VERİ SIKIŞTIRMA

İstatistiksel Yöntemler-5

Prof.Dr. Banu DİRİ

MNP5 – Microcom Network Protocol

- Microcom tarafından modemler için geliştirilmiş bir sıkıştırma yöntemidir.
- MNP5 ve MNP7 diye iki versiyonu mevcuttur. En çok tercih edilen MNP5'dir.
- İki aşamalı bir yöntemdir. Run Length Encoding – Adaptive Frequency Encoding
- RLE (tekrar eden üç ve üçten fazla sembol varsa üçü arka arkaya yazılıp, dördüncü byte tekrar sayısı yazılır)

Byte	Freq.	Token	Byte	Freq.	Token	Byte	Freq.	Token	Byte	Freq.	Token
0	0	000|0	9	0	011|001	26	0	111|1010	247	0	111|1110111
1	0	000|1	10	0	011|010	27	0	111|1011	248	0	111|1111000
2	0	001|0	11	0	011|011	28	0	111|1100	249	0	111|1111001
3	0	001|1	12	0	011|100	29	0	111|1101	250	0	111|1111010
4	0	010|00	13	0	011|101	30	0	111|1110	251	0	111|1111011
5	0	010|01	14	0	011|110	31	0	111|1111	252	0	111|1111100
6	0	010|10	15	0	011|111	32	0	101|00000	253	0	111|1111101
7	0	010|11	16	0	111|0000	33	0	101|00001	254	0	111|1111110
8	0	011|000	17	0	111|0001	34	0	101|00010	255	0	111|11111110

18 to 25 and 35 to 246 continue in the same pattern.

İkinci aşama, birinci aşamanın sonuçlarını kullanır ve Huffman kodunun bir versiyonu olarak anlattığımız yönteme benzer.

- 256 satır ve 2 sütunluk bir tablo ile başlar.
- Her bir satır 8 bitlik 0-255 arası sayıya karşılık gelir.
- Birinci kolon frekans değeridir ve başlangıç değeri 0'dır.
- İkinci kolon «token» adı verilen değişken uzunluktaki koddan oluşur.
- Her bir «token» 3 bitlik header (başlık) bilgisi ile başlar ve bunları kod bilgisi izler.
- Üç farklı durum dışında, 2 tane 1 bitlik (0-1), 4 tane 2 bitlik (0-3), 8 tane 3 bitlik (0-7), 16 tane 4 bitlik (0-15), 32 tane 5 bitlik (0-31), 64 tane 6 bitlik (0-63) ve 128 tane 7 bitlik (0-127) adet koddan oluşur.

Byte	Freq.	Token	Byte	Freq.	Token	Byte	Freq.	Token	Byte	Freq.	Token
0	0	000|0	9	0	011|001	26	0	111|1010	247	0	111|1110111
1	0	000|1	10	0	011|010	27	0	111|1011	248	0	111|1111000
2	0	001|0	11	0	011|011	28	0	111|1100	249	0	111|1111001
3	0	001|1	12	0	011|100	29	0	111|1101	250	0	111|1111010
4	0	010|00	13	0	011|101	30	0	111|1110	251	0	111|1111011
5	0	010|01	14	0	011|110	31	0	111|1111	252	0	111|1111100
6	0	010|10	15	0	011|111	32	0	101|00000	253	0	111|1111101
7	0	010|11	16	0	111|0000	33	0	101|00001	254	0	111|1111110
8	0	011|000	17	0	111|0001	34	0	101|00010	255	0	111|11111110

18 to 25 and 35 to 246 continue in the same pattern.

- Bahsedilen 3 farklı kodun, ilk iki kodu (000|0, 000|1) ve son kodu da (111|11111110) dır.

İkinci adım;

- Tablodaki frekans kolonu 0 ile başlar. Input stream'den okunan ilk byte karşılık gelen «token» tablodan çıkış stream yazılır ve frekans değeri bir artırılır.
- Frekans değeri büyük olan byte kendi üzerinde yer alan daha kısa token ile gösterilebilmesi için token'ların yeri değiştirilir (sadece token'lar).
- Frekans değeri 8 bitlik bir alanda tutulur ve her değişiklikte maksimum değere ulaşılıp ulaşılmadığı kontrol edilir. Eğer ulaşılmış ise tüm sıklık değerleri ikiye bölünür.

Örnek

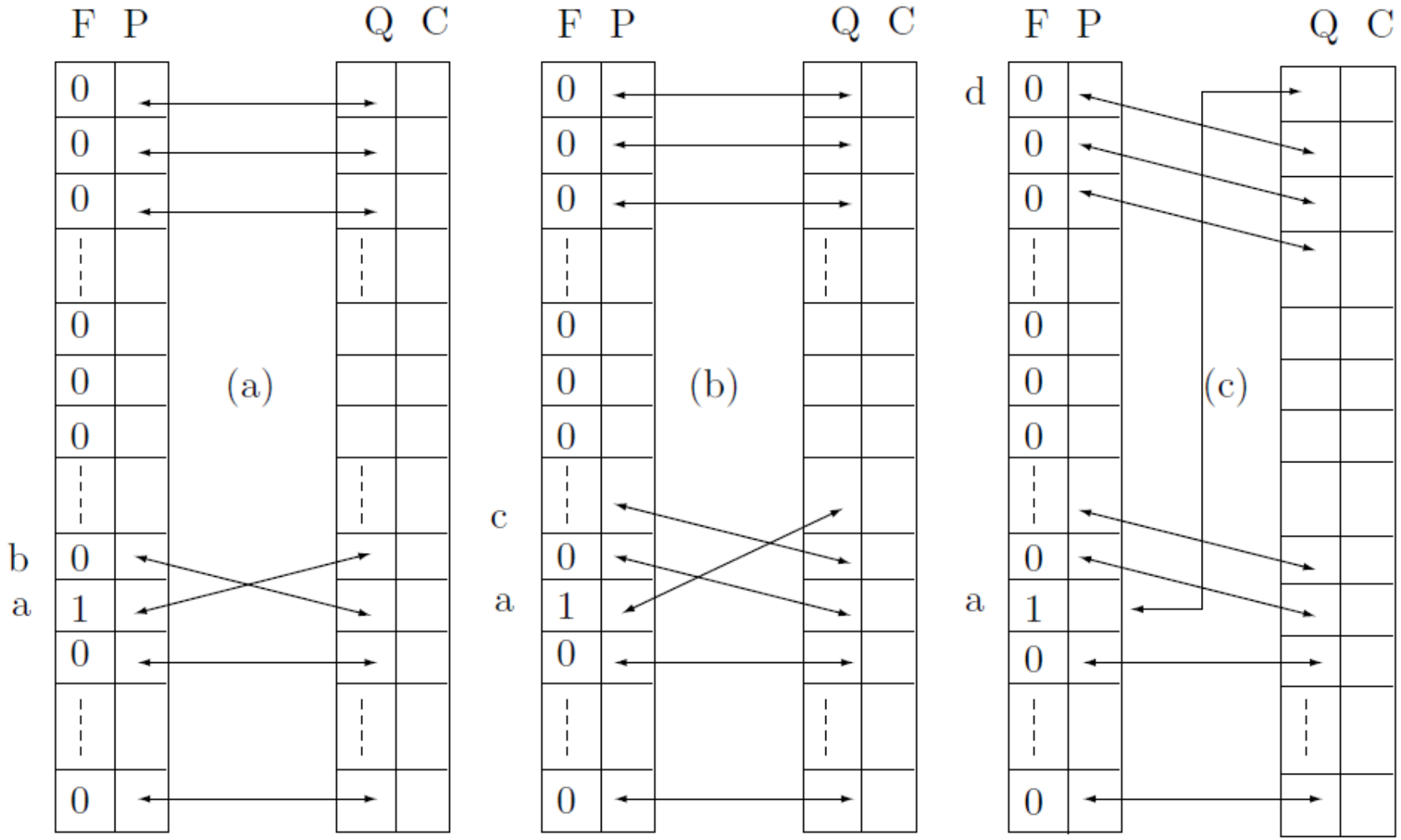
«52 52 52 6»

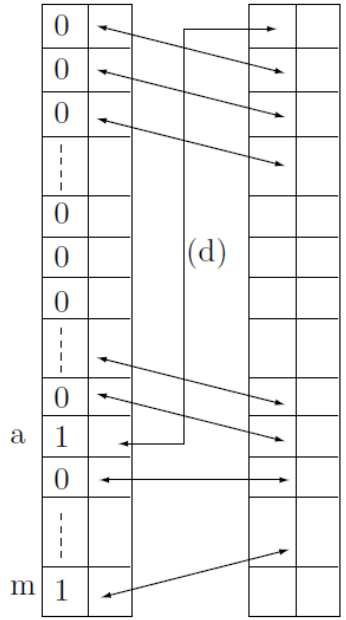
- Tabloda 53.sıradaki token değeri stream'e yazılır
- Her byte okunduktan sonra tabloda yer değiştirme olabilir ve 52 değeri farklı token değerlerine sahip olabilir.
- 6 değeri için de, tabloda 7.sırada yer alan token değeri yazılır, fakat frekans değeri arttırılmaz.

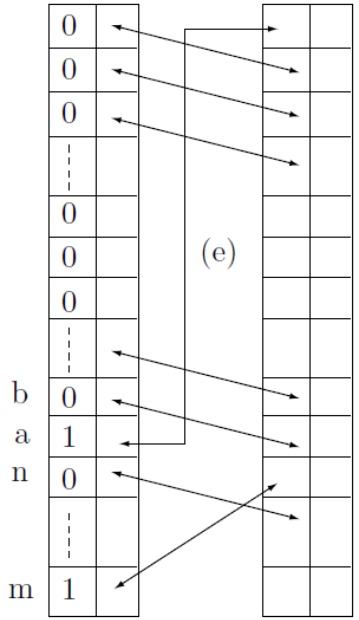
- Çıkış değerinde yer alan «token» lar 4 ile 11 bit arasında değer alabilir. Bu çıktılar 8 bitlik paketler halinde çıkış stream yazılır.
- En sona 11 tane birden oluşan «flush token» adı verilen kod eklenir (gerekirse 8'e tamamlamak için 1 eklenir) ve output stream sonlandırılır.
- MNP5'in iyi sonuç vermesi tamamiyle orijinal verinin içeriğine bağlıdır.

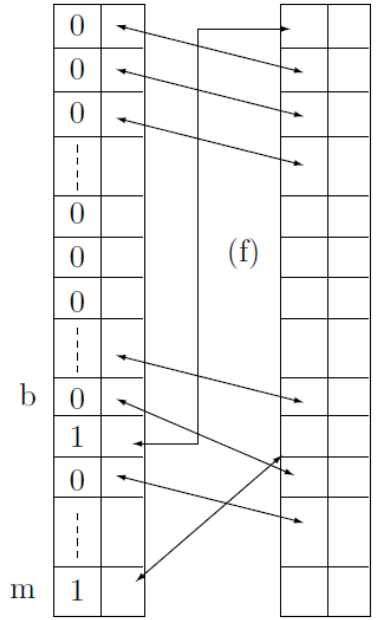
Tablonun güncellenmesi iki şekilde yapılır.

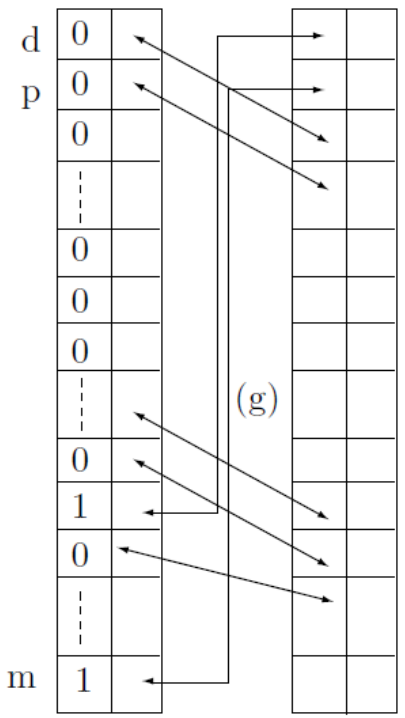
1. Her sıklık değeri arttırıldığında, bütün tablo yeniden sıralanır
2. Tablo içerisinde pointer kullanılarak sıralama gerçekleştirilir

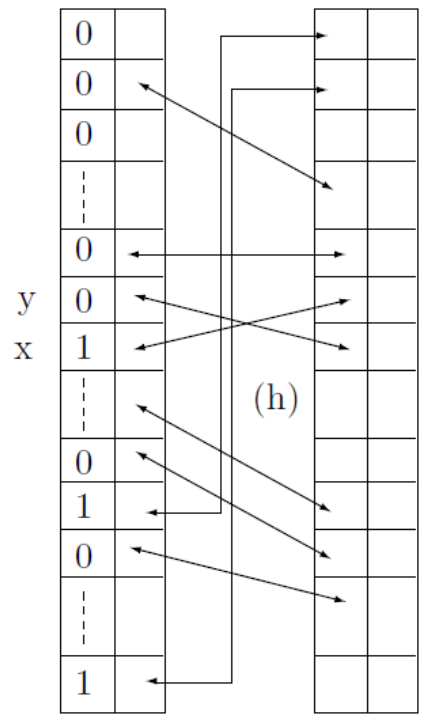


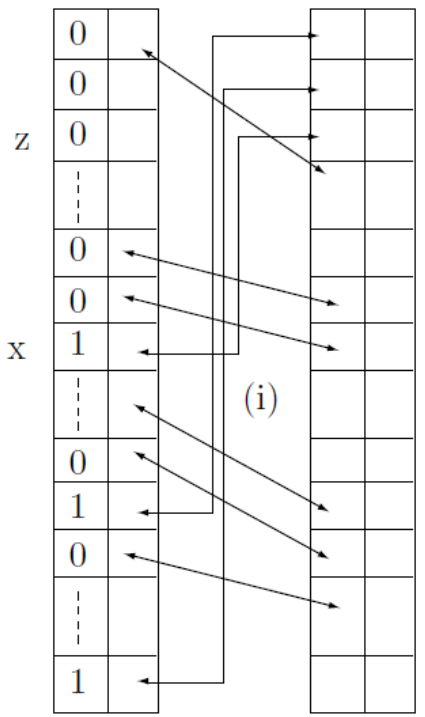

0
0
0
0
0
0
0
a 1
0
m 1
(d)


0
0
0
0
0
0
b 0
a 1
n 0
m 1
(e)


0
0
0
0
0
0
b 0
1
0
m 1
(f)


d 0
p 0
0
0
0
0
0
0
1
0
m 1
(g)


0
0
0
0
y 0
x 1
0
1
0
1
(h)


0
0
z 0
0
0
x 1
0
1
0
1
(i)

MNP7 – Microcom Network Protocol

- MNP5'den daha karışık bir algoritmaya sahiptir.
- İki boyutu dinamik Huffman kodlaması ile RLE'nin birleşimidir.
- İlk bölüm RLE ile başlar. (3 3 3 0 → 3 tane 3; 6 6 6 7 → 10 tane 6)
- İkinci bölüm iki boyutlu Huffman kodlamadır
- MNP7, first-order Markov modeline dayanır. Her sembol kendisinin önünde yer alan sembole göre işlenir.

	Current character						
	0	1	2	...	254	255	
	0 0	0 0	0 0	...	0 0	0 0	
	1 0	1 0	1 0	...	1 0	1 0	
	2 0	2 0	2 0	...	2 0	2 0	...a b c d e...
	3 0	3 0	3 0	...	3 0	3 0	t l h o d
Preced.	⋮	⋮	⋮		⋮	⋮	h e o a r
Char.	254 0	254 0	254 0	...	254 0	254 0	c u r e s
	255 0	255 0	255 0	...	255 0	255 0	⋮ ⋮ ⋮ ⋮ ⋮

- 256 adet, 256 x 2 boyutunda tablodan oluşur.
- Her satır 0 ile 255 arasındaki karakteri temsil eder
- 256x2 boyutundaki her tablonun da ilk sütunu 0-255 arasındaki sayılardan ve ikinci sütunu da kullanım sıklıklarından oluşur ki başlangıç değeri 0'dır.
- Kodlar ayrı bir tabloda tutulur ve asla yer değiştirme yapılmaz.

Facsimile Compression – Faks Sıkıştırma

İletişim hattı üzerinden görüntü verisi gönderileceği zaman sıkıştırma önemli hale gelir
Faks makineleri ile görüntünün cihazlar arasında gönderilmesi gündeme gelmiştir

ITU-T (International Telecomunications Union) tarafından bir veri sıkıştırma standardı geliştirildi
T2 (Grup 1) T3 (Grup 2)

Daha sonra bu standartların yerini **T4 (Grup 3) ve T6 (Grup4)** aldı

- Grup 3 (T4) PSTN (Public Switched Telephone Network – 9600 baud)
- Grup 4 (T6) ISDN (Integrated Services Digital Network – 64K baud)

- Her iki yöntemde de sıkıştırma oranı 10:1 veya daha yüksektir
- Önceden bir sayfanın iletilme hızı ~1dak. sürüyor iken, kullanılan bu standartlar ile birkaç saniyeye düşmüştür.

- Faks makineleri, dokümanı satır satır tarar ve satırları «pels» (picture elements) adı verilen küçük siyah beyaz noktalara çevirir

Yatay çözünürlük

- Her zaman 8,05 pels/mm (205 pels / inch) (1" = 25,4 mm)
- 8,5" genişliğinde taranan her satır 1.738 pels çevrilir
- T4 standardında tarama için önerilen satır genişliği 8,2"

Dikey çözünürlük

- Standard mode → 3,85 scanline/mm
- Fine mode → 7,7 scanline /mm
- Very fine Mode → 15,4 scanline/mm

10" yüksekliğinde olan bir sayfada toplam pels sayısı sıkıştırma yapılmak ise 3 farklı moddaki gönderim süresi ne olur

	Scan lines	Pels per line	Pels per page	Time (sec.)	Time (min.)
254mm * 3,85mm → standart mode	978	1664	1.670M	170	2.82
254mm * 7,7mm → fine mode	1956	1664	3.255M	339	5.65
254mm * 15,4 mm → very fine mode	3912	1664	6.510M	678	11.3

Sürelerdeki uzunluk, faks iletişimde veri sıkıştırmanın ne kadar önemli olduğunu göstermektedir.

- Grup 3 kodu, birçok doküman üzerinde siyah ve beyaz pels'lerin run-length lerinin analiz edilmesi ile elde edilir
- Her bir run uzunluğu için Huffman algoritması ile değişken uzunlukta kodlar atanır
- En fazla kullanılan run uzunluklarının 2, 3 ve 4 siyah pels değerleri olduğu tespit edilmiş ve bu pels değerleri için kısa kodlar atanmıştır
- 2 ve 7 arasındaki run değerlerine sahip olan beyaz pels'lere de daha uzun kodlar atanmıştır
- Grup 3, RLE ve Huffman kodlamanın bir bileşimidir
- Run değerlerinin uzun olabileceği düşünüldüğünden Huffman algoritması üzerinde değişiklikler yapılmıştır. Bu sebeple Grup 3 standartı **MH-Modified Huffman** olarak adlandırılır
- 1 ile 63 run uzunluğundaki pels değerleri için **Termination kodlar**, bu pels değerlerinin 64 katı değeri içinde **Make-up** kodlar kullanılır

Run length	White code-word	Black code-word	Run length	White code-word	Black code-word
0	00110101	0000110111	32	00011011	000001101010
1	000111	010	33	00010010	000001101011
2	0111	11	34	00010011	000011010010
3	1000	10	35	00010100	000011010011
4	1011	011	36	00010101	000011010100
5	1100	0011	37	00010110	000011010101
6	1110	0010	38	00010111	000011010110
7	1111	00011	39	00101000	000011010111
8	10011	000101	40	00101001	000001101100
9	10100	000100	41	00101010	000001101101
10	00111	0000100	42	00101011	000011011010
11	01000	0000101	43	00101100	000011011011
12	001000	0000111	44	00101101	000001010100
13	000011	00000100	45	00000100	000001010101
14	110100	00000111	46	00000101	000001010110
15	110101	000011000	47	00001010	000001010111
16	101010	0000010111	48	00001011	000001100100
17	101011	0000011000	49	01010010	000001100101
18	0100111	0000001000	50	01010011	000001010010
19	0001100	00001100111	51	01010100	000001010011
20	0001000	00001101000	52	01010101	000000100100
21	0010111	00001101100	53	00100100	000000110111
22	0000011	00000110111	54	00100101	000000111000
23	0000100	00000101000	55	01011000	000000100111
24	0101000	00000010111	56	01011001	000000101000
25	0101011	00000011000	57	01011010	000001011000
26	0010011	000011001010	58	01011011	000001011001
27	0100100	000011001011	59	01001010	000000101011
28	0011000	000011001100	60	01001011	000000101100
29	00000010	000011001101	61	00110010	000001011010
30	00000011	000001101000	62	00110011	000001100110
31	00011010	000001101001	63	00110100	000001100111

Termination Codes

Make-Up Codes

Run length	White code-word	Black code-word	Run length	White code-word	Black code-word
64	11011	0000001111	1344	011011010	0000001010011
128	10010	000011001000	1408	011011011	0000001010100
192	010111	000011001001	1472	010011000	0000001010101
256	0110111	000001011011	1536	010011001	0000001011010
320	00110110	000000110011	1600	010011010	0000001011011
384	00110111	000000110100	1664	011000	0000001100100
448	01100100	000000110101	1728	010011011	0000001100101
512	01100101	0000001101100	1792	00000001000	same as
576	01101000	0000001101101	1856	00000001100	white
640	01100111	0000001001010	1920	00000001101	from this
704	011001100	0000001001011	1984	000000010010	point
768	011001101	0000001001100	2048	000000010011	
832	011010010	0000001001101	2112	000000010100	
896	011010011	0000001110010	2176	000000010101	
960	011010100	0000001110011	2240	000000010110	
1024	011010101	0000001110100	2304	000000010111	
1088	011010110	0000001110101	2368	000000011100	
1152	011010111	0000001110110	2432	000000011101	
1216	011011000	0000001110111	2496	000000011110	
1280	011011001	0000001010010	2560	000000011111	

31: Group 3 and 4 Fax Codes: (a) Termination Codes, (b) Make-Up Codes.

- Bir «run» kodu ya tek bir Termination kod veya bir veya daha fazla Make-up kod ve onu izleyen Termination koddan oluşur.

1. A run length of 12 white pels is coded as 001000.
2. A run length of 76 white pels $(= 64 + 12)$ is coded as 11011|001000.
3. A run length of 140 white pels $(= 128 + 12)$ is coded as 10010|001000.
4. A run length of 64 black pels $(= 64 + 0)$ is coded as 0000001111|0000110111.
5. A run length of 2561 black pels $(2560 + 1)$ is coded as 000000011111|010.

  - Her bir tarama satırı ayrı ayrı kodlanır ve kodlar özel EOL kodu ile sonlandırılır (0000000001)
  - Bir satır scan edildiğinde, her satırın soluna bir tane beyaz «pels» eklenir. Bu alıcı tarafından kod çözüldüğünde ortadan belirsizliği kaldırmak içindir

Örnek

14 pels oluşan bir satır 1w3b2w2b7wEOL

000111|10|0111|11|1111|0000000001

Örnek

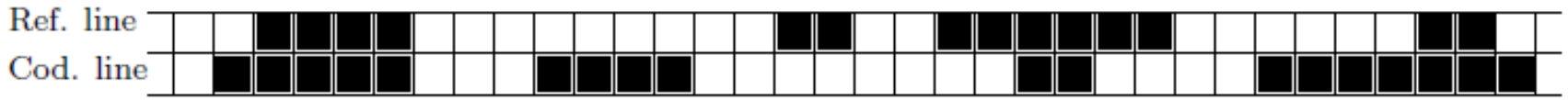
14 pels oluşan bir satır 3w5b5w2bEOL

1000|0011|1100|11|0000000001

- Grup 3, kodda hata düzeltmeye sahip değildir fakat hataların çoğunu bulabilir
- Huffman kodun doğasından dolayı iletim anında tek bir hatalı bit gönderilirse bile alıcı senkronizasyonun dışına çıkar ve hatalı bit stringi oluşur
- Bu sebepten dolayı, her satır ayrı ayrı kodlanır. Alıcı, hatayı bulmuşsa EOL koduna kadar olan bitleri atlar, böylece bir hata oluştuğunda alıcı tarafında en fazla bir satır düzeltilmeden diğer bilgiler alınır
- Eğer alıcı, satır sayısı kadar EOL kodunu göremez ise yüksek bir hata oranı olduğunu kabul eder ve işlemi iptal ederek durumu göndericiye bildirir
- Kodlar 2 ile 12 bit arasında olduğundan dolayı, alıcı 12 biti okuduktan sonra halen geçerli bir kod bulamamış ise hatayı bulur
- Her sayfanın sonunda 6 tane EOL kodu mevcuttur
- Bu yöntem **One-Dimensional Coding** olarak adlandırılır
- Sıkıştırma oranı görüntüye bağlıdır. Text ve B&W görüntülerde sıkıştırma oranı yüksektir
- Kısa «run» lardan oluşan görüntülerde ise özellikle gray scale görüntülerde başarı düşüktür

Facsimile Compression – 2 Dimensional Coding

- Tek boyutlu kodlama gray scale resimlerde iyi sonuç vermediğinden iki boyutlu kodlama geliştirilmiştir
- Grup 3'lü kodlama, faks makinelerinde seçimlik olarak bulunur
- Her EOL kodundan sonra fazladan bir bit kullanılır. Bu bit değeri 1 ise gelecek satırın 1D, 0 ise 2D ile kodlanacağı anlaşılır
- 2D kodlama MMR (Modified Modified Read) olarak adlandırılır. READ (Relative Element Address Designate)
- 2D kodlama yöntemi, mevcut scan line (coding line) ile bir önünde yer alan (reference line) satırı karşılaştırarak, aralarındaki farkları kayıt ederek çalışır
- Dokümanlarda bazen peş peşe gelen satırlar arasında çok küçük farklar olabilir. Örneğin, sayfanın başlangıcı sadece beyaz pels'lerden oluşmuş bir satır olabilir. Bu satır 1D ile kodlanır. İkinci satırda, bir üstte kodlanmış olan satır referans alınarak kodlanır ki bu da 2D kodlamadır



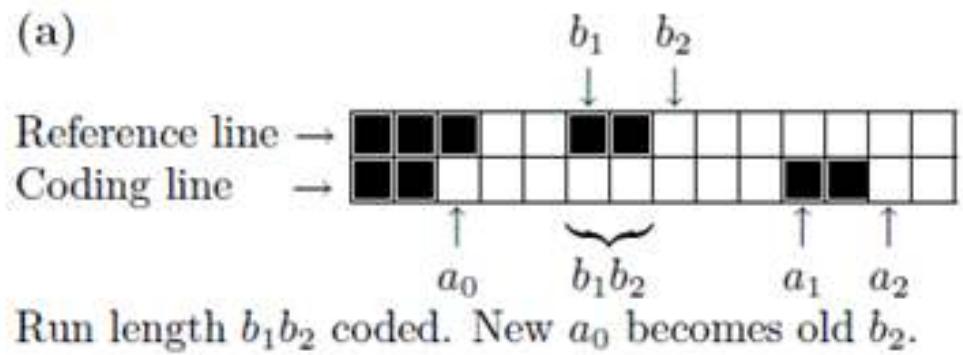
- 2D kodlama, 1D kodlamadan daha az güvenilirdir. Bir hata oluştuğunda bütün doküman etkilenir. Bu sebepten dolayı belli aralıklarla 2D kodlama kesilir ve bir sonraki satır 1D kodlama ile kodlanır
- T4 (Grup 3) bir satır 1D kodlama ile kodlandıktan sonra en fazla k-1 satır 2D kodlama ile kodlanır (standart çözünürlük k=2 ve yüksek çözünürlük k=4)
- T6 (Grup 4) kodlamada böyle bir zorunluluk yoktur

2D kodlamada, 3 farklı mod vardır. 2D kodlama, bu 3 farklı mod arasında değişimli kodlama yapar.

1- Pass Mode

b_2, a_1'in sol tarafındadır
a_1 ve b_1 aynı hizada olamaz
Bu mod tanımlandığında Pass Mode (0001) + (b_1 b_2) uzunluğu yazılır

a_0, b_2'nin altına taşınır
b_1 ve b_2 güncellenir
a_1 ve a_2 değişmez

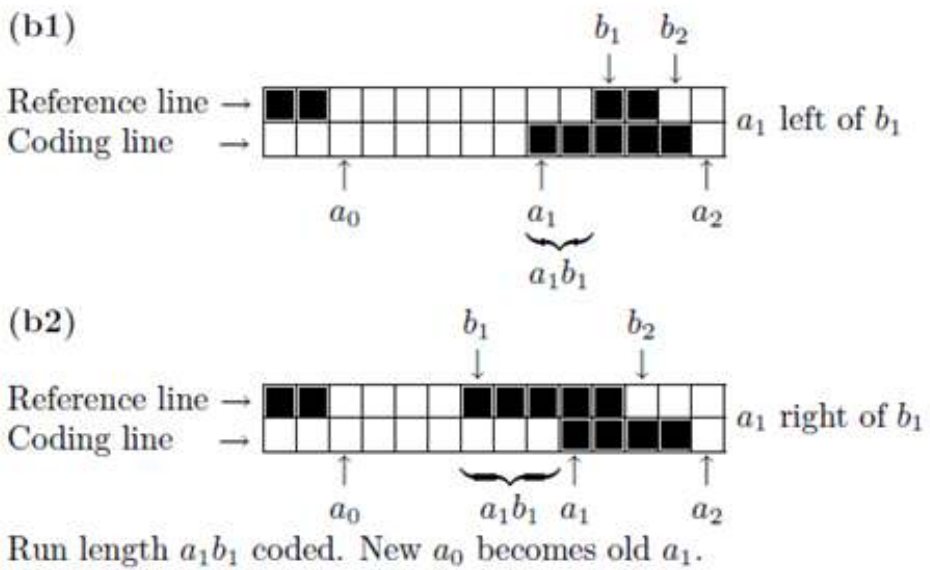


Run length b_1b_2 coded. New a_0 becomes old b_2.

2- Vertical Mode

a_1 ile b_1 arasında sağa veya sola doğru 2'den fazla «pels» olmamalıdır. Yedi farklı durum için 7 farklı kod üretilir.

$a_1, b_1 = 0$ V(0) → 1
$a_1, b_1 = -1$ VR(1) → 011
$a_1, b_1 = -2$ VR(2)→ 000011
$a_1, b_1 = -3$ VR(3)→ 0000011
$a_1, b_1 = 1$ VL(1) → 010
$a_1, b_1 = 2$ VL(2) → 000010
$a_1, b_1 = 3$ VL(3) → 0000010



Run length a_1b_1 coded. New a_0 becomes old a_1.

Vertical Mode bilgisi gönderildikten sonra

a_0, a_1 'in yerine
a_1, a_2 'in yerine
a_2 diğer «run» nın başına geçer

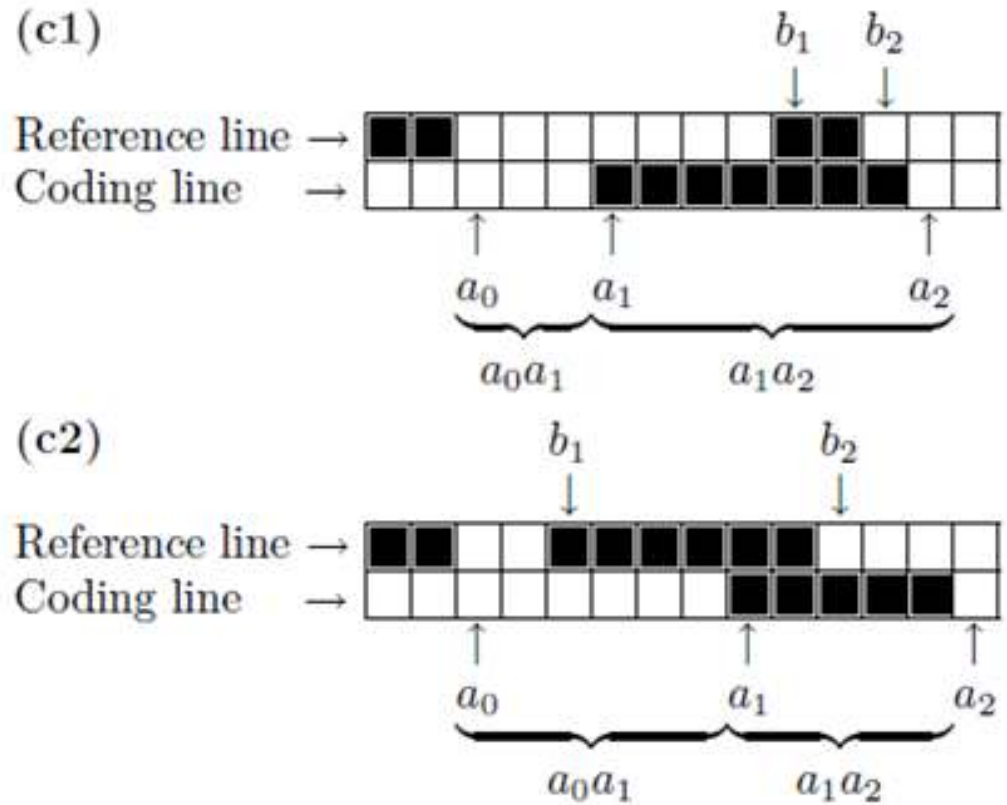
b_1, b_2 'in yerine
b_2 diğer «run» nın başına geçer

3- Horizontal Mode

a_1 ile b_1 arasında sağa veya sola doğru 3'den fazla «pels» olmalıdır

Horizontal Modun kodu 001 ve a_0a_1 ve a_1a_2 arası pels'in MH göre kod değeri gönderilir

a_0, a_2 'nin yerine geçer
a_1, a_2, b_1 ve b_2 güncellenir



New a_0 becomes old a_2.

Run lengths a_0a_1 (white) and a_1a_2 (black) coded.

Mode	Run length to be encoded	Abbre-viation	Codeword
Pass	b_1b_2	P	0001+coded length of b_1b_2
Horizontal	a_0a_1, a_1a_2	H	001+coded length of a_0a_1 and a_1a_2
Vertical	$a_1b_1 = 0$	V(0)	1
	$a_1b_1 = -1$	VR(1)	011
	$a_1b_1 = -2$	VR(2)	000011
	$a_1b_1 = -3$	VR(3)	0000011
	$a_1b_1 = +1$	VL(1)	010
	$a_1b_1 = +2$	VL(2)	000010
	$a_1b_1 = +3$	VL(3)	0000010
Extension			0000001000

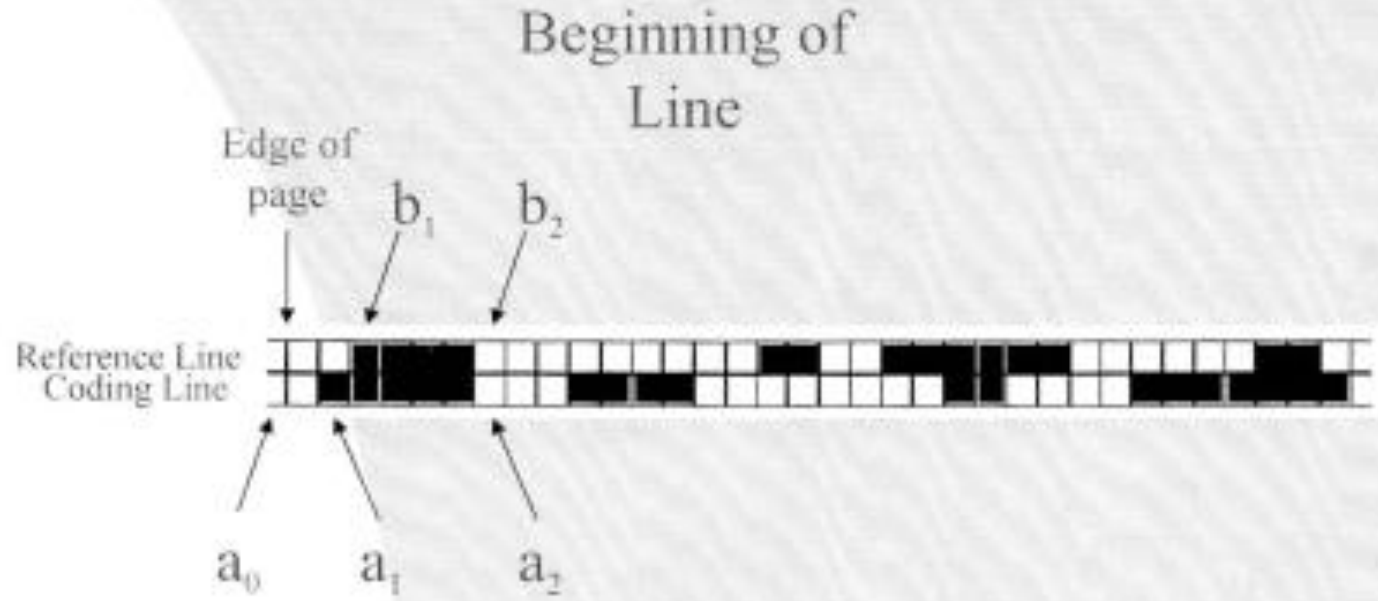
2D Codes for the Group 4 Method.

Tarama işlemi başladığında

a_0, coding line'da ilk başlangıç beyaz «pels»i
a_1, coding line'da ilk siyah «pels»in başlangıcını
a_2, coding line'da ilk beyaz «pels»in başlangıcını gösterir

b_1 ve b_2 ise referans line'daki birinci ve ikinci «run» ların başlangıcını gösterir
(0000001000) Extension kod olup, sayfanın sonuna gelmeden kodlama işlemini iptal etmek veya sayfanın geri kalan kısmını farklı kodla veya sıkıştırmadan iletmek için kullanılır

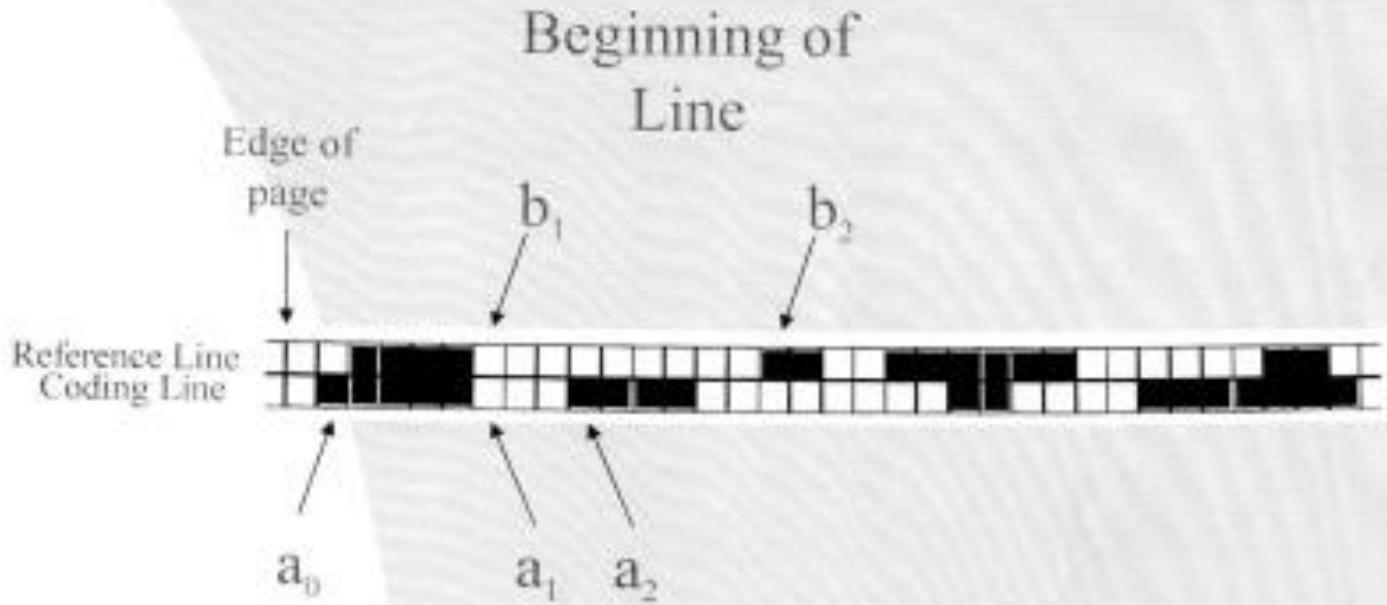
Short example



$a_1 b_1 = 1$, Vertical Mode

010

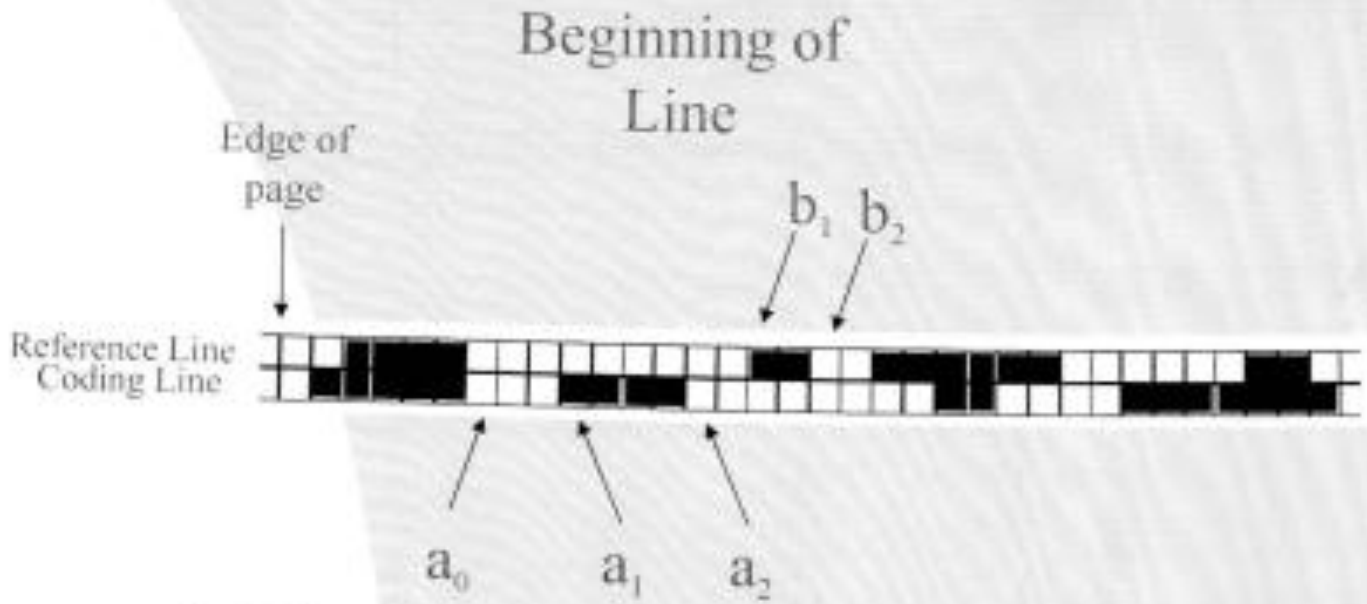
Short example



$a_1 b_1 = 0$, Vertical Mode

010 1

Short example

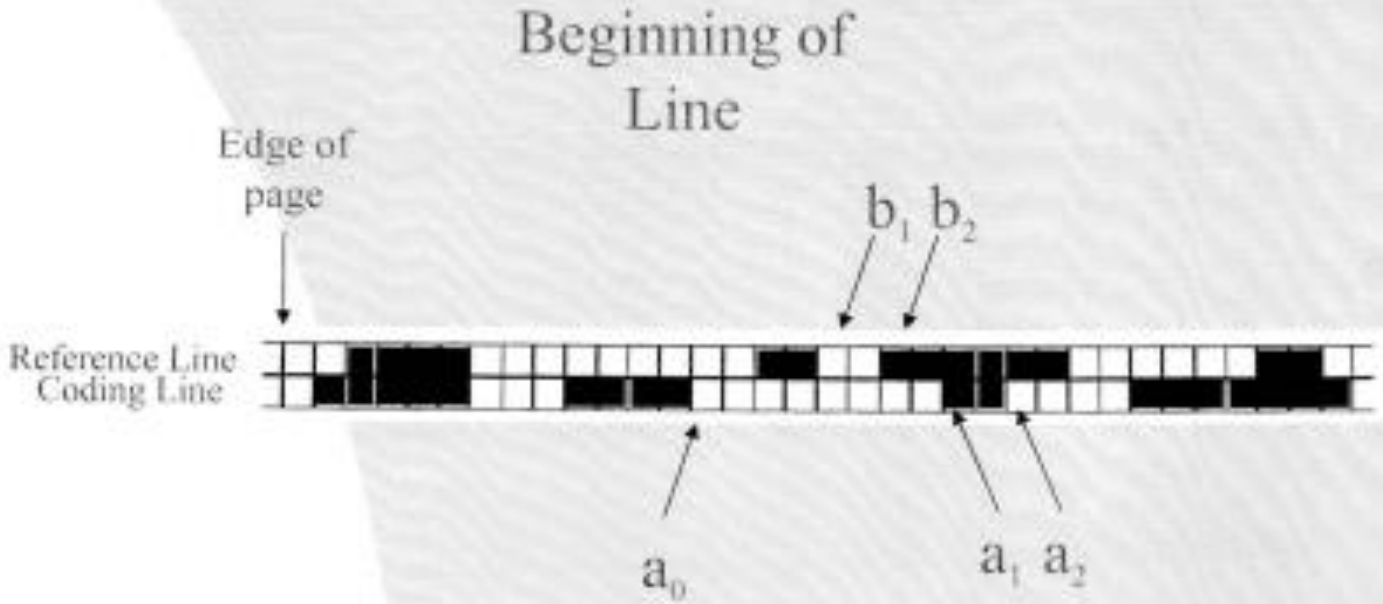


$|a_1 b_1| > 3$, Horizontal Mode

3 White, 4 Black

010 1 001 +1000 +011

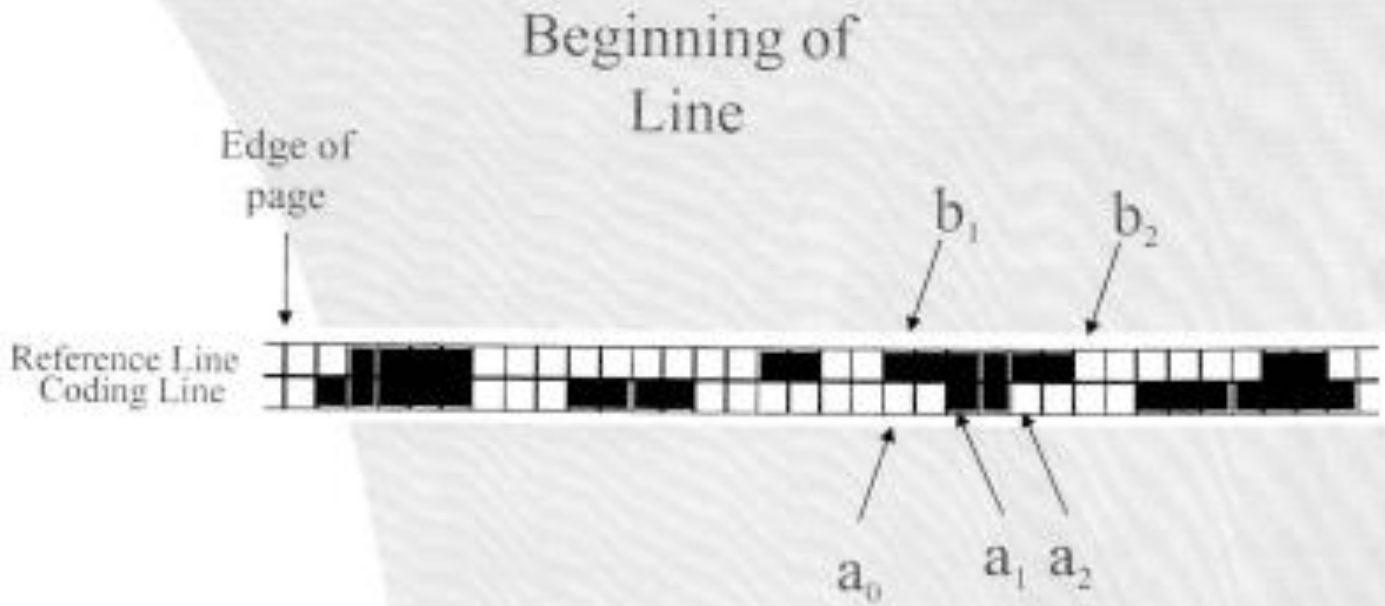
Short example



B_2 to the left of a_1,
Pass Mode + b1b2 (2 white)

010 1 001 1000 011 0001 +0111

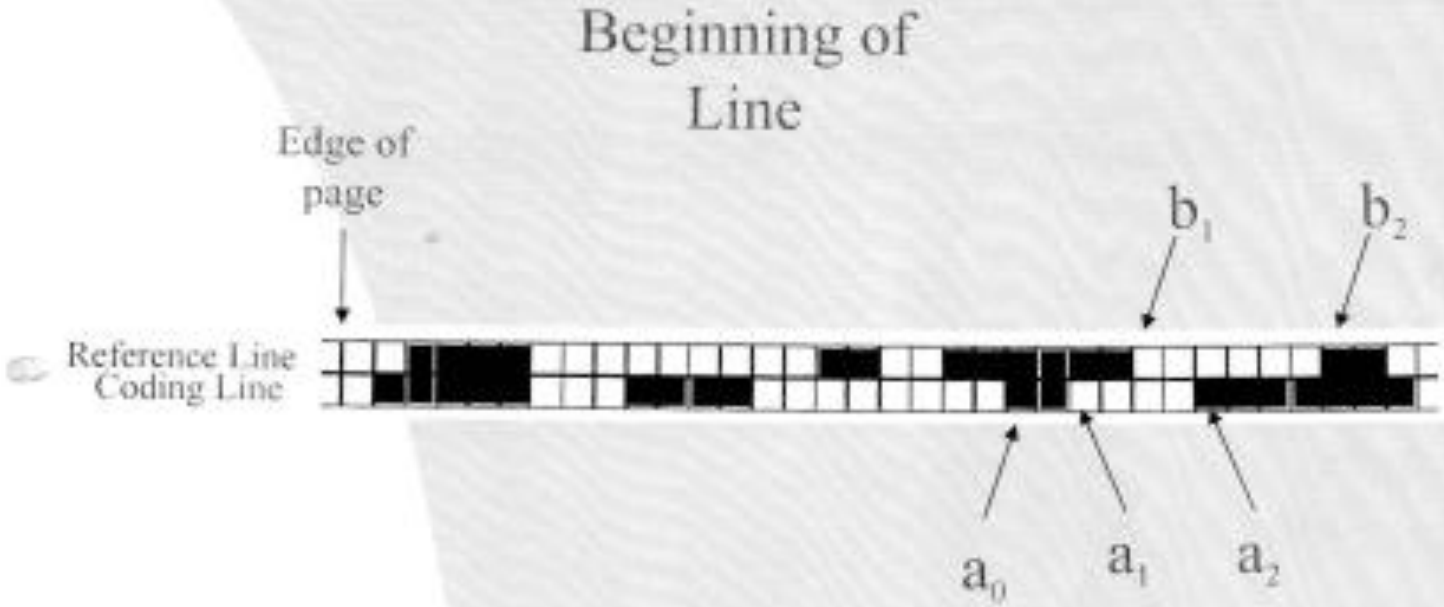
Short example



$a_1 b_1 = -2$, Vertical Mode

010 1 001 1000 011 0001 0111 000011

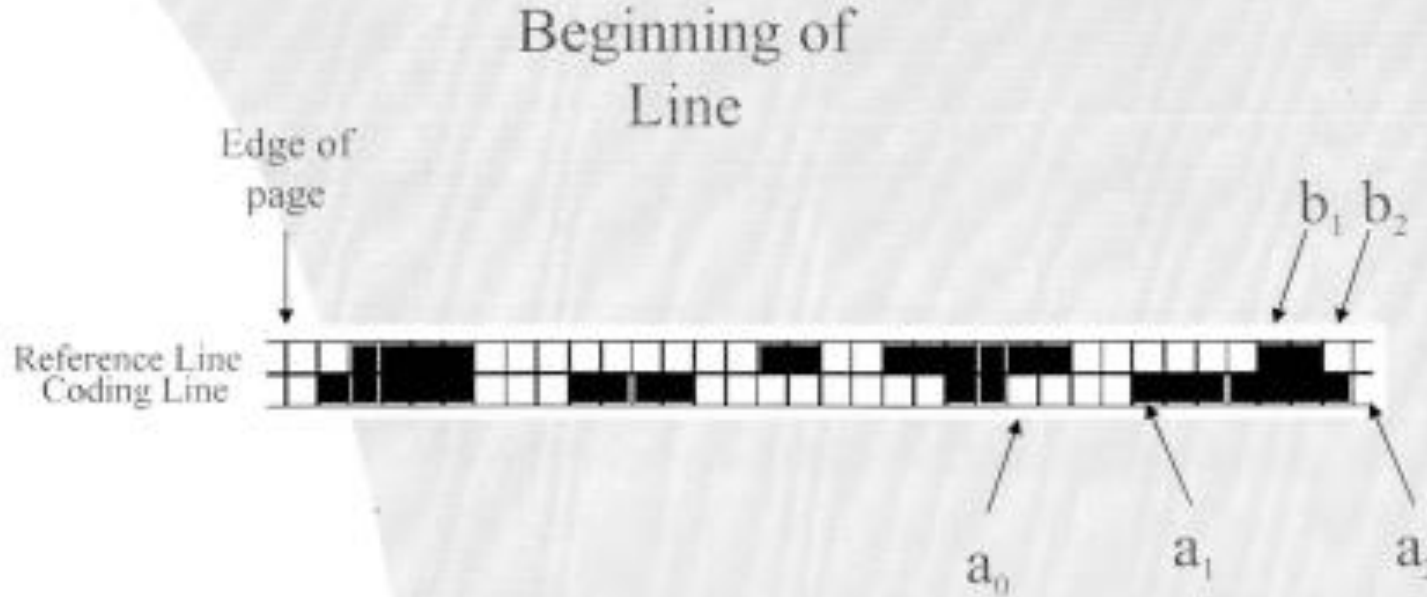
Short example



$a_1 b_1 = 2$, Vertical Mode

010 1 001 1000 011 0001 0111 000011 000010

Short example



$|a_1 b_1| > 3$, Horizontal Mode

4 White, 7 Black

010 1 001 1000 011 0001 0111 000011 000010 001 +1011+00011