

# Making the Best Use of Fifty (or more) Shades of Gray: Intelligent Contrast Optimisation for Image Segmentation in False-Colour Video

Spyridon Bakas, Gordon Hunter  
Digital Imaging Research Centre  
Faculty of Science, Engineering and Computing  
Kingston University  
London, U.K.  
{S.Bakas, G.Hunter}@kingston.ac.uk

Bastien Labbé  
Acquisition and Image Processing  
Télécom Physique Strasbourg  
Strasbourg, France  
Bast.Labbe@gmail.com

**Abstract** — The problem of automatically segmenting images into different components of interest, and then tracking such components through a sequence of frames in a video, is relevant to many applications of video analysis in Intelligent Environments. Several video capture technologies pre-process the image frames, by adding false-colour, in order to improve the contrast for human viewers. However, such false-colouring can make automatic segmentation and object tracking from such videos much more difficult. In this paper, several intelligent dynamic approaches to these problems are proposed and evaluated, in the context of locating, segmenting and tracking salient regions in an example video dataset.

**Keywords** - false colour video analysis, contrast optimisation, segmentation, object tracking

## I. INTRODUCTION

Many applications of Digital Imaging – including those in medicine, remote sensing (e.g. from satellites), and video surveillance using infrared or ultraviolet cameras – give rise to video sequences using “false-colours”. These are often set by the imaging equipment for improving visibility of features to the human eye. However, automated processing of such video sequences is complicated, particularly if the method of allocating false-colour to pixels is not invertible or specified. Segmenting images into regions of interest corresponding to salient features, and tracking such features through a video sequence, is important in many applications – including those mentioned above – relevant to Intelligent Environments. Manual segmentation tends to be extremely time-consuming and prone to human error. Also, use of false-colour can greatly complicate the automation of these processes, since the correspondence between false-colour and true pixel intensities may be unknown, making the detection of contrast between two different salient regions difficult. Recently, automated analysis of false-coloured images of the Amazon rainforest captured by satellites has been shown to give incorrect information about the health of vegetation during periods of drought [2]. Thus, a dynamic, intelligent method for maximising the contrast between such regions in false-coloured video sequences would be of great benefit to the problem of segmenting such images, allowing tracking of salient features.

The current paper forms part of a wider project investigating methods for segmentation and tracking of salient regions through video sequences. The particular aspect of interest in this paper is to assist in the identification of a high contrast frame for the initialisation

of such a region [1]. Thus, research towards the development of such approaches is described and these are evaluated on a pilot dataset with a view to them being applied more widely in video analysis.

The remainder of this paper is structured as follows: In section II, the problem of initialising and then tracking salient regions through a video sequence is outlined, noting the importance of optimising the contrast between regions corresponding to different features. In section III, proposed approaches for transforming false-coloured images to greyscale in a way which maximises this contrast are introduced. Results from applying these approaches to a pilot dataset are presented in section IV. Finally, in section V, we state our conclusions and propose future work.

## II. INITIALISING AND TRACKING SALIENT REGIONS

In many of the video analysis applications mentioned in Section I, initialising and subsequently tracking a salient region corresponding to an object of interest are essential processes. For these to be carried out automatically, it is necessary to have a high degree of contrast between such a region and its surroundings. However, in some cases of both natural and false-coloured images from video sequences, this contrast can be rather poor.

In order to make the differentiation between regions easier for a human eye the manufacturers of imaging equipment often choose a false-colour scheme (e.g. shades of brown, or a full “rainbow” spectrum) instead of greyscale to denote different levels of intensity of some quantity of interest. As noted in Section I, such false-colouring schemes are common in various imaging modalities in many domains.

In [1] the issue of identifying the optimal frame for initialising a region, with a view to subsequently tracking and classifying it through such a video sequence was discussed. However, data acquired from much commercially produced video imaging equipment for various applications in a variety of modalities makes use of false-colour in order to improve the appearance of the different regions to the human eye. As noted previously, use of such false-colour does not give best contrast in the quantitative sense and the method of assigning colours to intensity levels, represented by brightness of image pixels, or vice-versa may not be specified by the manufacturers of the equipment. Thus, in order to facilitate the automation of the initialisation, segmentation, tracking and classification processes it is necessary to first convert the images to greyscale in a manner which will optimise the quantitative contrast. In this study three different approaches are proposed, which would allow this to be

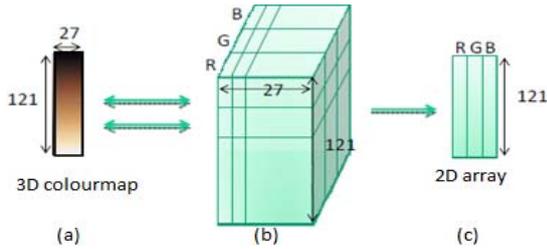


Figure 1. After being extracted from the original image, the 3D colourmap (a) can be processed as a 3D matrix (b), which consists of 121 rows, 27 columns and 3 planes. Averaging across the 27 columns of each row for each of the planes (RGB channels) results in a 2D array of 121 rows and 3 columns (c), each row indicating the RGB values best representing one “brownscale” value used in the colourmap.

carried out in a dynamic intelligent way, optimising the contrast for each individual video sequence. This methodology should be equally applicable to any type of false-coloured video, where the scheme for assigning the false-colours is either unknown or not invertible. For the purposes of this paper the standard deviation (SD) of pixel brightness intensities across an image provides a quantitative measure of contrast, also known as RMS contrast [5]. This measure is more robust to moderate levels of noise than others, such as the Michelson contrast [4] (also known as Visibility), that depend on the full range of intensity values and are hence affected more by noise.

In this study the video frames were false-coloured using 121 shades of brown. The relation of this “brownscale” to actual brightness intensities was set at the start of each video acquisition by the equipment settings specified by the operator, and would not necessarily provide the optimal contrast throughout the whole video sequence. Furthermore, these settings would vary from case to case. Therefore, we seek a method to convert such “brownscale” images to greyscale in a way which optimises the contrast.

### III. PROPOSED APPROACHES

In this section, various approaches are proposed for transforming a false-coloured image to greyscale without losing any useful information from the images. Many programming environments and languages, including Matlab, include a built-in function for converting colour images into greyscale. However, these are based on the USA’s National Television System Committee (NTSC) standard [6], originally devised for converting colour TV images to black and white, and which computes the luminance,  $Y$ , (brightness intensity) using the formula:

$$Y = 0.299*R + 0.587*G + 0.114*B \quad (1)$$

where  $R$ ,  $G$  and  $B$  are the intensity values for the red, green and blue channels, respectively. However, this is based on human perception of colour TV images, noting that the human vision system is most sensitive to green and least sensitive to blue light. Thus, it is not necessarily optimal for converting false-colour images to greyscale. Although some well-established methods exist for contrast enhancement, such as histogram equalisation [3], most of these apply a highly non-linear transformation to the brightness intensity data and are therefore undesirable in some contexts.

Three types of approaches are investigated here. Specifically, the first two make direct use of a “colourmap”, which is included within each video of the dataset used. The first finds the nearest neighbour point in the colourmap (in RGB space) to each pixel in the current image and then uses a rescaling function to map to greyscale within the specified range (i.e. [0, 255]). The second approach allows the mapping from RGB to greyscale to be non-linear and applies regression analysis to fit the best curves. The third uses a statistical approach with eigenanalysis in order to find the direction in RGB space which maximises the variation in brightness intensity and thus give the relative weightings of the colour channels which yield maximum contrast in an image.

#### A. Distance Minimisation in the “RGB Space”

This approach uses a direct mapping of the “brownscale” colour values used in the original image to “greyscale” levels, based on a “colourmap” included in the original image. This colourmap gives information about the dynamic range of shades of brown potentially used in the images, and is determined by the equipment and the settings selected for the video acquisition.

After being extracted from the original image, this 3D colourmap (Fig. 1(a)) can be processed as a 3D matrix, which consists of 121 rows, 27 columns and 3 planes (Fig. 1(b)). Each plane represents a colour channel (i.e.  $R$ ,  $G$  or  $B$ ). The rows represent 121 possible “brown levels”. In principle, the colour should be uniform across the 27 columns of any horizontal row of this colourmap. However, in practice, due to the colourmap having been extracted from the image, there is some variation in RGB values across each row. Thus, before attempting to compute a mapping from the colourmap to greyscale, averaging is needed across each row for each of the channels. This results in a 2D array of 121 rows and 3 columns, each row giving the RGB values best representing each “brownscale” value used in the images (Fig. 1(c)). We therefore wish to map from 121 “brownscale” values to an appropriate set of greyscale values in the range [0, 255] in a way which enable best differentiation between the regions of interest. Since there are only 121 different possible values in the “brownscale” colourmap, they are re-scaled to give a total range [0, 255], but with only 121 distinct values, in the resulting greyscale.

The “distance” (e.g. Euclidean or Manhattan) in RGB space between the values for each pixel in the original image and those of the “nearest” point in the 2D colourmap array is calculated, in order to find the closest correspondence. The equivalent greyscale value, using the scaling described above, is then computed and stored at the same location (i.e. same coordinates) as the original pixel, but in a new image matrix with the same dimensions as the original image.

#### B. Curve Fitting

The second approach is based on approximating the relation between each of the RGB components of pixels in the colourmap and the 121 “brownscale” levels by fitting a quadratic curve to each of the channels.

The RGB components for each pixel within each row of the colourmap were plotted against the row number, which corresponds to the “brownscale” value. Since the colourmap may vary considerably from case to case,

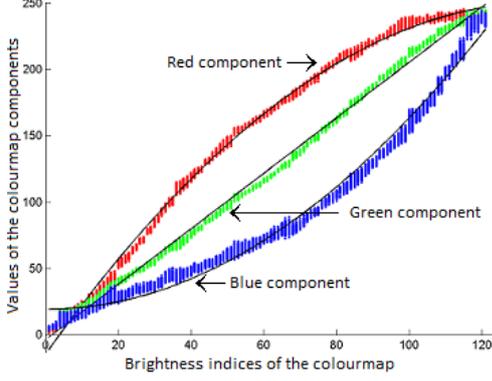


Figure 2. Approximation of each component (R, G, B) of the extracted colourmap by a quadratic curve.

depending on the acquisition machine settings, a different graph was produced for each case. A quadratic (second order polynomial) curve was fitted for each colour channel simultaneously using least squares regression. In Fig. 2, the RGB values of the colourmap of one case are displayed as an example. The  $x$  axis represents the 121 possible levels of the colourmap and the  $y$  axis represents the original pixel R, G or B values in the range of  $[0, 255]$ . For this example, the best fitting curves were:

$$\begin{aligned} R &= -0.0140x^2 + 3.8604x - 15.0318 \\ G &= 0x^2 + 2.0901x - 4.1014 \\ B &= 0.0144x^2 - 0.0053x + 19.0637 \end{aligned} \quad (2)$$

Note that the curve fitting process did not put any explicit constraints on the coefficient values, which explains why two of the intercepts with the vertical axis are negative. Also, in contrast to the NTSC standard RGB to grey conversion [6], our fitted curves appear to be dominated by the red channel, instead of the green channel.

### C. Eigenanalysis of Pixel Statistics

This approach is totally independent of the colourmap and the conversion to greyscale is done by eliminating the chrominance information from the false-coloured image, keeping only the luminance. The Matlab function “rgb2gray” uses weightings of the RGB components consistent with the NTSC standard as given in (1). The idea of the currently proposed approach is to find the optimal weightings tailored to the modality of interest based on the variation of intensities of components across each video sequence individually, instead of adopting the fixed NTSC weightings.

Once again, the RGB components for each image pixel are regarded as coordinates of a point in a 3D space. Over an image frame (or even a whole video sequence) the set of such RGB points form an approximately ellipsoidal cloud. Using the distribution of such points allows the principal axis of the ellipsoid to be calculated using eigenanalysis, in a manner analogous to principal components analysis. The principal axis of the ellipsoid corresponds to the direction in RGB space giving the largest variation in brightness intensity, and hence best contrast in the image, or video.

The relative ratios of the RGB components for this major axis vector correspond to the best weightings of  $R$ ,  $G$  and  $B$  to maximise this contrast. Firstly, the covariance

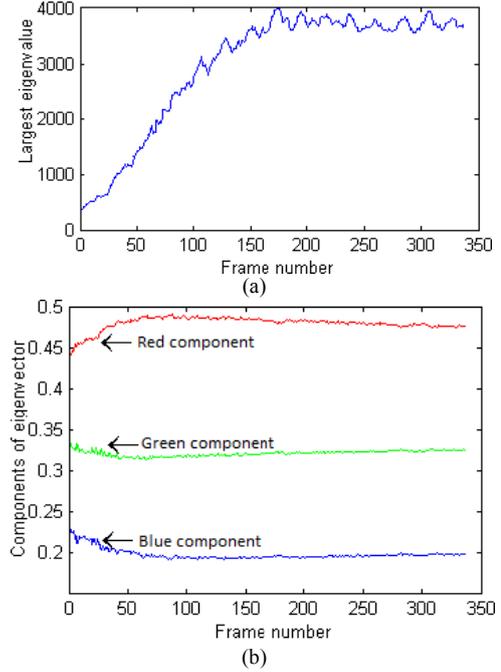


Figure 3. (a) Value of the largest eigenvalue (b) Component values of its eigenvector over the frames of a video

matrix ( $\Sigma$ ) between the three channels ( $R, G, B$ ) across the pixels is calculated:

$$\Sigma = \begin{bmatrix} \sigma_R^2 & \sigma_{RG} & \sigma_{RB} \\ \sigma_{GR} & \sigma_G^2 & \sigma_{GB} \\ \sigma_{BR} & \sigma_{BG} & \sigma_B^2 \end{bmatrix} \quad (8)$$

where  $\sigma_i$  represents the SD over channel  $i$  and  $\sigma_{ij}$  is the covariance between channels  $i$  and  $j$  ( $i, j \in \{R, G, B\}$ ). The eigenvalues of  $\Sigma$  are then computed and the components of the eigenvector that corresponds to the largest eigenvalue and the weightings for the “optimal” greyscaling transformation. Fig. 3(a) shows the evolution of the largest eigenvalue of  $\Sigma$  over the frames of a video sequence in our dataset and Fig. 3(b) shows the normalised components of the corresponding eigenvector. It can be observed from both figures that both the largest eigenvalue and its corresponding eigenvector remain approximately constant after a certain point in the sequence. Note also that, in contrast to the NTSC standard weightings but in a similar manner to the results found by approach B (Curve Fitting) the conversion in our case is dominated by the red channel (R) rather than the green channel (G).

## IV. EXPERIMENTS AND RESULTS

The dataset used for evaluation was a proprietary and sensitive one and therefore precise details cannot be discussed. However, the proposed methods were evaluated over 14 example videos of between 80-500 frames.

### A. Distance Minimisation in the “RGB Space”

The Euclidean and the Manhattan distance were each used to calculate the distance between the RGB triplet of each pixel of the original image and the 2D colourmap array. These gave satisfactory contrast, as measured by the

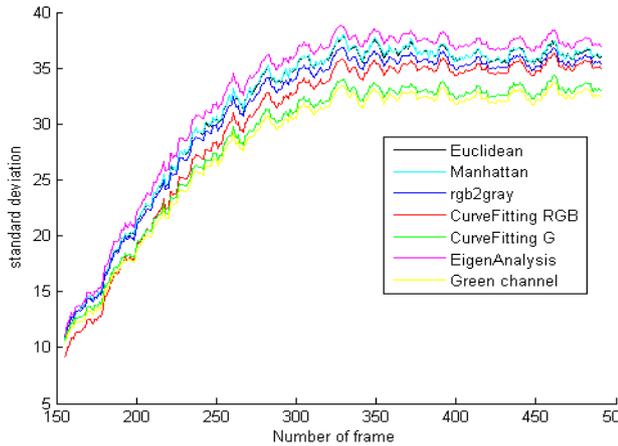


Figure 4: Standard deviation calculated over each frame, by method

SD of pixel brightness intensities, but proved computationally very expensive (Table I & Fig. 4).

### B. Curve Fitting

The best-fitting quadratic functions for each of the RGB channels are given by (2) and shown in Fig. 2. It was observed that the curve for the green channel ( $G$ ) was precisely linear (the coefficient of  $x^2$  being exactly zero). This suggested that use of the green channel alone might provide an appropriate method for conversion of RGB images to greyscale. Once again the contrast achieved was reasonable (Fig. 4), but the computational time required was high for both curve fitting alternatives (Table I).

### C. Eigenanalysis of Pixel Statistics

The covariance matrix of the RGB values for the pixels in each frame of a video was computed, along with the largest eigenvalue and corresponding eigenvector of the matrix. This was repeated for each frame of every video, and the evolution of the eigenvalue and eigenvector monitored over the frame sequence. It was observed that both eigenvalue and eigenvector stabilised after an initial transient period (Fig. 3). The relative component values of the eigenvector give the weightings of the RGB components in the linear combination which yields the largest variation, and this can be considered to be the mapping which maximises the contrast according to our definition. Thus, these weightings are chosen for the mapping from the false colour values to greyscale. For the example shown in Fig 3, this transformation is given by:

$$Y = 0.48*R + 0.32*G + 0.20*B \quad (9)$$

As noted previously, and similarly to approach B, the largest weighting is for the red channel, as opposed to the NTSC standard (1) where the green channel gives the largest contribution.

This approach gives the best contrast (as measured by the SD over the images) of all methods investigated, and was also substantially faster than the distance minimization or curve fitting approaches. However, for comparison purposes, the performance of the Matlab function “rgb2gray” and of using the values of the green channel only with respect to our data were investigated. These gave slightly lower contrast (reducing the SD by about 5% to

10%) but were faster than any of the other methods. In particular, the computational time required using the green channel only was significantly lower (Table I).

TABLE I. COMPUTATIONAL TIME PER FRAME (IN MILLISECONDS)

	Different Methods						
	1	2	3	4	5	6	7
SD	13.6	15.1	1830	1929	0.173	2.1	0.0629
Min	2689	2458	1333	1156	4.7	116.7	0.6487
Max	2733	2511	7396	7905	5.4	125.0	0.8929
Mean	2704	2470	5248	5249	4.8	120.0	0.7089
Median	2699	2464	5720	5573	4.7	119.3	0.6884

Index of methods: 1-Euclidean, 2-Manhattan, 3-Curve Fitting RGB, 4-Curve Fitting G only, 5-rgb2gray, 6-Eigenanalysis, 7-Green Channel only

## V. CONCLUSIONS

The various methods implemented and evaluated gave rather similar results, in terms of the contrast (measured by the SD of brightness intensity across each frame of the acquired video sequences). Although the eigenanalysis approach gives slightly better contrast than any of the other methods it is not as fast as use of just the green channel value for each pixel. The latter only reduces the SD (and hence our measure of contrast) by about 10% relative to the eigenanalysis whilst giving a speed-up factor averaging about times 160 over the provided dataset compared with the time required for the eigenanalysis. Use of the green channel alone gave the fastest performance of all the methods. Thus, the ranking of the various methods will depend on the application taking into account whether speed or achieving the best contrast is considered the most important factor.

The proposed approaches should be appropriate to false-coloured sequences from any imaging modality relevant to various applications for intelligent environments. For example, in an intelligent building, or smart agricultural/horticultural application, analysis of sequences of false-coloured images from a weather satellite feed could be used to predict rainfall, windspeed and outside temperature changes, which can subsequently be used for control of the local conditions within the environment.

## REFERENCES

- [1] S. Bakas, G. Hunter, C. Thiebaud and D. Makris, “Spot the Best Frame: Towards Intelligent Automated Selection of the Optimal Frame for Initialisation of Focal Liver Lesion Candidates in Contrast-Enhanced Ultrasound Video Sequences”, 9th International Conference on Intelligent Environments (IE13), pp. 196–203, 16-17 July 2013, doi:10.1109/IE.2013.20.
- [2] BBC WALES, Website, <http://www.bbc.co.uk/news/uk-wales-26262986>, 19 Feb.2014, Accessed:24 Feb. 2014.
- [3] R. C. Gonzalez and R. C. Woods, “Digital Image Processing”, Prentice Hall, 3rd Edition, 2007.
- [4] A. Michelson, “Studies in Optics”, University of Chicago Press, 1927.
- [5] E. Peli, “Contrast in Complex Images”, Journal of the Optical Society of America A, 7(10), pp. 2032-2040, 1990.
- [6] International Telecommunications Union (ITU), Recommendation ITU-R BT.470-6\*, Conventional Television Systems, 1998 ([http://www.itu.int/dms\\_pubrec/itu-r/rec/bt/R-REC-BT.470-6-199811-S!!PDF-E.pdf](http://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.470-6-199811-S!!PDF-E.pdf))