数字媒体技术基础

Meng Yang

www.smartllv.com

SUN YAT-SEN University

机器智能与先进计算教育部重点实验室

智能视觉语言学习研究组

目 录

5 视觉媒体信息表示

Course Outline

目 录

5. 1 图形图像的数据类型

5.1 图形图像的数据类型

The number of file formats used in multimedia continues to proliferate. For example, Table shows a list of some file formats used in the popular product Macromedia Director.

Table 1:Macromedia Director File Formats

File Import					File Export		Native
Image	Palette	Sound	Video	Anim.	Image	Video	
.BMP, .DIB, .GIF, .JPG, .PICT, .PNG, .PNT, .PSD, .TGA, .TIFF, .WMF	.PAL .ACT	.AIFF .AU .MP3 .WAV	.AVI .MOV	.DIR .FLA .FLC .FLI .GIF .PPT	.BMP	.AVI .MOV	.DIR .DXR .EXE

5. 1 图形图像的数据类型

- The most common data types for graphics and image file formats——24-bit color and 8-bit color.
- Some formats are restricted to particular hardware/ operating system platforms, while others are "cross-platform" formats.
- Even if some formats are not cross-platform, there are conversion applications that will recognize and translate formats from one system to another.
- Most image formats incorporate some variation of a **compression** technique due to the large storage size of image files. Compression techniques can be classified into either **lossless** or **lossy**.

5. 1. 1 黑白图像(1-bit Image)

- Each pixel is stored as a single bit (0 or 1), so also referred to as **binary image.**
- Such an image is also called a 1-bit **monochrome** image since it contains no color.
- Fig.1 shows a 1-bit monochrome image (called “Lena“ by multimedia scientists ——this is a standard image used to illustrate many algorithms).

5.1.2 灰度图像(8-bit Gray-level Images)

- Each pixel has a gray-value between 0 and 255. Each pixel is represented by a single byte; e.g., a dark pixel might have a value of 10, and a bright one might be 230.

- **Bitmap:** The two-dimensional array of pixel values that rep-resents the graphics/image data.

- **Image resolution** refers to the number of pixels in a digital image (higher resolution always yields better quality).

  —Fairly high resolution for such an image might be $1\,600 \times 1{,}200$, whereas lower resolution might be 640×480.

5.1.2 灰度图像(8-bit Gray-level Images)

- **Frame buffer**: Hardware used to store bitmap.

  ——**Video card** (actually a *graphics card*) is used for this purpose.

  ——The resolution of the video card does not have to match the desired resolution of the image, but if not enough video card memory is available then the data has to be shifted around in RAM for display.

5.1.2 灰度图像(8-bit Gray-level Images)

- 8-bit image can be thought of as a set of 1-bit **bit-planes**, where each plane consists of a 1-bit representation of the image at higher and higher levels of "elevation": a bit is turned on if the image pixel has a nonzero value that is at or above that bit level.
- Fig.2 displays the concept of bit-planes graphically as follow:

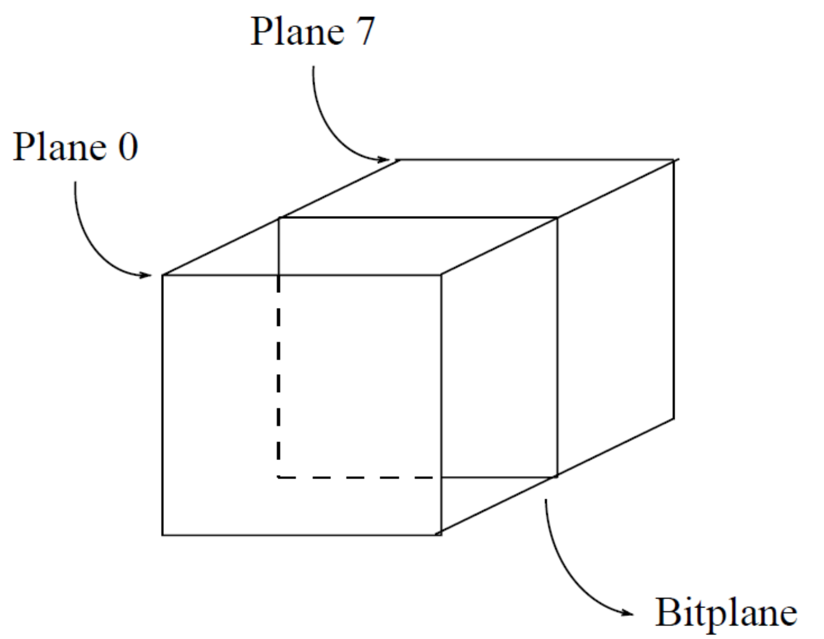


Fig.2 Bit-planes for 8-bit grayscale image.

5.1.2 灰度图像(Multimedia Presentation)

- Each pixel is usually stored as a byte (a value between 0 to 255), so a 640 × 480 grayscale image requires 300 kB of storage (640 ×480 = 307,200).

- Fig.3 shows the Lena image again, but this time in grayscale.

Fig. 3 Grayscale image of Lena.

5.1.2 灰度图像(Dithering)

- **Dithering** is used to calculate patterns of dots such that values from 0 to 255 correspond to patterns that are more and more filled at darker pixel values, for printing on a 1-bit printer.

- When an image is printed, the basic strategy of **dithering** is used, which trades intensity resolution for spatial resolution to provide ability to print multi-level images on 2-level (1-bit) printers.

5.1.2 灰度图像(Dithering)

- The main strategy is to replace a pixel value by a larger pattern, say 2×2 or 4×4, such that the number of printed dots approximates the varying-sized disks of ink used in analog, in **halftone printing** (e.g., for newspaper photos) as see next page:

  1. Half-tone printing is an analog process that uses smaller or larger filled circles of black ink to represent shading, for newspaper printing.

  2. For example, if we use a 2×2 **dither matrix**

$$\begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$$

5.1.2 灰度图像(Dithering)

- we can first re-map image values in 0..255 into the new range 0..4 by (integer) dividing by 256/5. Then, e.g., if the pixel value is 0 we print nothing, in a 2×2 area of printer output. But if the pixel value is 4 we print all four dots.

- The rule is:
  - If the intensity is > the dither matrix entry then print an on dot at that entry location: replace each pixel by an $n \times n$ matrix of dots.

5.1.2 灰度图像(Dithering)

- Note that the image size may be much larger, for a dithered image, since replacing each pixel by a 4×4 array of dots, makes an image 16 times as large.

- A clever trick can get around this problem. Suppose we wish to use a larger, 4×4 dither matrix, such as

$$\begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}$$

5.1.2 灰度图像(Dithering)

- An **ordered dither** consists of turning on the printer out- put bit for a pixel if the intensity level is greater than the particular matrix element just at that pixel position.
- Fig.4 (a) shows a grayscale image of "Lena". The ordered-dither version is shown as Fig.4 (b), with a detail of Lena's right eye in Fig. 4 (c).

(a) (b) (c)

5.1.2 灰度图像(Dithering)

- An algorithm for ordered dither, with $n \times n$ dither matrix, is as follows:

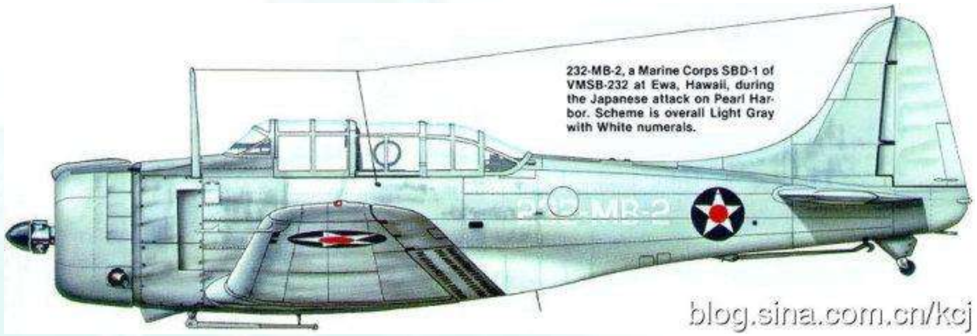
BEGIN
  for $x = 0$ to x_{max} // columns
    for $y = 0$ to y_{max} // rows
      $i = x \ mod \ n$
      $j = y \ mod \ n$
      // $I(x, y)$ is the input, $O(x, y)$ is the output,
      //D is the dither matrix.
      if $I(x, y) > D(i, j)$
        $O(x, y) = 1$;
      else
        $O(x, y) = 0$;
END

5.1.2 灰度图像(Dithering)

- 抖动趣闻

二战，美国飞机飞行方向和投弹的抛物线的计算都离不开一台精度不高的计算机

在天上计算的结果反而比在地面用的时候更准确



举个例子，假设计算机在完成一个任务的时候，一个参数先后被乘了五次，这个参数的精确值是5.4，但是计算机只能用整数，5.4只好每次都被四舍五入成了5，正确的结果应该是5.4*5.4*5.4*5.4*5.4=4591.65，可是计算的结果只有5*5*5*5*5=3125

机器在晃动的时候，会再额外产生一些的小误差，假设五次里5.4三次变成成了5，两次被化成了6，这样计算的结果是5*5*5*6*6=4500,比起3125精确了好多

5.1.3 彩色图像(24-bit Color Images)

- In a color 24-bit image, each pixel is represented by three bytes, usually representing RGB.
  - This format supports $256 \times 256 \times 256$ possible combined colors, or a total of 16,777,216 possible colors.
  - However such flexibility does result in a storage penalty: A 640×480 24-bit color image would require 921.6 kB of storage without any compression.

5.1.3 彩色图像(24-bit Color Images)

- **An important point**: many 24-bit color images are actually stored as 32-bit images, with the extra byte of data for each pixel used to store an *alpha* value representing special effect information (e.g., transparency).

5.1.3 彩色图像(24-bit Color Images)

- Fig.5 shows the image **forestre.bmp.**, a 24-bit image in Microsoft Windows BMP format. Also shown are the grayscale images for just the Red, Green, and Blue channels, for this image.

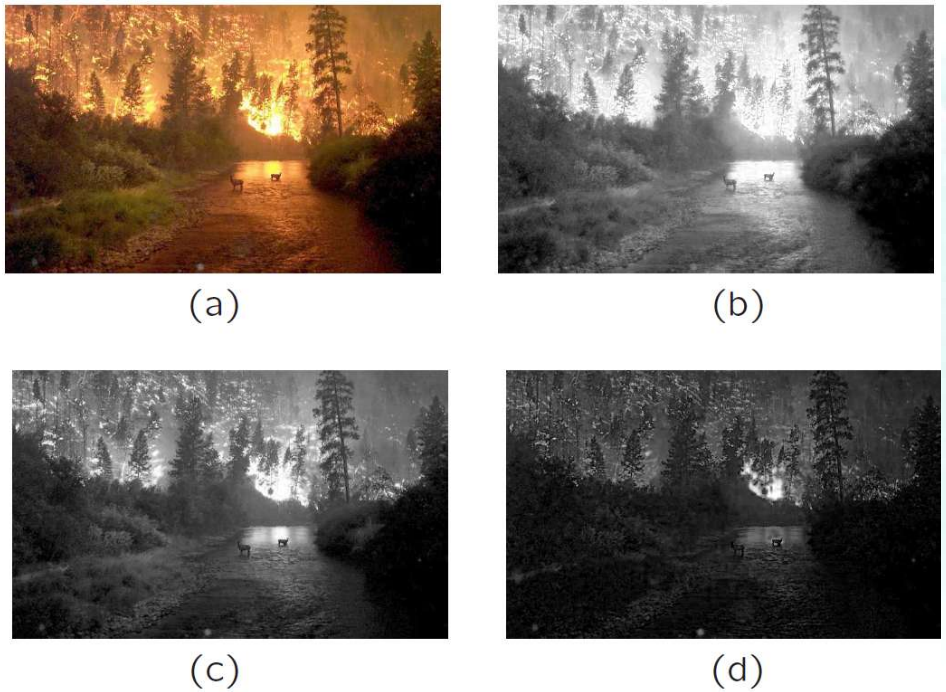


Fig. 5 High-resolution color and separate R, G, B color channel images. (a): Example of 24-bit color image "forestre.bmp". (b, c, d): R, G, and B color channels for this image

5.1.3 彩色图像(8-bit Color Images)

- Many systems can make use of 8 bits of color information (the so-called “256 colors”) in producing a screen image.

- Such image les use the concept of a **lookup table** to store color information.
  - Basically, the image stores not color, but instead just a set of bytes, each of which is actually an index into a table with 3-byte values that specify the color for a pixel with that lookup table index.

5.1.3 彩色图像(8-bit Color Images)

- Fig.6 shows a 3D histogram of the RGB values of the pixels in “forestre.bmp”.

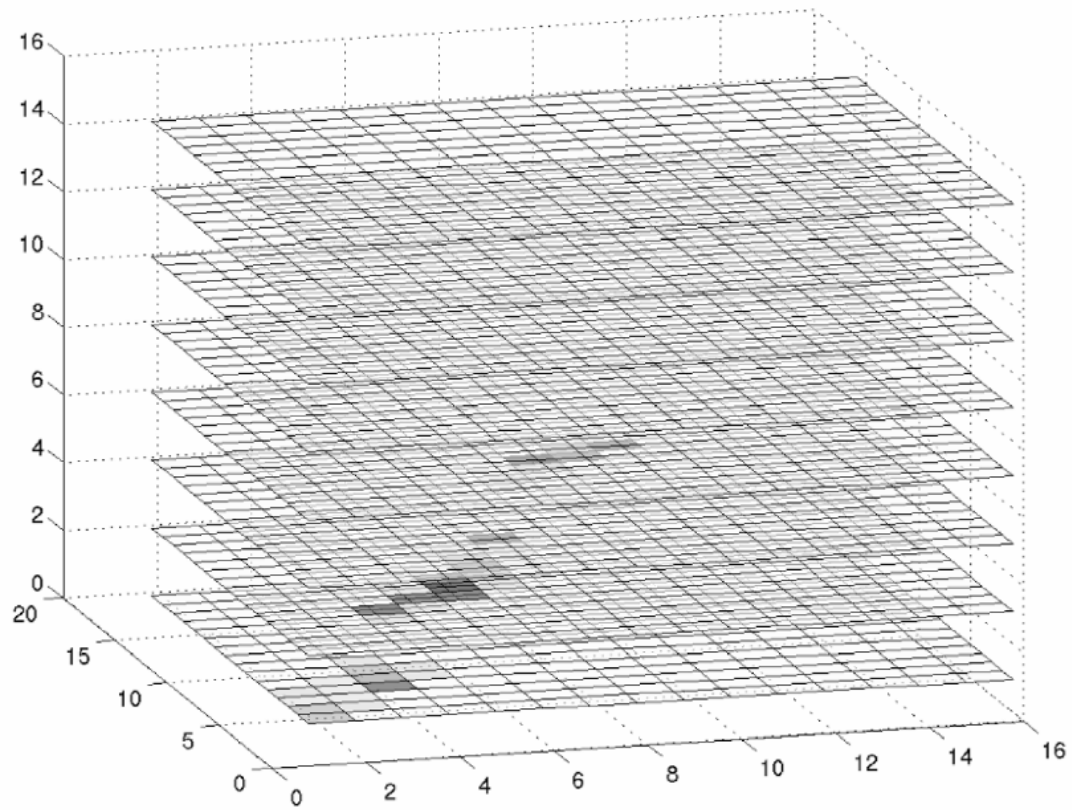


Fig 6 3-dimensional histogram of RGB colors in “forestre.bmp”.

5.1.3 彩色图像(8-bit Color Images)

- Fig.7 shows the resulting 8-bit image, in GIF format.

Fig.7 Example of gif image.

- Note the great savings in space for 8-bit images, over 24-bit ones: a 640×480 8-bit color image only requires 300 kB of storage, compared to 921.6 kB for a color image (again, without any compression applied).

5.1.3 彩色图像(Color Look-up Tables)

- The idea used in 8-bit color images is to store only the index, or code value, for each pixel. Then, e.g., if a pixel stores the value 25, the meaning is to go to row 25 in a color look-up table (LUT).

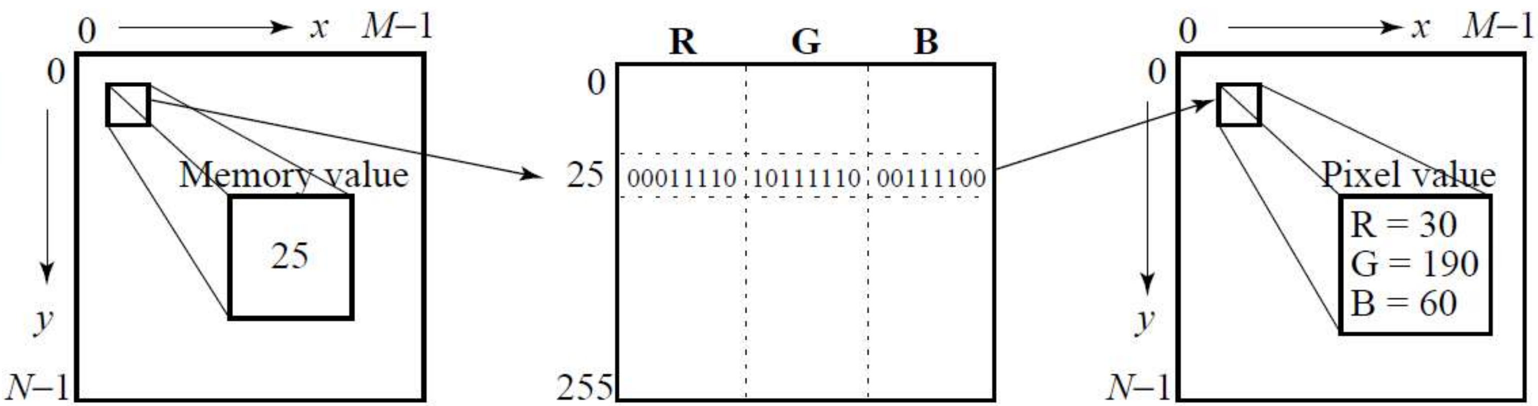


Fig 8: Color LUT for 8-bit color images.

5.1.3 彩色图像(Color Look-up Tables)

- A **Color-picker** consists of an array of fairly large blocks of color (or a semi-continuous range of colors) such that a mouse-click will select the color indicated.

  - In reality, a color-picker displays the palette colors associated with index values from 0 to 255.

  - Fig.9 displays the concept of a color-picker: if the user selects the color block with index value 2, then the color meant is cyan, with RGB values (0, 255, 255).

5.1.3 彩色图像(Color Look-up Tables)

- A very simple animation process is possible via simply changing the color table: this is called **color cycling** or **palette animation**.

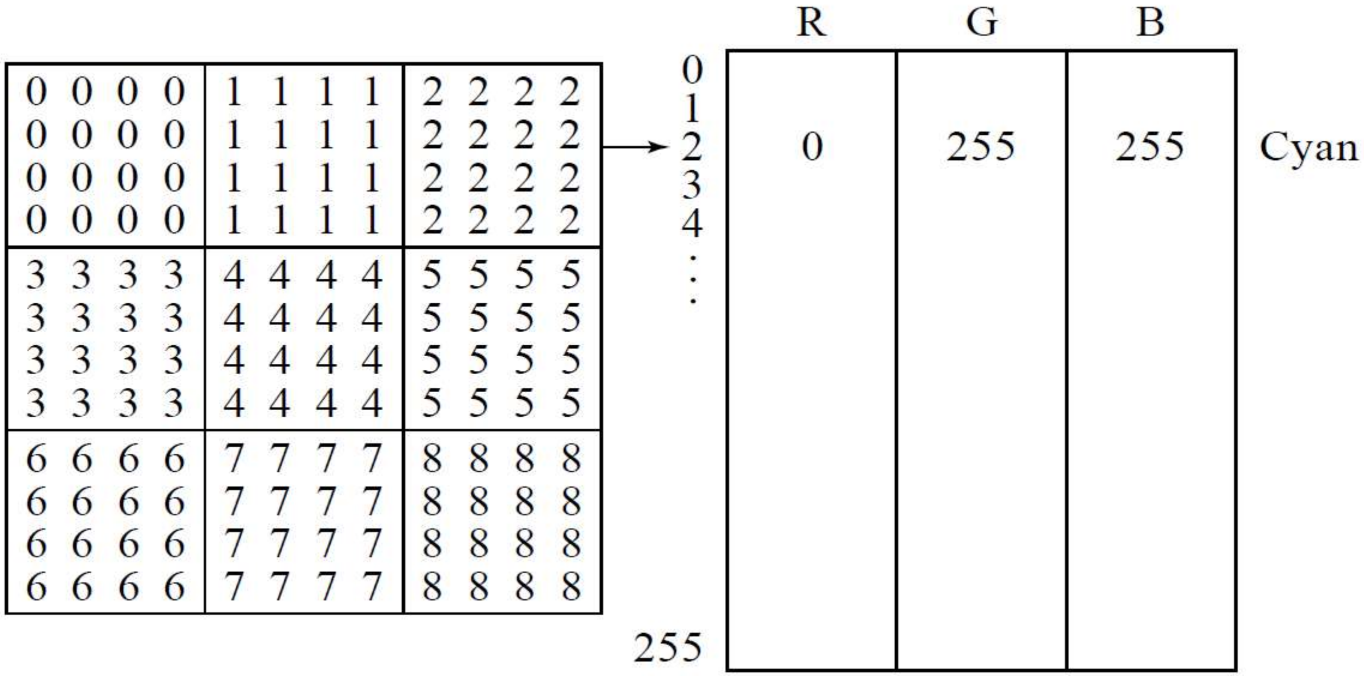


Fig.9: Color-picker for 8-bit color: each block of the color-picker corresponds to one row of the color LUT

5.1.3 彩色图像(Color Look-up Tables)

- Fig.10 (a) shows a 24-bit color image of “Lena”, and Fig.10 (b) shows the same image reduced to only 5 bits via dithering. A detail of the left eye is shown in Fig. 10 (c).

(a) (b) (c)

Fig.10: (a): 24-bit color image “lena.bmp”. (b): Version with color dithering. (c): Detail of dithered version.

5.1.3 彩色图像(Color Look-up Tables)

- **How to devise a color look-up table:**

The most straightforward way to make 8-bit look-up color out of 24-bit color would be to divide the RGB cube into equal slices in each dimension.

1. The centers of each of the resulting cubes would serve as the entries in the color LUT, while simply scaling the RGB ranges 0..255 into the appropriate ranges would generate the 8-bit codes.

5.1.3 彩色图像(Dithering)

图 1

图 2

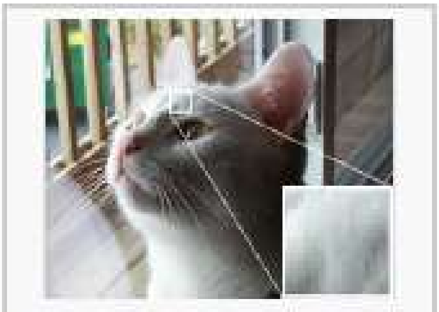


Figure 1. Original photo; note the smoothness in the detail.

Figure 2. Original image using the web-safe color palette with no dithering applied. Note the large flat areas and loss of detail.

Figure 3. Original image using the web-safe color palette with Floyd–Steinberg dithering. Note that even though the same palette is used, the application of dithering gives a better representation of the original.

Figure 4. Here, the original has been reduced to a 256-color optimized palette with Floyd–Steinberg dithering applied. The use of an optimized palette, rather than a fixed palette, allows the result to better represent the colors in the original image.

5.1.3 彩色图像(Color Look-up Tables)

- How to devise a color look-up table：

2. Since humans are more sensitive to R and G than to B, we could shrink the R range and G range 0..255 into the 3-bit range 0..7 and shrink the B range down to the 2-bit range 0..3, thus making up a total of 8 bits.

3. To shrink R and G, we could simply divide the R or G byte value by (256/8)=32 and then truncate. Then each pixel in the image gets replaced by its 8-bit index and the color LUT serves to generate 24-bit color.

5.1.3 彩色图像(Color Look-up Tables)

- **Median-cut algorithm**: A simple alternate solution that does a better job for this color reduction problem.

  (a) The idea is to sort the R byte values and find their median; then values smaller than the median are labelled with a “0” bit and values larger than the median are labelled with a “1” bit.

5.1.3 彩色图像(Color Look-up Tables)

- **Median-cut algorithm**:

  (b) This type of scheme will indeed concentrate bits where they most need to differentiate between high populations of close colors.

  (c) One can most easily visualize finding the median by using a histogram showing counts at position 0..255.

  (d) Fig.11 shows a histogram of the R byte values for the forestfire.bmp image along with the median of these values, shown as a vertical line.

5.1.3 彩色图像(Color Look-up Tables)

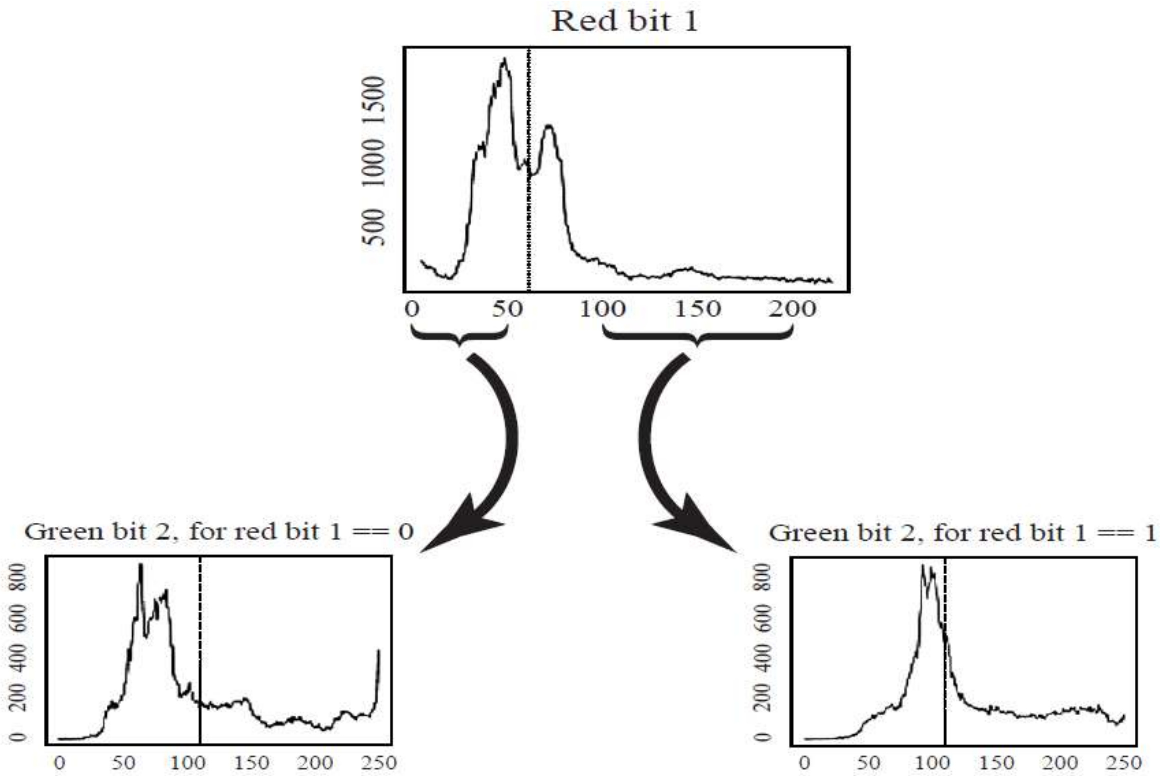


Fig.11 Histogram of R bytes for the 24-bit color image “forestfire.bmp” results in a “0” bit or “1” bit label for every pixel. For the second bit of the color table index being built, we take R values less than the R median and label just those pixels as “0” or “1” according as their G value is less than or greater than the median of the G value, just for the “0” Red bit pixels. Continuing over R, G, B for 8 bits gives a color LUT 8-bit index

目 录

5.2 视频图像的表示

5.2.1 视频图像的表示(Types of Video Signals)

- **Component video**

Component video: Higher-end video systems make use of three separate video signals for the red, green, and blue image planes. Each color channel is sent as a separate video signal.

(a) Most computer systems use Component Video, with separate signals for R, G, and B signals.

5.2.1 视频图像的表示(Types of Video Signals)

- Component video

Component video:

(b) For any color separation scheme, Component Video gives the best color reproduction since there is no "crosstalk" between the three channels.

(c) This is not the case for S-Video or Composite Video, discussed next. Component video, however, requires more bandwidth and good synchronization of the three components.

5.2.1 视频图像的表示(Types of Video Signals)

- Component video-1 signal

Composite video: color ("chrominance") and intensity ("luminance") signals are mixed into a *single* carrier wave.

a) **Chrominance** is a composition of two color components (I and Q, or U and V).

b) In NTSC TV, e.g., I and Q are combined into a chroma signal, and a color subcarrier is then employed to put the chroma signal at the high-frequency end of the signal shared with the luminance signal.

5.2.1 视频图像的表示(Types of Video Signals)

- Component video-1 signal

Composite video:

c) The chrominance and luminance components can be separated at the receiver end and then the two color components can be further recovered.

d) When connecting to TVs or VCRs, Composite Video uses only one wire and video color signals are mixed, not sent separately. The audio and *sync* signals are additions to this one signal.

- Since color and intensity are wrapped into the same signal, some interference between the luminance and chrominance signals is inevitable.

5.2.1 视频图像的表示(Types of Video Signals)

- S video-2 signal
- **S-Video**: as a compromise, (Separated video, or Super-video, e.g., in S-VHS) uses two wires, one for luminance and another for a composite chrominance signal.
- As a result, there is less crosstalk between the color information and the crucial gray-scale information.
- The reason for placing luminance into its own part of the signal is that black-and-white information is most crucial for

5.2.1 视频图像的表示(Types of Video Signals)

- S video-2 signal
- visual perception.

In fact, humans are able to differentiate spatial resolution in gray-scale images with a much higher acuity than for the color part of color images.

As a result, we can send less accurate color information than must be sent for intensity information—we can only see fairly large blobs of color, so it makes sense to send less color detail.

5.2.2 模拟视频(Analog Video)

- An analog signal $f(t)$ samples a time-varying image. So-called "progressive" scanning traces through a complete picture (a frame) row-wise for each time interval.

- In TV, and in some monitors and multimedia standards as well, another system, called "interlaced" scanning is used:

  a) The odd-numbered lines are traced first, and then the even-numbered lines are traced. This results in "odd" and "even" fields | two fields make up one frame.

  b) In fact, the odd lines (starting from 1) end up at the middle of a line at the end of the odd field, and the even scan starts at a half-way point.

5.2.2 模拟视频(Analog Video)

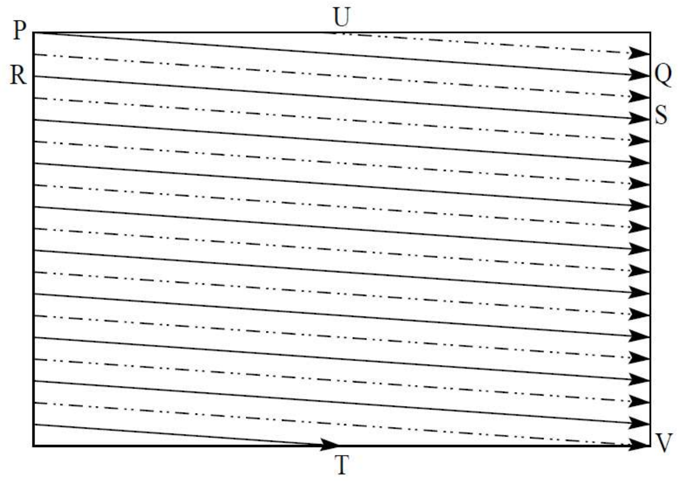


Fig. 12: Interlaced raster scan

c)Figure 12 shows the scheme used. First the solid (odd) lines are traced, P to Q, then R to S, etc., ending at T; then the even field starts at U and ends at V.

d)The jump from Q to R, etc. in Figure 12 is called the **horizontal retrace**, during which the electronic beam in the CRT is *blanked*. The jump from T to U or V to P is called the **vertical retrace**

5.2.2 模拟视频(Analog Video)

- Because of interlacing, the odd and even lines are displaced in time from each other—generally not noticeable except when very fast action is taking place on screen, when blurring may occur.

- For example, in the video in Fig. 13, the moving helicopter is blurred more than is the still background.

5.2.2 模拟视频(Analog Video)

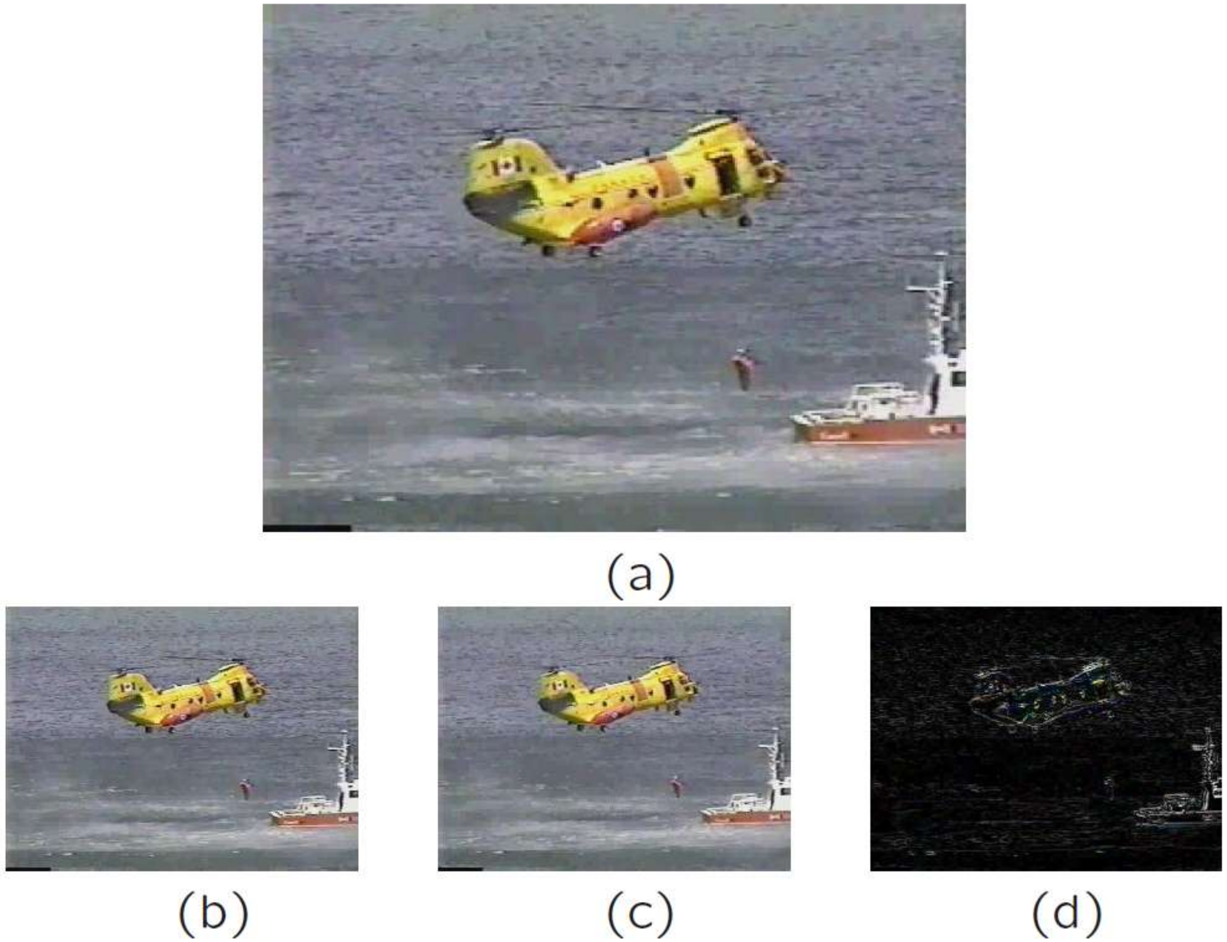


Fig. 13: Interlaced scan produces two elds for each frame. (a) The video frame, (b) Field 1, (c) Field 2, (d) Difference of Fields

5.2.2 模拟视频(Analog Video)

- Since it is sometimes necessary to change the frame rate, resize, or even produce stills from an interlaced source video, various schemes are used to "deinterlace" it.
  1. The simplest de-interlacing method consists of discarding one field and duplicating the scan lines of the other field. The information in one field is lost completely using this simple technique.
  2. Other more complicated methods that retain information from both fields are also possible.

*Analog video use a small voltage offset from zero to indicate "black", and another value such as zero to indicate the start of a line. For example, we could use a "blacker-than-black" zero signal to indicate the beginning of a line.

5.2.2 模拟视频(NTSC Video)

3. Since the horizontal retrace takes 10.9 μsec, this leaves 52.7 μsec for the active line signal during which image data is displayed (see Fig 14).

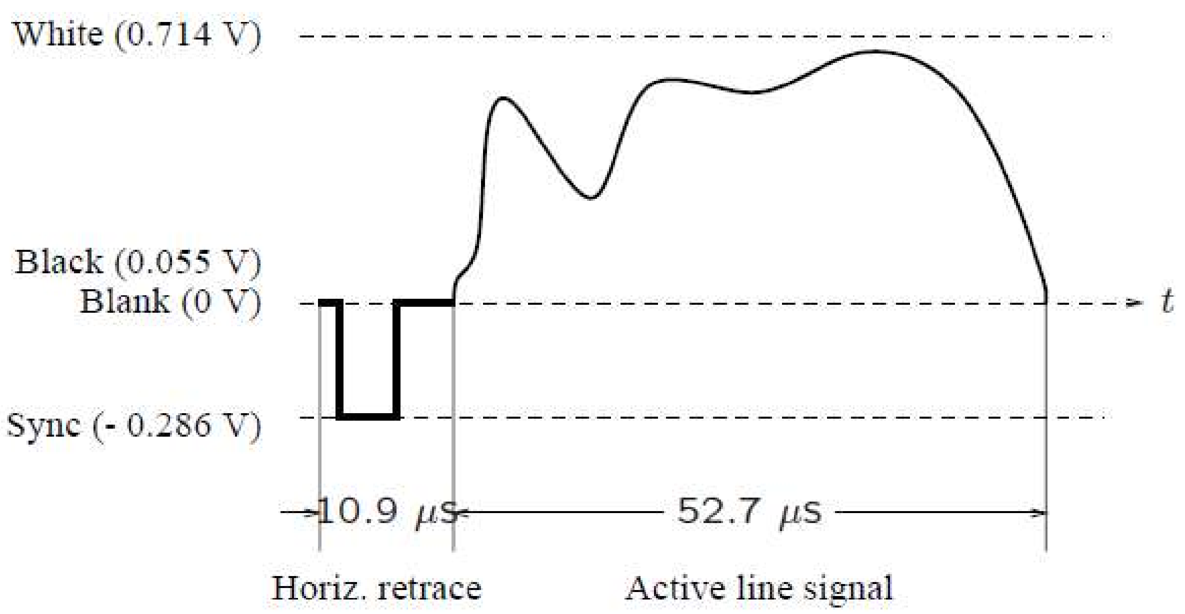


Fig.14 Electronic signal for one NTSC scan line.