



LAPI – Laboratorul de  
Analiza și Prelucrarea  
Imaginilor



Universitatea  
POLITEHNICA din  
București



Facultatea de Electronică,  
Telecomunicații și  
Tehnologia Informației

# Analiza și Prelucrarea Secvențelor de Imagini

Ș.I.dr.ing. Bogdan IONESCU

București, 2010

## Plan Curs

- Aplicații & bibliografie
- M1. Informația video
- M2. Filtrare spațio-temporală
- M3. Informația de culoare
- M4. Informația temporală
- M5. Informația de mișcare
- A1. Clasificarea automată după gen

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU
1

## > Aplicații & bibliografie

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU
2

### Cateva domenii de aplicație

- > **supraveghere video:**  
monitorizare trafic, securizare, urmărire persoane, ...
- > **automatizări industriale:**  
inspecție linie de producție, detecție defecte fabricație, ...
- > **televiziune:**  
compresie, transmisie, filtrare semnal video digital, HDTV, ...
- > **interfațare om-mașină:**  
realitate virtuală, control roboți, sisteme biometrice, ...
- > **indexare multimedia:**  
gestionarea bazelor multimedia, sisteme de căutare după conținut, ...
- > ...

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU
3

### Conferințe științifice din domeniu

- > **ACM International Conference on Multimedia**  
ACM MM, Association for Computing Machinery
- > **IEEE International Conference on Multimedia & Expo**  
IEEE ICME, Institute of Electrical and Electronics Engineers
- > **IEEE International Conference on Image Processing**  
IEEE ICIP, Institute of Electrical and Electronics Engineers
- > **IS&T/SPIE Electronic Imaging**  
SPIE EI, International Society Advancing an Interdisciplinary Approach to the Science and Application of Light
- > **IEEE International Conference on Acoustic, Speech and Signal Processing**  
IEEE ICASSP, Institute of Electrical and Electronics Engineers

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU
4

### Câteva materiale bibliografice de sinteză

- > Cees G.M. Snoek, M. Worring, *Multimodal Video Indexing: A Review of the State-of-the-art*, Multimedia Tools and Applications, 25(1), pag. 5–35, 2005.
- > A. F. Smeaton, P. Over, W. Kraaij, *High-Level Feature Detection from Video in TRECVID: A 5-Year Retrospective of Achievements*, Multimedia Content Analysis, Theory and Applications, pp. 151–174, Springer Verlag, Berlin 2009.
- > B. T. Truong, S. Venkatesh, *Video abstraction: a systematic review and classification*, ACM Transactions on Multimedia Computing, Communications, and Applications 3(1), 3 2007.
- > D. Brezeale, D.J. Cook, *Automatic Video Classification: A Survey of the Literature*, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 38(3), pp. 416-430, 2008.
- > Alan C. Bovik, *The Essential Guide to Video Processing*, Academic Press, ISBN: 978-0-12-374456-2, 2009.
- > B. Ionescu, *Analiza și Prelucrarea Secvențelor Video: Indexarea Automată după Conținut*, Ed. Tehnică București, 2009.

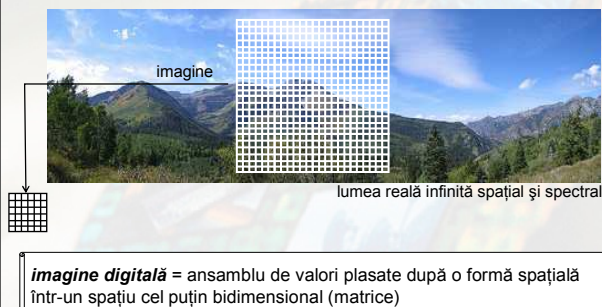
Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU
5

## > M1. Informația video

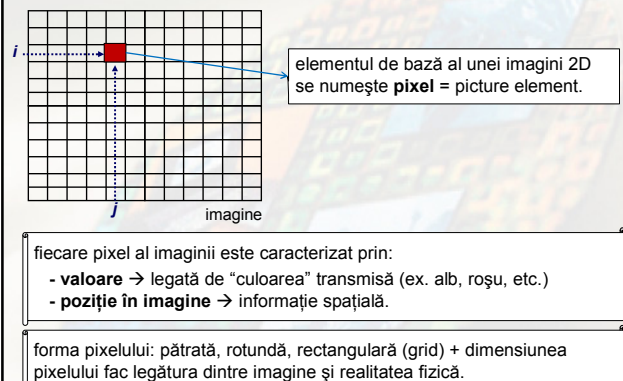
- 1.1. [ Noțiuni de bază ]
- 1.2. [ Compresia informației ]
- 1.3. [ Manipulare video – utilitare ]
- 1.4. [ Conceptul de indexare ]

## > M1. Informația video [ Noțiuni de bază ]

### Noțiuni de bază: ce reprezintă o imagine ?



### Noțiuni de bază: ce reprezintă o imagine ?



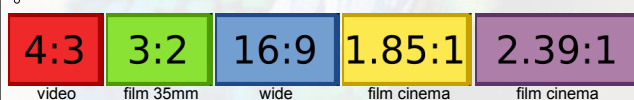
### Noțiuni de bază: rezoluție - aspect

[sursă imagini Wikipedia]

**rezoluția de pixeli** = numărul de pixeli pe orizontală X numărul de pixeli pe verticală (dimensiune imagine)  
→ rezoluție mare înseamnă o mai bună reprezentare a realității (mai multe detalii)



**rată de aspect** = lățimea imaginii în pixeli împărțită la înălțimea în pixeli.

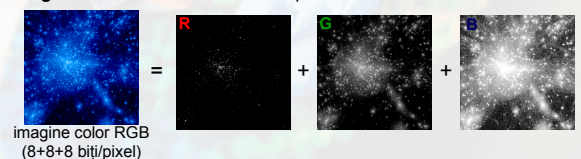


### Noțiuni de bază: cum sunt reprezentate imaginile digitale ?

↳ **imagini scalare** = valoarea oricărui pixel este un scalar proporțional cu intensitatea luminoasă

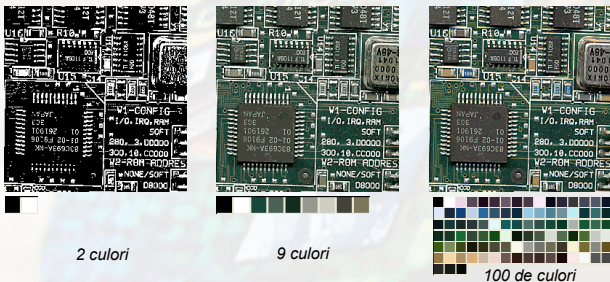


↳ **imagini vectoriale** = valoarea unui pixel este un vector



## Noțiuni de bază: paletă de culoare

**paletă de culoare** = mulțimea tuturor culorilor (intensităților) folosite de imagine.



## Noțiuni de bază: ce reprezintă o secvență de imagini ?

> video = o succesiune temporală de imagini ("imagini în mișcare")



**rezoluție temporală (frame rate)** = număr de imagini pe secundă;

→ timp real: PAL/SECAM 25 cadre/s sau NTSC 30 cadre/s,

→ "high speed": 50 - 200,000,000 cadre/s.



> video = volum de date important:

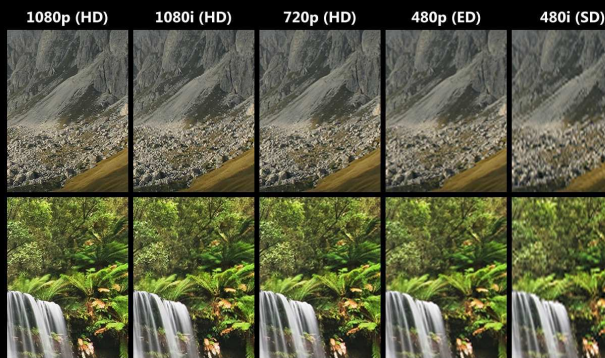
- 1 minut > 1.500 imagini statice;

- 1 minut =  $768 \times 576 \times 24 \times 25 \times 60 = 1.85$  GB (fără compresie)

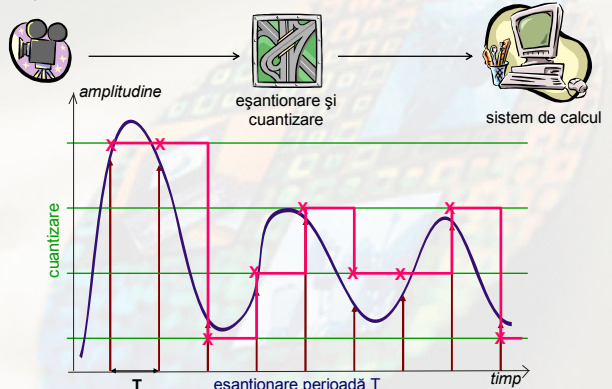
→ protocoale de transfer de mare viteză: CameraLink (255MB/s), FireWire (~400MB/s), GigE Vision (Lan Gigabit).

## Noțiuni de bază: SD vs. HD

[sursă imagini Wikipedia]



## Noțiuni de bază: eșantionare și cuantizare



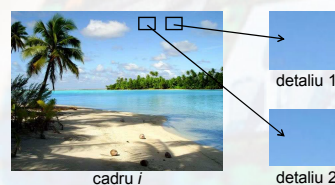
## > M1. Informația video [ Compresia informației ]

## Principiul compresiei datelor

**motivație simplă** = volum de date prea mare (pentru stocare și/sau transmisie) → SDTV ~ 20MB/s, HDTV ~ 120MB/s;

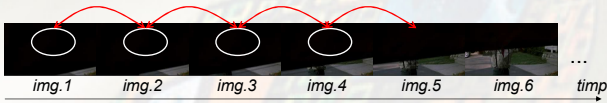
**principiu:** reducerea **redundanței** datelor (anumite informații se repetă sau pot fi deduse pe baza altora)

↳ **redundanță spațială** = valorile pixelilor vecini sunt corelate la nivel de cadru:

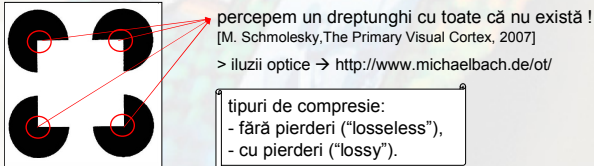


## Principiul compresiei datelor

→ **redundanță temporală** = valorile pixelilor din cadrul curent sunt corelate cu cele ale pixelilor din cadre vecine temporale:



→ **redundanță psihovizuală** = SVU este bazat pe aproximare, ochiul este puțin sensibil la detalii lângă contururi sau schimbări de cadru:



## Tehnici de codare video: JPEG

Joint Photographic Experts Group 1986-1992 – compresia imaginilor cu niveluri de gri și truecolor,  
→ primul pas major: **compresie în frecvență**.

> modul de codare:

1. RGB → YCbCr:

separarea luminozității (Y) de componentele cromatice (Cb și Cr) asigură o compresie mai bună cu pierderea minimă a calității percepționale,

(detalii M2 – Informația de Culoare)

2. subeșantionare spațială:

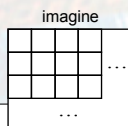
ochiul uman percepe mai multe detalii fine la nivel de intensitate (Y) decât în nuanță și saturație (Cb și Cr):  
→ downsampling Cb și Cr: 4:2:2 (1/2 pe oriz.) sau 4:2:0 (1/2 pe oriz. & vert.)

## Tehnici de codare video: JPEG

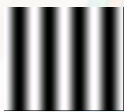
> modul de codare (continuare):

3. împărțire în blocuri:

fiecare componentă este împărțită în blocuri nesuprapuse de regulă de 8x8 pixeli (în funcție de subeșantionare);



4. DCT – Discrete Cosinus Transform: **motivație**



→ spațial (coordonate) observăm 5 obiecte luminoase pe un fond întunecat, asemănătoare unor bare verticale;

- anumite informații nu sunt disponibile direct în domeniul de definiție inițial al imaginii:



→ imaginea poate fi interpretată ca o oscilație cosinus (frecvențial),  
informație ≈ frecvență + amplitudine + DC (eventual)

## Tehnici de codare video: JPEG

> modul de codare (continuare):

4. DCT – Discrete Cosinus Transform: **transformare**

[A.K. Jain, Fundamentals of Digital Image Processing, 1989]

$$v(k, l) = \alpha(k) \alpha(l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u(m, n) \cdot \left\{ \cos \frac{\pi(2m+1)k}{2N} \cos \frac{\pi(2n+1)l}{2N} \right\}$$

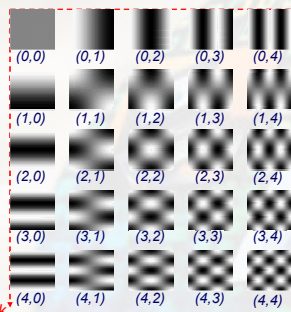
unde  $u(m, n)$  reprezintă imaginea de dimensiune  $N \times N$ ,  $k$  ~ frecvența verticală iar  $l$  ~ frecvența orizontală,  $v(k, l)$  este imaginea transformatei, iar  $\alpha(k)$  este o constantă dată de:

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} & k = 0 \\ \sqrt{\frac{2}{N}} & \text{altfel} \end{cases} \quad \begin{matrix} \rightarrow \text{matriceal se poate scrie:} \\ V = C \cdot U \cdot C^T \\ C = \left\{ \alpha(k) \cos \frac{\pi(2m+1)k}{2N} \right\} \end{matrix}$$

## Tehnici de codare video: JPEG

> modul de codare (continuare):

4. DCT – Discrete Cosinus Transform: **transformare**



→ transf. inversă (C este unitară):

$$C^{-1} = C^T, \Rightarrow U = C^T \cdot V \cdot C$$

→ imaginea U poate fi scrisă ca:

$$U = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} B_{k,l} \cdot v(k, l)$$

unde  $B_{k,l}$  formează baza de imagini a transformatei DCT (~ "frecvențe").

$$\Rightarrow B_{k,l} = c_k \cdot c_l^T$$

unde  $c_k$  reprezintă coloana  $k$  a matricei  $C^T$ .

bază imaginii  $B_{k,l}$  transformata DCT (N=64)

## Tehnici de codare video: JPEG

> modul de codare (continuare):

4. DCT – Discrete Cosinus Transform: **transformare**

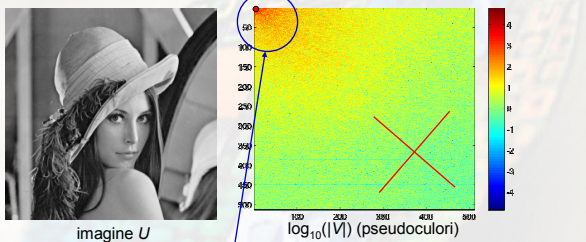
→ cu alte cuvinte imaginea U este o sumă ponderată de "frecvențe" spațiale:

$$U = \begin{matrix} \text{[Image]} \end{matrix} \cdot v(0,0) + \begin{matrix} \text{[Image]} \end{matrix} \cdot v(0,1) + \begin{matrix} \text{[Image]} \end{matrix} \cdot v(1,0) + \begin{matrix} \text{[Image]} \end{matrix} \cdot v(1,1) + \dots + \begin{matrix} \text{[Image]} \end{matrix} \cdot v(4,4) + \dots$$

## Tehnici de codare video: JPEG

> modul de codare (continuare):

### 4. DCT – Discrete Cosinus Transform: exemplu



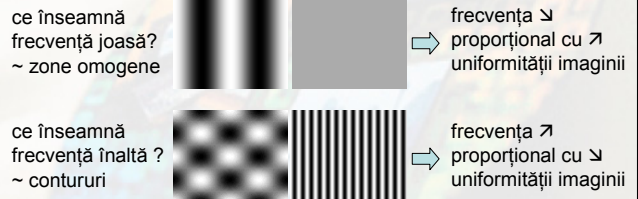
- energia este concentrată în doar câțiva coeficienți de "frecvențe joase";
- primul coeficient este cel mai important (DC = valoare medie);

## Tehnici de codare video: JPEG

> modul de codare:

### 4. DCT – Discrete Cosinus Transform: compresie

→ compresie = reprezentarea grosieră a coeficienților de pondere mică ce corespund "frecvențelor înalte";



[ Matlab: DCT 1D  $dct(u)$  și  $idct(v)$ ; DCT 2D  $dct2(U)$  și  $idct2(V)$  ]

## Tehnici de codare video: JPEG

[sursă imagine Wikipedia]

> modul de codare:

### 5. cuantizare:

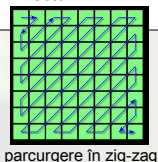
coeficienții DCT sunt scalați individual (împărțiți la o constantă ~ tabelă de cuantizare) și rotunjiți la primul întreg;  
= eliminare detalii (coeficienți 0) și valori mici pentru "frecvențe" joase;

### 6. codarea entropiei:

coeficienții sunt parcurși în ordinea importanței, matrice → vector și apoi codați fără pierdere:

> RLE (run-length encoding);  
+ Huffman;

> pentru coeficienții DC se codează doar diferențele (DPCM = Differential Pulse Code Modulation);

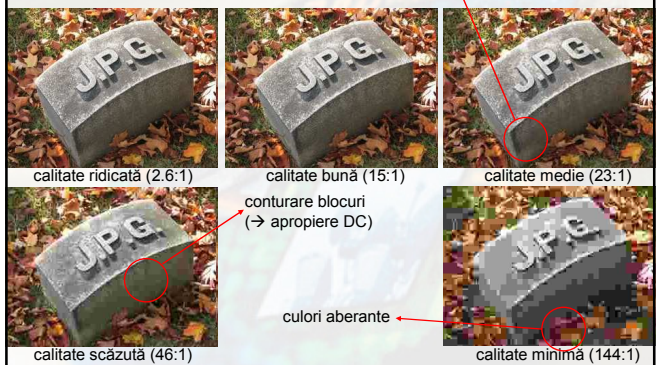


parcursire în zig-zag

## Tehnici de codare video: JPEG

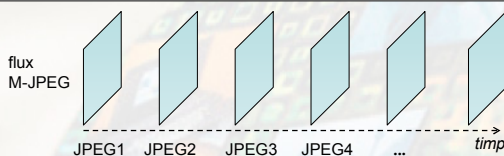
[sursă imagini Wikipedia]

> exemple:



## Tehnici de codare video: M-JPEG

Motion-JPEG – fiecare cadrul al secvenței de imagini este comprimat folosind standardul JPEG (codare intra-cadru),  
→ primul pas spre compresia mișcării;



- datorită codării individuale fluxul video poate fi editat
- rată de compresie între 2:1 și 20:1 (stocare DVC – Digital Videotape);

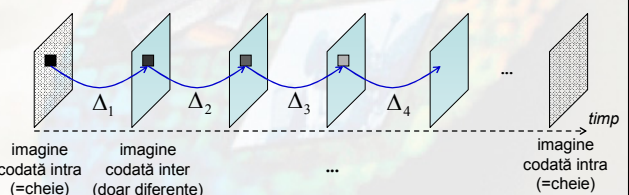
> ineficient !

## Tehnici de codare video: MPEG-1

Moving Picture Experts Group 1993 – MPEG-1 primul standard de compresie video în adevăratul sens al cuvântului (stocare VideoCD);  
→ al doilea pas major: **compresia temporală (inter-cadru)**.

↳ **compresie intra-cadru** = folosește principiul JPEG (DCT);

↳ **compresie inter-cadru** = se folosește de redundanța temporală, pe principiul următor:  
[C. Poynton, Digital Video and HDTV Algorithms and Interfaces, 2003]



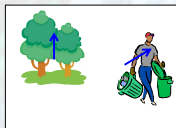
## Tehnici de codare video: MPEG-1

↳ **compresie inter-cadru** (continuare) [D. Marshall, Multimedia Course, 2001]

**problemă:** mișcarea obiectelor face ca valorile pixelilor să varieze foarte mult între cadre succesive temporale:  
→ diferențele (eroarea de predicție) sunt mari încât codarea inter devine total neeficientă;

**soluția:** *estimarea mișcării și compensarea mișcării.*

> ipoteză: diferența dintre cadre consecutive este dată de deplasarea obiectelor (diferență nulă = se transmite doar un cadru);



> deplasare cameră în jos;  
> mișcare obiect;

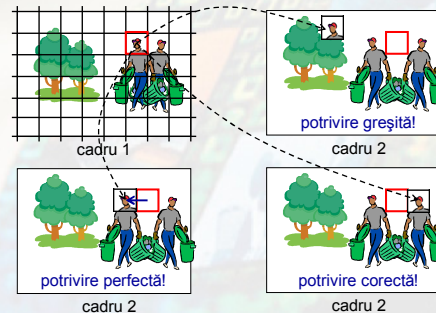
cadru 1

cadru 2

## Tehnici de codare video: MPEG-1

↳ **compresie inter-cadru** (continuare) (detalii M4 – Informația de Mișcare)

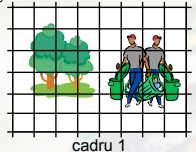
estimarea mișcării (principiu): se caută noua poziție a fiecărui bloc de pixeli;



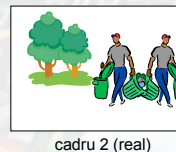
## Tehnici de codare video: MPEG-1

↳ **compresie inter-cadru** (continuare)

compensarea mișcării (principiu): cadrul este reconstituit prin deplasarea blocurilor în concordanță cu vectorii de mișcare;



cadru 1



cadru 2 (real)

> în realitate deplasările pixelilor sunt complexe  
→ nu ne putem baza doar pe vectorii de mișcare;

> codăm și eroarea de aproximare (reziduală);  
→ precizie estimare 7, eroare 3;

> dacă estimarea nu are o precizie acceptabilă  
→ bloc codat intra.



cadru 2 (estimat/compensat)



cadru 2 (eroare reziduală)

## Tehnici de codare video: MPEG-1

> modul de codare al imaginilor:

o secvență MPEG este partiționată în grupuri succesive de imagini (GOP – groups of pictures) ce conțin următoarele tipuri de cadre:

→ **I-picture** (intra-picture):

- imagine codată independent de celelalte cadre (codată intra);
- este prima imagine dintr-un GOP;
- constituie un punct de editare al fluxului MPEG fiind "autosuficientă";
- pe baza ei sunt reconstituite alte cadre codate non-intra (=anchor);

→ **P-picture** (forward predicted):

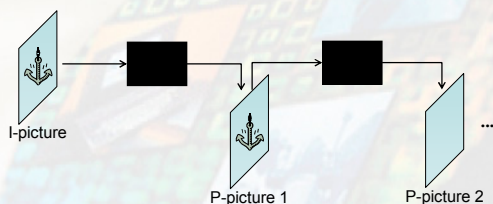
- imagine reconstituită pe baza celei mai apropiate imagini cheie (ex. I-picture);
- după ce este reconstituită devine noua imagine cheie (anchor).

## Tehnici de codare video: MPEG-1

> modul de codare al imaginilor (continuare):

→ **P-picture** (forward predicted): exemplu

[C. Poynton, Digital Video and HDTV Algorithms and Interfaces, 2003]



> P-picture 1 este dedusă/estimată (predicted) din I-picture;

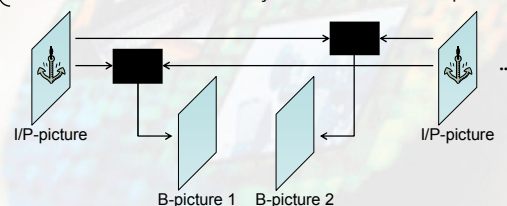
P-picture 2 este dedusă din P-picture 1.

## Tehnici de codare video: MPEG-1

> modul de codare al imaginilor (continuare):

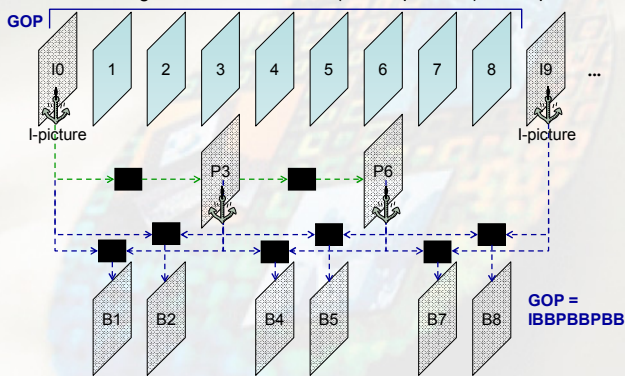
→ **B-picture** (bidirectional predicted, inovație MPEG):

- imagine reconstituită bidirecțional (forward and backward prediction) prin medierea compensării mișcării din cadre cheie anterioare cât și ulterioare;
- doar vizualizată, nu este folosită pentru predicția altor cadre!
- contribuie semnificativ la creșterea factorului de compresie.



## Tehnici de codare video: MPEG-1

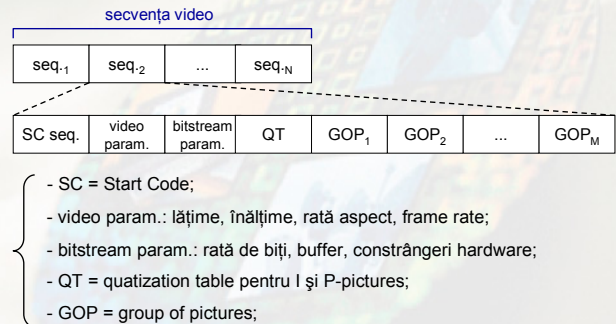
> ierarhia imaginilor în fluxul MPEG (I, P, B-pictures): exemplu



## Tehnici de codare video: MPEG-1

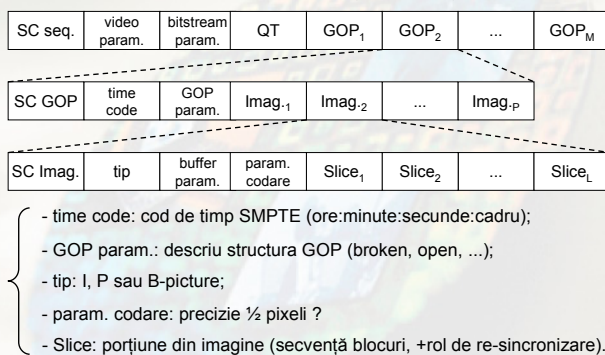
> structura schematică a fluxului MPEG:

[D. Marshall, Multimedia Course, 2001]



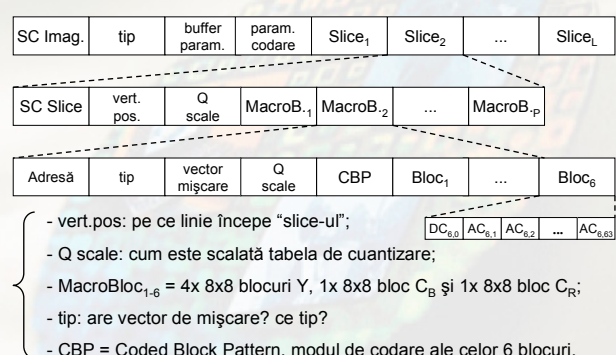
## Tehnici de codare video: MPEG-1

> structura schematică a fluxului MPEG (continuare):



## Tehnici de codare video: MPEG-1

> structura schematică a fluxului MPEG (continuare):



## Tehnici de codare video: MPEG-2

Moving Picture Experts Group 1994 – MPEG-2 în principal extensia lui MPEG-1 pentru scanare de tip i (interlaced);  
→ țintă: video de înaltă calitate la 4-15Mb/s: Video on Demand (VOD), TV digital și satelit (SDTV și HDTV), stocare digitală DVD - Video;

> câteva date:

[S. Aramvith, M.-T. Sun, MPEG-1 and MPEG-2 Video Standards, 1999]

| nivel                | profil |                                                          |        |        |         |
|----------------------|--------|----------------------------------------------------------|--------|--------|---------|
|                      | Simple | video pentru teleconferințe, întârzieri mici (low-delay) |        |        |         |
| High 1920x1152@60fps | -      | 80Mb/s                                                   | -      | -      | 100Mb/s |
| High 1440x1152@60fps | -      | 60Mb/s                                                   | -      | 60Mb/s | 80Mb/s  |
| Main 720x576@30fps   | 15Mb/s | 15Mb/s                                                   | 15Mb/s | -      | 20Mb/s  |
| Low 352x288@30fps    | -      | 4Mb/s                                                    | 4Mb/s  | -      | -       |

## Tehnici de codare video: MPEG-2

Moving Picture Experts Group 1994 – MPEG-2 în principal extensia lui MPEG-1 pentru scanare de tip i (interlaced);  
→ țintă: video de înaltă calitate la 4-15Mb/s: Video on Demand (VOD), TV digital și satelit (SDTV și HDTV), stocare digitală DVD - Video;

> câteva date (continuare):

| nivel                | profil |        |                                                                       |        |        |
|----------------------|--------|--------|-----------------------------------------------------------------------|--------|--------|
|                      | Simple | Main   | ~cel mai uzual, video de înaltă calitate: VOD, DVD, Digital TV, HDTV; |        |        |
| High 1920x1152@60fps | -      | 80Mb/s |                                                                       |        |        |
| High 1440x1152@60fps | -      | 60Mb/s | -                                                                     | 60Mb/s | 80Mb/s |
| Main 720x576@30fps   | 15Mb/s | 15Mb/s | 15Mb/s                                                                | -      | 20Mb/s |
| Low 352x288@30fps    | -      | 4Mb/s  | 4Mb/s                                                                 | -      | -      |

## Tehnici de codare video: MPEG-2

Moving Picture Experts Group 1994 – MPEG-2 în principal extensia lui MPEG-1 pentru scanare de tip i (interlaced);  
→ țintă: video de înaltă calitate la 4-15Mb/s: Video on Demand (VOD), TV digital și satelit (SDTV și HDTV), stocare digitală DVD - Video;

> câteva date (continuare):

| nivel                | profil |        |              |                                                                       |        |
|----------------------|--------|--------|--------------|-----------------------------------------------------------------------|--------|
|                      | Simple | Main   | SNR scalable | (Signal-to-Noise-Ratio) permite mai multe niveluri de calitate video; |        |
| High 1920x1152@60fps | -      | 80Mb/s | -            |                                                                       |        |
| High 1440x1152@60fps | -      | 60Mb/s | -            |                                                                       |        |
| Main 720x576@30fps   | 15Mb/s | 15Mb/s | 15Mb/s       | -                                                                     | 20Mb/s |
| Low 352x288@30fps    | -      | 4Mb/s  | 4Mb/s        | -                                                                     | -      |

## Tehnici de codare video: MPEG-2

Moving Picture Experts Group 1994 – MPEG-2 în principal extensia lui MPEG-1 pentru scanare de tip i (interlaced);  
→ țintă: video de înaltă calitate la 4-15Mb/s: Video on Demand (VOD), TV digital și satelit (SDTV și HDTV), stocare digitală DVD - Video;

> câteva date (continuare):

| nivel                | profil                                            |        |        |                    |         |
|----------------------|---------------------------------------------------|--------|--------|--------------------|---------|
|                      | suportă mai multe niveluri de rezoluție spațială; |        |        | Spatially scalable | High    |
| High 1920x1152@60fps | -                                                 | -      | -      | -                  | 100Mb/s |
| High 1440x1152@60fps | -                                                 | 60Mb/s | -      | 60Mb/s             | 80Mb/s  |
| Main 720x576@30fps   | 15Mb/s                                            | 15Mb/s | 15Mb/s | -                  | 20Mb/s  |
| Low 352x288@30fps    | -                                                 | 4Mb/s  | 4Mb/s  | -                  | -       |

## Tehnici de codare video: MPEG-2

Moving Picture Experts Group 1994 – MPEG-2 în principal extensia lui MPEG-1 pentru scanare de tip i (interlaced);  
→ țintă: video de înaltă calitate la 4-15Mb/s: Video on Demand (VOD), TV digital și satelit (SDTV și HDTV), stocare digitală DVD - Video;

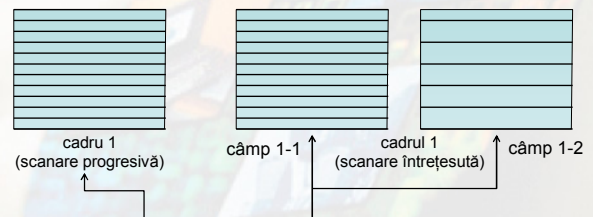
> câteva date (continuare):

| nivel                | profil                                                                                                |        |        |        |         |
|----------------------|-------------------------------------------------------------------------------------------------------|--------|--------|--------|---------|
|                      | suportă mai multe niveluri de calitate, rezoluție spațială și subeșantionare (4:4:4, 4:2:2 și 4:2:0); |        |        |        | High    |
| High 1920x1152@60fps | -                                                                                                     | -      | -      | -      | 100Mb/s |
| High 1440x1152@60fps | -                                                                                                     | 60Mb/s | -      | 60Mb/s | 80Mb/s  |
| Main 720x576@30fps   | 15Mb/s                                                                                                | 15Mb/s | 15Mb/s | -      | 20Mb/s  |
| Low 352x288@30fps    | -                                                                                                     | 4Mb/s  | 4Mb/s  | -      | -       |

## Tehnici de codare video: MPEG-2

> principiul de codare MPEG-2 similar cu al lui MPEG-1 (codare intra și inter), totuși există diferențe majore practice (cele mai importante):  
[S. Aramvith, M.-T. Sun, MPEG-1 and MPEG-2 Video Standards, 1999]

↳ **codare video întrețesută (interlaced):**



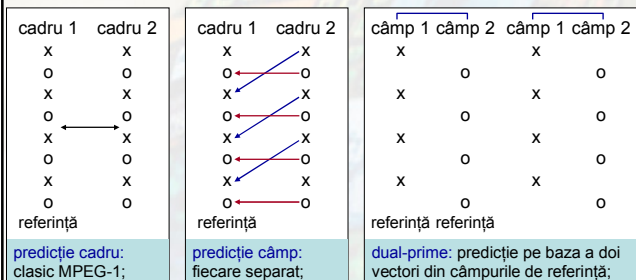
> codare la **nivel de cadru** sau la **nivel de câmp**;

## Tehnici de codare video: MPEG-2

> diferențe majore față de MPEG-1 (continuare):

↳ **codare video întrețesută (continuare):**

> compensarea mișcării este adaptată la nivel de cadru sau câmp  
→ dual-prime motion compesation (mai eficientă):

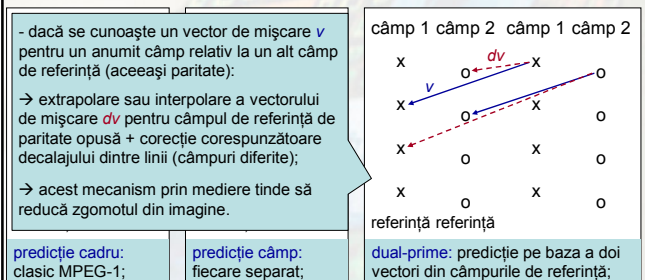


## Tehnici de codare video: MPEG-2

> diferențe majore față de MPEG-1 (continuare):

↳ **codare video întrețesută (continuare):**

> compensarea mișcării este adaptată la nivel de cadru sau câmp  
→ dual-prime motion compesation (mai eficientă):



## Tehnici de codare video: MPEG-2

> diferențe majore față de MPEG-1 (continuare):

↳ **codare scalabilă**: "codare pe straturi"



→ 1-2 straturi de detaliu (enhancement-layers) permit îmbunătățirea calității:  
- rezoluție spațială mai mare,  
- rezoluție temporală mai mare, ...

→ un strat de bază (basis-layer) ce furnizează o calitate video "standard";

## Tehnici de codare video: MPEG-2

> diferențe majore față de MPEG-1 (continuare):

↳ **codare scalabilă (continuare)**: 4 moduri

### 1. scalabilitate SNR

**basis**: coeficienți DCT cuantizați la o calitate moderată (rată de bit  $\searrow$ );  
**enhancement**: diferența dintre coeficienții DCT necuantizați și cei cuantizați;  
→ în funcție de preferințe se decodează doar basis sau ambele straturi.

### 2. scalabilitate spațială

**basis**: subeșantionare spațială prin decimare pentru o rezoluție mică;  
**enhancement**: supraeșantionare prin interpolare + stocare predicție mișcare;  
→ furnizează simultan rezoluții diferite în funcție de terminalul utilizatorului ex.: trecere de la SDTV la HDTV;

## Tehnici de codare video: MPEG-2

> diferențe majore față de MPEG-1 (continuare):

↳ **codare scalabilă (continuare)**: 4 moduri

### 3. scalabilitate temporală

**basis**: codare la o rată de cadre redusă;  
**enhancement**: compensarea mișcării pentru reconstituirea unor cadre suplimentare;  
→ comunicații fără fir, conexiune limitată → reducere număr cadre/s;

### 4. scalabilitate hibridă

- SNR + spațial : HDTV, SDTV, Videotelefon;  
- Spațial + temporal: HDTVp + res. temp.  $\nearrow$  și HDTVi & SDTVi basic;  
- SNR + temporal : HDTVp îmbunătățit și HDTVp basic.

## Tehnici de codare video: MPEG-2

> diferențe majore față de MPEG-1 (continuare):

↳ **partiționarea datelor**: pentru transmisie robustă a datelor în medii predispușe la erori;

> blocurile de 8x8 coeficienți DCT sunt împărțite în partiții:

 de nivel scăzut = informație critică (frecvențe  $\searrow$ ),  
high priority channel.

 de nivel ridicat = detalii (frecvențe  $\nearrow$ ),  
low priority channel.

↳ **depășirea erorilor (error resilience)**:

erori ce apar: în antet secvență, antet cadre, antet "slice", antet macrobloc, coeficienți DCT, vectori de mișcare, ... → problemă!

- **vectori de mișcare ascunși** ("concealment motion vectors") în antetul macroblocurilor, dacă datele din macrobloc sunt corupte  
→ compensare din cadrul anterior;

## Tehnici de codare video: MPEG-2

> diferențe majore față de MPEG-1 (continuare):

↳ **depășirea erorilor (continuare)**:

- **structura "slice-urilor"**: reprezintă markeri pentru resincronizare dacă are loc o eroare de transmisie,  
→ slice-uri cu mai puține blocuri;

- **localizare temporală**: este folosită pentru a opri propagarea unei erori de decodare de la un cadru la altul (natural o eroare se oprește la următoarea imagine I-picture),  
→ creșterea numărului de imagini cheie I-picture (periodice);

> avantaje: acces aleator în fluxul video, facilități fast-forward și fast reverse play, ...

- **codarea scalabilă** (vezi slide anterior)



## Tehnici de codare video: MPEG-4

Moving Picture Experts Group 1999 – MPEG-4: mai multe standarde (27) denumite "părți" ce acoperă diferite aspecte de codare;

→ țintă principală: codare la o rată de bit foarte redusă (mult mai eficient);

→ al treilea pas major: **codarea obiectelor video** (=video processing):

- **video object coding**: codarea obiectelor video rectangulare sau de formă arbitrară, atât naturale cât și de sinteză;

- **mesh object coding**: codarea obiectelor vizuale pe baza "suprafeței" acestora (mesh-wireframe);

- **model-based coding**: codarea pe bază de model a animației de sinteză a feței și corpului;

- **still texture coding**: codarea texturilor statice;

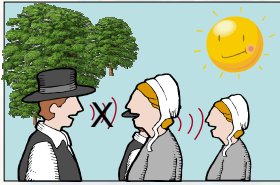
[B. Erol, F. Kossentini, A. Joch, G.J. Sullivan, L. Winger, MPEG-4 Visual and H.264/AVC: Standards for Modern Digital Video, 2009]

## Tehnici de codare video: MPEG-4

**obiect audio-vizual (AVO)** = un segment video de dimensiune arbitrară ce are o semnificație semantică ( $\supset$  inf. vizuală, audio sau mixtă);

**scenă audio-vizuală** = o mulțime de AVO identificate prin coordonatele spațiale și temporale, ex.  $\{AVO_1, AVO_2, \dots, AVO_N\}$ ;

→ informații transmise separat ("scene description information") ce permit reconstrucția scenei la nivelul utilizatorului;



ex. scenă audio-vizuală

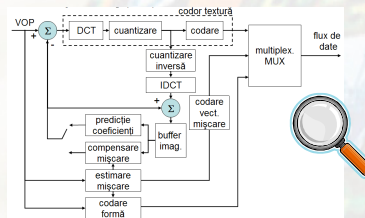
AVO<sub>1</sub>: copaci (natural),  
AVO<sub>2</sub>: soare (sinteză),  
AVO<sub>3</sub>: persoane (sinteză),  
AVO<sub>4</sub>: voce,  
AVO<sub>3+4</sub>: persoane+voce (compus),

→ la utilizator, AVO pot fi ascunse, șterse, modificate (scalare, fps 7, ...)

## Tehnici de codare video: MPEG-4

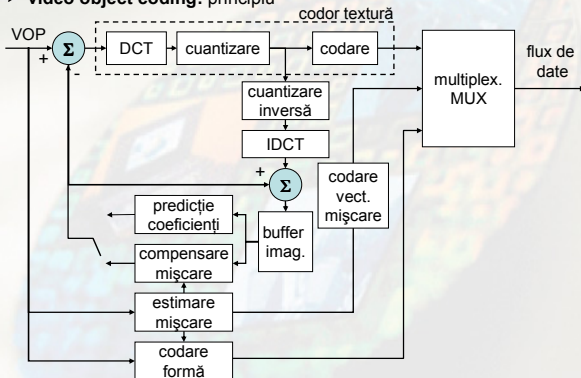
→ **video object coding**: presupune codarea vectorilor de mișcare, textură, forme și fundal (sau "sprite"):

**Video Object Plane (VOP)** = o proiecție 2D a unui VO (obiect video,  $\subset$  AVO) la un moment  $t$ , ce este definită prin textură (luminanță-crominanță) și formă;  
→ acces la nivel de VO cât și VOP, implică segmentarea scenei!



## Tehnici de codare video: MPEG-4

→ **video object coding**: principiu



## Tehnici de codare video: MPEG-4

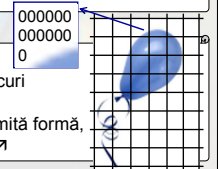
→ **video object coding** (continuare)

**codare vectori de mișcare:**

- vectori de mișcare nerestricționați = au voie să indice în afara zonei codate din VOP-ul de referință;
- compensarea mișcării cu blocuri de dimensiuni variabile;
- compensare globală a mișcării (camera video);
- precizie "quarter-sample" (QPel) = precizia deplasării este de 1/4 pixeli, vectori de mișcare fracționari → posibil prin interpolare;

**codare textură:**

- proces similar cu MPEG-1 și MPEG-2: macroblocuri
- blocuri → DCT ...
- dacă un macrobloc depășește un VOP de o anumită formă, → este bordat cu 0 pe acea regiune = compresie 7

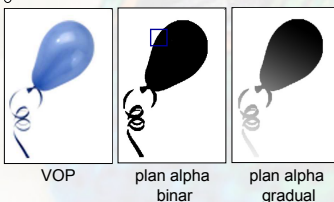


## Tehnici de codare video: MPEG-4

→ **video object coding** (continuare)

**codare formă:**

- pentru a face distincția între textură și formă se folosește conceptul de "alpha-plane" (hartă binară sau cu niveluri de gri);
- 0 = total transparent,
- 255 = total opac,
- [1;254] niveluri progresive de transparentă.



→ **codare cu pierderi sau fără:** blocurile de frontieră codate aritmetic; plan transparentă ~ codare textură;

→ **codare intra sau inter:** local, pe baza valorilor vecine sau estimare mișcare formă și codare diferență.

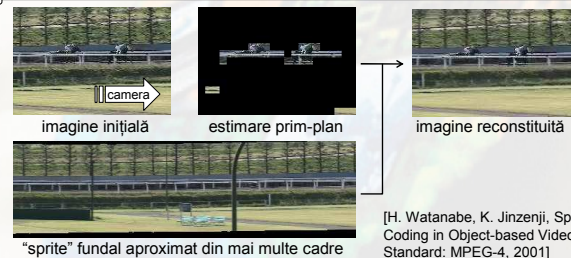
## Tehnici de codare video: MPEG-4

→ **video object coding** (continuare)

**codare "sprite":**

(detalii M4 – Informația de Mișcare)

- codarea separată a VO ce sunt, fie statice pe durata unei scene video sau a căror modificări pot fi approximate pe baza unor VO de plecare;
- > izolare fundal de prim plan (codat o singură dată)



[H. Watanabe, K. Jinzenji, Sprite Coding in Object-based Video Coding Standard: MPEG-4, 2001]

## Tehnici de codare video: MPEG-4

> MPEG-4 vedere de ansamblu (părți):

### 1999 part 1: Systems

descrie sincronizarea și multiplexarea video și audio;

### 1999 part 2: Visual (prezentat anterior)

compresia datelor vizuale (video, texturi, imagini de sinteză)  
= Advanced Simple Profile;

### 1999 part 3: Audio

codarea perceptuală a semnalului audio;

### 2000 part 4: Conformance testing

protocol de testare a conformității cu alte părți ale standardului;

### 2000 part 5: Reference software

model de implementare de referință

## Tehnici de codare video: MPEG-4

> MPEG-4 vedere de ansamblu (părți): continuare

### 1999 part 6: Delivery Multimedia Integration Framework

interfața dintre aplicație și nivelul de transport (rețele);

### 2002 part 7: Optimized reference software

modele de implementare ale unor eventuale îmbunătățiri ale standardului;

### 2004 part 8: Carriage of contents over IP networks

transport MPEG-4 pe rețele cu IP;

### 2004 part 9: Reference hardware description

model de implementare hardware;

### 2003 part 10: Advanced Video Coding

codare semnale video ~ H.264;

### 2005 part 11: Scene description and application engine

conținut interactiv 2D și 3D; XMT – reprezentare textuală conț. multimedia;

## Tehnici de codare video: MPEG-4

> MPEG-4 vedere de ansamblu (părți): continuare

### 2004 part 12: ISO base media file format

format de fișier de stocare conținut media;

### 2004 part 13: Intellectual Property Management and Protection

protecția și distribuția materialelor cu drept de autor;

### 2003 part 14: MP4 file format

container pentru MPEG-4 bazat pe part 12;

### 2004 part 15: Advanced Video Coding file format

container pentru MPEG-4 ce permite stocarea și în alte formate;

### 2004 part 16: Animation Framework eXtension

model de reprezentare a graficii 3D (geometrie, textură, animație,...);

### 2006 part 17: Streaming text format

subtitrări;

## Tehnici de codare video: MPEG-4

> MPEG-4 vedere de ansamblu (părți): continuare

### 2004 part 18: Font compression and streaming

manipulare text;

### 2004 part 19: Synthesized texture stream

crearea de clipuri de sinteză cu o rată de bit foarte redusă;

### 2006 part 20: Lightweight Application Scene Representation

multimedia dedicat dispozitivelor mobile, integrate, ...;

### 2006 part 21: MPEG-J Graphics

mediu de programare pentru multimedia interactiv MPEG + Java;

### 2007 part 22: Open Font Format

fonturi scalabile;

### 2008 part 23: Symbolic Music Representation

## Tehnici de codare video: MPEG-4

> MPEG-4 vedere de ansamblu (părți): continuare

### 2008 part 24: Audio and system interaction

fișiere MPEG-4 și MPEG-4 audio;

### 2009 part 25: 3D Graphics Compression Model

interconectare grafică 3D MPEG-4 cu grafică din alte standarde;

### 2010 part 26: Audio conformance

### 2010 part 27: 3D Graphics conformance

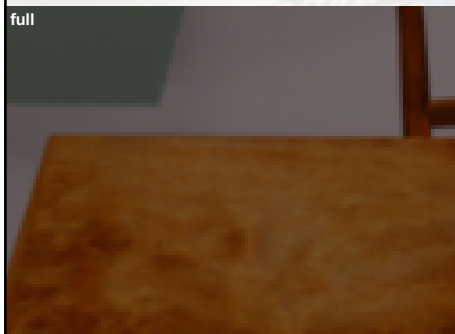
> concluzie: pachet de tehnici de codare ce deservește o gamă extrem de largă de aplicații + adaptare continuă la cerințele actuale;

> din păcate, din motive practice, codarea obiectelor rămâne în mare doar un concept de interes științific!

## Tehnici de codare video: exemple

> comparație calitate vizuală

full



zoom 600%

PLAT

- imagini sinteză 3D  
- 4s (100 cadre),  
- 1280x720 pixeli,  
- 2.496 kbps,

**full:** 264.611KB

**M-JPEG:**  
3.518KB (75:1)

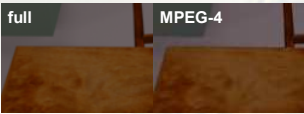
**MPEG-1:**  
1.040KB (254:1)

**MPEG-2:**  
1.042KB (254:1)

**MPEG-4 p10:**  
1.017KB (260:1)

### Tehnici de codare video: exemple

> comparație calitate vizuală



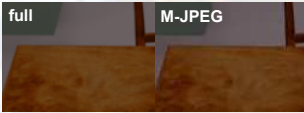
**MPEG-4 p10:** 1.017KB



**MPEG-2:** 1.042KB



**MPEG-1:** 1.040KB



**M-JPEG:** 3.518KB

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU 66

### Tehnici de codare video: exemple

> comparație calitate vizuală la o rată de bit redusă

- imagini sinteză 3D  
- 4s (100 cadre),  
- 1280x720 pixeli,  
- 1.008 kbps,

→

- 528 kbps,

↓

**MPEG-1:** 359KB (737:1)  
**MPEG-2:** 364KB (727:1)  
**MPEG-4 p2:** 382KB (693:1)  
**MPEG-4 p10:** 205KB (1291:1)

**MPEG-4 p10:** 391KB (677:1)  
**MPEG-2:** 579KB (457:1)  
**MPEG-1:** 576KB (459:1)



zoom 500%

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU 67

### Tehnici de codare video: exemple

> comparație calitate vizuală la o rată de bit redusă



**MPEG-4 p2:** 382KB



**MPEG-2:** 364KB



**MPEG-1:** 359KB

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU 68

### > M1. Informația video

[ Manipulare video – utilitare ]

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU 69

### Formate video

> **tehnici de codare:** algoritmi/metode de compresie video;

> **codec video:** un dispozitiv sau program ce furnizează un pachet de proceduri ce permit *compresia* (encoding) și *decompresia* (decoding) datelor video digitale;  
→ implementarea tehnicilor de codare;

> **fișier video:** un container ce specifică modul de stocare pe disc a informației video codate.

> **player video:** un pachet de utilitare ce furnizează o interfață grafică de vizualizare a conținutului video;  
→ se folosesc de codec-urile existente în sistem,

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU 70

### Formate video

↳ **codec-uri video:**



- proprietar DivX, Inc. (fost DivXNetworks, Inc.),  
- include MPEG-4 Part 2 ASP și H.264/MPEG-4 Part 10 AVC.  
[<http://www.divx.com>]



- open source, concurența directă DivX (←XviD),  
- multiplatformă, include MPEG-4 Part 2 ASP.  
[<http://www.xvid.org>]



- realizat de 3ivx Technologies, sisteme embedded (viteză de prelucrare redusă)  
- include MPEG-4 Part 2 ASP.  
[<http://www.3ivx.com>]



- RealVideo, proprietar RealNetworks, destinat flux video în rețea ("streaming", conținutul video este vizualizat progresiv),  
- mai multe versiuni rv10-rv20: H.263, rv30-40: H.264, ...  
[<http://www.realnetworks.com>]

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU 71



## Accesare informație video: mediul Matlab

↳ informații AVI: funcția **aviinfo()**

```
mov_info=aviinfo('nume_fisier');
```

> deschide fișierul *nume\_fisier.avi* din calea curentă, și extrage în structura *mov\_info* informații despre conținutul video:

```
.Filename: 'HDMPEG4@528kbps.avi'
.FileSize: 390732
.FileModDate: '01-Nov-2010 15:14:38'
.NumFrames: 98
.FramesPerSecond: 25
.Width: 1280
.Height: 720
.ImageType: 'truecolor'
.VideoCompression: 'DIVX'
.Quality: 0
.NumColorMapEntries: 0
```

## Accesare informație video: mediul Matlab

↳ creare fișier AVI: funcția **avifile()**

```
new_mov=avifile('nume_fisier.avi');
new_mov.Fps=25;
new_mov.Compression='Indeo5';
new_mov.Quality=90;

% adaugare cadre
for i=1:125
    cadru=zeros(480,640,3);
    new_mov=addframe(new_mov,cadru);
end
new_mov=close(new_mov);
```

se deschide pe disc un flux video în *nume\_fisier.avi* controlat prin *new\_mov*;

setări parametri de codare:  
- codec: 'Indeo3', 'Indeo5', 'Cinepak', 'MSVC', 'RLE' sau 'None',  
- calitate, fps, ...

adăugare secvențială de cadre la secvența *new\_mov*,

închidere flux video (obligatorie).

## Accesare informație video: mediul Visual C++

> librăria Microsoft AVIFile

[<http://msdn.microsoft.com/en-us/library/dd756808%28VS.85%29.aspx>]

↳ inițializare și variabile control flux video

```
AVIFileInit();
```

inițializare librărie AVIFile (o singură dată);  
**AVIFileExit()**; la terminare program!

```
PAVIFILE avi;
```

```
PAVISTREAM pStream;
```

pointer la o interfață IAVIFile (deschidere și manipulare fișiere);

pointer la o locație de memorie în care va fi stocat un handle către un flux de date (video, audio, ...);

```
PGETFRAME pFrame;
```

pointer la o interfață IGetFrame (extragere, decompresie ...);

```
BOOL MovieIsOpen;
```

```
BITMAPINFOHEADER bih;
```

```
AVIFILEINFO avi_info;
```

antet fișier DIB (Device Independent Bitmap) și fișier AVI → acces cadre.

## Accesare informație video: mediul Visual C++

> librăria Microsoft AVIFile (continuare)

↳ deschidere flux video

```
// închidere flux anterior
if (MovieIsOpen)
{
    MovieIsOpen=False;
    AVIStreamGetFrameClose(pFrame);
    if (pStream)
        AVIStreamRelease(pStream);
    if (avi)
        AVIFileRelease(avi);
    pFrame=NULL; pStream=NULL; avi=NULL;
}
```

MovieIsOpen indicator boolean (definit, nu este standard!)

eliberare resurse folosite la decompresie cadre;

închidere flux de date;

închidere fișier avi;

pointeri nuli;

## Accesare informație video: mediul Visual C++

> librăria Microsoft AVIFile (continuare)

↳ deschidere flux video (continuare)

```
// deschidere fisier nou
```

```
int res;
```

```
res=AVIFileOpen(&avi, szFileName, OF_SHARE_DENY_WRITE, NULL);
```

deschide un fișier AVI specificat de *szFileName* și returnează în *avi* pointerul la interfața acestuia;

deschidere non-exclusivă (alte programe pot citi fișierul)

```
if (res!=AVIERR_OK)
```

```
{
```

```
    // unable to open the AVI file!
```

```
    ...
```

```
}
```

eroare la deschidere;

## Accesare informație video: mediul Visual C++

> librăria Microsoft AVIFile (continuare)

↳ deschidere flux video (continuare)

```
// deschidere flux de date nou
MovieIsOpen=True;
```

returnează un pointer la un flux de date din fișierul referențiat de *avi*, în *pStream*;

```
res=AVIFileGetStream(avi, pStream, streamtypeVIDEO, 0);
```

streamtypeVIDEO = flux video (AUDIO, MIDI, TEXT);

```
if (res!= AVIERR_OK)
```

```
{
```

```
    // stream open error! ...
```

```
}
```

eroare la deschidere;

```
AVIFileInfo(avi, &avi_info, sizeof(AVIFILEINFO));
```

extrage antet fișier avi → *avi\_info*;  
.dwWidth x .dwHeight = dimensiuni;  
.dwRate / .dwScale = fps.

### Accesare informație video: mediul Visual C++

> librăria Microsoft AVIFile (continuare)

↳ deschidere flux video (continuare)

```
// pregătire accesare cadre video
int FrameNumber=AVIStreamLength(*pStream);
if (FrameNumber==0)
{
    // stream length is zero! ...
}
int FirstFrame=AVIStreamStart(*pStream);
if (FirstFrame==0)
{
    // error retrieving the first frame! ...
}
```

returnează numărul de "eșantioane" = cadre din fluxul video;

eroare, nu există nici un cadru video;

căutare poziție de plecare în flux = primul cadru video;

eroare;

### Accesare informație video: mediul Visual C++

> librăria Microsoft AVIFile (continuare)

↳ deschidere flux video (continuare)

```
// deschidere flux de cadre video
ZeroMemory(&bih, sizeof(BITMAPINFOHEADER));
bih.biBitCount=24; bih.biClrImportant=0;
bih.biClrUsed = 0; bih.biCompression = BI_RGB;
bih.biPlanes = 1; bih.biHeight = avi_info.dwHeight;
bih.biWidth = avi_info.dwWidth;
bih.biSize = sizeof(BITMAPINFOHEADER);
bih.biSizeImage = bih.biWidth*bih.biHeight*3;
*pFrame=AVIStreamGetFrameOpen(*pStream, &bih);
if (*pFrame==NULL)
{
    // frame format error! ...
}
```

alocare și scriere antet DIB → bih; (la adresa pointerului bih se inițializează cu 0 memorie pentru antet);

pregătește decompresia din fluxul pStream și dă controlul prin pFrame (imaginile conform bih)

eroare la format imagine;

### Accesare informație video: mediul Visual C++

> librăria Microsoft AVIFile (continuare)

↳ deschidere flux video (continuare)

```
// copiere cadru video
unsigned char *pDataImage= NULL;
unsigned char *pDIB = (unsigned char*)
    AVIStreamGetFrame(*pFrame, 100);
if (pDIB==NULL)
{
    // frame decompression failed! ...
}
BITMAPINFOHEADER BmpInfo;
memcpy(&BmpInfo.biSize, pDIB,
    sizeof(BITMAPINFOHEADER));
pDataImage = new unsigned char[BmpInfo.biSizeImage];
memcpy(pDataImage, pDIB+
    sizeof(BITMAPINFOHEADER), BmpInfo.biSizeImage);
```

pointer la cadru decompresat (copie);

returnează adresa la cadrul decompresat de indice 100 din fluxul pFrame → pDIB (char = 8biți);

eroare decompresie;

copiere date pDIB → pDataImage (pentru a modifica imaginea).

### Accesare informație video: mediul Visual C++

> librăria Microsoft AVIFile (continuare)

↳ deschidere flux video (continuare)

```
// modificare cadru, afișare etc.
for (int y=0; y<BmpInfo.biHeight; y++)
    for (int x=0; x<BmpInfo.biWidth; x++)
    {
        int Red=pDataImage[(BmpInfo.biHeight-1-y)*
            BmpInfo.biWidth*3+x*3+2];
        int Green=pDataImage[(BmpInfo.biHeight-1-y)*
            BmpInfo.biWidth*3+x*3+1];
        int Blue=pDataImage[(BmpInfo.biHeight-1-y)*
            BmpInfo.biWidth*3+x*3];
    }
...
delete pDataImage;
```

parcurește imagine pe linii și coloane;

componenta de roșu (y,x)

componenta de verde (y,x)

componenta de albastru (y,x)

după calcule eliberare memorie pDataImage (copia cadrului decompresat).

### Manipulare informație video: biblioteca FFMPEG

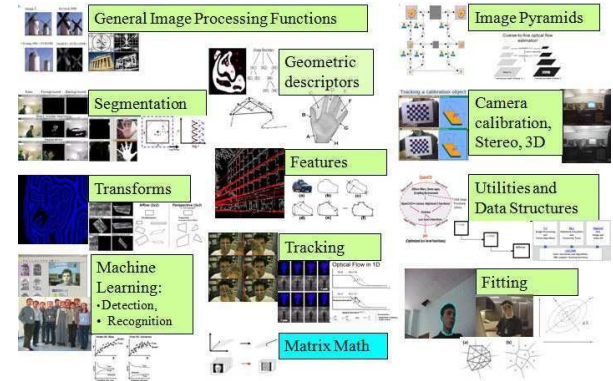


un pachet de librării "open source" de manipulare a datelor multimedia;

[<http://www.ffmpeg.org/>]

- > **ffmpeg**: utilitar conversie între diverse formate video (linie de comandă);
- > **ffserver**: server de streaming multimedia live;
- > **ffmpeg**: media player;
- > **libavutil**: librărie de funcții specifice de programare (numere aleatoare, structuri de date, funcții matematice, ...);
- > **libavcodec**: librărie de codec-uri audio-video (codor-decodor);
- > **libavformat**: librărie multiplexare-demultiplexare pentru containere;
- > **libswscale**: librărie de funcții optimizate pentru scalare și conversie din diverse spații de culoare;
- > **libavdevice**: librărie de renderizare și extragere cadre ...

### Manipulare informație video: biblioteca OpenCV



### Utilitare de lucru cu informația video: VirtualDub

utilitar "open source" ce permite prelucrarea informației audio-video;  
(codul este disponibil + captura video)  
[http://www.virtualdub.org/]

video original (in)

video modificat /filtrat (out)

definire segment video [start-end]

play, stop, ...

demo ...

bară de navigație (time line)

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU 90

### Utilitare de lucru cu informația video: Super Converter

utilitar "freeware" ce permite conversia în/din marea parte a  
formatelor video existente;  
[http://www.erightssoft.com/SUPER.html]

opțiuni video: rezoluție, aspect, fps,  
rată de biți, ...

opțiuni audio: frecvență eșantionare,  
rezoluție, canale, rată de biți, ...

demo ...

listă fișiere de convertit (implicit  
fișierul rezultat este salvat în același  
director)

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU 91

### Utilitare de lucru cu informația video: VLC

VideoLAN: utilitar "open source" de vizualizare și streaming  
→ deschide flux video incomplet, stricat !  
[http://www.videolan.org/vlc/]

vizualizare (deschide DVD, flux  
video online, captură video, ...)

bară de navigare temporală

control: play, stop, forward, ...

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU 92

### > M1. Informația video [ Conceptul de indexare ]

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU 93

### Conceptul de indexare ("revisited")

> societatea actuală este una bazată pe informație → oamenii  
schimbă/accesază un volum de date multimedia important:

- text, ex. eBooks, eDocuments, etc.
- audio, ex. muzică, voce, etc.
- imagini, ex. poze, grafică, etc.
- secvențe de imagini ex. filme, flux video, etc.

> se estimează că în viitorul apropiat >60% din datele vehiculate  
pe Internet vor fi video (ex. YouTube ~ upload 24h pe minut);

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU 94

### Conceptul de indexare ("revisited")

> **problemă:** cum accesăm **informația relevantă** din aceste  
colecții imense de date ?

**video = cantitate mare de date**

- 1 minut > 1.500 imagini (la 25 cadre/s)
- 1 bază video > mii de filme (>135.000.000 imagini)

- vrem să *căutăm* o anumită informație,
- vrem să "răsfoim" baza de date,
- vrem să vizualizăm eficient conținutul (video),

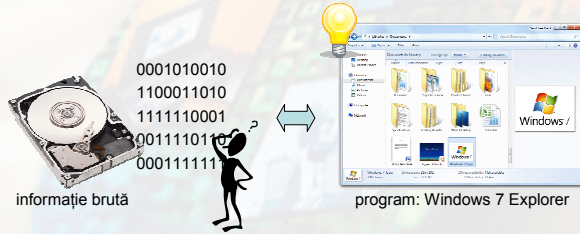
> **soluția existentă:** sistemele de indexare automată după conținut  
sau Content-based Indexing Systems  
(sunet: CBAR, imagini: CBIR, video: CBVR, etc.)

Analiza și Prelucrarea Secvențelor de Imagini, Ș.I. Bogdan IONESCU 95

## Conceptul de indexare ("revisited")

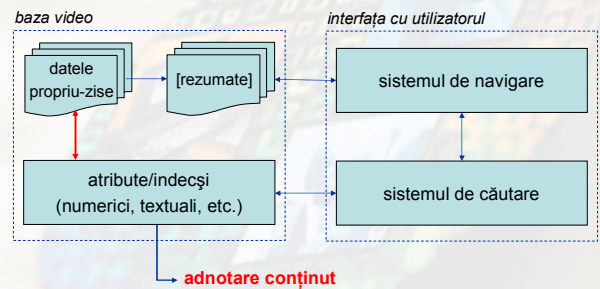
> **indexare** = adăugarea de informații suplimentare relative la conținutul datelor ce se doresc a fi indexate = **adnotare**

> exemplu simplu:



## Cum se realizează indexarea

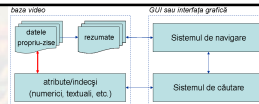
> diagrama de principiu a unui sistem de indexare video:



## Cum se realizează indexarea



mecanism indexare:



1. **adnotarea bazei ("offline")**: sunt generate atributele/indechi ce descriu proprietățile datelor (definiție *spațiu de caracteristici*):

- ↳ poate fi automată (ideal), semi-automată sau manuală;
- ↳ opțional, se pot genera rezumate de conținut (ex. trailer, imagini cheie) ce vor facilita navigarea ulterioară în bază;

2. **căutarea în bază** a informațiilor dorite:

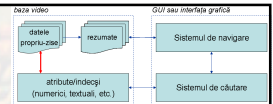
- ↳ formularea cererii de căutare ("query")

- |                                    |                               |
|------------------------------------|-------------------------------|
| • cuvânt cheie (query-by-keyword); | • gesturi (query-by-gesture); |
| • exemplu (query-by-example);      | • concept (query-by-concept); |
| • manuală, ~ sist. de navigare;    | • mimare (query-by-humming);  |

## Cum se realizează indexarea



mecanism indexare (continuare):

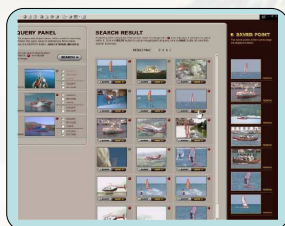


2. **căutarea în bază** a informațiilor dorite (continuare):

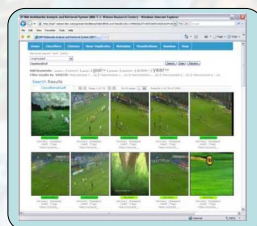
- ↳ cererea de căutare este convertită de sistem în atribute de conținut folosind aceleași informații ca în cazul etapei 1; (proiecție în spațiul de caracteristici)
- ↳ **căutarea propriu-zisă**: atributele obținute anterior sunt comparate succesiv cu cele din bază folosind o anumită măsură de similaritate (distanță);
- ↳ interacția cu utilizatorul: rezultatele sunt furnizate în ordinea relevanței de regulă folosind sistemul de navigare, opțional **feedback** utilizator;

## Sistemul de navigare

> permite accesul utilizatorului la informațiile din baza de date, ("concretizarea" sistemului de indexare):



Fischlär – Dublin City University  
orientat utilizator  
[<http://www.cdvp.dcu.ie/>]



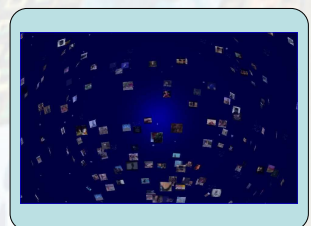
iMars – IBM Research  
orientat Web  
[<http://mp7.watson.ibm.com/>]

## Sistemul de navigare

> permite accesul utilizatorului la informațiile din baza de date, (exemple, continuare):



MediaMill – University of Amsterdam  
navigare temporală  
[<http://www.science.uva.nl/research/mediamill/>]



VideoVerse – LAPI România  
sistem 3D  
[<http://imag.pub.ro/VideoIndexingRP2/>]

## Principalele puncte critice ale indexării

- **atributele folosite:** care sunt atributele cele mai relevante ?  
→ maximizarea informației reprezentate și minimizarea dimensiunii spațiului de caracteristici.
- **măsura de distanță:** când două secvențe sunt similare ?  
(problematică mai complexă decât în cazul imaginilor)  
→ secvențele similare după un anumit criteriu trebuie să returneze distanțele minime.
- **performanța indexării:** cum evaluăm rezultatele indexării ?  
→ subiectiv (studii bazate pe utilizator), obiectiv (măsuri numerice, precision, recall, etc.).

## Paradigme

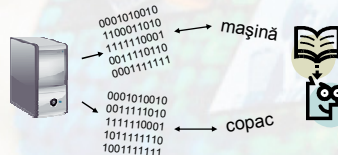
[C. Snoek, A. Smeulders, Tutorial on Video Search Engines, 2010]

- > există o multitudine de reprezentări ale aceluiaș obiect: iluminare diferită, perspectivă, suprapunere obiecte, etc.



→ **sensory gap**

- > discrepanță între informațiile extrase în mod automat și semnificația semantică pe care le-o putem atribui acestora,



→ **semantic gap**

## Paradigme

[C. Snoek, A. Smeulders, Tutorial on Video Search Engines, 2010]

- > există o multitudine de lucruri în lumea înconjurătoare:



→ **model gap**

- > multitudine de "întrebări" ale aceluiași concept,

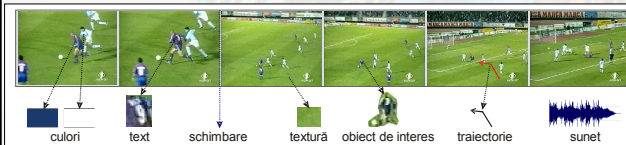


→ **interface gap**

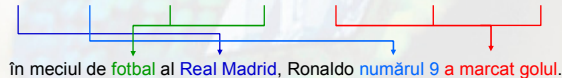
## Tendința sistemelor actuale

- > două mari categorii de sisteme:

- **adnotare sintactică (low-level)**



- **adnotare semantică (high-level)**



## Surse de informație

- > video = informație spațio-temporală:

- **culoare:** conținut vizual;
  - **textură:** materiale din scenă;
  - **formă:** obiecte, personaje;
  - **structura temporală:** ritm, acțiune;
  - **mișcare:** – locală sau globală;
  - **sunet:** – muzică, vorbire;
  - **alte surse:** – ex. culoare piele, text încrustat, etc.
- moștenite de la sistemele de indexare de imagini
- specifice video

## MPEG-7

Moving Picture Experts Group (2002 – prezent) = "Multimedia Content Description Interface": standard de descriere a conținutului multimedia, → al patrulea pas major: **indexarea conținutului video.**

[<http://mpeg.chiariglione.org/>]

### Obiective principale:

- > gestionarea informației audio-vizuale la nivel: audio, voce, video, imagini, grafică și 3D;
- > furnizarea de metode rapide de căutare, filtrare și identificare a conținutului multimedia ("căutarea multimedia la fel de simplă precum cea textuală");
- > **descriptori de conținut** (numerici, structuri, modele, colecții ...);
- > descrierea modului de relaționare a obiectelor pentru a forma scena;
- > independență fizică între partea de descriere și informația propriu-zisă (totuși sunt relaționate, multiplexate);

## MPEG-7

Descrierea conținutului multimedia (Multimedia Description Schemes)  
= structuri de metadata ce descriu și adnotează conținutul AV:

↳ textuale (XML ce poate fi citit → editare, căutare, filtrare)  
binare (comprimate → stocare sau transmisie)

**Descriptors (D):** descrierea "low-level" a proprietăților scenei audio-vizuale (culoare, textură, mișcare, ...) + proprietăți fizice (locație, timp, calitate, ...);  
→ a fi extrase automat!

**Description Schemes (DSs):** descrierea "high-level" a conținutului AV (regiuni, segmente, obiecte, evenimente, ...)  
→ necesită intervenție umană!

**Description Definition Language (DDL):** descrierea pe baza XML a relațiilor structurale între descriptori (creare și modificare de DSs și D);

**System tools:** binarizare, sincronizare, transport + stocare descriptori;

## MPEG-7: descriptori de conținut

- **color space:** RGB, YCbCr, HSV, HMM, ...  
(detalii M2 – Informația de Culoare)
- **color quantization:** reducere de culori,  
(detalii M2 – Informația de Culoare)
- **dominant color:** culori reprezentative, %, coerență spațială, varianță;
- **scalable color:** histogramă de culori HSV (scalabilă nr. culori, biți, ...);
- **color layout:** distribuția spațială a culorilor (foarte compactă, independent de rezoluție);
- **color structure:** informație despre conținutul de culoare (histogramă) dar și structură (prin element structurant);
- **GoP color:** extensia descriptorului de culoare scalabilă pentru secvențe de imagini (histograma medie, mediană și intersecție = minim "least common").

[T. Sikora, The MPEG-7 Visual Standard for Content Description - An Overview, 2001]

## MPEG-7: descriptori de conținut

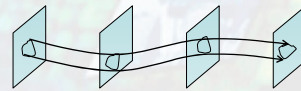
- **homogenous texture:** descriptori de textură pe bază de filtre Gabor (~ SVU);
- **texture browsing:** caracterizare perceptuală precum regularitate, direcționabilitate și granularitate;
- **edge histogram:** distribuția spațială a 4 tipuri de orientări de contur (vertical, orizontal, 45°, 135°);
- **region shape:** distribuția pixelilor într-un obiect 2D (obiecte complexe = regiuni disconecte, bazate pe 2D Angular Radial Transformation - transf. unitară);
- **contour shape:** proprietăți obiect pe baza formei conturului (bazate pe reprezentarea în Curvature Scale-Space – reprezentare 1D a conturului);



(exemple de obiecte similare)

## MPEG-7: descriptori de conținut

- **shape 3D:** reprezentare folosind proiecții 2-D ale diferitelor ipostaze ale obiectului 3D;
- **region locator:** localizarea regiunilor la nivel de cadru prin specificarea unui dreptunghi sau poligon (scalabil);
- **spatio-temporal locator:** localizarea spațio-temporală a regiunilor (mișcare obiecte); poate descrie și obiecte ce nu sunt conectate spațial;



## MPEG-7: descriptori de conținut

- **camera motion:** parametri de mișcare globală la un anumit moment temporal;  
(detalii M4 – Informația de Mișcare)
- **motion trajectory:** traiectoria deplasării obiectelor în timp (localizare temporală și spațială);
- **parametric motion:** descrierea mișcării obiectelor pe baza modelelor parametrice (afine);  
(detalii M4 – Informația de Mișcare)
- **motion activity:** ~nivelul de acțiune (lent, rapid, acțiune), bazată pe dispersia amplitudinii vectorilor de mișcare (pot avea o reprezentare spațială și temporală);

> Sfârșit M1. Informația video