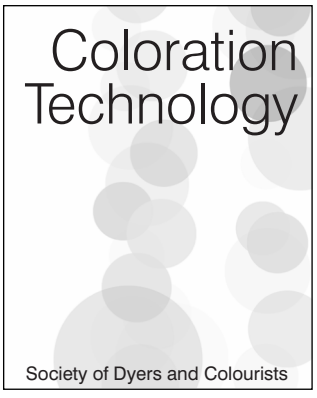
Recolouring digital textile printing design with high fidelity

J H Xin* and H L Shen

Institute of Textiles and Clothing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
Email: tcxinjh@inet.polyu.edu.hk

Received: 23 July 2003; Accepted: 22 October 2003



An algorithm of recolouring of polychromatic digital textile printing images is proposed based on the investigation of texture and colour distribution in different channels and image segmentation. A polychromatic image is first separated into monochromatic regions based on watershed transformation in CIELAB colour space. The initial markers are selected by hierarchical histogram analysis to eliminate the inherent drawbacks of over-segmentation in the watershed algorithm. Then the individual monochromatic regions can be mapped with different colours to obtain desirable designs. The artefacts in the boundaries of different regions are reduced by a technique of colour mixing through Gaussian blurring. The experimental simulation results indicated that the performance of the algorithm was quite good in both texture and colour fidelity.

Introduction

It is a common practice and very useful for designers to use existing designs as references for new designs [1]. Traditionally, a designer has to manually paint an image again using different colours to visualise the colour effect. Employing a computer aided design (CAD) system can certainly make the visualisation of the different colour effects simpler [2]. However, in most cases, only graphic designs created in the CAD system can be used for this purpose. It is difficult for a designer to visualise the different colour effect based on existing physical samples, such as a printed fabric or a graphic design printed on paper, etc. To solve this problem, so that a designer can easily manipulate colours based on an existing image, the algorithm of recolouring of a polychromatic digital texture image was developed in this study.

It is difficult to deal with a printed textile fabric since in most cases it consists of differently coloured regions. In addition, a textile fabric itself is textured, being plain woven, twill, single jersey, and so on. Therefore, to recolour a printed textile design, two obstacles need to be overcome, i.e. the segmentation of differently coloured regions and the colour remapping on those separated regions that are textured. Two algorithms, namely texture image segmentation of polychromatic images and colour remapping on monochromatic textured regions, were developed in this study to support the recolouring work.

The study of colour remapping of a texture image in this work deals with reproducing a colour image with high fidelity both in colour and texture so that the reproduced image is perceptually close to the target physical sample. Using the techniques developed in this work, it is possible to visualise the final coloured textured pattern before a print is actually produced. The segmentation involves the partition of a polychromatic texture image into disjoint regions such that it is perceptually homogeneous within each region and inhomogeneous between regions. Texture and colour are the most important characteristics in textile fabric images. Although the complicated interaction of texture and colour was investigated in the areas of texture synthesis and texture analysis, it is still far from being satisfactorily solved [3–5]. The study of the recolouring of a polychromatic texture image is not a trivial task, and the effects of texture and colour need to be carefully studied.

For image segmentation, there are four main approaches, namely histogram-based, edge-based, region-based and hybrid techniques [5]. As the histogram-based technique does not consider the spatial distribution of pixels, it often produces unsatisfactory segmentation results for colour images of natural scenes [5]. The edge-based technique detects the intensity variation between adjacent pixels and then regards them as regional boundaries based on an appropriate threshold decision. The region-based technique assumes that the values of pixels are very similar for the same region [6]. In order to obtain satisfactory results, the homogeneous criterion should be carefully determined [7]. The hybrid technique is a combination of both the edge- and region-based techniques and therefore can provide improved segmentation results. The watershed transformation used in this study is an example of this segmentation category [8–10]. The watershed transformation is generally applied on the gradient of an image, and considers the homogeneous criterion between connected pixels at the same time. In this current study however, considering there are sometimes no rapid changes between different regions after the smoothing of the texture, the watershed transformation is applied on the difference image instead of the gradient image to produce better segmentation results. The details of this method are discussed later in the paper.

Although many studies have been conducted in the research of texture synthesis, few of them have considered the problem of colour accuracy of the reproduced images [11,12]. Hong *et al*. proposed an approach for on-screen texture visualisation by assuming that the pixels in the reproduced image differed only in the luminance and had the same chromaticity coordinates [13]. Unfortunately, this assumption is too simplified to apply for the majority of texture images [14]. Some proprietary products claimed

that they were suitable for colour mapping of textile fabrics [15], however, so far, only single colour mapping can be performed and no study on the accuracy of colour has ever been published. In this current study, the interaction of texture and colour was investigated globally and locally to achieve satisfactory segmentation results and photorealistic recolouring effects with good colour accuracy for polychromatic textile printed fabrics were obtained.

Development of the algorithm of colour remapping

The texture images used in this study were dyed or printed textile fabrics and they were scanned using an Epson GT-10000+ flatbed colour image scanner. The fabric samples consisted of 16 differently woven patterns, and each pattern was dyed into five colours: green, orange, purple, pink and turquoise blue. In total, there were 80 monochromatic texture images. The polychromatic samples were printed fabrics with different colour ways. All these fabric samples were scanned at a resolution of 72 dpi so that the obtained images gave approximately equal visual appearances to those of the physical samples when viewed at a normal viewing distance of about 25–30 cm.

The *RGB* space is the most elementary colour space available as it directly corresponds to the input/output signal of imaging devices. For colour images, the pixel responses of the *RGB* channels are highly correlated and not independent [14,16]. According to the calculation devised in a previous study [14], the average channel correlation coefficient was found to be as high as 0.85 for 80 monochromatic texture images of textile fabrics. This high correlation coefficient indicates the existence of significant redundancy in *RGB* channels. It also indicates the possibility of deducing spatial distribution from one channel to another. This channel correlation is the precondition of the algorithm of colour remapping.

If $m_n(p)$ denotes the response of the pixel p in channel n, the deviation $\Delta m_n(p)$ can be calculated according to Eqn 1:

$$\Delta m_n(p) = m_n(p) - \bar{m}_n \tag{1}$$

where $\bar{m}_n$ is the average value of channel n (n = 1,2,3 or *RGB*, equivalently). It was found that the $\Delta m_n(p)$ distribution between any two different channels were linearly proportional, which means that a pixel contains proportional amounts of redness $\Delta m_1(p)$, greenness $\Delta m_2(p)$ and blueness $\Delta m_3(p)$. The relationship between the pixel deviations for two channels of a typical texture image is illustrated in Figure 1. It can be seen that the pixel deviations form a straight-line passing through the origin. This means that it is possible to calculate the deviation of one channel (i.e. redness) from those of other channels (greenness and blueness) [14].

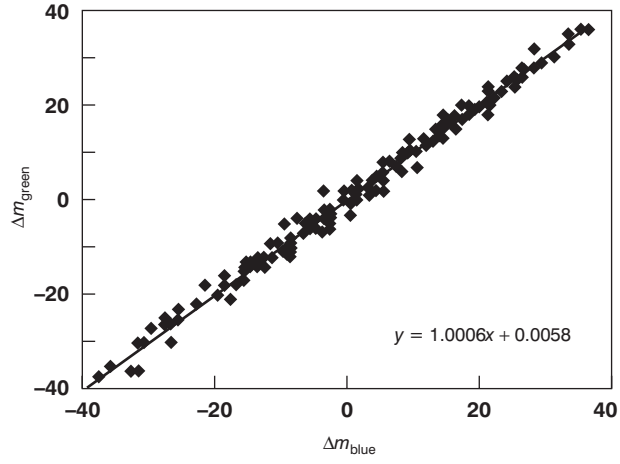


Figure 1 Relationship between pixel deviations for blue and green channels of a typical texture image

In a simplified case, it can be assumed that the deviation $\Delta m_n(p)$ is unrelated to mean colour m_n. In this case, if we want to recolour a texture image with a target mean colour (m_1', m_2', m_3'), the new colour of pixel p can be simply calculated as $[m_1' + \Delta m_1(p), m_2' + \Delta m_2(p), m_3' + \Delta m_3(p)]$. However, in a real situation $\Delta m_n(p)$ is related to mean colour m_n [14], therefore, the new pixel deviation $\Delta m_n'(p)$ should be calculated for different target mean colours. In this study, $\Delta m_n'(p)$ is calculated by linear interpolation using $\Delta m_n(p)$ and m_n (n = 1–3). More precisely, for pixel p, a term of proportionality $\delta(\bullet;p)$ can be defined to describe the colour deviation in each channel with respect to a new target mean colour in the recolouring process, the value of which can be calculated by linear interpolation using the existing inter-channel information. As there are various texture regions, a further term $g(\bullet)$ is used to describe the texture strength for the region under consideration, and then the new colour (m_n^c) for pixel p can be calculated according to Eqn 2.

$$m_n^c(m_n';p) = m_n' + g(m_n') \times \delta(m_n' + \Delta m_n(p);p) \tag{2}$$

For a monochromatic texture image, $g(\bullet)$ is a constant equal to 1.0, and for a polychromatic image, the term $g(\bullet)$ is relative to the statistical characteristic of the texture image, which will be discussed in the following section. As the target response m_n' may differ to the original mean colour m_n, the term of proportionality $\delta(\bullet;p)$ can be calculated using linear interpolation (Eqn 3, below) where $m_n(p)$ is sorted in ascending sequence. A typical shape of $\delta(\bullet;p)$ is shown in Figure 2. The value of $\delta(\bullet;p)$ is clipped into $\Delta m_1(p)$ and $\Delta m_3(p)$ at the two ends, respectively, to avoid errors caused by extrapolation. In-between $m_1(p)$ to $m_3(p)$, the proportionality given in Eqn 3 applies.

$$\delta(x;p) = \begin{cases} \Delta m_1(p) & 0 \le x \le m_1(p) \\ \dfrac{\Delta m_1(p) - \Delta m_2(p)}{m_1(p) - m_2(p)}\left[x - m_2(p)\right] + \Delta m_2(p) & m_1(p) < x \le m_2(p) \\ \dfrac{\Delta m_2(p) - \Delta m_3(p)}{m_2(p) - m_3(p)}\left[x - m_3(p)\right] + \Delta m3(p) & m_2(p) < x \le m_3(p) \\ \Delta m_3(p) & m_3(p) < x \le 255 \end{cases} \tag{3}$$

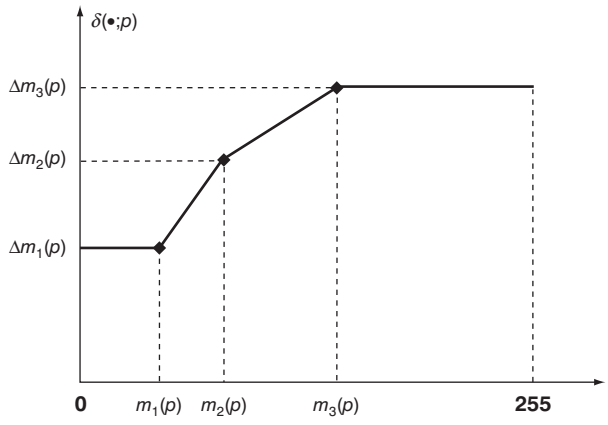


Figure 2 A typical shape of the proportionality term $\delta(\bullet;p)$ for $m_1(p)$, $m_2(p)$ and $m_3(p)$

An example of applying Eqn 2 is given as the following. Suppose the desired target colour ($m_1' = 25$, $m_2' = 45$, $m_3' = 87$) is chosen for colour remapping, the colour of pixel p in the original texture image is [$m_1(p) = 50$, $m_2(p) = 100$, $m_3(p) = 150$], along with its deviation to the average colour [$\Delta m_1(p) = 20$, $\Delta m_2(p) = 30$, $\Delta m_3(p) = 38$]. In this case, the new colour at pixel p can be calculated as the following: $m_1^c(m_1';p) = (25 + 20) + 20 = 65$; $m_2^c(m_2';p) = (45 + 30) + \{[(20 - 30)/(100 - 50) \times (75 - 100)] + 20\} = 100$; and $m_3^c(m_3';p) = (87 + 38) + \{[(30 - 38)/(150 - 100) \times (125 - 150)] + 30\} = 159$.

Image segmentation and recolouring

Algorithm of the watershed-based segmentation

For a polychromatic texture image, it is necessary to partition the whole image into several homogeneous regions before applying the algorithm of colour remapping. As the watershed-based segmentation performs well both in locating outlines and clustering pixels [9], it was adopted in this study. The watershed transform can be explained by the concept of flooding simulation, as shown in Figure 3.

Consider the image as a topographic surface and assume that holes are punched in each regional minimum of the surface. The surface is then slowly immersed in water. Starting from the minima at the lowest altitude, the water will progressively flood the catchment basins of the image. In addition, dams are raised at positions where the waters coming from two different minima would merge. At the end of this flooding procedure, each minimum is surrounded by dams delineating its associated catchment basin. The whole set of dams correspond to the watersheds and the input image is partitioned into different catchment basins.

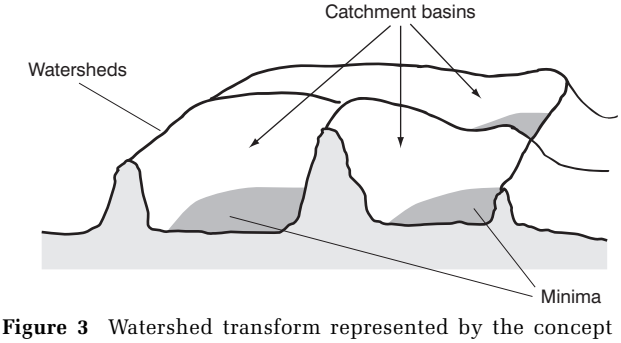


Figure 3 Watershed transform represented by the concept of flooding simulation

The watershed is usually applied on the gradient of an image to find the edges [called watershed of gradient (WG)], supposing that the gradient is relatively large at the boundary of different regions. However, for a large amount of textile fabric images, the boundaries tend to be relatively wide between different regions, within which the values of the pixels vary gradually from one region to another. Thus it is difficult to decide into which region these pixels should be classified. According to the homogeneous character of a region, it was hypothesised that the difference between pixels within the same region is smaller than that in a different region. Based on this hypothesis, a new algorithm of watershed of image difference (WD) is proposed instead of WG (Figure 4).

In order to compare the performance of the WD and WG algorithm, let us consider a one dimensional signal $s(j)$ (Eqn 4).

$$s(j) = \begin{cases} 100 & j \in [0,20) \\ 100 + (j-20) \times 4 & j \in [20,40) \\ 180 & j \in [40,60) \\ 100 + (j-60) \times 4 & j \in [60,80) \\ 100 & j \in [80,100] \end{cases} \quad (4)$$

The signal $s(j)$ consists of three parts, with two parts of level 100 standing at the left and right side, respectively, and one part of level 180 in the middle. Between the adjacent parts, there are two transitional boundaries in which the signal changes slowly with a step of 4 units. From the definition of the signal $s(j)$, the ideal outlines of the parts lie at positions $j = 30$ and $j = 70$, respectively. In order to simulate the effect of texture, 50% Gaussian noise with standard deviation of 6.0 is added to the signal (Figure 4a). The difference between $s(j)$ and level 100 and between $s(j)$ and level 180 are plotted in Figure 4b. The lower values of the two lines at positions between 0 and 100 are plotted in Figure 4c. The maxima in Figure 4c should represent the watersheds dividing different regions. It is clear that the watersheds in Figure 4c remain at the correct positions ($j = 30$ and $j = 70$), but those in Figure 4d are seriously biased by the simulated texture. Therefore, in the segmentation of the texture image, the algorithm of WD was used instead of the traditional algorithm of WG.

It is known that the problem of over-segmentation is the inherent drawback of the watershed-based algorithm [9]. To solve this, initial markers should be chosen for the algorithm of WD-based segmentation. The markers were decided by the method of hierarchical histogram-based segmentation, which is described as follows [17]:

- transform the *RGB* value of each pixel into CIE $L^*a^*b^*$ colour space;
- smooth the L^* channel (containing only texture information) using median filter [18];
- find the peaks and valleys of the histogram of channel a^*, and partition the image into a cluster of region R_a by histogram analysis;
- for every individual region in cluster R_a, build the sub-histogram of channel b^*, and further partitioning R_a into R_{ab};
- similar to the previous step, further partitioning is carried out for R_{ab} into R_{Lab} in channel L^*.

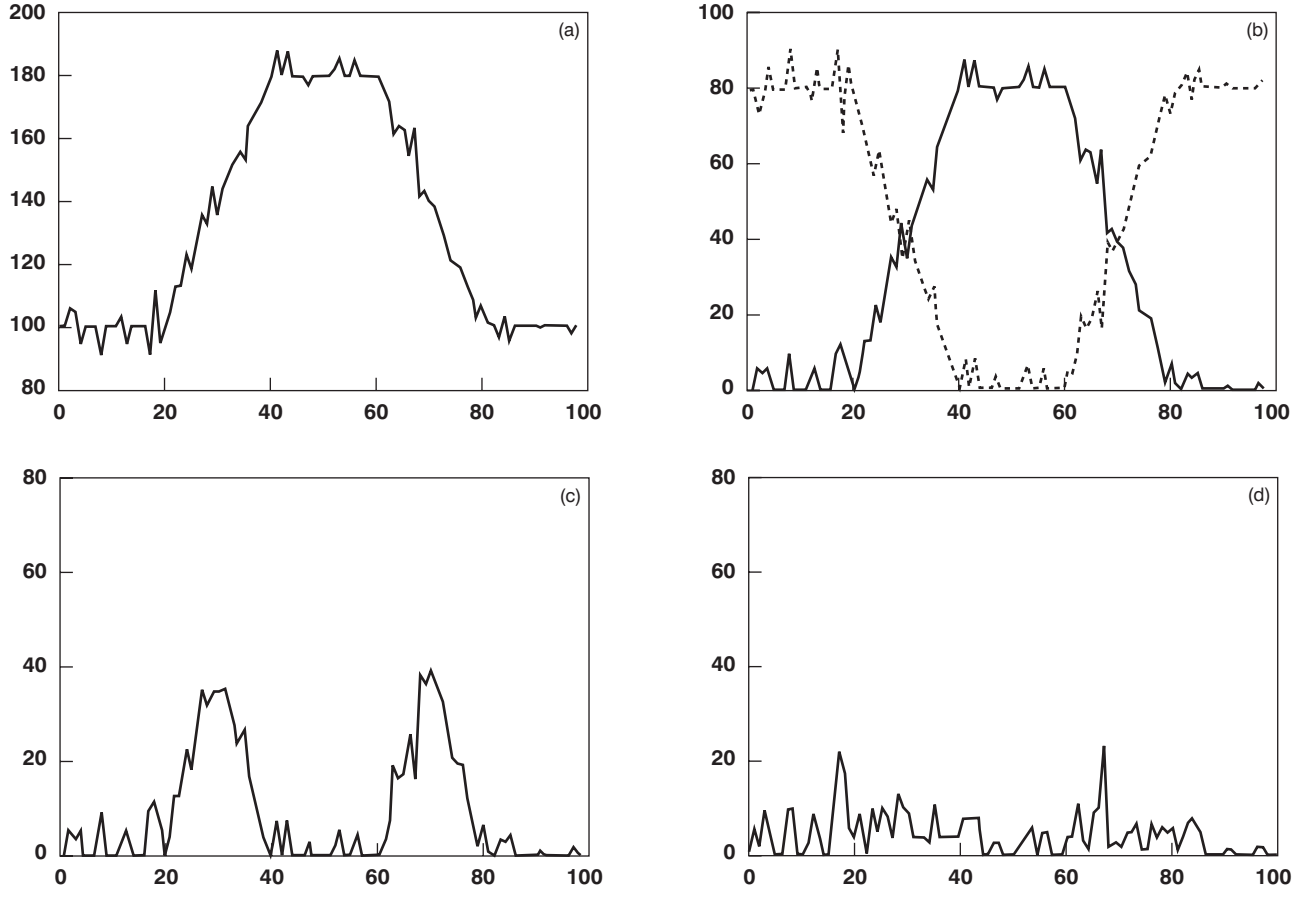


Figure 4 Comparison of image difference and image gradient using one-dimensional signal: (a) one-dimensional signal $s(j)$ with simulated texture; (b) difference between $s(j)$ and the two levels 100 (solid line) and 180 (dashed line); (c) lower values of the two lines in plot (b); (d) gradient of the signal in plot (a)

In the histogram analysis, the morphological filter was used to eliminate the noise, and the thresholds were carefully selected to reach a good segmentation result [17]. The final meaningless trivial regions can be either integrated into large regions or deleted interactively. Suppose there are C regions in the image, and the mean colour for the kth region is ${L_k}^0{a_k}^0{b_k}^0$. For every pixel in the image, the weighted distance to the mean colour of region k is calculated according to Eqn 5:

$$d_k = \sqrt{\eta[L(p)-L_k^0]^2+[a(p)-a_k^0]^2+[b(p)-b_k^0]^2} \qquad (5)$$

where $L(p)$, $a(p)$, $b(p)$ are the CIELAB attributes of pixel p, and η is the weight of the luminance. The value of η is defined as:

$$\eta = \frac{\overline{w}_a + \overline{w}_b}{2\overline{w}_L} \qquad (6)$$

where $\overline{w}_L$, $\overline{w}_a$ and $\overline{w}_b$ is the average histogram width of all the regions in channels L^*, a^* and b^*, respectively. As the variation of L^* (mainly texture information) is much stronger than that of a^* and b^* (mainly chromaticity information), the value of η is commonly in the range of 0 to 1. Find the minimum distance $d_{k0}(p)$ and the next to minimum distance $d_{k1}(p)$, if Eqn 7 holds:

$$d_{k0}(p) \le \min[\beta \times d_{k1}(p), T] \qquad (7)$$

where T is an experiential threshold (value defined by the user) and β is a distance ratio in the range of 0 to 1, then pixel p is selected as the marker of region $k0$, otherwise it should be reserved for classification in the DW algorithm. Figure 5† shows the segmentation of a polychromatic printed textile image. Figure 5b demonstrates the markers selected in the case of $\eta = 0.1$ and $\beta = 0.2$.

In order to simplify the coding of the WD algorithm, the list data structure was used to store the pixels for classification in ascending order, according to their distances to the mean colour of the regions. At every time of flooding, a pixel p is popped out and its neighbours are checked. If all of the already-labelled neighbours of p have the same label, this label is assigned as pixel p, otherwise, the label corresponding to the smallest distance is assigned to pixel p. The pseudo language of the WD algorithm is given in appendix I, and the segmentation result is shown in Figure 5c, in which the regions are represented by its average colour. The segmentation result obtained is quite good especially in the region boundaries.

† Figures 5,7–9 can be viewed in colour in the online version of this article (www.sdc.org.uk/publications/online.htm)

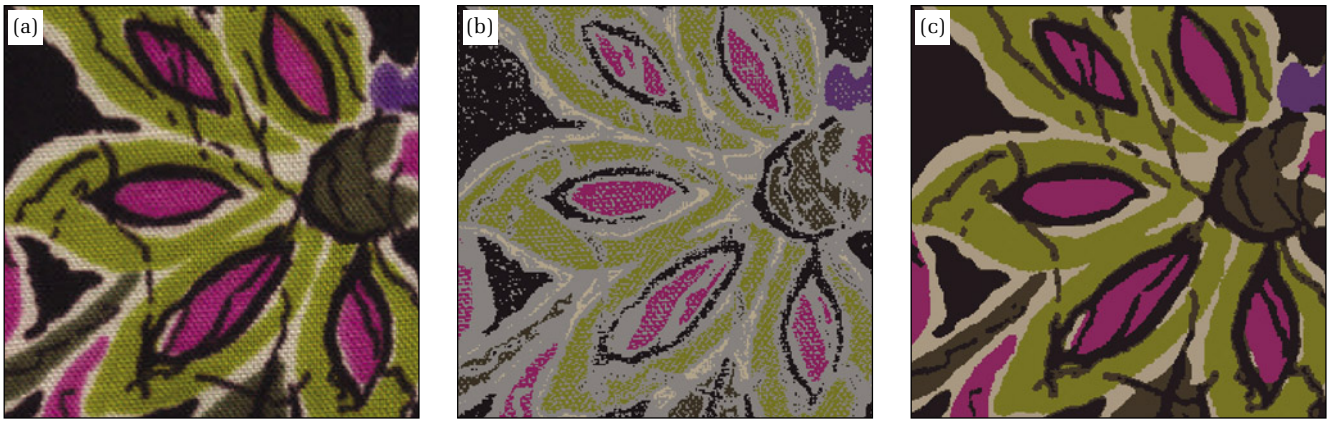


Figure 5 Segmentation of a polychromatic printed textile image: (a) original image; (b) pixels not selected as markers are shown in light-grey; (c) result of the WD-based segmentation

Recolouring of the polychromatic texture images

For every monochromatic region in a texture image, the corresponding luminance Y is calculated (Eqn 8) [18].

$$Y = 0.299R + 0.587G + 0.114B \tag{8}$$

As the texture in the textile fabric is mainly caused by the combination of yarns, there are usually no angular effects with strong contrast. Thus the histograms showing luminance Y for each individual region are quite similar to each other and contain only one peak. The width of the peak is directly related to the texture strength of each region. Shown in Figure 6 is the relationship of histogram widths with respect to the mean luminance of different regions for the texture image of Figure 5a.

After carrying out further investigations on various different texture images, it was found that the relationships were different, depending on the different properties of the physical samples such as material, colour and texture pattern. Figure 6 is only an example of the relationship for a certain image. Therefore, the practical way to describe a relationship between the texture strength and the luminance of a region is either to find the appropriate reasonable trends using data regression, or to simply apply interpolation and extrapolation [19]. If $t(\bullet)$ is the relationship, the term of texture strength $g(\bullet)$ for the kth region can be defined as shown in Eqn 9:

$$g(x) = \frac{t(x)}{t(\overline{Y}_k)} \tag{9}$$

where $\overline{Y}_k$ denotes the average luminance of the kth monochromatic region. Substituting Eqn 9 into Eqn 2 gives Eqn 10.

$$m_n^c(m_n', p) = m_n' + \frac{t(x)}{t(\overline{Y}_k)} \times \delta(m_n' + \Delta m_n(p); p) \tag{10}$$

For common images, especially for textile fabric images, there are usually no abrupt changes along the outlines in-between the regions, due to the inter-reflection in the boundary [20] and diffusion of colorants. In this study, we use the term colour mixture to refer to this characteristic of the texture image. However, when different solid colours are remapped to the segmented regions, there will be abrupt changes at the outlines in-between these regions. Therefore, to simulate a natural image, this problem should be solved. The method proposed in this study is based on colour mixing using Gaussian blurring (Eqns 11 and 12):

$$\text{gauss}(x, y) = K_0 \times \exp\left(-\frac{x^2 + y^2}{2\sigma_0^2}\right) \tag{11}$$

$$\text{gauss}(x) = K_1 \times \exp\left(-\frac{x^2}{2\sigma_1^2}\right) \tag{12}$$

where K_0,K_1 are the constants for normalisation and σ_0,σ_1 are the standard deviation for the two-dimensional (Eqn 11) and one-dimensional (Eqn 12) Gaussian filter, respectively. The gauss(x,y) is applied on a^*b^* chromaticity plane with $\sigma_0 = 1.0$, and gauss(x) is applied on L^* channel with $\sigma_1 = 0.4$. Here, $\sigma_0 > \sigma_1$ is assumed, because colour information

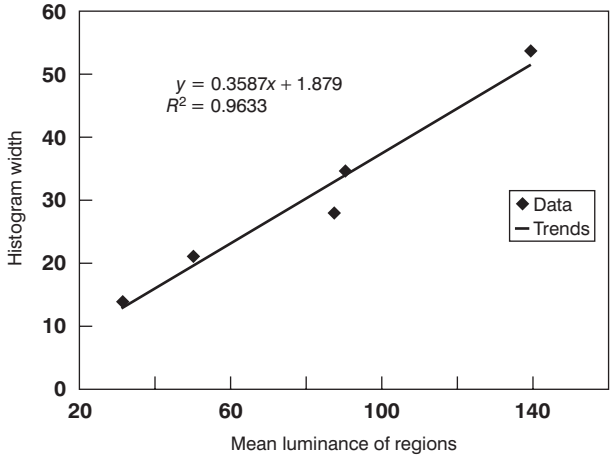


Figure 6 Relationship between the histogram width and the mean luminance of regions

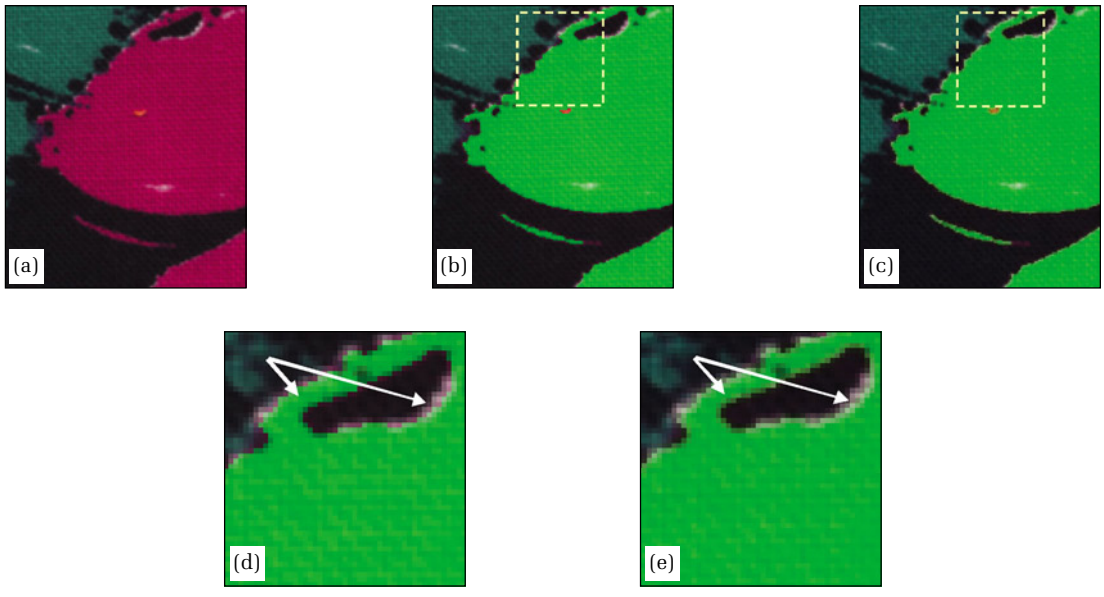


Figure 7 Comparison of the algorithms of colour remapping with or without colour mixing treatment: (a) original image; (b) colour remapping excluding colour mixing; (c) colour remapping including colour mixing; (d) magnified image of the marked rectangular area in image (b); (e) magnified image of the marked rectangular area in image (c) [the white arrows in (d) and (e) indicate the improvement of image fidelity after colour mixing treatment]

is affected more by the inter-reflection and diffusion. The comparison between the algorithms including and excluding colour mixing treatment is shown in Figure 7.† The residual red pixels at the boundary of dark regions were correctly eliminated when including colour mixing treatment, which indicated the capability of the colour mixing treatment in dealing with artefacts along the outlines of the regions.

Experiment and simulation

The process of the whole application can be briefly outlined as follows:

- separate the different regions in the texture image using the WD-based segmentation algorithm;
- find the relationship between the texture strength and the mean luminance from the separated regions;
- change the colours of the regions according to Eqn 10;
- perform colour mixture using Gaussian filtering using Eqns 11 and 12.

The algorithm of colour remapping discussed earlier in this paper is the core in the recolouring technique. To evaluate the colour accuracy of the colour remapping algorithm, monochromatic original and target images were used. The mean colour of the target image was mapped on the original image to simulate a new texture image. Details can be found in a previous study [14].

The texture patterns of the original images and the target images are selected to be similar so that the colour and texture appearances of the target and reproduced images are comparable. The performance of the algorithm can be evaluated either using a psychophysical experiment [14] or by means of colour image similarity measurement [21–23]. As there was no pixel correspondence between the reproduced and the target texture images, the calculation of colour image difference cannot be directly applied [21,22]. In this study, the evaluation of image similarity was conducted using a statistical approach instead, i.e. histogram intersection [23].

For each pixel in a texture image, its colour difference to the mean colour is calculated in the display colour space. The histogram of the colour difference is then constructed. Let I be the target texture image, I' be the reproduced image and $H_j(\bullet)$ be the frequency of the jth bin of the histogram, then the image colour similarity can be calculated by the method of histogram intersection (Eqn 13) [23]:

$$p(I,I') = \frac{\sum_{j=1}^{M} \min[H_j(I), H_j(I')]}{\sum_{j=1}^{M} H_j(I)} \qquad (13)$$

where M is the number of bins in the histograms. When the texture distribution of the two image, I and I', are similar, the term $p(I,I')$ should be appropriate for the assessment of the image colour similarity. The more similar the target image and the reproduced image, the closer the term $p(I,I')$ approaches 1.0. In this study, image similarity was calculated using the histogram intersection method. The mean of the image similarity was found to be 0.83 for the 10 pairs of target and reproduced images. Considering that the texture details of the two images for comparison were not identical, this image similarity result was quite satisfactory.

In addition to the colour remapping of monochromatic texture images, examples of recolouring on polychromatic texture images are also shown in Figures 8 and 9.† In these figures, the separated regions are represented by their average colour in the label images. The simulation tests showed that the appearances of recolouring results were of high fidelity.

Figure 8 First example of recolouring including segmentation and colour remapping; (a) original image; (b) label image after segmentation; (c) new simulated image

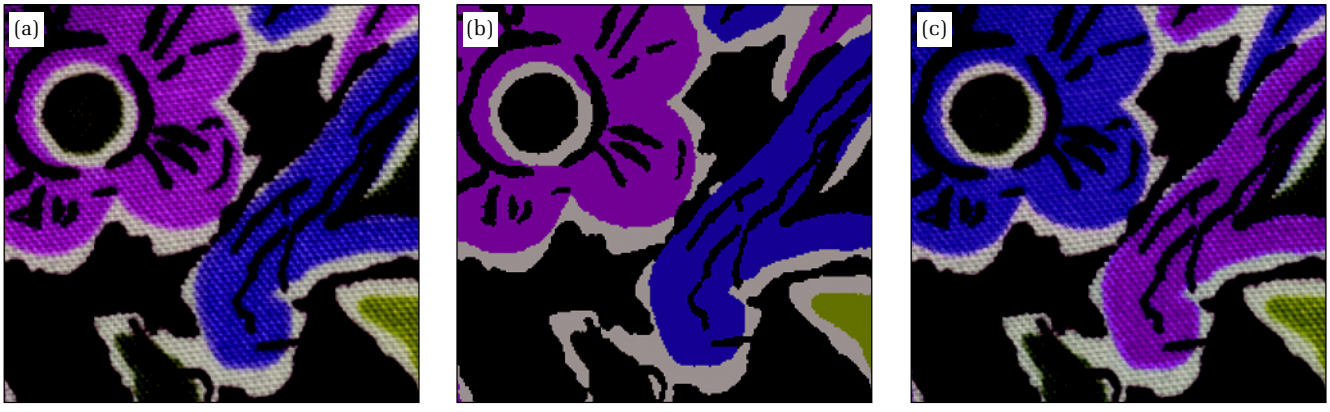


Figure 9 Second example of recolouring including segmentation and colour remapping; (a) original image; (b) label image after segmentation; (c) new simulated image

Conclusions

A method of recolouring both monochromatic and polychromatic texture images was developed in this study. It was clear that the *RGB* channels are highly correlated from the investigation of the channel correlation. According to the basic proportionality relationship between channels, the algorithm of colour remapping on texture images was developed. For polychromatic texture images, the new algorithm of WD-based segmentation was proposed to separate different regions. This method has the capability of dealing with relatively broad transitional boundaries. To overcome the problem of over-segmentation of watershed, the initial major markers are selected using the hierarchical histogram-based analysis in $L^*a^*b^*$ colour space. After WD-based segmentation, the algorithm of colour remapping was applied on each individual region of the polychromatic texture images. To achieve image fidelity, colour mixing in $L^*a^*b^*$ space by Gaussian filtering was applied to eliminate the artefacts that were introduced during the colour remapping. The overall performance of the recolouring method developed in this study was satisfactory and could be applied to the visualisation of new pattern design in the textile printing and other graphic design areas.

Acknowledgements

The authors wish to acknowledge the funding of this project by the Hong Kong Polytechnic University and the Research Grants Council of Hong Kong SAR government (project reference: PolyU 5153/01E).

References

1. J M Valiente, M C Carretero, J M Gomis and F Albert, *12th Int. Conf. on Design Tools and Methods in Indust. Eng.*, Rimini, Italy (2001) 38.
2. S Gray, *CAD/CAM in Clothing and Textiles* (Hampshire: Gower Publishing Ltd, 1998).
3. K J Dana, B Ginneken, S K Nayar and J J Koenderink, *Assoc. Comput. Machinery, Trans. Graphics*, **15** (1999) 1.
4. M Mirmehdi and M Petrou, *IEEE Trans. Pattern Anal. Machine Intell.*, **22** (2000) 142.
5. K N Plataniotis and A N Venetsanopoulos, *Color Image Processing and Applications* (New York: Springer-Verlag, 2000).
6. Y T Liow and T Pavlidis, *Proc. IEEE Int. Conf. on Robotics and Automation*, Vol. 3, Philadelphia, USA (1988) 1567.
7. T Pavlidis and Y T Liow, *IEEE Trans. Pattern Anal. Machine Intell.*, **12** (1990) 225.
8. L Vincent and P Soille, *IEEE Trans. Pattern Anal. Machine Intell.*, **13** (1991) 583.
9. P Soille, *Morphological Image Analysis: Principles and Applications* (New York: Springer-Verlag, 1999).
10. K Haris, S N Efstratiadis, N Maglaveras and A K Katsaggelos, *IEEE Trans. Image Process.*, **7** (1998) 1684.
11. P Campisi and G Scarano, *IEEE Trans. Image Process.*, **11** (2002) 37.
12. A Kokaram, *Proc. Int. Conf. Image Process.*, Vol. 1, New York, USA (2002) 325.

13. G Hong, B Han and M R Lou, *Proc. Int. Conf. Image Process.*, Vancouver, Canada (2000) 741.
14. J H Xin and H L Shen, *J. Electron. Imaging*, **12** (2003) 697.
15. Colorite, www.datacolor.com
16. R D Queiroz, in *Digital Color Imaging Handbook*, Ed. G Sharma (London: CRC Press, 2002) 559.
17. H L Shen, *Study of Middle and Low Level Algorithm in Image Understanding*, PhD Thesis, Zhejiang University, China (2002).
18. *Colour Image Processing Handbook*, Eds. S J Sangwine and R E N Horne (Boston: Kluwer, 1998).
19. W H Press, C S A Teukolsky, C W T Vetterling and B P Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd Edn (Cambridge: Cambridge University Press, 1993).
20. *Physics-Based Vision: Principle and Practice*, Eds G E Healey, S A Shafer and L B Wolff (Boston: Jones and Bartlett Publishers, 1992).
21. G Hong and R Luo, *Proc. SPIE*, Vol. 4421 (2001) 618.
22. X Zhang and B A Wandell, *SID 1996 Digest*, **27** (1996) 731.
23. M J Swain and D H Ballard, *Int. J. Comput. Vision*, **7** (1991) 11.

Appendix I Pseudo language of the WD algorithm

- Definitions:

I	Input image
L	Label image
N	Amount of markers
C	Amount of regions in input image
marker [N]	Pixels selected as markers
mean [C]	Mean value of the pixels belonging to regions
list	List structure for pixels storing
p,p'	Pixel

- For all pixels p, let $L(p) \leftarrow -1$
- Let p be all marker[N], and $L(p)$ be the corresponding labels
- Calculate mean[N] of all regions
- Find p', the neighbours of marker [N] which satisfies $L(p') = -1$, and store them into list
- While list is not empty, do
  - Pop out a pixel p from the beginning of list
  - Check all the labels of its neighbours p' that satisfy $L(p') > -1$
  - If all the labels are the same
    - Assign that label to $L(p)$
  - Otherwise
    - Calculate the distances between p and mean[i], where i is the label of each neighbour of p
    - Find the pixel with the minimal distance and its corresponding label i_0
    - $L(p) \leftarrow i_0$
  - End if
  - Update mean[$L(p)$]‡
  - For all neighbours (p') of p that satisfy $L(p') = -1$
    - Calculate the distance d between p' and mean[$L(p)$]
    - Insert p' into list according to d in ascending order
  - End for
- End do

‡ This term means that when a new pixel is assigned into a region, the mean colour of the corresponding region should be recalculated accordingly