Student: Rudeanu Paul Emil

Calculatoare an III, Grupa 3132A

prelucrarea numerică a imaginilor

Proiect Transformata Cosinus Discretă

Profesor îndrumător: S.I.Dr.Ing Prodan Remus

Conținut

[Tema și motivația alegerii 2](#_Toc453177013)

[1.Aspecte teoretice 2](#_Toc453177014)

[1.1 Descrierea problemei 2](#_Toc453177015)

[1.2 Abordarea teoretică a problemei 3](#_Toc453177016)

[2.Implementare 4](#_Toc453177017)

[2.1 Tehnologii folosite 4](#_Toc453177018)

[2.2 Algoritmul de calcul DCT 4](#_Toc453177019)

[2.3. Interfața cu utilizatorul 5](#_Toc453177020)

[3.Concluzie 8](#_Toc453177021)

[4. Bibliografie 9](#_Toc453177022)

# 

# Tema și motivația alegerii

Tema proiectului – realizarea unei aplicații ce ilustrează modul de calcul a transformatei cosinus discrete, utilizând mediul de dezvoltare C++ Builder.

Mi-am propus ca lucrând la acest proiect să aprofundez cunoștințele matematice care au aplicabilitate în prelucrarea imaginilor deoarece tema este importantă în diverse domenii ale ingineriei, dar și pentru a-mi îmbunătăți modul de comunicare a unor concepte folosind diverse controale grafice.

# 1.Aspecte teoretice

## 1.1 Descrierea problemei

Transformata DCT este foarte utilizată în tratamentul sunetului și al imaginii, în special pentru algoritmele de compresie. Ea exprimă o secvență finită de puncte de date ca termeni ai unei sume de funcții cosinus ce oscilează la frecvențe diferite.

Această transformată este importantă pentru numeroase aplicații în știință și inginerie, de la compresie lossy (cu pierderi de informații) a semnalului audio (în cadrul formatului MP3) și a imaginilor (în cadrul formatului JPEG) la metode spectrale pentru obținerea soluției numerice a ecuațiilor cu derivate parțiale.

DCT este o funcție [liniară](https://ro.wikipedia.org/wiki/Aplica%C8%9Bie_liniar%C4%83) inversibilă RN → RN sau altfel spus o matrice pătrată N × N inversibilă.

C:\Users\mihoc\Desktop\dct1.PNG

Figura 1. DCT I

Cea mai utilizată variantă a transformatei cosinus discrete este DCT-II, denumită deseori “transformata cosinus discretă”, iar funcția inversă DCT-III, deseori nominalizată ca fiind “transformata cosinus discretă inversă (IDCT)”.

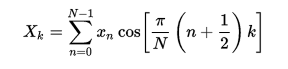


Figura 2. DCT II

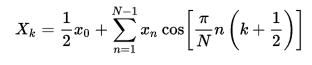


Figura 3. DCT III

## 1.2 Abordarea teoretică a problemei

Transformata cosinus discretă este o transformare ortogonală – dacă transformata directă este de forma *Y = c \* X*, atunci transformata inversă este *X = c -1 \* Y*.

Din punct de vedere matematic, DCT directă se exprimă utilizând formula:

(1)

Inversa transformatei cosinus discrete va fi:

(2)

unde C(u) =

Pentru analiza semnalului bidimensional (2D) cum sunt imaginile, este necesară o versiune 2D a transformatei cosinus discrete. Punctele imaginii în fiecare bloc sunt numerotate de la (0, 0) la (M-1, N-1), iar f(x, y) reprezintă valoarea pixelului aflat în punctul de coordonate (x, y).

Astfel, formulele de calcul pentru DCT și IDCT pentru semnale 2D vor fi:

(3)

respectiv

(4)

unde C(u), C(v) =

Rezultatul aplicării transformatei cosinus discrete îl constituie un set de frecvențe F(u, v) care indică gradul în care se modifică valorile de intrare la o anumită rată de eșantionare.

Valorile semnificative ale acestei matrice se află în colțul stânga sus. Pentru un bloc de 8x8 valori, prin F(0, 0) se notează valoarea ce indică la ce grad valorile nu se modifică – media celor 64 de valori de intrare, denumită și coeficientul DC (“curent continuu”).

Toate celelalte valori sunt denumite coeficienți AC (“curent alternativ”).

Secvența rezultată va fi parcursă in zig-zag, întrucât valorile coeficienților AC sunt reduse și continuă să descrească, iar de la un anumit coeficient toate valorile devin 0.

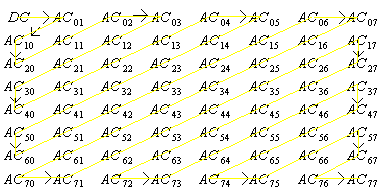


Figura 4. Parcurgerea în zig zag a matricei rezultate

# 2.Implementare

## 2.1 Tehnologii folosite

Această aplicaţie este realizată integral în C++, în mediul de lucru integrat (IDE) Borland C++ Builder 6.0, fiind concepută ca aplicație desktop, care rulează pe sisteme Windows și nu necesită conexiune la Internet.

## 2.2 Algoritmul de calcul DCT

Calculul DCT se realizează într-o serie de etape evidențiate pe fiecare tab din cadrul interfeței utilizator.

Mai întâi, se va alege o zonă de 8x8 pixeli din cadrul unei imagini, valorile nuanțelor de gri fiind automat preluate într-o matrice. Trebuie reținut că utilizarea unei imagini color nu influențează calculul transformatei cosinus deoarece aplicația determină automat nivelul de gri ca medie a valorilor celor trei componentele RGB și lucrează în continuare doar cu acestea.

În cazul codării unei imagini b = 8, astfel încât valorile originale cuprinse în intervalul [0, 255] sunt deplasate în intervalul [-128, 127].

Calculul propriu-zis al DCT se va realiza pe baza valorilor ce rezultă în urma translării, impunând rotunjirea la întreg a acestora și astfel o anumită pierdere de informații.

|  |
| --- |
| int S[8][8]; //matricea nivelelor de gri aferente unei portiuni de 8x8 pixeli din cadrul unei imagini  int T[8][8]; //matricea ce va contine rezultatul transformarii  double rez, INVERS\_RAD\_2 = 1/sqrt(2);  //transformarea cosinus discreta  for (int y = 0; y < 8; y++)  {  for (int x = 0; x < 8; x++)  {  rez = 0;  // rezultatul este resetat inainte de a calcula valorile din matricea DCT  double Cx = 1;  double Cy = 1;  if (x == 0) Cx = INVERS\_RAD\_2;  if (y == 0) Cy = INVERS\_RAD\_2;  for (int i = 0; i < 8; i++)  {  for (int j = 0; j < 8; j++)  {  rez += S[i][j] \* cos((2\*j+1)\*x\*M\_PI/16) \* cos((2\*i+1)\*y\*  M\_PI/16);  }  }  //salvarea valorilor rezultate in matricea DCT  T[y][x] = Cx \* Cy \* rez \* 0.25;  }  } |

Acest ultim pas poate fi ilustrat prin cod C++ astfel:

## 2.3. Interfața cu utilizatorul

Mai întâi, utilizatorul va încărca o imagine în format permis ( BMP sau JPEG).

Deplasarea cu mouse-ul pe deasupra unui “pixel” mărit va determina selectarea valorii numerice care corespunde nuanței de gri respective

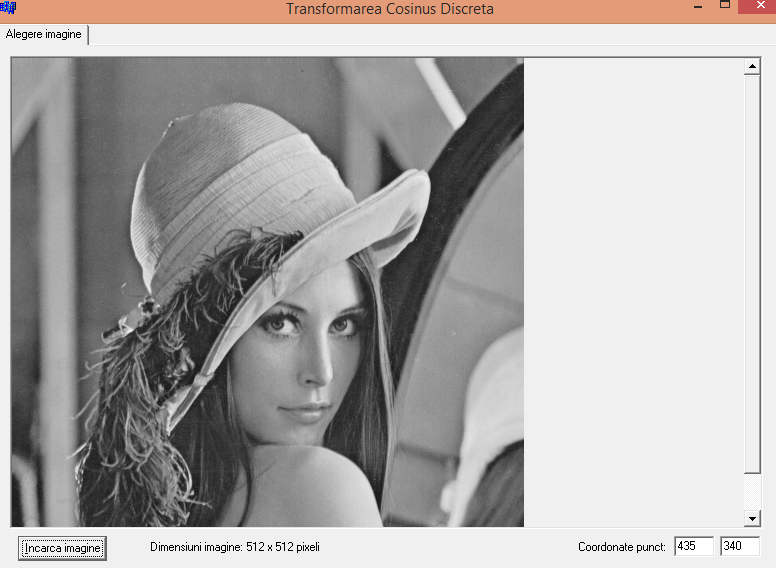


Figura 5. Interfața cu utilizatorul

După aceasta, Tabul următor va fi vizibil, în el utilizatorul poate vedea detaliul selectat din imagine și nivelele de gri aferente acestuia.



Figura 6. Tabul Detaliu imagine și valori de gri corespunzătoare

Următoarea etapă este cea de translare a valorilor din intervalul [0, 255] în intervalul [-128, 127]; această centrare permite obținerea unei precizii mai bune în calculul transformatei cosinus.

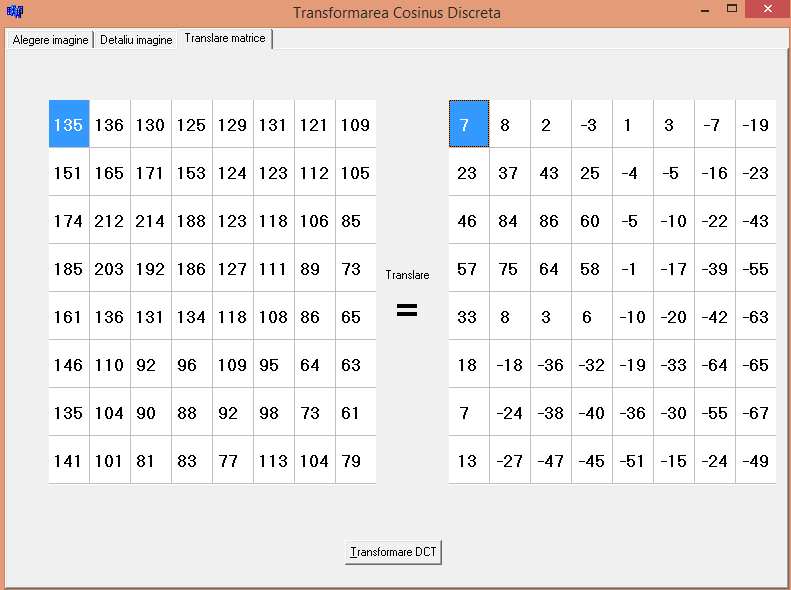


Figura 7. Translarea valorilor

Ultimul Tab ilustrează Trasformarea Cosinus Discretă, în stânga avem rezultatul obținut în urma translării, iar în dreapta avem matricea obținută după aplicarea DCT.

Formula de calcul este aplicată fiecărui coeficient al matricei rezultat în urma translării:

unde C(u), C(v) =

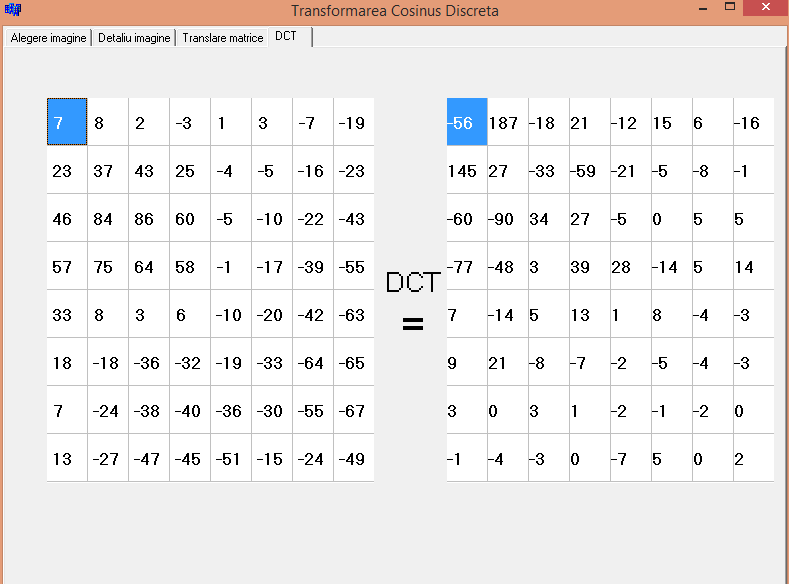


Figura 8. Rezultatul aplicării DCT

Coeficienţii din colţul din stânga-sus corespund frecvenţelor joase, variaţii lente de intensitate între pixeli, iar pe masură ce se avansează către colţul din dreapta-jos coeficienţii corespund frecvenţelor înalte, variaţii rapide de intensitate date de detaliile fine din imagine. În general, într-o imagine reală frecvenţele înalte au o pondere mai redusă decât cele joase ceea ce explică valorile obţinute în urma transformării.

Se poate reveni oricând la pasul anterior. Fiecare Tab afișează o etapă a calculului, facilitând înțelegerea DCT.

Datele de ieşire ce rezultă în urma procesării informaţiilor în cadrul acestei aplicații tutorial sunt cele vizibile în fiecare etapă a procesului de calcul:

* valorile numerice corespunzătoare zonei selectate de utilizator
* valorile numerice translatate
* valorile ce rezultă în urma calculului propriu-zis al transformatei cosinus

# 3.Concluzie

La finalizarea acestui proiect eu consider că am înmagazinat informații nu numai despre C++ Buider, ci și despre prelucrarea imaginilor. Calculul pixelilor și aplicarea transformărilor duce la punerea în practică a calculelor matematice învățate până acum.

# 4. Bibliografie

1. <http://apollo.eed.usv.ro/>~remus
2. <https://ro.wikipedia.org/wiki/Transformata_cosinus_discret%C4%83>
3. <https://www.scribd.com/doc/218989084/Transformata-DCT>
4. <http://www.academia.edu/13941552/DPCM_DCT_FOR_SATELLITE_IMAGE_COMPRESSION>
5. <http://www-math.mit.edu/~gs/papers/dct.pdf>
6. <http://waset.org/publications/3030/parallel-image-compression-and-analysis-with-wavelets>