

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

# 5.1. Reprezentarea numerică a imaginilor

O imagine este o suprafață de obicei dreptunghiulară caracterizată, la nivelul oricărui punct al ei, de o anumită culoare. Ideal, culoarea variază continuu în oricare direcție. Din păcate, în sistemele numerice, nu se pot utiliza mărimi care variază continuu, ci doar forma discretizată a acestora.

Astfel, o imagine trebuie să fie discretizată înainte de a se pune problema reprezentării numerice. Discretizarea constă în împărțirea imaginii într-un caroiaj asemănător unei table de șah. Fiecare porțiune de imagine delimitată de acest caroiaj va fi considerată ca având o culoare uniformă – o medie a culorii existente pe această secțiune. Aceste secțiuni mai sunt denumite și pixeli sau puncte, numărul acestora definind rezoluția imaginii. De exemplu, pentru o imagine oarecare care are o rezoluție de 640 x 480 pixeli, înseamnă că pe suprafața acesteia s-a definit un caroiaj care o împarte pe orizontală în 640 de secțiuni iar pe verticală, în 480.

Pasul următor îl constituie găsirea unei reprezentări pentru culoare. Orice culoare poate fi descompusă în trei culori primare (de exemplu roșu – R, verde – G și albastru – B), cu alte cuvinte orice imagine poate fi obținută prin suprapunerea aditivă a trei radiații luminoase având aceste trei culori și intensități diferite. Deci, pentru a reprezenta numeric o culoare, este suficient să se reprezinte intensitățile luminoase ale celor trei culori primare.

## 5.1. Reprezentarea numerică a imaginilor

Dacă se alocă câte 8 biți pentru fiecare componentă, se pot coda 256 nivele de intensitate, astfel, absența culorii (intensitate zero) se codifică prin valoarea 00000000 în binar sau 00 în hexazecimal, iar intensitatea maximă, prin cea mai mare valoare ce poate fi reprezentată pe 8 biți, și anume, 11111111 în binar sau FF în hexazecimal. Această reprezentare, însă, ține mai mult de modalitățile tehnice de captare și reproducere a imaginii și mai puțin de mecanismul fiziologic de percepere a culorii. Prin diferite experimente s-a constatat că din punct de vedere al capacității de percepere a detaliilor, ochiul este mai sensibil la intensitatea luminoasă a culorii decât la nuanță. Din acest motiv prezintă interes o altă modalitate de reprezentare a culorii care să țină cont de această observație, un exemplu fiind reprezentarea YUV utilizată în televiziunea în culori. În acest caz, în locul celor trei componente primare R,G,B se utilizează alte trei mărimi derivate din acestea, și anume:

$$\begin{cases} Y = 0,3R + 0,59G + 0,11B \\ U = R - Y = 0,7T - 0,59G - 0,11B \\ V = B - Y = -0,3R - 0,59 + 0,89B \end{cases} \quad (5.1)$$

## 5.1. Reprezentarea numerică a imaginilor

În cazul acestei reprezentări, componenta Y corespunde *intensității luminoase* percepute pentru respectiva culoare (coeficienții 0,30, 0,59 și 0,11 reprezintă *strălucirile* relative la alb ale celor trei culori primare roșu, verde și, respectiv, albastru). Această componentă mai este întâlnită și sub numele de *luminanță*. Componentele U și V sunt cele care definesc nuanța culorii, din acest motiv, sunt denumite *componente de crominanță*. Acestea se calculează ca diferența dintre componenta roșie, respectiv albastră, și cea de luminanță.

Avantajul reprezentării YUV este acela că separă componenta de luminanță, pentru care ochiul este foarte sensibil la detalii, de componentele de nuanță pentru care sensibilitatea este mai redusă. Acest lucru face posibilă reducerea informației asociate unei imagini prin utilizarea unei rezoluții mai reduse pentru componentele de crominanță. În cazul televiziunii în culori se realizează o "compresie" prin limitarea benzii de frecvență alocate semnalelor de crominanță (de exemplu în sistemul PAL semnalele U și V au o bandă de 1,3 MHz față de semnalul Y care are o bandă de 6 MHz).

## 5.2. Reprezentarea imaginii în format necompresat

O imagine se reprezintă ca o matrice de puncte (de obicei de forma pătrată), fiecare punct fiind caracterizat de o culoare. De exemplu, pentru imaginea din figura 5.1, *a* se poate evidenția acest lucru dacă se mărește o secțiune a imaginii astfel încât matricea de puncte să devină vizibilă, ca în figura 5.1, *b*.

Pentru a reprezenta o astfel de imagine trebuie să se utilizeze un mod de reprezentare numeric al culorii. Pentru aceasta se pornește de la observația că orice culoare poate fi obținută prin amestecul în diferite proporții a trei culori de bază (culori primare). În practică se utilizează ROȘU (R), VERDE (G) și ALBASTRU (B). Intensitatea luminoasă a unei culori primare poate fi reprezentată numeric sub forma unui întreg de 8 biți, valoarea 0 corespunzând intensității nule iar cea maximă (255) intensității maxime. În acest fel, o culoare va fi reprezentată numeric printr-un triplet de întregi pe 8 biți (R,G,B). De exemplu culoarea GALBEN va avea o reprezentare de forma (255, 255, 0). În aceste condiții imaginea se reprezintă sub forma unei matrice  $IM(N_x, N_y)$  unde  $N_x$  reprezintă numărul de puncte pe orizontală și  $N_y$  este numărul de puncte pe verticală, iar elementele matricei sunt tripleți de întregi pe 8 biți de tip (R,G,B).



*a)*



*b)*

Fig. 5.1. Imagine reprezentată ca o matrice de puncte

## 5.3. Metode și abordări ale compresiei imaginii

În continuare se redau câteva metode folosite în compresie, evidențiind aplicabilitatea lor în compresia de imagini.

1. *Cuantizarea scalară* poate fi folosită pentru a comprima imagini, dar performanțele ei sunt mediocre. De exemplu, o imagine cu 8 biți/pixel poate fi compresată prin cuantizare scalară eliminând cei mai ne semnificativi patru biți ai fiecărui pixel. Aceasta conduce la o rată de compresie de 0,5, care pe lângă faptul că nu este semnificativă, determină în același timp și reducerea numărului de culori (sau nuanțe de gri) de la 256 la doar 16. O astfel de reducere nu numai că descrește pe ansamblu calitatea imaginii reconstruite, dar poate crea benzi de diferite culori, un efect observabil și deranjant.



## 5.3. Metode și abordări ale compresiei imaginii

### *Exemplu*

Se consideră, de exemplu, un rând de 12 pixeli de culori similare, pornind de la 202 la 215. În notație binară aceste valori sunt:

```
11010111 11010110 11010101 11010011 11010010 11010001  
11001111 11001110 11001101 11001100 11001011 11001010.
```

Cuantizarea va produce următoarele 12 valori de 4 biți:

```
1101 1101 1101 1101 1101 1101 1100 1100 1100 1100 1100 1100,
```

din care se vor reconstrui cei 12 pixeli, prin adăugarea a 4 zerouri, fiecărei valori cuantizate:

```
11010000 11010000 11010000 11010000 11010000 11010000  
11000000 11000000 11000000 11000000 11000000 11000000.
```

Primii șase pixeli ai rândului acum au valoarea  $11010000_2 = 208$ , în timp ce următorii șase pixeli sunt  $11000000_2 = 192$ . Dacă rânduri adiacente au pixeli similari, primele șase coloane vor forma o bandă, clar diferită de banda formată de următoarele șase coloane. Acest fenomen de formare a benzilor, sau de conturare, este foarte evident pentru ochi, deoarece aceștia sunt sensibili la margini și rupturi într-o imagine.



## 5.3. Metode și abordări ale compresiei imaginii

2. *Cuantizarea vectorială* poate fi folosită cu mai mult succes pentru a comprima imagini.

Cuantizarea vectorială (CV) în sensul sau cel mai general este aproximarea unui semnal de amplitudine continuă printr-un semnal de amplitudine discretă (alcătuit din semnale distincte). În cazul aplicației noastre CV constă în a reprezenta un vector  $x$  de dimensiune  $k$  printr-un vector  $y$  de aceeași dimensiune aparținând unui set finit numit dicționar.

3. *Metodele statistice* funcționează mai bine când simbolurile ce trebuie compresate au probabilități diferite. O secvență de intrare în care mesajele au aceeași probabilitate nu se va comprima eficient. În acest sens, într-o imagine alb-negru sau color în tonuri continue, diferitele culori sau nuanțe de gri se dovedesc de multe ori a avea aproximativ aceleași probabilități. De aceea metodele statistice nu sunt o alegere bună pentru compresia unor astfel de imagini, și sunt necesare noi abordări. Imaginile cu discontinuități de culoare, în care pixeli adiacenți au culori foarte diferite, se compresează mai bine cu metodele statistice, dar în acest caz nu este ușoară predicția pixelilor (Operație rațională de anticipare a unui eveniment sau fenomen).

## 5.3. Metode și abordări ale compresiei imaginii

4. *Metodele de compresie bazate pe dicționar* tind, de asemenea, să nu aibă succes în cazul imaginilor cu tonuri continue. O astfel de imagine conține de obicei pixeli adiacenți în culori similare, dar nu conține modele repetitive. Chiar și o imagine care conține modele repetitive, cum sunt liniile verticale, le poate pierde când este digitizată. O linie verticală în imaginea originală poate deveni ușor oblică atunci când imaginea este digitizată. O linie verticală ideală este prezentată în figura 5.2, *a*. În figura 5.2, *b* linia este presupusă a fi perfect digitizată în zece pixeli, așezați vertical. Totuși, dacă imaginea este plasată în digitizor ușor oblic, procesul de digitizare poate fi imperfect, și pixelii rezultați pot arăta ca în figura 5.2, *c*.

O altă problemă a compresiei imaginilor bazate pe dicționar este aceea că astfel de metode scanează imaginea rând cu rând, și pot pierde astfel corelații verticale între pixeli. Un exemplu sunt cele două imagini din figura 5.3, *a*, *b*. Salvând ambele imagini în GIF89, un format de fișier grafic bazat pe dicționar, au rezultat fișiere de dimensiuni 1053, respectiv 1527 octeți.

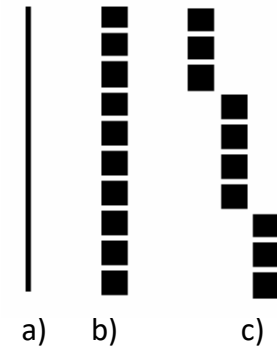


Fig. 5.2. Digitizare perfectă și imperfectă

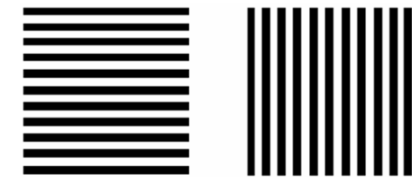


Fig. 5.3. Compresia bazată pe dicționar a liniilor paralele

## 5.3. Metode și abordări ale compresiei imaginii

Metodele tradiționale sunt nesatisfăcătoare pentru compresia de imagini, astfel încât au fost necesare abordări noi, care, deși diferite, se bazează pe eliminarea redundanței din imagine, folosind următorul principiu:

*Principiul compresiei de imagine.* Dacă se selectează aleator un pixel dintr-o imagine, există o probabilitate mare ca vecinii săi să aibă aceeași culoare sau culori foarte apropiate.

Compresia de imagine este, deci, bazată pe faptul că pixelii învecinați sunt puternic corelați. Această corelare se numește și *redundanță spațială*.

### *Exemplu*

În continuare este prezentat un exemplu simplu care ilustrează cum poate fi eliminată redundanța dintr-un șir de pixeli corelați. Următoarea secvență de valori reprezintă intensitățile a 24 de pixeli adiacenți dintr-un rând al unei imagini în tonuri continue: 12, 17, 14, 19, 21, 26, 23, 29, 41, 38, 31, 44, 46, 57, 53, 50, 60, 58, 55, 54, 52, 51, 56, 60.

Doar doi din cei 24 de pixeli sunt identici. Valoarea lor medie este 40,3. Scăzând perechi de pixeli adiacenți rezultă secvența: 12, 5, -3, 5, 2, 4, -3, 6, 11, -3, -7, 13, 4, 11, -4, -3, 10, -2, -3, 1, -2, -1, 5, 4.

Cele două secvențe sunt ilustrate în figura 5.4.

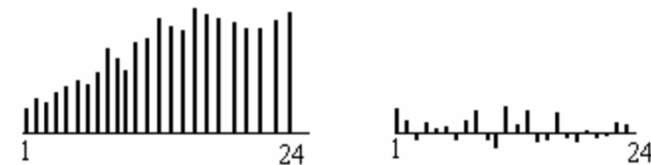


Fig. 5.4: Valorile și diferențele celor 24 pixeli adiacenți

## 5.3. Metode și abordări ale compresiei imaginii

Secvența de valori diferență are trei proprietăți care ilustrează potențialul ei de compresie:

1. Valorile secvenței diferență sunt mai mici decât valorile pixelilor originali. Media lor este de 2,58.
2. Există valori ale secvenței diferențelor care se repetă. Există doar 15 valori diferență distincte, deci, în principiu ele pot fi codate prin patru biți fiecare.
3. Valorile secvenței diferență sunt decorelate, adică valori diferență adiacente tind a fi diferite. Aceasta poate fi observată dacă se efectuează o nouă scădere a lor, rezultând secvența 12, - 7, - 8, 8, - 3, 2, - 7, 9, 5, - 14, - 4, 20, - 11, 7, - 15, 1, 13, - 12, - 1, 4, - 3, 1, 6, 1. Valorile acestea sunt mai mari decât diferențele anterioare.

În general, metodele de compresie pentru imagini sunt proiectate pentru un tip particular de imagine și în continuare se prezintă câteva din aceste metode specifice. Imaginile particulare vizate sunt imagini cu două nivele, imagini cu tonuri de gri și imagini color.

## 5.3. Metode și abordări ale compresiei imaginii

*Abordarea 1.* Aceasta este folosită pentru imagini cu două nivele. Un pixel dintr-o astfel de imagine este reprezentat printr-un bit. Aplicând principiul compresiei de imagine asupra unei imagini cu două nivele, înseamnă că pixelii învecinați ai unui pixel  $P$  tind a fi identici cu  $P$ . Astfel, are sens folosirea unei codări RLC (Run length coding) pentru a comprima o astfel de imagine. O metodă de compresie pentru o astfel de imagine o poate scana în ordinea rastrului (rând cu rând), calculând lungimile șirurilor de pixeli albi și negri. Aceste lungimi sunt codate prin coduri de lungime variabilă și sunt înscrise în secvența compresată. Un exemplu de astfel de metodă este compresia facsimil.

Ar trebui accentuat faptul că aceasta este doar o abordare a imaginilor cu două nivele. În practică, detaliile metodelor particulare diferă, în funcție de aplicație. De exemplu, o metodă poate scana imaginea coloană cu coloană, sau în zig-zag, sau o poate scana regiune cu regiune.

## 5.3. Metode și abordări ale compresiei imaginii

*Abordarea 2* se aplică, de asemenea, pentru imagini cu două nivele. Principiul compresiei de imagine spune că vecinii unui pixel tind să fie similari lui. Se poate extinde acest principiu și concluziona că dacă pixelul curent are culoarea  $c$  (unde  $c$  este ori alb, ori negru), atunci pixelii de aceeași culoare anteriori și următori tind să aibă aceiași vecini imediați.

Această abordare urmărește  $n$  vecini apropiați ai pixelului curent și îi consideră ca un număr de  $n$  biți. Acest număr se numește *contextul* pixelului. În principiu pot exista  $2^n$  contexte, dar datorită redundanței imaginii, distribuția lor este neuniformă. Unele contexte ar trebui să fie foarte frecvente, iar celelalte, rare. Codorul numără de câte ori a fost găsit deja fiecare context pentru un pixel de culoarea  $c$ , și asignează corespunzător probabilități acestor contexte. Dacă pixelul curent are culoarea  $c$  și contextul său are probabilitatea  $p$ , codicatorul poate folosi coduri aritmetice adaptive pentru a codifica pixelul cu acea probabilitate. Această abordare este folosită de standardul JBIG (Joint Bi-level Processing Group).

În continuare, se consideră imagini în nuanțe de gri. Un pixel dintr-o astfel de imagine este reprezentat prin  $n$  biți și poate avea una din  $2^n$  valori. Aplicarea principiului compresiei de imagine asupra unei imaginii în nuanțe de gri implică faptul că vecinii imediați ai unui pixel  $P$  tind să fie similari cu  $P$ , însă nu în mod necesar identici cu el. Astfel, nu mai poate fi folosită codarea RLE (run length encoding) pentru compresia unei astfel de imagini, ci sunt folosite următoarele două abordări.

## 5.3. Metode și abordări ale compresiei imaginii

*Abordarea 3.* Se separă imaginea în tonuri de gri în  $n$  imagini pe două nivele și apoi se compresează fiecare cu un cod RLE instantaneu. Principiul compresiei de imagini, în acest caz, s-ar formula prin afirmația că doi pixeli adiacenți care sunt similari în imaginea în tonuri de gri vor fi identici în cele mai multe imagini cu două nivele. Acest lucru însă nu este adevărat, așa cum reiese din următorul exemplu.

### *Exemplu*

Se consideră o imagine în tonuri de gri cu  $n = 4$  (adică 4 biți/pixel sau 16 nuanțe de gri). Imaginea poate fi separată în patru imagini bi-nivel. Dacă doi pixeli adiacenți din imaginea originală au valorile 0000 și 0001, atunci sunt similari. De asemenea, sunt identici în trei din cele patru imagini bi-nivel. Totuși, doi pixeli adiacenți cu valori 0111 și 1000 sunt de asemenea similari în imaginea în tonuri de gri (valorile lor sunt 7, respectiv 8), dar diferă în toate cele patru imagini alb-negru.

Această problemă apare deoarece reprezentarea binară a numerelor întregi adiacente poate diferi prin mai mulți biți. Codurile binare pentru 0 și 1 diferă printr-un bit, cele pentru 1 și 2 diferă prin doi biți, și cele pentru 7 și 8 prin patru biți. Soluția este proiectarea unor coduri speciale, astfel încât codurile oricăror două numere întregi consecutive  $i$  și  $i+1$  să difere numai printr-un singur bit. Un exemplu de astfel de cod este codul Gray reflectat.

## 5.3. Metode și abordări ale compresiei imaginii

*Abordarea 4.* Se folosește contextului unui pixel pentru a prezice valoarea sa. Contextul unui pixel este dat de valorile câtorva dintre vecinii săi. Se examinează câțiva vecini ai unui pixel  $P$ , se calculează o medie  $A$ , a valorilor lor, și se prezice că  $P$  va avea valoarea  $A$ . Principiul compresiei de imagini spune că predicția va fi corectă în cele mai multe cazuri, aproape corectă în multe cazuri, și complet greșită în puține cazuri. Valoarea prezisă a pixelului  $P$  reprezintă informația redundantă în  $P$ , astfel încât se calculează diferența :  $\Delta = P - A$ , și se asignează coduri instantanee de lungime variabilă pentru diferitele valori  $\Delta$ . Dacă  $P$  poate lua valori de la 0 la  $m - 1$ , atunci  $\Delta$  va avea valori în intervalul  $[-(m - 1), +(m - 1)]$ , și numărul cuvintelor de cod necesare este  $2m - 1$ .

Experimente cu un număr mare de imagini sugerează că valorile lui  $\Delta$  tind să fie distribuite după distribuția Laplace. O metodă de compresie ar putea să folosească această distribuție pentru a asigna o probabilitate fiecărei valori a lui  $\Delta$ , și apoi să se folosească codarea aritmetică pentru a coda eficient valorile  $\Delta$ . Acesta este principiul metodei progresive multinivel MLP.

Contextul unui pixel poate fi constituit din unul sau doi din vecinii săi imediați. Dacă, însă, se consideră mai mulți pixeli vecini în obținerea contextului, se pot obține rezultate mai bune. Media  $A$  într-un astfel de caz ar trebui ponderată cu vecinii apropiați, care au o pondere mai mare. Pentru ca decodorul să poată decoda o imagine, ar trebui să poată calcula contextul fiecărui pixel. Acest lucru înseamnă că în context ar trebui să fie incluși doar pixelii care au fost deja codați. Dacă imaginea este scanată în ordinea rastrului, contextul ar trebui să conțină doar pixeli localizați deasupra pixelului curent sau pe același rând cu el, la stânga.



## 5.3. Metode și abordări ale compresiei imaginii

*Abordarea 5.* Se aplică o transformare valorilor pixelilor, și se codează valorilor transformate. Se reamintește că pentru a realiza compresia, trebuie redusă sau eliminată redundanța. Redundanța unei imagini este cauzată de corelația dintre pixeli, deci transformând pixelii într-o reprezentare în care aceștia sunt decorelați, se elimină redundanța. De asemenea este posibil ca o transformare să fie apreciată în funcție de entropia imaginii. Într-o imagine puternic corelată, pixelii tind a avea valori echiprobabile, ceea ce duce la o entropie maximă. Dacă pixelii transformați sunt decorelați, anumite valori de pixeli devin mai frecvente, având astfel probabilități mari, în timp ce alte valori sunt rare, fapt ce conduce la o entropie mică. Cuantizând valorile transformate, se poate produce o compresie cu pierdere de informație, eficientă, a imaginii. Se dorește ca valorile transformate să fie independente, deoarece codarea valorilor independente face mai simplă construirea unui model statistic.

În cazul imaginilor în culori, un pixel este constituit din trei componente de culoare, roșu, verde și albastru. Majoritatea imaginilor color sunt ori în tonuri continue, ori în tonuri discrete.

## 5.3. Metode și abordări ale compresiei imaginii

*Abordarea 6.* Principiul acestei abordări constă în separarea unei imagini color în tonuri continue în trei imagini în tonuri de gri și compresia fiecăreia din ele separat, folosind abordările 2, 3 și 4.

Pentru o imagine în tonuri continue, principiul compresiei de imagini implică faptul că pixelii adiacenți au culori similare, dacă nu chiar identice. Totuși, culori similare nu înseamnă valori similare ale pixelilor. Se consideră, de exemplu, valori pe 12 biți ale pixelilor, în care fiecare componentă de culoare este exprimată în patru biți. Astfel, cei 12 biți 1000|0100|0000 reprezintă un pixel a cărui culoare este o mixtură de opt unități de roșu (aproape 50%, din valoarea maximă de 15 unități), patru unități de verde (circa 25%), și deloc albastru. Se consideră doi pixeli adiacenți cu valorile 0011|0101|0011 și 0010|0101|0011. Aceștia au culori similare, din moment ce doar componentele lor roșii diferă printr-o unitate. Cu toate acestea, când se consideră ca numere de 12 biți, cele două numere 001101010011 și 001001010011 sunt diferite, pentru că diferă într-un bit cu pondere semnificativă.

O caracteristică importantă a acestei abordări este folosirea unei reprezentări tip luminanță – crominanță, YUV, în loc de reprezentarea comună RGB. Avantajul acestei reprezentări este că ochiul este sensibil la modificări mici ale luminanței, dar nu și la ale crominanței. Aceasta permite pierderea unei cantități considerabile de date în componentele de crominanță, fără o pierdere vizibilă de calitate.

## 5.3. Metode și abordări ale compresiei imaginii

*Abordarea 7.* O abordare diferită este necesară pentru imaginile în tonuri discrete. Se reamintește că o astfel de imagine conține regiuni uniforme care pot apărea de mai multe ori într-o imagine. Un exemplu îl constituie o pagină scrisă la calculator care constă din text și icoane. Fiecare caracter de text și fiecare icoană este o regiune, și fiecare regiune poate apărea de mai multe ori în imagine. O modalitate posibilă de compresie a unei astfel de imagini este scanarea sa, identificarea regiunilor, și găsirea regiunilor care se repetă. Dacă o regiune B este identică cu o regiune A deja găsită, atunci B poate fi compresată prin înregistrarea unui pointer corespunzător lui A în secvența compresată. Metoda descompunerii în blocuri este un exemplu de implementare a acestei abordări.

## 5.4. Transformări folosite în compresia imaginilor

*Abordarea 8.* Se împarte imaginea în regiuni (care se suprapun sau nu) și se compresează prin procesarea părților una câte una. Se presupune că următoarea parte de imagine neprocesată este partea cu numărul  $n$ . Se încearcă regăsirea ei în părțile  $1 \div n - 1$ , care au fost deja procesate. Dacă partea  $n$  poate fi exprimată, de exemplu, ca o combinație a unor părți anterioare scalate și rotite, atunci doar cele câteva numere care specifică combinația trebuie salvate, și partea  $n$  poate fi ignorată. Dacă partea  $n$  nu poate fi exprimată ca o combinație de părți deja procesate, aceasta este procesată și salvată în secvența compresată.

Această abordare este baza diferitelor metode fractale pentru compresia de imagini. Se aplică principiul compresiei de imagine asupra părților de imagine, în loc de pixelii individuali. Aplicat în acest fel, principiul afirmă că imaginile ce urmează a fi compresate au un anumit volum de auto-similaritate, adică părți de imagine sunt identice sau similare cu întreaga imagine sau cu alte părți.

## 5.4. Transformări folosite în compresia imaginilor

Conceptul matematic de transformare este important în multe domenii, printre care și cel al compresiei de imagini. O imagine poate fi compresată prin transformarea pixelilor săi (care sunt corelați) într-o reprezentare unde aceștia sunt decorelați. Compresia este obținută dacă valorile noi sunt mai mici, în medie, decât cele originale. Compresia cu pierdere de informație poate fi obținută prin cuantizarea valorilor transformate. Decodorul primește valorile transformate din secvența compresată și reconstruiește datele originale (exacte sau approximate), prin aplicarea transformării inverse. Transformările discutate în continuare sunt ortogonale.

Termenul de decorelare se referă la faptul că valorile transformate sunt independente unele de altele. Ca urmare, ele pot fi codate independent, ceea ce face mai simplă construirea unui model statistic. O imagine poate fi compresată, dacă reprezentarea sa are redundanță. Redundanța în imagini derivă din corelarea pixelilor. Dacă se transformă imaginea într-o reprezentare în care pixelii sunt decorelați, se elimină redundanța și imaginea a devenit în totalitate compresată.

# ÎNTREBĂRI