

Pregled 2D verižnih kod

Borut Žalik, Damjan Strnad, Krista Rizman Žalik, Andrej Nerat,
Niko Lukač, Bogdan Lipuš, David Podgorelec

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko
E-pošta: borut.zalik@um.si

On 2D Chain Codes

The Freeman chain code in eight directions is the oldest, the best known, and the most frequently used 2D chain code. However, there exist other chain codes, which are, together with their features, surveyed briefly in this paper. Beside Freeman chain code in eight directions, also Freeman chain code in four directions, derivations of Freeman's chain codes, Vertex Chain Code, Three-Orthogonal Chain Code, Unsigned Manhattan Chain Code, and Mid-Crack Chain Code are presented and their features are exposed.

1 Uvod

Od uvedbe rastrskih naprav je primerna predstavitev geometrijskih objektov v rastrskem prostoru še kako pomembna. V mnogih primerih je dovolj, če opišemo samo mejo objekta, za kar lahko uporabimo koordinate mejnih pikselov ali, kar je učinkoviteje, zaporedje preprostih ukazov, ki nam povedo, kako se od podanega izhodiščnega piksla premikati po obodu objekta. Temu zaporedju ukazov pravimo verižna koda (angl. chain code). Verižne kode uporabljamo v raznih aplikacijah od razpoznavanja objektov [1] do stiskanja podatkov [2]. Človek ne vidi in ne pomni vzorcev fotonov na mrežnici, ki jih modelira tradicionalna rastrska slika, ampak so možgani sposobni v delčku sekunde segmentirati videno (sliko) v semantične komponente in jih povezati s predhodnimi zaznavami objektov. Zato je pogosto smiselno tudi rastrsko sliko predstaviti v segmentirani obliki, pri čemer lahko posamezne segmente učinkovito predstavimo z verižno kodo, morebitne barvne informacije pa shranjujemo ločeno za vsak segment.

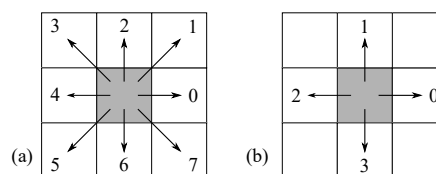
Verižne kode že same po sebi kompaktno predstavljajo rastrske objekte, saj potrebujejo do 3 bite na piksel, kar je občutno manj kot zapis piksla s koordinatami. Kljub temu pa verižne kode pogosto še dodatno stiskamo, pri čemer največkrat uporabljamo prilagoditve poznanih metod stiskanja. Statistično (Huffmanovo ali aritmetično) kodiranje je logična izbira, kadar so simboli abecede verižne kode izrazito neenakomerno porazdeljeni. Nadaljnje izboljšave rabe statističnih metod so bile dosežene z vplejavo posebnih kod za pogoste pare simbolov [3, 4] ali z rabo več statističnih modelov v t.i. kontekstnih pristopih [5]. Predlagane so bile tudi nestatistične metode stiskanja, temelječe na transformacijah nizov [2, 6]. Stiskanje

verižnih kod je običajno brezizgubno, poznamo pa tudi pristope z nadzorovanimi izgubami [4].

V tem prispevku bomo najpogostejše verižne kode jedrnato opisali in jih podkrepili s primeri.

2 Predstavitev 2D verižnih kod

2.1 8-smerna Freemanova verižna koda



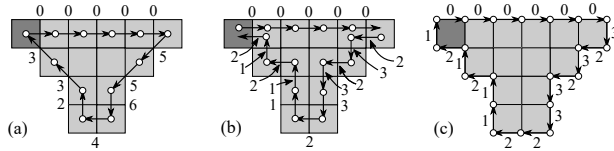
Slika 1: Ukazi verižnih kod F8 in F4

8-smerna Freemanova verižna koda (angl. Freeman chain code in eight directions – F8) je najstarejša verižna koda [7]. Iz referenčnega robnega piksla se lahko premaknemo v enega izmed osmih sosednjih pikselov. Abeceda kode ima 8 simbolov: $\Sigma(F8) = \{0, 1, 2, 3, 4, 5, 6, 7\}$ (slika 1a). Postopek tvorbe verižne kode F8 je sestavljen iz naslednjih korakov: izberemo začetni robni piksel in shranimo njegovi koordinati (x, y) . Odločimo se za smer potovanja po robnih pikslih (sourna ali protiurna), nato pa skonstruiramo kodo tako, da poiščemo sosednji robni piksel in zapišemo Freemanovo kodo premika. Koda, ki se premika skozi središča pikselov, pravimo središčna koda. Lastnosti verižne kode F8 so:

- položaj objekta je odvisen le od koordinat začetne točke verižne kode (ta lastnost je dejansko skupna vsem verižnim kodam),
- povečevanje objekta za faktor s , $s \in \mathbb{N}$, izvedemo tako, da vsak simbol F8 shranimo s -krat.
- zaporedje enakih vrednosti predstavlja zaporedje pikselov, ki si sledijo v ravni črti (vodoravni, navpični ali pod kotom $\pm 45^\circ$),
- če vsakemu simbolu F8 v zaporedju prištejemo vrednost naravnega števila n po modulu 8, objekt zavrtimo za $n \times 45^\circ$ v smeri, nasprotni urinemu kazalcu, in obratno, če vrednosti odštejemo, objekt zavrtimo v sourni smeri.

2.2 4-smerna Freemanova verižna koda

4-smerna Freemanova verižna koda (angl. Freeman chain code in four directions – F4) dovoli premik iz trenutnega piksla v sosednje piksle samo preko skupnih robov. Abeceda kode $\Sigma(F4) = \{0, 1, 2, 3\}$ je zato manjša (slika 1b). Konstrukcija kode F4 je enaka kot pri F8, tudi lastnosti so enake, le da lahko objekt zavrtimo le za kote $\pm 90^\circ$ in 180° .



Slika 2: verižne kode (a) F8, (b) F4 in (c) lomna koda

Kodo F4 lahko uporabimo na dva načina. V prvem načinu potujemo skozi središča pikslov tako kot pri F8, v drugem primeru pa po mejnih robovih pikslov. Takšni kodi pravimo lomna verižna koda (angl. crack chain code). Primer objekta, opisanega z različnimi Freemanovimi verižnimi kodami, vidimo na sliki 2.

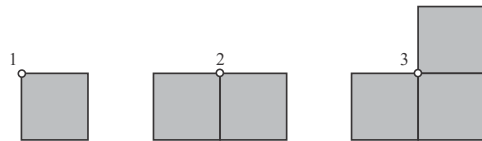
2.3 Izpeljanke Freemanove verižne kode

1. Freeman je predlagal tudi diferenčno verižno kodo (angl. Chain-Difference Coding – CDC) [8]. Vsak mejni piksel p_i (razen prvega p_0 in drugega p_1) zakodiramo z relativno razliko kotov $\delta = \angle(p_i - p_{i-1}, p_{i-1} - p_{i-2})$, relativne razlike pa zapišemo s Freemanovo abecedo $\Sigma(F8)$. Izkaže se, da so statistično najbolj verjetne razlike kotov 0° , ki jim sledijo koti $\pm 45^\circ$. To lastnost sta izkoristila Liu in Žalik [9], ki sta predlagala eno prvih metod za stiskanje verižnih kod, temelječo na Huffmanovih kodah.
2. Bribiesca je predstavil izpeljanko verižne kode F8, ki jo imenujemo usmerjena 8-smerna Freemanova verižna koda (angl. Directional Freeman Chain Code of eight directions, DF8) [10]. Simbole F8 je obravnaval kot števila, nato pa zaporedne vrednosti odštel. Če je vsota vrednosti pri sourni orientaciji 0, oz. pri protiumni -8, je verižna koda sklenjena.
3. Varianto verižne kode F4 je predstavil Nunes s sodelavci [11]. Predlagali so kodiranje relativne spremembe obhoda, kodo pa poimenovali verižna koda razlik (angl. Differential Chain Code, DCC). Abeceda $\Sigma(DCC) = \{R, L, S\}$, kjer R pomeni zasuk v desno, L , zasuk v levo in S nadalj v isti smeri. Koda DCC je lomna koda.
4. Žalik in Lukač sta s ciljem boljšega stiskanja verižne kode F8 predstavila kodirno shemo NAD (angl. Normalised Angle-Difference code) [6]. Abeceda ima štiri simbole, $\Sigma(NAD) = \{0, 1, 2, 3\}$, kjer simbol 3 označuje, da je sprememba kota, ki sledi, večja od $\pm 45^\circ$. Spremembe kotov so predstavljene s kodami NAD na naslednji način: $0^\circ = 0$, $45^\circ = 1$, $-45^\circ = 2$, $90^\circ = 300$, $-90^\circ = 310$, $135^\circ = 301$,

$-135^\circ = 311$, $180^\circ = 312$. Avtorja sta pokazala, da je to kodiranje bolj stisljivo.

2.4 Ogliščna verižna koda

Leta 1999 je Bribiesca izumil zanimivo verižno kodo, ki jo imenujemo ogliščna verižna koda (angl. vertex chain code – VCC) [12]. Koda VCC je definirana s številom robnih pikslov objekta v opazovanem oglišču, kot vidimo na sliki 3. Abeceda vozliščne kode VCC sestoji iz treh simbolov, $\Sigma(VCC) = \{1, 2, 3\}$. VCC je lomna koda in se vedno sklene. Zaporedje simbolov $\langle 1, 1, 1, 1 \rangle$ opiše objekt iz enega samega piksla, primer večjega objekta pa kaže slika 5a. Koda VCC je uporabna tudi, ko piksli niso



Slika 3: Simboli vozliščne kode VCC

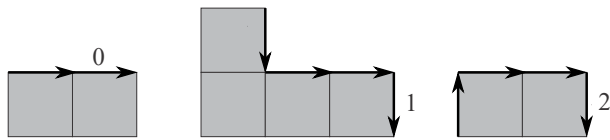
štirikotniki, a v tem primeru se spremeni tudi abeceda. Pri pikslih trikotne oblike je abeceda $\{1, 2, 3, 4, 5\}$, pri šestkotnih pa $\{1, 2\}$. Verižna koda VCC ima naslednje lastnosti:

- ravno (navpično ali vodoravno) črto karakterizira zaporedje simbolov 2,
- diagonalno črto pod kotom $\pm 45^\circ$ označuje izmenjujoče se zaporedje simbolov 1 in 3,
- koda VCC je neobčutljiva na vrtenje (za $\pm 90^\circ$ in 180°) in zrcaljenje,
- kodo VCC lahko normaliziramo, kar nam olajša preverjanje, ali dve verižni kodi VCC opisujeta enak objekt. Zaporedje kod vrtimo tako dolgo, da zaporedje simbolov predstavlja število z najmanjšo vrednostjo, kot vidimo na naslednjem primeru:
1311232121321213113312; zasuk v desno
3112321213212131133121; zasuk v desno
1123212132121311331213; normalizirana koda
- kodo VCC lahko enostavno stisnemo; ker je statistično najpogostejši simbol 2, ga predstavimo z enim bitom, ostala dva simbola pa z dvema.

2.5 Tri-ortogonalna verižna koda

Tri-ortogonalno verižno kodo (angl. Three Orthogonal chain code – 3OT) sta predstavila Sánchez-Cruz and Rodríguez-Díaz [13]. Abeceda $\Sigma(3OT) = \{0, 1, 2\}$ sestoji iz treh simbolov, katerih pomen je naslednji (slika 4):

- če je smer premika enaka, kot je bila predhodna smer premika, je koda 0,
- če se smer premika spremeni tako, da je enaka smeri pred predhodno spremembo, potem je koda 1,
- sicer je koda 2.



Slika 4: Simboli verižne kode 3OT

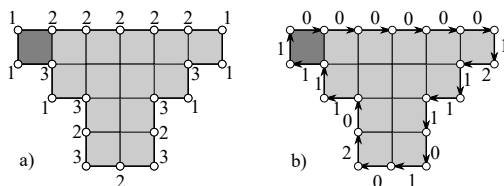
3OT je lomna koda, katere postopek konstrukcije je naslednji:

- izberemo ekstremno oglišče (na primer zgoraj levo),
- izberemo smer obhoda,
- prvi rob, ki ni vodoraven, dobi kodo 1,
- preostale robove zakodiramo s pravili 3OT.

Lastnosti kode 3OT so naslednje:

- ravno (navpično ali vodoravno) črto označuje zaporedje simbolov 0,
- diagonalno črto pod kotom $\pm 45^\circ$ predstavlja zaporedje simbolov 1,
- koda 3OT je neobčutljiva na vrtenje (za $\pm 90^\circ$ in 180°) in zrcaljenje.

Slika 5b kaže objekt, predstavljen s 3OT.



Slika 5: Geometrijski objekt, opisan z (a) VCC in (b) 3OT

2.6 Nepredznačena verižna koda Manhattan

Nepredznačena verižna koda Manhattan (angl. Unsigned Manhattan Chain Code – UMCC) je predstavljena v [14]. Koda opisuje premikanje po robnih pikslih ločeno v smereh x in y . Če od koordinat x in y opazovanega piksla odštejemo koordinati x in y njegovega predhodnika, dobimo predznačeno verižno kodo Manhattan (angl. Manhattan Chain Code – MCC). Abeceda verižne kode MCC je sestavljena iz treh simbolov, $\Sigma(MCC) = \{-1, 0, 1\}$. Na sliki 6a vidimo primer opisa meje geometrijskega objekta z MCC, kode pa povzema tabela 1.

Tabela 1: Verižna koda MCC za objekt iz slike 6

MCC_x	1	1	1	1	1	-1	-1	0	-1	0	-1	-1
MCC_y	0	0	0	0	0	-1	-1	-1	0	1	1	1

Ugotovimo, da nikoli ne moreta biti oba simbola verižne kode MCC po koordinatah x in y hkrati 0. Zato lahko $x = 0$ in $y = 0$ uporabimo za preklapljanje med predznaki, ki ji sledi par simbolov 0 ali 1, ki pove, v kateri

smeri se je spremenil predznak. Na primer: 00 10 pomeni, da spremenimo predznak kodi x , 00 01 spremenimo predznak kodi y , 00 11 pa obema kodama. Na ta način dobimo nepredznačeno verižno kodo Manhattan z abecedo iz dveh simbolov, $\Sigma(UMCC) = \{0, 1\}$. Verižno kodo UMCC za objekt s slike 6 vidimo v tabeli 2, pri čemer predpostavimo, da je začetni predznak v obeh smereh pozitiven. Poudarjeni elementi so elementi, s katerimi preklapljammo predznak.

Tabela 2: Verižna koda UMCC za objekt iz slike 6

$UMCC_x$	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1
$UMCC_y$	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1

Kodo lahko še nekoliko zmanjšamo. Namreč, v smeri, ki smo jo označili z 1 in pomeni spremembo predznaka, vemo, da bo naslednji simbol 1, zato ga pri zapisu lahko izpustimo. Rezultat vidimo v tabeli 3, kjer simbol '□' označuje mesto izpuščenega simbola.

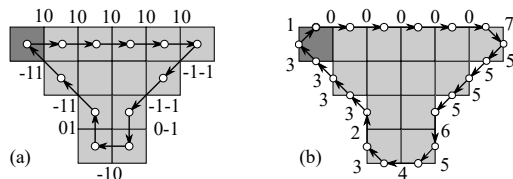
Tabela 3: Reducirana verižna koda UMCC za objekt iz slike 6

$UMCC_x$	1	1	1	1	1	0	1	□	1	0	1	0	0	1	1
$UMCC_y$	0	0	0	0	0	0	1	□	1	1	0	0	1	□	1

UMCC ima naslednje lastnosti [14]:

- Povečevanje objekta za faktor s izvedemo tako, da vsak simbol, s katerim ne spreminjamo predznaka, shranimo s -krat.
- Kot smo omenili, predpostavimo, da je začetna vrednost koordinat x in y pozitivna. Vrtenje za 180° dosežemo, če pričnemo obhod v smereh $-x$ in $-y$, vrtenje za 90° pa z zamenjavo vrednosti $UMCC_x$ in $UMCC_y$. Za vrtenje za 270° zamenjamo vrednosti $UMCC_x$ in $UMCC_y$ in pričnemo z obhodom v smereh $-x$ in $-y$.
- Zrcaljenje glede na os Y dobimo s spremembo smeri obhoda v smeri x in, analogno, s spremembo smeri obhoda v smeri y objekt prezrcalimo preko osi X .
- Vodoravne in navpične črte zaznamo kot zaporedje enakih simbolov v $UMCC_x$, oziroma $UMCC_y$.
- Zaporedje simbolov 1 v $UMCC_x$ in $UMCC_y$ določa diagonalne črte.
- Normalizacijo verižne kode dosežemo z naslednjim postopkom:
 - začasno odstranimo vse simbole za spremembo predznaka;
 - vrtimo simbole v $UMCC_x$, dokler $UMCC_x$ ne predstavlja najmanjše vrednosti, pri čemer označimo število operacij vrtenja s k_x ;
 - analogno poiščemo vrednost k_y za $UMCC_y$;
 - določimo $k = \min\{k_x, k_y\}$;

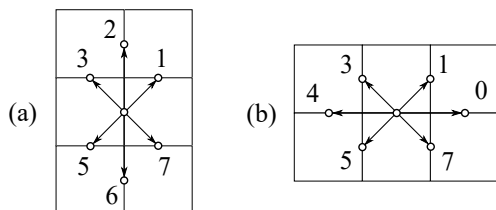
- simbole v izvornih zaporedjih $UMCC_x$ in $UMCC_y$ z odstranjenimi znaki za nadzor predznaka zavrtimo k -krat;
 - nazadnje na ustrezna mesta vstavimo simbole za spremembo predznaka.
- Pri verižni kodi UMCC enostavno določimo tudi monotone dele geometrijskega objekta, in sicer, deli med simboli, ki nadzirajo predznak v $UMCC_x$, so monotoni glede na os X , in analogno, deli, ki so med dvema simboloma za spremembo predznaka v $UMCC_y$, so monotoni glede na os Y .



Slika 6: Geometrijski objekt, opisan s (a) predznačeno verižno kodo Manhattan in (b) s središčno-lomno verižno kodo

2.7 Središčno-lomna verižna koda

Središčno-lomno (angl. Mid-Crack - MC) verižno kodo sta predlagala Shih in Wong [15] s ciljem bolje oceniti lastnosti izvornega nerasteriziranega objekta, kot sta obseg in ploščina. MC kombinira središčno verižno kodo F8 z lomno kodo F4, pri tem pa povezuje središčne točke robov pikselov. Abeceda $\Sigma(MC) = \Sigma(F8)$, pri čemer pri navpičnih robovih pikselov ne dovolimo simbolov 0 in 4 (slika 7a), pri vodoravnih pa ne 2 in 6 (slika 7b). Opis objekta z verižno kodo MC vidimo na sliki 6b. Lastnosti središčno-lomne verižne kode so enake kodi F8.



Slika 7: Simboli središčno-lomne verižne kode

3 Zaključek

Ker je obe Freemanovi verižni kodi (8-smerno in 4-smerno) najlažje razumeti in implementirati, ju v praksi tudi najpogosteje srečamo. F4 lahko uporabimo kot središčno ali lomno kodo. V tem članku smo želeli opozoriti, da obstajajo tudi druge verižne kode, katerih lastnosti bi lahko bile v dani situaciji primernejše. Pri prav vseh verižnih kodah lahko zaznamo ravne črte, med vodoravnimi in navpičnimi črtami pa ne moremo ločiti pri kodah VCC in 3OT. Diagonalno črto lahko zaznamo z vsemi verižnimi kodami. Neobčutljive na vrtenje in zrcaljenje so kode VCC, 3OT in UMCC, vrtenje vsaj za kote $\pm 90^\circ$ in 180° omogočajo vse kode razen 3OT in VCC. Vse kode razen

VCC in 3OT omogočajo skaliranje. Monotone dele geometrijskega objekta enostavno zazna koda UMCC, koda MC pa daje najboljši približek dejanskega obsega in ploščine izvornega geometrijskega objekta. Če želimo verižne kode dodatno stisniti, je najbolj stisljiva verižna koda 3OT [2].

Zahvala

Projekt (Posplošene simetrije in ekvivalence v geometrijskih podatkih, št. N2-0181) je sofinancirala Javna agencija za raziskovalno dejavnost Republike Slovenije iz državnega proračuna.

Literatura

- [1] N.B. Boodoo-Jahangeer, S. Baichoo, Face Recognition Using Chain Codes, *Journal of Signal and Information Processing* 4(3B), 2013, 154–157.
- [2] B. Žalik, D. Mongus, N. Lukač, K.R. Žalik, Efficient chain code compression with interpolative coding, *Information Sciences* 439–440, 2018, 39–49.
- [3] Y.-K. Liu, W. Wei, P.-J. Wang, B. Žalik, Compressed vertex chain codes. *Pattern Recognition*, 40(11), 2007, 2908–2913.
- [4] Y.-K. Liu, B. Žalik, P.-J. Wang, D. Podgorelec, Directional difference chain codes with quasi-lossless compression and run-length encoding, *Signal Processing: Image Communication*, 27(9), 2012, 973–984.
- [5] A. Akimov, A. Kolesnikov, P. Franti, Lossless compression of map contours by context tree modeling of chain codes, *Pattern Recognition*, 40(3), 2007, 944–952.
- [6] B. Žalik, N. Lukač, Chain code lossless compression using move-to-front transform and adaptive run-length encoding, *Signal Processing Image Communication* 29(1), 2014, 96–106.
- [7] H. Freeman, On the Encoding of Arbitrary Geometric Configurations, *IRE Transactions on Electronic Computers*, EC-12(2), 1961, 260–268.
- [8] H. Freeman, Computer processing of line drawing, *ACM Computing Surveys*, 6(1), 1974, 57–97.
- [9] Y.-K. Liu, B. Žalik, An efficient chain code with Huffman coding, *Pattern Recognition*, 38(4), 2005, 553–557.
- [10] E. Bribiesca, A geometric structure for two-dimensional and three-dimensional surfaces, *Pattern Recognition*, 25(5), 1992, 483–496.
- [11] P. Nunes, F. Pereira, F. Marqués, Multi-grid chain coding of binary shapes, *ICIP'97 Proceedings of the 1997 International Conference on Image Processing*, 3, 1997, 114–117.
- [12] E. Bribiesca, A new chain code, *Pattern Recognition*, 32(2), 1999, 235–251.
- [13] H. Sánchez-Cruz, R. M. Rodríguez-Díaz, Compressing bi-level images by means of a 3-bit chain code, *SPIE Optical Engineering*, 44(9), 2005, 1–8.
- [14] B. Žalik, D. Mongus, Y.-K. Liu, N. Lukač, Unsigned Manhattan Chain Code, *Journal of Visual Communication and Image Representation*, 38(C), 2016, 186–194.
- [15] F. Y. Shih, W.-T. Wong, A new single-pass algorithm for extracting the Mid-Crack Codes of Multiple Regions, *Journal of Visual Communication and Image Representation*, 3(3), 1992, 217–224.