

Elektronski fakultet, Niš



**KOMPRESIJA PODATAKA
PRIMENOM DVODIMENZIONALNOG
FAKSIMIL KODIRANJA**

**SEMINARSKI RAD IZ METODA I SISTEMA ZA
OBRADU SIGNALA**

Profesor:

Prof.dr Miloš Radmanović

Studenti:

Uroš Mitić

Danilo Vidanović

Niš, januar 2021

Sadržaj

Uvod.....	3
Statističke metode.....	4
Faksimil kompresija.....	7
Dvodimenzionalni faksimil kodiranje.....	10
Procedura kodiranja.....	13
Realizacija algoritma u programskom jeziku javascript.....	17
Primer testiranja.....	18
Literatura.....	20

Uvod

U obradi signala, kompresija podataka, kodiranje izvora,[1] ili redukcija bitne stope obuhvata kodiranje informacija koristeći manji broj bitova nego u originalnoj reprezentaciji.[2] Kompresija može da bude bilo sa gubicima ili bez gubitaka. Kompresija bez gubitka redukuje bitove putem identifikacije i eliminacije statističke izlišnosti. Informacije se ne gube pri tom vidu kompresije. Kompresija sa gubicima redukuje bitove uklanjanjem nepotrebnih ili manje važnih informacija.[3]

Proces redukovanja veličine datoteke sa podacima se obično naziva kompresijom podataka. U kontekstu prenosa podataka, taj proces se naziva kodiranjem izvora; kodiranje se tipično vrši na izvoru podataka pre njihovog skladištenja ili prenosa.[4] Kodiranje izvora ne treba mešati sa kodiranjem kanala radi detekcije i korekcije grešaka, ili kodiranjem linije sredstva za mapiranje podataka na signal.

Kompresija je korisna jer se njom redukuju resursi neophodni za skladištenje i prenos podataka. Računarski resursi se konzumiraju pri procesu kompresije i isto tako pri suprotnom procesu (dekompresiji). Kompresija podataka je zavisna od kompromisa prostorno-vremenske kompleksnosti. Na primer, kompresiona shema za video može da zahteva skup hardver da bi se video dovoljno brzo dekompresovao tako da se može gledati kao da je već bio dekomprimovao. Opcija da se video u potpunosti dekompresuje pre gledanja može da bude nepodesna ili da zahteva znatan dodatni prostor. Dizajn shema za kompresiju podataka obuhvata kompromise među mnoštvom različitih faktora, uključujući stepen kompresije, količinu unešene distorzije (pri korišćenju kompresije sa gubitkom), i računarske resurse neophodne za komprimovanje i dekomprimovanje podataka.

Kompresija podataka se postiže smanjenjem suvišnosti, ali to takođe čini podatke manje pouzdanim i sklonijim greškama. S druge strane, dodavanjem kontrolnih bitova i bitova parnosti povećava se veličina koda, a samim tim i redundancija. Kompresija podataka i pouzdanost su suprotnosti i zanimljivo je primetiti da je pouzdanost relativno nedavno polje, dok je kompresija podataka postojala i pre pojave računara. Primer toga je Morse-ova azbuka iz 1838. godine koja koristi jednostavne oblike kompresije.

Stoga se čini da smanjenje redundancije prirodno dolazi svima koji rade na kodovima. Moderne metode kompresije podataka koje se koriste danas nastale su tokom poslednjih 15-20 godina, dok su one starije neefikasne u poređenju sa sadašnjim i one su u današnjici samo od istorijskog značaja.

Statističke metode

Statističke metode predstavljaju metode koje koriste kodove promenljivih veličina sa kraćim kodovima dodeljenim simbolima ili grupama simbola koji se pojavljuju češće u podacima. Dizajneri i implementatori kodova promenljivih veličina moraju se suočiti sa dva problema: prvi je dodeljivanje kodova koji se mogu jednoznačno dekodirati i dodeljivanja kodova sa minimalnom prosečnom vrednošću.

Važni koncepti iz teorije informacija vode ka definiciji viška, tako da kasnije možemo jasno videti i izračunati kako se redundantnost smanjuje ili uklanja različitim metodama.

A	.-	N	-.	1	.----	Period	.-.-.
B	---	O	---	2	..---	Comma	---..
C	-..	P	-..	3	...--	Colon	---...
Ch	----	Q	--.-	4-	Question mark	..-..
D	-..	R	.-.	5	Apostrophe	.-....
E	.	S	...	6	-....	Hyphen	-....-
F	T	-	7	--...	Dash	-...-
G	--.	U	..-	8	---..	Parentheses	-...-
H	V	...-	9	----.	Quotation marks	.-...-
I	..	W	.-	0	-----		
J	.---	X	-..-				
K	-.-	Y	-.--				
L	...-	Z	--..				
M	--						

Ako se smatra da je trajanje tačke jedna jedinica, onda je crtica tri jedinice. Prostor između tačaka i crtica jednog znaka je jedna jedinica, između znakova tri jedinice, a između reči šest jedinica (pet za automatski prenos). Kako bi se ukazalo da je došlo do greške treba poslati osam tačaka.

Samuel Morse koristio je kodove promenljive veličine kada je dizajnirao svoj dobro poznati telegrafski kod. Verzija prvog njegovo koda bila je složenija od verzije koju je kasnije pisao. Prva verzija sadržala je kratke i duge crtice gde su sekvence tih crtica predstavljale brojeve. Svakoј reči dodeljen je kodni broj, Morse je kasnije stvorio i rečnik tih kodova.

Intuitivno znamo šta su informacije. Stalno primamo i šalјemo informacije u obliku teksta, govora i slika. Takođe smatramo da su informacije nedostižna nematematička veličina koja se ne može precizno definisati i izmeriti. Važnost teorije informacija je u tome što ona kvantifikuje informacije odnosno pokazuje kako se mere informacije kako bismo mogli odgovoriti na pitanje 'koliko je informacija uključeno u ovom podatku?' sa tačnim broјem.

Kvantifikovanje informacija zasniva se na zapažanju da je informativni sadržaj poruke ekvivalentan količini iznenađenja u poruci. Ako kažemo nekome nešto što ga zaista iznenađuje daćemo mu više informacija bez obzira na broj reči koje smo upotrebili.

Uzmimo za primer svima poznat događaj koji možemo lako da analiziramo, a to je bacanje novčića. Rezultat bacanja novčića u početku nije poznat tako da moramo baciti novčić da bismo rešili neizvesnost. Rezultat je ili glava ili rep koje možemo izraziti kao DA ili NE ili kao 0 ili 1 (preko bitova). Ono što ovaj primer čini važnim je činjenica da se lako može generalizovati. Mnogi problemi iz stvarnog života mogu se rešiti i njihova rešenja izraziti pomoću bitova. Princip toga jeste pronalaženje minimalnog broja da/ne pitanja na koja mora biti odgovoreno da bi se došlo do rezultata. Kako se odgovor na da/ne pitanje može izraziti јednim bitom, broj pitanja biće јednak broju bitova potrebnih za izražavanje informacija sadržanih u rezultatu.

Uzmimo u obzir i malo složeniji špil od 64 karte koje su numerisane broјevima 1-64. Neka je igrač A izvukao јednu kartu, a igrač B treba da pogodi koji je broj te karte. Postavlja se pitanje koji je minimalan broj pitanja da/ne koji su potrebni da se postave igraču A kako bi se došlo do pogotka. Rezultat se dobija primenom tehnike binarne pretrage. Koristeći ovu tehniku igrač B bi trebalo da podeli opseg 1-64 na dva opsega i postavlja pitanje da li je je rezultat između 1 i 32? Ako je odgovor ne onda je rezultat u rasponu od 33 do 64- Ovaj opseg je podeljen sa dva i sledeće pitanje bi bilo da li je rezultat između 33 i 48. Ovaj proces se nastavlja sve dok se opseg koji koristi B ne smanji na јedan broj, tako da bi konkretno za ovaj primer bilo potrebno tačno 6 da/ne pitanja za dobijanje rezultata. Rezultat je dobijen iz $\log_2 64 = 6$. Zbog toga kažemo da je logaritam matematička funkcija koja kvantifikuje informacije.

Sada ćemo se okrenuti predajniku, delu hardvera koji može prenositi podatke preko komunikacione linije (kanala). U praksi, takav predajnik šalje binarne podatke (npr. modem). Međutim, da bismo dobili opšte rezultate, pretpostavljamo da su podaci niz koji se sastoji od pojavljivanja n simbola a_1 do a_n . Takav skup je alfabet od n simbola. Što se hardvera tiče to znači da mora da bude u mogućnosti da prenosi n diskretnih nivoa. Ako predajniku treba $1/s$ vremenskih jedinica da pošalje jedan simbol, tada je brzina prenosa s simbola po vremenskoj jedinici. Uobičajeni primer je $s=28800$ baud (baud je termin za bitove u sekundi) što u prevodu znači $1/s \sim 34,7 \mu\text{s}$. U jednoj vremenskoj jedinici predajnik može da pošalje s simbola što je u pogledu sadržaja informacija ekvivalentno $= s \log_2 n$ bita. Označavamo sa $H = s \log_2 n$ količinu informacija, merenih u bitovima, prenete u svakoj vremenskoj jedinici. Sledeći korak je izražavanje H u terminima verovatnoće pojave n simbola. Pretpostavimo da se simbol a_i javlja u podacima sa verovatnoćom P_i . Zbir verovatnoća jednak je: $P_1 + P_2 + \dots + P_n = 1$. U posebnom slučaju gde je svih n verovatnoća jednakih, $P_i = 1/n$ dobijamo da je $1 = P_i = nP$ podrazumevajući da je $P = 1/n$ i $H = s \log_2 n + s \log_2 (1/P) = -s \log_2 P$. Generalno, verovatnoće su različite, a mi želimo da izrazimo H u smislu svih njih. S obzirom da se simbol a_i javlja u razdelu P_i vremena u podacima, on se javlja u proseku sP_i puta svake vremenske jedinice pa je njegov doprinos za H $-sP_i \log_2 P_i$. Zbir doprinosa svih n simbola H postaje $H = -s \sum_{i=1}^n P_i \log_2 P_i$.

Dakle, H je količina informacija u bitovima koju predajnik šalje u jednoj vremenskoj jedinici. Količina informacija sadržanih u jednom simbolu osnove n je H/s jer je potrebno vreme $1/s$ da se prenese jedan simbol.

Teoriju informacija razvio je krajem 1940-ih Claude Shannon i izabrao termin za količinu podataka entropija jer se taj termin koristi u termodinamici da bi se naznačila količina neuređenosti u fizičkom sistemu. Entropija podataka zavisi od pojedinačnih verovatnoća P_i i najveća je kada su svih n verovatnoća jednake. Glavna teorema koju je dokazao Shannon u osnovi govori da se poruka od n simbola može u proseku kompresovati do nH bitova, ali ne i dalje.

Faksimil kompresija

Kompresija podataka je posebno važna kada se prenose slike preko komunikacione linije jer korisnik obično čeka na prijemniku, željan da nešto brzo vidi. Dokumenti koji se prenose između faks uređaja šalju se kao bitmape pa je zbog toga bila potrebna standardna metoda za kompresiju podataka kada su te mašine postale popularne.

Prvi standardi kompresije podataka koje je razvijao ITU-T bili su T2 (takođe poznat kao grupa 1) i T3 (grupa 2). Oni su sada zastareli i zamenjeni su sa T4 (grupa 3) i T6 (grupa 4). Grupu 3 trenutno koriste sve faks mašine dizajnirane da rade sa javnom komutiranom telefonskom mrežom (PSTN). To su mašine koje imamo kod kuće, i u vreme pisanja teksta one rade brzinom od 9600 bauda. Grupu 4 koriste faks mašine dizajnirane za rad na digitalnoj mreži kao što je ISDN. One imaju tipične brzine od 64K bauda. Obe metode mogu stvoriti odnos kompresije od 10:1 ili bolji, smanjujući vreme prenosa tipične stranice na oko minut kod prve i nekoliko sekundi kod druge.

Faks mašina radi tako što skenira dokument red po red pretvarajući svaki red u male crno-bele tačke koje se zovu **pelovi**. Horizontalna rezolucija je uvek 8.05 pela po milimetru (oko 205 pela po inču). Tako se linija skeniranja široka 8,5 inča pretvara u 1728 pelova. Standard T4, međutim, preporučuje skeniranje samo oko 8,2 inča, stvarajući tako 1664 pela po liniji skeniranja. Vertikalna rezolucija je 3,85 linija skeniranja na milimetar (standardni režim) ili 7,7 linija po milimetru (fini režim). Mnoge faks mašine imaju vrlo fin način rada, gde skeniraju 15,4 linije po milimetru.

Naredna tabela predstavlja stranicu visoku 10 inča i prikazuje ukupan broj pelova po stranici i tipična vremena prenosa za tri režima bez kompresije. Vremena su dugačka što pokazuje koliko je važna kompresija podataka u prenosu faksa.

Scan	Pels per	Pels per	Time	Time
lines	line	page	(sec.)	(min.)
978	1664	1.670M	170	2.82
1956	1664	3.255M	339	5.65
3912	1664	6.510M	678	11.3

Deset centimetara je jednako 254mm. Broj pelova je u milionima, a vreme prenosa na 9600 bauda bez kompresije je između 3 i 11 minuta u zavisnosti od režima. Ako je stranica kraća od 10 inča ili ako je veći deo bele boje faktor kompresije može biti 10:1 ili bolji.

Da bi izvršio kod grupe 3 ITU-T je izbrojao sve dužine izvršenja belih i crnih pelova u skupu od osam dokumenata za koje smatraju da predstavljaju tipičan tekst i slike poslane faksom i koristio Haffmanov algoritam da dodeljivanje promenljive.

Utvrđeno je da su najčešće dužine izvršenja 2, 3 i 4 crna piksela, pa su im dodeljeni najkraći kodovi što možemo videti u sledećoj tabeli. Slede dužine pokretanja od 2-7 belih piksela, kojima su dodeljeni nešto duži kodovi. Većina dužina izvođenja bila je retka i dodeljivani su im dugi 12-bitni kodovi. Zato grupa 3 koristi kombinaciju RLE i Huffman kodiranje.

S obzirom da dužine izvršenja mogu biti velike, Huffmanov algoritam je modifikovan pa grupa 3 predstavlja modifikovani Huffmanov kod (takođe nazvan MH).

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

Svaka linija za skeniranje kodirana je odvojeno, a njen kod se završava posebnom 12-bitnom EOL šifrom 000000000001. Svaka linija dobija jednu belu pločicu koja joj je dodata levo kada se skenira. Ovo se radi kako bi se uklonila bilo kakva dvosmislenost kada se linija dekodira kraj prijema. Nakon čitanja EOL-a za prethodni red prijemnik pretpostavlja da nova linija započinje nizom belih pelova i ignoriše prvu od njih.

Na primer:

Linija od 14 pločica kodirana je kao 1 white, 3 black, 2 white, 2 black, 7 white i EOL. Dekoder ignoriše belu pločicu na početku.



Linija je kodirana kao 3 white, 5 black, 5 white, 2 black i EOL



Kod grupe 3 nema ispravku grešaka, ali mnoge greške se mogu otkriti. Zbog prirode Haffmanovog koda čak i jedan loš bit u prenosu može dovesti do toga da se prijemnik isključi iz sinhronizacije i da proizvede niz pogrešnih pelova. Ovo je zašto je svaka linija skeniranja posebno kodirana. Ako prijemnik otkrije grešku onda preskače bitove tražeći EOL. Na ovaj način jedna greška može dovesti do pogrešnog prijema najviše jedne linije skeniranja. Ako prijemnik ne vidi EOL nakon određenog broja linija, pretpostavlja veliku stopu grešaka i odmah prekida postupak obaveštavajući predajnik. Budući da su kodovi dugi između 2 i 12 bita, prijemnik otkriva grešku ako ne može dekodirati važeći kod nakon čitanja 12 bita. Budući da je svaka linija kodirana odvojeno ovaj metod je jednodimenzionalna šema kodiranja. Step kompresije zavisi od slike. Slike sa velikim susednim crnim ili belim površinama (tekst ili crno-bele slike) mogu biti jako kompresovane. Slike sa nijansama sive (poput skeniranih fotografija) mogu ponekad proizvesti negativnu kompresiju. Takve nijanse nastaju polutoniranjem koje pokriva područja sa mnogo naizmeničnih crnih i belih pelova.

Zamislamo liniju skeniranja gde svi pokreti imaju dužinu jedan. Kod jedne bele palete je 000111, a kod jedne crne palete 010. Dva uzastopna pola različitih boja su tako kodirana u 9 bitova. Pošto nekodirani podaci zahtevaju samo dva bita (01 ili 10) odnos kompresije je $9/2 = 4,5$ (kompresovani tok je 4,5 puta duži od nekompresovanog – veliko proširenje).

Dvodimenzionalno faksimil kodiranje

Dvodimenzionalno kodiranje je razvijeno jer jednodimenzionalno kodiranje ne daje dobre rezultate za slike sa sivim površinama. Dvodimenzionalno kodiranje nije obavezno na faks uređajima koji koriste grupu 3, ali je jedini metod koji koriste mašine namenjene za rad na digitalnoj mreži. Kada faks uređaj koji koristi grupu 3 podržava dvodimenzionalno kodiranje kao opciju svaki EOL prati jedan dodatni bit, što ukazuje na metod kompresije koji se koristi za sledeću liniju skeniranja. Taj bit je 1 ako je sledeći red kodiran jednodimenzionalnim kodiranjem, a 0 ako je kodiran dvodimenzionalnim kodiranjem.

Dvodimenzionalna metoda kodiranja se takođe naziva MMR za modifikovani modifikovani READ gde READ označava relativnu oznaku adrese elementa. Termin 'modifikovano modifikovano' koristi se jer je ovo modifikacija jednodimenzionalnog kodiranja, što je samo po sebi modifikacija originalne Haffmanove metode.

Dvodimenzionalna metoda kodiranja funkcioniše tako što se upoređuje trenutna linija skeniranja (koja se naziva linija za kodiranje) sa prethodnom (koja se naziva referentna linija) i beleže razlike između njih, pod pretpostavkom da će se dve uzastopne linije u dokumentu normalno razlikovati za samo nekoliko pelova. Metoda je zasnovana na tome da se iznad stranice nalazi potpuno bela linija koja se koristi kao referentna linija za prvu liniju skeniranja stranice. Nakon kodiranja prve linije ona postaje referentna linija, a druga linija skeniranja je kodirana. Kao I kod jednodimenzionalnog kodiranja, pretpostavlja se da svaka linija započinje belom pločom koju prijemnik ignoriše.

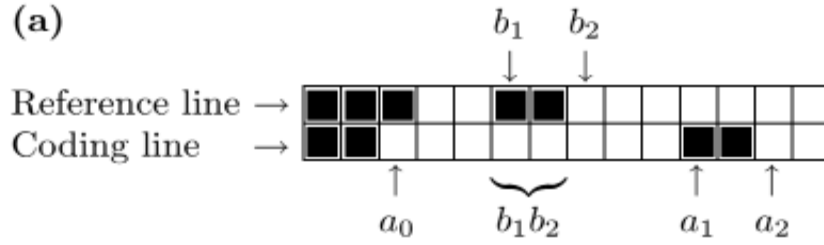
Skeniranje linije kodiranja I upoređivanje sa referentnom linijom rezultira u 3 slučaja. Režim se identifikuje upoređivanjem sledeće dužine izvođenja na referentnoj liniji (b1b2) sa trenutnom dužinom izvođenja (a0a1) i sledeće (a1a2) na liniji kodiranja. Svaka od ove tri mora biti ili crna ili bela. Tri načina su sledeća:

1. Prolazni režim. To je slučaj kada je b1b2 levo od a1a2, a b2 levo od a1 (slika a). Ovaj režim ne uključuje slučaj kada je b2 iznad a1. Kada se identifikuje ovaj režim, dužina izvođenja b1b2 kodira se pomoću kodova iz tabele) i prenosi. Pokazivač a0 pomeren je ispod b2 i ažurirane su četiri vrednosti b1, b2, a1 i a2.
2. Vertikalni režim. b1b2 preklapa se sa a1a2 za ne više od tri pela. Pod pretpostavkom da se uzastopne linije ne razlikuju mnogo ovo je najčešći slučaj. Kada se identifikuje ovaj način rada bira se jedan od sedam kodova koji se prenosi. Relativna udaljenost a1b1 može da preuzme jednu od sedam vrednosti V(0), VR(1), VR(2), VR(3), VL(1), VL(2) i VL(3), od kojih je svaka predstavljena posebnom kodom reči. Indeksi R i L označavaju da je a1 desno ili levo od b1, a broj u zagradama označava vrednost rastojanja a1b1. Pokazivači se ažuriraju kao u prethodnom slučaju.

3. Horizontalni režim. b_1b_2 se preklapa sa a_1a_2 za više od tri pela. Kada se identifikuje ovaj režim, dužine izvođenja a_0a_1 i a_1a_2 kodiraju s pomoću kodova iz tabele I prenose se. Pokazivači se ažuriraju kao u slučajevima 1 i 2.

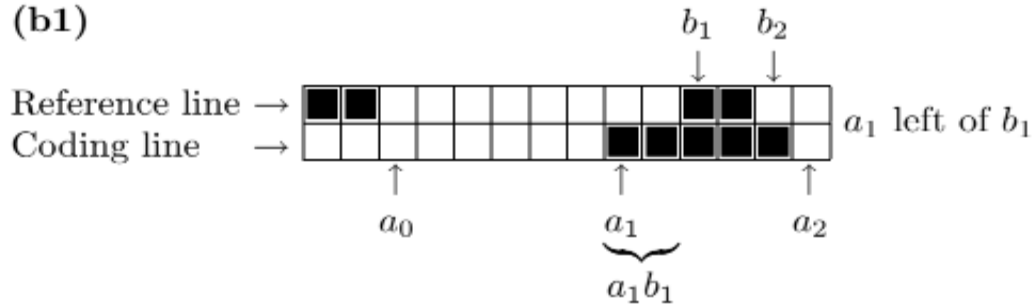
Mode	Run length to be encoded	Abbreviation	Codeword
Pass	b_1b_2	P	0001+coded length of b_1b_2
Horizontal	a_0a_1, a_1a_2	H	001+coded length of a_0a_1 and a_1a_2
Vertical	$a_1b_1 = 0$	V(0)	1
	$a_1b_1 = -1$	VR(1)	011
	$a_1b_1 = -2$	VR(2)	000011
	$a_1b_1 = -3$	VR(3)	0000011
	$a_1b_1 = +1$	VL(1)	010
	$a_1b_1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

(a)

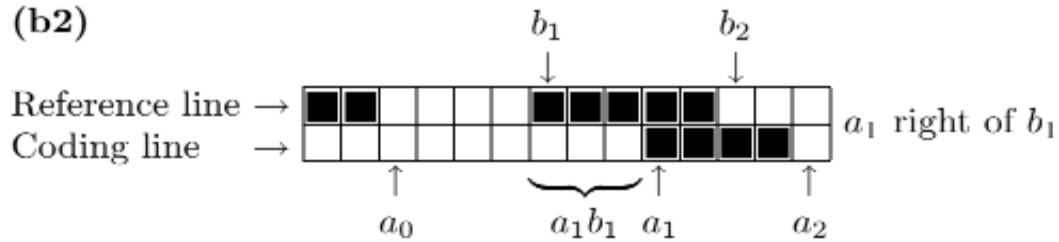


Run length b_1b_2 coded. New a_0 becomes old b_2 .

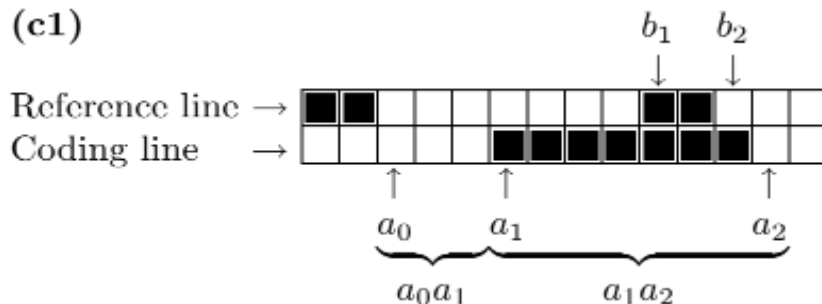
(b1)



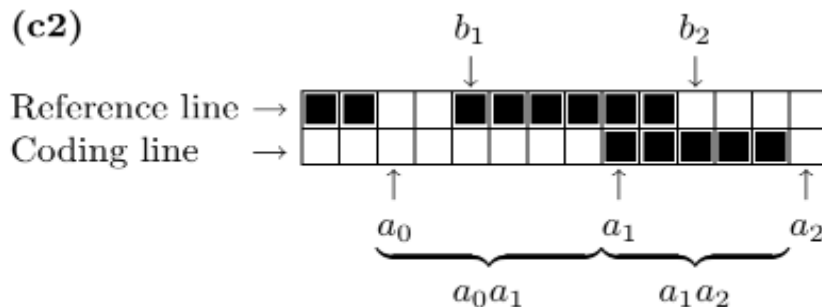
(b2)



(c1)



(c2)



Run lengths a_0a_1 (white) and a_1a_2 (black) coded. New a_0 becomes old a_2 .

Kada započne skeniranje, pokazivač a_0 je postavljen na zamišljenu belu pelu sa leve strane kodirajuće linije, a a_1 je postavljen tako da pokazuje na prvu crnu pelu na kodiranoj liniji. (budući da a_0 odgovara zamišljenoj peli, prva držina izvršenja biće $|a_0a_1| - 1$). Pokazivač a_2 postavljen je na prvu belu pelu koja sledi nakon toga. Pokazivači b_1 , b_2 postavljeni su da ukazuju na početak prvog i drugog izvršenja na referentnoj liniji, respektivno. Nakon identifikacije trenutnog režima i prenosa kodova prema tabeli, a_0 se ažurira kao što je prikazano na dijagramu toka, a ostali četiri pokazivača se ažuriraju u odnosu na novi a_0 . Proces se nastavlja sve dok se ne postigne kraj linije kodiranja.

1. a_0 je prva kuglica nove kodne reči i može biti crna ili bela
2. a_1 je prva bela desno od a_0 druge boje
3. a_2 je prva bela desno od a_1 druge boje
4. b_1 je prva bela na referentnoj liniji desno od a_0 druge boje
5. b_2 je prva bela na referentnoj liniji desno od b_1 druge boje

Dvodimenzionalni metod kodiranja je manje pouzdan od jednodimenzionalnog kodiranja jer će greška u dekodiranju linije prouzrokovati greške u dekodiranju svih njegovih naslednika i širiće se kroz ceo document. Zbog toga grupa 3 uključuje zahtev da se nakon što je linija kodirana jednodimenzionalnom metodom, da se najviše $K-1$ linija kodira dvodimenzionalnom metodom kodiranja. Za standardnu rezoluciju $K=2$, a za finu rezoluciju $K=4$. Grupa 4 nema ovaj zahtev i koristi isključivo dvodimenzionalno kodiranje.

Procedura kodiranja

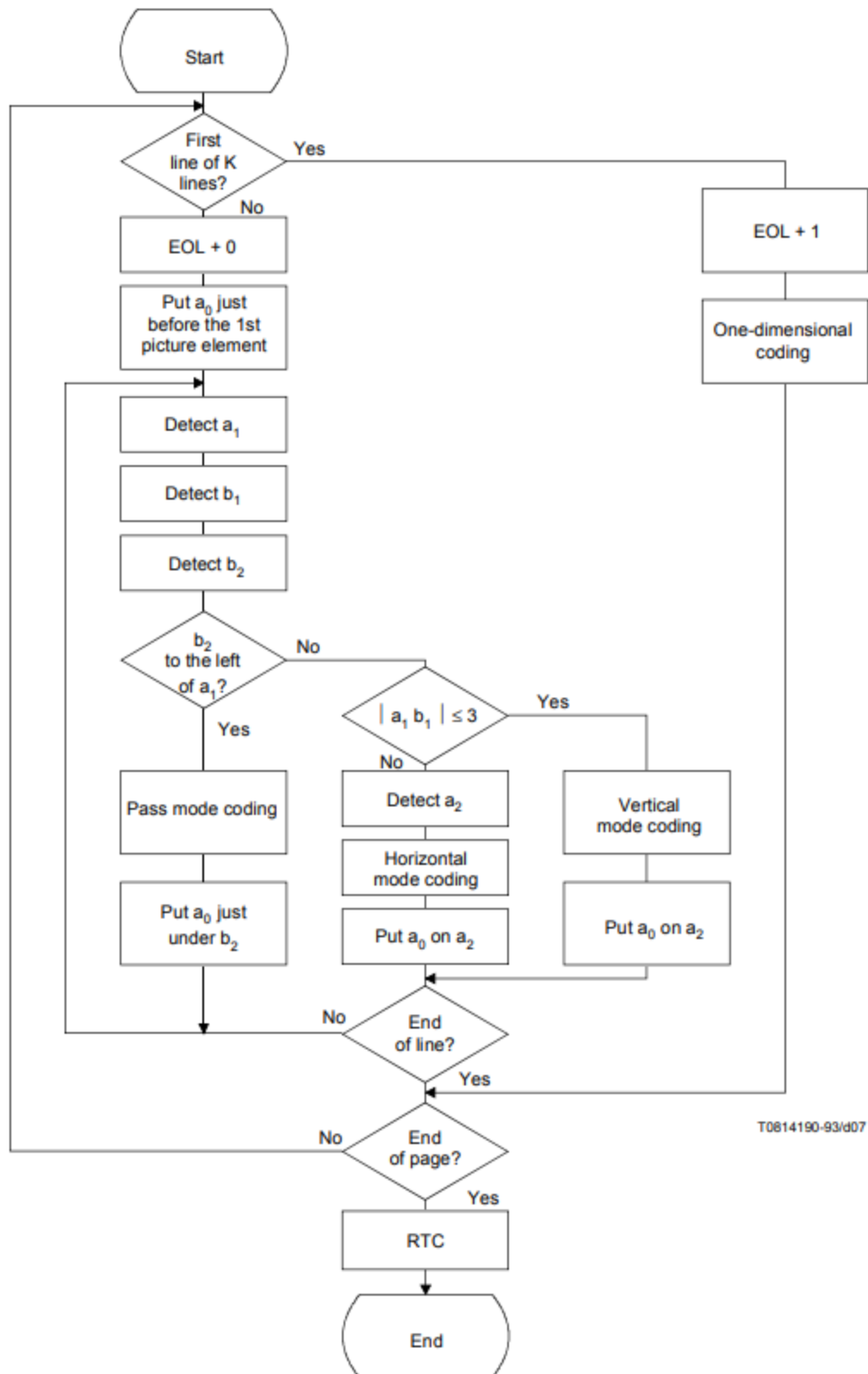
Procedura kodiranja identifikuje režim kodiranja koji će se koristiti za kodiranje svakog elementa koji se menja duž linije kodiranja. Kada je jedan od tri režima kodiranja identifikovan u skladu sa dole pomenutim korakom 1 ili korakom 2, odgovarajuća kodna reč bira se iz kodne tabele date u tabeli 5. Procedura kodiranja je kao što je prikazano u dijagramu toka na sledećoj strani.

Korak 1

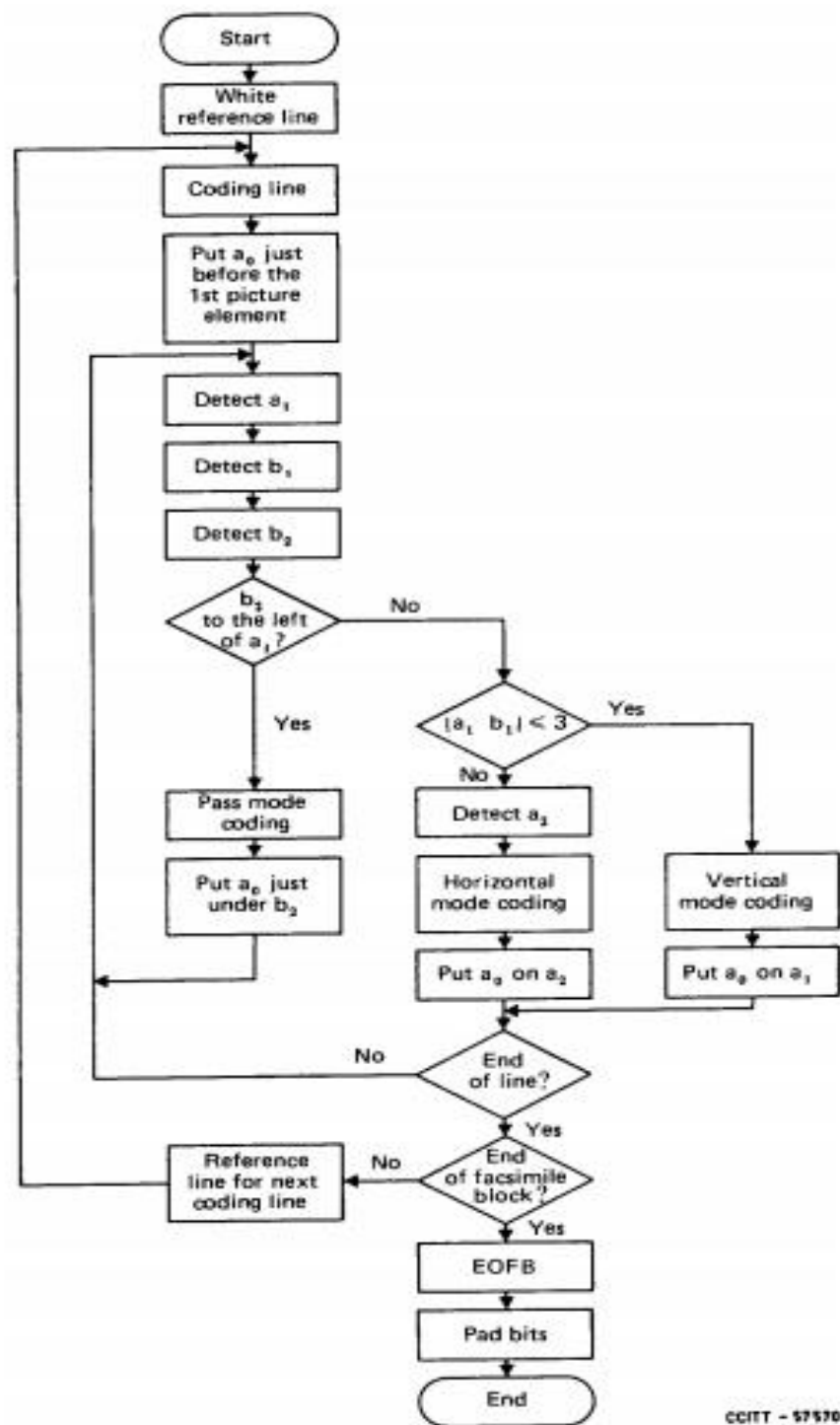
- i) Ako je režim prolaza identifikovan, ovo je kodirano pomoću reči 0001. Nakon ove obrade, element slike a'_0 ispod b_2 smatra se novim elementom početne slike a_0 za sledeće kodiranje (pogledajte sliku 4).
- ii) Ako režim prolaza nije otkriven, onda pređite na korak 2.

Korak 2

- i) Odredi apsolutnu vrednost relativne udaljenosti a_1b_1 .
- ii) Ako je $|a_1b_1| \leq 3$, a_1b_1 kodiran vertikalnim režimom, nakon čega se pozicija a_1 smatra novim elementom početne slike a_0 za sledeće kodiranje.
- iii) Ako su $|a_1b_1| > 3$, prateći horizontalni režim koda 001, a_0a_1 i a_1a_2 kodiran jednodimenzionalnim kodiranjem. Nakon ove pozicije obrade, a_2 se smatra novim elementom početne slike a_0 za sledeće kodiranje.



Algoritam kodiranja za grupu 3



CCITT - 57570

Algoritam kodiranja za grupu 4

Realizacija algoritma u programskom jeziku javascript

Na sledećoj slici prikazan je kod koji demonstrira algoritam sa prethodne slike (Algoritam kodiranja za grupu 4 za dvodimenzionalni faksimil)

```
function compression() {
  while (position < input.length) {
    referentLine = input[position - 1];
    codingLine = input[position];
    console.log(referentLine);
    console.log(codingLine);
    a0 = findFirstWhite(codingLine);
    let lineResult = "";
    while (
      findFirstDifferent(referentLine, a0 + 1, codingLine[a0]) != "Not found"
    ) {
      a1 = findFirstDifferent(codingLine, a0, codingLine[a0]);
      b1 = findFirstDifferent(referentLine, a0 + 1, codingLine[a0]);
      b2 = findFirstDifferent(referentLine, b1, referentLine[b1]);
      let lengthBetween = null;
      let code = null;
      if (b2 < a1) {
        //pass mode
        lengthBetween = b2 - b1;
        if (codingLine[b1] == "B") {
          code = modeTable[0] + blackCodes[lengthBetween];
        } else {
          code = modeTable[0] + whiteCodes[lengthBetween];
        }
      }
      a0 = b2;
    }
  }
}
```



```

else {
    if (Math.abs(a1 - b1) <= 3) {
        //vertikalni
        if (a1 > b1) {
            lengthBetween = a1 - b1;
            code = modeTable[lengthBetween + 2] + whiteCodes[lengthBetween]; //+
            a0 = a1;
        } else if (a1 < b1) {
            lengthBetween = b1 - a1;
            code = modeTable[lengthBetween + 2 + 3] + blackCodes[lengthBetween];
            a0 = a1; }
    } else {
        //horizontalni
        let a2 = findFirstDifferent(codingLine, a1, codingLine[a1]);
        lengthBetween = a1 - a0;
        let lengthBetween2 = a2 - a1;
        code =
            modeTable[1] +
            whiteCodes[lengthBetween] +
            blackCodes[lengthBetween2];
        a0 = a2; }
    }
    lineResult += code;}
result += lineResult + EOL;
position++; }
return result; }

```

Primeri testiranja:

```
► (14) ["W", "W", "W", "W", "W", "W", "W", "W", "W", "W", "W", "W", "W", "W"]
► (14) ["B", "B", "B", "W", "W", "B", "B", "W", "W", "W", "W", "W", "W", "W"]
Obrada linije završena, pretvorena je u kod: 000000000001
► (14) ["B", "B", "B", "W", "W", "B", "B", "W", "W", "W", "W", "W", "W", "W"]
► (14) ["B", "B", "W", "W", "W", "W", "W", "W", "W", "W", "B", "B", "W", "W"]
a0 2
a1 10
b1 5
b2 7
pass mode
Lenghbetween: 2
Dodaje se 00010111
Obrada linije završena, pretvorena je u kod: 0001011100000000001
Kraj algoritma, rezultat je:
000000000001
00010111000000000001
```

Prilikom obrade treće linije je data situacija iz primera pod a) gde je detektovan pass mod jer kao što vidimo b2 je manje (levo) u odnosu na a1. U tom slučaju kodirana je dužina veličine 2 (između b1 i b2) i pretvorena u sekvencu 00010111 iza čega sledi sekvenca (000000000001) EOL (oznaka za kraj linije).

```

▶ (14) ["White", "White", "White", "White", "White", "White", "White", "White", "White", "White", "White", "White", "White", "White"]
▶ (14) ["White", "Black", "White", "White", "White", "White", "White", "White", "White", "Black", "Black", "White", "White", "White"]
Obrada linije završena, pretvorena je u kod: 000000000001
▶ (14) ["White", "Black", "White", "White", "White", "White", "White", "White", "White", "Black", "Black", "White", "White", "White"]
▶ (14) ["White", "White", "White", "White", "White", "Black", "Black", "Black", "Black", "Black", "Black", "Black", "White", "White"]
a0 0
a1 5
b1 1
b2 2
pass mode
Lengthbetween: 1
Dodaje se 0001000111
a0 2
a1 5
b1 9
b2 11
horizontal mode
a2: 12
Length between a2 a1 : 7
Lengthbetween: 3
Dodaje se 001100000011
Obrada linije završena, pretvorena je u kod: 0001000111001100000011000000000001
Kraj algoritma, rezultat je:
000000000001
0001000111001100000011000000000001

```

U ovom premeru identifikovani su pass mod i horizontalni mod. Na slici su zelenom bojom označeni jedinstveni kodni počeci za date modove.

Literatura

David Salomon. Data compression 3rd edition.

<https://drive.google.com/file/d/18qt5ZleRwHOOimXVtKVaxSE6lM1h0Di6/view>

International telecommunication union (ITU-T), Standardization of group 3 facsimile terminals for document transmission

International telecommunication union (ITU-T), Facsimile coding schemes and coding control functions for group 4 facsimile apparatus