template can be created either manually, semi-automatically or fully automatically. Fig. 1 shows a typical model-based form processing system diagram. Note that in a manual template creation and recognition system, the form modelling process might not be present. However, in each of the cases, a blank form sample must be available. A model-less approach, on the other hand, is a form processing system that does not utilize any blank form knowledge to extract the filled-in data from filled form images. Whilst it is much more difficult to achieve good results using this approach, the model-less approach allows far more flexibility for the system when dealing with different kinds of form design. This is especially important if there are a large number of form types that have been produced at different places and times (for example scaling and design variations even within the same form type). Most of the time, methods that are

developed in a model-less system can be applied to a blank form to automatically generate the form template for the extraction process in a model-based system.

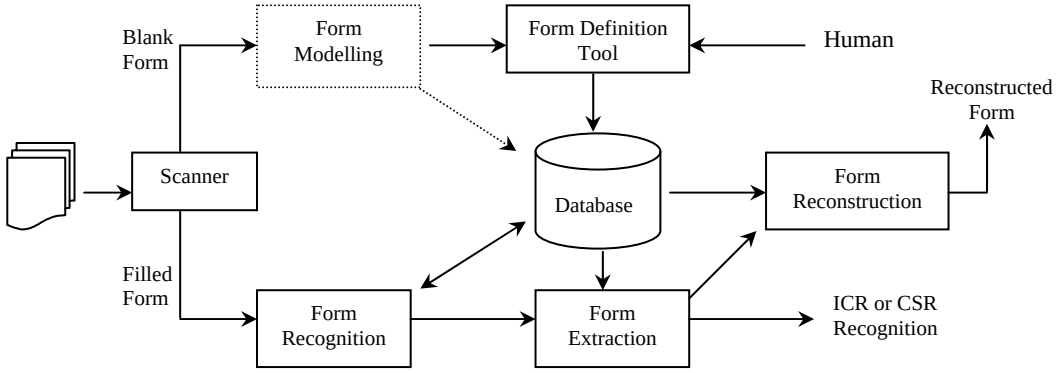


Fig. 1. A typical model-based form processing system diagram

In a typical form extraction system, the processes involved in extracting the filled-in data from a filled form are form registration, pre-printed entity subtraction and data restoration. Form registration is the process of mapping the template created in the form classification and definition process to a filled form so that the filled-in data can be located. This process is needed due to the skew and page offset that could be induced during the scanning process. Many robust methods had been proposed and developed that can produce a skew angle estimation accuracy of between +/-(1-3) degrees. Methods proposed include connected component projection profiles [4], Run Length encoding & Hough transform [5] and histograms analysis [6]. Skew detection and correction is considered a mostly resolved area now and many skew detection methods have even been implemented within the scanning devices themselves. The methods that are used to determine the vertical and horizontal offset between the template and the filled form rely on the same features that are used in the classification process. For example, in [10], line-crossing features are used to measure the

displacement. Within a +/- 25 pixel region of each of the template positions, the registration system will try to find a mapping offset value. A threshold of 3 pixels value is used to map a vertical or horizontal displacement point to the template. Mapped points are then summed and the scores used to determine the best-offset value for the form. Alternatively, in [11], a simple line location comparison is used to calculate the offset value. In this case, the form is first de-skewed and the line positions found in the blank and filled forms are compared. The difference value obtained is then the offset value used for the form.

Once the page offset has been determined, the filled-in data is then extracted using the information supplied by the template. This information is normally in the form of pairs of corner points indicating the positions of the filled-in areas or the pre-printed entities. In the case of filled-in areas, every pixel enclosed with the boxes formed by the corner points will be extracted and a further non-data removal process applied. The non-data removal process is needed because in some cases, pre-printed entities are enclosed within the boxes and are, thus, accidentally extracted by the system. If the template information is in the form of pre-printed entity location, a subtraction process is employed. All the pixels enclosed in the boxes will be removed and the remaining pixels should be the filled-in data.

In both of these cases, some portions of the filled-in data will be inevitably removed, especially when the filled-in data is touching the lines or pre-printed text. Text restoration is a process developed to resolve this degradation issue. In [11], a method is proposed to minimize the distortion of the filled-in data during the line removal process by using an intersection point analysis. In this process,

pixels that touch form lines are recorded and when the intersect points are bigger than a given threshold, the line portion at the intersect points is retained. In [7], several patching techniques were employed to restore all of the broken characters that result from the line removal process. Arc patching, binding arc patching and quadrilateral patching were employed to restore the detected intersection points between the lines and the text. The preliminary experimental results were promising but several issues were left unresolved, namely: filled-in words touching pre-printed text, confusion between filled-in characters and small check box areas and the differentiation of filled-in data from the pre-printed text. These problems were addressed later by X. Ye et al in [8] where 97.4% of the characters that were touching with pre-printed text were successfully separated using stroke width comparison. Unfortunately, the limitation of this method is that the filled-in stroke width must be different to the pre-printed text. This thus limits its scope to handwritten filled-in data that has a different stroke width to the pre-printed text.

Table 1 shows a summary of other proposed systems, detailing their approaches and reported performances. From the reported work, it appears that problems related to finding vertical and horizontal lines are well addressed and close to being fully resolved. The text degradation problems that are caused by the line removal process have also been fully investigated by many researchers and some very efficient methods have been developed.

Application Name (Approach/template creation method)	**Method**	**Achievement**
US IRS form processing system (Model-Based/Semi-Automatic)	Extraction by line intersections features [9] (Geometric Structure) Extraction by using ten line junctions features [10] (Geometric Structure)	Field Registration = 95% Form Recognition = 100% (Test set size = 25 forms) Extraction rate = 99.5% Form recognition = 98%
Intelligence Form Processing System (Model-Based/Manual)	Extraction by form mapping using line and boxes features (Geometric Structure) [11]	Extraction speed = 10 seconds
Generic Form Dropout System (Model-based/Automatic)	High level morphological subtraction [12] (Logical Approach) Four Directional Adjacency Graphs to locate form fields (Logical Approach) [13]	Very robust method for extracting data from grey-scale images Preliminary results are claimed to be promising but no detailed results are given
Generic Form Dropout System (Non-Model based)	Extraction by Block Adjacent Graph (BAG) method [14,15,16] Connected Component Analysis method with dilation and erosion for extraction of lines component [17] Extraction by using types of line segments features and fuzzy matching for form recognition [18] Form structure detection using strip projection [19,20]	Solves most lines/text overlapping issues 98.5% of fields detected 99.1% of lines detected 0.7 to 1.6 seconds Feature Extraction time = 1.77s to 6s Fast and robust, better than Hough transform and run length based algorithm

Table 1. Summary for some of the proposed form extraction systems, their proposed approaches and reported performances.

Line features are the strongest and most prominent characteristics in form documents, thus it is no surprise that most of the systems proposed are based around these features. However, as this feature has been fully explored other

avenues must be explored to improve performance. As demonstrated in [21], one of the possible features to be used is colour. In this work, colour has been used successfully to extract the signature and seal imprint from cheque images. It is thus believed that this concept can be extended to a form-processing environment where it has a great potential for further improvements in the current form processing systems.

2. A COLOUR BASED EXTRACTION SYSTEM

The concept of using colour to improve the performance of an automatic form processing system is not new. For instance, **non-read inks*** are extensively used in the form design & printing industry to help improve form capturing efficiency by allowing the automatic removal of pre-printed entities from the filled form. However, aside from taking advantage of the 'colour-blind' behaviour of the scanner, **colour knowledge** has rarely been used in any of the post scanning form-processing techniques.

In this paper, a new hybrid approach for extracting the filled-in data from form images is proposed. This novel method combines a software-based colour dropout and subtraction technique with colour information to provide a better form extraction solution.

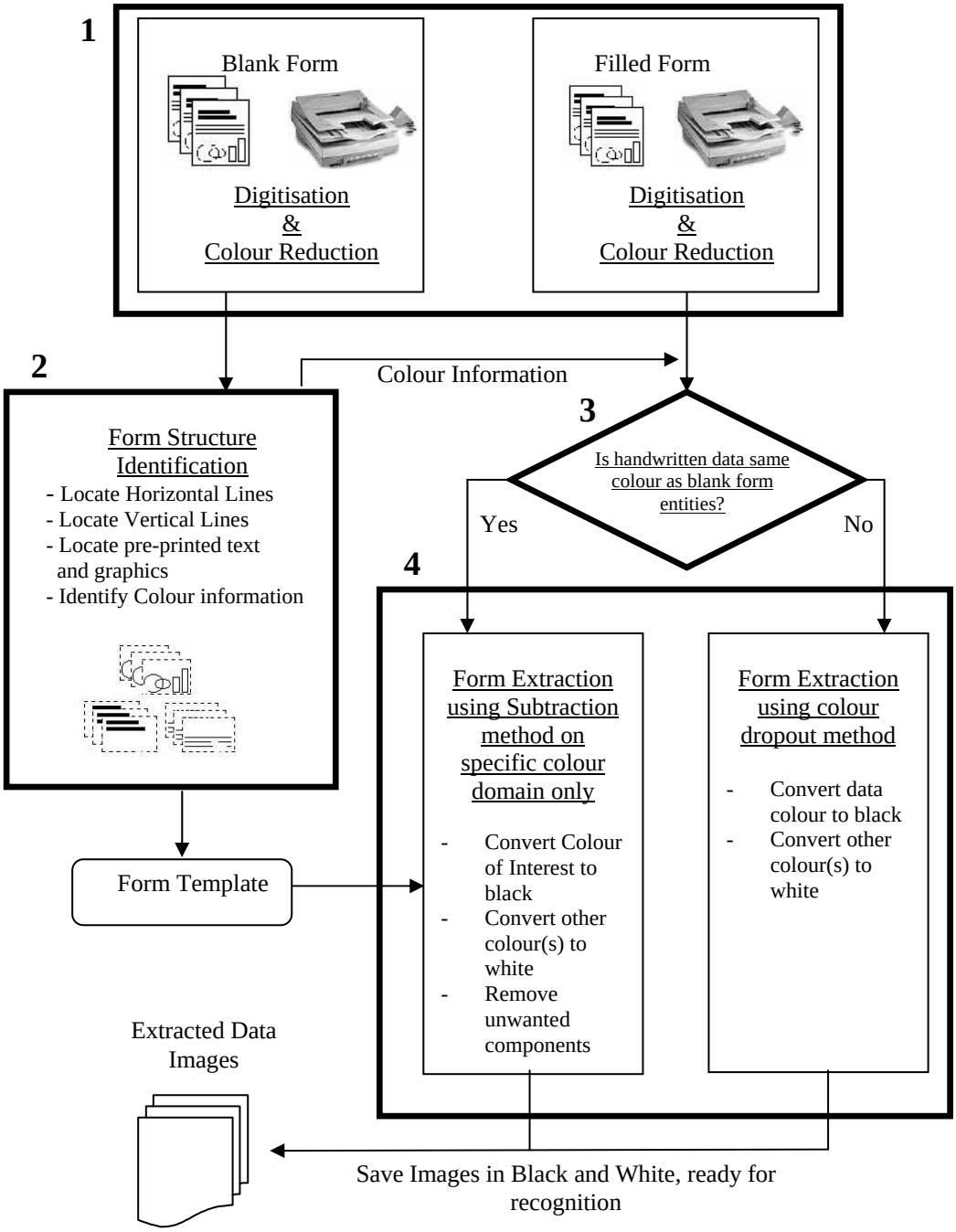
Some assumptions have been made in this study to enable focusing on the use of colour to extract data from form images:

* **non-read inks** are the colours that are invisible to the scanner device. Different light source used in each scanner will have different non-read ink and colour responses.

- The skew angle of the scanned images is assumed to be less than 3°
- Blank (unfilled) forms are available to the system
- Skew marks are not available

Fig. 2 shows the proposed colour-based form extraction system. It consists of 4 major parts:

1. Form digitization and quantization
2. Blank form structure analysis and identification
3. Colour analysis
4. Form extraction



1. ***Form Digitization and Quantization***
2. ***Blank Form Structure Analysis & Identification***
3. ***Colour Analysis***
4. ***Form Extraction***

Fig. 2. Block diagram of the proposed colour-based form extraction system.

2.1. Form Digitization & Quantization

As colour can provide useful information for layout analysis and text extraction, we propose to digitise all the targeted forms in colour. However, even with today's technology, processing a 24-bit full colour image directly is still impracticable. This is principally due to the nature of the digitization process (i.e. smoothing and anti-aliasing effects etc.). A single colour will normally be spread across a wide range of values after the digitisation process. For example, after a form has been converted into a 24-bit full colour image in the RGB domain, a red colour can easily range from 255,200,200 to 140,80,80. It becomes even more complicated if the colour response of the digitization device is considered. Optical scanners, for instance, will all have different colour sensitivities under different light sources [22].

Basically, there are two reasons for reducing the colour content in an image – first, to save memory and second, to reduce the system complexity and computational cost [23]. The first aim is to reduce the colour information in an image as much as possible whilst minimizing the visual degradation that could be caused by this action [24,25,26,27]. Its main advantages are: smaller images file size for faster transmission and lower storage cost for image storage and retrieval systems. The most common procedures for such applications are by transforming the colour format from a RGB colour domain to a human visual model domain [28]. These techniques attempt to keep the images as close as possible to human perception even after the colour reduction process. The second aim, which is most closely aligned with this work, attempts to reduce the complexity of the document analysis and processing system. Many techniques

[29,30,31,32,33,34,35,36,37,38] have been proposed that attempt to address the colour issue for text extraction from colour documents. Some of these work directly in the RGB colour model [30,31,33,35], while others convert the images to a HVS (Human Visual System) colour model [29,32,34]. The disadvantages of using a HVS model for such applications are the accuracy and efficiency loss during the conversion [39]. In fact, as suggested in [31], a better segmentation result can be obtained for colour text document when applying the colour reduction method directly onto the RGB space. Figure 3 shows a summarised process flow for each of the techniques proposed. It is important to point out that all of the techniques reported so far make very limited use of colour information for document analysis. Their main purpose is to reduce the colour content in order to simplify the text extraction process.

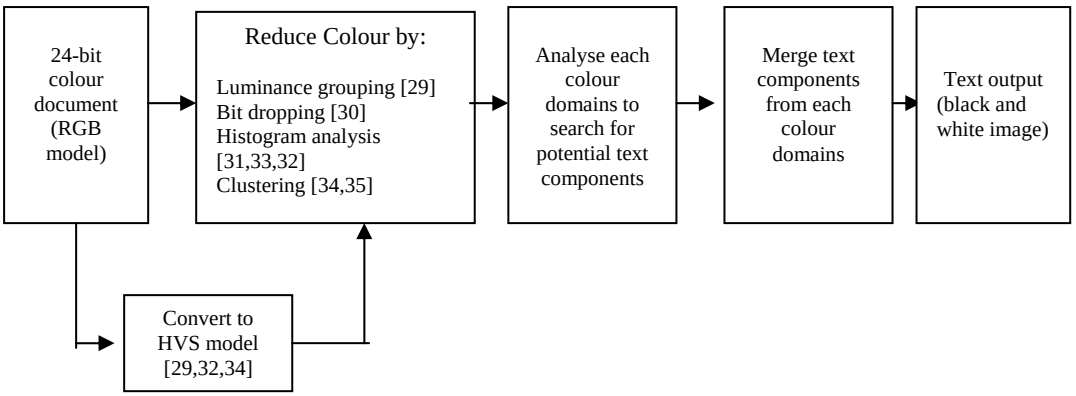


Fig. 3. The process flow for the techniques that have been proposed to extract text components from colour documents

The colour reduction process generally consists of two steps. First is the palette design, in which the total number of palette colours (i.e. the targeted number of colours that the image is intended to be reduced to) is defined. Second

is a pixel mapping, in which each colour pixel is assigned to one of the colours in the palette.

Bit dropping is the simplest method to use to reduce the number of colours in an image. The idea is that only a given number of significant bits are important to represent the pixels colour. For example, in a 24-bit RGB colour image, each pixel is represented by an 8-bit R, G and B value. If we used only the first two significant bits to represent the colour and ignored all the other bits, then the image will effectively be reduced to a 6-bit colour image. This method is fast and effective, especially for colour images that have just a few very distinct colours. However, the major drawback for such method is that the targeted number of colours that the image can be reduced to cannot be less than 64 (2 bits for each R, G and B component) without loosing a significant amount of information. In addition, the number of colours that it can reduce to is rigid.

Colour histograms are one of the most commonly used techniques to facilitate palette design. Palettes are first found by searching for the dominant peaks in the image colour histogram and assigning each of these peaks as the palette colours. The number of peaks defines the total number of colours that the image will reduce to. The main problem for this is not all colour in an image gives a clear peak in the histogram, especially complex colour images. Often text will be separated into several colour domains after this process and a text searching and merging process is then needed.

Luminance and chrome distances have been used in [29] to classify pixel colours into one of 42 pre-defined colours. A basic group of 21 quantized colours is first defined using the combination of chromes (Red, Green, Blue, Yellow,

Magenta, Cyan and Grey) and luminance values (Dark, Middle and Light). A derived group of another 21 quantized colours is then defined at run-time. This method suffers from the same problem as the histogram method, in that a separate technique is needed to re-merge the text that has been separated into different colour domains.

Pixel mapping is the process of merging the nearby pixel colours into one of the pre-determined palette colours. This process is sometimes referred to as clustering, where a cluster is used to represent each of the palette colours and a mapping process is applied to merge the pixels colour value to its nearest cluster. A frequently used clustering algorithm is the C-means clustering algorithm (CMA) [40], where cluster representatives are iteratively updated and labelled. Other clustering algorithm includes Fuzzy C-means clustering [41], learning vector quantization [42] and Kohonen Self-Organizing Maps (SOM) [43]. All these techniques have been implemented successfully and shown to be effective in mapping the pixel colours into the palette colours. Unfortunately, in all of these cases, no quantitative results have been given and the evaluation is mainly based on the visual output. Thus, the performance for each of the methods is very subjective and there is no simple way to compare their effectiveness and establish whether they can be in text extraction from colour form images.

Generally there are two possible methods in resolving the text extraction (colour segmenting) issue:

1). Employing a merging process after the quantization process
2). Defining a smaller number of palette colours before the quantization process

Unlike other types of document, a form contains relatively few colours. This is due to the nature of the form function. As a form is used to capture data from people, the area of interest (filled-in data area) often contains simple or single plain colours as the background. Thus, it is obviouse that the second method is more appropriate. Our observations so far have shown that the majority of forms in use today can be adequately represented using just a few colours from a group of 8 pre-defined colours.

As a scanner digitizes colour documents using an RGB model, methods that work directly on the RGB model will eliminate the extra processes needed to convert this colour model to other models. Processing time is crucial in any automated form processing system, thus it is important to process the colour image in its original colour model in order to minimize the extra burden imposed on the system as a result of the inclusion of colour into the process.

In the RGB colour model, colour is represented by the amount of red, green and blue components. A full colour image uses 8 bits per colour component to produce 16 million possible variations of colour in an image. However, as mentioned earlier, form images need only a few of these colours. Therefore, an aggressive colour reduction method is proposed that can reduce the total number of colour variation of an image from 8 bits per colour component to 1 bit per colour component. This not only reduces the colour variation of an image from 16 million colours to just eight colours it also reduces the memory consumption of the image and makes the image much easier to manipulate than the full colour original. The only draw back for using such an aggressive colour reduction

technique is that when the original form document contains more than 8 colours; then extra colour information will be lost in the colour reduction process. However in practice, this rarely happens and even when it does, the 8 colour domains still provide more information than the conventional black and white alternatives. Fig. 3-1 shows how these eight colours are formed in the RGB colour model.

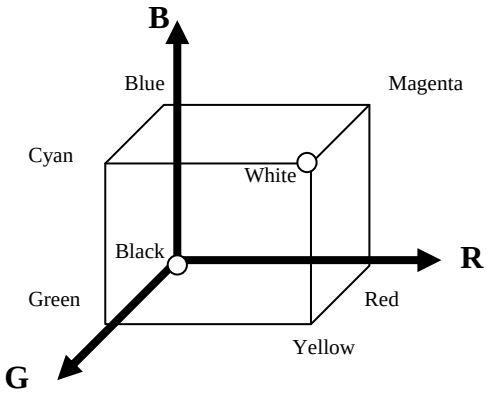


Fig 3-1. In the RGB colour model, each colour is represented by specifying the amount of red, green and blue components. In an 8-bit per component system, these values will range from 0 to 255.

In an RGB colour model, a pixel's colour can be determined by comparing the RGB component values. If all these RGB values are close to each other in value, then the pixel will appear as a black, grey or white colour. When one or two of the component values are significantly higher than the other component value(s), a pixel's colour will be more prominent and distinct. For example, a pixel will appear as red colour when the R component value is significantly higher than its G & B component values. Similarly, when the R & G component values are significantly higher than the B component value and the R & G values are close to each other, then the pixel will appear as a yellow colour. Fig. 3-2 illustrates how the eight pre-defined colours can be formed by comparing the RGB values of a pixel.

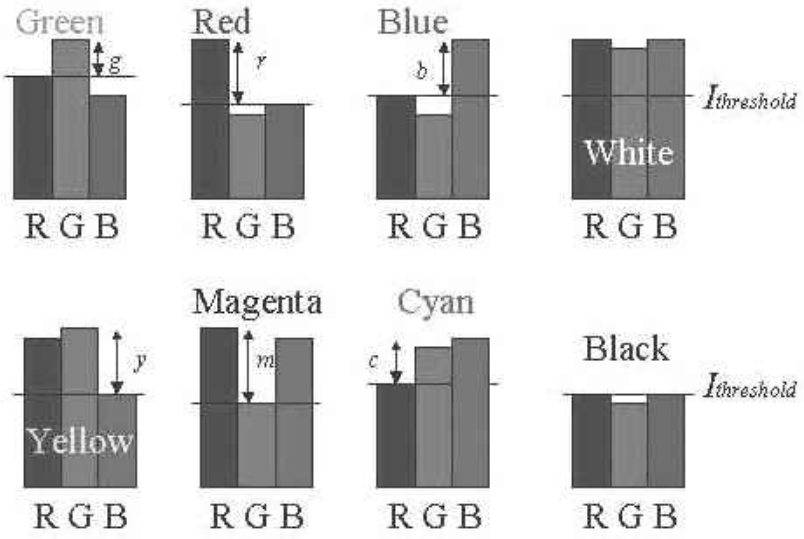


Fig 3-2. A pixel's colour can be determined by comparing the R, G & B component values within that pixel.

A pixel's colour can thus be determined by using the following expressions:

GREEN when G>R+g and G>B+g

RED when R>G+*r* and R>B+*r*

BLUE when B>R+*b* and B>G+*b*

YELLOW when R>B+*y* and G>B+*y*

MAGENTA when R>G+*m* and B>G+*m*

CYAN when G>R+*c* and B>R+*c*

If pixel's RGB values does not meet any of the above conditions

BLACK when $(R+G+B)/3 < I_{threshold}$

WHITE when $(R+G+B)/3 > I_{threshold}$

Depending on the brightness characteristic of the scanner, the $I_{threshold}$ value that determines a pixel's colour as either a black or white colour may range from 170 to 220. The r, g, b, y, m and c values vary according to the colour characteristic of the scanner. For a scanner with an operating light source wavelength of 700-800 millimicrons, the scanner will be more sensitive to the

white, yellow and red colours compared to the blue and green colours. Thus a red colour will be digitized to a pixel that has an R value that is significantly higher than its G & B component values when compared to another scanner that employs a 500-600 mili-microns light source. This scanned image thus requires a different (higher) r value compared to images scanned using the later light source. Such a system will also need significantly smaller g, c & b values than the r, y & m values. Empirical results[+] show that the r, y & m values may range between 35-50 and the g, c & b values may range between 10-25. These values were determined by repeatedly quantizing several different kinds of colour images using different threshold values; starting from 0 and increasing to a point where the quantization output fails to quantize the image colour correctly. For example, with a digitized image that contains only red and black colours, the acceptable threshold range for the r-value will be those r-values that allow all the distinguishable red colour pixels in that image to be quantized into red colour pixels. The test images in this experiment were carefully selected (each contains 2-3 colours) so that the colour content in these images covers all of the pre-defined eight colours and represents all of the most commonly found colours in form documents. Table 3-1 shows the experimental results for this test and a summary of the ranges of threshold values that produce the correct quantization results.

[+] Based on experiment performed on several different colour images that contained different colour content, digitized with 3 different scanners (Hewlett Packard 5200C, Hewlett Packard 3690C and Epson GT 6000)

	Threshold range
r	35-50
g	10-15
b	15-25
y	38-50
m	35-50
c	15-20
$I_{threshold}$	165-215

Table 3-1. Depending on the brightness and colour characteristics of the scanner, different threshold values are needed to quantize the colour images correctly. These are the workable threshold ranges found using 3 different scanner models.

Since it is not possible to determine the threshold ranges for all of the colour variations found in all documents, these thresholds will fail at some point. However, within the bounds set by this experiment, it is shown that the quantization method will perform its adequately using the given range of threshold values shown in table 3-1. Therefore, by choosing the mid-value of these ranges, the quantization method will perform its task correctly as long as a form does not contain a very large colour variation. The r, g, b, y, m, c and $I_{threshold}$ values chosen for this work are thus 43, 13, 20, 44, 43, 18 and 190 respectively.

Besides the scanner brightness and colour characteristic considerations, the scanning resolution must be taken into account as well, since it plays an important role in a pixel's colour generation. The effect of scanning resolution on the colour pixel production is illustrated in Fig. 3-3. Note that as the resolution reduces, an image starts to lose the detail of the original picture (in this case the black line) and, at 100dpi, a black line is merely represented by a row of dark yellow pixels.

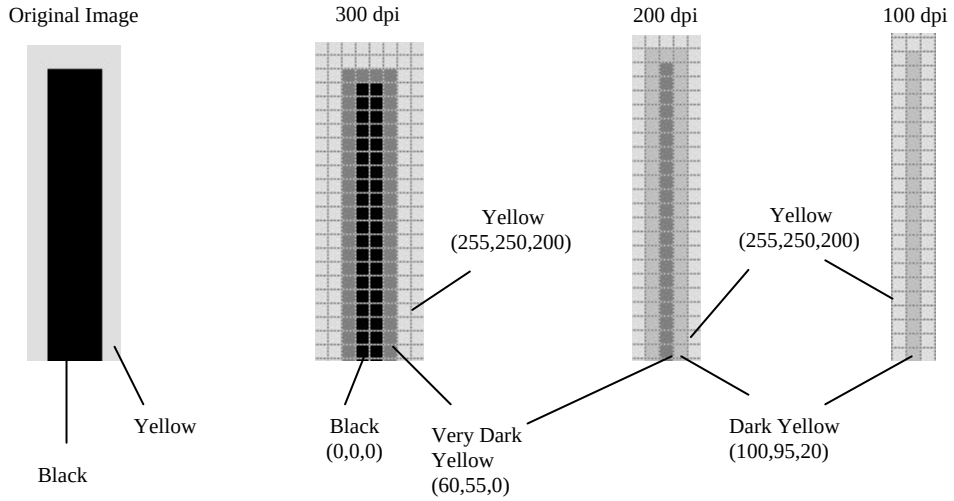


Fig 3-3. The effect of different scanning resolutions on the generated pixels' RGB values.

Thus, if the RGB comparison method is used alone to quantise the colour images, then, at 100dpi, the dark yellow line would be quantised to a yellow colour causing information to be lost in the quantised image. To solve this problem, a maximum reference value is first determined for each of the colours (up to maximum of eight as pre-defined earlier) present in an image. This can be done by applying a first pass pixel scan before the quantization process is carried out. In this process, each of the pixels is provisionally labelled as one of the eight possible colours based on the expressions mentioned earlier. The highest relevant value of each of these obtained colours can then be determined. For example, the red colour reference value will be the highest R-value $(R)_{max}$ found for a red-labeled pixel within the image, whilst for yellow it will be the highest mean value of the R and G components $((R+G)/2)_{max}$. The only exception is for the black colour reference value, which uses the minimum average value of the R,G and B components found in the image.

In a second pass, each of the pixels' colours is then decided by comparing its RGB values to their respective colour reference values using a distance threshold value D*. Take fig. 3-3 for example, by using the comparison technique on the 300dpi image, two colours will be identified (Yellow & Black - with a reference value of 252.5 (from 255,250,200) & 0 (from 0,0,0) respectively). Using these reference values, each of the pixels in the image will then be quantized into either a black or yellow colour depending on the differences between the pixel and reference RGB values. For instance, a dark yellow pixel with an RGB value of 60,55,0 will be quantized to black as the difference between the dark yellow pixels R & G values ((R+G)/2=(100+95)/2=97.5) and the yellow reference value ((255+250)/2=252.5) is greater than the threshold. Similarly, for the 100dpi image in figure 3-3, the dark yellow pixels will be quantized to a black colour while all other bright yellow pixels will be quantized to a yellow colour. Fig 3-4 shows an example of the quantization result using such an approach.

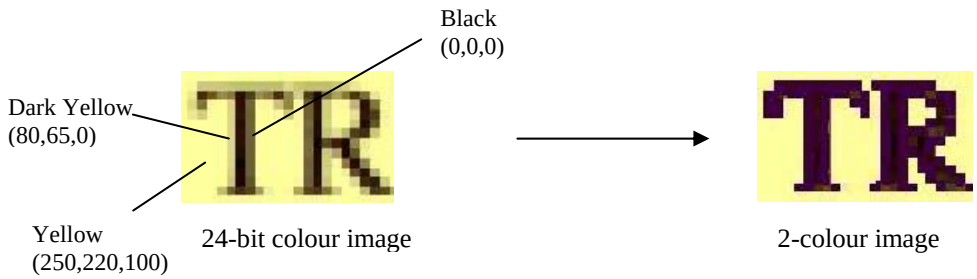


Fig 3-4. Example of how dark yellow pixels are quantised to black colour when using the value referencing method on a 200dpi image

This two pass method does successfully cope with the problems of low resolution scanning, however, as processing time is important in form processing

*By experiment, this D value is found to be in the range of 50-120 in order to obtain a correctly quantized output. The value chosen for this work is the mid-value of this range, i.e. 85

automation, the use of a two pass pixel may seem computationally expensive. To overcome this, the algorithm is improved by taking into account the colour visibility characteristic of the human vision system. As explained in [44], a human being is only able to distinguish colour when there is a sufficient amount of light (luminance) present. Thus, if a pixel's R, G and B component values are very low, then the colour is invisible to human vision and can thus be considered as a black colour. For instance, even when a red pixel RGB values are 90,0,0, it still appears as black colour to human eye. Therefore, to speed up the quantization process, when a pixel's RGB values are very low (empirically found to be R & G & B < 80), it is safe to quantize these low value pixels to a black colour without applying the above-mentioned RGB comparison method. This effectively reduces the number of pixels that need to be compared to the reference values thereby increasing the system performance.

2.2. Form Structure Analysis and Identification

As the skew angle of the form images is assumed to be very small, a run length pixel count method can be employed to locate any possible lines within the blank form. In a black and white image, a line can be defined as a long series of black pixels connected to each other in a row. Similarly, in a colour image, lines can be located in the same way as in bi-level images by examining each of the colour domains individually. However, in both cases, this method will fail to locate broken, dotted or dashed lines. Therefore, a Run-Length Smooth Algorithm (RLSA) [45] is needed to join the dotted lines together before the pixel count method is applied. For the purpose of defining line connectivity in this study, a line is assumed to have a maximum of 5 pixels gap between two line segments.

This is based on the assumptions that doted, dashed and broken lines are normally very closed to each other and when scanned at 200dpi, should produce a gap of less than 5 pixels width. Thus, when locating horizontal lines, a horizontal RLSA with a constant C value of 5 is applied throughout the image.

There are 3 possible types of lines in a form image – long run length lines that are used to form tables, boxes and fill-in spaces, short run length lines that are used to underline words or sentences and very short run length lines that are used as dashes or minus signs in a sentence. By using a run length pixel count method, long lines are easier to identify than short lines. This is due to the fact that text will often appear as short lines after the RLSA process. Hence, to avoid wrong classification, only lines that are obviously longer than 1 or 2 words are considered as targeted lines. In a 200dpi image, this is equivalent to a run length of approximately 100 pixels. Therefore, when there is a long series of black pixels connected to each other in a row for more than 100 pixels, this portion of pixels is retained as a potential line.

A line verification process is then applied to all these identified long run pixels using a contour tracing method [46]. Using this method, all the connected pixels are grouped together and enclosed in a bounding rectangle. As a line is always just a few pixels thick and is usually much thinner than text or other entities, a line can be ascertained by examining the height of this bounding rectangle (a typical horizontal line in a 200dpi image is not more than 4 pixels height). Once all the horizontal lines have been located and removed from the image, a vertical RLSA with a constant C of 5 is then applied across the image again followed by a run

length pixel count in the vertical direction (assumed vertical lines' gap are the same as horizontal lines). This process is necessary because when the horizontal lines are removed from the image, some of the vertical lines become broken. By applying the RLSA before the vertical pixel count process, all the broken vertical lines are re-connected back together again. The identified vertical lines are then also removed from the image using the same approach as described for horizontal lines.

Once all the true vertical and horizontal lines have been located and removed from the image, the remaining connected groups then represent the text and graphic components for that form. These groups of rectangular bounding box locations, together with the line bounding boxes are then recorded as a template for the form removal process.

Although the line-searching algorithm just described may seem rather simple to be able to identify all of the possible lines in a document image, the method does provide a quick solution for identifying long lines in non-complex documents such as form documents. It is by no means a robust method for finding lines in complex documents when compared to other methods reported in literature [47,48,49], however, the main advantage of using such a method is its ease of implementation. Besides, the main reason for finding lines in this case is to perform subtraction at a later stage. Thus, even when a line is not identified in the line searching process, it will be treated as one of the other pre-printed entities and will still be subtracted from the filled form at the extraction stage. Hence, the line-searching requirement in this system is less stringent than in other applications and thus the line-searching algorithm described appears as a viable method.

2.3. Colour Analysis and Extraction

After the filled forms have been scanned and quantized, the colour information in the filled form is compared to the blank form colour information. This is in order to decide whether the colour drop-out or the subtraction technique is to be adopted in order to extract the filled-in data from the image. As the number of colours and their pixel percentages are known from the quantization process, comparing the blank and filled form can easily identify the filled-in data colour. For instance, suppose a 3-colour blank form contained white (87%), black (9%) and red (4%) colours after the quantization process, the filled-in data colour on a completed form can be known instantly by comparing these figures against the quantized filled-in form values. This is shown in table 3-2.

		Filled-in colour
Blank form	White (87%), Black (10%), Red (3%)	-
Filled form #1	White (85%), Black (10%), Red (3%), Green (2%)	Green
Filled form #2	White (85%), Black (12%), Red (3%)	Black
Filled form #3	White (85%), Black (10%), Red (4%), Blue (1%)	Red & Blue

Table 3-2. The percentage changes for each of the colours contained in a form can be used to identify the filled-in data colour.

When the filled-in data has been entered using a different colour to that present in the blank form, data can be extracted readily using the colour dropout process. In this process, the filled-in data colour pixels are converted to black whilst all the other colour pixels are converted to white. The resultant black and white image is then ready for recognition. This method effectively eliminates problems such as page skew, page offset and abnormal filled data conditions (such as filled data out of the designated area) seen in the subtraction approach.

When the filled-in data has been entered using one or more of the colours that are used in the blank form, then a subtraction technique must be adopted. However, unlike other form extraction systems, only the filled-in colour domains need to be processed. This eliminates the necessity to process the whole image and hence improves the system efficiency.

As none of the commercially available forms used in this study contained marks that could be used to correlate the template entity locations to their respective filled-form equivalents, a pixel count method is used. This method identifies the first X- and Y-positions that contain more than a given threshold number of pixels, starting from the origin (0,0) position. For example, suppose a blank and filled form image have their first black pixel counts of more than the threshold number of pixels at coordinates 20,10 (x,y) and 18,15 (x,y) respectively, the offset value for these two forms is –2, and 5. The actual threshold value chosen for this work (50) is something of a compromise as the number of pixels count in one image at a particular value of x will usually vary from image to image due to the presence of image scanning noise, skew and offset. If the threshold value is set at a very low value (10 pixels for example), any slight noise added to the image will give a sufficient number of pixels to trigger the detection resulting in an incorrect offset. On the other hand, setting the value too high will impose unnecessary processing time on the form processing system and in some cases cause it to fail to locate the correct offset position due to the number of pixels in any x-position in the image being less than the threshold. Experimental results show that if this threshold is set at about 50 to 100 pixels, these problems can be avoided and a true offset obtained. Using these offset values, the pre-

printed entities on the filled form can be removed from the filled form using the position information provided by the blank form template obtained in the form structure identification process.

Although the image-offset problem can be resolved by employing this pixel count method, image skew can still cause problems for the subtraction process. Accurate skew detection and correction can be very computationally expensive and time consuming [50,51] whilst fast skew correction methods generally produce inaccurate and inefficient restorations (de-skew). Nevertheless, most of the current skew detection and correction techniques can correct the skew angle to within a range of +/-(1-3)° using a relatively fast and coarse approach [52,53,54,55]. Thus, when a subtraction approach is employed in a form extraction system, this skew angle must be taken into consideration otherwise the removal of the pre-printed entities will be incomplete. If we assume that the maximum skew angle is +/- 3°, the correlation Position Errors (PE) will then be +/- (tan (3°) x W) pixels in the Y-axis direction and +/- (tan (3°) x H) pixels in the X-axis direction. Fig. 3-8 shows this for a form's line entity.

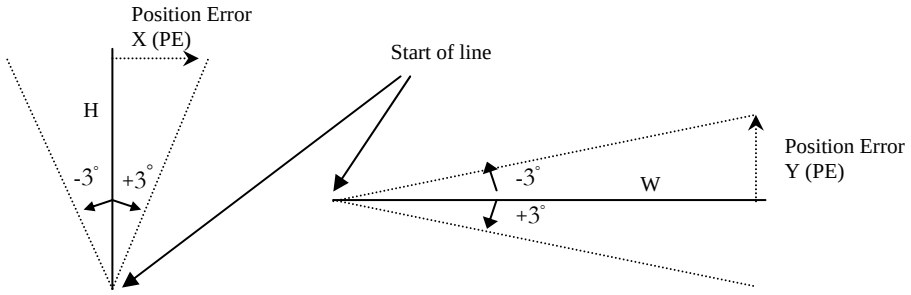


Fig 3-8. The X & Y position errors (in term of number of pixels) determined by the width (W) and height (H) of the entities and the skew angle.

Therefore, when an entity is subtracted from the image, a margin must be allocated to accommodate these position errors. For example, when removing a pre-printed entity with a width x height of 100 x 100 pixels and a skew angle of 3°, the subtracted area in the image must be 110 x 110 pixels (fig 3-9).

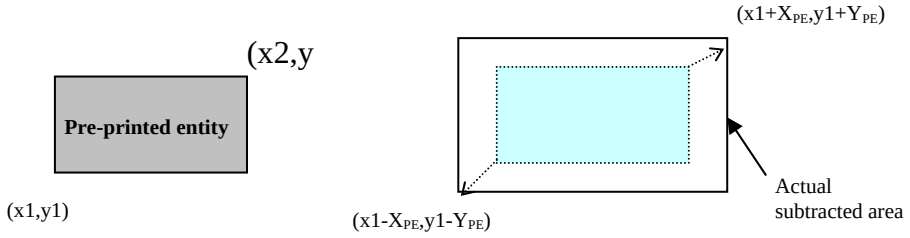


Fig 3-9. To accommodate the position errors induced by image skew, a margin is added to the entities size in order to ensure a complete removal of the pre-printed object from the filled form image.

The main disadvantages of this added margin method is that the removal process will occasionally remove some portion of the filled-in text, causing degradation in the extracted data quality. This problem mainly occurs in the line removal process as the filled-in data is normally filled in near or even onto the line position. Employing some of the more recently proposed text restoration algorithms (over 96% of accuracy results have been reported) [56, 57, 58, 59, 60] can virtually help fully resolve this problem. However, due to the extra processing time these algorithms would incur and the small amount of data that is affected by this factor (less than 5%), these text-repairing algorithms were not implemented here. It is believed that by employing a text repairing method in the system there would be a slight increase in recognition results for the black and white extraction system which would reduce the magnitude of the performance gain claimed for the colour extraction system. However, this performance gain is achieved at the expense of extra processing time required and thus the colour system would show

a greater improvement in terms of processing speed over the black and white system.

Once the pre-printed entities have been removed from the filled form, a noise removal process is then applied to remove any objects (e.g. noise) that are too small to qualify as text. In this study, the targeted filled-in data is assumed to be handwritten words, hence any connected components that are less than 8 pixels square are considered as noise. This is due to the fact that with typical handwriting words scanned at 200dpi, a handwriting word that is smaller than 8 pixels square is difficult, if not impossible to produce. Unfortunately, this removal process does occasionally remove some portions of some words; for example the dots for 'i' and 'j'. However, as the handwriting recognizer that is used in this work is only looking for word level features, the removal of small dots does not affect the recognition performance. The final resultant image after this noise removal process is then saved as a black & white format for the recognition process.

3. EXPERIMENTAL PLATFORM

The system has been implemented in C++ programming language under a Windows98 platform on a Pentium Celeron 450Mhz personal computer. Forms were digitized using a Hewlett Packard HP3690C scanner at 200dpi in 24-bit RGB format (except in experiment I where forms were digitized at several different resolutions ranging from 100 to 300 dpi).

The methods developed are designed to work on any type of colour form that is filled-in with unconstrained cursive handwriting data. As there is no

commercial CSR package available on the market at the moment, the recognizer that was used in the work is a modified version of a prototype CSR that has been developed within the IRS group at The Nottingham Trent University [18]. This recognizer extracts the vertical bars, loops and cups from the word image and compares these features with a list of words (lexicon) and their pre-defined set of features. Every word in the lexicon is scored according to how well the features match with the target word's set of features. The lexicon words and their scores are then ranked, with the highest ranked word being the likeliest match with the target word. The working resolution for this recognizer is 200x100 dpi, with the capability of accepting off-line (static) word image data as input. It has been designed and optimized to recognize unconstrained, lower case Roman script from any writer, with the assumption that all word images are in bi-level facsimile format. To accommodate the need for this work, the recognizer has been extended to recognize upper, lower and mixed case words.

3.1 Experiment I: Colour Reduction and OCR Performance

As the colour content of an image is greatly reduced using the proposed method, a study is needed to access the colour reduction effect on the image quality. One of the parameters that can be used to measure the output image quality is the OCR rate. This experiment aims to evaluate the impact on the OCR performance of using such a colour reduction technique under different resolutions. Sixteen colour forms of different designs, each containing 2 to 4 colours are used to compute the OCR rate that could be achieved both with and without the quantization process applied.

Table 3-3 shows the experimental OCR* results that apply to the proposed colour reduction method output images and the original 24-bit full colour images. In total, there are 7596 machine printed characters for these 16 forms and the OCR rate is computed manually by counting the total number of correctly recognized characters over the total number of characters.

Scanning Resolution	OCR rate (24-bit full colour)	OCR rate (quantized image)	Improvement
100dpi	85.0%	90.4%	+4.6%
150dpi	99.2%	98.8%	-0.4%
200dpi	99.3%	99.3%	0%
250dpi	99.6%	99.6%	0%
300dpi	99.8%	99.8%	0%

Table 3-3. Overall OCR performance tested on 16 forms at various resolutions (total characters=7596).

The quantized form OCR rate is almost identical to that of the 24-bit full colour image OCR rate above 150dpi resolution with a moderate improvement observed at 100dpi (4.6% increase). These results suggest that although the colour content of the quantized images is reduced to less than 8 colours, there is no significant detrimental effect on the OCR performance at resolutions of 150dpi and above. Indeed, at 100dpi, the OCR performance on the quantized image is greater than that on the original image. This shows that at 100dpi the colour reduction method proposed is better than that employed by the OCR engine. This is believed to be due to the fact that at 100dpi, the image details are smeared and distorted so much so that the original OCR colour handling technique is not able to recover all of the image details (the engine is probably optimized for 200-300dpi images). Whereas, with the comparison method applied, some of the

* The OCR engine used in this work is TextBridge Pro 9.0 from Xerox Imaging System due to its ability to allow OCR at different image resolutions.

details can be recovered from the scanning process hence leading to an increase in the OCR rate. The slight decrease in OCR rate at 150 dpi is believed to be due to the 'thickening' effect of the proposed colour handling technique. This can be proved by the fact that the increase in miss-recognized characters are mainly amongst characters such as 'e', 'u' and 'k'. The quantization method tends to darken the 'hole' for a character such as 'e' (resulting in an OCR output as a 'c') and join the open end of characters such as 'u' & 'k' (resulting in an OCR output as 'o' and 'h' respectively). At resolutions higher than 200dpi, the OCR rates become identical and the miss-recognized characters are then mainly due to small font size printed characters which are too small to be recognized by the OCR software.

3.3.2 Experiment II: Extraction Efficiency

Two parameters have been used to measure the efficiency of the extraction system - precision and recall rate. Precision is calculated as the number of correctly extracted objects (connected components) over the total number of objects extracted, whilst recall rate is calculated as the number of objects that are correctly extracted over the total number of expected objects that can be extracted from a given form. The recall rate provides a quantitative value of the expected objects that the system can extract, whilst the precision rate provides a quantitative measure of how many of the extracted objects are correct.

Precision= Number of correctly extracted objects / Total number of objects extracted

Recall = Number of correctly extracted objects / Total number of expected objects

For this experiment, another 16 different types of forms each containing a different type of layout design and number of colours were selected at random.

All of these forms were filled-in by a single writer using various types of colour pen. To compare the extraction efficiency, a black & white based extraction system was constructed and an intensity-based threshold binarization method was used to convert the 24-bit colour images to black & white images, i.e.

If $I < I_{threshold}$ (then pixel=black)
else if $I >= I_{threshold}$ (pixel=white)
where $I = (R+G+B)/3$

The pixel's intensity is the average value of the pixel's RGB value and the optimum $I_{threshold}$ value for the scanner is 190 (determined in section 3.1.1). The extraction method adopted on this black & white system is exactly the same as the subtraction technique used in the proposed colour-based extraction system except that instead of working in the filled-in colour domains only, it works with the entire binarized form images.

	Total	Best Case				Worst Case				Conventional			
Form	Target objects	Correct	Wrong	Precision	Recall	Correct	Wrong	Precision	Recall	Correct	Wrong	Precision	Recall
1	87	87	0	100.0%	100.0%	85	7	92.4%	97.7%	84	0	100.0%	96.6%
2	73	73	0	100.0%	100.0%	73	6	92.4%	100.0%	71	0	100.0%	97.3%
3	104	104	0	100.0%	100.0%	104	1	99.0%	100.0%	104	2	98.1%	100.0%
4	99	99	0	100.0%	100.0%	99	0	100.0%	100.0%	98	0	100.0%	99.0%
5	209	209	0	100.0%	100.0%	206	0	100.0%	98.6%	194	1	99.5%	92.8%
6	134	134	0	100.0%	100.0%	134	7	95.0%	100.0%	129	0	100.0%	96.3%
7	267	267	0	100.0%	100.0%	267	17	94.0%	100.0%	262	17	93.9%	98.1%
8	229	229	1	99.6%	100.0%	229	0	100.0%	100.0%	229	0	100.0%	100.0%
9	134	134	0	100.0%	100.0%	134	3	97.8%	100.0%	134	2	98.5%	100.0%
10	124	124	0	100.0%	100.0%	123	0	100.0%	99.2%	122	12	91.0%	98.4%
11	158	158	0	100.0%	100.0%	146	0	100.0%	92.4%	118	8	93.7%	74.7%
12	121	121	1	99.2%	100.0%	120	4	96.8%	99.2%	121	1	99.2%	100.0%
13	155	155	0	100.0%	100.0%	155	0	100.0%	100.0%	154	4	97.5%	99.4%
14	130	130	0	100.0%	100.0%	130	40	76.5%	100.0%	130	11	92.2%	100.0%
15	212	212	0	100.0%	100.0%	204	0	100.0%	96.2%	204	0	100.0%	96.2%
16	83	83	0	100.0%	100.0%	83	0	100.0%	100.0%	83	0	100.0%	100.0%
Average	145	145	0.13	99.9%	100.0%	143	5	96.6%	98.6%	140	4	97.2%	96.6%

Table 3-4. Comparison of the proposed colour form-extraction system recall and precision rate under worst and best cases to a black and white form extraction system

Table 3-4 shows the comparison of experimental results between the proposed colour extraction method and the black & white extraction method. On average, the colour-based extraction method gives a worst case recall rate of 98.6% and a best case recall rate of 100%. This compares with a 96.6% recall rate for the black and white extraction method. Thus, there is a clear 2 to 3% recall rate gain for an extraction system that utilizes colour information over a black & white extraction system. However, the precision for the worst case in the colour-based extraction system is lower than that of the black & white system (96.6% compared to 97.2%). This is due to the fact that since the black & white extraction system has a lower recall rate, fewer objects are extracted. This results in a lower number of unwanted objects being passed through to the output. When taken together though, we believe the worst case recall and precision rate for a colour based extraction system are similar to those of the black & white extraction system.

	Dimension		Load, Quantise & Extract (Colour-based)		Binarise + extract (sec)
	X	Y	Best Case (sec)	Worst Case (sec)	(Black & White system)
Form01	1320	852	2.14	10.98	8.23
Form02	1532	1095	2.91	5.33	7.58
Form03	1446	732	1.87	7.14	6.65
Form04	1422	736	1.80	6.20	5.50
Form05	1140	811	1.76	9.00	8.07
Form06	1520	1540	3.90	16.73	14.88
Form07	1465	1492	3.73	9.28	12.12
Form08	1422	1525	3.13	10.98	11.70
Form09	1496	1552	3.90	15.92	14.07
Form10	1254	1181	2.69	6.37	5.38
Form11	1614	2280	6.42	29.63	28.73
Form12	1454	2087	5.33	13.19	15.16
Form13	1594	2268	6.49	13.79	15.11
Form14	1492	1130	2.85	6.37	5.06
Form15	1508	1520	3.79	10.99	10.32
Form16	1477	720	1.97	4.22	7.63
Average	1447	1345	3.42	11.01	11.01

Table 3-5. Comparison of processing time required under different filled data colour conditions for the colour-based extraction system and the black and white extraction system

Table 3-5 shows the total processing time required for each of the forms under different colour conditions. As expected, the best case for the colour extraction method is when the filled data colour is of a different colour to that used in the blank form. The worst case is when the filled data colour is of the same colour as the second most dominant colour in the blank form (the most dominant colour for a document usually being the background colour). From the results shown in table 3-5, it can be seen that, in the best case, the proposed colour form extraction method is 3.22 times faster than that of the black & white system. However, when the filled data colour is the same as one of the blank form colours (worst case) then the colour-based system performs with almost identical speed to the black & white extraction system.

3.3.3 Experiment III: Extracted data quality

This experiment aims to assess the extracted data quality by comparing the CSR performance applied to the colour-based and the black & white extracted output images. In this experiment, 3 more different types of form were used each filled by 10 different writers using a colour pen that is a different colour to the colours used in the blank form (blue, black & green see Figure 12 for samples). The writers were deliberately instructed to use upper case only as this was found to be the most common and natural style for data entry in form documents. Indeed, when a lower case word restriction was initially imposed, people regularly miss-spelt words when trying to fill-out a form due to the unnatural mode of data entry. A holistic cursive word recognizer [61] was then used to assess the CSR performance of the extracted output from the colour-based and black & white extraction methods. This recognizer extracts the word features from the word

image and compares those features to the database. It then arranges the lexicons in the database from the highest match word (with the highest edit distance score) to the lowest match word.

	Colour extraction method	Black & White extraction method (without text repairing)	Overall Improvement
Form01 (266 words, 132 word lexicon)	58% @top 1	49% @top 1	+9%
Form02 (314 words, 215 word lexicon)	56% @top 1	49% @top 1	+7%
Form03 (256 words, 189 word lexicon)	61% @top 1	50% @top 1	+11%
Overall (836 words)	58% @top 1	49% @top 1	+9%

Table 3-6. The holistic recogniser performance when tested on the extraction output images that are extracted with and without the colour information

Table 3-6 shows the CSR results obtained. As there is no broken text repairing technique implemented in either system and as the recognizer is originally designed for lower case words only, the CSR performance is lower than that reported in [61]. However, the results do demonstrate a clear 9% overall improvement for the colour-based extraction method as compared to the black & white extraction system. This is due to the fact that when colour is not used to extract the data, the line removal process will degrade the text quality, especially when there are a lot of words or characters that are overlapping with the form lines or boxes. The text re-construction methods developed elsewhere [16] would reduce this problem greatly, but at the expense of incurring more computational overhead. Thus, a black & white extraction system that incorporates text-repairing algorithm could approach the CSR rate of the colour extraction system but at the expense of requiring more processing time.

4. CONCLUSIONS

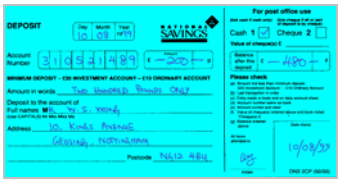
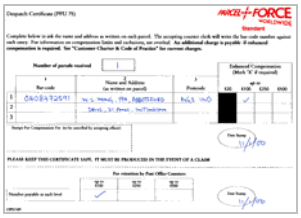
A new colour-based form extraction system has been presented. It is shown to be more efficient than a form extraction system that doesn't utilize any colour information. The effect of the colour reduction process on the image quality has been determined. The results suggest that the use of such a colour reduction strategy will aid content extraction and is shown to work well even with a massive reduction in the total number of colours in an image. The experimental results also demonstrate that at low resolution (100dpi), the colour reduction method is better than that employed by the OCR engine.

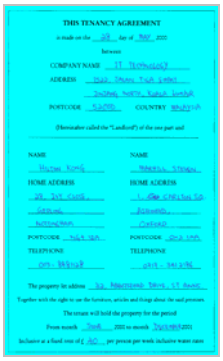
The extraction accuracy, recall rate and speed of the colour-based text extraction system has also been determined and compared to an extraction system that does not utilize colour information. With the proposed colour reduction method, colour information has been used successfully to reduce the system complexities and computational costs. By adopting colour features in form processing, the extraction speed has been increased up to 3.22 times. The extraction accuracy and recall rate are also shown to improve when using colour information to extract filled data from forms.

The extracted data quality has also been determined by examining the recognition results obtained from a CSR system applied to the output images from both systems. The experimental results suggest that an extraction system that utilizes colour information does produce a better output image quality than that of a black & white system.

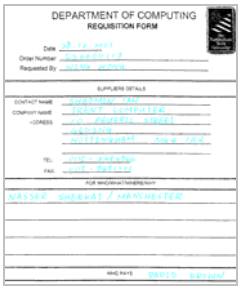
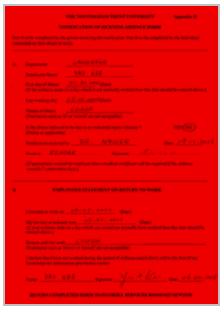
This paper has thus demonstrated the successful use of colour in an automatic form processing system. The processing time required could probably be further

improved if the colour handling method could be implemented in hardware or a faster platform. Future work aims to address further colour variation problems such as aging, folding and printing imperfections. Furthermore, it is well understood amongst the research community that end-users desire independent verification of findings across a large body of samples. In the case of hand filled documents collection of large datasets is fraught with data protection and privacy issues and creation of forms for the purpose of experimentation is rather resource intensive. It is hoped that the recent move towards collection of ground truth data such as those initiated but the ICDAR community will address this problem.



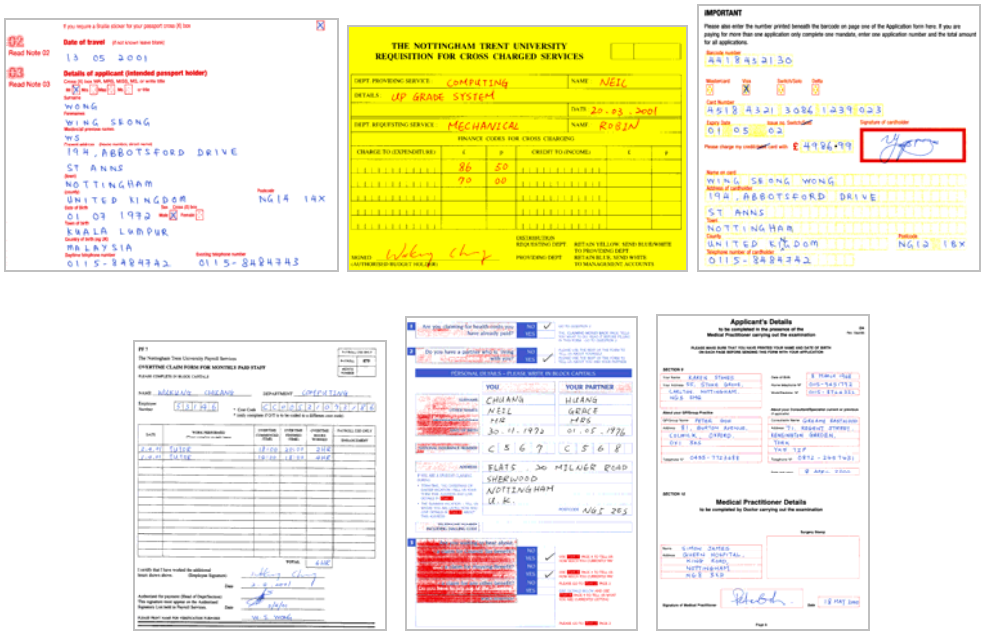
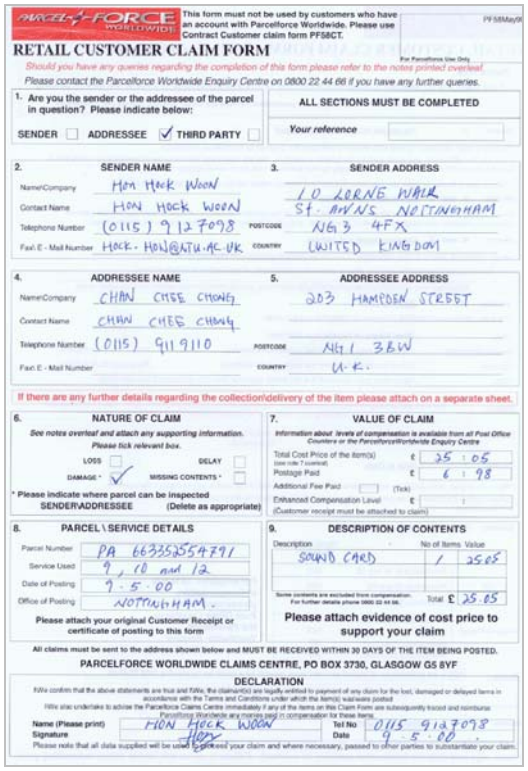


Fig 11. The 16 colour forms used in the second experiment.



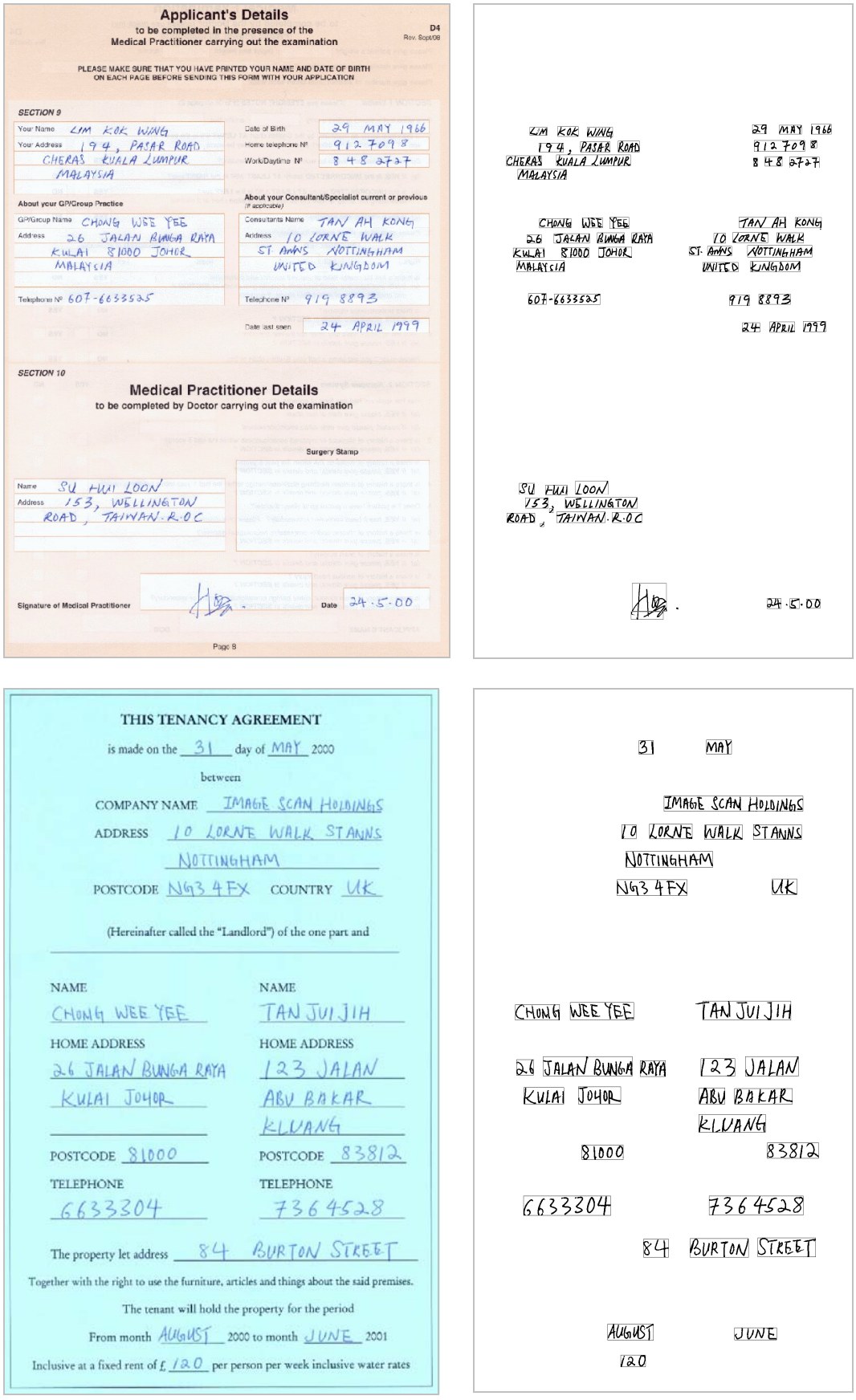


Fig 12. Samples of forms used in the third experiment.

5. REFERENCE

1. Y.Y. Tang and J Liu, "Information Acquisition and Storage of Forms in Document Processing", Proceeding International Conference on Document Analysis and Recognition, pp. 170-174, 1997
2. Richard Casey, David Ferguson, K. Mohiuddin and Eugene Walach, "Intelligent Forms Processing System", Machine Vision and Applications, Vol. 5, pp. 143-155, 1992
3. Michael D Garris and Darrin L Dimmick, "Form Design for High Accuracy Optical Character Recognition", IEEE transactions on Pattern Analysis and Machine Intelligence, Vol 18, No. 6, pp. 653-656, June 1996.
4. H.S. Baird, "The Skew Angle of Printed Documents", Proceeding of Society of Photographic Scientists and Engineers, Vol. 40, pp. 21-27, 1987
5. S.T. Hinds, J.L. Fisher and D.P. d'Amato, "A Document Skew Detection method Using Run-Length Encoding and Hough Transform", International Conference on Pattern Recognition, Vol. 1, pp. 464-468, 1990
6. A. Henig, G. Raza, N. Sherkat and R.J. Whitrow, "Detecting a Document's Skew: A Simple Stochastic Approach", Vision Interface'97, the 11th Canadian Conference on Computer Vision, Signal and Image Processing and Pattern Recognition, pp. 97-102, 1997
7. D. Wang and S.N. Srihari, "Analysis of Form Images", International journal of Pattern Recognition and Artificial Intelligence, 8(5), 1031-1052, 1994
8. X. Ye, M. Cheriet and C.Y. Suen, "A Generic System to Extract and Clean Handwritten Data from Business Forms", Proceedings 7th International Workshop on Frontiers in Handwriting Recognition, pp. 63-72, 2000
9. S. W. Lam, L. Javanbakht and S.N. Srihari, "Anatomy of a Form Reader", Proceedings International Conference on Document Analysis and Recognition", pp. 506-508, 1993
10. S.L. Taylor, Richard Fritzson and J.A. Pastor, "Extraction of Data from Preprinted Forms", Machine Vision and Applications, Vol. 5, pp. 211-222, 1992
11. R. Casey, D. Ferguson, K. Mohiuddin and E. Walach, "Intelligent Forms Processing System", Machine Vision and Applications, Vol. 5, pp. 143-155, 1992
12. M. Okada and M. Shridhar, "A Morphological Subtraction Scheme for Form Analysis", Proceedings of International Conference on Pattern Recognition, pp. 190-194, 1996
13. J. Yuan, Y.Y. Tang and C.Y. Suen, "Four Directional Adjacency Graphs (FDAG) and their Applications in Locating Fields in Forms", Proceedings of International Conference on Pattern Recognition, pp. 752-755, 1995
14. B. Yu and A.K. Jain, "A Generic System For Form Dropout", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 18, No. 11, pp. 1127-1134, 1996
15. B. Yu and A.K. Jain, "A Form Dropout System", Proceedings of International Conference on Pattern Recognition, pp. 701-705, 1996
16. S. H. Kim, S. H. Jeong and H.K. Kwag, "Line Removal and Character Restoration Using BAG Representation of Form Images", Proceedings 7th International Workshop on Frontiers in Handwriting Recognition, pp. 43-52, 2000
17. L. Xingyuan, D. Doermann, W.G. Oh and W. Gao, "A Robust Method for Unknown Form Analysis", International Conference on Document Analysis and Recognition, pp. 531-533, 1999
18. L.Y. Tseng and R.C. Chen, "Recognition and Data Extraction of Form Documents Based on Three Types of Line Segments", Pattern Recognition, 31(10), pp. 1525-1540, 1998
19. J.L. Chen and H.J. Lee, "An Efficient Algorithm for Form Structure Extraction Using Strip Projection", Pattern Recognition, 31 (9), pp. 1353-1368, 1998

20. J.L. Chen and H.J. Lee, "A Novel Form Structure Extraction Method Using Strip Projection", Proceedings of International Conference on Pattern Recognition, pp. 823-827, 1996
21. K. Ueda, "Extraction of Signature and Seal Imprint from bankchecks by Using Color Information", Proceedings International Conference on Document Analysis and Recognition", pp. 665-668, 1995
22. John Woods, "Recognition Technology for Data Entry – A Guide and Directory", ISBN 0-900458-68-2, Cimtech Ltd Publication, 1995
23. J.P. Braquelaire and L. Brun, "Comparison and Optimisation of Methods of Color Image Quantization", IEEE Transactions on Image Processing, Vol. 6, No. 7, pp. 1048-1052, 1997
24. M.S. Shyu and J.J. Leou, "A Genetic Algorithm Approach to Color Image Enhancement", Pattern Recognition 31, No.7, pp. 871-880, 1998
25. C.Y. Yang and J.C. Lin, "Color Quantization by RWM-cut", Proceeding International Conference Document Analysis and Recognition, pp. 669-672, 1995
26. D.C. Tseng, Y.F. Li and C.T. Tung, "Circular Histogram Thresholding For Color Image Segmentation", Proceeding International Conference Document Analysis and Recognition, pp.673-676, 1995
27. D. Zugaj and V. Lattuati, "A New Approach Of Color Images Segmentation based On Fusing Region and Edge Segmentations Outputs", Pattern Recognition, Vol. 31, No. 2, pp. 105-113, 1998
28. D.C. Tseng, Y.F. Li and C.T. Tung, "Circular Histogram Thresholding for Color Image Segmentation", International Conference Document Analysis and Recognition, pp. 673-676, 1995
29. W.Y. Chen and S.Y. Chen, "Adaptive Page Segmentation For Color Technical Journals' Cover Images", Image and Vision Computing, pp. 855-877, 1998
30. A.K. Jain and Bin Yu, "Automatic Text Location In Images And Video Frames", Pattern Recognition 31, No. 12, pp. 2055-2076, 1998
31. Y. Zhong, K. Karu, A.K. Jain, "Locating Text In Complex Color Image", Pattern Recognition 28, pp. 1523-1535, 1995
32. H. Hase, T. Shinokawa, M. Yoneda, M. Sakai and H. Maruyama, "Character String Extraction from a color Document", International Conference Document Analysis and Recognition, pp. 75-78, 1999
33. N. Sherkat, Dhiraj Mighlani, R.J. Whithrow, "A Descriptive Retrieval Engine For Image Database", Proceedings of IEE colloquium on Intelligent Image databases, ISSN 0963-3308, pp. 11/1 – 11/6, 1997
34. H. Kasuga, M. Okamoto and H. Yamamoto, "Extraction Of Characters From Color Documents", IS&T/SPIE Conference on Document Recognition and Retrieval VII, pp. 278-285, 2000
35. M. Worring and L. Todoran, "Segmentation of Color Documents By Line oriented clustering using Spatial information", Proceeding International Conference Document Analysis and Recongition, pp. 67-70, 1999
36. C. Strouthopoulos, N. Papamarkos and A. E. Atsalakis, "Text extraction in complex color documents, Pattern Recognition", Volume 35, Issue 8, August 2002, Pages 1743-1758
37. Hirobumi Nishida and Takeshi Suzuki, "Correcting show-through effects on scanned color document images by multiscale analysis", Pattern Recognition, Volume 36, Issue 12, December 2003, Pages 2835-2847
38. R. Schettini, C. Brambilla, G. Ciocca, A. Valsasna and M. De Ponti, "A hierarchical classification strategy for digital documents", Pattern Recognition, Volume 35, Issue 8, August 2002, Pages 1759-1769
39. C. Connolly and T. Fliess, "A Study of Efficiency and Accuracy in the transformation from RGB to CIELAB Color Space", IEEE Transactions On Image Processing, Vol. 6, no. 7, pp.1046-1048, 1997
40. M. Celenk, "A Colour Clustering Technique for Image Segmentation", Computer Vision, Graph, Image Processing, Vol. 52, pp. 145-170, 1990
41. Y.W. Lim and S.U. Lee, "On the Colour Image Segmentation Algorithm Based on the Thresholding and the Fuzzy C-Means Techniques", Pattern Recognition, 23 (9), pp. 935-952, 1990

42. C. Kotropoulos, E. Auge and I. Pitas, "Two-Layer Learning Vector Quantizer for Colour Image Quantization", Signal Processing IV: Theories and Applications, Elsevier, pp. 1177-1180, 1992

43. T. Kohonen, "Self-Organizing Maps", Springer Series in Information Sciences, Springer-Verlog, 30, 1995

44. Steven W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing", California Technical Publishing, chapter 23, ISBN 0-9660176-3-3, 1997

45 Michael D. Garris, "Correlated Run Length Algorithm (CURL) for Detecting Form Structure Within Digitized Documents", 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 415-424, UNLV, April 1994.

46 Jaekyu Ha, Ihsin T. Phillips, and Robert M. Haralick, "Document Page Decomposition Using Bounding Boxes of Connected Components of Black Pixels", SPIE Vol. 2422, pp. 140-151, 1995.

47 Juan F. Arias, Atul Chhabra and Vishal Misra, "Finding Straight Lines in Drawing", IEEE proceedings International Conference Document Analysis and Recognition, pp. 788-791, 1997

48 Opas Chutatape and Linfeng Guo, "A Modified Hough Transform For Line Detection and Its Performance", Pattern Recognition 32, pp. 181-192, 1999

49 Bin Kong, Su Chen and Robert M. Haralick, "Automatic Line detection in Document Images Using Recursive Morphological Transforms", SPIE Vol. 2422, pp.163-174, 1995

50 Yi Kai Chen and Jhing Fa Wang, "Skew Detection and Reconstruction Based on Maximization of Variance of Transition-counts", Pattern Recognition 33, pp. 195-208, 2000

51 Yonki Min, Sung Bae Cho and Yillbyung Lee, "A Data Reduction for Efficient Document Skew Estimation Based on Hough Transformation", Proceedings of ICPR, pp. 732-736, 1996

52 Nadine Rondel and Gilles Burel, "Cooperation of Multi-Layer Perceptions for Estimation of Skew Angle in Text Document Images", 3rd Proceeding International Conference on Document Analysis and Recognition, pp. 1141-1144, 1995

53 Andreas Hennig, "Recognising a Page of Unconstrained Cursive Handwriting", PhD thesis, Department of Computing, The Nottingham Trent University, 1999

54 Huei Fen Jiang, Chin Chuan Han and Kuo Chin Fan, "A Fast Approach to Detect and Correct Skew Documents", Proceedings of ICPR, pp. 742-746, 1996

55 Avanindra and Subhasis Chaudhuri, "Robust Detection of Skew in Document Images", IEEE transactions on Image Processing, Vol. 6, No. 2, pp. 344-349, 1997

56 S. H. Kim, S. H. Jeong and H.K. Kwag, "Line Removal and Character Restoration Using BAG Representation of Form Images", 7th International Workshop on Frontiers in Handwriting Recognition, pp. 43-52, September 2000

57 XiangYun Ye, Mohamed Cheriet and Ching Y Suen, "A Generic System to Extract and Clean Handwritten data From Business Forms", 7th International Workshop on Frontiers in Handwriting Recognition, pp. 63-72, September 2000

58 D Wang and S. N. Srihari, "Analysis of Form Images", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8 No. 5, pp. 1031-1052, 1994

59 Y. Chung, K. Lee, J. Paik and Y. Lee, "Extraction and Restoration of Digits Touching or Overlapping Lines", IEEE Proceedings of ICPR, pp. 155-159, 1996

60 Bin Yu and A. K. Jain, "A Generic System for Form Dropout", IEEE transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 11, pp. 1127-1134, 1996

61 N. Sherkat and T.J. Allen, "Whole Word Recognition in Facsimile Images", 5th Proceeding International Conference Document Analysis and Recognition, pp. 547-550, 1999.