Columbia University in the City of New York

EE4830 Digital Image Processing
Lecture 12

Image/Video Compression

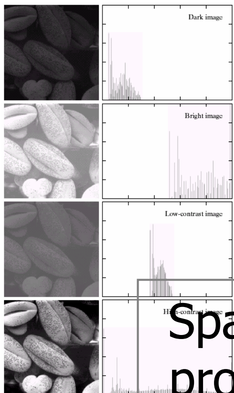
April 23, 2007

Lexing Xie
xlx at ee.columbia.edu

Announcements

- PS#6 extended to Wednesday 10am
  - libSVM compiler issues
  - SVMLight have ready packages for R14SP3 (v7.1)
    - e.g. http://webspace.ship.edu/thbrig/mexsvm/download.html
- PS#7 to be assigned by Wednesday 10am
  - analytical + mini practical
- Final Exam on May 7th 7pm~10pm
  - Similar Spec. to Midterm
  - 5 problems
  - Open book, notes, calculator
  - Coverage: Lectures 1-13

Roadmap to Date



Spatial Domain processing and enhancement

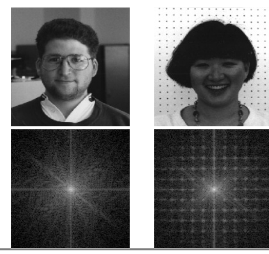


Image Transform and Filtering

Morphological Processing

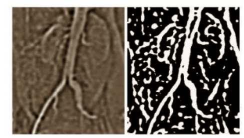


Image Segmentation

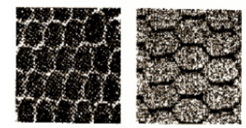


Image Descriptors

Applications:
Object recognition, Image/Video Compression
Indexing and Retrieval, Reconstruction

Lecture Outline

- Image/Video compression: What and why
- Source coding
  - Basic idea
  - Entropy coding for i.i.d. symbols
  - Coding symbol sequences
- Source coding systems
  - Compression standards
    - JPEG / MPEG / …
- Recent developments and summary

The Need for Compression

- Image: 6.0 million pixel camera, 3000x2000
  - 18 MB per image → 56 pictures / 1GB

- Video: DVD Disc 4.7 GB
  - video 720x480, RGB, 30 f/s → 31.1MB/sec
  - audio 16bits x 44.1KHz stereo → 176.4KB/s
    - → 1.5 min per DVD disc

- Send video from cellphone:
  352*240, RGB, 15 frames / second
  - 3.8 MB/sec → $38.00/sec levied by Cingular

Data Compression

- Wikipedia: "data compression, or source coding, is the process of encoding information using fewer bits (or other information-bearing units) than an unencoded representation would use through use of specific encoding schemes."

- Applications
  - General data compression: .zip, .gz …

  - Image over network: telephone/internet/wireless/etc
  - Slow device:
    - 1xCD-ROM 150KB/s, bluetooth v1.2 up to ~0.25MB/s
  - Large multimedia databases

Why Can We Compress?

- Two main reasons
  - Remove redundancy (Lossless): preserve all information, perfectly recoverable.
  - Reduce irrelevance (Lossy): cannot recover all bits.

- Three types of operations
  - Symbol redundancy: give common values shorts codes and uncommon values longer codes.
  - Inter-pixel redundancy: adjacent pixels are highly correlated.
  - Perceptual redundancy: not all information is perceived by eye/brain, so throw away those that are not.

Psychovisual Redundancy

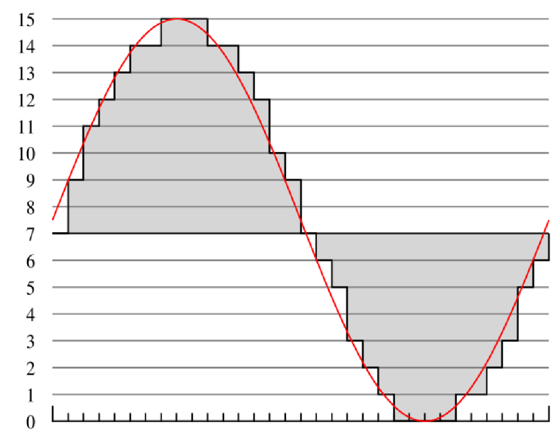
a b c

FIGURE 8.4
(a) Original image.
(b) Uniform quantization to 16 levels. (c) IGS quantization to 16 levels.

PCM

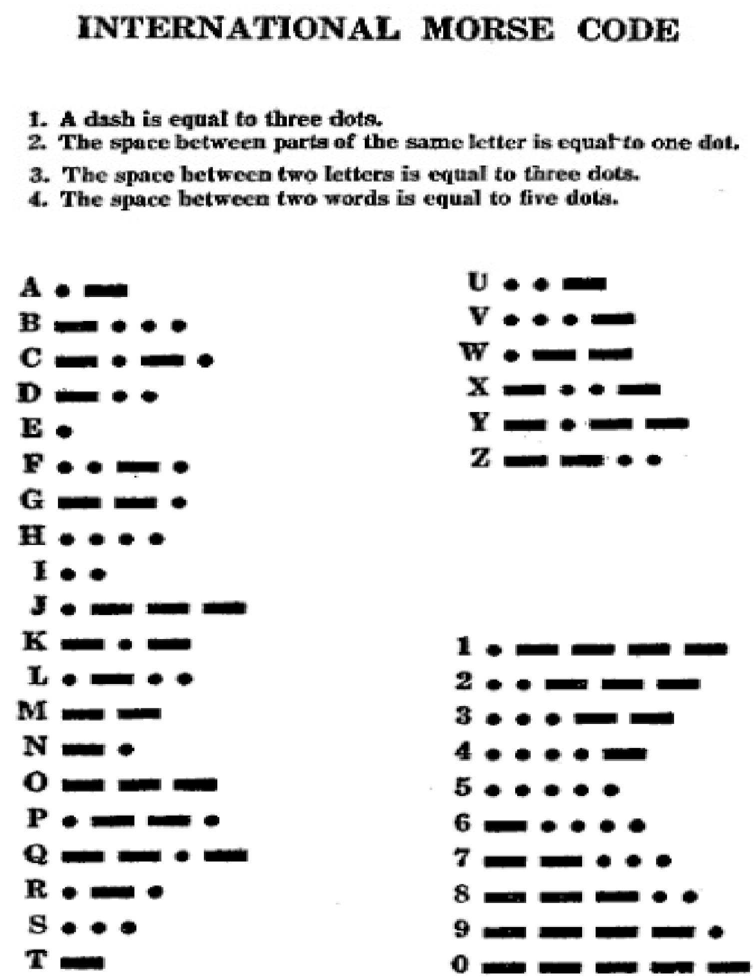
Pulse-code modulation (PCM) is a digital representation of an analog signal where the magnitude of the signal is sampled regularly at uniform intervals.

http://en.wikipedia.org/wiki/Pulse-code_modulation



Symbol/inter-symbol Redundancy

- Letters and words in English
  - e, a, i, s, t, …
    q, y, z, x, j, …
  - a, the, me, I …
    good, magnificent, …
  - fyi, btw, ttyl …
- In the evolution of language we naturally chose to represent frequent meanings with shorter representations.



INTERNATIONAL MORSE CODE

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to five dots.

A · —	U · · —
B — · · ·	V · · · —
C — · — ·	W · — —
D — · ·	X — · · —
E ·	Y — · — —
F · · — ·	Z — — · ·
G — — ·	1 · — — — —
H · · · ·	2 · · — — —
I · ·	3 · · · — —
J · — — —	4 · · · · —
K — · —	5 · · · · ·
L · — · ·	6 — · · · ·
M — —	7 — — · · ·
N — ·	8 — — — · ·
O — — —	9 — — — — ·
P · — — ·	0 — — — — —
Q — — · —	
R · — ·	
S · · ·	
T —	

Pixel/inter-pixel Redundancy

- Some gray level value are more probable than others.
- Pixel values are not i.i.d. (independent and identically distributed)

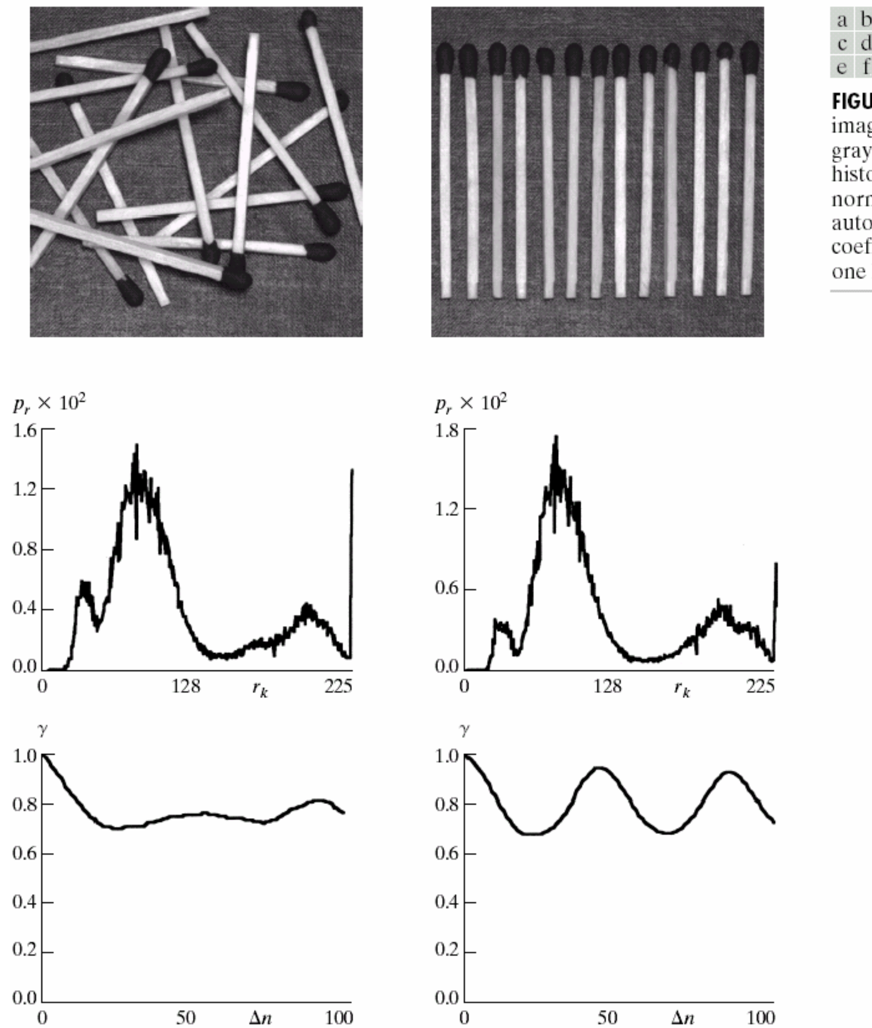


FIGURE 8.2 Two images and their gray-level histograms and normalized autocorrelation coefficients along one line.

How much can we compress a picture?

IMG_0470.jpg	944 KB	JPEG Image	11/11/2006 4:29 PM	11/11/2006 3:29 PM	1600 x 1200
IMG_0471.jpg	892 KB	JPEG Image	11/11/2006 4:30 PM	11/11/2006 3:30 PM	1600 x 1200
IMG_0472.jpg	876 KB	JPEG Image	11/11/2006 4:31 PM	11/11/2006 3:31 PM	1600 x 1200
IMG_0473.jpg	1,214 KB	JPEG Image	11/11/2006 4:38 PM	11/11/2006 3:38 PM	1600 x 1200
IMG_0474.jpg	1,117 KB	JPEG Image	11/11/2006 4:38 PM	11/11/2006 3:38 PM	1600 x 1200
IMG_0475.jpg	1,208 KB	JPEG Image	11/11/2006 4:38 PM	11/11/2006 3:38 PM	1600 x 1200
IMG_0476.jpg	795 KB	JPEG Image	11/11/2006 4:39 PM	11/11/2006 3:39 PM	1600 x 1200
IMG_0477.jpg	1,042 KB	JPEG Image	11/11/2006 4:39 PM	11/11/2006 3:39 PM	1600 x 1200
IMG_0478.jpg	1,027 KB	JPEG Image	11/11/2006 4:40 PM	11/11/2006 3:40 PM	1600 x 1200
IMG_0479.jpg	1,010 KB	JPEG Image	11/11/2006 4:40 PM	11/11/2006 3:40 PM	1600 x 1200
IMG_0480.jpg	790 KB	JPEG Image	11/11/2006 4:41 PM	11/11/2006 3:41 PM	1600 x 1200
IMG_0481.jpg	959 KB	JPEG Image	11/11/2006 4:41 PM	11/11/2006 3:41 PM	1600 x 1200
IMG_0482.jpg	1,073 KB	JPEG Image	11/11/2006 4:42 PM	11/11/2006 3:42 PM	1600 x 1200
IMG_0483.jpg	990 KB	JPEG Image	11/11/2006 4:43 PM	11/11/2006 3:43 PM	1600 x 1200
IMG_0484.jpg	1,046 KB	JPEG Image	11/11/2006 4:45 PM	11/11/2006 3:45 PM	1600 x 1200
IMG_0485.jpg	878 KB	JPEG Image	11/11/2006 4:46 PM	11/11/2006 3:46 PM	1600 x 1200
IMG_0486.jpg	774 KB	JPEG Image	11/11/2006 4:46 PM	11/11/2006 3:46 PM	1600 x 1200
IMG_0487.jpg	830 KB	JPEG Image	11/11/2006 4:47 PM	11/11/2006 3:47 PM	1600 x 1200
IMG_0488.jpg	1,011 KB	JPEG Image	11/11/2006 4:47 PM	11/11/2006 3:47 PM	1600 x 1200
IMG_0489.jpg	957 KB	JPEG Image	11/11/2006 4:47 PM	11/11/2006 3:47 PM	1600 x 1200
IMG_0490.jpg	961 KB	JPEG Image	11/11/2006 4:48 PM	11/11/2006 3:48 PM	1600 x 1200

Fundamentals of Source Coding

- i.i.d random variable $X \sim p(x), \; x \in \mathcal{X}$
- Entropy $H(X) = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)}$

- Source code $C(x) : \mathcal{X} \rightarrow \mathcal{C}$
- Length of the codeword $l(x), \; x \in \mathcal{X}$
- Expected length of C $L(C) = \sum_{x \in \mathcal{X}} p(x) l(x)$

- An example

$$\mathcal{X} = \{a, b, c, d\}$$

$$\begin{aligned} P(X=a) &= 1/2 \\ P(X=b) &= 1/4 \\ P(X=c) &= 1/8 \\ P(X=d) &= 1/8 \end{aligned} \qquad \begin{aligned} C(a) &= 1 \\ C(b) &= 10 \\ C(c) &= 110 \\ C(d) &= 111 \end{aligned}$$

$H(X) = ?$, $L(C) = ?$

What Makes a Good Code

- Source coding theorem (Shannon 1948)

  $X_1, X_2, ..., X_n$ i.i.d (stationary process)

  $L_n \rightarrow H(\mathcal{X})$

- Desired properties of good codes:
  - Non-singular: every symbol in X maps to a different code word
  - Uniquely decodable: every sequence $\{x_1, \ldots x_n\}$ maps to different codeword sequence
  - Instantaneous: no codeword is a prefix of any other codeword

Huffman Codes

- Revisit example

$$\mathcal{X} = \{a, b, c, d\}$$

$$\begin{aligned} P(X = a) &= 1/2 \\ P(X = b) &= 1/4 \\ P(X = c) &= 1/8 \\ P(X = d) &= 1/8 \end{aligned} \qquad \begin{aligned} C(a) &= 1 \\ C(b) &= 10 \\ C(c) &= 110 \\ C(d) &= 111 \end{aligned}$$

$$H(X) = 1.75 \ , \quad L(C) = 1.75$$

- Is this code: non-singular /uniquely decodable / instantaneous?
- If not, how to improve it?

Arithmetic Coding

- Huffman code is optimal but must be integer length

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)

TABLE 8.6 Arithmetic coding example.

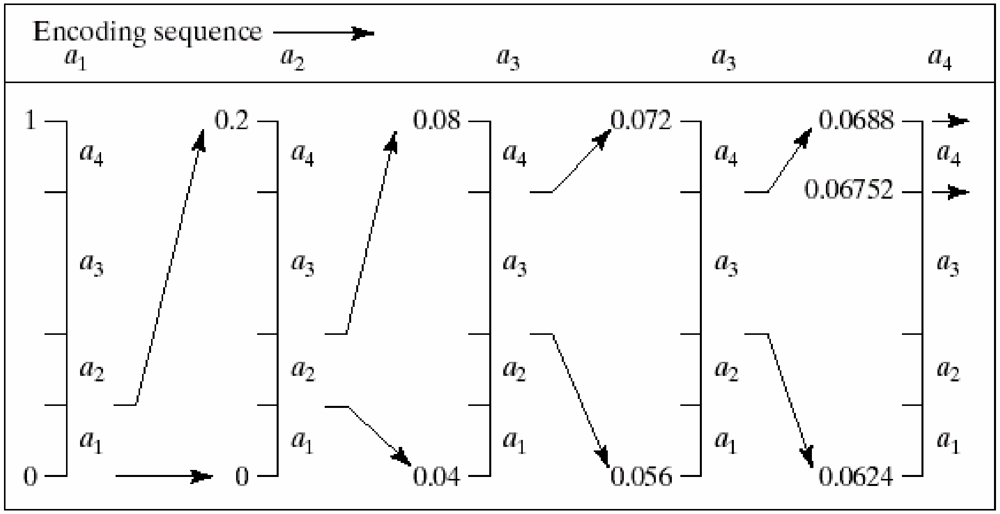


FIGURE 8.13 Arithmetic coding procedure.

Universal Data Compression

- What if the symbol probabilities are unknown?
- LZW algorithm (Lempel-Ziv-Welch)

encoding

```
w = NIL;
    while ( read a character k )
         {
           if wk exists in the dictionary
            w = wk;
           else
             add wk to the dictionary;
             output the code for w;
             w = k;
         }
```

decoding

```
read a character k;
   output k;
   w = k;
   while ( read a character k )
  /* k could be a character or a code. */
        {
          entry = dictionary entry for k;
          output entry;
          add w + entry[0] to dictionary;
          w = entry;
        }
```

- Widely used: GIF, TIFF, PDF …
- Its royalty-free variant (DEFLATE) used in PNG, ZIP, …
  - Unisys U.S. LZW Patent No. 4,558,302 expired on June 20, 2003 http://www.unisys.com/about__unisys/lzw

LZW

Example

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

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		

TABLE 8.7
LZW coding example.

- Exercise: verify that the dictionary can be automatically reconstructed during decoding. (G&W Problem 8.16)

Lecture Outline

- Image/Video compression: What and why
- Source coding
  - Basic idea
  - Entropy coding for i.i.d symbols
  - Coding symbol sequences
- Source coding systems
  - Compression standards
    - JPEG / MPEG / …
- Current developments and future directions

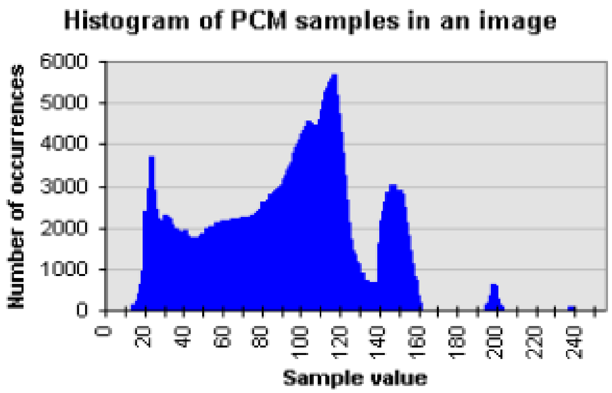
Run-Length Coding

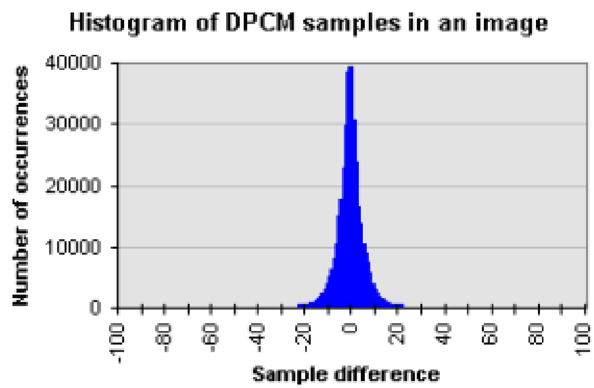
- Encode the number of consecutive '0's or '1's
- Used in FAX transmission standard

- Why is run-length coding with $P(X=0) >> P(X=1)$ actually beneficial?
  - See Jain Sec 11.3 (at course works)

Predictive Coding

- Signals are correlated → predict and encoding the difference lowers the bitrate
- Good prediction is the key: e.g. LPC (linear-predctive) speech coding





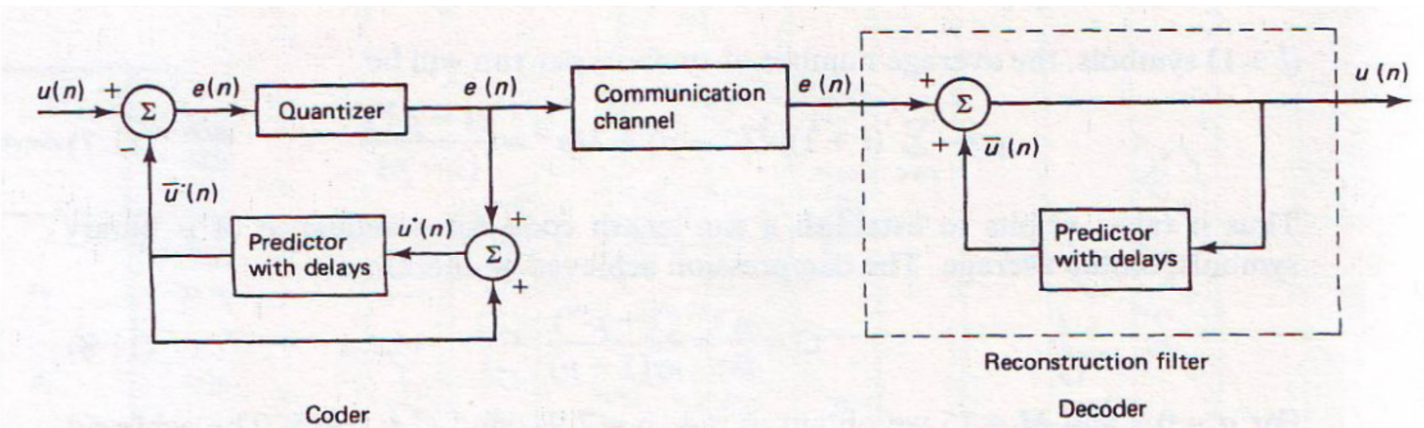


Figure 11 5 Differential pulse code modulation (DPCM) CODEC

G&W Sec. 8.5.1, Jain Sec. 11.3

Transform Coding

- Review: properties of unitary transform
  - De-correlation: highly correlated input elements → quite uncorrelated output coefficients
  - Energy compaction: many common transforms tend to pack a large fraction of signal energy into just a few transform coefficients

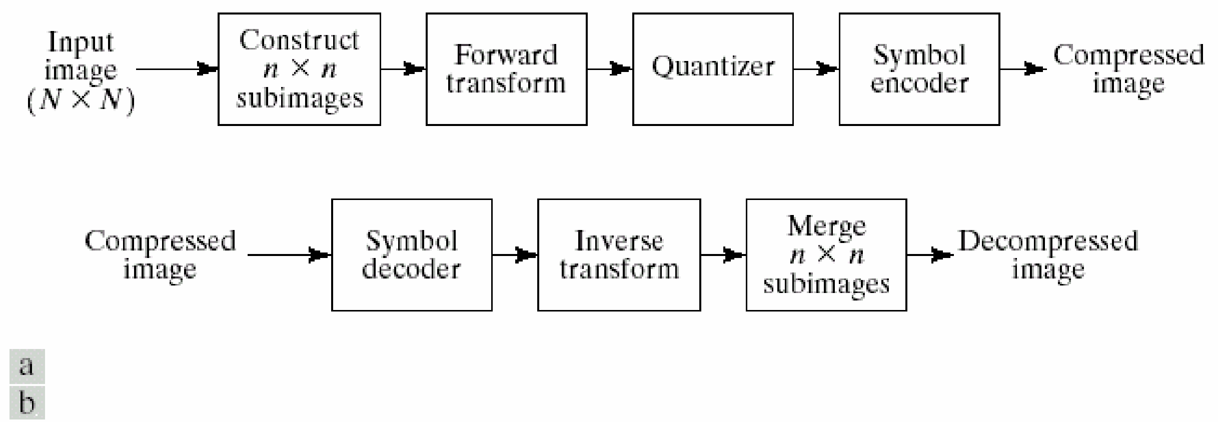


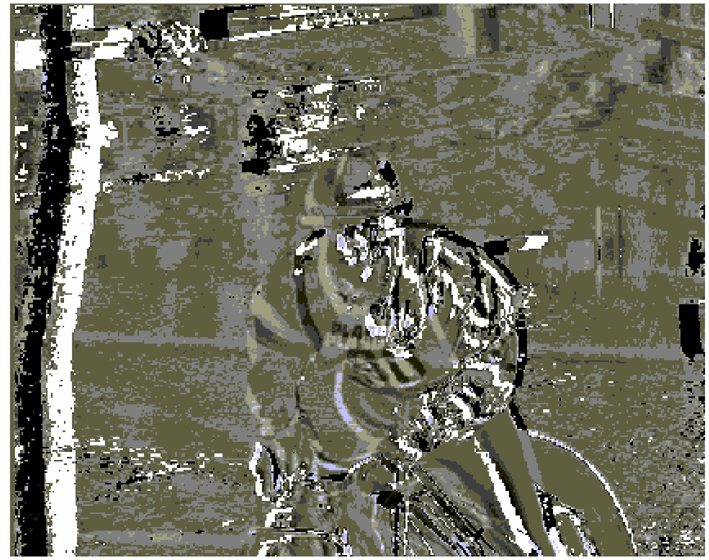
FIGURE 8.28 A transform coding system: (a) encoder; (b) decoder.

Video ?= Motion Pictures

- Capturing video
  - Frame by frame => image sequence
  - Image sequence: A 3-D signal
    - 2 spatial dimensions & time dimension
    - continuous $I(x, y, t)$ => discrete $I(m, n, t_k)$
- Encode digital video
  - Simplest way ~ compress each frame image individually
    - e.g., "motion-JPEG"
    - only spatial redundancy is explored and reduced
  - How about temporal redundancy? Is differential coding good?
    - Pixel-by-pixel difference could still be large due to motion

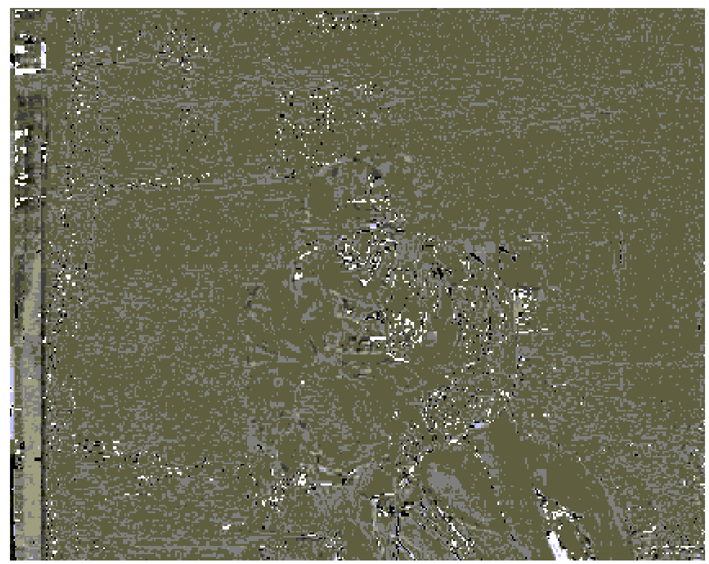
➔ Need better prediction

"Horse ride"



Pixel-wise difference w/o motion compensation

Motion estimation



Residue after motion compensation

Lecture Outline

- Image/Video compression: What and why
- Source coding
  - Basic idea
  - Entropy coding for i.i.d symbols
  - Coding symbol/pixel/image sequences
- Source coding systems
  - Quality measures
  - Image compression system and algorithms: JPEG
  - Video compression system and algorithms: MPEG
- Current developments and future directions

Image Quality Measures

- Quality measures
  - PSNR (Peak-Signal-to-Noise-Ratio)

$$PSNR = 10\log_{10}\left[\frac{255^2}{\frac{1}{MN}\sum_{xy}|f'(x,y)-f(x,y)|^2}\right]$$

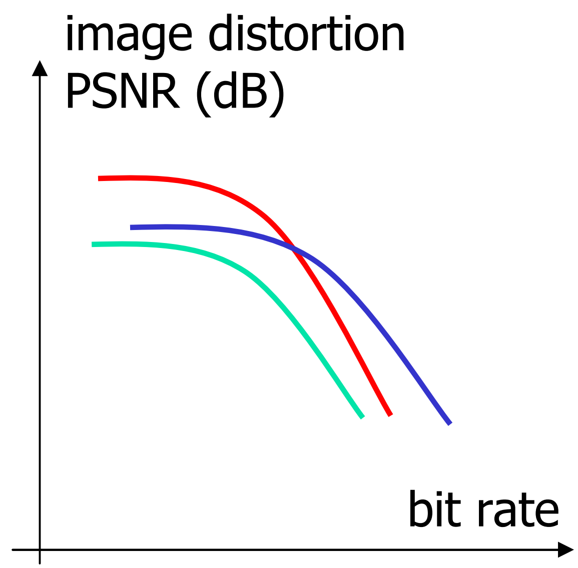
    - Why would we prefer PSNR over SNR?
  - Visual quality
    - Compression Artifacts
    - Subjective rating scale

TABLE 8.3 Rating scale of the Television Allocations Study Organization. (Frendendall and Behrend.)

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

Measures for Coding System

- End-to-end measures of source coding system: Rate-Distortion
- Other considerations
  - Computational complexity
  - Power consumption
  - Memory requirement
  - Delay
  - Error resilience/sensitivity
  - Subjective quality



bpp: bit-per-pixel;

Kbps: Kilo-bits-per-second

Image/Video Compression Standards

- Bitstream useful only if the recipient knows the code!
- Standardization efforts are important
  - Technology and algorithm benchmark
  - System definition and development
  - Patent pool management
- Defines the bitstream (decoder), not how you generate them (encoder)!

v • d • e **Multimedia compression formats** [hide]

Video compression formats	**ISO/IEC**	MPEG-1 · MPEG-2 · MPEG-4 ASP · MPEG-4/AVC
	ITU-T	H.261 · H.262 · H.263 · H.264
	Others	AVS · Bink · Dirac · Indeo · MJPEG · RealVideo · Theora · VC-1 · VP6 · VP7 · WMV
Audio compression formats	**ISO/IEC MPEG**	MPEG-1 Layer III (MP3) · MPEG-1 Layer II · AAC · HE-AAC
	ITU-T	G.711 · G.722 · G.722.1 · G.722.2 · G.723 · G.723.1 · G.726 · G.728 · G.729 · G.729.1 · G.729a
	Others	AC3 · Apple Lossless · ATRAC · FLAC · iLBC · Monkey's Audio · μ-law · Musepack · Nellymoser · RealAudio · SHN · Speex · Vorbis · WavPack · WMA
Image compression formats	**ISO/IEC/ITU-T**	JPEG · JPEG 2000 · lossless JPEG · JBIG · JBIG2 · PNG · WBMP
	Others	APNG · ICER · MNG · BMP · GIF · ILBM · PCX · TGA · TIFF · HD Photo
Media container formats	**General**	3GP · ASF · AVI · DMF · DPX · FLV · Matroska · MP4 · MXF · NUT · Ogg · Ogg Media · QuickTime · RealMedia · VOB
	Audio only	AIFF · AU · WAV

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK
RECORD

VOL. 23, NO. 4 September 26, 1997

Digital TV Patent License Fees to Go to Columbia Very Soon

By Bob Nelson

Digital television is on its way and Columbia, the only academic institution in the patent pool created to license the MPEG-2 digital video compression standard, expects to begin receiving license fees from the technology as early as this year.

Columbia and eight companies together hold 33 patents that now comprise MPEG-2, which allows the transmission of high-quality video and audio signals over limited bandwidth. Dimitris Anastassiou, professor of electrical engineering at Columbia's School of Engineering and Applied Science and director of the Columbia New Media Technology Center, developed one of the MPEG-2 compression technologies with one of his graduate students.

"We believe the patent pool approach offers Columbia an excellent opportunity to receive significant royalty payments over the next few years," said Jack Granowitz, executive director of the Columbia Innovation Enterprise (CIE), the University's technology licensing office. Granowitz, along with

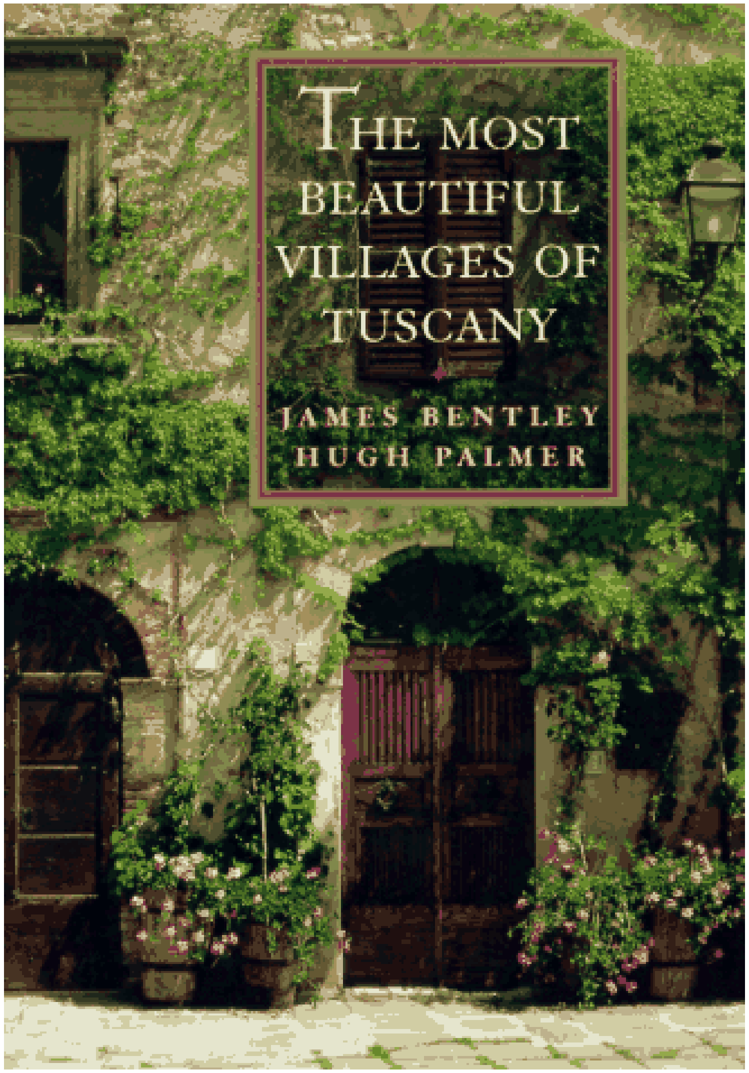
Audio coding vs. Image coding

	MP3 (wideband audio coding)	JPEG
Data Unit	Frame	Block
Transform	MDCT	DCT
Quantization	Fixed Quantization matrix base on psychoacoustic masking	Baseline quantization matrix + adaptive rate control
Entropy coding	Huffman code	Huffman code, run-length, differential

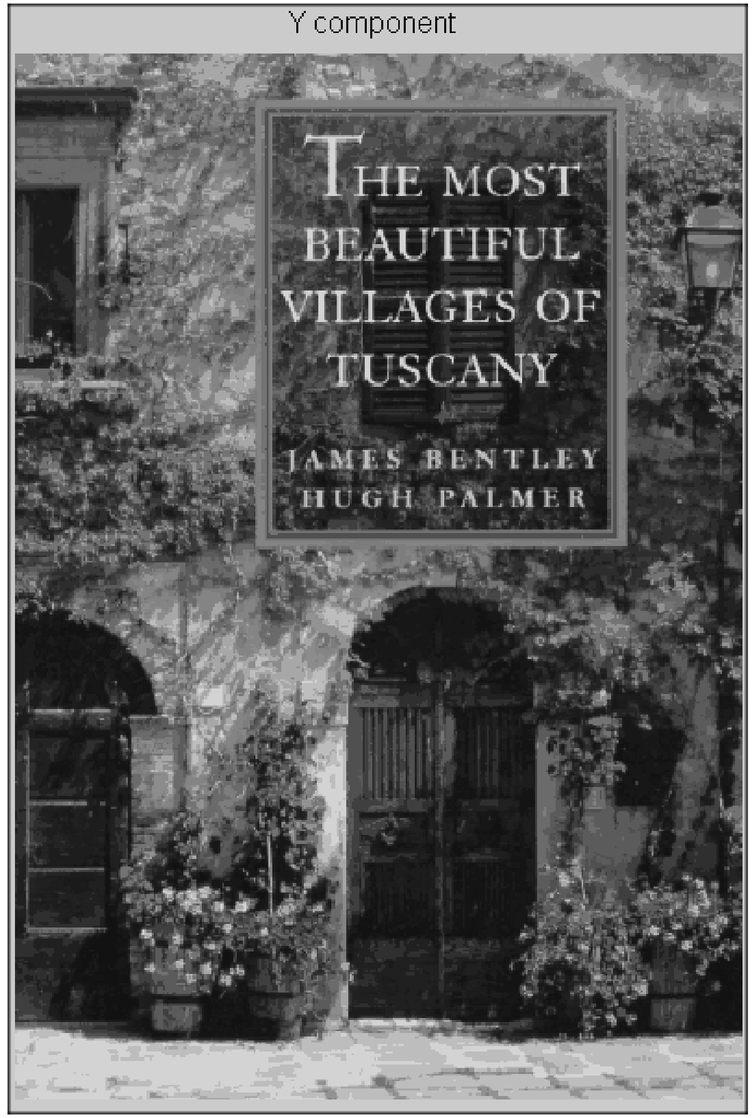
JPEG Compression Standard (early 1990s)

- JPEG - Joint Photographic Experts Group
  - Compression standard of generic continuous-tone still image
  - Became an international standard in 1992
- 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

Representation in JPEG



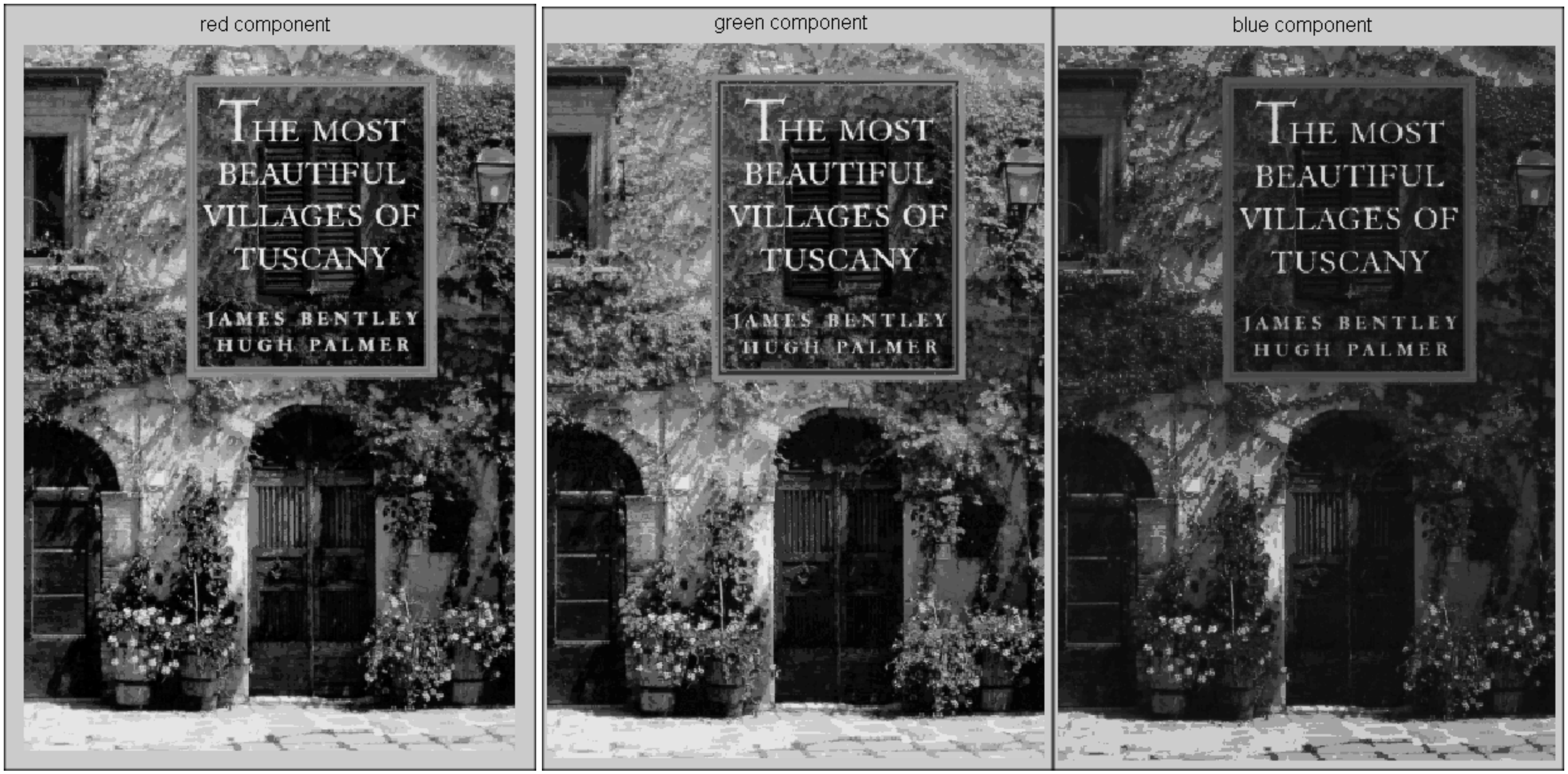
475 x 330 x 3 = 157 KB



luminance

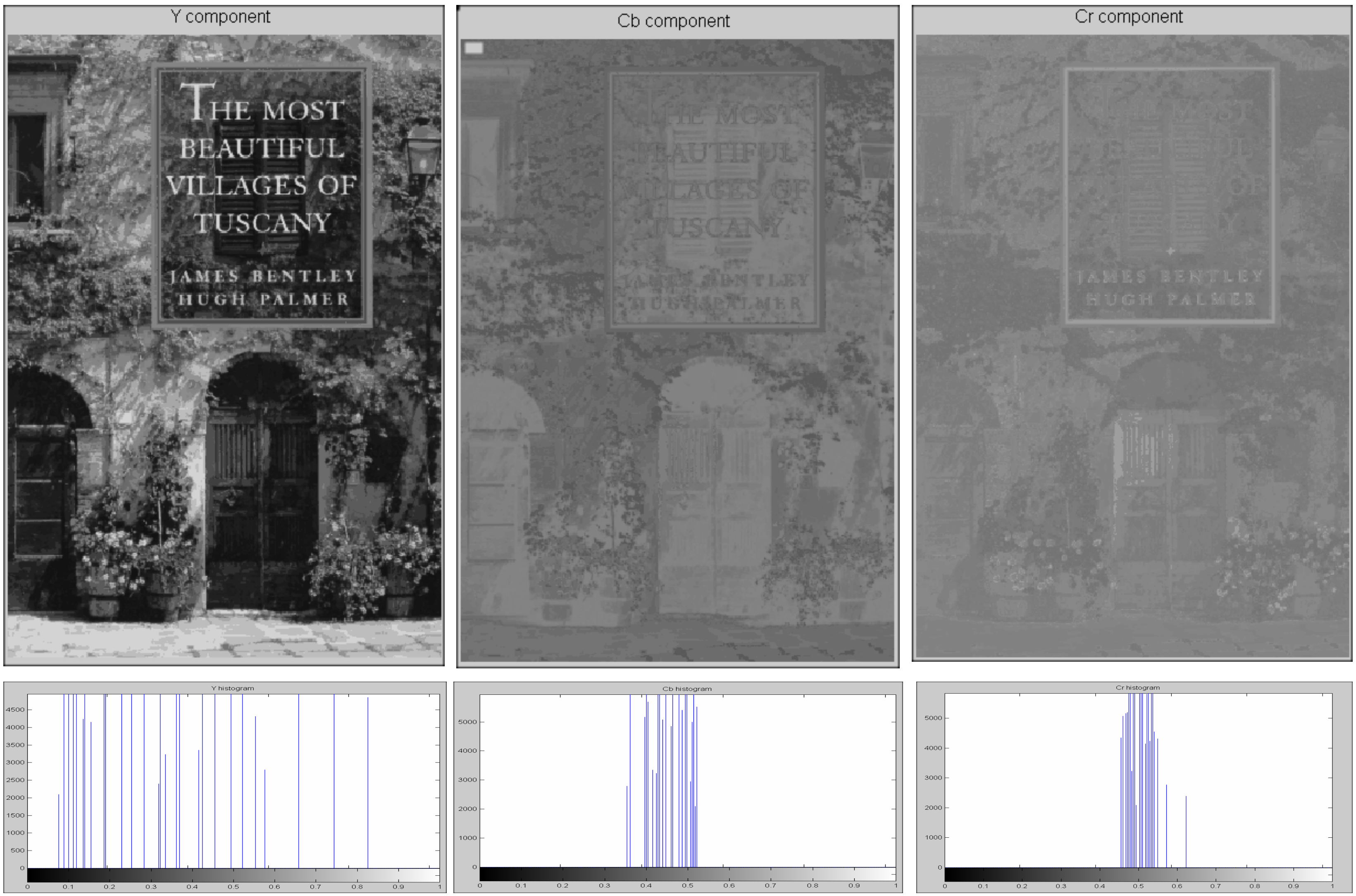
From Liu's EE330 (Princeton)

RGB Components



From Liu’s EE330 (Princeton)

Y U V (Y Cb Cr) Components



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

From Liu's EE330 (Princeton)

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

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

(Based on Wang's video book Chapt.1)

  - Interleaving color components

Block-based Transform

- Why block based?
  - High transform computation complexity for larger blocks
    - O(m *log* m × m) per block in transform for (MN/m^2) blocks
  - High complexity in bit allocation
  - Block transform captures local info
- Commonly used block sizes: 8x8, 16x16, 8x4, 4x8 …

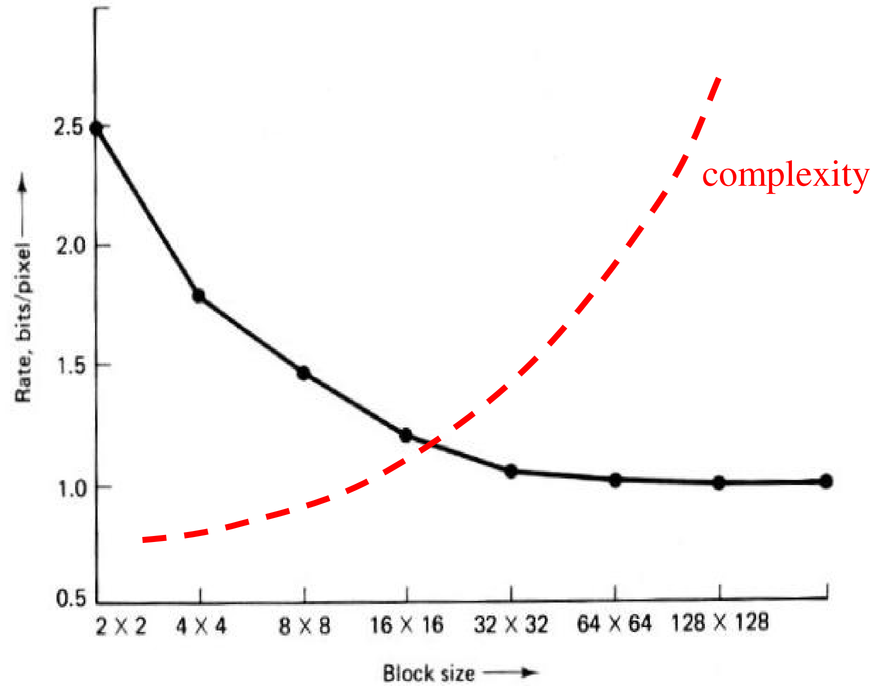
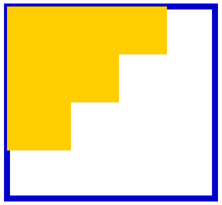


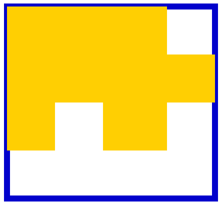
Figure 11.16 Rate achievable by block KL transform coders for Gaussian random fields with separable covariance function, $\rho = \rho_2 = 0.95$, at distortion $D = 0.25\%$.

From Jain's Fig.11.16

Zonal Coding and Threshold Coding



- Zonal coding
  - Only transmit a small predetermined zone of transformed coeff.



- Threshold coding
  - Transmit coeff. that are above certain thresholds
- Compare
  - Threshold coding is inherently adaptive
    - introduce smaller distortion for the same number of coded coeff.
  - Threshold coding needs overhead in specifying index of coded coeff.
    - run-length coding helps to reduce overhead

How Quantization is Performed

- Input:
  - 8x8 DCT image X(u,v)
  - Quantization table Q(u,v)
- The quantizer output is:

  I(u,v)=Round[X(u,v)/Q(u,v)]
  - "round" is to the nearest integer
- JPEG default luminance table shown on the right
  - Smaller Q(u,v) means a smaller step size and hence more resolution, vice-versa
  - Q(u,v) may be scaled by a quality factor

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

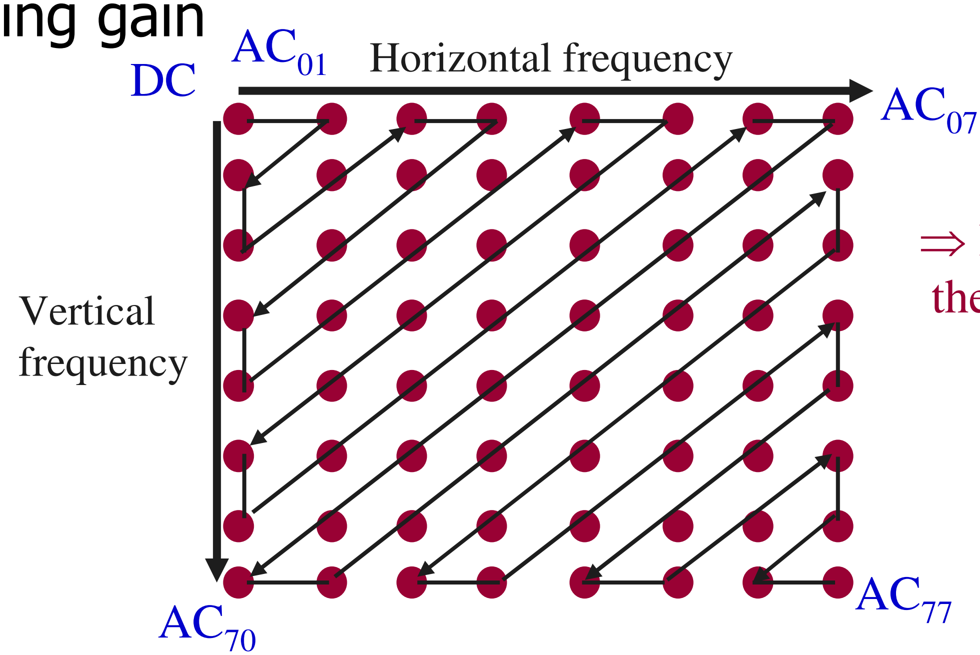
Quantization of Coefficients JPEG

- Default quantization table
  - "Generic" over a variety of images
- Adaptive Quantization (bit allocation)
  - Different quantization step size for different coeff. 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

- 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

Encoding a Block in JPEG

- Basic tools
  - Run-length coding
  - Predictive coding (esp. for DC coefficient)
  - Entropy coding (Huffman, etc.)
- Scan order
  - zig-zag scan for block-DCT to better achieve run-length coding gain



⇒ low-frequency coefficients, then high frequency coefficients

Encoding a Block in JPEG (2)

- Differentially encode DC (and quantize)
  - (SIZE, AMPLITUDE), with amplitude range in [-2048, 2047]

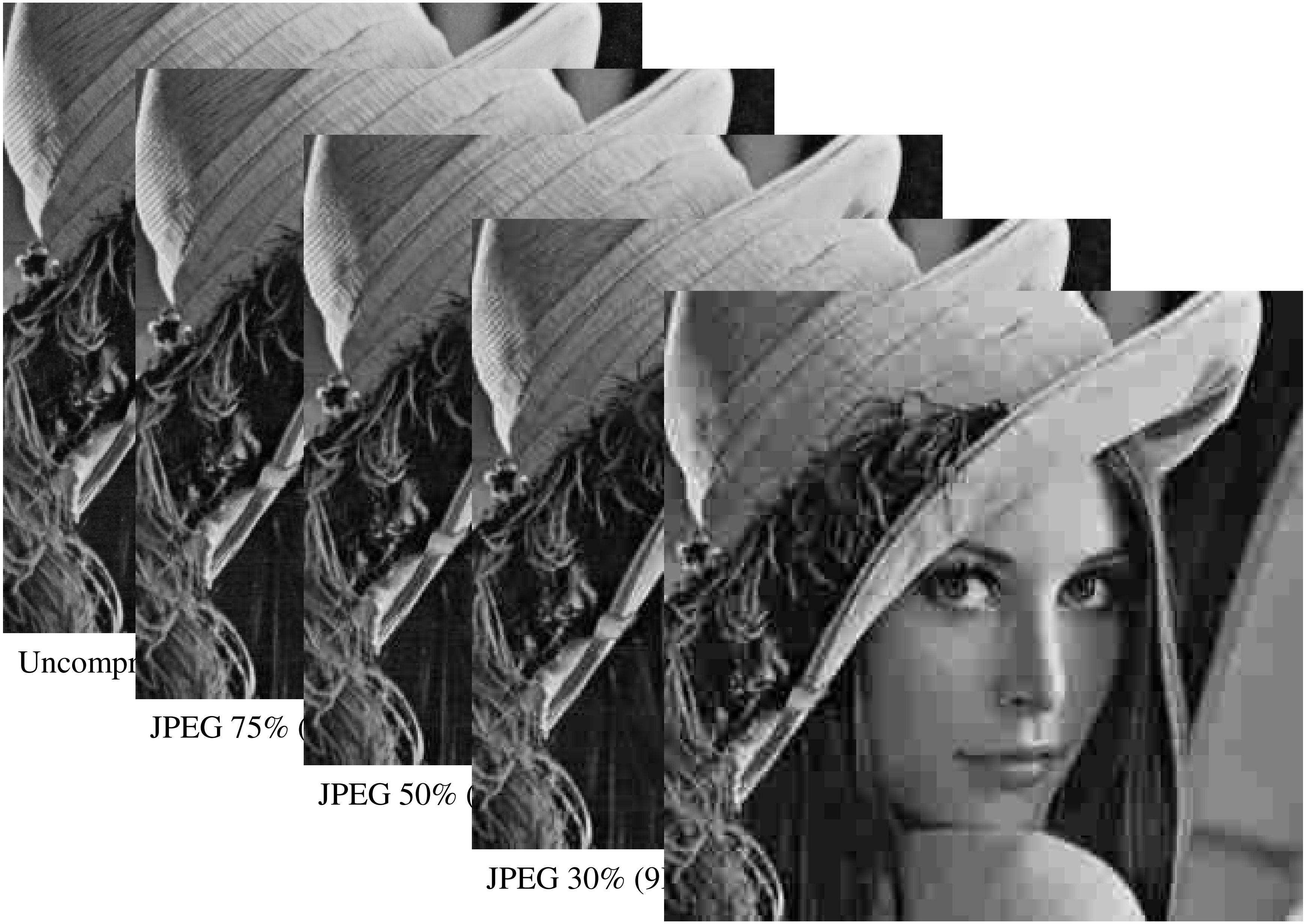
- AC coefficients in one block
  - Zig-zag scan after quantization for better run-length
    - save bits in coding consecutive zeros
  - Represent each AC run-length using entropy coding
    - use shorter codes for more likely AC run-length symbols

    - Symbol-1: (RUNLENGTH, SIZE) ➔ Huffman coded
    - Symbol-2: AMPLITUDE ➔ Variable length coded

RUNLENGTH ∈ [0,15]
# of consecutive zero-valued AC coefficients
preceding the nonzero AC coefficient ∈ [0,15]

SIZE ∈ [0 to 10 in unit of bits]
# of bits used to encode AMPLITUDE

AMPLITUDE ∈ in range of [-1023, 1024]



Uncompr

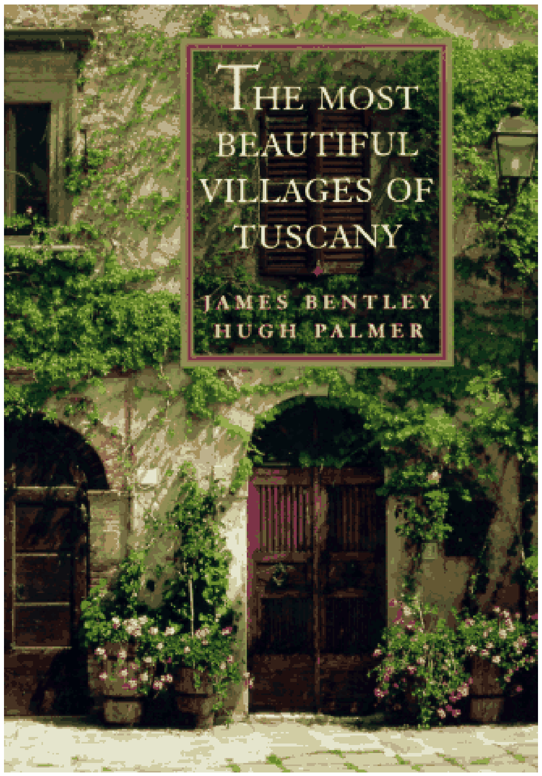
JPEG 75% (

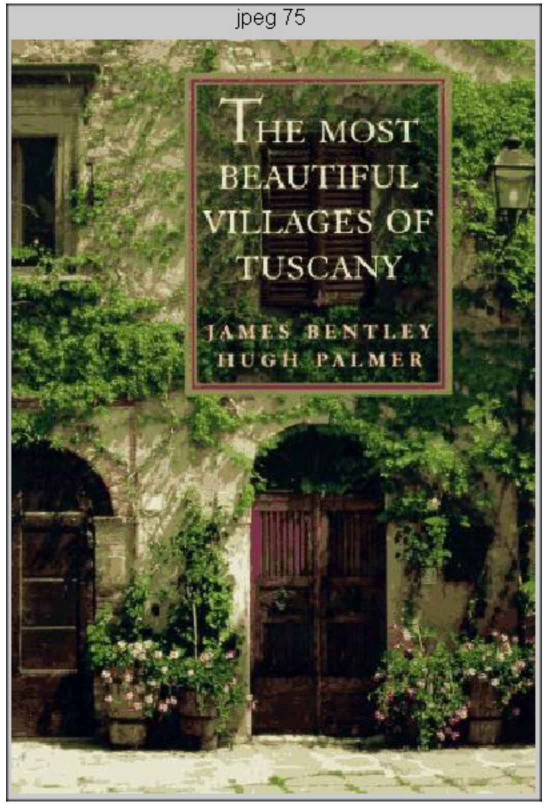
JPEG 50% (

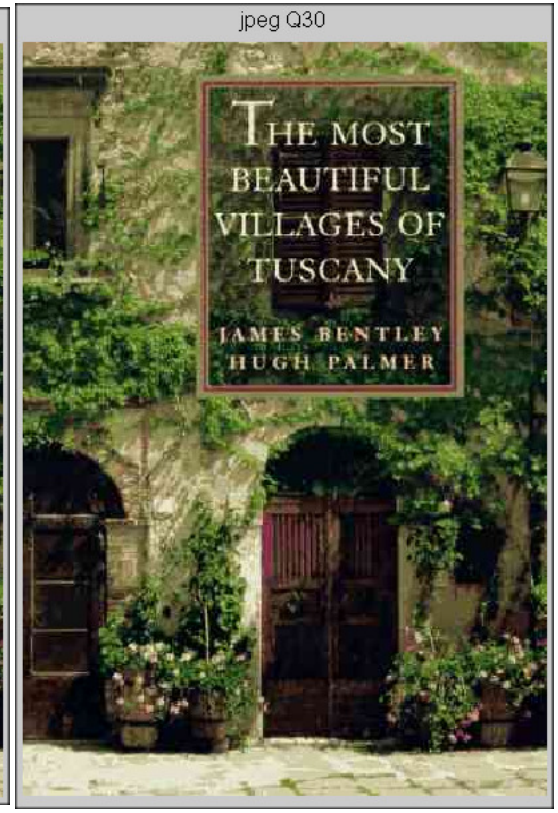
JPEG 30% (9

JPEG 10% (5KB)

JPEG Compression (Q=75% & 30%)







45 KB

22 KB

From Liu's EE330 (Princeton)

Y Cb Cr After JPEG (Q=30%)

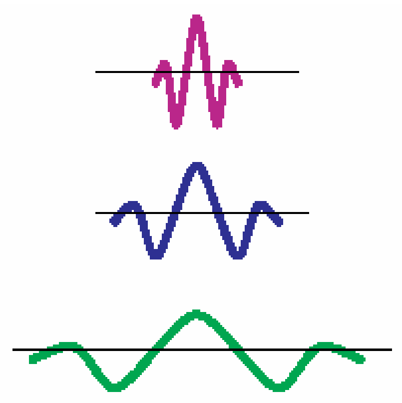
From Liu's EE330 (Princeton)

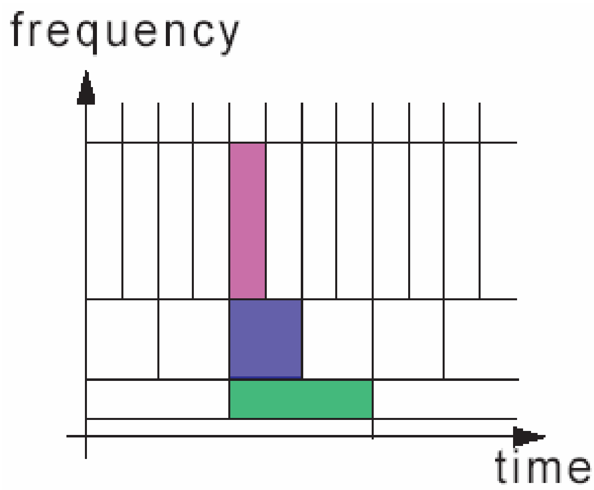
JPEG 2000

- Better image quality/coding efficiency, esp. low bit-rate compression performance
  - DWT
  - Bit-plane coding (EBCOT)
  - Flexible block sizes
  - ...
- More functionality
  - Support larger images
  - Progressive transmission by quality, resolution, component, or spatial locality
  - Lossy and Lossless compression
  - Random access to the bitstream
  - Region of Interest coding
  - Robustness to bit errors

Wavelets

- A wavelet is a square integrable function whose translates and dilates form an orthonormal basis for Hilbert space $L_2(R^N)$.
- Theory
  - Algebra, Geometry
  - Analysis (mainly studying functions and operators)
    - Fourier, Harmonic, Wavelets





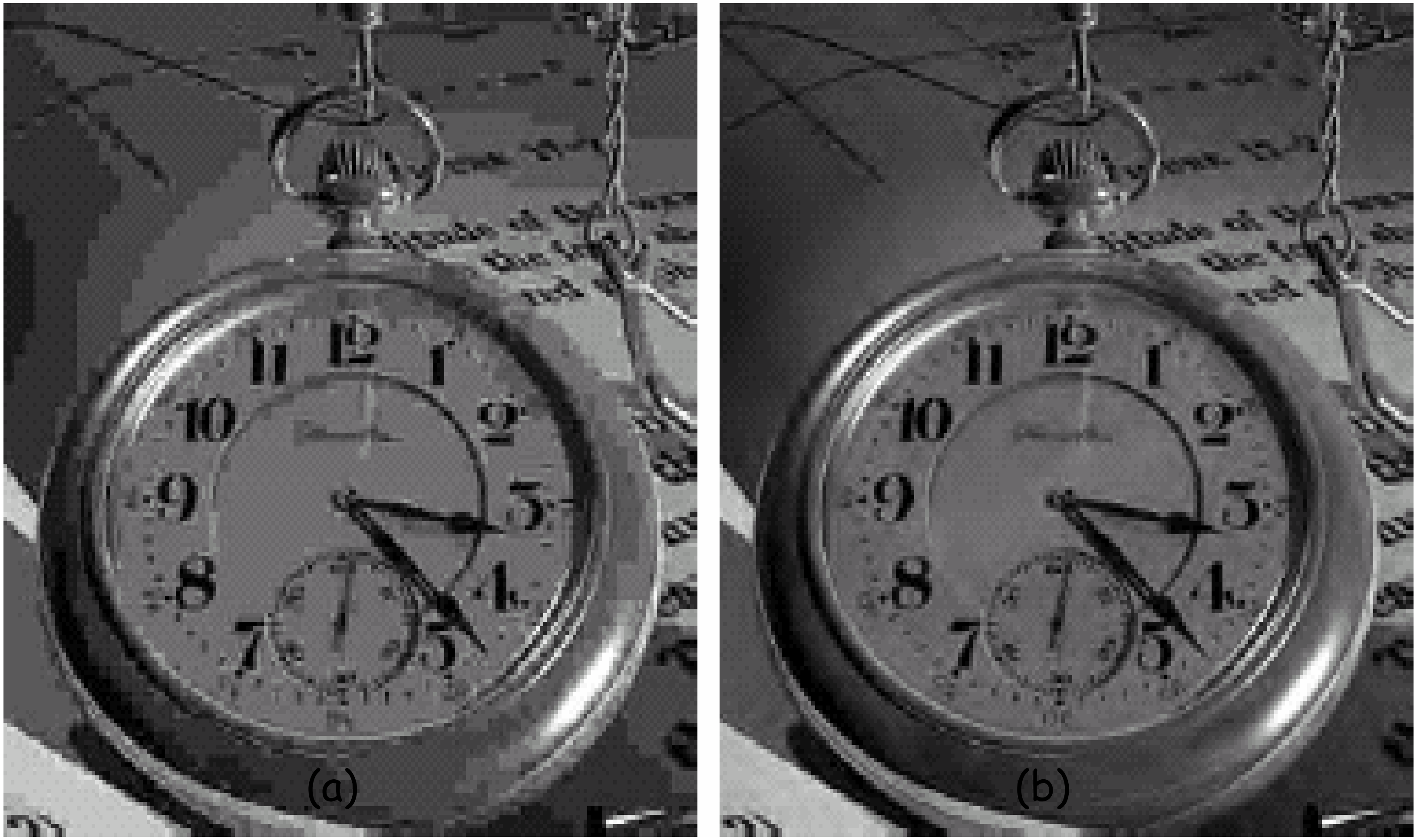
JPEG-2000 V.S. JPEG

(a)

(b)

Compression at 0.25 b/p by means of (a) JPEG (b) JPEG-2000

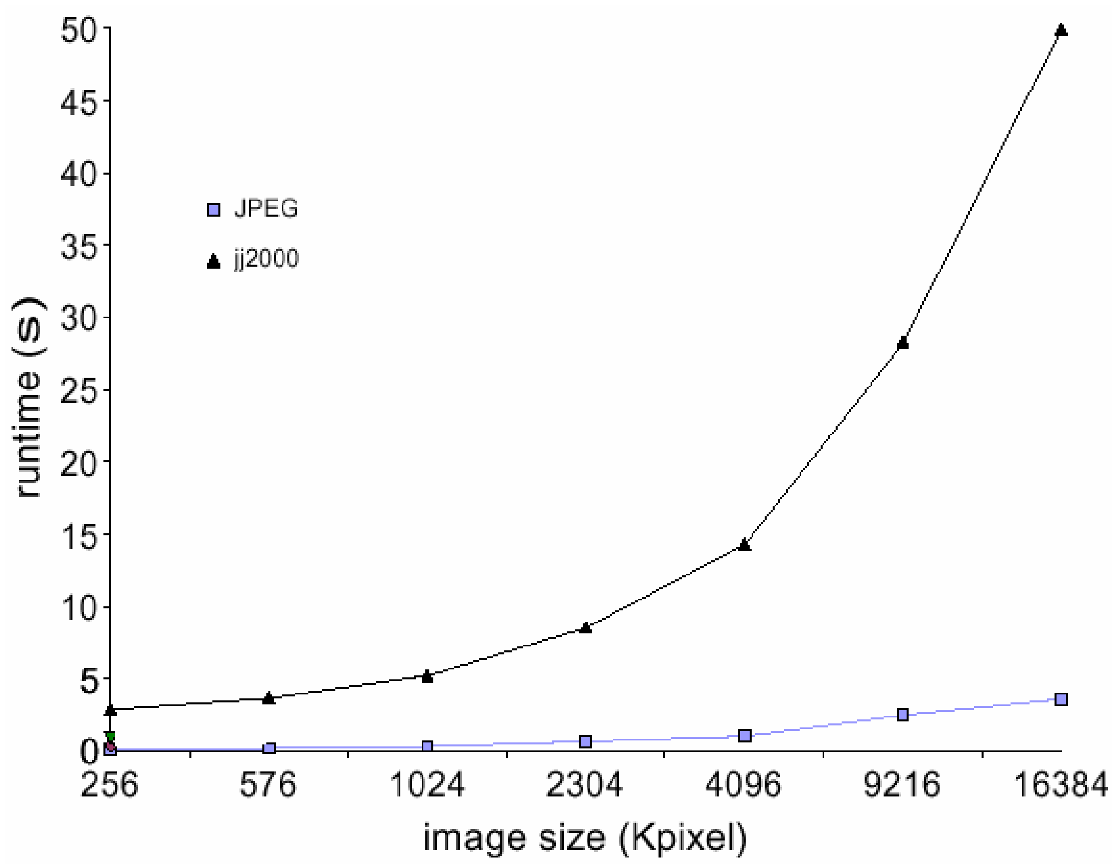
JPEG-2000 V.S. JPEG



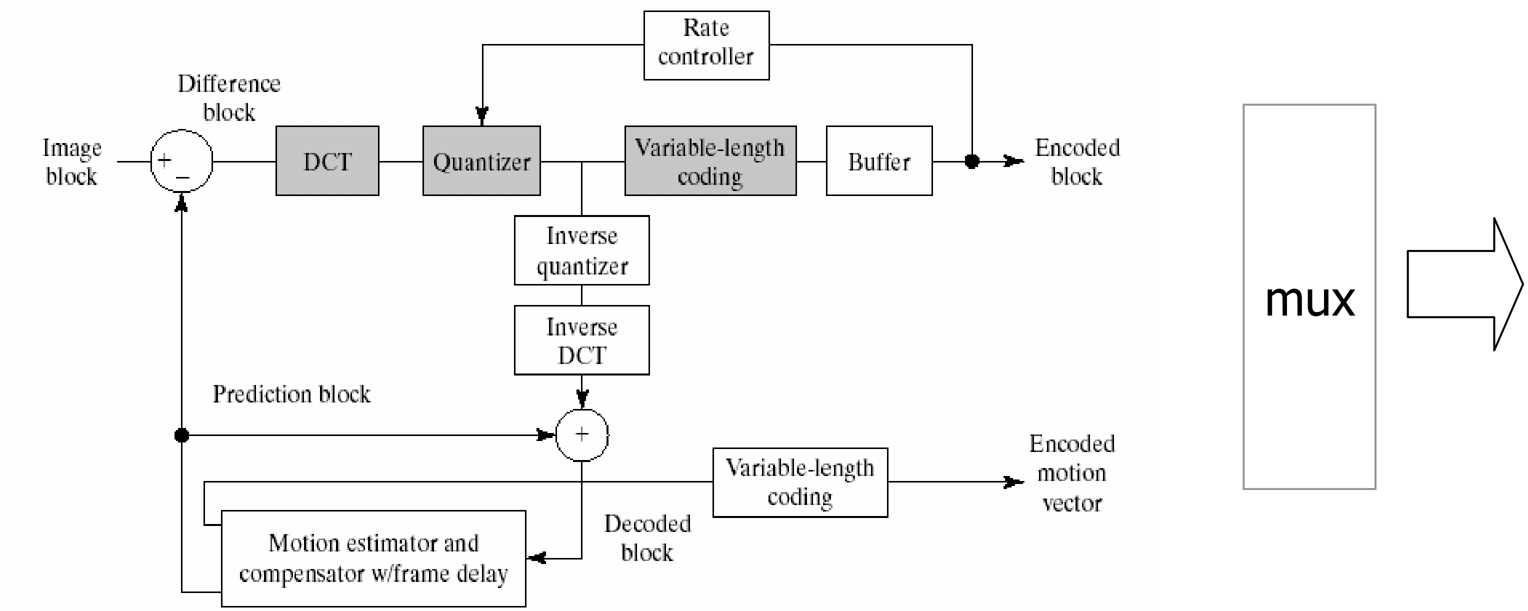
Compression at 0.2 b/p by means of (a) JPEG (b) JPEG-2000

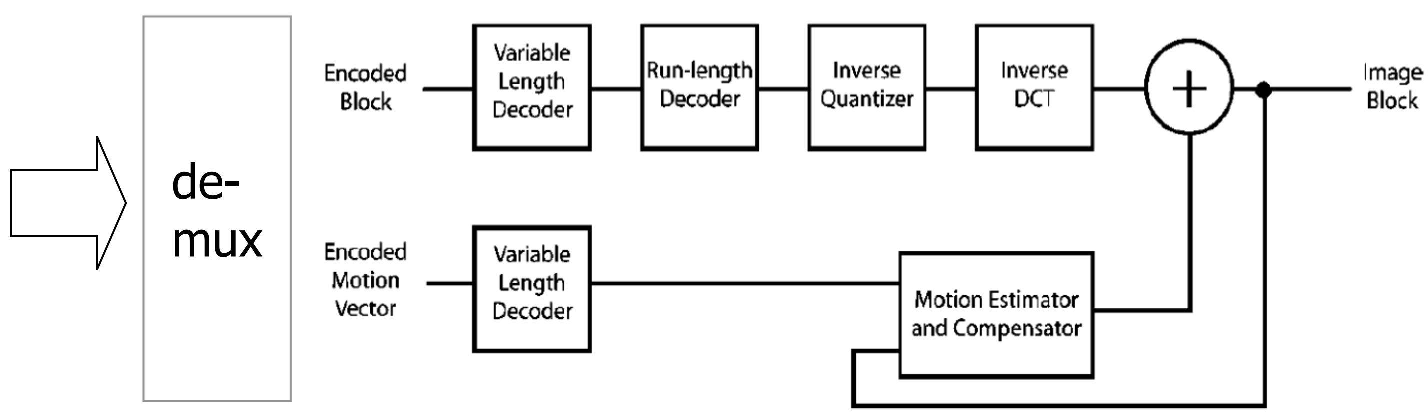
The trade-off:

JPEG2000 has a much Higher computational complexity than JPEG, especially for larger pictures.



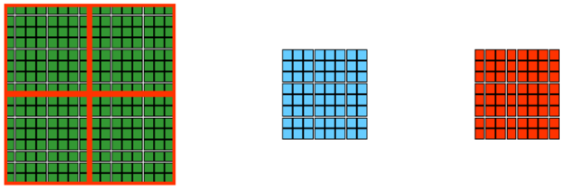
Hybrid Video Coding System



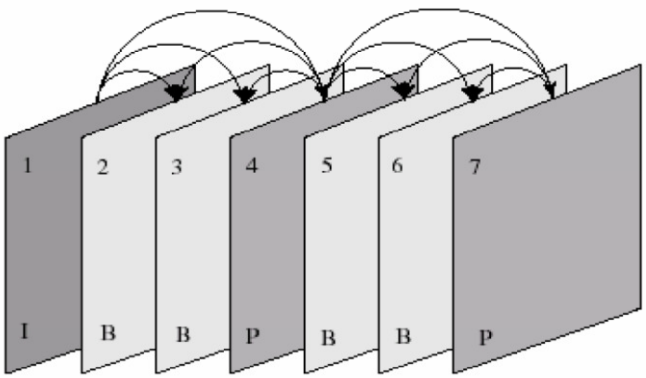


A Few Key Ideas in Video Coding

- Work on each macroblock (MB) (16x16 pixels) independently for reduced complexity
  - Motion compensation done at the MB level
  - DCT coding at the block level (8x8 pixels)



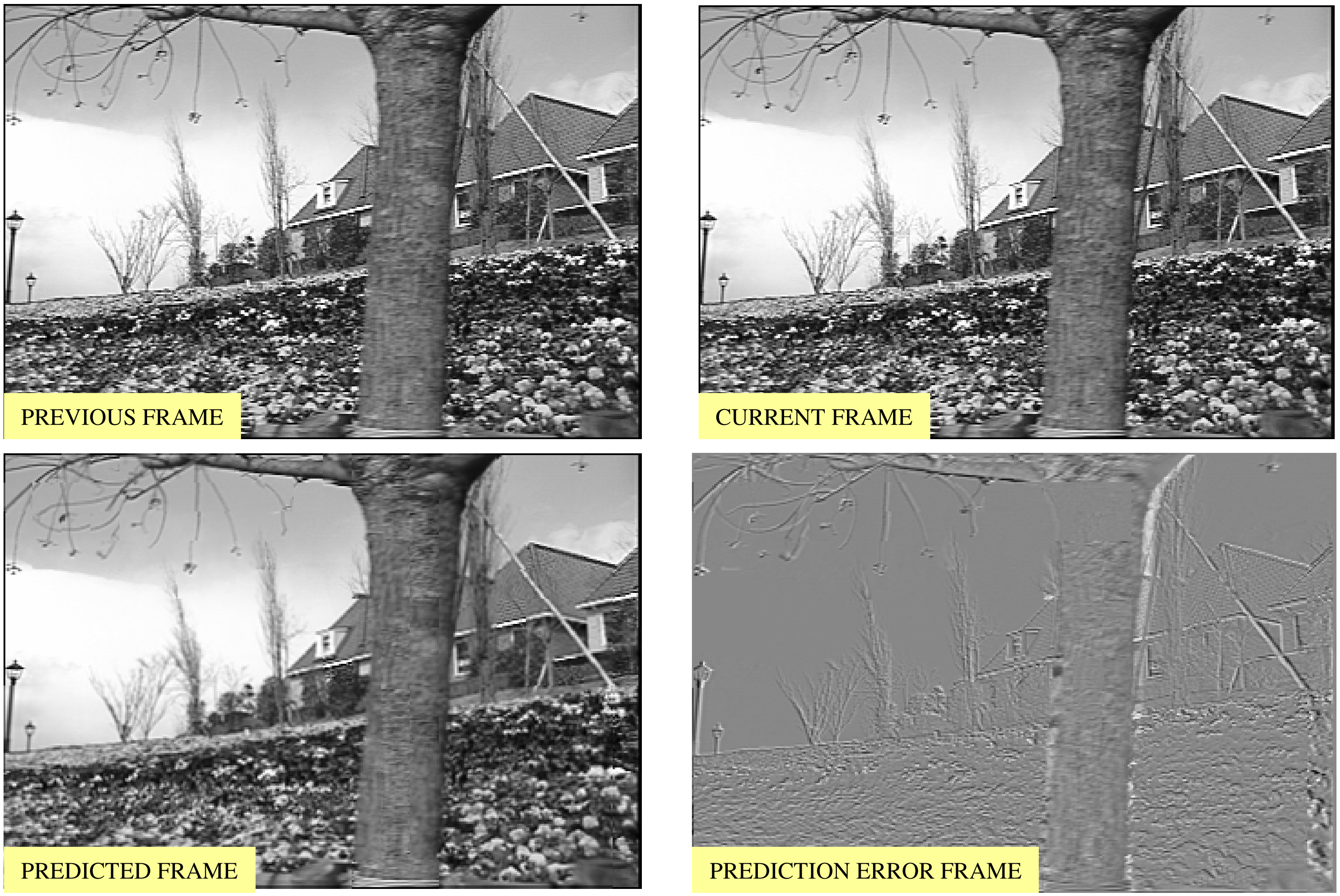
- Use block or frame correlation to predict new data from seen data
  - Predict a current block from previously coded blocks in the same frame --- *Intra* prediction (introduced in the latest standard H.264)
  - Predict a new frame from a previous frame and only code the prediction error --- *Inter* prediction on "B" and "P" frames
  - Prediction errors have smaller energy than the original pixel values and can be coded with fewer bits
    - DCT on the prediction errors
  - Those regions that cannot be predicted well will be coded directly using DCT --- Intra coding without intra-prediction



Encoding order: 1 4 2 3 7 5 6

Motion Compensation

- Help reduce temporal redundancy of video



Motion Estimation

- Help understanding the content of image sequence
  - For surveillance
- Help reduce temporal redundancy of video
  - For compression
- Stabilizing video by detecting and removing small, noisy global motions
  - For building stabilizer in camcorder
- A hard problem in general!

Block-Matching by Exhaustive Search

- Assume block-based translation motion model
- Search every possibility over a specified range for the best matching block
  - MAD (mean absolute difference) often used for simplicity

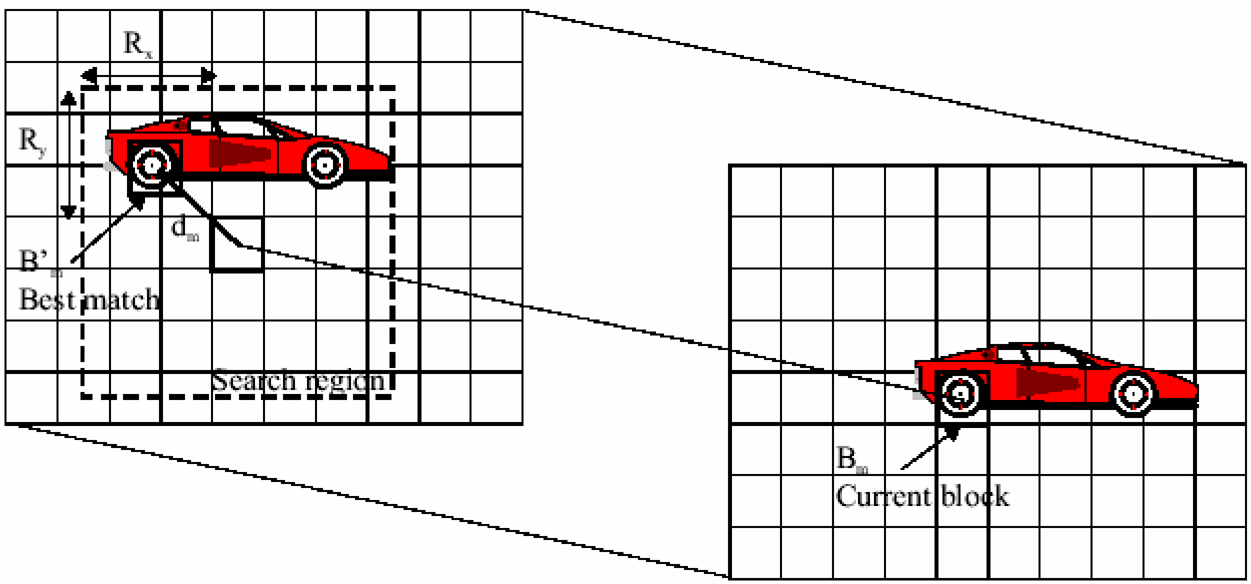


Figure 6.6. The search procedure of the exhaustive block matching algorithm.

From Wang's
Preprint Fig.6.6

Fractional Accuracy Search for Block Matching

- For motion accuracy of 1/K pixel
  - Upsample (interpolate) reference frame by a factor of K
  - Search for the best matching block in the upsampled reference frame
- Half-pel accuracy ~ K=2
  - Significant accuracy improvement over integer-pel (esp. for low-resolution)
  - Complexity increase

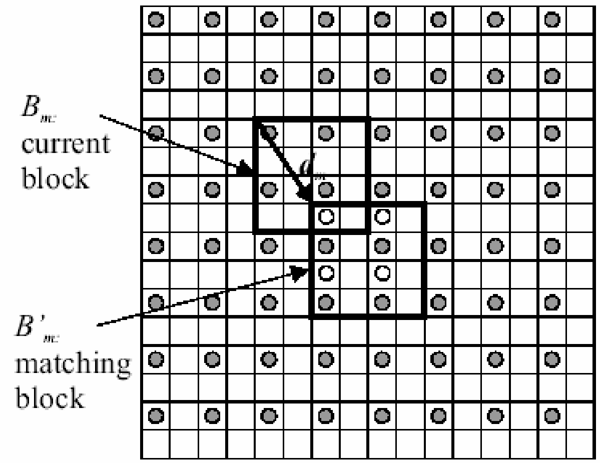


Figure 6.7. Half-pel accuracy block matching. Filled circles are samples existing in the original tracked frame, open circles are samples to be interpolated for calculating the matching error, for a candidate MV $\mathbf{d}_m = (-1, -1.5)$. Instead of calculating these samples on-demand for each candidate MV, a better approach is to pre-interpolate the entire tracked frame.

(From Wang's Preprint Fig.6.7)

Complexity of Exhaustive Block-Matching

- Assumptions
  - Block size NxN and image size S=M1xM2
  - Search step size is 1 pixel ~ *"integer-pel accuracy"*
  - Search range +/–R pixels both horizontally and vertically
- Computation complexity
  - # Candidate matching blocks = $(2R+1)^2$
  - # Operations for computing MAD for one block ~ $O(N^2)$
  - # Operations for MV estimation per block ~ $O((2R+1)^2 N^2)$
  - # Blocks = S / N^2
  - Total # operations for entire frame ~ $O((2R+1)^2 S)$
    - i.e., overall computation load is independent of block size!
- E.g., M=512, N=16, R=16, 30fps

  => On the order of 8.55×10^9 operations per second!
  - Was difficult for real time estimation, but possible with parallel hardware

Exhaustive Search: Cons and Pros

- Pros
  - Guaranteed optimality within search range and motion model
- Cons
  - Can only search among finitely many candidates
    - What if the motion is "fractional"?
  - High computation complexity
    - On the order of [search-range-size * image-size] for 1-pixel step size

➔ How to improve accuracy?
  - Include blocks at fractional translation as candidates
    => require interpolation

➔ How to improve speed?
  - Try to exclude unlikely candidates

Fast Algorithms for Block Matching

- Basic ideas
  - Matching errors near the best match are generally smaller than far away
  - Skip candidates that are unlikely to give good match

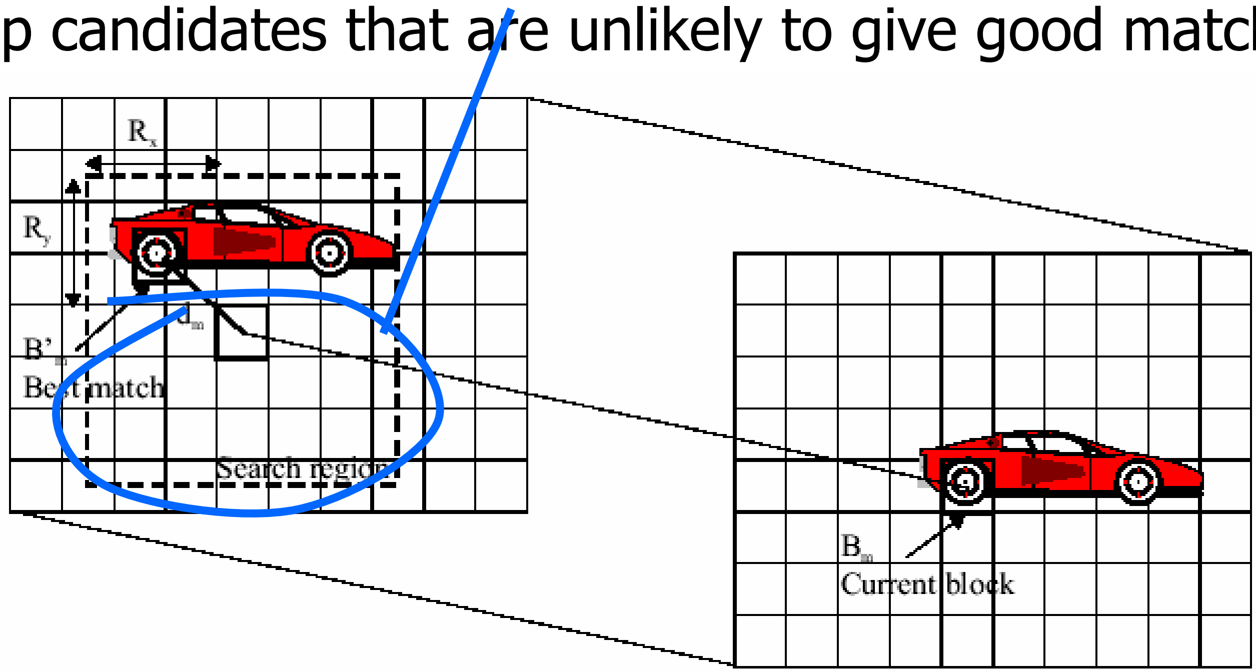
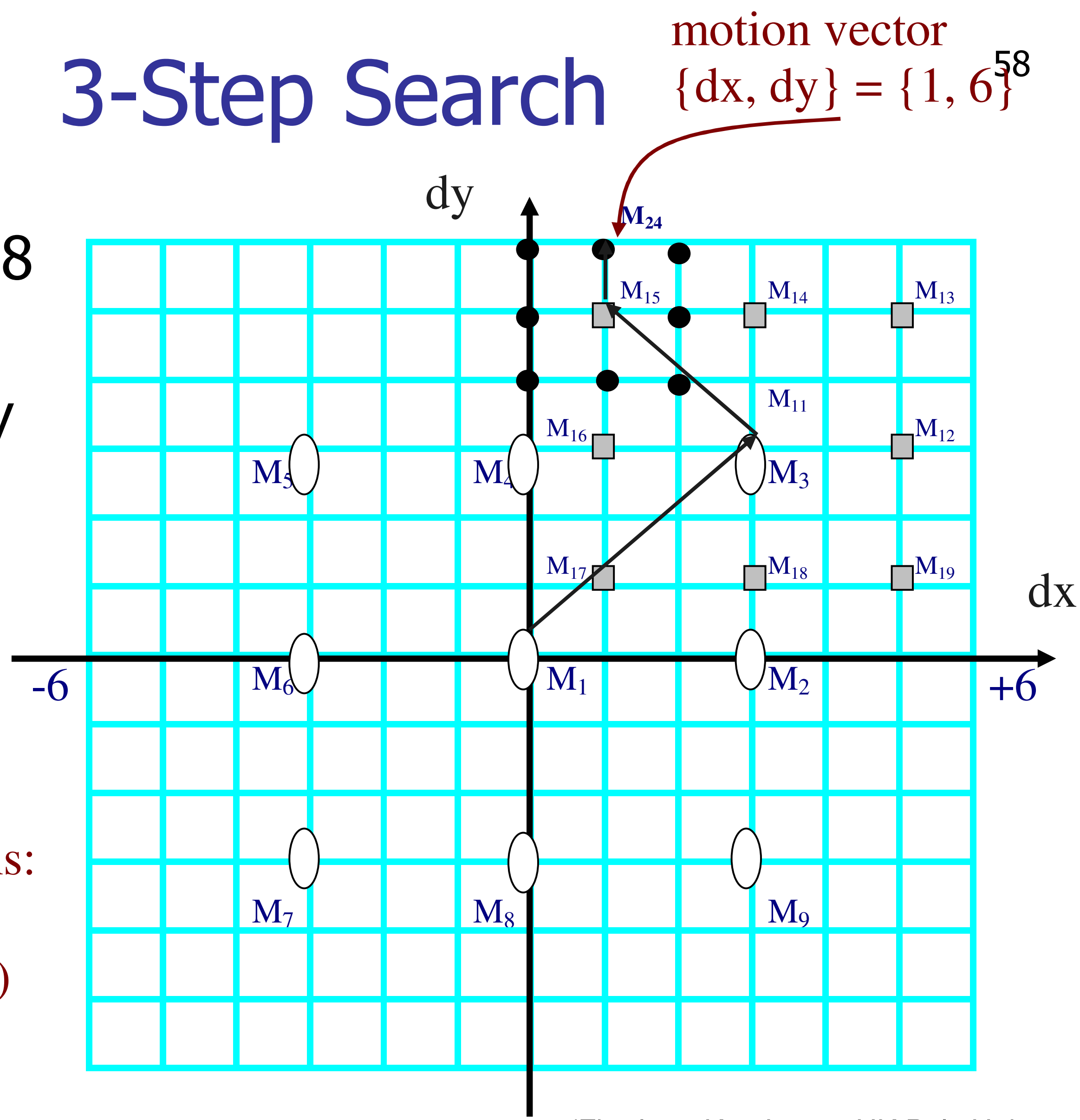


Figure 6.6. The search procedure of the exhaustive block matching algorithm.

(From Wang's Preprint Fig.6.6)

Fast Algorithm: 3-Step Search

- Search candidates at 8 neighbor positions
- Step-size cut down by 2 after each iteration
  - Start with step size approx. half of max. search range



Total number of computations:

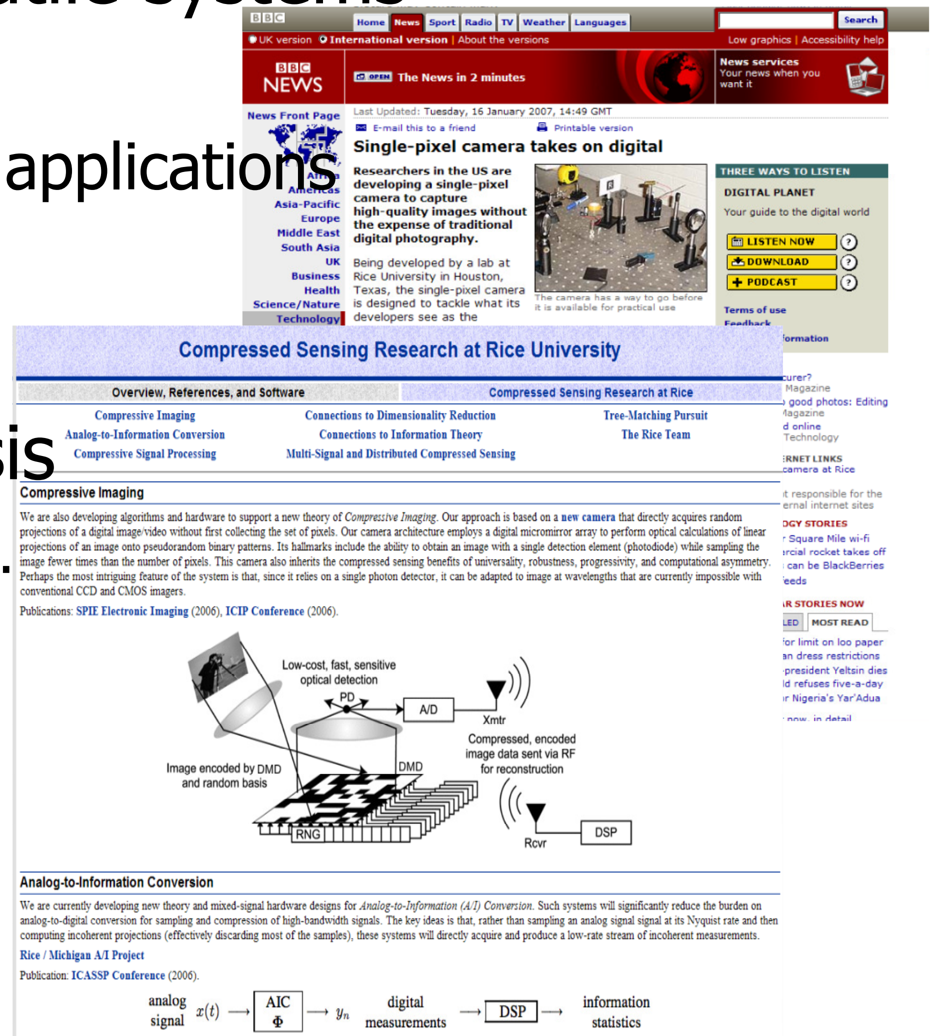
$9 + 8\times 2 = 25$ (3-step)

$(2R+1)^2 = 169$ (full search)

(Fig. from Ken Lam – HK Poly Univ. short course in summer'2001)

Recent Activities in Image Compression

- Build better, more versatile systems
  - High-definition IPTV
  - Wireless and embedded applications
  - P2P video delivery
- In search for better basis
  - Curvelets, contourlets, …
- “compressed sensing”



Summary

- The image/video compression problem
- Source coding
  - For i.i.d. symbols
  - For symbol streams
- Image/video compression systems
  - MPEG/JPEG and beyond
- Next time: multimedia indexing and image reconstruction in medical applications

Part of the slides/materials gratefully taken from:
Wade Trappe (Rutgers), Min Wu (UMD), Yao Wang (poly tech), Xiuzhen Huang (UCSB), Tony Lin (PKU)