

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Baigiamasis magistro darbas

Interaktyvus transporto numerio atpažinimas

Atliko: II mag. kurso, 1 grupės studentas
Vitalij Tučkovskij

Darbo vadovas:
Algirdas Bastys, docentas, mokslų daktaras

Vilnius 2006

Turinys

Anotacija.....	3
Summary.....	4
Įvadas.....	6
Darbo tikslas.....	6
Pradinių duomenų formatas.....	7
Taikomų metodų ir algoritmų aprašymas.....	9
Stebėjimo zonos išskyrimas.....	9
Fono eliminavimas.....	10
Gauso filtras.....	11
Sobel filtras.....	13
Maxmin filtras.....	14
Numerio išskyrimas.....	15
MIDP.....	17
Display ir Displayable.....	18
MIDP API.....	19
Eksperimento aprašymas.....	21
Išvados.....	23
Literatūra.....	25

Vilniaus universitetas
Matematikos ir informatikos fakultetas
Kompiuterijos katedra

Data 2006 05 25

Magistrinis darbas (tezės)
Transporto numerio atpažinimas, web kameros, eismo filtravimo eksperimentas.
V. Tučkovskij

Kalba



lietuvių
užsienio

Baigiamajame magistro darbe “Interaktyvus transporto numerio atpažinimas” nagrinėjami bei pritaikyti metodai ir algoritmai. Taip pat atlikta metodų bei filtrų analizė. Sukurta programa duomenų įvedimui į kompiuterį iš vaizdo kameros, video failo arba interneto. Atliktas eismo filtravimo eksperimentas, gauti ir pateikiami apdorojimo rezultatai vartotojui į mobilųjį telefoną.

Reikšminiai žodžiai: Numerio atpažinimas, triukšmas, vaizdas, metodas, fonas, kameros, filtras, MIDP, mobilus telefonas.

Summary

This paper presents an approach to license plate localization. A proposed method is designed to perform recognition of any kind of license plates under any environmental conditions. The main assumption of this method is the ability of recognition of all license plates which can be found in an individual picture. The algorithm takes a raster image as input, and yields the position of a plate in the image. After the position is determined, the algorithm can determine the locations of the license plate characters, which could be easily combined with an OCR algorithm to convert the license plate number into an ASCII string. User could view collected license plate localizations results with mobile phone, which has JAVA (MIDP 1.0) support and internet access. Future work could focus on determining the positions of characters more accurately. Determining the correct positions of all characters in a license plate would dramatically improve the results of the license plate recognition.

Išvadas

Mūsų gyvenamu laiku paprastas dirbantis pilietis, gyvenantis mieste arba kaime, gali leisti sau įsigyti mašiną. Tai veda prie padidėjusio automobilių judėjimo gatvėse, keliuose bei magistralėse. Atsižvelgiant į tai, atsiranda transporto eismo kontrolės bei analizavimo problemos. Didžiausiuose miestuose bandoma jas spręsti analizuojant mašinų srautų judėjimą, jo padidėjimą arba sumažėjimą.

Analizuodami mašinų srautą mes galime iš anksto nustatyti būsimą problemą bei bandyti rasti jos sprendimo metodus. Pavyzdžiui, mašinų skaičiaus padidėjimas kelio ruožuose gali mums pranešti, kad greitai laiku ten gali susidaryti kamščiai. Tokiu atveju reikia kuo greičiau rasti problemos sprendimo būdus, imtis tam tikrų veiksmų, kad išvengtumėm klaidų ir nelaimių.

Eismo srauto analizavimui geriausiai naudoti skaitmenines priemones, o ne analogines. Mes galime išskirti kaip pavyzdį kelis svarbiausius privalumus:

1. Saugumas. Naudojant skaitmeninę įrangą visi duomenys yra saugomi skaitmeniniame formate ir lengvai gali būti paimti bei perduoti į kitus analizavimo įrenginius.
2. Mobilumas. Šiuolaikinės technologijos leidžia be problemų įrengti nedaug vietos užimanti skaitmeninį sekimo įrenginį.
3. Vaizdo apdorojimo metodų gausa. Skaitmeninį vaizdą yra lengviau apdoroti bei gauti iš jo mus dominantį rezultatą.
4. Duomenys yra lengvai pasiekiami. Pavyzdžiui, turint kelis analizavimo taškus, visi duomenys gali būti perduodami į analizavimo centrą, kur jie bus analizuojami toliau.

Šiame darbe bus analizuojamas atvejis, kuriame svarbiausia nustatyti transporto priemonės numerį. Mūsų sistemos pagalba iš mums dominančio video vaizdo duomenų srauto yra apskaičiuojami atpažinti transporto numeriai naudojant filtrus. Duomenys galima gauti tokiais būdais: naudojant video kamerą, išsaugotas duomenų failas, arba video srautas iš interneto.

Darbo tikslas

1. Pritaikyti vaizdų apdorojimo metodus ir algoritmus, kurie gali būti taikomi numerio išskyrimui, atpažinimui.
2. Sukurti programą duomenų įvedimui į kompiuterį iš vaizdo kameros, video failo arba interneto.
3. Atlikti eismo filtravimo eksperimentą, gauti ir išanalizuoti apdorojimo rezultatus.
4. Statistinių duomenų pateikimas galutiniam vartotojui naudojant mobiliąs technologijas.

Pradinių duomenų formatas

Tokie standartai kaip MPEG-1 ir MPEG-2 buvo sėkmingai pritaikyti tokiose sferose, kaip telekomunikacija, transliacija ir t.t. Šių duomenų formatų naudojimas smarkiai plinta tarp vartotojų ir programinės įrangos kūrėjų. Kiti standartai, tokie kaip MPEG-4, H.26x ir Image stream(web kameros) bei kiti standartai, turi daug panašumų video ir audio kodavime. Taigi galimybės, kurios čia bus aprašytos, gali būti taikomos ir tiems standartams taip pat.

Video kompresijos technologija

Skaitmeninė video informacija turi būti suspausta tolimesniam saugojimui ir perdavimui. Video kompresija liečia daug tyrimu sričių, tokias kaip komunikacija, informacijos teorija, vaizdų apdorojimas ir taip toliau. Daug metų analizavimų šiuose srityse davė tokius formatus kaip MPEG-1, MPEG-2, H.26x ir kitus. Šitie standartai turi tokius metodus, kaip bloko pagrindų transformacija, prognozuojamas kodavimas, entropinis dekodavimas, judėjimo kompensacija bei kiti metodai. Svarbiausios yra bloko pagrindų transformacija ir judėjimo kompensacija. Bloko pagrindų kodavimo transformacija mažina erdvinį perteklių skaitmeniniame vaizde (arba skaitmeninėje nuotraukoje).

MPEG-1 ir MPEG-2

MPEG-1 ir MPEG-2 yra aprašyti tarptautiniame standarte ISO/IEC numeriais 11172 ir 13818. MPEG formatas turi keturis sluoksnius: grupė paveikslukų (GOP), paveikslukas, blokas, makroblokas (MB).

MPEG-1 standartas taikomas kodavimui be pasikartojimų video vaizduose su bitreitu iki 1.5 Mb/s, vaizdams, turintiems dydžius iki 352x240 pikselių.

MPEG-2 standartas taikomas kodavimui didesnėms rezoliucijoms (720x480) su gana dideliu bitreitu nuo 4 iki 9Mb/s. MPEG-2 sėkmingai naudojamas DVD ir yra taikomas skaitmeninėms teletransliacijoms. MPEG-2 standartas turi keletą naujų savybių:

- 4:2:2 ir 4:4:4 makroblokų formatą
- Daugiau lankstus ir sudėtingesnis judėjimo kompensavimas
- Kadro ir lauko paveiksluko struktūra
- Naujas skanavimas DCT(diskreti kosinusų transformacija) koeficientams
- Kiti patobulinimai kode

Web kameros

Web kameras galima sugrupuoti į 3 grupes pagal funkcionalumą:

- Video streaming – gali saugoti ir arba pateikti maksimum iki 30 kadrų per sekunde. Vaizdas pateikimas video formatų.
- Image streaming – pateikia maksimum 5 kadrus per sekunde. Vaizdas pateikiamas keičiančiais paveiksliukais. Programa turi kreiptis į šaltinį per tam tikro laiko intervalus arba laukti atnaujinimo iš šaltinio.
- FTP upload – iki 2 kadrų per sekunde. Paveiksliukai siunčiami į FTP.

Privalumai ir trukumai web kamerų:

- Geram vaizdo apdorojimui reikia spartaus interneto.
- Maža rezoliucija gauto vaizdo.
- Norimo laiko intervalo duomenų prieinamumas(Image streaming bei FTP upload atvėjų).

Taikomų metodų ir algoritmų aprašymas

Stebėjimo zonos išskyrimas

Šis metodas taikomas kadro informacinei daliai išskirti.



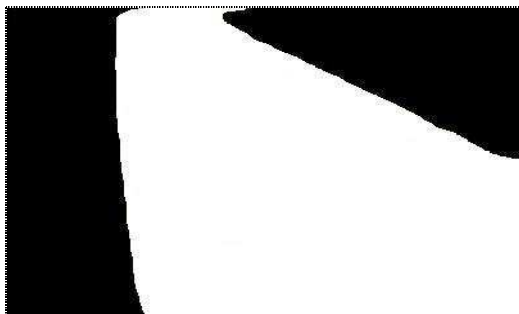
1 pav. Pradinis kadras.

Metodo taikymas supaprastina automobilių vaizdų išskyrimą iš kadru. Be to, iš kadro eliminavus vaizdo informacijos dalį, galima smarkiai padidinti vaizdo apdorojimo greitį.

Maskavimo algoritmas aprašomas formule:

$$N(p) = M(p) \times V(p),$$

kur $M(p)$ yra vaizdo taško p intensyvumo reikšmė pradiniame kadre; $V(p)$ kaukės reikšmė taške p (2 pav.). $V(p) = 0$, jeigu vaizdo taškas yra eliminuojamas ir $V(p) = 1$ priešingu atveju; $N(p)$ – šio taško intensyvumas maskuotame vaizdo kadre (3 pav.).



2 pav. Kadro kaukė.



3 pav. Kadras po maskavimo.

Fono eliminavimas

Šis algoritmas iš stebėjimo zonos šalina visus nejudamus objektus, palikdamas tik automobilių vaizdus bei kai kurias detales, kurios keičiasi kadru peržiūros metu. Šios detalės atsiranda dėl įvairių triukšmų, o taip pat dėl tokių gamtos reiškinių, kaip saulės blizgesiai, šviesos šešėliai, debesiai, lietus, sniegas, kameros drebėjimas ir t.t. Taip pat turi būti atsižvelgiama į dienos ir metų laiką, orą bei kitokius faktorius. Visa tai reiškia, kad realiose sąlygose fono pakeitimai turi būti nuolat kontroliuojami, o jis pats, reikalui esant, laikas nuo laiko turi būti koreguojamas.

Fonas $F(p)$ skaičiuojamas kiekvienam vaizdo taškui p kaip n fono kadru vidurkis. Jį skaičiuojame atskirai kiekvienai RGB reikšmei:

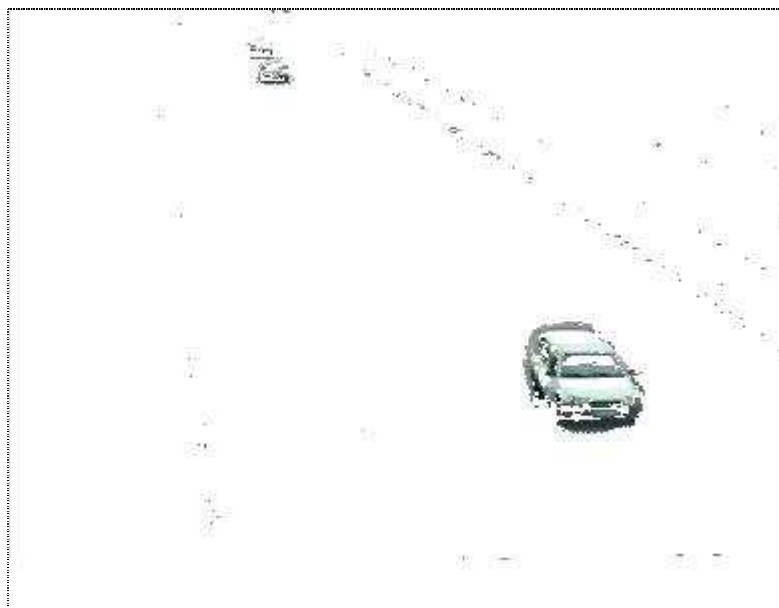
$$F(p) = [I_B(1, p) + \dots + I_B(k, p) + \dots + I_B(n, p)] / n ,$$

kur $I_B(k, p)$ yra taško p pikselio reikšmė k -me kadre. Fono šalinimas iš vaizdo $I(k, p)$ generuoja RGB vaizdą, kaip Euklido atstumą tarp $I(k, p)$ ir $F(p)$:

$$D(k, p) = [\{ I_R(k, p) - F_R(p) \}^2 + \{ I_Z(k, p) - F_Z(p) \}^2 + \{ I_M(k, p) - F_M(p) \}^2]^{1/2} ,$$

kur indeksai R, Ž, M atitinka raudonai, žaliai, ir mėlynai spalvai.

Šis rezultatas pavaizduotas žemiau (4 pav.).



4 pav. Kadro vaizdas su eliminuotu fonu.

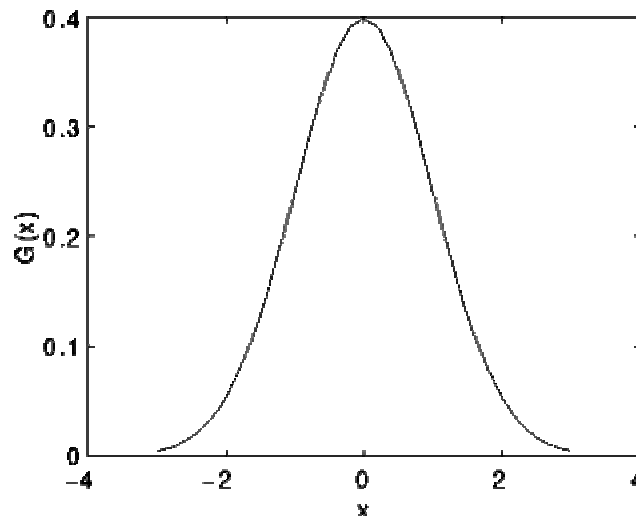
Gauso filtras

Gauso glodinimo operatorius yra dvimatis, kuris naudojamas vaizdams glodinti, t.y. šalinti detalumą ir nereikalinga triukšmą. Kai kuria prasme filtras panašus į vidutinį filtrą, bet šitas filtras naudoja kitokias kaukės. Šitos kaukės turi šiokiu tokiu ypatumu kurie bus aprašyti žemiau.

Gauso vienmatis pasiskirstymas turi pavidalą:

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

kur koeficientas $\sigma=1$. Taip pat mes turime omenyje, kad centras taške $x=0$. Gauso filtro pasiskirstymas parodytas 5 pav.

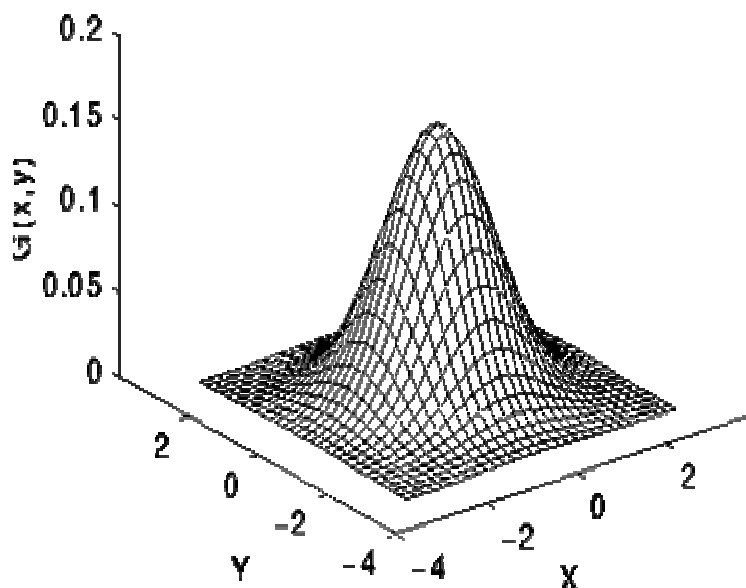


5 pav. Vienmatis Gauso funkcijos pasiskirstymas su centru 0 ir $\sigma=1$.

Dvimatis izotropinis (simetriškas) Gauso filtras turi pavidalą:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Reikšmių pasiskirstymas parodytas 6 pav.



6 pav. Dvimatės Gauso funkcijos reikšmių pasiskirstymas su centru (0,0) ir koeficientu $\sigma=1$.

Gauso glodinimo idėja yra tokia, naudoti šią dvimatį pasiskirstymą kaip „taškų pasiskirstymo“ funkciją, tai pasiekama konvoliucijos būdu. Kadangi vaizde duomenys - diskrečių pikselių rinkinys, mums reikia padaryti diskrečią aproksimaciją Gauso funkcijai prieš pritaikant konvoliuciją. Teorijoje Gauso pasiskirstymas nenulinis visur, kuris reikalautų didžiulės kaukės. 7 pav. rodo priimtina sveikų skaičių kaukę, kuri aproksimuoja Gauso funkciją su koeficientu $\sigma=1$.

	1	4	7	4	1
	4	16	26	16	4
$\frac{1}{273}$	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

7 pav. Diskreti aproksimacija Gauso funkcijos su $\sigma=1$.

Kai tinkama kaukė apskaičiuota, Gauso glodinimas gali būti pritaikytas naudojant standartinius konvoliucijos metodus. Konvoliucija gali būti greitai pritaikyta, kadangi dvimatė izotropinė Gauso lygtis, minėta aukščiau, gali būti padalinta į X ir Y komponentės. 8 pav. rodo vienmatės X

komponentės kaukė, kuri bus panaudota gauti kaukė parodyta 13 pav. (po dalijimo iš 273, apvalinant ir mažinant vieną stulpelį pikselių aplink sienos, nes jie dažniausiai turi reikšmes lygias nuliams. Šitos operacijos paverčia 7x7 kaukę į 5x5 kaukę, pavaizduotą aukščiau.) Y komponentė tokia pat, tik vertikali.

.006	.061	.242	.383	.242	.061	.006
------	------	------	------	------	------	------

8 pav. Vieną iš poros vienmačių kaukių.

Sobel filtras

Sobel operatorius apskaičiuoja paveiksluko gradientą kiekviename taške, keturiomis kryptimis. Galutiniame rezultate matosi kaip staigiai arba tolygiai paveikslukas keičiasi šitame taške, taip pat galima nustatyti ar šito paveiksluko dalis yra kraštas, bei nustatyti kaip šitas kraštas orientuotas.

Matematiškai šitas operatorius naudoja 3x3 kaukę aproksimuoti vaizdo gradientą – vertikalią ir horizontalią kryptimi. Jeigu mes pažymėsime pradinį vaizdą A, G_x ir G_y dalines išvestines kiekviename paveiksluko taške - galutinis rezultatas G atrodys taip:

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

9 pav. Sobel operatorius

Galutine operatoriaus reikšme neturi išsiplėsti už RGB ribų, t.y jeigu gaunamas neigiamas skaičius jis apnulinamas, jeigu virš 255 jam priskiriama reikšme 255.

Žinodami vertikalią ir horizontalią kryptimi vaizdo gradientą mes galime rasti gradiento orientaciją:

$$\Theta = \arctan \left(\frac{G_y}{G_x} \right)$$

10 pav. Gradiento orientacija

Iš operatoriaus trukumu galima pažymėti tokius:

1. Sobel operatorius apskaičiuoja apytiksliai paveiksluko gradientą, bet to tikslumo pakanka realiuose taikymuose.
2. Operatorius atsižvelgia į intensyvumo reikšmes tik 3x3 regione aplink kiekvieną tašką.
3. Operatorius naudoja tik sveikus skaičius kaukeje.

Maxmin filtras

Maxmin operatorius pirmoje stadijoje randa minimalias ir maksimalias paveiksluko reikšmes taške keturiomis kryptimis. Paskutiniame stadijoje operatorius randa maksimalias reikšmes iš jau apskaičiuotų minimalių ir maksimalių reikšmių. Galiausiai galima dar pašalinti atsitiktinį triukšmą, atmetant „mažas“ reikšmes, ty. palikti daugiau taškelių turinčių juodos spalvos komponentės. Galutiniame rezultate matosi paryškinti konturai pradinio vaizdo.

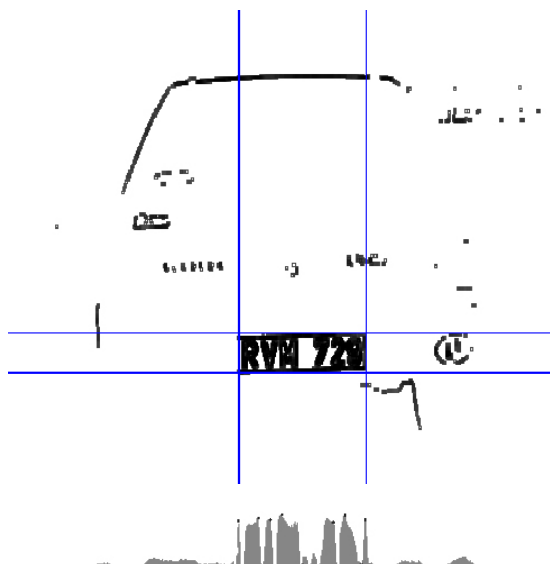


11 pav. Iš kairės pusės paveikslukas apdorotas su Sobel filtru, iš dešinės pusės Sobel + Maxmin filtrai.

Numerio išskyrimas

Dauguma algoritmų pagrįsti tokiu faktu: jie ieško objektų, panašių į simbolius. Jeigu buvo rasta minimum 3 objektai vienoje horizontalioje pozicijoje, algoritmas ieško toliau aplinkui panašių objektų. Šitas metodas nėra efektingas laiko bei resursų atžvilgiu. Algoritmui reikia atsižvelgti į visus imanomus variantus. Tai brangiai kainuoja.

Kad pagerinti algoritmo efektyvumą, buvo pasiūlytas naujas metodas. Šitas metodas pagrįstas tokiu faktu, kad vertikalios ir horizontalios linijos ant numerio turi aiškiają signatūrą, kurią galima atskirti nuo kitų linijų kadre (vaizde) arba bent apytiksliai rasti vietą, kur toliau ieškoti. Pavyzdys rodo tokias linijas.



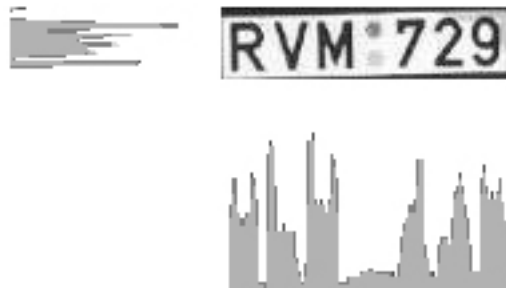
12 pav. Vertikalios ir horizontalios numerio linijos.

Numerio signatūra susideda iš „kalnelių“, kurie turi maždaug lygius intervalus. Šitą algoritmo stadiją galima dar potobulinti:

1. Galima ne kiekvieną liniją tikrinti, tarkim, kas penktą.
2. Sukaupiti tam tikrą duomenų bazę, į kurią atsižvelgs programa, t.y kokios signatūros gali būti.

Kai horizontalios ir vertikalios linijos, kurios kerta numerį, buvo rastos, šita informacija panaudojama numerio radimui. Kitas žingsnis – rasti tikslesnę numerio vietą. Tai daroma

naudojant vertikalias ir horizontalias vaizdo projekcijas. Mastelio dydis parenkamas dinamiškai, priklausomai nuo maksimumo ir minimumo lygio.



13 pav. Vertikalios ir horizontalios atpažinto numerio projekcijos

Numerio segmentavimas daromas pagal Lu [Lu95] pasiūlyta metodą, taip vadinamąjį „peak-to-valley“. Šitas metodas ieško „kalnelių“ vertikalioje projekcijoje. Segmentavimo rezultatai pateikti žemiau.



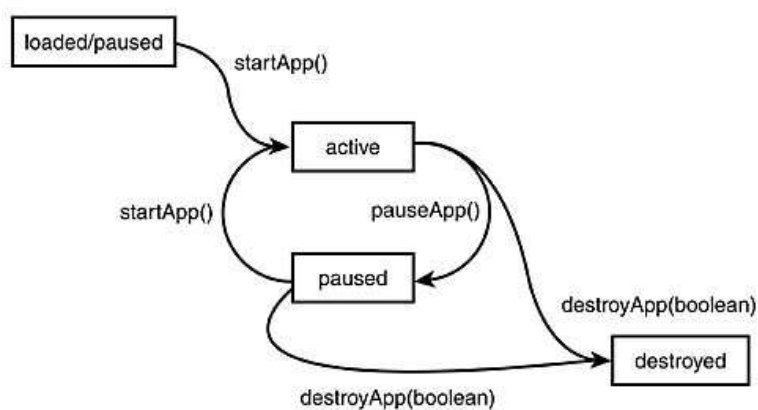
14 pav. Segmentuotas numeris

MIDP

MIDP - Mobile Information Device Profile [Web05]. Tai specifikacija mobiliesiems telefonams arba PDA (skaitmeninis asmeninis asistentas) įrenginiams. MIDP specifikacijas ruošia ir išleidžia Java Community Process (www.jcp.org) organizacija. Šiuo metu išleistos dvi galutinės specifikacijos JSR 37 (MIDP 1.0) ir JSR 118 (MIDP 2.0) versijoms. Nuo 2006 metų pradėta ruošti MIDP 3.0 specifikacija su numeriu JSR 271.

MIDP 1.0

Visos programos, kurių pagrindas yra MIDP, turi iškviešti specialią klasę MIDlet. MIDlet klasė atsakinga už programos gyvavimo ciklą. Ši klasė yra pakete `javax.microedition.midlet`. MIDletus galima palyginti su J2SE apletais, išskyrus tai, kad jų būseną mažiau priklausoma nuo ekrano (display) būsenos. MIDletas gali egzistuoti keturiuose būsenose: pakrautas, aktyvus, sustabdytas, panaikintas. 15 paveiksluke parodytas MIDleto gyvenimo ciklas. Kai MIDletas pakrautas įrenginyje ir iškvieštas konstruktorius – jis pakrautoje būsenoje. Tai gali būti bet kuriuo metu prieš programai iškviečiant metodą `startApp()`. Kai iškvieštas metodas `startApp()` MIDletas yra aktyvioje būsenoje, kol programa neiškvies `pauseApp()` arba `destroyApp()` metodų; `pauseApp()` metodas sustabdo programą, `destroyApp()` metodas panaikina MIDletą.



15 pav. MIDleto gyvavimo ciklas

Iškviečius `pauseApp()` metodą, programa turi sustabdyti animaciją ir atlaisvinti nereikalingus resursus. Šita elgsena panaikina resursų konfliktavimą su kitomis veikiančiomis programomis ir beprasmę baterijos naudojimą. `destroyApp()` metodas turi besąlyginį parametrą. Jeigu jam priskirta reikšmė `false`, MIDletas gali ir nepanaikinti programos mesdamas išimti

MIDletStateChangeException. MIDletai gali paprašyti pratęsti veikimą iškviečiant metodą *resumeRequest()*. Jeigu MIDletas pereina į sustabdytą būseną, jis gali perspėti programos menedžerį iškviesdamas metodą *notifypaused()*. Prieš baigiant darbą MIDletas gali iškviešti metodą *notifyDestroyed()*. Pastebėtina, kad *System.exit()* nepalaikoma MIDP specifikacijoje, programa rodys „išimti“ vietoj panaikinant programą.

Pastaba:

Kai kurie mobilūs telefonai tam tikrom aplinkybėm gali panaikinti MIDletą neiškviečiant metodo *destroyApp()*. Kaip pavyzdys būtų įeinantis skambutis arba kada baterijos išsekusios. Todėl visiškai patikėti galutine vartotojo įvesta arba pakeista informacija saugoti metodui *destroyApp()* negalima.

Display ir Displayable

MIDletai gali būti foninėmis arba interaktyviosiomis programomis. Display klasė ir visos kitos vartotojo aplinkos klasės yra pakete *javax.microedition.lcdui*. Display klasė turi metodą *setCurrent()*, kuris pakrauna MIDleto informaciją. Šitas metodas perspėja programos menedžerį, kad nori turimą Displayable objektą atvaizduoti. Skirtumas tarp Display ir Displayable tame, kad Display klasė reprezentuoja ekraną (*display*), o Displayable - tai kažkas, ką galima atvaizduoti ekrane. MIDletas gali iškviešti metodą *isShown()*, kad įsitikintų, ar tikrai tas turinys atvaizduotas ekrane.

HelloMidp Pavyzdys

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class HelloMidp extends MIDlet {
//Konstruktoriuje gaunamas Display.
    Display display;
    Form mainForm;
    public HelloMidp () {
        mainForm = new Form ("HelloMidp");
    }
}
```

```

//Kai MIDletas startuojamas pirmą kartą arba jo darbas pratęsimas po sustabdymo, programos
//mendžeris iškviečia metodą startApp()
public void startApp() {
    display = Displayable.getDisplay (this);
    display.setCurrent (mainForm);
}
//Kai programa sustabdoma, reikia atlaisvinti resursus. Šitas metodas privalomas
public void pauseApp() {
}
//DestroyApp() metodas privalomas, nors jis ir tuščias.
public void destroyApp(true) {
}
}

```

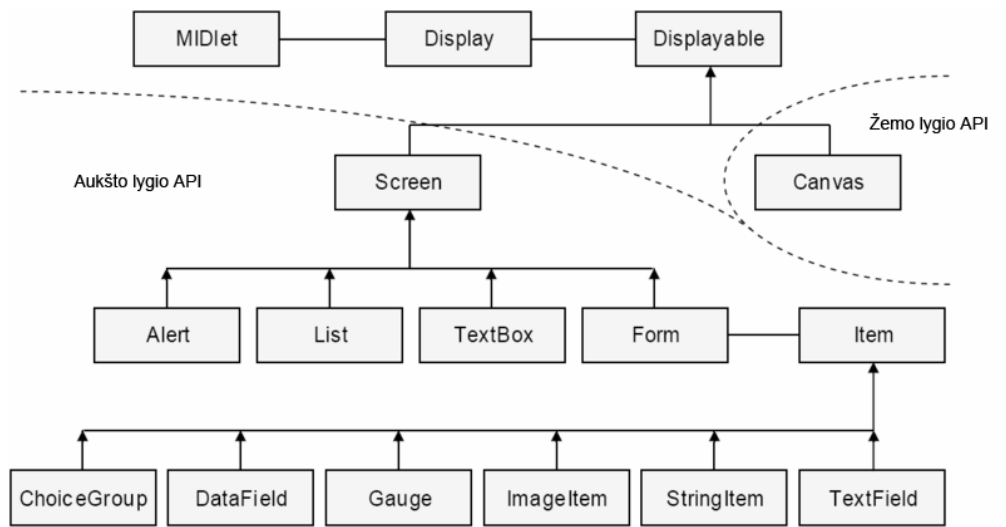
Pastaba

Šitame pavyzdyje nebuvo sukurta išėjimo komanda (mygtukas), kadangi mobilus telefonas pagal nutylėjimą sukuria išėjimo iš programos mygtuką.

MIDP API

MIDP vartotojo interfeiso API padalintas į aukšto ir žemo lygio API (žr. 16 pav.). Aukšto lygio API suteikia įvedimo elementus tokius kaip teksto įvedimo, pasirinkimo elementus. Jeigu lyginti MIDP vartotojo aplinką su J2SE AWT, MIDP aukšto lygio komponentai negali būti pozicionuojami. Yra tik dvi fiksuotos klasės: *Screen* ir *Item*.

Aukšto lygio *Screen* ir žemo lygio klasė *Canvas* turi bendrą pagrindinę klasę *Displayable*. Visos subklasės *Displayable* užpildo visą telefono ekraną. Subklasės *Displayable* *Display* objektas gali būti išvedamas į telefoną *setCurrent()* metodo pagalba.



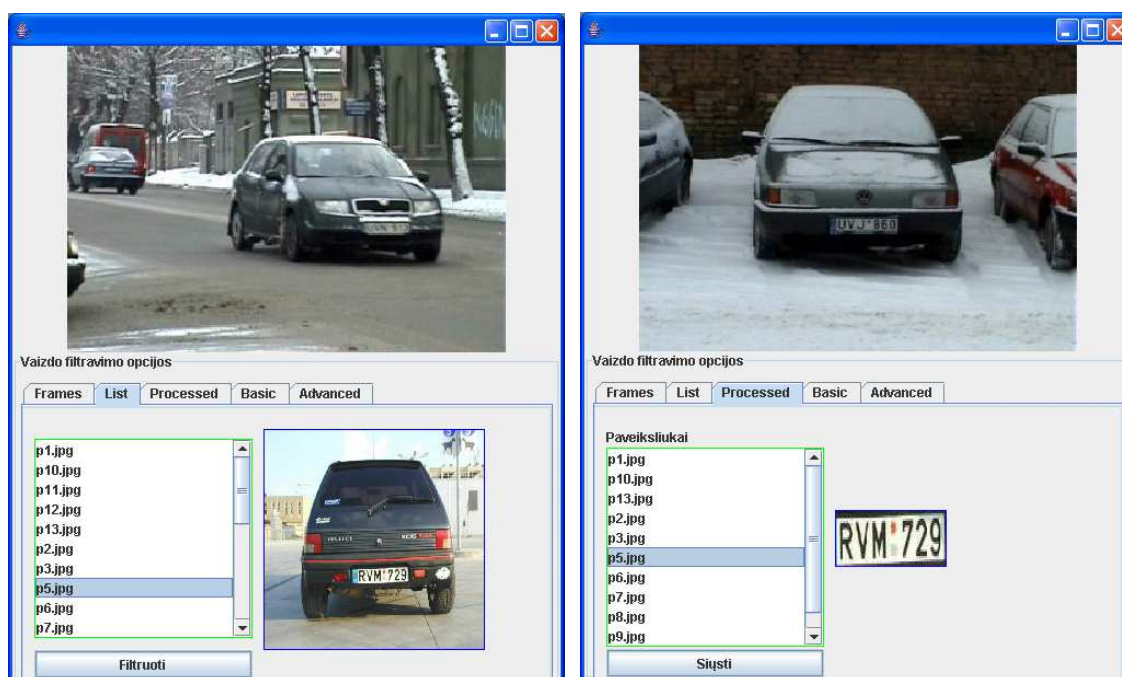
16 pav. MIDP API

Eksperimento aprašymas

Norint pritaikyti minėtus metodus bei taikymus buvo sukurtos programos. Jų pagalba buvo atliktas eksperimentas bandant atpažinti transporto numerius. Eksperimento realizavimas bei rezultatai pateikti žemiau.

Eksperimento realizavimas

Video duomenų apdorojimo programai realizuoti buvo panaudota JAVA kalba bei Java Media Framework praplėtimas (darbas su media duomenimis). Vartotojo interfeisas sukurtas naudojant SWING grafinį paketą (žr. 17 pav). Programai reikia paduoti video šaltinį – tai gali būti video failas arba web kamera. Programa „fotografuoja“ kadrus, išsaugo jas į atskirą direktoriją tolimesniam apdorojimui. Iš sukauptų kadrų programa bando atpažinti transporto numerius, išsaugoja atpažintus numerius nurodytoje direktorijoje. Galutiniame etape programa generuoja statistiką ir siunčia ją į privačią (private) direktoriją serveryje naudojant FTP protokolą.



17 pav. Vartotojo sąsaja video apdorojimo programoje

Statistinių duomenų pateikimui galutiniam vartotojui naudojant mobilias technologijas buvo panaudota JAVA kalba atsižvelgiant į MIDP 1.0 specifikaciją [Web05]. Programa veikia mobiliajame telefone, turinčiame JAVA MIDP 1.0 palaikymą bei interneto ryšį (žr. 18 pav.). Duomenims gauti iš interneto naudojamas HTTP protokolas (HTTPS protokolas palaikomas tik MIDP 2.0 specifikacijoje). Duomenys kraunami vartotojui iš privačios (private) direktorijos atlikus autorizaciją serveryje. Sukompiliuoti ir surinkti galutinius failus mobiliajam telefonui buvo panaudota Sun Java Wireless Toolkit 2.2 programa [Web06].



18 pav. Vartotojo sąsaja mobiliajame telefone

Realizavimui panaudoti įrenginiai:

Kompiuteris:

- Procesorius - P4 3.0Ghz
- Operatyvioji atmintis – 512 Mb RAM
- Grafinė plokštė – Geforce 6600 GT 128 Mb RAM
- Operacinė sistema – Windows, FreeBSD

Web kamera:

Logitech PC Cam (640x480 pikselių) Windows sistemoje



Mobilusis telefonas:

Siemens SX1 (palaiko tik MIDP 1.0)

Eksperimento rezultatai

Visi bandymai buvo atlikti naudojant duomenų bazę iš 85 nuotraukų, 20 iš jų buvo parinktos atsitiktinai. Šiose nuotraukose daug skirtingų automobilių: lengvosios mašinos, sunkvežimiai, autobusai, mikroautobusai. Figūruoja kelių šalių automobilių numeriai. Nuotraukos buvo darytos iš skirtingų kampų bei skirtingo atstumo. Vidutinė kokybė visų nuotraukų gali būti apibrėžta 3 elementais: 640x480 pikseliai, 24 bitų spalvų erdvė, JPEG kompresija. Atlikto testo rezultatai pateikti lentelėje (Lentelė 1).

	Atpažinti numeriai	Neatpažinti numeriai	Atpažinti netikri numeriai
Nuotraukos su automobilio numeriais	46	17	2
Atsitiktinės nuotraukos	6	12	2

Lentelė 1 Atpažintų numerių rezultatai.
Atpažinimo efektyvumas maždaug 70%

Pastaba:

Atpažinimo teorijoje taip pat svarbus taip vadinamas "false positive" atpažintų numerių procentas. Atliktuose testuose jis siekia iki 30%.

Išvados

Video apdorojimai tapo svarbiu uždaviniu ir rado savo egzistavimą tokiose sferose, kaip informacijos teorija, telekomunikacija, transliacija. Tyrimo rezultatai parodė, kad neįmanoma parinkti tokių filtrų arba metodų kombinacijų, kurios leistų bet kokiomis sąlygomis efektingai atpažinti transporto numerį vaizduose (kiekvienas metodas ar filtras turi trūkumus, todėl reikia bandyti rasti „aukso vidurį“). Kiekvienu atveju reikia atskirai parinkti filtrų ar metodų kombinaciją, kuri leistų prisitaikyti prie oro sąlygų, kameros padėties, kameros drebinimo, kameros gaunamo vaizdo, pradinių duomenų ir t.t.

Literatūra

1. [Dav90] **E. Davies** *Machine Vision: Theory, Algorithms and Practicalities*, Academic Press, 1990, pp 42 - 44.
2. [GonWo92] **R. Gonzalez and R. Woods** *Digital Image Processing*, Addison-Wesley Publishing Company, 1992, p 191.
3. [Mar91] **A. Marion** *An Introduction to Image Processing*, Chapman and Hall, 1991, p 274. , Chap. 5.
4. [BoyTho88] **R. Boyle and R. Thomas** *Computer Vision: A First Course*, Blackwell Scientific Publications, 1988, pp 32 - 34.
5. [Jai83] **A. Jain** *Fundamentals of Digital Image Processing*, Prentice-Hall, 1986, Chap. 7.
6. [Ham83] **R. Hamming** *Digital Filters*, Prentice-Hall, 1983.
7. [Sch89] **R. Schalkoff** *Digital Image Processing and Computer Vision*, John Wiley & Sons, 1989, Chap. 4.
8. [Lu95] **Lu, Y**, "Machine printed character segmenation", *Pattern Recognition*, vol. 28, n. 1, pp 67-80, Elsevier Science Ltd, UK, 1995.
9. [ShcNep91] **Shchepin E V & Nepomnyashchii G M**, *Character Recognition via Critical Points'*, *International Journal of Imaging Systems and Technology*, vol.3 (1991), pp. 213-221.
10. [StoMil94] **Storer R, Milford D J, Bulas-Cruz J & Dagless E L**, *Developing Embedded Applications in an Array of Specialised Transputer Modules*, Proc. WOTUG, Bristol, March (1994).
11. [Bul95] **Bulas-Cruz J**, *Image Processing Applications using a Transputer-based System*, PhD Thesis, University of Bristol, UK (1995).
12. [AliDag91] **Ali A T & Dagless E L**, "Computer Vision-Aided Road Traffic Monitoring", Proc. ISATA 24th. Int. Conf. Road Transport Informatics (RTI) & Intelligent Vehicle-Highway Systems (IVHS), Florence, Italy, pp.55-61, May 1991.
13. [AliBul93] **Ali A T, Bulas-Cruz J & Dagless E L**, "Vision Based Road Traffic Data Collection", Proc. ISATA 26th International Conference, (ATT & IVHS), Aachen, Germany, Sept. (1993).
14. [Dag92] **Dagless E L & Ali A T**, "Real-Time Vision Aided Traffic Monitoring", Proc. ACEP '92 Workshop, Borowice, Poland, April (1992).
15. [Web01] http://en.wikipedia.org/wiki/Automatic_number_plate_recognition

16. [Web02] <http://www.buckeyetraffic.org/webcams/>
17. [Web03] <http://video.dot.ca.gov/>
18. [Web04] <http://www.swisswebcams.ch/english/>
19. [Web05] <http://java.sun.com/products/midp/>
20. [Web06] <http://java.sun.com/products/sjwtoolkit/>