

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 448

**RASPOZNAVANJE PROMETNIH ZNAKOVA
KRIŽNOM KORELACIJOM**

Dominik Rojković

Zagreb, siječanj 2009

Izvornik zadatka završnog rada

Zahvala:

Zahvaljujem se mentoru doc. dr. sc. Siniši Šegviću na korisnim savjetima, razumijevanju i pomoći koju mi je pružio pri izradi ovog rada te na velikom trudu koji je nesebično uložio.

Od srca se zahvaljujem svojoj obitelji i mojoj najvećoj podršci - Jeleni...

Također se zahvaljujem svima ostalima koji su mi pomogli na bilo koji način oko ovog rada.

Sadržaj

1	Uvod.....	3
1.1	Umjetna inteligencija	4
1.2	Računalni vid.....	4
2	Križna korelacija u raspoznavanju objekata	6
2.1	Definicija križne korelacije	6
2.2	Upotreba križne korelacije u raspoznavanju objekata	11
3	Normalizacija veličine slike bilinearnom interpolacijom	13
3.1	Definicija bilinearne interpolacije	13
3.2	Upotreba bilinearne interpolacije u normalizaciji veličine slike	14
4	Programska izvedba	17
4.1	Programsko okruženje	17
4.1.1	Parametri komandne linije	18
4.1.2	Formati ulaznih datoteka	20
4.1.3	Interaktivne naredbe ljske	20
4.2	Integracija algoritma za raspoznavanje prometnih znakova u programsko okruženje i njegovi glavni dijelovi.....	21
4.2.1	Opis algoritma.....	21
4.3	Korištene biblioteke i razvojni alati	22
5	Eksperimentalni rezultati	24
5.1	Primjeri rada algoritma	24
5.1.1	Slike izlučene iz videa snimljenog u automobilu.	25
5.1.2	Slike snimljene mobilnim telefonom.....	28
5.1.3	Slike idealnih znakova	34

6	Zaključak	35
	Literatura	36
	Sažetak	38
	Summary	39
	Privitak	40

1 Uvod

Zamislimo automobil koji bi imao sposobnost vožnje zadanim putem pod upravljanjem računala. Prednost takvog vozila je što ne bi zahtijevalo osobu u ulozi vozača te njegova sigurnost ne bi ovisila o vozačevim sposobnostima i njegovom trenutnom stanju.

Takvo vozilo trebalo bi imati sposobnost „gledanja“, tj. praćenja situacije u prometu. Vozilo bi pratilo situaciju snimanjem situacije ispred sebe te donošenjem odluka o daljnjem tijeku upravljanja na temelju informacija koje bi dobivalo ugrađeno računalo.

Jedna od najvažnijih informacija koja bi se dobivala iz slijeda slika snimanjem situacije ispred sebe su informacije koje pružaju prometni znakovi. Stoga, program koji bi pokretao vozilo mora uključivati algoritam za *prepoznavanje* i *raspoznavanje* prometnih znakova duž ceste.

U ovom radu bit će razrađen jedan od algoritama za *raspoznavanje* prometnih znakova u slijedu slika pribavljenih iz vozila u pokretu, te njegova programska implementacija. Razmatra se raspoznavanje znakova pod pretpostavkom da su položaj u slici i veličina prethodno već otkriveni nekom drugom metodom. Rad se fokusira na raspoznavanje primjenom križne korelacije, uz korištenje postojeće programske infrastrukture za računalni vid.

Područje kojem pripada takav algoritam zove se *računalni vid*. Računalni vid je podskup *umjetne inteligencije*. Stoga u ostatku uvodnog dijela uvodimo pojmove *umjetna inteligencija* i *računalni vid*.

1.1 Umjetna inteligencija

Umjetna inteligencija (*artificial intelligence*) je grana računalnih znanosti koja se bavi proučavanjem i oblikovanjem računalnih sustava koji pokazuju neki oblik inteligencije. To su sustavi koji mogu učiti nove koncepte, sustavi koji mogu zaključivati i donositi uporabne zaključke o svijetu koji ih okružuje, sustavi koji mogu razumjeti prirodni jezik ili spoznati i tumačiti složene vizualne scene, sustavi koji mogu obavljati i druge vrste vještina koje zahtijevaju čovjekovu vrstu inteligencije [1].

Umjetna inteligencija je naziv koji pridajemo svakom neživom sustavu koji pokazuje sposobnost snalaženja u novim situacijama (inteligenciju). Umjetna inteligencija proučava zamisli koje omogućuju računalima obavljanje zadataka koji se ljudima čine inteligentnim.

Mnoge ljudske aktivnosti kao što su, na primjer, rješavanje slagaljki i zagonetki, igranje igara, rješavanje matematičkih problema ili pak vožnja automobila podrazumijevaju inteligenciju. Ako računala mogu obaviti takve zadatke tada se pretpostavlja da ta računala (skupa s pripadajućim programima) imaju određen stupanj umjetne inteligencije.

Uobičajeno je da se to ime pridaje računalnim sustavima. Izraz se neosnovano primjenjuje i na robote, budući da njihov sustav ne mora nužno biti inteligentan.

Jedno od područja umjetne inteligencije čine:

- ✓ **Računalni vid, raspoznavanje uzoraka i analiza scene**

1.2 Računalni vid

Računalni vid je područje umjetne inteligencije koje se bavi izlučivanjem informacije iz slike. Jedan od zadataka računalnog vida je i prepoznavanje dvodimenzionalnih i trodimenzionalnih predmeta – na primjer, znakova, predmeta, ljudskog lica. Bez razvijenog računalnog vida robot se teško može snalaziti u prostorima koji su oblikovani za potrebe ljudi koji za snalaženje u prostoru koriste prvenstveno vid.

Cilj računalnog vida je izgradnja umjetnih sustava koji mogu gledati i vidjeti. Sve je počelo davne 1967. kad je Marvin Minsky zadao jednom svom studentu ljetni projekt koji je trebao dati univerzalno rješenje strojnog gledanja. Međutim, svi problemi nisu riješeni ni do danas, a sigurno ni neće tako skoro. Ipak, značajan napredak i vrlo zanimljivi rezultati ne mogu se poreći.

2 Križna korelacija u raspoznavanju objekata

2.1 Definicija križne korelacije

Korelacija (lat. con = sa, relatio = odnos) predstavlja suodnos ili međusobnu povezanost između različitih pojava predstavljenih vrijednostima dvaju varijabli. Pri tome povezanost znači da je vrijednost jedne varijable moguće sa određenom vjerojatnošću predvidjeti na osnovu saznanja o vrijednosti druge varijable. Klasični primjeri povezanosti su npr. saznanje o utjecaju količine padalina na urod žitarica, o povezanosti slane hrane i visokog krvnog tlaka i slično. Promjena vrijednosti jedne varijable utječe na promjenu vrijednosti druge varijable [2].

Varijabla koja svojom vrijednošću utječe na drugu varijablu naziva se neovisna varijabla. Varijabla na koju se utječe naziva se ovisna varijabla. Npr. unošenje više soli u organizam utječe na porast krvnog tlaka, dok porast krvnog tlaka ne utječe na povećanje unošenja soli u organizam. U ovom primjeru unošenje soli u organizam je neovisna varijabla, a povećanje krvnog tlaka je ovisna varijabla. Mogući su slučajevi da dvije varijable istovremeno utječu jedna na drugu, pa su u tom slučaju obje varijable istovremeno i ovisne i neovisne [2].

Odnos između varijabli

Međusoban odnos između dvije varijable, grafički možemo prikazati pomoću dvodimenzionalnog grafa, tzv. *Scatter dijagram* (dijagrama raspršenja). Vrijednosti jedne varijable prikazane su na x osi, a druge na y osi dijagrama. Točke događaja kreću se oko određenog pravca koji se naziva linija regresije. Što su točke bliže pravcu, korelacija je veća. Što su točke raspršenije korelacija je manja.

U praksi je vizualno vrlo teško, osim u slučaju „savršene“ korelacije odrediti stupanj povezanosti između varijabli. Ovisno o međusobnom odnosu dvaju varijabli među kojima postoji korelacija, ona može biti linearna ili nelinearna. Kod **linearne korelacije**, točke su grupirane oko pravca. Kod **nelinearne korelacije**, točke su grupirane oko neke druge krivulje.

Dvije varijable koje promatramo sa ciljem utvrđivanja njihove korelacijske povezanosti mogu biti u 4 različita odnosa:

1. kada mala vrijednost jedne varijable odgovara maloj vrijednosti druge varijable, kao i kada velika vrijednost jedne varijable odgovara velikoj vrijednosti druge varijable, radi se o pozitivnoj korelaciji.
2. kada mala vrijednost jedne varijable odgovara velikoj vrijednosti druge varijable i obratno, radi se o negativnoj korelaciji.
3. kada vrijednost jedne varijable u nekim intervalima odgovara maloj vrijednosti druge varijable, a u drugim intervalima velikoj vrijednosti, radi se o *nemonotonoj korelaciji*. Ako se korelacija više nego jednom mijenja od pozitivne prema negativnoj, takva korelacija naziva se *ciklička korelacija*.
4. kada se na osnovu vrijednosti jedne varijable ne može zaključiti ništa o vrijednosti druge varijable, tada *korelacija ne postoji*. Točke u takvom grafu su raspršene.

Koeficijenti korelacije

Koeficijenti korelacije izražavaju mjeru povezanosti između dvije varijable u jedinicama neovisnima o konkretnim jedinicama mjere u kojima su iskazane vrijednosti varijabli. Postoji više koeficijenata korelacije koji se koriste u različitim slučajevima. U praksi se prilikom rada s linearnim modelima najčešće koristi Pearsonov koeficijent korelacije. Pearsonov koeficijent korelacije koristi se u slučajevima kada između varijabli promatranog modela postoji linearna povezanost i neprekidna normalna distribucija [2].

Vrijednost Pearsonovog koeficijenta korelacije kreće se od $+1$ (savršena pozitivna korelacija) do -1 (savršena negativna korelacija). Označava se malim latiničkim slovom r . Veličina r ukazuje na to koliko su točke blizu pravcu: ako je $r = +1$ ili $r = -1$, tada postoji potpuna povezanost sa svim točkama koje leže na pravcu (to je u praksi gotovo nemoguće); ako je $r = 0$, tada nema linearne povezanosti (premda može postojati ne-linearna povezanost); što je r bliži ekstremnim vrijednostima $(+1, -1)$ to je i stupanj linearne povezanosti veći.

Predznak koeficijenta nas upućuje na smjer korelacije – da li je pozitivna ili negativna. Pearsonov koeficijent korelacije bazira se na usporedbi stvarnog utjecaja promatranih varijabli jedne na drugu u odnosu na maksimalni mogući utjecaj dviju varijabli.

Za izračun koeficijenta korelacije potrebne su tri različite sume kvadratnog odstupanja (SS): suma kvadratnog odstupanja varijable y , suma kvadratnog odstupanja varijable x i suma umnožaka varijabli x i y .

Suma kvadratnog odstupanja varijable y jednaka je sumi kvadratnog odstupanja vrijednosti varijable y od njezine prosječne vrijednosti:

$$SS_{XX} = \sum_{i=1}^n (x_i - \bar{X})^2 \quad (1)$$

Prosječna vrijednost varijable X jednaka je:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i$$

Suma kvadratnog odstupanja varijable y jednaka je sumi kvadratnog odstupanja vrijednosti varijable y od njezine prosječne vrijednosti:

$$SS_{YY} = \sum_{i=1}^n (y_i - \bar{Y})^2 \quad (2)$$

Prosječna vrijednost varijable Y jednaka je:

$$\bar{Y} = \frac{1}{n} \sum_{i=1}^n y_i$$

Suma umnožaka kvadratnog odstupanja varijabli x i y jednaka je sumi umnožaka odstupanja vrijednosti varijabli x i y od njihovih prosjeka:

$$SS_{XY} = \sum (x_i - \bar{X})(y_i - \bar{Y}) \quad (3)$$

Koeficijent korelacije jednak je omjeru:

$$r = \frac{SS_{XY}}{\sqrt{SS_{XX} \cdot SS_{YY}}} \quad (4)$$

Uvrštavanjem jednadžbi 1, 2 i 3 u jednadžbu 4 dobivamo slijedeći izraz:

$$r = \frac{\sum (x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum (x_i - \bar{X})^2 \sum (y_i - \bar{Y})^2}}$$

Nadalje, uvrstimo li standardnu devijaciju varijable x , $\sigma_x = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{X})^2}$ i

standardnu devijaciju varijable y , $\sigma_y = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{Y})^2}$ u gornju formulu slijedi

$$r = \frac{1}{n-1} \sum_i \frac{(x_i - \bar{X})(y_i - \bar{Y})}{\sigma_x \sigma_y}$$

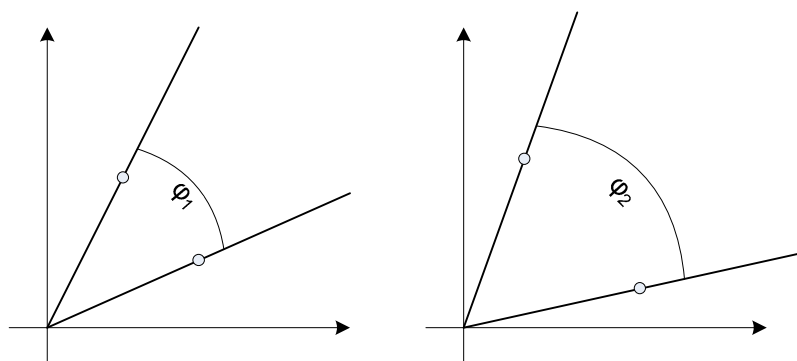
Za programe koji se bave obradom slika u kojima svjetlina slike i predložka može varirati uslijed promjene osvjetljenja i uvjeta u kojima se nalazi, potrebno je prvo normalizirati sliku. To se događa u svakom koraku oduzimanjem srednje vrijednosti ($\bar{f}$ i $\bar{t}$) i dijeljenjem trenutnog rezultata standardnom devijacijom (σ_f i σ_t). Matematički opis križne korelacije predložka $t(x, y)$ sa realnom slikom $f(x, y)$ bio bi:

$$r = \frac{1}{n-1} \sum_{x,y} \frac{(f(x, y) - \bar{f})(t(x, y) - \bar{t})}{\sigma_f \sigma_t} \quad (5)$$

gdje je n broj slikovnih elemenata (*piksela*) u $t(x, y)$ i $f(x, y)$, a x i y su koordinate slikovnog elementa u slici.

Geometrijsko objašnjenje korelacije

Seriju od n podataka može se shvatiti kao vektor u n -dimenzionalnom prostoru. Pearsonov koeficijent korelacije tada se može interpretirati kao kosinus kuta između takva dva jedinična vektora.



Slika 2.1: Geometrijsko objašnjenje korelacije

Vrijedi:

- 1) $-1 \leq \cos \varphi \leq 1$
- 2) Ako je $\cos \varphi > 0$ onda je kut među pravcima šiljast, a ako je $\cos \varphi < 0$ onda je taj kut tup
- 3) Što je $\cos \varphi$ bliži 1 ili -1 vektori su sve bliže tome da budu proporcionalni (linearno zavisni), a što je $\cos \varphi$ bliži 0, vektori su to manje kolinearni.
- 4) Ako je $\cos \varphi = 1$ ili $\cos \varphi = -1$ onda su vektori linearno zavisni; tada je kut od 0 stupnjeva i $\mathbf{v} = c \cdot \mathbf{u}$, za $c > 0$, ili je kut od 180 stupnjeva i $\mathbf{v} = c \cdot \mathbf{u}$, za $c < 0$.
- 5) Ako je $\cos \varphi = 0$ onda su vektori okomiti pa su najudaljeniji od kolinearnosti.

Dakle, $\cos \varphi$ ima svojstva analogna onima koje ima r . To znači da je $\cos \varphi$ koeficijent kolinarnosti dvaju vektora (slično kako je r koeficijent linearne korelacije dviju serija podataka).

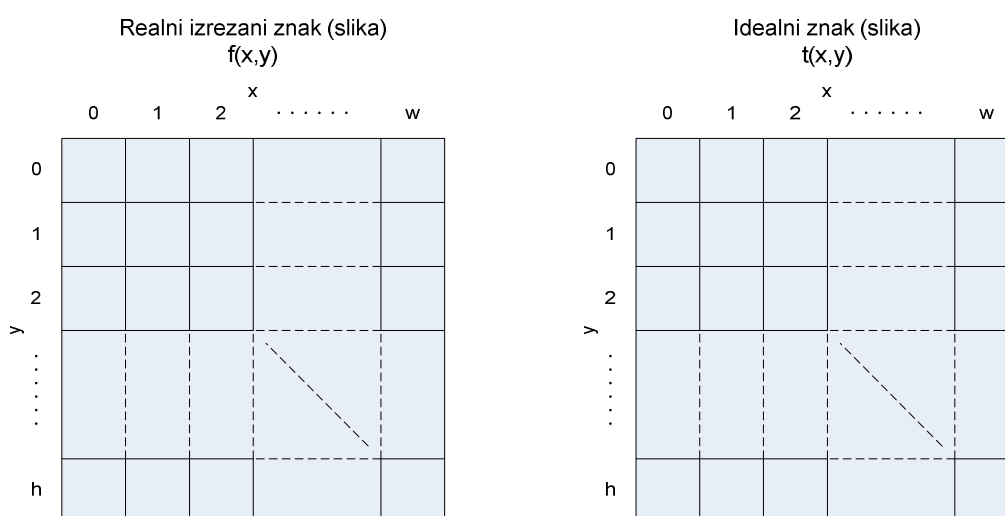
2.2 Upotreba križne korelacije u raspoznavanju objekata

U ovom radu metoda križne korelacije upotrebljava se za uspoređivanje sličnosti dvaju slika. Jedna slika (realna) $f(x,y)$ dobiva se iz ulazne, a druga $t(x,y)$ iz skupa idealnih slika znakova. Kao izlaz dobivamo koeficijent koji opisuje koliko su te dvije slike slične.

Slika 2 prikazuje način rada koreliranja dviju izrezanih realnih slika. Najčešće ne koreliramo cijele slike, nego njihove isječke. Položaji isječaka se kao hipoteze položaja traženih objekata dobivaju metodama pronalaženja objekata koje se ovdje ne razmatraju.

Dva isječka slika uspoređuju se tako da se usporede ekvivalentni slikovni elementi iz oba isječka te se dobiju parovi slikovnih elemenata koji predstavljaju istu informaciju. Ti parovi zatim se unose u formulu (5) koja će kao rezultat dati broj $r \in [-1, 1]$ koji opisuje kolika je sličnost (korelacija) između ta dva slikovna isječka.

Ako kao rezultat prethodnog izračuna dobijemo da je koeficijent korelacije $r = 1$, zaključujemo da su navedeni slikovni isječci jednaki po svakom slikovnom elementu (idealna situacija). Naprotiv, ako kao rezultat dobijemo da je $r = 0$, navedeni slikovni isječci potpuno su različiti, to jest, svi ekvivalentni slikovni elementi su različiti. I konačno, ako je rezultat korelacije $r = -1$ slikovni isječci su antikorelirani tj. jedan slikovni isječak je negativ drugoga.



Slika 2.2: Ispitna i idealna slika

Iz ovog opisa moguće je uvidjeti da će realni znak biti to sličniji idealnom što je koeficijent korelacije r bliži broju 1.

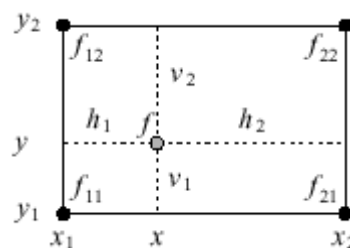
3 Normalizacija veličine slike bilinearnom interpolacijom

Veličina znaka u slici koja se trenutno obrađuje je proizvoljna i ovisi o položaju znaka u ulaznoj slici. To ovisi o udaljenosti s koje je znak snimljen. Da bismo realni znak izrezan iz ulazne slike mogli korelirati s idealnim znakovima iz baze potrebno je slikovni isječak $f(x, y)$ tj. znak skalirati na normaliziranu veličinu. Znak ćemo skalirati na normaliziranu veličinu metodom *bilinearne interpolacije*.

3.1 Definicija bilinearne interpolacije

Interpolacija slike je postupak kojim se definira prostorno kontinuirana slika iz skupa diskretnih uzoraka. Interpolacija slike pripada u temeljne postupke digitalne obrade slike i rabi se u drugim složenijim postupcima za obradu slike kao što su translacija, skaliranje, rotacija i slično, u kojima je potrebno odrediti nove vrijednosti elemenata slike na položajima na kojima oni ne postoje u izvornoj slici. Interpolirane vrijednosti elemenata slike na tim položajima se proračunavaju kao težinske funkcije srednjih vrijednosti susjednih elemenata slike. Težinska funkcija se naziva jezgra interpolacije.

Interpolacija se izvodi na sirovim podacima. Postoji nekoliko metoda za interpolaciju: od najjednostavnije, metode najbližeg susjeda, linearne interpolacije pa do puno kompleksnijih kao što su interpolacije nazvane prema tvorcima: Cok, Freeman, Laroche, Prescott, Hamilton-Adams i drugi [6].



Slika 3.1: Bilinearna interpolacija

U 1D prostoru interpolacijski postupak treba najviše 2 točke da bi se mogla izračunati interpolirana vrijednost. U 2D prostoru se takva vrsta linearne interpolacije zove *bilinearna*, a treba najviše 4 okolne točke za proračunavanje jedne točke u interpoliranoj slici, dok se kod bikubične interpolacije (*bicubic*

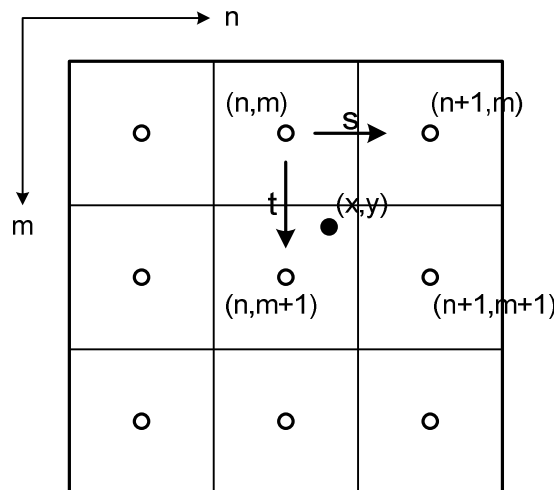
interpolation) vrijednost novog piksela izračunava na temelju vrijednosti osam susjeda.

Bilinearna interpolacija (*bilinear interpolation*) je vrsta interpolacije kod koje se vrijednost novog slikovnog elementa izračunava na temelju vrijednosti četiriju susjeda: lijevog, desnog, gornjeg i donjeg. Na taj način kreiraju se vrijednosti slikovnih elemenata u točkama izvan rešetke.

3.2 Upotreba bilinearne interpolacije u normalizaciji veličine slike

Kao što je rečeno bilinearna interpolacija je vrsta linearne interpolacije koja za dobivanje novog slikovnog elementa treba četiri susjedne točke.

Sada će biti opisano kako se vrši interpolacija slike.



Slika 3.2: Nepoznati piksel izvan rešetke

$$s = \frac{(x - x_n)}{X} \quad (6)$$

$$t = \frac{(y - y_m)}{Y} \quad (7)$$

Dana je funkcija dviju diskretnih varijabli f , koja je definirana u točkama (x_n, y_m) , $n \in [1, n_{max}]$, $m \in [1, m_{max}]$:

$$f(s, t) = (1 - s)(1 - t)f_{n,m} + s(1 - t)f_{n+1,m} + (1 - s)tf_{n,m+1} + stf_{n+1,m+1} \quad (8)$$

Pretpostavljamo da je „rešetka“ nad kojom je funkcija definirana pravokutna:

$$x_n - x_{n-1} = X, \quad y_m - y_{m-1} = Y, \quad \forall m, n > 0$$

Zanima nas koja bi bila vrijednost interpolirane funkcije f_i u nekoj nepoznatoj točki (x, y) izvan rešetke (Slika 3.2: Nepoznati piksel izvan rešetke).

Prvo nalazimo „najbliže manje“ koordinate rešetke (x_n, y_m) u odnosu na zadanu točku (x, y) :

$$0 \leq x - x_n < X$$

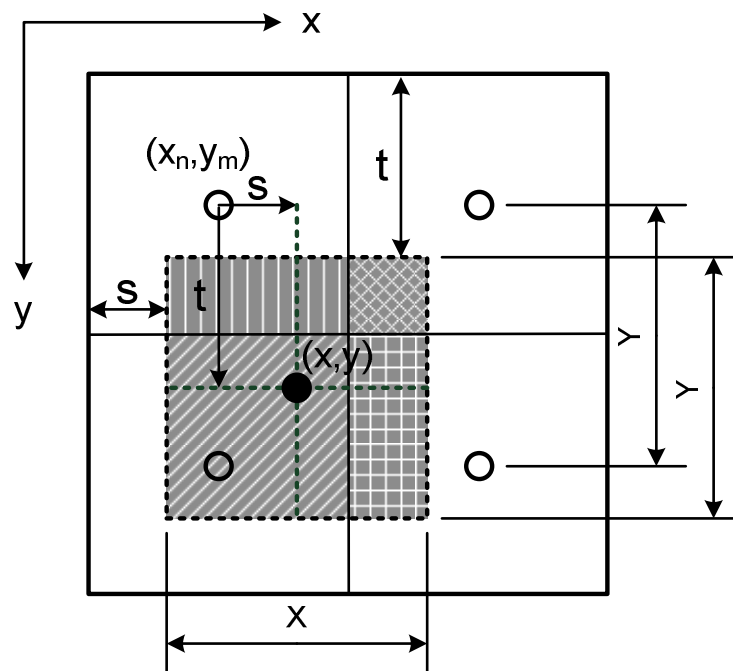
$$0 \leq y - y_m < Y$$

X i Y su širina odnosno visina slikovnog elementa, najčešća vrijednost je 1.

Uvrštavanjem formula 6 i 7 u formulu 8 slijedi:

$$\begin{aligned} f_i(x, y) = & f_i(x_n + s * X, y_m + t * Y) = \\ & (1 - s) * (1 - t) * f(x_n, y_m) + \\ & s * (1 - t) * f(x_{n+1}, y_m) + \\ & (1 - s) * t * f(x_n, y_{m+1}) + \\ & s * t * f(x_{n+1}, y_{m+1}) \end{aligned}$$

gdje su $s = \frac{(x-x_n)}{X}$, $t = \frac{(y-y_m)}{Y}$.



Slika 3.3: Geometrijsko objašnjenje gornjeg izraza

Dobiven je izraz za izračun vrijednosti slikovnih elemenata u točkama izvan rešetke!

Potrebno je napomenuti da kod dobivanja manje slike iz veće, pomoću ovakve interpolacije, može doći do problema pri dobivanju potpuno ispravne slike zbog *aliasinga*.

Rješenje ovog problema je da bi se trebalo zamutiti izvornu sliku s prikladnim parametrom σ ; σ bi bio proporcionalan odnosu dimenzija ispitanog isječka i idealne slike.

Računski, to bi bio dosta zahtjeva postupak i ugrozio bi mogućnost rada u stvarnom vremenu. Zbog toga, opisani mogući problemi bit će zanemareni u ovome radu radi usredotočivanja na zadatak rada.

4 Programska izvedba

Tema ovog poglavlja je opis izvedbe programske potpore za raspoznavanje prometnih znakova u nizu slika.

U prvom dijelu poglavlja, opisuje se programsko okruženje koje je korišteno za dohvat slika i pokretanje algoritma.

U drugom dijelu poglavlja pokazuje se način na koji je algoritam za raspoznavanje prometnih znakova implementiran u programsko okruženje te od kojih glavnih dijelova se sastoji.

U trećem dijelu poglavlja, navode se odabrane biblioteke i razvojni alati koji su olakšali izvedbu algoritma.

4.1 Programsko okruženje

Razvojna ljuska računalnog vida omogućava interaktivno ispitivanje raznih postupaka iz domene računalnog vida te olakšava eksperimentiranje. Zbog određene sličnosti s korisničkom ljuskom koja je uobičajena na operacijskim sustavima UNIX, program je nazvan ljuskom računalnog vida (eng. `cvsh` - computer vision shell).

Program se izvodi u jednom ili više izvedbenih ciklusa, koji se zadaju pozivom programa u naredbenom retku ili interaktivnim dijalogom. U svakom ciklusu dohvaća se jedna ili više slika iz izvora definiranog odabranom komponentom paketa `vs`, slike se obrađuju postupkom kojeg definira odabrani algoritam izveden iz razreda `alg_base`, dok se odredišna slika šalje u jedan ili više ponora slike definiranih komponentama paketa `vd`. Algoritmi obično imaju parametre koji se mogu interaktivno podešavati između svaka dva izvedbena ciklusa. Ukoliko je barem jedan od ponora prozor, ljuska nudi mogućnost iscrtavanja pronađenih značajki preko odredišne slike. Konfiguracija programa se može obaviti i pri pozivu programa i interaktivno, a obuhvaća odabire izvora slike, postupka obrade, ponora slike, te parametara postupka obrade.

Dodavanje novih algoritama u programsko okruženje iznimno jednostavno, jer se komponente napisane u skladu s konvencijama povezuju s ostatkom aplikacije automatskim metodama.

Sadašnja implementacija razvojne ljuske podržava samo tekstualno korisničko sučelje, dok se grafičke mogućnosti modernih računala koriste samo pri prikazu rezultata obrade.

4.1.1 Parametri komandne linije

Ljuska omogućuje dva temeljna načina rada: neinteraktivni i interaktivni. U slučaju neinteraktivnog rada, parametri naredbenog retka u potpunosti određuju tijek izvođenja, dok u suprotnom oni definiraju samo početnu konfiguraciju programa koja se tokom interaktivnog rada može promijeniti. Naredbenim retkom može se utjecati na sljedeće parametre izvođenja programa:

- izvor slijeda slika (opcija -s) - sklopovlje ili datoteka;
- postupak obrade (opcija -a) - ugrađeni algoritam ili put do dinamičke biblioteke;
- konfiguracijski niz (opcija -c) - ovisan o postupku obrade;
- interaktivni rad (opcija -i) - može se zadati početna naredba interaktivnog rada;
- ponor slijeda slika (opcija -d) - datoteka ili prozor;
- maksimalna performansa (opcija -m) - definira minimalno vrijeme obrade svake slike;

Kod interaktivnog rada, podrazumijeva se prozor kao jedini ponor slike.

Sasvim općenito, program se može pozvati upotrebom jednog od sljedećih sintaksnih oblika:

1. neinteraktivni rad:

```
cvsh -s<Source>[#<Range>] -a=<Algorithm> -c=<ConfigString> -d<Dest> [-
    d<Dest> ... ]
```

2. interaktivni rad:

```
cvsh -s<Source>[#<Range>] -a=<Algorithm> -c=<ConfigString>-  
i[=<CommandString>
```

Sintaksa pojedinih opcija je opisana u nastavku teksta, a njen kraći prikaz može se dobiti pozivanjem programa bez argumenata.

Opcija `-s<Source>[#<Range>]`

Specificira se izvor slijeda slika `<Source>`, te interval slika `<Range>` koji se želi obraditi. Izvor slika općenito može biti datoteka ili međusklop za pribavljanje slike pa prema tome sintaksa tijela opcije `<Source>` može imati različite oblike. Navodi se najčešće korišteni oblik:

1. `f = filename;`

- gdje `filename` predstavlja valjanu putanju do željene izvorne datoteke,
- format datoteke mora odgovarati ekstenziji ulazne datoteke;
- ako se želi učitati cijeli direktorij pojedinačnih slika tada se na kraju valjane putanje do željenog direktorija dodaje „\\“

Interval slika `<Range>` se uzima u obzir samo ako je izvor slike datoteka i služi za ograničavanje obrade na slike datoteke čiji redni brojevi su u intervalu sadržani.

Opcija `-a=<Algorithm>`

Znakovni niz `<Algorithm>` definira postupak obrade. Moguće je odabrati jedan od ugrađenih postupaka obrade (npr. Algoritam za raspoznavanje prometnih znakova), ili valjanu putanju izvršne datoteke s vanjskim postupkom.

Opcija `-c=<ConfigString>`

Specificira se konfiguracijski niz znakova za postupak određen opcijom `-a`. Konfiguracijski nizovi ovise o pojedinom algoritmu.

Opcija `-d<Dest>`

Specificira se odredište slijeda slika `<Dest>`, koji općenito može biti datoteka ili prikazna jedinica.

Opcija `-i [=<CommandString >]`

Specificira se interaktivni način rada i, eventualno, prva interaktivna naredba.

Primjer poziva korištenja programa

```
cvsh -sf="ulaz/slike\\" -a=korel -i="p t" -c="ulaz/oznake/oznake.txt"
```

4.1.2 Formati ulaznih datoteka

Kod opcija `-sf=...` i `-df=...`, ekstenzija specificirane datoteke mora odgovarati njenom formatu, kao što slijedi:

- ekstenzija `.avi`: datoteka je u Microsoftovom AVI formatu (format nije podržan na operacijskim sustavima koji nisu proizvedeni od strane Microsofta);
- ekstenzija `.srs`: datoteka se sastoji od slijeda slika u nekomprimiranom Sun raster formatu;
- ekstenzije `.ppm`, `.bmp`, `.sr`: datoteka se sastoji od jedne slike u portable pixmap, Microsoft bmp odnosno Sun raster formatu;
- „\\“ – ako se doda na kraj imena direktorija ovaj znak učitati će se sadržaj tog direktorija

4.1.3 Interaktivne naredbe ljske

Ako je u komandnoj liniji specificiran interaktivni način rada (opcija `-i`), ljska prilikom pokretanja izvršava početnu naredbu (ako je ona zadana) i prepušta daljnje upravljanje korisniku, preko tekstualnog korisničkog sučelja. Naredbe koje sučelje podržava se mogu svrstati u tri skupine:

- naredbe za podešavanje parametara izvođenja (algorithm, source i config);
- naredbe za prikaz i obradu slika ulaznog slijeda (process i show);
- ostale naredbe (help i quit).

4.2 Integracija algoritma za raspoznavanje prometnih znakova u programsko okruženje i njegovi glavni dijelovi

Zbog svojstvenog načina ljuske na vrlo jednostavan način moguće je dodavati nove mogućnosti i nove algoritme. To se postiže nasljeđivanjem osnovnog razreda `alg_base`.

Poziv korisničkog algoritma izveden je virtualnim pozivom metode `alg_signsCorrelation::process` koja je deklarirana u razredu `alg_signsCorrelation` i to na sljedeći način:

```
virtual void process(
    const img_vectorAbstract& src,
    const win_event_vectorAbstract& events,
    int msDayUtc, int nframe);
```

4.2.1 Opis algoritma

Algoritam se sastoji od nekoliko glavnih dijelova. U prvom dijelu se učitavaju informacije o položaju znaka u ulaznoj slici.

Kada se učitava redak iz datoteke koja sadrži hipotetizirane koordinate znakova u pojedinoj slici program dolazi do petlje u kojoj se obavlja glavni dio posla.

Metodama križna korelacija i bilinearna interpolacija (koje su opisane u poglavlju 2 odnosno poglavlju 3) transformiraju se podaci tj. slikovni elementi iz ulazne slike u željenu informaciju, a to je šifra znaka ili znakova koji se nalaze na trenutnoj slici.

To se izvodi na način da izrezani znak iz ulazne slike bilinearnom interpolacijom normaliziramo na veličinu jednaku idealnim znakovima u bazi. Zatim normalizirani realni znak uspoređujemo tj. koreliramo sa svakim idealnim znakom. Rezultat korelacije je koeficijent sličnosti realnog i idealnog znaka. Traženjem najveće korelacije nalazimo najsličniji znak i rezultat obrade je identifikator idealnog znaka.

Program izlazi iz petlje nakon što su svi znakovi na trenutnoj slici obrađeni i raspoznati.

4.3 Korištene biblioteke i razvojni alati

Razvoj ovog rada potpomognut je raznim bibliotekama i razvojnim alatima. U daljnjem tekstu bit će navedeni uz kratko objašnjenje.

1. Standardna biblioteka jezika C++
 - Ova biblioteka sastavni je dio jezika C++ i uključuje temeljne komponente za strukturiranje podataka (dinamičko polje, lista ...) i odgovarajuće temeljne algoritme (iteraciju, prebrojavanje, pretraživanje, sortiranje), te transparentno baratanje tokom znakova neovisno o izvorištu i odredištu (memorija, konzola ili datoteka).
2. Biblioteka Boost
 - Boost je biblioteka koja proširuje funkcionalnost standardne biblioteke programskog jezika C++;
 - Sastoji se od skupa biblioteka opće namjene koje uključuju pametne pokazivače, apstrahiranje OS-a, operacije linearne algebre te ostalih biblioteka koje olakšavaju rad korisniku jezika C++.
3. PixelRuler
 - Pri ručnom označavanju stvarnih znakova iz ulaznog slijeda slika korišten je jednostavan, a vrlo koristan program PixelRuler koji olakšava posao određivanja koordinata znakova u ulaznim slikama;
 - Program se može pronaći u trial verziji na stranici:
 - <http://www.mioplanet.com/products/pixelruler/index.htm>
4. Cygwin
 - Cygwin je Unix emulacijski sloj za operacijski sustav Windows;

- Cygwin ljuška je alternativa standardnoj ljušci Windowsa (cmd) koja ima ugrađene napredne mogućnosti dostupne na operacijskom sustavu Unix;
- Cijeli Cygwin paket može se besplatno skinuti i instalirati na Windows stroju sa stranice:
 - <http://www.cygwin.com/>

5. ImageMagick

- Skup programa otvorenog koda za baratanje slikama;
- Podržano je gotovo stotinu formata;
- Koristi se pretežito za obavljanje različitih zahvata promjena i pretvorbi na slikama;
- Koristan za obradu više slika odjednom, npr. ako želimo promijeniti veličinu više slika odjedanput;
- Dostupan kao zaseban program, ali i u raznim drugim izvedbama kao što je paket za cygwin koji je korišten u radu.

5 Eksperimentalni rezultati

Dva pojma *pronalaženje (detekcija, prepoznavanje)* i *raspoznavanje* mogu imati u nekim slučajevima isto značenje no bitno je naglasiti da u ovom kontekstu imaju različita značenja. Da ne bi došlo do krive interpretacije dat ću kratak opis značenja tih pojmova.

Pronalaženje (detekcija, prepoznavanje) znakova je postupak koji odgovara na pitanje: *Gdje su objekti (znakovi) i da li ih ima?*

Raspoznavanje znakova je postupak kojim se raspoznavanju znakovi jedan od drugoga. Dok se kod prepoznavanja samo traži gdje se znak nalazi, raspoznavanje pod pretpostavkom da je poznato mjesto znaka otkriva *koji* je to znak, otkriva tip znaka (npr. „obavezan smjer“).

Ovaj rad bavi se raspoznavanjem znakova te će u ovom poglavlju to biti pokazano na primjerima.

Potrebno je naglasiti da **točnost raspoznavanja** prometnih znakova uvelike ovisi o preciznosti položaja znakova bilo da su dobiveni ručno ili programski. Osim toga algoritam za raspoznavanje to će bolje raditi što su ulazne slike kvalitetnije i oštrije.

Program prije korelacije realnih slikovnih isječaka primjenjuje masku. Maska je 1-bitna slika gdje bijeli piksel označava da se piksel pretpostavljenog znaka u slikovnom elementu uzima u obzir prilikom koreliranja.

Kako postoje maske kvadratnih, kružnih i trokutastih znakova, znakovi romboidnog oblika lošije se raspoznaju jer u korelaciju ulaze i pikseli koji nisu dio znaka već dio okoliša u koje se znak nalazi.

5.1 Primjeri rada algoritma

Rad algoritma bit će prikazan na primjerima slika dobivenih iz tri izvora:

1. videa,
2. mobitela – slike manje kvalitete
3. te na kraju na idealnim znakovima.

5.1.1 Slike izlučene iz videa snimljenog u automobilu.

Primjer 1:

```

C:\WINDOWS\system32\cmd.exe
UIloop: got command process next 1
      Algoritam KORELACIJA - zapoceo!
-----
C08B671,73:P,60#B57E673,126:K,44
Oblik: P
      C08 <0.340438>
      C09 <0.336404>
      C18 <0.297986>
Oblik: K
      B57 <0.627999>
      B40 <0.419081>
      B39 <0.400823>
Broj pronađenih znakova je: 2
-----
Overall profile:
  fetch=63.4ms
  process=246.7ms
  store=381.8ms
Processed 1 frames in 0.7 s
Throughput: 1.4 fps.
slike\11\7515
  
```

a)



b)

Slika 5.1: Izlaz algoritma, a) ispis bitnih informacija, b) pregled raspoznatih znakova

Znak prava prvenstva (C08 – na slici 5.1.b gornji znak) prepoznat je s najvećom korelacijom u odnosu na druge znakove, znači bez pogreške što se vidi na Slika 5.1.a, iako nije pravokutan već romboidan što otežava njegovo raspoznavanje. To je razlog malom korelacijskom koeficijentu ($r = 0.34$).

Znak dopuštenih smjerova (B57 – na slici 5.1.b donji znak) prepoznat je s velikim korelacijskom koeficijentu ($r = 0.63$).

Primjer 2:

```

C:\WINDOWS\system32\cmd.exe
Uiloop: got command process next 1
      Algoritam KORELACIJA - zapoceo!
-----
C02E622,160:P,37#B39E534,196:K,16
Oblik: P
      C10 <0.396949>
      C126 <0.372225>
      C02 <0.354581>
Oblik: K
      B61 <0.328983>
      B46 <0.158599>
      B44 <0.124910>
Broj pronadjenih znakova je: 2
-----
Overall profile:
  fetch=63.2ms
  process=277.9ms
  store=388.9ms
Processed 1 frames in 0.7 s
Throughput: 1.4 fps.
slike\15\7515
  
```

a)



b)

Slika 5.2: Izlaz algoritma, a) ispis bitnih informacija, b) pregled raspoznatih znakova

Na ovom primjeru možemo uočiti kako algoritam nije uspio dobro raspoznati znakove „obilježen pješački prijelaz“ (C02) i „zabrana zaustavljanja i parkiranja“ (B39) zbog zamućenosti slike i vrlo malih znakova u slici. Rješenje problema je približavanje automobila (kamere) znakovima te upotreba kvalitetnije video opreme.

Primjer 3:

```

C:\WINDOWS\system32\cmd.exe
Uiloop: got command process next 1
      Algoritam KORELACIJA - započeo!
-----
B31@647,147:K,46#A17@647,101:T,53
Oblik: K
      B31 <0.310967>
      B16 <0.310317>
      B26 <0.301686>
Oblik: T
      A17 <0.618419>
      A24 <0.584525>
      A18 <0.530697>
Broj pronađenih znakova je: 2
-----
Overall profile:
  fetch=63.5ms
  process=176.4ms
  store=386.9ms
Processed 1 frames in 0.6 s
Throughput: 1.6 fps.
slike\l8\7515
  
```

a)



b)

Slika 5.3: Izlaz algoritma, a) ispis bitnih informacija, b) pregled raspoznatih znakova

Primijetimo kako su oba znaka pronađena na slici (znak „ograničenje brzine“ (B31) i znak „neravan kolnik“ (A17)) jako dobro raspoznati, jedan od njih čak i sa velikom korelacijom (A17: $r = 0.62$), iako se nalaze na relativno tamnoj i sličnoj podlozi.

Kod raspoznavanja znaka „ograničenje brzine“ (B31) dolazi do male pogreške zbog razlike u broju koji označava maksimalnu dozvoljenu brzinu. Na primjeru ove slike to je brzina od 60km/h dok je na slici idealnog znaka brzina od 40km/h.

5.1.2 Slike snimljene mobilnim telefonom

Slike iz ove vrste izvora uzete su radi provjere rada algoritma na slikama manje kvalitete i oštine. Pogledajmo rezultate.

Primjer 4:

```

C:\WINDOWS\system32\cmd.exe
Ulloop: got command process next 1

      Algoritam KORELACIJA - započeo!
-----
A42@156,304:T,63#B30@142,454:K,48#C08@980,355:P,30
Oblik: T
      A42 <0.611199>
      A24 <0.596275>
      A39 <0.591445>
Oblik: K
      B30 <0.716937>
      B10 <0.538457>
      B29 <0.513534>
Oblik: P
      C34 <0.490053>
      C43 <0.463904>
      C58 <0.458886>
Broj pronadjenih znakova je: 3
-----
Overall profile:
  
```

a)



b)

Slika 5.4: Izlaz algoritma, a) ispis bitnih informacija, b) pregled raspoznatih znakova

Znak “tramvajska pruga” (A42) koji označuje blizinu mjesta na kojem cesta prelazi preko tramvajske pruge u razini i znak “zabrana polukružnog okretanja” (B30) koji označuje mjesto na kojem je zabranjeno polukružno okretanje vidimo na lijevoj strani slike 5.4.b. Iako je kvaliteta vrlo slaba i veličina na ulaznoj slici mala znakovi su raspoznati sa velikim korelacijskim koeficijentom što je izuzetno dobro.

Primjer 5:

```

C:\WINDOWS\system32\cmd.exe
Uiloop: got command process next 1
      Algoritam KORELACIJA - započeo!
-----
B01@586,333:K,28#B61@366,375:K,24
Oblik: K
      B01 <0.366399>
      B50 <0.274770>
      B54 <0.268766>
Oblik: K
      B61 <0.410065>
      B46 <0.300965>
      B49 <0.287361>
Broj pronadjenih znakova je: 2
-----
Overall profile:
  fetch=28.8ms
  process=158.1ms
  store=6.2ms
Last frame processing profile:
  scaling=0.0ms
Processed 1 frames in 0.2 s
Throughput: 5.3 fps.
mob\I3/111$
  
```

a)



b)

Slika 5.5: Izlaz algoritma, a) ispis bitnih informacija, b) pregled raspoznatih znakova

Znakovi, „raskrižje s cestom s prednošću prolaska“ (B01) i „obavezno obilaženje“ (B61), uhvaćeni u ovoj sceni vrlo su mali te se okom jedva mogu prepoznati. Programski su točno prepoznati unatoč lošim uvjetima.

Primjer 6:

```

C:\WINDOWS\system32\cmd.exe
scaling=0.0ms
Processed 1 frames in 0.2 s
Throughput: 6.4 fps.
mob\14\111$
UIloop: got command process next 1

    Algoritam KORELACIJA - započeo!

-----
C35@507.325:P,140
Oblik: P
          C35 <0.806904>
          C36 <0.480163>
          C32 <0.444940>
Broj pronađenih znakova je: 1
-----
Overall profile:
  fetch=28.8ms
  process=158.1ms
  store=6.7ms
Last frame processing profile:
  scaling=1.1ms
Processed 1 frames in 0.2 s
Throughput: 5.3 fps.
mob\15\111$

```

a)



b)

Slika 5.6: Izlaz algoritma, a) ispis bitnih informacija, b) pregled raspoznatih znakova

U ovome primjeru, u kojem se obrađuje znak „parkiralište“ (C35), bitnu ulogu igra dobro pronalaženje znaka zbog čega je korelacijski koeficijent dostigao $r = 0.81$. Kada bi došlo do gubitka preciznosti položaja znak ne bi bio točno raspoznat.

Testiranjem je utvrđeno za ovaj primjer da algoritam za raspознаvanje ne bi točno raspoznao znak „parkiralište“ (C35) ili bi točno raspoznao ali sa vrlo malim korelacijskim koeficijentu ukoliko bi označili veći prostor ili pomaknuli oznaku za mjesto na kojem se nalazi znak.

Primjer 7:

```

C:\WINDOWS\system32\cmd.exe
Uiloop: got command process next 1

    Algoritam KORELACIJA - započeo!
-----
B01@252,198:K,177#C08@696,475:P,45
Oblik: K
      B01 <0.470911>
      B02 <0.153412>
      B61 <0.128614>
Oblik: P
      C10 <0.206724>
      C09 <0.262526>
      C08 <0.254069>
Broj pronađenih znakova je: 2
-----
Overall profile:
  fetch=30.4ms
  process=106.6ms
  store=7.3ms
Last frame processing profile:
  scaling=1.1ms
Processed 1 frames in 0.2 s
Throughput: 4.6 fps.
mob\I9\111$
  
```

a)



b)

Slika 5.7: Izlaz algoritma, a) ispis bitnih informacija, b) pregled raspoznatih znakova

Unatoč rotaciji znaka “križanje s cestom s prednošću prolaska” (B01), koji označuje blizinu križanja na kojem vozač mora dati prednost svim vozilima što se kreću cestom na koju on nailazi, raspoznat je s koeficijentom $r = 0.47$. Bitno je naglasiti da znak s drugim najvećim koeficijentom korelacije ima tek $r = 0.15$ što daje dodatnu garanciju za točnost raspoznavanja.

Drugi znak koji se nalazi na ovoj slici je znak „cesta s prednošću prolaska“ (C08). Slabo je uočljiv zbog velike rotacije. To je glavni razlog zbog kojeg znak nije uspješno raspoznat. Prepoznavanje u ovakvim krajnostima nije ni potrebna, nije uvjet za dobar rad algoritma.

Primjer 8:

```

C:\WINDOWS\system32\cmd.exe
Uiloop: got command process next 1

    Algoritam KORELACIJA - započeo!

-----
B03@70,214:K,87#B39@934,215:K,89
Oblik: K
      B03 <0.979651>
      B16 <0.848918>
      B24 <0.804798>
Oblik: K
      B40 <0.501625>
      B39 <0.497505>
      B02 <0.334906>
Broj pronađenih znakova je: 2
-----
Overall profile:
  fetch=30.3ms
  process=150.8ms
  store=6.9ms
Last frame processing profile:
  scaling=0.0ms
Processed 1 frames in 0.2 s
Throughput: 5.3 fps.
mob\110\1115
  
```

a)



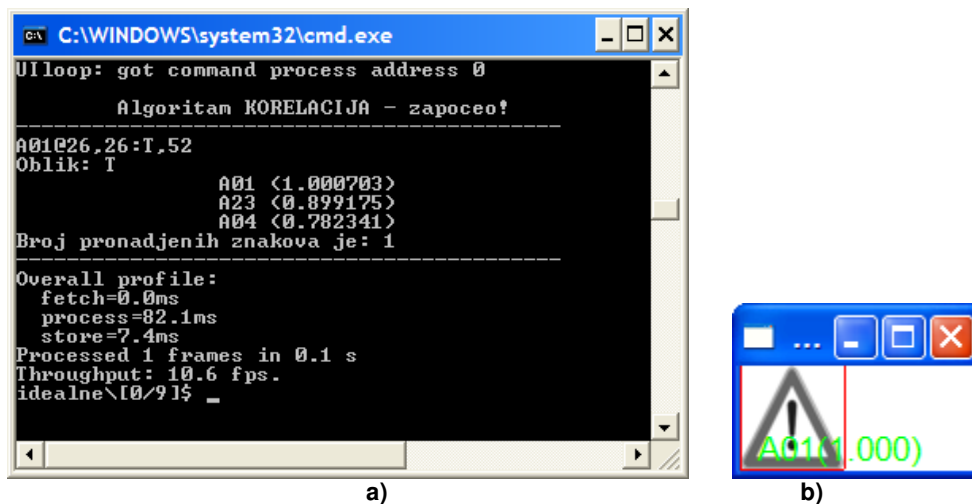
b)

Slika 5.8: Izlaz algoritma, a) ispis bitnih informacija, b) pregled raspoznatih znakova

Znak “zabrana prometa u oba smjera” (B03), koji označuje cestu odnosno dio ceste na kojem je zabranjen promet svim vozilima u oba smjera, vrlo je zanimljiv radi činjenice što je raspoznat uz gotovo idealan korelacijski koeficijent ($r = 0.98$).

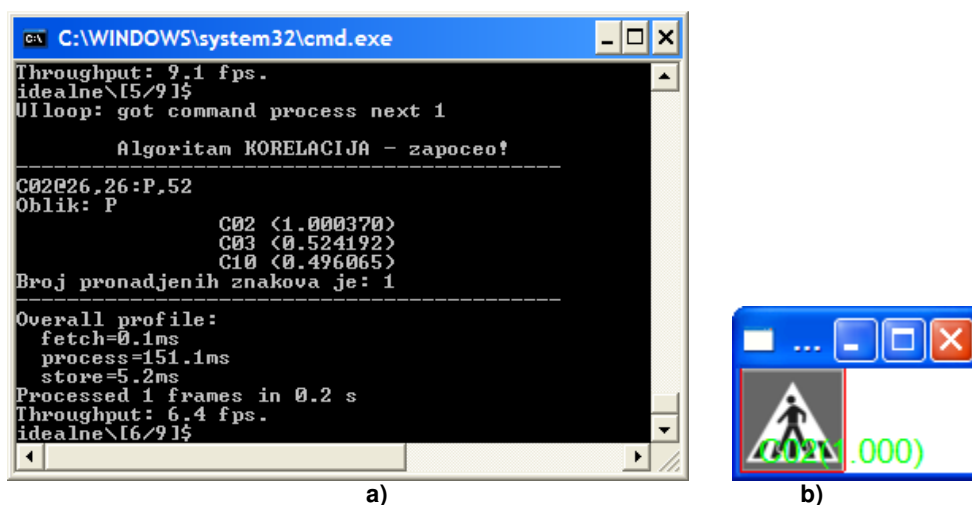
5.1.3 Slike idealnih znakova

Primjer 9:



Slika 5.9: Izlaz algoritma, a) ispis bitnih informacija, b) pregled raspoznatih znakova

Primjer 10:



Slika 5.10: Izlaz algoritma, a) ispis bitnih informacija, b) pregled raspoznatih znakova

Na temelju primjera 9 i 10, u kojem su kao ulaz u program dani idealni znakovi „opasnost na cesti“ (A01) i „obilježen pješački prijelaz“ (C02), zaključujemo da je algoritam za raspoznavanje točno izveden jer za idealan ulaz (idealni ulazni znak) daje idealan izlaz ($r = 1$).

6 Zaključak

Isprobali smo metodu križne korelacije na temelju koje smo izgradili algoritam za raspoznavanje znakova u prometu. Raspoznavali smo znakove jer kroz njih dobivamo veliku količinu informacije o trenutnoj situaciji u kojoj se vozilo nalazi.

Dobili smo zadovoljavajuće rezultate testiranjem napravljene programske potpore. Takvi rezultati obećavaju ulaz kvalitetnih informacija u sustav relevantnih za odlučivanje o kretanju automobila upravljanog računalom.

Kako je ovo prva verzija algoritma za raspoznavanje prometnih znakova uz pomoć metode križne korelacije, postoji još dosta mjesta za poboljšavanje rada algoritma (kao što je primjena križne korelacije na slike u boji, ubrzavanje algoritma,...), a time i dobivanje još boljih rezultata što se preslikava na dobivanje većeg koeficijenta korelacije, u idealnom slučaju $r = 1$.

Dominik Rojković

Literatura

1. Patterson D. W.: „Introduction to Artificial Intelligence and Expert Systems“, Prentice-Hall Inc., New Jersey, USA, 1990
2. Wikipedia: Korelacija, s interneta, <http://hr.wikipedia.org/wiki/Korelacija>, siječanj 2009
3. Budin, Leo; Dalbelo Bašić, Bojana; Pavešić, Nikola; Ribarić, Slobodan: „Intelligentni sustavi“, Fakultet elektrotehnike i računarstva, Zagreb, 2001
4. Wikipedia: Računalni vid, s interneta, http://hr.wikipedia.org/wiki/Ra%C4%8Dunalni_vid, 2008
5. Wikipedia: Umjetna inteligencija, s interneta, http://hr.wikipedia.org/wiki/Umjetna_inteligencija, 2008
6. Ciceli Tomislav: „Nastajanje boje u CCD senzoru“, Ekscentar, Studentski Zbor Geodetskog fakulteta Sveučilišta u Zagrebu, Zagreb, 2007
7. Štambuk Ljubica: „Matematika sa statistikom“, Veleučilište u Rijeci, Rijeka, 2005
8. Zisserman, A.: Two dimensional signal analysis, s interneta, <http://www.robots.ox.ac.uk/~az/lectures/sa/lect12.pdf>, 2008
9. Wikipedia: Bilinear interpolation, s interneta, http://en.wikipedia.org/wiki/Bilinear_interpolation, 2008
10. Wikipedia: Correlation, s interneta, <http://en.wikipedia.org/wiki/Correlation>, 2008
11. Wikipedia: Cross-correlation, s interneta, http://en.wikipedia.org/wiki/Cross-correlation#Normalized_cross-correlation, 2008
12. Bourke, Paul: Cross Correlation, s interneta, <http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/correlate/>, 2008

13. Šegvić, Siniša: „Višeagentsko praćenje objekata aktivnim računarskim vidom“, Doktorska dizertacija, Fakultet elektrotehnike i računarstva, Zagreb, 2004
14. H. Lohninger: „Teach/Me Data Analysis“, Springer-Verlag, Berlin-New York-Tokyo, 1999
15. Maletić, Mladen; Fajt, Siniša: „Korelacijska funkcija i spektar“, Fakultet elektrotehnike i računarstva, Zagreb, 2005
16. Karadjole, Ivo: „Korelacija i regresija“, predavanja iz Statistike, Veterinarski fakultet, Zagreb, 2005
17. SwitchOnTheCode: C# Tutorial - Convert a Color Image to Grayscale, s interneta, <http://www.switchonthecode.com/tutorials/csharp-tutorial-convert-a-color-image-to-grayscale>, 2008
18. Marko Čupić: Skripta iz računalne grafike, s interneta, http://www.zemris.fer.hr/predmeti/rg/seminari/00_Cupic/rg_komplet.pdf, listopad 2008

Sažetak

Ovaj rad razmatra raspoznavanje prometnih znakova u stvarnim slikama pod pretpostavkom da su položaji znakova već pronađeni nekom drugom metodom. Raspoznavanje se obavlja križnom korelacijom isječka stvarne slike sa slikama idealnih znakova. Isječci su svedeni na dimenzije idealne slike bilinearnom interpolacijom.

Detaljno su opisane upotrijebljene metode te je opisana njihova uloga u cjelokupnom postupku. Dan je sažeti opis programske izvedbe postupka za raspoznavanje prometnih znakova te njegovih sastavnih komponenti. Navedene su korištene vanjske biblioteke i razvojni alati. Opisan je način rukovanja razvijenim programom. Na kraju su prikazani i komentirani eksperimenti na stvarnim slikama.

Ključne riječi:

Umjetna inteligencija, računalni vid, raspoznavanje objekata, križna korelacija, linearna i bilinearna interpolacija, usporedba sličnosti slikovnih isječaka, normalizacija veličine slike

Summary

This work considers recognition of the traffic signs in real images, under assumption that the positions of the signs are detected by some other method. Recognition is performed by cross correlation of real image clips with images of ideal signs. Image clips are resized to the size of ideal images by bilinear interpolation.

The employed methods are described in detail, as well as their role in the whole procedure. The developed software components for recognizing traffic signs are described. The employed libraries and development tools are provided as well as instructions for running the developed program. The software implementation of the procedure is described. Finally, the experiments on real images are presented and discussed.

Keywords:

Artificial intelligence, computer vision, object recognition, cross correlation, linear and bilinear interpolation, image clips similarity, normalizing image size

Privitak

Uz rad je priložen DVD-ROM s izvornim kodom ljuške računalnog vida s implementiranim algoritmom za raspoznavanje znakova te izvršna verzija iste. Uz to, priloženi su i paketi eksperimentalnih slika korištenih za ispitivanje rada programske potpore.