Digital Image Processing 18MIT31C

UNIT-III: Image Compression: Fundamentals – Some basic compression methods – Huffman coding – Arithmetic coding – LZW coding - Bit-Plane coding – Run-Length coding. Image Segmentation: Fundamentals – Point, Line, and Edge Detection.

Textbook :

Gonzalez R C., and Woods R.E., “Digital Image Processing”, Prentice Hall, Third Edition

Prepared By : Mrs. G. Shashikala, Assistant Professor, Department of Information Technology

Image compression

- to reduce the amount of data required to represent an image

for a standard definition (SD) tv movie using 720 x 480 x 24 bit pixel arrays, with 30 fps (frames per second) SD digital video data must be accessed at

$$30\ \frac{frames}{sec} \times (720 \times 480)\ \frac{pixels}{frame} \times 3\ \frac{bytes}{pixel}$$

$$= 31{,}104{,}000 \text{ bytes / sec}$$

and a two - hour movie consist of 31,104,000 bytes / sec x (60^2) sec /hr x 2 hrs

$$\approx 2.24 \times 10^{11} \text{ bytes or 224 GB of data}$$

Fundamentals.

Data computer refers to the process of reducing the amount of data required to represent a given quantity of information. If we have b & b' denote the no. bits is two representation of the same information, the relative data redundancy R of representation with b bits is $R = 1 - \frac{1}{C}$

where C is the compression representation, & $C = \frac{b}{b'}$

If c=10 (written as 10 : 1), the larger representation has 10 bits of data for every 1 bit of data in the smaller representation, i.e, for the larger representation, R=0.9 or the 90% of the data is redundant.

Three types of redundancies in 2D intensity arrays are :

1. Coding redundancy :
   A code is a system of symbols (letters, numbers, etc.) used to represent a body of information. Each piece of information is assigned a sequence of code symbols, called a code word. The number of symbols in each code word is its length. [eg 8 bit codes].
2. Spatial & temporal redundancy:
   Because of the pixels of most, 2-D intensity arrays are correlated spatially (i.e., each pixel is similar or dependent on neighbouring pixels), information is unnecessarily replicated in the correlated pixels. In a video sequence, temporally correlated pixels (nearly frames) also duplicate information.
3. Irrelevant info:
   Most, 2-D intensity arrays contain information that is ignored by the human visual system. It is redundant in the sense that it is not used.

(i) Coding redundancy

Assuming that r_k in the interval [0, L-1] is used to represent the intensities of an M x N image and that each r_k occurs with probability Pr (r_k)

$$\Pr(r_k) = \frac{nk}{MN} \qquad k=0,1,2,\ldots\ldots,L-1$$

Where L is the number of intensity value and nk is the no of times that the k^{15} intensity appears in the image. If the no. of bits used to represt each value of r_k is l(r_k) then the average no. of bits required represent each pixel is

$$L_{avg} = \sum_{k=0}^{L-1} l(rk).\Pr(rk)$$

The total no.of bits required to represent a M x N image is MN Lavg.

If the intensities are represented using a m-bit fixed- length code, then l(r_k) = m

$$\text{So } L_{avg} = m \sum_{k=0}^{L-1} \Pr(rk)$$

Since Pr (r_k) = 1 , L_{avg} = m

Example of variable length coding:

r_k	Pr (r_k)	Code 1	l_1 (r_k)	Code 2	l_2(r_k)
r_{87}=87	0.25	01010111	8	01	2

r_{128}=128	0.47	10000000	8	1	1
r_{186}=186	025	11000100	8	000	3
r_{255}=255	0.03	11111111	8	001	3
r_k for $k \neq$ 87,128,186,255	0	-	8	-	0

If code 1 is used, $l_1(r_k)$ = 8 bits for all r_k

If code 2 is used

L_{avg} = 0.25 (2) + 0.47 (1) + 0.25(3) +0.03(3) = 1.81 bits

The total no. of bits required to represent the entire image is, MNL = 256 x 256 x 1.81

= 118,621 bits

Then $C = \frac{256\ X\ 256\ X\ 8}{118{,}621} = \frac{8}{1.81} \approx 4.42$

And $R = 1 - \frac{1}{4.42} = 0.774$

Thus 77.4% of the data is redundant. So, in the variable length code, fewer bits are assigned to move probable intensity values.

(ii)Spatial & Temporal Redundancy

For an image of constant intensity lines, in the corresponding 2D intensity array:

1. All 256 intensities are equally probable
2. Pixels are independent is vertical direction
3 Since pixels along each line are identical they are maximally correlated in the horizontal direction

When the image of constant intensity lines is represented as conventional 8 bit intensity array it cannot be compressed by variable -length coding alone. Such image can be represented as a sequence of run-length pairs, where run length pair specified the start of a new intensity and the no. of consecutive pixels that havethat intensity.

A run length based representation Compresses the original 2D, 8 best int array by

(256 x 256x 8) / [(256 +256) x 8] or 128 : 1.

Each 256-pixel line of the original representation is replaced by a single 8-bit intensity value & length 256 is the run-length representation.

Since most pixel intensities can be predicted from neighbouring intensities, the information carried by a single pixel is small. The information is redundant since intensities can be inferred from neighbours. To reduce the redundancy differences between adjacent pixels can be used. Such transformations are called mappings. If the pixels can be reconstructed without error, the mapping is reversible, otherwise it is irreversible.

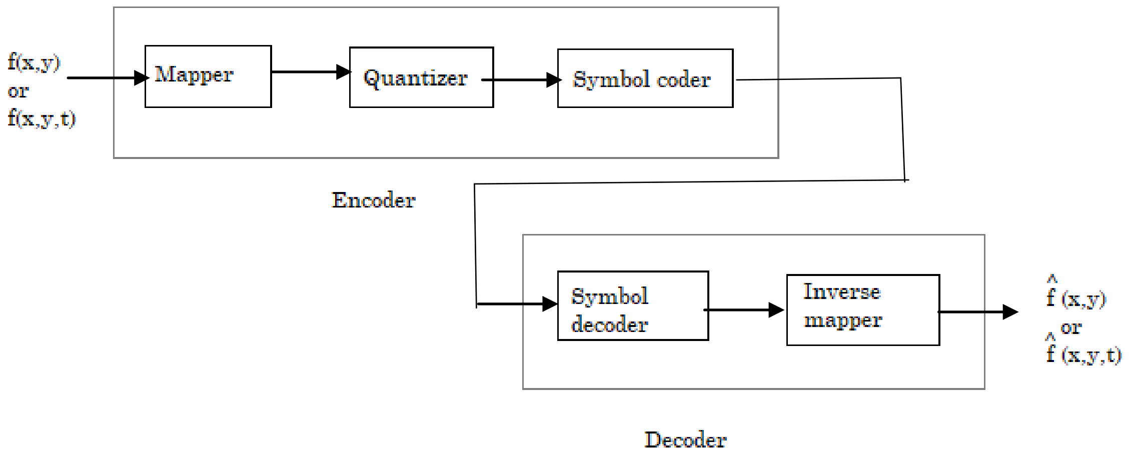
(iii)Irrelevent Information

Information that is ignored by the human visual system or which is extraneous can be omitted. When an image is a homogeneous field of grey it can be represented by its average intensity alone a single 8-bit value. The original 256 x 256 x 8 bit array reduced to a single byte and c = (256 x 256 x 8) / 8 or 65,536 : 1

Even if several intensity values are present, the human visual system averages these intensities, perceives only the average value, & ignores the small changes is intensity. A histogram equalized version of the image will make the intensity changes visible. So essential information like in medical application should not be omitted.

Image compression models.

An image compression system is composed of an encoder and a decoder. The encoder performs compression, and the decoder performs decompression. A coder is used for encoding & decoding



Input image f(x,y) is fed into the encoder, which creates a compressed representation of the input. The decoder generates a reconstructed output image f^(x, y) when f^(x, y) is an exact replica of f(x,y), the compression is error free, lossless, or information preserving. Otherwise, the reconstructed output image is distorted and the compression system is lossy.

The encoding or Compession Process

In the first stage of the encoding process, a mapper transforms f(x, y) into a format to reduce spatial and temporal redundancy. This is a reversible operation. Example Run-length coding

The quantizer reduces the accuracy of the mappers output is accordance with a pre-established fidelity criterion, to keep irrelevant information out of the compressed representation. This is omitted when error-free compression is desired.

The symbol coder generates a fixed or variable length code to represent. The quantizer output & maps the output to a code. The shortest code words are assigned to the most frequently occurring quantizer output values thus minimizing coding redundancy. This operation is reversible

The decoding or decompression process

The decoder contains a symbol decoder and an inverse mapper. They perform the inverse operations of symbol encoder & mapper.

Image formats, containers & Compression standards

Image file format tells how the data is arranged & the type of Compression used.Image container – similar to file format & handles multiple types of image data. Image compression standard define procedures for compressing & de compressing.

IEC - International Electro-technical Commission

ITU-T - International Telecommunication Union

CCITT - Consultative Committee for International Telephony and Telegraphy

s.no	Name	Orgnaisation	Description
	Bit-level still images		
1.	CCITT Group 3	ITU-T	a facsimile (fax) method for transmitting binary document over telephone lines
2.	CCITT Group 4	ITU -T	a streamlined version of CCITT Group 3 and Supports 2D run length coding only
3.	JBIG or JBIG1	ISO / IEC / ITU-T	A Joint Bi-level image experts Group for lossless compression of bi-level image.
4.	JBIG2	ISO / IEC / ITU-T	similar to JBIG1 for bi-level image in desktop, internet & FAX application.
	Continuous tone still images		
5.	JPEG	ISO / IEC / ITU-T	A Joint Photographic Experts Group standard for images of photo quality - lossy baseline coding system…use DCT (Discrete Cosine Transform)
6.	JPEG – LS	ISO / IEC / ITU-T	
7.	JPEG - 2000	ISO / IEC / ITU-T	

	VIDEO		
8.	DV	IEC	Digital Videos
9.	H.261	ITU – T	A two way video conferencing standard for ISDN lines. A DCT based compression is used. Frame to frame prediction differencing
10.	H.262	ITU – T	Similar to MPEG – 2
11.	H.263	ITU - T	Enhanced version of H.261 designed for telephone modems (i.e. 28.8 kb/s).
12.	H.264	ITU – T	Extension of H.262 and H.263 for video conferencing. Internet streaming and TV broadcasting.
13.	MPEG1	ISO / TEC	A Motion Picture Expert Group for CD-ROM applications with non-interlaced video at 1.5 Mbps . Similar to H.26 1, but frame predictions are based on previous format, next format, and an interpolation of both
14.	MPEG2	ISO / TEC	Extension of MPEG1 designed for DVDs with transfer rates 15 Mb/s supports interlaced video & HDTV.
15.	MPEG4	ISO / TEC	Extension of MPEG2 that supports variable block size & prediction differencing within frames
16.	MPEG4 AVC	ISO / TEC	Advanced video coding similar to H.264
	Continuous tone still images		
17	BMP	Microsoft	Windows bitmap used for simple uncompressed image
18	GIF	Compuserve	Graphic Interchange Format – For small animations and low resolution films for WWW, uses lossless LZW coding
19	PDF	Adobe systems	Portable Document Format – to represent 2D document
20	PNG	WWW Consortium (W3C)	Portable Network Graphics - losslessly compresses full color images by coding the difference between each pixel's value and a predicted value based on past pixels
21	TIFF	Aldus	Tagged Image File Format – supports many image compression standards like JPEG, JPEG-LS
	Video		
22	AVS		Audio - video std. similar to H. 264, uses exponential Golomb coding
23	HDV	Company Consortium	High Definition Video, An extension of DV
24	M-JPEG	Various Companies	Motion JPEG, each frame is compressed independently using JPEG.
25	Quick-Time	Apple computer	a media container supporting DV, H.261, 262, 264, MPEG-1, MPEG-2,
26	VC-1 WMV9	SMPTE Microsoft	Most used on the Internet , Adopted for HD & Blu-ray high-defn DVDS.

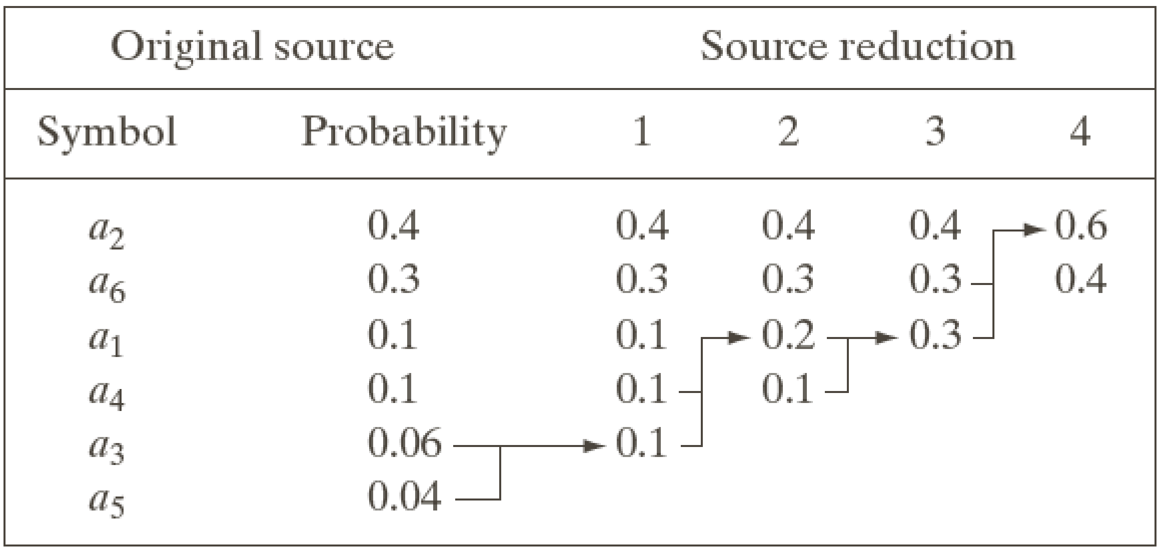
Some Basic Compression Methods

Some basic compression methods are

- Huffman Coding
- Golomb Coding
- Arithmetic Coding
- LZW Coding
- Run-Length Coding
- One-dimensional CCITT compression
- two-dimensional CCITT compression
- Symbol-Based Coding

Huffman coding

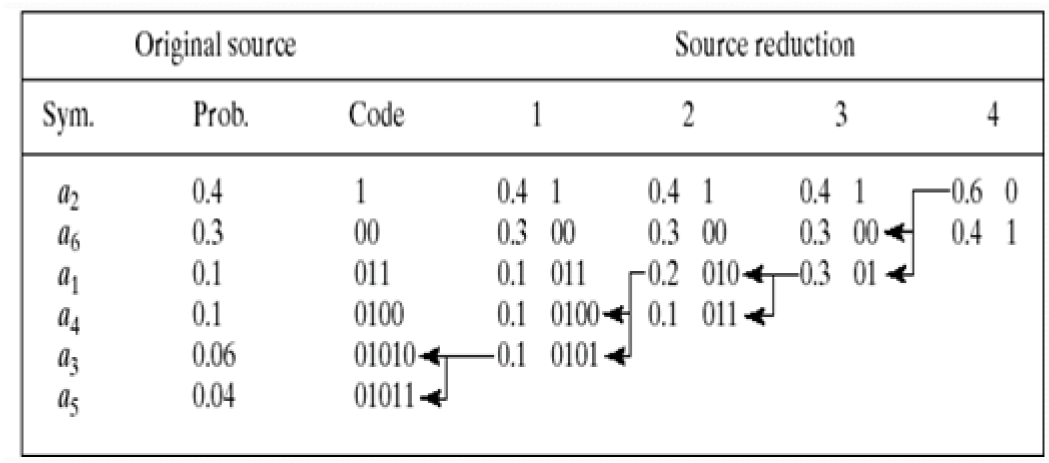
This is One of the most popular techniques for removing coding redundancy . When coding the symbols of an information source individually, Huffman code yields the smallest possible number of code symbols per source symbol. In terms of Shannon's theorem , the resulting code is optimal for a fixed value of n, subject to the constraint that the spouce symbols be be coded one at a time. . (The source symbols may be either the intensities of an image or the output of an intensity mapping operation like pixel differences,run length etc)



Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Huffman Source Reduction.

- The first step in Huffman's approach is to create a series of source reductions by ordering the probabilities of the symbols under consideration and combining the lowest probability symbols into a single symbol that replaces them in the next source reduction.
- The second step in Huffman's procedure is to code each reduced source, starting with the smallest source and working back to the original source. The code for a two symbol source are 0 and 1. These are assigned to two symbols on the right. Since the prob. 0.6 was generated by combining two symbols, the 0 used to code it is now assigned to both of these symbos, & a 0 & are 1 are appended to each. This is repeated to each reduced source



Original source			Source reduction							
Sym.	Prob.	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

The symbols are coded one at a time and decoding is done using a simple lookup table. The code used here is a block code, since each source symbol is mapped into a fixed sequence of code symbols. It is uniquely decodable.

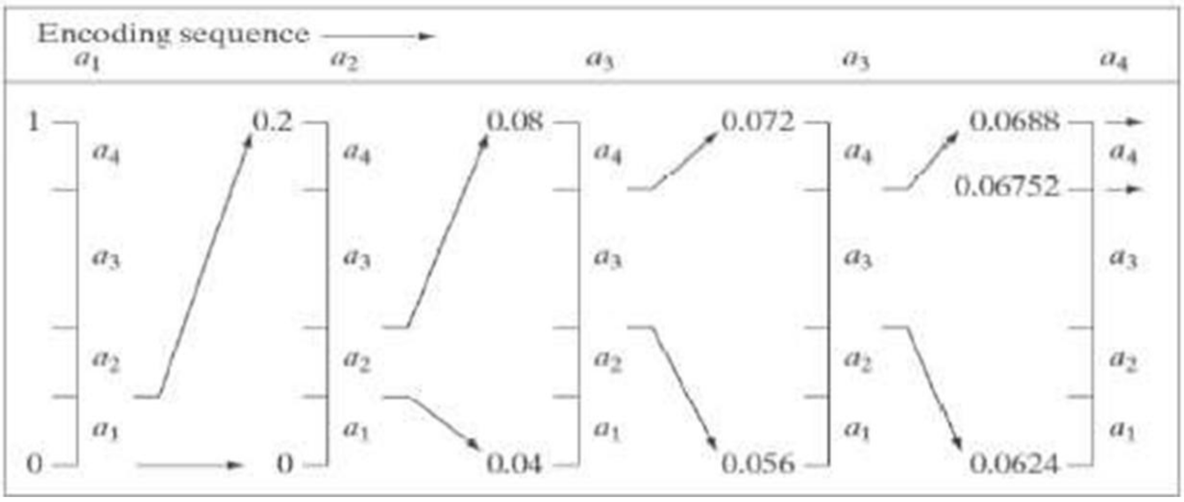
Arithmetic coding

This generates nonblock codes. A one-to-one correspondence between Source symbols & code words does not exist . But, an entire sequence of source symbol (or message), is assigned a single arithmetic code word. The code word defines an internal of real numbers between 0 & 1. As the no. of symbols in the message increases, the interval used to represent it becomes smaller & the no. of information units (bits) required to represent the interval, become larger.

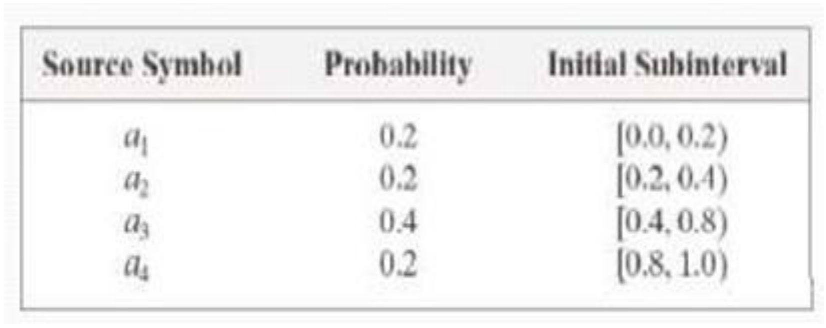
for ex a five symbol sequence or message, a1 a2a3a3 a4, from a four symbol source is coded. The internal [0, 1]is sub divided into four regions based on the probabilities of each source symbol. Symbol a1, is associated with subinterval [0, 0.2] and the message interval is initially narrowed to [0, 0.2]. This is then subdivided according to original source symbol probabilities and the proces continues with the next message symbol .

Symbol a2 narrows the subinterval to [0.04, 0.08] , a3 to [0.056, 0.072] & so on. The final message symbol, reserved for end-of -message indicator, narrows the range to [0.06752, 0.0688] . Any number within this subinterval like 0.068 can be used to represent the message.

Here 3 decimal digits are used to represent the five symbol message i.e., 0.6 decimal digits are used per source symbol.



Arithmetic coding procedure.



Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

Arithmetic coding Example.

LZW coding

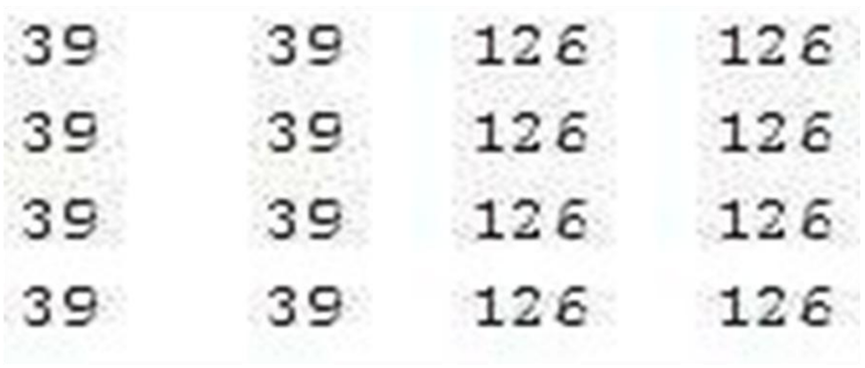
This addresses spatial redundancies in an image. The technique called Lempel –Ziv -Welch (LZW) coding, assigns fixed-length code words to variable length sequences of source symbols.

* Apriori knowledge of the probabilities of occurrences of the symbols to be encoded is not required.

During the coding process, a codebook dictionary containing the source symbols to be coded is constructed . For 8-bit monochrome images, the first 256 words of the dictionary are assigned to intensities 0,1,2, … 255. The encoder sequentially examines image pixels and intensity sequences are put in successive locations. Fox ex, the sequence "255-255" might be assigned to location 256. Whenever two consecutive white pixels are encountered, code word 256 is used to represent them

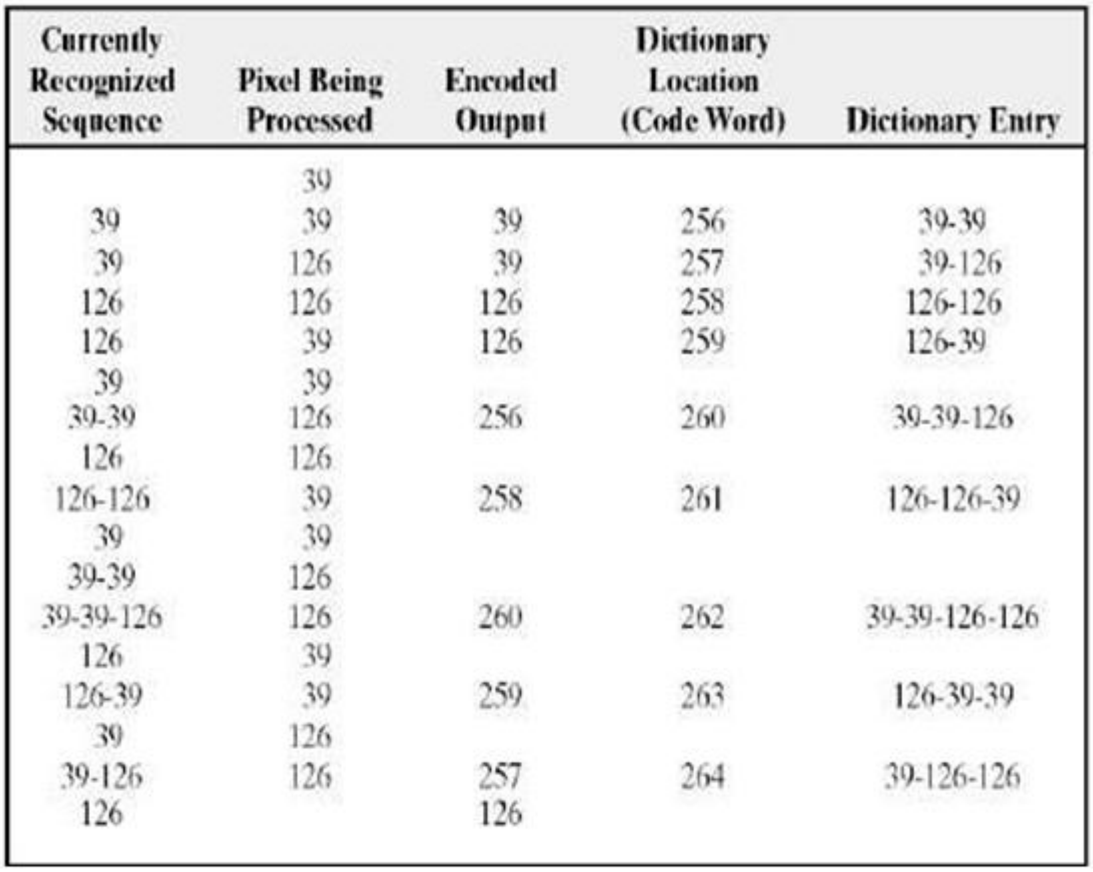
LZW compression has been integrated into a variety of mainstream imaging file formats, including GIF, TIFF, and PDF. The PNG format was created to get around LZW licensing requirements

Consider the following 4x4, 8 bit image of a vertical edge



39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

The image is encoded by processing its pixels in a left-to- right, top-to-bottom manner. The dictionary is searched for each concatenated sequence. If the concatenated sequence is not found, it is added to the dictionary. As per the ex., nine additional code words are defined. Now the dictionary contains 265 code words .



Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

In LZW coding, the coding dictionary or code book is created while the data are being encoded

Run Length coding

Images with repeating intensities along their rows (or columns) can be compressed by representing runs of identical intensities as run-length pairs where each ruin-length pair specifies the start of a new intensity and the no. of consecutive pixels that have that intensity. This technique is known as run-length encoding (RLE) & used in fascimile (FAX) coding. Compression is achieved by eliminating a simple form of spatial redundancy – i.e., groups of identical intensities.

In a BMP file format, image data is represented in two different Mode : encoded and absolute, and any mode can occur anywhere in the image. In encoded mode, a two byte RLE representation is used. The first byte specifies the no. of consecutive pixels that have the color index contained in the second byte. The color index is chosen from a table of 256 possibilities.

In absolute mode, the first byte is 0, and the second byte signals one of four possible conditions, as shown below.

Second byte value	Condition
0	End of line
1	End of image
2	Move to a new position
3-255	Specify pixels individually

if the second byte is between 3 – 255 , it specifies the no. of uncompressed pixels that follow

RLE is effective when compressing binary images, because there are only two possible intensies (black & white).

Bit - plane coding

This is based on the concept of decomposing a multilevel (Monochxome or Color) image into a series of binary images and compressing each binary image via one of several binary compression methods.

The intensities of an m-bit monochrome image can be represented as

am-12m-1 + am-2 2 m-2 + ... + a1 21 + a0 20

The m-coefficients of the polynomial can be separated into m 1 bit planes.

The lowest order bit plane contains the a_0 pixels of each pixel, and the highest order bit plane contains the a_{m-1} bits. The disadvantage of this method is that small changes in intensity can have a significant impact on the complexity of the bit planes.

An alternative method (which reduces the effect of small intensity variations) is to first represent image by an m-bit Gray code . The m-bit Gray code $g_{m-1}\ldots\, g_2 g_1 g_0$ that corresponds to the polynomial can be computed from

$$g_i = a_i \oplus a_{i+1} \quad 0 <= i <= m-2$$

$$g_{m-1} = a_{m-1}$$

Here $\oplus$ denotes the exclusive OR operation

Image Segmentation

Image segmentation is the operation of partitioning an image into a collection of connected sets of pixels

- The purpose of image segmentation is to partition an image into *meaningful* regions with respect to a particular application
- The segmentation is based on measurements taken from the image and might be *greylevel*, *colour*, *texture*, *depth* or *motion*

Point, Line & Edge Detection

The 3 types of image features are isolated points, lines & edges . Edge pixels are pixels at which the intensity of an image function changes abruptly, and edges are sets of connected edge pixels.

A line is an edge segment in which the intensely of the background on either side of the line is either much higher or much lower than the intensity of the line pixels. An isolated point is a line whose length & width are equal to one pixel.

Background

Derivatives of a digital function are defined in terms of differences. The first derivative :

a) must be zero in areas of constant intensity

b) must be non zero at the onset of an intensity step or ramp.

c) must be nonzero at points along an intensity ramp

A second derivative

a) must be zero is areas of constant intensity

b) must be nonzero at the onset and end of an intensity step or ramp

c) must be zero along intensity ramps.

Segmentation algorithms generally are based on one of two basis properties of intensity values

• Discontinuity: to partition an image based on abrupt changes in intensity (such as edges)

• Similarity: to partition an image into regions that are similar according to a set of predefined criteria.

Point, Line and Edge Detection

- First order derivatives produce thick edges at ramps.
- Second order derivatives are non zero at the onset and at the end of a ramp or step edge (sign change).

- Second order derivatives respond stronger at isolated points and thin lines than first order derivatives.

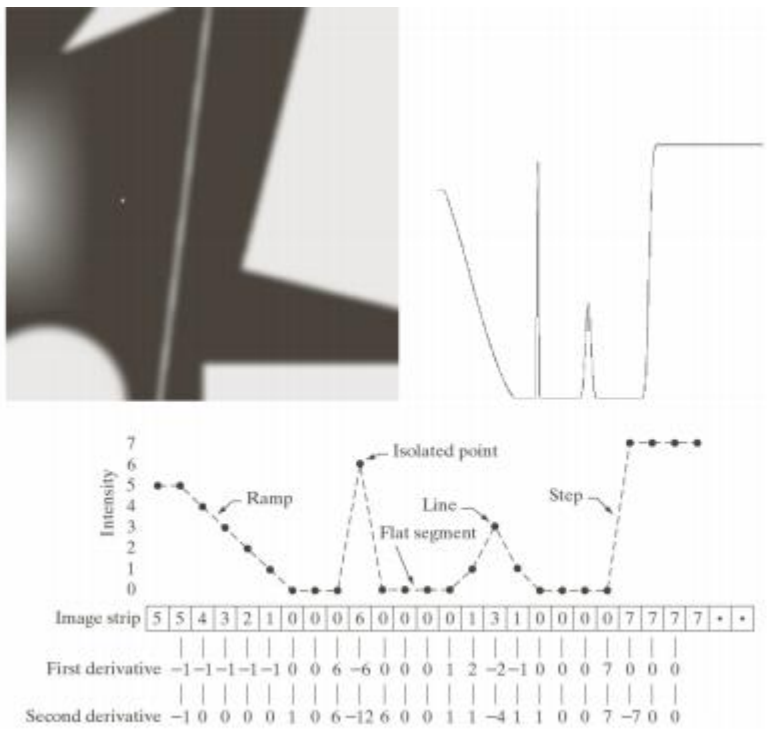


Image strip 5 5 4 3 2 1 0 0 0 6 0 0 0 0 1 3 1 0 0 0 0 7 7 7 7 • •

First derivative −1 −1 −1 −1 −1 0 0 6 −6 0 0 0 1 2 −2 −1 0 0 0 7 0 0 0

Second derivative −1 0 0 0 0 1 0 6 −12 6 0 0 1 1 −4 1 1 0 0 7 −7 0 0

The Laplacian is isotropic.

Direction dependent filters localize 1 pixel thick lines at other orientations (0, 45, 90).

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

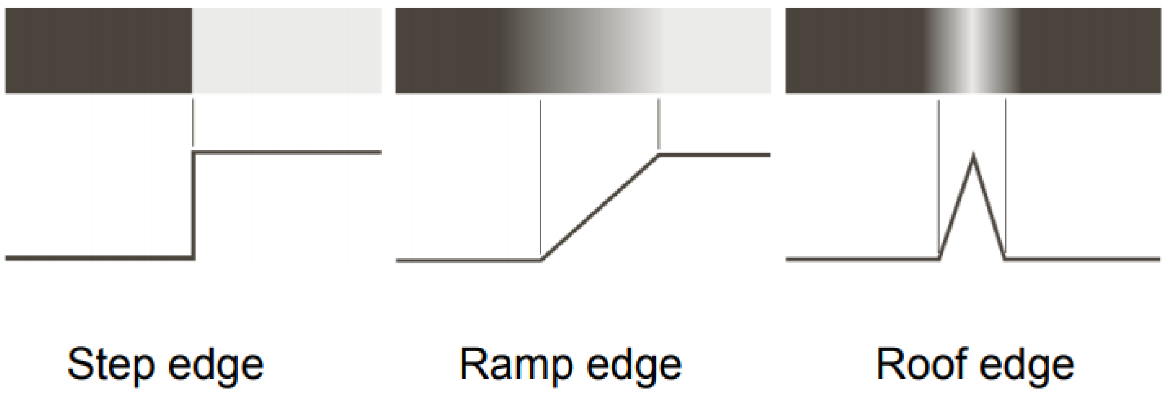
−1	−1	2
−1	2	−1
2	−1	−1

−45°

Edge models

– Ideally, edges should be 1 pixel thin.

– In practice, they are blurred and noisy



Marr-Hildereth algorithm: – Filter image with a nxn Gaussian low-pass filter – Compute the Laplacian of the filtered image using an appropriate mask – Find the zero crossings of this image This operator is based upon a 2nd derivative operator and can be scaled using the parameter to fit a particular image or application, i.e., small operators for sharp detail and large operators for blurry edges

Canny Edge operator

1. Smooth image with a Gaussian filter
2. Compute gradient magnitude M[x,y] and direction [x,y]
3. Apply non-maximal suppression to the gradient magnitude
4. Use double thresholding (and subsequent connectivity analysis) to detect and link edges