

TEHNICI DE COMPRESIE A IMAGINILOR

C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



Compresie = reducerea cantitatii de date necesare pentru reprezentarea unei imagini

Compresia trebuie sa fie reversibila (functie inversabila).

Compresie

fara pierderi (eficienta sursei de informatie, Th. 1 Shannon)
cu pierderi

Compresie

in domeniul valorilor (cuantizare, reducere numar de culori)
instante ale algoritmilor de clustering

in domeniul spatial si al valorilor

C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



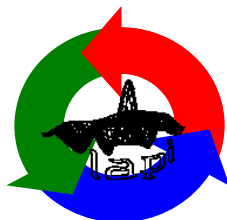
Cerinte contradictorii :

Raport de compresie :: Calitatea imaginii reconstruite

(cantitate de date SNR, PSNR, MSE)

C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



Compresie fara pierderi = codarea surselor de informatie pentru canale **fara** perturbatii

In aceasta categorie intra metode generale (indiferente de semnificatia valorilor) sau metode dedicate continutului de tip imagine (se bazeaza pe corelatia valorilor vecine).

Metode generale :

- codare entropica (Huffman)

- codare aritmetica

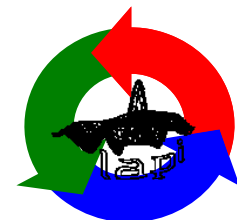
- codare Ziv-Lempel

Metode specifice:

- RLE (Run Length Encoding)

- WBS (White Block Skipping)

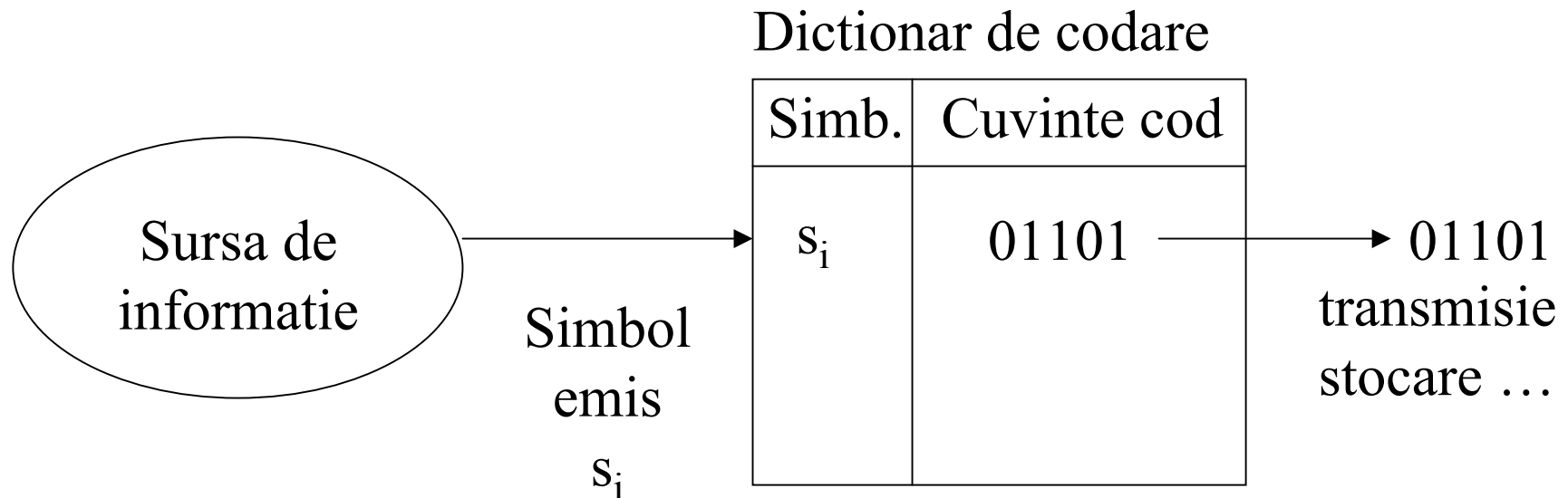
C. VERTAN



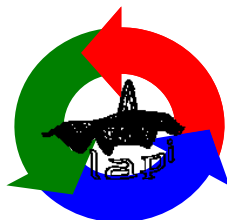
Codarea Huffman

Este o codare cu lungime variabila a cuvintului de cod si dictionar de cuvinte de cod (dictionarul - sau Look-Up Table - arata care sunt cuvintele de cod ce corespund unui anumit mesaj al sursei).

Codarea inseamna construirea dictionarului de codare.



C. VERTAN



Principiul de codare :

cuvintele de cod de lungime mica (scurte) se alocă simbolurilor
mai probabile

cuvintele de cod de lungime mare (lungi) se alocă simbolurilor
mai puțin probabile

Algoritm :

construirea de surse de informație restrânse prin reunirea simbolurilor
cele mai puțin probabile ale sursei anterioare.

C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



Codarea Huffman

Sursa de informatie emite 6 simboluri, cu probabilitatile

$$S = [s_1 ; s_2 ; s_3 ; s_4 ; s_5 ; s_6]$$

$$P = [0.25 ; 0.3 ; 0.2 ; 0.1 ; 0.1 ; 0.05].$$

s_2 0.3

s_1 0.25

s_3 0.2

s_4 0.1

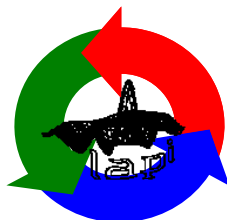
s_5 0.1]

s_6 0.05]

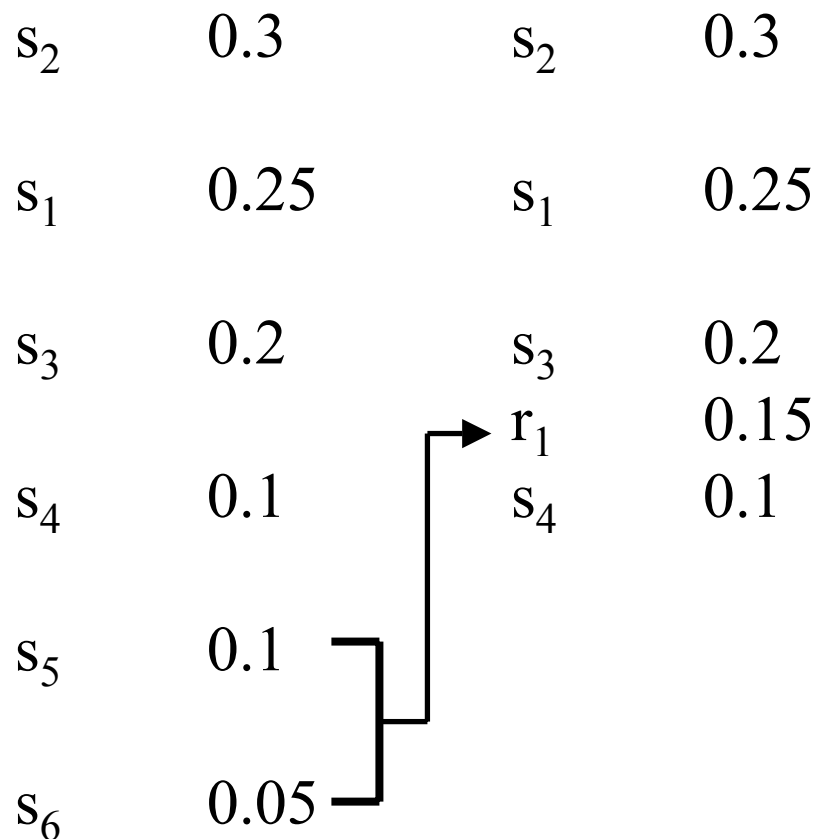
Ordoneaza simbolurile sursei
in ordine descrescatoare a
probabilitatilor.

Ultimele doua simboluri (cele mai
putin probabile) sunt reunite intr-un
singur simbol, formand o
sursa restransa.

C. VERTAN

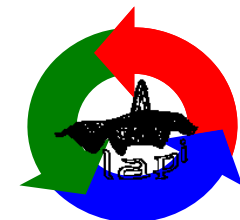


Codarea Huffman



Continua procedeul
de restrangere pana
la obtinerea unei
surse restranse cu
2 simboluri.

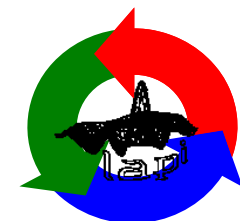
C. VERTAN



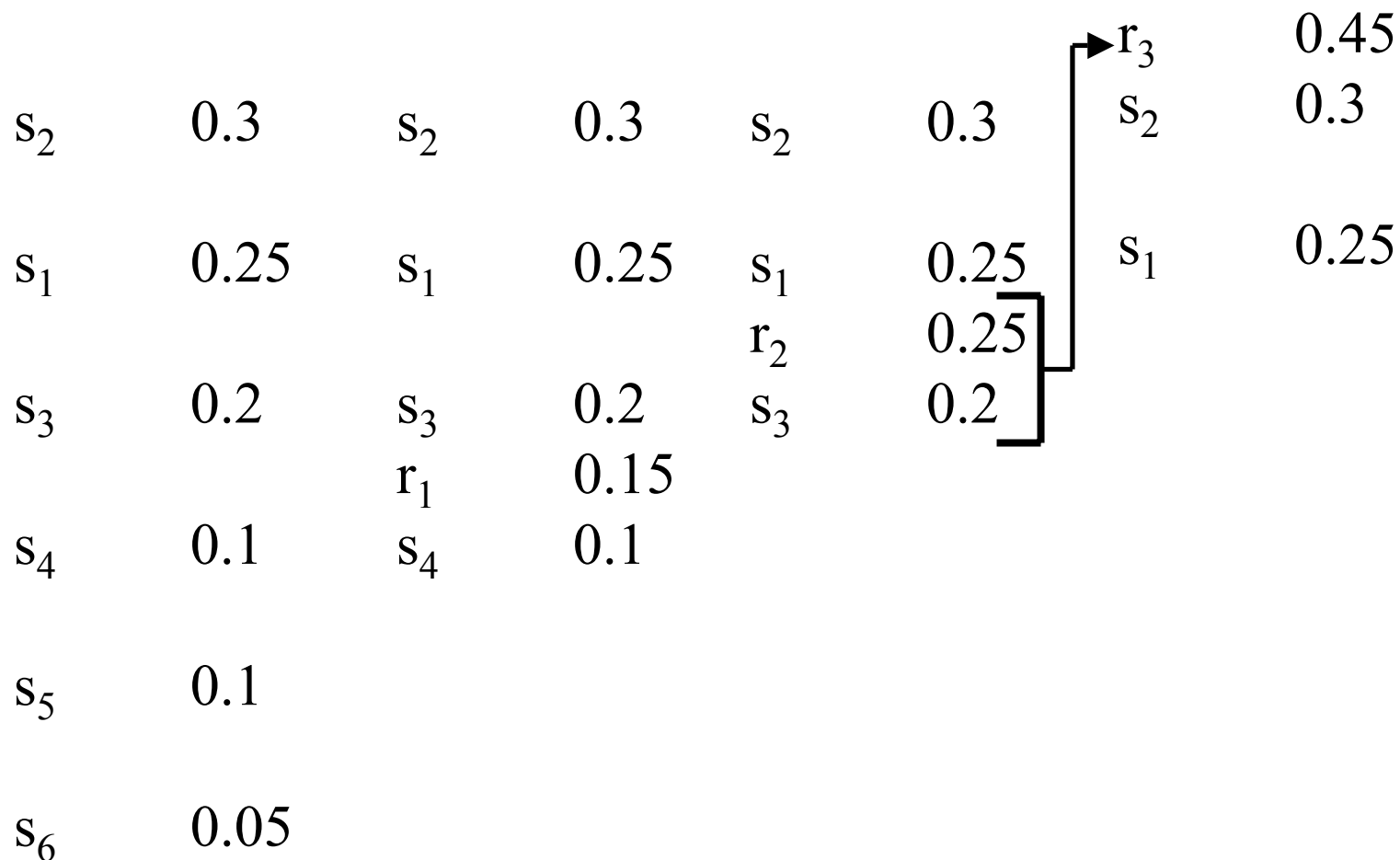
Codarea Huffman

s_2	0.3	s_2	0.3	s_2	0.3
s_1	0.25	s_1	0.25	s_1	0.25
s_3	0.2	s_3	0.2	r_2	0.25
s_4	0.1	r_1	0.15	s_3	0.2
s_5	0.1	s_4	0.1		
s_6	0.05				

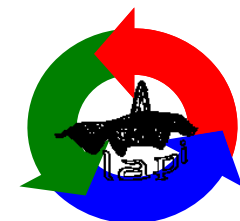
C. VERTAN



Codarea Huffman



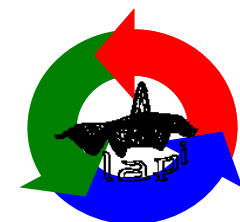
C. VERTAN



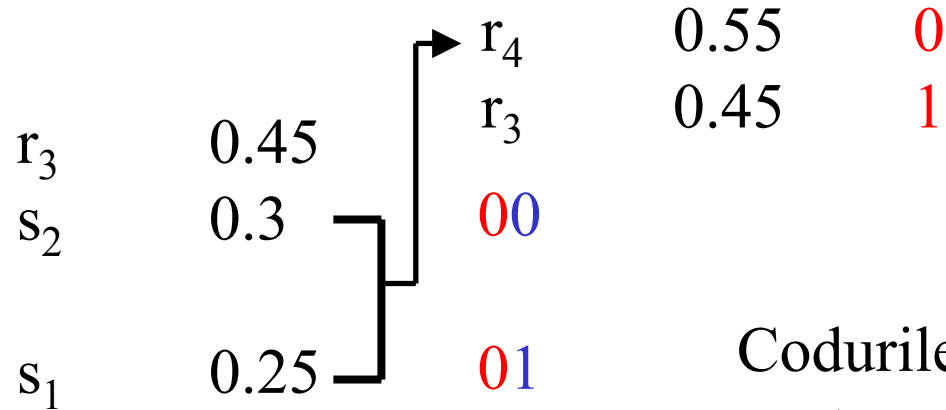
Codarea Huffman

							r_4	0.55
							r_3	0.45
					r_3	0.45	}	
s_2	0.3	s_2	0.3	s_2	0.3	s_2	0.3	}
s_1	0.25	s_1	0.25	s_1	0.25	s_1	0.25	
				r_2	0.25			
s_3	0.2	s_3	0.2	s_3	0.2			
		r_1	0.15					
s_4	0.1	s_4	0.1					
s_5	0.1							
s_6	0.05							

C. VERTAN



Codarea Huffman

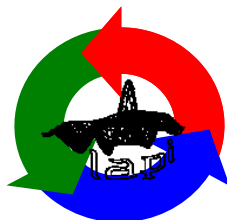


Incepand de la ultima sursa restransa se alocata cate un bit de 0 si 1 celor doua simboluri.

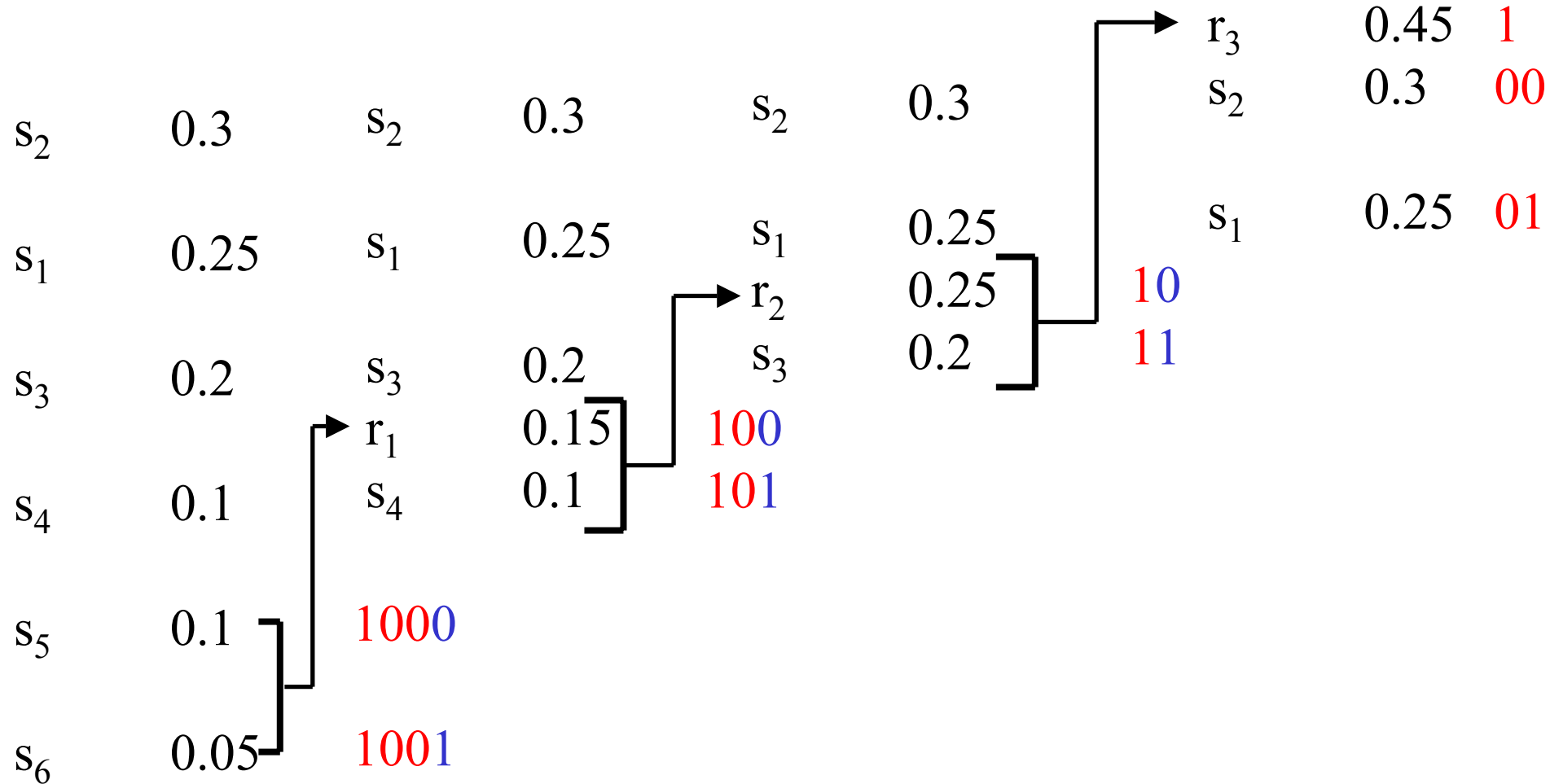
Codurile simbolurilor reunite sunt prefix pentru codurilor simbolurilor ce le compun.

Se repeta alocarea de biti pana cand s-au codat toate simbolurile sursei initiale.

C. VERTAN

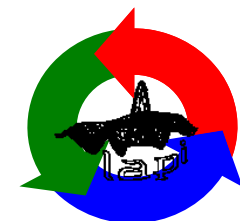


Codarea Huffman



C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



Tabel de codare

Codarea Huffman

Simb.	Prob.	Cuvant cod	Lung. cod	Lung. medie a cuvintelor de cod :
s_2	0.3	00	2	$2 * 0.3 +$
s_1	0.25	01	2	$2 * 0.25 +$
s_3	0.2	11	2	$2 * 0.2 +$
s_4	0.1	101	3	$3 * 0.1 +$
s_5	0.1	1000	4	$4 * 0.1 +$
s_6	0.05	1001	4	$4 * 0.05 = 2.4 \text{ biti}$

C. VERTAN



Codarea Huffman

Tabelul de codare trebuie transmis impreuna cu sirul de cuvinte de cod (sau tabelul de codare trebuie pre-stabilit).

Codarea este eficienta pentru sursa (deci setul de probabilitati) pentru care s-a generat tabelul de codare.

Codarea nu se poate face on-line (sunt necesare toate simbolurile emise de sursa pentru a putea calcula probabilitatile acestora).

Codarea Huffman este folosita in [aproape] toate standardele de codare a imaginilor.

C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



Codare Ziv-Lempel

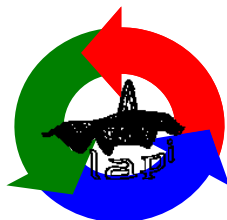
dictionarul de codare sa nu trebuiasca transmis (dictionarul se va genera la receptie)

codarea sa fie on-line (pe masura parcurgerii datelor)

1. initializare dictionar cu cuvintele de lungime 1 (0, 1)
2. se inspecteaza sirul de codat, cautand cel mai lung cuvant care apare deja in dictionarul de codare
3. la iesire se trimite indicele de dictionar al acestui cuvant
4. se adauga cuvantului gasit urmatorul simbol din sirul de intrare; acest nou cuvant este adaugat dictionarului.
5. se repeta de la 2, incepand cu ultimul simbol al cuvantului nou adaugat dictionarului

C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



Codare Ziv-Lempel

sir de codat (“intrare”)

0 1 1 0 1 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0

dicționar
initializat.

sir codat (“iesire”)

Dicționar	
Index (poziție)	Cuvânt
0	0
1	1
2	
3	
4	
5	
6	
7	
....	

Codare Ziv-Lempel

sir de codat (“intrare”)

0 1 1 0 1 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0

cel mai lung
cuvant existent
in dictionar

bitul urmator cu
care se formeaza
noul cuvant ce se inscrie
in dictionar

sir codat (“iesire”)

0

indexul cuvantului cel mai lung

Dictionar	
Index (pozitie)	Cuvant
0	0
1	1
2	
3	
4	
5	
6	
7	
....	

Codare Ziv-Lempel

sir de codat (“intrare”)

0 1 1 0 1 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0

bitul urmator cu
care se formeaza
noul cuvant ce se inscrie
in dictionar

Dictionar	
Index (pozitie)	Cuvant
0	0
1	1
2	01
3	
4	
5	
6	
7	
....	

sir codat (“iesire”)

0

Codare Ziv-Lempel

sir de codat (“intrare”)

0 1 1 0 1 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0

cel mai lung
cuvant existent
in dictionar

bitul urmator cu
care se formeaza
noul cuvant ce se inscrie
in dictionar

sir codat (“iesire”)

0 1

indexul cuvantului cel mai lung

Dictionar	
Index (pozitie)	Cuvant
0	0
1	1
2	01
3	
4	
5	
6	
7	
....	

Codare Ziv-Lempel

sir de codat (“intrare”)

0 1 1 0 1 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0

bitul urmator cu
care se formeaza
noul cuvant ce se inscrie
in dictionar

Dictionar	
Index (pozitie)	Cuvant
0	0
1	1
2	01
3	11
4	
5	
6	
7	
....	

sir codat (“iesire”)

0 1

Codare Ziv-Lempel

sir de codat (“intrare”)

0 1 **1 0** 1 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0

cel mai lung
cuvant existent
in dictionar

bitul urmator cu
care se formeaza
noul cuvant ce se inscrie
in dictionar

sir codat (“iesire”)

0 1 1

indexul cuvantului cel mai lung

Dictionar	
Index (pozitie)	Cuvant
0	0
1	1
2	01
3	11
4	
5	
6	
7	
....	

Codare Ziv-Lempel

sir de codat (“intrare”)

0 1 **1 0** 1 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0

bitul urmator cu
care se formeaza
noul cuvant ce se inscrie
in dictionar

Dictionar	
Index (pozitie)	Cuvant
0	0
1	1
2	01
3	11
4	10
5	
6	
7	
....	

sir codat (“iesire”)

0 1 1

Codare Ziv-Lempel

sir de codat (“intrare”)

0 1 1 **0 1** 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0

cel mai lung
cuvant existent
in dictionar

bitul urmator cu
care se formeaza
noul cuvant ce se inscrie
in dictionar

sir codat (“iesire”)

0 1 1 2

indexul cuvantului cel mai lung

Dictionar	
Index (pozitie)	Cuvant
0	0
1	1
2	01
3	11
4	
5	
6	
7	
....	

Codare Ziv-Lempel

sir de codat (“intrare”)

0 1 1 **0 1 1** 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0

bitul urmator cu
care se formeaza
noul cuvant ce se inscrie
in dictionar

Dictionar	
Index (pozitie)	Cuvant
0	0
1	1
2	01
3	11
4	011
5	
6	
7	
....	

sir codat (“iesire”)

0 1 1 2

Codare Ziv-Lempel

sir de codat (“intrare”)

0 1 1 0 1 **1 1** **0** 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0

cel mai lung
cuvant existent
in dictionar

bitul urmator cu
care se formeaza
noul cuvant ce se inscrie
in dictionar

sir codat (“iesire”)

0 1 1 2 3

indexul cuvantului cel mai lung

Dictionar	
Index (pozitie)	Cuvant
0	0
1	1
2	01
3	11
4	011
5	
6	
7	
....	

Codare Ziv-Lempel

sir de codat (“intrare”)

0 1 1 0 1 **1 1 0** 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0

bitul urmator cu
care se formeaza
noul cuvant ce se inscrie
in dictionar

Dictionar	
Index (pozitie)	Cuvant
0	0
1	1
2	01
3	11
4	011
5	110
6	
7	
....	

sir codat (“iesire”)

0 1 1 2 3

Si asa mai departe..... pana cand:

- se termina sirul de codat
- se “umple” dictionarul

Daca dictionarul s-a umplut, sunt posibile variantele:

- continuarea codarii cu dictionarul construit
- resetare dictionar si continuare cu un dictionar initializat

C. VERTAN

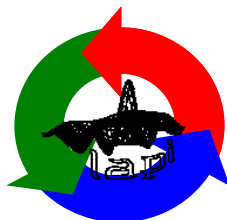
LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



Decodare Ziv-Lempel

1. initializare dictionar cu cuvintele de lungime 1 (0, 1)
2. se extrage un simbol din sirul codat; se scrie in sirul de iesire (sirul decodat) cuvantul care se afla in dictionar la adresa (indexul) dat de simbol; acest cuvnt se scrie incepand cu ultima pozitie din sirul de date deja decodat;
3. se adauga in dictionar o noua intrare ce corespunde cuvntului scris in sirul decodat la care se adauga un bit (momentan necunoscut)
4. se expliciteaza in dictionar ultimul bit al cuvntului de cod adaugat la etapa precedenta (este primul bit din noul cuvnt de cod)
5. se repeta de la 2, incepand cu ultimul simbol al cuvntului nou adaugat dictionarului

C. VERTAN



Run Length Encoding

Pentru un sir binar se codeaza numarul simbolurilor succesive de acelasi fel.

Ex. : sirul 0 0 0 1 1 1 1 0 0 1 0 1 0 0 0 se va coda ca :

0 3 4 2 1 1 1 3

Valoarea
primului
simbol din
sir

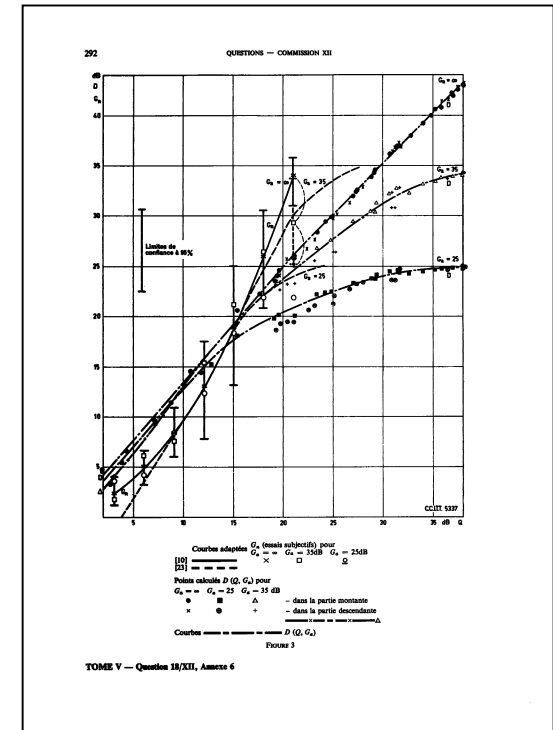
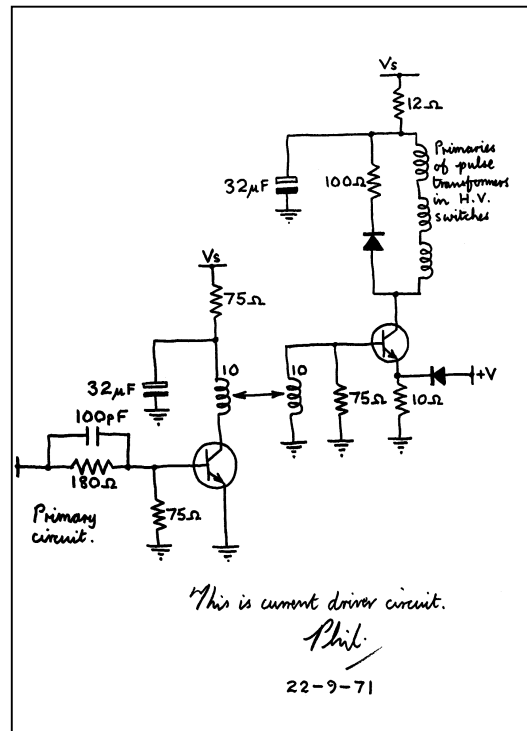
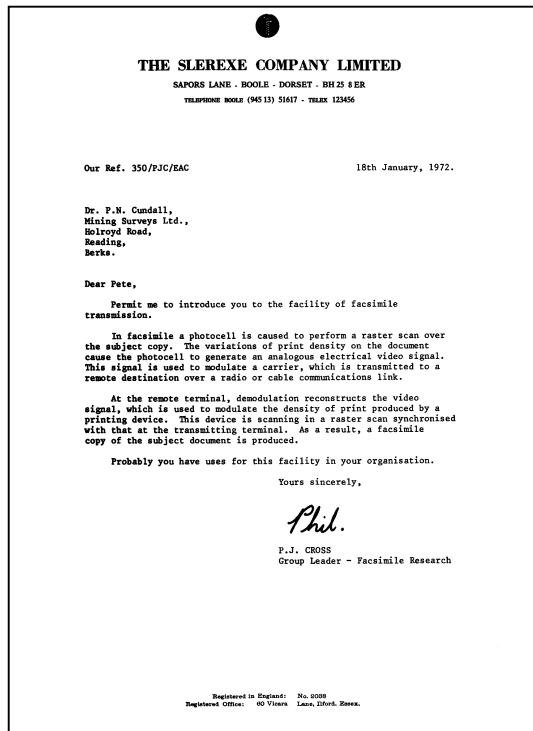
Lungimile blocurilor de simboluri
succesive de acelasi fel.

C. VERTAN



[illegible]

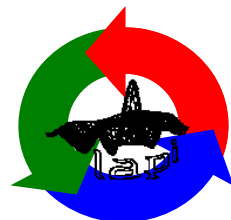
Codarea RLE fax



Imagini de test CCITT (alb/negru)
(total 8 imagini: text, scheme, tabele)

C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



Transmisia fax: (CCITT)

RLE + codare Huffman a lungimilor (*runs*)

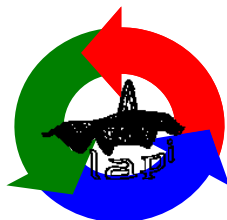
Tabele de codare Huffman sunt standardizate (sunt deci pre-stabilite) si transmisia lor nu este necesara.

1782 pixeli/ linie (aprox. 200 dpi); 7.7 linii/ mm

Limitare a lungimii maxim codate la 63. Daca un segment are o lungime de cel mult 63, este codat ca atare, si codul se numeste *terminator*. Daca un segment are mai mult de 63 pixeli, se codeaza subsegmentele de lungime multiplu de 64 (64, 128,..., 1728, *EOL mark-up code*) si restul (*terminator code*).

Se codeaza separat lungimile de alb si negru.

C. VERTAN



White Block Skipping

Sirul binar de codat este impartit in grupuri (blocuri) de cate B biti.

Un bloc alb (toti bitii ce il compun de valoare 0) este codat cu un unic bit de 0.

Un bloc ne-alb (nu are toti bitii 0) este prefixat cu un bit de 1 si copiat.

Ex. : sirul $000|111|100|101|000$ se va coda cu $B=3$ ca :

$0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0$



Cand este codarea eficienta ?

Fie un sir ce are n_0 blocuri albe dintr-un total de n blocuri.

Fiecare bloc alb este inlocuit cu 1 bit

Fiecare bloc ne-alb este inlocuit de $B+1$ biti

Lungime initiala sir : nB

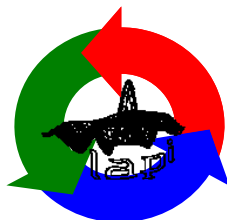
Lungime sir comprimat : $n_0 + (n-n_0)(B+1)$

Compresie : $n_0 + (n-n_0)(B+1) < nB$

$$n < n_0 B$$

$$n_0 / n > 1/B$$

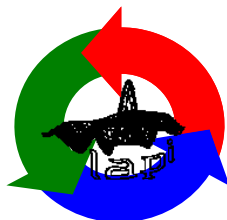
C. VERTAN



TEHNICI DE COMPRESIE A IMAGINILOR (cont.)

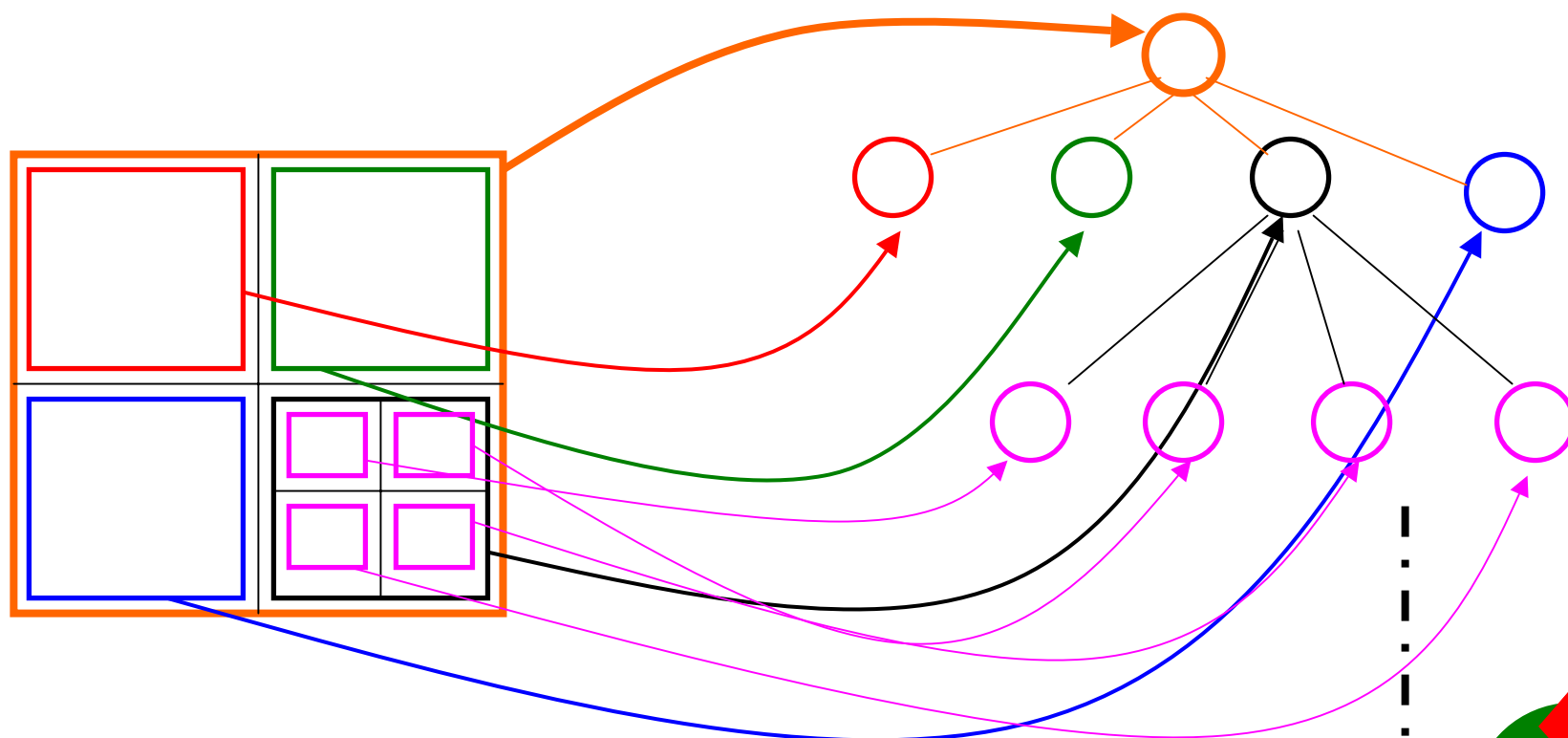
C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR

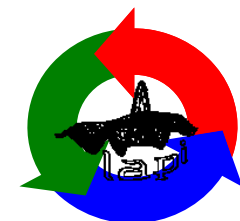


Compresia cu arbori cuaternari (quadtrees)

Ideea de baza : divizarea regulata a suportului spatial al imaginii, daca acesta are valori uniforme, si asocierea de noduri dintr-un arbore.



C. VERTAN



Probleme :

cand se construiesc descendentii ai unui nod ?
cum se stabileste legatura dintre pozitia nodurilor si cea a regiunilor corespunzatoare ?

Nodurile devin ne-terminale (cu descendentii) daca zonele din imagine care le corespund nu sunt suficient de uniforme (“suficient” se traduce prin respectarea unei conditii de uniformitate la nivelul regiunii corespunzatoare.

Deci : regiunile neuniforme sunt “taiate” in patru parti egale (sferturi).

Regula de alocare a descendentilor este indiferenta, atata vreme cat este aceeaasi pentru constructia intregului arbore.

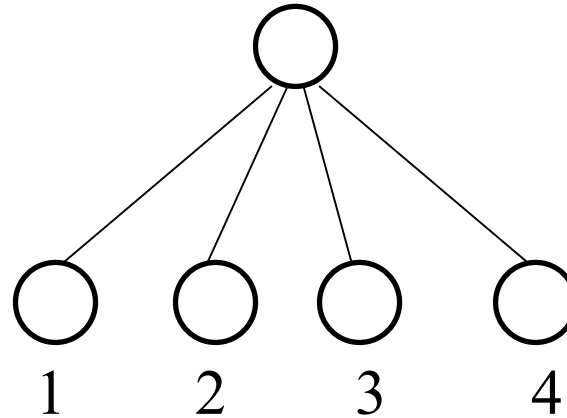
C. VERTAN



Ex. reguli de alocare a descendentilor

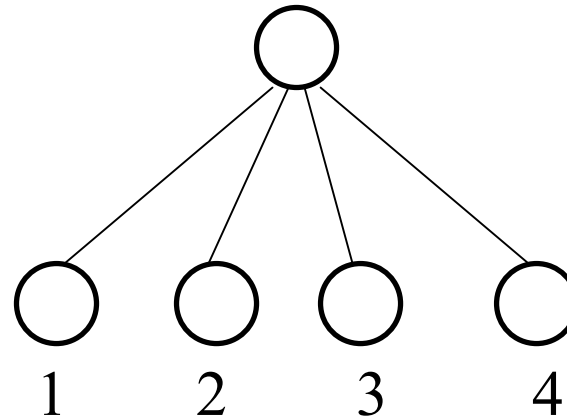
1	2
4	3

sens orar



1	2
3	4

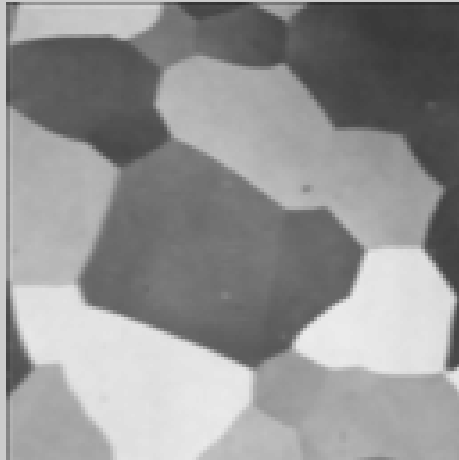
curba Z



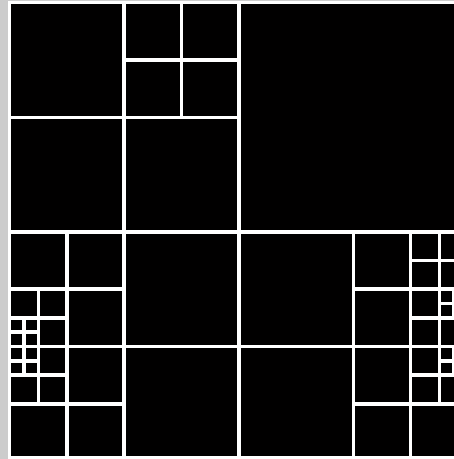
Quadtree: exemplu

Select an image:

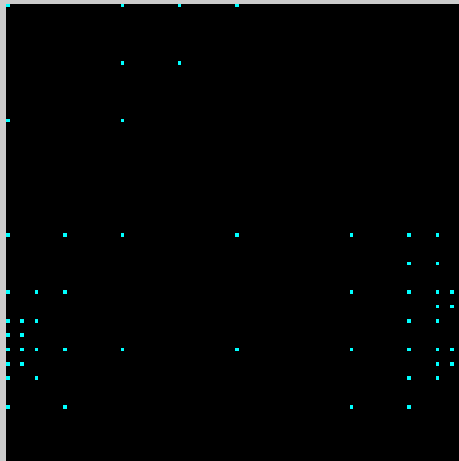
Aluminum ▼



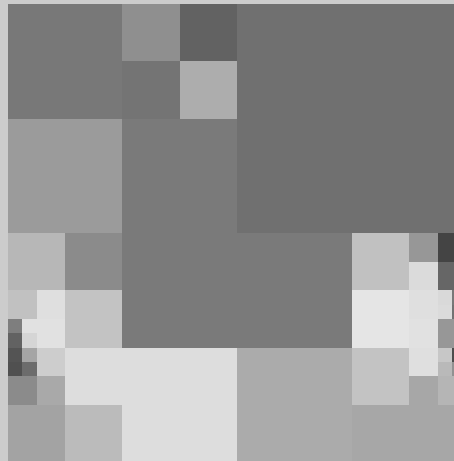
Quadtree decomposition



Sparse representation



Block means



Number of blocks: 55

Threshold (0-1): 0.60



Apply

☐ Animated computation

Info

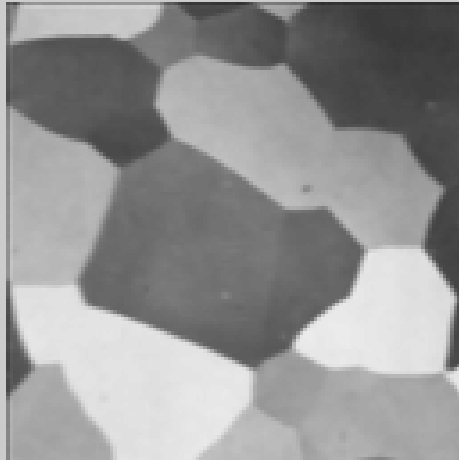
Close



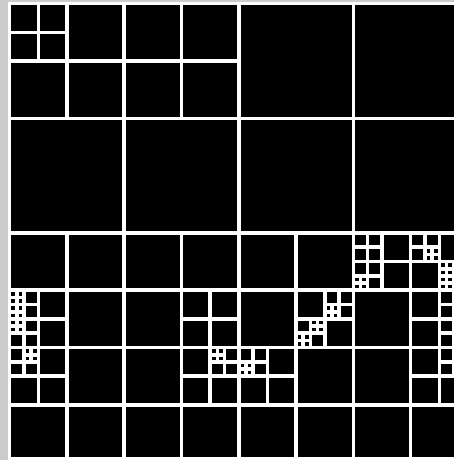
Quadtree: exemplu

Select an image:

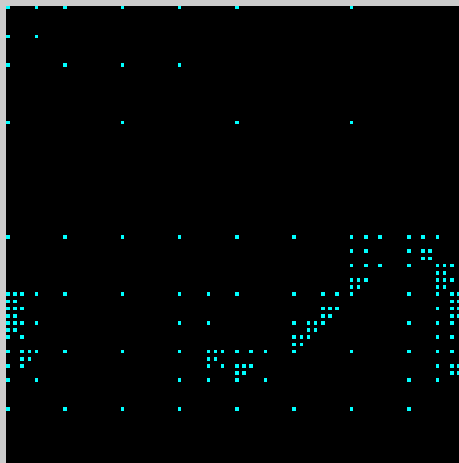
Aluminum ▼



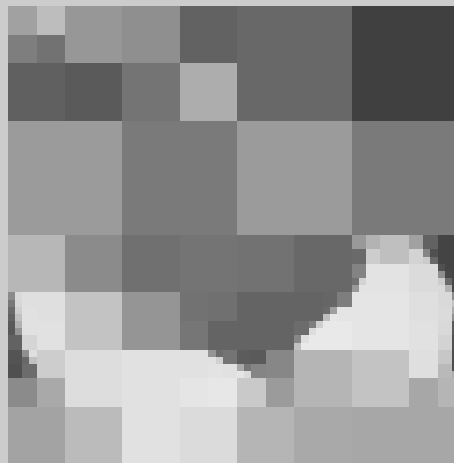
Quadtree decomposition



Sparse representation



Block means



Number of blocks: 169

Threshold (0-1): 0.50

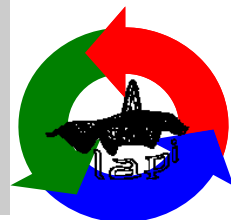


Apply

☐ Animated computation

Info

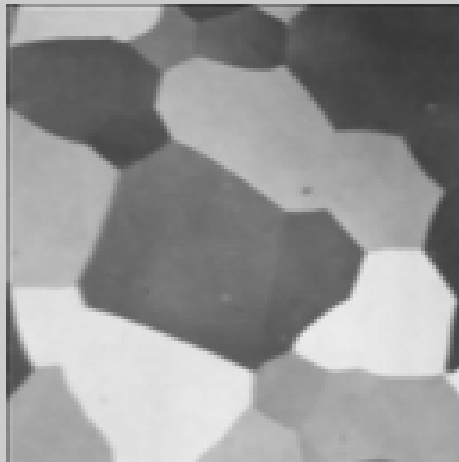
Close



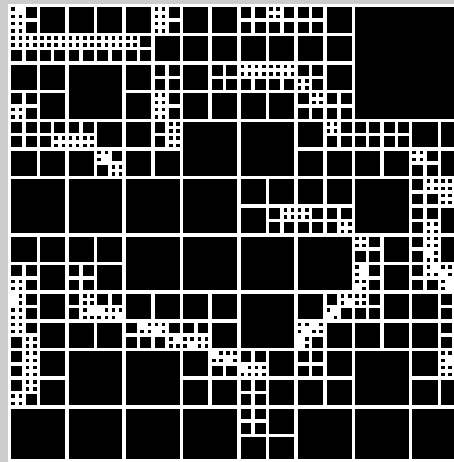
Quadtree: exemplu

Select an image:

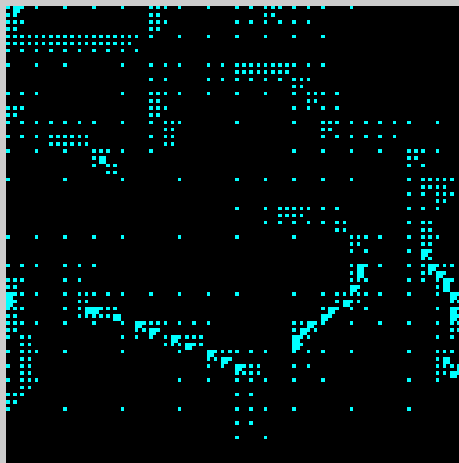
Aluminum ▼



Quadtree decomposition



Sparse representation



Block means



Number of blocks: 703

Threshold (0-1): 0.30

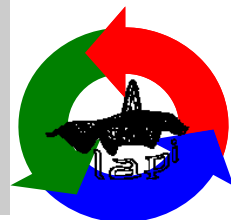


Apply

☐ Animated computation

Info

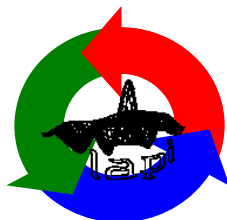
Close



Algoritm :

1. Regiunea curenta = intreaga imagine
aceasta este radacina arborelui.
2. Verifica uniformitatea regiunii curente:
daca regiunea este uniforma, nodul din arbore este terminal
daca regiunea nu este uniforma, regiunea este taiata in patru
si nodul curent capata patru descendenti.
3. Fiecare noua regiune devien regiune curenta.
4. Repeta de la 2 pana cand :
regiunile au o dimensiune minima prestabilita (la limita 1 pixel)
regiunile sunt uniforme.

C. VERTAN



Algoritmul, in descrierea sa anterioara, este o varianta bazata pe descompunerea regiunilor (split); arborele se construiește incepand cu radacina, si deci algoritmul este de tip

top - down.

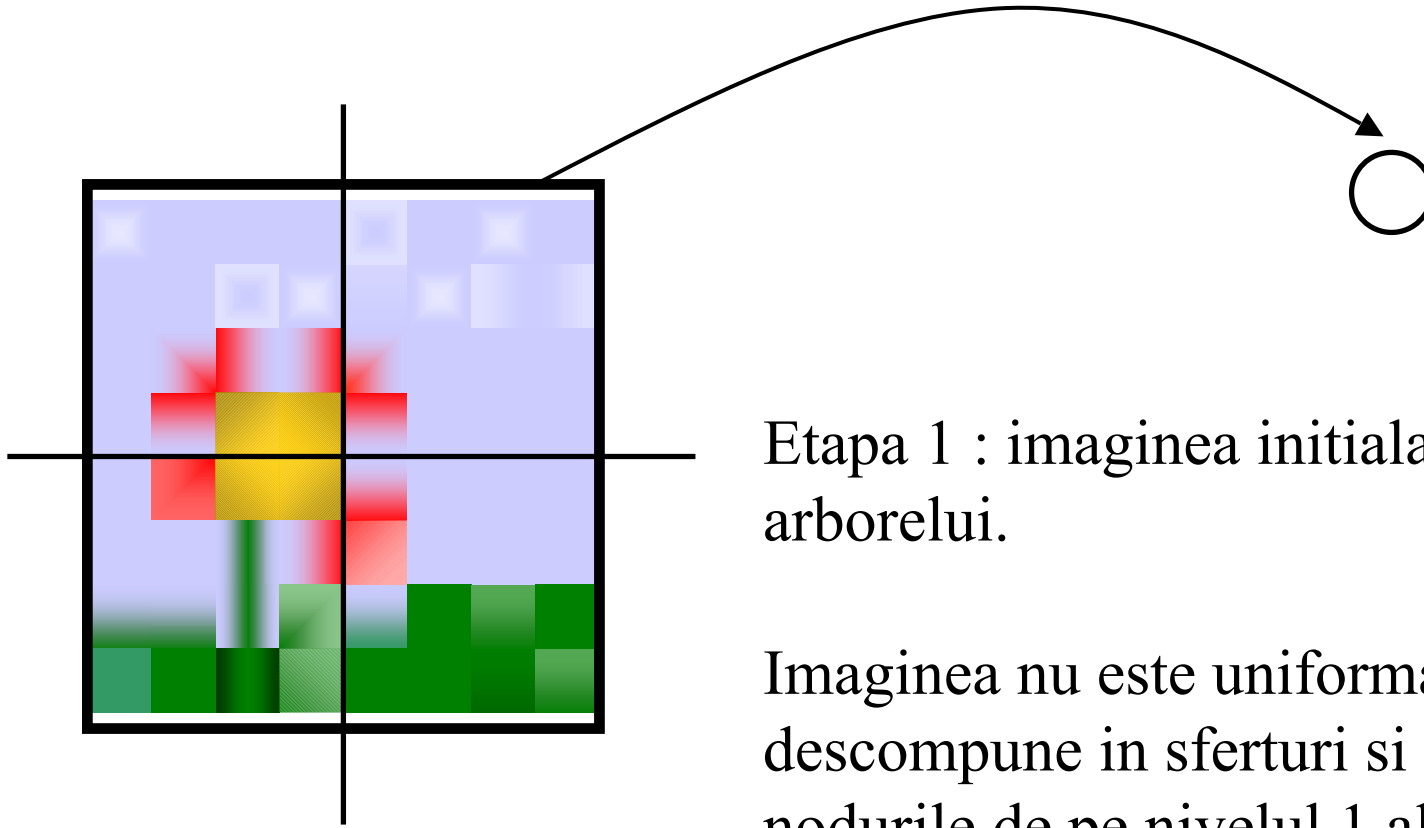
Complexitatea este mare, pentru ca fiecare pixel va fi parcurs de un numar de ori egal cu adancimea sa in arbore.

Varianta eficienta a algoritmului va necesita citirea o singura data a fiecarui pixel si agregarea regiunilor uniforme; arborele se construiește incepand cu frunzele, si deci algoritmul devine

bottom - up.



Quadtree color



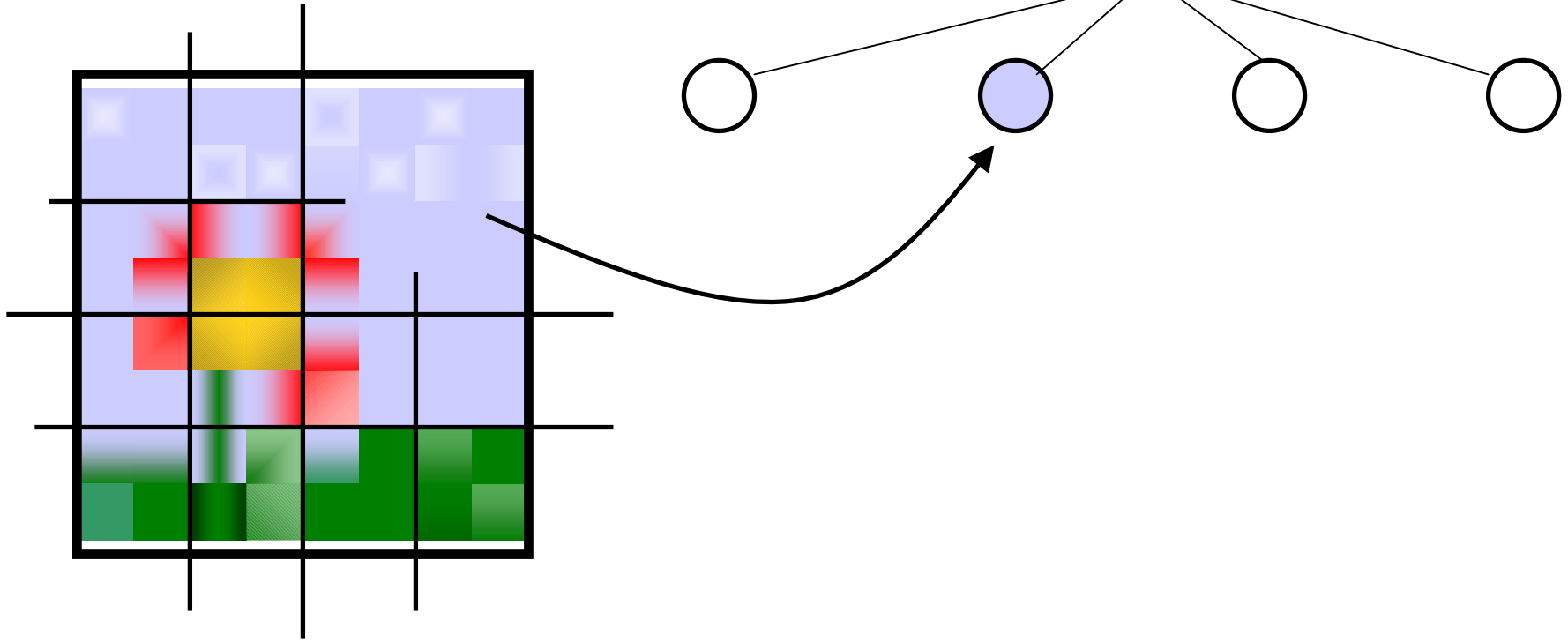
Etapa 1 : imaginea initiala este radacina arborelui.

Imaginea nu este uniforma, deci se descompune in sferturi si se construiesc nodurile de pe nivelul 1 al arborelui.

C. VERTAN



Quadtree color

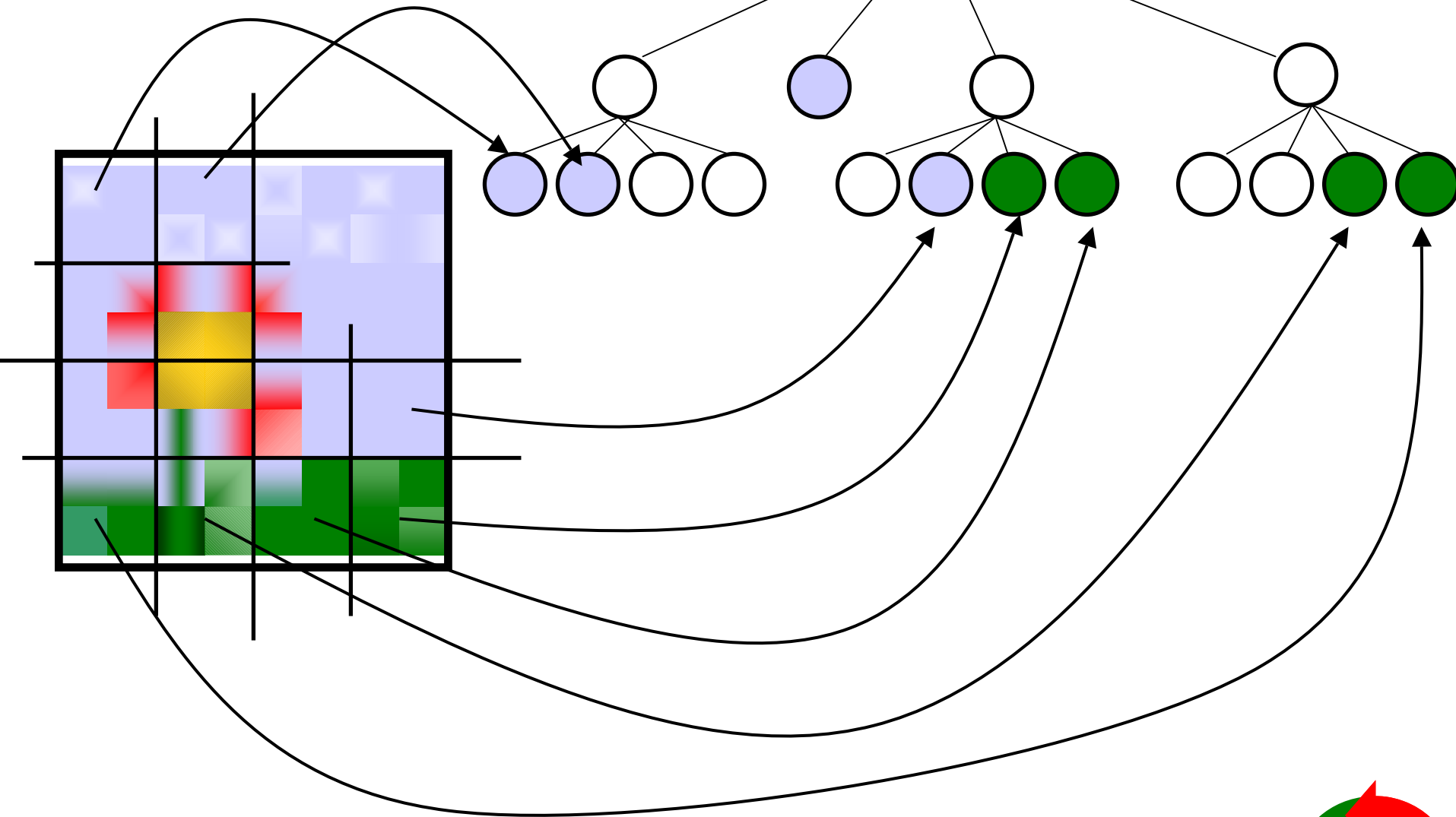


Numai sfertul 2 este suficient de uniform; nodul corespunzator devine terminal si primeste “culoarea medie” din regiune. Celelalte regiuni vor trebui descompuse.

C. VERTAN



Quadtree color

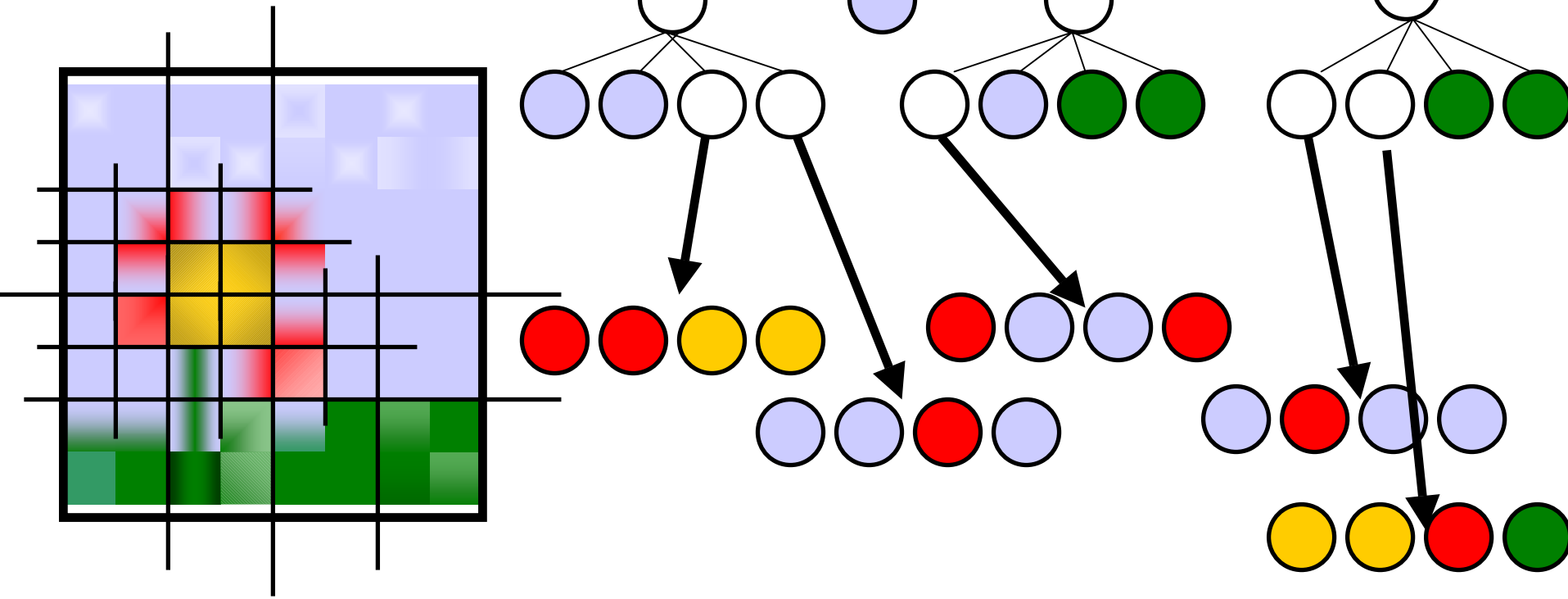


C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



Quadtree color



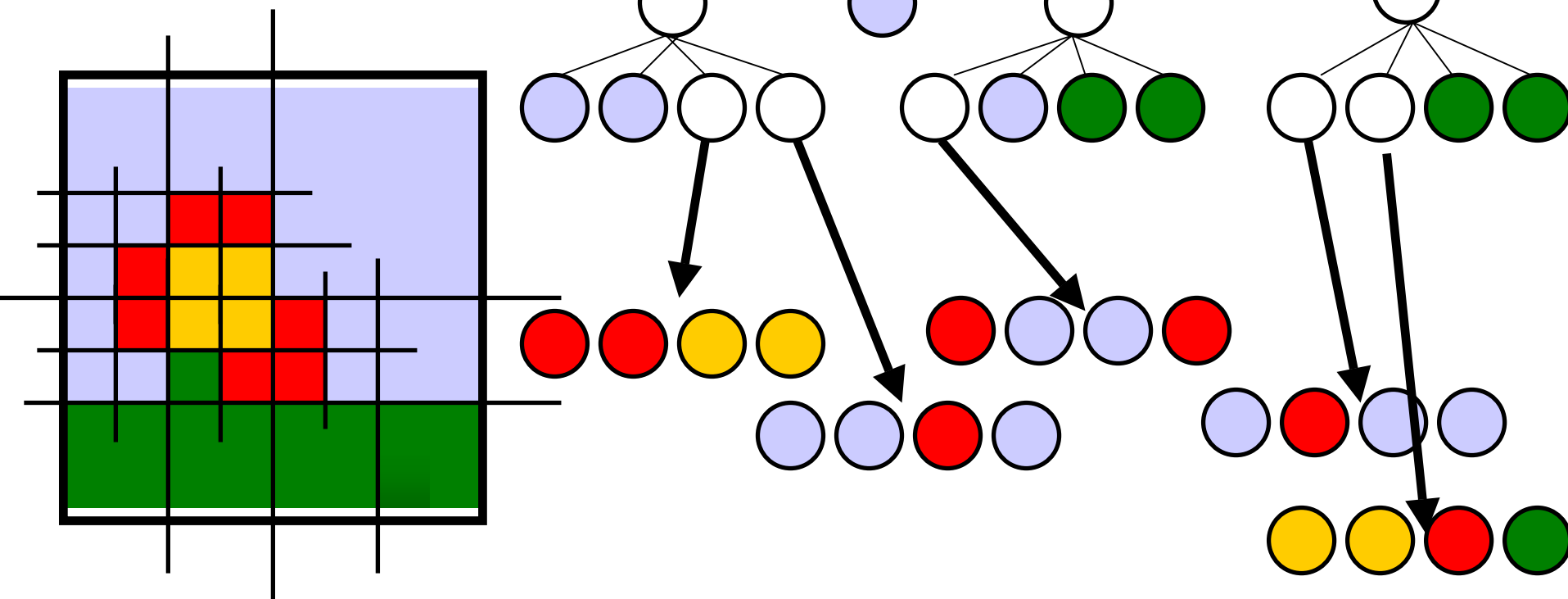
Compresia este data de cantitatea de informatie necesara pentru a memora pozitia in arbore a nodurilor terminale.

C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



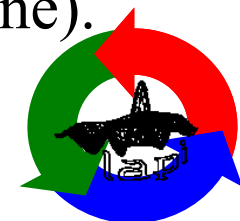
Reconstructia



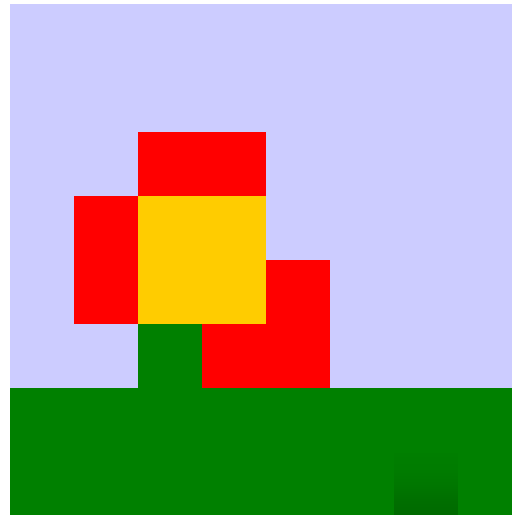
Pozitia nodurilor terminale in arbore (adancime, cale de la radacina) stabileste reconstructia (ce dimensiune are blocul, unde se pune).

C. VERTAN

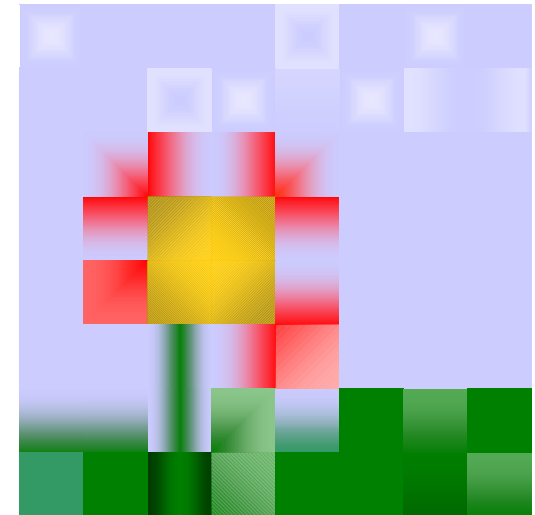
LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



Erorile (si gradul de compresie) sunt determinate de pragul de uniformitate acceptata pentru regiuni.

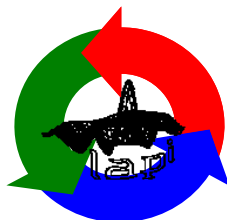


Reconstructie



Original

C. VERTAN



TEHNICI DE COMPRESIE A IMAGINILOR (cont.)

Cuantizarea vectoriala (vector quantization) (VQ)

C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



Compresia prin cuantizare vectoriala (VQ)

Tip de compresie bazata pe “dictionar de cuvinte de cod” : portiuni din imagine (blocuri) sunt inlocuite cu pozitia la care se gaseste in dictionar cuvantul de cod (blocul standard de codare) cel mai similar.

Blocurile sunt de forma regulata si impart imaginea fara suprapuneri (dimensiuni tipice 4×4 , 8×8 , 16×16).

Compresia este cu atat mai puternica cu cat blocurile sunt de dimensiune mai mare.

Compresia este cu atat mai precisa cu cat dictionarul este mai mare.

C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR



Proiectarea VQ dupa rata de compresie

Blocurile de cod din dictionar sunt de dimensiune $m \times n$

Dictionarul are N blocuri de cod; indexul e reprezentat pe $\log_2 N$ biti

Valoarea oricarui pixel este reprezentata pe P biti.

Fiecare grup de $m \times n$ pixeli din imagine (care este reprezentat initial pe $m \times n \times P$ biti) este reprezentat dupa codare de indexul unui bloc de cod, deci de $\log_2 N$ biti.

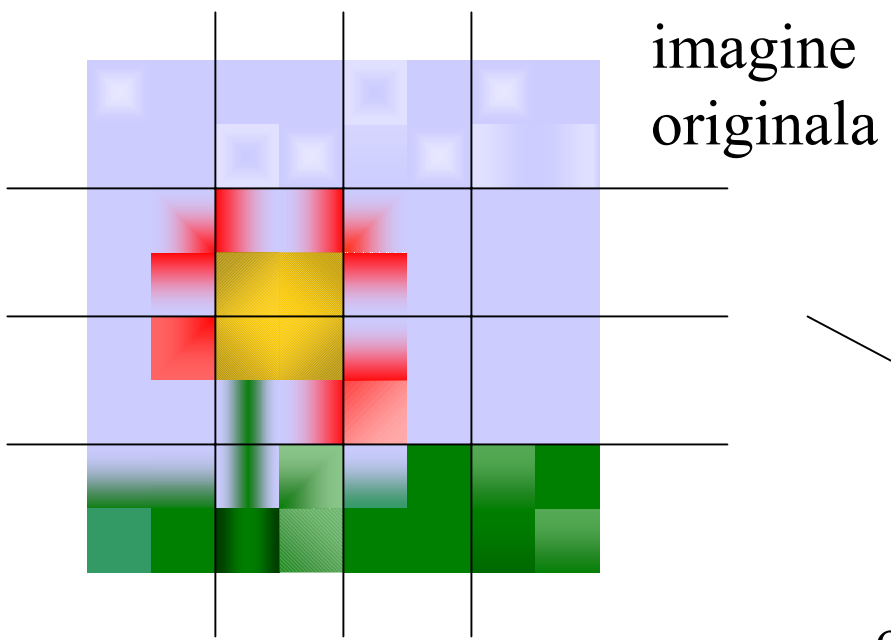
Compresia va fi data de :

$$C = \frac{mnP}{\log_2 N}$$

! Raportul anterior presupune cunoasterea dictionarului; alfel, dictionarul trebuie transmis (memorat) si el impreuna cu codul imaginii.

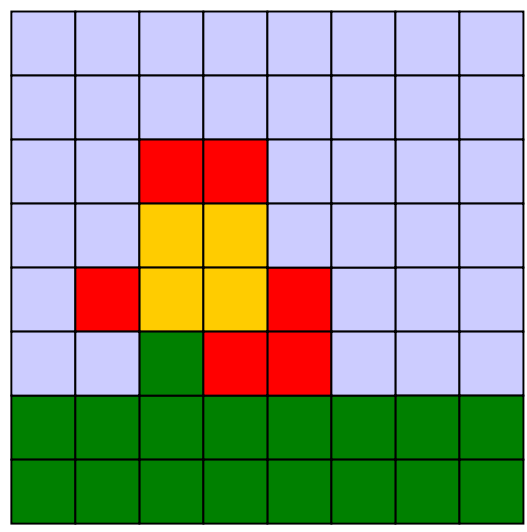
C. VERTAN





cod

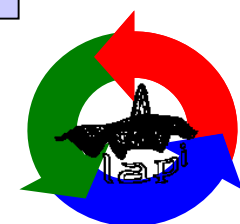
1	1	1	1
1	6	1	1
7	5	4	1
2	2	2	2



dictionar

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

C. VERTAN



Calitatea compresiei este data de dictionarul folosit.

Ideea este de a sintetiza dictionarul pe un set suficient de mare de imagini “tipice”, astfel ca blocurile de cod sa reprezinte bine caracteristicile imaginii.

Sinteza dictionarului este o instanta a unui algoritm de **clustering** : din blocurile-exemplu extrase din imaginile setului de antrenament sa se extraga blocuri caracteristice (prototipuri).

Cazul cel mai defavorabil pentru compresia cu VQ : folosirea unui dictionar dedus pentru imagini cu caracteristici total diferite.



Punerea problemei :

un set de N puncte, descrise de vectori de dimensiune p
trebuie impartit in C clase (grupuri, clustere).

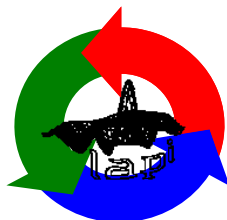
$$X = \{ \mathbf{x}_i \}, i = 1, 2, \dots, N$$

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})$$

Impartirea (partitionarea) setului de puncte in clase :

indice de apartenenta a fiecarui punct (carei clase ii apartine)

Exprimarea cantitativa a conceptului de “partitionare buna”.
criterii de calitate a partitiei.



Apartenența punctelor la clase

Apartenența punctului \mathbf{x}_i la clasa j :

$$u_{ij}, \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, C$$

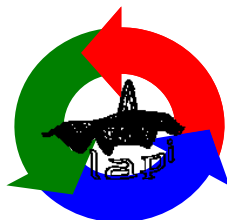
Modele de clustering :

Net (binar) :

Nuantat (fuzzy) :

$$u_{ij} = \begin{cases} 1, & \mathbf{x}_i \in \text{Clasa}_j \\ 0, & \mathbf{x}_i \notin \text{Clasa}_j \end{cases}$$

$$u_{ij} \in [0, 1] \\ \forall \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, C$$



Masuri de calitate a claselor

clase compacte : centrul clasei este “aproape” de toate punctele clasei (punctele clasei sunt bine approximate de centrul clasei).

clasa are suficient de multe puncte

clase bine separate : distantele dintre centrele claselor sa fie cat mai mari.

Cele doua cerinte sunt adeseori contradictorii.



Basic ISODATA (k-means, C-means)

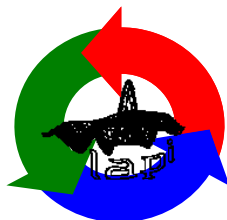
ISODATA = **I**terative **S**elf **O**rganizing **D**ata **A**nalysis **T**echnique

Se fixeaza numarul de clase dorit, C .

Calitatea partitiei (a claselor) e caracterizata de eroarea globala de aproximare a vectorilor de date prin prototipurile claselor.

$$J = J(u_{ij}, \mu_j) = \sum_{j=1}^C \varepsilon_j = \sum_{j=1}^C \left(\sum_{i=1}^N u_{ij} \|\mathbf{x}_i - \mu_j\|^2 \right)$$

μ_j prototipul (centroidul) clasei j



Basic ISODATA (C-means)

$$\frac{\partial J}{\partial \boldsymbol{\mu}_j} = 2 \sum_{i=1}^N u_{ij} (\mathbf{x}_i - \boldsymbol{\mu}_j) = 0$$

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^N u_{ij} \mathbf{x}_i}{\sum_{i=1}^N u_{ij}}$$

prototipurile claselor sunt mediile aritmetice ale vectorilor de date ce apartin claselor.

$$u_{ij} = \begin{cases} 1, & \|\mathbf{x}_i - \boldsymbol{\mu}_j\| \leq \|\mathbf{x}_i - \boldsymbol{\mu}_k\|, \forall k \neq j \\ 0, & \text{rest} \end{cases}$$

orice vector apartine clasei de al carei prototip este cel mai apropiat.



Basic ISODATA (C-means)

1. alege un set aleator de prototipuri
2. calculeaza apartenenta fiecarui vector la una dintre clasele partitiei (vectorii apartin clasei de al carei prototip sunt cei mai apropiati)
3. calculeaza prototipurile claselor ca media aritmetica a vectorilor apartind fiecarei clase
4. evalueaza criteriu de oprire :
 - eroare globala suficient de mica ?
 - numar de iteratii suficient de mare ?
 - au fost vectori care sa isi schimbe apartenenta ?
 - au fost prototipuri care s-au modificat semnificativ ?
5. repeta de la 2 daca e cazul.

C. VERTAN

LABORATORUL DE ANALIZA ȘI PRELUCRAREA IMAGINILOR

