



KAUNO TECHNOLOGIJOS UNIVERSITETAS
FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS
TAIKOMOSIOS MATEMATIKOS KATEDRA

Lina Vaišnoraitė

DISKREČIOSIOS DAUBECHIES $9/7$
TRANSFORMACIJOS SU DALINE BLOKŲ
DEKORELIACIJA SAVYBIŲ TYRIMAS

Magistro darbas

Vadovas
prof. dr. J. Valantinas

KAUNAS, 2011



KAUNO TECHNOLOGIJOS UNIVERSITETAS
FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS
TAIKOMOSIOS MATEMATIKOS KATEDRA

TVIRTINU
Katedros vedėjas
doc. dr. N.Listopadskis
2011 06 06

DISKREČIOSIOS DAUBECHIES 9/7
TRANSFORMACIJOS SU DALINE BLOKŲ
DEKORELIACIJA SAVYBIŲ TYRIMAS

Taikomosios matematikos magistro baigiamasis darbas

Vadovas
prof. dr. J.Valantinas
2011 05 27

Recenzentas
doc. N. Morkevičius
2011 05 27

Atliko
FMMM - 9 gr. stud.
L. Vaišnoraite
2011 05 27

KAUNAS, 2011

KVALIFIKACINĖ KOMISIJA

Pirmininkas: Leonas Saulis, profesorius (VGTU)

Sekretorius: Eimutis Valakevičius, docentas (KTU)

Nariai: Algimantas Jonas Aksomaitis, profesorius (KTU)

Vytautas Janilionis, docentas (KTU)

Vidmantas Povilas Pekarskas, profesorius (KTU)

Rimantas Rudzkis, habil. dr., vyriausiasis analitikas (DnB NORD Bankas)

Zenonas Navickas, profesorius (KTU)

Arūnas Barauskas, dr., vice-prezidentas projektams (UAB „Baltic Amadeus“)

Vaišnoraitė L. Analysis of properties of the discrete Daubechies 9/7 with partial block decorrelation: Master Thesis work in applied mathematics / supervisor prof. dr. J. Valantinas; Department of Applied mathematics, Faculty of Fundamental Sciences, Kaunas University of Technology. – Kaunas, 2011. – 50 p.

Summary

Many kernels (“mother” wavelets) can be used for the discrete wavelet transform DWT, like those of Daubechies, Morlet, discrete Le Gall transform (DLGT) or the discrete Haar transform (HT). Cohen-Daubechies-Feauveau (CDF 9/7) wavelet are the historically first family of biorthogonal wavelets, which was made popular by Ingrid Daubechies. These are not the same as the orthogonal Daubechies wavelets, and also not very similar in shape and properties. However their construction idea is the same. The JPEG 2000 compression standard uses the biorthogonal CDF 5/3 wavelet (also called the LeGall 5/3 wavelet) for lossless compression and a CDF 9/7 wavelet for lossy compression.

In this paper, two distinct DWT (CDF 9/7 and CDF 9/7 with decorrelation) as well as their computational algorithms are discussed, analyzed and compared. Comparison criteria are chosen to be one – dimensional hyperbolic filters and smoothness level of the digital signal under processing.

TURINYS

Summary.....	4
Lentelių sąrašas.....	6
Paveikslų sąrašas.....	7
Įvadas.....	8
1 Diskrečiosios bangelių transformacijos (DBT).....	9
1.1 Bendrosios bangelių savybės	10
1.2 Iteracinės DBT apskaičiavimo procedūros.....	12
1.3 Diskrečioji Daubechies 9/7 transformacija (CDF 9/7): savybės, skaičiavimo algoritmai	13
1.4 CDF 9/7 su daline blokų dekoreliacija.....	20
2 Bazinės CDF 9/7 ir CDF 9/7 su daline blokų dekoreliacija palyginamoji analizė.....	21
2.1 Bendra schema.....	22
2.1.1 Dvimačių hiperbolinių filtrų kontekstas.....	22
2.1.2 Filtravimo paklaidų įvertinimas	24
2.2 Programinės realizacijos ypatumai	25
3 Eksperimento rezultatų analizė.....	26
Išvados.....	35
Literatūra.....	36
Priedai.....	37
1. Priedas. Programų tekstai. Programa Matlab.....	37
1.1 CDF 9/7 spektro radimas vienmačiu atveju.	37
1.2 CDF 9/7 vaizdo radimas vienmačiu atveju.....	38
1.3 CDF 9/7 su daline blokų dekoreliacija vienmačiu atveju, spektro radimas.	39
1.4 CDF 9/7 su daline blokų dekoreliacija vienmačiu atveju, vaizdo radimas.	41
1.5 CDF 9/7 spektro radimas dvimačiu atveju.	43
1.6 CDF 9/7 vaizdo radimas dvimačiu atveju.	44
1.7 CDF 9/7 su daline blokų dekoreliacija dvimačiu atveju, spektro radimas.	46
1.8 CDF 9/7 su daline blokų dekoreliacija dvimačiu atveju, vaizdo radimas.	48
1.9 Pagrindinė programa naudojama paklaidų radimui hiperbolinio filtro kontekste.....	49

Lentelių sąrašas

1.1 lentelė. Žemo ir aukšto dažnio filtrai	14
2.1 lentelė. Suspaudimo koeficiento β reikšmės.....	26
3.1 lentelė. Palyginamoji 3.1 pav. paklaidų lentelė.....	27
3.2 lentelė. Palyginamoji 3.2 pav. paklaidų lentelė.....	29
3.3 lentelė. Palyginamoji 3.3 pav. paklaidų lentelė.....	31

Paveikslų sąrašas

1.1 pav. Transformacijos schema	9
1.2 pav. Piramidinio tipo algoritmas	12
1.3 pav. Bazinės CDF 9/7 transformacijos signalinis grafas	19
1.4 pav. CDF 9/7 su daline blokų dekoreliacija signalinis grafas, kai $m=3$	20
1.5 pav. CDF 9/7 su daline blokų dekoreliacija signalinis grafas, kai $m=2$	21
2.1 pav. Bendroji schema	22
2.2 pav. Hiperbolinių dvimačių skaitmeninių vaizdų filtravimas	23
2.3 pav. Hiperbolinio vaizdų filtro veikimo schema	24
3.1 pav. Barbara512.png	26
3.2 pav. Hill512.png	29
3.3 pav. Lake512.png	31
3.4 pav. CDF 9/7 su daline blokų dekoreliacija paklaidų grafikas	33
3.5 pav. CDF 9/7 su daline blokų dekoreliacija paklaidų grafikas	33
3.6 pav. Bazinės CDF 9/7 ir CDF 9/7 su daline blokų dekoreliacija paklaidų grafikas .	34

Ivadas

Diskrečiosioms bangelių transformacijoms galima panaudoti daugybę „motininių“ bangelių, tokių kaip Daubechies, Morlet'o, Le Gall'o arba Haar'o. Daubechies 9/7 (CDF 9/7) bangelės istoriškai yra pirmoji bangelių šeima, kuri buvo išpopuliarinta Ingridos Daubechies (1987 m. Ingrid Daubechies suformavo vieną iš pagrindinių bangelių bazių). CDF 9/7 bangelės yra ypač veiksmingos ortogonaliosios bangelės, plačiai naudojamos praktikoje (FTB pirštų atspaudų glaudinimas, vaizdų kodavimo standartas JPEG2000 ir pan.).

Šiame darbe yra aptariamoms ir algoritmizuojamos dvi CDF 9/7 transformacijos versijos, būtent: bazinė CDF 9/7 ir CDF 9/7 su daline blokų dekoreliacija. Darbo tikslas – atlikti palyginamąją bazinės CDF 9/7 ir CDF 9/7 su daline blokų dekoreliacija analizę vaizde sukauptos energijos „pakavimo“ spektrinėje bangelių srityje savybės kontekste. Palyginimo kriterijumi yra pasirenkamas dvimatis hiperbolinis skaitmeninių vaizdų filtras, t.y. apdorojamas vaizdas pervedamas į spektrų bangelių sritį, taikant abi CDF 9/7 transformacijos versijas. Gautieji spektrai apdorojami fiksuoto lygmens hiperboliniais filtrais. Po filtravimo įvertinama atkurtų vaizdų kokybė.

1 Diskrečiosios bangelių transformacijos (DBT)

Diskrečioji bangelių transformacija (DBT) yra viena naujausių priemonių, leidžiančių įveikti su Furjė transformacija (DFT) susijusias problemas (trūkumus), taip pat ir svarbiausią iš jų – lokalizavimo erdvėje problemą. Bangelių analizėje, apdorojant vieną ar kitą skaitmeninį vaizdą, natūraliai formuojamas kintamo dydžio (mastelio) langas, kuris slenka išilgai laiko (erdvės) ašies, ir kiekvienai lango pozicijai apskaičiuojamas bangelių spektras (transformacija). Procesas kartojamas daug kartų. Galutinis rezultatas – vaizdo išraiškų (vaizdavimų pagal dažnį ir erdvėje) rinkinys.

Diskrečioji bangelių transformacija atliekama skaičiaus 2 laipsnio skalėse ir postūmiais, lygiais skaičiaus 2 laipsniui. Transformacijos koeficientai dalomi į dvi lygias dalis: suvidurkinti koeficientai ir skirtuminiai koeficientai. Suvidurkinti koeficientai – didelio mastelio, žemo dažnio koeficientai. Skirtuminiai koeficientai – mažo mastelio, aukšto dažnio koeficientai. DBT gali būti ne vieno lygio, o iteracinė, t.y. DBT atliekama su gautais suvidurkintais koeficientais. Taip gauname žemesnio skirtuminio lygio vaizdo komponentus. Toks algoritmas vadinamas piramidiniu algoritmu. Transformacija gali būti atliekama iki tol, kol suvidurkintus koeficientus sudarys 2 koeficientai.

1.1 pav. Transformacijos schema

Vaizdo atkūrimo arba sintezės procesas matematiškai vadinamas atvirkštine diskrečiąja bangelių transformacija (ADBT). ADBT atliekama analogiškai DBT: suvidurkintus ir spektrinius koeficientus filtruojame, naudojant atitinkamus atkūrimo filtrus. Veiksmai kartojami, kol pasiekiamasis pirmas lygis.

1.1 Bendrosios bangelių savybės

Pradėsime nuo tolydžiosios bangelių transformacijos (TBT) apibrėžimo, būtent:

$$\Psi(x) = \Psi(sx + \tau), \quad (1.1)$$

kur Ψ yra bazinės funkcijos (bangelės). Kintamieji s ir τ (atitinkamai mastelis ir poslinkis) yra nauji bangelių transformacijos parametrai.

Atvirkštinė bangelių transformacija užrašoma taip:

$$\Psi(x) = \Psi\left(\frac{x - \tau}{s}\right). \quad (1.2)$$

Bangelės generuojamos, panaudojant bazinę („motininę“) bangelę Ψ , t.y.

$$\Psi(x) = \Psi\left(\frac{x - \tau}{s}\right). \quad (1.3)$$

Pastarojoje išraiškoje, s yra mastelių keičiantis parametras, ir τ – poslinkio parametras. Daugiklis $\frac{1}{|s|}$ yra normalizavimo parametras, įvertinant skirtingas mastelio parametrų reikšmes.

Kaip matome iš (1.1) išraiškos, vienmatės funkcijos (vaizdo) bangelių transformacija yra dvimatė, o dvimatės funkcijos bangelių transformacija, akivaizdu, yra keturmatė.

Aptarsime kai kurias pagrindines bangelių savybes. Bene svarbiausios iš jų – „tinkamumo“ ir „reguliarumo“ sąlygos. Sakoma, jog integruojama kvadratu funkcija Ψ , tenkina „tinkamumo“ sąlygą (savybę), jeigu

$$\int_{-\infty}^{\infty} |\Psi(x)|^2 dx < \infty. \quad (1.4)$$

Šioje išraiškoje Ψ žymi funkcijos Furjė transformaciją. Iš (1.4) nelygybės tiesiogiai išplaukia, kad Furjė transformacija prilygsta nuliui, kai dažnis yra lygus 0, t.y.

$$\Psi(0) = 0. \quad (1.5)$$

Tai reiškia, jog bangelės turi turėti dažnių juostą, apribotą pagal dažnį. Tuo pačiu bangelės (kaip vaizdo) pastovi dedamoji lygi nuliui, t.y.

(1.6)

Kitaip tariant, turi būti bangelė tikraja to žodžio prasme.

Bangelėms (bangelių funkcijoms) dažnai keliamos papildomos sąlygos, leidžiančios pagerinti bangelių skleidinių konvergavimo greitį. Šios sąlygos siejamos su bangelių „reguliarumo“ savybe. Pastaroji savybė suprantama kaip reikalavimas, jog bangelė būtų pakankamai glodi ir sukoncentruota tiek laiko, tiek dažnių srityse. „Reguliarumas“ yra gana sudėtinga sąvoka. Jos paaiškinimui ir interpretacijai pasinaudosime „nykstančiais“ pradiniais momentais.

Išskleidę bangelių transformaciją ((1.1) išraišką) Teiloro eilute taško $t = 0$ aplinkoje (paprastumo dėlei, imkime $\tau = 0$), gausime:

$$= - - ; \quad (1.7)$$

čia žymi p -osios eilės (funkcijos f) išvestinę, o - skleidinio liekamąjį narį. Toliau, pažymėję bangelės p -osios eilės pradinį momentą , t.y.

(1.8)

galime (1.7) išraišką perrašyti taip:

$$= \frac{\text{---}}{\text{---}} . \quad (1.9)$$

Remiantis „tinkamumo“ sąlyga, galime teigti, kad . Tokiu būdu, pirmasis (1.9) išraiškos narys lygus nuliui. Jeigu ir daugiau (tarkim, iki n -osios eilės) pradinių momentų prilygtų nuliui, skleidinio (bangelių transformacijos) koeficientai tolydžiam signalui

„gestų“ greičiu . Beje, pradiniai momentai nebūtinai turi būti lygūs nuliui - pakanka, jog jie būtų artimi nuliui.

Apibendrinant tai, kas buvo pasakyta, galima teigti, jog „tinkamumo“ sąlygos išpildymas sąlygoja bangelės svyravimus, o „reguliarumo“ sąlygos išpildymas bei nykstantys momentai – greitesnį bangelių skleidinių konvergavimą.

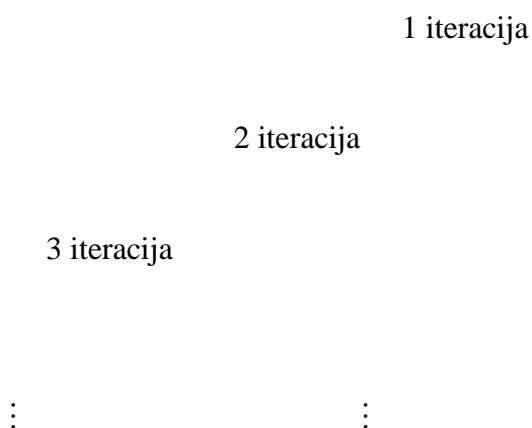
1.2 Iteracinės DBT apskaičiavimo procedūros

Kiekvienas diskrečiosios bangelių transformacijos (DBT) realizavimo žingsnis (iteracija) panaudoja taip vadinamą mastelio funkciją įvesties duomenims (skaitmeniniam vaizdui) apdoroti. Jeigu pradinis vaizdas X turi N reikšmių, tai mastelio funkcija bus panaudota tam, kad būtų apskaičiuotos $N/2$ suvidurkintos reikšmės. Gautosios reikšmės saugomos viršutinėje duomenų vektoriaus (iš N elementų) dalyje.

Kita (bangelių) funkcija, kiekviename DBT realizavimo žingsnyje, taip pat yra panaudojama įvesties duomenims apdoroti. Jei pradinis vaizdas X turi N reikšmių, tai bangelių funkcija bus pritaikyta tam, kad apskaičiuoti $N/2$ skirtumines (atspindinčias informacijos pasikeitimus vaizde X) reikšmes. Šios skirtuminės reikšmės yra saugomos apatinėje duomenų vektoriaus (iš N elementų) dalyje.

Kitoje iteracijoje (žingsnyje) abi funkcijos (mastelio ir bangelės) taikomos tikrai suvidurkintų reikšmių vektoriui, gautam prieš tai atliktoje iteracijoje. Po baigtinio iteracijų (žingsnių) skaičiaus iš skaitmeninio vaizdo X yra suformuojamas DBT spektras Y .

DBT spektras (vektorius) Y apima vienintelę suvidurkintą reikšmę (gautą n -tojoje iteracijoje) ir sutvarkytą skirtuminių reikšmių rinkinį (gautą ankstesnėse iteracijose).



1.2 pav. Piramidinio algoritmo schema

1.3 Diskrečioji Daubechies 9/7 transformacija (CDF 9/7): savybės, skaičiavimo algoritmai

Diskrečioji CDF 9/7 transformacija realizuojama taikant iteracines procedūras. Tarkime, turime pradinį duomenų vienmatį vaizdą X :

(1.10)

Po i – tosios iteracijos ()

gauname du tarpinius duomenų vektorius:

ir . (1.11)

Pastarųjų vektorių ir gavimui panaudojamos CDF 9/7 matricos, būtent:

$$D(1) = \begin{matrix} h_4 & h_3 & h_2 & h_1 \\ & g_3 & g_2 & \begin{pmatrix} h_0 & h_1 \\ g_1 & g_0 \end{pmatrix} \end{matrix} \begin{matrix} h_2 & h_3 & h_4 \\ g_1 & g_2 & g_3 \end{matrix}, \quad (1.12)$$

$$D(2) = \begin{matrix} h_4 & h_3 & h_2 & h_1 \\ & h_4 & h_3 & \begin{pmatrix} h_0 & h_1 & h_2 & h_3 \\ g_1 & g_0 & g_1 & g_2 \end{pmatrix} \\ & g_3 & g_2 & \begin{pmatrix} g_3 & g_2 & g_1 & g_0 \end{pmatrix} \end{matrix} \begin{matrix} h_4 \\ h_2 & h_3 & h_3 \\ g_1 & g_2 & g_3 \end{matrix}, \quad (1.13)$$

$$D(3) = \begin{matrix} h_4 & h_3 & h_2 & h_1 \\ & h_4 & h_3 & \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & h_4 & 0 & 0 & 0 \\ h_2 & h_1 & h_0 & h_1 & h_2 & h_3 & h_4 & 0 \\ h_4 & h_3 & h_2 & h_1 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & h_4 & h_3 & h_2 & h_1 & h_0 & h_1 \\ g_1 & g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 \\ g_3 & g_2 & g_1 & g_0 & g_1 & g_2 & g_3 & 0 \\ 0 & 0 & g_3 & g_2 & g_1 & g_0 & g_1 & g_2 \\ 0 & 0 & 0 & 0 & g_3 & g_2 & g_1 & g_0 \end{pmatrix} \end{matrix} \begin{matrix} h_4 \\ h_2 & h_2 & h_3 \\ g_1 & g_2 & g_3 \end{matrix}, \quad (1.14)$$

⋮

$$D(n) = \begin{matrix} h_4 & h_3 & h_2 & h_1 \\ & h_4 & h_3 & \\ & & g_3 & g_2 \end{matrix} \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & h_4 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ h_2 & h_1 & h_0 & h_1 & h_2 & h_3 & h_4 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & h_4 & h_3 & h_2 & h_1 & h_0 & h_1 & h_2 & h_3 & h_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & h_4 & h_3 & h_2 & h_1 & h_0 & h_1 & h_2 \\ g_1 & g_0 & g_1 & g_2 & g_3 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g_3 & g_2 & g_1 & g_0 & g_1 & g_2 & g_3 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & g_3 & g_2 & g_1 & g_0 & g_1 & g_2 & g_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & g_3 & g_2 & g_1 & g_0 & g_1 \end{pmatrix} \begin{matrix} h_4 \\ h_3 \\ h_2 \\ h_1 \\ h_0 \\ h_1 \\ h_2 \\ h_3 \end{matrix} \cdot \quad (1.15)$$

Aukščiau pateiktoms matricoms sudaryti panaudojame aukšto ir žemo dažnio filtrų koeficientus, t. y. h_0, h_1, h_2, h_3, h_4 ir g_0, g_1, g_2, g_3 (1.1 lentelė).

Žemo ir aukšto dažnio filtrai

1.1 lentelė.

h	Žemo dažnio filtras	g	Aukšto dažnio filtras
h_0	0,6029490182363579	g_0	1,11508705
h_1	0,2668641184428723	g_1	-0,591271763114
h_2	-0,07822326652898785	g_2	-0,057543526229
h_3	-0,01686411844287495	g_3	0,091271763114
h_4	0,026748741080976	-	

Ir čia, kaip matome, susiduriame su taip vadinama „krašto“ problema, nes matricos elementai išeina už jos ribų. Yra keletas „krašto“ problemos sprendimo būdų (periodinis ir kt.), tačiau vienas iš šiuo metu esančių efektyviausių „krašto“ problemos įveikimo būdų yra „veidrodinio atspindžio“ taikymas. Tai reiškia, kad per pirmąjį ir paskutinįjį matricos stulpelius taikome veidrodinį atspindį ir elementus išeinančius už matricos ribų pridedame prie elementų esančių matricos viduje iš eilės pradėdant nuo antrojo ir priešpaskutinio matricos stulpelių (nes per pirmąjį ir paskutinįjį, kaip minėjome, taikome veidrodinį atspindį). Atlikus veiksmus CDF9/7 matricos įgys tokius pavidalus:

$$D(1) = \begin{pmatrix} h_0 + 2h_2 + 2h_4 & 2h_1 + 2h_3 \\ 2g_1 + 2g_3 & g_0 + 2g_2 \end{pmatrix}, \quad (1.16)$$

$$D(2) = \begin{pmatrix} h_0 & 2h_1 & 2h_2 + 2h_4 & 2h_3 \\ h_2 + h_4 & h_1 + 2h_3 & h_0 + h_2 + h_4 & h_1 \\ g_1 & g_0 + g_2 & g_1 + 2g_3 & g_0 \\ 2g_3 & 2g_2 & 2g_1 & g_0 \end{pmatrix}, \quad (1.17)$$

$$D(3) = \begin{pmatrix} h_0 & 2h_1 & 2h_2 & 2h_3 & 2h_4 & 0 & 0 & 0 \\ h_2 & h_1 + h_3 & h_0 + h_4 & h_1 & h_2 & h_3 & h_4 & 0 \\ h_4 & h_3 & h_2 & h_1 & h_0 & h_1 & h_2 + h_4 & h_3 \\ 0 & 0 & h_4 & h_3 & h_2 + h_4 & h_1 + h_3 & h_0 + h_2 & h_1 \\ g_1 & g_0 + g_2 & g_1 + g_3 & g_2 & g_3 & 0 & 0 & 0 \\ g_3 & g_2 & g_1 & g_0 & g_1 & g_2 & g_3 & 0 \\ 0 & 0 & g_3 & g_2 & g_1 & g_0 & g_1 + g_3 & g_2 \\ 0 & 0 & 0 & 0 & 2g_3 & 2g_2 & 2g_1 & g_0 \end{pmatrix}, \quad (1.18)$$

Bendru atveju:

$$D(n) = \begin{pmatrix} h_0 & 2h_1 & 2h_2 & 2h_3 & 2h_4 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ h_2 & h_1 + h_3 & h_0 + h_4 & h_1 & h_2 & h_3 & h_4 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & h_4 & h_3 & h_2 & h_1 & h_0 & h_1 & h_2 + h_4 & h_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & h_4 & h_3 & h_2 + h_4 & h_1 + h_3 & h_0 + h_2 & h_1 \\ g_1 & g_0 + g_2 & g_1 + g_3 & g_2 & g_3 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g_3 & g_2 & g_1 & g_0 & g_1 & g_2 & g_3 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & g_3 & g_2 & g_1 & g_0 & g_1 + g_3 & g_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 2g_3 & 2g_2 & 2g_1 & g_0 \end{pmatrix}. \quad (1.19)$$

Atlikus n iteracijų gaunamas (suformuojamas) CDF 9/7 spektras Y vaizdui X , būtent:

$$Y = D \cdot X, \quad (1.20)$$

Matome, jog viršutiniuose skliausteliuose įrašyti skaičiai (kintantys nuo 1 iki n) žymi, kurios iteracijos metu buvo gauti atitinkami spektriniai koeficientai.

Diskrečiąją CDF 9/7 transformaciją galima užrašyti matriciniu pavidalu:

$$Y = D \cdot X, \quad (1.21)$$

kur D yra CDF 9/7 transformacijos matrica, nusakoma išraiška

$$D = \prod_{i=1}^n D^*(i), \quad (1.22)$$

$$D^*(1) = \begin{pmatrix} D(1) & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & E(n-1) \end{pmatrix}, \quad (1.23)$$

$$D^*(2) = \begin{pmatrix} D(2) & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & E(n-2) \end{pmatrix}, \quad (1.24)$$

$$\vdots$$

$$D^*(i) = \begin{pmatrix} D(i) & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & E(n-i) \end{pmatrix}, \quad (1.24)$$

Su visais $i = 1, 2, 3, \dots, n-1$, $E(0) = 0$,

ir

$$D^*(n) = D(n). \quad (1.25)$$

Norint atkurti pradinį vaizdą reikia taikyti atvirkštinę CDF 9/7 transformaciją (ACDF 9/7). Bendru atveju, ACDF 9/7 matricos turi tokį pavidalą:

$$AD(1) = \begin{pmatrix} -h_3 & g_2 & -h_1 \begin{pmatrix} g_0 & -h_1 \end{pmatrix} g_2 & -h_3 \\ h_4 & -g_3 & h_2 \begin{pmatrix} -g_1 & h_0 \end{pmatrix} -g_1 & h_2 & -g_3 & h_4 \end{pmatrix}, \quad (1.26)$$

$$AD(2) = \begin{pmatrix} -h_3 & g_2 & -h_1 \begin{pmatrix} g_0 & -h_1 & g_2 & -h_3 \end{pmatrix} \\ h_4 & -g_3 & h_2 \begin{pmatrix} -g_1 & h_0 & -g_1 & h_2 \end{pmatrix} -g_3 & h_4 \\ -h_3 & g_2 & -h_1 \begin{pmatrix} g_0 & -h_1 & g_2 & -h_3 \end{pmatrix} g_2 & -h_3 \\ h_4 & -g_3 & h_2 \begin{pmatrix} -g_1 & h_0 & -g_1 & h_2 \end{pmatrix} -g_1 & h_2 & -g_3 & h_4 \end{pmatrix}, \quad (1.27)$$

$AD(3) =$

$$\begin{array}{c}
 -h_3 \quad g_2 \quad -h_1 \\
 h_4 \quad -g_3 \quad h_2 \\
 -h_3 \\
 h_4
 \end{array}
 \left(\begin{array}{cccccccc}
 g_0 & -h_1 & g_2 & -h_3 & 0 & 0 & 0 & 0 \\
 -g_1 & h_0 & -g_1 & h_2 & -g_3 & h_4 & 0 & 0 \\
 g_2 & -h_1 & g_0 & -h_1 & g_2 & -h_3 & 0 & 0 \\
 -g_3 & h_2 & -g_1 & h_0 & -g_1 & h_2 & -g_3 & h_4 \\
 0 & -h_3 & g_2 & -h_1 & g_0 & -h_1 & g_2 & -h_3 \\
 0 & h_4 & -g_3 & h_2 & -g_1 & h_0 & -g_1 & h_2 \\
 0 & 0 & 0 & -h_3 & g_2 & -h_1 & g_0 & -h_1 \\
 0 & 0 & 0 & h_4 & -g_3 & h_2 & -g_1 & h_0
 \end{array} \right)
 \begin{array}{c}
 -g_3 \quad h_4 \\
 g_2 \quad -h_3 \\
 -g_1 \quad h_2 \quad -g_3 \quad h_4
 \end{array}, \quad (1.28)$$

⋮

$AD(n) =$

$$\begin{array}{c}
 -h_3 \quad g_2 \quad -h_1 \\
 h_4 \quad -g_3 \quad h_2 \\
 -h_3 \\
 h_4 \\
 \vdots \\
 0 \\
 0 \\
 0
 \end{array}
 \left(\begin{array}{cccccccccccccccc}
 g_0 & -h_1 & g_2 & -h_3 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -g_1 & h_0 & -g_1 & h_2 & -g_3 & h_4 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 g_2 & -h_1 & g_0 & -h_1 & g_2 & -h_3 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -g_3 & h_2 & -g_1 & h_0 & -g_1 & h_2 & -g_3 & h_4 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & h_4 & -g_3 & h_2 & -g_1 & h_0 & -g_1 & h_2 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & -h_3 & g_2 & -h_1 & g_0 & -h_1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & h_4 & -g_3 & h_2 & -g_1 & h_0
 \end{array} \right)
 \begin{array}{c}
 -g_3 \quad h_4 \\
 g_2 \quad -h_3 \\
 -g_1 \quad h_2 \quad -g_3 \quad h_4
 \end{array}$$

(1.29)

Išsprendus “krašto” problemą gautume tokią matricą (pažymėkime ją $\hat{D}(n)$):

$$\hat{D}(1) = \begin{pmatrix} g_0 + 2g_2 & -2h_1 - 2h_3 \\ -2g_1 - 2g_3 & h_0 + 2h_2 + 2h_4 \end{pmatrix}, \quad (1.30)$$

$$\hat{D}(2) = \begin{pmatrix} g_0 & -2h_1 & 2g_2 & -2h_3 \\ -g_1 & h_0 + h_2 + h_4 & -g_1 - 2g_3 & h_2 + h_4 \\ g_2 & -h_1 - 2h_3 & g_0 + g_2 & -h_1 \\ -2g_3 & 2h_2 + 2h_4 & -2g_1 & h_0 \end{pmatrix}, \quad (1.31)$$

$$\hat{D}(3) = \begin{pmatrix} g_0 & -2h_1 & 2g_2 & -2h_3 & 0 & 0 & 0 & 0 \\ -g_1 & h_0 + h_2 & -g_1 - g_3 & h_2 + h_4 & -g_3 & h_4 & 0 & 0 \\ g_2 & -h_1 - h_3 & g_0 & -h_1 & g_2 & -h_3 & 0 & 0 \\ -g_3 & h_2 + h_4 & -g_1 & h_0 & -g_1 & h_2 & -g_3 & h_4 \\ 0 & -h_3 & g_2 & -h_1 & g_0 & -h_1 & g_2 & -h_3 \\ 0 & h_4 & -g_3 & h_2 & -g_1 & h_0 + h_4 & -g_1 - g_3 & h_2 \\ 0 & 0 & 0 & -h_3 & g_2 & -h_1 - h_3 & g_0 + g_2 & -h_1 \\ 0 & 0 & 0 & 2h_4 & -2g_3 & 2h_2 & -2g_1 & h_0 \end{pmatrix}, \quad (1.32)$$

⋮

$$\hat{D}(n) = \begin{pmatrix} g_0 & -2h_1 & 2g_2 & -2h_3 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -g_1 & h_0 + h_2 & -g_1 - g_3 & h_2 + h_4 & -g_3 & h_4 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g_2 & -h_1 - h_3 & g_0 & -h_1 & g_2 & -h_3 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -g_3 & h_2 + h_4 & -g_1 & h_0 & -g_1 & h_2 & -g_3 & h_4 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & h_4 & -g_3 & h_2 & -g_1 & h_0 + h_4 & -g_1 - g_3 & h_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & -h_3 & g_2 & -h_1 - h_3 & g_0 + g_2 & -h_1 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 2h_4 & -2g_3 & 2h_2 & -2g_1 & h_0 \end{pmatrix} \quad (1.33)$$

Pritaikius atitinkamo dydžio matricą gautajam vaizdo spektrui, atlikus tam tikrą iteracijų skaičių galime atkurti pradinį vaizdą. Vaizdo atkūrimo procedūra tokia:

$$S^{(n-1)} = D^{-1}(1) \cdot \begin{pmatrix} S^{(n)} \\ D^{(n)} \end{pmatrix}, \quad (1.34)$$

$$S^{(n-2)} = D^{-1}(2) \cdot \begin{pmatrix} S^{(n-1)} \\ D^{(n-1)} \end{pmatrix}, \quad (1.35)$$

$$S^{(n-3)} = D^{-1}(3) \cdot \begin{pmatrix} S^{(n-2)} \\ D^{(n-2)} \end{pmatrix}, \quad (1.36)$$

$$\vdots$$

$$S^{(i)} = D^{-1}(n-i) \cdot \begin{pmatrix} S^{(i+1)} \\ D^{(i+1)} \end{pmatrix}, \quad (1.37)$$

$$\vdots$$

$$S^{(0)} = D^{-1}(n) \cdot \begin{pmatrix} S^{(1)} \\ D^{(1)} \end{pmatrix} \equiv X. \quad (1.38)$$

Bazinę CDF 9/7 transformacijos apskaičiavimo procesą geriausiai iliustruoja jos signalinis grafas (1.3 pav.). Tarkime, turime duomenų vektorių $N=16$. Tuomet pirmosios iteracijos metu taikysime $D(4)$ matricą (16x16). Po pirmosios iteracijos žemo dažnio filtro pagalba gautam vektoriui jau taikysime $D(3)$ matricą, tuomet po antrosios iteracijos gautam duomenų vektoriui taikysime $D(2)$ matricą ir taip toliau, kol galiausiai liks pritaikyti $D(1)$ matricą ir gauti spektrą Y . Iteracijų skaičius, savaime aišku, priklauso nuo to, kokio dydžio bus pradinis duomenų vektorius.

1 iteracija

2 iteracija

3 iteracija

4 iteracija

1.3 pav. Bazinės CDF 9/7 transformacijos signalinis grafas

1.4 CDF 9/7 su daline blokų dekoreliacija

Norint rasti CDF 9/7 spektrą su daline blokų dekoreliacija visų pirma daliname duomenų vektorių X į blokelių po m elementų; čia $m < n$. Tuomet, kiekvienam blokeliui atskirai taikome CDF 9/7 matricas (priklausomai nuo pasirinkto blokelių dydžio). Procedūra bus taikoma m kartų. Po to, likusiose $(n - m)$ iteracijose panaudojama CDF 9/7 matrica $D(1)$ (1.4 pav.)

Rezultatas – CDF 9/7 su daline dekoreliacija spektras vaizdui X .

Tarkime turime duomenų vektorių $N=16$ ir daliname jį į du blokelių. Tuomet, kiekvienam blokeliui taikysime atitinkamai CDF 9/7 matricas. Kadangi padalinus blokelių į dvi dalis lieka du duomenų vektoriai po 8, tai ir matricas taikysime pirmojoje iteracijoje $D(3)$ abiem blokeliams atskirai. Toliau, antrojoje iteracijoje jau taikysime matricą $D(2)$ ir trečioje bei ketvirtoje iteracijose matricą $D(1)$. Taikyti dalinę blokų dekoreliaciją yra labai patogiu, kai reikia iširti tik dalį duomenų, o nebūtinai visą duomenų masyvą.

1 iteracija

2 iteracija

3 iteracija

4 iteracija

1.4 pav. CDF 9/7 su daline blokų dekoreliacija signalinis grafas, kai $m=3$

1 iteracija 2 iteracija 3 iteracija 4 iteracija

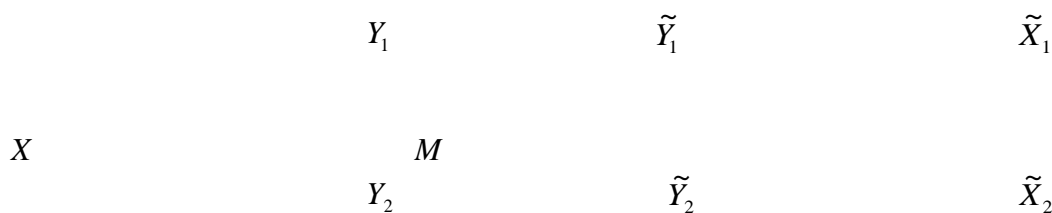
1.5 pav. CDF 9/7 su daline blokų dekoreliacija signalinis grafas, kai $m=2$

Kaip matome (1.5 pav.), CDF 9/7 su daline blokų dekoreliacija transformacijos realizacija, kai $N=16$ ir $m=2$. Pirmojoje iteracijoje bus naudojama matrica $D(2)$ (kiekvienam blokui), antrojoje ir likusiose iteracijose jau bus taikoma tik $D(1)$ (kiekvienam blokeliui atskirai) tol, kol bus gautas spektras.

2 Bazinės CDF 9/7 ir CDF 9/7 su daline blokų dekoreliacija palyginamoji analizė.

Tam, kad galėtume išsiaiškinti, kaip dalinė blokų dekoreliacija įtakoja kitas CDF 9/7 transformacijų savybes (būtent energijos „pakavimo“ savybę), buvo pasirinkta tyrimo schema, kurioje pagrindinis vaidmuo tenka hiperboliniams vaizdų filtrams (žr. 2.1 skyrelis 2.1 pav.). Būtent hiperboliniai vaizdų spektrai leidžia daryti išvadas apie tai, kaip greitai „gęsta“ apdorojamą vaizdą sudarančios aukšto dažnio harmonikos (tai ir apibūdina energijos „pakavimo“ savybes).

2.1 Bendra schema



2.1 pav. Bendroji schema.

Tarkime, jog turime kokį nors dvimatį vaizdą $[X(m_1, m_2)]$, jam taikome bazinę CDF 9/7 ir CDF 9/7 su daline blokų dekoreliacija, gauname spektrus Y_1 ir Y_2 , jiems panaudojame hiperbolinį filtrą su fiksuotu lygmeniu M , tokiu būdu gauname \tilde{Y}_1 ir \tilde{Y}_2 . Kad galėtumėme atkurti vaizdą dar reikia panaudoti atvirkštinę bazinę CDF 9/7 ir atitinkamai atvirkštinę CDF 9/7 su daline blokų dekoreliacija, taip galiausiai gauname \tilde{X}_1 ir \tilde{X}_2 . Gavus atkurtus vaizdus įvertiname vidutinės kvadratinės paklaidas $\delta_1 = \delta(X, \tilde{X}_1)$, $\delta_2 = \delta(X, \tilde{X}_2)$. Čia

$$\delta = \left(\frac{1}{N^2} \sum_{m_1, m_2=0}^{N-1} (X(m_1, m_2) - \tilde{X}(m_1, m_2))^2 \right)^{\frac{1}{2}} \quad (2.1)$$

2.1.1 Dvimačių hiperbolinių filtrų kontekstas

Tegu M yra tam tikras teigiamas skaičius ($1 \leq M \leq (N-1)^2$). Tarkime, kad $H(M) = \{(k_1, k_2) \mid \bar{k}_1 \cdot \bar{k}_2 \leq M\}$, t. y. $H(M)$ yra dvimačio masyvo $N \times N$ eilučių ir stulpelių sutvarkytų eilės numerių porų (k_1, k_2) aibė tokia, kad $\bar{k}_1 \cdot \bar{k}_2 \leq M$. Aišku, jog aibės $H(M)$ kraštas plokštumoje turi hiperbolės pavidalą.

Hiperbolinis vaizdų filtravimas remiasi tuo, jog žmogaus akis nėra labai jautri aukštomis vaizdą sudarančioms harmonikoms. Atliekant hiperbolinį dvimačio skaitmeninio vaizdo filtravimą, spektriniai CDF 9/7 koeficientai $Y(k_1, k_2)$, kurių indeksai $(k_1, k_2) \notin H(M)$, yra atmetami. Kadangi pašalintų harmonikų nereikia išsaugoti, gaunamas suspaudimo efektas.

Atkuriant pradinį vaizdą $[X(m_1, m_2)]$, atmetieji spektriniai koeficientai keičiami nuliais. Tai daryti galima ir tikslinga dėl dviejų priežasčių:

- atmetami spektriniai koeficientai yra nykstamai maži, todėl juose kaupiama informacija apie vaizdą $[X(m_1, m_2)]$ yra neesminė;
- šie spektriniai koeficientai atspindi vaizdą $[X(m_1, m_2)]$ sudarančias aukšto dažnio dedamąsias, kurioms žmogaus akis mažai jautri.

Viena šio metodo ypatybių – hiperboliniai filtrai gali dirbti tiek vienmatėje, tiek dvimatėje, tiek trimatėje erdvėje.

Pastebėsime, jog hiperboliniam vaizdo filtravimui gali būti naudojama bet kuri diskrečioji transformacija (DT), duodanti kompaktišką nenulinių spektrinių koeficientų išsidėstymą (kosinusinė (DKT), Volšo ir Adamaro (VAT), Haaro (HT) bei Daubechies 9/7 (CDF 9/7)).

$$\begin{array}{ccc}
 N-1 & k_2 & (0 < M < N-1) & & N-1 & k_2 & (N-1 < M < (N-1)^2) \\
 \\
 M & & H(M) & & & & H(M) \\
 \\
 & & & k_1 & & & k_1 \\
 0 & & M & N-1 & 0 & & N-1
 \end{array}$$

2.2 Pav. Hiperbolinių dvimačių skaitmeninių vaizdų filtravimas

Spektrinių koeficientų keitimo nuliais procedūrą patogiau realizuoti įvedant M lygio hiperbolinius filtrus $H_{\omega}(M, m_1, m_2)$. Pastarojo filtro dvimatis CDF 9/7 spektras užrašomas taip:

$$\hat{H}_{\omega}(M, k_1, k_2) = \begin{cases} 1, & \text{kai } (k_1, k_2) \in H(M), \text{ t.y. } \bar{k}_1 \cdot \bar{k}_2 \leq M, \\ 0, & \text{kai } (k_1, k_2) \notin H(M), \text{ t.y. } \bar{k}_1 \cdot \bar{k}_2 > M. \end{cases} \quad (2.2)$$

Čia: $\bar{k}_i = \max\{k_i, 1\}, i \in \{1, 2\}$; M - hiperbolinio filtro lygis dvimatėje erdvėje, $1 \leq M \leq N^2$.

Daugiamačiams vaizdams diskretieji spektrai apskaičiuojami pakartotinai taikant atitinkamas vienmates transformacijas (pagal kiekvieną erdvinę vaizdo koordinatę atskirai).

Imkime dvimatį skaitmeninį vaizdą $[X(m_1, m_2)]$, jo dvimatį diskretųjį spektrą pažymėsime $[Y(k_1, k_2)]$. Hiperbolinio vaizdų filtro veikimas nusakomas išraiška:

$$\tilde{Y}(k_1, k_2) = \begin{cases} Y(k_1, k_2), & \bar{k}_1 \cdot \bar{k}_2 \leq M, \\ 0, & \bar{k}_1 \cdot \bar{k}_2 > M. \end{cases} \quad (2.3)$$

$[X(m_1, m_2)]$

$[Y(k_1, k_2)]$

$[\tilde{Y}(k_1, k_2)]$

$[\tilde{X}(m_1, m_2)]$

2.3. pav. Hiperbolinio vaizdų filtro veikimo schema

Kaip matome, spektriniai koeficientai, kurių indeksų sandauga didesnė už filtro lygį, yra atmetami (suspaudimo efektas). Atkuriant vaizdą, atmestieji spektriniai koeficientai keičiami nuliais – gaunamas vaizdo įvertis.

Aprašytoji hiperbolinio vaizdų filtravimo schema gali būti taikoma vienmačiams, dvimačiams bei trimačiams vaizdams atitinkamai parenkant vienmatę, dvimatę ar trimatę diskrečiąsias transformacijas.

Hiperbolinis vaizdų filtras leidžia suspausti vaizdą 5-10 kartų, kartu išsaugant gana neblogą atkurto vaizdo (vizualinę) kokybę. Svarbiausias metodo privalumas yra tas, jog galima spausti įvairiamačius vaizdus, ir vaizdo suspaudimo efektyvumas stipriai susijęs su vaizdo glodumu.

2.1.2 Filtravimo paklaidų įvertinimas

Tarkime, jog turime dvimatį vaizdą $[X(m_1, m_2)]$, apskaičiuojame šiam signalui CDF 9/7 spektrus. Spektriniai koeficientai (realiems duomenims) absoliutiniu didumu turi tendenciją mažėti, didėjant jų eilės numeriams. Šį faktą išnaudojame atlikdami hiperbolinį apdorojamo signalo filtravimą spektrinėje Daubechies 9/7 srityje. Hiperbolinio filtravimo esmė tokia – prieš taikant vienokią ar kitokią atvirkštinę diskrečiąją transformaciją, tam tikra dalis aukštas harmonikas signale atitinkančių spektrinių koeficientų keičiama nuliais (nufiltruojama). Likusieji spektriniai koeficientai išsaugomi.

Įvertinus paklaidas, nustatome, kurios diskrečiosios transformacijos (CDF 9/7 ir CDF 9/7 su daline dekoreliacija) geriau atkuria duomenis.

Paklaidoms įvertinti naudosime tokias išraiškas (tegu - pradinis signalas):

Vidutinė kvadratinė paklaida:

$$\delta = \delta(X, \tilde{X}) = \left(\frac{1}{N^2} \sum_{m_1, m_2=0}^{N-1} (X(m_1, m_2) - \tilde{X}(m_1, m_2))^2 \right)^{\frac{1}{2}} \quad (2.3)$$

čia - atvirkštinės DBT taikymo nufiltruotam diskrečiajam spektrui rezultatas, t.y. - pradinio vaizdo X įvertis.

Signalų ir triukšmo santykio maksimali reikšmė

$$PSNR = 20 \cdot \lg\left(\frac{255}{\delta}\right). \quad (2.4)$$

2.2 Programinės realizacijos ypatumai

Eksperimentas (aprašytas 3 skyrelyje) atliekamas programos Matlab aplinkoje (programos tekstai pateikti darbo priede). Programa suranda:

- CDF 9/7 spektrą vienmačiu ir dvimačiu atvejais;
- CDF 9/7 vaizdą vienmačiu ir dvimačiu atvejais;
- CDF 9/7 su daline blokų dekoreliaciją spektrą vienmačiu ir dvimačiu atvejais;
- CDF 9/7 su daline blokų dekoreliaciją vaizdą vienmačiu ir dvimačiu atvejais;
- Pagrindinė programa apskaičiuoja paklaidas atlikus eksperimentą su bazine CDF 9/7 ir su CDF 9/7 su daline blokų dekoreliacija hiperbolinių filtrų kontekste.

Programoje priklausomai nuo nagrinėjamo paveiksliuko dydžio naudojami suspaudimo koeficientai (žr. 2.1 lentelę). Programoje būtent naudojamos M reikšmės, greta parašyta kokio lygio suspaudimas (β). Eksperimente lyginami trys vienodo dydžio paveikslėliai (Barbara512.png., Hill512.png. ir Lake512.png.). Kuomet vaizdas yra kitokio dydžio, jam atitinkamai naudojami kiti suspaudimo koeficientai. 2.1 lentelėje pateikti suspaudimo koeficientai paveikslėliams, kurių dydžiai 512×512 ir 256×256 .

Programa veiksmus atlieka gana greitai (1-3 sekundės). Kiek ilgiau užtrunka atkurdamą vaizdus, kai pasirenkamas mažas blokelių dydis (7-15 sekundžių).

Suspaudimo koeficiento β reikšmės

2.1 lentelė

Vaizdo dydis	512×512	Vaizdo dydis	256×256
M	β	M	β
261121	1	65025	1
48687	2	12102	2
26366	3	6544	3
17569	4	4349	4
12959	5	3199	5
10163	6	2500	6
8298	7	2039	7
6972	8	1709	8
5991	9	1463	9
5235	10	1273	10
...
525	60	151	60
...

PC, kuriuo atliekamas darbas parametrai: *Microsoft Windows XP 2002*.

3 Eksperimento rezultatų analizė

1. **Pvz.:** Eksperimentas atliekamas su paveikslėliu Barbara512.png.



3.1 pav. Barbara512.png.

Atkreipkime dėmesį, kad šis vaizdas glodumo prasme nėra glodus. Viso nagrinėsime 3 paveikslėlius (Barbara512.png., Hill 512.png., Lake512.png), todėl paklaidas, gautas atkuriant vaizdus, vėliau galėsime palyginti ir įvertinti, kokią reikšmę vaizdo atkūrimui turi vaizdo glodumas.

Atlikus eksperimentą su aukščiau matomu 3.1. pav. Barbara512.png. gauname tokią paklaidų lentelę:

Palyginamoji 3.1 pav. paklaidų lentelė

3.2 lentelė

Vaizdo suspaudimo koef. B	CDF 9/7	CDF 9/7 su daline dekoreliacija					
		Blokelio dydis, m					
		4	5	6	7	8	9
2	11,4084	11,7745	11,5544	11,4615	11,4222	11,4123	11,4084
3	13,2488	13,7624	13,4091	13,3444	13,2973	13,2817	13,2488
4	14,2162	14,6693	14,4446	14,3763	14,2974	14,2808	14,2162
5	14,8852	15,5190	15,1857	15,0619	14,9797	14,9435	14,8852
6	15,2401	16,0565	15,6457	15,4647	15,3475	15,2991	15,2401
7	15,4885	16,4182	15,9781	15,7344	15,6100	15,5421	15,4885
8	15,7816	16,7475	16,3205	16,0326	15,9033	15,8326	15,7816
9	16,2324	17,1939	16,7712	16,5022	16,3781	16,2843	16,2324
10	16,4858	17,5392	17,0682	16,7693	16,6705	16,5627	16,4858

Kaip matome iš 3.1 lentelės, didinant blokelių dydį m paklaidos mažėja, kol pasidaro lygios bazinės CDF 9/7 transformacijos paklaidų reikšmėms (kai $m=9$). Taip pat atkreipkime dėmesį, kad paklaidos išties nemažos, taip yra dėl to, kad pasirinktas paveikslėlis glodumo prasme nėra glodus. Taigi, kuo mažiau glodų paveikslėlių pasirinksime, tuo paklaidos bus didesnės.

Nagrinėkime atvejį, kai $\beta=5$, o m kinta. Kai $m=9$, bus pasiekiamas vaizdo atkūrimo efektas kaip ir taikant bazinę CDF 9/7 transformaciją.

 $\beta=9, m=2, \delta=17,1219$  $\beta=9, m=4, \delta=17,1939$

Matome, kad kai suspaudimo koeficientas $\beta=9$, o blokelių dydis $m=2$ vaizdo atkūrimo kokybė nėra labai gera, vaizdas šiek tiek liejasi, tačiau kontūrai aiškūs, pakankamai gerai matomi daiktai. Pakeitus blokelių dydį $m=4$ vaizdo kokybė pagerėja, vaizdas ryškesnis.


 $\beta=9, m=7, \delta=16,3781$

 $\beta=9, m=9, \delta=16,2324$

Toliau turime atvejį, kai suspaudimo koeficientas $\beta=9$, o blokelių dydis $m=7$. Lyginant su prieš tai gautais vaizdais (kai $\beta=9, m=2$ ir $\beta=9, m=4$) vaizdo kokybė ženkliai geresnė. O kai parenkame suspaudimo koeficientą $\beta=9$ ir blokelių dydį $m=9$ atkurto vaizdo kokybė tokia pati kaip ir taikant bazinę CDF 9/7 transformaciją.

Kitas nagrinėjamas atvejis, kai suspaudimo koeficientas $\beta=15$, o blokelių dydį keičiame. Eksperimentas atliekamas su tuo pačiu paveikslėliu Barbara512.png.


 $\beta=15, m=2, \delta=18,4885$

 $\beta=15, m=4, \delta=18,8811$

Padidinus suspaudimo koeficientą akivaizdžiai pablogėja atkurto vaizdo kokybė. Ir nors didinant blokelių dydį vaizdas atkuriamas aiškesnis, kokybiškesnis, tačiau kur kas blogesnės kokybės lyginant su pradiniu vaizdu (žr. 3.1 Pav. Barbara512.png.). Taigi, kuo didesnis suspaudimo koeficientas β , tuo atkuriamo vaizdo kokybė yra blogesnė.


 $\beta=15, m=7, \delta=17,5742$

 $\beta=15, m=9, \delta=17,2433$

2. **Pavyzdys.** Tegų blokelių dydis būna pastovus $m = 5$, o keiskime suspaudimo koeficientą β . Paveikslėlis Hill512.png.



3.2 pav. Hill512.png.

Palyginamoji 3.2 pav. paklaidų lentelė

3.2 lentelė

Vaizdo suspaudimo koef. β	CDF 9/7	CDF 9/7 su daline dekokoreliacija					
		Blokelių dydis, m					
		4	5	6	7	8	9
2	3,7259	4,1302	3,9214	3,8137	3,7642	3,7374	3,7259
3	5,1505	5,7889	5,4499	5,2852	5,2115	5,1663	5,1505
4	6,0968	7,0441	6,5490	6,2929	6,1857	6,1166	6,0968
5	6,7518	7,8878	7,3279	7,0099	6,8566	6,7796	6,7518
6	7,3822	8,6095	8,0414	7,7022	7,5257	7,4175	7,3822
7	7,8838	9,2322	8,6928	8,2811	8,0615	7,9200	7,8838
8	8,2941	9,7088	9,1474	8,7253	8,4914	8,3413	8,2941
9	8,6990	10,201	9,6953	9,2057	8,9305	8,7475	8,6990
10	9,0727	10,6436	10,1146	9,5969	9,3172	9,1238	9,0727

Čia taip pat matome didinant blokelių dydį paklaidos mažėja, kol pasidaro lygios bazinės CDF 9/7 transformacijos paklaidų reikšmėms. Kadangi šis paveikslėlis yra kur kas

glodesnis už prieš tai nagrinėtą Barbara512.png (3.1 pav.), tai ir paklaidos gaunamos ženkliai mažesnės.



$m=5, \beta=5, \delta=7,3279$



$m=5, \beta=7, \delta=8,6928$

Čia nagrinėjamas atvejis, kai blokelių dydis m yra pastovus, o keičiamas suspaudimo koeficientas β . Palyginimui eksperimentas atliekamas su šešiomis suspaudimo koeficiento β reikšmėmis. Akivaizdu, jog didinant suspaudimo koeficientą atkuriamo vaizdo kokybė ženkliai blogėja. Čia $m=5$, o 3.2 lentelėje galime pamatyti kaip didėja paklaidos keičiant suspaudimo koeficientą $\beta=2, \dots, 10$. Eksperimentas atliktas ir kai $\beta=11, 13, 21$, kad vizualiai taip pat būtų galima įvertinti atkuriamo vaizdo kokybės blogėjimą.



$m=5, \beta=9, \delta=9,6953$

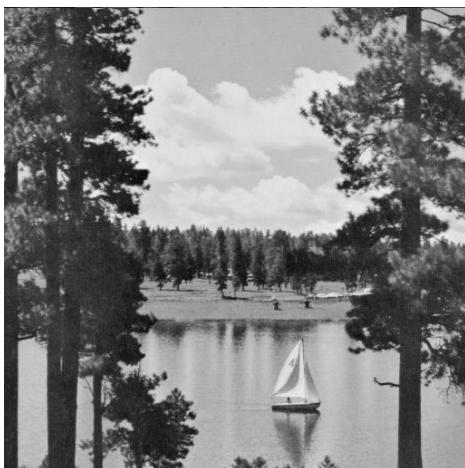


$m=5, \beta=11, \delta=10,4199$


 $m=5, \beta=13, \delta=10,9943$

 $m=5, \beta=21, \delta=12,8106$

3. Pavyzdys. Nagrinėjamas paveikslėlis Lake512.png.



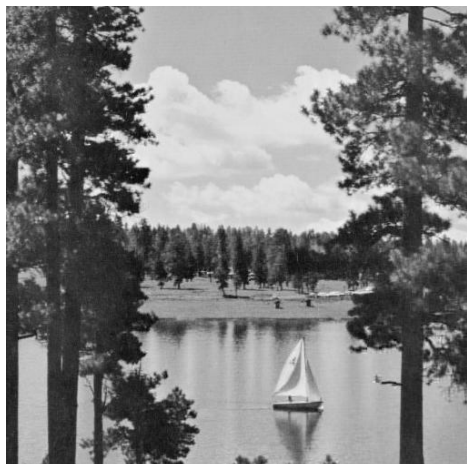
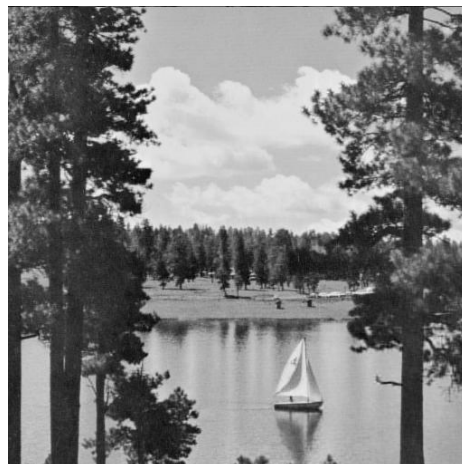
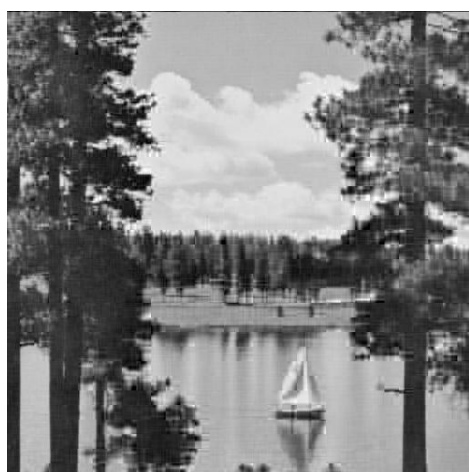
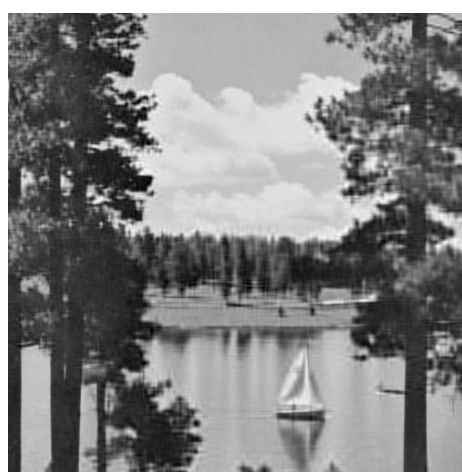
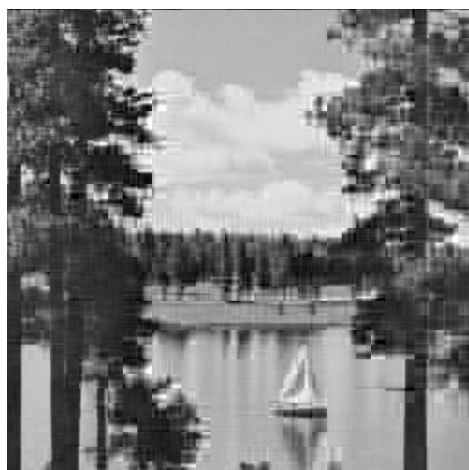
3.3 pav. Lake512.png.

Palyginamoji 3.3 pav. paklaidų lentelė

3.3 lentelė

Vaizdo suspaudimo koef. β	CDF 9/7	CDF 9/7 su daline dekoreliacija					
		Blokelių dydis, m					
		4	5	6	7	8	9
2	4,6737	5,1176	4,9021	4,7725	4,7403	4,6906	4,6737
3	6,3878	7,2493	6,8002	6,5805	6,4986	6,4177	6,3878
4	7,9687	9,2260	8,5710	8,2799	8,1228	8,0145	7,9687
5	8,8196	10,3747	9,5800	9,1861	9,0155	8,8698	8,8196
6	9,8318	11,7856	10,8694	10,3062	10,1176	9,9080	9,8318
7	10,7407	12,9583	11,9635	11,3043	11,0904	10,8493	10,7407
8	11,1621	13,5449	12,4935	11,7707	11,5436	11,2708	11,1621
9	11,8487	14,4025	13,3040	12,5011	12,2565	11,9477	11,8487
10	12,5096	15,1712	14,0294	13,2306	12,9566	12,6113	12,5096

Šis paveikslukas glodumu panašus į 3.2 pav., todėl ir paklaidos gaunamos artimesnės


 $\beta=2, m=4, \delta=5,1176$

 $\beta=2, m=9, \delta=4,6737$

 $\beta=7, m=4, \delta=12,9583$

 $\beta=7, m=9, \delta=10,7407$

 $\beta=15, m=4, \delta=17,7147$

 $\beta=15, m=9, \delta=14,7689$

4. Pavyzdys.

Ištirkime, kaip kinta CDF 9/7 transformacijos su daline blokų dekoreliacija paklaidos priklausomai nuo suspaudimo koeficiento β arba blokelių dydžio m . Imkime atvejį, kai β

pastovus ir kinta blokelių dydis m . Kadangi paveiksluko dydis 512×512 , tai $m = 1, \dots, 9$. Kada $m = 9$, tai bazinės CDF 9/7 ir CDF 9/7 su daline blokų dekoreliacija paklaidos sutampa.

3.4 pav. CDF 9/7 su daline blokų dekoreliacija paklaidos ($\beta = 5$, o m – kintantis).

Iš grafiko matome, kai $m > 3$ CDF 9/7 su daline blokų dekoreliacija paklaidos labai artimos bazinės CDF 9/7 paklaidoms, kitaip sakant, atkuriamo vaizdo kokybė labai panaši tiek taikant bazinę CDF 9/7, tiek CDF 9/7 su daline blokų dekoreliacija. Taigi, galime daryti išvadą, kad energijos „pakavimo“ savybė yra tenkinama.

Tirkime atvejį, kai vaizdas yra dar labiau suspaustas – $\beta = 10$, $m = 1, \dots, 9$.

3.5 pav. CDF 9/7 su daline blokų dekoreliacija paklaidos ($\beta=10$, o m – kintantis).

Iš grafiko matome, kad kol $m < 4$ paklaidos didėja, o kai $m > 4$ paklaidos tendencingai mažėja. Kadangi suspaudimo koeficientas β dvigubai didesnis, nei nagrinėtame 3.4 pav., tai ir

paklaidos didesnės. Tačiau bendru atveju, bazinės CDF 9/7 transformacijos ir CDF 9/7 transformacijos su daline blokų dekoreliacija atkuriamų vaizdų kokybė išlieka panaši, kai blokelių dydis $m > 4$.

Dabar palyginkime bazinės CDF 9/7 transformacijos ir CDF 9/7 transformacijos su daline blokų dekoreliacija ($m = 5$) paklaidas.

3.6 pav. Bazinės CDF 9/7 ir CDF 9/7 su daline blokų dekoreliacija paklaidos ($m=5$, o β – kintantis). Vientisa linija žymi paklaidų kitimą esant transformacijai su daline blokų dekoreliacija, o punktyrinė linija – bazinis transformacijos atvejis hiperbolinio filtro kontekste.

Kadangi paklaidos gautos su daline blokų dekoreliacija didesnės už bazinės CDF 9/7 paklaidas, tai galima daryti išvadą, jog CDF 9/7 su daline blokų dekoreliacija transformacijos energijos „pakavimo“ savybė pablogėja. Tačiau tai ryškėja tik didėjant suspaudimo koeficientui β . Bendru atveju, kai suspaudimo koeficientas β nėra didelis, tai energijos „pakavimo“ savybė yra išsaugoma.

Išvados

1. Diskrečių bangelių transformacija (DBT) praktiniu požiūriu turi labai svarbią savybę – apdorojamame skaitmeniniame vaizde sukaupia informacija išdėstoma ne tik pagal dažnį, bet ir erdvėje. Deja, pilnas lokalizavimas erdvėje būdingas tik paprasčiausiai DBT šeimos narei – Haar'o transformacijai. Aukštesnės eilės DBT pasižymi daliniu lokalizavimu erdvėje.
2. Atlikti tyrimai leidžia teigti, jog modifikuota (su daline blokų dekoreliacija) CDF 9/7 transformacija šiek tiek pablogina energijos „pakavimo“ savybę, lyginant ją su bazine CDF 9/7 transformacija. Tačiau, kai dekoreliuojami blokai yra santykinai dideli ($n \in \{1, n-2, n-3; N = 2^n\}$), minėtoji savybė išlieka praktiškai nepakitusi. Todėl naująją CDF 9/7 transformacijos versiją galima, patogiu ir tikslingu taikyti praktikoje.
3. Tiek bazinės CDF 9/7, tiek modifikuotos CDF 9/7 transformacijos taikymo efektyvumas (tuo pačiu ir jų savybės) priklauso nuo apdorojamo vaizdo glodumo, t. y. nuo to kaip sparčiai „gęsta“ vaizdą sudarančios aukšto dažnio harmonikos. Pastebėta, kad kuo glodesnis vaizdas, tuo efektyviau jos gali būti panaudotos vaizdams glaudinti (hiperbolinių filtrų kontekste).

Literatūra

1. Jonas Valantinas. Diskrečiosios transformacijos (mokomoji knyga), Technologija, 2009.
2. Patrick J. Van Fleet. Wavelets and lossless compression in the jpeg2000 image compression standard (mokomoji priemonė), University of St. Thomas, 2006.
3. Interneto enciklopedija Vikipedija. [žiūrėta 2011-02-15]. Prieiga per internetą: http://en.wikipedia.org/wiki/Cohen-Daubechies-Feauveau_wavelet.
4. Zhong Guangjun, Cheng Lizhi, Chen Chuowang. A simple 9/7 – tap wavelet filter based on lifting scheme (mokomoji priemonė), School of Computer, National University of Defence Technology Changsha, China. [žiūrėta 2011-02-20]. Prieiga per internetą: <http://www.infinitealgorithms.com/Docs/VideoLinks/9-7lowcomplexity.pdf>.
5. Interneto enciklopedija Vikipedija. [žiūrėta 2011-02-20]. Prieiga per internetą: http://computervision.wikia.com/wiki/Daubechies_wavelet.

Priedai

1. Priedas. Programų tekstai. Programa Matlab.

1.1 CDF 9/7 spektro radimas vienmačiu atveju.

```
function spektras = CDF97(duom)

X = duom;
Y = X;
n = log2(length(X));
lp = [.026748757411 -.016864118443 -.078223266529 .266864118443];
a = [lp .602949018236 fliplr(lp)];
hp = [.045635881557 -.028771763114 -.295635881557];
b = 2.*[hp .557543526229 fliplr(hp)];
for i = 1:n
    s = zeros(1,length(X)/2);
    d = s;
    for k = 1:2:2^(n-i+1)
        if length(X) >= 8
            if k-4 == -3
                s(ceil(k/2)) =
a(1)*X(5)+a(2)*X(4)+a(3)*X(3)+a(4)*X(2)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+
3)+a(9)*X(k+4);
                d(ceil(k/2)) =
b(1)*X(3)+b(2)*X(2)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(k+4);
            end

            if k-4 == -1
                s(ceil(k/2)) = a(1)*X(3)+a(2)*X(2)+a(3)*X(k-2)+a(4)*X(k-
1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+3)+a(9)*X(k+4);
                d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(k+4);
            end

            if k+4 == length(X)+1
                s(ceil(k/2)) = a(1)*X(k-4)+a(2)*X(k-3)+a(3)*X(k-2)+a(4)*X(k-
1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+3)+a(9)*X(length(X)-1);
                d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(length(X)-1);
            end

            if k+4 == length(X)+3
                s(ceil(k/2)) = a(1)*X(k-4)+a(2)*X(k-3)+a(3)*X(k-2)+a(4)*X(k-
1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(length(X)-1)+a(8)*X(length(X)-2)+a(9)*X(length(X)-
3);
                d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(length(X)-1)+b(6)*X(length(X)-2)+b(7)*X(length(X)-
3);
            end

            if k-4 > 0 && k+4 <= length(X)
                s(ceil(k/2)) = a(1)*X(k-4)+a(2)*X(k-3)+a(3)*X(k-2)+a(4)*X(k-
1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+3)+a(9)*X(k+4);
                d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(k+4);
            end
        end
        if length(X) == 4

            if k-4 == -3
                s(ceil(k/2)) =
a(1)*X(3)+a(2)*X(4)+a(3)*X(3)+a(4)*X(2)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+
3)+a(9)*X(3);
```

```

        d(ceil(k/2)) =
b(1)*X(3)+b(2)*X(2)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(3);
    end

    if k-4 == -1
        s(ceil(k/2)) = a(1)*X(3)+a(2)*X(2)+a(3)*X(k-2)+a(4)*X(k-
1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(3)+a(8)*X(2)+a(9)*X(1);
        d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(3)+b(6)*X(2)+b(7)*X(1);
    end
end
if length(X) == 2

    if k-4 == -3
        s(ceil(k/2)) =
a(1)*X(1)+a(2)*X(2)+a(3)*X(1)+a(4)*X(2)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(1)+a(8)*X(2)+a
(9)*X(1);
        d(ceil(k/2)) =
b(1)*X(1)+b(2)*X(2)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(1)+b(6)*X(2)+b(7)*X(1);
    end

end
end
X = s;
for j = 1:2^(n-i)
    Y(j+2^(n-i)) = d(j);
end
end
Y(1:length(s)) = s;
spektras = Y;

```

1.2 CDF 9/7 vaizdo radimas vienmačiu atveju.

```

function duom = ICDF97(spektras)

Y = spektras;
n = log2(length(Y));
lp = [.026748757411 -.016864118443 -.078223266529 .266864118443];
lp = [lp .602949018236 fliplr(lp)];
hp = [.045635881557 -.028771763114 -.295635881557];
hp = 2.*[hp .557543526229 fliplr(hp)];
a = hp .* [-1 1 -1 1 -1 1 -1];
b = lp .* [1 -1 1 -1 1 -1 1 -1 1];
for i = n:-1:1
    X = 1:2^(n-i+1);
    X(1:2:length(X)) = Y(1:length(X)/2);
    X(2:2:length(X)) = Y(length(X)/2+1:length(X));
    s = zeros(1,length(X)/2);
    d = s;
    for k = 1:2:2^(n-i+1)
        if length(X) >= 8
            if k-3 == -2
                s(ceil(k/2)) =
b(8)*X(4)+a(6)*X(3)+b(6)*X(2)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                d(ceil(k/2)) =
b(9)*X(4)+a(7)*X(3)+b(7)*X(2)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(
k+4)+b(1)*X(k+5);
            end

            if k-3 == 0
                s(ceil(k/2)) = b(8)*X(2)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                d(ceil(k/2)) = b(9)*X(2)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(k+4)+b(1)*X(k+5);
            end

            if k+5 == length(X)+2

```

```

        s(ceil(k/2)) = b(8)*X(k-3)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
        d(ceil(k/2)) = b(9)*X(k-3)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(length(X)-
1)+b(1)*X(length(X)-2);
        end

        if k+5 == length(X)+4
            s(ceil(k/2)) = b(8)*X(k-3)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(length(X)-1)+b(2)*X(length(X)-2);
            d(ceil(k/2)) = b(9)*X(k-3)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(length(X)-1)+b(3)*X(length(X)-2)+a(1)*X(length(X)-
3)+b(1)*X(length(X)-4);
            end

            if k-3 > 0 && k+5 <= length(X)
                s(ceil(k/2)) = b(8)*X(k-3)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                d(ceil(k/2)) = b(9)*X(k-3)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(k+4)+b(1)*X(k+5);
                end
            end
            if length(X) == 4

                if k-3 == -2
                    s(ceil(k/2)) =
b(8)*X(4)+a(6)*X(3)+b(6)*X(2)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                    d(ceil(k/2)) =
b(9)*X(4)+a(7)*X(3)+b(7)*X(2)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(
3)+b(1)*X(2);
                    end

                    if k-3 == 0
                        s(ceil(k/2)) = b(8)*X(2)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(3)+b(2)*X(2);
                        d(ceil(k/2)) = b(9)*X(2)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(3)+b(3)*X(2)+a(1)*X(1)+b(1)*X(2);
                        end
                    end
                    if length(X) == 2

                        if k-4 == -3
                            s(ceil(k/2)) =
b(8)*X(2)+a(6)*X(1)+b(6)*X(2)+a(4)*X(1)+b(4)*X(2)+a(2)*X(1)+b(2)*X(2);
                            d(ceil(k/2)) =
b(9)*X(2)+a(7)*X(1)+b(7)*X(2)+a(5)*X(1)+b(5)*X(2)+a(3)*X(1)+b(3)*X(2)+a(1)*X(1)+b(1
)*X(2);
                            end
                        end
                    end
                end
            for j = 1:2^(n-i)
                Y(2*j) = d(j);
                Y(2*j-1) = s(j);
            end
            Y() = YY()
        end
        duom = Y;

```

1.3 CDF 9/7 su daline blokų dekoreliacija vienmačiu atveju, spektro radimas.

```

function spektras = CDF97D(duomenys,p)

YY = zeros(1,size(duomenys,1));
nn =log2(size(duomenys,2));

```

```

pp = nn-p+1;
for kk = 1:2^(pp-1)
    duom = duomenys((kk-1)*2^p+1:kk*2^p);
    X = duom;
    n = log2(length(X));
    lp = [.026748757411 -.016864118443 -.078223266529 .266864118443];
    a = [lp .602949018236 fliplr(lp)];
    hp = [.045635881557 -.028771763114 -.295635881557];
    b = 2.*[hp .557543526229 fliplr(hp)];
    for i = 1:n
        s = zeros(1,length(X)/2);
        d = s;
        for k = 1:2:2^(n-i+1)
            if length(X) >= 8
                if k-4 == -3
                    s(ceil(k/2)) =
a(1)*X(5)+a(2)*X(4)+a(3)*X(3)+a(4)*X(2)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+
3)+a(9)*X(k+4);
                    d(ceil(k/2)) =
b(1)*X(3)+b(2)*X(2)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(k+4);
                end

                if k-4 == -1
                    s(ceil(k/2)) = a(1)*X(3)+a(2)*X(2)+a(3)*X(k-2)+a(4)*X(k-
1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+3)+a(9)*X(k+4);
                    d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(k+4);
                end

                if k+4 == length(X)+1
                    s(ceil(k/2)) = a(1)*X(k-4)+a(2)*X(k-3)+a(3)*X(k-2)+a(4)*X(k-
1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+3)+a(9)*X(length(X)-1);
                    d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(length(X)-1);
                end

                if k+4 == length(X)+3
                    s(ceil(k/2)) = a(1)*X(k-4)+a(2)*X(k-3)+a(3)*X(k-2)+a(4)*X(k-
1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(length(X)-1)+a(8)*X(length(X)-2)+a(9)*X(length(X)-
3);
                    d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(length(X)-1)+b(6)*X(length(X)-2)+b(7)*X(length(X)-
3);
                end

                if k-4 > 0 && k+4 <= length(X)
                    s(ceil(k/2)) = a(1)*X(k-4)+a(2)*X(k-3)+a(3)*X(k-2)+a(4)*X(k-
1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+3)+a(9)*X(k+4);
                    d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(k+4);
                end
            end
            if length(X) == 4

                if k-4 == -3
                    s(ceil(k/2)) =
a(1)*X(3)+a(2)*X(4)+a(3)*X(3)+a(4)*X(2)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+
3)+a(9)*X(3);
                    d(ceil(k/2)) =
b(1)*X(3)+b(2)*X(2)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(3);
                end

                if k-4 == -1
                    s(ceil(k/2)) = a(1)*X(3)+a(2)*X(2)+a(3)*X(k-2)+a(4)*X(k-
1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(3)+a(8)*X(2)+a(9)*X(1);
                    d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(3)+b(6)*X(2)+b(7)*X(1);
                end
            end
        end
    end
end

```



```

end
if length(X) == 2
    if k-4 == -3
        s(ceil(k/2)) =
a(1)*X(1)+a(2)*X(2)+a(3)*X(1)+a(4)*X(2)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(1)+a(8)*X(2)+a
(9)*X(1);
        d(ceil(k/2)) =
b(1)*X(1)+b(2)*X(2)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(1)+b(6)*X(2)+b(7)*X(1);
    end

end
end
for j = 1:2^(n-i)
    YY(j+2^(nn-i)+(kk-1)*size(X,2)/2) = d(j);
end
X = s;
end
YY(kk) = s;
end
for k = 1:nn-p
    Y = zeros(1,2^(nn-p-k+1));
    for i = 1:2^(nn-p-k)
        X = YY(1+2*(i-1):2*i);
        Y(i) =
a(1)*X(1)+a(2)*X(2)+a(3)*X(1)+a(4)*X(2)+a(5)*X(1)+a(6)*X(2)+a(7)*X(1)+a(8)*X(2)+a(9
)*X(1);
        Y(2^(nn-p-k)+i) =
b(1)*X(1)+b(2)*X(2)+b(3)*X(1)+b(4)*X(2)+b(5)*X(1)+b(6)*X(2)+b(7)*X(1);
    end
    YY(1:2^(nn-p-k+1)) = Y;
end
spektras = YY;

```

1.4 CDF 9/7 su daline blokų dekoreliacija vienmačiu atveju, vaizdo radimas.

```

function duom = ICDF97D(spektras,p)

ZZ = [];
YY = spektras;
nn = log2(length(YY));
lp = [.026748757411 -.016864118443 -.078223266529 .266864118443];
lp = [lp .602949018236 flipplr(lp)];
hp = [.045635881557 -.028771763114 -.295635881557];
hp = 2.*[hp .557543526229 flipplr(hp)];
a = hp .* [-1 1 -1 1 -1 1 -1];
b = lp .* [1 -1 1 -1 1 -1 1 1];
for k = nn-p:-1:1
    Y = zeros(1,2^(nn-p-k+1));
    for i = 1:2^(nn-p-k)
        X = [YY(i) YY(2^(nn-p-k)+i)];
        Y(1+2*(i-1)) =
b(8)*X(2)+a(6)*X(1)+b(6)*X(2)+a(4)*X(1)+b(4)*X(2)+a(2)*X(1)+b(2)*X(2);
        Y(2*i) =
b(9)*X(2)+a(7)*X(1)+b(7)*X(2)+a(5)*X(1)+b(5)*X(2)+a(3)*X(1)+b(3)*X(2)+a(1)*X(1)+b(1
)*X(2);
    end
    YY(1:2^(nn-p-k+1)) = Y;
end
n = p;
for kk = 1:2^(nn-p)
    Y = YY(kk);
    for iii = 1:p
        Y = [Y YY(2^(nn-p+iii-1)+kk+(kk-1)*(2^(iii-1)-1):2^(nn-p+iii-1)+kk+(kk-
1)*(2^(iii-1)-1)+2^(iii-1)-1)];
    end
end

```

```

end
for i = p:-1:1
    X = 1:2^(n-i+1);
    X(1:2:length(X)) = Y(1:length(X)/2);
    X(2:2:length(X)) = Y(length(X)/2+1:length(X));
    s = zeros(1,length(X)/2);
    d = s;
    for k = 1:2:2^(n-i+1)
        if length(X) >= 8
            if k-3 == -2
                s(ceil(k/2)) =
b(8)*X(4)+a(6)*X(3)+b(6)*X(2)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                d(ceil(k/2)) =
b(9)*X(4)+a(7)*X(3)+b(7)*X(2)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(
k+4)+b(1)*X(k+5);
            end

            if k-3 == 0
                s(ceil(k/2)) = b(8)*X(2)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                d(ceil(k/2)) = b(9)*X(2)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(k+4)+b(1)*X(k+5);
            end

            if k+5 == length(X)+2
                s(ceil(k/2)) = b(8)*X(k-3)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                d(ceil(k/2)) = b(9)*X(k-3)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(length(X)-
1)+b(1)*X(length(X)-2);
            end

            if k+5 == length(X)+4
                s(ceil(k/2)) = b(8)*X(k-3)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(length(X)-1)+b(2)*X(length(X)-2);
                d(ceil(k/2)) = b(9)*X(k-3)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(length(X)-1)+b(3)*X(length(X)-2)+a(1)*X(length(X)-
3)+b(1)*X(length(X)-4);
            end

            if k-3 > 0 && k+5 <= length(X)
                s(ceil(k/2)) = b(8)*X(k-3)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                d(ceil(k/2)) = b(9)*X(k-3)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(k+4)+b(1)*X(k+5);
            end
        end
        if length(X) == 4

            if k-3 == -2
                s(ceil(k/2)) =
b(8)*X(4)+a(6)*X(3)+b(6)*X(2)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                d(ceil(k/2)) =
b(9)*X(4)+a(7)*X(3)+b(7)*X(2)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(
3)+b(1)*X(2);
            end

            if k-3 == 0
                s(ceil(k/2)) = b(8)*X(2)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(3)+b(2)*X(2);
                d(ceil(k/2)) = b(9)*X(2)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(3)+b(3)*X(2)+a(1)*X(1)+b(1)*X(2);
            end
        end
        if length(X) == 2

            if k-4 == -3

```

```

        s(ceil(k/2)) =
b(8)*X(2)+a(6)*X(1)+b(6)*X(2)+a(4)*X(1)+b(4)*X(2)+a(2)*X(1)+b(2)*X(2);
        d(ceil(k/2)) =
b(9)*X(2)+a(7)*X(1)+b(7)*X(2)+a(5)*X(1)+b(5)*X(2)+a(3)*X(1)+b(3)*X(2)+a(1)*X(1)+b(1)
)*X(2);
        end

    end

    end
    for j = 1:2^(n-i)
        Y(2*j) = d(j);
        Y(2*j-1) = s(j);
    end
end
ZZ = [ZZ Y];
end
duom = ZZ;

```

1.5 CDF 9/7 spektro radimas dvimačiu atveju.

```

function spektras = CDF2D(duomenys)

for dd = 1:2
    for m = 1:size(duomenys,1)
        duom = duomenys(:,m);
        X = duom;
        Y = X;
        n = log2(length(X));
        lp = [.026748757411 -.016864118443 -.078223266529 .266864118443];
        a = [lp .602949018236 fliplr(lp)];
        hp = [.045635881557 -.028771763114 -.295635881557];
        b = 2.*[hp .557543526229 fliplr(hp)];
        for i = 1:n
            s = zeros(1,length(X)/2);
            d = s;
            for k = 1:2:2^(n-i+1)
                if length(X) >= 8
                    if k-4 == -3
                        s(ceil(k/2)) =
a(1)*X(5)+a(2)*X(4)+a(3)*X(3)+a(4)*X(2)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+
3)+a(9)*X(k+4);
                        d(ceil(k/2)) =
b(1)*X(3)+b(2)*X(2)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(k+4);
                    end

                    if k-4 == -1
                        s(ceil(k/2)) = a(1)*X(3)+a(2)*X(2)+a(3)*X(k-2)+a(4)*X(k-
1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+3)+a(9)*X(k+4);
                        d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(k+4);
                    end

                    if k+4 == length(X)+1
                        s(ceil(k/2)) = a(1)*X(k-4)+a(2)*X(k-3)+a(3)*X(k-
2)+a(4)*X(k-1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+3)+a(9)*X(length(X)-1);
                        d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(length(X)-1);
                    end

                    if k+4 == length(X)+3
                        s(ceil(k/2)) = a(1)*X(k-4)+a(2)*X(k-3)+a(3)*X(k-
2)+a(4)*X(k-1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(length(X)-1)+a(8)*X(length(X)-
2)+a(9)*X(length(X)-3);
                        d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(length(X)-1)+b(6)*X(length(X)-2)+b(7)*X(length(X)-
3);
                    end
                end
            end
        end
    end
end

```

```

end

if k-4 > 0 && k+4 <= length(X)
    s(ceil(k/2)) = a(1)*X(k-4)+a(2)*X(k-3)+a(3)*X(k-
2)+a(4)*X(k-1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+3)+a(9)*X(k+4);
    d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(k+4);
end
end
if length(X) == 4

if k-4 == -3
    s(ceil(k/2)) =
a(1)*X(3)+a(2)*X(4)+a(3)*X(3)+a(4)*X(2)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+
3)+a(9)*X(3);
    d(ceil(k/2)) =
b(1)*X(3)+b(2)*X(2)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(3);
end

if k-4 == -1
    s(ceil(k/2)) = a(1)*X(3)+a(2)*X(2)+a(3)*X(k-2)+a(4)*X(k-
1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(3)+a(8)*X(2)+a(9)*X(1);
    d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(3)+b(6)*X(2)+b(7)*X(1);
end
end
if length(X) == 2

if k-4 == -3
    s(ceil(k/2)) =
a(1)*X(1)+a(2)*X(2)+a(3)*X(1)+a(4)*X(2)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(1)+a(8)*X(2)+a
(9)*X(1);
    d(ceil(k/2)) =
b(1)*X(1)+b(2)*X(2)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(1)+b(6)*X(2)+b(7)*X(1);
end

end
end
X = s;
for j = 1:2^(n-i)
    Y(j+2^(n-i)) = d(j);
end
end
Y(1:length(s)) = s;
duomenys(:,m) = Y;
end
duomenys = duomenys';
end
spektras = duomenys;

```

1.6 CDF 9/7 vaizdo radimas dvimačiu atveju.

```

function duom = ICDF2D(duomenys)

for dd = 1:2
    for m = 1:size(duomenys,1)
        spektras = duomenys(:,m);
        Y = spektras;
        n = log2(length(Y));
        lp = [.026748757411 -.016864118443 -.078223266529 .266864118443];
        lp = [lp .602949018236 flipplr(lp)];
        hp = [.045635881557 -.028771763114 -.295635881557];
        hp = 2.*[hp .557543526229 flipplr(hp)];
        a = hp .* [-1 1 -1 1 -1 1 -1];
        b = lp .* [1 -1 1 -1 1 -1 1 -1 1];
        for i = n:-1:1

```

```

X = 1:2^(n-i+1);
X(1:2:length(X)) = Y(1:length(X)/2);
X(2:2:length(X)) = Y(length(X)/2+1:length(X));
s = zeros(1,length(X)/2);
d = s;
for k = 1:2:2^(n-i+1)
    if length(X) >= 8
        if k-3 == -2
            s(ceil(k/2)) =
b(8)*X(4)+a(6)*X(3)+b(6)*X(2)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
            d(ceil(k/2)) =
b(9)*X(4)+a(7)*X(3)+b(7)*X(2)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(
k+4)+b(1)*X(k+5);
        end

        if k-3 == 0
            s(ceil(k/2)) = b(8)*X(2)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
            d(ceil(k/2)) = b(9)*X(2)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(k+4)+b(1)*X(k+5);
        end

        if k+5 == length(X)+2
            s(ceil(k/2)) = b(8)*X(k-3)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
            d(ceil(k/2)) = b(9)*X(k-3)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(length(X)-
1)+b(1)*X(length(X)-2);
        end

        if k+5 == length(X)+4
            s(ceil(k/2)) = b(8)*X(k-3)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(length(X)-1)+b(2)*X(length(X)-2);
            d(ceil(k/2)) = b(9)*X(k-3)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(length(X)-1)+b(3)*X(length(X)-2)+a(1)*X(length(X)-
3)+b(1)*X(length(X)-4);
        end

        if k-3 > 0 && k+5 <= length(X)
            s(ceil(k/2)) = b(8)*X(k-3)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
            d(ceil(k/2)) = b(9)*X(k-3)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(k+4)+b(1)*X(k+5);
        end
    end
    if length(X) == 4

        if k-3 == -2
            s(ceil(k/2)) =
b(8)*X(4)+a(6)*X(3)+b(6)*X(2)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
            d(ceil(k/2)) =
b(9)*X(4)+a(7)*X(3)+b(7)*X(2)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(
3)+b(1)*X(2);
        end

        if k-3 == 0
            s(ceil(k/2)) = b(8)*X(2)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(3)+b(2)*X(2);
            d(ceil(k/2)) = b(9)*X(2)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(3)+b(3)*X(2)+a(1)*X(1)+b(1)*X(2);
        end
    end
    if length(X) == 2

        if k-4 == -3
            s(ceil(k/2)) =
b(8)*X(2)+a(6)*X(1)+b(6)*X(2)+a(4)*X(1)+b(4)*X(2)+a(2)*X(1)+b(2)*X(2);

```



```

                s(ceil(k/2)) = a(1)*X(k-4)+a(2)*X(k-3)+a(3)*X(k-
2)+a(4)*X(k-1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(length(X)-1)+a(8)*X(length(X)-
2)+a(9)*X(length(X)-3);
                d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(length(X)-1)+b(6)*X(length(X)-2)+b(7)*X(length(X)-
3);
                end

                if k-4 > 0 && k+4 <= length(X)
                    s(ceil(k/2)) = a(1)*X(k-4)+a(2)*X(k-3)+a(3)*X(k-
2)+a(4)*X(k-1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+3)+a(9)*X(k+4);
                    d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(k+4);
                end
            end
            if length(X) == 4

                if k-4 == -3
                    s(ceil(k/2)) =
a(1)*X(3)+a(2)*X(4)+a(3)*X(3)+a(4)*X(2)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(k+2)+a(8)*X(k+
3)+a(9)*X(3);
                    d(ceil(k/2)) =
b(1)*X(3)+b(2)*X(2)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(k+2)+b(6)*X(k+3)+b(7)*X(3);
                end

                if k-4 == -1
                    s(ceil(k/2)) = a(1)*X(3)+a(2)*X(2)+a(3)*X(k-
2)+a(4)*X(k-1)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(3)+a(8)*X(2)+a(9)*X(1);
                    d(ceil(k/2)) = b(1)*X(k-2)+b(2)*X(k-
1)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(3)+b(6)*X(2)+b(7)*X(1);
                end
            end
            if length(X) == 2

                if k-4 == -3
                    s(ceil(k/2)) =
a(1)*X(1)+a(2)*X(2)+a(3)*X(1)+a(4)*X(2)+a(5)*X(k)+a(6)*X(k+1)+a(7)*X(1)+a(8)*X(2)+a
(9)*X(1);
                    d(ceil(k/2)) =
b(1)*X(1)+b(2)*X(2)+b(3)*X(k)+b(4)*X(k+1)+b(5)*X(1)+b(6)*X(2)+b(7)*X(1);
                end
            end
        end
        end
        for j = 1:2^(n-i)
            YY(j+2^(nn-i)+(kk-1)*length(X)/2) = d(j);
        end
        X = s;
    end
    YY(kk) = s;
end
for k = 1:nn-p
    Y = zeros(1,2^(nn-p-k+1));
    for i = 1:2^(nn-p-k)
        X = YY(1+2*(i-1):2*i);
        Y(i) =
a(1)*X(1)+a(2)*X(2)+a(3)*X(1)+a(4)*X(2)+a(5)*X(1)+a(6)*X(2)+a(7)*X(1)+a(8)*X(2)+a(9
)*X(1);
        Y(2^(nn-p-k)+i) =
b(1)*X(1)+b(2)*X(2)+b(3)*X(1)+b(4)*X(2)+b(5)*X(1)+b(6)*X(2)+b(7)*X(1);
    end
    YY(1:2^(nn-p-k+1)) = Y;
end
duomuo(:,m) = YY;
end
duomuo = duomuo';
end
spektras = duomuo;

```

1.8 CDF 9/7 su daline blokų dekoreliacija dvimačiu atveju, vaizdo radimas.

```

function duom = ICDF2DD(duomenys,p)

for dd = 1:2
    for m = 1:size(duomenys,1)
        spektras = duomenys(:,m);
        ZZ = [];
        YY = spektras';
        nn = log2(length(YY));
        lp = [.026748757411 -.016864118443 -.078223266529 .266864118443];
        lp = [lp .602949018236 fliplr(lp)];
        hp = [.045635881557 -.028771763114 -.295635881557];
        hp = 2.*[hp .557543526229 fliplr(hp)];
        a = hp .* [-1 1 -1 1 -1 1 -1];
        b = lp .* [1 -1 1 -1 1 -1 1 -1 1];
        for k = nn-p:-1:1
            Y = zeros(1,2^(nn-p-k+1));
            for i = 1:2^(nn-p-k)
                X = [YY(i) YY(2^(nn-p-k)+i)];
                Y(1+2*(i-1)) =
b(8)*X(2)+a(6)*X(1)+b(6)*X(2)+a(4)*X(1)+b(4)*X(2)+a(2)*X(1)+b(2)*X(2);
                Y(2*i) =
b(9)*X(2)+a(7)*X(1)+b(7)*X(2)+a(5)*X(1)+b(5)*X(2)+a(3)*X(1)+b(3)*X(2)+a(1)*X(1)+b(1)
)*X(2);
            end
            YY(1:2^(nn-p-k+1)) = Y;
        end
        n = p;
        for kk = 1:2^(nn-p)
            Y = YY(kk);
            for iii = 1:p
                Y = [Y YY(2^(nn-p+iii-1)+kk+(kk-1)*(2^(iii-1)-1):2^(nn-p+iii-
1)+kk+(kk-1)*(2^(iii-1)-1)+2^(iii-1)-1)];
            end
            for i = p:-1:1
                X = 1:2^(n-i+1);
                X(1:2:length(X)) = Y(1:length(X)/2);
                X(2:2:length(X)) = Y(length(X)/2+1:length(X));
                s = zeros(1,length(X)/2);
                d = s;
                for k = 1:2:2^(n-i+1)
                    if length(X) >= 8
                        if k-3 == -2
                            s(ceil(k/2)) =
b(8)*X(4)+a(6)*X(3)+b(6)*X(2)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                            d(ceil(k/2)) =
b(9)*X(4)+a(7)*X(3)+b(7)*X(2)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(
k+4)+b(1)*X(k+5);
                        end

                        if k-3 == 0
                            s(ceil(k/2)) = b(8)*X(2)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                            d(ceil(k/2)) = b(9)*X(2)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(k+4)+b(1)*X(k+5);
                        end

                        if k+5 == length(X)+2
                            s(ceil(k/2)) = b(8)*X(k-3)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                    end
                end
            end
        end
    end
end

```



```

        d(ceil(k/2)) = b(9)*X(k-3)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(length(X)-
1)+b(1)*X(length(X)-2);
        end

        if k+5 == length(X)+4
            s(ceil(k/2)) = b(8)*X(k-3)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(length(X)-1)+b(2)*X(length(X)-2);
            d(ceil(k/2)) = b(9)*X(k-3)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(length(X)-1)+b(3)*X(length(X)-2)+a(1)*X(length(X)-
3)+b(1)*X(length(X)-4);
            end

            if k-3 > 0 && k+5 <= length(X)
                s(ceil(k/2)) = b(8)*X(k-3)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                d(ceil(k/2)) = b(9)*X(k-3)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(k+4)+b(1)*X(k+5);
            end
        end
        if length(X) == 4

            if k-3 == -2
                s(ceil(k/2)) =
b(8)*X(4)+a(6)*X(3)+b(6)*X(2)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(k+2)+b(2)*X(k+3);
                d(ceil(k/2)) =
b(9)*X(4)+a(7)*X(3)+b(7)*X(2)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(k+2)+b(3)*X(k+3)+a(1)*X(
3)+b(1)*X(2);
            end

            if k-3 == 0
                s(ceil(k/2)) = b(8)*X(2)+a(6)*X(k-2)+b(6)*X(k-
1)+a(4)*X(k)+b(4)*X(k+1)+a(2)*X(3)+b(2)*X(2);
                d(ceil(k/2)) = b(9)*X(2)+a(7)*X(k-2)+b(7)*X(k-
1)+a(5)*X(k)+b(5)*X(k+1)+a(3)*X(3)+b(3)*X(2)+a(1)*X(1)+b(1)*X(2);
            end
        end
        if length(X) == 2

            if k-4 == -3
                s(ceil(k/2)) =
b(8)*X(2)+a(6)*X(1)+b(6)*X(2)+a(4)*X(1)+b(4)*X(2)+a(2)*X(1)+b(2)*X(2);
                d(ceil(k/2)) =
b(9)*X(2)+a(7)*X(1)+b(7)*X(2)+a(5)*X(1)+b(5)*X(2)+a(3)*X(1)+b(3)*X(2)+a(1)*X(1)+b(1
)*X(2);
            end

        end
        end
        for j = 1:2^(n-i)
            Y(2*j) = d(j);
            Y(2*j-1) = s(j);
        end
        end
        ZZ = [ZZ Y];
    end
    duomenys(:,m) = ZZ;
end
duomenys = duomenys';
end
duom = duomenys;

```

1.9 Pagrindinė programa naudojama paklaidų radimui hiperbolinio filtro kontekste.

```
function Pagrindine()
```

```
duom = double(imread('Barbara512.png'));
spektras = CDF2DD(duom,4);
M = 8298;
for i = 0:size(duom,1)-1
    for j = 0:size(duom,2)-1
        k = i;
        l = j;
        if i == 0
            k = 1;
        end
        if j == 0
            l = 1;
        end
        if k*l >= M
            spektras(i+1,j+1) = 0;
        end
    end
end
rez = ICDF2DD(spektras,4);
for i = 1:size(rez,1)
    for j = 1:size(rez,2)
        if rez(i,j) > 256
            rez(i,j) = 256;
        end
        if rez(i,j) < 1
            rez(i,j) = 1;
        end
        rez(i,j) = round(rez(i,j));
    end
end
paklaida = sum(sum((duom - rez).^2))/size(duom,1)^2
imwrite(rez,gray(256),'rez.png','png');
```