Figure 2.9 Red, Green and Blue components of colour image

appears brighter in the red component. The RGB colour space is well-suited to capture and display of colour images. Capturing an RGB image involves filtering out the red, green and blue components of the scene and capturing each with a separate sensor array. Colour Cathode Ray Tubes (CRTs) and Liquid Crystal Displays (LCDs) display an RGB image by separately illuminating the red, green and blue components of each pixel according to the intensity of each component. From a normal viewing distance, the separate components merge to give the appearance of 'true' colour.

2.4.2 YCbCr

The human visual system (HVS) is less sensitive to colour than to luminance (brightness). In the RGB colour space the three colours are equally important and so are usually all stored at the same resolution but it is possible to represent a colour image more efficiently by separating the luminance from the colour information and representing luma with a higher resolution than colour.

The YCbCr colour space and its variations (sometimes referred to as YUV) is a popular way of efficiently representing colour images. Y is the luminance (luma) component and can be calculated as a weighted average of R, G and B:

$$Y = k_r R + k_g G + k_b B \tag{2.1}$$

where k are weighting factors.

The colour information can be represented as *colour difference* (chrominance or chroma) components, where each chrominance component is the difference between R, G or B and the luminance Y:

$$\begin{aligned} Cb &= B - Y \\ Cr &= R - Y \\ Cg &= G - Y \end{aligned} \tag{2.2}$$

The complete description of a colour image is given by Y (the luminance component) and three colour differences Cb, Cr and Cg that represent the difference between the colour intensity and the mean luminance of each image sample. Figure 2.10 shows the chroma components (red, green and blue) corresponding to the RGB components of Figure 2.9. Here, mid-grey is zero difference, light grey is a positive difference and dark grey is a negative difference. The chroma components only have significant values where there is a large

Figure 2.10 Cr, Cg and Cb components

difference between the colour component and the luma image (Figure 2.1). Note the strong blue and red difference components.

So far, this representation has little obvious merit since we now have four components instead of the three in RGB. However, $Cb + Cr + Cg$ is a constant and so only two of the three chroma components need to be stored or transmitted since the third component can always be calculated from the other two. In the YCbCr colour space, only the luma (Y) and blue and red chroma (Cb, Cr) are transmitted. YCbCr has an important advantage over RGB, that is the Cr and Cb components may be represented with a *lower resolution* than Y because the HVS is less sensitive to colour than luminance. This reduces the amount of data required to represent the chrominance components without having an obvious effect on visual quality. To the casual observer, there is no obvious difference between an RGB image and a YCbCr image with reduced chrominance resolution. Representing chroma with a lower resolution than luma in this way is a simple but effective form of image compression.

An RGB image may be converted to YCbCr after capture in order to reduce storage and/or transmission requirements. Before displaying the image, it is usually necessary to convert back to RGB. The equations for converting an RGB image to and from YCbCr colour space and vice versa are given in Equation 2.3 and Equation 2.4[1]. Note that there is no need to specify a separate factor k_g (because $k_b + k_r + k_g = 1$) and that G can be extracted from the YCbCr representation by subtracting Cr and Cb from Y, demonstrating that it is not necessary to store or transmit a Cg component.

$$\begin{aligned} Y &= k_r R + (1 - k_b - k_r)G + k_b B \\ Cb &= \frac{0.5}{1 - k_b}(B - Y) \\ Cr &= \frac{0.5}{1 - k_r}(R - Y) \end{aligned} \tag{2.3}$$

$$\begin{aligned} R &= Y + \frac{1 - k_r}{0.5} Cr \\ G &= Y - \frac{2k_b(1 - k_b)}{1 - k_b - k_r} Cb - \frac{2k_r(1 - k_r)}{1 - k_b - k_r} Cr \\ B &= Y + \frac{1 - k_b}{0.5} Cb \end{aligned} \tag{2.4}$$

[1] Thanks to Gary Sullivan for suggesting the form of Equations 2.3 and 2.4

ITU-R recommendation BT.601 [1] defines $k_b = 0.114$ and $k_r = 0.299$. Substituting into the above equations gives the following widely-used conversion equations:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= 0.564(B - Y) \\ Cr &= 0.713(R - Y) \end{aligned} \tag{2.5}$$

$$\begin{aligned} R &= Y + 1.402Cr \\ G &= Y - 0.344Cb - 0.714Cr \\ B &= Y + 1.772Cb \end{aligned} \tag{2.6}$$

2.4.3 YCbCr Sampling Formats

Figure 2.11 shows three sampling patterns for Y, Cb and Cr that are supported by MPEG-4 Visual and H.264. 4:4:4 sampling means that the three components (Y, Cb and Cr) have the same resolution and hence a sample of each component exists at every pixel position. The numbers indicate the relative sampling rate of each component in the *horizontal* direction, i.e. for every four luminance samples there are four Cb and four Cr samples. 4:4:4 sampling preserves the full fidelity of the chrominance components. In 4:2:2 sampling (sometimes referred to as YUY2), the chrominance components have the same vertical resolution as the luma but half the horizontal resolution (the numbers 4:2:2 mean that for every four luminance samples in the horizontal direction there are two Cb and two Cr samples). 4:2:2 video is used for high-quality colour reproduction.

In the popular 4:2:0 sampling format (‘YV12’), Cb and Cr each have half the horizontal and vertical resolution of Y. The term ‘4:2:0’ is rather confusing because the numbers do not actually have a logical interpretation and appear to have been chosen historically as a ‘code’ to identify this particular sampling pattern and to differentiate it from 4:4:4 and 4:2:2. 4:2:0 sampling is widely used for consumer applications such as video conferencing, digital television and digital versatile disk (DVD) storage. Because each colour difference component contains one quarter of the number of samples in the Y component, 4:2:0 YCbCr video requires exactly half as many samples as 4:4:4 (or R:G:B) video.

Example

Image resolution: 720 × 576 pixels
Y resolution: 720 × 576 samples, each represented with eight bits

4:4:4 Cb, Cr resolution: 720 × 576 samples, each eight bits
Total number of bits: 720 × 576 × 8 × 3 = 9 953 280 bits

4:2:0 Cb, Cr resolution: 360 × 288 samples, each eight bits
Total number of bits: (720 × 576 × 8) + (360 × 288 × 8 × 2) = 4 976 640 bits

The 4:2:0 version requires half as many bits as the 4:4:4 version.

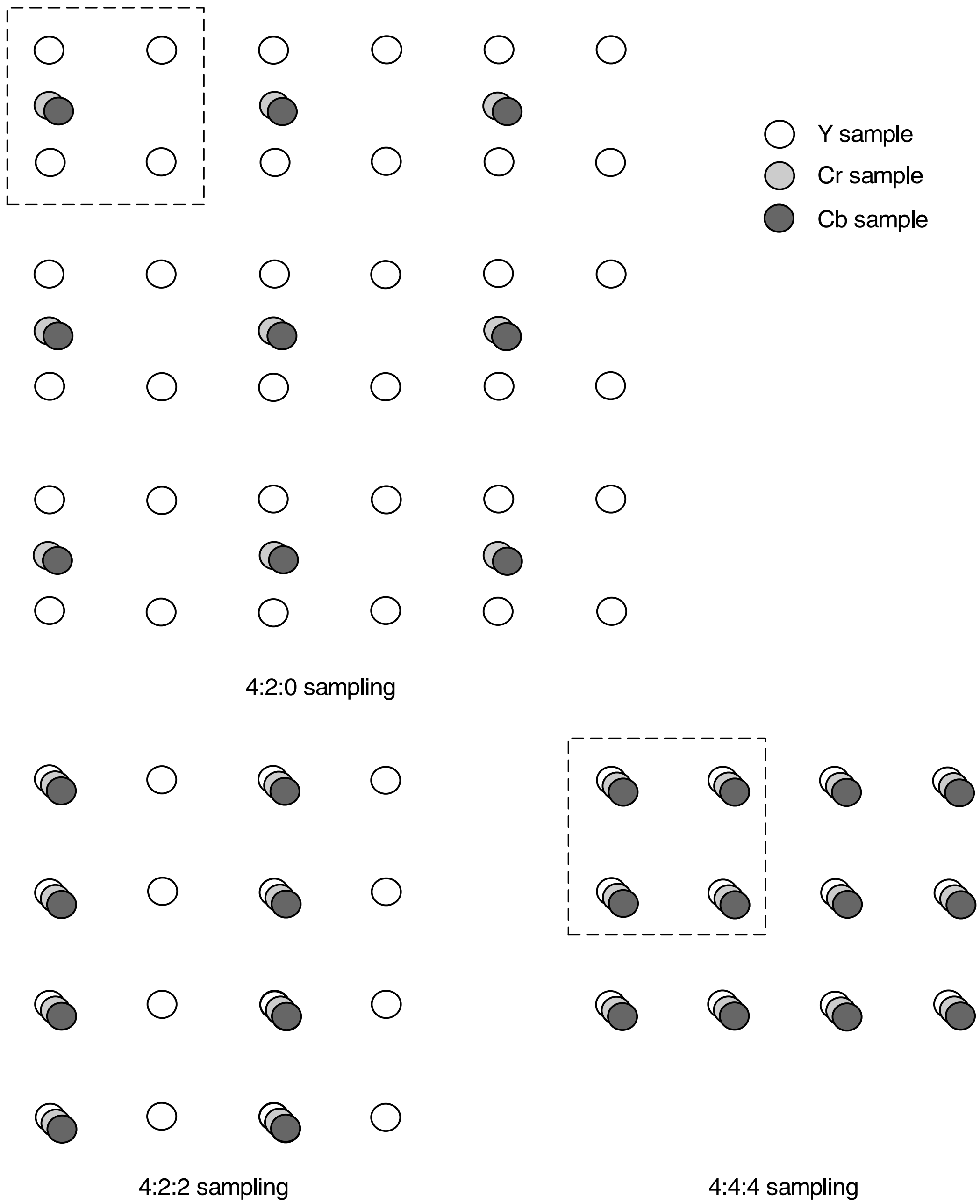

Figure 2.11 4:2:0, 4:2:2 and 4:4:4 sampling patterns (progressive)

4:2:0 sampling is sometimes described as '12 bits per pixel'. The reason for this can be seen by examining a group of four pixels (see the groups enclosed in dotted lines in Figure 2.11). Using 4:4:4 sampling, a total of 12 samples are required, four each of Y, Cb and Cr, requiring a total of $12 \times 8 = 96$ bits, an average of $96/4 = 24$ bits per pixel. Using 4:2:0 sampling, only six samples are required, four Y and one each of Cb, Cr, requiring a total of $6 \times 8 = 48$ bits, an average of $48/4 = 12$ bits per pixel.

In a 4:2:0 interlaced video sequence, the Y, Cb and Cr samples corresponding to a complete video frame are allocated to two fields. Figure 2.12 shows the method of allocating