Working with Compressed Video Data

Problem presented by

Glynn Wright

Aralia Systems Ltd

ESGI100 was jointly hosted by
Smith Institute for Industrial Mathematics and System Engineering
University of Oxford

with additional financial support from
Engineering and Physical Sciences Research Council
European Journal of Applied Mathematics
Oxford Centre for Collaborative Applied Mathematics
Warwick Complexity Centre

Report author

David Barton (University of Bristol)

Executive Summary

Aralia Systems Ltd is interested in feature and colour extraction technologies for MPEG video streams, particularly for video streams of relatively low quality where visual artefacts are common. Here we present findings on removing visual artefacts and how colours can be compared in a manner more consistent with the human visual system.

Version 1.0
July 11, 2014
iv+16 pages

Contributors

Russell Davies (University of Cardiff)
Jens Gravesen (Technical University of Denmark)
Marta Polak (Wrocław University of Technology)
Clarice Poon (University of Cambridge)
Monika Sikora (Wrocław University of Technology)

Contents

1 Introduction

(1.1) Aralia Systems Ltd process large quantities of MPEG videos to enable searching of video streams for particular features (e.g., red cars or blue T-shirts). While feature recognition is not considered to be an issue (at least for the purposes of the study group), colour recognition is a issue.

(1.2) To process the videos, the key frames are isolated from the video streams and are then segmented to determine the key features in the frame.The features are then stored in a database for easy querying and retrieval at a later date. The key frames themselves are can be thought of as JPEG images within the video stream.

(1.3) There appear to be two key problems with this process.

(1.4) Firstly, the segmentation algorithm is not robust to artefacts in the decompressed image. The nature of the video streams is that the visual data is highly compressed to conserve bandwidth. As such, the extracted images are relatively low quality. A common artefact of low-quality JPEG images is the presence of blocks of colour in the image (i.e., the image looks blocky). This arises from the fact that JPEG images are encoded in a regular block pattern. The end result is sudden changes in colour (rather than smooth transitions) that cause the segmentation algorithm to treat the sudden change as the start of a new feature.

(1.5) Secondly, the human eye does not have uniform sensitivity to all colours. For certain colours (e.g., green) we are able to distinguish many different shades and tones, whereas for others (e.g., red) we are unable to. As such, the simple Euclidean distance metric on the RGB colour space does not match with our visual perception.

(1.6) Here we present our findings on both of these two problems.

2 Treating the JPEG transform as an inverse problem

(2.1) The JPEG transform is a lossly transformation, as such the inverse transform in a so-called inverse problem, for which there are many potential solutions. The inverse JPEG transform as defined in the JPEG standard is one solution, but other solutions may be obtained by considering different regularisations of the problem (e.g., by enforcing smoothness by use of wavelets).

(2.2) To understand the problem of inverting the JPEG transform, we first consider how JPEG compresses an image.

Given $f \in \mathbb{R}^{N\times N}, Q \in \mathbb{R}^{8\times 8}$, let

$$J_Q : f \mapsto \{(J_Q f)_j \in \mathbb{R}^{8\times 8} : j = 1, \dots, N^2/64\}$$

denote the following operations:
First divide f into blocks of 8×8 pixels to obtain $\{f_j \in \mathbb{R}^{8\times 8} : j = 1, \dots, N^2/64\}$. Then, to each f_j

1. apply the discrete cosine transform: $DCT(f_j)$.
2. apply a quantization transform:

$$DCT(f_j) \mapsto \text{round}(DCT(f_j)/Q) =: (Jf)_j,$$

where / denotes the pointwise division of two matrices.

Due to this rounding operation at the end, the operator J is non-invertible.

(2.3) The JPEG compression process on an image I is as follows,

$$JPEG : \begin{pmatrix} I_R \\ I_G \\ I_B \end{pmatrix} \xrightarrow{T} \begin{pmatrix} I_Y \\ I_{C_b} \\ I_{C_r} \end{pmatrix} \xrightarrow{J} \begin{pmatrix} J_{Q_{Y,q}}(I_Y) \\ J_{Q_{C,q}}(I_{C_b}) \\ J_{Q_{C,q}}(I_{C_r}) \end{pmatrix}$$

where T transform the (R, G, B)-colour space representation of the I to the (Y, C_b, C_r)-colour space representation of I, and the quantization matrices $Q_{Y,q}$ and $Q_{C,q}$ are dependent on a quality factor $q \in [1, 100]$.

2.1 The standard JPEG reconstruction

(2.4) For each JPEG compressed component of the form $J_Q(f)$, the standard JPEG decompression process $\hat{f} = \hat{J}_Q(J_Q(f) \in \mathbb{R}^{N\times N}$ is such that by dividing $\hat{f}$ into blocks of 8×8 pixels to get $\{\hat{f}_j : j = 1, \dots, N^2/64\}$, each block $\hat{f}_j$ satisfies

$$\hat{f}_j = DCT^{-1}(Q \cdot (J_Q f)_j). \tag{1}$$

where $\cdot$ denotes a pointwise multiplication of two matrices.

(2.5) **Note that since J is non-invertible, $\hat{f}$ is simply an approximation of f and can exhibit undesirable artefacts**. The JPEG decompression process is written as

$$\begin{pmatrix} J_{Q_{Y,q}}(I_Y) \\ J_{Q_{C,q}}(I_{C_b}) \\ J_{Q_{C,q}}(I_{C_r}) \end{pmatrix} \xrightarrow{\hat{J}} \begin{pmatrix} \hat{I}_Y \\ \hat{I}_{C_b} \\ \hat{I}_{C_r} \end{pmatrix} \xrightarrow{T^{-1}} \begin{pmatrix} \hat{I}_R \\ \hat{I}_G \\ \hat{I}_B \end{pmatrix}$$

where

$$\begin{pmatrix} \hat{I}_Y \\ \hat{I}_{C_b} \\ \hat{I}_{C_r} \end{pmatrix} = \begin{pmatrix} \hat{J}_{Q_{Y,q}}(J_{Q_{Y,q}}(I_Y)) \\ \hat{J}_{Q_{C,q}}(J_{Q_{Y,q}}(I_{C_b})) \\ \hat{J}_{Q_{C,q}}(J_{Q_{Y,q}}(I_{C_r})) \end{pmatrix}$$

and $(\hat{I}_R, \hat{I}_G, \hat{I}_B)$ is the (R, G, B) representation of $(\hat{I}_Y, \hat{I}_{C_b}, \hat{I}_{C_r})^T$.

2.2 The proposed solution

(2.6) The artefacts introduced by the JPEG decompression process arise because J is not invertible due to the rounding operation it carries out, and in general $\hat{J}_Q(J_Q(f)) \neq f$.

(2.7) We propose a regularized reconstruction. Given the data $J_Q f$, the JPEG decompression process can be improved if we can recover f by solving for $x \in \mathbb{R}^{N\times N}$,

$$J_Q x = J_Q f. \tag{2}$$

The problem is that there can be many solutions x satisfying this equation.

(2.8) However, typical images which are of interest tend to be 'sparse' with respect to wavelets. This means that for a typical image f and some discrete wavelet transform W, most of the entries of Wf will be either zero or insignificant. When faced with this type of ill-posed inverse problems, a common approach is to apply *wavelet regularization*. So, given the compressed data $J_Q(f)$, instead of approximating f using $\hat{J}$ described above, we let the reconstruction of f be f_{reg}, where

$$f_{reg} \in \operatorname*{argmin}_{x\in\mathbb{R}^{N\times N}} \|Wx\|_1 \text{ subject to } Jx = Jf. \tag{3}$$

(2.9) Note that whilst $\hat{J}_Q(J_Q(f))$ does not necessarily satisfy (2), f_{reg} is a solution of (2).

(2.10) The final hurdle to implementing this is to develop a method to solve the minimization problem (3), which is equivalent to solving

$$\min_{x\in\mathbb{R}^{N^2}} \|x\|_1 \text{ subject to } J_Q W^{-1} x = J_Q f.$$

Over the last decade, there have been many algorithms developed for the purpose of solving such minimization problems. During ESGI100, I used a software package called spgl1 [9, 8], which is an l^1 minimization solver, designed for solving minimization problems of the form

$$\min_{x\in\mathbb{R}^{N}} \|x\|_1 \text{ subject to } Ax = b$$

where b is the observed data, and A is a linear operator. In our case, J_Q is a nonlinear operator, so spgl1 is not exactly appropriate for (3). Despite this, we obtained promising numerical results by letting $A = J_Q W^{-1}$ where W is the wavelet transform for boundary corrected Daubechies wavelets. (In order to solve an l^1 minimization problem using spgl1, we are required to implement the forward operator A and the adjoint operator A^*. In our case, the adjoint of $J_Q W^{-1}$ does not exist. So, what I did was to implement implement the forward operator correctly as $J_Q W^{-1}$, but the adjoint as $W\tilde{J}_Q^*$, there $\tilde{J}_Q$ is J_Q without any quantization. Doing so means that spgl1 will not necessarily converge to a minimizer.)

(2.11) In the following examples, the quality of the original images have been compressed with quality $q = 25$, where q is the standard measure of quality in JPEG, as describe in [1]. **It is likely that there will be significant improvement in the numerical results by developing a more appropriate solver.** We present the reconstructions obtained via solving (3) (using spgl1 and boundary corrected Daubechies wavelets) with the standard JPEG decompressed reconstruction of (1).

2.3 Kernel recovery Techniques

(2.12) As an alternative to general inverse problem methods, kernel methods [7] have established themselves as powerful tools in recovering smooth functions from unstructured data. There are fast algorithms available for optimal recovery of smooth approximations from unstructured data — this sort of method is likely to be much faster than solving more general inverse problems.

(2.13) This kind of recovery is likely to improve decompressed images where the underlying image is smooth. It is not clear how well it would work in recovering images with sharp discontinuities.

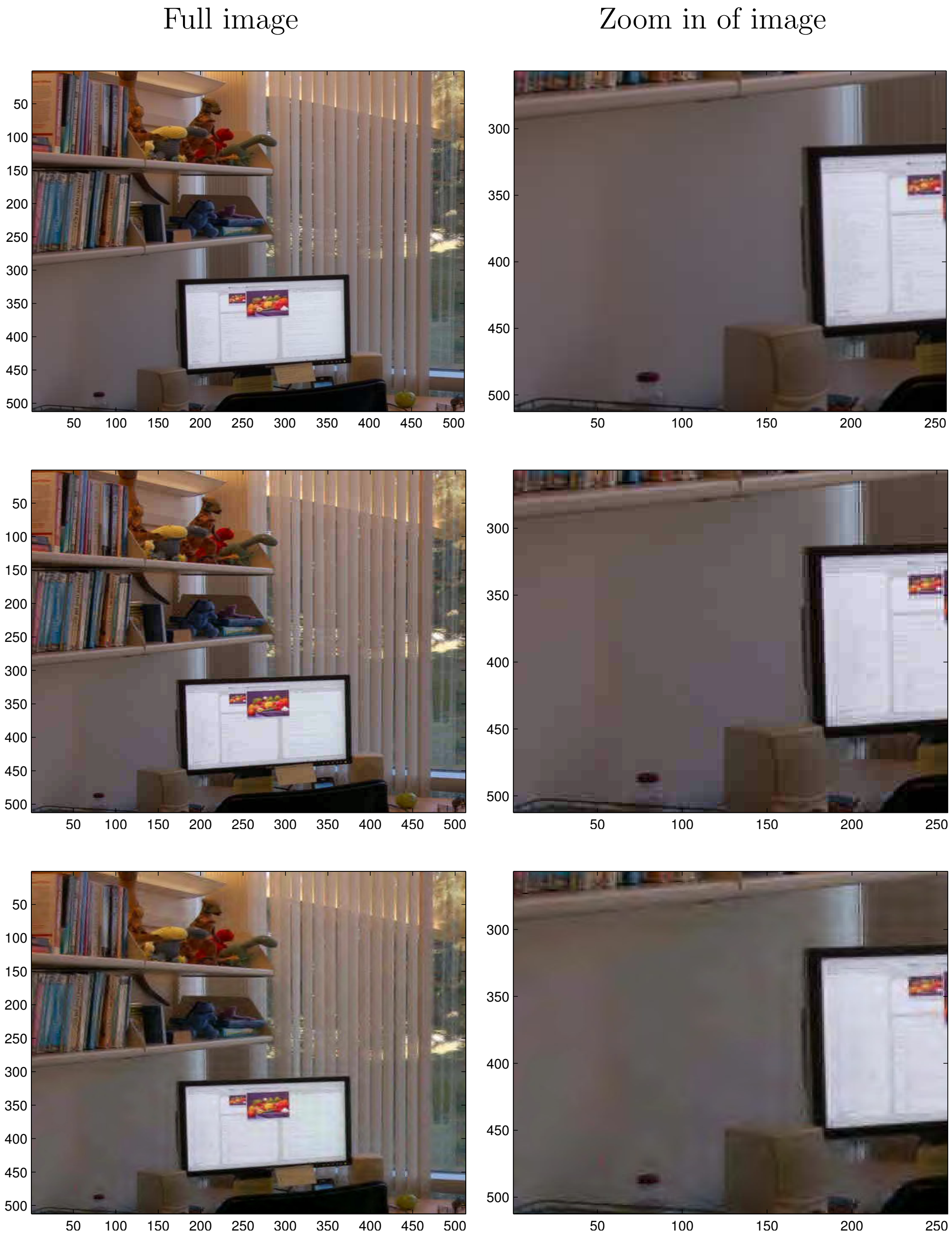
3 The colour clustering

(3.1) We based on the algorithm named Region Growing, which chooses the region with similar colour as a seed point. We modified MATLAB function created by Dirk-Jan Kroon [5], where input agruments are image, coordinates of the seed point and maximum intensity distance. The difference between a pixel's intensity value and the region's mean, is used as a measure of similarity. This process stops when the intensity difference between region mean and new pixel become larger than a certain treshold. Output arguments are mean colour of the region and the logical output image of region.

(3.2) The main goal of segmentation is partition an image into regions with similar colours. In function `im_seg` using `region_growing2` image is segmented as follows:

- start from the seed point coordinates $(1, 1)$;
- `region_growing2` marks the region and the mean colour;
- choose first point which is not classified to any of regions before and repeat `region_growing2`.

The process stops when every pixel is classified into the certain region.



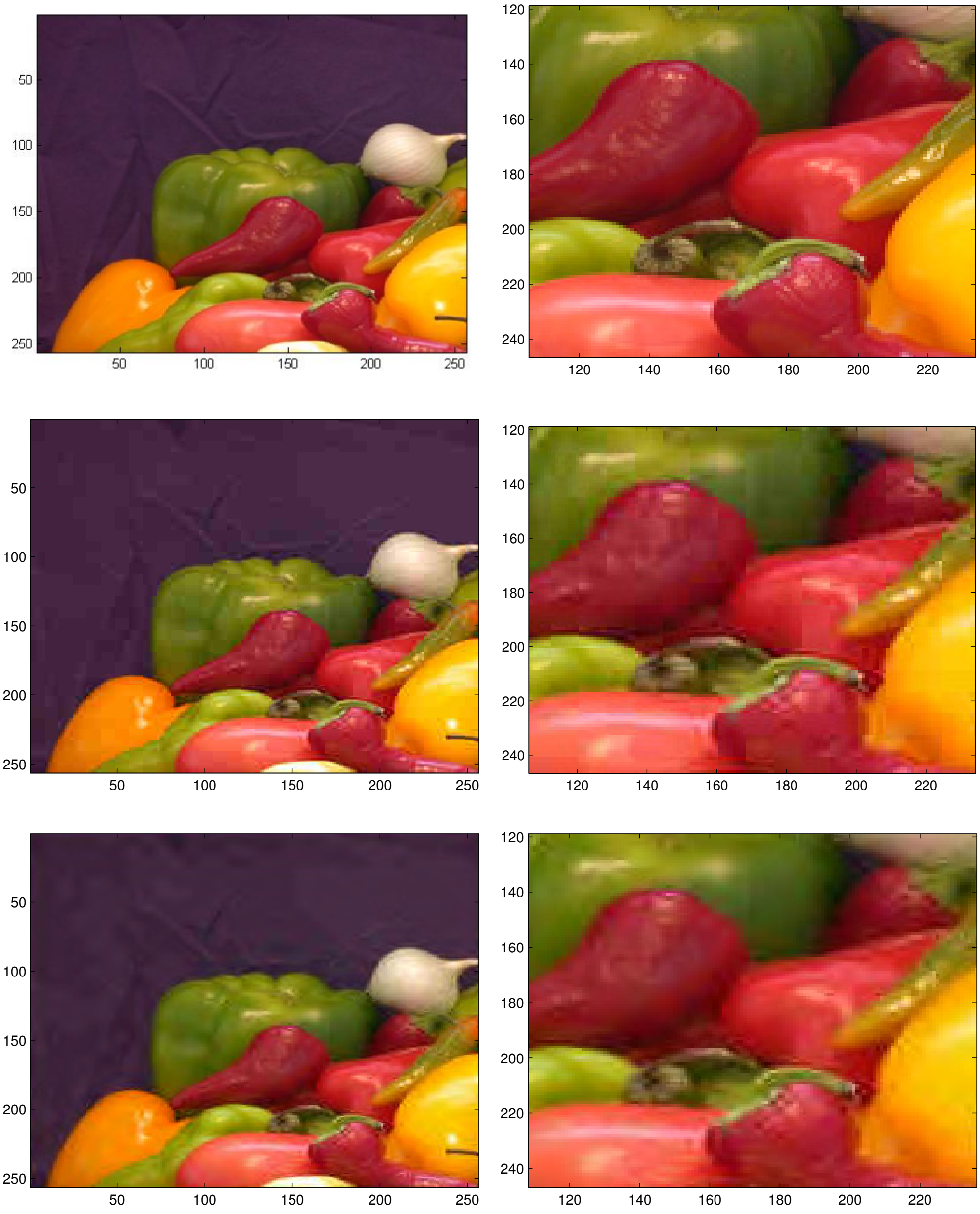
Full image

Zoom in of image

Figure 1: The reconstructions here are of a 512 × 512 crop of the MATLAB image offices_4.jpg. Top row: Original image. Middle row: Standard JPEG reconstruction. Bottom row: Regularized reconstruction with DB-6 wavelets

Full image

Zoom in of image



Standard JPEG reconstruction

DB-7 wavelet regularized reconstruction

Figure 2: The reconstructions here are of a 256 × 256 crop of the MATLAB image peppers.png. Top row: Original image. Middle row: Standard JPEG reconstruction. Bottom row: Regularized reconstruction with DB-7 wavelets

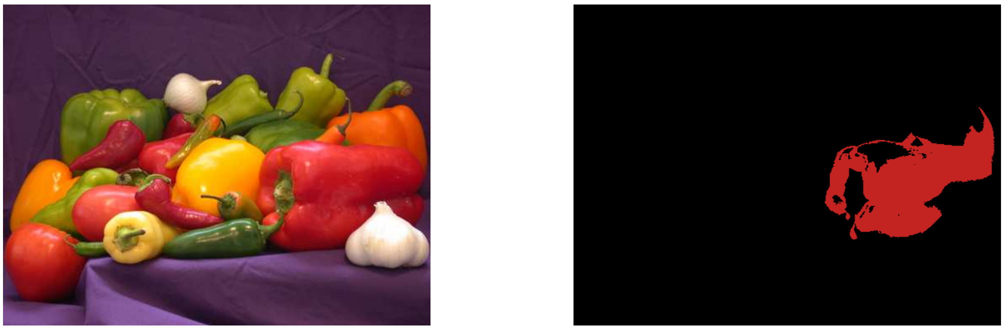


Figure 3: The original image (on the left) and the region chose by `region_growing2` with the certain seed point (on the right).

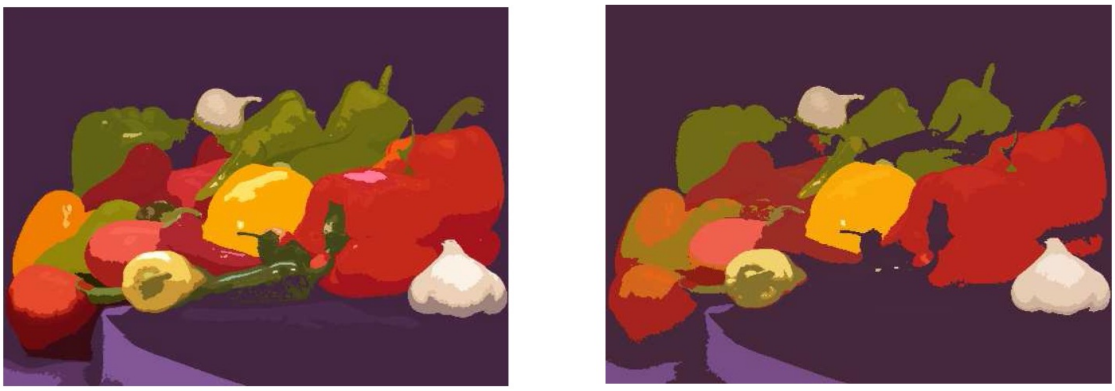


Figure 4: The image segmented by function `im_seg` with maximum intensity distance 0.2 (on the left) and 0.3 (on the right).

(3.3) The last step was to segment image which was in the $L*ab$ colour space, not in RGB as before (`im_seg_lab` function). Comparing Figure 4 and Figure 5 we can see that better segmentaion is received using $L * ab$ colour space (function `im_seg_lab`).

4 Colour metrics

(4.1) We would like define a distance between colours that corresponds to human perception. This is an old problem that has attracted interest from many scientists, including names such as Helmholtz and Schrödinger. We will describe a couple of approaches that can solve the problem approximately.

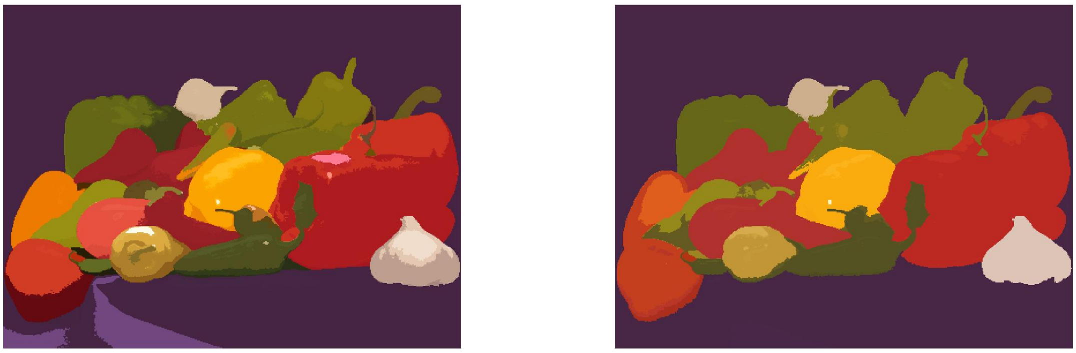


Figure 5: The image segmented by function `im_seg_lab` with maximum intensity distance 20 (on the left) and 30 (on the right).

4.1 Colour Space

(4.2) This section provides some background material on what colour is and is mostly lifted from an earlier study gropup report [4].

(4.3) The human retina has three types of colour photorecepter cone cells, with different spectral sensitivities. A fourth type of photorecepter cells, the rod, is also present. The rods are only used at extremely low light levels (night vision), and do not contribute to the perception of colour.

(4.4) Because the human has exactly three type of colour photorecepter, three real numbers are necessary and sufficient to describe a colour. We can think of the three numbers as the power received by each of the three different colour photorecepters.

(4.5) The International Commission On Illumination, (CIE[1]) has defined several parametrization of the space of colours. If $S : [\lambda_1, \lambda_2] \to \mathbb{R}_+$ is the intensity function for the light, then the *CIE XYZ components* are defined by

$$(X, Y, Z) = k \int_{\lambda_1}^{\lambda_2} S(\lambda) \left(\overline{x}(\lambda), \overline{y}(\lambda), \overline{z}(\lambda)\right) \mathrm{d}\lambda \,, \tag{4}$$

where the functions $\overline{x}(\lambda)$, $\overline{y}(\lambda)$, and $\overline{z}(\lambda)$ are the CIE 1931 Standard Colorimetric Observers, see Figure 6, and k is a normalization constant which makes $Y = 100$ for a standard light source $S(\lambda)$, i.e.,

$$k = \frac{100}{\int_{\lambda_1}^{\lambda_2} S(\lambda)\, \overline{y}(\lambda)\, \mathrm{d}\lambda} \,. \tag{5}$$

[1] `http://www.cie.co.at`

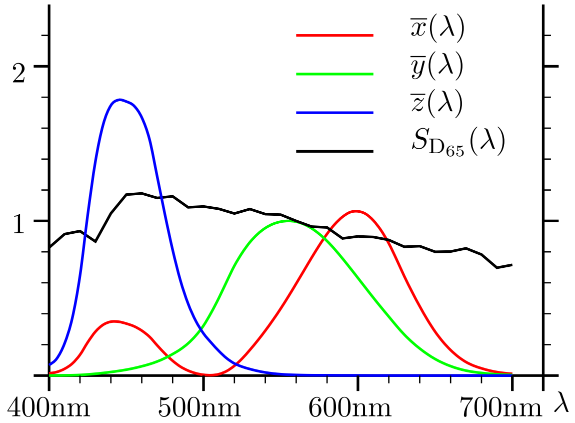


Figure 6: CIE 1931 Standard Colorimetric Observers and the spectral distribution for the CIE illuminant D_{65}. They are tabulated in [10] and can also be found at the CIE web-site [3].

For the CIE standard illuminant D_{65}, see Figure 6, we have $k = 0.047332$. The number Y is called the *luminance* and is an attempt to define the total observed intensity of the light.

(4.6) The *CIE xy chromatic coordinates* are given by

$$x = \frac{X}{X+Y+Z}\,, \qquad y = \frac{Y}{X+Y+Z}\,. \tag{6}$$

Sometimes a third coordinate $z = Z/(X + Y + Z)$ is defined, but it can always be found from the relation $x + y + z = 1$. The two chromatic coordinates x and y describe "pure" colour, in the absence of luminance (or brightness). When monochromatic light sweeps over the visual light range from 400nm to 700nm, it traces a curve in the xy-space, see Figure 7. The line connecting the two ends of the curve is called the line of purples. It joins extreme blue with extreme red and consists consequently of mixtures of blue and red.

(4.7) A colour can be specified by chromaticity (x, y) and luminance Y in the form of the *CIE xyY components*. To recover X and Z the following formulas are used:

$$X = Y\,\frac{x}{y}\,, \qquad Z = Y\,\frac{1-x-y}{y}\,. \tag{7}$$

(4.8) The colours on a computer screen or a television is given by mixing three *primaries*: red, green, and blue. The three primaries corresponds to three points in xy-space and the screen can reproduce all colours in the triangle spanned by the three primaries, the *gamut* of the primaries. In Figure 7 the primaries for the HDTV, [2], are plotted and it is easily seen that not all colours can be obtained. The actual colours in the plot need not be correct, they depends on the computer screen, or on the printer and the

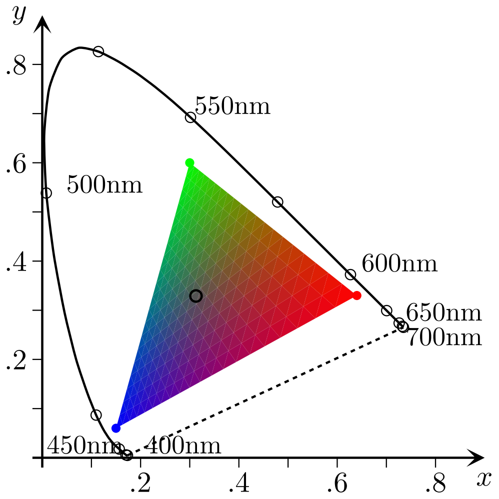


Figure 7: The tristimulus diagram. The monochromatic colours lie on the curved part of the boundary. The dashed line joining the the end of the visible spectrum [400nm, 700nm] is the line of purples. The triangle are the colours that can be produced by the primaries of the Rec. 709 RGB specifications (HDTV), [2]. The circle indicates the D_{65} white point.

illumination. Other devices such as a computer screen, a projecter, etc. also have three primary colours and can only reproduce the colours in some triangle.

4.2 Colour Differences

(4.9) The human perception of similar colours has not much to do with the Euclidean distance in the xy-plane. Indeed, some sixty years ago MacAdam conducted some colour matching experiments where a person was asked to match a colour with given chromatic coordinates (x, y) by adjusting another colour by a single control that traced a line through (x, y) in the chromatic plane. The standard deviation turned out to be ellipses in the chromatic plane, see Figure 8.

(4.10) If the ellipses are enlarged approximately three times they define the *just noticeable difference*, i.e., colours inside the enlarged ellipse appears to be the same as the one at the center while colours outside appears to be different from the colour at the center. This discrepancy between human perception and the Euclidean distance has spawned several attempts to define parameters which are more uniform with respect to the human perception.

(4.11) MacAdams experiments has been repeated and also extended to include intensity so we have ellipsoids in the $xy\ell$ colour space, where $\ell = 0.2\log_{10}(Y)$, see [10, Tables I and II(5.4.2), and I(5.4.3–4)]. We don't have the 3D results in electronic form so all numerical experiments are done using the MacAdam ellipses in the 2D xy colour space.

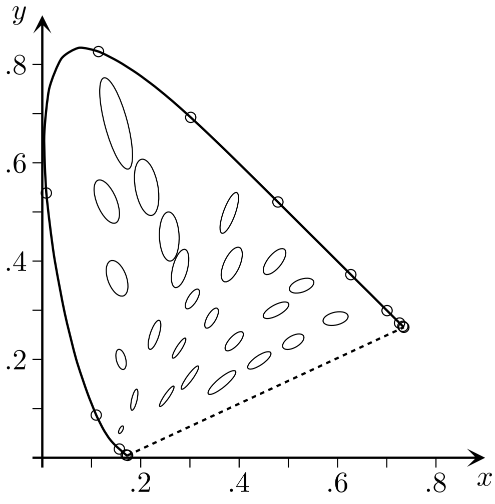


Figure 8: The MacAdam ellipses, [6], enlarged 10 times. If the depicted ellipses are diminished by a factor of three then colours on the ellipse can just be seen to be different from the colour at the center. The parameters of the ellipses was found in [10, Table 2(5.4.1)].

4.3 Small Distances

(4.12) Assuming the metric (or first fundamental form or line element) is constant inside the ellipse the elleipse determine the metric. Indeed, if the major and minor semiaxis has length a and b respectively and the major semiaxis forms the angle θ with the x-axis then the components of the metric is given by

$$\begin{aligned}\begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} &= \begin{pmatrix} E & F \\ F & G \end{pmatrix} \\ &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1/a^2 & 0 \\ 0 & 1/b^2 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \\ & \begin{pmatrix} \dfrac{\cos^2\theta}{a^2} + \dfrac{\sin^2\theta}{b^2} & \left(\dfrac{1}{a^2} - \dfrac{1}{b^2}\right)\cos\theta\,\sin\theta \\ \left(\dfrac{1}{a^2} - \dfrac{1}{b^2}\right)\cos\theta\,\sin\theta & \dfrac{\sin^2\theta}{a^2} + \dfrac{\cos^2\theta}{b^2} \end{pmatrix} . \end{aligned} \tag{8}$$

(4.13) The length of a vector $(\Delta x, \Delta y)$ originating from the centre can now be found as

$$\|(\Delta x, \Delta y)\|^2 = \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} . \tag{9}$$

(4.14) There are two problems with this approach. The most severe is that the metric should be close to constant along the vector, i.e., it is only good for small distances and is probably not suitable for the application at hand. The other problem is that the metric is only known in a finite number of points so we have to extend it to all of colour space, or at least to the RGB-triangle.

4.4 Extrapolating the MacAdam Ellipses or the Metric

(4.15) We are given the centres $(\widehat{x}_k, \widehat{y}_k) \in [0,1]^2$ and values $\widehat{a}_k, \widehat{b}_k, \widehat{\theta}_k$ of the major axis, the minor axis and the angle of the major axis of the MacAdam ellipses, respectively. We extrapolate these values to the rectangle $\Omega = [0.1, 0,7] \times [0.0, 0.7]$ by solving the following linear constrained qudratic optimisation problem:

$$\text{minimise} \int_0^1 \int_0^1 \left|\frac{\partial^2 a}{\partial x^2}\right|^2 + 2\left|\frac{\partial^2 a}{\partial x \partial y}\right|^2 + \left|\frac{\partial^2 a}{\partial y^2}\right|^2 \, \mathrm{d}x \, \mathrm{d}y \,, \tag{10}$$

$$\text{such that } a(\widehat{x}_k, \widehat{y}_k) = \widehat{a}_k, \quad k = 1, \ldots, K \,, \tag{11}$$

$$\text{and } a(x,y) \geq 0 \quad (x,y) \in \Omega \,, \tag{12}$$

and similar for b and θ, except that (12) is omitted for θ. Using B-splines to represent the functions a, b, and θ the resulting ellipses can be seen to the left in Figure 9.

(4.16) Note that the domain ω doen't include all of the visible colours, but it does include the RGB-triangle. If we want to include all of the visible colours we just have to enlarge the domain Ω.

(4.17) The resulting field of ellipses correspond to a metric tensor (first fundamental form or line element) with components given by (8).

(4.18) This does not generalise easily to 3D so alternatively we could extrapolate the components g_{ij} of the metric, but it is difficult to keep the extrapolated metric positive definite. To overcome this we could take the matrix logarithm of the metric tensor found at the centre of the MacAdam ellipses, extrapolate these using (10) and (11) and finally take the matrix exponential of the result.

4.5 Colour Distance

(4.19) We now seek a map $(\xi, \eta) : \Omega \to \mathbb{R}^2$ such that the euclidean distance in $\xi\eta$ space is in good agreement with human perception of colour distance. That is, we want the that the image of the ellipses to be circles of equal size. This is the same as saying that we have an isometry with respect to the extrapolated metric on Ω and the standard euclidean metric on $\mathbb{R}^2$.

(4.20) If the major and minor axis of the ellipses, or equivalently the eigenvectors of the metric tensor, maps to the standard basis in $\mathbb{R}^2$ then we do have an isometry. The Jacobian of our map is

$$J = \begin{pmatrix} \partial\xi/\partial x & \partial\xi/\partial y \\ \partial\eta/\partial x & \partial\eta/\partial y \end{pmatrix} \,, \tag{13}$$

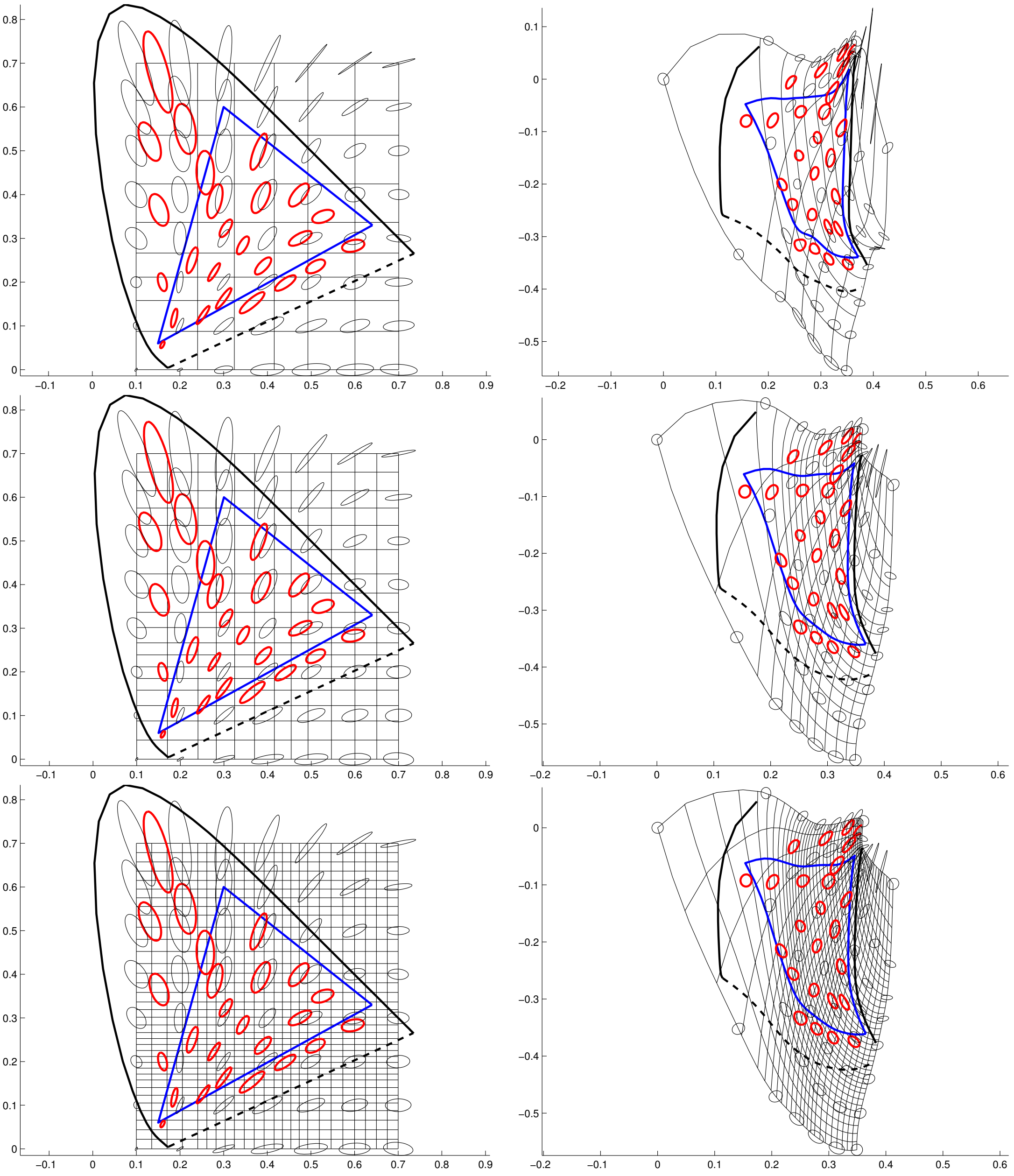


Figure 9: To the left the xy-colour space with the MacAdam ellipses in red. The black ellipses are the result of interpolating (and extrapolating) the MacAdam ellipses. To the right the image under a spline map that tries to map the non constant metric in colour space to the standard euclidean metric. We have used different knots in the different lines, indicated by the thin lines.

the eigenvectors are $(a\,\cos\theta, a\,\sin\theta)$ and $(-b\,\sin\theta, b\,\cos\theta)$, respectively, and they maps to the standard basis if

$$J\begin{pmatrix} a\,\cos\theta & -b\,\sin\theta \\ a\,\sin\theta & b\,\cos\theta \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{14}$$

or equivalently

$$J = \begin{pmatrix} a\,\cos\theta & -b\,\sin\theta \\ a\,\sin\theta & b\,\cos\theta \end{pmatrix}^{-1} = \frac{1}{ab}\begin{pmatrix} b\,\cos\theta & b\,\sin\theta \\ -a\,\sin\theta & a\,\cos\theta \end{pmatrix}\,. \tag{15}$$

We now solve the quadratic optimisation problem

$$\text{minimise} \int_0^1\int_0^1 \left(\frac{\partial\xi}{\partial x} - \frac{\cos\theta}{a}\right)^2 + \left(\frac{\partial\xi}{\partial y} - \frac{\sin\theta}{a}\right)^2 \mathrm{d}x\,\mathrm{d}y \tag{16}$$

$$\text{such that } \xi(0.1, 0.0) = 0\,, \tag{17}$$

and similar for η. Expressing ξ and η using the same B-splines a before we obtain the images to the right in Figure 9.

(4.21) Looking at larger versions of these images we can see that the map in the top row is not one to one, but the maps in the other two rows are. We can also see that there is very little difference between the maps in the second and third row.

(4.22) If we look at the map in the second row we see that the ellipses have become more similar in size and more like circles, especially inside the RGB-triangle.

(4.23) The $\xi\eta$ map is also defined for colours outside the visible and also tries to make the ellipses into circles here. This is of course unnecessary and it would be a good idea to introduce a weight function that has a higher weight on the visible colours. Alternatively it could have a higher weight inside the RGB-triangle than outside, see Figure 10.

(4.24) The advantage of this approach is that it leads to a quadratic programming problem which has unique solution and is realtively cheap to solve. The disadvantage is that we restrict the set of possible sollutions. We ask that the images of the eigenvectors (or semiaxis) maps to the standard basis but it is sufficient (and necessary) that they map to an arbitrary orthonormal basis. Equivalently, the map is an isometry if and only if

$$J^T J = \begin{pmatrix} \xi_x^2 + \eta_x^2 & \xi_x\,\xi_y + \eta_x\,\eta_y \\ \xi_x\,\xi_y + \eta_x\,\eta_y & \xi_y^2 + \eta_y^2 \end{pmatrix} = \begin{pmatrix} E & F \\ F & G \end{pmatrix}\,, \tag{18}$$

where ξ_x, ξ_y, η_x, and η_y denotes the partial derivatives. So we try solve the constrained optimisation problem

$$\text{minimise} \int_0^1\int_0^1 \left(\xi_x^2 + \eta_x^2 - E\right)^2 + 2\left(\xi_x\,\xi_y + \eta_x\,\eta_y - F\right)^2 + \left(\xi_x^2 + \eta_x^2 - G\right)^2 \mathrm{d}x\,\mathrm{d}y \tag{19}$$

$$\text{such that } \det J \geq 0\,, \quad (x,y)\in\Omega\,. \tag{20}$$

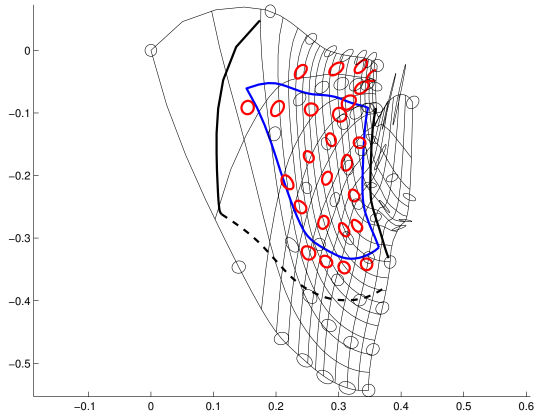


Figure 10: To the left the result of the optimisation when the visible colors are weighted a 100 times more. To the right when the colours in the RGB triangle are weighted 100 times more.

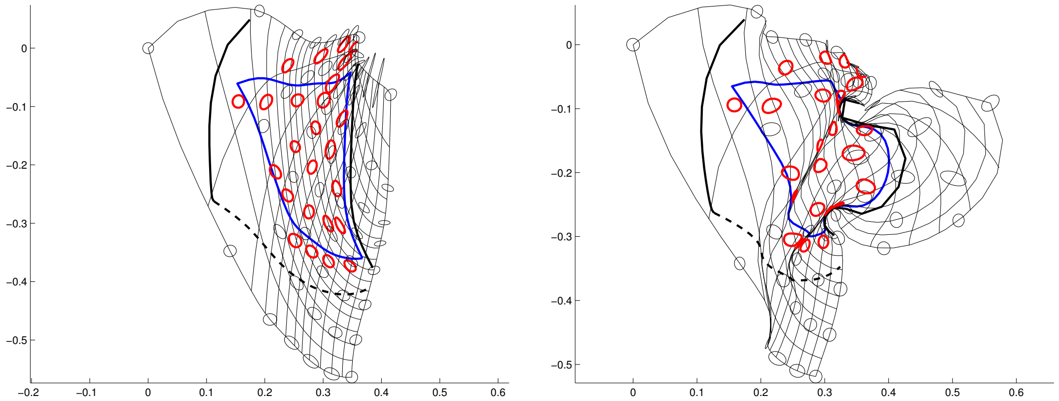


Figure 11: To left map from the second row of Figure 9 which is used as the initial guess in the optimisation problem (19). To the right the result of the optimisation. The objective function (19) has the value 0.198571 at the start and the value 0.0170004 at the end of the optimisation. There are many active constraints at the end which means the map is (nearly) singular.

(4.25) The result can be seen in Figure 11. The result doesn't look very good, but closer inspection reveals that the ellipses indeed are more like circles especially outside the set of visible colours. So here it is definitely a good idea to use weight functions that emphasise the visible colours (or the *RGB* colours) at the expense of the non visible colours.

Bibliography

[1] `http://www.ams.org/samplings/feature-column/fcarc-image-compression`.

[2] ITU-R Recommendation BT. 709. *Basic Parameter Values for the HDTV*

Standard for the Studio and for the International Program Exchange. ITU, 1211 Geneva 20, Switzerland, 1991.

[3] CIE. Cie colour matching functions. `http://cvrl.ioo.ucl.ac.uk/cmfs.htm`.

[4] J. Gravesen and S Noble. Sequencing spinning lines. Technical report, European Study Group with Industry, ESGI 43 Lancaster, 2002. `http://www.maths-in-industry.org/miis/17/1/Sequencing.pdf`.

[5] Dirk-Jan Kroon. Region growing. `http://www.mathworks.com/matlabcentral/fileexchange/19084-region-growing`.

[6] D. L. MacAdam. Visual sensitivities to color differences in daylight. *J. Optical Soc. Amer.*, 32:247–274, 1942.

[7] R. Schaback and H. Wendland. Kernel techniques: From machine learning to meshless methods. *Acta Numerica*, 15:543–639, 2006.

[8] E. van den Berg and M. P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction, June 2007. `http://www.cs.ubc.ca/labs/scl/spgl1`.

[9] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.

[10] Günter Wyszecki and W. S. Stiles. *Color Science, Concepts and Methods, Quantitative Data and Formulae.* John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 2. edition, 1982.