

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
INFORMATIKOS KATEDRA

**Kliūčių atpažinimas kelyje naudojant dirbtinius  
neuroninius tinklus**

**Obstacle recognition in the way using artificial neural networks**

Magistro baigiamasis darbas

Atliko: Nerijus Gaidjurgis (parašas)

Darbo vadovas: Mindaugas Eglinskas (parašas)

Recenzentas: Dr. Valdas Dičiūnas (parašas)

Vilnius – 2009

## **Santrauka**

Šiame darbe yra nagrinėjama kliūčių atpažinimo vaizde problema. Ši problema susideda iš skirtumų vaizdo formavimo, vaizdo paruošimo dirbtiniam neuroniniam tinklui ir objektų klasifikavimo, naudojant dirbtinį neuroninį tinklą, uždavinių. Darbe siekiama išnagrinėti esamus vaizdo formavimo, apdorojimo ir dirbtinio neuroninio tinklo klasifikavimo būdus ir pateikti uždavinių sprendimo variantą kaip tai galima padaryti geriau.

Remiantis autoriaus siūlomais sprendimais yra sukurta programinė įranga, kuri sudaryta iš trijų modulių: skirtumų žemėlapių vaizdo formavimo, sukurtojo skirtumų žemėlapių vaizdo pirminio apdorojimo ir DNT kliūčių identifikavimo apdorotame skirtumų žemėlapių vaizde.

Raktiniai žodžiai: skirtumų žemėlapis, dirbtinis neuroninis tinklas, kliūtis, atpažinimas

## Summary

The problem of obstacle recognition on way is analyzed by author in this work. This problem consists of view formation, view preparation for Artificial Neural Network and object classification using neural networks tasks. It is striving to analyze the formation of view, processing of view and ways of ANN classification, and suggest the better way of task solutions in this thesis.

It is compiled software using authors suggested solutions which consists of three modules: disparity map formation, filtering preparation of created one and obstacle recognition using ANN.

Keywords: disparity map, artificial neural network, obstacle, recognition

# Turinys

Santrauka.....	2
Summary .....	3
Turinys .....	4
Žodynėlis.....	6
1. Įvadas .....	7
1.1 Darbo problematika.....	7
1.2 Tikslai ir uždaviniai.....	8
1.3 Laukiami rezultatai.....	8
2. Temos apžvalga ir analizė.....	9
2.1 Atpažinimas ir kliūtys .....	9
2.2 Skirtumų žemėlapis.....	11
2.2.1 Skirtumų žemėlapio formavimas .....	12
2.2.2 Kamerų nustatymas .....	13
2.2.3 Stereo poros .....	15
2.2.4 SŽ matematinis modelis .....	16
2.2.5 SŽ formavimo apribojimai .....	17
2.2.6 SŽ vaizdo formavimas.....	18
2.3 Pirminis apdorojimas.....	19
2.4 Dirbtinis neuroninis tinklas .....	22
2.4.1 DNT struktūra.....	23
2.4.2 DNT mokymas ir testavimas .....	24
2.4.3 Matematinė DNT išraiška.....	25
2.5 Skirtumo žemėlapio ir DNT sujungimas.....	26
3. Kliūčių atpažinimo kelyje uždavinio sprendimas .....	28
3.1 Skirtumų žemėlapis .....	28
3.1.1 Skirtumų žemėlapio formavimas.....	28
3.2 Skirtumų žemėlapio pirminis apdorojimas .....	32
3.2.1 Papildomas filtravimas.....	33
3.3 Dirbtinis neuroninis tinklas .....	34
3.3.1 DNT struktūros parinkimas .....	35
3.3.2 DNT mokymas ir testavimas .....	35
3.4 SŽ ir DNT sujungtas modelis.....	36
3.5 Galutinis sistemos vaizdas .....	39

3.6 Darbo realizacija.....	40
3.6.1 SŽ formavimas .....	40
3.6.2 SŽ filtravimas .....	41
3.6.3 SŽ segmento perdavimas DNT.....	42
3.6.4 DNT klasifikavimas.....	42
3.6.5 Atlikti bandymai atpažinti kliūtis .....	43
4. Rezultatai ir išvados .....	46
5. Informacijos šaltinių sąrašas .....	47

## Žodynėlis

**DNT** – (angl., *Artificial Neural Network*) Dirbtinis neuroninis tinklas;

**DN** – (angl., *Artificial Neuron*) Dirbtinis neuronas;

**SŽ** – (angl., *Disparity map*) Skirtumo žemėlapis;

**GŽ** – (angl., *Depth map*) Gylio žemėlapis;

**VSP** – (angl., *Single Layer Perceptron*) Vienasluoksnis perceptronas;

**DSP** – (angl., *Multi Layer Perceptron*) Daugiasluoksnis perceptronas;

**DI** – (angl., *Artificial Intelligence*) – Dirbtinis intelektas;

**CM** – (angl., *Convolution Matrix*) – Kompozicinė matrica.

# 1. Įvadas

Kliūčių aptikimas yra sudėtingas uždavinys, nes kliūtys neturi vienos konkrečios formos, apibrėžimo ar kilmės. Atpažinti kliūtis yra sudėtinga, nes jos dažnai neturi savybių, pagal kurias būtų galima lengvai identifikuoti. Dažniausiai paprastos mašinos nustatyti kliūtis negali, nes kliūtis identifikavimas yra kompleksinis uždavinys, apimantis kelias sritis, kurių pagalba įmanomas sprendimas. Viena iš tokių kompleksinio sprendimo dalių yra aplinkos vaizdo formavimas. Šioje vietoje labai tikslinga panaudoti kiek įmanoma realesnę aplinką atkuriantį vaizdą. Dažniausiai tai būna 3-matis aplinkos vaizdas, nes taip lengviau atpažinti kliūtis. Tačiau vien 3-mačio vaizdo nepakanka, reikia, kad jis dar būtų ir šiek tiek apdorotas, paruoštas analizavimui. Tokio vaizdo paruošimas yra kita kliūčių atpažinimo uždavinio sprendimo dalis. Jos metu iš aplinkos vaizdo pašalinama perteklinė informacija ir paliekama tik esminė. Tačiau šioje vietoje nėra jokio kito patogaus įrankio galinčio klasifikuoti kliūčių triukšmingame vaizde. Todėl naudojamas dinamiškas darinys, panašus į biologinį neuronų tinklą, galintį identifikuoti kliūtis net ir triukšmingame vaizde.

Šiuo metu dauguma vaizdo atpažinimo algoritmų yra skirti atpažinti plokščią vaizdą gautą iš vaizdo kameros, tačiau turint galimybę rekonstruoti 3-matį vaizdą galima žymiai pagerinti vaizdo atpažinimo kokybę. Tačiau atpažinimo uždaviniai tampa sudėtingesni ir labiau komplikuoti, nes papildomai tenka sukurti 3-matį modelį iš 2-matės erdvės vaizdo.

Suformuotas erdvinis vaizdas yra sunkiai interpretuojamas ir šioje vietoje labai tikslingą panaudoti dirbtinius neuroninius tinklus, kaip universalius klasifikatorius, kurie yra atsparūs triukšmingiems duomenims, nes kliūtys gali judėti erdvėje ar keisti formą.

## 1.1 Darbo problematika

Pasaulyje buvo atlikta daug bandymų sukurti sistemas, kurios galėtų sukurti aplinkos vaizdą ir jame klasifikuoti objektus ir kliūtis, tačiau būdavo susiduriama su problemomis, iš kurių pagrindinės yra tokios [H06]:

- Lėtas SŽ formavimo algoritmas;
- Iškraipytas, netiksliai suformuotas ar per daug triukšmo turintis SŽ vaizdas;
- Blogas SŽ skaidymo į segmentus algoritmas („praleidžiami objektai“);
- DNT blogai klasifikuoja objektus;
- Neiškiriami objektai SŽ naudojant DNT;
- Bloga DNT struktūra, blogas DNT mokymas, blogi testavimo duomenys;
- Neteisingai sukalibruotos kameros SŽ formavimui.

Šiame darbe nagrinėjama SŽ formavimo ir jo vaizde esančių kliūčių atpažinimo naudojantis DNT problematika, kurią galima išskaidyti į esmines problemas:

- SŽ vaizdo formavimas. Problema slypi SŽ formavime, nes SŽ vaizdas formuojamas iš dviejų paveikslukų, kuriuose vaizdas yra iškraipytas ir turi triukšmo;
- SŽ vaizdo pirminis apdorojimas. Čia problema yra perteklinės informacijos pašalinimas atrenkant kas yra reikalinga;
- SŽ vaizdo regiono išskyrimas – sunku pasirinkti objektų išskyrimo į regioną kriterijus;
- Išskirtojo regiono padavimas dirbtiniam neuroniniam tinklui. Problematiškas yra regiono parinkimas, kuriame gali būti ne visas objektas arba labai iškraipytas objektas;
- Objektų klasifikavimas naudojant DNT – problema yra dirbtinio neuroninio tinklo struktūros parinkime, apmokyme ir testavime.

## 1.2 Tikslai ir uždaviniai

Darbo tikslas – atpažinti kliūtis kelyje, naudojant skirtumo žemėlapio vaizdą ir dirbtinius neuroninius tinklus, kur kelio vaizdui naudojamas skirtumų žemėlapio vaizdas.

Darbo uždaviniai yra:

1. sukurti optimalų SŽ vaizdo formavimo algoritmą, kuris veiktų realiu laiku ir sukurtų tikslų SŽ vaizdą;
2. SŽ vaizdą segmentuoti taip, kad jame esantys objektai ir kliūtys būtų perduoti DNT;
3. SŽ vaizdo segmente dirbtinis neuroninis tinklas turi klasifikuoti kliūtis – gairelę arba stulpą.

## 1.3 Laukiami rezultatai

Siekama sukurti greitą SŽ formavimo algoritmą, kuris veiktų realiu laiku, atlikti pirminį SŽ vaizdo apdorojimą, objektų išskyrimą į regioną ir sukurtame SŽ vaizdo regione atpažintų kliūtis naudojant DNT.

Tikslui pasiekti bus sukurta sistema, kuri bus sudaryta iš trijų dalių:

1. SŽ vaizdo formavimas – sukurtas optimalus SŽ vaizdo formavimo algoritmas, gebantis formuoti SŽ vaizdus realiu laiku;
2. SŽ pirminis apdorojimas ir objektų išskyrimas į segmentus – sukurtas optimalus pirminio apdorojimo ir filtravimo mechanizmas, gebantis veikti realiu laiku;
3. SŽ vaizdo regione esančių kliūčių klasifikavimas ir įspėjimas apie jas naudojant DNT – parinkta struktūra bei mokymo ir testavimo metodikos kliūtims atpažinti.



## 2. Temos apžvalga ir analizė

### 2.1 Atpažinimas ir kliūtys

Atpažinimu vadinamas gebėjimas identifikuoti objektus naudojant sukaupią patirtį. Objektai, kurie nėra sukauptos patirties dalimi – yra neatpažįstami ir laikomi aplinkos dalimi.

Atpažinimas yra sudėtingas uždavinys, nes norimas atpažinti šaltinis yra kintamas:

1. Kiekvienas objektas klasėje yra bent kažkiek pakitęs lyginant su kitais objektas iš tos pačios klasės;
2. Kiekvienas objektas yra skirtingai apšviestas, todėl paprastas atitikimo tikrinimas netinka;
3. Objekto orientacija ir pozicijos gali būti pakitusios kameros ar stebėtojo atžvilgiu.

Jitendra Malik [J99] apibrėžia, kad objektai atpažįstami pagal bendriausias savybes. Dažniausiai tai būna vizualios ir nevizualios savybės.

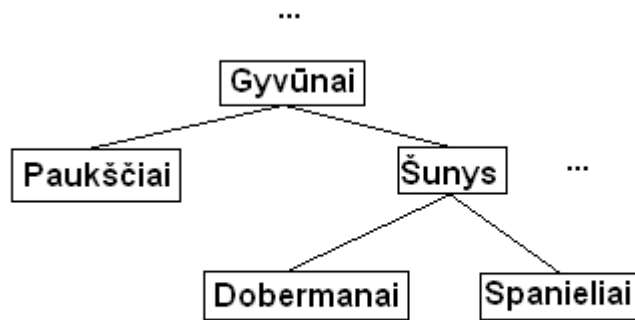
Dažniausiai naudojamos vizualios savybės:

- Spalva;
- Forma;
- Tekstūra;
- Judesys.

Dažniausiai naudojamos ne vizualios savybės:

- Kvapas;
- Skonis;
- Garsas.

Objektų atpažinimas paremtas kategorizavimu ir funkcija. Yra siūlomi keli variantai kaip tai traktuoti – objektus skirstyti pagal funkciją ir kategorizuoti. Kategorizavimas remiasi objektų skirstymu į klases ir kategorijas (1 pav.). Beveik visi mokslo pasaulio atstovai remiasi objektų skirstymu į kategorijas.



*1 pav. Objektų hierarchija*

Kliūtimis vadinami objektai, kurie yra subjekto judėjimo ar regėjimo kryptyje ir nejuda arba juda link subjekto. Dažnai objektai būna ne iki galo matomi ar iškraipyti, todėl juos atpažinti yra sunku.

Pagrindiniai kriterijai pagal kuriuos atpažįstamos kliūtys yra:

- Spalva;
- Forma;
- Padėtis erdvėje.

Kliūtis atpažinti yra sudėtinga, nes egzistuoja sunkinančios aplinkybės, kurios iškraipo aplinkos vaizdą.

Dažniausios atpažinimą sunkinančios aplinkybės:

- Daug įvairių objektų;
- Susimaišiusios arba vienodos objektų tekstūros;
- Didelis aplinkos triukšmas.

Norint nustatyti ar objektas yra aplinkos dalis reikia atsižvelgti į objektų ir subjekto judėjimą erdvėje. Subjekto atžvilgiu objektai gali:

- Nejudėti ir būti judančio subjekto kelyje;
- Nejudėti ir būti nejudančio subjekto kelyje;
- Judėti ir būti judančio subjekto kelyje;
- Judėti ir būti nejudančio subjekto kelyje.

Galiausiai reikia nustatyti ar objektas kelia grėsmę ar ne.

## 2.2 Skirtumų žemėlapis

Skirtumų (angl. *Disparity Map*) žemėlapis yra 3-matis aplinkos vaizdas pateiktas 2-mačiame vaizde, kuriame atstumai iki objektų išreiškiami spalvų intensyvumu. SŽ būna 2-jų rūšių – daugiaspalviai (angl., *color*) arba juodai balti (angl., *grayscale*). Informacinių technologijų srityje naudojami juodai balti paveikslukai, nes spalva laikoma pertekliniais duomenimis, kurie neneša tikslingos informacijos. Vaizdo gylis tokiuose SŽ yra išreiškiamas spalvų intensyvumu: arčiau esantys objektai yra šviesesnės spalvos, o toliau esantys – tamsesnės [RH04][SS01]. Kiekvienas pikselis tokiam žemėlapyje atitinka pikselį realaus vaizdo paveiksluke, kur juodai baltos spalvos (*grayscale*, angl.) intensyvumas SŽ atitinka atstumą iki kameros (2 pav.) [RH04] ir [SS01].



2 pav. Kairiosios ir dešinėsios kameros vaizdai bei realus skirtumų žemėlapyje vaizdas

Atstumas ir skirtumas iki objektų skirtumų žemėlapyje matematiškai išreiškiamas taip:

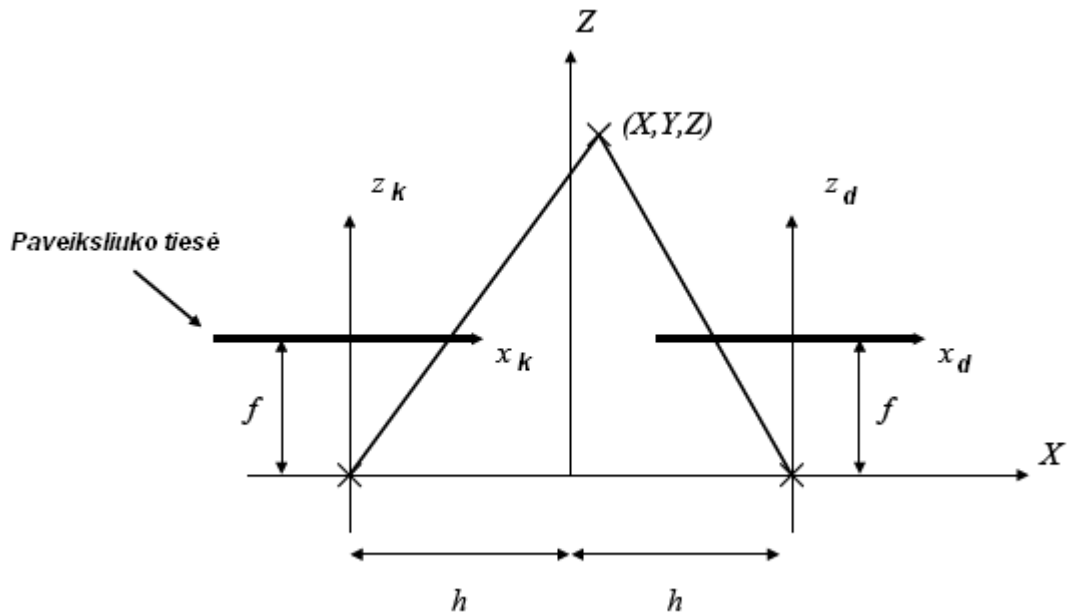
$$Z = \frac{2hf}{d} \quad (1), \text{ kur:}$$

$Z$  – atstumas nuo tiesės, kurioje stovi kameros, iki objekto,

$2h$  – atstumas tarp kamerų, kurios stovi vienoje tiesėje,

$f$  – atstumas nuo kameros atvaizdo iki pačios kameros,

$d$  – kamerų vaizdų horizontalaus pikselių poslinkio skirtumas.



3 pav. Stereo vaizdo geometrija

SŽ dažnai panaudojamas ten, kur reikalingas detalesnis aplinkos vaizdas. Dažnai tai būna 3-matis aplinkos vaizdas, gautas naudojant lazerinę, garso ar optinę įrangą. Pagrindinis paskutiniosios technologijos privalumas lyginant su kitais 3-mačiais aplinkos vaizdo formavimo metodais yra tas, kad nereikalinga jokia sudėtinga techninė įranga. Šiam SŽ vaizdai formuoti yra sukurti keletas algoritmų, tačiau jie pasižymi greičio ir tikslumo santykių skirtumų –greiti ir netikslūs arba lėti ir tikslūs, be to, formuojant SŽ dar susiduriama su įvairiais iškreipimais, triukšmais ir panašiomis problemomis. Paskutiniais šio šimtmečio metais sukurta nauja SŽ formavimo strategija, kuri leidžia efektyviau ir tiksliau formuoti SŽ vaizdą, tačiau skaičiavimų apimtimi yra sudėtingesnė už seniau sukurtąsias [D02], todėl SŽ formavimo algoritmo sudarymas vis dar tebėra aktualus uždavinys.

### 2.2.1 Skirtumų žemėlapių formavimas

Skirtumų žemėlapių formavimas yra vienos scenos dviejų ar daugiau vaizdų analizavimo ir/ar apskaičiavimo bei vertimo į 3-matį aplinkos vaizdą, išreikštą juodai balta spalva (*grayscale*, angl.) arba spalvotą vaizdą, procesas.

Per kelis dešimtmečius buvo sugalvota keletas SŽ formavimo algoritmų, kurie naudoja skirtingas SŽ formavimo strategijas:

- Požymiais paremta (*feature*, angl.) – kai SŽ vaizdas formuojamas pagal vaizde esančių objektų kontūrus ar kitus požymius;
- Sritimis paremta – kai SŽ formavimas yra paremtas vaizdo išskyrimo į sritis, pagal pilkai baltos (*grayscale*, angl.) spalvos intensyvumą;

- Faze paremta (*phase*, angl.) – kai SŽ formavimas yra paremtas Furjė faze, gradientinio skirtumo metodais, dažnai naudojamas, kad nepatekti į lokalių minimumą;
- Energija paremta (*energy*, angl.) – kai SŽ formavimas yra paremtas minimizavimo ir reguliarizavimo formulavimu.

Paskutinioji strategija yra jauniausia ir laikoma, kad yra tiksliausia, nes apima kitų strategijų privalumus ir išsprendžia visų strategijų trūkumus. Nedidelių vaizdų SŽ formavimui geriausia rinktis sritimis paremtą strategiją, nes ji yra greita, tačiau netiksli, tikslesnė yra požymiais paremta, bet ji nėra greita. Strategijų palyginime nėra energija paremtos strategijos, nes ji yra gan nesena, kad būtų galima pakankamai objektyviai vertinti palyginimą ir išrinkti greičiausią ir tiksliausią, todėl manoma, kad visos strategijos ar jų kombinacijos yra taikomos, priklausomai nuo to kokie yra tikslai.

Formuojant SŽ, realaus vaizdo taškų sutapimai apskaičiuojami keliais būdais [D02]:

- SSD (*Squared Sum Difference*, angl.) – SŽ vaizdo taškų atitikimas apskaičiuojamas naudojant kvadratinės sumos skirtumą;
- Normalized SSD (*Normalised Square Sum Difference*, angl.) – SŽ vaizdo taškų atitikimas apskaičiuojamas naudojant normalizuotą kvadratinės sumos skirtumą;
- NCC (*Normalized Cross-Correlation*, angl.) – SŽ vaizdo taškų atitikimas apskaičiuojamas naudojant normalizuotą kelių vaizdų koreliaciją (iš tikrųjų yra tas pats kas ir NSSD);
- SAD (*Sum of Absolute Differences*, angl.) – SŽ vaizdo taškų atitikimas apskaičiuojamas naudojant absoliutaus skirtumo sumą;
- Zero Mean SAD – SŽ vaizdo taškų atitikimas apskaičiuojamas naudojant nulinės reikšmės absoliutaus skirtumo sumą;
- Ranking – SŽ vaizdo taškų atitikimas apskaičiuojamas naudojant reitingus;
- Census – SŽ vaizdo taškų atitikimas apskaičiuojamas remiantis cenzais.

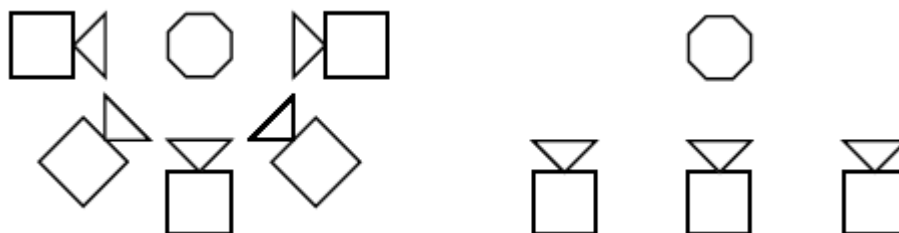
Dažnai sutinkamas atvejis, kad kombinuojami taškų atitikimo apskaičiavimai, įvedant atsižvelgimo į apribojimus pataisas ar įvairiais modifikacijas greičiui ir tikslumui pasiekti.

## 2.2.2 Kamerų nustatymas

Vienos scenos SŽ formuoti reikia bent dviejų kamerų paveikslėlių. Šiems paveikslėliams gauti yra du būdai (4 pav.) [H06]:

- Fotografuoti objektus iš daugelio kampų nukreipus kameras į tą patį objektą – kai kamerų vaizdo židinių tiesės susikerta fotografuojamame objekte;

- Fotografuoti objektą kameromis, pastatytomis vienoje tiesėje – kai kamerų vaizdo židinių tiesės yra lygiagrečios.

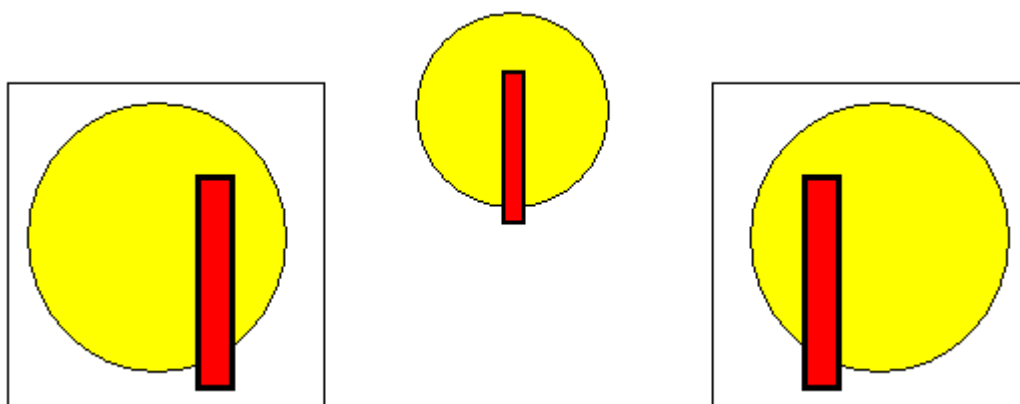


4 pav. Kamerų išdėstymas: apsuptai apie objektą ir vienoje tiesėje

Pirmuoju atveju kameros turi apsupti objektą bent dalinai, kas nėra patogiu, bet iš kitos pusės, tai padeda gauti fotografijas iš įvairių kampų nukreipiant kameras tiesiai į objektą, esantį kamerų vaizdo židinių tiesių susikirtimo vietoje. Antruoju atveju nėra galimybės gauti fotografijų iš įvairių kampų, bet naudojantis epipoliariniu apribojimu (*epipolar constrain*, angl.) abiejuose kameroje gautuose vaizduose tampa matomi bendri vieno ar kelių objektų taškai. Šiuo atveju egzistuoja dar tam tikrų apribojimų, kurių pagalba pasinaudojus galima paprasčiau nei pirmuoju atveju suformuoti SŽ.

Jeroen Hol ir Thomas Ericsson [*H06*] taip teigia, kad antrasis būdas yra labiau priimtinas, nes:

- pirmuoju būdu kuriant SŽ būtų nežinoma kokio dydžio kliūtis gali būti, kurios apsuptų vaizdų reiktų gauti;
- dažniausiai nežinomas tikslus atstumas iki objektų;
- dažniausiai žinoma, kad objektas ar objektai yra beveik vienodai nutolę nuo kamerų.



5 pav. Stereo vaizdo kūrimo principas – kamerų gauti paveikslukai (šonuose) ir realus vaizdas (viduryje)

### 2.2.3 Stereo poros

SŽ vaizdai formuojami iš dviejų ar daugiau paveikslėlių, gautų fotografuojant objektą kameromis išdėstytomis aplink objektą ar lygiagrečiai išdėstytomis vienoje tiesėje. Tokie paveikslėliai yra vadinami „stereoskopiniais“ paveiksliais arba tiesiog stereo poromis. SŽ yra lengviau formuojamas kai kameros yra išdėstytos lygiagrečiai vienoje tiesėje, todėl kiti kamerų išdėstymo atvejai SŽ formuoti nėra aktualūs.

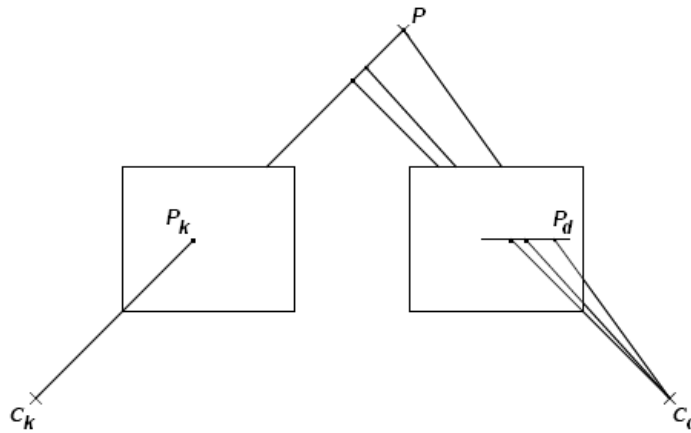
Stereo porų paveikslukams gauti yra naudojamos kameros iš tos pačios scenos, lygiagrečios viena kitai, pastatytos vienoje tiesėje, kurių vaizdo židinių tiesės yra lygiagrečios (3 pav.). Šis stereo vaizdo, tiksliau SŽ vaizdo, formavimas turi tam tikrų apribojimų [D02]:

- 1-matė paieška – dėl epipoliarinio apribojimo (*epipolar constrain*, angl.) paiešką užtenka atlikti horizontalioje vaizdo tiesėje;
- Vienodas abiejų kamerų atstumas iki objekto – atstumai turi būti lygūs, arba labai artimi lygiems, priešingu atveju sunku atvaizduoti objektus SŽ vaizde;
- Vienodas abiejų kamerų vaizdų šviesumas – turi būti naudojamas vienodas apšvietimas;
- Objektų unikalumas – kiekvienas taškas SŽ vaizde turi atitikti atitinkamą tašką viename (iš dviejų ar daugiau paveikslukų) paveiksluke;
- SŽ erdvės apribojimas – erdvė neturi būti begalinė, turi būti baigtinė;
- Požymių panašumas – objektai turi būti panašūs: ilgis, aukštis, forma – turi būti panašūs arba sutapti.

Pagrindinis iš jų yra tas, kad kameros turi būti vienoje tiesėje, t.y. kamerų vaizdo židinių tiesės turi būti lygiagrečios. Gylio apskaičiavimo metodika būtent ir remiasi šiuo apribojimu. Kitas svarbus apribojimas yra tas, kad kamerų atvaizdo židinio nuotoliai  $f$  lygūs. Ir tam, kad supaprastinti skaičiavimus, pridedamas apribojimas, kad kameros yra statmenos horizontaliajai plokštumai.

Kai du paveikslukai yra sukuriami iš skirtingų scenos pozicijų, 3D taško projekcija paveikslukuose gali būti skirtingose vietose. Horizontalusis poslinkis tarp taškų yra vadinamas skirtumu, ir žymimas  $d$ . Norint rasti tokį pikselių poslinkį, reikia paskaičiuoti atitikimą tarp pikselių tuose dvejuose paveikslėliuose. Tai yra pagrindinė atitikimo problema ir ji atsako į klausimą „Kuris taškas kairiajame paveiksluke atitinka tikrąjį tašką dešiniajame paveiksluke?“. Šiai problemai išspręsti, reikia pasirinkti pikselį viename paveiksluke, tada ieškoti 2-matėje erdvėje aplink tą patį pikselį kitame paveiksluke, kad surasti atitinkantį tašką arba geriausiai atitinkantį tašką pirmajame paveiksluke. Galima parodyti, kad 1-matės erdvės paieškos užtenka dėl epipoliarinio apribojimo (*epipolar constraint*, angl.), nes šio apribojimo teorema įrodo, kad

jei  $P_k$  yra scenos objekto  $P$  projekcijos taškas, tai tos pačios scenos projekcijos taškas  $P_d$  bus epipoliarinėje tiesėje. Epipoliarinė tiesė yra tiesė einanti per kamerų atvaizdų plokštumumas ir kurioje yra  $P_k$  ir du projekcijų centrai  $C_k$  ir  $C_d$  tiesė (4 pav.).



6 pav. Epipoliarinio apribojimo pavyzdys

Jei kameros yra statmenos plokštumai ant kurios jos pastatytos, tai epipoliarinė tiesė sutampa su horizontaliąją paveiksluko linija (6 pav.). Kadangi tėra tik horizontalusis skirtumas, tai skaičiavimai supaprastėja. Tikrąjį skirtumą dažniausiai tikimasi gauti, pavyzdžiui, tarp 0 ir 50 pikselių ir tai padaro paieškos erdvę mažesnę ir sumažinta skaičiavimų kiekį. Tokiu būdu padidinamas SŽ vaizdo formavimo algoritmo greitumas.

Kai žinomas poslinkis (taško padėties poslinkis horizontalioje padėtyje kamerų vaizdų atžvilgiu), galima gauti skirtumą supaprastinus skaičiavimus pikselių „poslinkyje“ (pikseliais) tarp dviejų taškų.

Sutrumpinę viską ką anksčiau aptarėme, mes gauname:

- Kameros turi tą pačią paveiksluko plokštumą ir laikoma, kad vienodo ilgio židinio nuotolius;
- Kameros yra statmenos horizontaliai plokštumai ant kurios stovi;
- Skirtumas yra apskaičiuojamas kaip horizontaliosios pikselio padėties skirtumas tarp to pačio taško dviejuose kamerų vaizdo paveikslukuose.

## 2.2.4 SŽ matematinis modelis

Horizontalioji taško padėtis abiejų kamerų paveikslėliuose matematiškai apskaičiuojama pagal atitinkamas lygybes (2.1):

$$x_k = f \frac{h + X}{Z}, \quad x_d = f \frac{X - h}{Z} \quad (2.1)$$

y koordinatė yra ta pati abiejuose paveikslėliuose remiantis epipoliariniu apribojimu (6 pav.).



Tai reiškia, kad nėra vertikalaus skirtumo:

$$y_k = f \frac{Y}{Z}, \quad y_d = f \frac{Y}{Z} \quad (2.2)$$

Horizontalus skirtumas yra laikomas horizontaliuoju pikselių poslinkiu tarp dviejų paveikslėlių:

$$d = x_k - x_d \quad (2.3)$$

Pagal lygybes (2.1) ir (2.3) perrašius skirtumą  $d$  ir išsprendus  $Z$ , gaunamasi atstumo išreiškimas iki objektų paveikslėliuose:

$$d = \frac{2hf}{Z} \Rightarrow Z = \frac{2hf}{d} \quad (2.4)$$

kur:

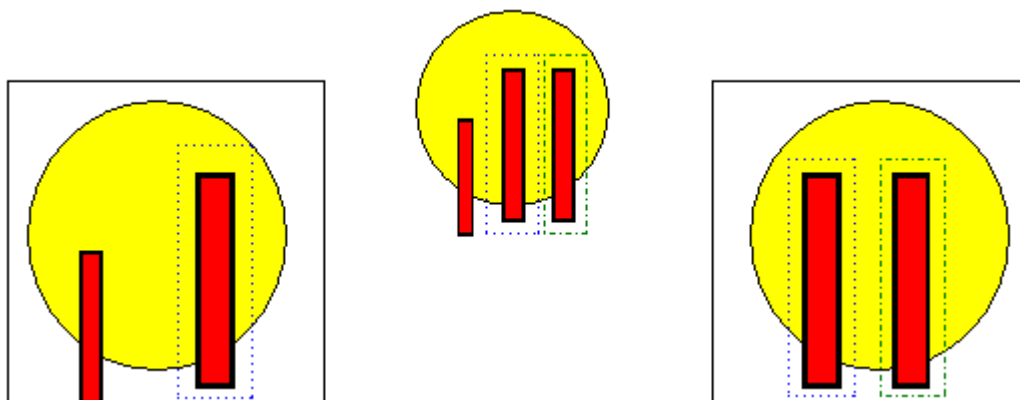
$Z$  – atstumas nuo kamerų iki objekto,

$2h$  – atstumas tarp kamerų,

$f$  – atstumas nuo kameros atvaizdo židinio iki kameros.

Stereo vaizdo teorija yra pakankamai paprasta ir ją galima supaprastintai išreikšti (2.1) – (2.4) lygtimis. Iš kitos pusės, išspręsti atitinkamą problemą yra netrivialu – vaizdo apdorojimo specialistai ją sprendžia jau daug laiko [BN04].

### 2.2.5 SŽ formavimo apribojimai



7 pav. Stereo vaizdo formavimo problematika – kamerų gauti paveikslukai (šonuose) ir realus, kur  $\cdots\cdots\cdots$  pažymėtas abiejuose kameroje matomas vaizdas, o  $\cdots\cdots\cdots$  pažymėtas tik dešiniojoje kameroje matomas vaizdas

Viena iš pagrindinių problemų, su kuriomis susiduriama yra ta, kad labai sunku nustatyti, to paties taško poziciją dviejų kamerų vaizduose. Jei scenos vaizde vienas iš objektų yra labai arti vienos kameros, tai kitoje kameroje to objekto gali nesimatyti ir dėl to sunku sudaryti SŽ, nes tampa neaišku kuris objektas yra kuris, nes jei atsikartoja tekstūra ar kitas objektų elementas, tampa labai sunku identifikuoti (7 pav.) kur yra kas.

Kita problema, jei objektai yra apšviesti skirtingomis spalvomis, skirtingu intensyvumu ir iš skirtingų kampų. Atsiranda poreikis tikrinti ar atitinka pikseliai ar jų grupės abiejuose paveikslukuose.

Daugumoje literatūros šaltinių siūloma imti vieną iš dviejų paveikslukų kaip pagrindinį, o kitą paveiksluką kaip pirmojo vaizdo skirtumą ir pagal šį skirtumą formuoti SŽ. Bet tai tik padeda išspręsti anksčiau minėtus apribojimus: sunku nustatyti, kuris objektas yra kuris ir kur iš tikrųjų turėtų būti tik vienoje kameroje matomas objektas, bei kaip teisingai nustatyti jo atstumai, kai yra skirtingi apšvietimai.

Be minėtų problemų egzistuoja ir kitos problemos, tokios kaip laiko sąnaudos, resursų ištekliai ir kitos, kurios iš tikrųjų tampa antrame plane prie laiko ir resursų panaudojimo optimizavimo. Minėtoms problemoms spęsti būtų tinkama sritimis paremta SŽ formavimo metodika, nes ji remiasi sričių išskyrimu pagal tam tikrus lygius, objektų dalis ir kitus požymius. Tačiau šiuo SŽ formavimo metodikos atveju reikia abu paveikslukus – stereo poras, iš kurių formuojamas SŽ, išanalizuoti ir surasti tas sritis, kurios nutolusios kaip sluoksniai toliausiai ir kurios yra arčiausiai, išskaidyti į sluoksnių aibę. Toks procesas reikalauja daugiau laiko ir resursų lyginant su kitomis SŽ formavimo metodikomis ir yra pakankamai sunku įgyvendinti vykdamą realiu laiku. Dar būtų tinkama išskyrimo į langus (angl., *Window*) metodika, bet ji taip pat reikalauja daug laiko ir sunku realizuoti realaus laiko skaičiavimą.

## 2.2.6 SŽ vaizdo formavimas

SŽ vaizdo formavimo algoritmui yra išskiriami esminiai etapai [SS2001], kurie turi būti įgyvendinti norint sukurti SŽ vaizdą:

1. Lyginami du scenos vaizdai pasirenkant vieną iš dviejų paveikslukų kaip pagrindinį, o kitą kaip pirmojo vaizdo skirtumą;
2. Pagal pirmojo ir antrojo vaizdų skirtumą sudarinėjamas SŽ;
3. Iš esamų sukurtų vaizdų kuriami juodai balti (*greyscale*, angl.) vaizdai – pašalinama perteklinė informacija, nes DNT turi apdoroti daug duomenų, todėl pateikiama tik esminė bei optimali;
4. Juodai baltuose vaizduose išskiriami objektų kraštai ar objektų formų kontūrai – tai apdorojus DNT lengviau atpažinti objektus pagal formą, nes jie išskiriami visame vaizde;
5. Išskiriami objektai – pagal iškirtas formas ar objektų kraštų kontūrus išskiriami patys objektai;

6. Objektai užpildomi juodai balta (*greyscale*, angl.) spalva (jeigu tai įmanoma), kur spalvos intensyvumas priklauso nuo objekto padėties erdvėje ir objekto formos – taip gaunamas trimatės erdvės pojūtis ir DNT tampa paprasčiau atskirti ir išskirti objektus.

siūlomo algoritmo psiaudokodas:

**Duomenys:** Stereoskopiniai paveiksliukai

**Rezultatas:** Skirtumų žemėlapis

inicializacija;

**for** Visos pakopos **do**

    Paimami kairysis ir dešinysis paveiksliukai;

    Apskaičiuojamos kompleksinės filtro reikšmės;

**end**

**for** Visos pakopos **do**

**for** *Visos Iteracijos* **do**

        Deformuojamos filtro reikšmės,  $q_l$  atitinkamai nuo pridedomojo skirtumo;

        Apskaičiuoti  $\Delta\phi$  ir  $\omega_x$ , (lokali fazė ir dažnis);

        Apskaičiuoti  $c$  ir  $d$ , (atitikimas ir skirtumų matricos);

        Filtruojamas (perkeičiamas)  $d$  ir  $c$ ;

        Pridedamas  $d$  ir  $c$  prie filtruotų  $d$  ir  $c$  atitinkamai;

**end**

    Interpoliuojamas pridėtasis  $d$  ir  $c$ , kad atitiktų kitos pakopos lygį;

**end**

## 2.3 Pirminis apdorojimas

Pirminiu apdorojimu yra vadinamas gautojo SŽ vaizdo paruošimas dirbtiniam neuroniniam tinklui taip, kad DNT beliktu tik klasifikuoti objektus. Šis paruošimas apima vaizdo perteklinės informacijos ir triukšmo pašalinimą. Pirminis apdorojimas reikalingas tam, kad suformuotą SŽ vaizdą tiesiogiai paduoti į DNT yra nelogiška, nes DNT tampa perkrautas pertekline informacija ir labai apsunkina DNT darbą. Kad to nebūtų, klasifikuojamas vaizdas yra papildomai apdorojamas – atliekamas pirminis apdorojimas.

Pirminiam apdorojimui taikomi įvairūs metodai perteklinei informacijai nustatyti ir ją pašalinti, kai kuriais atvejais, metodai, priklausomai nuo sąlygų, turi papildomų kriterijų, pagal kuriuos nustatomas perteklinės informacijos būvimas.

### **Pavyzdžiui:**

Sistema turi klasifikuoti apskritimą ir kūgį, todėl galima imti SŽ vaizdo segmentą, kuriame būtų objektų kraštai, ko užtenka DNT klasifikavimui.

Pirminiam apdorojimui priskiriami tokie galimi grafiniai SŽ vaizdo apdorojimai [G01]:

- Objekto ar kelių objektų išskyrimas SŽ vaizde pagal spalvą;
- Objekto ar kelių objektų išskyrimas SŽ vaizde pagal formą;
- Objekto ar kelių objektų išskyrimas SŽ vaizde pagal aplinką;
- SŽ vaizdo filtravimas naudojant triukšmo šalinimo funkciją;
- SŽ vaizdo filtravimas naudojant taisykles.

Labiausiai aktualus yra objekto ar kelių objektų išskyrimas SŽ vaizde pagal formą, jos kontūrus ar kraštus. Tam naudojamas kompozicinės matricos (angl., *Convolution Matrix*) filtravimas, kurio metu gaunamas objekto ar kelių objektų kontūras. Filtravimas įgyvendinamas dauginant kiekvieną SŽ vaizdo pikselį ir jo aplinkinius kaimynus iš 3x3 arba 4x4 matricos elementų ir pakeičiant pikselį gautąja visų sudaugintų elementų pikselių suma. Į skaičiavimus neįtraukiami visi kraštiniai paveiksluko taškai. Viso proceso supaprastinta formulė, kai filtravimo matrica yra 3x3 atrodo taip:

$$P = \begin{bmatrix} x_{11} & x_{21} & x_{31} & x_{41} & \dots & x_{m1} \\ x_{12} & x_{22} & x_{32} & x_{42} & \dots & x_{m2} \\ x_{13} & x_{23} & x_{33} & x_{43} & \dots & x_{m3} \\ x_{14} & x_{24} & x_{34} & x_{44} & \dots & x_{m4} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{1n} & x_{2n} & x_{3n} & x_{4n} & \dots & x_{mn} \end{bmatrix} * \begin{bmatrix} k_{11} & k_{21} & k_{31} \\ k_{12} & k_{22} & k_{32} \\ k_{13} & k_{23} & k_{33} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{21} & x_{31} & \dots & x_{m1} \\ x_{12} & \bar{x}_{22} & \bar{x}_{32} & \dots & x_{m2} \\ x_{13} & \bar{x}_{23} & \bar{x}_{33} & \dots & x_{m3} \\ \dots & \dots & \dots & \dots & \dots \\ x_{1n} & x_{2n} & x_{3n} & \dots & x_{mn} \end{bmatrix} \quad (3)$$

$$\bar{x}_{ij} = \sum_{i-1}^{i+1} \sum_{j-1}^{j+1} x_{ij} k_{ij} \quad , \text{ kur } 1 < i < n, \quad 1 < j < m \text{ ir } 0 \leq \bar{x}_{ij} \leq 255$$

### **Pavyzdys:**

$$\bar{x}_{22} = (x_{11}k_{11} + x_{21}k_{21} + x_{31}k_{31} + x_{12}k_{12} + x_{22}k_{22} + x_{32}k_{32} + x_{13}k_{13} + x_{23}k_{23} + x_{33}k_{33})$$

Vaizdo gerinimui ir triukšmo šalinimui yra naudojamas vidurkio (angl., *Median filter*) filtras, kuris „išlygina“ pikselių reikšmes pagal aplinkinių kaimynų vidutinę reikšmę (8 pav.). Imant paveiksluką kaip dvimatį masyvą, išlyginamos „smarkiai išsišokančios“ reikšmės. Pašalinis šio filtravimo efektas – išryškunami kraštai.

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Mediano reikšmė: 124

Kaimynų reikšmės:

115, 119, 120, 123, 124,  
125, 126, 127, 150

8 pav. Mediano filtravimas pagal aplinkinių kaimynus

Papildomai sumažinti informacijos kiekiui SŽ, vaizdo dalį galima sumažinti iki tiek, kad matytųsi tik kelio dalis ir dalinai būtų apkarpyti SŽ vaizdo šonai (9 pav.).

Pirminis apdorojimas atlieka dar ir kitą darbą – parenka optimalų SŽ vaizdo segmento dydį, nes jei segmentas yra per didelis arba per mažas, DNT gali blogai identifikuoti SŽ vaizdo fragmente esantį objektą.

Išskirto objekto, esančio SŽ vaizdo segmente, padavimas dirbtiniam neuroniniam tinklui yra komplikotas, nes ne visas objektas yra matomas aplinkoje, todėl pirminio apdorojimo metu būtina parinkti optimalų išskirto objekto pateikimo dirbtiniam neuroniniam tinklui būdą.



9 pav. Atribota matomo vaizdo dalis

Nicolas Soquet ir Donald Rosselot and Ernest L. Hall [N07] [D04] darbų praktika rodo, kad tik tam tikros važiuojamosios kelio dalies išskyrimas nėra tikslingas, nes pakankamai didelė dalis tos srities vis tiek lieka nepanaudota. Todėl autoriai siūlo važiuojamosios kelio dalies sritį dar labiau sumažinti, kad būtų stebima tik arti esanti važiuojamoji dalis, į perspektyvą nukreiptoji transporto priemonės pločio dalis stabdymo keliui ir horizonto linija, kurioje būtų tiesiog stebima ar nėra jokių objektų.

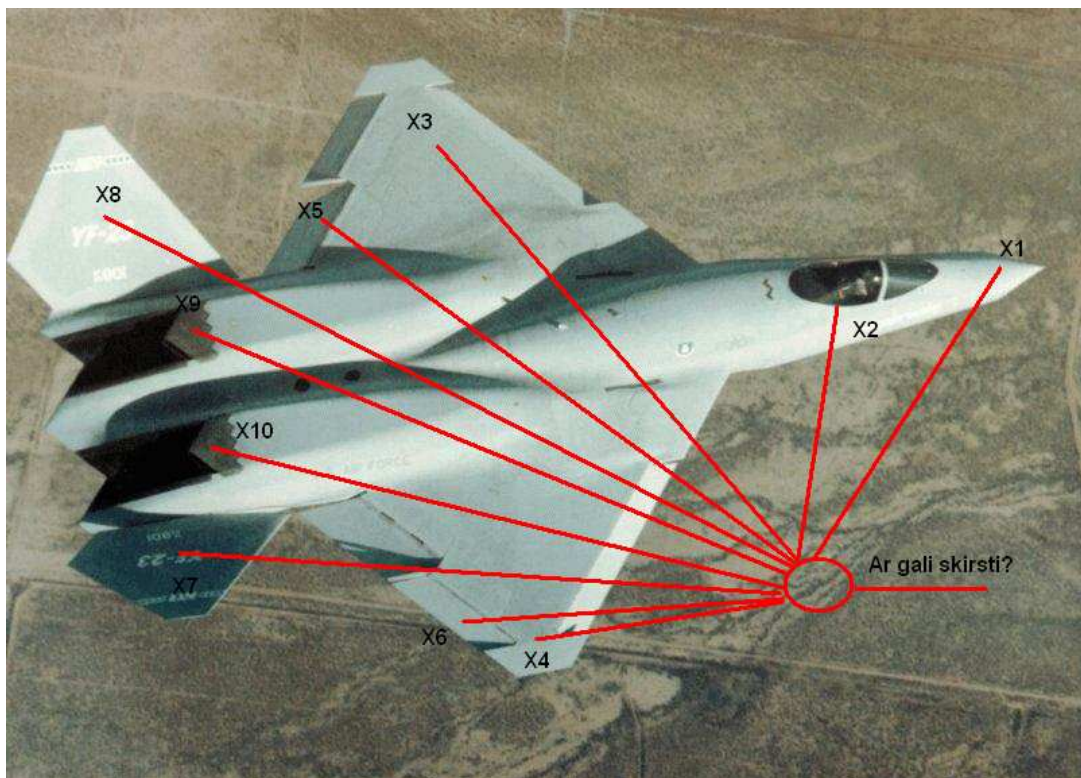
Šitokio dydžio kelio vaizdo apdorojimo pakanka, kad nustatyti, ar kelyje yra kliūtis. Pasiūlytas autorių būdas nustatyti ar kelyje yra kliūtis yra skirtas tiesiog kliūčių aptikimui, bet ne klasifikavimui.

## 2.4 Dirbtinis neuroninis tinklas

Dirbtinis neuroninis tinklas – tai matematinis ar skaičiuojamasis modelis, paremtas biologiniais neuronų tinklais. Jis sudarytas iš susietų dirbtinių neuronų grupių ir apdoroja informaciją skaičiavimui naudodamas sąryšius tarp neuronų [W07]. Dažniausiai DNT yra adaptyvi sistema, kurios struktūra kinta dinamiškai priklausomai nuo išorinių ar vidinių duomenų, kurie naudojami dirbtinių neuroninių tinklų apmokymui. Praktiniu požiūriu, DNT yra netiesinis statistinis duomenų modeliavimo įrankis.

DNT veikia panašiai kaip ir biologinis neuronų tinklas – priima informaciją, ją apdoroja ir išveda rezultatą. Informacija priimama iš išorinio šaltinio arba kito neurono. Priimama informacija yra diskreti. Perduodama informacija yra arba kitam neuronui, arba tam, kas siejasi su DNT išvestimi.

Robbert Callan [K01] duoda paprasčiausiais DNT veikimo panaudojimo pavyzdį, pavaizduotą paveiksliuke (10 pav.): DNT nusprendžia ar lėktuvas yra tinkamas skristi, ar reikalingas avarinis nusileidimas, remdamasis įvairių įrenginių ir daviklių parodymais, kurių kiekvieno svoris yra kitoks bei naudodamas slenkstinę aktyvavimo funkciją, kuri duoda tik dvi išvestis (gali skirsti arba reikia nusileisti).



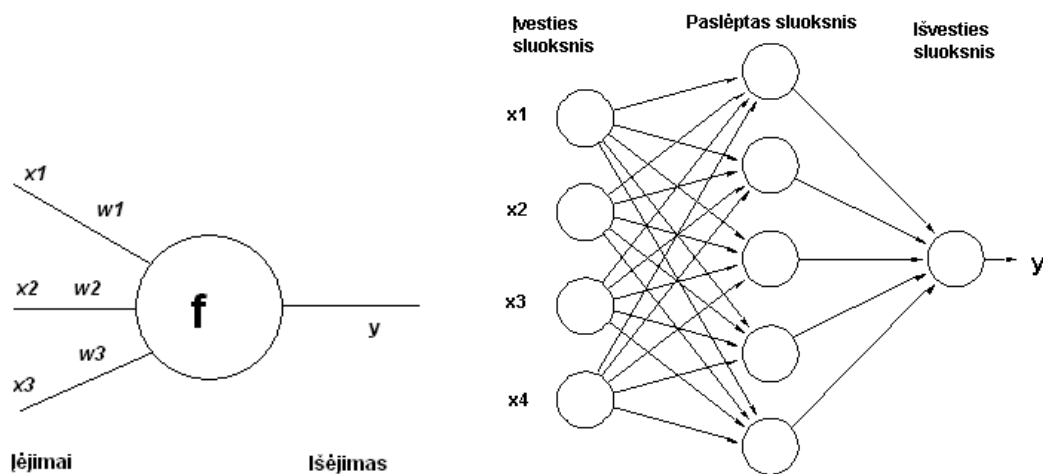
10 pav. Lėktuvo būsena nustatoma naudojant DNT

Atpažinimui būtų galima naudoti įvairius matematinius statistinius atpažinimo metodus, tačiau DNT labiau tinka, nes jo struktūra yra dinamiška: apmokymo metu dinamiškai keičiasi, artėja prie realaus modelio. Be to, DNT dažniausiai naudojamas ten, kur pradiniai ir tarpiniai duomenys nėra pastovūs, yra iškraipyti ir su triukšmu, kur paprasti metodai negali tiksliai klasifikuoti pagal iškraipytus duomenis.

## 2.4.1 DNT struktūra

DNT būna vienasluoksniai ir daugiasluoksniai. Vienasluoksniai DNT yra sudaryti iš vieno neuronų sluoksnio ir gali turėti daugiau nei vieną įėjimą ir išėjimą (11 pav.). Daugiasluoksnių perceptronų tinklas gali turėti daugiau nei vieną įėjimą ir išėjimą, ir daugiau nei vieną paslėptą sluoksnį (11 pav.). Vienasluoksnis tinklas neturi vidinės struktūros, nes paduodami duomenys tiesiai į perceptronus ir iškart po išvedami klasifikuoti. Daugiasluoksnis tinklas turi įvairią struktūrą, su grįžtamuoju ryšiu ir be jo, bei kitų „anomalijų“.

Vienasluoksniai su savo paprasčiausia struktūra gali išspręsti tik tiesinius uždavinius. Norint spręsti sudėtingesnes problemas ar uždavinius, reikia naudoti daugiasluoksnius, todėl vaizdo atpažinimui yra naudojami daugiasluoksniai DNT [H05].

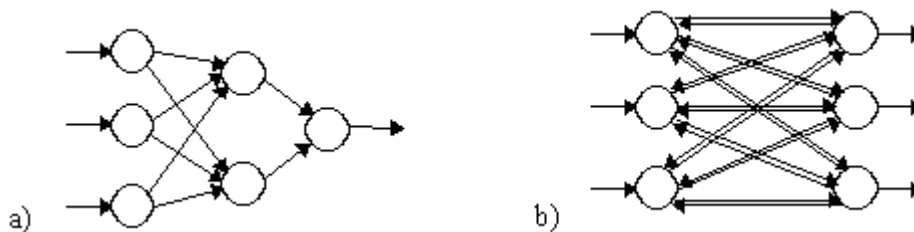


11 pav. Vienasluoksnis ir daugiasluoksnis perceptronai

DNT tinklai būna įvairių struktūrų, skirtingų paslėptų sluoksnių kiekių, su grįžtamaisiais ryšiais ir panašiai. Dažniausiai naudojamos DNT struktūros [P99] [K01] yra šios:

1. Feedforward neuronų tinklai (VSP, DSP, Adaptyvus Linijinis Elementas (Neuronas), RBF („Radial Basis Function“), SOM („Self Oriented Map“) ir aibė kitų) – tai tokie DNT tinklai, kuriuose informacija juda viena kryptimi, nuo įėjimo sluoksnio į paslėptą sluoksnį (jei DNT yra daugiasluoksnis) ir iš ten į išėjimo sluoksnį (12 a pav.);

2. Feedback neuronų tinklai – tai tokie DNT, kur informacija juda ne tik viena kryptimi – nuo įėjimo sluoksnio link paslėpto sluoksnio ir iš ten gali judėti į išėjimo sluoksnį, bet ir turi grįžtamąjį ryšį – gali eiti nuo išėjimo sluoksnio į paslėptą sluoksnį ar nuo paslėpto sluoksnio į įėjimo sluoksnį (12 b) pav.);
3. Aibė kitų DNT, įvairių jų modifikacijų, neįvardintų pirmame ir antrame punkte.



12 pav. Skirtingos DNT struktūros: a) Feed-Forward tinklas; b) Feed-Backward tinklas;

Michiel W.H. Remme ir Wytse J. Wadman [M04] nurodo, kad kiekviena iš išvardintų struktūrų turi savus privalumus ir trūkumus, taikymo sritis ir kitus tik jiems vieniems būdingus bruožus. Todėl vaizdo atpažinimui dažniausiai naudojamos FeedForward arba BackForward DNT struktūros nes jos yra sukurtos taip, kad jas apmokius jos sukaupią patirtį laikytų ir naudotų atpažinimui.

## 2.4.2 DNT mokymas ir testavimas

DNT darbas paremtas mokymu ir testavimu, kurio metu DNT minimizuoja kainą arba paprasčiau tariant kiekvieno mokymo iteracijos metu patikslina savo svorio koeficientus – keičia vidinę struktūrą. DNT besimokydamas gali persimokyti ar patekti į lokalųjį minimumą. Šiai problemai spręsti siūloma arba pakeistas iteracijų žingsnio dydis, kad DNT sugebėtų išlipti iš lokalaus minimumo, arba paprasčiausiai keisti pradines sąlygas (kas dažnai nelabai tinka).

Visi DNT gali būti apmokomi juos stebint ir leidžiant jiems patiems mokytis. Pagal tai, DNT skaidomi į dvi kategorijas:

- Prižiūri;
- Neprižiūrimi.

Prižiūrimajame mokyme, yra žinomas rezultatas, todėl DNT yra „verčiamas“ keisti savo struktūrą, kol pasiekia norimą rezultatą. Neprižiūrimajame mokyme, DNT mokymasis paremtas duomenimis ir jokio tikslo rezultato įvesčiai nėra nurodyta, pats DNT turi apskaičiuoti rezultatą.

DNT mokymui ir testavimui yra sugalvota daug metodikų ir būdų, kai kurie iš jų yra specifiniai ir taikomi tik konkrečioms DNT struktūroms ar jų atmainoms, tačiau egzistuoja keletas labiausiai naudojamų:



- Porinių ir neporinių elementų imtyje paskirstymas po lygiai mokymui ir testavimui atitinkamai;
- Santykinio dydžio parinkimas, pavyzdžiui: 67% imties mokymuisi, o likę 33% - testavimui;
- Atsitiktinis imties parinkimas mokymui ir atsitiktinis imties parinkimas testavimui.

Labai svarbu DNT mokymui ir testavimui yra parinkti pakankamą duomenų imčių dydį, nes dėl per mažos duomenų imties DNT gali blogai išmokti.

### 2.4.3 Matematinė DNT išraiška

Matematinis neurono modelis išreiškiamas taip [H05]:

$$y = \varphi(w_0x_0 + w_1x_1 + \dots + w_nx_n) = \varphi\left(\sum_{j=0}^n w_jx_j\right) \quad (4)$$

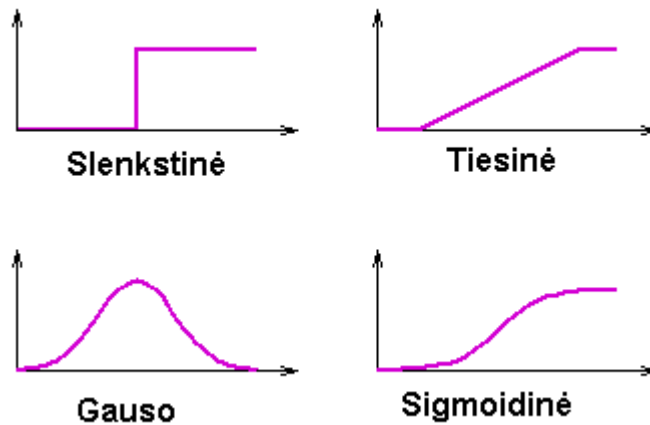
kur:  $\varphi$  – aktyvavimo funkcija,  $w_j$  – svorio koeficientas,  $x_j$  – įėjimo koeficientas,  $j \in 0,1,2,\dots,n$ ;

Matematinis DNT modelis išreiškiamas taip [H05]:

$$y_k(x+1) = \gamma_k(s_k(x)) = \gamma_k\left(\sum_j w_{jk}(x)y_j(x) + \theta_k(x)\right) \quad (5)$$

Labiausiai paplitusios aktyvavimo funkcijos yra šios (13 pav.):

- Slenkstinė –  $f(x) = \begin{cases} 0, & 0 > x \\ 1, & x \geq 0 \end{cases}$ , naudojama kai norima gauti 2 reikšmes: 0 ir 1 ;
- Tiesinė –  $f(x) = mx + b$ , kartais dar vadinama gradientinė, naudojama kai reikia išvesti reikšmę iš intervalo;
- Gauso –  $f(x) = ae - \frac{(x-b)^2}{2c^2}$ , naudojama ten, kur reikalingas staigus šuolis;
- Sigmoidinė –  $f(x) = \frac{1}{1+e^{-x}}$ , labai populiari ir dažnai naudojama DNT, nes paprasta išvestinė:  $f'(x) = f(x)(1-f(x))$ .



13 pav. Labiausiai paplitusios DNT aktyvavimo funkcijos

## 2.5 Skirtumo žemėlapiu ir DNT sujungimas

SŽ vaizdo kaip duomenų padavimas į DNT įėjimo sluoksnį apdorojimui yra vadinamas sujungimu.

SŽ vaizdą, kaip duomenų padavimą į DNT galima atlikti keliais būdais [A04]:

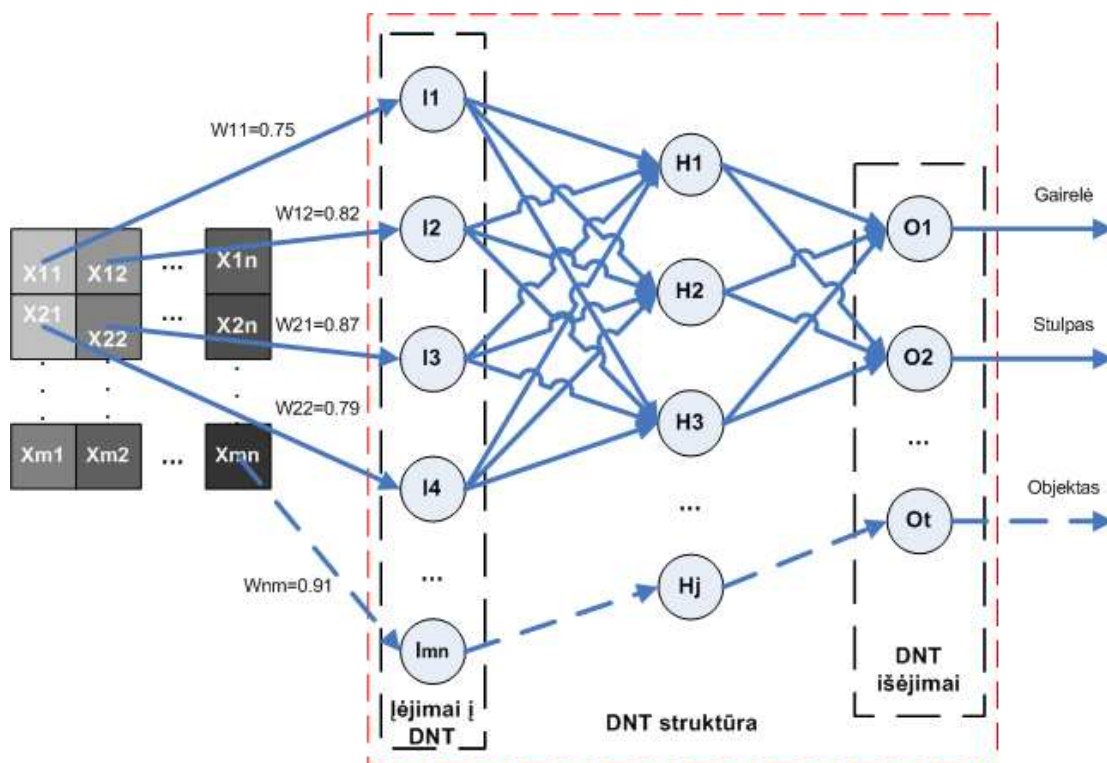
- Paduoti SŽ vaizdo skirtumus, intensyvumą, orientaciją – kai DNT perduodami detalizuoti SŽ vaizdo parametrai;
- Paduoti SŽ vaizdo taškus – kai kiekvienas  $n \times m$  dydžio SŽ vaizdo taškas perduodamas DNT į įėjimą;
- SŽ vaizdo formavimas naudojant atskirą DNT – kai dirbtiniam neuroniniam tinklui paduodamas apmokymui ir testavimui kamerų vaizdai ir gautasis SŽ vaizdas perduodamas į kitą DNT.

Labai svarbus momentas tas, kad pirmasis būdas tinkamas tada, kai SŽ vaizdo formavimui turima aukštos kokybės ir detali vaizdo medžiaga, iš kurios formuojamas SŽ vaizdas. Tačiau tokiu atveju, SŽ vaizdo formavimas užtrunka daug daugiau laiko, nei SŽ formavimo iš vidutinės ar žemos kokybės vaizdų, kur neįmanoma išgauti detalumo lyginant su antruoju būdu. Trečiuoju atveju, darbo greitis tiesiogiai priklauso nuo kamerų vaizdo dydžio, iš kurių norima suformuoti SŽ vaizdą, reikalingas atskiras DNT, kuris turi būti apmokytas ir kurio darbo greitis nėra realus ar artimo realiam laikui [B03][ZMJ03].

Mes nagrinėsime antrąjį būdą, kur kiekvienas SŽ vaizdo, kaip bitų sekos, taškas turi tam tikrą juodai baltos (angl., *grayscale*) spalvos gylį perduodamą DNT apdoroti. Duomenys į DNT paduodami per įėjimo sluoksnį (14 pav.), kur kiekvienas norimo klasifikuoti vaizdo pikselis yra paduodamas į dirbtinio neuroninio tinklo įėjimą [B03][W00]. Paduodamos duomenų struktūros

dydis yra ribotas – lygus dirbtinio neuroninio tinklo įėjimų skaičiui. Arba tiesiog lygus norimo klasifikuoti vaizdo pločio ir aukščio (dimensijos) sandaugai. Išėjimų skaičius lygus norimų klasifikuoti objektų skaičiui arba paprasčiau tariant, išvesties sluoksnio perceptronų kiekis lygus norimo klasifikuoti objektų skaičiui.

Į DNT paduoti visą SŽ vaizdą yra neoptimalu: labai didelis kiekis perteklinės informacijos stabdo DNT darbą, įneša triukšmo. Todėl reikalingas pirminis apdorojimas ir SŽ vaizdo segmentavimas – papildomi SŽ vaizdo optimizatoriai. Vaizdo regionas yra juodai baltos spalvos ir jame matosi tik objekto regionai, todėl informacijos kiekis SŽ vaizdo regione sumažintas maksimaliai. Kiekvienas tokio regiono pikselis perduodamas į DNT išėjimą.



14 pav. Vaizdo regiono padavimas į dirbtinio neuroninio tinklo įėjimus

### 3. Kliūčių atpažinimo kelyje uždavinio sprendimas

Kliūties atpažinimu kelyje užduotį formuluosime taip: sukurti dvejomis kameromis gautų atvaizdų skirtumų žemėlapi, kurį apdorojus pirminiu apdorojimu, perduosime į dirbtinį neuroninį tinklą klasifikavimui.

Prielaidos:

- dvi kameros yra pastatytos vienoje tiesėje, kurių atvaizdų židinių tiesės yra lygiagrečios;
- kameros yra vienodai nutolusios nuo norimo klasifikuoti objekto ar objektų;
- kameros yra statmenos horizontaliajai plokštumai;
- SŽ vaizdas formuojamas iš 2 paveikslėlių;
- SŽ vaizdas juodai baltos spalvos;
- SŽ turi būti sumažintas perteklinių duomenų kiekis, išskiriant ar sumažinant SŽ vaizde esančius objektus;
- SŽ vaizde po pirminio apdorojimo pašalinti viršutinę vaizdo dalį;
- SŽ vaizdo pirminio apdorojimo rezultatas – SŽ vaizdo regionas  $n \times m$  pikselių dydžio arba labai artimas jam.

Sąvokos:

- Kelias – SŽ vaizdo dalis, kurioje DNT turi identifikuoti kliūtis;
- Kelyje galimos pasitaikyti kliūtys – gairėlė arba stulpas (priklauso nuo paveikslėlių mokymo imties);
- SŽ vaizdo regionas – SŽ vaizdo dalis, kurioje yra norimas identifikuoti objektas ar objektai.

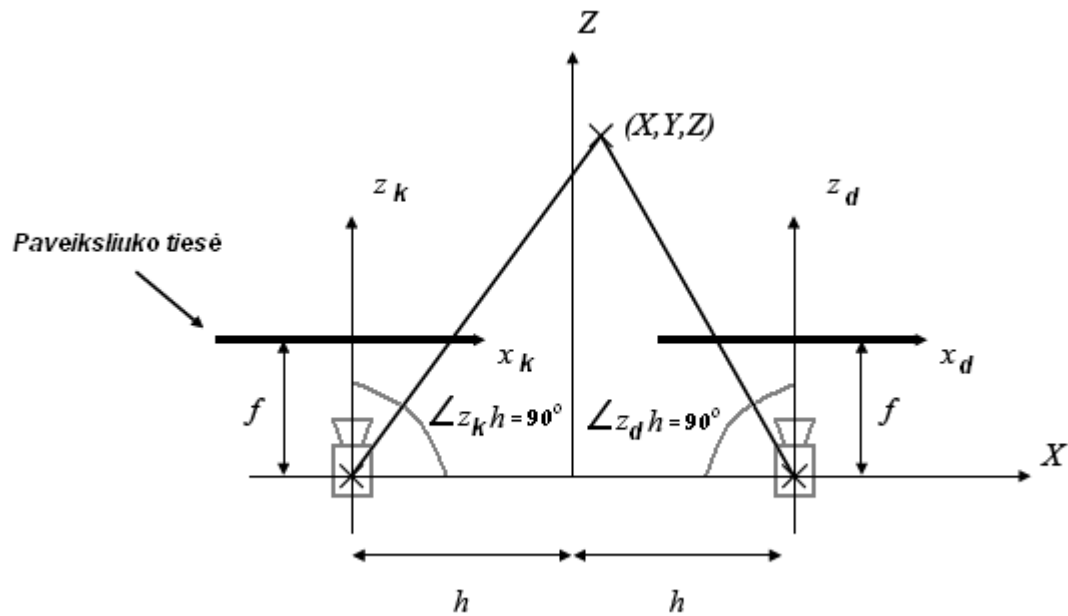
#### 3.1 Skirtumų žemėlapis

SŽ vaizdas formuojamas iš dviejų kamerų, išdėstytų vienoje tiesėje, kurių atvaizdo židinių tiesės yra lygiagrečios, gautų vaizdų, kur kameros nuo objektų nutolusios vienodu atstumu. Scenos vaizdų skirtumas randamas lyginant kairiojo  $I_k$  paveikslėlio  $P_k[1,1]$  tašką su dešiniojo paveikslėlio  $I_d$  tašku  $P_d[1,1+n]$ , kur  $n$  – yra pikselio poslinkis taško  $P_k[1,1]$  atžvilgiu.

##### 3.1.1 Skirtumų žemėlapio formavimas

SŽ vaizdo formavimo apribojimai:

- Kamerų atvaizdo židinių tiesės yra lygiagrečios viena kitai;
- Kameros yra statmenos plokštumai ant kurios jos yra;
- Kamerų židinių  $f$  atstumai iki objektų yra lygūs.
- Iš jų svarbiausias *epipolarinis* apribojimas (*epipolar constrain*, angl.), nes juo remiantis apskaičiuojamas pikselių atitikimas ir vaizdo gylis.



15 pav. Kamerų pozicijos ir apribojimų vaizdas (horizontalus pjūvis)

Remiantis šiais apribojimais ir epipolarinio apribojimo (*epipolar constraint*) teorema, galima SŽ formavimą supaprastinti iki vienodų taškų ieškojimo eilutėje, lyginant pikselius, paimtus iš kairiosios ir dešinėsios kamerų vaizdų eilučių. Gautas taškų gylis išreiškiamas panaudojant pilką spalvą, kur tamsesnės sritys reiškia didesnę atstumą, o šviesesnės - artimesnę. Ankstesniame skyriuje pateikta supaprastinta SŽ formavimo schema, pagal kurią sudaromas formavimo algoritmas, pavaizduota 15 paveiksliuke.

Įvairių straipsnių autorių pasiūlyti algoritmai yra per lėti, nes sunaudoja labai daug resursų, kurių būtų galima atsisakyti jei dalį tikslaus SŽ formavimo funkcijų paliktume dirbtiniam neuroniniam tinklui, kuris ir be šių žingsnių galėtų klasifikuoti objektus. Todėl nustatoma kurie žingsniai sunaudoja daugiausia resursų ir bandoma jų atsisakyti arba rasti optimizaciją:

1. Atidaromi analizei paveiksliukai – 10% (0,101 s);
2. Vykdomas kalibravimas – 50% (2,197 s);
3. Kuriamas SŽ prototipas 35% (0,0453s);
4. Atvaizduojamas SŽ 5% (0,010 s);

Kadangi tikslus SŽ sudarymo algoritmas naudoja daug resursų, pakoreguojamas SŽ formavimo algoritmas, vaizdo formavimui realiu laiku. Priimama, kad: kelio vaizde kliūtys bus matomos abiejuose paveikslukuose ir kliūtys bus nutolusios vienodu atstumu nuo kamerų.

SŽ vaizdo formavimo algoritmo psiaudo kodas atrodo taip:

```
Duomenys: Stereoskopiniai scenos paveikslukai
Rezultatas: Skirtumų žemėlapis
Paveikslėlių nuskaitymas
Vaizdo kalibravimo inicializacija
for Visas kairiojo paveikslėlio plotis do
    Paimamas kairiojo paveikslėlio pikselių stulpelis
    Paimtas stulpelis lyginamas su dešiniojo paveikslėlio stulpeliu
    Tikrinamas pikselių sutapimų kiekis
end
Apskaičiuojamas didžiausias sutapimų kiekis
Skirtumų žemėlapio formavimo inicializacija
for Visi kairiojo paveikslėlio stulpeliai do
    for Visi kairiojo paveikslėlio stulpelio pikseliai do
        Lyginami  $P_k$  ir  $P_d$  sutapimai tarp paveikslukų
        Skirtumams suteikiamos spalvos
    end
end
Atvaizduojamas skirtumų žemėlapis
```

Nagrinėtos literatūros autoriai siūlo žemiau pateiktą SŽ formavimo algoritmą, tačiau reikalingas greitesnis ir paprastesnis algoritmas, nes norima, kad sistema veiktų realiu laiku ir kurtų kiek įmanoma tikslesnį SŽ. Be to, šis algoritmas naudoja sritymis paremtą SŽ formavimo strategiją, kurią iš tikrųjų galima optimizuoti.

Todėl siūlomas algoritmas turėtų atrodyti panašiai į štai tokį:

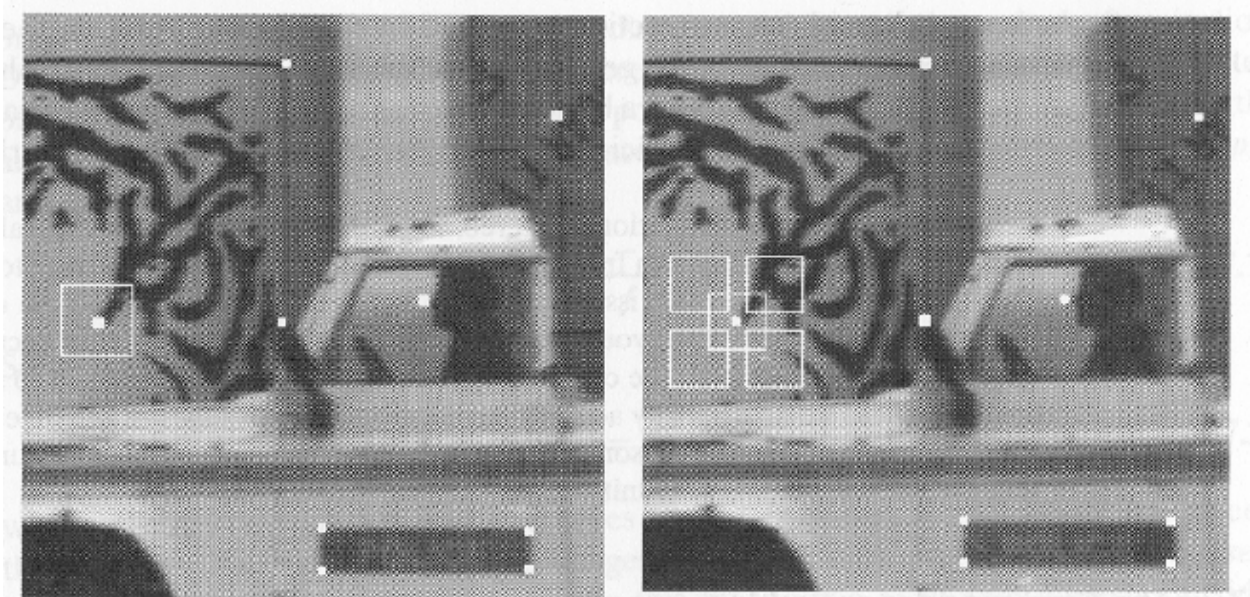
```
// ...
for (int i = 0; i < bitmap1.width.length; i++){
    for (int j=0; j < bitmap1.height.length; j++){
        //disparityBitmap[i, j] = 0 // pildomi SZ geriausio atitikimo taskai
        if((bitmap1.GetPixel(i,j).R == bitmap2.GetPixel(i,j).R)&&
            (bitmap1.GetPixel(i,j).G == bitmap2.GetPixel(i,j).G)&&
            (bitmap1.GetPixel(i,j).B == bitmap2.GetPixel(i,j).B)){
```

```

        disparityBitmap.SetPixel(i,j) = Color.RGB(0,0,0);
    }else if((bitmap1.GetPixel(i,j).R != bitmap2.GetPixel(i,j).R)&&
        (bitmap1.GetPixel(i,j).G != bitmap2.GetPixel(i,j).G)&&
        (bitmap1.GetPixel(i,j).B != bitmap2.GetPixel(i,j).B)){
        disparityBitmap.SetPixel(i,j) = Color.RGB(128,128,128);
    }
    //...
}
}
// ...

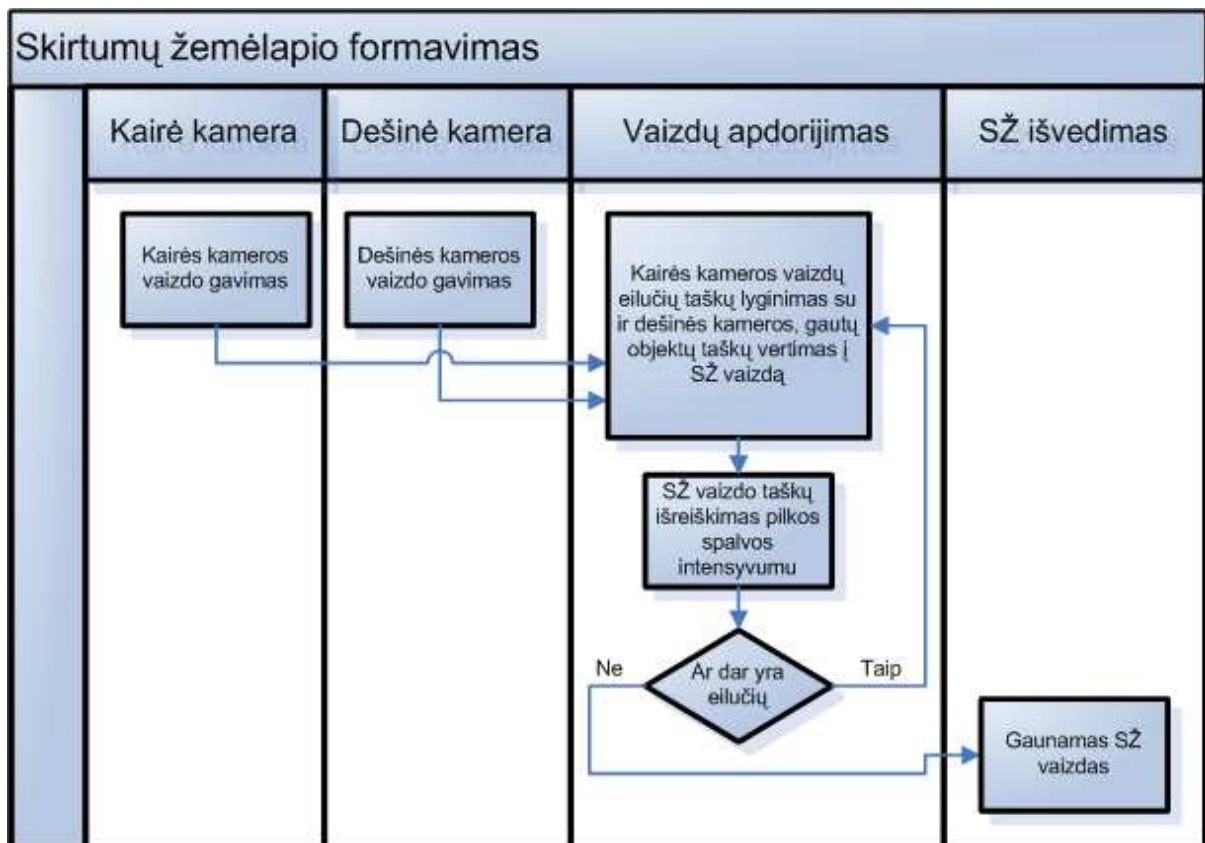
```

Algoritme reikia padaryti, kad nebūtų galima pasirinkti kito taško, esančio ne šalia toje panašioje vaizdo epipoliarinėje tiesėje (16 pav.). Dažnai atsitinka taip, kad aplinkoje esančių tekstūrų ar objektų paviršiai, ar detalės pasikartoja ir jei yra toks pat apšvietimas ar spalvos intensyvumas, algoritmas paprasčiausiai gali parinkti ne tą kairiojo paveiksluko  $I_k$  atitinkantį tašką dešiniajame paveiksliuke  $I_d$ . Tam naudojamas arčiausiojo kaimyno nustatymo metodas, kur paimami iš šonų esantys kaimynai ir bandoma juos lyginti. Tačiau toks poslinkis tikrinamas iki 50 pikselių poslinkio skirtumo ir jei viršija paprasčiausiai ignoruojamas ir laikoma, kad jo spalva yra tokia pat kaip abiejų iš šonų esančių kaimynų aritmetinis vidurkis.



16 pav. Vaizdo kalibravimas kai yra pasikartojantys vaizdo fragmentai

Sujungus visas SŽ formavimo ir pirminio apdorojimo funkcionalumo dalims gaunama tokia supaprastinta principinė SŽ vaizdo formavimo algoritmo schema (pavaizduota 17 paveiksliuke):



17 pav. Supaprastinta SŽ formavimo schema

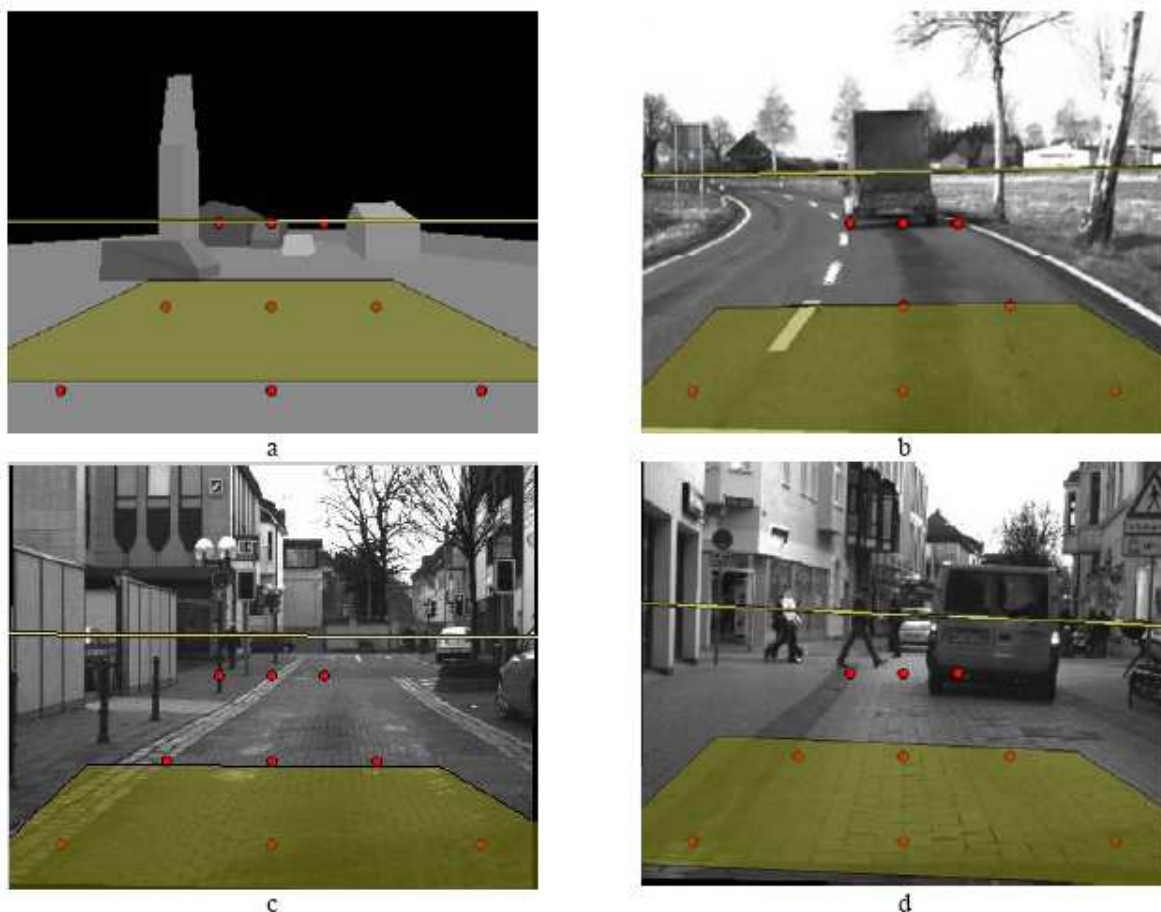
### 3.2 Skirtumų žemėlapiu pirminis apdorojimas

SŽ pirminis apdorojimas yra naudojamas nereikalingo kelio dalies vaizdo ir iškraipytų duomenų pašalinimui ir kelyje esančio objekto ar objektų išskyrimui į segmentą, kurį reikės paduoti į dirbtinio neuroninio tinklo įėjimą. Objektų atpažinimui geriausia pasirinkti atpažinimą pagal formą, nes tai yra viena iš savybių, kuri leidžia lengviausiai identifikuoti objektus. Atpažinimui pagal formas vykdyti reikia išskirti objektų formas, todėl naudojamas objektų kraštų filtravimas

SŽ vaizdo išskyrimui į regionus, kuriuose bus pašalinta perteklinė informacija ir išskirti objektai, turi būti atlikti šitie veiksmai:

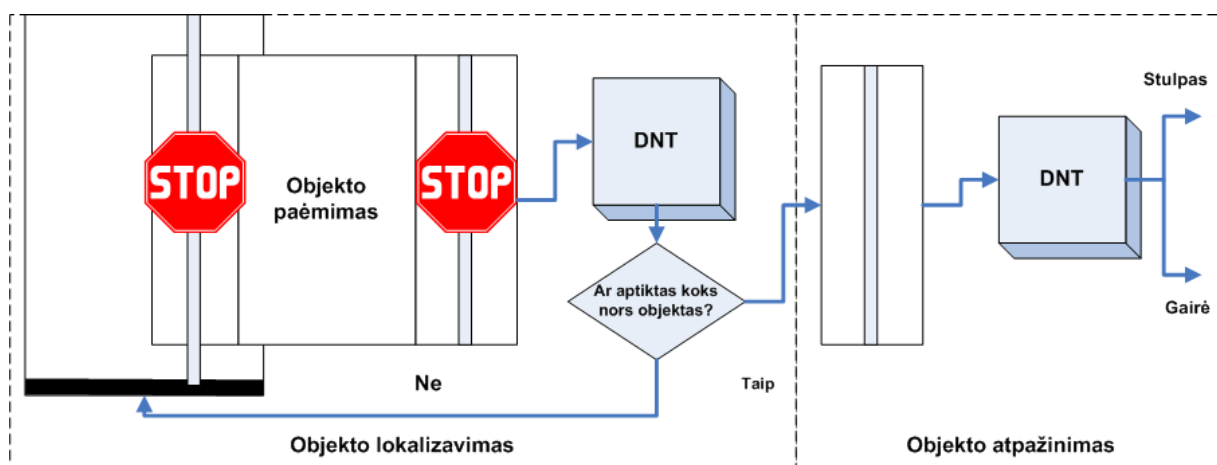
- Pašalinti dalį apatinės ir didesniąją dalį viršutinės vaizdo dalies (danguje nebus kliūčių), nes jose nėra kelio, o šis veiksmas sumažins SŽ vaizdo dydį ir DNT galės greičiau atpažinti objektus jei jie bus kelyje (18 pav.);
- Prieš paduodant SŽ vaizdą į dirbtinį neuroninį tinklą atlikti pirminį objekto aptikimą pagal pilkos spalvos intensyvumą ir gautą vaizdą sumažinti iki reikiamo dydžio regiono;
- Objektus aptikti pagal objektų kontūrus, tada į segmentą patalpinti aptiktą objektą ar objektus ir perduoti dirbtiniam neuroniniam tinklui klasifikuoti.





18 pav. Vaizdo dalys: a) virtuali; b) reali su kliūtimi; c) reali mieste, kur nėra kliūčių; d) reali mieste su kliūtimis

Taip pat SŽ vaizdo išskyrimas į segmentą su norimu identifikuoti objektu duoda didesnę tikimybę, kad objektas nebus pražiopsotas (19 pav.).



19 pav. Objekto aptikimas ir atpažinimas

### 3.2.1 Papildomas filtravimas

SŽ praleidus per kraštų aptikimo filtrą, iš SŽ vaizdo pašalinama likusi perteklinė informacija ir lieka tik objektų kraštai, pagal kuriuos DNT ir turi identifikuoti kliūtis.

Kraštų filtrui naudojama kompozicinė matrica (angl., *Convolution Matrix*), kuri SŽ vaizdą, kaip pikselių žemėlapi, interpretuoja kaip pikselių matrica ir kiekvieną jos elementą sudaugina su filtravimo matrica [G06].

Tokiose matricose neigiamų narių išdėstymas duoda tokį efektą: viršutiniai ar apatiniai elementai duoda objekto viršutinių ir apatinių kontūrų išskyrimą, kairiojo ir dešiniojo šonų išdėstymas duoda objektų kairiojo ir dešiniojo kontūrų išskyrimą.

Priklausomai nuo matricos dydžio atitinkamas resursų sunaudojimas – kuo didesnė matrica, tuo daugiau resursų sunaudojama ir atvirkščiai, todėl naudojama 3x3 dydžio matrica. Filtravimo matrica geriausiai atskiria kraštus, kai centrinis narys lygus visų aplinkinių narių sumai, pavyzdžiui:

$$\begin{bmatrix} -1 & -1 & 0 \\ -1 & 6 & -1 \\ 0 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Atlikus filtravimą, vaizdas tampa juodai baltos spalvos, kuriame šviesiomis spalvomis atvaizduojami objektų kontūrai (20 pav.).



20 pav. Vaizdo filtravimas: originalus vaizdas ir po kraštų aptikimo filtravimo

### 3.3 Dirbtinis neuroninis tinklas

DNT yra naudojamas objektų klasifikavimo uždavinio sprendimui, kaip universalus klasifikatorius triukšmingiems duomenims atpažinti. Į dirbtinio neuroninio tinklo įėjimus paduodamas SŽ vaizdo segmentas, kuriame reikia identifikuoti objektą yra kelyje.

DNT yra sukurtas 100x100 pikselių dydžio vaizdo fragmentams apdoroti, kuriuose yra norimas klasifikuoti objektas, todėl įėjimų skaičius lygus:  $100 \times 100 = 10000$ . Paduodamas vaizdas jau yra juodai baltos spalvos, todėl idealiai tinka dirbtiniam neuroniniam tinklui, nes turi nedaug perteklinės informacijos, lyginant su spalvotu vaizdu. Reikia nepamiršti, kad SŽ pirminio

apdoravimo rezultatas gražina beveik visiškai pašalintą vaizdo informaciją išskyrus objektų kontūrus.

Kai kurie autoriai iš literatūros šaltinio nurodė savus viso, dalinio ar pirminio vaizdo apdoravimo, spalvų intensyvumo sumažinto ar paversto juodai balto vaizdo padavimo į dirbtinį neuroninį tinklą būdus. Tačiau netinka viso ir dalinio vaizdo padavimas, nes:

- DNT turės apdoroti didelį kiekį duomenų;
- DNT gali ir neatpažinti dalinio objekto vaizdo išskirto į regioną;
- DNT turi veikti greitai, kad veiktų realiu laiku;
- DNT klasifikavimui reikia laiko, kuris priklauso ir nuo duomenų kiekio, todėl kuo mažiau duomenų, tuo greičiau vykdomas DNT darbas.

### 3.3.1 DNT struktūros parinkimas

Kliūtims kelyje identifikuoti tinkamas daugiasluoksnis DNT, nes vienasluoksnis DNT negali susidoroti su ne tiesiniu uždaviniu. Labiausiai tinkama yra *FeedForward* DNT struktūra, sudaryta iš kelių sluoksnių perceptronų. Remiantis apžvelgta mokslininkų patirtimi, geriausiai šiam tikslui tinka *BackPropagation* tipo dirbtiniai neuroniniai tinklai, kurie mokosi su mokytojo pagalba (angl., *supervised*) ir mokymų bei testavimų imtys jiems jau būna paruoštos.

Objektams klasifikuoti DNT reikia  $n \times m$  kiekio įėjimų, kur  $n$  – yra SŽ paveiksliuko regiono plotis, o  $m$  – yra SŽ paveiksliuko regiono aukštis pikseliais. DNT užtenka 1-o paslėpto sluoksnio, nes reikalingas greitis.

Naudojama sigmoidinė aktyvavimo funkcija, kurios išvestinė yra patogi skaičiavimams, „nešauna taip greitai“ kaip kitos ir nėra tokia „statiškai pastovi“:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

### 3.3.2 DNT mokymas ir testavimas

Mokymui naudojami kelio vaizdai su kliūtėmis, kurios gali kelti pavojų ir kurios nekelia jokio pavojaus. Mokymui parenkamos paveiksliukų sekos, sukurtos naudojant SŽ formavimo ir pirminio apdoravimo filtravimo modulį.

DNT pradeda mokytis su pradiniais svoriais iš intervalo  $[-0.5; +0.5]$ , nes su kitomis reikšmėmis veikia blogiau.

DNT mokymui atpažinti kliūtis naudojama imtis, kurioje yra tokių vaizdų:

- Vaizdų su pašaliniais objektais;
- Nepilnų vaizdų;

- Iškraipytų vaizdų;
- Vaizdų su dideliu triukšmu.

Skirtumas tarp mokymo ir testavimo imčių yra tame, kad testavimo imtyje yra daugiau iškraipytų, daugiau nepilnų, ir daugiau vaizdų su pašaliniais objektais.

Mokymai tęsiasi tol kol pastebima, kad DNT pateko į lokalųjį minimumą arba tol, kol DNT nebesimoko. DNT mokomas su idealaus, triukšmingo, dalinio ir kitų variantų SŽ vaizdo fragmentais, kuriuose yra norimi aptikti objektai. Po to paduodami norimi klasifikuoti SŽ esantys objektai iškraipyti triukšmo. Tokiu būdu stebimas DNT mokymosi procesas. Gauti mokymosi ir atpažinimo rezultatai fiksuojami, kad vėliau būtų galima juos palyginti su kitokio mokymosi rezultatais. Todėl optimaliam DNT mokymuisi ir testavimo rezultatui pasiekti reikia išbandyti šias mokymosi ir testavimo metodikas:

- Idealių pilnų SŽ vaizdų imtis mokymuisi ir bandymams imant lyginius ir nelyginius imčių objektus mokymuisi ir testavimui atitinkamai;
- Idealių dalinių SŽ vaizdų imtis mokymuisi ir bandymams, imant lyginius ir nelyginius imčių objektus mokymuisi ir testavimui atitinkamai;
- Triukšmingų pilnų SŽ vaizdų imtis mokymuisi ir bandymams imant lyginius ir nelyginius imčių objektus mokymuisi ir testavimui atitinkamai;
- Triukšmingų dalinių SŽ vaizdų imtis mokymuisi ir bandymams imant lyginius ir nelyginius imčių objektus mokymuisi ir testavimui atitinkamai.

Optimaliam rezultatui gauti, mokymosi ir testavimo imtis bus galima parinkti ir kitaip, atsitiktiniu ar santykinu dydžiu, tačiau kaip rodo literatūroje nagrinėtų autorių patirtis, mokymo ir testavimo imtis realiai nesudaro labai didelio skirtumo, išskyrus, tuos atvejus kai mokymosi ir testavimo imtis yra labai maža – tada maža patirtis gali lemti ženklų skirtumą mokymo ir testavimo imčių parinkime.

Mokymui panaudoti viename kataloge sudėti 100x100 pikselių paveikslukai, kuriuose yra pilnas, nepilnas, su triukšmu arba kitų objektų pridengti objektai – kliūtys.

### 3.4 SŽ ir DNT sujungtas modelis

SŽ sujungimas su DNT įgyvendinamas paduodant SŽ vaizdo regioną su jame esančiu objektu į DNT įėjimą. Todėl SŽ vaizdo padavimas DNT į įėjimą atrodo taip: kiekvienas SŽ vaizdo regiono (dimensija  $n \times m$  pikselių)  $n \times m$  pikselis yra paduodamas į dirbtinio neuroninio

tinklo įėjimus. Kadangi SŽ vaizdo regiono dydis yra  $n \times m$  pikselių ir yra žinoma matematinė išraiška, tai objektams klasifikuoti reikalingas toks DNT:

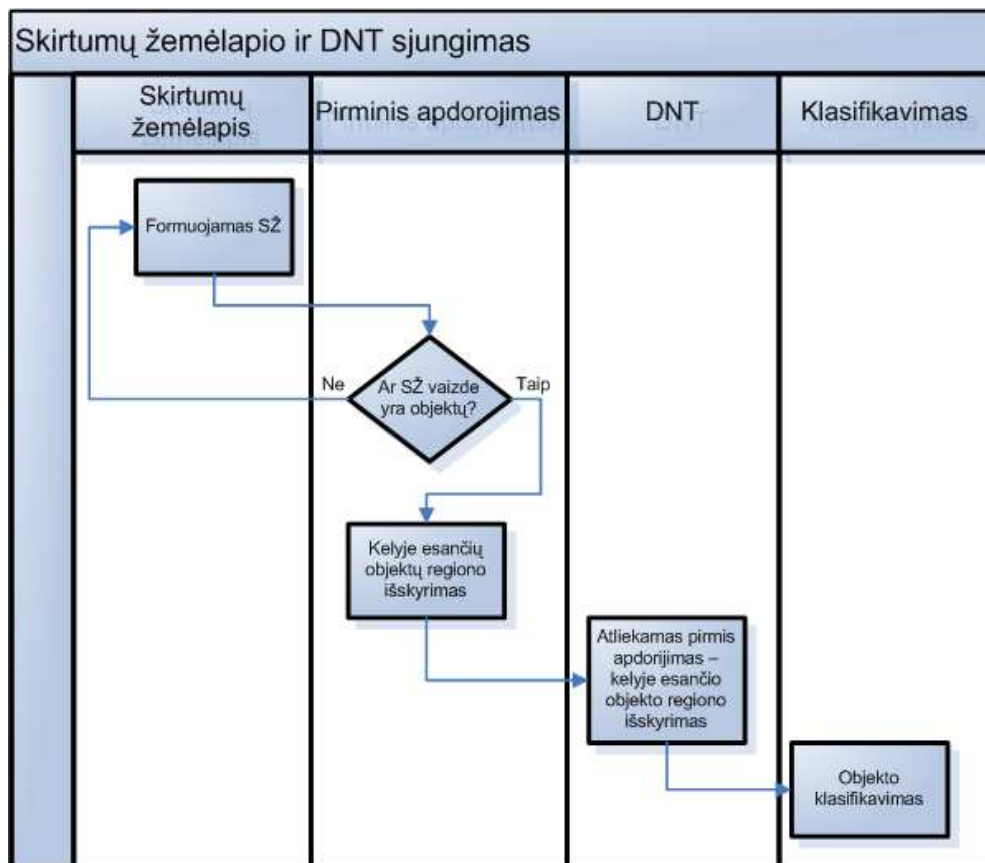
$$y_k(x+1) = \gamma_k(s_k(x)) = \gamma_k\left(\sum_j w_{jk}(x)y_j(x) + \theta_k(x)\right) \quad (6),$$

kur:  $w_i$  - svorio koeficientas,  $t$  - įėjimo koeficientas,  $k \in 0,1,2,\dots,n$ ;

SŽ vaizdo regiono sudaro  $n$  kiekis vektorių  $\vec{m}_i$ , tokių, kad

$$\vec{m} = \{m_0, m_1, m_2, \dots, m_m\}: n \times m \quad (7)$$

Sujungtų DNT ir SŽ vaizdo modelis atrodo kaip pavaizduotas paveikslėlyje (11 pav.). Tai yra supaprastinta struktūrinė schema.



21 pav. Objekto identifikavimo modelis

SŽ vaizdo regiono dydis gali būti per didelis, todėl SŽ pirminio apdorojimo mechanizmas specialiai pakeis dydį, tinkamą DNT klasifikuoti kliūtis.

SŽ vaizdo regionas dirbtiniam neuroniniam tinklui perduodamas į jėjimą kaip skaitinė informacija (22-23 pav.), išreikšta atitinkamai:

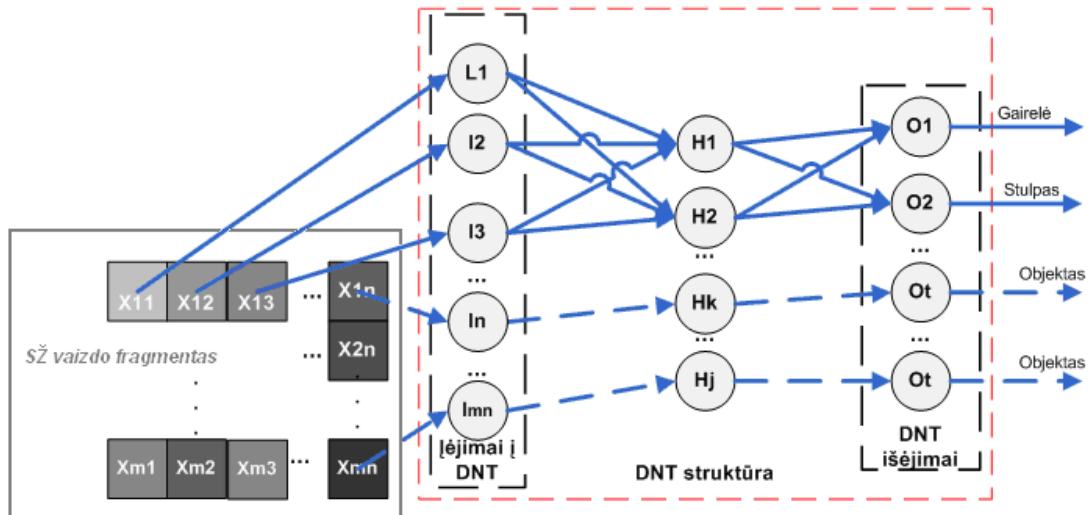
Juoda spalva – 0;

Tamsiai pilka spalva – 0,25;

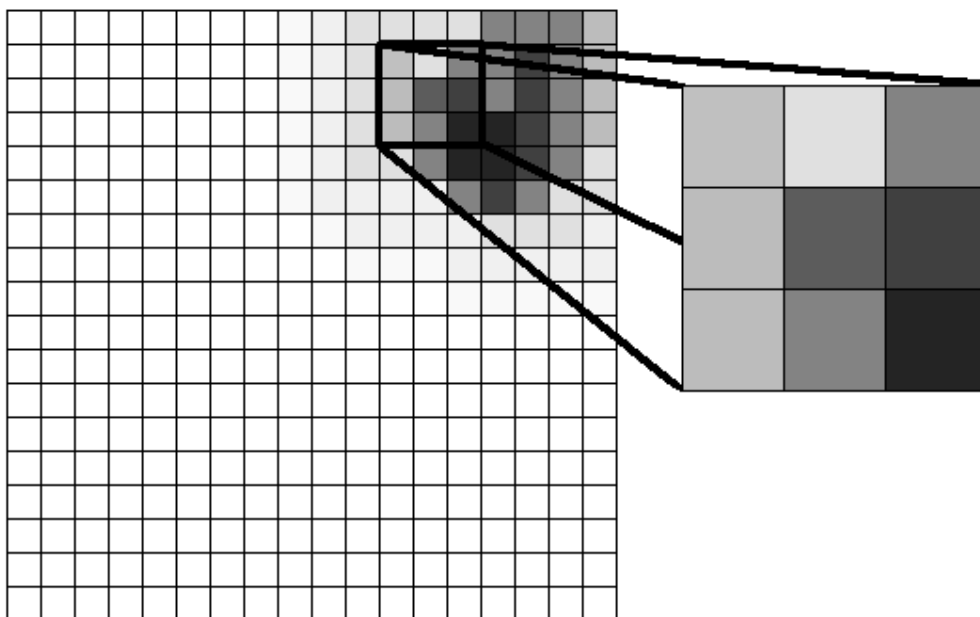
Pilka spalva – 0,5;

Šviesiai pilka spalva – 0,75;

Balta spalva – 1.



22 pav. SŽ vaizdo padavimo į DNT įėjimus schema



23 pav. Vaizdo išskyrimas į pikselius (pikselių masyvas 3x3)

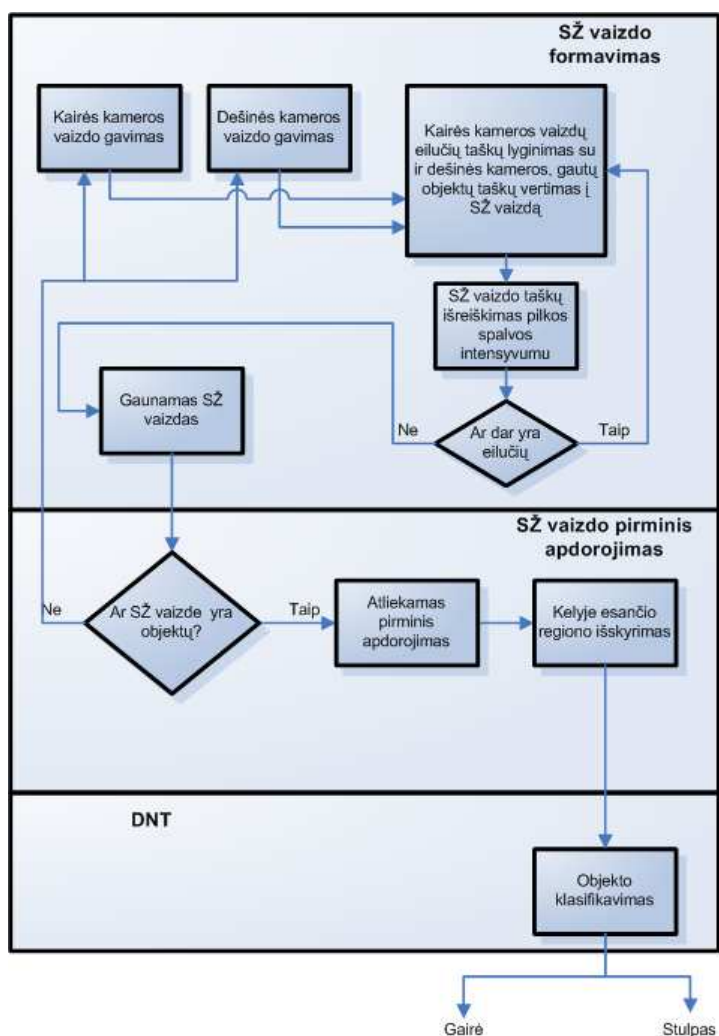
### 3.5 Galutinis sistemos vaizdas

Kliūčių atpažinimo sistemą sudaro: SŽ vaizdo formavimas, pirminis SŽ vaizdo apdorojimas, apdoroto SŽ vaizdo regiono su jame esančiais objektais perdavimą į DNT, kliūčių identifikavimas.

Siūlomos kliūčių atpažinimo sistemos struktūriniai moduliai:

- SŽ vaizdo formavimo modulis – formuos SŽ vaizdą iš dviejų kamerų gautų vaizdų išreikšdamas SŽ vaizde esamų objektų atstumą iki iki kameros pilkos spalvos ryškumu;
- SŽ vaizdo pirminio apdorojimo modulis – SŽ vaizdo apatinis kraštas ir didesnė viršutinė SŽ vaizdo dalis bus pašalinta (paversta juoda spalva) ir išskirtas regionas su objektais esančiais kelyje, t.y. SŽ vaizde;
- SŽ vaizdo regione esančių objektų klasifikavimas naudojant dirbtiniu neuroninius tinklus – DNT kelyje atpažins ar tarp objektų yra apibrėžti objektai.

Tikslesnė objektų klasifikavimo principinė schema pavaizduota 24 paveiksliuke.



24 pav. SŽ vaizdo formavimo, SŽ vaizdo pirminio apdorojimo, išskirto regiono perdavimo į DNT ir klasifikavimo schema

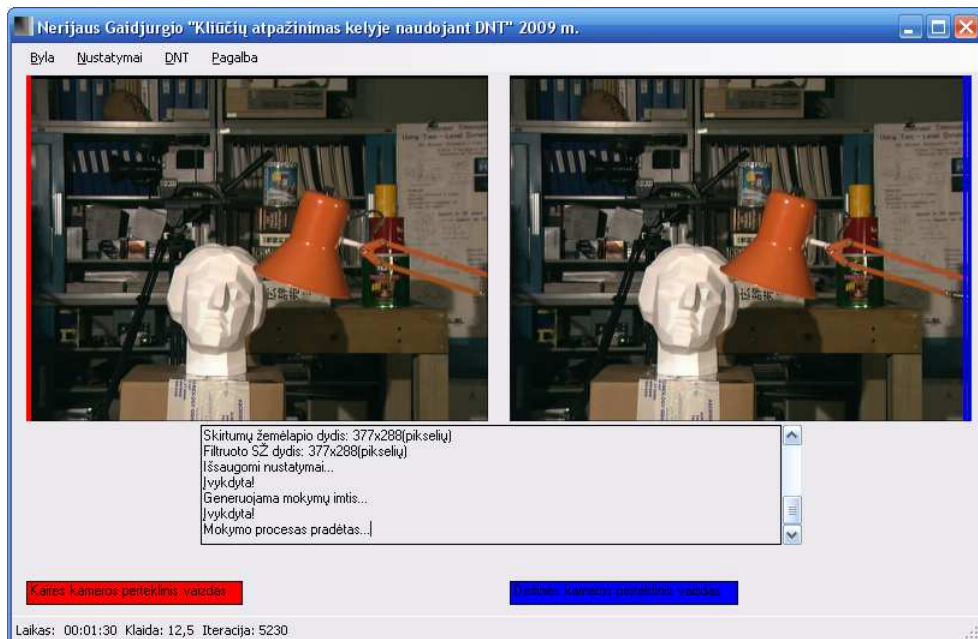
### 3.6 Darbo realizacija

Praktinei darbo daliai įgyvendinti pasirinktas Microsoft .NET 3.5 karkasas. DNT realizacijai naudojamas (atvirojo kodo) dirbtinių neuroninių tinklų variklis, SŽ formavimui pasirinkta Microsoft .NET karkaso technologija Windows FORMS. Sistemos modulių funkcionalumas įgyvendintas naudojant C# kalbą. Sistemos veikimui reikalinga mokymosi skirtų paveikslukų aibė. Vietą kur yra pastarieji paveikslukai galima konfigūruoti.

Sistemas buvo testuojama naudojant 1,6GHz Intel Core Duo procesorių, 1GB darbinės atminties kompiuterį, kuriame veikė Windows XP Professional SP2 operacinė sistema. Stereo poroms buvo naudojami internete rasti scenų paveikslukai. Žemiau pateiktuose paveikslukuose galima pamatyti sistemos veikimo darbo rezultatus. Kelio vaizdo atpažinimui panaudoti stereo paveikslukai, kuriuose yra pastatyta galva ir šviestuvai, o už jų pašaliniai objektais, kadangi realius stereo paveikslukus labai sudėtinga gauti. Panaudoti paveikslukai matomi (25 pav.). SŽ formavimui ir filtravimui panaudotas tiesioginis priėjimo prie atminties metodas, nes dirbant su paveikslukais .NET framework standartiniai metodai veikia labai lėtai. Todėl norėdami greitesnio būdo reikia optimalaus kodo.

#### 3.6.1 SŽ formavimas

SŽ formavimui naudojami 2 tos pačios scenos paveikslukai. Paveikslukai gali būti šiek tiek pasislinkę vienas kito atžvilgiu, programa pati paskaičiuoja per kiek yra pasislinkęs paveikslukas kito atžvilgiu ir sukalibruoja SŽ paveikslukų pikselių pozicijas.

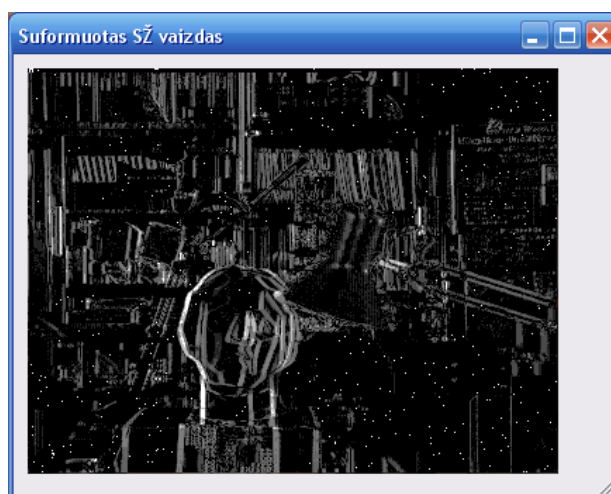


25 pav. SŽ vaizdo formavimui naudojami tos pačios scenos vaizdai



Sukalibravusi vaizdą nematomas dalis abiejuose paveikslukuose pažymi raudona (kairėje) ir mėlyna (dešinėje) spalva (25 pav.). Po šio žingsnio paveikslukai nuskaitomi į darbinę atmintį, kur panaudojami SŽ formavimui, filtravimui ir klasifikavimui naudojant dirbtinius neuroninius tinklus.

Nuskaityti duomenys iškart yra apdorojami ir pradedamas formuoti skirtumų žemėlapis. Pirmojo analizavimo iteracijos metu ieškoma sutampančių taškų (arba labai artimų) ir jie pažymimi juoda spalva. Kitos iteracijos metu ieškomi šiek tiek pasislinkę taškai ir pažymimi šviesesne spalva. Trečiosios iteracijos metu ieškomi dar labiau nutolę nei antrosios ir pažymimi dar šviesesne spalva. Galiausiai balta spalva pažymimi taškai, kurie visiškai nesutampa, arba ten, kur vienoje kameroje aptinkamas objektas, kurio nėra kitoje (26 pav.).



26 pav. Suformuotas SŽ kuriame yra triukšmo

### 3.6.2 SŽ filtravimas

Gautasis SŽ vaizdas yra perduodamas kitai apdorojimo stadijai – pirminiam apdorojimui ir filtravimui. Pirminis apdorojimas SŽ vaizde išskiria objektų kontūrus ir sumažina „išsišokusius kontūrus“. Išskirtus objektų kontūrus (27 pav.) galima pamatyti paveiksluke.



27 pav. Atfiltruotas SŽ vaizdas su objektu kraštų išskyrimu

Po šio veiksmo darbas su grafine informacija, kaip su vaizdine baigiamas ir pereinama prie vaizdo duomenų perdavimo į DNT.

### 3.6.3 SŽ segmento perdavimas DNT

Vaizdo duomenų perdavimas vyksta panaudojant pirminio apdorojimo metu gautą regioną. Regiono paveiksluko pikseliai paduodami į DNT įėjimus. Kadangi naudojamas vaizdo santykį išlaikantis mechanizmas, tai paveikslukas nėra apkarpomamas, nors DNT gali priimti tik 100x100 pikselių dydžio vaizdą (28 pav.).



28 pav. SŽ vaizdo segmentai

### 3.6.4 DNT klasifikavimas

Po mokymų ir testavimų DNT išmoksta atpažinti gautuose paveikslukuose matomas kliūtis (29 pav.) ir apie taip praneša išvesdamas sistemos lange.



29 pav. Atpažinta kliūtis labai triukšminguose duomenyse kai mokymų imtis yra 3 paveikslukai




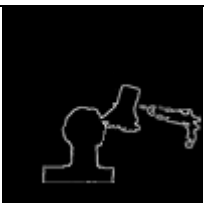


30 pav. Atpažinta gairė

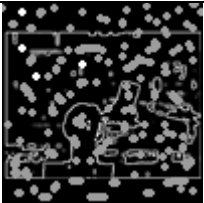
### 3.6.5 Atlikti bandymai atpažinti kliūtis



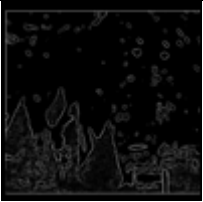

Atlikti bandymai atpažinti kliūtis, naudojant kelio vaizde esančią stabo galvą ir šviestuvą su fone esančiu triukšmu, kai mokymo imtyje naudojami tokie pat vaizdai tik be triukšmo.

*Lentelė 1: Bandymai su vienasluoksniu perceptronu*




Dirbtiniam neuroniniam tinklui paduotas vaizdas	Ar atpažino	Mokymosi imtis	Iteracijų skaičius	Atpažinimo Klaida	Mokymosi trukmė
	taip	5	10	0,23	1s
	taip	20	24	0,5	5s
	taip	60	139	0,8	12s
	taip	100	367	1,8	47s


*Lentelė 2: Bandymai su vienasluoksniu perceptronu ir triukšmingais duomenimis*

Dirbtiniam neuroniniam tinklui paduotas vaizdas	Ar atpažino	Mokymosi imtis	Iteracijų skaičius	Atpažinimo Klaida	Mokymosi trukmė
	taip	50	403	2,47	47s

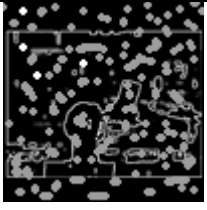


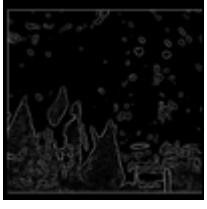

	taip	50	374	1,2	38s
	taip	50	176	0,8	12s
	ne	50	537	3,4	73s
	ne	50	467	3,21	54s

*Lentelė 3: Bandymai su daugiasluoksniu perceptronu*

Dirbtiniam neuroniniam tinklui paduotas vaizdas	Ar atpažino	Sluoksnių skaičius/neuronų skaičius sluoksnyje	Mokymosi imtis	Iteracijų skaičius	Atpažinimo Klaida	Mokymosi trukmė
	ne	2/300	5	1000	>10	357s
	ne	2/300	20	1000	>10	804s
	ne	2/300	60	1000	>10	1398s

	ne	2/300	100	1000	>10	3833s
---	----	-------	-----	------	-----	-------

Lentelė 4: Bandymai su daugialuoksniu perceptronu ir triukšmingais duomenimis

Dirbtiniam neuroniniam tinklui paduotas vaizdas	Ar atpažino	Sluoksnių skaičius/neuronų skaičius sluoksnyje	Mokymosi imtis	Iteracijų skaičius	Atpažinimo Klaida	Mokymosi trukmė
	ne	2/300	50	1000	>10	47s
	taip	2/300	50	1000	3	38s
	ne	2/300	50	1000	>10	12s
	ne	2/300	50	1000	>10	73s
	ne	2/300	50	1000	>10	54s

Peržvelgus atpažinimo rezultatus matosi, kad geriausi rezultatai būna tada, kai yra naudojamas vienasluoksnis perceptronas. Į dirbtinį neuroninį tinklą įdėjus paslėptą sluoksnį jis nebesimoko greitai ir jį reikia labai ilgai mokinti ir dažnai gaunasi taip, kad labiau išmokyti nepavyksta. Taip pat matome, kad padidinus mokymų imties kiekį, VSP sunkiau atpažįsta

objektus. Todėl galima teigti, kad DNT veikia su nedidelia mokymų imtimi, kai imtis neviršija 100 paveikslukų.

#### **4. Rezultatai ir išvados**

1. Sukurtas SŽ formavimo algoritmas, kuris yra greitas ir tinkamas SŽ vaizdo formavimui realiu laiku. SŽ formuojamas naudojant tos pačios scenos dvi kameras, kurios yra lygiagrečios viena kitai. SŽ formavimas pagrįstas epipoliariniu apribojimu, kuriuo remiantis užtenka pikselių paiešką atlikti vienmatėje erdvėje;
2. Realizuotas pirminis apdorojimas ir filtravimas pagal objektų kontūrus. Pirminis apdorojimas atlieka dalį DNT darbo – atlieka pirminį objektų išskyrimą pagal formą. Taip pat, pirminis apdorojimas išskiria aptiktus objektus į regionus, kurių pakanka DNT kliūčių atpažinimui;
3. DNT atpažįsta kliūtis kelyje. DNT mokymuisi ir testavimui naudoja juodai baltos spalvos SŽ vaizdo regionus su išskirtais objektų kontūrais. Geriausi dirbtinio neuroninio tinklo rezultatai pasiekti naudojant 1 sluoksnio tinklą ir mokymosi imtį, neviršijančią 100 paveikslukų. Dirbtinio neuroninio tinklo darbo sparta pakankama, kad sistemą būtų galima naudoti realiu laiku.

## 5. Informacijos šaltinių sąrašas

1. **[H05] Simon Haykin - Neural Networks - A Comprehensive Foundation – Second Edition, Pearson Education, McMaster University, Hamilton, Ontario, Canada, 2005**
2. **[K01] Роберт Каллан, Основные концепции нейронных сетей, Саутгемптонский институт, Издательский дом “Вильямс”, 2001**
3. **[G01] Mani Ghasemlou – 308-558B: Fundamentals of Computer Vision Marr & Poggio’s Cooperative Stereopsis Algorithm: An Implementation In C, 2001**
4. **[RH04] Donald Rosselot and Ernest L. Hall, Processing real-time stereo video for an autonomous robot using disparity maps and sensor fusion, Center for Robotics Research, 3-7 puslapiai, Department of Mechanical, Industrial, and Nuclear Engineering, University of Cincinnati, 2004**
5. **[BT98] Stan Birchfield and Carlo Tomasi, Depth Discontinuities by Pixel-to-Pixel Stereo, Computer Science Department, Stanford University, 1998**
6. **[N96] Claus Neubauer, Recognition of Handwritten Digits and Human Faces by Convolutional Neural Network, International Computer Science Institute, Berkeley, California, December 1996;**
7. **[SS03] Daniel Scharstein and Richard Szeliski, High-Accuracy Stereo Depth Maps Using Structured Light – In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, puslapiai 195–202, Madison, WI, 2003**
8. **[ZMJ03] Qing Zheng, Bruce K. Milthorpe, and Allan S. Jones – Direct Neural Network Application for Automated Cell Recognition, Graduate School of Biomedical Engineering, University of New South Wales, Sydney, Australia, 4 August 2003**

9. **[D07] DARPA (Defense Advanced Research Projects Agency) puslapis, apie projektą, skirtą sukurti DNT valdomą transporto priemonę, 2007, 164 KB - URL: <http://www.darpa.mil>**
  
10. **[W00] Dane Walsh – Face Recognition: A Combination of Various Techniques – Honours Project Department of Computer Science University of Stellenbosch – South Africa – 12th February 2000**
  
11. **[SS01] Daniel Scharstein, Richard Szeliski – A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms, puslapiai 19, 24-32, Dept. of Math and Computer Science, Middlebury College, Middlebury, VT 05753 and Microsoft Research, Microsoft Corporation, Redmond, WA 98052, 2001**
  
12. **[BN04] Stanley T. Birchfield and Braga Natarajan – Correspondence as Energy-based Segmentation, puslapiai 13-14 puslapiai, Clemson University, Clemson, SC 29634, 2004**
  
13. **[A04] Shashank Araokar, Visual Character Recognition using Artificial Neural Networks, MGM's College of Engineering and Technology, University of Mumbai, India, 2004**
  
14. **[B03] Sven Behnke, Hierarchical Neural Networks for Image Interpretation, 7-45 puslapiai, June 13, 2003**
  
15. **[B04] Anna Bartkowiak, NEURAL NETWORKS and PATTERN RECOGNITION, Institute of Computer Science, University of Wrocław, Fall 2004**
  
16. **[J98] Shen Jianqing, Graphical Item Recognition Using Neural Networks, Document Analysis and Technologies Research Group, Università degli Studi di Firenze, Italy, 1998**



17. **[H06]** Jeroen Hol, Thomas Ericsson, Intermediate View Interpolation Of Stereoscopic Images For 3-d Display, Linköping, Sweden 2006
18. **[D02]** Daniel Scharstein, Richard Szeliski, A taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms, International Journal of Computer Vision, vol. 47, no. 1-3, pp. 7-42, 2002
19. **[P99]** Petri Hodju, Jokko Halme, puslapis apie DNT, jų struktūrą ir topologiją, 1999, 11,18 KB – URL:  
<http://koti.mbnet.fi/~phodju/nenet/NeuralNetworks/NeuralNetworks.html>
20. **[ ]** NeuronDotNet puslapis, kuriame yra .NET sistemos klasės ir bibliotekos DNT realizavimui, 2009, 3,27KB – URL:  
<http://neurondotnet.freehostia.com/index.html>
21. **[N08]** Nikolay Chumerin, Marc M. Van Hulle, Ground Plane Estimation Based on Dense Stereo Disparity, Katholieke Universiteit Leuven, Laboratorium voor Neuro- en Psychofysiologie, Herestraat 49, B-3000 Leuven, Belgium, pp27-30, 2008
22. **[G06]** GIMP User Manual, 7.2 Convolution Matrix, 2006 11 29, 36,05 KB – URL:  
[http://www.linuxtopia.org/online\\_books/graphics\\_tools/gimp\\_user\\_manual/en/plugin-in-convmatrix.html](http://www.linuxtopia.org/online_books/graphics_tools/gimp_user_manual/en/plugin-in-convmatrix.html)
23. **[M04]** Michiel W.H. Remme, Wytse J. Wadman, Control of a local neural network by feedforward and feedback inhibition, Elsevire B.V. 2004
24. **[J99]** Jitendra Malik, Computer Vision, Oct. 25, 1999
25. **[N07]** Nicolas Soquet, Didier Aubert, Road Segmentation Supervised by an Extended V-Disparity Algorithm for Autonomous Navigation, IEEE Intelligent Vehicles Symposium, 2007

**26. [D04] Donald Rosselot and Ernest L. Hall, Center for Robotics Research,  
Department of Mechanical, Industrial, and Nuclear Engineering, University of  
Cincinnati, 2004**