



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
KOMPIUTERIŲ INŽINERIJOS KATEDRA

Leonas Jasevičius

**INTEGRINIŲ GRANDYNŲ RASTRINIŲ VAIZDŲ VEKTORIZAVIMO
SISTEMA**
VECTORIZATION SYSTEM FOR RASTER IMAGES OF INTEGRATED CIRCUITS

Baigiamasis magistro darbas

Elektronikos studijų programa, valstybinis kodas 62401T209

Elektroninės aparatūros ir kompiuterinės technikos specializacija

Elektronikos inžinerijos studijų kryptis

Vilnius, 2009



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
KOMPIUTERIŲ INŽINERIJOS KATEDRA

TVIRTINU
Katedros vedėjas

(parašas)
Julius Skudutis
(Vardas, pavardė)

(Data)

Leonas Jasevičius

**INTEGRINIŲ GRANDYŲ RASTRINIŲ VAIZDŲ VEKTORIZAVIMO
SISTEMA**

VECTORIZATION SYSTEM FOR RASTER IMAGES OF INTEGRATED CIRCUITS

Baigiamasis magistro darbas

Elektronikos studijų programa, valstybinis kodas 62401T209

Elektroninės aparatūros ir kompiuterinės technikos specializacija

Elektronikos inžinerijos studijų kryptis

Vadovas

(Moksl. laipsnis, vardas, pavardė) (Parašas) (Data)

Konsultantas

(Moksl. laipsnis, vardas, pavardė) (Parašas) (Data)

Konsultantas

(Moksl. laipsnis, vardas, pavardė) (Parašas) (Data)

Vilnius, 2009

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
KOMPIUTERIŲ INŽINERIJOS KATEDRA

MAGISTRANTŪROS STUDIJŲ BAIGIAMOJO DARBO (TEZIŲ) UŽDUOTIS

Elektronikos fakultetas
Kompiuterių inžinerijos katedra
Technikos mokslų sritis
Elektros ir elektronikos inžinerijos mokslo kryptis
Elektronikos inžinerijos studijų kryptis
Elektronikos studijų programa
Elektroninės aparatūros ir kompiuterinės technikos
specializacija

TVIRTINU
Kompiuterių inžinerijos katedros vedėjas
prof. Julius Skudutis
2008 gruodžio d.

Studentui (ei) Leonui Jasevičiui
(vardas, pavardė)

Baigiamojo darbo (tezių) tema: Integrinių grandynų rastrinių vaizdų vektorizavimo sistema
Vectorization system for raster images of integrated circuits

Patvirtinta 200 ____ m. ____ mėn. ____ d. dekanų įsakymu Nr. ____
Baigiamojo darbo (tezių) užbaigimo terminas 200 ____ m. ____ mėn. ____ d.

Darbo tikslas:
Sukurti vektorizavimo sistemą, skirtą integrinių grandynų rastrinių vaizdų vektorizavimui

BAIGIAMOJO DARBO (TEZIŲ) UŽDUOTIS:

- 1) Literatūros analitinė apžvalga.
- 2) Darbo tikslo ir uždavinių formulavimas. Reikalavimai kuriamai programinei įrangai ir jos dokumentacijai.
- 3) Algoritmų tobulinimas. Jų veikimo palyginimas.
- 4) Vektorizavimo metodų ir sistemų tyrimas.
- 5) Apibendrinimas

Baigiamojo darbo (tezių) rengimo konsultantai: Doktorantas Giedrius Masalskis

(vardas, pavardė, mokslinis laipsnis ir vardas)

Vadovas: _____
(parašas) **Prof. habil. dr. Romualdas Navickas**
(vardas, pavardė, mokslinis laipsnis ir vardas)

Užduotį gavau: _____
(studento parašas) _____
(data)

Vilniaus Gedimino technikos universitetas

Elektronikos fakultetas

Kompiuterių inžinerijos katedra

ISBN

ISSN

Egz. sk.

Data-....-....

Elektronikos studijų programos baigiamasis magistro darbas

Pavadinimas **INTEGRINIŲ GRANDYŲ RASTRINIŲ VAIZDŲ VEKTORIZAVIMO SISTEMA**

Autorius **Leonas Jasevičius**

Vadovas prof. hab. dr. **Romualdas Navickas**

Kalba



lietuvių



užsienio

Anotacija

Baigiamajame magistro darbe išnagrinėta integrinių grandynų sluoksnių taškinių vaizdų vektorizavimo problema ir aptarti įvairūs linijų taškiniuose vaizduose išskyrimo būdai, jų ypatybės integrinių grandynų atvirkštinio projektavimo procesuose. Išnagrinėta vektorizavimo sistemų padėtis Lietuvoje ir pasaulyje. Detaliai aptarti visi vektorizavimo proceso etapai ir problemos. Suformuluoti integrinių grandynų vektorizavimo sistamai keliami tikslai ir reikalavimai. Pasiūlytas optimalus taškinių vaizdų vektorizavimo metodas, leidžiantis panaudojant įprastus vektorizavimo algoritmus gauti integrinių grandynų sluoksnių topologijos projekto duomenis, geriausiai atitinkančius rankiniu būdu atkurtą topologijos projektą. Optimalaus metodo nustatymui buvo vertinama vektorizavimo algoritmo rezultatų priklausomybė nuo pradinio taškinių vaizdų skeleto gavimo algoritmo. Atliktas vektorizavimo metodų ir sistemų tyrimas. Statistiškai įvertinta sukurto vektorizavimo sistemos rezultatų kokybė.

Darbą sudaro 6 dalys: įvadas, literatūros analitinė apžvalga, darbo tikslo ir uždavinių formulavimas, reikalavimai kuriamai programinei įrangai, algoritmų tobulinimas, jų veikimo palyginimas, vektorizavimo metodų ir sistemų tyrimas, apibendrinimas, literatūros sąrašas.

Darbo apimtis – 67 p. teksto be priedų, 72 iliustr., 8 lent., 27 bibliografiniai šaltiniai.

Atskirai pridedamas darbo priedas, kuriame pateikta vektorizavimo sistemos vartotojo sąsaja.

Prasminiai žodžiai: taškinis vaizdas, rastrinis vaizdas, vektorinis vaizdas, vektorizavimas, integriniai grandynai, lustai, atvirkštinis integrinių grandynų projektavimas.

Vilniaus Gediminas technical university
Faculty of **Electronics**
Computer Engineering department

ISBN ISSN
Copies No.
Date-....-....

Electronics Engineering study programme master's thesis.

Title: **Vectorization system for raster images of integrated circuits**

Author **Leonas Jasevičius** Executive prof. hab. dr. **Romualdas Navickas**

Thesis language

☐

Lithuanian

☒

Foreign (English)

Annotation

Integrated circuit layer visual analysis requires raster image vectorization stage to extract layer topology data to computer-aided design (CAD) tools. In this paper vectorization problems of raster IC layer images are presented. Various line extractions from raster images algorithms and their properties are discussed. Situation of vectorization systems in Lithuania and in world is presented. Vectorization process is particularly discussed. Tasks and requirements of created vectorization system are formulated. Optimal raster image vectorization method was developed which allows utilization of common vectorization algorithms to achieve best possible extracted vector data match with perfect manual vectorization results. To develop the optimal method, vectorized data quality dependence on initial raster image skeleton filter selection was assessed. Statistical evaluation of vectorization system results is made, by comparing with results of existing methods and systems.

Structure: introduction, analytical review of literature, goals and tasks formulation, requirements for vectorization system creation, development of algorithms, research of vectorization methods and systems, conclusions and suggestions, references.

Theses consist of 67 p. texts without extras, 72 pictures, 8 tables, 27 bibliographical entries.

Appendix with user interface included.

Keywords: rastre image, vectors image, vectorization, integrated circuits, chips, reverse engineering of integrated circuits

Turinys

IVADAS.....	8
1. LITERATŪROS ANALITINĖ APŽVALGA.....	10
1.1. Atvirkštinis projektavimas.....	10
1.2. Taškiniai ir vektoriniai vaizdai	11
1.2.1. Taškinis vaizdas.....	11
1.2.2. Vektorinis vaizdas	13
1.3. Taškinų ir vektorinių vaizdų palyginimas	13
1.4. Linijų išskyrimo metodai.....	15
1.5. Vektorizavimo sistemos ir padėtis Lietuvoje	18
1.6. Teorinis modelis	19
1.7. Taškinio vaizdo ploninimas.....	20
1.8. Sankirtų žymėjimas	24
1.9. Grandinės sekimas.....	25
1.10. Sankirtų sujungimas	26
1.11. Grandinės apkirpimas.....	28
1.12. Segmentacija.....	29
1.12.1. Keturių žingsnių segmentacija.....	29
1.13. Segmentų sujungimas	33
1.13.1. Perdengimas	33
1.13.2. Sutraukimas	34
1.14. Išvados	35
2. DARBO TIKSLO IR UŽDAVINIŲ FORMULAVIMAS. REIKALAVIMAI KURIAMAI PROGRAMINEI ĮRANGAI.....	36
2.1. Darbo tikslo ir uždavinių formulavimas.....	36
2.2. Tikslumo reikalavimas	36
2.3. Sistemos funkcijos.....	37
2.4. Sistemos struktūrinė schema	37
2.5. Vartotojo poreikiai.....	38
2.6. Bendri apribojimai.....	38

2.7. Sistemos ribos pagal panaudojimo atvejus	39
2.8. Taškinio vaizdo paruošimo vektorizavimui reikalavimai	41
2.9. Taškinio vaizdo vektorizavimo kriterijų parinkimo reikalavimai	42
2.10. Vektorių redagavimo reikalavimai	42
2.11. Vektorių saugojimo reikalavimai	43
2.12. Išvados	43
3. ALGORITMŲ TOBULINIMAS. JŲ VEIKIMO Palyginimas.	45
3.1. Ploninimas Delaunay trianguliacija.....	45
3.2. Ploninimo algoritmas.....	47
3.3. Segmentų verteksų šalinimo algoritmas	49
3.4. Patobulintas vektorizavimo algoritmas	49
3.5. Vektorizavimo rezultatų palyginimas naudojant skirtingus ploninimo algoritmus.....	49
3.6. Išvados	50
4. VEKTORIZAVIMO METODŲ IR SISTEMŲ TYRIMAS	52
4.1. Pradiniai topologijos taškiniai vaizdai.....	52
4.2. Vektorizavimo metodai	54
4.2.1. Canny kontūrų išskyrimas	54
4.2.2. Globalus linijų tinklas.....	55
4.2.3. Eksperimentiniai rezultatai	56
4.3. Vektorizavimo sistemos	61
4.3.1. Eksperimentiniai rezultatai	62
4.4. Išvados	65
5. APIBENDRININIMAS	67
LITERATŪRA	69
PRIEDAS. VEKTORIZAVIMO SISTEMOS VARTOTOJO SĄSAJA	71

IVADAS

Šiandieną integriniai grandynai (lustai) naudojami visose srityse, juos projektuojančios ir gaminančios firmos konkuruoja tarpusavyje vis didėjančiais tempais. Esant tokioms sąlygoms, iškyla įvairios problemos, tokios kaip defektų aptikimo, intelektualios nuosavybės apsaugos, suderinamumo. Taip pat praktikoje kartais iškyla būtinybė išnagrinėti ir patobulinti senas, tačiau vis dar naudojamas sistemas, kurių gamintojai jau pasitraukė iš rinkos, o dokumentacija yra neprieinama. Labai dažnai tokiais atvejais yra reikalingas lusto atvirkštinio projektavimo proceso taikymas.

Atvirkštinis projektavimas – sistemos analizės procesas, siekiant atskirti jos komponentus bei jų tarpusavio ryšius ir sukurti kitos formos arba aukštesnio abstrakcijos lygmens sistemos atitikmenį.

Atvirkštinio projektavimo procese galima išskirti du nepriklausomus etapus: fizinio lygmens ir loginio lygmens analizės. Kiekvienas iš šių etapų reikalauja spręsti visiškai skirtingas užduotis.

Analizuojant fizinį lygmenį, tikslas yra atkurti kuo tikslesnę visų lusto gamybos proceso sluoksnių topologiją. Šis etapas yra reikalingas tuo atveju, jei nėra prieinami fotošablonų duomenys arba jeigu yra ieškoma klaidos lusto gamybos procese. Analizuojant lustus, kurių projektavimo duomenys neprieinami, tenka jų sluoksnių topologiją atkurti tiesiog nagrinėjant pačius lustus. Šiuo atveju yra atliekami tokie technologiniai žingsniai kaip lusto pakuotės atidengimas, lusto cheminis ar plazminis ėsdinimas ir poliravimas, siekiant atidengti kiekvieną iš fotolitografijos proceso metu sudarytų sluoksnių. Atitinkamai kiekvienas sluoksnis yra fotografuojamas, naudojant optinės arba elektroninės mikroskopijos įrenginiu, o gauta taškinių vaizdų informacija yra toliau apdorojama ir analizuojama siekiant gauti vektorinius lusto topologijos duomenis.

Vienas iš atvirkštinio projektavimo proceso etapų yra vektorizavimas. Pastarojo etapo metu gaunami sluoksnių topologijų vektoriniai duomenys. Vektorinių duomenų formatas yra tinkamesnis ir patogesnis nei taškinių vaizdų formatas. Vektorinius duomenis paprasta panaudoti tolesniuose lustų analizės etapuose, kur naudojama įvairi kompiuterinio dizaino (CAD) programinė įranga.

Šio darbo tikslas yra sukurti vektorizavimo sistemą, skirtą integrinių grandynų rastrinių vaizdų vektorizavimui.

Šiame darbe yra aptarta atvirkštinio projektavimo proceso dalis - vektorizavimas, aprašytos pagrindinės vektorizavimo procese vartojamos sąvokos, algoritmai ir metodai. Aptarti kuriamos sistemos reikalavimai ir apribojimai. Pristatoma sukurta taškinių vaizdų vektorizavimo algoritmo taikymo procedūra, leidžianti iš integrinių grandynų sluoksnių taškinių vaizdų atkurti topologijos projekto duomenis. Pasiūlytas patobulintas vektorizavimo ploninimo metodu algoritmas, kuriame

naudojamas Delaunay trianguliacija paremtas taškinių vaizdų ploninimas ir apdorotų tiesių linijų vektorių išskyrimas. Palyginti vektorizavimo metodai ir sistemos pagal rezultatų tikslumą.

1. LITERATŪROS ANALITINĖ APŽVALGA

Šioje darbo dalyje aptarsime atvirkštinio projektavimo procesą, kurio vieną iš etapų sudaro struktūrų vektorizavimas. Šio etapo metu turime gauti kuo tikslesnius duomenis, kadangi jie bus naudojami tolimesniuose proceso žingsniuose. Yra sukurta daug vektorizavimo sistemų skirtų įvairiems uždaviniams spręsti. Jų pagalba galime vektorizuoti linijas, lankus, paskritimus, tekstus ir t.t. Tačiau nėra sukurta sistemos kuri būtų skirta integrinių rastrinių vaizdų vektorizavimui, todėl šiame skyriuje ir aptarsime egzistuojančias projektavimo sistemas, egzistuojančius linijų išskyrimo algoritmus. Šiame skyriuje apžvelgsime vektorizuotų duomenų pranašumą prieš taškinis vaizdus. Taip pat aptarsime vektorizavimo proceso žingsnius, kad vėliau juos galėtumėme patobulinti ar pakeisti taikymo procedūras.

1.1. Atvirkštinis projektavimas

Atvirkštinis projektavimas yra sistemos analizės procesas, siekiant atskirti jos komponentus bei jų tarpusavio ryšius ir sukurti kitos formos arba aukštesnio abstrakcijos lygmens sistemos atitikmenį. Lustų srityje šis procesas dažniausiai naudojamas suprojektuotų produktų funkcinei verifikacijai atlikti, kai iš gautų projektavimo metu lusto sluoksnių gamybos šablonų duomenų gaunamas tranzistorių lygio aprašas, kuris yra konvertuojamas į loginių funkcijų lygmenį, o po to, dar pakėlus abstrakcijos lygmenį, gaunamas tikslus lusto elgsenos modelis, kuris yra naudojamas verifikuoti pradiniam projektuojamo lusto elgsenos modeliui. Tačiau tam tikrais atvejais atvirkštinio projektavimo procesą tenka atlikti naudojant jau pagamintą lustą, neturint jo projektavimo duomenų. Tuomet pradžioje yra analizuojami atidengtų lusto sluoksnių mikroskopu užfiksuoti taškiniai vaizdai, siekiant iš jų atkurti lusto topologijos projekto duomenis.

Atvirkštinio projektavimo procesą galima suskirstyti į du nepriklausomus etapai:

1. Fizinio lygmens analizė.
2. Loginio lygmens analizė.

Kiekvienas iš šių etapų reikalauja spręsti visiškai skirtingas užduotis.

Analizuojant fizinį lygmenį, tikslas yra atkurti kuo tikslesnę visų lusto gamybos proceso sluoksnių topologiją. Šis etapas yra reikalingas tuo atveju, jei nėra prieinami fotošablonų duomenys arba jeigu yra ieškoma klaidos lusto gamybos procese. Analizuojant lustus, kurių projektavimo duomenys neprieinami, tenka jų sluoksnių topologiją atkurti tiesiog nagrinėjant pačius lustus. Šiuo atveju yra atliekami tokie technologiniai žingsniai:

1. Lusto paruošimas (reikalingų sluoksnių atidengimas fotografavimui).
2. Mozaikinis fotografavimas optiniu arba elektroniniu mikroskopu.
3. Nuotraukų sujungimas.

4. Nuotraukų apdorojimas.
5. **Struktūrų vektorizavimas** (ikėlimas į lustų CAD projektavimo įrankius).
6. Tolesnė analizė.

Turint topologijos duomenis skaitmeniniu vektoriniu formatu, yra atliekamas lusto analizės loginio lygmens etapas. Duomenys šiam etapui gali būti gaunami iš prieš tai atlikto lusto topologijos tiesioginio arba atvirkštinio projektavimo.

Automatinėmis atvirkštinio projektavimo priemonėmis iš lusto sluoksnių topologinės informacijos yra atkuriami tranzistorių, ventilių, funkciniai lygmenys ir gautas modelis yra lyginamas su pradinio projektavimui naudotu elgsenos arba RTL (registrų perdavimo) lygmens modeliu.

Toliau darbe detaliai panagrinėsime penktąjį technologinį etapą – struktūrų vektorizavimą. Aptarsime pagrindines vartojamas sąvokas.

1.2. Taškiniai ir vektoriniai vaizdai

Šiame skyrelyje aptarsime pagrindines vartojamas sąvokas.

1.2.1. Taškinis vaizdas

Taškinis vaizdas yra dvimatė monochrominė šviesos intensyvumo funkcija. Ši funkcija aprašoma taip – $f(x, y)$, kur f reikšmė tam tikrose erdvės koordinatėse (x, y) grąžina vaizdo intensyvumą tame taške. Vaizdo intensyvumas gali būti 0 arba 1. Monochrominiai vaizdai taip pat gali būti vadinami dvinariais. Kiekvienas taškinio vaizdo elementas $f(x, y)$ yra vadinamas vaizdo elementu. Vaizdo elementai, kurių reikšmė yra 0, vadinami foniniais vaizdo elementais, o tie kurių reikšmė lygi 1 – objekto vaizdo elementais. Objekto vaizdo elementai yra žymimi juodais kvadratais. Vaizdo elementų poaibis $f(x, y)$ bus žymimas S [1].



1 pav. Dvinaris vaizdas – integrinio grandyno metalo sluoksniai

Vaizdo elementas p , esantis koordinatėse (x, y) , turi keturis gretimus vaizdo taškus – du *vertikalius* ir du *horizontalius*. Gretimų vaizdo taškų koordinatės apskaičiuojamos taip:

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$$

Šis vaizdo elementų rinkinys, vadinamas *keturiais p kaimynais*, bus žymimas $N_4(p)$. Keturi *įstrižainėse* esantys p kaimynai turės tokias koordinatas:

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)$$

Įstrižainėse esantys vaizdo elemento p kaimynai bus žymimi $N_D(p)$. Šie ir prieš tai minėti keturi taškai yra vadinami *aštuoniais p kaimynais* ir yra žymimi $N_8(p)$ [2].

Vaizdo elementų sąjunga yra, naudojama norint nustatyti objektų ribas taškiniame vaizde. Norint nustatyti ar vaizdo taškai yra sąjungoje reikia nustatyti ar jie yra kaimyniniai.

8	4/8	8
4/8	p	4/8
8	4/8	8

1 pav. Vaizdo elemento p sąjungos

Aptarsime tris sujungimo tipus:

1. *4-sąjunga* – du vaizdo taškai p ir q yra 4-sąjungoje jeigu q priklauso aibei $N_4(p)$.
2. *8-sąjunga* – du vaizdo taškai p ir q yra 8-sąjungoje jeigu q priklauso aibei $N_8(p)$.
3. *m-sąjunga (maišyta sąjunga)* – du vaizdo taškai p ir q yra m-sąjungoje jeigu:
 - a. q priklauso aibei $N_4(p)$, arba
 - b. q priklauso aibei $N_D(p)$ ir aibė $N_4(p) \cap N_4(q)$ yra tuščia (tai aibė taškų, kurie yra keturi p ir q kaimynai).

Maišyta sąjunga (m-sąjunga) yra modifikuota 8-sąjunga ir naudojama tam, kad galima būtų pašalinti pasikartojančių kelių sąjungas, kurios nuolat atsiranda naudojant 8-sąjungą.

Jeigu p ir q priklauso taškinio vaizdo poaibiui S , tai p yra sąjungoje su q poaibyje S tik tuomet kai egzistuoja kelias nuo p iki q . Kaimyninių vaizdo elementų seka yra vadinama vaizdo elementų *grandine*, kuri susideda tik iš poaibiui S priklausančių vaizdo elementų. Vaizdo elemento p , priklausančio poaibiui S , sąjungos elementai, taip pat priklausančys poaibiui S , yra vadinami *sujungtais poaibio S komponentais* [3].

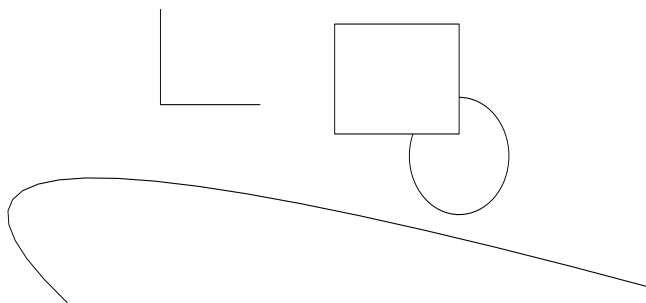
Objektu vadinsime kiekvieną sujungtą taškinio vaizdo poaibio S komponentą (numatytas poaibis S yra visas taškinis vaizdas).

1.2.2. Vektorinis vaizdas

Vektoriniu vaizdu galima vadinti aibę vektorinių esybių ir ryšius tarp tų esybių [4].

Vektorinės esybės dažniausiai yra geometriniai objektai, kurie gali būti išreikšti matematinėmis formulėmis. Pavyzdžiui linija gali būti išreikšta jos galų koordinatėmis arba pagrindo tašku, aukščiu ir ilgiu. Vektorines esybes galima lengvai valdyti bei pritraukti neprarandant vaizdo tikslumo kadangi jos nėra riboto skaičiaus vaizdo taškų kompozicijos.

Ryšiai tarp vektorinių esybių gali būti įvairūs, nuo paprasto grupavimo iki sudėtingesnių pavyzdžiui loginių operatorių.



3 pav. Vektorinis vaizdas

1.3. Taškinio ir vektorinio vaizdų palyginimas

Rastras ir vektorius yra dvi pagrindinės duomenų struktūros naudojamos saugant ir redaguojant skaitmeninius bei grafinius duomenis kompiuteryje. Visų pagrindinių CAD (kompiuterinio dizaino) sistemų veikimas yra paremtas šiomis duomenų struktūromis. [1]

Rastrinis vaizdas yra sudarytas iš atskirų pikselių, ir kiekvienas vaizdo elementas yra susietas su atskiru pikseliu, kurio reikšmė nusako spalvą, aukštį ar identifikacijos numerį.

Vektoriniai duomenys sudaryti iš skaičių bei linijų, abu šie elementai yra matematiškai ir geometriškai susieti. Taškai aprašomi koordinatėmis (pvz.: dvimatis taškas aprašomas taip – (x,y)). Linijos saugomos kaip eilė taškų porų, kur kiekviena pora nusako tiesų linijos segmentą nuo pirmo taško iki antro.

Vektoriniai duomenys užima mažiau atminties negu rastrinis vaizdas todėl, kad rastrinis vaizdas saugo informaciją apie visus pikselius, o vektoriniams duomenims pakanka saugoti tik taškų koordinates.

Kai geometrinės figūros turi būti tiksliai atvaizduotos CAD sistemose pirmenybė yra teikiama vektoriniam atvaizdavimui. Taip yra todėl, kad šis vaizdavimo būdas nėra apribotas erdvės ar pikselio dydžio, taip pat galima naudoti matematines formules kreivių lyginimui ar panašioms uždaviniams. [4]

Kitas vektorinio atvaizdavimo pranašumas yra tas, kad galima didinti arba mažinti vaizdą neprarandant vaizdo kokybės. Taip yra todėl, kad vektorinius duomenis galima atvaizduoti bet kokiame mastelyje fiziškai jų nekeičiant, kai tuo tarpu norint pakeisti rastrinio vaizdo mastelį jį reikia ištempti ir iškreipti. [5]

Dėl išvardintų priežasčių dauguma CAD programinės įrangos gamintojų renkasi vektorinę duomenų struktūrą.

1.4. Linijų išskyrimo metodai

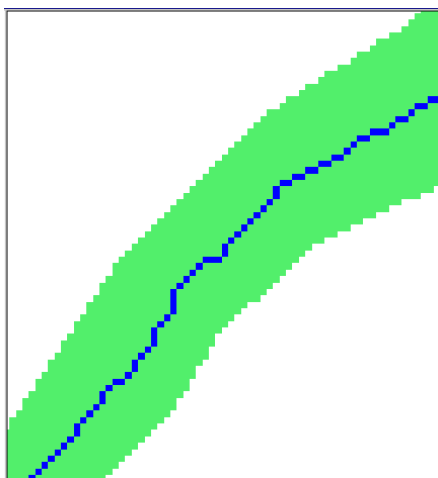
Dauguma nesudėtingų linijų išskyrimo skaitmeniniame vaizde metodų buvo sukurta daugiau nei prieš 30 metų. Šie metodai yra daugumos vektorizavimo algoritmų pagrindas. Linijų išskyrimo metodai yra skirstomi į 6 klases:

1. Paremti transformavimu (Hough transform (HT)) [6].

Metodai transformuoja skaitmeniniame vaizde esančias dideles erdvės struktūras į mažesnes struktūras parametro erdvėje. Šis sprendimas padeda išvengti problemų susijusių su globalios detekcijos problema (kai objektų reikia ieškoti visame skaitmeninio vaizdo plote), ją sumažinant iki mažų objektų paieškos parametro erdvėje. Tokiu būdu šiuo metodu paremti algoritmai kiekvieną vaizdo pikselį turi aplankyti tik vieną kartą.

2. Paremti ploninimu (Thinning) [7].

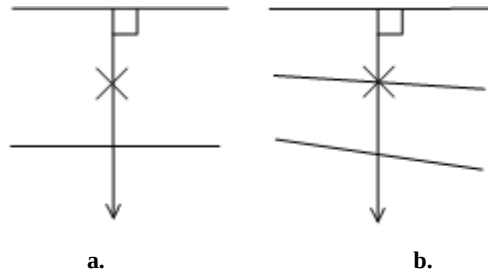
Linijos ploninimo (Skeletonizavimas) metodas paremtas iteraciniu viso rastro pikselių perėjimu. Kiekvieno perėjimo metu yra šalinami ribiniai linijų pikseliai taip galų gale gaunant skeletines linijas. Ploninimu paremtas vektorizavimas yra pavaizduotas žemiau esančiame paveikslėlyje.



4 pav. Vektorių išskyrimas naudojant linijų ploninimo metodą (tamsesnė linija yra gauta atlikus ploninimą)

3. Paremti kontūro radimu (Contour) [8].

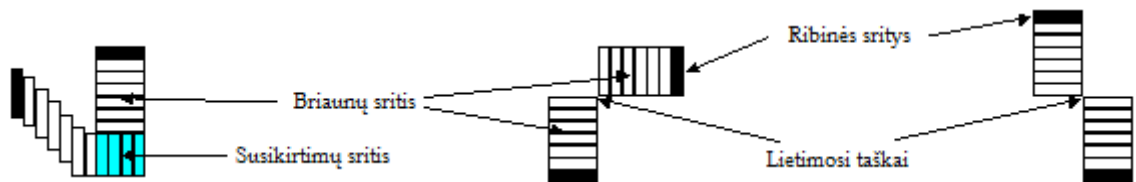
Pagrindinis šių metodų uždavinys yra rasti linijinių objektų kraštus (kontūrus), ir tada apskaičiuoti ašinių linijų taškus tarp abiejų kontūro lygiagrečių. Kontūrų radimu paremtas vektorizavimas yra pavaizduotas žemiau esančiame paveikslėlyje.



5 pav. Vektorių išskyrimas naudojant kontūrų radimą a – vidurio taškas tarp lygiagrečių b – apytikslis vidurio taškas

4. Paremti grafo perėjimu (run-graph) [9].

Naudojama norint atvaizduoti skaitmeninį vaizdą kaip pusiau vektorinę struktūrą. Tai palengvina linijų radimą bei kitus vektorizavimo veiksmus. Grafo perėjimu paremtas vektorizavimas yra pavaizduotas žemiau esančiame paveikslėlyje.

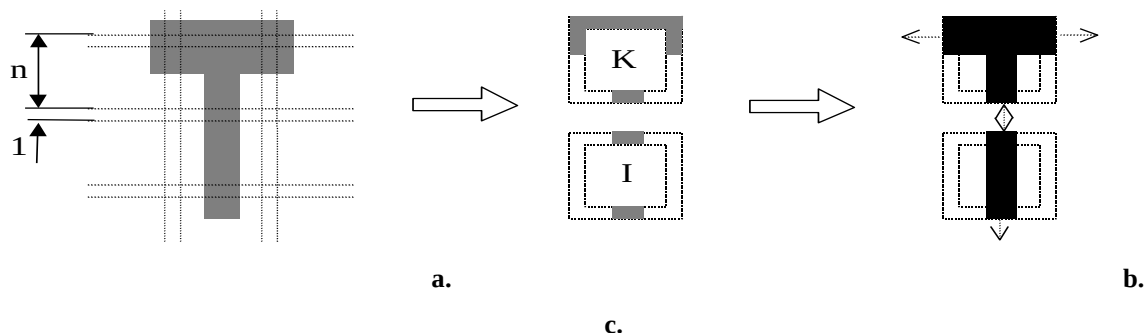


6 pav. Vektorių išskyrimas naudojant grafo perėjimą

Paveikslėlyje pavaizduota keletas grafo perėjimo būdų, kaip pastebime perėjimo kryptis gali būti horizontali bei vertikali. Šiuo atveju analizuojami tik juodi pikseliai.

5. Paremti tinklelio modeliu (Mesh pattern).

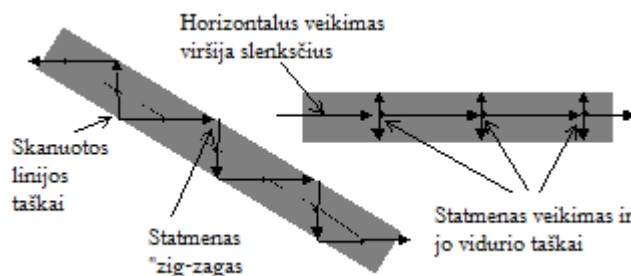
Skaitmeninis vaizdas tinklelio pagalba padalinamas į kvadratus, tada ieškoma charakteringų struktūrų tikrinant tik juodų pikselių išsidėstymą prie kiekvieno tinklelio elemento. Išskyrus charakteringas struktūras yra sudaromas kontrolinis vaizdo žemėlapis, kurį analizuojant yra išskiriami vektoriai. Tinklelio modeliu paremtas vektorizavimas yra pavaizduotas žemiau esančiame paveikslėlyje.



7 pav. Vektorių išskyrimas naudojant tinklelio modelį a – uždėtas tinklelis c – tinklelio elementams suteikti vardai ir sudarytas vaizdo kontrolinis žemėlapis iš dviejų centrinių vaizdo tinklelio elementų c – išskirti vektoriai analizuojant kontrolinį žemėlapi

6. Paremti pikselių sklaidymu (Sparse pixel) [10].

Kitaip žinomas kaip ortogonalus (stačiakampis) zig-zagas (Orthogonal Zig-Zag (OZZ)). Paremtas vieno pikselio kelio sekimu, kelias tampa stačiakampiu kai tik atsiranda ant juodų pikselių ploto ribos. Kiekvieno kelio sekimo vidurinysis taškas bus išskiriamos linijos taškas. pikselių sklaidymu paremtas vektorizavimas yra pavaizduotas žemiau esančiame paveikslėlyje.



8 pav. Vektorių išskyrimas naudojant pikselių sklaidymu paremtą metodą

Visi metodai, išskyrus paremtus transformavimu (HT), yra atliekami pagal tokią schemą:

1. Vidurinėsios ašies atrinkimas, šios procedūros metu yra nustatomi taškai, sudarantys centrinę vektorizaujamos linijos ašį.
2. Linijos sekimas. Pirmojo etapo metu atrinkti taškai yra susiejami į bendrą grandinę, kuri reprezentuoja vektorių.
3. Linijos segmento aproksimacija. Iš sudarytos grandinės pašalinami nekritiniai linijos taškai ir sukuriamas vektorius.

Žemiau pateikta lentelė, kurioje yra palyginti visi 6 linijų išskyrimo metodų tipai:

1 Lentelė. Vektorizavimo metodų palyginimas

Metodas	Sudėtingumas	Linijos kokybė	Linijos storio išlaikymas	Sankirtų atstatymas
transformavimo	kvadratinis	prasta	ne	taip
ploninimo	kubinis	aukšta	ne	ne
kontūrų	kvadratinis	prasta	taip	ne
Grafo-perėjimo	kvadratinis	prasta	taip	ne
tinklelio	linijinis	prasta	taip	taip
Pikselių sklaidymo	linijinis	prasta	taip	taip

1.5. Vektorizavimo sistemos ir padėtis Lietuvoje

Užsienyje yra sukurta nemažai skaitmeninių vaizdų vektorizavimo sistemų. Žemiau yra palyginta keletas populiariausių sistemų pagal pagrindinius vektorizavimo kriterijus:

2 Lentelė. Vektorizavimo sistemų palyginimas

Savybė / Sistema	Vextractor 2.95	Trac-Trix 5.0	EasyTrace 8.0	MagicTracer 2.0	R2V
Kontūrinių linijų išskyrimas	Taip	Taip	Taip	Taip	Taip
Centrinių linijų išskyrimas	Taip	Taip	Ne	Taip	Taip
Apskritimų išskyrimas	Taip	Ne	Taip	Taip	Ne
Lankų išskyrimas	Taip	Ne	Taip	Taip	Ne
Triukšmų šalinimas	Taip	Taip	Taip	Taip	Taip
Norimo regiono	Ne	Ne	Ne	Ne	Taip

pasirinkimas					
Rezultatas DXF formate	Taip	Taip	Ne	Taip	Taip
Teksto atpažinimas	Ne	Taip	Ne	Ne	Taip

Minėtas sistemas galima įsigyti per Internetą, tačiau dėl žemiau pateiktų priežasčių jos gali būti netinkamos Lietuvos rinkai:

- Uždaros išbaigtos sistemos, kurių negalima pildyti, pritaikytos konkrečioms atvejams.
- Ne lietuviška sąsaja su vartotoju - reikalauja nemažai laiko ir išlaidų apmokymams.
- Sistema gali būti nekokybiškai palaikoma.

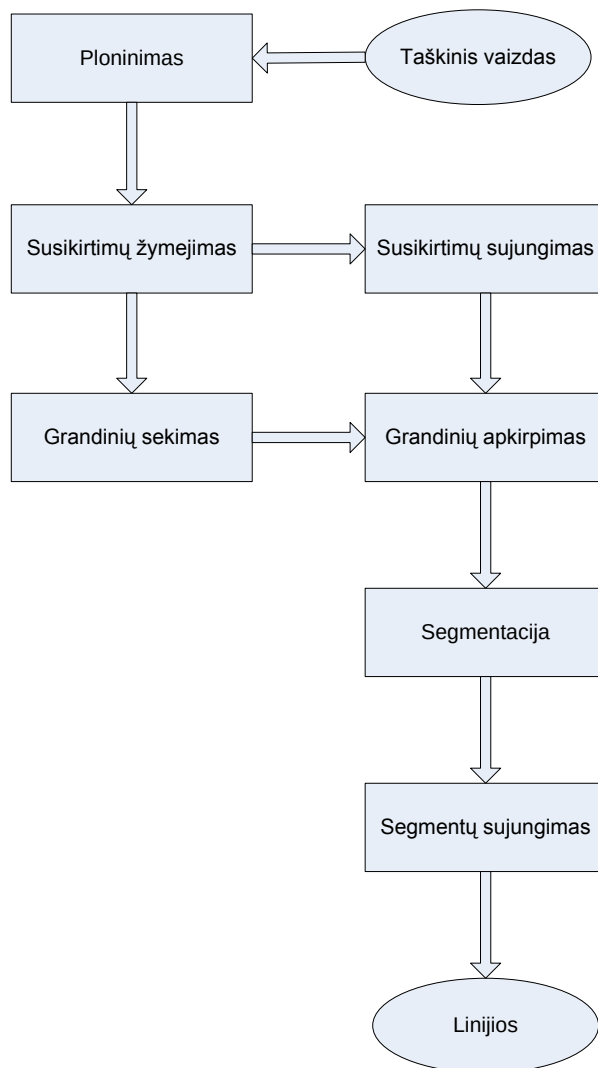
Lietuviškų oficialių programų sistemų, kurios spręstu skaitmeninių vaizdų vektorizavimo problemas nėra.

Šiuo metu Lietuvoje yra trys įmonės, kurios importuoja sistemas galinčias vektorizuoti skaitmeninius vaizdus. Pirmoji iš jų yra UAB „InfoEra“ importuojanti modulį „Raster Design“ [20], šis modulis yra kompanijos „Autodesk“ programinės įrangos (*MAP*, *ACAD*, *LAND* bei *CIVIL*) priedas, antroji UAB „HNIT Baltic GeoInfoServisas“ importuojantis „ArcScan“ modulį skirtą „ArcGIS“ programinei įrangai [21]. Trečioji bendrovė yra „AGA-CAD“ importuojanti priedą „Raster Design“ [22]. Šių sistemų trūkumas yra tai, kad jos yra skirtos didesnių programų sistemų funkcionalumo praplėtimui. Dėl to stipriai padidėja jų bendra kaina. Be to „Raster Design“ yra skirta ne tik skaitmeninių vaizdų vektorizavimui bet ir kietiems veiksams atlikti (pvz.: rastro kalibravimas ir pan.).

Tokia skaitmeninių vaizdų vektorizavimo sistema, kuri nebūtų priklausoma nuo kitų programų sistemų, bei atliktų tik vektorizavimui reikalingas funkcijas būtų žymiai pigesnė bei galėtų nukonkuruoti šiuo metu Lietuvos rinkoje esančias sistemas.

1.6. Teorinis modelis

Automatinio vektorizavimo proceso duomenų srautų diagrama susideda iš 9 žingsnių (9 pav. Vektorizavimo proceso duomenų srautų diagrama).



9 pav. Vektorizavimo proceso duomenų srautų diagrama

1.7. Taškinio vaizdo ploninimas

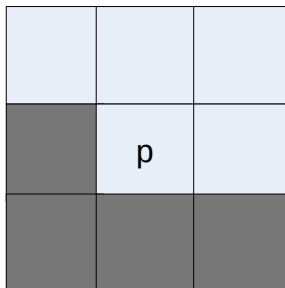
Originalus skaitmeninis vaizdas vektorizavimo procese atlieka įėjimo vaidmenį. Ploninimo (thining) tikslas yra sumažinti taškinio vaizdo objektų plotį iki vieno vaizdo taško. Ploninimo rezultatas turėtų atspindėti tikrųjų objektų, kurie turi būti išskirti iš taškinio vaizdo, formą, struktūrą bei sąjungas.

Egzistuoja du visiškai skirtingi ploninimo tipai:

- *Kontūrinių linijų išskyrimas.*
- *Centrinių linijų išskyrimas.*

Kontūrinių linijų išskyrimas

Kontūrinių linijų (outline) išskyrimo idėja yra rasti taškinio vaizdo objektų kontūrus, jis taip pat žinomas kaip *kontūrų sekimas* [3].

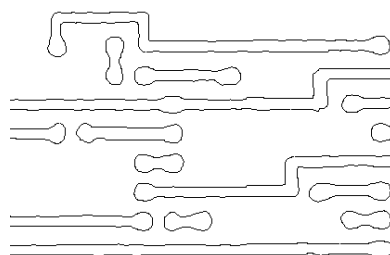


10 pav. Kontūro taškas p

Kontūro vaizdo elementas yra foninis vaizdo elementas, kuris turi bent vieną objekto vaizdo elementą savo aštuoniuose kaimynuose ($N_8(p)$).



11 pav. Originalus vaizdas



12 pav. Vaizdas pritaikius kontūrinių linijų išskyrimą

Centrinių linijų išskyrimas

Centrinių linijų (centerline) išskyrimo metu objektai yra suploninami iki jų *skeleto* (gali būti vadinama *skeletonizavimu*). Objekto skeletas gali būti aprašytas naudojant vidurinėsios ašies transformaciją (*medial axis transformation MAT*), pasiūlytą Blum [3]. Objekto R vidurinėsios ašies transformacija (MAT) su riba (arba kraštu) B yra randama taip: kiekvienam vaizdo taškui p objekte R reikia surasti artimiausią kaimyną riboje B . Jeigu p turi daugiau nei vieną tokį kaimyną, tai sakoma, kad vaizdo taškas priklauso objekto R MAT (skeletui).

Nors taikant MAT galima gauti norimą objekto skeltą, tačiau tai nėra optimaliausias būdas, kadangi jo metu reikia atlikti labai daug skaičiavimų. Reikėtų skaičiuoti atstumus nuo kiekvieno vidinio objekto taško iki kiekvieno objekto ribos taško. Buvo pasiūlyta keletas algoritmų pagerinančių skaičiavimų efektyvumą bei tuo pačiu gaunančių duoto objekto vidurinėsios ašies transformaciją. Tai yra ploninimo algoritmai, kurie trina objekto ribos vaizdo taškus laikantis trijų sąlygų:

1. Negalima trinti galinių taškų.
2. Reikia išlaikyti sujungimus.

3. Nesukelti per didelės objekto erozijos.

Standartinis ploninimo algoritmas, kurį pasiūlė Zhang and Suen [11], susideda iš dviejų žingsnių pritaikomų objekto kraštiniam vaizdo taškams. Toks pats algoritmas yra taikomas kraštinių vaizdo taškų radimui kontūrų sekime. Skirtumas tik tas, kad čia yra ieškoma vaizdo taškų turinčiu bent vieną foninį vaizdo tašką savo aštuoniuose kaimynuose ($N_8(p)$).

p9	p2	p3
p8	p1	p4
p7	p6	p5

13 pav. Kaimyninių vaizdo taškų išsidėstymas naudojant ploninimo algoritmą

Pagal aštuonių kaimynų išdėstymą (13 pav.) kur $p_i \in \{0, 1\}$, pirmasis žingsnis pažymi kraštinį vaizdo tašką p_1 trynimui tik tada kai tenkinamos šios sąlygos:

1. $2 \leq N(p_1) \leq 6$,
2. $S(p_1) = 1$,
3. $p_2 * p_4 * p_6 = 0$,
4. $p_4 * p_6 * p_8 = 0$.

Kur $N(p_1)$ yra kaimyninių vaizdo taško p_1 objektinių vaizdo taškų skaičius (kurių reikšmė nelygi nuliui). Taigi $N(p_1) = p_2 * p_3 * p_4 * p_5 * p_6 * p_7 * p_8 * p_9$. Tuo tarpu $S(p_1)$ yra perėjimų iš 0 į 1 ($0 \rightarrow 1$) skaičius surikiuotoje sekoje $p_2, p_3, \dots, p_8, p_9$. Pavyzdžiui jeigu $N(p_1) = 4$ ir $S(p_1) = 3$ tai turėtume tokį vaizdą kaip pavaizduota 14 pav..

	p	

14 pav. Vaizdas kai $N(p_1) = 4$ ir $S(p_1) = 3$

Antrajame žingsnyje pirmoji ir antroji sąlygos išlieka tokios pačios, tačiau keičiasi likusios dvi:

1. $2 \leq N(p_1) \leq 6$,
2. $S(p_1) = 1$,
3. $p_2 * p_4 * p_8 = 0$,
4. $p_2 * p_6 * p_8 = 0$.

Pirmasis žingsnis yra taikomas kiekvienam objekto krašto vaizdo taškui. Jeigu nors viena iš keturių sąlygų yra netenkinama tai vaizdo taško reikšmė yra nekeičiama. Jeigu visos sąlygos tenkinamos – tai vaizdo taškas yra pažymimas trynimui. Vaizdo taškas nėra trinamas tol, kol nėra apdoroti visi kraštiniai vaizdo taškai. Tai apsaugo nuo duomenų struktūros pasikeitimo algoritmo veikimo metu. Pažymėtieji trynimui vaizdo taškai yra ištrinami (pavyzdžiui jų reikšmė nustatoma į 0) tik tada kai pirmasis žingsnis yra baigtas. Antrasis žingsnis yra taikomas likusiems duomenims lygiai taip pat kaip ir pirmasis žingsnis.

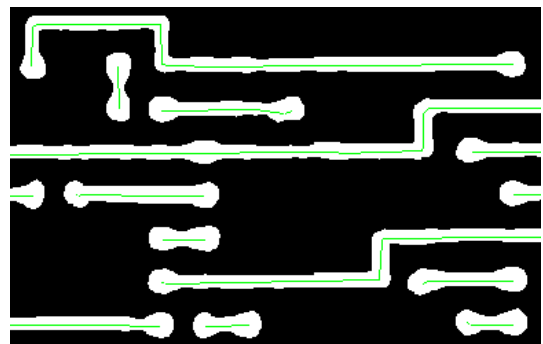
Remiantis ankstesniais samprotavimais galima daryti išvadą, kad viena ploninimo algoritmo iteracija susideda iš šių veiksmų:

1. Pirmojo žingsnio įvykdymas ir trinamų vaizdo taškų sužymėjimas.
2. Trynimui pažymėtų vaizdo taškų ištrynimasis.
3. Antrojo žingsnio įvykdymas ir trinamų vaizdo taškų sužymėjimas.
4. Trynimui pažymėtų vaizdo taškų ištrynimasis.

Ši procedūra yra vykdoma tol, kol neištrinamas nei vienas vaizdo taškas. Po algoritmo įvykdymo taškiniame vaizde lieka tik objektų skeletai.



15 pav. Originalus vaizdas



16 pav. Vaizdas pritaikius centrinių linijų metodą (salotinės linijos- išskirtos centrinės linijos)

Pirmoji sąlyga yra netenkinama tuomet kai kraštinis vaizdo taškas p_1 turi vieną arba septynis objektinius vaizdo taškus savo aštuoniuose kaimynuose. Jeigu p_1 turi tik vieną tokį kaimyną tai reiškia, kad jis yra galinis skeleto vaizdo taškas ir neturi būti trinamas. Jeigu p_1 turi septynis tokius kaimynus tai jo trynimasis galėtų sukelti per didelę objekto eroziją.

Antroji sąlyga yra netenkinama tuomet kai ji yra taikoma taškams, kurie yra vieno vaizdo taško storio linijoje. Tokių vaizdo taškų trynimasis atskirtų skeleto segmentus.

Trečia ir ketvirta sąlygos yra patenkinamos šiose situacijose:

$$(p_4 = 0) \vee (p_6 = 0) \vee (p_2 = 0 \wedge p_8 = 0)$$

Pagal aštuonių kaimynų išdėstymą (13 pav.) vaizdo taškas tenkinantis pirmas dvi bei šią sąlygas gali būti rytinis arba pietinis krašto taškas, arba šiaurės vakarų kampo vaizdo taškas. Bet kuriuo atveju vaizdo taškas p_1 nepriklauso skeletui ir turi būti trinamas.

Panašiai trečia ir ketvirta sąlygos yra patenkinamos šiose situacijose:

$$(p_2 = 0) \vee (p_8 = 0) \vee (p_4 = 0 \wedge p_6 = 0)$$

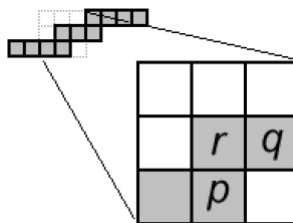
Čia vaizdo taškas tenkinantis šias sąlygas gali būti šiaurinis arba vakarinis krašto taškas, arba pietrytinio kampo taškas. Be to šiaurės rytų kampo vaizdo taškas, kuris turi $(p_2 = 0 \wedge p_4 = 0)$, bei pietvakarių kampo vaizdo taškas, kuris turi $(p_6 = 0 \wedge p_8 = 0)$, taip pat tenkina visas sąlygas.

Šis algoritmas atlieka labai daug skaičiavimų. Iteracijų skaičius priklauso nuo maksimalaus objekto storio. Laikas, per kurį vykdoma kiekviena iteracija, yra proporcingas rastų kraštinių vaizdo taškų skaičiui.

1.8. Sankirtų žymėjimas

Nei vienas iš ploninimo tipų, nei centrinių linijų nei kontūrinių linijų, nenutraukia sankirtų tarp objektų taškiniame vaizde. Rezultate liekantys vieno vaizdo taško pločio objektai yra struktūrinės 8-sąjungos grandinės, kurios gali būti sekamos nesudėtingu būdu. Tačiau reikia žinoti, kad kiekvienas objekto taškas grandinėje turi tik du savo kaimynus priklausančius ($N_8(p)$). Galiniai grandinės vaizdo taškai turi tik po vieną tokį kaimyną. Ši prielaida yra reikalinga tam, kad grandinės sekimo metu neiškiltų dviprasmybių ir kiekvienas sekantis vaizdo taškas galėtų būti unikalčiai identifikuojamas. Taigi sankirtų žymėjimo (junction mapping) funkcijos paskirtis yra garantuoti, kad grandinės sekimo metu bus galima identifikuoti kiekvieną sekantį vaizdo tašką.

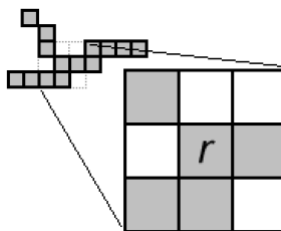
Sankirtų žymėjimas gali būti įsivaizduojamas kaip *taškinio vaizdo grafo* sudarymas. Tai nekryptingas grafas, kur susikirtimo taškai yra grafo viršūnės, o vaizdo taškų grandinės – grafo briaunos. Šis grafas, vaizduojantis taškinio vaizdo objektų susijungimų struktūrą, yra labai svarbus tolimesnėse vektorizavimo proceso fazėse.



17 pav. Vaizdo taškai p ir q yra 4-kaimynai bei 8-sąjungoje su vaizdo tašku r

Bendrai kalbant susikirtimo taškas yra tas, kuris priklauso daugiau negu vienai grandinei. Susikirtimo taško apibrėžimas vaizdo taškų lygyje reikalauja daugiau apsvarstymų. Kiekvienas vaizdo taškas, turintis 3 arba daugiau kaimynų yra potencialus kandidatas būti susikirtimo tašku,

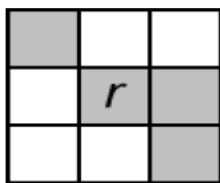
tačiau 17 pav. rodo, kad jis gali tokiu ir nebūti. Figūroje grandinę sudaro trys vaizdo taškai p , q ir r . Vaizdo taškas r turi tris kaimynus priklausančius ($N_8(r)$), tačiau tai akivaizdžiai nėra susikirtimo taškas, kadangi vaizdo taškai p ir q yra 4-kaimynai bei 8-sąjungoje su vaizdo tašku r . Šioje situacijoje, kur vaizdo taškas r turi ne daugiau kaip 3 kaimynus priklausančius ($N_8(r)$), galima pašalinti vaizdo tašką r išlaikant ryšį tarp p ir q . Tokių vaizdo taškų pašalinimą galime vadinti pirmuoju sankirtų žymėjimo algoritmo žingsniu. Reikia tiksliai nustatyti vaizdo taško r kaimynų skaičių, tam kad nebūtų per klaidą pašalinti sankirtų taškai (18 pav.).



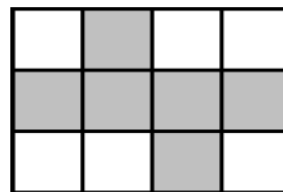
18 pav. Vaizdo taškas r negali būti pašalintas

Sankirtų taškų radimo procesą galime išskaidyti į tris žingsnius:

1. *Pirmas žingsnis.* Bet kuris vaizdo taškas r , turintis tris arba daugiau kaimyninių vaizdo taškų priklausančių ($N_8(r)$) ir kur nei viena jų pora nėra 4-sąjungoje, yra įdedamas į susikirtimo taškų sąrašą (19 pav.). Galiausiai r ir visi jo ($N_8(r)$) priklausančios kaimynai yra pašalinami.
2. *Antras žingsnis.* Šis žingsnis yra reikalingas tam, kad surinkti tuos susikirtimo taškus, kurių neaptiko pirmasis žingsnis (20 pav.). Antrojo žingsnio procedūra yra panaši kaip ir pirmajame žingsnyje, tik panaikinama sąlyga dėl 4-sąjungos, tai reiškia, kad bet kuris vaizdo taškas r , turintis tris arba daugiau kaimyninių vaizdo taškų priklausančių ($N_8(r)$), bus traktuojamas kaip susikirtimo taškas bei pašalintas iš taškinio vaizdo kartu su jo kaimynais.



19 pav. Vaizdo taškas r yra susikirtimo taškas



20 pav. Situacija, kurios neišsprendžia pirmasis sankirtų radimo algoritmo žingsnis

1.9. Grandinės sekimas

Grandinių sekimo (chain tracing) operacijos tikslas yra taškiniame vaizde išskirti atskiras vaizdo taškų atskiras ir paruošti jas segmentacijai. Operacija atliekama dviem žingsniais. Pirmuoju

žingsniu yra išskiriamos visos atviros grandinės, antruoju – uždaros grandinės, kurių neaptiko pirmasis žingsnis, bei tos kurios buvo uždaros bet pasidarė atviros po pirmojo žingsnio įvykdymo.

1. *Pirmas žingsnis – atviros grandinės.*

- Reikia surasti objekto vaizdo tašką p , kuris turi tik vieną kaimyną priklausančią ($N_8(p)$). Radus tokį tašką reikia sukurti sąrašą C , kuris saugos šios grandinės taškus ir pradėti antrąjį etapą (b). Jeigu toks vaizdo taškas nerandamas procedūra yra nutraukiama.
- Įtraukti vaizdo tašką p į sąrašą C ir pašalinti jį iš taškinio vaizdo.
- Surasti vaizdo taško p kaimyną q priklausančią ($N_8(p)$). Jeigu toks rastas nustatyti, kad $p = q$ ir kartoti antrąjį etapą (b), kitu atveju kartoti pirmąjį žingsnį iš naujo.



21 pav. Atviros grandinės sekimas

2. *Antras žingsnis – uždaros grandinės.*

- Reikia surasti objekto vaizdo tašką p . Radus tokį tašką reikia sukurti sąrašą C , kuris saugos šios grandinės taškus ir pradėti antrąjį etapą (b). Jeigu toks vaizdo taškas nerandamas procedūra yra nutraukiama.
- Įtraukti vaizdo tašką p į sąrašą C ir pašalinti jį iš taškinio vaizdo.
- Surasti vaizdo taško p kaimyną q priklausančią ($N_8(p)$). Jeigu toks rastas nustatyti, kad $p = q$ ir kartoti antrąjį etapą (b), kitu atveju kartoti pirmąjį žingsnį iš naujo.



22 pav. Uždaros grandinės sekimas

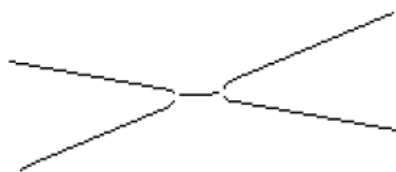
1.10. Sankirtų sujungimas

Sankirtų sujungimo (junction merging) funkcijos tikslas yra koreguoti *taškinio vaizdo grafą*, sukurtą sankirtų žymėjimo funkcijos metu. Funkcija suranda susikirtimo taškus, kurie yra labai arti vienas kito, poras ir pakeičia juos vienu susikirtimo tašku.

Dviejų susikertančių linijų (23 pav.) ploninimas gali duoti du susikirtimo taškus (24 pav.) nors iš tikro mes turime tik vieną susikirtimo tašką.

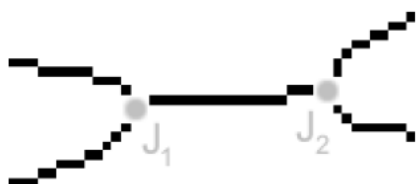


23 pav. Susikertančios linijos (8 vaizdo taškų storio)



24 pav. Vaizdas įvykių ploninimo operaciją

Grandinė tarp J_1 ir J_2 yra tuščia (25 pav.) ir gali būti pašalinta jei neviršija iš anksto nustatyto susikirtimo ilgio J_c .

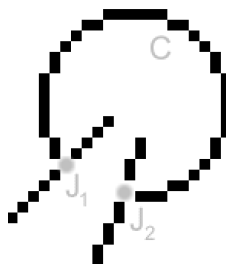


25 pav. Du susikirtimo taškai J_1 ir J_2



26 pav. Apjungtas susikirtimo taškas J

Minimalaus sankirtų ilgio nustatymas yra būtinas tam, kad nebūtu pašalinti išoriniai susikirtimai (27 pav.).



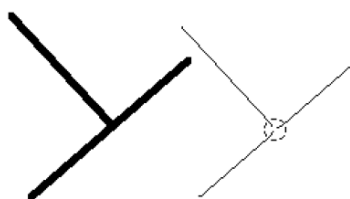
27 pav. Išorinė, du susikirtimus J_1 ir J_2 jungianti, grandinė C negali būti šalinama

Žemiau pateikiamas sankirtų sujungimo algoritmas:

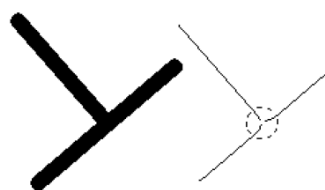
1. Patikrinti atstumus d tarp visų susikirtimo taškų J_1 ir J_2 porų.
2. Jeigu $d < J_e$ tai taškai priklauso tam pačiam susikirtimui.
3. Jeigu tarp rastų susikirtimo taškų yra grandinė kurios ilgis $l < J_e$ tai ją reikia pašalinti. Taipogi reikia atnaujinti *taškinio vaizdo grafą* pašalinant iš jo taškus J_1 ir J_2 ir vietoj jų įdedant naują tašką J . Grandinės priskirtos taškams J_1 ir J_2 turi būti priskirto taškui J .

1.11. Grandinės apkirpimas

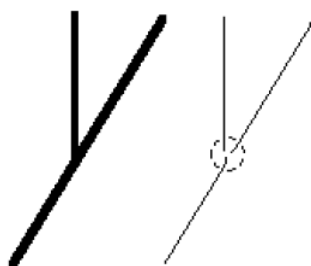
Grandinės apkirpimas (chain trimming) pašalina nepageidautinas deformacijas prie susikirtimo taškų. Deformacijos atsiranda ploninimo algoritmo metu, kadangi jis paprasčiausiai išskiria regiono viduriniąją ašį. Šios figūrų deformacijos bei iškraipymai yra paplitę tik prie susikirtimo taškų ir proporcingi susikirtimo plotui. Deformacijos apimtis priklauso nuo susikertančių objektų skaičiaus, jų pločio bei susikirtimo kampo. Figūros žemiau vaizduoja deformacijos efektą, kurį sukelia susikirtimai. Tokiu atveju pakaktų pašalinti keletą grandinės galo, kuris priklauso susikirtimui, taškų.



28 pav. τ -susikirtimas. 4 vaizdo taškų plotis



29 pav. τ -susikirtimas. 12 vaizdo taškų plotis



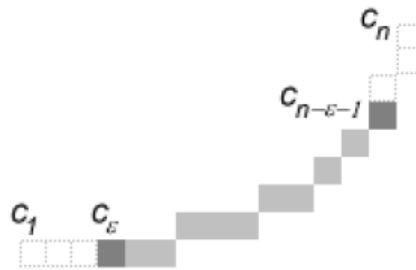
30 pav. λ -susikirtimas. 4 vaizdo taškų plotis



31 pav. λ -susikirtimas. 12 vaizdo taškų plotis

Žemiau pateiktas algoritmas aprašo procedūrą, kuri šalina grandinių galus. Reikia žinoti jog tokiu būdu gali būti pašalintos trumpos grandinės, jeigu abu jų galai yra pašalinami daugiau kaip per pusę grandinės ilgio.

1. Kiekvienai grandinei $C = c_1 c_2 \dots c_{n-1} c_n$, kur c_i yra vaizdo taškai grandinėje, reikia vykdyti pirmą ir antrą žingsnius.
2. Išgauti du susikirtimo taškus artimiausius grandinės galo taškams c_1 ir c_n . Grandinė yra nepririšta nei prie vieno susikirtimo taško jeigu nors vienas iš rastų susikirtimo taškų yra nutolęs toliau kaip per tu vaizdo taškus nuo jam priklausančio grandinės galo. Nepririštos prie susikirtimo taškų grandinės yra nemodifikuojamos.
3. Jeigu randama pririšta prie susikirtimo taško grandinė ir iš anksto nustatytas parametras *apkirpimo ilgis* $\varepsilon \geq \lceil l/2n \rceil$ tuomet grandinė C yra pašalinama iš *taškinio vaizdo grafo*. Jeigu $\varepsilon < \lceil l/2n \rceil$ tuomet grandinė C yra pakeičiama tokia $C' = c_{\varepsilon+1} c_{\varepsilon+2} \dots c_{n-\varepsilon-1} c_{n-\varepsilon}$. Kitaip tariant mes pašaliname ε vaizdo taškus iš grandinės galų (32 pav.).



32 pav. Grandinė $c_1 c_2 \dots c_{n-1} c_n$ yra apkirpta abiejuose galuose kai $\varepsilon = 3$

1.12. Segmentacija

Segmentacijos tikslas yra suskaidyti grandines į mažesnius prasmingus *segmentus*, kurie gali būti paverčiami į geometrines primityvas. Kiekvienos grandinės segmentacijos rezultatas – sujungtų linijų ir lankų sąrašas.



33 pav. Vaizdo taškų grandinė



34 pav. Segmentuota grandinė

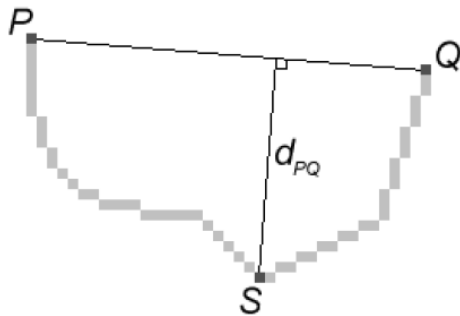
1.12.1. Keturių žingsnių segmentacija

Pirmieji du šio metodo žingsniai padalina grandinę į tiesias linijas. Kiti du žingsniai bando tas linijas paversti lankais. Jeigu norimam rezultatui pakanka tik linijų, tai paskutiniai du šio metodo žingsniai gali būti nevykdomi [12].

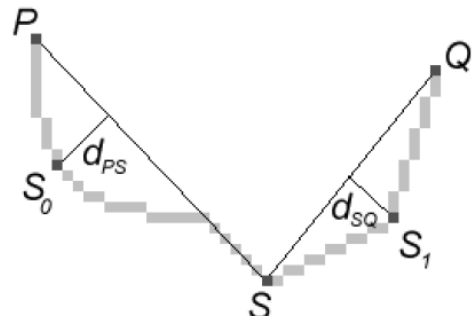
Linijų surinkimas

Linijų surinkimo (line fitting) tikslas yra rasti geriausią būdą kaip padalinti grandinę į sujungtas linijas. Tai yra daroma dalinant grandinę į dvi dalis tol kol iš anksto nustatytas kriterijus yra patenkinamas. Kriterijus vadinamas *išlenkimo tolerancija* ε , kuris nusako kokio tiesumo turi būti grandinė, kad ji būtų laikoma tiesia linija.

Reikia surasti ilgiausią statmenį tarp grandinės ir linijos jungiančios abu grandinės galus. Rastas grandinės taškas bus laikomas dalinimo tašku, kuris dalina grandinę į dvi dalis. Tarkim turim grandinę PQ (35 pav.) kur maksimalus statmuo yra d_{PQ} vaizdo taške S . Sakykime, kad $d_{PQ} > \varepsilon$ tuomet reikalingas dar vienas grandinės dalinimas vaizdo taške S . Taigi toliau bus dalinamos grandinės PS ir SQ . Jeigu $d_{SQ} \leq \varepsilon$, tai grandinė SQ yra laikoma tiesia linija ir jos toliau dalinti nebereikia. Linija SQ gali būti pridėta į sąrašą su *Lowe reikšmingumo atributu* σ , kuris apskaičiuojamas taip: $\sigma_{SQ} = d_{SQ}/||SQ||$. Grandinės PS atveju reikalingas tolimesnis dalinimas jeigu $d_{PS} > \varepsilon$.

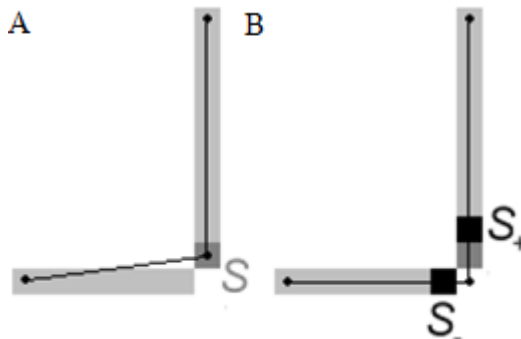


35 pav. Grandinės PQ dalinimo taškas S



36 pav. Grandinių PS ir SQ dalinimas

Sudėtingesnis linijų surinkimo metodas yra žinomas kaip *adaptyvus linijų surinkimas*. Šis metodas yra tikslesnis nei anksčiau aprašytas. Vietoj vieno grandinės dalinimo taško S čia yra naudojami du taškai S_- ir S_+ . Šie taškai yra gretimi taškui S (37 pav.).

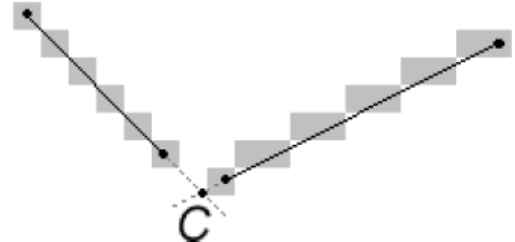


37 pav. Adaptivaus linijų surinkimo privalumas. A – linijų surinkimas; B – adaptyvus linijų surinkimas.

Adaptivus linijų surinkimo metodas gali suskaidyti linijas dėl to, kad išmeta tikruosius grandinės dalinimo taškus, kurie yra grandinės dalis. Sąjungos gali būti atstatytos projektuojant linijas į jų susikirtimus pvz.: 39 pav. vaizduoja linijų susikirtimo projekciją į tašką C .



38 pav. Adaptivūs grandinės dalinimo taškai



39 pav. Atskirti linijos segmentai turi būti sujungti taške C

Linijų derinimas

Linijų derinimo (line blending) metu yra analizuojamas linijų sąrašas gautas pirmo žingsnio metu. Linijos, kurios tenkina *Lowe* reikšmingumo kriterijų, gali būti pakeistos ilgesnėmis. Žemiau pateiktas algoritmas aprašo linijų derinimo procesą.

1. Iš linijų sąrašo, gauto pirmojo žingsnio metu, pasirinkti visas gretimų linijų poras PQ ir QR .
2. Apskaičiuoti statmenį d_{PR} bei *Lowe* reikšmingumo atributą σ_{PR} .
3. Jeigu $d_{PR} \leq \varepsilon$ bei $w\sigma_{PR} < \sigma_{PQ}$ ir $w\sigma_{PR} < \sigma_{QR}$ kur $w \in (0,1]$, tuomet galima suderinti linijas PQ ir PR bei pakeičiant jas linija PR .
4. Vykdyti pirmąjį žingsnį tol kol nebesuderinama nei viena linija.

Parametras w yra vadinamas *derinimo svoriu* kontroliuoja linijų derinimą nuslopindamas originalių linijų, gautų pirmojo žingsnio metu, *Lowe* reikšmingumo atributų reikšmes.



40 pav. Linijų segmentai, gauti įvykužius antrąjį linijų derinimo algoritmo žingsnį

Lankų surinkimas

Lankų surinkimo procedūra (arc fitting) bando linijų, gautų antrojo žingsnio metu, poaibius paversti lankais taip sukurdamas sujungtų linijų ir lankų sąrašą. Kad rezultatas būtų patenkinamas

pirmajame žingsnyje reikalinga žema kreivumo tolerancija ε , kadangi taip bus gauta daugiau linijų. Žemiau vartojamas terminas *linijų eilutė* reiškia sujungtų linijų sąrašą.

Lankų surinkimo metu linijų eilutė yra dalinama į dvi dalis, tol kol iš anksto žinomas kriterijus bus patenkintas. Kriterijus yra paremtas linijų iš linijų eilutės, deviacija nuo įsivaizduojamo lanko. Žemiau pateiktas algoritmas yra taikomas kiekvienai antrame žingsnyje gautai linijų eilutei. Tarkime kad du galiniai linijų eilutės taškai yra S_1 ir S_n .

1. Pasukti linijų eilutę S_1S_n taip kad linija S_1S_n būtų statmena X ašiai. Tuomet perkelti linijų eilutę taip, kad linijos S_1S_n vidurinis taškas M atsirastų koordinačių ašies pradžioje $[0, 0]$.
2. Surasti išlinkio centrą O . Tai atliekama remiantis dalinimo pusiau metodu. Tarkime kad $y_{min} = 0$, $y_{max} = \{y_i \mid S_i = [x_i, y_i], i = 2 \dots n - 1\}$. Parametras δ yra iš anksto nustatyta reikšmė nusakanti dalinimo pusiau tikslumą.
 - a. Tarkim $y = \frac{1}{2}(y_{min} + y_{max})$, $O = [0, y]$, $O^+ = [0, y + \delta]$ ir $O^- = [0, y - \delta]$.
 - b. Apskaičiuojam *radialinę nuokrypį* $\partial R^+ = \sum |R_1^+ + R_i^+|$ ir $\partial R^- = \sum |R_1^- + R_i^-|$ ($i = 2 \dots n$) kur $R_i^+ = ||O^+S_i||$ ir $R_i^- = ||O^-S_i||$.
 - c. Jeigu $\partial R^+ \approx \partial R^-$ tuomet vykdome trečią žingsnį.
 - d. Jeigu $\partial R^+ < \partial R^-$ tuomet nustatome $y_{max} = y$ ir kartojame iš pradžių (a).
 - e. Nustatome $y_{min} = y$ ir kartojame iš pradžių (a).
3. Apskaičiuojame nuokrypį d' tarp linijų eilutės S_1S_n ir įsivaizduojamo lanko (centras O , spindulys $R = ||OS_1||$). d' bus lygi ilgiausio statmens, tarp bent kurio linijos iš linijų eilutės taško ir bet kurio lanko taško, ilgiui.
4. Jeigu $d' > \varepsilon$ tuomet daliname linijų eilutę taške $T = S_j$, kur $|R_1 - R_j| = \max\{|R_1 - R_j| \mid j = 1 \dots n\}$ ir taikome šį algoritmą linijoms S_1T bei TS_n . Jeigu $d' \leq \varepsilon$ tai linijų eilutė laikoma lanku, kurio centras O' , o spindulys R . O' yra apskaičiuojamas taikant inversinę pirmojo žingsnio transformaciją taškui O . Galiausiai lankas yra įdedamas į sąrašą kartu su Lowe reikšmingumo atributu, kuris apskaičiuojamas taip: $\sigma' = d'/w$, kur w yra lanko ilgis.

Lankų derinimas

Lankų derinimo (arc blending) paskirtis yra tokia pati kaip ir linijų derinimo. Žemiau pateikiamas lankų derinimo algoritmas.

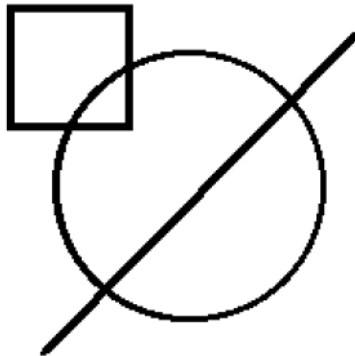
1. Iš lankų sąrašo, gauto trečiojo žingsnio metu, pasirinkti visas gretimų lankų poras PQ ir QR .
2. Apskaičiuoti nuokrypį d'_{PR} bei Lowe reikšmingumo atributą σ'_{PR} .

3. Jeigu $d'_{PR} \leq \varepsilon$ bei $w'\sigma'_{PR} < \sigma'_{PQ}$ ir $w'\sigma'_{PR} < \sigma'_{QR}$ kur $w \in (0,1]$, tuomet galima suderinti lankus PQ ir PR pakeičiant juos lanku PR .
4. Vykdyti pirmąjį žingsnį tol kol nebesuderinamas nei vienas lankas.

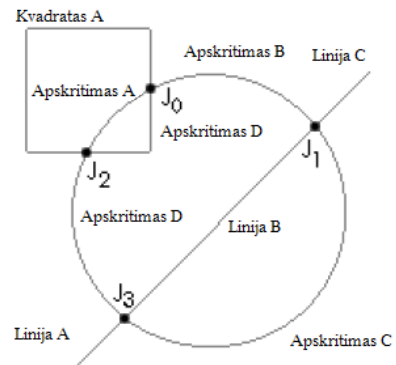
1.13. Segmentų sujungimas

Sankirtų žymėjimo metu grandinės buvo išskaidytos. Kadangi objektų vientisumas yra labai svarbus tai gretimi segmentai, gauti segmentacijos proceso metu, turi būti sujungti. Segmentų sujungimas niekaip nekeičia *taškinio vaizdo grafo*.

Panagrinėkime paprastą pavyzdį (41 pav.) su trimis besikertančiais geometriniais objektais. Po susijungimų žymėjimo ir grandinės sekimo šie objektai yra išskaidomos (42 pav.).



41 pav. Originalus vaizdas

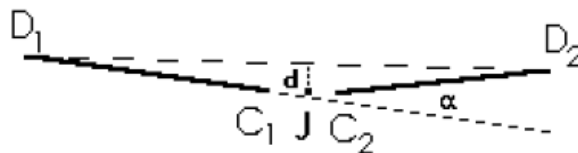


42 pav. Struktūriniai vaizdo komponentai

Aptarsime du segmentų sujungimo metodus. Teisingas segmentų sujungimo rezultatas pasiekiamas tik tuomet kai naudojami abu metodai. Pirmiau taikomas *perdengimas (spanning)* po kurio seka *sutraukimas (contracting)*. Segmentų sujungimo procedūra taikysime tik linijiniams segmentams.

1.13.1 Perdengimas

Tarkime, kad turime dvi gretimas linijas C_1D_1 ir C_2D_2 , kurios yra gretimos susikirtimo tašui J . Gretimi šių linijų taškai bus C_1 ir C_2 (43 pav.).



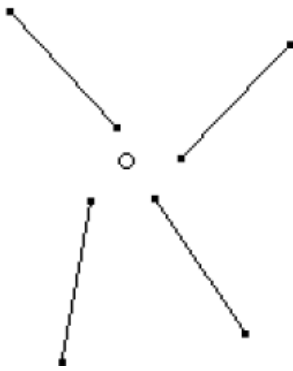
43 pav. Susikirtimo tašui J gretimos linijos C_1D_1 ir C_2D_2

Ši linijų pora yra sujungiama į *perdengiančią (span) liniją* D_1D_2 , jeigu statmens d ir kampo α nuokrypiai yra mažesni už iš anksto nustatytus slenksčius. Žemiau yra aprašytas perdengimo algoritmas.

1. Tarkime $S_j = \emptyset$ (perdengiančių linijų aibė) bei vykdome antrąjį žingsnį kiekvienam susikirtimo taškui J bei jam gretimų linijų aibei L_j .
2. Sujungiam labiausiai gretimų linijų poras $C_1D_1 \in L_j$, $C_2D_2 \in L_j$ kurios tenkina statmens ir kampo nuokrypius. Į aibę S_j įdedame perdengiančią liniją D_1D_2 (tariant, kad susikirtimo taškui J gretimi taškai yra C_1 ir C_2) ir kartojame antrą žingsnį tol kol nebesukuriame nei viena nauja perdengianti linija.

1.13.2 Sutraukimas

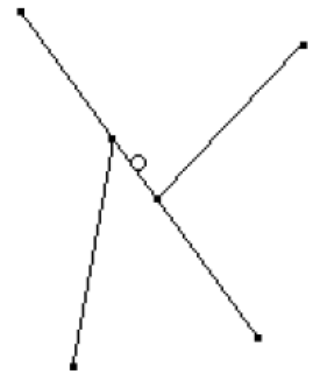
Išskirsime du sutraukimo (contracting) atvejus. Pirmuoju atveju, kai nerandama nei viena perdengianti linija arba kai perdengimas nebuvo naudojamas, linijos yra sujungiamos jų susikirtimo taškuose. Antruoju atveju, kai buvo rasta nors viena perdengianti linija likusios linijos yra pratęsiamos iki jų susikirtimo su perdengiančia linija.



44 pav. Originalus išdėstymas



45 pav. Sutraukimas be perdengimo



46 pav. Sutraukimas su perdengimu

Žemiau pateiktas algoritmas aprašo abu sutraukimo atvejus.

1. Kiekvienam susikirtimo taškui J vykdyti antrą žingsnį.
2. Jeigu $S_j = \emptyset$ tuomet vykdyti trečią žingsnį (pirmasis atvejis), kitu atveju (antrasis atvejis) kiekvienai linijai $CD \in L_j$, reikia surasti linijų CD ir $s \in S_j$ susikirtimo tašką P taip, kad atstumas $\|PC\|$ būtų minimalus (taškas C yra gretimas susikirtimo taškui J). Pakeisti liniją CD į liniją PD .
3. Pakeisti kiekvieną liniją $CD \in L_j$ į liniją JD (taškas C yra gretimas susikirtimo taškui J).

1.14. Išvados

1. Šiame skyriuje apžvelgėme atvirkštinio projektavimo proceso etapus. Detaliau išnagrinėjome vektorizavimo etapą, kuriame integrinio grandyno rastrinį vaizdą, gautą fotografuojant mikroskopu, turime paversti vektoriniu vaizdu. Pastarieji vaizdai naudojami, nes jie užima mažiau vietos, jų kokybė nepriklauso nuo vaizdo didinimo ar mažinimo, galima taikyti geometrines formules, taip pat vektorinius duomenis galima naudoti įvairiose CAD sistemose.
2. Aptarėme ir palyginome pagrindinius linijų išskyrimo metodus. Iš palyginimo matome (1 lentelė), kad aukščiausia linijos kokybė gaunama naudojant vektorizavimo algoritmus, kurie paremti linijų ploninimu. Todėl šiuos algoritmus ir naudosime kuriamai sistemai.
3. Palyginome egzistuojančias vektorizavimo sistemas. Visos sistemos atpažįsta didelį objektų tipų kiekį, tačiau nei viena nėra pritaikyta konkrečiam uždaviniui spręsti, todėl naudojant šias sistemas dažnai nepavyksta gauti reikiamo tikslumo ir kokybės. Kuriama sistema bus skirta konkretiems atvejams spręsti, gaunant maksimalų tikslumą bei įvertinant įvairias problemines situacijas. Lietuvoje nėra oficialių vektorizavimo sistemų, įmonės prekiaujančios šiomis sistemomis, yra užsienio gamintojų produktų perpardavinėtojai.
4. Apžvelgėme vektorizavimo procesą, kuris susideda iš devynių žingsnių: taškinio vaizdo pasirinkimas, ploninimas, sankirtų žymėjimas, sankirtų sujungimas, grandinės sekimas, grandinės apkirpimas, segmentacija, segmentų sujungimas, vektorių išskyrimas. Vektorizavimo rezultatai labiausiai priklauso nuo taškinio vaizdo ploninimo algoritmo. Šiame skyriuje apžvelgėme standartinį ploninimo algoritmą, kurį bandysime tobulinti kuriamoje vektorizavimo sistemoje.

2. DARBO TIKSLO IR UŽDAVINIŲ FORMULAVIMAS. REIKALAVIMAI KURIAMAI PROGRAMINEI ĮRANGAI

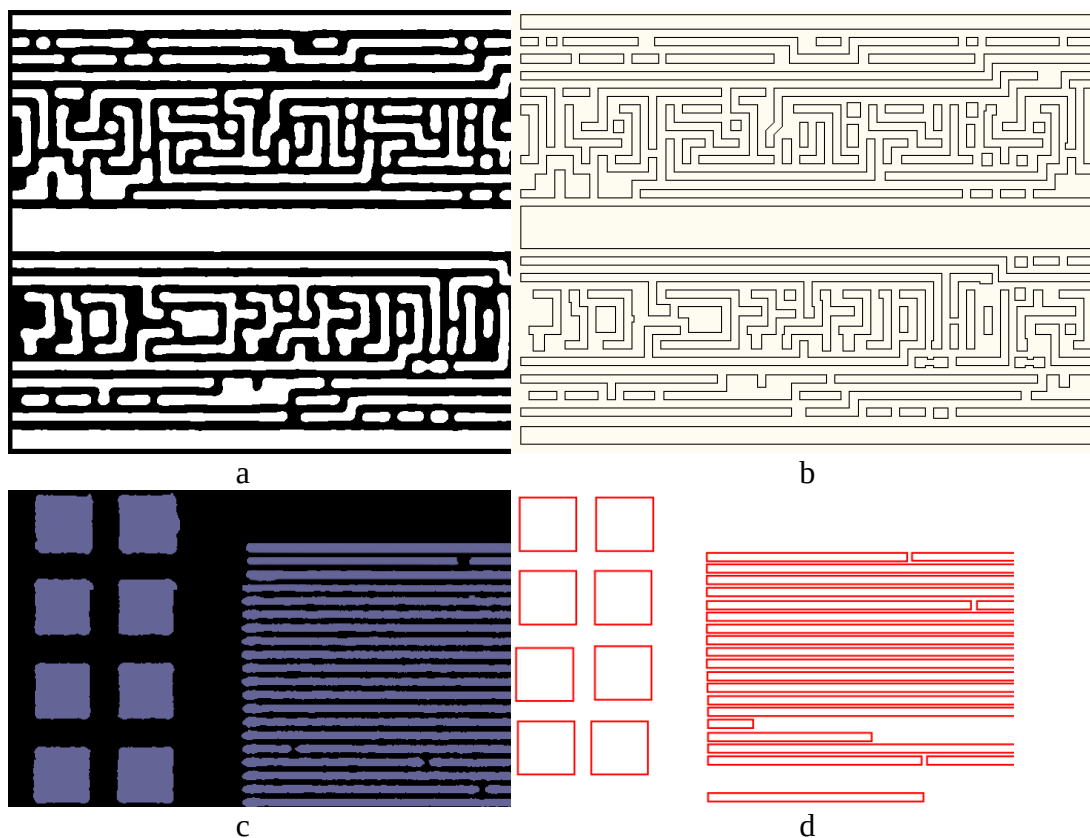
2.1. Darbo tikslo ir uždavinių formulavimas

Šio darbo tikslas yra sukurti vektorizavimo sistemą, skirtą integrinių grandynų rastrinių vaizdų vektorizavimui.

Pagrindinis uždavinys, sukurti sistemą gebančią vektorizuoti taškinius vaizdus. Naudojantis šia programa bus galima greitai vektorizuoti integrinių grandynų metalų sluoksnių rastrinius vaizdus, to pasakoje bus galima gautus vektorinius vaizdus panaudoti tolimesniuose atvirkštinio projektavimo proceso etapuose. Vektorizavimo rezultatai vektoriai galės būti išsaugomi į DXF formato failą, kurį palaiko dauguma kompiuterinio projektavimo (CAD) sistemų.

2.2. Tikslumo reikalavimas

Kuriamos sistemos pagrindinis reikalavimas yra reikalavimas tikslumui, t.y. vektorizuoti duomenys turi būti kuo artimesni idealiems duomenims, gautiems taškininį integrinių grandynų vaizdą vektorizuojant rankiniu būdu. Pastaruoju būdu gauti duomenys vadinami etaloniniais, tokių vektorinių vaizdų pavyzdžiai, kartu su pradiniais taškiniais vaizdais, pateikti 47 paveiksle.



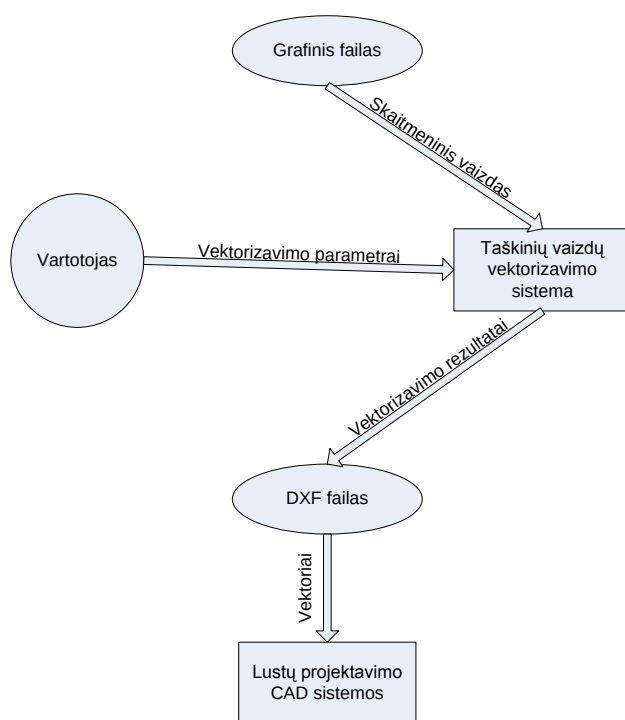
47 pav. Pradiniai taškiniai vaizdai (a) ir (c), etaloniniai vektoriniai vaizdai, gautais rankiniu būdu (b) ir (d)

Tikslumas įvertinamas geometriniais bei statistiniais dydžiais, lyginant kuriama sistema gautą vektorinį vaizdą su etaloniniu vaizdu. Įvertinamas segmentų galų (verteksų) skaičius, vektorių segmentų ilgiai, gautų rezultatų verteksų nuokrypių, nuo etaloninio vektorizuoto vaizdo, vidurkiai bei dispersijos.

2.3. Sistemos funkcijos

Sistemos vartotojas pasirenka taškinį vaizdą, kurį nori vektorizuoti. Tai gali atlikti naudodamas egzistuojantį grafinį failą, gautą iš fotografavimo optiniu arba elektroniniu mikroskopu. Užkrovus taškinį vaizdą vartotojas gali jį apdoroti: pritaikyti nelinijinius bei linijinius filtrus ar pasirinkti vektorizuojamą vaizdo ribą. Pasirinkus vektorizavimo metodą (išskiriamos centrinės linijos arba kontūrinės linijos) vaizdą galima vektorizuoti. Vektorizavimo metu bus sukurti vektorizavimo rezultatai (vektoriai), kuriuos vartotojas galės redaguoti (t.y. keisti jų spalvą, poziciją, storį). Vektorizavimo rezultatai galės būti išsaugoti DXF formato faile.

2.4. Sistemos struktūrinė schema



48 pav. Sistemos struktūrinė schema

Sistema duomenis gauna iš grafinio failo, kuris yra gaunamas mikroskopu fotografuojant metalo sluoksnius. Gauti vektoriai naudojami lustų projektavimo CAD sistemose, pavyzdžiui „L-Edit“.

2.5. Vartotojo poreikiai

Reikalavimai vartotojo požiūriu:

- Aiški ir paprasta sąsaja su vartotoju;
- Galimybė įgyti ir pasirinkti skaitmeninį vaizdą;
- Galimybė pasirinkti apdorojamą skaitmeninio vaizdo regioną;
- Galimybė pasirinkti vektorizavimo metodą;
- Galimybė paruošti skaitmeninį vaizdą vektorizavimo procedūrai;
- Greitas skaitmeninio vaizdo vektorizavimas;
- Galimybė redaguoti vektorizavimo rezultatus realiu laiku;
- Galimybė išsaugoti vektorizavimo rezultatus DXF formate.

2.6. Bendri apribojimai

Standartai :

- Sistema turi palaikyti populiariausius grafinių failų suglaudavimo formatus:

3 Lentelė. Palaikomi grafinių failų formatai

Tipas	Aprašymas	Bylos plėtinys
BMP	Windows and OS/2 bitmap	.bmp, .dib, .rle
JFIF, JPEG	Joint Photographics Expert Group	.jpg or .jpeg
PNG	Portable Network Graphic	.png
TGA	True Vision Raster-Based Data Format	.tga
TIFF	Tagged Image File Format	.tif or .tiff

- *DXF (Drawing eXchange Format)* – brėžinio formatas, kuriame turi būti išsaugomi vektorizavimo rezultatai (vektoriai).

Techninės įrangos platforma :

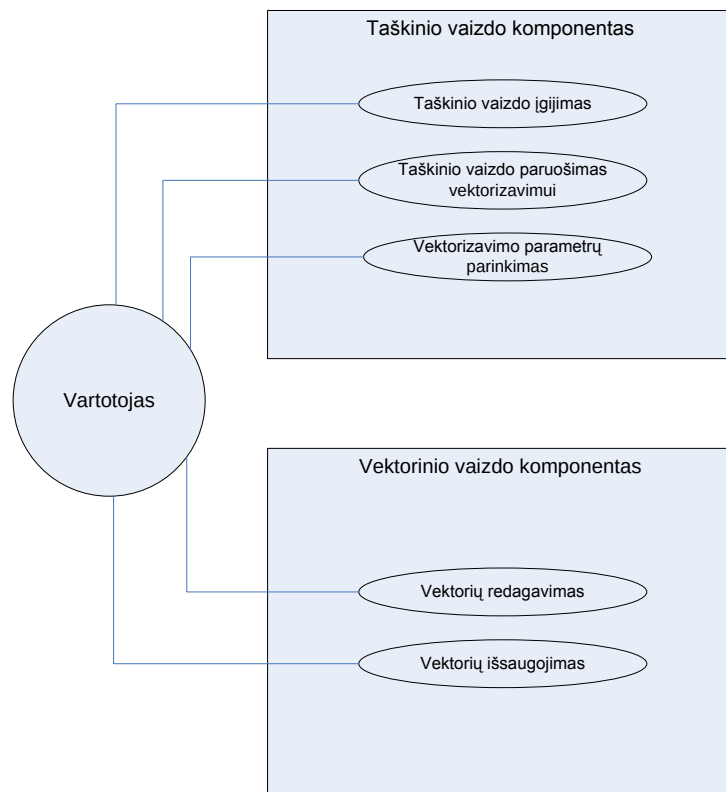
- Operacinė sistema - *Microsoft Windows Me/2000/XP/ Vista* bei vėlesnės versijos

- Aparatinė įranga: kompiuteriai *IBM PC* tipo.

2.7. Sistemos ribos pagal panaudojimo atvejus

Panaudojimo atvejai:

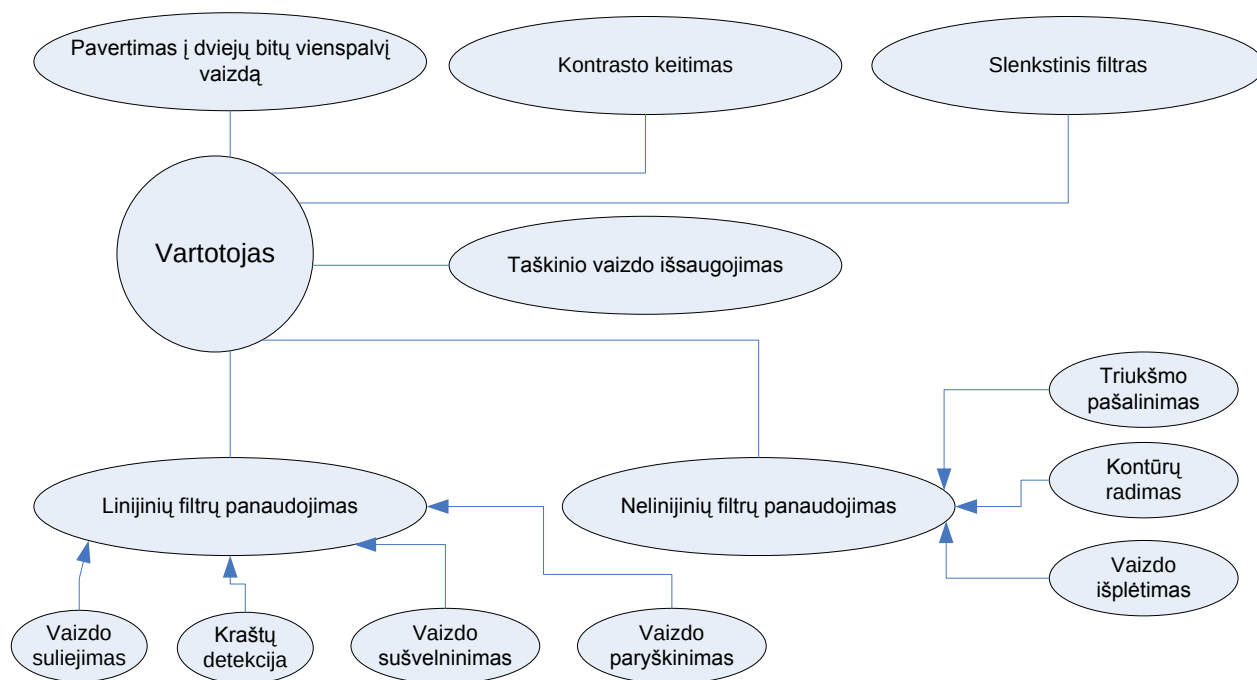
1. *Taškinio vaizdo įgijimas.* Šis panaudojimo atvejis leidžia vartotojams sukurti norimo formato grafinį failą taškinio vaizdo fotografavimo metu, bei jį atsidaryti.
2. *Taškinio vaizdo išsaugojimas* – šis panaudojimo atvejis leidžia vartotojams išsaugoti pakoreguotą taškinį vaizdą į norimo formato grafinį failą.
3. *Taškinio vaizdo redagavimas* – šis panaudojimo atvejis leidžia vartotojams koreguoti taškinį vaizdą taikant linijinius bei nelinijinius filtrus, arba keičiant spalvų gamą.
4. *Taškinio vaizdo vektorizavimas.* Šis panaudojimo atvejis apima taškinio vaizdo vektorizavimo procesą, kurio metu yra sukuriami vektorizavimo rezultatai (vektoriai).
5. *Vektorizavimo tipo pasirinkimas.* Suteikia galimybę vartotojams pasirinkti norimą vektorizavimo metodą. Galima rinktis centrinių arba kontūrinių linijų vektorizavimą bei kitus kriterijus.
6. *Vektorizuojamo regiono pasirinkimas.* Šis panaudojimo atvejis suteikia galimybę vartotojui pasirinkti taškinio vaizdo regioną, kuris bus vektorizuojamas.
7. *Vektorių redagavimas.* Šis panaudojimo atvejis suteikia galimybę vartotojui redaguoti vektorizavimo rezultatus (vektorius), keičiant jų spalvą, sluoksnį ar taškų koordinates.
8. *Vektorių Išsaugojimas.* Šis panaudojimo atvejis suteikia galimybę vartotojui išsaugoti vektorizavimo rezultatus (vektorius) *DXF* formato faile.



49 pav. Sistemos ribos

2.8. Taškinio vaizdo paruošimo vektorizavimui reikalavimai

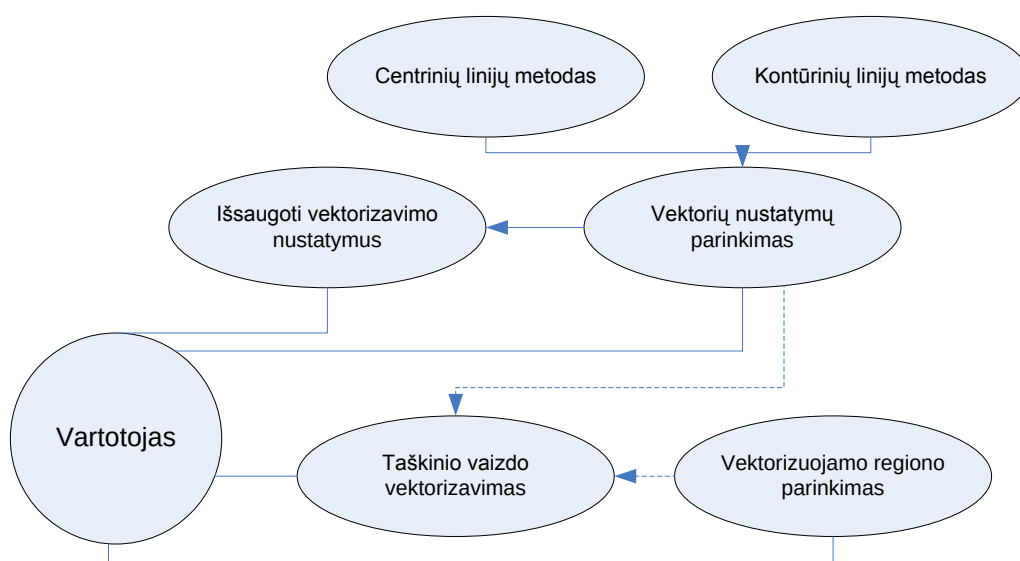
Skaitmeninio vaizdo paruošimo procedūros turi būti vykdomas kaip įmanoma greičiau. Taškinio vaizdo pavertimui į dviejų bitų vienspalvį taškinį vaizdą naudojami įvairūs filtrai. Taškiniam vaizdui paruošti turi būti įgyvendinti tiek linijiniai, tiek nelinijiniai filtrai, taip pat galimybė išsaugoti pakoreguotą taškinį vaizdą. Turi būti galima atidaryti grafinį failą sukurtą fotografavimo metu, tai reikalinga tam kad vartotojas galėtų ekrane matyti vektorizuojamą skaitmeninį vaizdą. Palaikomi populiariausi grafinių failų formatai (BMP, JPEG, TIFF, TGA, PNG). Pasirenkami linijiniai filtrai: vaizdo suliejimas (blur), vaizdo sušvelninimas (soften), vaizdo paryškinimas (sharpen), kraštų detekcija (edge). Pasirenkami nelinijiniai filtrai: vaizdo išplėtimas, kontūrų radimas, triukšmų pašalinimas. Turi būti galimybė koreguoti skaitmeninio vaizdo spalvų gamą. Spalvų gama koreguojama keičiant skaitmeninio vaizdo kontrastą arba taikant slenkstinį filtrą. Išsaugant duomenis į egzistuojantį duomenų failą turi būti pranešam, kad failas bus perrašytas, vartotojas turi žinoti jog senį duomenys bus prarasti.



50 pav. Taškinio vaizdo paruošimas vektorizavimui

2.9. Taškinio vaizdo vektorizavimo kriterijų parinkimo reikalavimai

Vartotojas turi galėti parinkti vektorizavimo nustatymus, pasirinkti vektorizavimo metodą, paleisti vektorizavimą, bei užsiduoti vektorizavimo regioną. Naudojami vektorizavimo tipai: kontūrinių linijų atpažinimas bei centrinių linijų atpažinimas. Jeigu prieš vektorizuodamas vaizdą vartotojas pasirenka vektorizuojamą regioną, vykdant vektorizavimą bus vektorizuojamas tik pasirinktas regionas. Programos vykdymo metu pasirinkti nustatymai bei įvesti duomenys turi būti išimunami. Vartotojui nereikės kiekvieną kartą vykdant programą vesti tų pačių duomenų.

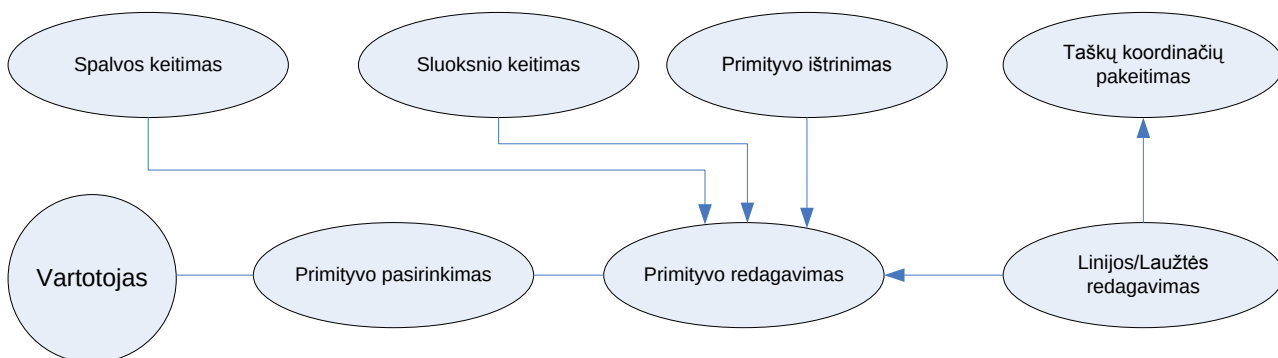


51 pav. Taškinio vaizdo vektorizavimo kriterijų parinkimas

2.10. Vektorių redagavimo reikalavimai

Turi būti galimybė keisti vektorizavimo rezultato (primityvo) sluoksnį, tai reikalinga tam, kad vartotojas galėtų redaguoti vektorizavimo rezultatus. Sluoksnis keičiamas į jau egzistuojantį arba kuriamas naujas. Turi būti galimybė keisti vektorizavimo rezultato (primityvo) spalvą. Spalva pasirenkama iš standartinės spalvų paletės. Turi būti galimybė ištrinti vektorizavimo rezultatą (primityvą). Primityvas ištrinamas be galimybės jį atstatyti. Turi būti galimybė keisti vektorizavimo rezultato (linijos) galų koordinates. Naujos koordinatės įvedamos klaviatūros arba pelės pagalba. Vektorizavimo rezultatų redagavimo metu vartotojui įvedus neteisingas koordinates programa turi

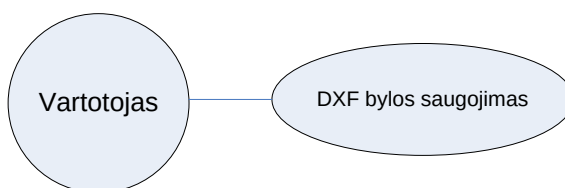
pranešti apie klaidą. Koordinatės traktuojamos neteisingomis tuomet kai jos išeina už skaitmeninio vaizdo ribų. Ekrane turi būti rodoma informacija apie redaguojamą vektorizavimo rezultatą (primityvą). Rodoma primitivo informacija: tipas, sluoksnis, spalva, taškų koordinatės.



52 pav. Vektorių (primityvų) keitimas

2.11. Vektorių saugojimo reikalavimai

Baigiant programos darbą turi būti paklausiama ar užsaugoti pakoreguotus duomenis. Vartotojo turi būti informuojamas, kad gali prarasti neužsaugotus duomenis. Duomenų failo saugojimo/atidarymo metu turi būti įsimenamas paskutinio saugoto/atidaryto failo kelias.



53 pav. Vektorių saugojimas

2.12. Išvados

1. Šiame skyriuje suformuotas pagrindinis daro tikslas, t.y. sukurti vektorizavimo sistemą, skirtą integrinių grandynų rastrinių vaizdų vektorizavimui.
2. Sukurti reikalavimai, kurie turės būti įgyvendinti kuriant sistemą. Pagrindinis reikalavimas yra reikalavimas tikslumui. Dauguma sukurtų sistemų, kurios naudoja standartinius vektorizavimo

algoritmus, neleidžia pasiekti norimų rezultatų, todėl kuriant sistemą bus keičiamas standartinis vektorizavimo algoritmas.

3. Šiame skyriuje apžvelgti pagrindiniai apribojimai. Sistema galės dirbti su pagrindiniais taškinių vaizdų formatais, bei galės būti įdiegta plačiausiai naudojamose operacinėse sistemose.
4. Kuriama sistema bus sudaryta iš dviejų, taškinio ir vektorinio vaizdo, komponentų,
5. Programa realizuos septynis pagrindinius panaudojimo atvejus: taškinio vaizdo įgijimas, taškinio vaizdo išsaugojimas, taškinio vaizdo redagavimas, taškinio vaizdo vektorizavimas, vektorizavimo tipo pasirinkimas, vektorizuojamo regiono pasirinkimas, vektorių redagavimas bei vektorių išsaugojimas.

.

3. ALGORITMŲ TOBULINIMAS. JŲ VEIKIMO PALYGINIMAS.

Sukurta sistema, kurioje įgyvendintas vektorizavimo algoritmas, paremtas standartiniu ploninimu, nedavė reikiamos kokybės rezultatų, todėl buvo nuspręsta patobulinti ploninimo algoritmą. Pasiūlytas vektorizavimo algoritmas, naudojantis Daleunay trianguliacija paremtą ploninimą. Kaip bus matyti eksperimento rezultatuose, dėl filtravimo metu gautų nelygių takelių kraštų taškiniuose vaizduose, vektorizavimo algoritmas, paremtas standartiniu ploninimu, duoda nepakankamai lygią vektorizuotą topologiją. Todėl ir buvo pasiūlytas vektorizavimo algoritmas, naudojantis Delaunay trianguliacija paremtą ploninimą, kuris yra skirtas tiesių linijų išskyrimui. Naudojant pastarąjį ploninimo algoritmą taškinis vaizdas yra padalinamas į aibę nepersidengiančių trikampių. Tuomet tam tikri trikampiai yra sujungiami pašalinant šiukšles. Takelių skeletas gaunamas sujungiant jos sudedamųjų dalių (trikampių) vidurio taškus. Eksperimento rezultatai rodo, jog šis algoritmas yra kur kas greitesnis, nei tipinis ploninimo algoritmas Zhang and Suen [13].

Taip pat įgyvendintas automatinis segmentų vertekso apdorojimas, kuris leidžia ištrinti nereikalingus segmentų vektorius, pagal vartotojo užduotus parametrus. Jeigu vartotojo užduoti parametrai tenkinami, vektoriaus segmento verteksas ištrinamas.

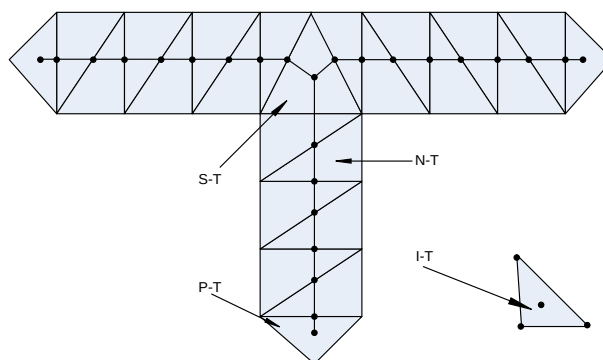
Šiame skyriuje pateiksime standartinio vektorizavimo algoritmo (žiūrėti skyrių 1.6.), paremtą taškinių vaizdų ploninimu, patobulinimą.

3.1. Ploninimas Delaunay trianguliacija

Objekto poligonai gali būti aprašyti tiesios linijos grafu (plokštumoje) $G = (E, V)$, kur kiekviena viršūnė yra kontūro vaizdo taškas, o kiekviena briauna atitinka kontūro segmentą. Grafo G trianguliacija yra lygi $G' = (V, E')$, kur $E \subseteq E'$ taip, kad nei viena briauna, jungianti dvi viršūnes iš aibės V ir kertanti jau egzistuojančią briauną aibėje G' , nebūtų įdėta į G' . Grafo G Delaunay trianguliacija (DT) yra tokia aibės G trianguliacija, kurioje apie kiekvieną trikampį apibrėžti apskritimai savo viduje neturi tokių grafo G viršūnių, kurios yra matomos iš trikampio viršūnės. Dvi grafo viršūnės yra matomos viena iš kitos jeigu jas jungianti tiesi linija nekerta grafo briaunos.

Taškinis vaizdas yra padalinamas į aibę nepersidengiančių trikampių. Tuomet tam tikri trikampiai yra sujungiami pašalinant šiukšles. Figūros skeletas gaunamas sujungiant jos sudedamųjų dalių (trikampių) vidurio taškus. Taškinio vaizdo Delaunay trianguliacija sudaro vidiniai ir išoriniai trikampiai. Vidiniai trikampiai dengia tik objektų vaizdo taškus, o išoriniai – taškinio vaizdo foną (vaizdo taškus lygius 0). Vidinio trikampio briauna vadinama išorine briauna

tuomet kai ji yra kontūro segmentas, kitu atveju ji vadinama – vidine briauna. Kiekviena vidinio trikampio vidinė briauna reiškia, kad šalia jos esantis trikampis taip pat yra vidinis. Yra keturi vidinių trikampių tipai (53 pav.): vidinis trikampis, kuris neturi nei vienos vidinės briaunos yra vadinamas izoliuotu trikampiu (I-T). Šio tipo trikampių skeletas yra jų centroidas. Vidinis trikampis turintis tik vieną vidinę briauną reiškia šakos pabaigą, kadangi jis jungiasi tik su vienu vidiniu trikampiu. Toks trikampis vadinamas pabaigos trikampiu (P-T). Šio trikampo skeletas yra tiesi linija jungianti trikampio centroidą su viduriniu vidinio trikampio vidinės briaunos tašku. Normaliu trikampiu (N-T) vadinamas toks vidinis trikampis, kuris turi dvi vidines briaunas. Toks trikampis jungiasi su dviem vidiniais trikampiais. Tokio tipo trikampių skeletas yra tiesi linija jungianti dviejų trikampių vidinių briaunų vidurio taškus. Vidinis trikampis turintis tris vidines briaunas vadinamas susikirtimo trikampiu (S-T). Šio tipo trikampių skeletas yra trys tiesios linijos jungiančios trikampio centroidą su vienos trikampio vidinės briaunos vidurio tašku.



54 pav. Delaunay trianguliacijos trikampiai

Po trianguliacijos nubrėžimo kiekvienas taškinio vaizdo objektas tampa padalintas į nepersikertančius trikampius. Iš to seka, kad objekto skeletas gali būti gautas iš jį sudarančių vidinių trianguliacijos trikampių skeletų (54 pav.).

Figūros skeletas gautas iš Delaunay trianguliacijos gali turėti šiukšlių, kurios skirstomos į dvi grupes: pakraščių ir sankirtų šiukšles.

Pakraštinė skeleto šaka yra ribojama galinio ir susikirtimo taškų. Tokia šaka yra vadinama pakraščio šiukšle, tuomet kai ji yra nereikšminė objekto šaka. Išsikišimo reikšmingumas yra išsikišimo ilgio l ir pločio w santykis: $\gamma = l/w$. Jeigu $\gamma < \gamma_{th}$ (čia: γ_{th} -pabaigos trikampio aukščio ir pločio santykis) tai pakraštinė skeleto šaka yra pašalinama, o ją sudarantys trikampiai – sujungiami. Rekomenduojama γ_{th} reikšmė yra 0,5.

Objektų susikirtimai, po trianguliacijos, dažniausiai yra padalinami į du ar daugiau susikirtimo trikampių. Klaidinga skeleto šaka yra sukurama tarp dviejų susikirtimo trikampių centroidų. Tokios

klaidingos skeleto šakos yra vadinamos sankirtų šiukšlėmis. Jos yra aptinkamos lyginant susikertančių šakų orientacijas, o pašalinamos sujungiant susijusius trikampus.

3.2. Ploninimo algoritmas

Pasiūlytas algoritmas susideda iš šių žingsnių:

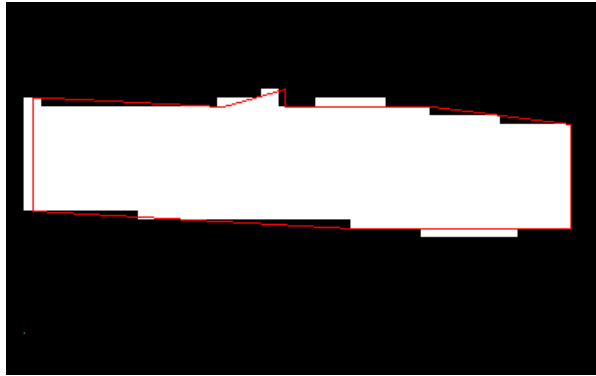
1. Pasirenkamas taškinis vaizdas.
2. Nustatomas objektų pločio mediana – apskaičiuojant objektų pločio medianą w , taškinis vaizdas yra „skaitomas“ horizontaliai ir vertikalčiai. Tuomet sudaroma skaitymo trukmės histograma. w reikšmė bus maksimali histogramos reikšmė..
3. Išskiriami kontūrai bei jiems sudaromi poligonai – Kiekvieno objekto S kontūrai yra sudaromas poligonas, kurio visos briaunos, išskyrus vieną, yra lygios $l = 0,3w$, kur w yra objekto S pločio mediana. Vienos poligono kraštinės ilgis bus trumpesnis..
4. Sudaroma Delaunay trianguliacija kontūrams.
5. Pašalinamos šiukšlės.
6. Išskiriami skeletai.

3.3. Segmentų verteksų šalinimas

Atliekant vektorių išskyrimą, naudojant standartinius vektorizavimo algoritmus, gauname iškraipytas sluoksnių topologijas, kurios neatitinka tikslumo reikalavimo. Taip atsitinka dėl pradinių taškinių vaizdų metalo takelių nelygumų, kurie atsiranda apdorojant (filtruojant) mikroskopu gautas nuotraukas. Pradinis taškinis vaizdas pateiktas 55-ame paveiksle.



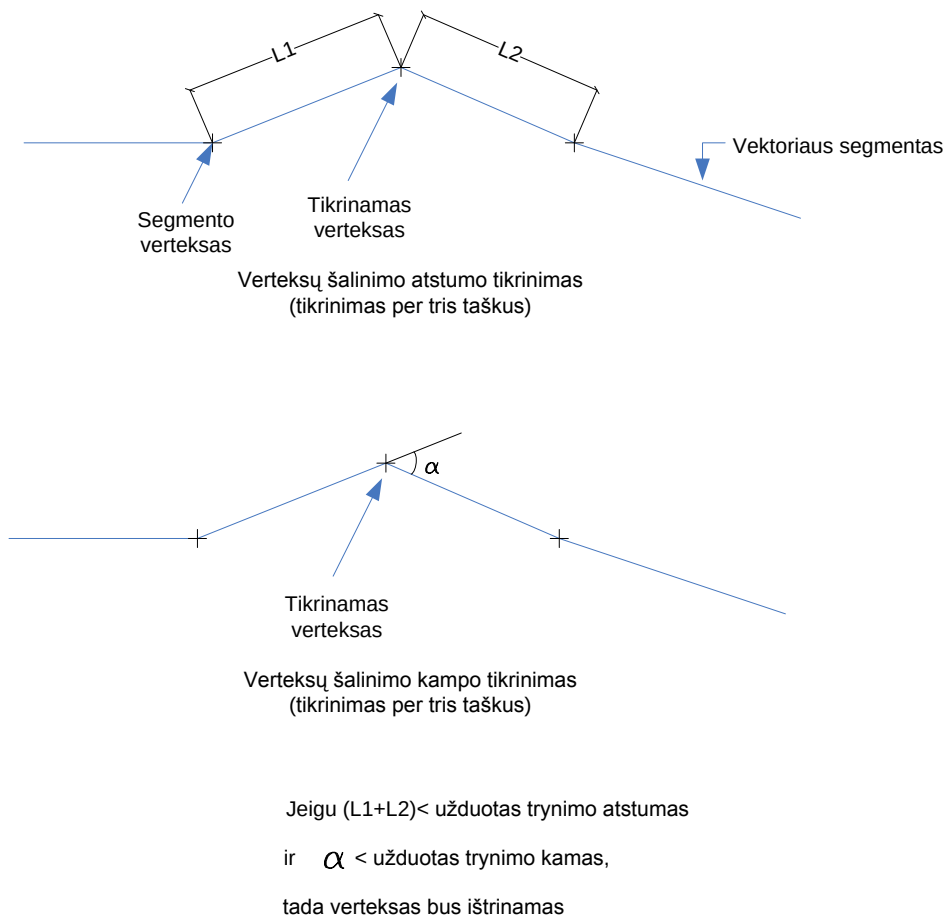
55 pav. Metalų takelio kraštų nelygumai, gaunami filtravimo metu



56 pav. Gaunami vektoriai (raudona linija) naudojant vektorizavimo algoritmą, paremtą standartiniu ploninimu

Iš vektorizavimo rezultatų pateiktą 56-ame paveiksle matome, kad gautas vektorius yra nelygus, t.y. turi nereikalingų verteksų. Perteklinių verteksų trynimui naudosime automatinį segmentų verteksų šalinimo algoritmą.

Verteksų trynimas vykdomas pagal užduotus parametrus: verteksų trynimo atstumą ir verteksų trynimo kampą. Trynimas vykdomas apskaičiuojant vertekso buvimo vietą, įvertinant ryšį su prieš jį ir už jo esančiais verteksais. Jeigu segmentų ilgis tarp šių trijų taškų yra mažesnis nei nurodytas verteksų trynimo atstumas ir jeigu nuokrypio kampas mažesnis nei verteksų trynimo kampas, tai vidurinis segmento taškas bus ištrintas (57 pav.)



57 pav. Gaunami vektoriai (raudona linija) naudojant vektorizavimo algoritmą, paremtą standartiniu ploninimu

3.3. Segmentų verteksų šalinimo algoritmas

Pasiūlytas algoritmas susideda iš šių žingsnių:

1. Automatiškai parenkami apjungti segmentai, gauti iš taškinio vaizdo.
2. Nuskaitomi užduoti verteksų trynimo parametrai, trynimo atstumas ir trynimo kampas.
3. Sudaromas grafas, nustatant verteksų buvimo vietas bei ryšius taro jų, lyginami segmentų ilgiai ir nuokrypių kampai, verteksai tenkinantys užduotas sąlygas pažymimi trynimui.
4. Ištrinami verteksai.

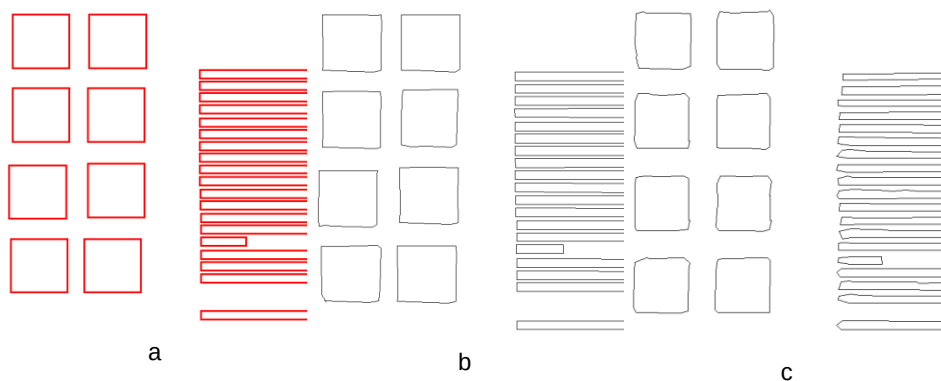
3.4. Patobulintas vektorizavimo algoritmas

Pasiūlytas algoritmas susideda iš šių žingsnių:

1. Pasirenkamas taškinis vaizdas.
2. Atliekamas taškinio vaizdo ploninimas, paremtas Delaunay trianguliacija.
3. Sankirtų žymėjimas.
4. Sankirtų sujungimas.
5. Grandinės sekimas.
6. Grandinės apkirpimas.
7. Segmentacija.
8. Segmentų sujungimas.
9. Apjungtų segmentų verteksų automatinis šalinimas.
10. Išskiriami vektoriai.

3.5. Vektorizavimo rezultatų palyginimas naudojant skirtingus ploninimo algoritmus

54-ame paveiksle pateikti idealūs vektorizavimo rezultatai (a) ir vektorizavimo rezultatų, gautų naudojant skirtingus ploninimo algoritmus, fragmentai. Fragmentas (b) gautas naudojant DT ploninimą, ir geriausiai atitinka idealų vaizdą. (c) fragmentas gautas naudojant tipinį ploninimo algoritmą.



54 pav. Vektorizavimo rezultatai naudojant skirtingus ploninimo algoritmus

4 lentelė. Vektorizavimo rezultatų priklausomybė nuo ploninimo algoritmo

Algoritmas \ Rezultatai	Laužčių vertėsų skaičius, vnt.	Vidutinis laužtės ilgis, nm	Vidutinis vertėsų postūmis nuo idealių rezultatų, nm	Vertėsų postūmių vidutinis kvadratinis nuokrypis, nm
Rankiniu būdu gauti idealūs vektorizavimo rezultatai	124	13,83	-	-
Vektorizavimas naudojant DT ploninimą	288	4,74	0,058	0,850
Vektorizavimas naudojant tipinį ploninimą	674	2,66	0,105	0,879

Iš gautų rezultatų matome, kad mažiausiai objektų vertėsų gaunama naudojant vektorizavimo algoritmą kuris paremtas Delaunay trianguliacija (DT). Vertėsų skaičius rodo, kokio sudėtingumo yra gauti vektoriai, kuo mažiau vertėsų tuo paprasčiau atlikti modifikavimo veiksmus po vektorizavimo proceso. DT paremtas ploninimo algoritmas skirtas tiesių linijų išskyrimas, todėl gaunamas mažesnis vertėsų skaičius, lyginant su kitais tirtais algoritmais . Kaip ir vektorių vertėsų skaičius, taip ir vidutinis laužčių ilgis apibrėžia objektų sudėtingumą bei gautų vektorizavimo rezultatų panašumą į idealius vektorizavimo rezultatus

3.6. Išvados

1. Šiame skyriuje apžvelgėme, vektorizavimo algoritmą, paremtą taškinių vaizdų ploninimu, kuris buvo įgyvendintas kuriamoje sistemoje. Pasirinkome pastarąjį algoritmą, nes jis geriausiai

išskiria tiksliai linijas (žiūrėti 1 lentelę). Atlikus bandymus su algoritmu, kuris naudoja standartinį ploninimo algoritmą, buvo gauti rezultatai, kurie netenkina tikslumo reikalavimo. Todėl buvo pasiūlytas vektorizavimo algoritmas, naudojantis Delaunay trianguliacija paremtą ploninimą, kuris yra skirtas tiesių linijų išskyrimui. Taip pat pasiūlytas papildomas vektorizavimo algoritmo žingsnis, automatinis verteksų trynimas.

2. Pasiūlytas ploninimo algoritmas duoda tikslesnį rezultatą nei tipinis ploninimo algoritmas. Palyginimo rezultatai rodo, kad statistiškai arčiausiai idealaus vektorinio vaizdo buvo vektorizavimo rezultatai gauti naudojant Delaunay trianguliacija paremtą ploninimą.
3. Pasiūlytas segmentų verteksų šalinimo algoritmas, kuris leidžia gauti vektorines topologijas su lygiais metalo sluoksnių kraštais, net jeigu ir po filtravimo gaunami pradiniai taškiniai vaizdai su iškraipytomis briaunomis. Verteksų trynimas vykdomas pagal užduotus parametrus: verteksų trynimo atstumą ir verteksų trynimo kampą. Kadangi, prieš vykdant trynimą įvertinamas ryšys tarp verteksų, tai panaikinama galimybė ištrinti reikalingus topologijos taškus.
4. Pakoreguotas tipinis ploninimu paremtas vektorizavimo algoritmas.

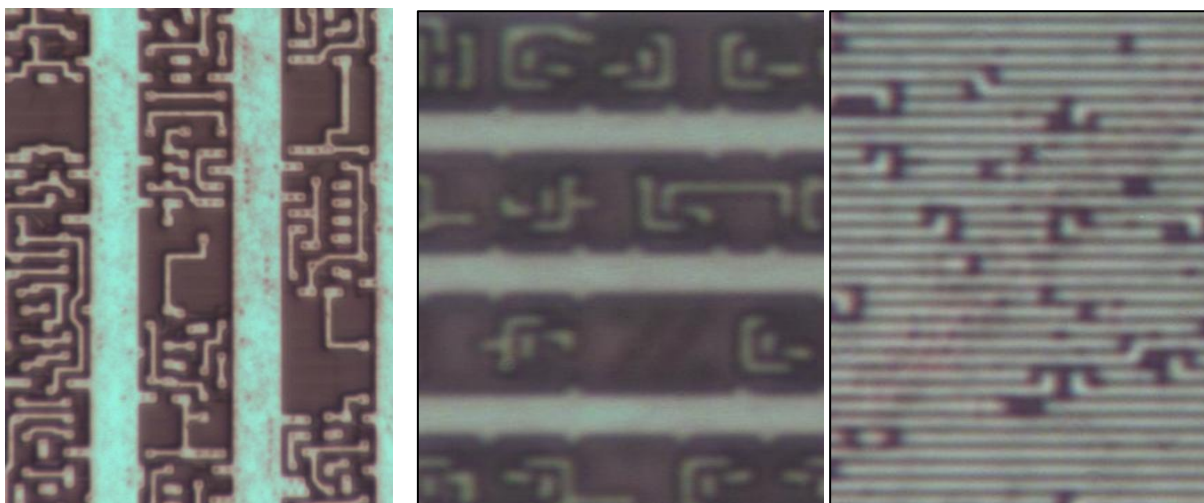
4. VEKTORIZAVIMO METODŲ IR SISTEMŲ TYRIMAS

Algoritmo tobulinimo metu, buvo palyginti vektorizavimo algoritmai paremti standartiniu ploninimu bei Delaunay trianguliacija. Šiame skyriuje pateiksime vektorizavimo sistemoje įgyvendinto vektorizavimo metodo palyginimo su kitais vektorizavimo metodais rezultatus. Aptarsime rezultatus gautus bandant sukurtą sistemą ir jau esančias sistemas. Kaip pradiniai taškiniai vaizdai bus naudojami optiniu ar elektroniniu mikroskopu nufotografuoti lustų metalo sluoksniai. Pagrindinis vertinimo kriterijus yra tikslumas, t.y. automatiškai vektorizuotų vaizdų panašumas į rankiniu vektorizavimo būdu gautus vektorizuotus rezultatus. Įvertinimas atliekamas naudojant standartines geometrines, atkarpos ilgio ir taško atstumo iki tiesės, formules, taip pat statistines vidurkio bei dispersijos formules.

4.1. Pradiniai topologijos taškiniai vaizdai

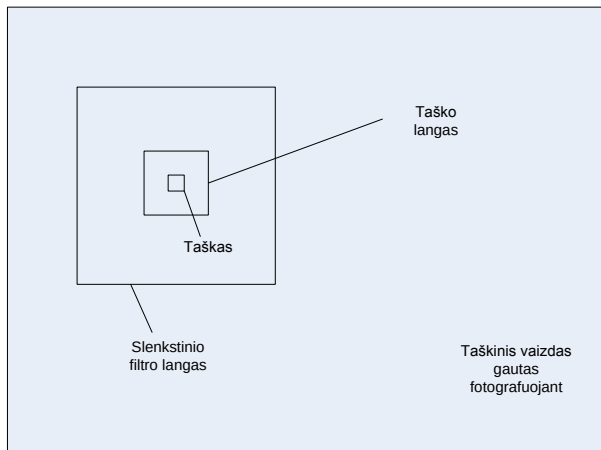
Analizuojant lustus, kurių projektavimo duomenys neprieinami, tenka jų sluoksnių topologiją atkurti tiesiog nagrinėjant pačius lustus. Šiuo atveju yra atliekami tokie technologiniai žingsniai kaip lusto pakuotės atidengimas, lusto cheminis ar plazminis šėdinimas ir poliravimas, siekiant atidengti kiekvieną iš fotolitografijos proceso metu sudarytų sluoksnių. Atitinkamai kiekvienas sluoksnis yra fotografuojamas, naudojant optinės arba elektroninės mikroskopijos įrenginius ir gauta taškinių vaizdų informacija yra toliau apdorojama ir analizuojama siekiant gauti vektorinius lusto topologijos duomenis.

Prieš vektorizavimo etapą, integrinio grandyno (IG) nuotraukos yra apjungiamos į bendrą vektorizuojamo ploto taškinį vaizdą, išlyginamas pasukimo kampas (55 pav.).



55 pav. IG sluoksnio nuotrauka gauta fotografuojant optiniu mikroskopu

Apdorojant nuotraukas pritaikoma eilė įvairių filtrų, kurie išskiria vektorizavimo objektus - IG sluoksnio topologijos struktūras. Vektorizavimo rezultatai priklauso nuo pradinių taškų kokybės bei filtravimo. Filtravimui naudojamas adaptyvus slenkstinis filtras (56 pav.).



56 pav. Slenkstinis filtras

Adaptyvus slenkstinis filtras turi tris prisitaikančius parametrus, w_n – taško lango krašto ilgis, w_m – slenkstinio filtro lango krašto ilgis, s - jautrumo reikšmė. Slenksčio filtro lango reikšmė paskaičiuojama pagal formulę:

$$t = \frac{1}{m} \sum_{j=0}^m I_j; \quad (1)$$

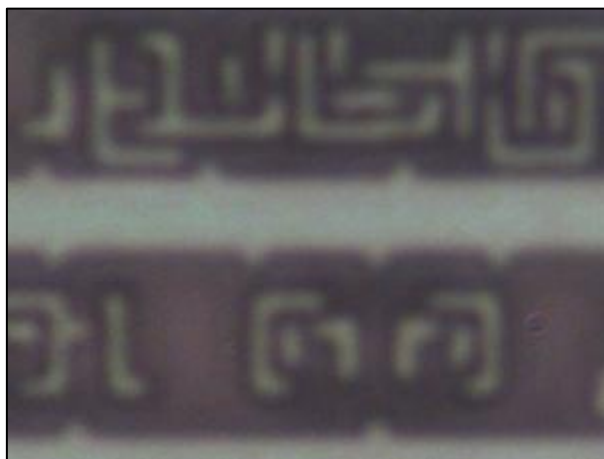
čia $m = w_m^2$.

Taško lango reikšmė apskaičiuojama pagal formulę:

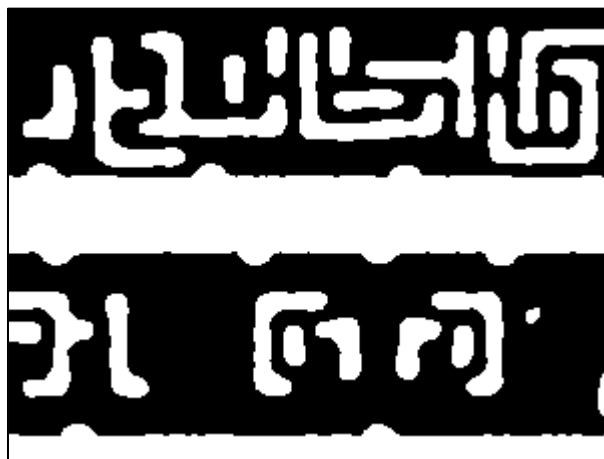
$$p = \frac{1}{n} \sum_{i=0}^n I_i; \quad (2)$$

čia $n = w_n^2$.

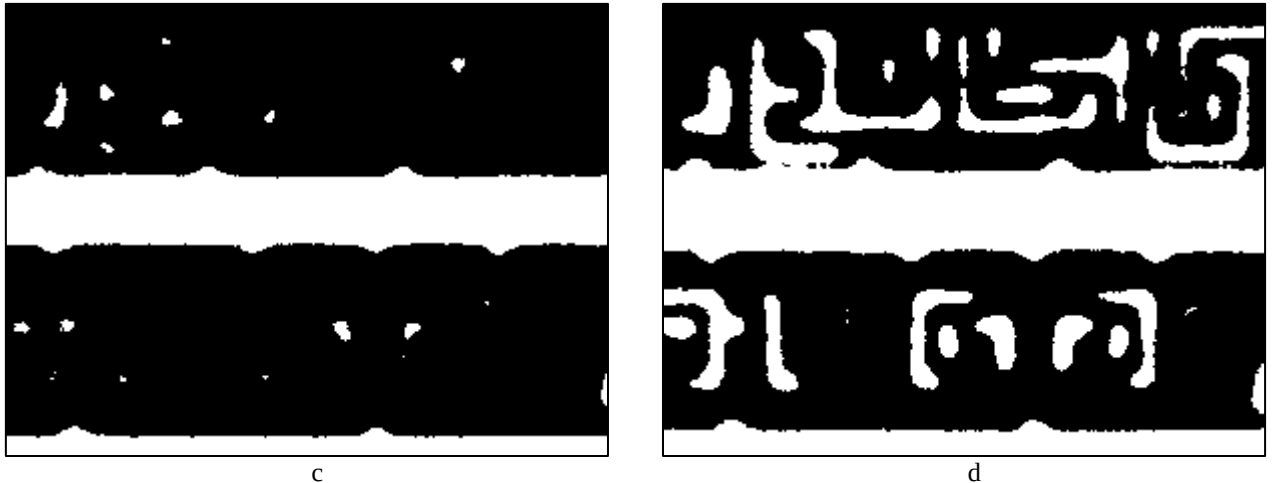
Gauti apdorotų taškinių vaizdų rezultatai pateikti 57 paveiske.



a



b



56 pav. Slenkstinis filtro taikymo rezultatai. Neapdorotas taškinis vaizdas (a), rezultatas gautas naudojant adaptyvinį slenkstinį filtrą (b), rezultatas gautas naudojant iteracinį slenkstinį filtrą (c) ir rezultatas gautas naudojant standartinį slenkstinį filtrą (d)

4.2. Vektorizavimo metodai

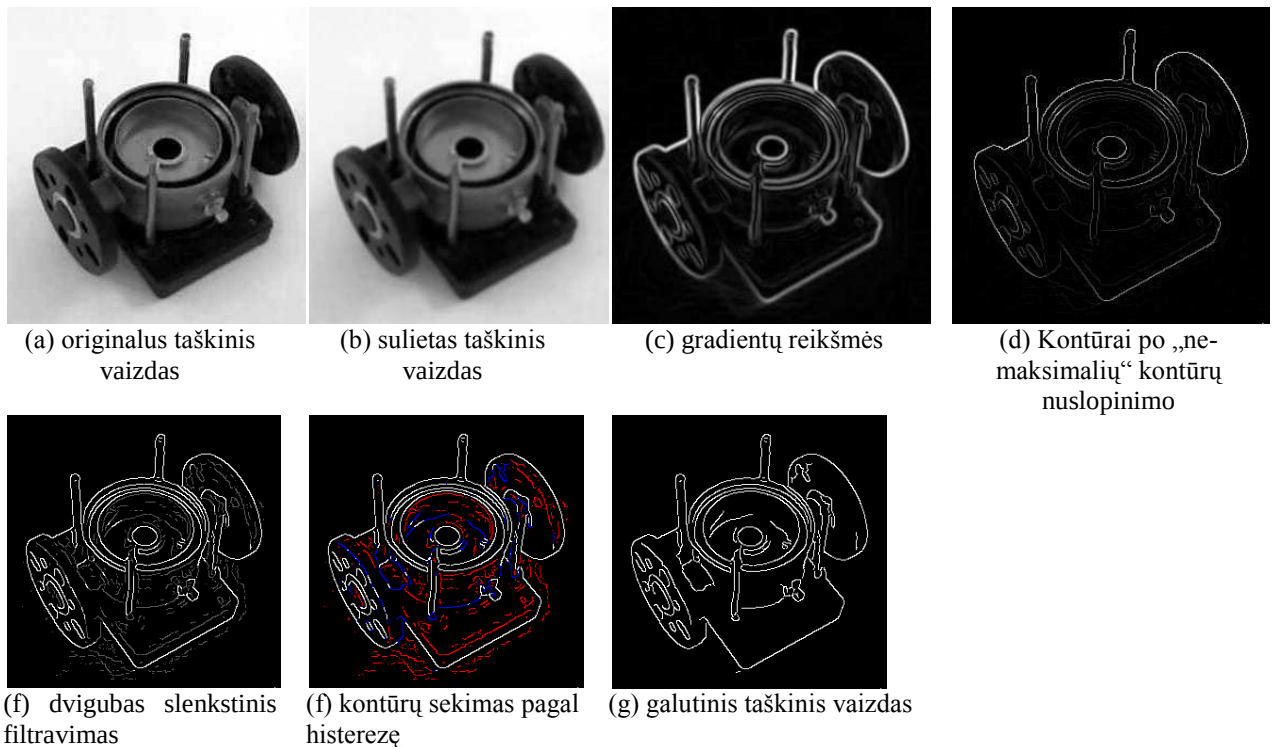
Šiame skyriuje aprašytas vektorizavimo metodų eksperimentinis tyrimas. Pagrindinis vertinimo kriterijus yra tikslumas, t.y. automatiškai vektorizuotų vaizdų panašumas į rankiniu vektorizavimo būdu gautus vektorizuotus rezultatus. Vektorizavimo metodai, pagal linijos išskyrimo būdą (žiūrėti 1.4. skyrių), gali būti klasifikuoti į šešias grupes: paremti ploninimu, Hough transformacija, kontūrų radimu, grafo perėjimu, tinklelio modeliu ir taškųsklaidymu. Taip pat metodai gali būti skirstomi į vieno ir dviejų žingsnių metodus. Pirmo tipo metodai objektus atpažįsta tiesiai iš taškinio vaizdo, kai tuo tarpu antro tipo metodai iš pradžių paverčia taškinį vaizdą į vektorinį formatą, o vėliau, taikant surinkimo-derinimo algoritmus, atpažįsta objektus. Kuriamoje sistemoje įgyvendintas dviejų žingsnių vektorizavimo metodas. Šiame skyriuje tirsime dviejų žingsnių Canny kontūrų radimu paremtą [26] bei vieno žingsnio globalaus linijų tinklo [27] vektorizavimo metodus. Gautus rezultatus palyginsime su vektorizavimo sistemoje įgyvendintu vektorizavimo metodu.

4.2.1. Canny kontūrų išskyrimas

Canny kontūrų išskyrimo vektorizavimo algoritmas yra vienas iš fundamentalių kontūrų išskyrimo algoritmų. Jis remiasi šiais pagrindiniais kriterijais: tikimybė surasti kontūro tašką turi būti didinama, o tikimybė surasti ne kontūro tašką turi būti sumažinta, aptikti kontūrai turi būti kuo artimesni realiems kontūrams taškiniam vaizde. Vektorizavimo algoritmas susideda iš penkių etapų:

1. Lyginimas: Taškinis vaizdas suliejamas, kad išvengtų klaidų.

2. Gradientų suradimas: turi būti pažymimi kontūrai, kur taškinio vaizdo gradientai turi didžiausias reikšmes.
3. „Ne-maksimumo“ slopinimas: tik lokalūs maksimumai gali būti pažymėti kaip tikrieji kontūrai.
4. Dvigubas slenkstinio filtro pritaikymas: potencialūs kontūrai nustatomi slenkstinio filtravimo metu.
5. Kontūrų sekimas pagal histerezę: slopinami visi kontūrai, kurie nesiriboja su tikru (stipriu) kontūru.



57 pav. Visi Canny kontūrų išskyrimo algoritmo etapai

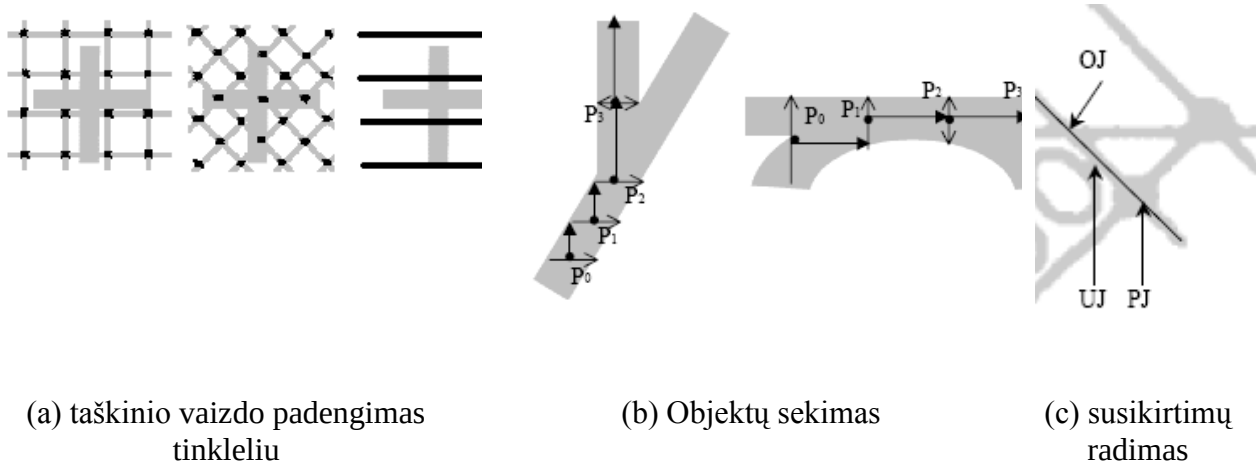
4.2.2. Globalus linijų tinklas

Globalaus linijų tinklo išskyrimo vektorizavimo metodas pirmiausia išskiria pradinius segmentus, kad galėtų paimti objektų pločius ir kryptis iš taškinio vaizdo objektų, vėliau išskiria taškus, vadovaujantis nustatyta kryptimi, taip, kad būtų galima sekti susikirtimus bei to neįtakotų taškinio vaizdo triukšmai. Šis metodas yra vieno žingsnio vektorizavimo metodas, t.y. išskirtiems vektoriams netaikomas joks vėlesnis apdorojimas. Ryšys tarp linijų naudojamas linijų tinklui sudaryti.

Globalaus linijų tinklo metodo algoritmas susideda iš keturių etapų:

1. Taškinio vaizdo padengimas besikeičiančiu tinkleliu.

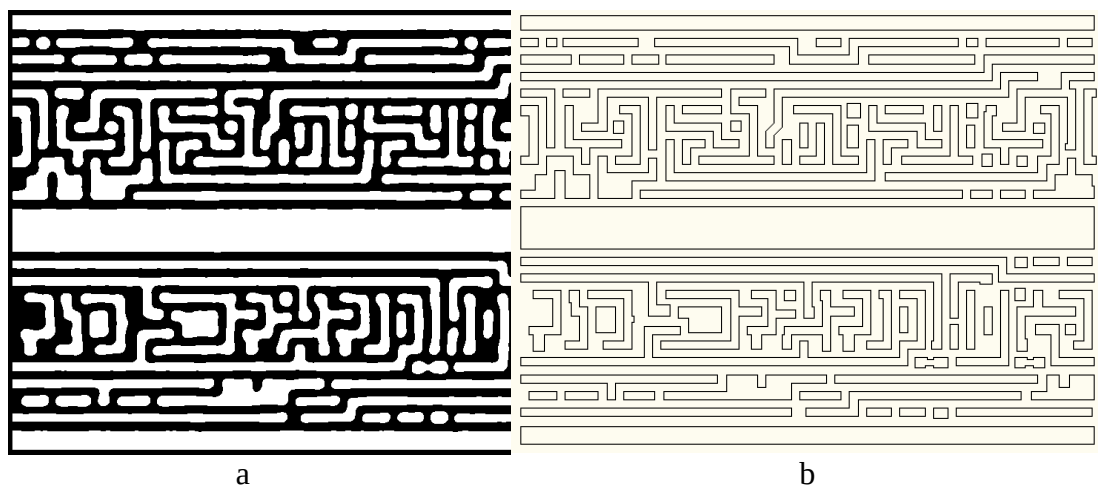
2. Objektų sekimas: sekimas vykdomas horizontaliai, vertikaliai ir įstrižai (zig-zag metodu), šio etapo metu išskiriami pradinės esybės, nustatomi jų storiai ir kryptys.
3. Susikirtimų radimas: Susikirtimai suklasifikuojami į tris tipus, susikirtimų tipai turi hierarchiją pagal kurią vykdomas vektorizavimas.
4. Objektų vektorizavimas

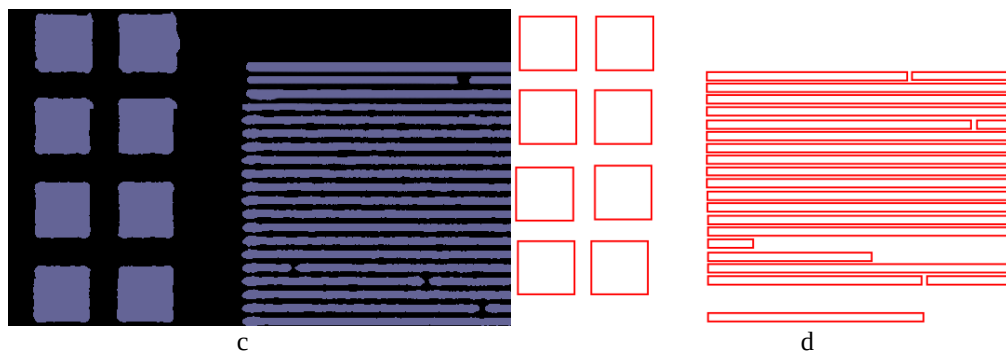


58 pav. Globalaus linijų tinklo algoritmo etapai

4.2.3. Eksperimentiniai rezultatai

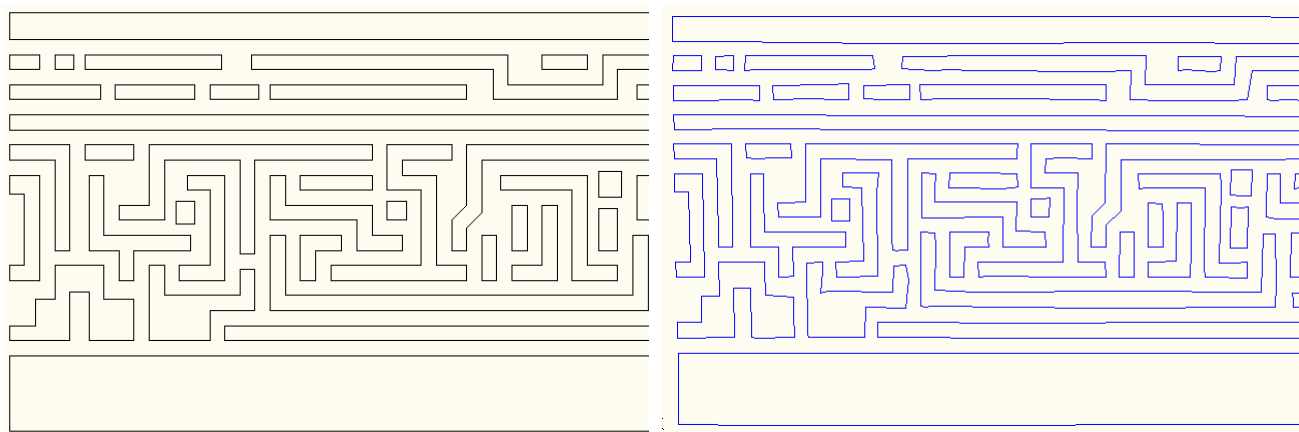
Pradiniai duomenys taškiniai vaizdai, kurie bus naudojami tiriant vektorizavimo metodus pateikti 59 paveiklėse. Jame taip pat pateikiami idealūs vektorizavimo rezultatai, gauti rankiniu būdu, pagal juos bus įvertinami vektorizavimo metodų rezultatai.





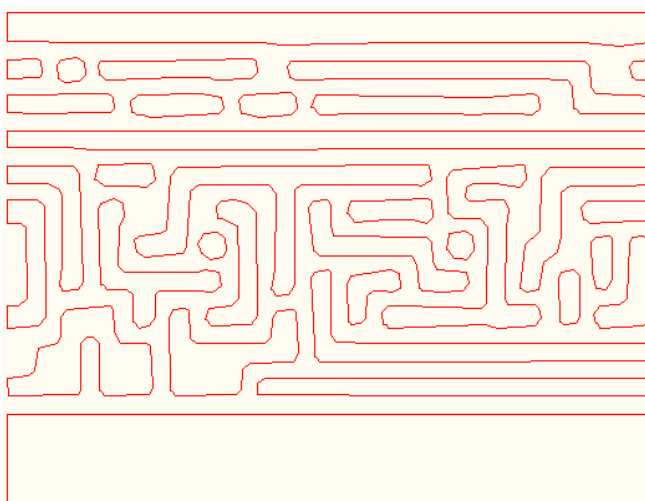
59 pav. Pradiniai taškiniai vaizdai (a) ir (c), etaloniniai vektoriniai vaizdai, gauti rankiniu būdu (b) ir (d) 59-ame paveiksle (b) ir (d) paveikslėlius atitinkamai pavadinkime „eksperimento paveikslėliukas Nr. 1“ ir „eksperimento paveikslėliukas Nr. 2“.

Gauti rezultatai su skirtingais vektorizavimo metodais pateikti 60-ame paveiksle.

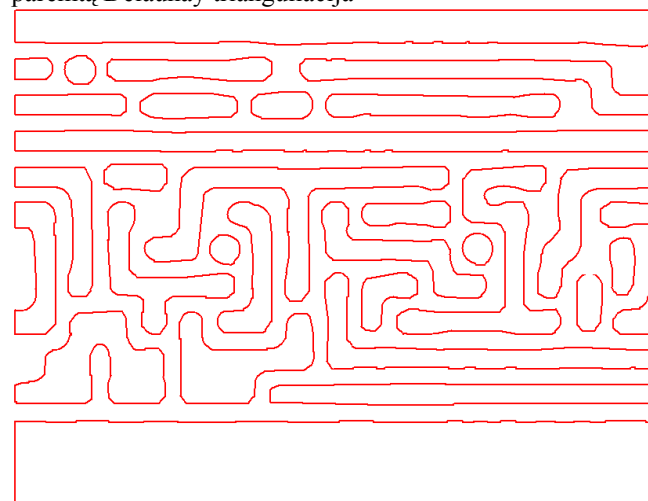


(a) idealus vektorizavimo rezultatas, gautas rankiniu būdu

(b) vektorizavimo rezultatai gaunami naudojant metodą paremtą Delaunay trianguliacija



(c) vektorizavimo rezultatai su Canny kontūrų išskyrimo metodu

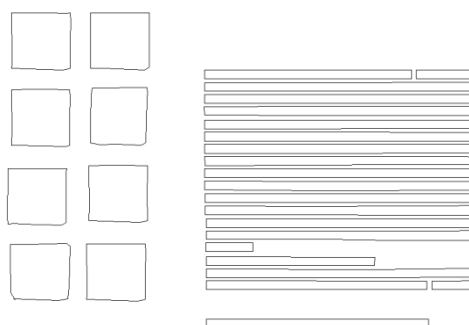


(d) vektorizavimo rezultatai gauti su globalaus linijų tinklo metodu

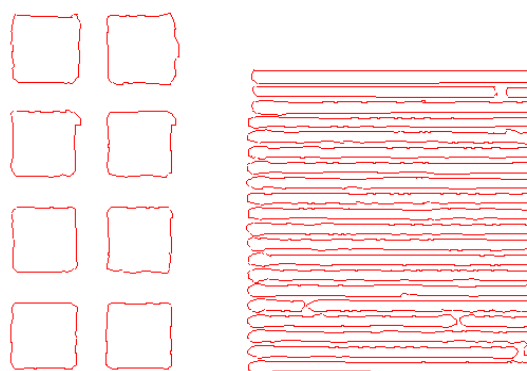
60 pav. Vektoriniai vaizdai gauti su eksperimentiniu taškiniu vaizdu Nr. 1



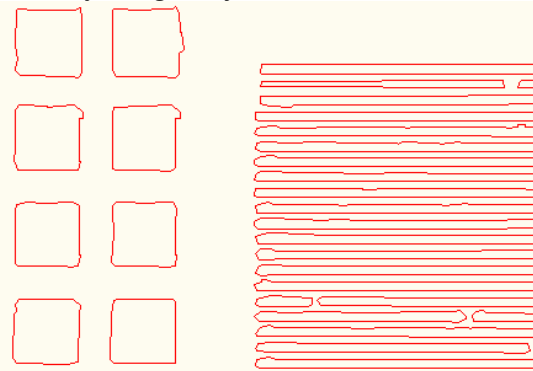
(a) idealus vektorizavimo rezultatas, gautas rankiniu būdu



(b) vektorizavimo rezultatai gaunami naudojant metodą paremtą Delaunay trianguliacija



(c) vektorizavimo rezultatai su Canny kontūrų išskyrimo metodu



(d) vektorizavimo rezultatai gauti su globalaus linijų tinklo metodu

61 pav. Vektoriniai vaizdai gauti su eksperimentiniu taškiniu vaizdu Nr. 2

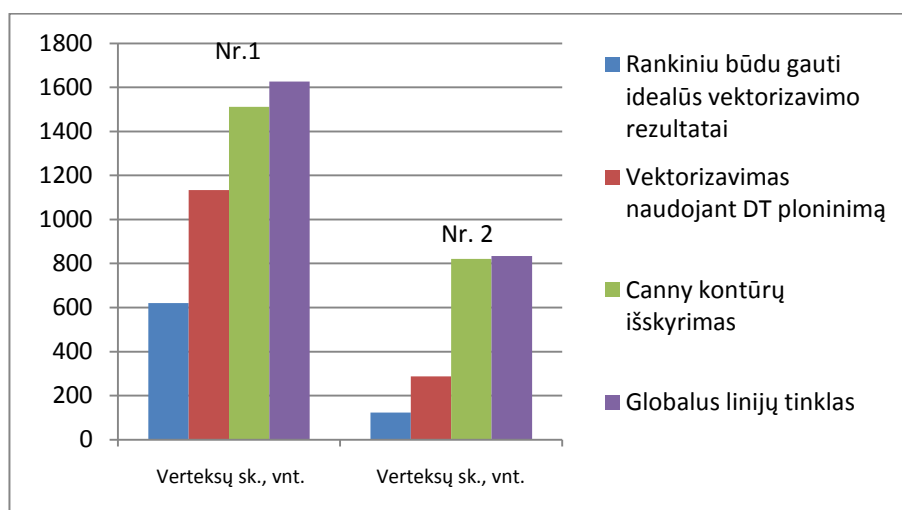
Gauti statistiniai įvertinimai pateikti 5 lentelėje.

5 lentelė. Statistiniai gautų vektorizavimo rezultatų su eksperimentiniu taškiniu vaizdu Nr.1 įvertinimai.

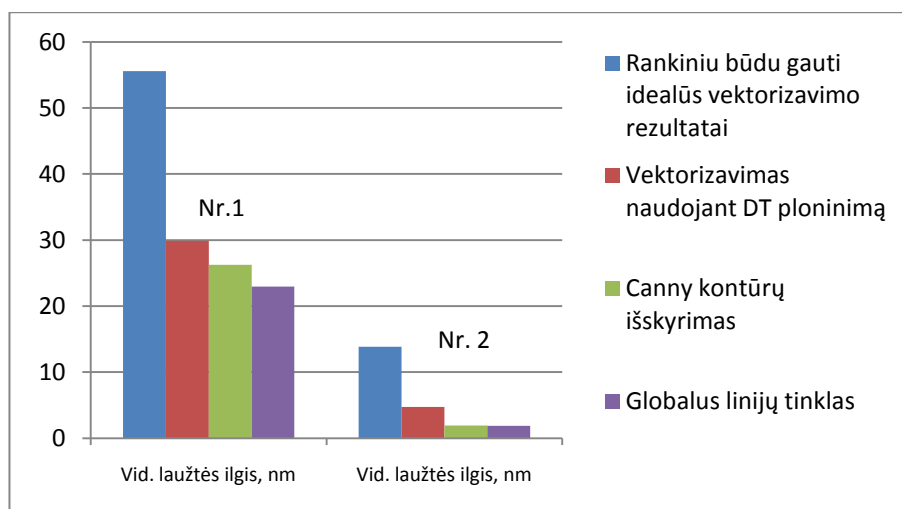
Rezultatai Vektorizavimo metodas	Laužčių verteksų skaičius, vnt.	Vidutinis laužtės ilgis, nm	Vidutinis verteksų postūmis nuo idealių rezultatų, nm	Verteksų postūmių vidutinis kvadratinis nuokrypis, nm
Rankiniu būdu gauti idealūs vektorizavimo rezultatai	620	55,58	-	-
Vektorizavimas naudojant DT ploninimą	1134	29,88	0,219	0,296
Canny kontūrų išskyrimas	1512	26,23	0,421	0,302
Globalus linijų tinklas	1626	22,98	0,556	0,973

6 lentelė. Statistiniai gautų vektorizavimo rezultatų su eksperimentiniu taškiniu vaizdu Nr.2 įvertinimai.

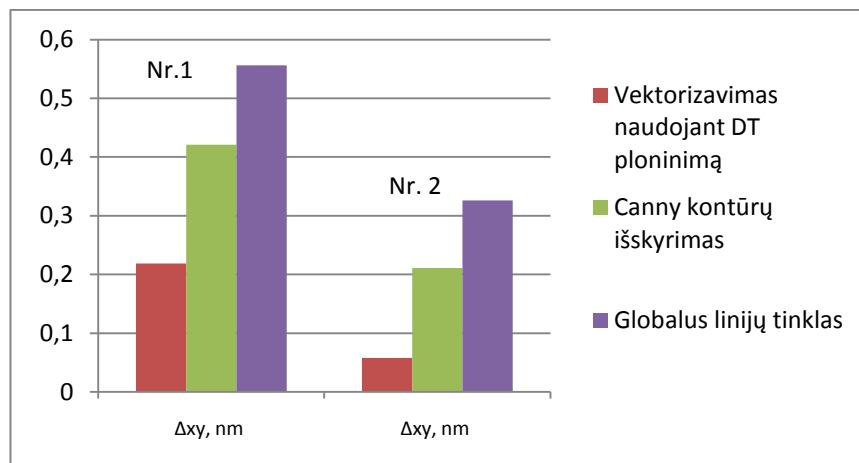
Rezultatai Vektorizavimo metodas	Laužčių verteksų skaičius, vnt.	Vidutinis laužtės ilgis, nm	Vidutinis verteksų postūmis nuo idealių rezultatų, nm	Verteksų postūmių vidutinis kvadratinis nuokrypis, nm
Rankiniu būdu gauti idealūs vektorizavimo rezultatai	124	13,83	-	-
Vektorizavimas naudojant DT ploninimą	288	4,74	0,058	0,241
Canny kontūrų išskyrimas	821	1,91	0,211	0,459
Globalus linijų tinklas	834	1,87	0,531	0,984



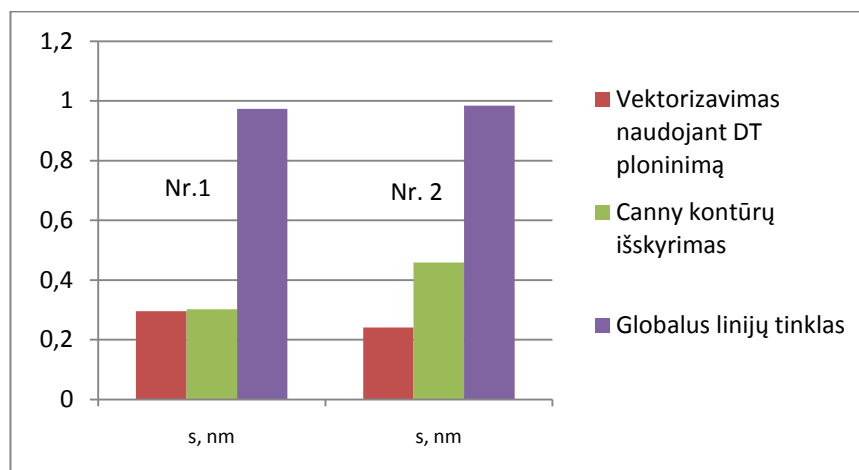
62 pav. Vektorizuoto vaizdo verteksų skaičius, naudojant eksperimentinius taškinis vaizdus Nr.1 ir Nr. 2



63 pav. Vektorizuoto vaizdo vidutinis segmentų ilgis, naudojant eksperimentinius taškinis vaizdus Nr.1 ir Nr. 2

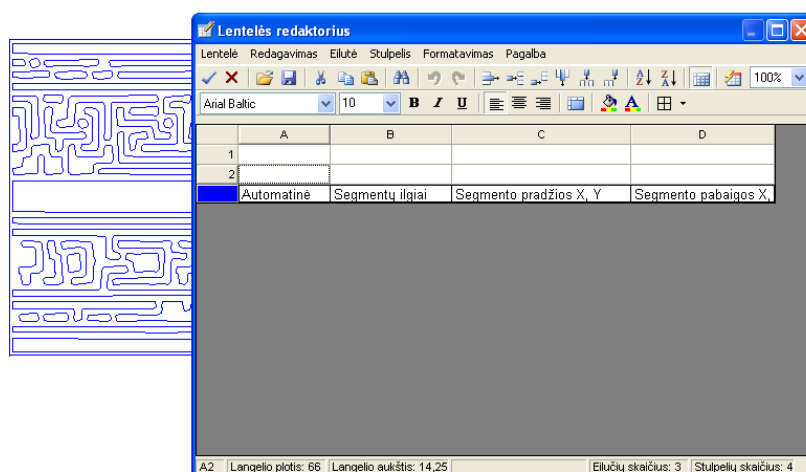


64 pav. Vektorizuoto vaizdo segmentų vertėsų postūmiai nuo idealių vektorizavimo rezultatų, naudojant eksperimentinius taškinius vaizdus Nr.1 ir Nr. 2



65 pav. Vektorizuoto vaizdo segmentų vertėsų postūmių nuo idealių vektorizavimo rezultatų vidutiniai kvadratiniai nuokrypiai, naudojant eksperimentinius taškinius vaizdus Nr.1 ir Nr. 2

Vektorių geometrinei informacijai gauti naudojami specialūs CAD įrankiai (66 pav.)



66 pav. CAD įrankiai naudojami vektorių geometrinei informacijai gauti

Vidutinis vertekų postūmis nuo idealių rezultatų ir vertekų postūmių vidutinis kvadratinis nuokrypis gaunami programinės įrangos „Microsoft Excel 2007“ pagalba, taikant statistines formules:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (2)$$

Iš gautų rezultatų matome, kad geriausi rezultatai gaunami naudojant vektorizavimo metodą kuris paremtas Delaunay trianguliacija (DT). Šio metodo rezultatai, vertinant pagal vidutinės vertekų nuokrypių reikšmes, yra tiksliausi, taip yra dėl patobulinto algoritmo, kuris yra skirtas tiesių linijų išskyrimui. Taip pat šiame metode įgyvendintas automatinis perteklinių vertekų trynimasis. Canny kontūrų išskyrimo metodas duoda tikslesnius rezultatus nei globalaus linijų tinklo metodas. Taip yra todėl, kad Canny metodas yra dviejų žingsnių, t.y. jis naudoja papildomą vektorių segmentų apdorojimo metodą. Globalus linijų tinklo metodas yra vieno žingsnio, todėl vektorizavimo rezultatai labai priklauso nuo pradinių taškinių vaizdų. Kadangi, abiejuose eksperimentiniuose taškiniuose vaizduose metalo takelių kraštai gaunami su nelygumais, todėl is globalaus linijų tinklo metodas duoda netiksliausius rezultatus.

DT paremtas ploninimo algoritmas skirtas tiesių linijų išskyrimui, todėl gaunamas mažesnis vertekų skaičius, lyginant su kitais tirtais algoritmais. Kaip ir vektorių vertekų skaičius, taip ir vidutinis laužčių ilgis apibrėžia objektų sudėtingumą, gautų vektorizavimo rezultatų panašumą į idealius vektorizavimo rezultatus ir vektorinio vaizdo dydį.

4.3. Vektorizavimo sistemos

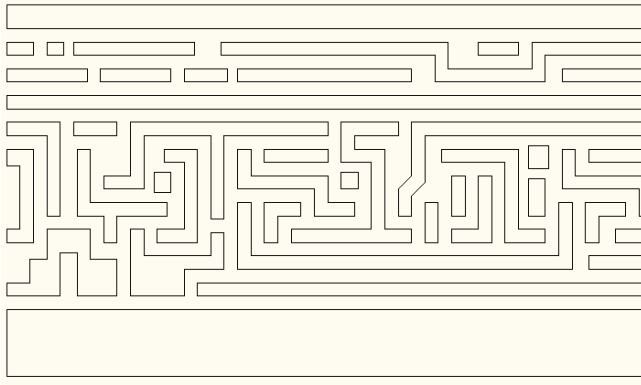
Šiame skyriuje palyginsime rezultatus gautus bandant sukurtą sistemą ir jau esančias sistemas. Kaip ir vektorizavimo metodų atveju pagrindinis vertinimo kriterijus yra tikslumas, t.y. automatiškai vektorizuotų vaizdų panašumas į rankiniu vektorizavimo būdu gautus vektorizuotus rezultatus. Kaip jau buvo aptarta 1.5. skyriuje, egzistuoja daug sukurtų sistemų, eksperimentiniam tyrimui pasirinksiame tas sistemas, kurios turi daugiausia galimybių, t.y. „Vextractor 4.60“ [22] ir „MagicTracer 2.0“ [23].

Abiejų pasirinktų esamų vektorizavimo sistemų algoritmų nežinome. Tiek „Vextractor“, tiek „MagicTracer“ palaiko visus grafinius failus. Abi sistemos turi papildomus įrankius taškinių vaizdų apdorojimui. Kaip ir sukurta IG rastrinių vaizdų vektorizavimo sistema („IGVS“), taip ir pasirinktos programos vektorizavimą atlieka pilnai automatiškai. Automatinės sistemos visas operacijas atlieka

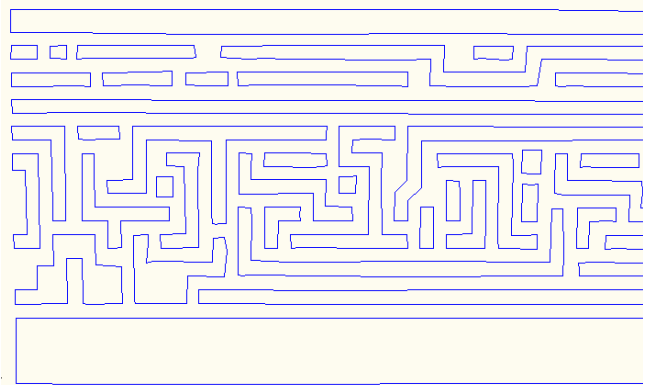
pačios be jokio vartotojo įsikišimo. Pagrindinis šių sistemų privalumas yra veikimo sparta, tačiau rankinis rezultatų redagavimas daugeliu atvejų yra neišvengiamas.

4.3.1. Eksperimentiniai rezultatai

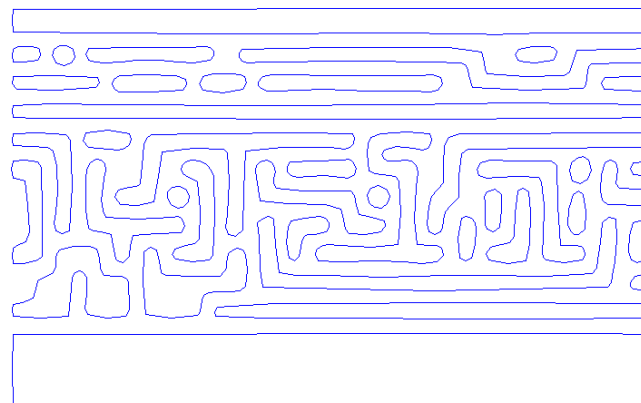
Kaip ir vektorizavimo metodų bandymų metu naudosime tuos pačius pradinis taškinis vaizdus, juos galima pažiūrėti 59- ame paveiksle. Trimis sistemomis gauti vektoriniai vaizdai pateikti 67- ame ir 68-ame paveiksluose.



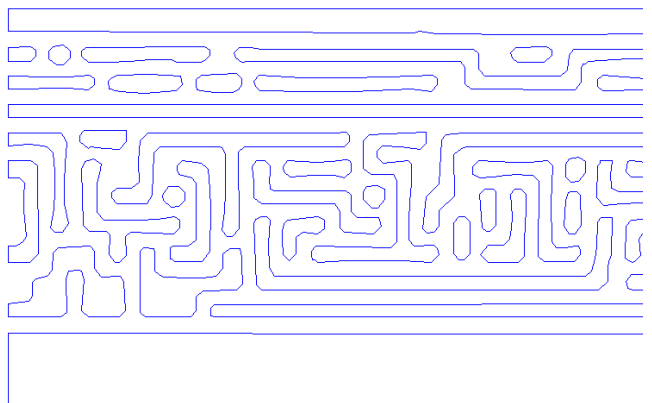
(a) idealus vektorizavimo rezultatas, gautas rankiniu būdu



(b) vektorizavimo rezultatai gaunami su „IGVS“



(c) vektorizavimo rezultatai gaunami su „Vextractor 4.60“

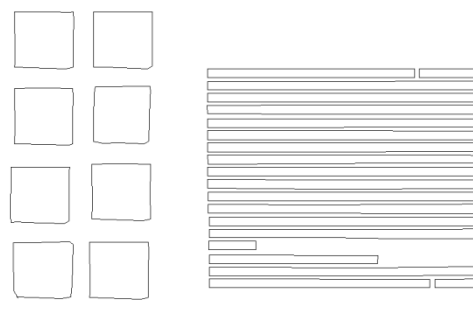


(d) vektorizavimo rezultatai gauti su „MagicTracer 2.0“

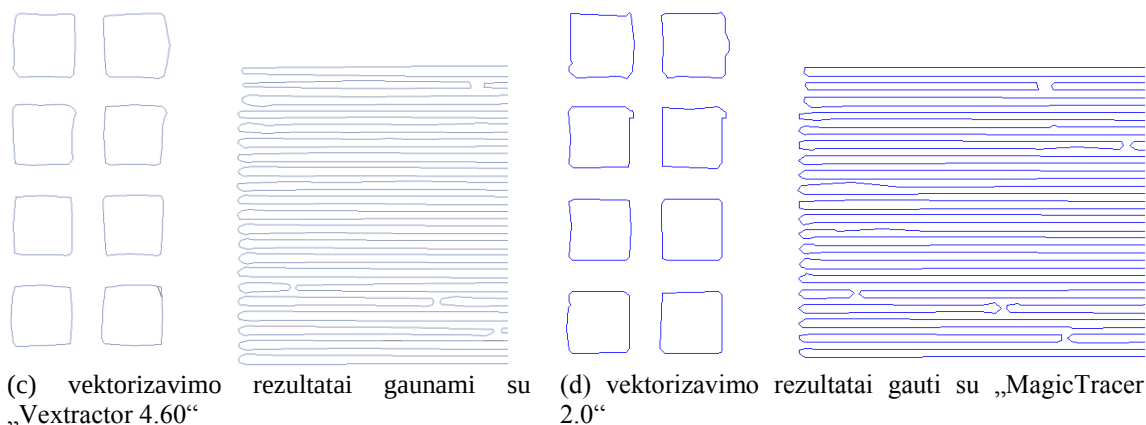
67 pav. Vektoriniai vaizdai gauti su eksperimentiniu taškiniu vaizdu Nr. 1



(a) idealus vektorizavimo rezultatas, gautas rankiniu būdu



(b) vektorizavimo rezultatai gaunami su „IGVS“



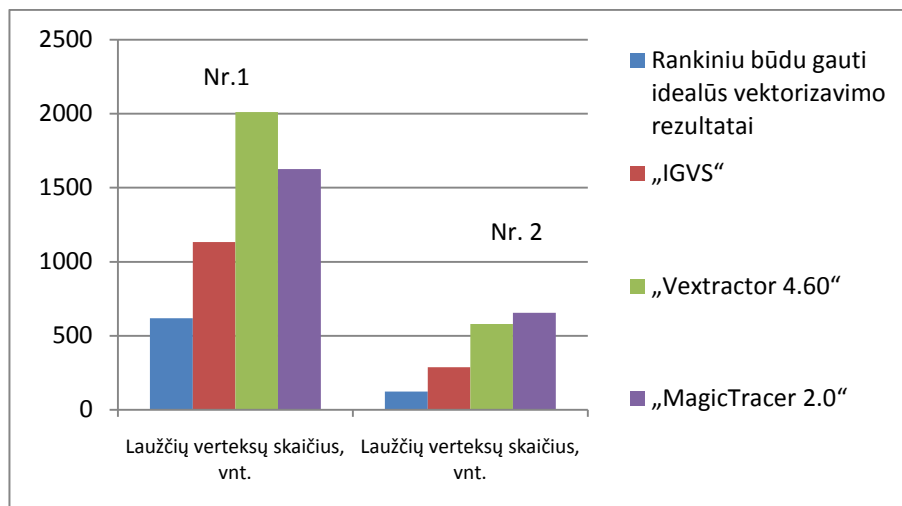
68 pav. Vektoriniai vaizdai gauti su eksperimentiniu taškiniu vaizdu Nr. 2

7 lentelė. Statistiniai gautų vektorizavimo rezultatų su eksperimentiniu taškiniu vaizdu Nr.1 įvertinimai.

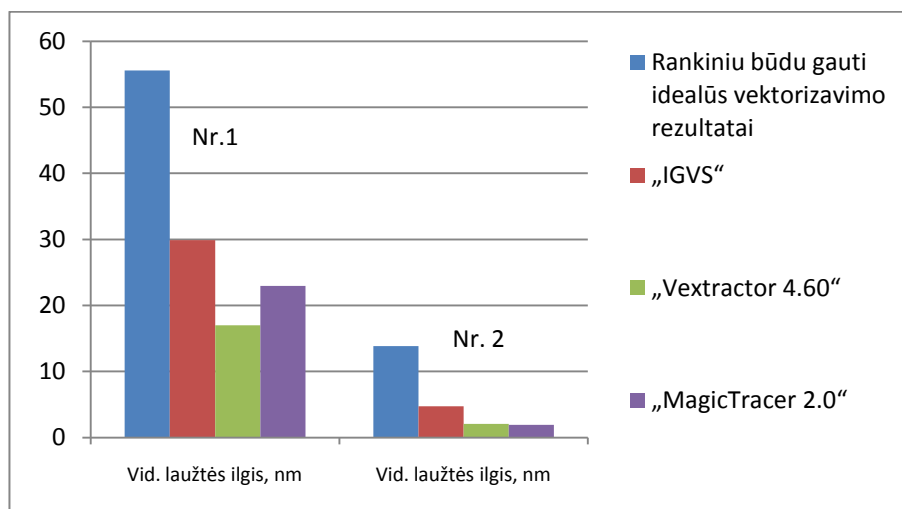
Rezultatai Vektorizavimo sistema	Laužčių verteksų skaičius, vnt.	Vidutinis laužtės ilgis, nm	Vidutinis verteksų postūmis nuo idealių rezultatų, nm	Verteksų postūmių vidutinis kvadratinis nuokrypis, nm
Rankiniu būdu gauti idealūs vektorizavimo rezultatai	620	55,58	-	-
„IGVS“	1134	29,88	0,219	0,296
„Vextractor 4.60“	2011	16,99	0,936	0,822
„MagicTracer 2.0“	1626	22,98	0,337	0,349

8 lentelė. Statistiniai gautų vektorizavimo rezultatų su eksperimentiniu taškiniu vaizdu Nr.2 įvertinimai.

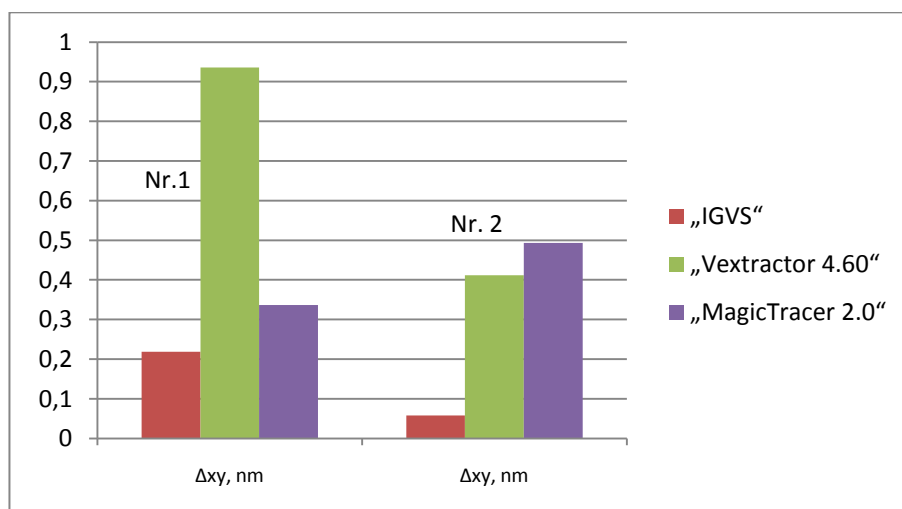
Rezultatai Vektorizavimo sistema	Laužčių verteksų skaičius, vnt.	Vidutinis laužtės ilgis, nm	Vidutinis verteksų postūmis nuo idealių rezultatų, nm	Verteksų postūmių vidutinis kvadratinis nuokrypis, nm
Rankiniu būdu gauti idealūs vektorizavimo rezultatai	124	13,83	-	-
„IGVS“	288	4,74	0,058	0,241
„Vextractor 4.60“	581	2,08	0,412	0,683
„MagicTracer 2.0“	655	1,93	0,493	0,904



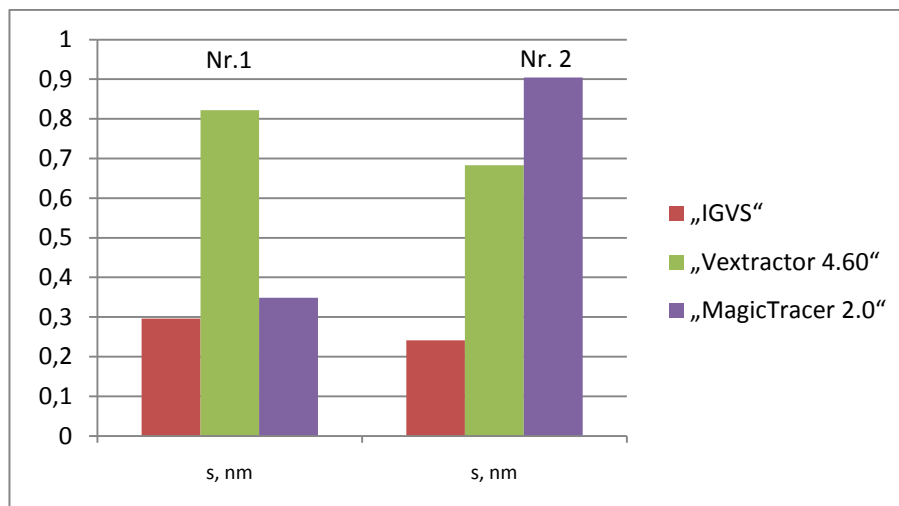
69 pav. Vektorizuoto vaizdo verteksu skaičius, naudojant eksperimentinius taškinius vaizdus Nr.1 ir Nr. 2



70 pav. Vektorizuoto vaizdo vidutinis segmentų ilgis, naudojant eksperimentinius taškinius vaizdus Nr.1 ir Nr. 2



71 pav. Vektorizuoto vaizdo segmentų verteksu postūmiai nuo idealių vektorizavimo rezultatų, naudojant eksperimentinius taškinius vaizdus Nr.1 ir Nr. 2



72 pav. Vektorizuoto vaizdo segmentų verteksų postūmių nuo idealių vektorizavimo rezultatų vidutiniai kvadratiniai nuokrypiai, naudojant eksperimentinius taškinis vaizdus Nr.1 ir Nr. 2

Atlikome vektorizavimo sistemų tyrimus, iš gautų rezultatų matome, kad tiksliausi vektorizavimo rezultatai gaunami naudojant „IGVS“ sistemą. Su šia sistema gaunamas mažiausias vidutinis verteksų postūmis nuo idealių rankiniu būdu gautų vektorizavimo rezultatų.

Visos tirtos sistemos yra automatinės, todėl gali atsirasti poreikis rankiniam duomenų redagavimui. Išskirti vektoriai su „IGVS“ turi mažiausiai verteksų ir didžiausius vidutinius segmentų ilgius, todėl šie rezultatai turi mažesnę informacijos kiekį, t.y. jais paprasčiau operuoti.

Iš 7-oje lentelėje pateiktų tyrimo rezultatų matome, kad sistema „MagicTracer 2.0“ išskiria tikslesnius ir kokybiškesnius duomenis dirbdama su netiesiais metalo takeliais. Šios sistemos rezultatai suprastėja naudojant taškinis vaizdus su tiesiais metalo takeliais.

4.4. Išvados

1. Šiame skyriuje apžvelgėme pradinių taškinį vaizdų filtravimą. Šis integrinių grandynų atvirkštinio projektavimo etapas yra labai svarbus, kadangi nuo pradinių taškinį vaizdų priklauso tolesnis vektorizavimo procesas. Rastrų apdorojimui naudojami patobulinti filtravimo metodai, kurie leidžia panaikinti nepageidaujamus triukšmus.
2. Šioje darbo dalyje buvo atliktas vektorizavimo metodų tyrimas. Pagrindinis vertinimo kriterijus buvo tikslumas, t.y. automatiškai vektorizuotų vaizdų panašumas į rankiniu vektorizavimo būdu gautus vektorizuotus rezultatus. Eksperimentiniam tyrimui buvo pasirinkti vieno ir dviejų žingsnių vektorizavimo metodai. Atlikus bandymus, geriausius rezultatus gavome naudojant vektorizavimo metodą, kuris paremtas Delaunay trianguliacija (DT). Šio metodo rezultatai, vertinant pagal vidutinės verteksų nuokrypių reikšmes, yra tiksliausi, taip yra dėl patobulinto

algoritmo, kuris yra skirtas tiesių linijų išskyrimui. Taip pat šiame metode įgyvendintas automatinis perteklinių vertekų trynimas. Canny kontūrų išskyrimo metodas duoda tikslesnius rezultatus nei globalaus linijų tinklo metodas. Taip yra todėl, kad Canny metodas yra dviejų žingsnių.

3. Atliktas vektorizavimo sistemų tyrimas parodė, kad tiksliausi vektorizavimo rezultatai gaunami naudojant integrinių grandynų vektorizavimo „IGVS“ sistemą. Išskirti vektoriai su „IGVS“ turi mažiausiai vertekų ir didžiausius vidutinius segmentų ilgius, todėl šie rezultatai turi mažesnę informacijos kiekį, t.y. jais paprasčiau operuoti.

5. APIBENDRININIMAS

1. Darbe apžvelgėme atvirkštinio projektavimo proceso etapus. Detaliai išnagrinėjome vektorizavimo etapą, kuriame integrinio grandyno rastrinį vaizdą, gautą fotografuojant mikroskopu, turime paversti vektoriniu vaizdu. Pastarieji vaizdai naudojami, nes jie užima mažiau vietos, jų kokybė nepriklauso nuo vaizdo didinimo ar mažinimo, galima taikyti geometrines formules, taip pat vektorinius duomenis galima naudoti įvairiose CAD sistemose. Aptarėme ir palyginome pagrindinius linijų išskyrimo metodus. Aukščiausia linijos kokybė gaunama naudojant vektorizavimo algoritmus, kurie paremti linijų ploninimu. Todėl šį vektorizavimo metodą panaudojome sukurtoje vektorizavimo sistemoje. Detaliai išnagrinėjome vektorizavimo procesą. Apžvelgėme vektorizavimo sistemų padėtį Lietuvoje bei pasaulyje. Visos sistemos atpažįsta didelį objektų tipų kiekį, tačiau nė viena nepritaikyta konkrečiam uždaviniui spręsti, todėl ir buvo sukurta specializuota vektorizavimo sistema, kuri skirta konkretiems atvejams spręsti, gaunant maksimalų tikslumą bei įvertinant įvairias problemines situacijas.
2. Darbo metu buvo suformuluoti ir įgyvendinti tikslai, reikalavimai vektorizavimo sistemai. Sistema realizuoja šiuos pagrindinius panaudojimo atvejus: taškinio vaizdo įgijimas, taškinio vaizdo išsaugojimas, redagavimas, vektorizavimas, vektorizavimo tipo pasirinkimas, vektorizuojamo regiono pasirinkimas, vektorių redagavimas bei vektorių išsaugojimas. Vektorizavimo sistema sudaryta iš dviejų pagrindinių taškinio ir vektorinio vaizdo komponentų.
3. Darbe pasiūlytas ploninimo algoritmas, kuris duoda tikslesnį rezultatą nei tipinis ploninimo algoritmas. Palyginimo rezultatai rodo, kad statistiškai arčiausiai idealaus vektorinio vaizdo buvo vektorizavimo rezultatai, gauti naudojant Delaunay trianguliacija paremtą ploninimą. Taip pat įgyvendintas segmentų vertekų šalinimo algoritmas, kuris leidžia gauti vektorines topologijas su lygiais metalo sluoksnių kraštais, net jeigu ir po filtravimo gaunami pradiniai taškiniai vaizdai su iškraipytomis briaunomis. Patobulintas tipinis ploninimu paremtas vektorizavimo algoritmas.
4. Vektorizavimo metodų ir sistemų tyrimo metu statistiškai įvertinti vektorizavimo rezultatai, gaunami naudojant pasiūlytą vektorizavimo metodą ir sukurta vektorizavimo sistemą. Rezultatai buvo palyginti su egzistuojančių vektorizavimo metodų ir sistemų rezultatais. Atlikus bandymus, geriausius rezultatus gavome naudojant vektorizavimo metodą, kuris paremtas Delaunay trianguliacija (DT). Šio metodo rezultatai, vertinant pagal vidutines vertekų nuokrypių reikšmes, yra tiksliausi, taip yra dėl patobulinto algoritmo, kuris yra skirtas tiesių linijų išskyrimui. Taip pat šiame metode įgyvendintas automatinis perteklinių vertekų trynimas. Atliktas vektorizavimo sistemų tyrimas parodė, kad tiksliausi vektorizavimo rezultatai

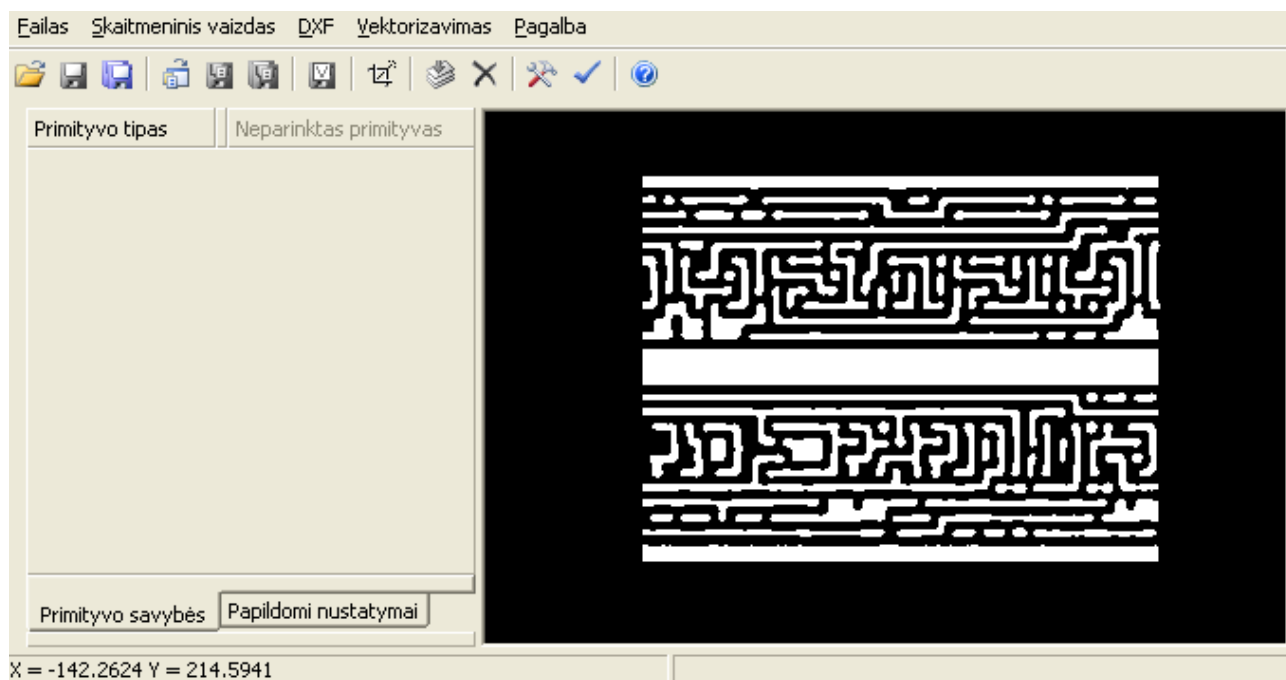
gaunami naudojant integrinių grandynų vektorizavimo „IGVS“ sistemą. Išskirti vektoriai su „IGVS“ turi mažiausiai verteksų ir didžiausius vidutinius segmentų ilgius, todėl šie rezultatai turi mažesnę informacijos kiekį, t.y. jais paprasčiau operuoti.

LITERATŪRA

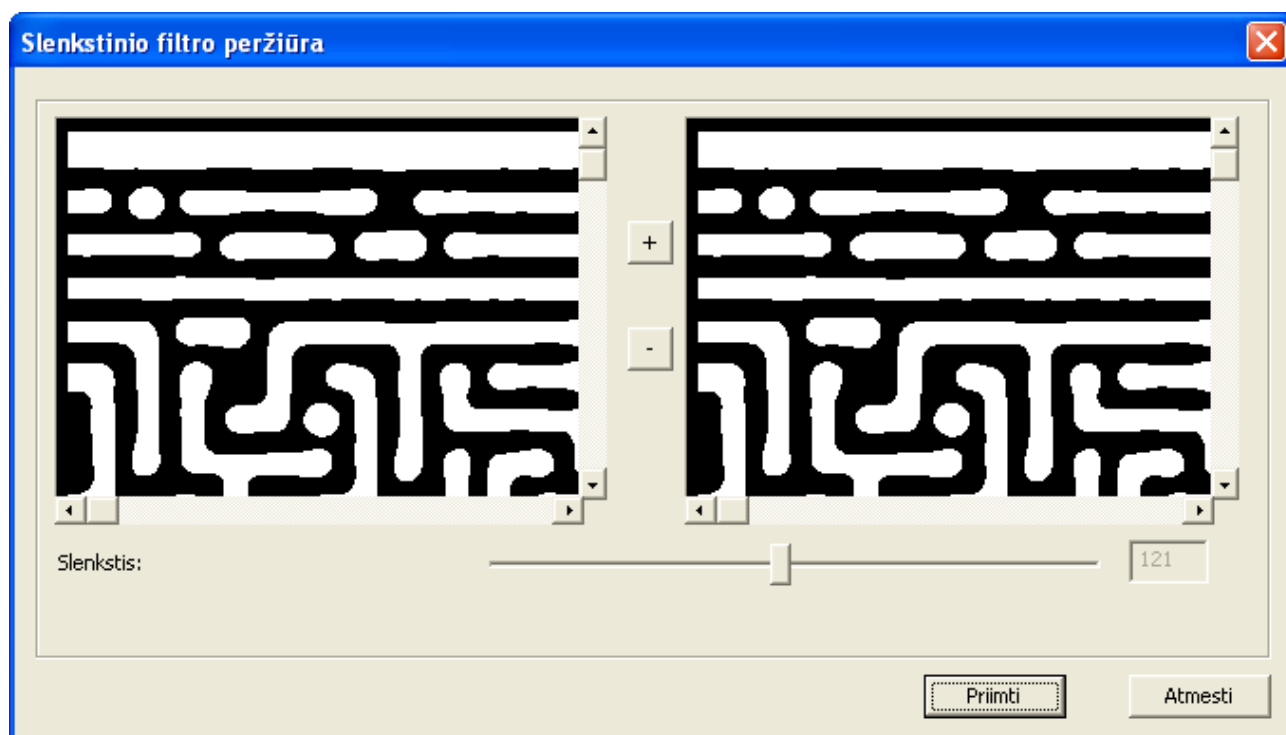
1. Y. Wu, "Raster, Vector, and Automated Raster-to-Vector Conversion", in "Moving Theory into Practice: Digital Imaging for Libraries and Archives", book eds. by Anne R. Kinney and Oya Y. Rieger, 2000, RLG, Cornell Univ. Library
2. R. C. Gonzalez, P. Wintz, Digital Image Processing“ (2nd edition), Addison Wesley, 1987, p. 105 – 123.
3. F. van Dam, F. Hughes, Computer Graphics Principles and Practice“ (2nd edition), Addison Wesley, 1992, p. 567 – 580.
4. L.R. Poos and Y. Wu, "Digitizing History: GIS and Historical Research", GIS World, July 1995, p. 48-51
5. J.R. Parker, "Algorithms for Image Processing and Computer Vision", 1997, John Wiley & Sons.
6. Dori D. Orthogonal Zig-Zag: an Algorithm for Vectorizing Engineering Drawings Compared with Hough Transform. Advances in Engineering Software 1997; 28(1):11-24.
7. Tamura H. A Comparison of line thinning algorithms from digital geometry viewpoint. In Proceedings of the 4th International Conference on Pattern Recognition, Kyoto, Japan, 1978, p. 715-719.
8. Nalwa VS. A Guided Tour of Computer Vision. Addison-Wesley, New York, 1993.
9. Chai I, and Dori D. Orthogonal Zig-Zag: An Efficient Method for Extracting Lines from Engineering Drawings. In: Arcelli C, Cordella LP, Sanniti di Baja G (eds). Visual Form. Plenum Press, New York London, 1992, p. 127-136.
10. Monagan G and Roosli M. Appropriate Base Representation Using a Run Graph. In Proceedings of the 2nd International Conference on Document Analysis and Recognition, Tsukuba, Japan, 1993, p. 623-626.
11. T. Zhang, and C. Suen, A fast parallel algorithm for thinning Digital patterns, Communications of the ACM, 1984, Nr. 27, p. 236 – 239.
12. P. L. Rosin, G. A. W. West, Nonparametric segmentation of curves into various Representations, Pattern Analysis and Machine Intelligence, 1995, Nr. 17, p.1140 – 1153.
13. J. J. Zou, H. Yan, Vectorization of Cartoon Drawings, ACM International Conference Proceeding Series, 2002, Nr. 9, p. 77 – 78.
14. A. Chhabra, Graphic Symbol Recognition: An Overview, 1 Lecture Notes In Computer Science, 1998, Nr. 1389, p . 68 – 79.

15. K. Tombre, and A.K. Chhabra, Graphics Recognition - Algorithms and Systems, Lecture Notes in Computer Science, 1998, Nr. 1389, p. 420 – 422.
16. J. Chiang, S. Tue, Y. Leu, A new algorithm for line image vectorization, PR, 1998, Nr. 39, p. 1541-1549
17. J. Song, F. Su, C. Tai, J. Chen, S. Cai, Line Net Global Vectorization: an Algorithm and Its Performance Evaluation, CVPR`00, 2000, Nr. 1, p. 1383 – 1390.
18. V. A. Kovalevsky, New definition and fast recognition of digital straight segments and arcs, Pattern Recognition Proceedings, 1990. Nr. 2, p. 31 - 34.
19. E. Bodansky, G. Alexander, P. Morakot, Post-processing of lines obtained by raster-to-vector conversion, Vision (machine Vision Association of SME), 2002, Nr. 18, p. 45 - 52.
20. UAB InfoEra, vektorizavimo sistemų importuotoja [interaktyvus] [žiūrėta 2009-01-21]. Prieiga per Internetą:
<http://www.infoera.lt/index.php/pageid/705>
21. UAB „HNIT-BALTIC“, vektorizavimo sistemų importuotoja [interaktyvus] [žiūrėta 2009-01-21]. Prieiga per Internetą:
<http://www.hnit-baltic.lt/DesktopDefault.aspx?tabID=3595&lang=lt-LT>
22. UAB „AGA-CAD“, vektorizavimo sistemų importuotoja [interaktyvus] [žiūrėta 2009-01-21]. Prieiga per Internetą:
http://www.aga-cad.lt/produktai/geoerdviniams_sprendimams/autocad_raster_design/237
23. Vektorizavimos sistema „Vextractor“ [interaktyvus] [žiūrėta 2009-01-28]. Prieiga per Internetą:
<http://www.vextrasoft.com/>
24. Vektorizavimos sistema „MagicTracer“ [interaktyvus] [žiūrėta 2009-01-28]. Prieiga per Internetą:
<http://www.magictracer.com/>
25. L. Gorman, Basic Techniques and Symbol-Level Recognition An Overview, Graphic Recognition. Methods and Application, 1996, Nr. 1, p. 1 – 12.
26. John Canny, A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6), 1986, p. 679–698
27. Jiqiang Song, Feng Su, Jibing Chen, Chiewlan Tai, Shijie Cai, Line Net Global Vectorization: an Algorithm and Its Performance Evaluation, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Cvpr, Nr. 1, p.1383-1395

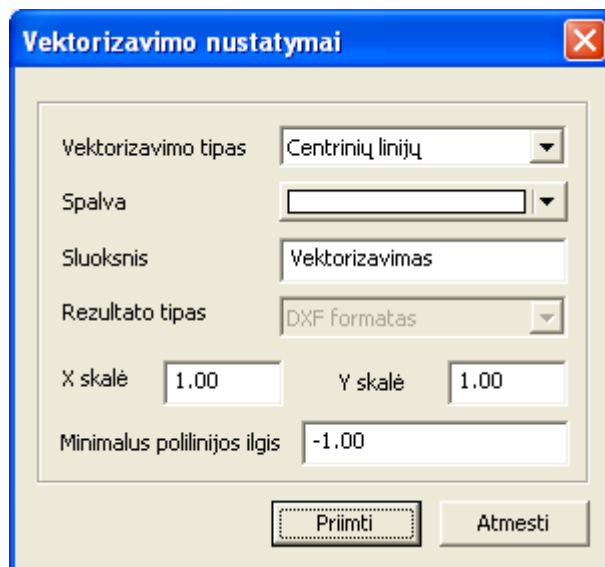
PRIEDAS. VEKTORIZAVIMO SISTEMOS VARTOTOJO SĄSAJA



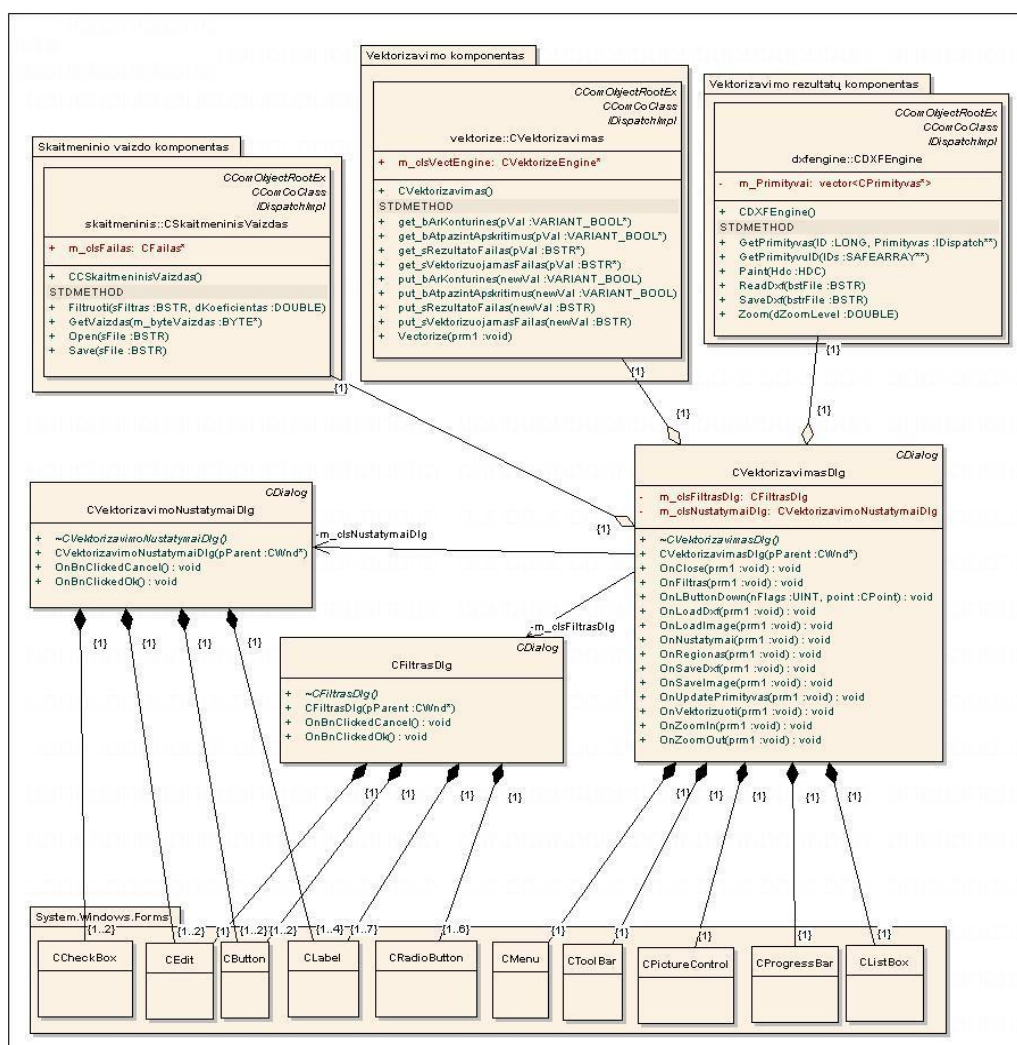
73 pav. Vektorizavimo sistemos pagrindinis valdymo langas



74 pav. Slenkstinio filtro valdymo langas



75 pav. Vektorizavimo nustatymų valdymo langas



76 pav. Vartotojo sąsajos detali klasių diagrama