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

数字媒体压缩技术

Course Outline

10数字媒体压缩技术

10.1 无损压缩技术

10.2 有损压缩技术

10.3 图像JEPG压缩标准

10.4 视频图像MPEG压缩标准

10.5 音频压缩技术

图像JPEG编码

Spatial Frequency and DCT（空间频率和离散余弦变换）

- Spatial frequency indicates how many times pixel values change across an image block.（空间频率指示像素值变化次数）
- The DCT formalizes this notion with a measure of how much the image contents change in correspondence to the number of cycles of a cosine wave per block.（DCT公式化了图像内容的变化）
- The role of the DCT is to decompose the original signal into its DC and AC components; the role of the IDCT is to reconstruct (re-compose) the signal.（DCT将原始信号分解为直流和交流信号）

图像JPEG编码

Definition of DCT:

Given an input function f(i, j) over two integer variables i and j (a piece of an image), the 2D DCT transforms it into a new function F (u, v), with integer u and v running over the same range as i and j. The general definition of the transform is:

$$F(u,v) = \frac{2\,C(u)\,C(v)}{\sqrt{MN}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \cos\frac{(2i+1)\cdot u\pi}{2M} \cdot \cos\frac{(2j+1)\cdot v\pi}{2N} \cdot f(i,j) \tag{8.15}$$

where $i, u = 0, 1, \ldots, M-1$; $j, v = 0, 1, \ldots, N-1$; and the constants $C(u)$ and $C(v)$ are determined by

$$C(\xi) = \begin{cases} \frac{\sqrt{2}}{2} & if \quad \xi = 0, \\ 1 & otherwise. \end{cases} \tag{8.16}$$

图像JPEG编码

2D Discrete Cosine Transform (2D DCT):

$$F(u,v) = \frac{C(u)\,C(v)}{4}\sum_{i=0}^{7}\sum_{j=0}^{7}\cos\frac{(2i+1)u\pi}{16}\cos\frac{(2j+1)v\pi}{16}f(i,j) \qquad (8.17)$$

where $i, j, u, v = 0, 1, \ldots, 7$, and the constants $C(u)$ and $C(v)$ are determined by Eq. (8.5.16).

2D Inverse Discrete Cosine Transform (2D IDCT):

The inverse function is almost the same, with the roles of f(i, j) and F (u, v) reversed, except that now C(u)C(v) must stand inside the sums:

$$\tilde{f}(i,j) = \sum_{u=0}^{7}\sum_{v=0}^{7}\frac{C(u)\,C(v)}{4}\cos\frac{(2i+1)u\pi}{16}\cos\frac{(2j+1)v\pi}{16}F(u,v) \qquad (8.18)$$

where $i, j, u, v = 0, 1, \ldots, 7$.

The DCT is a linear transform:

In general, a transform T (or function) is linear, iff

$$\mathcal{T}(\alpha p + \beta q) = \alpha \mathcal{T}(p) + \beta \mathcal{T}(q) \qquad (8.21)$$

where α and β are constants, p and q are any functions, variables or constants.

From the definition in Eq. 8.17 or 8.19, this property can readily be proven for the DCT because it uses only simple arithmetic operations.

图像JPEG编码

The Cosine Basis Functions

- 正交基：Function Bp(i) and Bq(i) are orthogonal, if

$$\sum_i [B_p(i) \cdot B_q(i)] = 0 \qquad if \quad p \neq q \qquad (8.22)$$

- 标准正交基：Function Bp(i) and Bq(i) are orthonormal, if they are ortho

$$\sum_i [B_p(i) \cdot B_q(i)] = 1 \qquad if \quad p = q \qquad (8.23)$$

- It can be shown that:

$$\sum_{i=0}^{7} \left[\cos\frac{(2i+1)\cdot p\pi}{16} \cdot \cos\frac{(2i+1)\cdot q\pi}{16}\right] = 0 \quad if \quad p \neq q$$

$$\sum_{i=0}^{7} \left[\frac{C(p)}{2}\cos\frac{(2i+1)\cdot p\pi}{16} \cdot \frac{C(q)}{2}\cos\frac{(2i+1)\cdot q\pi}{16}\right] = 1 \quad if \quad p = q$$

图像JPEG编码

问题？

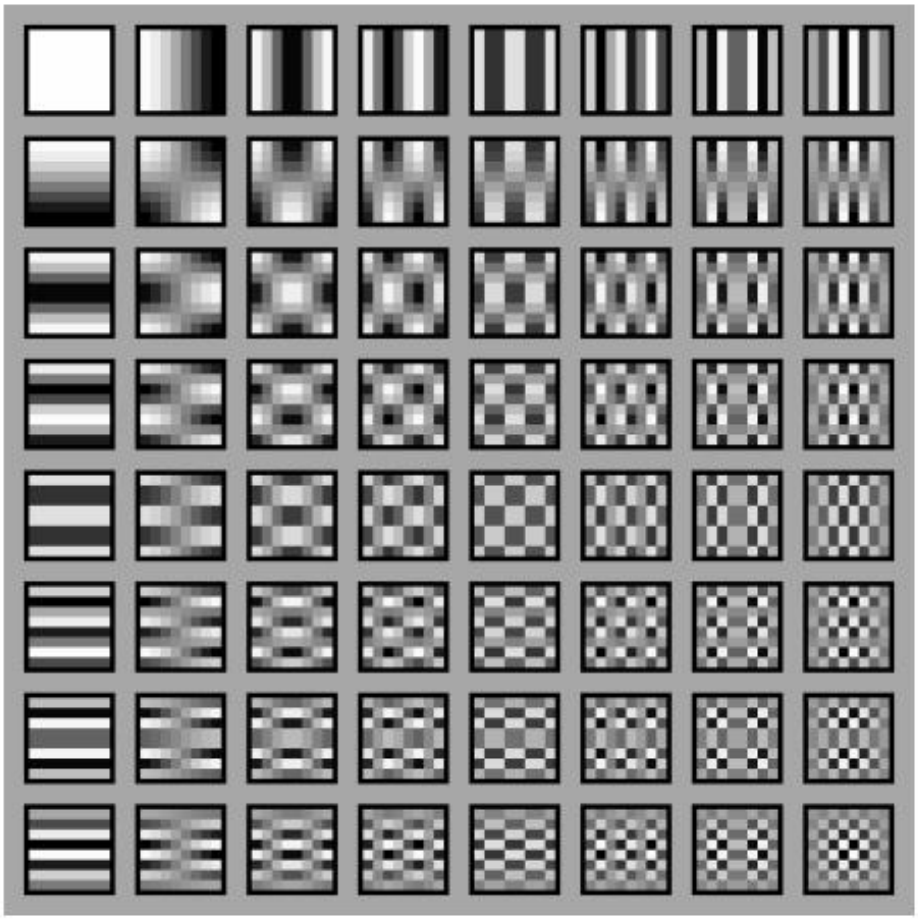
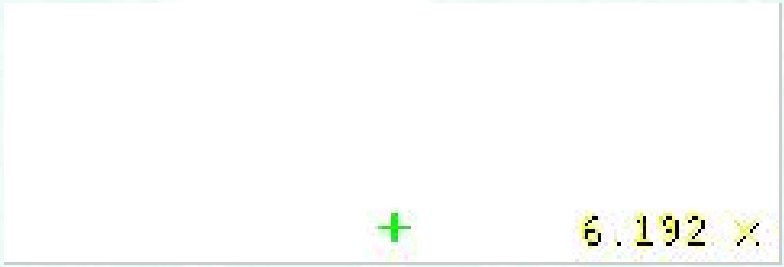


Fig. 8.9: Graphical Illustration of 8×8 2D DCT basis.

JPEG图像

- 图像分成8x8像素块，在JPG上做的所有操作都是基于8x8像素块



上面是一张正在重建的图片(最左边那个区域)。每一帧我们都使用右侧面版新的基准值，来乘一个权重值(右侧区域的文字)来产生图片(中间区域)。

图像JPEG编码

The JPEG Standard

- JPEG is an image compression standard that was developed by the "Joint Photographic Experts Group". JPEG was formally accepted as an international standard in 1992.
- JPEG is a 有损图像压缩 method. It employs a **transform coding** method using the DCT (Discrete Cosine Transform).
- An image is a function of i and j (or conventionally x and y) in the spatial domain.

  The 2D DCT is used as one step in JPEG in order to yield a frequency response which is a function F (u, v) in the spatial frequency domain, indexed by two integers u and v.

JPEG图像压缩的特性

问题？

Observations for JPEG Image Compression

- The effectiveness of the DCT transform coding method in JPEG relies on 3 major observations:

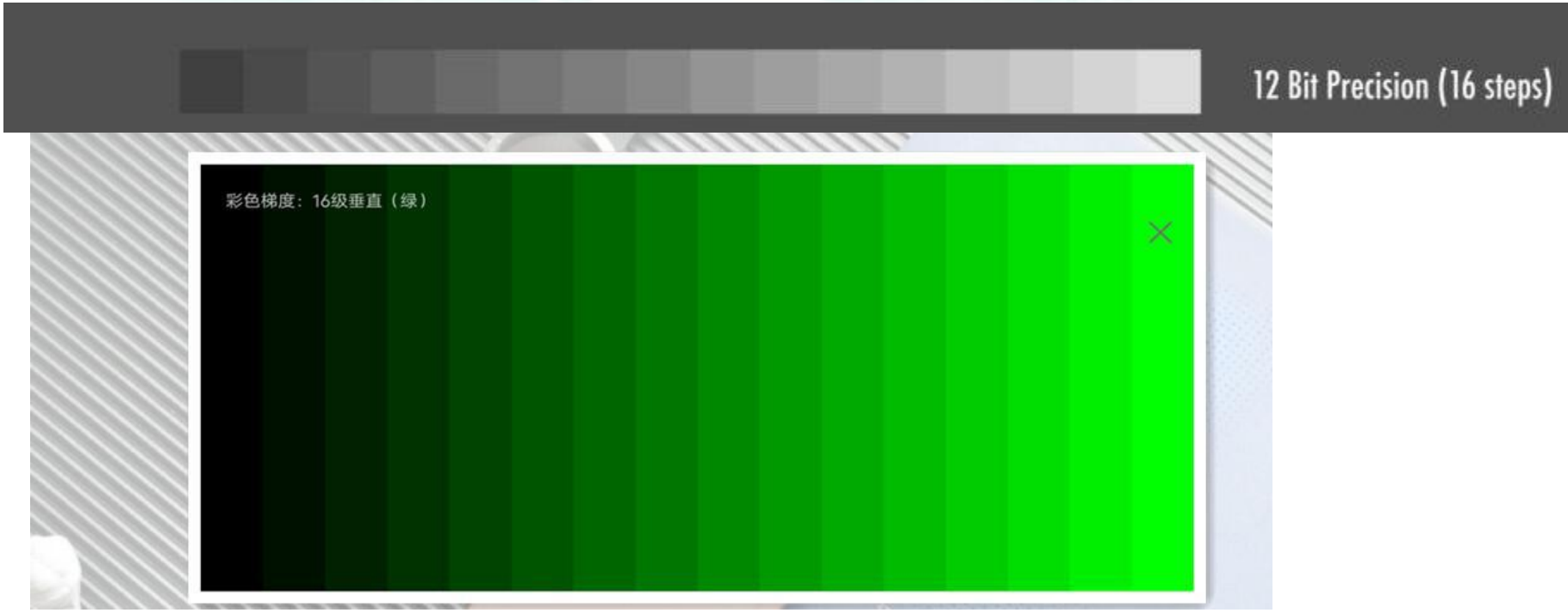
- **Observation 1**: 有用的图像内容变化相对缓慢, i.e., it is unusual for intensity values to vary widely several times in a small area, for example, within an 8 × 8 image block.
  - much of the information in an image is repeated, hence “spatial redundancy”.

Observations for JPEG Image Compression

- **Observation 2**: 心理学实验表明，在空间域内，人类对高频分量损失的感知能力远远低于对低频分量损失的感知能力。
  - the spatial redundancy can be reduced by largely reducing the high spatial frequency contents.

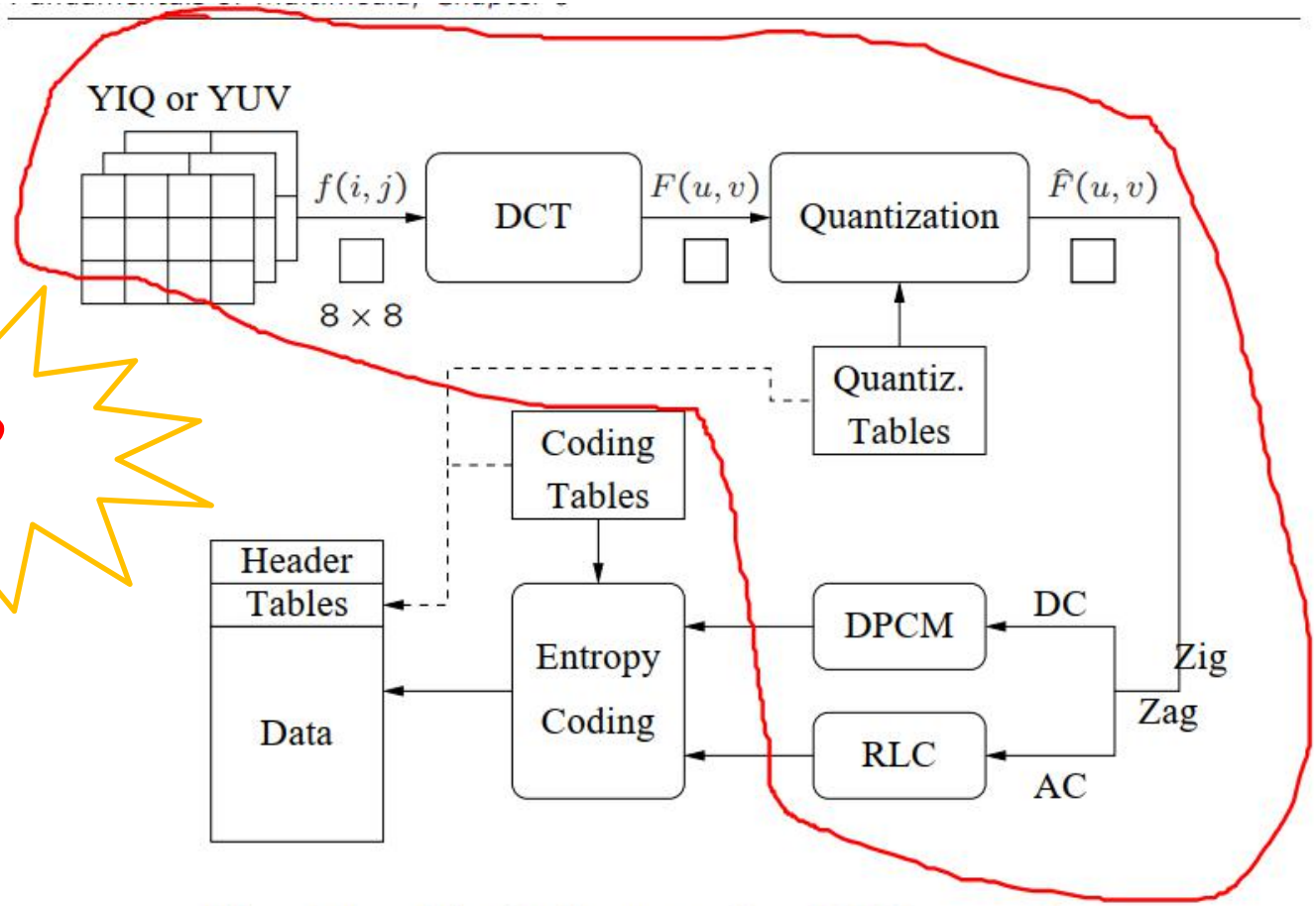
图像JPEG编码

- **Observation 3**: 人类对灰度（黑和白）的视觉敏感度（区分相近空间线的准确度）要远远高于对彩色的敏感度。
  - chroma subsampling (4:2:0) is used in JPEG.



图像JPEG编码

Observations for JPEG Image Compression



问题？

Fig. 9.1: Block diagram for JPEG encoder.

图像JPEG编码

Main Steps in JPEG Image Compression

- Transform RGB to YIQ or YUV and subsample color.
- DCT on image blocks.
- Quantization.
- Zig-zag ordering and run-length encoding.
- Entropy coding.

DCT on image blocks

- Each image is divided into 8 × 8 blocks. The 2D DCT is applied to each block image f(i, j), with output being the DCT coefficients F(u, v) for each block.

- Using blocks, however, has the effect of isolating each block from its neighboring context. This is why JPEG images look choppy (“blocky”) when a high compression ratio is specified by the user.

图像JPEG编码

Quantization

$$\hat{F}(u,v) = round\left(\frac{F(u,v)}{Q(u,v)}\right) \qquad (9.1)$$

- F (u, v) represents a DCT coefficient, Q(u, v) is a “quantization matrix” entry, and $\hat{F}$(u, v) represents the quantized DCT coefficients which JPEG will use in the succeeding entropy coding.
  - **The quantization step is the main source for loss in JPEG compression.**
  - The entries of Q(u, v) tend to have larger values towards the lower right corner. This aims to introduce more loss at the higher spatial frequencies —— a practice supported by Observations 1 and 2.
  - Table 9.1 and 9.2 show the default Q(u, v) values obtained from psychophysical studies with the goal of maximizing the compression ratio while minimizing perceptual losses in JPEG images.

图像JPEG编码

Table 9.1 The Luminance Quantization Table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Table 9.2 The Chrominance Quantization Table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

图像JPEG编码

An 8×8 block from the Y image of 'Lena'

200	202	189	188	189	175	175	175
200	203	198	188	189	182	178	175
203	200	200	195	200	187	185	175
200	200	200	200	197	187	187	187
200	205	200	200	195	188	187	175
200	200	200	200	200	190	187	175
205	200	199	200	191	187	187	175
210	200	200	200	188	185	187	186

$f(i,j)$

515	65	-12	4	1	2	-8	5
-16	3	2	0	0	-11	-2	3
-12	6	11	-1	3	0	1	-2
-8	3	-4	2	-2	-3	-5	-2
0	-2	7	-5	4	0	-1	-4
0	-3	-1	0	4	1	-1	0
3	-2	-3	3	3	-1	-1	3
-2	5	-2	4	-2	2	-3	0

$F(u,v)$

Fig. 9.2: JPEG compression for a smooth image block.

图像JPEG编码

32	6	-1	0	0	0	0	0
-1	0	0	0	0	0	0	0
-1	0	1	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\hat{F}(u,v)$

512	66	-10	0	0	0	0	0
-12	0	0	0	0	0	0	0
-14	0	16	0	0	0	0	0
-14	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\tilde{F}(u,v)$

199	196	191	186	182	178	177	176
201	199	196	192	188	183	180	178
203	203	202	200	195	189	183	180
202	203	204	203	198	191	183	179
200	201	202	201	196	189	182	177
200	200	199	197	192	186	181	177
204	202	199	195	190	186	183	181
207	204	200	194	190	187	185	184

$\tilde{f}(i,j)$

1	6	-2	2	7	-3	-2	-1
-1	4	2	-4	1	-1	-2	-3
0	-3	-2	-5	5	-2	2	-5
-2	-3	-4	-3	-1	-4	4	8
0	4	-2	-1	-1	-1	5	-2
0	0	1	3	8	4	6	-2
1	-2	0	5	1	1	4	-6
3	-4	0	6	-2	-2	2	2

$\epsilon(i,j) = f(i,j) - \tilde{f}(i,j)$

Fig. 9.2 (cont'd): JPEG compression for a smooth image block.

图像JPEG编码

Another 8×8 block from the Y image of 'Lena'

70	70	100	70	87	87	150	187
85	100	96	79	87	154	87	113
100	85	116	79	70	87	86	196
136	69	87	200	79	71	117	96
161	70	87	200	103	71	96	113
161	123	147	133	113	113	85	161
146	147	175	100	103	103	163	187
156	146	189	70	113	161	163	197

$f(i,j)$

-80	-40	89	-73	44	32	53	-3
-135	-59	-26	6	14	-3	-13	-28
47	-76	66	-3	-108	-78	33	59
-2	10	-18	0	33	11	-21	1
-1	-9	-22	8	32	65	-36	-1
5	-20	28	-46	3	24	-30	24
6	-20	37	-28	12	-35	33	17
-5	-23	33	-30	17	-5	-4	20

$F(u,v)$

Fig. 9.3: JPEG compression for a textured image block.

图像JPEG编码

-5	-4	9	-5	2	1	1	0
-11	-5	-2	0	1	0	0	-1
3	-6	4	0	-3	-1	0	1
0	1	-1	0	1	0	0	0
0	0	-1	0	0	1	0	0
0	-1	1	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\hat{F}(u,v)$

-80	-44	90	-80	48	40	51	0
-132	-60	-28	0	26	0	0	-55
42	-78	64	0	-120	-57	0	56
0	17	-22	0	51	0	0	0
0	0	-37	0	0	109	0	0
0	-35	55	-64	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\tilde{F}(u,v)$

70	60	106	94	62	103	146	176
85	101	85	75	102	127	93	144
98	99	92	102	74	98	89	167
132	53	111	180	55	70	106	145
173	57	114	207	111	89	84	90
164	123	131	135	133	92	85	162
141	159	169	73	106	101	149	224
150	141	195	79	107	147	210	153

$\tilde{f}(i,j)$

0	10	-6	-24	25	-16	4	11
0	-1	11	4	-15	27	-6	-31
2	-14	24	-23	-4	-11	-3	29
4	16	-24	20	24	1	11	-49
-12	13	-27	-7	-8	-18	12	23
-3	0	16	-2	-20	21	0	-1
5	-12	6	27	-3	2	14	-37
6	5	-6	-9	6	14	-47	44

$\epsilon(i,j) = f(i,j) - \tilde{f}(i,j)$

Fig. 9.3 (cont'd): JPEG compression for a textured image block.

图像JPEG编码

Run-length Coding (RLC) on AC coefficients

- RLC aims to turn the $\hat{F}(u; v)$ values into sets {#-zeros-to-skip , next non-zero value}.
- To make it most likely to hit a long run of zeros: a zig-zag scan is used to turn the 8×8 matrix $\hat{F}(u; v)$ into a 64-vector.

问题？

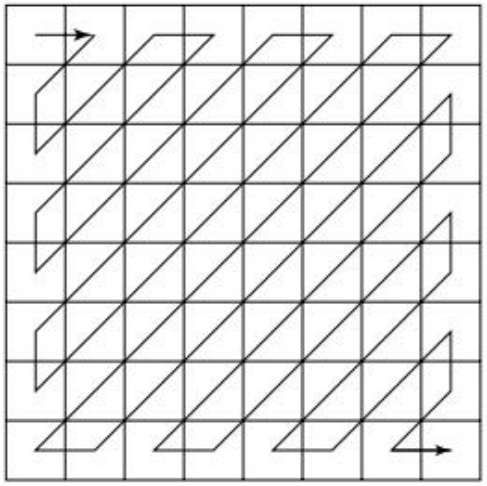


Fig. 9.4: Zig-Zag Scan in JPEG.

DPCM on DC coefficients

- The DC coefficients are coded separately from the AC ones. Differential Pulse Code Modulation (DPCM) is the coding method.

- If the DC coefficients for the first 5 image blocks are 150, 155, 149, 152, 144, then the DPCM would produce 150, 5, -6, 3, -8, assuming $d_i = DC_{i+1} - DC_i$, $and\ d_0 = DC_0$.

图像JPEG编码

Entropy Coding

- The DC and AC coefficients finally undergo an entropy coding step to gain a possible further compression.
- Use DC as an example: each DPCM coded DC coefficient is represented by (SIZE, AMPLITUDE), where SIZE indicates how many bits are needed for representing the coefficient, and AMPLITUDE contains the actual bits.
- In the example we're using, codes 150, 5, −6, 3, −8 will be turned into (8, 10010110), (3, 101), (3, 001), (2, 11), (4, 0111) .
- SIZE is Huffman coded since smaller SIZEs occur much more often. AMPLITUDE is not Huffman coded, its value can change widely so Huffman coding has no appreciable benefit.

图像JPEG编码

Table 9.3 Baseline entropy coding details – size category.

SIZE	AMPLITUDE
1	-1, 1
2	-3, -2, 2, 3
3	-7..-4, 4..7
4	-15..-8, 8..15
.	.
.	.
.	.
10	-1023..-512, 512..1023

压缩率：10

压缩率：50

运动补偿

4.1-1 the first frame

图 4.1-2 the second frame

问题？

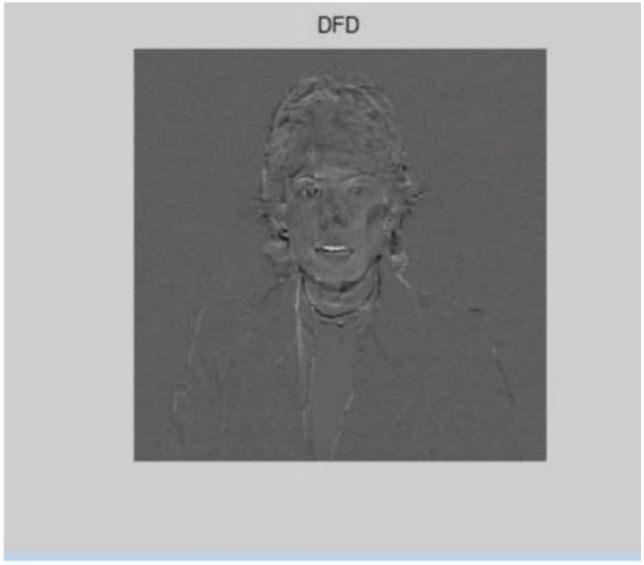


图 4.1-4 DFD

视频图像MPEG压缩标准

Introduction to Video Compression

- A video consists of a time-ordered sequence of frames, i.e., images.
- An obvious solution to video compression would be predictive coding based on previous frames.

  Compression proceeds by subtracting images: subtract in time order and code the residual error.
- It can be done even better by searching for just the right parts of the image to subtract from the previous frame.

视频图像MPEG压缩标准

Video Compression with Motion Compensation

- Consecutive frames in a video are similar | temporal redundancy exists.
- **Temporal redundancy** （时间冗余） is exploited so that not every frame of the video needs to be coded independently as a new image.

  The difference between the current frame and other frame(s) in the sequence will be coded | small values and low entropy, good for compression.
- Steps of Video compression based on Motion Compensation (MC):
  1. Motion Estimation (motion vector search).
  2. MC-based Prediction.
  3. Derivation of the prediction error, i.e., the difference.

Motion Compensation

- Each image is divided into macroblocks of size N × N.
  - By default, N = 16 for luminance images. For chrominance images, N = 8 if 4:2:0 chroma subsampling is adopted.
- Motion compensation is performed at the macroblock level.
  - The current image frame is referred to as Target Frame.
  - A match is sought between the macroblock in the Target Frame and the most similar macroblock in previous and/or future frame(s) (referred to as Reference frame(s)).
  - The displacement of the reference macroblock to the target macroblock is called a motion vector MV.
  - Figure 10.1 shows the case of forward prediction in which the Reference frame is taken to be a previous frame.

视频图像MPEG压缩标准

Motion Compensation

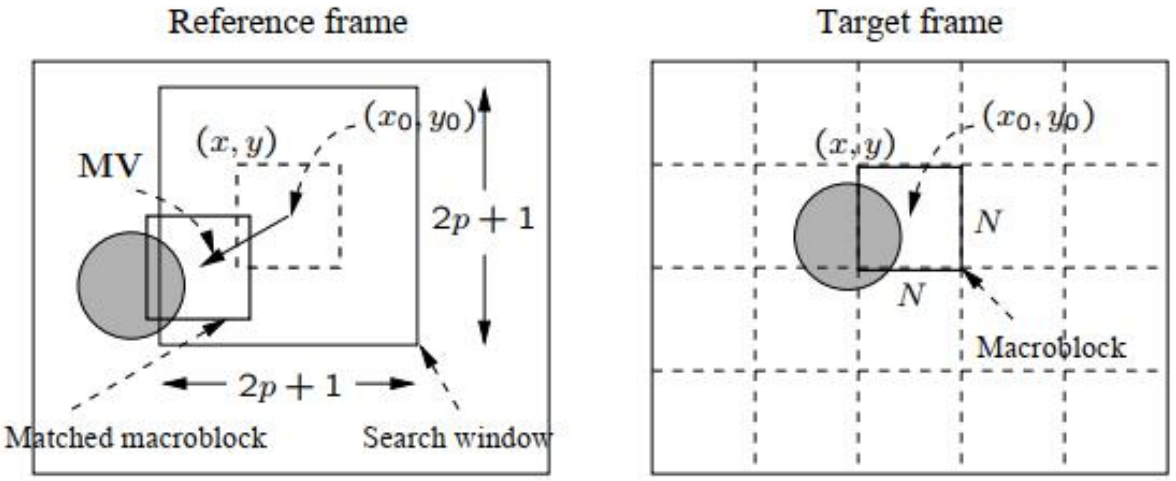


Fig. 10.1: Macroblocks and Motion Vector in Video Compression.

- MV search is usually limited to a small immediate neighborhood — — both horizontal and vertical displacements in the range [−p, p]:

  This makes a search window of size (2p + 1) × (2p + 1).

Search for Motion Vectors

- The difference between two macroblocks can then be measured by their Mean Absolute Difference (MAD):

$$MAD(i,j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)| \qquad (10.1)$$

N – size of the macroblock,
k and l – indices for pixels in the macroblock,
i and j – horizontal and vertical displacements,
$C(x+k, y+l)$ – pixels in macroblock in Target frame,
$R(x+i+k, y+j+l)$ – pixels in macroblock in Reference frame.

- The goal of the search is to find a vector (i, j) as the motion vector MV = (u, v), such that MAD(i, j) is minimum:

$$(u,v) = [\ (i,j)\ |\ MAD(i,j) \ \text{ is minimum}, \ i \in [-p,p], \ j \in [-p,p]\ ] \qquad (10.2)$$

Sequential Search

- **Sequential search**: sequentially search the whole (2p + 1) × (2p + 1) window in the Reference frame (also referred to as Full search).
  - a macroblock centered at each of the positions within the window is compared to the macroblock in the Target frame pixel by pixel and their respective MAD is then derived using Eq. (10.1).
  - The vector (i, j) that offers the least MAD is designated as the MV (u, v) for the macroblock in the Target frame.
  - sequential search method is very costly | assuming each pixel comparison requires three operations (subtraction, absolute value, addition), the cost for obtaining a motion vector for a single macroblock is $(2p+1)\cdot(2p+1)\cdot N^2\cdot 3 \Rightarrow O(p^2N^2)$.

视频图像MPEG压缩标准

Motion-vector: sequential-search

```
begin
  min_MAD = LARGE_NUMBER;        /* Initialization */
  for i = -p to p
    for j = -p to p
      {
        cur_MAD = MAD(i, j);
        if cur_MAD < min_MAD
          {
            min_MAD = cur_MAD;
            u = i;        /* Get the coordinates for MV. */
            v = j;
          }
      }
end
```

视频图像MPEG压缩标准

Overview

- **MPEG**: Moving Pictures Experts Group, established in 1988 for the development of digital video.
- It is appropriately recognized that proprietary interests need to be maintained within the family of MPEG standards:
  - Accomplished by defining only a compressed bitstream that implicitly defines the decoder.
  - The compression algorithms, and thus the encoders, are completely up to the manufacturers.

视频图像MPEG压缩标准

MPEG-1

- MPEG-1 adopts the CCIR601 digital TV format also known as SIF (Source Input Format).
- MPEG-1 supports only non-interlaced video. Normally, its picture resolution is:
  - 352 × 240 for NTSC video at 30 fps
  - 352 × 288 for PAL video at 25 fps
  - It uses 4:2:0 chroma subsampling
- The MPEG-1 standard is also referred to as ISO/IEC 11172. It has five parts: 11172-1 Systems, 11172-2 Video, 11172-3 Audio, 11172-4 Conformance, and 11172-5 Software.

Motion Compensation in MPEG-1

- Motion Compensation (MC) based video encoding in H.261 works as follows:
  - In Motion Estimation (ME), each macroblock (MB) of the Target P-frame is assigned a best matching MB from the previously coded I or P frame - **prediction**.
  - **prediction error**: The difference between the MB and its matching MB, sent to DCT and its subsequent encoding steps.
  - The prediction is from a previous frame —— **forward prediction**.

视频图像MPEG压缩标准

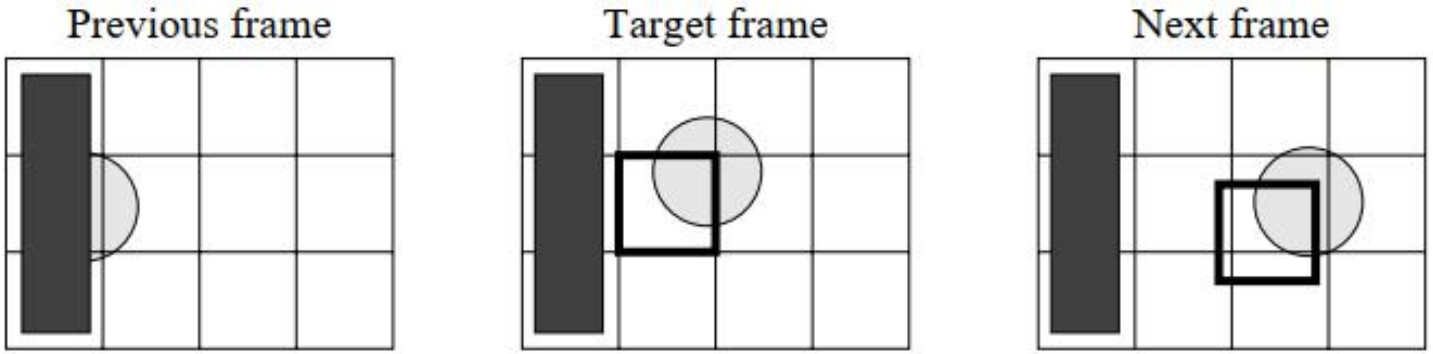


Fig 11.1: The Need for Bidirectional Search.

The MB containing part of a ball in the Target frame cannot find a good matching MB in the previous frame because half of the ball was occluded by another object. A match however can readily be obtained from the next frame.

问题？

视频图像MPEG压缩标准

Motion Compensation in MPEG-1

- MPEG introduces a third frame type | B-frames, and its accompanying bi-directional motion compensation.
- The MC-based B-frame coding idea is illustrated in Fig. 11.2:
  - Each MB from a B-frame will have up to two motion vectors (MVs) (one from the forward and one from the backward prediction).
  - If matching in both directions is successful, then two MVs will be sent and the two corresponding matching MBs are averaged (indicated by '%' in the figure) before comparing to the Target MB for generating the prediction error.
  - If an acceptable match can be found in only one of the reference frames, then only one MV and its corresponding MB will be used from either the forward or backward prediction.

视频图像MPEG压缩标准

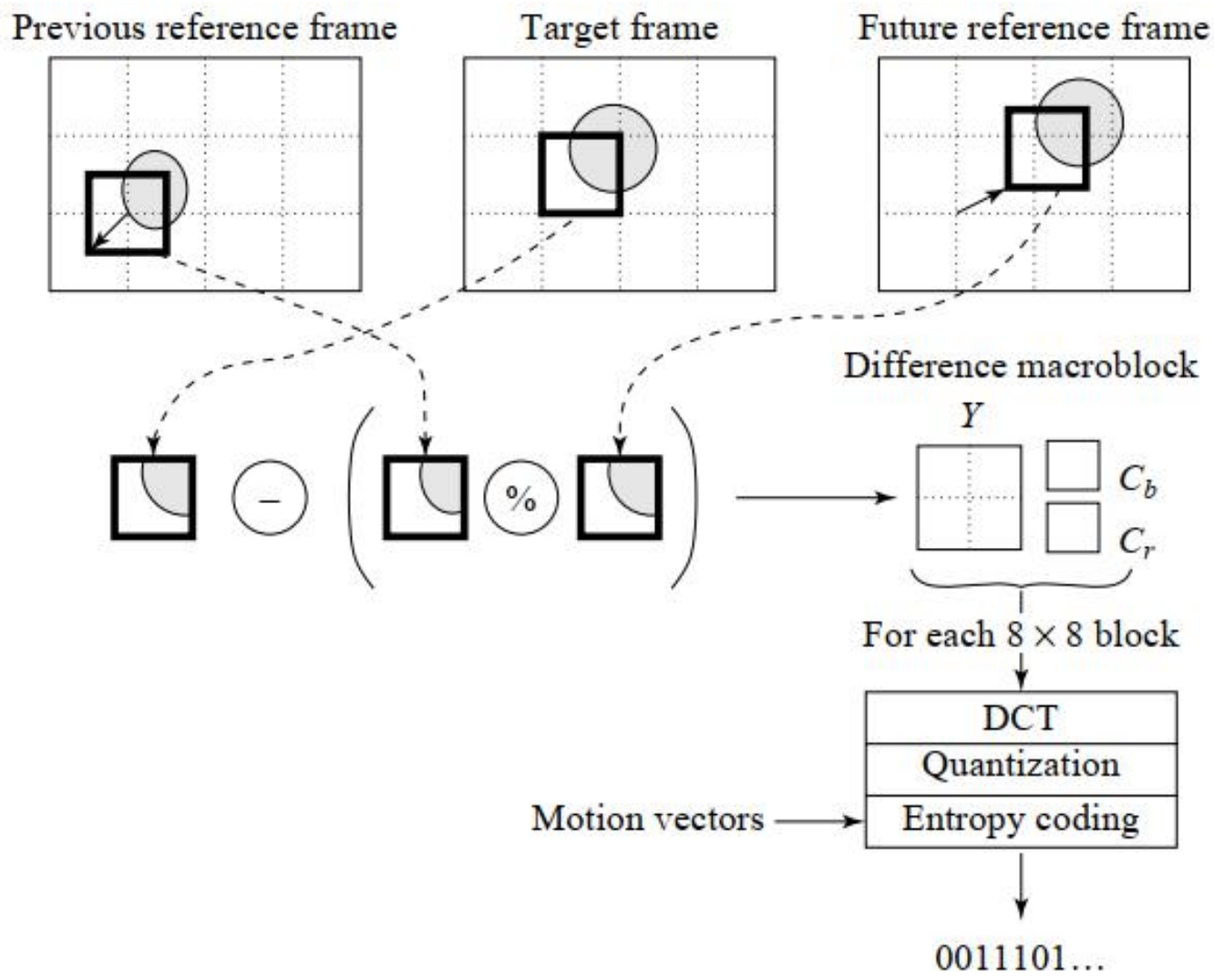


Fig 11.2: B-frame Coding Based on Bidirectional Motion Compensation.

视频图像MPEG压缩标准

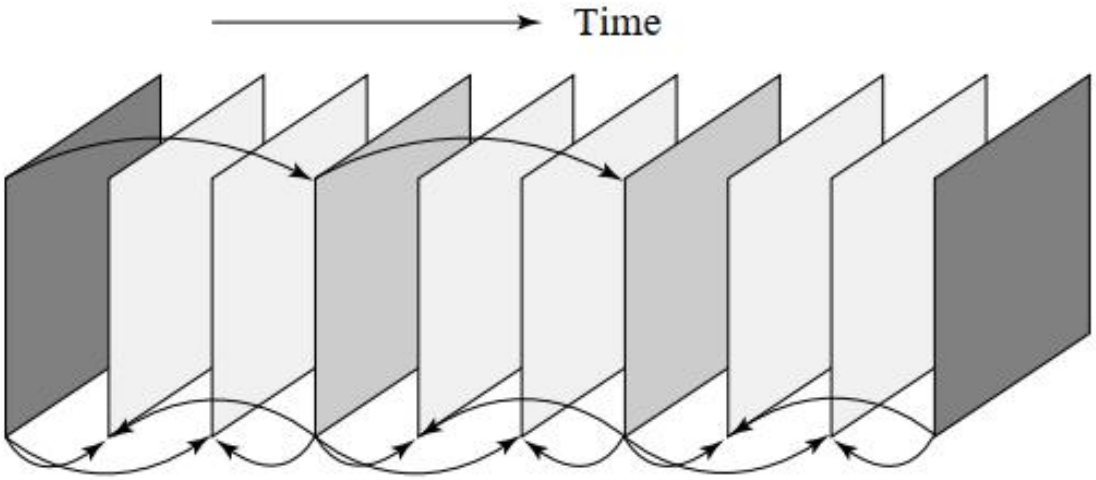


Fig 11.3: MPEG Frame Sequence.

G.726 ADPCM

- ITU G.726 supersedes ITU standards G.721 and G.723.
- **Rationale**: works by adapting a fixed quantizer in a simple way. The different sizes of codewords used amount to bitrates of 16 kbps, 24 kbps, 32 kbps, or 40 kbps, at 8 kHz sampling rate.
- The standard defines a multiplier constant α that will change for every difference value, e_n, depending on the current scale of signals. Define a scaled difference signal g_n as follows:

$$\begin{aligned} e_n &= s_n - \hat{s}_n \,, \\ g_n &= e_n/\alpha \,, \end{aligned} \qquad (13.1)$$

$\hat{s}_n$ is the predicted signal value. g_n is then sent to the quantizer for quantization.

G.726 ADPCM

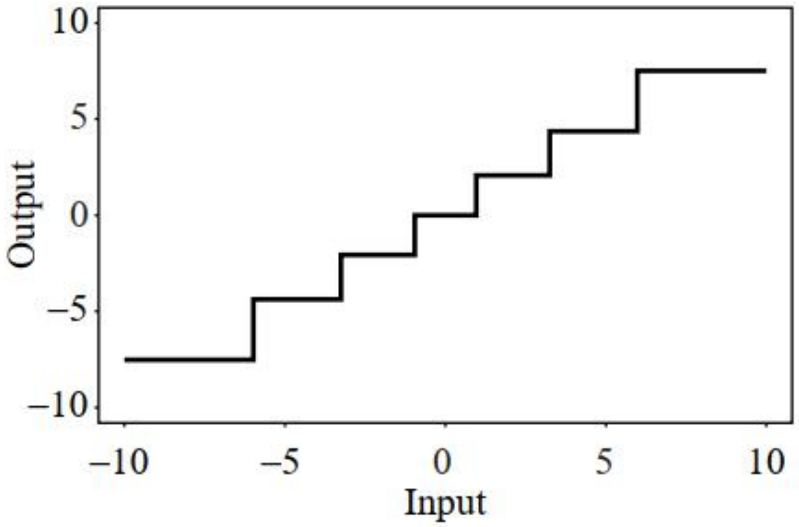


Fig. 13.2: G.726 Quantizer

- The input value is a ratio of a difference with the factor α.
- By changing α, the quantizer can adapt to change in the range of the difference signal —— a backward adaptive quantizer.

Backward Adaptive Quantizer

- **Backward adaptive** works in principle by noticing either of the cases:
  - too many values are quantized to values far from zero -- would happen if quantizer step size in f were too small.
  - too many value fall close to zero too much of the time -- would happen if the quantizer step size were too large.
- **Jayant quantizer** allows one to adapt a backward quantizer step size after receiving just one single output.
  - Jayant quantizer simply expands the step size if the quantized input is in the outer levels of the quantizer, and reduces the step size if the input is near zero.

The Step Size of Jayant Quantizer

- Jayant quantizer assigns multiplier values Mk to each level, with values smaller than unity for levels near zero, and values larger than 1 for the outer levels.
- For signal f_n, the quantizer step size Δ is changed according to the quantized value k, for the previous signal value f_{n-1}, by the simple formula

$$\Delta \leftarrow M_k \Delta \qquad (13.2)$$

- Since it is the quantized version of the signal that is driving the change, this is indeed a backward adaptive quantizer.

Vocoders

- **Vocoders** —— voice coders, which cannot be usefully applied when other analog signals, such as modem signals, are in use.
  - concerned with modeling speech so that the salient features are captured in as few bits as possible.
  - use either a model of the speech waveform in time (LPC(Linear Predictive Coding) vocoding), or ... →
  - break down the signal into frequency components and model these (channel vocoders and formant vocoders).
- Vocoder simulation of the voice is not very good yet.

Phase Insensitivity （相位不敏感性）

- A complete reconstituting of speech waveform is really unnecessary, perceptually: all that is needed is for the amount of energy at any time to be about right, and the signal will sound about right.
- **Phase** is a shift in the time argument inside a function of time.
  - Suppose we strike a piano key, and generate a roughly sinusoidal sound $\cos(\omega t)$, with $\omega = 2\pi f$.
  - Now if we wait sufficient time to generate a phase shift $\pi/2$ and then strike another key, with sound $\cos(2\omega t + \pi/2)$, we generate a waveform like the solid line in Fig. 13.3.
  - This waveform is the sum $\cos(\omega t) + \cos(2\omega t + \pi/2)$.

Phase Insensitivity

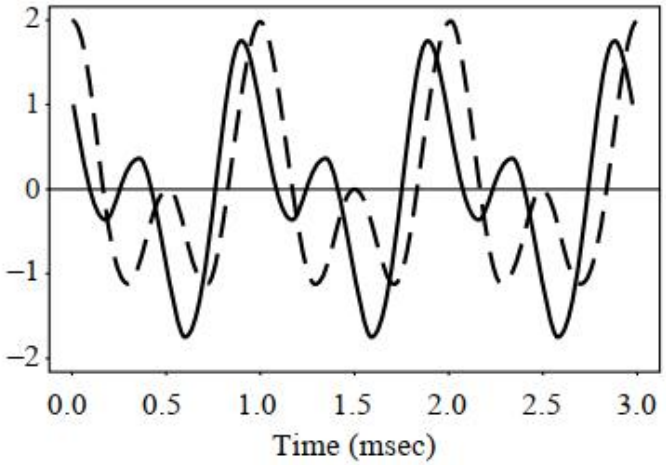


Fig. 13.3: Solid line: Superposition of two cosines, with a phase shift. Dashed line: No phase shift. The wave is very different, yet the sound is the same, perceptually.

- If we did not wait before striking the second note, then our waveform would be $\cos(\omega t) + \cos(2\omega t)$. But perceptually, the two notes would sound the same sound, even though in actuality they would be shifted in phase.

Channel Vocoder （通道声音合成器）

- Vocoders can operate at low bit-rates, 1-2 kbps. To do so, a channel vocoder first applies a filter bank to separate out the different frequency components:

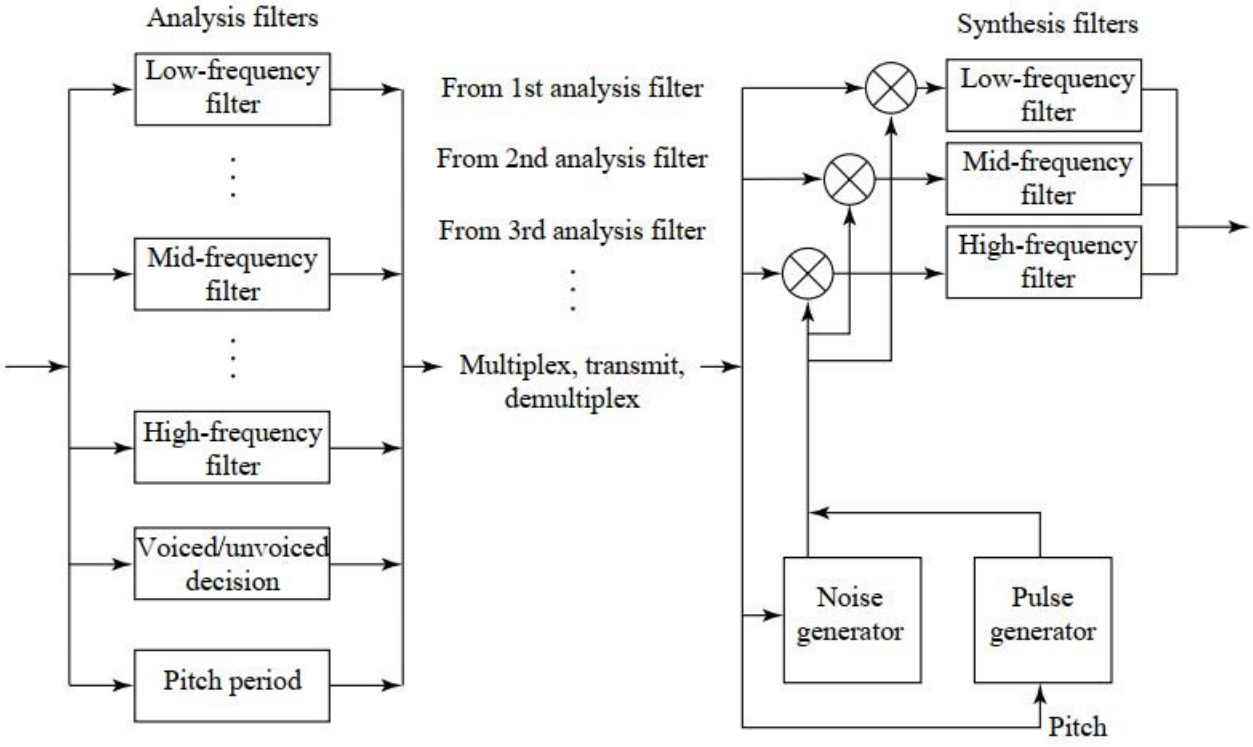


Fig 13.4: Channel Vocoder

Channel Vocoder

- Due to Phase Insensitivity (i.e. only the energy is important):
  - The waveform is “rectified” to its absolute value.
  - The filter bank derives relative power levels for each frequency range.
  - A subband coder would not rectify the signal, and would use wider frequency bands.
- A channel vocoder also analyzes the signal to determine the general pitch of the speech (low —— bass, or high —— tenor), and also the excitation of the speech.
- A channel vocoder applies a vocal tract transfer model to generate a vector of excitation parameters that describe a model of the sound, and also guesses whether the sound is voiced or unvoiced.

Format Vocoder

- **Formants**: the salient frequency components that are present in a sample of speech, as shown in Fig 13.5.
- Rationale: encoding only the most important frequencies.

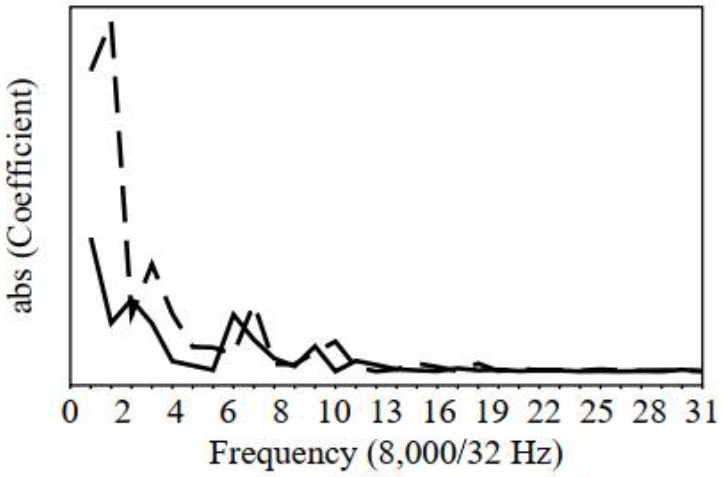


Fig. 13.5: The solid line shows frequencies present in the first 40 msec of the speech sample in Fig. 6.15. The dashed line shows that while similar frequencies are still present one second later, these frequencies have shifted.