



KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMATIKOS STUDIJŲ PROGRAMA

VAIDAS DZEDULIONIS  
PASIRINKTŲ OBJEKTŲ RETUŠAVIMO BEI ŠALINIMO  
VIDEO SRAUTE TYRIMAS  
Magistrinis darbas

Darbo vadovas:  
doc. A. Ostreika

KAUNAS, 2015



KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMATIKOS STUDIJŲ PROGRAMA

VAIDAS DZEDULIONIS  
PASIRINKTŲ OBJEKTŲ RETUŠAVIMO BEI ŠALINIMO  
VIDEO SRAUTE TYRIMAS

Magistrinis darbas

Darbo vadovas:  
doc. A. Ostreika  
2015 - 01 -

---

*(parašas)*

Recenzentas

2015 - -

---

*(parašas)*

Studentas  
V.Dzedulionis  
2015 - -

---

*(parašas)*

## **SANTRAUKA**

Šiame darbe pateikiamas apibendrintas logotipų šalinimo iš vaizdo įrašų algoritmas, detaliai aptariami algoritmo žingsniai ir galimas našumo didinimas, lyginant su komercinėmis programomis. Tyrime aprašyti logotipų šalinimo bandymai iš vaizdo įrašų, panaudojant kelias komercines programas kaip alternatyvas.

Atlikti dviejų diskretizavimo lygmenų tyrimai „Matlab“ aplinkoje, norint sužinoti, kurioje situacijoje kuris metodas tinkamesnis. Norint išgauti kuo kokybiškesnį vaizdą, kiekvienam lygmeniui buvo pritaikytas medianinis filtras. Kadangi lyginant medianinį ir gauso filtrą, medianinis filtras akivaizdžiai geriau nufiltravo atkurtus vaizdo taškus. Logotipų šalinimo iš vaizdo įrašų programoje ateityje reikalinga tobulinimo galimybė.

## **SUMMARY**

This work provides a generalized logos removal from video algorithm, algorithm steps are discussed in detail and the potential for improved efficiency comparing with commercial programs. The study describes the logo removals from the video record tests, through the use of videos in a few commercial programs as alternatives.

As well as to carry out two levels of sampling studies "Matlab" environment, in order to find out which method is appropriate in different situation. In order to get a better picture of the possible adaptation of the median filter for each level. For comparing median filter and the filter of Gaussian , median filter apparently better recovered the video spots. In the future in this program there are still possibility for improvement.

## TURINYS

Lentelių sąrašas.....	6
Paveikslų sąrašas.....	7
Įvadas.....	9
1. Analizė.....	11
1.1. Vaizdo suvokimas.....	11
1.2. Spalvos.....	11
1.2.1. RGB.....	12
1.2.2. CMY.....	12
2. Metodų analizė.....	13
2.1. Skirtingų platformų šalinimo būdai.....	13
2.1.1. Aperture programa.....	13
2.1.2. Aftereffect programa.....	18
2.2. „FinalCut Pro“ programos šalinimo galimybės.....	21
2.2.1. Triukšmo šalinimas „NoiseReduction V1.0“.....	21
2.2.2. Dėmių šalinimas su „Dirt Remover V1.0“ įrankiu.....	22
Automatinis juodų ir baltų dėmių pašalinimas iš vaizdo įrašo.....	22
2.2.3. „DropoutEliminator V1.0“.....	25
2.3. Gauso filtras.....	29
2.4. Branduolys.....	32
2.5. Sąsūka.....	32
2.6. Vaizdo apdorojimo slenkstis.....	34
2.7. Medianinis filtras.....	36
2.8. Artimiausio kaimyno metodas.....	36
3. Realizacija ir testavimas.....	39
3.1. Programos algoritmo kūrimas.....	39
3.1.1. Pirmasis etapas.....	39
3.1.2. Uždavinio optimizavimas.....	43
3.2. Skirtingų įrankių palyginimas.....	47
4. Rezultatų apibendrinimas ir išvados.....	52
5. Literatūros sąrašas.....	53
6. Priedai.....	54

## LENTELIŲ SĄRAŠAS

2.1 lentelė Grafo kelio svoriai.....	37
3.1 lentelė Kompiuterio charakteristikos .....	39

## PAVEIKSLŲ SARAŠAS

Pav. 1.1 Spinduliuojančios ir absorbuojančios spalvos .....	12
Pav. 2.1 Užstrigęs pikselis .....	13
Pav. 2.2 Vaizdo įrašas, kuriame yra dulkių pėdsakai .....	14
Pav. 2.3 Vaizdo atkarpos įkėlimas.....	14
Pav. 2.4Šalinamos vaide esančios dulkės .....	15
Pav. 2.5Dulkių fiksavimo įrankis .....	15
Pav. 2.6 Vaizdo kadru peržiūra.....	16
Pav. 2.7 Vaizdo kadru nustatymas.....	17
Pav. 2.8 Įkeliamas vaizdobykla į “Adobe aftereffect” programą.....	18
Pav. 2.9 Įkeltas vaizdo įrašas .....	18
Pav. 2.10 Vaizde užstrigę vaizdo taškai .....	19
Pav. 2.11 Vaizdo įrašas.....	19
Pav. 2.12 Parenkamas įrankis .....	20
Pav. 2.13 Taškų parinkimas .....	20
Pav. 2.14 Prieš ir po panaudojus CC SimpleWireRemoval efektą.....	20
Pav. 2.15 Vienmatis gauso pasiskirstymas su vidurkiu 0 ir standartiniu nuokrypiu $\sigma = 1$ ..	30
Pav. 2.16 Dvimatis gauso pasiskirstymas su vidurkiu (0,0) ir standartiniu nuokrypiu $\sigma = 1$ .....	30
<b>Pav. 2.17</b> Diskrečioji aproksimacija Gauso funkcija su standatiniu nuokrypiu $\sigma = 1.0$ .....	31
Pav. 2.18 Vienas iš 1-D sąsukos metodų, naudojamų apskaičiuojant visą branduolį, veikiantis greičiau nei priestai esantis metodas. ....	32
Pav. 2.19 Sąsukos branduolys vidurkio filtrui su 3x3 kaimynais. ....	32
Pav. 2.20 Mažas paveikslukas (kairėje) ir branduolys (dešinėje) iliustruoja sąsuką. Reikšmės kiekvieno grotelių kvadrato viduje yra panaudotos, kad identifikuotų kiekviena kvadrata.....	33
Pav. 2.21 Vaizdo apdorojimas slenksčio metodu. ....	34
Pav. 2.22 Kaip matoma centrinio taško reikšmė 150 labai skiriasi nuo aplink esančių , ir ji yra pakeičiama medianos reikšme - 124. Čia naudojama 3x3 kaimynų kvadratas. ....	36
Pav. 2.23 Grafo pavizdys .....	37
Pav.3.1 Algoritmo sudarymo schema .....	40
Pav.3.2 Pirmojo testo pradinis variantas.....	41
Pav.3.3 Pirmojo testo gautasis variantas.....	41
Pav.3.4 Antrojo testo pradinis variantas. ....	42
Pav.3.5 Atnrojo testo gautasis variantas. ....	42
Pav.3.6 Trečiojo testo pradinis variantas. ....	43
Pav.3.7 Trečiojo testo galutinis variantas. ....	43
Pav.3.8 Galutinio algoritmo sudarymo schema. ....	44
Pav.3.9 Galutinio įrankio pirmo testo pateikiamas vaizdo įrašas. ....	45
Pav.3.10 Galutinio įrankio pirmojo testo gautas vaizdo įrašo rezultas.....	45
Pav.3.11 Galutinio įrankio antro testo pateikiamas vaizdo įrašas. ....	46
Pav.3.12 Galutinio įrankio antro testo gautas vaizdo įrašo rezultas. ....	46
Pav.3.13 Galutinio įrankio pirmo testo pateikiamas vaizdo įrašas. ....	47
Pav.3.14 Galutinio įrankio pirmojo testo gautas vaizdo įrašo rezultas.....	47
Pav.3.15 Televizijos logotipo šalinimas iš pirmo vaizdo įrašo.....	49
Pav.3.16 Televizijos logotipo šalinimas iš antrojo vaizdo įrašo. ....	49
Pav.3.17 Televizijos logotipo šalinimas iš trečiojo vaizdo įrašo. ....	50
Pav.3.18 Amžiaus cenzo šalinimas iš pirmojo vaizdo įrašo. ....	50
Pav.3.19 Amžiaus cenzo šalinimas iš antrojo vaizdo įrašo. ....	50
Pav.3.20 Amžiaus cenzo šalinimas iš trečiojo vaizdo įrašo. ....	51

## TERMINŲ IR SANTRUMPŲ ŽODYNAS

<b>RGB</b>	Spalvų modelis, kuris susidaro iš trijų spalvos palečių: raudonos, žalios ir mėlynos.
<b>CMY</b>	(cyan, magenta, yellow) ciano mėlis, purpurinė ir geltona. Naudojama išfiltruoti ir atspindėti tam tikra spektro dalimi.
<b>YIQ</b>	Žmogaus akis yra labiau jautresnė spalvos ryškumui, nei spalvai. Dažniausiai naudojamas spalvotuose televizoriuose.
<b>HSI</b>	H– „gryna“ šviesos spalva (angl. hue). S– spalvos sodrumas (angl. saturation) nusako, kiek baltos spalvos yra sumaišyta su „gryna“ spalva. I – spalvos ryškumas (angl. intensity).
<b>XYZ</b>	Pagal CIE standartą sudarytas teorinis XYZ modelis, kurio spalvos yra raudona, žalia, mėlyna.
<b>DVD</b>	yra optinis diskas informacijai skaitmenine forma įrašyti, saugoti ir platinti.
<b>TIFF</b>	(angl. <i>Tagged Image File Format</i> ) – žymėtasis atvaizdų failų formatas yra gana paplitęs, tačiau dėl kitos priežasties nei JPG.



## IVADAS

**Darbo tikslas** – atlikti tyrimą, kurio pagalba išsiaiškinti vaizdo sraute esančių titrų, subtitrų, logotipų, dulkių, subraižymų, ištepimų retušavimo bei šalinimo galimybes skirtingose aplinkose.

### **Darbo uždaviniai:**

- Pasirinkus vaizdo atkarpą, ar keletą vaizdų, įrankio pagalba panaikinti esamus televizijos logotipus;
- Kuriamam įrankiui pritaikyti du diskretivavimo lygmenis;
- Iširti kelis skirtingus komercinius pašalinimo įrankius ir palyginti su kuriamąja programa;

Skaitmeninis vaizdo apdorojimas yra daugelio disciplinų arba tarpdisciplininis tyrimų objektas, kuris remiasi fizikos, matematikos, inžinerijos ir kompiuterių mokslais. Jis plačiai taikomas mokslo ir inžinerijos, branduolio fizikos ir psichologijos, orų stebėjimo bei prognozės ir socialiniuose moksluose, kosminės erdvės įsisavinime ir medicinoje. Vaizdas paprastai turi įvairios informacijos apie jame esančius objektus – tai objektų tarpusavio padėtis, dydis ar atstumas tarp jų, objektų spalvos ir tekstūros. Todėl skaitmeniniai paveikslai užima didelę dalį atminties kompiuteryje. Dažnai, uždaviniui išspręsti, pakanka tik dalies šios informacijos, tuo pačiu taip yra sumažinama apdorojamų duomenų apimtis ir padidinama apdorojimo sparta. Viena iš tokių vaizdo informacijos dalių yra kontūras.

Atsiradus kompiuteriams, atsirado ir nelinijinio vaizdo redagavimo idėja. Iki tol pasaulis nežinojo tik tiesines video redagavimo sistemas. Palyginimui galima sakyti, kad skirtumas tarp abiejų redagavimo tipų atitinka skirtumą tarp garso kompaktinio disko ir garso kasetės. Linijinio redagavimo atveju vaizdas yra perrašomas tiesiai iš vienos juostos į kitą. Norint pasiekti tam tikrą juostos fragmentą, būtina persukti dalį kino ar vaizdo juostos. Būtent dėl nuolatinio juostos persukinėjimo linijinis redagavimas yra nepatogus ir užima daug laiko. Nelinijinio redagavimo atveju vaizdajuostės turinys perrašomas į kompiuterio kietąjį diską skaitmeniniu pavidalu. Ir tada visa vaizdo medžiaga matoma kompiuterio ekrane: galima akimirksniu karpyti, stumdyti, keisti vietomis vaizdo fragmentus neperrašinėjant ir nepersukinėjant vaizdo kasetės. Galutinis redagavimo rezultatas įrašomas arba atgal į juostą, arba į skaitmeninę laikmeną: CD, DVD. Kompiuteris gali būti panaudotas abiejų tipų vaizdui redaguoti. Linijinio redagavimo atveju AK tiesiog atlieka tarpinę grandį tarp vaizdo šaltinio ir įrašymo įrenginio. Profesionaliose linijinio montažo sistemose kompiuteris tarnauja ne tik kaip vaizdo stebėjimo įrankis, bet taip pat valdo atkuriančius ir įrašančius vaizdo magnetofonus arba kameras. Kai kuriais atvejais, pavyzdžiui, vadinamosiose hibridinėse nelinijinio montažo stotyse, kompiuteris skaičiuoja dar ir trimatę grafiką. Bet visos kompiuterio galimybės iš tiesų panaudojamos tik nelinijinio redagavimo atveju, kai pagrindiniai

medžiagos apdorojimo darbai atliekami kompiuteryje naudojant kietąjį diską, centrinį procesorių, vaizdo plokštę ir kitas kompiuterio dalis.

Šiais laikais visiems vaizdo redagavimo veiksams užtenka turėti asmeninį personalinį kompiuterį ir vaizdo redagavimo studiją namuose, kadangi galima vaizdą iškirpti, pasukti, panaudoti vaizdo spalvų keitimą.

## 1. ANALIZĖ

### 1.1. Vaizdo suvokimas

Skaitmeninis vaizdas – tai skaitmenizuotas vietovės ar aplinkos reginys arba sintetiniu (programiniu) būdu gautas duomenų masyvas. Vaizdas gali būti statinis, dar vadinamas paveikslu, arba dinaminis – judantis. Vaizdų apdorojimas arba skaitmeninis vaizdų apdorojimas – tai vaizdų apdorojimas, kai yra naudojamos kompiuterinės programos ar algoritmai. Aplinkoje objektai gali būti dviejų tipų – spinduliuojantys ir atspindintys. Spinduliuojantiems ir atspindintiems vaizdams yra taikomi skirtingi parametrai. Žmogaus matomą spinduliuojamą vaizdą nulemia du parametrai: atspalvis ir intensyvumas. Spalvą nulemia bangos ilgis, apšviestumą – šviesos kiekis.

Kaip mes suvokiame atspindintį paveikslą, apibrėžia trys parametrai: atspalvis, sotis arba sodrumas ir apšviestumas. Atspalviais vadiname parametrus, kai kalbama apie gryną spalvą, pavyzdžiui, žalią, geltoną. Tai priklauso nuo to, kokie bangos ilgiai pasiekia stebėtoją: 430 – 480nm yra mėlyna; 500 - 550 nm - žalia, 570 – 600 nm - geltona ir daugiau nei 610 nm - raudona. Balta, pilka ir juoda nėra atspalviai – tai kelių spalvų proporcingas mišinys. Spalvos sotis (saturacija) nurodo spalvos grynumą ir kiek ji artima baltai spalvai.

Didėjant sočiui spalva ryškėja, mažėjant – blanksta, kol tampa balta. „Nesočioji" spalva yra išblukusi, „sočioji" – ryški ir kontrastinga, pavyzdžiui, raudona, yra „sočioji" spalva, rausva - „nesočioji". Gryna spalva turi 100 proc. saturacijos, balta spalva turi 0 proc. saturacijos. Baltos ir „sočiosios" spalvos diapazonas gali būti nuo 0 iki 100 proc.

Apšviestumą galima laikyti saturacijos atvirkštiniu parametru, jis parodo, kiek spalva yra apšviesta, t.y. jos intensyvumą, lyginant su juoda spalva; jei apšviestumas 0 proc., gaunama juoda spalva, jei 100 proc., gaunama balta spalva. Keičiant apšviestumą, spalvos kitimas vyksta perpilkumo (nuo juodos iki baltos) skalę. Kontrastas yra santykinis dydis, kuris parodo paveikslo tamsiausio ir šviesiausio plotų santykį.

Didelio kontrastingumo paveikslai turi didelius tamsius ir didelius šviesius plotus. Didėjant kontrastui, stebėtojui atrodo, kad didėja paveikslo detalumas (detalių kiekis), tačiau paveikslo informacijos kiekis išlieka toks pat. Taip yra todėl, kad žmogaus akis yra jautri šviesos kontrastui ir mažiau jautri absoliučiam šviesos intensyvumui, kai akis neturi su kuo palyginti šviesos intensyvumo dydžio.

### 1.2. Spalvos

Spalvų modeliai arba paletė aprašo spalvas ir jų tarpusavio ryšius. Praktikoje pagal poreikį yra naudojamos kelios spalvų paletės. Pavyzdžiui, spaustuvės naudoja CMY spalvų modelį: monitoriams, televizoriams ir pan. Naudojamas RGB spalvų modeliai. Sistemos, kurios naudoja

atspalvį, saturaciją ir apšviestumą, dirba su HSI spalvų modeliu. Yra sukurta daug įvairių spalvų modelių, tačiau dauguma yra sudaryta iš trijų dedamųjų: YIQ, XYZ, YUV, UVW ir pan. Toliau nagrinėsime keturis, dažniausiai naudojamus, spalvų modelius: RGB, CMY, HSI ir YCbCr. Dažniausiai visi spalvų modeliai yra suderinami tarpusavyje, t.y. galima pereiti nuo vieno prie kito, atliekant matematinius ar trigonometrinius perskaičiavimus.

### 1.2.1. RGB

RGB spalvų paletė sudaryta iš trijų pirminių spalvų: R - raudonos, G - žalios ir B - mėlynos. Maišant šias pirmines spalvas yra gaunama bet kokia kita spalva. Dažniausiai RGB spalvos vaizduojamos kaip kubo kraštinės. Kai visos dedamosios yra lygios 0, gaunama juoda spalva, o kai maksimalios reikšmės – balta. Maišant visų spalvų po lygiai, gaunami pilki tonai.

RGB yra patogi technikoje – daugelis vaizdo įtaisų veikia šios paletės pagrindu. Tačiau skalė nėra universali – ji netinka vaizdams apdoroti. Priežastis – R, G, B dedamosios tarpusavyje labai koreliuoja. O tai sukelia problemų naudojant vaizdo apdorojimo ir analizės algoritmus. Pavyzdžiui, histogramos tankiui išlyginti geriau naudoti HSI skalę, kurioje yra spalvų intensyvumai. Daugeliu atvejų vaizdams apdoroti yra reikalingi ne spalvoti, o juodos ir baltos spalvų su pilkais atspalviais vaizdai, toliau dėl paprastumo tokį vaizdą vadinsime tiesiog nespalvotu vaizdu, kurio reikia nesumašyti su vaizdais, kuriuose yra tik juoda arba balta spalva.

### 1.2.2. CMY

CMY spalvų modeliu sudarytas iš žydros, raudonai rausvos ir geltonos spalvų. Jei RGB spalvos yra laikomos adityvinės, t.y. spinduliuojančios pirminės spalvos, tai CMY yra substraktinės, t.y. absorbuojančios pirminės spalvos (Pav. 1.1).



**Pav. 1.1** Spinduliuojančios ir absorbuojančios spalvos

## 2. METODŲ ANALIZĖ

Skaitmeninės kameros trūkumai filmavimo metu yra dulkės ant objektyvo ekrano ar atsirandantys nefunkcionuojantys pikseliai, kitaip dar vadinami karštaisiais arba mirusiais pikseliais (Pav.2.1.).



**Pav. 2.1** Užstrigęs pikselis

Vienas iš šios problemos sprendimo būdų yra įsigyti autorizuotą, tikrai kokybišką techninę įrangą, tačiau net ir pati kokybiškiausia įranga gali turėti trūkumų. Jeigu filmuojant brangia skaitmenine kamera turinčia daugiau nei 21 milijoną taškų, vistiek atsiras bent keletas taškų, kurie vistiek bus nekorektiški. Fotografuojant pikselių šalinimas yra labai paprastas. Profesionalios nuotraukų retušavimo programos, kaip Lightroom ar Aperture dažnai automatiškai aptinka ir sutvarko užstrigusius pikselius, kai tik nuotrauka įkeliama į programą. Su vaizdo medžiaga yra kurkas sudėtingiau. Kol kas nėra tokios programos, kuri pati surastų ir automatiškai pašalintų vaizdo įrašė pastebimus pikselių gedimus. Ir nors vienas, ar keli nefunkcionuojantys pikseliai iš 21 milijono nėra labai pastebimas reiškinys, tačiau labiau įgudusi akis pastebės.

### 2.1. Skirtingų platformų šalinimo būdai

#### 2.1.1. Aperture programa

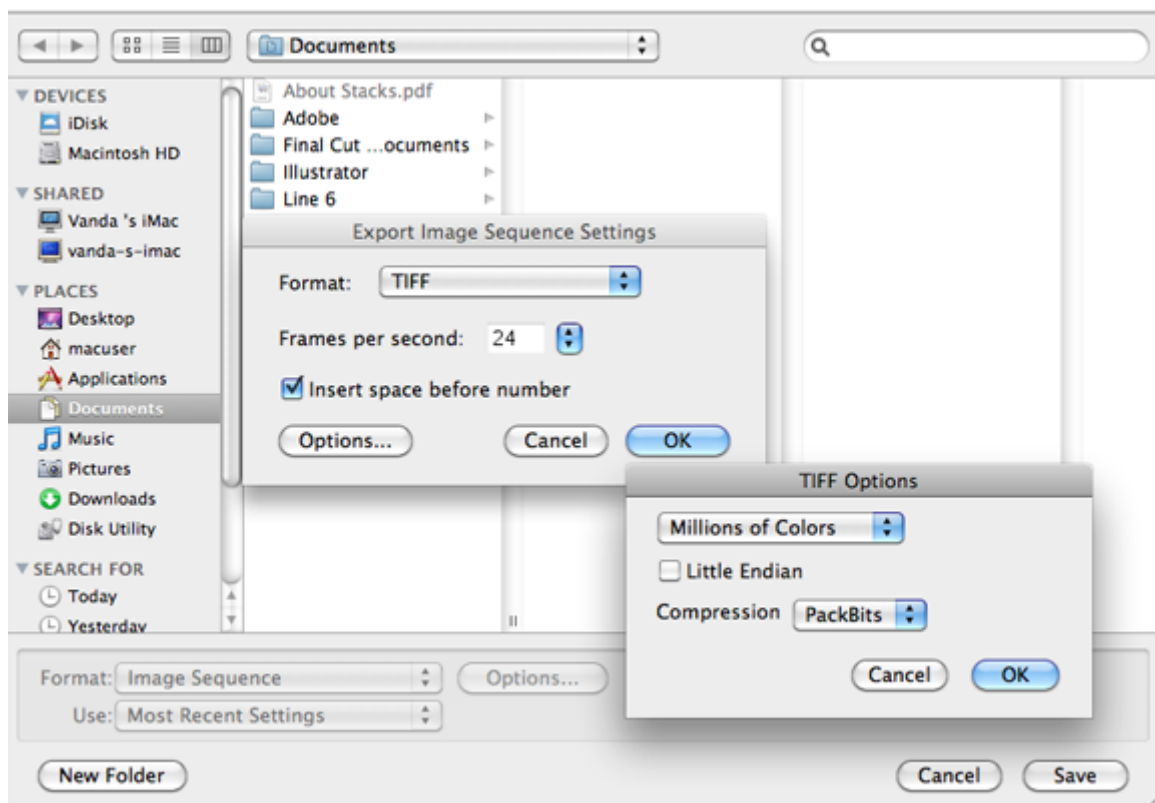
Šalinimas „Aperture“ programa. Šios programos pagalba galima pašalinti dulkes ir kitus nešvarumus nuo vaizdo įrašo. Dulkių pašalinimas iš foto medžiagos yra nesunkiai atliekamas procesas, tačiau judančiuose paveikslėliuose (vaizdo medžiagoje) nevisada pavyksta pašalinti, arba tai užima daug laiko ir pastangų.

Pavyzdžiui, bus naudojama vaizdo klipo ištrauka, kurioje aiškiai matomi dulkių pėdsakai(Pav.2.2).



**Pav. 2.2** Vaizdo įrašas, kuriame yra dulkių pėdsakai

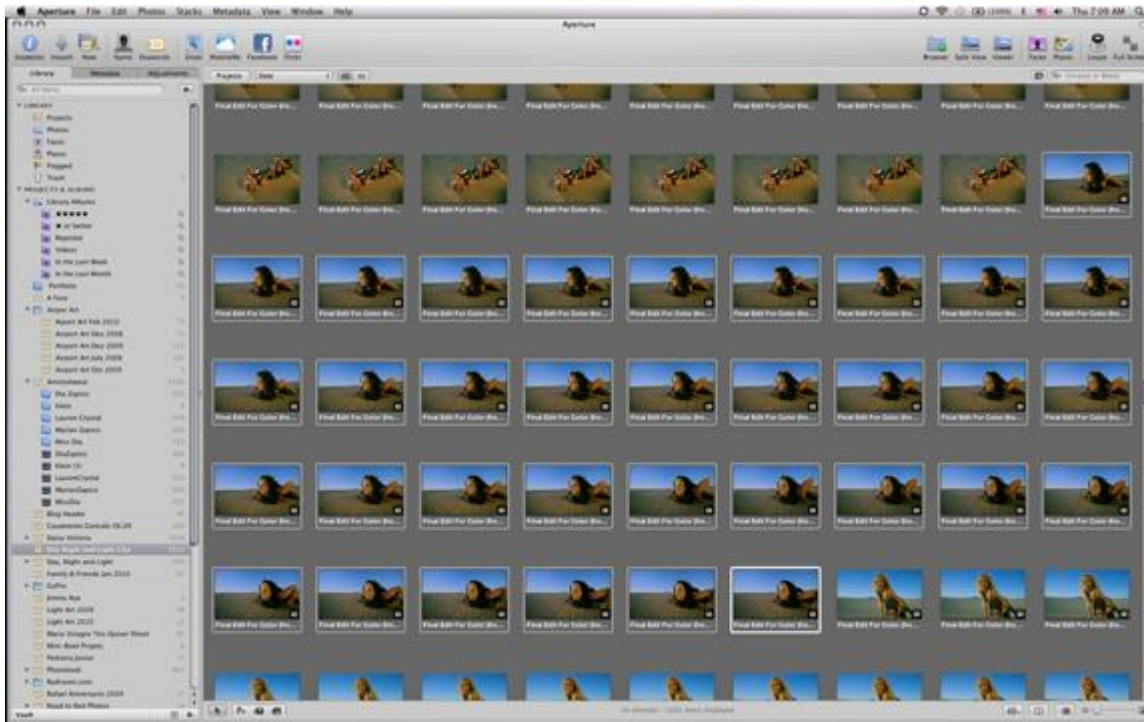
„Aperture“ turi kelias retušavimo galimybes, kurios nesunkiai susitvarko su dulkių pėdsakais vaizdo įrašė. Įkėlimui į programą buvo reikalingas TIFF formato vaizdo įrašas, kadangi įrašas buvo trumpas (vaizdo klipo atkarpa), įkėlimas neužėmė daug laiko ir pastangų (Pav.2.3).



**Pav. 2.3** Vaizdo atkarpos įkėlimas

Dulkių šalinimui buvo naudojama „Spot&Patch“ įrankis. Viskas iš tikrųjų vyksta taip pat, lyg būtų šalinamos dulkės nuo nuotraukos. Viskas sužymima ir leidžiama programai šalinti dulkių žymes iš įrašo atkarpos (Pav.2.4).



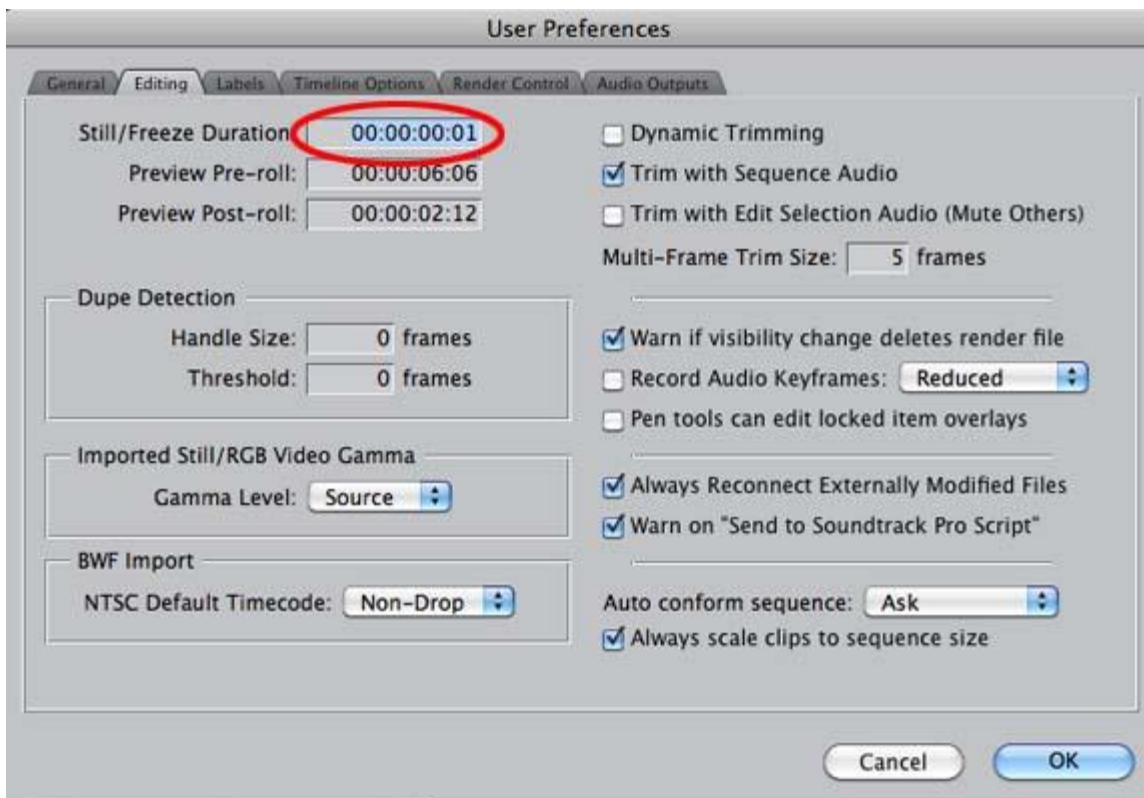


**Pav. 2.6** Vaizdo kadru peržiūra

„Spot&Patch“ įrankis nors ir labai geras, tačiau irgi yra ribotas. Puikiai veikia prie vienodo fono, tačiau kai yra daugiau detalių, daugiau judesio, iškart atsiranda įrankio spragos ir nestabilumai. Taip pat naudojant šį metodą galima mėginti keisti tos vietos spindulį, minkštumą ir nepermatomumą, kas turėtų padėti sumažinti problemas, tačiau kartais išvis to nedaryti ir visai nenaudoti šio metodo. Taip pat dažnai, kai dėmė pereina per daugelį detalių, ji beveik nepastebima ir tiesiog reikia nuspręsti ar bandyti šalinti šią problemą, ar ne.

Svarbus dalykas norint eksportuoti vaizdo įrašą yra tas, kad Aperture neturi 10 bit galimybės, todėl būtina naudoti 16 bit, kad būtų neprarastos vaizdo įrašo spalvų informacijos. Taip pat svarbu prieš importuojant bet kokį vaizdo įrašą, nueiti į nustatymus ir „Editing“ skiltyje nusistatyti „Still/FreezeDuration“ į 1 kadra, kitaip gali būtų, kad nepavyks tiksliai perteikti reikalingo kadro (Pav.2.7)



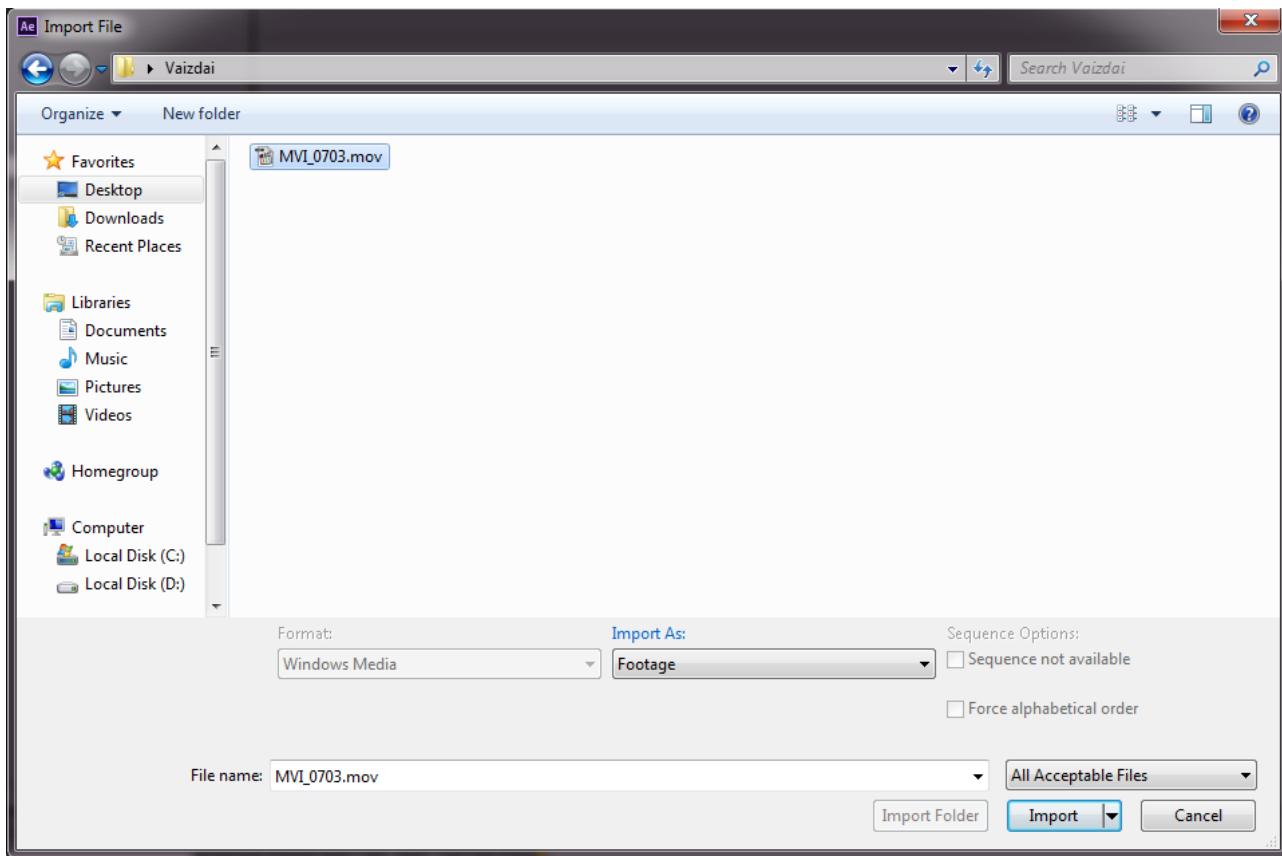


**Pav. 2.7** Vaizdo kadru nustatymas

Tai žinoma tik maža dalis šio metodo, tačiau ji tikrai reikalinga ir gali padėti išspręsti ne vieną su vaizdo medžiagomis iškilusią problemą.

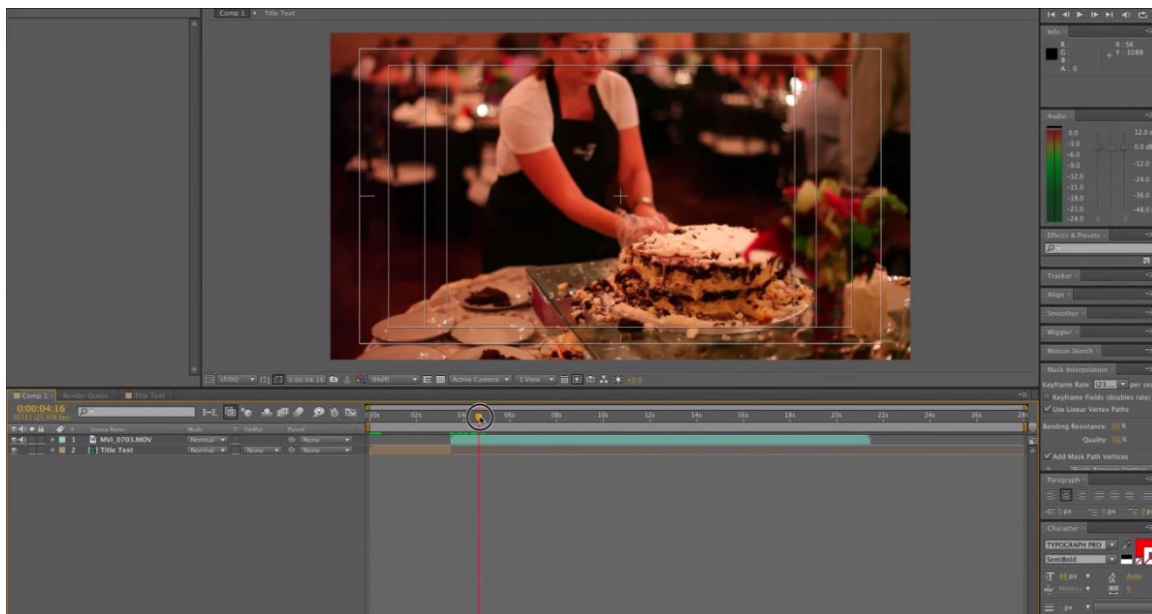
## 2.1.2. Aftereffect programa

Pirmiausiai įkeliame vaizdo failą, kuriame yra užstrigusių pikselių.(Pav.2.8).



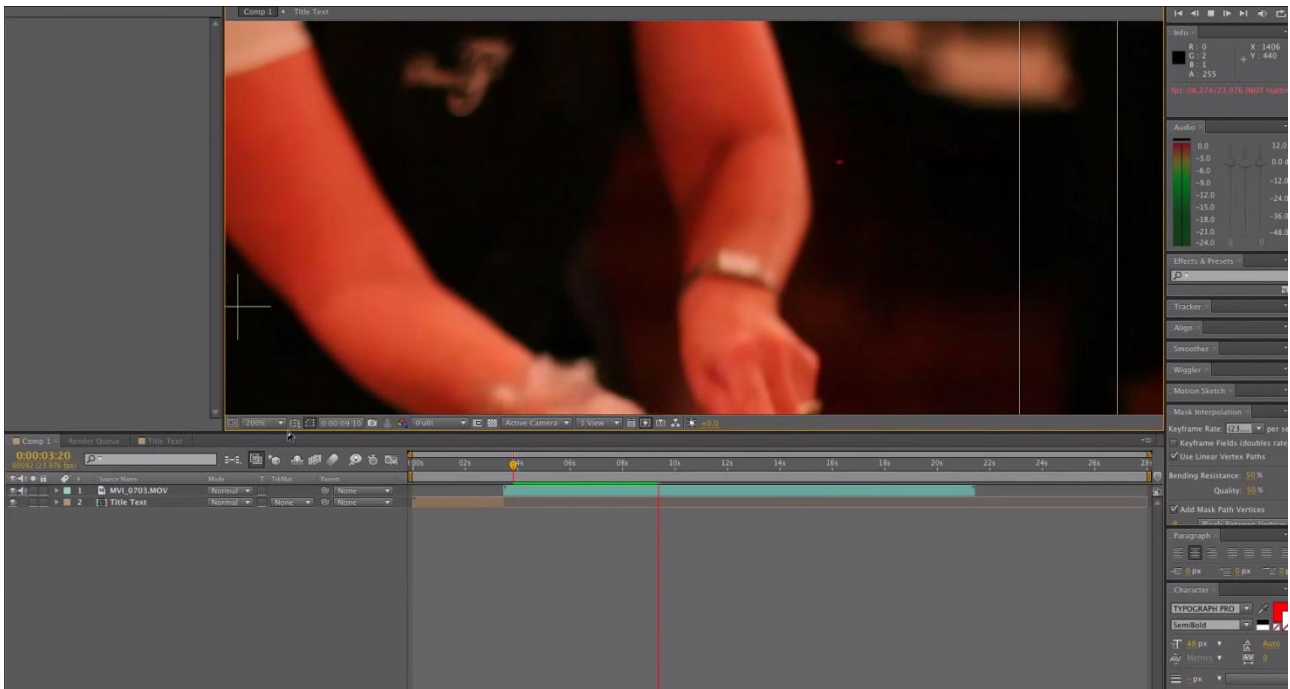
**Pav. 2.8** Įkeliama vaizdobykla į “Adobe aftereffect” programą

Įkėlę vaizdo bylą, matome iš pirmo žvilgsnio taisyklingai atrodantį vaizdo įrašą (Pav.2.9).Vaizdo įrašas filmuotas skaitmenine vaizdo kamera „Canon 5d“, ši kamera yra profesionali, bet ir tokiose vaizdo kamerose gali būti užstrigusių pikselių.



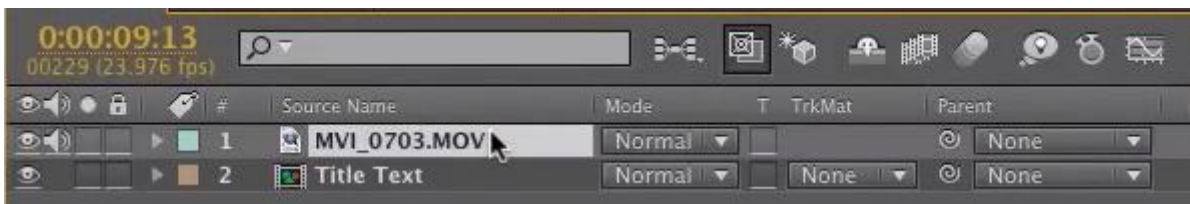
**Pav. 2.9** Įkeltas vaizdo įrašas

Priartinus vaizdą ir apžiūrėjus visą esantį vaizdo įrašo lauką, pastebime, kad vienoje vietoje yra užstrigusių vaizdo taškų (Pav. 2.10). Naudojant net ir geriausias vaizdo kameras, negalime būti tikri, kad neatsiras nekorektiškų vaizdo taškų.



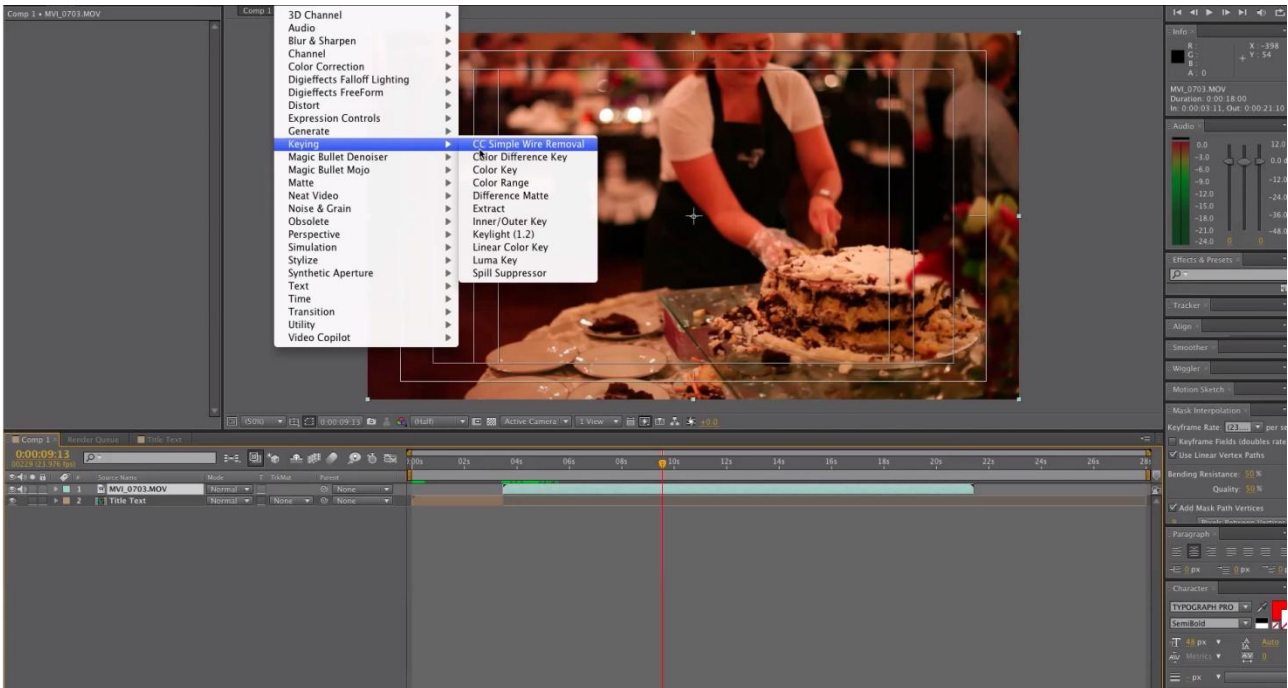
**Pav. 2.10** Vaizde užstrigę vaizdo taškai

Norint panaikinti užstrigusius vaizdo taškus, pirmiausiai pasižymime vaizdo įrašą (Pav. 2.11).



**Pav. 2.11** Vaizdo įrašas

Pažymėjus vaizdo įrašą spaudžiame Effect ->Keying-> CC SimpleWireRemoval (Pav.2.12).



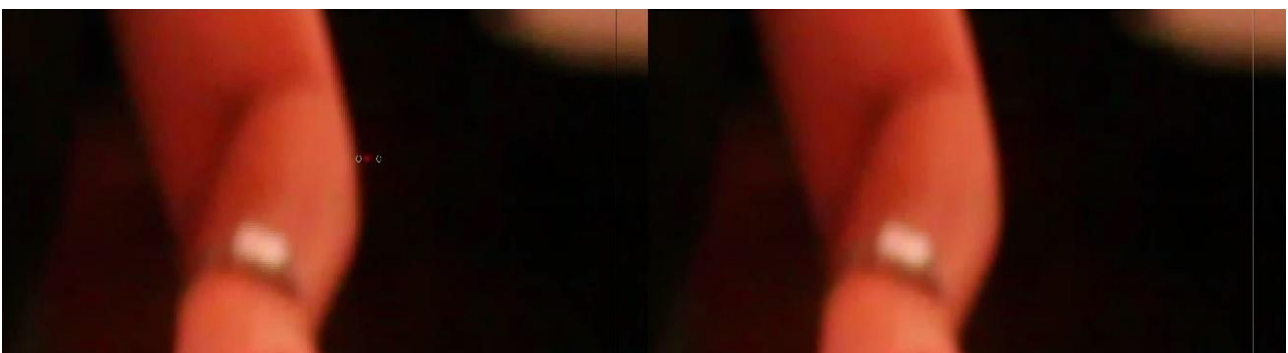
**Pav. 2.12** Parenkamas įrankis

Norint panaikinti užstrigusius taškus reikia pasirinkti du taškus A ir B (Pav. 2.13).



**Pav. 2.13** Taškų parinkimas

Pasirinkus du priešingus apie pikselį išsidėsčiusius taškus padidiname storį (ang. Thickness). Šiuo atveju iš 2.00 pakeičiame į 5.00. Pakeitus šį parametą pradingsta užstrigę vaizdo taškai (Pav. 2.14).



**Pav. 2.14** Prieš ir po panaudojus CC SimpleWireRemoval efekta

Išsaugome ir iškeliame vaizdo failą. Panaudojus šį efekta panaikinome esantį užstrigusį vaizdo tašką iš esančio vaizdo įrašo.

## 2.2. „FinalCut Pro“ programos šalinimo galimybės

- Pašalinti triukšmą vaizdo įrašė.
- Nuimti purvą ir įbrėžimus.
- Mirusių pikselių pašalinimas.

### 2.2.1. Triukšmo šalinimas „NoiseReduction V1.0“

Tai paprasta naudotojui programa. Nors palyginus tai labai galinga programa, tačiau ji turi 3 svarbius veikimui parametrus.

„NoiseReduction“ nuskaito tam tikrą laiko atkarpą, palygina visus tos atkarpos rėmus ir vertina matematiškai, kokia dalis video yra realus turinys ir kokia dalis yra triukšmas, kuris turi būti eliminuojamas. Tokiu būdu galima pasiekti daug geresnių rezultatų, nei bet kuria kita, panašaus tipo, triukšmo šalinimo programa.

	(C) 2006 by chv-electronics, The
	[ General Settings ]
Noise (to eliminate)	← [slidmušelis] → 20
Used Timespan	← [slidmušelis] → 34
	[ Scene-change ]
Sensitivity	← [slidmušelis] → 120
Calc and show	<input type="checkbox"/>
	www.chv-plugins.com, Christoph
	Noise Reduction V1.0

Parametrai:

#### **Triukšmas (kurį pašalinti)**

Čia galima nustatyti triukšmo kiekį, kurį reikia ištrinti. Pastaba: nereikia nustatyti per didelio jo kiekio, nes gali atsitikti taip, kad programa ištrins ir reikalingus vaizdo įrašo komponentus.

Pavyzdžiai:



Pavyzdžiui, pasirenkame labai prastos kokybės įrašo kadra, kuriame yra labai daug triukšmo.



Pabandydami pasirenkam ties triukšmo šalinimu 15. Kaip matosi paveikslėlyje triukšmo beveik neliko, o visi kadre matomi objektai tebėra ryškūs.



Triukšmo šalinimas nustatomas ties 50. Taigi, iš kadro matome, kad ne tik triukšmas, bet ir kadro objektai išsitrynė, teliko blankus fonas, su šiek tiek matomais objektais. Tai puikus pavyzdys, kaip neverta persistengti su parametru reikšmėmis.

**„UsedTimespan“ parametras**

Šis parametras nusako programai, kiek įrašo trukmės paimti triukšmo šalinimui. Optimalus parametro pasirinkimas 10- 40, tai leidžia optimaliai veikti programai, be jokių išorinių vaizdo įrašų pažeidimų.

### **Jautrumo (ang. Sensitivity) parametras**

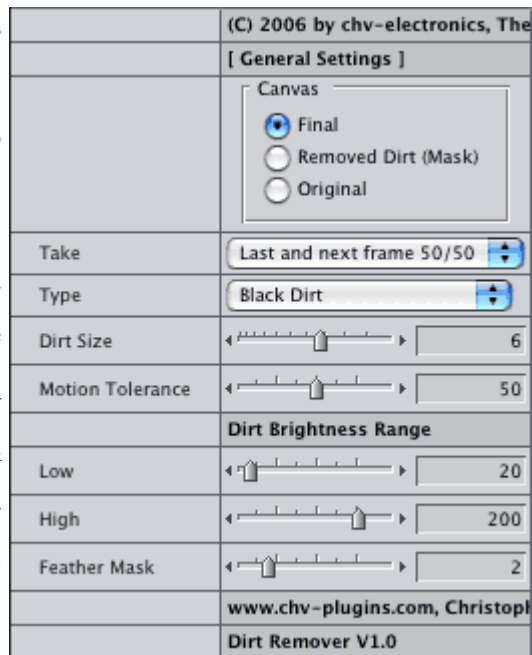
Tai labai svarbus programos parametras. Triukšmo mažinimo veikimo principas yra toks, kad jis apima aplinkinius rėmus, į kuriuos atsižvelgiama norint įvertinti esamą vaizdo įrašo triukšmą. Jei dabartinis kadras patiria pokyčius (t.y. dėl supjaustymo, nukirpimo), tai nėra naudinga triukšmo mažinimo įvertinimui. Todėl triukšmo mažinimas apskaičiuoja reikšmę, kuri apibūdina ryšį tarp atskirų kadrų skirtumo. Kuo didesnė vertė, tuo labiau tikėtina, kad kadras pasikeitė. Tai ir yra jautrumo parametras, kuris nustato ribą „NoiseReduction“ ir pasako didžiausią leistiną kiekį tarp kadrų skirtumo.

### **2.2.2. Dėmių šalinimas su „Dirt Remover V1.0“ įrankiu**

Automatinis juodų ir baltų dėmių pašalinimas iš vaizdo įrašo.

Programa pašalina tamsios ir šviesios spalvos dėmes iš vaizdo kadru naudodama greta esančių rėmų atspalvius. Galima pasirinkti tarp tamsios, šviesios spalvų pašalinimo ar abiejų iškarto.

„Dirt Remover“ stebi aplinkinius rėmus, kad išsiaiškintų, ar nustatyta vieta tikrai yra dėmė, o ne judantis objektas. Jei dėmė identifikuota, ji bus pakeista prieš tai buvusio, ar einančio po jo kadru, arba abiejų kadru kombinacija. Taip keliaudama per kadrus programa sukuria tam tikrą dėmių sąrašą, kurį galima peržiūrėti programoje.

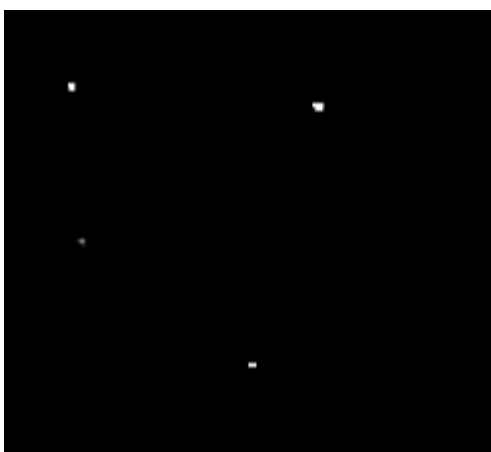


### Programa pateikia 3 pagrindines galimybes:



#### „Original“:

Programa parodo nepakeistą vaizdo medžiagą, kuri leidžia įsitikinti, kad nustatymai tikrai geri, kurie pašalins tik purvą ir dėmes, o ne reikalingus objektus ir detales kadre.



#### „Removed Dirt (Mask)“:

Čia parodomas vietos, paimtos iš vaizdo medžiagos kadro (kitą tariant, parodo purvo ir dėmių vietas vaizdo kadre). Tai tarsi programos pradžia prieš keičiant dėmių vietas kitų kadru komponentais ir spalvomis.



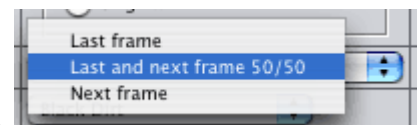
### „Final“:

Programa parodo sutvarkytą ir subalansuotą vaizdo medžiagos kadrą. Nereikia rankiniu būdu keisti nustatymų, programa pati tai atlieka ir sutvarko visą vaizdo įrašą nuo šių ir panašių dėmių ir purvo likučių.

### Trumpai apie kitus parametrus:

#### „Take“

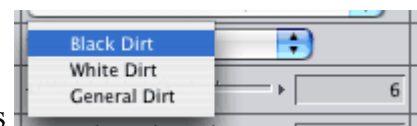
Šis parametras nurodo programai, kurią kadro dalį naudoti, kai aptinkama dėmė ir norima ją pašalinti. Galimi variantai: prieš tai buvęs kadras, po jo einantis, ar kadrų suliejimas.



Pasirinkimas „Last and next frame 50/50“ yra abiejų kadrų suliejimas. Tai pats geriausias ir protingiausias dėmių šalinimo būdas, užtikrinantis, kad dėmė tikrai pasišalins naudojant vaizdo įrašo komponentus. Ypač svarbu jei kadruose keičiasi šviesos intensyvumas, tai ši funkcija parenka optimalų šviesos kiekį toje vietoje, kur buvo dėmė ar purvo likučiai.

#### „Type“

Norimas valymo tipas, nurodoma kokios spalvos dėmes siekiame išvalyti: juodos, baltos, ar abiejų spalvų.



#### „Dirt Size“

Šis nustatymas nurodo maksimalų purvo kiekį kadruose, kurį reikia pašalinti. Tai pats svarbiausias programos parametras. Svarbu, kad dėmės, kurios bus mažesnės už nurodytą dydį, bus pašalintos, o visos kitos išliks.

#### „MotionTolerance“

Norint tiksliai įvertinti dėmių ir purvo vietas vaizdo kadruose, programa turi įdėmiai nuskaityti vaizdo įrašą. Jai labai paprasta nustatyti tokias vietas, jei vaizdas mažai keičiasi ir jame nėra didelio judėjimo. Tačiau, jei vaizdai greitai keičiasi, vyksta daug veiksmo ir judėjimo, programai sunku atskirti, ar tai purvas ir dėmės, ar tiesiog greitai besikeičiantys vaizdai. Dėl to yra sukurta ši funkcija. Didesnė funkcijos nustatymo reikšmė – mažiau purvo, valiklis rūpinsis vaizdo įrašo judėjimo objektais ir gali būti, kad kai kurie objektai bus pašalinti. Reiktų pasirūpinti nepersistengti su šia funkcija. Optimalus nustatymas yra tarp 1 ir 40 daugumoje atvejų.



## **„Dirt Brightness Range“**

Purvas ne visą laiką atrodo taip pat, skirtinguose kadruose jis gali turėti skirtingus baltos ir tamsios spalvos atspalvius. Naudojant „Low“ ir „High“ parametrus nusakome programai šviesumo spektrą vaizdo įrašė.

## **„FeatherMask“**

Ši papildoma kontrolė gali padėti kai kuriais atvejais, kai jūs gaunate sienas aplink ištrintas dėmės. Su šia kontrole galima pašalinti tas sienas.

### **Svarbu:**

„Dirt Remover“ programa nesugebės surasti dėmių vietų, jei bus pakeistas vaizdo įrašo greitis. Būtina ištrinti visas dėmes prieš keičiant vaizdo įrašo greitį.

### **Keli patarimai:**

Dėmės bus surastos ir ištrintos tik tuo atveju, jei dėmė bus matoma tik viename kadre, o ne kadru virtinėje, jei dėmę matome per keletą kadru, ji nebus aptikta ir ištrinta. Jei dėmė labai didelė, verta pamastyti apie kitą dėmės šalinimo būdą, naudojant kitą programą: pvz.: „DeadPixel“ ar „DropoutEliminator“.

### **2.2.3. „DropoutEliminator V1.0“**

Pataiso net ir blogiausių nubėgimus iš šalia esančių kadru komponentų.

Programa sukurta nubėgimams ir išdegimams kadruose pašalinti. Jei vaizdo įrašė yra dideli trūkumai, kai susilieėjimas ar išdegimas, programa tai pašalina naudodama šalia esančių kadru komponentus. Aišku, reikėtų suprasti, kad programai nepavyks pašalinti sugadintų komponentų, jei nebus greta esančių nesugadintų kadru, kuriais būtų galima pasinaudoti siekiant sutvarkyti vaizdo įrašo spragas.

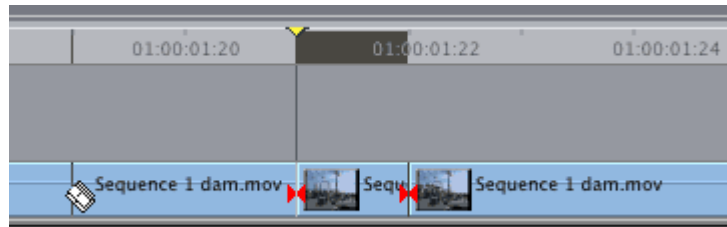
Pirmiausia, reikia nustatyti ar pažeidimas viename lauke, ar visame kadre. Vaizdo įrašai dažniausiai būna pažeisti visame kadre. Būtina tai iš anksto žinoti, siekiant maksimalaus rezultato. Programa suskirstyta į kelias rūšis.

Žvilgsnis į sugadintą pavyzdį:



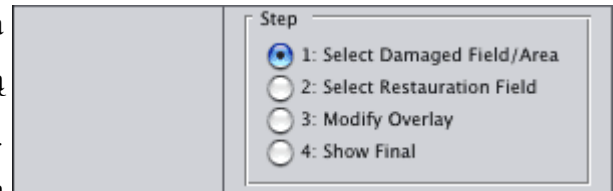
Aiškiai matomi vaizdo pažeidimai. Paveikslėlis dešinėje parodo, kad tik vienas laukas yra pažeistas.

Tai išsiaiškinius reikia atskirti būtent tą pažeistą vietą. Reikia pasinaudoti „blade-tool“ įrankiu (tiesiog paspaudžiama B raidė) ir iškerpamas sugadintas kadras.



### Žingsnis po žingsnio:

Pataisyti sugadintą vaizdo įrašo dalį programa naudoja 4 žingsnių metodą. Tiesiog reikia viską atlikti žingsnis po žingsnio ir visi sugadinimai dings. Pirmiausia reikia pasirinkti sugadintą vietą, tada pasirinkti nesugadintą vietą, kuri yra panaši į tą, kurią siekiame sutvarkyti. Dažnai pasitaiko, kad tinka toliau einantis, ar prieš tai ėjęs kadras. Verta pasižiūrėti ir atrinkti labiausiai tikusį kadra. Kitas žingsnis pakeisti įkeltą lauką, kad jis atitiktų judesį ir spalvas. Galiausiai galima peržiūrėti sutvarkytas vietas ir jas išsaugoti.



### **1 žingsnis: Pasirinktos vietas pasirinkimas**

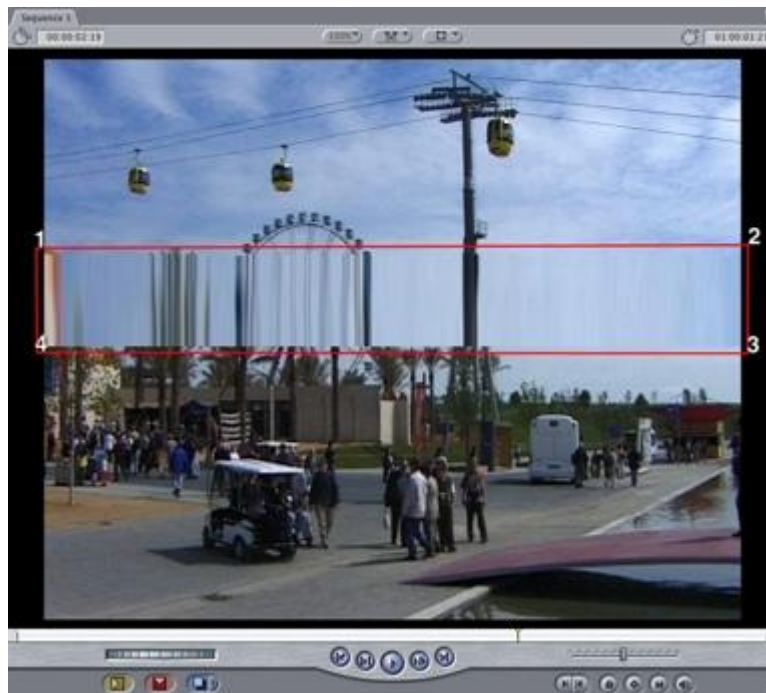
Pirmiausia, reikia pasirinkti sugadintą įrašo dalį. Šiuo atveju, tai „Field 1“.



[ 1: Select Damaged Field/Area ]	
Select Field	Field 0
Corner 1	(+) -288 , -57.6
Corner 2	(+) 288 , -57.6
Corner 3	(+) 288 , 57.6
Corner 4	(+) -288 , 57.6



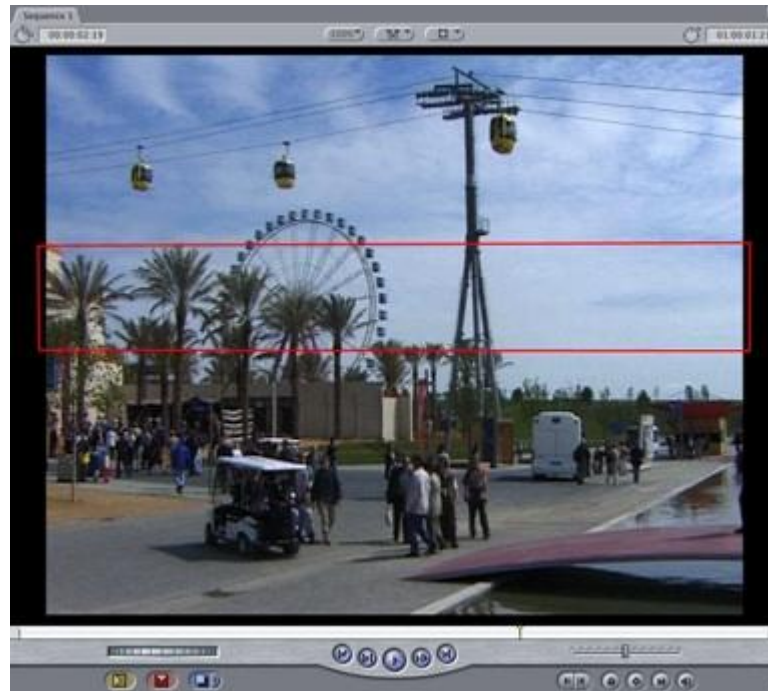
Dabar reikia pažymėti ją, vadinamąją kampų kontrolę. Patartina pasirinkti šiek tiek didesnę vietą, nei sugadinto kadro dydis.



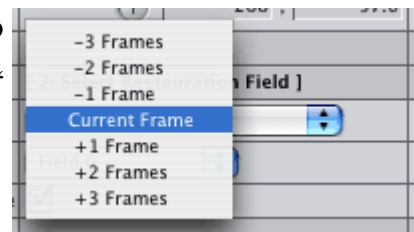
## 2 žingsnis: pasirinkti nesugadintą, panašų kadrą

Reikia pasirinkti nesugadintą, panašų kadrą. Pagal šį pavyzdį, „Field1“ yra sugadintas kadras, o „Field 0“ ne, kaip parodyta paveikslėlyje žemiau:

[ 2: Select Restoration Field ]	
Used Frame	Current Frame
Select Field	Field 0
Frame Damaged Are	<input checked="" type="checkbox"/>



Jeif pasirodytų, kad ir tas laukas pažeistas, ar neapima viso pažeisto lauko, tai galima pasirinkti kitą kadrą „UsedFrame“ skiltyje, kurioje galima pasirinkti tarp 6 skirtingų kadrų.



### 3 žingsnis: „ModifyOverlay“

Čia galima keisti pasirinktą (gerą) lauką, ar kadrą. Nereikia nieko keisti, jei vaizdo įrašas nepasikeitė per tuos kadrus (pažeistą ir ne).

Tačiau, jei kamera sujudėjo, kontrastas, šviesumas, spalva pasikeitė galima kompensuoti skirtumą šioje skiltyje.

Taip pat galima rasti „FeatherOverlay“ parametą. Jei modifikuotas laukas visiškai neatitinka naujojo lauko, tai gali padėti šis parametras, kuris leidžia paslėpti nežymius skirtumus tarp objektų ir kampų, paslėpdamas nežymius skirtumus tarp kadru.

Svarbu prisiminti: programa sukurta taisyti sugadintus kadrus ir vietas remiantis kitais to pačio vaizdo įrašo kadrais. Jei nebus kadru, galinčių kompensuoti pažeistų vietų, programai nepavyks atlikti savo darbo ir ištaisyti bei pakeisti sugadintų vietų ir kadru.

### 4 žingsnis: „ShowFinal“

Paskutiniam žingsniui tereikia nuspręsti ar visi nustatymai tinkami ir paleisti programą.

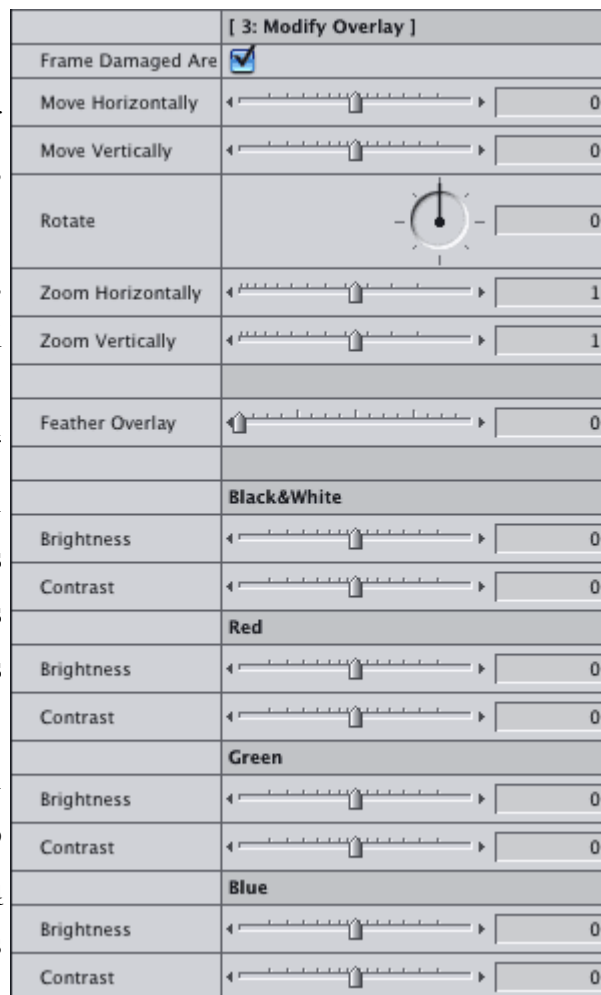
## 2.3. Gauso filtras

Matematikoje, **Gauso funkcija** yra funkcija apibrėžiama kaip (1):

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}, a, b, c \in R, a > 0, c > 0. (1)$$

Gauso funkcijos grafikas yra būdingas savo simetriniu varpo pavidalu, kuris greitai nuslopsta funkcijos argumento vertėms artėjant į teigiamą arba neigiamą begalybę. Parametras  $a$  yra kreivės maksimumo aukštis, dydis  $b$  yra maksimumo centro padėtis, o dydis  $c$  nulemia "varpo" plotį.

Gauso funkcijos yra plačiai paplitusios įvairiose mokslo srityse – statistikoje, kur jos aprašo normalųjį tikimybės tankio skirstinį, signalų teorijoje, kur jos aprašo Gauso filtras, vaizdų apdorojime, kur dvimatė Gauso funkcija naudojama "blur" filtro algoritme, bei fizikoje, kur jos yra šilumos pernašos ir parabolinės difrakcijos teorijos diferencialinių lygčių sprendiniai.

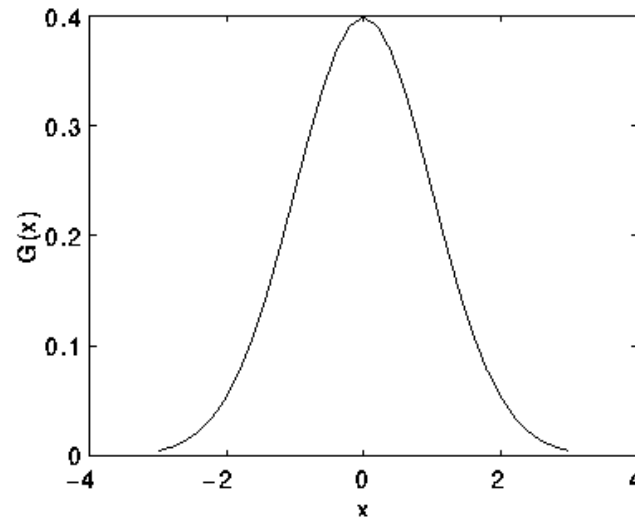


Gauso filtras yra paveikslėliams filtruoti, suteikia paveikslėliams išliejimo efektą (primenantį nesufokusuotos kameros vaizdą).

Vienmatis gauso filtras aprašomas formule (2):

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (2)$$

čia  $\sigma$  yra gauso pasiskirstymo funkcijos standartinis nuokrypis; pasiskirstymo funkcijos vidurkis yra laikomas  $x=0$ . Gauso pasiskirstymas pavaizduotas 2.15 paveiksle.

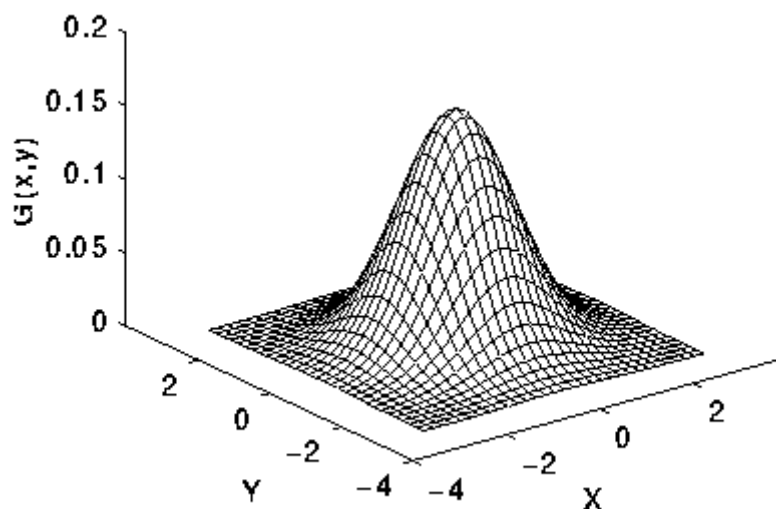


**Pav. 2.15** Vienmatis gauso pasiskirstymas su vidurkiu 0 ir standartiniu nuokrypiu  $\sigma = 1$ .

Paveikslėlio filtravimui yra naudojamas dvimatis gauso filtras (3):

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

Jo grafikas pavaizduotas 2.16 paveiksle.



**Pav. 2.16** Dvimatis gauso pasiskirstymas su vidurkiu (0,0) ir standartiniu nuokrypiu  $\sigma = 1$ .

Gausinis pasiskirstymas yra visur teigiamas, taigi, begalinio dydžio. Kadangi paveikslėlis gali būti interpretuojamas kaip dvimatė pikselių matrica, tai atliekant filtravimą reikia diskretizuoti Gauso filtrą. Jo dydis yra pasirenkamas toks, kad toliausiai nuo centro nutolusios reikšmės jau būtų pakankamai artimos nuliui.

Gauso išlyginimo idėja yra naudoti šį 2-D pasiskirstymą, kaip apskaičiavimo funkciją, o tai pasiekama Konvoliucijos metodu. Kai vaizdas yra saugomas kaip atskirų taškų rinkinys, tada turime gaminti atskirą suderinimą su Gauso funkcija, kol galime atlikti sutraukimo funkciją.

Teoriškai Gauso pasiskirstymas ne visose situacijose yra lygus nuliui, todėl yra reikalingas arba naudojamas sąsūkos metodas. Bet praktikoje arba realioje situacijoje veiksmingas nulis yra daugiau trys standartiniai nuokrypiai nuo vidurkio ir šioje situacijoje mes galėtume trumpinti branduolį. 2.17 paveiksle matyti, tinkamą sveikųjų simbolių sutraukimo branduolys, artimas Gauso  $\sigma = 1$ . Nėra aišku kaip pasirinkti tinkamai suderintą Gauso reikšmę, kuri yra užslėpta. Gauso vertė gali būti naudojama kaukės centriniame pikselyje, bet tai nebus tikslu, todėl Gauso vertė varijuoja netiesiškai pikselio. Gauso reikšmė integruojama šalia viso pilno pikselio pagal Gauso suminių padidėjimą 0.001. Integralai nėra sveikieji skaičiai tol, kol susiejamos masvyvo vertės, nepasiekė vertės 1. Galiausiai pasiekta vertė 273 yra visų kaukės svorių suma.

$\frac{1}{273}$	1	4	7	4	1
	4	16	26	16	4
	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

**Pav. 2.17** Diskrečioji aproksimacija Gauso funkcija su standartiniu nuokrypiu  $\sigma = 1.0$ .

Kai tik branduolys buvo apskaičiuotas, Gauso glotninimas gali būti atliktas naudojant standartinį sąsūkos metodą. Faktiškai, sąsūka gali būti atlikta labai greitai izotropinę Gauso 2-D lygtį padalinus į dvi atskiras x ir y komponentes. Ši 2-D sąsūka gali būti atlikta susiejant su 1-D Gauso branduolį x kryptimi, ir po to susiejant kitą 1-D Gauso branduolį su y kryptimi. 2.18. pav. parodytas 1-D x komponento branduolys, kuris gali būti panaudotas sudarant visą sąsūkos branduolį 2.17. pav. Y komponentas panašiai yra toks pat, tik orientuotas vertikaliai 2.18. pav. Vienas iš 1-D

šąsūkos branduolių, naudojamas apskaičiuoti visą branduolį, parodytą 2.17. pav. (dauginant 273, apvalinant ir nupjaunant vieną eilę vaizdo elementų aplink sieną todėl, kad jie daugiausia yra 0. Tai mažina 7x7 matrica iki 5x5 parodytos anksčiau). Y komponentas yra lygiai toks pat, bet orientuotas vertikaliai..

.006	.061	.242	.383	.242	.061	.006
------	------	------	------	------	------	------

**Pav. 2.18** Vienas iš 1-D šąsūkos metodų, naudojamų apskaičiuojant visą branduolį, veikiantis greičiau nei prieš tai esantis metodas.

Kitas būdas apskaičiuoti Gauso suglotninimą su dideliu standartiniu nuokrypiu, tai apskaičiuoti paveikslą keletą kartų su mažesniu Gauso branduoliu. Jei skaičiavimo kompleksas turi konvejerinio apdorojimo aparatinės priemonės.

## 2.4. Branduolys

Branduolys paprastai yra maža skaičių matrica, kuri yra naudojama vaizdo šąsūkai. Branduoliai, turintys savyje skirtingą skaičių struktūrą, sukelia skirtingus rezultatus po šąsūkos., Pavyzdžiui, 2.19. paveiksle yra 3x3 branduolys, kuris įgyvendina vidurkio (mean) filtrą.

1	1	1
1	1	1
1	1	1

**Set of coordinate points =**  
**{ (-1, -1), (0, -1), (1, -1),**  
**(-1, 0), (0, 0), (1, 0),**  
**(-1, 1), (0, 1), (1, 1) }**

**Pav. 2.19** Šąsūkos branduolys vidurkio filtrui su 3x3 kaimynais.

## 2.5. Šąsūka

Šąsūka yra paprasta matematinė operacija, kuri yra svarbiausia daugeliui bendrų vaizdo apdorojimo operatorių. Šąsūka yra priemonė "sudauginimui" dviejų skaičių rinkinių, apskritai skirtingų dydžių, bet to paties matmens, kad pagamintų trečią to paties matmens skaičių rinkinį. Tai gali būti panaudojama vaizdo apdorojime, kad įgyvendintų operatorius, kurių išvesties vaizdo elementų vertės yra paprastos linijinės tam tikrų įvesties vaizdo elementų verčių kombinacijos. Vaizdo apdorojimo kontekste, vienas iš įvesties rinkinių yra normalus tik nespaltotas su pustoniais (gray level) paveikslas. Antras rinkinys yra paprastai daug mažesnis, ir yra taip pat dviejų matmenų



ir žinomas kaip branduolys. 2.20. pav. parodytas paveikslo ir branduolio pavyzdys, kad iliustruoti sąsūką.

<b>I<sub>11</sub></b>	<b>I<sub>12</sub></b>	<b>I<sub>13</sub></b>	<b>I<sub>14</sub></b>	<b>I<sub>15</sub></b>	<b>I<sub>16</sub></b>	<b>I<sub>17</sub></b>	<b>I<sub>18</sub></b>	<b>I<sub>19</sub></b>
<b>I<sub>21</sub></b>	<b>I<sub>22</sub></b>	<b>I<sub>23</sub></b>	<b>I<sub>24</sub></b>	<b>I<sub>25</sub></b>	<b>I<sub>26</sub></b>	<b>I<sub>27</sub></b>	<b>I<sub>28</sub></b>	<b>I<sub>29</sub></b>
<b>I<sub>31</sub></b>	<b>I<sub>32</sub></b>	<b>I<sub>33</sub></b>	<b>I<sub>34</sub></b>	<b>I<sub>35</sub></b>	<b>I<sub>36</sub></b>	<b>I<sub>37</sub></b>	<b>I<sub>38</sub></b>	<b>I<sub>39</sub></b>
<b>I<sub>41</sub></b>	<b>I<sub>42</sub></b>	<b>I<sub>43</sub></b>	<b>I<sub>44</sub></b>	<b>I<sub>45</sub></b>	<b>I<sub>46</sub></b>	<b>I<sub>47</sub></b>	<b>I<sub>48</sub></b>	<b>I<sub>49</sub></b>
<b>I<sub>51</sub></b>	<b>I<sub>52</sub></b>	<b>I<sub>53</sub></b>	<b>I<sub>54</sub></b>	<b>I<sub>55</sub></b>	<b>I<sub>56</sub></b>	<b>I<sub>57</sub></b>	<b>I<sub>58</sub></b>	<b>I<sub>59</sub></b>
<b>I<sub>61</sub></b>	<b>I<sub>62</sub></b>	<b>I<sub>63</sub></b>	<b>I<sub>64</sub></b>	<b>I<sub>65</sub></b>	<b>I<sub>66</sub></b>	<b>I<sub>67</sub></b>	<b>I<sub>68</sub></b>	<b>I<sub>69</sub></b>

<b>K<sub>11</sub></b>	<b>K<sub>12</sub></b>	<b>K<sub>13</sub></b>
<b>K<sub>21</sub></b>	<b>K<sub>22</sub></b>	<b>K<sub>23</sub></b>

**Pav. 2.20** Mažas paveikslukas (kairėje) ir branduolys (dešinėje) iliustruoja sąsūką. Reikšmės kiekvieno grotelių kvadrato viduje yra panaudotos, kad identifikuotų kiekvieną kvadratą.

Sąsūka yra atliekama praslystant branduoliu virš paveikslo, apskritai pradėdant iš viršutinio kairiojo kampo, kad perkeltų branduolį per visas padėtis, kur branduolys atitinka visišką atvaizdą sienų viduje. Kiekviena branduolio padėtis atitinka vieną išėjimo vaizdo elementą, kurio vertė yra apskaičiuojama, dauginant branduolio vertę ir pagrindinę atvaizdo vaizdo elemento vertę kiekvienai iš ląstelių branduolyje, ir sudedant visus šituos skaičius.

Taip, pavyzdyje, prie apatinio dešiniojo vaizdo elemento vertės, paveikslo išėjimas duos (4):

$$O_{57} = I_{57}K_{11} + I_{58}K_{12} + I_{59}K_{13} + I_{67}K_{21} + I_{68}K_{22} + I_{69}K_{23} \quad (4)$$

Jei atvaizdas turės M eilučių ir N stulpelių, ir branduolys turės m eilučių ir n stulpelių, tai išėjimo atvaizdo dydis turės M - m + 1 eilučių, ir N - n + 1 stulpelių.

Matematiškai mes galime parašyti sąsūką taip (5):

$$O(i,j) = \sum_{k=1}^m \sum_{l=1}^n I(i+k-l, j+l-1)K(k,l) \quad (5)$$

kur i kinta nuo 1 iki M - m + 1 ir j nuo 1 iki N - n + 1.

Pastebėtina, kad dauguma sąsūkos įgyvendinimų parodė didesnę paveikslo išvestį, todėl kad jie atliko mažesnę spaudimą, kad branduolys būtų pajudintas į pozicijas, kur jis visiškai atitinka paveikslėlį. Be to, šie įgyvendinimai tipiškai paslinko branduolį į tas pozicijas, kur tik viršutinis kairinis kampas siejasi su paveikslu. Taip pat, branduolys iš dalies uždengia paveikslo kontūrus apačioje ir dešinėje. Vienas teigiamas bruožas šito priartėjimo yra tas, kad išvesties paveikslas yra tokio paties dydžio kaip ir įvesties paveikslas. Deja, ta tvarka apskaičiuojant išvesties taško vertes iš paveikslo kontūrų apačios ir dešinės, reikalinga sukurti įvesties taško vertes toms vietoms, kur branduolys nusitęsia į jų pabaigas.

Jeigu jūs naudojate sąsūkas, darant tai, yra geriau prisegti paveikslą, kad pašalinti netikras vietas. Pašalinant  $n-1$  taškus iš dešinės pusės ir  $m-1$  taškus iš apačios, viską sutvarkys.

Sąsūkos naudojamos, įgyvendinant daug skirtingų operatorių, ypač erdvinius filtrus ir savybių detektorius.

## 2.6. Vaizdo apdorojimo slenkstis

Daugelyje vaizdų apdorojimo programų, naudinga sugebėti atrinkti paveikslo ribas, atitinkančias objektus, kuriais mes domimės, nuo paveikslo sričių, kurios atitinka foną. Slenkstinė vertė dažnai suteikia lengvą ir patogų būdą įvykdyti šį atskyrimą pasinaudojant skirtingu intensyvumu ar spalvomis priekiniame plane ir foninėse atvaizdo srityse.



**Pav. 2.21** Vaizdo apdorojimas slenkščio metodu.

Taip pat, dažnai naudinga turėti galimybę pamatyti, kokios atvaizdo sritys susideda iš vaizdo elementų, kurių vertės yra apibrėžtame diapazone, ar priklauso intensyvumo ar spalvos grupei. Slenkstinė vertė tam taip pat gali būti naudojama.

Slenkstinės vertės apdorojimas gali būti atliktas nespalvotam arba spalvotam paveikslui, išvestis yra dvejetainis vaizdas, kuriame baltas taškas yra vaizdas, o juodas taškas – fonas.

Paprastas skirstymas yra nustatomas vienu parametru, žinomu kaip intensyvumo slenkstis. Vienu praėjimu, kiekvienas vaizdo elementas paveiksle yra palyginamas su šiuo slenkščiu. Jei vaizdo elemento intensyvumas yra aukštesnis negu slenkstis, vaizdo elementas yra nustatomas baltu, jei intensyvumas yra mažesnis, negu slenkstis, tai elementas nustatomas juodu.

Sudėtingiau įgyvendinami, daugialypiai slenkščiai, kai intensyvumo verčių grupė nustatoma kaip balta tuo metu, kai visi kiti taškai nustatomi kaip juodi. Spalvotiems ar daugiaspektriniams paveikslams, gali būti nustatomi skirtingi slenkščiai kiekvienam spalviniam kanalui, ir taip išskirti tik tuos vaizdo elementus kurie yra apibrėžto kuboido viduje RGB erdvėje.

Kitas variantas, nustatyti kaip juodus visus taškus susietus (priklausančius) fonui, bet paliekant vaizdo taškus jų originalia spalva/ryškumu (priešingai nei paverčiant juos baltais), tada ta informacija nebus prarasta.

## 2.7. Medianinis filtras

Kaip ir vidurkio filtras medianinis filtras atsižvelgia į kiekvieną tašką paveiksle iš eilės ir žiūri į artimiausius kaimynus nuspręsdamas, kokios spalvos jis bus. Užuoat pakeitęs reikšmę vidurkiu, jis pakeičia reikšmę aplinkinių reikšmių mediana. Mediana yra apskaičiuojama surūšiuojant kaimyninius taškus pagal jų spalvines reikšmes didėjimo tvarka ir išrenkant vidurinio taško reikšmę. Jei kaimynų skaičius yra lyginis, tai naudojamas dviejų vidurinių taškų vidurkis.

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

**Neighbourhood values:**

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

**Median value: 124**

**Pav. 2.22** Kaip matoma centrinio taško reikšmė 150 labai skiriasi nuo aplink esančių, ir ji yra pakeičiama medianos reikšme - 124. Čia naudojama 3x3 kaimynų kvadratas.

Medianos filtras turi du privalumus lyginant su vidurkio filtru:

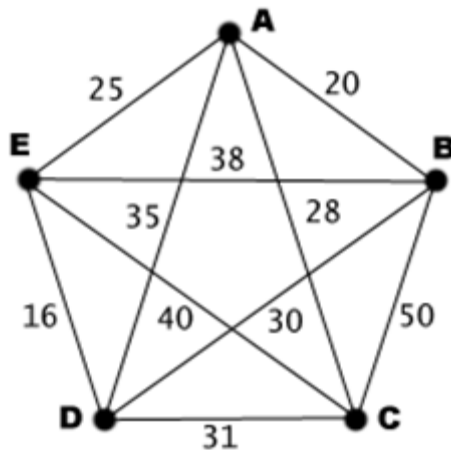
- Vienas nebūdingas kaimyninis taškas neturės pastebimo efekto skaičiuojant medianą.
- Kadangi medianos reikšmė turi būti vieno iš kaimyninių taškų reikšmė, tai šis filtras nesukuria naujų nenaudojamų taškų reikšmių, procesui artėjant prie kraštų.

## 2.8. Artimiausio kaimyno metodas

Beveik viską pasako pats pavadinimas. Algoritmo principas toks:

1. tolimesnę viršūnę reikia rinktis iš dar neaplankytų pačią „artimiausią“. Kai nebelieka neaplankytų viršūnių – grįžtame į pradinę.

Panagrinėkime pavyzdį. Turime grafą:



Pav. 2.23 Grafo pavyzdys

2.1 lentelė Grafo kelio svoriai.

AB	20
AC	28
AD	35
AE	25
BC	50
BD	30
BE	38
CD	31
CE	40
DE	16

Pradedame nuo viršūnės D. Pasinaudoję perrinkimu galime rasti trumpiausią ciklą: DCABED, kurio svoris yra 133 (ilgiausias ciklas yra DBCEAD, ir jo svoris – 180).

Dabar raskime ciklą naudodamiesi „artimiausio kaimyno“ metodu:

1. trumpiausia jungtis einanti iš D yra DE (16);
2. trumpiausia jungtis einanti iš E yra ED, bet D jau aplankyta, dėl to renkamės kitą – EA (25);
3. trumpiausia jungtis einanti iš A yra AB (20);
4. vienintelė jungtis einanti iš B į dar neaplankyta viršūnę yra BC (50);
5. grįžtame į D (CD - 31).

Gavome ciklą DEABCD, kurio svoris yra 142.

Kaip matome pasinaudoję apytiksliau algoritmu sutaupėme daug laiko ir gavome neblogą (nors ir ne geriausią) sprendimą.

Dabar raskime ciklą naudodamiesi „artimiausio kaimyno“ metodu pradėdami nuo viršūnės A:

1. trumpiausia jungtis einanti iš A yra AB (20);

2. trumpiausia jungtis einanti iš B yra BA, bet A jau aplankyta, dėl to renkamės kitą – BD (30);
3. trumpiausia jungtis einanti iš D yra DE (16);
4. vienintelė jungtis einanti iš E į dar neaplankytą viršūnę yra EC (40);
5. grįžtame į A (AC – 16).

Gavome ciklą ABDECA, kurio svoris yra 134.

Šį kartą gautasis ciklas labai mažai skiriasi nuo optimalaus. Iš to, taip pat matome, kad algoritmo duodamas sprendimas priklauso nuo pasirinktos viršūnės. Tai pastebėję galime „patobulinti“ algoritmą (pavadinkime „pasikartojančio artimiausio kaimyno“ metodu):

1. iš pradžių imame bet kurią viršūnę ir taikome „artimiausio kaimyno“ algoritmą, užsirašome sprendimą;
2. kartojame pirmąjį žingsnį tik perrašome sprendimą jei naujai gautasis yra geresnis.

Tokio algoritmo antrąjį žingsnį galime kartoti visoms viršūnėms (arba ir ne visoms). Kadangi gautas atsakymas yra ciklas, tai bet kokį ciklą galime perrašyti, kad bet kuri viršūnė būtų pirmoji.

### 3. REALIZACIJA IR TESTAVIMAS

Programos byloje yra "programa.m" failas paleidžia pirmą programos etapą kuriuo programa nuskaito toje pačioje direktorijoje esančius duomenis.

Failas " medfilt3.m" paleidžia papildomą programos dalį. Tai medianinis filtras, kuris apdoroja kiekvieną kadrą.

Kadangi Medianinio transformacijos reikalavimai pasižymi didele atmintim, ilgu skaičiavimo darbu– kuris didėja eksponentiškai nuo užduotų parametrų kiekio, tai reikalauja didesnių kompiuterio resursų. Ypač aktualu jei vaizdo įrašas yra ganėtinai ilgas.

Kompiuterio parametrai (procesorius, darbinė atmintis, kietasis diskas) kuriais buvo atlikti bandymai:

#### 3.1 lentelė Kompiuterio charakteristikos

Procesorius	Intel (R) Core(TM) i7-4702 2.2Ghz
Vidinė atmintis	7.88 GB of RAM (DDR3)
Standusis diskas	7.200 rpm Buf 6 Gb SATA III

Pasirinktų objektų retušavimo vaido įrašė, buvo pasirinkti trys neilgi vaizdo įrašai ir sukurtu įrankiu pamėginta panaikinti televizijos logotipą. Toliau eigoje buvo surasti, jau sukurti įrankiai, su kuriais bus lyginama sukurtasis įrankis. Prieš tai esančiame skyriuje yra aprašyti metodai, kurių pagalba sukurtas televizijos logotipo šalinimo įrankis.

Įrankis yra realizuotas „Matlab“ aplinkoje, pasitarus su kuruojančiu dėstytoju. Kadangi „Matlab“ – tai viena seniausių ir galingiausių specialios paskirties kompiuterinių programų, skirtų automatizuoti mokslinius ir inžinierinius skaičiavimus. Darbas su šia galinga programa nėra labai sudėtingas, tad turint programavimo įgudžius galima lengvai atlikti daug sudėtingų operacijų su vaizdo įrašais.

#### 3.1. Programos algoritmo kūrimas

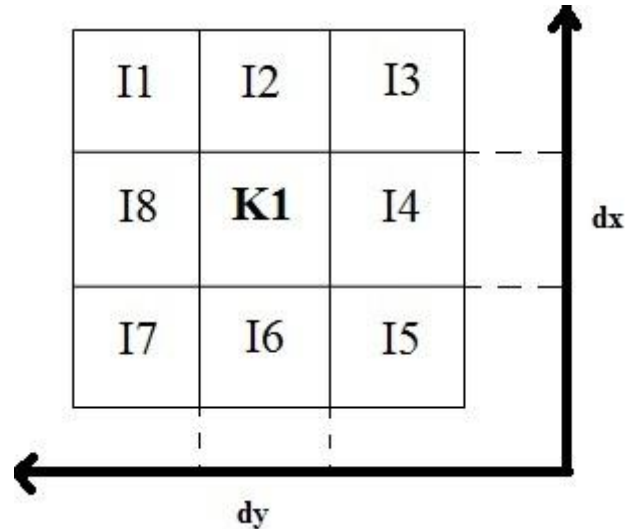
##### 3.1.1. Pirmasis etapas

Visų pirma, programa nuskaito vaizdo įrašą .avi formatu. Kiekvieną vaizdo įrašo kadrą paverčia .jpg formatu. Kiekvienas vaizdo įrašo kadras programos veikimo metu saugomas laikinoje atmintyje. Norint yra galimybė visus pakeistus vaizdo įrašo kadrus išsisaugoti programos darbiname aplankale.

Pirmiausiai, vaizdo įrašė iškerpamas kvadratas, toje vietoje kur yra televizijos logotipas. Vietoje iškirptojo kvadrato užpildau balta spalva pagal RGB spavų paletę (255,255,255) svoriais.

Galima pasirinkti ir juodą spalvą RGB(0,0,0), bet atliekant vidurkio skaičiavimą juoda spalva iškraipo gaunamą rezultatą.

Iškirptas kvadratis vaizdo įrašas pažymimas K1 reikšme. Apinkui esantys kvadratis pažymimi I1...I8, iš šių kvadratų bus gaunama K1 reikšmė.



**Pav.3.1** Algoritmo sudarymo schema

Vidurinis K1 iškirptasis langelis yra sudaromas pagal formulę (6):

$$K1 = \frac{I1}{8} + \frac{I2}{8} + \frac{I3}{8} + \frac{I4}{8} + \frac{I5}{8} + \frac{I6}{8} + \frac{I7}{8} + \frac{I8}{8} \quad (6)$$

Gauti K1 reikšmę paskaičiuojama aplink esančių langelių vidurkis ir užpildoma tuščia vieta. Programa ima kiekvieną pagamintą kadrą .jpg formatu iškerpa kvadratėlį vietoje televizijos logotipo. Kiekvienam kadrai paskaičiuojami aplink esančių langelių bendras vidurkis ir gauta reikšmė įklijuojama į K1 langelį. Visi pakeisti kadrai sudedami į vaizdo įrašą, kurio formatas yra .mov.

Testavimui naudosiu tris vaizdo įrašus, kad būtų galima palyginti gautus rezultatus.

Testas Nr. 1.





**Pav.3.2** Pirmojo testo pradinis variantas.



**Pav3.3** Pirmojo testo gautasis variantas.

Testas Nr. 2.



**Pav.3.4** Antrojo testo pradinis variantas.



**Pav.3.5** Antrojo testo gautasis variantas.

Testas Nr.3



**Pav.3.6** Trečiojo testo pradinis variantas.



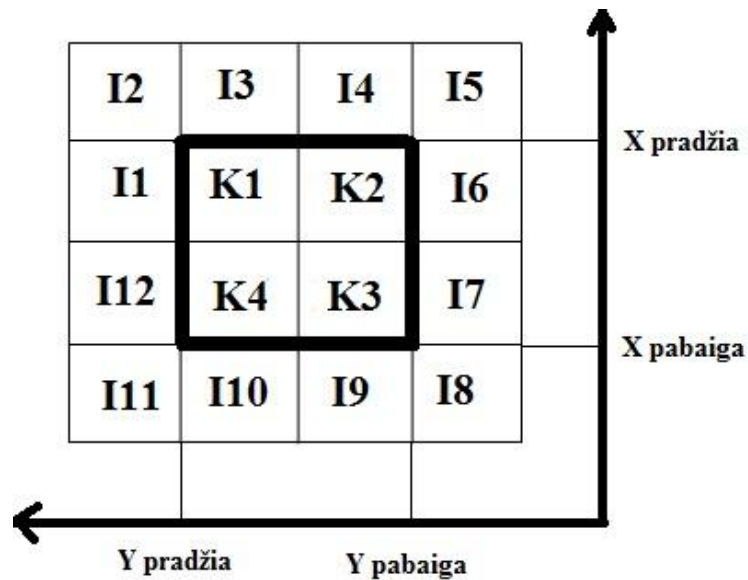
**Pav.3.7** Trečiojo testo galutinis variantas.

Įvertinęs rezultatus, priėjau išvados, kad ne visada šį įrankį bus galima naudoti. Kadangi logotipas yra vaizdo įrašo dešiniame viršutiniame kampe, gaminant iškirptą kvadratą paimami kraštiniai erdvėje esantys taškai. Taip pat pastebėjau – naudojant daugiau taškų, atkurti iškirptam kvadratui, pasiekama geresnė kokybė. Todėl antru epatu bus atliekama programos optimizacija.

### **3.1.2. Uždavinio optimizavimas**

Optimizuojant uždavinį iškirptas kvadratas padalinamas dar į keturias dalis ir priskiriama jam balta spalva. Kiekvieną  $K1 \dots K4$  iškirptą kvadratą gauname iš šalia esančių trijų kvadratėlių, visų trijų kvadratėlių apskaičiuojame vidurkį, gautą rezultatą įklijuojame į  $K$  reikšmę. Taip

apskaičiuojame kiekvieną K paeiliui užpildome vidurkio reikšme. Šios operacijos atliekamos kiekvienam vaizdo įrašo kadrai. Kiekvienas pakeistas kadras atgal yra surašomas į vaizdo įrašą .mov formatu.



**Pav.3.8** Galutinio algoritmo sudarymo schema.

K1...K4 iškirpti langeliai yra sudaromi pagal formules(7, 8, 9, 10):

$$K1 = \frac{I1}{3} + \frac{I2}{3} + \frac{I3}{3} \quad (7)$$

$$K2 = \frac{I4}{3} + \frac{I5}{3} + \frac{I6}{3} \quad (8)$$

$$K3 = \frac{I7}{3} + \frac{I8}{3} + \frac{I9}{3} \quad (9)$$

$$K4 = \frac{I10}{3} + \frac{I11}{3} + \frac{I12}{3} \quad (10)$$

Testavimui naudosiu tuos pačius tris vaizdo įrašus, kad būtų galima palyginti gautus rezultatus.

Testas Nr. 1.



**Pav.3.9** Galutinio įrankio pirmo testo pateikiamas vaizdo įrašas.



**Pav.3.10** Galutinio įrankio pirmojo testo gautas vaizdo įrašo rezultatas.

Testas Nr. 2



**Pav.3.11** Galutinio įrankio antro testo pateikiamas vaizdo įrašas.



**Pav.3.12** Galutinio įrankio antro testo gautas vaizdo įrašo rezultatas.

Testas Nr. 3



**Pav.3.13** Galutinio įrankio pirmo testo pateikiamas vaizdo įrašas.



**Pav.3.14** Galutinio įrankio pirmojo testo gautas vaizdo įrašo rezultatas.

### 3.2. Skirtingų įrankių palyginimas

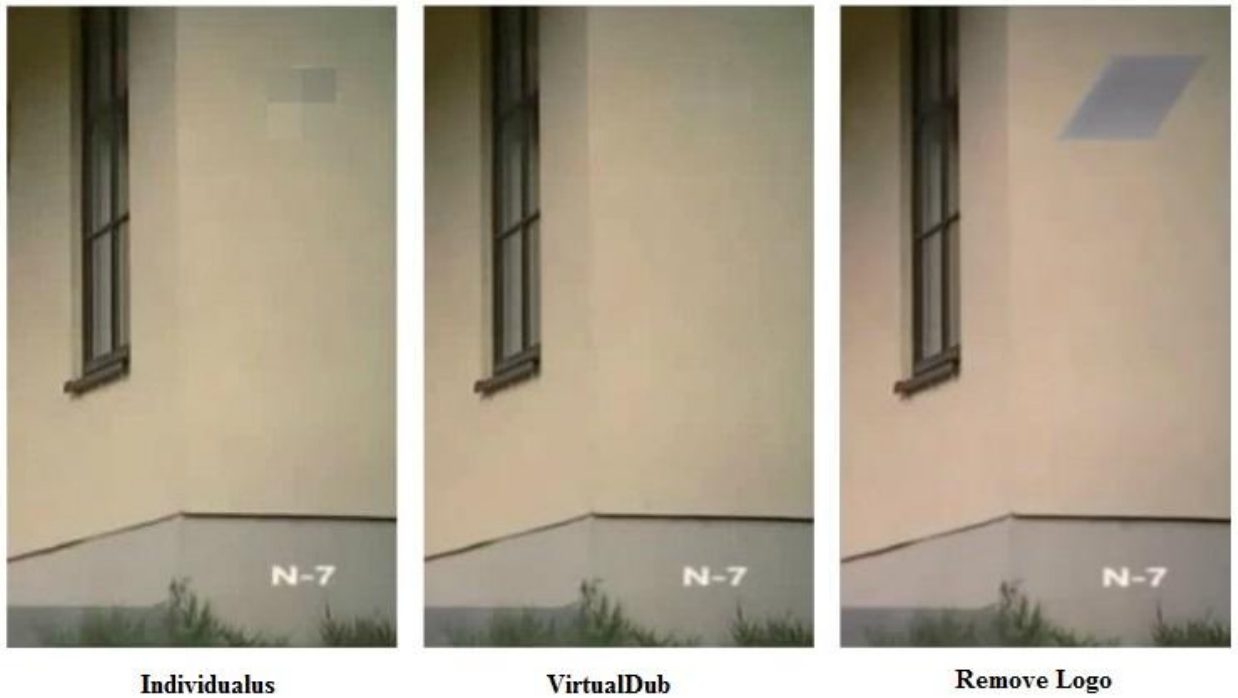
Įrankių palyginimui naudosiu :

- Matlab aplinkoje sukurtą individualų įrankį;
- VirtualDub komercinį įrankį;
- Remove Logo komercinį įrankį.

Šiais įrankiais bus panaikintas televizijos logotipas ir amžiaus cenzas. Bus palyginta kuris įrankis geriau iš vaizdo įrašo pašalina nereikalingus logotipus ir amžiaus cenzus. Tam atlikti bus naudojami trys vaizdo įrašai, kad stipriai skirtūsi kontrastai.



- Televizijos logotipo pašalinimas iš pirmojo vaizdo įrašo.



**Pav.3.15** Televizijos logotipo pašalinimas iš pirmo vaizdo įrašo.

- Televizijos logotipo pašalinimas iš antrojo vaizdo įrašo.



**Pav.3.16** Televizijos logotipo pašalinimas iš antrojo vaizdo įrašo.

- Televizijos logotipo pašalinimas iš trečiojo vaizdo įrašo.



Individualus



VirtualDub



Remove Logo

**Pav.3.17** Televizijos logotipo šalinimas iš trečiojo vaizdo įrašo.

- Amžiaus cenzo pašalinimas iš pirmojo vaizdo įrašo.



Individualus



VirtualDub



Remove Logo

**Pav.3.18** Amžiaus cenzo šalinimas iš pirmojo vaizdo įrašo.

- Amžiaus cenzo pašalinimas iš antrojo vaizdo įrašo.



Individualus



VirtualDub



Remove Logo

**Pav.3.19** Amžiaus cenzo šalinimas iš antrojo vaizdo įrašo.

- Amžiaus cenzo pašalinimas iš trečiojo vaizdo įrašo.



**Pav.3.20** Amžiaus cenzo šalinimas iš trečiojo vaizdo įrašo.

Palyginus skirtingus įrankius, padarytos išvalgos, kad „Individualus“ ir „VirtualDub“ veikia geriau tam tikruose spalvų kontrastuose. Tolimesniam programos tobulinimui, galima pritaikyti antrinkimo metodą. Iš histogramų grafikų parinkti kuris įrankis tinka pagal spalvų kontrastus.

#### 4. REZULTATŲ APIBENDRINIMAS IR IŠVADOS

- Buvo realizuotas medianinis filtro panaudojimas, iškirpto fragmento vaizdo įrašo užpildymui – logotipo šalinimui. Buvo pasirinkti du diskretizacijos lygmenys 1 ir 4, t.y. paveikslėlyje iškirptas laukas yra aproksimuojamas vienu arba keturiais aplinkinių regionų laukais. Keturių laukelių diskretizacijos medianinis filtras tinkamenis. Todėl naudingas tik vaizdo šalinimui, kurio fone egzistuoja didelių kontrastų fragmentai, o vieno laukelio bendru atveju yra artimesnis gauso filtrui ir yra universalus.
- Tirta komercinė „VirtualDub“ vaizdu apdorojimo programinė įranga parodė geresnius rezultatus. O „Remove Logo“ gerokai nusileidžia ne tik „VirtualDub“ bet ir individualaus darbo panaudojimui.
- Siekiant geresnių rezultatų reikalinga didinti iškerpamo laukelio diskretizacijos lygį. Metodiką reikėtų tobulinti įtraukiant iškerpamo regiono kontrastingumo nustatymą. Kaip kontrastingumo nustatymo kriterijus galėtų būti laukelio klasterizavimas, ir sluoksnių skaičiaus susiejimas su diskretizacijos skaičiumi medianiniame filtre.

## 5. LITERATŪROS SARAŠAS

1. Straipsnis iš elektroninio žurnalo: „Gaussian Smoothing“. [žiūrėta 2015 04 06].  
Prieiga per Internetą:  
<<http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>>
2. Straipsnis iš elektroninio žurnalo: „Convolution“. [žiūrėta 2015 04 15].  
Prieiga per Internetą:  
<<http://homepages.inf.ed.ac.uk/rbf/HIPR2/convolve.htm>>
3. Straipsnis iš elektroninio žurnalo: „Gaussian Smoothing“. [žiūrėta 2015 04 06].  
Prieiga per Internetą:  
<<http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>>
4. Straipsnis iš elektroninio žurnalo: „Kernel“. [žiūrėta 2015 04 15].  
Prieiga per Internetą:  
<<http://homepages.inf.ed.ac.uk/rbf/HIPR2/kernel.htm>>
5. Straipsnis iš elektroninio žurnalo: „Thresholding“. [žiūrėta 2015 05 02].  
Prieiga per Internetą:  
<<http://homepages.inf.ed.ac.uk/rbf/HIPR2/threshld.htm>>
6. Straipsnis iš vikipedijos: [žiūrėta 2014 10 14].  
Prieiga per internetą:  
<<http://lt.wikipedia.org/wiki/RGB>>

## **6. PRIEDAI**