

LATVIJAS UNIVERSITĀTE

DATORIKAS FAKULTĀTE

# **IESPIEDPLAŠU ELEKTRISKĀS SHĒMAS ATPAZĪŠANA**

BAKALaura DARBS

Autors: **Jurģis Brauers**

Stud. apl. jb06026

Darba vadītājs:

Dr . sc. com. Kārlis Freivalds

Docents, Latvijas Universitāte

RĪGA 2011

## **Anotācija**

Attēlu atpazīšana ir viens no grūtākajiem uzdevumiem datoram. Uzdevumu atpazīt no attēla konkrētu objektu cilvēks varētu paveikt vienkāršāk un precīzāk nekā dators.

Darbā apskatītas metodes attēlu atpazīšanai, lai varētu atpazīt iespaidplašu elektriskās shēmas. Tiek pētīti jau esoši risinājumi par līdzīgu uzdevumu, automatisku ceļu atpazīšanu no aero vai satelītattēliem. Apskatītajām metodēm un risinājumiem tiek meklēts pielietojums iespaidplašu elektriskās shēmas atpazīšanā.

Atslēgvārdi: attēlu atpazīšana, norādes apli, iespaidplates

## **Abstract**

Image recognition is one of most difficult task for computer. Similar task to recognize object from image human coul'd do easier and precisely than computer.

This work looks on methods for recognizing printed circuit board electrical schema. Research has been done on similar task automatac road extraction from aerail and satellite images. Viewed methods and solutions are tried to use for printed circuit board electric schema recognition.

Keywords: image recognition, reference circle, printed circiut board

## Saturs

1. Ievads .....	5
2. Esošās elektrisko shēmu veidošanas programmas.....	6
2.1. PSPICE 9.1 .....	6
2.2. LTspice IV .....	6
2.3. EDwinXP .....	7
2.4. ProfiCad .....	8
2.5. Rimu Schematic.....	8
2.6. SmartDraw.....	9
2.7. Secinājumi .....	9
3. Izstrādājamās programmatūras vīzija .....	11
4. Esošo risinājumu analīze.....	12
4.1. Ceļu tīklu atpazīšana .....	12
4.2. Ceļu atpazīšana no aero un satelītattēliem .....	14
4.3. Secinājumi .....	17
5. Iespiedplašu shēmas atpazīšana.....	18
5.1. Attēla uzlabošana.....	18
5.1.1. Kontrasts .....	18
5.1.2. Slietņošana .....	19
5.1.3. Viena konkrēta pamatkrāsa .....	20
5.1.4. Citas metodes .....	20
5.2. Attēla sadalīšana .....	21
5.2.1. Robežu noteikšana .....	21
5.2.1.1. Prewitt maska .....	21
5.2.1.2. Sobel maska.....	22
5.2.1.3. Roberts maska .....	23
5.2.1.4. Laplacian maska .....	24
5.2.2. Canny robežu noteikšana .....	25
5.2.3. Norādes apli.....	26
5.2.4. Centrālie pikseļi.....	27
5.3. Punktu atpazīšana .....	27
5.4. Secinājumi .....	28
6. Rezultāti .....	29
7. Secinājumi .....	31
8. Izmantotā literatūra un avoti: .....	32
1. Pielikums. Kompaktdisks.....	33

## 1. Ievads

Darbā apskatīta iespēja automātiski atpazīt iespiedplates elektrisko shēmu no attēla. Apskatīti līdzīgi risinājumi par automātisku ceļu atpazīšanu no aero un satelītattēliem. Tāpat tiek pētītas dažādas attēla apstrādes funkcijas un to ietekme uz iespiedplašu attēliem.

Darba mērķis ir iegūt vispārējas zināšanas par attēlu apstrādi un atpazīšanas iespējām un izpētīt pieejamos materiālus un algoritmus iespiedplašu elektriskās shēmas atpazīšanas programmas izveidei. Līdz šim vēl nav veidotas šāda veida programmas, kas atpazītu elektriskās shēmas no attēliem, jo attēlu atpazīšana datoram ir viens no sarežģītākajiem un grūtākajiem uzdevumiem. Tomēr cilvēkam attēlu atpazīšana veicās vieglāk, atšķirīgās domāšanas pēc.

Pirmajā nodaļā apskatītas esošo elektrisko shēmu veidošanas programmas, ka arī izpētītas iespējas šādām programmām ievaddatos padot trešo pušu veidotus datus.

Otrajā nodaļā aprakstīta jaunizveidojamā programma, kas veiktu iespiedplašu elektriskās shēmas atpazīšanu. Tiek sniegts ieskats programmas galvenajās funkcijās un raksturojošās iezīmes.

Trešā nodaļā izpētīti pieejamie risinājumi. Tā kā nav pieejami risinājumi, par darbā aprakstīto tēmu, tad tiek pētīti līdzīgi risinājumi, šajā gadījuma ceļu atpazīšana no aero un satelītattēliem.

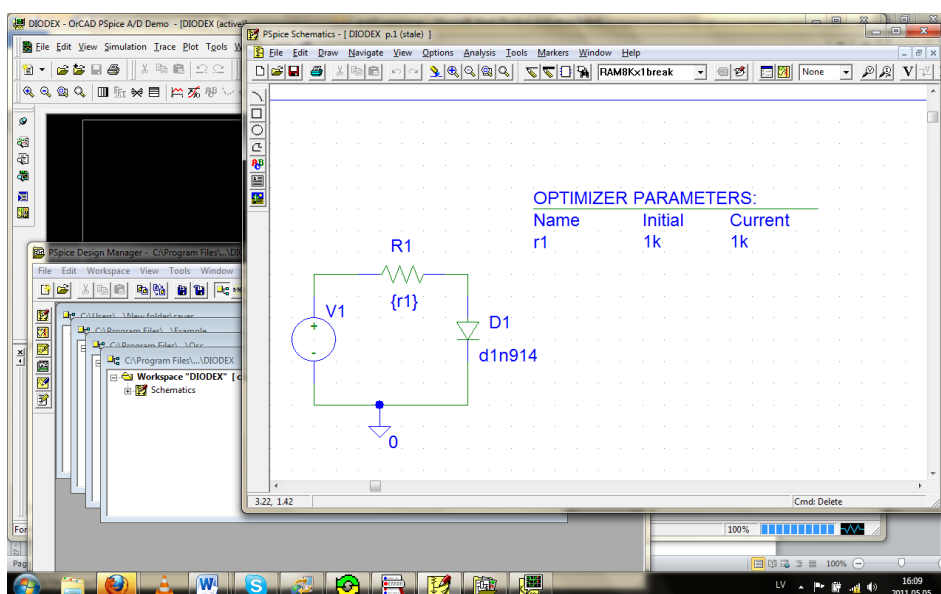
Ceturtajā nodaļā aprakstītas dažādas attēlu apstrādes funkcijas un to pielietojuma derīgums darba mērķa risināšanai.

## 2. Esošās elektrisko shēmu veidošanas programmas

Šajā nodaļā apskatītas dažādas elektrisko shēmu veidošanas programmas. Tāpat ir apskatīti arī failu formāti, ar kādiem šīs programmas strādā, kā arī iespēja ģenerēt failus konkrētajai programmai izmantojot citas programmas. Citas programmas: piemēram, programma, kuras vīzija aprakstīta šajā darbā.

### 2.1.PSPICE 9.1

*PSPICE 9.1*[1] ir vairāku programmu kopums, kas nodrošina darbu ar elektriskām shēmām, kā arī ar iespiesto plašu projektēšanu. Pieejama arī programmas komponente, kas paredzēta gan analogu, gan digitālu shēmu simulēšanai.

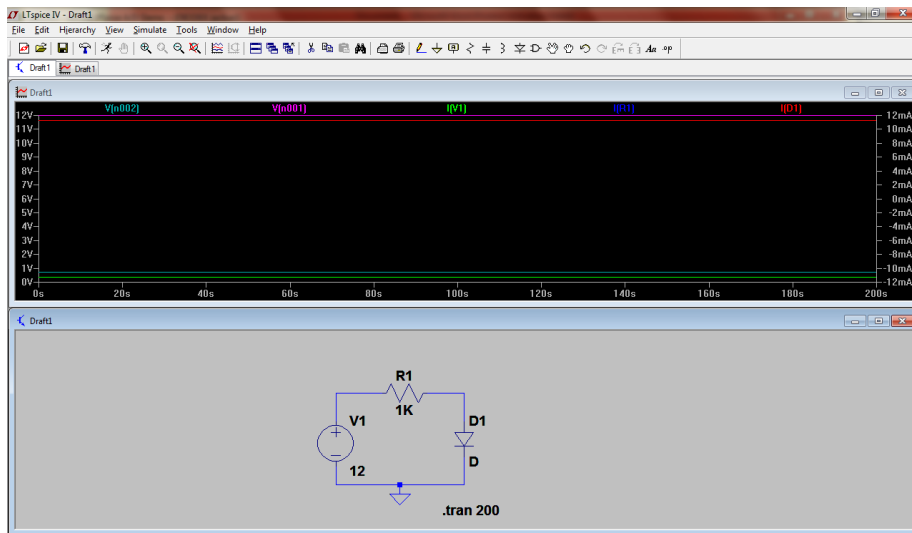


#### 2.1. att. Programmas *PSpice 9.1* shēmas veidošanas logs

Shēmas iespējams saglabāt \*.sch failu formātā, kas ir tekstuāls, lasāms formāts, vai arī \*.edif formātā. \*.edif (*Electronic Design Interchange Format*)[2] elektriskās projektēšanas apmaiņas formāts tika veidots ar mērķi izveidot vienotu apmaiņas formātu elektriskās projektēšanas automatizācijai.

### 2.2.LTspice IV

*LTspice IV*[3] līdzīgi kā *PSPICE 9.1* arī šī programma nodrošina elektrisko shēmu izveidošanas labošanas un simulēšanas iespējas. Pieejams plašs klāsts iepriekš definētu elektrisko komponentu shēmu veidošanai.

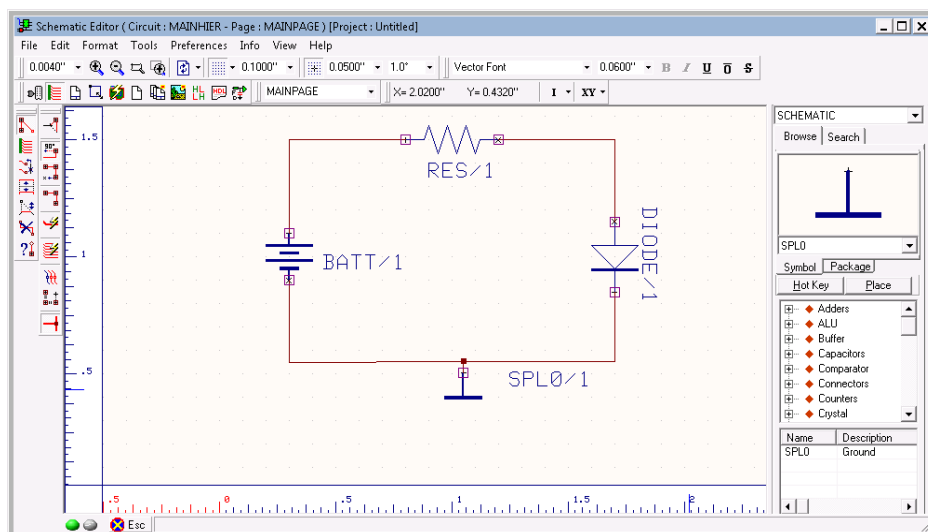


## 2.2. att. Programmas *LTspice IV* shēmas veidošanas un simulēšanas logs

Programma nodrošina darbu tikai ar \*.asc formāta failiem, tomēr tie ir lasāmi un viegli uztverami, kā arī piemēroti rediģēšanai izmantojot citas programmas.

## 2.3. EDwinXP

*EDwinXP*[4] iespējams veidot un labot elektriskās shēmas. Pieejams arī liels klāsts elektrisko komponentu shēmām. Papildus tam var arī saglabāt sevis veidotu elektrisku komponenti, ko vēlāk var izmantot kādā citā shēmā. Programma atbalsta shēmu simulēšanu.

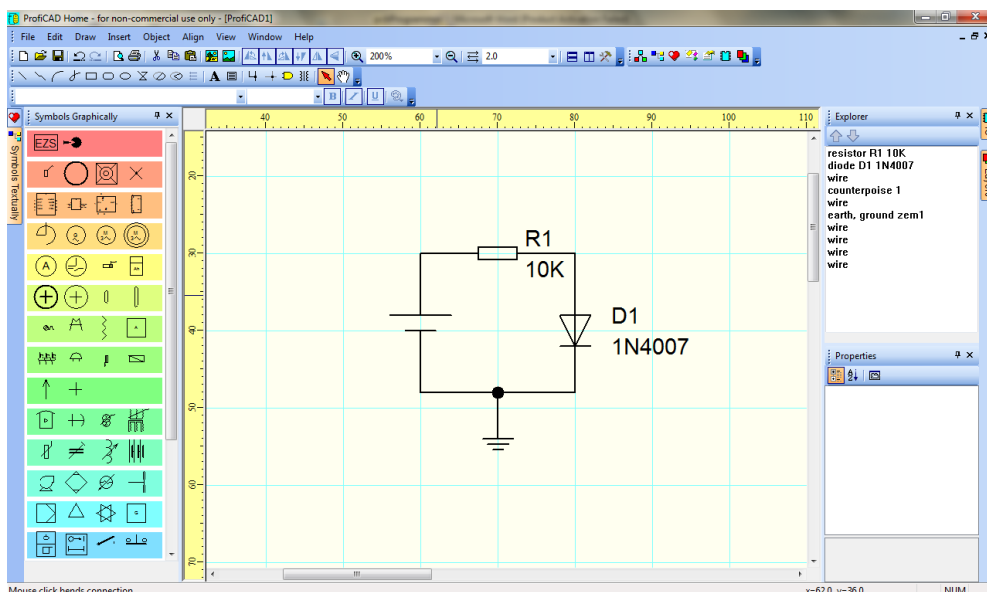


## 2.3. att. Programmas *EDwinXP* shēmas veidošanas logs

Failu saglabāšanai tiek izmantoti \*.epx formāta faili. Šie faili nav lasāmi, līdz ar to ja tādus mēģina veidot ar kādu citu programmu, nepieciešama papildus informācija. Tomēr programma nodrošina importa funkcionalitāti izmantojot \*.jedec, \*.cupl, \*.xilinx, \*.altera formātu failus, kas ir paredzēti dažādu shēmu apmaiņai ar citām programmām.

## 2.4. ProfiCad

*ProfiCad*[5] programma, kas ir zīmēšanas nevis veidošanas rīks, tomēr neskatoties uz to pieejams neliels klāsts ar elektriskajām komponentēm, ko var izmantot zīmēšanā.

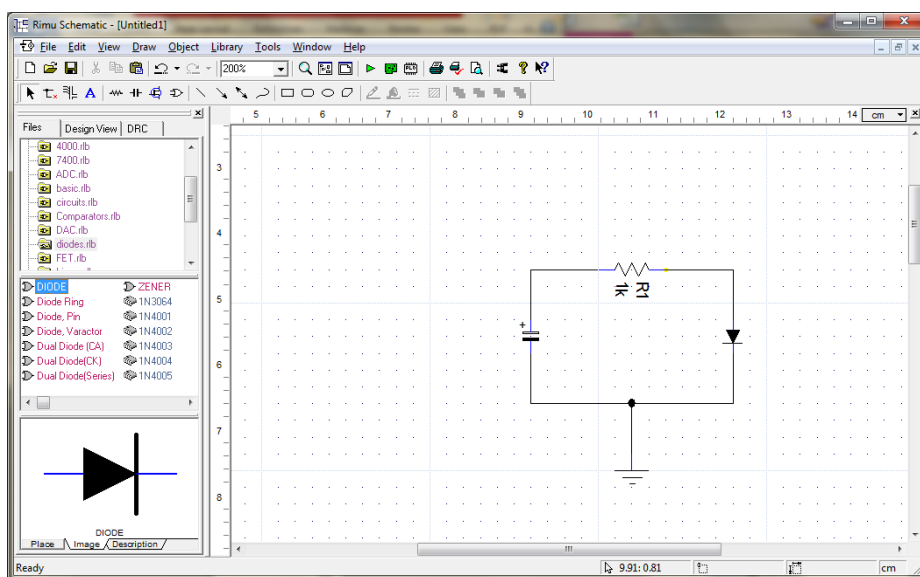


2.4. att. Programmas *ProfiCad* shēmas veidošanas logs

Darbs notiek tikai ar \*.sxe formāta failiem, nav pieejams imports vai iespēja atvērt citu formātu failus. Šādi saglabāti faili ir lasāmi un to struktūra ir veidota XML valodā.

## 2.5. Rimu Schematic

*Rimu Schematic*[6] programma elektrisko shēmu veidošanai un labošanai. Pieejams plašs klāsts ar elektriskajām komponentēm, kā arī ir iespēja pašam veidot kādas komponentes, ko vēlāk var izmantot. Izveidotās shēmas iespējams arī simulēt.

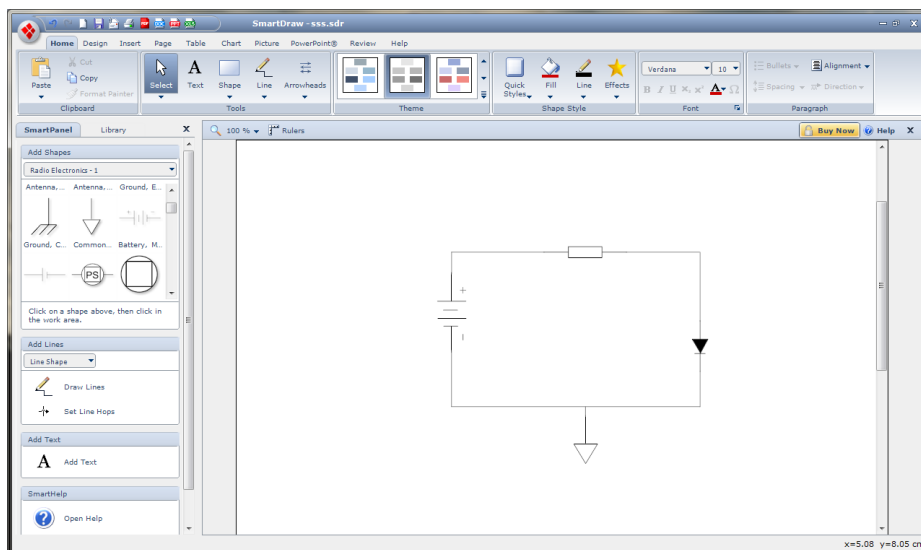


2.5. att. Programmas *Rimu Schematic* shēmas veidošanas logs

*Rimu Schematic* darbs notiek tikai ar \*.sch failiem. Šajos failos esošā informācija tiek saglabāta binārā veidā, padarot tos grūtāk izmantojamus citam programmām. Programmā nav realizēts failu imports.

## 2.6. SmartDraw

*SmartDraw*[7] programma, kas ir zīmēšanas rīks ar iespēju zīmēt arī elektriskās shēmas. Shēmu zīmēšanai pieejams neliels klāsts elektrisko komponentu.



### 2.6. att. Programmas *SmartDraw* shēmas veidošanas logs

*SmartDraw* darbs notiek ar \*.sdr formāta failiem. Failos informācija tiek glabāta binārā veidā, kas apgrūtina iespēju šādus failus ģenerēt izmantojot trešo pušu programmas.

## 2.7. Secinājumi

Ir pieejamas dažādas programmas elektrisko shēmu zīmēšanai gan veidotas speciāli šim uzdevumam, gan universālas, kas paredzētas dažādu plānu un shēmu zīmēšanai. Nevienai no programmām nav iespēja atpazīt iespiedplates elektrisko shēmu vai tās daļu no attēlā vai kādā citā veidā. Universālās jeb zīmēšanas programmas pārsvarā saglabā datus bināros failos, kuriem nepieciešama papildus informācija, lai to varētu izmantot trešo pušu programmas kā datu ievadi. Savukārt speciālās elektrisko shēmu veidošanas programmas pārsvarā nodrošina datu saglabāšanu lasāmos failos, kas atvieglo šādu failu veidošanu citām programmām, arī autora iecerētajai programmai. Otra iespēja veidot failus visiem zināmos formātos, ko konkrētā programma spēj importēt un tādā veidā iegūt datu ievadi. Autora iecerētajai programmai nav paredzēti sarežģīti ievades dati, tikai elektrisko celiņu tīkls, to savienojumu un lodējumu vietas. Šādā gadījumā nav nepieciešams izvēlēties sarežģītu datu formātu, bet gan tādu, kas maksimāli atvieglo programmēšanas darbus. Speciālās shēmu veidošanas

programmas piedāvā izveidotām shēmām pārbaudīt pareizību, un iespēju arī simulēt shēmas darbību. Priekš iecerētās programmas vispiemērotākais atzīts *LTspice IV* veidotais \*.asc failu formāts, jo tas ir viegli lasāms un intuitīvi saprotams, tāpat pati programma arī nodrošina shēmas simulēšanas iespēju.

### 3. Izstrādājamās programmatūras vīzija

Balstoties uz iepriekšējā nodaļā izpētītajām elektrisko shēmu veidošanas programmām, tika secināts, ka neviena no apskatītajām programmatūrām nenodrošina iespēju atpazīt shēmu pēc attēla. Līdz ar to ir nepieciešams izveidot jaunu programmu, vai kādas esošas programmas komponenti, kas šādu iespēju nodrošinātu. Jaunizstrādājamās programmas būtu neatkarīga programma, kas darbināma no katra datora uz kura tā ir uzstādīta un pieejamas sekojošas galvenās funkcijas:

- Iespēja izvēlēties konkrētus reģionus celiņu atpazīšanai,
- Automātiska bildes apstrāde, kas izceļ celiņus,
- Manuāla korigēšana bildes apstrādei,
- Automātiska celiņu atpazīšana,
- Priekšskatījums atpazītajiem celiņiem,
- Dažāds izejas failu formāts, atbilstoši izvēlētajai programmai.

Jaunizstrādājamā programmatūra ir neatkarīga no šobrīd pieejamām elektrisko shēmu veidošanas programmām, bet tomēr nodrošinātu sasaiti ar tām, veidojot failus, kurus var atvērt vai importēt izmantojot konkrētu programmu. Programma izveidotu failu, kas būtu atpazīts elektriskās plates celiņu tīkls, ko varētu izmantot kāda no elektrisko shēmu veidošanas programmām, ka ievades datus elektriskās shēmas veidošanai.

## 4. Esošo risinājumu analīze

Iepazīstoties ar bibliotēkā pieejamo informāciju par elektrisko shēmu atpazīšanu, secināts, ka līdz šim nav veikti kādi pētījumi iespiedplašu shēmu automātiskā atpazīšanā no attēliem.

Tā kā jau iepriekšējās šī darba nodaļās noskaidrots, ka nav vēl izveidota programma, kas spēj automātiski atpazīt elektrisko shēmu no iespiedplates un nav arī veikti pētījumi šajā jomā, tad tiek apskatīti risinājumi, kas ir līdzīgi dotajam uzdevumam. Piemēram, ceļu atpazīšana no satelītu vai aerofoto attēliem. Līdzības, kas saskatāmas abu veidu attēlos:

- Attēli pieejami dažādā kvalitātē un ar atšķirīgu trokšņu līmeni,
- Ceļu un celiņu platums var būt mainīgs,
- Ceļu un celiņu novietojums un iedomāts virziens nav noteikts,
- Ir apkārtējās vides traucējumi ceļiem – ēnas, mašīnas, dažādi segumi, elektriskajiem celiņiem – dažādi uzraksti uz platēm, lodējumu vietas,
- Fons var būt gan viendabīgs, gan raibs,
- Fonā var būt redzamas vietas, kas izskatās kā atpazīstamie objekti, kurus nav vajadzības izdalīt.

Atšķirības, kas novērojamas abu veidu attēlos:

- Vairumā gadījumu visi ceļi būs tieši savienoti savā starpā, kamēr uz plates var būt vairāki pilnīgi neatkarīgi elektriskie celiņi,
- Celiņiem svarīgas ir arī lodējumu vietas, ne tikai ceļi un krustojumi,
- Celiņiem pārsvarā kopējais laukums būs lielāks kā fonam, ceļiem otrādāk.

Tā kā būtiskākās lietas gan ceļiem, gan elektriskajiem celiņiem ir līdzīgas, tad ceļu atpazīšana no satelītu vai aerofoto attēliem ir uzskatāma par piemērotu izziņas materiālu meklēšanas virzienu.

### 4.1. Ceļu tīklu atpazīšana

Sekojošās nodaļas izklāsts balstīts [8] pieejamo informāciju.

Lai iegūtu informāciju par ceļiem no attēliem, iespējams izvēlēties pusautomātiskās pieejas, kurās operators atzīmē dažus sākumpunktus priekš ceļu atpazīšanas algoritma. Šo pieeju var attīstīt līdz pilnīgi automātiskai sistēmai, kur sākumpunkti tiek izvēlēti automātiski, tomēr šādu metožu rezultāti ir ļoti atkarīgi no izvēlētajiem sākumpunktiem. Lai uzlabotu šo algoritmu darbību, tiek izvēlēta pieeja izmantot Markova objektu procesu, lai modelētu

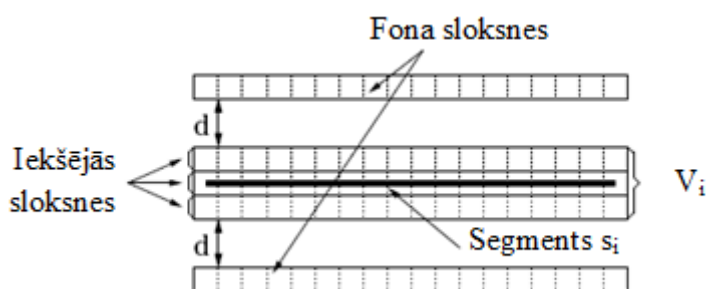
autoceļu tīklu, kā arī ir izstrādāts „Quality Qandy” modelis, kurš izmanto kvalitātes koeficientus starp objektiem, lai labāk modelētu līkumus un ceļu krustojumus.

Ceļu tīklu pamatā ir raksturo mijiedarbības: nepārtrauktība, izliekumi, krustojumi noteiktos leņķos. Ja ceļu tīkls tiek uzskatīts kā segmentu saraksts, šīs savstarpējās iedarbības var tikt uzskatītas kā mijiedarbības starp segmentiem. Pirmā mijiedarbība ir savienojums  $e$  diviem segmentiem. Brīvie un vien punktu segmenti netiek apskatīti. Kvalitāti katram savienojumam ar segmentiem  $s_i$  un  $s_j$  nosaka koeficients  $g_c(\{s_i, s_j\})$ , kurš tiek rēķināts no segmentu galu attāluma un novietojuma. Savienojumi lielā leņķī tiek pieņemti, ja diviem segmentiem  $s_i$  un  $s_j$  ar savstarpējo novietojumu  $\theta_i$  un  $\theta_j$ ,  $g_c(\{s_i, s_j\}) < 0$  ja  $\min\{|\theta_i - \theta_j|, \pi - |\theta_i - \theta_j|\} < \tau$ , kur  $\tau$  ir pozitīvs sliekšnis. Turklāt sliktas mijiedarbības orientācija ir nepieciešama, lai izvairītos no segmentu pāriem, kuru viduspunkti ir pārāk tuvu, kā arī no paralēliem segmentiem. Aprēķinātais potenciāls ( $g_o(p) > 0$ ) ir vienmēr pozitīvs visiem segmentu pāriem( $p$ ), kuriem ir slikta orientācija. Blīvums tiek aprēķināts izmantojot sekojošu formulu:

$$h_p(S) \propto \beta^n \exp(-w_1 n_f - w_2 n_s - w_3 \sum_{p \in C} g_c(p) - w_4 \sum_{p \in O} g_o(p))$$

Kur  $w_{i,i=1..4}$  ir pozitīvi, konstanti svāri,  $\beta > 0$  norāda procesa intensitāti,  $n$  kopējais segmentu skaits,  $n_f$  brīvo segmentu skaits,  $n_s$  vien punktu segmentu skaits,  $C$  un  $O$  attiecīgi saistīto un slikti orientēto segmentu pāri,  $g_c$  un  $g_o$  attiecīgi koeficients un potenciāls. Šis blīvums nosaka lokālu, stabilu punktu procesā, kas garantē ergodisku konvergenci atgriezeniskajam Markova ķēdes Monte Carlo algoritmam.

Datu nosacījumi tiek balstīti uz sekojošiem pieņēmumiem: ( $H_1$ )pelēkā līmeņa izmaiņa ceļam un fonam ir liela un ( $H_2$ )lokālā vidējā pelēkā līmeņa vērtība ceļam ir viendabīga. Lai pārliecinātos, ka segments  $s_i$  pieder datiem, apskata pikseļu kopu  $V_i$  atbilstoši segmentam un divas blakus esošas sloksnes, kas pieder fonam. Šīs divas sloksnes atrodas attālumā  $d$  no  $V_i$ , lai būtu iespēja apskatīt dažādu platumu ceļus, kā tas ir redzams 4.1. att. Ceļa līnija ar tai piederošām iekšējām sloksnēm un fona sloksnes.



4.1. att. Ceļa līnija ar tai piederošām iekšējām sloksnēm un fona sloksnes

Nemot vērā pikseļu vērtības katrā reģionā populācijas izlasei, Stjudenta t-tests tiek izmantots, lai noteiktu vai abu reģionu vidējie rādītāji nozīmīgi atšķiras. Ar  $T_{H_1}(s_i)$  vērtību tiek pārbaudīta pirmā pieņēmuma  $H_1$  atbilstību. Otrais pieņēmums  $H_2$  pārbaudīts sadalot  $V_i$  vairākās sloksnēs un izskaitļojot  $T_{H_2}(s_i)$ . Statistiskā vērtība  $t_i$  atbilstošajam  $s_i$  ir šo divu pieņēmumu rezultātu attiecība, pietam papildus nosacījums, ka  $T_{H_2}(s_i)$  jābūt mazākam par 1, lai izvairītos no dalīšanas ar bezgalību, kā arī lai izvairītos no pārāk viendabīgiem reģioniem

$$t_i = \frac{T_{H_1}(s_i)}{\max\{1, T_{H_2}(s_i)\}}$$

Tad izmantojot sliekšņošanu un  $t_i$  vērtības pārvēršanu no  $[0, \infty]$  uz  $[-1, 1]$ , lai saglabātu dubulto potenciālu,  $\delta_i$  saistot ar segmentu  $s_i$  var izvēlēties negatīvās un pozitīvās vērtības. Visbeidzot konfigurācijas enerģija  $S$  tiek definēta ka summa no  $S$  piederošajiem

$$h_d(S) \propto \exp(-\gamma_d \sum_{s_i \in S} \delta_i)$$

potenciāliem. Kopumā datu nosacījumi ir: Kur  $\gamma_d$  ir pozitīvs konstants svars.

Lai varētu iegūt ceļu tīklus no attēliem, nepieciešams noteikt konfigurāciju, kura ir vislielākā blīvumam  $f$  izmantojot *Poisson* procesa mērījumus. Optimizācija panākta imitējot atdzišanu, kas sastāv no veiksmīgas blīvuma  $f^{1/T}$  sadales  $\pi_T$  simulācijas,  $T$  pakāpeniski samazinot līdz 0. Atgriezeniskais Markova ķēdes Monte Carlo algoritms tika izvēlēts sadales  $\pi_T$  simulēšanai, kas sastāvēja no diskrētas Markova ķēdes imitēšanai ar Metropolis-Hastings-Green dinamiku. Šis iteratīvais algoritms nepaļaujas uz sākuma stāvokli, bet katrā posmā pāreja no pašreizējā stāvokļa  $S$  uz jauno stāvokli  $S'$  atbilstoši kodola piedāvājumam  $Q(S \rightarrow .)$ . Pāreja tiek apstiprināta balstoties uz *Greena* attiecību. Šī apstiprināšanas attiecība tiek rēķināta, lai pārbaudītu balansa nosacījumus. Tie ir nosacījumi uz kuriem algoritms konverģē uz nenormētu mērījumu no telpas punkta procesa  $\pi_T$ .

## 4.2. Ceļu atpazīšana no aero un satelītattēliem

Šīs nodaļas izklāsts balstīts [9] pieejamo informāciju.

Ceļu atpazīšana no attēliem ļoti atkarīga no ceļu attēlojuma. Šajā rakstā apskatīta ceļu atpazīšana no augstas izšķirtspējas attēliem, piemēram, satelītu un aerofoto. Šādas izšķirtspējas attēliem ceļi vairs nav tieva līnija, un mašīnas un koki ir viegli identificējami. Šādi attēli ir bagāti ar informācijas daudzumu, diemžēl tas uzreiz sarežģī ceļu atpazīšanas automatizāciju. Dažādi objekti dažādi izvietoti, un to ēnas vai līdzīgās iezīmes izmaina cits citu. Arī paši ceļi var izskatīties atšķirīgi vienā attēlā. Līdz ar to ir neiespējami paredzēt kur atradīsies kāds objekts un kā tas izskatīsies. Visas šīs atšķirības padara automātisku ceļu

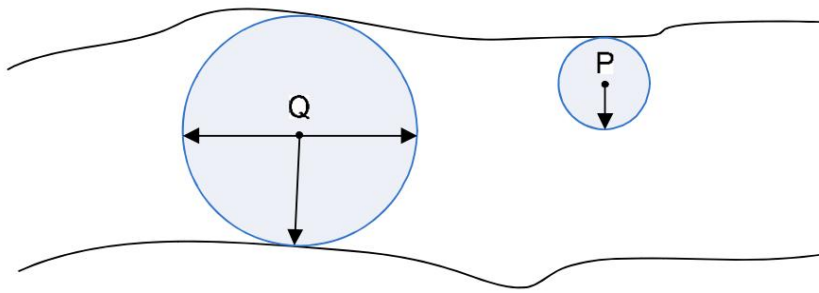
atpazīšanu ļoti sarežģītu. Lai arī ir daudz pētīts, vēl arvien nav efektīvas metodes, kas spēj izšķirt dažādos attēlus.

Rakstā tiek izstrādāta shēma automātiskai ceļu atpazīšanai, kas balstās nevis uz ceļu robežu noteikšanu, bet gan pikseļu meklēšanu, kas pieder konkrētajam ceļam. Pētījumi parāda, ka ceļi ir ļoti mainīgi no to fiziskā stāvokļa, ka arī no materiāla, kas padara neiespējami izveidot ceļa modeli pēc krāsas vai spektra. Shēma uzskata ceļu kā līdzīgu pikseļu grupu. Jauna shēma vispirms apskata vizuālās un ģeometriskās pikseļu īpašības izmantojot jaunu metodi – norādes apli (angl. *Reference circle*). Tad tiek izvēlēti centrālie pikseļi katram reģionam. Visi reģioni tiek pārbaudīti pēc dažādām vizuālām un ceļu iezīmēm, tādējādi izslēdzot mājas jumtu, vai garu zālāju identificēšanu kā ceļu.

Mērķis piedāvātajai sistēmai ir izstrādāt pilnīgu un praktiski automātisku risinājumu ceļu atpazīšanai. Tā sastāv no diviem galvenajiem posmiem: atpazīt viegli identificējamus ceļus un atpazīt grūti identificējamus ceļus. Daži ceļi ir viegli identificējami, jo tie ir izcelti un satur mazu trokšņu līmeni. Ceļi vienā attēlā pārsvarā ir līdzīgi un informācija, kas iegūta no viegli identificējamiem ceļiem, piemēram, spektrs, var tik izmantota grūti identificējamu ceļu atpazīšanai. Otrā daļa kalpo kā papildinājums pirmajai daļai, lai aizpildītu iztrūkstošos posmus. Tādējādi divu posmu process nodrošina maksimālu ceļu atpazīšanas precizitāti. Katrs posms sastāv no trim galvenajiem soļiem: filtrēšana, segmentēšana un grupēšana un optimizēšana.

Divi pieņēmumi, kas nepieciešami, lai ceļš būtu identificējams: vairums viena ceļa pikseļu ir līdzīgs spektrs, kas atšķirās no fona, ceļš ir šaurāks un garāks reģions kā pārējie fona objekti. Šie divi pieņēmumi ir atšķirīgi no vairākuma līdzšinējo pētījumu. Vizuālajam nosacījumam nepieciešams tikai lai ceļš atšķirtos no vairākuma fona, nevis būtu viss vienkrāsains. Ģeometriskajam nosacījumam nepieciešama, lai ceļa forma būtu šaura un gara, nevis ar izteiktu viendabīgu malu. Šādi nosacījumi ir maz prasīši no attēla un nodrošina ceļu pārrāvumu un dažādu objektu rezultātu neietekmēšanu.

Filtrēšanas solis identificē atslēgas pikseļus, kas palīdzēs noteikt, vai reģions ir ceļš, balstīts uz vizuālo nosacījumu. Izmantojot robežu atpazīšanas metodes, iegūts attēls, kurš nenosaka ceļus, bet labākajā gadījumā reģionus, kur varētu būt ceļi. Lai pilnīgi identificētu un atdalītu ceļu reģionus no attēla, piedāvāta jauna metode, kas sevī ietver divas galvenās idejas: norādes apli un centrālo pikseli. Pikselis  $p$  un tā norādes aplis  $C(p)$  ir lielākais aplis ar centru  $p$ , kurš nesatur robežu punktus. Pikselis ir centrālais, ja tam pieder vislielākais aplis salīdzinot ar tā tiešajiem kaimiņiem.



**4.2. att. Norādes apli**

Attēlā 4.2. att. Norādes apli attēloti divi pikseļi Q un P, no kuriem Q ir centrālais pikselis. Motivācija šiem abiem konceptiem ir, ka ideālos apstākļos visi centrālie pikseļi veido ceļa centra līniju. Augstā kļūdu pielāide nodrošina, ka nepilnīgu un neprecīzu robežu gadījumā centrālie pikseļi nobīdīsies tikai nedaudz. Katram centrālā pikseļa norādes aplī esošie centrālie pikseļi ideālā gadījumā ir centra līnija ceļam, tomēr vairumā gadījumu var novilkt līniju, izmantojot mazāko kvadrātu metodi (angl. *method of least squares*). Filtrēšanas soļa izvads ir kopa ar centrālajiem pikseļiem katram reģionam. Šajā gadījumā centrālo pikseļu meklēšanai attēlam vispirms izmantota robežu atpazīšana. Otrās daļas filtrēšanas solī objekti, kas ietekmē ceļu atpazīšanu var tikt izņemti izmantojot morfoloģisku atjaunošanu, jo ceļa spektrs ir zināms pēc pirmās daļas.

Segmentācijas solis nepieciešams, lai noteiktu kurš reģions ir iespējams ceļa reģions skatoties pēc centrālajiem pikseļiem. Centrālie pikseļi satur ne tikai centra līniju reģionam, bet arī reģiona kopējo formu. Piemēram, kvadrātam būs tikai viens centrālais pikselis, bet šauram un garam reģionam būs daudz centrālie pikseļi ar ļoti līdzīgiem norādes apļa lielumiem. Tādējādi attiecība kopējam centrālajiem pikseļu skaitam pret vidējo norādes apļa lielumu ir labs reģiona formas attēlotājs. Tikai reģioni, kam ir konkrēts attiecības lielums ir uzskatāmi, kā ceļu kandidāti. Minimālais reģiona platums ir noteikts, lai maksimāli izfiltrētu vajadzīgo informāciju pirmajā daļā, tomēr šis platums otrajā daļā netiks noteikts. Izmantojot *Hough* pārveidojumus tiek pārbaudīti kolineārie centrālie punkti pēc to virziena.

Segmentācijas rezultātā lielā trokšņa dēļ, ceļa segmenti parasti ir nepilnīgi un pārtraukti. Mērķis šim solim ir sagrupēt šādus segmentus. Optimizācija var tikt izlaista pirmajā daļā, ja ceļa spektra noteikšanai ir pietiekoši daudz datu. Ir divas problēmas šajā solī - kuri segmenti ir saistīti un ja tie ir saistīti, tad kā sagrupēt vienā. Pirmajai problēmai algoritms pārbauda sekojošās īpašības:

- Kolineāri centrālie pikseļi,
- Līdzīgs spektrs,

- Līdzīgs vidējais platums,
- Tuvums,
- Sasaiste.

Tādējādi mazu pārtraukumu starp diviem kolineāriem segmentiem algoritms sagrupē vienā grupā. Arī ja divus kolineārus segmentus savieno trešais segments, ar līdzīgu spektru, apvieno vienā grupā. Lai atrisinātu otro problēmu, katrai ceļu segmentu grupai tās centra līnija tiek aprēķināta izmantojot centrālo pikseļu līnijas. Kas būtībā nozīmē, ka segmenti tiek optimizēti, likvidējot šauros leņķus starp punktiem. Leņķis ir šaurs, ja tas ir lielāks kā  $\pi/8$ . Lai savienotu pārtrauktus segmentus, savienojums aprēķināts lai izlīdzinātu leņķus abos galos. Kad ceļa centra līnija ir atrasta, ceļu var iezīmēt balstoties uz šo līniju un ceļa platumu aprēķinātu pēc centrālo pikseļu norādes apļa vērtībām.

### 4.3. Secinājumi

Ceļu atpazīšanai no satelītu vai aerofoto piedāvā dažādas metodes kuras izmanto dažādus paņēmienus, tomēr vairākums izmanto kādu no robežu atpazīšanas algoritmiem un arī kādu no attēla izlīdzināšanas algoritmiem. Ceļu atpazīšanai pārsvarā tiek veikti sekojoši soļi:

1. Attēla uzlabošana,
2. Attēla sadalīšana derīgos segmentos vai dažādu robežu noteikšana,
3. Atrasto segmentu vai robežu grupēšana un apvienošana.

Pirmajā solī pārsvarā tiek izmantoti dažādi vienkārši filtri trokšņa samazināšanai. Citas metodes izmanto dažādus līdzekļus, lai izceltu ceļus un samazinātu fonu. Otrajā solī attēls tiek sadalīts izmantojot specifisku algoritmu, vai arī meklētas robežas ar kādu no pieejamiem robežu noteikšanas algoritmiem, piemēram, *Canny*, *Prewitt* vai *Sobel* robežu noteikšanas algoritmi. Trešajā solī tiek izmantotas dažādas transformācijas un specifiski algoritmi. Tā kā elektrisko shēmu atpazīšana ir līdzīga ceļu atpazīšanai, tad arī veicamie soļi iespiedplašu atpazīšanai būs līdzīgi.

## 5. Iespiedplašu shēmas atpazīšana

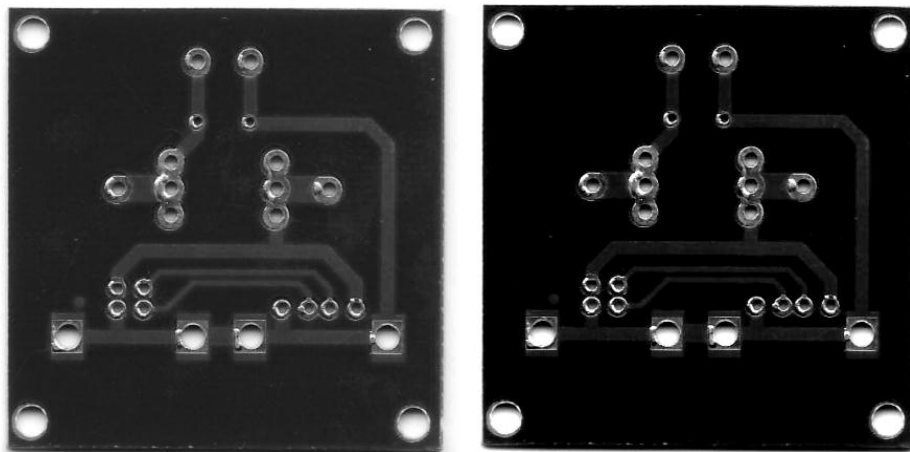
Izpētot risinājumus par ceļu atpazīšanu, redzams, ka ir dažādas pieejas ceļu atpazīšanā, bet tomēr pamat soļi pārsvarā visos risinājumos ir līdzīgi. Šajā nodaļā apskatītas pirmo divu soļu iespējamās metodes. Tiek pieņemts, ka iespiedplašu attēliem būs nebūtiski mazs trokšņu līmenis, kas neiespaidos galarezultātus.

### 5.1. Attēla uzlabošana

Elektrisko celiņu atpazīšanu ļoti atvieglo viennozīmīga derīgo datu izšķiršana no attēla fona. Šajā nodaļā aplūkota iespiedplašu apstrāde ar dažādiem algoritmiem, ar mērķi maksimāli izcelt vajadzīgo informāciju. Galvenā ideja ir panākt, lai fons viscaur būtu maksimāli viendabīgs, bet celiņi un lodējumu vietas būtu viegli un viennozīmīgi atšķirami no fona. Izvēlētām shēmu ilustrācijām ir zems trokšņu līmenis, tādēļ nav apskatītas metodes trokšņu samazināšanai, bet gan datu izcelšanai.

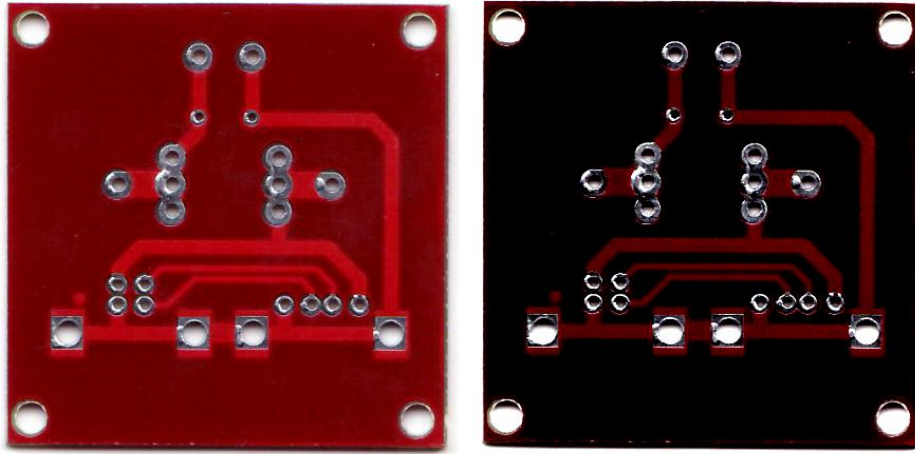
#### 5.1.1. Kontrasts

Kontrasts ir vizuālo īpašību atšķirība, kas padara objektu vieglāk atšķiramu no citiem objektiem un arī no fona[10]. Kontrastu nosaka krāsu, kā arī to spilgtumu atšķirība diviem objektiem, vai arī objektam no fona. Jo lielāka ir krāsu un spilgtuma atšķirība, jo lielāks kontrasts.



5.1. att. Oriģināls un palielināta kontrasta attēls

Attēlā 5.1. att. Oriģināls un palielināta kontrasta attēls attēlota parasta iespiedplate, un tā pati plate, tikai ar palielinātu kontrastu. Rezultāti ir pozitīvi, jo jaunizveidotajam attēlam praktiski viss fons ir viendabīgi melns un tikai celiņi un to lodējumu vietas ir izcelti. Tāpat kontrasta palielināšana nav izmainījusi attēlojamās datus. Nav saplūduši kopā vairāki attēla elementi.



**5.2.att. Oriģināls un palielināta kontrasta krāsains attēls**

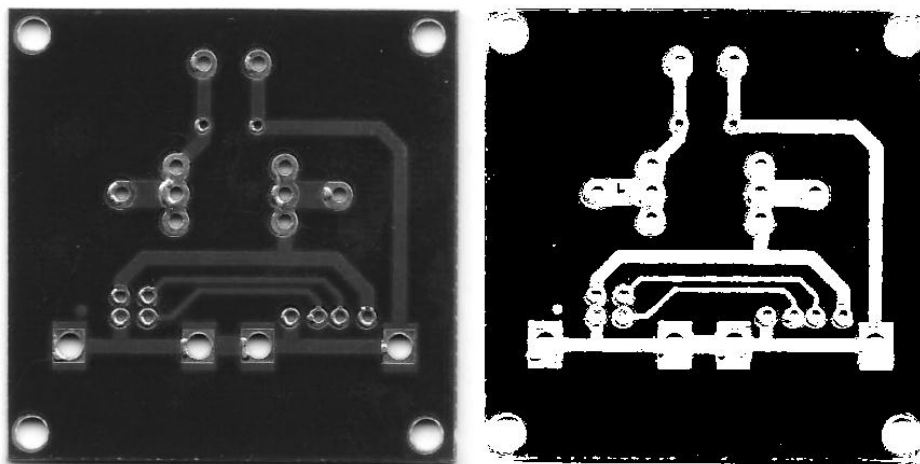
Palielinot kontrastu arī krāsainam attēlam 5.2.att. Oriģināls un palielināta kontrasta krāsains attēls iegūst pozitīvus rezultātus ar labi saskatāmiem celiņiem un viendabīgu fonu. Līdz ar to nav īpaša atšķirība vai attēla kontrasta korekcijas tiek veiktas krāsainam vai melnbaltam attēlam. Turklāt saglabājot atsevišķas krāsas celiņiem un lodējumu vietām, tiks atvieglots darbs pilnai iespiedplates atpazīšanai.

### **5.1.2. Slietkšņošana**

Slietkšņošana (angl. *Thresholding*)[11] ir attēla sadalīšana divās daļās, kuras pēc tam vieglākai uztverei tiek aizstātas ar attiecīgi baltiem vai melniem pikseļiem. Visvienkāršākā slietkšņošana ir iespējama, ja tiek skatīts katra punkta pelēkās krāsas līmenis un salīdzināts ar iepriekš definētu slietksni  $T$ .

$$g(x,y) = \begin{cases} 1 & \text{ja } f(x,y) > T \\ 0 & \text{ja } f(x,y) \leq T \end{cases}$$

Šāda attēla transformācija rezultātā dos bināru attēlu, kuru tālāk var izmantot apstrādē.



### 5.3. att. Oriģināls un sliekšņots attēls

Attēlā 5.3. att. Oriģināls un sliekšņots attēls pielietota sliekšņošana ar sliekšņa vērtību  $T = 52$ . Kā redzams ir panākts vēlams efekts ar izceltiem plātes celiņiem, tomēr šim attēlam lodējumu vietas ir saplūdušas kopā un šādā veidā attēls nav piemērots izmantošanai tālākai apstrādei.

#### 5.1.3. Viena konkrēta pamatkrāsa

Mūsdienās iespiestās plātes pieejamas dažādās lakošanas krāsās:

- Zaļa,
- Zila,
- Sarkana,
- Melna,
- Balta.

Kā zināms katra krāsa sastāv no attiecīgām daļām katras pamatkrāsas sarkanas, zaļas un zilās. Sadalot attēlu pamatkrāsās, iegūst attēlu, kuram katrs punkts ir aprakstīts ar konkrētās krāsas intensitāti. Apskatot attēlus, kas ir iespiedplašu attēla pamatkrāsas veidots, redzams, ka netiek panākts vēlamais efekts vairumam attēlam. Tas ir tāpēc, ka fona un celiņu krāsas atšķiras ļoti maz un arī katra pamatkrāsa gan celiņiem gan fonam ir līdzīga. Tiem attēliem, kuru sadalīšana pamatkrāsās, sniedz vēlamo rezultātu celiņu izcelšanā, līdzvērtīgu rezultātu var panākt arī ar kontrasta palielināšanu. Šiem attēliem pārsvarā atšķiras celiņu un fona krāsas. Līdz ar to šī pieeja nav noderīga kā attēlu uzlabošanas metode.

#### 5.1.4. Citas metodes

Lai uzlabotu kādu attēlu ne vienmēr pietiek tikai ar kontrasta palielināšanu vai sliekšņošānu. Dažu attēlu uzlabošanā tika izmantotas arī citas metodes. Piemēram, krāsu

piesātinājuma mainīšana, pretējo krāsu izmantošana un krāsu līmeņu mainīšana. Atsevišķu attēlu uzlabošanai neieciešams izmantot vairāku metožu kombināciju.

## 5.2. Attēla sadalīšana

Tā kā lielāko daļu no izvēlētajiem iespiedplašu attēliem izdevās uzlabot, izmantojot kādu no aprakstītajām metodēm, tad šajā nodaļā esošos pētījumus var balstīt izmantojot jau uzlabotus attēlus.

### 5.2.1. Robežu noteikšana

Robežu noteikšanai ir svarīga loma attēla sadalīšanai segmentos, jo lielākā daļa segmentēšanas algoritmu izmanto kādu no robežas noteikšanas metodēm. Vienkāršākie robežu noteikšanas algoritmi izmanto maskas. Robežu noteikšana ir krāsas līmeņa izmaiņa konkrētā vietā. Tas nozīmē ka robežas var tikt noteiktas ar konkrētu virzienu. Un lai uzzinātu visas attēla robežas nepieciešams šādas maskas pielietot „pagrieztas”, jeb izmainīt maskas vērtības, lai tā atrastu robežu citā virzienā. Robežu noteikšanai tika izvēlētas *Sobel*, *Prewitt*, *Roberts* un *Laplacian* maskas, kuru vērtības aprakstītas[11]. Kā arī apskatīts *Canny* robežu noteikšanas algoritms.

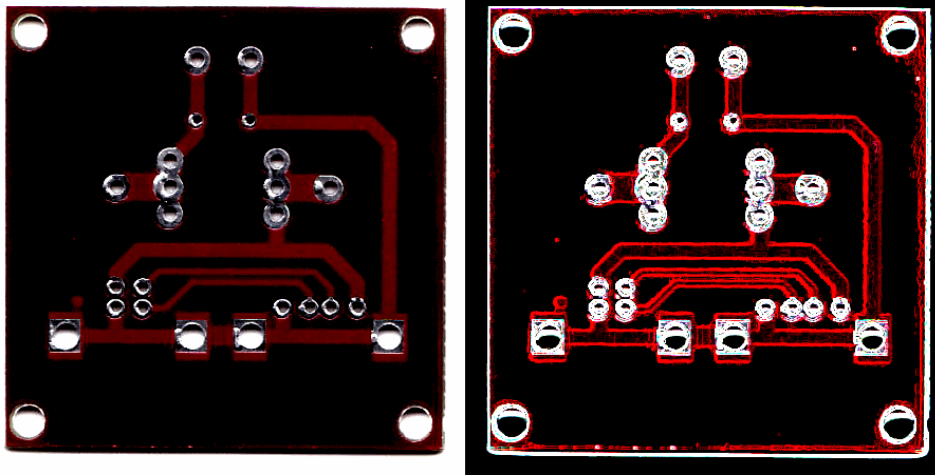
#### 5.2.1.1. Prewitt maska

*Prewitt* maskas piedāvātās vērtības dažādos virzienos attēlotas tabulā 5.1. tabula *Prewitt* maskas vērtības.

5.1. tabula *Prewitt* maskas vērtības

Robežas virziens	0°	45°	90°	135°
Maska	-1 -1 -1	0 -1 -1	1 0 -1	1 1 0
	0 0 0	1 0 -1	1 0 -1	1 0 -1
	1 1 1	1 1 0	1 0 -1	0 -1 -1
Robežas virziens	180°	225°	270°	315°
Maska	1 1 1	0 1 1	-1 0 1	-1 -1 0
	0 0 0	-1 0 1	-1 0 1	-1 0 1
	-1 -1 -1	-1 -1 0	-1 0 1	0 1 1

Aprēķinot visu masku summāro vērtību katram attēla pikselim iegūst robežu vērtību katrā punktā. Uzskatāmības palielināšanai to var pārveidot arī krāsainā attēlā.



5.4. att. Oriģināls un *Prewitt* maskas robežas noteikts attēls

Robežu noteikšanas galarezultāts nav piemērots tālākai izmantošanai, tomēr attēlā 5.4. att. Oriģināls un *Prewitt* maskas robežas noteikts attēls ir izceltas lodējumu vietas, kas var palīdzēt vēlāk punktu meklējumos.

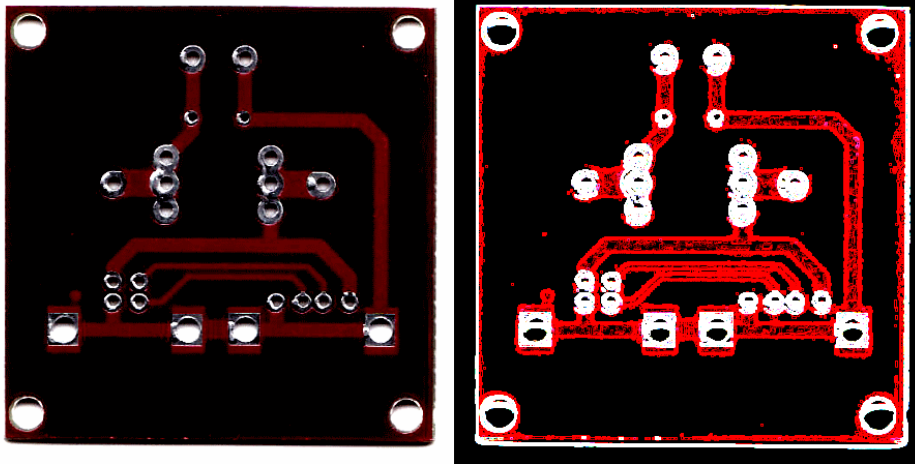
#### 5.2.1.2. *Sobel* maska

*Sobel* piedāvātās maskas vērtības apskatāmas tabulā 5.2. tabula *Sobel* maskas vērtības.

5.2. tabula *Sobel* maskas vērtības

Robežas virziens	0°	45°	90°	135°
Maska	-1 -2 -1	0 -1 -2	1 0 -1	2 1 0
	0 0 0	1 0 -1	2 0 -2	1 0 -1
	1 2 1	2 1 0	1 0 -1	0 -1 -2
Robežas virziens	180°	225°	270°	315°
Maska	1 2 1	0 1 2	-1 0 1	-2 -1 0
	0 0 0	-1 0 1	-2 0 2	-1 0 1
	-1 -2 -1	-2 -1 0	-1 0 1	0 1 2

Aprēķinot visu masku summāro izmantojot tādu pašu metodi, kā *Prewitt* operatoram.



**5.5. att. Oriģināls un *Sobel* maskas robežas noteikts attēls**

Kā redzams, attēlā 5.5. att. Oriģināls un *Sobel* maskas robežas noteikts attēls, arī šis robežas noteikšanas algoritms sapludina kopā celiņus, tādējādi sabojājot attēlā attēlotos datus. Tomēr līdzīgi kā *Prewitt* operatoram arī *Sobel* operators izceļ lodējumu vietas.

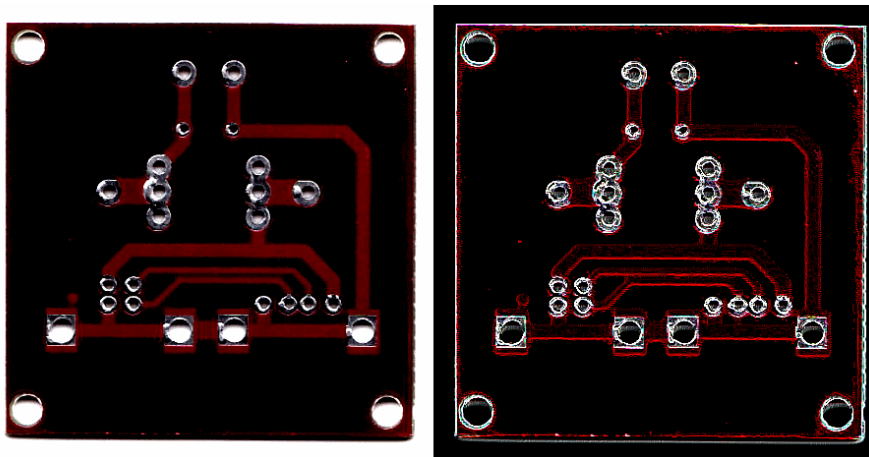
*5.2.1.3. Roberts maska*

*Roberts* maskai atšķirībā no iepriekšējām maskām izmērs ir 2x2, masku vērtības redzamas tabulā 5.3. tabula *Roberts* maskas vērtības.

**5.3. tabula *Roberts* maskas vērtības**

Robežas virziens	45°	135°	225°	315°																
Maska	<table border="1"> <tr> <td>-1</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> </tr> </table>	-1	0	0	1	<table border="1"> <tr> <td>0</td> <td>-1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>	0	-1	1	0	<table border="1"> <tr> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>-1</td> </tr> </table>	1	0	0	-1	<table border="1"> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>-1</td> <td>0</td> </tr> </table>	0	1	-1	0
-1	0																			
0	1																			
0	-1																			
1	0																			
1	0																			
0	-1																			
0	1																			
-1	0																			

Pielietojot tikai šīs četras maskas katram attēla punktam Galarezultātā iegūts attēls 5.6. att. Oriģināls un *Roberts* maskas robežas noteikts attēls.



**5.6. att. Oriģināls un Roberts maskas robežas noteikts attēls**

Šajā attēlā redzams, ka robežas ir ļoti vāji un neizteiksmīgi noteiktas. Tuvumā esošie celiņi ir saplūduši kopā.

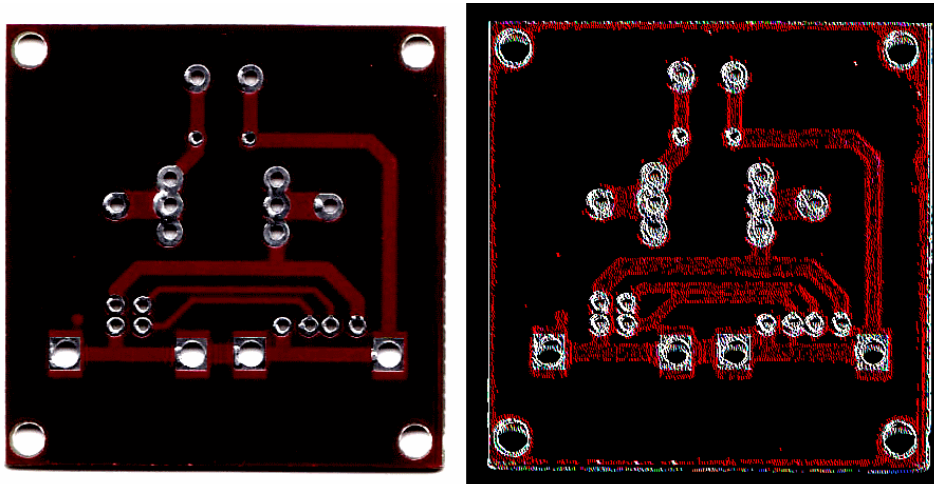
#### 5.2.1.4. *Laplacian maska*

*Laplacian* maskas būtiskākā atšķirība ir tās uzbūvē, jo tā ir simetriska un nav nepieciešams šo masku rotēt, lai iegūtu pilnīgu gala rezultātu. Logaritmiskās *Laplacian* maskas vērtības attēlotas tabulā 5.4. tabula *Laplacian* maskas vērtība.

**5.4. tabula *Laplacian* maskas vērtība**

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Pielietojot šādu masku attēla katram pikselim, iegūst rezultātu, kurš redzams attēlā 5.7. att. Oriģināls un *Laplacian* maskas robežas noteikts attēls.

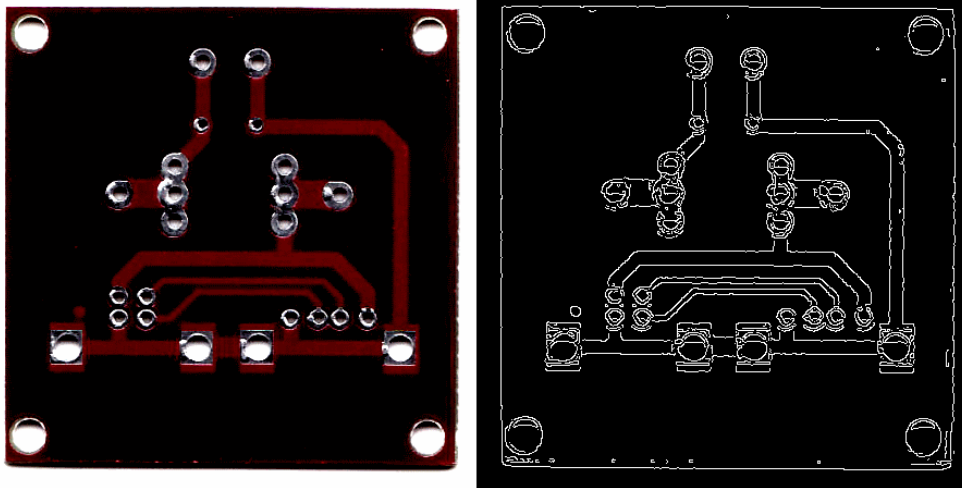


5.7. att. Oriģināls un *Laplacian* maskas robežas noteikts attēls

Iegūtais attēls ir sliktas kvalitātes, jo šī maska nav noteikusi izteiktas robežas, bet gan atrasti daudz dažādi punkti vietās kur jābūt elektriskajiem celiņiem un lodējumu vietām.

### 5.2.2. Canny robežu noteikšana

Canny[12] robežu noteikšanai izmanto vairāku soļu algoritmu robežu noteikšanai. Algoritms sastāv no trokšņu samazināšanas, attēla gradienta intensitātes meklēšanas, maksimumu izcelšanas, robežu trasēšanas un histerēzes sliekšņošanas, ģeometriskās variācijas formulēšanas. Trokšņu samazināšanai tiek izmantots Gausa filtrs, kas izejā dod attēlu kurš ir nedaudz izplūdis. Attēla gradienta intensitātei tiek izmantots kāda no robežu noteikšanas metodēm, piemēram, *Roberts*, *Prewitt*, *Sobel*. Atrastās robežas tiek sagrupētas vertikālajās horizontālajās vai diagonālās robežās. Maksimumu izcelšanai tiek meklēti lokālie maksimumi iepriekš sagrupētajās robežās. Sliekšņošanai izmanto divus sliekšņus augstu un zemu. Augstais sliekšnis nepieciešams, lai atbrīvotos no trokšņu radītiem rezultātiem, kuri nav vajadzīgi. Savukārt zemais sliekšnis nepieciešams malu izsekošanai, kur lokālo maksimumu izcelšana devusi zemu vērtību. No šiem izejas datiem ar dažādu ģeometrisku variāciju palīdzību iegūst galarezultātu – nepārtrauktu robežas līniju.

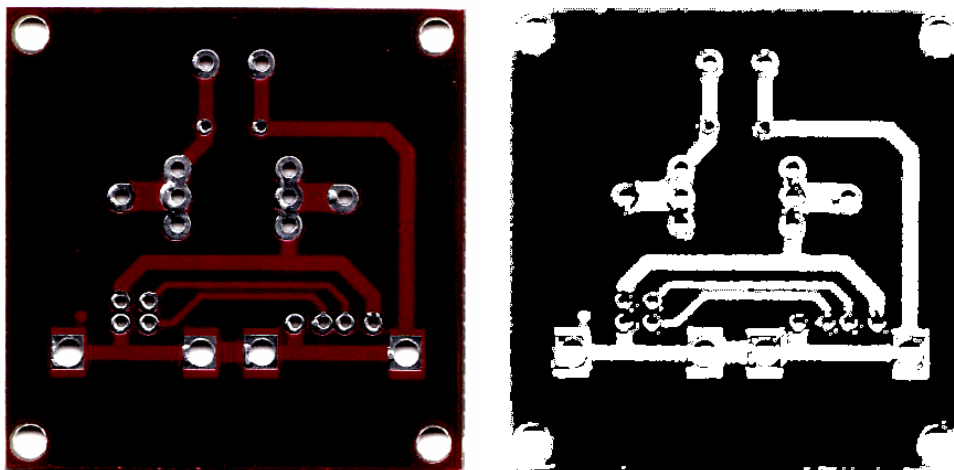


**5.8. att. Oriģināls un *Canny* robežas noteikts attēls**

Attēlā 5.8. att. Oriģināls un *Canny* robežas noteikts attēls redzams tiešsaistē pieejamā *Canny* robežu atpazīnēja[13] ievade un izvade. Redzams, ka ir precīzi atrastas robežas un iezīmētas to kontūras.

### 5.2.3. Norādes apli

Norādes apli jau ir pieminēti trešajā nodaļā. Kur tie tiek piedāvāti kā jauna metode daļa ceļu atpazīšanai. Attēlā 5.9. att. Oriģināls un norāžu apli noteikts attēls redzami norādes apli veidotie reģioni.

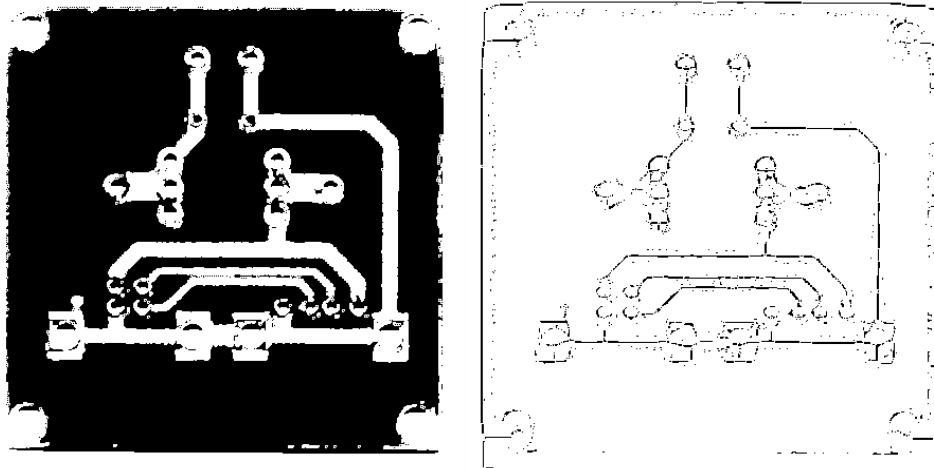


**5.9. att. Oriģināls un norāžu apli noteikts attēls**

Šie reģioni iegūti no ievades attēla to sliekšņojot ar sliekšņa vērtību  $T = 10$ , kad visi pikseļi, kas ir zem sliekšņa, tiek uzskatīti par fonu un tiek izmantoti, kā konkrētā norādes apla ierobežojošie elementi. Attēlā redzams, ka divi celiņi ir daļēji saplūduši kopā, tomēr tas neietekmē rezultātu, ja kopā saplūdušajās vietās vienlaidus saplūdumi ir mazāki kā saplūstošo celiņu attiecīgie platumi.

#### 5.2.4. Centrālie pikseļi

Kā jau tas tika aprakstīts, norādes apļiem vieniem pašiem nav plaša pielietojuma, tomēr, ja no šiem apļiem tiek meklēti centrālie pikseļi, tad norādes apļi kļūst svarīgi. Centrālo pikseļu uzdevums ideālā gadījumā ir iezīmēt konkrētā celiņa viduslīniju.



5.10. att. Norāžu apļu un tiem atbilstoši centrālie pikseļi

Rezultātā iegūts attēls 5.10. att. Norāžu apļu un tiem atbilstoši centrālie pikseļi, kuram iegūtie centrālie pikseļi veido izteiksmīgas celiņu līnijas un arī lodējumu vietās tiek veidotas lodējumu aprises. Šāds attēls ir pilnībā pietiekošs, lai varētu tālāk to apstrādāt ar kādu grupēšanas, kas sagrupētu iespiedplates punktus ar atbilstošajām līnijām.

#### 5.3. Punktu atpazīšana

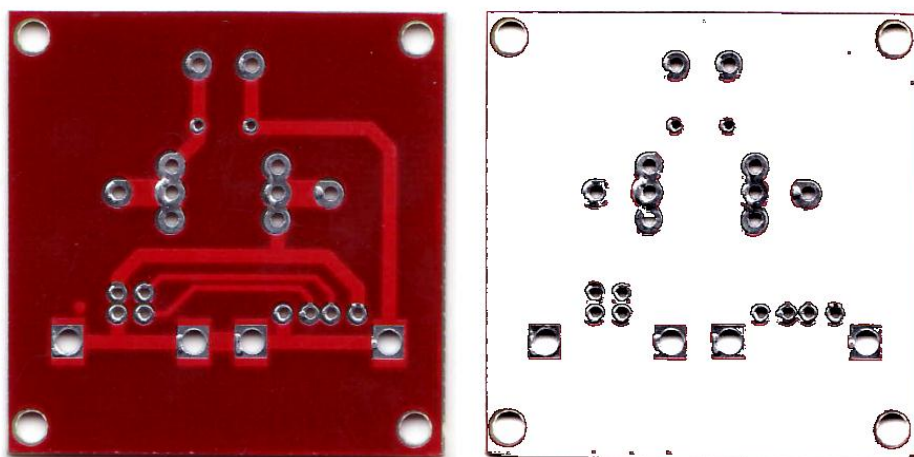
Līdz šim pārsvara apskatīta tikai celiņu izcelšana un atpazīšana, kas ir vajadzīga, tomēr pilnīgai iespiedplates atpazīšanai nepieciešams arī korekti atpazīt punktus jeb lodējumu vietas. Šādiem punktiem ir atšķirīgi izmēri un uzbūve:

- Apaļš punkts celiņa galā vai vidū, ar atšķirīgiem izmēriem,
- Vairāki apaļi punkti cieši blakus, kas ir savienoti neizmantojot celiņu,
- Taisnstūra veida punkts.

Problēmu pastiprina arī fakts, ka zinot celiņu platumu, nevar viennozīmīgi noteikt punkta izmērus, un otrādi. Tomēr atpazīšanu var uzlabot fakts, ka šie punkti nav noklāti ar laku un tādējādi ir redzama to dabiskā krāsa - sudrabota.

Punktu atpazīšanai var izmantot kādu konkrētu masku, kas atpazītu lodējumu vietas, tomēr katra šāda vieta ļoti daudz atšķiras no citām un pārsvarā pat uz vienas plates novērojami vairāku formu veidu un izmēru punkti.

Daudz efektīvāks veids punktu atpazīšanai izmantot labāko no robežas noteikšanas rezultātā iegūtajiem attēliem. Atkarība no ievades attēla var iegūt dažādus rezultātus, bet ja iespiedplates lodējumu vietas ir to dabiskajā krāsā un nav pārklātas ar laku, tad robežu noteikšanas algoritmi izceļ šīs vietas baltā krāsā. Plates fons ir melns, bet celiņu krāsa atbilst ievades faila celiņa krāsai. Šīs vietas var droši uzskatīt par meklētajiem punktiem, gadījumā ja viens punkts sevī ietver vairākas lodējumu vietas, tad var uzskatīt, ka tās ir savstarpēji savienotas, vai arī apstrādi var atkārtot ar augstākas izšķirtspējas attēlu, kuram problemātiskās vietas tiks nošķirtas izmantojot lielāku pikseļu skaitu.



**5.11. att. Oriģināls un iespējamie punktu reģioni**

Attēlā 5.11. att. Oriģināls un iespējamie punktu reģioni, redzami visi atrastie punkti kā segmenti, kas atbilst aprakstītajam algoritmam. Tā kā netika apstrādāts konkrēts reģions tad redzama ir plates mala un arī tehnoloģiskie caurumi. Jaunizstrādājamai funkcijai ir paredzēta iespēja atpazīt tikai konkrētu reģionu no attēla, kas novērsīs šādus nepilnīgus rezultātus.

#### 5.4. Secinājumi

Attēla uzlabošanai lielu efektu dod kontrasta palielināšana, jo daudziem no testa attēliem pilnīgi pietika tikai ar kontrasta palielināšanu, lai izceltu celiņus un izlīdzinātu fonu. Turpretī pilnīgi neapstrādātiem attēliem sliekšņošana vairumā gadījumu tikai pazaudēja vajadzīgos datus. Bet ne visiem attēliem pietiek tikai ar kontrasta izmaiņām, lai panāktu vēlamo efektu. Jāizmanto ir arī tādas metodes kā, krāsu līmeņu mainīšana, krāsu piesātinājuma mainīšanas un citas, komplekta ar kontrasta mainīšanu.

Robežu noteikšana nepieciešama lodējumu vietu atrašanai attēlā, un tā nedod papildus ieguvumu celiņu noteikšanā. Turpretī celiņu izdalīšanā no attēla labus rezultātus uzrāda norādes apļu meklēšana un tiem atbilstošo centrālo pikseļu izvēlēšanās.

## 6. Rezultāti

Apskatītas sešas elektrisko shēmu veidošanas programmas. No kurām par vispiemērotāko integrēšanai ar jaunizveidojamu sistēmu, atzīta programma *LTspice IV*. Šī programma ir viegli uztverama un ar plašām iespējām. Turklāt tās atbalstītā failu formāta \*.asc failus iespējams viegli izveidot izmantojot kādu citu programmu.

Iespiedplašu shēmas atpazīšanas pētīšanai izvēlēti 20 dažādu shēmu attēli. No kuriem 11 izmantota laka sarkanā krāsā, 6 zaļa krāsā, 2 bez lakas pārklājuma, veido brūnu toni un 1 ar zilās krāsas laku. Kopējie rezultāti atspoguļoti tabulā 6.1. tabula Eksperimentu rezultātu vērtības. Eksperimentos iegūtos attēlus var apskatīt (1.pielikumā).

### 6.1. tabula Eksperimentu rezultātu vērtības

Rezultāts Darbība	Izmantojams turpmākai apstrādei (pozitīvs)	Nav viennozīmīgi nosakāms (neitrāls)	Nav izmantojams turpmākai apstrādei (negatīvs)
Kontrasta palielināšana	16	1	3
Sliekšņošana	13	5	2
Attēlu uzlabošana	17	3	0
Robežu noteikšana (Prewitt maska)	8	9	3
Robežu noteikšana (Sobel maska)	5	7	8
Robežu noteikšana (Roberts maska)	0	16	4
Robežu noteikšana (Laplacian maska)	0	6	14
Robežu noteikšana (Canny algoritms)	0	20	0
Norādes apļu meklēšana un centrālo pikseļu noteikšana	16	4	0

Tabulā atspoguļotas darbības un tām atbilstošo rezultātu skaits. Izmantojams turpmākai apstrādei (pozitīvs) nozīmē ka veicot konkrētu darbību ar attēlu, vēlamais mērķis tiek sasniegts uzreiz. Nav viennozīmīgi nosakāms (neitrāls) apzīmē attēlus, kuriem izvēlēta mērķa sasniegšana ir neprecīza. Tas nozīmē, ka nepieciešams veikt papildus citas darbības ar šo attēlu lai sasniegt cerēto mērķi. Nav izmantojams turpmākai apstrādei (negatīvs) norāda attēlu skaitu, kam izvēlēta darbība nav devusi vēlamus rezultātus un šo rezultātu sasniegšanai nepieciešamas citas darbības, viennozīmīgi izslēdzot konkrēto darbību kā pirmo un vienīgo.

Attēla uzlabošanai ļoti augstus rezultātus uzrādīja kontrasta palielināšana, jau pirmajā solī vairumam attēlu sasniedzot izvirzīto mērķi – izcelt elektriskos celiņus un izlīdzināt fonu. Arī sliekšņošanai ir vērā ņemami rezultāti, tomēr sliekšņošanas rezultātā palielinās celiņu platums, tādējādi palielinot iespēju, ka celiņi saplūds kopā. Izmantojot dažādas attēlu uzlabošanas metodes, izdevās panākt, ka tikai 3 attēliem ir neskaidrs rezultāts. Attēlu uzlabošanai apskatītas metodes, kuras var darbināt ar attiecīgajiem algoritmiem, bez lietotāja iejaukšanās.

Attēla sadalīšanai izmantojot vienas pašas klasiskās metodes, uz uzlabotiem attēliem, dod pārsvarā neviennozīmīgus rezultātus. Tomēr izmantojot robežu noteikšanai *Prewitt* un *Sobel* operatorus, iegūst attiecīgi 8 un 5 attēlus, kurus var izmantot autora piedāvātajai punktu noteikšanas metodei. Savukārt *Canny* robežu noteikšanas algoritms izdod korekti atpazītus attēlus, taču šim algoritmam nav vēl izdomāts pielietojums iespiedplašu atpazīšanā un tāpēc visi rezultāti uzskatāmi kā neitrāli. Norādes apļu meklēšana un centrālo pikseļu noteikšana ir jauna metode, kas aprakstīta ceļu noteikšanas risinājumu nodaļā un pateicoties ceļu līdzībai ar elektriskajiem celiņiem parāda augstus rezultātus, korekti atpazīstot celiņu viduslīnijas.

## 7. Secinājumi

Darbā aplūkotas vairākas metodes attēlu uzlabošanai, kuras pamatā balstījās uz vienkāršiem attēla pārveidojumiem. Kā arī vairākas metodes attēla sadalīšanai derīgos datos apgabalos, ko vēlāk var izmantot dažādi meklēšanas algoritmi. Apskatītajos pētījumos un metodēs nav redzams, ka uzstādīto uzdevumu, atpazīt elektrisko plašu shēmu, nevarētu izpildīt.

No apskatītajām elektrisko shēmu veidošanas programmām var izvēlēties dažādas programmas, kurām padot atpazītās shēmas tālākai apstrādei, tomēr pēc autora domām piemērotākā ir *LTsipce IV*.

Darbā apskatīti tikai daži ceļu atpazīšanas risinājumi, tomēr šai problēmai ir plašs piedāvāto risinājumu klāsts. Līdz ar to veicot papildus pētījumus šajā virzienā, iespējams tiks atrasta piemērotāka metode darbā aprakstītajai problēmai.

Tālākie pētījumi iespējami divos virzienos. Pirmajā virzienā ietilpst papildus jau veiktajiem darbiem, pētījumi par attēla uzlabošanu un derīgo datu atpazīšanu. Kuru problēmu var pastiprināt ar citiem ievaddatiem, kuriem ir paaugstināts trokšņu līmenis un pašam attēlam ir visādu veidu kropļojumi, piemēram, daļējs lakas iztrūkums, dažāds izgaismojums dažādās iespiedplates vietās un ļoti grūti nošķirami celiņi no fona. Otrajā virzienā ietilpst tālāki pētījumi apgabalu audzēšanas, grupēšanas, kā arī dažādiem meklēšanas algoritmiem, kas meklē datus no jau pieejamiem centrālo pikseļu attēliem.

## 8. Izmantotā literatūra un avoti:

1. Cadence Design Systems[tiešsaiste]. California: Cadence Design Systems, 2011 – [atsauce 02.05.2011.]. Pieejams: <http://www.cadence.com/us/pages/default.aspx>.
2. EDIF[tiešsaiste]. Wikimedia Foundation, 20.12.2010 – [atsauce 02.05.2011.]. Pieejams: <http://en.wikipedia.org/wiki/EDIF>
3. Linear Technology[tiešsaiste]. Linear Technology, 2010 – [atsauce 02.05.2011.]. Pieejams: <http://www.linear.com/designtools/software/>
4. Visionics[tiešsaiste]. Norlinvest Ltd – [atsauce 02.05.2011.]. Pieejams: <http://www.edwinxp.com/index.html>
5. ProfiCad[tiešsaiste]. ProfiCad – [atsauce 03.05.2011.]. Pieejams: <http://www.proficad.com/>
6. Hutson Systems [tiešsaiste]. New Zealand : Hutson Systems 05.10.2004 – [atsauce 03.05.2011.]. Pieejams: <http://www.hutson.co.nz/rimu.htm>
7. SmartDraw[tiešsaiste]. SmartDraw, 2011 – [atsauce 03.05.2011.]. Pieejams: <http://www.smartdraw.com/>
8. CarolineLacoste, XavierDescombes, JosianeZerubia Road network extraction in remote sensing by a Markov object process, 2004 – [atsauce 06.05.2011.]. Pieejams: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.58.9756&rep=rep1&type=pdf>
9. Yan Li, Ronald Briggs Automatic Extraction of Roads from High Resolution Aerial and Satellite Images with Heavy Noise. World Academy of Science, Engineering and Technology, 2009, vol. 54. p. 416 – 422. Pieejams: <http://www.waset.org/journals/waset/v54/v54-74.pdf>
10. Contrast (vision) [tiešsaiste]. Wikimedia Foundation, 03.05.2011 – [atsauce 09.05.2011.]. Pieejams: [http://en.wikipedia.org/wiki/Contrast\\_%28vision%29](http://en.wikipedia.org/wiki/Contrast_%28vision%29)
11. Rafael C. Gonzalez, Richard E. Woods Digital Image Processing Third Edition : Pentice Hall, 2008, p. 579-611
12. Canny edge detector[tiešsaiste]. Wikimedia Foundation, 28.03.2011 – [atsauce 12.05.2011.]. Pieejams: [http://en.wikipedia.org/wiki/Canny\\_edge\\_detector](http://en.wikipedia.org/wiki/Canny_edge_detector)
13. Canny edge detector (Canny filter) for image processing and computer vision[tiešsaiste]. 18.07.2008. – [atsauce 12.05.2011.]. Pieejams: <http://matlabserver.cs.rug.nl/cannyedgedetectionweb/web/index.html>

## **1. Pielikums. Kompaktdisks**

Darbam pievienots viens kompaktdisks, kas satur:

- Darba elektronisko kopiju,
- Izvēlētos 20 testa attēlus un šo attēlu eksperimentos iegūtos attēlus.

## Dokumentārā lapa

Bakalaura darbs „Iespiedplašu elektriskās shēmas atpazīšana” izstrādāts Latvijas  
Universitāte Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie  
informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai. Piekrītu sava darba  
publicēšanai internetā.

Autors: ..... / Jurgis Brauers /

Ar savu parakstu apliecinu, ka esmu lasījis augšminēto bakalaura darbu un atzīstu to  
par piemērotu aizstāvēšanai Latvijas Universitātes datorzinātņu bakalaura studiju  
programmas gala pārbaudījuma komisijas sēdē.

Darba vadītājs: ..... / Kārlis Freivalds /

Recenzents: .....

Darbs iesniegts Datorikas fakultātē .....

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā  
sistēmā.

Metodiķis(-e): .....

Darbs aizstāvēts bakalaura darbu gala pārbaudījuma komisijas sēdē

..... prot. Nr. ...., vērtējums .....

Komisijas sekretārs(-e): .....