

Šiaulių universitetas  
Technologijos fakultetas  
Elektronikos katedra

Giedrius Liubavičius

Autotransporto priemonių numerio  
atpažinimas vaizdų analizės metodu

Magistro darbas

**Vadovas**

dr. Donatas Dervinis

Šiauliai, 2009

Šiaulių universitetas  
Technologijos fakultetas  
Elektronikos katedra

**TVIRTINU**

Katedros vedėjas

Doc. dr. Gintautas Daunys

Autotransporto priemonių numerio  
atpažinimas vaizdų analizės metodu

Magistro darbas

Šiauliai, 2009

## Turinys

Įvadas.....	4
Tikslas.....	4
Uždaviniai.....	4
Apžvalga.....	4
Siūlomas algoritmas.....	22
Struktūrinė schema.....	47
Algoritmo adaptavimas realaus laiko sistemoms.....	48
Algoritmo adaptavimas pravažiavimo kontrolės sistemoms.....	50
Rezultatai.....	53
Išvados.....	55
Literatūra.....	56

## **Ivadas**

Šiai dienai dar nėra standartizuoto autotransporto priemonių valstybinių numerių atpažinimo algoritmo, o dauguma siūlomų algoritmų turi didelę priklausomybę nuo šviesumo, kontrasto ar kadro kokybės. Numerio atpažinimą bandoma įvairiausiai būdais: dirbtinių neuronų tinklų treniravimu, ieškant kontrasto šuolių, pakankamo dydžio bei vienodo intensyvumo (HSI spalvų koduotėje) plotų ar morfologinėmis operacijomis. Daug maž visose metodikose turimi vaizdiniai duomenys verčiami į pilkumo skalę ir po papildomo apdorojamo verčiami į dvejetainę formą, kur vienas taškas gali turėti tik dvi vertes – juodai balta spalvų gama. Tačiau dėl tokio vertimo prarandama informacija, o vaizdas iškraipomas. Siūlomame algoritme naudojama kadro lango, kaip duomenų srauto, koreliacija su atraminiu duomenų srautu – kitu kadru. Ieškomi kadro koreliacijos su vertikalia ir horizontalia rėmo kraštine koeficientų verčių šuoliai. Dėl nedidelio kiekio pagrindinių duomenų operacijų, atsiranda galimybė įgyvendinti algoritmą ar jo dalį aparatūros lygmenyje – reikalingas tik kelių atminties zonų tarpusavio srautinis sumavimas, atimtis ar daugyba.

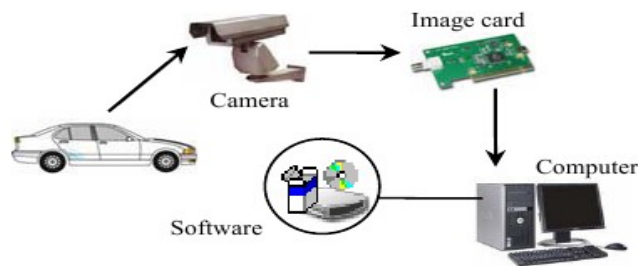
## **Tikslas**

Sudaryti naują autotransporto priemonės registracijos numerio atpažinimo būdą naudojant koreliaciją.

## **Uždaviniai**

- Išanalizuoti egzistuojančius autotransporto priemonės registracijos numerio atpažinimo būdus.
- Sudaryti autotransporto priemonės registracijos numerio atpažinimo būdą, tinkantį naudoti įvairiomis aplinkos sąlygomis.
- Sudarytas būdas atpažinimo rezultatą turi pateikti su atpažinimo tikimybe, kuri leis įvertinti rezultato tikslumą.
- Adaptuoti siūlomą būdą realaus laiko bei pravažiavimo kontrolės sistemoms

## Apžvalga



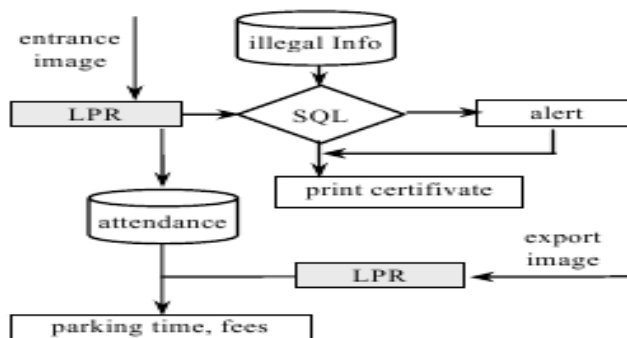
Pav 1. Numerio atpažinimo sistemos struktūra [1]

Šiai dienai dar nėra standartizuoto autotransporto priemonių valstybinių numerių atpažinimo algoritmo. Be to, dauguma siūlomų algoritmų turi didelę priklausomybę nuo šviesumo, kontrasto, nuotraukos kokybės ar purvo.

Tipinė numerio atpažinimo sistema (Pav 1) susideda iš vaizdo kameros, infra-raudonų spindulių imtuvo, plokštės į kompiuterį, kompiuterio bei aptarnaujančios programinės įrangos[1]. Infra-raudonųjų spindulių imtuvas naudojamas autotransporto priemonės buvimui aptikti. Šiam suveikus inicijuojamas vaizdo kadro fiksavimas bei gavimas. Numerio vietos aptikimas labai priklauso nuo aplinkos sąlygų (pavyzdžiui rūko), todėl atpažinimo procesas gali būti iš vis nebaigtas. Dėl to pasiūlytas pastovus automobilio numerio sekimas ar pačios transporto priemonės sustabdymas. Sistemoje naudojant ne vieną kadra, o jų seką, supaprastėja sistemos struktūra – atsisakoma vaizdo fiksavimo plokštės, galima įvertinti judančias kadro dalis, taip sumažinant vertintino vaizdo kadro tarškų skaičių, o tai įtakoja skaičiavimas būtiną laiką. Spalvoto vaizdo apdorojimas perdėm sudėtingas procesas – skaičiavimuose tektų įvertinti bent tris dedamąsias kiekvienam vaizdo kadro taškui. Todėl po kadro gavimo, būtinas vaizdo pradinis filtravimas: kadro konvertavimas į pilkumo skalę (angl. *Grayscale*), išlyginimas, kontrasto didinimas, tačiau jau šioje stadijoje atsiranda nemažai aukšto dažnio triukšmų dėl ko atsiranda ir sistemos priklausomybė nuo apšvietimo. Kontrastas padidinamas turimų 256 pustonų verčių pasiskirstymą išdalinant į 32 vertes tarp 0 ir 255 [1]. Dėl to, kad kadras aprašomas sveikų skaičių masyvu, esant per dideliu ar per mažu apšvietimui, verčių pasiskirstymas atitiks perstumtą Gauso natūralųjį pasiskirstymą, o tai reiškia, kad didinant kontrastą tiek numeris tiek aplinkinis fonas gali patekti į tą patį verčių ruožą ir taip būtų efektyviai panaikintas. Vis dėl to, jei numerio ir aplinkinio fono vertės nėra tapačios, kontrasto didinimas leistų išryškinti numerio vietą, tačiau tik su sąlyga, kad numeris nepatenka į tą patį verčių ruožą kaip ir aplinkinis fonas. Galima būtų ginčytis ar tokios kokybės kadro apdorojimas išvis reikalingas, tačiau taip būtų įvedamas dar vienas

nerikalingas sistemos darbo sąlygų apribojimas. Šioje vietoje susiduriama su ribinėmis vertėmis (angl. *Threshold*) – kaip nustatyti, kurios vertės turi būti verčiamos viena verte, o kurios kita. Egzistuoja trijų tipų ribinės vertės: statinės, dinaminės ir statinės su dinamine korekcija. Ribinės vertės parametro parinkimas (dinaminė), pagal konkretų kadra - brangus skaičiavimo laiko prasme, todėl naudojamas daugiau rankinis parinkimas bei dinaminė korekcija. Šioje vietoje gautą kadra jau galima naudoti numerio rėmui rasti.

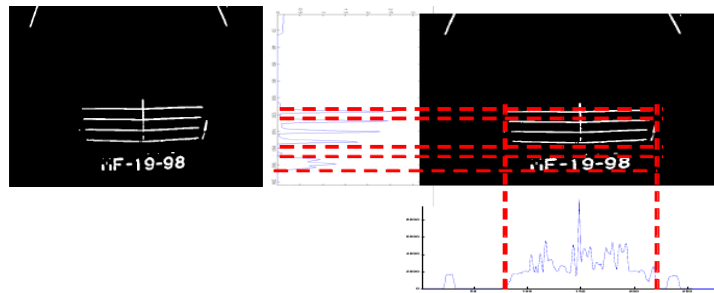
Numerio rėmui aptikti naudojantis Hough transformacija gaunama pernelyg daug skaičiavimų realaus laiko numerio aptikimui, todėl naudojama matematinė morfologija, pasikartojančios informacijos pašalinimui, bei gradiento sekimas, o tai ženkliai sumažina būtinus skaičiavimo pajėgumus. Dėl naudojamų korozijos bei išplėtimo matematinės morfologijos operatorių, dar kartą apribojama sistemos gebama apdoroti kadro kokybė. Šioje



Pav 2. Programinė įranga [1]

vietoje turimas tik neatpažintas numerio rėmas, atpažinimas - didžiausių skaičiavimo pajėgumų reikalaujanti dalis. Tiesioginis numerio atpažinimas reikalauja didelių skaičiavimo pajėgumų, todėl turimas numerio kadras pirmiausiai suskaidomas į atskirus simbolius, o tai ženkliai supaprastina numerio atpažinimo procesą – jau žinomi simbolio rėmai, belieka tik atpažinti, kuris simbolis iškirptas. Turint išankstinių žinių apie numerio formavimą kaip tarpų dydis, santykis su simboliu dydžiu ar simbolių formatas, galima apriboti tikrinamų simbolių aibę ar ištaisyti karpymo klaidą. Atlikus simbolių išskaidymą būtinas skaičiavimo resursas sumažinamas ir lieka tik atskirų raidžių atpažinimas. Simbolių išskaidymas vykdomas pagal paveikslų analizės šablonų atpažinimo teoriją. Vietoje dirbtinių neuronų tinklų (angl. *NN – Neural Network*) ar statistinės analizės, [1] siūlomas pakaitalas – šablonai, simbolio kontūro lyginimas, o tai dar labiau sumažina būtinus skaičiavimo pajėgumus. Praktiškai įgyvendinus šią sistemą (Pav 2), gautas ~98% tikslumas, atpažinimo procesas trunka ~0,5s.

Pagal Macao[2] modelį lygiai taip pat numerio vietos radimui naudojamos matematinės morfologinės operacijos, sistema apribota konkrečiau simbolių formato, numerio



Pav 3. Apdorotas vaizdas bei rėmo paieška projekcijomis [2]

dydžio bei spalvos. Be to, priimta, kad numerio vieta pakankamai gero kontrasto, todėl išblukinus ir atėmus iš pradinio paveikslo, lieka tik didelio kontrasto vietos – numeris, grotelės ir pan. Galima numerio vieta surandama pagal kadro ploto elementų sumines vertes – didžiausia vertė priimama (Pav 3). Tačiau taip gaunamas tik galimas regionas, į kurį patenka ir pašaliniai objektai, todėl tenka vertinti visų turimų kandidatų atitikimą pagal dydį bei kontrastą. Toliau vykdomas simbolių atskyrimas pagal vertikalią projekciją, tačiau taip galima atskirti tik švarius numerius – kai nėra papildomų šiuokščių (purvo) tarp numerio simbolių, tokiu atveju keičiama ribinė vertė. Iš 147 kadrų gaunamas 95% skaidymo simbolių tikslumas. Pagrindinės problemos kyla aptinkant numerio kraštus bei skaidant simbolius, tačiau nevertinant išankstinių metodo apribojimų, šis metodas pakankamai tiksliai aptinka numerio kraštus, ko negalima pasakyti apie daug mažiau problemų keliančią algoritmo fazę – simbolių atpažinimą: panašūs simboliai maišomi (8 su 0 ar B), o gaunamas tikslumas 82,3%.

Literatūroje [3] aprašomas Saudo Arabiškiems numeriams pritaikyta sistema.



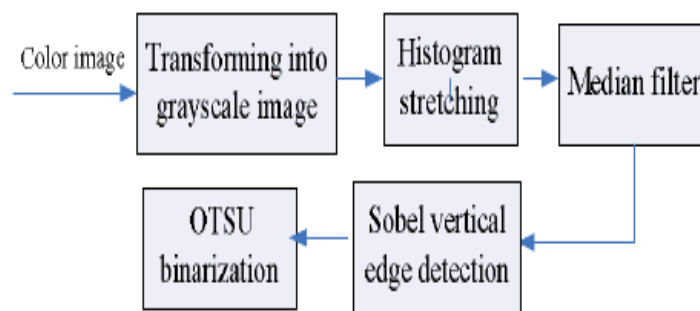
Pav 4. Vertikalūs numerio kraštai [3]

Algoritmo pradžia praktiškai identiška jau aptartiems algoritmams – kadras apdorojamas pilkumo skalėje. Šis algoritmas, kaip ir siūlomoje metodikoje, vertina vertikalius brūkšnius (Pav 4), nes horizontalių brūkšnių natūraliai yra daugiau, numerio vietos radimui. Šiam tikslui naudojami Sobel ir Prewitt algoritmais (vėliau pasirinktas Sobel dėl dinaminių ribinių verčių ir geresnių rezultatų), bei savybinis filtravimas: priimami tik 8 iš eilės einantys taškai,

tinkamo dydžio bei aspekto regionai - vertikalios kraštinės ir horizontalios kraštinės ilgių santykis galimas tik tarp 1.75 – 2.25. Šioje vietoje, jau turint numerio rėmą, 30% viršutinės rėmo dalies nukerpama, taip įvedant papildomų apribojimų pagal vietinių numerių formavimo taisykles. Be to, gautas vaizdas verčiamas į bitinį pavidalą. Toliau skaidoma simboliais, skaičiuojant uždegtų bitų skaičių vertikalios projekcijos kiekviename stulpelyje. Simboliai kerpami tose vietose, kur suskaičiuotas bitų kiekis viršiją ribinę vertę (nenurodyta kaip sudaroma). Iškirpti simboliai normalizuojami taip, kad iš kiekvieno šono nebūtų jokio tarpo, bei keičiamas mastelis iki 40x40 dydžio kadro. Lyginimas, kaip ir anksčiau aprašytais metodais, vykdomas pagal šabloną iš duomenų bazės skaičiuojant Hamming atstumą iš formulės:

$$\sum_{i=0}^{eiluciu\ stulpeliu} \sum_{j=0}^{neatitikimas_{i,j}} \dots (1)$$

Čia neatitikimas yra 1, jei bitas šablone ir kadre sutampa ir 0, jei priešingai. Šios sistemos tikslumas ~96,05%, tačiau tai pirma arabiškiems numeriams atpažinti skirta sistema.



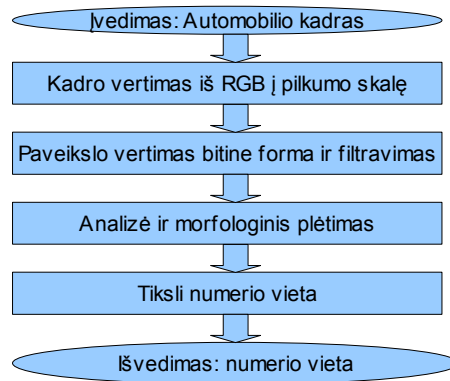
Pav 5. Pradinis filtravimas [4]

Literatūroje [4] pateikiama apmokamų kelių stočiai tinkama atpažinimo sistema. Čia siūloma išfiltruoti judantį objektą pagal turimą fono kadą bei naudojant nekintančią ribinę vertę. Taip skaičiavimai vykdomi tik tada, kai aptinkamas kadre užfiksuotas pašalinis objektas be fono, tačiau toks judančio objekto ieškojimas gali veikti tik turint fono atvaizdą. Be to, judančiam objektui tapus nejudančiu, objektas vistiek būtų fiksuojamas kaip nejudantis kiekvienu sekančiu kadru. Tam pasiūlytas pakeitimas papildomai vertinti prieš tai buvusį kadą kaip foną ir taip eliminuoti nejudančiais tapusius objektus. Kaip ir aukščiau nagrinėtoje literatūroje, numerio vieta randama pasitelkiant matematinės morfologijos operacijas: eroziją, uždarymą, atidarymą. Numerio vieta, po morfologinių operacijų (Pav 5), ieškoma ieškant vertikalių kraštinių. Tol, kol nieko nerandama, mažinama ribinė vertė, naudojama numerio pločiui tikrinti. Jau turint numerio vietą būtinas simbolių atskyrimas, tačiau prieš tai kadras verčiamas į bitinę formą taip, kad visos pilkumo vertės virš 20% priskiriamos 0, o likusios –



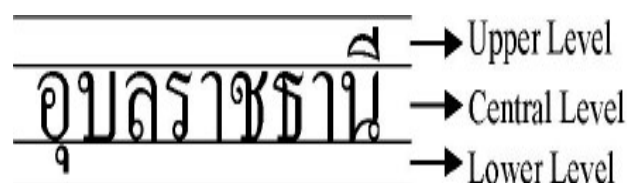
255. Simbolių atskyrimas vykdomas ieškant vientisų linijų bei įvertinant rasto simbolio plotį. Gautas 89,7% skaidymo tikslumas bei 98,4% numerio radimo tikslumas. Radimo bei simbolių skaidymo laikas ~150ms su 1,7G 256M RAM.

Malaizijos teisėsaugos bei plataus naudojimo numerių atpažinimo problematika, kaip alternatyva radiobanginėms identifikacijos kortelėms (angl. *RFID – Radio Frequency Identification Device*), reikalaujančioms brangios įrangos, naudojant numerio atpažinimą iš vaizdo, aptarta [5]. Šio, kaip ir daugelio kitų, autoriaus numerio atpažinimo metodika skaidoma į vaizdo registravimą, tvarkymą, numerio radimą, skaidymą simboliais bei šių atpažinimą. Kaip ir anksčiau paminėtuose straipsniuose, dar kartą patvirtinama koks svarbus tikslios numerio vietos radimas, ženkliai įtakojantis viso likusio proceso tikslumą. Įvardinti keli probleminiai faktoriai, sėkmingam numerio aptikimui: mažas numerio dydis, formatas, stiliaus ar spalvos skirtumai tarp šalių, nėra išankstinių duomenų, kurioje kadro vietoje bus numeris, prastos kokybės nuotrauka bei realaus laiko būtinybė. Iš esmės, aprašytas algoritmas skiriasi tik matematinės morfologijos plėtimo (angl. *dilation*) naudojimu numeriui rasti. Sistema nepritaikyta tarptautiniams numeriams, tačiau tokia galimybė yra ypač lotyniško alfabeto numeriams. Vienas didžiausių šios sistemos atpažinimo minusų - tamprus ryšys su nacionalinių numerių spalviniais bei simboliniais formatais. Kadro vertimas bitiniu pavidalu - svarbiausia šio metodo numerio vietos radimo proceso dalis dėl galimo nedidelio gradiento pasikeitimo tarp simbolių bei simbolių fono, nes sistema ieško pakankamo kontrasto vietų, kaip potencialių numerio vietų, per mažo kontrasto ar apšvietimo. Vertimas į bitinį pavidalą atliekamas naudojantis plačiai taikomu OTSU metodu, dėl pakankamo tikslumo verčiant kadrą į bitinį pavidalą bei didelio pralaidumo. Toliau valomas gautas kadras, eliminuojant visus, susietus su numerio rėmo kraštine, objektus, neįtakojuant simbolių, dėl juos supančio tamsaus fono. Po objektų valymo atliekamas mažų objektų ryškinimas, atsižvelgiant į dydį, bei morfologinė plėtimo operacija, naudojant struktūrinius elementus (angl. *SE - Structured Elements*) atitinkančius 7x28 taškus, objektų sujungimui. Pagal nacionalinius numerius parinkti atskiri struktūriniai elementai dviejų bei vienos simbolių eilutės numeriams. Atlikus morfologinį apdorojimą, rezultatai tikrinami pagal elemento aukščio bei pločio santykį, plotą bei taškų tankį. Jei po šio valymo gaunami keli rezultatai, prioritetas teikiamas objektams, kurie: apatiniuose dviejuose kadro trečdaliuose, atitinka standartinio numerio dimensijas bei toliausiai nuo kadro krašto esančiam objektui. Sistemos galimybės patikrintos su 640x480 dydžio bei 24 bitų spalvų gylio 350 nuotraukų, gautų fotografuojant apytikriai 4 metrų atstumu ir gautas 97,4% patikimumas.



Pav 6. Algoritmo seka [5]

Kiek kitoks nei kituose darbuose siūlymas aprašytas [6]: neatsižvelgti į išankstinius duomenis apie numerį. Numerio atpažinimui labai tiktų komerciniai OCR (Optinis Simbolio atpažinimas, angl. *Optical Character Recognition*) produktų atpažinimo varikliai, iš esmės atpažįstantys tekstą iš šviesaus bei vienodo intensyvumo foto ir tamsaus fono simbolių bei pasiekiantys 98-99% tikslumą. Nors numerio rėmas iš esmės atitinka labai žemos kokybės nuskenuotą tekstą, papildomai užterštą nehomogeniniu daugia-modaliniu intensyvumo pasiskirstymu ir dažnai neprognozuojamu triukšmu, tačiau tokie OCR paketai visiškai nesugeba suskaidyti simbolių. Įvairūs statistiniai duomenys, naudingi teksto skaidymui simboliais, tokie kaip teksto žingsnis (angl. *pitch*), negali būti patikimai prognozuojami tokiam trumpame tekste. Visa tai sąlygoja ženklų OCR pralaidumo sumažėjimą. Skaidymas simboliais kritiškas visam atpažinimo procesui, todėl simbolių skaidymui naudojamas sujungtų komponentų analizės metodas per daug optimistiškas realiomis sąlygomis. Skaidant simbolius naudojamas dinaminis rekursinis numerio filtravimas pagal raidžių susiliejimą: tokiu atveju, mažinama ribinė vertė ir filtruojama dar kartą. Atpažįstant simbolius, atpažinimo proceso efektyvumas padidėja 10-12%, jei vietoje bitinės formos simbolio, atpažinimui naudojamas pilkumo skalės simbolis. Be to, po atpažinimo vykdomas papildomas filtravimas gautų rezultatų patikslinimui ar patikrinimui pagal aprašytus numerių formatus. Sistema išbandyta 500 nuotraukų su Bulgarijos ir Izraelio numeriais bei gautas 81,2% atitikimas aprašytiems numerių formatams, o likusi dalis 100% atmesta. Simbolių atpažinimo tikslumas lyginant su klaida atitinkamai 85,2% ir 3,4%.



Pav 7. Tailando numerio pavyzdys [7]

Dėl didelio simbolių tarpusavio panašumų, [7] atsiranda hierarchinių ryšio koreliacijų (ARTMAP) naudojimas Tailando numeriams atpažinti. Tailando numeriams (Pav7) naudojami nestandartiniai simboliai (Pav 8), be to, simboliai rašomi trimis vertikaliais lygiais, o tai keliai didelių problemų bandant atpažinti įprastais algoritmais. Šiame metode,

Standard Characters	No-head Characters
๕ ๖ ๗	5 5 5
๘ ๙ ๐	๘ ๙ ๐
๑ ๒ ๓	๑ ๒ ๓
๔ ๕ ๖	๔ ๕ ๖

Pav 8. Naudojamų simbolių pavyzdys [7]

atpažinimo algoritmas kiekvieną lygį atpažįsta atskiru dirbtiniu neuronų tinklu. Viršutinio lygio atpažinimui, įvedimo sluoksnyje naudojami 9 segmentai, kurių kiekvienas turi tiek įvedimo kanalų, kiek gaunama savybinių komponentų (angl. *feature*) iš kiekvieno segmento. Antrame paslėptame sluoksnyje kiekvienas komponentas visiškai sujungtas su visais to segmento įvedimo kanalais pirmame sluoksnyje. Trečias blokinis (angl. *cluster*) sluoksnis bei antras sluoksnis aprašomi mokymosi fazėje, kur svorinis koeficientas aprašo ryšį tarp antro sluoksnio segmento bei trečio sluoksnio komponento. Paskutinis ketvirtas sluoksnis – išvedimo, kur kiekvienas komponentas atitinka simbolių klasę. Suskaidžius simbolius, kiekvienas simbolis yra ploninamas juos aprašant 9 kryptiniais kodais. Atlikta šešiolika skirtingų eksperimentų, dviem duomenų bazėms, kiekvieną kartą apmokius atsitiktinai parinktais skirtingais šriftais. Iš 506 numerių, gautas maksimalus 94,96% tikslumas, kai tuo tarpu „Fuzzy ARTMAP“ modelis pasiekė 84,57%, o komerciniai OCR paketai „ArnThai“ bei „Thai OCR“ atitinkamai 49,22% ir 43,88%.

Keleto kitų autorių, bei jau aptarti darbai išanalizuoti [8]. Autoriaus pabrėžta, kad nėra net bendro sutarimo kaip derėtų vertinti ar išbandyti sistemos darbą. Todėl buvo surinkta keletas graikiškų numerių, grupuojant pagal tipą, spalvą, apšvietimą, kampą bei lauko ar vidaus sąlygas. Pateiktas kategorinis literatūroje pateikiamų numerio aptikimų metodų skaidymas: *bitinės formos apdorojimas*, *pilkumo formos apdorojimas*, *spalvotos formos apdorojimas*, *klasifikatorius*.

Bitinės formos apdorojimas – krašto statistikos ir matematinės morfologijos kombinacija. Metodai grindžiami didesniu ir dažnesniu šviesumo pasikeitimu numerio vietoje, tačiau per daug jautrus nepageidaujamiems kraštams, o kaip tik šią problemą pakankamai gerai



Pav 9. Vertikalių brūkšnių paieška skenuojant eilutes [8]

eliminuoja morfologinės operacijos. Kaip sub-kategorija pateikta *sujungtų komponentų analizės metodikos*, analizuojančios jau bitinio pavidalo kadro taškų sujungimą ir pagal tai priskiriančios taškams etiketes. Dažnai vietos, aspekto bei orientacijos tikrinimas integruojamas į numerio vietos radimo algoritmus. Vertikaliems kraštams rasti naudojamas *Sobel operatorius*, gebantis 97% tikslumu apdoroti 384x288 kadra per 47,9ms.

Pilkumo formos apdorojimas suskaidytas į *bendrą vaizdo apdorojimą, dalinę kadro analizę, statistinius matavimus, hierarchinę reprezentaciją, regiono skaidymą, tikimybinį objekto sekimą* bei *kadro transformacijas*.

*Bendra vaizdo analizė* apibūdinama kaip vertikalus kadro skanavimas (Pav 9), ieškant 15 ir daugiau taškų zonoje esančio intensyvumo pasikeitimo nuo minimalios iki maksimalios vertės. Metodas priklauso nuo atstumo iki kameros bei realaus simbolių dydžio.

*Dalinė kadro analizė* apibūdinamas kaip supaprastinta bendro vaizdo analizės atmaina – vykdomas toks pats skanavimas, bet tik kas N-tą stulpelį. Numerio vieta skaitoma ten, kur kraštų skaičius viršija iš anksto numatytą ribinę vertę, nors numerio neradus, vertė mažinama. Tačiau kaip ir aukščiau aprašytas algoritmas, šis taip pat priklausomas nuo kameros nuotolio iki objekto bei objekto dydžio.

*Statistinis matavimas* – blokinis pilko kadro apdorojimas, praktiškai identiškas dviem prieš tai aprašytiems algoritmams, tačiau nepriklausomas nuo dydžio bei tinkamas kai numerio vieta neaiški.

*Hierarchinė reprezentacija* – metodika vektorių kvantavimo pagrindu, kai didesnio kontrasto vietoms suteikiamas mažesnis blokas.



Pav 10. Pradinis ir SCW apdorotas kadras [8]

*Regiono skaidymas* – naudojama adaptyvi vaizdo skaidymo metodika (Pav 10) – slenkantys koncentriniai langai (angl. *SCW – Sliding Concentric Window*), naudojama aprašyti vietinius kadro nukrypimus. Metode kaip euristika naudojamos tokia statistika kaip standartinis nuokrypis ar vidurkis. Metodikos esmė: jei dviejų, nevienodo dydžio, koncentrinų langų, šliaužiančių per kadrą iš kairės į dešinę bei iš viršaus į apačią, vidurkis ar standartinis nuokrypis abiejuose languose viršija nustatyti ribinę vertę - centrinis taškas yra laikomas numerio dalimi. Rezultate gaunamas bitinės formos kadras be pasikartojančių regionų.

*Tikimybinis objekto sekimas* – algoritmas, sudarytas iš kaupimo algoritmo tikimybinio tankio pasiskirstymo bei pustonių skalės optimizavimo pagal diferencinės evoliucijos algoritmą.

*Kadro transformacijos* - Gabor filtrai tekstūros analizei, leidžiantis analizuoti bet kuria neribota kryptimi bei dydžiu, tačiau labai lėtas. Metodikose, kur taikoma Hough transformacija (angl. *HT - Hough Transformation*), pirmiausiai aptinkami kraštai ir tik tada vykdoma transformacija numerio regionus. Transformacijai būtina pernelyg daug skaičiavimo operacijų, todėl taikomas transformacijos ir kontūro algoritmo hibridas, tačiau pati transformacija labai jautri kraštų deformacijoms, todėl bandymai atlikti tik iš labai nedidelio atstumo. Taikant banginę transformaciją (angl. *WT, Wavelet transformation*), transformacijos randamos kontrasto savybės vėliau naudojamos numerio radimui. Taikant apibendrintos simetrijos transformaciją (angl. *GST – Generalised Symmetry Transformation*), gaunamas dviejų taškų tolygios simetrijos savybės. Vienas minusų – kuo mažesnis atstumas, tuo daugiau skaičiavimų.

Spalvotos formos apdorojimas – vertinamos iš anksto žinomos numerio fono bei teksto spalvos, tačiau sistema ypatingai nestabili dėl spalvų įvairovės prie skirtingų apšvietimo sąlygų bei didelio jautrumo deformacijos ar purvui. Suskaidyta sub-kategorijomis: *spalvos modelio transformacija, Fuzzy komplekto teorija, histogramos apdorojimas*.

*Spalvos modelio transformacija* – numeris kraštai ieškomi pagal spalvos pasikeitimus.

*Fuzzy komplektų teorija* – kadras aprašomas šviesumu, tamsia ir šviesia sekos, tekstūra ir geltonumu, numerio vertikalios ir horizontalios pozicijos radimui. Sistema apribota veikimo

nuotolio. Klasifikatorių *skaičiavimo intelektas* - įvairios dirbtinių neuronų tinklų (angl. *ANN – Artificial Neural Network*) architektūros, bendrinis programavimas (angl. *GP – general programming*) bei genetiniai algoritmai (angl. *GA – Genetic Algorithm*). Daugiausiai skaičiavimo resursų reikalaujančios fazės gali būti įgyvendintos diskretinio laiko korinio neuronų tinklo (angl. *DTCNN – Discrete-Time Cellular Neural Network*). Literatūroje aprašytuose algoritmuose, eksperimento sąlygų paruošime, dėl nuspėjamų savybių gavimo, tokių kaip perspektyva, atstumas, apšvietimas, fonas ar objekto pozicija, dažniausiai sąlygos apribojamos iki gerai žinomų darbinių parametrų. Besikeičiančio apšvietimo problemai apeiti buvo sėkmingai pradėti naudoti infra-raudonųjų spindulių moduliai, kurie išvystyti dėl numerio plokštelių atspindinčio paviršiaus. Kai kurie autoriai teigima, kad jų sistemos veikimo ruožas tarp objekto ir kameros labai platus, tačiau dauguma eksperimentų patvirtina, kad visada egzistuoja ryšys tarp tikslumo ir minimalios raiškos [8]. Kai kuriose studijose teigiama apie sėkmingą lokalizavimą net tuomet, kai numerio aukštis tarp 10 ir 16 taškų, tačiau esant tokiai mažai raiškai negalimas patikimas simbolių atskyrimas, todėl šios studijos analizuoja tik numerio radimą. Dauguma atvejų numerio aukštis svyruoja tarp 20 ir 40 taškų, be to, simbolių atskyrimas bus sėkmingas, jei numeris geros būklės, o aukštis bent 20 taškų. Simbolių atskyrimo technika sudalinta kategorijomis, tačiau daugumoje darbų iškart aprašyta keletas metodų, todėl sekanti klasifikacija nėra unikali.

Bitinės formos apdorojimas: *projekcijos ir bitiniai algoritmai, matematinė morfologija, kontūrai*. *Projekcijos ir bitiniai algoritmai* – paprasčiausias ir dažniausiai aprašytas literatūroje metodas. Esmė – sumuojami stulpeliai ar eilutės projekcijai gauti, kurios minimumai leidžia suskaidyti simbolius. Lygiai taip pat intensyviai naudojama ir sujungtų komponentų analizė (angl. *CCA – Connected Component Analysis*) kartu su bitinės formos objektų dydžio, ploto ar orientacijos matavimais. Kitais atvejais, sujungtų komponentų analizė palaikoma vektorinio kvantavimo ar matematinės morfologijos. Dažniausiai sujungtų komponentų analizė keičia taško reikšmę komponentu pagal 8 kaimyninių taškų sujungimą.

*Kontūrai* – kontūro sekimo metodas dar vadinamas vabalo sekimu (angl. *Bug following*), dažnai naudojamas simbolių atskyrimui. Viename darbe siūlyta simbolių atpažinimui panaudoti ryšio koreliaciją, tačiau dėl didelių skaičiavimo sąnaudų, kai 90% laiko sunaudojama ryšio koreliacijų skaičiavimams, metodas netobulintas.

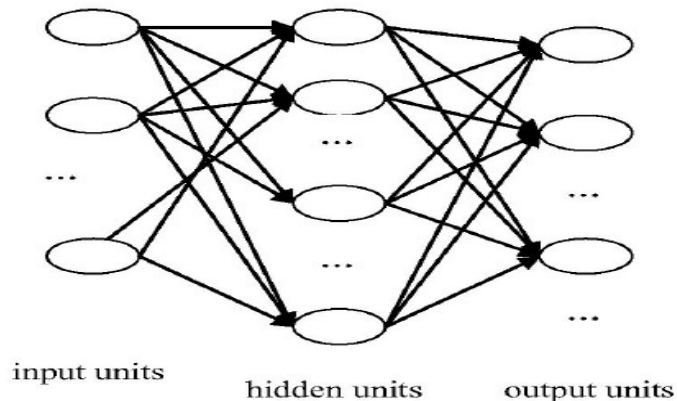
Spalvinis numerio apdorojimas aptartas [9]. Esamų numerio atpažinimo sistemų darbo sąlygos apribojamos konkrečioms sąlygoms kaip: vidaus sąlygos, stacionarus fonas, fiksuotas apšvietimas, riboti greičiai ar net konkretus atstumas iki objekto. Šio darbo pagrindas – kaip įmanoma sumažinti šių apribojimų skaičių. Iš visų darbo sąlygų apribojimų, išorinės sąlygos

Vehicle category	Plate color	Character color	Example
Private automobile	White	Black	<b>E1•2345</b>
Taxi	White	Red	E1•234
Tour bus	Red	White	E1•234
Truck	Green	White	E1•234
Government vehicle	White	Green	E1•234

Pav 11. Vertinamų numerių slapvinimas [9]

bei stacionarus fonas gali būti pagrindiniai faktoriai įtakojantys kadro kokybę, o dėl to ir naudojamų metodikų sudėtingumą. Lauko sąlygų apšvietimas keičiasi dėl paros laiko, oro sąlygų ar pačių objektų. Be to, iš kamerų gaunamas dinaminis vaizdas dėl galimo kameros sukinėjimo, lenkimo ar didinimo. Tokiose nuotraukose gali visai nebūti arba būti daugiau nei vienas numeris skirtingais dydžiais, orientacija ar vieta. Esant sudėtingam fonui, numerio vieta gali būti sunkiai randama. Dažniausiai numerio atpažinimas susideda iš numerio radimo bei identifikavimo, kai numeris randamas pagal dažnai pasitaikančias savybes: forma, simetrija, aspektas, spalva, pilkumas bei intensyvumo pasiskirstymas. Ne laboratorinių sąlygų sistemai reikia naudoti keletą lanksčių, patikimų ir lengvai aptinkamų objektų savybių. Simbolių atskyrimui dažniausiai naudojamos projekcijos, morfologija ar sujungtų komponentų analizė, tačiau projekcijai reikia žinomos orientacijos, o morfologijai būtinas simbolių dydis, nei vienas iš šių netinka. Todėl pasirinktas sujungtų komponentų analizės ir bloko spalvinimo metodų hibridas. Simbolių atpažinimo algoritmų daugybė, tarp kurių: genetiniai algoritmai, dirbtiniai neuronų tinklai, fuzzy vidurkiai, atraminio vektoriaus klasifikatoriai (angl. *SVM - Support vector machine*) ar Markov procesai. Visi algoritmai gali būti skirstomi į ne žingsninius ir žingsninius, kurie pasiekia geresnį tikslumą laiko kaina. Darbe daugiau dėmesio skiriama tikslumui, todėl būtent tikslesnis metodas ir bus naudojamas, jei yra pasirinkimas. Dirbtinio neuronų tinklo ir mechanikos pagrindu sukurtas naujas atpažinimo algoritmas. Numerio vietos radimui pasirinktas spalvos kodo (Pav 11) tikrinimas, kurių gali būti tik keturi (sistema orientuota Taivano numeriams). Numerio krašto paieškai pasirinktas spalvinio krašto ieškojimas: raudona, žalia arba juoda su balta. Be to, tokiam numerio vietos radimui naudojamas HSI spalvų kodavimas, vietoje RGB ar net pilkumo skalės. Tačiau čia pasikeitimai ir baigiasi – tiek numerio skaidymas į atskirus simbolius, tiek atpažinimas nepakitę: lygiai taip pat verčiama į bitinę formą ir lyginama su šablonu - spalva naudojama tik rėmo radimui. Vis dėl to simbolių atpažinimas vietoje pilno viso ploto tikrinimo, vykdomas kontūrai bei papildomai lyginamas skylių kiekis. Dėl problemų atskiriant kai kuriuos simbolių panašumus, naudojamas konfliktuojančių dalių tikrinimas. Galutinis tikslumas: 100% (iš 639 bandymų). Dažniausia klaidų priežastis ne simbolių

atpažinimas, o teisingas jų atskyrimas bei rėmo aptikimas. Šiuo atveju 100% yra aptiktų rėmų tikslumas, 97,6% - rėmo radimas. Šio metodo sparta 0,4s galimų numerio rėmo kandidatų radimui ir 2s (NN dalis) kiekvienam rėmui atpažinti, naudojant Pentium4 1,6GHz CPU.



Pav 12. Naudojamo dirbtinio neuronų tinklo architektūra [10]

Simbolių atskyrimas išnagrinėtas [10], kur simboliams atskirti siūloma naudotis iš anksto žinomais duomenimis: tarpai, plotis, aukštis ar forma, taip įvedant praktiškai laboratorines geras kadro sąlygas. Be to, kaip ir daugelyje kitų darbų, operuojama bitinės formos kadru. Įvertinta skaičiavimo resursų problematika simbolių atpažinimui, kaip sprendimas siūlomas dvigubas tikrinimas: pirmiausia atpažinti charakteringus taškus, filtruoti ir tik tada vykdyti apribotą galutinį atpažinimą. Taip ženkliai padidinamas skaičiavimų pralaidumas bei tikslumas. Simbolių atpažinimo algoritmams rinktasi tarp simbolių charakteringų taškų radimo bei jų apdorojimo atitinkamu dirbtinių neuronų tinklu ir tiesioginio simbolio apdorojimo dirbtiniu neuronų tinklu, kuris ir pasirinktas dėl mažesnių laiko sąnaudų nerenkant charakteringų taškų. Dirbtinių neuronų tinklo pasirinkimą argumentuotas ir tuo, kad galimas prisitaikantis mokymasis bei duomenų savaiminis organizavimas mokymo metu, o skaičiavimus galima vykdyti paraleliai. Be to, ruošiama tokius skaičiavimus palaikanti aparatinė įranga dirbtinių neuronų tinklų įgyvendinimui, o pačios dirbtinių neuronų tinklo struktūros pažeidimas nesunaikina duomenų – skaičiavimas tiesiog sulėtėja. Tokia savybė labai tinka vaizdų apdorojimui dėl juose esančių triukšmų.



Pav 13. dviejų dimensijų eilutėmis adresuojamas masyvas su parametrais:  
Tipas 0, pradžia 3, žingsnis 1, eilutės kiekis 7, sekanti eilutė 93, kiekis 28 [11]

Numerio atpažinimo sistemų įgyvendinimo galimybės aparatinėje įrangoje aptartos

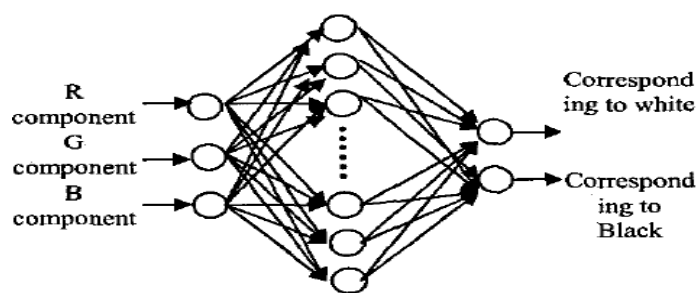


[11]. Įrangos funkcinis keitimas apribojamas dėl nedidelių kiekių projektavimo ir gamybos kaštų. Tipiniai skaliariniai procesoriai, tokie kaip Virtex PPC405 ar Altera nios, nepakankami didelį pralaidumą bei paralelinį programavimą, tiek kodo tiek duomenų lygyje, naudojančioms programoms įgyvendinti integruotose sistemose. Šiems apribojimams apeiti siūlomas FPGA (angl. *Field Programmable Gate Array*) SoC (angl. *System on Chip*), gebantis, nuo pagrindinio branduolio atskirtais papildomais skaičiavimo vienetais, apdoroti žinomo atminties adresavimo pobūdžio srautinius duomenis. Programavimo metu reikia nurodyti duomenų masyvo formą, vietą bei norimas atlikti iš anksto numatytas operacijas sDFG (angl. *Streaming Data Flow Graph*) kalba. Taip duomenis apdorojantis pagalbinis procesorius(-iai) gali iš anksto gauti reikalingus duomenis iš atminties, be to mažiau apkraunama sisteminė magistralė. Srautinių duomenų registrai aprašymui reikalinga (Pav 13): duomenų tipas (0 – baitas, 1 – 16 bitų žodis), adreso pradžia, žingsnis baitais, eilutės elementų kiekis, atstumas iki sekančios eilutės bei bendras elementų kiekis. Toks aparatinis srautinių duomenų apdorojimas leidžia ženkliai paspartinti aritmetiškai intensyvių duomenų apdorojimą – kadro filtravimą. sDFG sudaro baziniai komponentai aritmetikai, loginėms operacijoms ir kryptiniai kraštai, nurodantys operacijos priklausomybę prieš tai buvusios operacijos išvedimo duomenims. Kiekvienas sDFG komponentas aprašo srautinės operacijos tipą, gaunamus duomenis iš kitų komponentų bei išvedimo duomenų dydį ir neigiamo ženklo buvimą. Įvedimo duomenys vietoje konkrečių registų nurodomi nuoroda į prieš tai einančius komponentus, taip eliminuojant bereikalingą registų eikvojimą. Sistema sėkmingai išbandyta ieškant numerio rėmo su atidarymo bei uždarymo morfologinėmis operacijomis, ieškant horizontalių brūkšnių bei ieškant vertikalinių brūkšnių. Bandymų metu numerio atpažinimo sistema pagreitinama iki 2,6 karto.

[12] pasiūlytas MGD - maksimalaus gradiento skirtumo metodas, numerio rėmui rasti. Perėjimo vietoje tarp fono ir teksto, teksto plotis dažniausiai būna 2-3 taškai, todėl pagal šiuos duomenis ieškoma gradiento minimumo bei maksimumo vietiniame bloke. Jei gradiento pasikeitimas viršija nustatytą ribinę vertę – rasta numerio vieta.

Darbe [13] aptartas atraminių vektorių klasifikatorių (angl. *SVM - Support Vector Machine*) naudojimas numerio atpažinimui. Numerio rėmai aptinkami ieškant krašto. Simbolių atskyrimas, kaip ir daugumoje kitų darbų, vykdomas pagal projekcijų ribines vertes, tačiau šiuo atveju ribinė vertė skaičiuojama iš kadro.

Dirbtinių neuronų tinklų naudojimas numerių atpažinimui aprašytas [14]. Spalvų ar spalvų histogramų šuolių ieškančios sistemos netinkamos dėl didelės aplinkos įtakos. Siūloma pirmiausiai perversti RGB kadra į bitinę formą naudojantis dirbtinių neuronų tinklų. Pagal



Pav 14. dirbtinis neuronų tinklas, verčianti kadrą bitine forma [14]

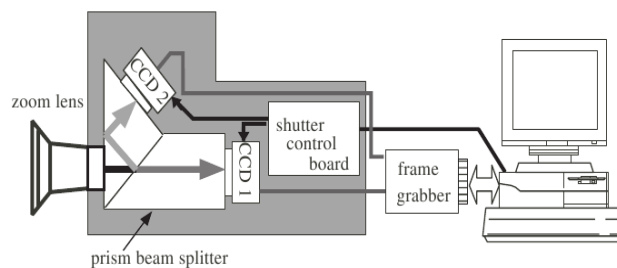
numerio kontrasto ypatumus, numerio vieta ieškoma sudarant intensyvumo žemėlapius, kurie vėliau apdorojami dirbtinio neuronų tinklo. Simboliai skaidomi į dėšningumo vektorius (angl. *pattern vector*) iš anksto numatytu atstumu pagal plotį be aukštį. Simbolių atpažinimui naudojamas Hopfield tinklas.

Miesto stebėjimo sistemos galimybių studija pateikta [15]. Iš esmės, nagrinėjama galimybė stebėti galinius transporto numerius srauto analizei. Siūlomas ne pastovus vaizdo sekimas, o sistemos aktyvavimas suveikus davikliui, kuris gali būti tiek aparatinis tiek programinis. Programinis transporto buvimas nustatomas pagal fono vaizdo kadro ir gauto vaizdo kadro skirtumus. Numerio vietos radimui tinka keletas vertintinų metodikų, kaip krašto radimas ar spalvos radimas. Simbolių atskyrimui siūloma sekti pilkumo skalės lygio pasikeitimus, o kirpimo vietą nustatyti pagal panašių pilkumo verčių ruožą. Atpažinimo procesas, kaip ir daugumoje kitų darbų, susiejamas su iš anksto žinomais duomenimis – numerio formatu. Numerio atpažinimui tikrinamas ne pats simbolis, o simbolių histogramų vertės.

Dar viena apmokamo kelio, nestojamo apmokėjimo posto (angl. *Non-stop*), klientų atpažinimo sistemos alternatyva RFID (angl. *Radio Frequency Identification Device*) pateikta [16]. Suveikus aparatiniams davikliui, apdoroti siunčiamas 768x256 dydžio 24bitų spalvų gylis kadras. Kadras PCI plokštės išsaugomas į operatyvinę atmintį, kur vėliau apdorojamas programinės įrangos. Prieš ieškant numerio vietos, kadras verčiamas į skirtuminį žemėlapi, kuris savo ruožtu verčiamas į bitinę formą. Pašalinamas dydžio bei formos neatitinkantis triukšmas. Likusiam kadrai skaičiuojamos histogramos, pagal kurias atrenkama numerio vieta. Simboliai atskiriami įprastu projekcijos būdu. Atpažinimas vykdomas, simbolių pervertus bitinę formą, randant Euklidinį atstumą tarp simbolio savybių (projekcija, perimetras) ir duomenų bazės.

Dar vienas algoritmas naudojantis krašto aptikimą aprašytas [17]. Horizontalių linijų kadre būna ženkliai daugiau nei vertikalių, todėl numerio vieta ieškoma pagal vertikalius kraštus. Visi kraštai, kurie mažesni nei 8 nenutrūkstamų taškų linija - atmetami. Likę

rezultatai išfiltruojami pagal plotį, aukštį bei atitikimą santykiui 2:1 (sistema taikoma Korėjai). Dėl to, kad aspekto koeficientas kadre nebūtinai visiškai atitinka nurodytą santykį, iš tikro tikrinama santykio priklausomybė režiuui nuo 1.4:1 iki 3.3:1. Be to, galima numerio vieta atmetama, jei vieno krašto koordinatės skiriasi daugiau nei pusė krašto ilgio nuo kito krašto koordinatės arba jei vertikalių kraštų ilgių santykis neatitinka ribos [0,8; 1,2] - apribojamas sistemos gebamas apdoroti pavertimo kampas bei perspektyva ar šoninis pasukimas. Kiekvienas likęs potencialus numerio rėmas papildomai tikrinamas kiek procentų šio ploto sudaro balti taškai. Jei baltų taškų mažiau nei 10% arba daugiau nei 40% - numeris atmetamas. Radus numerio rėmą, kadro dydis keičiamas iki 200x100 taškų. Prieš skaidant simboliais verčiama į bitinę formą pagal ribines pilkumo vertes. Ribinė vertė parenkama pagal fono spalvą. Gautas numeris filtruojamas pagal dydį ir formą. Simboliai atskiriami pagal histogramas bei normalizuojami iki 20x20 arba 8x16 taškų, pagal tai, kokia simbolio klasifikacija: raidė ar skaičius. Simbolių atpažinimas vykdomas lyginant šablonus, atsižvelgiant į nacionalinį numerių formatą.



Pav 15. Dviejų kamerų sistemos struktūra [18]

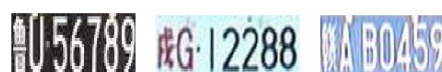
No.		angle(deg.)
1.	0 1 2 3 4 5 6 7 8 9	-50
2.	0 1 2 3 4 5 6 7 8 9	-40
3.	0 1 2 3 4 5 6 7 8 9	-30
4.	0 1 2 3 4 5 6 7 8 9	-20
5.	0 1 2 3 4 5 6 7 8 9	-10
6.	0 1 2 3 4 5 6 7 8 9	0
7.	0 1 2 3 4 5 6 7 8 9	+10
8.	0 1 2 3 4 5 6 7 8 9	+20
9.	0 1 2 3 4 5 6 7 8 9	+30
10.	0 1 2 3 4 5 6 7 8 9	+40
11.	0 1 2 3 4 5 6 7 8 9	+50

Pav 16. Simbolių šablonai, pasuktiems simboliams atpažinti [18]

Dviejų skirtingo išlaikymo kamerų naudojimas aprašytas [18]. Dėl būtinybės gebėti fotografuoti 100km/h ir greičiau važiuojančias transporto priemones, būtina fotografuojant išlaikymą laikyti mažiau 1/600 sekundės. Pasinaudojant prizme, sujungus dviejų skirtingo išlaikymų kameras (Pav 15), išgaunamas kadras, kurio šviesumo verčių ruožas platesnis nei

originalus vaizdo. Mechanizmas sukonstruotas taip, kad vaizdas perėjęs prizmę patektų tiksliai į tą patį matricos tašką (CCD, angl. *Charge-Coupled Device*) abiejose kamerose. Vertikalių ar horizontalių kraštų ieškantys algoritmai prastai veikia šios sistemos sąlygomis, dėl atsirandančios vienos ar kelių ašių perspektyvos bei pasukimo. Kaip ir daugelyje kitų darbų, pradiniu kadro apdorojimu vaizdas verčiamas į bitinę formą, tačiau čia naudojama dinaminė ribinė vertė slenkančiam langui vietoje bendros ribinės vertės. Langas parinktas 10x10 taškų, nes naudojant didesnius langus, numerio simboliai neperverčiami tvarkingai, o mažesniems langams atsiranda daug triukšmo fragmentų. Atpažįstant numerio rėmo vietą, kaip ir kituose šaltiniuose, neįvertinamas numerio rėmo pasukimas ar pavertimas, tačiau šis vaizdo netikslumas įvertinamas skaidant numerį į simbolius, nors pats numerio pasukimas / pavertimas nėra atstatomas – lyginama su duomenų bazėje esančiu *pasukto* simbolio šablonu (Pav 16). Sistemos tikslumas 97%, esant 40 laipsnių kampui.

Kiek kitoks kadro apdorojimas pateikiamas [19]: numerio rėmui rasti pirmiausia randama spalvų dažnumo lentelė, iš kurios dažniausiai pasitaikančios spalvos, „geltonumas“ bei teksto galimybė - vertintinos numerio vietos. Numerio vieta randama tikrinant visas geltonumo vietas, sudarančias tinkamo dydžio stačiakampį. Simboliai atskiriami nustačius ribinę vertę pagal vidutinę taškų vertę, didžiausio gradiento 100 taškų plote. Gautam bitinės formos kadrui vykdoma sujungtų komponentų analizė. Skaidymo simboliais fazėje atmetama ~24,6% numerių. Simbolių atpažinimui randamos simbolių savybės: vertikali projekcija bei sujungtų komponentų kiekis ir horizontali projekcija bei sujungtų komponentų kiekis. Šios savybės naudojamos kaip DCNN dirbtinio neuronų tinklų įvedimo duomenys. Kiekvienai savybei naudojamas atskiras dirbtinis neuronų tinklas. Simbolis atpažįstamas, jei dirbtinių neuronų tinklo išvedimas viršija 0,85 lygį, o kitų dirbtinių neuronų tinklų išvedimas mažiau 0,25 lygio. Po simbolių atpažinimo tikrinama numerių sintaksė.



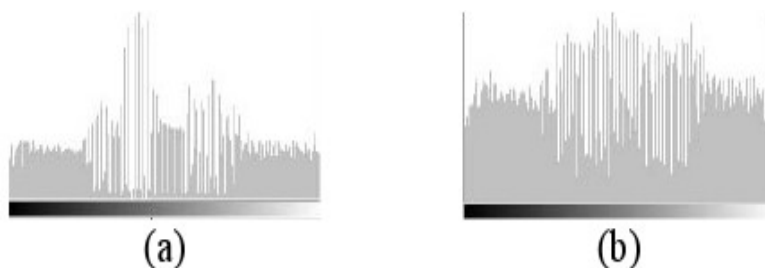
(a) Primitive car license images

车牌字符: 车牌字符: 车牌字符:  
鲁U56789 成G12288 赣A0459

Pav 17. Numerio plokštelė su atpažintais simboliais [20]

Kadro, OSTU metodu verčiamo į bitinę formą, ribinių verčių dinaminio parinkimas patobulintas [20]. Tipiniams, šabloną tikrinantiems algoritmams būtina aukšta kadro kokybė. Statistiniai simbolių atpažinimo modeliai atpažįsta simbolio charakteringas savybes, tačiau reikalauja daug skaičiavimo resursų ir gali lengvai apsirikti. Dirbtinių neuronų tinklai,

naudojantys simbolį kaip įvedimo duomenis, reikalauja ilgo skaičiavimo laiko ir daugybės pavyzdžių. Patobulintas šablono lyginimo metodas sujungia tradicinį šablonų lyginimą ir statistinį modelį: įvertinamas simbolio taško staigus pasikeitimas. Pirmiausia fonas prilyginamas 0, o visi simbolio bei triukšmo taškai didesni už 0. Tikrinama kiekviena simbolio eilutė ieškant pasikeitimo iš 0 į daugiau už 0, o tokiu atveju, pasikeitęs taškas tikrinamas ar priklauso simboliui. Jei taškas nepriklauso simboliui – prilyginamas fonui ir tikrinama toliau. Jei sekantis taškas yra daugiau už 0, tuomet taškas priklauso simboliui. Taškų priklausomybė simboliui ar triukšmui tikrinama tiek eilutėmis, tiek stulpeliais. Taškai, priklausantys simboliui sudedami į atskirą masyvą. Jei eilutės ir stulpelio masyvas sutampa, šie masyvai lyginami su duomenų baze (Pav 17). Šio metodo tikslumas patikrintas 200 nuotraukų: šablono lyginimu gautas 90% tikslumas, statistiniu modeliu – 85%, o bendru šablono lyginimui bei statistiniu modeli – 98%. Šabloninio tikrinimo modelis lengvai klysta padidėjus triukšmui, tačiau turi gerą atpažinimo patikimumą. Statistinis modelis savo ruožtu atsparus įvairiems triukšmams, tačiau žemas atpažinimo patikimumas.



Pav 18. a – klasikinis histogramos išlyginimas,  
b – histogramos išlyginimas su linijine interpoliacija [21]

Numerio rėmo aptikimas, kai kadras per šviesus - viršytas išlaikymas, aptartas [21]. Naudojant tradicinę išlyginimo funkciją prarandami duomenys, dėl naujų pilkumo verčių skyrimo – atsiranda nenaudojami histogramos tarpai. Šiai problemai spręsti pasiūlytas histogramos išlyginimas su linijine interpoliacija (Pav 18). Numerio vietai rasti ieškoma aukšto taškų pasikeitimo vietos. Dėl daugybės kraštų numerio vietoje, numerio vieta ieškoma pagal vertikalias ir horizontalias linijas, tam naudojant 3x3 kaukės Sobel operatorius. Išfiltruotas kadras verčiamas bitine forma, o gautas kadras dalinamas į 32x64 dydžio blokus. Kiekvieno šių blokų juodų taškų skaičius saugojamas matricoje. Jei bloko juodų taškų skaičius viršija nurodytą ribinę vertę, kiekvienas šio bloko taškas yra numerio viršutinio kairiojo kampo kandidatas. Be to, šie taškai turi nukrypimą, todėl turi būti perstumta skaičiuojant kiek juodų taškų yra tarp 16x32 taškų, kai tikrinamas taškas yra 32x64 lango centras. Šioje vietoje gautos numerių vietos patikrinamos dar kartą: išlyginama histograma,

kadras verčiamas bitine forma pagal ribinę vertę 128, 3x3 vykdomas morfologinis uždarytas, histogramų slėnių bei jų dydžio skaičiavimas. Histogramos slėnio plotis turi būti tarp kadro pločio padalinto iš 10 ir kadro pločio padalinto iš 5. Šio metodo efektyvumas 96%, tačiau tinkamas tik per daug apšviestiems kadrui, be to naudojama nekintanti ribinė vertė morfologinėms operacijoms, o pradinio kadro kokybė ženkliai viršija kituose darbuose naudojamas: 1024x768 lyginant su 320x240.

Labai svarbūs pradinio kadro apdorojimo algoritmai aptarti [22]: kontrasto ar apšviestumo išlyginimas, „Top-hat“, „bottom-hat“ morfologinės operacijos.

„Top-hat“ - morfologinė operacija kai taško vertė pakeičiama pagal kaimyninių taškų vertes, o taip gerai šalinami smulkūs triukšmai. Šios operacijos atitikmuo – morfologinis skirtumas tarp originalaus ir morfologiškai atverto kadru.

Tuo tarpu „bottom-hat“ - identiška „top-hat“, tik šiuo atveju atitinka morfologinę skirtumo operaciją tarp originalaus ir morfologiškai užverto kadru.

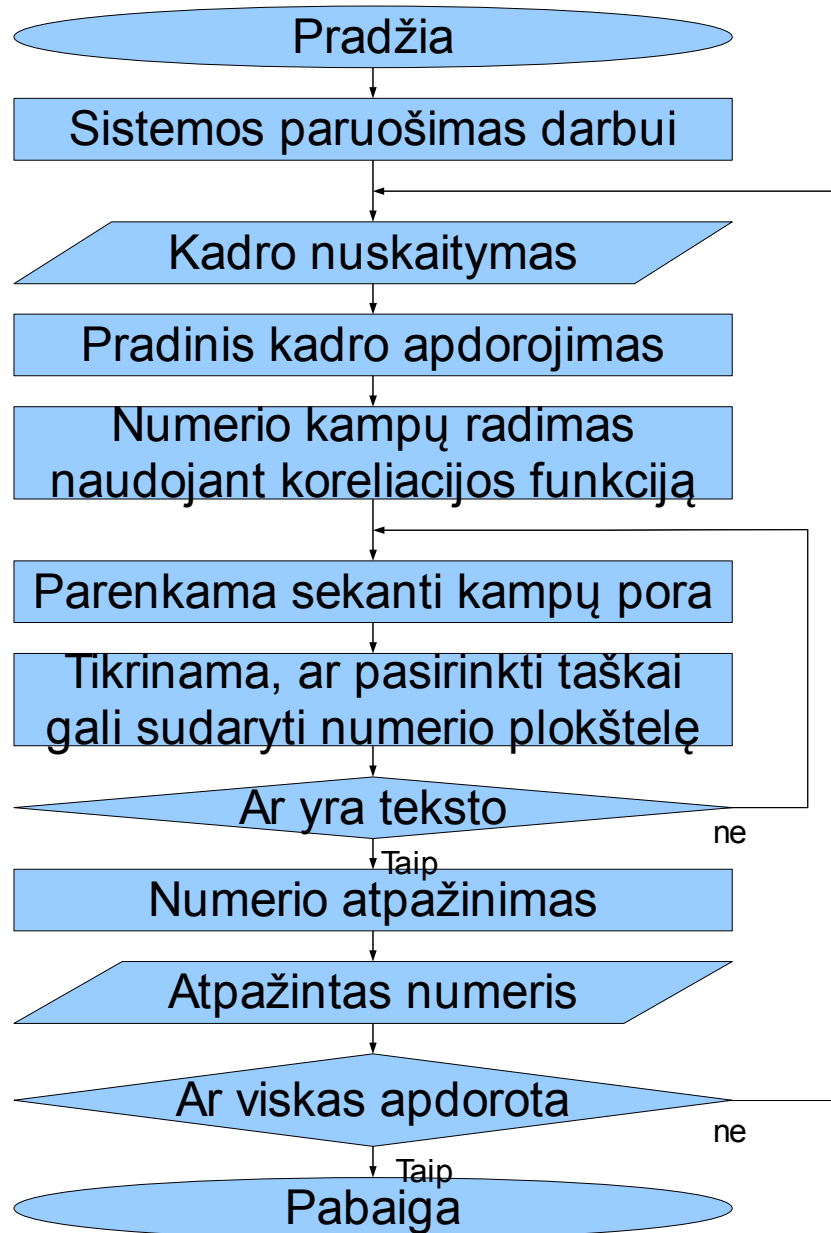
*Dekoreliacija* – skirtingų ruožų išplėtimas, naudojantis trimis RGB ruožais gaunant pagerintas HSI koduotės SI vertes nepakeičiant H komponentės. Šis metodas tinkamas ir pilkumo skalei pagerinti. Be šių metodų galimi ir kadro aštrinimas bei prisitaikantis histogramos lyginimas, tačiau šių metodų, kaip ir dekoreliacijos metodo, rezultatai gauti palyginus prasti, nors šie metodai nereikalauja tiek skaičiavimo laiko kiek „top-hat“, „bottom-hat“ ar iškart abejos morfologinės operacijos.

Tarptautinių numerių atpažinimas aptartas [23]. Pagal autorių, nėra net bandymo standartizuoti numerio formatą ar stilių, ypač dėl mašininio apdorojimo, tačiau griežtai vienodo tipo ir stiliaus numerių formatas labai apribotų vietinės valdžios pelno galimybes iš asmeninio formatavimo numerių. Dauguma numerio plokštelių gaminama iš specialaus, šviesą atspindinčio paviršiaus, kuris atšviestas tam tikru kampu tuo pačiu keliu atspindi visą šviesą. Būtent dėl šių savybių galima išgauti iki 16 kartų daugiau energijos, kuri sėkmingai išnaudojama kamerose su integruotu infra-raudonųjų spindulių apšvietimu. Be to, vis dar ne visose šalyse naudojamos atspindinčios plokštelės. Dažniausiai numerio simboliai žymimi tamsiai, o fonas šviesus, nors pasitaiko ir atvirkščiai. Kai kurių numerių apšvietimas gali labai suprastinti kadro kokybę dėl naudojamų dažų konkretaus dažnio bangų absorbuavimo skirtumų.

[24] siūlomas numerio rėmo radimas naudojantis ne pilkumo skale apskritai, o trimis RGB gradiento skalėmis - kiekvienai spalvai atskirai. Numerio vieta ieškoma Sobel operatoriumi atrenkant vertikalinius kraštus. Kraštui priklauso tie taškai, kurių kiekvieno spalvos kanalo gradientas viršija nustatytą ribinę vertę. Gradiento skirtumas

numerio regione išskirtinai stabilus bei aukštas, nei kituose regionuose.

## Siūlomas algoritmas



Pav 19. Struktūrinė schema

Visos numerio atpažinimo sistemos vienaip ar kitaip įvertina autotransporto priemonės buvimą konkrečioje zonoje: vienoje naudojamas aparatinis daviklis, kitose programinis. Kadangi ši sistema (Pav 19) projektuojama kiek įmanoma universali bet kokiomis sąlygomis, transporto priemonės atpažinimą derėtų tikrinti programiniu būdu. Tai galima realizuoti iš

kameros gaunamą vaizdo kadru seką kaupiant cikliniame buferyje, iš kurio vėliau nustatoma ar kadre gali būti transporto priemonė. Vis dėl to, integruotų sistemų ar realaus laiko reikalaujančių uždavinių atžvilgiu, programinis kadro tikrinimas tikrai ne geriausia priemonė tinkamų kadru filtravimui tiek energetine prasme, tiek turimų skaičiavimo resursų prasme. Pats greičiausias transporto priemonės arba apskritai bet kokio objekto atsiradimas kadre gali būti patikrintas atimant vieną kadrą iš kito arba fono kadro, taip išryškinant nesutampančias šių kadru vietas. Tokios paprastos operacijos įgyvendinimui pilnai pakanka ir integruotų sistemų ar realaus laiko reakcijos reikalaujančių sistemų resurso. Gautas skirtuminis vaizdas bus kaip žemėlapis, kuriose vietose pasikeitė duomenys – atsirado arba išnyko objektas. Būtent šioje vietoje tenka spęsti dilemą, kaip patikrinti ar turimas skirtumas pakankamas ir kokią ribinę vertę tam pasirinkti. Vienas iš galimų sprendimo variantų - lyginti pasikeitusių taškų kiekį, tačiau tam reikia žinoti kameros parametrus kaip: aprėpiamą plotą bei galimą perspektyvą. Kiek kitokia galimybė – gautą skirtuminį vaizdą apdoroti morfologiškai, taip pašalinant smulkius pasikeitimus ir apjungiant stambius blokus. Dėl to, kad sistemos darbo objektas – autotransporto priemonė, šioje vietoje galima įvertinti pasikeitusio objekto formą: atpažinti ar fono vaizdą užstojantis objektas yra autotransporto priemonė ar pašalinis triukšmas, kaip lapai ar vėjo judinama kamera. Tačiau toks apdorojimo būdas jau pačioje algoritmo pradžioje ženkliai apribotų galimą sistemos darbą, be to tokiems skaičiavimams būtini atitinkami skaičiavimo pajėgumai, o tai neigiamai atsiliepia integruotose sistemos ar realaus laiko reakcijos reikalaujančiose sistemos. Dauguma numerio atpažinimo sistemų sudaro kamera bei numerio atpažinimą vykdomas asmeninis ar servisinis kompiuteris, su atitinkama programine įranga, todėl ši sistema taptų lygiai taip pat priklausoma nuo galingo skaičiavimo resurso – prarandama bet kokia galimybė algoritmą realizuoti integruotose sistemos. Be to, mano asmenine nuomone, pasikeitusių taškų skaičiaus tikrinimas yra pilnai pakankamas. Kitas šios problemos aspektas – kokioje spalvų koduotėje reikalingas judesio tikrinimas – spalvotas ar pilkumo skalės. Viena vertus, spalvoto vaizdo pasikeitimų tikrinimas garantuotų bet kokio smulkaus pasikeitimo užfiksavimą, tačiau šiuo atveju ieškomas pakankamai didelis objektas, todėl ši koduotė kaip ir per daug tiksli. Antra vertus, neaišku kokiomis apšvietimo sąlygomis turi dirbti sistema, todėl visai įmanoma, kad gauto vaizdo histograma pasiskirsčiusi arčiau vertikalios ašies, dėl galingo į kadrą patekusio šviesos šaltinio ar natūraliai tamsių aplinkos sąlygų. Būtent dėl šios priežasties įmanoma, kad pilkumo skalės kadre atsiradęs pašalinis objektas gali likti nepastebėtas, dėl objekto intensyvumo panašumo su uždengtu fono plotu, todėl kadrą vertėtų tikrinti su nepakitusių spalvų kodavimu. Vienas programinio objekto atsiradimo kadre daviklio pliusų – vėlesniuose



skaičiavimuose atsirandanti galimybė apriboti kadre tikrinamų taškų plotą ir tikrinti tik į pasikeitimo plotą patenkančius taškus, be to, žinant kameros kadru kiekį per sekundę, galima nuspėti objekto judėjimo preliminarią greitį, tačiau šiuo atveju tektų buferyje kaupti pakankamą kadru kiekį, o tai abejotina, naudojant integruotas sistemas ar realaus laiko reakcijos reikalaujančias sistemas. Tarkime, kad kamera nutaikyta į šlagbaumu ar valdomais vartais apsaugotą pravažiavimą. Tokiu atveju autotransporto priemonės dydis gaunamuose kadruose priklausys tik nuo kameros didinimo, o pagal tai galima nustatyti ir ribinį pasikeitusių taškų kiekį. Dėl to, kad apdorojamas ne vienas kadras, o kadru seka, gaunamas jau iš anksto apribotas galimos numerio vietos paieškos regionas. Integruotose sistemos ar realaus laiko reakcijos reikalaujančiose sistemos, šioje vietoje tenka vertinti kuri algoritmo dalis reikalauja daugiau skaičiavimo pajėgumų: galimų numerio taškų tikrinimas ar judesio detekcija. Žinant didesnės resursų dalies reikalaujantį algoritmą, jau galima vertinti, kurią iš sistemų ir ar aplamai pašalinti, skaičiavimo resursų mažinimui.



Pav 20. RGB (kairėje) ir pilkumo skalės kadrai

Jau turint informacijos, kad kadre galimai yra autotransporto priemonė, kadru būtina paruošti tolesniems skaičiavimams. Kaip jau minėta anksčiau, sistemos darbo sąlygos nėra žinomos, o pati sistema projektuojama kiek įmanoma universaliam darbui, todėl pradiniam kadro filtravimui galima iškirpti nesikeičiančias kadro vietas – foną. Dėl to, kad spalvinis kadro apdorojimas pernelyg brangus, tiek laboratorinių sąlygų skaičiavimams, tiek integruotoms sistemos ar realaus laiko reakcijos reikalaujančios sistemos, skaičiavimo resursų prasme, kadru būtina konvertuoti į programavimo prasme patogesnę pilkumo skalės kodavimą (Pav 20), o tai nesudėtingai atliekame pagal formulę:

$$X_g = (R + G + B) / 3 \quad (2)$$

Čia R, G, B atitinka RGB spalvų kodavimo atitinkamas komponentes: raudoną, žalią bei mėlyną spalvą. Šioje vietoje galima būtų ginčytis dėl konvertavimo formulės koeficientų

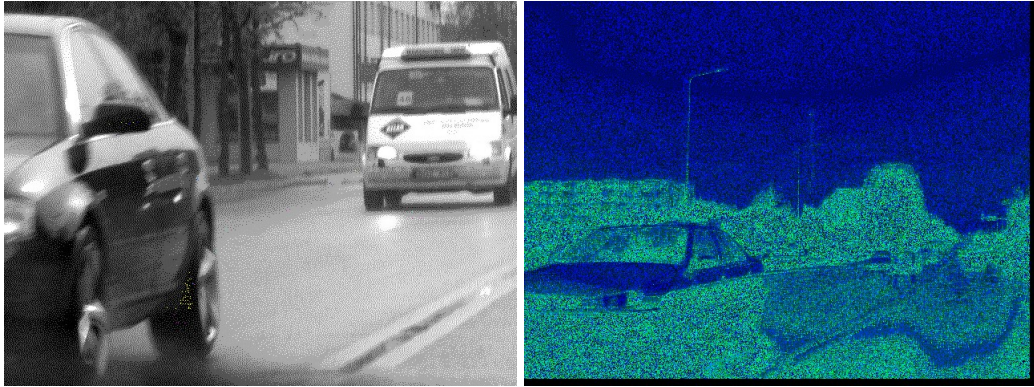
atskiroms dedamosios, tačiau tai daugiau konkrečiau techninio įgyvendinimo klausimai ir susiję tik su konkrečia naudojama aparatūra: kameros šviesai jautrios matricos elemento subalansavimo bei esamo ar ne iškraipymų kompensacijos. Be to, būtini skaičiavimai plaukiančio kabelio operacijoms didesni nei sveikųjų skaičių operacijoms, o integruotose sistemose plaukiantis kabelis gali būti visiškai nerealizuotas aparatinėje pusėje, todėl skaičiavimams naudojami sveikieji skaičiai, o dėl to jau šioje stadijoje prarandama dalis informacijos.

Iš objekto aptikimo kadre algoritmo, žinomi kadro foną užstojuojantys regionai, todėl tolesniems skaičiavimams galima naudoti būtent šį regioną, taip sumažinant būtinus skaičiavimus ir galimai pašalinant per šviesius ar per tamsius objektus. Būtent dėl to, jau šioje vietoje galimas kadro verčių pasiskirstymo tankio (histogramos) poslinkis, dėl galimai pašalinto apšvietimo triukšmo ar pašalinio tamsaus regiono. Bet kuriuo atveju, pasikeitusi histograma, kaip rodo ir eksperimentai, skaičiavimams nenaudinga, todėl histogramą būtina išlyginti, taip išdidinant kontrastą kiek įmanoma daugiau. Kaip jau minėta aukščiau, šio žingsnio metu kai kurios histogramos vertės, dėl naudojamų sveikųjų skaičių, dingsta iš kadro, o šios problemos sprendimui galima panaudoti literatūroje pateiktus alternatyvius algoritmus: „Top-Hat“, „Bottom-Hat“, prisitaikantį histogramos išlyginimą ar histogramos išlyginimą su verčių interpoliacija. Vis dėl to, integruotų sistemų ar realaus laiko reakcijos reikalaujančių sistemų atžvilgiu, tokie skaičiavimai tampa nepagrįsti, dėl atsirandančių plaukiančio kabelio operacijų, o dėl to ir būtinybė emuliuoti plaukiančio kabelio operacijas operuojant sveikaisiais skaičiais, taip ištempiant operacijos apdorojimo laiką bent kelis kartus. Bet kuriuo atveju, pradiniam bandymams tinka ir klasikinis histogramos išlyginimas pagal formulę:

$$X' = (X - X_{min}) * 255 / X_{max} \quad (3)$$

kur  $X$  – pradinė taško vertė,  $X_{min}$  - minimali taškų vertė,  $X_{max}$  - maksimali taškų vertė, o  $X'$  - naujai gauta taško vertė

Pagal aukščiau paminėtus autorių darbus, galima teigti, kad populiariausi numerio rėmo radimo algoritmai: morfologinėmis operacijomis išfiltruoti didelio kontrasto vietas, krašto paieška ar spalvos gradiento pasikeitimo paieška. Daugelis šių metodų grindžiami bitinės formos vaizdu, be to labai priklausomi nuo apšvietimo bei iškraipymų sąlygų. Visi darbuose aprašyti algoritmai vienaip ar kitaip ieško numerio vietos charakteringų taškų ar savybių. Kadangi kadras bet kuriuo atveju atvaizduojamas kaip dvimatis duomenų masyvas,



Pav 21. Kadras (kairėje) ir atitinkamas koreliacijos

visas kadro eilutes sujungus į vieną eilutę, kadra galima būtų interpretuoti kaip konkretaus ilgio ir begalinio spektro signalą. Be to, elektronikoje sėkmingai pritaikyta koreliaciją, reikalingo signalo aukštam stiprinimo koeficientui triukšmo atžvilgiu gauti, todėl šioje situacijoje užtenka tik surasti reikalingą dažnį – numerio savybę. Kituose darbuose tikrinamos savybės: dideli kontrasto gradiento šuoliai, konkreti spalva, konkretus dydis, konkreti forma ir pan. Praktiškai nė vienas šių algoritmų pilnai netinka kaip atitikmuo dažniui signale, todėl numerio vietos paieškai nuspręsta naudoti vertikalius panašaus intensyvumo brūkšnius: t.y. žemą virpesių dažnį signalo prasme. Šiam tikslui, per visą kadra, iš kairės į dešinę ir iš viršaus į apačią, bet nebūtinai taip, skaičiuojami nenormuoti koreliacijos koeficientai tikrinamam taškui. Eksperimentų metu, taip gautas svorinis žemėlapis (Pav 21), neturintis apatinės ar viršutinės koeficientų ribos, ir priklausantis nuo to, kaip buvo vykdomas tikrinimas, bei kokio ilgio bazinis „signalas“ naudojamas. Praktiškai pastebėta, kad naudojant didesnius nei 10 taškų etaloninius kadrus koreliacijai, numerio rėmai nebe aptinkami, o naudojant mažesnius - koreliacijos koeficientų žemėlapis bendras vidurkis didėja, o tai reiškia, kad ir bendras atitikimas užduotam „signalui“ auga. Toks sistemos darbas netoleruotinas, todėl etalonu geriau parinkti 10 taškų kadra. Eksperimentų metu pastebėta, kad naudojama koreliacijos funkcija, iš formulės 4, ne visais atvejais tinka dėl vieno ar net kelių komponentų galimo standartinio nuokrypio nebuvimo: kai visos masyvo reikšmės vienodos.

$$r_{x,y} = \frac{\sum X_i Y_i - n \bar{X} \bar{Y}}{(n-1) S_x S_y} \quad (4)$$

Čia  $X_i$  - kadro kadro i-toji ląstelė,  $Y_i$  - etaloninio kadro i-toji ląstelė,  $S_x$  - kadro standartinis nuokrypis,  $S_y$  - etaloninio kadro standartinis nuokrypis,  $n$  – tikrinamų taškų skaičius,  $\bar{X}$  - kadro reikšmių vidurkis,  $\bar{Y}$  - etaloninio kadro reikšmių vidurkis. Be to, vienam iš taškų atitikus 0 ar visam tikrinam rėžiui esant 0, koeficientas nedidėja. Dėl to, kad

apdorojamas ne bitinės formos kadras, o pilkumo pustonių skalės kadras, koreliacijos koeficientas turėtų didėti sutampant verčių pasikeitimo dėsningumui, todėl koreliacijos funkcija modifikuota sekančiai:

$$r_{X,Y} = \sum 255 - \|X_i - X_{i-1} - Y_i\| \quad (5)$$

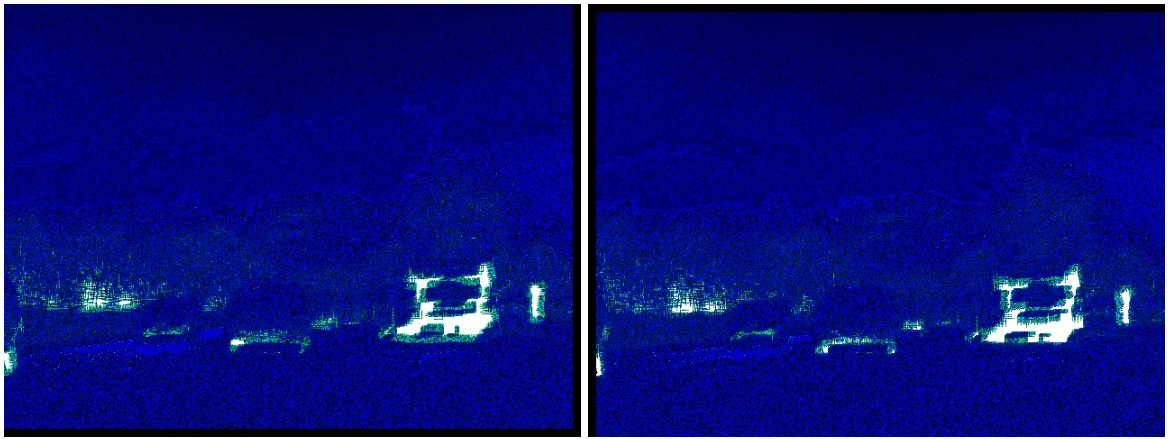


Pav 22. Pagal koreliacijos koeficientus tikrinami taškai:  
 raudoni taškai – netinkamos vietos  
 geltoni taškai – galimos vietos  
 žali taškai – rastos vietos

Kintamieji tokie pat, kaip ir 4 formulėje. Eksperimentų metu formulės pakeitimas pasiteisino, išryškinant visas, bet kokio intensyvumo, vertikalias linijas. Be to, ženkliai supaprastintas formulės skaičiavimas, o tai ypatingai naudinga integruotose sistemos ar realaus laiko reakcijos reikalaujančiose sistemose. Šioje vietoje jau turimas vertikalių linijų svorinis žemėlapis, tačiau iš to nėra naudos, nes vien remiantis šiuo žemėlapiu tikrinant visas galimas linijas būtų bereikalingai eikvojami resursai. Todėl nuspręsta apskaičiuoti dar vieną svorinį koreliacijos koeficientų žemėlapi, tik šiuo atveju horizontalioms linijoms ir naudojant 10x1 taškų etaloninį kadrą. Kaip ir vertikalios linijoms, pateikta formulė 4 netiko, todėl ir šiems skaičiavimams naudojama alternatyvi modifikuota formulė 5. Vien turint tiek vertikalių, tiek horizontalių linijų koreliacijos koeficientų svorinius žemėlapius, būtini skaičiavimo resursai, visiems galimiems taškams patikrinti, per dideli. Kiekvienas numeris turi tiek vertikalius tiek horizontalius kraštus, o dėl to ir atitinkamas linijas numerio krašte, todėl tarpusavyje sudauginti šie koreliacijos koeficientų svoriniai žemėlapiai, prieš tai jų vertes normalizavus – perstūmus koreliacijos koeficientų reikšmes taip, kad nebūtų neigiamų reikšmių, atitiks bendrą abiejų žemėlapių vaizdą. Dėl to bendras šių koreliacijos koeficientų svorinių žemėlapių rezultatas atitiks labiausiai tikėtinus numerio rėmo viršutinio kairiojo kampo taškus

(Pav 23), ką patvirtina ir eksperimentiniai bandymai. Praktiniais bandymais pastebėta, kad koreliacijos koeficientai neįprastai dideli kadro žemo intensyvumo vietose (Pav 22). Galima numanyti, kad tokios vertės gaunamos dėl modifikuotos koreliacijos funkcijos, tačiau galimas ir alternatyvus variantas: žemo intensyvumo regionuose, visos reikšmės yra labai panašios vienos į kitas, todėl dėl panašumų ieškojimo automatiškai atsiranda vietų išryškėjimas. Vienas iš galimų sprendimo variantų šiai problemai – apriboti apdorojamo vaizdo intensyvumo vertes, tačiau taip mažesnio intensyvumo srityse esančios numerio plokštelės būtų ignoruojamos ir jau šioje vietoje nebe aptinkamos. Analogiškai ir didelio intensyvumo vietose, tačiau čia, dėl didesnių gradiento pokyčių tarp gretimų verčių, išryškėja tik labiau tinkami taškai. Įvertinus turimo algoritmo gaunamus potencialius taškus, nuspręsta koreliacijos algoritmo pataisas atidėti vėlesniam sistemos tobulinimui. Be to, vien šių koreliacijos koeficientų žemėlapių sudarymui, sistema kiekvienam žemėlapiui apskaičiuoti užtrunka apie 2-10 sekundžių, naudojant 4x 64 bitų 3,6GHz 800MHz FSB Intel x86 platformos tarnybinę stotį bei 320x240 raiškos nuotraukas su 24 bitų spalvų gyliu (pilnas RGB). Tokios skaičiavimo sąlygos gali tiktai nebent laboratorinio skaičiavimo sąlygomis, tačiau tikrai netinkamos integruotų sistemų ar realaus laiko reakcijos reikalaujančių sistemų sąlygomis. Be to, skaičiavimai iš populiaraus, prievarta brukamo mokamo MATLAB paketo, dėl labai ilgų skaičiavimų vienam koreliacijos koeficientų žemėlapiui apskaičiuoti (~1-2min), perkelti į Qt4 bibliotekų pagrindu parašytą programą, o skaičiavimai realizuoti C++ klasėse atskira, architektūros specifine, procesoriaus atšaka, be to programinis kodas transliatoriaus optimizuojamas aukščiausiu įmanomu laipsniu. Būtina pabrėžti, kad šiuo atveju C++ realizacijos koreliacijos skaičiavimo greitis nėra labai aktualus dėl, kaip aukščiau paminėta, rinkoje atsirandančių srautinių duomenų apdorojimo platformų, leidžiančių aparatinėje pusėje įgyvendinti konkrečias sudėtingas aritmetines operacijas. Dėl to, kad daugumos praktiškai taikomų sistemų atitinka schemą: daviklis – kamera – periferija – kompiuteris - programinė įranga, koreliacijos koeficientų žemėlapių sudarymas bei visas pradinis apdorojimas galimas periferinėje PCI plokštėje su viena iš tokių FPGA „sistemos lustuose“ (angl. *SoC*) sistema, taip didžiąją dalį mažai keičiamo algoritmo įgyvendinant pačioje kameroje, jei nenaudojama PCI plokštė vaizdo fiksavimui kompiuterio operatyvinėje atmintyje.

Turint vieno kampo koreliacijos koeficientų svorinius žemėlapius (Pav 23) jau galimas šių numerio kandidatų tikrinimas. Iš koreliacijos koeficientų svorinių žemėlapių sandaugos, galima daryti prielaidą, kad visi tikrinami kampai bus viršutiniai kairieji, todėl reikalingas tik numerio rėmo egzistavimo žemyn ir į dešinę patikrinimas. Kaip jau anksčiau minėta, bet koks



Pav 23. Koreliacijos koeficientų žemėlapiai

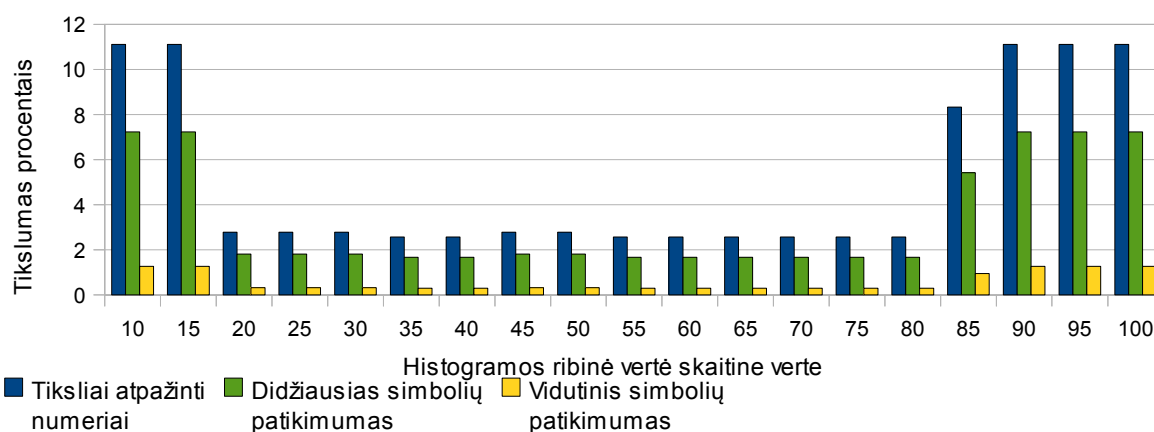
nekintančios ribinės vertės parinkimas – netinkamas sprendimas, todėl tikrinimui apsiribojame potencialų taškų skaičių iš koreliacijos koeficientų svorinio žemėlapiu pagal šio žemėlapiu vidutines vertes: jei koreliacijos koeficientas viršija vidutinę koreliacijos koeficientų svorinio žemėlapiu vertę, taškas tikrinamas. Šiuo būdu vietoje  $320 \times 240 = 76800$  taškų eksperimentų metu tenka tikrinti tik apie 1500 – 3000 taškų. Praktiškai išbandžius šitoks koeficientų apribojimas nelabai tinka, todėl pasirinktas kiek kitoks tinkamų koeficientų apribojimas. Dėl to, kad numerio rėmo vietose koreliacijos koeficientas turi padidėti šuoliu, aplinkinių koreliacijos koeficientų atžvilgiu, pirmiausiai surandamas koreliacijos koeficientų svorinio žemėlapiu gretimų verčių skirtumas pagal formulę:

$$X_i = X_i - X_{i-1} \quad (6)$$

Čia  $X_i$  - i-toji reikšmė. Taip koreliacijos koeficientų svoriniame žemėlapyje išlieka tik tos koreliacijos koeficientų vertės, kurių gradientas keičiasi, todėl belieka tik apriboti minimalų tinkamą gradiento pasikeitimo greitį. Kaip anksčiau minėta, bet koks statinis ribinės vertės nustatymas nėra tinkamas, todėl kaip ir anksčiau, ribinė vertė nustatoma pagal gauto skirtuminio žemėlapiu vidutinę vertę, taip eliminuojant pakankamai nedidelius foninius pasikeitimus. Šiuo būdu, eksperimentų metu vertų tikrinti kadro taškų skaičius sumažintas iki ~700-1000. Vis dėl to ir šio algoritmo galimų taškų kiekis viršija bet kokius lūkesčius, todėl skirtuminis koreliacijos koeficientų svorių žemėlapis apribotas ne pagal skirtuminių koreliacijos koeficientų svorių vidutinę vertę, o pagal 1/10 maksimalios vertės. Tai yra, galimi taškai apribojami pagal didžiausio gradiento šuolio vertę, todėl šiuo atveju vertų tikrinti galimų numerio vietos taškų, eksperimentų metu, gaunama pastebimai mažiau: tik ~150-400. Toks potencialių taškų kiekis jau gera pradžia taškų tikrinimui ir integruotose sistemos ar realaus laiko reakcijos reikalaujančiose sistemose.

Įvertinus galimus kadro scenarijus, logiška manyti, kad kadre numeris nebus prie pat bet kurio iš kraštų, todėl, jei taškas yra arčiau, nei atitinkamai dešimtoji kadro pločio ar aukščio dalis, yra atmetamas. Be to, logiška manyti, kad tikrinami numerio vietos taškų kandidatai visada bus kairieji viršutiniai numerio plokštelės taškai, dėl praktinės šių numerio vietos taškų radimo specifikos: koreliacijos koeficientas skaičiuojamas pirmajam tikrinamam vertinamo kadro elementui – viršutiniam kairiajam taškui. Dėl to, belieka rasti į dešinę ir žemyn besitęsiančios juostos pabaigą, o tam puikiai tinka koreliacijos koeficientų sekos, į dešinę bei žemyn, vienodumo tikrinimas. Šiam tikslui naudotina sekos tikrinimo funkcija lygina užbrėžtos sekos gretimų vertų skirtumo reikšmę. Šiuo atveju tenka naudotis pusiau statine ribine verte, atitinkančia šio galimo numerio rėmo taško skirtuminio žemėlapiu reikšmę. Šitokiu algoritmu eksperimentams naudojama sistema sėkmingai susidoroja su *aukštos kokybės* kadrais, turinčiais švarią, nedeformuotą numerio plokštelę, tačiau dirba pagal atsitiktinį dėsnį pakeitus apšvietimo sąlygas, kadro ar numerio rėmo kokybę. Būtent dėl šios priežasties reikia atsisakyti numerio rėmo tikrinimo, ieškant numerio kraštinės pabaigos, kaip koreliacijos koeficientų žemėlapiu panašių reikšmių sekos pabaigos. Algoritmas patobulintas ieškoti kito tinkamo koreliacijos koeficiento nuo tikrinamo taško į dešinę bei žemyn atitinkamai 1x10 ir 10x1 dydžio langais. Ribinė vertė koreliacijos koeficiento radimui parenkama pagal tikrinamo ruožo vidutinę koreliacijos koeficientų vertę. Šitokiu algoritmo pakeitimu eliminuojama didžioji dalis kadro intensyvumo pasikeitimo vietose randamų potencialių numerio rėmo taškų: šešėliai, padangų žymė ir pan. Šiuo atveju eksperimentinės sistemos stabilumas pagerintas, tačiau vistiek priklausomas nuo nedeformuoto numerio plokštelės ar panašaus intensyvumo autotransporto priemonės bei numerio rėmo krašto. Šiai problemai spręsti pasinaudojama jau anksčiau pritaikytu algoritmu – tikrinamos ne konkrečios koreliacijos koeficientų vertės, o koreliacijos koeficientų gretimų vertų skirtumas, be to, ribinė vertė tinkamam gradientui rasti parenkama pagal tikrinamos juostos skirtuminių koreliacijos koeficientų svorių žemėlapiu vidutinę vertę. Po tokių pakeitimų, algoritmo stabilumas bei atsparumas iškraipymams pagerinamas, tačiau vis dar lieka daugybė pašalinių kandidatų, kuriuos būtina atmesti iki simbolių atpažinimo. Šiuo tikslu įvedamas papildomas apribojimas numerio rėmui: numerio rėmo pločio ir aukščio santykis negali viršyti 10 ir analogiškai atvirkščiai. Dauguma autorių numerio rėmo aspektą apriboja tiek koeficientu 3 ar 2, tačiau tokios siauros numerio rėmų aspekto ribos gali ženkliai suprastinti sistemos daugia-nacionališkumo galimybes, todėl pasirinktas koeficientas, esantis skaitine verte toliau nei bet koks įmanomas numerio rėmas, tačiau tuo pačiu ne per toli, klaidingų rezultatų eliminavimui. Šio apribojimo įvedimas, pagerino sistemos galimų numerio rėmo kandidatų sąrašą iki 5-30

potencialių taškų. Bandymu metu pastebėta, kad iškerpama perteklinė informacija numerio rėmui atpažinti, be to, didžioji dalis likusių galimų numerio rėmo taškų aptikti tam pačiam rėmui, tačiau 1 - 5 taškų atstumų nuo kitų, to paties numerio rėmo, taškų. Vis dėl to, sistemos patikimumas, net ir išfiltravus potencialius numerio rėmo taškus iki tinkamo galimų numerio kandidatų kiekio, nedžiugina – sėkmingai randama tik ~1% numerio rėmų. Be to, bandymai atliekami su įvairių apšvietimo, kokybės bei iškraipymų sąlygų nuotraukomis, įskaitant nuotraukas su keliais numerio rėmais bei nuotraukas, kuriose aplamai nėra jokio numerio. Vis dėl to, sistema, kaip ir anksčiau, sėkmingai susidoroja su aukštos kokybės bei gero apšvietimo nuotraukomis – blogiausiu atveju, tarp galimų numerio rėmo taškų paliekama 1 - 2 klaidingi taškai. Atsižvelgiant į tai, sistemą, po detalių išbandymų bei dar egzistuojančių programinės įrangos klaidų patikros, galima bandyti taikyti pilnai ar pusiau automatiniais pravažiavimo kontrolės ar statistikos, tyrimo tikslais, uždaviniams vykdyti. Vis dėl to, rezultatams netenkinant norimų rezultatų, galimų numerio rėmo taškų sistema keičiama iš esmės.

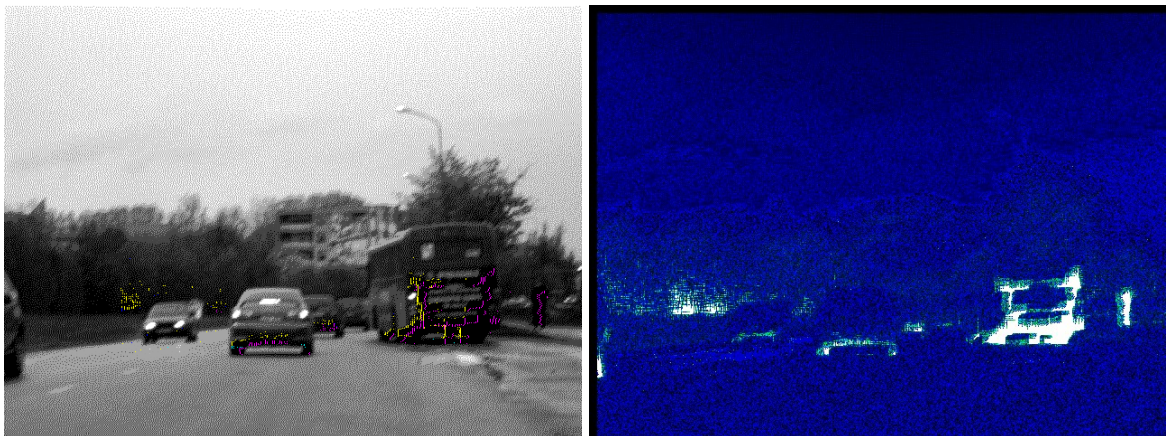


Pav 24. Histogramos ribinės vertės įtaka simbolių atskirymui

Įvedant rekursinį galimų numerio rėmo taškų porų tikrinimą, būtina aprašyti kriterijus, kuriais remiantis taškų pora būti priimama ar atmetama. Kaip žinoma, numerio plokštelės sudarytos iš vienos ar daugiau eilučių teksto, todėl būtų logiška galimą rėmą tikrinti pagal horizontalios histogramos verčių šuolius, ieškant tinkamo, jei tiksliau pakankamo, dydžio simbolių eilučių (Pav 24). Tokiu būdu, galimi rėmai, kurių viduje nėra atskiriamos teksto eilutės, yra netinkami ir todėl atmetami. Įvertinant jau panaudotus koreliacijos koeficientų gradiento žemėlapius, panašus algoritmas pritaikomas ir šiam tikslui. Koreliacijos koeficientai skaičiuojami ir įrašomi pirmo tikrinamo taško vietoje, nors šiuo atveju reikalingas paskutinis, atitinkamai pirmo stulpelio ar eilutės, taškas. Ši problema lengvai apeinama šiek tiek modifikavus koreliacijos koeficientus skaičiuojančią funkciją, tačiau vien koreliacijos



koeficientus skaičiuojančios funkcijos pakeitimo nepakanka. Ieškant apatinio dešiniojo numerio rėmo krašto, koreliacijos koeficientų svorinių žemėlapių sandaugai būtini identiškose taškuose, atitinkančius nurodytus kriterijus, esantys koreliacijos koeficientai (Pav 25). Dėl to, galimas jau paskaičiuotų koreliacijos koeficientų svorinių žemėlapių perstūmimas atitinkamai dešinėn ar žemyn per etaloninio kadro atitinkamą plotį ar aukštį, tačiau tokiai operacijai atlikti negalima panaudoti aukščiau aptartoje literatūroje aptartos galimybės srautiniu būdu apdoroti duomenis, todėl parinkta dar kartą perskaičiuoti koreliacijas, šįkart koreliacijos koeficientus saugant paskutinio taško vietoje. Taip gautas koreliacijos koeficientų svorinis žemėlapis, tačiau vis dar neparinkta ribinė vertė, tinkamoms koreliacijų koeficientų vertėms. Įvertinus gana prastus sistemos rezultatus, vertinant koreliacijos koeficientus tiesiogiai, nutarta, kaip ir pirmuoju atveju, perskaičiuoti koreliacijos koeficientų svorių žemėlapių sandaugos žemėlapi į skirtuminį koreliacijos koeficientų žemėlapi. Ribinė vertė nustatyta koreliacijos koeficientų skirtuminių verčių žemėlapio didžiausios vertės dešimtajai daliai. Tokiu būdu, kiekvienam galimam viršutinio kairiojo



Pav 25 Bendras kadro (kairėje) koreliacijos koeficientų žemėlapis (dešinėje) numerio rėmo taškui, tikrinamas kiekvienas toliau į dešinę bei žemyn esantis galimas apatinis dešinysis numerio rėmo taškas. Taip sistemos patikimumas padidintas iki ~3%, tačiau šįkart galimų kandidatų tikrinimams sistema užtrunka ~40-160 sekundžių, o tokie rezultatai visiškai netenkina naudojant integruotas sistemas ar realaus laiko reakcijos reikalaujančias sistemas, todėl nuspręsta pakeisti ribinę vertę ir prilyginti skirtuminio koreliacijos koeficientų žemėlapio didžiausios vertės 3/10 daliai. Po šių pakeitimų, galimų numerio rėmo taškų rekursinis tikrinimas užtrunka ~30-90 sekundžių, o atlikus detalius bandymus, sistemos patikimumo ar stabilumo pablogėjimo nepastebėta. Vis dėl to, rekursinis galimų numerio taškų tikrinimas užtrunka per ilgai, todėl nutarta pakeisti ribinę vertę į 5/10 skirtuminių koreliacijos koeficientų svorių žemėlapio maksimalios vertės dalį, o to užteko kiekvieno

galimo numerio taško rekursinio tikrinimo laikui sumažinti iki 0,5-2 sekundžių. Toks rezultatas bandomai sistemai, mano nuomone, pakankamas. Vis dėl to, bendras kadro apdorojimo laikas vis dar per didelis ~120-300 sekundžių, be to, naudojamas numerio rėmo tikrinimo algoritmas neatmeta kadru, kuriuose vietoje teksto galima rasti buferio juostą ar galinės autotransporto priemonės dalies žibintus, dėl jų horizontalių histogramų panašumų į simbolių eilutės horizontalią histogramą. Nors šio apribojimo įvedimas sumažino galimų numerio rėmų sąrašą, tačiau pastebėta neaiški tendencija: algoritmas pakankamai tiksliai atpažįsta lauko reklamų stendus. Šią sistemos savybę galima būtų išnaudoti reklamos turinio kontrolei, tačiau tai jau atskira tema ir atskiras, daug laiko reikalaujantis, darbas. Rekursinis galimų numerio rėmo taškų porų tikrinimo algoritmas vis dar tinkamai neišfiltruoja tinkamų kandidatų, todėl nutarta įverti papildomą tikrinimą kiekvienai, pagal horizontalios histogramos slėnius, įmanomai simbolių eilutei – vertikalios histogramos slėnių skaičių. Kadro suskaidymui eilutėmis, histogramos ribinę vertę parenkame pagal tos pačios histogramos vidutinę vertę, nors teoriškai pakaktų tikrinimo ar vertę daugiau nei histogramos minimumas. Kitaip tariant, pakaktų tikrinti ar išlyginto kadro histogramos eilutė viršija 0, tačiau įvertinant potencialius numerio plokštelės triukšmus, paliekamas ribinės vertės atitikimas vidutinei histogramos vertei. Tokiu būdu visos histogramos vertės, didesnės nei ribinės vertė – galimas tekstas, todėl tinkamam eilutės išpjovimui visiškai pakanka karda kirpti nuo, ribinę vertę atitinkančių verčių, pirmos pozicijos iki „kalvos“, ribinę vertę atitinkančių verčių, paskutinės pozicijos. Įvertinus galimą triukšmo įtaką histogramos vertėms, reikalingas kirpimo vietų patikslinimas, tačiau vertikaliam histogramai rasti pilnai pakanka ir esamos ribinės vertės skaidymo. Iškirptoms simbolių eilutėms, kadre vykdomas rekursinis horizontalios histogramos patikrinimas: tokiu pačiu principu kaip ir vertikaliam histogramai, randamas histogramos „slėnių“ kiekis. Ribine verte histogramos karpymui, kaip ir vertikaliam histogramai, pasirenkamas histogramos vidutinė vertė, tačiau praktinių bandymų metu pastebėta, kad tokia vidutinė vertė ne visada atitinka sąlygas, ypačingai C, F, H, N, M ir panašioms simboliams, kurių kirpimo rezultatu gaunamas į dvi ar daugiau dalių sudalintas simbolis. Dėl šios problemos, nuspręsta mažinti ribinę vertę iki 3/10 histogramos maksimalios vertės dalies. Taikant šią ribinę vertę, praktiniais bandymais nepastebėtas kelių simbolių apjungimas į vieną ar panašios klaidos. Atskyrus galimo numerio tikėtinos eilutės tikėtinius simbolius, reikia numatyti, koks simbolių kiekis yra pilnai ir visiškai pakankamas bei kelių tarpų ar simbolių tikėtinas numeris neatitinka tikrovės. Eksperimento dėlei, ribinė vertė nustatyta sekančiai: tikėtina simbolių eilutė turi būti sudaryta tarp 5 ir 10 simbolių. Šio apribojimo įvedimas nepakeitė numerio vietos radimo patikimumo tiek, kiek tikėtasi, todėl

būtinai papildomas apribojimas. Žinant, kad numerio plokštelių stiliaus formatavimai yra įvairūs, tačiau daugiau ar mažiau atitinka bendrą formatą, kai simboliai išdėstyti vienodais atstumais bei dydžiais, galima daryti išvadą, jog suskaidytų galimų simbolių ar jų tarpų pločiai turi būti vienodo ar bent panašaus dėsningumo. Šią taisyklę lengva patikrinti standartiniu nuokrypiu, rodančiu galimą skaičiaus svyravimą vidurkio atžvilgiu. Tarkime, kad visi numerio simboliai bei jų tarpai vienodo pločio. Tokiu atveju, numerio plokštelės simbolių bei jų tarpų pločiai gali būti įtakojami tik triukšmo – didelis purvo kiekis ir/arba prastos gamybos numerio plokštelės. Tokiu atveju, simbolio plotis gali būti padidėjęs, dėl ribinės vertės paklaidos, taip padidinant simbolio plotį ir sumažinant tarpo plotį. Kadangi sistema projektuojama kiek tik įmanoma universali, logiška manyti, kad simbolių bei tarpų dydžiai skirsis tarp šalių ar net šalies viduje naudojamos registracijos klasifikacijos. Dėl šios aplinkybės, numerio simbolių pločio vertinti, pagal atitikimą kokiems nors santykiams su numerio plokštelės dydžiu nėra prasmės, apart simbolio ribų išsiskyrimą iš mažiausiai 5 ir daugiau 10 simbolių numerio plokštelėje, kaip jau paminėta aukščiau. Tokiu atveju, lieka vienintelė galimybė – tikrinti atskirtų tariamų simbolių ir jų tarpų pločio standartinį nuokrypį, neviršijantį pusės numerio plokštelės pločio padalinto iš aptiktų galimų simbolių kiekio ar tiesiog minimalaus simbolių kiekio - 5. Būtina pabrėžti, kad tariamų simbolių iškirpimas (Pav 26) gali būti klaidingas, todėl standartinio nuokrypio apribojimas pagal atrastų tikėtinų simbolių kiekį – kiek nelogiškas, dėl galimo pasitaikančių šiukšlių kiekio, be to, praktiniais bandymais pastebėta, kad standartinio nuokrypio įvedimas tik pablogino situaciją, dėl esamų aukšto dažnio triukšmų – labai mažo pločio „simbolių“. Šiai situacijai spręsti galimi du keliai: keisti histogramos ribinės vertės skaičiavimą arba vertinti tik simbolius, kurių plotis atitinka minimalius reikalavimus – bent 6 taškus. Paprastumo dėlei, pasirinktas histogramos „kalvų“ pločio filtravimas, todėl galimi simboliai, kurių plotis nesiekia 6 taškų tiesiog ignoruojami, dėl ko atsiranda papildoma galimybė patikrinti histogramą – tikrinti simbolių vietas, nors antra vertus, ignoruojamų šuolių vietas susumuojamos prie tarpų pločio, todėl papildomo tikrinimo kaip ir nebereikia. Po šio pataisymo, sistemos patikimumas atstatytas, tačiau pagerėjo tik iš dalies: tikrų simbolių eilučių atpažinimas nepagerintas, tačiau padidintas išbrokuotų kandidatų kiekis. Sekantis loginis žingsnis - patikrinti turimos galimos simbolių eilutės kontrastą, tačiau taip būtų apribojamos sistemos galimybės apdoroti prasto kontrasto numerio plokšteles, o iš to apribojamas ir sistemos universalumas. Įvertinus tai, kad atpažinimo algoritme vistiek bus reikalingi vienokie ar kitokie patikrinimai, nuspręsta ištaisyti tik tikrų numerių rėmų neteisingą išbrokavimą, dėl simbolių eilutės priekyje ar gale esančių logotipų, specializuotų simbolių ar kitų žyminių formatavimų. Šią problemą pašalinti galima

tik eliminuojant netendencingai didesnę histogramos vertę turinčias vietas. Vienintelė reali galimybė atlikti tokį valymą, atsižvelgiant į skaičiavimo resursus: po skaidymo, jei simbolių kiekis netenkina užduotos sąlygos, eliminuoti pirmą bei paskutinę „kalvą“ iš histogramos vertinimo, bei skaidyti dar kartą. Jei antru tikrinimu gaunamas tinkamas simbolių kiekis, patikrinti kuri histogramos dalis buvo netinkama. Praktiniai, numerio vietos atrinkimo iš galimų kandidatų, algoritmo variacijų bandymo darbai dar vykdomi, todėl pateikti dalinius ar rinkti, konkrečių šio pakeitimo, rezultatus nėra prasmės. Šis apribojimas kai kurioms nuotraukos ženkliai padidina sistemos patikimumą, klaidingai neišbrokuojant visų galimų numerio rėmo variantų, tačiau dėl šio pakeitimo taip pat padidėjo ir vienos galimos numerio rėmo taškų poros tikrinimo laikas.

Turint numerio vietą, būtinas numerio simbolių atpažinimas, tačiau be papildomo valymo, kadre galima nieko neaptikti, ypač jei kadro kontrastas labai žemas ar histogramos verčių pasiskirstymas neproporcingai arti krašto. Todėl prieš bet kokius tolimesnius kadro apdorojimus, būtinas kadro kontrasto bei šviesumo sutvarkymas, kam puikiai tinka histogramos išlyginimas bei „top-hat“ ir „bottom-hat“ morfologinių operacijų naudojimas. Paprastumo dėlei naudojamas histogramos išlyginimas, taip padidinant kontrastą ir iš dalies sutvarkant šviesumą. Dėl to, kad numerio rėmas nebūtinai sudarytas iš numerio simbolių, tačiau gali turėti ir papildomų reklaminių, į numerio formatą patenkančių užrašų ar kitą simbolių eilutę pildančių ženklinimų, būtinas numerio plokštelės skaidymas eilutėmis. Kaip



Pav 26. Numerio skaidymas.

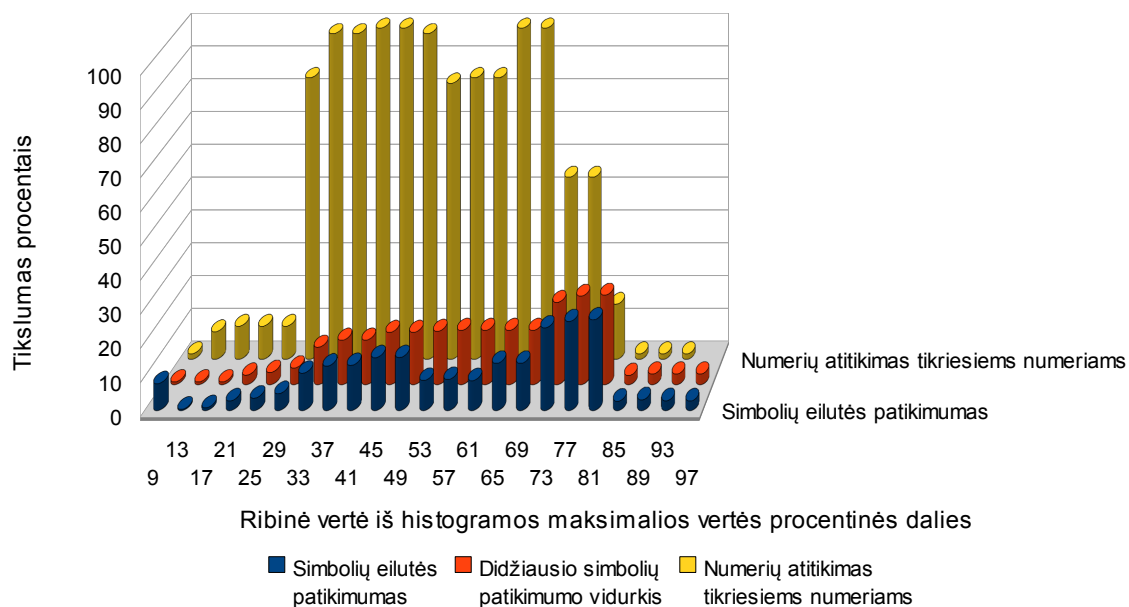
Iš kairės: Originali numerio plokštelė, negatyvas, vertikali projekcija.

Raudona linija žymi histogramos ribinės vertės poziciją

jau anksčiau minėta, simbolių skaidymui eilutėmis puikiai tinka histogramos slėnių paieška, kur, kaip ir kitiems filtruojantiems algoritmams, būtina ribinė vertė. Ribinės vertės nustatymą nuspręsta parinkti pagal histogramos vidutinės vertės 6/10 dalį. Tokiu ribinės vertės nustatymu, numerio plokštelės simbolių eilutės daugeliu atvejų sėkmingai išskaidomos. Vis dėl to, pastebėta purvo bei apšvietimo įtaka skaidymo tikslumui, todėl nuspręsta apdoroti

kadra negatyvo forma (Pav 27), kam reikalingas turimo kadro pervertimas. Vertimas į negatyvą – gana paprasta, didelių skaičiavimų nereikalaujanti aritmetinė operacija, tačiau dėl kadro apdorojimui naudojamo išlyginimo, gaunamos pakartotinos masyvo didžiausios bei mažiausios reikšmės paieškos. Negatyvas randamas pagal formulę:

$$x'_i = x_{MAX} - x_i \quad (7)$$



Pav 27. Ribinė vertė pagal histogramos maksimalios vertės dalį

Čia  $x_i$  - i-toji taško reikšmė,  $x_{MAX}$  – maksimali visų taškų reikšmė, o  $x'_i$  - nauja i-tojo taško reikšmė. Šių pakeitimų integravimas į sistemą leido tiksliau atskirti simbolių eilutes, tačiau vis dar neatliktas simbolių eilutės skaidymas atskirais simboliais, ką galima pasiekti pasinaudojus tuo pačiu principu – histogramos slėniais, tik šiuo atveju stulpeliams. Tokiu atveju, tenka tik parinkti tinkamą histogramos ribinę vertę, pagal kurią priklausys simbolių atskyrimo tikslumas. Šiam tikslui, histogramos ribinė vertė (Pav 27) nustatoma 6/10 histogramos vidutinės vertės dalies, taip priimant visas įmanomas reikšmes, viršijančias nurodytą histogramos ribinę vertę. Vis dėl to histogramos ribinė vertė nurodo tik dominančią liniją histogramos grafike, tačiau neparodo kur ir kokio pločio simboliai yra. Tam, gautuose rezultatuose, būtina ieškoti sekos: skaičiuoti, kiek pozicijų nuo pirmos atitinkančios vertės, atitinka užsibrėžtą limitą. Šitoks sekos ieškojimas duoda dvejopus rezultatus: simbolių pradžias bei pločius ir tarpų tarp šių simbolių pozicijas bei pločius. Turint šiuos duomenis jau galima organizuoti ir pačių simbolių atpažinimą. Šioje vietoje tenka naudoti rekursiją kiekvienam kadro ir duomenų bazės simboliui. Iškirptas simbolis tik laboratorinėmis

sąlygomis gali sutapti su duomenų bazėje saugomo simbolio dydžiu ar pilkumo pustonų skalėje apdorojamo, kaip ir duomenų bazėje saugomų simbolių, vertės, todėl šiuo atveju koreliacija tik pagerins simbolio atpažinimo procesą, kas abejotina bitinės formos vaizdą lyginančiuose algoritmuose. Vien vaizdo konvertavimo į bitinę formą metu, gaunami papildomi vaizdo iškraipymai, mažinantys galimybę kiek įmanoma tiksliai atpažinti esamus simbolius. Dėl to, kad tik praktiškai laboratorinėmis sąlygomis įmanomas simbolių dydžių bei aspektų atitikimas, prieš tikrinant galimus simbolius, būtina iškirptam simboliui pakeisti mastelį iki duomenų bazėje laikomų simbolių dydžio, taip supaprastinant simbolio atpažinimo procedūrą. Tačiau vien dydžio pakeitimas gali įvesti papildomus triukšmus, jei simbolis turi būti padidintas, o ne sumažintas: didinant iškirpto simbolio paveikslo matmenis, reikia užpildyti susidariusius tarpus, kitu atveju, vaizde atsiradusių triukšmų gali užtekti simbolio atpažinimo klaidai įsivelti. Tokiu atveju, jei duomenų bazėje saugomi simboliai yra aukštos kokybės, tai reikštų vienareikšmišką sistemos patikimumo nebuvimą. Šioje vietoje galima pastebėti saugomo simbolio kokybės problematiką, dėl kurios būtina apsvarstyti sistemai tinkamą simbolio etalono raišką, nusprendžiančią sistemos simbolio atpažinimo algoritmo tvarkingą darbą.

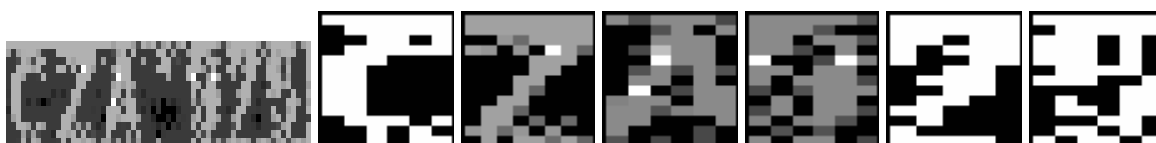
Kaip jau buvo teigta literatūroje, mažiausias tinkamas atpažinti simbolis turi būti bent 25 taškų aukščio. Tačiau šis teiginys daugiau tinkamas nebent bitinės formos kadra lyginantiems algoritmams, dėl šių prasto atsparumo dideliems triukšmams. Mano nuomone, koreliacijos naudojimas leidžia šį limitą sumažinti bent iki 8 taškų privalomo simbolio aukščio, todėl galima teigti, kad tinkamiausia duomenų bazės simbolio raiška – 8 taškų aukščio simbolis. Turint omeny, kad simboliai tiek tekstinėse konsolėse, tiek grafinėse sąsajose, tiek šriftus apibūdinančiuose failuose aprašomi kvadrato arba dvigubo aukščio stačiakampio forma, duomenų bazės simbolių saugojimui tinkamiausias 8x8 taškų simbolis. Tačiau jau šioje vietoje susiduriama su per mažos kokybės bazinio simbolio problema: koreliacijos koeficiento skaičiavimui turimi tik  $8 \times 8 = 64$  reikšminiai taškai, o tiek yra pastebimai per mažai, ženkliam atitikusio simbolio koreliacijos koeficiento iškilimui virš bendros koreliacijos koeficientų masės. Be to, tokios žemos raiškos bazinio simbolio naudojimas išprovokuoja per daug ryškų pasikeitimą į simbolio kadra įsivėlus, net ir palyginus mažiems, triukšmams ar ne visiškai tiksliai aprašius simbolius, baziniais simboliais, duomenų bazėje. Dėl to, kad sistema projektuojama kaip įmanoma universali, mažos raiškos bazinio simbolio naudojimas sudarytų problemų atpažįstant panašų, tačiau netapatų, numerio plokštelėje naudojamą, šriftą, o toks sistemos nestabilumas daugia-nacionalinių numerių

prasmė yra netoleruotinas. Galima pastebėti, kad šioje vietoje atsiranda dviprasmiška situacija: didinant bazinių simbolių raišką, mažinama sistemos simbolio atpažinimo tikimybė prastesnės kokybės kadrams, tačiau ir sumažinus bazinių simbolių raišką, sistemos tikimybė sėkmingai atpažinti simbolį nepadidėja, o veikia dar labiau sumažinama dėl potencialiai galimų simbolių šriftų neatitikimo ar net baziniame simbolio vaizde įsivėlusios klaidos. Šiai dviprasmiškai problemai spręsti nutarta naudoti bent 64x64 dydžio bazinius simbolius, taip užtikrinant pakankamai mažą vieno blogai aprašyto taško įtaką bendram koreliacijos koeficientui. Be to, tokia raiška turėtų būti pilnai pakankama net ir įmantriausių hieroglifų aprašymui, ar plačiai naudojamiems lotyniškiems simboliams. Žinant, kaip nuo bazinio simbolio raiškos priklauso simbolio atpažinimo procese naudojamo, iš kadro iškirpto, simbolio kokybė, būtina keisti ir, iš kadro iškerpamų, simbolių mastelio keitimo algoritmą. Vienas galimų šios problemos sprendimo variantų – didinant kadro simbolį, taškų vertes apskaičiuoti naudojantis gretimų taškų interpoliacija, taip užpildant tarpinius taškus maksimalia pilkumo skalės taško gradiento reikšme – 255, kai pradiniam kadre neegzistavęs taškas atsiranda simbolio silueto viduryje, ir analogiškai atvirkščiai. Be to, tokios sistemos naudojimas leistų bent iš dalies kompensuoti simbolio silueto kraštuose esančių taškų išnykimą, keičiantis simbolio masteliui, taip bent iš dalies neprarandant koreliacijos koeficiento sudedamųjų teigiamų verčių, radus simbolio bei jo etaloninio duomenų bazės simbolio panašumą. Šis dalinis kraštinių taškų kompensavimas atsiranda dėl simbolio silueto kraštuose atsirandančio fono bei simbolio taškų gradiento vertės vidurkio ar vidurkio atitinkamos dalies, priklausomai nuo to, kiek papildomų taškų tarp turimų taškų atsiranda. Iš to, kadro simbolius didinant daugiau nei dvigubai, iš numerio plokštelės iškirptas simbolis gali tapti sunkiai atpažįstamas dėl panašių į išblukinimo operaciją efektų, o dėl to ir atsirandantis netinkamumas naudojimui sistemoje. Tačiau šioje vietoje galima pastebėti per daug mažų simbolių ar per daug didelės raiškos bazinių simbolių sukeliama, numerio plokštelės simbolių didinimo metu atsirandančio, išblukinimo efekto. Žinant šį faktą, galima daryti prielaidą, kad per daug maži numerio plokštelės simboliai nėra tokia didelė problema kaip manyta: užtenka turimą išdidintą simbolį apdoroti, atliekant aštrinimo operaciją arba tokiu pačiu lygiu išblukinti bazinius simbolius. Pastarasis atvejis labiau priimtinas dėl mažesnių skaičiavimo resursų reikalaujančio algoritmo: išblukinimą užtektų įvykdyti tik vieną kartą tai pačiai numerio plokštei. Atliekant praktinius bandymus, patikrinta dauguma iš čia aprašytų tezių: pradžioje naudojant 8x8 dydžio etaloninį simbolį pastebėtas per didelis sistemos nestabilumas - atpažįstama vos ~20% numerio plokštelių, kurių patikimumo procentas neviršija  $10^{-6}$ , o vidutinė simbolio „tikrumo“ tikimybė  $\sim 5,6 \cdot 10^{-8}$ . Šiems

dydžiams įvertinti, sistemos pilnas patikimumas vertinamas pagal atitikimą auto-koreliacijos funkcijos koreliacijos koeficientui, o neatitikimo tikimybė atitinka ryšio-koreliacijos koeficientą tarp simbolio ir simbolio negatyvo. Pagal šiuos atitikimo bei neatitikimo koreliacijų koeficientus nustatoma, kad pilnas atitikimas atitinka skaitinę vertę 1000, o analogiškai neatitikimas – 0. Taip koreliacijos koeficientus normuojami tarp 0 ir 1000, gaunamos atitikimo tikimybės per žemos. Todėl, dėl tokių prastų atitikimo rezultatų, nuspręsta pakeisti bazinio simbolio raišką iki 64x64 taškų. Toks pakeitimas sąlygojo numerio atpažinimo atitikimo koeficiento padidėjimą iki ~400-600. Simbolių aprašymai nėra ir tik laboratorinėmis sąlygomis įmanoma, kad būtų visiškai tikslūs, be to, atitikimo koeficiento viršutinė riba nustatyta pagal auto-koreliacijos koeficientą, reiškiantį simbolio visišką įmanomą atitikimą, todėl ~60% (600 iš 1000) atitikimo tikslumas yra daugiau nei tikėtasi. Be to, auto-koreliacijos funkcijos koreliacijos koeficientas nusako visišką simbolio sutapimą, o numerio kadre tokios sąlygos praktiškai galimos tik laboratorinėmis sąlygomis, dėl visada egzistuojančių triukšmų dėl pilkumo skalės gradiento verčių nepastovumo, tiek dėl pastovaus visam kadruil pilkumo skalės gradiento verčių svyravimų, sukeltų kameros CCD elemento neidealaus suderinimo, tiek dėl aplinkos sąlygų: numerio dažų, šešėlių ar terpės, tarp objekto ir kameros CCD elemento, savybių ar sudedamųjų dalių. Šios svyravimus sėkmingai pašalina bitinę formą naudojančios simbolių atpažinimo algoritmai, tačiau, šios sistemos atveju, šie svyravimai turi būti įvertinti ir pridėti, vertinant gaunamus rezultatus. Būtent dėl šių svyravimų, galima suprasti, kodėl atpažinimo koeficientas svyruoja apie 400 – 600. Tarkime, kad svyravimų iškraipymai sudaro ~30%, tuomet iš naujo normavus atpažinimo koeficientus 400 - 600 būtų atitinkamai ~580 – 860, o 86% galimybė, kad rezultatas idealiai atitinka duomenų bazės simbolius, mano nuomone, yra pakankamai aukšta. Šie koeficientai nerodo realios tikimybės, o tik atitikimą etalonui duomenų bazėje, be to, verta apsvarstyti kitą rezultato patikimumo matavimo metodiką: sekti geriausio bei antro geriausio atitikimo simbolių, normuotus koreliacijos atitikimo koeficientus. Taip galima išreikšti atpažinto simbolio atotrūkį nuo sekančio potencialiai tinkančio simbolio, bet tokiu atveju šiek tiek padidėja, papildomų skaičiavimų prasme, klaidingai atpažinto simbolio paieška, pagal kitus to paties regiono atpažintus numerius, dėl netiesioginės koreliacijos atitikimo koeficientų kaupimo. Praktiniais bandymais numerio simbolių atpažinimo algoritmu apdorojant iš anksto iškirptus numerius, pastebėta, kad simboliai atpažįstami ~86% tikslumu, o įvertinus atitikimo koeficientą – ~92%. O tai reiškia, kad simbolių atpažinimo modulis sėkmingai atpažįsta simbolius, atmesdamas mažai tikėtinius simbolius. Be to, dėl aktyvaus programinės įrangos derinimo, taisymo, pildymų bei bandymų ciklo, galima daryti prielaidą, kad toks, reliatyviai



žemas, rezultatas gaunamas tik dėl nepilno programinės įrangos išgryninimo bei nevisai tinkamos koreliacijos funkcijos formulės. Sistema projektuojama ne specifinėms darbo sąlygoms, o universaliam darbui bet kokiais sąlygomis: neapsiribojant žinomu šviesumu, žinomu numerio formatu ar tuo labiau žinoma kadro bei kadre esančio numerio plokštelės kokybe. Vis dėl to, bet kuriuo atveju, šis tikslumas galioja tik rankiniu būdu iškerpamoms nuotraukoms ir, kaip jau daugelio kitų autorių pabrėžta, pagrindinės numerio atpažinimo sistemos klaidos gaunamos dėl netinkamo šviesumo ar netikslaus numerio vietos bei simbolių karpymo. Praktiniais bandymais pastebėta, kad prastesnės kokybės numerio plokštelėse esant panašioms simboliais, o sistemai neatsižvelgiant į bet kokius iš anksto žinomus duomenis apie numerio formatą, panašių simbolių koreliacijos atitikimo koeficientai pernelyg mažai skiriasi nuo tikrojo simbolio koreliacijos atitikimo koeficiento vertės – skirtumas vos 20-50 vienetų, todėl nuspręsta simbolių atpažinimo rekursijoje vertinti ne simbolių atitikimo koeficientą, o simbolių atitikimo koeficiento ir negatyvo simbolio atitikimo koeficiento santykį. Taip iš dalies eliminuojamas, pagal duomenų bazės simbolio aprašymą, fono dalis, apsprendžianti kitokio simbolio atpažinimą. Vis dėl to, toks pakeitimas sąlygoja dvigubą, būtinų koreliacijos koeficientų, skaičiavimo kiekį, o dėl to, esant didesniems atpažintinų numerio plokštelių kiekiams, ženkliai sumažėja sistemos pralaidumas.



Pav 28. Numerio simboliai.

Iš kairės: numerio plokštelė bei atskirti simboliai

Atpažinus atskirus numerio plokštelės simbolius (Pav 28) kol kas nieko negalima pasakyti apie galutinį numerį apskritai: žinomi tik atskiri numerio simboliai bei jų koreliacijos atitikimo koeficientai. Turint simbolių koreliacijos atitikimo koeficientus, galima rasti visos simbolių eilutes bendrą koreliacijos atitikimo koeficientą. Be to, turint atpažintus simbolius, galimos išvados apie ką tik apdorotos numerio plokštelės tikrumą bei taisyklingumą, atsižvelgiant į konkrečias numerio formato taisykles bei kur šie duomenys yra, be to pridėti papildomų duomenų apie numerį – valstija, miestas, regionas ar auto-salonas. Šių duomenų saugojimas leidžia kaupti statistinę informaciją apie geriausiai ar blogiausiai atpažįstamas numerio plokšteles bei jų trūkumus: nepakankamas šviesos šaltinio atspindėjimas, apšvietimo bangų dažnio absorbavimas, skirtingas šriftas ir panašiai bei pagal tai testuoti sistemos darbo tyrinėjimus ar siekti tokių numerio plokštelių patobulinimo. Be to, turint jau atpažintus

simbolių, bei žinant numerio formatą, galima specifinių žymų paieška, kaip autotransporto priemonės techninės apžiūros galiojimas, tačiau tai jau atskira tema. Kaip jau anksčiau minėta, tam pačiam numerio regionui gali būti aptiktas ne vienas numerio regiono galimas taškas, todėl ir numerio atpažinimui perduodamų tų pačių numerio plokštelių kiekis gali būti didesnis nei viena numerio plokštelė. Žinant šią sistemos savybę, galimas antrinis atpažinto numerio patikrinimas, įvertinant tame pačiame kadro regione atpažintų simbolių vienodumą: tikrinti tame pačiame regione atpažintų numerių simbolių tapatumą tarpusavyje. Taip iš trijų ar daugiau, to paties regiono atpažintų numerių, galima išskaičiuoti viename iš numerių esančią klaidą. Sistemos stabilumo prasme, net jei vienas ar keli numeriai klaidingai atpažino vieną ar daugiau simbolių neteisingai, galutinis rezultatas vistiek gaunamas teisingas. Tačiau ir šioje vietoje galima įžvelgti potencialią spragą, kai prastai atpažinto simbolio ar simbolių koreliacijos atpažinimo koeficientai nenusako skirtumo, tarp tinkamo ir į jį panašaus simbolio, ar tampa didesni nei tikrojo simbolio koreliacijos atitikimo koeficientai. Be to, tokį atvejį galima ištaisyti turint simbolių pasiskirstymą bei nuokrypį nuo kitų, tame pačiame regione atpažintų, simbolių, tačiau šioje vietoje tenka daryti prielaidą, kad daugelio kitų, to paties regiono, numerių simbolių atpažinimas yra teisingas, o tai nebūtinai gali būti tiesa. Todėl šioje vietoje susiduriama su dilema: jei keli, to paties regiono, atpažinti numeriai yra teisingi, tačiau keli numeriai ar numerių simboliai atpažįstami klaidingai, klaidą galima atstatyti įvertinant kitus, to paties regiono, atpažintus numerius, tačiau, jei tarp atpažintų, to paties regiono, numerių klaidingai atpažinta didžioji dalis simbolių ar numerių, vienintelė patikros priemonė lieka atpažinto simbolio koreliacijos atitikimo koeficientas bei išankstinės žinios apie numerio formatą. Vis dėl to, numerio formato taikymas numerio patikrinimui tinka tiek tuo atveju, kai žinomi visi numerio formatai: tiek nacionaliniai, tiek galimi pašaliniai, be to, neturint visų įmanomų numerio formatų, šio patikrinimo taikyti apskritai negalima, dėl galimo formatų konflikto ir dėl to sąlygoto tinkamo numerio atmetimo. Vis dėl to, būtina pabrėžti, kad pravažiavimo kontrolei tikrinami numeriai dauguma atveju yra žinomi, nebent sistema taikoma sustiprinto saugumo zonų pravažiavimo kontrolei – fiksuoti į zoną patenkančias bei išvykstančias autotransporto priemones bei šių numerius tikrinti duomenų bazėje. Šiuo atveju vykdant pravažiavimo kontrolę, pagal atpažintą numerį, numerio formato taikymas apskritai nebūtinai arba reikalingas tik kaip papildoma patikros priemonė – bet kuriuo atveju leidžiami numeriai yra žinomi ir tik žinomo numerio atpažinimas sąlygoja transporto priemonės įleidimą. Sugriežtinto saugumo pravažiavimo stebėjimo posto atžvilgiu, papildomas numerio formos stebėjimas tinkamas nebent preliminariai nustatant numerio kilmės šalį ar klasę – kurios šalies formatavimas ar kokio pobūdžio numeris: asmeniniai, specializuoti, karinių

dalinių ir pan. O tai leidžia daryti prielaidą, kad bendriniam sistemos darbui, kam ir yra projektuojama sistema, numerio formato turėjimas apskritai nereikalingas ar net kenksmingas. Specializuotais atvejais, numerio formatas gali būti tikrinamas už sistemos ribų – papildomose sistemose, kurios gauna galutinį šios sistemos rezultatą: atpažintą ar atpažintus numerius. Be to, dėl algoritmo realizavimui naudojamų Qt bibliotekų bei C++ kalbos, papildomo kodo, kuriuo gali būti atliekami papildomi tikrinamo numerio rėmo simbolių išdėstymo formos patikrinimai, realizavimas pačioje programoje itin paprastas, dėl iš anksto paruoštos priedų architektūros, neapribojančios sistemos darbo konkrečioje aparatinėje architektūroje. Įvertinus visa tai, galima daryti prielaidą, kad sistemos patikimumui ir tikslumui užtikrinti būtinas tik numerio radimo ar karpymo algoritmų sutvarkymas bei naudojamos koreliacijos funkcijos optimizavimas, turimam signalo pobūdžiui, kai kadras traktuojamas kaip ištinis bei baigtinio ilgio signalas. Integruotų sistemų ar realaus darbo reikalaujančių sistemų atžvilgiu, abejotino būtinumo numerio formato tikrinimas tik bereikalingai išnaudoja svarbius skaičiavimo resursus, todėl tokiose sistemos galimas tik, anksčiau paminėtas, sprendimas, naudoti išoriškai prijungiamus priedus – sistemos dalį, atsakingą už sąsają su vartotoju. Tokiu būdu, sistemos branduolio apdorotas vaizdas siunčiamas vartotojo daliai, kurioje ir atliekami papildomi patikrinimai, susiję su sistemos naudojimo sąlygų specifiška.

Be numerio formato atpažinimo problematikos, taikant sistemą dideliai simbolių duomenų bazei, simbolio atpažinimo proceso pralaidumas tiesiogiai priklauso nuo duomenų bazėje turimų simbolių kiekio, todėl būtinas kitoks simbolių atpažinimo algoritmas, gebantis atpažinti simbolius hierarchiniu būdu. Tokia sistema jau buvo aptarta literatūroje, kai pirmiausia atpažįstami bendri kelių simbolių požymiai ir tik tada atpažįstamas galutinis simbolis. Tokiu būdu bendrus panašumus turintys, net ir skirtingų šriftų simboliai, gali būti aprašyti keliomis bendromis savybėmis. Literatūroje paminėtas hierarchiniu būdu simbolius atpažįstantis algoritmas leidžia naudoti tik dviejų lygių hierarchinę simbolių aprašymo struktūrą, tačiau šios sistemos atveju, neribotas hierarchinio simbolių aprašymo gylis leidžia realizuoti visų šriftų simbolių bendrus panašumus keliais duomenų bazės hierarchinės struktūros lygiais, taip pradinį algoritmo būtiną skaičiavimo resursą sumažinant bent pusiau.

Dar kartą įvertinus literatūroje pateikiamas rekomendacijas sistemų apšvietimui stabilizuoti, būtina apsvarstyti ir programinius šviesumo kompensavimo metodus, antraip sistemai tinkamas apšvietimo ruožas gali riboti sistemos darbą tik konkrečiam paros metui ar apšvietimo sistemai. Šią dilemą programiškai galima išspręsti naudojant gama korekciją,



Pav 29. Šviesumo kompensavimas.  
Originalus kadras (kairėje) ir gama kompensuotas kadro negatyvas

tačiau tokia kompensacija tinka tik naudojant pakankamo jautrumo kameros CCD elementus, kitu atveju bus gaunamos tik kelios kadro pilkumo skalės reikšmės, o to visiškai nepakanka kadro šviesumo bei kontrasto kompensavimui (Pav 29). Įvertinus šias problemas, logiška būtų teigti, kad bent minimalus infra-raudonųjų spindulių apšvietimas naudojamai kamerai yra būtinas, kitu atveju kameros kaina, dėl ne būtiniam naudojimui skirtų elementų, kyla pagal norimą jautrumą ar šviesumo bei kontrasto darbinį verčių ruožą. Vis dėl to, gama šviesumo korekcijos programinis algoritmas leidžia kompensuoti per daug šviesius kadrus, kai kamera apšviečiama pašalinio parazitinio intensyvios šviesos šaltinio: daugelis stebėjimo kamerų turi automatinį šviesumo reguliavimą pagal gaunamus CCD matricos lygių pasikeitimus, tačiau tik aukštos kokybės kameros geba pakankamai greitai reaguoti į atsiradusius šviesumo pasikeitimus. Atsižvelgiant į sistemos per sekundę apdorojamų kadrų kiekį, sistemoje įgyvendinta gama korekcija leistų operatyviau reaguoti į pasikeitusias apšvietimo sąlygas. Be to, programinė gama korekcija gali būti sujungta su kameros vidinio mikrovaldiklio programine dalimi, taip įtakojant CCD jautrumą. Šis papildomas programinis apdorojimas, kaip ir turi būti, įtakos sistemos darbo pralaidumą, tačiau toks papildomas algoritmas tik padidina sistemos universalumą, ką galima pastebėti iš, sistemos priklausomybės nuo šviesumo, grafiko. Integruotų sistemų ar realaus laiko būtinumo atžvilgiu, šviesumo įvertinimas nėra būtinas, šias funkcijas galima įgyvendinti ir aparatinėje įrangos pusėje ar pačios kameros programinėje dalyje. Bet kuriuo atveju, gaunamo šviesumo dinamikos ignoravimas sistemos darbo patikimumo tikrai nedidina, todėl tiek aparatinio įgyvendinimo atveju, tiek programinio įgyvendinimo atveju, gama kompensacija tik praplečia sistemos darbinės sąlygas, o tai ir yra siekiamas rezultatas. Gama kompensacijos algoritmui įgyvendinti, nutarta rasti verčių pasiskirstymą tankį keliais kanalais: lygiai taip pat, kaip skaičiuojamos histogramos, tik šiuo atveju vertinti ne kanalų kiekį, tapatų verčių gradiento kiekiui, o rankiniu būdu nurodytam konkrečiam kanalų kiekiui, taip visas vertes suskirstant į atitinkamus kanalų verčių ruošus. Paprastumo dėlei, pasirinkti 8 kanalai: iš 256 galimų kadro verčių, histograma skaičiuojama lyg verčių būtų tik 8. Tokiu būdu, per visą 256 verčių ruožą gaunami 8 histogramos stulpeliai, atitinkantys 256/8 verčių žingsnį: nulinis ruožas atitinka vertes nuo 0 iki 31, pirmas ruožas atitinka vertes nuo 32 iki 63 ir t.t. Histogramos prasmė –

parodyti verčių pasiskirstymo tankį, todėl išskaidžius visas įmanomas kadro vertes keliais kanalais, galima vertinti, kuris kanalas yra tankiausiai naudojamas, kadro verčių prasme. Logiška, kad žemutiniams kanalams rodant neproporcingai didesnę didesnę verčių tankį nei aukštesniems kanalams - tikrinamas kadras per tamsus (Pav 29), todėl bendras kadro intensyvumas turi būti didinamas. Didinimo proceso metu, aukščiausiam kanalui priskaičiuotos vertės atsiranda už, buvusios maksimalios, kadro verčių ribos, todėl tenka viršutines vertes ignoruoti, jas prilyginant maksimaliai leistinai vertai ir taip prarandant duomenis. Duomenų praradimo problematiką galima spręsti papildomai į intensyvumo kėlimo algoritmą įvedant interpoliacijos pagrindu grįstą algoritmą aplinkinių verčių procentiniam, o ne tiesiniam didinimui. Toks galimas problemos sprendimo būdas kol kas neišbandytas praktiškai, dėl pakankamos sistemos darbui kadro kokybės, po gama kompensacijos. Vis dėl to, tokia gama kompensacija ypatingai intensyvi skaičiavimo resursų prasme, todėl integruotose sistemose ar realaus laiko reakcijos reikalaujančiose sistemose šios papildomos funkcijos programinės realizacijos naudojimas spęstinas. Būtina pabrėžti, kad šis, gama kompensacijos, algoritmas išbandytas tik tamsiomis prieblandos sąlygomis, o ne naktinio apšvietimo sąlygomis, todėl tolimesnis šio algoritmo, ypač be papildomų funkcijos patobulinimų, naudojimas spęstinas. Kaip rodo praktika, kartais gaunami apdorojimui kadrai būna per šviesūs, todėl būtinas ir atvirkščias ribotos histogramos kanalų įvertinimas. Tai nesudėtingai galima padaryti lyginant aukščiausių histogramos kanalų verčių pasiskirstymo tankį su žemesnių histogramos kanalų verčių tankiu. Praktiniams funkcijos bandymams parinkta tiek šviesinimo tiek tamsinimo ribinė vertė – 1,2x kitų kanalų suminės vertės: t.y. jei žemiausių kanalų suminė vertė 1,2 karto didesnė, nei aukščiausių ar vidurinių kanalų suminė vertė, kadrai būtina šviesinti ir analogiškai atvirkščiai. Vien nusakyti, kad kadrai būtinas šviesinimas ar tamsinimas neužtenka – būtina žinoti šviesinimo ar tamsinimo ribas, todėl šių operacijų intensyvumo postūmio gradientas priskirtas pagal maksimalaus gradiento verčių tankio kanalo numerio skirtumą iki centrinio kanalo numerio. Vien vertinant kanalo numerius, gama kompensacijos rekursinis algoritmas gali būti skaičiuojamas šimtus kartų iki tenkinančių kadro verčių tankio pasiskirstymo sąlygų, todėl numerio skirtumas dauginamas iš kanalo dydžio pagal formulę:

$$\Delta x = (C_{VID} - C_{MAX}) \cdot N_C \quad .(8)$$

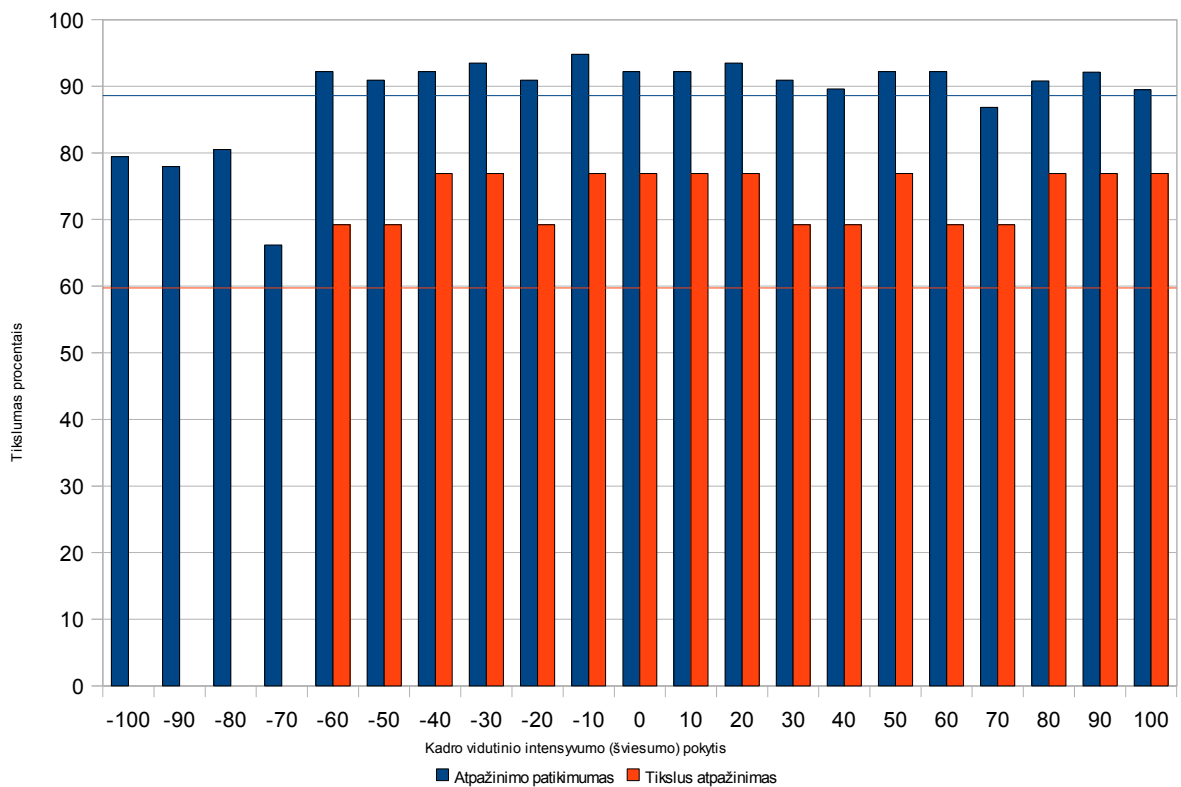
Čia  $\Delta X$  - verčių poslinkis,  $C_{VID}$  - vidurinio kanalo numeris,  $C_{MAX}$  - maksimalią vertę turinčio kanalo numeris, bei  $N_C$  - kanalo dydis. Šiuo atveju laikoma, kad kadro verčių tankiai turi pasiskirstyti pagal Gauso natūraliojo pasiskirstymo funkciją, todėl ir vertinamas

kanalų atstumas iki centrinio kanalo. Kanalo dydis paskaičiuojamas iš kadro galimų verčių kiekio, padalinto iš kanalų kiekio iš formulės:

$$N_C = (X_{MAX} - X_{MIN}) / X_C \quad (9)$$

Čia  $X_{MAX}$  – maksimali taškų vertė,  $X_{MIN}$  - minimali taškų vertė, bei  $X_C$  - vieną kanalą atitinkančių verčių kiekis, o  $N_C$  – kanalų kiekis. Tokiu būdu, kadro verčių pasiskirstymo tankio grafikui esant arčiau koordinatų centro, verčių tankio funkcijos grafikas perstumiamas tolyn nuo koordinatų pradžios taško per reikšminių kanalų kiekį atstojančią vertę. Dėl tokio verčių perstūmimo, pilnai pakanka tik vienos ar kelių rekursijos ciklų gama kompensacijai užbaigti, tačiau būtent dėl rekursijos ciklo atsiranda poreikis riboti verčių pasiskirstymo tankio grafiko postūmius pagal rekursijos ciklo kartą, todėl poslinkio funkcija modifikuojama pagal formulę:

$$\Delta X = (C_{VID} - C_{MAX}) \cdot X_C / n \quad (10)$$



Pav 30. Sistemos atpažinimo priklausomybė nuo vidutinio intensyvumo (šviesumo)

Čia kintamieji tokie pat, kaip formulėje 8, o  $n$  - rekursijos ciklo pasikartojimo kartas. Įvertinant galimą programos klaidą ar nerandamą tinkamą gama korekcijos poslinkį, papildomai įvedamas galimų gama kompensacijos algoritmo rekursijos ciklo kartų ribojimas. Šiuo atveju rekursijos ciklų kiekis apribojamas iki 10 galimų ciklų. Bet kuriuo atveju, toks

intensyvus kadro apdorojimas resursų prasme, galimas tik laboratorinėmis sąlygomis vykdomiems skaičiavimams ar tarnybinę stotį naudojančioms pravažiavimo kontrolės sistemoms. Dėl to, kad aparatinėje dalyje nesudėtinga įgyvendinti masiškai paralelinius skaičiavimus, galimas ir dar vienas alternatyvus, FPGA SoC integruotoms sistemoms ar ypač realaus laiko reakcijos reikalaujančioms sistemoms tinkantis, sprendimas: aparatinėje pusėje įgyvendinti už kadro gama kompensaciją atsakingą ar atsakingus papildomus lustus, kurių vienintelis rezultatas – gama kompensacijos atžvilgiu sutvarkytas kadras, o tokia schema jau tinka bet kokiai, laiko atžvilgiu jautriai sistemai. Sistemos priklausomybė nuo šviesumo pateikta 29 paveiksle.

### ***Algoritmo adaptavimas realaus laiko sistemoms***

Pagrindinis šių sistemų reikalavimas – labai žemas reakcijos į žinomą įvykį laikas. Tokiose sistemos negalima naudoti jokių, intensyvių skaičiavimus vykdančių algoritmų arba šiuos vykdyti masiškai paraleliai taip, kad rezultatas būtų gaunamas per palyginus trumpą laiko tarpą. Būtent šie reikalavimai apriboja aukščiau aptartos daugiau universalios autotransporto priemonių numerių radimo bei atpažinimo sistemos funkcionalumą. Peržvelgus aprašytą metodiką įdėmiau, galima pastebėti, kad per visus metodikos algoritmus naudojamos vien dinaminės ribinės vertės, o tai netgi labai tinka universaliai sistemai, dirbančiai bet kokiomis sąlygomis, tačiau ne visada tinkama greito reagavimo reikalaujančiose sistemose. Be to, universaliajame šios sistemos variante įvesta aibė papildomų apribojimų bei filtravimų kiek įmanoma tikslesniam sistemos darbui, tačiau realaus laiko reakcijos reikalaujančiose sistemose toks tikslumas bet kokiomis aplinkos sąlygomis ne visada reikalingas ar naudingas, todėl kelių algoritmų atsisakymas ar supaprastinimas, prieš tai netinkamą, sistemą paverčia tinkama, šio tipo uždaviniams spręsti. Viena pirmųjų universaliajame algoritme naudojamų funkcijų – autotransporto priemonės buvimo kadre aptikimas. Ši funkcija nėra tokia sudėtinga ar reikalaujanti didelių skaičiavimo resursų, tačiau kiekviena papildomai išnaudota procesoriaus instrukcija ilgina skaičiavimo laiką, todėl vietoje pilno programinio įgyvendinimo, verta rinktis iš dalies programinį bei aparatinį ar pilnai aparatinį įgyvendinimą, taip eliminuojant bet kokį neatitikusių taškų skaičiavimo poreikį, o tuo ir bereikalingą skaičiavimo resursų eikvojimą. Deja šioje vietoje įveliamas papildomas sistemos nežinomas – aparatinis autotransporto priemonės fiksavimas, sąlygojamas nežinomo tipo bei savybių daviklio. Vis dėl to, tokia konstrukcija leidžia nurodyti, kad gautame kadre yra transporto priemonė ir greičiausiai yra užfiksuotas autotransporto priemonės valstybinis registracijos

numeris, tačiau ši konstrukcija visiškai nenusako kodėl suveikė aparatinis daviklis. Šio neapibrėžtumo problema – negalima tiksliai pasakyti, kad aparatinis daviklis nesuveikė nuo infra-raudonų spindulių barjerą kirtusio katino, elektromagnetinį barjerą kirtusio dviračio ar judesio daviklio veikimo zoną kliudžiusio žmogaus. Vis dėl to, šių ir panašių atvejų filtravimas nėra pagrindinis šios sistemos uždavinys, todėl pilno ar dalinio aparatinio daviklio panaudojimas yra tinkamas. Viena iš universaliojo metodo objekto aptikimo algoritmo dalių naudoja turimą fono kadrą ar prieš tai buvusių kadrus, esamo kadro pasikeitimo zonai skaičiuoti. Toks kadro apdorojimas nereikalauja daug skaičiavimo resursų: vienas pilno kadro taškų kiekio ciklas atimties operacijai atlikti, tačiau vien šios atimties operacijos nepakanka, norint aprašyti pasikeitusias kadro sritis, kurios leistų sumažinti reikalingus skaičiavimo pajėgumus sekančiame numerio paieškos algoritme. Vis dėl to, papildomų operacijų naudojimas šioje vietoje teikia didesnę naudą nei laiko taupymas, todėl algoritmo panaudojimas jau šioje bendro algoritmo fazėje eliminuoja didelę dalį potencialių, numerio plokštelei nepriklausančių, taškų, taip padidinant sistemos patikimumo indeksą.

Po pradinio objekto aptikimo, bet kurio tipo sistemose būtinas pradinis kadro valymas, todėl šioje vietoje jokių skaičiavimo resursų sutaupyti nepavyks: histogramos išlyginimas bei kadro vertimas į pilkumo pustonijų skalę – būtini. Vis dėl to, šioje vietoje galimas kitoks – daugiau integruotas kadro apdorojimas, tiek kadro vertimą į pilkumo skalę, tiek histogramos išlyginimą realizuojant vienoje funkcijoje ir taip eliminuojant vieną papildomą ciklą viso kadro reikšmėms įvertinti. Laimėjimas šioje vietoje nedidelis, resursų ar laiko prasme, tačiau tokių funkcijų per visą algoritmą yra ne viena, o realaus laiko sistemose skaičiuojama kiekviena papildoma instrukcija, todėl pakeitimas – būtinas. Vertikalių ir horizontalių linijų koreliacijos koeficientų radimui, lygiai taip pat galima naudoti tik vieną bendrą funkciją, taip eliminuojant dar vieną pakartotiną skaičiavimą ar vieno iš koreliacijos koeficientų žemėlapiu perstūmimą. Dėl koreliacijos algoritmui reikalingų skaičiavimo resursų, apjungtas koreliacijos koeficientų skaičiavimas iš karto dviems skirtingiems žemėlapiams, leidžia sumažinti būtiną skaičiavimo resursą, dėl intensyvių aritmetikos operacijų sumažinimo kiekvienam kadro bei etaloninio kadro taškui. Anksčiau minėta, kad koreliacijos koeficientų sekos paieška, numerio rėmui patikrinti, nebuvo tokia sėkminga, kaip manyta, tačiau dėl to, kad realaus laiko reakcijos reikalaujančioms sistemoms renkant galimą algoritmą, parenkamas algoritmas mažesniu apdorojimo laiku, todėl ir šiuo atveju, vietoje visų įmanomų numerio rėmo kampų taškų tikrinimo, reikalingas paprastesnio ir greitesnio algoritmo naudojimas. Tam puikiai tinka jau anksčiau aprašytas, numerio rėmo priešingų kraštų paieška, ieškanti pakankamų



koreliacijos koeficiento pasikeitimų tiek į dešinę, tiek žemyn nuo tikrinamo potencialaus numerio rėmo taško. Kaip jau anksčiau minėta, šis algoritmas nebenaudojamas dėl nepakankamo atsparumo triukšmui, tačiau realaus laiko reakcijos reikalaujančiose sistemose, šio algoritmo panaudojimas leidžia sumažinti sistemai būtinus skaičiavimo pajėgumus. Kaip alternatyvą šiam sprendimui, galima naudoti kelias lygiagrečiai dirbančias FPGA SoC sistemas, su srautiniu duomenų apdorojimo konvejeriu, paraleliam taškų tikrinimui, tačiau tokiu atveju kyla galutinės sistemos aparatinės įrangos kaina bei energetinės sąnaudos. Be to, lygiagrečiam galimų numerio kandidatų taškų tikrinimui, būtina bendra išorinė atmintis, galimų numerio rėmo taškų koreliacijos koeficientų bei kadro bendram adresavimui, bei bendras atminties užraktų valdymas, kad viena sistema neperrašytų kitos sistemos duomenų. Išorinė atmintis yra adresuojama lėčiau nei vidinė, tačiau ir bendrai adresuojama atmintis, priklausomai nuo atminties tipo bei konstrukcijos ar jungimo schemos, lygiagrečiai vykstančius skaičiavimus sulėtina – atmintį naudoti vienu metu gali tik viena sistema, tačiau duomenys reikalingi visoms sistemoms vienu metu, dėl nuo kitų sistemų nepriklausomų lygiagrečių skaičiavimų. Vienas iš galimų sprendimo variantų – lygiagrečiams skaičiavimams paruoštų sistemų vidinės atminties užpildymas, pasinaudojant šių sistemų įvedimo bei išvedimo sąsajomis kaip I2C. Tokiu būdu, kadro galimų numerio rėmo taškų koreliacijos koeficientai perduodami į kiekvienos sistemos, nepriklausančios nuo kitų sistemų, vidinę atmintį ir taip visi lygiagretūs skaičiavimai atliekami be bendros adresų ar duomenų magistralės, neskaitant I2C sąsajos, kurią skaitymui galima naudoti vienu metu visoms sistemoms, pasinaudojant vienu bendru, I2C protokole siunčiamu, įrenginio numeriu, o sąsajos greitis priklauso tik nuo naudojamo kvarco dažnio. Paspartinus galimų numerio rėmo taškų skaičiavimus, lieka tik numerio atpažinimo algoritmas, kurį taip pat galima atlikti lygiagrečiai, rezultatus gražinant į pagrindinį sistemos modulį. Įgyvendinus tokią schemą, visos sistemos darbas organizuojamas paraleliai, todėl apdorojimo greitis priklauso tik nuo naudojamo kvarco dažnio ir lygiagrečiai dirbančių sistemų kiekio, valdomų pagrindinės sistemos ar plokštės lusto.

### ***Algoritmo adaptavimas pravažiavimo kontrolės sistemoms***

Pagrindinis šių sistemų reikalavimas – aukštas patikimumas bei toleruotinas reakcijos laikas. Šių sistemų privalumas, lyginant su universaliuoju modeliu, žinomos aplinkos sąlygos: apšvietimas, kadro kokybė, numerio dydis ir pan. Dėl sistemos atliekamos funkcijos, pačioje

algoritmo pradžioje galima laikyti, kad kadro apšvietimas bus, jei ne nekintantis, kintantis nedidelėse ribose. Be to, iš anksto žinoma potenciali numerio vieta ir numerio dydis. Dėl to, kad apšvietimo problema eliminuojama aparatinės įrangos bei aplinkos sąlygų, galima atsisakyti gama korekcijos algoritmo, taip sumažinant tiek programinę algoritmo bazę, tiek reikalingus skaičiavimo pajėgumus. Be to, dėl tų pačių aplinkos sąlygų, galima daryti prielaidą, kad visi autotransporto nuotraukų kadrai bus aukštos kokybės, o numerio plokštelė išsiskirianti kontrastu iš aplinkinio vaizdo, dėl numerio plokštelėms naudojamų dažų bei kameros apšvietimo. Tokiu būdu, numerio rėmo paieškos algoritmą galima pertvarkyti ne vertikalių bei horizontalių linijų paieškai, o kontrasto gradiento pokyčiams aptikti. Be to, šiam pakeitimui įgyvendinti, didelių algoritmo pertvarkymų nereikia: užtenka pakeisti koreliacijai naudojamo atraminio signalo pobūdį, kai kadras interpretuojamas kaip baigtinio ilgio bei begalinio spektro signalas, bei algoritmo eigos eiliškumą. Šiu atveju, kadro koreliacijai skaičiuoti, būtinas pradinio kadro vertimas skirtumine forma vietoje koreliacijos koeficientų vertimo skirtumine forma arba pakeisti koreliacijai skaičiuoti naudojamo atraminio vaizdo aprašymą. Bet kuriuo atveju, koreliacija vistiek skaičiuojama numerio plokštelės kraštui, bei plokštelės ir simbolių sąlyčio taškams, todėl šioje realizacijoje atsiranda papildoma problema: koreliacija išryškins ne vien numerio rėmus bei pašalinius taškus, bet ir tos pačios numerio plokštelės simbolius, kaip galimus numerio rėmo kraštus. Šia savybe galima pasinaudoti vėlesniame galimų taškų filtravime. Kaip alternatyva šiam algoritmo pakeitimui, galimas kiek kitoks koreliacijos skaičiavimas, pirma sudarant skirtuminį kadro reikšmių žemėlapi ir tik tada vykdant koreliacijos skaičiavimą ar skirtuminio žemėlapio didžiausių verčių šuolių paiešką. Bet kuriuo atveju, rezultatas gaunamas toks pat – skiriasi tik pradinio filtravimo algoritmo skaičiavimo intensyvumas bei forma. Tačiau taikant sistema pravažiavimo kontrolei geriau tinka didesnio tikslumo algoritmai, tačiau pakankamu skaičiavimo laiku. Be to, kuris algoritmas labiau tinkamas, priklauso nuo to, kokia kameros nuotraukų apdorojimo sistema naudojama: jei nuotraukos apdorojamos integruotose sistemos ar nedidelio skaičiavimo galingumo kompiuteriuose, tuomet numerio rėmo radimui tikrai netinka didelių skaičiavimo pajėgumų reikalaujantys algoritmai, tačiau jei nuotraukos apdorojamos didelius skaičiavimo pajėgumus turinčiose sistemose, kaip servisinis kompiuteris ar specializuota aparatinė įranga, šių algoritmų naudojimas leidžia sumažinti tikrinamų numerio rėmo galimų taškų kiekį.

Tarkime, kad sistemai, kaip autorių darbuose minima, naudojamas dedikuotas servisinis kompiuteris, todėl ir galimų numerio taškų paieškai bus naudojama koreliacijos funkcija. Turint koreliacijos koeficientų žemėlapi, kaip ir universaliojo metodo variantu,

būtina nustatyti ribinę vertę, pagal kuria šie koeficientai filtruojami. Dėl anksčiau paminėtų numerio plokštelės savybių bei dėl to gaunamų koreliacijos koeficientų staigių vertės pasikeitimų, galima daryti prielaidą, kad numerio rėmo zonos koreliacijos koeficientų pokyčiai bus didžiausi, todėl ir ribinė vertė, priešingai nei universaliojo modelio, su neaiškiais apšvietimo sąlygomis, turi būti nustatyti pagal maksimalią koreliacijos koeficientų gretimų verčių skirtumo vertę. Šioje vietoje atsiranda ribojimo dviprasmybė: ribinę vertę taip, kad liktų tik didžiausi skirtuminio koreliacijos koeficientų žemėlapiu taškai, numerio rėmas bus randamas, tačiau tik tuo atveju, jei nėra kito objekto su panašiomis gradiento pasikeitimo savybėmis. Tokiu atveju, ribinę vertę reikia šiek tiek sumažinti, įvedant papildomą koeficientą ir ribinę vertę nustatant skirtuminio koreliacijos koeficientų žemėlapiu maksimalios vertės 8/10 daliai, taip eliminuojant visus tuos pačius taškus, kaip ir be koeficiento, tačiau neišbrokuojant tikrų numerio rėmo taškų, kai egzistuoja kitas panašių savybių objektas ar kitas, skirtingo spalvinio kodo, numeris (vardinis numeris už stiklo bei valstybinis numeris). Dėl išskirtinių numerio plokštelės savybių, galimų numerio taškų kiekis priklauso tik nuo kitų objektų, panašiomis savybėmis, egzistavimo. Bet kuriuo atveju, bendras potencialių taškų skaičius, įvertinant sistemos darbo sąlygas, neturi viršyti universalios sistemos taškų skaičiaus.

Turint numerio rėmo potencialus taškus, būtinas šių taškų tikrinimas. Dėl koreliacijos koeficientų koncentravimosi pačiame numeryje savybės, galima daryti prielaidą, kad kiekvienam simboliui numeryje gaunama tiek pat koreliacijos koeficientų šuolinių pasikeitimų, kiek simbolis turi kontūro taškų šonuose, todėl numerio rėmui patikrinti nebūtina ieškoti kito numerio plokštelės krašto ar tikrinti visas įmanomas taškų poras. Šiuo atveju galima pasinaudoti literatūroje aprašytu blokiniu kadro tikrinimu: sudaryti kelis kadro žemėlapius, kuriuose sužymimi taškai, nuo kurių, konkretaus dydžio blokais, tikrinama ar yra tinkančių taškų. Toks kadro blokinių tikrinimas leistų įvertinti, ar tikrinamo taško bloke buvo kitų atitinkančių kriterijus taškų, taip įvertinant tinkamų taškų koncentracijos laipsnį. Vis dėl to šiam algoritmui įgyvendinti trukdo ribinės vertės parinkimas: jei ribinė vertė, taško tinkamumui įvertinti, parenkama pagal tikrinamo bloko vertes, kiekvienam blokui tikrai atsiras tinkamų kandidatų, tačiau ribines vertes parenkant iš viso kadro, galima nepriskaičiuoti tinkamų taškų, nustačius per aukštą ribinę vertę. Kai ribinė vertė per žema, lygiai taip pat gaunamas netinkamas rezultatas, tačiau šiuo atveju ieškant didžiausios taškų koncentracijos bloko, tinkamas blokas vistiek atrenkamas. Tokiu būdu atrenkami tik tie regionai, kuriuose galimai aptikta numerio plokštelė bei numerio plokštelės simbolių kraštai. Turint tinkamą

numerio regioną, dėl nežinomo numerio dydžio bei aspekto, kaip aprašoma ir literatūroje, reikalingas tinkamo bloko skaidymas mažesniais blokais, taip patikslinant numerio vietą. Kai ir pradinių blokų atveju, ribinę vertę atitinkantys taškai. Priešingai pradinių blokų tikrinimui, čia jau žinoma, kad bloko koncentracija tinkama, tačiau nežinomas bloko dydis bei forma, todėl ribinę vertę galima nustatyti iš atitikusio bloko verčių. Be to, skaidant pradinį bloką mažesniais blokais, galima sąlyga, kai tikras numerio regionas patenka už atitikusio bloko ribų. Tokiu atveju, skaidant smulkesniais blokais būtina praplėsti pradinio bloko ribas. Tokiu būdu, net jei tikrojo numerio regionas patenka už pradinio bloko ribų, skaidant mažesniais blokais patikrinami ir šie regionai. Be to, atitikę mažesni blokai, savo tinkamų taškų koncentracija, apibrėžia tinkamą regioną, todėl gaunamas bendras, didesnio skaidymo, blokas, perstumtas į tikrojo numerio poziciją, pirminio bloko atžvilgiu. Tokiu būdu, iš bendro kadro išfiltruojami tik galimi numerio regionai, o iš šių didesnio tikslumo galimi numerio regionai. Priklausomai nuo to, koku tikslumu reikia aptikti numerio regioną, galimas tolimesnis blokų skaidymas smulkesniais blokais, tačiau tokiu būdu eikvojami sistemos skaičiavimo resursai. Net tuo atveju, kai sistemoje kadra apdoroja servisinis kompiuteris, paprastesnių algoritmų naudojimas leidžia rezervuoti turimus skaičiavimo resursus programinės įrangos tobulinimams, išvengiant apdorojančios aparatinės įrangos atnaujinimo, todėl šiuo atveju tolimesnis blokų skaidymas nevykdomas. Šiam tikslui, kaip ir universaliojo metodo atveju, puikiai tinka kadro skaidymas projekcijomis. Ribinės vertės parenkamos pagal universaliojo metodo ribines vertes. Kaip ir universaliojo metodo atveju, būtina patikrinti galimo numerio regiono atitikimą užsibrėžtiems reikalavimams: numerio plokštelėje turi būti bent 5 ir ne daugiau 10 simbolių, be to, turi tikti numerio aspektas. Jei numerio regionas netinkamas, numerio regiono paieškos algoritmą būtina kartoti, tačiau šįkart su mažesnėmis ribinėmis vertėmis. Vis dėl to, netinkama numerio plokštelė gali reikšti ir klaidingą transporto priemonės užfiksavimą sistemos davikliu, todėl šioje numerio paieška gali būti nutraukta be papildomos, detalesnės numerio paieškos. Turint numerio plokštelės regioną, numerio plokštelę, kaip ir universaliojo metodo atveju, būtina skaidyti simbolių eilutėmis, o šias simboliais. Ši algoritmo dalis jau įgyvendinta universaliojo metodo atveju, todėl sistemos šioje vietoje nereikalingas. Išskaidyti simboliai, kaip ir universaliojo metodo atveju, atpažįstami koreliacijos algoritmu, kurių rezultatas apdorojamas papildomos algoritmo dalies. Toks algoritmo realizavimas leidžia atskirti pagrindinę numerio paieškos bei atpažinimo sistemą nuo numerio patikros sistemos. Tokiu būdu, pasikeitus sistemos rezultatus apdorojančiai sistemai, užtenka pakeisti tik šią sistemos dalį realizuojantį priedą.

## Rezultatai

Algoritmo realizacija sėkmingai įgyvendinta, numerių atpažinimo bandymo rezultatai pateikti pirmoje lentelėje. Tikėtina, kad dėl programinės realizacijos klaidų, naudojamos koreliacijos funkcijos ar per aukštų sistemai keliamų reikalavimų, numerio vietos aptikimo algoritmo patikimumas siekia tik 3-5%. Visiems bandymams naudotos 320x480 taškų dydžio bei 24 bitų spalvų gylio nuotraukos, gautos fotografuojant miesto srautą judančiame automobilyje. Pagal 30 paveiksle pateiktą grafiką, kai vidutinis kadro intensyvumas ~120, numerio atpažinimo procesui šviesumo įtaka sumažinta, o sistemos darbinis šviesumas, išreikštas vidutiniu kadro intensyvumu, galimas tarp 60 ir 220, kai skalė tarp 0 ir 255.

Kelių literatūros ir siūlomo algoritmo, atpažinimo tikslumo palyginimas:

Lentelė 1

<b>Algoritmas</b>	<b>Tikslumas</b>
[1] Design and Implementation of License Plate Recognition System	98,00 %
[2] A Macao licence plate recognition system	93,20 %
[4] An Algorithm for License Plate Recognition System for Non-stop Toll Stations	89,70 %
[20] Research and Realization of Improved Pattern Matching in License Plate Recognition	98,00 %
Siūlomas algoritmas	94,81 %

## Išvados

- Padaryta egzistuojančių autotransporto priemonių registracijos numerių atpažinimo būdų analizė. Pagal gautas išvadas ir rekomendacijas sudarytas naujas būdas.
- Sudarytas būdas pagrįstas koreliacijos funkcijomis geba tiksliai atpažinti simbolius plačiame šviesumo diapazone: darbo sąlygas tenkina visi kadrai, kurių vidutinis intensyvumas tarp 60 ir 220. Švaraus kadro simbolių (simbolių atitikimo tikslumas siekia 40%) atpažinimo tikslumas siekia 92-95%, o bent 1% simbolių koreliacijos atitikimą turinčių simbolių atpažinimo tikslumas ~30%
- Sistemos rezultatai išvedami kartu su simbolių patikimumu, taip leidžiant aptarnaujančiam operatoriui spręsti ar pasitikėti sistemos rezultatais.
- Pritaikymas realaus laiko sistemoms galimas, tačiau nekeičiant programos turi būti didinamas įrangos našumas, taip didinant jos kainą. Dažniausiai pravažiavimo kontrolės sistemoms galimas supaprastintas numerio vietos radimo algoritmas, dėl žinomų numerio savybių ir sistemos aplinkos sąlygų, todėl realiose situacijose realaus laiko sistemose galima pristinti algoritmą atsisakant kai kurių apdorojimo veiksmų: tokiu kaip koreliacijos naudojimą brūkšnių paieškai, vietoje to ieškant didžiausio gradiento pasikeitimo regionus (toks algoritmas siūlytas ir literatūroje).
- Pasiūlytas būdas buvo realizuotas C++ klasėse bei pritaikytas masiškai paraleliems skaičiavimams ir tinka bet kuriai aparatinei ar operacinės sistemos platformai, palaikomai TrollTech / QT Software (Nokia Corporation) QT v4 bibliotekų, todėl realizacijos pritaikymas įgyvendinimui mikro-valdikliuose ypatingai paprastas.

## Literatūra

1. Hongwan Yang, Liancheng Xu, Lei Shi, Design and Implementation of License Plate Recognition System. Jinan, P.R.China, 2007
2. Cheokman Wu, Lei Chan On, Chan Hon Weng, Tong Sio Kuan, Kengchung Ng, A Macao license plate recognition sysmte [online]. Macao, China [2005]
3. M J. Ahmed, M Sarfaz, A. Zidouri, K G. Al-Khatib, License plate recognition system. Dhahran-31261, Saudi Arabia. 2003
4. Bo Li, Zhi-Yuan Zeng , Jian-Zhong Zhou, An Algorithm for License Plate Recognition System for Non-stop Toll Stations. Wuhan, China. 2008
5. Marwan D. Saleh1, H. Mellah1, Ahmed Mueen2. N. D. Salih1, An Efficient Method for Vehicle License Plate Extraction. Malaysia. 2008
6. Vladimir Shapiro, Georgi Gluhchev, Multinational License Plate Recognition System: Segmentation and Classification: 17th International Conference on Pattern Recognition (ICPR'04). IEEE, 2004
7. Pruegsa Duangphasuk, Arit Thammano, Thai Vehicle License Plate Recognition Using the Hierarchical Cross-correlation ARTMAP: 3rd International IEEE Conference Intelligent Systems. IEEE, 2006 09
8. Christos-Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulos, Ioannis D. Psoroulas, Vassili Loumos, Eleftherios Kayafas, License Plate Recognition From Still Images and Video Sequences: A Survey: IEEE Transactions on intelligent transportation systems, VOL. 9, NO. 3, 2008 09
9. Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, and Sei-Wan Chen , Automatic License Plate Recognition. 2004
10. Zhang Jian, Fan Xiaoping, Huang Cailun, Research on Characters Segmentation and Characters Recognition in Intelligent License Plate Recognition System: 25th Chinese Control Conference. Harbin, Heilongjiang, 2006
11. Nikolaos Bellas, Sek M. Chai, Malcolm Dwyer, Dan Linzmeier, FPGA implementation of a license plate recognition SoC using automatically generated streaming accelerators. 2006
12. Song Huansheng, Wang Guoqiang, The high performance car license plate recognition system and its core techniques. 2005
13. Xiaojun Chi, Junyu Dong Aihua Liu, Huiyu Zhou, A Simple Method for Chinese License Plate Recognition Based on Support Vector Machine. UK, China. 2006
14. Vivek Seetharaman, Vidhya Sudhan L Nathan, Sathyakala A., Sunder P., License Plate Recognition System Using Hybrid Neural Networks. India, 2004
15. Choudhury A. Rahman, Wael Badawy, Ahmad Radmanesh, A Real Time Vehicle's License Plate Recognition System: IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03). Canada. 2003.
16. Dai Yan, Ma Hongqing, Liu Jilin and Li Langang, A High Performance License Plate Recognition System Based On The Web Technique: IEEE Intelligent Transportation Systems Conference, Oakland (CA), USA, 2001
17. Mei Yu, Yong Deak Kim, An Approach to Korean License Plate Recognition Based on Vertical Edge Matching. Korea, 2000
18. T. Naito, T. Tsukada, S. Yamamoto, License plate recognition method for inclined plates outdoors. Japan
19. J.A.G. Nijhuis, M.H. ter Brugge, K.A. Helmholt, J.P.W. Pluim, L. Spaanenburg, R.S. Venema, M.A. Westenberg, Car License Plate Recognition with Neural Networks and Fuzzy Logic. Netherlands
20. Ping Wang, Wei Zhang, Research and Realization of Improved Pattern Matching in License Plate Recognition: International Symposium on Intelligent Information Technology Application Workshops. China, 2008
21. Hsiao-Yun Tseng, Ching-Hao Lai, Shyr-Shen Yu, An effective license-plate detection method for overexposure and complex vehicle images: International Conference on Convergence and Hybrid Information Technology. Taiwan, 2008
22. Marwan D. Saleh, Herminiawati, H. Mellah, N. D. Salih , omparison of Contrast Enhancement Techniques in License Plate Recognition. Malaysia, 2008
23. D. W. Tindall, Deployment of automatic license plate recognition systems in multinational environments: European Conference on Security and Detection, IEEE, U.K., 1997
24. Bo Li, Zhiyuan Zeng, Jianzhong Zhou, Mu Zhou, An Adaptive Algorithm for License Plate Orientation and Character Segmentation: IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application. China, 2008