CHAPTER

Digital Picture Formats and Representations

4

CHAPTER OUTLINE

Since the origins of digital video communications, there has been a need to understand the links between the human visual system and picture representations. Building on Chapters 1, 2, and 3, this chapter describes the picture formats, processing techniques,

For colour versions of Figures 4.10, 4.11, 4.14, 4.15, 4.17 and 4.18 please refer to the electronic version or the website.

Communicating Pictures. http://dx.doi.org/10.1016/B978-0-12-405906-1.00004-0

and assessment methods that underpin the coding process; it introduces the processing methods and sampling structures that enable compression to be so effective.

Building on the mathematical preliminaries from Chapter 3, we first show how samples are represented in digital images and video signals, emphasizing the important aspects of color and spatio-temporal representations. We examine how samples are grouped to form pictures and how pictures are grouped to form sequences of pictures or videos.

The mapping of light levels to brightness as perceived by the human visual system is considered in Section 4.4, showing how the signal from the camera must be transformed using a non-linear mapping (gamma correction) to ensure best use of the bits available. Next, developing further the trichromacy theory of vision introduced in Chapter 2, we introduce the area of color spaces in Section 4.5; these are important since, with relatively little effort, we can create representations that reduce the bit rate for a digital video by 50%. Finally, we introduce the important topic of image and video quality assessment—both as a means of comparing the performance of different compression systems and as a means of controlling encoder modes during compression.

4.1 Pixels, blocks, and pictures

4.1.1 Pixels, samples, or pels

A still image or picture is a spatial distribution of sample values that is constant with respect to time. These are often referred to as pixels or pels.

Monochrome image

A monochrome image is a two-dimensional matrix of luma samples, **S,** with spatial dimensions (often called resolution) $X \times Y$. This is shown in Figure 4.1. If we zoom in on an actual image, such as that shown in Figure 4.2, where the pixel resolution (in

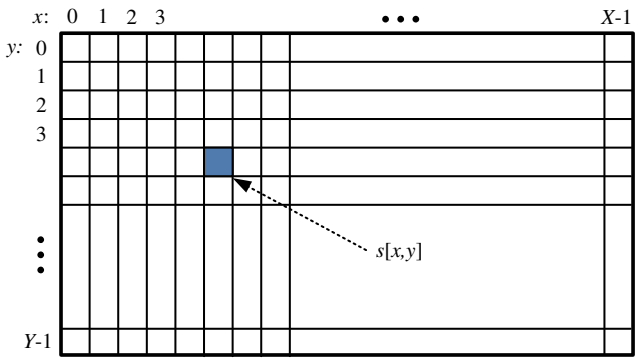

FIGURE 4.1

Image sample array.

FIGURE 4.2

Image samples.

terms of the angle it subtends at the retina) starts to be within the resolving power of the HVS, we begin to see pixelation effects. As we zoom in still further, we lose the structure of the underlying image and the pixel structure dominates.

Figure 4.3 presents this in a slightly different way, through the use of four different sample rates. If you move away from this figure you will see (at around 5 m) that the top two images start to look very similar. If you can get even further away (at around 10 m), all four images will take on a similar appearance. This emphasizes the fact that resolution is not about the number of horizontal and vertical samples, but about the angle that each pixel subtends at the viewer's retina.

Color image

In the case of a color picture, each element of $\mathbf{S}$ is itself a vector $\mathbf{s}$, with three dimensions representing the intensities in a given color space (e.g. RGB, HSI, or YC_bC_r).

So we have in matrix form:

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}[0,0] & \mathbf{s}[0,1] & \cdots & \mathbf{s}[0,X-1] \\ \mathbf{s}[1,0] & \mathbf{s}[1,1] & \cdots & \mathbf{s}[1,X-1] \\ \vdots & \vdots & & \vdots \\ \mathbf{s}[Y-1,0] & \mathbf{s}[Y-1,1] & \ldots & \mathbf{s}[Y-1,X-1] \end{bmatrix} \tag{4.1}$$

where (e.g. for an RGB signal):

$$\mathbf{s}[x,y] = \begin{bmatrix} s_R[x,y] & s_G[x,y] & s_B[x,y] \end{bmatrix}$$

FIGURE 4.3
Pixelation at varying resolutions. Top left to bottom right: 256×256; 64×64; 32×32; 16×16.

4.1.2 Moving pictures

A video signal is a spatial intensity pattern that changes with time. Pictures are captured and displayed at a constant frame rate, $1/T$ where T is the sampling period. In order to trade off temporal artifacts (flicker, judder etc.) against available bandwidth, a frame rate of 24, 25, or 30 Hz has conventionally been chosen. More recently, as screens have become larger and spatial resolutions have increased, higher frame rates have been introduced at 50 and 60 Hz. The new UHDTV standard Rec.2020 specifies a maximum frame rate of 120 Hz. The reader is referred to Chapter 12 for a more in-depth discussion of frame rate requirements.

In terms of our discrete signal representation, we add an extra dimension (time or frame index), z, to the two spatial dimensions associated with a still image, thus:

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}[0,0,z] & \mathbf{s}[0,1,z] & \cdots & \mathbf{s}[0,X-1,z] \\ \mathbf{s}[1,0,z] & \mathbf{s}[1,1,z] & \cdots & \mathbf{s}[1,X-1,z] \\ \vdots & \vdots & & \vdots \\ \mathbf{s}[Y-1,0,z] & \mathbf{s}[Y-1,1,z] & \cdots & \mathbf{s}[Y-1,X-1,z] \end{bmatrix} \tag{4.2}$$

where (again for an RGB signal):

$$\mathbf{s}[x, y, z] = \begin{bmatrix} s_R[x, y, z] & s_G[x, y, z] & s_B[x, y, z] \end{bmatrix}$$

4.1.3 Coding units and macroblocks

As we will see later, video compression algorithms rarely process information at the scale of a picture or a pixel. Instead the coding unit is normally a square block of pixels. In standards up to and including H.264, this took the form of a 16×16 block, comprising luma and chroma information, called a macroblock. We will examine the nature of color spaces and chroma sub-sampling in Section 4.5. So for now let us simply examine the high level organization of this macroblock.

Macroblocks

A typical macroblock structure is illustrated in Figure 4.4. The macroblock shown corresponds to what is known as a 4:2:0 format (see Section 4.5) and comprises a

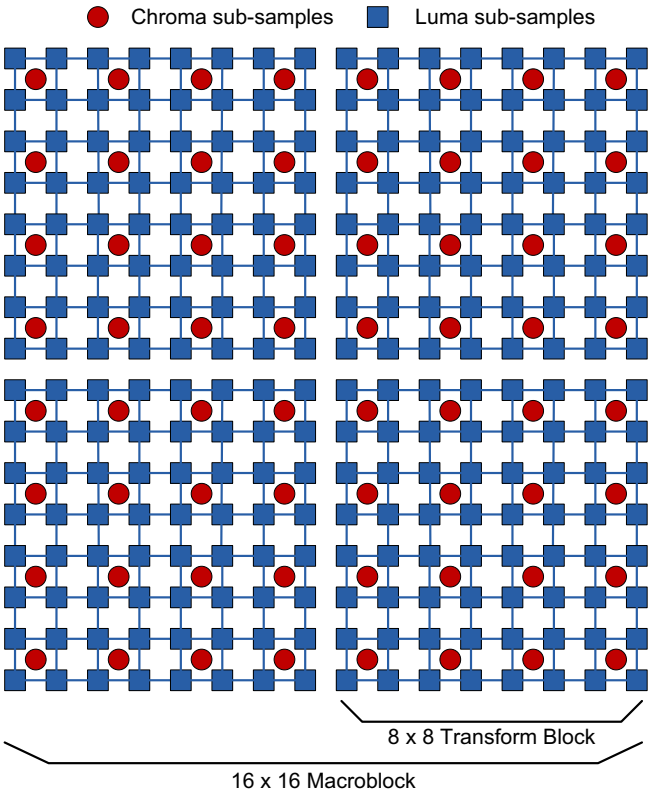

FIGURE 4.4

Typical macroblock structure.

16 × 16 array of luma samples and two sub-sampled 8 × 8 arrays of chroma (color difference) samples. This macroblock structure, when coded, must include all of the information needed to reconstruct the spatial detail. For example, this might include transform coefficients, motion vectors, quantizer information, and other information relating to further block partitioning for prediction purposes. A 16 × 16 block size is normally the base size used for motion estimation; within this, the decorrelating transforms are normally applied at either 8 × 8 or 4 × 4 levels.

Coding tree units

HEVC has extended the size of a macroblock up to 64 × 64 samples to support higher spatial resolutions, with transform sizes up to 32 × 32. It also provides much more flexibility in terms of block partitioning to support its various prediction modes. Further details on this are provided in Chapter 12.

4.1.4 Picture types and groups of pictures

Frame types

Most video coding formats support three types of picture (or frame[1]), classified in terms of their predictive properties. These are described below:

- **Intra-coded (I) pictures:** I pictures (or frames) are coded without reference to any other picture, but can be used as a reference for other pictures.
  - They provide anchor pictures and support random access.
  - They offer the highest resilience to errors as they do not propagate errors from previous pictures.
  - Because they do not exploit temporal correlation, they have the lowest compression ratio of all frame types.
- **Predicted (P) pictures:** P pictures are inter-coded, i.e. they contain the compressed difference between the current frame and a prediction of it based on a previous I or P frame.
  - Like I frames, they can be used as a reference for other frames.
  - In order to decode a P frame, the transformed residual must be accompanied by a set of motion vectors, that provide information on the prediction reference location.
  - P pictures are more efficient than I pictures as they exploit both spatial and temporal correlations.
  - Unlike I frames, they will propagate errors due to predictive coding.
- **Bi-predicted (B) pictures:** B pictures comprise the compressed difference between the current frame and a prediction of it based on up to two I, P, or B frames.

[1]These terms can generally be used interchangeably.

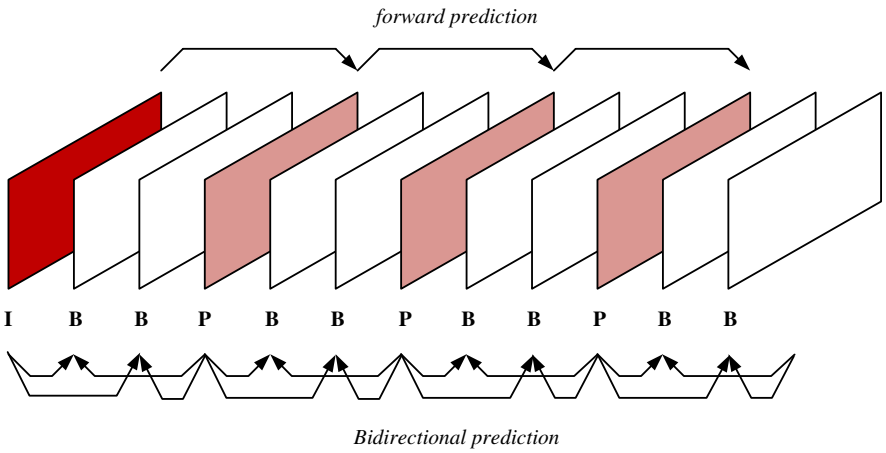

FIGURE 4.5

Typical group of pictures structure and prediction modes.

- In standards up until H.264/AVC, B pictures were not used as a reference for predicting other pictures. However, H.264/AVC and HEVC have generalized their definition and in these cases they can be used as a reference, although normally in a hierarchical fashion (see Chapter 12).
- They require accompanying motion vectors.
- They provide efficient coding due to exploitation of temporal and spatial redundancies and can be quantized more heavily if not used as a reference.
- They will not propagate errors if not used as a reference for other pictures.

Groups of pictures (GOPs)

All video coding methods and standards impose some bitstream syntax in order for the bitstream to be decodable. Part of this relates to the order of encoding, decoding, and display of pictures and the way in which pictures of different types are combined. A typical GOP structure that illustrates this (typical of standards such as MPEG-2) is shown in Figure 4.5.

Example 4.1 (Intra- vs inter-frame coding—the benefits of prediction)
Consider a 512 × 512 @30 fps color video with 8 bit samples for each color channel. Compare the intra-compressed, inter-compressed, and uncompressed bit rates if the average number of bits per pixel after compression are as follows:

- **Intra-frame mode:** 0.2 bpp (assume this is an average value that applies to all color channels).
- **Inter-frame mode:** 0.02 bpp (assume a GOP length of 100 pictures and that this value includes any overheads for motion vectors).

Solution.

(a) Uncompressed
The uncompressed bit rate can be calculated as:

$$512 \times 512 \times 30 \times 24 \simeq 189 \text{ Mb/s}$$

(b) Compressed intra-mode
In intra-mode, all frames are compressed independently to give a bit rate of:

$$512 \times 512 \times 30 \times 0.2 \times 3 \simeq 4.7 \text{ Mb/s}$$

(c) Compressed inter-mode
In inter-mode, 1 in every 100 frames are intra-frames and the other 99 are inter-frames. Hence the bit rate is:

$$512 \times 512 \times 30 \times \left(\frac{0.6 + 99 \times 0.06}{100} \right) \simeq 514 \text{ kb/s}$$

This indicates that we can achieve significant savings (the figures given here are not unrealistic in practice) through the use of inter-frame coding to exploit temporal as well as spatial correlations.

4.2 Formats and aspect ratios

We examined the range of commonly used video formats in Chapter 1, in the context of their raw bit rates and the limitations of available transmission bandwidths. The video industry has come a long way since it started coding QCIF (176×144 luma samples) images for low bit rate conferencing applications around 1989. We are now poised, with the introduction of HEVC, to be able to acquire, process, store, deliver, and display UHDTV formats, initially at a resolution of 3840×2160, but soon rising to 8k (7680×4320) for some applications. The relative sizes of these formats are illustrated in Figure 4.6, where the size drawn is normalized to the viewing angle subtended for a single pixel at the retina. The top subfigure is particularly startling in showing the relative dimensions of QCIF and UHDTV formats.

4.2.1 Aspect ratios

Table 4.1 illustrates some of the more common video format aspect ratios. The choice of aspect ratio has been the subject of much debate; choosing one closest to the ratio of viewing angles of the retina would seem the most obvious choice. These are approximately 95° temporally and roughly 70° above and below the horizontal meridian. It is not difficult to see, however, that this causes a problem—it argues that the aspect ratio should be infinite! So let's have another go—what about the foveal visual field? This approximately circular region of our retina covers 2° and 3° of our visual

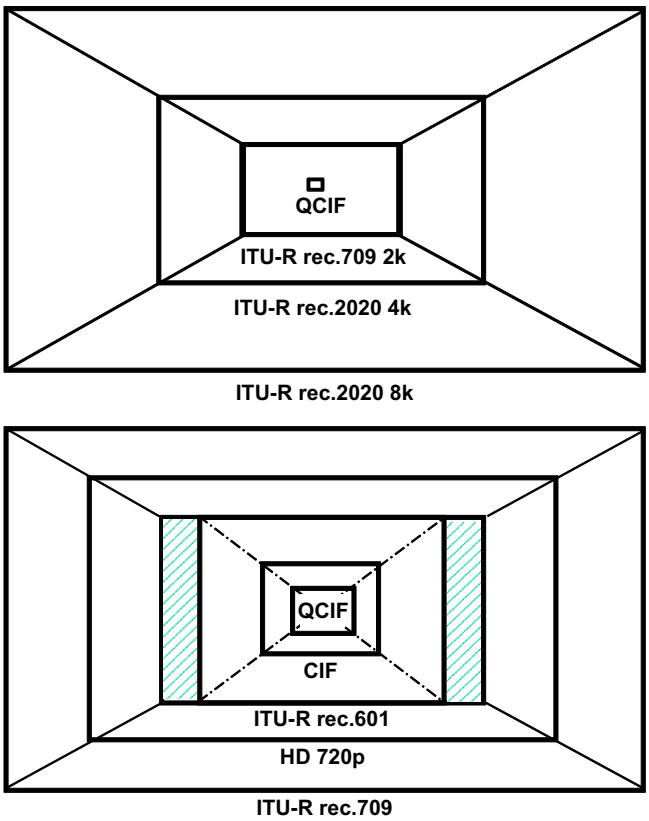

FIGURE 4.6

Aspect ratios of common formats, normalized according to resolution.

Table 4.1 Aspect ratios of film and TV.

Format	Aspect ratio
SDTV	1.33 (4:3)
WSTV/HDTV	1.78 (16:9)
Widescreen 35 mm film	1.85
70 mm film	2.10
Cinemascope anamorphic 35 mm film	2.35
Cinemascope anamorphic 35 mm (1970+)	2.39

field and visual acuity rapidly decreases away from the fovea, typically 20/20 in the fovea and 20/200 at a field angle of 20°. So because this is roughly circular, should we assume that a square aspect ratio is best or that the screen should be circular? The answer is clearly somewhere between 1:1 and infinity and, maybe, we do not

need rectangular screens! Some argue that the *golden ratio* of 1.61:1 is the most aesthetically pleasing, although the reason for this is not particularly clear.

A practical solution, certainly for larger and closer screens, is to create the impression that the screen fills our peripheral vision, without completely doing so. This provides a better sense of immersion in the content while not requiring overly wide screens—hence the convergence on typical aspect ratios of 1.77–2.39. However, none of this accounts for the changes in viewing pattern with vary large screens, where we exhibit a greater tendency to track objects with eye and head movements. This introduces all kinds of new problems that we will touch on again in Chapter 13.

In practice the 16:9 format, selected for most consumer formats, is a compromise that optimizes cropping of both film and 4:3 content. The 16:9 ratio (1.78) is very close to the geometric mean of 1.33 and 2.39, so all formats occupy the same image circle. At the end of the day, the best aspect ratio will depend a lot on composition and the relationships between the main subject, the edges of the frame, and the amount of uninteresting space. The art of the cinematographer is to compose the content in a manner that is compelling, regardless of frame aspect ratio.

Figure 4.7 gives an example of some common aspect ratios superimposed on a scene. The preference, of course, is also dictated by the viewing conditions. Within the constraints of comfortable viewing, comparing 4:3 and 16:9 formats based on identical width, height, or area, most subjects prefer the format that provides the greatest viewing area.

For the case of a sensor or display with N_p pixels and an aspect ratio, r, it is straightforward to compute the horizontal and vertical pixel counts, X and Y, as follows:

$$X = \sqrt{rN_p}$$
$$Y = \sqrt{\frac{N_p}{r}} \tag{4.3}$$

FIGURE 4.7
Widescreen formats.

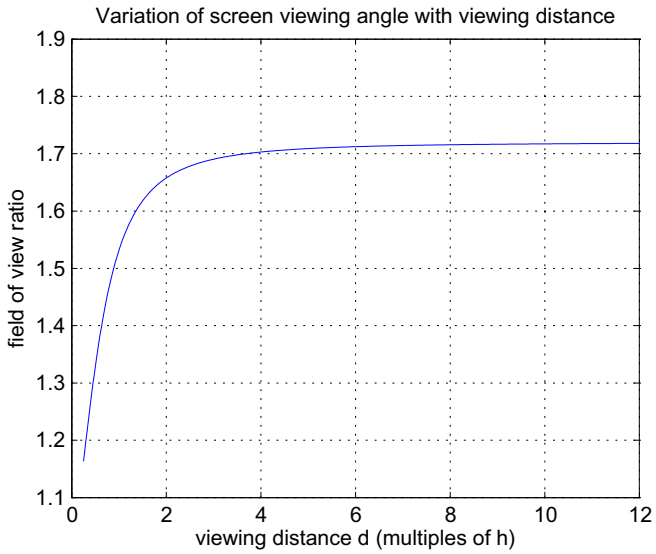

FIGURE 4.8

Variation of field of view with viewing distance (aspect ratio = 16:9 here).

Field of view ratio

Let us consider the viewing angle ratio (or field of view ratio) rather than the aspect ratio. It can easily be shown that this is independent of screen size, s, but will vary non-linearly with viewing distance, d. The commonly adopted TV and computer monitor format, 16:9, is used as an example in Figure 4.8, but other aspect ratios exhibit similar characteristics. This figure plots the viewing angle ratio against the viewing distance in multiples of screen height H. A number of important things can be taken from this graph. Firstly, as d increases, the viewing angle ratio $\theta_w/\theta_h \to r$ where r is the aspect ratio of the screen. Secondly, as θ_w/θ_h decreases below $3H$ there is a rapid fall-off in this ratio down to unity at $d = 0$. For a 16:9 aspect ratio, the viewing distance of $1.5H$ (specified for UHDTV) corresponds to a ratio of approximately 1.65, actually quite close to the golden ratio mentioned above! Example 4.2 examines this in more detail.

Example 4.2 (Field of view ratio)

Derive an expression for the field of view ratio for a screen of size s (diagonal dimension), with an aspect ratio, r, viewed at a distance, d.

Solution. Frequently the dimensions of a screen are given in terms of its diagonal dimension, s. It is a simple matter to calculate the screen height H and width W, using equation (4.4):

$$H = \frac{s}{\sqrt{1+r^2}}; \quad W = \frac{s}{\sqrt{1+1/r^2}} \tag{4.4}$$

The horizontal and vertical viewing angles that correspond to a screen of aspect ratio $r(W \times H)$ viewed at a distance d are then:

$$\theta_w = 2 \arctan\left(\frac{W}{2d}\right) \tag{4.5}$$

$$\theta_h = 2 \arctan\left(\frac{H}{2d}\right) \tag{4.6}$$

and their ratio, defined here as the field of view ratio, Ψ, is thus:

$$\Psi = \frac{\arctan\left(\frac{W}{2d}\right)}{\arctan\left(\frac{H}{2d}\right)} \tag{4.7}$$

This is plotted for a 16:9 aspect ratio screen for various viewing distances in Figure 4.8.

4.2.2 Displaying different formats

The existence of multiple formats (not to mention 3-D, 2-D, etc.) creates significant extra work for content producers, especially film makers. It is quite common for films or older TV programmes to be cropped or distorted in some way and in general this is annoying for the consumer who feels that they are not getting the most out of their screen!

For example, displaying a 2.39:1 format on a 16:9 screen produces a *letter box* appearance. This can be avoided by distorting the original to fit or by zooming, but both have their problems. The other way around, for example a 4:3 format displayed on a 16:9 screen, produces a *pillar box* appearance. Again this can be stretched to produce short and fat subjects or zoomed to lose edge content.

Pan and scan and Active Format Description

Active Format Description (AFD) is a mechanism used in digital broadcasting to avoid the problems described above. This is a means of signaling the active content of a picture to enable intelligent processing of the picture at the receiver, in order to map the format to the display in the best way. Considering, for example, Figure 4.7, if we wish to watch the 2.39:1 version on a 16:9 display, the blue box (representing the 16:9 picture) does not need to stay in the center of the 2.39:1 picture. It could instead move left or right to frame the action in the best way for the smaller screen.

This capability was introduced as a non-normative feature in MPEG-2 on a GOP basis and is also supported by newer standards (e.g. through SEI data in H.264/AVC). AFD codes include information about the location of the active video and about any protected area which must be displayed. The picture edges outside of this area can be discarded without major impact on the viewing experience.

4.3 Picture scanning

4.3.1 Interlaced vs progressive scanning

Interlaced scanning is an attempt to trade-off bandwidth, flicker, and resolution, by constructing each frame from two consecutive fields. Fields are sampled at different times such that consecutive lines belong to alternate fields; in the US a 60 Hz format is adopted, whereas in Europe a 50 Hz format is employed.

Early analog television systems needed to minimize bandwidth while maintaining flickerless viewing. Interlacing was devised as a compromise between a high temporal update rate (a field rate at 50 or 60 Hz) which reduces flicker for most content, and lower effective bandwidth (due to the 25 or 30 Hz frame rate). The persistence of the phosphor in early CRT displays, coupled with the persistence of human vision, meant that, provided the vertical dimension of a picture is scanned rapidly enough, the HVS does not see the effect of alternating lines and experiences continuous motion. An illustration of the difference between interlaced and progressive scanning is given in Figure 4.9. In modern digital cameras and displays the scanning, as shown in Figure 4.9, is no longer performed by deflecting an electron beam. The row-wise and field-wise scanning pattern is however the same as indicated in the figure.

Notation. The notation we will adopt here is that used by the European Broadcasting Union (EBU) and SMPTE:

$$Y\pi/f$$

where $Y =$ the number of vertical lines in the frame; $\pi =$ p (progressive) or i (interlaced); and $f =$ frame rate.

For example: 1080p/50 represents an HD progressive format with spatial resolution 1920×1080 and temporal sampling at 50 fps. Similarly 1080i/30 represents the same spatial resolution, but this time interlaced at 30 and 60 fps.

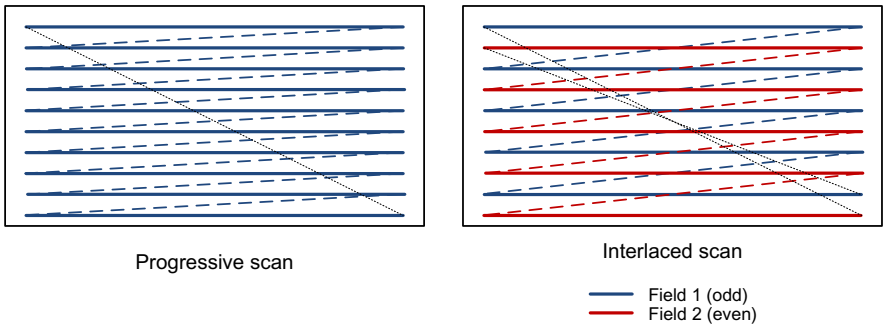

FIGURE 4.9

Interlaced vs progressive frame scanning.

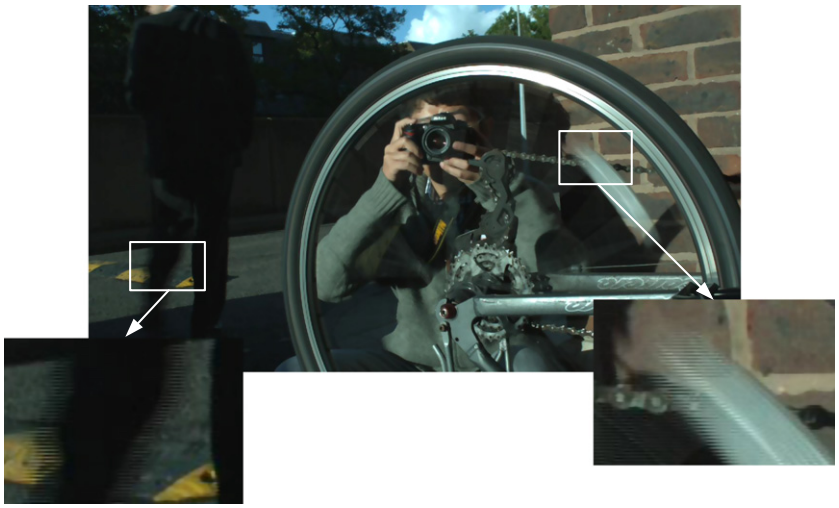

FIGURE 4.10

Example of effects of interlaced scanning with poor deinterlacing.

Problems caused by interlacing

The effects of interlacing become apparent in the presence of fast motion as illustrated by the frame shown in Figure 4.10. The inset shows the *combing* artifacts that characterize this format, due to the temporal offset between the two constituent fields. Interlace also introduces a second problem of aliasing, commonly referred to as *twitter*. This occurs where information changes vertically at a rate similar to the field scanning rate. This becomes particularly prevalent when the vertical resolution of the content is similar to the resolution of the format. For example, the clothes of a person wearing fine horizontal stripes will appear to twitter. Filtering is often applied to reduce this effect. An example of twitter can be observed in Ref. [2].

It is finally worth noting that the bandwidth advantages of interlacing associated with analog systems is reduced in the presence of digital video compression, especially if a frame with two fields is processed as one. MPEG-2 and subsequent standards have introduced an adaptive field-frame mode in an attempt to improve this situation.

The artifacts associated with interlacing can be minimized if the receiver and display employ good deinterlacing methods [5]. These employ locally adaptive filtering to the two fields as part of the combining process.

4.3.2 Standards conversion

3:2 pull-down

The need to convert between film and TV formats has been a requirement since the beginnings of public television broadcasting. Film and digital cinema content is

traditionally captured at 24 fps using progressive scanning. This needs to be converted to 29.97 fps in the US based on their 60 Hz interlaced scanning and 25 fps in Europe based on 50 Hz. For the case of 25 fps it is normal for the film to be simply speeded up by a factor of 25/24. This produces some audiovisual changes but these are not generally noticeable. For motion to be accurately rendered at 29.97 fps, however, a telecine device must be employed to resample the 24 fps signal. A technique known as 3:2 pull-down is normally employed.

3:2 pull-down uses the approach described in Algorithm 4.1 and it is illustrated in Figure 4.11.

Some streaming sites, such as Netflix, have strict requirements for their content and require an inverse telecine operation to detect and remove 3:2 pull-down from telecine video sources, thereby reconstructing the original 24 frames per second film format. This improves the quality for compatibility with non-interlaced displays and eliminates some of the redundant data prior to compression.

Algorithm 4.1 3:2 pull-down.

1. Play the film at 23.976 frames/s;
2. Split each cinema frame into video fields. At 23.976 frames/s, there are four frames of film for every five frames of 60 Hz video: i.e. $23.976/29.97 = 4/5$;
3. Convert these four frames to five by alternately placing the first film frame across two fields, the next across three, and so on.

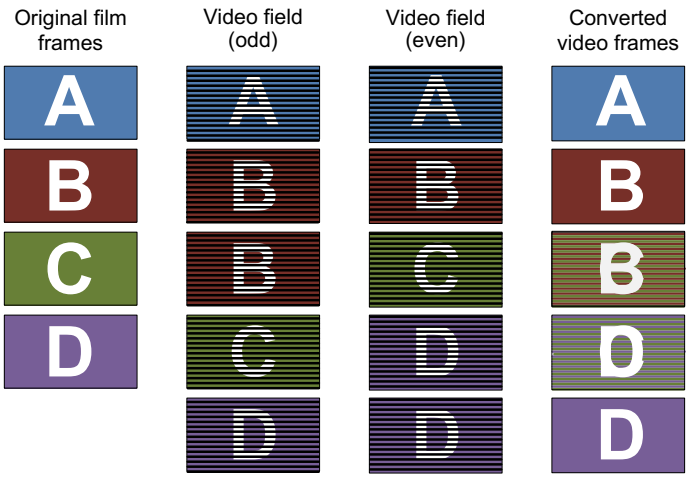

FIGURE 4.11

3:2 pull-down example.

Motion compensated standards conversion

More general standards or format conversion is provided by motion compensation methods such as those based on phase correlation (see Chapter 6). These are required to convert content between interlaced and progressive formats, between HD and SD and other formats, and between formats of different frame rates and color spaces. Phase correlation is a Fourier-based method that exploits duality between displacement in the time (or space) domain and phase shift in the frequency domain. It offers better immunity to noise and luminance changes than conventional block matching methods. Subpixel accuracy can easily be achieved, as is required for the generation of new frames in the target format.

4.4 Gamma correction

Gamma correction [5] is used to correct the differences between the way a camera captures content, the way a display displays content and the way our visual system processes light. Our eyes do not respond to light in the same way that a camera captures it. In basic terms, if twice the number of photons hits a digital image sensor, then the output voltage will be twice as big. For the case of older CRT-based cameras this was not, however, the case and there was a highly non-linear relationship between light intensity and output voltage. The HVS also has a very non-linear response to luminance levels, being more sensitive to small changes in darker areas and much less sensitive in light areas (recall Weber's law from Chapter 2). Without correction, to avoid banding effects due to perceptible jumps between coding levels, around 11 bits

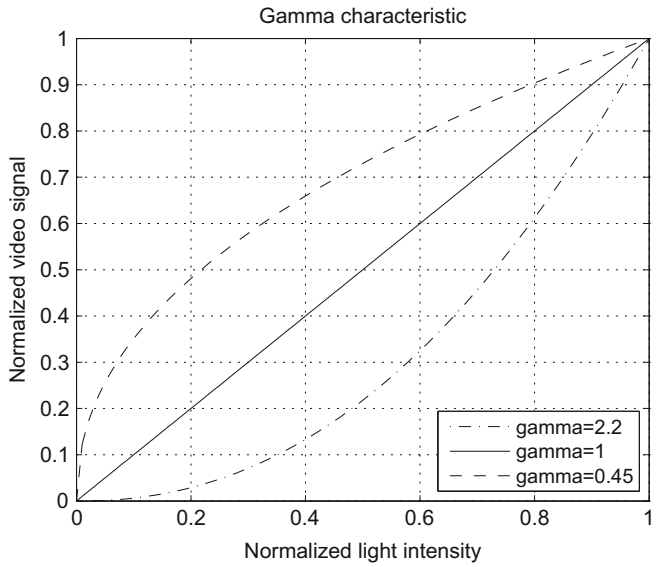

FIGURE 4.12

Gamma curves for $\gamma = \{0.45, 1, 2.2\}$.

would be needed. However, over most of the higher end of the scale, the coding would be highly redundant since coding levels would be imperceptible. With gamma pre-correction we can get an acceptable representation with only 8 bits.

Photopic human vision characteristics can be approximated by a power law function. Correction of the signal output from the camera using this type of function ensures that not too many bits are allocated to brighter regions where humans are less discriminating and more to darker regions, where they are more so. Video cameras therefore normally perform a non-linear mapping of illumination intensity to output value, providing a more uniform perceptual scale with finer increments for darker values. This is known as the Gamma Characteristic of the device as is described by the transfer function:

$$V = c_1 \Phi^{\gamma} + c_2 \tag{4.8}$$

where Φ is the luminous flux (normalized) and V is the value (voltage) of the signal, c_1 is the camera sensitivity and c_2 is an offset value. Figure 4.12 shows transfer characteristic plots for three gamma values, $\gamma = \{0.45, 1, 2.2\}$. These values are typical of those used in modern TV systems. Figure 4.13 shows some corrected examples of an image at various gamma levels.

FIGURE 4.13

Examples of gamma correction.

It should be noted that modern formats such as Rec.709 use a piecewise transfer characteristic from the luminance to the output. This is linear at the lower region and then changes to the power law for the rest of the range. For Rec.709 this is:

$$V = \begin{cases} 4.5\Phi & \Phi < 0.018 \\ 1.099\Phi^{0.45} - 0.099 & \Phi \geq 0.018 \end{cases} \tag{4.9}$$

Although a topic of some confusion, it is becoming accepted to refer to the gamma-corrected luminance signal as *luma*.

4.5 Color spaces and color transformations

4.5.1 Color descriptions and the HVS

Trichromacy theory

As we have examined in Chapter 2, the cone cells in the retina are the main photoreceptors for normal photopic conditions and have sensitivities peaking around short (S: 420–440 nm), middle (M: 530–540 nm), and long (L: 560–580 nm) wavelengths. (The low light sensors (rods) peak around 490–495 nm.) The sensation of color in humans is the result of electromagnetic radiation in the range 400–700 nm, incident on the retina.

One of the challenges when color television was introduced was the trade-off between delivering realistic colors to the screen vs conserving bandwidth. In simple terms, the transmission of color information (assuming the use of three color channels (RGB)) occupies three times the bandwidth of a monochrome transmission. The solution to this for analog TV was (inspired by the trichromatic theory of vision (see Chapter 2)) to encode the color information separately from the luma information, with significantly reduced resolution in order to conserve bandwidth. This had the advantage that the luma signal was backwards compatible with older monochrome TV sets while, for color sets, the higher resolution luma and the lower resolution color signals combined in the HVS to produce a perceptually high-resolution color image. As technology has progressed since the introduction of analog color television, there has been substantial effort invested in developing new color space representations appropriate for modern digital encodings, formats and displays. Some of these are reviewed below. For further information, the reader is referred to Ref. [1] and the very good introduction to color spaces presented by Poynton [3,5].

Much of colorimetry is based on the tri stimulus theory of color—i.e. that any color can be formed from three primaries so long as they are orthogonal, but also on Grassman's laws which reflect the linearity and additivity rules of color. These state that the color match between any two stimuli is constant when the intensities of the two stimuli are increased or decreased by the same factor. For example, if we have two sources, C_1 and C_2, then if:

$$\begin{aligned} C_1 &= R_{C1}(R) + G_{C1}(G) + B_{C1}(B) \\ C_2 &= R_{C2}(R) + G_{C2}(G) + B_{C2}(B) \quad \text{and} \\ C_3 &= C_1 + C_2 \end{aligned} \tag{4.10}$$

then:

$$C_3 = [R_{C1} + R_{C2}](R) + [G_{C1} + G_{C2}](G) + [B_{C1} + B_{C2}](B) \tag{4.11}$$

Color spaces

We saw in Chapter 2 that the CIE has defined the coordinates of color spaces, based on a trichromatic representation, which describes the set of color sensations experienced by the human observer. The first of these, CIE 1931 (developed in 1931), was based on the concept of a standard observer, matching the observations from a large number of (normal) subjects to controlled stimuli. These stimuli were based on the primaries:

$$[\,R_0 \quad G_0 \quad B_0\,] = [\,700 \quad 546.1 \quad 435.8\,]\ \text{nm}$$

Color space transformations

According to the basic principles of colorimetry, various other linear combinations of the RGB stimulus values can be derived that might offer better properties than the original values. CIE introduced an alternative reference system (or color space), referred to as XYZ, which could provide a more flexible basis for representing tristimulus values. A color space maps a range of physically produced colors (e.g. from mixed lights) to an objective description of those color sensations based on a trichromatic additive color model, although not usually the LMS space. CIE XYZ can represent all the color sensations that an average person can experience and is a transformation of the $[R_0, G_0, B_0]$ space. The CIE have defined this transform as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2.365 & -0.515 & 0.005 \\ -0.897 & 1.426 & -0.014 \\ -0.468 & 0.089 & 1.009 \end{bmatrix} \begin{bmatrix} R_0 \\ G_0 \\ B_0 \end{bmatrix} \tag{4.12}$$

CIE defined XYZ such that the Y component corresponds to the luminance signal, normalized to equal energy white. Also, unlike some RGB systems where contributions have to be negative to match a color, the XYZ system is wholly positive. However, the problem with this is that it causes the XYZ values to be complex numbers, i.e. they cannot be physically realized by an actual stimulus.

Chromaticity values can also be defined in the XYZ system as follows:

$$\begin{aligned} x &= \frac{X}{X+Y+Z} \\ y &= \frac{Y}{X+Y+Z} \\ z &= \frac{Z}{X+Y+Z} \end{aligned} \tag{4.13}$$

Large x values correspond to red or orange hues, y values correspond to green, blue-green or yellow-green, and large z corresponds to large blue, violet or purple hues. Because $x + y + z = 1$, only two of these values are actually required, so normally this is expressed as an Yxy system.

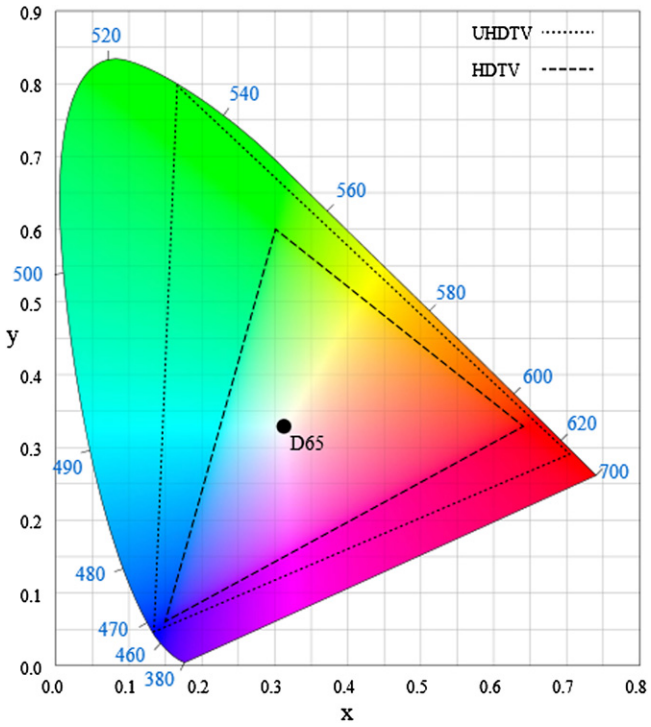

FIGURE 4.14
CIE 1931 chromaticity diagram. (Public domain image from: http://commons.wikimedia.org/wiki/File:CIExy1931_Rec_2020_and_Rec_709.svg.)

Chromaticity diagrams

The chromaticity coordinates can be expressed on a 2-D chromaticity diagram like that shown in Figure 4.14. This diagram shows all the hues perceivable by the standard observer for various (x, y) pairs and indicates the spectral wavelengths of the dominant single frequency colors. The D65 point shown on this diagram is the white point, which in this case corresponds to a color temperature of 6500K (representative of average daylight).

CIE XYZ or CIE 1931 is a standard reference which has been used as the basis for defining most other color spaces. This maps human color perception based on the two CIE parameters x and y. Although revised in 1960 and 1976, CIE 1931 is still the most commonly used reference.

If we consider again the CIE 1931 color space in Figure 4.14, this also shows two triangles that correspond to the colors representable by the Rec.709 (HDTV) and the Rec.2020 (UHDTV) color spaces. It is interesting to note that the latter can represent many colors that cannot be shown with Rec.709. The Rec.2020 primaries correspond to 630 nm (R), 532 nm (G), and 467 nm (B). The Rec.2020 space covers approximately 76% of the CIE 1931 space whereas Rec.709 covers only 36%, thus offering better coverage of more saturated colors.

Color spaces for analog TV

In almost all cases, a format YC_1C_2 is used for analog and digital formats, where Y represents signal luminance. This is done for the two reasons provided above:

- Compatibility with monochrome displays.
- Possible bandwidth reduction on the chrominance signals with no loss of perceptual quality.

For example, in the case of US NTSC systems for color TV, the primaries are defined in terms of CIE xyz coordinates as follows:

$$\mathbf{m} = \begin{array}{c} \\ R: \\ G: \\ B: \end{array} \begin{array}{ccc} x & y & z \\ 0.67 & 0.33 & 0.00 \\ 0.21 & 0.71 & 0.08 \\ 0.14 & 0.08 & 0.78 \end{array} \tag{4.14}$$

The actual values used in the TV transmission format are those after gamma correction. Consequently, after various normalizations and rotations [1] the resulting color space, defined as YIQ, is given by the following transformation (where R', G', B represent the gamma corrected camera output signals):

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{4.15}$$

A similar mapping for PAL systems in Europe is given by:

$$\begin{bmatrix} Y \\ U_t \\ V_t \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & -0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{4.16}$$

Color spaces for digital formats

With the introduction of digital formats, starting with SDTV (Rec.601), there was a need to adjust the color space transformation to deal with these digital formats and finite wordlengths. The components again are referenced against the gamma corrected camera outputs and are quantized to 8 bits. The representation for the ITU-R Rec.601 system is known as YC_bC_r and is given by the following transformation from corrected 8-bit RGB signals in the range 0 to 255 (N.B. Studio $R'G'B'$ signals are in the range 17 to 235 and require a modified conversion matrix):

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{4.17}$$

In order to provide coding headroom, the full range of 256 values is not used. Instead a range of 220 levels from 16 to 235 are employed. It should be noted that some authors add a prime to the Y (luma) component above to differentiate it from the luminance signal, because it results after non-linear encoding based on gamma

corrected RGB primaries. C_b and C_r are the blue-difference and red-difference chroma components.

Similarly the ITU-R Rec.709 system for HDTV formats is given by:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.183 & 0.614 & 0.062 \\ -0.101 & -0.338 & 0.439 \\ 0.439 & -0.399 & -0.040 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{4.18}$$

Consider as an example the image of Thala Beach shown in Figure 4.15. This displays the Y, C_b, and C_r components. Several things can be observed from this—for example, the white clouds are represented as middle values in both chroma components, the blue sky is strong in the blue channel; the sand and browner aspects of the tree show strongly in the red channel. Furthermore, the two chroma components appear fuzzy to the human eye relative to the luma channel and this provides some evidence that we could downsample them without loss of overall perceived detail.

FIGURE 4.15
Color difference images using the YC_bC_r decomposition. Top left: original. Right: luma channel. Bottom left: C_b channel. Right: C_r channel.

4.5.2 Sub-sampled color spaces

Most capture devices produce an RGB output, where each component requires high precision and high resolution to provide perceptually high quality images. As discussed above, this format is rarely used for transmission as it requires three times the bandwidth of a monochrome channel. Hence, as we have seen, a color space transformation is applied to produce a luminance signal Y and two color difference signals, C_b and C_r. The motivations for this are:

1. **Chroma sub-sampling:** As we have discussed in Chapter 2 the human visual system has a relatively poor response to color information. The effective resolution of our color vision is about half that of luminance information. Separation of luma and chroma components thus means that we can sub-sample the chroma by a factor of 2 without affecting the luma resolution or the perceived image quality. This provides an immediate compression benefit as we are reducing the bit rate by 50% before we start compression proper!
2. **Luma processing:** As well as having compatibility with monochrome displays, the availability of an independent luma signal is useful during compression. For example, we normally perform motion estimation only on the luma component and use the resulting vectors to compensate both luma and chroma components. We also often base objective quality assessments only on the luma component.

Chroma sub-sampling

The human eye has a relatively poor response to chroma spatial detail. Chroma components are thus normally sub-sampled (usually by a factor of 2) to reduce bandwidth requirements. The most commonly used sub-sampling patterns are shown in Figure 4.16. This shows the relationships between luma and chroma samples for a range of common sub-sampling formats. These are:

- **4:4:4**—This is the full resolution image with luma and chroma samples at all pixel locations. It is used when there is a requirement for the highest quality studio work.
- **4:2:2**—This is the conventional format for studio production. Most professional cameras output video in a 4:2:2 format. It provides sub-sampling horizontally but not vertically. This is advantageous in the case of interlaced scanning as it eliminates problems with color degradation.

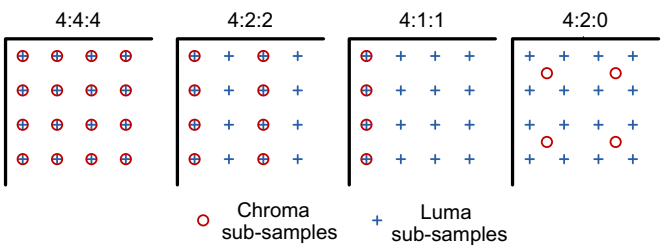

FIGURE 4.16

Common chroma sub-sampling formats.

- **4:2:0**—This system is the most common format for broadcast delivery, internet streaming and consumer devices. It sub-samples the chroma signals by a factor of 2 in both horizontal and vertical directions, maximizing the exploitation of perceptual color redundancy.
- **4:1:1**—This system was used in DV format cameras. It sub-samples by a factor of 4 horizontally but not vertically. Its quality is inferior to 4:2:0 formats, so it is rarely used now.
- **4:0:0**—This is not shown in the figure but represents the simple case of a luma only or monochrome image.

The question that arises is how are the chroma signals downsampled? The answer is by pre-processing with a linear phase FIR filter designed to avoid aliasing. Similarly signals are upsampled for reconstruction on an image to display using an appropriate linear phase FIR interpolation filter. It should be noted that the sub-sampling filter will dictate where the chroma samples are sited relative to the luma samples.

Notation 4:*X*:*Y*. This notation is confusing to most people and there have been several attempts to fit rules that explain its meaning, all of which fail under some circumstances. The reason for this is that, historically, the 4:*X*:*Y* format was devised at the time of PAL and NTSC systems when subcarrier-locked sampling was being considered for component video. The first digit represented the luma horizontal sampling reference as a multiple of $3\frac{3}{8}$ MHz. The second number *X* represented the horizontal sampling factor for chroma information relative to the first digit. The third number, *Y*, was similar to *X*, but in the vertical direction.

It does not take a lot of thinking to realize that the 4:2:0 and 4:1:1 formats do not comply with this definition. So people have started thinking up other rules that fit. This author's view is that we should just use them as labels for the relevant format. However, if you need a rule, then the best one is as follows (not very elegant but it sort of works for most formats):

1. Assume a 4×2 block of pixels, with 4 luma samples in the top row.
2. *X* is the number of chroma samples in the top row.
3. *Y* is the number of additional chroma samples in the second row.

An issue of course is that with some 4:2:0 formats (as shown in Figure 4.16), the chroma samples are not actually co-sited with the luma samples.

Example 4.3 (Color space transformations in ITU-R Rec.601)
Consider the YC_bC_r format used with ITU-R Rec.601. Compute the component values for the following cases:

(a) $R'G'B' = [0\ 0\ 0]$.
(b) $R'G'B' = [220\ 0\ 0]$.
(c) $R'G'B' = [0\ 220\ 0]$.

Solution. Using equation (4.17), we can compute these values as follows:

(a) $YC_bC_r = [16\ 128\ 128]$

A signal at the base black level in terms of luminance and at central and equal chroma levels:

(b) $YC_bC_r = [73\ 95\ 224]$

A signal that is higher in the red color difference channel with mid levels of luminance and blue channel:

(c) $YC_bC_r = [127\ 64\ 47]$

Notice the difference between this and (b). The green channel is weighted more heavily as it contributes more to our sense of brightness.

Example 4.4 (Chroma sub-sampling and compression)
If a color movie of 120 min duration is represented using ITU-R.601 (720 × 576 @30 fps@8 bits, 4:2:0 format):

(a) What hard disk capacity would be required to store the whole movie?
(b) If the movie is MPEG-2 encoded at a compression ratio CR = 50:1 and transmitted over a satellite link with 50% channel coding overhead, what is the total bit rate required for the video signal?

Solution.

(a) The bit rate is given by:

$$720 \times 576 \times 30 \times 8 \times \frac{3}{2} = 149{,}299{,}200 \text{ b/s} = 149 \text{ Mb/s}$$

For a 120 min movie, the total storage space required is:

$$120 \times 60 \times 149{,}299{,}200 = 1{,}074{,}954{,}240{,}000 \text{ bits} = 134 \text{ GB}$$

(b) With a compression ratio of 50:1, the bit rate would be approximately 3 Mbps. With 50% coding overhead, this represents a total bit rate of 4.5 Mbps.

4.5.3 Bayer filtering

Professional cameras often have three CCD or CMOS sensors, one for each color channel. An alternative approach is to use a single sensor in conjunction with a Color Filter Array. The color filter array (often referred to as a Bayer filter or Bayer mosaic) is a pattern of color filters applied to a single image sensor in order to enable it

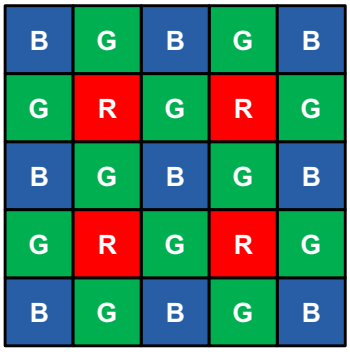

FIGURE 4.17

Bayer filter mosaic.

to produce R, G, and B samples. This approach is used in most single-chip digital image sensors and increasingly in professional equipment. A Bayer pattern array [4] is shown in Figure 4.17. Half of the filter elements are green and the remainder are split between blue and red. This approximates human photopic vision where the M and L cones combine to produce a bias in the green spectral region.

Obviously the effective sensor resolution of the Bayer filtered array for color signals is reduced compared to a three or four sensor array, and the degree of degradation is largely influenced by the demosaicing algorithm used. As a rule of thumb, according to tests on actual resolutions, you can roughly divide the number of pixels in the array by two, so a 10 Mpixel array effectively becomes a 5 Mpixel array after demosaicing.

There are many classes of demosaicing algorithm and we will not consider them in detail here. These range from relatively simple bilinear interpolation, to bicubic, or spline interpolation. These methods are however prone to artifacts and more sophisticated methods, for example based on the exploitation of local pixel statistics, or gradient estimation, can provide improvements. More advanced methods based on superresolution techniques have also been proposed to address aliasing issues.

4.6 Measuring and comparing picture quality

We introduce some basic topics in image and video quality assessment here. An excellent introduction to video quality metrics and subjective assessment is provided by Winkler [6]. We also discuss these topics in much more detail in Chapter 10.

4.6.1 Compression ratio and bit rate

A very basic measure of compression performance is the compression ratio. This is simply defined as:

$$C_1 = \frac{\text{No. bits in original video}}{\text{No. bits in compressed video}} \quad \text{(unitless)} \tag{4.19}$$

This provides an absolute ratio of the compression performance and only if used in the context of similar content, similar sized images, and for similar target qualities, can it be used as a comparator—otherwise it should not be. A commonly employed alternative, used for still images, is the normalized bit rate:

$$C_2 = \frac{\text{No. bits in compressed video}}{\text{No. pixels in original video}} \quad \text{(bits per pixel)} \tag{4.20}$$

Again this will vary according to content type and image resolution but also in terms of the original signal wordlength. It is just a rate indicator and provides no assessment of quality. A commonly used parameter for video is the actual bit rate:

$$C_3 = \frac{\text{No. bits in compressed video}}{\text{No. frames in original video}} \times \text{frame rate} \quad \text{(bits per second)} \tag{4.21}$$

Again this gives no information on the quality of the compressed signal, but it does provide us with an absolute value of how much channel capacity is taken up by the signal coded in a given manner.

In order for all of these measures to be useful they need to be combined with an associated measure of image or video quality. These are considered below.

4.6.2 Objective distortion and quality metrics

Mean square error (MSE)

Consider an image $\mathbf{S} = s[x, y]\colon\ x = 0, \ldots, X-1;\ y = 0, \ldots, Y-1$ and its reconstructed version $\tilde{\mathbf{S}}$. If the area of the image (or equivalently for a monochrome image, the number of samples in it) is $A = XY$ then the MSE is given by:

$$\text{MSE} = \frac{1}{A}\left(\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}(s[x, y] - \tilde{s}[x, y])^2\right) = \frac{1}{A}\left(\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}(e[x, y])^2\right) \tag{4.22}$$

A signal to noise ratio is also common in many communications applications:

$$\text{SNR} = 10 \cdot \log\left(\frac{E\left\{\left(s[x, y] - \mu_s\right)^2\right\}}{E\left\{\left(s[x, y] - \tilde{s}[x, y]\right)^2\right\}}\right) = 20 \cdot \log\left(\frac{\sigma_s}{\sigma_e}\right) \text{ dB} \tag{4.23}$$

If we extend the MSE calculation to a video sequence with K frames then the MSE is given by:

$$\text{MSE} = \frac{1}{KA}\left(\sum_{z=0}^{K-1}\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}(s[x, y, z] - \tilde{s}[x, y, z])^2\right) \tag{4.24}$$

Peak signal to noise ratio (PSNR)

Mean Square Error-based metrics are in common use as objective measures of distortion, due mainly to their ease of calculation. An image with a higher MSE will generally express more visible distortions than one with a low MSE. In image and video compression, it is however common practice to use Peak Signal to Noise Ratio (PSNR) rather than MSE to characterize reconstructed image quality. PSNR is of particular use if images are being compared with different dynamic ranges and it is employed for a number of different reasons:

1. MSE values will take on different meanings if the wordlength of the signal samples changes.
2. Unlike many natural signals, the mean of an image or video frame is not normally zero and indeed will vary from frame to frame.
3. PSNR normalizes MSE with respect to the peak signal value rather than signal variance, and in doing so enables direct comparison between the results from different codecs or systems.
4. PSNR can never be less than zero.

The PSNR for our image **S** is given by:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{A s_{\max}^2}{\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}(e[x,\,y])^2}\right) \tag{4.25}$$

where for a wordlength B: $s_{\max} = 2^B - 1$. We can also include a binary mask, $b[x,\,y]$, so that the PSNR value for a specific arbitrary image region can be calculated. Thus:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{s_{\max}^2 \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} b[x,\,y]}{\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}(e[x,\,y])^2}\right) \tag{4.26}$$

Although there is no perceptual basis for PSNR, it does fit reasonably well with subjective assessments, especially in cases where algorithms are compared that produce similar types of artifact. It remains the most commonly used objective distortion metric, because of its mathematical tractability and because of the lack of any widely accepted alternative. It is worth noting, however, that PSNR can be deceptive, for example when:

1. There is a phase shift in the reconstructed signal. Even tiny phase shifts that the human observer would not notice will produce significant changes.
2. There is visual masking in the coding process that provides perceptually high quality by hiding distortions in regions where they are less noticeable.
3. Errors persist over time. A single small error in a single frame may not be noticeable, yet it could be annoying if it persists over many frames.
4. Comparisons are being made between different coding strategies (e.g. synthesis based vs block transform based (see Chapters 10 and 13)).

Consider the images in Figure 4.18. All of these have the same PSNR value (16.5 dB), but most people would agree that they do not exhibit the same perceptual qualities. In particular, the bottom right image with a small spatial shift is practically

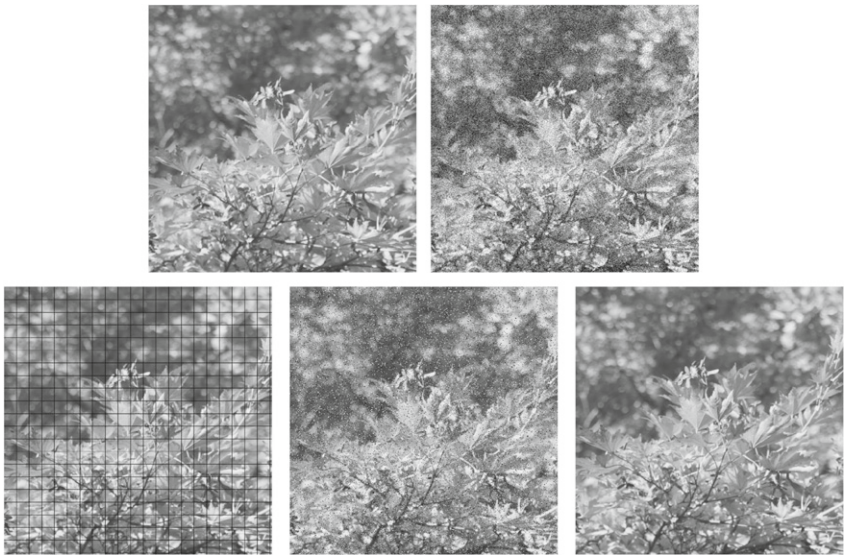

FIGURE 4.18
Quality comparisons for the same PSNR (16.5 dB). Top left to bottom right: original; AWGN (variance 0.24); grid lines; salt and pepper noise; spatial shift by 5 pixels vertically and horizontally.

indistinguishable from the original. In cases such as this, perception-based metrics can offer closer correlation with subjective opinions and these are considered in more detail in Chapter 10.

Example 4.5 (Calculating PSNR)
Consider a 3×3 image $\mathbf{S}$ of 8-bit values and its approximation after compression and reconstruction, $\tilde{\mathbf{S}}$. Calculate the MSE and the PSNR of the reconstructed block.

1	2	3
4	5	6
7	8	9

$\mathbf{S}$

1	2	2
4	4	8
7	8	8

$\tilde{\mathbf{S}}$

Solution. We can see that $X = 3, Y = 3, s_{\max} = 255$, and $A = 9$. Using equation (4.22) the MSE is given by:

$$\text{MSE} = \frac{7}{9}$$

and from equation (4.25), it can be seen that the PSNR for this block is given by:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{9 \times 255^2}{7}\right) = 49.2 \text{ dB}$$

PSNR for color images and for video

For color images with three *RGB* values per pixel, PSNR can be calculated as above, except that the MSE is now the sum over all squared value differences divided by image size and by three. Alternatively, the PSNR could be computed separately for each color component. However, because the human eye is not equally sensitive to all channels, it is common to just compute the luma PSNR. This can easily be performed when using a format such as YC_bC_r, where the luma component represents a weighted average of the color channels.

Furthermore, it is important to note that the PSNR calculation for a video signal, based on averaging PSNR results for individual frames, is not the same as computing the average MSE for all frames in the sequence and then computing the PSNR. The former biases algorithms that produce SNR fluctuations, so normally the latter is used.

Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, where higher is better. Values over 40 dB are normally considered very good and those below 20 dB are normally unacceptable. When the two images are identical, the MSE will be zero, resulting in an infinite PSNR.

Mean absolute difference (MAD)

Mean square error-based metrics, although relatively simple to implement, still require one multiplication and two additions per sample. In order to reduce complexity further, especially in some highly search-intensive operations such as motion estimation, other simpler metrics have been employed. One such metric is Mean Absolute Difference (MAD). The MAD between two images or regions, s_1 and s_2, is defined as:[2]

$$\text{MAD} = \frac{1}{A}\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}|s_1[x, y] - s_2[x, y]| \tag{4.27}$$

where A is defined as previously.

4.6.3 Subjective assessment

Mathematical, distortion-based metrics such as MSE and PSNR can in some cases be poor indicators of subjective quality. Therefore, it has been necessary to develop perception-based criteria to achieve better (subjectively meaningful) quality assessments. One example of a perception-based quality assessment tool is the Mean Opinion Score (MOS). With MOS measurements, a number of observers view an image

[2] *Note:* In search applications it is normal to use Sum of Absolute Difference (SAD), omitting the division operation to reduce complexity.

Table 4.2 ITU-R Rec.BT.500 subjective evaluation impairment scale.

Score	Impairment	Quality rating
5	Imperceptible	Excellent
4	Perceptible but not annoying	Good
3	Slightly annoying	Fair
2	Annoying	Poor
1	Very annoying	Bad

and assess its quality on a five-point scale from bad to excellent. The subjective quality of the image (the MOS) is then characterized by performing statistical analysis on the ratings across a representative number of subjects.

Probably the most common testing procedure still is the Double Stimulus Continuous Quality Scale (DSCQS) approach, which is described in ITU-R Rec.BT.500 [7], although several multimedia quality evaluations will be based on single stimulus tests. DSCQS testing employs a five-point impairment scale as shown in Table 4.2. ITU-R Rec.BT.500 sets out a wide range of testing criteria and methods, covering viewing conditions, choice of test material, characteristics of the observer ensemble, methods for conducting the test, and methods for analyzing the results. Further details are provided in Chapter 10.

The problem with subjective tests is that they are time consuming and can be expensive to run. Hence this provides additional impetus for the development of robust perceptual quality metrics that can be used, with confidence, in their place.

4.7 Rates and distortions

4.7.1 Rate–distortion characteristics

The bit rate for any compressed sequence will depend on:

- The encoding algorithm used (intra- vs inter-frame, integer or subpixel motion estimation, coding modes, or block sizes available).
- The content (high spatio-temporal activity will in general require more bits to code).
- The encoding parameters selected. At the coarsest level this includes things such as spatial resolution and frame rate. At a finer granularity, issues such as quantizer control, intra- vs inter-modes, and block size choices will be key. The difference between an encoding based on good parameter choices vs one based on poor choices is huge.

Following the argument in Section 4.6.1 we need a description of coder performance that captures how it trades rate against distortion. This is achieved using a graph that plots rate vs distortion (or rate vs quality) for a given codec and a given video file. This allows us to compare codec performances and to assess how parameter

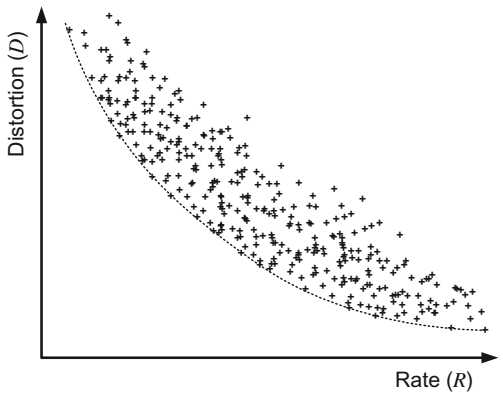

FIGURE 4.19

Rate–distortion plot for various coding parameter choices.

selections within a single codec can influence performance. However, if we randomly select coding parameters, some of these combinations might produce good results and some will inevitably produce poor results. The plot of points might, for many different parameter choices, look something like that in Figure 4.19. The aim is, of course, to ensure that the operating points produced by the codec in practice are those on the dotted curve—the Pareto curve that produces the lowest possible distortion for any given coding rate. To achieve this we must perform Rate–Distortion (RDO) or Rate–Quality Optimization (RQO).

Take for example a simple codec where the only control over RD performance is quantizer step size. In this case, adjusting the quantizer would produce different codec operating points as shown in Figure 4.20.

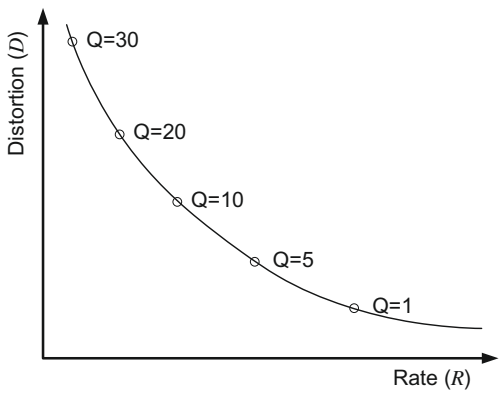

FIGURE 4.20

Rate–distortion plot showing quantizer controlled operating points.

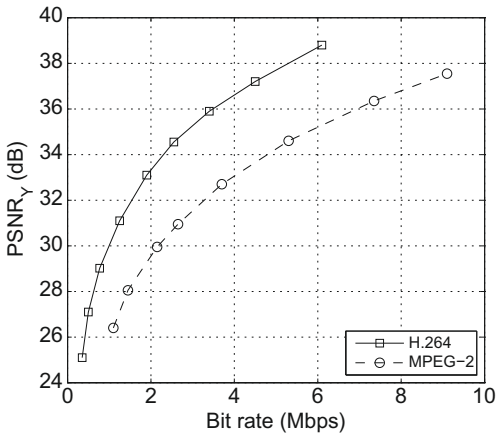

FIGURE 4.21
Example rate–quality comparison curves for typical standard definition entertainment content.

An example Rate–Quality curve (based on PSNR rather than distortion) is shown in Figure 4.21. This compares the performance of two codecs (MPEG-2 and H.264/AVC).

4.7.2 Rate–distortion optimization

Rate–Distortion Optimization (RDO) aims to maximize image or video quality, subject to bit rate constraints. RDO requirements and methods are non-normative in all video coding standards, but nonetheless are a key differentiator in terms of encoder performance. Many of the advanced features (e.g. block size selections) included in modern standards such as H.264/AVC and HEVC will not deliver savings without optimization included as part of the coder control process. The RDO must select the best modes and parameter sets for each region of the video. Often a Lagrangian optimization approach is adopted where the distortion measure, D, is based on sum-of-squared difference and the rate, R, includes all bits associated with the decision including header, motion, side information, and transform data.

For all possible parameter vectors, $\mathbf{p}$, the aim is to solve the constrained problem:

$$\min_{\mathbf{p}} D(\mathbf{p}) \quad \text{s.t.} \quad R(\mathbf{p}) \leq R_T \tag{4.28}$$

where R_T is the target bit rate, or to solve the unconstrained Lagrangian formulation:

$$\mathbf{p}_{opt} = \arg\min_{\mathbf{p}} \{D(\mathbf{p}) + \lambda R(\mathbf{p})\} \tag{4.29}$$

where λ controls the rate–distortion trade-off. Further details on RDO methods can be found in Chapter 10.

4.8 Summary

This chapter has introduced the digital representations, formats, processing techniques, and assessment methods that underpin the coding process. Techniques such as gamma correction and color space conversion have been shown to be essential in providing a digital description of a video signal that is best suited to further compression. Indeed, by exploiting the color processing and perception attributes of the HVS, we have shown that appropriate luma/chroma formats can be designed that reduce the bit rate for a digital video by 50%, even before we apply the conventional compression methods that are described in the following chapters. Finally, we have emphasized that assessment methods are an essential element in video codec design, both as a means of comparing the performance of different compression systems and as a basis for optimizing a codec's rate–distortion performance.

References

[1] A. Netravali, B. Haskell, Digital Pictures: Representation, Compression and Standards, second ed., Plenum Press, 1995.
[2] <http://en.wikipedia.org/wiki/Interlaced_video>, September 2013.
[3] C. Poynton, A Guided Tour of Color Space, August 2013. <http://www.poynton.com/PDFs/Guided_tour.pdf>.
[4] B. Bayer, Color Imaging Array, US Patent 3,971,065, 1976.
[5] C. Poynton, Digital Video and HD, second ed., Morgan Kaufmann, 2012.
[6] S. Winkler, Digital Video Quality, John Wiley, 2005.
[7] Recommendation ITU-R BT.500-13, Methodology for the Subjective Assessment of the Quality of Television Pictures, ITU-R, 2012.