

UN ALGORITM DIN CLASA VPIC PENTRU COMPRIMAREA IMAGINILOR CU MAI MULTE NIVELE DE GRI.

mat. Breabăn Ancuța

Institutul de Cercetări în Informatică

Rezumat:

Multiplicarea și dezvoltarea aplicațiilor care gestionează și prelucurează imagini a determinat în ultimii ani necesitatea găsirii unor algoritmi performanți de comprimare/decomprimare a informației vizuale, performanțele măsurându-se atât prin timpul de răspuns, cât și prin rata de comprimare realizată. În această lucrare este prezentat un algoritm original de comprimare bazat pe modele vizuale, care poate fi încadrat în clasa algoritmilor cu trunchiere pe blocuri, utilizând în plus și tehnici din clasa de algoritmi cu cuantificare vectorială.

1. Tehnici de comprimare a imaginilor. Noțiuni generale.

În ultimii ani asistăm la o creștere spectaculoasă a domeniului prelucrării imaginilor și vederii artificiale, începând cu aplicațiile industriale, de inspecție optică de calitate, de proiectare tehnică și tehnologică asistată de calculator și terminând cu comunicațiile video prin satelit și cu sistemele multimedia. Din studiile efectuate de IBM [19] rezultă că numai aproximativ 10% din informațiile dintr-o întreprindere pot fi stocate în mod codificat, majoritatea informațiilor impunând cu necesitate stocarea sub formă de imagini. Cu toate progresele realizate în domeniul tehnicii de calcul în stocarea și prelucrarea imaginilor, prin apariția hardware-ului specializat și a tehnologiilor de tip VLSI, având în vedere volumul imens de date supuse prelucrării, reducerea spațiului de memorie utilizat pentru reprezentarea (codificarea) imaginilor, proces cunoscut sub numele de comprimare a imaginilor, (image sau picture compression), rămâne una din problematicile majore ale prelucrării digitale a imaginilor, subiect actual de cercetare teoretică și practică.

În ultimii ani, organisme ca CCITT (Comité Consultatif International pour la Télégraphie et Téléphonie), ISO (International Standard Organisation), JPEG (Joint Photographics Experts Group), firme producătoare de hardware și software specializat pentru aplicații în care sînt utilizate imagini (INTEL, IBM, Gerber Scientific Instrument Company...), laboratoare de cercetare de renume, în domeniul prelucrării imaginilor, au ca preocupare constantă găsirea unor algoritmi performanți în domeniul comprimării imaginilor.

Caracteristicile imaginilor determinante în proiectarea

unui algoritm de comprimare sînt următoarele:

- elementele vecine din imagine sînt puternic corelate, ceea ce face ca informația să aibă un mare grad de redundanță; primul obiectiv în proiectarea unui algoritm de comprimare îl constituie găsirea unei reprezentări în care elementele vecine să fie decorelate;
- imaginile au numeroase regiuni omogene, în care nivelul de gri al elementelor este același; aceasta face ca, pentru reducerea complexității calculului, să fie indicată divizarea imaginii în blocuri; alegerea dimensiunii blocurilor trebuie să urmărească diminuarea corelației dintre elementele de imagini din blocuri vecine, care conduce la distorsiuni ("block effect");
- o altă caracteristică a imaginilor o constituie faptul că numai o parte a informației este semnificativă sau poate fi receptată de observatorul uman; astfel, spre exemplu, numărul de nivele de gri sau de culori poate fi redus substanțial, fără ca receptorul uman să poată percepe diferența dintre imaginea astfel obținută și imaginea originală.

Tehnicile de comprimare utilizate în prelucrarea de imagini pot fi împărțite în două clase:

- tehnici fără pierderi, caracteristice aplicațiilor unde se cere memorarea cu maximă fidelitate a imaginii;
- tehnici cu pierderi, utilizate în special pentru aplicațiile în care imaginile sînt vizualizate de un receptor uman sau în aplicațiile în care informația vizuală este redundantă.

În timp, tehnicile de comprimare au evoluat de la tehnicile care utilizează algoritmi și metode din teoria matematică a informației, către tehnicile considerate de generația a II-a, care se bazează pe cunoștințele acumulate în percepția biologică a imaginilor.

Astfel, tehnicile de generația I sînt tehnicile de cuantificare vectorială ("vector quantization"), tehnicile bazate pe transformarea imaginilor ("transform compression"), tehnicile predictive ("predictive compression") și tehnicile bazate pe aproximarea blocurilor din imagine ("block truncation compression"), toate aceste tehnici avînd ca idei de bază reducerea redundanței informației și decorelarea elementelor din imagine. Pentru măsurarea calității imaginilor comprimate, sînt utilizate măsurile standard din teoria informației, care nu au nici o legătură cu atributele semnificative ale imaginii sau cu noțiunile perceptuale de fidelitate a imaginii.

Începînd din anul 1983 [2], își fac apariția în literatura de specialitate tehnicile de comprimare a imaginilor de generația a II-a, care apropie acest domeniu de biologie, ținînd seama atât de proprietățile vizuale ale imaginilor, cât și de proprietățile vederii biologice. O descriere de detaliu a caracteristicilor tehnicilor de generația a II-a este prezentată în [10].

În prezent, tendințele observate în domeniul

comprimării imaginilor sînt următoarele:

- se încearcă o standardizare în domeniul comprimării, decompimării și al reprezentării imaginilor, atât în cadrul organismelor specializate (ISO, CCITT), cît și al firmelor producătoare (exemplu: IBM);
- pentru obținerea unui timp de răspuns acceptabil, în cadrul sistemelor aplicative se utilizează hardware specializat;
- rata de comprimare obținută prin utilizarea algoritmilor clasici de comprimare este cuprinsă între 6:1 și 10:1; un raport mai bun de comprimare se obține în cazul sistemelor dedicate, în care se ține cont de caracteristicile imaginii;
- rezultatele raportate de firmele producătoare (INTEL) și de grupurile JPEG, MPEG sînt obținute prin utilizarea unor tehnici care combină algoritmi clasici cu algoritmi de generația a II-a.

În această lucrare se propune un algoritm original pentru comprimarea imaginilor cu mai multe nivele de gri, algoritm care are la bază idei din tehnica de comprimare cu modele vizuale ("Visual Pattern Image Coding"-VPIC) introdusă de D. Chen și A.C. Bovik în [10], modificate și completate cu metode similare celor utilizate în tehnicile de tip VQ ("vector quantization"), ceea ce conduce la o rată de comprimare mai mare decît cea anunțată în [10]. Pentru ca îmbunătățirea comprimării să nu se facă în detrimentul timpului de răspuns, am modificat parțial caracteristicile imaginilor și algoritmi de calcul propuși în [10], în sensul eliminării totale a operațiilor consumatoare de timp. Am menținut, în schimb, pentru început acei parametri definiți în tehnica VPIC, care asigură calitatea imaginii comprimate, conform rezultatelor raportate în [10], urmînd ca într-o etapă ulterioară să experimentăm și alte tipuri de caracteristici și de metode de calcul.

2. Prezentarea algoritmului

În această lucrare prin "imagine" se va înțelege o "imagine digitală", reprezentată printr-o funcție $f(x,y)$ cu valori numere întregi, unde x și $y \in Z$ sînt coordonate carteziene ale unui punct din planul imaginii, iar $f(x,y)$ reprezintă valoarea nivelului de gri al punctului din planul imaginii, numit și element de imagine, punct digital sau pixel.

Prin comprimarea imaginii se înțelege procesul de reprezentare a unei imagini într-o nouă formă f' , care să ocupe un spațiu de memorie mai mic, păstrîndu-se integritatea "observabilă" a imaginii originale.

Algoritm propus în această lucrare consideră imaginea împărțită în blocuri B_{ij} de $4 * 4$ pixeli, fiecare bloc putînd fi sau un bloc uniform, în care toate elementele au același nivel de gri sau un bloc de tip muchie.

$$B_{ij} = \{ f(4i+k, 4j+l) / k, l=0,1,2,3 \}$$

Pentru a determina tipul blocului, algoritmul este următorul: Se calculează μ , Δ_x și Δ_y pentru fiecare bloc, unde μ este media aritmetică, iar Δ_x , Δ_y se calculează după formulele:

$$\Delta_x = \frac{\sum_{k=0}^3 [f(4i+2,4j+k) + f(4i+3,4j+k) - f(4i,4j+k) - f(4i+1,4j+k)]}{8}$$

$$\Delta_y = \frac{\sum_{k=0}^3 [f(4i+k,4j+2) + f(4i+k,4j+3) - f(4i+k,4j) - f(4i+k,4j+1)]}{8}$$

Se observă că Δ_x și Δ_y au valori cuprinse între 0 și $N-1$, unde N este numărul de nivele de gri din imagine. Magnitudinea gradientului pentru fiecare bloc se determină utilizînd relația:

$$\nabla = \nabla(|\Delta_x|) + \nabla(|\Delta_y|)$$

unde $|\cdot|$ definește partea întregă a numărului, iar ∇ este un vector ale cărui componente sînt $0, 1^2, 2^2, 3^2, \dots, (N-1)^2$.

Magnitudinea gradientului este utilizată pentru determinarea blocurilor uniforme: se alege un prag t_1 pentru care, dacă $\nabla < t_1$, blocul este considerat uniform. Altfel, blocul este considerat de tip muchie și algoritmul continuă cu determinarea orientării și polarității muchiei, pentru găsirea modelului ideal căruia îi corespunde aceasta.

Cele patru modele ideale, necesare și suficiente pentru reprezentarea blocurilor de tip muchie sînt următoarele:

$$\begin{matrix} \begin{pmatrix} + + + + \\ + + + + \\ - - - - \\ - - - - \end{pmatrix} & \begin{pmatrix} + + - - \\ + + - - \\ + + - - \\ + + - - \end{pmatrix} & \begin{pmatrix} + + + + \\ + + + - \\ + + - - \\ + - - - \end{pmatrix} & \begin{pmatrix} + + + + \\ - + + + \\ - - + + \\ - - - + \end{pmatrix} \\ \text{(a)} & \text{(b)} & \text{(c)} & \text{(d)} \end{matrix}$$

Figura 1.

Fiecărui model de bază îi corespund mai multe modele echivalente ca orientare, dar cu polaritate diferită. Altfel, spre exemplu, modelul (c) din figura 1 este echivalent cu modelele din figura 2.

$$\begin{matrix} \begin{pmatrix} + + + + \\ + + - - \\ + - - - \\ + - - - \end{pmatrix} & \begin{pmatrix} - - - - \\ - - - + \\ - - + + \\ - + + + \end{pmatrix} \\ \text{(e)} & \text{(f)} \end{matrix}$$

Figura 2.

Dacă modelul (e) poate fi substituit prin modelul (c), cu dezavantajul introducerii unor zgomote, poziția valorilor fiind simbolizate în figură prin '+' și '-', diferențierea între modelele (c) și (f) trebuie determinată în mod unic.

Orientarea muchiei se determină în modul următor:

se aleg două praguri t_1, t_2 astfel ca:

dacă $|\Delta_y| < t_1$ și $|\Delta_x| > t_2$, atunci $\theta = 0^0$;

dacă $|\Delta_x| < t_1$ atunci $\theta = 90^0$;

altfel, dacă $\text{sign } \Delta_x = \text{sign } \Delta_y$, atunci $\theta = 45^\circ$;
altfel $\theta = -45^\circ$

Pentru determinarea poziției valorilor de nivel de gri se procedează astfel:

dacă $\theta = 0^\circ$ și $\Delta_x > 0$, atunci poziția = 1;
dacă $\theta = 90^\circ$ și $\Delta_y > 0$, atunci poziția = 1;
dacă $\theta = 45^\circ$ și $\Delta_x > 0$, atunci poziția = 1;
dacă $\theta = -45^\circ$ și $\Delta_x < 0$, atunci poziția = 1;
altfel, poziția = 0.

Recapitulând, un bloc de 4x4 elemente poate fi uniform, și în acest caz valoarea mediei μ este suficientă pentru reconstruirea imaginii sau de tip muchie, și în acest caz pentru reconstruire sînt necesare următoarele:

- valoarea mediei;
- Δ_x sau Δ_y ;
- orientarea muchiei și poziția nivelelor de gri.

Pentru obținerea unei rate de compresie superioară, se grupează blocurile similare din imagine astfel:

- blocurile uniforme se grupează ierarhic în superblocuri de dimensiuni variabile multiplu de 4: 4x8, 8x4, 8x8, 8x16...

Descrierea acestor superblocuri se realizează prin memorarea valorii medii și a adresei de început și de sfîrșit a superblocului (colțul stînga sus și colțul dreapta jos al dreptunghiului care îl încadrează).

Reprezentarea în memorie a blocurilor uniforme se va face printr-o structură de date variabilă:

VALOARE MEDIE	ADRESA DE ÎNCEPUT		ADRESA DE SFÎRȘIT	
	x	y	x	y	

- blocurile de tip muchie nu se grupează în superblocuri, de aceea pentru ele nu este necesară memorarea adresei de sfîrșit a blocului.

Reprezentarea blocurilor de tip muchie se face printr-o structură de date variabilă, astfel:

VALOARE MEDIE	Δ	INDEX MODEL	POZIȚIE NIVEL DE GRI	ADRESA DE ÎNCEPUT	
	Δ_x sau Δ_y			x	y

Reconstrucția imaginii se face prin parcurgerea structurilor de date astfel definite și determinarea nivelelor de gri din modele. Dacă notăm nivelele de gri cu a și b, poziția 1 are semnificația practică $a > b$, deci a va înlocui simbolul '+' din model, iar b va înlocui simbolul '-'.

Valorile nivelelor de gri notate a și b se determină astfel:

$$\text{pentru modelul (a): } a = \mu + \frac{\Delta_x}{2} \quad b = \mu - \frac{\Delta_x}{2}$$

$$\text{pentru modelul (b): } a = \mu + \frac{\Delta_y}{2} \quad b = \mu - \frac{\Delta_y}{2}$$

$$\text{pentru modelul (c): } a = \mu + \frac{3\Delta_x}{4} \quad b = \mu + \Delta_x - \frac{\Delta_x}{4}$$

$$\text{pentru modelul (d): } a = \mu + \Delta_x - \frac{\Delta_x}{4} \quad b = \mu - \Delta_x - \frac{\Delta_x}{4}$$

Se observă că algoritmul de comprimare necesită:

- pentru blocuri uniforme, 49 operații de adunare, o împărțire la 2 și o comparație;
- pentru blocuri de tip muchie, 49 operații de adunare, o împărțire la 2 și 4 comparații.

Teoretic, algoritmul de reconstrucție (decomprimare) durează mai puțin, necesitînd maximum trei operații de adunare și o împărțire la 2 pentru reconstrucția fiecărui bloc.

3. Concluzii

Volumul imens de date supuse prelucrărilor în aplicațiile care utilizează imagini, face ca problema codificării imaginilor să constituie punctul critic al sistemelor respective. Cu toate progresele înregistrate prin apariția hardware-ului specializat și a tehnologiilor de tip VLSI, această problemă rămîne în continuare un subiect de cercetare de mare actualitate.

Algoritmul de comprimare/decomprimare a imaginilor propus în această lucrare face parte din categoria algoritmilor de generația a II-a, care au la bază caracteristicile vizuale ale imaginii. Este un algoritm de tip "lossy" (cu pierderi) care utilizează modelele (pattern-urile) vizuale ale blocurilor din imagine, acestea putînd fi de tip muchie sau uniforme.

Comparat cu algoritmul similar prezentat în [10], algoritmul propus oferă un timp de răspuns mai bun și o rată de compresie superioară, în condițiile menținerii calității imaginii.

Algoritmul a fost implementat în limbajul C sub sistemul de operare MS-DOS în cadrul bibliotecii de programe de comprimare/decomprimare a imaginilor, CODEC, realizată în ICI în anul 1991.

Rezultatele obținute în urma testărilor efectuate asupra unor imagini cu 256 nivele de gri de dimensiune 256x256 sînt deplin satisfăcătoare.

BIBLIOGRAFIE

1. ROSENFELD, A., KAK, A.C.: *Digital Picture Processing*, Academic Press, New York, 1982.
2. BART, P., ADELSON, E.: *The Laplacian Pyramid as a Compact Image Code*. In: IEEE Trans. COM 31, nr. 4, April 1983, pp 532-540.
3. CABRELLI, C.A., MOLTER, V.M.: *Automatic Representation of Binary Images*. In: IEEE Trans. PAMI 12, nr. 12, dec 1990, pp. 1190-1196.
4. MANOHAR, M., RAO, P.S., IYENGAR, S.S.: *Template Quadrees for Representing Region and Line Data Present in Binary Images*; Co. Vi, Gr. and Im Proc. nr. 51, 1990, pp 338-354.
5. KELLER, J.M., CHEN, S.: *Texture Description and Segmentation through Fractal Geometry*; Co. Vi, Gr. and Im Proc. nr 45, 1989, pp. 150-166.
6. YOKOYO, N., YAMAMOTO, K.: *Fractal Based Analysis and Interpolation of 3D Natural Surface Shapes and their Application to Terrain Modelling*; Co Vi, Gr. and Im Proc, nr. 46, 1989, pp. 284-302.

7. **STORER, J.A.:** *Data Compression. Methods and Theory*, Computer Science Press, 1987.
8. **PRATT, W.K.:** *Digital Image Processing*, John Wiley & Sons, New York, 1978.
9. **ROSE, K., HEIMAN, A., DINSTEIN, I.:** *DCT/DST Alternate Transform Image Coding*. In: IEEE COM 38, nr. 1, Jan. 1990, pp. 94-101.
10. **CHEN, D., BOVIK, A.C.:** *Visual Pattern Image Coding*. In: IEEE Com 38, nr. 12, dec. 1990, pp. 2137-2145.
11. **CHEUNG, W.N.:** *Calculation of Conditional Probabilities in Hierarchical Block Coding Systems*. In: IEEE Com 38, nr. 6, June 1990, pp. 747-748.
12. **ALGAZI V.R., s.a.:** *Compression of Binary Facsimile Images by Processing and Color Shrinking*. In: IEEE com 38, nr. 9, sept. 1990, pp 1592-1598.
13. **MARTENS J.B.:** *Application of Scale Space to Image Coding*. In: IEEE Com 38, nr. 9, Sept. 1990, pp 1585-1598.
14. **PEARLMAN W.A.:** *Adaptive Cosine Transform Image Coding with Constant Block Distortion*. In: IEEE Com 38, nr. 5, May 1990, pp 698-703.
15. **CHEN T.C.:** *A Lattice Vector Quantization Using a Geometric Decomposition*. In: IEEE Com 38, nr. 5, May 1990, pp 704-714.
16. **ZEGER, K., GERSHO, A.:** *Pseudo - Gray Coding*. In: IEEE Com 38, nr. 12, Dec. 1990, pp 2147-2158.
17. **NASRABADI, N.M., FENG, Y.:** *Image Compression Using Address - Vector Quantization*. In: IEEE Com 38, nr. 12, Dec. 1990, pp 2166-2173.
18. **DINSTEIN, I., ROSE, K., HEIMAN, A.:** *Variable Block-Size Transform Image Coder*. In: IEEE Com 38, nr. 11, Nov. 1990, pp 2073-2078.
19. * IBM System Journal, Vol 29, nr. 3, 1990.