COMPRESIA FRACTALĂ A IMAGINILOR

1. Introducere

Compresia fractală a imaginilor ("Fractal Image Compression" - FIC) este o metodă relativ nouă, descrisă pe scurt de următoarele caracteristici:

1. Este o metodă de compresie cu pierderi ("lossy")
2. Este o metodă promiţătoare, superioară (pentru rapoarte de compresie mari) standardului JPEG.
3. Fractalii implicaţi în FIC sunt Sisteme de Funcţii Iterate (IFS).
4. Este o formă de cuantizare vectorială care foloseşte un dicţionar virtual ("virtual codebook").
5. Îmbunătăţirea rezoluţiei este o caracteristică importantă, dar totuşi NU poate ajuta la obţinerea unor rapoarte de compresie exagerate.
6. Comprimarea este lentă iar decomprimarea este rapidă.
7. Metoda este patentată.

Naşterea geometriei fractale este legată de numele matematicianului Benoit B. Mandelbrot, care a publicat în 1977 lucrarea "The Fractal Geometry of Nature", titlu de referinţă în domeniu. Lucrarea a enunţat o teorie importantă: geometria tradiţională, bazată pe linii drepte şi suprafeţe netede, nu se aseamănă geometriei naturii (copacilor, norilor, munţilor), spre deosebire de geometria fractalilor, cu muchiile lor curbe şi detalii "ad infinitum".

Această teorie a deschis numeroase posibilităţi. Cercetătorii din domeniul calculatoarelor au avut la dispoziţie un sistem matematic capabil să genereze peisaje artificiale dar care totuşi arată natural, iar matematicienii - un întreg univers de entităţi geometrice de studiat. Evident, matematicienii au început să caute existenţa unor similitudini în cadrul acestei diversităţi. John Hutchinson a demonstrat în 1981 că aceste similitudini există, şi anume în cadrul Teoriei Funcţiilor Iterate ("Iterated Function Theory"). Ulterior, Michael Barnsley a publicat renumita carte "Fractals Everywhere", în care prezintă teoria "Iterated Function Systems" (IFS) şi unde formulează "Collage Theorem". Aceasta enunţă proprietăţile pe care un IFS trebuie să le aibă pentru a putea fi folosit la reprezentarea unei imagini.

A apărut astfel următoarea problemă: dacă, pe de o parte, teoria fractalilor este utilă pentru generarea unor imagini cu aspect "natural", nu cumva poate fi folosită şi în sens invers, la comprimarea imaginilor ? (având în vedere şi faptul că anumiţi fractali din spatiul complex, foarte spectaculoşi, se generează pe baza unor ecuaţii foarte simple). Pornind de la o imagine dată, obţinerea unui IFS care o poate genera (identic sau cât mai similar) a fost denumită "**the inverse problem**". Această problemă rămânea nerezolvată.

Michael Barnsley afirmă în 1988 că a rezolvat problema folosind "Collage Theorem". El a patentat metoda şi a raportat rezultate spectaculoase (rapoarte de compresie 10.000:1), dar numai pe imagini care erau construite manual. Deoarece algoritmul lui Barnsley este foarte lent şi necesită asistenţă din partea unui operator uman, patentul lui Barnsley este cunoscut sub denumirea ironică de *"Graduate Student Algorithm"*:

- acquire a graduate student
- give the student a picture
- and a room with a workstation
- lock the door
- wait until the student has reverse engineered the picture
- open the door

Rezultatele obţinute de unul dintre doctoranzii lui Barnsley au permis depăşirea limitărilor drastice ale "graduate student algorithm". Arnaud Jacquin renunţă la determinarea IFS pentru întreaga imagine, propunând o variantă modificată a IFS numită "Partitioned Iterated Function Systems" (PIFS), de asemenea patentată, şi realizează o implementare soft a algoritmului. Algoritmul nu era nici sofisticat, nici rapid, însă era complet automat. Preţul plătit este renunţarea la rapoartele de compresie spectaculoase anunţate de Barnsley: pentru imagini color cu 24 bpp, rata de compresie obţinută varia între 8:1 şi 50:1 în condiţiile unei calităţi acceptabile.

Fractalii din cadrul FIC nu fac parte dintre cei din planul complex (cum sunt setul Mandelbrot sau seturile Julia), ci din categoria IFS. Matematicianul Heinz-Otto Peitgen compară IFS cu **o "Multiple Reduction Copying Machine**" (MRCM), care reprezintă un copiator cu următoarele proprietăţi:

1. Există mai multe sisteme de lentile, care creează copii multiple (posibil) parţial suprapuse.
2. Fiecare sistem de lentile reduce mărimea originalului.
3. Copiatorul operează în buclă, rezultatul unui pas fiind intrare pentru următorul. Ca imagine iniţială se poate alege orice.

Prima caracteristică defineşte IFS ca fiind un sistem. Datorită celei de a doua, funcţiile unui IFS sunt contractive. Cea de-a treia face IFS un sistem iterativ.

Aşadar, un IFS este un set de transformări contractive care mapează o anumită zonă rectangulară în regiuni mai mici ale ei. Pentru aceasta, aproape întotdeauna se folosesc transformări afine, care translatează, scalează şi rotesc punctele planului.

Caracteristica importantă a IFS este că evaluarea iterativă a setului de transformări (funcţionarea MRCM) conduce spre o imagine unică, numită atractorul IFS-ului. Conform "Collage Theorem", atractorul este complet independent de imaginea iniţială.

Pentru o imagine care trebuie codificată, găsirea IFS-ului optim este cunoscută sub numele de "inverse problem". După cum s-a arătat anterior, această problemă nu a fost rezolvată. Pentru a putea face faţă diversităţii imaginilor reale (naturale) se folosesc PIFS. În PIFS, transformările nu mapează întreaga imagine în porţiuni, ci mapează porţiuni mai mari din imagine în porţiuni mai mici, datorită variaţiilor calitative ale imaginii (ex. alternanţa nori – cer - nori). PIFS stabileşte o legătură între porţiuni din imaginea iniţială care au un aspect similar. Conform notaţiei introduse de Jacquin, regiunile mari sunt numite "domain" (**D**) iar cele mici "range" (**R**) - transformările operând **de la** un **"**domain" **la** un "range". Vom folosi în cele ce urmează notaţiile de **domeniu** pentru “domain” şi cea de **codomeniu** pentru “range”. Am preferat această soluţie pentru avantajul de a indica faptul că transformata fractală din cadrul PIFS se aplică unui domeniu şi rezultă un codomeniu. În mod riguros ar trebui folosit de fiecare dată termenul de *bloc domeniu* respectiv *bloc codomeniu* dar, din motive de simplificare a expunerii nu vom repeta uneori şi *bloc*, presupunând că acest lucru este subînţeles. E necesar ca fiecare pixel din imaginea originală să aparţină cel puţin unui codomeniu. Dispunerea codomeniilor în cadrul imaginii constituie partiţionarea imaginii.

Deoarece acest sistem de mapări este contractiv, prin iteraţii el va converge rapid spre atractorul său. Construirea PIFS constă tocmai în găsirea perechilor codomeniu-domeniu şi a transformărilor afine ce realizează mapările optime.

Prin urmare, o codificare fractală a unei imagini constă în:

1. Partiţionarea imaginii (forma şi poziţia codomeniilor)
2. Descrierea transformărilor afine (câte una pentru fiecare codomeniu)

Esenţa procesului de compresie este găsirea perechilor codomeniu-domeniu pentru care ***eroarea de colaj*** ("collage error") - de reprezentare a unui codomeniu printr-o versiune transformată a domeniului - este minimă. Procesul de căutare este din acest motiv foarte lent. Deşi teoria nu impune nici o restricţie asupra

formei blocurilor, în practică se consideră doar forme pătrate, dreptunghiulare sau triunghiulare doar pentru a face problema abordabilă.

În general găsirea unui PIFS bun pentru o imagine dată cuprinde următoarele etape:

1. Partiționarea imaginii în codomenii.
2. Alegerea setului de domenii.
3. Alegerea tipului de transformări ce vor fi folosite.
4. Selectarea unei metrici (distanțe) pentru blocuri (o măsură a erorii).
5. Specificarea unei metode de determinare optimă a perechilor codomeniu-domeniu.

Optimizarea fiecăreia dintre aceste etape este importantă şi prezintă în continuare interes deosebit în cadrul FIC. Cu toate acestea, se poate afirma că principala problemă a FIC rămâne **creşterea vitezei de codificare**.

O imagine obținută prin achiziție prezintă o rezoluție determinată de performanțele dispozitivului. Mărirea imaginii prin soft ("zoom") nu oferă detalii suplimentare peste un anumit nivel, ci doar replică valoarea pixelilor. Cu o imagine fractală lucrurile se comportă diferit: deoarece transformările afine sunt contractive d.p.d.v. spațial, la fiecare iterație sunt create detalii suplimentare (are loc o creştere a rezoluției). Sunt create astfel detalii cu autoasemănare la orice nivel de rezoluție, până la nivel infinitezimal. Acest fapt conferă imaginilor fractale **independență de rezoluție**. Din această cauză, la orice "zoom" într-o imagine fractală, aceasta va arăta cum "este de aşteptat", fără efectul de bloc produs de replicarea pixelilor. Trebuie atrasă însă atenția că, în cazul imaginilor fractale, detaliile nu au fost reținute ci au fost **generate** (evaluarea raportului de compresie pe baza unei imagini mărite constituie o posibilă confuzie). În fapt, FIC reprezintă o *modalitate avansată de interpolare*.

Comparația FIC cu cuantizarea vectorială (CV) indică:

- **CV**: blocurile codomeniu şi domeniu au aceeaşi mărime; **IFS**: blocurile domeniu sunt întotdeauna mai mari.
- **CV**: blocurile domeniu sunt copiate fără modificări; **IFS**: fiecare bloc domeniu suferă o transformare afină.
- **CV**: dicționarul este stocat separat de imagine; **IFS**: dicționarul nu este dat în mod explicit, ci cuprinde porțiuni din atractor pe măsură ce acesta este generat în timpul iterațiilor (motiv pentru care e numit dictionar virtual ("virtual codebook"). El nu există separat de transformările afine ce definesc un IFS.
- **CV**: dicționarul este comun pentru mai multe imagini; **IFS**: "dicționarul virtual" este specific pentru fiecare imagine.

Există o variantă a CV numită "Mean-Removed Gain-Shape Vector Quantization", care permite şi scalarea şi deplasarea luminanței, variantă care apropie şi mai mult cele două domenii.

În procesarea de imagini se depun mari eforturi pentru a evalua (cuantifica) cât mai precis eroarea prezentă într-o imagine comprimată şi apoi pentru a minimiza această măsură (discutabilă) a erorii. Cele mai des folosite criterii de evaluare a erorii sunt raportul semnal-zgomot SNR sau PSNR (în special), eroarea medie pătratică şi eroarea medie absolută.

Testele efectuate pentru a compara (d.p.d.v. al RSZ) JPEG şi FIC arată că JPEG este mai bun pentru rate mai mici de compresie, iar FIC este mai bună pentru rate mai mari de compresie, limita fiind de obicei în jurul ratei de 40:1. Susținătorii JPEG afirmă că această valoare favorizează JPEG, întrucât peste această limită imaginile sunt sever afectate de distorsiuni, ceea ce le face de cele mai multe ori inutilizabile practic. Adepții FIC, pe de altă parte, susțin că SNR nu e un parametru de eroare potrivit şi că distorsiunile au un aspect mult mai "natural" decât aspectul "rectangular" al JPEG, atât la rate de bit mici cât şi mari (observație în principiu corectă dar neacceptată unanim). Ceea ce lipseşte este un parametru de eroare uşor de calculat şi care să reflecte fidel impresia subiectivă asupra subiecților umani. Până la găsirea acestui parametru, ochiul uman este cel mai bun arbitru.

2. Puțină matematică

Cât de mare este un fractal? Când doi fractali sunt similari unul altuia într-un oarecare sens ? Există mai multe numere, asociate cu fractali, care pot fi folosite pentru a-i compara. Ele sunt de obicei denumite dimensiuni fractale şi reprezintă încercări de a cuantifica impresia subiectivă pe care o avem în legatură cu cât de dens ocupă fractalul spațiul său metric. Dimensiunea fractală reprezintă o măsură obiectivă pentru a compara fractalii. Prezentăm în continuare câteva fundamente matematice în ceea ce priveşte dimensiunea fractală, conform cărții de referință a lui Barnsley "Fractals everywhere".

Fie (X,d) un spațiu metric complet şi fie $A \subset H(X)$ un subset compact nevid al lui X. Fie $\varepsilon > 0$. Notăm $B(x,\varepsilon)$ sfera închisă de rază ε şi având centrul în punctul $x \in X$. Definim $N(A,\varepsilon)$ ca fiind numărul minim de sfere închise de rază ε necesare să acopere A.

$N(A,\varepsilon)$ = cel mai mic întreg pozitiv M astfel că $A \subset \bigcup_{n=1}^{M} B(x_n, \varepsilon)$

pentru un set de puncte $\{x_n : n = 1,2,...,M\} \subset X$.

Ideea intuitivă aflată în spatele noțiunii de dimensiune fractală este că un set A are dimensiunea fractală D dacă

$$N(A, \varepsilon) \approx C \left(\frac{1}{\varepsilon}\right)^D \text{ pentru o valoare pozitivă C}$$

Riguros, avem următoarea definiție:

Definiție *Fie $A \in H(X)$ unde (X,d) este un spațiu metric. Pentru fiecare $\varepsilon > 0$ fie $N(A,\varepsilon)$ cel mai mic număr de sfere închise de rază ε necesare a acoperi A. Dacă*

$$D = \lim_{\varepsilon \to 0} \left\{ \frac{\ln(N(A,\varepsilon))}{\ln(1/\varepsilon)} \right\}$$

există, atunci D este numită ***dimensiunea fractală a lui A****.*

Vom folosim în continuare notația D = D(A) şi spunem "A are dimensiunea fractală D". Se poate arăta uşor că dimensiunea fractală a unui punct este 0 şi a unui segment este 1.

Procesul de determinare a dimensiunii fractale este simplificat de următoarea teoremă în care se înlocuieşte variabila reală ε cu o variabilă discretă ε_n.

"The Box Counting Theorem" *Fie $A \in H(R^m)$ şi metrica euclidiană. Se acoperă R^m cu cutii închise pătrate de latură $(1/p^n)$ ca în Figura 1. Fie $N_n(A)$ numărul de asemenea cutii care intersectează atractorul. Dacă*

$$D = \lim_{n \to \infty} \left\{ \frac{\ln(N_n(A))}{\ln(p^n)} \right\}$$

există atunci A are dimensiunea fractală D.

În Figura 1 s-a reprezentat cazul spațiului bidimensional (m=2) pentru p=3 şi n=2. Folosind această teoremă rezultă uşor că dimensiunea fractală a unei regiuni pătratice este 2.

Un exemplu tipic de fractal este "mulțimea lui Cantor" (C): pornind de la intervalul [0,1] se elimină "treimile din mijloc". Aplicând teorema anterioară pentru p=3, m=1 şi având $N_n(C) = 2^n$.

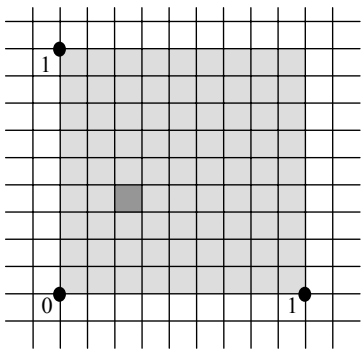
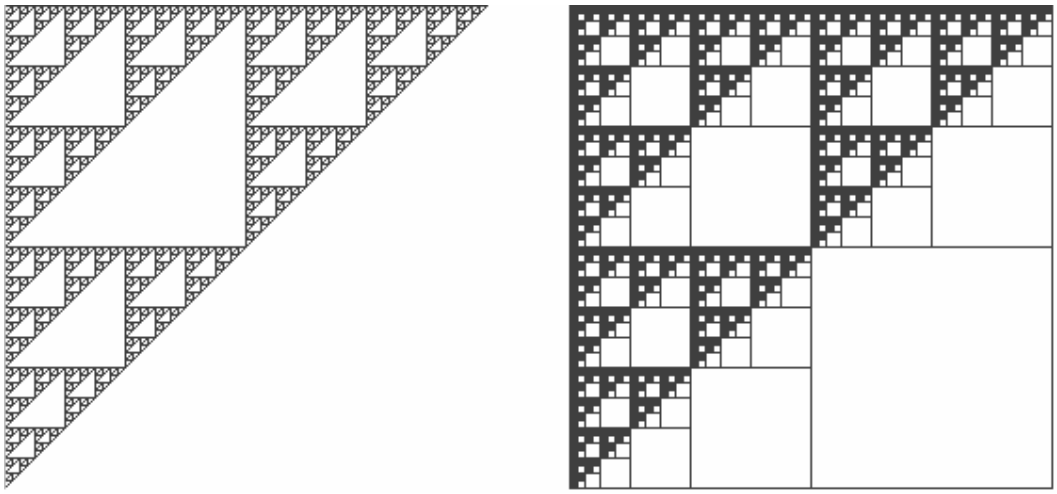


Figura 1 Acoperire cu cutii închise



a b

Figura 2 Triunghiul lui Sierpinski

$$D = \lim_{n\to\infty}\left\{\frac{\ln(\ N_n(C)\ )}{\ln(p^n)}\right\} = \lim_{n\to\infty}\left\{\frac{\ln(\ 2^n\ )}{\ln(\ 3^n\ )}\right\} = \frac{\ln\ 2}{\ln\ 3}$$

Un alt exemplu pe care se poate aplica uşor teorema anterioară este "triunghiul lui Sierpinski" (S) obţinut prin înlocuirea unui triunghi cu trei replici ale sale reduse la jumătate, reprezentarea sa fiind dată în Figura 2a. În Figura 2b este reprezentată împărţirea în cutii închise de latură $(1/2)^n$ corespunzând cazului m=2 şi p=2 din Teoremă în momentul n = 6. Rezultă că avem nevoie de un număr de 3^n asemenea cutii rezultând

$$D = \lim_{n\to\infty}\left\{\frac{\ln(\ N_n(S)\ )}{\ln(p^n)}\right\} = \lim_{n\to\infty}\left\{\frac{\ln(\ 3^n\ )}{\ln(\ 2^n\ )}\right\} = \frac{\ln\ 3}{\ln\ 2}$$

Dimensiune fractală pentru atractorii IFS

O teoremă spectaculoasă care permite determinarea rapidă a dimensiunii unui fractal este următoarea, care se poate aplica fractalilor obţinuţi ca şi atractori ai IFS.

Teoremă *Fie $\{R^m; w_1, w_2, \ldots, w_N\}$ un IFS hiperbolic şi fie A atractorul său. Fie w_n o asemănare având factorul de scalare s_n pentru fiecare $n \in \{1,2,3,\ldots,N\}$. Dacă IFS este complet nesuprapus sau doar alăturat atunci atractorul are dimensiunea fractală D(A), care este dată de soluţia unică a ecuaţiei*

$$\sum_{n=1}^{N} |s_n|^{D(A)} = 1, \quad \mathrm{D(A)} \in [0,\mathrm{m}]$$

Dacă IFS este suprapus, atunci $D(A) \le \overline{D}$ unde $\overline{D}$ este soluţia ecuaţiei

$$\sum_{n=1}^{N} |s_n|^{\overline{D}} = 1, \quad \overline{D} \in [0,\infty)$$

Această teoremă ne permite determinarea rapidă a dimensiunii fractale ca în exemplele următoare:

- privind "mulţimea lui Cantor" (C) ca şi atractorul unui IFS hiperbolic

$$\left\{[0,1];\ \ w_1(x) = \frac{1}{3}x;\ \ w_2(x) = \frac{1}{3}x + \frac{2}{3}\right\}$$

obţinem având $s_1 = 1/3$ şi $s_2 = 1/3$

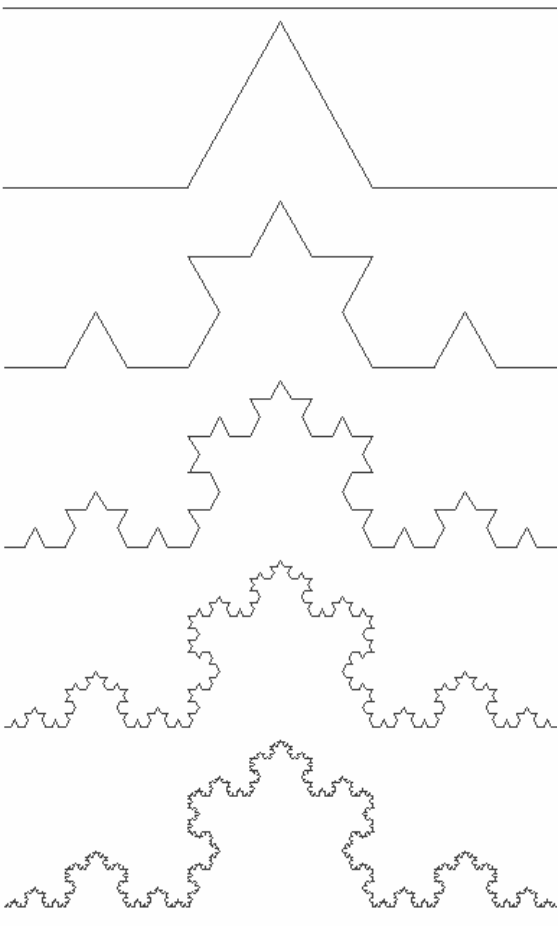


Figura 3 Curba lui Koch

$$\left(\frac{1}{3}\right)^{D(C)}+\left(\frac{1}{3}\right)^{D(C)}=1$$

$$\Rightarrow 2=3^{D(C)}\Rightarrow D(C)=\frac{\ln 2}{\ln 3}$$

regăsind evident valoarea determinată anterior.

- privind "triunghiul lui Sierpinski" (S) ca şi atractorul unui IFS hiperbolic avem trei transformări cu factorii de scalare $s_1=s_2=s_3=1/2$.

$$\left(\frac{1}{2}\right)^{D(S)}+\left(\frac{1}{2}\right)^{D(S)}+\left(\frac{1}{2}\right)^{D(S)}=1$$

$$\Rightarrow 3=2^{D(S)}\Rightarrow D(S)=\frac{\ln 3}{\ln 2}$$

regăsind evident valoarea determinată anterior

- privind clasica "curbă Koch" (K) ca şi atractorul unui IFS hiperbolic avem patru transformări cu factorii de scalare $s_1=s_2=s_3=s_4=1/3$.

$$\left(\frac{1}{3}\right)^{D(K)}+\left(\frac{1}{3}\right)^{D(K)}+\left(\frac{1}{3}\right)^{D(K)}+\left(\frac{1}{3}\right)^{D(K)}=1$$

$$\Rightarrow 4=3^{D(K)}\Rightarrow D(K)=\frac{\ln 4}{\ln 3}$$

Se constată că dimensiunea fractală a seturilor considerate ca şi exemple este fracţionară (nu este un număr întreg). Aceasta valoare **fract**ionară este cea care a stat la baza introducerii termenului de **fract**al pentru aceste seturi (figuri).

3. Comparație cu alte metode

Compresia de imagini bazată pe DCT şi standardizată în JPEG este o tehnică de succes pentru rapoarte de compresie de până la aproximativ 40:1. Peste această limită, **aspectul de bloc** devine pregnant iar calitatea imaginii scade devenind inutilizabilă în practică. Eliminarea componentelor de frecvență ridicată determină distorsiuni mari în cadrul imaginii, efectele fiind vizibile mai ales pe muchiile obiectelor din imagine, aspect cunoscut sub numele de fenomenul Gibb.

Un alt dezavantaj al compresiei JPEG constă în **dependența de rezoluție** a imaginii comprimate. În cazul în care este necesară afişarea imaginii la o rezoluție superioară celei originale este necesară duplicarea pixelilor, efectul de bloc devenind supărător.

Diferite puncte de vedere asupra compresiei fractale a imaginilor

Pe măsura creşterii interesului pentru compresia fractală a imaginilor, cercetătorii au apropiat acest domeniu de diferite alte domenii. Cele mai importante puncte de vedere propuse pentru abordarea compresiei fractale a imaginilor au la bază asemănarea cu:

SISTEMELE CU FUNCȚII ITERATE (IFS) - Asemenea sisteme sunt de fapt operatori în spațiul metric şi au fost introduse în matematică de o lucrare a lui Hutchinson în 1981 care a aratat că au, ca şi atractori, subseturi fractale. Barnsley a intuit pentru prima dată folosirea IFS în compresia imaginilor prin descrierea imaginilor ca atractori. Arnaud Jacquin propune în 1989 o metodă care renunță la privirea întregii imagini ca şi un fractal şi pune bazele **PIFS** ("Partitioned IFS"), moment care a constituit de fapt începutul unei noi direcții de cercetare foarte promițătoare în domeniul compresiei de imagini.

CUANTIZAREA VECTORIALĂ - Compresia fractală a imaginilor este foarte apropiată de un caz particular de cuantizare vectorială, numit "mean-removed shape-gain vector quantization" MRSG-VQ. În MRSG-VQ un bloc este aproximat prin suma unei componente continue (constante) şi o versiune scalată a unui reprezentant din dicționar. Specificul compresiei fractale constă în aceea că dicționarul nu este prezent explicit la decodare, ci doar implicit prin transformarea fractală. Din acest motiv dicționarul mai este numit şi "*dicționar virtual*" ("virtual codebook").

Nici unul din aceste puncte de vedere nu este general (complet), toate împreună contribuind la înțelegerea mai profundă a acestei tehnici şi la dezvoltarea acesteia prin includerea de idei existente deja şi care şi-au dovedit cu prisosință utilitatea în acele domenii.

Alegerea modelului codor-decodor al FIC se face pe baza punctului de vedere al **Sistemelor de Funcții Iterate** (IFS), în timp ce pentru **determinarea parametrilor** codorului se adoptă în principal punctul de vedere al **cuantizării vectoriale**.

4. Problema Inversă a Sistemelor de Funcții Iterate

Fie (M,d) un spațiu metric al imaginilor digitale, unde d este o metrică - o măsură a distorsiunii, şi fie μ_{orig} o imagine care se doreşte a se coda. *Problema inversă a teoriei funcțiilor iterate* este construcția unei transformări de contracție a imaginii, definită din spațiul (M,d) în el însuşi, pentru care μ_{orig} să fie punct fix. Notăm $\mathbf{F}$ setul transformărilor permise: un subset specific, definit a priori, al spațiului tuturor contracțiilor în (M,d).

Cerințele asupra transformării τ sunt următoarele:

$$\exists\ s < 1 \text{ astfel încât } \forall\ \mu,\nu \in M \qquad d(\tau(\mu),\tau(\nu)) \le s\,d(\mu,\nu) \qquad (1)$$

şi $\quad d(\mu_{orig},\tau(\mu_{orig}))$ să fie cât mai apropiată de zero $\qquad (2)$

Scalarul s este numit contractivitatea transformării τ. În aceste condiții, şi considerând că τ are o

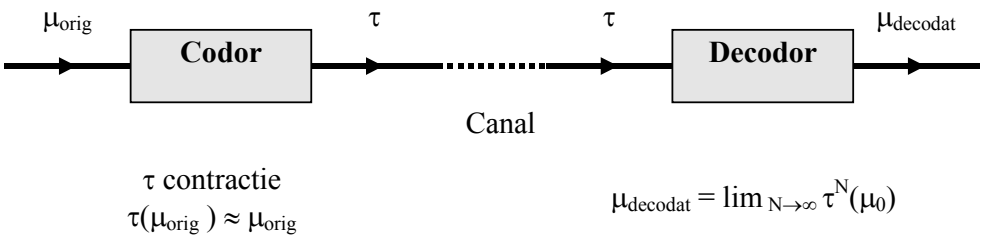


Figura 4 Schema generală de codare-decodare FIC

complexitate mai mică decât imaginea originală, τ poate fi privită ca şi o codificare (cu pierderi) a imaginii μ_{orig} .

Prin aplicarea repetată a inegalității triunghiului în (M,d) şi folosind contractivitatea lui τ se poate arăta că pentru orice imagine μ_0 şi orice întreg pozitiv n avem:

$$d(\mu_{orig}, \tau^n(\mu_0)) \le \frac{1}{1-s} d(\mu_{orig}, \tau(\mu_{orig})) + s^n d(\mu_{orig}, \mu_0) \qquad (3)$$

Din precedentele două relații şi considerând s < 1, observăm că, după un număr inițial de iterații, termenii oricărei secvențe iterate de forma

$$\{ \mu_n = \tau^n(\mu_0) \}_{n \ge 0} \qquad (4)$$

unde μ_0 este o imagine inițială oarecare, se grupează în jurul imaginii originale (rezultat cunoscut în literatură ca şi **"Collage Theorem"**). În spațiul imaginilor discrete cuantizate, secvența converge spre o imagine stabilă care, datorită modului de construcție, se spune că este fractală.

În raport cu cele prezentate anterior, considerăm utile următoarele observații:
1. Apropierea lui $\tau^n(\mu_0)$ de μ_{orig} este condiționată de distanța $d(\mu_{orig}, \tau(\mu_{orig}))$.
2. Convergența secvenței μ_n depinde în mod esențial de restricția ca s să fie mai mic decât 1.
3. Transformarea constantă $\tau = \mu_{orig}$ care are contractivitatea 0 şi satisface în mod evident (1) şi (2) nu este în $\mathbf{F}$ deoarece suntem interesați doar de transformări care pot fi descrise pe un număr de biți mai mic decât imaginea inițială - o cerință esențială pentru compresie.

Numim o asemenea transformare τ din $\mathbf{F}$ care satisface (1) şi (2) un ***cod fractal*** pentru μ_{orig} şi spunem că μ_{orig} este aproximativ auto-transformabilă în raport cu τ. Această terminologie are la bază faptul că imaginile obținute prin această procedură sunt rezultatul aplicării iterative a unei transformări asupra unei imagini inițiale, procedură care este tipică generării de seturi fractale deterministe.

În Figura 4 se prezintă schema generală a unui sistem de codare-decodare fractală.

Observația că redundanța din cadrul imaginilor poate fi modelată prin autoasemănarea la nivel de blocuri ne îndreaptă atenția spre o clasă de transformări având forma generală:

$$\forall \mu \in M,\ \tau(\mu) = \sum_{0 \le i < N} \tau(\mu)\rceil_{R_i} = \sum_{0 \le i < N} \tau_i(\mu \rceil_{D_i}) \qquad (5)$$

unde $\mathbf{P} = \{R_i\}_{0 \le i < N}$ reprezintă o partiționare a imaginii în N blocuri nesuprapuse numite "range". Prin $\mu\rceil_{R_i}$ am notat restricția lui μ la "range"-ul R_i. Obținem astfel

$$\mu = \sum_{0 \le i < N} \mu \rceil_{R_i} \qquad (6)$$

relație care indică faptul că imaginea este reuniunea restricțiilor sale la celulele partiției. În relația (5) τ_i reprezintă o transformare elementară **de la** un domeniu ("domain") D_i **la** un codomeniu ("range") R_i.

Se consideră în cadrul FIC pentru τ_i doar clasa **transformărilor afine** având expresia cea mai generală

$$\begin{bmatrix} x_t \\ y_t \\ i_t \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ i \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \qquad (7)$$

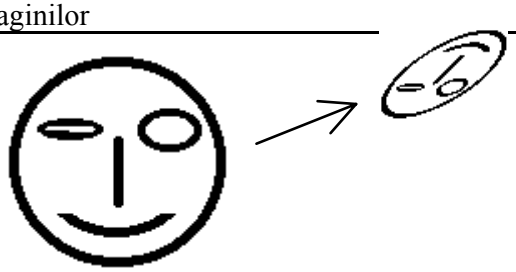


Figura 5 Exemplu de transformare afină

de mapare a unui pixel având coordonatele x,y şi intensitate i în pixelul având coordonatele x_t,y_t şi intensitate i_t printr-o transformare caracterizată de parametrii a_i, b_i, c_i, d_i, e_i, f_i, s_i şi o_i. Caracterul de bloc al unei transformări este dat de faptul că transformarea tuturor pixelilor unui anumit domeniu foloseşte un acelaşi set de coeficienţi a_i, b_i, c_i, d_i, e_i, f_i, s_i şi o_i. Semnificaţia coeficienţilor este oarecum diferită: a_i, b_i, c_i, d_i, e_i, şi f_i sunt responsabili de noua poziţie a pixelului (modificare geometrică) iar s_i şi o_i de noua sa intensitate (modificare masică). De cele mai multe ori coeficienţii transformării geometrice nu se dau explicit sub forma coeficientilor a_i, b_i, c_i, d_i, e_i, f_i, ci implicit prin descrierea poziţiei şi orientării domeniului în raport cu codomeniul.

Deci, pentru claritate, τ_i se poate scrie ca o compunere a două transformări S_i şi T_i

$$\tau_i = T_i \circ S_i \qquad (8)$$

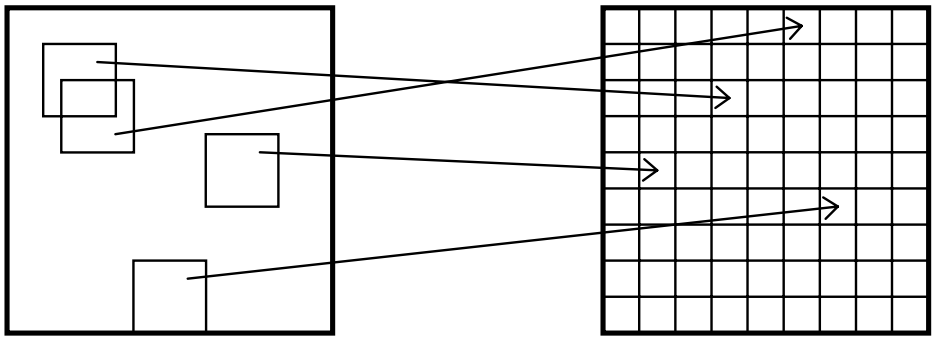
în care S_i este numită partea geometrică iar T_i partea masică a transformării. O reprezentare a transformărilor este dată în Figura 6.

$$\mu\rceil_{Di} \xrightarrow{S_i} S_i(\mu\rceil_{Di}) \xrightarrow{T_i} (T_i \circ S_i)(\mu\rceil_{Di}) = \tau(\mu)\rceil_{Ri} \xrightarrow{\Sigma} \tau(\mu)$$

Figura 6 Transformările realizate în cadrul FIC

Determinarea codului fractal τ corespunzător unei imagini μ_{orig} va fi făcută **prin determinarea transformărilor elementare τ_i** independente una de alta. Codarea imaginii μ_{orig} presupune găsirea, pentru fiecare indice al partiţionării i, a transformării τ_i de la un domeniu D_i la un codomeniu R_i astfel încât transformatul domeniului $\tau_i(\mu_{orig}\rceil_{Di})$ să fie cât mai apropiat de codomeniul iniţial $\mu_{orig}\rceil_{Ri}$ în sensul metricii considerate.

Partea geometrică a transformării S_i realizează maparea imaginii din domeniul D_i în codomeniul R_i. În cazul simplu, întâlnit aproape exclusiv în practică, al mapării D = 2×B (înjumătăţire pe ambele axe) avem



blocuri **domeniu**
undeva în cadrul imaginii

blocuri **codomeniu**
acoperind întrega imagine

Figura 7 Principiul mapării domeniu → codomeniu

(pe lângă stabilirea poziției domeniului în cadrul imaginii)

$$(S\mu_{\lceil Ri})_{x,y} = S_i(\mu_{\lceil Di}) = [\ (\mu_{\lceil Di})_{2x,2y}+(\mu_{\lceil Di})_{2x+1,2y}+(\mu_{\lceil Di})_{2x,2y+1}+(\mu_{\lceil Di})_{2x+1,2y+1}\ ] / 4. \qquad (9)$$

corespunzătoare medierii valorii celor 4 pixeli. Un exemplu de realizare a mapării domeniu→ codomeniu este ilustrat în Figura 7.

Partea masică a transformării T_i este, în cazul cel mai întâlnit, dată de o transformare liniară aplicată unui bloc B×B

$$(T\mu)_{y,x} = s\ \mu_{y,x} + o \qquad (10)$$

în care s este un **factor de scalare** şi o este un **offset**.

O soluție foarte des întâlnită este luarea în considerare (în cadrul transformării geometrice) şi a celor 8 transformări izometrice ale unui bloc de dimensiune B×B. Acestea sunt:

1. identitatea
   $(i_1\mu)_{y,x} = \mu_{y,x}$
2. reflexia față de axa verticală (x=B/2)
   $(i_2\mu)_{y,x} = \mu_{y,B-1-x}$
3. reflexia față de axa orizontală (y=B/2)
   $(i_3\mu)_{y,x} = \mu_{B-1-y,x}$
4. reflexia față de prima diagonală (x=y)
   $(i_4\mu)_{y,x} = \mu_{x,y}$
5. reflexia față de a doua diagonală (x+y=B-1)
   $(i_5\mu)_{y,x} = \mu_{B-1-x,B-1-y}$
6. rotirea în jurul centrului cu 90°
   $(i_6\mu)_{y,x} = \mu_{y,B-1-x}$
7. rotirea în jurul centrului cu 180°
   $(i_7\mu)_{y,x} = \mu_{B-1-y,B-1-x}$
8. rotirea în jurul centrului cu 270°
   $(i_8\mu)_{y,x} = \mu_{B-1-y,x}$

O reprezentare grafică a acestor izometrii este dată în Figura 8 iar în Figura 9 se prezintă exemplul de aplicare a lor pentru (întreaga) imagine Lena (cu toate că în practică izometriile se aplică asupra unui bloc şi nu asupra întregii imagini). Creşterea dimensiunii "domain pool" prin considerarea celor 8 izometrii în scopul îmbunătățirii calității imaginii este o opțiune unanim acceptată.

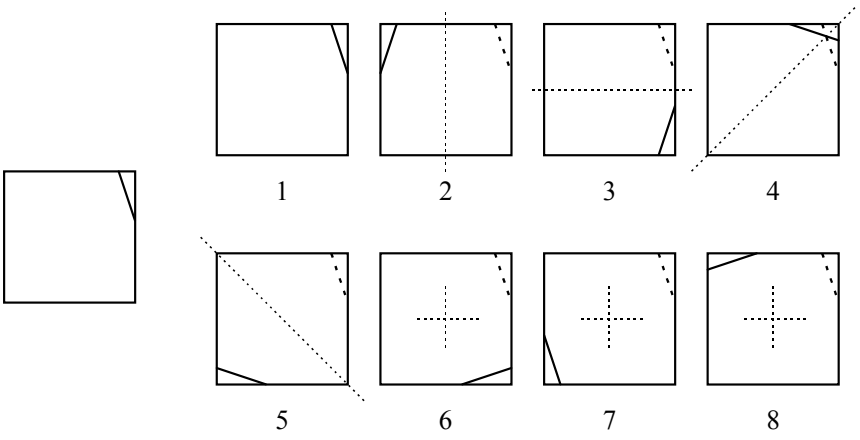


Figura 8 Cele 8 izometrii

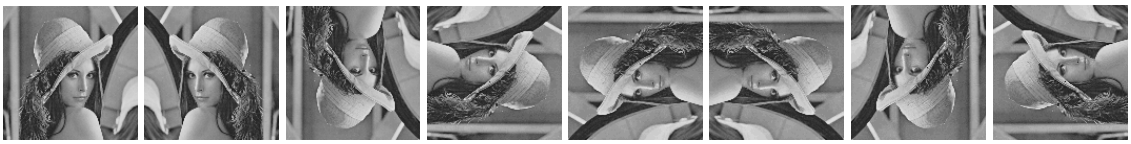


Figura 9 Exemplu de aplicare a izometriilor

5. Creşterea rezoluției folosind Sisteme de Funcții Iterate

O facilitate interesantă a FIC este independența de rezoluție a imaginii comprimate, ceea ce permite creşterea rezoluției folosind IFS. Această caracteristică provine din aceea că, în urma codificării, rezultă parametrii transformărilor din cadrul IFS, fără o legătură directă cu dimensiunea blocurilor asupra cărora se aplică transformările. În acest fel, aplicând aceleaşi transformări unei imagini mărite se obține o versiune de asemenea mărită a imaginii codificate. Detaliile "noi" cu siguranță nu sunt unele "originale" ci unele construite dar, datorită autoasemănărilor existente în imaginea originală, detaliile "create" respectă "forma" imaginii, ele părând mult mai naturale decât cele care ar apare prin duplicarea pixelilor. În Figura 10 prezentăm o schemă a acestui proces. În mod evident, dacă scopul este mărirea rezoluției şi nu compresia, este indicat a se renunța la cuantizarea coeficienților transformatei.

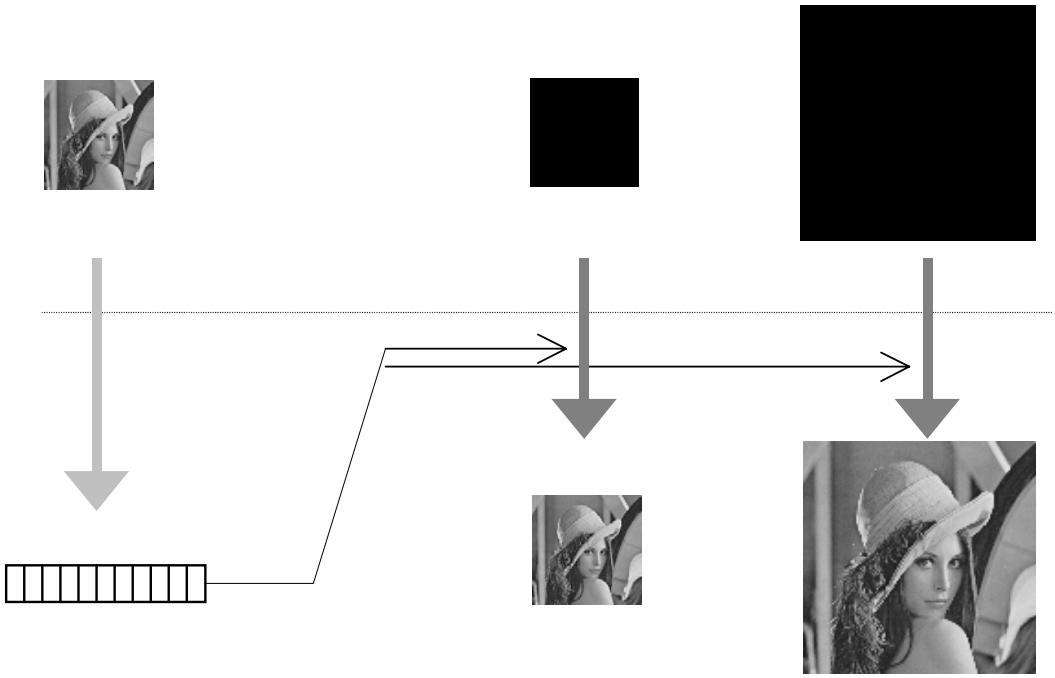


Figura 10 Mărirea rezoluției folosind IFS

6. Determinarea parametrilor transformării

În continuare ne vom concentra atenția doar asupra determinării optime a parametrilor transformării masice (de luminanță), considerând transformarea geometrică deja aplicată.

Considerând două blocuri de imagine D şi R în ca Figura 11, având intensitățile pixelilor $d_1,d_2,...,d_n$ şi corespunzător $r_1,r_2,...,r_n$, expresia de minimizat (eroarea medie pătratică) devine :

$$E(D,R,s,o)^2 = \sum_{i=1}^{n}\left[r_i - \left(s \cdot d_i + o\right)\right]^2 \qquad (11)$$

Deoarece $E(D,R,s,o)^2$ este o funcție de gradul 2 în raport cu s şi cu o, având coeficientul termenului de grad 2 pozitiv, deci prezentând un minim, pentru a obține minimul anulăm derivatele în raport cu aceşti parametri s şi o

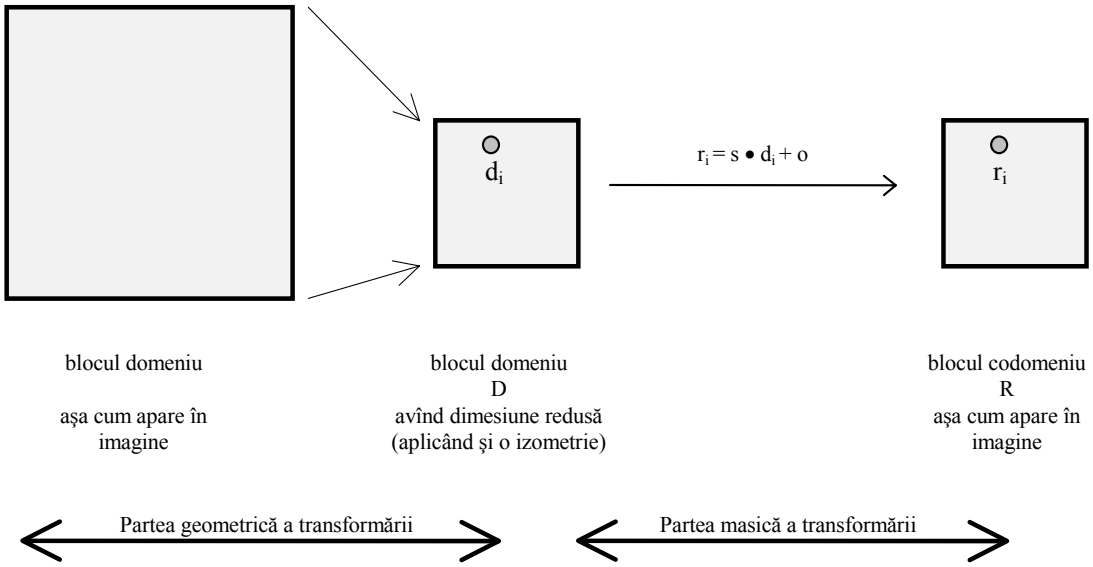


Figura 11 Determinarea parametrilor *s* şi *o* ai transformării

$$\begin{cases} \dfrac{\partial E(D,R,s,o)^2}{\partial s} = 0 \\ \dfrac{\partial E(D,R,s,o)^2}{\partial o} = 0 \end{cases} \tag{12}$$

Concret, folosind pentru $E(D,R,s,o)^2$ expresia (11) obţinem

$$\begin{cases} \sum_{i=1}^{n} 2\left[r_i - (s \cdot d_i + o)\right](-d_i) = 0 \\ \sum_{i=1}^{n} 2\left[r_i - (s \cdot d_i + o)\right] = 0 \end{cases} \tag{13}$$

Izolând s şi o obţinem sistemul

$$\begin{cases} s \cdot \sum_{i=1}^{n} {d_i}^2 + o \cdot \sum_{i=1}^{n} d_i = \cdot \sum_{i=1}^{n} r_i d_i \\ s \cdot \sum_{i=1}^{n} d_i + o \cdot n = \cdot \sum_{i=1}^{n} r_i \end{cases} \tag{14}$$

Rezultă apoi uşor valoarea lui s

$$s = \frac{n \cdot \left(\sum_{i=1}^{n} r_i d_i\right) - \left(\sum_{i=1}^{n} r_i\right) \cdot \left(\sum_{i=1}^{n} d_i\right)}{n \cdot \left(\sum_{i=1}^{n} {d_i}^2\right) - \left(\sum_{i=1}^{n} d_i\right)^2} \tag{15}$$

şi respectiv o

$$o = \frac{1}{n}\left[\left(\sum_{i=1}^{n} r_i\right) - s \cdot \left(\sum_{i=1}^{n} d_i\right)\right] \tag{16}$$

Bineînţeles, dacă numitorul expresiei (15) este zero, considerăm

$$\text{s = 0 şi } o = \frac{1}{n}\left(\sum_{i=1}^{n} r_i\right) \tag{17}$$

Cu s şi o date mai sus, expresia erorii $E(D,R,s,o)^2$ devine:

$$E(D,R)^2 = \frac{1}{n}\left\{\left(\sum_{i=1}^{n} {r_i}^2\right) + s \cdot \left[s \cdot \left(\sum_{i=1}^{n} {d_i}^2\right) - 2 \cdot \left(\sum_{i=1}^{n} r_i d_i\right) + o \cdot 2 \cdot \left(\sum_{i=1}^{n} d_i\right)\right] + o \cdot \left[o \cdot n - 2 \cdot \left(\sum_{i=1}^{n} r_i\right)\right]\right\} \tag{18}$$

Evident, pentru a obține compresie se vor aloca un număr redus de biți valorilor s şi o. În acest scop, aceste valori se vor cuantiza uniform rezultând $\bar{s}$ şi $\bar{o}$ (de obicei cu 5 respectiv 7 biți).

7. Algoritmului neadaptiv de compresie folosind tehnici fractale

O descriere a algoritmului elementar de compresie fractală este:

Inițializarea:

- Segmentarea imaginii - se împarte imaginea în blocuri de dimensiune fixă, de exemplu 8×8 pixeli. Blocurile rezultate le vom numi blocuri codomeniu R_i.
- Se creează o listă de blocuri domeniu D_i de dimensiune dublă în raport cu blocurile codomeniu, prin parcurgerea imaginii cu un pas l (de exemplu 1,4 sau 8) pe ambele direcții. Aceste blocuri se subeşantionează (mediază) pentru a avea dimensiunea egală cu codomeniile.
- Se calculează şi memorează pentru fiecare codomeniu Σr_i şi Σr_i^2 şi pentru fiecare domeniu Σd_i şi Σd_i^2.

Căutarea:

- Pentru fiecare codomeniu R se determină aproximarea optimă $R \approx s\,D + o\,\mathbf{1}$ astfel:
  - Pentru fiecare domeniu D_i se determină aproximarea optimă $R \approx s\,D_i + o\,\mathbf{1}$.
    - Se determină $\Sigma r_i d_i$.
    - Se determină conform formulelor (15) şi (16) coeficienții reali *s* şi *o*.
    - Se cuantizează uniform aceşti coeficienți obținând $\bar{s}$ şi $\bar{o}$.
    - Se recalculează $E(D_i,R)$ folosind coeficienții cuantizați.
  - Se determină domeniul D_k cu eroarea minimă $E(D_k, R) = \min_i E(D_i, R)$.
  - Se scriu în fişierul comprimat valorile cuantizate $\bar{s}$ şi $\bar{o}$ şi indicele domeniului ales (poziția sa).

S-au indicat cu caractere mai mici unele recomandări care duc la o implementare mai eficientă a algoritmului. O problemă a codificării fractale este aceea că selecția codomeniului se face după eroarea de colaj care este diferită de cea de la decodare, fapt care duce la o limitare a performanțelor.

În Figura 12 se prezintă rezultatele obținute în compresia fractală a imaginilor Lena şi Peppers având dimensiunea 512×512 şi 8 bpp. Codorul folosit prezintă următorii parametri:

- dimensiune codomeniu = 8×8, fixă
- dimensiune domeniu 16×16, fixă
- pasul la considerarea domeniilor = 8 pe fiecare axă
- căutare completă în mulțimea domeniilor
- cuantizare uniformă cu 5 respectiv 7 biți pentru s şi o (soluția uzuală)
- considerarea celor 8 izometrii.

În acest caz avem 64×64 = 4096 codomenii şi 63×63 = 3969 domenii. Dificultatea etapei de căutare constă în necesitatea realizării a 4096×3969×8 = 130.056.192 evaluări ale expresiei (18).

Raportul de compresie obținut în acest caz este

$$C = \frac{NrBitiPePixel \times DimensiuneBlocInPixeli}{NrBitiFactorScala + NrBitiOffset + \log_2[NrIzometrii \times NrDomenii]} \qquad (19)$$

În Figura 12 se prezintă şi eroarea introdusă de procesul de codare fractală. Pentru o mai bună vizualizare, imaginea s-a scalat astfel încât eroarea maximă să fie adusă la nivelul maxim de negru. Se constată că erorile sunt mai semnificative în zona detaliilor pronunțate, zonele uniforme (fără detalii) şi cele cu detalii medii reprezentându-se foarte bine.

Particularizând pentru cazul nostru concret avem un raport de compresie

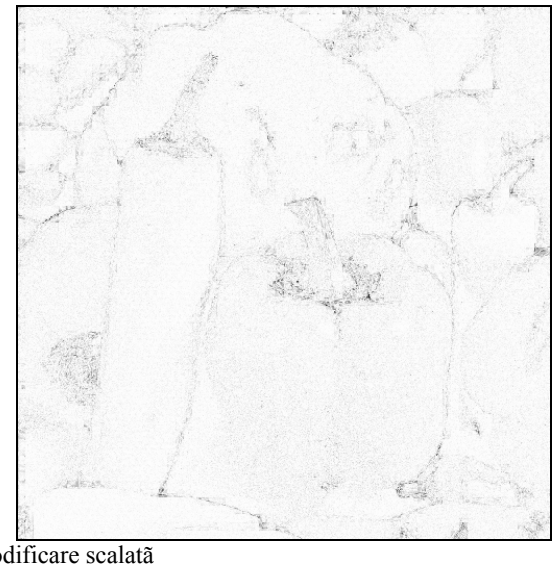
$$C = \frac{8 \times (8 \times 8)}{5 + 7 + \log_2[8 \times (64 \times 64)]} = \frac{512}{5 + 7 + 15} = 18.96 \qquad (20)$$

Imaginea Lena originală

Imaginea Peppers originală

Lena Rap.Compr. 18.96 PSNR = 31.32 dB

Peppers Rap.Compr. 18.96 PSNR = 31.61 dB



Eroarea de decodificare scalată

Figura 12 Rezultate ale algoritmului neadaptiv

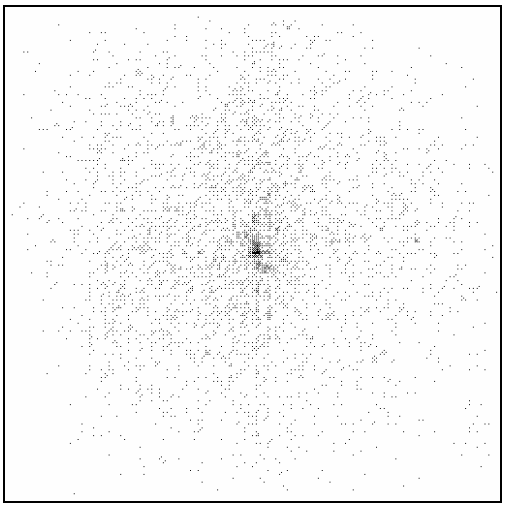


Figura 13 Statistica poziției codomeniu-domeniu

În implementare s-a făcut o simplificare codându-se pe câte 6 biți poziția domeniului pe fiecare axă (deşi poziția 31, nefiind validă, nu va apare niciodată - numărul de domeniilor fiind de fapt 63×63 şi nu 64×64 pentru a nu depăşi limita imaginii).

Se consideră în continuare statistica poziției domeniului în raport cu codomeniul. În Figura 13 se prezintă grafic această distribuție, fiecare pixel al imaginii reprezentând o realizare a acestei distribuții (pe ambele axe ale figurii domeniul de reprezentare este [-64,64]).

În Figura 14 se prezintă câteva corespondențe domeniu-codomeniu aşa cum au rezultat ele în urma codificării imaginii Lena. Se constată din nou tendința ca domeniul să fie apropiat codomeniului. Remarcăm perechile domeniu-codomeniu de pe umăr, din colțul dreapta-sus şi de pe borul pălăriei, asemănarea domeniu-codomeniu fiind în acest caz evidentă (în contextul transformării liniare permise). Această tendință de localitate a perechilor a condus la o metodă de creştere a vitezei de codificare prin considerarea doar a unei căutări locale (cu mici înrăutățiri asupra caracteristicii rată-distorsiune).

Accentuăm pe baza acestei figuri că, în urma căutării în cadrul întregii imagini, aceasta se codifică modelând fiecare bloc pe baza unei versiuni mărite a sa, fructificând astfel ***autoasemănările din cadrul imaginii***.

Decodificarea constă în aplicarea iterativă a transformării fractale asupra unei imagini inițiale. Acest pas determină caracterul de PIFS al metodelor fractale de compresie. Se constată practic că după mai puțin de 10 iterații, algoritmul atinge starea stabilă atractor (atinge punctul fix al transformării). Faptul că starea atractor nu depinde de imaginea inițială permite ca aceasta să fie una oarecare. În Tabelul 1 prezentăm calitatea imaginii după fiecare iterație în diferite cazuri de inițializare şi în Figura 15 evoluția sa grafică (pentru imaginea Lena).

Figura 14 Exemple de perechi domeniu-codomeniu

Atractor	Imagine inițială	Iterația									
		1	2	3	4	5	6	7	8	9	10
Lena	Negru	11.32	16.18	20.67	24.97	30.28	31.25	31.33	31.33	31.32	31.32
Lena	Peppers	16.02	20.74	25.17	29.02	31.15	31.31	31.32	31.32	31.32	31.32
Peppers	Negru	11.92	16.80	21.28	25.34	30.04	31.46	31.60	31.61	31.61	31.61
Peppers	Lena	15.70	20.62	24.77	28.90	31.32	31.58	31.60	31.61	31.61	31.61

Tabelul 1 Convergența procesului iterativ de decodificare (PSNR dB)

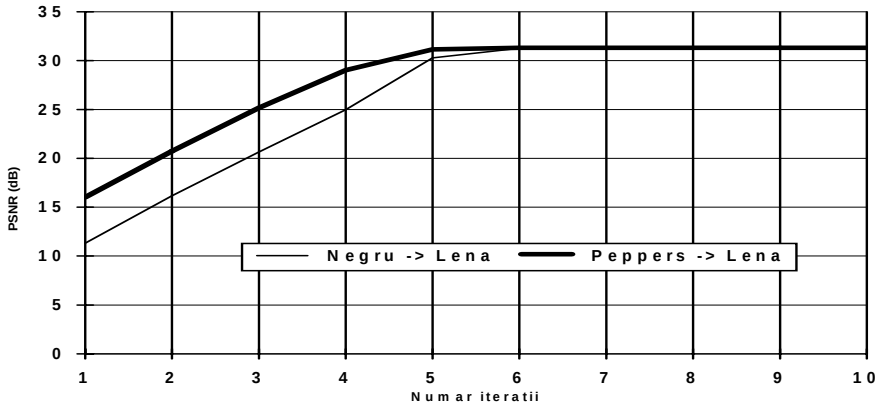


Figura 15 Convergența procesului iterativ de decodificare

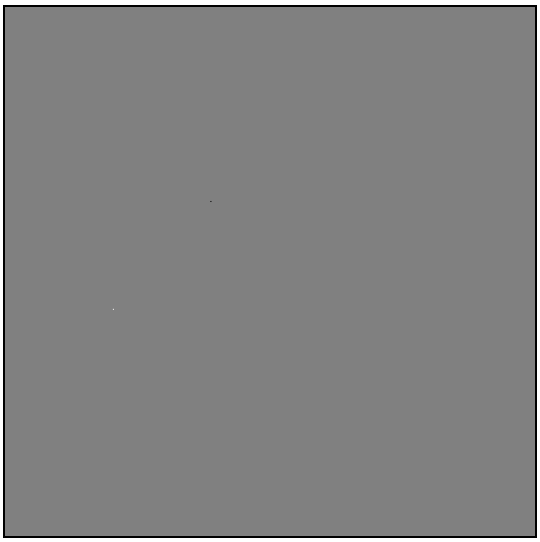
În Figura 16 şi Figura 17 prezentăm imaginile obținute după primele 5 iterații în cazul decodificării imaginii Lena atât în cazul inițializării cu o imagine neagră, cât şi în cazul inițializării cu imaginea Peppers. Se constată creşterea rezoluției la fiecare iterație în cazul inițializării uniforme. După prima iterație se poate remarca faptul că transformarea este definită la nivel de bloc şi că în cadrul unui bloc nu există încă detalii. Odată cu creşterea numărului de iterații rezoluția creşte, imaginea îmbunătățindu-se în detalii. După cum era de aşteptat, sistemul evoluează spre acelaşi atractor, remarcabil fiind însă că evoluția este mai rapidă în cazul inițializării cu o altă imagine decât în cazul inițializării uniforme. Această caracteristică poate fi utilă în cazul aplicării acestei tehnici unei secvențe video prin inițializarea decodificării fiecărui cadru cu cadrul anterior. Cum, în general, cadrele sunt apropiate, convergența procesului de decodificare este o dată în plus crescută.

8. Algoritm adaptiv de compresie folosind partiționare quadro

După cum se poate constată din subcapitolul anterior, eroarea obținută prin acest algoritm este neuniform distribuită în cadrul imaginii: mai mică în zonele uniforme şi mai mare în zonele bogate în detalii. Pe de altă parte, în cadrul algoritmului de codare fractală se impune ca domeniul să fie dublul codomeniului (deşi nu este unica soluție din punct de vedere teoretic, este singura întâlnită în practică) dar nu se face nici o referire la mărimea codomeniului.

Aceste considerente conduc la ideea de a folosi o partiționare adaptivă a imaginii (şi nu una fixă ca anterior). În limita unei erori acceptate, zonele omogene ale imaginii se vor codifica folosind blocuri de dimensiune mai mare, în timp ce pentru zonele bogate în detalii se va folosi o partiționare în blocuri de dimensiune mai mică. În acest fel imaginea este codificată mai uniform din punct de vedere al erorii (locale) dar codorul obținut prezintă o rată de bit variabilă. Un avantaj major al acestor metode adaptive este acela că se reduce numărul de biți necesari stocării coeficienților transformării (având mai puține codomenii).

Soluția, devenită ulterior "clasică", este aceea de partiționare folosind arbori quadro ("**quadtree partitioning**"). În această abordare se consideră pentru un codomeniu diferite dimensiuni cum ar fi 32,



Imaginea inițială (negru)

Iterația 1 (PSNR 11.32 dB)

Iterația 2 (PSNR 16.18 dB)

Iterația 3 (PSNR 20.67 dB)

Iterația 4 (PSNR 24.97 dB)

Iterația 5 (PSNR 30.28 dB)

Figura 16 Primele 5 iterații ale secvenței de decodificare Negru → Lena
(Iterația 10 este prezentată în Figura 12)

Imaginea inițială (Peppers)

Iterația 1 (PSNR 16.02 dB)

Iterația 2 (PSNR 20.74 dB)

Iterația 3 (PSNR 25.17 dB)

Iterația 4 (PSNR 29.02 dB)

Iterația 5 (PSNR 31.15 dB)

Figura 17 Primele 5 iterații ale secvenței de decodificare Peppers → Lena

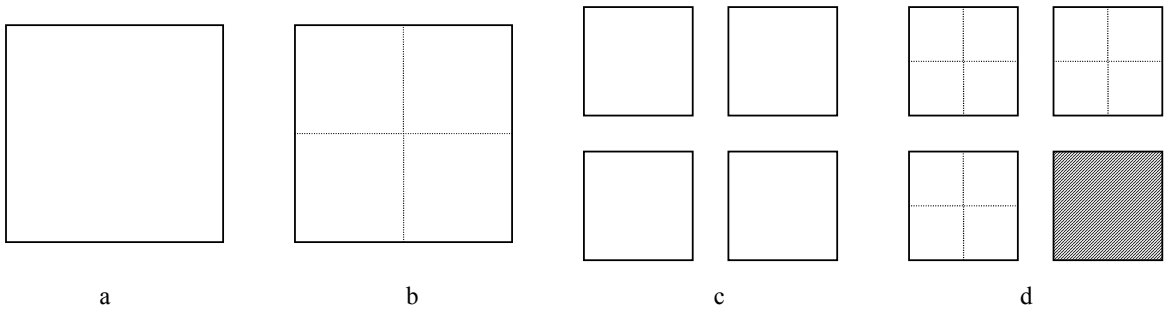


Figura 18 Un exemplu de aplicare a partiționării quadro

16, 8, 4. Se încearcă codificarea folosind blocuri de dimensiune maximă. Dacă acest lucru nu se poate face în limitele unui criteriu de eroare impus, blocul se divide în 4 fii şi se încearcă codificarea independentă a fiecărui fiu. Acest procedeu de divizare poate continua până în momentul atingerii dimensiunii minime a blocului, când codificarea se face pe baza acestei dimensiuni indiferent de îndeplinirea criteriului de eroare. Un exemplu de divizare este dat în Figura 18. Blocul (a), care nu respectă criteriul de eroare, este divizat (b) în patru fii (c). Aceia dintre ei care nu respectă criteriul de eroare sunt la rândul lor divizați (d). Cei care respectă criteriul de eroare (blocul haşurat) sunt folosiți ca şi codomeniu în cadrul PIFS.

O descriere algoritmică este dată în continuare:

Algoritm "quadtree":

1. Se alege o valoare a erorii medii pătratice ca şi criteriu limită de eroare. Se aleg limitele pentru dimensiunea blocului. Se partiționează imaginea folosind dimensiunea maximă a blocului.
2. Se inițializează o stivă şi se introduc în ea blocurile de dimensiune maximă.
3. Cât timp stiva nu este vidă se execută următorii paşi:
   - Se extrage din stivă un bloc codomeniu R şi se caută domeniul D pe baza căruia R poate fi aproximat optim $R \approx sD+o1$. Se determină eroarea E(R,D).
   - Dacă criteriul de eroare este satisfăcut sau dacă dimensiunea blocului este cea minimă, se scriu în fluxul de date comprimat valorile s, o, izometria şi poziția lui D.
   - Altfel, se divide blocul în cele 4 blocuri fiu şi se introduc aceste blocuri în stivă.

Deşi algoritmul a fost descris în termenii implementării unei stive, în practică de cele mai multe ori se consideră o implementare recursivă în care "stiva" algoritmului este înlocuită de stiva sistemului utilizată de compilator pentru apelul (recursiv) de funcții.

Algoritmul de determinare a domeniului optim prin căutare şi transformările considerate în maparea domeniu-codomeniu sunt aceleaşi ca şi la algoritmul neadaptiv. Decodificarea se face de asemenea prin aplicarea iterativă a transformărilor pe o imagine inițială.

Un avantaj al acestui algoritm este acela că, prin alegerea de diferite valori pentru criteriul de eroare, se pot obține diferite rapoarte de compresie şi calități ale imaginii putând astfel trasa, prin puncte, caracteristica rată-distorsiune a metodei (prezentată în Tabelul 2 şi în Figura 19). Un dezavantaj al acestei metode constă în faptul că nu se pot prescrie calitatea imaginii refăcute şi rata de bit (raportul de compresie) dorite în mod direct.

Prezentăm în Figura 20 imaginile obținute prin considerarea pentru criteriul de eroare a valorilor 8 şi 20 care au dus la obținerea unor rapoarte de compresie de 39.59 respectiv 93.48 în condițiile unui PSNR de 30.10 dB respectiv 25.88 dB. Ca şi în cazul algoritmului neadaptiv, pentru vizualizarea erorii imaginea a fost scalată astfel încât erorii maxime să îi corespundă nivelul maxim de negru.

Toleranța	Lena			Peppers		
	Timp (sec)	Raport Compresie	PSNR (dB)	Timp (sec)	Raport Compresie	PSNR (dB)
2	304	20.50	30.87	292	20.23	30.86
4	220	27.21	30.78	246	23.94	30.80
6	181	33.07	30.51	191	30.84	30.58
8	159	39.59	30.10	156	37.58	30.24
10	128	46.76	29.46	136	43.37	29.75
12	112	53.52	28.85	123	48.78	29.27
14	100	59.83	27.28	108	54.84	28.74
16	88	69.88	27.28	96	63.04	27.90
18	77	79.65	26.77	85	70.86	27.03
20	66	93.48	25.88	76	80.19	26.39
22	58	108.23	25.27	63	95.56	25.48
24	47	134.08	24.43	53	115.68	24.59

Tabelul 2 Performanţele metodei "quadtree" în funcție de criteriul de eroare

Se constată partiționarea diferită din cadrul imaginii: în zonele uniforme codorul a ales blocuri de dimensiune mai mare, iar în cele cu detalii blocuri de dimensiune mai mică (minimă). În cazul unui criteriu de eroare mai restrictiv (primul caz în Figura 20) partiționarea a fost mai fină, eroarea obținută mai mică, dar prețul plătit a fost mai mare (în sensul numărului de biți necesari codificării transformării tuturor blocurilor şi implicit al raportului de compresie obținut). Raportul de compresie poate fi crescut suplimentar prin aplicarea unei codări entropice.

9. Compresia fractală a secvențelor video

Aplicarea tehnicilor fractale în **domeniul video** este de dată mai recentă, tehnicile fractale dezvoltându-se în contextul imaginilor alb-negru (în nivele de gri) statice. În principal, abordările fractale ale compresiei video au la bază următoarele:

- **Codare cadru cu cadru** bazată pe tehnici fractale. Se folosesc uneori informații de mişcare pentru a restrânge (accelera) căutarea. Tehnica este o extensie simplistă a metodelor dedicate imaginilor statice (Figura 21).

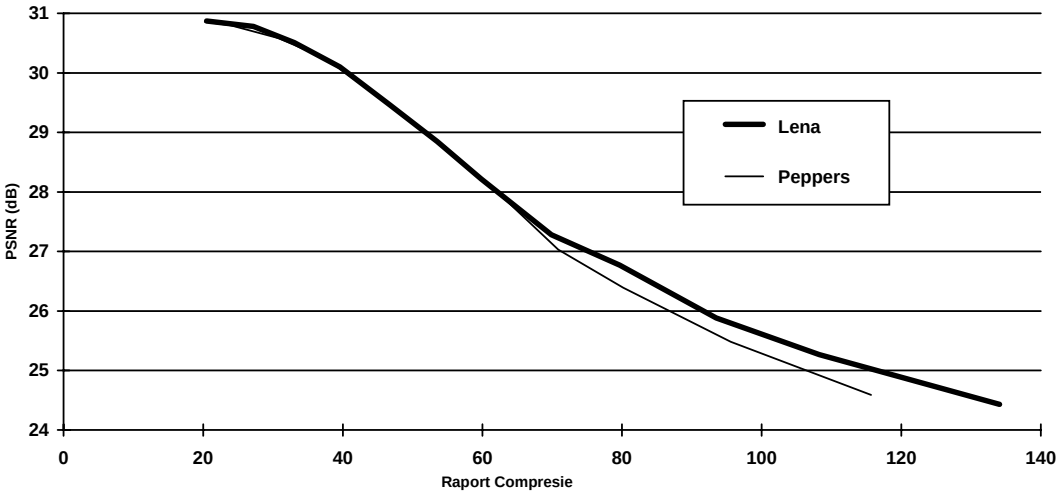
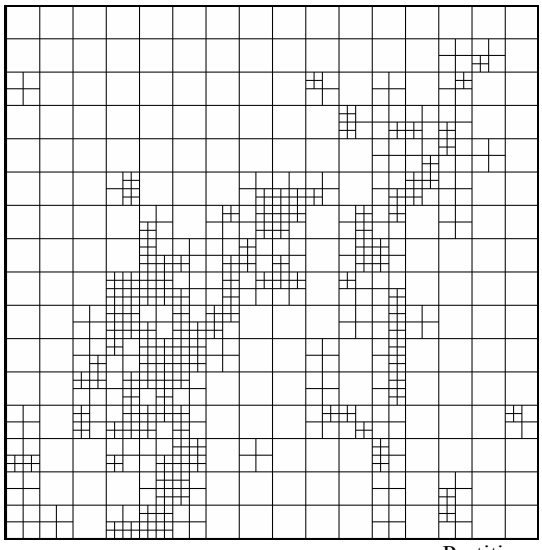
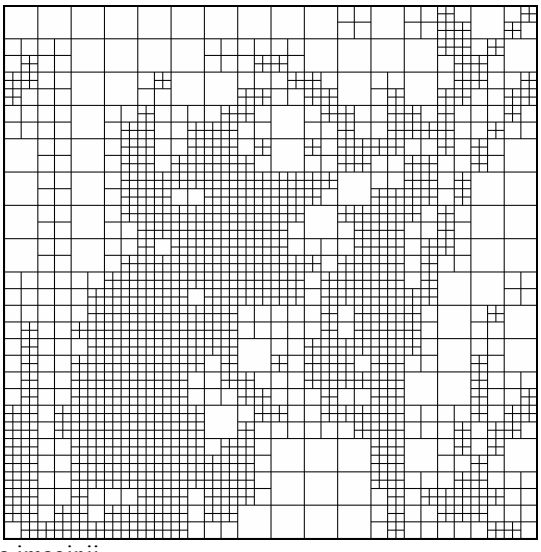


Figura 19 Caracteristica Rată-Distorsiune

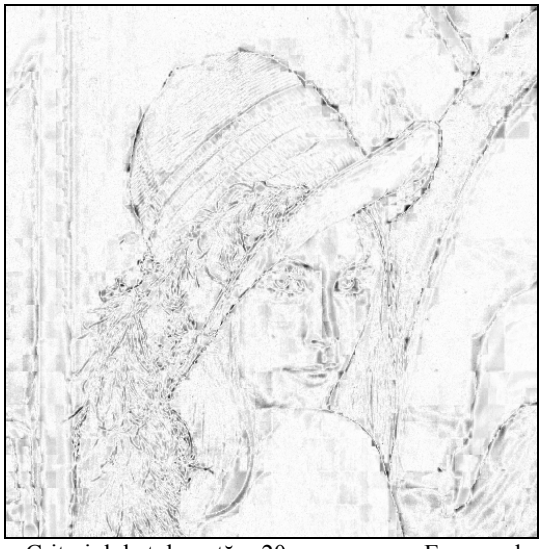
Rap.Compr. = 93.48 PSNR=25.88 dB

Rap.Compr. =39.59 PSNR=30.10 dB





Partiționarea imaginii



Criteriul de toleranță = 20

Eroarea de refacere (scalată)

Criteriul de toleranță = 8

Figura 20 Rezultatele obținute folosind codorul adaptiv bazat pe partiționare quadro

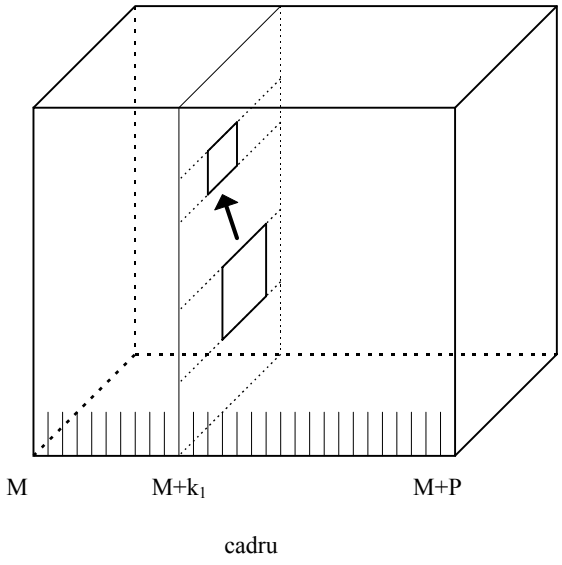


Figura 21 Codare cadru cu cadru

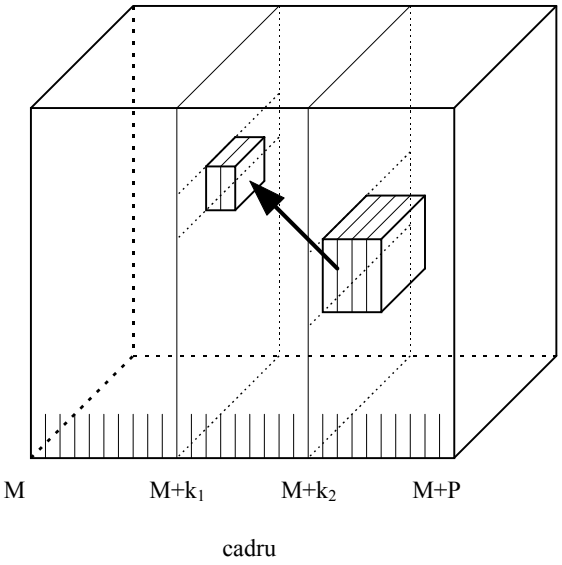


Figura 22 Codare de volum

- **Codare de volum** ("cube coding scheme"). Esența acestor metode constă în extinderea sistemelor PIFS de la 2D la 3D prin considerarea timpului (numărului de cadru) ca şi a treia dimensiune şi determinarea unei mapări contractive (pe toate cele trei dimensiuni). Secvența de codat se împarte în secvențe de cadre denumite G.O.P. ("Group of Picture") care se codează împreună. În loc să se împartă fiecare cadru în codomenii pătratice, se împarte "volumul" unui asemenea G.O.P. în cuburi codomeniu (mai general în paralelipipede, de cele mai multe ori dreptunghice) nesuprapuse care se codifică ca şi PIFS. Transformările PIFS considerate mapează fiecare asemenea volum domeniu într-un volum codomeniu după cum se indică în Figura 22.