A Measure for Perceived Detail in Computer-Generated Images

Martin Reddy

Technical Report ECS-CSG-19-96
Department of Computer Science,
University of Edinburgh,
Edinburgh, EH9 3JZ

January 1996.

Abstract

This report will present a procedure for assessing the degree of visual detail which a user can perceive in an arbitrary computer-generated image. This will involve describing an image in terms of its component *spatial frequencies* (c/deg); a measure which is commonly employed in the field of visual perception to characterise the efficacy of the human visual system. The text will present a brief introduction to this visual metric, and then demonstrate how a full-colour computer-generated image can be described in terms of this metric. The approach advocated here is based upon an image segmentation algorithm: Fourier techniques will be considered, and shown to be inappropriate for this particular application.

Keywords: Computer graphics, visual perception, visual acuity, spatial frequency.

1 Introduction

For decades, the field of computer graphics has striven to produce ever-more visually realistic displays. This has seen a progression from simple, bland wireframe figures to full-colour, solid, shaded and textured representations. With the advent of these more accurate rendering models, the ability to quantify the perceptual quality of a computer-generated image has become increasingly desirable (Funkhouser & Séquin, 1993). That is, to analyse the degree of visual detail which the user can perceive in an arbitrary scene.

This information could be used for a number of useful purposes. For example, we could compare the perceptual impact of two different rendering algorithms, or analyse the effect of introducing various anti-aliasing techniques. However, the desire which motivated this particular study is to be able to assess the visibility of regions within a scene; with an aim towards optimising the amount of visual detail which is displayed by a real-time graphics system (Nakayama, 1990). Therefore, this report will concern itself primarily with this goal.

A number of researchers have performed investigations which are closely related to this theme. A few of these are therefore worth mentioning here.

There has been a large body of work in the area of *image quality metrics* (Jacobson, 1995). This field is dedicated to the concise perceptual evaluation of display devices. As such, these metrics are not concerned with the quality of arbitrary images displayed on these devices; and hence are of little benefit to us here.

A number of image segmentation algorithms have been founded on perceptual models (Reed & du Buf, 1993). Many of these have resorted to Gestalt rules of grouping (Katz, 1951) to resolve all perceptually disparate components (Khan & Giles, 1992). Other feature extraction techniques have included attempts to generate a number of *feature maps* to locate 'meaningful wholes' in an image based upon a number perceptual criteria (Soufi & Scrivener, 1992), and the use of localised frequency domain techniques to categorise object groupings (Reed & Wechsler, 1990). In general, all of these algorithms are concerned with the identification of discrete objects within a scene, and not with the quantification of absolute detail within the image as a whole. Also, many of these approaches have only been applied to binary or grey-scale images: little consideration has been given to the analysis of full-colour images.

The human visual system is thought to be composed of a number of parallel *channels*; each sensitive to a particular size of visual detail (Sekuler & Blake, 1994). These channels can be modelled using 2D Gabor functions (the product of a 2D sinusoid and a 2D Gaussian). Some researchers have therefore tried convolving an image using these Gabor functions in an attempt to develop and validate computational models of human vision (Watson, 1987; Beck *et al.*, 1987). For any single analysis, this convolution will report the degree to which a particular level of detail exists in an image. It does not report the entire range of perceptually visible detail which the image contains. That is, these techniques focus on a different problem to ours, i.e. to what degree one particular visual channel is stimulated by a certain amount of detail: whereas we are effectively concerned with *which* channel(s) are optimally stimulated by a certain amount of detail.

It can be observed that the requirements of our application are slightly different to all of those mentioned above. As such, a modified approach must be formulated. Accordingly, the remainder of this report will begin with a brief investigation into some perceptual theory with regards to how much detail a user can perceive, and how this may be modelled. This will present a convenient metric for perceived detail which will be used for the purposes of this study. The subsequent discourse will then detail how a computer-generated image can be described in terms of this metric.

2 Background

2.1 In Search of a Metric for Perceived Detail

A user's ability to perceive detail (denoted as their *visual acuity*) has been thoroughly analysed by vision scientists for a number of decades. The most common way to measure this is with a device called a *contrast grating*. This is simply a figure containing a smoothly-varying pattern of light and dark vertical bars (see Figure 1).

There are two factors which affect the appearance of a contrast grating: its *contrast* and *spatial frequency*. Contrast is simply a measure of the luminance difference between adjacent light and dark bars; whereas spatial frequency is a measure of the spacing between bars.

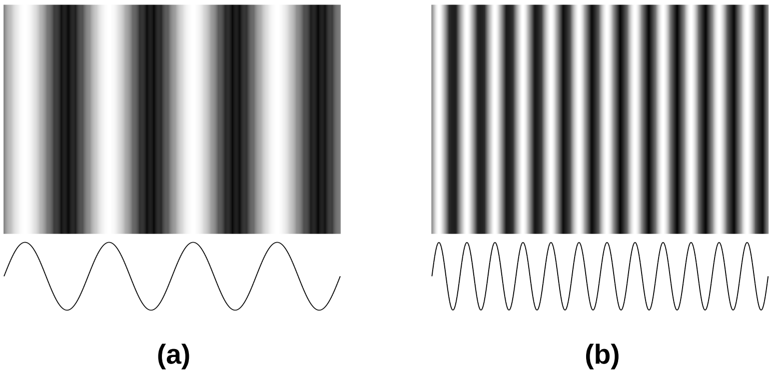


Figure 1: *An illustration of two contrast gratings displaying: (a) a low and, (b) a higher spatial frequency. The curve below each of the gratings shows the sinusoidal nature of the intensity distribution*

For a number of different contrast gratings, the limits of human vision can be investigated and recorded in terms of these two parameters. In this fashion, it has been found that a user's visual acuity varies in relation to the orientation of a contrast grating (Campbell *et al.*, 1966), its velocity across the retina (Kelly, 1979) and the degree to which it is placed in the user's peripheral field (Rovamo & Virsu, 1979). All of these effects can be modelled in terms of contrast and spatial frequency.

Spatial frequency is normally described in units of cycles per degree (c/deg). For example, if the contrast grating in Figure 1(a) were to be positioned such that it occupied 1 degree of a user's visual field, then this would constitute a spatial frequency of 4 c/deg (because it contains four complete cycles of luminance).

Since Schade's initial experiments on the contrast-sensitivity of the eye (Schade, 1956), there has been a substantial amount of work done to describe the characteristics and limitations of the human visual system in terms of spatial frequency. Therefore, if we were able to describe a computer-generated image in terms of its spatial frequency content, then we would have a wealth of data at hand to assess the perceptual nature of that image.

2.2 Applying the Metric

It has been shown that the simple, 1D measure of spatial frequency can be applied to a complex, 2D computer image (Reddy, 1995). To summarise this assertion, there are three principal factors which we must consider: 1) whether visibility of a 1D sine-wave grating can be applied to a square-wave grating (because a pixel-based display is essentially a square-wave distribution of light), 2) whether knowledge of a simple 1D grating can be applied to a 1D complex grating, and 3) whether a 1D grating can be applied to a 2D image. These three cases are illustrated in Figure 2.

In the first instance, Campbell and Robson showed how contrast gratings other than sine-wave gratings can be related to the simple harmonic case. In their work, they showed that the visibility of a square-wave distribution is simply related to that of a sine-wave grating (Campbell & Robson, 1968). This relationship breaks down for lower spatial frequencies, but as we are mostly concerned with the high spatial frequencies in an image (i.e.

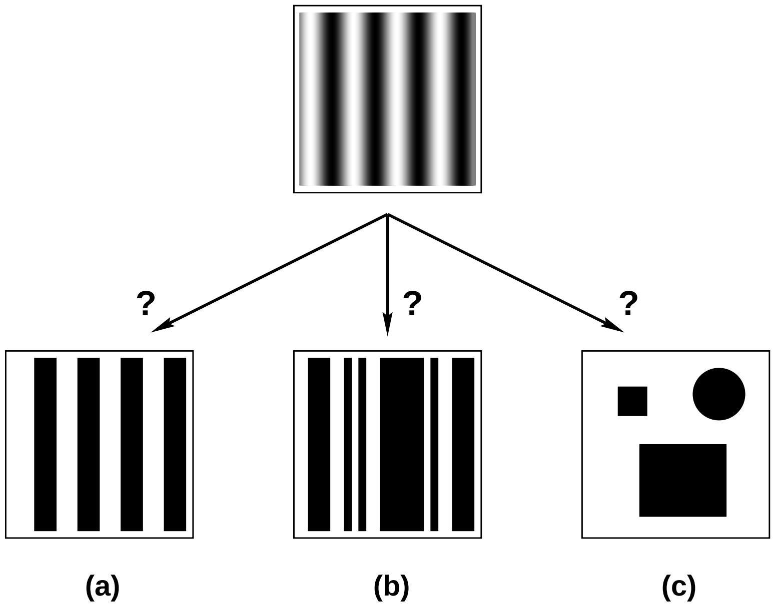


Figure 2: *How does a simple 1D harmonic contrast grating relate to: a) square-wave gratings, b) complex gratings, and c) 2D gratings?*

high detail) this will not affect us to any great degree.

Campbell and Robson also performed experiments with compound contrast gratings in order to investigate how the visibility of these is related to that of their component harmonic gratings. They found that the appearance of a compound grating is characterised by the independent contributions from each of the harmonic components. Their results showed that if a compound grating is displayed such that some of its high frequency components are below threshold, then these features will not be visible in the compound grating and can be removed without any perceivable change being made to the grating.

Finally, a 2D image can be described in terms of spatial frequency if we introduce an orientation parameter for each frequency. For example, the circle in Figure 2(c) will have a horizontal spatial frequency (0°), a vertical spatial frequency (90°), and frequencies at all intermediate angles. (Because the diameter of a circle is the same for every orientation, all of these spatial frequencies will be equal in this particular case.)

2.3 Calculating Spatial Frequency

We have resolved that we wish to calculate the spatial frequency content of a computer-generated image; and we have also shown that this is, at least in theory, possible and meaningful. We can now consider how this analysis may be performed.

The process of extracting all of the relevant spatial frequencies from an image can be broken down into three stages, listed below. The subsequent three sections of this report will elucidate further on each of these processes.

1. *Find all of the visual features in an image.* This will be done using an image segmentation algorithm with a suitable perceptually-based feature extraction mechanism.

2. *Extract all of the relevant spatial frequencies from each feature.* At this stage, these frequencies are *relative*, in that they are in terms of pixels only; with no immediate concept of size. I.e. the units of spatial frequency at this stage are cycles per pixel (c/pixel).

3. *Scale the relative spatial frequency values into units of c/deg.* This transformation can be performed once we know the field of view (FOV) of the display device.

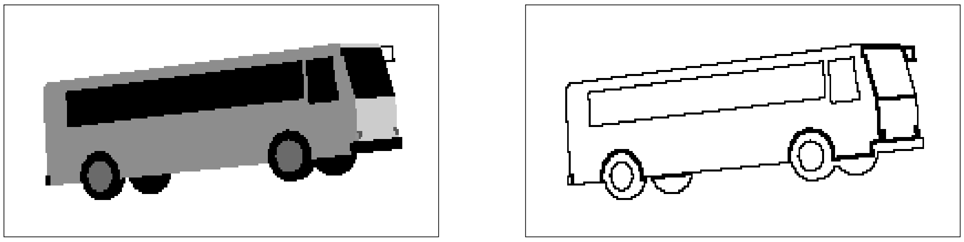
3 Extracting the Visual Features

The aim of this first stage is to extract all of the visually atomic 2D features within an image—the absolute elements of detail in the image. From a physiological standpoint, this would be the extent of a region which maximally stimulates a single neural channel in the vision system. Unfortunately, the mechanism which the human visual system uses to decide this delimitation is still unclear. We must therefore formulate our own, albeit simple, model for this process.

3.1 How to Define the Extent of a Feature

The mechanism which will be adopted here for the task of locating each visual feature in an image is based upon an *image segmentation* paradigm, i.e. a process which takes an image and segments it into a number of individual regions for independent analysis. This is done by taking a single pixel and then attempting to grow this pixel into a region; merging adjacent pixels with the region based upon a certain *segmentation criterion.* The crux of our dilemma therefore rests in the specification of this segmentation criterion.

The trivial case for defining the extent of a feature would be to only merge pixels which are exactly the same colour. This would be a valid definition if we were using a simple flat-shading polygon renderer (such as the Superscape VRT, REND386 and VR386 packages); or if we decided to use the flat-shading mode of our graphics renderer for time-efficiency reasons. Figure 3 illustrates this simple case of feature extraction.



(a) **(b)**

Figure 3: *(a) a flat-shaded image of a bus, and (b) a representation of the boundary of each visual feature in this image. In this case, a visual feature is simply a group of pixels with exactly the same colour.*

However, many contemporary graphics renderers offer far more sophisticated display algorithms than just flat-shading. For example, polygons can often be smooth-shaded (normally using Gouraud or Phong interpolation), anti-aliased and/or texture mapped (Foley *et al.*, 1990). We should therefore be able to cope with these more realistic representations in addition to the simple case of flat-shaded primitives. Consequently, it is evident that we must relax the segmentation criterion slightly in order to include colours which are similar to the target colour, but not necessarily exactly the same. But to what extent can we relax this threshold?

In order to define this, we will utilise the body of research into *perceptually uniform colour spaces*, and *just noticeable differences* (JNDs). This work attempts to define the degree to which an observer perceives two colours as being distinct. The premise for adopting this approach is that, in a smoothly shaded region, one does not inherently notice the colour difference between adjacent pixels; but one perceives a gradual colour gradient over the region as a whole. Using JNDs, we can therefore decide whether two adjacent colours form part of a colour gradient (i.e. part of a feature) or are perceived as the edge of a feature.

3.2 Calculating Just Noticeable Differences (JNDs)

Before we can make any conjecture about the perceived nature of two colours, we must transform these colours into a device independent and perceptually uniform colour space—the standard RGB colour space used by most computer systems is not sufficient. The term device independent is used here to mean a colour space where the colour does not depend upon the device used to produce it; and perceptually uniform refers to the attribute of a colour space whereby the numerical difference between two colours is directly related to their perceptual distance.

Once this has been done, a suitable *colour difference formula* can be used to calculate the number of JNDs between two colours. This value can then be used by the segmentation process to decide whether a pixel should be included or excluded from a feature. Carlson and Cohen consider that 1 JND is practically insignificant (Barten, 1990). They state that 3 JNDs represent a significant perceptual difference (and that 10 JNDs produce a substantial discrepancy). We will therefore adopt a threshold of 3 JNDs.

3.2.1 In Search of a Colour Difference Formula

In 1931, the CIE (Commission Internationale de l'Éclairage) proposed the XYZ tristimulus coding scheme to describe colours in a device-independent manner. This provided an accurate and convenient means of specifying an absolute colour, but it did not consider the perceptual relationship between points in the colour space. In fact, the perceptual non-uniformity of the XYZ colour space is about 80:1 (Poynton, 1993), i.e. the discrepancy between the perceptual distance and the numerical distance of two XYZ colours can vary by up to about 8,000%.

It was not until 45 years later, in 1976, that the CIE introduced the CIE $L^*a^*b^*$ and CIE $L^*u^*v^*$ specifications which attempted to provide a more perceptually uniform colour space. These two systems managed to improve the perceptual non-uniformity of the XYZ system to around 6:1. The formula for calculating the CIE $L^*a^*b^*$ colour difference, ΔE^*_{ab}, was later improved with the advent of the CMC(1:1) colour difference formula. This was developed from experiences in the colourant industries and offers a more accurate correlation for small colour differences (Hunt, 1987). In an experimental evaluation of colour difference models, MacDonald *et al.* found that the CMC formula provides one of the best measures of colour discrimination (MacDonald *et al.*, 1990).

We will therefore use the CMC colour difference formula to estimate the perceptual significance of two adjacent colours. The method for calculating this value (ΔE^*_{cmc}) for two RGB colours has been compiled in Appendix A of this report. Therefore, given any two RGB colours, these can be considered part of the same feature if $\Delta E^*_{cmc} \leq 3$ (3 JNDs).

3.2.2 Gamma Correction

In order to convert RGB colours into a device independent colour space (such as those based upon the CIEXYZ system) we must ensure that these values are *linear* with respect to their intensity on the display device.

In any CRT monitor, the intensity of light which is produced on the screen is proportional to the voltage applied to the electron guns, raised to a certain power. The value of this power is referred to as gamma, γ, and is theoretically equal to 2.5 (Poynton, 1993)[1]. In order to obtain linear RGB signals we must therefore pre-compensate for this non-linearity of the display device. This process is known as *gamma correction*, and can be described by the following formulae; where R, G and B are the (normalised) input pixel values, and R_γ, G_γ and B_γ are the (normalised) gamma corrected signals.

$$R_\gamma = R^{1/\gamma}, \quad G_\gamma = G^{1/\gamma}, \quad B_\gamma = B^{1/\gamma}.$$

N.B. a number of computer systems already incorporate some degree of gamma correction in software. For example, Silicon Graphics IRIS workstations perform partial gamma correction on all RGB colours ($\gamma = 1.7$). This is also the case for the QuickDraw software on Apple Macintosh computers. As a result, images displayed on these machines often look brighter than on, for example, a Sun workstation or PC-compatible. Therefore this intermediate scaling must also be taken into consideration when performing gamma correction. It should be noted that in order to obtain an accurate value for gamma, a light meter should be used to sample the intensity profile for the particular display device (Cowan, 1983).

3.2.3 Viewing Condition Issues

There are a number of factors which can affect the appearance of colours on a computer graphics display (MacDonald *et al.*, 1990; Meyer & Greenberg, 1980). Some of these are beyond our simple control (such as the ambient light in the user's environment and the user's colour perception idiosyncrasies). However, some of the more accountable factors include:

1. **The Size of the Colour Stimuli** : Colour differences vary in relation to the size of the two colours. Most of the CIE colour spaces are based upon colours occupying 2° of visual arc, and therefore do not necessarily provide accurate colour measures for significantly smaller or larger regions of detail. We should therefore attempt to attenuate the results of the CMC formula based upon the size of each colour patch

[1] Recent studies at the BBC have shown that for most CRTs, $\gamma = 2.35 \pm 0.1$ (Roberts & Ford, 1995).

being tested for inclusion in the current feature. Phillips provides empirical data to help perform this scaling (Phillips, 1986).

2. **The Surrounding Colour(s)** : The brightness of a colour can appear different for different surrounding colours. This effect is know as *simultaneous contrast*. It has been reported that this can induce colour difference variations in the order of 10–20% (Phillips, 1986). However, when compared with the variability of a typical observer, this error is not statistically significant. It is therefore felt that we do not need to compensate for this effect.

3. **The Monitor Adjustments** : The value of gamma can appear to vary depending upon the monitor's black level, or brightness, adjustment (Poynton, 1993). Care should therefore be taken to ensure that the display's brightness control is set so that dark elements are reproduced correctly.

3.3 Implementing the Feature Extraction Stage

Taking into consideration the above discussion, we can briefly present the current implementation of the feature extraction process.

A boolean flag is associated with each pixel in the image. This is used to record whether a pixel has been processed yet. All of these flags are initially set to `FALSE` (an optional procedure can be applied which marks all of the background pixels in an image as being processed, so that they are not considered in the analysis). The extraction routine visits each unmarked pixel in turn and attempts to grow it into into a visual feature by merging adjacent pixels, or groups of pixels, if they meet the perceptual criteria defined in the previous section. Each pixel which is found is copied to a *feature bitmap* in order to isolate the feature from the image (and is subsequently marked as having been processed). The feature bitmap does not need to hold full colour information for each pixel: only whether a pixel is in the current feature or not. The feature bitmap therefore need only contain binary information (e.g. a monochrome bitmap).

In summary, the result of the feature extraction stage is a binary feature bitmap containing only those pixels which constitute one visual feature. This bitmap is then passed onto the next stage for further processing. The above steps are then repeated until all visual features have been found and processed.

4 Calculating the Relative Spatial Frequencies

We have now located a 2D visual feature in the image. The next stage is to calculate all of the appropriate relative spatial frequencies which define that feature. These frequencies will be relative because they are only in terms of pixels: so the units of spatial frequency at this stage will be in cycles per pixel (c/pixel).

4.1 Developing a Methodology

We have seen from Section 2.2 that in order to describe a 2D feature in terms of spatial frequency we must incorporate an orientation parameter. Spatial frequency is simply a measure of the size of a stimulus (to be correct, it is inversely proportional to size). Therefore, we effectively want to calculate the largest size of the feature—i.e. the longest contiguous line of pixels—at a number of orientations. Figure 4 attempts to illustrate this notion by presenting a feature along with three of its relative spatial frequencies.

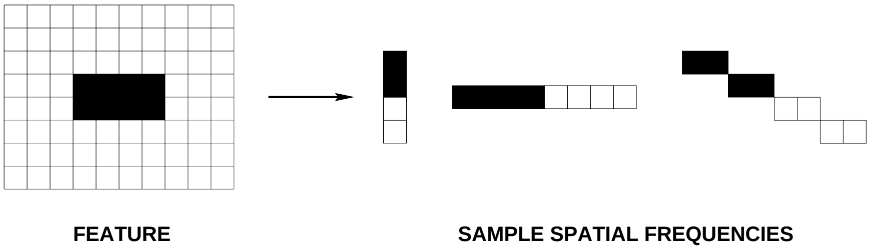


Figure 4: *A simple rectangular feature (left) and three of the feature's relative spatial frequencies at various orientations (right).*

If we take the case of the horizontal spatial frequency (0°) in Figure 4, then we can use this example to describe the method of calculating the c/pixel values for any feature. The longest horizontal line in the feature is 4 pixels in length, so the relative spatial frequency will be inversely proportional to 4 at that orientation. We can note that in terms of the bars in a contrast grating, our line of pixels is actually half a contrast cycle: a full cycle has a peak and a trough, i.e. two features. We must therefore apply a scaling of 1/2 to our calculation to compensate for this fact. Putting all of this together, we can state that the horizontal relative spatial frequency of the feature in Figure 4 is: $1/2 \times 1/4 = 1/8$ c/pixels. That is, one contrast cycle over 8 pixels; as illustrated in Figure 4.

From this, we can develop a general relationship. If we know $l(\theta)$, the length of the longest contiguous line of pixels in a feature at orientation θ, then we can calculate the value of $RSF(\theta)$, the relative spatial frequency of the feature at orientation θ, as follows:

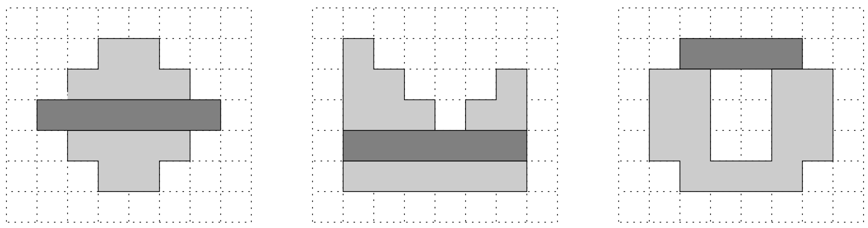
$$RSF(\theta) = \frac{1}{2l(\theta)}. \tag{1}$$

4.2 Implementing the Frequency Extraction Stage

Using Equation 1, we can find the relative spatial frequencies in a feature at any orientation. In order to resolve this however, we need to know how to evaluate $l(\theta)$: the length of the longest contiguous line of pixels in the feature at orientation θ. In the example above, this was a trivial exercise because the feature was a simple rectangle; but how do we calculate $l(\theta)$ for more complex shaped features? This problem resolves to: how do we find the longest contiguous line of pixels, for a particular orientation, which can occur anywhere in the feature? The technique which was used to solve this problem is described below.

Rather than attempt to formulate a mathematical solution based upon the geometry of the feature, a direct analysis of the feature bitmap was employed: for any particular angle, a line of that orientation is notionally scanned through the feature bitmap. At each position, the largest number of contiguous (lit) pixels which exist on that line is recorded. The value of $l(\theta)$ is therefore the largest such result for every line which is passed through the feature. This process is repeated for any number of required orientations.

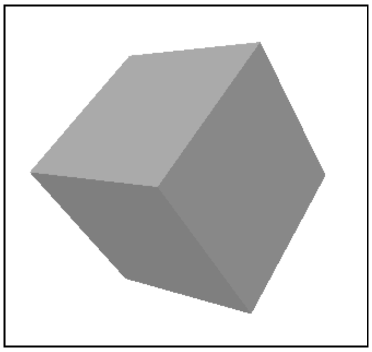
This method of direct analysis has the convenient benefit of implicitly handling all of the degenerate cases of feature shapes, e.g. concave features, and features with holes (sub-features). Figure 5 provides a depiction of this concept in three general cases.



(a) (b) (c)

Figure 5: *An illustration of the maximum horizontal length of a number of features, showing (a) a convex feature, (b) a concave feature, and (c) a feature with a hole. The bold line of pixels represents the first occurrence of the longest contiguous line of pixels which will ultimately be used to calculate the spatial frequency at that orientation.*

The current implementation of this process accepts one parameter: the number of orientations to be sampled for each feature. The product of this stage is the generation of a text file containing the c/pixel values for every visual feature in an image, at the specified number of orientations (this file is identified by a `.rsf` extension). Figure 6 illustrates this result by displaying an image of a flat-shaded cube. Adjacent to this is the corresponding `.rsf` file which was generated for this image with four sample orientations specified, i.e. 0°, 45°, 90° and 135° (note that these values will be the same for orientations of 180°, 225°, 270° and 315°, respectively). That is, if the user specifies n sample orientations, then the extraction process will find frequencies at intervals of $180/n$. Also, any frequency value at $x°$, where $x \in [0..180]$, will be equal to the frequency value at $(180+x)°$.



```
#RSF-1.0 (cube.rgb)
317 297 # pixel resolution of image
4       # Columns of c/pixel values
3       # No. of rows (features)
!
  0 deg  45 deg  90 deg 135 deg
0.0039  0.0071  0.0041  0.0039
0.0034  0.0052  0.0021  0.0047
0.0048  0.0042  0.0054  0.0091
```

Figure 6: *An image of a cube showing three sides (left) and a corresponding* `.rsf` *file which records the relative spatial frequencies for each visual feature in the image at four different orientations (right).*

5 Scaling the Relative Spatial Frequencies

The result of the previous two stages has been to calculate the relative spatial frequency for each feature in an image (at a number of orientations). These values are provided in units of cycles per pixel (c/pixel); however we want them to be available in units of cycles per degree (c/deg) in order to make any perceptual classifications. This naturally requires us to know the horizontal and vertical field of view (FOV) which the display device occupies. In a head-mounted display (HMD) system, this is a trivial matter because the FOV information is provided by the manufacturer. However for a standard monitor or projection screen, the FOV must be calculated as a function of the display size and viewing distance. This can be solved by referring to the tan rule for right-angled triangles, e.g.

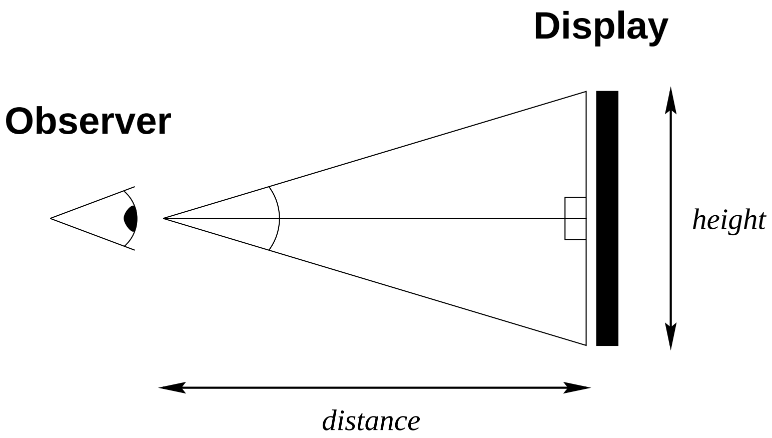


Figure 7: *Calculating the field of view of an arbitrary display device.*

$$\begin{aligned} FOV_{horiz} &= 2 \times \tan^{-1}\left(\frac{width/2}{distance}\right) \\ FOV_{vert} &= 2 \times \tan^{-1}\left(\frac{height/2}{distance}\right) \end{aligned}$$

From these transforms we can calculate the visual arc subtended by one pixel and hence convert the values of c/pixel into c/deg for all orientations. In the process of performing this, we wish to be able to support any arbitrary rectangular display FOV and any (not necessarily similar) rectangular display resolution. That is, the scaling factor in the vertical direction will not necessarily be the same as the scaling factor in the horizontal direction. In order to accommodate this, we can implement the process by: extracting the horizontal and vertical components of the frequency, scaling these components independently, and then recombining these to give the final result. These three steps are detailed below:

1. Extracting the horizontal and vertical components of the spatial frequency, $RSF(\theta)$, at orientation, θ, can be achieved by applying the trigonometrical formulae $cos(\theta) = a/h$ and $sin(\theta) = o/h$ for a right-angled triangle; where the hypotenuse of the triangle equals $RSF(\theta)$, i.e.

$$\begin{aligned} C_{horiz} &= RSF(\theta)\cos(\theta), \\ C_{vert} &= RSF(\theta)\sin(\theta). \end{aligned}$$

2. In order to scale the value of c/pixel into c/deg, we need to know the resolution of the display device (in pixels) and its FOV (in degrees). Then, we can define the horizontal and vertical scaling factors as:

$$\begin{aligned} S_{horiz} &= width/FOV_{horiz}, \\ S_{vert} &= height/FOV_{vert}. \end{aligned}$$

3. Finally, once both the vertical and horizontal components have been independently scaled, we can recombine the two components using Pythagoras' equation to give the resulting absolute spatial frequency, $SF(\theta)$, in units of c/deg:

$$SF(\theta) = \sqrt{(S_{horiz}C_{horiz})^2 + (S_{vert}C_{vert})^2} \tag{2}$$

One final point which should be noted is the differential in perceived line length which is experienced with relation to orientation. E.g. A horizontal line of 10 pixels will appear shorter than a 45° line of 10 pixels. This can be easily explained by referring once again to Pythagoras' equation: a right-angled triangle with both perpendicular sides equal to 10, will have a hypotenuse of $\sqrt{10^2 + 10^2} = 14.142$. I.e. a 45° line of 10 pixels—which spans 10 pixels horizontally and 10 pixels vertically—will be perceived 1.4 times larger than a horizontal line of 10 pixels. We should therefore make an attempt to appropriately scale the relative spatial frequency value, $RSF(\theta)$, depending upon its orientation, before converting it into an absolute value, $SF(\theta)$. This can be achieved if we assume that a line of x pixels length (at any orientation) will always span either x pixels horizontally or x pixels vertically. This is the case for most line drawing algorithms, e.g. the standard Bresenham's algorithm (Kingslake, 1991). Therefore, we can calculate the value of $RSF'(\theta)$, which incorporates the appropriate compensatory scaling factor for any orientation, $\theta = [0°..180°]$, as:

$$RSF'(\theta) = \begin{cases} \sqrt{RSF(\theta)^2 + (RSF(\theta)\tan(\theta))^2}, & \text{when, } 0° \leq \theta \leq 45° \\ & \text{or } 180° \leq \theta \leq 135° \\ \sqrt{RSF(\theta)^2 + (RSF(\theta)\tan(90° - \theta))^2}, & \text{when } 45° \leq \theta \leq 135° \end{cases}$$

6 What's Wrong with Fourier Analysis?

6.1 Introduction to Fourier Analysis

The technique of Fourier analysis can be used to decompose an image function into the set of harmonic intensity functions which sum to give the original image (Bracewell, 1965). This transformation is normally represented mathematically as $F(u, v) = \mathcal{F}\{f(x, y)\}$, where $f(x, y)$ represents the spatial domain of the original image and $F(u, v)$ represents the frequency domain of the Fourier transformed result.

The Fourier transform requires a continuous function to operate on; however, there exists a machine-computable method for calculating the Fourier transform of a discrete function which is called the Discrete Fourier Transform (DFT). The formula for the 2D DFT can be defined as follows:

$$F(u,v) \equiv \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-i2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}. \tag{3}$$

In practice however, computing this function directly is impractical. The more common method is to use the Fast Fourier Transform (FFT) which drastically reduces the complexity of the DFT calculation (Brigham, 1974). Algorithms to compute the FFT can be found in (Press *et al.*, 1974).

6.2 Problems with Fourier Analysis

At first inspection, Fourier analysis sounds like a perfect solution to our problem: it takes a 2D source image and returns a frequency domain containing all of the relative spatial frequencies in that image. However, this is not actually what we require. To illustrate this, Figure 8 presents the FFT of two simple images: a contrast grating and a square.

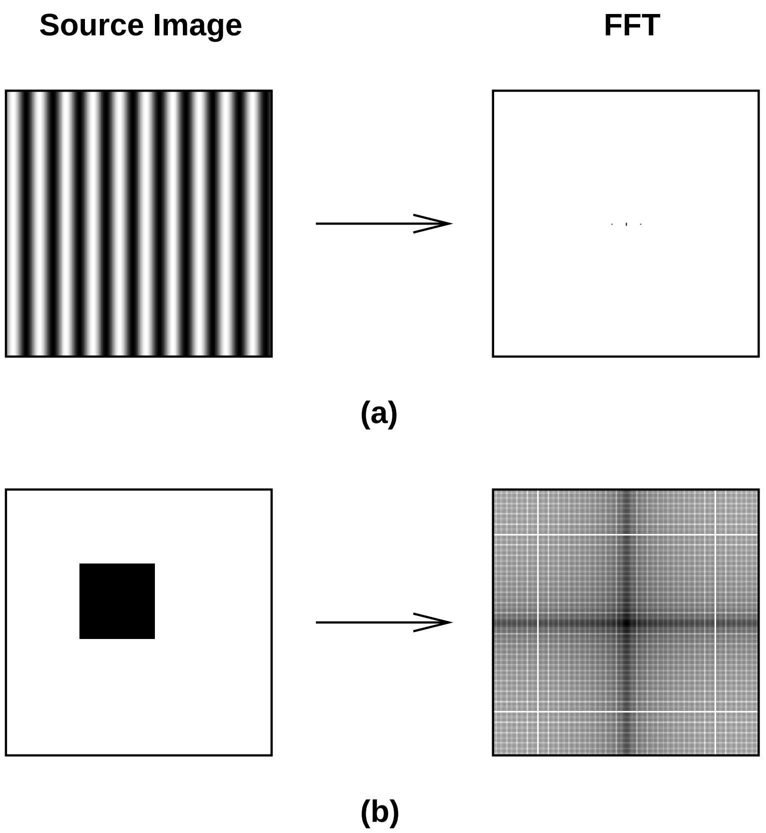


Figure 8: *The Fourier transform of two images: (a) an image of a contrast grating, and (b) an image of a square. The figures on the right represent the Fourier transform of the figures on the left, computed using the FFT algorithm.*

In the first instance, the FFT of the contrast grating gives the expected results. There are three visible points, the centre point is the D.C. term and is not relevant to the analysis. The other two points represent a relative spatial frequency at orientations 0° and 180° which correlates with that of the contrast grating (Note that the Fourier domain is symmetrical about the D.C. term).

However, when we look at the FFT for the image of a single square, we are presented with a frequency domain which contains values across the entire spectrum of relative spatial frequencies (including ones which cannot possibly exist in the image). Part of the reason for this is that Fourier analysis only decomposes the sinusoidal intensity functions of an image: but the simple square image is effectively a square-wave intensity distribution and so it is rich in harmonics. Therefore, if we apply Fourier analysis to a square-wave intensity function (as might be produced by a flat-shaded feature in an image), then this will instantly produce spatial frequencies over the entire range of frequencies.

Specifically, we can conclude that Fourier analysis is not applicable to our application because of the following confounding factors:

1. Fourier analysis will only reveal harmonic spatial frequencies. Any square-wave variances will introduce substantial artifacts in the frequency domain. These cannot be removed from the frequency domain because, given any arbitrary point, we have no information to discern whether this was produced by a physical feature in the image or by the sine-wave approximation of a square-wave feature. That is, Fourier analysis will only (accurately) extract perfectly harmonic variances in intensity[2].

2. Fourier analysis will return values for all spatial frequencies at every point in the image. So for example, if we take the situation with the image of the square: the 0° (horizontal) spatial frequency at the centre of the square will be relative low, but the 45° frequency towards the top-left corner will be very high. That is, any 2D feature with a non-smooth boundary edge will always return frequency values across the entire spectrum.

The image segmentation algorithm circumvents these problems because it only extracts the *fundamental* spatial frequency (i.e. the lowest frequency) for every visual feature; at each desired orientation. The rationale behind only extracting the fundamental frequency is that if the lowest frequency is not detectable by the observer, then none of the higher frequencies will be detectable either—so we only need to record the fundamental frequency for a feature to decide whether it is visible under any specific viewing condition. This concept cannot be easily encoded into a Fourier process.

6.3 Spatial versus Frequency Domain Analysis

Given the above discussion, we can see that the image segmentation approach offers a number of advantages over the Fourier analysis method. These can be summarised as follows:

1. The image segmentation system can be applied to images containing both flat and smooth shaded features (including anti-aliased and texture-mapped features).

[2]There exist other transforms, such as the Walsh Transform (Gonzalez & Woods, 1992), which can locate square-wave intensity components. However, these will then suffer from an inability to locate smooth intensity gradients, i.e. any such features will be recorded as a series of small square-waves instead of a more appropriate sine-wave coding.

2. The corresponding frequency domain is noise-free with no high frequency artifacts. It is therefore much more amenable to accurate analysis.

3. The results are restricted to the fundamental frequency of each visual feature; instead of overloading us by reporting every spatial frequency in the image.

4. We have access to information about the physical position of a feature in the image. This is not available under the Fourier method.

The impact of these advantages can be easily ascertained by inspecting Figure 9. This presents a simple image with three squares of various sizes, and then displays the FFT of that image alongside the results of the image segmentation algorithm (plotted in Fourier space). As can be observed, the FFT result is highly noisy and it is dubious whether this could be of any use as an accurate measure of visual detail. On the other hand, the image segmentation process shows a concise and accurate result: it contains three discernible circles which represent the fundamental spatial frequencies of the three squares at each orientation.

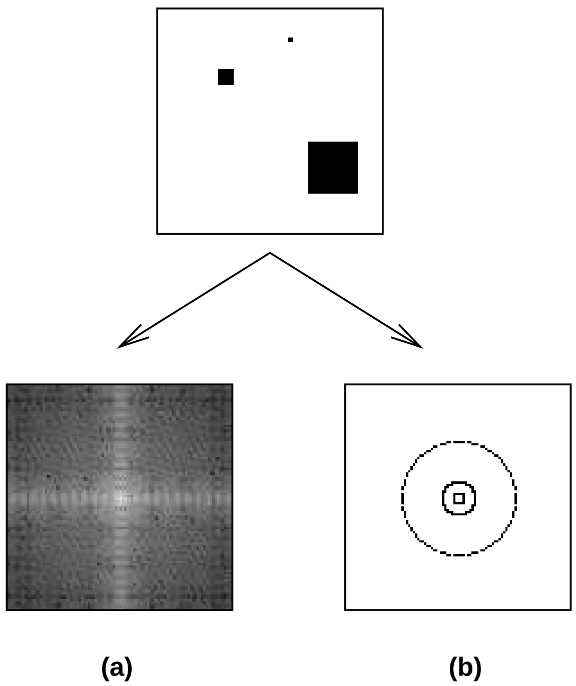


Figure 9: *A comparison of (a) the Fourier based, and (b) the image segmentation based approaches to locating the relative spatial frequencies within a sample image. (Both results are plotted in the Fourier domain.)*

7 Discussion

Having developed a mechanism to describe a computer-generated image in terms of spatial frequency, it would be beneficial to briefly highlight some of the possible applications of this endeavour.

It is well known that the complexity of a geometric model affects the speed at which it can be processed and displayed (Deering, 1994). In general, a more geometrically complex model implies a more visually complex model. So in a real-time graphics system, it becomes necessary to find the optimal amount of detail to render at any juncture. Based upon the visual limitations of the human vision system (as described in Section 2.1), a number of researchers have proposed graphics systems where detail can be varied using criteria such as: an object's distance from the viewpoint, its velocity, and its position in the user's peripheral field (Hitchner & McGreevy, 1993; Funkhouser & Séquin, 1993).

In order to accurately predict when a particular level of detail should be selected, we need to be able to model how much detail the user can perceive at any instant. This has been the focus of the current study. As a direct result, we can now describe the content of an image in terms of its component spatial frequencies. A number of models exist which define how much detail a user can perceive at any instant in terms of this metric (Kelly, 1979; Rovamo & Virsu, 1979). We are therefore now in a position to decided how much detail a graphics system needs to render, based upon the physiological limits of human vision.

An interesting corollary of this result is that we have also conceived a tool which could feasibly be employed by perceptual psychologists to further analyse the intricacies of our visual system using the power and flexibility of current computer graphics technology.

As a final note, we have thus far only considered the calculation of spatial frequencies (c/deg) in a computer-generated image. However, Section 2.1 reports that a user's visual acuity is also dependent upon the contrast of a feature. To be complete, we should therefore also consider how to calculate the contrast of a feature. This topic is not treated in any depth here because it has already received thorough investigation by other researchers. For a good review of contrast in complex computer images, the reader is referred to (Peli, 1990).

8 Conclusion

This report has investigated and developed a means of describing a computer-generated image in terms of a metric for visual acuity: namely spatial frequency (c/deg). We have also seen how this analysis could be used to optimise the performance of a graphics system in a principled manner.

The approach which was adopted is based around an image segmentation process in which pixels are grouped into atomic visual features based upon JND determinations (using the CMC colour difference formula). These features are then analysed to extract their fundamental relative spatial frequencies (c/pixel); and then finally these values are transformed into units of c/deg, based upon the FOV of the display device. During our discussion, we looked at Fourier methods and found these to be inapposite for our purposes: Fourier analysis is a valid technique for qualitative comparisons of image content or frequency filtering operations, but it has severe limitations as an accurate and concise measure for perceived detail.

In conclusion, the image segmentation technique described here provides a good initial solution to the problem of assessing the degree of visual detail in a computer-generated image. Further work can be identified however. In particular, the feature extraction stage may benefit from further investigation and a more formal definition of the extent of a visual feature.

Acknowledgements

The author would like to thank the following people for their various and valued contributions: Dr. William Hossack (Department of Physics, University of Edinburgh) for his lucid and patient chats on Fourier theory. Dr. Ronnier Luo (Design Research Centre, University of Derby) for his helpful comments on colour-difference formulae. Rycharde Hawkes (Department of Computer Science, University of Edinburgh) for providing feedback on an early draft of this report; and last, but most certainly not least, Alina Moga (Signal Processing Laboratory, University of Tampere) for catalysing the initial development of the image segmentation algorithm.

References

Barten, Peter G. J. 1990. Evaluation of Subjective Image Quality with the Square-Root Integral Method. *Journal of the Optical Society of America A (Optics and Image Science)*, **7**(10), 2024–2031.

Beck, Jacob, Sutter, Anne, & Ivry, Richard. 1987. Spatial Frequency Channels and Perceptual Grouping in Texture Segregation. *Computer Vision, Graphics, and Image Processing*, **37**, 299–325.

Bracewell, Ron. 1965. *The Fourier Transform and Its Applications.* New York, NY: McGraw-Hill, inc.

Brigham, E. Oran. 1974. *The Fast Fourier Transform.* Englewood Cliffs, NJ: Prentice-Hall, inc. ISBN 0-13-307496-X.

Campbell, F. W., & Robson, J. G. 1968. Application of Fourier Analysis to the Visibility of Gratings. *Journal of Physiology*, **197**, 551–566.

Campbell, F. W., Hulikowski, J. J., & Levinson, J. 1966. The Effect of Orientation on the Visual Resolution of Gratings. *Journal of Physiology*, **187**, 427–436.

Cowan, William B. 1983. An Inexpensive Scheme for Calibration of a Colour Monitor in Terms of CIE Standard Coordinates. *Computer Graphics*, **17**(3), 315–321.

Deering, Michael F. 1994 (August). Data Complexity for Virtual Reality: Where do all the Triangles Go? *Pages 357–363 of: Proceedings of the VRST'94 conference.*

Foley, James D., van Dam, Andries, Feiner, Steve K., & Hughes, James F. 1990. *Computer Graphics: Principles and Practice.* Second edn. Reading, MA: Addison-Wesley. ISBN 0-201-12110-7.

Funkhouser, Thomas A., & Séquin, Carlo H. 1993 (August). Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments. *Pages 247–254 of: Computer Graphics (SIGGRAPH'93 Proceedings)*, vol. 27.

Gonzalez, Rafael C., & Woods, Richard E. 1992. *Digital Image Processing.* Reading, MA: Addison Wesley. ISBN 0-201-50803-6.

Hitchner, Lewis E., & McGreevy, Michael W. 1993. Methods for user-based reduction of model complexity for Virtual Planetary Exploration. *Pages 622–36 of: Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 1913.

Hunt, R. W. G. 1987. *Measuring Colour*. Chichester, UK: Ellis Horwood Ltd. ISBN 0-7458-0125-0.

Jacobson, R. E. 1995. Image Quality Metrics. *The Journal of Photographic Science*, **43**(2), 42–43.

Katz, D. 1951. *Gestalt Psychology*. London: Methuen and Co. Ltd.

Kelly, D. H. 1979. Motion and Vision. II. Stabilized Spatio-Temporal Threshold Surface. *Journal of the Optical Society of America*, **69**(10), 1340–1349.

Khan, Gul N., & Giles, Duncan F. 1992. Extracting Contours by Perceptual Grouping. *Image and Vision Computing*, **10**(2), 77–88.

Kingslake, Richard. 1991. *An Introductory Course in Computer Graphics*. Second edn. Chartwell-Bratt. ISBN 0-86238-284-X.

MacDonald, L. W., Luo, M. R., & Scrivener, S. A. R. 1990. Factors Affecting the Appearance of Coloured Images on a Video Display Monitor. *Journal of Photographic Science*, **38**(4–5), 177–186.

Meyer, Gary W., & Greenberg, Donald P. 1980. Perceptual Color Spaces for Computer Graphics. *Computer Graphics*, **14**, 254–261.

Nakayama, Ken. 1990. Properties of early motion processing: Implications for the sensing of egomotion. *Pages 69–80 of:* R.Warren, & A.H.Wertheim (eds), *The Perception and Control of Self Motion*. Hillsdale, NJ: Lawrence Erlbaum.

Peli, Eli. 1990. Contrast in Complex Images. *Journal of the Optical Society of America A (Optics and Image Science)*, **7**(10), 2032–2040.

Phillips, P. L. 1986. Minimum Colour Differences Required to Recognize Small Objects on a Colour CRT. *Journal of the Institution of Electronic and Radio Engineers*, **56**(3), 123–129.

Poynton, Charles A. 1993. "Gamma" and its Disguises: The Nonlinear Mappings of Intensity in Perception, CRTs, Film and Video. *SMPTE Journal (Soc. of Motion Picture and Television Engineers)*, **102**(12), 1099–1108.

Press, William H., Teukolsky, Saul A., Vetterling, William T., & Flannery, Brian P. 1974. *Numerical Recipes in C: The Art of Scientific Computing*. Second edn. Cambridge University Press. ISBN 0-521-43108-5.

Reddy, Martin. 1995 (December 18–19). A Perceptual Framework for Optimising Visual Detail in Virtual Environments. *Pages 189–201 of: Proceedings of the FIVE '95 Conference*.

Reed, Todd R., & du Buf, J. M. Hans. 1993. A Review of Recent Texture Segmentation and Feature Extraction Techniques. *CVGIP: Image Understanding*, **57**(3), 359–372.

Reed, Todd R., & Wechsler, Harry. 1990. Segmentation of Textured Images and Gestalt Organisation Using Spatial/Spatial-Frequency Representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(1), 1–12.

Roberts, Alan, & Ford, Adrian. 1995 (May). *Colour Space Conversions: Colour Equations FAQ*. Available on-line from `ftp://ftp.wmin.ac.uk/pub/itrg`.

Rovamo, J., & Virsu, V. 1979. An Estimation and Application of the Human Cortical Magnification Factor. *Experimental Brain Research*, **37**, 495–510.

Schade, O. H. 1956. Optical and Photoelectric Analog of the Eye. *Journal of the Optical Society of America*, **46**, 721–739.

Sekuler, Robert, & Blake, Randolph. 1994. *Perception*. third edn. New York, NY: McGraw-Hill, Inc. ISBN 0-07-113683-5.

Soufi, B., & Scrivener, S. A. R. 1992 (July). *Perceptual Grouping Algorithms and Object Identification*. Tech. rept. 92/C/LUTCHI/0148. Loughborough University of Technology, LUTCHI Research Centre.

Travis, David. 1991. *Effective Color Displays: Theory and Practice*. London, UK: Academic Press Ltd. ISBN 0–12-697690-2.

Watson, Andrew B. 1987. The Cortex Transform: Rapid Computation of Simulated Neural Images. *Computer Vision, Graphics, and Image Processing*, **39**, 311–327.

A Calculating the CMC Colour Difference

This appendix presents in detail the procedure for evaluating the CMC(1:1) colour difference, ΔE^*_{cmc}, for any two colours specified in RGB coordinates. The CMC formula provides a relatively accurate device independent and perceptually uniform system from which we can deduce the perceptual difference between any two colours. For a full treatise on the history and development of the CMC colour difference formula, the interested reader is referred to (Hunt, 1987).

The task of finding the value of the CMC colour difference, ΔE^*_{cmc}, between two (normalised and gamma corrected) RGB colours can be broken down into a series of steps. Each of these will be described presently.

1. Convert both RGB colours into CIEXYZ space (XYZ).
2. Convert both CIEXYZ colours into CIELAB space ($L^*a^*b^*$).
3. Calculate the hue-angle, h_{ab}, and chroma, C^*_{ab} for each of the CIELAB colours.
4. Calculate the differences in L^*, C^*_{ab} and a measure that correlates with the hue-angle. (These differences are referred to as ΔL^*, ΔC^*_{ab} and ΔH^*_{ab}, respectively.)
5. Compute the value of the CMC colour difference formula for the two colours.

A.1 Converting RGB $\rightarrow$ CIEXYZ

In order to convert the RGB values which are used by the graphics system into the device independent CIEXYZ colour space, we are required to know the chromaticity coordinates of the red, green and blue phosphors of the particular monitor being used. These values can be discovered directly by using a spectrophotometer to measure the CIE tristimulus values and then converting to chromaticity coordinates (Hunt, 1987), or by referring to the manufacturer's figures for the display. Once know, the relevant XYZ values can be computed from an RGB specification as follows (Travis, 1991):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ z_r & z_g & z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

If the exact chromaticity values are not know, then the values of an appropriate colour standard could feasibly be used. The ITU Recommendation 709 standard is the most appropriate standard because it is used (or closely approximated) in most contemporary monitors and supersedes the older CCIR 601-1 recommendation used by the NTSC standard (Poynton, 1993). The following equation specifies the RGB $\rightarrow$ CIEXYZ conversion based upon the ITU-709 chromaticity values:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412135 & 0.357675 & 0.180357 \\ 0.212507 & 0.715350 & 0.072143 \\ 0.019319 & 0.119225 & 0.949879 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

N.B. The values for R, G and B must be normalised, i.e. lie in the range [0..1], and they must also be appropriately gamma corrected to compensate for the non-linearity of the display device (see Section 3.2.2).

A.2 Convert CIEXYZ $\rightarrow$ CIELAB

In order to convert between the CIE XYZ and $L^*a^*b^*$ colour spaces, we need to know the X, Y, and Z values for the monitor's White Point (the colour when all three of the CRT guns are firing maximally, e.g. $R = G = B = 1$). The X, Y and Z of the White Point are referred to as X_n, Y_n and Z_n, respectively. The White Point defined within ITU Recommendation 709 is the CIE standard illuminant D65, where $X_n = 0.95045$, $Y_n = 1.00000$, $Z_n = 1.08892$. With this information, we can perform the conversion as follows (Hunt, 1987):

$$\begin{aligned} L^* &= 116(Y/Y_n)^{1/3} - 16 && \text{where, } Y/Y_n > 0.008856 \\ L^* &= 903.3(Y/Y_n) && \text{where, } Y/Y_n \leq 0.008856 \\ a^* &= 500[(X/X_n)^{1/3} - (Y/Y_n)^{1/3}] \\ b^* &= 200[(Y/Y_n)^{1/3} - (Z/Z_n)^{1/3}], \end{aligned}$$

Where, if any of the values of X/X_n, Y/Y_n or Z/Z_n is less than or equal to 0.008856, then it is replaced by the formula:

$$7.787F + 16/116,$$

Where F is the respective ratio, X/X_n, Y/Y_n or Z/Z_n.

A.3 Calculate Hue-Angle and Chroma

Once the $L^*a^*b^*$ values have been computed for an XYZ specification, we can now find the relevant correlates of hue and chroma, referred to as h_{ab} (CIE 1976 a,b hue-angle) and C^*_{ab} (CIE 1976 a,b chroma), respectively. These are defined as:

$$\begin{aligned} h_{ab} &= \arctan(b^*/a^*) \\ C^*_{ab} &= (a^{*2} + b^{*2})^{1/2}. \end{aligned}$$

A.4 Calculating ΔL^*, ΔC^*_{ab} and ΔH^*_{ab}

By this stage, we have the L^*, C^*_{ab} and h_{ab} values for both of the colours which we are comparing. We now want to find the difference between each of these, denoted as ΔL^*, ΔC^*_{ab} and ΔH^*_{ab} respectively.

For the luminance and chroma values, this can be found by simply subtracting one colour's value from the other. However, we do not use the difference in h_{ab} values to calculate ΔH^*_{ab} because hue-angle is an angular measure and cannot be combined with L^* and C^* easily (Hunt, 1987). Instead, we can calculate ΔH^*_{ab} from the CIELAB colour difference formula, as follows:

$$\Delta H^*_{ab} = [(\Delta E^*_{ab})^2 - (\Delta L^*)^2 - (\Delta C^*_{ab})^2]^{1/2}$$

where,

$$\Delta E^*_{ab} = [(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2]^{1/2}$$

(Δa^* and Δb^* are simply the differences between the a^* and b^* values for the two colours).

A.5 The CMC Colour Difference Formula

The CMC colour difference formula improves the perceptual uniformity of the CIE $L^*a^*b^*$ formula by varying the relative weightings of the ΔL^*, ΔC^*_{ab} and ΔH^*_{ab} differences based upon the position of the colour in the CIELAB space. The formula for the CMC colour difference can now be defined as:

$$\Delta E^*_{cmc} = [(\Delta L^*/S_L)^2 + (\Delta C^*_{ab}/S_C)^2 + (\Delta H^*_{ab}/S_H)^2]^{1/2}, \tag{4}$$

where

$$
\begin{aligned}
S_L &= 0.040975L^*/(1 + 0.01765L^*) && \text{when}, L^* \geq 16 \\
S_L &= 0.511 && \text{when}, L^* < 16 \\
S_C &= 0.0638C^*_{ab}/(1 + 0.0131C^*_{ab}) + 0.638 \\
S_H &= (fT + 1 - f)S_C,
\end{aligned}
$$

and

$$
\begin{aligned}
f &= [C^{*4}/(C^{*4} + 1900)]^{1/2} \\
T &= 0.36 + |0.4\cos(h + 35)| && \text{when}, 164^\circ < h < 345^\circ \\
T &= 0.56 + |0.2\cos(h + 168)| && \text{otherwise.}
\end{aligned}
$$