IMAGE COMPRESSION

Lempel-Ziv-Welch coding(LZW) assigns fixed-length code words to variable length sequences of source symbols. It requires no a priori knowledge of the probability of occurrence of the symbols to be coded.

Dictionary Location	Entry
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

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		

IMAGE COMPRESSION

Running-length coding(RLC/RLE) compresses image by representing runs of identical intensities as run-length pairs, where each run-length pair specifies the start of a new intensity and the number of consecutive pixels that have that intensity.

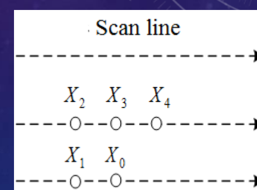
RLE is particularly effective when compressing binary images

000000000000000000000000000000000000000000000000000000000 (uncompressed)
57 zeros (compressed)

before——57 bytes after——2 bytes

IMAGE COMPRESSION

- Bit-plane coding
- Block transform coding
- Predictive coding

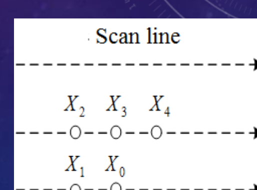


$$\hat{f}(x,y) = a_1 f(x,y-1) + a_2 f(x-1,y-1) + a_3 f(x-1,y) + a_4 f(x-1,y+1)$$

$$\hat{f}(x,y) = round\left[\sum_i \sum_j \sum_k a_{ijk} f(x-i, y-j, t-k)\right]$$

IMAGE COMPRESSION

- Predictive coding



$$\hat{f}_1(x,y) = 0.97 f(x,y-1)$$

$$\hat{f}_2(x,y) = 0.5 f(x,y-1) + 0.5 f(x-1,y)$$

$$\hat{f}_3(x,y) = 0.75 f(x,y-1) + 0.75 f(x-1,y) - 0.5 f(x-1,y-1)$$

$$\hat{f}(x,y) = \begin{cases} 0.97 f(x,y-1) & if \ | f(x-1,y) - f(x-1,y-1) | < | f(x,y-1) - f(x-1,y-1) | \\ 0.97 f(x-1,y) & otherwise \end{cases}$$

IMAGE COMPRESSION

- Predictive coding

$$\hat{f}(x,y) = round\left[\sum_{i=1}^{2} 0.5 f(x-i)\right]$$

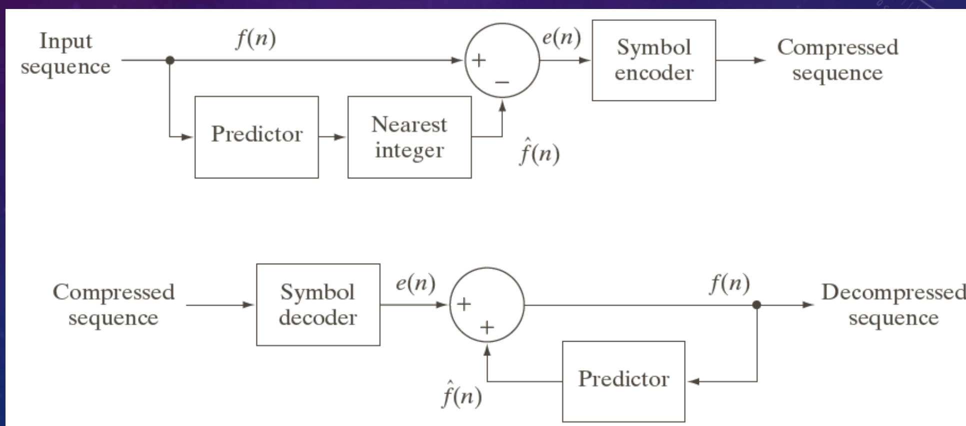


IMAGE COMPRESSION

- JPEG(Joint Photographic Expert Group)
- MPEG(Moving Picture Expert Group): MPEG-1、MPEG-2、MPEG-4

changshu@uestc.edu.cn

IMAGE COMPRESSION

- Digital image watermarking
  - Watermark is one or more items of information inserted into digital images.
  - Watermarked images protect the rights of digital media owners in a variety of ways, including copyright identification, user identification or fingerprinting, authenticity determination, automated monitoring, copy protection, etc.

IMAGE COMPRESSION

- Visible watermark
  - An opaque or semi-transparent sub-image or image is placed on top of another image

$$f_w = (1-\alpha) f + \alpha w$$

  - α controls the relative visibility of the watermark and the underlying image

IMAGE COMPRESSION

- Visible watermark (exp. 1)
  - Load logo.tif as wartemark imge w, load peppers.png as f
  - Create a watermarked image

$$f_w = (1-\alpha) f + \alpha w$$

  - Adjust the only parameter

IMAGE COMPRESSION

- Invisible watermark
  - Fragile invisible watermark(LSB watermarked image)

$$f_w = 4(f / 4) + w / 64$$

  - Cannot be seen with the naked eye.
  - Inserting watermark as visually redundant information
  - FIWs are destroyed by any modification of the images in which they are embeded

IMAGE COMPRESSION

- Invisible watermark (exp. 2)
  - Load logo.tif as wartemark imge w, load peppers.png as f
  - Created a watermarked image

$$f_w = 4(f/4) + w/64$$

  - Recover the watermark from the watermarked image
  - Modify the watermarked image and try to recover the watermark again

IMAGE COMPRESSION

- Invisible watermark
  - Robust invisible watermark
  - Mark insertion and extraction can be performed in spatial domain or in the transform domain.

IMAGE COMPRESSION

- Invisible watermark (exp.3)
  - Compute the 2D DCT of the image(peppers.png) to be watermarked
  - Locate its K largest coefficients $c_1, \ldots, c_i, \ldots c_K$ by magnitudes
  - Create a watermark by generating a K-element pseudo-random sequence of numbers, $w_1, \ldots, w_i, \ldots w_K$ taken from a Gaussian distribution with 0 mean and unit variance
  - Embed the watermark(logo.tif) from step 3 into the K largest DCT coefficients from step 2 using the following eqution

$$c_i' = c_i(1 + \alpha w_i) \qquad 1 \le i \le K$$

replace the original with the computed from above equation

  - Compute the inverse DCT of the result from step 4

IMAGE COMPRESSION

- Invisible watermark (exp.3)
  - Compute the 2D DCT of the image in question
  - Extract the K coefficients in the position corresponding to $c_1, \ldots, c_i, \ldots c_K$ of step 2 in the watermarking procedure and denote the coefficients as $\hat{c}_1, \ldots, \hat{c}_i, \ldots \hat{c}_K$
  - Compute watermark $\hat{w}_1, \ldots, \hat{w}_i, \ldots \hat{w}_K$, using

$$\hat{w}_i = \hat{c}_i - c_i \qquad 1 \le i \le K$$

  - Measure the similarity using the correlation

$$\gamma = \frac{\sum_{i=0}^{N-1} (\hat{w}_i - \bar{\hat{w}})(w_i - \bar{w})}{\sqrt{\sum_{i=1}^{K} (\hat{w}_i - \bar{\hat{w}})^2 \sum_{i=1}^{K} (w_i - \bar{w})^2}}$$

  - Compare the measured similarity and make a binary detection decision

SUMMARY

- fundamentals --- entropy, Shannon's first theorem, compression ratio, average coding length, coding efficiency, etc.
- Image data redundancy --- coding redundancy, spatial and temporal redundancy, irrelevant information
- Basic compression methods
  - Huffman coding
  - Arithmetic coding
  - LZW coding
  - Run-length coding
- Watermarking --- Visible Watermarking, Fragile Invisible Watermarking, Robust Invisible Watermarking