A colour-map plugin for the open source, java based, image processing package, ImageJ

Keagan Moodley, Hugh Murrell*

Department of Computer Science, University of Natal, Durban 40412, South Africa

Received 2 July 2003; received in revised form 19 February 2004; accepted 11 March 2004

Abstract

We present an interactive approach to the pseudo-colouring of greyscale images. We implement the technique by computing mappings from a three-dimensional (3D) colour space to a one-dimensional greyscale space (i.e. $R^3$ to $R$). To compute our maps, we employ both linear and nonlinear interpolation in 3D colour space. We validate our work by applying our maps to greyscale images resulting in significant image enhancement. Applications include space imagery, geological topographies, medical scans and many more. Our tool is coded as a Java plug-in for the open source image processing package, ImageJ.

Keywords: Greyscale; Image enhancement; Pseudo-colouring; Interpolation; Colour models

1. Introduction

Image enhancement is probably the most researched area in image processing. Enhancements such as histogram equalization are usually applied to greyscale images so as to reveal detail.

After a greyscale image has been enhanced a final improvement is usually a pseudo-colouring of the image. This involves the assignment of single colours to specific monochrome intensities present in the greyscale image. Practitioners are usually left to their own devices when generating such pseudo-colouring maps. Colours are selected in a manner that enhances various densities of greys according to their relevance to the application in question. For example, infected blood cells in a microscope scan could be characterized by a density of grey that is not noticeable to the human eye in a greyscale image. Pseudo-colouring can reveal the infection.

This introduction of colour to greyscale images can certainly make image analysis easier. The human eye can distinguish thousands of colour shades, as opposed to about two-dozen shades of grey (Burger and Gillies, 1989). Therefore colour image processing can be considered as a form of image enhancement. We focus specifically on pseudo-colouring.

We present a technique of pseudo-colouring that is based on interpolations. Colour-maps or look-up-tables (LUTs) are created by interpolating between selected colours within specific three-dimensional (3D) colour spaces. Appropriate maps can be generated and applied to images resulting in enhancements. Interpolation produces colour-maps that consist of a diverse range of colours, of continuously varying hues, tones and shades.

This technique is standard on many image analysis systems (GEOMATICA, ERDAS, ENVI and ER-MAPPER) and GIS systems (ArcGIS and MapInfo). However our implementation offers Bezier interpolation in colour space and this may be a new approach to pseudo-colouring.

*Corresponding author. Fax: +27-31-260-2280.
E-mail address: murrellh@nu.ac.za (H. Murrell).

© 2004 Elsevier Ltd. All rights reserved.

doi:10.1016/j.cageo.2004.03.017
We utilize the theoretical discussions on colour models in (Burger and Gillies, 1989) to model various colour spaces and apply interpolation methods in these spaces. Incorporating interpolation lightens the workload for the user and thus results in a significant reduction in the time spent on creating colour-maps.

This paper provides an overview of pseudo-colouring. Section 2 provides a background to pseudo-colouring and to colour models. Section 3 discusses methodology and covers our techniques and approaches. Section 4 illustrates the application of our methods. Section 5 discusses the implications of our research.

2. Background

Colour has been used as a descriptor in a variety of applications and it has proven to be a powerful concept in image processing. It can greatly enhance detail and improve ones perception of images. Colour manipulation in image processing can make significant changes as to how we interpret images and the ease with which we are able to notice irregularities. By bringing out the detail in images, tasks such as object detection and identification are simplified. Colour can certainly make visible, areas of interest that are not clear in greyscale images.

2.1. Pseudo-colouring

Pseudo-colouring involves the assignment of colours to a range of greyscales. It is in ones best interest to manipulate these colours to be of varying shades and tones so as to bring out the appropriate detail in images. This point brings to light the problem tackled in this paper—the task of selecting colours with which to perform a mapping.

8-bit greyscale images can contain a maximum of 256 greyscales ranging from black (0) to white (255). It is therefore necessary to have 256 colours in order to create a colour-map.

Consider the RGB colour cube for argument purposes. There are a total of 16.7 million (≈256^3) colours present in the cube, since each component or colour gun can produce a maximum of 256 shades. It is with no doubt impractical to manually choose colours from such a wide array of possibilities.

The solution to colour selection (for colour-maps) is, therefore, to design a system whereby 256 colours from the 16.7 million available can be generated automatically but in a manner that relates them to the application at hand.

2.2. Colour models

A colour model is a specification of colours in a 3D co-ordinate system. It usually satisfies some specific standard and is generalised by a geometric solid. Each solid representing a model is a visible subspace in the co-ordinate system. Each colour that is defined in the subspace is represented as a single point. Essentially, the main purpose of colour models is to facilitate colour specifications within various colour gamuts.

Many models have been developed to represent the range of visible colours, each with properties that are appropriate for particular applications. Some of them, although being able to display spectral colours properly, still have limitations. Some colour models that have been successful in illustrating all or most of visible colours are the RGB (red, green, blue) cube, the HSI (hue, saturation, intensity) triangular diamond, the HSV (hue, saturation, value) hexagonal pyramid, and the HLS (hue, lightness, saturation) hexagonal diamond. Although not essential for this paper, these models do form an integral part of the research carried out—for more information refer to (Burger and Gillies, 1989; Foley et al., 1990; Gonzalez and Wood, 1992).

3. Methodology

3.1. Software

The methods to be described in this paper were written as part of a comprehensive plug-in for an image processing package (ImageJ). The package was developed and is maintained by Wayne Rasband (wayne@csd.ornl.gov), at the research services branch of the National Institute of Mental Health that is situated in Bethesda, Maryland in the USA.

ImageJ was developed using the Java language, making it a platform independent program that will run on any operating system with at least the Java 1.1 virtual machine installed. As a result of its open architecture design, its functionality can easily be extended via Java plug-ins. This allows for the development and incorporation of user-written plug-ins that are customized to solve specific problems.

Apart from being able to perform a variety of standard image processing tasks, ImageJ can handle 8-bit, 16-bit and 32-bit images. ImageJ can read from and write to a number of image formats and it is also multithreaded.

The package including its source is freely available for download from (ImageJ). The site provides an overview of ImageJ with documented help and detailed installation instructions. Also available for download is an extensive range of plug-ins written for either processing or analysis problems.
Our package will be submitted to the ImageJ website to become part of their standard distribution. In the meantime the interested reader can download our plug-in source code from the IAMG website where a READ.ME file explains how to integrate them into ImageJ.

3.2. Colour selection

All of the colour models mentioned in 2.2 were implemented in a GUI via the plug-in. We also include an alternate version of the HSV colour model which we denote as iHSV (Inverted HSV). Slices of the models can be traversed by using a slider and are updated in real-time. This responsiveness allows the user to immediately see the colours available in the different slices.

The colour cube in Fig. 1 shows a 3D view of the RGB colour space. In the plug-in, this colour space has been modelled in 2D, displaying slices that are parallel to any but only one of the above three colour planes as illustrated in Fig. 2. A slice is perpendicular to the component axis that governs the value of that component throughout that slice, i.e. if the slice is perpendicular to the ‘red axis’ then traversing through the axis from 0 to 255 will vary the value of the red component in the slices accordingly. Consequently, the position of the slice on the axis will determine the value of the corresponding component throughout the slice. The axis can be set according to the preference of the user.

The colour models implemented in our tool are illustrated in Figs. 2–6. Slices in each of the models are taken perpendicular to the main axis of the solid that describes that model.

The HSV model (Fig. 4), corresponds to the projection of the RGB colour cube as seen along its main diagonal from white to black. Once inverted (essentially changing the viewpoint of the diagonal—from black to white) one has an entirely different perspective of the colours contained as shown in Fig. 5. The models were implemented in a manner that illustrates their differences clearly making visible their pros and cons.

Once a colour space has been chosen for an interpolation, colours can be selected to serve as key or pivot points. The colours, of course, are chosen to suit the intended application. As one might suspect, users should be skilled in their field to make appropriate selections.
3.3. Interpolation in colour space

Interpolation is carried out between the pivot points and can be either linear or nonlinear. Pivot colours are separated into their respective three components depending on the preferred colour model selected for interpolation. For example, if the interpolation is to be done within the HSV colour model then the 3 components will refer to hue, saturation and value. Three component arrays of length 256 are used to store the pivots and the final interpolated values of each component.

Interpolation is applied to each component individually so as to fill in the missing component values and attain the desired range. On completion of the three interpolations, the values in the corresponding positions in the component arrays are combined appropriately to form a colour map.

Piecewise linear interpolation and nonlinear interpolation along Bézier curves were investigated and will be discussed in further detail towards the latter part of this section. After interpolation, the colour-map is generated. The maps are composed of a range of 256 RGB colours. These are mapped onto the 256 shades of grey, in a greyscale image. This is illustrated in Fig. 7. The process essentially performs a mapping from $R^3$ (colour space) to $R$ (greyscale space). The inverse of the colour-map is used to pseudo-colour the greyscale image.

Before continuing with discussions on the 2 types of interpolations experimented with, we briefly state some important advantages of using interpolation:

- Interpolation utilizes processing power to generate maps thereby increasing efficiency.
- Values that are not amenable to user specification can end up in the colour-map.
Interpolation produces colours that are similar in properties to those selected—keeping hues, saturation, intensities, shades and tones similar throughout the map.

Pivot colours are automatically blended irrespective of their differing properties.

3.4. Linear interpolation

Linear interpolation is performed on each component and the interpolated points are combined to form the colour map. Hence, only two key points, \( P_0 \) and \( P_1 \) are used to generate a map of size 256. The key points are vectors in 3D space. Fig. 8 illustrates a typical linearly interpolated colour-map.

To compute the interpolated values of all components, the following algorithm is used:

\[
\text{ColourMap}[i] = \text{ConvertToRGB}(P)
\]

where \( \text{ColourMap} \) is an array containing the interpolated colours and \( \text{ConvertToRGB}(P) \) represents a function that merely combines the interpolated positions (according to the definition of the appropriate colour space) to form the RGB colour they define. This is the RGB standard that is required for display.

3.5. Non-linear interpolation

Bézier curves were the focus for nonlinear interpolation and are based on the theoretical methods and discussions in (Ammeraal, 1998; Barsky, 1988; Bartels et al., 1987). These curves are third degree polynomials requiring four pivot points. They are usually referred to as cubic Bézier curves and can be defined as the sum of 4 terms as follows:

\[
B(t) = (1 - t)^3 P_0 + 3t(1 - t)^2 P_1 + 3t^2(1 - t) P_2 + t^3 P_3,
\]

where \( P_0 \) and \( P_3 \) are the end-points of the curve in colour space and \( P_0 \) and \( P_3 \) are control points.

Eq. (1) can be modified to

\[
B(t) = (-P_0 + 3P_1 - 3P_2 + P_3)t^3 + 3(P_0 - 2P_1 + P_2)t^2 - 3(P_0 + P_1)t + P_0
\]

which can be further simplified by re-labelling to

\[
B(t) = C_3 t^3 + C_2 t^2 - C_1 t + C_0 = ((C_3 t + C_2) t + C_1) t + C_0.
\]

B(t) can be thought of as a vector valued function of time, where \( t \in [0, 1] \). As we move at constant velocity from the beginning \( (P_0) \) to the end \( (P_3) \), starting at \( t = 0 \) and finishing at \( t = 1 \), we traverse this parametric curve in 3D-space. The Bézier curve can be either 2 or 3-dimensional depending on the choice of control points. The algorithm to implement Bézier interpolation would be as follows:

\[
dt = 1/225, \quad C_3 = -P_0 + 3(P_1 - P_2) + P_3, \quad C_2 = 3(P_0 - (2P_1 - P_2) + P_3, \quad C_1 = 3(P_1 - P_0), \quad C_0 = -P_0.
\]

\[
\text{for } i = [0, ..., 225] \{ \quad t = i \times dt \quad P = ((C_3 \times t + C_2) \times t + C_1) \times t + C_0 \quad \text{ColourMap} [i] = \text{ConvertToRGB}(P),
\]

where \( \text{ColourMap} \) and \( \text{ConvertToRGB}(P) \) are as described earlier. In the above algorithm \( P_i, C_i \) and \( P \) are vectors in 3D-space.

Fig. 9 illustrates a typical Bézier interpolated colour-map. We note that due to the convex hull property of Bézier curves, such curves will always lie within the colour spaces.

3.6. Piecewise maps

Piecewise maps are the result of interpolation among more than two colour selections and can be classified as being either continuous or discontinuous.

Our tool includes a widget, MapCombiner, for producing new colour-maps from existing maps. Two maps are compressed into a single map by the widget. This is accomplished by simply selecting two previously
created colour-maps and specifying a cross-over point in the widget. The cross-over point (cp) determines the ratio of the colours in the two selected colour-maps that will be present in the new piecewise map. As a result, the first map of the pair is compressed into the interval [0,cp] and the second into the interval [cp+1255]. The effects on the image in use can be viewed in real-time if one prefers.

3.7. The shutter effect

The MapCombiner widget has additional features to aid one in producing useful colour-maps. Firstly, one can focus on a specific region in a colour-map. This is achieved by using what we refer to as left and right shutters. Using the appropriate sliders will cause these to close-in on the displayed colour-map from the edges inward, filling the map with the colour black. As a result, a visibly range or window is produced and is defined by the portion of the map not covered by the shutters as illustrated in Fig. 11.

Secondly, the window created by the shutters can also be moved within the range [0,255], thus exposing and displaying different portions of the colour-map. Again, effects on the image in use can be viewed in real-time if one prefers.

4. Results

To validate the techniques described, the implemented algorithms were used to produce colour-maps that were applied to a variety of greyscale images. Original greyscale and the pseudo-coloured slides are shown side by side. Histograms of the greyscales present in the images are shown with the respective colour-maps. The dominant colours in the map can be identified and the mapping transformation can be seen.

4.1. MRI scan

Fig. 12 shows a collection of slides from an MRI-scan. As a result of the nature of these scans, the original map
Fig. 12. A collection of MRI scans before and after application of pseudo-colouring. Histograms show colours that are mapped to specific grey scales in each case.
is actually a negative. Therefore, the map shown depicting the original greyscales is reversed. The 0 and 255 markings are guides, showing the original distribution of greys in the greyscale colour-map of the nonnegative image.

Two colour-maps were combined using our colour-map combiner. The first was an eight pivot map created using piecewise-continuous linear interpolation in the RGB colour space and the second was a standard greyscale map. The final colour-map was produced using a 90:10 ratio of the respective maps.

The colour-map emphasizes the cavities, brain tissue and other soft tissue contained in the skull.

4.2. Microscopic cell image

Fig. 13 shows a microscope image of blood cells infected with HIV. The applied colour map was produced using piecewise-continuous linear interpolation in the HSI colour model with three pivots. The map was then altered by applying a right shutter to remove unnecessary intensities in the image.

The effects of the colour-map, highlights cell contours and membranes. In certain cells, the fluorescent green shows clearly the HIV budding from cell membranes.

4.3. Geological features

Fig. 14 shows a section of the sea-bed before and after pseudo-colouring. To produce the map, linear interpolation was performed in the HSI colour space between two pivots. This colour-map is now employed by the Marine Geo-Science Unit at Natal University to colour all their sea-bed images.

Enhancements can be observed when comparing the images. The histograms show precisely which colours where used in the mapping and the grey densities to which they were mapped.

4.4. Space imagery

Fig. 15 shows a satellite image, from a weather station, of the earth before and after pseudo-colouring.

The original image was taken from (EUMETSAT). Two colour-maps were combined. The first was a standard greyscale map and the second was a linearly interpolated map in the HSI colour space. The final colour-map was produced using a 40:60 ratio of the respective maps. Thereafter, a right shutter was applied to the map, mapping the colour black to all grey intensities above, approximately, 235.

![Fig. 13. Microscope images of blood cells infected with HIV before and after pseudo-colouring.](image)
Again, enhancements can be observed when comparing the images. An image of this type will be particularly useful when one is concerned with rainfalls, global temperatures or to forecast heavy winds.

5. Conclusions and future work

Pseudo-colouring does form an integral part of image enhancement. As can be seen, hidden features become easily distinguishable after pseudo-colouring. The colour-maps used, although seemingly effective, merely illustrate the possibilities of our techniques and can vary according to the requirements and preferences of the user.

Any field that only has access to greyscale images for conducting analysis can benefit from the work carried out here. Among the many fields, are electron microscopy, oceanography, astronomy (satellite imagery) and medical sciences.

In the medical field, for example, scanning technologies such as the MRI-scan and the CT-scan that still employ greyscale imaging can be supplemented through pseudo-colouring. Medical practitioners could use our tool to produce a library of colour-maps—each one designed for a specific detection criteria.

Future work in this field should include the implementation and application of other interpolation paths within colour spaces. Analysis of the results could be undertaken and comparisons can be made with those techniques already in effect.
Acknowledgements

The financial assistance of the Department of Labour towards this research is hereby acknowledged. We acknowledge the assistance of Dr. P. Moodley from the Department of Virology, University of Natal Medical School. We thank the Marine GeoScience Unit from the University of Natal who first brought the problem to our attention.

References


Fig. 15. Satellite Images of earth before and after pseudo-colouring.