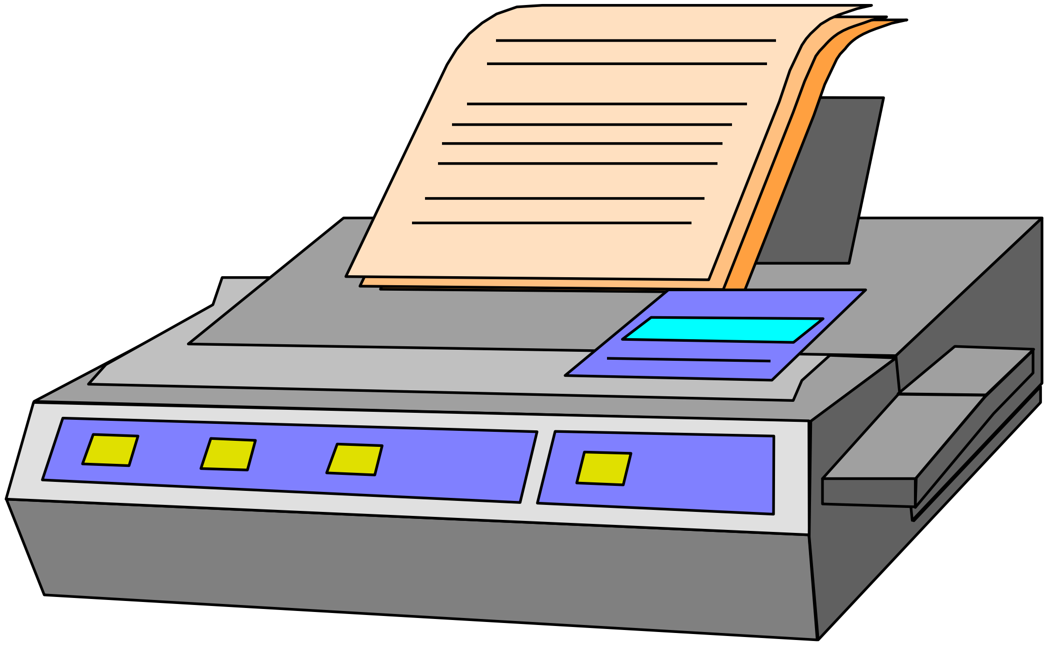
FAX Image Compression



Nimrod Peleg
Update: May 2009

FAX: Historical Background

- Invented in 1843, by Scottish physicist Alexander Bain (English Patent No. 9,745 for recording telegraph, facsimile unit)
- Based on paper, saturated with electrolytic solution, changes its color when electric current passes through it
- Note that:
  - Telegraph, (Morse) : 1844
  - Telephone, (A.G Bell) : 1876

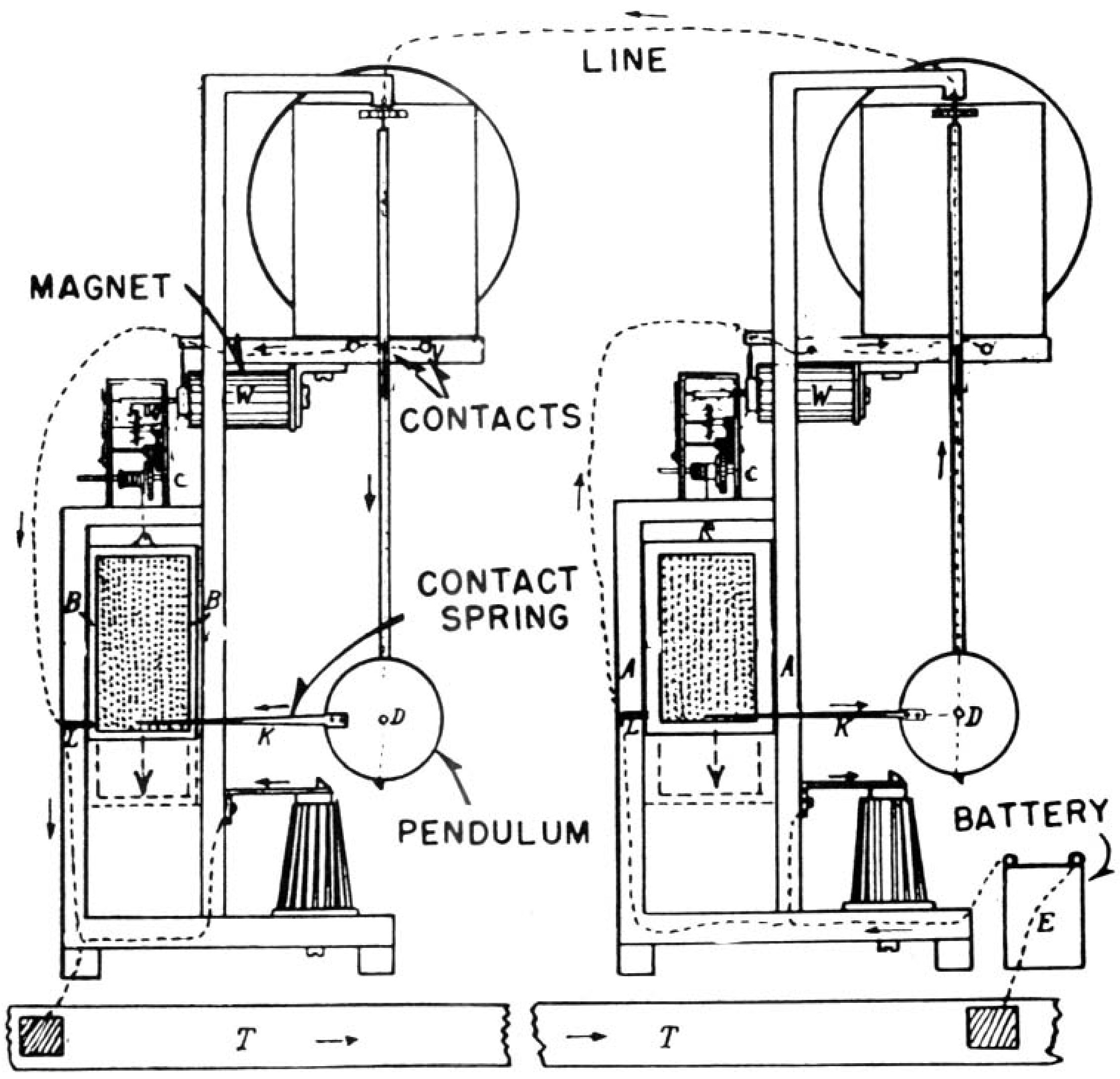


Figure 1.1.1 Bain's recording telegraph.

Pantelegraph, 1861

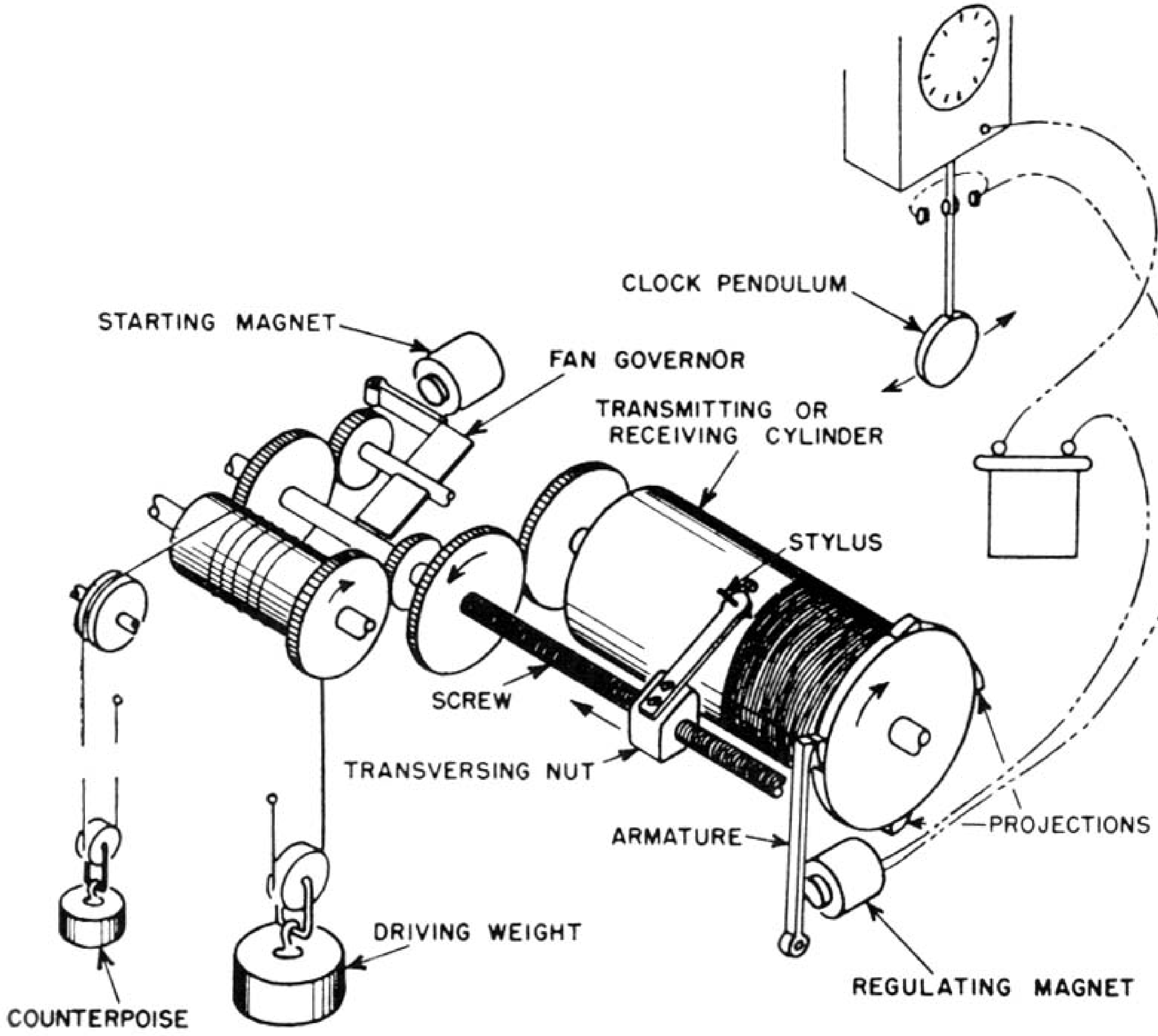
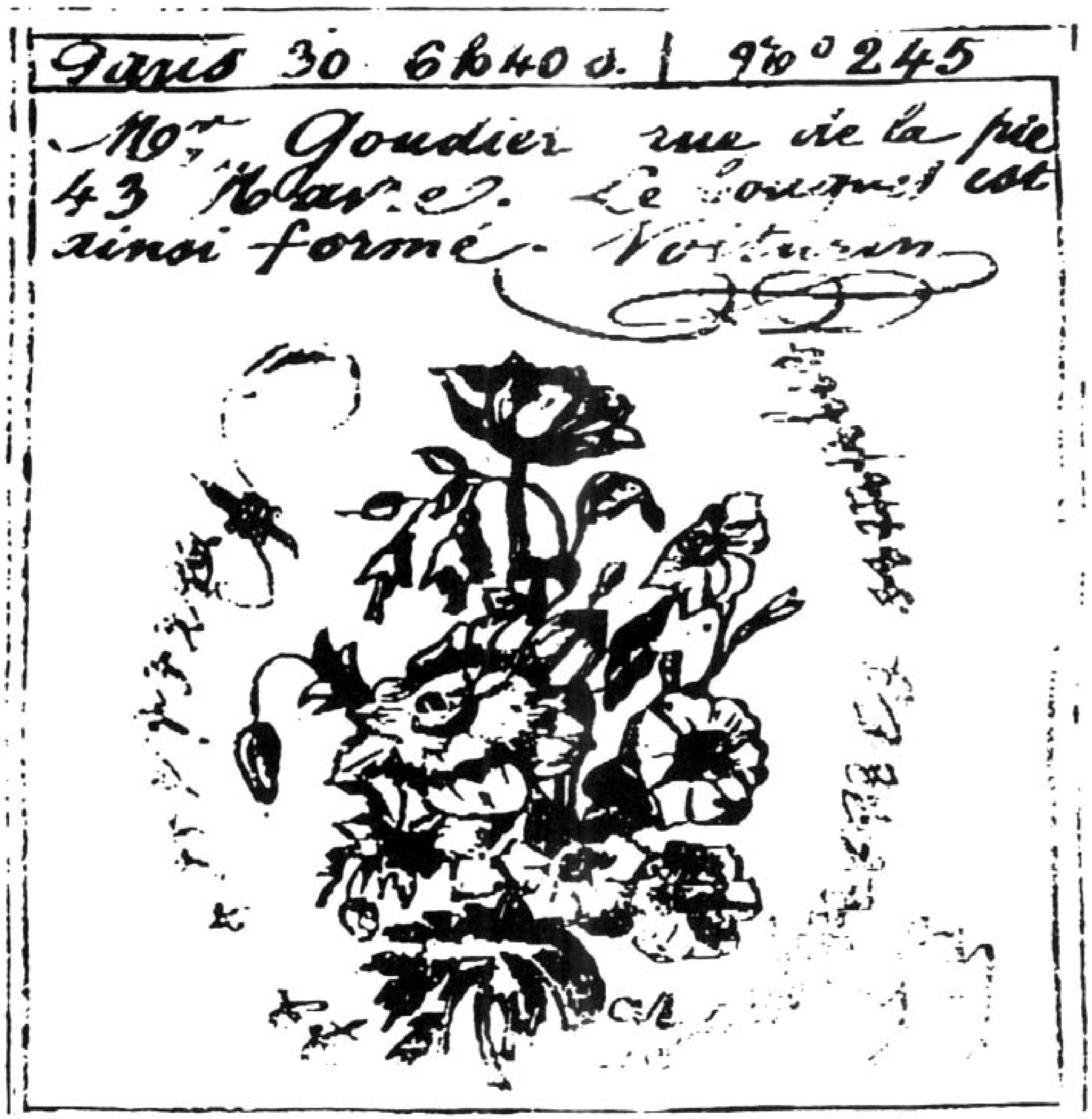


Figure 1.1.2 Bakewell's rotating cylinder.



Paris 30 6h40 s. | N° 245

Mr Goudier rue de la pie 43 Havre. Le bouquet est ainsi formé. [illegible]

Figure 1.1.3 Received copy — Pantelegraph, 1861.

Facsimile technical progress

- 1843: Bain’s Pendulum type
- 1844: Telegraph (Morse)   Entropy coding !
- 1850: Rotating drum (England)
- 1865: 1st commercial **FAX** (Caselli)
- 1876: **Telephone** (Bell)
- 1902: Optical scan
- 1917: Teletype, AT&T
- - 1968: many proprietary machines (AT&T, RCA, Artzt, Teledeltos, Xerox, ...)

Facsimile technical progress (Cont’d)

- 1968: CCITT Group 1 Rec.
- 1976: CCITT Group 2 Rec.
- 1976 - 1980: CCITT Group 3 Rec.
- 1984: Group 4 Rec.
- 1988: Error free G.3
- 1991: ISO/IEC CD 11544 **JBIG**

- **2003:**
  - **56Kbps Fax/Modem**
  - **1.5Mbps ADSL/Cable**

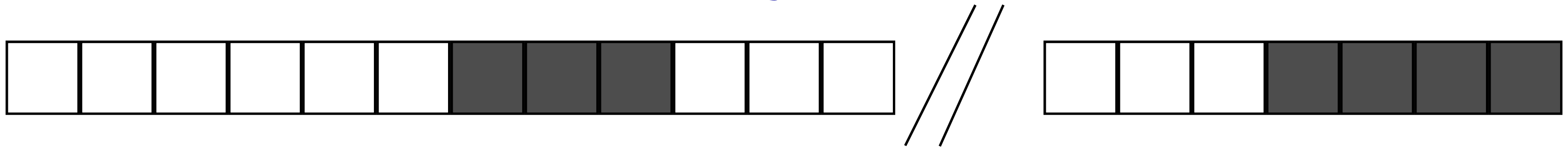
CCITT Group 1,2 (1968)

Standardization for machines outside N. America:

Parameter	G.1	AM.6M	G.2
Lines/Min.	180	180	360
Modulation	FM	FM	VSB AM/PM
Carrier Freq.			2100Hz ± 10Hz
White signal	1300Hz	1500Hz	Max carrier
Black signal	2100Hz	2400Hz	26dB max. lower

Digital FAX

- Run-Length (RL) developed by D.Weber, removes redundancy from data:



New Line

100 white dots

RL Coding: W6,3,100,4,...

First (non-standard) machine: DACOM (1974), with 4800bps modem over PTS (10 times slower than Telex…)

CCITT Group 3 - T.4 recommendation (1980-88)

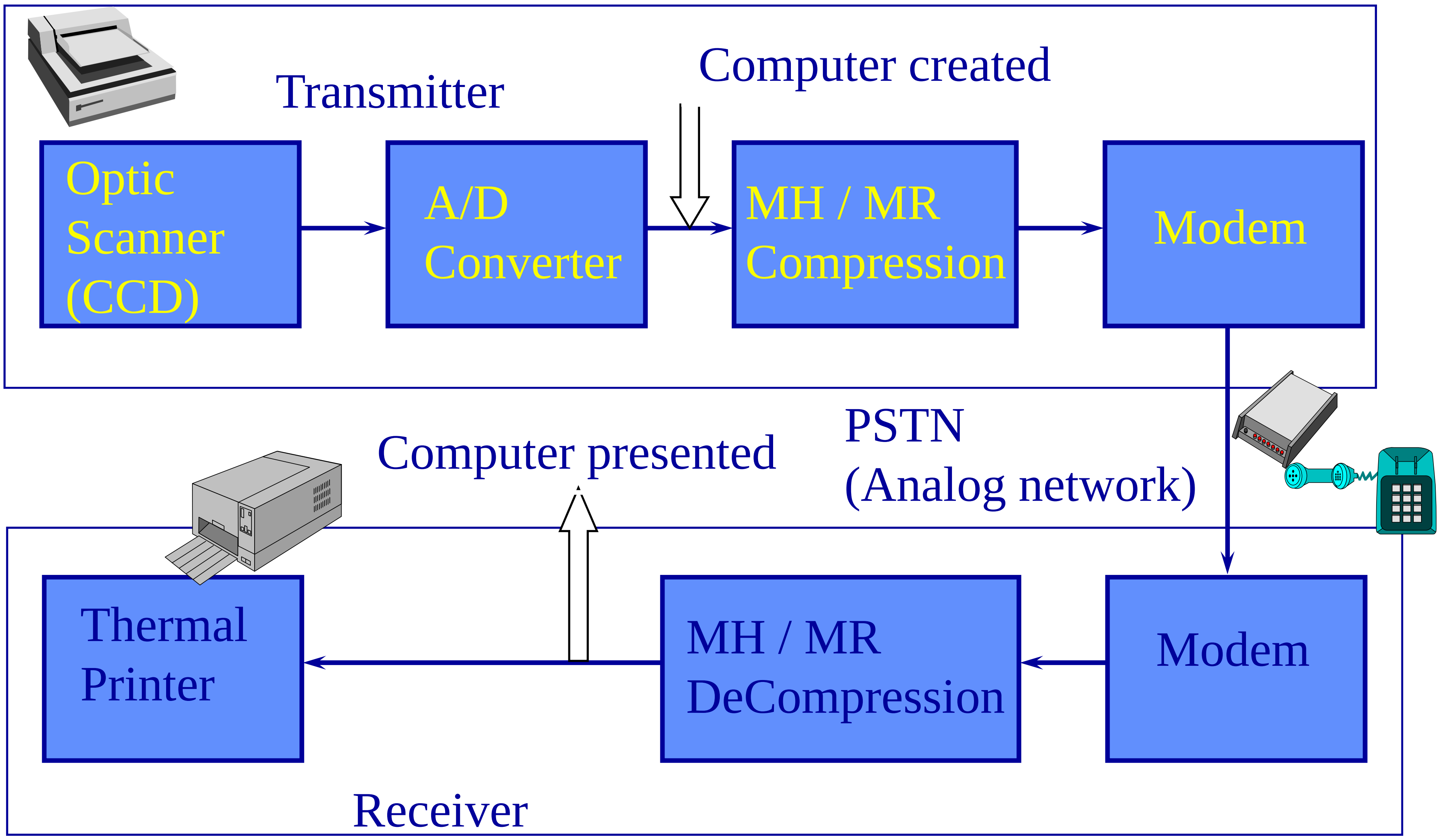
- V.29 Modem, 9600bps

	Standard	Optional	
Scan Direction:	Left to Right, top to Bottom		
Scan Width (mm)	215	255	303
Pels per Line	1728	2048	2432
Horiz. Pel	8mm(203”)		
Vert. Pel	3.85(97.8)	7.7(195.6)	
Coding	Modified H.	Modified Read	
Data Modem	v.27(ter)	v.29	
bps	4800/2400	9600/7200	

CCITT G.3 (1980-88) (Cont'd)

Signaling Modem	v.21	v.27(ter)
bps	300	2400
mSec./Line	20	0,5,10,40
Starting pel	Always White	
# of Elements/Code Word	0 - 63 / 64-1728	
End-of-Line Code (EOL)	000000000001	
End-of-Page Code	6 x EOL	

G.3 Architecture



Group 3 & Group 4

GROUP 3

•MODIFIED HUFFMAN METHOD (MHM) – Unidimensional coding method based on the coding of the lenght of alternate black and white pixel runs using Huffman coding.

GROUP 4 (Also Group 3 Options)

•MODIFIED READ METHOD (MRM) – Bidimensional coding method based on the coding of the variations of the positions of tone transition pixels (black-white or white-black) in relation to the previous line; unidimensional coding may be used every k lines.

•MODIFIED-MODIFIED READ METHOD (MMRM) – Similar to MRM but without periodic unidimensional coding.

G.3 Compression

- 5% - 20% of source (up to 95% comp.)
- Modified Huffman Coding:
  - Assume long *White Runs* between black pixels
  - *White Runs* can long complete line:

    (9 bits coding, meaning 192:1 compression)
  - 92 different codes: 28 groups of pX64 pixels, and 64 *short-runs* of 0-63 pixels
  - 13 more codes for long runs (1792 - 2560)

Huffman Codes

White run length	Code word	Black run length	Code word
0	00110101	0	0000110111
1	000111	1	010
2	0111	2	11
3	1000	3	10
4	1011	4	011
5	1100	5	0011
6	1110	6	0010
7	1111	7	00011
8	10011	8	000101
9	10100	9	000100
10	00111	10	0000100
11	01000	11	0000101
12	001000	12	0000111
13	000011	13	00000100
14	110100	14	00000111
15	110101	15	000011000
16	101010	16	0000010111
17	101011	17	0000011000
18	0100111	18	0000001000
19	0001100	19	00001100111
20	0001000	20	00001101000
21	0010111	21	00001101100
22	0000011	22	00000110111
23	0000100	23	00000101000
24	0101000	24	00000010111
25	0101011	25	00000011000
26	0010011	26	000011001010
27	0100100	27	000011001011
28	0011000	28	000011001100
29	00000010	29	000011001101
30	00000011	30	000001101000
31	00011010	31	000001101001

Huffman Codes Cont’d

White run length	Code word	Black run length	Code word
32	00011011	32	000001101010
33	00010010	33	000001101011
34	00010011	34	000011010010
35	00010100	35	000011010011
36	00010101	36	000011010100
37	00010110	37	000011010101
38	00010111	38	000011010110
39	00101000	39	000011010111
40	00101001	40	000001101100
41	00101010	41	000001101101
42	00101011	42	000011011010
43	00101100	43	000011011011
44	00101101	44	000001010100
45	00000100	45	000001010101
46	00000101	46	000001010110
47	00001010	47	000001010111
48	00001011	48	000001100100
49	01010010	49	000001100101
50	01010011	50	000001010010
51	01010100	51	000001010011
52	01010101	52	000000100100
53	00100100	53	000000110111
54	00100101	54	000000111000
55	01011000	55	000000100111
56	01011001	56	000000101000
57	01011010	57	000001011000
58	01011011	58	000001011001
59	01001010	59	000000101011
60	01001011	60	000000101100
61	00110010	61	000001011010
62	00110011	62	000001100110
63	00110100	63	000001100111

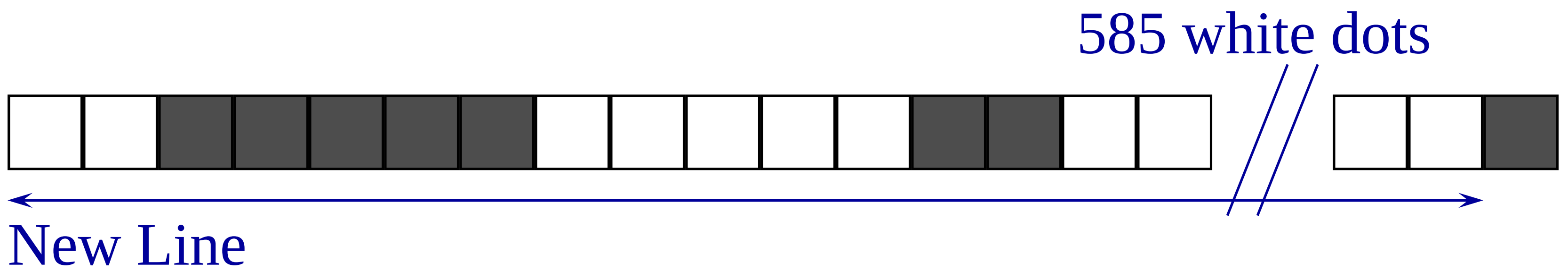
Make-up codes between 64 and 1728

hite run length	Code word	Black run length	Code word
64	11011	64	0000001111
128	10010	128	000011001000
192	010111	192	000011001001
256	0110111	256	000001011011
320	00110110	320	000000110011
384	00110111	384	000000110100
448	01100100	448	000000110101
512	01100101	512	0000001101100
576	01101000	576	0000001101101
640	01100111	640	0000001001010
704	011001100	704	0000001001011
768	011001101	768	0000001001100
832	011010010	832	0000001001101
896	011010011	896	0000001110010
960	011010100	960	0000001110011
1024	011010101	1024	0000001110100
1088	011010110	1088	0000001110101
1152	011010111	1152	0000001110110
1216	011011000	1216	0000001110111
1280	011011001	1280	0000001010010
1344	011011010	1344	0000001010011
1408	011011011	1408	0000001010100
1472	010011000	1472	0000001010101
1536	010011001	1536	0000001011010
1600	010011010	1600	0000001011011
1664	011000	1664	0000001100100
1728	010011011	1728	0000001100101

Make-up codes between 1792 and 2560

Run length (black and white)	Make-up codes
1792	00000001000
1856	00000001100
1920	00000001101
1984	000000010010
2048	000000010011
2112	000000010100
2176	000000010101
2240	000000010110
2304	000000010111
2368	000000011100
2432	000000011101
2496	000000011110
2560	000000011111

Modified Huffman example



RL: 2W 5B 5W 2B 585W

MH: 0111 0011 1100 11 01101000 10100

- Total pixel count: 599, MH: 27 bits,
- Compression Ratio: 599/27=22.2 (~4.5%)

Modified READ Coding

- READ: Relative Element Address Designate: exploits the correlation between successive lines
- *Element*: A group of pixels of same color
- *Changing Element*: An *element* with different color from previous *element*
- The position of every *changing element* is coded relatively to a *reference element* in the current line or the *reference line* (above)

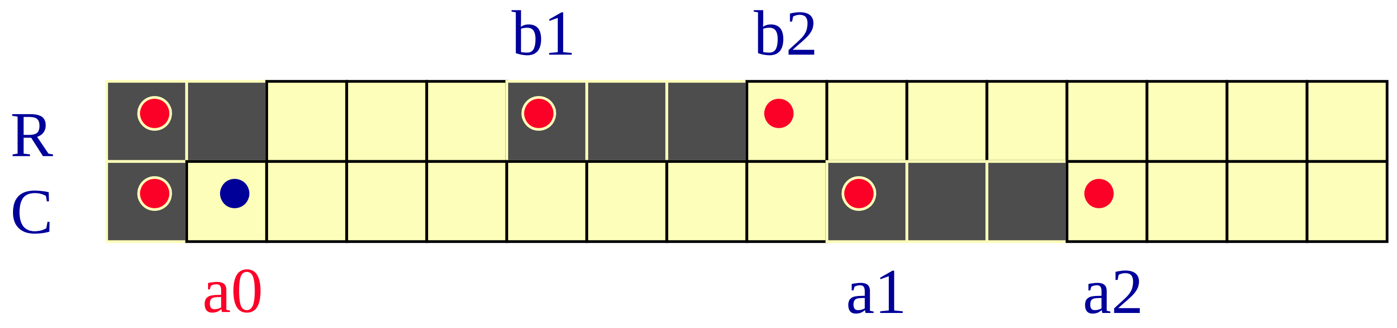
G.4 is a simplified version of G.3 in which only 2D coding is allowed

READ Coding example (Cont'd)

- If a couple of matching elements are positioned less than 3 pixels horiz. distance we code them in *vertical mode*
- If more than 3 pixels distance:
  - *Pass Mode* if the change in *reference line*
  - *Horizontal mode* if the change in *current line*

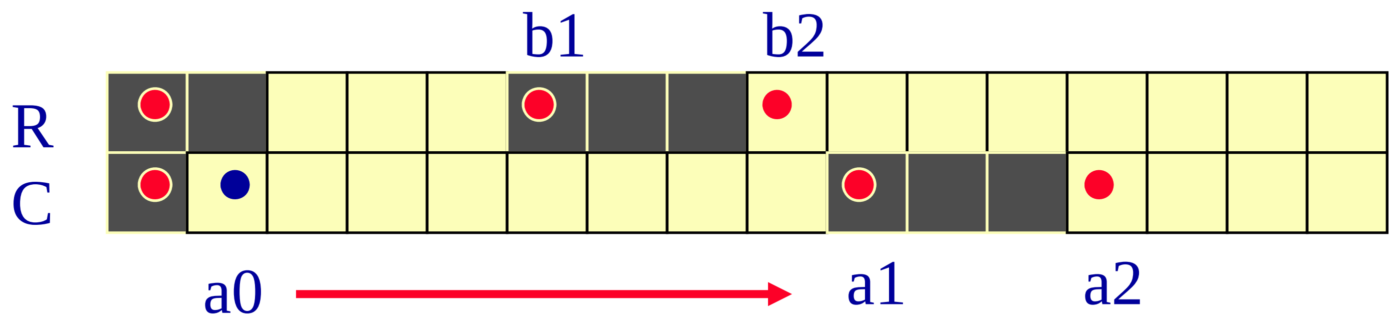
  use horizontal mode for next two changes in current line and back to vertical mode (if possible)

READ: details



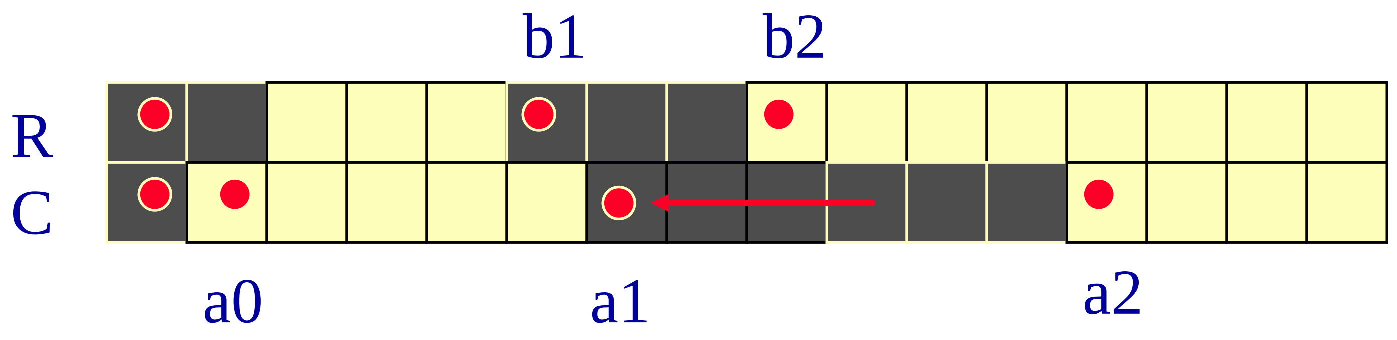
- <u>Coding line:</u>
  - a0: last pixel known to both encoder and decoder
  - a1: 1st transition right to a0 (known to encoder only)
  - a2: 2nd transition right to a0 (known to encoder only)
- <u>Reference line</u>
  - b1: 1st transition right to a0 location (opposite color)
  - b2: 1st transition right to b1

READ example 1 (Cont'd)



- b1 and b2 are between a0 and a1: this is a 'Pass Mode': the decoder knows that all the pixels to the right of a0 until below b2 are same color.
- So, the last known is changing to a0 and a new b1, b2 should be defined

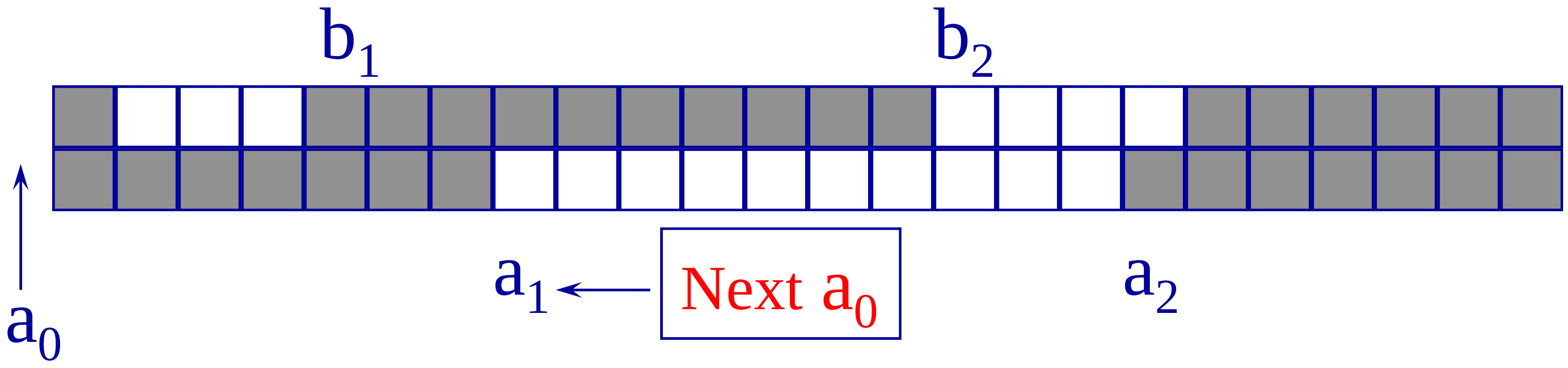
READ example 2



- In this case it can’t be Pass Mode.
- since the distance between a1 and below b1 is no more than 3 (1 in this case) - it is a ‘Vertical Mode’: a1 location is encoded relative to b1, <u>and a1 becomes a0</u>.
- If the distance is more than 3 pixels we change to ‘Horizontal Mode’: the distances (a0,a1) and (a1,a2) are encoded using <u>Modified Huffman (MH)</u>

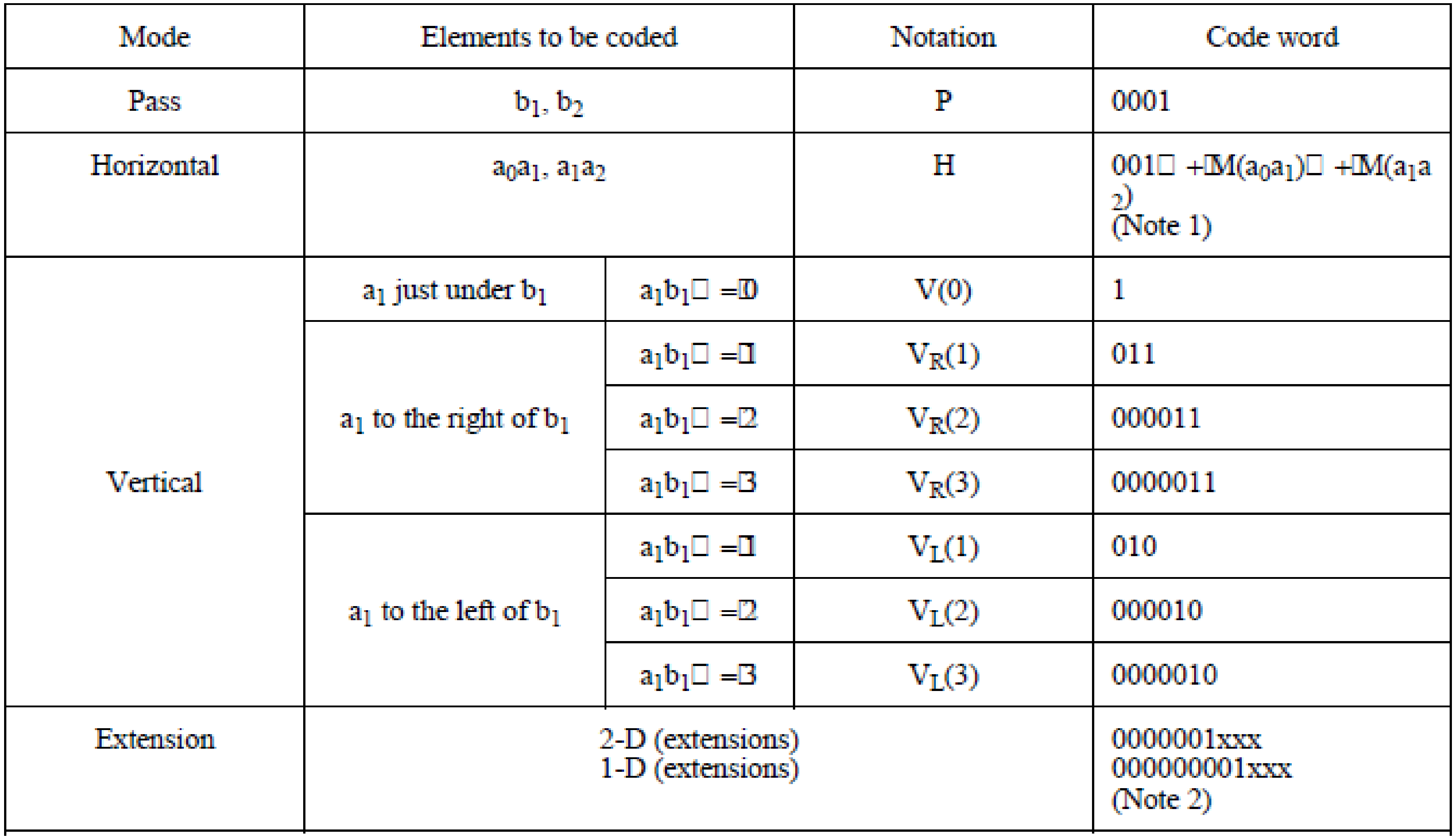
Another Coding Example

- a_0: Reference element
- a_1, a_2, b_1, b_2,: Changing elements
- b_2 is to the right of a_1, and the distance between a_1 and b_1 is equal to 3 : its a vertical mode, and a new a_0 assigned:



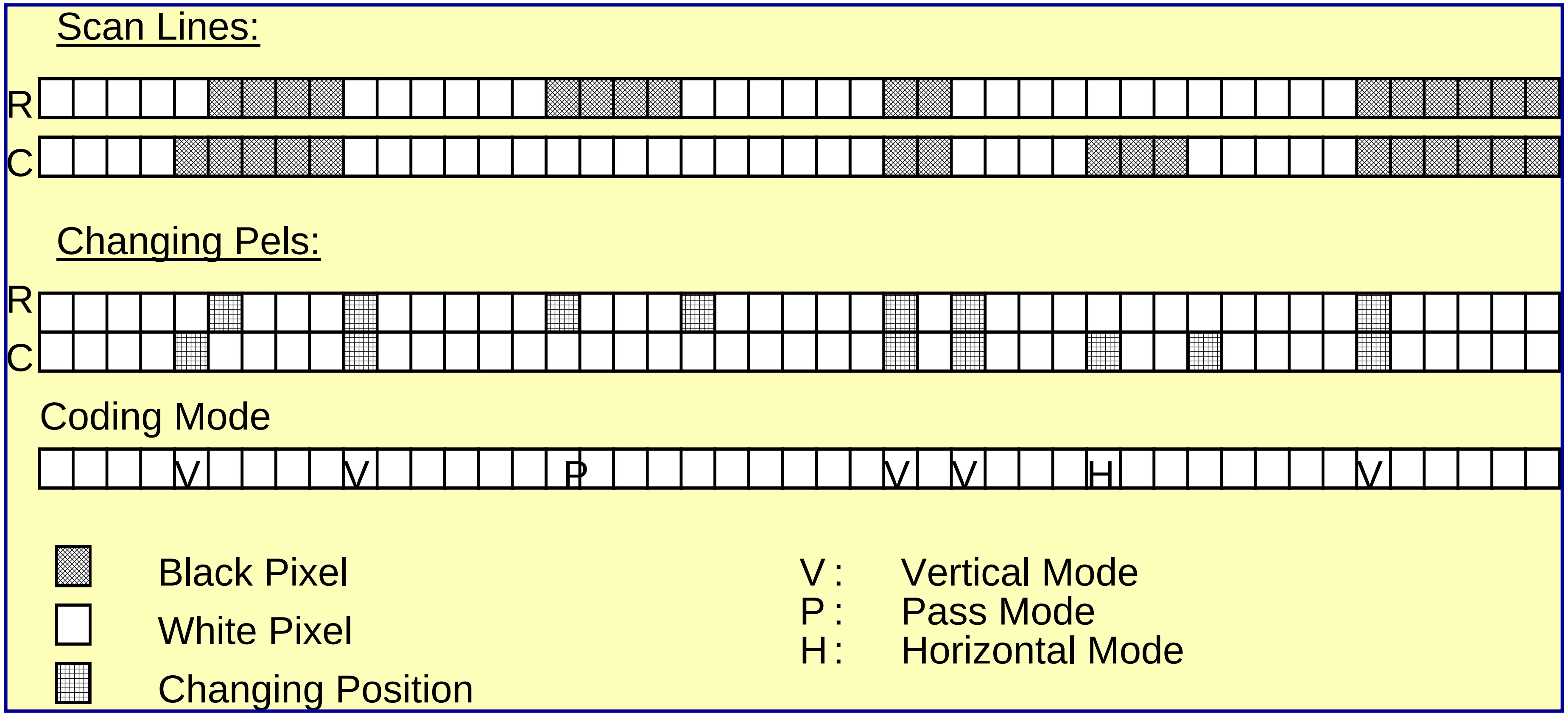
A Coding Example (Cont'd)

- Once the compression mode is determined, a corresponding code can be formed:



Mode	Elements to be coded		Notation	Code word
Pass	b_1, b_2		P	0001
Horizontal	a_0a_1, a_1a_2		H	001□ +□M(a_0a_1)□ +□M(a_1a_2) (Note 1)
Vertical	a_1 just under b_1	a_1b_1□ =□0	V(0)	1
	a_1 to the right of b_1	a_1b_1□ =□1	$V_R(1)$	011
		a_1b_1□ =□2	$V_R(2)$	000011
		a_1b_1□ =□3	$V_R(3)$	0000011
	a_1 to the left of b_1	a_1b_1□ =□1	$V_L(1)$	010
		a_1b_1□ =□2	$V_L(2)$	000010
		a_1b_1□ =□3	$V_L(3)$	0000010
Extension	2-D (extensions) 1-D (extensions)			0000001xxx 000000001xxx (Note 2)

A more complicated example



K-Factor

- When distorted by noise pulse, errors are made in the received copy.
- To prevent from propagating down the page, a MH coded line is sent periodically.
- After one line is coded in 1-dimensional mode, K-1 lines will be coded in 2-D mode: At normal resolution every second line (K=2), and at fine resolution K=4 (3 lines)

G.3 Enhancements (option)

- Error Concealment
- Error Control
- Dither Coding
- RS-232 Interface
- High Resolution
- Small Page Size (A5, A6)
- Non-Standard Operation

Group 4: CCITT T.6

- Identical to T.4 with a slight difference: only 2-D mode is allowed (Called MMR).

A Comparison of binary image coding (after [3]):

Source image*	MH **	MR	MMR	JBIG
letter	20,605 B	31%	59%	68%
Sparse text	26155 B	37%	62%	71%
Dense text	133,705 B	23%	33%	48%

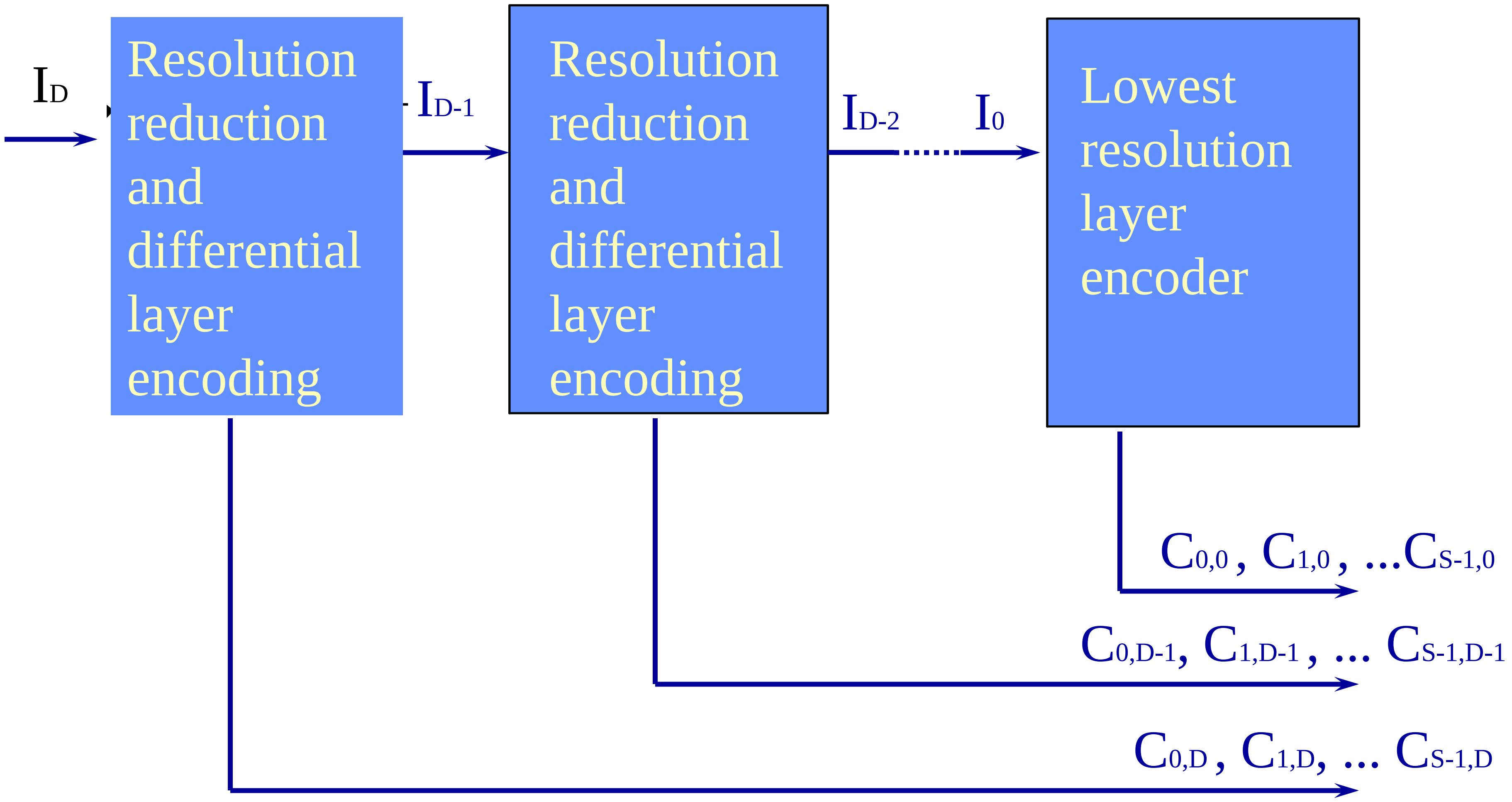
* Source images are of size: 4352x3072 dots (1 bit pixels)

** MH results in bytes, all other results are better than MH in x%

JBIG

- Lossless Compression.
- Progressive Coding.
- Sequential Coding.
- Arithmetic Encoder/Decoder.
- Resolution Reduction Algorithm
-     (optional, can be replaced).

Encoder Scheme



I_D
Resolution reduction and differential layer encoding
I_{D-1}
Resolution reduction and differential layer encoding
I_{D-2}
I_0
Lowest resolution layer encoder
$C_{0,0}, C_{1,0}, \ldots C_{S-1,0}$
$C_{0,D-1}, C_{1,D-1}, \ldots C_{S-1,D-1}$
$C_{0,D}, C_{1,D}, \ldots C_{S-1,D}$

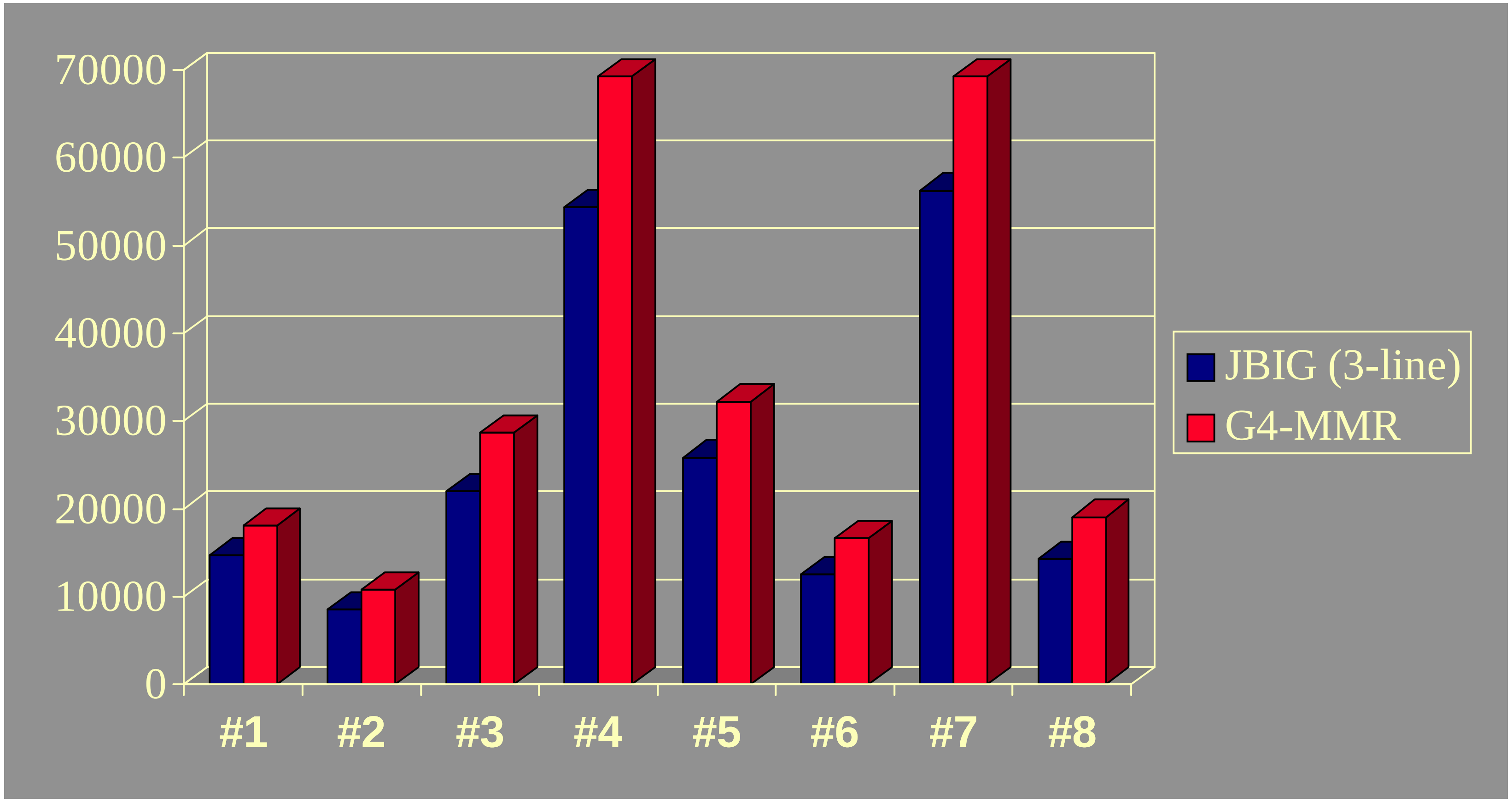
Resolution Reduction Technique

- Creates low resolution images.
- Combines decimation and filtering in one action.
- Uses 9 high resolution and 3 low resolution pixels to determine color of target pixel.
- Preserves gray-levels achieved with halftoning.

JBIG Pro’s and Con’s

- ☺ Progressive (for binary images).
- ☺ Better compression for images with up to
- ☺ 6 bit/pixel (Vs. JPEG)
- ☺ Better compression than G3 and G4.
- ☹ Slow and complicated (Vs. JPEG, G.3/4)
- ☹ Consumes memory resources and needs frame buffers.

Compression Comparison (in bytes)



CCITT FAX reference images (#1 - #8)

Original size: 513216 bytes

FAX] reference images

Compression Comparison (Cont’d)

Typical results for:

- Scanned text and line drawings:

JBIG ~20-25% better than G4

- Computer generated line-drawing:

JBIG ~75% better than G4

- Scanned dither halftones images:

JBIG ~85%-90% better than G4

References:

- McConnel, Bodson and Schaphorst, *FAX: Digital Facsimile Technology and Applications*, Artech 1989
- K, Sayood, *Introduction to Data Compression*
- R.Arps and T.Truong, *Comparison of International Standards for Lossless Image Compression*. Proc. Of IEEE, 82:889-899, June 1994
- ITU-T (Former CCITT) Blue Book T.0-T.63, 1989, Recommendations T.4, T.6