Concepte și aplicații în Vederea Artificială

Bogdan Alexe
bogdan.alexe@fmi.unibuc.ro

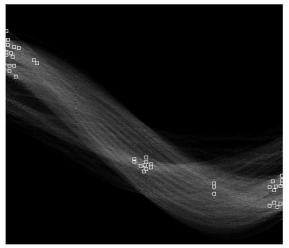
Radu Ionescu
radu.ionescu@fmi.unibuc.ro

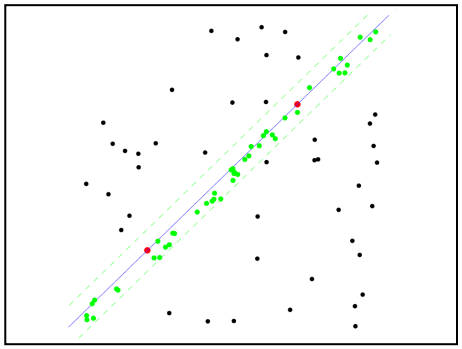
Curs opțional

semestrul I, 2023-2024

Cursul trecut

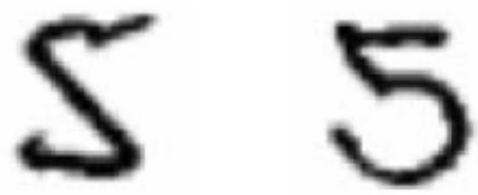
- Aplicație: detectarea liniilor cu
  - transformata Hough
  - RANSAC



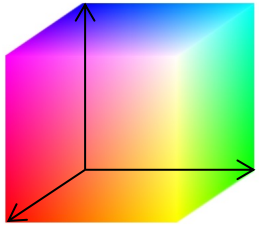


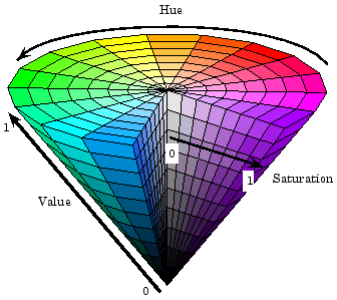
Cursul de azi

- Compararea contururilor

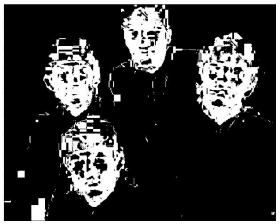


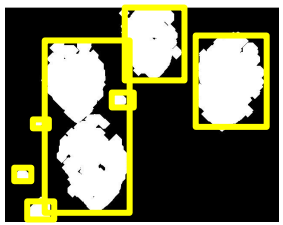
- Spațiile de culori RGB și HSV





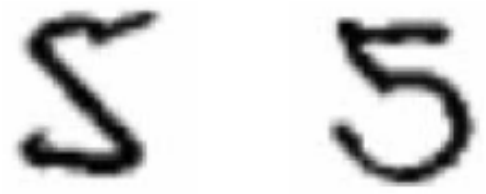
- Imagini binare
  - aplicarea unui prag
  - operatori morfologici
  - componente conexe
  - proprietăți





- Recunoaşterea claselor de obiecte
  - privire de ansamblu
  - ce este o clasă de obiecte?

Compararea contururilor



Exemple de două imagini cu cifre scrise de mână.
Comparând pixel cu pixel, cele două imagini sunt foarte diferite.
Totuşi, în termeni de formă, cele două imagini sunt similare.

Slide adaptat după S. Belongie

Distanța Chamfer

- Distanța medie dintre punctele unui ***template T*** şi o mulțime de puncte (e.g. ***puncte cu intensitate = 0***, puncte cu anumite caracteristici) dintr-o imagine I.

$$d_{chamfer}(T, I) = \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

template

$I =$ mulțime de puncte dintr-o imagine

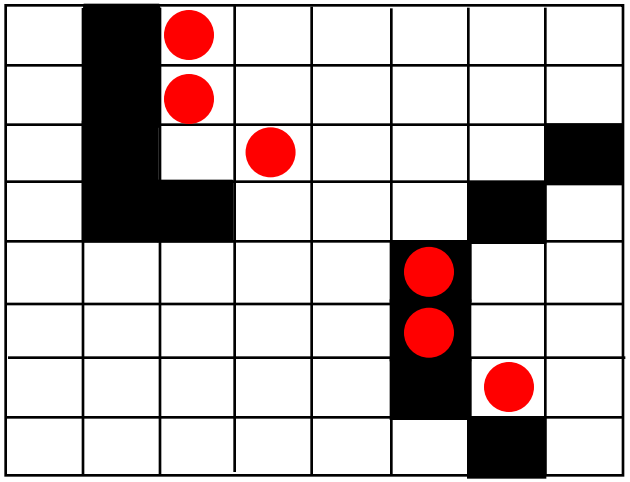
$T =$ mulțime de puncte ale unui ***template***

$d_I(t) =$ distanța minimă dintre ***punctul t*** şi un punct din I (Manhattan, Euclidiană)

Slide adaptat după K. Grauman

Distanța Chamfer

Puncte din I (2D)

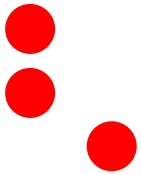


Distanța Manhattan:

$\frac{1}{3}(1+1+2) = 1.33$

$\frac{1}{3}(0+0+1) = 0.33$

T



$$d_{chamfer}(T, I) = \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

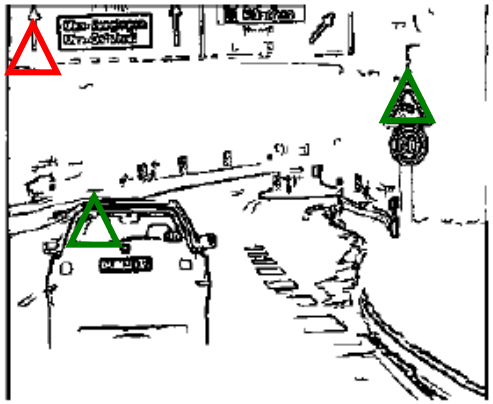
Slide adaptat după K. Grauman

Distanța Chamfer

- distanța medie până la cel mai apropiat punct din ***I***

$$d_{chamfer}(T, I) = \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

Cum diferă distanța Chamfer de filtrarea cu o mască de forma lui T?



Imagine cu muchii

Răspunsul variază mult mai puțin abrupt decât în cazul filtrării! (dacă mă mut la stânga/dreapta cu un pixel am valori apropiate)

Implementare naivă – complexitate?

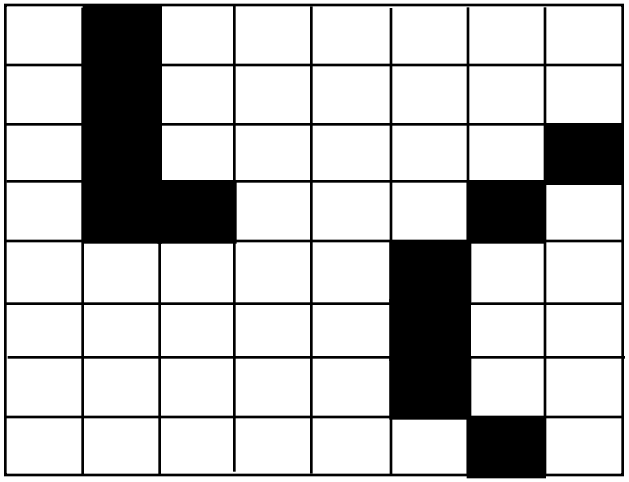
$|T| * |I|^2$ operații

Minimele locale ale funcției care calculează distanța Chamfer pentru fiecare fereastră

Slide adaptat după K. Grauman

Transformata Distanță (distance transform)

Puncte din I (2D)



Transformata Distanță

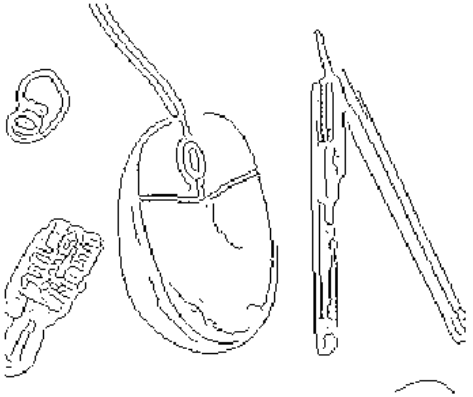
1	0	1	2	3	4	3	2
1	0	1	2	3	3	2	1
1	0	1	2	3	2	1	0
1	0	0	1	2	1	0	1
2	1	1	2	1	0	1	2
3	2	2	2	1	0	1	2
4	3	3	2	1	0	1	2
5	4	4	3	2	1	0	1

Transformata Distanță (TD) este o funcție care pentru fiecare pixel *p* asignează un număr pozitiv *TD(p)* corespunzând distanței de la *p* la cel mai apropiat punct din mulțimea *I*

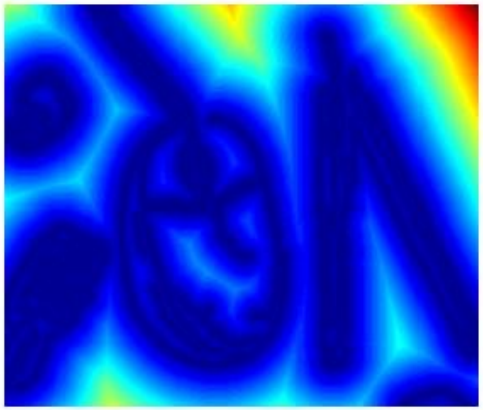
Slide adaptat după Y. Boykov

Transformata Distanță

imagine inițială



muchii



transformata distanță

Calculăm o singură dată TD(), apoi citim din acest tabel distanța la cel mai apropiat punct de pe muchie.

Valoarea la (x,y) indică cât de departe (x,y) este de cel mai apropiat punct de pe muchie
albastru – valori mici,
roșu – valori mari

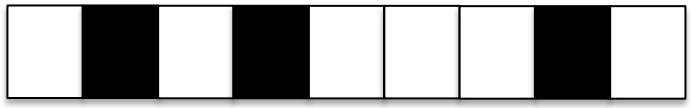
OpenCV: cv.distanceTransform

Slide adaptat după K. Grauman

Transformata Distanță – 1D

Două treceri de complexitate O(n)

Exemplu

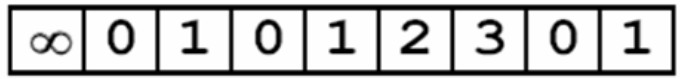


1. Inițializare

```
for j = 1:n
        TD(j) = { 0,   dacă j este în I
                { ∞, altfel
end
```

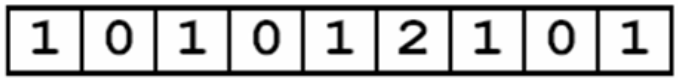
2. Trecere înainte (vecin stânga)

```
for j = 2:n
        TD(j) = min(TD(j),TD(j-1)+1)
end
```



3. Trecere înapoi (vecin dreapta)

```
for j = n-1:-1:1
        TD(j) = min(TD(j),TD(j+1)+1)
end
```

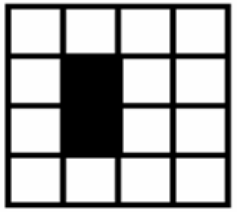


Slide adaptat după D. Huttenlocher

Transformata Distanță – 2D

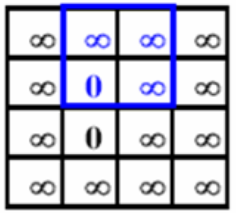
Analog cu 1D

Exemplu



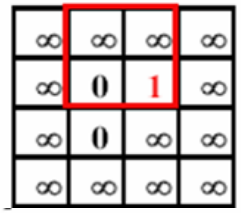
1. Inițializare

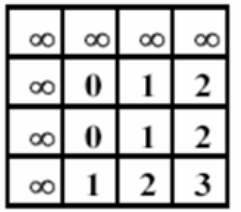
$$TD(i,j) = \begin{cases} 0, & \text{dacă } (i,j) \text{ este în } I \\ \infty, & \text{altfel} \end{cases}$$



2. Trecere înainte (vecin stânga, sus)

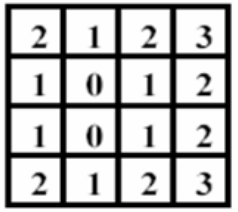
$$TD(i,j) = \min(TD(i,j), TD(i,j-1)+1, TD(i-1,j)+1)$$





3. Trecere înapoi (vecin dreapta, jos)

$$TD(i,j) = \min(TD(i,j), TD(i,j+1)+1, TD(i+1,j)+1)$$



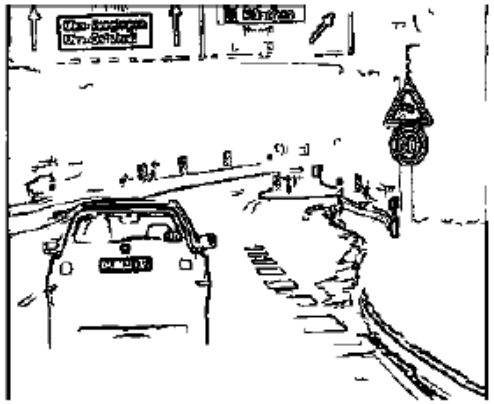
Slide adaptat după D. Huttenlocher

Distanța Chamfer

- distanța medie până la cel mai apropiat punct din ***I***

$$D_{chamfer} = (T, I) = \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

Complexitate implementare naivă:
$|T| * |I|^2$ operații
Complexitate transformata distanță:
$|T| * |I|$ operații



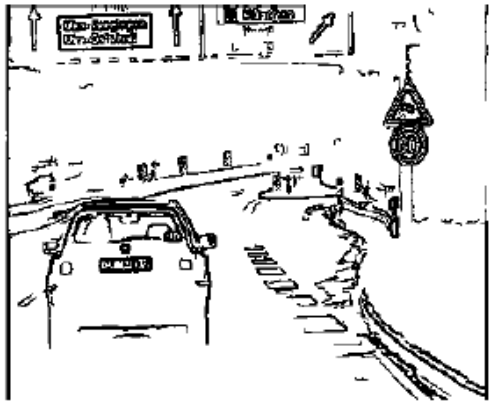
Imagine cu muchii

transformata distanță
alb – distanță mare,
negru - distanță mică

Slide adaptat după K. Grauman

Recunoaşterea semnelor de circulaţie

Recunoaștere = detectare + clasificare
Detectare = localizăm conturul în imagine (pe baza de template: triunghi, cerc)
Clasificare = analizăm pixelii din interiorul conturului și clasificăm semnul de circulație într-o clasă (limitare de viteză 60 km, limitare de viteză 80 km, semafor, etc)



Imagine cu muchii

transformata distanță
alb – distanță mare,
negru - distanță mică

Slide adaptat după K. Grauman

Recunoaşterea semnelor de circulaţie

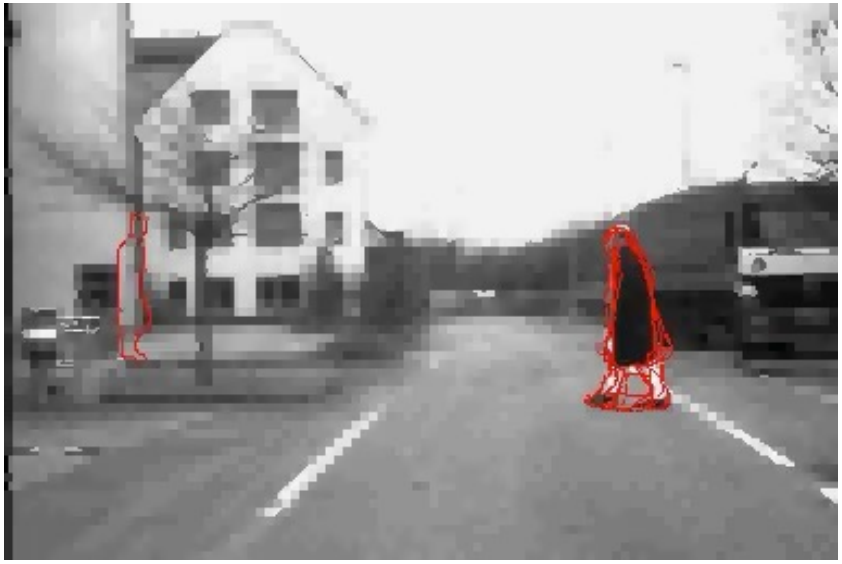
condiţii de zi

condiţii de noapte

Video de la D. Gavrila, DAGM 1999

Detectarea pietonilor

Detectare pietonilor = localizăm contururi în imagine asemănătoare cu contururi de pietoni dintr-o mulțime de template-uri



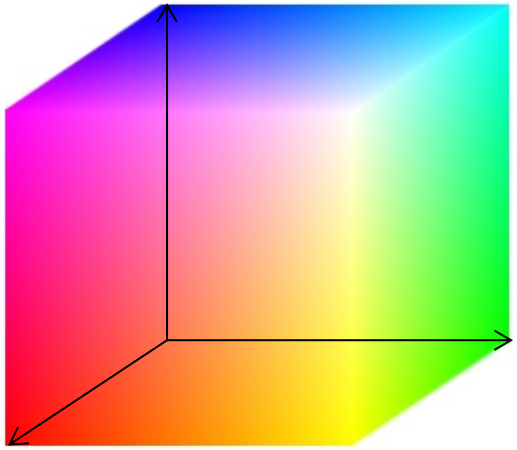
http://www.gavrila.net/Research/Chamfer_System/chamfer_system.html

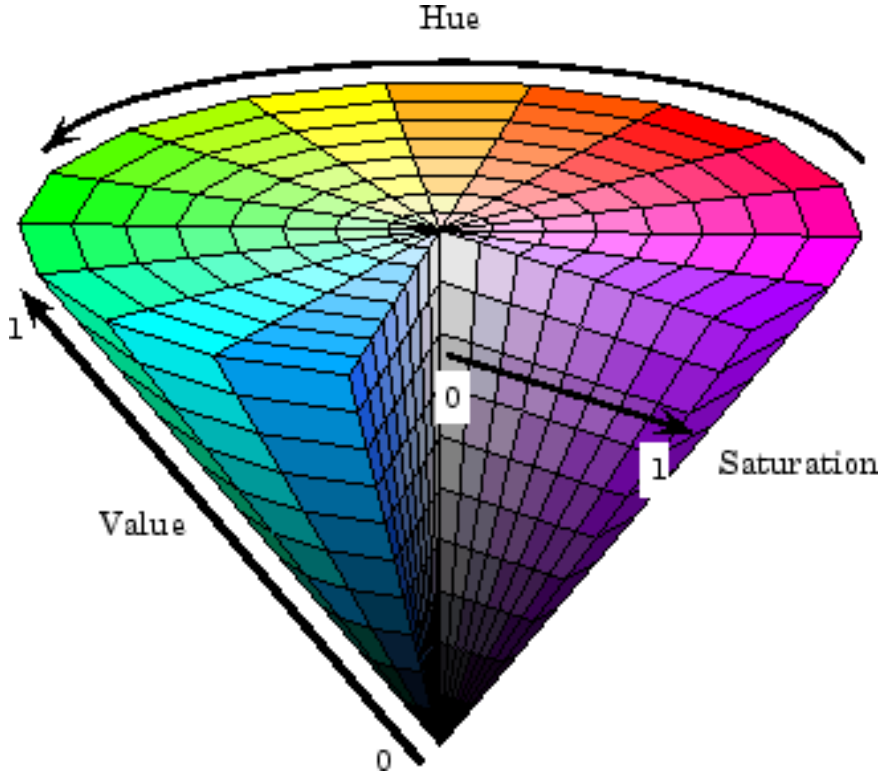
Distanța Chamfer: proprietăți

- Avantaje
  - simplă de implementat
  - foarte rapidă
  - tolerantă la forme care diferă puțin de template
- Dezavantaje
  - senzitivă la mărime (scale) şi rotație
  - număr mare de template-uri pentru forme diverse ca mărime şi ca rotație
  - detecții false în regiuni cu multe muchii

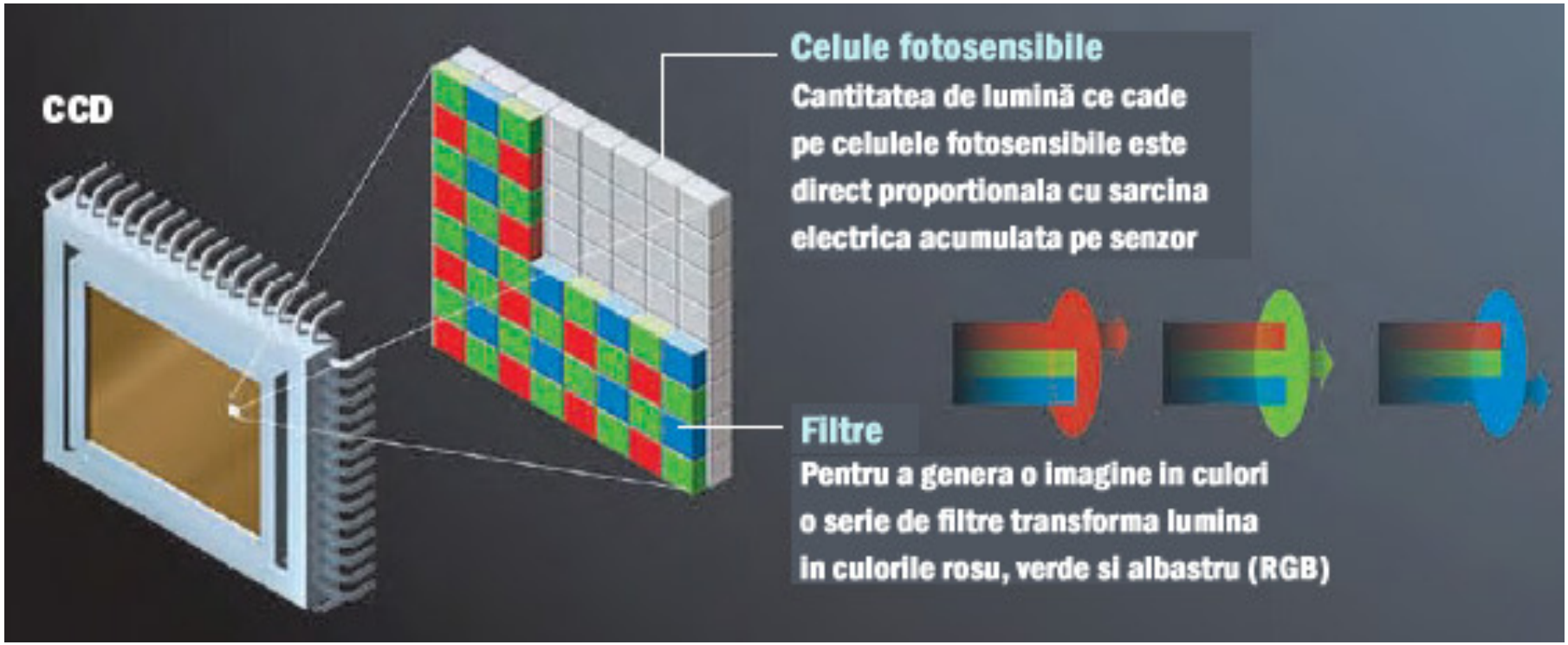
Slide adaptat după K. Grauman

Spațiile culori RGB și HSV





Imagini digitale color



Senzor de imagine Filtru Bayer

Tipuri de imagini digitale

Binare

Grayscale
(tonuri de gri)

Color

Luminozitate	**negru, alb**	**tonuri de gri**	**R G B**
Valori	{**0**,**1**}	{**0**, … , **255**}	{0, …,255}^{3}
Culori	**negru - 0**, **alb - 1**	**negru - 0**, **gri - 128**, **alb - 255**	**(255,0,0)**, **(0,255,0)**, **(0,0,255)**, **(0,0,0)**,**(255,255,255)**, **(255,255,0)**, **(255,125,0)**, **(0,255,255)**, **(255,0,255)**
Memorie/pixel	1 bit/pixel	8 biți/pixel	24 biți/pixel

RGB2GRAY

Color

Grayscale
(tonuri de gri)

$$gray = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

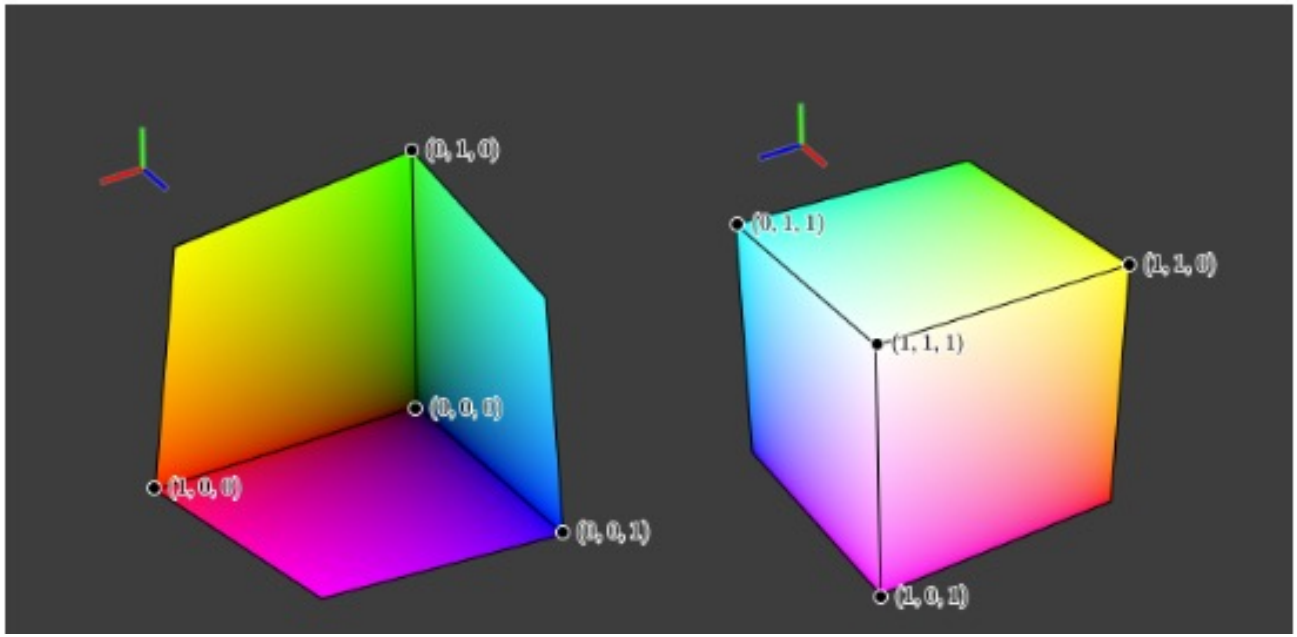
coeficienți determinați pe
baza percepției vizuale umane

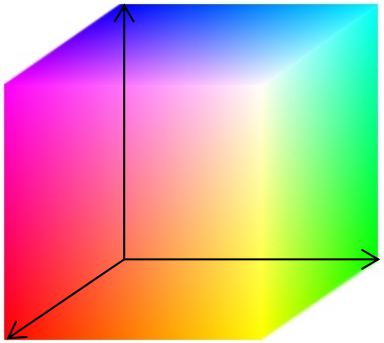
https://en.wikipedia.org/wiki/Color_vision

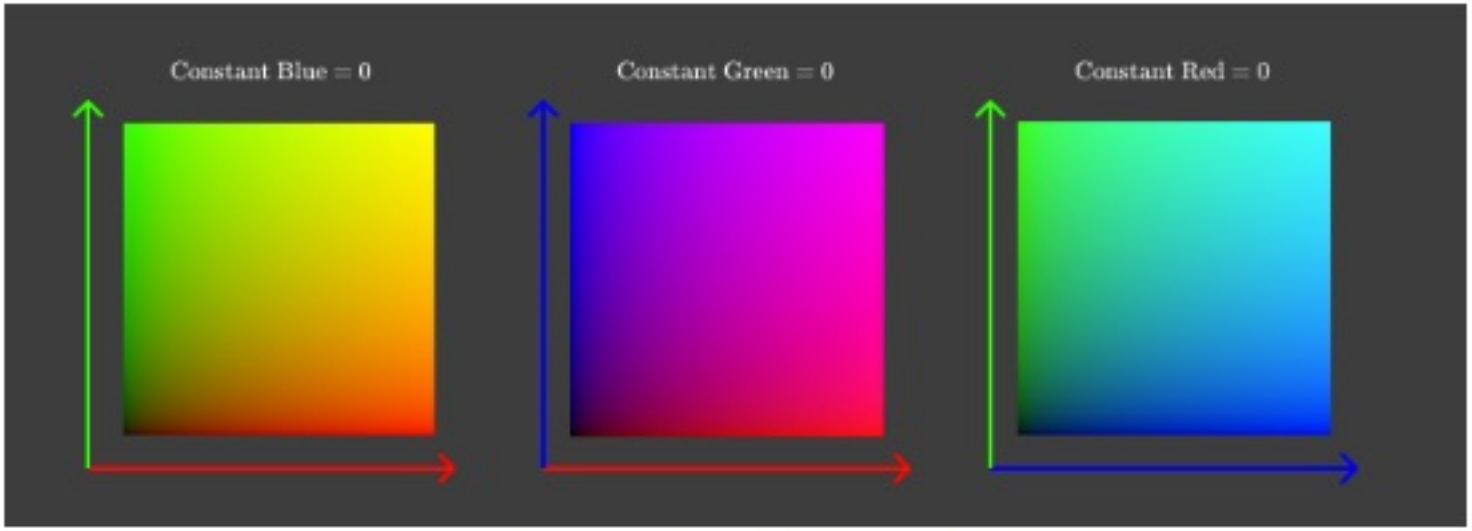
Acest text este mai greu de citit

Acest text este mai usor de citit

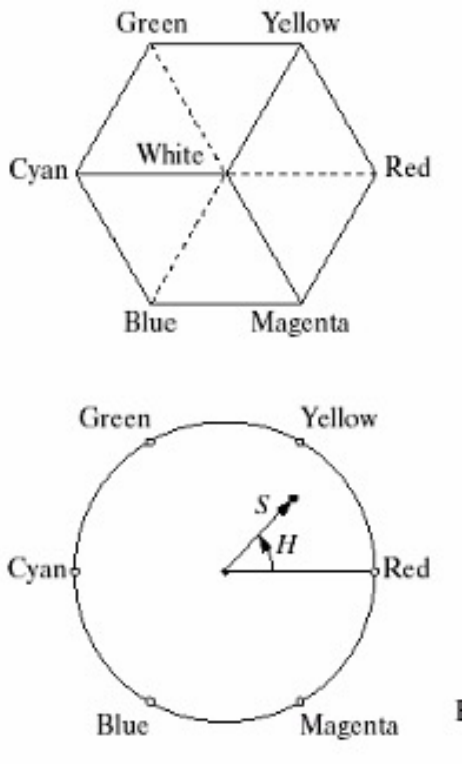
Spațiul liniar de culori - RGB

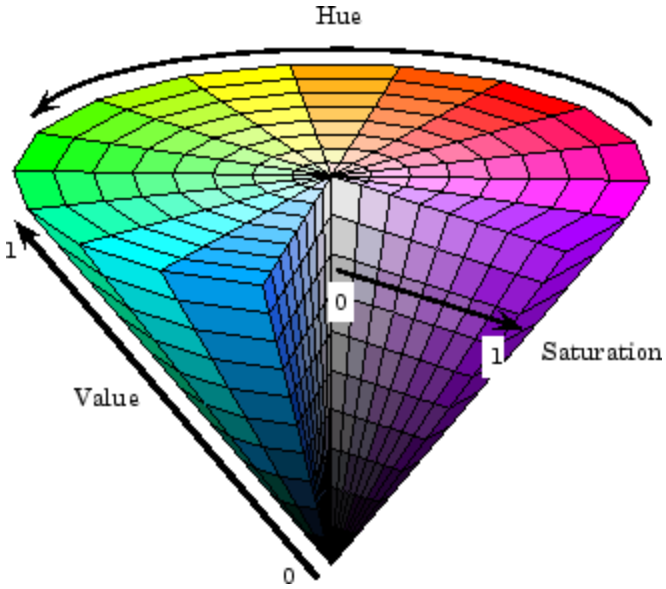


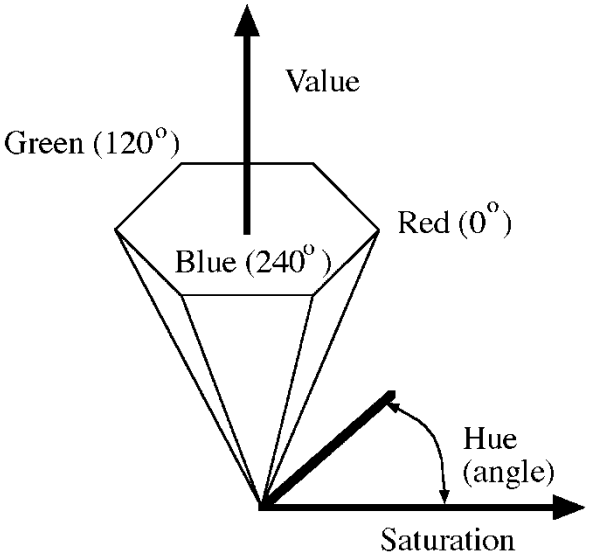




Spațiu de culori neliniar HSV

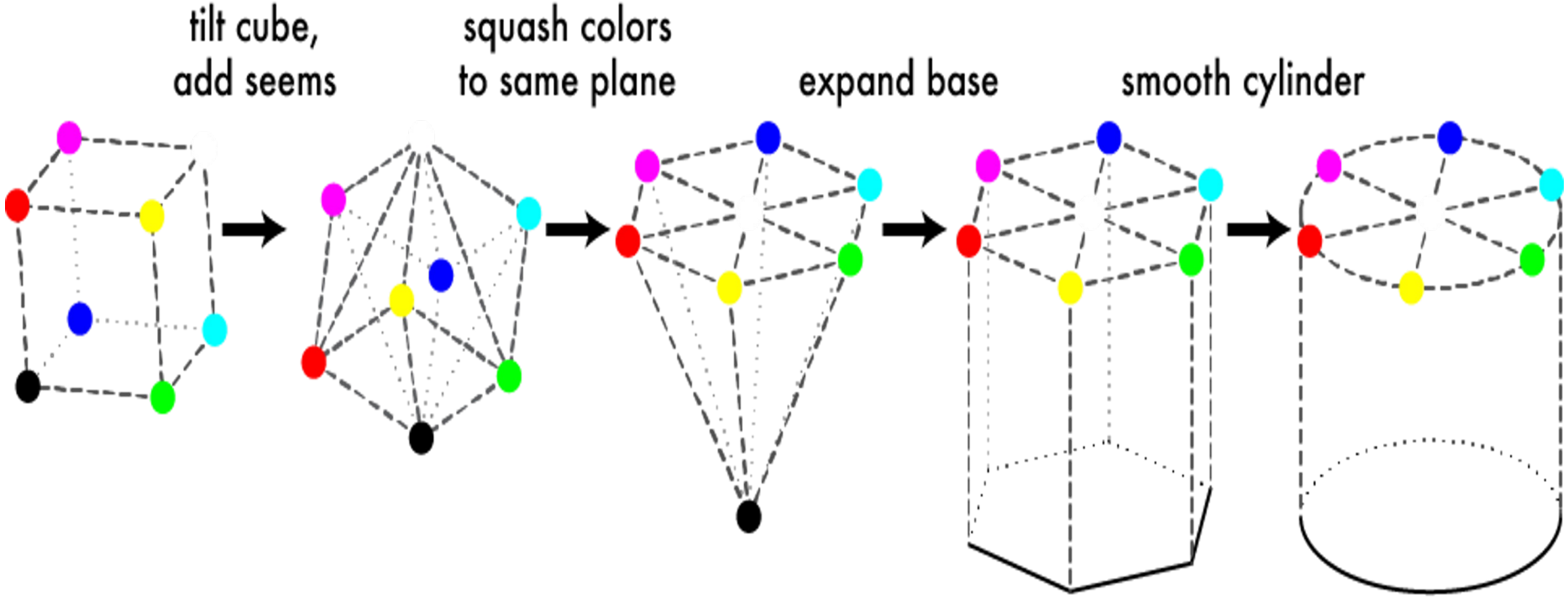


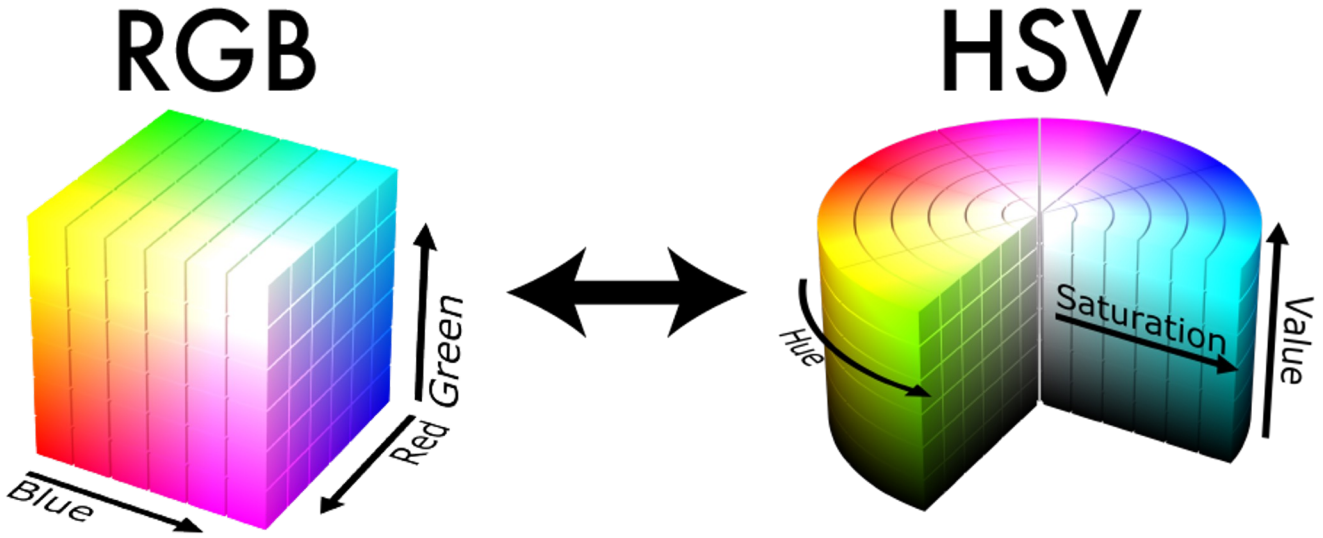




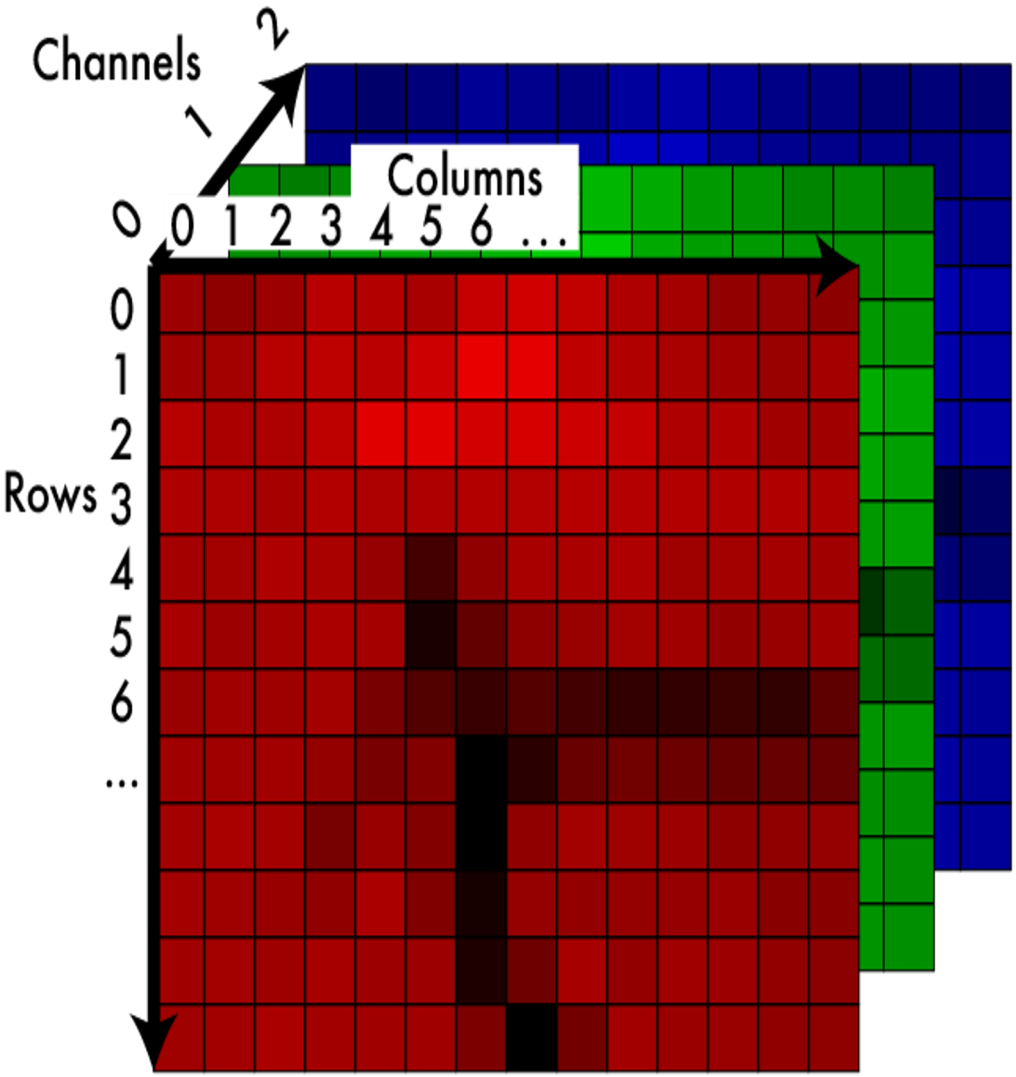
- definit pe baza modului în care oamenii percep culorile
  - Hue = nuanță: ce culoare
  - Saturation = saturație: cât de multă culoare
  - Value = valoare: cât de multă luminozitate

RGB2HSV

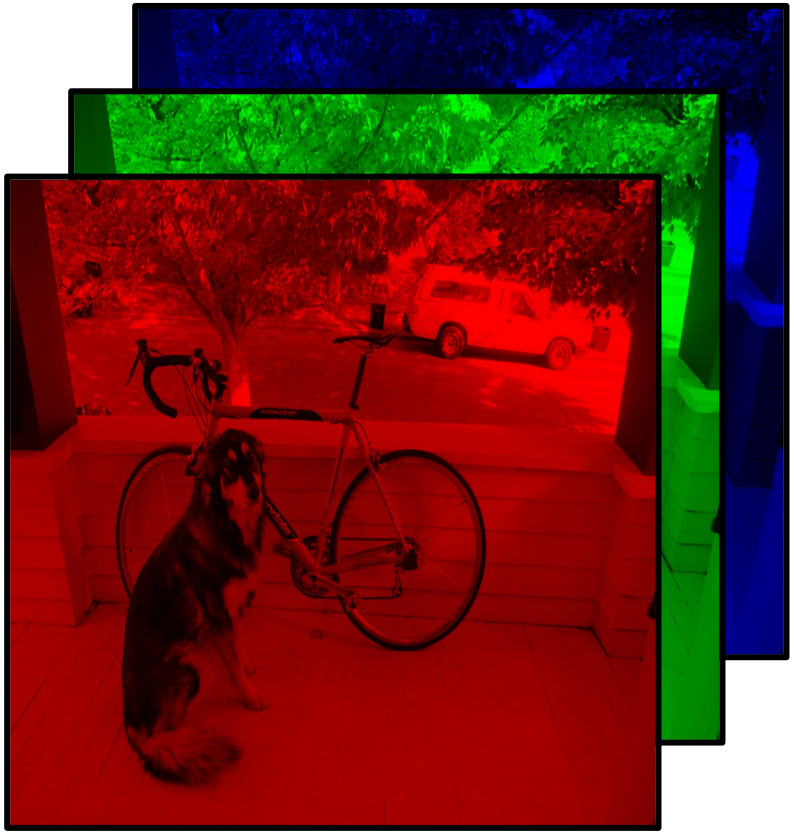

tilt cube,
add seems
squash colors
to same plane
expand base
smooth cylinder

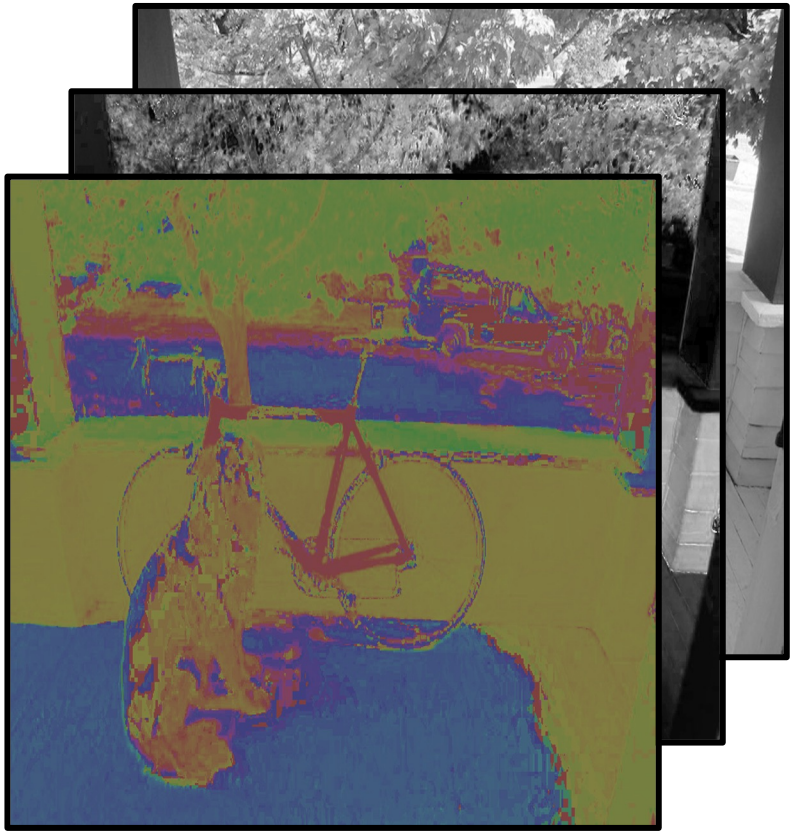

RGB
HSV
Blue
Red
Green
Hue
Saturation
Value

Imagine în spațiul de culori RGB



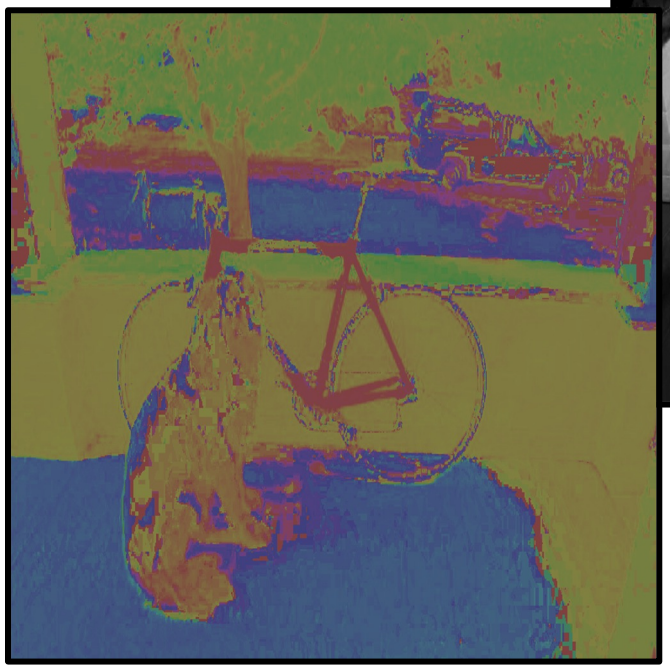
RGB vs HSV

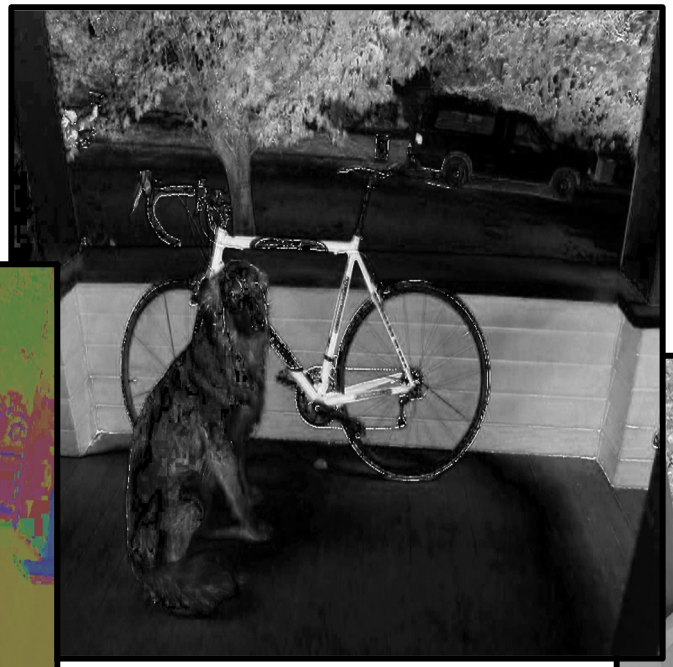




Saturation

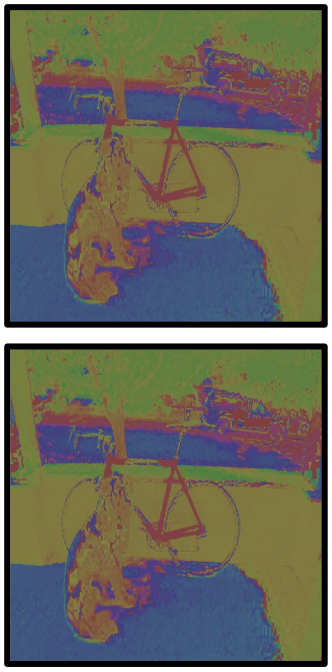
Hue





Value

Creștem saturația = culori mai intense

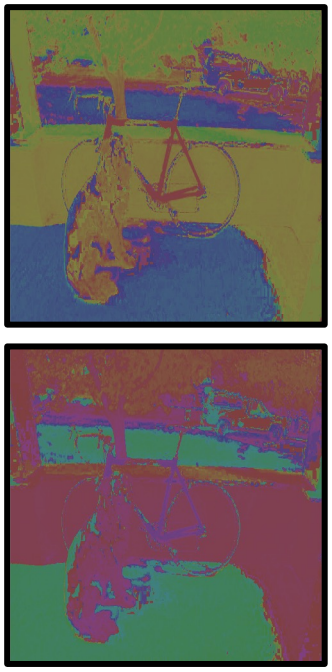


2x

Creștem valoarea = imagine cu luminozitate mai mare

2x

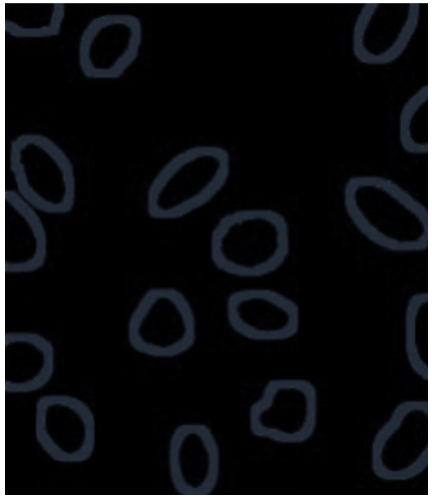
Modificăm nuanța = modificăm culorile

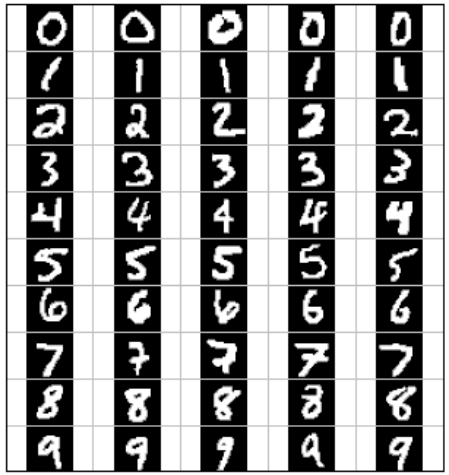


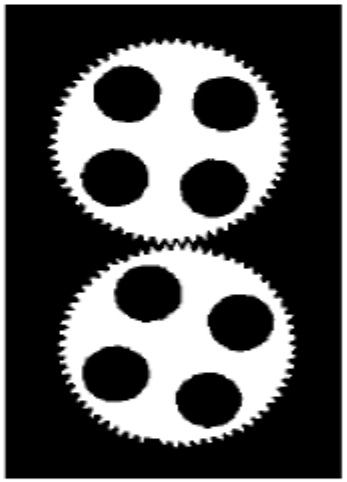
-.2

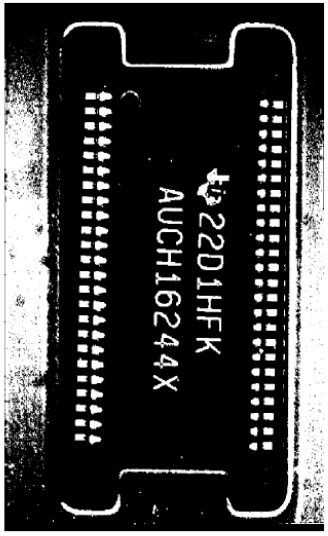
Imagini binare

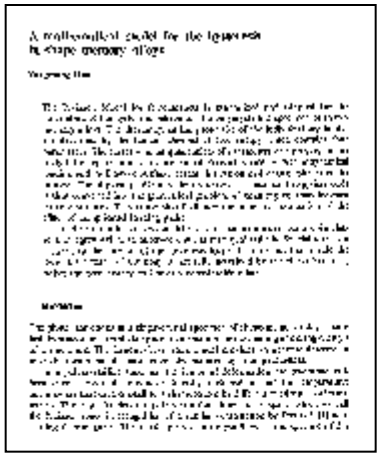
Imagini binare







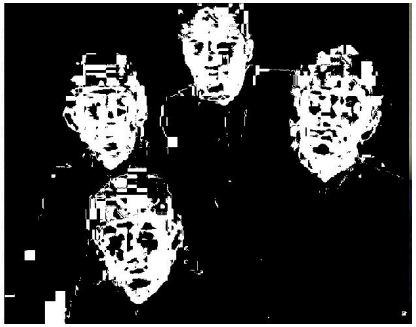


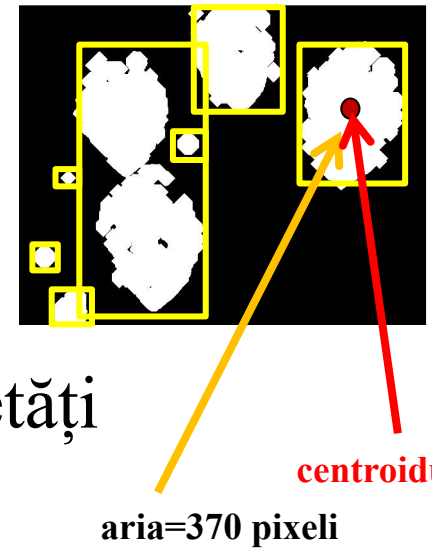


Slide adaptat după K. Grauman

Analiza imaginilor binare: paşi de urmat

- Convertim imaginea în imagine binară
  - aplicarea unui prag
- Curățăm imaginea obținută
  - operatori morfologici (dilatare, eroziune)
- Extragem regiuni separate
  - componente conexe
- Caracterizăm regiunile cu anumite proprietăți



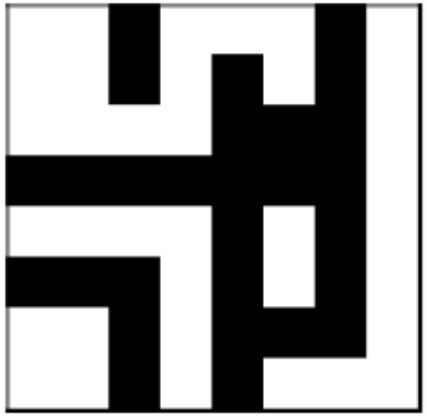


Slide adaptat după K. Grauman

Imagini binare

- Pixelii iau două valori
  - foreground – valoare 1, pixeli albi, obiectul de interes
  - background – valoare 0, pixeli negri, mediul ambient

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



Slide adaptat după K. Grauman

Aplicarea unui prag (thresholding)

- Imagine de intensitate (tonuri de gri) → imagine binară
- Folositoare dacă ceea ce căutăm în imagine are o distribuție a intensităților diferită de background

$$F_T(i,j) = \begin{cases} 1, & \text{dacă } F(i,j) \geq T \\ 0, & \text{altfel} \end{cases}$$

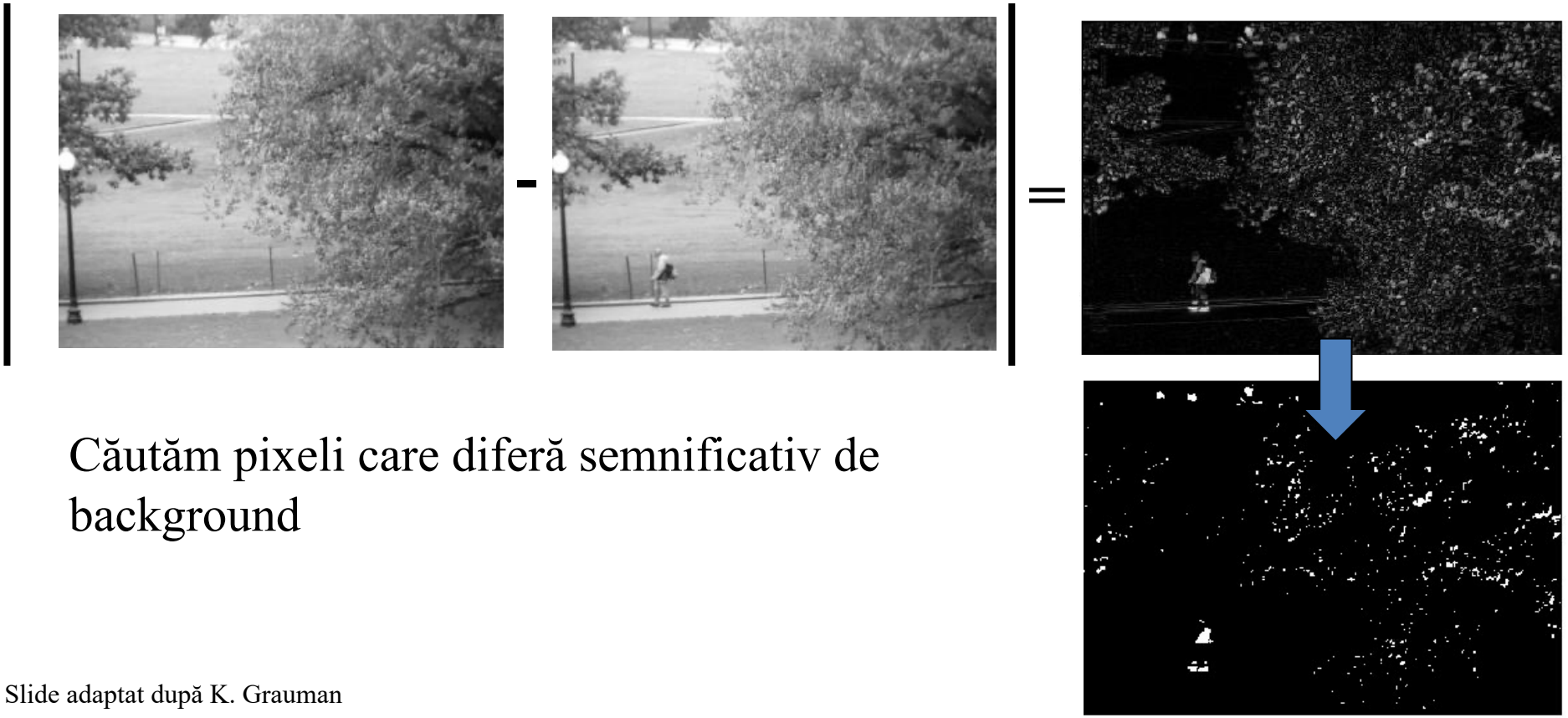
$$F_{T1,T2}(i,j) = \begin{cases} 1, & \text{dacă } T_1 \leq F(i,j) \leq T_2 \\ 0, & \text{altfel} \end{cases}$$

Slide adaptat după K. Grauman

Aplicarea unui prag

- Imagine de intensitate (tonuri de gri) → imagine binară

Exemplu: extragerea background-ului



Căutăm pixeli care diferă semnificativ de background

Slide adaptat după K. Grauman

Aplicarea unui prag

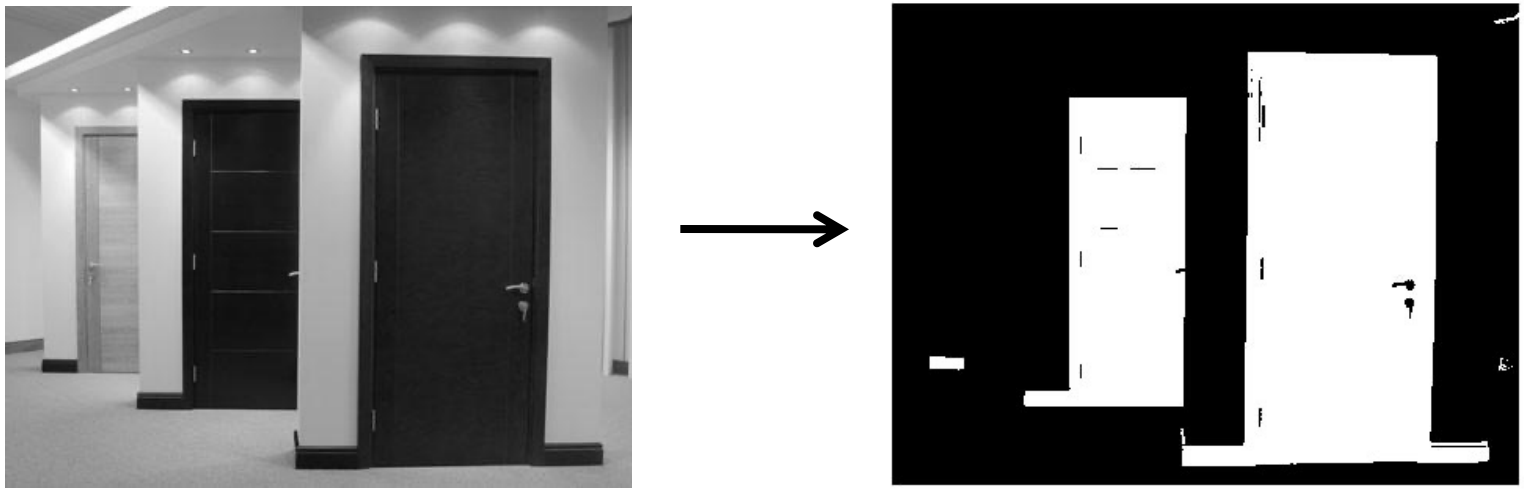
- Imagine de intensitate (tonuri de gri) → imagine binară

Exemplu: extragerea background-ului

Aplicarea unui prag

- Imagine de intensitate (tonuri de gri) → imagine binară

Exemplu: detectarea pe bază de intensitate



fg_pix = find(im < 65);

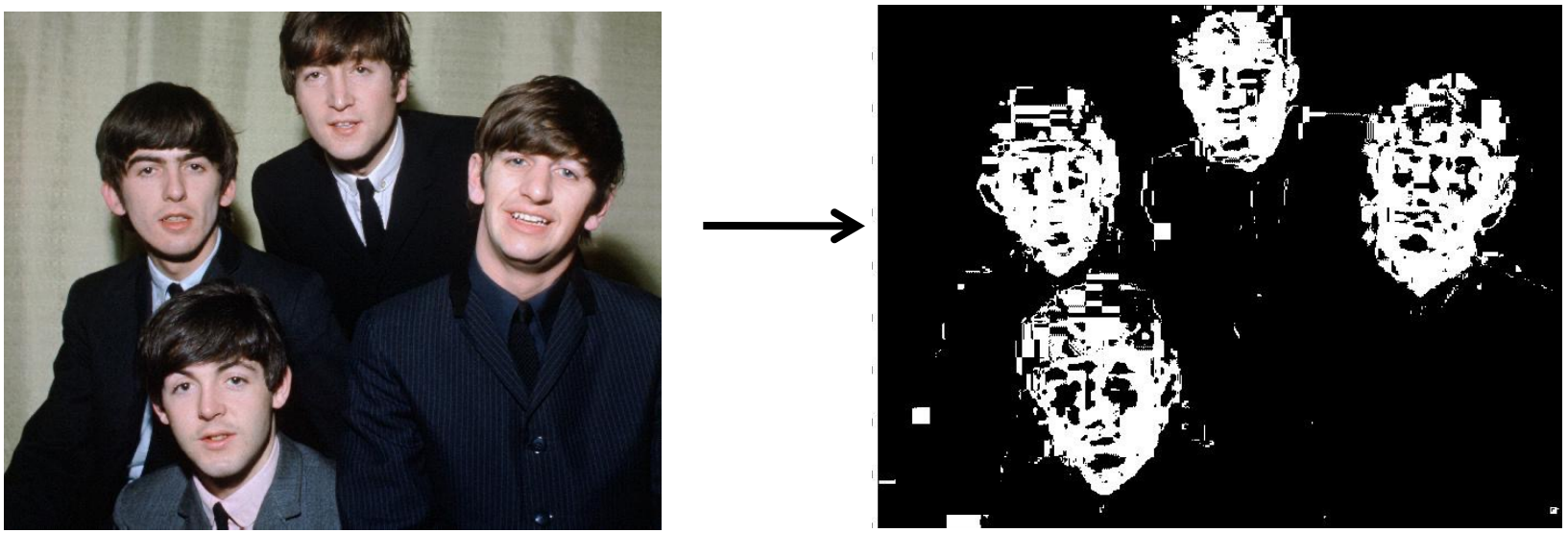
Căutăm pixelii cu intensitatea scăzută

Slide adaptat după K. Grauman

Aplicarea unui prag

- Imagine de intensitate (tonuri de gri) → imagine binară

Exemplu: detectarea pe bază de culori

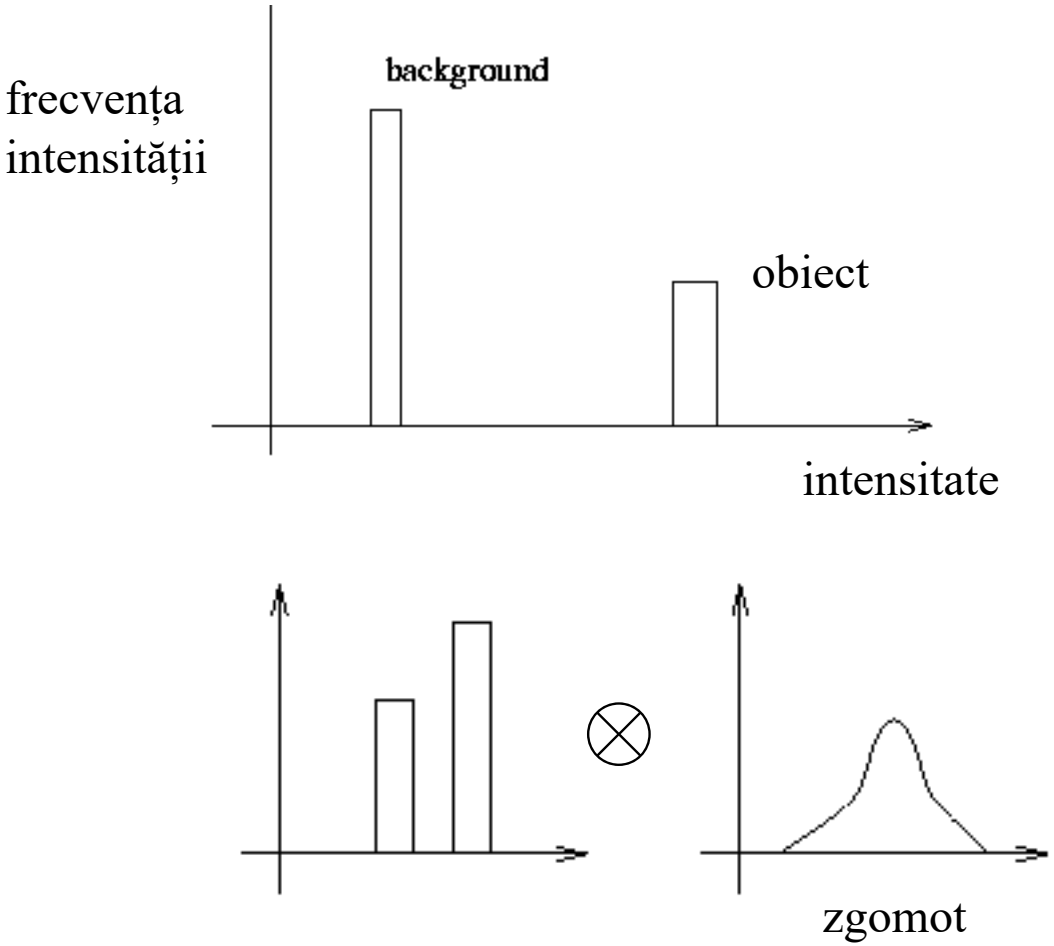


fg_pix = find(hue > t1 & hue < t2);

Căutăm pixeli cu o anumită nuanță a culorii (culoarea pielii).

Slide adaptat după K. Grauman

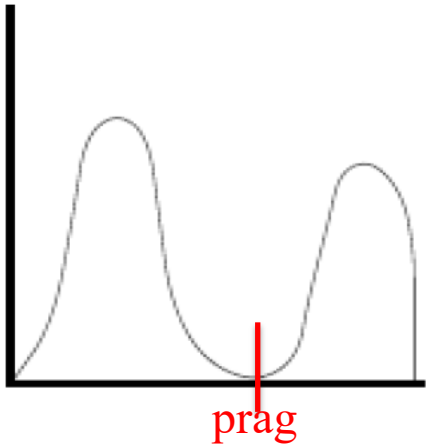
Caz ideal: histograme de intensitate bimodale



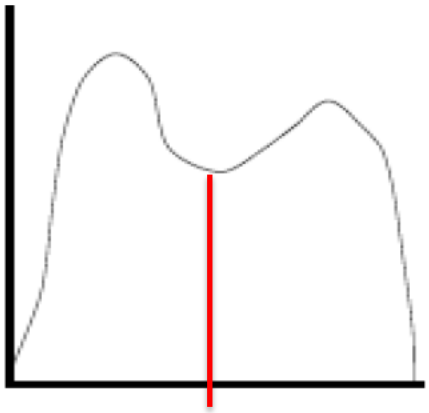
cazul ideal: histogramă bi-modală

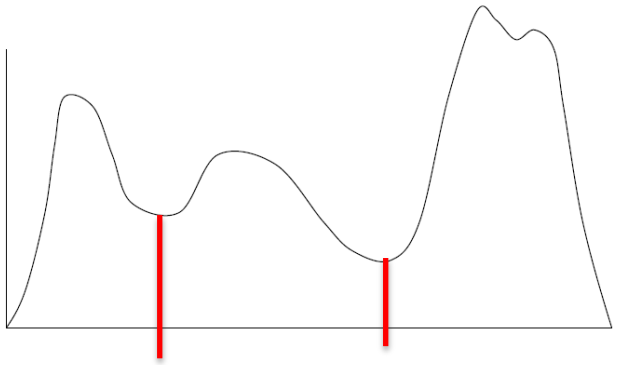
obiect luminos într-un background întunecat

Cazuri mai puțin ideale



două distribuții distincte



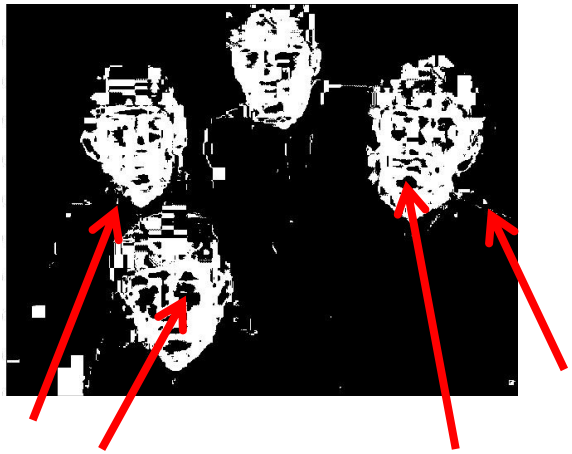


distribuții ce se suprapun

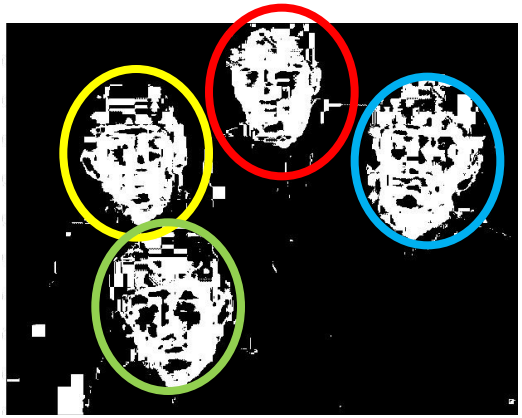
Slide adaptat după Shapiro

Probleme

- Ce facem cu imaginile binare rezultate ca cea din dreapta?
  - găuri
  - fragmente foarte mici, izolate



- Cum delimităm regiunile de interes?
  - numărăm obiecte
  - calculăm diverse proprietăți pentru un obiect



Slide adaptat după K. Grauman

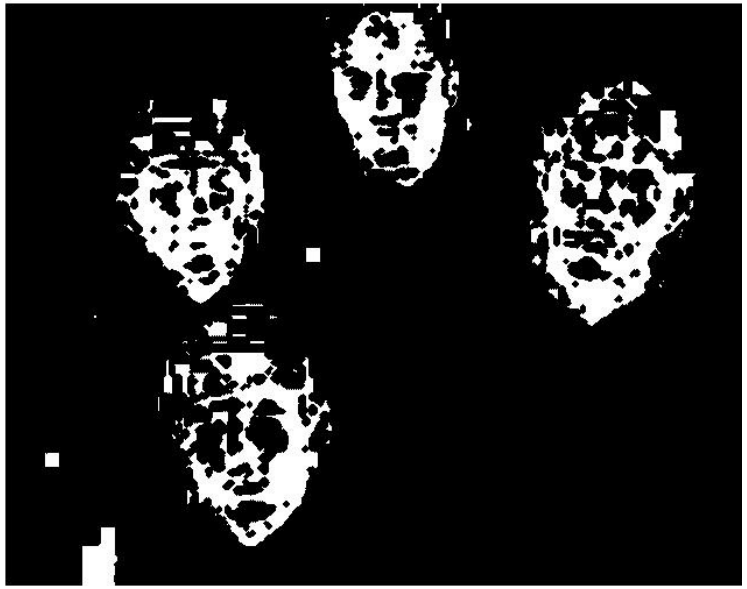
Operatori morfologici

- Schimbă forma regiunilor de foreground folosind operații de intersecție/uniune dintre un element structural şi o imagine binară.
- Eliminăm efectele nedorite rezultate în urma aplicării unui prag
- Operatorii de bază:
  - Dilatare
  - Eroziune

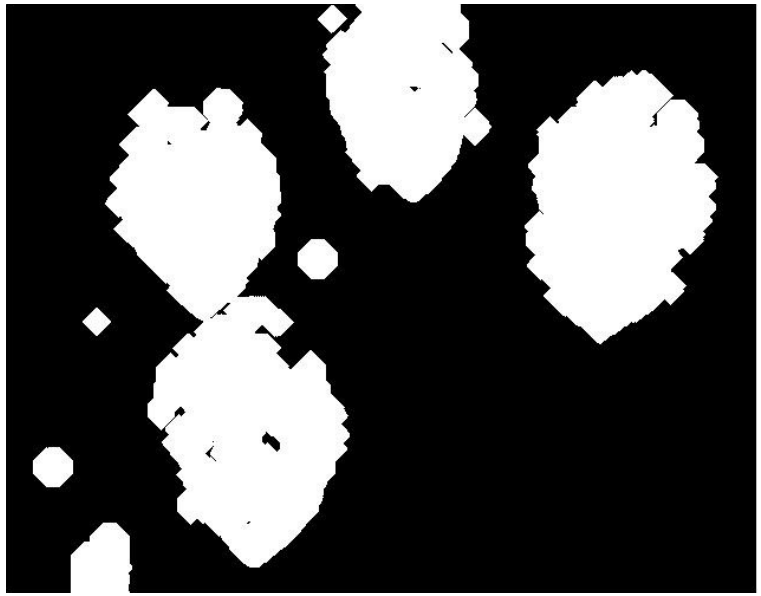
Slide adaptat după K. Grauman

Dilatare

- Măreşte componentele conexe
- Adaugă pixeli albi
- Umple găurile



Înainte de dilatare



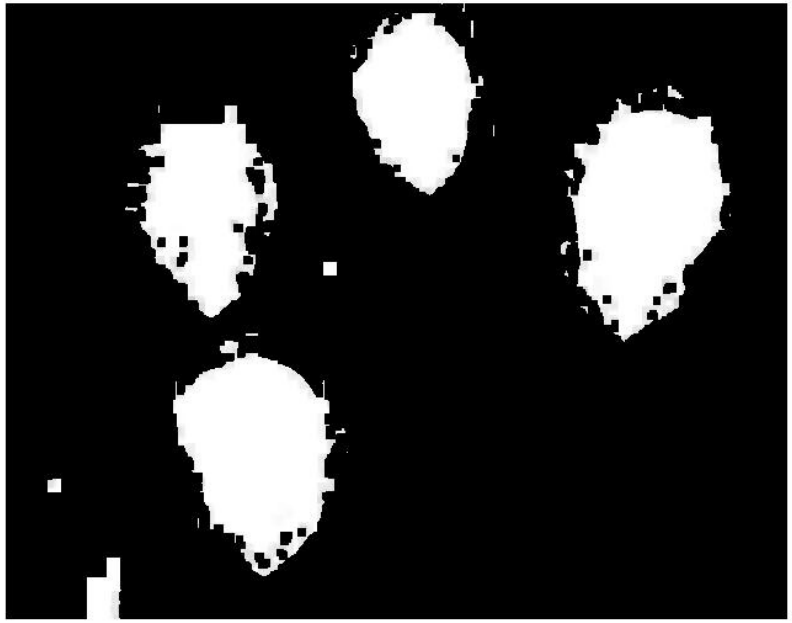
După dilatare

Slide adaptat după K. Grauman

Eroziune

- Micşorează componentele conexe
- Elimină pixeli albi
- Elimină fragmentele mici, zgomot

Înainte de eroziune

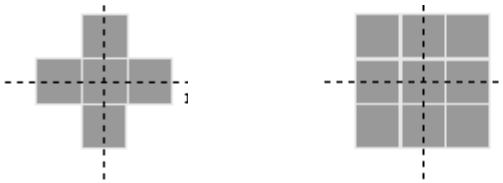


După eroziune

Slide adaptat după K. Grauman

Elemente structurale

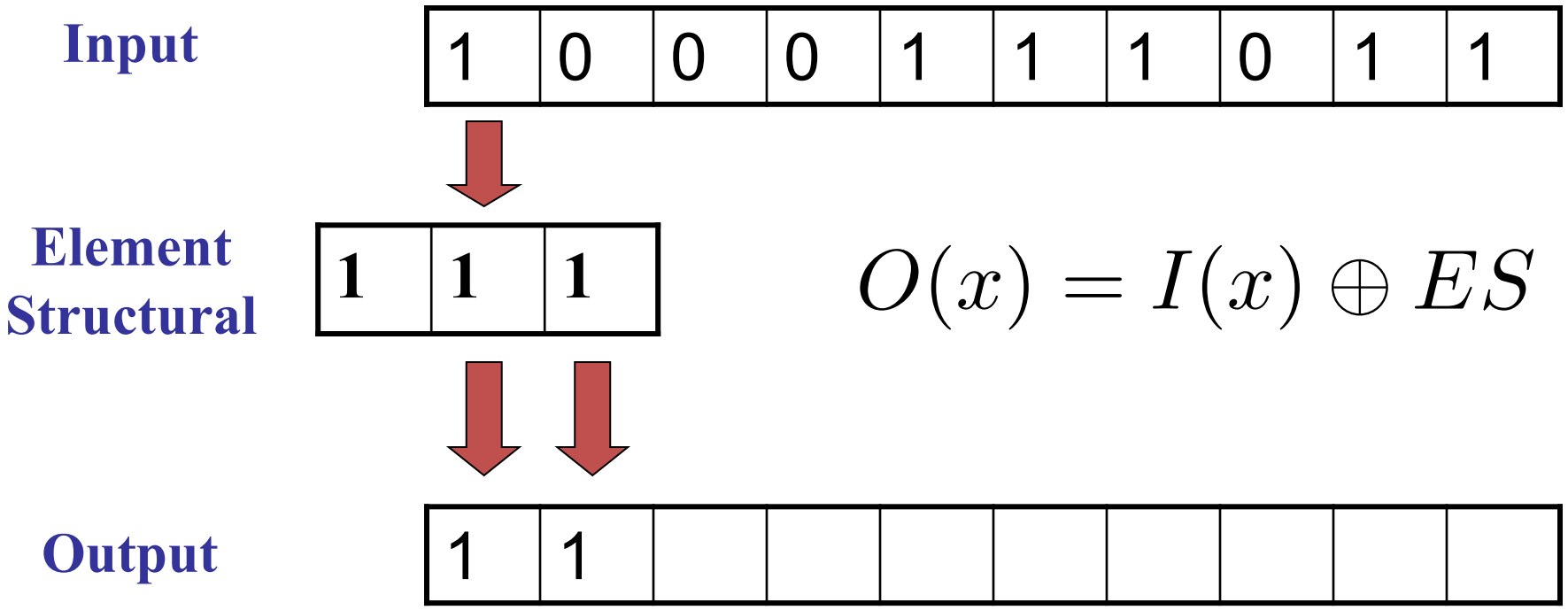
- **Măşti** de diverse forme şi dimensiuni folosite de operatori morfologici:



- Suprapunem masca peste imaginea binară (un fel de corelație)

Slide adaptat după K. Grauman

Dilatare – exemplu (1D)

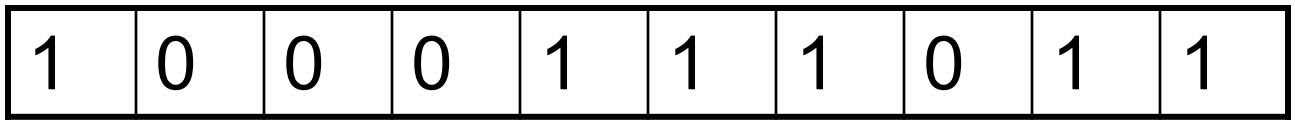


$$O(x) = I(x) \oplus ES$$

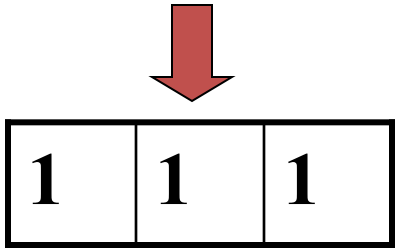
Slide adaptat după T. Moeslund

Dilatare – exemplu

Input



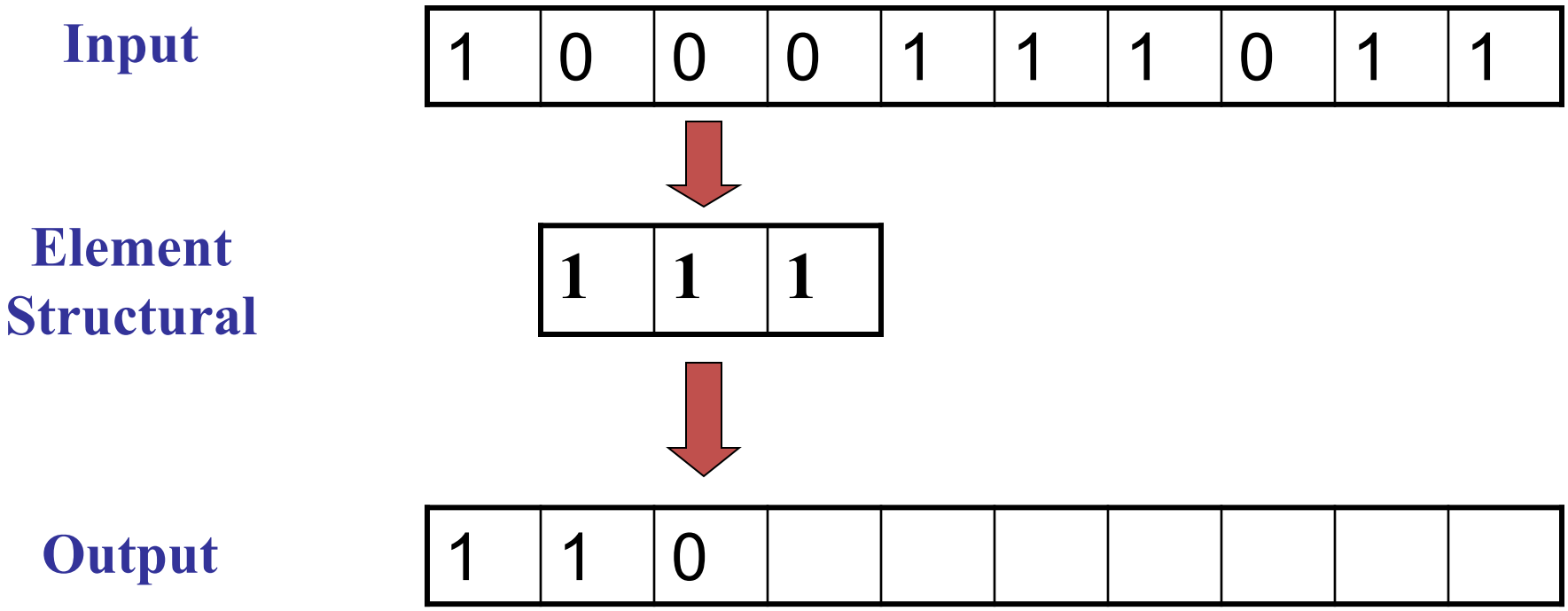
Element Structural



Output

Slide adaptat după T. Moeslund

Dilatare – exemplu



Slide adaptat după T. Moeslund

Dilatare – exemplu

Input

1	0	0	0	1	1	1	0	1	1

Element Structural

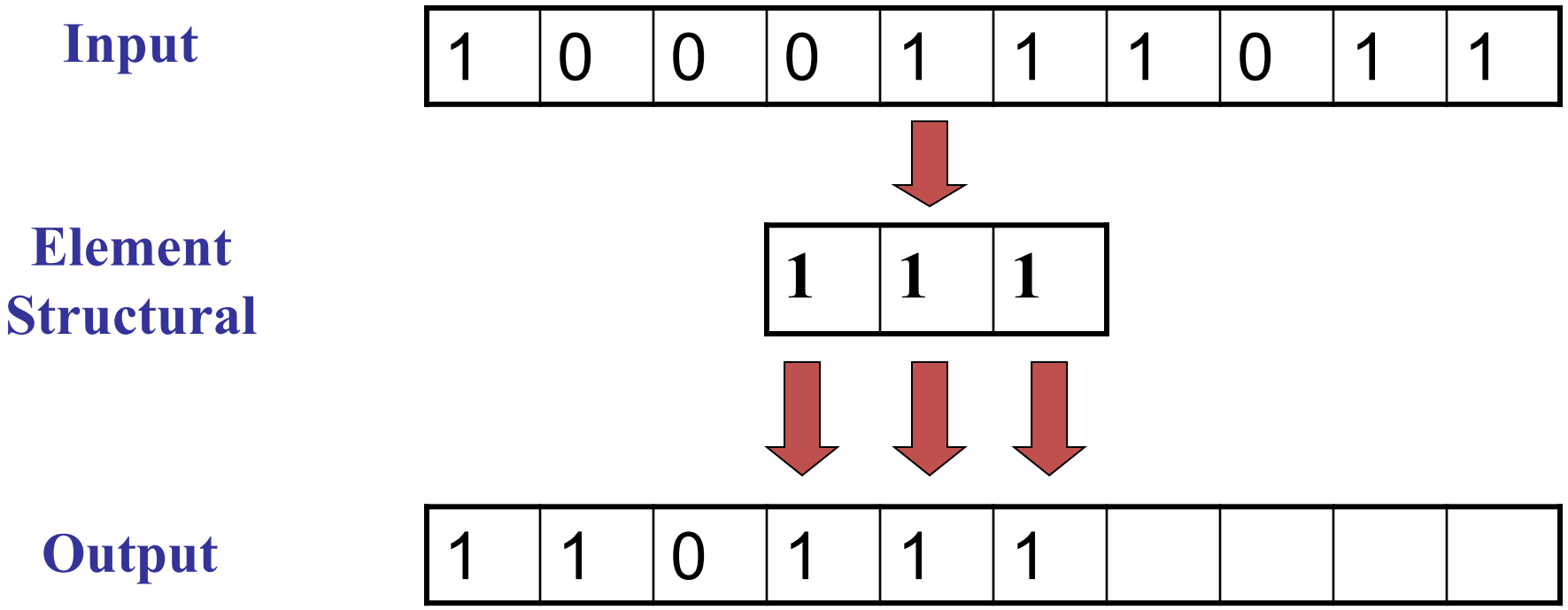
1	**1**	**1**

Output

1	1	0	0						

Slide adaptat după T. Moeslund

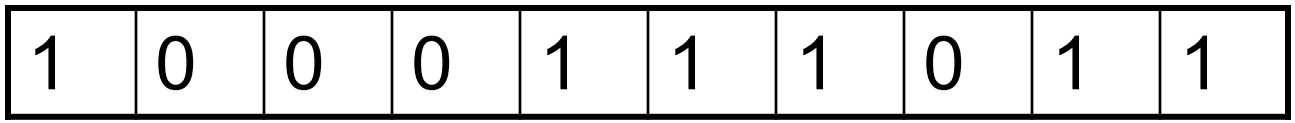
Dilatare – exemplu



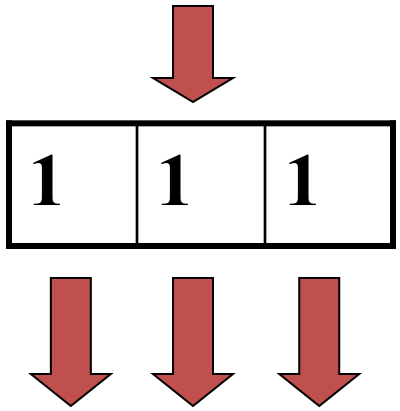
Slide adaptat după T. Moeslund

Dilatare – exemplu

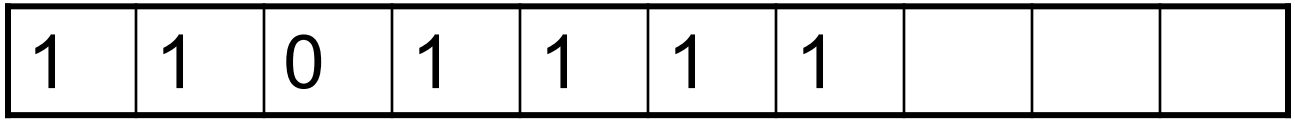
Input



Element Structural

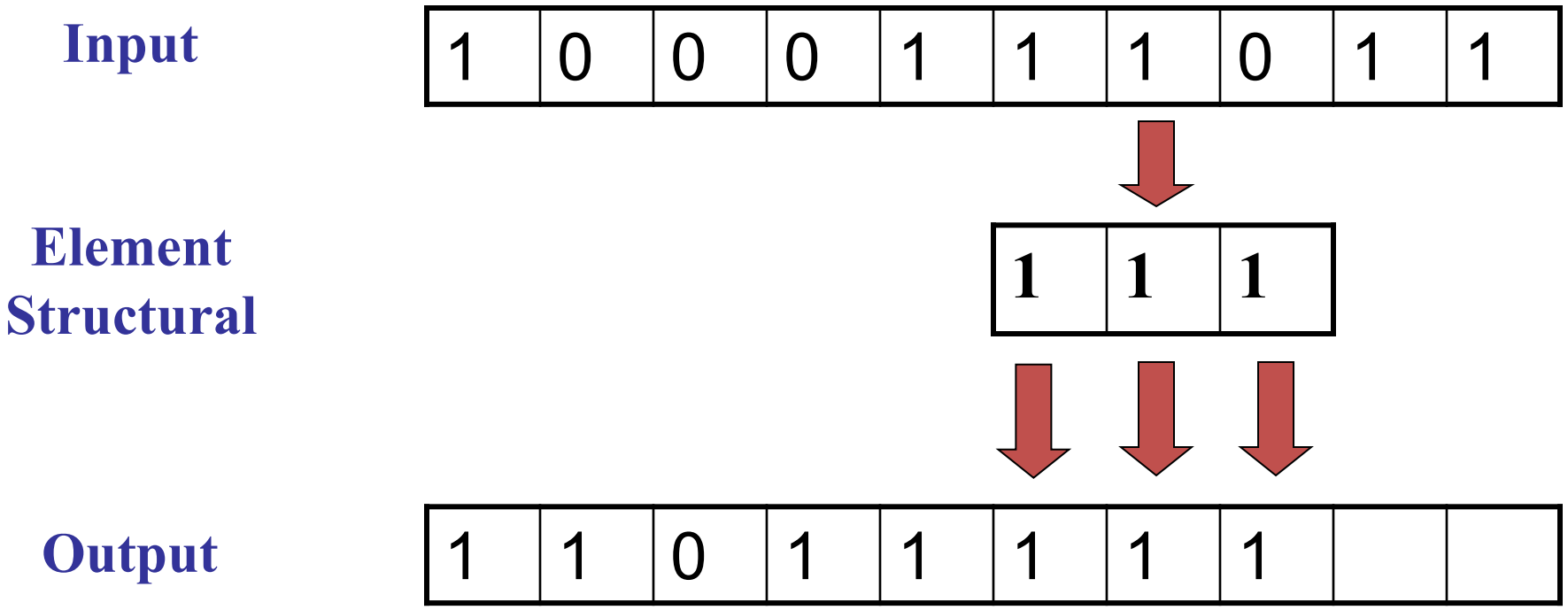


Output



Slide adaptat după T. Moeslund

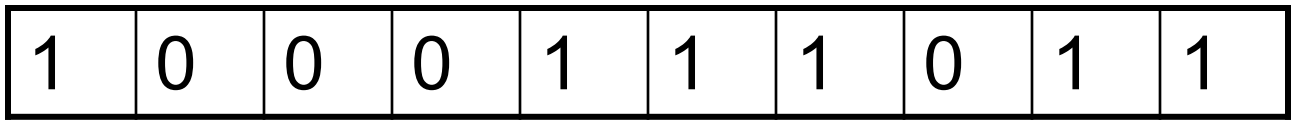
Dilatare – exemplu



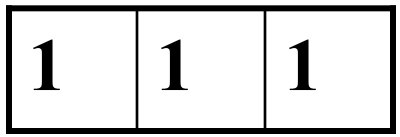
Slide adaptat după T. Moeslund

Dilatare – exemplu

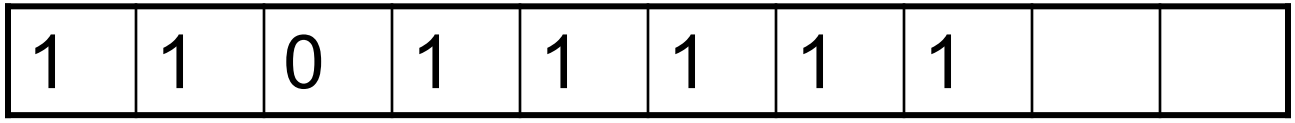
Input



Element Structural

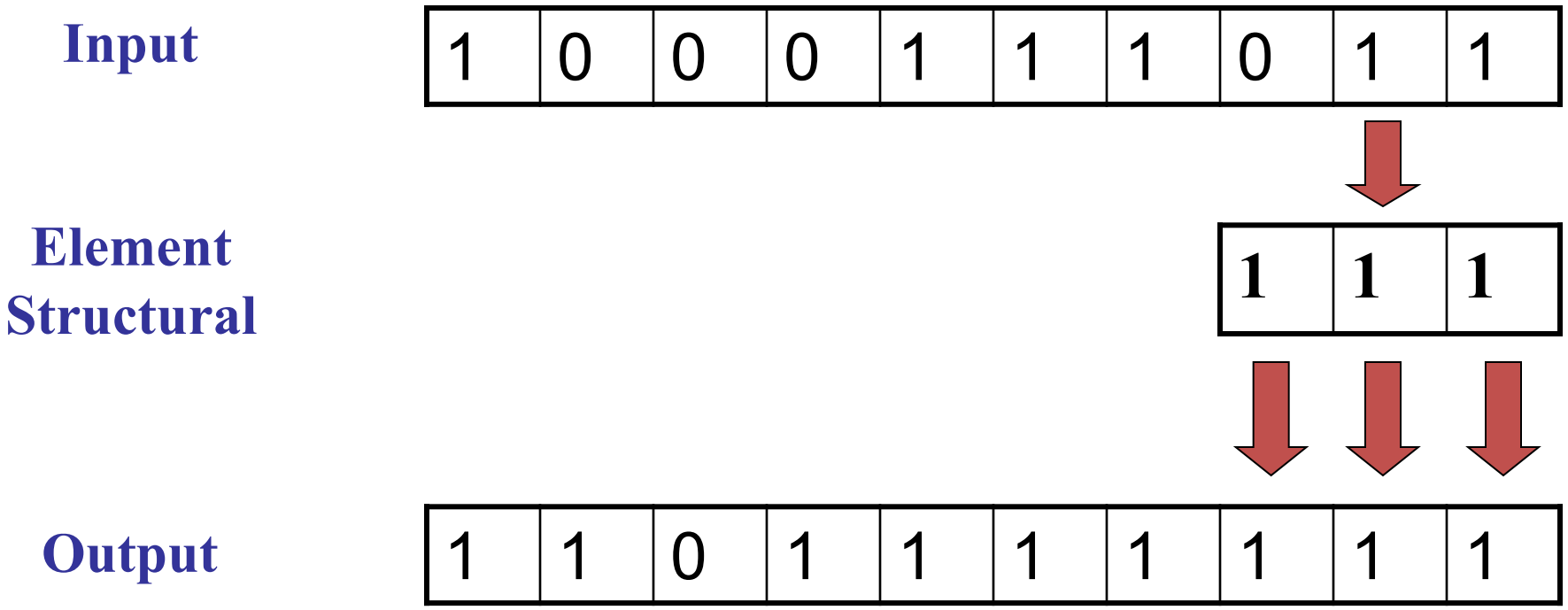


Output



Slide adaptat după T. Moeslund

Dilatare – exemplu



Foreground-ul devine mai mare, găurile sunt umplute.

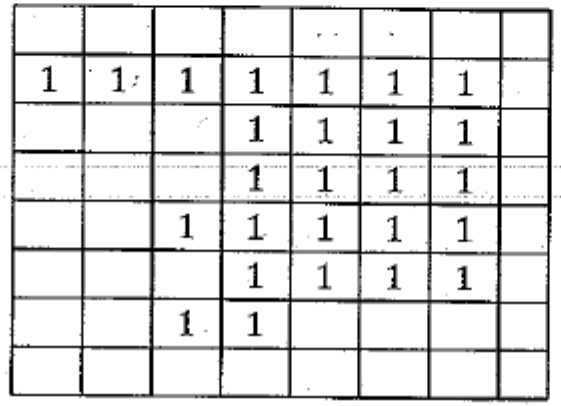
Slide adaptat după T. Moeslund

Dilatare vs. Eroziune

Pentru fiecare pixel:

- **Dilatare**: dacă originea elementului structural se suprapune peste un pixel alb/foreground din imagine, atunci toți pixelii acoperiți de valori 1 din elementul structural devin albi/foreground.

Dilatare – exemplu (2D)



Imagine binară **B**

Element structural **S**

Slide adaptat după Shapiro

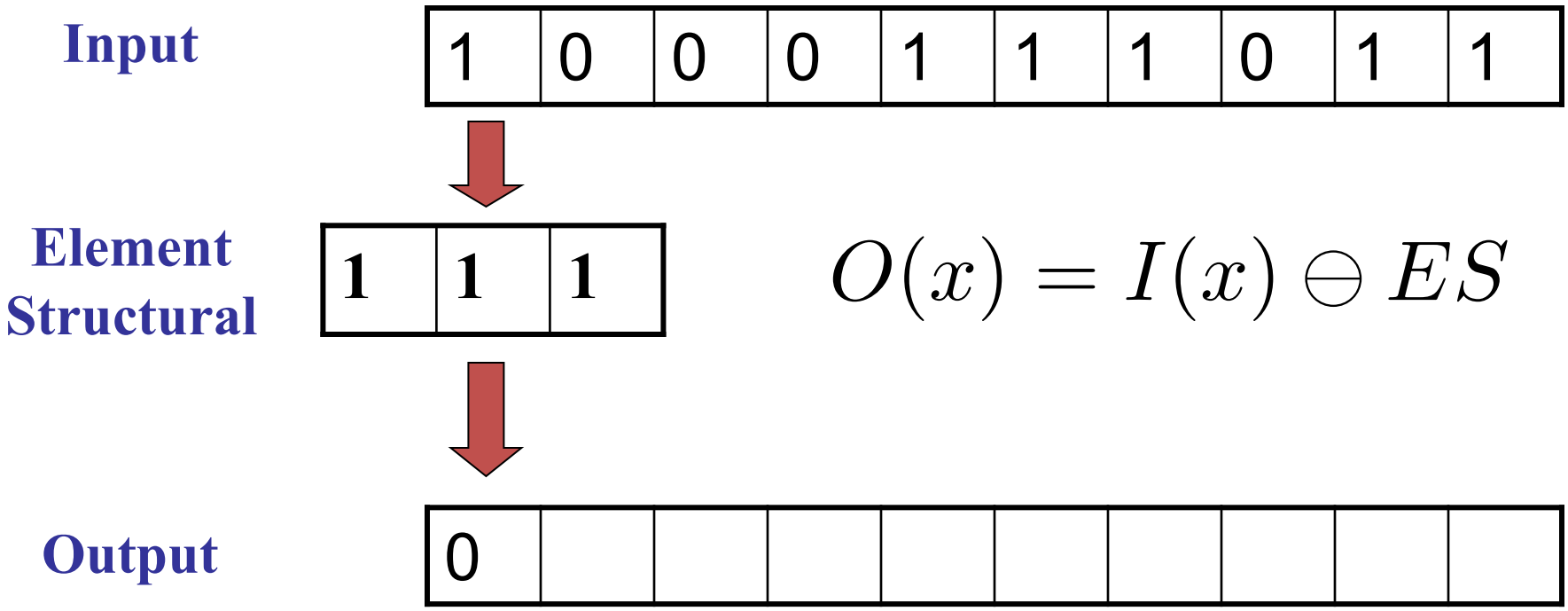
Dilatare vs. Eroziune

Pentru fiecare pixel:

- **Dilatare**: dacă originea elementului structural coincide cu un pixel alb/foreground din imagine, atunci toți pixelii acoperiți de valori 1 din elementul structural devin albi/foreground

- **Eroziune**: dacă originea elementului structural se suprapune peste un pixel alb/foreground din imagine, şi există un pixel alb/foreground din elementul structural care se suprapune peste un pixel negru/background din imagine atunci pixelul devine negru/background

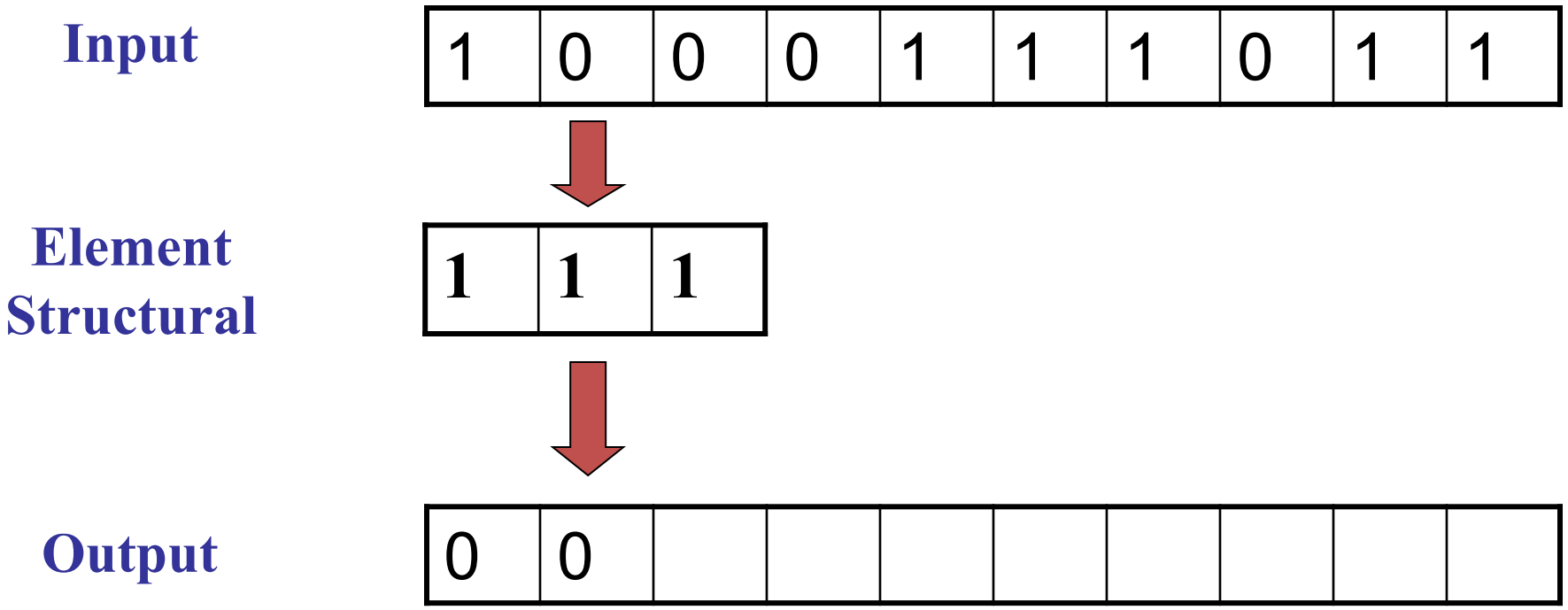
Eroziune – exemplu (1D)



$$O(x) = I(x) \ominus ES$$

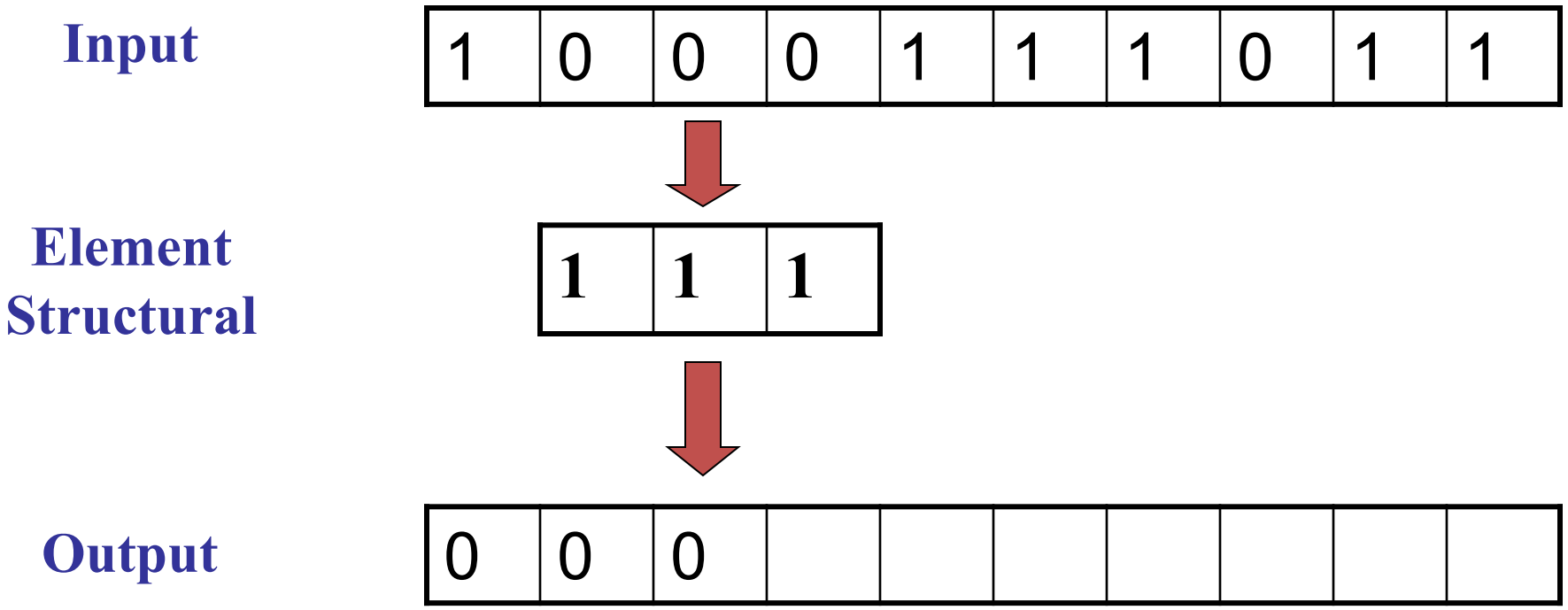
Slide adaptat după T. Moeslund

Eroziune – exemplu



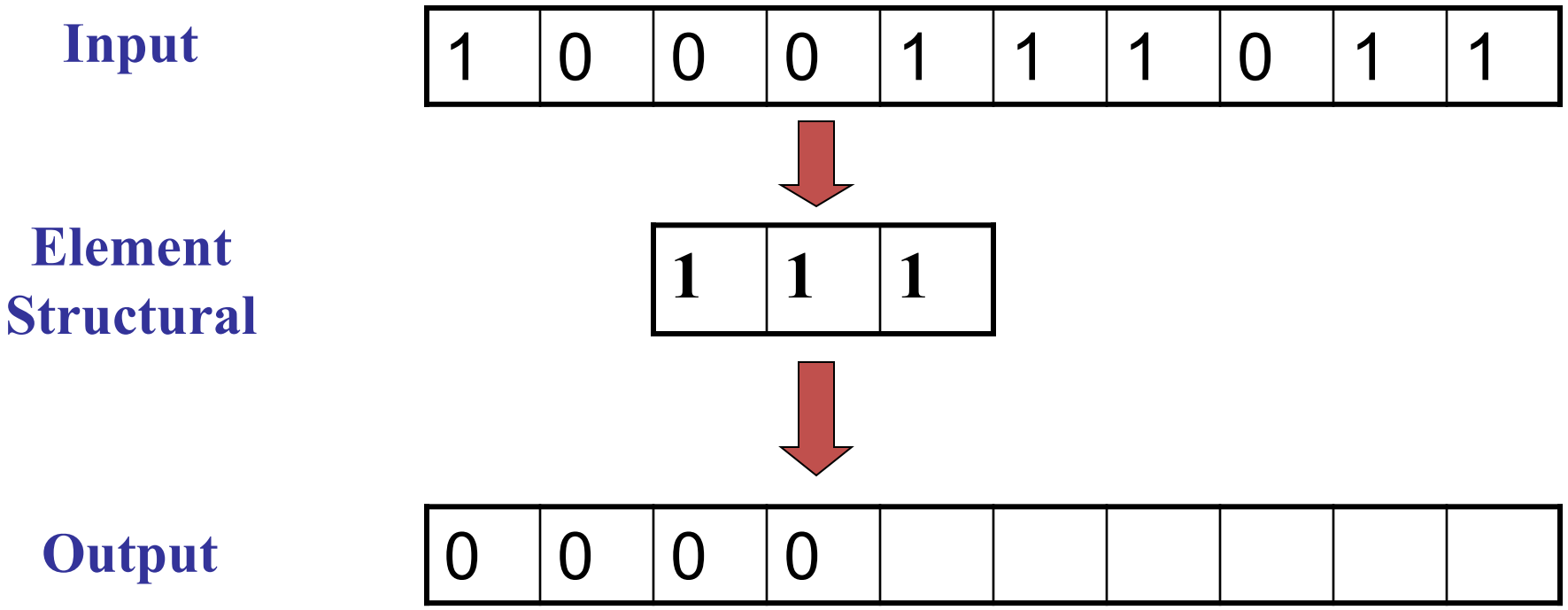
Slide adaptat după T. Moeslund

Eroziune – exemplu



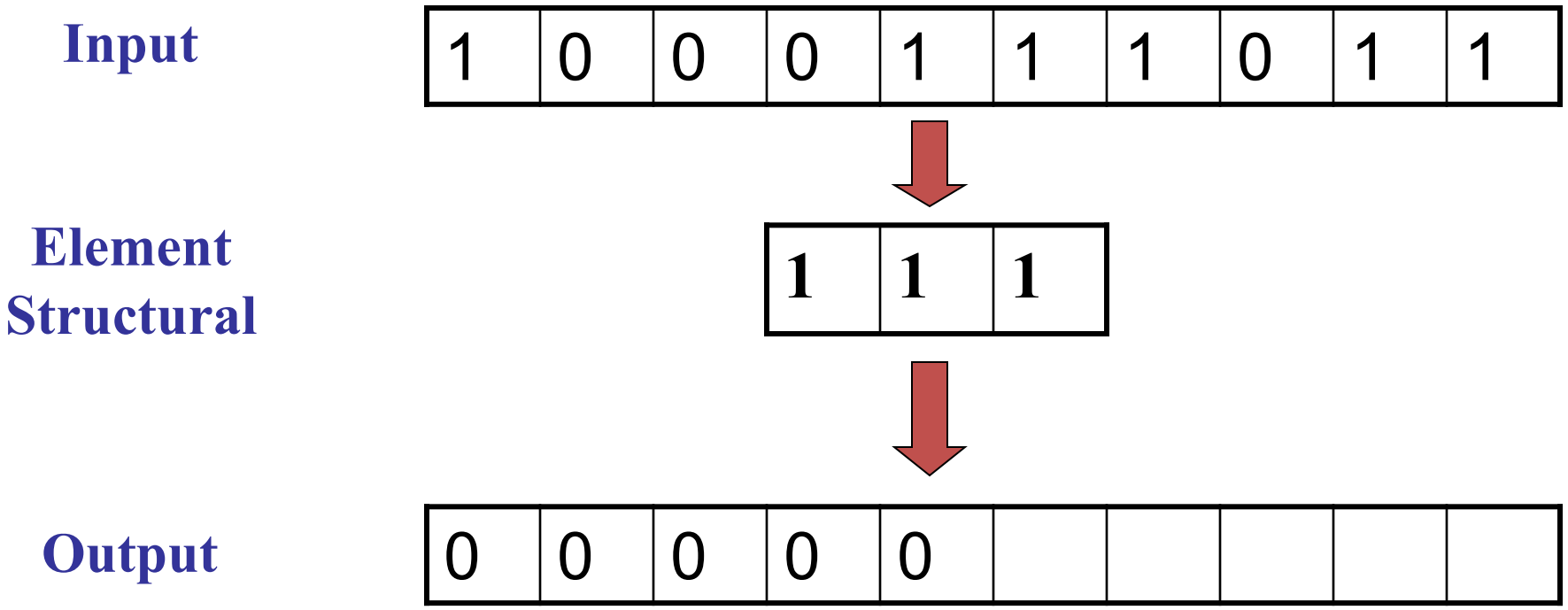
Slide adaptat după T. Moeslund

Eroziune – exemplu



Slide adaptat după T. Moeslund

Eroziune – exemplu



Slide adaptat după T. Moeslund

Eroziune – exemplu

Input

1	0	0	0	1	1	1	0	1	1

Element Structural

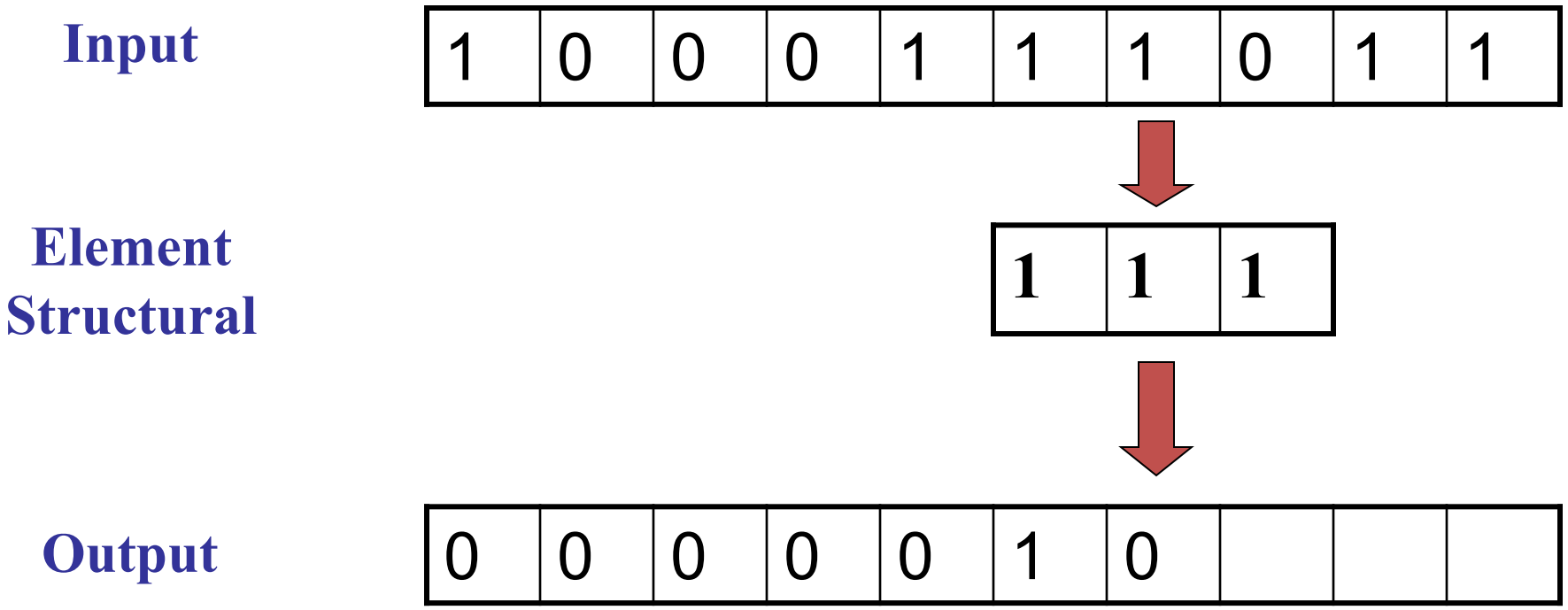
1	**1**	**1**

Output

0	0	0	0	0	1				

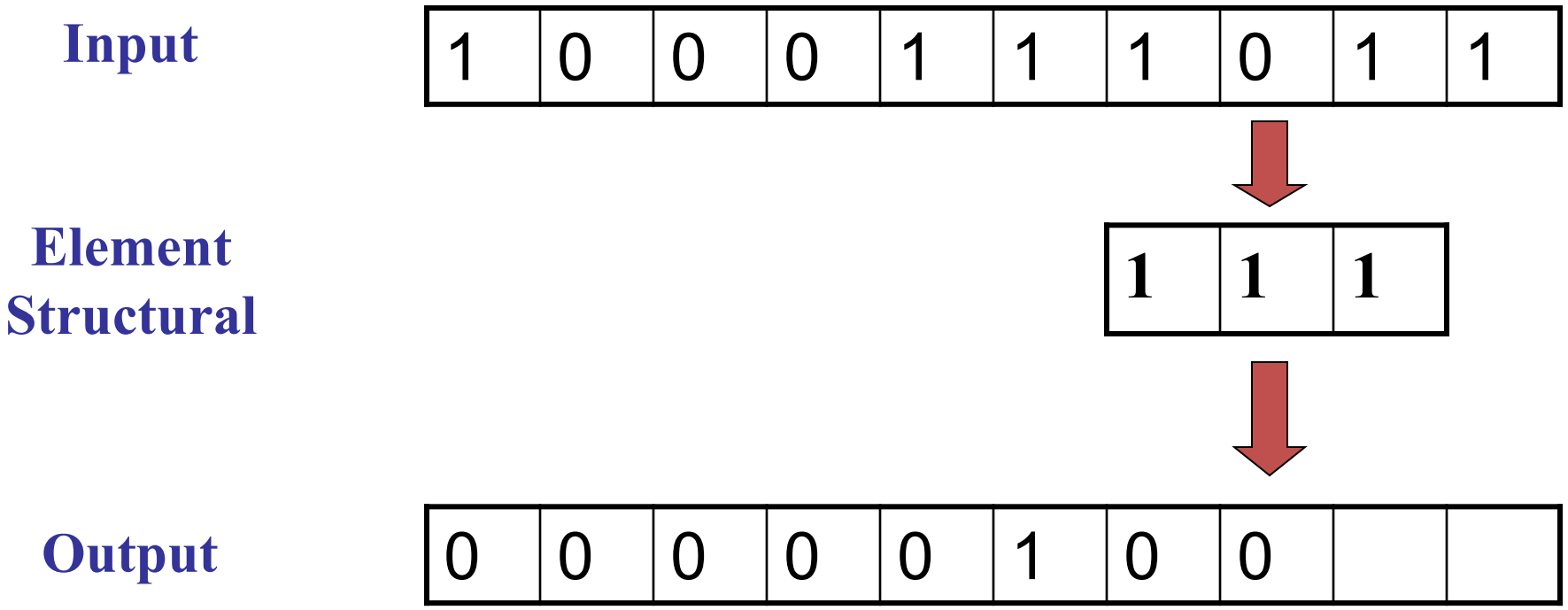
Slide adaptat după T. Moeslund

Eroziune – exemplu



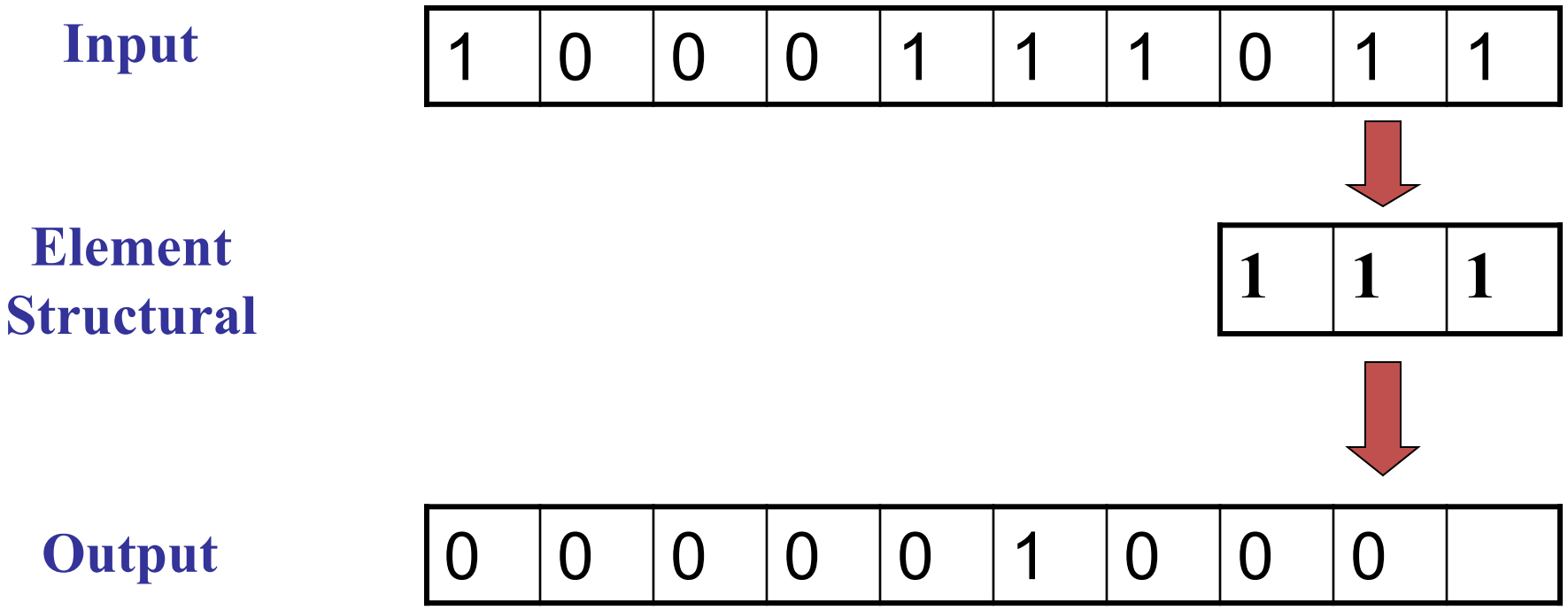
Slide adaptat după T. Moeslund

Eroziune – exemplu



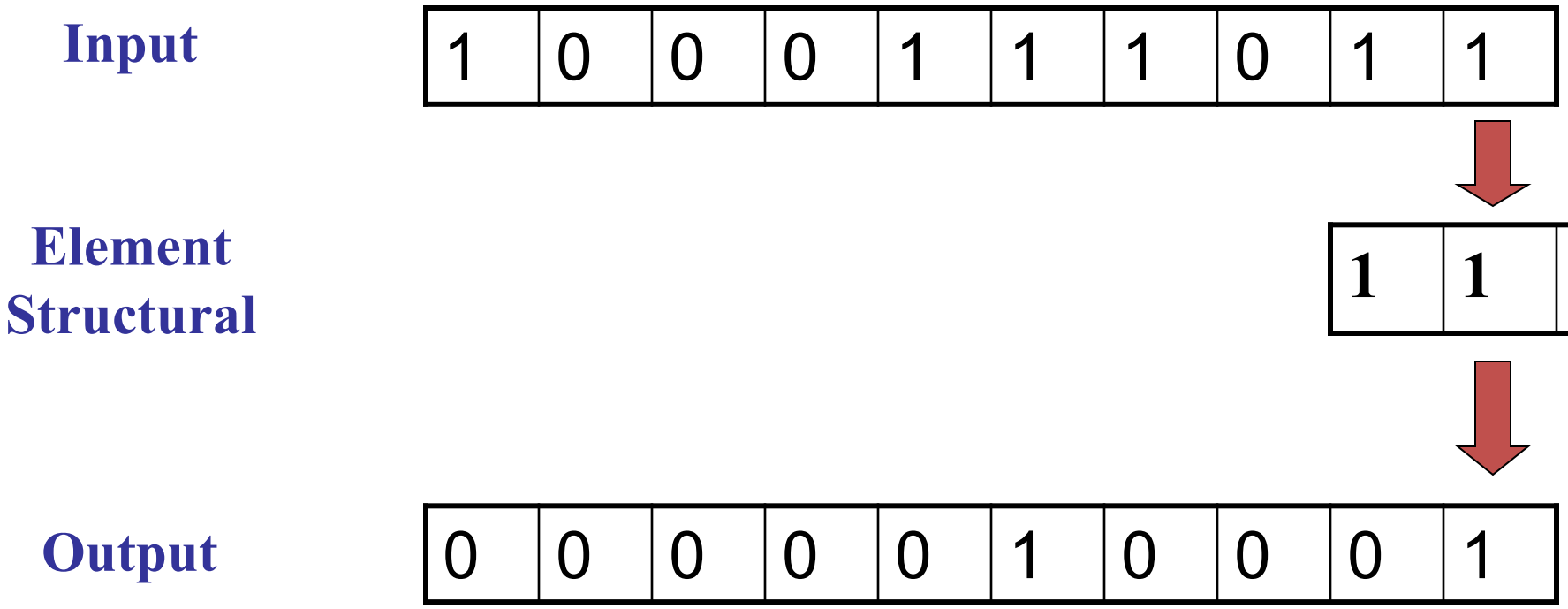
Slide adaptat după T. Moeslund

Eroziune – exemplu



Slide adaptat după T. Moeslund

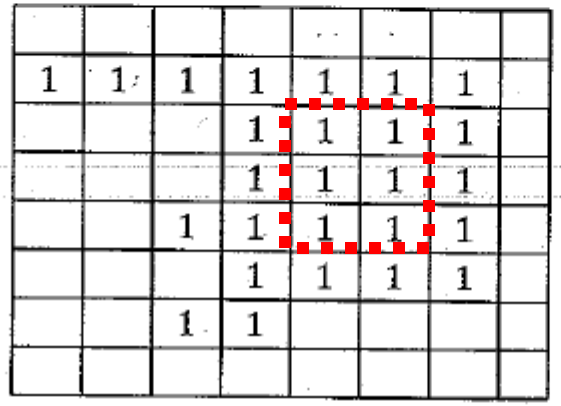
Eroziune – exemplu



Foreground-ul se micşorează.

Slide adaptat după T. Moeslund

Eroziune – exemplu (2D)



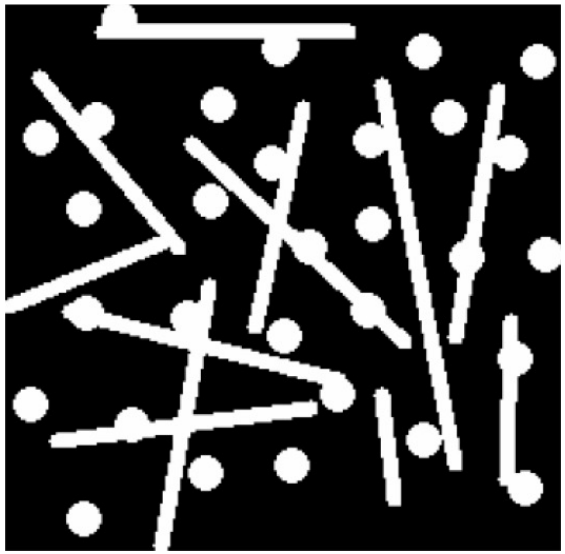
Imagine binară **B**

Element structural **S**

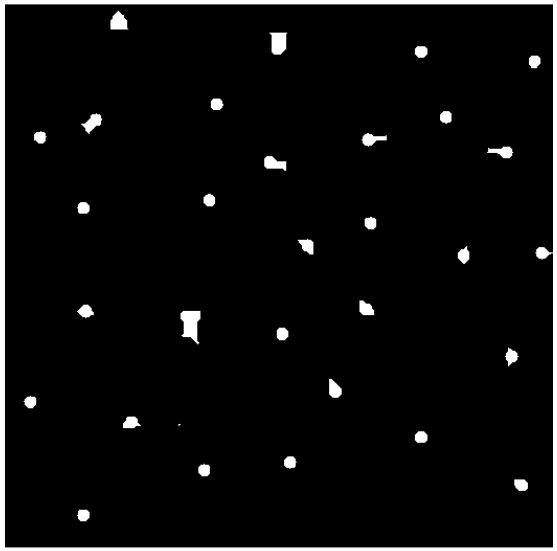
Slide adaptat după Shapiro

Deschidere = eroziune + dilatare

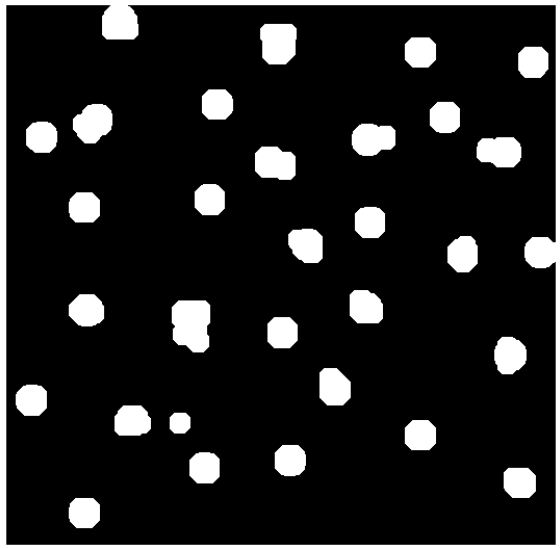
- Eroziune, apoi dilatare
- Elimină obiecte mici, păstrează forma originală



Imagine inițială



Eroziune



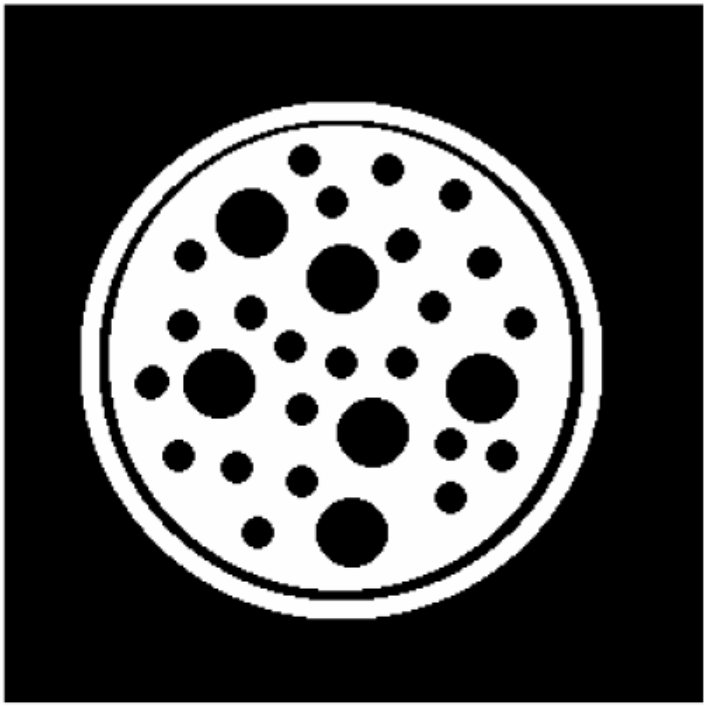
Eroziune + Dilatare

Element de structură

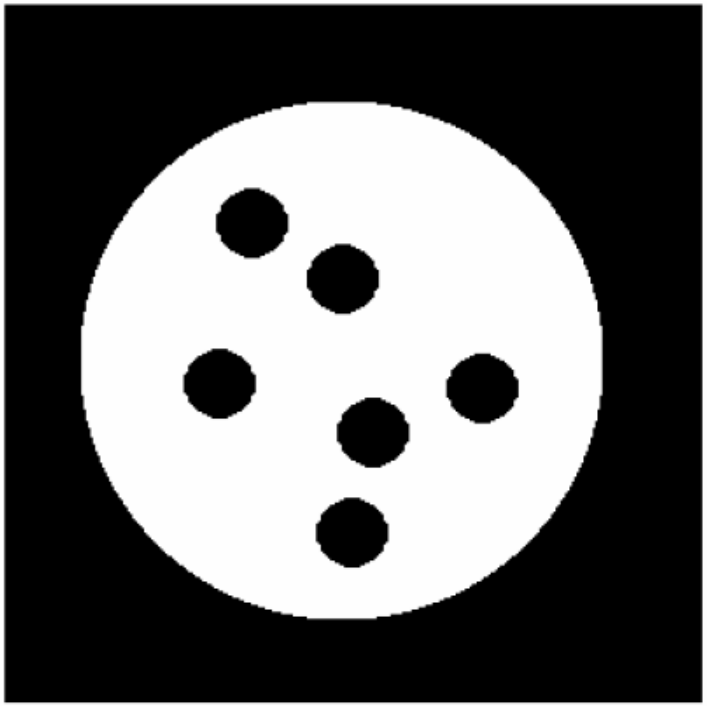
Slide adaptat după K. Grauman

Închidere = dilatare + eroziune

- Dilatare, apoi eroziune
- Umple găurile, păstrează forma originală



Înainte de închidere



După închidere

Element de structură 

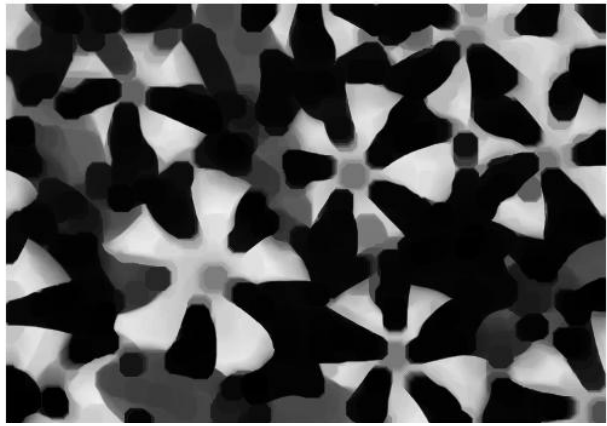
Slide adaptat după K. Grauman

Operatori morfologici aplicați imaginilor de intensitate

- Dilatarea şi eroziune - în principal pentru imagini binare.
- Pentru imagini de intensitate:
  - dilatare: ia maximumul pe vecinătate ES
  - eroziune: ia minimul pe vecinătatea ES

imagine de intensitate

dilatare



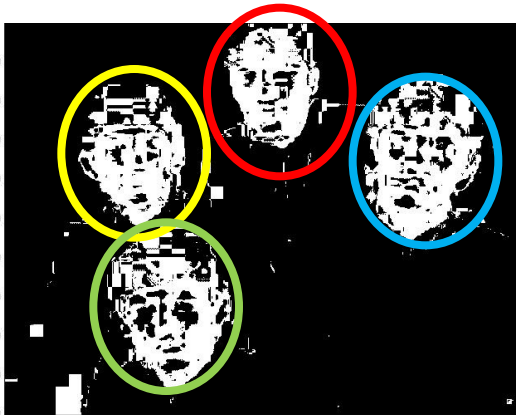
eroziune

Slide adaptat după K. Grauman

Probleme

- Ce facem cu imaginile binare rezultate ca cea din dreapta?
  - găuri
  - fragmente foarte mici, izolate

- Cum delimităm regiunile de interes?
  - numărăm obiecte
  - calculăm diverse proprietăți pentru un obiect



Slide adaptat după K. Grauman

Componente conexe

- Identifică regiuni distincte de pixeli “conectați”

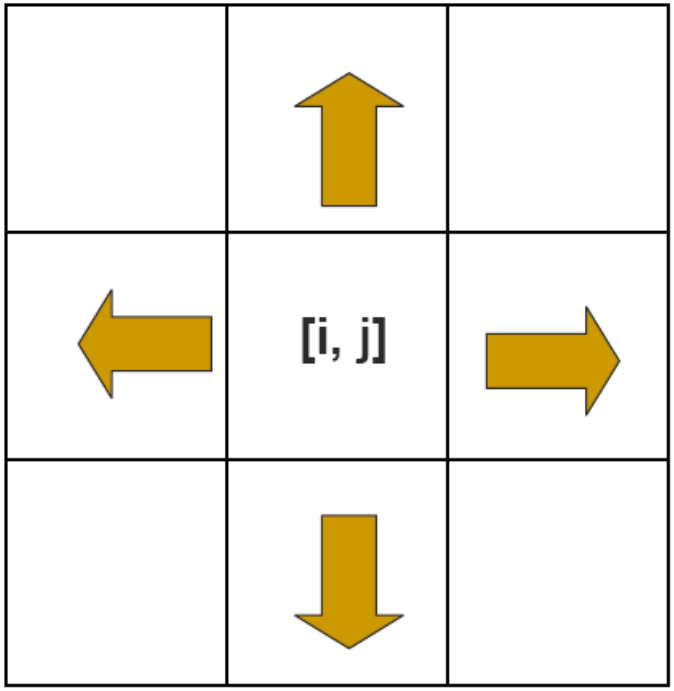
1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

Imagine binară

Slide adaptat după Shapiro

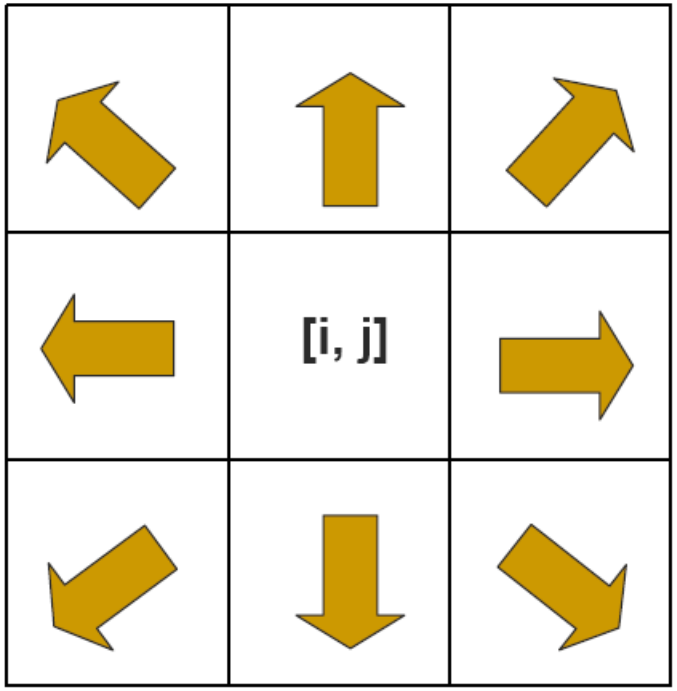
Conectivitate

- Defineşte ce pixeli sunt consideraţi vecini (“conectaţi”)



conectivitate – 4

(4 vecini)



conectivitate – 8

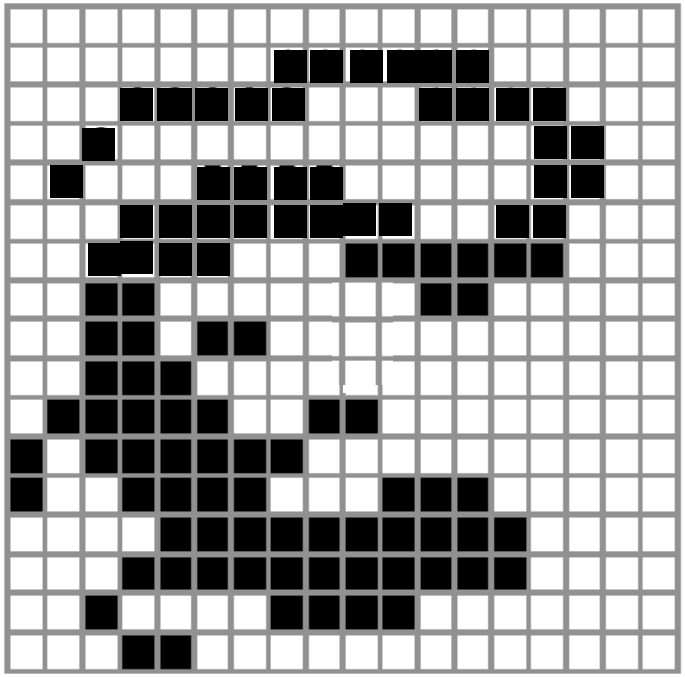
(8 vecini)

Slide adaptat după C. Chandra

Algoritm pentru determinarea componentelor conexe

- Considerăm un algoritm secvențial liniar în numărul de pixeli ai imaginii (2 treceri).

- **Input**: imagine binară
- **Output**: imagine "etichetată", fiecare pixel are asignat un număr al componentei conexe



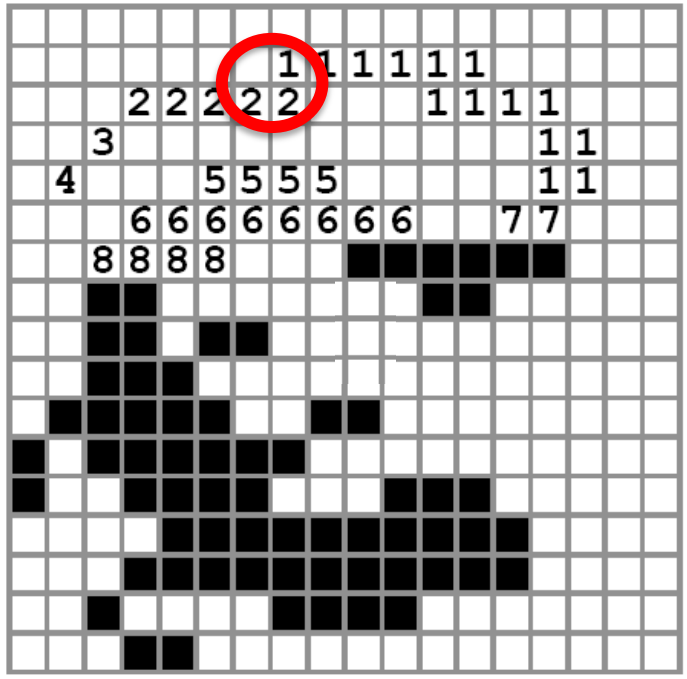
- Foreground-ul este reprezentat în acest exemplu de pixeli negri

- Conectivitate - 4

Slide adaptat după J. Neira

Algoritm pentru determinarea componentelor conexe

- Etichetăm un pixel foreground pe baza valorilor vecinilor săi
- Vecinii - dați de conectivitate (aici – 4)
- Procesăm pixelii din imagine secvențial de la stânga la dreapta, de sus în jos
- Comparăm pixelul curent cu vecinul din stânga si cel de sus

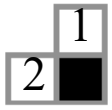


(a) (b) (c)

aceeaşi componentă conexă – preia eticheta

(d)

altă componentă conexă – adaugă etichetă



(e)

aceeaşi componentă conexă – etichete diferite
Cum procedăm?

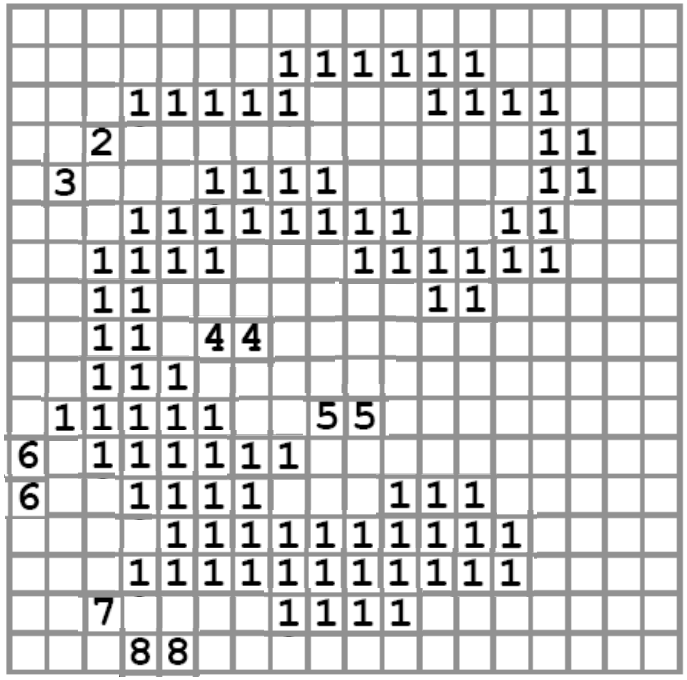
Tabel de echivalență

{1, 2, 7}
{3}
{4}
{5, 6, 8}

Slide adaptat după J. Neira

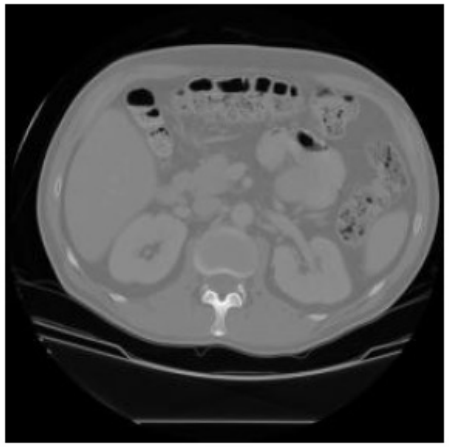
Algoritm pentru determinarea componentelor conexe

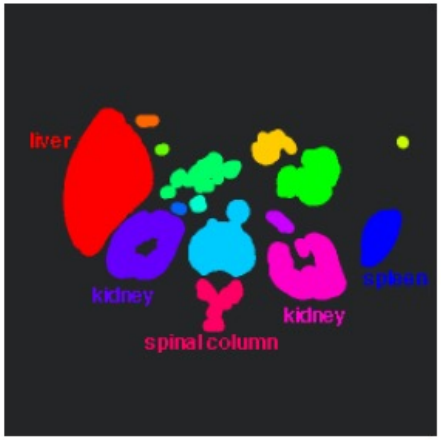
- Etichetăm un pixel foreground pe baza valorilor vecinilor săi
- Vecinii - dați de conectivitate (aici – 4)
- Procesăm pixelii din imagine secvențial de la stânga la dreapta, de jos în sus
- Comparăm pixelul curent cu vecinul din stânga si cel de sus
- Pe baza tabelului de echivalență renumerotăm componentele



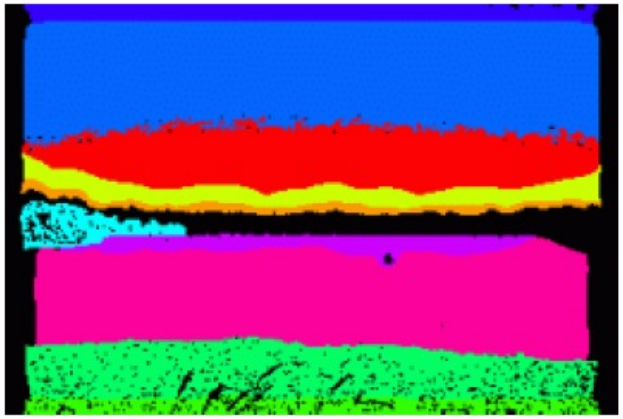
Slide adaptat după J. Neira

Componente conexe - exemple





componente conexe obținute în urma aplicării unui prag

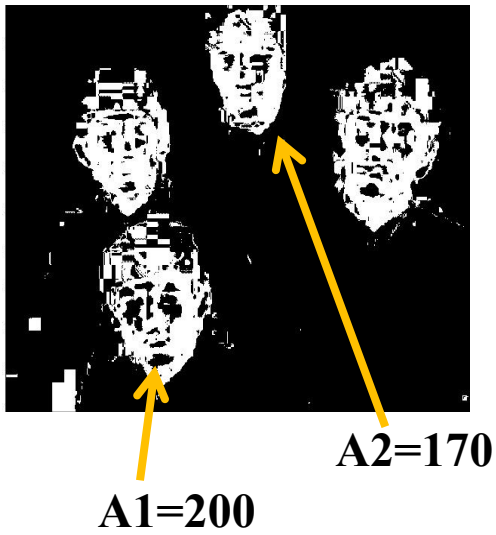


componente conexe obținute în urma grupării pixelilor în clustere

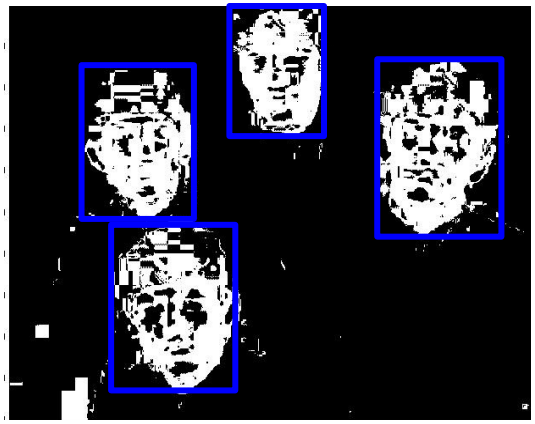
Slide adaptat după P. Duygulu

Proprietăți ale regiunilor

- Fiind date componente conexe, putem calcula proprietăți pentru caracterizarea lor:
  - aria (numărul de pixeli în regiune)
  - centroidul (media coordonatele x and y pentru pixelii din regiune)
  - fereastra dreptunghiulară ce acoperă regiunea (pe baza coordonatelor minime și maxime x,y)



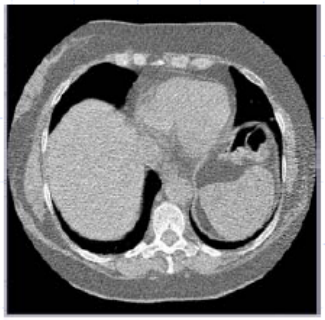
centroizii regiunilor



dreptunghi ce acoperă regiunea

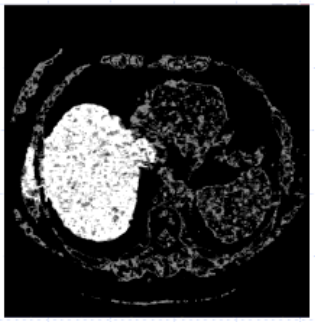
Slide adaptat după K. Grauman

Analiza imaginilor binare - exemplu segmentare

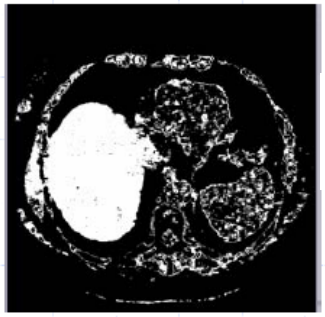


Aplicare prag

Extrage regiunea cea mai mare



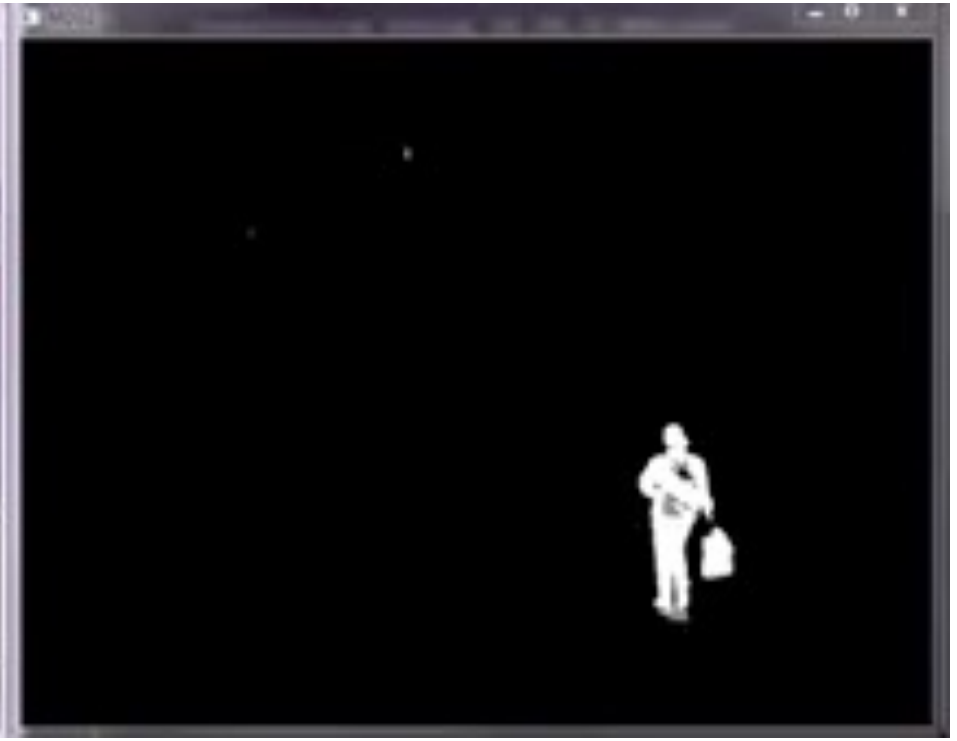
umple golurile



Extrage regiunea cea mai densă

Slide adaptat după L. Shen

Analiza imaginilor binare - exemplu extragerea background-ului, detectare regiuni



Imagini binare

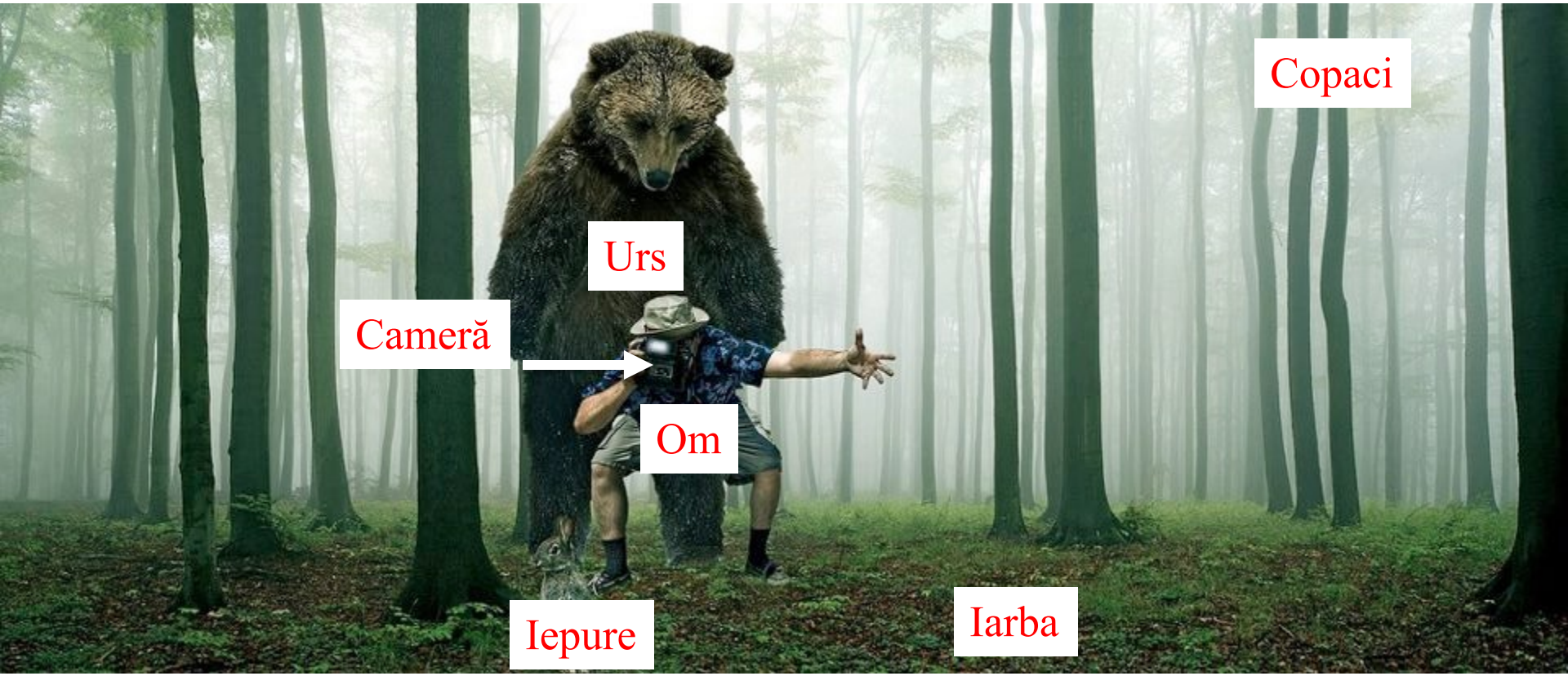
- Avantaje
  - foarte rapid de procesat
  - necesită memorie mică
  - putem obține descriptori compacți pentru forme

- Dezavantaje
  - greu să obținem contururi perfecte (siluete)
  - zgomot ce apare în imagini reale
  - pierdem prea multă informație prin binarizare

Slide adaptat după J. Neira

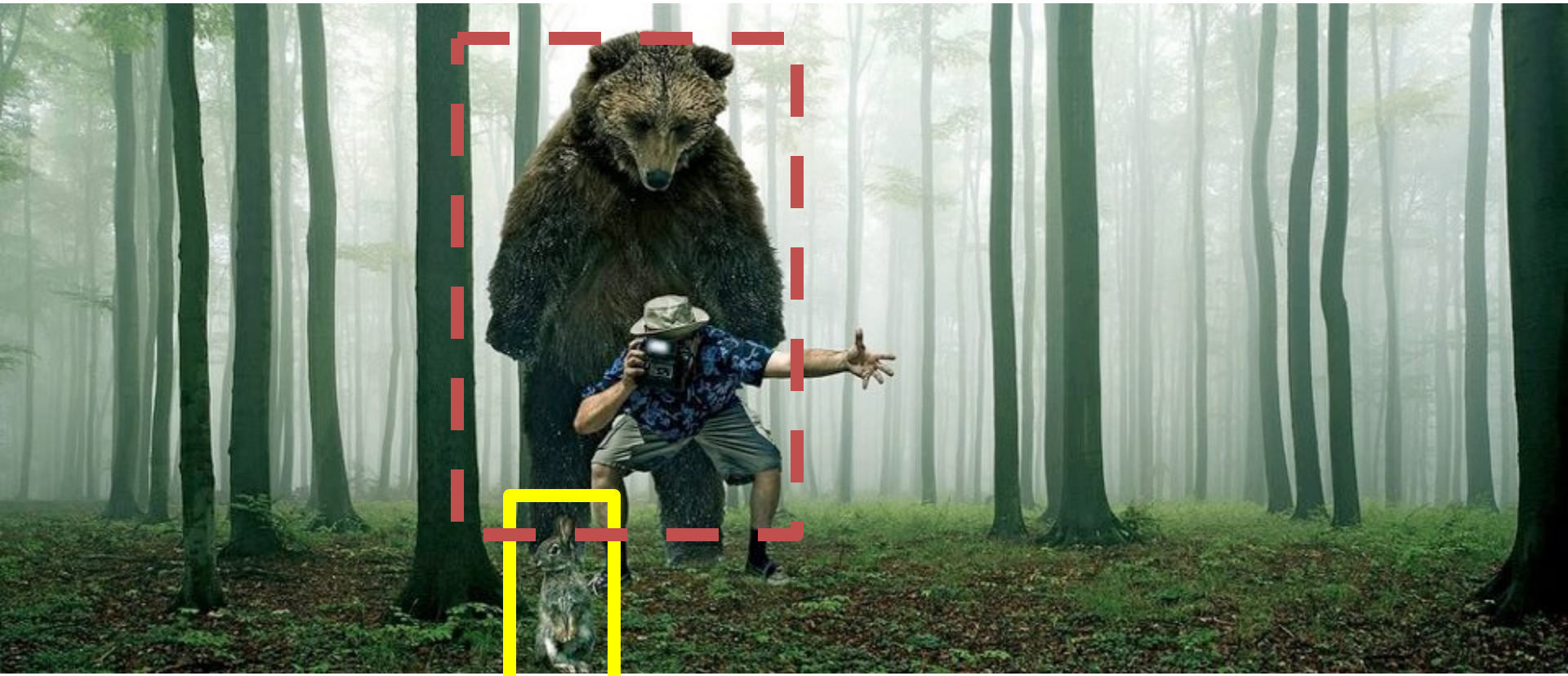
Recunoașterea claselor de obiecte

Ce vedeți în această imagine?



Pădure

Slide adaptat după Derek Hoiem



Este **periculos**?

Este **viu**?

Cât de **repede** aleargă?

Are **coadă**?

Slide adaptat după Derek Hoiem

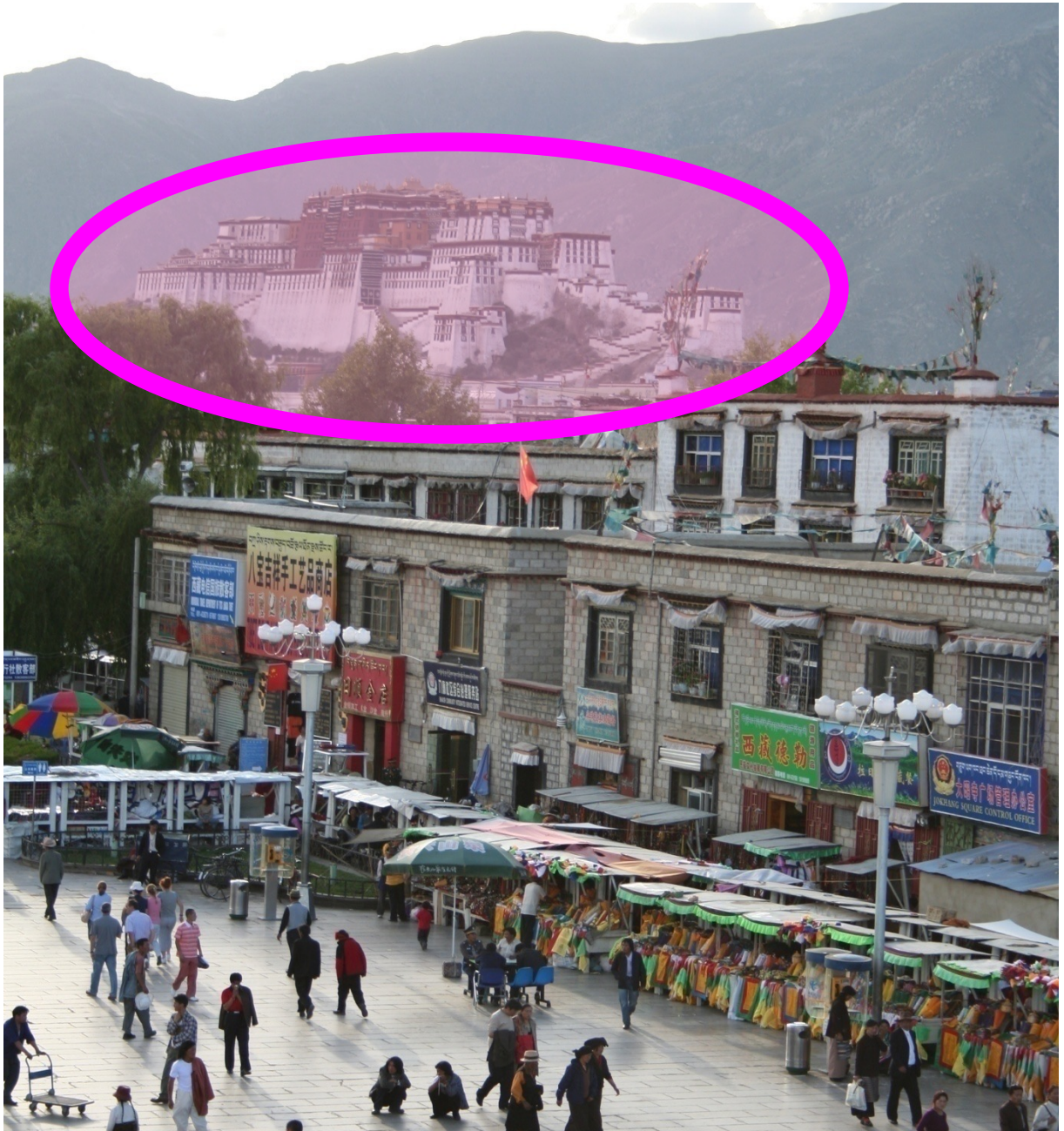
Recunoaşterea vizuală a claselor de obiecte

Clasificare: este vreun felinar în imagine? **Da/Nu**

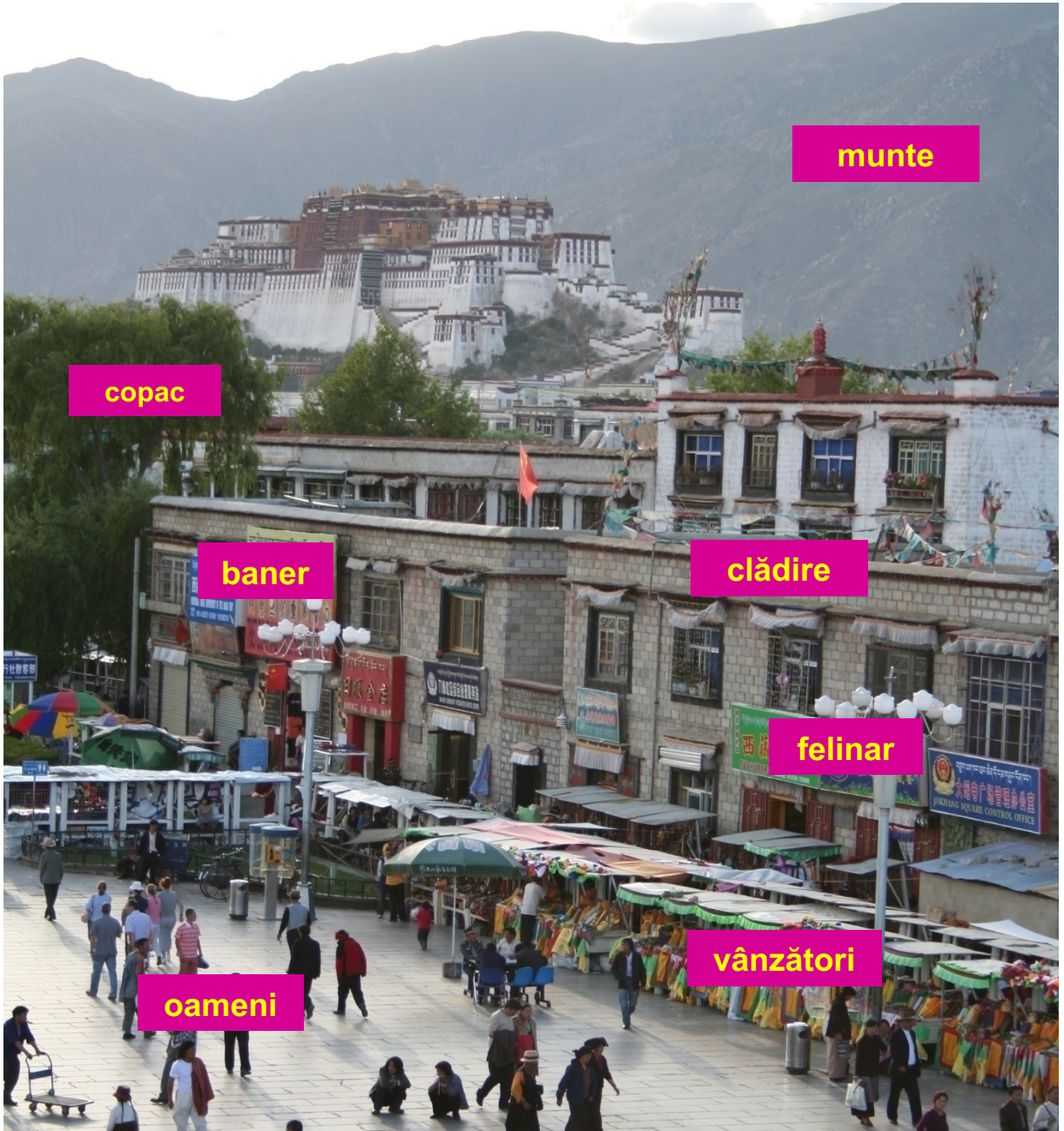
Detectare: unde este felinarul? **Localizare**

Recunoaștere: care palat?

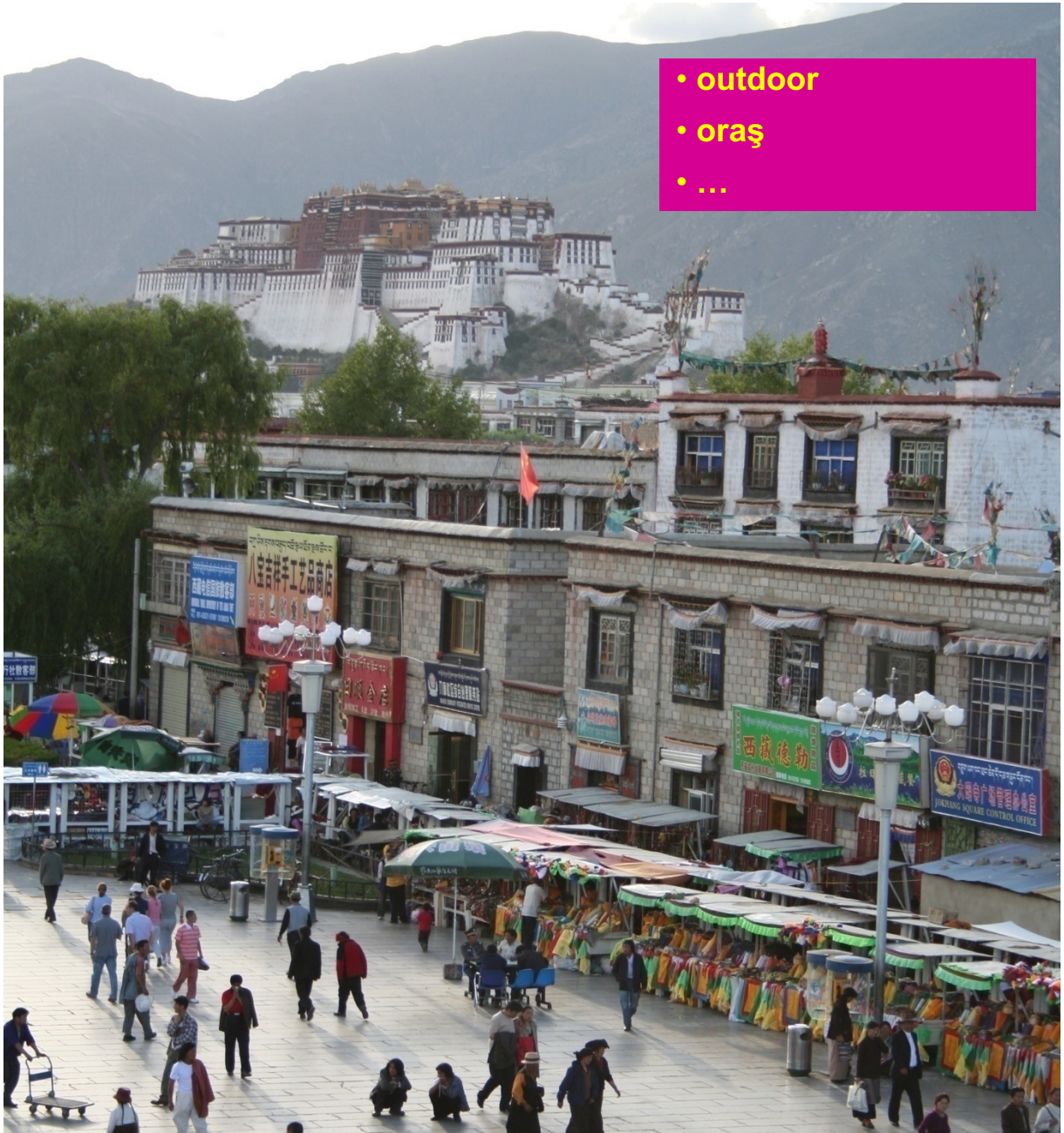
Instanțiere



Clasificarea regiunilor din imagine



Clasificarea întregii scene



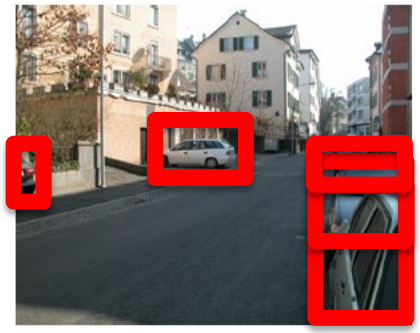
Recunoaşterea claselor de obiecte la nivel de instanță

maşina lui Mihai

Slide adaptat după K. Grauman

Recunoaşterea claselor de obiecte la nivel generic




Vrem să recunoştem toate instanțierile unei clase de obiecte

Slide adaptat după K. Grauman

Recunoaşterea claselor de obiecte

- Descrierea problemei:
  - "Fiind dat un număr mic de imagini de antrenare ale unei clase de obiecte, recunoaşteți instanțieri ale clasei de obiecte asignând eticheta corespunzătoare clasei de obiecte."

- Pentru care clasă de obiecte puteți asocia repede o imagine?

Slide adaptat după K. Grauman

Organizarea ierarhică a claselor de obiecte (Rosch, 1976)

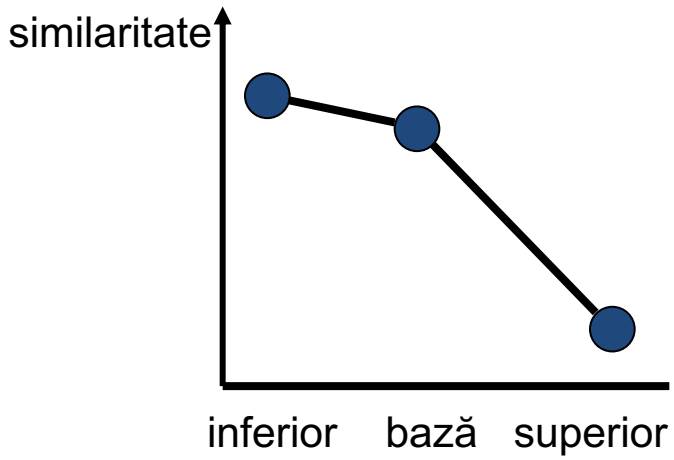
Nivelul de bază:

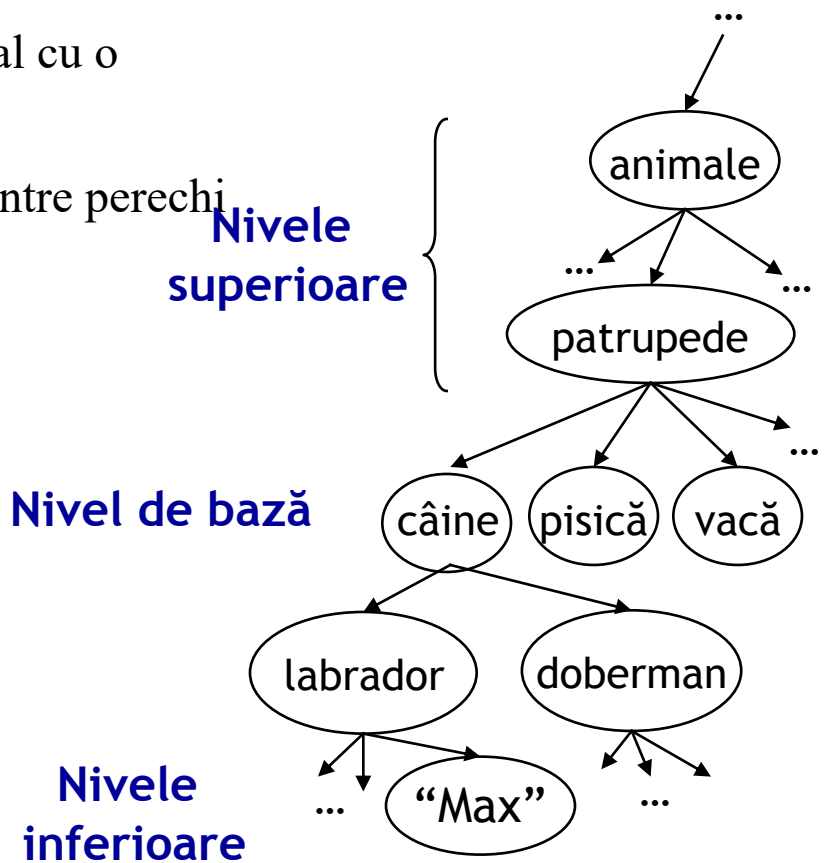
• clasele de obiecte la nivel de bază sunt clasele de nivel cel mai înalt pentru care membrii lor au forme percepute similare.

• nivelul cel mai înalt pentru care reprezentăm mental cu o singură imagine întreaga clasă de obiecte

• există un număr seminificativ de atribute comune între perechi de membri

• primul nivel numit si înțeles de copii





Slide adaptat după K. Grauman

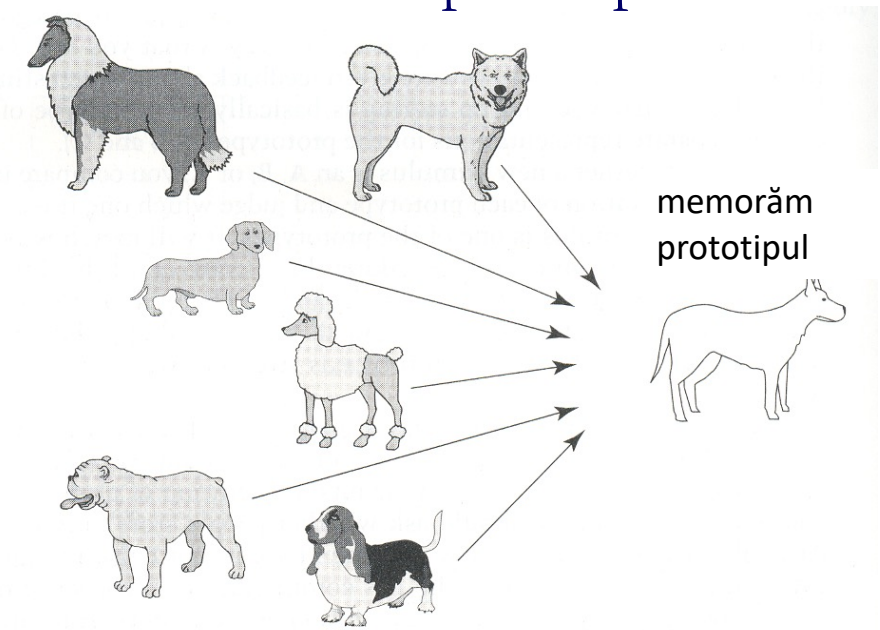
Organizarea ierarhică a claselor de obiecte (Rosch, 1976)

- Rosch a găsit următoarele:
  - oamenii clasifică clasele de obiecte mult mai repede la nivel de bază
  - tendință de a clasifica după nivelul de bază (‘câine’) înainte de nivelul superior (‘animal’) sau nivelul inferior (“golden retriever”)
  - mai întâi clasificare la nivel de bază iar apoi recunoaștere
  - acord la nivel de bază (de cele mai multe ori)
    - desktop, laptop, tablete, pc, mac – nivel de bază
    - calculator – nivel de bază
      - desktop, laptop, tablete, pc, mac – nivel inferior
      - ține de nivelul de expertiză

Slide adaptat după K. Grauman

Recunoaşterea claselor de obiecte

Modelul prototip

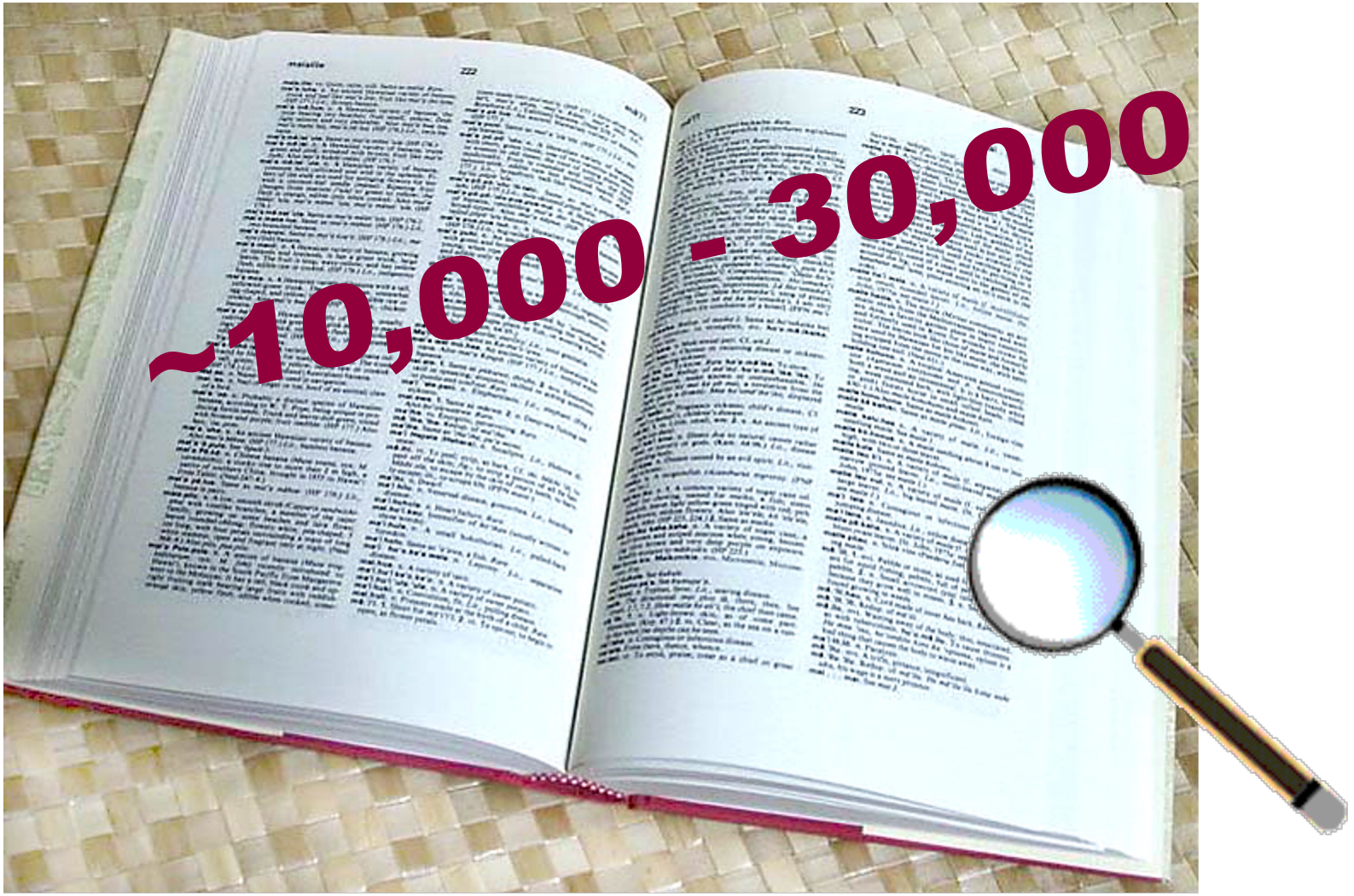


Realizăm clasificarea într-o clasă de obiecte prin compararea unui nou exemplu cu prototipul

Slide adaptat după A. Torralba

Clase de obiecte - exemple

Cât de multe clase de obiecte există?



Biederman 1987

Slide adaptat după A. Torralba

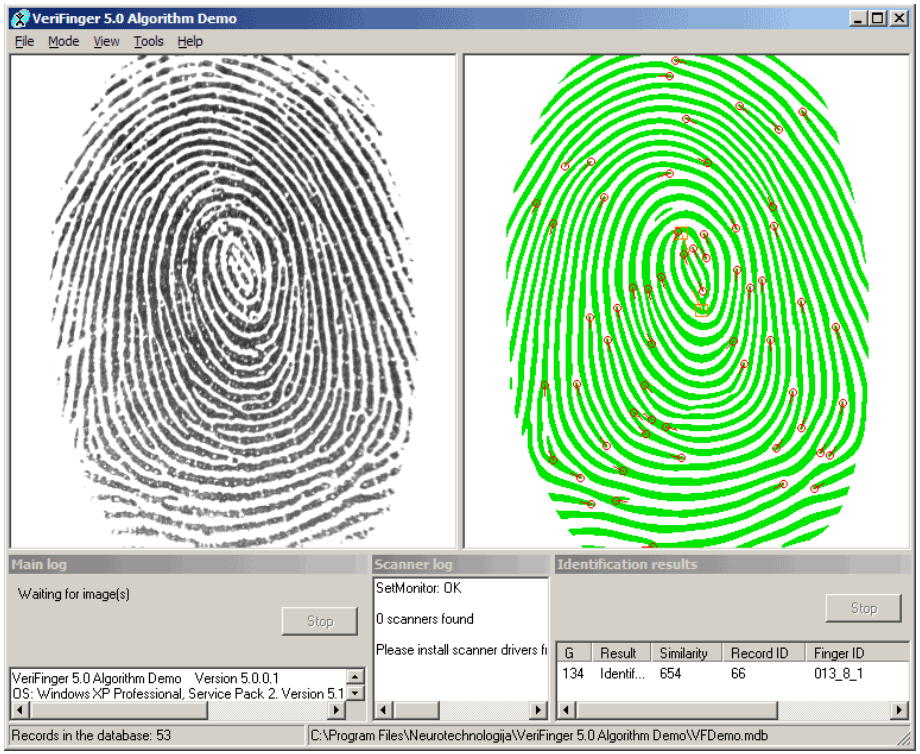
Recunoaşterea claselor de obiecte în Vederea Artificilă - ce funcționează bine

- OCR – plăcuțe de înmatriculare, coduri poştale, etc.

Slide adaptat după S. Lazebnik

Recunoaşterea claselor de obiecte în Vederea Artificilă - ce funcționează bine

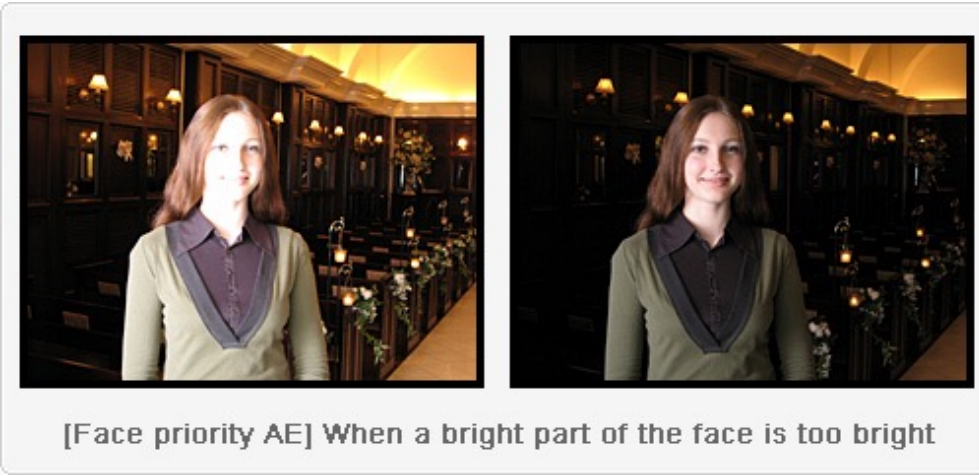
- OCR – plăcuțe de înmatriculare, coduri poştale, etc.
- Recunoaşterea amprentei



Slide adaptat după S. Lazebnik

Recunoaşterea claselor de obiecte în Vederea Artificilă - ce funcționează bine

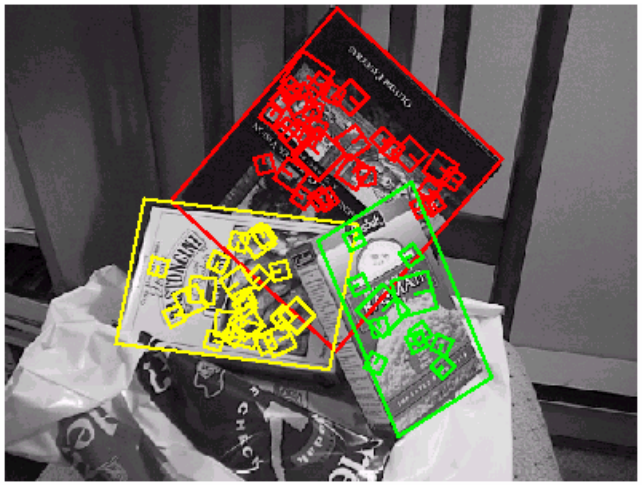
- OCR – plăcuțe de înmatriculare, coduri poştale, etc.
- Recunoaşterea amprentei
- Detectarea facială



Slide adaptat după S. Lazebnik

Recunoaşterea claselor de obiecte în Vederea Artificilă - ce funcționează bine

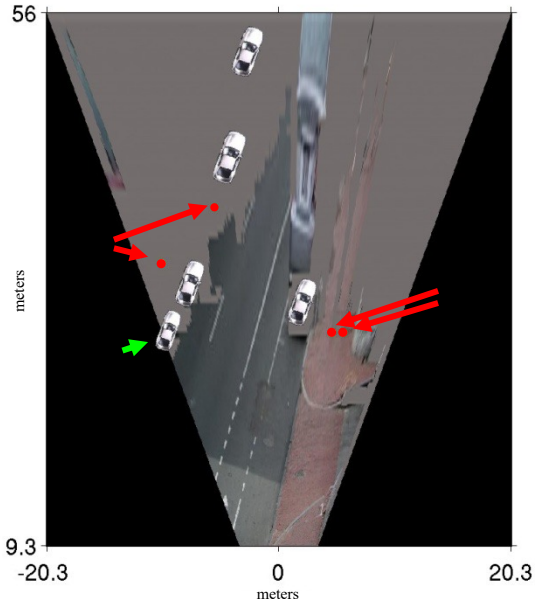
- OCR – plăcuțe de înmatriculare, coduri poştale, etc.
- Recunoaşterea amprentei
- Detectarea facială
- Recunoaşterea obiectelor cu textură (coperți CD)

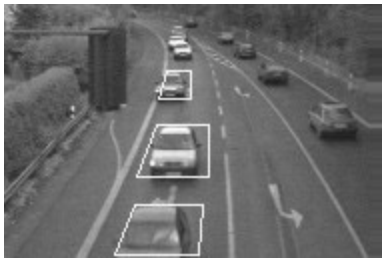


Slide adaptat după S. Lazebnik

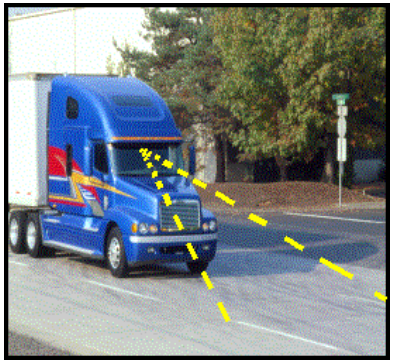
Alte aplicații: conducere asistată

Detectarea maşinilor şi a pietonilor



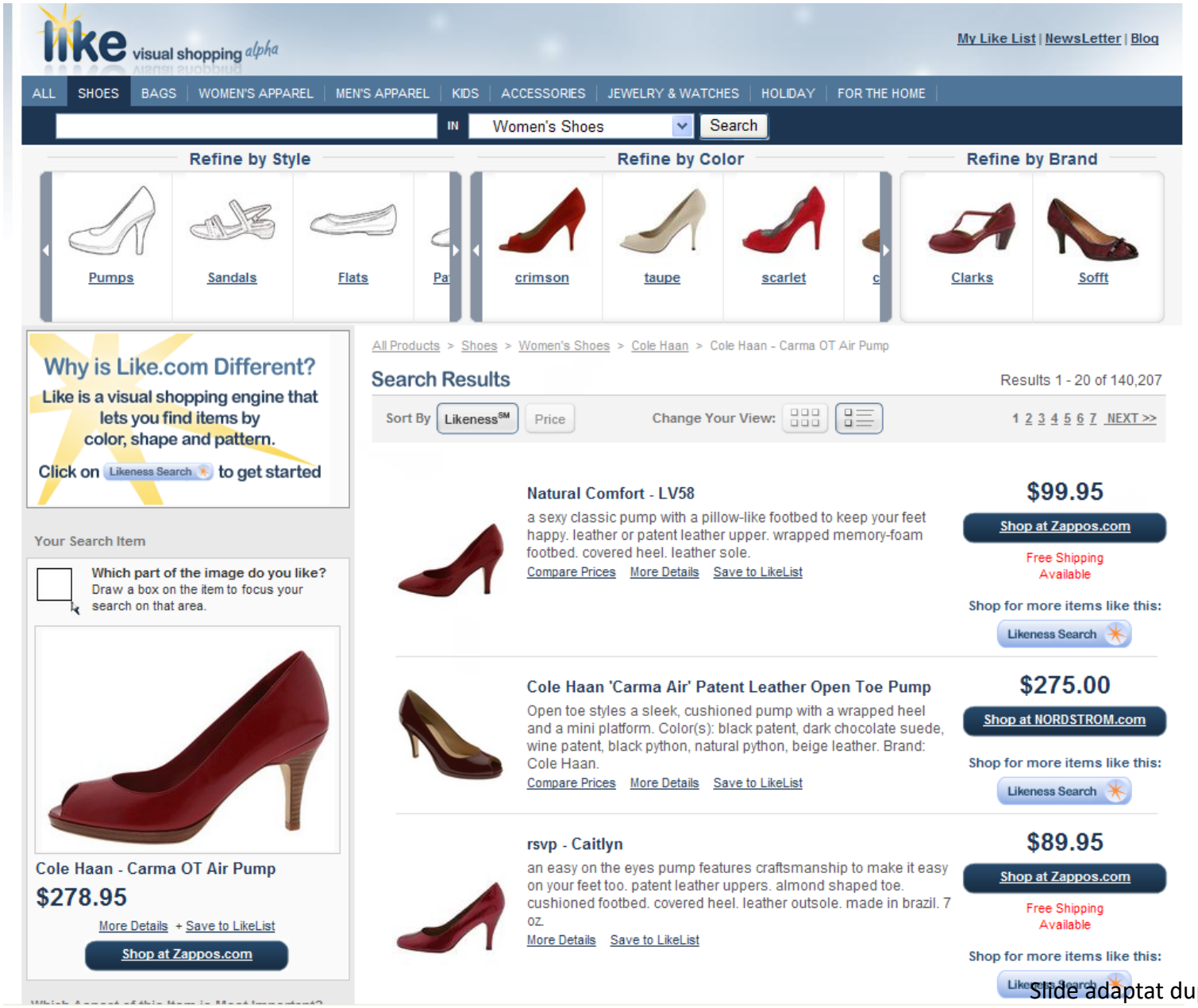


Detectarea liniilor de marcaj



Slide adaptat după S. Lazebnik

Găsirea obiectelor vizual similare

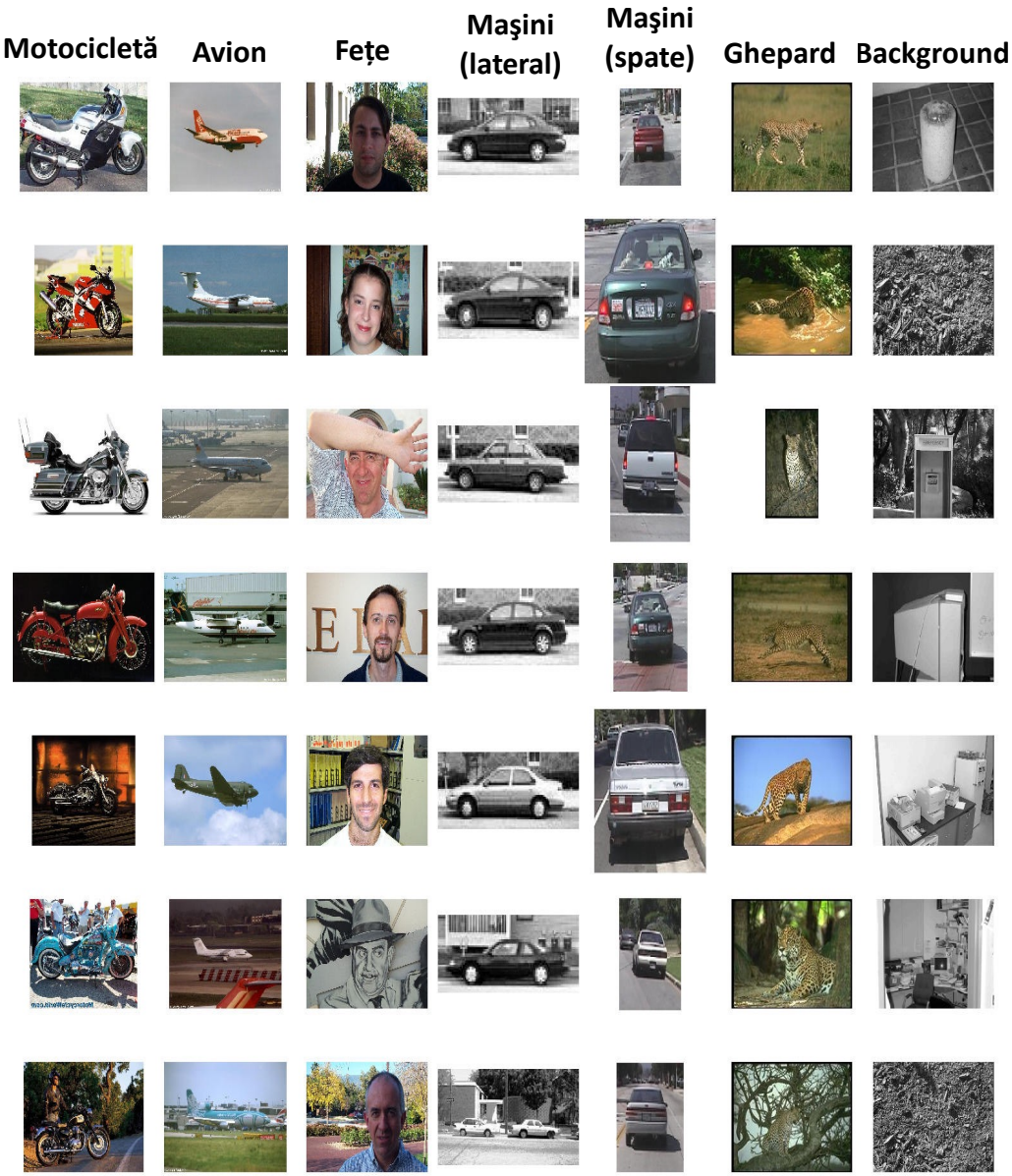


Slide adaptat după K. Grauman

Recunoaştere în Vederea Artificială

Dificultăți:

- Variabilitatea intra-clasă
- Clase de obiecte ce arată similar (maşină, camion, dubă)
- Variabilitate în mărime a obiectelor
- Confuzia cu backgroundul



Clasificarea imaginilor

- Conține o imagine test instanțieri ale unei clase de obiecte/categorii ?
  - răspuns binar: DA/NU

- Exemplu: clasificator de imagini pentru ‘câine’

DA

NU

DA

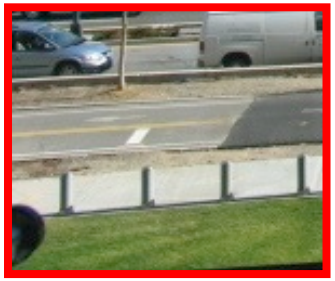
Clasificarea imaginilor

Învățăm un clasificator al imaginilor pe bază de exemple

Exemple pozitive: imagini care conțin câini




Exemple negative: imagini care NU conțin câini

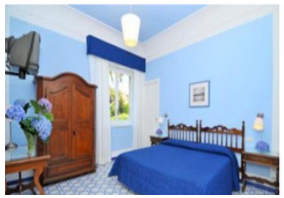
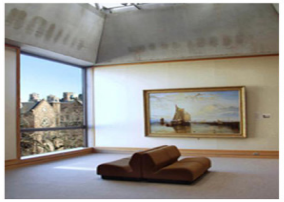
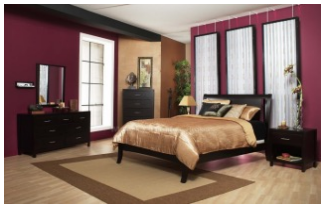
Alt exemplu: 2 clase - indoor/outdoor

Indoor – exemple







Outdoor - exemple









Clasificare = asocierea unei etichete (indoor, outdoor) pentru exemple noi

Imagini test:

outdoor

outdoor

indoor

Învățarea supervizată

Exemple de învățare

Etichete ale exemplelor de învățare

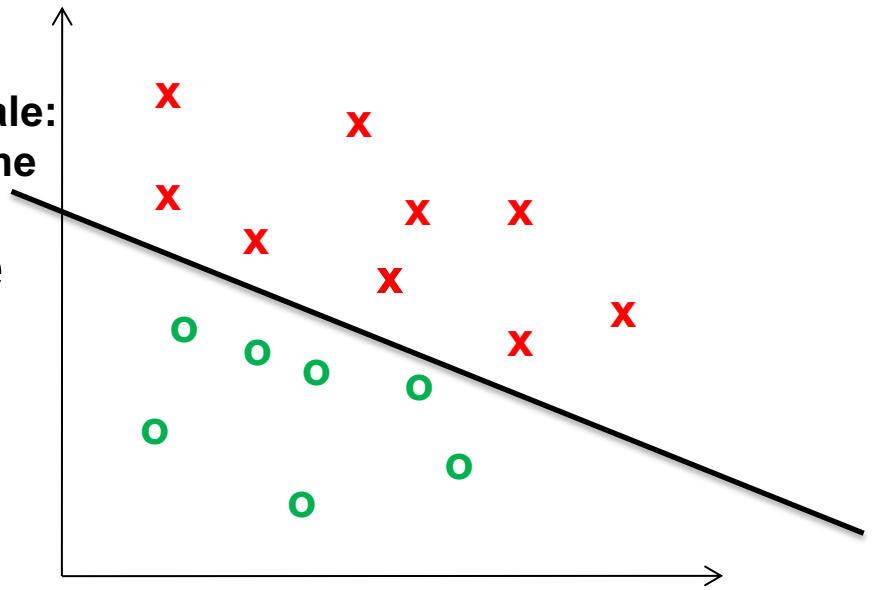
Extragere de caracteristici din imagini

Învățarea unui clasificator

Clasificator învățat

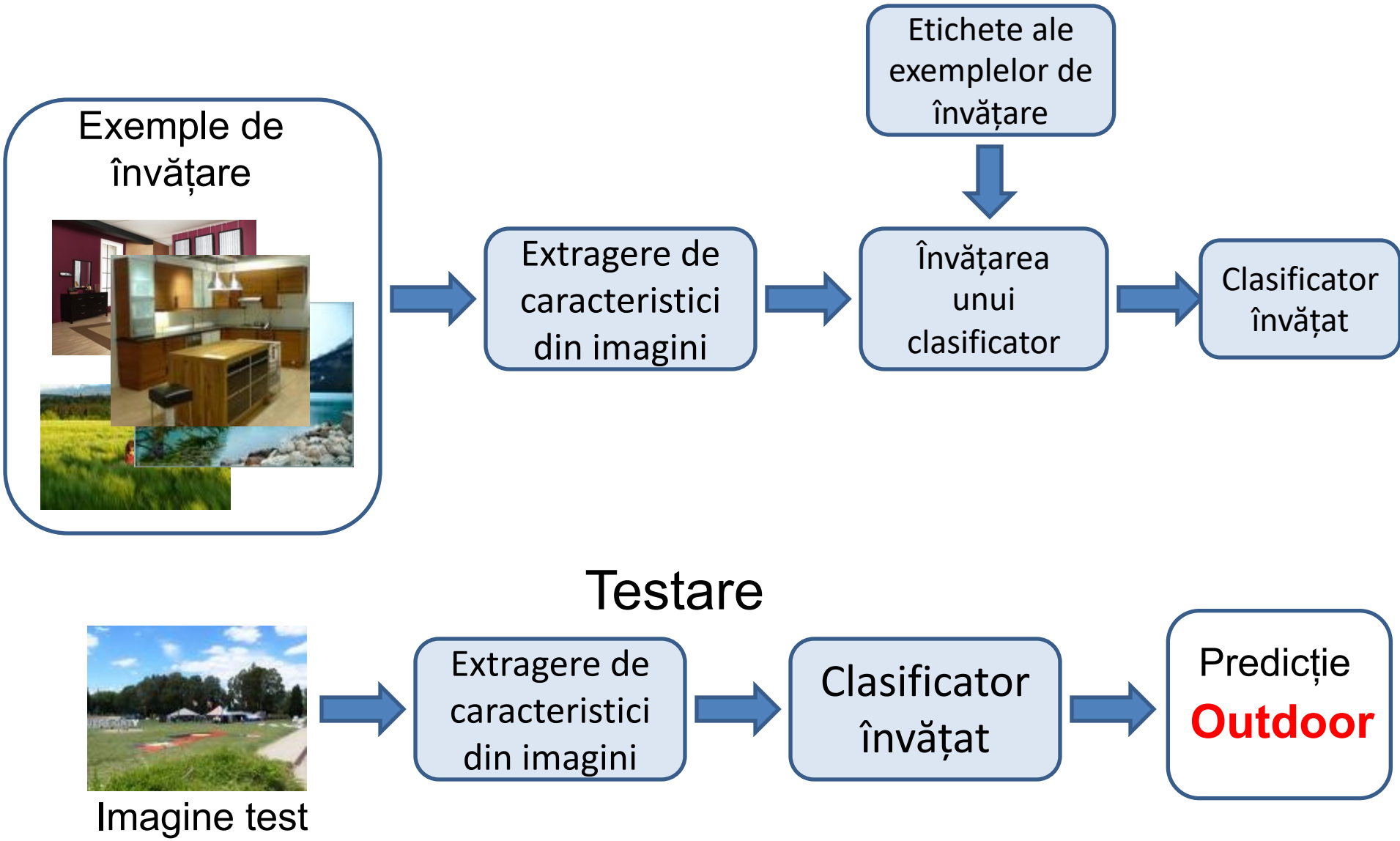
descriptori vizuali ai conținutului imaginii (puncte N-dimensionale: de exemplu histograme de culori)

Testare



Exemple
- **negative**
- **pozitive**

Etapa de testare



Extragere de caracteristici din imagini