Multimedia Information Systems

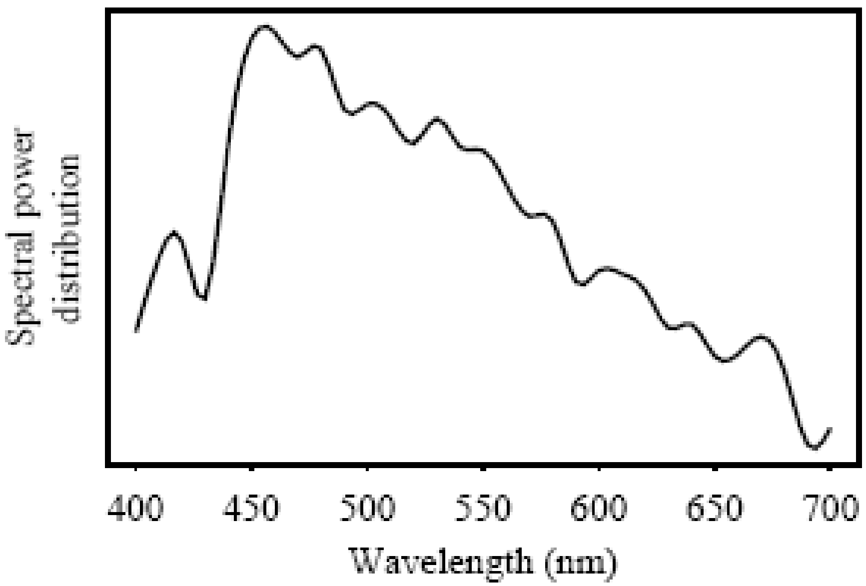
Samson Cheung

EE 639, Fall 2004

Lecture 3 & 4: Color, Video, and Fundamentals of Data Compression

Color Science

- **Light is an electromagnetic wave. Its color is characterized by the wavelength content of the light**
  - **Visible : 400nm – 700 nm**



Spectral power distribution $E(\lambda)$ of a signal (daylight)

Human Vision

- **Sensors**
  - **Rods – night vision**
  - **Cones – red, green, blue**

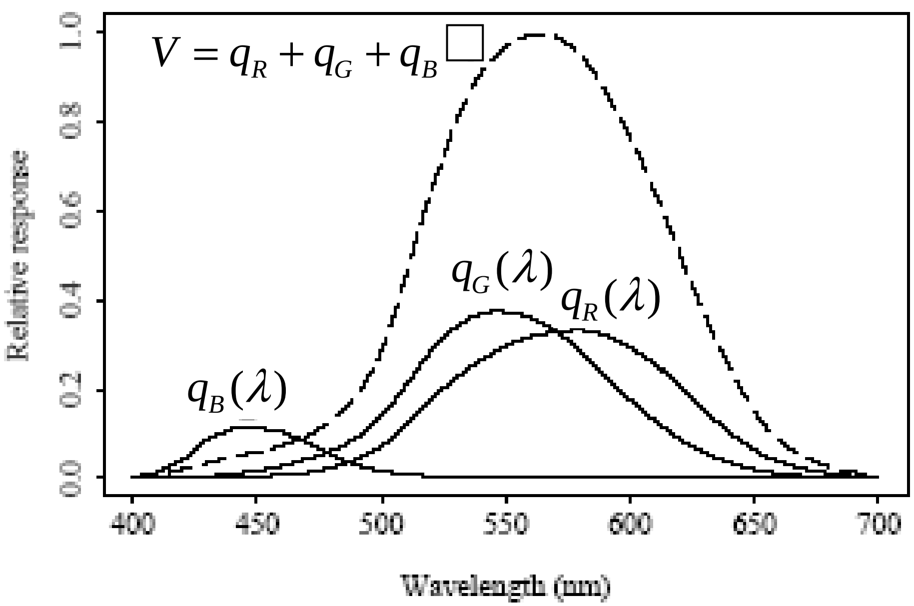
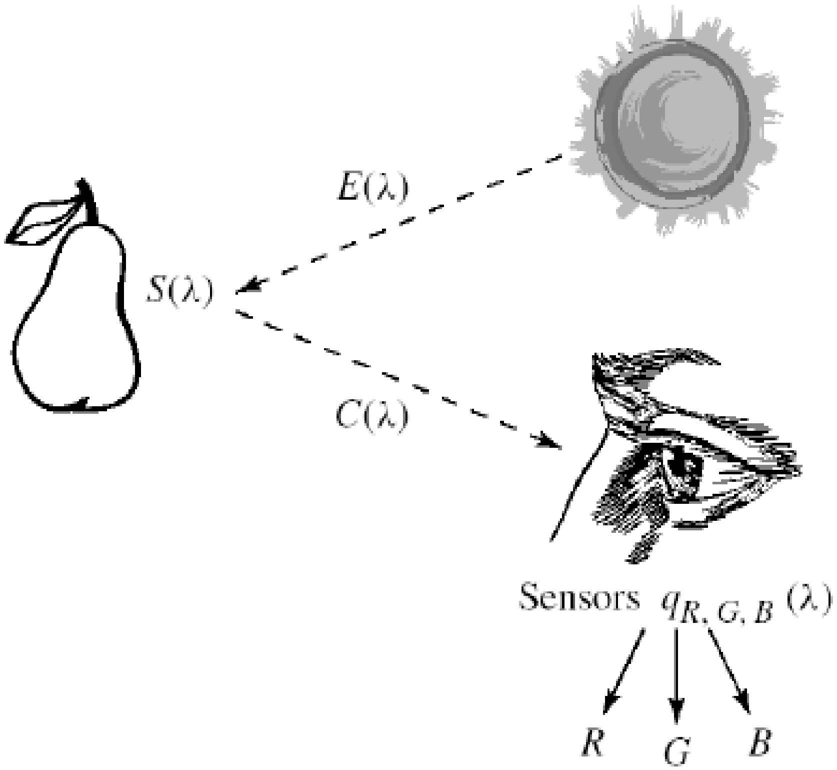
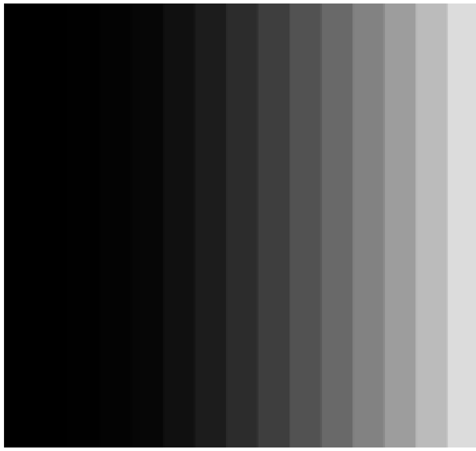


Image Formation

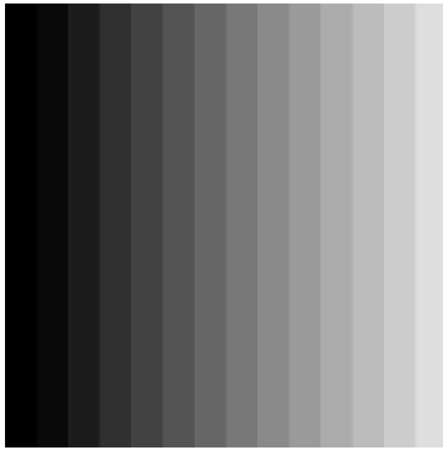


Gamma Correction (CRT)

- **The light emitted from CRT is roughly proportional to the voltage raised to a power.**
- **This power is called gamma** $\gamma \sim 2.2$
- **Gamma correction :**



Not Corrected

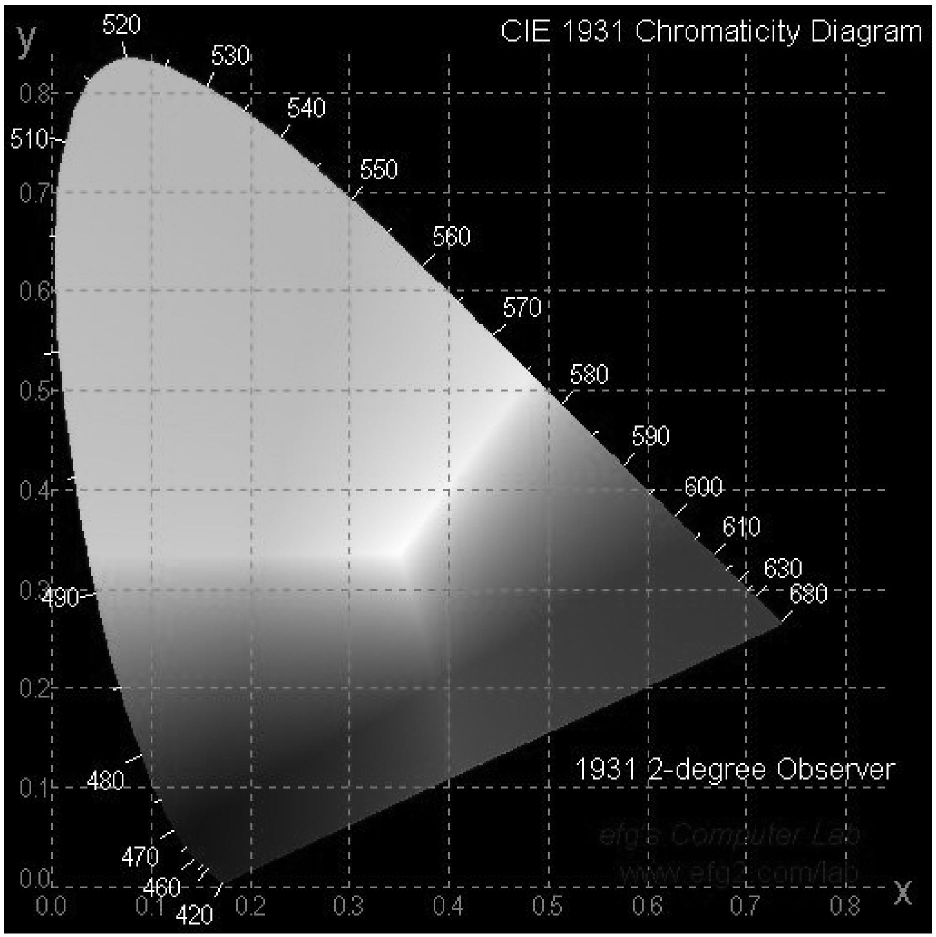


Gamma Corrected

Different Color Coordinates

- Different numerical representations to facilitate the applications
- Coordinate systems:
  - RGB - monitor
  - CMYK - printer
  - CIE Chromaticity XYZ
  - L*a*b* and L*u*v*
  - HSV, HSV, HLS
  - YIQ, YUV, YCbCr

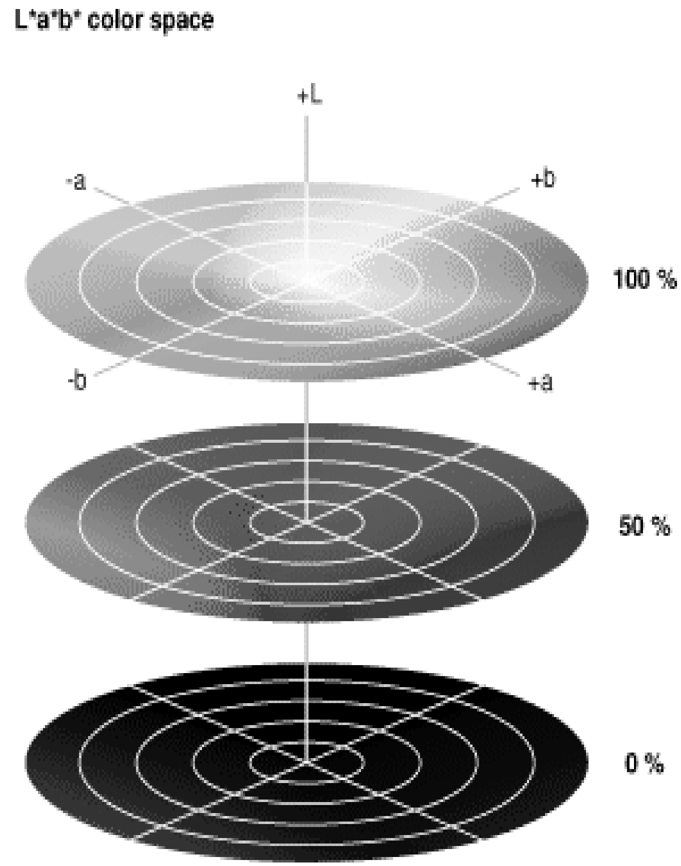
CIE 1931 Chromaticity Diagram



- Visible color
- Normalized intensity

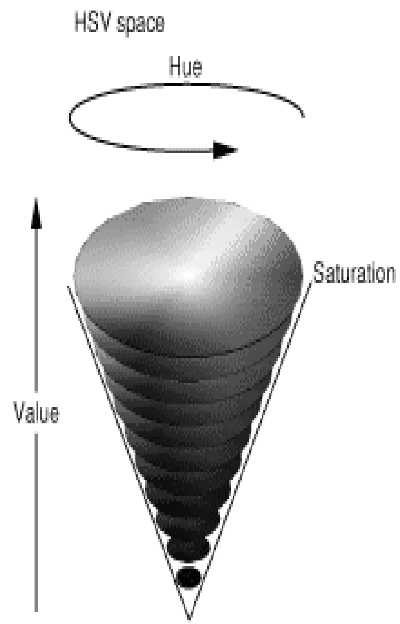
L*u*v* and L*a*b*

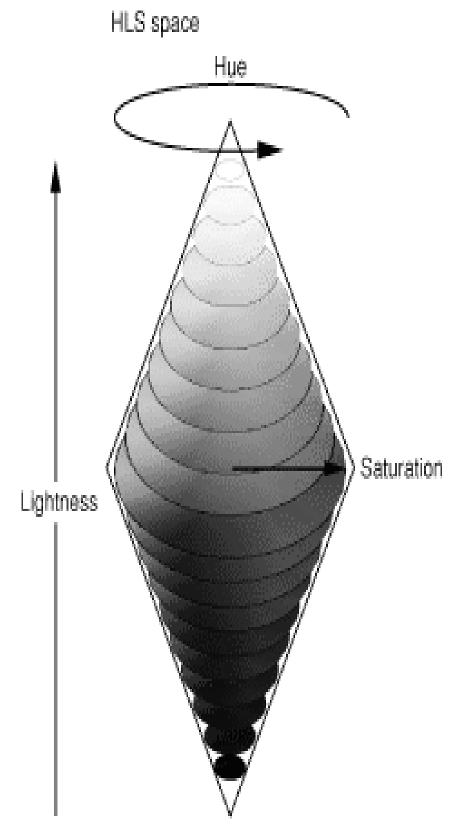
- Nonlinear transformations of the XYZ tristimulus space.
- Designed to have a more uniform correspondence between geometric distances and perceptual distances between colors that are seen under the same reference illuminant.
- L*u*v* for additive color
- L*a*b* for subtractive



HSI, HLS, HSV

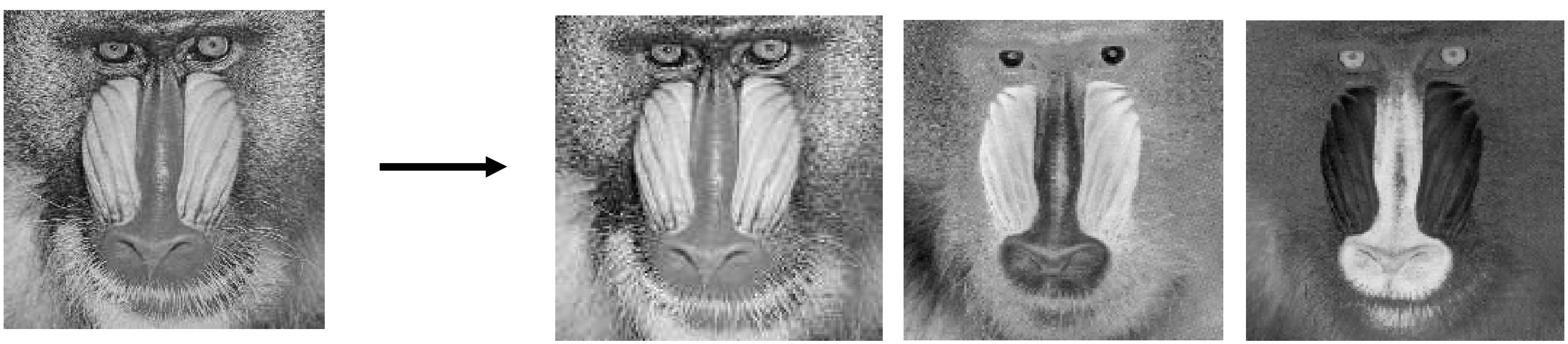
- Intuitive Color
  - HLS = Hue, Lightness, Saturation
  - HSI = Hue, Saturation, Intensity
  - HSV = Hue, Saturation, Value
- Non-linear transform





YUV, YCbCr, YIQ

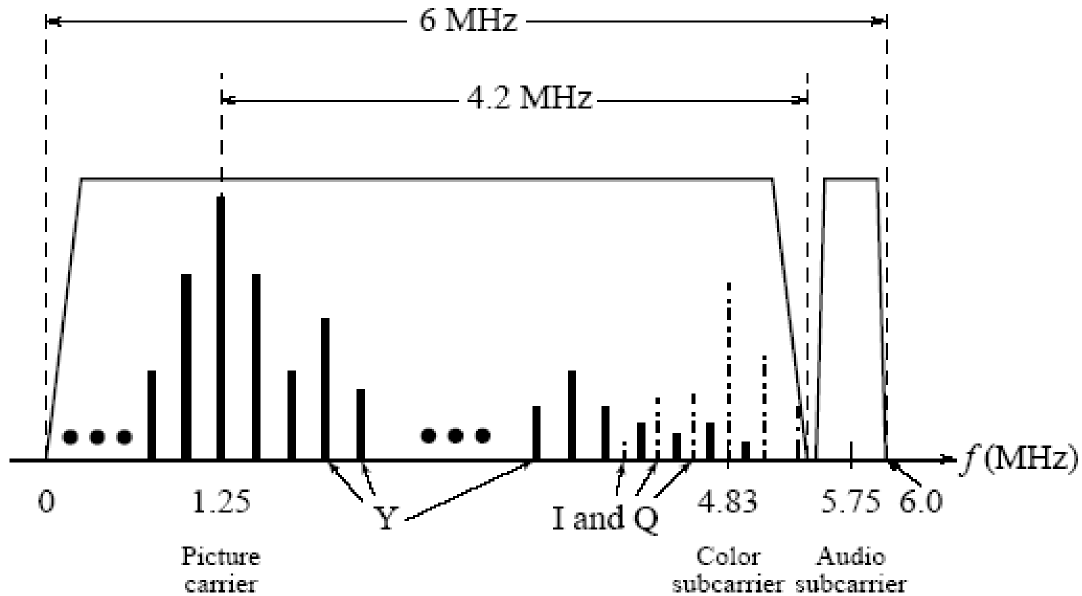
- **Linear transform of RGB to achieve energy compression**
  - **YUV = analog color space for television**
  - **YIQ = analog color space for NTSC composite, rotate YUV by 33^o to further band-limit Q**
  - **YCbCr = discrete and scaled YUV**



Y U V

Analog Video Signals

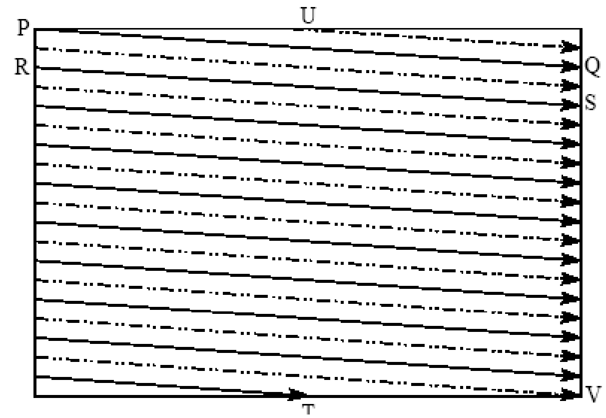
- **Type of analog signals**
  - **3 wires – component (R,G,B)**
  - **2 wires – S-video (1 for luminance, 1 for chrom.)**
  - **1 wire – composite (frequency multiplexed all together)**



NTSC frequency sprectrum

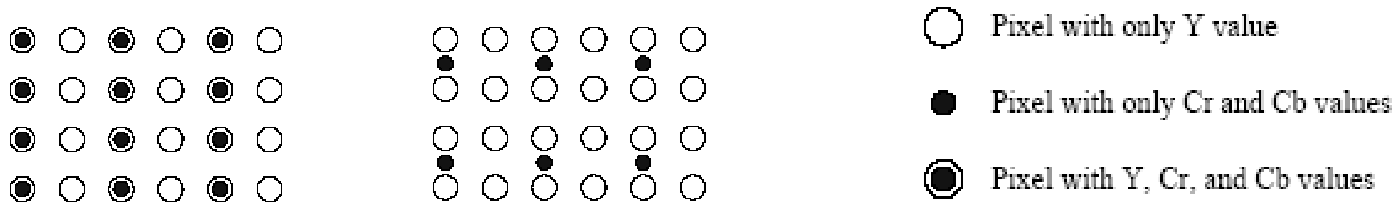
Analog television

- **All interlace signals**



TV System	Frame Rate (fps)	# of Scan Lines	Total Channel Width (MHz)	Bandwidth Allocation (MHz)		
				Y	I or U	Q or V
NTSC	29.97	525	6.0	4.2	1.6	0.6
PAL	25	625	8.0	5.5	1.8	1.8
SECAM	25	625	8.0	6.0	2.0	2.0

Digital Video Signals



4:2:2 4:2:0

	CCIR 601 525/60 NTSC	CCIR 601 625/50 PAL/SECAM	CIF	QCIF
Luminance resolution	720 × 480	720 × 576	352 × 288	176 × 144
Chrominance resolution	360 × 480	360 × 576	176 × 144	88 × 72
Color Subsampling	4:2:2	4:2:2	4:2:0	4:2:0
Aspect Ratio	4:3	4:3	4:3	4:3
Fields/sec	60	50	30	30
Interlaced	Yes	Yes	No	No

HDTV

Table 5.4: Advanced Digital TV formats supported by ATSC

# of Active Pixels per line	# of Active Lines	Aspect Ratio	Picture Rate
1,920	1,080	16:9	60I 30P 24P
1,280	720	16:9	60P 30P 24P
704	480	16:9 & 4:3	60I 60P 30P 24P
640	480	4:3	60I 60P 30P 24P

- **Larger frame size**
- **Non-interlaced or progressive (P) format**
- **16:9 aspect ratio**
- **Standard by 2006 mandated by FCC**

Fundamental of compression

- **Two types:**
  - **Lossless**
    - file, high-quality media
  - **Lossy**
    - trade-off quality with bitrate
- **Information content: average entropy**

- **Shannon's Source Coding Theorem:**
  N i.i.d. random variables, each with entropy $H(X)$, can be compressed into $NH(X)$ bits with negligible loss as $N \rightarrow \infty$.

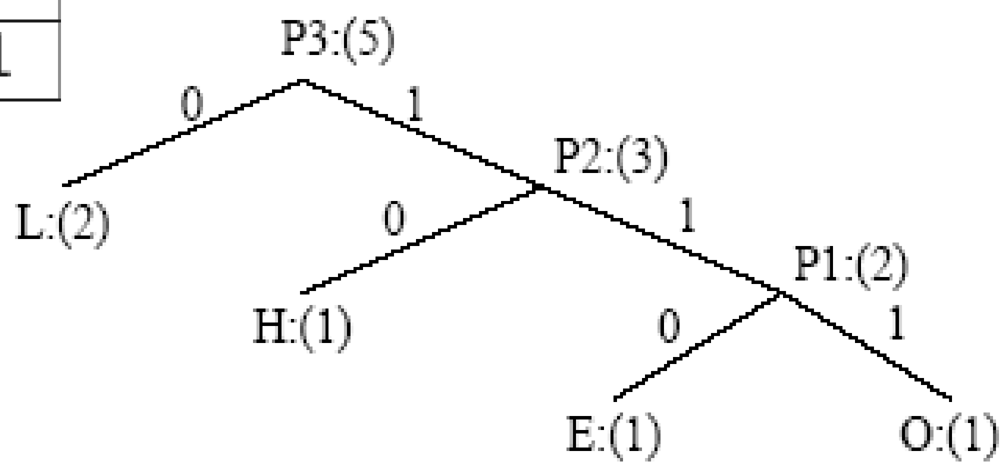
Lossless compression

- **Memory-less sources: approximate entropy using variable length code**
  - **Huffman codes**
  - **Arithmetic codes**
- **Memory sources: exploit redundancy**
  - **Run-length codes**
  - **Library based codes**
  - **Context-based methods**
  - **Transform coding**
    - Linear prediction
    - Others

Huffman Code

- Binary Huffman tree
  - leaf nodes represent symbol and each branch represents 0 or 1.
  - Traversal from root to leaf represents the codeword.

Symbol	H	E	L	O
Count	1	1	2	1



Huffman code (cont.)

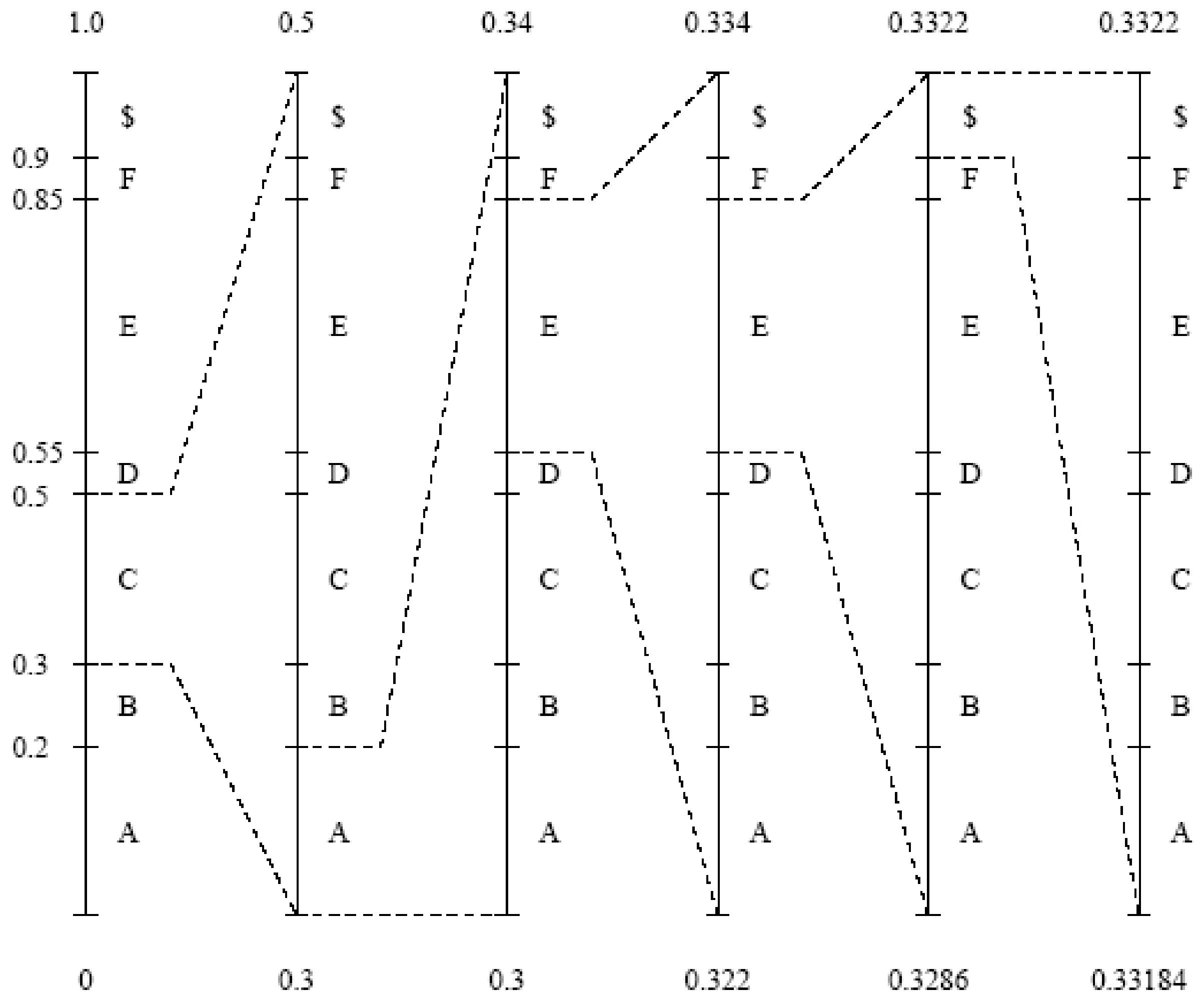
- **Advantages:**
  - **Pre-fix property – no decoding ambiguity**
  - **Easy to decode – traverse down the tree**
- **Disadvantages:**
  - **Each symbol must use at least one bit.**
    - Blocking
  - **Changing statistics**
    - Update Huffman tree to reflect statistics

Arithmetic Coding

- Represent the entire sequence of symbols as a real number
- Each symbol is mapped to an interval whose length is equal to its probability occurrence
- Higher probability → longer interval → lower precision needed → fewer bits
- High-level steps
  1. Select the interval that corresponds to the symbol
  2. Send leading bits if known
  3. Renormalize the selected interval
  4. Partition again based on probability
  5. Go back to 1.

Example of Arithmetic coding

Symbol	Probability	Range
A	0.2	[0, 0.2)
B	0.1	[0.2, 0.3)
C	0.2	[0.3, 0.5)
D	0.05	[0.5, 0.55)
E	0.3	[0.55, 0.85)
F	0.05	[0.85, 0.9)
$	0.1	[0.9, 1.0)



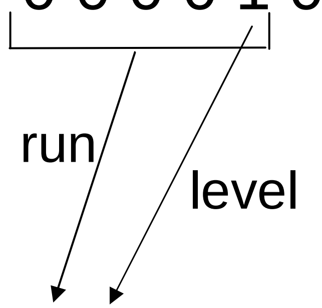
Arithmetic Coding (cont.)

- Advantages
  - Asymptotically optimal
  - Use less than one bit to code a symbol
  - Easier than Huffman to update statistics – does not require tree rebuilt
- Disadvantages
  - Higher implementation cost – need multiplication for normalization (except binary)
  - Very sensitive to transmission error

Run-length code

- **Commonly used in image and video coding**
- **Run + Level**

0 0 0 0 1 0 0 0 0 0 5 0 0 4 0 0 0 0 8 0 0 9 0

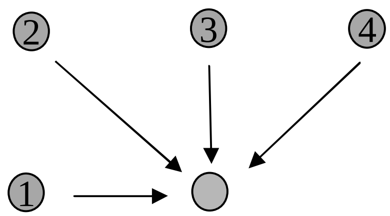


(5,1) (6,5) (3,4) (5,8) (3,9) (1,0)

Library based codes

- **Uses fixed-length codewords to represent variable-length strings of symbols/characters that commonly occur together**
- **The encoder and decoder build up the same dictionary dynamically while receiving the data.**
- **The codec places longer and longer repeated entries into a dictionary, and then emits the *code* for an element, rather than the string itself, if the element has already been placed in the dictionary.**
- **LZ77: gzip; LZW: UNIX *compress*, GIF, V.42 bis**

Context-based prediction



1234 =0000

●	Pb.
0	0.4
1	0.6

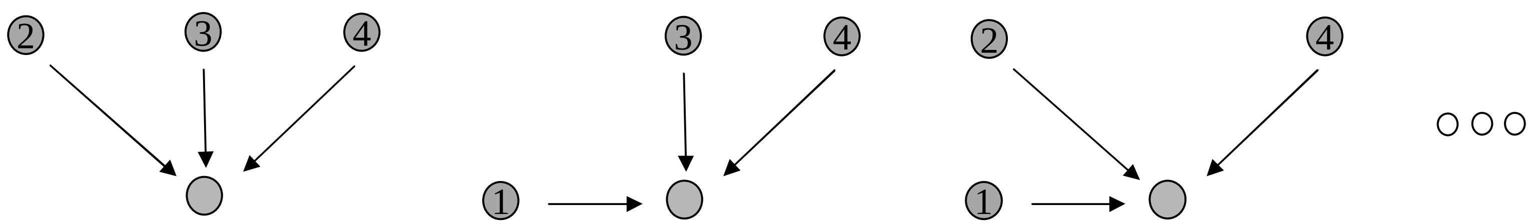
1234 =0001

●	Pb.
0	0.9
1	0.1

1234 =0010

●	Pb.
0	0.1
1	0.9

○○○



- One PDF for each context
- Even simple context requires LOTS of tables : 81
- Two solutions:
  - Model ex. Context-Weighting Trees (Willems, Shtarkov, Tjalkens 95)
  - Hand-picked like in H.264

Transform Coding

- **Need to code**

- **Goal: Devise a transform**

1. **Burrows Wheeler Transform**
2. **Decorrelated (Whitening) Transform: Transform X's into independent Y's**

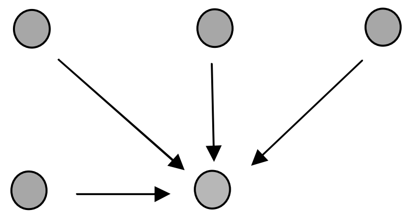
- Equality hold if they are all independent.

Burrows Wheeler Transform

BURROWS		BURROWS	S		ABC…:18	
URROWSB		OWSBURR	R		SAB…:18	
RROWSBU		ROWSBUR	R		RSA…:0	
ROWSBUR	→ sort	RROWSBU	U	→ MTF	RSA…:20	→ Runlength And Huffman
OWSBURR		SBURROW	W		URS…:21	
WSBURRO		URROWSB	B		WUR…:5	
SBURROW		WSBURRO	O		BWU…:18	

- **Used in bzip, bzip2; best text compression to date**
- **Burrows and Wheeler 1994.**
- **Readable account by Mark Nelson, Dr. Dobb's Journal, 1996**

Linear Prediction



- (-4 -1 -1 -1) and all its shifted versions form the transform
- Many other transforms : DCT, Wavelet
- Typically considered as lossless because of the floating representation.