

KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR VALDYMO INŽINERIJOS
FAKULTETAS
PROCESŲ VALDYMO KATEDRA

Saulius Sinkevičius

**PĖSČIŲJŲ SRAUTO MONITORINGAS
HIERARCHINĖS LAIKINĖS ATMINTIES
ATPAŽINIMO ALGORITMŲ PAGRINDU**

Magistro darbas

Darbo vadovas

prof. habil. dr. Rimvydas Simutis

Kaunas, 2011

KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR VALDYMO INŽINERIJOS
FAKULTETAS
PROCESŲ VALDYMO KATEDRA

Saulius Sinkevičius

**PĖSČIŲJŲ SRAUTO MONITORINGAS
HIERARCHINĖS LAIKINĖS ATMINTIES
ATPAŽINIMO ALGORITMŲ PAGRINDU**

Magistro darbas

Recenzentas

dr. Vidas Raudonis

2011-05-25

Vadovas

prof. habil. dr. Rimvydas Simutis

2011-05-25

Atliko

IFM-9/1 gr. stud.

Saulius Sinkevičius

2011-05-25

TURINYS

LENTELIŲ SĄRAŠAS.....	4
PAVEIKSLŲ SĄRAŠAS.....	4
ĮVADAS.....	6
1. LITERATŪROS ANALIZĖ.....	9
1.1. PRITAIKOMŲ JUTIKLIŲ APŽVALGA.....	9
1.1.1. <i>Mechaniniai arba elektroniniai skaičiuokliai</i>	9
1.1.2. <i>Turniketai</i>	10
1.1.3. <i>Svorio matavimai</i>	10
1.1.4. <i>IR spinduliniai skaičiuokliai</i>	11
1.1.5. <i>IP arba CCTV kameros</i>	12
1.1.6. <i>Terminės ir IR (infraraudonųjų spindulių) kameros</i>	13
1.1.7. <i>WEB kameros</i>	14
1.1.8. <i>Ultragarso atstumo jutikliai</i>	15
1.1.9. <i>Kompleksiniai sprendimai</i>	15
1.2. METODŲ APŽVALGA.....	16
1.2.1. <i>Sprendimai ir idėjos</i>	16
1.2.2. <i>Klasifikavimas ir atpažinimas</i>	17
1.2.3. <i>Objektų išskyrimas</i>	17
1.2.4. <i>Judėjimo krypties nustatymas</i>	17
1.2.5. <i>Fono pašalinimas</i>	17
1.2.6. <i>Filtravimas</i>	18
1.3. ESAMŲ SPRENDIMŲ TRŪKUMAI IR APRIBOJIMAI.....	18
1.4. TAIKYMO SRITYS.....	20
1.5. SWOT ANALIZĖ.....	21
1.5.1. <i>SWOT (SSGG) analizės faktoriai</i>	21
1.5.2. <i>Faktorių analizė su WinSWOT</i>	22
1.5.3. <i>Faktorių analizė su Excel</i>	26
1.5.4. <i>Miglotą pažintinio kelio (MPP) sudarymas</i>	27
1.5.5. <i>Galimybių didinimas ir grėsmių mažinimas</i>	28
1.5.6. <i>Mazgų reakcijos funkcijos</i>	30
1.5.7. <i>Dinaminė analizė</i>	32
1.5.8. <i>Monitoringas</i>	34
1.6. ANALIZĖS IŠVADOS.....	35
2. PĖSČIŲJŲ SKAIČIAVIMO METODIKA.....	37
2.1. PROTOTIPAS.....	37
2.1.1. <i>Techninės įrangos pasirinkimas</i>	37
2.1.2. <i>Pradinė sprendimo idėja</i>	38
2.1.3. <i>Metodų pasirinkimas</i>	40
2.1.4. <i>Programinė įranga</i>	41
2.1.5. <i>Sistemos tikslas</i>	42
2.1.6. <i>Dėmesio matmenų optimizavimas</i>	44
2.1.7. <i>Filtro optimizavimas</i>	44
2.1.8. <i>Prototipo testavimo rezultatai</i>	46
2.1.9. <i>Prototipo testavimo išvados</i>	47
2.2. PASIŪLYTA METODIKA.....	48
2.2.1. <i>Patobulinimų esmė</i>	48
2.2.2. <i>Realaus laiko skaičiavimo algoritmas</i>	50
2.2.3. <i>Fazė 1. Duomenų nuskaitymas ir paruošimas</i>	51
2.2.4. <i>Fazė 2. Praeivių atpažinimas</i>	53
2.2.5. <i>Fazė 3. Krypties nustatymas</i>	55
2.2.6. <i>Fazė 4. Praeivių skaičiavimas</i>	56
2.2.7. <i>Algoritmo išplėtimas platesniems praėjimams</i>	56
3. TYRIMŲ REZULTATAI.....	58
3.1. VIENO SEGMENTO REZULTATAI.....	58
3.2. DVIEJŲ SEGMENTŲ REZULTATAI.....	60

3.3.	HTM IR OPTINIO SRAUTO (OPTICAL FLOW) METODŲ PALYGINIMAS	61
3.4.	„TECHNORAMA 2011“ PARODOS LANKYTOJŲ SKAIČIAVIMO REZULTATAI	62
IŠVADOS		64
SUMMARY		65
LITERATŪRA.....		66
TERMINŲ IR SUTRUMPINIMŲ ŽODYNAS.....		69
PRIEDAI		70
1	PRIEDAS. SUTARTIS DĖL STUDENTŲ MOKSLINIŲ TYRIMŲ VYKDYMO	70
2	PRIEDAS. STRAIPSNIS LEIDINYJE „THE 5TH INTERNATIONAL CONFERENCE ON ELECTRICAL AND CONTROL TECHNOLOGIES“	72
3	PRIEDAS. STRAIPSNIS MOKSLO ŽURNALUI „ELECTRONICS AND ELECTRICAL ENGINEERING“ (PRIIMTA SPAUSDINIMUI)	76
4	PRIEDAS. STRAIPSNIS LEIDINYJE „THE 6TH INTERNATIONAL CONFERENCE ON ELECTRICAL AND CONTROL TECHNOLOGIES“	82
5	PRIEDAS. PROGRAMINĖS ĮRANGOS REALIZACIJA	86
	<i>Pagrindinės sistemos funkcijos</i>	86
	<i>Duomenų bazės schema</i>	86
	<i>HTM tinklų struktūra</i>	88
	<i>Grafinis vartotojo interfeisas</i>	89
6	PRIEDAS. PADĖKA UŽ PRAKTINĮ DARBO TAIKYMĄ „TECHNORAMA 2011“ PARODAI	92

LENTELIŲ SĄRAŠAS

1	lentelė. SWOT (SSGG) analizės faktoriai	21
2	lentelė. Galimybių įverčiai	22
3	lentelė. Grėsmių įverčiai	22
4	lentelė. Stiprybių įverčiai	23
5	lentelė. Silpnybių įverčiai	24
6	lentelė. Momentinės vertės	26
7	lentelė. Trumpalaikės vertės	27
8	lentelė. Ilgalaikės vertės	27
9	lentelė. Eksperimento rezultatai	60
10	lentelė. Parodos rezultatai	63
11	lentelė. Naudojamos sąvokos, terminai ir sutrumpinimai	69

PAVEIKSLŲ SĄRAŠAS

1	pav. a) mechaninis skaičiuoklis b) elektroninis skaičiuoklis	9
2	pav. Turniketai	10
3	pav. Svorio matavimo jutiklis, lengvai užmaskuojamas.....	10
4	pav. a) IR barjero veikimo principas b) IR barjero pavyzdys.....	11
5	pav. Du IR barjero įrengimo variantai: su krypties nustatymu ir be.....	11
6	pav. a) IP kamera b) CCTV kamera	12
7	pav. Terminės kameros įrengimo pavyzdys	13
8	pav. a) terminė kamera b) kamera su IR apšvietimu	13
9	pav. WEB kamera	14
10	pav. Ultragarso jutiklis.....	15
11	pav. Kompleksinis kompiuterinės regos sprendimas su daug jutiklių.....	15
12	pav. Galimybių įverčiai.....	23
13	pav. Grėsmių įverčiai.....	23
14	pav. Stiprybių įverčiai.....	24

15 pav. Silpnųjų įverčiai	24
16 pav. SSGG grafikas su normalia rizika.....	25
17 pav. Grafikas su didžiausia rizika (blogiausias atvejis).....	26
18 pav. Galimybės ir grėsmės momentiniu (1), trumpuoju (2) ir ilguoju (3) laikotarpiu.....	27
19 pav. Nusistovėjęs MPP	28
20 pav. ST3 vertės poveikis GL3 įtakai ir visų galimybių sumai.....	29
21 pav. SL3 vertės poveikis GR2 įtakai ir visų grėsmių sumai.....	29
22 pav. ST3 vertės poveikis GR2 įtakai ir visų grėsmių sumai.....	30
23 pav. SL3 vertės poveikis GL3 įtakai ir visų galimybių sumai.....	30
24 pav. Stiprybių ir silpnųjų mazgų reakcijos funkcijos	31
25 pav. Galimybių ir grėsmių mazgų reakcijos funkcijos	32
26 pav. Grėsmių priklausomybė nuo iteracijų.....	33
27 pav. Galimybių priklausomybė nuo iteracijų.....	33
28 pav. Monitoringo schema	34
29 pav. Grėsmių pokyčiai	34
30 pav. Galimybių pokyčiai.....	35
31 pav. Sistemos techninė realizacija	37
32 pav. Sistemos matomi vaizdai	38
33 pav. Numatomas žmonių pasiskirstymas erdvėje.....	38
34 pav. a) skenuojami visi segmentai b) atpažinimas vienu metu tik viename segmente.....	39
35 pav. Numatomi skenavimo rezultatai	39
36 pav. HTM koncepcija ir trijų lygių tinklo struktūra.....	40
37 pav. Dvi duomenų klasės a) vien tik žmonės b) visa kita.....	41
38 pav. Viršugalvio dydžio pasiskirstymas	41
39 pav. Identifikavimo uždavinys.....	43
40 pav. Dėmesio dydžio gerumo funkcija	44
41 pav. Triukšmo filtro gerumo funkcija.....	45
42 pav. a) žmogaus sekimo pavyzdys b) skenavimo rezultatų pavyzdys.....	46
43 pav. Judesio aptikimas	47
44 pav. Segmentacijos pavyzdys a) vienas segmentas b) du segmentai.....	48
45 pav. Numatomi rezultatai, tikimybė, kad segmente yra žmogus	49
46 pav. Pėsčiųjų srauto monitoringo algoritmo schema.....	51
47 pav. Segmentacijos pavyzdys	52
48 pav. Keletas pavyzdžių iš apmokymo duomenų: a) tik praeiviai b) visa kita	52
49 pav. Juosto aktyvumo pokyčiai įeinant žmogui į patalpą.....	56
50 pav. Dviejų segmentų pavyzdys platesniam praėjimui.....	57
51 pav. TBI ir „maxProp“ atpažinimo palyginimas	58
52 pav. Suprojektuoto tinkle ir „Vision toolkit“ tinkle palyginimas	59
53 pav. Žmonių atpažinimo duomenys buvo apdoroti naudojant slenksčius	59
54 pav. Juostelių aktyvumo duomenys buvo apdoroti naudojant slenksčius	59
55 pav. Dalis rezultatų abiem segmentams. 1- vienas žmogus įėjo, -1 – vienas išėjo.....	60
56 pav. Grafinis interfeisas parodos lankytojų skaičiavimo rezultatams atvaizduoti.....	62
57 pav. Sistemos duomenų bazės schema	87
58 pav. „params.py“ failas tinklo sukūrimui ir apmokymui „Vision toolkit“ sistemoje.....	88
59 pav. WEB kameros vaizdai automatiškai rašomi į DB.....	89
60 pav. Galimybė atlikti skirtingus eksperimentus su tais pačiais duomenimis.....	89
61 pav. Patogus būdas grafiškai tvarkyti sensorius	90
62 pav. Grafinis interfeisas leidžia greitai paruošti tūkstančius paveikslukų apmokymui.....	90
63 pav. Apmokymo paveikslukų iškarpymas iš kameros vaizdų ir sistemos apmokymas	91
64 pav. Rezultatų peržiūros ir tyrimų langas	91
65 pav. Padėka už praktinį darbo taikymą „Technorama 2011“ parodai	92

IVADAS

Pėsčiųjų srauto duomenys viešajame transporte, muziejuose, galerijose, parduotuvėse, prekybos centruose, ir kitokiuose objektuose, padeda sumažinti veiklos išlaidas, didinti darbo efektyvumą, geriau paskirstyti resursus. Dažnai tokie duomenys naudojami ir saugumo reikmėms. Gaisrininkams tikslus žmonių kiekis ar žmonių pasiskirstymas pastate padėtų planuoti gelbėjimą. Muziejuose saugumo sistema išpėtų, jei ne visi lankytojai paliko patalpas. O esant reikalui, tokia sistema galėtų peržiūrėti labai ilgus vaizdo įrašus ir aptikti tas scenas, kuriose buvo užfiksuotas žmogus. Tai padarytų greitai ir be žmogaus įsikišimo. Tokie duomenys taip pat naudingi versle, nes padeda prognozuoti lankytojų elgesį laike ir atitinkamai pakoreguoti veiklos modelį.

Pritaikymo ir poreikio sričių tokioms sistemoms yra daug. Dažniausiai šį darbą atlieka žmonės, tačiau tai labai neefektyvu. Žmogus stebėtojas labai greitai pavargsta, negali išlaikyti pastovaus dėmesio kelias ar keliolika valandų be perstojo, jis negali būti vienu metu keliose vietose, be to jam reikia mokėti už darbo laiką. Pavyzdžiui viešojo transporto įmonės dažnai samdo žmogų, kuris važiuoja jam sutartais maršrutais ir fiksuoja keleivių srautą. Šie surinkti duomenys yra naudojami sprendžiant maršrutų planavimą, tačiau jei mieste yra kokie 15-20 maršrutų, šis vienas žmogus niekaip negali jų visų išbandyti, juolab įvertinti koks yra konkrečių stotelių srauto pasikeitimas kai vyksta šventės, skirtingu paros metu, negali palyginti stotelių apkrautumo tarpusavyje. Tokie duomenys dar yra ir netikslūs, nes labai sudėtinga sektis didesnius srautus, žmogus nepajėgus ilgai išlaikyti dėmesio į daug objektų vienu metu.

Paskutiniaisiais metais, kai jutikliai ir kompiuteriai tampa vis geresni, pigesni, lengviau prieinami, atsiranda vis naujų bandymų šią problemą spręsti kompiuterinės regos pagalba. Pirmame, literatūros analizės, skyriuje yra apžvelgiami šiuo metu prieinami jutikliai, bandoma juos įvertinti ir taip pasirinkti racionalesnį sprendimą. Taip pat susipažįstama su esamais sprendimais. Tai leidžia geriau suvokti kokios sprendimų idėjos vyrauja ir ką būtų galima daryti kitaip, kad tuos apribojimus, su kuriais tyrėjai susiduria, būtų galima apeiti.

Deja, nei vienas literatūroje sutinkamas metodas dar negali išspręsti visų problemų susietų su pėsčiųjų srauto skaičiavimu. Sudėtingiausios situacijos yra kai kinta apšvietimo sąlygos, o pėstieji vizualiai susilieja. Šiais atvejais visos sistemos yra nepakankamai tikslios. Taip pat tokioms sistemoms yra keliamas reikalavimas, kad jos veiktų realiuoju laiku, tačiau sudėtingi matematiniai metodai yra per lėti, todėl daugelis tiriamų metodų sunkiai ar visai netinkami vykdymui realiu laiku.

Antrame skyriuje yra ne tik pasiūloma idėja kaip šią problemą būtų galima spręsti, parenkama įranga, bet ir išbandžius prototipą, atliekami metodikos patobulinimai. Sprendimui buvo pasirinkta hierarchinė laikinė atmintis (HTM) [1], gana naujas ir inovaciškas atpažinimo metodas. Literatūroje nepavyko aptikti bandymų šį metodą naudoti sprendžiant pėsčiųjų monitoringo problemą.

HTM algoritmai yra dar visai nauji ir nebrandūs, jie paremti idėja, kad pirmiausia reikia išsiaiškinti kaip veikia intelektas [2] užuot kurti dirbtinį intelektą nuo nulio, todėl koncepcija susideda iš įvairių mokslo šakų rezultatų, sujungiant juos į vieną visumą. Autoriai šiuos metodus po truputi tobulina ir artimiausiu metu galima tikėtis antros versijos. Jie įkūrė įmonę pavadinimu Numenta, kurioje savo idėjas vysto toliau. Daug medžiagos bei programinės įrangos yra prieinama viešai. Kūrėjai tikisi, kad pavyks suburti bendruomenę ir taip vizija perprasti intelektą taps realybe žymiai greičiau.

Šiuo metu Numenta yra sukūrusi „NuPIC“ karkasą, kuriuo galima projektuoti HTM tinklus, juos apmokyti, prijungti įvairiausių jutiklius, pavyzdžiui: vaizdus, garsus ar skaičius, bei atlikti atpažinimą. „Vision“ karkasas yra išplėsta „NuPIC“ karkaso dalis, skirta dirbti su vaizdais. Juo lengva kurti tinklus ir atlikti eksperimentus.

Deja ši programinė įranga yra gana lėta, veikia „Python“ kalbos pagrindu. Darbo metu išbandyta „NuPIC“ sistema paleisti klasteryje. Pavyko, tačiau skaičiavimai nebuvo greitesni nei personaliniame kompiuteryje. Dėl to, kad dalis kodo užkoduota, buvo neįmanoma išsiaiškinti, kodėl programinė įranga net ir klasteryje veikia lėtai.

Darbo tikslas: pasiūlyti pėsčiųjų srauto monitoringo metodiką.

Darbo uždaviniai:

- apžvelgti kokius jutiklius įmanoma taikyti, kokie jų privalumai ir trūkumai;
- peržvelgti literatūrą kaip šį tikslą bando pasiekti kiti, kas jiems pavyksta, kas nepavyksta, su kokiomis kliūtėmis ir apribojimais susiduriama;
- pasiūlyti pirminę idėją tikslui pasiekti, patikrinti ją sukuriant ir išbandant prototipą;
- pasiūlyti patobulintą metodiką pėsčiųjų srauto monitoringui atsižvelgiant į gauto prototipo trūkumus;
- sukurti pasiūlytai metodikai programinę įrangą ir išbandyti su realiais duomenimis.

Eigoje buvo atlikta daug ir įvairių tyrimų:

- nuo 2010 m. vasario 1 d. iki 2010 m. gegužės 31 d. buvo atlikti Lietuvos mokslo tarybos rengiami studentų moksliniai tyrimai (smt.smpraktika.lt). Mokslinio

tiriamąo darbo tema „Hierarchinės laikinės atminties atpažinimo algoritmų taikymas keleivių srauto monitoringui“ (žr. priedą 1);

- konferencijoje „The 5th International Conference on Electrical and Control Technologies“ perskaitytas pranešimas ir jos leidinyje išspausdintas straipsnis „Humans Tracking in a Public Places with WEB Camera Using Hierarchical Temporal Memory Algorithms“ (žr. priedą 2);
- konferencijoje „The 15th International Conference of ELECTRONICS“ perskaitytas pranešimas. Mokslo žurnalui „Electronics and Electrical Engineering“ parengtas ir priimtas spausdinimui straipsnis „Monitoring of Humans Traffic Using Hierarchical Temporal Memory Algorithms“ (žr. priedą 3);
- konferencijoje „The 6th International Conference on Electrical and Control Technologies“ perskaitytas pranešimas ir jos leidinyje išspausdintas straipsnis „Optical Flow Versus Hierarchical Temporal Memory Algorithms For Human Traffic Monitoring System“ (žr. priedą 4);
- nuo 2011 m. kovo mėn. vykdomi darbai prie projekto „Aptarnavimo paslaugų teikimo klientų srautų monitoringo ir valdymo sprendimas“ (APTiKSiM) (esparama.lt), pritaikant magistro darbo rezultatus žmonių monitoringui. Realizuotos programinės įrangos trumpas aprašymas pateiktas prieduose (žr. priedą 5);
- paruoštas eksponatas „Technorama 2011“ parodai, o šių metų gegužės ketvirtą dieną sistema skaičiavo parodos lankytojus realiuoju laiku. Eksponatas gavo padėkos raštą už praktinį sistemos pritaikymą (žr. priedą 6).

Darbe pasiūlyta pėsčiųjų srauto monitoringo metodika patikrinta praktiškai, išmėginta pakankamai realiomis sąlygomis.

Aprašyta metodika gali būti tobulinama kuriant tikslesnius HTM tinklus, ieškant geresnės apmokymo strategijos, tiriant apšvietimo ir kitokių aplinkos sąlygų poveikį. Pradiniai rezultatai yra geri, numatoma sistemą vystyti iki komercinio produkto.

1. LITERATŪROS ANALIZĖ

Šiame skyriuje apžvelgiama kokius jutiklius būtų galima pritaikyti, kokie jų privalumai ir trūkumai, kaip kiti sprendžia nagrinėjamą uždavinį, su kokiomis problemomis susiduria, kurios iš jų jau išspręstos, o kurių įveikti dar nepavyko. Taip pat atkreipiamas dėmesys į tokio pobūdžio sistemų pritaikomumą bei atliekama nedidelė SWOT analizė pagal pasirinktus faktorius rizikai sumažinti.

1.1. Pritaikomų jutiklių apžvalga

Nuo pasirinktų jutiklių ir jų įrengimo būdo smarkiai priklauso gaunami duomenys, o tai labai didelis apribojimas ir iš esmės nulems kuriuos metodus ir algoritmus naudoti galima, o kurie jau netinka. Be to metodų efektyvumas gali gana ženkliai skirtis priklausomai nuo įrangos. Todėl pabandydysime apžvelgti rinkoje esamų jutiklių variantus.

1.1.1. Mechaniniai arba elektroniniai skaičiuokliai

Tai paprasčiausias sprendimas nesudėtingoms situacijoms. Tokius skaičiuoklius galima kabinti prie durų ar praėjimų užtvarų. Jie turi mygtuką, kurį nuspaudus, skaitliuko kiekis padidinamas vienetu. Gali būti mechaniniai arba elektroniniai.



1 pav. a) mechaninis skaičiuoklis b) elektroninis skaičiuoklis

Privalumai:

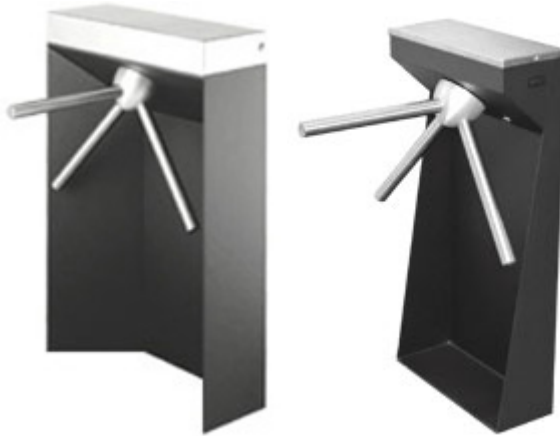
- labai maža kaina;
- paprastumas;
- nereikia didelių skaičiavimo pajėgumų;

Trūkumai:

- netinka dideliems srautams ar platiems praėjimams;
- tenka vesti žurnalą;
- nenustato krypties.

1.1.2. Turniketai

Vienas paprasčiausių sprendimų, plačiai naudojamų vienos krypties žmonių skaičiavimui. Praėjimas atidaromas, jei įmetama moneta, bilietas ir pan.



2 pav. Turniketai

Privalumai:

- paprastumas;
- naudojamas kaip barjeras;
- puikus tikslumas, nes vienu metu gali praeiti tik vienas žmogus.

Trūkumai:

- nepatogus;
- didelė kaina;
- visai netinka dideliems srautams ar platesniems praėjimams.

1.1.3. Svorio matavimai

Matuojant svorį galima gauti netiesioginius žmonių kiekių įverčius. Pavyzdžiui, du laipteliai su svorio matuokliais gali padėti nustatyti žmogaus judėjimo kryptį.



3 pav. Svorio matavimo jutiklis, lengvai užmaskuojamas

Privalumai:

- galima įrengti nepastebimai;
- derinant kelis įmanoma nustatyti judėjimo kryptį;

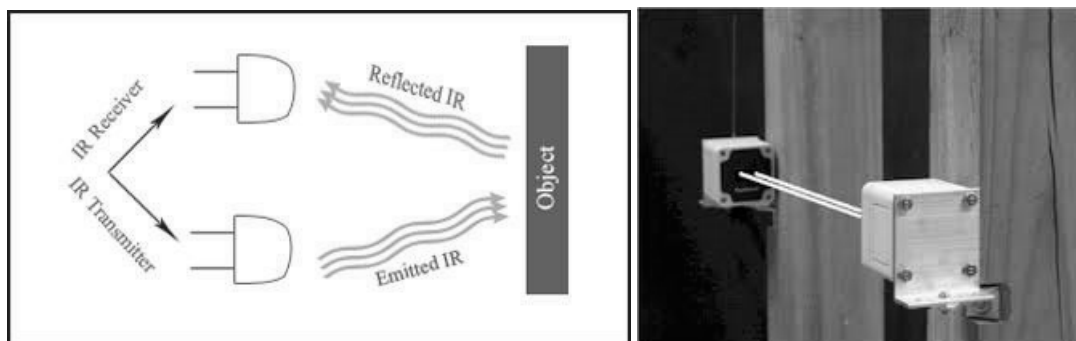
Trūkumai:

- netinka platiems praėjimams;
- sensorių galima peržengti arba kliudyti tik dalinai.

1.1.4. IR spinduliniai skaičiuokliai

Tokie skaičiuokliai įrengiami praėjimuose kaip barjeras. Objektui kertant spindulį, spindulys nutraukiamas ir daviklis priskaičiuoja prie skaitliuko vienetą. Tokie įranginiai paprastai sujungti bevieliu tinklu su personaliniu kompiuteriu bei turi minimalų duomenų atvaizdavimą.

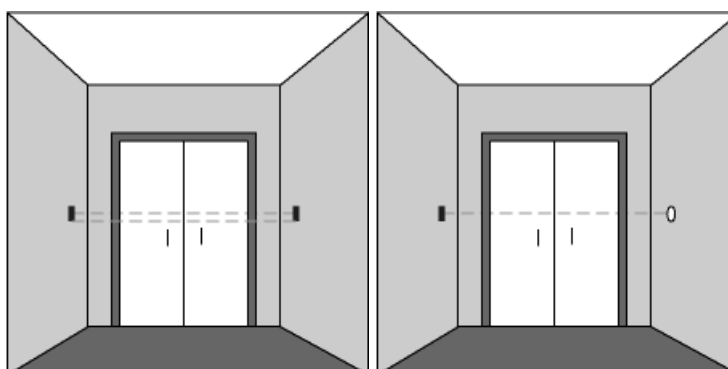
Toks įrenginys turi dvi dalis, kurių viena siunčia infraraudonąsias bangas, o kita priima tų bangų atspindį nuo objekto.



a)

b)

4 pav. a) IR barjero veikimo principas b) IR barjero pavyzdys



5 pav. Du IR barjero įrengimo variantai: su krypties nustatymu ir be

Kai kurie spinduliniai skaičiuokliai gali nustatyti kryptį, tačiau tikslumas labai priklauso nuo srauto pobūdžio. Jei žmonės eina pavieniui, tai tikslumas bus gerokai didesnis, nei eitų grupės.

Šie skaičiuokliai veikia nuo 2,5 iki 6 metrų atstumu, kai kada tokius skaičiuoklius įrengus vertikaliai, tikslumas gali padidėti.

Privalumai:

- maža kaina;
- paprasta įrengti;
- tinka platesniam praėjimui.

Trūkumai:

- daugelis negali nustatyti krypties;
- negali išskirti žmonių einančių šalia;
- negali paskaičiuoti didelio ar nenutrūkstamo srauto;
- didelė tikimybė nebeveikti, jei kas nors pastoviai stovės šalia praėjimo;
- nuo tiesioginės saulės tikslumas gali labai sumažėti.

1.1.5. IP arba CCTV kameros

IP kamera gali duomenis persiųsti kompiuterių tinklu arba internetu. Necentralizuotos IP kameros turi galimybę rašyti duomenis tiesiai į kietąjį diską ar kitokią talpyklą, o centralizuotoms reikalingas „Network Video Recorder (NVR)“ įrašymo valdymas.

CCTV kameros būna analoginės arba skaitmeninės. Analoginių kamerų prie mažesnių „fps“ vaizdas gali pastebimai išsiliesti arba vaizdas trukčioja, todėl žmonių skaičiavimui tai gali trukdyti. Skaitmeninėms kameroms nereikia vaizdo plokštės verčiančios analoginį signalą į skaitmeninį ir ji gali duomenis saugoti tiesiai į kompiuterį. Šios kameros labai paplitusios norint užtikrinti saugumą.



a)

b)

6 pav. a) IP kamera b) CCTV kamera

Privalumai:

- puiki rezoliucija, bent 640x480;
- net iki 30 kadrų per sekundę;

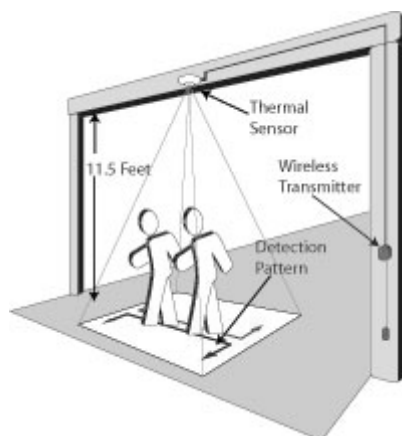
- kamera pasiekama per atstumą iš įvairių įrenginių;
- gali veikti bevieliu tinklu.
- didelis tikslumas, jei gerai priderinta;
- lengva integruoti į kitas sistemas norint gauti išvestinius rezultatus;
- įmanoma nustatyti judėjimo kryptį.

Trūkumai:

- reikalingas didesnis tinklo pralaidumas;
- galimos saugumo problemos jungiant tokią kamerą prie tinklo;
- didelis jautrumas apšvietimo ir aplinkos sąlygoms;
- kiekvienu individualiu atveju sistemą tenka sukalibruoti.

1.1.6. Terminės ir IR (infraraudonųjų spindulių) kameros

Šios kameros fiksuoja šilumos šaltinius arba kitaip tariant – infraraudonuosius spindulius. Tai gali būti patogu skaičiuoti žmones, nes tokiu būdu matoma nuo jų sklindanti šiluma. Tikslumas gali būti geresnis nei CCTV ar IP kamerų, nes jos mažiau jautrios apšvietimui. Be to, nebūtini sudėtingi algoritmai fono pašalinimui.



7 pav. Terminės kameros įrengimo pavyzdys



a)



b)

8 pav. a) terminė kamera b) kamera su IR apšvietimu

Privalumai:

- įmanomas krypties nustatymas;
- mažai jautri apšvietimui;
- gali skaičiuoti net jei visiškai tamsa;
- didelis tikslumas, jei gerai suderinta;
- ilga gyvavimo trukmė.

Trūkumai:

- didesnė kaina;
- mažesnis matymo laukas nei video sistemų;
- negali būti naudojamos, jei lubos yra per žemos.

1.1.7. WEB kameros

Šios kameros labai greitai paplito dėl savo paprastumo, pigumo ir sąlyginai geros kokybės. Yra be laidžių ir jungiamų per USB jungtį. Tokios kameros labai tinka įvairiems su vaizdais susijusiems eksperimentams, nes jas lengva įsigyti ir taikyti, labai tinka dirbant su sistemų prototipais.



9 pav. WEB kamera

Privalumai:

- maža kaina;
- gera kokybė;
- lengvai pritaikoma.

Trūkumai:

- mažas atsparumas atšiaurioms gamtos sąlygoms;
- didelis jautrumas apšvietimui.

1.1.8. Ultragarso atstumo jutikliai

Toks sensorius iki objekto siunčia ultragarso bangas ir gaudo atspindėjusias. Taip gaunamas laikas, kurį užtruko sklindantis signalas iki objekto ir galima nustatyti kokių atstumu objektas yra nutolęs.



10 pav. Ultragarso jutiklis

Privalumai:

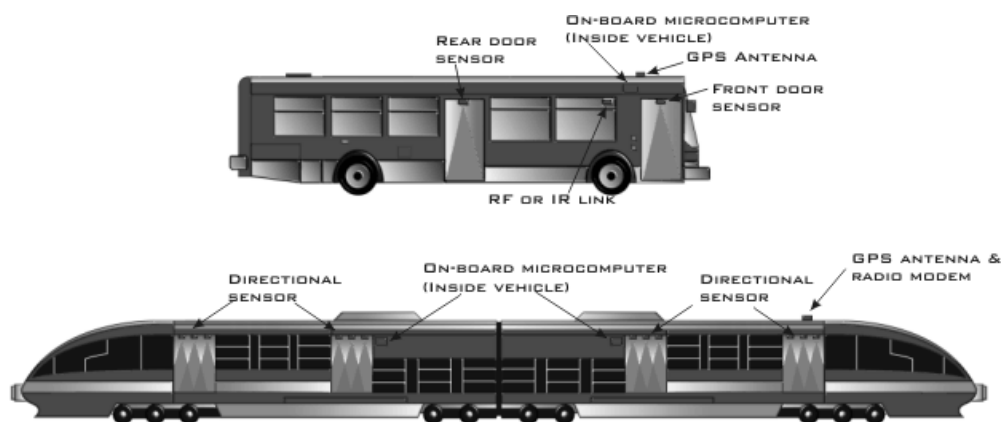
- maža kaina;
- paprastumas;

Trūkumai:

- didelė paklaida;
- jautrus triukšmams.

1.1.9. Kompleksiniai sprendimai

Dažnai sistemos yra rengiamos taip, kad būtų galima pritaikyti dideliems pastatams ar viešajam transportui, kuomet yra daug praėjimų, jie skiriasi savo parametrais. Tokiais atvejais reikia daug sensorių, jie derinami tarpusavyje, o rezultatai gaunami apdorojant duomenis iš visų sensorių.



11 pav. Kompleksinis kompiuterinės regos sprendimas su daug jutiklių

Privalumai:

- sprendimas adaptuojamas dideliems objektams;
- gaunami išsamūs rezultatai;

Trūkumai:

- labai didelė kaina, reikia didelių pradinių investicijų;
- technologiškai sudėtinga;
- būtina gerai parengta programinė įranga rezultatams apdoroti.

1.2. Metodų apžvalga

Nors uždavinys iš pirmo žvilgsnio atrodo paprastas - skaičiuoti pėsčiuosius, tačiau apžvelgus literatūrą akivaizdu, kad problema yra kompleksinė. Ir ji sprendžiama suskaidant užduotį į dalis, kurių kiekviena gali būti sprendžiama skirtingo tipo algoritmais ir metodais. Labai svarbu ne tik atpažinti žmones, bet ir tinkamai parinkti jutiklius, jų įrengimo būdą, atlikti žmonių paiešką, sekti laike, nustatyti judėjimo kryptį ir suskaičiuoti.

Šiame skyriuje apžvelgtos įvairios priemonės ar idėjos, kurios jau yra taikomos problemai spręsti arba galėtų būti taikomos.

1.2.1. Sprendimai ir idėjos

Sprendžiama problema yra labai komplikauta dėl realybės įvairovės, todėl visi sprendimai pasiteisina tik iš dalies. Dažniausiai naudojamus sprendimus pabandydysime klasifikuoti pagal įvairius požymius. Bendros idėjos yra tos pačios ir jos atsikartoja.

Jei padėti pedala matuojantį svorį, galima aptikti žmogaus kūno kinematikos dėsninumus [3], kurie leistų nustatyti žmogaus judėjimo kryptį. Pastačius vaizdo kamerą iš šono, galima sekti veidus ir atpažinti žmones pagal veido formą [4] arba pagal odos toną [5], naudoti slenkstį spalvoms [6] ar sukurti odos spalvos modelį [7, 8]. Norint nustatyti ar žmonių yra daug ar mažai, galima bandyti surasti kontūrus, ir jei vaizdo įrašas yra iš šono, tai vertikalių kontūrų kiekis galėtų nemažai pasakyti apie žmonių kiekį [9,10]. Turint 2D vaizdą, su truputi korekcijų, galima ieškoti objektų simetrijos [11], nes žmogus su tam tikromis išimtimis iš šalies atrodo gana simetriškas. Įmanoma net sudaryti 3D žmogaus formos modelį [12], nors naudojami ir 2D formos paieška [13], tačiau pozų įvairovės problemos išspręsti pilnai nesigauna. Kai kurias problemas galima sėkmingai spręsti stereo vaizdų [14, 15] pagalba, tačiau dažniausiai renkamosi paprasčiausia vaizdo kamera ir ieškomi spartesni būdai. Dažnai labai lemia sensorių parinkimas, pavyzdžiui naktiniam matymui galima taikyti infraraudonųjų spindulių vaizdo kameras [16].

Šio uždavinio specifika, kad žmonės juda, keičiasi. Vadinasi galima pabandyti aptikti judesį [17] ir tuo pasinaudojant segmentuoti atskirus objektus. Jei keli objektai pradžioje sekami pavieniui, jų susiliejimą mes kartais galime numatyti [18] visiškai neprarandant skaičiavimo tikslumo. Erdvė gali kisti ir pati, kas sukelia pakankamai nemažai sunkumų judesio analizei.

Aiškiausi sprendimai yra su vaizdo kameromis, tačiau galima rasti sistemų, naudojančių du ar daugiau jutiklių [19]. Tokios sistemos leidžia rasti dėsningumus tarp skirtingų jutiklių duomenų ir tuos duomenis apjungti. Naudojamų jutiklių yra labai įvairių, pavyzdžiui akustiniai [20], infraraudonųjų spindulių kameros, ultragarsas ir t.t.

1.2.2. Klasifikavimas ir atpažinimas

Šiam uždaviniui spręsti dažniausiai naudojami metodai yra SVM (Support vector machines), neuroniniai tinklai (Artificial neural networks - ANN), genetinis programavimas (Genetic programming - GP), klasteriai (Clustering), Bajeso tinklai (Bayesian networks), Eigenface, požymių parinkimas (features extraction), formų tikrinimas (shape matching).

1.2.3. Objektų išskyrimas

Norint atskirti objektus vieną nuo kito, naudojama segmentacija (segmentation), kontūrų radimas (edge detection), Snake algoritmas, kontūrų svorių žemėlapis (Weight-map, Weighted-Gradient image), judesio (motion) analizė. Yra įvairių spalvų ar tekstūrų atskirti padedančių metodų kaip spalvų erdvė (color space) vietoj RGB imant LAB, t.y. spalvos išreikštos skirtingu šviesumu ir intensyvumu, taip lengviau atskirti foną. Naudojami HSV ir RGB odos modeliai. Netinkami objektai eliminuojami tikrinant jų formą.

1.2.4. Judėjimo krypties nustatymas

Toks uždavinys sprendžiamas analizuojant vaizdo sekas. Dažnai panaudojami metodai: optinis srautas (Optical flow), branduolio pagrįstas sekimas (Kernel-based tracking), kontūrų sekimas (Contour tracking), savybių atitikimo tikrinimas (feature matching), įvairūs filtrai (Kalman, Particle), įvairūs slenksčiai (threshold).

1.2.5. Fono pašalinimas

Fono atskyrimui nuo objektų naudojami fono pašalinimo metodai (background-substraction), Gauso mišinio modelis (Gaussian Mixture Model), kadru skirtumas (frame difference) ir kitokie, deja dažnu atveju labai nepraktiški.

1.2.6. Filtravimas

Atliekami įvairaus pobūdžio duomenų filtravimai su Gauso (Gaussian filter), Kalmano (Kalman filter) filtrais. Filtrų pobūdis labai įvairus, gali būt kontrasto normalizavimo, kontūrų išskyrimo (Gabor), triukšmų šalinimo filtrai.

1.3. Esamų sprendimų trūkumai ir apribojimai

Visi bandymai skaičiuoti pėsčiuosius susiduria su aibe problemų ir apribojimų, kurių nepavyksta peržengti arba pavyksta tik iš dalies. Šiuos apribojimus turi ir komercinės sistemos ir publikacijose naujai siūlomi sprendimai. Tuos apribojimus bandoma įveikti skirtingais metodais ir algoritmais, tačiau dažnu atveju, konkretus sprendimas tinka tik labai apibrėžtomis sąlygoms, o universalus dar nėra.

Apžvelgus literatūra, galima nesunkiai sudaryti apribojimų sąrašą:

- tokios sistemos realiai veikia tik tada, jei aplinkos sąlygos tinkamos ir žmonės nesigrūda, kitu atveju sistema tampa arba labai netiksli arba visai „apanka“;
- skaičiuojant srauto balansą, reikia nustatyti objektų judėjimo kryptį, o tai praktiškai neįveikiama kliūtimi, jei keli žmonės bando prasilenkti vienu metu. Joks metodas dar negali atskirti vieno žmogaus nuo kito, kai jie yra labai arti. Problema ta, kad to paties žmogaus kūnas gali judėti į skirtingas puses tuo pačiu metu;
- spūsties atveju individualius objektus atskirti jau nepavyksta;
- sistemos apmokymo duomenys turėtų kuo mažiau priklausyti nuo to, kur sistema po to bus naudojama t.y. geriausia, jei sistemą užtektų apmokyti vieną kartą ir būtų galima naudoti kuo didesnei atvejų įvairovei. Tačiau čia susiduriama su skirtingo fono problema ar skirtingomis aplinkos sąlygomis;
- dažniausiai praeivius reikia skaičiuoti visame pastate ar bent keliuose praėjimuose (pavyzdžiui dideliame pastate su daug parduotuvių), todėl kaupiasi labai daug duomenų apdorojimui ir darosi sunku pasiekti realaus laiko apdorojimą. Jei darbas vyksta su vaizdais, tai sistema negali skirti vieno vaizdo apdorojimui daugiau laiko nei kad tą vaizdą užtruko paimti iš jutiklio;
- įrenginio kainą labai lemia jutiklių ir skaičiavimo įrangos sudėtis. Dažnai naudojamos termininės kameros leidžia pasiekti tikrai labai gerų rezultatų, tačiau jos yra pernelyg brangios, kad tokie įrenginiai būtų taikomi masiškai. Tačiau paprasti jutikliai, ypač barjeriniai, nors dažnu atveju pigesni, tačiau ir nepatogūs ir netinka didesniems srautams;

- kai kurios aplinkos yra labai kintančios laike, pavyzdžiui liftas ar lauko erdvės. Tokiu atveju reikia pasiekti, kad jutikliai arba algoritmai būtų kuo mažiau jautrūs aplinkos sąlygų pasikeitimams;
- vaizdo skenavimas sunaudoja labai daug skaičiavimo pajėgumų, todėl jo bandoma išvengti ir ieškomi kitokie sprendimai, antraip sunku pasiekti realaus laiko skaičiavimų;
- vengiama, kad vieni žmonės užstotų kitus, pavyzdžiui kamera kabinama virš durų. Praktiškai su visais jutikliais iškyla ta pati problema, kai vienas žmogus vienaip ar kitaip užstoja kitą ar kitus. Negalima versti žmonių eiti „tvarkingai“, todėl bandoma įvairiai „gudrauti“. Pavyzdžiui kamera kabinama virš praėjimo arba naudojamas „stereo“ vaizdas. Net ir tai neišsprendžia visų problemų, nes žmonės gali susigrūsti ar eiti apsikabinę;
- daugelis sistemų skirstomos į galinčias veikti tik vidaus sąlygomis ir tas, kurios gali veikti išorės sąlygomis. Išorės sąlygos paprastai būna gerokai sudėtingesnės. Jutikliai turi būti atsparūs šilumos pokyčiams, bei dideliems šilumos skirtumams. Sistema turi išlikti tiksli nepriklausomai nuo apšvietimo intensyvumo, metų laiko, dienos meto, oro sąlygų. Gali lyti, snigti, kristi dideli ir ryškūs šešėliai, sklisti įvairūs oro srautai, o sistema turi likti tiksli. Todėl lauko sąlygomis dažniau naudojami barjeriniai skaičiuokliai ar infraraudonųjų spindulių kameros;
- labai daug lemia žmonių skirtingumas. Peržvelgus pavyzdžius akivaizdu, kad variantų kiekis yra begalinis. Gali skirtis apranga, plaukų spalva, tekstūra, ūgis, judėjimo sparta. Žmogus gali praeidamas pasižiūrėti į viršų, pasisukti ar pasilenkti;
- yra sąlygų diapazonas, kuomet bet kuris jutiklis apanka. Kiekvienam jutikliui šios sąlygos yra skirtingos. Tais atvejais elgiamasi dvejopai, arba bandoma pakeisti tas sąlygas arba apskritai tomis sąlygomis įrenginys neturėtų būti naudojamas. Su vaizdais dažniausia problema yra apšvietimas, kuris ypač nepastovus lauko sąlygomis. Gali įtakoti net paprasčiausias rūkas;
- pati didžiausia visų sistemų problema, kad jos negeba susidoroti su ypatingomis situacijomis, kurių gali būti labai daug. Kartais žmonės sustoja tarpduryje ir kurią laiką ten stovi, kartais praeina pro šalį arba nuo tarpdurio vidurio grįžta atgal. Gana komplikota tampa, jei pro praėjimą gali judėti ir kitokie objektai. Galbūt kažkas vedasi dviratį, neša ar veža krepšius, pralekia šuo, žaidžia vaikai. Kaip

elgtis, jei atsiranda norinčių išbandyti sistemą? Jie moja ranka ties jutikliais, daug kartų praeina pro praėjimą pirmyn ir atgal. Tokių nestandartinių situacijų daug ir jas sunku eliminuoti;

- tikslumo apribojimas yra gana svarbus. Vieni sakosi, kad jų sistemos skaičiuoja 95-100 procentų tikslumą, kiti kiek mažiau. 90 procentų jau yra gana geras variantas, tačiau visais atvejais, kiekviena sistema turi papildomų sąlygų, o demonstraciniai pavyzdžiai dažniausiai yra kruopščiai parinkti, kad sistema atrodytų tikslesnė nei iš tikrųjų yra;
- ne visos sistemos ir sprendimai tinka bet kokiems praėjimams. Praėjimai labiausiai skiriasi pločiu, kuris įtakoja kiek žmonių vienu metu gali praeiti. Mažesniuose praėjimuose srautas bus mažesnis, o didesniuose laisvai gali prasilenkti po kelis pėsčiuosius.

Apribojimų be abejonės yra ir daugiau, tačiau šie yra pagrindiniai, kuriuos vis dar bando išspręsti dirbantys šioje srityje.

1.4. Taikymo sritys

Žmonių skaičiavimas arba monitoringas dažniausiai vienaip ar kitaip siejasi su viešomis erdvėmis, viešuoju saugumu arba srautų optimizavimu. Panagrinėjus literatūrą galima rasti neįtikėtinais daug ir įvairių pritaikymo galimybių. Nenaudojant jokių įrengimų, skaičiuoti praeivių srautus yra be galo brangu, nes tuomet tai turi daryti žmonės, kurie vienu metu gali aprėpti vos vieną, galbūt du praėjimus. Be to tikslumas prie didesnių srautų tikrai nebus geresnis nei mažiau tiksli sistema.

Taikant pėsčiųjų skaičiavimo sistemą gaunama informacija gali suteikti labai vertingos informacijos ir įmanoma:

- surinkti realaus laiko praeivių kiekius bei statistiką pagal laiką;
- prognozuoti būsimų praeivių kiekį pagal praeities statistiką;
- išmatuoti ir palyginti praėjimų apkrautumą;
- tiksliau planuoti ir derinti resursus priklausomai nuo apkrautumo;
- mažinti kaštus, kai apkrovimai maži;
- palyginti skirtingus praėjimus pagal srautą;
- žinoti kiek žmonių šiuo metu yra pastate ar patalpoje;
- aptikti ir sumažinti eiles ar kamščius.

Objektai kuriems galėtų būti pritaikoma: parduotuvės, koncertai, stadionai, barai, bibliotekos, prekybos centrai, kazino, klubai, supermarketai, muziejai, oro uostai, traukiniai, autobusai, bankai, galerijos ar bet kokie kitokie objektai turinys praėjimus.

Be abejonės yra daug galimybių taikyti lauko sąlygomis, pavyzdžiui įjungiamas raudonas šviesoforas, jei yra einančių per perėją. Be to, gali būti skaičiuojami ne tik žmonės, bet ir transporto priemonės kelyje, dviračiai, kitokie objektai. Informacija apie dominančių objektų srautus gali padėti racionaliau tuos srautus paskirstyti.

1.5. SWOT analizė

Srauto monitoringo taikymo viešajam transportui analizė atlikta remiantis sistemų teorijos paskaitų medžiaga [21].

1.5.1. SWOT (SSGG) analizės faktoriai

Faktoriai yra vidinės ir išorinės įtakos suskirstytos į keturias kategorijas: stiprybės, silpnybės, galimybės ir grėsmės, arba tiesiog SSGG.

Išorė mūsų atveju yra visuomenė, o sistemos įgyvendinimas jai suteiktų tam tikrų galimybių, tačiau sukeltų ir grėsmių (žr. 1 lentelę). Šių išorinių sąlygų įtakoti negalima, bet įvertinti jų įtaką bendrai projekto sėkmei yra naudinga.

Vidinės įtakos yra autoriaus silpnybės ir stiprybės kuriant šią sistemą. Šios įtakos apsunkina arba palengvina projekto įgyvendinimą. Jeigu išorinių įtakų mes keisti negalime, tai vidines koreguoti pavyktų, todėl atlikę SSGG analizę, turėtume matyti, kurias stiprybes verta didinti, o kokias silpnybes verta mažinti.

1 lentelė. SWOT (SSGG) analizės faktoriai

Galimybės	GL1 - išvengiamos žmonių spūstys (optimalūs maršrutai); GL2 - ekonomiškai naudojamos transporto priemonės; GL3 - sutaupyta lėšos (neboreikia samdyti žmonių, kurie atliktų šį darbą); GL4 - platus pritaikymas (žmonių monitoringas kitokuose objektuose).
Grėsmės	GR1 - reti maršrutai (jei mažai apkrauti); GR3 - neadekvatūs rezultatai (keleiviai gali pažeisti monitoringo sistemą).
Stiprybės	ST1 - patirtis dirbant su panašiais projektais; ST2 - lengva paruošti duomenis; ST3 - daug laiko.
Silpnybės	SL1 - menkas tikslumas skaičiuojant žmonių srautus; SL2 - patirties stoka (su šiuo metodu); SL3 - lėtas metodas.

Projekto vystymo eigoje gali atsirasti naujų faktorių, kurie turės didelę įtaką, todėl projekto kūrimo metu reiktų atnaujinti tiek faktorius, jei jie keitėsi, tiek ir pačią analizę, įvertinant naują patirtį.

1.5.2. Faktorių analizė su WinSWOT

Norint atlikti faktorių analizę [22], pirmiausia juos reikia paversti kiekybiniais dydžiais išreiškiamais ta pačia vertinimo skale. Todėl atlikdami analizę sudarysime faktorių įvertinimo lenteles.

Kiekvienas faktorius vertinamas pagal tai, kiek esame dėl jo tikri, kad taip ir bus (**tikrumas**) ir pagal tai, kaip jis įtakoja projektą (**įtaka**). Faktoriaus tikrumas ir įtaka išreiškiami realiais skaičiais [0;1] skalėje, pavyzdžiui „0,4“.

Faktoriai vertinami laike, t.y. momentinės vertės, projektui įpusėjus ir sistemai įdiegus. Tai svarbu, nes faktorių reikšmės gali gana stipriai kisti projekto eigoje ir mums rūpi ne tik trumpalaikė, bet ir ilgalaikė sėkmė.

Pirmiausia sudaromi galimybių įverčiai:

2 lentelė. Galimybių įverčiai

Galimybės	Momentinės verės		Trumpalaikės vertės		Ilgalaikės vertės	
	Tikrumas	Įtaka	Tikrumas	Įtaka	Tikrumas	Įtaka
GL1	0,7	0,4	0,5	0,6	0,4	0,8
GL2	0,5	0,3	0,3	0,5	0,2	0,6
GL3	0,8	0,7	0,6	0,8	0,5	0,9
GL4	0,7	0,3	0,5	0,4	0,4	0,5

Visi įverčiai yra įvertinti pagal jų svarbumą („importance“) ir pažymėti skirtingomis spalvomis (žr. galimybių verčių grafiką esantį žemiau). Svarbumas paskaičiuojamas kiekvienam faktoriui atskirai pagal pirmą formulę.

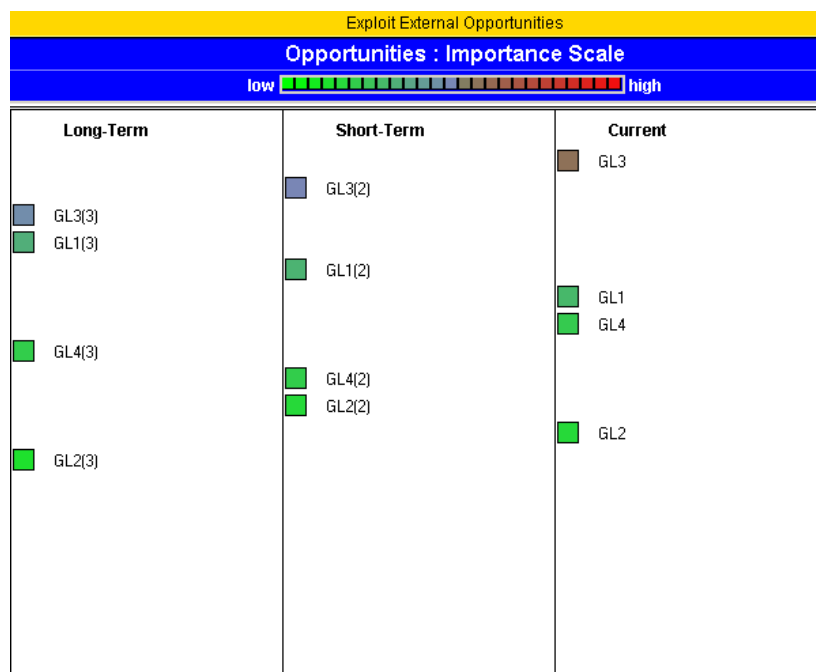
$$\text{Svarbumas} = \text{Tikrumas} \times \text{Įtaka} \quad (1)$$

Kaip matome pati svarbiausia galimybė yra „GL3“, o mažiausiai svarbi „GL2“.

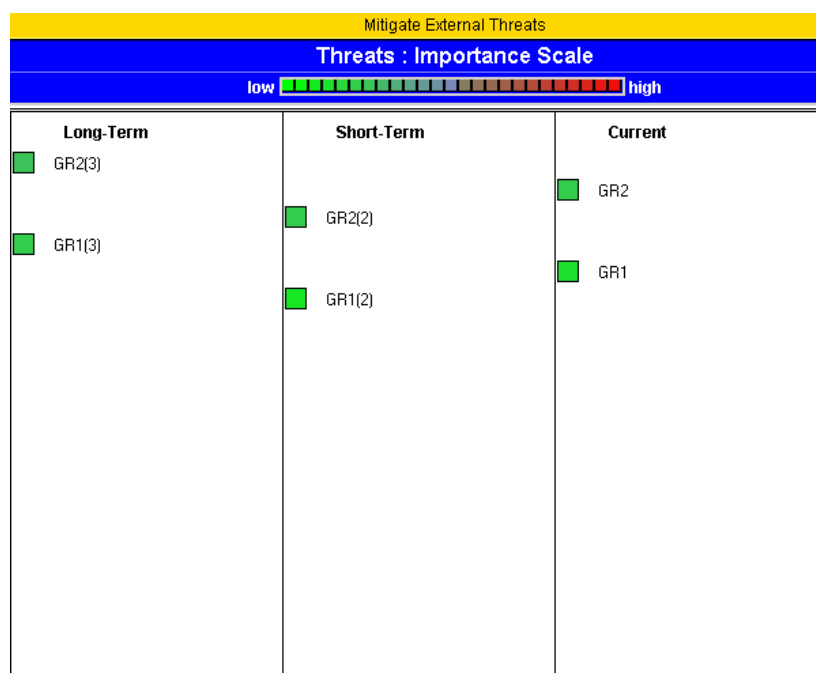
Analogiškai gaunami grėsmių, stiprybių ir silpnybių įverčiai:

3 lentelė. Grėsmių įverčiai

Grėsmės	Momentinės verės		Trumpalaikės vertės		Ilgalaikės vertės	
	Tikrumas	Įtaka	Tikrumas	Įtaka	Tikrumas	Įtaka
GR1	0,6	0,2	0,5	0,2	0,4	0,5
GR4	0,5	0,4	0,4	0,5	0,4	0,6



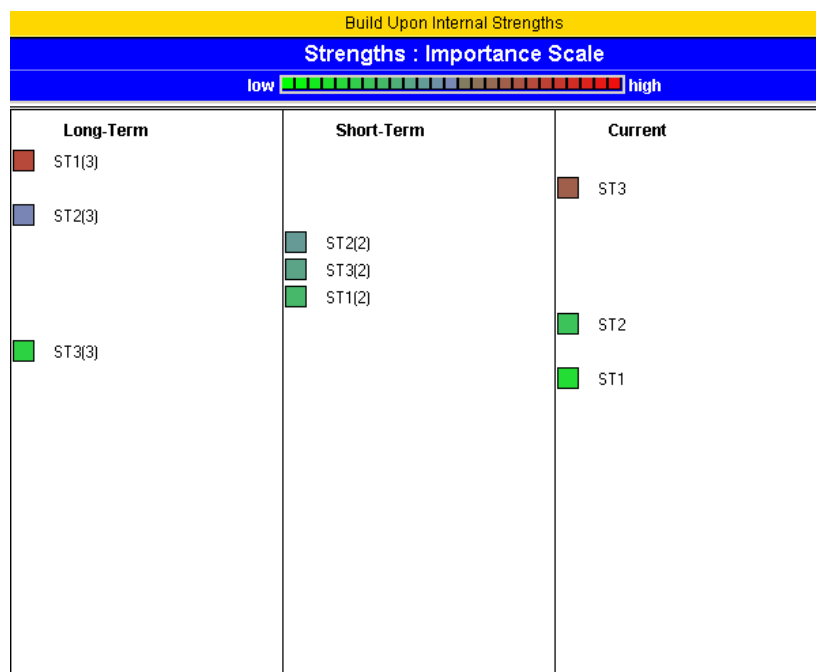
12 pav. Galimybių įverčiai



13 pav. Grėsmių įverčiai

4 lentelė. Stiprybių įverčiai

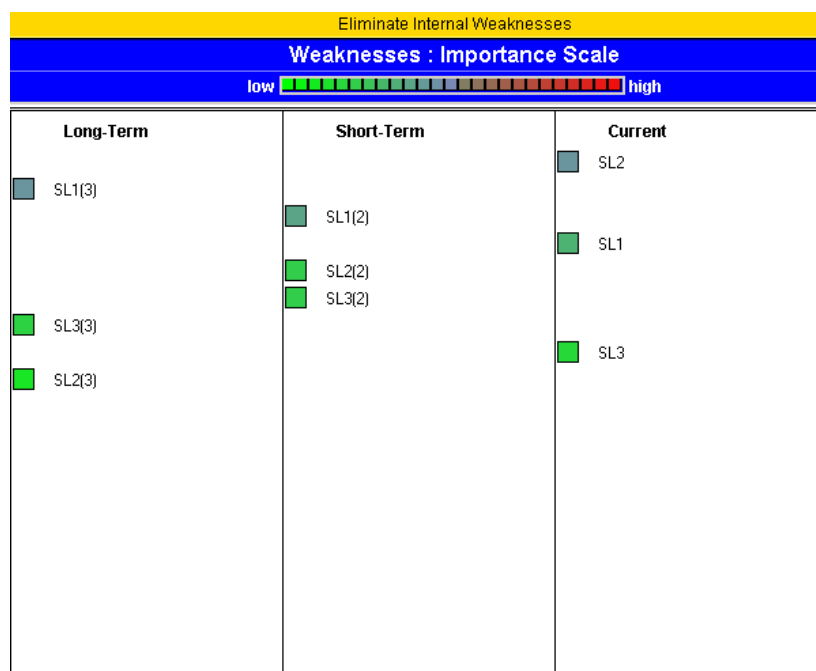
Stiprybės	Momentinės vertės		Trumpalaikės vertės		Ilgalaikės vertės	
	Tikrumas	Įtaka	Tikrumas	Įtaka	Tikrumas	Įtaka
ST1	0,8	0,5	0,8	0,6	0,8	0,7
ST2	0,5	0,3	0,5	0,5	0,5	0,6
ST3	0,7	0,7	0,6	0,5	0,6	0,3



14 pav. Stiprybių įverčiai

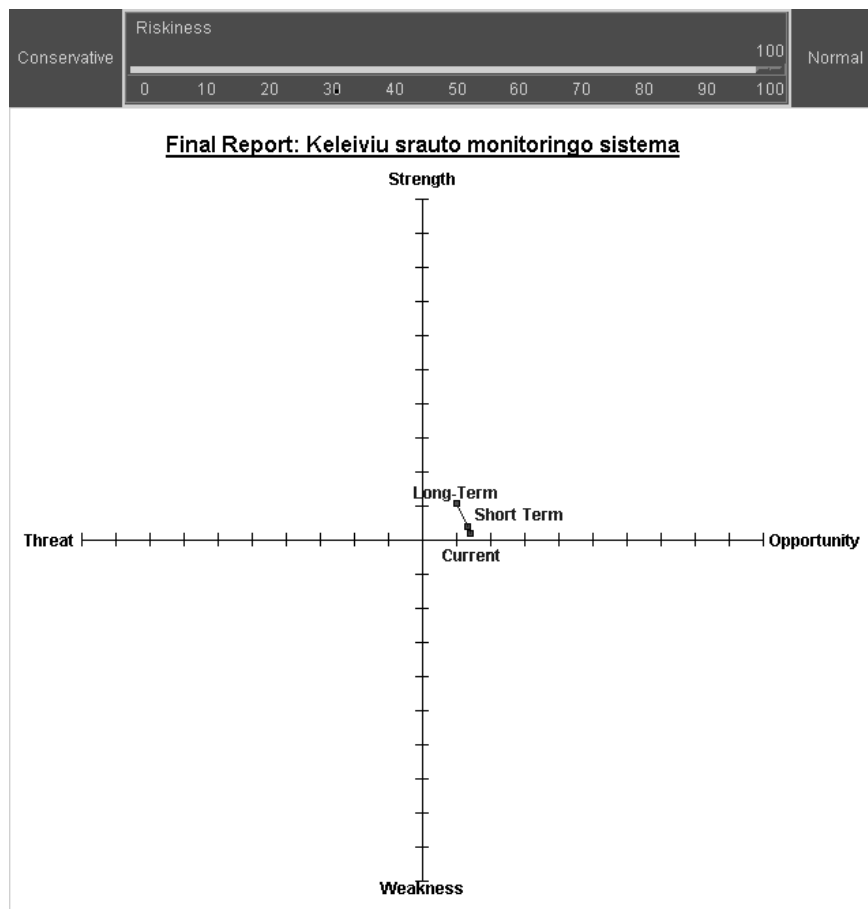
5 lentelė. Silpnųjų įverčiai

Silpnybės	Momentinės vertės		Trumpalaikės vertės		Ilgalaikės vertės	
	Tikrumas	Įtaka	Tikrumas	Įtaka	Tikrumas	Įtaka
SL1	0,6	0,5	0,6	0,6	0,6	0,7
SL2	0,7	0,6	0,5	0,4	0,5	0,2
SL3	0,5	0,3	0,4	0,5	0,3	0,6



15 pav. Silpnųjų įverčiai

Galutinė WinSWOT ataskaita:



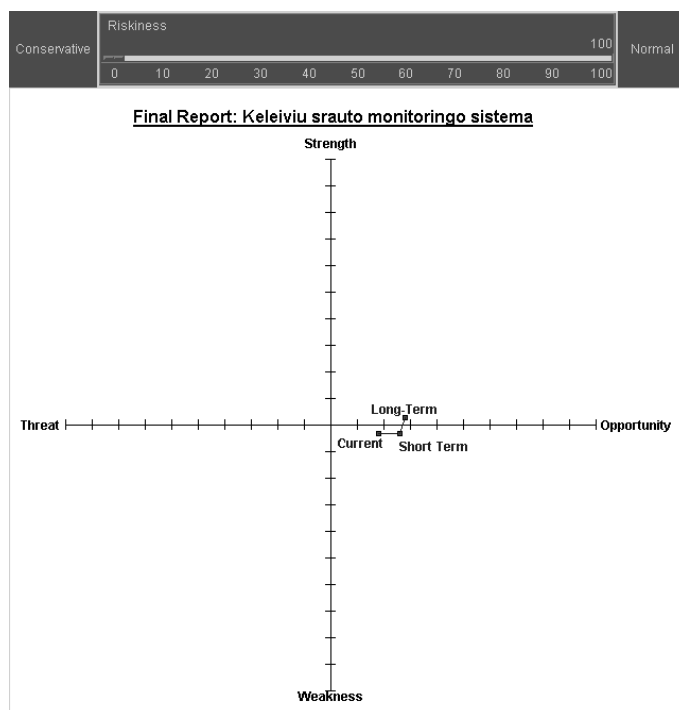
16 pav. SSGG grafikas su normalia rizika

Grafike atidėti trys taškai, kurie paskaičiuojami pagal antrą formulę kiekvienam taškui atskirai.

$$\text{Stiprybės/Silpnybės/Galimybės/Grėsmės} = \text{Sum (Tikrumas x Įtaka)} \quad (2)$$

Pagal gautą grafiką matome, kad pradžioje galimybės išnaudojamos labiau nei ilguoju laikotarpiu, o einant laikui stiprybės gerokai išauga, silpnybės sumažėja.

Grafike galime keisti rizikos laipsnį. Normaliu atveju tikrumas ir įtaka vertinami vienodai, tačiau skaičiuojant konservatyvų požiūrį, tikrumas vertinamas mažiau nei įtaka. Tokiu būdu rezultatai mažiau priklauso nuo tikrumo ir labiau nuo įtakos. Kai rizika yra nulis, tuomet laikoma, kad tai pats blogiausias scenarijus.



17 pav. Grafikas su didžiausia rizika (blogiausias atvejis)

Pagal rezultatus su didžiausia rizika, tikėtina, kad ilgalaikėje perspektyvoje galimybės bus panaudotos labiausiai. Tačiau ryškiai sumenksta stiprybių ir silpnųjų skirtumas.

1.5.3. Faktorių analizė su Excel

Pagal per paskaitas duotas formules (žr. 3-4 formules) užpildome momentines, trumpalaikes ir ilgalaikes vertes (žr. 6-8 lenteles).

$$OP_{\Sigma} = \sum_{o=1}^O \left\{ c_o \left(\mu_o + \sum_{s=1}^S ST_{os} + \sum_{w=1}^W WK_{ow} \right) \right\} \quad (3)$$

$$TH_{\Sigma} = \sum_{t=1}^T \left\{ c_t \left(\mu_t + \sum_{s=1}^S ST_{ts} + \sum_{w=1}^W WK_{tw} \right) \right\} \quad (4)$$

6 lentelė. Momentinės vertės

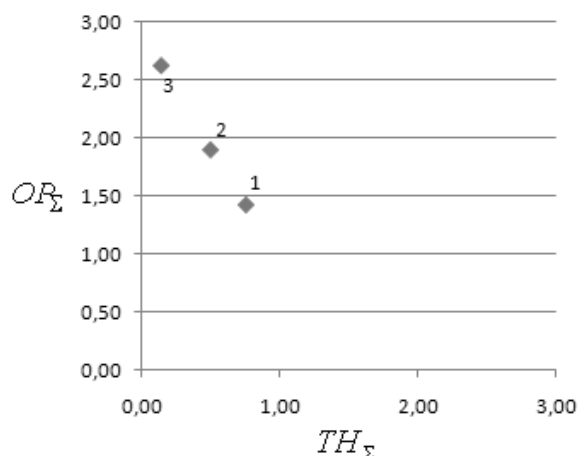
		Tikrumas	Išlaidos	Stiprybės			Silpnybės			Suma
				ST1	ST2	ST3	SL1	SL2	SL3	
Galimybės	GL1	0,7	0,4		0,3	0,4	-0,6	-0,5	0,120	
	GL2	0,5	0,3	0,1	0,5		-0,7	-0,6	-0,060	
	GL3	0,8	0,7		0,7	0,8		-0,6	1,190	
	GL4	0,7	0,3	0,4	0,4		-0,4	-0,5	0,180	
Bendroji suma:									1,430	
Grėsmės	GR1	0,6	0,2	-0,2			0,2		0,120	
	GR2	0,5	0,4	-0,2	-0,1	-0,2	0,5	0,4	0,7	0,640
Bendroji suma:									0,760	

7 lentelė. Trumpalaikės vertės

		Tikrumas	Įtaka	Stiprybės			Silpnybės			Suma
				ST1	ST2	ST3	SL1	SL2	SL3	
Galimybės	GL1	0,5	0,6		0,4	0,2	-0,4		-0,3	0,240
	GL2	0,3	0,5	0,2	0,6		-0,5	-0,4		0,100
	GL3	0,6	0,8		0,8	0,4			-0,4	1,120
	GL4	0,5	0,4	0,6	0,5		-0,2		-0,3	0,440
Bendroji suma:										1,900
Grėsmės	GR1	0,5	0,2	-0,2			0,2			0,100
	GR2	0,4	0,5	-0,4	-0,3	-0,1	0,4	0,3	0,5	0,400
Bendroji suma:										0,500

8 lentelė. Ilgalaikės vertės

		Tikrumas	Įtaka	Stiprybės			Silpnybės			Suma
				ST1	ST2	ST3	SL1	SL2	SL3	
Galimybės	GL1	0,4	0,8		0,5	0,1	-0,3		-0,2	0,400
	GL2	0,2	0,6	0,3	0,7		-0,4	-0,3		0,300
	GL3	0,5	0,9		0,9	0,2			-0,3	1,170
	GL4	0,4	0,5	0,8	0,6		-0,1		-0,2	0,750
Bendroji suma:										2,620
Grėsmės	GR1	0,4	0,5	-0,2			0,2			0,200
	GR2	0,4	0,6	-0,6	-0,7	-0,1	0,3	0,2	0,4	-0,060
Bendroji suma:										0,140

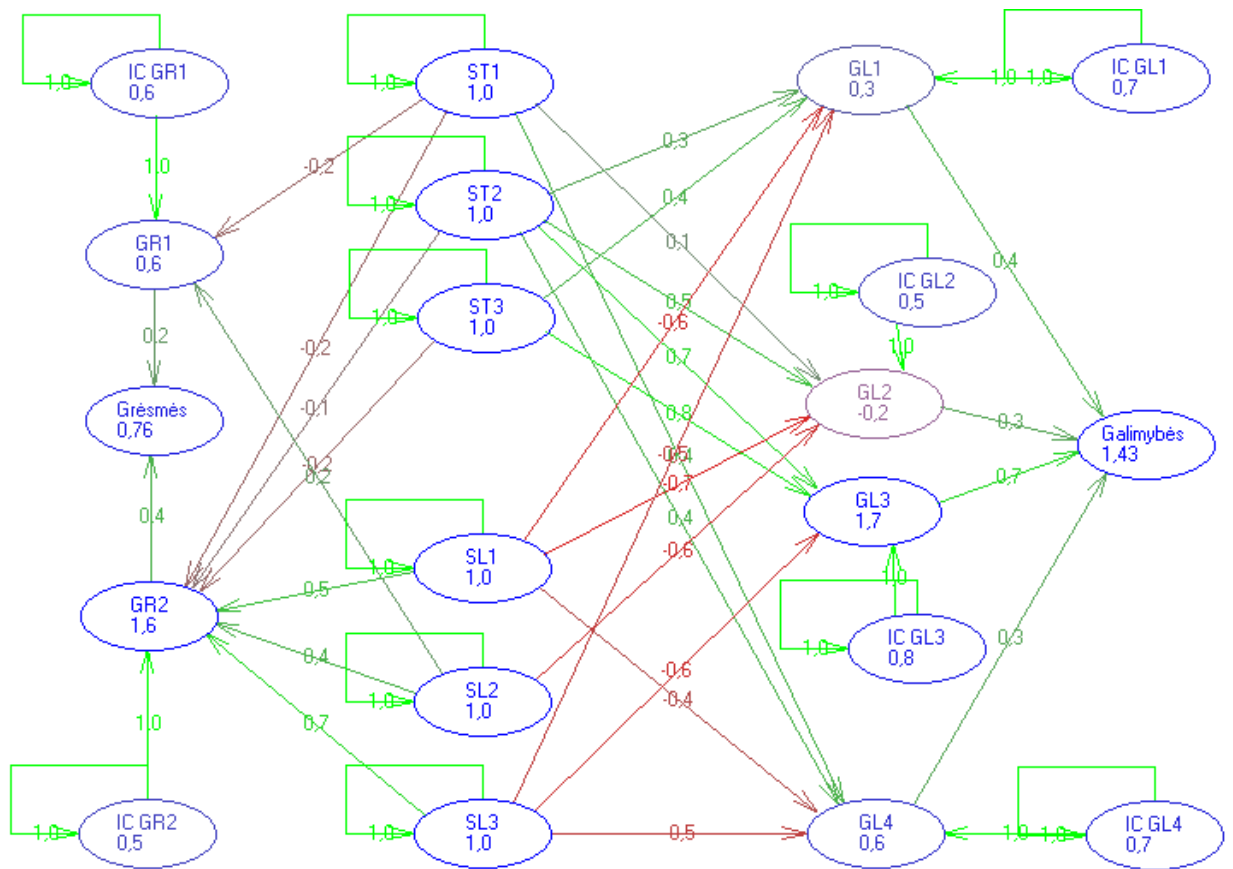


18 pav. Galimybės ir grėsmės momentiniu (1), trumpuoju (2) ir ilgoju (3) laikotarpiu

Gauname, kad galimybės laikui einant didėja, o grėsmės mažėja. Tai rodo, kad projektas turi geras ilgalaikes perspektyvas.

1.5.4. Miglotojo pažintinio kelio (MPP) sudarymas

Naudosime programą [23] MPP sudarymui. Suvedame momentines reikšmes, sugeneruojame MPP ir pakoreguojame gautą tinklą. Randame nusistovėjusį MPP:



19 pav. Nusistovėjęs MPP

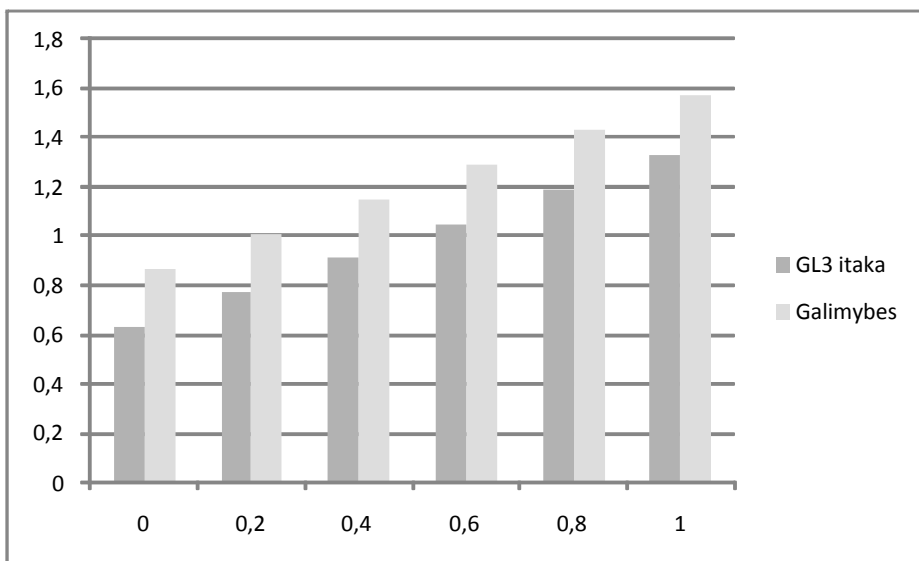
Matome, kad galimybių ir grėsmių sumos („1,43“ ir „0,76“) yra lygios toms, kurias gavome skaičiuojant Excel programoje (žr. 6 lentelę).

1.5.5. Galimybių didinimas ir grėsmių mažinimas

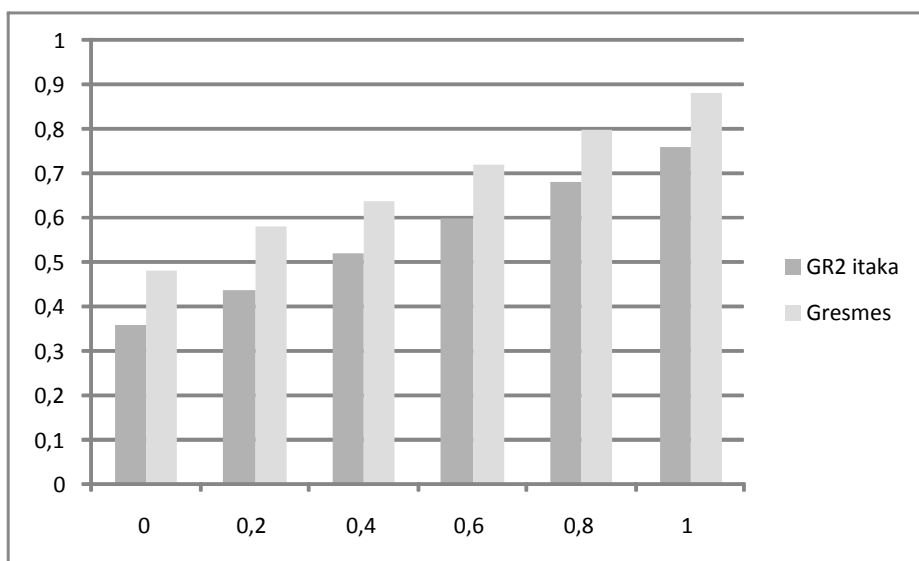
Didžiausią teigiamą įtaką projektui daro „GL3“, o didžiausią neigiamą - „GR2“ (matome iš 1-2 paveikslukų). Tuomet reiktų stengtis, kad „GL3“ būtų didinama, o „GR2“ mažinama. Tam nustatome kurios stiprybės ir silpnybės labiausiai šias reikšmes lemia. „GL3“ labiausiai įtakoja „ST3“, o „GR2“ – „SL3“.

Nubraižome keturis grafikus tikrindami kaip abu pasirinkti faktoriai veikia grėsmes arba galimybes:

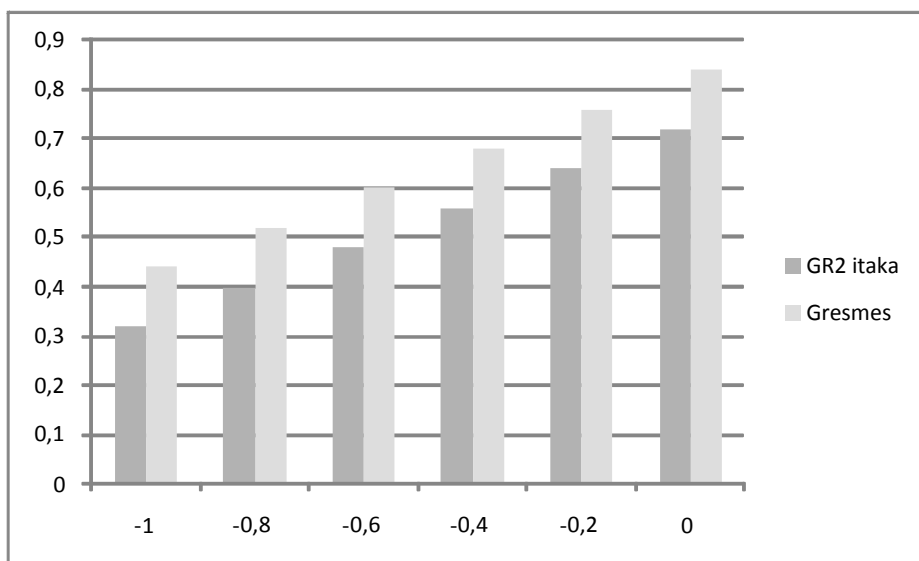
- „ST3“ poveikis „GL3“ ir galimybių sumai (žr. 20 paveiksluką);
- „SL3“ poveikis „GR2“ ir grėsmių sumai (žr. 21 paveiksluką);
- „ST3“ poveikis „GR2“ ir grėsmių sumai (žr. 22 paveiksluką);
- „SL3“ poveikis „GL3“ ir galimybių sumai (žr. 23 paveiksluką).



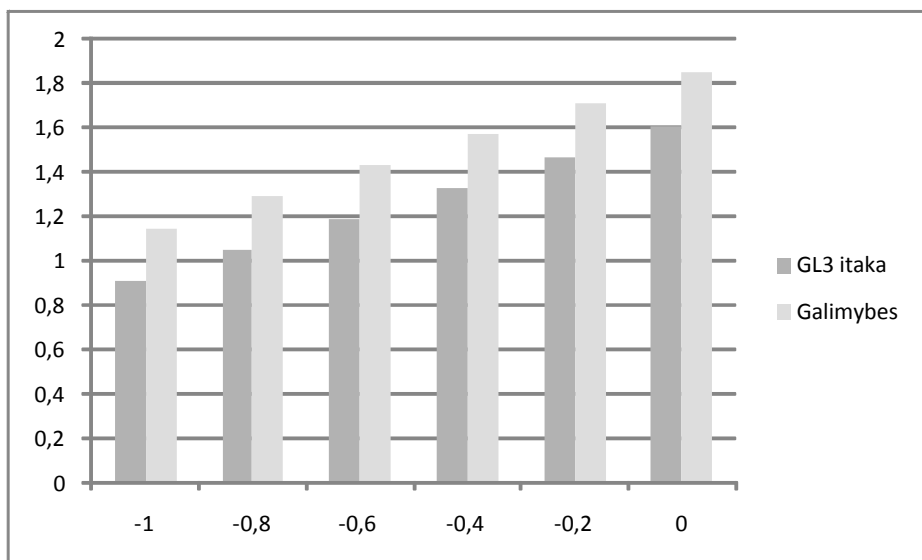
20 pav. ST3 vertės poveikis GL3 įtakai ir visų galimybių sumai



21 pav. SL3 vertės poveikis GR2 įtakai ir visų grėsmių sumai



22 pav. ST3 vertės poveikis GR2 įtakai ir visų grėsmių sumai

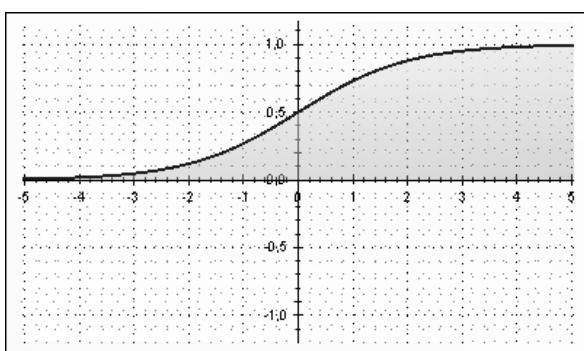


23 pav. SL3 vertės poveikis GL3 įtakai ir visų galimybių sumai

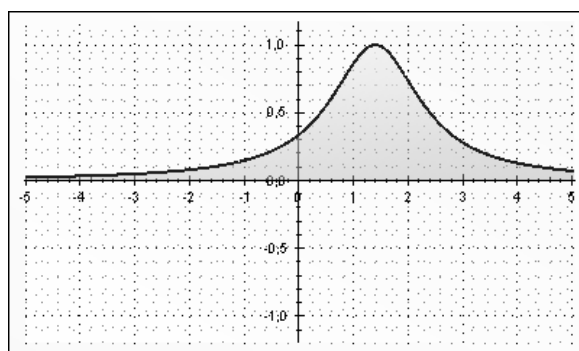
Kaip matome, priklausomybės yra tiesinės. Jei keičiame atitinkamą silpnybę ar stiprybę, sumos reaguoja tiesiškai ir nedelsiant.

„SL3“ poveikis yra didesnis už „ST3“, todėl mažinant silpnybę pasiekiamas efektas bus didesnis. Tačiau abiejų faktorių įtakos santykinai su kitomis yra labai didelės, todėl verta dirbti su abejomis.

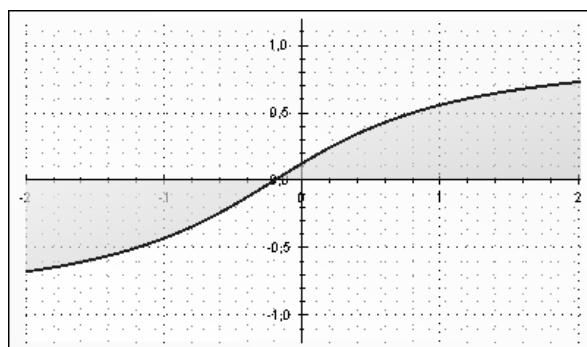
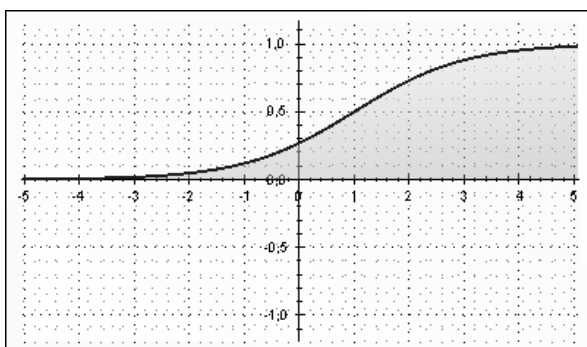
1.5.6. Mazgų reakcijos funkcijos

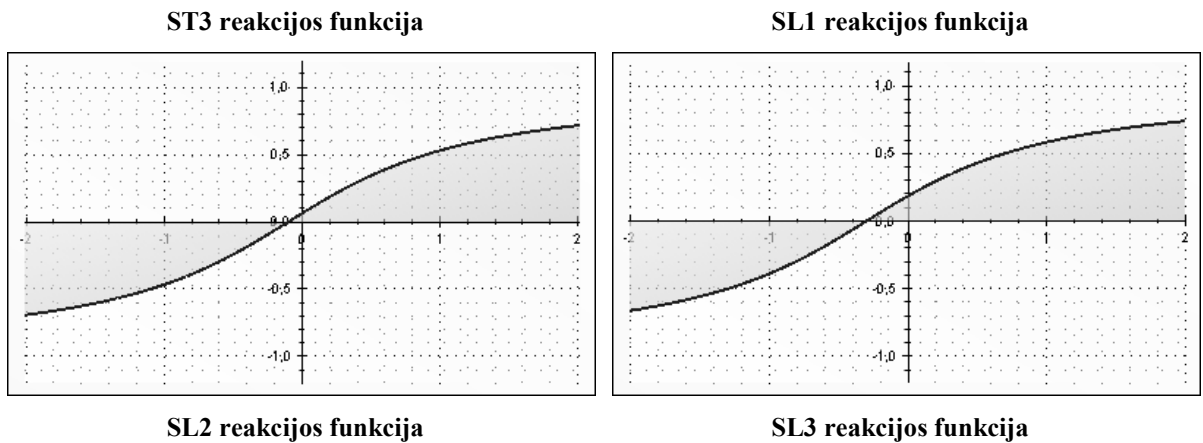


ST1 reakcijos funkcija



ST2 reakcijos funkcija



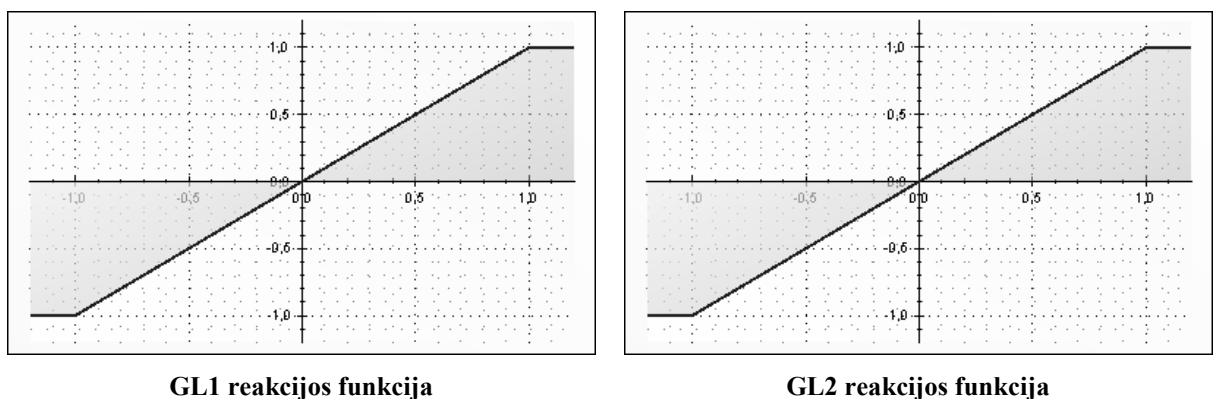


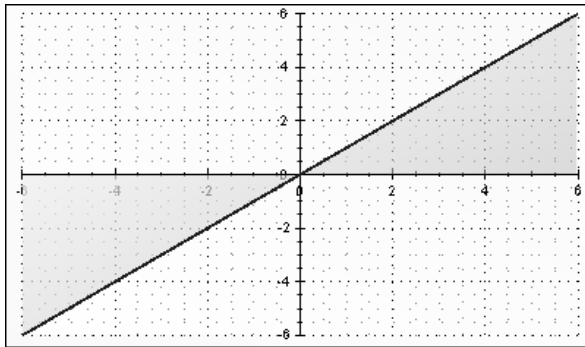
24 pav. Stiprybių ir silpnybių mazgų reakcijos funkcijos

Tik „ST2“ stiprybė laikui einant didėja, nes duomenų prikaupti iš karto neįmanoma, tai tik su laiku. Kuo daugiau turima duomenų, tuo lengviau realizuoti sistemą, todėl įtaka auga.

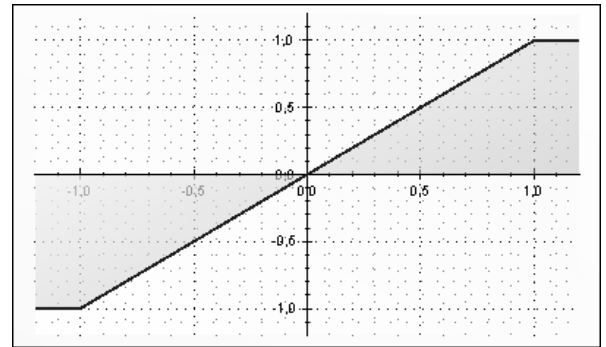
„ST1“ ir „ST3“ didžiausias poveikis yra projekto pradžioje. Turima patirtis su kitais projektais leidžia lengviau pradėti darbus, tačiau ilginiui turimos patirties įtaka sumažės. „ST3“ analogiškai, projekto pradžioje laiko yra daug, projektui įpusėjus dažniausiai atsiranda kitų darbų, kuriuos taip pat reikia atlikti, o projekto gale visada laiko būna labai mažai. Neretai įdiegtas sistemas prižiūri kiti žmonės, nei jas sukūrė, nes autoriai imasi kitų darbų.

Silpnybių reakcijos labai panašios, tačiau skiriasi jų mažėjimo sparta. Pvz., „SL2“, patirties stoka su šiuo metodu, mažėja, nes projekto eigoje daug išmokstama. Metodo tikslumas ir sparta bus labiausiai patobulinti iki projekto vidurio, o po to mažai keisis.

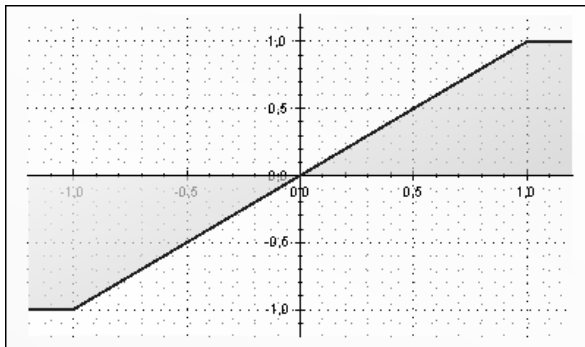




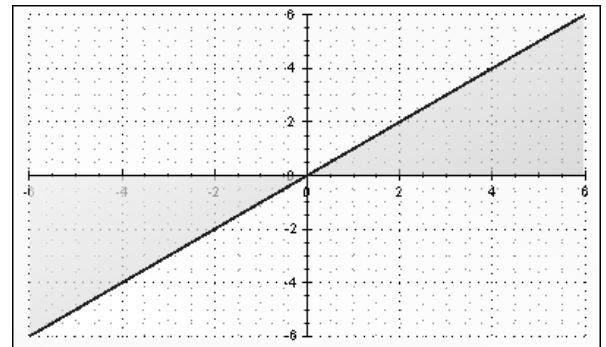
GL3 reakcijos funkcija



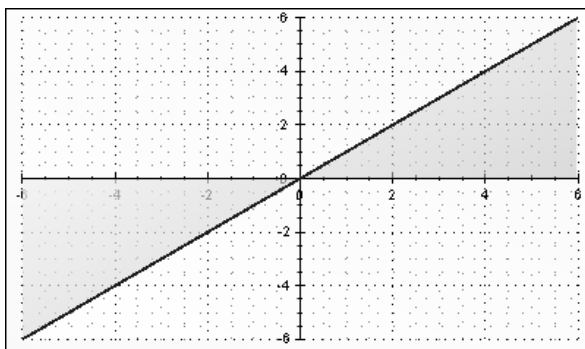
GL4 reakcijos funkcija



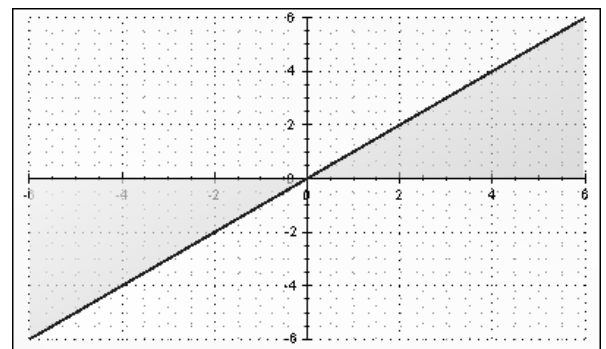
GR1 reakcijos funkcija



GR2 reakcijos funkcija



Galimybių reakcijos funkcija



Grėsmių reakcijos funkcija

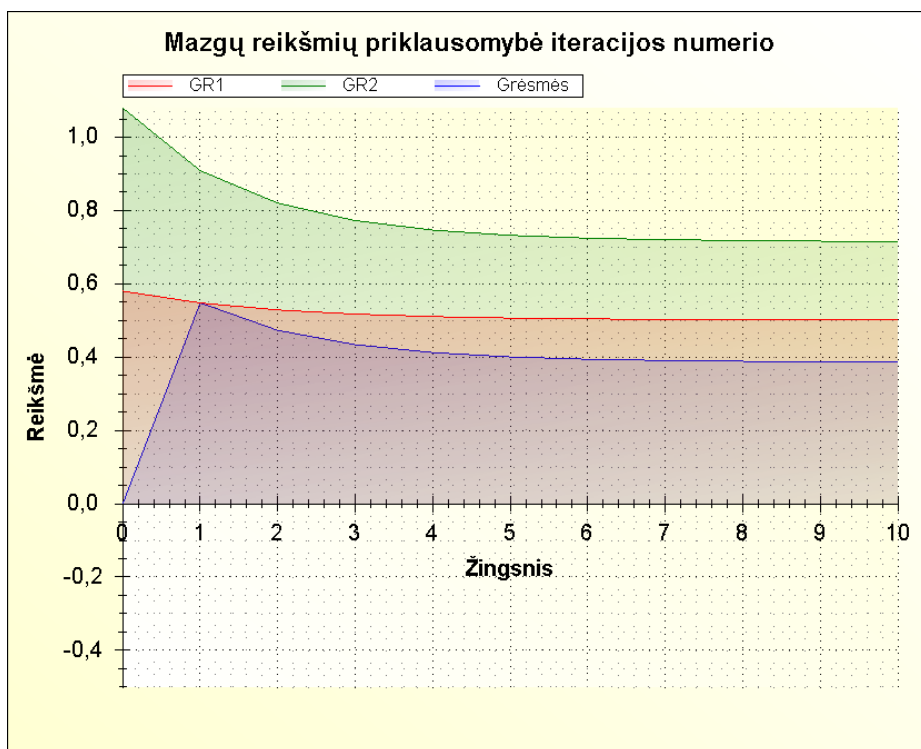
25 pav. Galimybių ir grėsmių mazgų reakcijos funkcijos

„GL3“ ir „GR2“ reikšmės perkopia vieneta, todėl ten nurodyta tiesinė priklausomybė.

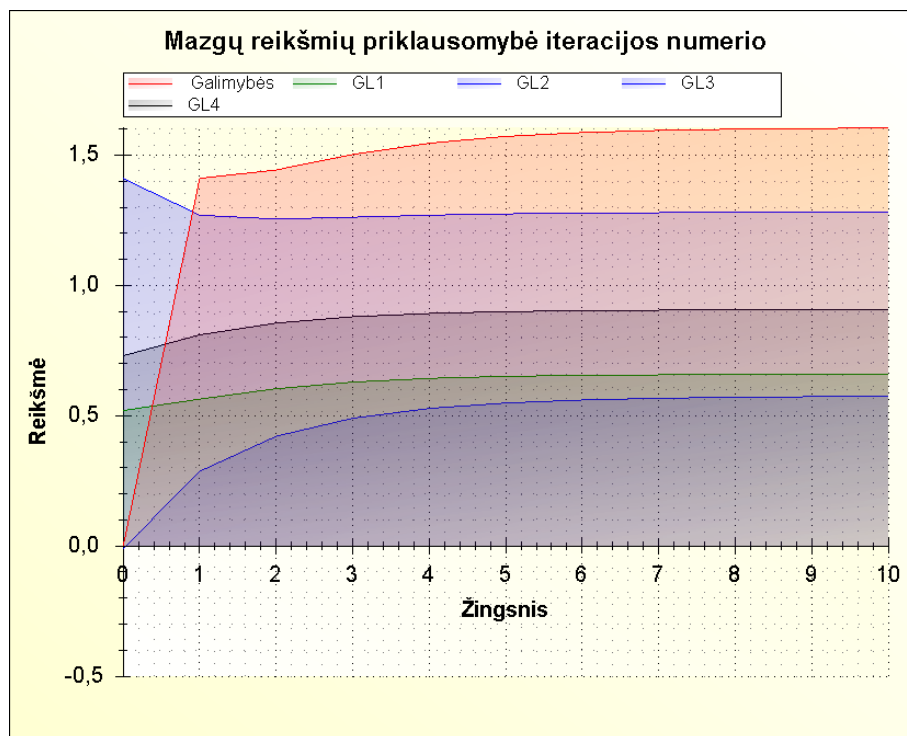
Visų galimybių ir grėsmių reakcijos funkcijos parinktos tiesinės, nes jos tiesiogiai priklauso nuo silpnybių ir stiprybių. Norint atlikti detalesnę analizę, reiktų gerokai atidžiau įvertinti šių faktorių vėlinimą bei inerciją. Atliktų eksperimentų tikslas buvo pažiūrėti kaip stiprybės ir silpnybės tiesiogiai įtakoja paskiras ir sumines galimybes bei grėsmes.

1.5.7. Dinaminė analizė

Nustatome mazgų charakterio funkcijas ir surandame nusistovėjusį MPP.



26 pav. Grėsmių priklausomybė nuo iteracijų

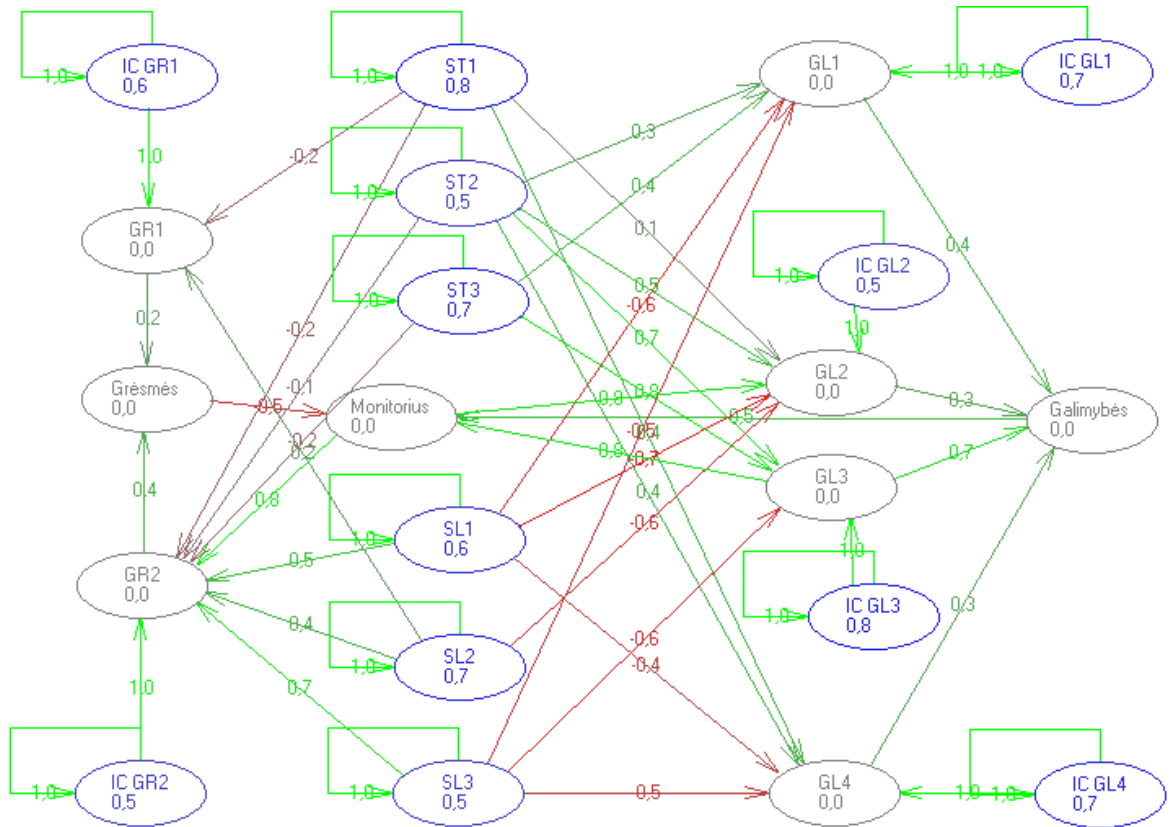


27 pav. Galimybių priklausomybė nuo iteracijų

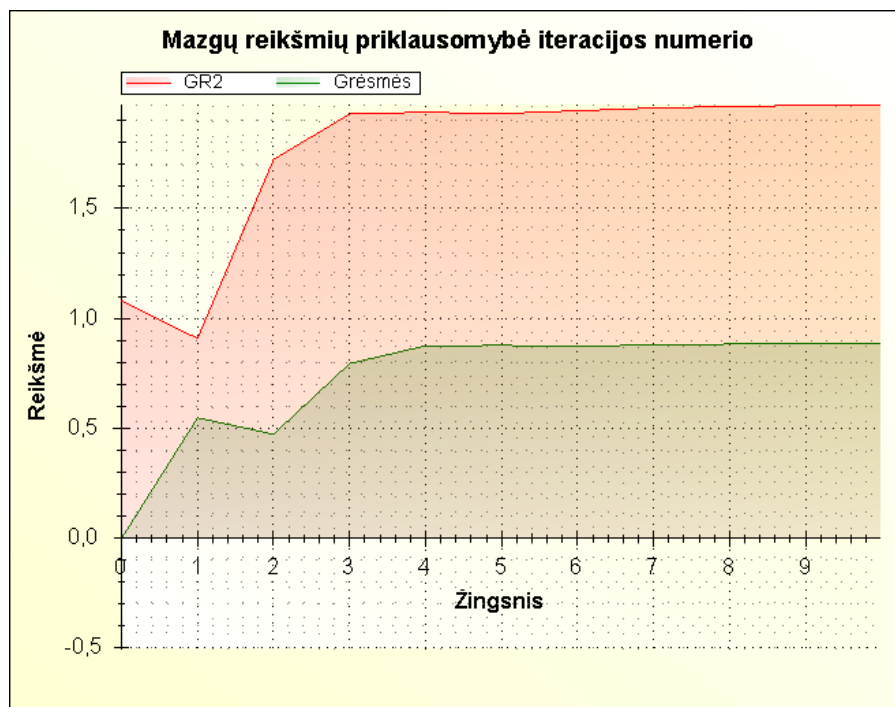
Gauname labai panašius rezultatus į SWOT analizės, kuomet galimybės laikui einant šiek tiek išauga, o grėsmės sumažėja.

1.5.8. Monitoringas

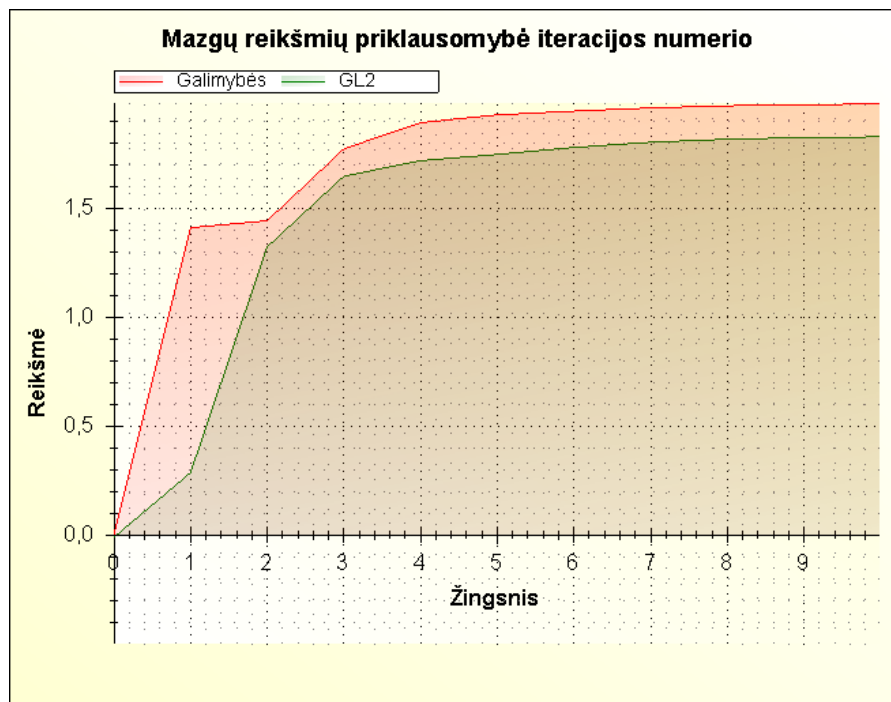
Monitoringą atliksime pasirinktinai „GR2“, „GL2“ ir „GL3“ mazgams. Sumines grėsmes nukreipiame į papildomą mazgą „Monitorius“. Tai grįžtamasis ryšys.



28 pav. Monitoringo schema



29 pav. Grėsnių pokyčiai



30 pav. Galimybių pokyčiai

Iš rezultatų matome, kad grįžtamasis ryšys grėsmėms turėjo neigiamą poveikį, nes nuo antros iteracijos bendras grėsmės lygis padidėjo, o po to nusistovėjo.

Galimybės nuo antros iteracijos irgi išaugo, tačiau mažiau nei grėsmės.

1.6. Analizės išvados

1. Tokių sistemų tikslumas ir nauda labai priklauso nuo pasirinktų jutiklių. Pigūs jutikliai nėra tikslūs, o brangūs dažniausiai jau neapsimoka. Todėl perspektyviausia būtų naudoti kiek įmanoma paprastesnes vaizdo kameras. Tuomet vertingiausia dalis būtų algoritmai gebantys paskaičiuoti pėsčiuosius. WEB kameros yra pačios pigiausios, dar galėtų tikti IR arba IP kameros, nes matymas nepriklausomai nuo apšvietimo kai kada labai svarbus, o prisijungimas nuotoliniu būdu leistų diegti papildomas paslaugas. Pavyzdžiui, įmonė išsinuomoja tokį pėsčiųjų skaičiuoklį ir gali bet kada internetu prisijungti prie svetainės, kurioje matytų visą statistiką laike;
2. Iš visų apžvelgtų sprendimų, panašu, kad racionaliausia jutiklius kabinti virš praėjimo. Tik tokiu atveju vieni praeiviai neužstoja kitų. Tačiau spūsties-susiliejiimo problema lieka ir ją reikia spręsti algoritmų pagalba;
3. Literatūroje nepavyko aptikti sprendimų, kurie bandytų panaudoti HTM algoritmus, todėl ši sritis dar visai netirta ir galbūt galima pasiekti įdomių rezultatų;

4. Sprendimas, kuris veiktų vidaus ir lauko sąlygomis, būtų pakankamai tikslus net jei žmonės grūdasi ir tiktų bet kokio pločio praėjimams būtų labai universalus, nes tiek apribojimų dar neišsprendė nei viena sistema;
5. Tokių sistemų pritaikomumas yra neįtikėtinai platus. Vienintelė problema kodėl jos taikomos retai yra ta, kad tikslios sistemos labai brangios arba nepatogios. Netgi palyginant su žmogaus darbo apmokėjimu, jos dar per brangios, tačiau pasiekus, kad būtų naudojami pigūs jutikliai ir tikslūs algoritmai, jos taptų ekonomiškai atsiperkančios per gana trumpą laiko tarpą;
6. Dažniausiai formuluojamas atpažinimo uždavinys, nes manoma, kad atpažinus objektus ir radus jų koordinates, skaičiavimo etapas tampa trivialus. Todėl visi metodai daugiau ar mažiau siejasi su atpažinimu ir filtravimu;
7. Visi rasti sprendimai pasiteisina tik dalinai, išsprendus vienas problemas, dėl metodų savybių atsiranda kitos;
8. SSGG ir MPP padėjo nustatyti, kurie faktoriai labiausiai lemia galutines galimybes ir grėsmes, bei paskaičiuoti jų įtaką. Galimybės laikui einant didėja, o grėsmės mažėja. Tai rodo, kad projektas turi geras ilgalaikes perspektyvas, net jei faktoriai yra labai neryškūs.

2. PĖSČIŪJŲ SKAIČIAVIMO METODIKA

Pirmiausia buvo sukurtas ir išmėgintas pirminės idėjos prototipas. Jis pasiteisino tik iš dalies, todėl buvo pasiūlyti pataisymai ir sudaryta galutinė metodika praėivių skaičiavimui.

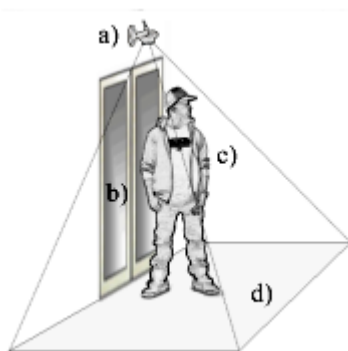
2.1. Prototipas

Norint patikrinti ar praėivių skaičiavimo idėja galėtų pasiteisinti, pradzioje reiktų apgalvoti ir realizuoti prototipą. Tai leistų nustatyti potencialias metodikos problemas, pasikoreguoti ir pasiekti geresnių rezultatų.

2.1.1. Techninės įrangos pasirinkimas

Analizės metu buvo pastebėta, kad racionalus sprendimas būtų imti pigų jutiklį, o visus skaičiavimus atlikti kompiuteriu. Kompiuterių pajėgumai yra gerokai pigesni nei brangūs jutikliai, be to kompiuteriai ir toliau gana sparčiai pinga. Todėl pradinis sprendimo variantas galėtų būti paprasčiausia WEB kamera, o algoritmai vykdomi personaliniame arba nešiojamame kompiuteryje. Jei sprendimas pasiteisintų, būtų galima pereiti prie IR arba IP kamerų. Toks techninis sprendimas kainuotų labai nebrangiai, ypač jei algoritmams nereiktų labai daug skaičiavimo pajėgumų.

Taip pat analizės metu buvo pastebėta, kad jutiklį patogiausia kabinti virš praėjimo, nes taip lengviau pasiekti didesnę tikslumą.



31 pav. Sistemos techninė realizacija

Virš durų (b) pakabinta vaizdo kamera (a) fiksuoja bet kokį judesį žemiau jos. Kameros matavimo ribos nėra didelės (d), vienu metu matoma vos vienas kitas žmogus (c). Sistemos gaunami duomenys pavaizduoti žemiau.



32 pav. Sistemos matomi vaizdai

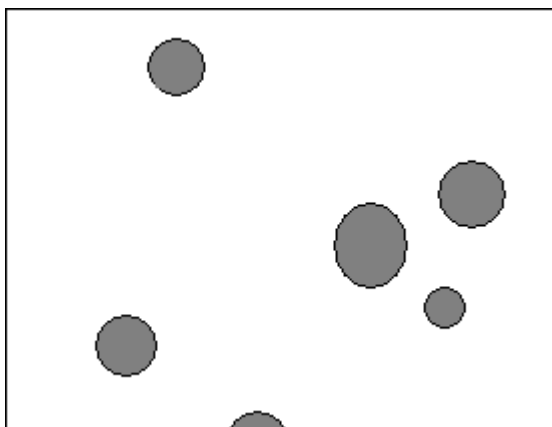
Toks techninis sprendimas leidžia patikrinti algoritmų efektyvumą, nes nereikia jokios specialios įrangos.

Visi bandymai atliekami su Intel Pentium 4 (3GHz procesorius, 2 GB RAM atminties). Vaizdai gaunami su Logitech® QuickCam® Pro 9000.

2.1.2. Pradinė sprendimo idėja

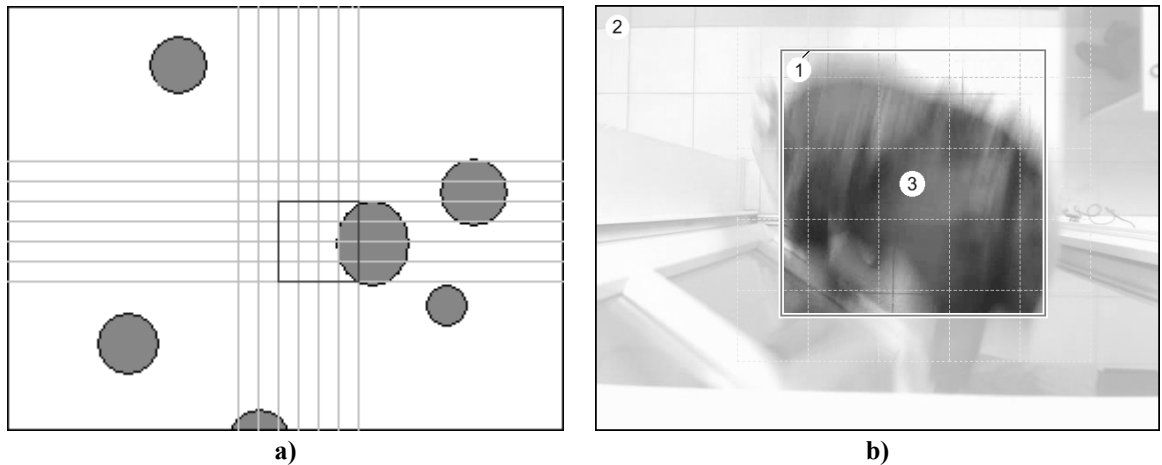
Norėdami suskaičiuoti pėsčiuosius, bandysime lokalizuoti viršugalvį. Pasirinksime kokį nors metodą, kuriuo galima bus sistemą apmokyti skirti viršugalvį nuo fono ir taip klasifikuosime paveikslėlius.

Tarkime, kad vaizde, gautame iš kameros, realus žmonių pasiskirstymas erdvėje galėtų būti kaip žemiau patektame paveikslėlyje.



33 pav. Numatomas žmonių pasiskirstymas erdvėje

Tada sistema paveikslėlių segmentuotų ir kiekviename segmente bandytų paskaičiuoti kokia tikimybė, kad ten yra žmogus.

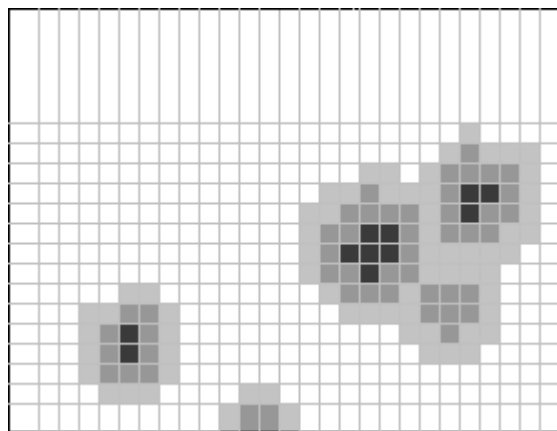


34 pav. a) skenuojami visi segmentai b) atpažinimas vienu metu tik viename segmente

Atpažinimo metu paduodami paėiliui tik norimi segmentai, o ne visas paveiksliukas. T.y. visa kita paveiksliuke išvaloma, paliekama tik segmento sritis. Jei vaizdas iš kameros yra 640x480 pikselių dydžio, tai segmento dydis galėtų būti 80x80 pikselių.

Segmentai turėtų persidengti, nes truputi pastūmus matomumą ribą, toje matomoje srityje atpažinimo rezultatas didėja arba mažėja ir taip gauname tolydesnius duomenis skenuojant visą paveiksliuką.

Susumavus visų segmentų tikimybes, gautume žemėlapi, kuriame surašytos tikimybės kur yra žmonės. Kuo tamsiau, tuo tikimybė didesnė, o kai kur gautume tikimybes, kad būtent ten žmogaus tikrai nėra (balta), kai kur neapibrėžtumą.



35 pav. Numatomi skenavimo rezultatai

Galima sakyti, kad turime funkciją F , kuriai paduodama x ir y koordinatė, o funkcija grąžina skaičių, kuris reiškia tikimybę, kad toje vietoje yra žmogus.

Tada galima surasti šios funkcijos visus maksimumus pagal kokį nors ekstremumų paieškos algoritmą ir taip suskaičiuoti kiek yra žmonių ir kokios jų koordinatės.

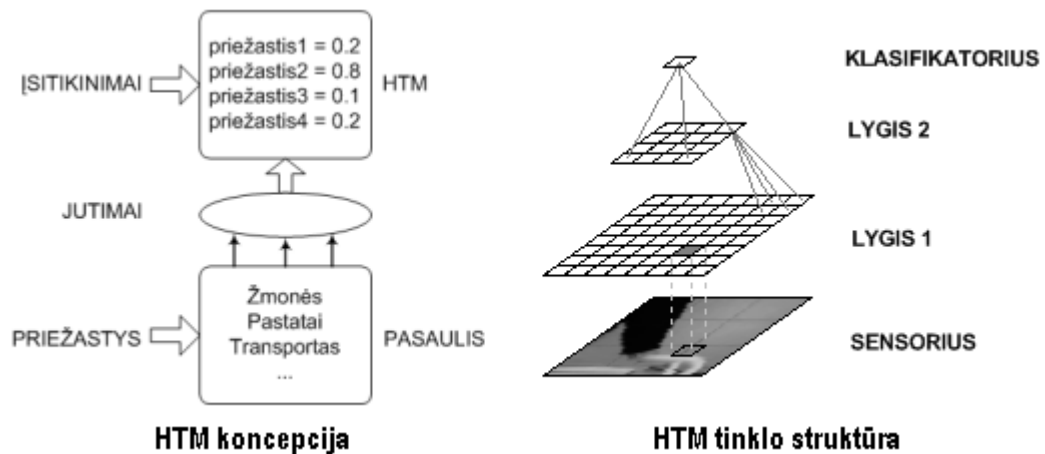
Čia būtų du variantai, arba pirma sudaromas tikimybių žemėlapis ir tada ieškomi maksimumai, arba pats paveiksluko skenavimas atliekamas ieškant maksimumų iš karto taip bandant sutaupyti laiką ir neskenuoti visko.

Pradiniame uždavinio formulavime tarsime, kad paveikslėlyje gali būti tik vienas žmogus ir bandysime jį lokalizuoti bei sekti jo judėjimą.

2.1.3. Metodų pasirinkimas

Atpažinimo uždaviniui spręsti pasirinktas HTM metodas. Jis yra pakankamai naujas ir literatūroje dar nėra jo bandymų jį taikyti šiai problemai spręsti.

Šis metodas realizuoja smegenų žievės (angl. neocortex) algoritmines savybes [24] ir leidžia išspręsti daugelį atpažinimo uždavinių pagal tą pačią schemą, neprogramuojant individualių sprendimų. Tuomet daugiausia darbo yra skiriama HTM tinklo modeliavimui, pradinių duomenų paruošimui ir apmokymo strategijai parinkti.



36 pav. HTM koncepcija ir trijų lygių tinklo struktūra

HTM tinklas apmokomas duomenis paduodant per pasirinktą jutiklį. Tai gali būti vizualūs, garso duomenys ar tiesiog skaičių sekos, atspindinčios realių procesų šablonus (angl. patterns). Duomenys iš jutiklių patenka į pirmą tinklo lygį ir kyla aukštyn iki paskutiniojo. Kiekviename lygyje suformuojamas vis abstraktesnis suvokimas apie tai, kas yra matoma jutiklyje. Tinklas randa erdvines ir laikines savybes bei sąryšius ir tokiu būdu gautos sąsajos leidžia nustatyti kuriai kategorijai priklauso matomas vaizdas.

Apmokius HTM tinklą, jis naudojamas atpažinimui. Tarkime mes į apatinį lygį iš jutiklio paduodame paveikslą ir kaip rezultatą gauname kategorijos numerį su tikimybe, kad matomas vaizdas priklauso būtent tai kategorijai. Šiuo atveju reikės dvejų kategorijų, kurių pirmą reikš, kad tai viršugalvis, o antroji, kad tai bet kas kita.

Pats HTM metodas suskaičiuoti objektų negali, jis gali tik atpažinti, ar pateiktame vaizdo fragmente yra dominantis objektas ar nėra. Todėl uždavinys susideda iš dviejų etapų,

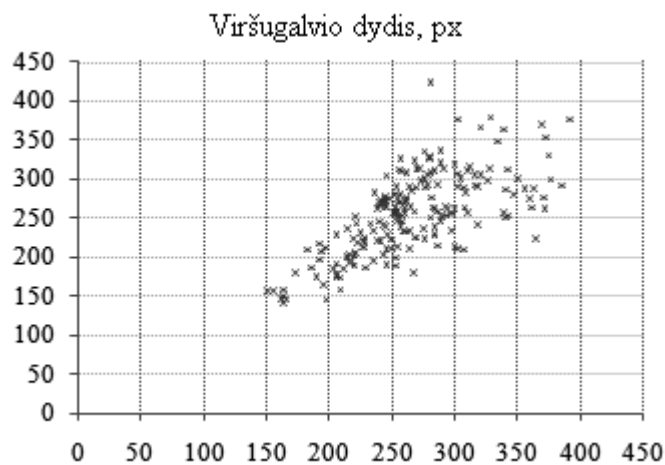
pradžioje surandami visi dominantys objektai, o tik tada žinodami kur jie yra, juos suskaičiuojame. Tai daroma skenuojant visą gautą vaizdą iš kameros ir ieškant, kuriuose fragmentuose žmogaus buvimo tikimybė yra aukšta.

Apmokant sistemą sukuriamos dvi kategorijos duomenų: žmonės ir bet kas kita. Kiekvieną kategoriją sudaro po keli šimtai paveikslėlių.



37 pav. Dvi duomenų klasės a) vien tik žmonės b) visa kita

Bendras viršugalvių pasiskirstymas pagal dydį pikseliais:



38 pav. Viršugalvio dydžio pasiskirstymas

Metodas kol kas dirba tik su nespalvotais paveikslėliais.

2.1.4. Programinė įranga

Numatoma naudoti programinę įrangą daugiausiai susieta su pačiu metodu:

- Vision toolkit 1.7.1 – programa su grafiniu vartotojo interfeisu leidžianti apmokyti, testuoti ir peržiūrėti standartinį septinių lygių HTM tinklą. Šis tinklas jau yra optimizuotas, suderinti parametrai, parinkti filtrai, sensoriaus dydis ir t.t. Programos privalumas, kad pradiniai keli lygmenys jau yra apmokyti gausiu kiekiu duomenų, todėl gerokai greičiau trunka apmokymas, nes apmokomi tik viršutiniai tinklo lygmenys;
- NuPIC 1.7.1 – karkasas HTM metodo pagrindu. Jame yra paruoštos priemonės tinklui sukurti, redaguoti, apmokyti ir testuoti. Dalis HTM algoritmų jau

realizuota ir veikia labai gerai, tačiau keletas idėjų dar neįgyvendinta arba nepublikuota;

- Vision Framework – karkasas parengtas NuPIC pagrindu ir orientuotas į vaizdo atpažinimo problemas. Leidžia patogiau ir lengviau sukurti tinklus, kurių sensoriai priima paveiksliukus, lengviau apmokyti ir testuoti, nes sukurtos priemonės leidžia automatizuoti kai kuriuos procesus;
- Python – NuPIC karkasas yra realizuotas Python kalbos pagrindu, todėl bet koks algoritmas naudojantis NuPIC karkaso galimybes turi būti parašytas Python kalba;
- MatLab – kai kurie uždaviniai gerokai paprasčiau ir greičiau sprendžiami ne Python kalba, o tiesiog MatLab priemonėmis. Ypač tai aktualu, kai kalba eina apie filtravimą, kitokį paveiksliukų apdorojimą bei rezultatų atvaizdavimą grafikais;
- Microsoft Visual Studio, .NET 4 framework, C# - bet kokį grafinį interfeisą atvaizduoti gerokai paprasčiau pasinaudojus šiomis priemonėmis nei Python kalba ar MatLab programine įranga.

Visi eksperimentai daromi Windows XP operacinėje sistemoje, darbui su dokumentais naudojamas Microsoft Office 2003.

2.1.5. Sistemos tikslas

Iš esmės, ši sistema yra dėmesio mechanizmas. Ji vienu metu „žiuri“ į tam tikrą kameros vaizdo sritį ir išduoda rezultatą, kokia tikimybė, kad tame fragmente yra žmogus. Sistemos duomenys yra atsitiktiniai veiksniai, išreikšti kameros vaizdo fragmentais \vec{x} . Tuose fragmentuose arba yra žmogus arba nėra.

Sistemos būseną yra atpažinimo rezultatas, tikimybė, kad duotame fragmente yra žmogus t.y. $\vec{k} \in [0;1]$. Jeigu tikimybė artėja prie nulio, vadinasi greičiausiai ten žmogaus nėra, jeigu tikimybė artėja prie 1, labai tikėtina, kad ten yra žmogus.

Mes tariame, kad sistemą galima optimizuoti taip, kad jos būseną \vec{k} priartėtų prie realybės \vec{k}^* t.y. mes norime, kad sistemos rezultatas kuo mažiau skirtųsi nuo realybės. Taigi, optimali sistemos būseną yra \vec{k}^* .

Sistema yra statinė, be grįžtamojo ryšio. Optimizavimo uždavinį formuluojame sistemos algoritmui patobulinti, taip, kad atpažinimo klaidų būtų kuo mažiau. Yra bent keletas

parametrų $Q = (Q_1, \dots, Q_n)$, kuriuos mes galime keisti ir stebėti kaip gerėja ar prastėja rezultatai. Tačiau apsiribosime tik dvejais:

- Q_1 yra dėmesio fragmento dydis, kai $Q_1 \in [0;600]$ pikselių. Šis dydis tiesiogiai priklauso nuo vaizdo kameros savybių. Jei tai būtų didesnio kampo kamera, apimanti daugiau erdvės, matuotume kitaip;
- Q_2 yra triukšmo filtras, kai $Q_2 \in [0;100]$ filtro stiprumas. Mes tariame, kad gautas vaizdas turi įvairių triukšmų, kuriuos nufiltravus, galima tikėtis gauti geresnį atpažinimo procentą.

Taigi, visas uždavinys susideda iš šių dviejų keičiamų parametrų optimizavimą, taip, kad sistemos būseną labiausiai atitiktų optimalią, todėl gerumo funkciją galima užrašyti taip:

$$G(Q) = \sum (\bar{k} - \bar{k}^*)^2 \quad (5)$$

Tuomet sistemos tikslo funkcija:

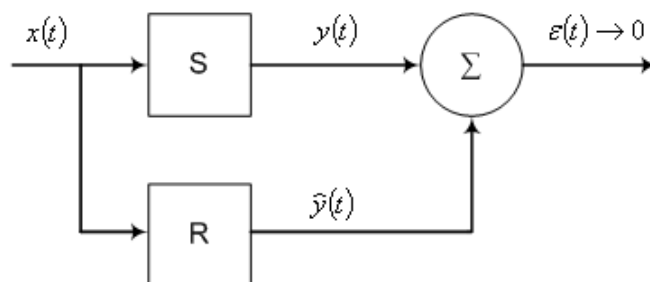
$$\min_Q \sum (\bar{k} - \bar{k}^*)^2 = \min_Q G(Q) \quad (6)$$

Sistemos esama būseną \bar{k} priklauso tik nuo keičiamų parametrų ir gautų fragmentų \bar{x} . Tačiau visiškai nepriklauso nuo to, kokia sistemos būseną buvo prieš tai.

Abu parametrus laikysime tarpusavyje nepriklausomais t.y. optimizuojant vieną parametru, nereikia sekėti kito parametro įtakos ir galime abu optimizuoti atskirai.

Išsprendus uždavinį tikimės, kad bus galima pasakyti tiksliai, kokio dydžio dėmesio sritis turi būti ir kiek labai reiktų nufiltruoti triukšmą, kad sistemos sprendinys kuo labiau atitiktų realybę.

Šis uždavinys taip pat galėtų būti formuluojamas kaip identifikavimo [26] uždavinys, nes abu parametrai nežinomi ir juos reikia nustatyti.



39 pav. Identifikavimo uždavinys

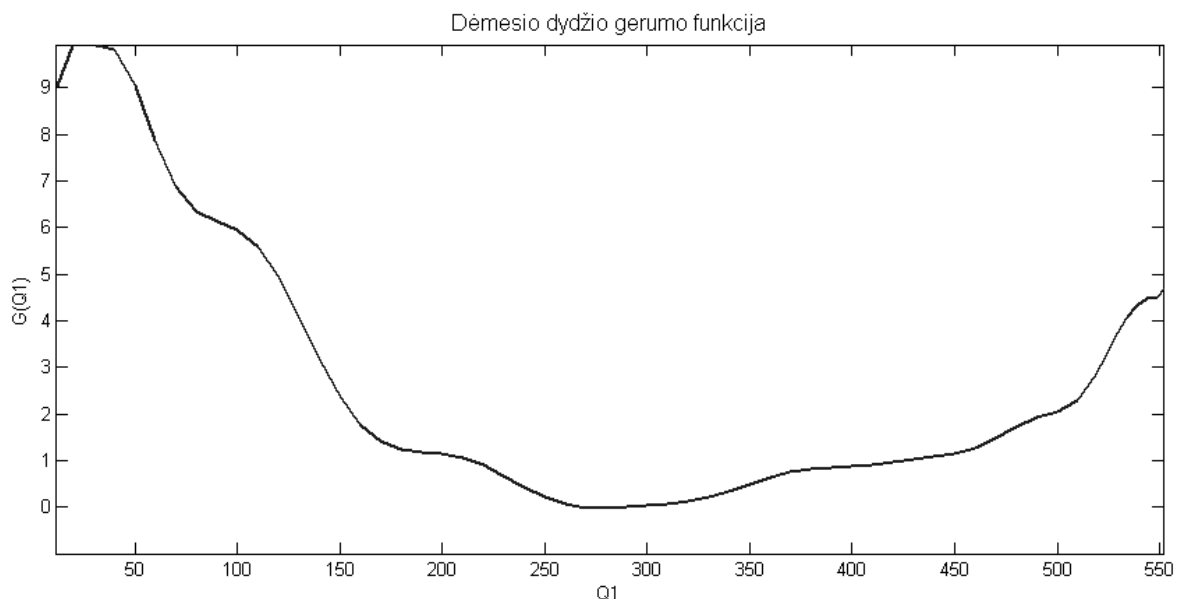
Čia pavaizduota reali sistema S , t.y. ji atpažįsta tiksliai. Mes kuriame sistemą R , kuri yra panaši į realią. Duomenys, vaizdo fragmentai, yra $x(t)$, realios sistemos atsakas $y(t)$, o

sukurtos $\hat{y}(t)$. Mes siekiame, kad šių sistemų rezultatų skirtumas būtų kuo mažesnis $\varepsilon(t) \rightarrow 0$.

2.1.6. Dėmesio matmenų optimizavimas

Abu dėmesio matmenys vienodi, t.y. aukštis ir plotis tokie pat, o forma kvadratinė. Su tokiais ribojimais, reikia rasti gerumo funkcijos (1) taškus priklausomus nuo Q_1 , interpoliuoti ir gauti minimumą.

Paruošiamė 20 paveiksliukų, iš kurių vienoje dalyje yra žmogus, o kitoje nėra. Mes tiksliai žinome, kuriuose žmogus yra ir kur būtent, t.y. S sistemos būseną mes žinome visada. Belieka keisti dėmesio dydį ir žiūrėti koks yra būsenų skirtumo kvadratas. Žingsnis tegu būna 10, pradinė vertė 0, o didžiausia 600. Paskaičiuojame ir nubrėžiame gerumo funkciją (1), Q_1 charakteristikai.



40 pav. Dėmesio dydžio gerumo funkcija

Kaip matome iš gautos gerumo funkcijos, maždaug iki 250 pikselių didinant dėmesio matmenis, sistemų rezultatų nesutapimas mažėja, o nuo 300 pikselių, rezultatai prastėja. Kitaip tariant, optimali reikšmė yra maždaug tarp 250 ir 300. Kadangi funkcija toje vietoje beveik tiesė, nebūtina ieškoti funkcijos analitinės išraiškos, minimumą randame imdami mažiausią žinomą reikšmę – 270 pikselių.

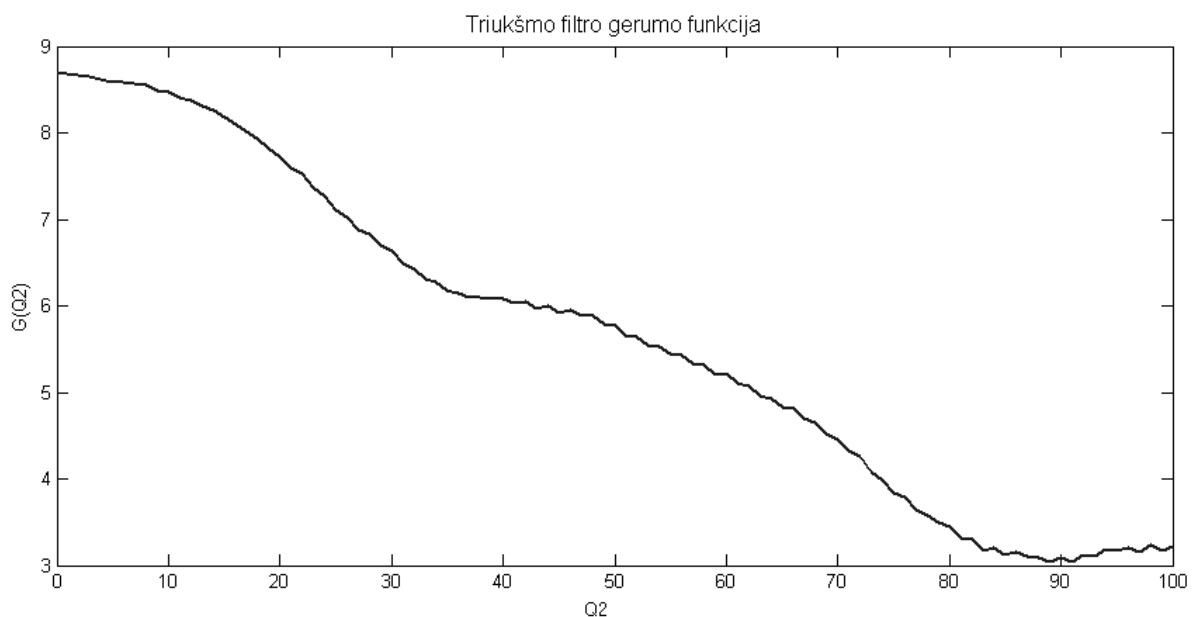
2.1.7. Filtro optimizavimas

Triukšmų pobūdis būna labai įvairus. Šiais eksperimentais tiesiog pamėginsime išsiaiškinti, ar filtravimas daro sistemai kokią nors įtaką ir kokią būtent. Bandysime filtruoti

taip vadinamą „Salt and pepper“ triukšmą naudodami „2-D median“ filtrą. Šis filtras dažnai naudojamas pašalinti triukšmą išsaugant briaunas (angl. edges).

Matlab turi ir daugiau priemonių pašalinti triukšmą [27], kaip pavyzdžiui „Kalman“, „Wiener“, vidurkinimo (angl. averaging), „Gaussian“ filtrai. Analogiškai triukšmai skirstomi į įvairias kategorijas, pvz.: „Gaussian“, „Salt and pepper“, „Poisson“, „Speckle“. Yra būdai susikonstruoti savo filtro modelį, tai gana sudėtinga procedūra. Mes nesigiliname kokie būtent triukšmai galėtų būti mūsų atveju, tiesiog išbandome nufiltruoti „Salt and pepper“ tipo triukšmus naudodami „2-D median“ filtrą.

Gerumo funkcija, Q_2 charakteristikai, gaunama atliekant analogiškus eksperimentus. Turime 20 paveiksliukų, iš jų iškerpame dominančius fragmentus, kuriuose yra žmogus arba nėra. Fragmento dydį naudojame 270 pikselių, nes išsiaiškinome, kad toks dydis yra optimalus. Kiekvieną fragmentą filtruojame didinant filtro stiprumą, kurį nusako dydis $Q_2 \in [0;100]$, rezultatus įsistatome į gerumo funkciją (1) ir nubrėžiame.



41 pav. Triukšmo filtro gerumo funkcija

Tikėtasi, kad filtro įtaka pradžioje bus pozityvi, pasiekus optimalią reikšmę vėl suprastės. Tačiau gauti rezultatai gana stebinantys, nes filtro stiprumą nustačius apie 90, vaizdas gaunamas labai išsiliejęs, net ir žmogui atpažinti darosi labai sunku. Norint įsitikinti, kad rezultatai tikrai tikslūs, reiktų naudoti gerokai daugiau duomenų, nes dvidešimt paveiksliukų yra labai mažai. Tikriausiai minimumas persistumtų labiau į kairę.

Funkcija ties minimumu beveik tiesė, todėl galime paprasčiausiai imti mažiausiai rastą reikšmę ir neieškoti analitinės funkcijos išraiškos. Gavome, kad optimalus filtro stiprumas yra 91. Tai reiškia, kad detalės HTM tinklo rezultatą stipriai įtakoja.

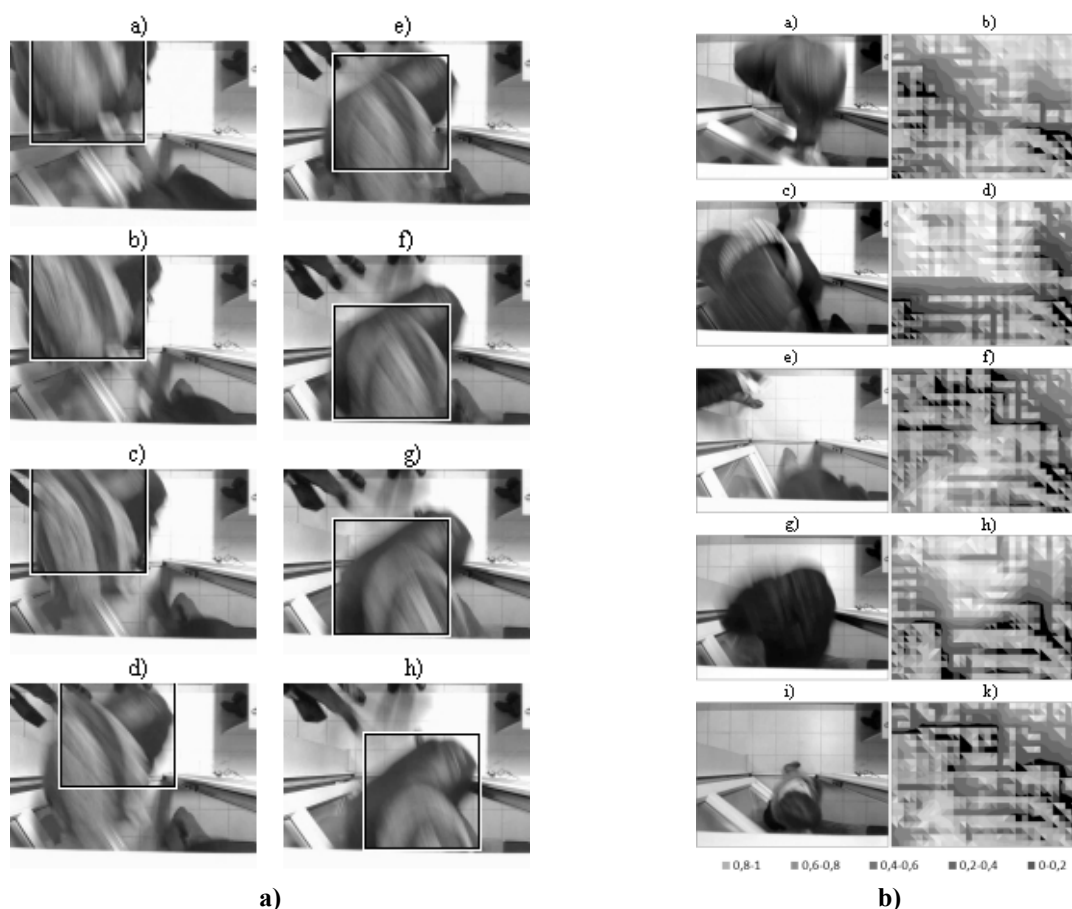
2.1.8. Prototipo testavimo rezultatai

Plačiau gauti rezultatai pateikti publikuotame straipsnyje (žr. 2 priedą), o čia pagrindinių rezultatų santrauka.

Apmokymo [28] ir testavimo metu buvo naudojama maždaug 50 žmonių įeinančių ir išėinančių iš pastato kadrai. Maždaug 85 procentai kadrų atrinkti apmokymui, visi kiti buvo naudojami testavimui.

Iškirpus žmonių fragmentus ir ne žmonių fragmentus, buvo gautos dvi bibliotekos: 348 ir 410 paveikslukų. Apmokytas tinklas labai tiksliai atpažindavo net ir tuos atvejus, kurių apmokymo metu nematė. Tokio atpažinimo tikslumas maždaug nuo 90 iki 100 procentų.

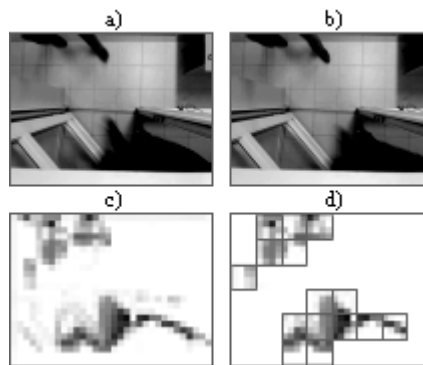
Deja naudojant standartinį septinių lygių tinklą, vieno atpažinimo laikas yra maždaug 0.2-0.3 sekundės, o norint skenuoti visą kadrą pirma suskaidžius į 48 segmentus, reikia maždaug 15 sekundžių. Deja skenavimas yra brangi praktika ir reikėtų gero algoritmo, kuris patartų, kurioje vietoje bandyti atpažinti, o kurias praleisti.



42 pav. a) žmogaus sekimo pavyzdys b) skenavimo rezultatų pavyzdys

Kaip matome iš skenavimo rezultatų, pikus rasti lengva, jei yra tik vienas žmogus, tačiau, jei būtų keli, jie tiesiog susilietu. Priimant prielaidą, kad gali būti tik vienas žmogus, vieno kadro skenavimas gali sumažėti iki 3 sekundžių. Be to, aptikus žmogų, kituose kadruose jau nereikia skenuoti viso vaizdo, pakanka pradėti nuo ten, kur jis jau buvo rastas. O

tam pakanka sekundės dalių. Pradėti skenuoti galima dar ir nuo to vietos, kur buvo užfiksuotas didžiausias judesys.



43 pav. Judesio aptikimas

Pateiktame paveikslėlyje (a) yra prieš tai buvęs vaizdas, o (b) dabartinis. Jei juos labai sumažinti, suskaidyti į mažus segmentus, maždaug po 4x4 pikselių, tai tie segmentai, kuriuose skirtumo tarp abiejų vaizdų yra daugiausiai – ten ir judesio yra daugiausiai. Matome kaip aptinkama, kad daugiausiai judesio yra užsidarančios durys ir žmogaus kojų judesys. Todėl galėtume pradėti skenuoti pasirinktinai iš tų dvejų vietų.

2.1.9. Prototipo testavimo išvados

Suformulavus siekiamos sistemos tikslo funkciją, atlikus eksperimentus, optimizavus pagal du pasirinktus keičiamus parametrus, matome, kad:

- optimalūs dėmesio srities matmenys yra 270 pikselių ilgio ir 270 pikselių pločio fragmentai;
- triukšmų filtravimas gali ženkliai sumažinti paklaidos dydį, tačiau norint gauti patikimesnį rezultatą, reiktų optimizavimo uždavinį spręsti su gerokai daugiau duomenų;
- HTM tinklas labai jautrus detalėms, jas nufiltravus, rezultatai gerėja;
- atpažinimo rezultatas labai priklauso nuo to, kaip duomenys apdorojami prieš paduodant juos atpažinimui, galima būtų ištirti kokią įtaką daro kitokio pobūdžio filtrai, kaip šviesumo keitimas, vaizdo pasukimas ir t.t.;
- „Vision“ tinklas (7 lygių, 204x204 pikselių jutiklis) pernelyg lėtas realaus laiko apdorojimui, todėl reiktų bandyti pasidaryti gerokai mažesnę tinklą, galbūt su mažesniais jutiklio išmatavimais;
- algoritmo sparta galėtų būti geresnė, jei pavyktų tiksliau parinkti pradinius taškus, kuriuose skenuoti ir kuriuos praleisti, nes tikimybė rasti objektą yra labai maža;

- skenavimas neišsprendžia objektų persidengimo problemos, reikia ieškoti būdų objektus atskirti vieną nuo kito;
- HTM metodas yra labai gerai atpažįsta, tačiau ne visais atvejais to pakanka.

Sudarytas matematinis modelis leido pagerinti sistemos funkcionavimo tikslumą ir geriau pažinti pačios sistemos savybes.

Matome, kad toks uždavinio sprendimas pasiteisins tik tuo atveju, jei pavyks supaprastinti tinklą neprarandant atpažinimo kokybės, ir bus rastas būdas atskirti objektus. Kitu atveju, reiktų ieškoti geresnių sprendimų.

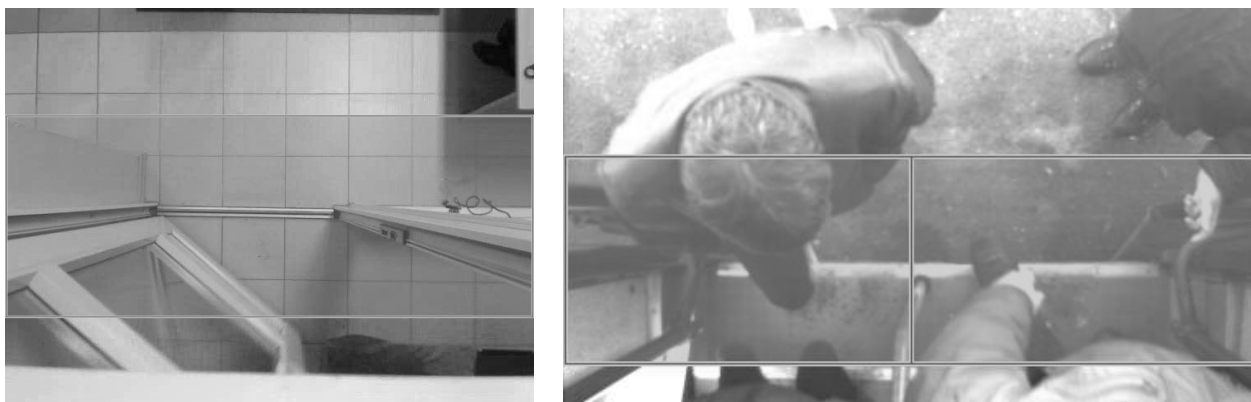
2.2. Pasiūlyta metodika

Prototipas leido patikrinti pirminę pėsčiųjų skaičiavimo idėją. Ji pasiteisino tik iš dalies, todėl šiame skyriuje pirminė idėja yra tobulinama.

2.2.1. Patobulinimų esmė

Bus naudojami vos vienas ar du segmentai, kiekviename iš jų randama žmogaus buvimo tikimybė ir judėjimo kryptis dviem atskirais HTM tinklais. Įvertinus koreliaciją tarp segmentų, gautas tikimybes segmentuose ir judėjimo kryptis, paskaičiuojame žmonių balansą.

Vaizdas suskaidomas į tiek segmentų, kiek natūraliai, vienu metu, gali prasilenkti žmonių. Viena kamera pajėgi matyti vos du prasilenkiančius žmones.



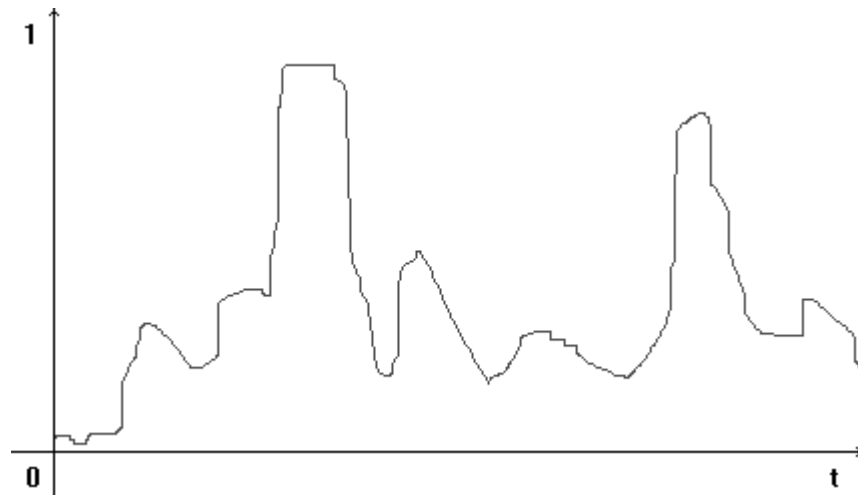
44 pav. Segmentacijos pavyzdys a) vienas segmentas b) du segmentai

Gaunama juosta padalinanti vaizdą į sritis, dažniausiai 1-2, tačiau gali būti 3 ir daugiau, jei praėjimas labai platūs – prekybos centras.

Darome dvi esmines prielaidas, kurios bus pamatiniai sistemos apribojimai:

- viename segmente, vienu metu, gali būti tik vienas žmogus;
- žmogus, vienu metu, gali būti tik viename arba dvejuose segmentuose.

Kiekviename segmente bandoma atpažinti ar tai žmogus ir gaunama žmogaus buvimo tikimybė. Skaičiuojant tikimybes laike, bus gautas kiekvieno segmento grafikas, kurie galėtų atrodyti kažkaip taip (iš anksto nežinom):



45 pav. Numatomi rezultatai, tikimybė, kad segmente yra žmogus

Tuomet:

- jeigu dviejų segmentų duomenys koreliuoja (kaip būtent, iš anksto nežinom, reikia iširti), tuomet reiškia, kad žmogus vienu metu yra dvejuose segmentuose;
- būtų surasti slenksčiai: kiek ilgai turi tęstis aukštos tikimybės ir kiek jos turi būti aukštos, kad galėtume teigti, jog segmente yra žmogus;
- rasti požymius, kada galima teigti, kad vienas žmogus jau praėjo ir pradėjo eiti kitas.

Atpažinimas be krypties nustatymo veiktų ne geriau, nei spindulinis skaičiuotuvas, kuris negali skaičiuoti balanso, o tik paskaičiuoja kiek praėjo žmonių iš viso.

Objekto judėjimo segmente kryptis būtų randama su antru tinklu, kuris būtų apmokytas atpažinti ar tam tikroje vaizdo dalyje yra judėjimas ar nėra. Tuomet uždėtume du arba tris slenksčius, kuriuos žmogus eidamas kirstų ir mes pagal peržengtų slenksčių eiliškumą nustatytume judėjimo kryptį.

Eigoje reikėtų atlikti atitinkamus tyrimus:

- vieno segmento tyrimas: yra, nėra, jau nebėra žmogaus;
- koreliacijos tarp dviejų segmentų išmatavimas bandant nustatyti ar tai vienas žmogus per du segmentus ar du žmonės;
- slenksčių paieška, kad būtų galima kuo tiksliau pasakyti kada vienas žmogus pateko į segmentą, kada jis išėjo, kada įėjo kitas;
- judėjimo krypties nustatymo tinklo paruošimas, apmokymas, išbandymas;

- spartaus ir patikimo tinklo paruošimas žmogui atpažinti;
- filtravimo įtakos bandymai.

Skenavimo naudoto prototipe visiškai atsisakyti, nes:

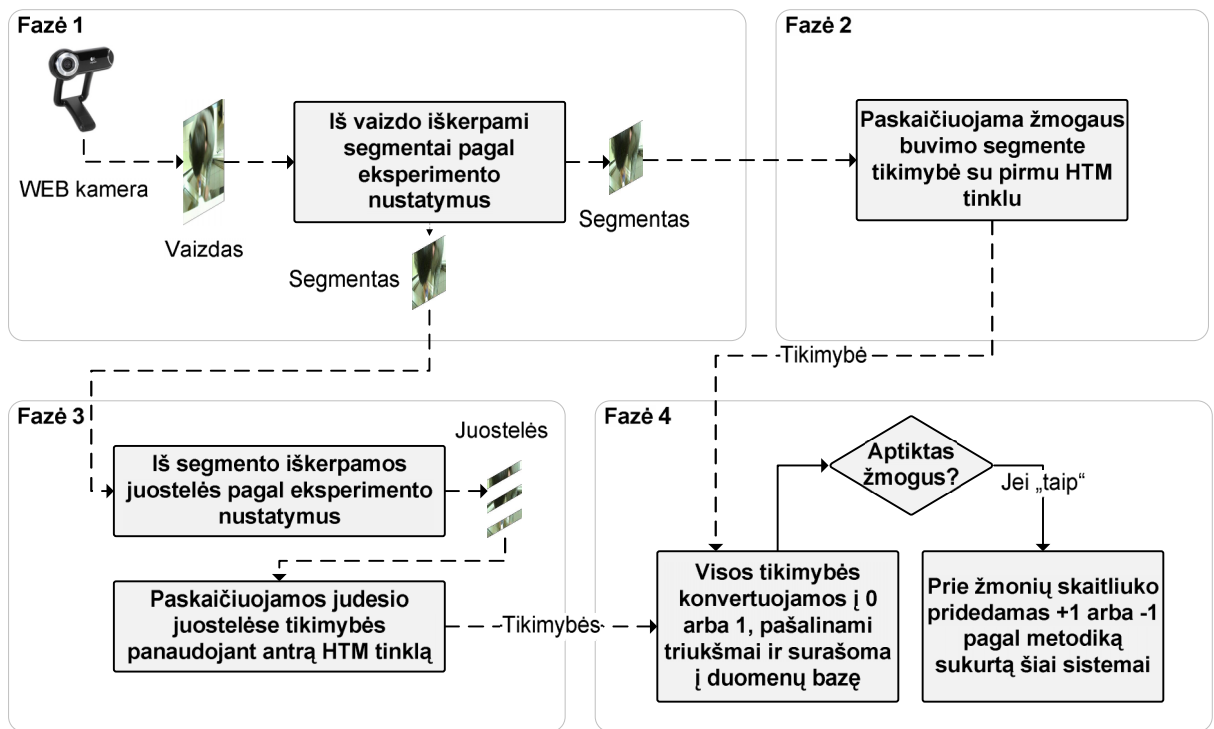
- skenavimo atveju net ir radus visus objektus, jų skaičiavimas netampa trivialus;
- neišsprendžiama persidengimo problema;
- apkrauna skaičiavimais derinimo procesą, tenka daug laiko gaišti laukiant valandų valandas kol bus paruošti truputi kitokie tinklai, kad juos išbandyti;
- skenavimas labai blogas realaus laiko skaičiavimams, tuomet algoritmų sparta labai priklauso nuo rezoliucijos ar kiek daug yra skenuojama;
- HTM naudojamas šiek tiek ne pagal paskirtį, nes lokalizaciją numato „Top-Down attention“ algoritmai, kurie dar nepublikuoti.

Naujai idėjai patikrinti reikia dviejų tinklų, žmogaus atpažinimui ir objekto judėjimo krypties nustatymui.

Turėtų gautis sistema, kuri geba paskaičiuoti realiuoju laiku (kol kas tik su PC) žmonių balansą, kai žmonės nesigrūda, o praėjimo pralaidumas maždaug 1-3 žmonės vienu metu. Prototipas skirtingiems kontekstams turi būti apmokytas atskirai, pavyzdžiui autobusams ir prekybos centrams, neužtenka padalinti į segmentus.

2.2.2. Realus laiko skaičiavimo algoritmas

Algoritmas turi dirbti realiuoju laiku. Diagramoje supaprastintai pavaizduotas sistemos algoritmas.



46 pav. Pėsčiųjų srauto monitoringo algoritmo schema

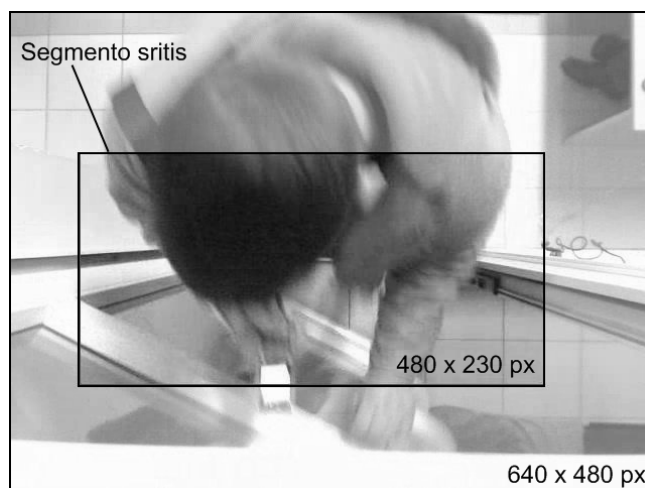
Sistema suskaidyta į keturias fazes. Kiekvienoje atliekamas svarbūs paskaičiavimai, kurie reikalingi kitai fazei. Kartojant ciklą iš naujo, kaupiami rezultatai.

2.2.3. Fazė 1. Duomenų nuskaitymas ir paruošimas

Duomenys nuo prototipo nesikeičia, kamera lieka kaboti virš praėjimo.

Praėjimai daugiausia vienas nuo kito skiriasi savo pločiu. Sunku pasakyti koks yra praėjimų pasiskirstymas pagal plotį, tačiau labai dažnai sutinkamos durys, pro kurias gali praeiti vos 1-2 žmonės nesigrūdami. Rečiau pasitaiko 3 žmonėms ar platesnės. Bazinis sprendimas yra skirtas praėjimams, kurių plotis toks, kad vienu metu laisvai praeiti gali tik vienas žmogus. Tačiau turėtų būti galima sistemą išplėsti ir pritaikyti platesniems praėjimams, pro kurias laisvai gali prasilenkti 2-3 žmonės.

Prieš sistemos paleidimą, reikia pasirinkti segmentą, kuriame ieškosime žmogaus. Geriausia tai padaryti tarpduryje, nes tikimybė, kad ten kas nors užsibus sustojęs yra pati mažiausia. Pasirenkame pakankamai platų segmentą (480x230 px) tam, kad joks praeivis neprasmuktų.



47 pav. Segmentacijos pavyzdys

Fonas nėra pašalinamas. Tiesiog sistema bus apmokoma, kad fonas yra ne žmogus. Žinoma, kad fono pašalinimas galėtų padaryti sistemą universalesne, tačiau ši užduotis nėra triviali ir reikalauja labai daug derinimo, kad tikrai veiktų be informacijos praradimo.

HTM tinklo apmokymui ir testavimui buvo naudojamos 53 situacijos, kai žmonės įeina arba palieka patalpą. Kai kurios situacijos yra labai komplikotos, nes pėstieji juda labai greitai arba yra labai mažas laiko tarpas tarp dvejų praeivių.

Apmokymui visos situacijos suklasifikuotos į dvi kategorijas. Pirmoje kategorijoje yra tik tie paveikslukai, kuriuose matoma didelė žmogaus dalis, o antroje kategorijoje bet kas kita (fonas, šešėliai, galūnės ir pan.). 70 procentų duomenų buvo panaudota apmokymui, likę 30 procentui testavimui.



48 pav. Keletas pavyzdžių iš apmokymo duomenų: a) tik praeiviai b) visa kita

Bandymų metu buvo atkreiptas dėmesys į keletą labai svarbių aspektų. Jei praėjimas turi duris, tuomet praeiviai pradžioje sulėtina greitį, nes reikia atidaryti duris. Jei jų nebūtų, greitis būtų gerokai didesnis. Kai kada praeiviai yra labai greiti ir pasirodo po kamera vos kelėtai kadru, o kai kurie stovi po kamera po keletą ar keliolika sekundžių visai nejudėdami. Be to, labai daug įtakos turi žmogaus ūgis. Vaikų viršugalvis vizualiai atrodo labai nedidelis palyginant su aukštaūgiu žmogumi, todėl labai svarbu tinkamai parinkti kameros pakabinimo aukštį, kad viršugalviai nebūtų pernelyg dideli ar per maži. Taip pat svarbu atkreipti dėmesį, kad praeiviai ne visada juda tiesa trajektorija.

Kai kuriuose laiko perioduose kadrai yra tušti, pavyzdžiui jei tai būtų autobusas, tai kol durys uždarytos – nėra jokio reikalo atlikti skaičiavimus su tais vaizdais, nes niekas tikrai pro duris nevaikšto. Tokios ir panašios situacijos gali padėti sutaupyti skaičiavimų resursus.

Apmokymo paveikslukai buvo kategorizuojami taip, kad pirmoje (praeivių) kategorijoje atsidurtų tie segmentai, kuriuose didelė žmogaus dalis matoma. Buvo peržiūrėti visi 12600 kadrai ir suklasifikuoti į dvi kategorijas.

2.2.4. Fazė 2. Praeivių atpažinimas

Segmente matomas vaizdo fragmentas iškerpamas ir paduodamas HTM tinklo sensoriui. Tinklas paskaičiuoja žmogaus buvimo tikimybę segmente ir ją gražina.

Jeigu prieš tai buvęs segmentas nuo dabartinio skiriasi visai nedaug, būtų galima jam tikimybės neskaičiuoti, bes ji bus labai panaši į prieš tai buvusio segmento. Yra įvairūs metodai, kuriais galima išmatuoti dviejų vaizdų tarpusavio skirtumą.

Didžiausias darbas yra suprojektuoti HTM tinklą, kuris būtų pakankamai tikslus ir pakankamai spartus. Problema ta, kad didinant tikslumą, mažėja sparta, ir atvirkščiai. Buvo išbandyta naudoti „Vision toolkit“, kuris išanalizuoja duomenis ir parenka vieną iš dvejų labai gerai suderintų tinklų. Šie tinklai yra labai tikslūs, tačiau mūsų sprendžiamai problemai jie pernelyg dideli ir lėti. Todėl HTM tinklą reikia susiprojektuoti patiems.

Sukonstruotas tinklas yra trijų lygių (žr. 5 priedą). Daugelis parametrų buvo derinami empiriškai, paprasčiausiai išbandant ar rezultatai gerėja. Pačių parametrų yra dešimtys, o viso tinklo apmokymas per naują ir testavimas trunka apie valandą ar daugiau, todėl tokie derinimai užima be galo daug laiko.

Pradinis tinklo sensoriaus dydis yra 120x60 pikselių. Toks dydis yra pakankamai mažas, kad tinklas vis dar būtų greitas ir pakankamai dydis, kad dar galėtų išmokti atskirti žmones nuo viso kito.

„Multi-resolution“ naudojamas, kai tinklą apmokyti dar ir su pamažintais-padidintais apmokymo paveiksliukais. Kadangi mes tinklą apmokome su nuosekliais vaizdais, tai ši savybė buvo neaktuali ir ji buvo išjungta.

Ši sistema turėtų gerai veikti ir vidaus ir lauko sąlygomis, todėl duomenys bus labai įvairūs. Juose gali būti staigūs šviesos pokyčiai, šešėliai, lietus, sniegas ar net rūkas. Kad šie poveikiai būtų kiek sumažinti, yra naudojamas kontrasto normalizavimo filtras, kuris yra NuPIC programinėje įrangoje.

Eksperimentų metu nebuvo palyginti dienos ir nakties vaizdai. Galbūt to ir nereikės daryti su WEB kamera, nes naudosime infraraudonųjų spindulių kamerą. Buvo išbandyta nufilmuoti keletą situacijų ir ryškumas gavosi gerokai stabilesnis nei WEB kameros.

HTM tinklui dar yra nurodytas paveiksliukų dydžio keitimo filtras. T.y. jei į sensorių pateks paveiksliukas didesnis už numatytą arba mažesnis, jis paprasčiausiai bus sumažintas iki 120x60 pikselių išmatavimų.

Svarbu pažymėti, kad HTM kol kas nedirba su spalvomis, todėl visi paveiksliukai yra konvertuojami į pilkos spalvos atspalvius.

Pirmieji eksperimentai buvo atlikti naudojant vadinamą erdvinį mazgų tipą (angl. spatial pooler), tačiau jis nepasiteisino. Reikėjo, kad pirmas lygmuo prie tapačių duomenų, sukurtų tokias pačias erdvines grupes, tačiau sukurdavo skirtingas. Todėl rezultatai tapdavo sunkiai prognozuojami, nes kas kart vis skirtingas erdvinis alfabetas būdavo sukurtas. Vėliau buvo išbandytas „Gabor“ filtras vietoje mazgų ir taip buvo laimėta, kad grupės pirmame lygyje jau susiformuodavo visą laik vienodos prie tų pačių duomenų, o paties filtro nereikia apmokyti, todėl negaištamas brangus laikas.

Antras tinklo lygis sukuria laikines grupes. Tokių mazgų tipas yra „TemporalPoolerNode“. Be to, mazgai yra klonuojami, tam, kad dalintųsi tarpusavyje ta pačia duomenų matrica. Šio lygio algoritmą nustatėme į „maxProp“, nes jį galima naudoti su laiku pagrįstu atpažinimu TBI, nors „hardcoded“ tipas yra gerokai labiau priderintas.

„Multi-resolution“ antrame lygyje taip pat išjungta su „scaleRF“ ir „scaleOverlap“ parametrais. Naudojama „transitionMemory“, kuris padeda susieti prieš tai buvusius duomenis su naujais laike. Grupės suvienodinamos pagal dydį.

Trečias tinklo lygis yra klasifikatorius. Buvo išbandytas SVM ir PM klasifikatoriai, tačiau SVM mūsų atveju veikė tiksliau. „outputElementCount“ nurodyta „2“, nes yra tik dvi grupės: žmogus arba visa kita.

„minC“ ir „minGamma“ yra minimalūs „C“ ir „Gamma“ parametrai. Šie SVM parametrai įtakoja klaidų dažnį. „SVMClassifierNode“ atlieka optimizavimo procesą, kurio metu bando surasti tokį „C“, su kuriuo klaidų dažnis duomenims yra mažiausias.

Po to, kai tinklas yra paruoštas, jį reikia apmokyti. Apmokymui naudojama „Flash“ tipo apmokymą, kurio metu visi duomenų paveikslukai paduodami nuosekliai. Mūsų atveju šis variantas tinka labiausiai, nes mus domina vaizdo sekos.

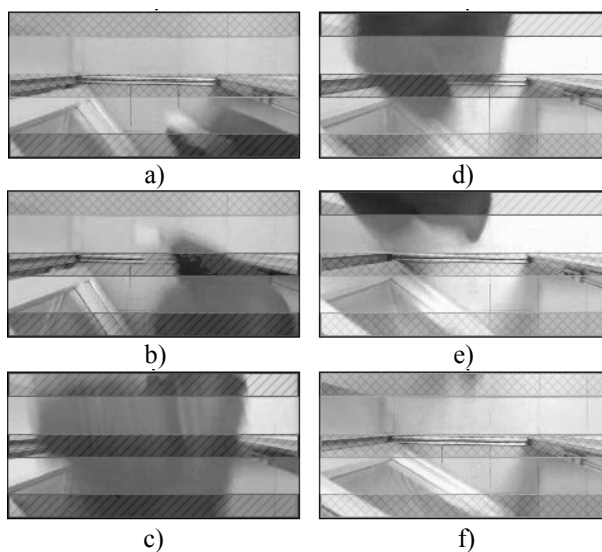
Taip pat išbandėme naudoti HOT algoritmus aukštesniems tinklo lygiams ir gavome gana įdomių rezultatų, tačiau šie algoritmai yra ypatingai lėti ir su jais verta dirbti tik tada, jei eksperimentai daromi galingame klasteryje.

2.2.5. Fazė 3. Krypties nustatymas

Atpažinti ar yra žmogus segmente yra nepakankama, reikia dar nustatyti ir jo judėjimo kryptį. Buvo išanalizuotos 183 realios situacijos, kai žmonės įeina ir išeina iš pastato. Kai eina tik vienas žmogus, gana nesunku pasakyti kokia yra judėjimo kryptis bet kuriuo metodu, tačiau darosi labai sunku, jei keli žmonės eina labai arti vienas kito. Prieš tai buvo nuspręsta, kad viename segmente, vienu metu, gali būti tik vienas žmogus, tačiau realybėje net jei ir mažai vietos, vis tiek būna situacijų, kai žmonės bando prasilenkti, t.y. vienu metu patenka į tą patį segmentą. Kartais žmonės eina labai arti vienas paskui kitą. Šių situacijų ši metodika negali teisingai atpažinti, todėl tam tikra paklaida neišvengiama.

Buvo išbandyti HOT algoritmai krypties atpažinimui, tačiau prie daugiau duomenų paprasčiausiai pritrūksta atminties ir programa nutrūksta. Todėl HOT algoritmai gali būti naudojami tik klasteriuose. O TBI apmokyti atpažinti krypties neįmanoma, nes jis sugeba tik sumažinti erdvinį triukšmą pirmame lygyje.

Su kryptimis buvo atliktas dar vienas eksperimentas, kai visame segmente buvo išdėliotos mažos sritys, kuriose buvo tikrinama ar ten yra judėjimas su apmokytu HTM tinklu. Šis būdas nepasiteisino, nes sritys buvo per mažos. Tada buvo nuspręsta bandyti dėti tris didesnes sritys, viršuje, viduryje ir apačioje segmento. Praeivis eidamas jas kliudytų paeiliui ir tada būtų žinoma kuria kryptimi buvo eita.



49 pav. Juosto aktyvumo pokyčiai įeinant žmogui į patalpą

Labai sudėtingose situacijose būtų galima teigti, kad jei eina keli žmonės paeiliui, tai jie eina ta pačia kryptimi.

Antras HTM tinklas yra toks pat kaip ir pirmas, tik skiriasi sensoriaus dydis, nes juostelės yra siauresnės. Buvo naudojamos 230x20 pikselių dydžio juostos. Apmokymo strategija ta pati, pirmiausia apmokoma su pirma juoste, tada antra ir galiausia su trečia.

2.2.6. Fazė 4. Praeivių skaičiavimas

Paskutinėje fazėje skaičiuojama kiek žmonių buvo aptikta apskritai, kiek įėjo ir kiek išėjo.

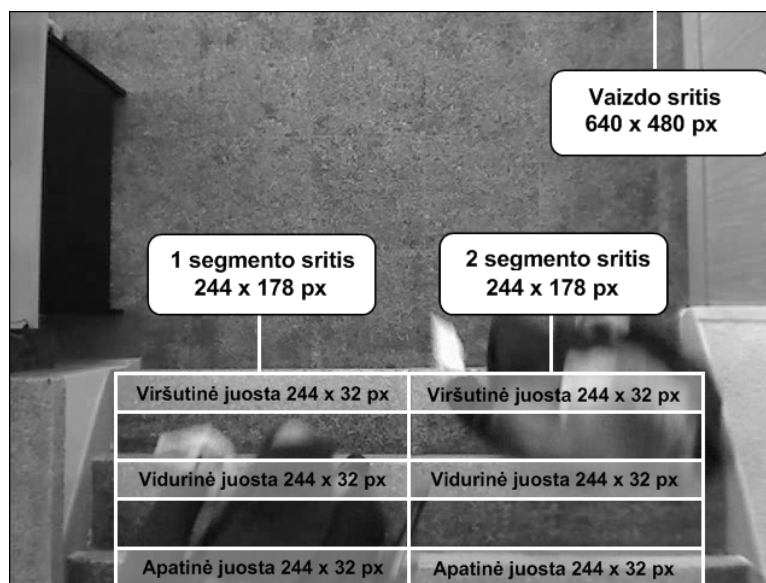
Pirmiausia visos tikimybės gautos ir segmento ir juostelių konvertuojamos į 0 arba 1 reikšmes. Labai aukštos tikimybės konvertuojamos į 1, o mažesnės į 0. Trumpi pikai yra atmetami visai, o įtrūkimai sulyginami su greta esančiomis reikšmėmis. Tokiu būdu pašalinami triukšmai.

Buvo išanalizuotos visos situacijos ir prieita išvada, kad kryptį galima nustatyti pagal juostelių aktyvumą segmente kol jame dar yra žmogus. T.y. skaičiuoti aktyvumo, kai žmogaus ten nėra, neverta.

2.2.7. Algoritmo išplėtimas platesniems praėjimams

Išbandžius su gana siauromis durimis, buvo atlikti tyrimai ir su platesnėmis durimis, kai laisvai gali prasilenkti du ar net trys žmonės. Čia pateikta išplėtimų idėja ir realizacija.

Pagrindinė idėja yra praėjime išdėlioti tiek segmentų, kiek natūraliai praeiviai gali prasilenkti nesistumdant. Platesnėms durims vieno segmento jau nepakanka. Išplėtimui išbandyta naudoti du segmentus.



50 pav. Dviejų segmentų pavyzdys platesniam praėjimui

Kiekviename segmente, šiuo atveju dvejuose, skaičiuojama praeivio jame buvimo tikimybė. Jeigu tam tikrame vaizdų intervale aptinkamas praeivis, tuomet žiūrima į juostelių aktyvumo nuoseklumą ir taip atpažįstama kryptis. Matydami, kad segmente buvo žmogus ir žinodami kuria kryptimi jis judėjo, galime prie balanso pridėti atitinkamai -1 arba +1.

Apmokymui buvo naudojama nedidelė, vos 30 procentų, duomenų dalis, po 3000 paveikslėlių kiekvienam tinklui. Visi kiti duomenys buvo skirti testavimui.

Padidinus segmentų kiekį, skaičiavimų apimtys gerokai išaugo, nes dabar reikia atlikti 8 atpažinimus (2 segmentai, 6 juostelės), kai prieš tai pakakdavo 4 (1 segmentas, 3 juostelės).

Praeivių skaičiavimo idėja lieka ta pati, bus numanoma, kad realiai žmogus gali būti viename segmente arba abejuose, todėl teks tikrinti abejų segmentų panašumą pagal tai kokia judėjimo kryptis, pagal laiką kada aktyvumas prasidėjo ir kada baigiasi. Jei segmentai pagal šiuos požymius labai panašūs, tai bus galima teigti, kad praėji vienas žmogus, o ne du.

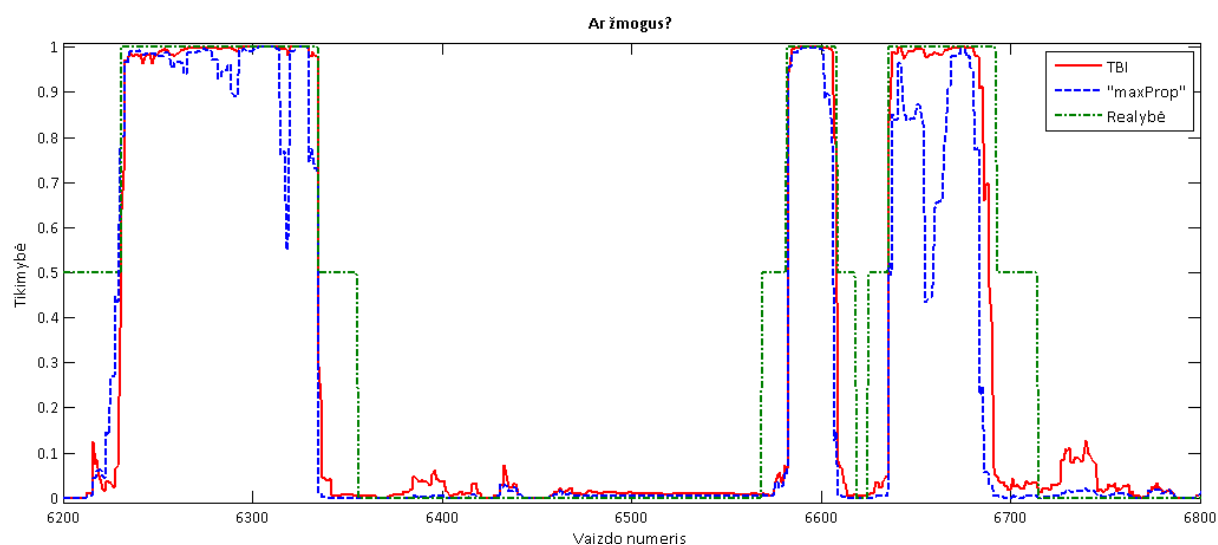
3. TYRIMŲ REZULTATAI

Buvo atlikti bandymai su siauru (vienam žmogui) ir platesniu (2-3 žmonėms) praėjimais, be to, algoritmai išbandyti realiuoju laiku „Technorama 2011“ parodoje.

3.1. Vieno segmento rezultatai

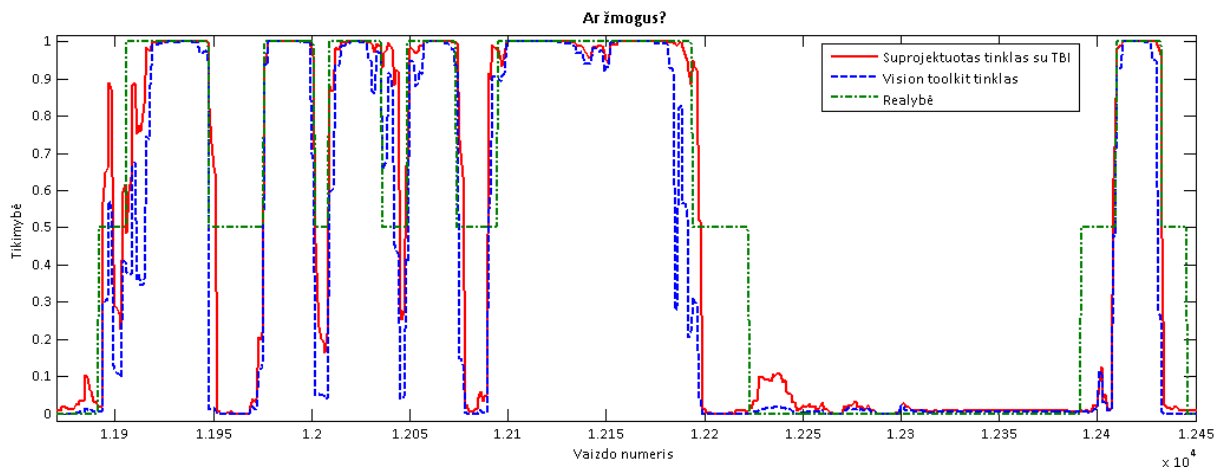
Algoritmai buvo išbandyti 53-joms realioms situacijoms, kai žmonės įeina į pastatą arba išeina. Visos situacijos buvo suklasifikuotos teisingai. Rezultatai rodo, kad atskirti paėiliui einančius praeivius įmanoma, jei yra bent keturi tušti kadrai tarp abejų praeivių, o tai sekundžių dalys laike.

Buvo atlikti keli įvairių tinklų palyginimai pagal tikslumą ir spartą. Tikslumas tarp TBI ir „maxProp“ parodyti 51 paveiksle. Taip pat buvo palygintas sukonstruotas trijų lygių tinklas ir su „Vision toolkit“ septynių lygių tinklu (52 paveikslas).



51 pav. TBI ir "maxProp" atpažinimo palyginimas

Žalia spalva pavaizduota realybė (0.5 – tik maža dalis žmogaus matoma, kai jis įeina arba palieka patalpą, 1 – praeivis pilnai matomas segmente). Akivaizdu, kad su TBI rezultatai yra stabilesni.

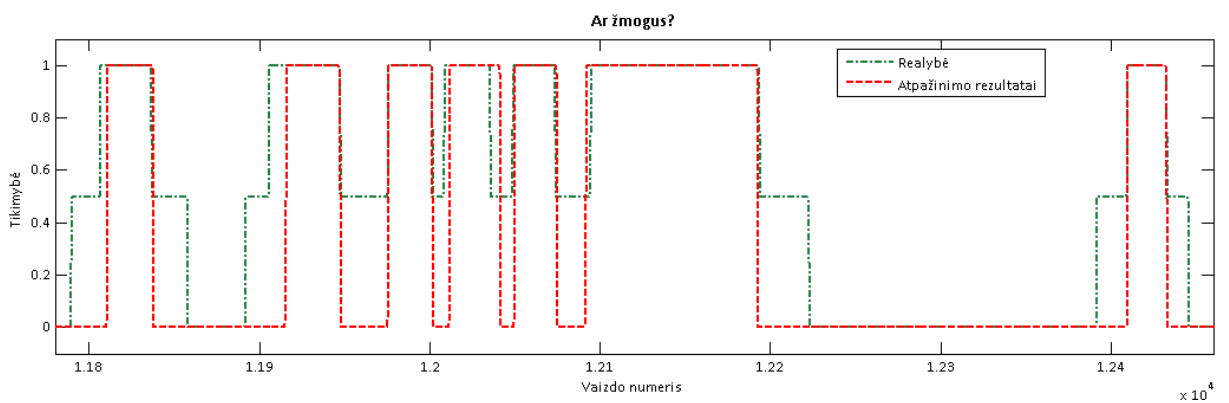


52 pav. Suprojektuoto tinklo ir „Vision toolkit“ tinkle palyginimas

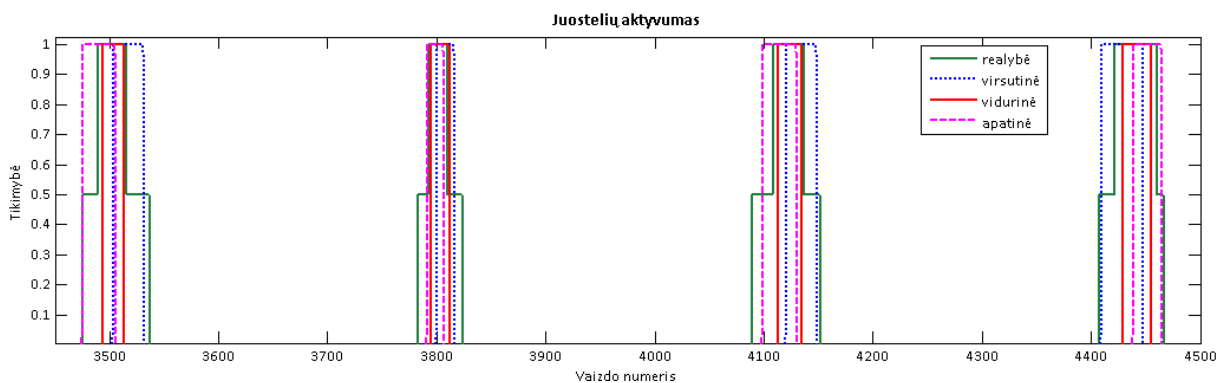
Abu tinklai duoda gana tikslius rezultatus net jei yra labai maži laiko intervalai tarp paskui vienas kitą einančių praeivių.

Pagal spartą, sukonstruotas HTM tinklas su įjungtu TBI, yra 1.6 karto spartesnis negu „Vision toolkit“ paruoštas tinklas. Tai parodo, kad suprojektuotas ir paruoštas tinklas yra geriau priderintas šiai problemai nei siūlomas „Vision toolkit“ tinklas, nes pastarąjį pralenkia sparta, o tikslumu beveik neatsilieka.

Dalis atpažinimo rezultatų po triukšmų pašalinimo ir diskretizavimo yra žemiau pateiktuose grafikuose. Turint tokius duomenis, nesunkiai pėsčiusius galima suskaičiuoti pagal galinių frontų kiekį.



53 pav. Žmonių atpažinimo duomenys buvo apdoroti naudojant slenksčius



54 pav. Juostelių aktyvumo duomenys buvo apdoroti naudojant slenksčius

Rezultatai geri, tikslumas ir sparta daug žadantys.

3.2. Dviejų segmentų rezultatai

Dalis rezultatų pavaizduota žemiau esančiame paveiksle, o visi rezultatai yra suskirstyti į keturias grupes.

9 lentelė. Eksperimento rezultatai

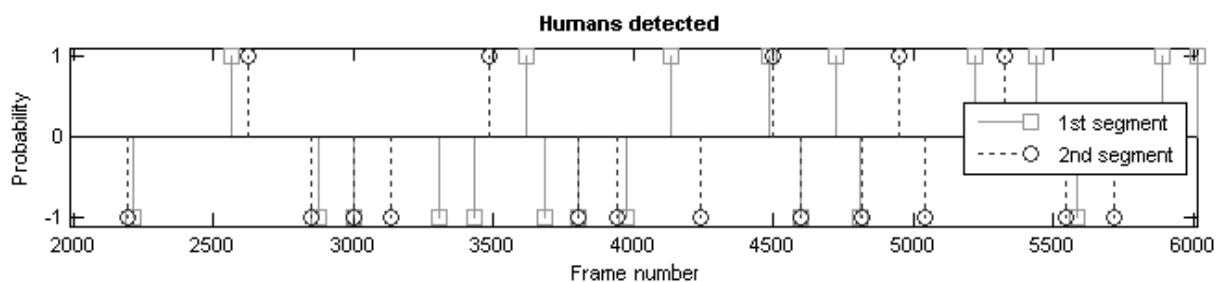
Rezultatų grupės numeris, pavadinimas	Žmonės	Rezultatai
1, tikslu	58	76.32%
2, žmogus eina per vidurį	10	13.16%
3, žmogus matomas abejuose segmentuose	6	7.89%
4, klaidos	2	2.63%
Viso:	76	100.00%

Pirma rezultatų grupė yra pati tiksliausia ir aiškiausia. Jei praeivis eina per kairįjį segmentą, tai ten matoma aukšta tikimybė, jei per dešinįjį – tai didelė tikimybė yra dešiniajame segmente. Kitaip tariant, rezultatas tuomet yra visiškai tikslus. Tokių situacijų duomenyse buvo net 76.32 procentai.

Antroje grupėje yra tos situacijos, kai praeivis eina per abiejų segmentų vidurį, tačiau yra aptinkamas tik viename iš tų segmentų. Iš esmės jokios klaidos nėra, nes realiai eina vienas, o aptinkamas irgi vienas. Tokių situacijų 13.16 procentų.

Trečioji grupė sudėtingesnė, nes praeivis eina per vidurį abiejų segmentų, tačiau aptinkamas abejuose. Tokiu atveju sistema bando žiūrėti ar abu segmentai yra panašūs pagal juostelių aktyvumą ir ar panašūs pagal laiko momentus, kuomet buvo užfiksuotas žmogus segmentuose. Jei abiejų segmentų panašumas didelis, laikoma, kad praėjo ne du, o vienas žmogus ir tai leidžia gauti tikslų skaičių. Tokių situacijų yra 7.89 procentai.

Paskutinė grupė yra klaidos. Iš viso buvo dvi. Pirmoji, kai trys praeiviai eina vienos kito labai arti ir tokiu būdu visi trys atsiduria abejuose segmentuose vienu metu. Sistemą šią situaciją mato kaip du žmones, o ne tris. O antroji klaida buvo tada, kai mažesnio ūgio pėstysis ėjo viduryje praėjimo ir abu segmentai užfiksavo jį, tačiau nesugebėjo nustatyti krypties.



55 pav. Dalis rezultatų abiem segmentams. 1- vienas žmogus įėjo, -1 – vienas išėjo.

Susumavus visas tris grupes, kurias pavyko teisingai klasifikuoti, gauname 97.37 procentų tikslumą. Be abejonės, realių situacijų būtų gerokai daugiau, tačiau šie bandymai parodė, kad net ir dviems segmentams idėja dar pakankamai gyvybinga ir gali būti pritaikoma.

3.3. HTM ir optinio srauto (Optical flow) metodų palyginimas

Buvo atlikti papildomi tyrimai bandant palyginti HTM ir optinio srauto (angl. optical flow) metodų taikymą pėsčiųjų skaičiavimui, o rezultatai paskelbti atskiru straipsniu (žr. 4 priedą). Optinio srauto algoritmai realizuoti „MatLab“, o HTM metodai „Python“ kalboje, todėl vertinant išvadas reiktų tai turėti omenyje. Palyginimo išvados ir pastebėjimai:

- abu metodus prieš naudojant reikia sukalibruoti. Su optiniu srautu tai padaryti yra lengviau, tereikia nurodyti kelis parametrus, kaip sekamo objekto dydis ir jį jau galima leisti. HTM metodai veiks tik tada, jei bus apmokyti abu HTM tinklai, todėl būtina paruošti realių apmokymo duomenų ir atlikti mokymą;
- optinis srautas veikia taip pat tiksliai nepriklausomai nuo fono įvairumo, tačiau iškyla tikslumo problemų, jei nuo lempų dažnio ar dėl kitų priežasčių vaizde atsiranda judančios juostos. HTM tokio pobūdžio triukšmai nėra svarbūs, tačiau dėl fono pasikeitimų sistema gali „apakti“, jei nebuvo tinkamai apmokyta;
- optinis srautas gali būti vieną kartą paruoštas ir naudojamas įvairiuose kontekstuose beveik be žmogaus įsikišimo, o HTM būtina apmokyti panaudojant bent dalį duomenų iš naujo konteksto;
- optinis srautas yra maždaug du-tris kartus spartesnis negu HTM metodas. Jis tampa dar spartesnis, jei vaizde yra tik maži judesiai. HTM būtų gerokai lėtesnis, jei reiktų naudoti daugiau segmentų, be to, jam vis tiek reikia gaišti laiką vykdant atpažinimą, net jei judesys yra mažas;
- abu metodai gali praleisti kai kuriuos kadrus be apdorojimo. Tai labai svarbi savybė, nes gana dažnu atveju, tuščių vaizdų gali būti net iki 50-90 procentų, kai retkarčiais pasirodo vienas kitas praeivis. Visas tas laikas gali būti panaudojamas atlikti kokį nors naudingą darbą arba tiesiog laukti naujų duomenų;
- svarbiausias skirtumas tarp abiejų metodų yra tai, kad HTM išmoksta atpažinti praeivius ir išskirti juos iš kitokių objektų, kurie nedomina. Optinis srautas to negali. Pavyzdžiui, HTM gali lengviau atmesti šešėlius, jei buvo gerai apmokytas, o optinis srautas šešėlių gali būti užskaitytas kaip praeivį. Be to HTM gali būti apmokytas skirti ne dvejų tipų kategorijas, o kelias, t.y. atpažinti ne tik

žmones, bet ir kitokius objektus. Optinis srautas galėtų skirti tik pagal objekto dydį;

- abu metodai yra gana tikslūs šiame kontekste ir abu netikslūs, kai keli praeiviai vizualiai susilieja. HTM tikslumas buvo 97.37 procentai, o optinio srauto 94.70 procentų;
- abu metodai yra gerokai tikslesni aptinkant objektą, nei nustatant jo judėjimo kryptį. Ši savybė gali būti panaudota, kai tariamame objekte yra tik vienas praėjimas ir visi žmonės kažkuriuo paros metu būtinai objektą palieka. Tuomet aptiktų praeivių kiekis padalintas iš dviejų yra lygus apsilankiusių kiekiui. To kartais pakanka.

Abu metodai šiam uždaviniui spręsti pasiteisino ir yra panašūs savo tikslumu ir sparta. Esminis skirtumas yra tai, kad vieną reikia apmokyti, o kito ne.

3.4. „Technorama 2011“ parodos lankytojų skaičiavimo rezultatai

Parodos lankytojai buvo skaičiuojami realiuoju laiku.



Technorama 2011

Kiek žmonių apsilankė parodoje?
Žmonių srauto monitoringo sistema

Naujausiai aptiktas: 2011-05-04 15:28:01

Jau apsilankė: **1162**

Šiuo metu parodoje yra: **115**

Rėmėjai:

RMD Technologies **KAUNO AUTOBUSAI**

Autorius:
Saulius Sinkevičius
saulius@sinkevicius.net

Vadovas:
prof. habil. dr. Rimvydas Simutis
rimvydas.simutis@ktu.lt

56 pav. Grafinis interfeisas parodos lankytojų skaičiavimo rezultatams atvaizduoti

Paroda vyko daugiau nei šešios valandos. Per tą laiką smarkiai keitėsi apšvietimas, nes pirmoje dienos pusėje buvo apsiniaukę, o po pietų išlindo saulė ir apšvietimas iš koridoriaus tapo labai intensyvus. Žmonės ėjo srautais, dažnai beveik grūsdamiesi, todėl duomenys gavosi labai realūs.

Po parodos dalis duomenų buvo peržiūrėta ir taip patikrintas sistemos tikslumas. Visos situacijos sugrupuotos į tris grupes.

10 lentelė. Parodos rezultatai

Situacija	Atpažino	Padaugino	Suliejo	Neaptiko	Realus žmonių kiekis	Sistemos aptiktų žmonių kiekis	Žmogaus aptikimo tikslumas, %	Krypties nustatymo tikslumas, %
Vienas žmogus	712	41		101	813	753	83,37	76,98
Eilė	281	26	49	6	336	307	77,62	73,35
Prasilenkia	169	11	30	10	209	180	76,82	58,66
Viso:	1162	78	79	117	1358	1240	80,94	73,44

Pirmoje grupėje yra paprasčiausios situacijos, kai praeina vienas žmogus. Šios grupės tikslumas turėtų būti didžiausias. Dėl to, kad sistema skaičiavo kas 5 vaizdą ir tik radus aukštą žmogaus buvimo segmente tikimybę, „dairosi“ į gretimus kadrus, nemaža dalis tokių praeivių buvo nepastebėta. Dažniausia tada, kai jie bėgdavo arba labai greitai eidavo. Taip pat dalis neaptikta, dėl to, kad tie praeiviai buvo smulkesni ir sistema jų nepažindavo. Kai kuriose situacijose, vienas žmogus atpažįstamas kaip du, tai irgi tikslumą sumažino. Bendras tikslumas vieno žmogaus atpažinimui gavosi 83,37 %.

Antroje grupėje yra tos situacijos, kai keli praeiviai eina vienas paskui kitą arba šalia vienas kito, tačiau visi ta pačia kryptimi. Šios situacijos sudėtingesnės, nes jei tik praeiviai eina pernelyg arti vienas kito, jie suliejami. Dalis tokių praeivių buvo neaptikti arba sudvejinti. Bendras grupės tikslumas yra 77,62 %.

Trečia grupė yra sudėtingiausia, nes joje yra tos situacijos, kurių metodika ir neturėtų gebėti teisingai atskirti, nes keli žmonės atsiduria tame pačiame segmente vienu metu. Bendras grupės tikslumas yra 76,82 %.

Krypčių tikslumai gavosi gana prastas, nes tikslumas skaičiuojamas teisingai aptiktoms situacijoms, todėl visoms realioms situacijoms jis yra dar mažesnis ir tai rodo, kad juostelių naudojimas arba nepakankamai suderintas arba nepasiteisino.

Bendras parodos lankytojų atpažinimo tikslumas 80,94 %, o krypties – 73,44 %.

IŠVADOS

Atlikus literatūros analizę, sukūrus metodiką pėsčiųjų srauto monitoringui, išbandžius prototipą, patobulinus metodiką atsižvelgiant į prototipo trūkumus, sukūrus realaus laiko sistemą ir išbandžius ją realiomis sąlygomis, matome, kad:

1. Problema dar nėra išspręsta, o egzistuojantys sprendimai veikia tik labai apibrėžtomis sąlygomis, bet kokie nukrypimai nuo apribojimų mažina tikslumą drastiškai. Dažniausiai problema sprendžiama įvairiais barjeriais, tačiau vis daugėja bandymų panaudoti kompiuterinę regą;
2. Pasirinkto sprendimo didžiausias trūkumas, kad vienoje apibrėžtoje srityje negali būti du žmonės vienu metu, antraip jie bus matomi tarsi vienas. Šią problemą dalinai pavyksta išspręsti turint tris aktyvumo stebėjimo juosteles, kurias žmonės eidami kerta. Kai eina keli žmonės paėiliui, tai vidurinės juostelės aktyvumas trumpam užgęsta, jei yra nors menkiausias tarpas tarp dvejų žmonių. Tačiau, jei tarpo nėra, žmonės susilieja į vieną;
3. Suprojektuotas trijų lygių tinklas yra gerokai greitesnis už esamus „NuPIC“ programinėje įrangoje, nors tikslumas dar pakankamas;
4. Atpažinimo rezultatai su įjungtu TBI yra geresni, tačiau pats atpažinimas sulėtėja nuo kelių iki keliolikos kartų, todėl TBI dar netinka realaus laiko skaičiavimams, juolab, kad jį galima taikyti tik pirmame laikinių grupių lygyje kai norima tinklą padaryti atsparesniu smulkiam erdviniam triukšmui;
5. Eksperimentai parodė, kad labai didelę įtaką turi apšvietimas. Todėl sistema turi būti labai gerai apmokyta tais duomenimis, kuriame kontekste bus naudojama. Jei nepakanka sensoriaus ploto, jį reikia didinti. Jei net ir to mažai, galima bandyti projektuoti daugiau lygių turintį tinklą taip didinant tikslumą, bet prarandant spartą;
6. Pasiūlytos metodikos tikslumas platesniam praėjimui yra mažesnis nei siauram, tačiau prie nesudėtingų sąlygų vis dar aukštas. Platesniuose praėjimuose iškyla problema kaip nustatyti ar gretimuose segmentuose matomas tas pats žmogus ar skirtingi ir jei jie vizualiai labai susilieję, atskirti ne visada pavyksta. Bandymų metu pastebėta, kad daugiau nei dviejų segmentų naudoti nelabai pavyktų, nebent kamerą įmanoma pakabinti labai aukštai, nes kitaip ji jau negali aprėpti viso praėjimo;
7. Palyginus HTM su optinio srauto metodu, akivaizdu, kad kai kada pastarasis yra pranašesnis, o kitais atvejais atvirkščiai, todėl būtų galima galvoti apie abiejų metodų apjungimą ir taip pasiekti stabilesnį tikslumą.

SUMMARY

Monitoring of Pedestrian Traffic Using Hierarchical Temporal Memory Algorithms

The main purpose of this paper is to investigate the application of Hierarchical Temporal Memory (HTM) mechanism for pedestrian traffic calculation in a public places using WEB cameras. The proposed approach distinguishes humans from other objects in a current video frame, identifies the movement in both directions and calculates balance of the pedestrians' traffic. The proposed methodic can be used even for wider passage-ways. As a result, all people entering and leaving room can be counted and information about pedestrian-traffic can be acquired in real time. This information is very useful for further traffic monitoring and can be used for various tasks.

Pėsčiųjų srauto monitoringas hierarchinės laikinės atminties atpažinimo algoritmu pagrindu

Pagrindinis šio darbo tikslas yra pritaikyti hierarchinę laikinę atmintį (Hierarchical Temporal Memory - HTM) žmonių srauto balansui skaičiuoti viešosiose vietose su WEB kamera. Pasiūlyta metodika leidžia apmokyti sistemą atskirti žmones nuo kitų objektų, sekti judėjimo kryptį į abi kryptis ir paskaičiuoti srauto balansą. Atlikti tyrimai rodo, kad metodika tinka ir platesniems praėjimams bei skaičiavimams realiu laiku. Ši informacija gali būti panaudota optimizuojat sistemas, kuriose žmonių srautai, eilės bei apkrautumas yra labai svarbūs.

LITERATŪRA

1. Hawkins J., Blakeslee S. On Intelligence. New York: Times Books, 2005, 174 p;
2. Hawkins J., George D. Hierarchical Temporal Memory. Concepts, Theory, and Terminology. 2007. Prieiga per internetą: <http://numenta.com/Numenta_HTM_Concepts.pdf>;
3. Fang Zhu, Junhua Gu, Ruixia Yang, Zhifeng Zhao. Research on Counting Method of Bus passenger Flow Based on Kinematics of Human Body and SVM. International Conference on Computer Science and Software Engineering, Wuhan, Hubei, 2008. p. 201-204;
4. Duan-Yu Chen, Chih-Wen Su, Yi-Chong Zeng, Shih-Wei Sun, Wei-Ru Lai, Hong-Yuan Mark Liao. An Online People Counting System for Electronic Advertising Machines. IEEE International Conference on Multimedia and Expo, New York, 2009. p. 1262–1265;
5. Chunhui Zhao, Quan Pan, Stan Z.Li. Real Time People Tracking and Counting in Visual Surveillance. The Sixth World Congress on Intelligent Control and Automation, Dalian, 2006. p. 9722–9724;
6. Ying Xin, Guangmin Sun, Qiang Wu. A Preprocessing Method for Tracking and Counting Pedestrians in Bus Video Monitor. International Conference on Industrial Informatics, Daejeon, 2008. p. 1689-1693;
7. Li-hua Zou, Yun-cheng Liu. A New Algorithm of Counting Human Based on Segmentation of Human Faces in Color Image. International Conference on Computational Intelligence and Security, Beijing, 2009. p. 505-509;
8. Z. Jiang, D. Q. Huynh, W. Moran, S. Challa, and N. Spadaccini. Multiple Pedestrian Tracking using Colour and Motion Models. Digital Image Computing: Techniques and Analysis, Sydney, Australia, 2010, Nov/Dec. p. 328-333;
9. David Ryan, Simon Denman, Clinton Fookes, Sridha Sridharan. Scene Invariant Crowd Counting for Real-Time Surveillance. 2nd International Conference on Signal Processing and Communication Systems, Gold Coast, 2008. p. 1-7;
10. Kulapat Jaijing, Pakorn Kaewtrakulpong, Supakorn Siddhichai. Object Detection and Modeling Algorithm for Automatic Visual People Counting System. 6th International Conference on Electrical Engineering/Electronics, Computer,

- Telecommunications and Information Technology, Pattaya, Chonburi, 2009. p. 1062-1065;
11. Chih-Wen Su, Hong-Yuan Mark Liao. A Vision-based People Counting Approach Based on the Symmetry Measure. IEEE International Symposium on Circuits and Systems, Taipei, 2009. p. 2617–2620;
 12. Lu Wang, Nelson H.C. Yung. Crowd Counting and Segmentation in Visual Surveillance. 16th IEEE International Conference on Image Processing (ICIP), Cairo, 2009. p. 2573 – 2576;
 13. Zhe Lin, Larry S. Davis. Shape-Based Human Detection and Segmentation via Hierarchical Part-Template Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010. p. 604-618 vol. 32, no. 4;
 14. E.Terada, D. Yoshida, S. Oe J. Yamaguchi. A Method of Counting the Passing People by Using the Stereo Images. International Conference on Image Processing, Kobe, 1999. p. 338-342 vol. 2;
 15. B. D. Nam, S. I. Kang, H. K. Hong. Pedestrian Detection System based on Stereo Vision for Mobile Robot. Frontiers of Computer Vision (FCV), 2011. p. 1–7;
 16. Hao Sun, Cheng Wang, Boliang Wang. Night Vision Pedestrian Detection Using a Forward-Looking Infrared Camera. Multi-Platform/Multi-Sensor Remote Sensing and Mapping (M2RSM), 2011. p. 1 – 4;
 17. Yu Ming, Liu Wei. A New Method for Passenger Flow Counting System Based on Surveillance Video. Second International Conference on Intelligent Networks and Intelligent Systems, Tianjin, 2009. p. 453-456;
 18. Thiago Teixeira, Andreas Savvides. Lightweight People Counting and Localizing in Indoor Space Using Camera Sensor Nodes. First ACM/IEEE International Conference on Distributed Smart Cameras, Vienna, 2007. p. 36-43;
 19. Kazuhiko Hashimoto, Katsuya Morinaka, Nobuyuki Yoshiike, Chjihiro Kawaguchi, Satoshi Matsueda. People Count System Using Multi-Sensing Application. International Conference on Solid State Sensors and Actuators, Chicago, IL, 1997. p. 1291-1294 vol. 2;
 20. S. Umesh Babul, C. S. Kumar, R. V. Raja Kumar. Modelling and Simulation of a Sensor Network for Tracking a Moving Object. Third International Conference on Intelligent Sensing and Information Processing, 2005. p. 200–206;

21. Sistemų teorijos paskaitų medžiaga. Prieiga per internetą: <<http://www.ifko.ktu.lt/~raimund/>>;
22. WinSWOT manual. Prieiga per internetą: <http://www.cymeon.com/swot/swot_man.asp>;
23. SSGG ir MPP modeliavimo programa. Prieiga per internetą: <<https://opensvn.csie.org/SWOT2/trunk/docs/Ataskaita1.doc>>;
24. Plukas K. Skaitiniai metodai ir algoritmai. Vadovėlis. Kaunas: Naujasis Lankas, 2000;
25. The MathWorks. Prieiga per internetą: <<http://www.mathworks.com/access/helpdesk/help/helpdesk.html>>;
26. Paskaitų medžiaga: Sistemos funkcionavimo tikslų, identifikacijos ir filtracijos skyriai;
27. Noise reduction. Prieiga per internetą: <http://en.wikipedia.org/wiki/Noise_reduction>;
28. George D., Jaros B. The HTM Learning Algorithms. Prieiga per internetą: <http://numenta.com/for-developers/education/Numenta_HTM_Learning_Algos.pdf>, 2007.

TERMINŲ IR SUTRUMPINIMŲ ŽODYNAS

11 lentelė. Naudojamos sąvokos, terminai ir sutrumpinimai

Sąvoka	Paaškinimas
CCTV (angl. closed-circuit television)	Vaizdo kamera.
DB	Duomenų bazė.
Flash	Atpažinimo arba apmokymo būdas, kai po vieną nuosekliai sensoriui paduodami vaizdai.
fps (angl. frame rate)	Kadrų kiekis per sekundę.
Gabor	Linijinis filtras kontūrų aptikimui.
hardcoded	Laikinių grupių apmokymo algoritmas
HOT (angl. higher-order temporal)	Atpažinimo algoritmai aukštesniems laikinių grupių lygiams.
HTM (angl. hierarchical temporal memory)	Hierarchinė laikinė atmintis
IP (angl. internet protocol address)	Įrenginio identifikatorius interneto tinkle.
IR (angl. infrared)	Infraraudonieji spinduliai.
MatLab	Techninio apdorojimo programavimo kalba.
maxProp	Laikinių grupių skaičiavimo algoritmas.
MPP	Miglotasis pažintinis kelias.
Multi-resolution	Apmokymo būdas, lygiagrečiai apmokoma su mažesniais-didesniais paveikslais.
Numenta	Jeff Hawkins įkurta įmonė.
NuPIC framework	Karkasas eksperimentuoti su HTM metodais.
Optical flow	Optinio srauto metodas.
Python	Programavimo kalba.
px (angl. pixel)	Taškų matavimas vaizdo dydžiui.
Spatial pooler	Erdvinių savybių mazgas.
SSGG	Stiprybės, silpnybės, galimybės ir grėsmės.
SVM (angl. support vector machine)	Metodas mokyti sistemą klasifikuoti.
SVMClassifierNode	SVM klasifikatorius-mazgas.
SWOT (angl. Strengths, Weaknesses, Opportunities, and Threats)	Analizė skirta įvertinti stiprybes, silpnybes, galimybes ir grėsmes.
TBI (angl. time based inference)	Laiku pagrįstas atpažinimas.
Temporal groups	Laikinės grupės.
TemporalPoolerNode	Laikinių grupių mazgo iškiepis.
Vision framework	Karkasas dirbti su HTM metodais vaizdams.
Vision toolkit	Priemonė apmokyti HTM tinklus.

PRIEDAI

1 priedas. Sutartis dėl studentų mokslinių tyrimų vykdymo

SUTARTIS DĖL STUDENTŲ MOKSLINIŲ TYRIMŲ VYKDYMO

2010 m. sausio 29 d. Nr. K-1393/SMT 10P-168

Kaunas
(sudarymo vieta)

Lietuvos mokslo taryba (toliau – LMT), atstovaujama LMT Mokslo fondo direktorės Aušros Vilutienės, veikiančios pagal LMT nuostatus, ir institucija (firma), kurioje bus atliekami studentų moksliniai tyrimai, (toliau – institucija), Kauno technologijos universitetas (KTU), atstovaujama studijų prorektoriaus Prano Žiliuko, veikiančio pagal rektorius isakymą 2007-11-15 Nr. A655 ir studentas

Saulius Sinkevičius, 38303211179, Nemuno g. 34-54, LT-35237 Panevėžys

(nurodoma vardas, pavardė, asmens kodas ir nuolatinė gyvenamoji vieta)

studijuojantis KTU Informatikos fakultetas, I kursas, informatikos magistrantūra,

(nurodoma aukštoji mokykla, fakultetas, kursas, studijų programa)

vadovaudamiesi Lietuvos Respublikos teisės aktais sudaro šią sutartį:

I. BENDROSIOS NUOSTATOS

1. Ši sutartis sudaroma, įgyvendinant Lietuvos mokslo tarybos projekto „Studentų mokslinės veiklos skatinimas“, vykdomo pagal finansavimo ir administravimo sutartį Nr. VP1-3.1-ŠMM-01-V-01-002, pasirašytą 2009 m. kovo 27 d. tarp Europos socialinio fondo agentūros, Švietimo ir mokslo ministerijos bei Lietuvos mokslo tarybos, veiklą – studentų moksliniai tyrimai (laisvu nuo studijų metu).
2. Studentas vykdys mokslinius tyrimus pagal vadovo parengtą programą, aprašytą bendroje mokslinio darbo vadovo ir studento paraiškoje – anketoje, pateiktoje studentų mokslinių tyrimų konkursui.
3. Šia sutartimi nustatoma, kad:

3.1. Mokslinio darbo vadovas yra prof. habil. dr. Rimvydas Simutis

3.2. Mokslinio tiriamojo darbo tema yra „Hierarchinės laikinės atminties atpažinimo algoritmų taikymas keleivių srauto monitoringui“

3.3. Mokslinių tyrimų atlikimo laikas – 2010 m. vasario 1 d. iki 2010 m. gegužės 31 d.
(nurodoma pradžios ir pabaigos datos)

Bendra mokslinių tyrimų vykdymo trukmė keturi kalendoriniai mėnesiai, mokslinis darbas organizuojamas 20 akademinį valandų darbo savaitės tvarkaraščiu.

3.4. Institucijai, kurioje atliekami tyrimai skiriama iki 312,00 (trijų šimtų dvylikos) Lt. Šios lėšos turi būti naudojamos mokslinių tyrimų sąnaudoms iš dalies kompensuoti, t.y., medžiagoms ar priemonėms, reikalingoms moksliniams tyrimams atlikti, įsigyti. Tinkamomis apmokėti bus tik tos sąnaudos, kurios bus susijusios su šios sutarties vykdymu, pagrįstos išlaidas įrodančių dokumentų patvirtintomis kopijomis ir patirtos nuo 2010 m. vasario 1 d. iki 2010 m. gegužės 31 d.

3.5. Lėšos (stipendija) studentui, visam mokslinių tyrimų laikotarpiui yra 2080,00 (du tūkstančiai aštuoniasdešimt) Lt.

18. Ši sutartis galioja iki visų įsipareigojimų pagal šią sutartį visiško įvykdymo.
 19. Ši sutartis sudaryta 3 egzemplioriais, turinčiais vienodą teisinę galią, po vieną kiekvienai sutarties šaliai.

VI. ŠALIŲ REKVIZITAI

LIETUVOS MOKSLO TARYBA

Biudžetinė įstaiga Lietuvos mokslo taryba, kodas 188716281, Gedimino pr. 3, LT- 01103 Vilnius, tel. (8-5) 2124 933, faksas (8-5) 2618 535, el. p. lmt@ktl.mii.lt, atsisk. sąsk. Nr.LT757300010094690397, AB bankas „Swedbankas“, kodas 73000. Įregistruota Juridinių asmenų registre.

INSTITUCIJA

Kauno technologijos universitetas, Įmonės kodas 111950581, K. Donelaičio g. 73, LT-44029
Kaunas, atsisk. sąsk. SEB bankas LT38 7044 0600 0310 4481
Vadovas Rimvydas Simutis, tel. +370 37 300296

Nurodyti: pavadinimą, įmonės kodą, adresą, banko pavadinimą ir banko sąskaitos Nr.
 Papildomai nurodyti praktikos vadovo vardą, pavardę ir tel. numerį, kuriuo galima būtų susisiekti.

STUDENTAS



Saulius Sinkevičius, 38303211179, AB SEB, LT307044000446611423, mob. +370 652 60978.
(Vytenis Sinkevičius – tėvas – +370 699 94308)

Nurodyti: vardą(us), pavardę, asmens kodą, banko pavadinimą ir banko sąskaitos (banko kortelės Nr. netinka) Nr.
 Papildomai nurodyti studento telefono numerį ir studento tėvų, globėjų, arba kito artimo žmogaus adresą ir telefoną(jeigu yra), kuriuo, esant reikalui, galima būtų susisiekti.

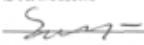
VII. ŠALIŲ PARAŠAI

Lietuvos mokslo taryba


 Aušra Viliutienė
 A.V.


Institucija

 Pranas Žiliukas
 A.V. *studijų prorektorius*
 Pranas ŽILIUKAS


Studentas


 Saulius Sinkevičius

2 priedas. Straipsnis leidinyje „The 5th International Conference on Electrical and Control Technologies“

HUMANS TRACKING IN A PUBLIC PLACES WITH WEB CAMERA USING HIERARCHICAL TEMPORAL MEMORY ALGORITHMS

Saulius SINKEVIČIUS*, Rimvydas SIMUTIS**

*Faculty of Informatics, Kaunas University of Technology, Lithuania

**Department of Process Control, Kaunas University of Technology, Lithuania

Abstract: The main purpose of this work is to investigate the application of alternative cognitive mechanism using Hierarchical Temporal Memory (HTM) for humans tracking in a public places with a web camera. The proposed approach allows to distinguish humans from other objects in a current video frame and to track them in next frames. As a result, all people entering and leaving a room can be counted and information about people-traffic can be acquired. This information is very useful for further traffic monitoring and optimization tasks.

Keywords: object tracking, Hierarchical Temporal Memory, cognitive mechanism, computer vision.

1. Introduction

This paper explores possibilities to use Hierarchical Temporal Memory (HTM) [2] to track humans in a public places, like rooms, trolleybuses, buses, shops, etc. This information can be used removing human-traffic bottlenecks, reducing loads, distributing traffic and adjusting business processes according to human's activity.

Recently, innovative IT company Numenta released a very promising framework [5,6] for a vision cognition tasks, but it's still in development and some essential algorithms are not implemented yet. In this paper we propose some special extensions and alternative use of Numenta methods.

In our application the cognitive mechanism is implemented by cropping different areas from video frames and searching for humans in them. Such areas are fed to HTM network one by one, each time the network estimates the probability of human appearance in that area.

The paper contains five sections. In the second section theoretical background of HTM is overviewed. Third section gives more information about data and HTM network preparation. Fourth section covers a human tracking system and an algorithm of human position

detection. Experimental results are represented in section 5.

2. Hierarchical Temporal Memory

HTM is a memory system that implements the structural and algorithmic properties of the neocortex [3,4]. Instead of programming individual solutions for each problem, this method uses common algorithms to solve many cognitive tasks.

One of basic HTM functions [1] is to discover causes. Anything that interacts with our senses, like physical objects or even abstract things like ideas, can be expressed as patterns. And those patterns always have causes. So, every input pattern is related to concrete causes (Fig. 1).

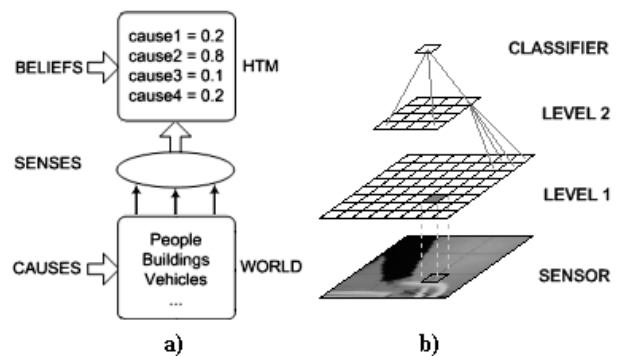


Fig. 1. a) HTM concept b) three levels of the HTM network

A belief is the moment-to-moment distribution of possible causes. Very interesting ability of HTM is to predict what is likely to happen next, when history of observations is known. This is very useful for noise filtering when sounds or other temporal observations or data are presented.

By applying the HTM the spatial-temporal aspects are essential. A spatial property allows creating a finite alphabet of input patterns and a temporal property allows relating input variations in time to same causes.

A HTM network learns using continuously changing data over time, while causes remain relatively stable. Network is organized as tree shaped hierarchy of nodes [1] and each node realizes the same algorithms. Learning data are exposed to a sensor; which is at the bottom of network. Then, the information flows up through the network hierarchy to the top of the network and at each network-level more abstract beliefs are formed. In a learning phase, value of category goes directly to the top node (a supervised learning). By the time, when data comes to the top node, it's already known to what category the input belongs. In the end of learning procedure the hierarchical model of the world is formed. When all levels of the network are learned, we can apply this network to patterns recognition tasks (inference). There are two types of inference: Flash Inference and Time Based Inference (TBI) [7]. TBI can be used only in the first level to reduce a spatial noise of observations. According to the creators of HTM, this method is very useful when solving cognitive problems that are easy for humans, but hard for computers.

3. Preparing Data and HTM network

Humans tracking in a public places starts from picking a good position for a web camera. The easiest way to record all moving objects through some passageway is to do it from above. Then everything what moves below is clearly visible. Trying to record from other directions than top is worse, because some objects can hide others. In our application data are frames, acquired from video camera placed above the doors as shown in Figure 2. In this application video camera with 640x480 pixels resolution is used.

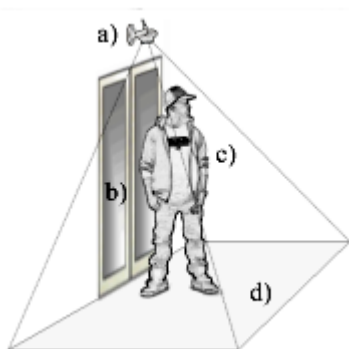


Fig. 2. The observation scheme. Web camera (a) is installed above the doors (b). Visible area is a square (d). Humans (c) enters or leaves a building through the door (b)

The main task of the experiments was to scan whole video frame looking for anything that looks like a human and to track it in the next frames. But to do so, we need to define, what exactly we are looking for. When using observation scheme presented in the Figure 2, the tracked humans have no simple shape that can be cropped and fed to HTM network in learning phase. Shape of human in each frame looks very different, because of moved body parts, like arms and legs. This makes hard to say where one human ends and where the other begins. Another problem relates to coordinates of the human. For example, when a human raises one hand then more space is occupied, and

camera can easily detect some movement between frames, but actually a human didn't moved, only some part of body did.

So, instead of tracking a whole body, we are trying to track only a head (Fig. 3). In this way it is easier to separate one head from another, because the heads tends to be away one from another, even in very crowdie places. Another good thing about tracking heads is that they have a much more stable shape. Head does not changes drastically in few seconds while human is under the camera.



Fig. 3. A few examples from training data sets: a) including only heads; b) including everything else

In cognitive mechanism two categories of objects are used: "a human" and "not a human". Human's category contains a set of head fragments cropped from video frames. This set includes even those heads that are only partially visible, because we expect in video scanning to see a lot of partial heads. All sizes of used heads are shown in Figure 4. Actually the size of cognition area is a bit bigger than an average head is (300x300 pixels).

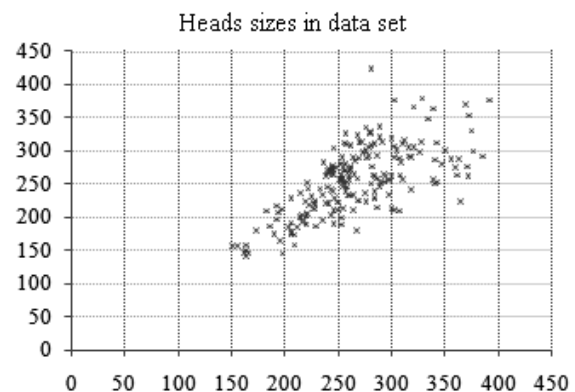


Fig. 4. Heads sizes in the experimental data set, where horizontal axis is width in pixels and vertical axis is height in pixels

Experimental data for the second cognitive category "not a human" was created partially automated. The software script analyzed all frames, cropped random the frame parts and saved them as images. Then only best "not a human" cases were selected, e.g., the door, floor, table, part of the arm and so on. Everything that is not a head is "not a human" (Fig. 3, b).

HTM network was created and trained with Numenta Vision Toolkit (Fig. 5) [8]. This toolkit uses a prepared network for visual experiments with ability to test a trained network and to optimize some parameters. It is fast in learning, despite a big amount of experimental data. To prepare a cognitive network using this Toolkit is a good start to our visual problem. This network can

be easily extended by creating alternative networks with Numenta Vision framework.

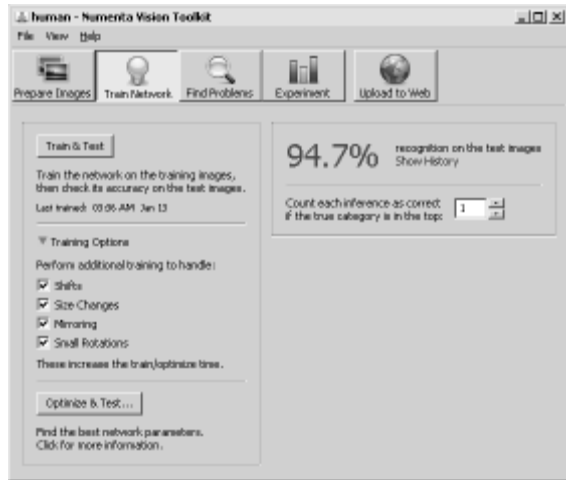


Fig. 5. Vision Toolkit GUI used for HTM network training

After network was trained, it was exported to NuPIC [6] and used for human tracking in a way we need it.

4. Human tracking algorithm

In the further investigation the human tracking task was simplified as follows: first of all, we assumed that only one head can appear in a frame in the same time, like our recorded videos examples are. When a head is detected in a video frame, the scanning stops.

The next modification of algorithm was based on video-frame's segmentation. Whole frame of 640x480 pixels was divided into segments by 80x80 pixels and proposed algorithm was looking for a human in those segments one by one (Fig. 6). Such segmentation makes scanning faster, because number of the recognition areas is reduced.

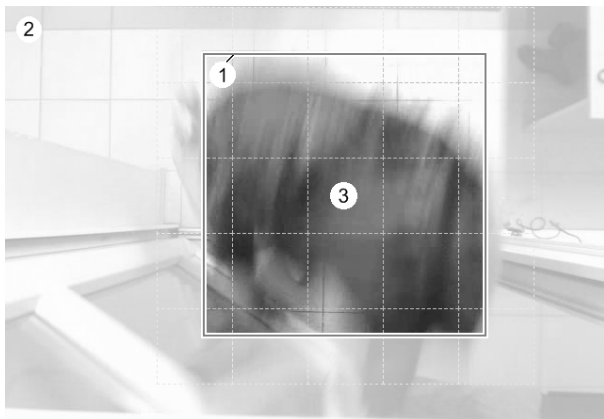


Fig. 6. Whole frame is segmented into squares by 80x80 pixels and active one is marked by number 3. It's a center of cognition area, witch is bounded by box marked 1. The rest area (2) is invisible in this iteration and a sensor sees only a space bounded by line marked 1. In next iteration, another segment becomes center of cognition and whole box shifts accordingly

In our case, most frames are empty and only in some sequences humans are presented. So, we need somehow to decide witch one frames worth to scan, because each

recognition attempt takes a valuable time. We did it by detecting motion in every segment between each two frames. For that reason the former and the current frame was resized to 32x24 pixels, and then the absolute difference between each pixel was calculated. If the difference was big enough, such pixel was counted as a motion (Fig. 7, c). If several such pixels in a chosen segment were identified, the segment was marked for head's recognition (Fig 7, d).

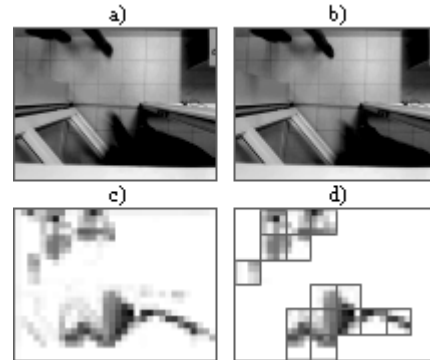


Fig. 7. Motion detection example: a) the former frame b) the current frame c) altered pixels between two frames d) only the segments with big difference is marked for recognition

If the head was found, then we expect to see it again somewhere near the same location in next frame again, because the frames change relatively slowly. Therefore, in the next frame, we can start to scan the segments that are nearly this location and then increase the distance from the location, where a human was saw last time.

5. Experiments and results

As mentioned before, HTM network was trained with the Vision Toolkit [8]. About 50 humans were recorded coming in and out of building. The human's heads were cropped out from those frames and saved as separate files (Fig. 3). The experimental set contains 348 head and 410 "non a head" pictures accordingly. The 85 percent of data was used for training and the rest for testing. The trained network was very accurate even recognizing those heads, that weren't used during training. Recognition accuracy was between 90 and 100 percents.

All experiments were performed on Intel Pentium 4 3 GHz processor with 2 GB RAM. One recognition task using Vision Toolkit network of seven layers takes about 0.2-0.3 seconds to process. If we want to scan whole picture using segments, we need to realize 48 recognitions and it takes about 15 seconds to process one frame. A few examples of object tracking results are shown in the Figure 8.

At the beginning of this research work we expected to scan whole frame and to find all areas where probabilities of head object (network output) are very high. But in this case, we confronted with two problems: a) scanning was very time consuming, and b) some wrong areas of "head location" were detected (Figure 9).

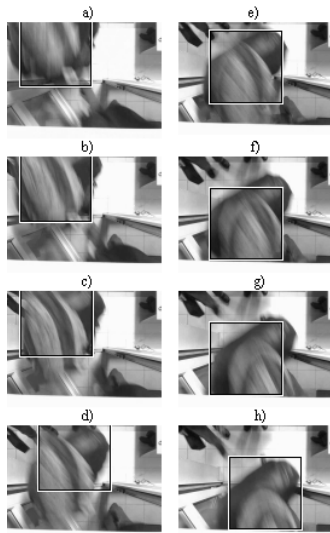


Fig. 8. Human tracking example using algorithms described before

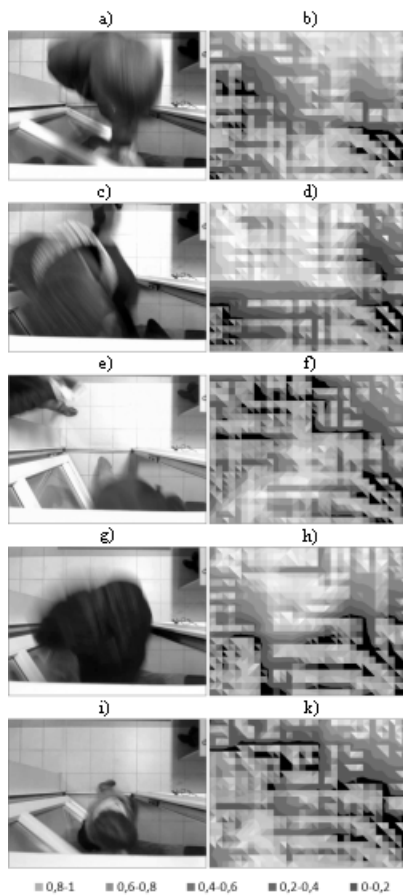


Fig. 9. In the left part some real pictures are presented, in the right - results of very dense scanning. The bright area shows a very big probability of head location there. During the scanning the cognition area 300x300 was shifted by 20 pixels at each time and 768 iterations in total were processed

Therefore, we assumed that only one human is located in the current video-frame and then tried other ways to reduce recognition iterations. Despite of that, recognition procedure lasted about 3 seconds and this is too long for real application.

6. Conclusions and future work

In this paper an alternative cognition mechanism was explored for humans tracking using HTM. The proposed approach allows to distinguish humans from other objects in a current video frame and to track them in next frames. The trained network with seven levels and 204x204 pixels sensor was very slow, and can be useful only if it's enough time for processing. Early attempts shows, that reducing network can increase performance of recognition ten times or more. In this case human tracking could become fast enough for real time processing. The frame segmentation reduces accuracy, but it was implemented only for performance reasons. A future work is to create, to train and to test a much smaller network. There are a huge variation of heads with caps and clothes in real environments. Therefore, data sets for training should be increased at least several times.

7. Acknowledgment

The authors appreciate the financial support of the Lithuanian state science and studies foundation.

8. References

1. Jeff Hawkins, Dileep George. Hierarchical Temporal Memory. Concepts, Theory, and Terminology. http://numenta.com/Numenta_HTM_Concepts.pdf, 2007.
2. Jeff Hawkins, Sandra Blakeslee. On Intelligence. Times Books, New York (2005).
3. Dileep George, Bobby Jaros. The HTM Learning Algorithms. http://numenta.com/for-developers/education/Numenta_HTM_Learning_Algos.pdf, 2007.
4. Dileep George. Ph.D Thesis. How the Brain Might Work: a Hierarchical and Temporal Model for Learning and Recognition, June 2008. p. 161-163.
5. Dileep George, Jeff Hawkins. A Hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex. In Proceedings of the International Joint Conference on Neural Networks, volume 3, p. 1812-1817. IEEE, 2005.
6. NuPIC 1.7.1., <http://numenta.com/for-developers/software.php>.
7. Time Based Inference, <http://www.numenta.com/for-developers/education/tbi-overview.php>.
8. Vision Toolkit 1.7.1, <http://numenta.com/vision/vision-toolkit.php>.

Monitoring of Humans Traffic Using Hierarchical Temporal Memory Algorithms

S. Sinkevičius, R. Simutis, V. Raudonis

Institute of Automation and Control Systems (AVATI), Kaunas University of Technology, Faculty of Electrical and Control Engineering

Studentų str. 48-327, Kaunas LT-51367, Lithuania, +370 37 300296, saulius@sinkevicius.net, rimvydas.simutis@ktu.lt

Introduction

This paper explores possibilities of using Hierarchical Temporal Memory (HTM) [1] mechanism to calculate the humans' traffic balance in a public places, like rooms, trolleybuses, buses, shops, etc. This information can be used for removing of human-traffic bottlenecks, reducing loads, distributing traffic and adjusting business processes according to human's activity.

In recent years there have been many interesting researches and developments concerning to this problem. For human detection and classification tasks various modification of support vector machines (SVM) algorithms [2], human body kinematics [3], shape matching as well as features extraction algorithms and artificial neural networks are used [4]. Furthermore edge-detection algorithms [5], optical flow methods [6], weight map algorithms [7], and motion analysis [8] were applied for human detection tasks. Universality in different context is acquired with background extraction, calculating frame difference or using some sort of filters like Kalman [9] or Gaussian.

None of those methods can handle all issues of problems connected with calculation of humans' traffic in public places. Most difficult situations arise, when lighting changes rapidly, the place is crowded or humans visually merges and so on. In those situations described methods are very inaccurate, because they don't have any memory of previews frames and can't adequately react to changing environment. To solve these problems we choose an approach, based on HTM algorithms [10].

Recently, innovative IT company Numenta released a very promising framework [11,12] for a vision cognition tasks, but some essential algorithms are not implemented yet. In this paper we propose a special extension and alternative use of Numenta methods.

In our application the cognitive mechanism is implemented by recognizing when a human appears in a particular area of the frame and identifying the direction of

the movement. With those information elements the proposed algorithm calculates balance of humans' traffic.

Two HTM networks were trained for different purposes. One instance of HTM network recognizes humans in a particular area of frame and another one identifies direction of movement.

Theoretical background

HTM is a memory system that implements the structural and algorithmic properties of the neocortex [13,14]. Instead of programming individual solutions for each problem, this method uses common algorithms to solve many cognitive tasks.

One of basic HTM functions is to discover causes. Anything that interacts with our senses like physical objects or even abstract things like ideas can be expressed as patterns. And those patterns always have causes. So, every input pattern is related to concrete causes (Figure 1, a). A belief is the moment-to-moment distribution of possible causes.

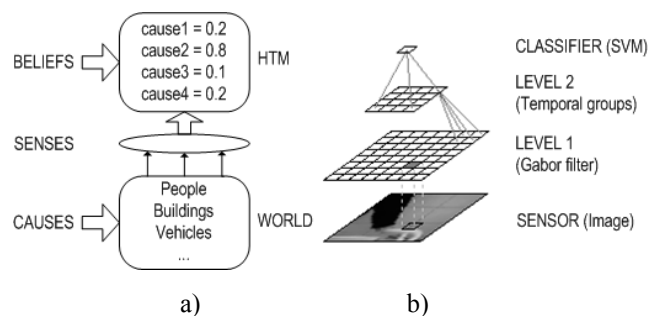


Fig. 1. a) HTM concept b) three levels of the HTM network

By applying the HTM the spatial-temporal aspects are essential. A spatial property allows creating a finite alphabet of input patterns and a temporal property allows relate input variations in time to same causes. A HTM network is trained using continuously changing data over time while causes remain relatively stable.

Network is organized as tree shaped hierarchy of nodes and each node executes the same algorithms (see Figure 1, b). Learning data are exposed to a sensor; which is at the bottom of network. Then, the information flows up through the network hierarchy to the top of the network and at each network-level more abstract beliefs are formed. In a learning phase, value of category goes directly to the top node (a supervised learning). By the time, when data comes to the top node, it's already known to what category the input belongs. In the end of training procedure the hierarchical model of the world is formed. When all levels of the network are trained, we can apply this network to patterns recognition tasks (inference).

There are two types of inference: Flash Inference (FI) (we used “maxProp” mode which computes a more peaked score for the group based on the current input only) and Time Based Inference (TBI) [15]. TBI can help the network to produce better inference results where each successive input is temporally related to the previous one, because the output is computed based on the current and previous inputs. The system takes into account the likelihood of each coincidence to follow another (based on the order of coincidences seen during training) and uses this information to compute the output probabilities after each input. For example, if you have an image application and you are feeding in successive frames of a movie, TBI mode will produce better inference results than FI. The FI is best suited for applications where each successive inference input is completely independent from the previous one.

Experimental setup

The proposed human traffic counting system contains four phases (see Figure 2). In the first phase frames are prepared for processing. In second humans are differentiated from other objects using first HTM network. In third phase, activity of several stripes of the segment is calculated. This activity is used to identify movement direction. And in fourth phase, using movement direction and human presenting probabilities, the balance can be calculated.

Observation setup and preprocessing in phase 1

Humans tracking in a public places starts from picking a good position for a web camera. The easiest way to record all moving objects through some passageway is to do it from above the passage-way. Then everything what moves below is clearly visible. Trying to record from other directions than top is worse, because some objects can hide others. In our application data are frames, acquired from video camera placed above the doors as shown in Figure 3. In this application video camera with 640x480 pixels resolution is used.

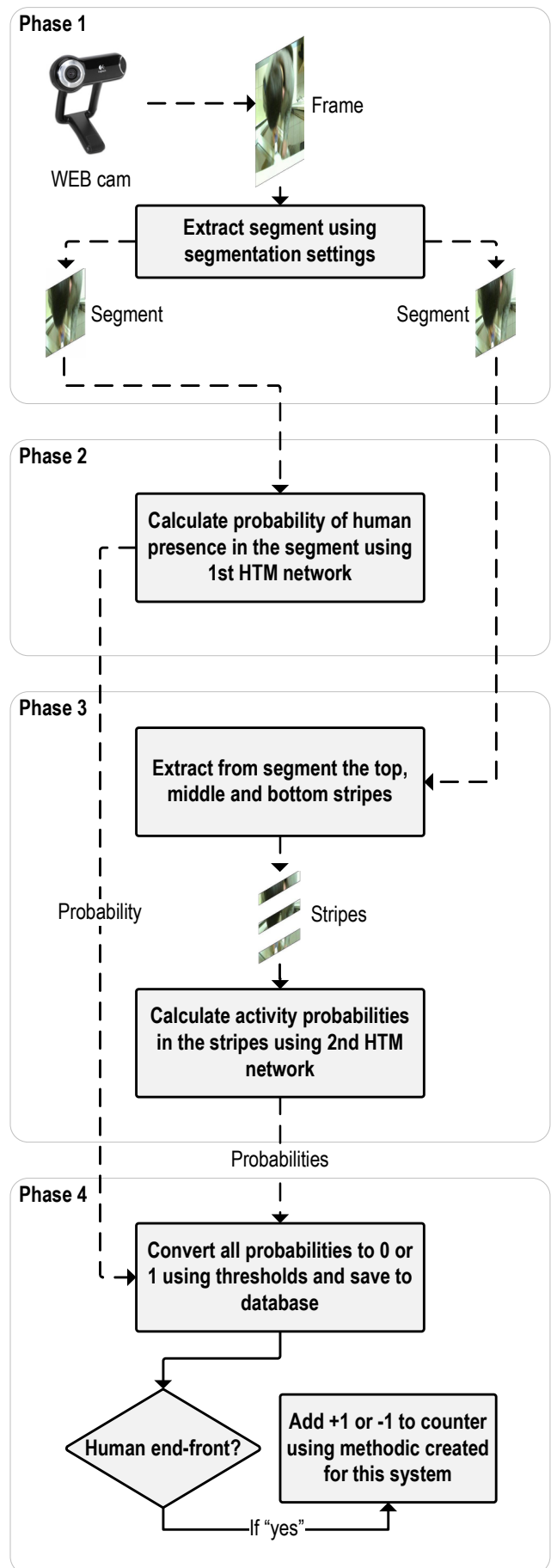


Fig. 2. Schema of the human traffic counting system

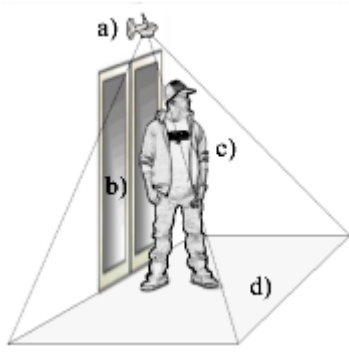


Fig. 3. The observation scheme. Web camera (a) is installed above the doors (b). Visible area is a square (d). Humans (c) enters or leaves a building through the door (b)

Doors differs from each other mostly only by wide. It's hard to say what distribution of wideness is, but definitely doors can be sorted by how many people can pass it in the same moment without collision. In most cases this number can vary from 1 to 3. The system was tested with narrow doors like for 1-2 people, but algorithm should still work with wider passage-ways.

Before starting the system we define area-segment in the whole frame which will be tracked (see Figure 4). In our system this area is in the middle of doors with proportions of 2x1 and 480x230 size in pixels.



Fig. 4. An example of segmentation is shown. Size and location of a segment should be tuned by context specifics. We placed it in the middle of doors

Designing an HTM network

Numenta provided some tools to experiment with HTM algorithms. The core is NuPIC [12] framework based on Python programming language. It allows creating a HTM network in any structure, to train, to debug and to run inference on data sets after training is done. Visio Framework is an extension of NuPIC to design networks for vision problems only in a parameterized way without coding python. Vision Toolkit [16] has GUI, 2 optimized HTM networks for universal vision problems and let's to run experiments in only few clicks. All these tools were used for experimenting and some comparisons of accuracy and performance are shown in results section.

The biggest challenge is to design a HTM network which is fast enough, accurate and universal. Firstly we tried to create networks with Vision toolkit. This toolkit analyses data and choose one of two already prepared

networks. Those networks are optimized, but they are too big for our problem, slower than necessary and can't be used with TBI ability. It's why we decided to create own network.

Networks designed by Numenta are too large, so we designed own network of only 3 levels. Most parameters were tuned empirically, others taking into account desirable outcome. Here are some adjustments explained.

Sensor dimensions are 200x60 pixels in size. We found that this size is small enough to get reasonable good performance and is big enough not to lose important information.

We use grayscale pictures, because HTM can't deal with colors yet and it converts them to grayscale anyway.

First experiments was done using spatial pooler in the first level, but we found what it creates different outputs even if training set doesn't changes and this can lead to unpredictable results, because each time different spatial alphabet is formed. So, we have used NuPIC instrument "GaborNode" witch works like Gabor filter and always has same results with same input.

After network is designed, it should be trained in the proper way. Our data are temporal, so we should train the network in the sequence of movie.

HTM network for humans' detection (Phase 2)

We used 1358 real situations when humans enter or leave the room for training and testing an HTM network. Most situations are really complicated, because persons passes fast with tiny time spans in between or merges visually. The video is few hours long and the light is changing rapidly.

About 7000 pictures from different parts of video were used for training and are divided in two categories (Figure 5).

Categorization of training data was empirical. We reviewed all training frames and classified them into two categories. The frames where big parts of humans are visible were assigned to the first category and other frames to the second category.

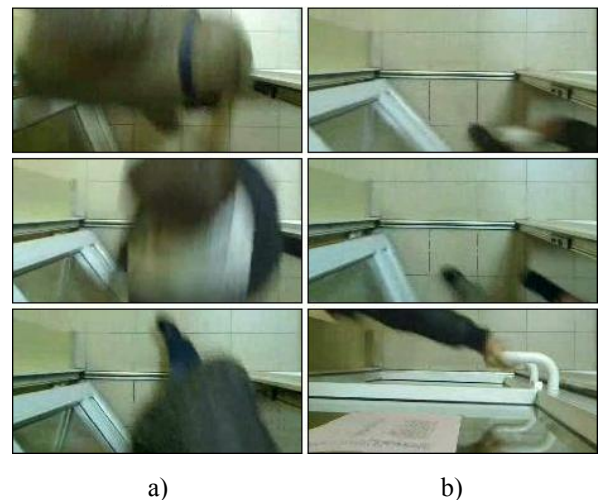


Fig. 5. A few examples from training data sets: a) frames with humans; b) others frames

Human's detection was done simply by analyzing one by one all segments in the testing data set. The HTM network each time provides a probability between zero and one. Probability one is obtained for the cases when a human appears in the segment.

HTM network for movement detection (Phase 3)

When a human in the segment is detected the second task is to determine the movement direction of this human. If only one human pass the doors, it's easy to calculate the movement direction using various methods [6,8]. But this is a complicated task when several humans create one long moving object without enough time spans in between. In our analysis we have assumed that only one human could be in one segment in the particular moment of the time and all humans move from segment's top to bottom or from bottom to top. We put some small "zones" in the segment and then analyzed the activation sequences of these zones. The HTM network for movement detection must detect objects in the "zone". We placed three such activity monitoring "zones" in the top, middle and bottom of the segment. They are implemented by cutting 3 stripes from a segment (see Figure 6) and calculating activity in each strip by HTM network for movement detection.

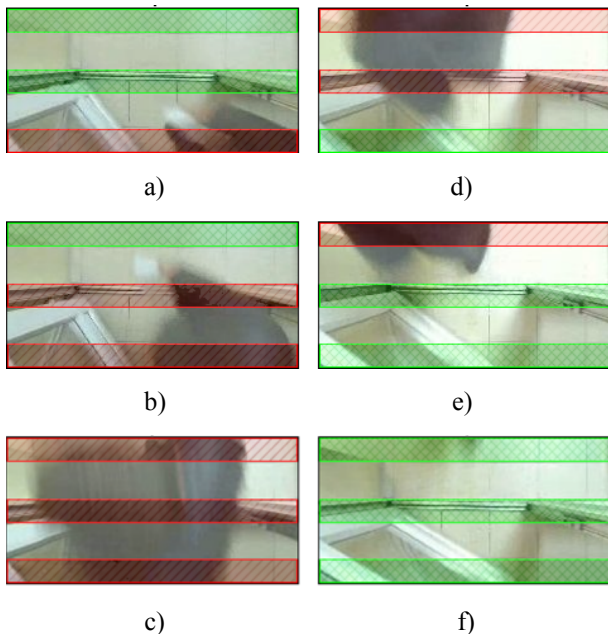


Fig. 6. An example of stripes activity when a human enters room (diagonal cross– idle, backward diagonal– active)

To detect humans' movement the sequence of stripes' activity peaks should be analyzed. When human enters the room, the bottom stripe is most active at first, then the middle stripe is activated and lastly the top stripe is active. If the sequence is otherwise: top, middle, bottom, we can state, that one human exited the room.

In more complex situations without time spans between humans, we could only look at first and last human assuming that all of them go in the same direction.

Parameters of the HTM network for movement detection are the same as for the first HTM (HTM for human detection) except the size of sensor. We use stripes

in 230x20 pixels in size. Learning strategy was also the same. First we train a network with one stripe showing all variations of it, then the second one and then the third one.

Balance calculation (Phase 4)

In the final phase the system calculates how much humans enters and exits the room. Firstly we convert human and movement recognition probabilities to two possible values: 0 or 1. All very high probabilities are converted to 1 and other to 0. Tiny peaks are not used in calculations at all. Balance calculation is realized according the schema presented at the figure 7. The decisions about humans' movement in frames are made by HTM networks, using information about activity of stripes changes when human is detected in a segment.

Results

The proposed schema of the human traffic counting system was tested using 1358 real situations when humans enters and exits a room (see Table 1).

Table 1. Experimental results

Situation	Humans	Accuracy	Direction
One pedestrian	813	83.37 %	76.98 %
Queue	336	77.62 %	73.35 %
Passes each other	209	76.82 %	58.66 %
Total:	1358	80.94 %	73.44 %

All situations are classified into three categories. First category is when one pedestrian walks thru passage-way and it is at least few empty frames until another pedestrian walks. Most missed situations are when pedestrians walk very fast or run. The system had the problem with short pedestrians like kids and sometimes one pedestrian is split into two. There are some possibilities to improve accuracy of this category by better training data and by tuning some parameters of algorithm.

Second category is those situations when two or more pedestrians walk in the same direction. The problem with this category is when they walk too close each other.

The third category is when two or more pedestrians walk in the opposite direction and merges in the segment.

An overall accuracy of the system is 80.94 % in the current context.

We have done few comparisons between various kinds of network models by accuracy and by performance. The accuracy difference between TBI and FI modes is shown in Figure 7. Also we compared our 3 levels network versus Visions toolkit 7 levels network (Figure 8). By performance our network with TBI processes faster 1.6 times than Vision Toolkit network. So, those results show what our network with TBI mode is better tuned for this particular problem.

After using thresholds for recognition results we got fronts as shown in Figures 9 and 10. Humans' quantity now can be calculated by counting end-fronts.

So, using these simple algorithms (making data discrete and analyzing stripes fronts of activity) we can identify if a human enters or exits the room.

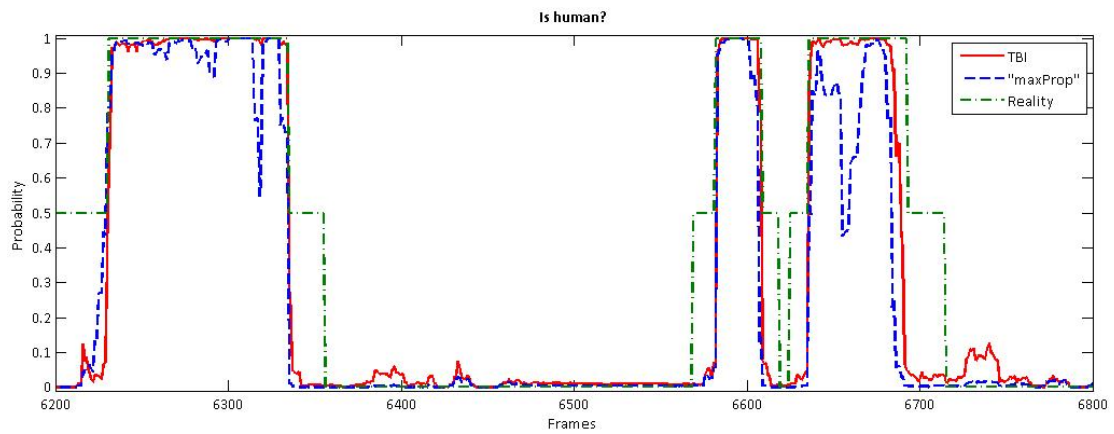


Fig. 7. TBI versus FI ("maxProp"). „Reality“ series visualizes actual values (0.5 – only small parts of humans are visible before entering or leaving, 1 – a human visible in the segment). Certainly TBI makes results more stable and smooth.

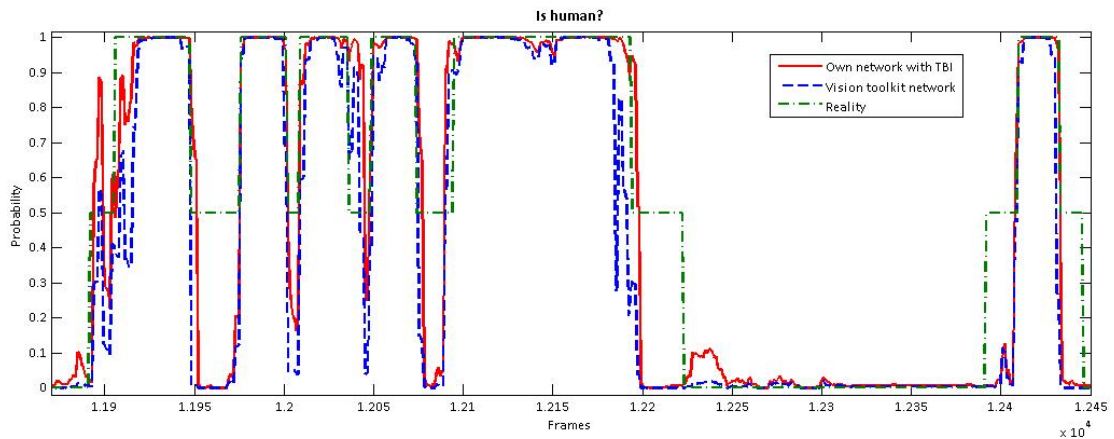


Fig. 8. Our network versus Vision toolkit network. Both networks get similar accuracy results; even with tiny time spans in between.

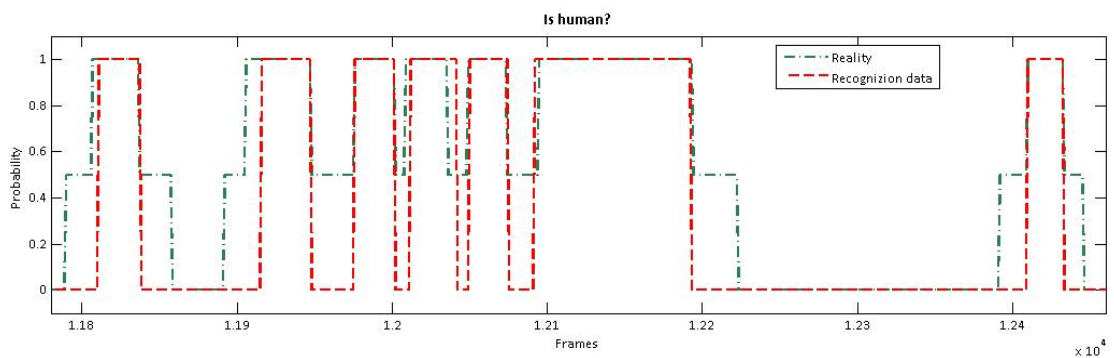


Fig. 9. After humans' recognition data were processed using thresholds (see Figure 8 to compare with original data)

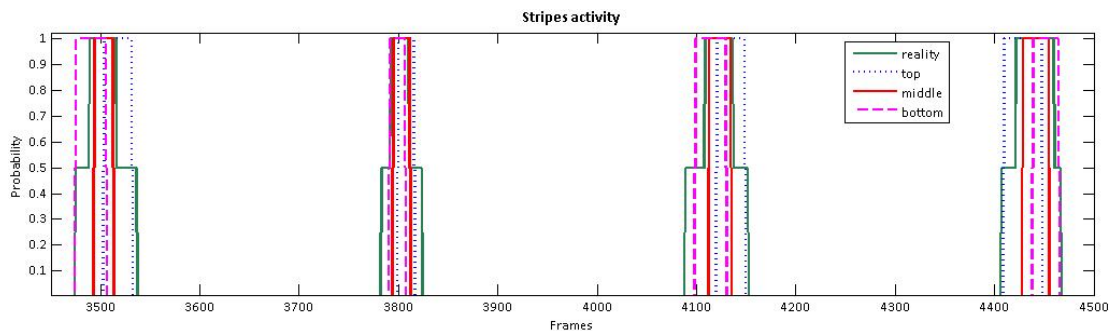


Fig. 10. After stripes activity data were processed using thresholds

Conclusions and future work

In this paper an alternative recognition mechanism was proposed for humans balance calculation in public places using HTM networks. The proposed approach allows distinguishing humans from other objects like background or shadows in a current video frame, to identify direction of movement, and to calculate how many humans entered or leaved any passageway. Proposed algorithms can be applied to various kinds of traffic balance calculations.

The trained network with three levels is faster and more stable than Vision toolkit optimized networks. With TBI inference we get better accuracy results but these methods are about 10 times slower, so if there is no necessity to use TBI then FI could be much faster.

The system classified correctly 80.94 % situations of 1358 and accuracy of direction recognition was 73.44 %. As expected, the algorithm works more accurate if just one pedestrian walks.

In the future we should try to use some normalization techniques for input frames. Also we are planning to use infrared camera, because lighting conditions are essential and we are looking for ways to reduce it.

References

1. **Hawkins J., George D.** Hierarchical Temporal Memory. Concepts, Theory, and Terminology. http://numenta.com/Numenta_HTM_Concepts.pdf, 2007.
2. **Chen D., Su C., Zeng Y., Sun S., Lai W., Liao H.** An Online People Counting System for Electronic Advertising Machines // IEEE International Conference on Multimedia and Expo. - New York, 2009. - P. 1262–1265.
3. **Zhu F., Gu J., Yang R., Zhao Z.** Research on Counting Method of Bus passenger Flow Based on Kinematics of Human Body and SVM // International Conference on Computer Science and Software Engineering. - Wuhan, Hubei, 2008. - P. 201-204.
4. **Xin Y., Sun G., Wu Q.** A Preprocessing Method for Tracking and Counting Pedestrians in Bus Video Monitor // International Conference on Industrial Informatics. - Daejeon, 2008. - P. 1689-1693.
5. **Ryan D., Denman S., Fookes C., Sridharan S.** Scene Invariant Crowd Counting for Real-Time Surveillance // 2nd International Conference on Signal Processing and Communication Systems. - Gold Coast, 2008. - P. 1-7.
6. **Ming Y., Wei L.** A New Method for Passenger Flow Counting System Based on Surveillance Video // Second International Conference on Intelligent Networks and Intelligent Systems. - Tianjin, 2009. - P. 453-456.
7. **Jaijing K., Kaewtrakulpong P., Siddhichai S.** Object Detection and Modeling Algorithm for Automatic Visual People Counting System // 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. - Pattaya, Chonburi, 2009. - P. 1062-1065.
8. **Zhao C., Pan Q., Li S.** Real Time People Tracking and Counting in Visual Surveillance // The Sixth World Congress on Intelligent Control and Automation. - Dalian, 2006. - P. 9722–9724.
9. **Teixeira T., Savvides A.** Lightweight People Counting and Localizing in Indoor Space Using Camera Sensor Nodes // First ACM/IEEE International Conference on Distributed Smart Cameras. - Vienna, 2007. - P. 36-43.
10. **Hawkins J., Blakeslee S.** On Intelligence // Times Books. - New York, 2005. – 174 p.
11. **George D. Hawkins J.** A Hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex // In Proceedings of the International Joint Conference on Neural Networks. - IEEE, 2005. – VOL. 3. - P. 1812-1817.
12. NuPIC 1.7.1. // <http://numenta.com/for-developers/software.php>.
13. **George D., Jaros B.** The HTM Learning Algorithms // http://numenta.com/for-developers/education/Numenta_HTM_Learning_Algos.pdf, 2007.
14. **George D.** Ph.D Thesis. How the Brain Might Work: a Hierarchical and Temporal Model for Learning and Recognition // - June 2008. - P. 161-163.
15. Time Based Inference, <http://www.numenta.com/for-developers/education/tbi-overview.php>.
16. Vision Toolkit 1.7.1, <http://numenta.com/vision/vision-toolkit.php>.

Date of paper submission

S. Sinkevičius, R. Simutis, V. Raudonis. Monitoring of Humans Traffic Using Hierarchical Temporal Memory Algorithms // Electronics and Electrical Engineering. – Kaunas: Technologija, 20XX. – No. X(XXX). – P. XX–XX.

The main purpose of this paper is to investigate the application of Hierarchical Temporal Memory (HTM) mechanism for humans traffic calculation in a public places using web cameras. The proposed approach distinguishes humans from other objects in a current video frame, identifies direction of movement and calculates balance of the humans' traffic. As a result, all people entering and leaving room can be counted and information about people-traffic can be acquired. This information is very useful for further traffic monitoring and can be used for various traffic organization tasks. III. 10, bibl. 16 (in English; abstracts in English and Lithuanian).

S. Sinkevičius, R. Simutis, V. Raudonis. Hierarchinės laikinosios atminties algoritmų taikymas žmonių srauto monitoringo sistemai // Elektronika ir elektrotechnika. – Kaunas: Technologija, 20XX. – Nr. X(XXX). – P. XX–XX.

Pagrindinis šio darbo tikslas yra pritaikyti hierarchinę laikinąją atmintį (Hierarchical Temporal Memory) žmonių srauto balansui skaičiuoti viešosiose vietose su WEB kamera. Pasiūlyta metodika leidžia atskirti žmones nuo kitų objektų, sekti judėjimo kryptį ir paskaičiuoti srauto balansą. Ši informacija gali būti panaudota optimizuojant sistemas, kuriose žmonių srautai, eilės bei apkrautumas yra kritiniai. II. 10, bibl. 16 (anglų kalba; santraukos anglų ir lietuvių k.).

4 priedas. Straipsnis leidinyje „The 6th International Conference on Electrical and Control Technologies“

OPTICAL FLOW VERSUS HIERARCHICAL TEMPORAL MEMORY ALGORITHMS FOR HUMAN TRAFFIC MONITORING SYSTEM

Paulius LENGVENIS*, Saulius SINKEVIČIUS**, Rimvydas SIMUTIS*
*Department of Process Control, Kaunas University of Technology, Lithuania
**Faculty of Informatics, Kaunas University of Technology, Lithuania

Abstract: pedestrian flow data of public transport, museums, galleries, shops and other buildings can help to maximize return on expenditure, to increase efficiency of business or to monitor unusual behavior for security reasons. Managing agents, owners, retailers, even security executives are always welcome for data that lets to predict customers or visitors behavior and to tune business afterwards.

This paper investigates two methods for pedestrian counting in both directions using web camera: Optical Flow and Hierarchical Temporal Memory (HTM). These methods were applied to same data and the quick overview of implementation and results is presented.

Keywords: object recognition, movement tracking and classification, Optical Flow method, Hierarchical Temporal Memory method, computer vision.

1. Introduction

This paper presents two methods, Optical Flow [1] and Hierarchical Temporal Memory (HTM) [2], for counting pedestrian flow in both directions in a public places, like public transport, museums, shops, etc. This information can be used for removing pedestrian traffic bottlenecks, reducing loads, distributing traffic and adjusting business processes according to pedestrian activity.

The paper contains six main sections. In the second section experimental setup is overviewed; it's same for both methods. Third section gives more information about Optical Flow method and fourth section covers HTM methods extension for people counting. Comparison of both methods is represented in section 5. And conclusions including future work are in section 6.

2. Experimental setup

There were used experimental equipment: notebook type computer, web camera (Logitech® QuickCam® Pro 9000) to collect video material for this investigation.

Place of camera installation is chosen above the passage, the image is formed perpendicularly to the floor (fig. 1) and this allows us to determine when pedestrian enters or leaves a building accurately.

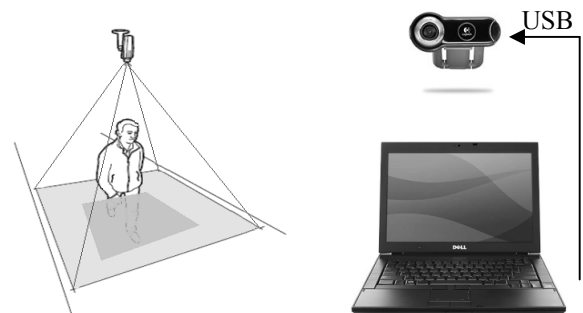


Fig. 1. a) Camera mounting schema b) Equipments

Recording resolution was set to 640x480 at 25 fps. Total over 2 hour video material was collected. During this time 76 pedestrians passed. There were situations when 2 peoples entered or passed each other in same time. A few situations presented in fig. 2.



Fig. 2. A few examples of observation data

3. Optical flow

Optical flow [1-2] is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. Optical flow techniques [3-4] such as motion detection, object segmentation, time-to-collision and focus of expansion calculations, motion compensated encoding, and stereo disparity measurement utilize this motion of the objects surfaces, and edges.

The optical flow methods calculate the motion between two image frames which are taken at times t and $t + \delta t$ at every volumetric pixel (voxel) position. This feature can be used in video or live-video material analysis for motion estimation. An example is shown in fig. 3.



Fig. 3. An example of optical flow direction (by lines)

3.1. Pedestrian counting algorithm

Primary processing was made using optical flow. In an image given below (fig. 4) four areas are presented (PX1, PX2, PX3 and PX4), where a difference initiated by the persons going in/out the building is studied.

$$Intensity1 = \sum change(PX1) \quad (1)$$

$$If Intensity1 > threshold1, then object1 = 1 \quad (2)$$

$$Else object1 = 0$$

where, $PX1$ are zone near enter to building (fig. 4), $PX2$, $PX3$ and $PX4$ have same check. As the experiment results showed it is rational to choose *threshold* value equal to 50% of pedestrian total change witch is calculated when he enters or leaves.

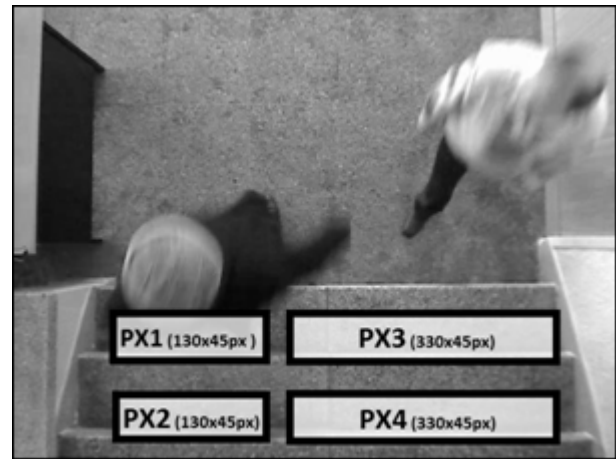


Fig. 4. An example of selected zones is shown

The direction of a passenger was registered by IF...THEN logic: if PX1 area is crossed first and then PX2 one – the pedestrian gets in otherwise gets out. Same logic used with PX3 and PX4 zones.

Other places of the picture are not analyzed and this enables a quick-acting of the proposed algorithm.

3.2. Experiments and results

Part of final results is shown in fig. 5 (video frames 2000-6000). Algorithm accuracy results are listed in table 1.

Table 1. Results summary

Total humans in video material	76
Detected	72
Accuracy of detection	94.70%
Correct direction from detected humans	71
Accuracy of direction	98.60%

Algorithm accurately detects a single pedestrian getting in/out the building, however it cannot detect 3 pedestrians when they are passing in same time (we had one situation), also algorithm missed one pedestrian going by corner and two getting out very close by other pedestrians.

We get 94.7 % accurate of algorithm in current context for human detection and 98.6% accuracy of direction detection.

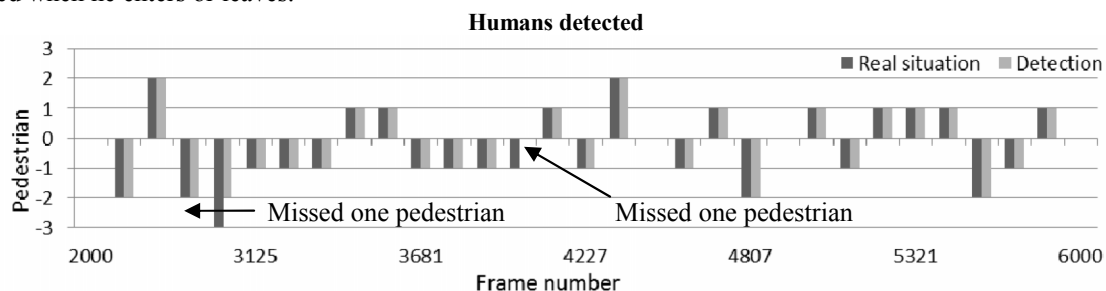


Fig. 5. Some results

4. Hierarchical Temporal Memory

HTM is a memory system that implements the structural and algorithmic properties of the neocortex [5-9]. Instead of programming individual solutions for each problem, this method uses common algorithms to solve many cognitive tasks.

Numenta provided some tools to experiment with HTM algorithms. The core is NuPIC [10] framework based on Python programming language. It allows creating a HTM network in any structure, to train, to debug and to run inference on data sets after training is done. Visio Framework [11] is an extension of NuPIC to design networks for vision problems only in a parameterized way without coding python. We used both tools to get the results.

4.1. Pedestrian counting algorithm

The main idea of this method is to decide how much pedestrians can pass each other naturally without jostle and to dedicate same amount of segments along passageway. In the current context there are two segments (fig. 6), but in most contexts one is enough, because in those cases pedestrians tend to wait while another one is walking through.

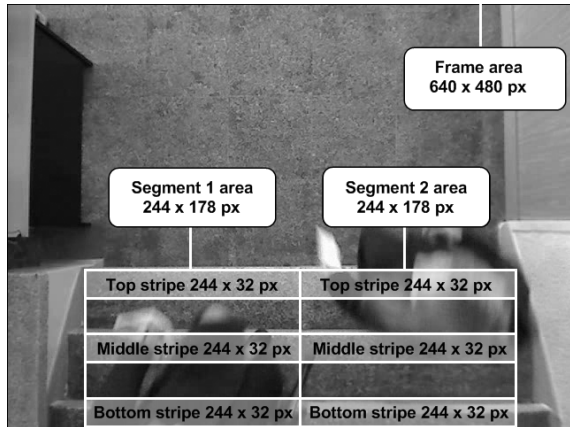


Fig. 6. An example of segmentation is shown

In each segment we track for probability of pedestrian being and stripes activity. A high probability of pedestrian being in the segment tells us, what probably where is a pedestrian inside the segment and activity sequence of stripes lets identify the direction of movement. So, if we know, that where a pedestrian inside the segment is and we know which stripe was activated first, with one the last one, we can add to pedestrian counter +1 or -1 accordingly.

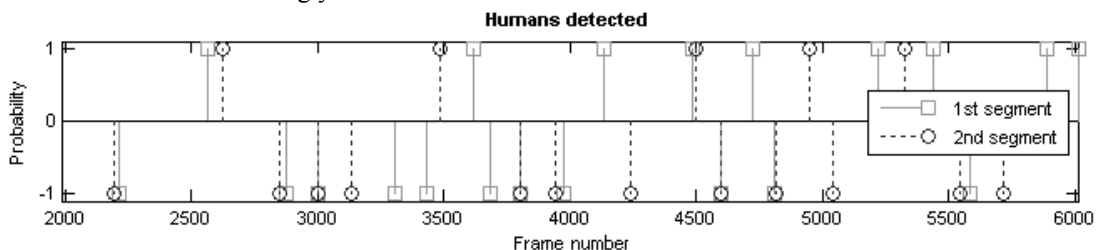


Fig. 7. Some results for both segments. 1 – one human entered, -1 – one human exited

We prepared two HTM networks, they differs only by size of sensor. First one is trained to recognize pedestrian in the segment and the second one is trained to recognize an activity of stripe. For each network we used sets of 3000 classified images and it's only one third of all not empty data.

So, for each frame we calculate 8 floating point probabilities (2 segments, 6 stripes) and we know how probabilities changes in time after each frame. Using few rules, we convert those probabilities into 0 or 1 integer values. Now pedestrians can be counted, because we know exact time intervals when a pedestrian was seen and which direction he was walking.

4.2. Experiments and results

Part of results for HTM algorithm is shown in fig. 7 and all results divided in 4 groups are shown in table 2

Table 2. Experimental results

Group number, name	Humans	Results
1, accurate	58	76.32%
2, walks in the middle	10	13.16%
3, sees double	6	7.89%
4, errors	2	2.63%
Total:	76	100.00%

First group is the most confident. If the pedestrian really walks through the left segment, we detect him in the left segment. The same is with the right segment. Second group are cases when a pedestrian walks through the middle of passageway, but algorithms counts him only in the one segment (right or left) and it's correct. The third group is a bit tricky, because in those cases a pedestrian walks in the middle as in second group, but is detected in both segments at the same time. Those cases can be classified correctly too by applying following rule. If two segments that are very similar by activity (ends and starts almost at the same moment, and has the same direction) then they can be merged. Instead of counting two pedestrians with same direction, we add only one to the counter. The final group is errors. In one of those situations, 3 pedestrians walk at the same time very close to each other and we miss one of them. The second error, when one short pedestrian walks in the middle of frames. Both segments detected, what there is a pedestrian, but couldn't detect a direction of movement. If we sum all three groups as correctly detected pedestrian we have 97.37% accurate of algorithm in current context.

5. Optical flow versus HTM

Methods are implemented in different programming languages and this should be considered. Optical flow algorithms are done in MatLab and HTM in Python programming language.

Both methods need some sort of calibration. Optical flow is easier to setup and works with same accuracy even if background varies. After configuration is optimized this method can be applied to different applications without further requirement of additional effort. HTM algorithms require additional work for each installation, because networks should be retrained using at least few hundred classified images from the new background and this can be done only manually. However, both algorithms use some thresholds to optimize accuracy and these parameters should be optimized anyway for desired background or for set of similar backgrounds.

Optical flow processes one frame per 0.32 second in comparison to 0.76 second of HTM methods. So, it's faster and can perform even faster if only small movements in images are detected. Unfortunately time of one recognition cycle by HTM is always the same and do not depends on activity in images. This implies that Optical flow is 3 or more times faster.

Luckily both methods can skip frames without processing them if only background is visible and no movement detected. In not crowded places sum of such empty frames can exceed more than 50-90 percent of total frames. Its huge idle time spans that can be used to process already acquired frames from camera or to do any useful preprocessing work.

The most important difference between both methods is that HTM methods actually learn to recognize pedestrians and differentiate them from other sharp objects. It is much more accurate and stable when any type of noise is presented comparing to Optical flow and actually we can train HTM networks to recognize noise as not pedestrians in any specific context. For example, if HTM will encounter a shadow it will likely to recognize it as a not pedestrian which is much more complicated with Optical flow method. This HTM ability to recognize objects and classify to categories, even if those are more than 2, makes it more flexible in many contexts. If only some kind of moving objects should be counted, HTM can handle it.

Trying to compare methods by accuracy we can see, that they both are inaccurate when few pedestrians merges visually. HTM accuracy is 97.37% comparing to Optical flow 94.70%.

Sometimes directions aren't important. For example, most public buildings have only one entrance. If we detect and count all pedestrians without direction then in the end of the day we can divide that number by 2 and get almost accurate visitors' number. This is useful because both methods are much more accurate in objects detection than recognizing direction.

6. Conclusions and future work

Two methods, Optical flow and HTM, were implemented and tested for counting pedestrians flow. The same data set was used for both of them.

We intentionally have chosen wider entrance to test methods accuracy when two or more pedestrians can freely walk close each other in same or opposite direction at the same moment. Methods were fairly accurate and they can be applied to real situations.

Optical flow is more suitable when simpler and fast approach is required without extended involvement of additional people work. HTM is more preferable if classification of moving objects is a key.

Both methods can be applied to various contexts but they have own pros and cons that should be taken into account.

In near future we will try to make these methods able to perform in real time and to install an application for pedestrians counting in real business environment.

7. Acknowledgment

The authors appreciate the financial support of the Lithuanian state science and studies foundation.

8. References

1. S. S. Beauchemin, J. L. Barron. The Computation of Optical Flow. New York, 1995.
2. A. Burton, John Radford. Thinking in Perspective: Critical Essays in the Study of Thought Processes. Rutledge, 1990.
3. David H. Warren, Edward R. Strelow. Electronic Spatial Sensing for the Blind: Contributions from Perception. Springer, 1985.
4. Kelson R. T. Aires, Andre M. Santana, Adelardo A. D. Medeiros. Optical Flow Using Color Information. New York, 2008.
5. Jeff Hawkins, Dileep George. Hierarchical Temporal Memory. Concepts, Theory, and Terminology. http://numenta.com/Numenta_HTM_Concepts.pdf, 2007.
6. Jeff Hawkins, Sandra Blakeslee. On Intelligence. New York, 2005.
7. Dileep George, Bobby Jaros. The HTM Learning Algorithms. http://numenta.com/for-developers/education/Numenta_HTM_Learning_Algos.pdf, 2007.
8. Dileep George. Ph.D Thesis. How the Brain Might Work: a Hierarchical and Temporal Model for Learning and Recognition. 2008. p. 161-163.
9. Dileep George, Jeff Hawkins. A Hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex. Proceedings of the International Joint Conference on Neural Networks. 2005. p. 1812-1817.
10. NuPIC 1.7.1., <http://numenta.com/for-developers/software.php>.
11. Vision Toolkit 1.7.1, <http://numenta.com/vision/vision-toolkit.php>.

5 priedas. Programinės įrangos realizacija

Sistemą sudaro penkios aplikacijos. Pirmoji (žr. 58 pav.) skaito duomenis iš kameros. Antrosios pagalba sistema apmokoma atpažinti žmogų (žr. 59-62 pav.). Trečioji realizuoja realaus laiko skaičiavimus, ketvirtoji atvaizduoja parodos lankytojų rezultatus, o penktoji skirta rezultatų tyrimams (žr. 63 pav.).

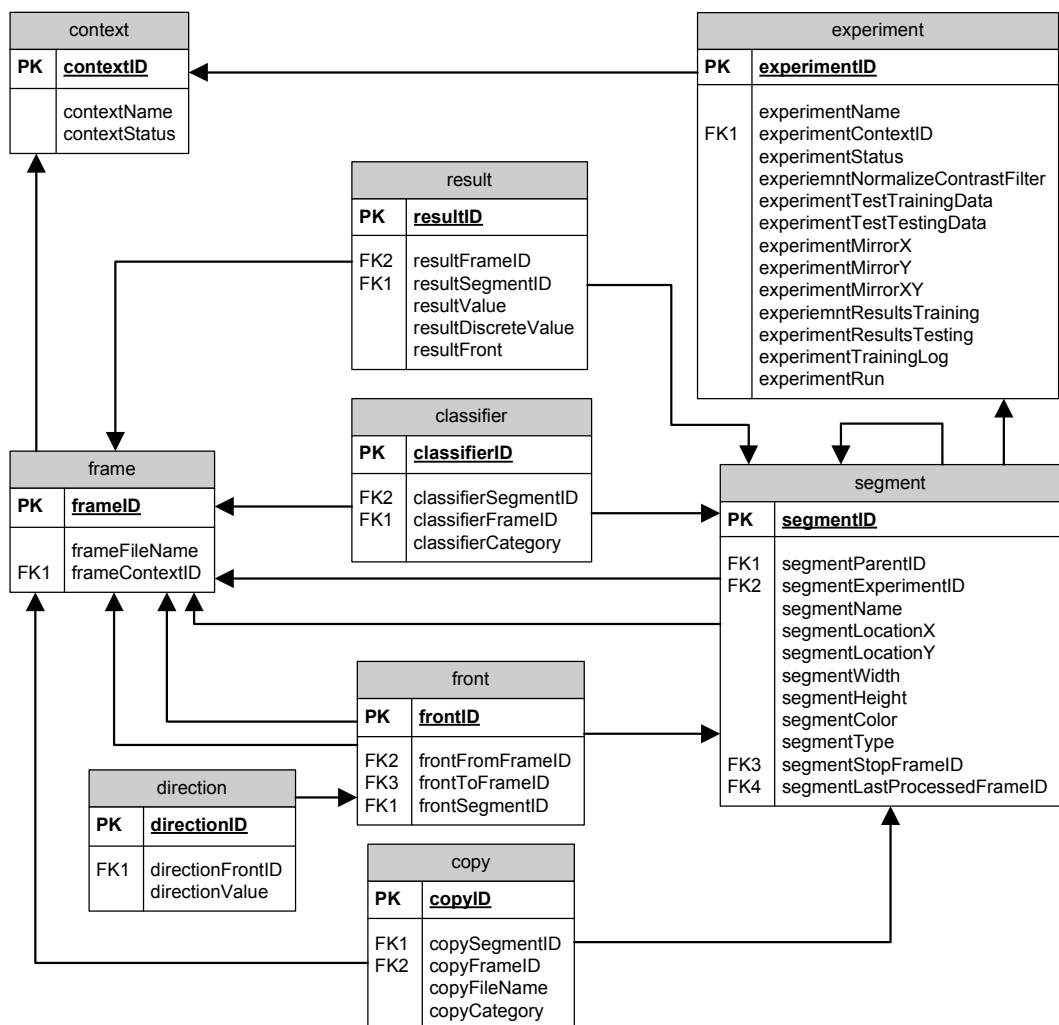
Pagrindinės sistemos funkcijos

Pradinės funkcijos, kurios buvo įdiegtos, kad sistemą būtų galima naudoti tyrimams:

- realaus laiko vaizdų iš kameros arba iš video failo nuskaitymas ir rašymas į DB;
- kontekstų ir eksperimentų kūrimas, redagavimas, šalinimas;
- grafinis interfeisas segmentams išdėstyti, keisti jų parametrus, išsaugoti į DB;
- galimybė grafiškai klasifikuoti kiekvieną segmentą tam tikrai duomenų sekai;
- segmentų iškarpymas ir vaizdų pagal klasifikavimo duomenis ir išsaugojimas;
- automatizuotas abiejų HTM tinklų apmokymas nurodant mokymo parametrus;
- galimybė grafiškai peržiūrėti rezultatus, matyti kaip kinta segmentų atpažinimo reikšmės laike, atvaizduoti visus rezultatus grafikais;
- galimybė bet kurį tyrimų etapą pakartoti iš naujo su naujais parametrais pašalinant prieš tai buvusius tarpinius rezultatus;
- realaus laiko pėsčiųjų skaičiavimo algoritmas, kuris visus duomenis ima iš DB, o visus tarpinius ir galutinius rezultatus rašo į DB;
- grafinis langas atvaizduoti „Technorama 2011“ skaičiavimo rezultatus realiu laiku, atvaizduoti paskutinį aptiktą pėsčiąjį, kiek parodoje šiuo metu ir kiek lankytojų apsilankė iš viso.

Duomenų bazės schema

Visi duomenys iš kameros, tarpiniai skaičiavimo rezultatai, sistemos apmokymo parametrai, galutiniai rezultatai yra saugomi duomenų bazėje. Tai leidžia patogiau atlikti tyrimus, nes prieinami bet kokiame etape gauti skaičiavimai.



57 pav. Sistemos duomenų bazės schema

Lentelės:

- context – kontekstas, kuriame daromi tyrimai, pvz. Technorama, degalinė ir t.t.;
- frame – kiekvienas kontekstas turi skirtingą seką vaizdų, failai saugomi atskirai;
- experiment – kiekvienas kontekstas susideda iš eksperimentų, o pastarieji turi individualius parametrus, tačiau dirbama su to paties konteksto paveikslukais;
- segment – tai HTM tinklų sensoriai matantys tik tam tikras vaizdų vietas, o ne visą vaizdą;
- classifier – žmogaus paruošti-suklasifikuoti duomenys HTM tinklų apmokymui;
- copy – prieš apmokant paveikslukai pagal segmentų ir eksperimento parametrus yra iškerpami ir saugomi atskirai;
- result – kiekvieno segmento kiekvienam vaizdai atpažinimo rezultatas;
- front – rezultatų frontai surašomi į atskirą lentelę;
- direction – kiekvienam frontui yra nustatoma jo kryptis (įeina-išeina).

HTM tinklų struktūra

Sistemai suprojektuotas ir išbandytas HTM tinklas.

```
trainingData      = 'PLACE_HOLDER'
testingData       = 'PLACE_HOLDER'
numCategories      = 2
imageDimW         = 'PLACE_HOLDER'
imageDimH         = 'PLACE_HOLDER'
gaborNumOrientations = 4
gaborPhaseMode    = 'single'
gaborCenterSurroud = False
dataLength        = N # total images

from nupic.pynodes.GaborNode import GaborNode
numGroupsL2       =
GaborNode.calcNumPlanes(numOrientations=gaborNumOrientations, phaseMode=gaborPhaseMode,
centerSurroud=gaborCenterSurroud)

network = [
  { # Sensor
    'width': imageDimW,
    'height': imageDimH,
    'scaleCount': 1,
    'scaleDecimation': 1,
    'filters': [['Resize', {'size': [imageDimW, imageDimH]}], ['NormalizeContrast']]
  },
  { # Level 1
    'nodeType': 'py.GaborNode',
    'filterDim': 11,
    'numOrientations': gaborNumOrientations,
    'gainConstant': 0.5,
  },
  { # Level 2
    'nodeType': 'TemporalPoolerNode',
    'clonedNodes': True,
    'requestedGroupCount': numGroupsL2,
    'equalizeGroupSize': True,
    'transitionMemory': 24,
    'spatialRF': (0.1, 0.1),
    'spatialOverlap': (0.5, 0.5),
    'scaleRF': 1,
    'scaleOverlap': 0,
    'temporalPoolerAlgorithm': 'maxProp',
  },
  { # Classifier / 3
    'nodeType': 'py.SVMClassifierNode',
    'outputElementCount': numCategories,
    'minC': 1.6,
    'minGamma': -3.4,
    'doSphering': True
  }
]

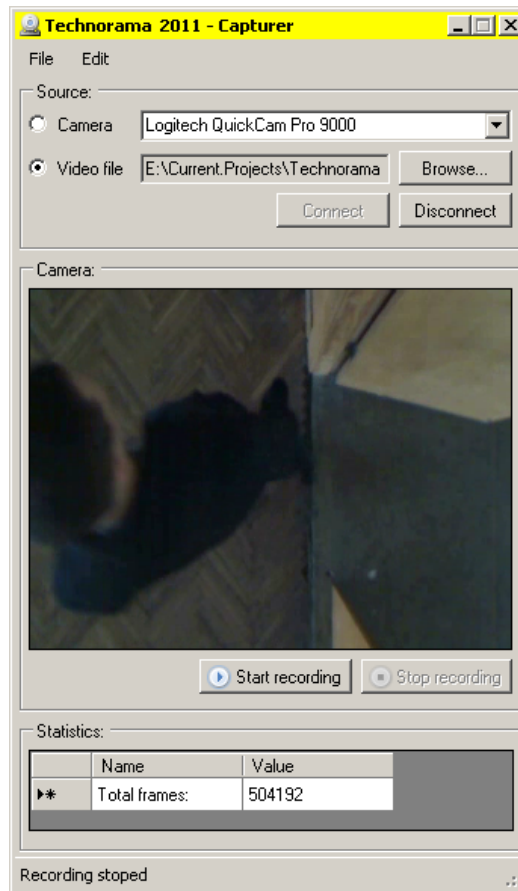
training = [
  { # Level 1
    'data': ['loadMultipleImages', {'imagePath': trainingData}],
    'explorer': 'Flash',
    'numIterations': 0
  },
  { # Level 2
    'numIterations': dataLength,
    'explorer': 'Flash',
  },
  { # Classifier / 3
    'explorer': "Flash"
  }
]

testing = [
  {
    'data': ['loadMultipleImages', {'imagePath': testingData }]}
]
```

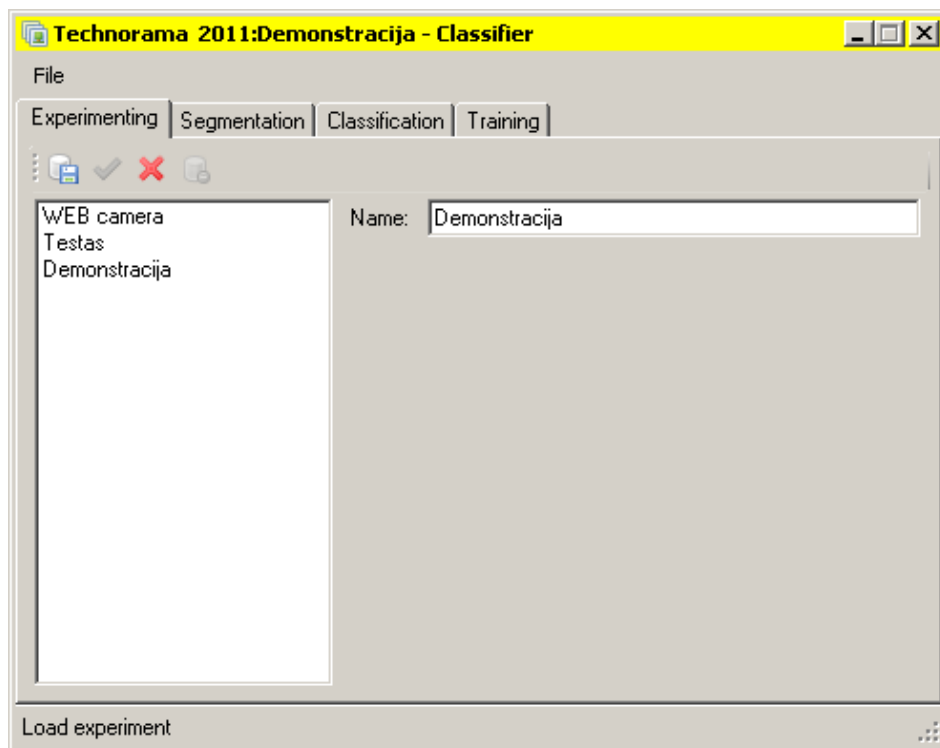
58 pav. „params.py“ failas tinklo sukūrimui ir apmokymui „Vision toolkit“ sistemoje

Grafinis vartotojo interfeisas

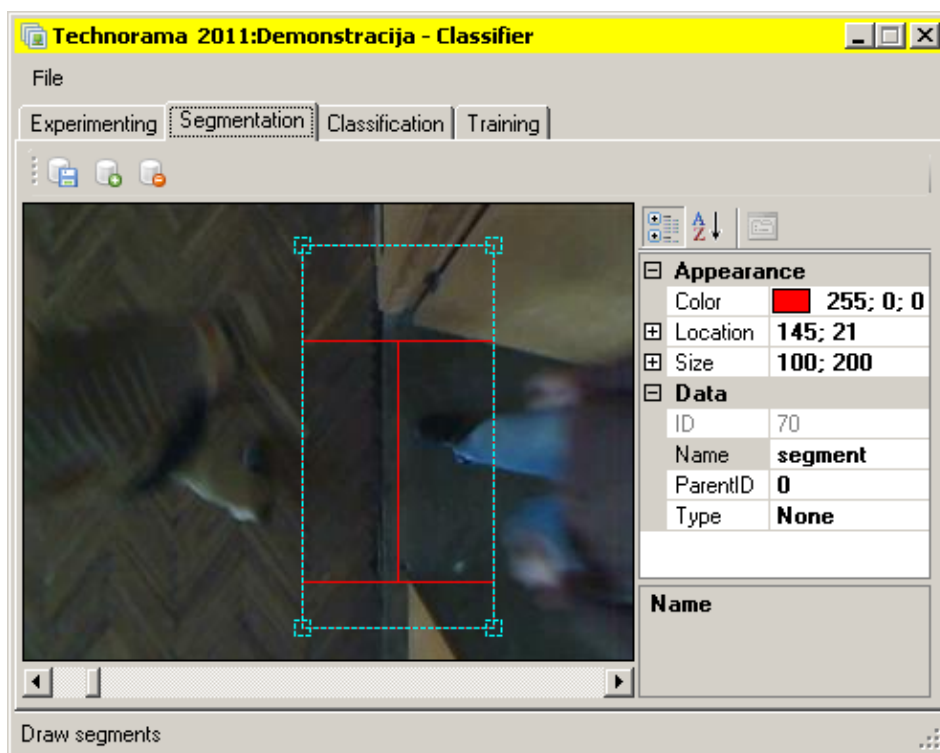
Čia pateikiami keletas grafinio interfeiso langų.



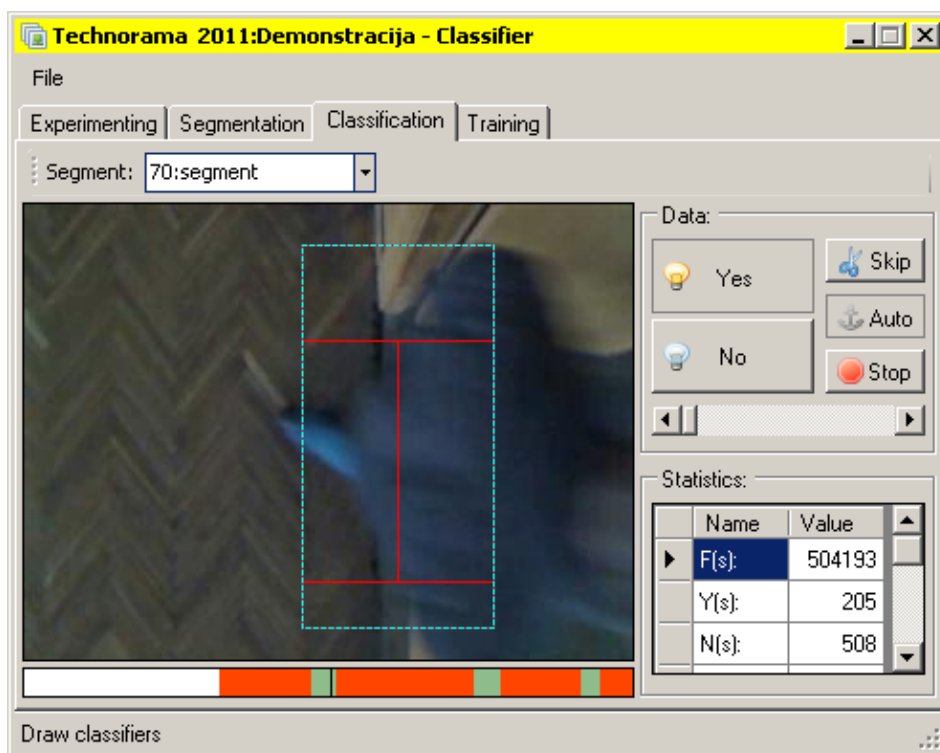
59 pav. WEB kameros vaizdai automatiškai rašomi į DB



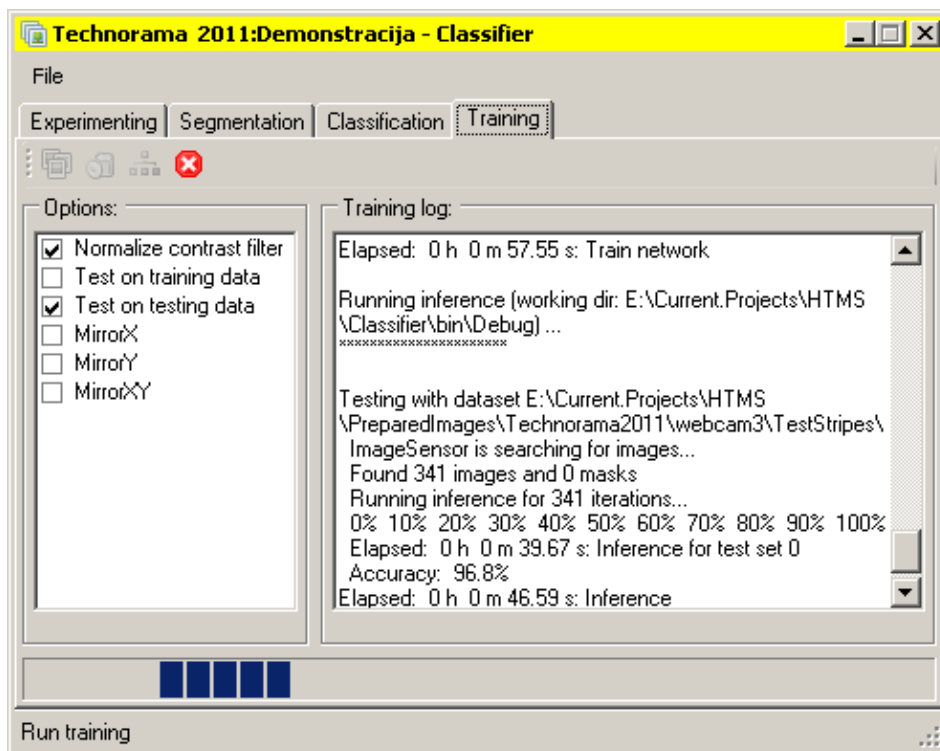
60 pav. Galimybė atlikti skirtingus eksperimentus su tais pačiais duomenimis



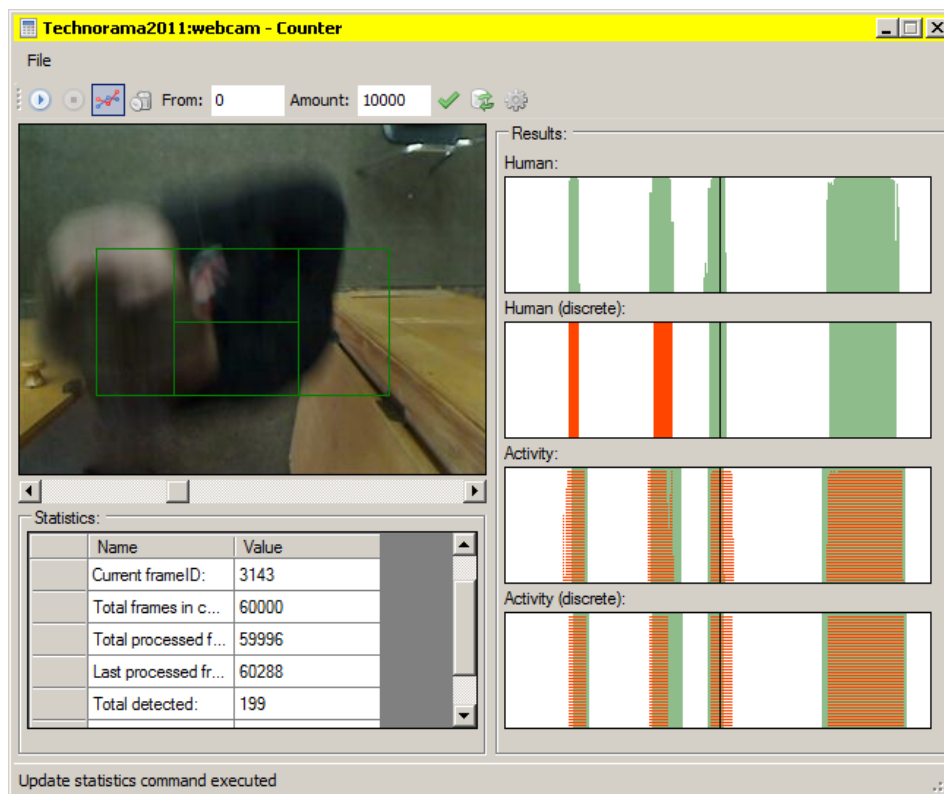
61 pav. Patogus būdas grafiškai tvarkyti sensorius



62 pav. Grafinis interfeisas leidžia greitai paruošti tūkstančius paveikslėlių apmokymui

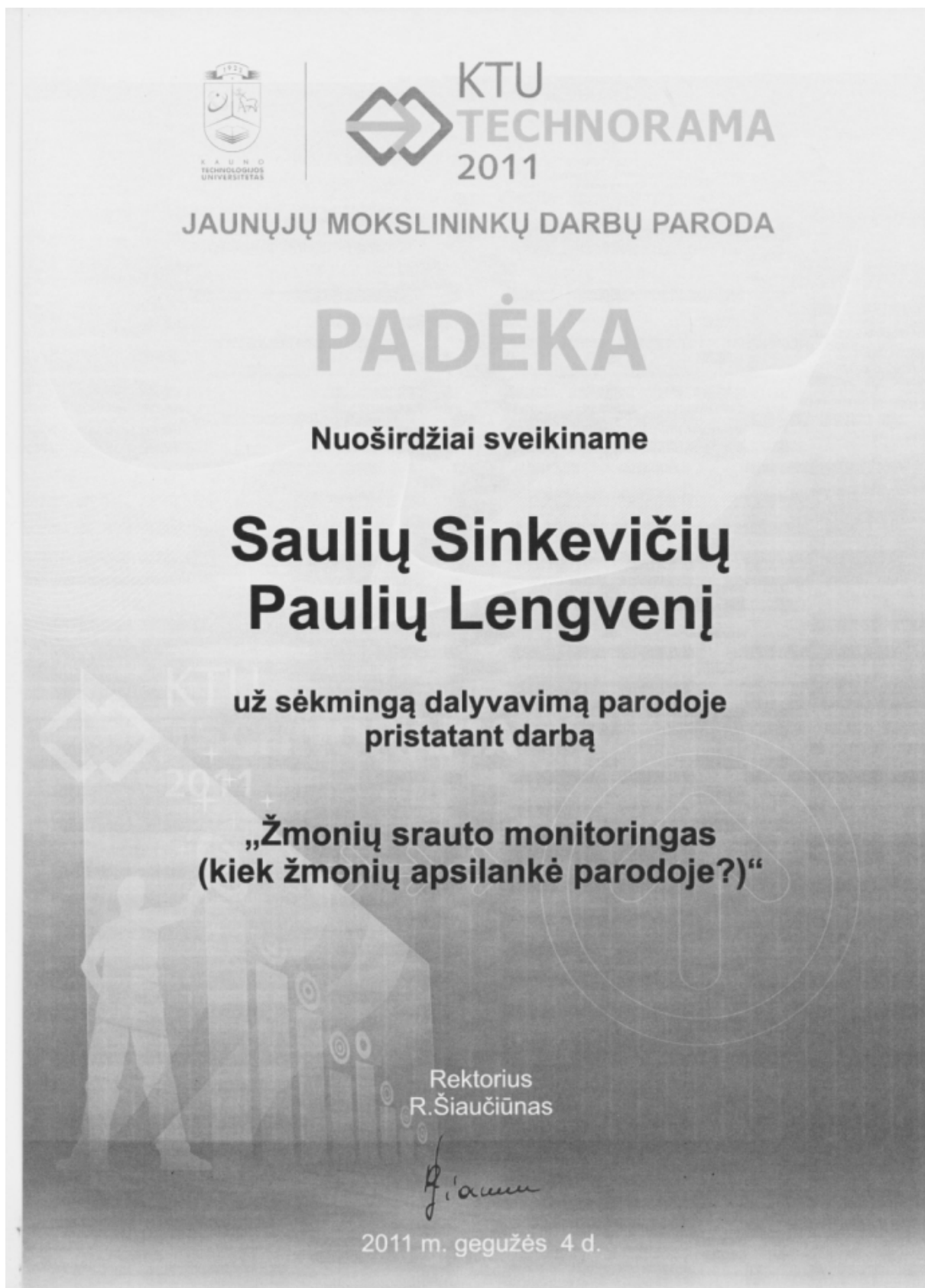


63 pav. Apmokymo paveikslukų iškarpymas iš kameros vaizdų ir sistemos apmokymas



64 pav. Rezultatų peržiūros ir tyrimų langas

6 priedas. Padėka už praktinį darbo taikymą „Technorama 2011“ parodai



65 pav. Padėka už praktinį darbo taikymą „Technorama 2011“ parodai