CS5342: Multimedia Computing and Applications

Lecture 1

Introduction to Multimedia Computing

Class Organization

Roger Zimmermann

15 January 2016

Basic Information

- Lecturer
  - Roger Zimmermann (rogerz@comp.nus.edu.sg) & (dcsrz@nus.edu.sg)
    - Office: AS6 #05-05
    - Office hours:
      - By appointment (6516-7949)
- TA
  - Wang Yuhui (wangyuhui@u.nus.edu)
- Web-page:

  IVLE; and

  http://www.comp.nus.edu.sg/~cs5342

Prerequisites

- Pre-requisites:
  - CS5240 *Theoretical Foundations of Multimedia* (**desirable**)
  - If you have never done any image or video processing until now, you should not take this course.

- Implementation Skills: ***necessary but it can be on*** ANY ***platform which can be demonstrated (MATLAB/C/C++/C#/Java/…)***

Grading Policy

- **Weightage** Allocation:
  - ***20% Survey Paper (due Wednesday 10 February 2016)***
    - Will require reading all recent papers in one topic
    - Understand and critique the area and write it as a paper
    - Can be strategically used as a basis for deciding the project
  - ***20% Assignment (due Monday 7 March 2016)***
    - A combination of theory/practice requiring theoretical understanding and then implementation of some technique or algorithm
    - Can be strategically used as a baseline technique for comparison in the project
  - ***60% Project (due Monday 11 April 2016)***
- Regular reading homework
  - **Nothing** to be submitted and will **not** be graded
  - But **essential** for proper understanding of the course

Teaching Resources

- Textbook
  - None
  - Additional reading material will be provided

- Attending the lectures is compulsory but not monitored

Important Notice

- Any class which I cannot take will either be re-scheduled (subject to the consensus of the entire class) or will be converted into an e-learning lecture
- **eLearning Week: None**
- Need to decide about the following lecture:
  - ***Good Friday: 25 March 2015***
  - Any unanticipated future re-scheduling will be done similarly

Syllabus

- **Introduction to Multimedia Computing (1 Lecture)**
- **Content-based Retrieval (3 Lectures)**
- **Multimedia Content Processing (2 Lectures)**
- **Multimedia Surveillance (1 Lecture)**
- **Multimedia Summarization (1 Lecture)**
- **Multimedia Data Mining (1 Lecture)**
- **Multimedia Security (1 Lecture)**
- **Computational MM Advertisement (1 Lecture)**
- **Current Issues & Trends (project presentations in the last lecture)**

(Complete lecture schedule is available on the ~cs5342 website)

Project Types

- **Project**: will be individual.
- **Compulsory** – if you do not submit the project, you get a **F** grade.
- It is a very significant component of the grade.
- It needs to have a major implementation component.

Project Flexibility

- Project can be combined with
  - Your MS/PhD thesis work
  - Or your specific research interest
  - Or some other course project
  - Or your company's work
  - Check with me if you need assistance
- But you need to tell this to **all** the parties concerned and obtain consent.
  - Failure to do this may lead to loss of full credit for this component of the grade.

Project Planning

- Process (one possible way)
  - Select a recent multimedia systems paper
  - Completely understand it
  - Implement it fully
  - Do thorough testing and verify claims
  - If possible, suggest & implement extension or a novel approach for the same thing
    - Necessary for obtaining the top grade
  - Has to be of sufficient complexity
  - Simple/standard implementations not acceptable
- Then submit project

Project Logistics

- Initial proposal (due Friday, 5 February 2016)
  - 1 or 2 pages proposal of topic with Schedule
  - Will be finalized in 1 or 2 weeks after iterative feedback
- Intermediate report (due Friday, 11 March 2016)
  - Maximum 5 pages progress report
  - Sanity check to see how it is going
  - Not to be graded but submission is mandatory
- Final report (due Monday, 11 April 2016)
  - ACM paper style report (softcopy)
  - 20 minutes PowerPoint presentation (softcopy)
  - Code + test data (softcopy)
  - **Demo and Presentation in class will be required**

- **Details have been posted on the course homepage**

CS5342: Multimedia Computing and Applications

Lecture 1

Introduction to Multimedia Computing

Fundamentals

Roger Zimmermann

15 January 2016

- Human Perceptual System is composed of senses:
  - Visual: input
  - Acoustical: input/output
  - Haptic: input/output (touch, skin sensors, motor system)
  - Taste
  - Smell
- Brain is the processing, controlling and coordination center for the senses

- Human Senses are perfectly coordinated into a **fully** integrated system
- This system can receive and process multimodal information from the senses and produce multimedia output effortlessly (so it appears!)
- Many encoding levels of information – from basic to abstract:
  - sounds, speech-language, music, gestures, reading, writing
- How this is done in the brain, we do not know exactly how...
- **Human beings are <u>amazing</u> multimedia systems**

Binary 1-bit Lena Image

Gray-scale 8-bit Lena Image

Color 24-bit Mandrill Image

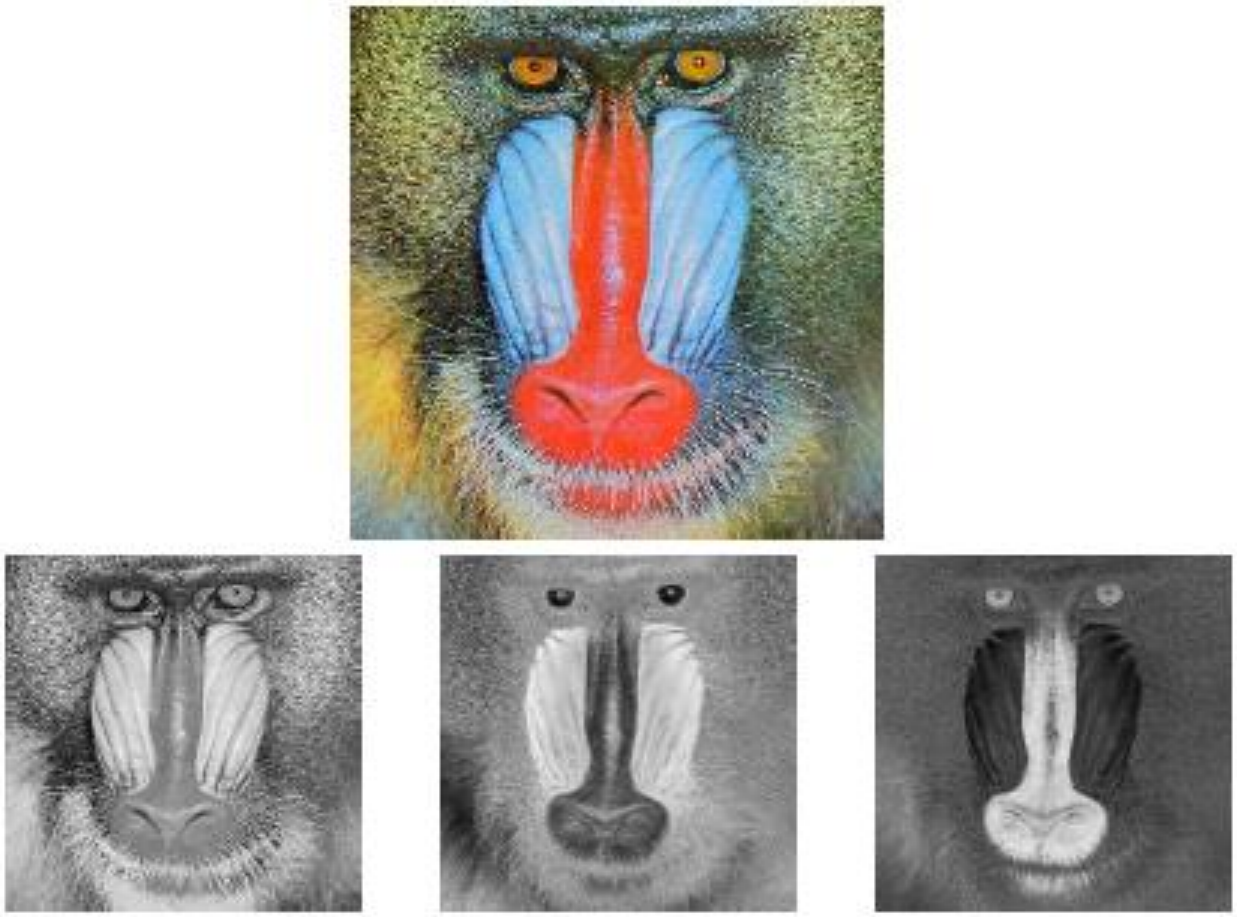


Fig. 4.18: $Y'UV$ decomposition of color image. Top image (a) is original color image; (b) is Y'; (c,d) are (U, V)

YUV Color Model

(a) YUV codes a luminance signal (for gamma-corrected signals) equal to Y' in Eq. (4.20). the "luma".

(b) **Chrominance** refers to the difference between a color and a reference white at the same luminance. $\longrightarrow$ use color differences U, V:

$$U = B' - Y', \qquad V = R' - Y' \tag{4.27}$$

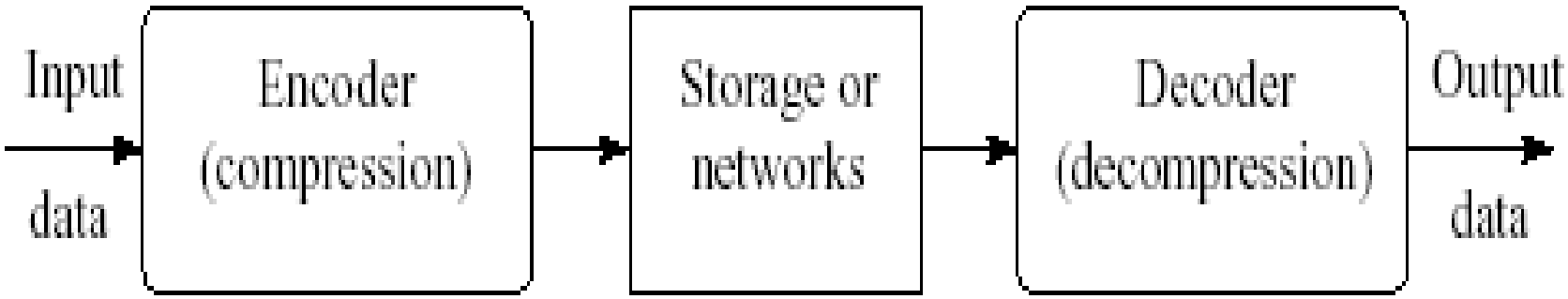
From Eq. (4.20),

$$\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{4.28}$$

(c) For gray, $R' = G' = B'$, the luminance Y' equals to that gray, since $0.299 + 0.587 + 0.114 = 1.0$. And for a gray ("black and white") image, the chrominance (U, V) is zero.

General Data Compression Scheme

- **Compression:** the process of coding that will effectively reduce the total number of bits needed to represent certain information.



Types of Redundancy

- Coding Redundancy
  - information-theoretic basis
- Statistical Redundancy
  - transform coding
- Perceptual Redundancy
  - heart of **multimedia** compression

Block-based Transform Coding

- Encoder
  - Step-1 Divide an image into *m* x *m* blocks and perform transform
  - Step-2 Determine bit-allocation for coefficients
  - Step-3 Design quantizer and quantize coefficients (lossy!)
  - Step-4 Encode quantized coefficients
- Decoder

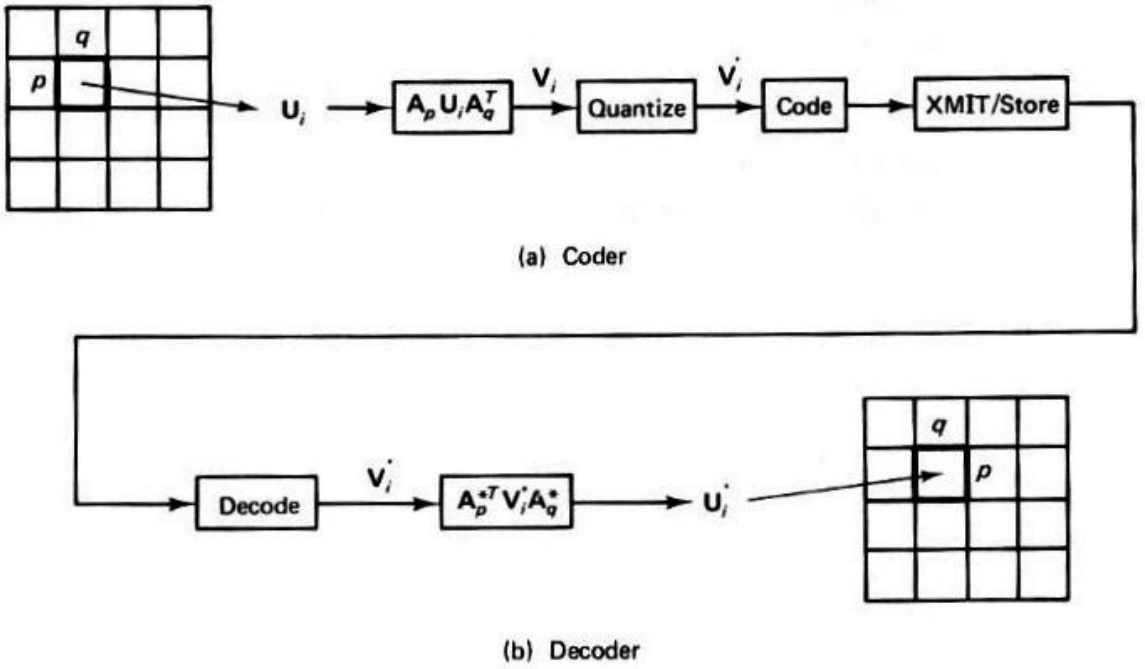


Figure 11.17 Two-dimensional transform coding.

Spatial Frequency and DCT

- *Spatial frequency* indicates how many times pixel values change across an image block.
- 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.
- 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

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

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

2D DCT

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

Cosine Basis Functions

- Function $B_p(i)$ and $B_q(i)$ are *orthogonal*, if

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

- Function $B_p(i)$ and $B_q(i)$ are *orthonormal*, if they are orthogonal and

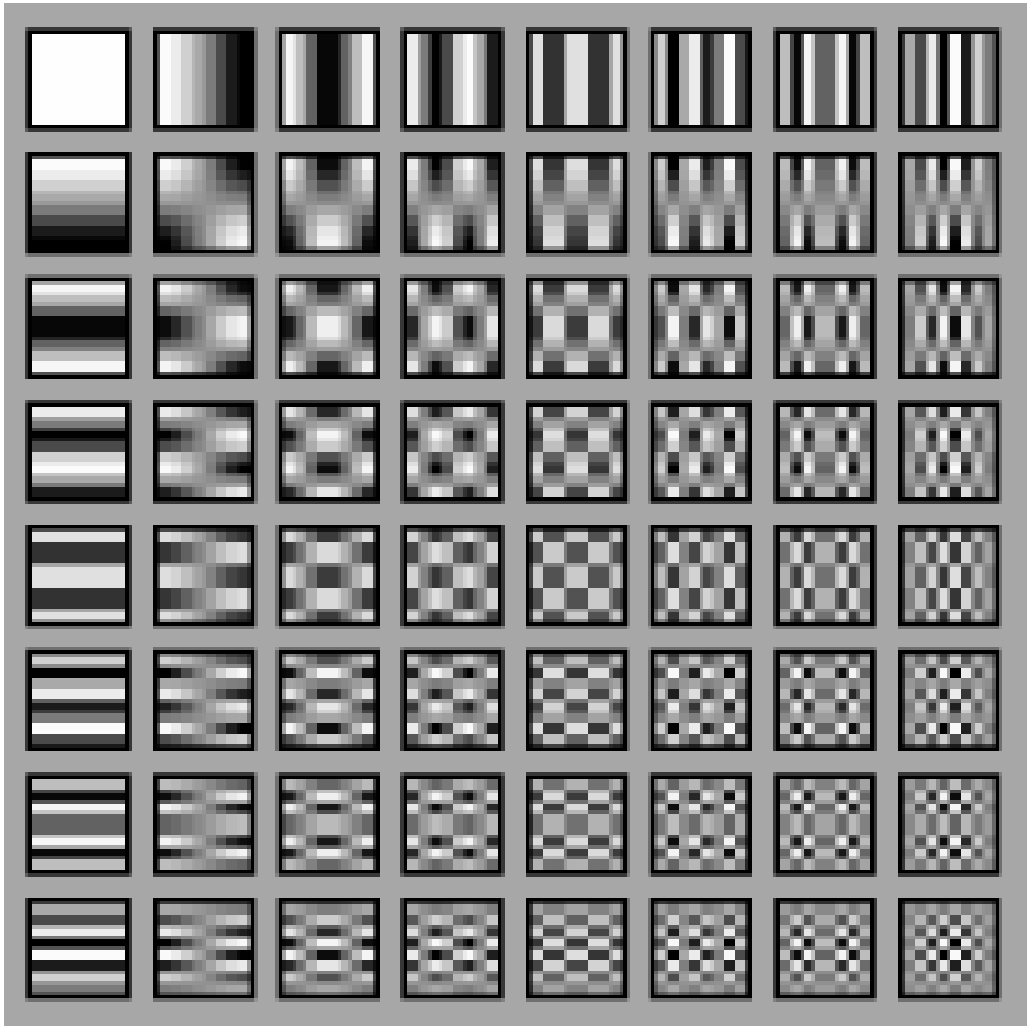
$$\sum_i [B_p(i) \cdot B_q(i)] = 1 \qquad if \quad p = q \tag{8.23}$$

- It can be shown that:

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

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

2D Cosine Basis Functions



The JPEG Standard 1

- 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 **lossy** image compression 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.

The JPEG Standard 2

- Allow for lossy and lossless encoding of still images
  - Part-1 DCT-based lossy compression
    - average compression ratio 15:1
  - Part-2 Predictive-based lossless compression
- Sequential, Progressive, Hierarchical modes
  - Sequential ~ *encoded in a single left-to-right, top-to-bottom scan*
  - Progressive ~ *encoded in multiple scans to first produce a quick, rough decoded image when the transmission time is long*
  - Hierarchical ~ *encoded at multiple resolution to allow accessing low resolution without full decompression*

Main Steps for JPEG

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

Baseline JPEG Algorithm

- "Baseline"
  - Simple, lossy compression
    - Subset of other DCT-based modes of JPEG standard
- A few basics
  - 8x8 block-DCT based coding
  - Shift to zero-mean by subtracting 128 ➔ [-128, 127]
    - Allows using signed integer to represent both DC and AC coeff.
  - Color (YCbCr / YUV) and downsample
    - Color components can have lower spatial resolution than luminance

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

  - Interleaving color components

Block Diagram of JPEG Baseline

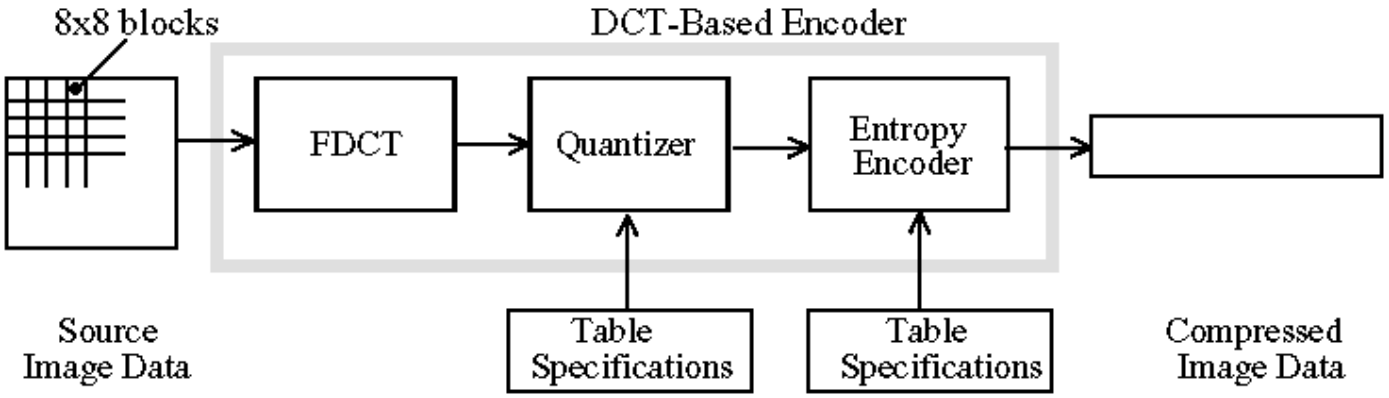


Figure 1. DCT-Based Encoder Processing Steps

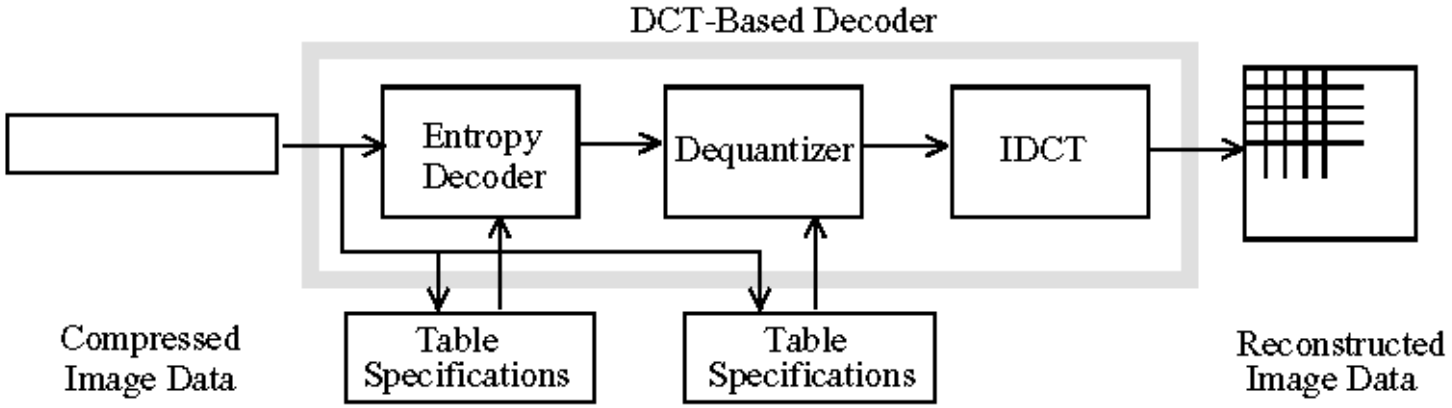
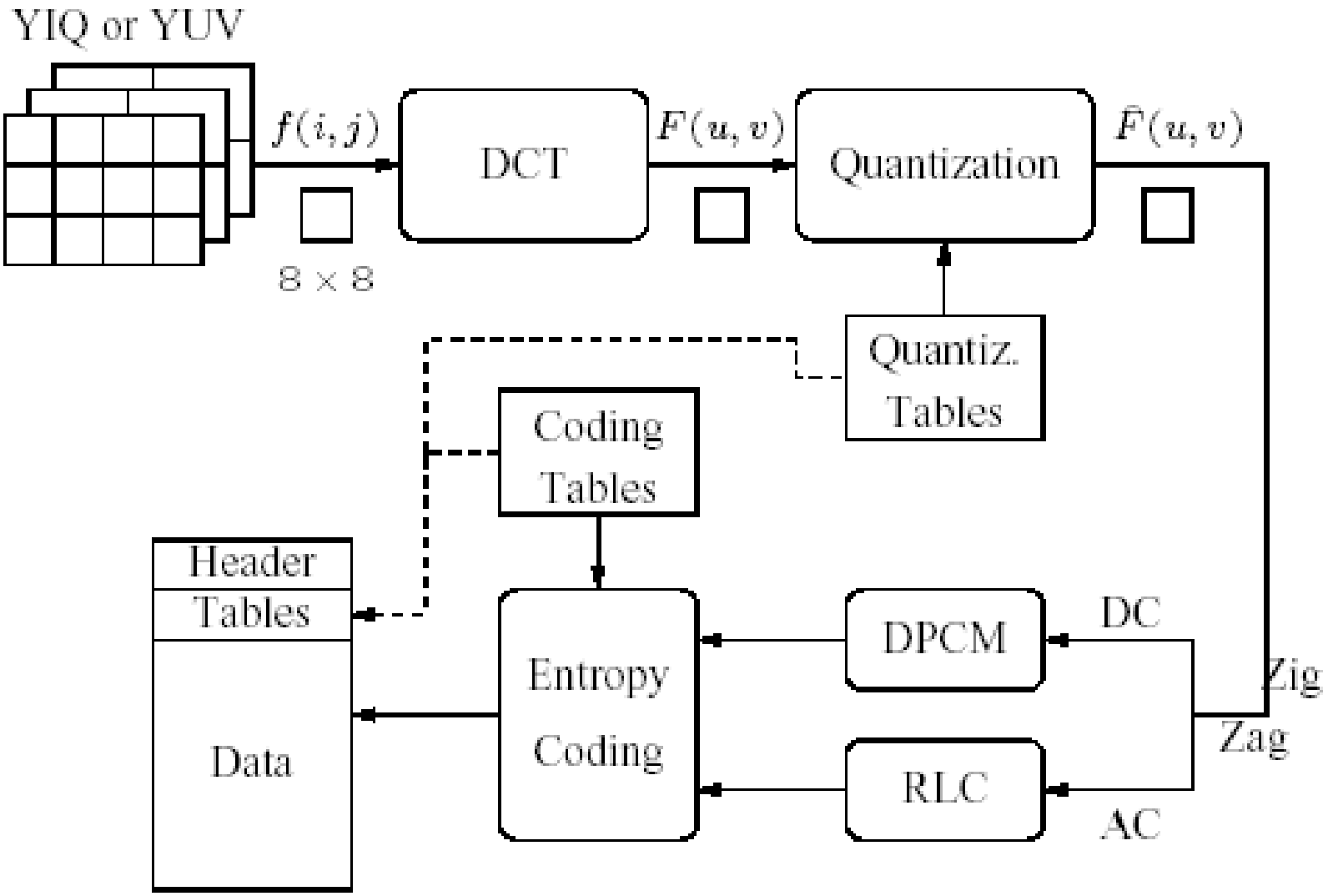
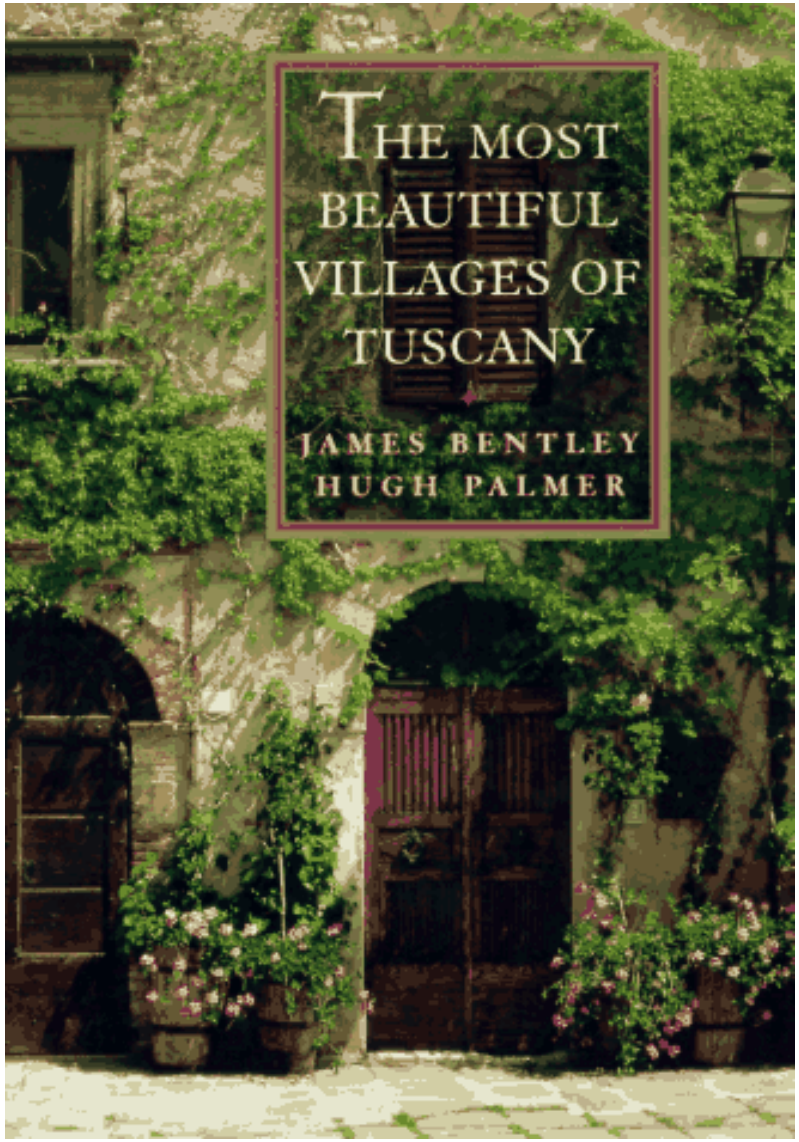


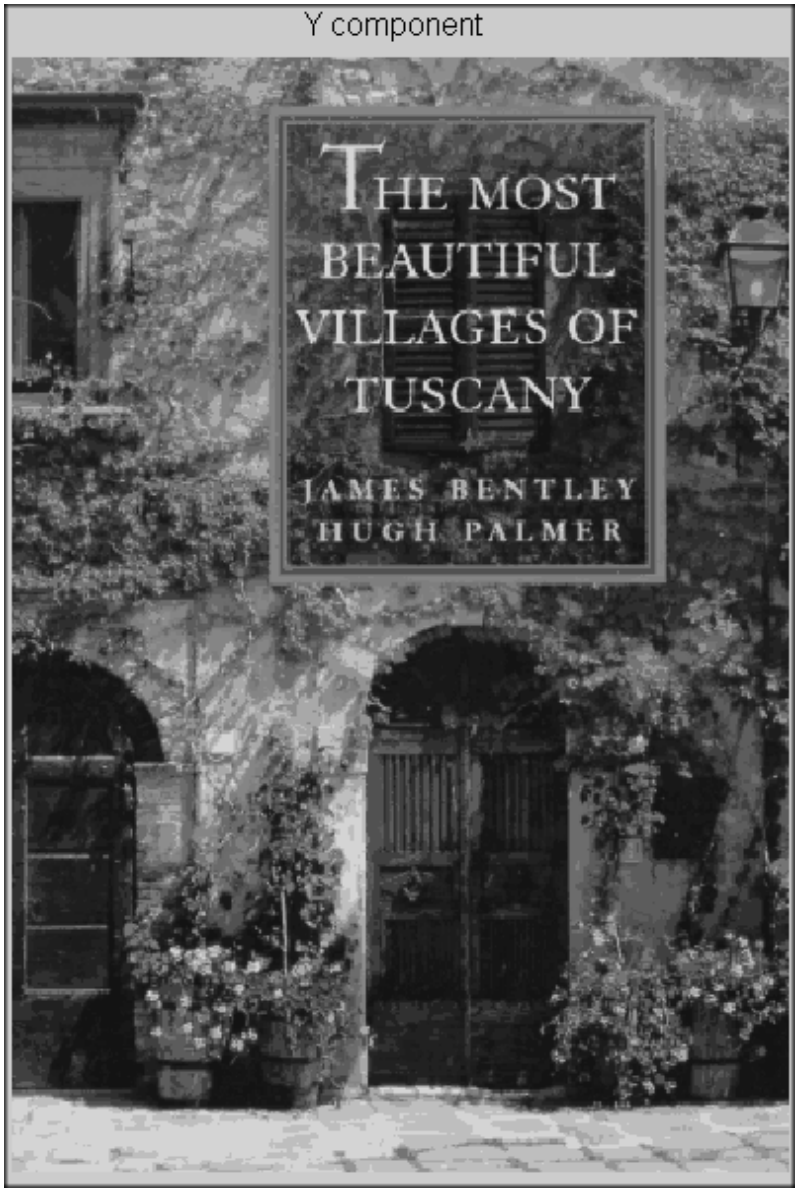
Figure 2. DCT-Based Decoder Processing Steps

JPEG Encoder



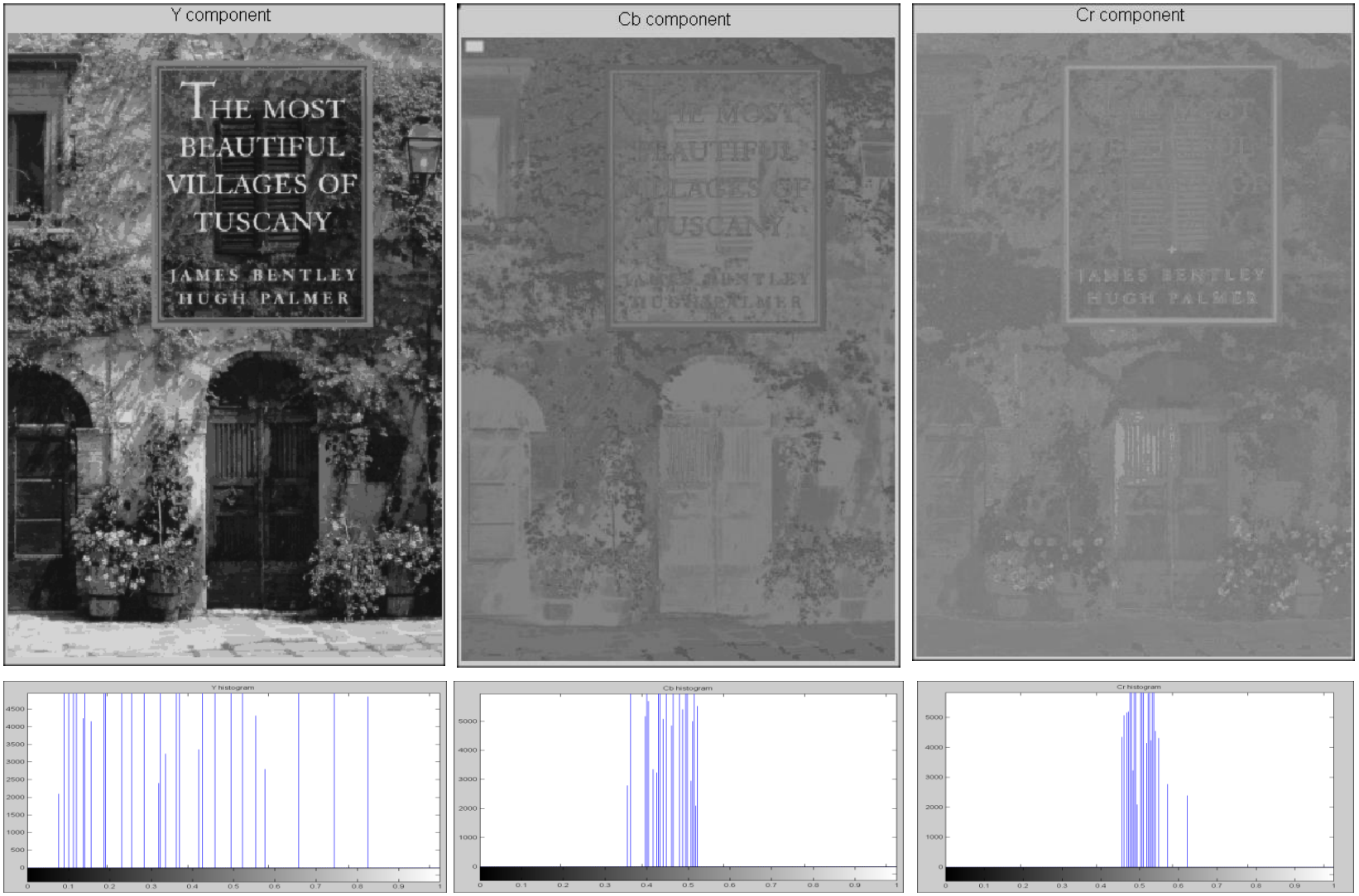


475 x 330 x 3 = 157 KB



Luminance

Y U V (Y Cb Cr) Components



Assign more bits to Y, less bits to Cb and Cr

Block DCT

- 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.

JPEG Quantization

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

- $F(u,v)$ 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.

Quantization Tables

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

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

$F(u,v)$

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

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

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

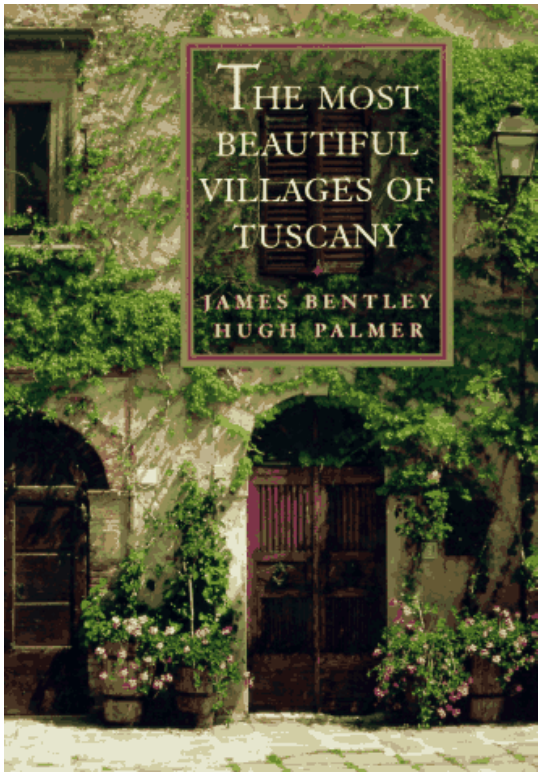
$\epsilon(i,j) = f(i,j) - \tilde{f}(i,j)$

Lossy Part in JPEG

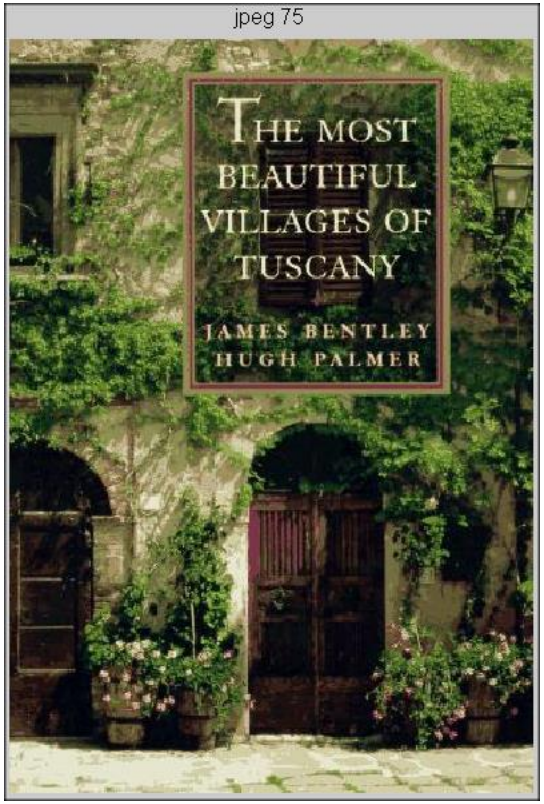
- Adaptive bit allocation scheme
  - Different quantization step size for different coefficient bands
  - Use same quantization matrix for all blocks in one image
  - Choose quantization matrix to best suit the image
  - Different quantization matrices for luminance and color components
- Default quantization table
  - “Generic” over a variety of images
- Quality factor “Q”
  - Scale the quantization table
  - Medium quality Q = 50% ~ no scaling
  - High quality Q = 100% ~ unit quantization step size
  - Poor quality ~ small Q, larger quantization step
    - visible artifacts like ringing and blockiness


Uncompr
JPEG 75% (
JPEG 50% (
JPEG 30% (9
JPEG 10% (5KB)

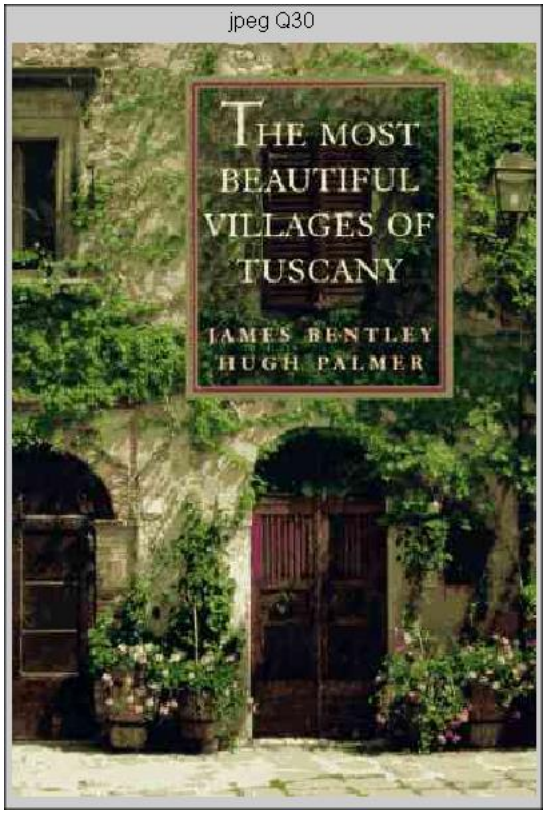
JPEG Example (Q=75% & 30%)



157 KB

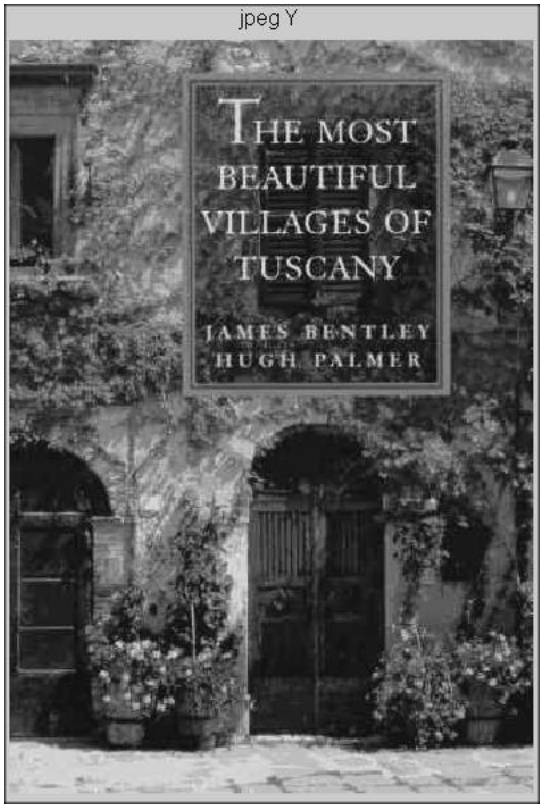


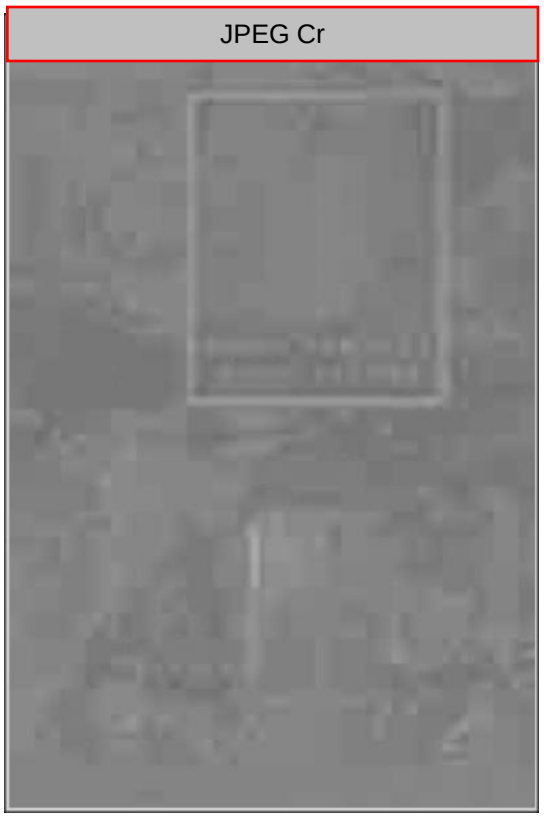
45 KB



22 KB

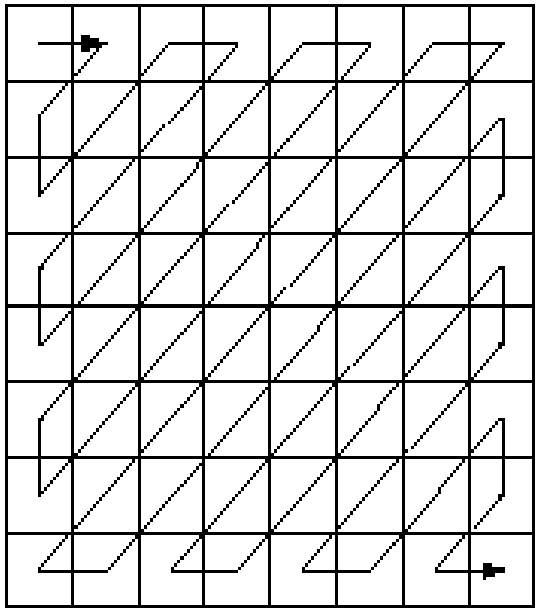
Y Cb Cr After JPEG (Q=30%)



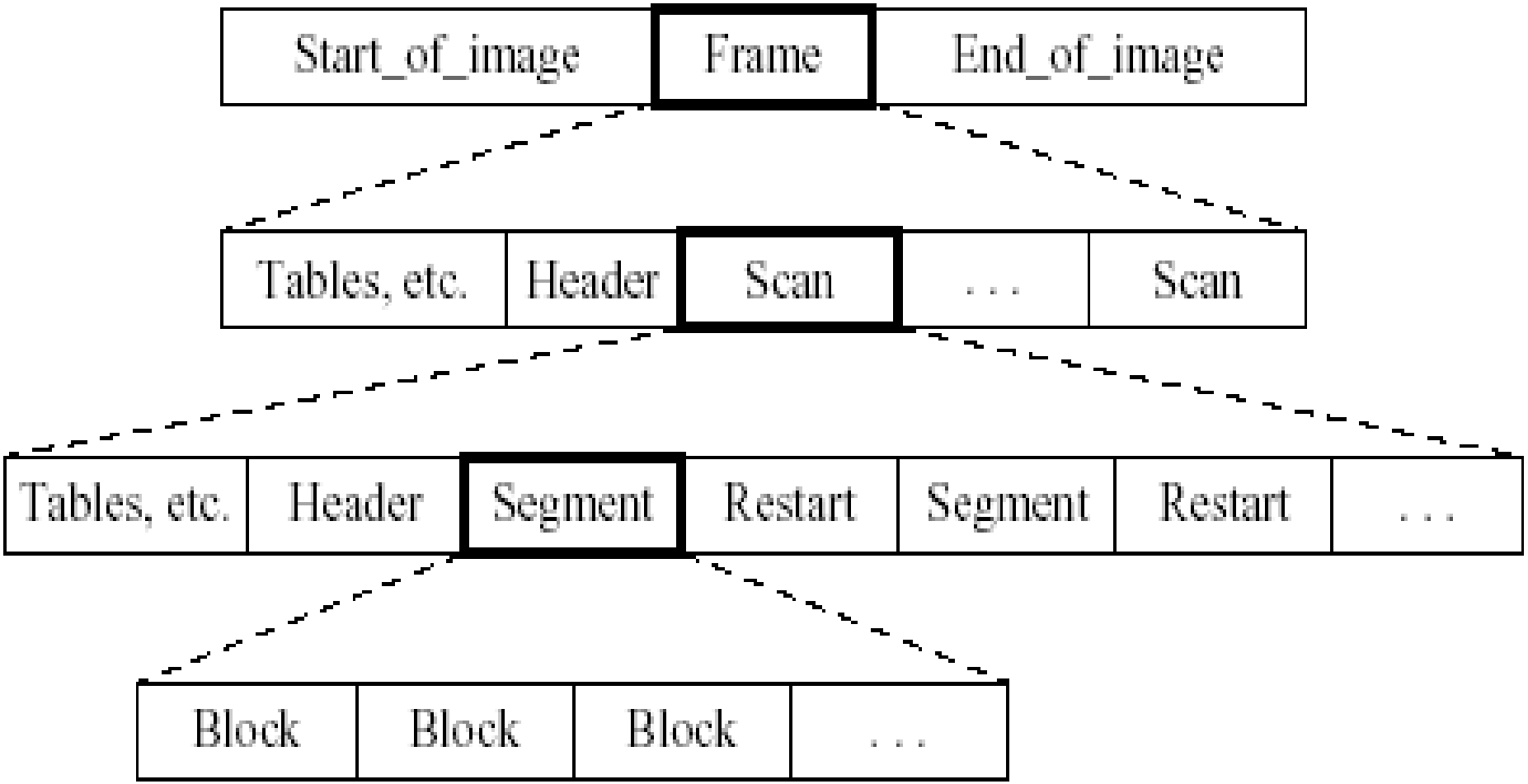


RLC of AC Coefficients

- 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*.



JPEG Bitstream

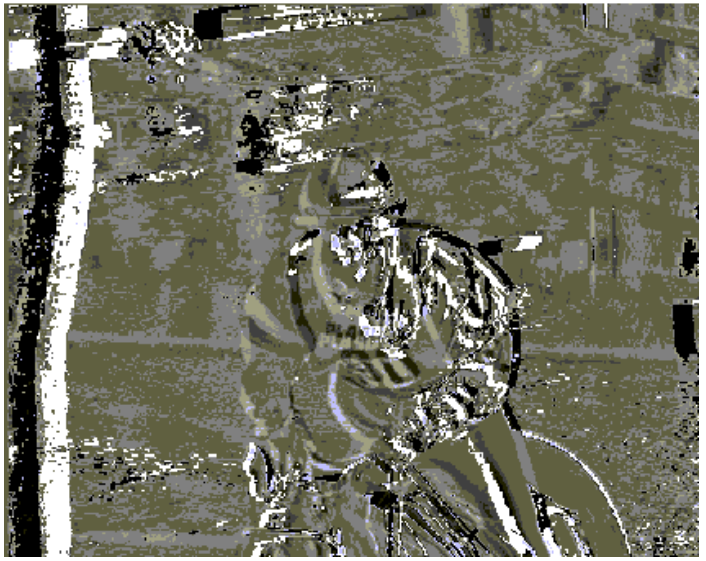


Bringing in Motion for Video

- 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.

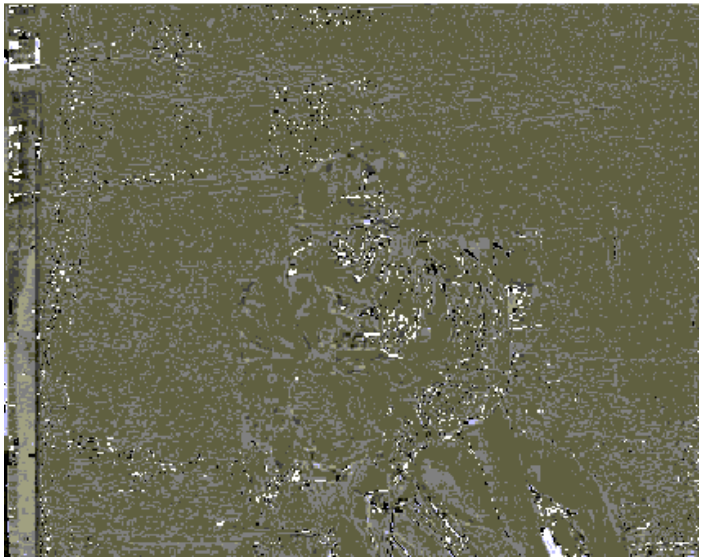
  Compression proceeds by subtracting images: subtract in time order and code the residual error.
- It can be done even better by searching for just the right parts of the image to subtract from the previous frame.

“Horse ride”



Pixel-wise difference w/o motion compensation

Motion estimation



Residue after motion compensation

Temporal Redundancy

- Consecutive frames in a video are similar — temporal redundancy exists.
- **Temporal redundancy** 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 2

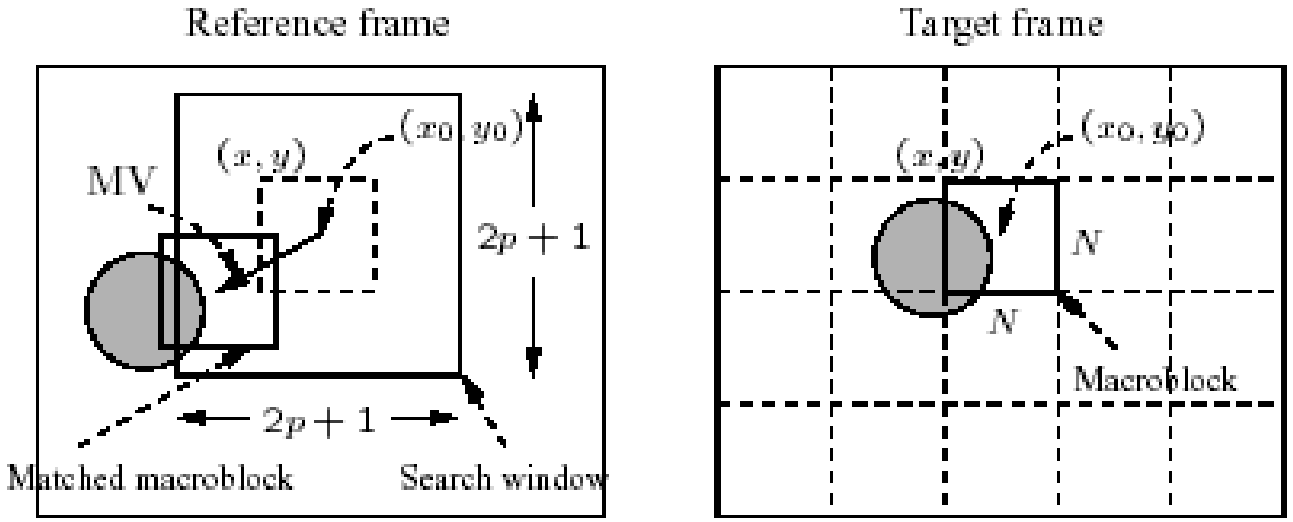


Fig. 10.1: Macroblocks and Motion Vector in Video Compression.

- 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 $(2p + 1) \times (2p + 1)$.

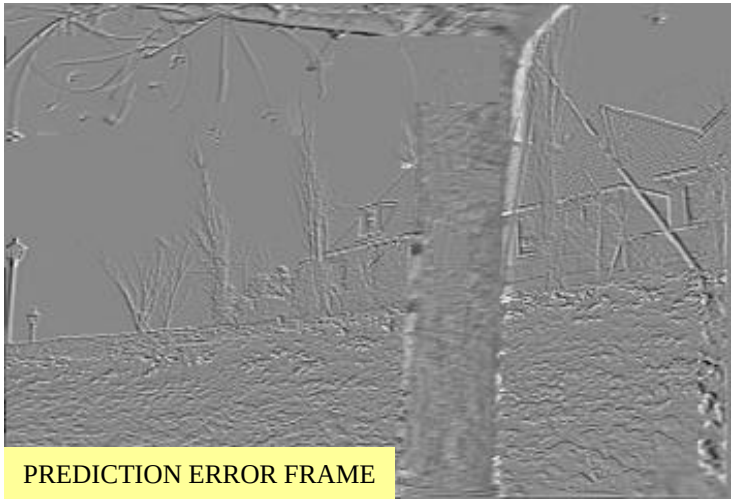
Motion Compensation

– Helps reduce temporal redundancy of video

PREVIOUS FRAME

CURRENT FRAME

PREDICTED FRAME



PREDICTION ERROR FRAME

MPEG

- **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.

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.

MPEG-1 Motion Compensation 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**.

MPEG-1 Motion Compensation 2

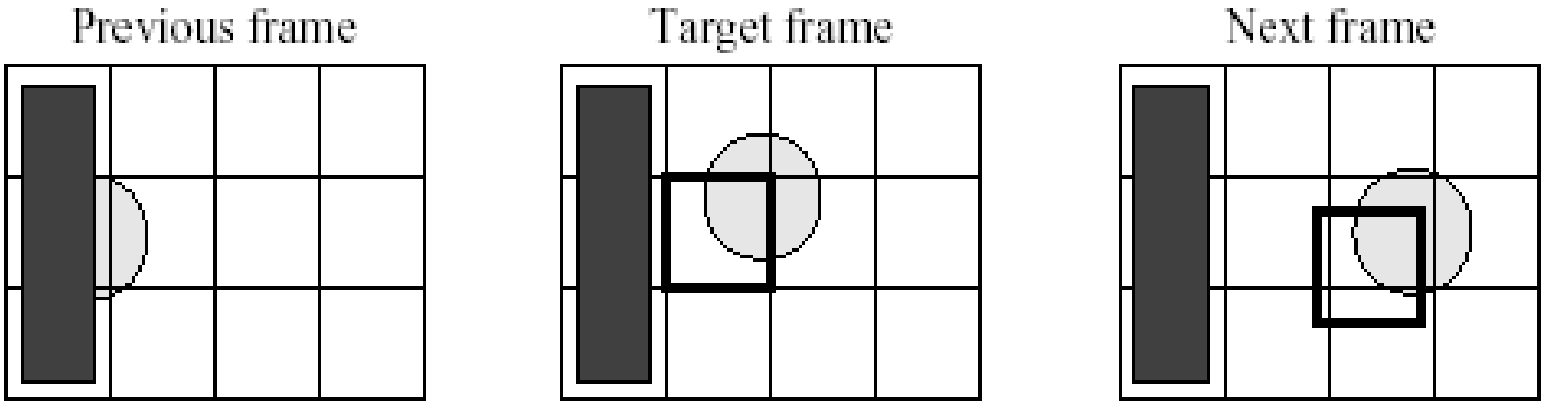


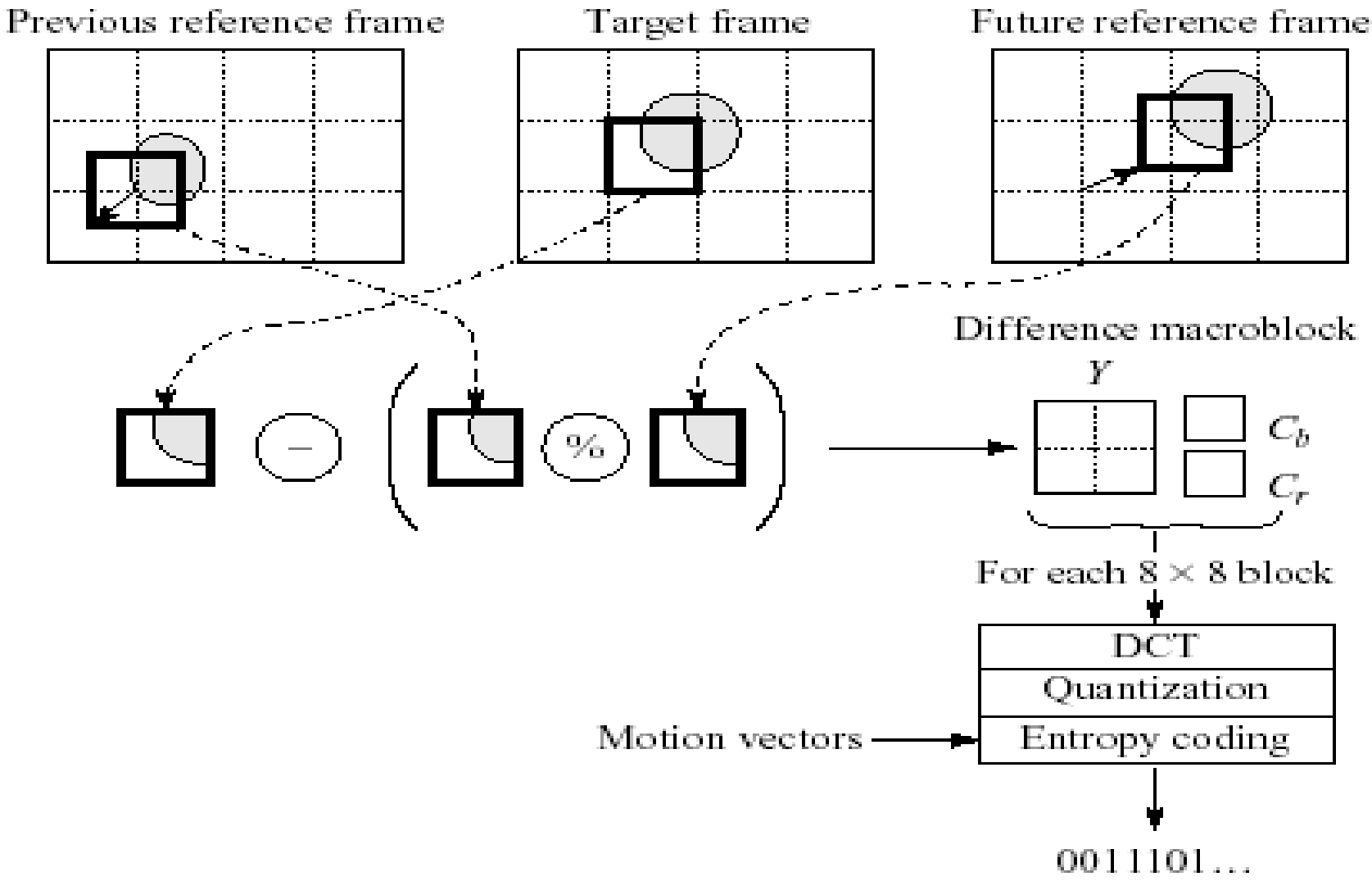
Fig 11.1: The Need for Bidirectional Search.

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.

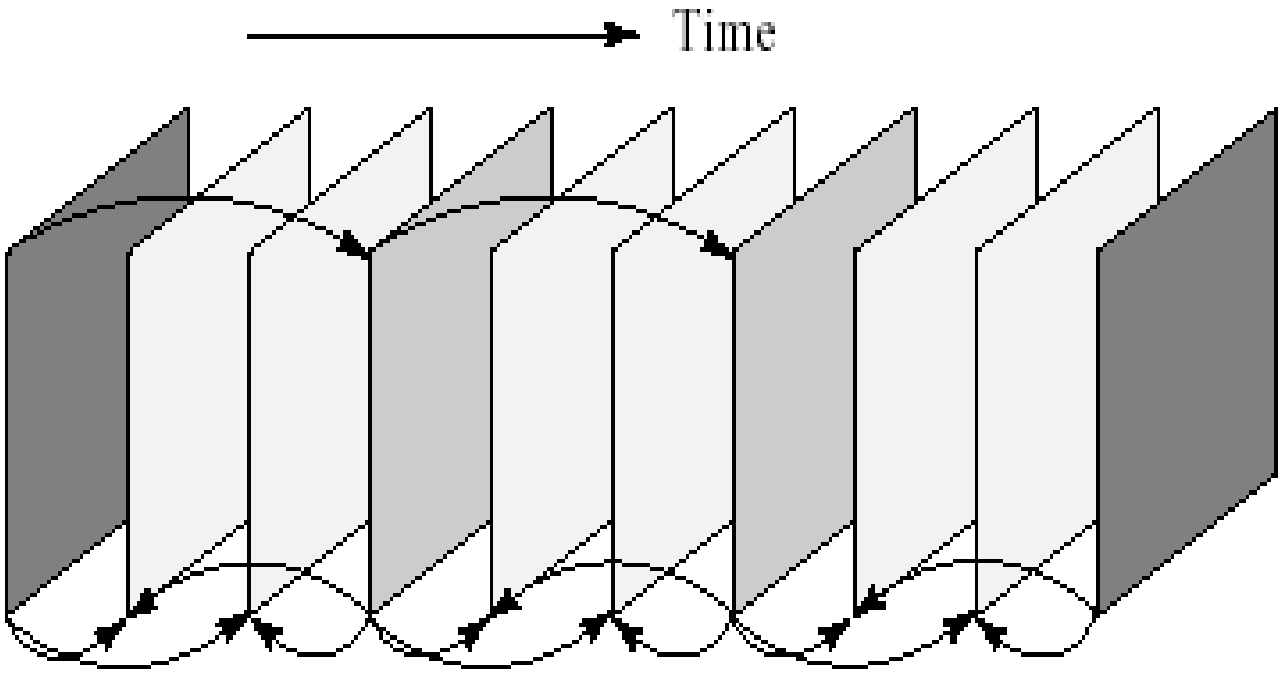
MPEG-1 Motion Compensation 3

- 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.

B Frame

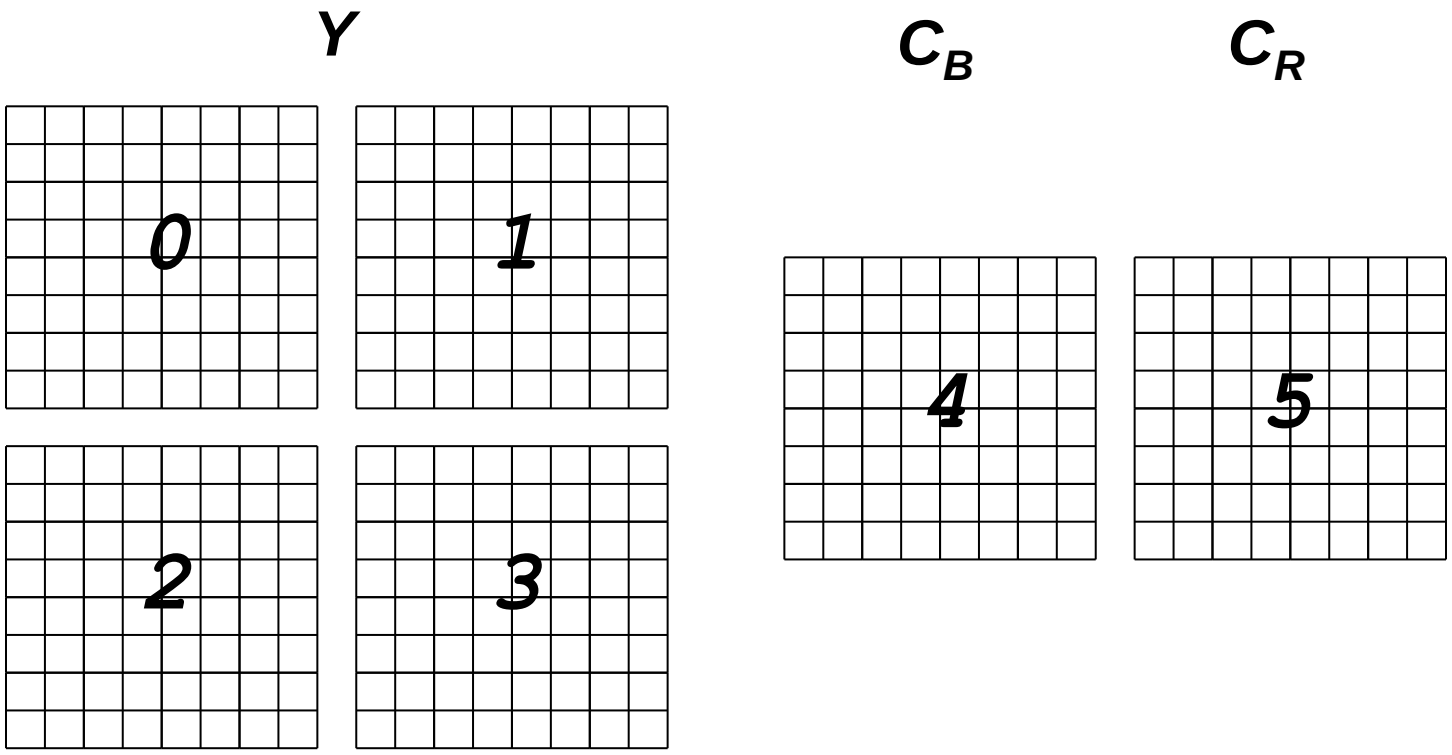

Previous reference frame
Target frame
Future reference frame
Difference macroblock
Y
C_b
C_r
For each 8 × 8 block
DCT
Quantization
Entropy coding
Motion vectors
0011101...

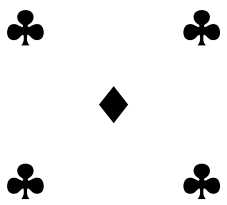
MPEG-1 Frame Sequence



Display order	I	B	B	P	B	B	P	B	B	I
Coding and transmission order	I	P	B	B	P	B	B	I	B	B

Coding of Macroblocks



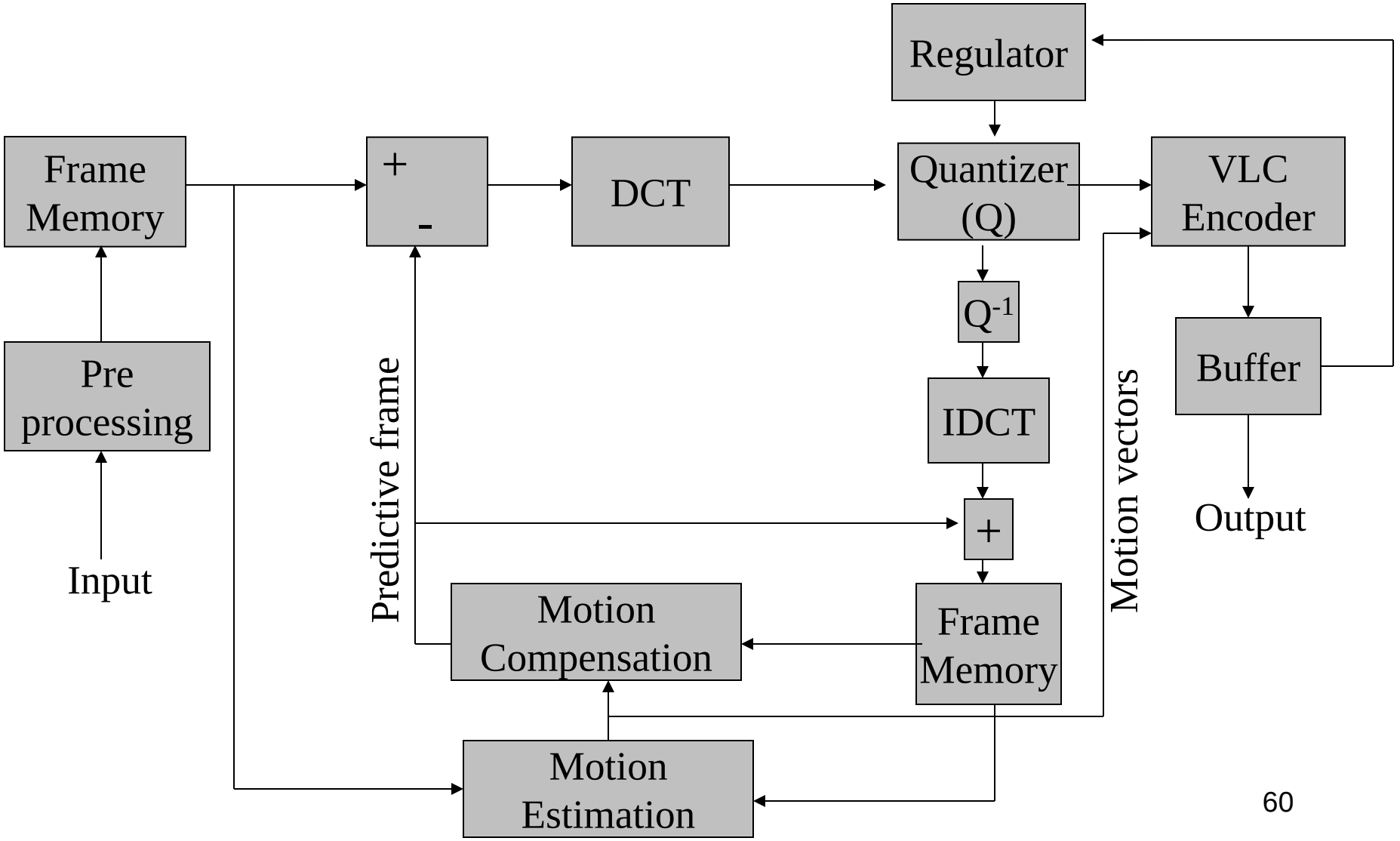


Spatial sampling relationship for MPEG-1

♣ -- Luminance sample

♦ -- Color difference sample

A Simplified MPEG Encoder



Size of MPEG-1 Frames

- The typical size of compressed P-frames is significantly smaller than that of I-frames — because temporal redundancy is exploited in inter-frame compression.

- B-frames are even smaller than P-frames — because of (a) the advantage of bi-directional prediction and (b) the lowest priority given to B-frames.

Table 11.4: Typical Compression Performance of MPEG-1 Frames

Type	Size	Compression
I	18 kB	7:1
P	6 kB	20:1
B	2.5 kB	50:1
Avg	4.8 kB	27:1

Summary

- Course Logistics
- Intro to Multimedia Computing
- JPEG
- MPEG

Next week: start of 3-part series on multimedia search, starting with image retrieval