Colour image compression by grey to colour conversion

Mark S. Drew[a], Graham D. Finlayson[b] and Abhilash Jindal[c]

[a]School of Computing Science, Simon Fraser University, Burnaby, BC, Canada V5A 1S6
[b]School of Computing Sciences, University of East Angelia
[c]Electrical Engineering Department, Indian Institute of Technology, Kanpur, UP, India 208016

ABSTRACT

Instead of de-correlating image luminance from chrominance, some use has been made of using the correlation between the luminance component of an image and its chromatic components, or the correlation between colour components, for colour image compression. In one approach, the Green colour channel was taken as a base, and the other colour channels or their DCT subbands were approximated as polynomial functions of the base inside image windows.

This paper points out that we can do better if we introduce an addressing scheme into the image description such that similar colours are grouped together spatially. With a Luminance component base, we test several colour spaces and rearrangement schemes, including segmentation, and settle on a log-geometric-mean colour space. Along with PSNR versus bits-per-pixel, we found that spatially-keyed s-CIELAB colour error better identifies problem regions. Instead of segmentation, we found that rearranging on sorted chromatic components has almost equal performance and better compression. Here, we sort on each of the chromatic components and separately encode windows of each.

The result consists of the original greyscale plane plus the polynomial coefficients of windows of rearranged chromatic values, which are then quantized. The simplicity of the method produces a fast and simple scheme for colour image and video compression, with excellent results.

Keywords: Image Compression, Colour, Grey-to-colour

1. INTRODUCTION

It has been recognized that some correlation exists between the luminance component of an image and its chromatic components, or the correlation of one chromatic component with the second, with the aim being more efficient encoding of the chrominance[1] — after all, colour changes are usually associated with material changes, rather than illumination changes (cf. Ref.[2]). This approach is in contrast to the standard approach of *de-correlating* luminance from colour via a colour space transform, as in JPEG or JPEG2000, wherein the de-correlated colour components are then encoded separately. However, typically luminance-chrominance correlation has been addressed in the context of quite complex algorithms such as in a discrete wavelet transform.[3] As well, it is possible to more thoroughly and adaptively exploit patterns in image colour by utilizing independent components.[4]

A simple approach was set out in Ref.[5], where one colour channel (Green) is taken as a base, and the other colour channels (Red and Blue) are approximated as polynomial functions of the base inside each image window. This simple approach was modified in Ref.[1] by expanding each DCT subband instead of each image block as a whole.

In a novel approach, this paper goes on to point out that we can in fact do better if we introduce an addressing scheme into the image description such that similar colours are grouped together spatially. In this way, image windows contain pixel values that are better mapped by a simple parametric expansion. We utilize polynomial regression based on the Luminance component, and test several candidate colour spaces and rearrangement schemes. We find that the "gold standard" scheme involves segmentation of the chromatic part of the image: if we first group chromatic components such that they now occur in the rearranged image in larger blocks that are all similar, then clearly a parametric regression will better approximate the original image, and we found that such an approach indeed produces a substantive improvement. One reason for this is

Further author information: (Send correspondence to Mark S. Drew)
Mark S. Drew: E-mail: mark@cs.sfu.ca, Telephone: 1 778 782 4277
Graham D. Finlayson: E-mail: graham@cmp.uea.ac.uk, Telephone: +44 1603 593347
Abhilash Jindal: E-mail: ajindal@iitk.ac.in, Telephone: +91 9889914733

that e.g. in a Lambertian model of colour image formation, many colour space transforms will have the effect of removing shading from chromatic components — e.g., going over to Luminance plus standard L_1-norm Chromaticity[6] means that whereas the achromatic channel contains both intensity and shading information, the chromaticity has no shading and hence appears as a collection of almost uniformly coloured regions, which are of course much easier to segment. So the question of which colour space to use is important, and here we found the best performance by making use of a log-geometric-mean colour space. We found that instead of simply measuring image reconstruction goodness in terms of PSNR, it was beneficial to use as well as the spatially-keyed s-CIELAB colour error metric, which tends to identify problem areas rather more than does the per-pixel error metric. However, instead of segmentation we found that a much less computationally complex scheme consisting of simply rearranging on sorted chromatic components has almost equal performance. Here, we simply sort on each of the chromatic components and separately encode windows of each. The result consists of the original greyscale plane plus the polynomial coefficients of windows of rearranged chromatic values, which are then quantized. The simplicity of the method produces a fast and simple scheme for colour image and video compression, with excellent results.

2. COLOUR MODELS

Each of the pixels in any image can be decomposed into one Luminance component and two Chrominance components. The colour information in the image is represented by chrominance and the illumination intensity and shading in the image, i.e. the greyscale value, is represented by luminance. The basic idea of this paper is to break the input image into parts and then plot the chrominance components against luminance for each part. For each individual part, we carry out polynomial regression to obtain regression coefficients. Now, this is certainly a lossy technique for compression as the polynomial regression cannot exactly capture the complete variation of chrominance with luminance. But, since human vision is not able to differentiate between slight differences in colours, this approach can nevertheless be used to provide an adequate reconstructed colour image. After obtaining the coefficients, they are quantized before being transmitted or stored. Also, the greyscale image is compressed using the standard JPEG technique before transmission. Essentially, we wish to send the greyscale image of the original colour image plus the coefficients obtained by polynomial regression in a compressed form which can later be used to extract colour from the greyscale image.

Note that there are a variety of colour models possible to use, with each of them having both positive and negative properties. We tested several different colour spaces for our colour-compression approach and compare results for these below in the Results section.

2.1 Log-Geometric-Mean Luminance

The Weber-Fechner Law (see, e.g., Ref.[7]) is a traditional approach towards encapsulating the relationship between human perception and physical stimuli. It states that, in a number of circumstances, human perception is proportional to the logarithm of physical stimuli:

$$p = k \log \frac{S}{S_0} \tag{1}$$

where p in eq. (1) stands for perception, and S denotes stimulus. Working with logs means that we are more concerned with relative error rather than absolute error.

To make use of this approach, the RGB space is first converted into $\log R$, $\log G$, $\log B$. Luminance is simply calculating by taking the average of $\log R$,$\log G$,$\log B$. The method for calculating chrominance from RGB space can be found in Appendix A.1.

2.2 AC_1C_2 Space

AC_1C_2 is an opponent colour space due to Wandell[8] . The opponent colour theory states that the human visual system sees colours using signals from three different types of cones (as well as achromatic rods at low light levels). The different types of cones have substantial overlaps in the wavelengths of the light to which they respond. It is thus more efficient to work with differences between these cones rather than their individual responses. This theory suggests that the three opponent channels are red versus green, blue versus yellow, and an achromatic grey channel. The achromatic channel detects illuminance variation, or luminance. For completeness, obtaining AC_1C_2 space from RGB space is described in Appendix A.2.

2.3 YIQ Space

YIQ space is specified in the NTSC colour TV standard and is widely used. I stands for the in-phase component and Q stands for quadrature component. Y represents the 'luma' information whereas I and Q hold the chromaticity components. Conversion of RGB space into YIQ space is shown in Appendix A.3. (and see, e.g., Ref.[7]).

The YIQ system is intended to take advantage of human colour-response characteristics. The eye has been found to be more sensitive to changes in the orange-blue (I) range than in the purple-green range (Q) hence lesser bandwidth is required for Q than for I. Hence, we can use this to our advantage, by using higher order polynomials for the fitting of I channel and lower order polynomials for the fitting of Q channel.

2.4 YC_bC_r Space

YC_bC_r space is mainly used in digital video. Again, Y denotes the luma component, and C_b and C_r are blue-difference and red-difference chroma components respectively. (see, Ref.[7]).

2.5 Mean Luminance

In this method, we directly compute the mean of R, G, B channels to obtain the luminance. This colour space has widely been used. The method for computing the chrominance is very much similar to the Log-Geometric-Mean Luminance (the only difference is that the log-geometric-mean method operates in $\log$ space). The same technique when applied in the RGB space gives the mean luminance and its chrominances.

2.6 HSV Space

HSV (Hue, Saturation and Value) space has been widely used in computer graphics, colour modification, etc. The V value can be used as a luminance component and the hue and saturation can be used as chrominance components.

Our results for all the colour spaces are compared below in the Results section, in Fig. 5.

3. ALGORITHM

The colour image is initially divided into several parts and polynomial regression is used in each individual part to fit the variations of chrominance components against the luminance. The coefficients obtained after regression are further compressed and then transmitted or stored.

We adopted mainly four techniques to divide the image:

3.1 Windowing Method

The input RGB image is first converted into the appropriate colour space being tested, to obtain luminance and chrominance components. Then, the image is divided into non-overlapping tiles of fixed size. Each chrominance component is fitted against luminance components separately for each tile, using Tikhonov Regularization (our approach to the use of this method is explained in Appendix B). The window size and the degree of polynomial have to be controlled in this method. An example output and input image is displayed in Fig. 1, for 21×21 pixel windows and a second-order polynomial. The fitting performance for the first box in Fig. 1 is shown in Fig. 2 and is seen to be excellent.

We compared the use of the several different colour spaces described above, for this fixed-window approach. As is later shown in the results section, the log-geometric-mean space was found to be the best overall for this Windowing method.

This method is quite inexpensive in terms of computational complexity. However, the method shows significant errors in windows where there is an edge for two completely different colours. This error can be seen by visual inspection in Fig. 1. As well, the poor fitting performance for this window can be seen in Fig. 3.

To address this major problem, we consider the next method, based on zeroing in on problem areas.

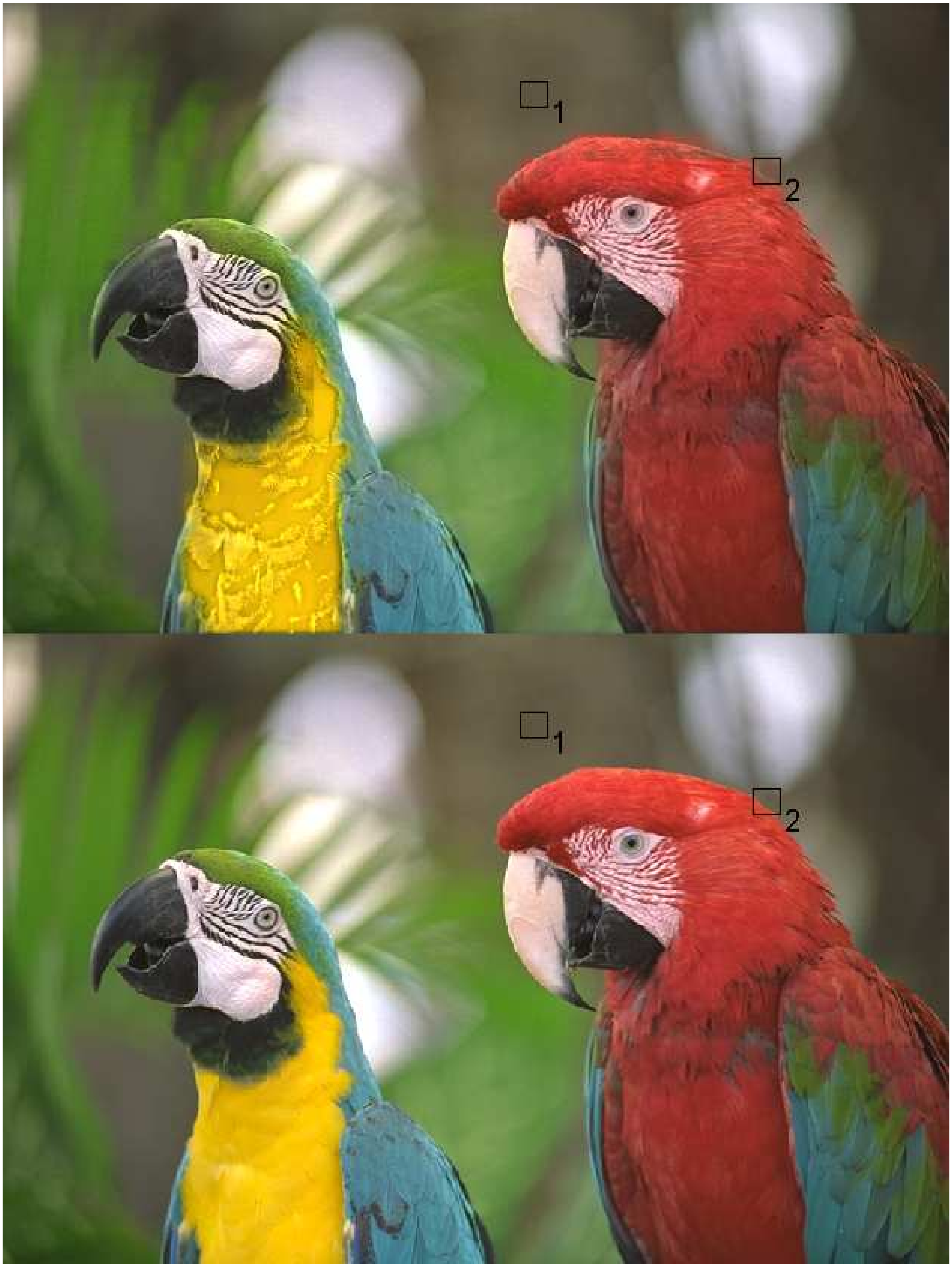


Figure 1. The bottom image shows the original image and the top image shows the compressed image using the windowing method. The window size used is 21×21 pixels here. In `box 1`, where the colour is more or less uniform, it can be seen that the colour in the compressed image is also correct. But, in `box 2`, it can be seen that, due to the presence of the edge, the colour of the compressed image is wrong.

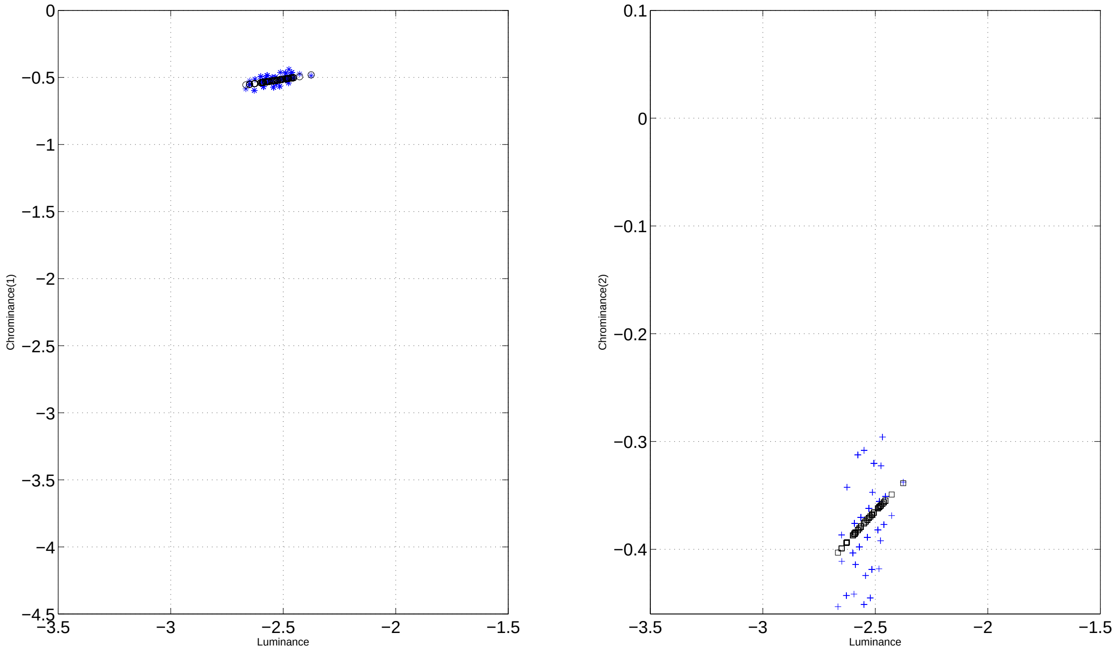


Figure 2. Regression Results for Box 1 using the log-geometric-mean colour space and tiles of size 21×21. Since the box does not contain any edge, colours are very uniform and hence the performance of polynomial regression is quite good — the maximum error is only of order 10^{-2} in a scale of $0..1$.

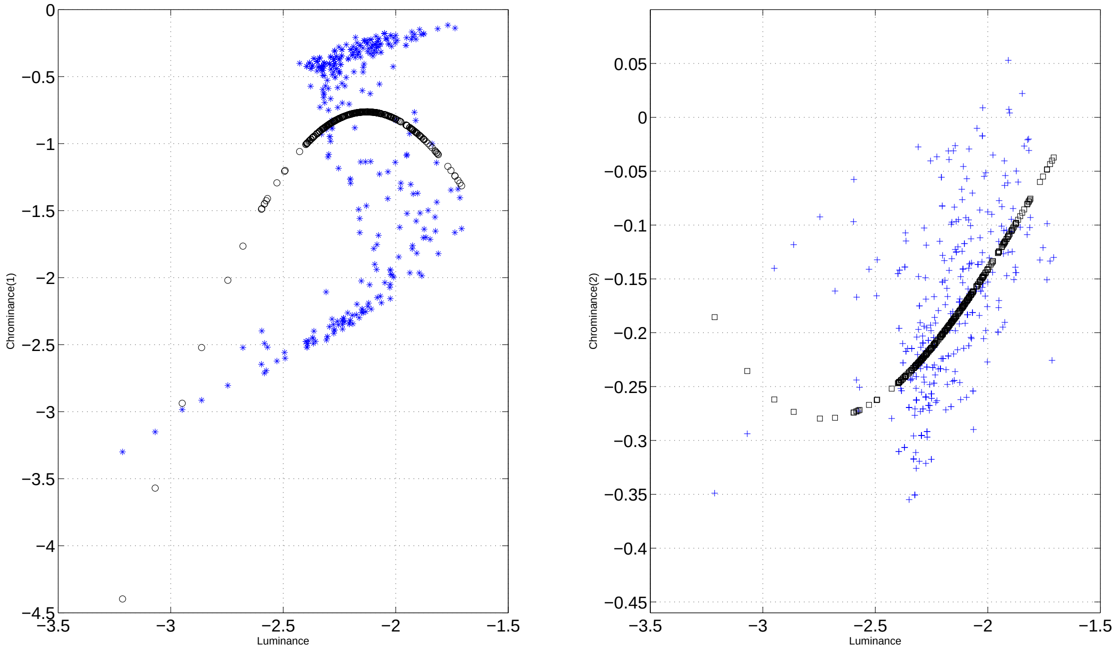


Figure 3. Regression Results for Box 2. Due to the edge in the original image, the chrominance plot is very scattered. Hence, the regression performance is not good here.

3.2 Pyramid Method

In this approach, we start by plotting the chromaticity vs. luminance for the *whole image*, and carry out polynomial regression. If the error is greater than a pre-defined threshold then the image is split into four parts. Then, we repeat this process recursively over all the four parts. E.g., we first do polynomial regression on the first part. If the error in that part is less than a threshold, we continue on to the next part. But if the error is greater than the threshold, then we again break the part into four. We continue splitting the windows until a minimum allowable size of windows (7 pixels wide) is reached.

Clearly, the performance of overall compression can be easily controlled by changing the performance threshold. Also, if an edge is present in a window, the error will be much higher. Hence, it is broken into parts until the edge is eliminated from the windows or the minimum allowable size of windows is reached.

But, this method is very computationally expensive. Unlike the windowing method, the windows can be of arbitrary sizes. Moreover, at the decompression stage, it is necessary to know how the images is broken into parts. Hence this involves an extra overhead, for sending the quantized coefficients.

3.3 Segmentation Method

As we saw in windowing method, a major problem arises in the case of sudden change of chrominance inside a window. So, it is best to form the windows according to spatial similarity in chrominance components. Hence, under this approach, we segment the image using just the chrominance components, here using the Mean-Shift Segmentation technique.[9] After performing the segmentation, we bring the pixels belonging to one segment together and plot their chrominance with respect to their luminance. For each individual segment, we again use polynomial regression to obtain the coefficients.

The Mean-Shift segmentation parameters (h_c and h_s) and the degree of polynomial have to be adjusted in this method. Since this method involves segmentation, then of course this method is considerably more computationally expensive than the windowing method. Also, to again arrange the pixels into their segments at the decoder side, we will have to send the segment label for each of the pixels. Nevertheless, in this method, since the similar pixels are grouped together, better fitting is expected.

3.4 Sorting chrominance channels separately

Considering the complexity problem concomitant with using segmentation, our next goal is to arrange the pixels into similar chromaticity areas in a smaller amount of time. So, under this approach, we propose bringing together pixels having similar chromaticity simply by using sorting. We build two different sequences of pixels by separately sorting the two chromaticity components. These are shown for our exemplar image in Fig. 4. We then take the rearranged sequences one by one and break them into chunks of fixed sizes. Then the chrominance is plotted against the luminance for each chunk and polynomial regression is applied. Since the chrominance was sorted beforehand, a considerably better fitting can be expected in each chunk. The coefficients are then quantized before they are sent. Now, again the pixels have to be rearranged before one can perform decompression. Therefore, the positions of each pixel in the rearranged sequences are sent along with the quantized coefficients.

But since there is no segmentation involved and sorting is, by far, faster than segmentation, this method is computationally much faster than the segmentation method proposed earlier (although perhaps suboptimal with respect to grouping similar chrominance values). Also, since the pixels were sorted initially, its performance should certainly be better than the original windowing method. And as we will learn later from the results, its performance is in fact much better.

4. PERFORMANCE MEASURE FOR DIFFERENT TECHNIQUES

Here all the compression techniques are evaluated in terms of rate distortion functions, $R(D)$, i.e., the minimum bitrate required to achieve distortion D. Compression techniques essentially try to lower the rate given distortion D. Or in other words, the lower the rate distortion function value, the better the compression technique.

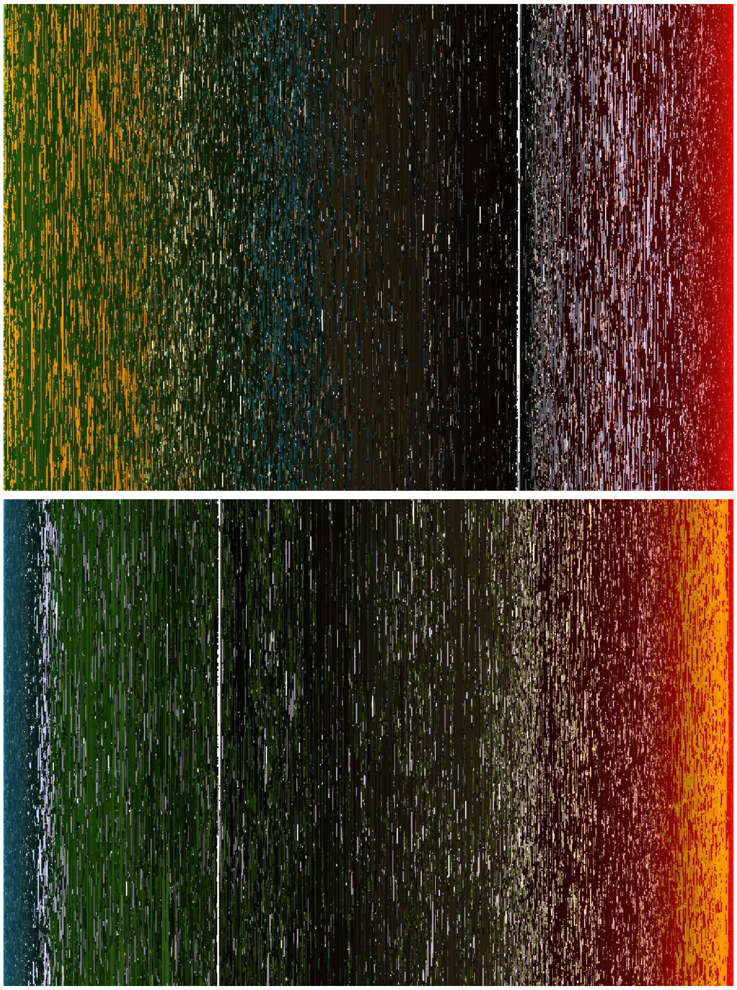


Figure 4. Sorted chrominance components (in log-geometric-mean colour space). Here, the original RGB image is shown with addressing according to sorting respectively the first and second chrominance channels.

4.1 Distortion (Comparision metric between two images)

We make use of two different but complementary error metrics:

1. **Peak Signal to Noise Ratio (PSNR)** In case of lossy compression, for comparing two images the Peak Signal to Noise Ratio has been widely used.[10]

$$PSNR = 10 \log_{10} \frac{G^2}{MSE} \tag{2}$$

$$MSE = \frac{\sum_i^N \sum_j^M \sum_{k \epsilon RGB} (p_k(i,j) - o_k(i,j))^2}{3MN} \tag{3}$$

   where G is the maximum signal value, $o_k(i,j)$ is the original image pixel value in colour channel k, and $p_k(i,j)$ is the compressed version.

   But, as is evident from eq. (2), PSNR does not capture any spatial variation between the two images being compared, and hence can show some contradicting results.

2. **S-CIELAB error** To capture the spatial variation in the images, we have used Spatial-CIELAB[8] error (ΔE_s) as a comparison metric between the images. This error was specially designed to simulate human vision. In this method, a spatial blurring is applied to the colour image data to simulate the human visual system. Also, the blurring function is adjusted differently for three different colour planes according to human psychophysical measurements of colour appearance. More details on this topic can be found in Appendix C.

4.2 Compression Rate

The coefficients obtained from polynomial regression have to be quantized before sending or storing. We quantized the coefficients linearly over their maximum and minimum limit. After we obtain these discrete numbers we can carry out any kind of source coding: e.g. Huffman Coding, LZW, etc. Here, we indicate compression rate by simply stating the lower bound, given by the Shannon entropy:

$$H(X) = \sum_x p(x) \log_2 p(x) \tag{4}$$

where $p(x)$ is the probability distribution of x. Hence, we use the entropy given by (4), not including overheads and the bitrate for the grey plane, as the rate parameter in our case.

5. RESULTS

First of all, we need to decide on the colour space in which we wish to work. In Fig. 5, the performance for the windowing method is compared for the different colour spaces in §2. We consider the behaviour of both the PSNR and the S-CIELAB error, as we change bits-per-pixel. It can clearly be seen that the performance of the log-geometric-mean colour method is the best.

Next, in Fig. 6 we compare the performance of the different algorithms that are described above in §3, now fixing on using the log-geometric-mean colour space. Clearly, the performance for the idea of separately sorting each chrominance component is much better than all the other methods. Fig. 7 shows visually how the different algorithms behave, for one particular image

6. CONCLUSION

Notwithstanding the excellence of a segmentation approach in terms of an accurate nonlinear regression performance, the separate-sorting algorithm greatly outperforms all other methods when bitrate is taken into account, (although overheads, not included in the entropy, may alter this conclusion). The other main result determined in the paper is that a log-geometric-mean luminance component, with chrominance given by projection of log values into the plane orthogonal to the luminance vector, is found to be the best colour space.

In future work, we plan to extend the methods used here into the video domain, taking advantage of the insights gained here.

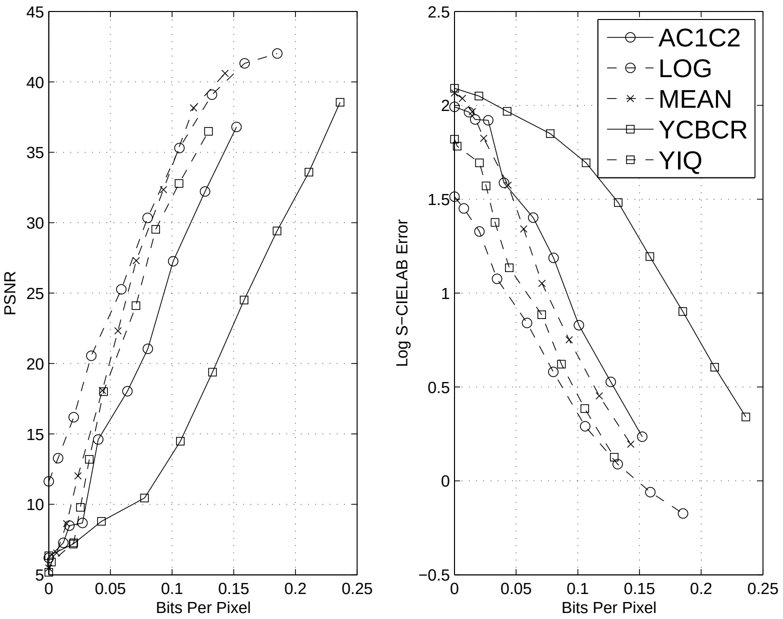


Figure 5. Colour-space performance comparison: The left-hand plot shows the PSNR vs. bits per pixel; the right-hand plot shows the S-CIELAB Error ΔE vs. bits per pixel. The simple windowing method of §3.1 was used for all colour spaces in this comparison.

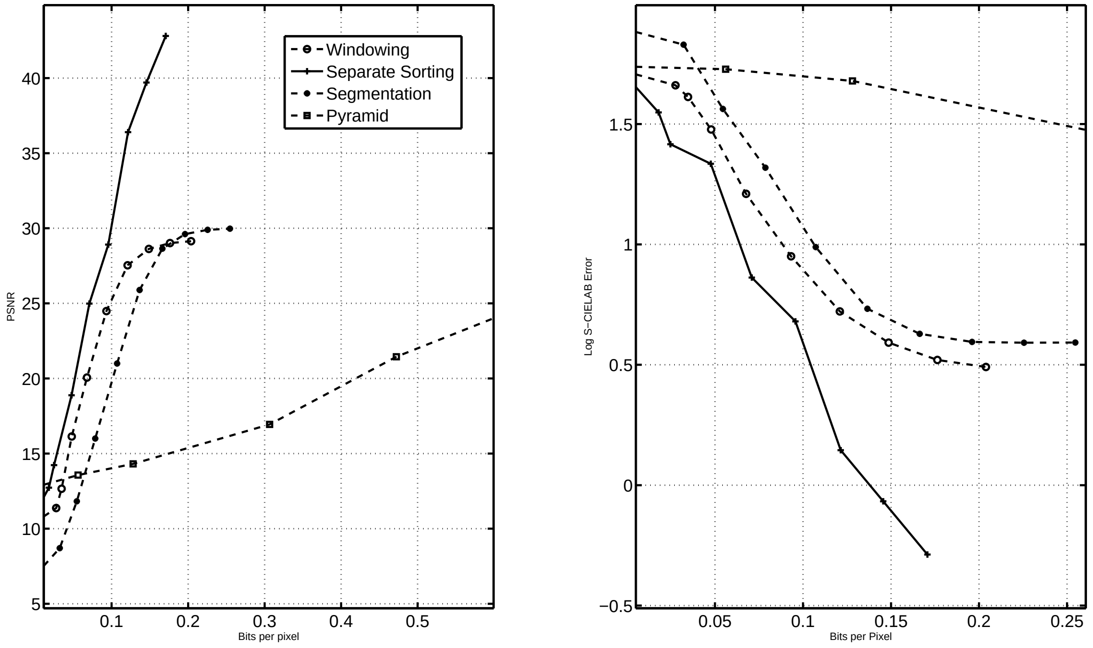


Figure 6. Algorithm performance comparision: The left-hand plot shows the PSNR vs. bits per pixel (not including greyscale information and the overheads); the right-hand plot shows the S-CIELAB error vs. bits per pixel. The log-geometric-mean colour space was used for all algorithms compared here.

APPENDIX A. OBTAINING COLOUR SPACE

A.1 Log-Geometric-Mean Luminance

We firstly transform RGB colour space into a new colour space, with axes $\log R$, $\log G$ and $\log B$. Then we project all the pixels along the vector $\overrightarrow{u} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$. These projections essentially give the log-geometric-mean of the RGB colour space, which we adopt as the luminance of the pixels. Specifically, $\overrightarrow{u} \cdot \log \overrightarrow{RGB}$ is $\sqrt{3}$ times the log of the geometric mean of R, G, B. Next, we have to find two normalized vectors orthogonal to $\overrightarrow{u}$ to act as the chromaticity of the pixels. Gram-Schmidt Orthogonalization may be used here to find these orthonormal bases. But a faster technique to transform one orthonormal basis to another orthonormal basis is by using the Singular Value Decomposition. The projector onto the 2-D subspace orthogonal to the 1-D subspace along $\overrightarrow{u}$ is

$$\boldsymbol{O} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \overrightarrow{u}\,\overrightarrow{u}^T \tag{5}$$

SVD states

$$\boldsymbol{O} = U \Sigma U^T \tag{6}$$

where, U is an orthogonal matrix, and Σ is a diagonal matrix with entries $(1, 1, 0)$. The first two columns of U represent the chromaticity basis vectors. After finding these basis vectors, we calculate the chromaticity of the image by taking the projection of pixel values over these basis vectors.

A.2 AC_1C_2

For completeness, we also collect pertinent information about the other colour spaces tested.

AC_1C_2 is an opponent colour space, with A being the achromatic component and C_1,C_2 being the chromaticity components.[8] First RGB (with gamma correction removed) is converted into XYZ space, with a normalizing white point for standard illuminant D65 (roughly corresponding to mid-day in Western Europe). Here we use the standard sRGB to XYZ conversion.[11] This is then converted to AC_1C_2 space using the set of equations (7):

$$A = 0.2787X + 0.7218Y - 0.1066Z; \tag{7a}$$

$$C_1 = -0.4488X + 0.2898Y + 0.0772Z; \tag{7b}$$

$$C_2 = 0.0860X - 0.5900Y + 0.5011Z; \tag{7c}$$

C_1 is red-blue and C_2 is blue-yellow.

A.3 YIQ

RGB space is converted into YIQ space using the set of equations (8) (cf. Ref.[7]).

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

$$I = 0.595716R - 0.274453G - 0.321263B \tag{8b}$$

$$Q = 0.211456R - 0.522591G + 0.311135B \tag{8c}$$

APPENDIX B. TIKHONOV REGULARIZATION

Tikhonov Regularization, also known as ridge regression, is used for solving non-linear least squares problems. That is, we wish to solve the linear equation given by

$$Ax = b \tag{9}$$

If A is singular in eq. (9), then it yields a non-unique solution. In general, using regularization will generate a better behaved solution than a straightforward Moore-Penrose inverse. A regularization term can also be included in the minimization of the least squares problem to give preference to certain solutions.

The solution obtained after regularization, $\hat{x}$, can be written as

$$\hat{x} = (A^T A + \Gamma^T \Gamma)^{-1} A^T b \tag{10}$$

where Γ denotes the Tikhonov Matrix which is usually taken as the identity matrix times a small value.

APPENDIX C. S-CIELAB ERROR

As discussed in,[8] the input image is initially converted into AC_1C_2 opponent colour space.

After transforming into opponent colour space, this method applies a special combination of spatial blurrings to the image.

$$f = k \sum_i w_i E_i \tag{11}$$

where E_i in (11) denotes a Gaussian filter

$$E_i = k_i \exp\left[-\frac{x^2 + y^2}{\sigma_i^2}\right] \tag{12}$$

The scale factor k_i is adjusted to make the sum of E_i equal to 1. By adjusting w_i and σ_i different filters are used in different channels to match the blurring in the human visual system.

Table 1. Weights and spreads for different planes

Plane	Weights w_i	Spreads σ_i
Lum	0.921	0.0283
	0.105	0.133
	-0.108	4.336
Red-green	0.531	0.0392
	0.330	0.494
Blue-yellow	0.488	0.0536
	0.371	0.386

REFERENCES

[1] Gershikov, E., Lavi-Burlak, E., and Porat, M., "Correlation-based approach to color image compression," *Signal Processing: Image Commun.* **22**(9), 719–733 (2007).

[2] Funt, B. V., Drew, M. S., and Brockington, M., "Recovering shading from color images," in [*ECCV-92: 2nd European Conf. on Computer Vision*], Sandini, G., ed., 124–132 (1992).

[3] San, X., Cai, H., and Li, J., "Color image coding by using inter-color correlation," in [*ICIP06*], 3117–3120 (2006).

[4] Drew, M. and Bergner, S., "Spatio-chromatic decorrelation for color image compression," *Signal Processing: Image Communication* **23**(8), 599–609 (2008).

[5] Goffman-Vinopal, L. and Porat, M., "Color image compression using inter-color correlation," in [*ICIP02*], (2002).

[6] Wyszecki, G. and Stiles, W., [*Color Science: Concepts and Methods, Quantitative Data and Formulas*], Wiley, New York, 2nd ed. (1982).

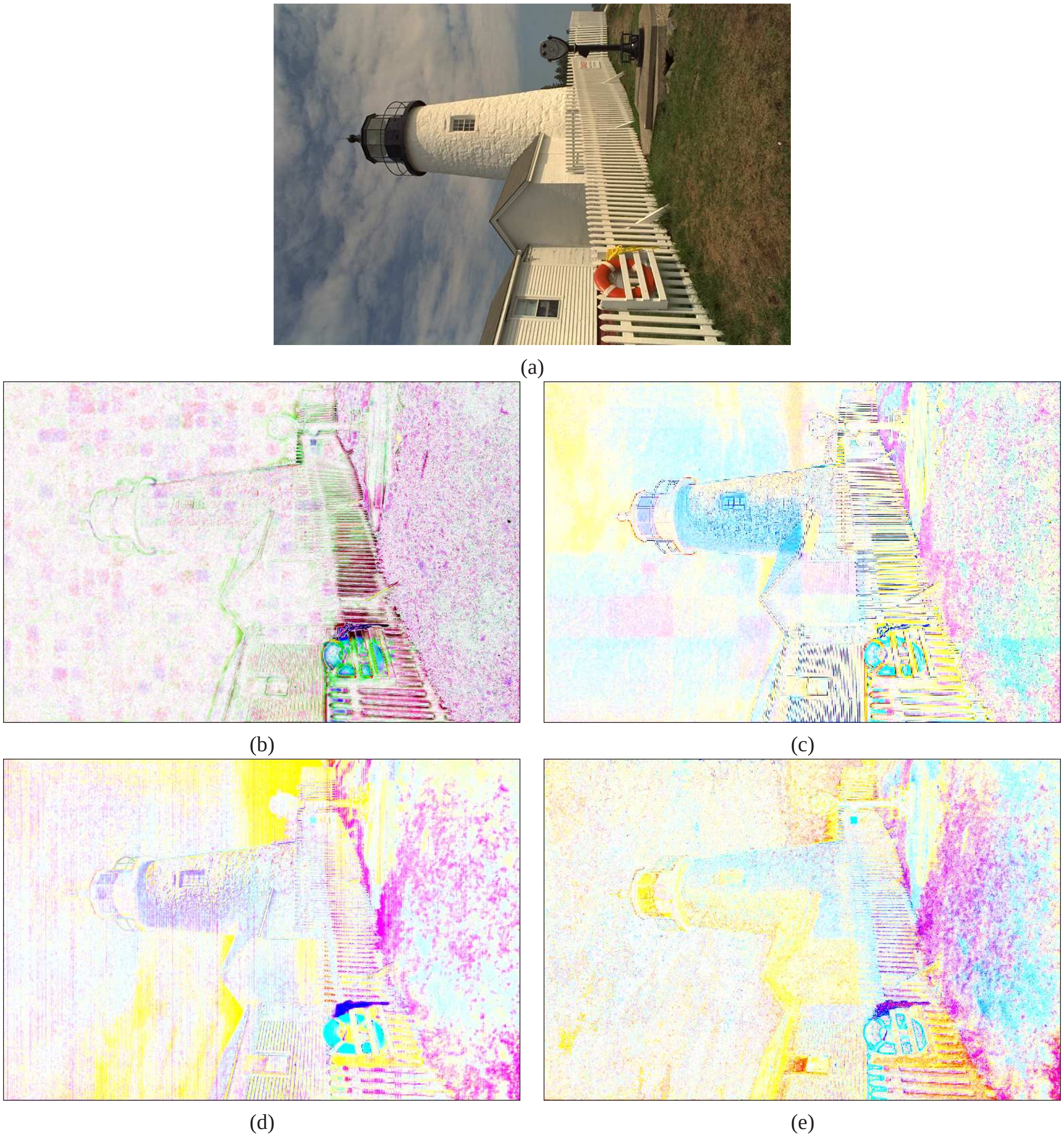
[7] Li, Z.-N. and Drew, M., [*Fundamentals of Multimedia*], Prentice-Hall (2004).

[8] Zhang, X. and Wandell, B., "A spatial extension of CIELAB for digital color image reproduction," *SID Journal* (1997).

[9] Comaniciu, D. and Meer, P., "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 603–619 (2002).

[10] Huynh-Thu, Q. and Ghanbari, M., "Scope of validity of PSNR in image/video quality assessment," *Electronics Letters* **44**(13) (2008).

[11] International Electrotechnical Commission, "Multimedia systems and equipment – colour measurement and management – part 2-1: Colour management – default RGB colour space – sRGB." IEC 61966-2-1:1999.



(a)

(b)

(c)

(d)

(e)

Figure 7. (a): Original image. (b): Error (inverse from white) for Windowing method, psnr=34.82, ΔE=1.95, bpp=0.1583. (c): Error for Pyramid method, psnr=33.40, ΔE=2.75, bpp=0.1330. (d): Error for Segmentation method, psnr=34.00, ΔE=2.11, bpp=0.1492. (e): Error for Separate method, psnr=**36.80**, ΔE=**1.32**, bpp=0.1662. Separately sorting chrominance components gives the best results.