

# Topic 4: Point Processing

## 4.1 Introduction

This section on simple statistical properties, grey level modifications including simple visual techniques and histogram equalisation. A basic survey of hardware to display images via look-up tables.

## 4.2 Basic Image Statistics

Statistical measures of images are the key to understanding image content and the effect of the various processing algorithms that can be applied. This chapter covers from the simplest statistical measures, that consider the image pixel values as a random variable, to noise models which can be used to identify the information content of an image.

When considering the statistics of an image, the simplest two measures are that of *mean* and *variance*. These two where for an N by N digital image  $f(i, j)$  the *mean* is given by,

$$\mu = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j)$$

and the *variance* given by,

$$\sigma^2 = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (f(i, j) - \mu)^2$$

These two simple scalar values can be interpreted as the *average brightness* of the image, and the *deviation* from this average value. In most cases the average brightness is of little interest since the information in the image is represented by the relative pixel values, so that the variance gives some measure of the amount of useful information in the image. This will be explored in greater detail later in section 5 when we consider signal to noise ratio measures.

It is convenient to define a short-hand description for the averaging or *expectation* operation. If we have a one dimensional digital function,  $f(i)$ , for  $i = 0, \dots, N-1$ , then we can write the average or mean as

$$\mu = \frac{1}{N} \sum_{x=0}^{N-1} f(x) = \langle f(x) \rangle$$

where the  $\langle \dots \rangle$  notation assumes a normalised summation over the free variable. The two dimensional equivalent operation is then,

$$\mu = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) = \langle f(i, j) \rangle$$

where the averaging is now implied over both the parameters. Similarly the variance can be written as,

$$\sigma^2 = \langle |f(i, j) - \mu|^2 \rangle$$

It initially appears that to calculate  $\sigma^2$  a double scan through the image is required, first to calculate  $\mu$ , then  $\sigma^2$ , but noting that  $\mu$  is a constant, we can expand  $\sigma^2$ , to give,

$$\begin{aligned} \sigma^2 &= \langle |f(i, j) - \mu|^2 \rangle \\ &= \langle |f(i, j)|^2 \rangle - 2\langle f(i, j) \rangle \mu + \mu^2 \\ &= \langle |f(i, j)|^2 \rangle - \langle f(i, j) \rangle^2 \end{aligned}$$

so that both  $\mu$  and  $\sigma$  can both be formed by a single scan through the image by forming  $\langle f(i, j) \rangle$  and  $\langle |f(i, j)|^2 \rangle$ <sup>1</sup>.

### 4.3 Histograms

The next level of statistical analysis of a digital image is to consider the distribution of pixel values. For a digital image  $f(i, j)$ , it is useful to consider the pixel values  $f$ , as a random variable, where for an 8-bit image  $f$  can take on the values  $0 \rightarrow 255$ . For randomly chosen pixels, we can define the *Probability Distribution Function*  $P(f)$ , as,

$$P(f) = \text{Probability that pixel value is } < f$$

so that,

$$\begin{aligned} 0 &\leq P(f) \leq 1 \\ P(f_{\max}) &= 1 \end{aligned}$$

where  $f_{\max}$  is the maximum value that an image pixel can take, being 255 for a 8-bit image.

The first differential of the *Probability Distribution Function*, is known as the *Probability Density Function* (PDF), being given by,

$$p(f) = \frac{dP(f)}{df}$$

For a digital image the *Probability Density Function* for the value  $f_0$  can be estimated from the number of image pixels that take values between  $f_0$  and  $f_0 + \Delta f$ . If there are  $M_0$  such pixels, then we have that,

$$p(f_0) = \frac{M_0}{N^2 \Delta f}$$

where there are a total  $N^2$  pixels in the image. If we have the image pixels taking integer values, as is typically the case, we set  $\Delta f = 1$ , then the *Probability Density Function*, is given by the *normalised* grey level histogram,  $h(f)$  being the number of pixels with a given grey level, so that

$$p(f) = \frac{h(f)}{N^2}$$

Therefore for an image where the pixels take integer values, the histogram can easily be formed by counting the number of pixels with each grey level in a single scan through the image using the looping structure of,

```

for i = 0 to N-1 {
  for j = 0 to N-1 {
    increment h(f(i, j))
  }
}

```

This gives, for the standard *toucan* image the histogram shown in figure 1.

---

<sup>1</sup>If  $\langle |f(i, j)|^2 \rangle \approx \langle f(i, j) \rangle^2$  then it is necessary to form a double scan to prevent numerical errors from taking the difference to two similar numbers.

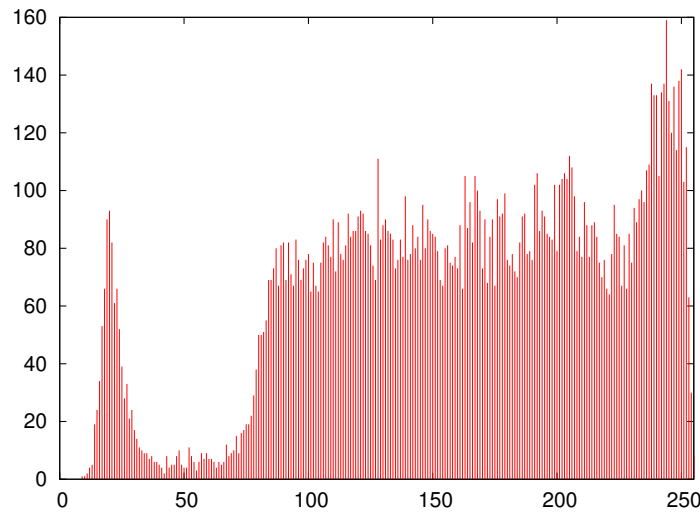


Figure 1: Greylevel histogram of the toucan image.

The *mean* and *variance* of an image can be obtained from its *Probability Density Function* (which can be estimated from its grey level histogram). The *mean* being given by

$$\mu = \int_{-\infty}^{\infty} f p(f) df$$

and the *variance* by

$$\sigma^2 = \int_{-\infty}^{\infty} (f - \mu)^2 p(f) df$$

which are the first and second moments of the distribution respectively.

In the digital case, from the discrete grey level histogram, given by,

$$\mu = \frac{1}{N^2} \sum_{f=0}^{f_{\max}} f h(f)$$

and

$$\sigma^2 = \frac{1}{N^2} \sum_{f=0}^{f_{\max}} (f - \mu)^2 h(f)$$

where the pixel values are assumed to be integers in the range  $(0 \rightarrow f_{\max})$ . Since the histogram contains relatively small quantities of data, typically 256 integers, then the mean and standard deviation are simply, and very efficiently calculated from the histogram. From a computational efficiency viewpoint if both the histogram and the first order statistics of an image are required, the histogram should be calculated and then the statistics formed, however if only the statistics are required it is typically more efficient to calculate them from the pixel and power sums discussed above.

It is seen above that the histogram, or *probability density function* and the first order statistics of an image are closely linked, so that any modification of the histogram, will result in a modification of the first order statistics.

### 4.3.1 Range and meaning of first order statistics

The *mean* is simply the *average brightness* of the image, so *dark* images, for example images of star field with large areas of dark sky will have low *means* while *bright* images for example arial photography of glassier field will have *high* means. However in many imaging systems the actual pixel brightness will often depend on the collection parameters.<sup>2</sup>

The *variance* is a measure of the *spread* of pixel values away from the mean and is thus a measure of image contrast,

- Low variance: low contrast image
- High variance: high contrast image

It is also used as a quick calculation of image *sharpness* being used in simple autofocus digital cameras<sup>3</sup>.

The values of the first order statistics depends on the image, however it is possible to put some limits of there values. One interesting case is when all the pixel values from  $0 \rightarrow f_{\max}$  occur with equal probability, or the histogram is flat. In this case the PDF is given by

$$p(f) = \frac{1}{f_{\max}} \quad \text{for } 0 \leq f < f_{\max}$$

The mean and variance can then be calculated, see workshop question 4.2, by simple integration to give

$$\mu = \frac{1}{2}f_{\max} \quad \& \quad \sigma^2 = \frac{1}{12}f_{\max}^2$$

while the maximum possible variance of an image can be shown, to be

$$\sigma^2 = \frac{1}{4}f_{\max}^2$$

but this is a rather *odd* image, again see, workshop question 4.2 for details.

## 4.4 Point-by-Point Processing Techniques

The simplest set of processing techniques are those that modify the grey level pixel value of an image on a point-by-point basis, ie. the value of a pixel after processing depends *only* on its initial value and not on its location or neighbours. Since this operation effects image grey levels on a point-by-point basis, the image histogram, and thus the image *PDF* will be altered in a similar way. These techniques can therefor be considered as *Histogram Modification* techniques, or modifying the *First Order* statistics of the image.

For a two dimensional digital image  $f(i, j)$ , we introduce a grey level transformation function  $T()$ , such that the output image is given by,

$$g(i_0, j_0) = T\left(f(i_0, j_0)\right) \quad \text{for } 0 \leq i_0, j_0 \leq N - 1$$

Since this transformation is not position dependent it can be represented by the one dimensional transformations,

$$g = T(f)$$

---

<sup>2</sup>In many video system the image exposure automatically varies to form a mean of  $\approx 0.5 f_{\max}$ .

<sup>3</sup>Modern, good autofocus systems use more complex schemes based of edge sharpness.

where input and output grey levels are given by  $f$  and  $g$  respectively.

As a simple example, consider a grey level stretch of a range of the grey levels of an image, shown graphically in figure 2, and can be written as:

$$\begin{aligned} f < t_0 &\rightarrow g = 0 \\ t_0 \leq f \leq t_1 &\rightarrow g = \frac{g_{max}}{(t_1 - t_0)}(f - t_0) \\ f > t_1 &\rightarrow g = g_{max} \end{aligned}$$

so that pixels of value less than  $t_0$  are set to zero and pixels greater than  $t_1$  are set to  $g_{max}$ . The remaining pixels values are then *stretched* linearly between the zero and the maximum

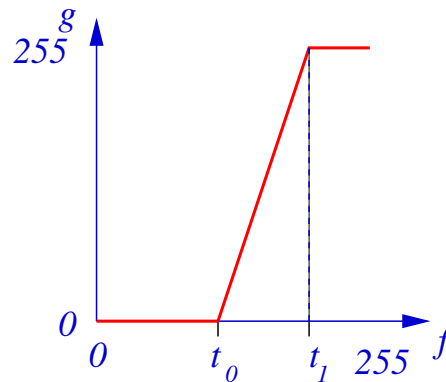


Figure 2: Selective Grey Level stretch

output value. This is a selective contrast stretch, and is similar to the effect of photographically producing an image on a high contrast media as shown in figure 3.



Figure 3: Linear stretch of the toucan image showing enhanced contrast, but also regions of the image set to 0 and 255 respectively.

This operation can be taken one stage further to produce a *two levelled* or (*binary*) output image by the specification of  $T(f)$  such that

$$\begin{aligned} f < t_0 &\rightarrow g = 0 \\ f \geq t_0 &\rightarrow g = 1 \end{aligned}$$

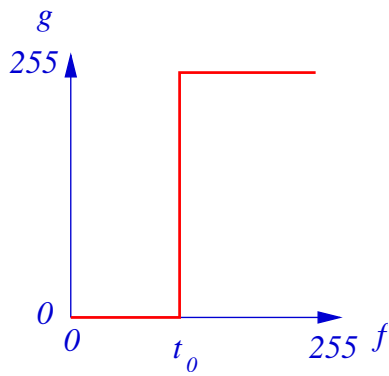


Figure 4: Example of binary threshold, giving a binary output image with pixels set to 0 and 255 only.

to that only pixels greqter that  $t_0$  are set to a constant values and all other pixels set to zero, with typical results shown in figure 4.

Similarly a lower and upper threshold can be set, and the then obtain that

$$\begin{aligned} f < t_0 &\rightarrow g = 0 \\ t_0 \leq f \leq t_1 &\rightarrow g = 1 \\ f > t_1 &\rightarrow g = 0 \end{aligned}$$

to that all pixels in the range  $t_0 \rightarrow t_1$  are set to unity, and all other pixels set to zero, as shwon in figure 5.

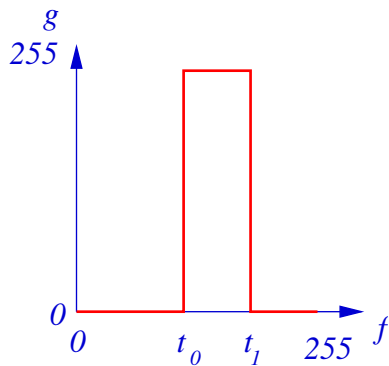


Figure 5: Example of a low and upper binary threshold giving a binary output image.

It is possible to devise a whole range of *Grey Level transformations*, for example, to correct for the *gamma* response of a photographic image. which is given by the non-linear transformation of

$$g(x,y) = f(x,y)^\gamma$$

where,  $f(x,y)$  is the real itensity,  $g(x,y)$  the recorded intensity and  $\gamma$  a constant. If we digitise and display  $g$  as  $g(i,j)$ , the to correct we need a grey level transformation given by

$$T(f) = f^{1/\gamma} \quad \text{really} \quad T(f) = f_{\max} \left( \frac{f}{f_{\max}} \right)^{1/\gamma}$$

The effect of this transformation is shown in figure 6 for  $\gamma = 2.0$  which has the effect of reducing the contrast, and in figure 7 for  $\gamma = 0.5$  which has the effect of increasing the contrast.

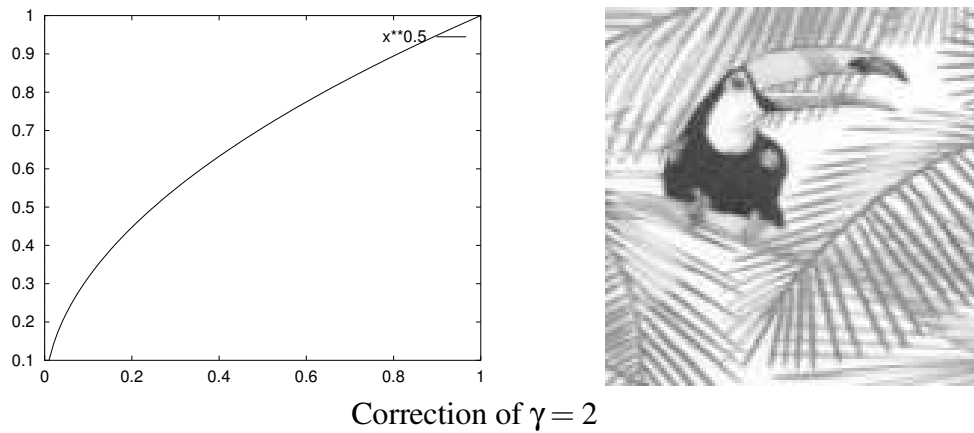


Figure 6: Gamma grey level transformations for  $\gamma = 2$

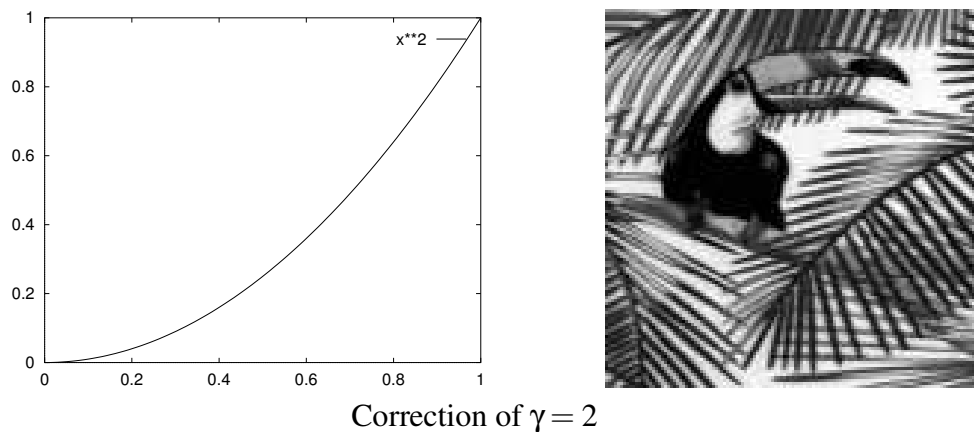


Figure 7: Gamma grey level transformations for  $\gamma = 0.5$

From these graphs it can be seen that when  $\gamma < 1$  the contrast of the low pixel values are contrast enhanced, while when  $\gamma > 1$  the high pixel are contrast enhanced. When we have a full colour image, that consists colour bands, being red, green and blue, then by applying different *gamma correction* to each band the colour balance of the image can be altered. A typical example would be colour balancing for indoor lighting by *boosting* the blue contrast. This is the basis of much simple *colour image enhancement* and is the scheme behind most processing packages supplied with digital cameras to make pictures look better.

These schemes are all based on ad-hoc modification of the grey levels usually via interactive control. However in all cases, the histogram, or thus the *probability density function* of the image has been altered and thus the image statistics have also been altered.

## 4.5 Histogram Equalisation

If we consider that the *important* information in an image is contained in the grey level values that occur most frequently, then a grey level transformation that *stretches* the contrast in regions

of high PDF and compresses the contrast in areas of low PDF would enhance the displayed image. It should be noted that there is an implied assumption about the image structure, that the most frequently occurring pixels values contain the *required information*. If this assumption is not true then strange results occur; which will be discussed later.

If we consider an image with a PDF (image histogram) as shown in figure 8, then qualitatively,

we can consider 3 regions which we can

A	Few pixels: compress grey levels	so in effect,
B	Many pixels: Stretch grey levels	
C	Few pixels: compress grey levels	

we want to evenly distribute the image pixel value across the whole grey level range with a constant probability density. We can thus consider a grey level transformation where the output image has a constant PDF or *equalised histogram*.

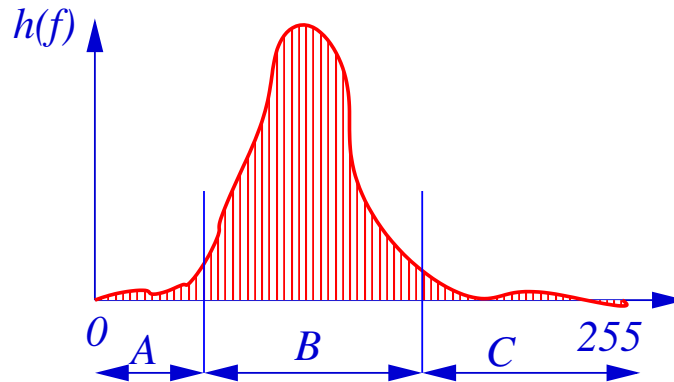


Figure 8: Typical Image PDF (histogram) showing regions requiring grey level stretch and compression

So formally if we have an input image  $f(i, j)$ , with a PDF  $p_f(f)$ , then we wish to form a grey level transformation of

$$g = T(f)$$

to form an output image  $g(i, j)$  with a constant PDF of  $p_g(g) = \text{const}$ . Now we have that

$$p_g(g) = p_f(f) \frac{df}{dg} = \alpha$$

where  $\alpha$  is a constant. Therefore we have that,

$$\frac{dg}{df} = \beta p_f(f)$$

where  $\beta = 1/\alpha$ . This can now be integrated to give the transformation of,

$$g = T(f) = \beta \int_0^f p_f(a) da = \beta P_f(f)$$

so that the required grey level transformation is just a scaled version of the *Probability Distribution Function* of the input image which for a digital image can be estimated from the integrated grey level histogram. The scaling factor  $\beta$ , is typically set so that

$$g_{max} = T(f_{max}) \rightarrow \beta = g_{max}$$

since by definition  $P_f(f_{max}) = 1$ .

So to implement *histogram equalisation* we have to

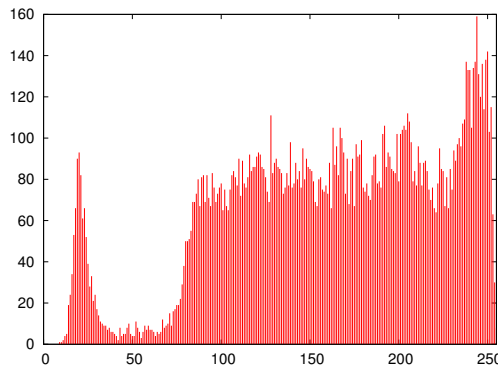


1. Calculate  $h_f(f)$  the grey level histogram of the input image  $f(i, j)$ .
2. Form  $p_f(f)$  which is just the normalised version of  $h_f(f)$
3. Form  $P_f(f)$  which is the summation of  $p_f(f)$

we can then use  $P_f(f)$  as the transformation  $T(f)$  to modify the pixel grey levels.

If we consider the discrete case, where the pixel value take on integer values, then the PDF is replaced by  $h_f(f)$ , the histogram of  $f(i, j)$ , and the grey level transformation by the running sum of the histogram, estimating the *Probability Distribution Function*. The output image is then formed from this digitally estimated transformation. It should be noted that the output image  $g(i, j)$ , is also quantised into integer grey levels, so that all pixels in the input image of grey level value  $f_0$  are modified to have a grey level  $g_0$  which is also an integer value. As a result of this, the histogram (or PDF) of the output image is NOT flat, as expected, but the grey level peaks are spread out in the regions of high PDF and compressed in the regions of low PDF so that the local integral of the output PDF is constant. This is shown schematically in figure 9

Original:



Histogram Equalised:

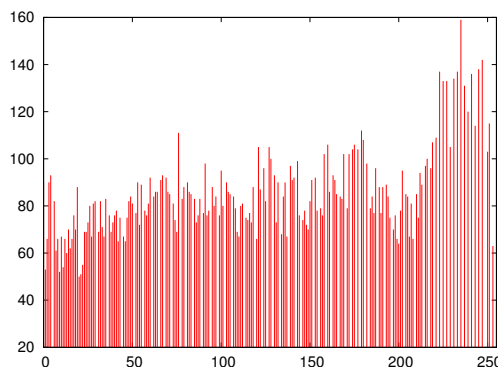


Figure 9: Effect of histogram equalisation for an image taking integer grey level values.

In many applications this histogram equalisation procedure gives very good visual enhancement of low contrast images, and is used routinely throughout image processing. However a word of caution is required: if the *image model* is wrong, ie. we have the case where the most frequently occurring pixel values do not contain the *important* information, then the outcome may at best be unexpected, and at worst totally obliterate the required information. Take the

case of an astronomical image of a cluster of galaxies where the number of pixels associated with the galaxies (bright pixels), is very small compared with the background (dark pixels). Applying histogram equalisation to this image would stretch the grey levels associated with the background (enhancing the typically spurious background noise), and compressing the grey levels associated with the required galaxies, which would very possibly completely disappear in the background noise. See workshop question 4.4 for an example of this.

## 4.6 False Colour Image Display

All the above transformation techniques involve a brightness, or intensity image being transformed to another intensity image on a pixel by pixel basis. Many image display devices have the ability to utilise colour displaying an output composed of the three primary colours, red, green and blue. We can introduce colour into the display of the images by considering three grey level transformations to produce three outputs from a single input image, being

$$\begin{aligned} g_R &= T_R(f) && \text{Red image} \\ g_G &= T_G(f) && \text{Green image} \\ g_B &= T_B(f) && \text{Blue image} \end{aligned}$$

where it should be noted that each image contains the same information but has had its grey levels transformed by a different point-to-point grey level transformation and then assigned the *arbitrary* colours Red, Green, Blue. Also if we have the special case of

$$T_R(f) = T_G(f) = T_B(f) \quad \text{for all } f$$

then the false colour transformation is reduced to the grey level transformations described above.

The effect of this false colour transformations is best shown graphically with a typical set of three curves shown in figure 10 where in this case the zero of the input image is display as *blue* and the maximum of the image as *red*.

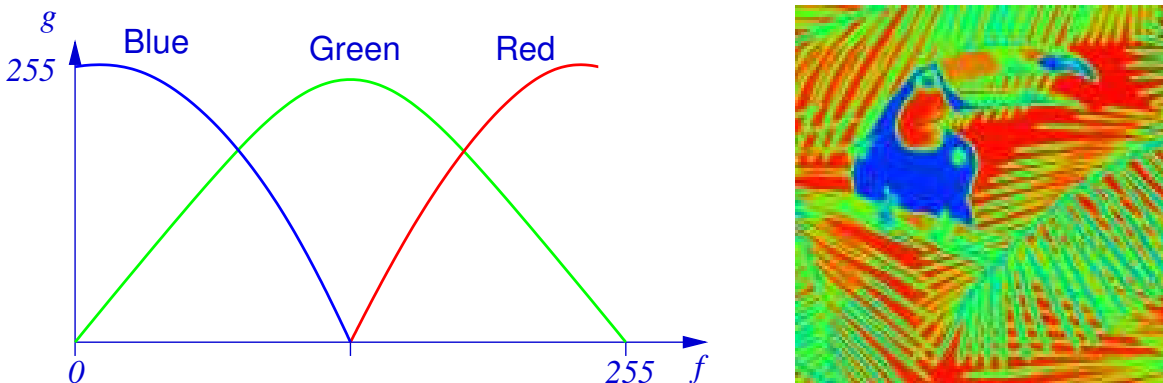


Figure 10: A typical false colour display transformation, being colour option 3 in vx Color Edit menu

In practice vast visual differences in images can be obtained by changing these transformation curves and many different features in the image can be highlighted. It should be noted that almost all enhancement by this technique is subjective, and great care should be exercised in

interpretation of images displayed with false colour. In particular the human visual system is very sensitive to very small changes in *green*, while is rather insensitive to changes in *red* and *blue*, so for example the false colour display curves shown in figure 10 will show a spurious contrast enhancement in the grey level ranges corresponding to a rapid variation in the *green* curve.

One colour display system used throughout radio and infra-red images deserves special mention, that being the *Temperature Mapping*. In this the low valued pixels are considered to represent *cold* regions and the high valued pixels to represent *hot* pixels. The colour display is then modelled on the *Blackbody Radiation Curve*, such that a *cold* body radiated *long* wavelength radiation (RED), and then as the body temperature rises the radiation peak shifts towards the shorter wavelength (BLUE), and finally the spectrum becomes a continuum (WHITE). Such a set of colour transformations is shown in figure 11. The actual slope of the three colour transformations is again typically chosen subjectively, usually being small deviations from a set of Gaussian curves.

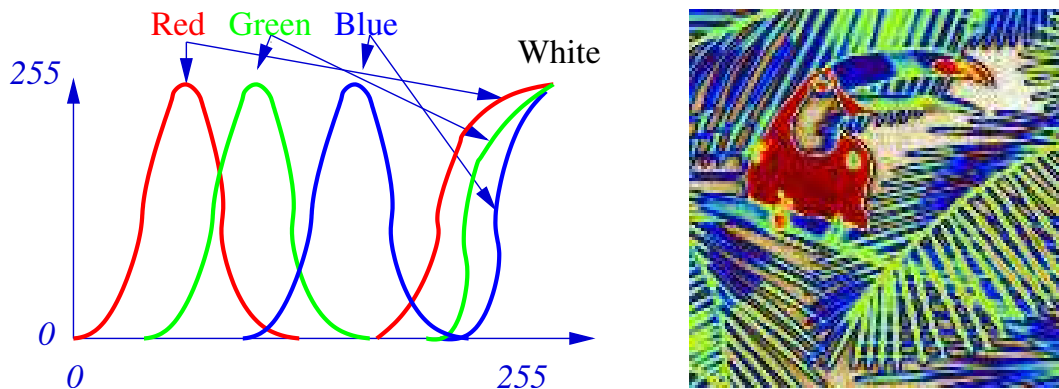


Figure 11: The *temperature* colour map.

If the image data is cyclic, for example it represent a optical or radar phase difference, them 0 and  $2\pi$  are the same value, so when display should be displayed as the same *colour*. If the image is digitised in the usual 8-bit range, them pixel values of 0 and 255 should be thye same colour. A typical such *phase* colour-map is shown in figure 12, where both 0 and 255 are displayed at *red*. Clearly when applied to a non-cyclic image the results are somewhat confusing.

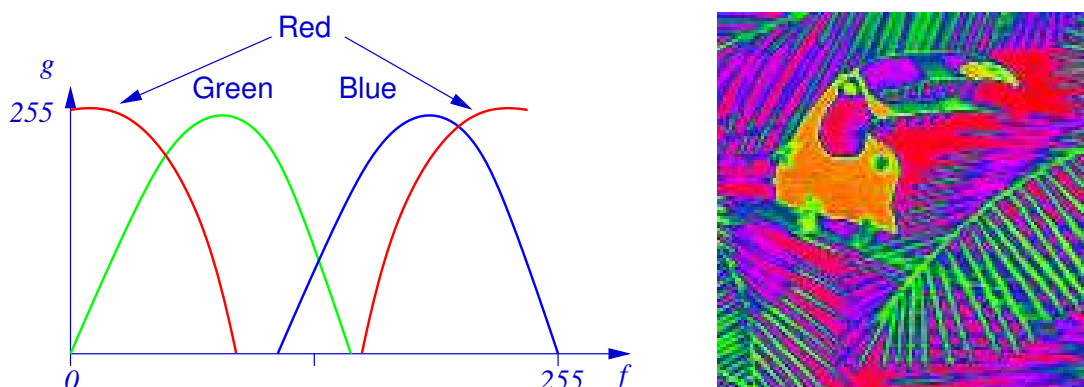


Figure 12: The *phase* colour map where low and high pixels are displayed as the same colour, red in this case.

## 4.7 Implementation of Point-by-Point Processing

In all of these point-by-point operations the output value of each pixel depends only on its initial value, ie. other pixel values in the image have no effect. Therefore each pixel can be treated independently, and the point-by-point transformations can be implemented as an integral part of the image display system. To form a display system, the image data held in memory is read out pixel by pixel, input to an analogue to digital converter (A/D), and then sent to a video display system, as shown schematically in figure 13.

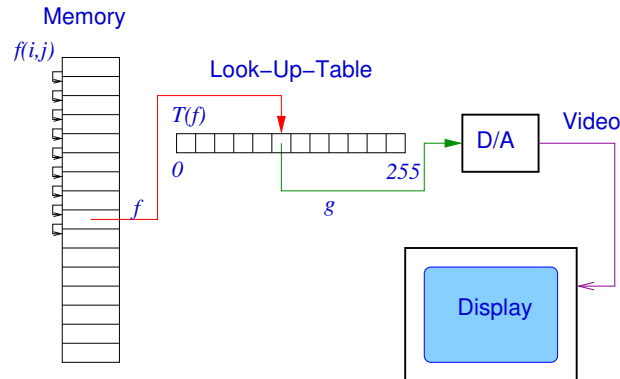


Figure 13: Schematic view of single channel image display system

With such a system the point-by-point pixel grey level transformation can take place as display time, where the pixel value is input to a *look-up table*, and the output of the *look-up table* is sent to the A/D system. Then if the transformation  $T(f)$  is loaded into the look-up table the transformation will take place as each pixel is displayed. A typical colour mapped system is shown in figure 14, where there are three *look-up tables*, each of which is fed with the same pixel value. The outputs of these *look-up table* are then sent to the **Red**, **Green**, and **Blue** guns of a colour video monitor. This allows false colour transformations to be performed, where it should be noted that if the same transformation is loaded into each *look-up table* then a grey level transformation will be displayed on the colour output device.

## 4.8 Summary

The image processing techniques covered in this section that do not rely on the image histogram can easily be implemented as video rates by special purpose hardware in the form of a colour mapped display. Many image processing systems have contrast stretch, binary threshold, gamma correction, and manipulation of the colour tables available via an interactive device (joystick or tracker ball). These techniques can also be applied to live video signals by analogue electronics systems. The techniques involving the histogram of an image requires that the image is stored digitally and then scanned to form the histogram. The grey level modification function  $T(f)$ , can then be written to the image display look-up tables. If it is required to process a real time video image it may not be necessary to calculate the histogram of each individual frame, but the histogram can be calculated for every  $K$  frames and the grey level function modified if required.

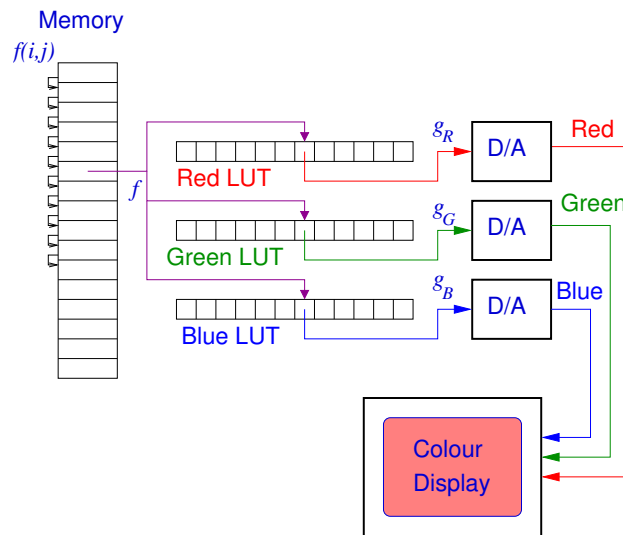


Figure 14: Schematic diagram of false colour image display system

## Workshop Questions

### 4.1 Zero mean in Fourier Space

If we have an  $N \times N$  image  $f(i, j)$  and subtract the mean to form

$$\tilde{f}(i, j) = f(i, j) - \langle f(i, j) \rangle$$

what is the difference between the Fourier transform of  $\tilde{f}(i, j)$  and  $f(i, j)$ .

### 4.2 Range of Statistics

Calculate the *mean* and *variance* if an image with a constant histogram (or probability density function) where the image values are between  $0 \rightarrow f_{\max}$ .



Calculate the maximum possible variance of an image with values between  $0 \rightarrow f_{\max}$ . What is special about this image.

### 4.3 Grey Level Stretch and Threshold

Use the image `wjh/dia/images/junction.pgm` with the `xv` utility to investigate the following operations. All operations are obtainable via the *Color Edit* control panel.

Use the *Intensity* edit display bottom of the middle panel to implement:

- Grey level linear stretch with bound at 40 and upper at 175. (You need 4 control points at the linear interpolation scheme).
- Optimise a grey level stretch to make the “Give-Way” signs readable.
- Perform a binary threshold to try and extract the light car entering the junction from the left. (You need seven control points).

## 4.4 Histogram Equalisation

Use the utility `pgmhist` to form the histogram of `horseshoe.pgm` file, and display this histogram using one of the standard graph plotting utilities, (*gnuplot*, or *xmgrace*).

**Note:** you will have to edit the output of `pgmhist` to remove the first three header lines.

Use `xv` to form the histogram equalised version, save this is a `pgm` file and plot the equalised histogram. Explain what you get.

## 4.5 False Colour Maps

The most common “colour map” for false colour display is called the “Temperature” map. This is based on the ideal of heating a piece of material so that

Values	Colour
Low	Red
Medium	Green
High	Blue
Very High	White

Design an implement such a false colour transformation using `xv` and display a range of images.