

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
MULTIMEDIJOS INŽINERIJOS KATEDRA

Jurgita Genytė

## **Rastrinių vaizdų vektorizavimo sistema**

Magistro darbas

Darbo vadovas

doc. dr. A. Lenkevičius

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
MULTIMEDIJOS INŽINERIJOS KATEDRA

Jurgita Genytė

**Rastrinių vaizdų vektorizavimo sistema**

Magistro darbas

Kalbos konsultantas

Lietuvių k. katedros lekt.

J. Jonušas

2006-04-20

Vadovas

doc. dr. A. Lenkevičius

2006-05-20

Recenzentas

doc. V. Jusas

2006-05-25

Atliko

IFM – 0/3 gr. stud.

Jurgita Genytė

2006-05-20

Kaunas, 2006

## Turinys

1. Įvadas .....	4
2. Analitinė dalis .....	5
2.1. Nagrinėjamos srities analizė.....	5
2.2. Panašių produktų rinkoje analizės rezultatai.....	7
2.3. Egzistuojančių vektorizavimo algoritmų analizė .....	9
2.3.1. Linijų sekos procedūros .....	10
2.3.2. Vektorizavimo metodų palyginimas .....	13
2.3.3. SPV esmė .....	15
2.4. Rastrinio vaizdo vektorizavimo sistemos įgyvendinimo etapai .....	15
2.5. Išvados .....	17
3. Projektinė dalis.....	18
3.1. Reikalavimų projektuojamai sistemai specifikacija (Volere šablonas).....	18
3.1.1. Projekto varovai .....	18
3.1.2. Projekto apribojimai.....	20
3.1.3. Funkciniai reikalavimai.....	22
3.1.4. Nefunkciniai reikalavimai .....	34
3.1.5. Projekto išėiga .....	39
3.2. Duomenų struktūra .....	42
3.2.1. Sistemos statinis vaizdas (klasių diagramos) .....	42
3.3. Projektuojamos sistemos architektūra .....	45
3.3.1. Sistemos modulinė schema.....	45
3.3.2. Sistemos dinaminis vaizdas.....	46
3.3.3. Išdėstymo (deployment) vaizdas .....	49
3.4. Programinių modulių ar objektų specifikacijos.....	50
3.4.1. Vartotojo sąsajos komponentas .....	50
3.4.2. Veiklos paslaugų komponentas .....	53
3.4.3. Duomenų struktūrų komponentas.....	55
3.4.4. Automatinio vektorizavimo algoritmas.....	57
3.5. Testavimo medžiaga .....	61
3.6. Testavimo išvados .....	62
4. Vartotojo dokumentacija.....	63
4.1. Sistemos funkcinis aprašymas.....	63
4.2. Sistemos vadovas.....	64
4.3. Sistemos instaliavimo dokumentas.....	67
4.3.1. Instaliavimo instrukcija .....	68
4.3.2. Reikalavimai techninei bei programinei įrangai.....	68
4.3.3. Sistemą sudarantys failai .....	68
5. Produkto eksperimentinis tyrimas .....	69
5.1. Tyrimas .....	69
5.1.1. Vektorizavimo rezultatų palyginimas .....	69
5.1.2. Formato (BMP, JPG, GIF) įtaka vektorizavimo rezultatams .....	71
5.1.3. Parametro <i>th</i> įtaka vektorizavimo rezultatams .....	73
5.1.4. Rezultatų failo struktūra .....	74
5.1.5. Tirtų sistemų privalumai ir trūkumai .....	75
5.2. Tyrimo išvados .....	76
6. Išvados .....	77
7. Literatūra.....	78
8. Raster images vectorization system .....	80
9. Terminų ir santrumpų žodynas.....	81

10. Priedai .....	82
1 priedas. Pagrindine.java kodas .....	82
2 priedas. Metodai.java kodas .....	86
3 priedas. Metodų saveMatrica() ir saveRez() kodai .....	89
4 priedas. Tyrimams naudoti rastriniai originalai .....	91
5 priedas. Kompaktinis diskas su programos kodu .....	96

# 1. Įvadas

Profesionaliems ir patyrusiems kompiuterinio projektavimo (angl. *Computer Aided Design (CAD)* ) vartotojams vis dažniau prireikia ranka braižytus brėžinius perkelti į kompiuterį bei redaguoti. Todėl nuskaitytas brėžinys iš pradžių turi būti vektorizuotas, nes kitaip jo nesupras CAD sistemos. Taigi, tam reikalinga gera ir tiksli vektorizavimo programinė įranga. Ji reikalinga šiose srityse: architektūroje, mechanikoje, elektrotechnikoje, kartografijoje ir pan.

Popieriniai brėžiniai laikui bėgant nusidėvi, todėl buvo užsibrėžtas tikslas juos perkelti į kompiuterį. Tačiau, kai brėžinių kiekis toks didelis, rankiniu būdu viską vektorizuoti tapo neįmanoma. Tam būtų sunaudota per daug laiko ir energijos. Todėl per paskutinius dvejus metus buvo analizuojama ir tiriama rastrinių vaizdų vektorizavimo problema. Buvo pastebėta, jog šiuo metu rinkoje nėra daug ir stiprių konkurentų šioje srityje, nes gerą rezultatą pasiekti yra pakankamai sudėtinga. Dar nėra tokios programos (algoritmo), kuri idealiai atliktų tai, ko iš jos tikimasi. Gaunami rezultatai yra tik artimi originalui, nes susiduriama su nemažai problemų, kurios yra išanalizuotos ir pateiktos šiame darbe.

Taigi šio **darbo tikslas** - sukurti rastrinių vaizdų vektorizavimo sistemą.

Tikslą pasiekti buvo iškelti tokie uždaviniai:

- išanalizuoti rastrinių vaizdų vektorizavimo probleminę sritį bei jos aktualumą;
- išanalizuoti esamus vektorizavimo algoritmus, aptarti privalumus bei trūkumus;
- suprojektuoti ir realizuoti rastrinių vaizdų vektorizavimo sistemą;
- parengti reikalingą vartotojo dokumentaciją;
- testuoti sistemą ir aptarti rezultatus;
- atlikti eksperimentinį tyrimą.

Darbą sudaro šios dalys:

- analitinė - kurioje išanalizuota nagrinėjama sritis, problema ir jos sprendimo būdai;
- projekcinė - kurioje nustatyti sistemos reikalavimai, aprašyta pati kuriama sistema, jos architektūra bei galimybės;
- vartotojo dokumentacija - kurioje pateiktos instrukcijos sistemos vartotojui;
- produkto eksperimentinis tyrimas - kuriame pateikti produkto tyrimo rezultatai bei išvados.

## 2. Analitinė dalis

### 2.1. Nagrinėjamos srities analizė

Vektorizavimas - tai procesas, kurio metu skaitmeniniai-rastriniai (angl. *bitmap*) vaizdai paverčiami vektoriniais. Vektorizavimas yra plačiai taikomas dokumentų analizės ir atpažinimo pradiniam etape. Po jo seka aukšto lygio objektų atpažinimas, pavyzdžiui, optinis simbolių ar grafinių objektų atpažinimas. Vektorizavimo proceso pagrindinė užduotis - iš rastrinio vaizdo išgauti tik reikalingus taškus (angl. *pixel*), juos sujungti į specialias grandinėles, kurios turi savus atributus (pvz. charakteringi taškai, linijos plotis).

Kaip žinome, rastriniai vaizdai sudaryti iš taškų, todėl:

- užima daug vietos kompiuterio atmintyje;
- juos sudėtinga redaguoti (didelė skaičiavimų apimtis);
- "laiptų efektas" keičiant dydį;
- sunku dirbti su tekstu;
- reikia žinoti apie skiriamąją gebą bei spalvinį gylį;
- bet kokios vaizdo transformacijos iškraipo vaizdą.

Vektorinė grafika charakterizuojama matematiniais objektais – vektoriais, kurie apibrėžia geometrines formas (tieses, apskritimus ir t.t.). Dirbant su vektorine grafika redaguojamos formos, trajektorijos bei jų grupės. Vektoriniai vaizdai aprašomi matematinėmis formulėmis, todėl:

- užima gerokai mažiau vietos diske;
- nesudėtingas redagavimas ir manipuliavimas (maža skaičiavimų apimtis);
- dydis keičiamas be kokybės nuostolių;
- didelis piešimo tikslumas, todėl geresnė ir spausdinimo kokybė;
- lengvai transformuojama į taškine grafiką (angl. *rasterization*), tai atvirkščias procesas vektorizavimui.

Vektorinė grafika turi ir trūkumų. Didžiausias iš jų - neįmanoma automatizuoti grafines informacijos įvedimo. Todėl bandoma tai ištaisyti vektorizavimo programomis.

Vektorizavimas yra sudėtingas procesas, kuriam reikia nemažų techninių bei laiko sąnaudų. Dar nėra tokios programinės įrangos, kuri pateiktą paveikslėlį preciziškai paverstų į vektorinį, nes programa nežino, kokio galutinio rezultato tikisi vartotojas. Todėl dažniausiai, kol pasiekiamas galutinis rezultatas, dabartinės vektorizavimo programos vartotojui pateikia ne vieną tarpinį rezultatą. Todėl vartotojas gali keisti tam tikrus parametrus ir pagal savo poreikius daryti įtaką vektorizavimo procesui ir galutiniam jo rezultatui.

Iš taikymo sričių reikėtų išskirti žemėlapių ir brėžinių vektorizavimą. Daugelis autorių teigia, jog ateityje bus naudojami tik skaitmeniniai žemėlapiai, nes popieriniai laikui bėgant dėvisi, prarandama juose esanti informacija. Skiriami keturi popierinio žemėlapio ar techninio brėžinio vektorizavimo būdai (3):

- vizualus perbraižymas;
- skaitmenizavimas, naudojant tam skirtus įrenginius;
- skaitmenizavimas iš ekrano;
- automatinis vektorizavimas.

Pastaruoju metu dažniausiai naudojami du paskutiniai minėti būdai.

Skaitmenizavimo iš ekrano privalumai:

- nereikia specialios įrangos, išskyrus vienkartinį nuskaitymą didelio formato skaitytuvais;
- galimybė dirbti su didesne nei skaitytuvo skiriamąja gėba;
- nereikia brangių specialių programinių paketų.

Trūkumai:

- koordinačių įvedimo paklaidos;
- sunku apdoroti dideles duomenų rinkmenas (angl. *file*);
- skaitant ir sujungiant atsiranda sujungimo klaidos.

Automatinio vektorizavimo privalumai:

- įvedamų koordinačių tikslumas;
- nereikia specialios įrangos, išskyrus vienkartinį skaitymą didelio formato skaitytuvais;
- minimalios laiko sąnaudos;
- galima apdoroti didelio formato failus.

Trūkumai:

- reikalingi brangūs specializuoti programiniai paketai;
- kai kuriose srityse gaunami klaidingi rezultatai.

Bendras abiejų būdų bruožas – tai vaizdo (žemėlapio/brėžinio) skaitymas pradiniame etape, po to rastrinio failo apdorojimas.

Visiškai atsisakyti rankinio skaitmenizavimo kompiuterio ekrane kol kas neįmanoma, nes vykdant automatinį vektorizavimą susiduriama su tokiomis problemomis (9):

- nepakankama originalo kokybė turi įtakos galutiniam rezultatui (tai popieriaus lenkimo vietos, nusitrynimai, pašaliniai simboliai, nusidėvėjimas);
- daugiareikšmės linijos (kai ta pati linija priklauso keliems objektams);

- susikertančių linijų problema (geresni rezultatai pasiekiami turint spalvotus žemėlapius, tačiau tai nepadedą, jei susikerta trys ir daugiau linijos);
- susiliečiančių linijų problema (neaiškus lietimosi taškas ir pan.);
- neištisinių linijų (taškinių, brūkšninių) bei jų galinių taškų nustatymo problema;
- specialių simbolių bei teksto atpažinimas;
- tekstinė informacija gali dengti grafinę, kurią reikia atkurti (automatiniu vektorizavimu tai išvis neįmanoma).

Taip pat prie automatinio vektorizavimo problemų būtų galima priskirti šias:

- kampų iškraipymai;
- vektorizavus dingsta pagrindiniai charakteringi taškai arba kaip tik jų atsiranda daugiau negu reikia;
- kreivės, kurios turėtų būti apvalios, yra tiesios ar atvirkščiai;
- smarkiai išstorėjančių linijų problema;
- prieš atliekant vektorizavimą dažnai tenka apriboti naudojamų rastre spalvų skaičių;
- vektorizavimo algoritmai imlūs skaičiavimams, laikui bei atminčiam.

Kartais vektorizavimas rankiniu būdu užtrunka mažiau laiko ir pasiekiamas tikslesnis norimas rezultatas. Tačiau toks vektorizavimas daugiau tinka mažiems ir nesudėtingiems darbams atlikti, pvz.: firmos ženklui (logotipui) vektorizuoti. Tačiau brėžiniams, diagramoms, žemėlapiams ir kitiems stambiams vaizdams vektorizuoti naudojamos automatinio ar pusiau automatinio vektorizavimo programos. Gautas rezultatas eksportuojamas vektoriniais formatais ir toliau gali būti redaguojamas CAD priemonėmis.

## **2.2. Panašių produktų rinkoje analizės rezultatai**

Atlikus panašių produktų rinkoje paiešką, buvo aptikta visa eilė produktų, skirtų būtent rastriniams vaizdams vektorizuoti. Būtų galima paminėti keletą iš jų: Corel TRACE, Adobe Illustrator, Acme TraceArt v3.7, Vextractor, RasterVect, Vector Eye, TraceIt, Potrace, ArcScan for ArcGIS, AutoTrace ir kt. Sunku pasakyti, kurie iš jų patys efektyviausi, kadangi tai lemia tokie veiksniai:

- originalaus (rastrinio) vaizdo savybės: vaizdo sudėtingumas, spalvų skaičius, skiriamoji geba, kontrastas;
- naudojamas vektorizavimo algoritmas (kiekvienam konkrečiam atvejui gali prireikti vis kitokio algoritmo, nes universalus nėra);



- pagalbinės rastro paruošimo priemonės (pvz. triukšmų filtrai).

Taigi, vektorizavimo procesas yra ilgas ir sudėtingas. Jis reikalauja nemažai kantrybės bei patirties. Galutinis rezultatas pasiekiamas tik perėjus visus tarpinius etapus. Dažnai kartu pateikiami ir rastro redagavimo įrankiai, skirti išvalyti rastro teritorijas ar poredaguoti rastrą prieš atliekant konvertavimą tam, kad automatizuotas vektorizavimas būtų efektyvesnis ir reikalautų kuo mažiau rankinio darbo konvertavimui pabaigti. Tokiose sistemose pagrindiniai vektorizavimo proceso etapai yra:

- rastrinio vaizdo įkėlimas (importavimas, bet tik tų formatų kuriuos palaiko programa);
- rastrinio vaizdo paruošimas vektorizuoti (triukšmų filtravimas, kontrasto didinimas, spalvų modelio keitimas ir pan.);
- automatinis vektorizavimas (pasirinkus galimus parametrus);
- vektorizuoto vaizdo redagavimas rankiniu būdu, tam skirtomis priemonėmis (ne visose yra);
- rezultato eksportavimas tam tikru vektoriniu formatu (galima pasirinkti tik iš tų, ką siūlo konkreti sistema).

Jau egzistuojančių rastrinio vaizdo vektorizavimo sistemų apdorojami rastrinių failų formatai bei eksportuojami vektoriniai failų formatai pateikti 1 lentelėje. Toje pačioje lentelėje pateiktos ir skirtingų produktų kainos (2006 m. gegužė).

1 lentelė. Failų formatai naudojami skirtingose vektorizavimo sistemose.

<b>Programinio produkto pavadinimas</b>	<b>Importuojami rastriniai formatai</b>	<b>Eksportuojami vektoriniai formatai</b>	<b>Kaina</b>
Corel TRACE X3	PPF, CPT, BMP, PCX, TGA, GIF, JPG, JP2, TIF, FPX, PP4, PSD, PP5, RIFF, PNG, CAL, PCT	AI, CGM, CMX, DWG, DXF, EMF, PCT, PLT, WMF, WPG	€ 489
Adobe Illustrator CS2			€ 718
Vector Eye	BMP, JPG, TIFF, PNG	SVG, PS, EPS	€ 49
TraceIt	BMP, GIF, TIF, JPG, PSD, WMF, PIC, PPM, CEL, PCD	ILD, LDA, LPC, LP, PLT, LDS	nemokama
RasterVect	BMP, TIFF, PCX	DXF, WMF, EMF, EPS, AI	\$ 80
Potrace	PBM, PGM, PPM, BMP	EPS, PostScript, SVG ( <i>scalable vector graphics</i> ), Xfig, Gimppath, PGM	nemokama
Vextractor	BMP, GIF, TIFF, JPEG, PCX, TGA, PNG	DXF, DXB, WMF, EMF, SHP, MIF/MID, XYZ, SVG, EPS	\$ 99
Acme TraceArt v3.7	BMP, TIFF, TGA, JPEG ir kt.	DXF, WMF, EPS, HP-GL(.PLT), SVG, EPS, PDF	\$ 80
AutoTrace	BMP, TGA, PNM, PPM, PGM, PBM	Postscript, SVG, XFIG, SWF, EMF, DXF, CGM, MIF, P2E ir kt.	nemokama

Corel TRACE priklauso Corel DRAW programų šeimai, todėl prireikus vektorizavimo priemonės tektų įsigyti visą grafikos paketą (2). Jį sudaro tokios programos: CorelDRAW® X3, Corel® PowerTRACE™ X3, Corel® PHOTO-PAINT® X3, Pixmantec® RawShooter™ Essentials, Corel CAPTURE™ X3. 2006 metų gegužės mėnesį Europoje šis produktas kainavo € 489 (t.y. apie 1688 Lt). Kitų panašių produktų kainos pateiktos 1 lentelėje.

Kokie vektorizavimo metodai bei algoritmai naudojami kiekviename produkte neskelbiama, tačiau vektorizuojant galima pasirinkti tokius parametrus:

- vektorizavimo tipas (*centerline* arba *outline*);
- slenkstinė vertė (angl. *threshold*) (nuo 0 iki 255);
- iteracija, taškų atstatymas (angl. *iteration, node reduction*) (Corel TRACE programoje);
- spalvų modelis (kai kurie vektorizavimo tipai reikalau konkretaus spalvų modelio, pavyzdžiui, su *centerline* galima vektorizuoti tik *black and white* spalvų modelio vaizdus)
- ir kt.

### 2.3. Egzistuojančių vektorizavimo algoritmų analizė

Vektorizavimu pradėta domėtis dar prieš daugiau kaip trisdešimt metų. Sukurtas ne vienas tam skirtas algoritmas. Galima būtų paminėti: vektorizavimo algoritmą pagal Voronoi L- diagramą (6) (taikomas simbolių atpažinimui), uždarų regionų vektorizavimo algoritmą (5) (efektyvus tik uždarų regionų atpažinimui) ir k.t. Taigi, buvo ieškoma būdų, kaip kuo paprasčiau ir efektyviau rastrinį vaizdą paversti į vektorinį, kad po to jį būtų galima redaguoti visai kitomis priemonėmis. Buvo pastebėta, kad vienas ar kitas vektorizavimo algoritmas efektyvus tik vienai ar kitai sričiai, tačiau niekam nepavyksta sukurti universalų, tinkamą bet kokiam atvejui. Visi literatūroje bei moksliniuose darbuose minimi vektorizavimo metodai skirstomi į šešias grupes (10):

- „Hough“ transformacija paremti metodai (angl. *Hough Transform (HT) based methods*);
- ploninimu paremti metodai (angl. *thinning based methods*);
- kontūriniai metodai (angl. *contour based methods*);
- bėgančio-grafo metodai (angl. *run-graph based methods*);
- tinklelio šablono metodai (angl. *mesh pattern based methods*);
- išsklaidytų pikselių metodai (angl. *sparse pixel based methods (SPV)*).

Išskyrus pirmą metodų grupę, tipišką vektorizavimo procesą sudaro tokie pagrindiniai etapai (10):

1. Vidurio ašies taškų atranka. Tai esminis procesas informacijai gauti. Jo metu ženkliai sumažinamas jos kiekis, nes išsaugoma informacija tik apie tuos atrinktus svarbius taškus.
2. Linijų sekos procedūros (angl. *line tracking*). Vidurio ašies taškai jungiami į vientisą tų taškų grandinę, gaunami linijų segmentai.
3. Linijos segmentų aproksimavimas arba uždarų segmentų interpretavimas daugiakampiais (angl. *polygonalization*). Šio etapo metu nekritiniai taškai pašalinami iš taškų grandinių, surastų antro etapo metu, o likę kritiniai taškai sujungiami į atkarpas bei polilijas.

Prieš tai minėtos šešios vektorizavimo metodų grupės skiriasi viena nuo kitos būtent pirmais dviem vektorizavimo proceso etapais. Trečiasis etapas gali būti arba toks pat, arba panašus.

### 2.3.1. Linijų sekos procedūros

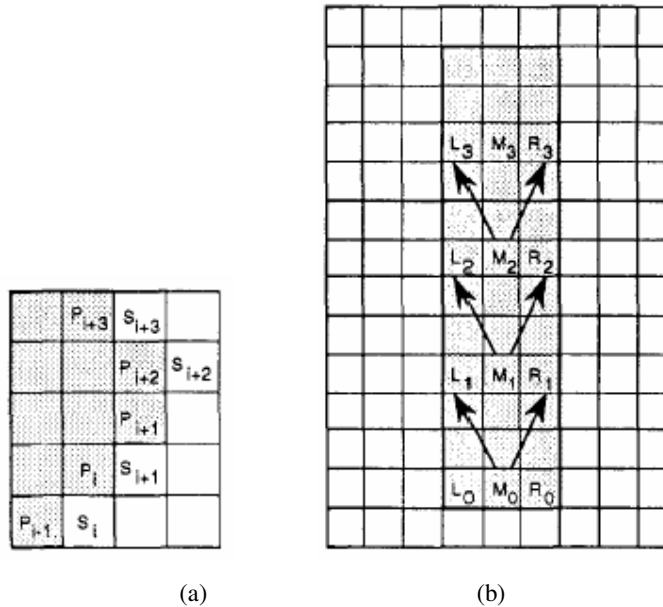
Bendras bruožas, jungiantis skaitmenizavimą iš ekrano ir automatinį vektorizavimą, yra linijos sekos procedūros. Tai autonominės paprogramės, kurioms reikia nurodyti pradinį linijos tašką, o toliau pati procedūra seka nurodyta linija ir fiksuoja jos mazgus. Automatinio vektorizavimo principas ir yra paremtas šiomis procedūromis. Rankiniame skaitmenizavime, jos naudojamos tik kaip pagalbinis įrankis (1).

Dar 1993 metais buvo kalbama apie algoritmus, kuriais galima gauti tiesius linijų segmentus, apvalius lankus bei rodyklių galus iš rastrinio vaizdo. Tokioms linijoms gauti naudojamas statmenas *zig-zag* metodas. Jis eina nuo vieno sekamos linijos krašto iki kito, paeiliui horizontaliomis ir vertikaliomis linijomis. Šio metodo privalumas – rastrinis vaizdas tikrinamas retais intervalais, labiausiai susikoncentruojant ties aptiktais primityvais, kuriuos reikia tirti ir vektorizuoti (8).

Po metų buvo sukurtas integruotas linijos sekos algoritmas. Metodas jungia kontūro sekimą bei vidurio ašies nustatymą. Trumpai tariant vidurio linija yra apskaičiuojama sekant linijos kontūrą. Tai reiškia, jog išvengiama linijų ploninimo bei kontūrų vektorizavimo procedūrų. Be to, galutinis rezultatas yra ne kodo grandinė (angl. *chain code*), bet greitai aptinkama polilinja.

Taigi, liniją tuo pačiu metu seka du kontūrų sekliai (angl. *contour followers*) – vienas iš kairės, kitas iš dešinės linijos pusės (8). Laikoma, kad rastrinis vaizdas yra dviejų lygių

(linijų struktūros bei fonas). Kontūrai sekami reikalingas pradinis taškas  $P_0$ . Toliau ieškomi jo kaimynai, vykdant gretimų taškų patikrinimą einant prieš laikrodžio rodyklę. Žiūrima, ar taškas priklauso linijai ar fonui, pagal tai atitinkamai jiems suteikiamos reikšmės. Šis procesas (kaimyninių taškų identifikavimas) kartojamas tol, kol pasiekiamas pradinis taškas  $P_0$ .

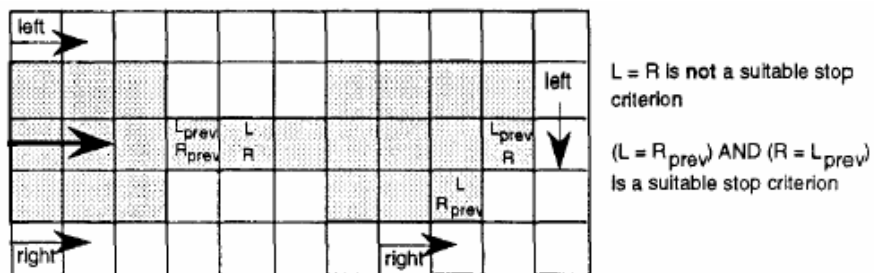


1 pav. a) kontūro sekimas prieš laikrodžio rodyklę; b) kontūrų sekimas iš abiejų pusių.

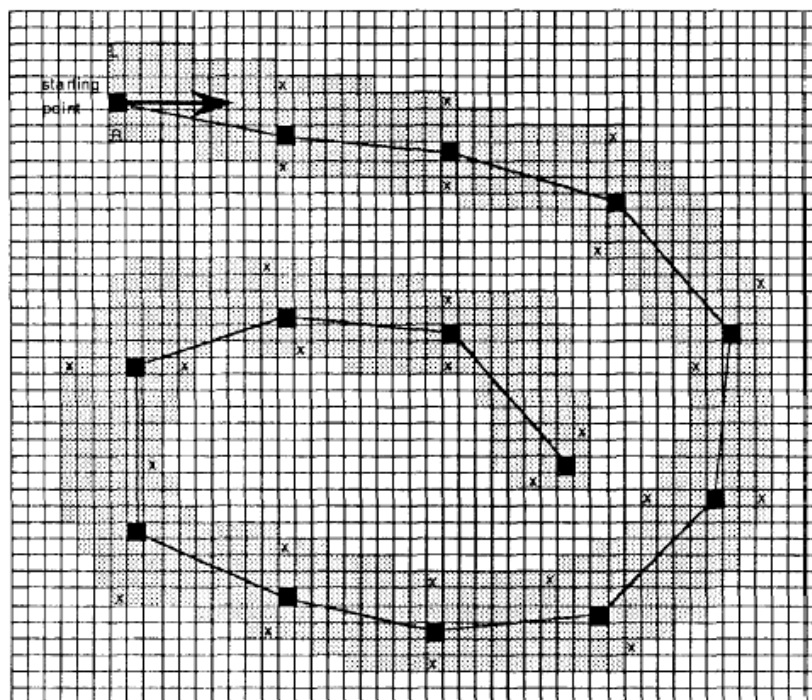
Sekant iškart iš abiejų pusių, pirmiausia veikia vienas „seklus“ tam tikrą laiką, po to kitas (paeiliui). Fiksuojami nauji taškai  $L_i$  bei  $R_i$ , tarp jų apskaičiuojamas vidurio taškas  $M_i$  (1 pav. b).

Šio algoritmo įvedimo duomenys yra ne tik rastrinis vaizdas, bet ir vartotojo nurodytas pradinis taškas bei linijos sekimo kryptis.

Procedūros pabaigos kriterijus  $L=R$  netinka tuo atveju, jei linija turi tokių vietų, kur jos plotis siekia vos vieną pikselį. Tai išsprendžia pritaikius sąlygą, pateiktą 2-ame paveiksle:  $(L = R_{prev}) \text{ AND } (R = L_{prev})$ . Kitas atvejis, lemiantis procedūros pabaigą yra uždaros kreivės, t.y. sustojama, pasiekus pradinį linijos tašką (8).

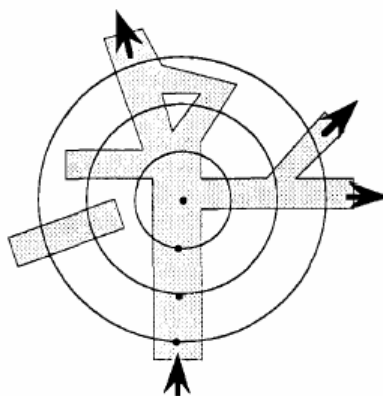


2 pav. Procedūros ( linijos pabaigos) kriterijus.



3 pav. Išskirta linija pagal linijos sekos algoritmą (slenksčio (angl. threshold) reikšmė 2).

Pagal šio algoritmo idėją, vienu metu sekama tik viena linija, todėl priėjus išsišakojimą, apskritimų metodu nustatomos visos galimos kryptys ir pateikiamos vartotojui, kad šis išsirinktų tinkamą. Tuomet linijos sekimo procesas tęsiasi.



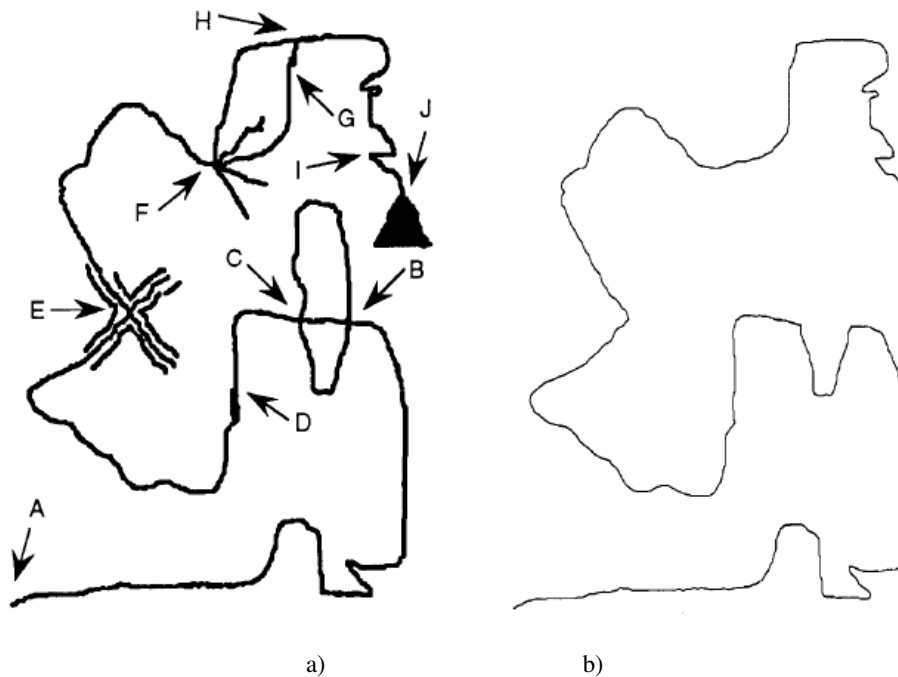
4 pav. Apskritimai apskritimų sekimo procese (angl. circlefollowing).

Kiek rasta naujų taškų porų ant didžiausio apskritimo, sekant liniją, tiek pasirinkimo variantų pateikiama vartotojui, kad šis išsirinktų tolesnę linijos sekimo kryptį.

Algoritmo parametrai, nuo kurių priklauso galutinis rezultatas, yra šie:

- išvedamų apskritimų skaičius taške:  $n$ ;
- slenksčio reikšmė:  $t$ ;
- maksimalus linijos pločio išsiplėtimo faktorius:  $e$ ;

- maksimalus tarpelių plotis: g.



5 pav. Algoritmui pateiktas rastrinis vaizdas (a) ir gautas vektorinis vaizdas (b).

Buvo patikrintos įvairios situacijos, gauti rezultatai parodyti 5 paveiksle. Algoritmo trūkumai: vektorizuoja tik po vieną liniją; vektorizuoja tik monochrominį vaizdą; problemų sukelia per storos linijos bei užpildyti plotai.

### 2.3.2. Vektorizavimo metodų palyginimas

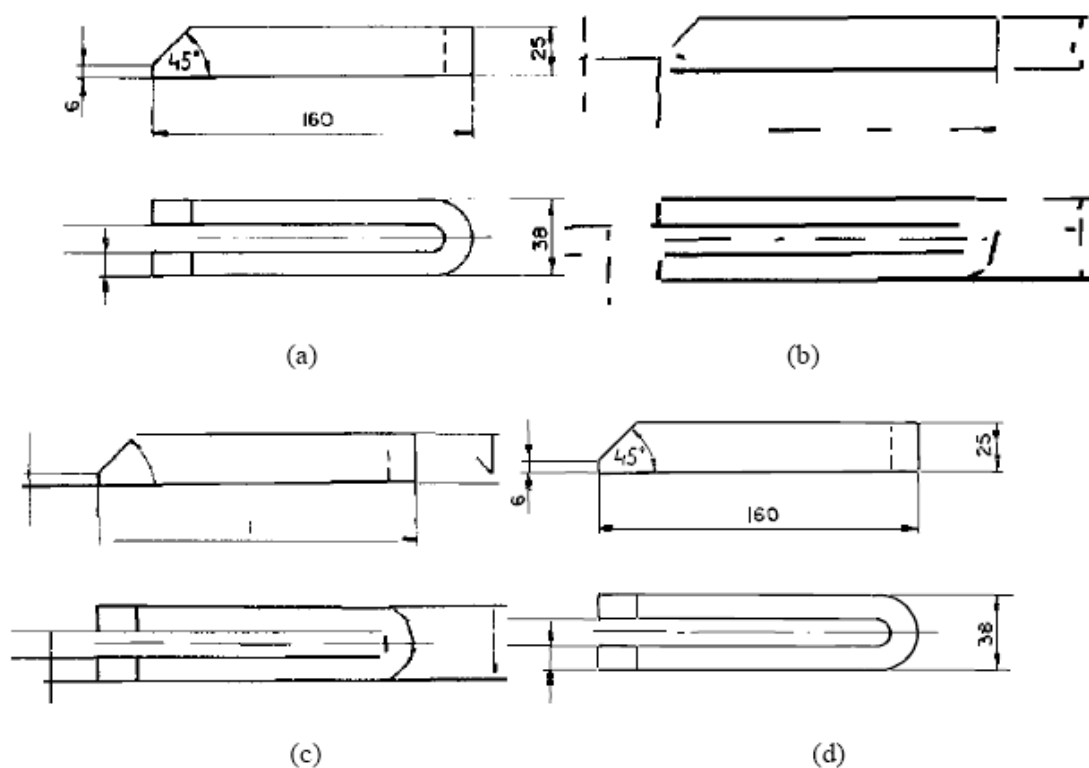
Gerai vektorizavimo metodai kaip galima tiksliau turėtų išsaugoti geometrinę informaciją apie aptiktus objektus, t.y.: linijos storį, linijos tipą, susikirtimo sujungimą ir pan. Praktinę vertę inžinerinėse sistemose įgauna tik pakankamai greiti vektorizavimo metodai.

Liu Wenyin ir Dov Dori (10) savo darbe palygino visas šešias vektorizavimo metodų grupes. Gauti rezultatai pateikti 2 lentelėje.

(\* s reiškia vidurio ašies taškų paiešką, t – linijos sekos procedūra)

Metodas	Subprocesų seka	Laiko sudėtingumas	Linijų geometrijos kokybė	Linijų pločio išsaugojimas	Sujungimų atstatymas	Vaizdo apribojimai (image constraints)
HT pagrindu	HT, t	kvadratinis	prasta	ne	taip	Išsibarstęs, tiesus
Pakartotino ploninimo	s, t	kubinis	labai gera	ne	ne	Švarus, plonas
Kontūro	Krašto paieška, t	kvadratinis	prasta	taip	ne	Tiesus
Bėgančio grafo (Run-graph)	Bėgančio grafo konstrukcija, t	kvadratinis	prasta	taip	ne	Tiesus
Tinklelio	s kol t	linijinis	prasta	taip	taip	Išsibarstęs, ilgas
OZZ	s kol t	linijinis	prasta	taip	taip	Tiesus
SPV	s kol t	linijinis	gera	taip	taip	

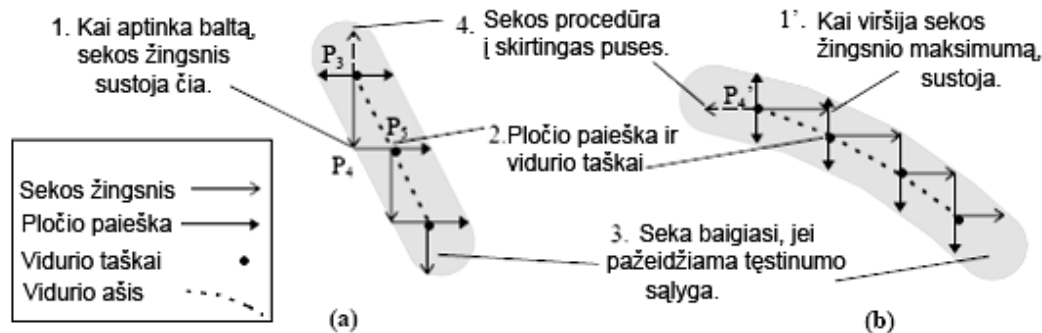
Iš 2 lentelės duomenų matyti, jog optimaliausias metodas bei geriausi vektorizavimo rezultatai pasiekiami SPV metodais. Tai įrodo ir vizualūs rezultatai. 6 paveiksle (a) parodytas originalus vaizdas, (b) – HT vektorizacija, (c) – OZZ vektorizacija, (d) – SPV.



6 pav. HT, OZZ bei SPV rezultatų palyginimas. (a) Originalas. (b) HT vektorizacija. (c) OZZ vektorizacija. (d) SPV.

### 2.3.3. SPV esmė

SPV vektorizavimo metodai tiria rastro taškus ne paeiliui, bet tam tikrais intervalais. Šių metodų esminė idėja grafiškai parodyta 7 paveiksle (4, 10, 11).



7pav. Iliustruota pagrindinė Sparse Pixel Tracing procedūra. (a) vertikalus atvejis; (b) horizontalus atvejis.

Pirmiausia randamas atkarpos pradžios taškas, kuris nepriklauso nuo vaizdo triukšmų. Toliau vykdoma vidurio taškų paieškos procedūra (7 pav.), kol tenkinamos tęstinumo sąlygos. Jei sekos žingsnis horizontalia kryptimi ilgesnis už vertikalią, tuomet linijos pasvirimo kampas mažesnis už  $45^\circ$ , o linijos pločio ieskoma statmena kryptimi, t.y. vertikaliai. O jei sekos žingsnis didesnis vertikalia kryptimi, tuomet pločio ieskoma horizontalia kryptimi. Kai bent viena tęstinumo sąlyga pažeidžiama, tuomet vykdoma sujungimų (susikirtimų) atstatymo procedūra. Taigi vektorizavimo algoritmas susideda iš atskirų aiškių etapų.

### 2.4. Rastrinio vaizdo vektorizavimo sistemos įgyvendinimo etapai

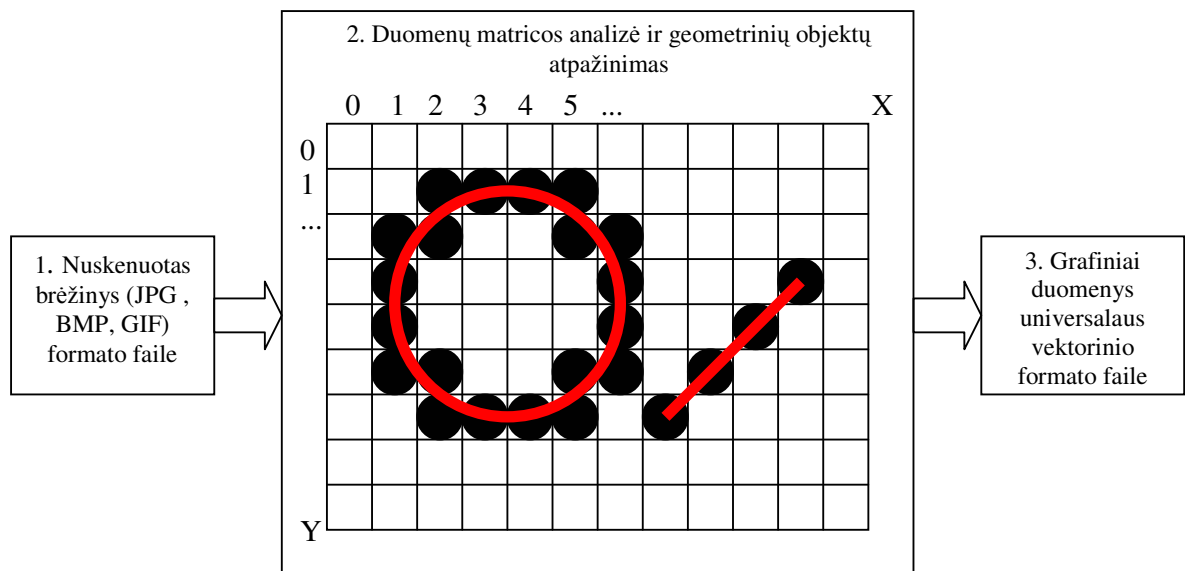
Remdamiesi ištyrinėta medžiaga bei aptiktais vektorizavimo metodų pavyzdžiais nusprendėme sukurti savo vektorizavimo algoritmą. Iš dalies jį būtų galima priskirti šeštai (SPV) vektorizavimo metodų grupei, nes rastrinio vaizdo taškai tiriami tam tikrais intervalais, o ne paeiliui. Taip pat linijos sekos, linijos pločio nustatymo bei atkarpos segmentų aproksimavimo procedūros buvo integruotos į bendrą vektorizavimo algoritmą (viskas atliekama vienu metu). Pats vektorizavimo algoritmas detaliau aprašytas projektinėje šio darbo dalyje, 3.4.4 skyriuje.



Buvo nuspręsta kurti universalią vektorizavimo sistemą. Nuskaitytą bmp, jpg ar gif formato rastrinį vaizdą konvertuosime į mūsų siūlomą universalų vektorinį formatą. Jame bus saugoma grafinė informacija gauta iš rastrinio vaizdo. Jį vėliau bus galima konvertuoti į bet koki kitą vektorinį formatą (dxf, wmf ir t.t.).

Taigi, visą vektorizavimo procesą daliname į tokius etapus (žr. 8 paveikslą):

- 1) Iš pradžių turime nuskaitytą rastrinį vaizdą (brėžinį). Jį importuojame į programą.
- 2) Sistema iš grafinės duomenų rinkmenos pasiima jai reikalingus duomenis ir sugeneruoja darbinę duomenų matricą (paprastumo dėlei toliau bus vadinama – darbinis failas), kuri bus toliau analizuojamas ieškant geometrinių objektų (atkarpu, apskritimų).
- 3) Analizės rezultatai išsaugomi kompiuterio atmintyje kaip universalus vektorinio formato failas.



8 pav. Pagrindiniai vektorizavimo proceso žingsniai.

## 2.5. Išvados

Remdamiesi atlikta analize galime daryti tokias išvadas:

- Popieriniai brėžiniai laikui bėgant nusidėvi, todėl juos reikia perkelti į kompiuterį - vektorizuoti.
- Brėžinių kiekis paprastai būna didelis, todėl rankiniu būdu viską vektorizuoti yra neįmanoma.
- Jau yra sukurta daug ir įvairių vektorizavimo algoritmų, tačiau efektyviausi SVP tipo (taškai tiriami tam tikrais intervalais).
- Kai kurios egzistuojančios ir gana gerus vektorizavimo rezultatus pateikiančios sistemos pakankamai brangios.
- Vektorizavimo algoritmo pagrindiniai žingsniai: vidurio ašies taškų atranka, linijų sekos procedūros, linijos segmentų aproksimavimas arba uždarų segmentų interpretavimas daugiakampiais.
- Vektorizavimo sistemos darbo etapai: rastrinio vaizdo importavimas, vektorizavimas, vektorinės informacijos eksportavimas.
- Optimali išeitis geram vektorizavimo rezultatui pasiekti – tai automatinio bei rankinio vektorizavimo sintezė. Be to, rastro paruošimas prieš vektorizavimą, taip pat gali pagerinti rezultatus.
- Tam tikrais atvejais vektorizavus vis tiek gaunami klaidingi rezultatai.
- Problemų sukelia tokie vaizdo elementai kaip: simboliai, linijų susikirtimo vietos, kampai, linijų bei jų galų tipas, užpildyti plotai.

Remdamiesi šiomis išvadomis suprojektavome rastrinio vaizdo vektorizavimo sistemą.

## 3. Projektinė dalis

### 3.1. Reikalavimų projektuojamai sistemai specifikacija (Volere šablonas)

*Volere šablonas* naudojamas pradiniam sistemos kūrimo etape kaip bazinis šablonas vartotojų reikalavimų užregistravimui. Šablonas suskirstytas į skyrius pagal reikalavimų tipą. Šablonas padeda sukaupti reikalavimus, kuriuos pateikia vartotojai per interviu arba kurie yra užregistruoti analizuojamo objekto veiklą reglamentuojančioje dokumentacijoje. Remiantis šablonu, galima aprašyti pagrindinius funkcinius bei nefunkcinius reikalavimus, sudarančius bendrą vartotojo reikalavimų modelį.

#### 3.1.1. Projekto varovai

##### 3.1.1.1. Sistemos paskirtis

Kuriama rastrinio vaizdo vektorizavimo sistema turinti palengvinti profesionalių ir patyrusių CAD vartotojų darbą, t.y. jiems vis dažniau prireikia ranka braižytus brėžinius perkelti į kompiuterį bei redaguoti. Didžiausios problemos kyla dėl to, jog vektorizavimo programa tiksliai nežino, kokio galutinio rezultato tikisi vartotojas. Kai kurių esamų panašių produktų kaina Lietuvos pirkėjui yra per didelė.

##### Projekto kūrimo pagrindas (pagrindimas)

Vektorizavimo priemonės reikalingos tokiose srityse kaip: architektūra, mechanika, elektrotechnika, kartografija ir t.t. Tačiau pats vektorizavimo procesas sukelia begales problemų, kurias būtina spręsti. Šiuo metu jau yra sukurta ne viena vektorizavimo programa, tačiau nei su viena iš jų neįmanoma pasiekti idealaus rezultato, nes tai sritis, kurioje nėra vieno ir tikslaus sprendimo būdo, todėl kuriamos galimos euristikos. Jų efektyvumas tiriamas lyginant gautus rezultatus (kiek jie artimi idealiam sprendimui) su tais pačiais pradiniais duomenimis. Optimaliausia vektorizavimo išeitis – rankinio ir automatinio vektorizavimo sintezė, kuri bus realizuota šiame projekte.

##### Sistemos tikslai (paskirtis)

Rastrinio vaizdo vektorizavimo sistemos pagrindinis tikslas yra rastrinio (kitaip dar skaitmeninio) vaizdo vektorizavimas. Šiam tikslui pasiekti, programa privalo sugebėti atlikti tokius pagrindinius veiksmus:

- vaizdo importavimą į programą;
- importuoto vaizdo redagavimas prieš vektorizavimą;
- automatinis vektorizavimas;

- interaktyvus vektorinio vaizdo redagavimas kompiuterio ekrane;
- vaizdo eksportavimas vektoriniu formatu.

### **3.1.1.2. Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys**

Užsakovas – konkretaus užsakovo nėra.

Pirkėjas – konkretaus pirkėjo nėra. Programinė įranga kuriama moksliniais tikslais.

Kiti sistema suinteresuoti asmenys. Kadangi projektas yra magistrinis darbas, juo suinteresuoti jo kūrėjai, t.y. magistrinio darbo vadovas – doc. Antanas Lenkevičius ir vykdytoja – KTU Informatikos fakulteto Multimedijos inžinerijos katedros magistrantė Jurgita Genytė.

### **3.1.1.3. Vartotojai**

Programinės įrangos vartotojais bus:

- universitetai, mokslinės grupės, studentai;
- kartografai, braižytojai;
- inžinieriai.

Svarbiausi iš jų yra: kartografai, braižytojai, inžinieriai. Į jų nuomonę ir poreikius atsižvelgiama pirmiausia.

Vartotojų tikslai:

- produktas turi būti lengvai naudojamas;
- intuityviai suprantamas;
- pilnai dokumentuotas ir pilna vartotojo pagalba darbo metu;
- maksimaliai išnaudojamas automatizavimas;
- produkto palaikymas ir tolimesnių funkcijų praplėtimas reikalui esant.

Sprendžiami uždaviniai:

- skaitmeninio vaizdo importavimas (konvertavimas į darbinį failą);
- jo redagavimas prieš vektorizavimą (triukšmų šalinimo filtrai, spalvų skaičiaus keitimas, kontrasto keitimas);
- pilnas automatinis vektorizavimas;
- interaktyvus vektorinio vaizdo redagavimas kompiuterio ekrane (charakteringų taškų panaikinimas ar pridėjimas, jų vietos keitimas, galimybė naudoti linijos sekos procedūras, galimybė redaguoti tik dalį bendro vaizdo);
- vaizdo eksportavimas vektoriniame formate.

Būtinasis vartotojo pasirengimas: bendros žinios apie rastrinę bei vektorinę grafiką, patirtis dirbant su programiniais grafikos redaktorais ar braižymo priemonėmis (pvz. AutoCAD).

### **3.1.2. Projekto apribojimai**

#### **3.1.2.1. Įpareigojantys apribojimai. (Mandated constraints)**

##### **Apribojimai sprendimui**

- Sistema skirta versti vaizdą iš skaitmeninės formos (BMP, JPG ar GIF formato) į vektorinę, bet ne atvirkščiai;
- Sistemos funkcijos turi atitikti vartotojų sprendžiamus uždavinius;
- Galimybė ateityje nesunkiai praplėsti programinės įrangos (PI) funkcijas atsiradus naujiems vartotojų reikalavimams (pavyzdžiui, kad PI suprastų ir kitus vaizdinių failų formatus: TIFF ir t.t.);
- Kuriama sistema turi veikti Windows 98 ir vėlesnėse operacinės sistemos versijose.

##### **Diegimo aplinka**

Sistema bus diegiama konkrečioje kompiuterizuotoje darbo vietoje. Minimalūs reikalavimai kompiuterio techninei bei programinei įrangai:

- Procesorius: ne mažiau 2GHz (nes vektorizavimo procesas imlus skaičiavimams ir laikui, todėl spartesnis procesorius iš dalies išsprendžia šią problemą);
- Operatyvinė atmintis: ne mažiau 512MB;
- Kietojo disko talpa: ne mažiau 80 GB;
- *Windows 98/2000/XP* operacinė sistema;
- Skaitytuvas (skeneris) ar plačiaformatis skaitytuvas (nebūtinai).

##### **Bendradarbiaujančios sistemos**

Kuriama sistema yra savarankiškas produktas, todėl darbo metu bendradarbiaujančių sistemų nėra.

##### **Komerciniai specializuoti programų paketai**

Į kuriamą sistemą nenumatoma įtraukti jokių papildomų specializuotų programų paketų.

##### **Numatoma darbo vietos aplinka**

- Stacionari tradicinė kompiuterizuota darbo vieta;
- Programa nereikalauja autorizuoto vartotojo prisijungimo;
- Laisvas priejimas prie reikiamų resursų (darbo metu sukurtų failų);
- Skaitytuvas (skeneris), priklausomai nuo turimų pradinių duomenų (ar popieriniame pavidale, ar skaitmeniniame).

### **Sistemos kūrimo terminai**

- Projekto aplinkos analizė ( 2005 m. sausio mėn.);
- Reikalavimų analizė (2005 m. kovo mėn.);
- Projektavimas (2005 m. birželio mėn.);
- Posistemių realizavimas ir testavimas (2006 m. vasario mėn.);
- Posistemių integracija ir testavimas (2006 m. balandžio mėn.);
- Programų sistemos diegimas (2006 m. gegužės mėn.).

### **Sistemos kūrimo biudžetas**

Kadangi projektas vykdomas studijų ribose, biudžetas ir finansavimas nėra numatyti.

#### **3.1.2.2. Terminų žodynas**

PĮ – programinė įranga.

Produktas – rastrinio vaizdo vektorizavimo sistema.

Rastrinis vaizdas – vaizdas elektroninėje formoje, sudarytas iš atskirų taškų, kurių kiekvienas turi savo informaciją apie spalvas.

Vektorinis vaizdas – vaizdas sudarytas iš matematinių primityvų (atkarpu, apskritimų ir pan.), kurie turi savo atributus (pvz. atkarpos galų koordinatės  $x$ ,  $y$ , linijos storis ir pan.).

Sistemos vartotojas – bet kuris asmuo paminėtas 3.1.1.3 šio dokumento skyriuje.

#### **3.1.2.3. Svarbūs faktai ir prielaidos**

Faktai: nėra.

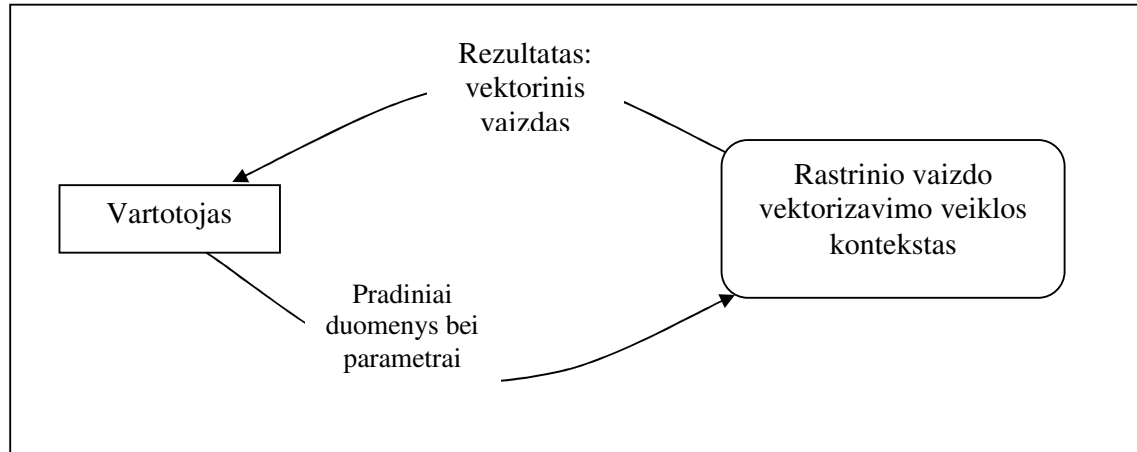
Prielaidos:

- Atskiros programos funkcijos gali būti pakeistos (patikslintos) pagal vartotojų pageidavimą;
- Atskirų komponentų kūrimas gali nesutapti su numatytu terminu;
- Tikimasi toliau tobulinti produktą ir pritaikyti pardavimui.

### 3.1.3. Funkciniai reikalavimai

#### 3.1.3.1. Veiklos sfera

##### Veiklos kontekstas (pateikiama konteksto diagrama)



9 pav. Konteksto diagrama.

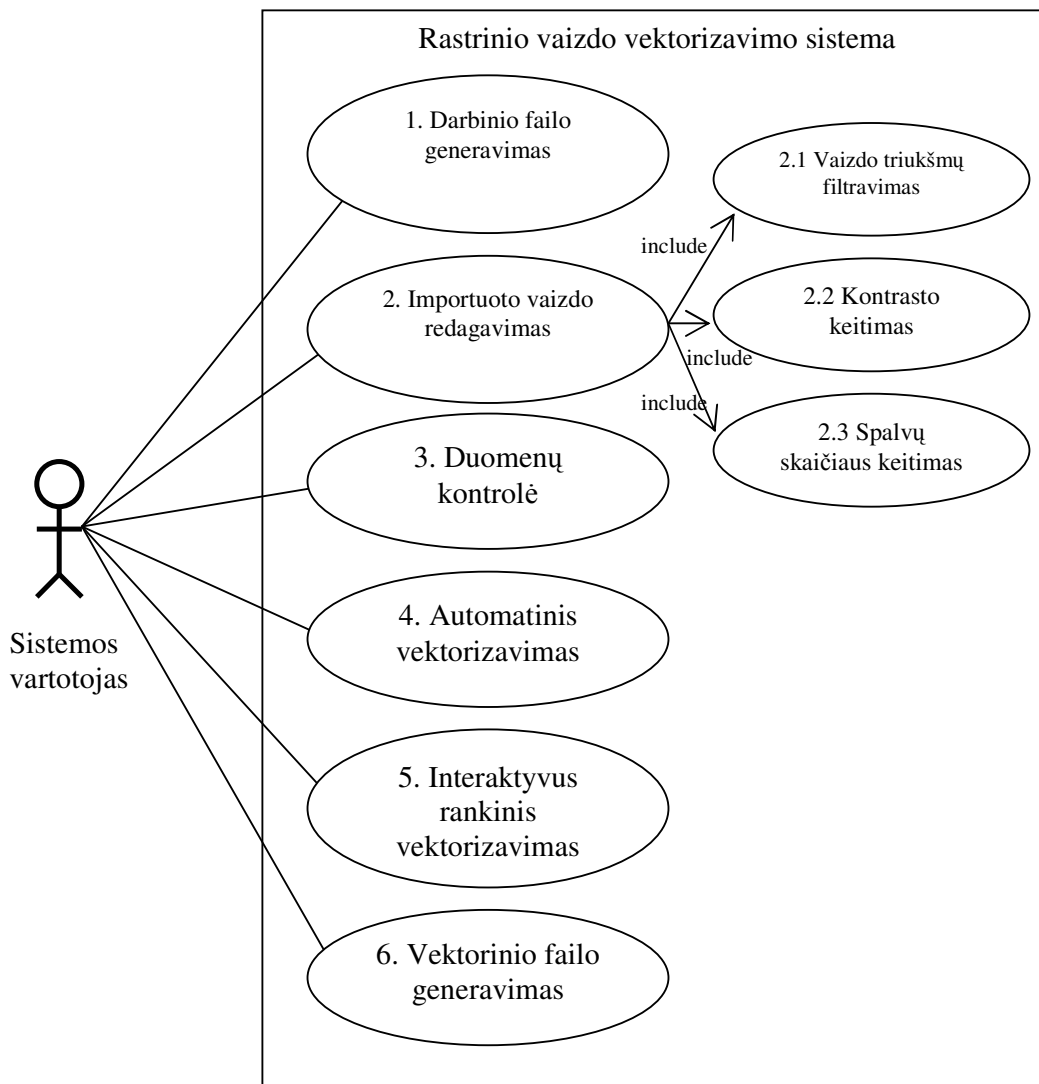
#### Veiklos padalinimas

3 lentelė. Veiklos įvykių sąrašas

<i>Eil. Nr.</i>	<i>Įvykio pavadinimas</i>	<i>Įeinantys(in)/Išeinantys(out) informacijos srautai</i>
1	Vartotojas paduoda sistemai rastrinį vaizdą.	BMP, JPG arba GIF formato failas (in)
2	Sistema atlieka failo konvertavimą į savo darbinę duomenų struktūrą – darbinį failą.	Rezultatai_d (out)
3	Vartotojas nurodo vaizdo redagavimo parametrus (triukšmų šalinimo filtrų parametrus, kontrasto bei spalvų skaičiaus vertes).	Parametrai_r (in)
4	Sistema perskaičiuoja turimus duomenis ir atvaizduoja gautą rezultatą ekrane.	Rezultatai_t0 (out)
5	Vartotojas nurodo vektorizavimo parametrus	Parametrai_v1 (in)
6	Sistema atlieka turimų duomenų (vaizdinio failo) kontrolę ir praneša apie situaciją vartotojui.	Pranešimas (out)
7	Sistema atlieka automatinį vaizdo vektorizavimą.	Rezultatai_t1 (out)
8	Vartotojas interaktyviai redaguoja gautą rezultatą.	Parametrai_v2 (in) Rezultatai_t2 (out)
9	Vartotojas patvirtina gauto rezultato tinkamumą (redagavimo pabaiga).	Patvirtinimas (in)
10	Sistema formuoja vektorinio formato failą ir jį atvaizduoja ekrane.	Rezultatai_g (out)

### 3.1.3.2. Produkto veiklos sfera

#### Sistemos ribos



10pav. Panaudojimo atvejų diagrama.

#### Panaudojimo atvejų sąrašas

<b>1. PANAUDOJIMO ATVEJIS:</b>	Darbinio failo generavimas.
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas.
<b>Aprašas:</b>	BMP, JPG arba GIF formato failas verčiamas į darbinę duomenų struktūrą.
<b>Prieš sąlyga:</b>	Vartotojo kompiuterio kietajame diske turi būti bent vienas BMP, JPG ar GIF formato failas.



<b>Sužadavimo sąlyga:</b>	Vartotojas turi pasirinkti vieną iš prieš tai išvardintų formatų failą, esantį kompiuterio kietajame diske.
<b>Po-sąlyga:</b>	Pasirinktas failas išanalizuojamas, paimami tik reikalingi duomenys, o vaizdas parodomas kompiuterio ekrane.

<b>2. PANAUDOJIMO ATVEJIS:</b>	Importuoto vaizdo redagavimas.
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas.
<b>Aprašas:</b>	Redaguojama sugeneruota darbinių duomenų struktūra. Rezultatai atvaizduojami ekrane.
<b>Prieš sąlyga:</b>	Vartotojas ekrane turi matyti redaguojamą vaizdą, kad galėtų įvertinti, kokius redagavimo parametrus pateikti sistemai.
<b>Sužadavimo sąlyga:</b>	Vartotojas sistemai įveda reikalingus redagavimo parametrus.
<b>Po-sąlyga:</b>	Perskaičiuoti duomenys (pakoreguotas vaizdas) turi būti atvaizduotas kompiuterio ekrane. Galimybė pakartoti veiksmus iš naujo arba grįžti atgal, jei rezultatas netenkina.

<b>2.1 PANAUDOJIMO ATVEJIS:</b>	Vaizdo triukšmų filtravimas.
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas.
<b>Aprašas:</b>	Filtruojami vaizdo triukšmai. Rezultatas atvaizduojamas ekrane.
<b>Prieš sąlyga:</b>	Vartotojas ekrane turi matyti darbinio failo vaizdą, kad galėtų įvertinti, kokius vaizdo filtravimo parametrus paduoti sistemai.
<b>Sužadavimo sąlyga:</b>	Vartotojas sistemai paduoda filtravimo parametrus.
<b>Po-sąlyga:</b>	Perskaičiuoti duomenys (pakoreguotas vaizdas) turi būti atvaizduotas kompiuterio ekrane. Galimybė pakartoti veiksmą iš naujo arba grįžti žingsniu atgal, jei rezultatas netenkina.

<b>2.2 PANAUDOJIMO ATVEJIS:</b>	Kontrasto keitimas.
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas.
<b>Aprašas:</b>	Didinamas vaizdo kontrastas. Rezultatas atvaizduojamas ekrane.
<b>Prieš sąlyga:</b>	

<b>Sužadavimo sąlyga:</b>	Vartotojas ekrane turi matyti darbinio failo vaizdą, kad galėtų įvertinti, kokią kontrasto reikšmę paduoti sistemai.
<b>Po-sąlyga:</b>	Vartotojas sistemai paduoda kontrasto reikšmę. Perskaičiuoti duomenys (pakoreguotas vaizdas) turi būti atvaizduotas kompiuterio ekrane. Galimybė pakartoti veiksmą iš naujo arba grįžti žingsniu atgal, jei rezultatas netenkina.

<b>2.3 PANAUDOJIMO ATVEJIS:</b>	Spalvų skaičiaus keitimas.
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas.
<b>Aprašas:</b>	Mažinamas spalvų skaičius vaizde. Rezultatas atvaizduojamas ekrane.
<b>Prieš sąlyga:</b>	Vartotojas turi būti informuotas apie pradinį spalvų skaičių vaizde ir matyti pasirinkimo variantus iki kiek galima būtų sumažinti spalvų skaičių.
<b>Sužadavimo sąlyga:</b>	Vartotojas pasirenka spalvų skaičiaus reikšmę iš tų, ką siūlo sistema.
<b>Po-sąlyga:</b>	Perskaičiuoti duomenys (pakoreguotas vaizdas) turi būti atvaizduotas kompiuterio ekrane. Galimybė pakartoti veiksmą iš naujo arba grįžti žingsniu atgal, jei rezultatas netenkina.

<b>3. PANAUDOJIMO ATVEJIS:</b>	Duomenų kontrolė
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas.
<b>Aprašas:</b>	Tikrinama darbinės duomenų struktūros būseną/duomenys, kad būtų įvertinta ar galimas automatinis vektorizavimas.
<b>Prieš sąlyga:</b>	Normalios būsenos darbinė duomenų struktūra.
<b>Sužadavimo sąlyga:</b>	Vartotojas paleidžia automatinį vektorizavimą.
<b>Po-sąlyga:</b>	Sistema generuoja pranešimą, kad automatinis vektorizavimas galimas arba ne.

<b>4. PANAUDOJIMO ATVEJIS:</b>	Automatinis vektorizavimas
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas.
<b>Aprašas:</b>	Atliekamas pilnai automatinis skaitmeninio vaizdo (darbinio failo) vektorizavimas (t.y. vaizdo taškų analizė, atitinkamų taškų koordinatų fiksavimas).

<b>Prieš sąlyga:</b>	Duomenų kontrolės rezultatas turi būti teigiamas.
<b>Sužadinimo sąlyga:</b>	Vartotojas sistemai nurodo vektorizavimo parametrus.
<b>Po-sąlyga:</b>	Vektorizavimo rezultatas turi būti atvaizduotas kompiuterio ekrane. Galimybė pakartoti veiksmą iš naujo, nurodžius kitus vektorizavimo parametrus, arba sugrįžti į vaizdo redagavimą dar prieš jį vektorizuojant, jei rezultatas netenkina.

<b>5. PANAUDOJIMO ATVEJIS:</b>	Interaktyvus rankinis vektorizavimas
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas.
<b>Aprašas:</b>	Interaktyviai atliekami vaizdo pakeitimai kompiuterio ekrane.
<b>Prieš sąlyga:</b>	Sėkmingai vektorizuotas vaizdas.
<b>Sužadinimo sąlyga:</b>	Kompiuterinės pelės mygtuko paspaudimai atitinkamose vaizdo vietose sužadina vaizdo redagavimo galimybę toje vietoje.
<b>Po-sąlyga:</b>	Rezultatas iškart matomas kompiuterio ekrane. Galimybė pakartoti veiksmą iš naujo arba grįžti žingsniu atgal, jei rezultatas netenkina.

<b>6. PANAUDOJIMO ATVEJIS:</b>	Vektorinio failo generavimas.
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas.
<b>Aprašas:</b>	Vektorizavimo metu gauti duomenys surašomi į universalaus formato failą.
<b>Prieš sąlyga:</b>	Sistemoje vektorizuotas vaizdas normalioje būsenoje ir tenkinantis vartotoją.
<b>Sužadinimo sąlyga:</b>	Vartotojas pasirenka failo išsaugojimo funkciją.
<b>Po-sąlyga:</b>	Vektorinio formato failas kompiuterio kietajame diske ir sistemos patvirtinimas apie sėkmingą darbo pabaigą.

### 3.1.3.3. Funkciniai reikalavimai ir reikalavimai duomenims

#### Funkciniai reikalavimai

<b>Reikalavimas #:</b>	R1	<b>Reikalavimo tipas:</b>	Įvykis/panaudojimo atvejis	1
<b>Aprašymas:</b>	Sistemoje turi būti priemonės skirtos BMP, JPG bei GIF formato failų paėmimui iš vartotojo kompiuterio pastovios atminties į sistemos aplinką.			

<b>Pagrindimas:</b>	Jokie kiti failų formatai nepasiekiami.		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Failo pasirinkimo/atidarymo metu, matomi tik sąrašas BMP, JPG bei GIF formatų failai.		
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.		

<b>Reikalavimas #:</b>	R2	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	1
<b>Aprašymas:</b>	Sistema pasirinktą failą konvertuoja į savo darbinį failą (t.y.: pasiima iš jo, tik jai reikalingus duomenis ir suformuoja tų duomenų masyvą).			
<b>Pagrindimas:</b>	Numatyta tam, kad ateityje būtų galima sistemą praplėsti (t.y. kad būtų galima dirbti ir su kitais vaizdinių failų formatais).			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Sistemos suformuotas darbinis failas (duomenų masyvas).			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

<b>Reikalavimas #:</b>	R3	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	1,2,1,2,2 ,2,3,4,6
<b>Aprašymas:</b>	Sistema informuoja apie sėkmingą arba ne operacijos pabaigą.			
<b>Pagrindimas:</b>	Vartotojas turi žinoti, kada gali imtis tolesnių veiksmų ir apskritai ar gali toliau dirbti su gautais duomenimis.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Sistemos pranešimas apie atitinkamos operacijos pabaigą.			
<b>Priklausomybės:</b>		<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

<b>Reikalavimas #:</b>	R4	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	1,2,2,1,2 ,2,2,3,4
<b>Aprašymas:</b>	Sistema gautą rezultatą atvaizduoja kompiuterio ekrane.			

<b>Pagrindimas:</b>	Vartotojas turi vizualiai įvertinti gautą rezultatą (nuo to priklauso tolesni jo veiksmai).		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Atvaizdas kompiuterio ekrane.		
<b>Priklausomybės:</b>		<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.		

<b>Reikalavimas #:</b>	R5	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	2
<b>Aprašymas:</b>	Vartotojas turi turėti bent kelias priemones vaizdo redagavimui.			
<b>Pagrindimas:</b>	Redagavimo priemonės prieš vektorizavimą užtikrina geresnį vektorizavimo rezultatą.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Vaizdo redagavimo priemonių sąrašas su pasirinkimo galimybe.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

<b>Reikalavimas #:</b>	R6	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	2
<b>Aprašymas:</b>	Turi būti fiksuojama redagavimo žingsnių istorija.			
<b>Pagrindimas:</b>	Jei rezultatas netenkina vartotojas galės nesunkiai grįžti atgal.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Visada aktyvus redagavimo žingsnių sąrašas.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

<b>Reikalavimas #:</b>	R7	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	2.1, 2.2, 2.3
<b>Aprašymas:</b>	Vartotojas pats pateikia sistemai parametrus.			

<b>Pagrindimas:</b>	Kiekvienu atveju gali reikėti vis kitokių reikšmių skirtingiems redagavimo algoritmams, tad reikalinga ši funkcija, kad nereikėtų programuotojo įsikišimo.		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Tam tikra forma sistemoje, į kurią vartotojas gali įrašyti parametrus.		
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.		

<b>Reikalavimas #:</b>	R8	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	2.1, 2.2, 2.3
<b>Aprašymas:</b>	Pateiktų parametrų bei jų sintaksės patikrinimas.			
<b>Pagrindimas:</b>	Vartotojas per klaidą gali įvesti nekorektiškus duomenis, todėl sistema turi būti nuo to apsaugota (kad išvengti klaidų, aklaviečių, sistemos užlūžimo ir t.t.).			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Sistema patikrina vartotojo įvestus parametrus/duomenis.			
<b>Priklausomybės:</b>	R7	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

<b>Reikalavimas #:</b>	R9	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	2.1
<b>Aprašymas:</b>	Sistema turi įvykdyti vaizdo triukšmų filtravimą.			
<b>Pagrindimas:</b>	Triukšmų filtravimas vaizde pagerina vektorizavimo rezultatą. Gaunamas tikslesnis galutinis vaizdas (be įvairių šiukšlių ir pan.).			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Korektiškai veikiantis filtravimo algoritmas.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

<b>Reikalavimas #:</b>	R10	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	2.2
<b>Aprašymas:</b>	Sistema turi leisti keisti vaizdo kontrastą.			

<b>Pagrindimas:</b>	Kontrasto didinimas vaizde pagerina vektorizavimo rezultata. Išryškėja objektų vaizde kontūrai.		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Korektiškai veikiantis kontrasto keitimo algoritmas.		
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.		

<b>Reikalavimas #:</b>	R11	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	2.3
<b>Aprašymas:</b>	Sistema turi leisti keisti spalvų skaičių vaizde.			
<b>Pagrindimas:</b>	Spalvų skaičiaus mažinimas vaizde pagerina vektorizavimo rezultata. Išryškėja objektų vaizde kontūrai.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Korektiškai veikiantis spalvų skaičiaus mažinimo algoritmas.			
<b>Priklausomybės:</b>	R1	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

<b>Reikalavimas #:</b>	R12	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	3
<b>Aprašymas:</b>	Sistema laukia ženklo iš vartotojo apie tai, kad laikas vektorizuoti turimą vaizdą.			
<b>Pagrindimas:</b>	Sistema nežino, kada vartotojas yra patenkintas turimu vaizdu ir, kad galima vektorizuoti.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Sistemoje egzistuoja automatinio vektorizavimo paleidimo mygtukas.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

<b>Reikalavimas #:</b>	R13	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	3
<b>Aprašymas:</b>	Sistema turi atlikti turimų duomenų (redaguoto darbinio failo) patikrinimą, ar tinkamas automatiniam vektorizavimui.			

<b>Pagrindimas:</b>	Vaizdas, kuris bus vektorizuojamas turi atitikti tam tikrus reikalavimus (pvz. spalvų skaičius ir pan.). Vartotojas galėjo atsitiktinai pamiršti kai kurias korekcijas, todėl sistema turi pati tai kontroliuoti ir priminti, jei reikia, vartotojui apie jo neatliktus ar atliktus neteisingai veiksmus.		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Vaizdo tinkamumo vektorizavimui patikrinimas.		
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.		

<b>Reikalavimas #:</b>	R14	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	3
<b>Aprašymas:</b>	Sistema turi išvesti į ekraną duomenų kontrolės rezultatus (sąrašo forma).			
<b>Pagrindimas:</b>	Jei sistema neleidžia toliau automatiškai vektorizuoti, vartotojas turi žinoti kokių toliau veiksmų imtis (pvz. atlikti papildomą vaizdo redagavimą).			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Duomenų kontrolės rezultatai išrašomi sąrašo forma kompiuterio ekrane.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

<b>Reikalavimas #:</b>	R15	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	4
<b>Aprašymas:</b>	Sistema turi paprašyti vartotojo įvesti vektorizavimo parametrus.			
<b>Pagrindimas:</b>	Kiekvienu atveju gali reikėti vis kitokių reikšmių vektorizavimo algoritmui, todėl reikalinga ši funkcija, kad nereikėtų programuotojo įsikišimo.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Sistemos tam tikra forma į kurią vartotojas gali įvesti atitinkamus parametrus.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

<b>Reikalavimas #:</b>	R16	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	4
------------------------	-----	---------------------------	--------------------------------------	---



<b>Aprašymas:</b>	Sistema turi patikrinti ar vartotojas teisingai įvedė vektorizavimo parametrus.		
<b>Pagrindimas:</b>	Parametrai reikalingi vektorizavimo algoritmui. Be jų, vektorizavimo procesas nebus sėkmingas.		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Sistema patikrina vektorizavimo parametrų teisingumą.		
<b>Priklausomybės:</b>	R15	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.		

<b>Reikalavimas #:</b>	R17	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	5
<b>Aprašymas:</b>	Interaktyvus vaizdo redagavimas kompiuterio ekrane (pelytės pagalba).			
<b>Pagrindimas:</b>	Automatinio vektorizavimo trūkumams pašalinti.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Paspaudus su pelyte ant atitinkamos vaizdo vietos aktyvizuojamas jo redagavimas.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

<b>Reikalavimas #:</b>	R18	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	5
<b>Aprašymas:</b>	Sistema nuolat tikrina ar vartotojas neatlieka jokių neleistinių veiksmų su vaizdu, interaktyvaus jo redagavimo metu. Jei atliekamas neleistinas veiksmas sistema apie tai informuoja vartotoją.			
<b>Pagrindimas:</b>	Sistemos galimybės nėra beribės. Tam tikri vartotojo veiksmai gali būti neapgalvoti ar netyčiniai ir sistema privalo atitinkamai į tai reaguoti.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Sistema seka vartotojo veiksmus.			
<b>Priklausomybės:</b>	R17	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

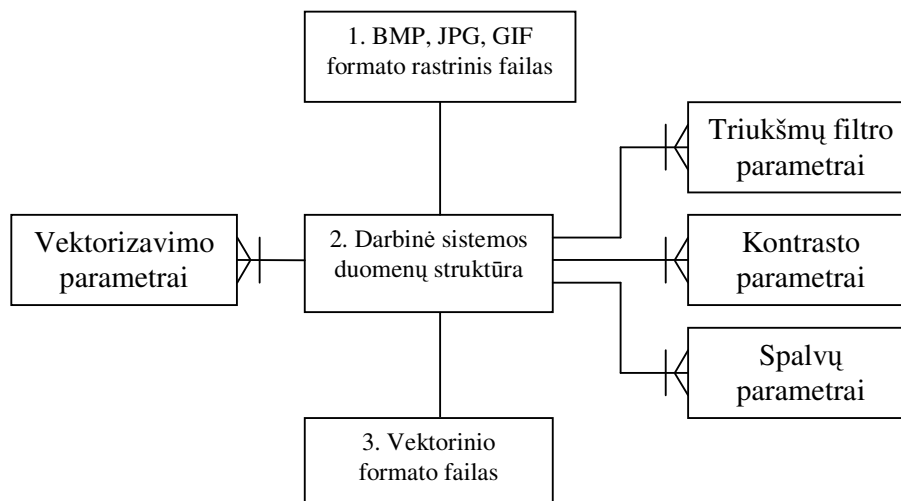
<b>Reikalavimas #:</b>	R19	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	5
------------------------	-----	---------------------------	--------------------------------------	---

<b>Aprašymas:</b>	Interaktyvaus redagavimo metu gaunami rezultatai realiu laiku vaizduojami ekrane.		
<b>Pagrindimas:</b>	Vartotojas turi matyti tai, ką daro su turimu vaizdu realiu laiku, kad jo neišdarytų.		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Redaguojamo vaizdo pakitimų atvaizdavimas realiu laiku.		
<b>Priklausomybės:</b>	R17, R18	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.		

<b>Reikalavimas #:</b>	R20	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	6
<b>Aprašymas:</b>	Sistemos atitinkamas mygtukas ar meniu punktas, kuriuo vartotojas liepia sistemai eksportuoti vektorizuotą failą, t.y. iš sistemos darbinio failo suformuoti atitinkamą vektorinio formato failą.			
<b>Pagrindimas:</b>	Sistema leidžia vartotojui pačiam nuspręsti apie tai ar gautas rezultatas tenkina, ar ne, ar reikia dar grįžti prie vaizdo redagavimo, ar nebe.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Sistemos mygtukas ar vienas iš meniu punktų, reiškiantis failo eksportavimą.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

<b>Reikalavimas #:</b>	R21	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	6
<b>Aprašymas:</b>	Sistema generuoja galutinį vektorinį failą ir išsaugoją jį kompiuterio atmintyje.			
<b>Pagrindimas:</b>	Sistemos darbo metu gauto rezultato įtvirtinimas, reikalingas kaip galutinis sistemos darbo produktas.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Kompiuterio pastoviojoje atmintyje išsaugotas vektorinis failas.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 5d.			

## Reikalavimai duomenims



11 pav. Duomenų modelio diagrama.

11 paveiksle pavaizduota duomenų modelio diagrama: 1) Įvedami duomenys (rastrinis vaizdas) gali būti tik standartinių BMP, JPG arba GIF formatų. 2) Darbinė sistemos duomenų struktūra – tai vaizdo duomenų matrica, t.y. dvimatis masyvas. 3) Vektorinio formato failas – mūsų pasiūlytas universalus vektorinis formatas. Vienas rastrinis failas sistemoje transformuojamas į vieną darbinę duomenų struktūrą ir vieną galutinį vektorinio formato failą (ryšys **1:1**). Parametrai (vektorizavimo, triukšmų filtro, kontrasto bei spalvų) turi baigtinį skaičių verčių, iš kurių vartotojas gali pasirinkti tinkamą konkrečiu atveju (ryšys **1:n**, t.y. vienai darbinei duomenų struktūrai gali būti parinkti keli parametrai, keletą kartų, kol bus pasiektas norimas rezultatas).

### 3.1.4. Nefunkciniai reikalavimai

#### 3.1.4.1. Reikalavimai sistemos išvaizdai

- Intuityviai suprantama ir vartotojui „draugiška“ grafinė sąsaja.
- Profesionalus, techninis dizainas.

#### 3.1.4.2. Reikalavimai panaudojamumui

Reikalavimas #:	R22	Reikalavimo tipas:	Įvykis/panaudojimo atvejis #:	Visi
Aprašymas:	Sistema turi būti paprasta naudotis.			

<b>Pagrindimas:</b>	Sistema bus naudojama kasdien, todėl ji turi būti pakankamai paprasta, kad darbuotojas nevengtų naudotis sistema.		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Žinantis funkcinės sistemos galimybes darbuotojas sugebėtų naudotis sistema be apmokymo.		
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. kovo 15d.		

<b>Reikalavimas #:</b>	R23	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Specialių simbolių ir žymėjimų panaudojimas.			
<b>Pagrindimas:</b>	Sistema turi būti pritaikyta inžinieriniams darbams.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Vartotojui turi būti aiškūs ir suprantami visi sistemoje naudojami žymėjimai. Nauji (specifiniai) žymėjimai turi būti trumpai paaiškinti arba suprantami intuityviai.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 15d.			

<b>Reikalavimas #:</b>	R24	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Sistemos vartotojo sąsaja turi būti tiek lietuviška, tiek angliška.			
<b>Pagrindimas:</b>	Sistema bus platinama tiek šalies viduje, tiek užsienyje.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Sistemos instaliavimo metu galimybė pasirinkti vartotojo sąsajos kalbą: lietuvių arba anglų.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 15d.			

### 3.1.4.3. Reikalavimai vykdymo charakteristikoms

<b>Reikalavimas #:</b>	R25	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Resursų panaudojimo efektyvumas.			

<b>Pagrindimas:</b>	Sistema turi efektyviai išnaudoti turimus resursus, nekenkiant kitoms gretimai instaliuotoms sistemoms, netrikdant jų veiklos.		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Sistema netrikdo kitų programų darbo. Leidžia vartotojui lygiagrečiai dirbti su keliomis programomis.		
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. kovo 15d.		

<b>Reikalavimas #:</b>	R26	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Sistemos išplečiamumas.			
<b>Pagrindimas:</b>	Galimybė integruoti į sistemą naujus komponentus arba patobulinti esamus.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Naujo programinio komponento integravimas neturi sukelti didelių problemų, net jeigu šio projekto vykdytojai negalės tiesiogiai dalyvauti sistemos pakeitimuose.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 15d.			

<b>Reikalavimas #:</b>	R27	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Korektiškumas.			
<b>Pagrindimas:</b>	Sistemoje neturi būti klaidų.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Sistema funkcionuoja be klaidų arba klaidų skaičius yra minimalus ir netrikdo vartotojo darbo.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 15d.			

#### 3.1.4.4. Reikalavimai veikimo sąlygoms

<b>Reikalavimas #:</b>	R28	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
------------------------	-----	---------------------------	--------------------------------------	------

<b>Aprašymas:</b>	Sistema - nepriklausomas išbaigtas produktas.		
<b>Pagrindimas:</b>	Sistema turi būti pritaikyta darbui konkrečioje nepriklausomoje kompiuterizuotoje darbo vietoje. Ypatingų darbo sąlygų nėra.		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Sistema funkcionuoja nepriklausomoje kompiuterizuotoje darbo vietoje.		
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. kovo 15d.		

### 3.1.4.5. Reikalavimai sistemos priežiūrai

Sistemos priežiūros veiksniai:

- Atsiradusios klaidos (kūrimo metu, testavimo metu);
- Užsakovo reikalavimų pasikeitimas;
- Personalo kaita (kaip projekto vykdytojų komandoje, taip ir užsakovo pusėje);
- Techninės įrangos ir kuriamo produkto suderinamumo klausimai.

Efektyvesniam sistemos palaikymui ir priežiūrai sistemą būtina dokumentuoti. Dokumentacija turi būti laiku atnaujinama, vos atsiradus sistemos pakeitimams.

Reikalavimas #:	R29	Reikalavimo tipas:	Įvykis/panaudojimo atvejis #:	Visi
<b>Aprašymas:</b>	Minimalus rezervas palaikymo darbams atlikti.			
<b>Pagrindimas:</b>	Daugelis projektų vėluoja dėl nenumatytų darbų atsiradimo projekto pabaigoje, todėl turi būti iš anksto suplanuotas laikas, kuris bus skirtas palaikymo darbams atlikti.			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Nevėluojantis projektas (vėluojantis minimaliai).			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 15d.			

Reikalavimas #:	R30	Reikalavimo tipas:	Įvykis/panaudojimo atvejis #:	Visi
<b>Aprašymas:</b>	Dokumentavimo sistema.			

<b>Pagrindimas:</b>	Palaikyti tokią dokumentavimo sistemą, kuri vėliau leistų atlikti palaikymo darbus minimaliomis sąnaudomis.		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Laiku atnaujinama dokumentacija.		
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. kovo 15d.		

### 3.1.4.6. Reikalavimai saugumui

<b>Reikalavimas #:</b>	R31	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Vientisumas.			
<b>Pagrindimas:</b>	Sistemai perduodami šaltinio duomenys neturi būti iškraipomi.			
<b>Šaltinis:</b>	Vartotojas.			
<b>Tinkamumo kriterijus:</b>	Sistemos duomenys vienareikšmiškai atitinka šaltinio duomenis (pvz., nėra iškraipomi nuskaitymo metu).			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 15d.			

Kuriama sistema bus prieinama laisvai, t.y. autorizuotas vartotojų prisijungimas prie sistemos realizuotas nebus. Dėl šios priežasties reikalavimai konfidencialumui nėra keliami.

### 3.1.4.7. Kultūriniai-politiniai reikalavimai

<b>Reikalavimas #:</b>	R32	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Sistema nenaudoja įžeidžiančių terminų ir simbolių.			
<b>Pagrindimas:</b>	Sistemą gali naudoti įvairių tikėjimų ir įsitikinimų žmonės, todėl privalo būti naudojamos tik visoms vartotojų kategorijoms priimtinos notacijos.			
<b>Šaltinis:</b>	Vartotojas.			
<b>Tinkamumo kriterijus:</b>	Produktas atitinka etikos normas.			

<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. kovo 15d.		

### 3.1.4.8. Teisiniai reikalavimai

<b>Reikalavimas #:</b>	R33	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Sistema nepažeidžia Lietuvos valstybės įstatymų.			
<b>Pagrindimas:</b>	Visi sistemos atliekami veiksmai yra teisėti, neprieštarauja LR įstatymams.			
<b>Šaltinis:</b>	Vartotojas.			
<b>Tinkamumo kriterijus:</b>	Sistema nėra naudojama neteisėtai arba uždraustų (nepageidaujamų, teroristinių) organizacijų viduje.			
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. kovo 15d.			

### 3.1.5. Projekto išeiga

#### 3.1.5.1. Atviri klausimai

Nėra.

#### 3.1.5.2. Egzistuojantys sprendimai

##### Pagamintos sistemos, kurios gali būti nupirktos

Nei viena iš šiuo metu egzistuojančių panašių sistemų (arba jų programinių komponentų) nebus taikoma kuriamame produkte kaip pakartotinio panaudojimo komponentė.

##### Pagaminti komponentai, kurie gali būti panaudoti

Tokių nebus.

##### Galimas pakartotinas panaudojimas

Pakartotino panaudojimo nebus, produktas kuriamas naujai.

#### 3.1.5.3. Naujos problemos

##### Problemos diegimo aplinkai

Jei kompiuterio techninės charakteristikos atitinka prieš tai minėtus reikalavimus, problemų diegimo aplinkai neturėtų būti.

##### Įtaka jau instaliuotoms sistemoms



Kuriama sistema neturi jokios įtakos jau instaliuotoms sistemoms.

#### **Neigiamas vartotojų nusiteikimas**

Vartotojų neigiamo nusiteikimo nėra.

#### **Kliudantys diegimo aplinkos apribojimai**

Nėra.

#### **Galimos naujos sistemos sukeltos problemos**

Nėra.

#### **3.1.5.4. Uždaviniai**

##### **Sistemos pateikimo žingsniai (etapai)**

1. Reikalavimų nustatymas:
  - poreikių surinkimas;
  - poreikių specifikuojimas.
2. Analizė:
  - reikalavimų sistemai specifikuojimas.
3. Projektavimas:
  - architektūros specifikuojimas;
  - detalus architektūros specifikuojimas.
4. Realizavimas:
  - funkcijų hierarchijos sudarymo modulis;
  - veiksmų, formuojančių rezultatus, diagramos modulis;
  - funkcinių modulių realizacija.
5. Testavimas:
  - modulių testavimas;
  - bendras sistemos testavimas.
6. Diegimas ir eksploatavimas.
7. Vartotojo apmokymas.

##### **Vystymo etapai**

Vektorizavimo sistemos vystymo etapai:

- BMP formato failo konvertavimo į sistemos darbinį failą operacijos;
- Vaizdo redagavimo priemonių vystymas;
- Automatinio vektorizavimo operacijos;
- Interaktyvaus redagavimo priemonių vystymas;
- Failo eksportavimo operacijos.

### 3.1.5.5. Pritaikymas

#### Reikalavimai esamų duomenų perkėlimui

Jokių duomenų perkelti nereikia.

#### Reikalingas duomenų transformavimas perkeliant į naują sistemą

Duomenų transformavimas nereikalingas.

### 3.1.5.6. Rizikos

#### Galimos sistemos kūrimo rizikos

4 lentelė. Galimos sistemos kūrimo rizikos

<i>Rizikos faktorius</i>	<i>Tikimybinis įvertinimas</i>
Projekte dalyvaujančio personalo patirtis ir sugebėjimai	8
Algoritmų ir metodų naudojimas pirmą kartą	10
Vartotojo reikalavimų pasikeitimai	6

#### Atsitiktinumų (rizikų) valdymo planas

Rizikos faktoriai ir numatomi planai problemoms spręsti surašyti 5 lentelėje.

5 lentelė. Rizikos faktoriai ir planai problemoms spręsti.

<i>Rizikos faktorius</i>	<i>Problemos sprendimas</i>
Projekte dalyvaujančio personalo patirtis ir sugebėjimai	Susitarti su vadovu naudoti programinę įrangą, su kuria personalas turi bent minimalios darbo patirties.
Algoritmų ir metodų naudojimas pirmą kartą	Reikia pasistengti, kad naujų algoritmų ir metodų naudojimas kuo įmanoma mažiau vėlintų projektą.
Vartotojo reikalavimų pasikeitimas	Nuolatinis bendravimas su vartotoju, kad reikalavimų pasikeitimai nebūtų netikėti ar kardinalūs.

### 3.1.5.7. Kaina

Atviras klausimas.

### 3.1.5.8. Vartotojo dokumentacija ir apmokymas

- Programos įdiegimo vadovas;
- Vartotojo vadovas (sistemos funkcijos bei galimybės);
- Dažniausiai kylančių problemų sąrašas su aprašymais, kaip jas išspręsti.

### 3.1.5.9. Perspektyviniai reikalavimai

Konfidencialumo reikalavimas: autorizuotas vartotojų naudojimas sistema.

## 3.2. Duomenų struktūra

Sistemos architektūra pateikta remiantis RUP rekomendacijomis ir Rational Architecture Practice gairėmis (7). Visos reikalingos diagramos bus vaizduojamos UML notacija.

Specifiniai reikalavimai ir apribojimai: duomenys sistemai turi būti pateikti korektiškai, kad sistema galėtų juos panaudoti ir nesutriktų jos veikimas.

Projektavimo ir įgyvendinimo strategija: sistema projektuojama pagal RUP metodiką.

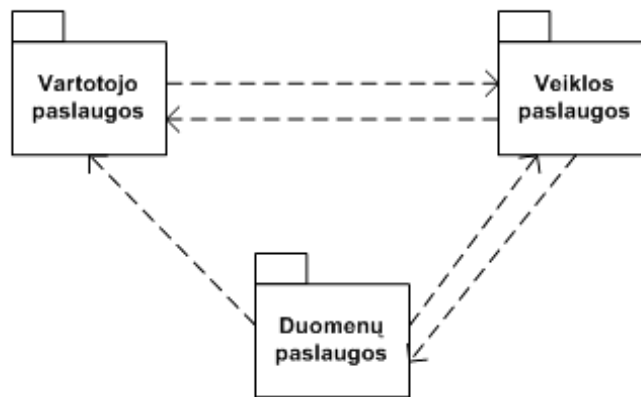
Projektavimo įrankiai: MS Word, MS Visio.

Sistemos panaudojimo atvejų diagrama bei aprašymai pateikti šio darbo 3.1.3 skyriuje. Šiame skyriuje pateikiama loginė sistemos architektūra, jos išskaidymas į paketus bei kiekvieno paketo klasių diagramos – sistemos statinis vaizdas.

### 3.2.1. Sistemos statinis vaizdas (klasių diagramos)

#### 3.2.1.1. Apžvalga

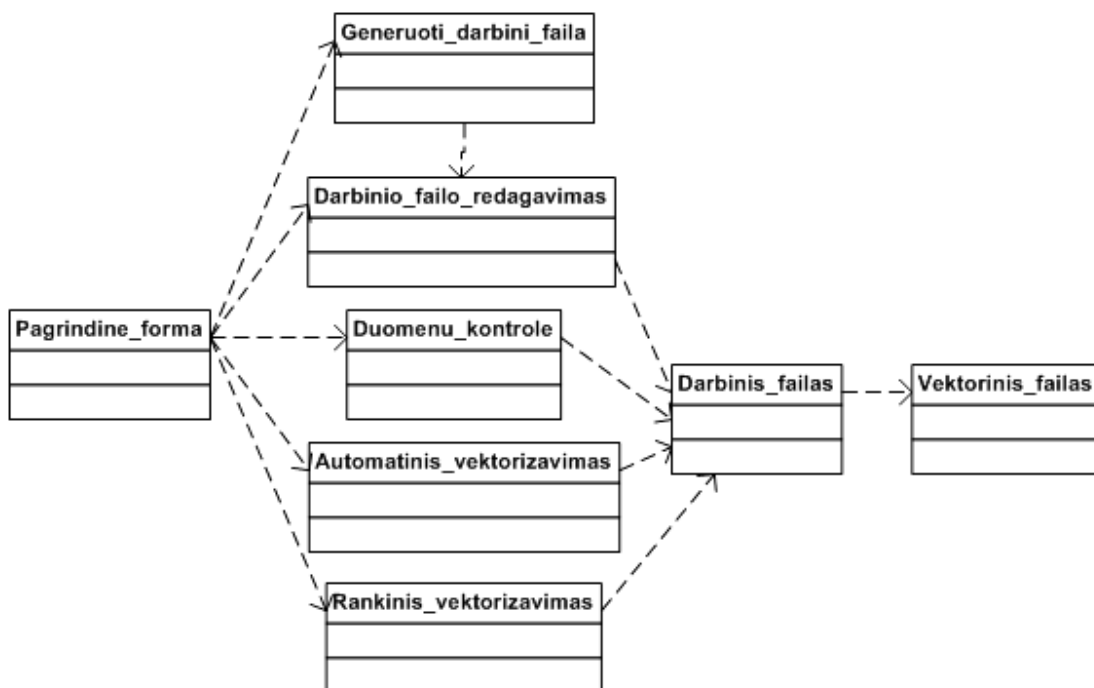
Vektorizavimo sistemai pasirinktas trijų lygių architektūros modelis (žr. 12 pav.).



12 pav. Rastrinio vaizdo vektorizavimo sistemos struktūra.

Kiekvienas paketas atitinka veiklos tipą:

- Vartotojo paslaugų paketas – vartotojo sąsajos projektavimas.
- Veiklos paslaugų paketas – programos (sistemos funkcionalumo) projektavimas.
- Duomenų paslaugų paketas – duomenų struktūrų projektavimas.

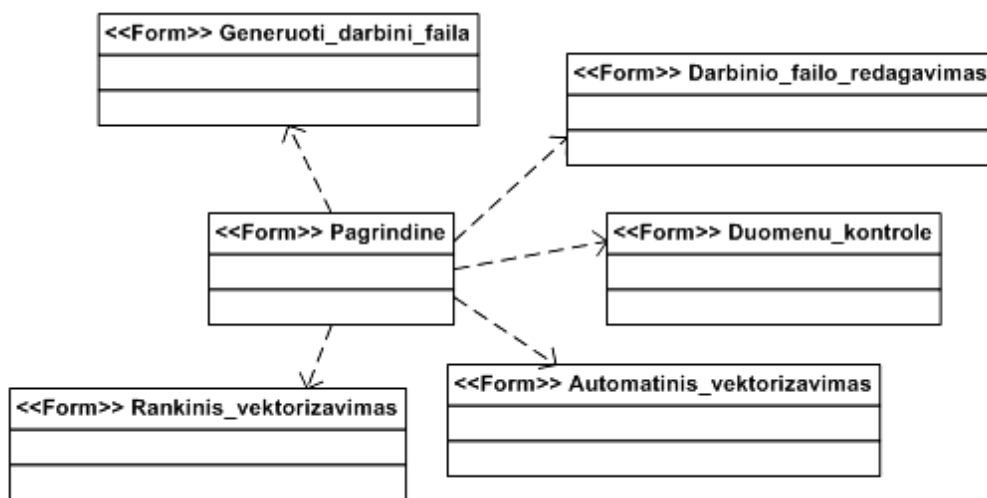


13 pav. Trijų lygių klasių diagrama.

### 3.2.1.2. Paketų detalizavimas

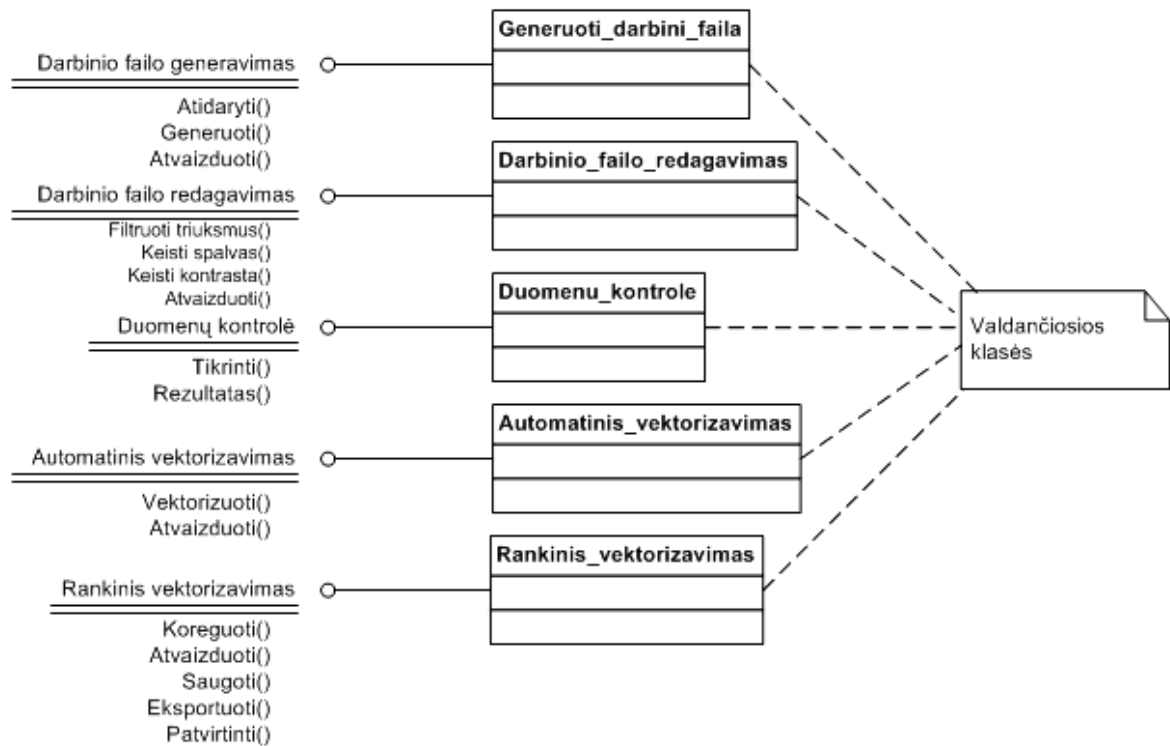
#### Vartotojo paslaugų paketas

Paketas atitinka vartotojo interfeiso projektavimo klases (žr. 14 pav.).



14 pav. Vartotojo interfeiso navigavimo planas.

## Veiklos paslaugų paketas



15 pav. Interfeiso elementai ir juos realizuojančios valdančiosios klasės.

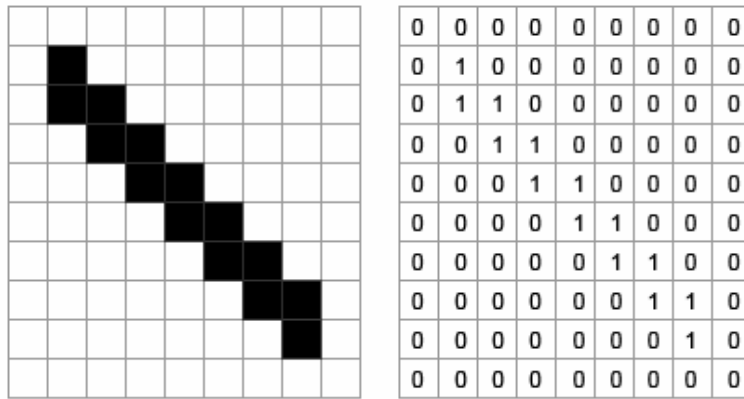
## Duomenų paslaugų paketas

Duomenų bazė sistemoje nenaudojama. Duomenys saugomi failo pavidalu kompiuterio atmintyje: pradinis (bmp\_failas), darbinis (darbinis\_failas) bei galutinis (vektorinis\_failas) išsaugomi kompiuterio kietajame diske. Išsaugojus darbinį failą, bus galima juo pasinaudoti vektorizuojant kitus formatus.

Darbinio failo struktūra – dvimatis masyvas (**atkMas[j][i]**), kurio elementai igyja reikšmes tik **0** (fonas) arba **1** (linijos ir pan.). Kiekvienas masyvo elementas atitinka vieną rastrinio vaizdo tašką. Darbinio failo struktūra parodyta 16 paveiksle.

Darbinio failo struktūros privalumai:

- masyvo elemento indeksai atitinka taško koordinates (**x, y**) vaizde ( t.y. **i = x, j = y**);
- ieškant geometrinių objektų (vektorizavimo metu), nesudėtinga iš anksto paskaičiuoti indeksų kitimo dėsnius.

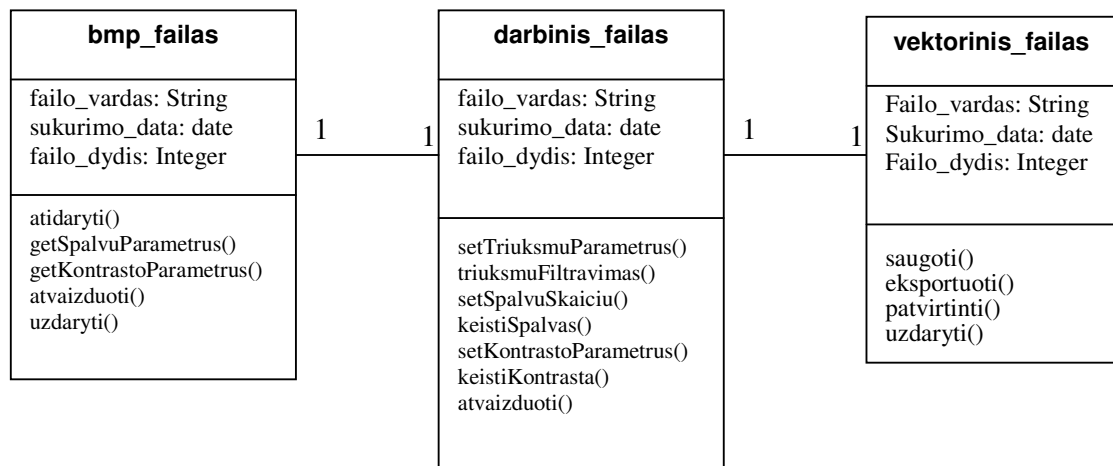


(a)

(b)

16 pav. (a) rastrinis vaizdas kompiuterio ekrane; (b) darbinė duomenų matrica kompiuterio ekrane.  
Stulpeliai x, eilutės y.

Taigi duomenų paslaugų paketas apima klases ir metodus duomenų importavimui (i) bei eksportavimui (iš) sistemos, taip pat duomenų saugojimui kompiuterio pastoviojoje (diskinėje) atmintyje. Duomenų struktūros vaizdas parodytas 17 paveiksle.



17 pav. Duomenų struktūros vaizdas.

### 3.3. Projektuojamos sistemos architektūra

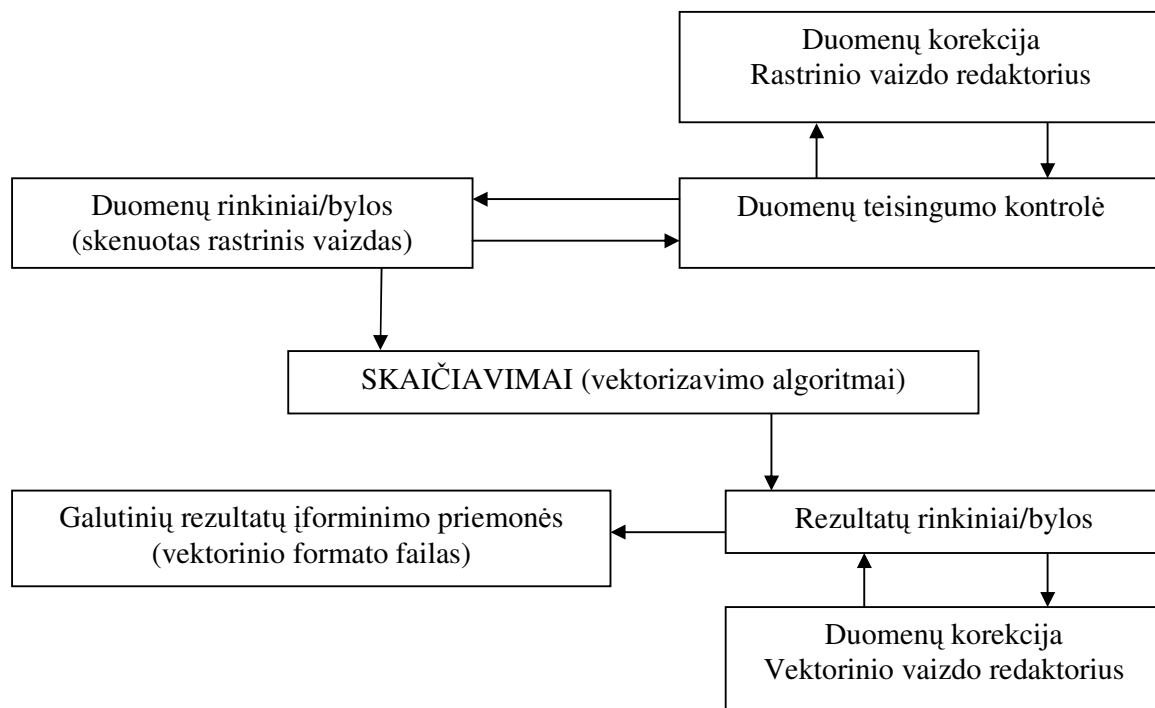
#### 3.3.1. Sistemos modulinė schema

Optimaliausia vektorizavimo išeitis – rankinio ir automatinio vektorizavimo sintezė. Pagrindiniai programinio vektorizavimo produkto funkciniai moduliai:

- rastrinių vaizdų redaktorius;

- automatinio vektorizavimo modulis;
- interaktyvaus gautų rezultatų redagavimo modulis.

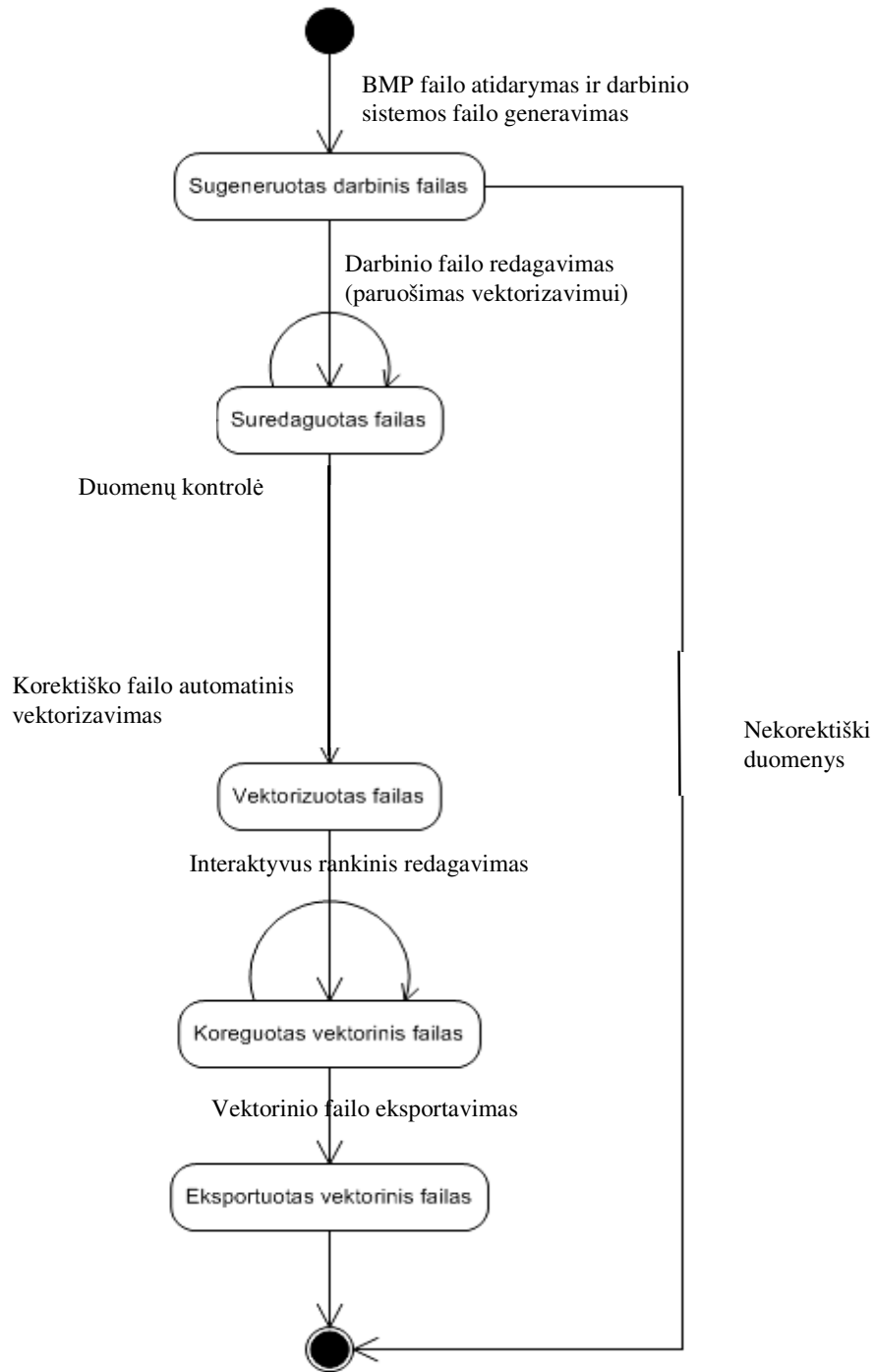
18 paveiksle parodyta vektorizavimo programos modulinė schema.



18 pav. Vektorizavimo programos modulinė schema.

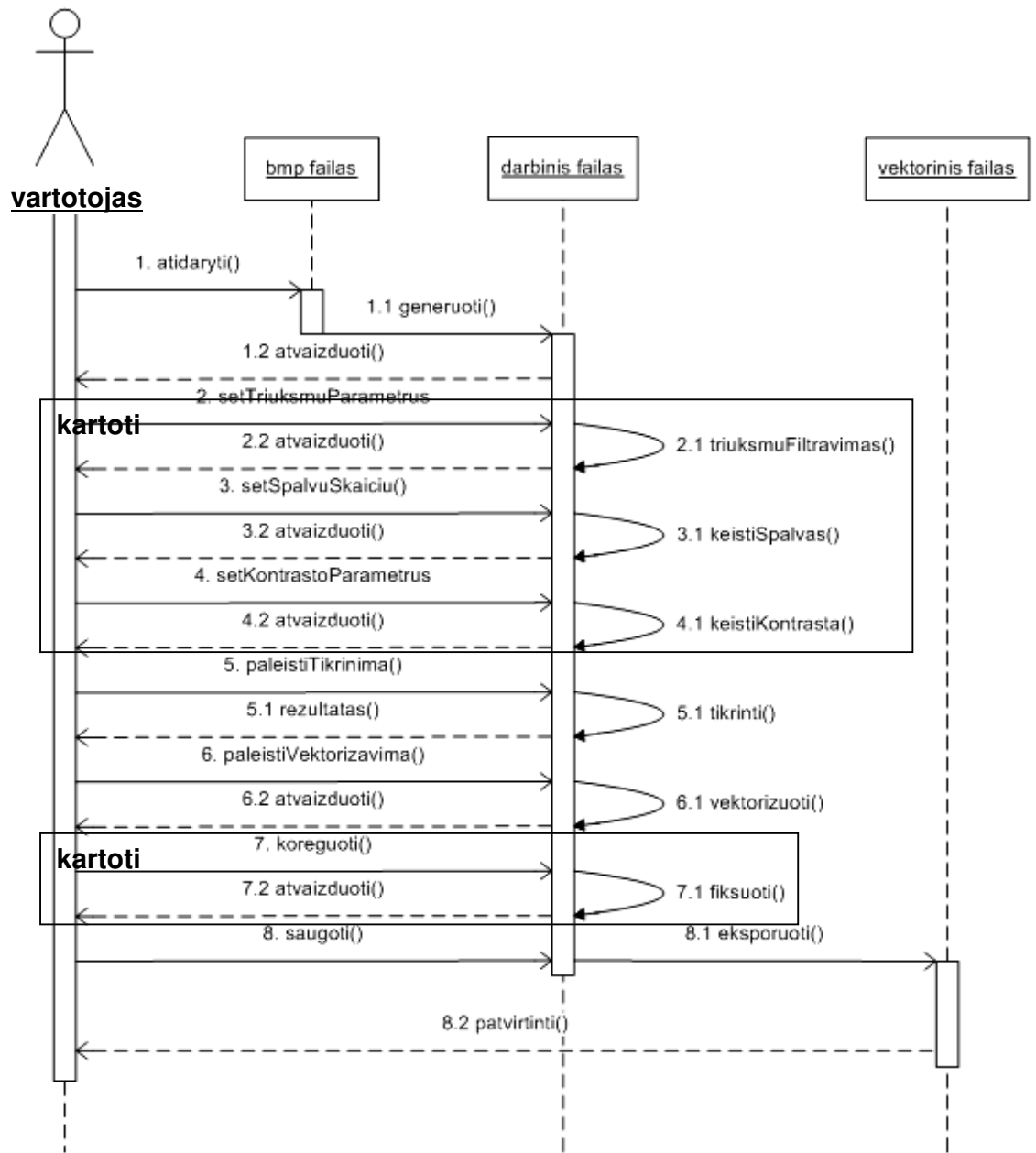
### 3.3.2. Sistemos dinaminis vaizdas

Pagrindinis sistemos objektas yra darbinis failas. Su juo atliekami reikalingi vektorizavimo veiksmai. Jam sudaryta būsenų diagrama pavaizduota 19 paveiksle.

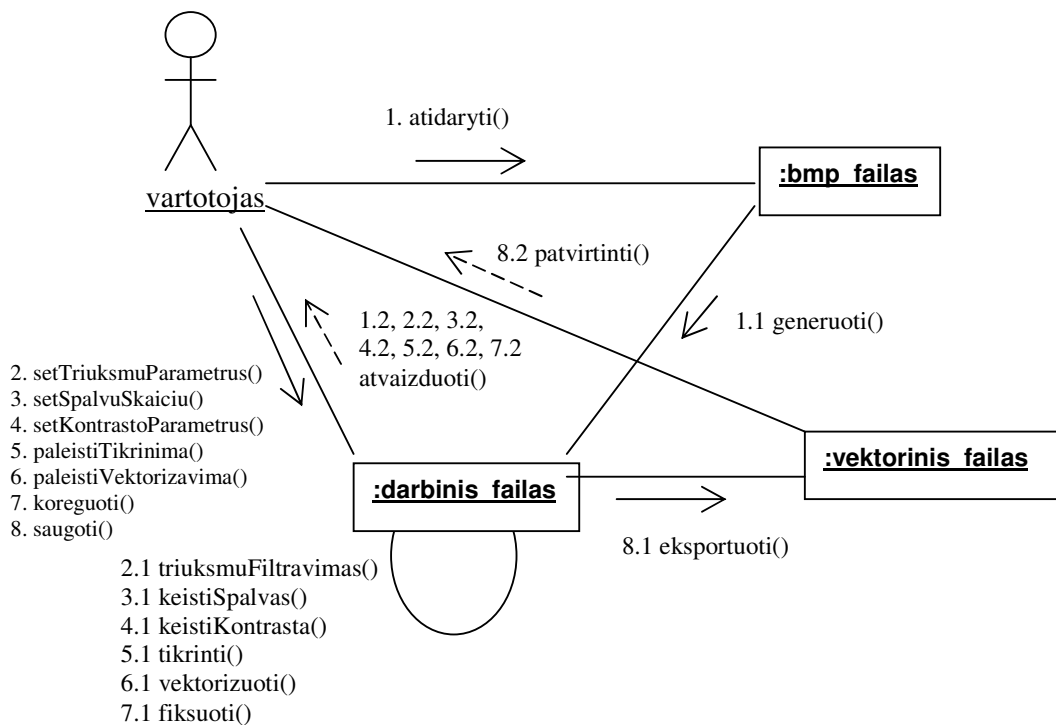


19 pav. Vektorizavimo proceso būsenų diagrama.





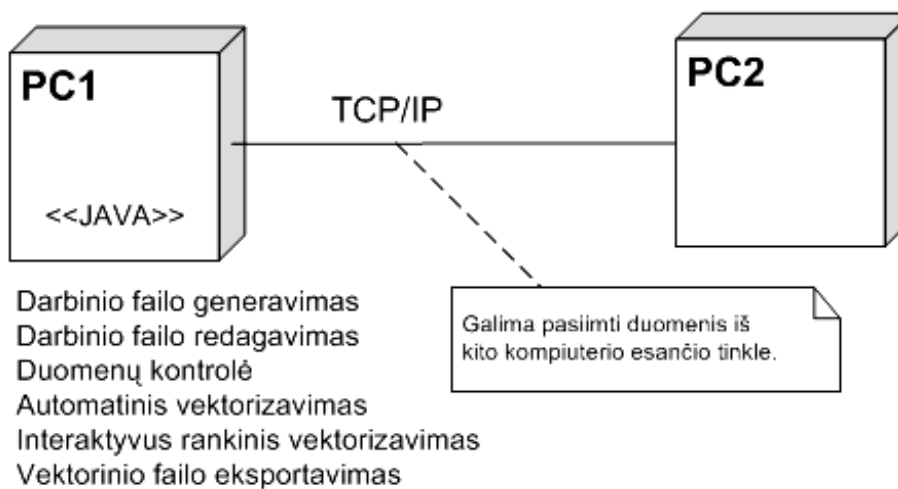
20 pav. Vektorizavimo sistemos sekų diagrama.



21 pav. Vektorizavimo sistemos bendradarbiavimo diagrama.

### 3.3.3. Išdėstymo (deployment) vaizdas

Kuriama sistema nėra paskirstyta. Ji bus įdiegta viename kompiuteryje. Prireikus, duomenys gali būti paimti iš kito kompiuterio, esančio tinkle (žr. 22 pav.).



22 pav. Fizinė sistemos išdėstymo diagrama.

Minimalūs reikalavimai kompiuteriui, į kurį bus įdiegta sistema:

- Procesorius: ne mažiau 2GHz;
- Operatyvinė atmintis: ne mažiau 512MB;
- *Windows 98/2000/XP* operacinė sistema.

### **3.4. Programinių modulių ar objektų specifikacijos**

#### **3.4.1. Vartotojo sąsajos komponentas**

##### **Klasifikacija**

Paketas.

##### **Apibrėžimas**

Vartotojo sąsaja – kuriamos sistemos modulis, tarpininkas tarp sistemos ir jos vartotojo. Jį sudaro sąsajos klasės.

Vartotojo sąsajos komponentas – tai įvedimo formų, meniu, įrankių bei rezultatų vizualizavimo klasės, kurių pagalba sistemos vartotojas gali valdyti sistemos veiklos eigą ir gauti atitinkamus rezultatus (t.y.: vartotojas pasirenka norimus veiksmus, įveda duomenis į sistemą, gauna iš sistemos vizualų veiklos rezultatą ir pan.).

##### **Atsakomybės**

Vartotojo sąsajos komponentas skirtas užtikrinti, kad sistemos vartotojas galėtų lengvai pasirinkti norimus veiksmus su sistema: įvestų reikiamus duomenis, gautų atitinkamus rezultatus, vizualizuotus kompiuterio ekrane. Turi būti intuityviai suprantama ir draugiška vartotojo sąsaja. Tam bus naudojama Java kalba. Duomenų įvedimas bei rezultatų atvaizdavimas naudojant atitinkamas formas bei dialogo langus.

##### **Apribojimai**

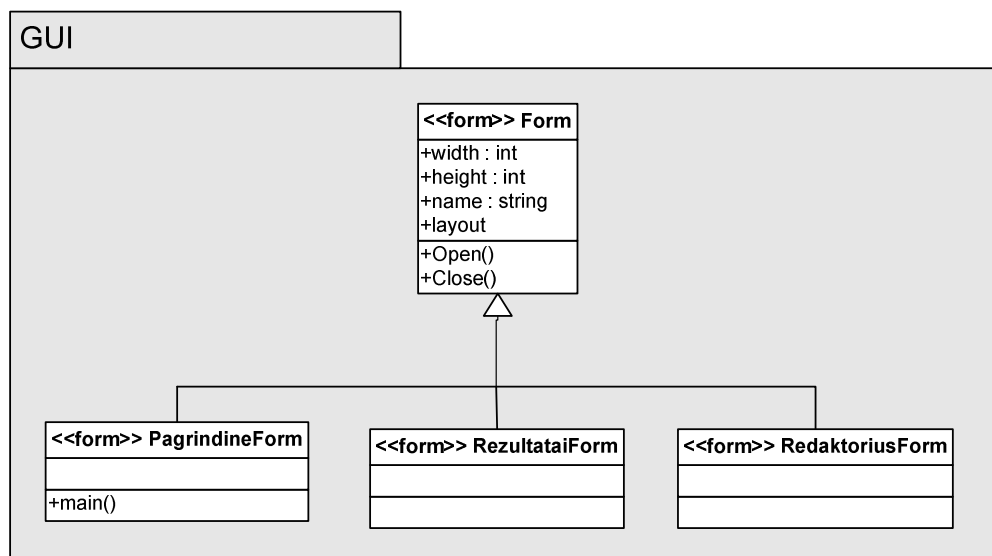
Turi turėti ryšį su veiklos paslaugų komponentu.

Turi užtikrinti apdorojamų duomenų korektiškumą.

Turi apsaugoti sistemos vartotoją nuo atsitiktinių klaidų atliekant svarbius veiksmus.

##### **Struktūra**

Vartotojo sąsajos struktūra pateikiama 23 paveiksle vartotojo sąsajos klasių diagrama.



23 pav. Vartotojo sąsajos klasių diagrama

6 lentelė. Klasė: PagrindineForm

<b>Komponentas</b>	<b>PagrindineForm</b>
<b>Klasifikacija</b>	Klasė.
<b>Apibrėžimas</b>	Pagrindinė vektorizavimo sistemos klasė turinti main() metodą, kurio pagalba paleidžiama sistema – atvaizduojama pagrindinė forma. Forma tarsi konteineris kitiems sistemos elementams (menui, įrankiams, rezultatų formai ir kt.) išdėstyti.
<b>Atsakomybės</b>	Turi užtikrinti ryšį su kitais formoje išdėstytais elementais. Elementai turi būti aiškiai ir tvarkingai išdėstyti, kad būtų suprantami vartotojui. Intuityvi ir draugiška aplinka. Tam bus naudojama Java programavimo kalba.
<b>Apribojimai</b>	Vartotojo sąsaja turi atitikti srities reikalavimus (specialūs simboliai, žymėjimai, kt.).
<b>Struktūra</b>	Šio komponento subkomponentai yra menui (pagrindinių sistemos funkcijų iškvietimui).
<b>Sąveikavimas</b>	Sąveikauja su kitomis vartotojo sąsajos klasėmis (RezultataiForm bei RedaktoriusForm), tai pat su veiklos paslaugų paketo klasėmis, duomenų struktūrų paketu bei sisteminiais JAVA paketais.

7 lentelė. Klasė: RezultataiForm

<b>Komponentas</b>	<b>RezultataiForm</b>
<b>Klasifikacija</b>	Klasė.
<b>Apibrėžimas</b>	Tai klasė reikalinga veiklos rezultatų vizualizavimui kompiuterio ekrane. Rezultatai (grafiniai duomenys) atvaizduojami atskiroje formoje su grafiniu lauku.
<b>Atsakomybės</b>	Turi užtikrinti korektišką duomenų atvaizdavimą.
<b>Apribojimai</b>	Nėra.
<b>Struktūra</b>	Šio komponento subkomponentas – grafinis laukas.
<b>Sąveikavimas</b>	Sąveikauja su kitomis vartotojo sąsajos klasėmis (PagrindineForm bei RedaktoriusForm), taip pat su veiklos paslaugų paketo klasėmis, duomenų struktūrų paketu bei sisteminiais JAVA paketais.

8 lentelė. Klasė: RedaktoriusForm

<b>Komponentas</b>	<b>RedaktoriusForm</b>
<b>Klasifikacija</b>	Klasė.

<b>Apibrėžimas</b>	Tai forma, išskviečiama iš PagrindineForm norint rankiniu būdu koreguoti vektorinį failą. Tai tarsi mini grafikos redaktorius (programa programoje). Šio darbo metu plačiau ji nebus detalizuojama.
<b>Atsakomybės</b>	Turi užtikrinti kokybišką interaktyvų ryšį tarp vartotojo ir atitinkamų veiklos paslaugų bei duomenų struktūrų paketų klasių. Greitas ir korektiškas duomenų atvaizdavimas.
<b>Apribojimai</b>	Turi atitikti srities reikalavimus (specialūs simboliai, žymėjimai, kt.).
<b>Struktūra</b>	Komponento subkomponentai yra menu (pagalbinėms sistemos funkcijoms iškvieti), grafikos laukas (rezultatams atvaizduoti).
<b>Sąveikavimas</b>	Sąveikauja su kitomis vartotojo sąsajos klasėmis (RezultataiForm bei PagrindineForm), taip pat su veiklos paslaugų paketo klasėmis, duomenų struktūrų paketu bei sisteminiais JAVA paketais.

### **Sąveikavimas**

Vartotojo sąsajos komponentas sąveikauja su veiklos taisyklių bei duomenų struktūrų komponentu. Visi trys komponentai yra vartotojo darbo kompiuteryje (sistema nėra paskirstyta).

### **Resursai**

Vartotojo sąsajos komponentas saugomas konkrečiame kompiuteryje ir yra neatsiejama bendros vektorizavimo sistemos dalis, kuri veikia Windows aplinkoje. Reikalingi tokie resursai kaip: atmintis, pradinis duomenų failas.

### **Skaičiavimai**

Vartotojo sąsajoj skaičiavimai neatliekami.

### **Sąsaja**

Vartotojo sąsajoj įvesti duomenys ar atlikti veiksmai tiesiogiai perduodami veiklos paslaugų klasėms. Vartotojo sąsają sudaro šios formos:

- PagrindineForm
- RezultataiForm
- RedaktoriusForm

Pagrindinėje formoje (PagrindineForm) per pagrindinių sistemos funkcijų menu dar galima pasiekti šiuos dialogo langus:

- FiltravimoParametraiDialog
- SpalvuSkaiciausDialog
- KontrastoParametraiDialog
- VektorizavimoParametraiDialog
- EksportoDialog

Jie naudojami atitinkamų parametrų perdavimui į sistemą (veiklos paketo klasėms), kurie reikalingi tam tikriems skaičiavimams atlikti.

### 3.4.2. Veiklos paslaugų komponentas

#### Klasifikacija

Paketas.

#### Apibrėžimas

Veiklos paslaugų komponentas realizuoja pagrindines sistemos funkcijas, tokias kaip: darbinio failo generavimas, vaizdo redagavimas, duomenų kontrolė, automatinis vektorizavimas, rankinis vektorizavimas bei galutinio failo eksportavimas. Tai yra tarpinis komponentas tarp vartotojo sąsajos ir duomenų struktūrų. Duomenų bazė sistemoje nenaudojama. Duomenys saugomi atitinkamų failų pavidalu kompiuterio atmintyje.

#### Atsakomybės

Veiklos paslaugų komponento pagrindinės funkcijos – priimti sistemos vartotojo įvestus duomenis, atlikti skaičiavimus, pateikti rezultatus atgal vartotojui, galutinio rezultato išsaugojimas.

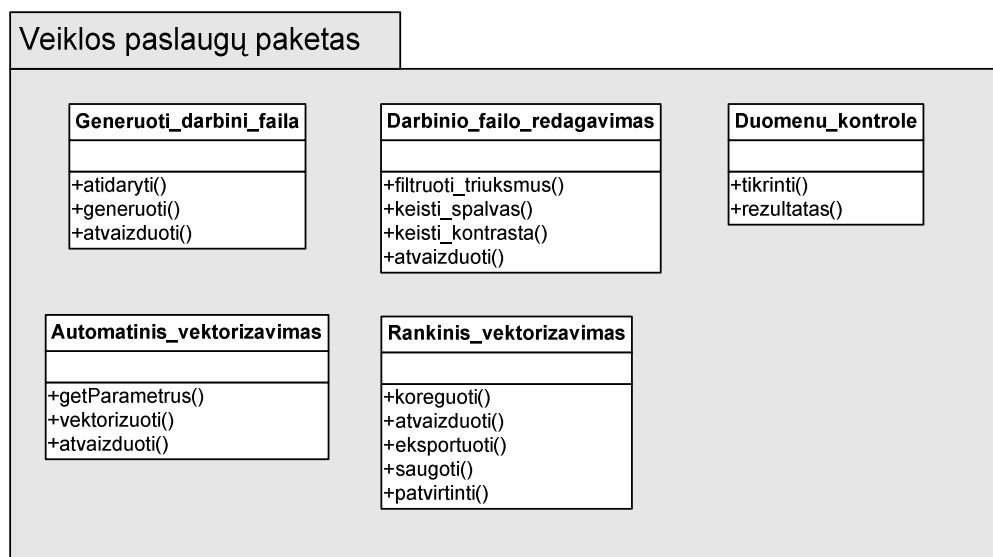
#### Apribojimai

Turi būti užtikrinamas apdorojamų duomenų korektiškumas.

Turi būti sąsaja su vartotojo sąsajos bei duomenų struktūrų komponentais.

#### Struktūra

Veiklos paslaugų komponento struktūra pavaizduota veiklos paslaugų klasių diagrama. Kiekviena sistemos funkcija realizuota atskira klase (24 pav.), kuri laikantis tam tikro eiliškumo iškviečiama vartotojo per vartotojo sąsają. Sistemos veiklos paslaugų klasės keičia turimų duomenų struktūrą, o tai ir yra kuriamos sistemos pagrindinis tikslas – rastrinio vaizdo failo formatą paversti į vektorinį.



24 pav. Veiklos paslaugų paketo klasės.

9 lentelė. Klasė: *Generuoti\_darbini\_faila*

<b>Komponentas</b>	<b>Generuoti_darbini_faila</b>
<b>Klasifikacija</b>	Klasė.
<b>Apibrėžimas</b>	Darbinio failo generavimo klasė. Iš BMP, JPG ar GIF formato failo išrenkanti reikalingus duomenis ir sudėliojanti juos taip (į savo darbinį failą), kad po to būtų galima juos dar koreguoti ir svarbiausia vektorizuoti.
<b>Atsakomybės</b>	Atsakinga už rastrinio failo atidarymą, darbinio failo generavimą bei atvaizdavimą RezultataiForm formoje.
<b>Apribojimai</b>	Gali atidaryti ir generuoti į darbinį tik BMP formato failą.
<b>Struktūra</b>	Pagrindiniai klasės metodai: atidaryti(); generuoti(); atvaizduoti().
<b>Sąveikavimas</b>	Sąveikauja su kitomis vartotojo sąsajos klasėmis, duomenų struktūrų paketu bei sisteminiiais JAVA paketais.

10 lentelė. Klasė: *Darbinio\_failo\_redagavimas*

<b>Komponentas</b>	<b>Darbinio_failo_redagavimas</b>
<b>Klasifikacija</b>	Klasė.
<b>Apibrėžimas</b>	Tai klasė apimanti metodus leidžiančius pageduoti darbinį failą prieš atliekant jo automatinį vektorizavimą. Taip pasiekiami geresni vektorizavimo rezultatai.
<b>Atsakomybės</b>	Turi leisti vartotojui nurodyti filtravimo, spalvų bei kontrasto parametrus per atitinkamus dialogo su sistema langus.
<b>Apribojimai</b>	Neturi leisti vartotojui įvesti netinkamas reikšmes.
<b>Struktūra</b>	Pagrindiniai klasės metodai: filtruoti_triuksmus(); keisti_spalvas(); keisti_kontrasta(); atvaizduoti().
<b>Sąveikavimas</b>	Sąveikauja su kitomis vartotojo sąsajos klasėmis, duomenų struktūrų paketu bei sisteminiiais JAVA paketais.

11 lentelė. Klasė: *Duomenu\_kontrolė*

<b>Komponentas</b>	<b>Duomenu_kontrolė</b>
<b>Klasifikacija</b>	Klasė.
<b>Apibrėžimas</b>	Duomenų kontrolės klasė, t.y. klasė, kuri patikrina ar turimas darbinis failas tinkamas vektorizuoti automatiškai.
<b>Atsakomybės</b>	Turi patikrinti duomenis ir apie rezultatus pranešti sistemos vartotojui (išvedamas tekstinis pranešimas į kompiuterio ekraną).
<b>Apribojimai</b>	Nėra.
<b>Struktūra</b>	Pagrindiniai klasės metodai: tikrinti(); rezultatas().
<b>Sąveikavimas</b>	Sąveikauja su kitomis vartotojo sąsajos klasėmis, duomenų struktūrų paketu bei sisteminiiais JAVA paketais.

12 lentelė. Klasė: *Automatinis\_vektorizavimas*

<b>Komponentas</b>	<b>Automatinis_vektorizavimas</b>
<b>Klasifikacija</b>	Klasė.
<b>Apibrėžimas</b>	Pagrindinę sistemos funkciją realizuojanti klasė, kuri vektorizuoja darbinį sistemos failą.
<b>Atsakomybės</b>	Turi vektorizuoti ir korektiškai atvaizduoti gautus vektorizavimo rezultatus.
<b>Apribojimai</b>	Gali vektorizuoti tik paruoštą korektišką darbinį sistemos failą.

<b>Struktūra</b>	Pagrindiniai klasės metodai: getParametrus(); vektorizuoti(); atvaizduoti(). Detaliau ši klasė bei vektorizavimo algoritmas aprašytas šio darbo 3.4.4 skyriuje.
<b>Sąveikavimas</b>	Sąveikauja su kitomis vartotojo sąsajos klasėmis, duomenų struktūrų paketu bei sisteminiais JAVA paketais.

13 lentelė. Klasė: Rankinis\_vektorizavimas

<b>Komponentas</b>	<b>Rankinis_vektorizavimas</b>
<b>Klasifikacija</b>	Klasė.
<b>Apibrėžimas</b>	Klasė atsakinga už vektorinio vaizdo koregavimą rankiniu būdu, tiesiai kompiuterio ekrane.
<b>Atsakomybės</b>	Galimybė interaktyviai koreguoti gautą vektorinį vaizdą. Po to jį išsaugoti, eksportuoti.
<b>Apribojimai</b>	Vartotojo veiksmų rezultatai turi būti iškart matomi kompiuterio ekrane.
<b>Struktūra</b>	Pagrindiniai klasės metodai: koreguoti(); atvaizduoti(); eksportuoti(); saugoti(); patvirtinti().
<b>Sąveikavimas</b>	Sąveikauja su kitomis vartotojo sąsajos klasėmis, duomenų struktūrų paketu bei sisteminiais JAVA paketais.

### Sąveikavimas

Veiklos paslaugų eigą (atitinkamas funkcijas/paslaugas) per vartotojo sąsają tiesiogiai įtakoja sistemos vartotojas. Darbas su sistema vyksta interaktyviai.

Veiklos paslaugų komponentas sąveikauja su vartotojo sąsajos bei duomenų struktūrų komponentais.

### Resursai

Duomenų bei darbinis failai, taip pat vartotojo nurodyti parametrai reikalingi atitinkamiems skaičiavimams, kompiuterio atmintis.

### Skaičiavimai

Automatinio vektorizavimo skaičiavimai bei algoritmas detalizuotas šio darbo 3.4.4 skyriuje.

### Sąsaja

Visos pagrindinių sistemos funkcijų realizavimo klasės iškviečiamos per vartotojo sąsajos PagrindineForm meniu.

### 3.4.3. Duomenų struktūrų komponentas

#### Klasifikacija

Duomenų failai.



## Apibrėžimas

Duomenų struktūrų komponentas skirtas visų sistemoje naudojamų duomenų saugojimui ir tvarkymui.

## Atsakomybės

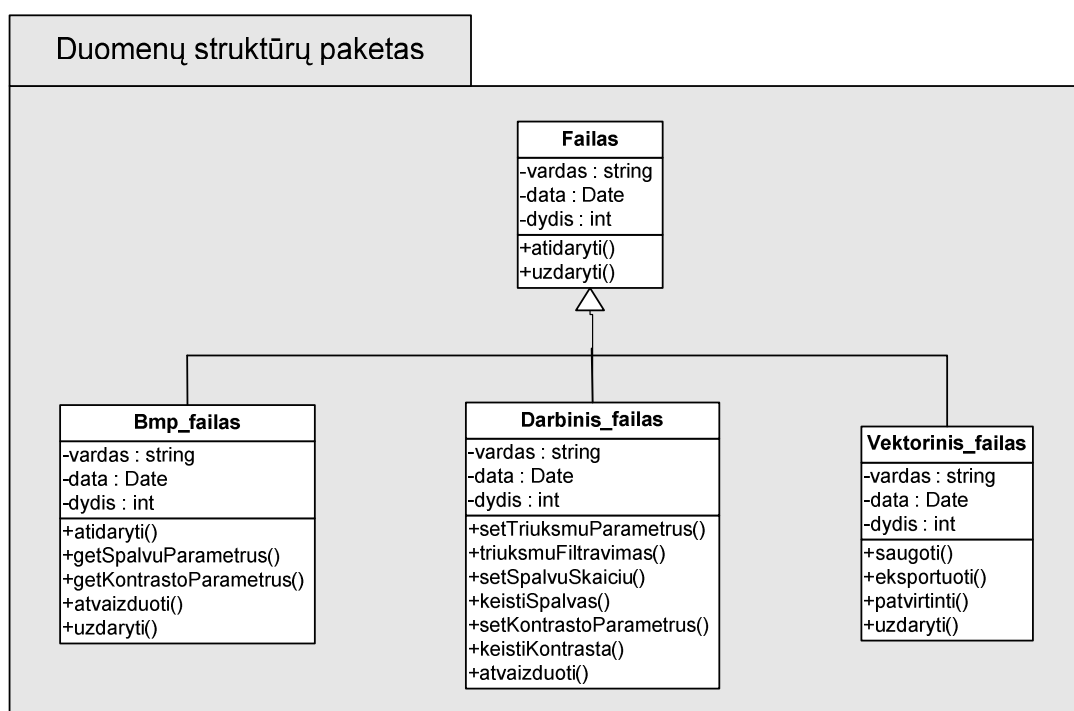
Pagrindinė šio komponento funkcija – duomenų saugojimas. Duomenys saugomi failuose.

## Apribojimai

Pradinis duomenų failas gali būti BMP, JPG arba GIF formato. Sistema generuoja tik jai būdingą darbinį failą. Rezultatas – mūsų siūlomas vektorinio formato failas.

## Struktūra

Komponento struktūra pateikiama 25 paveiksle.



25 pav. Duomenų struktūrų paketas (klasių diagrama).

14 lentelė. Failas: BMP\_failas

<b>Komponentas</b>	<b>Bmp_failas</b>
<b>Klasifikacija</b>	Failas.
<b>Apibrėžimas</b>	BMP formato failas, kurį vartotojas norėtų vektorizuoti.
<b>Atsakomybės</b>	Kad duomenys faile korektiški.
<b>Apribojimai</b>	Nėra.
<b>Struktūra</b>	Pagrindiniai failo atributai bei veiksmai, kuriuos galima su juo atlikti, pavaizduoti 24 paveiksle.
<b>Sąveikavimas</b>	Sąveikauja su vartotojo sąsajos bei veiklos paslaugų klasėmis, taip pat su sisteminiiais JAVA paketais.

<b>Komponentas</b>	<b>Darbinis_failas</b>
<b>Klasifikacija</b>	Failas.
<b>Apibrėžimas</b>	Darbinis sistemos failas.
<b>Atsakomybės</b>	Turi būti korektiškas. Galimybė keisti tam tikrus jo atributus (redaguoti) prieš automatinį vektorizavimą.
<b>Apribojimai</b>	Tai specifinė duomenų struktūra naudojama tik šioje sistemoje – duomenų matrica (dvimatis masyvas).
<b>Struktūra</b>	Pagrindiniai failo atributai bei veiksmai, kuriuos galima su juo atlikti, pavaizduoti 24 paveiksle.
<b>Sąveikavimas</b>	Sąveikauja su kitomis vartotojo sąsajos klasėmis, duomenų struktūrų paketu bei sisteminiiais JAVA paketais.

<b>Komponentas</b>	<b>Vektorinis_failas</b>
<b>Klasifikacija</b>	Failas.
<b>Apibrėžimas</b>	Vektorizuotas darbinis failas. Galutinis sistemos veiklos produktas.
<b>Atsakomybės</b>	Turi būti korektiškas. Turi būti eksportuotas iš sistemos ir išsaugotas kompiuterio atmintyje.
<b>Apribojimai</b>	Nėra.
<b>Struktūra</b>	Pagrindiniai failo atributai bei veiksmai, kuriuos galima su juo atlikti, pavaizduoti 24 paveiksle. Failo vidinė struktūra: surastų geometrinių primityvų (atkarpų, apskritimų, lankų) atributų sąrašas (koordinatės, linijos storis ir pan.).
<b>Sąveikavimas</b>	Sąveikauja su vartotojo sąsajos bei veiklos paslaugų klasėmis, taip pat su sisteminiiais JAVA paketais.

### **Sąveikavimas**

Duomenų struktūrų komponentas sąveikauja su veiklos paslaugų bei vartotojo sąsajos komponentu rezultatų vizualizavimui.

### **Resursai**

Failai saugomi kompiuterio atmintyje.

### **Skaičiavimai**

Duomenų struktūrų komponente skaičiavimai neatliekami. Failai yra naudojami tik rezultatų išsaugojimui.

### **Sąsaja**

Sąsaja su veiklos paslaugų sluoksniu (kaip resursai skaičiavimams) bei vartotojo sąsaja (kaip resursai rezultatų atvaizdavimui).

### **3.4.4. Automatinio vektorizavimo algoritmas**

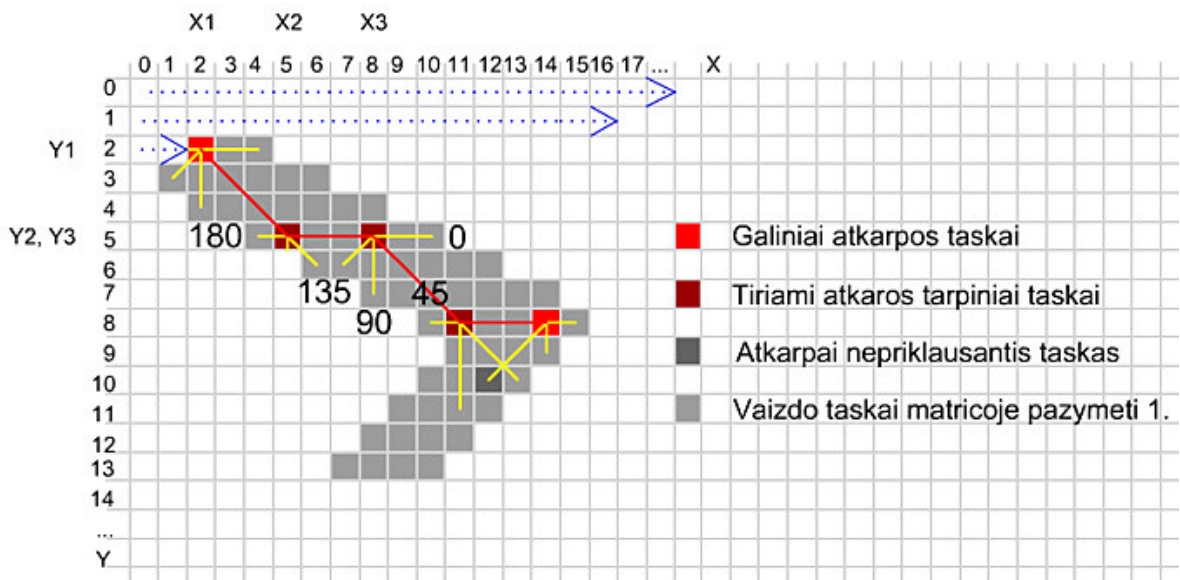
Šiame skyriuje detaliai aprašytas mūsų pasiūlytas integruotas linijų sekos, linijos storio nustatymo, bei linijų aproksimavimo algoritmas, toliau bus vadinamas tiesiog automatinio vektorizavimo algoritmu.

Taigi, į sistemą importuotas vaizdas verčiamas į **0** ir **1** rinkinį – matricą, dvimatis masyvas (kitai dar vadintas - darbinis sistemos failas, visi veiksmai atliekami analizuojant jo duomenis). **0** reiškia foną, jame vaizdo nėra (arba balta spalva), **1** - taškas priklauso kažkokiam objektui (tiesei ar apskritimui – juoda spalva). Matricos koordinatinių pradžia (**0, 0**) yra kairiajame viršutiniame kampe. **X** teigiama kryptis į dešinę, o **Y** – į apačią (žr. 26 pav.). Vaikščiodami po matricos elementus žinome kiekvieno taško, kuriame esame, koordinates, tad esmė – iš tiesės lygties (einančios per du žinomus taškus) susirasti krypties koeficientą bei pasvirimo kampą. Taigi „vaikštoma“ po duomenų matricą, tiriami jos taškai - analizuojamas vaizdas, iš jo išgaunama reikalinga informacija ir išsaugoma priimtina forma.

Automatinio vektorizavimo algoritmo pagrindiniai žingsniai:

1. Keliaujama matricos (sudarytos tik iš **0** ir **1**) eilutėmis ir stulpeliais (žr. 26 pav.), kol sutinkamas **1**. Fiksuojamos jo koordinatės (**x1**, **y1**) kaip atkarpos pradžia, įrašoma į **atkMas[][]**.

2. Iš to taško paleidžiami spinduliai 5 kryptim (**0°**, **45°**, **90°**, **135°** ir **180°** laipsnių kampais), kaip parodyta 26 paveiksle, kol sutinkamas **0** arba kol neviršijama **tH** slenkstinė reikšmė, kurią turi nurodyti vartotojas vektorizavimo paleidimo metu. 26 paveiksle **tH = 3**.



26 pav. Vektorizavimo algoritmas: spindulių sklaidimas 5-iomis kryptimis.

3. Surandamas ilgiausias iš spindulių, fiksuojamos jo galo koordinatės (**x2**, **y2**) (naujas taškas).

4. Iš tiesės lygties (kai žinomos atkarpos galų koordinatės) apskaičiuojame jos pasvirimo kampą  $\mathbf{a0} = \arctg(\mathbf{k})$ , kur **k** – krypties koeficientas,  $\mathbf{k} = (\mathbf{y2} - \mathbf{y1})/(\mathbf{x2} - \mathbf{x1})$ . Kad

išvengi dalybos iš 0 (kai  $x_2 = x_1$ ), iškart kampui  $a_0$  priskiriama reikšmė  $90^\circ$ . Be to, jei gaunamas neigiamas kampas, vadinasi tiesė pasvirusi kampu didesniu už  $90^\circ$ , todėl jį perskaičiuojame į teigiamą (skaičiavimų patogumo dėlei):  $a_0 = 180 - |a_0|$ .

5. Toliau iš naujo taško vėl paleidžiami 5 spinduliai, ieškomas ilgiausias, fiksuojamos jo koordinatės  $(x_3, y_3)$  (2 naujas taškas), randamas kampas  $a_1 = \arctg(k)$ , kur  $k$  – krypties koeficientas,  $k = (y_3 - y_1)/(x_3 - x_1)$ . Taikomos visos tos pačios taisyklės kaip ir 4 punkte.

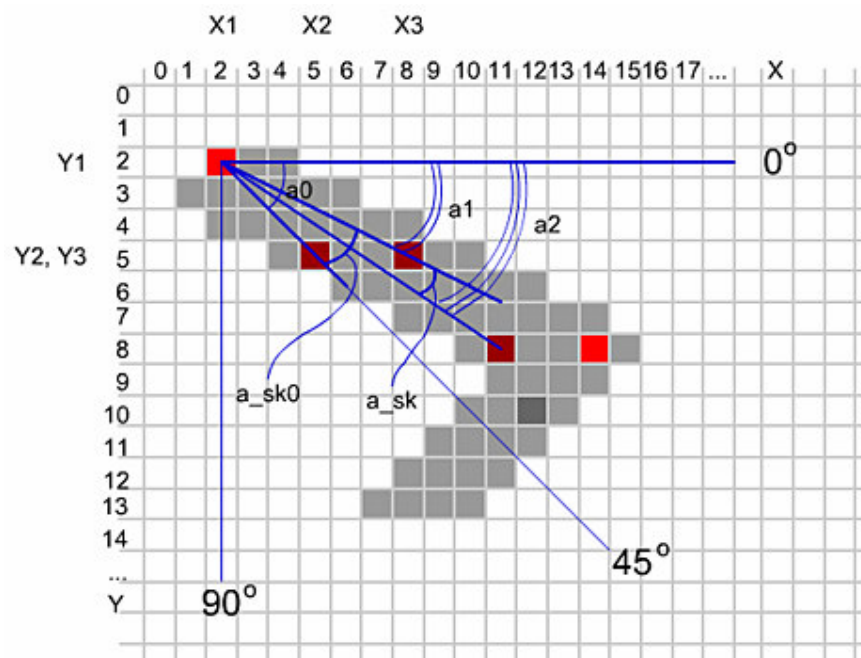
6. Surandamas kampų skirtumas ( $a_{sk0} = |a_1 - a_0|$ ) (žr. 27 pav.).

7. Vėl paleidžiami spinduliai iš taško, randamas  $a_2 = \arctg(k)$ , ieškoma kampų skirtumo ( $a_{sk} = |a_2 - a_1|$ ).

8. Palyginami kampų skirtumai, dabartinis ir prieš tai buvęs.

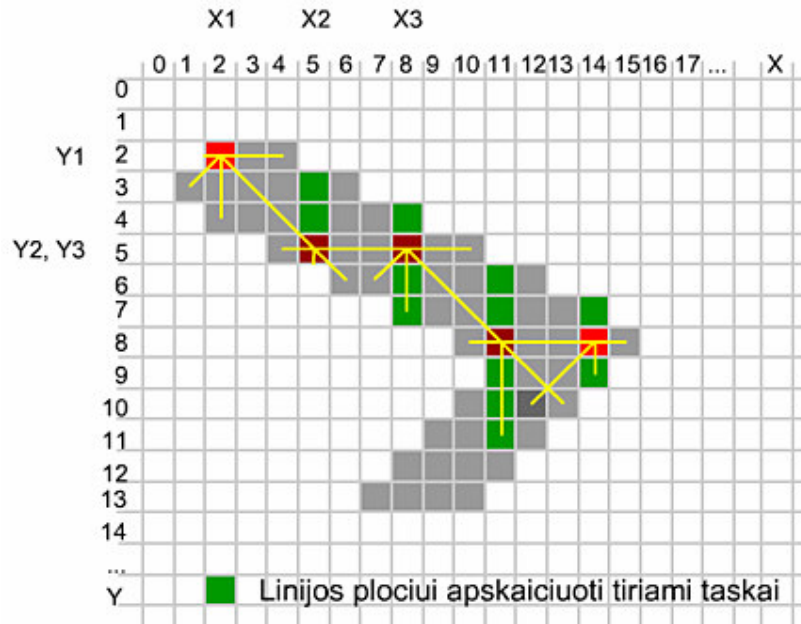
9. Jei dabartinis kampų skirtumas mažesnis arba lygus prieš tai buvusiam, tai daroma prielaida, kad taškas priklauso tai pradėtai atkarpai ir einama toliau (kartojam nuo 7 punkto), o jei ne, vadinasi atkarpa keičia kryptį, taškas atmetamas, o prieš tai buvęs užfiksuojamas kaip atkarpos pabaiga ir įrašomas į  $atkMas[][]$ . Tas pats atliekama ir tuo atveju, jei nėra nei vieno naujo spindulio iš paskutinio taško. Tuomet užfiksuojama atkarpos pabaiga į  $atkMas[][]$ . Po to vėl viskas kartojama nuo 1 punkto. O  $atkMas[][]$  masyve pereinama į naują eilutę.

Visi algoritmo eigos metu aplankyti taškai žymimi 2, kad nebūtų tiriami tie patys antrą kartą.



27 pav. Vektorizavimo algoritmas: ieškomi pasvirimo kampai bei jų skirtumai.

Vektorizavimo algoritmo metu, kiekviename naujai atrinktame taške, priklausomai nuo to, koks atkarpos pasvirimo kampas, ieškomas atkarpos storis taškais (**linijosStoris**) vertikalia (jei pasvirimo kampas  $[0^\circ, 45^\circ]$  arba  $[135^\circ, 180^\circ]$ ) arba horizontalia (jei pasvirimo kampas  $(45^\circ, 135^\circ)$ ) kryptimi (žr. 28 pav.). Skaičiuojami taškai kol nesutinkamas **0** arba kol neviršijama **th** reikšmė. Rasti atkarpos storiai sumuojami (**s**), o taškai, kuriuose jie buvo skaičiuoti, taip pat suskaičiuojami (**w\_sk**). Atkarpos pabaigos fiksavimo metu randamas atkarpos storio vidurkis (**linijosStoris = s / w\_sk**), kuris taip pat įrašomas į **atkMas[][]**, šalia tos atkarpos galų koordinatų.



28 pav. Vektorizavimo algoritmas: atkarpos storio paieška ( atkarpos pasvirimo kampas  $< 45^\circ$  ).

Taigi, iš pateikto algoritmo aprašymo bei iliustracijų matyti, jog linijos konstruojamos iš tarpinių taškų, esančių per visą tos linijos ilgį, nuo pradžios iki pabaigos taško. Tačiau rezultate fiksuojami tik galiniai taškai.

Dar vienas svarbus momentas yra tas, kad atkarpų „konstravimo“ bei jų storio paieškos metu, jei storis randamas mažiau nei trijuose tarpiniuose taškuose, tuomet algoritmas negali pasakyti, kas tai per objektas ir jį ignoruoja kaip šiukšlę. Šis sprendimas turi ir plusų, ir minusų. **Privalumas**: išvengiama triukšmų ir smulkių šiukšlių esančių vaizde, kurios galėjo atsirasti dėl brėžinio skaitymo skaitytuvu. **Trūkumas**: gali būti neatpažintos smulkios, tačiau vaizde reikalingos detalės. Iš dalies šią problemą išsprendžia parametro **th** pakankamai mažos reikšmės pasirinkimas, nes kuo mažesnis **th**, tuo smulkesnius objektus atpažįsta, t.y. neišmeta iš rezultatų rinkinio.

Taigi, tokiu būdu rastrinis vaizdas paverčiamas linijų, su savais atributais, rinkiniu. Toliau turėtų būti atliekama turimų atkarpų analizė ir jų apjungimas į apskritimus bei lankus

(jei tokių yra vaizde). Pagrindinės aptiktų objektų savybės (pvz.: centro koordinatės, spindulio ilgis, apskritimo linijos storis ir pan.) išsaugomos patogia forma. Apskritimų bei lankų atpažinimo algoritmas nedetalizuojamas.

### 3.5. Testavimo medžiaga

*Atliktas realizuoto sistemos modulio (automatinis vektorizavimas) testavimas:*

Sistemai buvo pateikiami pradiniai duomenys (rastrinis failas BMP, JPG arba GIF formato) ir tikrinami gauti pradiniai, tarpiniai bei galutiniai rezultatai. Po to buvo tikrinama, kaip sistema elgiasi pateikus klaidingus duomenis:

- Pateikus neegzistuojantį failo pavadinimą sistema išveda pranešimą:

```
Nepavyko nuskaityti is failo.
```

```
Pasitikrinkite ar gerai ivedete failo pavadinima,  
ar nepamirsote prideti .bmp, .gif, .jpg ar .jpeg?
```

- Bandant nurodyti failą vektorizavimui su plėtiniu, kurio vektorizuoti sistema negali (pvz. TIFF), sistema išveda tokį patį pranešimą kaip ir prieš tai esančiame punkte ir leidžia vartotojui pakartoti veiksmus, t.y. įvesti failą su plėtiniu, kurį vektorizuoti galima.
- Nurodant naujo failo pavadinimą, tačiau įvedus tik kokį nors simbolį (. / \ ir pan.), sistema į ekraną išveda pranešimą ir išeina iš programos. Pavyzdžiui nurodžius failo pavadinimą kaip “.”, gautas pranešimas:

```
Klaida nuskaitant . faila.
```

Ir buvo išeita iš programos. Toks elgesys nepageidaujamas. Vartotojui turi būti suteikta galimybė pakartoti veiksmus.

- Sistemai paprašius įvesti parametą **th**, nurodžius ne skaičių sistema išveda pranešimą:

```
Ivedete ne skaiciu - pakartokite ivedima!
```

Tačiau programa leidžiama įvesti bet kokį skaičių (tiek teigiamą, tiek neigiamą). Turėtų būti apribotas įvedamų skaičių intervalas.

Gauti testavimo rezultatai buvo apibendrinti ir pateikti testavimo išvadose 3.6. skyriuje.

Kitų modulių testavimas bus atliktas vėliau (juos realizavus). Šiuo metu svarbiausia – tai vektorizavimo procesas, todėl vartotojo sąsajos realizavimas bei testavimas atidėtas vėlesniam laikui.

### 3.6. Testavimo išvados

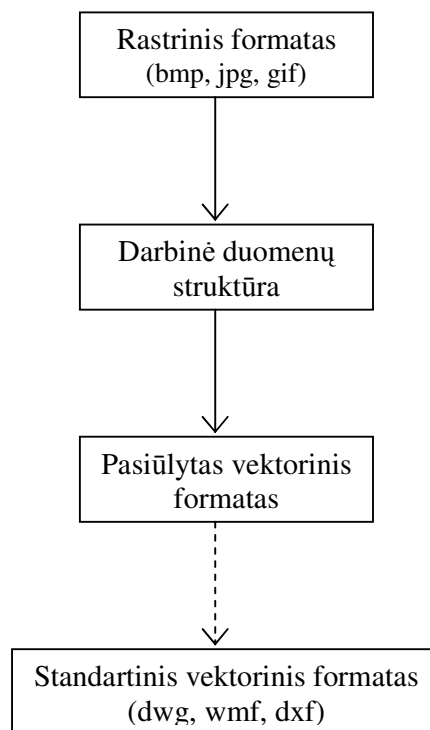
Atlikus sistemos testavimą ir apibendrinus rezultatus galima daryti tokias išvadas:

- esminių klaidų sistemoje nepastebėta, tačiau reikalingi apribojimai kai kurių duomenų įvedimui (tokiu būdu bus išvengta vartotojo sukeltų klaidų);
- sistema veikia stabiliai (“nelūžta”), o esant nesklaidumams vykdomas išėjimas iš sistemos;
- failų pavadinimams negalima naudoti simbolių (. / \ ir t.t.);
- įvedus klaidingus duomenis, sistema apie tai praneša vartotojui ir leidžia pakartoti veiksmus (ne visais atvejais);
- įvedant **th** reikšmę būtina elgtis pagal sistemos nurodymus, t.y. įvesti reikšmę intervale nuo 2 iki 10;
- sistema netrukdo kitų sistemų darbui;
- reikia ištaisyti aptiktas klaidas.

## 4. Vartotojo dokumentacija

### 4.1. Sistemos funkcinis aprašymas

Sistema skirta rastrinio vaizdo vektorizavimui. Rastrinis vaizdas - tai skenuoti brėžiniai, kurie sistemai turi būti pateikti BMP, JPG arba GIF formatu. Sistemos darbo rezultatas – pasiūlytas universalus vektorinio formato failas. Jame saugoma informacija apie brėžinyje aptiktus geometrinius objektus (linijas, apskritimus ir t.t.). Sistemos funkcinė schema pateikta 29 paveiksle. Sistema buvo kurta java programavimo kalba.



29 pav. Sistemos funkcinė schema.

Darbinė duomenų struktūra reikalinga tam, kad ateityje būtų galima prijungti ir kitų failų formatų vektorizavimą.

Rodyklė su pertrūkiais rodo kitą etapą vektorizavimo procese, kuris mūsų sistemoje nerealizuotas, tačiau tokiu būdu atspindi mūsų sistemos universalumą bei svarbą. Iš mūsų pasiūlyto vektorinio formato paversti į bet kokį kitą standartinį vektorinį formatą daug paprasčiau, nei vektorizuoti nuo pradžių.



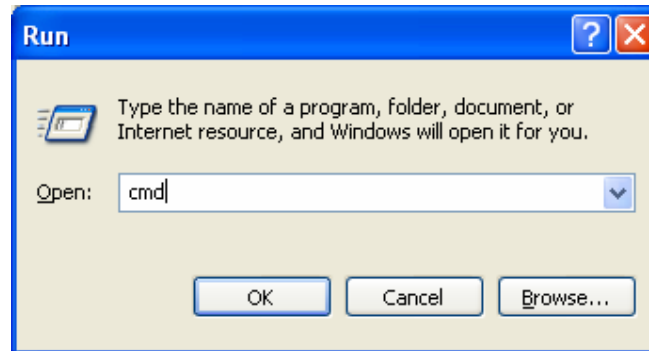
## 4.2. Sistemos vadovas

Sistema kol kas realizuota be vartotojo sąsajos, todėl ji paleidžiama per komandinę eilutę.

### 4.2.1 Programos paleidimas:

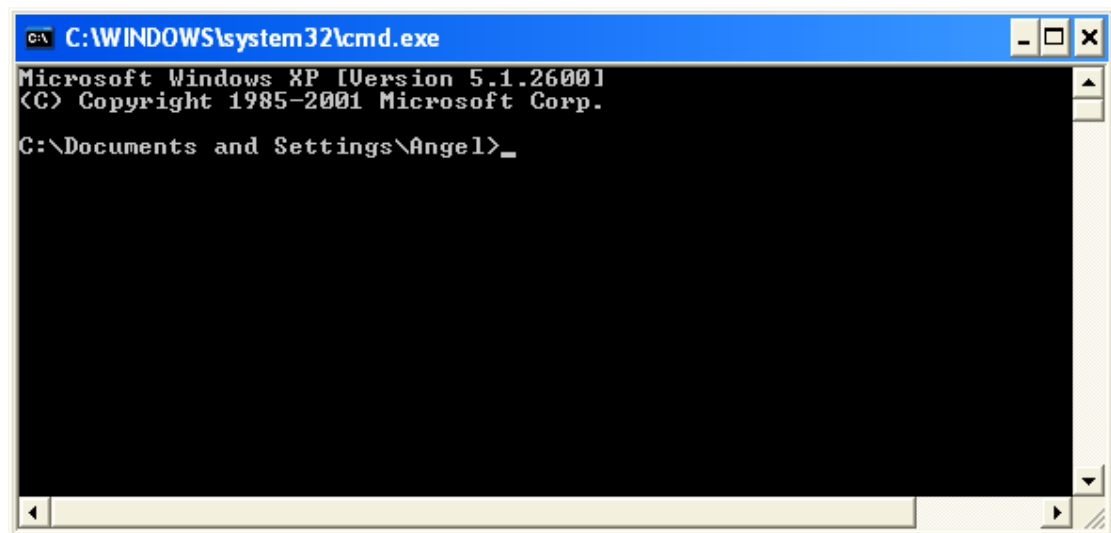
**Start > Run.**

4.2.2 Atsidariusiame dialogo lange įvedame **cmd** ir spaudžiam **OK** (žr. 30 pav.).



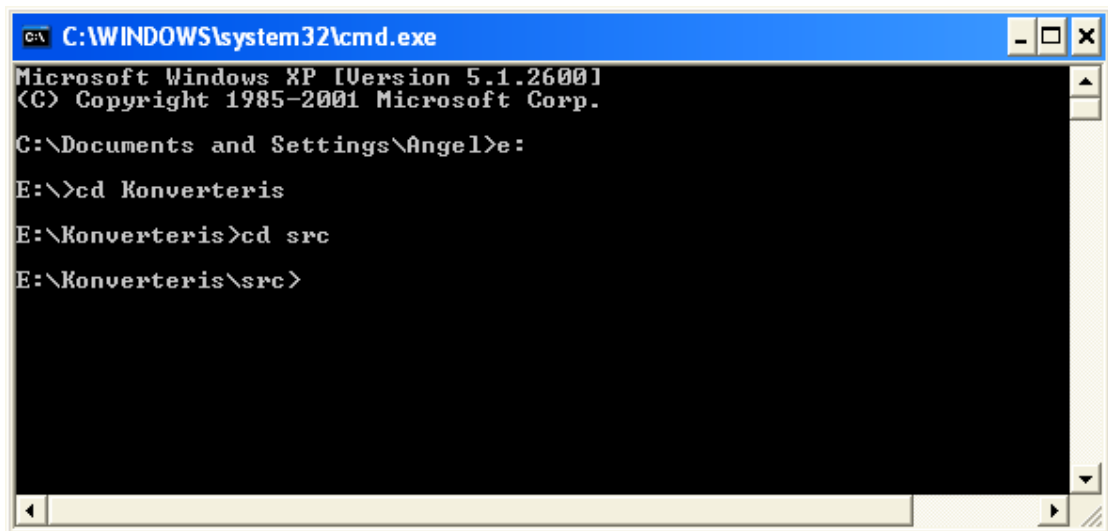
30 pav. Run dialogo langas.

4.2.3 Kompiuterio ekrane pasirodo langas, kaip parodyta 31 paveiksle.



31 pav. Komandinės eilutės langas (cmd.exe).

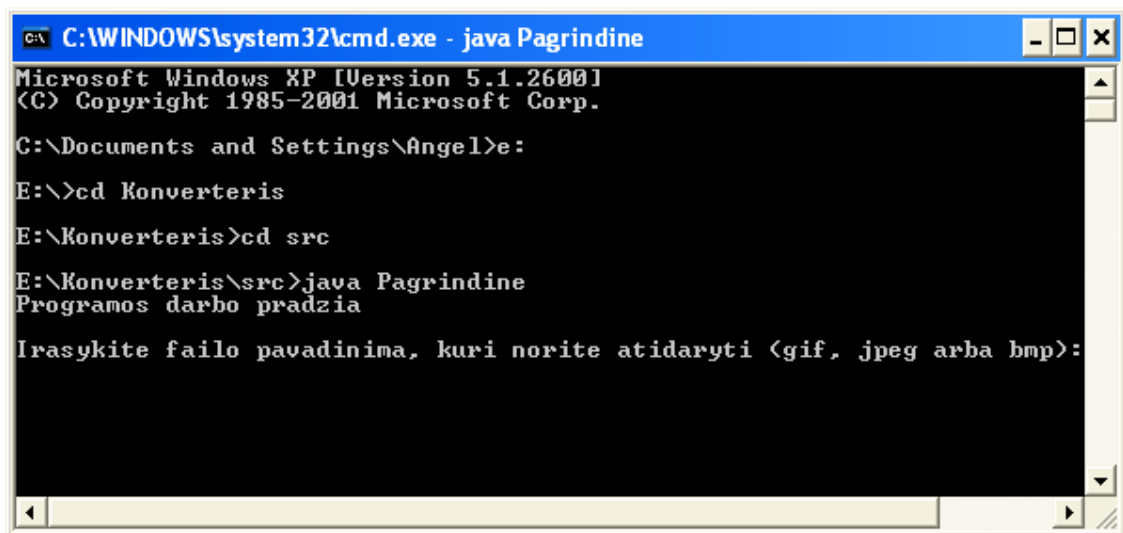
4.2.4 Reikia nurodyti kelią iki programos paleidžiamojo failo, kuris yra kataloge **Konverteris/src** (žr. 32 pav.). (įėjimas į kitą diską – žr. 32 pav., įėjimas į katalogą – komanda **cd** ir šalia įrašomas katalogo pavadinimas, spaudžiam **Enter**, kartojama kol pasiekiamas reikalingas katalogas).



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Angel>e:
E:\>cd Konverteris
E:\Konverteris>cd src
E:\Konverteris\src>
```

32 pav. Nurodomas kelias iki paleidžiamojo failo.

- 4.2.5 Įvedame: **java Pagrindine**, spaudžiam **Enter**. Programa pradėjo darbą (žr. 33 pav.).
- 4.2.6 Toliau vykdyti programos nurodymus, kuriuos ji išveda į ekraną (žr. 33 pav.).

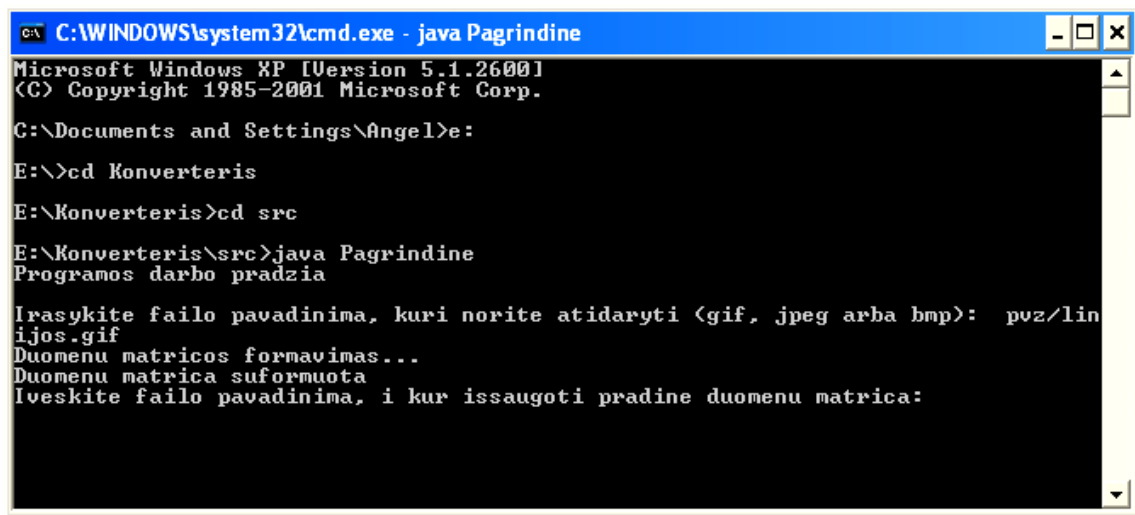


```
C:\WINDOWS\system32\cmd.exe - java Pagrindine
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Angel>e:
E:\>cd Konverteris
E:\Konverteris>cd src
E:\Konverteris\src>java Pagrindine
Programos darbo pradžia
Įrašykite failo pavadinimą, kuri norite atidaryti <gif, jpeg arba bmp>:
```

33 pav. Programos darbo pradžia.

- 4.2.7 Nurodant failo pavadinimą reikia įrašyti ir plėtinį (.jpg, .bmp, .gif) bei spausti **Enter**. Jei failas yra kataloge **Konverteris/src**, tuomet užtenka nurodyti tik failo pavadinimą su minėtu plėtiniu. Jeigu failas yra kažkur kitur, reikia nurodyti ir pilną kelią iki jo. Pavyzdžiui, failas linijos.gif yra

kataloge **Konverteris/src/pvz**, todėl reikia programai pateikti tokią informaciją: **pvz/linijos.gif** ir paspausti **Enter**.



```
C:\WINDOWS\system32\cmd.exe - java Pagrindine
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

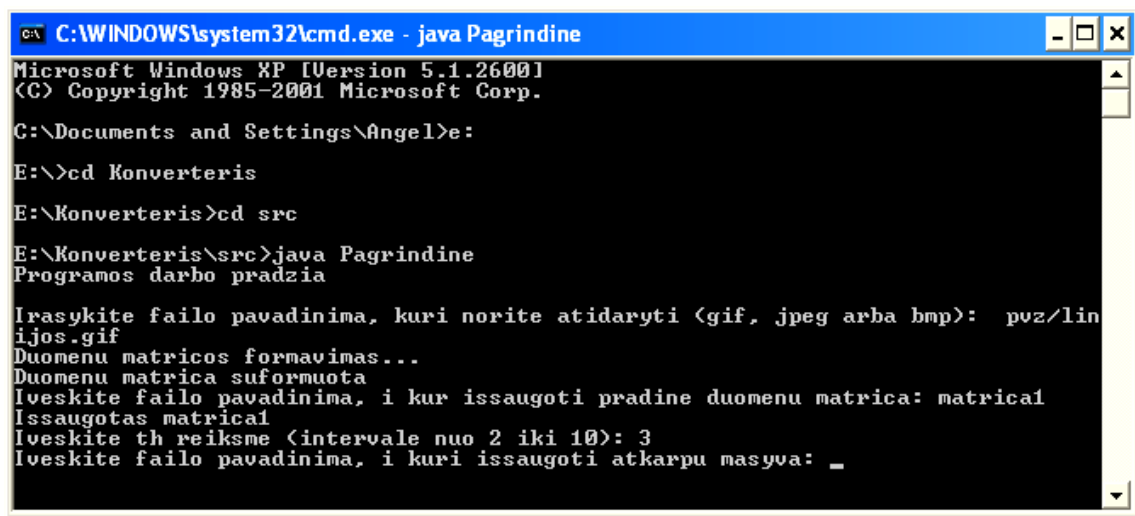
C:\Documents and Settings\Angel>e:
E:\>cd Konverteris
E:\Konverteris>cd src
E:\Konverteris\src>java Pagrindine
Programos darbo pradzia

Irasykite failo pavadinima, kuri norite atidaryti <gif, jpeg arba bmp>: pvz/linijos.gif
Duomenu matricos formavimas...
Duomenu matrica suformuota
Iveskite failo pavadinima, i kur issaugoti pradine duomenu matrica:
```

34 pav. Nurodomas failas, kurį norima vektorizuoti.

4.2.8 Toliau programa paprašo nurodyti failo pavadinimą, į kurį bus išsaugota pradinė duomenų struktūra (matrica, darbinis failas) (žr. 34 pav.). Laikytis tokių pačių taisyklių kaip parašyta 4.2.7 punkte, tačiau failo plėtinio nurodyti nebūtina. Pavyzdžiui, įvedus tik **matrica1** ir paspaudus **Enter**, failas bus išsaugotas kataloge **Konverteris/src**.

4.2.9 Programa paprašo įvesti **th** reikšmę intervale nuo 2 iki 10 (žr. 35 pav.). Rekomenduojama **th** reikšmė yra **3**. Taigi įvedę **3** paspauskite **Enter**.



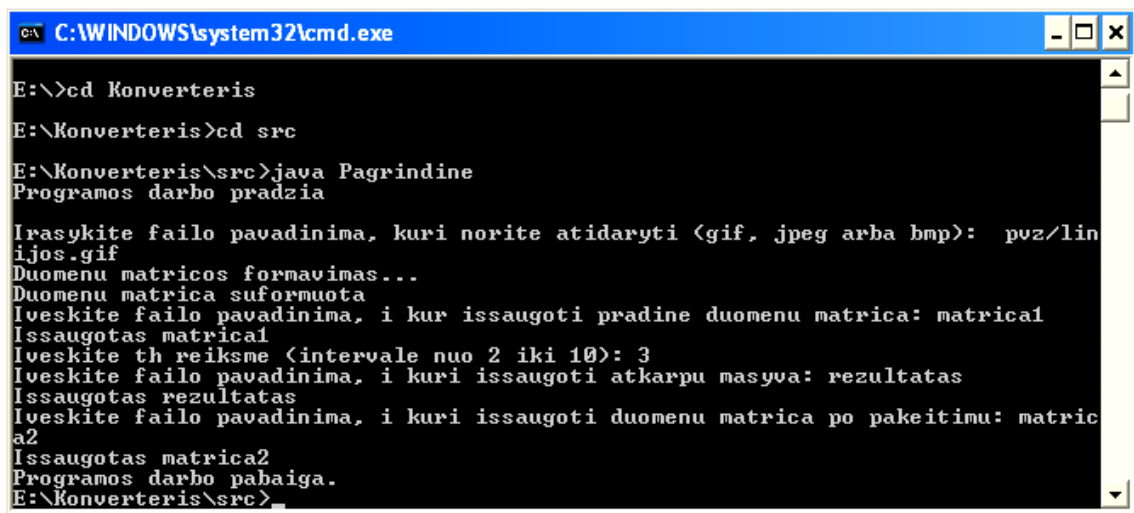
```
C:\WINDOWS\system32\cmd.exe - java Pagrindine
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Angel>e:
E:\>cd Konverteris
E:\Konverteris>cd src
E:\Konverteris\src>java Pagrindine
Programos darbo pradzia

Irasykite failo pavadinima, kuri norite atidaryti <gif, jpeg arba bmp>: pvz/linijos.gif
Duomenu matricos formavimas...
Duomenu matrica suformuota
Iveskite failo pavadinima, i kur issaugoti pradine duomenu matrica: matrica1
Issaugotas matrica1
Iveskite th reiksme <intervale nuo 2 iki 10>: 3
Iveskite failo pavadinima, i kuri issaugoti atkarpu masyva: _
```

35 pav. Nurodomas failas, į kuri išsaugoti pradinę duomenų struktūrą (**matrica1**), nurodoma parametro **th** vertė **3**.

4.2.10 Programa paprašo įvesti failo pavadinimą, į kurį bus išsaugoti surastų objektų duomenys (atkarpu masyvas) (žr. 35 pav.). Laikytis 4.2.8 punkte aprašytos tvarkos. 36 paveiksle pateiktame pavyzdyje įrašytas failo pavadinimas **rezultatas** ir paspausta **Enter**.



```
C:\WINDOWS\system32\cmd.exe
E:\>cd Konverteris
E:\Konverteris>cd src
E:\Konverteris\src>java Pagrindine
Programos darbo pradzia
Irasykite failo pavadinima, kuri norite atidaryti (gif, jpeg arba bmp): pvz/linijos.gif
Duomenu matricos formavimas...
Duomenu matrica suformuota
Iveskite failo pavadinima, i kur issaugoti pradine duomenu matrica: matrica1
Issaugotas matrica1
Iveskite th reiksme (intervale nuo 2 iki 10): 3
Iveskite failo pavadinima, i kuri issaugoti atkarpu masyva: rezultatas
Issaugotas rezultatas
Iveskite failo pavadinima, i kuri issaugoti duomenu matrica po pakeitimu: matrica2
Issaugotas matrica2
Programos darbo pabaiga.
E:\Konverteris\src>
```

36 pav. Rezultatų išsaugojimas ir darbo pabaiga.

4.2.11 Išsaugojus rezultatą, dar paprašoma išsaugoti ir duomenų matricą po pakeitimų. Laikytis 4.2.8 punkte aprašytos tvarkos. 36 paveiksle pateiktame pavyzdyje nurodytas failo pavadinimas **matrica2** ir paspausta **Enter**.

4.2.12 Programa baigia darbą automatiškai.

4.2.13 Norėdami vektorizuoti kitą failą, kartokite veiksmus nuo 4.2.5 punkto.

4.2.14 Jei darbą baigėte, išjunkite komandinės eilutės langą „kryžiuku“.

4.2.15 Rezultatų peržiūra:

4.2.15.1 BMP, JPG ar GIF rastrinius failus galima peržiūrėti su bet kokia grafikos peržiūros programa.

4.2.15.2 Vektorizavimo metu išsaugotus failus galima peržiūrėti su Microsoft WordPad.

### 4.3. Sistemos instaliavimo dokumentas

Sistemos instaliavimo dokumentas sudarytas iš: programos instaliavimo instrukcijos, reikalavimų programinei įrangai ir sistemą sudarančių failų.

Programa pateikiama kaip failų rinkinys CD laikmenoje.

### 4.3.1. Instaliavimo instrukcija

Kol kas, paprastumo dėlei, vartotojo sąsajos klasės nepateikiamos. Pradinius brėžinius galima peržiūrėti su bet kokia grafikos peržiūros programa, o rezultatai peržiūrimi su *WordPad*.

Sistema instaliuojama kai programos kodas, esantis CD disko kataloge **Konverteris**, perkeliamas į kompiuterį. Kompiuteryje turi būti įdiegtos JAVA runtime bibliotekos JRE (jas galima parsisiųsti iš <http://java.sun.com/javase/downloads/index.html>).

### 4.3.2. Reikalavimai techninei bei programinei įrangai

- Įdiegta Windows 98/2000/XP operacinė sistema;
- Procesorius: ne mažiau 2GHz;
- Kietojo disko talpa: ne mažiau 80 GB;
- Operatyvinė atmintis: ne mažiau 512MB;
- Skaitytuvas ar plačiaformatis skaitytuvas (nebūtina);
- Diskinis įrenginys CD-ROM (nebūtina, jei yra prieiga lokaliame tinkle prie instaliacinių failų);
- Kompiuteryje turi būti įdiegtos JAVA runtime bibliotekos JRE.

### 4.3.3. Sistemą sudarantys failai

Sistemą sudarantys failai yra kataloge **Konverteris/classes**:

- Pagrindine.class (java kodas 1 priede)
- imageIO.class (vienas iš metodų, t.y. saveMatrica() 3 priede)
- utils.class
- iObserver.class
- Metodai.class (java kodas 2 priede)

## 5. Produkto eksperimentinis tyrimas

Šiame skyriuje aprašytas sistemos eksperimentinis tyrimas. Tie patys originalai (4 priedas) buvo išsaugoti skirtingais formatais (BMP, JPG bei GIF) bei vektorizuoti tiek su Corel TRACE programa, tiek su mūsų sistema. Taip pat ištirta kai kurių parametų įtaka vektorizavimo rezultatui. Visi gauti rezultatai pateikti lentelių bei diagramų forma.

### 5.1. Tyrimas

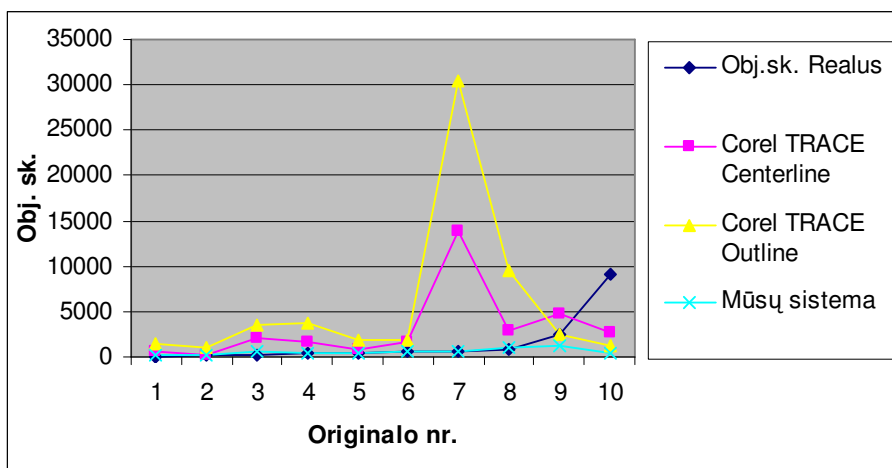
Corel TRACE programoje yra du pagrindiniai vektorizavimo tipai, tai *centerline* ir *outline*. Mūsų sistemoje pasirinkimo nėra. Turime 10 skirtingo sudėtingumo originalų (4 priedas). Žinomas tikslus objektų skaičius juose. Paprastumo dėlei jie buvo sunumeruoti nuo 1 iki 10, objektų skaičiaus didėjimo tvarka.

#### 5.1.1. Vektorizavimo rezultatų palyginimas

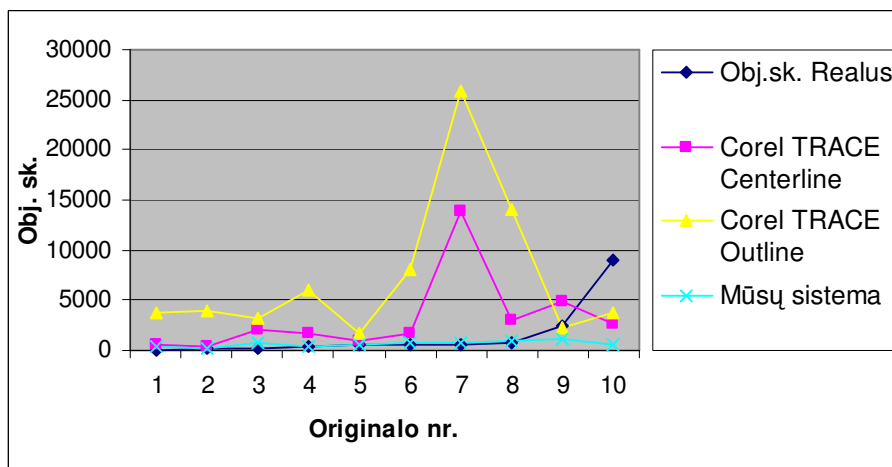
Atliktas **BMP**, **JPG** bei **GIF** originalų vektorizavimas, kai **threshold = 235** (abiejose sistemose), **th = 3** (mūsų sistemoje). Gauti vektorizavimo rezultatai – tai atpažintų objektų skaičius. Kiekvienu atveju šis skaičius skiriasi. Duomenys pateikti 17 lentelėje, o diagramos 37, 38 bei 39 paveiksluose.

17 lentelė. Vektorizavimo rezultatų suvestinė lentelė.

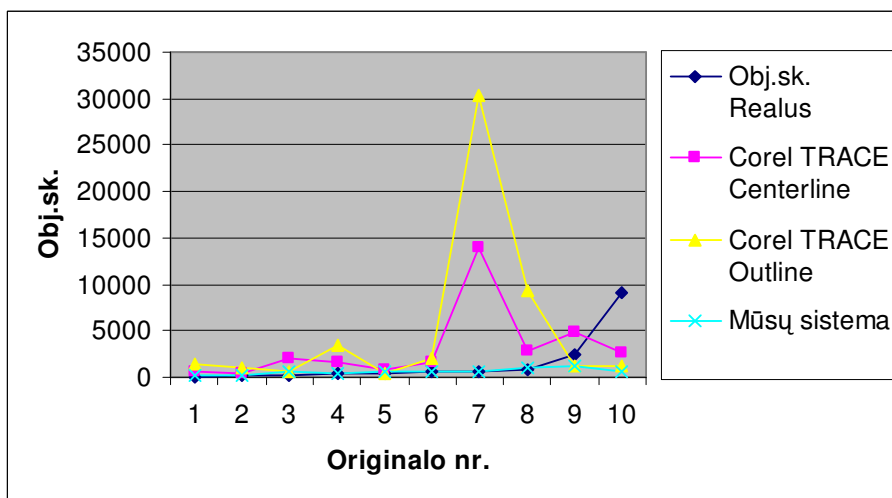
Ori- gin. nr.	Obj.sk. Realus	BMP			JPG			GIF		
		Corel TRACE Centerline	Corel TRACE Outline	Mūsų sistema	Corel TRACE Centerline	Corel TRACE Outline	Mūsų sistema	Corel TRACE Centerline	Corel TRACE Outline	Mūsų sistema
1	80	579	1430	266	580	3689	295	579	1430	266
2	154	307	953	254	307	3877	257	307	953	254
3	249	2061	3519	696	2061	3175	696	2061	674	696
4	319	1586	3723	380	1618	6048	407	1586	3526	380
5	469	881	1865	513	881	1658	513	881	443	513
6	544	1626	1936	648	1626	8045	657	1626	1936	648
7	549	13899	30474	707	13909	25783	711	13899	30417	707
8	832	2895	9491	932	3073	14102	958	2895	9326	932
9	2453	4851	2479	1169	4851	2255	1169	4851	1169	1169
10	9078	2592	1315	515	2592	3831	544	2592	1315	513



37 pav. **BMP** formato originalų vektorizavimo rezultatų palyginimas.



38 pav. **JPG** formato originalų vektorizavimo rezultatų palyginimas.



39 pav. **GIF** formato originalų vektorizavimo rezultatų palyginimas.

Iš pateiktų diagramų matyti, jog mažiausias nuokrypis nuo realaus dydžio buvo gautas vektorizuojant mūsų sistema. Tačiau šie duomenys nėra labai tikslūs, nes Corel sistemoje vektorizavimo rezultate gaunami objektai yra *paths* ir *nods*, mūsų sistemoje – *linijos*, o originaluose – tai ir *linijos*, ir *apskritimai*, ir *lankai*, ir *polilinijos* ir t.t. Originaluose esančių objektų skaičius rastas AutoCAD sistemos priemonėmis.

Didžiausias nuokrypis gautas su Corel sistema vektorizuojant 7-tą originalą. Jis ypatingas tuo, kad dalis plotų yra „užštrichuoti“. Todėl Corel kiekvieną „štrichą“ interpretavo kaip savarankišką objektą. Tokiu būdu objektų skaičius ryškiai išaugo.

Lyginant tik Corel vektorizavimo režimus, efektyvesnis (mažesnis nuokrypis nuo realių dydžių) yra *centerline* režimas.

Kuo sudėtingesnis originalas, t.y. kuo daugiau jame objektų, ir kuo jie įvairesni, tuo labiau vektorizavimo sistemų gauti rezultatai nukrypsta nuo realybės. Tiriant sistemą „juodosios dėžės“ principu (mūsų atveju tai Corel sistema, nes nežinome nei vektorizavimo algoritmo, nei kitų su tuo susijusių metodų, t.y. nežinoma vidinė sistemos struktūra), sunku paaiškinti ryškių nuokrypių priežastis. Savo sistemą tiriamo „baltosios dėžės“ principu, nes žinome vidinę struktūrą, todėl nesudėtinga suprasti aptiktus vektorizavimo reiškinius.

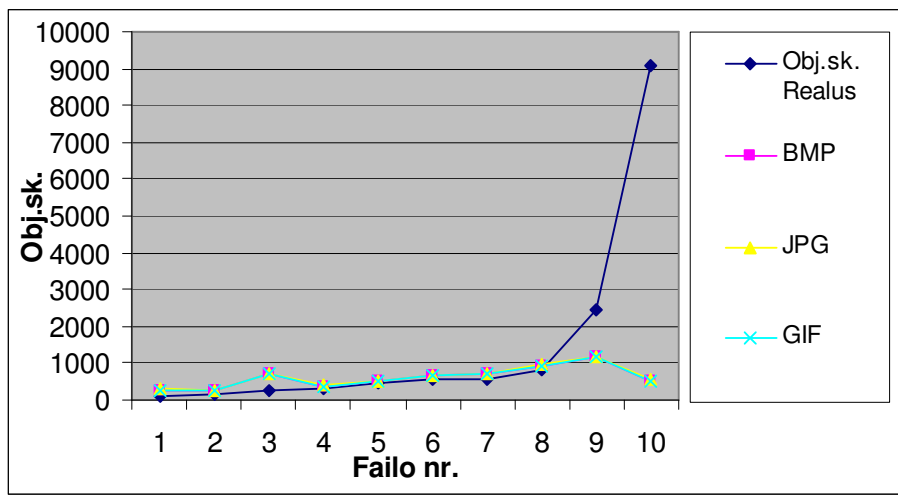
Vektorizavimo algoritmų darbo laiką tirtose sistemose palyginti sudėtinga, nes mūsų sistema kol kas yra be vartotojo sąsajos, tad rezultatas tik įrašomas į failą, o Corel sistemoje rezultatas pateikiamas ir vizualiai kompiuterio ekrane, todėl užtrunka ilgiau. Vizualizacija, kitaip dar vadinama rasterizacija - tai procesas atvirkščias vektorizavimui.

Dar vienas svarbus momentas – Corel sistemoje vektorizuojant *centerline* režimu, programa paprašo vaizdą paversti į *black and white*, nes kitaip negali atlikti vektorizavimo. Mūsų sistemoje, bet kokį pateiktą vaizdą (tiek BMP, tiek JPG, tiek GIF) sistema pati pasiverčia į jai reikalingą formą. Vartotojui nereikia atlikti jokių papildomų veiksmų.

### **5.1.2. Formato (BMP, JPG, GIF) įtaka vektorizavimo rezultatams**

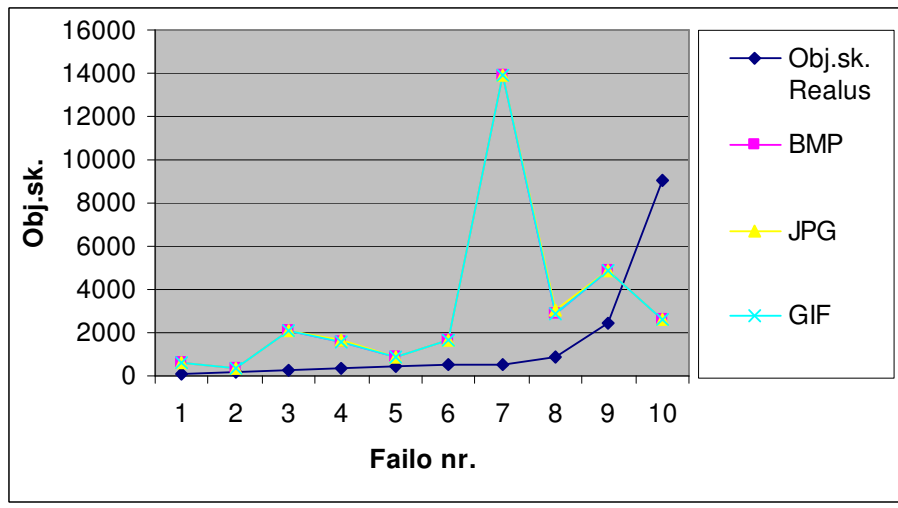
Duomenys, diagramoms atvaizduoti, paimti iš 17 lentelės.





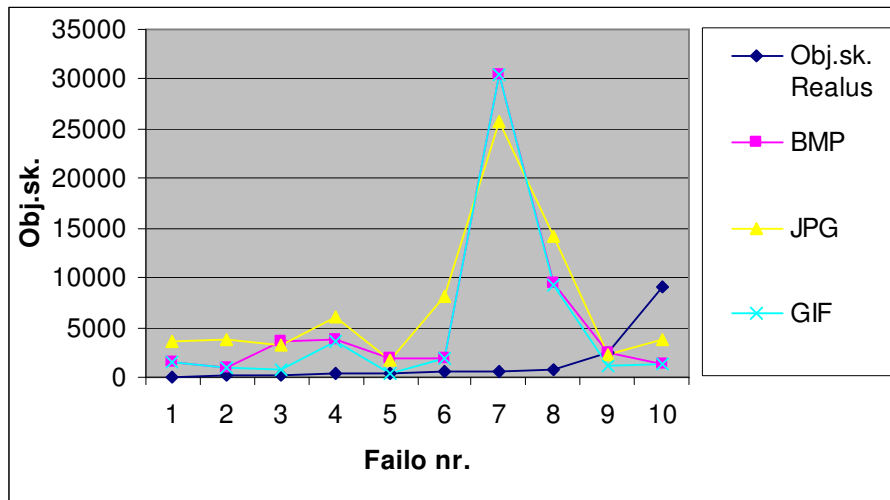
40 pav. Formato įtaka rezultatams vektorizuojant su *mūsų sistema*.

Originalaus (rastrinio) failo formatas (BMP, JPG, GIF) didelės įtakos rezultatams neturi, vektorizuojant tiek su mūsų sistema (žr. 40 pav.), tiek su Corel TRACE sistema (*centerline* režimu) (žr. 41 pav.).



41 pav. Formato įtaka rezultatams vektorizuojant su *Corel TRACE centerline* režimu.

Vektorizuojant su Corel TRACE sistema *outline* režimu (žr. 42 pav.), rezultatai priklausomai nuo rastrinio failo formato šiek tiek skiriasi.



42 pav. Formato įtaka rezultatams vektorizuojant su *Corel TRACE outline* režimu.

### 5.1.3. Parametro *th* įtaka vektorizavimo rezultatams

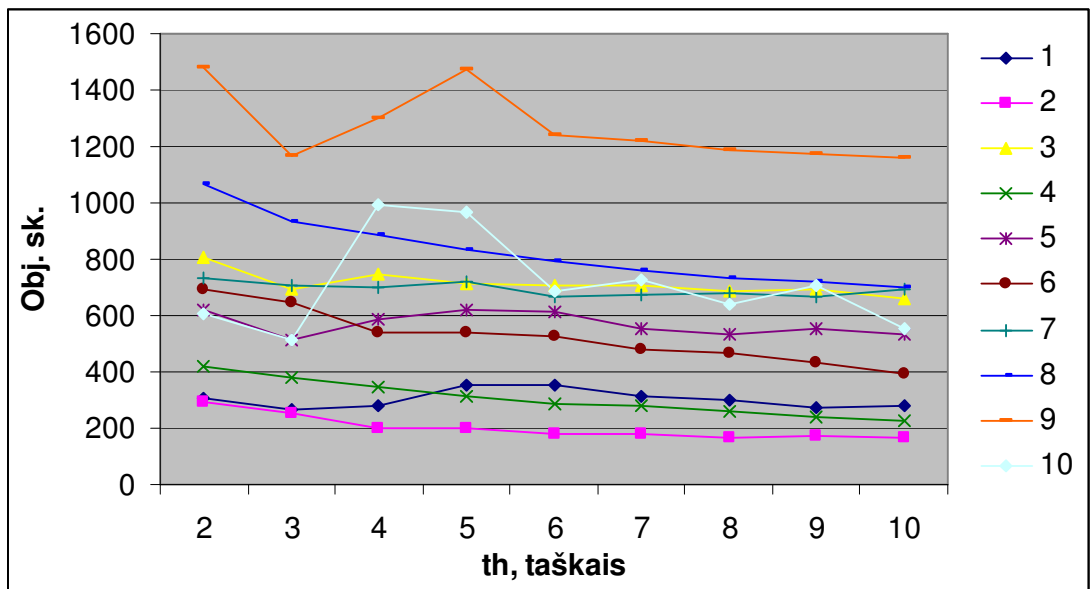
Ištirta parametro *th* įtaka vektorizavimo rezultatams (mūsų sistemoje). Gauti rezultatai pateikti 18 lentelėje bei 43 paveiksle. Tirtos *th* reikšmės nuo 2 iki 10 (18 lentelėje, vertikaliai). O skaičiais nuo 1 iki 10 (18 lentelėje, horizontaliai) sužymėti rastriniai originalai (4 priedas).

18 lentelė. *th* įtaka vektorizavimo rezultatui.

<i>th</i>	1	2	3	4	5	6	7	8	9	10
2	309	295	809	420	622	691	734	1064	1483	609
3	266	254	696	380	513	648	707	932	1169	515
4	281	200	744	346	586	539	701	888	1300	993
5	355	199	716	313	619	538	720	835	1471	969
6	354	180	707	285	615	529	669	793	1242	688
7	313	179	706	280	551	478	676	758	1222	724
8	303	166	684	260	531	468	679	733	1189	638
9	273	174	695	242	552	431	664	723	1176	710
10	278	170	663	229	533	392	691	701	1162	551

43 paveiksle pateiktoje diagramoje aiškiai matyti, jog didinant parametro *th* reikšmę, nesudėtinguose brėžiniuose (1 – 8 originalai) atpažintų objektų skaičius mažėja. Taip yra dėl to, kad kuo didesnė *th* reikšmė, tuo daugiau ir tuo didesni objektai yra ignoruojami kaip šiukšlės ir neįtraukiami į rezultatų sąrašą. Tai seka iš mūsų pasiūlyto vektorizavimo algoritmo (3.4.4. skyrius).

Sudėtinguose brėžiniuose (9 ir 10), didinant *th* reikšmę, objektų skaičius svyruoja neprognozuojamai, tačiau galiausiai pradeda tolygiai mažėti, kaip ir nesudėtinguose brėžiniuose.



43 pav. Parametro *th* įtaka vektorizavimo rezultatams.

#### 5.1.4. Rezultatų failo struktūra

Mūsų pasiūlyta vektorizavimo sistema generuoja pasiūlyto universalaus vektorinio formato failą, kurio vidinė struktūra parodyta 44 paveiksle. Kiekvienoje eilutėje surašyta grafinė informacija apie surastą objektą – atkarpą (liniją). Pirmi du skaičiai reiškia atkarpos pradžios koordinates **X1** ir **Y1**, kiti du skaičiai – tos atkarpos pabaigos koordinates **X2** ir **Y2**. Paskutinis skaičius (**s**) reiškia atkarpos storį taškais. Taigi, tokiu būdu pateikta grafinė informacija vėliau gali būti nesunkiai konvertuota į bet kokį standartinį vektorinį formatą (pvz. DWG, DXF, ir t.t.).

<b>X1</b>	<b>Y1</b>	<b>X2</b>	<b>Y2</b>	<b>s</b>
23	35	145	35	1
23	46	144	54	1
24	68	141	108	2
24	89	62	173	5
28	132	28	207	4
99	186	134	218	1
127	195	119	202	1
117	204	104	216	1

-----  
Liniju: 8

44 pav. Mūsų vektorizavimo algoritmo pateikti rezultatai, (*X1*, *Y1*) atkarpos pradžios taškas, (*X2*, *Y2*) atkarpos pabaigos taškas, *s* – atkarpos storis taškais.

### 5.1.5. Tirtų sistemų privalumai ir trūkumai

Tiek Corel TRACE, tiek mūsų sistema simbolių (raidžių, skaičių) neatpažįsta.

**Centerline** režimu vektorizuotas vaizdas geriau atitinka originalą, tačiau nelieka informacijos apie linijų storį. Tuo tarpu **Outline** režimu vektorizuotas vaizdas labiau iškreiptas, rezultatas labiau nutolęs nuo originalo, nei vektorizuojant pirmuoju būdu.

Mūsų sistema palyginti neblogai atpažįsta vertikalias, horizontalias, bet kokio storio linijas. Smulkūs bei trumpi objektai ignoruojami, todėl gali išnykti detalumas. Jei aptikta atkarpa kirto kitą, dar neatpažintą, tai vėliau toji neatpažintoji atkarpa fiksuojama, kaip dvi trumpesnės atkarpos.

Sistemų lyginamoji analizė apibendrinta 19 lentelėje.

19 lentelė. Tirtų vektorizavimo sistemų palyginimo lentelė.

	<b>Corel TRACE Centerline</b>	<b>Corel TRACE Outline</b>	<b>Mūsų vektorizavimo sistema</b>
Importuojami formatai	PPF, CPT, BMP, PCX, TGA, GIF, JPG, JP2, TIF, FPX, PP4, PSD, PP5, RIFF, PNG, CAL, PCT	PPF, CPT, BMP, PCX, TGA, GIF, JPG, JP2, TIF, FPX, PP4, PSD, PP5, RIFF, PNG, CAL, PCT	BMP, JPG, GIF
Paruošiamasis etapas	Rastrinį vaizdą reikia paversti į „ <b>black and white</b> “	Nereikia.	Nereikia.
Threshold (0 – 255)	Leidžia pasirinkti pačiam vartotojui.	-	Pastovus: 235. (nurodytas programos kode)
Kiti parametrai	<b>Iteration</b> bei <b>Node reduction</b> pasirenka pats vartotojas	<b>Accuracy</b> pasirenka pats vartotojas.	<b>tH</b> pasirenka pats vartotojas iš nurodyto intervalo (2 – 10).
Objektų tipas	<b>Paths</b> ir <b>Nods</b>	<b>Paths</b> ir <b>Nods</b>	<b>Linijos</b>
Bendras rastų objektų skaičius	+	+	+
Objektų išrūšiavimas pagal tipą	-	-	+
Linijų storis	-	?	+
Simbolių atpažinimas	-	-	-
Eksportuojami formatai	AI, CGM, CMX, DWG, DXF, EMF, PCT, PLT, WMF, WPG	AI, CGM, CMX, DWG, DXF, EMF, PCT, PLT, WMF, WPG	Universalus vektorinis, mūsų pasiūlytas, formatas (5.1.4 skyrelis)

„ + “ reiškia, kad ši savybė ar galimybė sistemoje yra, „ - “ nėra, „ ? “ neaišku ar yra ar ne.

## 5.2. Tyrimo išvados

Remiantis eksperimentinio tyrimo duomenimis galime daryti tokias išvadas:

- vektorizavimo rezultatas priklauso nuo:
  - ~ naudojamo vektorizavimo algoritmo;
  - ~ pasirinktų vektorizavimo parametrų;
  - ~ pradinio originalo kokybės.
- vektorizavimo procesas reikalauja patirties ir tos srities žinių iš sistemos vartotojo nurodant vektorizavimo parametrus;
- abiejų sistemų pateikti vektorizavimo rezultatai skiriasi, dėl prieš tai išvardintų veiksnių;
- rastrinio failo formatas turi mažai įtakos vektorizavimo rezultatams;
- parametras *th*, mūsų sistemoje, iš dalies veikia kaip šiukšlių filtras, tačiau nederėtų jo rinktis labai didelio, nes dings brėžinio detalumas;
- abiejų sistemų rezultatus palyginti gana sudėtinga dėl skirtingos rezultatų pateikimo formos;
- eksperimentinio tyrimo duomenys apytiksliai;
- mūsų sistema geriau išsaugo informaciją apie linijų storį;
- nei viena sistema neatpažįsta simbolių (raidžių, skaičių);
- mūsų sistemoje problemų sukelia susikertančios linijos, reikalinga susikirtimų atstatymo paprogramė;
- abiejų sistemų vektorizavimo rezultatai nėra idealūs;
- kuo brėžinys (rastrinis originalas) sudėtingesnis, kuo daugiau jame objektų, tuo sistemų vektorizavimo rezultatai labiau nutolę nuo pageidaujamo sprendinio;
- abi tirtos sistemos turi savų privalumų ir trūkumų;
- būtinos automatinio vektorizavimo algoritmo sudedamos dalys, kurios padėtų gauti tikslesnius rezultatus, yra:
  - atkarpų paieška (kartu su triukšmų filtru);
  - susikirtimų atstatymas;
  - surastų atkarpų jungimas į lankus bei apskritimus.

## 6. Išvados

1. Šiame darbe buvo išanalizuotas poreikis vektorizuoti rastrinius vaizdus. Ištirta vektorizavimo probleminė sritis bei jos aktualumas. Aptartos esminės su tuo susijusios problemos ir kiti svarbūs šios srities aspektai, kuriuos yra būtinybė automatizuoti kompiuterio pagalba.
2. Taip pat buvo išanalizuotos pagrindinės esamų vektorizavimo metodų grupės. Aptarti jų privalumai ir trūkumai. Rinkoje egzistuojantys panašūs produktai vis dar pakankamai brangūs ir dažniausiai yra didesnės sistemos dalis, o rezultatai ne patys geriausi. Dėl to, buvo nuspręsta pasiūlyti savą automatinio vektorizavimo algoritmą bei kurti naują rastrinio vaizdo vektorizavimo sistemą (RVVS), kurios rezultatas išsaugomas faile mūsų pasiūlytu universaliu vektoriniu formatu.
3. Atlikus srities analizę buvo padaryta išvada, jog optimali vektorizavimo išeitis – tai automatinio ir rankinio vektorizavimo sintezė. Buvo dokumentuoti vartotojų reikalavimai, atliktas sistemos specifikavimas pagal Volere šabloną.
4. Remiantis vartotojų iškeltais reikalavimais bei sukauptomis šios srities žiniomis, buvo suprojektuota RVVS.
5. Suprojektuota RVVS didelė ir sudėtinga, todėl buvo pasirinkta realizuoti tik pačią svarbiausią jos dalį – tai automatinio vektorizavimo modulį, panaudojant mūsų pasiūlytą vektorizavimo algoritmą. Realizacijai panaudota nuo platformos nepriklausanti JAVA programavimo kalba. Parengta reikalinga vartotojo dokumentacija.
6. Atliktas realizuotos sistemos dalies testavimas. Esminių klaidų neaptikta, tačiau dėl sprendžiamos problemos sudėtingumo, modulį dar reikia tobulinti. Pasiūlytam algoritmui dar reikia tam tikrų paprogramių, tam kad būtų gauti tikslesni vektorizavimo rezultatai.
7. Pastebėta, jog vektorizavimo rezultatai priklauso nuo: naudojamo vektorizavimo algoritmo; parinktų vektorizavimo parametrų; originalo kokybės bei vartotojo patirties šioje srityje.
8. Esminės vektorizavimo problemos: simboliai, linijų susikirtimo vietos, kampai, linijų bei jų galų tipas, užpildyti plotai.
9. Atliktas eksperimentinis tyrimas, kurio metu buvo palyginti vektorizavimo rezultatai, gauti su Corel TRACE bei su mūsų sistema. Ištirta rastrinio vaizdo formato bei parametrų įtaka vektorizavimo rezultatams.

## 7. Literatūra

1. DULINSKIENĖ, T. Skaitmeninio žemėlapio gavimo būdai ir perspektyvos. *Informacinės technologijos ir universitetinės studijos: konferencijos pranešimų medžiaga*. Kaunas, 1998, p. 31-33.
2. Oficialus Corel puslapis. [interaktyvus]. [žiūrėta 200-05-07]. Prieiga per internetą: <http://www.corel.co.uk/servlet/Satellite?pagename=Corel3Uk/Products/Display&pfid=1047024670787&pid=1047025895396>.
3. BYRNES, D. From Raster to Vector: A Road Map. *CADalyst*, 1994, Nr. 11(7), p. 40-50.
4. Dori Dov; Wenyin Liu. Sparse Pixel Vectorization: An Algorithm and Its Performance Evaluation. Iš *IEEE Transactions on Pattern Analysis and Machine Intelligence* [interaktyvus]. 1999, kovas [žiūrėta 2006-02-15]. Prieiga per internetą: <http://dori2.technion.ac.il/documents/articles/article180.pdf>.
5. Ferri Fernando; Grifoni Patrizia. A Vectorization Algorithm of Closed Regions in Raster Images. [interaktyvus]. 2002 [žiūrėta 2005-05-20]. Prieiga per internetą: <http://citeseer.ist.psu.edu/557642.html>.
6. GRIBOV, A.; BODANSKY, E. Vectorization with the Voronoi L-diagram. *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)*. Redlands, CA, USA, 2003, p. 1-5. [interaktyvus]. [žiūrėta 2005-03-20]. Prieiga per internetą: [http://www.cse.salford.ac.uk/prima/ICDAR2003/Papers/0186\\_417\\_gribov\\_a.pdf](http://www.cse.salford.ac.uk/prima/ICDAR2003/Papers/0186_417_gribov_a.pdf).
7. KRUCHTEN, P. Rational Unified Process, The: An Introduction, Third Edition. Boston, 2003.
8. Nieuwenhuizen Peter; Kiewiet Olaf; Bronsvort Willem. An Integrated Line Tracking and Vectorization Algorithm. Iš *Eurographics'94, volume 13, number 3* [interaktyvus]. 1994, [žiūrėta 2005-03-20]. Prieiga per internetą: <http://www.eg.org/EG/CGF/Volume13/Issue3/v13i3pp349-359.pdf>.
9. SIETHOF, J; Dr. DEVENTER, P. Based Map Conversion. *First Joint European Conference and Exhibition on Geographical Information*. Hague, 1995, p. 89.
10. Wenyin Liu; Dori Dov. From Raster to Vectors: Extracting Visual Information from Line Drawings. [interaktyvus]. 2005, liepa [žiūrėta 2006-03-20]. Prieiga per internetą: <http://www.cs.utah.edu/~tch/notes/wang/Raster.pdf>.

11. Wenyin Liu; Xiaoyu Wang; Long Tang; Dori Dov. Impact of Sparse Pixel Vectorization Algorithm Parameters on Line Segmentation Performance. 2001, p. 2.



## **8. Raster images vectorization system**

### *Summary*

The problem of raster images vectorization was analyzed and researched in this work. Existing vectorization systems are quite expensive, the results are inaccurate, and the manual vectorization of a large number of drafts is impossible. That's why our goal was to design and develop a new raster images vectorization system using our suggested automatic vectorization algorithm and the way to record results in a new universal vectorial file format. The work consists of these main parts: analysis, design, user documentation and experimental research.

Conclusions of the analysis of this field are that the optimal solution of vectorization is the synthesis of automatic and manual vectorization. Users' requirements were documented and raster image vectorization system designed. It was noticed that the result of vectorization depends on: vectorization algorithm; vectorization parameters; quality of raster image and user experience in this field. The core problems of vectorization are: symbol, intersection and corners recognition, lines' and their endings' types, filled areas. Vectorization results of our system and Corel TRACE were compared during the experimental research.

## 9. Terminų ir santrumpų žodynas

**CAD** – angl. *Computer Aided Design* – kompiuterinio projektavimo priemonės.

**PI** – programinė įranga.

**Produktas** – rastrinio vaizdo vektorizavimo sistema.

**Rastrinis vaizdas** – vaizdas elektroninėje formoje, sudarytas iš atskirų taškų, kurių kiekvienas turi savo informaciją apie spalvas.

**RUP** – angl. *Rational Unified Process* – apibendrintas projektavimo metodas.

**RVVS** - rastrinio vaizdo vektorizavimo sistema.

**Threshold** – slenkstinė vertė, naudojama kokiam nors dydžiui ar procesui apriboti.

**UML** – angl. *Unified Modeling Language* - apibendrinta modeliavimo kalba.

**Vektorinis vaizdas** – vaizdas sudarytas iš matematinių primityvų (atkarpų, apskritimų ir pan.), kurie turi savo atributus (pvz. atkarpos galų koordinatės  $x$ ,  $y$ , linijos storis ir pan.).

## 10. Priedai

Prieduose pateiktas ne visas programos kodas, o tik esminės jo dalys.

### 1 priedas. Pagrindine.java kodas

```
import java.util.*;
import java.io.*;

public class Pagrindine {
    final static int RED    = 0;
    final static int GREEN  = 1;
    final static int BLUE   = 2;
    final static int OFFSET = 3;

    // console naudojama vartotojo ivedamiems duomenims priimti
    static public BufferedReader console = new BufferedReader(
        new
InputStreamReader(System.in));

    public static void main(String[] args) {
        System.out.println("Programos darbo pradzia");
        System.out.println("");
        // programa turetu veikti be jokiu argumentu
        // vartotojui pateikus argumentus, isvedamas paaiskinamasis
pranesimas
        if (args.length!=0) {
            System.out.println("Vartosena: java Pagrindine");
            System.exit(1);
        }
        // vartotojo paprasoma ivesti failo varda, jo turinys
nuskaitomas i "testImage"
        int[][][] testImage = imageIO.loadImage();
        final int MAXROWS = testImage.length; // eiluciu skaicius
        final int MAXCOLS = testImage[0].length; // stulpeliu
skaicius
        // duomenu matricos, sudarytos tik is 0 ir 1, formavimas is
testImage
        // lieka tik du spalvos lygiai - juoda arba balta
        int[][] matrica = new int[MAXROWS][MAXCOLS];
        System.out.println("Duomenu matricos formavimas...");
        for (int row=0; row<MAXROWS; row++) {
            for (int col=0; col<MAXCOLS; col++) {
                if ((testImage[row][col][RED] >= 235) &&
                    (testImage[row][col][GREEN] >= 235) &&
                    (testImage[row][col][BLUE] >= 235)) {
                    matrica[row][col] = 0;
                } else {
                    matrica[row][col] = 1;
                }
            }
        }
        System.out.println("Duomenu matrica suformuota");
        // pradiniu duomenu issaugojimas i duomenu faila
```

```

        String pranesimas1 = "Iveskite failo pavadinima, i kur
issaugoti pradine duomeniu matrica: ";
        imageIO.saveMatrica(matrica, MAXROWS, MAXCOLS, pranesimas1);
        String th = null;
        int tH;
        // vartotojo prasoma nurodyti th parametro verte
        while (true) {
            System.out.print("Iveskite th reiksme (intervale nuo 2
iki 10): ");
            try {
                th = Pagrindine.console.readLine();
                try{
                    tH = (int)Integer.parseInt(th); //tH spindulio
keliaavimo slenkstine verte
                    break;
                } catch (NumberFormatException e){
                    System.out.println("Ivedete ne skaiciu -
pakartokite ivedima!");
                }
            } catch (IOException e){
                System.out.println(e);
                System.exit(1);
            }
        }

        // vektorizavimas (atkarpu paieška, galu koordinaciu bei
storio irasymas i masyva)
        int m = 500;
        int[][] atkMas = new int[m][5];
        m = 0;
        for (int y=0; y < MAXROWS; y++) {
            for (int x = 0; x < MAXCOLS; x++) {
                if (matrica[y][x] == 1) {
                    int x1 = x; // (x1,y1) tasko koordinates
                    int y1 = y;
                    atkMas[m][0] = x1; // atkarpos pradzia i masyva
                    atkMas[m][1] = y1;
                    // naujo tasko paieska
                    int[] nK = Metodai.spinduliai(matrica, x1, y1,
tH);
                    int x2 = nK[0]; // (x2,y2) naujo tasko
koordinates
                    int y2 = nK[1];
                    matrica[y][x] = 4;
                    int a0;
                    if (x2 == x1) {
                        a0 = 90; // atkarpa keliauja vertikaliai
zemy (90 l.k.)
                    } else {
                        a0 = (int) Math.atan((y2 - y1) / (x2 - x1));
// kampo skaiciavimas
                    }
                    if (a0 < 0) { // jei kampas neigiamas,
paverciamas i teigiama
                        a0 = 180 - Math.abs(a0);
                    }
                    // linijos plocio paieska tame taske
                    int s = 0;

```

```

        int linijosStoris =
Metodai.getLinijosStoris(matrica, a0, x2, y2, tH);
        s = s + linijosStoris;
        // 2 naujo tasko paieska
        nK = Metodai.spinduliai(matrica, x2, y2, tH);
        int x3 = nK[0]; // 2 naujo tasko koordinates
        int y3 = nK[1];
        double k1;
        if (x3 == x1) {
            k1 = 90;
        } else {
            k1 = Math.atan((y3 - y1) / (x3 - x1)); // 2
kampo skaiciav.
        }
        int a1 = (int) k1;
        if (a1 < 0) {
            a1 = 180 - Math.abs(a1);
        }
        linijosStoris =
Metodai.getLinijosStoris(matrica, a1, x3, y3, tH);
        s = s + linijosStoris;
        int a_sk0 = Math.abs(a1 - a0); // randamas kampu
skirtumas

        int[] atPab = new int[2];
        atPab[0] = x3; // atkarpos pabaigos koordinates
        atPab[1] = y3;
        boolean b = true;
        int w_sk = 2;
        int a2;
        while (b == true) {
            nK = Metodai.spinduliai(matrica, x3, y3,
tH); // naujo tasko paieska
            x3 = nK[0];
            y3 = nK[1];
            if (x3 == x1) {
                a2 = 90;
            } else {
                a2 = (int) Math.atan((y3 - y1) / (x3 -
x1)); // kampo skaiciavimas
            }
            if (a2 < 0) {
                a2 = 180 - Math.abs(a2);
            }
            linijosStoris =
Metodai.getLinijosStoris(matrica, a2, x3, y3, tH);
            s = s + linijosStoris;
            int a_sk = Math.abs(a2 - a1);
            // kampu skirtumas turi buti mazesnis arba
lygus pries tai buvusiam kampu skirtumui
            // ir turi buti rastas naujas taskas (o ne
likti sename)

            if ((a_sk <= a_sk0) && ((atPab[0] != x3) ||
(atPab[1] != y3))) {
                atPab[0] = x3; // jei tenkina salyga,
taskas
                atPab[1] = y3; // fiksojamas kaip
ieskomos atkarbos pabaiga
                a1 = a2;

```

```

        w_sk = w_sk + 1;
        b = true;
    } else {
        atPab[0] = atPab[0]; // jei ne,
fiksuojamas pries tai buves taskas
        atPab[1] = atPab[1]; // kaip atkarpos
pabaiga
        b = false; // atkarpa baigesi
    } // if pabaiga
} // while pabaiga
linijosStoris = (int) s / w_sk; // atkarpos
storio vidurkis
    atkMas[m][2] = atPab[0]; // atkarpos pabaigos
koordinates irasomos i atkarpu masyva
    atkMas[m][3] = atPab[1];
    atkMas[m][4] = linijosStoris;
    if (w_sk < 3) { // jei w_sk < 3, tai negali
pasakyti koks objektas, jis ignoruojamas kaip siuksle
        m = m;
    } else {
        m = m + 1; // atkarpu masyve pereinama i
sekancia eilute
        int a = atkMas[m][2];
        int c = atkMas[m][3];
        matrica[c][a] = 4;
    }
}
}
}
String pranesimas2 = "Iveskite failo pavadinima, i kuri
issaugoti atkarpu masyva: ";
imageIO.saveMatrica(atkMas, m, 5, pranesimas2); // atkarpu
galu koordinaciju masyvo irasyma si faila

String pranesimas3 = "Iveskite failo pavadinima, i kuri
issaugoti duomenu matrica po pakeitimu: ";
imageIO.saveMatrica(matrica, MAXROWS, MAXCOLS, pranesimas3);

// programos uzdarymas
System.out.print("Programos darbo pabaiga.");
try{
    console.close();
}
catch(IOException e) {
    System.out.println(e);
    System.exit(1);
}
System.exit(0);
}
}

```

## 2 priedas. Metodai.java kodas

```
import java.lang.Math.*;
import java.lang.*;

public class Metodai {
    public static int[] spinduliai(int[][]matrica, int tx, int ty,
int tH) {
        matrica[ty][tx] = 3; // pradinis taskas (jau aplankytas)
todel pazymimas 3
        // is jo iseina spinduliai 5 kryptimis, bus ieskomas
ilgiausias ir jo koordinates
        int sk1 = 1; // spindulys paleidziamas 1 kryptimi
        // sk - zingsniu skaicius, spindulio ilgis
        while ((matrica[ty][tx+sk1] == 1) && (sk1 <= tH)) {
            matrica[ty][tx+sk1] = 2; // aplankyti taskai pazymimi
2, kad neziuret tu paciu antra kart
            sk1 = sk1 + 1;
        }
        sk1 = sk1 - 1;
        int sk2 = 1; // spindulys paleidziamas 2 kryptimi
        while ((matrica[ty+sk2][tx+sk2] == 1) && (sk2 <= tH)) {
            matrica[ty+sk2][tx+sk2] = 2;
            sk2 = sk2 + 1;
        }
        sk2 = sk2 - 1;
        int sk3 = 1; // spindulys paleidziamas 3 kryptimi
        while ((matrica[ty+sk3][tx] == 1) && (sk3 <= tH)) {
            matrica[ty+sk3][tx] = 2;
            sk3 = sk3 + 1;
        }
        sk3 = sk3 - 1;
        int sk4 = 1; // spindulys paleidziamas 4 kryptimi
        while ((matrica[ty+sk4][tx-sk4] == 1) && (sk4 <= tH)) {
            matrica[ty+sk4][tx-sk4] = 2;
            sk4 = sk4 + 1;
        }
        sk4 = sk4 - 1;
        int sk5 = 1; // spindulys paleidziamas 5 kryptimi
        while ((matrica[ty][tx-sk5] == 1) && (sk5 <= tH)) {
            matrica[ty][tx-sk5] = 2;
            sk5 = sk5 + 1;
        }
        sk5 = sk5 - 1;
        int[] nK = new int[2];
        if ((sk1 == 0) && (sk2 == 0) && (sk3 == 0) && (sk4 == 0) &&
(sk5 == 0)) {
            // jei nauju spinduliu is tasko neiseina
            // paliekamos to pacio tasko koordinates
            nK[0] = tx;
            nK[1] = ty;
        } else {
            // jei spinduliu yra, randamos ilgiausio spindulio galo
koordinates, tai bus naujas taskas
            nK = ilgiausiasSpindulys(sk1, sk2, sk3, sk4, sk5, tx,
ty);
        }
    }
}
```

```

        return nK;
    } // metodo spinduliai() pabaiga

    private static int[] ilgiausiasSpindulys(int sk1, int sk2, int
sk3,
                                                int sk4, int sk5, int
tx, int ty) {
        int[] nT = new int[2];
        int max = sk5;
        nT[0] = tx - max;
        nT[1] = ty;
        if (max <= sk4) {
            max = sk4;
            nT[0] = tx - max;
            nT[1] = ty + max;
        } else {
            max = max;
            nT[0] = nT[0];
            nT[1] = nT[1];
        }
        if (max <= sk3) {
            max = sk3;
            nT[0] = tx;
            nT[1] = ty + max;
        } else {
            max = max;
            nT[0] = nT[0];
            nT[1] = nT[1];
        }
        if (max <= sk2) {
            max = sk2;
            nT[0] = tx + max;
            nT[1] = ty + max;
        } else {
            max = max;
            nT[0] = nT[0];
            nT[1] = nT[1];
        }
        if (max <= sk1) {
            max = sk1;
            nT[0] = tx + max;
            nT[1] = ty;
        } else {
            max = max;
            nT[0] = nT[0];
            nT[1] = nT[1];
        }
        return nT; // grazina ilgiausio spindulio galo koordinates
    }

    public static int getLinijosStoris(int[][] matrica, int a, int x,
int y, int tH) {
        if ((a <= 45) || (a >= 135)) {
            int s1 = 1; // spindulys paleidziamas i virsu
            while ((matrica[y-s1][x] != 0) && (s1 <= tH)) {
                //matrica[y-s1][x] = 5;
                s1 = s1 + 1;
            }
        }
    }

```



```

        int s2 = 1; // spindulys paleidziamas i apacia
        while ((matrica[y+s2][x] != 0) && (s2 <= tH)) {
            //matrica[y+s2][x] = 5;
            s2 = s2 + 1;
        }
        int storis = s1 + s2 - 1;
        return storis; // grazina linijos stori
    } else if ((a > 45) && (a < 135)) {
        int s3 = 1; // spindulys paleidziamas i desine
        while ((matrica[y][x+s3] != 0) && (s3 <= tH)) {
            //matrica[y][x+s3] = 5;
            s3 = s3 + 1;
        }
        int s4 = 1; // spindulys paleidziamas i kaire
        while ((matrica[y][x-s4] != 0) && (s4 <= tH)) {
            //matrica[y][x-s4] = 5;
            s4 = s4 + 1;
        }
        int storis = s3 + s4 - 1;
        return storis; // grazina linijos stori
    }
    return (0);
}
} // klases Metodai pabaiga

```

### 3 priedas. Metodų saveMatrica() ir saveRez() kodai

Abu metodai yra imageIO klasėje (faile: imageIO.java).

```
public static void saveMatrica(int[][] matrica, int MAXROWS, int
MAXCOLS, String pra){
    String saveName = null;
    // rartotojo prasoma nurodyti naujo failo pavadinima
    System.out.print(pra);
    try{
        saveName = Pagrindine.console.readLine();
    }
    catch (IOException e){
        System.out.println(e);
        System.exit(1);
    }
    // rasytas i failar
    PrintWriter rs = null;
    try {
        File f1 = new File(".", saveName); // issaugoma, kur ir
kiti class failai
        rs = new PrintWriter(new FileWriter(f1));
    } catch (IOException e) {
        System.out.println("Klaida atidarant " + saveName + "
faila.");
        System.exit(0);
    }
    // masyvo eilutes veciamos i simboliu eilutes ir irasoma i
faila
    for (int row=0; row<MAXROWS; row++) {
        String rasymui = "";
        for (int col = 0; col < MAXCOLS; col++) {
            rasymui = rasymui + matrica[row][col] + " ";
        }
        rs.println(rasymui);
    }
    if (rs != null) {
        rs.close(); // klasei PrintWriter IOException nemeta
    }
    System.out.println("Issaugotas " + saveName);
} // metodo saveMatrica() pabaiga
```

```
public static void saveRez(int[][] atkMas, int eil, int st, String
pra){
    String saveName = null;
    // vartotojo prasoma nurodyti naujo failo pavadinima
    System.out.print(pra);
    try{
        saveName = Pagrindine.console.readLine();
    }
    catch (IOException e){
        System.out.println(e);
        System.exit(1);
    }
    // rasytas i faila
    PrintWriter rs2 = null;
```

```

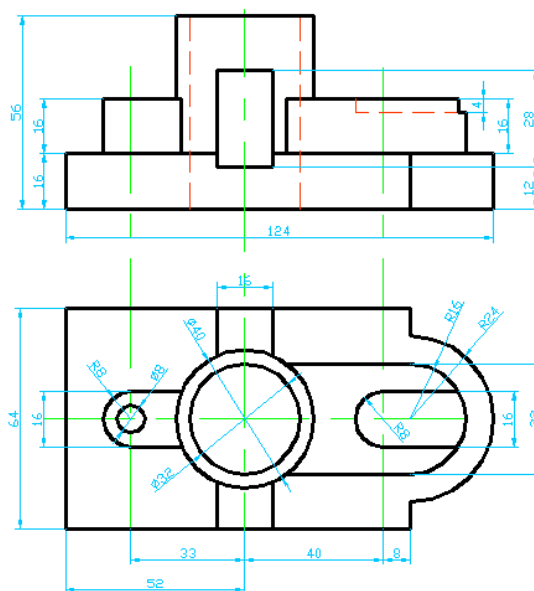
        try {
            File f2 = new File(".", saveName); // issaugoma, kur ir
kiti class failai
            rs2 = new PrintWriter(new FileWriter(f2));
        } catch (IOException e) {
            System.out.println("Klaida atidarant " + saveName + "
faila.");
            System.exit(0);
        }
        // masyvo eilutes veciamos i simboliu eilutes ir irasoma i
faila
        for (int row2=0; row2 < eil; row2++) {
            String rasymui2 = "";
            for (int col2 = 0; col2 < st; col2++) {
                rasymui2 = rasymui2 + atkMas[row2][col2] + " ";
            }
            rs2.println(rasymui2);
        }
        rs2.println("-----");
        rs2.println("Liniju: " + eil);
        if (rs2 != null) {
            rs2.close(); // klasei PrintWriter IOException nemeta
        }
        System.out.println("Issaugotas " + saveName);
    } // metodo saveRez() pabaiga

```

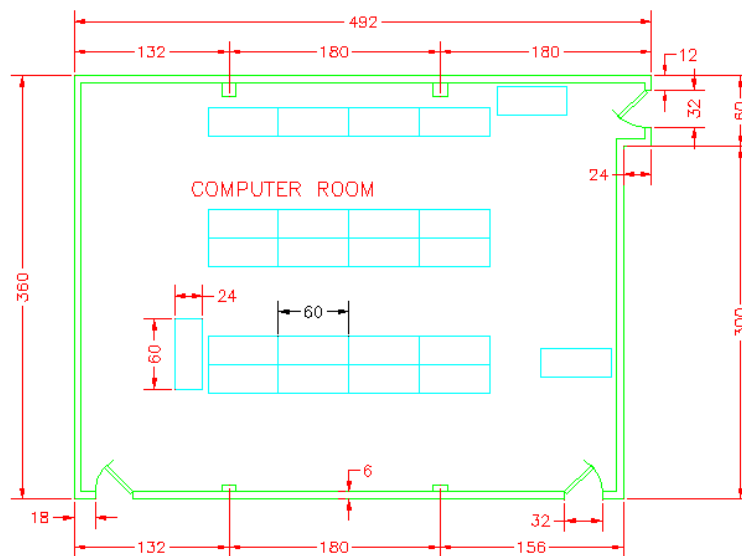
#### 4 priedas. Tyrimams naudoti rastriniai originalai

Originalų matmenys tyrimo metu buvo: 951 x 505 taškai (angl. *pixels*).

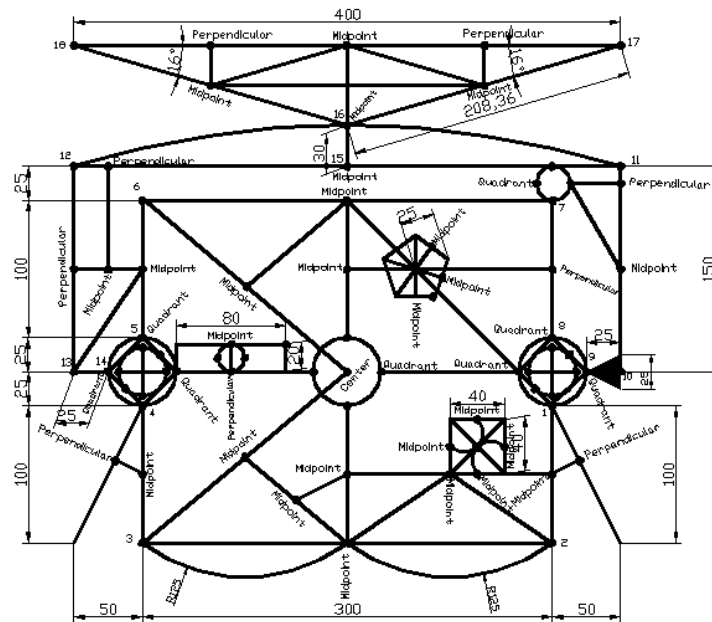
##### 1 originalas: detalė.



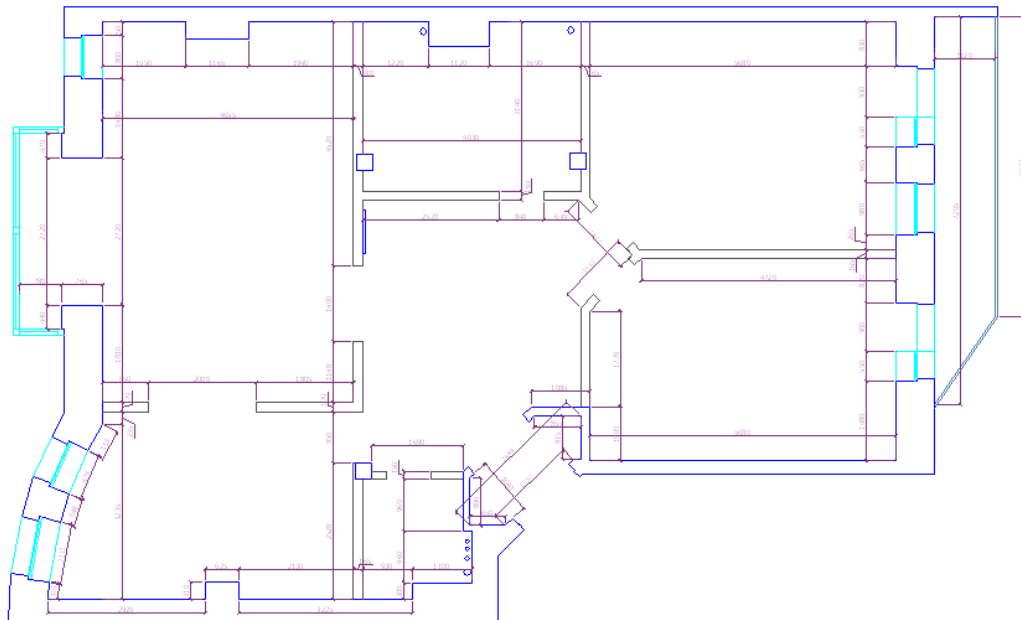
##### 2 originalas: kambario planas.



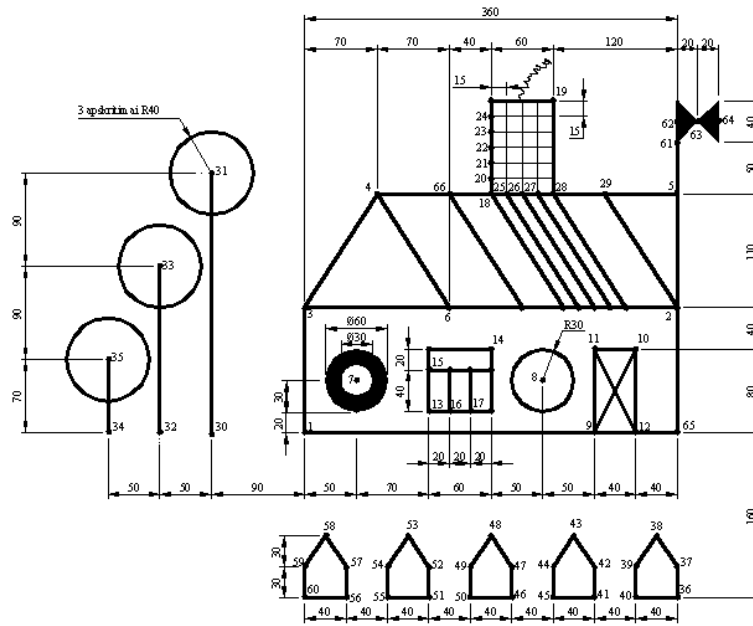
3 originalas: nestandartinis brėžinys.



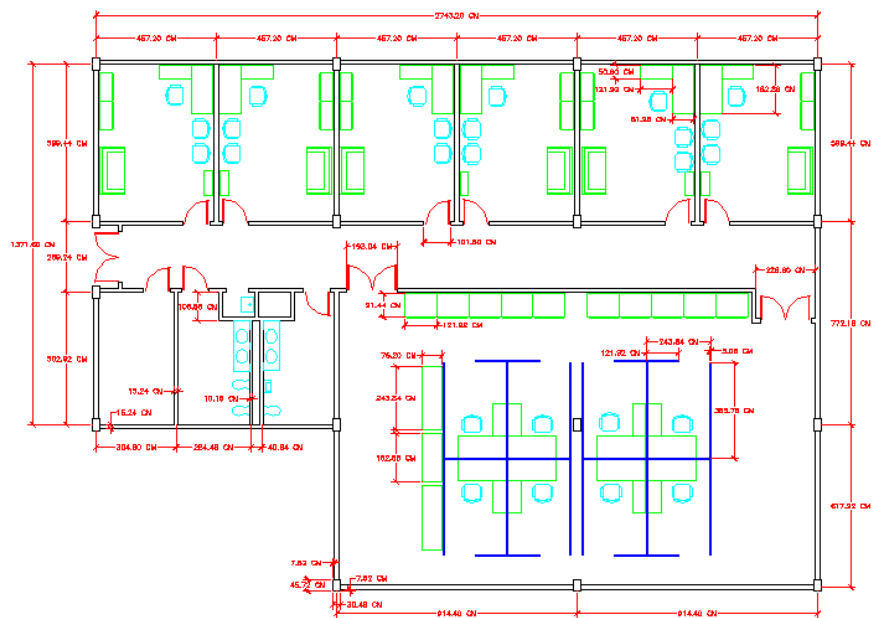
4 originalas: buto planas 1.



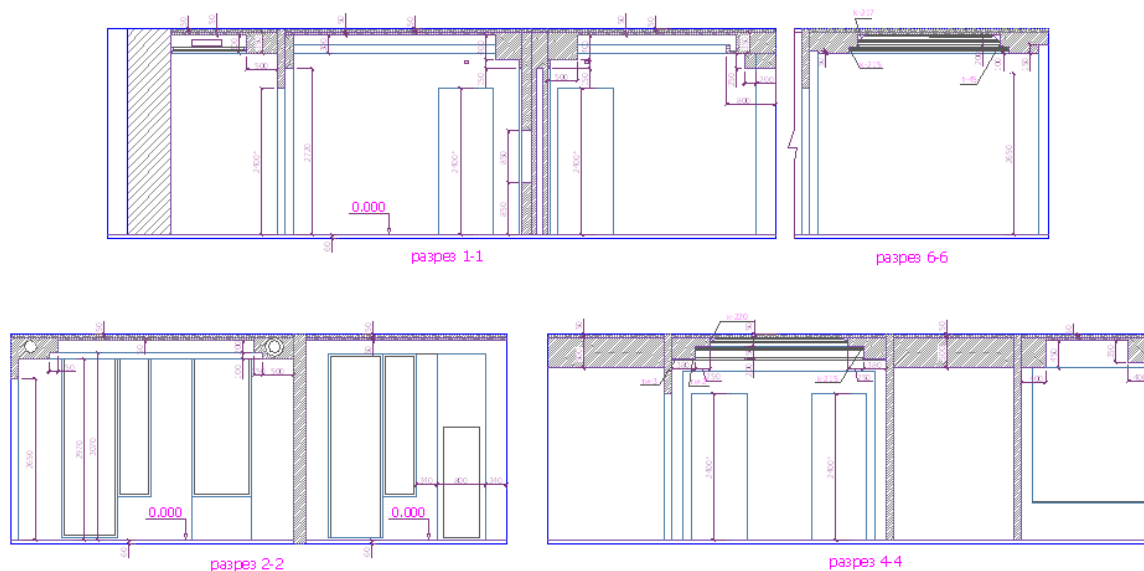
5 originalas: namukas.



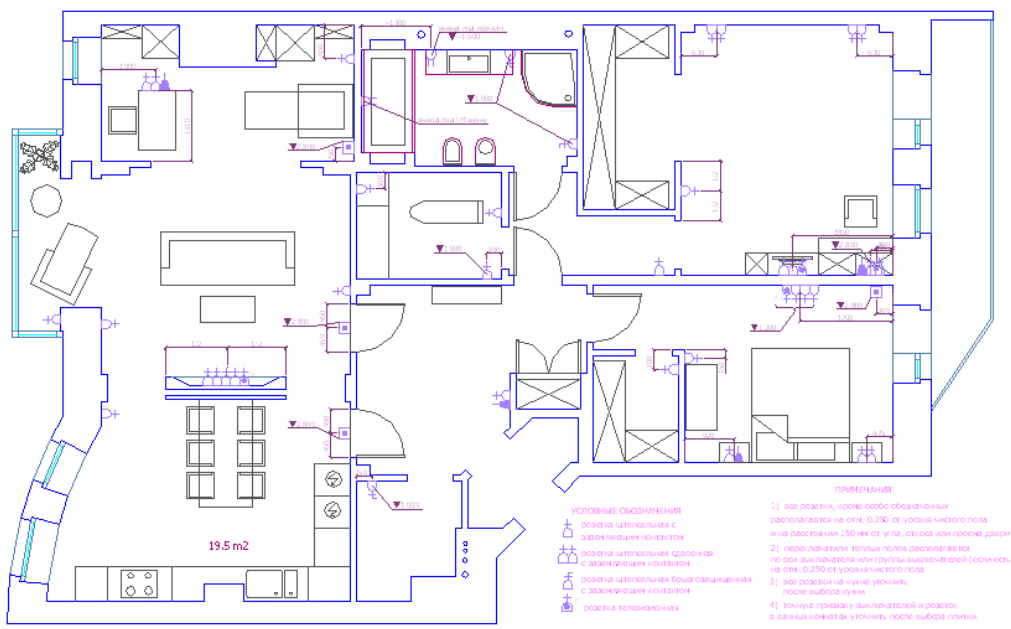
6 originalas: biuro patalpų planas.



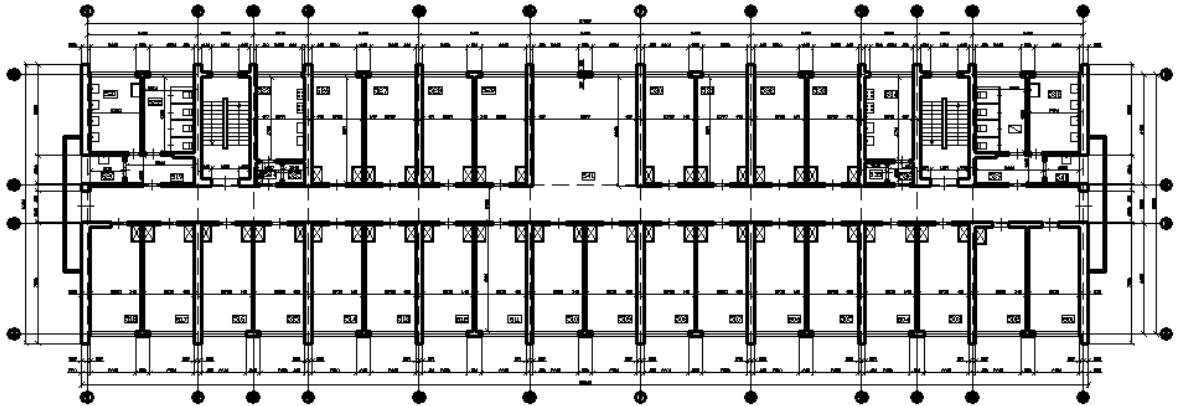
7 originalas: buto planas 2.



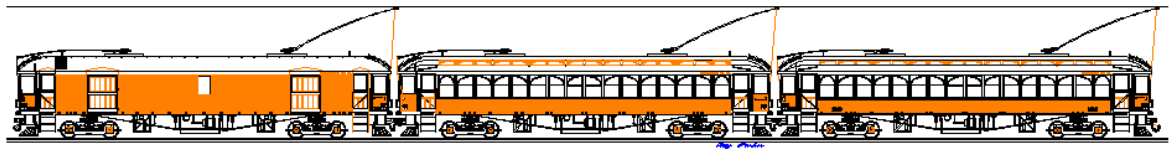
8 originalas: buto planas 3.



9 originalas: 5-to aukšto planas.



10 originalas: traukinys.



The BC Electric Interurbans



**5 priedas.** Kompaktinis diskas su programos kodu.