

UNIVERSITATEA TEHNICĂ A MOLDOVEI
Facultatea Calculatoare, Informatică și Microelectronică
Departamentul Informatică și Ingineria Sistemelor

GRAFICA PE CALCULATOR

TEMA 5. COMPRESIA DE IMAGINI

l. u., dr. NASTAS Andrei

- 5.1. Reprezentarea numerică a imaginilor
- 5.2. Reprezentarea imaginii în format necompresat
- 5.3. Metode și abordări ale compresiei imaginii
- 5.4. Transformări folosite în compresia imaginilor
 - 5.4.1. Transformări ortogonale
 - 5.4.2. Transformări bi-dimensionale
 - 5.4.3. Transformarea Karhunen-Loève (KLT)
 - 5.4.4. Transformarea Walsh-Hadamard (WHT)
 - 5.4.5. Transformarea Haar
 - 5.4.6. Transformata Discretă Cosinus (DCT)
 - 5.4.7. Transformarea Discretă Sinus
- 5.5. Compresia JPEG (Joint Photographers ExpertsGroup)
 - 5.5.1. Modul de bază (baseline)
 - 5.5.2. Modul de codare cu pierderi expandat
 - 5.5.3. Compresia secvențială JPEG fără pierderi
 - 5.5.4. Compresia JPEG ierarhică

5.5. Compresia JPEG (Joint Photographers Experts Group)

Domeniul compresiei (codării) de imagini este legat de minimizarea numărului de biți necesari pentru a reface o imagine, cu aplicații în special în transmisia și stocarea imaginilor. Aplicațiile din domeniul transmisiilor de imagini se întâlnesc în televiziunea radiodifuzată, comunicațiile spațiale, radar și sonar, rețele de telecomunicații, transmisii fax, teleconferințe etc. Compresia imaginilor este esențială din punct de vedere al memorării (stocării) imaginilor în aplicații de imagistică medicală, în tehnica video digitală, pentru realizarea documentelor multimedia etc. Noile tehnologii de compresie a imaginilor oferă o soluție posibilă pentru integrarea aplicațiilor de imagini și video digitale. Ratele de compresie au ajuns în prezent până la 1:100, depinzând de calitatea imaginii refăcute. Tehnica de compresie nu este suficientă pentru a putea rezolva problemele care apar în aplicațiile multimedia. Pentru a putea realiza portabilitatea aplicațiilor de imagini și secvențe video digitale pe mai multe sisteme, este necesară implementarea unor standarde pentru compresia datelor multimedia. Aceste standarde stabilesc modalitățile de stocare și transmisie a datelor compresate în vederea posibilității utilizării lor. Cel mai utilizat standard de compresie a imaginilor statice este standardul JPEG, creat de Joint Photographics Experts Group. Metoda de compresie este de tip "cu pierdere", fiind concepută astfel încât să se profite de limitările în percepția video a ochiului uman. Acest standard permite setarea raportului calitate/compresie și lucrează cu aceleași nivele de culoare, în număr de 24 (16,7 milioane de culori), indiferent de numărul total de culori din imagine. În momentul de față este unul dintre cele mai frecvent întâlnite formate de fișiere grafice.

5.5. Compresia JPEG (Joint Photographers Experts Group)

Formatul JPEG este recomandat pentru afișarea de imagini redade cu o foarte mare varietate de culori sau pentru imagini de precizie fotografică. JPEG folosește o tehnică de compresie variabilă, care are drept rezultat obținerea de fișiere foarte mici în comparație cu alte formate.

Standardul JPEG se bazează pe transformarea informației primare din domeniul timp în domeniul frecvență. Este cunoscut faptul că imaginile sunt puternic corelate spațial, adică un pixel de imagine conține informații și despre pixelii vecini. Corelația spațială ce caracterizează imaginile reprezintă redundanță din punct de vedere informațional și se diminuează prin transformări matematice care au rolul de a concentra energia imaginii în cât mai puține elemente. Transformările matematice din domeniul timp în domeniul frecvență nu reprezintă în sine compresie de date. Abia operațiunile ce urmează, și anume, cuantizarea și codarea entropică reprezintă compresie de date. Reducerea redundanței spațiale se face atât pentru imaginea sursă originală, cât și pentru eroarea reziduală, așa cum se va vedea în cele ce urmează. La refacerea imaginilor, după ce acestea au fost compresate JPEG, cantitatea de informație este mai mică decât cea inițială, fără o afectare vizibilă a calității. Prin transformarea imaginii din domeniul timp, (pixeli), în domeniul frecvență se rețin doar componentele de joasă frecvență ale imaginii. Componentele de frecvență înaltă pot fi reduse, fără o afectare deranjantă a percepției vizuale a imaginii. Evident că acest lucru este determinat de gradul de compresie acceptat.

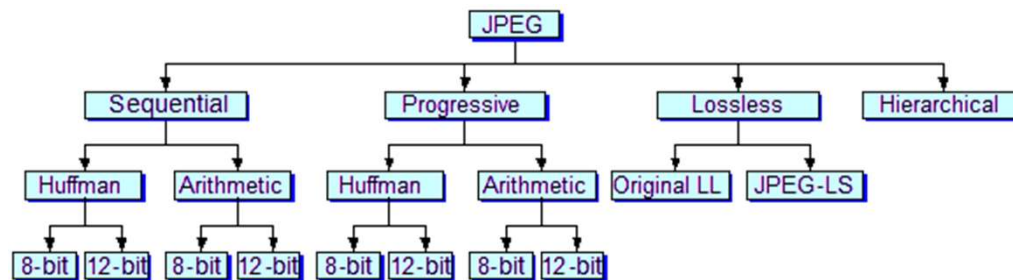
5.5. Compresia JPEG (Joint Photographers Experts Group)

JPEG este o metodă sofisticată de compresie cu pierdere pentru imagini color sau alb-negru (cu scară de gri). Un avantaj al JPEG este faptul că folosește mulți parametri, permițând astfel utilizatorului să regleze cantitatea de date pierdute și, de asemenea, rata de compresie. Momentan reprezintă cel mai bun standard existent în materie de compresie a imaginilor statice. Standardul este implementat atât în format software cât și hardware pentru a satisface necesitățile de prelucrare în timp real a aplicațiilor multimedia. Creat inițial pentru compresia imaginilor statice, standardul a fost extins și pentru secvențele video. Standardul realizat pentru secvențe video se numește M-JPEG (Motion JPEG). Practic în cazul secvențelor video digitale fiecare cadru este considerat ca o imagine fixă și compresat cu standardul JPEG. Metoda nu este cea mai eficientă din punctul de vedere al mărimii ratei de compresie, dar oferă o alternativă pentru compresia video digitală. Adesea, ochiul uman nu distinge nici o degradare a imaginii chiar la o rată de compresie de 10:1 sau 20:1.

5.5. Compresia JPEG (Joint Photographers Experts Group)

Există patru moduri principale de operare specificate de standardul JPEG:

- modul de bază, în care fiecare componentă a imaginii este codată printr-o singură scanare stânga-dreapta, respectiv sus-jos;
 - codarea expandată DCT cu pierderi, în care se realizează o codare progresivă a spectrelor imaginii de intrare;
 - codarea fără pierderi, în care imaginea este codată astfel încât se garantează reproducerea exactă la decodare;
 - codarea ierarhică, în care imaginea este codată la rezoluții multiple.
- În continuare, se va prezenta detaliat numai primul dintre acestea, celelalte numai esența.



5.5.1. Modul de bază (baseline)

Schema bloc a algoritmului de codare JPEG este arătată în figura 5.8.

Principiul compresiei prin metode de tip DCT.

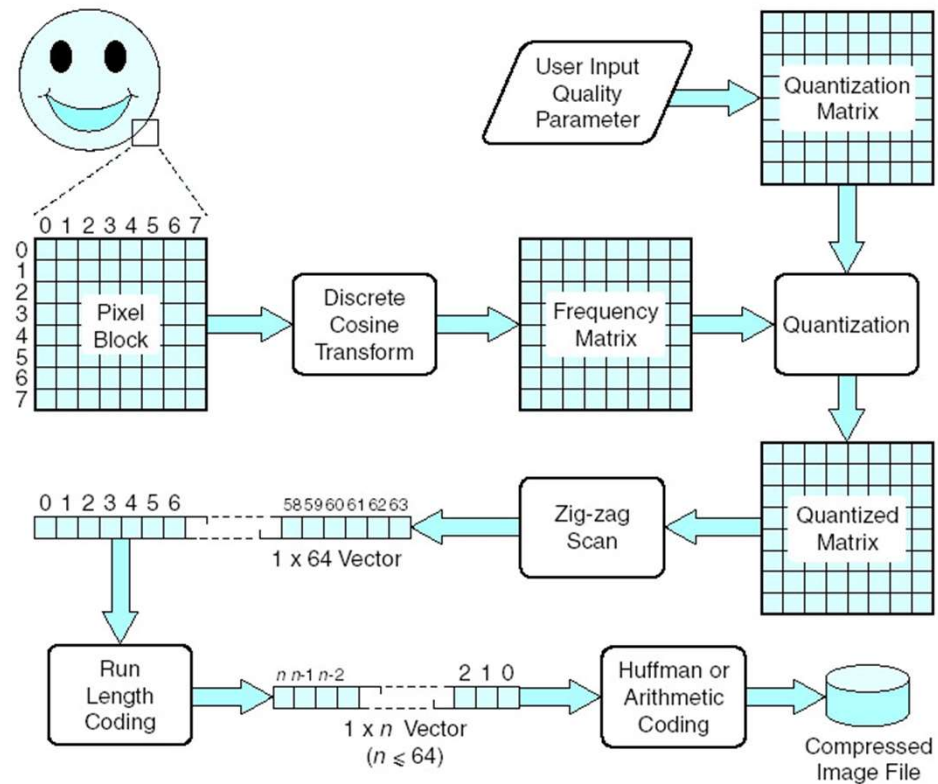


Fig. 5.8. Algoritmul de codare JPEG

5.5.1. Modul de bază (baseline)

Compresia cu pierderi presupune câteva etape de prelucrare, și anume:

- **Transformarea din reprezentarea (R,G,B) în reprezentarea (Y,U,V)**

În procedurile de compresie a imaginii se preferă o reprezentare a culorii diferită de cea normală (R,G,B), și anume, reprezentarea (Y,U,V), obținută cu relațiile 5.1. Valorile componentelor Y, U și V sunt cuprinse între -128 și 127. Utilitatea acestei reprezentări echivalente se poate evidenția în figura 5.9, în care sunt prezentate descompunerile în forma (R,G,B) respectiv (Y,U,V) ale unei imaginii.

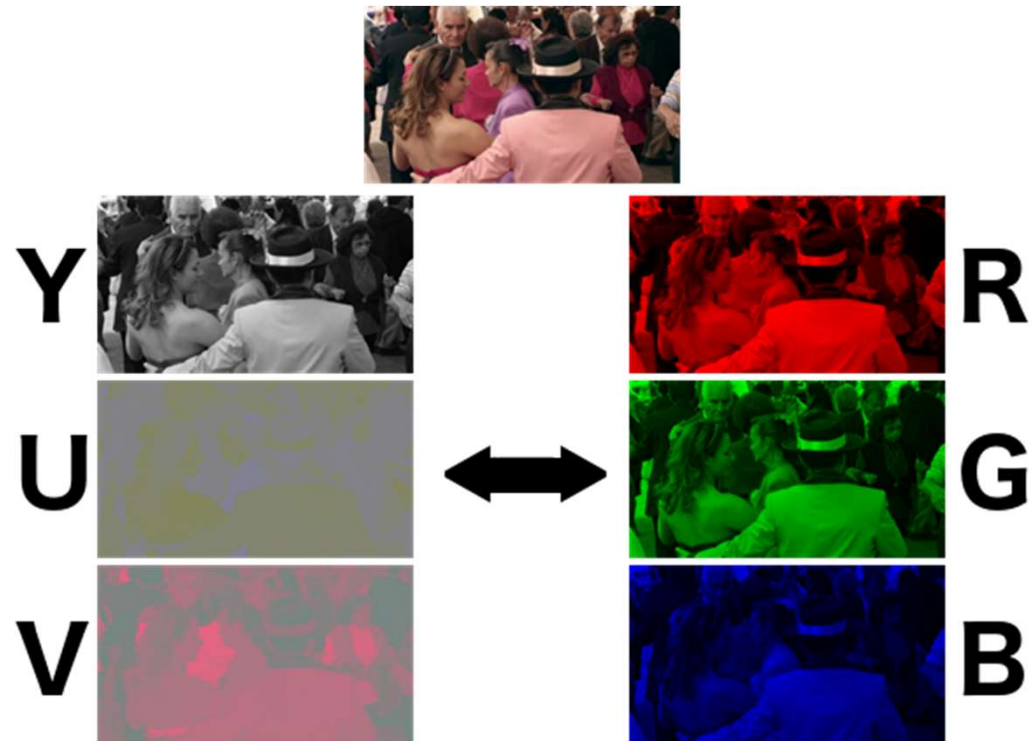


Fig. 5.9. Componentele R, G, B, Y, U, V ale unei imaginii

5.5.1. Modul de bază (baseline)

Analizând acest exemplu, se pot face câteva observații importante și anume:

- componenta Y corespunde unei imagini alb-negru;
- componentele U și V conțin mult mai puține detalii și prezintă un contrast mult mai redus.

Datorită absenței detaliilor și contrastului scăzut al componentelor U și V, acestora li se aplică o subeșantionare cu factorul 2 pe ambele direcții, verticală și orizontală, ținându-se cont de faptul că percepția ochiului este mai mică la semnalele de crominanță, față de cele de luminanță. Modul de realizare a subeșantionării constă în înlocuirea blocurilor de 2x2 puncte cu un singur punct care are intensitatea egală cu media celor 4. În aceste condiții imaginea va fi descrisă de componentele U' și V', din figura 5.10.

Prin aceste operații se realizează o primă compresie, cu factorul 2:1. Astfel, reprezentarea (R, G, B) pentru imaginea din exemplu cu rezoluția de 320 x 240 puncte necesită 3 componente a câte 320 x 240 = 76800 elemente, adică un total de 230400 elemente (octeți). Reprezentarea (Y, U', V') necesită 320 x 240 = 76800 elemente pentru Y și 160 x 120 = 19200 elemente pentru U' și V' adică un total de 115200 elemente (octeți).



Componenta U' (subeșantionată)



Componenta V' (subeșantionată)

Fig. 5.10. Componentele subeșantionate U' și V'

5.5.1. Modul de bază (baseline)

- **Descompunerea în blocuri**

Procedura de compresie se aplică unor blocuri de imagine de 8x8 puncte. Dacă nici una din dimensiunile imaginii nu este multiplu de 8, codorul copie ultima coloană sau linie până când lungimea finală este multiplu de 8. Aceste linii sau coloane suplimentare sunt îndepărtate în timpul procesului de decodare.

Cele trei componente Y, U' și V' sunt descompuse în blocuri de dimensiune 8x8. Datorită rezoluției reduse, în urma subeșantionării, rezultă că la 4 (2x2) blocuri ale componentei Y corespunde câte un singur bloc al componentelor U', respectiv V'.

În cazul formatului JPEG cele trei componente ale blocurilor de imagine sunt prelucrate întrețesut. Pentru o numerotare a blocurilor, conform figurii 5.11, ordinea prelucrării acestora va fi Y1, Y2, Y3, Y4, U1, V1, Y5, Y6, Y7, Y8, U2, V2,.....:

Fiecare bloc este prelucrat utilizând aceeași procedură.

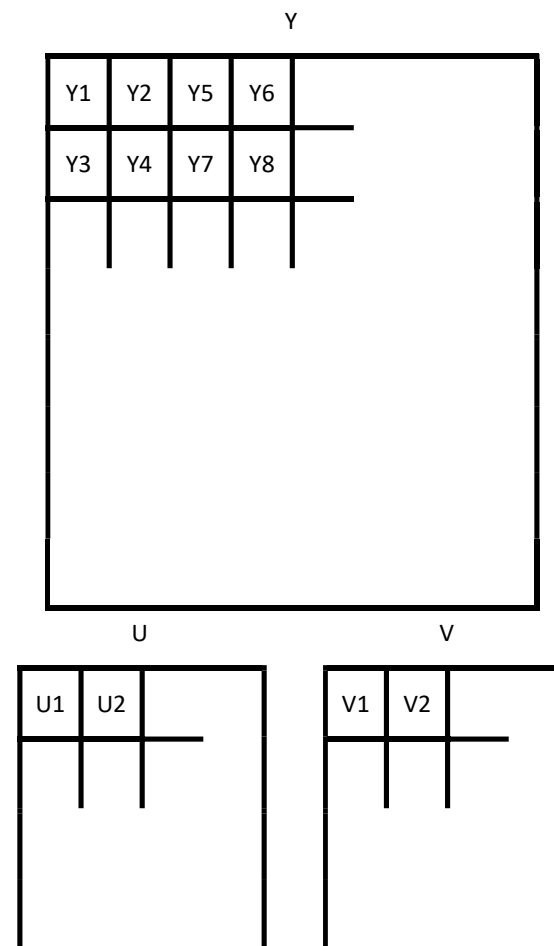


Fig. 5.11. Ordinea prelucrării blocurilor

5.5.1. Modul de bază (baseline)

- **Aplicarea transformării cosinus discrete**

Valorile originale ale componentelor Y , U' , V' sunt cuprinse în domeniul $[0, 2b-1]$, unde b reprezintă numărul de biți/eșantion. Aceste valori sunt deplasate în domeniul $[-2b-1, 2b-1 - 1]$, centrate față de zero, pentru a putea realiza o precizie de calcul mai mare la aplicarea DCT (Discrete Cosine Transform – Transformarea Cosinus Discretă). Pentru primul nivel de codare JPEG, $b = 8$, astfel încât valorile originale cuprinse în intervalul $[0, 255]$ sunt deplasate în intervalul $[-128, 127]$. Fiecare componentă este apoi divizată în blocuri de 8×8 pixeli, așa cum se poate observa și în figura 5.8. Fiecărui bloc astfel obținut i se aplică transformata cosinus discretă bi-dimensională, folosind ecuațiile (5.36), (5.37), (5.38):

Pentru a ilustra cum lucrează algoritmul, se va folosi un bloc de dimensiune 8×8 din componenta de luminanță a unei imagini, așa cum este prezentat în tabelul 5.4.

$$\text{DCT: } G_{ij} = \frac{1}{4} C_i C_j \sum_{x=0}^7 \sum_{y=0}^7 p_{xy} \cos\left(\frac{(2y+1)j\pi}{16}\right) \cos\left(\frac{(2x+1)i\pi}{16}\right) \quad (5.36)$$

$$\text{IDCT: } p_{xy} = \frac{1}{4} \sum_{i=0}^7 \sum_{j=0}^7 C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{16}\right) \cos\left(\frac{(2x+1)i\pi}{16}\right) \quad (5.37)$$

unde

$$C_i = C_j = \frac{1}{\sqrt{2}}, \quad \text{dacă } i = j = 0 \quad (5.38)$$

$$C_i = C_j = 1, \quad \text{în rest}$$

Tabelul 5.4. Bloc de dimensiune 8×8 al unei imagini

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 124 | 125 | 122 | 120 | 122 | 119 | 117 | 118 |
| 121 | 121 | 120 | 119 | 119 | 120 | 120 | 118 |
| 126 | 124 | 123 | 122 | 121 | 121 | 120 | 120 |
| 124 | 124 | 125 | 125 | 126 | 125 | 124 | 124 |
| 127 | 127 | 128 | 129 | 130 | 128 | 127 | 125 |
| 143 | 142 | 143 | 142 | 140 | 139 | 139 | 139 |
| 150 | 148 | 152 | 152 | 152 | 152 | 150 | 151 |
| 156 | 159 | 158 | 155 | 158 | 158 | 157 | 156 |

5.5.1. Modul de bază (baseline)

Considerând blocul de 8×8 pixeli din tabelul 5.4, se scade 128 din fiecare element și aplicând DCT, se obțin coeficienții DCT prezentați în tabelul 5.5.

Se poate observa că elementele acestei matrice au valorile mari concentrate în colțul din stânga sus, restul fiind valori mici, aproape nule. Explicația acestui fenomen este dată de faptul că transformata cosinus discretă realizează o descompunere "în frecvență". Astfel, coeficienții din colțul din stânga sus corespund frecvențelor joase - variații lente de intensitate între pixeli, iar pe măsură ce se avansează către colțul din dreapta-jos coeficienții corespund frecvențelor înalte - variații rapide de intensitate, date de detaliile fine din imagine. În general, într-o imagine reală frecvențele înalte au o pondere mai redusă decât cele joase, ceea ce explică valorile obținute în urma transformării. Această situație este ilustrată în figura 5.12

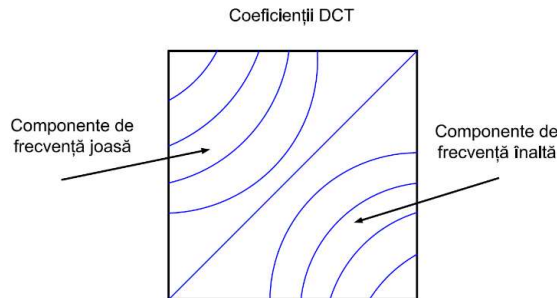


Fig. 5.12. Distribuția componentelor de frecvență într-o matrice de coeficienți DCT cuantizarea matricei transformate

Tabelul 5.5. Matricea G a coeficienții DCT corespunzător blocului de date al imaginii Lena după deplasare

| | | | | | | | |
|---------|-------|-------|-------|--------|-------|-------|-------|
| 39,88 | 6,56 | -2,24 | 1,22 | -0,37 | -1,08 | 0,79 | 1,13 |
| -102,43 | 4,56 | 2,26 | 1,12 | 0,35 | -0,63 | -1,05 | -0,48 |
| 37,77 | 1,31 | 1,77 | 0,25 | -1,50 | -2,21 | -0,10 | 0,23 |
| -5,67 | 2,24 | -1,32 | -0,81 | 1,41 | 0,22 | -0,13 | 0,17 |
| -3,37 | -0,74 | -1,75 | 0,77 | -0,62 | -2,65 | -1,30 | 0,76 |
| 5,98 | -0,13 | -0,45 | -0,77 | 1,99 | -0,26 | 1,46 | 0,00 |
| 3,97 | 5,52 | 2,39 | -0,55 | -0,051 | -0,84 | -0,52 | -0,13 |
| -3,43 | 0,51 | -1,07 | 0,87 | 0,96 | 0,09 | 0,33 | 0,01 |

5.5.1. Modul de bază (baseline)

•Cuantizarea matricii transformate

Operația de cuantizare este singura în care au loc pierderi de informație. Algoritmul JPEG utilizează coeficienți de cuantizare pentru a cuantiza diferenții coeficienți de intrare. Mărimea pasului de cuantizare este organizată într-un tabel, numit tabel de cuantizare. Un exemplu de tabel de cuantizare din recomandările JPEG este prezentat în tabelul 5.6. Fiecare valoare cuantizată este reprezentată de o etichetă. Prin cuantizare se înțelege împărțirea element cu element a matricii G cu o matrice de cuantizare Q , cu reținerea doar a părții întregi, rezultând o matrice I .

Eticheta corespunzătoare valorii cuantizate a coeficienților G_{ij} ai transformatei este

$$I_{i,j} = \left\lfloor \frac{G_{ij}}{Q_{ij}^t} + 0,5 \right\rfloor \quad (5.37)$$

unde Q_{ij}^t este elementul (i, j) din tabelul de cuantizare și $[x]$ este cel mai mare întreg mai mic decât x . Se consideră coeficientul G_{00} din tabelul 5.5, care este 39, 88. Din tabelul 5.6, Q_{00}^t este 16, deci,

$$I_{00} = \left\lfloor \frac{39,88}{16} + 0,5 \right\rfloor = \lfloor 2,9925 \rfloor = 2 \quad (5.38)$$

Tabelul 5.6. Coeficienții de cuantizare pentru luminanta (a) și pentru crominanta (b)

a)

| | | | | | | | |
|----|----|----|----|-----|-----|-----|-----|
| 16 | 11 | 10 | 16 | 24 | 40 | 51 | 61 |
| 12 | 12 | 14 | 19 | 26 | 58 | 60 | 55 |
| 14 | 13 | 16 | 24 | 40 | 57 | 69 | 56 |
| 14 | 17 | 22 | 29 | 51 | 87 | 80 | 62 |
| 18 | 22 | 37 | 56 | 68 | 109 | 103 | 77 |
| 24 | 35 | 55 | 64 | 81 | 104 | 113 | 92 |
| 49 | 64 | 78 | 87 | 103 | 121 | 120 | 101 |
| 72 | 92 | 95 | 98 | 112 | 100 | 103 | 99 |

b)

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 17 | 18 | 24 | 47 | 99 | 99 | 99 | 99 |
| 18 | 21 | 26 | 66 | 99 | 99 | 99 | 99 |
| 24 | 26 | 56 | 99 | 99 | 99 | 99 | 99 |
| 47 | 66 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |

5.5.1. Modul de bază (baseline)

Valoarea reconstruită este obținută din etichetă, prin multiplicarea acesteia cu intrarea corespunzătoare în tabelul de cuantizare. Deci, valoarea reconstruită a ϑ_{00} va fi $I_{00} \times Q_{00}^t$ adică $2 \times 16 = 32$. Eroarea de cuantizare în acest caz este $39,88 - 32 = 7,88$. Similar, din tabelele 5.5 și 5.6, G_{01} este 6,56 și Q_{01}^t este 11, deci

$$I_{01} = \left\lfloor \frac{6,56}{11} + 0,5 \right\rfloor = \lfloor 1,096 \rfloor = 1 \quad (5.39)$$

Valoarea reconstruită este 11 și eroarea de cuantizare este $11 - 6,56 = 4,44$. Continuând în acest mod, se obțin eșantioanele din tabelul 5.7.

Se observă că, în urma cuantizării, în matricea coeficienților cuantizați, fenomenul de concentrare a valorilor mari în colțul din stânga sus, și predominanța valorilor mici (chiar zero) în rest, este mult mai accentuată.

Tabelul 5.7. Coeficienții cuantizați obținuți folosind tabelul de cuantizare al coeficienților

| | | | | | | | |
|----|---|---|---|---|---|---|---|
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| -9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

5.5.1. Modul de bază (baseline)

Pierderea de informație se datorează realizării împărțirii cu reținerea doar a părții întregi a rezultatelor. În acest fel valorile pierd din precizie (cele mici transformându-se în zero). Efectul acestei pierderi, însă, nu este sesizabil deoarece, după cum se observă, pierderile cele mai mari sunt concentrate la nivelul coeficienților de înaltă frecvență, care au pondere redusă în imagine și care, corespunzând detaliilor fine, sunt mai puțin observabile de către ochiul uman.

Rolul operației de cuantizare este acela de a obține cât mai multe valori nule sau mici, acestea având avantajul unei codări eficiente realizată ulterior. Transformata sinus discretă oferă posibilitatea de a realiza această operație astfel încât efectul pierderii de informație să fie cât mai redus.

Din tabelul 5.7 care conține eșantioanele cuantizate se poate observa că mărimea pasului crește pe măsura îndepărtării de coeficientul DC. Deoarece eroarea de cuantizare este o funcție crescătoare de mărimea pasului, coeficienții de înaltă frecvență vor fi afectați de o eroare de cuantizare mai mare decât cei de joasă frecvență. Decizia asupra mărimii relative a pasului de cuantizare se bazează pe modul de percepere a erorilor de sistemul vizual uman. Diferiți coeficienți ai transformării au importanță perceptuală diferită. Erorile de cuantizare din coeficienți DC și AC de joasă frecvență sunt mai ușor de detectat decât erorile de cuantizare pentru coeficienții AC de înaltă frecvență. De aceea se folosește un pas mai mare pentru coeficienții mai puțin importanți perceptual.

5.5.1. Modul de bază (baseline)

Deoarece cuantizoarele au nivelul 0 ca nivel de reconstrucție, procesul de cuantizare funcționează, de asemenea, și ca operație de codare de prag. Toți coeficienții cu amplitudinea mai mică decât jumătate din mărimea pasului vor fi zero. Deoarece mărimea pasului la sfârșitul scanării în zig-zag este mare, probabilitatea găsirii unei secvențe lungi de zero crește. Acest efect poate reprezenta o modalitate de modificare a ratei. Prin mărirea pasului, se poate reduce numărul de valori diferite de zero necesare pentru a fi transmise, ceea ce înseamnă o reducere a numărului de biți necesari.

- **Codarea elementelor din matricea coeficienților cuantizați**

Coeficienții cuantizați sunt scanați zig-zag, în scopul obținerii unei secvențe unidimensionale, ce va fi aplicată codorului entropic. Așa cum s-a mai precizat, primul coeficient se numește coeficient de curent continuu, DC, și reprezintă media intensității blocului. Ceilalți 63 de coeficienți se numesc coeficienți AC (coeficienți de curent alternativ). Scanarea în zig-zag se face în ideea ordonării după spectrul de frecvență. Deoarece componentele de frecvență înaltă au valori aproximativ nule, în urma scanării zig-zag rezultă un șir de zerouri la sfârșitul secvenței, dând posibilitatea realizării unei codări eficiente RLC (Run Length Coding) și Huffman.

În algoritmul de compresie JPEG sunt utilizate două proceduri de codare diferite. Prima procedură este utilizată pentru codificarea elementului I_{00} , care este coeficientul DC, a doua procedură utilizându-se pentru codificarea celorlalți 63 de coeficienți AC. Coeficientul DC este codat diferențial față de coeficientul DC din blocul anterior, folosind algoritmul DPCM (Differential Pulse Code Modulation). Coeficienții AC sunt codați RLC. Coeficienții DC și AC astfel codați vor fi codați apoi entropic utilizând codarea Huffman.

5.5.1. Modul de bază (baseline)

Codarea coeficienților DC

Din figura 5.13 se observă că matricea de bază corespunzătoare coeficienților DC este o matrice constantă, astfel încât coeficientul DC este media (sau un multiplu al acesteia) pixelilor din blocul de dimensiune 8×8 . Valoarea medie a pixelilor din orice bloc 8×8 nu va diferi substanțial de valoarea medie din blocul 8×8 vecin; de aceea valorile coeficienților DC vor fi relativ apropiate, motiv pentru care este eficient a coda diferența între acesta și valoarea coeficientului DC corespunzătoare blocului de 8×8 puncte anterior din aceeași categorie (Y, U sau V), decât etichetele în sine.

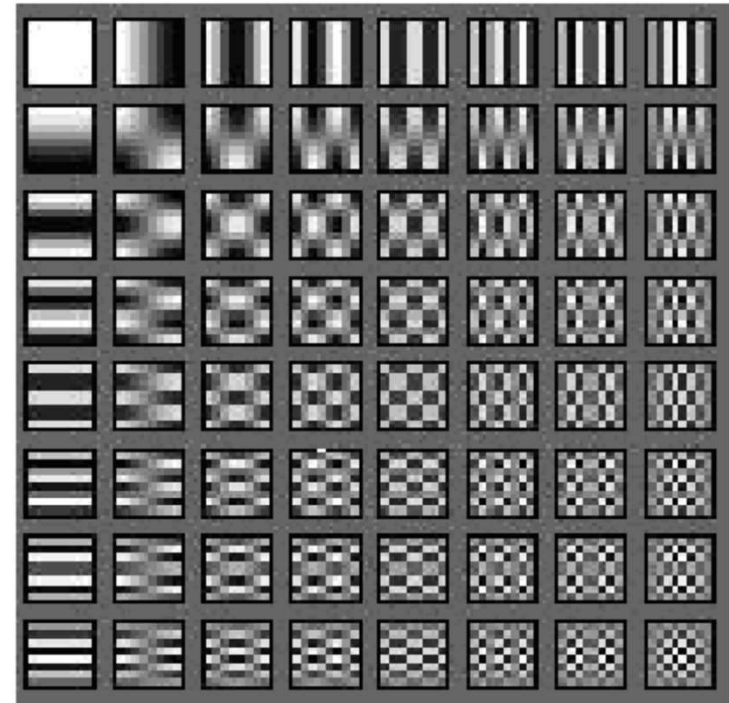


Fig. 5.13. Imaginile de bază pentru transformata discretă cosinus bi-dimenzională

5.5.1. Modul de bază (baseline)

În funcție de numărul de biți folosiți la codarea valorii pixelului, numărul de valori pe care le pot lua etichetele și, deci, diferențele dintre coeficienți, poate deveni destul de mare. Un cod Huffman pentru un alfabet așa mare ar fi greu de implementat. Recomandarea JPEG rezolvă această problemă prin partiționarea valorilor pe care le pot lua diferențele, în clase. Mărimea acestor clase crește în puteri ale lui doi. Astfel, clasa zero are un singur membru, clasa unu are doi membri, clasa doi are patru membri și așa mai departe. Numărul clasei este codat Huffman. Clasele și cuvintele de cod Huffman corespunzătoare acestora sunt prezentate în tabelul 5.8. Fiecare linie a tabelul 5.8 conține numere mai mari și mai multe decât ale liniei precedentei, neconținând numerele din linia precedentă. Linia i conține întregi din domeniul $[-(2i-1), + (2i-1)]$, din care lipsește intervalul din mijloc $[-(2i-1-1), +(2i-1-1)]$. În codarea coeficienților DC se transmite codul corespunzător clasei i (liniei din tabel) în care se încadrează numărul, urmat de i biți care reprezintă numărul coloanei din tabelul 5.8 în care se află numărul ce trebuie codat.

5.5.1. Modul de bază (baseline)

Numărul coloanei, C , este reprezentarea pe i biți a valorii diferenței, în cazul valorilor pozitive sau de cei mai puțin semnificativi i biți ai reprezentării valorii diferenței minus 1, în complement față de 2, dacă aceasta este negativă.

În cazul valorilor din tabelul 5.7, considerând că blocul nu este primul, eticheta corespunzătoare coeficientului DC este codată diferențial, adică se codează diferența dintre valoarea cuantizată a etichetei din acest bloc și valoarea cuantizată a esantionului din blocul anterior. Dacă blocul de procesat este primul, se transmite în clar valoarea coeficientului DC.

De exemplu, dacă valoarea diferenței care trebuie codificată este 21 și aparține componentei Y , se transmite întâi codul pentru clasa 5, care este 111110. Valoarea este pozitivă și se găsește în a 21-a coloană a liniei 5 (numărarea coloanelor începe cu 0), deci la acest cod trebuie adăugată reprezentarea pe 5 biți a numărului 21 care este 10101, codul complet fiind 11111010101.

Dacă, însă, valoarea care trebuie codată este -21, clasa este aceeași, diferența fiind valoarea negativă a coeficientului. Această valoare se găsește în coloana a 10-a a clasei 5. Așadar, fie se transmite reprezentarea pe 5 biți a numărului coloanei, 10, care este 01010, fie se transmit ultimii 5 biți ai reprezentării în complement față de 2 a numărului (-21), din care se scade 1, adică $(-22)_{2C} = 101010$, ai cărui ultimi 5 biți sunt identici cu ai numărului 10. Codul complet va fi în acest caz 11111001010.

5.5.1. Modul de bază (baseline)

În cazul coeficientului DC din tabelul 5.7, dacă se presupune că eticheta corespunzătoare din blocul anterior a fost -1, atunci diferența va fi 3. Din tabelul 5.8 se observă că această valoare aparține clasei 2. Deci, se va transmite codul Huffman pentru clasa 2, 110, urmat de o secvență de doi biți, 11, pentru a indica numărul coloanei în care se află numărul 3 sau valoarea din această clasă care a fost codată, adică, se va transmite 11011.

Numărul cuvintelor de cod Huffman este egal cu logaritmul în baza doi al numărului de valori posibile pe care le pot lua diferențele dintre etichete. Dacă diferențele pot lua 4096 de valori posibile, lungimea codului Huffman este $\log_2 4096 = 12$, număr folosit de obicei în codare.

Tabelul 5.8. Codarea diferențelor etichetelor DC

| Clasa | | | | | | | | Codul Huffman corespunzător clasei |
|-------|--------|----|--------|-------|-------|---|-------|---------------------------------------|
| 0 | | | | 0 | | | | 0 |
| 1 | | | -1 | | 1 | | | 10 |
| 2 | | -3 | -2 | | 2 | 3 | | 110 |
| 3 | -7 | . | -4 | | 4 | . | 7 | 1110 |
| 4 | -15 | . | -8 | | 8 | . | 15 | 11110 |
| 5 | -31 | . | -16 | | 16 | . | 31 | 111110 |
| 6 | -63 | . | -32 | | 32 | . | 63 | 1111110 |
| 7 | -127 | . | -64 | | 64 | . | 127 | 11111110 |
| 8 | -255 | . | -128 | | 128 | . | 255 | 111111110 |
| 9 | -511 | . | -256 | | 256 | . | 511 | 1111111110 |
| 10 | -1023 | . | -512 | | 512 | . | 1023 | 11111111110 |
| 11 | -2047 | . | -1024 | | 1024 | . | 2047 | 111111111110 |
| 12 | -4095 | . | -2048 | | 2048 | . | 4095 | 1111111111110 |
| 13 | -8191 | . | -4096 | | 4096 | . | 8191 | 11111111111110 |
| 14 | -16383 | . | -8192 | | 8192 | . | 16383 | 111111111111110 |
| 15 | -32767 | . | -16384 | | 16384 | . | 32768 | 111111111111111 |
| 16 | | | | 32768 | | | | |

5.5.1. Modul de bază (baseline)

În cazul codificării "normale" fiecare din acești coeficienți poate fi reprezentat pe 11 biți (1 bit de semn + 10 biți valoare). Acest mod de codificare duce practic la o reprezentare pe mai mulți biți decât în cazul imaginii necomprimată (se folosesc 11 biți pentru un coeficient, în loc de 8 biți pentru un pixel). Din acest motiv, în cazul algoritmului de compresie JPEG se recurge la o altă modalitate de codificare.

În acest caz se asociază un cod combinației dintre numărul de valori nule (dacă există) care precede un element diferit de zero și valoarea acestuia din urmă. Practic se codifică perechi (Număr de zerouri, Z – Valoare, x) în locul fiecărui coeficient în parte. În realitate, din considerente de reducere a numărului de combinații posibile, numărul de zerouri se limitează la 16. În cazul în care există mai mult de 16 elemente nule se emit coduri speciale (ZRL) care semnifică 16 zerouri care nu sunt urmate de un element diferit de zero. De exemplu, 18 zerouri urmate de un element cu valoarea -21 se vor coda printr-un ZRL urmat de codul corespunzător perechii (2,-21). De asemenea, în cazul în care dintr-un anumit punct al șirului până la sfârșitul acestuia nu mai există nici un element diferit de zero, se emite un alt cod special (EOB) în locul tuturor valorilor nule rămase.

5.5.1. Modul de bază (baseline)

În concluzie, pentru fiecare număr x , precedat de Z zerouri, care formează perechea (Z, x) , codorul trebuie:

- să găsească numărul de zerouri consecutive Z , care îl preced;

- să determine linia i și coloana C din tabelul 5.8 corespunzătoare numărului;

- să identifice din tabelul 5.9, după numărul Z și clasa i , codul Huffman corespunzător perechii;

- la cuvântul de cod Huffman găsit se concatenează reprezentarea pe i biți a coloanei C .

Ținând cont de aceste observații, pentru coeficienții din tabelul 5.7, rezultă următorul set de coduri care trebuie generate:

(0, 1)

(0, -9)

(0, -3)

(0, 0, ... 0) = EOB

Tabelul 5.9. Codarea HUFFMAN pentru coeficienții AC

| Z/i | Cuv. de cod | Z/i | Cuv. de cod | Z/i | Cuv. de cod |
|----------|-------------|-----|-------------|----------|------------------|
| 0/0(EOB) | 1010 | | | F/0(ZRL) | 11111111001 |
| 0/1 | 00 | 1/1 | 1100 | F/1 | 111111111110101 |
| 0/2 | 01 | 1/2 | 11011 | F/2 | 1111111111110110 |
| 0/3 | 100 | 1/3 | 1111001 | F/3 | 1111111111110111 |
| 0/4 | 1011 | 1/4 | 111110110 | F/4 | 1111111111111000 |
| 0/5 | 11010 | 1/5 | 11111110110 | F/5 | 1111111111111001 |
| ... | | | | | |

5.5.1. Modul de bază (baseline)

De exemplu, se dorește codificarea simbolului (0,1) din șirul de mai sus. Prima valoare, 1, aparține categoriei 1. Deoarece nu sunt zerouri care să precedă această valoare, se transmite codul Huffman corespunzător lui 0/1, care, din tabelul 5.10, este 00. Se concatenează cu acest cod un singur bit de 1 pentru a indica faptul că valoarea transmisă este 1. Cuvântul de cod pentru perechea (0,1) este deci 001. Analog, -9 este al șaselea element din clasa 4. De aceea se transmite șirul binar 1011, care este codul Huffman pentru 0/4, urmat de 0110, pentru a arăta că -9 este cel de-al șaselea element din clasă. Următoarea valoare este 3, care aparține clasei 2, deci se transmite codul Huffman 01, corespunzător lui 0/2, urmat de 11. Toate esantioanele după acest punct sunt zero, astfel încât se transmite codul Huffman EOB, care în acest caz este 1010.

Pentru exemplul considerat, datele sunt 11011 001 10110110 0111 1010, adică se folosește un număr de 24 de biți pentru reprezentarea blocului de dimensiune 8x8, adică o medie de 3/8 biți/pixel.

Se precizează că tabele prezentate pentru cuantizarea și codarea coeficienților DC și AC nu sunt singurele recomandate de standardul JPEG, acesta limitând însă la 4, tabelele de coduri Huffman pentru codarea coeficienților AC pentru luminanță și crominanță. Modul JPEG de bază folosește numai două asemenea tabele.

5.5.1. Modul de bază (baseline)

Lanțul de decodare JPEG este parcurs în ordine inversă codării. Astfel, imaginea compresată JPEG este supusă în primul pas unui decodor entropic. După decodarea entropică, se aplică decuantizarea, folosind aceiași coeficienți care au fost folosiți și la cuantizare, prezentați în tabelele 5.6, *a*, *b*. În urma decuantizării se obțin coeficienții transformatei cosinus discrete din tabelul 5.10. Acești coeficienți sunt “trimiși” blocului de transformare cosinus discretă inversă, IDCT, care aplică transformarea dată de relația 5.36. Aceștia li se adună 128 și se obține blocul reconstruit prezentat în tabelul 5.11. Calitatea acestei imagini depinde de numărul de coeficienți păstrați la codare. Dacă se vor păstra toți coeficienții nenuli, atunci imaginea reconstruită va fi foarte asemănătoare cu originalul.

Tabelul 5.10. Valorile coeficienților după decuantizare

| | | | | | | | |
|------|----|---|---|---|---|---|---|
| 32 | 11 | 0 | 0 | 0 | 0 | 0 | 0 |
| -108 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 42 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Tabelul 5.11. Blocul reconstruit

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 123 | 122 | 122 | 121 | 120 | 120 | 119 | 119 |
| 121 | 121 | 121 | 120 | 119 | 118 | 118 | 118 |
| 121 | 121 | 120 | 119 | 119 | 118 | 117 | 117 |
| 124 | 124 | 123 | 122 | 122 | 121 | 120 | 120 |
| 130 | 130 | 129 | 129 | 128 | 128 | 128 | 127 |
| 141 | 141 | 140 | 140 | 139 | 138 | 138 | 137 |
| 152 | 152 | 151 | 151 | 150 | 149 | 149 | 148 |
| 159 | 159 | 158 | 157 | 157 | 156 | 155 | 155 |

5.5.1. Modul de bază (baseline)

Deși reducerea este substanțială, de la 8 biți pe pixel la 3/8 biți pe pixel, reproducerea este remarcabil de apropiată de original.

Dacă se dorește o reproducere cu acuratețe mai mare, se poate face acest lucru cu prețul creșterii ratei, micșorând mărimea pasului de cuantizare din tabelul de cuantizare. Acest lucru va determina o creștere a numărului de biți transmiși. Similar, se poate scădea rata, prin creșterea pasului de cuantizare, cu prețul creșterii distorsiunilor.

5.5.2. Modul de codare cu pierderi expandat

Există trei diferențe principale între modul de bază și cel expandat, și anume:

- Modul expandat poate folosi 8-12 biți;
- Modul expandat poate folosi ca și codare entropică atât codarea Huffman, cât și codarea aritmetică;
- Modul expandat poate folosi atât codarea progresivă cât și cea secvențială.

În multe dintre aplicațiile de compresie a imaginilor apare dezavantajul că, datorită rezoluțiilor mari la care sunt reprezentate imaginile, timpul de codare, de transmisie și, respectiv, de decompresie este de ordinul minutelor. În astfel de aplicații se poate aplica un algoritm de compresie progresiv, care inițial produce o imagine brută, după care, prin scanări succesive, se îmbunătățește calitatea imaginii.

5.5.2. Modul de codare cu pierderi expandat

Dacă se presupune că valoarea unui pixel este apropiată de cea a vecinului său, se poate folosi valoarea pixelului învecinat pentru a prezice valoarea pixelului de codat. Ideea este aceea de a îndepărta orice redundanță care poate exista. Diferența dintre valoarea actuală și cea prezisă este codată și transmisă. Această diferență se numește *eroare de predicție* sau *reziduu*. Receptorul folosește același pixel învecinat pentru a face predicția pentru acel pixel și același algoritm de predicție ca și emițătorul. Din moment ce se folosește același pixel și același algoritm pentru a face predicția, receptorul ar trebui să genereze aceeași valoare de predicție ca și emițătorul. Această valoare, când este adăugată la eroarea de predicție dată de emițător, ar trebui să conducă exact la pixelul original recuperat. Dacă algoritmul folosit pentru predicție constă în alegerea unei combinații liniare a pixelilor învecinați, atunci această abordare se numește *predicție liniară*. Pentru ca, atât emițătorul cât și receptorul să folosească același pixel pentru generarea predicției, trebuie să se impună o ordine a pixelilor. În general, se presupune că pixelii unei imagini sunt generați linie cu linie, de la stânga la dreapta, și de sus în jos. Acest procedeu se numește ordine de scanare de tip rastru.

5.5.2. Modul de codare cu pierderi expandat

De exemplu, se consideră că "imaginea" din figura 5.15, *a* este imaginea originală. Dacă se folosește vecinul stâng al fiecărui pixel ca predicție a aceluși pixel, eroarea de predicție poate fi reprezentată ca o imagine reziduală, după cum se arată în figura 5.15, *b*.

Se observă numărul mare de zerouri din imaginea reziduală. Într-o imagine în care există cu precădere acest tip de redundanță sau structură, adică pixeli vecini au valori apropiate una de alta, această abordare va duce la o imagine reziduală care constă în principal din zerouri și numere mici. Imaginea reziduală poate fi codificată în general cu mult mai puțini biți decât imaginea originală. În acest exemplu se folosește vecinul din stânga pentru predicție. Dar se poate folosi, de asemenea, vecinul de sus sau o combinație avantajoasă a lor. Schemele de predicție pentru pixelul curent care se bazează pe vecini aflați pe direcții diferite sunt cunoscute sub denumirea de scheme de predicție bi-dimensionale. În ciuda simplității lor, tehnicile de predicție liniară sunt eficiente, iar performanțele lor sunt surprinzător de apropiate de performanțele obținute cu ajutorul altor tehnici mai sofisticate.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 8 | 4 | 8 | 1 | 1 | 1 | 1 |
| 1 | 2 | 4 | 6 | 5 | 1 | 1 | 1 |
| 8 | 4 | 5 | 5 | 5 | 5 | 5 | 5 |
| 2 | 4 | 8 | 5 | 7 | 9 | 5 | 5 |
| 2 | 4 | 6 | 7 | 7 | 7 | 9 | 9 |
| 2 | 2 | 2 | 3 | 4 | 9 | 7 | 3 |
| 3 | 3 | 6 | 6 | 6 | 6 | 7 | 7 |
| 7 | 7 | 7 | 7 | 6 | 7 | 7 | 7 |

a)

| | | | | | | | |
|---|----|----|----|----|----|----|----|
| 4 | 4 | -4 | 4 | -7 | 0 | 0 | 0 |
| 1 | 1 | 2 | 2 | -1 | -4 | 0 | 0 |
| 8 | -4 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 2 | 4 | -3 | 2 | 2 | -4 | 0 |
| 2 | 2 | 2 | 1 | 0 | 0 | 2 | 0 |
| 2 | 0 | 0 | 1 | 1 | 5 | -2 | -4 |
| 3 | 0 | 3 | 0 | 0 | 0 | 1 | 0 |
| 7 | 0 | 0 | 0 | -1 | 1 | 0 | 0 |

b)

Fig. 5.15. (a) Imaginea originală și (b) eroarea ei de predicție

5.5.2. Modul de codare cu pierderi expandat

Modul de compresie progresiv JPEG, se obține prin realizarea unui set de subimagini și fiecare subimagine va fi codată cu un set specific de coeficienți DCT. În consecință, codorul DCT va trebui să aibă un buffer în care datele (coeficienții DCT ai subimaginilor) să fie memorate înainte de codarea entropică. În figura 5.16 este prezentat modul de formare a imaginilor codate progresiv JPEG.

Compresia progresivă JPEG poate fi obținută folosind trei tipuri de algoritmi:

- un algoritm progresiv de selecție spectrală;
- un algoritm progresiv de aproximări succesive;
- un algoritm progresiv combinat.



Fig. 5.16. Compresia progresivă JPEG

5.5.2. Modul de codare cu pierderi expandat

Algoritmul progresiv de selecție spectrală grupează coeficienții DCT în câteva benzi de frecvență. De obicei, întâi sunt transmiși coeficienții de joasă frecvență, după care urmează coeficienții de înaltă frecvență. În figura 5.17 este prezentat un exemplu în care coeficienții DCT sunt împărțiți în patru benzi de frecvență. În banda 1 se găsesc numai coeficienții de curent continuu (DC), banda 2 cuprinde primii coeficienți AC, AC1, AC2, banda 3 conține coeficienții AC3, AC4, AC5, AC6, iar banda 4, coeficienții AC7,... , AC63.

Algoritmul progresiv cu aproximări succesive se bazează pe transmisia inițială, cu precizie scăzută, a tuturor coeficienților DCT, după care se transmite restul informației pentru a se obține o imagine de calitate superioară. În figura 5.18 este prezentat un exemplu în care coeficienții DCT sunt împărțiți în trei benzi de aproximări succesive: banda 1, care cuprinde toți coeficienții DCT împărțiți la 4, banda 2, cuprinzând coeficienții împărțiți la 2 și banda 3 conținând toți coeficienții DCT la precizia nominală.

✓ Spectral selection

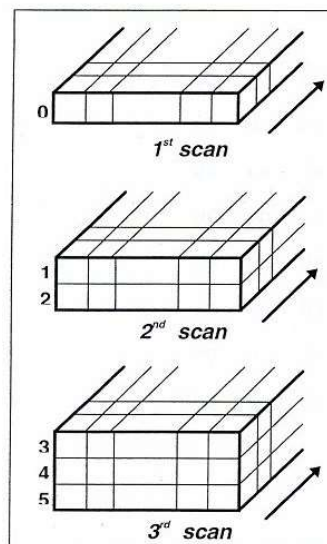


Fig. 5.17. Algoritmul progresiv cu selecție spectrală

✓ Successive approximation

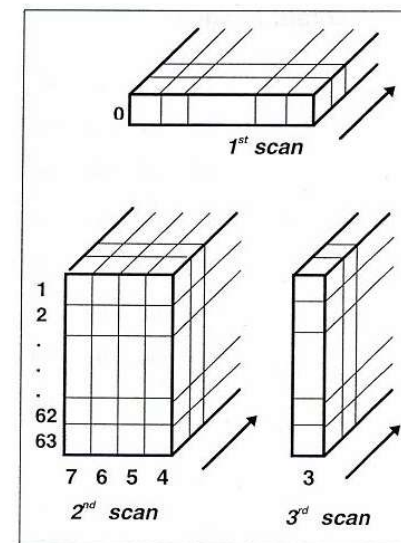


Fig. 5.18. Algoritmul progresiv cu aproximări succesive

5.5.2. Modul de codare cu pierderi expandat

Algoritmul progresiv combinat este o combinație a primilor doi algoritmi. În figura 5.19 este prezentat planul coeficienților DCT ai unei imagini, după aplicarea algoritmului combinat. S-au obținut 8 subsecvențe de coeficienți DCT, care vor fi transmiși în ordinea numerotării, și anume, coeficienții DC1 apoi AC1 și așa mai departe.

În figura 5.20 este prezentată imaginea decompresată progresiv JPEG pentru diferite momente de timp.

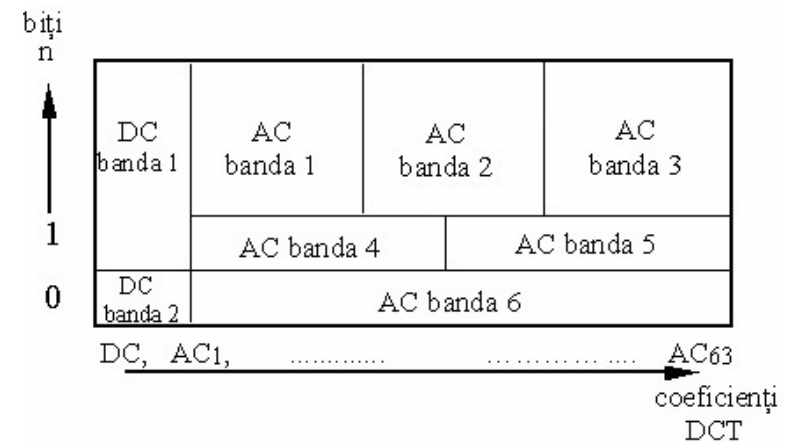


Fig. 5.19. Algoritmul progresiv combinat

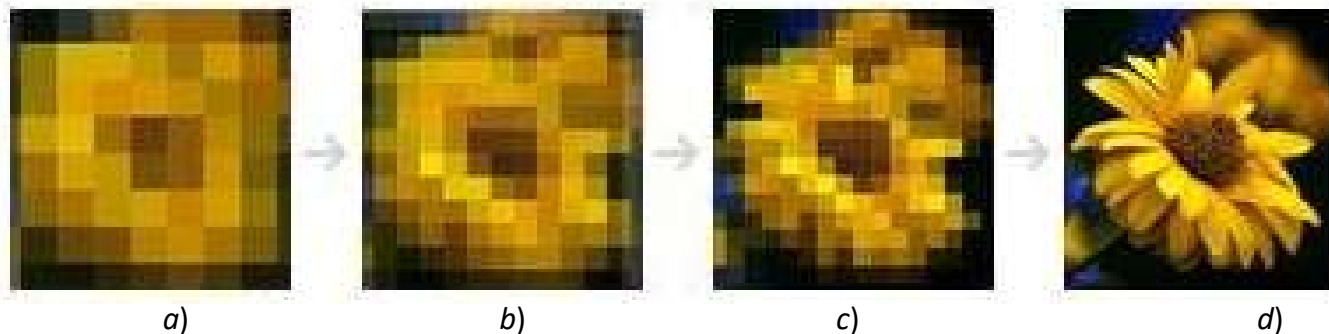


Fig. 5.20. Etapele de decompresie ale imaginii folosind algoritmul progresiv

- a) imagine la momentul t_1 b) imagine la momentul $t_2 > t_1$
 c) imagine la momentul $t_3 > t_2$ d) imagine originală ($t = t_4$)

5.5.3. Compresia secvențială JPEG fără pierderi

Standardul JPEG permite și folosirea unui algoritm de compresie fără pierderi, respectiv un algoritm de compresie predictivă în locul transformării DCT. În figura 5.21 este prezentată schema bloc a unui codor JPEG fără pierderi. Blocurile de cuantizare neuniformă din schema standard de compresie JPEG sunt înlocuite în cazul de față cu un bloc de codare predictivă.

Blocul predictor lucrează astfel: se calculează o predicție a eșantionului X' pe baza eșantioanelor precedente A, B și C, arătate în figura 5.22, după care se calculează diferența dintre eșantionul original X și cel prezis: $DX = X - X'$.

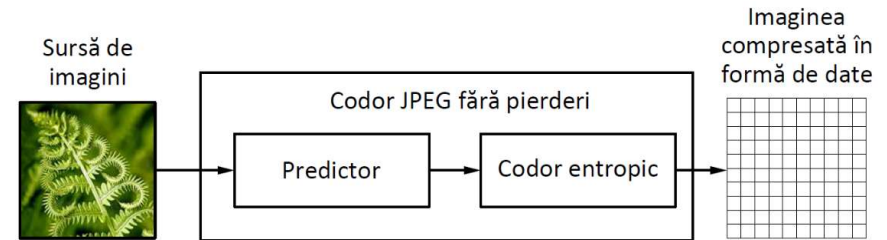


Fig. 5.21. Schema generalizată a codorului JPEG fără pierderi

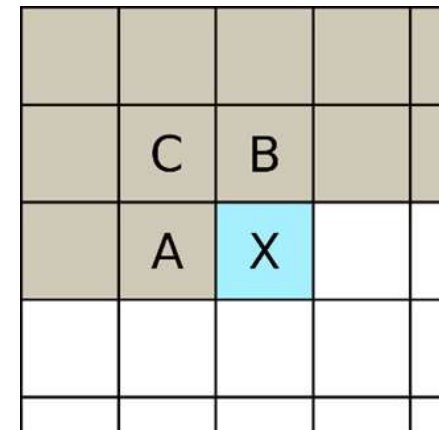


Fig. 5.22. Poziționarea eșantioanelor pe baza cărora se calculează predicția

5.5.3. Compresia secvențială JPEG fără pierderi

Această diferență este codată entropic cu ajutorul unui codor Huffman. În figura 5.23 sunt prezentați pașii parcurși în algoritmul de codare.

În tabelul 5.12 sunt date câteva formule folosite în predicție.

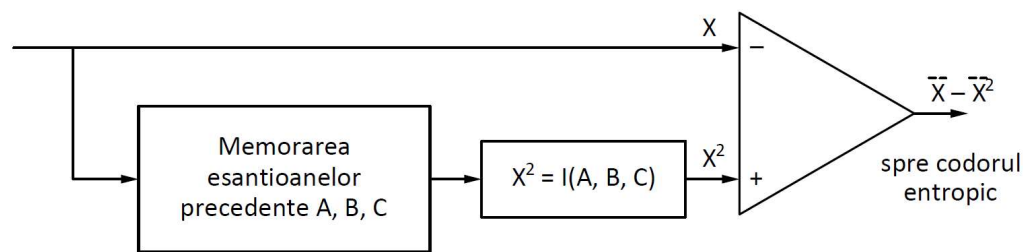


Fig. 5.23. Schema de codare cu pierderi JPEG

Tabelul 5.12. Predictorii folosiți pentru compresia cu pierderi

| Index | Formula de predicție |
|-------|----------------------|
| 0 | fără predictor |
| 1 | $X=A$ |
| 2 | $X=B$ |
| 3 | $X=C$ |
| 4 | $X=A+B-C$ |
| 5 | $X=A+(B-C)/2$ |
| 6 | $X=B+(A-C)/2$ |
| 7 | $X=(A+B)/2$ |

5.5.4. Compresia JPEG ierarhică

Compresia ierarhică JPEG permite o reprezentare progresivă a imaginii decodate, într-un mod similar cu algoritmul progresiv, dar, în plus față de acesta, permite imagini codate cu rezoluții multiple. Se creează astfel, un set de imagini compresate, începând cu imagini de rezoluție mică și continuând cu imagini a căror rezoluție crește către rezoluția imaginii originale. Acest proces se numește subeșantionare sau codare piramidală. În figura 5.24 este figurat un astfel de algoritm.

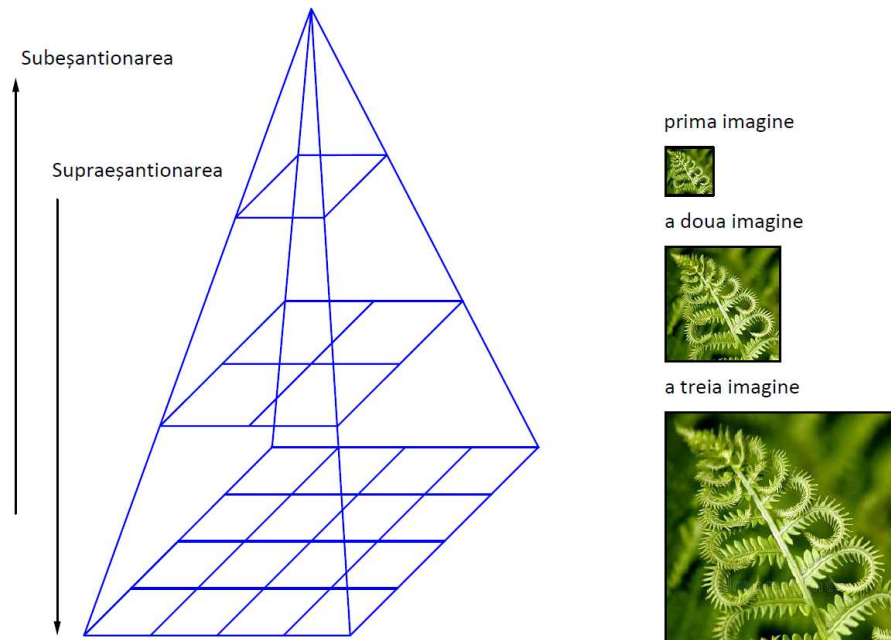


Fig. 5.24. Codare ierarhică JPEG

5.5.4. Compresia JPEG ierarhică

După procesul de subeșantionare, fiecare imagine de rezoluție joasă se scalează la imaginea cu rezoluție imediat superioară prin metoda inversă, numită supraeșantionare, fiind folosită ca predictor pentru următoarea etapă. Algoritmul de compresie ierarhică constă din următorii pași:

- filtrarea și subeșantionarea imaginii originale cu factorul dorit (multiplu al lui 2), de-a lungul fiecărei dimensiuni (vertical, orizontal);
- codarea imaginii de rezoluție mică folosind tehnicile de codare DCT secvențială, DCT progresivă sau codarea fără pierderi;
- decodarea imaginii de rezoluție redusă, interpolarea și supraeșantionarea cu un factor de 2, pe direcția orizontală și/sau verticală, folosind un filtru de interpolare identic cu cel de la decodor;
- folosirea imaginii supraeșantionate ca predictor al imaginii originale, și codarea diferențelor dintre cele două imagini, folosind una din tehnicile amintite;
- se repetă algoritmul până când se codează imaginea la rezoluția inițială.

Codarea ierarhică necesită o zonă de memorie destul de mare pentru a putea implementa algoritmul, dar avantajele sunt acelea că imaginea codată este imediat disponibilă la diferite rezoluții.

<https://www.youtube.com/watch?v=CJFUN6BrkGE>

КАК РАБОТАЕТ СЖАТИЕ?

ÎNTREBĂRI