Three Dimensional Colour Pickers

Yingxin Wu **Masahiro Takatsuka**

School of Information Technologies
The University of Sydney
National ICT Australia[#]

chwu@it.usyd.edu.au

masa@vislab.net

Abstract

Colours are three dimensional data. However, all commercial colour pickers project three-dimensional (3-D) colour spaces into two dimensional (2-D) palettes. The projection often results in a set of colours perceptually inappropriate to describe the data they present. This paper initially compares several popular 2-D colour pickers and then describes some novel colour pickers which allow colour selection in 3-D RGB, HSV and Munsell colour spaces. The colour palette is a slice cut off by a translucent plane from 3-D colour solids. The position of the plane tells the user where they are in the colour space, which we believe can facilitate the colour picking process. We also develop a mechanism to generate gradually changing colours between several pre-selected colours through interpolation. Users are allowed to change the interpolation path freely in the colour palette.

Keywords: colour picker, 3-Dimensional, interactive

1 Introduction

1.1 Colour Models and Colour Spaces

Colour is a visual sensation caused by an observed light upon the retina of the eye. It not only plays an important part in our everyday life, but is also very important to applications such as publishing, information visualization, cartography etc. A colour model specifies the basic components used to describe a colour. Although different colour models measure the colours in different ways, three parameters are necessary to specify a colour. This means that colour spaces are three dimensional. The following is a brief introduction to several popular colour models and their corresponding colour spaces.

The RGB model is commonly used in monitors and computer graphics. It uses the mixture of three primary colours: red, green and blue, to produce all the colours on screen. The values of red, green and blue primaries range from 0.0 to 1.0 which means that this colour space is a unit cube as shown in the left of Figure 1. Not all of the human observable colours can be produced by mixing red, green and blue light, so the RGB colour space contains only a subset of the colours that can be seen by the human eye.

The Hue, Saturation and Value (HSV) colour model is a non-linear transform from the RGB model. The algorithm to transform a colour from HSV to RGB or vice versa can be found in (Smith 1978). HSV arranges the Hue in a circle of 360 degrees while Saturation runs from neutral grey to full saturated colour with a range of (0.0,1.0). The range of Value is also 0.0 to 1.0, where 0.0 means black and 1.0 means white. The colour gamut of HSV is the same as RGB.

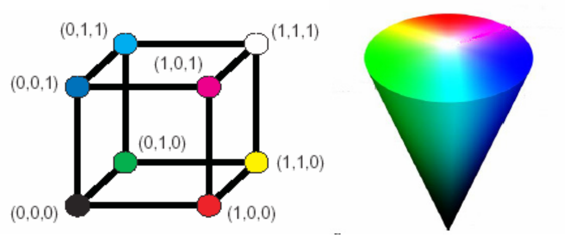


Figure 1 The RGB colour space (left) and the HSV colour space (right)

The Munsell colour model was created by an American artist, Albert Henry Munsell, in 1915. Each colour in the space has 3 attributes: Hue, Value and Chroma. The neutral colours are placed along the vertical Value axis with black at the bottom, white at the top and all grey shades in between. Hues are arranged in a circle while the Chroma scale is perpendicular to the Value axis pointing outwards. Each colour in the colour model is called a colour chip and the Munsell colour space looks like an asymmetric tree. This colour model is commonly used in reflectance models and industrial printing.

Another colour space, the CIELab, was created by the *Commission Internationale de l'Éclairage* (CIE) in 1976 based on the Opponent Theory (Lilley et. al). Three orthogonal axes are chosen to represent the three opposite channels which form colour vision. The central vertical axis represents the black/white (dark/bright) channel with a range from 0 to 100. Two other horizontal axes *a-a'* and *b-b'* represent red/green and blue/yellow channels respectively. Both the Munsell and CIELab colour spaces are designed to accommodate all the human observable colours.

[#] National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council

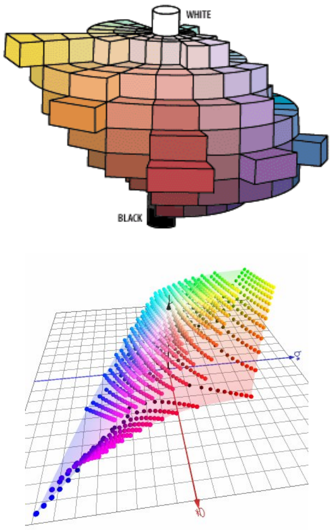


Figure 2 The Munsell Colour Space (top) and the CIELab colour space[1] (bottom)

1.2 Perceptually Uniform Colour Spaces

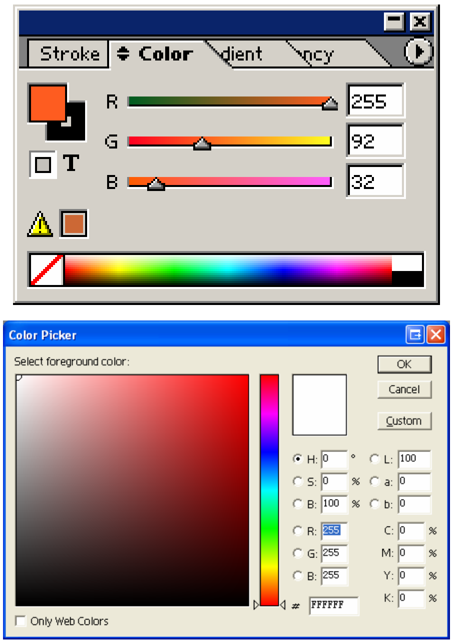
A uniform colour space is one where equal physical distances in the colour space represent visually the same difference in colour. The RGB and HSV colour spaces are non-perceptually uniform while the Munsell and CIELab colour spaces are perceptually uniform.

Although perceptually uniform colour spaces are currently not popular in computer graphics, they are very helpful in cartography and scientific visualization and their importance has been recognized (Meyer et. al. 1980, Slater 1989). As pointed out by Cynthia Brew (1999): "when people read colour graphs and other data graphics, they are working with perceptual dimensions of colour, even though you may have specified the colours using mixture systems like RGB". For example, in drawing a map describing pollution levels in different areas darker colours might be used to represent higher pollution values, progressing through to brighter colours as pollution levels decrease. This is called a *sequential colour scheme.* Using RGB colour models, it is difficult to produce such a systematic colour scheme since there is no concept of brightness. Although HSV does have a parameter (Value) for brightness, it is not perceptually uniform which mean equal variance in Value doesn't result in equal variance in brightness. On the other hand, it would be easy to produce such a sequential scheme using the Munsell or the CIELab colour space. We only need to select colours so that they are equally space along the brightness axis.

2 Existing Colour Pickers

2.1 Commercial Colour Pickers

A lot of research has been devoted to designing an efficient colour picking interface (Bauersfeld and Slater 1999, Montag, E. D., 1999). Currently all commercial colour pickers are at most two dimensional --- for example, the RGB and CIELab colour picker provided by Adobe Photoshop. The RGB picker allows the user to control the red, green and blue components of a colour by adjusting three slider bars. This is called a "3*1D" interface. The CIELab picker organizes two of the parameters in a plane and uses only one slider bar to modify the third dimension, which is called a "2-D+1-D" interface.



R 255
G 92
B 32

H: 0 °
S: 0 %
B: 100 %
R: 255
G: 255
B: 255
L: 100
a: 0
b: 0
C: 0 %
M: 0 %
Y: 0 %
K: 0 %
# FFFFFF

Figure 3 The RGB(top) and the CIELab (bottom) Colour Picker in Adobe Photoshop

There are two disadvantages to such colour pickers: first, it is difficult to think of colours in terms of their red, green and blue components and the "3*1D" interface does not supply any information about how the colours are related in the colour space. Most often, users are confused as to how to adjust the slider bars in order to obtain the colours they want. Secondly, by projecting the 3-D colour space into a 2-D palette, the relationships between colours may be distorted. Take the CIELab colour picker for example. As mentioned earlier, the CIELab colour space is perceptually uniform. From Figure 2 we can see that this colour space is irregularly shaped. However, in order to fit in the square colour gamut window, the "2-D+1-D" colour picker has to stretch the *a-b* dimension which means that the colours displayed on the 2-D palette are no longer perceptually uniform.

2.2 Colour Pickers for Perceptually Uniform Colour Spaces

There exist only a few colour pickers based on uniform colour spaces. An implementation of a Munsell colour picker is provided by several researchers at Penn State University (MacEachren et. al. 2001). In this implementation colour chips with the same Chroma are placed in a 2-D palette. A slider bar is used to adjust the Chroma value, which changes the colour chips to be displayed on the palette. Besides picking individual colours, a mechanism is provided to generate the

[1] Couleur.org, http://www.couleur.org/index.php,

intermediate colours between two or more pre-selected colours by interpolation. For example, in Figure 4, two colours--- green and purple---are selected. The in-between colours are interpolated and are displayed in the lower colour bar. The only drawing back is that it is impossible to perform the colour interpolation between two colour chips with different Chroma.

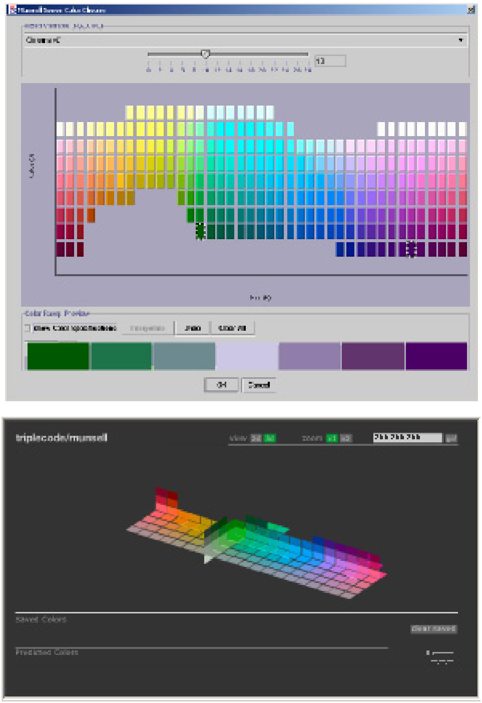


Figure 4 The Munsell Colour Picker by Penn State University (top) and the Triplecode Munsell Palette[2] (bottom)

David Young of the MIT media lab suggests another Munsell colour picker called “Triplecode Munsell Palette”. Unlike the Penn State Munsell colour picker, it arranges the colour chips in 3-D. The Hue circle is broken down and spread horizontally. Chroma is arranged along the *z*-axis (pointing inward) while the *y*-axis (pointing upward) represents the Value. The interface also provides a mechanism to predict the intermediate colours between selected chips. This colour picker is better than the Penn State picker in that it gives users a better sense of how the colours are related. However, the arrangement causes some chips to be occluded by others, and the user cannot directly scroll through the colour space: that is, they manually enter the RGB values of the focus colour.

2.3 Colour Interpolation

Colour interpolation is very useful in cartography and scientific visualization. For example, when drawing a relief map, the altitude information is often represented by colours. The colours of the sea should gradually change from dark blue to sky blue while mountainous terrain should be varied from brown to yellow. Picking colours one by one is quite tedious. A better way is to allow the user to select several separate colours such as brown and yellow and let the colour interface produce the intermediate ones. However, current colour pickers offer very limited support for this type of colour interpolation. The interface provided by Adobe Photoshop does not show how the colours are generated between pre-selected colours and the user is not able to adjust the way colours are interpolated.

3 3-D Colour Pickers

From the above discussion, we can see that there are several drawbacks to existing colour-picking interfaces. We attempt to solve these problems by designing a new 3-D colour picker as follows:

a) We preserve the relationship between colours by fixing their relative position in the colour space. This method also shows the user which part of the colour space they are working with, which increases the interface’s visual feedback. In 1999, Douglas and Kirkpatrick (1999) performed a colour matching experiment with two colour models (RGB and HSV) and two interfaces with different feedback levels. The result showed that there was no significant efficiency difference in using different colour models. But the interface with more visual feedback improved the colour matching accuracy by 26%. The authors concluded that increased visual feedback will be valuable and they hypothesized that human performance with 3-D is better than 2-D+1-D interface, which is better than 3*1D interface.

b) In order to navigate through the 3-D colour space, we use a translucent plane to cut a slice out of the 3-D colour solid. The slice is then shown in another window, serving as a colour palette. The user can rotate or translate the plane by mouse. The shape and colour of the palette will change as the position of the plane changes.

c) Besides picking individual colours, we also provide a mechanism to allow colour interpolation. The interface not only supports insertion and deletion of the control colours (the colours to be interpolated), but also shows the interpolation path. Currently the colours are interpolated using the “Kochanek-Bartels Cubic Spline” (Kochanek and Bartel 1984) method. The shape of this curve is controlled by the positions of anchor points and three other parameters—Tension, Continuity and Bias. In our colour picking interface, the colours selected by the user are considered to be the anchor points and the intermediate colours are sampled along the spline. A user can change the positions of the anchor points using a mouse, resulting in a change of the interpolation path. Anchor points can be added or removed by pressing the corresponding button in the control panel. Using a spline curve greatly increases the degree of freedom in colour interpolation. By adjusting the Tension, Continuity and Bias, the spline can be varied from a straight line to a very relaxed curve, thus changing the interpolation path.

[2] Triplecode-Munsell Palette, http://www.triplecode.com/munsell/

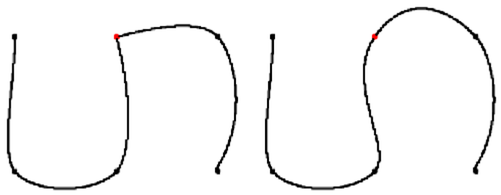


Figure 5 Two Kochanek-Bartels Cubic Splines with the same anchor points but different parameters.

3.1 3-D RGB Colour Picker

We have developed three colour pickers for the RGB, HSV and Munsell colour spaces respectively. They are all implemented using Java and Java3D.

3.1.1 Colour Space Visualization

The RGB colour space is visualized as a unit cube with the black colour point placed at the origin (0, 0, 0). The values of the primary colours (red, green and blue) are indicated by the x, y and z axes respectively. Under such an arrangement for a colour C(r, g, b) in the RGB colour cube, the x, y and z coordinates of C also equal r, g, and b. Theoretically, in order to render the colour on the surface of the unit cube, the surface squares need to be broken down into many sub-squares with each sub-square indicating a single colour. The colour of a sub-square can be specified by the x, y, z coordinates of the centroid of the sub-square. Here we make use of a fact that, given a polygon, Java3D will first do the triangulation and then, for each triangle, if the colours of the vertices are not the same, the graphics card will automatically interpolate the colour linearly inside. Therefore the colours inside each of the triangles will vary from point to point linearly in R, G, B values which is just the effect we desired. The approach we have taken is to assign the 4 vertices of the square with the correct colours. The speed of rendering using hardware (the graphics card) is much faster than software rendering.

3.1.2 Colour Palette

A simple control interface has been developed to manipulate the colour picker. To pick a colour, the user needs to press the button with the red ball icon in the control panel. A small grey sphere then appears at the centre of the palette. The colour under the sphere is selected and shown in the colour bar which is above the palette. The user can select another colour by change the position of the sphere using the mouse. If more than one sphere is created, the colour picker would automatically generate the colours in-between. Currently 64 colours are sampled evenly along the spline. The button with the cross icon is used to delete the selected control colours. The RGB values of the colour can be saved to a text file by pressing the save icon.

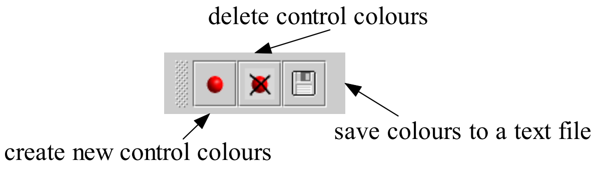


Figure 6 The control panel

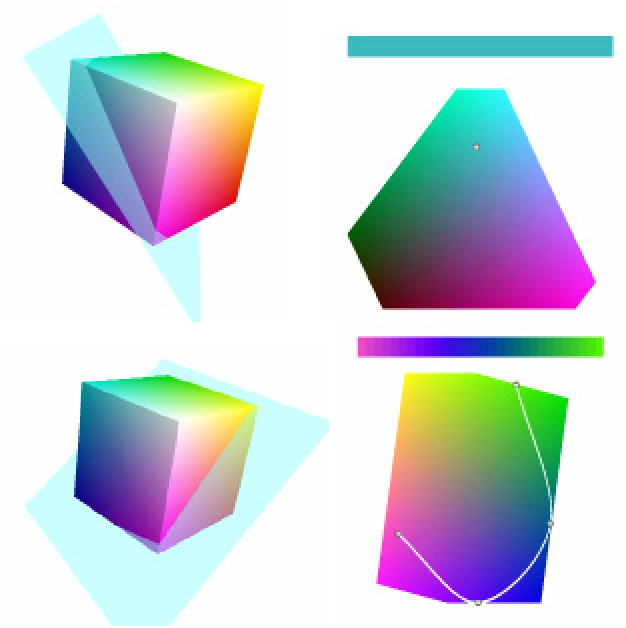


Figure 7 The RGB Colour Cube and colour palettes when the translucent plane is in different positions

3.1.3 Handling Boundary Conditions

Positions of spheres are limited inside the palette. However, a spline curve can still go outside the palette if the spheres are too close to the boundaries. In this case, the program will sample the colours along the boundary until the curve returns to the interior of the palette. The curve must be kept inside so that it represents the actual path where the intermediate colours exist. The solution is to clip the outside parts and replace them with the palette's boundary lines.

The first step is to find the points where the spline intersects the boundaries. One method of intersection testing treats the problem as a system of equations. Since the spline is a cubic equation, using numerical methods like Binary Search or Newton's Method to compute the points of intersection would be quite costly, especially when the control points are being constantly moved by the user. To make things worse, the shape of the palette is not constant. It is changing as the location and orientation of the cutting plane changes. Hence the number of its edges may also increase or decrease. It is difficult to compute the intersections when the number of edges is not constant. To simplify the whole process and speed up the program, we approximate each segment of the spline (the curve between two anchor points) with 20 line segments and store their vertices in an array. Thus to find out which part of the curve goes outside we simply test which vertices in the array are located outside of the boundaries.

We chose the Winding Number algorithm (Sunday) to perform the point inclusion test. Although this algorithm is simple and fast, discovering whether a point lies inside or outside the polygon requires testing it against all of the edges of the polygon. Suppose there are 4 spline segments and each segment is broken into 20 smaller line segments. Then there would be 20*4+1=81 points in the array. In RGB colour space the number of palette edges would be no more than 6. However, as we will discuss later, in the

HSV colour palette the polygon of intersection may contain more than 70 edges. We therefore have to perform 81*70=5670 tests every time the shape of the curve is changed. To speed up the algorithm, we split the polygon into several sub-regions according to the *y* coordinate. An example is shown in Figure 8. The polygon contains 12 edges. We split the polygon into 5 sub-regions and the height of each sub-region is 0.1. For a point *P*, which falls into the sub-region 3, we only need to test it with the edges of the red polygon V1-V2-V3-V4-V5-V6-V1 so that we reduce the number of tests from 12 to 6.

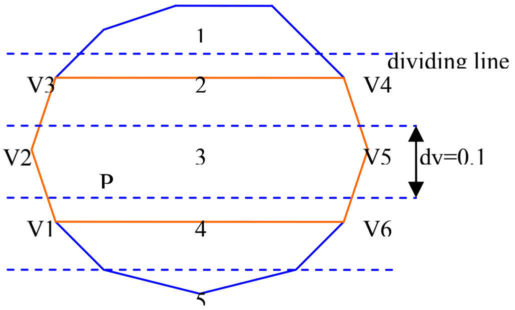


Figure 8 Division of the polygon into sub-regions.

3.2 3-D HSV Colour Picker

Visualizing the HSV colour space is more complex than visualizing the RGB space:

- The underlying colour model used by computer hardware is RGB. To show an HSV colour on screen, we need to do the conversion first.
- The HSV colour model is a non-linear transform from the RGB model so linear changes in Hue, Saturation and Value do not result in linear changes of RGB parameters. Because of this, for a slice cut out of the HSV cone, we cannot just assign the corresponding RGB colours to its vertices and let the graphics card automatically mix the colours inside. Remember that the colours generated by the graphics card are linearly changing in RGB parameters only.

In order to show the colours on the surface of the HSV cone or inside the colour palette, we need to transform them one by one to RGB format. This method requires a huge amount of calculation. However, by converting different colours from HSV to RGB, we discovered that if Hue remains the same, linear changes in Saturation and Value approximately result in linear change in RGB parameters. Based on this observation, we break the HSV cone into tetrahedrons for rendering. For each tetrahedron:

delta_Hue = 360 degree / number_of_tetrahedrons.

If the number of tetrahedrons is large enough (i.e. the Hue variation in each tetrahedron is small enough), we can consider that the Hue in each tetrahedron remains constant. Given this assumption, the RGB values change linearly within each tetrahedron from one point to another and we can make use of the hardware rendering again. Currently in the program the HSV cone is broken into 72 tetrahedrons for rendering. Note that in the illustration figures, the cone was only broken into 36 tetrahedrons for clarity.

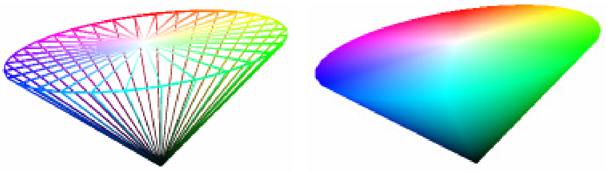


Figure 9 The HSV colour solid is divided into tetrahedrons for rendering

3.2.1 HSV Colour Palette

The palette in the HSV colour picker is also the intersection of a cutting plane through the HSV colour solid. Because the HSV cone is treated as a series of tetrahedrons, the slice will be composed of a series of triangles or trapezoids. Placing the cutting plane in different positions would result in three types of palettes:

1) When the cutting plane intersects with the central axis of the HSV cone as shown in Figure 10, the plane intersects with all the tetrahedrons and hence the palette will contain the full range of Hues. In this case, the palette is composed of 72 triangles with each triangle representing a single Hue. Hence, the HSV colour palette may contain more than 70 edges.

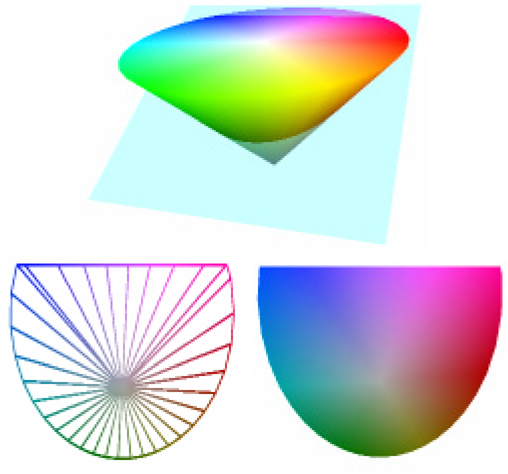


Figure 10 If the plane goes through the central axis, the colour palette (left) contains the full range of hues.

2) When the cutting plane does not intersect with the central axis as shown in Figure 11, the plane only intersects with several adjacent tetrahedrons and the palette contains a sub-range of Hues. Here the two sections of the first and last tetrahedron which intersect with the plane will be triangles while the rest of the sections are trapezoids. In our implementation, each of the trapezoids is further broken down into two triangles for rendering.

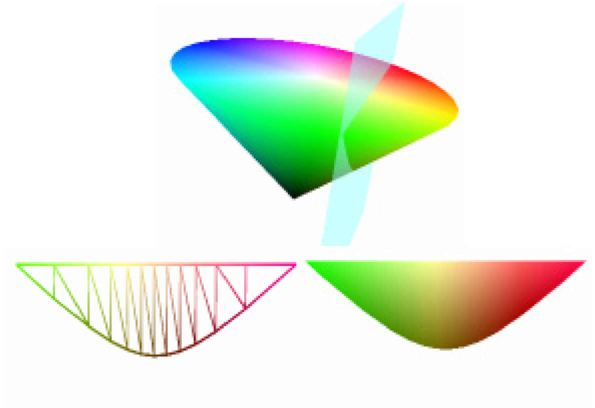


Figure 11 If the plane does not go through the centre axis, the colour palette (left) contains only a subset of hues.

3) When the central axis is on the cutting plane, the plane only intersects with 2 tetrahedrons which are opposite to each other. So the palette only contains two triangles and there are only 2 Hues in the colour palette.

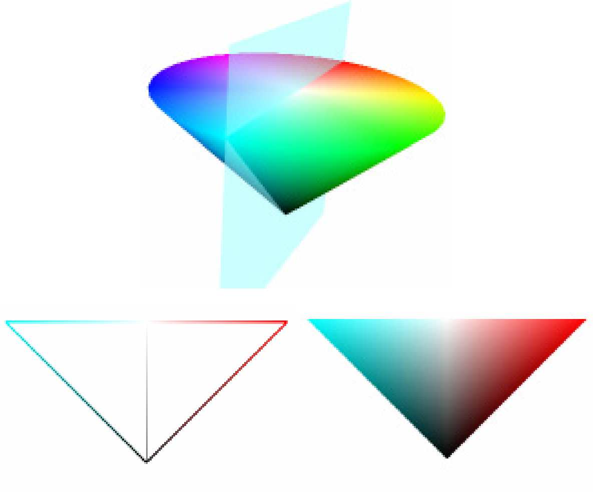


Figure 12 If the central axis is on the plane, there are only two hues on the palette.

The colour picking mechanism in the HSV palette is the same as in the RGB palette: a user can pick a single colour or can use a spline to the do colour interpolation.

3.3 3-D Munsell Colour Picker

3.3.1 Visualisation of the Munsell Colour Space

The Munsell colour space is an asymmetric tree made up of colour chips with each chip representing a single colour. Because this colour space was created based on observation, there is no mathematical formula to convert it back to any other mathematically defined colour spaces. Some organisations like GretagMacbeth[3] and the Munsell Colour Science Laboratory[4] provide tables to convert each colour in the Munsell colour space to CIEXYZ, CIELAB and RGB format. The conversion table used in this study is available from the GretagMacbeth website. There are 2350 colours in the table which are a subset of all the Munsell colours. Currently this is the only free conversion table available. More colours can be added to the Munsell colour picker if the resource is available. Some colours in the table already exceed the gamut of the RGB space so that their corresponding RGB values are only approximations.

In our Munsell colour picker, colour chips are drawn as hexahedrons as shown in Figure 13. The Hue circle starts at the *x*-axis and there are 40 Hues altogether. Value is indicated by the *y*-axis with a range from 1 to 9. The position of each hexahedron is determined by the Hue, Chroma and Value it represents.

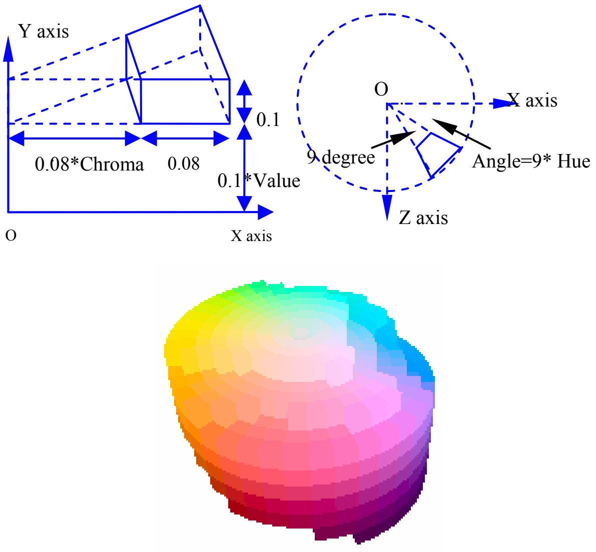


Figure 13 Diagram of a colour chip (top) and how they appear in the Munsell tree (bottom)

3.3.2 3-D Munsell Colour Palette

Since the Munsell colour space is made up of chips, cutting a slice out of it means determining which chips intersect with the plane. The simplest method is to test the chips one by one and find out whether each of the 8 vertices in a chip are on the same side of the plane. There are 2350 chips in the current Munsell colour space. We therefore need to execute 18800 (2350*8) tests every time the plane is repositioned. In the future, if more colours chips are added to the colour space, even more tests would have to be performed.

In order to speed up the program, we decided to test the colour chips group by group, which can reduce significantly reduce the amount of calculation required. The testing algorithm is as follows:

1) Divide the chips in the Munsell colour space into 40 groups according to Hue. All the chips in the same

[3] GretagMacbeth AG: http://www.gretagmacbeth.com/index/products/products_color-standards.htm

[4] The Munsell Color Science Laboratory: http://www.cis.rit.edu/mcsl/

group form a pentahedron shown in the leftmost Figure 14. Each pentahedron is further divided into 9 levels with each level indicating a single Value.

2) For each pentahedron, test whether it intersects with the plane (each pentahedron has 6 vertices). If not, none of the chips within this pentahedron intersect and we can search the next group of chips. Otherwise, go to step 3 for further testing.

3) Calculate the highest (H) and lowest (L) intersection points of the pentahedron and the plane (See the middle of Figure 14). Only those chips sitting between H and L need to be tested in the next step.

4) Search level by level from L to H. At each level, calculate the farthest (F) and closest (C) intersection points. All the chips between F and C intersect with the plane. Insert them into a list.

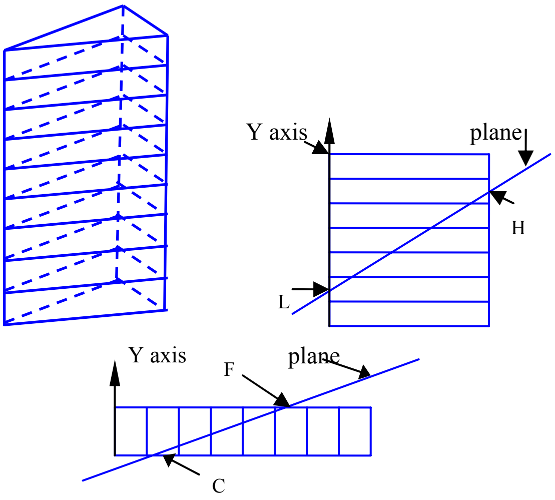


Figure 14 In intersection testing, the colour chips are tested group by group and level by level.

5) Go to step 2 to test the next group of chips until all 40 pentahedrons are tested.

After the intersecting chips are found, the next step is to arrange them in a colour palette. Initially, we tried to show the chips as squares on the X-Y plane. All the chips with the same Hue were arranged in the same column. In each column, chips were sorted by Value first and then by Chroma. However, the result was not satisfactory because the chips' relative positions were changed. As we discussed before, the biggest advantage of the Munsell colour space lies in the way it organises the colour chips: Equal geometric distance produces equal visual distance in colour. It is difficult to preserve this characteristic using a 2-D palette. Therefore we decided to draw the palette in 3-D by keeping the colour chips in the same positions as they are in the Munsell colour space. Figure15 shows the 3-D Munsell colour palette. The x-axis indicates where the Chroma circle starts and the y-axis indicates the brightness (Value). The user can rotate the 3-D palette by mouse.

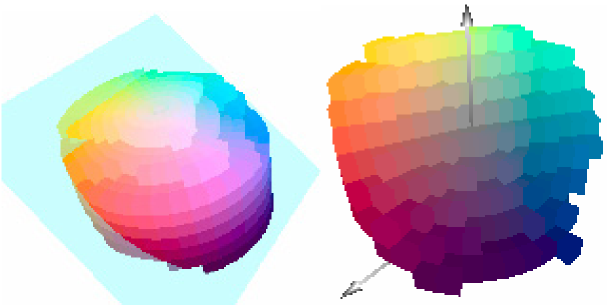


Figure 15 The 3-D palette of the Munsell colour picker.

3.3.3 Colour Interpolation in the Munsell colour palette

A 3-D palette works well in preserving the colour chips' relative positions but it also introduces some problems. Namely, we cannot use the spline for colour interpolation. As shown in Figure 16, the spline enters and leaves the chips, skipping many colours the user may want to be included in the interpolation.

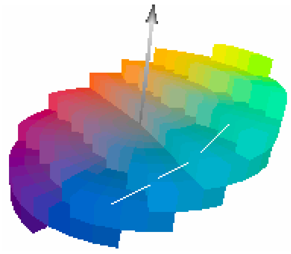


Figure 16 Using a spline to do the colour interpolation

We therefore try to consider the interpolation between two Munsell colours as finding a shortest path from one point to the other in a 3-D graph. It is clear that the length of a straight line segment is always the shortest distance between 2 points. Hence searching the shortest path between 2 chips can be treated as choosing chips in-between to approximate a "line segment".

However, due to the irregularity of the Munsell palette, it is quite possible that the colour chip in the "line segment" does not exist. Take Figure 17 for example. It shows the interpolation path from chip 1 to chip 6. When we arrive at chip 4, the next chip 5' is not a valid colour chip. In this case, we need to abandon chip 5' and search the other neighbours of chip 4 for one which is closest to the destination. Thus in Figure 17, chip 5 would be chosen as the next step.

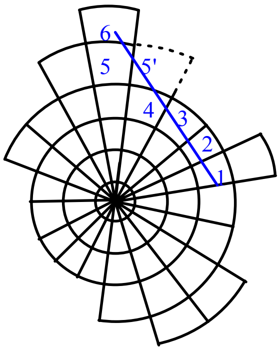


Figure 17 The shortest path between two colour chips

In the Munsell colour palette, the user can pick colour chips by mouse. When a colour chip is selected, it will be framed with dark-grey edges. If more than one colour is picked, the program will automatically calculate the in-between colours. Figure 18 shows the final interpolation result. The chips with light-grey edges are the interpolated colours. The user is also allowed to hide or show the unselected colour chips.

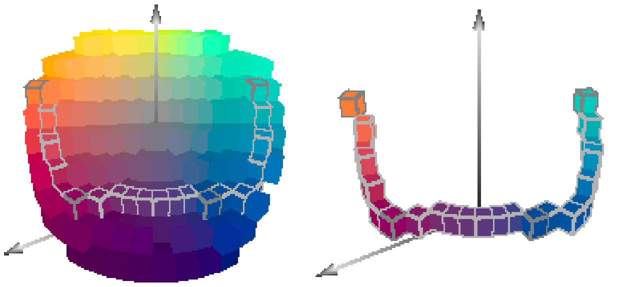


Figure 18 Colour interpolations in Munsell

4 Conclusion

In this study, we implemented 3-D colour pickers for the RGB, HSV, Munsell spaces and addressed various issues that arose in the process of visualising and manipulating the colour pickers. Although 3-D rendering consumes more CPU time, several techniques are used to reduce the amount of calculation and the resulting response time is satisfactory. We believe that 3-D colour pickers are better than their 2-D counterparts though comprehensive usability tests have not yet been carried out. We plan to validate our claim as a further study. We are also going to extend the simple control interface to provide better functionality. For example, in the RGB and HSV colour picker, 64 colours are evenly sampled out of the spline. In the future, the user should be allowed to specify how many colours are sampled in each segment of the spline.

It is possible to develop 3-D colour pickers based on other colour spaces. However, there are some problems yet to be solved, most significantly, some colour spaces contain colours exceeding the gamut of the RGB space and hence can only be approximated on screen. The methods of approximating these colour spaces need further investigation.

5 References

Bauersfeld P.F. and Slater J.L. (1999): User-Oriented Color Interface Design: Direct Manipulation of Color in Context, *ACM Transactions on Graphics*, Volume 18, Issue 2, pp 96-127

Brewer, C. A. (1999): Color Use Guidelines for Data Representation, *Proceedings of the Section on Statistical Graphics, American Statistical Association,* Alexandria VA. pp. 55-60.

Cardani D. (2001): Adventures in HSV Space, http://ww.buena.com/articles/hsvspace.pdf, accessed 28 Sep 2004

Kochanek Doris H. U., Bartels Richard H. (1984): Interpolating Splines with Local Tension, Continuity, and Bias Control, *Computer Graphics, ACM,* Vol 3, No. 18, pp.33-41

Lilley C., Lin, F., Hewitt W.T., Howard T.L.J.: Colour in Computer Graphics. Student notes, Department of Computer Science, University of Manchester

MacEachren A.M., Hardisty F., Gahegan M., Wheeler M., Dai X.P., Guo D.S., Takatsuka M. (2001): Supporting visual integration and analysis of geospatially-referenced data through web-deployable, cross-platform tools. Presented at Proceeding *National Conference for Digital Government Research*, Los Angeles, CA, 2001.

Meyer, Gary W. and Greenberg, Donald P. (1980): Perceptual Color Spaces for Computer Graphics, Vol. 14, No.3 *ACM SIGGRAPH'80*,

Montag, E.D. (1999) The Use of Color in Multidimensional Graphical Information Display. *IS&T/SID Seventh Color Imaging Conference*, Scottsdale.

Sarah A.Douglas, Arthur E.Kirkpatrick(1999): Model and Representation: The Effect of Visual Feedback on Human Performance in a Colour Picker Interface, *ACM Transactions on Graphics, 18, 96-127*

Slater J. (1989): A Perceptual Basis for the development of color tools in Computer Graphics, *SIGGRATH'89 Course Notes,* Course 1: Perceptual Discrepancies in Col0r Production Technology.

Smith A.R. (1978): Colour Gamut Transform Pairs, *Computer Graphics, Vol 12, No 3*

Sunday D.: Fast Winding Number Inclusion of a Point in a Polygon.: http://geometryalgorithms.com/Archive/algorithm_0103/algorithm_0103.htm, accessed 28 Sep 2004