Colour displays for categorical images

C.A. Glasbey[*]

Biomathematics and Statistics Scotland

King's Buildings, Edinburgh, EH9 3JZ, Scotland

G.W.A.M. van der Heijden

Biometris

P.O. Box 16, 6700 AA Wageningen, The Netherlands

V. Toh and A.J. Gray

Department of Statistics and Modelling Science

University of Strathclyde, Glasgow, G1 1XH, Scotland

October 26, 2006

Abstract

We propose a method for identifying a set of colours for displaying 2-D and 3-D categorical images when the categories are unordered labels. The principle is to find maximally distinct sets of colours. We either generate colours sequentially, to maximise the dissimilarity or distance between a new colour and the set of colours already chosen, or use a simulated annealing algorithm to find a set of colours of specified size. In both cases, we use a Euclidean metric on the perceptual colour space, CIE-LAB, to specify distances.

Keywords: pseudo-colour; look-up table; perceptually uniform colour space; RGB; LAB; colour quantization.

1   Introduction

Suppose we have a categorical image in 2-D or 3-D with several unordered labels. These arise in many contexts in image processing, including the labelling of connected components produced

---

[*]Corresponding author: Professor C.A. Glasbey, Biomathematics and Statistics Scotland, King's Buildings, Edinburgh, EH9 3JZ, Scotland. Email: chris@bioss.ac.uk. Tel: +(44) 131 650 4899. Fax: +(44) 131 650 4901.
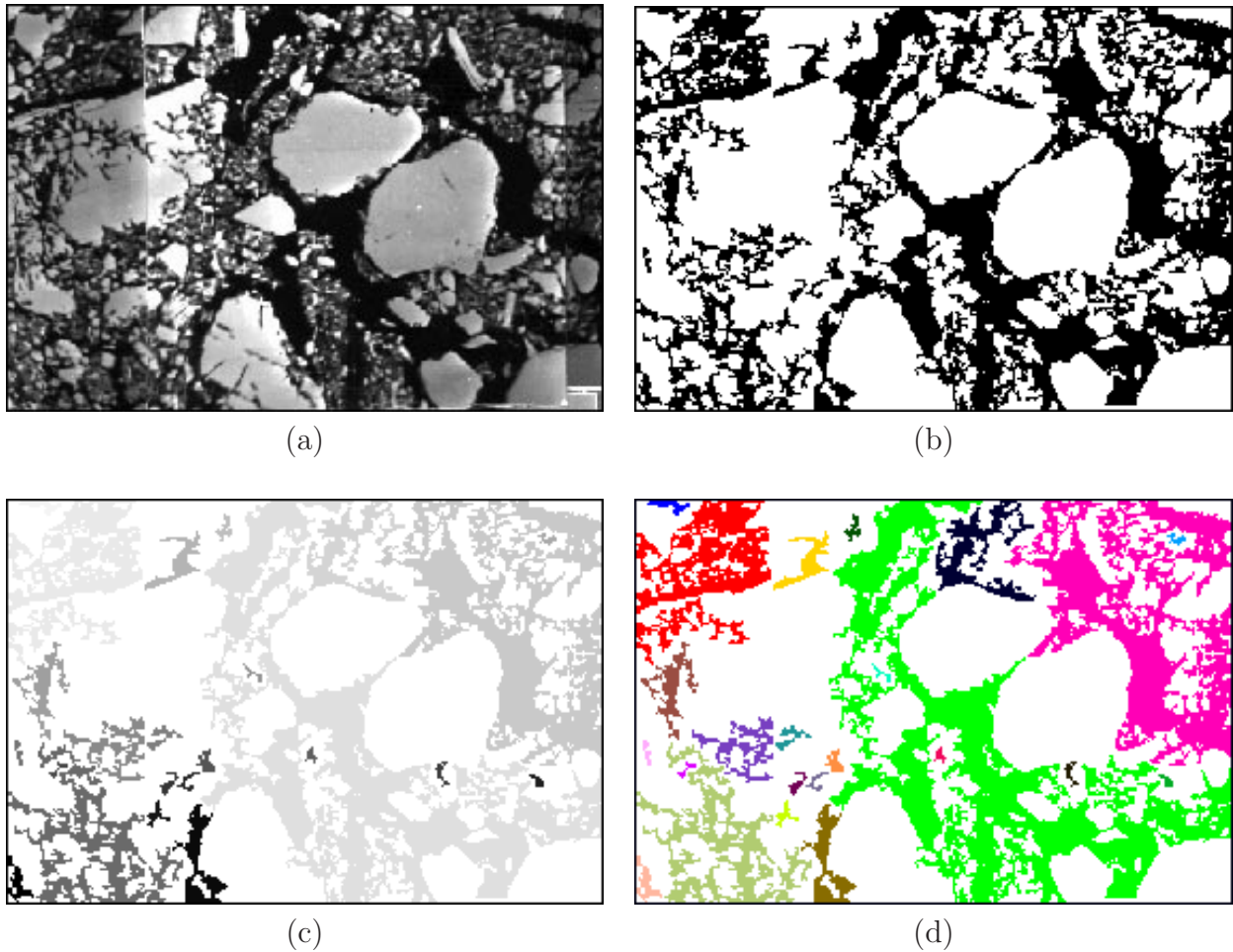
Figure 1: *Electron scanning micrographs of a soil aggregate embedded in acrylic resin:* **(a)** *original image,* **(b)** *result of segmentation with larger pores displayed as black,* **(c)** *connected pore components, each displayed as a different shade of grey,* **(d)** *connected pore components displayed using the colour labels given in Table 2.*

by segmentation algorithms. To produce a clear display of such an image we need to use a look-up table in which the categories are displayed as distinctly as possible. For example, Fig 1(a) shows an electron micrograph of a section through a soil aggregate [1]. The lighter areas are the inorganic and organic soil matrix and the black areas are soil pores. Fig 1(b) shows the result of a segmentation to identify larger pores in the section and Fig 1(c) shows a labelled image of the 24 connected pore components, using different shades of grey, although it is not straightforward to identify all the connected components. Colour vision provides a richer space in which to present variation, as can be seen in Fig 1(d), which will be discussed later. The human eye is capable of distinguishing many more colour differences than grey differences because shades of grey are merely one of at least three dimensions of colour discrimination, so pseudo-colour displays are useful for human viewing of categorical images, both in 2-D and even more in 3-D.

Our aim is to find a distinct set of colours, for use as pseudo-colours for the category levels,

or labels, in a categorical image. We propose to do this by designing an optimal colour map that maximizes the minimum dissimilarity or distance between neighbouring colours. In earlier work, Kelly [2] identified twenty two colours of maximum contrast and Boynton [3] identified a set of eleven colours that were *almost never confused*. Smallman and Boynton [4] subsequently showed that it was their separation in colour space that made them distinctive, not their having distinct names. Carter and Carter [5] proposed an algorithm for identifying high-contrast sets of colours whereas de Valk et al [6] made subjective choices. For subsequent algorithmic developments, see Campadelli et al [7] and references therein. The problem has similarities with, but is distinct from, that of colour quantization, which is concerned with selecting a set of colours, or colour palette, to display a colour image without noticeable perceived differences. For work on universal colour palettes, see, for example, Kolpatzik and Bouman [8], and on adaptive, or image-specific, colour palettes, see Gentile et al [9], Tremeau et al [10] and Ozdemir and Akarun [11].

Our paper updates Carter and Carter [5] using modern computing power and a broader consideration of colour metrics, while minimising the mathematical technicalities in more recent papers such as [7]. In §2 we formulate the method as an optimisation problem to be solved either using either a sequential search, to produce nested sets of colours of increasing size, or simulated annealing, to find a set of colours of a specified size. Then, in §3 we apply the method to modify Boynton's eleven colours, and generate sets of other sizes. Finally, in §4 we discuss extensions to the method.

2   Method

2.1   Problem Formalisation

We formalise the problem as: find a set of colours, $c_1, \ldots, c_n$ such that the minimum distance between any pair of them is made as large as possible, as did Carter and Carter [5]. Therefore

$$\{c_1, \ldots, c_n\} = \arg\max_{\{c_1,\ldots,c_n\}} \left[\min_{i \neq j} D(c_i, c_j) \right], \tag{1}$$

where $c \equiv (R, G, B)$ denotes a point in the RGB colour cube $(R, G, B = 0, \ldots, 255)$ and D is a dissimilarity or distance metric. The minimum distance is a simple criterion to use and can be justified because beyond a certain distance colours cease to increase in distinctiveness. Hence, if any D in the set is important to increase then the smallest is important to increase, and if any Ds in the set are not worth increasing, nor are the largest Ds worth increasing.

The distance measured in RGB space between two colours does not necessarily reflect the perceived dissimilarity to the human eye (see, for example, Sangwine and Horne [12]). In perceptually uniform colour spaces the distance between two colours is proportional to the perceived distance. Therefore, to compute D, we transform RGB colours displayed on a monitor to a perceptually uniform space, and use a Euclidean metric in this space. Approximations to perceptually uniform spaces include LAB, LUV, and approximations to Munsell colour space [13, 14, 15].

We have chosen to use the LAB colour space with illuminant D50 for measuring colour distances, because this is the default colour space in Adobe Photoshop and many other software packages. Since an image is normally stored and also displayed in RGB-space, we start with an RGB-image which has to be converted to LAB. The standard colour space for an RGB image for most non-calibrated monitors is the sRGB colour space, which is defined for illuminant D65 (daylight). Since the sRGB and Lab spaces use a different illuminant, it is necessary to apply chromatic adaptation correction as described by Lindbloom [16] and as implemented by Adobe Photoshop. The conversion algorithm is described in detail in Annex 1. The distance between colours c and c', with representations in LAB colour space of (L, a, b) and (L', a', b'), respectively, is given by

$$D(c, c') = \sqrt{(L - L')^2 + (a - a')^2 + (b - b')^2}. \tag{2}$$

2.2   Solutions

There are 256^3 possible choices of c in the RGB colour cube, so an exhaustive search to solve (1) for n colours would involve $[256^3 \times (256^3 - 1) \times \ldots \times (256^3 - n + 1)]/[2 \times 3 \times \ldots \times n]$ possible solutions, and so rapidly becomes computationally infeasible as n increases. We therefore adopted two alternative approaches: a sequential search algorithm and simulated annealing.

For the sequential method, given an initial colour, in order to create a set with two colours, we add to it the colour which is most distant from this one, as measured by D (2). Then to create a set with three colours, we add a third colour which is most distant from the existing two. In general, given a set of n colours, we choose as the $(n+1)$st colour the one which is most distant from the existing set. The advantages of this approach are that it is comparatively fast, as we only consider 256^3 possibilities at each iteration, and the sets of colours are nested for different value of n, which simplifies their use. We have used white, i.e. $c_1 = (255, 255, 255)$, as our initial colour, though alternatives are possible. In increasing the set from n to $(n+1)$ colours, we find the new colour that maximises the minimum distance between it and all existing colours. Therefore

$$c_{n+1} = \arg \max_c \left[\min_{i=1,\ldots,n} D(c_i, c) \right], \tag{3}$$

which can be solved by exhaustive search.

A disadvantage of the sequential method is that the results are not guaranteed to be optimal: another set of n colours may have a larger minimum distance and solve (1). Because iterative optimization methods can become trapped in local optima, we have used a stochastic method to reduce this risk, using the well established method of simulated annealing [17]. To solve (1) using a simulated annealing algorithm, we start with an initial set of n random colours. At each iteration we consider replacing one of the two colours which currently minimizes D, c_{OLD}, with a new, randomly chosen colour, c_{NEW}. We accept this replacement with probability

$$p = \min \left\{ 1 \ , \ \exp \left[\frac{D_{NEW} - D_{OLD}}{T} \right] \right\},$$

where, D_{OLD} and D_{NEW} denote the current and new minimum distances, and T (> 0) is a 'temperature' setting that decreases slowly as the iterations proceed. So, if $D_{NEW} \geq D_{OLD}$,

n	n'	$D(c_n, c_{n'})$	$c_n = (R,$	$G,$	$B)$	$= (L,$	$a,$	$b)$	description
1			255	255	255	100	0	0	white
2	1	149	0	0	255	30	68	−112	blue
3	1	116	255	0	0	54	81	70	red
4	1	114	0	255	0	88	−79	81	green
5	1	100	0	0	0	0	0	0	black
6	4	65	255	255	0	98	−16	93	yellow
7	2	65	255	0	255	60	94	−61	magenta
8	3	58	255	128	128	69	49	24	pink
9	1	46	128	128	128	54	0	0	grey
10	8	45	128	0	0	26	48	39	brown
11	3	38	255	128	0	68	45	75	orange

Table 1: *Boynton's colours, in optimised order using criterion (3): n' denotes the index to the nearest colour to c_n in the existing set.*

$p = 1$ and c_{NEW} will be chosen. Otherwise, c_{NEW} is chosen with a probability $p < 1$, which guards us against becoming trapped in a local optimum of (1). Further details are given in Annex 2.

3 Results

To explore the sequential search approach, we initially used it to order the eleven non-confusing colours of Boynton [3]. We do not use his precise set of colours in Ljg-space, but for simplicity we define the eleven colours as the 8 colours at the corners of the RGB colour cube, the grey in the middle of the cube and 2 mid-points on the edges of the cube for orange and pink. So, we solve (3), but restricting c to this set of eleven colours. The result is shown in Table 1, together with the minimum value of D between any pair of colours in each of the sets and the nearest colour ($c_{n'}$) to the new colour. We see that the minimum value of D decreases as the set size, n, increases, and that the colours at the corners of the RGB colour cube are chosen first. By comparing adjacent colours n and n', it is apparent that all eleven colours can be distinguished from their nearest neighbours. So, if a set of $n \leq 11$ were required for an application, we would recommend using the first n colours in Table 1.

However, these are not necessary the best possible sets of $n < 11$ colours, nor do they offer a solution for displaying categorical images with more than eleven categories, such as Figure 1(c). So, we also found colour sets using (3), but searching over the whole RGB colour cube. Table 2 shows the first 32 colours we obtained with the sequential algorithm. The first 5 colours are very similar to those in Table 1, but thereafter the lists diverge. We also see that minimum values of D are higher in Table 2, so these colours are more distinct than those in Table 1. Even for eleven colours, D is higher at 67 than the value of 38 for Boynton's colours. As n increases it becomes increasingly difficult to distinguish between some pairs c_n and $c_{n'}$, particularly for $n = 16, 19, 32$.

Fig 1(d) illustrates the results of using this look-up table on the soil image, with $n = 24$. The effectiveness of using this set of colours is clearly demonstrated. It is far easier to distinguish between connected pores than with the shades of grey shown in Figure 1(c).

Finally, for illustration, we used the simulated annealing algorithm to choose a set of eleven colours for comparison with Boynton's colours. We found many sets of colours with a minimised value of $D = 74$, which is a slight improvement on $D = 67$ for the first eleven colours in Table 2. Note, that in this table the colours are in no particular order.

4   Discussion

We have identified several sets of colours for displaying categorical images on a colour monitor when the categories are unordered labels, either limiting the choice to eleven colours (related to Boynton's colours that are almost never confused), or considering all possible colours in the RGB colour cube. We restricted attention to a computer monitor as the display device (sRGB(D65)), and the transformation to an approximately perceptually uniform space as used in Photoshop, LAB(D50). The two proposed algorithms are simple to use. In particular, the sequential approach is appealing as it is hardly worse than the simulated annealing approach, is much faster, and keeps a stable set of colours when increasing set size.

Our approach is open to several extensions and generalisations. For other display devices, such as colour printers and data projectors, other transformations may be needed [18, 16]. Since the gamut for printers is generally considerable smaller than that for monitors, it might be better to restrict the search to colours which are not outside the printable gamut. Out-of-gamut colours for a printer can easily be found using Adobe Photoshop. By using an image containing all 16 millions colours, and the option colour-range, we can select all out-of-gamut colours for a certain printing profile, e.g. Euroscale coated paper. By setting these to zero, we can later use this image for sampling from the restricted set of RGB-colours. One could also consider other colour spaces, using ones own specified ICC-profiles for reproduction. Also, there is the option to use a colour other than white as the first colour (c_1) in method (3).

One drawback of using (1) as a criterion is that it ignores the distances between all pairs of colours that are greater than the minimum distance, even though changing the colours so that these distances are increased may appear to improve the set. It is possible to apply our methods with other criteria, which take account of all distances, such as maximising the sum of distances

$$\max \sum_{i \neq j} D(c_i, c_j).$$

However, with this criterion, there may be particular pairs of colours with very small distances between them.

Finally, it would be possible to take account of forms of visual impairment, such as dichromacy colour blindness. In this case, we could first map the RGB-value to the LMS-colour space, then project it on the colourblind subspace as described by Brettel et al [19], convert it back to RGB and start with those values. Since the subspace is considerably smaller, it might be worthwhile

n	n'	$D(c_n, c_{n'})$	$c_n = (R,$	$G,$	$B)$	$= (L,$	$a,$	$b)$
1			255	255	255	100	0	0
2	1	149	0	0	255	30	68	−112
3	1	116	255	0	0	54	81	70
4	1	114	0	255	0	88	−79	81
5	2	103	0	0	51	2	11	−30
6	3	94	255	0	182	57	87	−24
7	5	86	0	83	0	30	−35	36
8	3	84	255	211	0	86	5	86
9	5	70	0	159	255	62	−8	−58
10	3	69	154	77	66	43	32	22
11	1	67	0	255	190	89	−64	16
12	9	58	120	63	193	40	45	−60
13	11	53	31	150	152	56	−32	−11
14	12	52	255	172	253	81	40	−28
15	8	51	177	204	113	79	−20	42
16	3	47	241	8	92	52	79	23
17	10	47	254	143	66	71	39	58
18	12	46	221	0	255	54	88	−70
19	5	46	32	26	1	10	0	14
20	10	46	114	0	85	25	49	−17
21	9	44	118	108	149	48	11	−21
22	4	43	2	173	36	62	−58	54
23	8	43	200	255	0	94	−37	88
24	15	42	136	108	0	47	5	53
25	1	39	255	183	159	81	25	23
26	15	38	133	133	103	55	−4	16
27	3	38	161	3	0	34	57	49
28	11	37	20	249	255	89	−48	−17
29	21	37	0	71	158	31	10	−54
30	14	36	220	94	147	58	54	−3
31	1	36	147	212	255	82	−13	−28
32	2	35	0	76	255	40	40	−95

Table 2: *The first 32 colours found by sequential search to solve (3): n' denotes the index to the nearest colour to c_n in the existing set.*

n	$c_n = (R,$	$G,$	$B)$	$= (L,$	$a,$	$b)$	description
1	91	0	13	17	38	22	brown
2	0	255	223	90	−57	0	cyan
3	23	169	255	66	−13	−53	light blue
4	255	232	0	92	−5	89	yellow
5	8	0	91	6	32	−52	dark blue
6	255	208	198	87	16	12	light pink
7	4	255	4	88	−79	81	light green
8	0	0	255	30	68	−112	blue
9	0	79	0	28	−34	35	dark green
10	255	21	205	58	88	−35	dark pink
11	255	0	0	54	81	70	red

Table 3: *The optimal set of eleven colours, found by solving (1) using simulated annealing.*

to construct a lookup table for the colours, so the conversion has to be done only once. Colour perception can also depend on region size, such as in small-field tritanopia [20], which would require optimising the allocation of colours to regions.

Acknowledgements

References

[1] C. A. Glasbey and G. W. Horgan. *Image Analysis for the Biological Sciences.* Wiley, Chichester, 1995.

[2] K. L. Kelly. Twenty two colors of maximum contrast. *Color Engineering*, 3:26–27, 1965.

[3] R. M. Boynton. Eleven colors that are almost never confused. In B. E. Rogowitz, editor, *Proceedings of the SPIE Symposium: Human Vision, Visual Processing, and Digital Display*, volume 1077, pages 322–332, Bellingham, WA, 1989. SPIE Int. Soc. Optical Engineering.

[4] H. S Smallman and R. M. Boynton. Segregation of basic colors in an information display. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 7:1985–1994, 1990.

[5] R. C. Carter and E. C. Carter. High-contrast sets of colors. *Applied Optics*, 21:2936–2939, 1982.

[6] J. P. J. de Valk, W. J. M. Epping, and A. Heringa. Colour representation of biomedical data. *Medical & Biological Engineering & Computing*, 23:343–351, 1985.

[7] P. Campadelli, R. Posenato, and R. Schettini. An algorithm for the selection of high-contrast color sets. *Color Research and Application*, 24:132–138, 1999.

[8] B. W. Kolpatzik and C. A. Bouman. Optimized universal color palette design for error diffusion. *Journal of Electronic Imaging*, 4:131–143, 1995.

[9] R. S. Gentile, J. P. Allebach, and E. Walawit. Quantization of color images based on uniform color spaces. *Journal of Imaging Technology*, 16:11–21, 1990.

[10] A. Tremeau, M. Calonnier, and B. Laget. Color quantization error in terms of perceived image quality. In *ICASSP-94 - Proceedings, Vol 5 - I - Image and Multidimensional Signal Processing*, pages 93–96, New York, 1994. IEEE.

[11] D. Ozdemir and L. Akarun. A fuzzy algorithm for color quantization of images. *Pattern Recognition*, 35:1785–1791, 2002.

[12] S. J. Sangwine and R. E. N. Horne, editors. *The Colour Image Processing Handbook*. Chapman and Hall, London, 1998.

[13] G. Wyszecki and W. S. Stiles. *Colour Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, New York, second edition, 1982.

[14] M. Miyahara and Y. Yoshida. Mathematical transform of (R,G,B) color data to Munsell (H,V,C) color data. In T. R. Hsing, editor, *Proceedings Visual Communications and Image Processing Conference*, volume 1001, pages 650–657, Cambridge, MA, 1988. Society of Photo-Optical Instrumentation Engineers.

[15] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall International, New Jersey, 1989.

[16] B. Lindbloom. Web page of colour space information. (http://www.brucelindbloom.com/), 2003.

[17] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–735, 1984.

[18] G. Sharma and H. J. Trussell. Digital color imaging. *IEEE Transactions on Image Processing*, 6:901–932, 1997.

[19] H. Brettel, F. Vienot, and J. D. Mollon. Computerized simulation of color appearance for dichromats. *Journal of the Optical Society of America, A*, 14:2647–2655, 1997.

[20] S. M. Highnote, G. Flint, and D. I. A. MacLeod. Color discrimination of small fields on self-luminous displays. *Society for Information Display*, 96:727–730, 1996.

[21] W. H. Press, editor. *Numerical Recipes in Fortran: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 2nd edition, 1994.

Annex 1

The sRGB(D65) and Lab(D50) are the default working spaces used by Adobe Photoshop. Therefore, we have restricted our algorithm to these colour spaces. The Lab colour is a perceptual uniform colour space and conversion from sRGB involves some non-linear transformations. The Lab values can be calculated as

$$
\begin{aligned}
L &= 116 f(Y/Y_0) - 16, \\
a &= 500[f(X/X_0) - f(Y/Y_0)], \\
b &= 200[f(Y/Y_0) - f(Z/Z_0)],
\end{aligned}
$$

where

$$
\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 1.047835, & 0.022897, & -0.050147 \\ 0.029556, & 0.990481, & -0.017056 \\ -0.009238, & 0.015050, & 0.752034 \end{pmatrix} \begin{pmatrix} 0.412424, & 0.357579, & 0.180464 \\ 0.212656, & 0.715158, & 0.072186 \\ 0.019332, & 0.119193, & 0.950444 \end{pmatrix} \begin{pmatrix} g(R/255) \\ g(G/255) \\ g(B/255) \end{pmatrix},
$$

using matrix notation. Here, the first matrix accounts for the Bradford chromatic adaptation, and has to be applied to solve for the differences in white point (due to differences in illuminants) between the two colour spaces. The second matrix is the standard matrix to convert sRGB values to XYZ. $(X_0, Y_0, Z_0)^T = (0.964221, \ 1, \ 0.825213)^T$ is the white point of D50 illuminant. Furthermore,

$$
f(z) = \begin{cases} z^{\frac{1}{3}} & \text{if } z > 0.008856 \\ 7.787z + \frac{16}{116} & \text{otherwise,} \end{cases}
$$

and

$$
g(z) = \begin{cases} \left(\frac{z+0.055}{1.055}\right)^{2.4} & \text{if } z > 0.04045 \\ \frac{z}{12.92} & \text{otherwise,} \end{cases}
$$

which represents the gamma correction of the image display.

For further details on conversions of colour spaces, see Lindbloom [16].

Annex 2

The simulated annealing algorithm is based on methods described in *Numerical Recipes* [21]. We start with an initial set of n random colours. At each of 100 iterations we consider a maximum of 25600 replacements of the two colours which currently minimizes D, c_{OLD}, with a new colour, c_{NEW}. Two rules are available for selecting c_{NEW}:

1. c_{NEW} is a random point from the 256^3 space;

2. c_{NEW} is a point close to c_{OLD}, selected at random within a $5 \times 5 \times 5$ cube in the RGB-space, centred on c_{OLD}.

Early in the search process, preference is given to rule 1 ($P(\text{rule } 1) = 0.99$) and this probability is linearly decreased at each iteration by 0.01, until iteration 100.

We accept a replacement with probability

$$p = \min \left\{ 1 \ , \ \exp \left[\frac{D_{NEW} - D_{OLD}}{T} \right] \right\},$$

where D_{OLD} and D_{NEW} denote the current and new minimum distances, and T (> 0) is a 'temperature' setting that decreases slowly as the iterations proceed. So, if $D_{NEW} \geq D_{OLD}$, $p = 1$ and c_{NEW} will be chosen. Otherwise, c_{NEW} is chosen with a probability $p < 1$, which guards us against becoming trapped in a local optimum of (1). The initial temperature T is set at 10 and an annealing factor of 0.9 is used at each iteration, so $T_{i+1} = T_i \times 0.9$.

We terminate an iteration and decrease the temperature when either the maximum of 25600 replacements have been considered, or more than 10% of the replacements considered have been accepted. The algorithm terminates when the maximum number of 100 iterations is reached, or no replacements are accepted during an iteration.