

OSNOVE ROBOTIKE
TEMA: ROBOTSKE VIZIJE

ROBOTSKE VIZIJE

NAPOMENA: OVO JE BETA TEKST, TJ. TEKST SEMINARA KOLEGE
KRESIMIRA PETRINECA. OPISANI ALGORITMI SU PROVJERENI U
MATLABU TE U PRAKTIČNOJ REALIZACIJI KOLEGE HRVOJA ŠAGOVCA.

PROF.DR.SC. ZDENKO KOVAČIĆ

Krešimir Petrinec

09.07.1998.

Robotska vizija

Moderna robotska postrojenja često sadrže vanjske senzore koji daju robotu informaciju o objektima u okolini. Jedan od najmoćnijih senzora po broju informacija koje proizvodi je sustav robotske vizije. Osnovni sustav robotske vizije sadrži jednu stacionarnu kameru postavljenu iznad radnog prostora. U nekim slučajevima sadrži i više stacionarnih kamera kako bi dobili višestruku perspektivu. No moguće je postići i ne ograničen broj perspektiva sa samo jednom pokretnom kamerom. To je tzv. aktivni oblik osjeta u kojem robot sam pozicionira senzor na temelju rezultata prethodnih mjerenja.

Sustav robotske vizije sadrži korisne informacije za automatizaciju manipulacije objekata. Korištenjem robotske vizije mogu se odrediti pozicija, orijentacija, identifikacija i stanja objekata u prostoru. U ovom tekstu biti će razmatran osnovni sustav robotske vizije sa samo jednom stacionarnom kamerom smještenom okomito iznad radnog prostora. U ovom tekstu se pretpostavlja da kamera daje crno-bijelu sliku u binarnom zapisu.

Robotskom vizijom moguće je sustave učiniti robustnijim, pouzdanijim, a ponekad i jeftinijim.

1. Prikaz slike

Dvodimenzionalna slika se može promatrati kao analogna funkcija $i(x,z)$ koja predstavlja intenzitet svjetla na svakoj koordinati (x,z) . Takva analogna funkcija se kvantizira, odnosno x -os se podijeli na m uzoraka, a y -os na n uzoraka tako da slika poprimi rezoluciju od $m \times n$ točaka. Najčešće se uzima da su m i n potencije broja 2. Intenzitet svjetla koji odgovara točki u k -tom redu i j -tom stupcu je srednja vrijednost intenziteta na tom djeliću slike i iznosi:

$$I_a(k,j) \triangleq \frac{\int_0^{\Delta x} \int_0^{\Delta y} i[(k-1)\Delta x + x, (j-1)\Delta y + y] dy dx}{\Delta x \Delta y} \quad (1)$$

Intenzitet se također kvantizira s preciznošću od b bita. Rezultat je 2^b mogućih vrijednosti intenziteta, tj. nijansi sive boje. Veličina prostora potrebnog za pohranu slike u memoriji računala ovisi o rezoluciji slike i o rezoluciji intenziteta, tako da je za sliku rezolucije 1024×1024 s 256 nijansi sive boje potreban 1 Mbyte memorijskog prostora.

2. Uspoređivanje uzoraka

Jedna od važnih primjena robotske vizije je utvrđivanje da li je dani predmet pripadnik određene grupe predmeta. Najjednostavniji način je tzv. uspoređivanje uzoraka (eng. template matching). Predmeti ili dijelovi koji su karakteristični za određenu grupu predmeta stavljaju se jedan po jedan pred kameru, snimaju i pohranjuju u biblioteku dijelova:

$$LIB = \{T_i(k,j): 1 \leq k \leq m, 1 \leq j \leq n, 1 \leq i \leq N\}. \quad (2)$$

$T_i(k,j)$ je i -ti reprezentativni uzorak ili maska za i -tu grupu predmeta. Na primjer, jednostavan uzorak romba 3×3 prikazan je na slici 1. Vidimo da je romb svijetliji s desne strane.

j

→

k

↓

0	4	0
3	0	5
0	4	0

Slika 1. Uzorak romba 3×3

Postoje razni načini traženja zadanog uzorka $T_i(k,j)$ u slici $I(k,j)$ dimenzija $m \times n$. Kako je uzorak obično manji od slike, vršimo pretraživanje slike s lijeva na desno od vrha prema dnu slike.

Translacija uzorka se kreće po koordinatama: $0 \leq x \leq m - m0$ i $0 \leq y \leq n - n0$, gdje su $m0$ i $n0$ dimenzije uzorka. Za svaku translaciju se računa razlika između slike $I(k,j)$ i uzorka $T_i(k,j)$ prema formuli:

$$\rho_i(x,y) \triangleq \sum_{k=1}^{m0} \sum_{j=1}^{n0} |I(k+x, j+y) - T_i(k,j)|, \quad 1 \leq i \leq N \quad (3)$$

Vidimo da je $\rho_i(x,y) \geq 0$ te ako je uzorak $T_i(k,j)$ nađen onda je $\rho_i(x,y) = 0$. Pronalaskom uzorka utvrđena je i njegova pozicija. Jednostavan algoritam usporedbe uzoraka glasi:

1. $i = 0, x = 0, y = 0, \varepsilon > 0$, nađen = istna.
2. izračunaj $\rho_i(x,y)$ prema formuli (3).
3. ako je $\rho_i(x,y) \leq \varepsilon$, stop.
4. $x = x + 1$. Ako je $x \leq m - m0$, idi na 2. korak.
5. $x = 0, y = y + 1$. Ako je $y \leq n - n0$, idi na 2. korak.
6. $x = 0, y = 0, i = i + 1$. Ako je $i \leq N$, idi na 2. korak.
7. nađen = laž.

Primjer 1. U slici 2., dimenzija 4×5 potrebno je pronaći romb sa slike 1.

	Y \longrightarrow				
X \downarrow	2	1	0	0	3
	0	0	5	0	0
	0	4	0	6	0
	1	0	5	0	0

Slika 2. Slika 4×5

U ovom primjeru moguće translacije su: $0 \leq x \leq 1$ i $0 \leq y \leq 2$. Računamo jednadžbu (3):

$$\begin{aligned} \rho(0,0) &= 8, & \rho(0,1) &= 32, & \rho(0,2) &= 16, \\ \rho(1,0) &= 30, & \rho(1,1) &= 4, & \rho(1,2) &= 32. \end{aligned}$$

Vidimo da je najbolje preklapanje $\rho(1,1) = 4$, ali pošto je $\rho(1,1) > 0$ onda nije idealno.

U primjeru 1. vidimo da preklapanje ne mora biti idealno. Ograničenje je da slike koje se uspoređuju moraju imati jednake srednje intenzitete, odnosno slike mogu imati različitu pozadinsku rasvjetu. Srednji intenziteti se definiraju na slijedeći način:

$$\|T_i\| \triangleq \left[\sum_{k=1}^{m0} \sum_{j=1}^{n0} T_i^2(k,j) \right]^{1/2}, \quad (4)$$

$$\|I_{x,y}\| \triangleq \left[\sum_{k=1}^{m0} \sum_{j=1}^{n0} I^2(k+x, j+y) \right]^{1/2}. \quad (5)$$

Osjetljivost na promjene pozadinskog svijetla može se ukloniti normiranom međukorelacijskom funkcijom:

$$\sigma_i(x,y) \triangleq \frac{\sum_{k=1}^{m0} \sum_{j=1}^{n0} I(k+x, j+y) T_i(k,j)}{\|I_{x,y}\| \cdot \|T_i\|}. \quad (6)$$

Primjer 2. Normirana međukorelacijska funkcija (eng. normalized cross correlation)

Računamo li vrijednosti normirane međukorelacijske funkcije za primjer 1. dobit ćemo slijedeće vrijednosti:

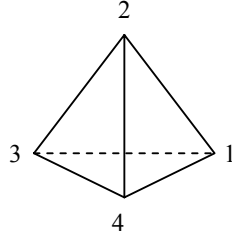
$$\begin{aligned} \sigma(0,0) &= 0.817, & \sigma(0,1) &= 0.000, & \sigma(0,2) &= 0.557, \\ \sigma(1,0) &= 0.000, & \sigma(1,1) &= 0.999, & \sigma(1,2) &= 0.000. \end{aligned}$$

Vidimo da se najbolje poklapanje pojavljuje za $\sigma(1,1) = 0.999$. Ovdje je $\sigma(1,1)$ vrlo blizu jedinici što znači da je nađeno vrlo dobro poklapanje.

Osim problema sa srednjim intenzitetom svjetla, ova metoda je osjetljiva na rotacije, skaliranja i promjene perspektive promatranog uzorka te na varijacije unutar zadane grupe predmeta gdje svi uzorci ne izgledaju kao reprezentativni uzorak ili maska.

3. Poliedarski oblici

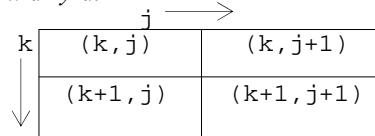
Kako roboti manipuliraju trodimenzionalnim objektima pogodno je razviti metode koje djelotvorno predstavljaju takve objekte. Primjer poliedra prikazan je na slici 3. Tetraedar je najjednostavniji trodimenzionalni objekt jer ima samo četiri trokutaste plohe.



Slika 3. Tetraedar

3.1. Detekcija rubova (eng. edge detection)

Pretpostavimo da su strane poliedra glatke, homogene i neprozirne te da je svjetlo uniformno i bez sjena. Pod tim uvjetima će intenzitet reflektiranog svjetla biti konstantan ili će se kontinuirano mijenjati duž svake plohe objekta. Nadalje, intenzitet reflektiranog svjetla će se mijenjati skokovito na rubovima susjednih ploha. Prema tome, na rubovima poliedričkih objekata gradijent intenziteta svjetla, $\nabla I(x,y)$, biti će beskonačan. Gradijent $\nabla I(x,y)$ je jednostavna 1×2 Jacobian matrica parcijalnih derivacija od $I(x,y)$ po x -u i y -u.



Slika 4. Točke korištene prilikom aproksimacije gradijenta intenziteta

Digitalna aproksimacija operatora gradijenta neće biti ništa drugo nego računanje srednje vrijednosti promjene intenziteta svjetla između dva susjedna para točaka i to se računa jednom u smjeru x -osi i jednom u smjeru y -osi:

$$\nabla I_1(k, j) \triangleq \frac{[I(k+1, j) - I(k, j)] + [I(k+1, j+1) - I(k, j+1)]}{2\Delta x}, \quad (7)$$

$$\nabla I_2(k, j) \triangleq \frac{[I(k, j+1) - I(k, j)] + [I(k+1, j+1) - I(k+1, j)]}{2\Delta y}. \quad (8)$$

Detekcija ruba objekta se svodi na računanje veličine gradijenta intenziteta i to na slijedeći način:

$$\|\nabla I(k, j)\| = \sqrt{\nabla I_1^2(k, j) + \nabla I_2^2(k, j)}. \quad (9)$$

Rub se nalazi na točki (k, j) ako je veličina $\|\nabla I(k, j)\|$ veća od $\epsilon > 0$ koji se odabire empirijski. On mora biti dovoljno mali da pokupi sve rubove, ali mora biti i dovoljno veliki da izbjegne lažne rubove prouzročene smetnjama u slici. Nesrazmjer između dužine točaka po x i y osi naziva se *aspect ratio* i računa se kao:

$$\rho_{xy} \triangleq \frac{\Delta x}{\Delta y}. \quad (10)$$

Ako ρ_{xy} nije jednak jedinici tada krug na analognoj slici prelazi u elipsu na digitalnoj slici.

Algoritam detekcije rubova:

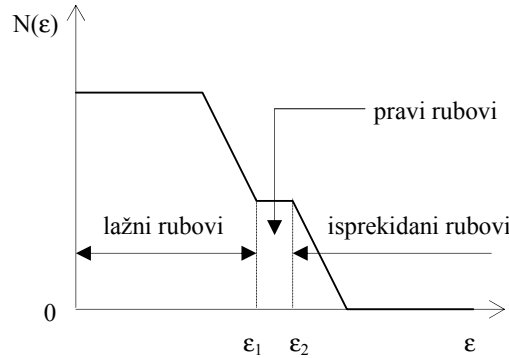
1. $k = 1, j = 1, \epsilon > 0$.
2. Izračunaj $\|\nabla I(k, j)\|$ koristeći jednadžbu (9).

3. Ako je $\| \nabla I(k, j) \| \geq \epsilon$, $L(k, j) = 1$; inače $L(k, j) = 0$;
4. $j = j + 1$, ako je $j < n$ idi na korak 2.
5. $L(k, n) = 0$.
6. $j = 1$, $k = k + 1$, ako je $k < m$ idi na korak 2.
7. For $j = 1$ to n , $L(m, j) = 0$.

Algoritam detekcije rubova uzima $m \times n$ sliku s nijansama sive boje $I(k, j)$ te stvara binarnu sliku $m \times n$ $L(k, j)$ u kojoj jedinice predstavljaju rubove. Ključni parametar u algoritmu je ϵ , on se postavlja empirijski, odnosno može predstavljati izlaz iz nekog drugog algoritma. Kod odabira ϵ korisno je razmotriti slijedeću funkciju raspodjele rubnih točaka:

$$N(\epsilon) \triangleq \sum_{k=j}^{m-1} \sum_{j=1}^{n-1} L(k, j). \quad (11)$$

$N(\epsilon)$ predstavlja ukupan broj rubnih točaka nađenih u slici $I(k, j)$. Kada je $\epsilon=0$ algoritam predstavlja sve točke rubnim točkama (slika 5.). Povećavamo li ϵ , broj lažnih rubova se smanjuje, da bi za mali opseg $[\epsilon_1, \epsilon_2]$ algoritam dao relativno dobru sliku rubova $L(k, j)$. Daljnjim povećanjem ϵ algoritam postaje neosjetljiv na rubove.



Slika 5. Funkcija raspodjele rubnih točaka

Kada se utvrde rubne točke tada se one mogu nadopuniti ravnim linijama. Sjecišta tih linija daju skup od M vrhova. A kako smo pretpostavili da je objekt poliedar, tada je on u potpunosti određen sa svojim vrhovima.

Možemo uočiti da smo originalnu informaciju sadržanu u $m \cdot n$ točaka reducirali na $2 \cdot M$ cjelobrojnih koordinata koje određuju vrhove poliedra.

3.2. Kutne točke (eng. corner points)

Ako se slika $I(k, j)$ sastoji od samo dvije sive boje, tada je ona binarna odnosno crno-bijela slika. Kod takvih slika moguće je direktno detektirati kutne točke. Detekcija se vrši usporedbom uzoraka kutnih točaka 3×3 ili maskama koje predstavljaju sve vrste kuteva, koji su barem dvije točke dugi. Takvih uzoraka ili maski ima ukupno osam (slika 6.).

0 1 1	1 1 1	1 1 0	1 0 0	0 0 0	0 0 0	0 0 0	0 0 1
0 1 1	0 1 0	1 1 0	1 1 0	1 1 0	0 1 0	0 1 1	0 1 1
0 0 0	0 0 0	0 0 0	1 0 0	1 1 0	1 1 1	0 1 1	0 0 1

Slika 6. Uzorci kutnih točaka

Da ne uspoređujemo sve točke s uzorcima kuteva, dovoljno je prvo naći točke koje su kandidati, a tada ih ispitati.

Na isti način možemo definirati uzorke za rupe u objektima ili krajeve linija, te ih na isti način pronaći u slikama.

3.3. Kodiranje slike brojanjem istovrsni točaka (eng. run-length encoding)

Slike velike rezolucije sadrže puno točaka zbog čega zauzimaju puno memorijskog prostora i prijenos preko komunikacijskog kanala traje dugo. Jedna od metoda za kompresiju slika, ali s malim nijansi sive boje, je pakiranje nekoliko točaka u jedan byte. Na primjer, u 1 byte stanu 2 točke s 16 nijansi sive boje. Nadostatak takve metode je to što treba sliku pakirati prije spremanja i otpakirati prije upotrebe. Alternativna metoda je *run-length* metoda. *Run* je niz susjednih točaka koje imaju istu vrijednost, a *length* je dužina, odnosno broj točaka u tom nizu. Umjesto da spremamo sve točke, spremamo prvu i broj koliko se je puta za redom pojavila. Dalje ponavljamo postupak. Ovisno o slici, run-length metodom moguće je reducirati veličinu potrebnog prostora za jedan ili više redova veličine.

Primjer 3. Kodiranje slike 7. run-length metodom.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	0
0	0	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0
0	0	0	1	1	1	1	0	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Slika 7. Zadana slika

Zadana slika 12×18 može se prikazati na sljedeći način:

$$\Gamma = [0, 44, 6, 10, 9, 7, 4, 3, 4, 5, 5, 5, 3, 7, 5, 1, 5, 8, 9, 12, 3, 16, 1, 44]^T$$

Nepakirana binarna slika zahtjeva $m \cdot n = 216$ bytova, pakirana binarna slika 27 bytova, a kodirana run-length metodom 24 bytova. Run-length kod Γ bio bi i kraći da slika nema rupu u sredini, dok bi binarna pakirana slika i dalje zahtjevala 27 bytova.

4. Analiza likova (eng. shape analysis)

Ako dijelovi koje treba ispitati ili kojima treba manipulirati nisu poliedarski oblici, tada nas opis vrhovima više ne zadovoljava. Rubovi oblika mogu biti neke krivulje, tako da nam trebaju drugi načini opisa tih dijelova.

4.1. Opis linija (eng. line descriptors)

Umjesto da pamtimo niz koordinata koje čine liniju, možemo upamtiti prvu koordinatu i inkrementalne promjene potrebne za generiranje uzastopnih točaka na krivulji. Ova metoda naziva se *lančano kodiranje* (eng. *chain coding*). Na slici 8. prikazana je točka p i njenih osam susjeda.

3	2	1
4	p	0
5	6	7

Slika 8. Lančani kod susjednih točaka točke p

Za označavanje susjednih točaka koriste se cijeli brojevi od 0 do 7 počevši od istočnog prema ostalima suprotno od kazaljke na satu. Lančano kodiranje je puno efikasnije od pamćenja koordinata točaka jer uzima samo tri bita po točki. Druga prednost je relativna prezentacija krivulja, dok je lista kooordinata apsolutna. Naime, dvije krivulje istog oblika, ali započete u različitim koordinatama generirati će identičan lančani kod.

Primjer 4. Lančani kod.

Na slici 9. jedinicama je označen objekt koji treba prikazati u obliku lančanog koda.

$$\begin{array}{cccccccc}
 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 8 \downarrow & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{array}$$

Slika 11. Binarna slika 8×10 s područjem R.

Prema definiciji (12) možemo izračunati momente:

$$\begin{aligned}
 m_{00} &= 9, \\
 m_{01} &= 45, & m_{10} &= 36, \\
 m_{02} &= 255, & m_{11} &= 192, & m_{20} &= 150.
 \end{aligned}$$

Prema (13) možemo izračunati slijedeće:

$$\begin{aligned}
 A &= m_{00} = 9 \\
 x_c &= m_{10} / m_{00} = 4 \\
 y_c &= m_{01} / m_{00} = 5
 \end{aligned}$$

Ako pogledamo sliku 11. vidjeti ćemo da se područje R uistinu sastoji od $A = 9$ točaka, centar je u točki $(4, 5)$.

Kad je jednom određeno $\{x_c, y_c\}$ području R, tada se može preći na računanje centralnih momenata.

Definicija centralnog momenta. Neka (x_c, y_c) predstavlja centar područja R, i neka (x, y) predstavlja redak i stupac točke u području R. Tada se centralni momenti definiraju na slijedeći način:

$$\mu_{kj} \triangleq \sum_{(x,y) \in R} (x - x_c)^k (y - y_c)^j, \quad k \geq 0, j \geq 0. \quad (14)$$

Pokazuje se da se translacijom R-a ne mijenja centralni moment R-a. Odnosno, za svaki R biti će $\mu_{10} = 0$ i $\mu_{01} = 0$.

Primjer 6. Centralni momenti.

Za primjer 5 (slika 11.) potrebno je izračunati momente inercija.

$$\begin{aligned}
 \mu_{00} &= 9, \\
 \mu_{01} &= 0, & \mu_{10} &= 0, \\
 \mu_{02} &= 30, & \mu_{11} &= 12, & \mu_{20} &= 6.
 \end{aligned}$$

Momenti μ_{02} i μ_{20} predstavljaju *momente inercije*, a μ_{11} je moment produkta. Centralni momenti se mogu normirati tako da poprime još jedan oblik neovisnosti.

Definicija normiranog centralnog momenta. Neka je $\{\mu_{kj}\}$ centralni moment od R. Tada se normirani centralni moment definira na slijedeći način:

$$v_{kj} \triangleq \frac{\mu_{kj}}{\mu_{00}^{(k+j+2)/2}}, \quad k \geq 0, j \geq 0. \quad (15)$$

Pokazuje se da se normirani centralni moment ne mijenja skaliranjem područja R.

4.3. Kut otklona

Kutom otklona može se postići neovisnost područja R na rotacije.

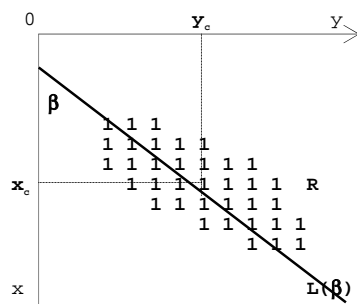
Definicija kuta otklona (eng. principal angle). Neka je $\{\mu_{kj}\}$ centralni moment od R. Tada se kut otklona definira na slijedeći način:

$$\phi \triangleq \frac{1}{2} \arctan 2(\mu_{11}, \mu_{20} - \mu_{02}). \quad (16)$$

Funkcija $\text{atan2}(y, x)$ definira se na slijedeći način:

1. ako je $x > 0$, tada je $\text{atan2}(y, x) = \arctan(y/x)$,
2. ako je $x = 0$, tada je $\text{atan2}(y, x) = [\text{sgn}(y)] \cdot \pi/2$,
3. ako je $x < 0$, tada je $\text{atan2}(y, x) = \arctan(y/x) + [\text{sgn}(y)] \cdot \pi$.

Fizička interpretacija načelnog kuta vezana je uz os $L(\beta)$ prikazanu na slici 12. Moment inercije područja R oko osi $L(\beta)$ ovisiti će o kutu β . Kut kod kojeg je moment inercije najmanji je kut otklona i iznosi $\beta = \phi$. Slijedi da je kut otklona dobro definiran za izdužene objekte, a neodređen za kružne oblike.



Slika 12. Načelni kut područja R.

Primjer 7. Kut otklona.

Potrebno je izračunati kut otklona za sliku 11, primjer 5.

Centralni momenati potrebni za izračunavanje kuta otklona izračunati su u primjeru 6.

Iz jednadžbe (16) slijedi da je kut otklona jednak:

$$\phi = \frac{a \tan 2(24, -24)}{2} = \frac{3\pi}{8} = 67.5^\circ$$

Za kut otklona ϕ , možemo prvo translirati R za $(-x_c, -y_c)$, a zatim rotirati R za kut $-\phi$. Tada će normirani momenti biti neovisni o translacijama, rotacijama i skaliranjima.

Literatura:

- [1] R.J.Schilling: Fundamentals of Robotics, Prentice-Hall Inc, Englewood Cliffe, New Jersey, 1990.
- [2] H. Šagovac: Upravljanje robotom primjenom kamere, Diplomski rad broj 966, Fakultet elektrotehnike i računarstva, Zagreb, 1996.