SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 808

PRONALAŽENJE PROMETNOG TRAKA ANALIZOM HISTOGRAMA GRADIJENTA

Šime Bašić

Zagreb, lipanj 2009

Sadržaj:

1. Uvod

Sigurnost ljudi i jednostavnije izvršavanje uobičajenih ljudskih aktivnosti su neprekidni pokretači tehnološkog razvoja i na području računalnog vida. U današnje vrijeme kada se na cestama događaju brojne prometne nesreće čiji ukupan broj stalno raste, sigurnost sudionika prometa je postalo važno pitanje na koje je potrebno dati efektivne pravovremene odgovore. Podaci iz Ujedinjenih Naroda govore da je u nesrećama koje su na bilo koji način povezane sa prometom poginulo 1.26 milijuna ljudi, odnosno oko 3000 ljudi po danu. Česti uzrok nesreća su umor i nepažnja vozača, ali i slučaj kada vozači zaspu za volanom nakon čega vozilo napusti svoj prvotni trak i može uzrokovati nesreću. Sustavi temeljeni na računalnom vidu bi trebali kroz detekciju i praćenje linija traka te putem alarmnog sustava uslijed skretanja sa inicijalnog traka povećati sigurnost u prometu i smanjiti broj nezgoda. S druge strane, praćenje prometnog traka bi također moglo pomoći u autonomnoj vožnji bez sudjelovanja čovjeka u upravljanju vozilom.

U ovom radu nastojati ćemo pronaći prometni trak na uzorku slika snimljenih iz vozila u pokretu. Pretpostavka je da cesta zauzima većinu donjeg dijela vidnog polja, a tamo su linije traka dominantni rubovi. Prometni trak ćemo pokušati pronaći analizom histograma gradijenta putem koje bismo trebali pronaći dominantne smjerove gradijenta na slici. Nakon što smo pronašli smjerove gradijenata koristimo Houghovu jednodimenzionalnu transformaciju kako bismo za svaki smjer izračunali odgovarajuću vrijednost parametra ρ. U tom postupku svaki piksel čiji je smjer jednak jednom od pronađenih dominantnih smjerova glasuje u akumulatorskom polju za one vrijednosti parametra ρ koje pripadaju pravcima kroz koje dotični piksel prolazi. Zatim ćemo pomoću dominantnih smjerova i njima odgovarajućih parametara ρ definirati jednadžbe pravaca koje opisuju detektirane pravocrtne segmente.

Rad je strukturiran na način da ćemo se na početku upoznati sa pomoćnim metodama (poglavlje 2.), odabranim pristupom (poglavlje 3.), a zatim će biti opisana programska implementacija i eksperimentalni rezultati (poglavlje 4., odnosno 5.).

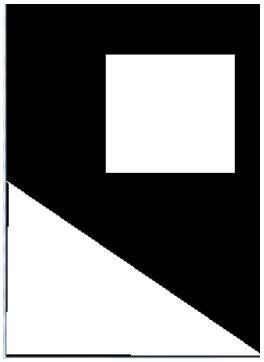
2. Pregled korištenih pomoćnih metoda

2.1. Glađenje

Prvi korak u izvedbi zadatka je proces glađenja. Glađenje je postupak filtriranja slike koji na izlazu daje blago zamagljeniju nego što je to bila početna slika. Glađenjem slike se eliminiraju lažni rubovi na slici što je bitno za kasniju detekciju rubova. Lažne rubove je moguće eliminirati pomoću linearnog niskopropusnog filtra kakav je Gaussov filtar. Gaussov filtar je u izvornom obliku dvodimenzionalan i koristi Gaussovu funkciju normalne raspodjele kako bi njome popunio matricu koju će primijeniti na piksele slike. Konačna vrijednost svakog piksela se određuje operacijom konvolucije na način da svaki element dobivene matrice množi odgovarajući piksel u susjedstvu, a zbroj svih tih umnožaka je jednak novoj vrijednosti promatranog piksela. Konvolucijom je svaki piksel određen svojim susjedstvom. Susjedni pikseli koji su bliže promatranom pikselu će imati veći utjecaj na konačnu vrijednost promatranog piksela od onih koji su dalje zbog djelovanja normalne raspodjele. Isti efekt koji se postiže primjenom dvodimenzionalnog Gaussovog filtra je moguće postići sa dva jednodimenzionalna filtra koji se popunjavaju vrijednostima Gaussove jednodimenzionalne funkcije (1). Dvodimenzionalna Gaussova funkcija je jednaka produktu dviju jednodimenzionalnih funkcija (1). Kod ovakvog pristupa slika se prvo sa jednodimenzionalnim vektorom prođe u jednom smjeru, a zatim se na tako izmijenjenu sliku primijeni isti vektor u drugom smjeru. Jednodimenzionalna Gaussova funkcija određuje vrijednosti svakog piksela na temelju njegovog susjedstva i time je intenzitet svakog piksela usklađen s okolinom.

Analitički, jednodimenzionalna Gaussova funkcija može se izraziti sljedećim izrazom:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (1)$$



Slika 1: testna slika napravljena u Paint-u

Slika 2: uzorci slike 1 zaglađeni sa različitim parametrom sigma (2, 3 i 5)

U izrazu (1), parametar σ je standardna devijacija Gaussove funkcije, x je udaljenost od centralnog piksela po x osi ili po y osi ovisno da li se vrši horizontalno ili vertikalno glađenje. Duljina vektora jednodimenzionalne Gaussove funkcije treba biti primjerena jer utječe na kvalitetu i brzinu izvođenja glađenja slike i neparna zbog jednoznačnog određivanja centralnog piksela. Gaussova razdioba je definirana na beskonačnom intervalu, ali je taj interval kod realne implementacije potrebno ograničiti. Zato se koristi preciznost od 3σ (3 sigma) koja osigurava točnost Gaussove razdiobe od 99.73%. Uzimajući u obzir oba intervala duljine 3 sigma i pazeći da duljina vektora mora biti neparna, konačni izraz za određivanje duljine vektora d je sljedeći:

$$d = 1 + 2*\text{round}(3*\sigma) \quad (2)$$

Poželjno je da se vrijednosti unutar vektora podese na način da njihov zbroj iznosi jedan jer u protivnom originalna slika gubi na svjetlini zbog gubljenja dijela vrijednosti koeficijenata. Da bi ukupan zbroj koeficijenata iznosio jedan svaki element vektora se dijeli s ukupnim zbrojem. Glađenje se vrši konvolucijom vektora sa slikom prolaskom vektora duž cijele slike vodoravno redak po redak počevši od donjeg lijevog kuta pa sve do gornjeg desnog dva puta, jednom vodoravno okrenutim vektorom, a zatim okomito okrenutim vektorom koristeći vodoravno zaglađenu sliku kao ulaz. Gaussov vektor veće duljine smanjuje brzinu izvođenja algoritma te uzrokuje veću zamagljenost izlazne slike, povečavajući tako grešku lokalizacije, ali i smanjujući utjecaj štetnog šuma.

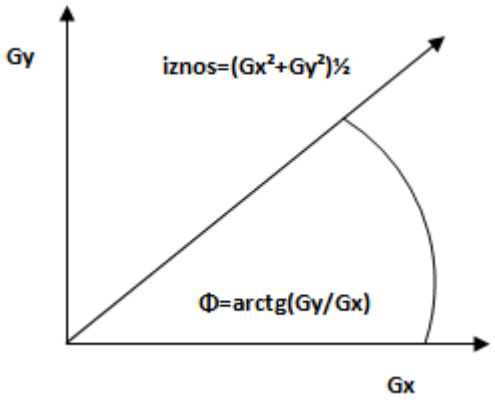
2.2. Određivanje gradijenta

Promotrimo sliku kao diskretiziranu skalarnu funkciju dvije varijable. Sada gradijent slike možemo definirati kao vektor parcijalnih derivacija funkcije po slobodnim varijablama x i y. Za svaki piksel određuje se iznos (amplituda) vektora gradijenta i pripadajući smjer pružanja. Pri izračunu iznosa gradijenta (3) koristimo podatke njegovih susjeda neposredno lijevo i desno, odnosno neposredno iznad i ispod promatranog piksela. Vodoravnu komponentu iznosa gradijenta (Gx) svakog piksela dobivamo kao apsolutnu vrijednost razlike svjetlina njegovog lijevog i desnog susjeda. Okomitu komponentu iznosa gradijenta (Gy) dobijemo kao apsolutnu vrijednost razlike svjetlina susjednih piksela iznad i ispod promatranog piksela. Nakon izračuna obiju vrijednosti (Gx i Gy) izračunavamo iznos vektora gradijenta prema formuli:

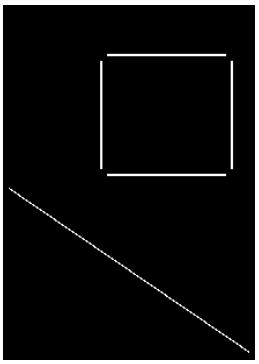
$$G = [(Gx*Gx) + (Gy*Gy)] \hat{}(1/2) \quad (3)$$

Kada su nam poznate obje komponente iznosa gradijenta, smjer pružanja gradijenta u radijanima možemo dobiti koristeći funkciju arkus tangens:

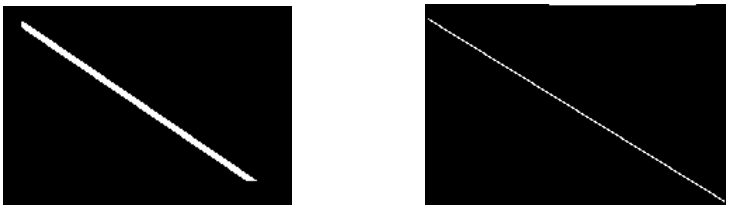
$$\Phi = \text{arctg}\ (Gy/Gx) \quad (4)$$



Slika 3: iznos i smjer pružanja gradijenta



Slika 4: slika amplituda gradijenata piksela sa slike 1



Slika 5: prikaz amplitude gradijenata donjeg dijela slike 4 prethodno zaglađene sa `sigma` = 2, odnosno `sigma` = 8

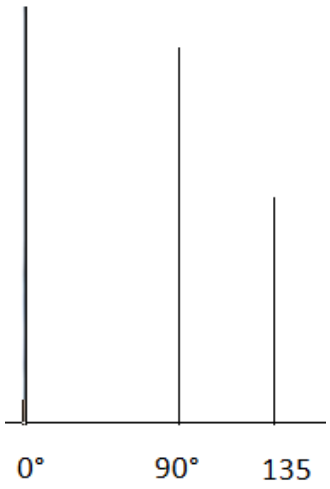
3. Odabrani pristup pronalaženja prometnog traka

3.1. Analiza histograma gradijenta

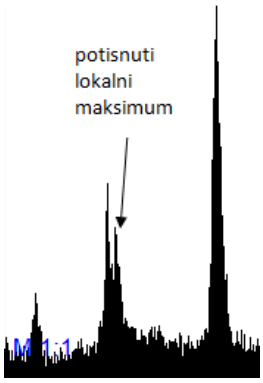
Histogram je diskretna distribucija frekvencija ishoda promatranog procesa nad domenom promatranog parametra. Histogram gradijenta je polje čiji elementi kazuju koliko piksela iz promatrane populacije ima smjer gradijenta u odgovarajućem intervalu. Moguće je umjesto ukupnog broja piksela u svaki element histograma pohraniti ukupnu sumu iznosa gradijenata svih piksela koji su glasovali za taj smjer pravca. Time bi se mogli eventualno eliminirati pikseli sa malim iznosom gradijenta, a čiji smjer gradijenta je odgovarajućeg iznosa.

Na početku našeg odabranog postupka vrši se glađenje ulazne sive slike, a zatim se izglađena slika podvrgava određivanju gradijenta. U postupku određivanja gradijenta se na slici za svaki piksel odredi iznos (amplituda) i smjer pružanja gradijenta piksela. Zatim se izračunava prag i stvara binarizirana slika gdje će se svi pikseli sa iznosom gradijenta većim od tog praga postaviti na 255, a manjim na 0. Izračunavanje praga je objašnjeno u poglavlju 4.3. Binarizacija i sve predstojeće faze postupka se provode samo na donjoj trećini slike jer je za očekivati da su tamo linije traka dominantni pravocrtni segmenti. Zatim svaki piksel binarizirane slike čija je vrijednost 255 glasuje prema iznosu smjera svoga gradijenta u odgovarajuće polje histograma smjerova gradijenta. Indeksi histograma smjerova se kreću od 0° do 179°. Element histograma s indeksom „i“ odgovara intervalu [i, i+1] u stupnjevima. Po

završetku glasanja u histogramu smjerova će uglavnom najveće vrijednosti imati upravo oni elementi čiji su indeksi jednaki smjerovima linija traka. Na početku analize histograma gradijenata je korisno eliminirati jako male ili jako velike smjerove za koje je sigurno da ne odgovaraju pružanju nijedne linije traka. Uvjet koji mora zadovoljiti smjer linije je da je on maksimum u nekom lokalnom okruženju histograma smjerova (Slika 7). Zanemarivanjem prethodnog uvjeta i uzimanjem elemenata histograma sa najvećim iznosom će rezultirati time da će susjedni smjerovi najdominantnijeg smjera biti krivo interpretirani kao ostale linije ceste. Nužno je da odabrani smjerovi histograma budu za određeni iznos razmaknuti jer linije ceste zbog kuta kamere ne mogu biti niti preblizu niti predaleko. Nakon što su svi odabrani smjerovi zadovoljili spomenute uvjete slijedi jednodimenzionalna Houghova transformacija. Indeks elementa histograma koji sadrži globalni maksimum predstavlja najbolju procjenu vrijednosti parametra α i ta vrijednost se uzima za daljnju obradu.



Slika 6: histogram smjerova gradijenata piksela sa slike 4



Slika 7: primjer potiskivanja lokalnog maksimuma

Slika 7 pokazuje kako se u histogramu uz glavne dominantne smjerove može pojaviti susjedni vrh koji je lokalni maksimum u svome odgovarajućem susjedstvu. Vrhovi histograma koji pripadaju pravim smjerovima linija traka moraju biti lokalni maksimumi u svome odgovarajućem susjedstvu koje je za potisnuti vrh sa slike 7 prema postavkama parametara algoritma bilo premalo.

3.2. Jednodimenzionalna Houghova transformacija

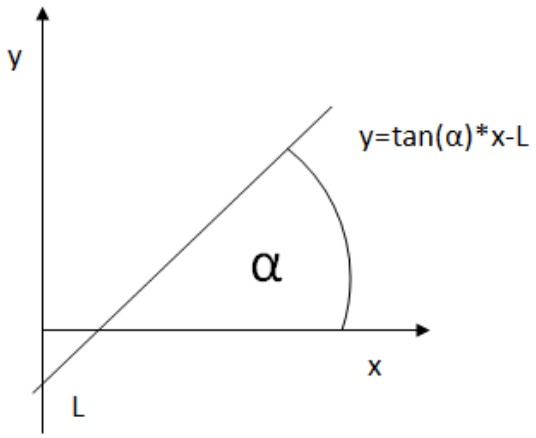
Houghova transformacija [8] je tehnika kojom se pronalaze parametri određenog matematičkog izraza kojim je opisan traženi geometrijski lik sa slike. To je metoda koja je primjenjiva u pronalaženju proizvoljnih oblika. Houghova transformacija za pravce je u svom izvornom obliku 2D i opisana je sljedećim izrazom:

$$\rho = y*\cos(\alpha) - x*\sin(\alpha) \qquad (5)$$

koji se može napisati u obliku jednadžbe pravca

$$y = x*\sin(\alpha)/\cos(\alpha) - \rho/\cos(\alpha)$$

$$y = \tan(\alpha)*x - L$$



Slika 8: elementi pravca

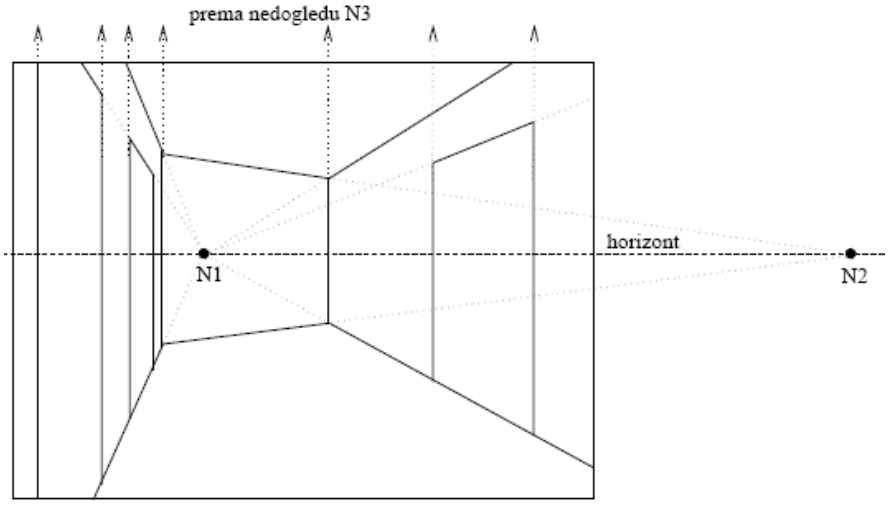
Općeniti postupak procjene parametara dvodimenzionalnom Houghovom transformacijom zahtijeva da se na početku odrede rasponi vrijednosti, odnosno moguće domene parametara ρ i α. Za programsku implementaciju vrijednosti parametara su kvantizirane. Nakon što su nam poznate domene, formira se dvodimenzionalno akumulatorsko polje koje je zapravo kartezijev produkt domena parametara ρ i α. Onda dolazi vremenski najzahtjevniji dio postupka, a sačinjava ga kontinuirano prolaženje kroz piksele koji su dio rubova elemenata sa slike. Svaki piksel čije x i y koordinate za odgovarajuće parametre α i ρ zadovoljavaju izraz (5) glasuje za taj par parametara koji prema izrazu (5) formiraju jednadžbu lika kojemu taj piksel pripada. Nakon što je svaki piksel glasovao za svoj par parametara u dvodimenzionalnom polju, svaki element će sadržavati ukupan broj piksela koji je glasovao baš za taj par parametara. Na kraju se jednostavno u akumulatorskom polju pronađe element sa najvećom vrijednošću broja piksela te se parametri ρ i α za taj element uzimaju za konačne vrijednosti. Prednosti Houghove transformacije su mogućnost pronalaženja parametara i kod nepotpunih i netočnih slika rubnih elemenata te što nije potrebno imati predznanje o položaju pravocrtnih segmenata u analiziranoj slici.

U ovom radu će se pokušati pronaći pravci koji predstavljaju rubove prometnih trakova. Poslije analize histograma gradijenta za svaki pronađeni smjer se traži jednadžba pravca prema (5). Transformacija koja će se koristiti u ovom radu je jednodimenzionalna jer je parametar α smjer određen prethodnim postupkom analize histograma gradijenata. Houghovom jednodimenzionalnom transformacijom se traži parametar ρ postupkom glasanja piksela u akumulatorsko polje čiji indeksi su jednaki rasponu mogućih vrijednosti parametra ρ pomnoženog sa dva. Domena parametra ρ se kreće od 0 do ±`diagonal` gdje je `diagonal` veličina dijagonale slike. Veličina akumulatorskog polja je jednaka 2*`diagonal` jer je potrebno mapirati i negativne vrijednosti parametra ρ. Za isti smjer je moguće imati dva pravca sa različitim parametrom ρ. Potom svaki piksel čiji je smjer gradijenta jednak jednom od dominantnih smjerova glasuje u akumulatorskom polju za parametar ρ koji sačinjava jednadžbu pravca koji prolazi tim pikselom. Element akumulatorskog polja je moguće umjesto uvećavanja za jedan uvećati za iznos gradijenta piksela koji glasuje za tu vrijednost parametra ρ. Nakon glasanja akumulatorsko polje bi trebalo sadržavati nekoliko maksimuma čiji indeksi su parametri ρ odgovarajućih pravaca linija traka. Iz smjerova pravaca α i parametara ρ dobivenih analizom histograma gradijenta, odnosno Houghovom transformacijom se jednostavno prema (5) formiraju jednadžbe pravaca detektiranih pravocrtnih segmenata. Među pronađenim pravcima se možda nalaze i oni koji ne pripadaju linijama traka, a koje ćemo filtrirati metodom opisanom u sljedećem odjeljku.

3.3. Iskorištavanje perspektivnih ograničenja

Određivanje nedogleda pravocrtnih segmenata na slici dobivenoj perspektivnom projekcijom doprinosi boljem razumijevanju trodimenzionalnog prostora. Ta metoda je posebno pogodna za scene sa velikim brojem ravnih paralelnih bridova uz mali skup različitih smjerova pružanja u trodimenzionalnom prostoru. Na temelju poznavanja nedogleda pravaca slike moguće je odrediti

trodimenzionalni smjer pružanja određenih bridova scene. Nedogled je najjednostavnije predstaviti kao točku na slici u kojoj se sjeku svi paralelni pravci koji imaju isti smjer pružanja u trodimenzionalnom prostoru. Nedogledi svih pravocrtnih segmenata sa istim smjerom pružanja u 3D prostoru se nalaze na pravcu koji se zove horizont. Svrha pronalaženja nedogleda u ovom radu je za potrebe eliminacije pravocrtnih segmenata koji imaju isti smjer pružanja kao i linije rubova traka, ali različiti nedogled. Svi detektirani pravocrtni segmenti koji odgovaraju rubovima trakova imaju isti nedogled ili su im nedogledi razmješteni na vrlo bliske međusobne udaljenosti.



Slika 9: primjer zatvorene scene gdje postoje 3 različita nedogleda za poprečne, vodoravne i uzdužne paralelne pravce (preuzeto iz [3])

Na slici 9 se jasno mogu pravocrtni segmenti podijeliti u 3 skupine prema položaju nedogleda. Svi paralelni bridovi u sceni se sjeku u nekoj točki prostora koja je za svaku skupinu različita, ali se sve te točke nalaze na istom horizontalnom pravcu. Jedna skupina uzdužnih segmenata ima nedogled smješten u točki N1, druga skupina poprečnih segmenata ima nedogled u točki N2, dok je nedogled vertikalnih segmenata u smjeru točke N3. Kada bi uzdužni pravci na slici 9 odgovarali linijama traka onda bi se određivanjem nedogleda jednostavno eliminirali poprečni pravci čiji je nedogled N2 različit od nedogleda N1.

4. Programska implementacija

Postupak je implementiran u klasi `alg_histogram_sime` u metodi `alg_histogram_sime::process` koja je deklarirana kao:

```
void alg_histogram_sime::process(
  const img_wrap& src,
  const win_event_vectorAbstract&,
  int);
```

4.1. Glađenje

Funkcija u kojoj je implementirano glađenje je deklarirana kao:

```
int filterGaussSeparatedDouble(
      double sigma,
      const img_wrap& imgGray,
      img_wrap& imgFirstPassSmooth,
      img_wrap& imgSmooth);
```

Funkcija na ulazu prima parametar `sigma` koji odgovara Gaussovoj standardnoj devijaciji, ulaznu sivu sliku sa vrijednostima boje u rasponu od 0 do 255 i dvije slike gdje će se u prvu sliku spremiti ulazna siva slika izglađena sa vertikalnim Gaussovim filtrom, a u drugu obrađena siva slika dodatno izglađena sa horizontalnim filtrom. Funkcija `filterGaussSeparatedDouble` stvara jednodimenzionalni Gaussov filtar pozivom funkcije `makeGaussKernel` koja je deklarirana kako slijedi:

```
void makeGaussKernel(
      int szKernel,
      double sigma,
      std::vector<double>& vektor);
```

Ova funkcija prima parametar `szKernel` kao veličinu Gaussovog vektora, vektor u koji će spremiti koeficijente glađenja izračunate pomoću Gaussove funkcije raspodjele (1) koristeći pri tome i ulazni parametar `sigma` za standardnu devijaciju. Zatim pozivajuća funkcija sa dobivenim filtrom prolazi po ulaznoj sivoj slici prvo vertikalno pa onda horizontalno poštivajući margine jednake polovici veličine filtra. Funkcija vraća marginu koja je jednaka `szKernel`/2, a preko reference izglađenu sliku.

4.2. Izračun gradijenta

Funkcija koja izračunava iznos i smjer gradijenata je deklarirana na sljedeći način:

```
int gradientDouble(
      int margin,
      const img_wrap& imgSmooth,
      img_wrap& imgGrad,
      img_wrap& imgPhi);
```

Funkcija prima izglađenu sliku na kojoj vrši izračunavanje amplitude i smjera gradijenta za svaki piksel uz uvećanje margine za jedan jer je prema jednadžbi (3) za računanje amplitude potrebno gledati susjedne piksele, a istovremeno ostati u izglađenom prostoru slike. Preko referenci funkcija vraća slike amplitude, odnosno smjera gradijenata, a inkrementiranu marginu kao pravu povratnu vrijednost.

4.3. Postupak binarizacije

Funkcija koja stvara binariziranu sliku je deklarirana na sljedeći način:

```
void binarize(
      const img_wrap& imgSrc,
      double consideredImagePart,
      int margin,
      int threshold,
      img_wrap& imgBinarized);
```

Funkcija na temelju slike amplituda gradijenata stvara binariziranu sliku, sliku koja će sadržavati samo vrijednosti 0 i 255, tako da rubovima proglašava one piksele čiji je iznos gradijenta veći od prethodno određene `threshold` granice. Parametar `consideredImagePart` je inicijalno postavljen na 0.33 kako bi pri obradi slika gledali samo donju trećinu slike jer ćemo tako zanemariti gornji dio slike gdje ima manje značajnih pravocrtnih segmenata. Funkcija koja izračunava prag te ga vraća kao povratnu vrijednost je:

```
int binaryThreshold(
      int margin,
      const img_wrap& imgSrc,
      double binParam);
```

Ova funkcija prema ulaznoj slici stvara histogram iznosa gradijenata u kojem bilježi u koliko piksela je zastupljen pojedini iznos gradijenta. Parametar `binParam` određuje koji postotak piksela od ukupnog broja na slici će se eliminirati, odnosno koja vrijednost iznosa gradijenta će postati granica. Prethodno opisani postupak obavlja funkcija:

```
int findThValFromThSum(
    std::vector<int>& histogram,
    double thSum,
    double th_Rel);
```

Funkcija uzima histogram raspodjele ukupnog broja piksela po svim mogućim iznosima gradijenta izvorne slike, parametar `thSum` koji predstavlja ukupan broj piksela izvorne slike bez margina te `th_Rel` koji je jednak parametru `binParam`. U petlji se ide po svim elementima histograma čije se vrijednosti zbrajaju sve dok ukupan zbroj ne postane veći od ukupnog broja piksela pomnoženog sa `th_Rel` parametrom. Kad se to dogodi, indeks zadnjeg elementa histograma prije izlaska iz petlje će biti vraćen kao povratna vrijednost. Funkcija `findThValFromThSum` se interno poziva unutar funkcije `binaryThreshold`. Njena povratna vrijednost je zapravo iznos gradijenta kojim će se kod binarizacije eliminirati pikseli sa iznosom gradijenta manjim od tog.

4.4. Određivanje histograma gradijenata

Ovu funkcionalnost je omogućila sljedeća funkcija:

```
void makeGradientDirectionHistogram(
  img_wrap& imgBinarized,
  img_wrap& imgPhi,
  int margin,
  std::vector<int>& histo);
```

Funkcija pretražuje binariziranu sliku `imgBinarized` i traži piksele koji su u njoj označeni sa vrijednošću 255. Vektor `histo` je prethodno postavljen na veličinu raspona kuteva od 0° do 180°. Za svaki piksel binarizirane slike sa vrijednošću 255 će se iz slike `imgPhi` izračunati smjer gradijenta tog piksela. Na temelju smjera gradijenta će se dobiti smjer pružanja pravocrtnog segmenta kojem taj piksel pripada

te će se uvećati odgovarajuće polje u vektoru `histo`. Elementi vektora `histo` će na kraju sadržavati ukupan broj piksela koji pripadaju pojedinom smjeru pružanja pravocrtnih segmenata. Za očekivati je da indeksi elemenata čije su vrijednosti maksimumi među elementima vektora `histo` predstavljaju dominantne smjerove pružanja pravocrtnih segmenata na slici.

4.5. Pronalaženje dominantnih smjerova iz histograma gradijenata

Funkcija pomoću koje filtriramo dominantne smjerove pružanja pravocrtnih segmenata je deklarirana kao:

```
void findMaxGradient(
       const std::vector<int>& histogram,
       std::vector<Line>& lines);
```

Funkcija prima vektor smjerova gradijenata i u njemu pronalazi elemente sa najvećim vrijednostima koji moraju zadovoljavati uvjete da su maksimumi u nekom svome susjedstvu (u radu je parametar `maxArea` postavljen na 8) i da su indeksi tih elemenata razmaknuti za određeni iznos (u radu je parametar `minLineDistance` postavljen na 10). U postupku traženja zanemareni su neki jako mali ili jako veliki indeksi, odnosno smjerovi (u radu su parametri `minAngle` i `maxAngle` postavljeni na 5, odnosno na 175). Da bi smjer bio prihvaćen potrebno je da za njega glasuje zadani minimalan broj piksela (u radu je parametar `minDirectionPixels` postavljen na 45). Parametar `lines` je vektor čiji su elementi tipa strukture `Line` koja sadrži parametre pravca. U ovom koraku u vektor `lines` spremamo pronađene smjerove. Smjerovi se pronalaze tako da se ide silazno od elemenata histograma sa većom vrijednošću prema onima sa manjom. Potraga staje u trenutku kada vrijednost trenutnog elementa histograma kojeg ispitujemo padne ispod vrijednosti parametra `minDirectionPixels.` Kako svaka linija traka ima i lijevi i desni rub, tako je u funkciji omogućeno pronalaženje smjera pružanja oba ruba linije traka. Drugi rub linije

traka je tražen u nekom ograničenom susjedstvu prvog ruba linije traka (u radu je parametar `edgeSearchArea` postavljen na 5) s time da je određena minimalna dopuštena razlika vrijednosti tih dvaju smjerova (u radu je parametar `minEdgeDistance` postavljen na 3).

4.6. Houghova jednodimenzionalna transformacija nad dobivenim orijentacijama

Funkcija koja vrši jednodimenzionalnu Houghovu transformaciju je deklarirana kao:

```
void Hough1DTransform(
      int margin,
      double consideredImagePart,
      const img_wrap& imgGrad,
      const img_wrap& imgBinarized,
      const img_wrap& imgPhi,
      std::vector<int>& accField,
      std::vector<Line>& lines);
```

Funkcija iterira po dominantnim smjerovima orijentacije gradijenta koji su pronađeni u prethodnom koraku. Za svaki pronađeni smjer se prolazi kroz dio slike određen parametrom `consideredImagePart` te svaki piksel čiji je smjer gradijenta jednak trenutnom promatranom smjeru glasuje u akumulatorskom polju za odgovarajući iznos parametra ρ prema jednadžbi (5). Veličina akumulatorskog polja je određena veličinom dijagonale slike puta dva jer je potrebno mapirati i negativne vrijednosti parametra ρ. U akumulatorsko polje se umjesto ukupnog broja piksela koji su glasovali za određenu vrijednost parametra ρ upisuje suma iznosa gradijenata piksela. Parametar ρ se za svaki pojedini dominantni smjer pružanja pronalazi kao indeks čiji element u akumulatorskom polju sadrži globalni maksimum vrijednosti. Vektor `accField` služi za spremanje ukupnog broja piksela koji su glasovali za pojedini ρ kroz sve iteracije pronađenih smjerova pružanja.

Slika 10: binarizirana slika donje polovice prometnog traka

Slika 11: histogram smjerova gradijenta slike 10

Slika 12: izlaz Houghove transformacije za sliku 10 je histogram broja piksela za određeni parametar ρ

Na slici 10 se može vidjeti da dominantne smjerove određuju kutevi pružanja lijeve i desne linije prometnog traka. Na slici 11 je dan prikaz sadržaja histograma gradijenta gdje dva dominantna vrha odgovaraju smjerovima orijentacije lijeve i desne

linije traka iznosa 50°, odnosno 140°. Nakon što su analizom histograma gradijenta pronađeni dominantni smjerovi vrši se jednodimenzionalna Houghova transformacija. Svaki piksel koji ima smjer gradijenta jednak trenutno promatranom smjeru glasuje u akumulatorskom polju za parametar ρ koji pripada pravcu koji prolazi tim pikselom. Na kraju glasovanja za jedan određeni smjer akumulatorsko polje će sadržavati, kao što se može vidjeti na slici 12, jedan dominantni vrh čiji indeks odgovara vrijednosti parametra ρ najdominantnijeg pravca za razmatranu orijentaciju. Histogram vrijednosti parametra ρ sa slike 12 je dobiven razmatranjem najdominantnijeg smjera na slici 10, a to je smjer desne linije traka.

4.7. Korištenje nedogleda u filtriranju lažnih smjerova

Funkcija koja obavlja traženu funkcionalnost je deklarirana kao:

```
void farAwayLookFunc(
       int imgWidth,
       int margin,
       int yHorizont,
       const img_wrap& imgSrc,
       std::vector<Line>& lines);
```

Parametar `yHorizont` na kojemu se nalaze svi nedogledi linija prometnih trakova je određen postupkom procjene y koordinate sjecišta dvaju pravaca koji odgovaraju linijama prometnih trakova na slijedu od nekoliko slika. Ta vrijednost horizonta je korištena kako bi se svakom pravcu iz vektora `lines` odredio nedogled, tj. x koordinata nedogleda. Pod pretpostavkom da prvi pravac iz vektora `lines` stvarno odgovara jednoj od linija prometnog traka, svi ostali pravci će biti proglašeni linijama ako je apsolutna vrijednost razlike njihove x koordinate nedogleda i x koordinate nedogleda prvog pravca manja od nekog iznosa (u radu je parametar `maxFarAwayLookDistance` postavljen na 40). Parametar `imgWidth` predstavlja širinu ispitne slike. Ovom funkcijom je implementirano da se eliminiraju pravci čija je x

koordinata nedogleda manja od `imgWidth`/4 ili veća od 3*`imgWidth`/4 jer je eksperimentalno utvrđeno da se x koordinate nedogleda linija ceste uvijek nalaze oko centra slike, odnosno `imgWidth`/2. U daljnjoj filtraciji su eliminirani svi pravci koji duž svog smjera širenja nemaju određeni kontinuitet piksela sa vrijednošću 255 (u radu je parametar `minPixelContinuity` postavljen na 45) kako bi se eliminirali pravci koji ne predstavljaju linije traka. Implementirano je da se ne iscrtavaju oba ruba linije ceste čije se sjecište sa x osi za određeni iznos (u radu je parametar `maxSdEdgeDeviation` postavljen na 5) nalazi van dimenzija slike jer su te linije uske te bi iscrtavanje oba ruba uzrokovalo nepreglednost.

4.8. Rad u razvojnom i programskom okruženju

Za potrebe implementacije ovog postupka korišteno je programsko okruženje Visual C++ 2008 Express Edition. Parametri naredbenog retka se u ovom programskom okruženju unose tako da se klikne desnom tipkom miša na projekt i zatim ide na *Properties → Debugging → Command Arguments* [4].

U naredbeni redak se može unijeti nekoliko različitih parametara:

- putanja izvorne datoteke (- sf)
- odredište izlazne datoteke (- df)
- algoritam koji se koristi (- a)
- interakcija za vrijeme izvođenja (- i)
- konfiguracija parametara (-c)

Prilikom izvođenja algoritma u razvojnom okruženju (ljusci), postoje opcije koje se mogu koristiti u korisničkom sučelju kao:

- „p n“ (obradi sljedeću sliku)
- „p p“ (obradi prethodnu sliku)
- „p a x“ (idi na x-tu sliku)
- „p t“ (obradi trenutnu sliku)

Postoji također i mogućnost konfiguracije trenutnog algoritma putem korisničkog sučelja bez potrebe za ponovnim pokretanjem na način da mijenjamo vrijednosti parametara algoritma.

5. Eksperimentalni rezultati

Cilj eksperimentiranja nad određenom skupinom slika je prikazati dobre i loše ishode primjene algoritma na nekim situacijama iz stvarnog života. Algoritam smo u nastavku testirali na većoj skupini slika na kojima se nalaze prometnice iz ruralnih krajobraza. Slike su snimljene po danu i po lijepom vremenu. Na takvim slikama su najčešći elementi stabla, travnate površine, prometni znakovi. Kut kamere kojom su snimljene ove slike je fiksan i ne mijenja se za vrijeme trajanja cijelog slijeda slika. Brzina auta iz kojeg su slike snimane je za vrijeme snimanja bila dovoljno dobra da nijedna slika nije zbog prevelike brzine ostala zamućena. Pošto se testiranje algoritma zapravo vrši nad slikom iznosa gradijenata izvorne slike u sivom rasponu boja (0 – 255), najveći problem će nam predstavljati površine ceste osvijetljene jakom sunčevom svjetlošću gdje je kontrast između linije traka i samog traka jako mali.

Ulazni parametri o kojima najviše ovisi uspješnost algoritma su `sigma`, `consideredImagePart, binParam` i `threshold`. `Sigma` predstavlja standardnu devijaciju kod Gaussove normalne razdiobe i u algoritmu se koristi prilikom glađenja. Vrijednost `sigme` je najčešće negdje oko 1. Povećanjem `sigme` povećava se veličina Gaussovog 1D filtra što kod glađenja uzrokuje povećanje vremena obrade svakog piksela. Uz veću `sigmu` će vrijednost svakog piksela kod glađenja biti određena prema vrijednostima piksela iz većeg susjedstva zbog čega će razlike u vrijednostima gradijenta piksela na većem području biti manje izražene. Također zbog ovog efekta rubovi elemenata na slici će biti deblji nego što bi bili da je `sigma` manja. Parametar `consideredImagePart` određuje maksimalnu vrijednost y koordinate piksela koja će se pri obradi uzimati u obzir. Kroz brojne eksperimente uočeno je da je 0.33 najbolja vrijednost pa će se samo gledati donja trećina slike u

kojoj bi trebali biti pravocrtni dijelovi linija traka. Time se eliminiraju brojni šumovi iz gornjih dijelova slike. `Threshold` je najmanja vrijednost iznosa gradijenta piksela koja će biti propuštena, a dobiva se kao povratna vrijednost funkcije `binaryThreshold` kojoj se prosljeđuje parametar `binParam`. Iako se određuje dinamički pokazalo se je da se dosta dobri rezultati dobivaju kada je statički postavljena na 10. Dinamičko određivanje parametra `threshold` je pogodno kada se minimalna vrijednost iznosa gradijenta piksela nastoji prilagoditi svakoj slici posebno respektirajući distribuciju intenziteta piksela ispitne slike. Statičkim postavljanjem zanemarujemo razlike iznosa gradijenta što je na konkretnim slikama korištenim u ovom radu dalo bolje rezultate.

U nastavku su predstavljene slike na kojima je algoritam uspješno ili neuspješno detektirao prometne trakove.

5.1. Uspješno pronalaženje prometnog traka

Ako nije drugačije naglašeno, sljedeći parametri su korišteni kod dobivenih slika:

dinamički parametri:
`threshold` = 10

statički parametri:
`sigma` = 2
`binParam` = 0.95
`consideredImagePart` = 0.33
`minLineDistance` = 10
`minEdgeDistance` = 3
`edgeSearchArea` = 5
`maxArea` = 8
`minAngle` = 5

```
maxAngle = 175
minPixelContinuity = 45
minDirectionPixels = 45
maxSdEdgeDeviation = 5
maxFarAwayLookDistance = 40
```

Na slici 13 se vidi da je algoritam uspješno detektirao lijevu i desnu liniju prometnog traka u kojem se nalazi vozilo unatoč tome što je postojala opasnost da detektira i pravocrtni segment između travnate površine i drveća sa desne strane traka. Travnata površina i drveće su u značajnom kontrastu.

Slika 13: uspješan pronalazak prometnog traka

Na sljedećoj slici (Slika 14) je algoritam pokazao izdržljivost i u situacijama kada sjene objekata i sunčeva svjetlost na cesti stvaraju snažan kontrast. Taj kontrast se na slici gradijenata preslikava u rub koji može izazvati lažan dojam linije traka ako je pravocrtan.

Slika 14: uspješan pronalazak prometnog traka unatoč kontrastu sjena

Slika 15: slika amplituda gradijenata piksela sa slike 14

Slika 16: uspješnost algoritma u prisustvu prometnog znaka kao alternativnog pravocrtnog segmenta

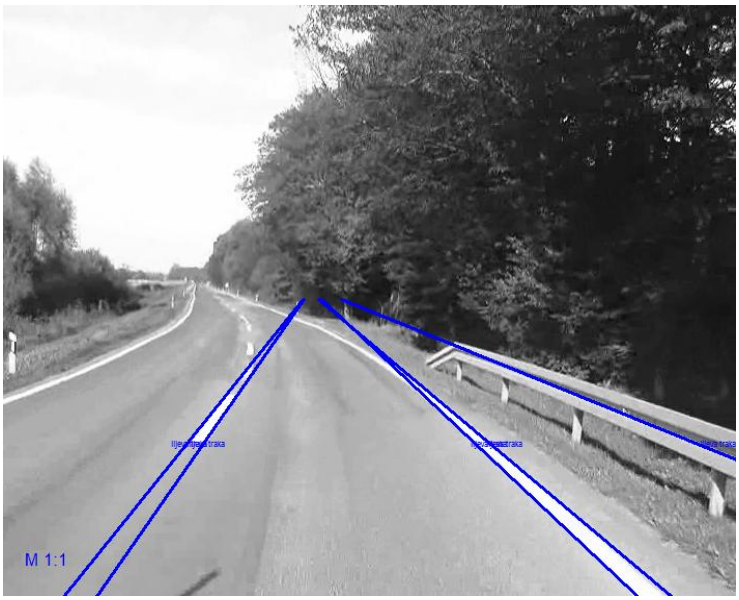
Na slici 16 je algoritam uspio eliminirati stup prometnog znaka čiji pravocrtni segment predstavlja neželjenu liniju. Taj efekt je postignut pomoću metode određivanja nedogleda. Stup prometnog znaka ne može biti smatran linijom jer je njegov nedogled previše udaljen od nedogleda detektiranih linija prometnog traka.

5.2. Pronalaženje lažnih rubova prometnog traka

Situacija na slici 17 pokazuje da algoritam može određena oštećenja na cesti ili prašinu koji su u značajnom kontrastu sa ostatkom površine krivo protumačiti kao linije prometnog traka. To bi se možda moglo riješiti povećanjem `sigme` ili da se odredi minimalna razlika između kuteva pružanja linija traka pri čemu bi se eliminirala ona linija koja je preblizu liniji za koju se sigurno zna da predstavlja liniju prometnog traka.

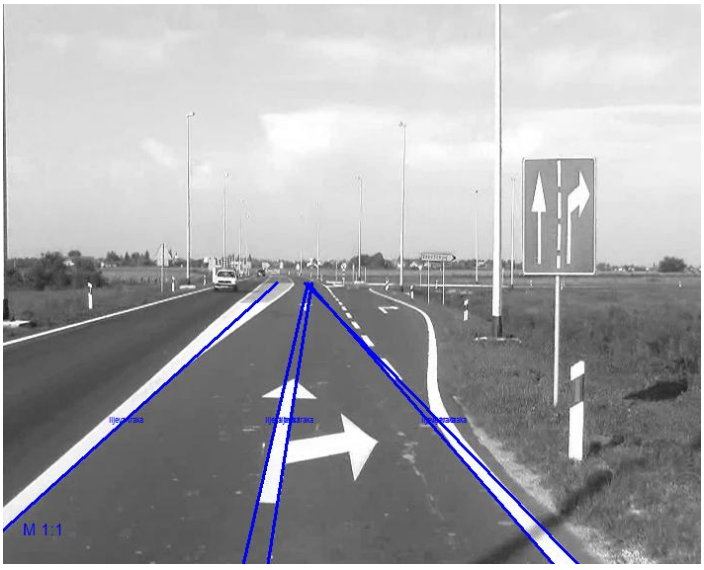
Slika 17: pronalaženje neželjenih linija

Na slici 18 je algoritam prepoznao ogradu uz cestu kao moguću liniju traka. Konkretni problem bi se možda riješio kad bi se tijekom vožnje znalo u kojem se traku vozilo trenutno nalazi . Ako se, npr. kao iznad, nalazi u najdesnijem traku onda bi se eliminirale sve pronađene linije čiji je pozicija na slici desno od traka u kojem je trenutno vozilo.



Slika 18: pojava neželjenih linija

Slika 19 pokazuje još jedno pronalaženje neželjenih segmenata gdje je algoritam pravocrtne znakove na cesti prihvatio kao moguće linije trakova.



Slika 19: kriva interpretacija strelice skretanja kao linije traka

Pomoću slike 20 ćemo demonstrirati zašto je prikladno gledati samo donju trećinu slike. Za scenu na slici 20 je `consideredImagePart` postavljen na 0.5 i algoritam je zbog toga detektirao gornji zakrivljeni dio lijeve linije traka čiji kut pružanja je različit od onog kojeg ima donji linearni dio lijeve linije traka.

Slika 20: pronalazak neželjenih segmenata nakon postavljanja parametra `consideredImagePart` na 0.5

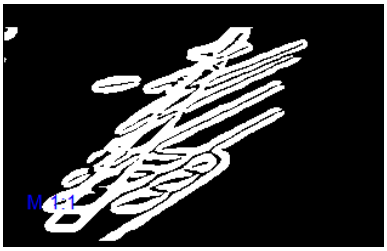
5.3. Propuštena detekcija rubova prometnog traka

Na slici 21 algoritam nije uspio pronaći lijevu liniju traka jer je `consideredImagePart` postavljen na 0.33 zbog čega se obrađuje samo donja trećina slike, a lijeva linija je isprekidana i algoritam trenutno obrađuje prazan prostor između dijelova linije. Rješenje se krije u povećanju `consideredImagePart` da obrađuje veći dio slike, ali to bi možda dovelo do povećanja šuma i detekcije lažnih linearnih segmenata u gornjim dijelovima slike.

Slika 21: neuspješno traženje centralne linije ceste

Slika 22 pokazuje klasičan problem u detekciji linija traka koji se javlja kad na liniju traka padne sjena vozila iz susjednog traka ili sjena statičnog objekta pokraj ceste. Na slici 23 se vidi kako bačene sjene skroz prekrivaju liniju i mijenjaju njen kut pružanja pa je dominantni kut pružanja onaj pod kojim se pružaju sjene. Možda bi se ovaj problem mogao riješiti tako da se linije sjena eliminiraju nedogledom jer će očito njihov nedogled biti negdje na periferiji, a onda bi se u slučaju da za liniju traka nema dovoljno informacija uzela ispravna linija sa prethodnih slika.

Slika 22: propuštena detekcija lijeve linije traka zbog sjene vozila iz susjednog traka



Slika 23: iznosi gradijenata lijeve linije traka sa slike 22

Unatoč tome što je na pojedinim slikama algoritam pronašao krive segmente ili ih nije uopće pronašao, na velikoj većini slika linije ceste su bile uspješno locirane. Pokazalo se je da su najčešći razlozi neuspjeha algoritma prejaka osvijetljenost ceste, sjene objekata koje padaju na cestu te pravocrtni objekti uz cestu poput ograda. Uspjeh postupka analize histograma gradijenata proporcionalno ovisi o jačini kontrasta objekata koje tražimo i njihove okoline, a obrnuto proporcionalno o jačini kontrasta neželjenih elemenata slike i njihove okoline. Jaka sunčeva svjetlost često

umanjuje kontrast bijelih linija i samog asfalta pa tako rubovi linija ostanu nezamijećeni, a kontrast sjena i asfalta je često jači od kontrasta linija i asfalta. Mnogi ovi problemi bi se možda riješili kada bi se znalo koliko traka ima cesta, u kojoj se traci trenutno nalazi vozilo ili kada bi bili poznati kutevi svih linija ceste prije početka vožnje.

Raspodjela vremena pri obradi svake od slika je sljedeća (za `sigma` = 2 i `consideredImagePart` = 0.33):

glađenje	103.5 ms
traženje orijentacije iz histograma smjera gradijenata	88.5 ms
jednodimenzionalna Houghova transformacija	9.4 ms
detekcija i prikaz linija	7.7 ms

Prosječno vrijeme obrade svake slike je oko 0.3 s, dok je algoritam radio prosječnom brzinom od oko 3.9 fps. Vidljivo je da se najviše vremena troši na glađenje gdje se svaki piksel slike obrađuje dva puta. Dolje ćemo pokazati koliko prosječno slika algoritam obradi u sekundi za razne kombinacije vrijednosti parametara `sigma` i `consideredImagePart`. Promjenama vrijednosti tih dvaju parametara raspodjela vremena po osnovnim procesima se nije ništa promijenila. Iz donje sheme je jasno da se povećanjem parametra `sigma` kod glađenja troši više vremena, nego što se troši kad se odlučimo obraditi veću površinu slike.

Tablica 1: broj slika obrađenih u sekundi za kombinacije parametara

sigma	consideredImagePart	fps
2	0.33	3.9
4	0.33	3.4
2	0.5	3.7
4	0.5	3.2

6. Zaključak

Analiza histograma gradijenta je algoritam kojim se mogu detektirati dominantni smjerovi pružanja na slici. Nakon što su pronađeni dominantni smjerovi primjenjuje se jednodimenzionalna Houghova transformacija kako bi se pronašao eksplicitni matematički izraz jednadžbe pravca linije traka. Algoritam na većini slika daje dobre rezultate, ali kod nekih specifičnih situacija potrebno ga je primjenjivati uz neke pomoćne tehnike kao što je određivanje nedogleda.

Uspješnost pronalaska smjerova linija traka analizom histograma gradijenata ovisi o broju ostalih pravocrtnih segmenata na slici koji ne pripadaju linijama traka, a čiji je ukupan broj piksela veći ili jednak broju piksela segmenata linija traka. Ako pravocrtni segmenti koji odgovaraju linijama traka sadrže najveći broj piksela od svih ostalih segmenata tada će njihovi smjerovi biti uspješno prepoznati. No da bi neki segmenti uopće bili prepoznati iznosi gradijenata njihovih piksela moraju proći zadani prag. Tu dolazi do najvećih poteškoća kod analize histograma gradijenata jer ukoliko je prag preniski proći će i neki neželjeni pravocrtni segmenti slabog intenziteta, a ako je previsok može se dogoditi da se linije traka skroz eliminiraju. Analiza histograma gradijenata može biti neuspješna kod pretjerane osvijetljenosti ceste jer se gubi kontrast linija traka i ceste pa visoki prag eliminira segmente slabog intenziteta. Slabi rezultati su prisutni i kad se na cesti nalaze sjene objekata iz okoline čiji je kontrast sa cestom vrlo jak pa bi eventualna primjena visokog praga sigurno eliminirala prave segmente linija. Ukoliko su sjene jačeg intenziteta od linija traka i ako ih velikim dijelom prekrivaju, linije traka je nemoguće pronaći. Najbolji rezultati na svakoj slici ovise o postavkama parametara specifičnima za tu sliku pa je za maksimalnu točnost potrebno za svaku sliku posebno konfigurirati parametre što nije prihvatljivo.

Pronalazak prometnog traka kao polazna točka pruža veliki prostor za daljnji razvoj u smjeru specifičnih potreba suvremenog prometa. U budućem radu razvijeni algoritam bi se mogao koristiti kod detekcije ukupnog broja trakova na cesti ili u postupku razlikovanja iscrtkanih linija od punih linija traka. Ono što bi sigurno pomoglo u navedenim projektima je znanje o položaju traka na nekoliko prethodnih slika.

Literatura

[1] Rosito Jung C., Roberto Kelber C., Lane following and lane departure using a linear-parabolic model, 27.9.2005., http://linkinghub.elsevier.com/retrieve/pii/S0262885605001265, 25.3.2009.

[2] Ballard H.D., Brown M.C., Computer vision: The Hough Method for Curve Detection, New Jersey: Prentice-Hall, 1982.

[3] Šegvić S., Uporaba projekcijske geometrije i aktivnog vida u tumačenju scena, magistarski rad, Fakultet elektrotehnike i računarstva, 2000.

[4] Žabčić A., Pronalaženje prometnog znaka analizom profila slikovnog gradijenta, završni rad, Fakultet elektrotehnike i računarstva, 2009.

[5] Majić A., Pronalaženje prometnog traka korištenjem upravljivih filtara, završni rad, Fakultet elektrotehnike i računarstva, 2009.

[6] Hough transform, *Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/wiki/Hough_transform, 15.4.2009.

[7] Canny edge detector, *Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/wiki/Canny_edge_detector, 9.4.2009.

[8] Šverko M., Houghova transformacija, seminarski rad, Fakultet elektrotehnike i računarstva, 2009.

Pronalaženje prometnog traka analizom histograma gradijenta

Sažetak

Kroz ovaj rad nastojimo pronaći prometni trak u slijedu slika snimljenih iz vozila u pokretu. Predloženi postupak sastoji se iz 2 koraka. U prvom koraku koristimo analizu histograma gradijenta kako bismo pronašli dominantne smjerove pružanja piksela koji bi trebali odgovarati linijama prometnog traka. U drugom koraku uz pomoć jednodimenzionalne Houghove transformacije pronalazi se matematička jednadžba pravca linije traka na način da se iz akumulatorskog polja uzima indeks za kojeg je glasovao najveći broj piksela. Postupak je implementiran u programskom jeziku C++ koristeći ranije razvijene komponente za glađenje i određivanje gradijenta. Razvijeni postupak je evaluiran na slijedu od oko 2000 stvarnih slika. Eksperimentalni rezultati su prikazani i komentirani.

Ključne riječi: računalni vid, detekcija prometnog traka, analiza histograma gradijenta, jednodimenzionalna Houghova transformacija, detekcija dominantnih smjerova

Traffic lane detection by analysing gradient histogram

Abstract

In this work we are trying to find traffic lane in series of images captured from a moving vehicle. Proposed procedure is composed of 2 steps. In first step we are using histogram gradient analysis to find dominant directions of pixel stretchings that should match traffic lane lines. In second step with the help of one-dimensional Hough transformation a mathematical equation of traffic lane line is found by taking an index from an accumulating field that was voted for by most of the pixels.

Procedure has been implemented in the programming language C++ using previously implemented smoothing and gradient calculation methods. Implemented procedure has been evaluated on series of about 2000 real images. Experimental results have been shown and commented.

Keywords: computer vision, traffic lane detection, gradient histogram analysis, one-dimensional Hough transformation, dominant directions detection