

## Vaizdų technologijos T 111

OPTIMALIAUS INTEGRINIŲ GRANDYŲ TAŠKINIŲ VAIZDŲ VEKTORIZAVIMO  
METODO TYRIMASLeonas Jasevičius<sup>1</sup>, Giedrius Masalskis<sup>2</sup>*Vilniaus Gedimino technikos universitetas**El. paštas: <sup>1</sup>leonas@infoera.lt; <sup>2</sup>giedrius@masalskis.net*

**Santrauka.** Publikacijoje nagrinėjama integrinių grandynų sluoksnių taškinių vaizdų vektorizavimo problema. Aptariami įvairūs linijų taškiniuose vaizduose išskyrimo būdai, jų ypatybės. Pristatomas atrastas 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. Taip pat publikacijoje yra pristatomas integrinių grandynų atvirkštinio projektavimo procesas, jo taikymo sritys, etapai ir problemos.

**Reikšminiai žodžiai:** taškinis, vektorinis vaizdas, vektorizavimas, integriniai grandynai, lustai, atvirkštinis projektavimas.

**Įvadas**

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 sistemos atitikmenį kitoje formoje arba aukštesniame abstrakcijos lygmenyje. Lustų srityje šis procesas dažniausiai naudojamas atlikti suprojektuotų produktų verifikaciją topologijos (Bourbakis *et al.* 2002; Li *et al.* 1998) ar funkciniam lygmenyje, kuomet iš gautų projektavimo metu lusto sluoksnių gamybos šablonų duomenų gaunamas tranzistorių lygio aprašas, kuris yra konvertuojamas į loginių funkcijų lygmenį (Kosteljik *et al.* 1991), o iš šio gaunamas tikslus lusto elgsenos modelis, kuris yra naudojamas verifikuoti pradiniam projektuojamo lusto elgsenos modeliui (Lester *et al.* 1998). 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 etapus:

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 kaip lusto pakuočių atidengimas, lusto cheminis ar plazminis esdinimas 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.

Turint topologijos duomenis skaitmeniniame vektoriniame formate, 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) lygio modeliu.

Šioje publikacijoje yra pristatomas vektorinių lusto topologijos duomenų gavimo iš taškinių vaizdų metodas ir jo rezultatų priklausomybė nuo išankstinio taškinių vaizdų apdorojimo. Metodas sukurtas parenkant optimalią žinomų vektorizavimo algoritmų kombinaciją, kurios rezultatai geriausiai tinka lusto topologijos duomenų atstatymui iš taškinių vaizdų gautų fotografuojant lusto technologinius sluoksnius optiniu mikroskopu.

Terminas taškinis vaizdas reiškia dvimatę monochrominę šviesos intensyvumo funkciją. Taškinis vaizdas yra sudarytas iš vaizdo elementų. *Vektorizavimu* yra vadinamas procesas, kurio metu taškinis vaizdas yra paverčiamas vektoriniu vaizdu. Vektoriniu vaizdu galima vadinti aibę vektorinių esybių ir ryšius tarp tų esybių. Vektorizuojant lusto sluoksnių taškinius vaizdus svarbu gauti vektorinį vaizdą kuo artimesnį pradiniam lusto topologijos projektui. Tam reikia parinkti tinkamus taškinių vaizdų apdorojimo, linijų išskyrimo ir nuskaitymo metodus.

### 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:

1. Paremti transformavimu (angl. *Hough transform*). Metodai transformuoja skaitmeniniame vaizde esančias dideles erdvės struktūras į mažesnes struktūras parametro erdvėje (Dori 1997). Šis sprendimas padeda išvengti problemų susijusių su globalaus aptikimo 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 tašką turi aplankyti tik vieną kartą.

2. Paremti ploninimu (angl. *thinning*). Linijos ploninimo (skeletonizavimo) metodas paremtas iteraciniu visų vaizdo taškų tikrinimu (Tamura 1978). Kiekvieno tikrinimo metu yra šalinami ribiniai linijų taškai, taip galų gale gaunant skeletines linijas.

3. Paremti kontūro radimu (angl. *contour*). Pagrindinis šių metodų uždavinys yra rasti linijinių objektų kraš-

tus (kontūrus), ir tada apskaičiuoti ašinių linijų taškus tarp abiejų kontūro lygiagrečių (Nalwa 1993).

4. Paremti grafo perėjimu (angl. *run-graph*). Naudojama norint atvaizduoti skaitmeninį vaizdą kaip pusiau vektorinę struktūrą (Monagan *et al.* 1993). Tai palengvina linijų radimą bei kitus vektorizavimo veiksmus.

5. Paremti tinklelio modeliu (angl. *mesh pattern*). Skaitmeninis vaizdas tinklelio pagalba padalinamas į kvadratus, tada ieškoma charakteringų struktūrų tikrinant tik juodų taškų išsidėstymą prie kiekvieno tinklelio elemento. Išskyrus charakteringas struktūras yra sudaromas kontrolinis vaizdo žemėlapis, kurį analizuojant yra išskiriami vektoriai.

6. Paremti taškų sklaidymu (angl. *sparse pixel*). Kitaip žinomas kaip ortogonalus zigzagas (angl. *orthogonal zig-zag*). Paremtas vieno taško kelio sekimu. Kelias tampa stačiakampiu kai tik atsiranda ant juodų taškų ploto ribos. Kiekvieno kelio sekimo vidurinis taškas bus išskiriamos linijos taškas (Chai *et al.* 1992).

Visų šešių linijų išskyrimo metodų tipų palyginimas pateikiamas 1 lentelėje. Visi metodai išskyrus paremtus transformavimu yra įgyvendinami pagal tokią schemą:

1. Vidurinėsios ašies atrinkimas. Šios procedūros metu yra nustatomi taškai sudarantys centrinę vektorizuojamos linijos ašį.

2. Linijos sekimas. Atrinkti taškai yra susiejami į bendrą grandinę, atitinkančią vektorių.

3. Linijos segmento aproksimacija. Iš sudarytos grandinės pašalinami nekritiniai linijos taškai, ir sukuriama vektorius.

Geriausia linijų kokybė yra gaunama naudojant linijų ploninimu paremtus metodus.

### Taškinio vaizdo ploninimas

Taškinio vaizdo vektorizavimo procesą, kai naudojamas ploninimo metodas sudaro septyni žingsniai:

1. Taškinio vaizdo ploninimas. Ploninimo tikslas yra sumažinti taškinio vaizdo objektų plotį iki vieno vaizdo taško. Ploninimas gali būti atliekamas dviem būdais: išskiriant kontūrines arba centrinės objektų linijas.

1 lentelė. Vektorizavimo metodų palyginimas

Table 1. Comparison of vectorization methods

Metodas	Sudėtingumas	Linijos kokybė	Linijos storio išlaikymas	Susikirtimų 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
Taškų sklaidymo	linijinis	prasta	taip	taip

2. Susikirtimų žymėjimo procedūros paskirtis yra *taškinio vaizdo grafo* sudarymas. Tai nekryptingas grafas, kur susikirtimo taškai yra grafo viršūnės, o vaizdo taškų grandinės – grafo briaunos.

3. Grandinės sekimo operacijos tikslas yra taškiniame vaizde išskirti atskiras vaizdo taškų grandines ir paruošti jas segmentavimui.

4. Susikirtimų sujungimo funkcija suranda susikirtimo taškų, kurie yra labai arti vienas kito, poras ir pakeičia juos vienu susikirtimo tašku.

5. Grandinės apkarpymas pašalina nepageidautinas deformacijas prie susikirtimo taškų. Deformacijos atsiranda ploninimo metu, kadangi jis paprasčiausiai išskiria viduriniąją regiono ašį.

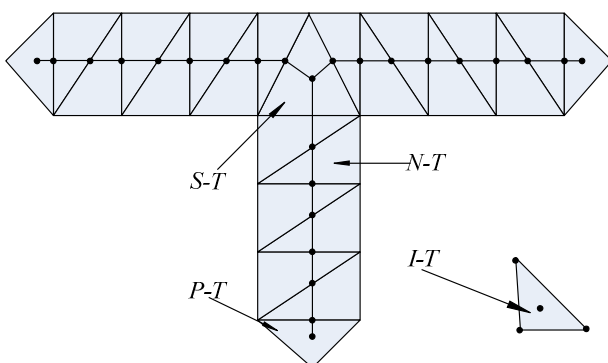
6. Segmentacijos tikslas yra suskaidyti grandines į mažesnius prasmingus *segmentus*, kurie gali būti paverčiami į geometrinius primityvus.

7. Segmentų sujungimas sujungia objektų segmentus gautus segmentacijos metu. Taip yra išlaikomas objektų vientisumas.

Tipinis ploninimo algoritmas, kurį pasiūlė Zhang ir Suen (1994), susideda iš dviejų žingsnių. Vykdamas ploninimą ieškoma vaizdo taškų, turinčių bent vieną foninį vaizdo tašką savo aštuoniuose kaimynuose,  $N_8(p)$ . Algoritmui baigus darbą, taškiniame vaizde lieka tik objektų skeletai, kurie yra naudojami tolesniame vektorizacijos procese.

### Delaunay trianguliacija

Norėdami sumažinti laiko sąnaudas ir pagerinti rezultatų kokybę, įgyvendinome Delaunay trianguliacija paremtą taškinio vaizdo ploninimo algoritmą, kuris yra patobulintas tipinis ploninimo algoritmas (Zou 2002).



1 pav. Vidiniai Delaunay trianguliacijos trikampiai

Fig. 1. Internal triangles used in Delaunay triangulation algorithm

Šio algoritmo esmė yra tokia: 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. 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 (1 pav.).

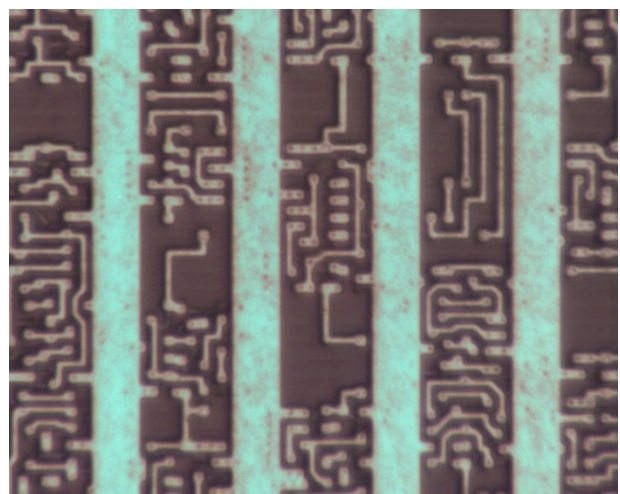
Figūros skeletas gautas iš Delaunay trianguliacijos gali turėti vadinamą *pakraščio* ar *susikirtimų* šiukšlių, kurios pašalinamos apjungiant jas sudarančius trikampius.

### Integrinių grandynų vaizdų vektorizavimas

Integrinių grandynų nuotraukos (taškiniai vaizdai) gaunamos fotografuojant atidengtus technologinius sluoksnius optiniu mikroskopu (2 pav.). Po pradinio apdorojimo šie vaizdai yra naudojami vektorizavimui.

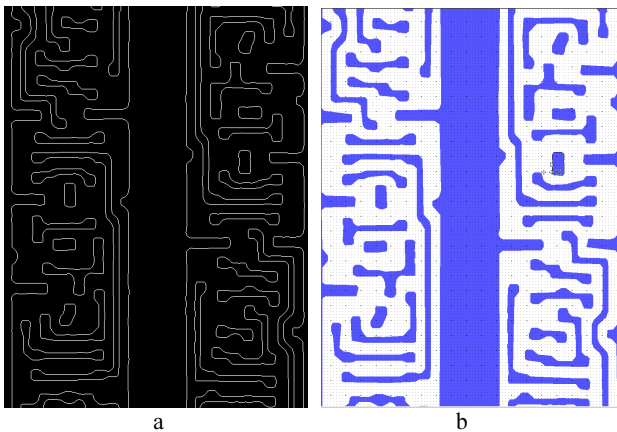
Prieš vektorizavimo etapą, IG nuotraukos yra apjungiamos į bendrą vektorizuojamo ploto taškinį vaizdą, išlyginamas pasukimo kampas. Apdorojant nuotraukas pritaikoma eilė įvairių filtrų, kurie išskiria vektorizavimo objektus – IG sluoksnio topologijos struktūras. Po vektorizavimo proceso gauti vektoriniai IG sluoksnio struktūrų duomenys yra išvedami IG CAD projektavimo programinei įrangai tinkamu formatu tolesnei analizei (3 pav.).

Pagrindinė užduotis vektorizuojant IG sluoksnių struktūras yra gauti duomenis kuo artimesnius pradiniam sluoksnių topologijos projektui. Pagrindinis reikalavimas vektorizavimo metodui yra topologijos struktūrų kontūrų linijos su minimaliu tarpinių taškų (verteksu) skaičiumi.



2 pav. IG sluoksnio nuotrauka gauta fotografuojant optiniu mikroskopu

Fig. 2. IC layer image obtained using optical microscope



**3 pav.** Vektorizavimo rezultatai. Filtruotas rastrinis vaizdas (a) ir vektorizavimo rezultatai IG projektavimo programoje (b)

**Fig. 3.** Vectorization results. Filtered raster image (a) and its vectorized form imported to IC CAD software (b)

Iš vidutinių vertekų postūmių nuo idealių rezultatų objektų, matome, kad tiksliausi vektorizavimo rezultatai gaunami, naudojant ploninimą, paremtą DT. Ploninimu paremti vektorizavimo algoritmai duoda tikslesnius metalo sluoksnio vektorizavimo rezultatus, nes juose yra galimybė nustatyti minimalų išskiriamos linijos ilgį. Dispersija atspindi labiausiai tikėtiną vertekso nukrypimą nuo vidutinės vertekų postūmio reikšmės. Gautos dispersijų reikšmės rodo, kad tiksliausias yra DT paremtas vektorizavimo algoritmas.

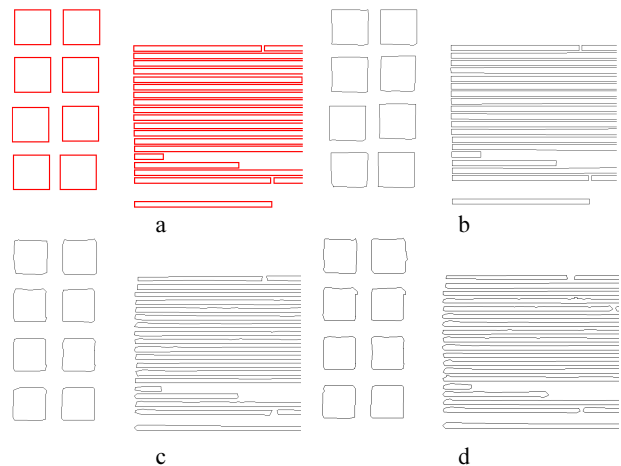
### Optimalaus vektorizavimo metodo nustatymas

Optimaliam vektorizavimo metodui surasti buvo kokybiškai įvertinta skirtingų ploninimo algoritmų įtaka galutiniams vektorizavimo rezultatams. Iš gautų duomenų matome, kad mažiausiai objektų vertekų gaunama naudojant vektorizavimo algoritmą kuris paremtas Delaunay trianguliacija (DT).

**2 lentelė.** Vektorizavimo rezultatų priklausomybė nuo ploninimo algoritmo

**Table 2.** Relation of vectorization results to used thinning algorithms

Algoritmas	Rezultatai	Laužčių vertekų skaičius	Vidutinis laužtės ilgis	Vidutinis vertekų postūmis nuo idealių rezultatų	Vertekų postūmių dispersija
Rankiniu būdu gauti idealūs vektorizavimo rezultatai		124	13,83	–	–
Vektorizavimas naudojant DT ploninimą		288	4,74	0,058	0,039
Vektorizavimas naudojant Canny kraštų išskirimą		821	1,91	0,211	0,148
Vektorizavimas naudojant tipinį ploninimą		674	2,66	0,105	0,076



**4 pav.** Vektorizavimo rezultatai naudojant skirtingus ploninimo algoritmus

**Fig. 4.** Vectorization results using different thinning algorithms

Vertekų skaičius rodo, kokio sudėtingumo yra gauti vektoriai, ku mažiau vertekų tuo paprasčiau atlikti modifikavimo veiksmus po vektorizavimo proceso. DT paremtas ploninimo algoritmas yra skirtas tiesių linijų išskirimui, todėl gaunamas mažesnis vertekų skaičius, lyginant su kitais tirtais algoritmais (2 lentelė). Kaip ir vektorių vertekų skaičius, taip ir vidutinis laužčių ilgis apibrėžia objektų sudėtingumą bei gautų vektorizavimo rezultatų panašumą į idealius vektorizavimo rezultatus.

4 pav. pateikti idealūs vektorizavimo rezultatai (a) ir vektorizavimo rezultatų, gautų naudojant DT ploninimą (b), tipinį ploninimą (c) ir Canny kraštų radimo algoritmą (d), fragmentai.

### Išvados

1. Šiame darbe aptarti taškinių vaizdų vektorizavimo metodai ir jų taikymas integrinių grandynų sluoksnių vaizdams, atliekant atvirkštinio projektavimo procesą.

Buvo nustatytas vektorizavimo metodas, geriausiai tinkantis IG sluoksnių struktūroms vektorizuoti. Optimalaus metodo kriterijus buvo ploninimo algoritmo įtaka vektorizavimo rezultatams, lyginant juos su idealiu vektoriniu vaizdu.

2. Palyginimo rezultatai rodo, kad statistiškai arčiau- siai idealaus vektorinio vaizdo buvo vektorizavimo rezul- tatai gauti naudojant Delaunay trianguliaciją. Šis metodas leidžia gauti tikslesnes, tiesesnes IG struktūrų linijas su netrūkia struktūra.

## Literatūra

- Bourbakis, N. G.; Mogzadeh, A.; Mertoguno, S; Koutsou- geras, C. 2002. A knowledge-based expert system for auto- matic visual VLSI reverse-engineering: VLSI layout version, *IEEE Transactions on Systems, Man and Cybernet- ics* 32(A): 428–436. doi:10.1109/TSMCA.2002.805765
- Chai, I.; Dori, D. 1992. Orthogonal zig-zag: An efficient method for extracting lines from engineering drawings, in *Visual Form*. New York, London: Plenum Press, 127–136.
- Dori, D. 1997. Orthogonal zig-zag: An algorithm for vectoriz- ing engineering drawings compared with Hough transform, *Advances in Engineering Software* 28(1): 11–24. doi:10.1016/S0965-9978(96)00035-X
- Kosteljik, T.; De Loore, B. 1991. Automatic verification of library-based IC designs, *IEEE Journal of Solid-State Cir- cuits* 26: 394–403. doi:10.1109/4.75019
- Lester, A.; Bazargan-Sabet, P; Greiner, A. 1998. AGLE, a sec- ond generation functional abstractor for CMOS VLSI cir- cuits, in *Proc. of the Tenth International Conference on Microelectronics*, 265–268. doi:10.1109/ICM.1998.825615
- Li, T.; Sung-Mo, K. 1998. Layout Extraction and Verification Methodology for CMOS I/O Circuits, in *Proc. of the 35th Design Automation Conference, San Francisco, USA*, 291– 296.
- Monagan, G.; Roosli, M. 1993. Appropriate base representation using a run graph, in *Proc. of the 2nd International Confer- ence on Document Analysis and Recognition, Tsukuba, Ja- pan*, 623–626.
- Nalwa, V. S. 1993. *A Guided Tour of Computer Vision*. New York: Addison-Wesley. ISBN 0-201-54853-4.
- Tamura, H. 1978. A Comparison of line thinning algorithms from digital geometry viewpoint, in *Proc. of the 4th Inter- national Conference on Pattern Recognition, Kyoto, Japan*, 715–719.
- Zhang, T.; Suen, C. 1984. A fast parallel algorithm for thinning digital patterns, *Communications of the ACM* 27: 236–239. doi:10.1145/357994.358023
- Zou, J. J.; Yan, H. 2002. Vectorization of cartoon drawings, *ACM International Conference Proceeding Series* 9: 77–78.

## INVESTIGATION OF OPTIMAL INTEGRATED CIRCUIT RASTER IMAGE VECTORIZATION METHOD

L. Jasevičius, G. Masalskis

Abstract

Visual analysis of integrated circuit layer requires raster image vectorization stage to extract layer topology data to CAD tools. In this paper vectorization problems of raster IC layer images are presented. Various line extraction from raster images algorithms and their properties are discussed. Optimal raster image vectorization method was developed which allows utili- zation of common vectorization algorithms to achieve the best possible extracted vector data match with perfect manual vec- torization results. To develop the optimal method, vectorized data quality dependence on initial raster image skeleton filter selection was assessed.

**Keywords:** raster image, vector image, vectorization, integrated circuit, chip, reverse engineering.