



Technical University of Cluj - Napoca
Computer Science Department

Interactiune Om-Calculator

Curs 7-8

Detectia fețelor.

**Detectia componentelor faciale: Detectia si urmarirea
ochilor. Detectia clipitului**

Identificarea fetelor umane si localizarea lor in imagine, indiferent de:

- Pozitie
- Scala
- Rotatie (in planul imaginii)
- Orientare (rotatie in afara planului imaginii)
- Iluminare



Figure 3: Samples of detection results of faces of various poses.



Figure 4: Example of detecting different sized faces



Figure 1: Detection results of occluded faces

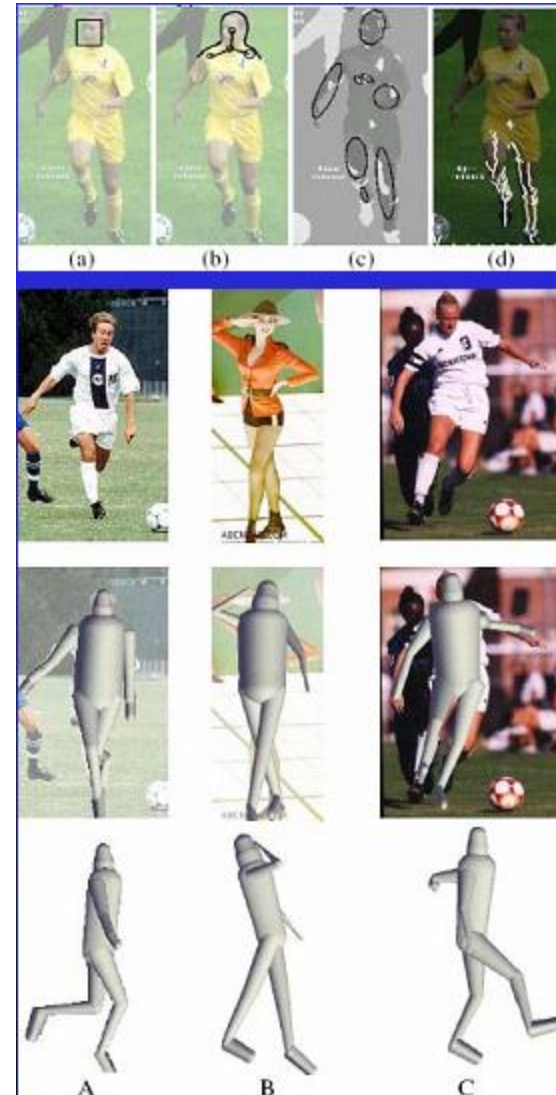


Figure 5: Detection result of a face with changes of expression



Utilitate

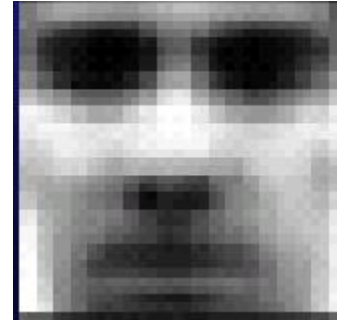
- Prima etapa in sistemele de recunoastere automata a fețelor
- Prima etapa in sistemele de “surveillance”
- Folosita ca ipoteza in identificarea persoanelor, a corpului uman, a partilor corpului uman
- Etapa de initializare in urmarirea (tracking-ul) fetelor sau a corpului in secvente de imagini
- Varietate mare de alte aplicatii





Dificultati

- Dimensionalitatea mare a spatiului se trasaturi:
Imagine grayscale (8 bits/pixel) de dim. 19×19
 $\Rightarrow 256^{361} = 2^{8 \times 361} = 2^{2888}$ combinatii posibile ale intensitatilor
- Orientarea fetei (rotatia in afara planului imaginii)
Axa capului verticala: frontal, profil, semiprofil
Axa capului non-verticala
- Rotatie (in planul imaginii / in jurul axei optice a camerei)
- Conditii variabile de achizitie a imaginilor:
iluminare, parametri camerei (senzor, castig, rezolutie imagine, obiectiv, umbre, ...)
- Ocluzii
- Expresia faciala: fata = obiect non-rigid cu aparenta variabila
- Prezenta / absenta elementelor structurale:
ochelari, barba, mustata etc.





Obiective

Probleme conexe

- **Localizarea fetei**
 - determinarea pozitiei unei singure fete intr-o imagine
- **Detectia elementelor faciale componente**
 - prezenta si locatia: ochi, sprancene, nas/nari, gura, buze, urechi etc.
- **Recunoasterea / identificarea fetelor**
- **Recunoasterea expresiei faciale**
- **Estimarea pozitiei corpului uman si urmarirea acestuia**

Teme de cercetare

- Reprezentarea fetei – cum se descrie o fata ?
- Scalarea – cum se rezolva problema scalei variabile ?
- Strategia de cautare – cum cautam o fata in imagine ?
- Viteza – cum marim viteza de procesare ?
- Precizia – localizare si rata de eroare a detectei (TPR, FPR, FNR)
- Postprocesare – combinarea rezultatelor detectiei



Metode de detectie si localizare a feței

1. Metode bazate pe cunostiinte:

- ex: relatii spatiale intre trasaturile faciale pe baza caracteristicilor antropomorfe

2. Bazate pe trasaturi invariante (la pozitie, orientare, perspectiva)

3. Metode bazate pe potrivirea de sabloane (template matching)

- sabloane pentru intreaga fata sau pentru anumite parti

4. Metode bazate pe aparente

- modele / template-uri invatate pe un set de imagini de antrenare care sa surprinda variabilitatea aprentelor faciale

Abordari specifice sursei de imagini:

- Imagini statice sau secvente video
- Imagini grayscale/color



Metode

Resurse

<http://www.facedetection.com/>

- BD de antrenament
- aplicatii
- documentatii



Metode bazate pe culoare

Se bazeaza pe distributia de culoare specifica culorii pielii (model de culoare)

- rasa/etnie
- conditii de iluminare

Spatii de culoare

- RGB, RGB normalizat, HSV, HIS, YCrCb, YIQ, UES, CIE, XYZ, CIE LIV

Analiza statistica

- histograma, LUT, model gausiene sau mixturi de modele gausiene



Metoda 1

Metoda de detectie a feței bazata pe culoare si potrivire de sabloane

A FACE DETECTION TUTORIAL:

In this tutorial, a simple face detection method is implemented by using Matlab 7.0.4. Several stages are involved in this method which includes skin detection, region analysis and template matching. The program implemented is not so good in terms of performance but hopefully this tutorial will give some basic idea of image processing for those who are new to this field.

Referinta si surse Matlab:

<http://se.cs.ait.ac.th/cvwiki/matlab:tutorial:detectface> (YCbCr)

Sau

<http://www-cs-students.stanford.edu/~robles/ee368/main.html> (rgb
normalizat)



Modelul de culoare YC_bC_r

1. Construirea unui model de culoare pentru piele

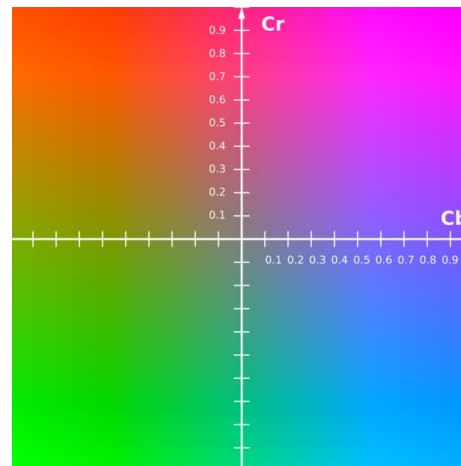
- Modele de culoare: rgb normalizat sau YC_bC_r

<http://en.wikipedia.org/wiki/YCbCr>

Y – luminanta; C_b , C_r – componente cromatice (diferenta albastru/ rosu)

$$\begin{pmatrix} Y \\ C_B \\ C_R \end{pmatrix} = \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} + \frac{1}{256} \begin{pmatrix} 65.738 & 129.057 & 25.064 \\ -37.945 & -74.494 & 112.439 \\ 112.439 & -94.154 & -18.285 \end{pmatrix} \cdot \begin{pmatrix} R_N \\ G_N \\ B_N \end{pmatrix}$$

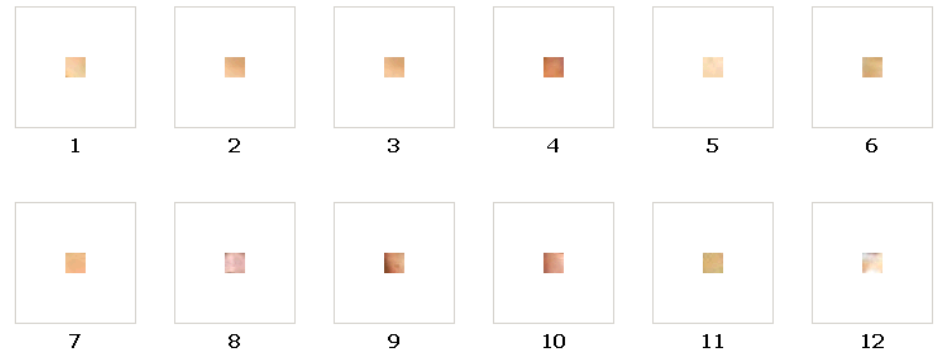
$$\begin{pmatrix} R_N \\ G_N \\ B_N \end{pmatrix} = \begin{pmatrix} 0.842 & 0.156 & 0.091 \\ -0.129 & 1.320 & -0.203 \\ 0.008 & -0.069 & 0.897 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$





Modelul de culoare pentru piele

Set de imagini de test –
se aleg zone de piele (ex:
16 x 16)



Histograma 2D pt. fiecare zona \Rightarrow Histograma cumulativa (model gaussian)

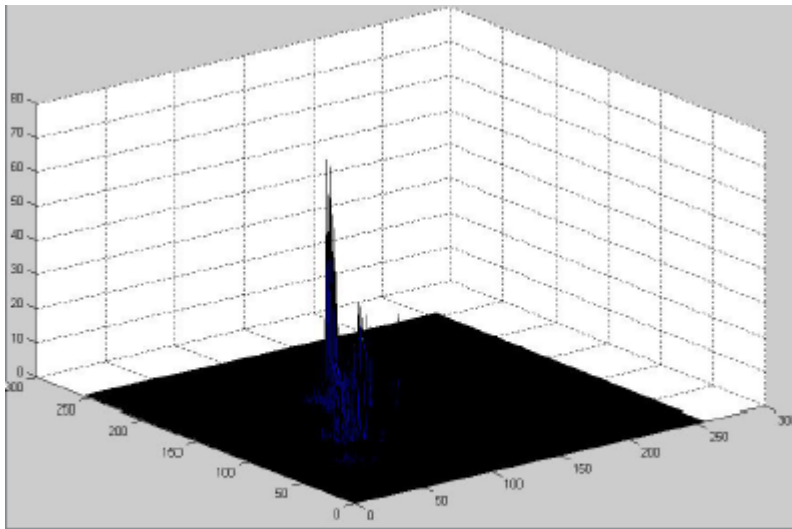


Fig. 1. 2D Histogram

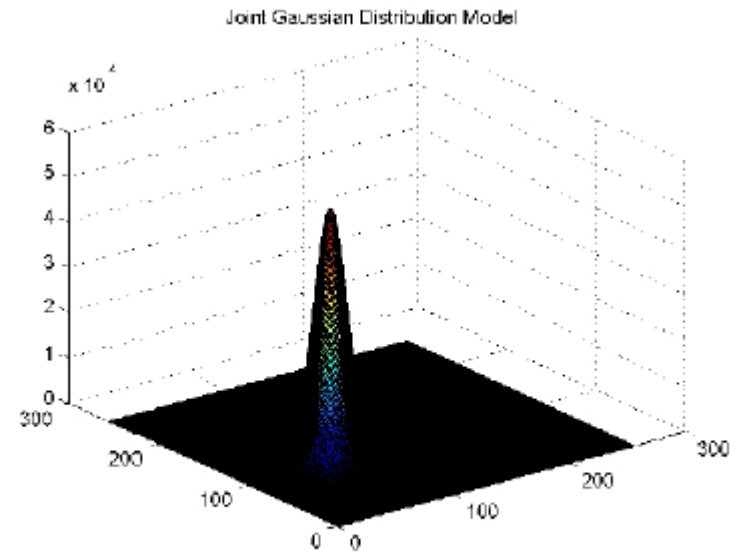


Fig. 2. Gaussian Skin Model

Modelul gaussian (Fig. 2): medie si covarianta (CbCr)



Media si covarianta

Caracterizarea statistica a variabilelor aleatoare

1. Expectation: reprezinta centrul de masa al unei densitati de probabilitate.

$$E[X] = \mu = \int_{-\infty}^{+\infty} x f_X(x) dx$$

2. Varianta: reprezinta "imprastierea" in jurul mediei

$$VAR[X] = E[(X - E[X])^2] = \int_{-\infty}^{+\infty} (x - \mu)^2 f_X(x) dx \quad STD[X] = VAR[X]^{1/2}$$

Caracterizarea statistica a vectorilor aleatori

Putem descrie partial un vector aleator prin urmatoarele valori:

1. Vectorul mediu:

$$E[X] = [E[X_1] E[X_2] \dots E[X_N]]^T = [\mu_1 \mu_2 \dots \mu_N] = \mu$$

2. Matrticea de covarianta

$$\begin{aligned} COV[X] = \Sigma &= E[(X - \mu)(X - \mu)^T] \\ &= \begin{bmatrix} E[(x_1 - \mu_1)(x_1 - \mu_1)] & \dots & E[(x_1 - \mu_1)(x_N - \mu_N)] \\ \dots & \dots & \dots \\ E[(x_N - \mu_N)(x_1 - \mu_1)] & \dots & E[(x_N - \mu_N)(x_N - \mu_N)] \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \dots & c_{1N} \\ \dots & \dots & \dots \\ c_{1N} & \dots & \sigma_N^2 \end{bmatrix} \end{aligned}$$



Matricea de covarianta

Matricea de covarianta indica tendinta fiecarei perechi de trasaturi (pozitii in vector) sa varieze impreuna, sau sa co-varieze. Covarianta are cateva proprietati importante:

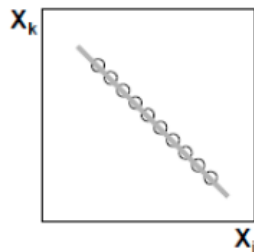
- Daca \mathbf{x}_i si \mathbf{x}_k cresc impreuna, atunci $\mathbf{c}_{ik} > 0$
- Daca \mathbf{x}_i tinde sa descreasca atunci cand \mathbf{x}_k creste, atunci $\mathbf{c}_{ik} < 0$
- Daca \mathbf{x}_i si \mathbf{x}_k sunt necorelate, atunci $\mathbf{c}_{ik} = 0$
- $|\mathbf{c}_{ik}| < \sigma_i \sigma_k$, unde i este deviatia standard a lui \mathbf{x}_i
- $\mathbf{c}_{ii} = \text{VAR}(\mathbf{x}_i)$

Termenii matricii de covarianta pot fi scrisi ca:

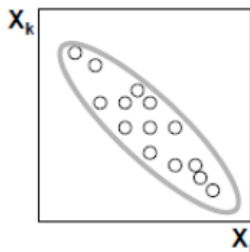
$$c_{ik} = E[(x_k - \mu_k)(x_i - \mu_i)]$$
$$c_{ii} = \sigma_i^2 \text{ and } c_{ik} = \rho_{ik} \sigma_i \sigma_k$$

unde ρ_{ik} este numit **coeficientul de corelatie**.

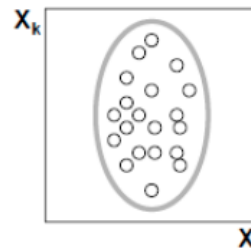
Exemple:



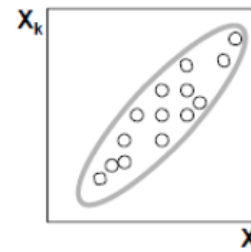
$$C_{ik} = -\sigma_i \sigma_k$$
$$\rho_{ik} = -1$$



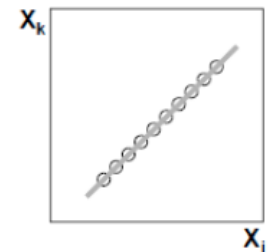
$$C_{ik} = -1/2 \sigma_i \sigma_k$$
$$\rho_{ik} = -1/2$$



$$C_{ik} = 0$$
$$\rho_{ik} = 0$$



$$C_{ik} = +1/2 \sigma_i \sigma_k$$
$$\rho_{ik} = +1/2$$



$$C_{ik} = \sigma_i \sigma_k$$
$$\rho_{ik} = +1$$



2. Segmentarea zonelor



Fig. 3. Original Image



Fig. 4. Likelihood Image

Pentru fiecare pixel se calculeaza distanta (mahalanobis) la media distributiei gaussiene antrenate \Rightarrow normalizare (0 ..1) \Rightarrow “skin likelihood” si reprezentare ca o imagine grayscale (Fig. 4)

Formally, the Mahalanobis distance of a multivariate vector $x = (x_1, x_2, x_3, \dots, x_N)^T$ from a group of values with mean $\mu = (\mu_1, \mu_2, \mu_3, \dots, \mu_N)^T$ and covariance matrix S is defined as:

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)}.^{[2]}$$

Daca S (cov.) diagonala:

$$d(\vec{x}, \vec{\mu}) = \sqrt{\sum_{i=1}^N \frac{(x_i - \mu_i)^2}{\sigma_i^2}}$$



Metoda 1

3. Binarizare adaptiva



Fig. 5. Skin Segmented Image

4. Operati morfologice

- umplere de regiuni, eroziune si dilatare, “and” logic cu img. pas 3



Fig. 6. Eroded Image



Fig. 7. Dilated Image



Metoda 1

5. Etichetare

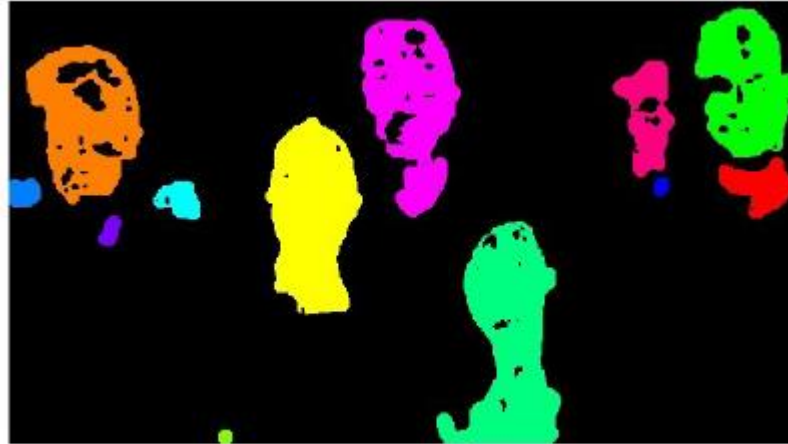


Fig. 8. Colored Labeled Region

6. Calcul nr. Euler (fiecare eticheta) \Rightarrow se retin doar regiuni cu nr. Goluri $= 1 - nr_E \geq 1$

Nr_E = total number of objects in the region minus the total number of holes

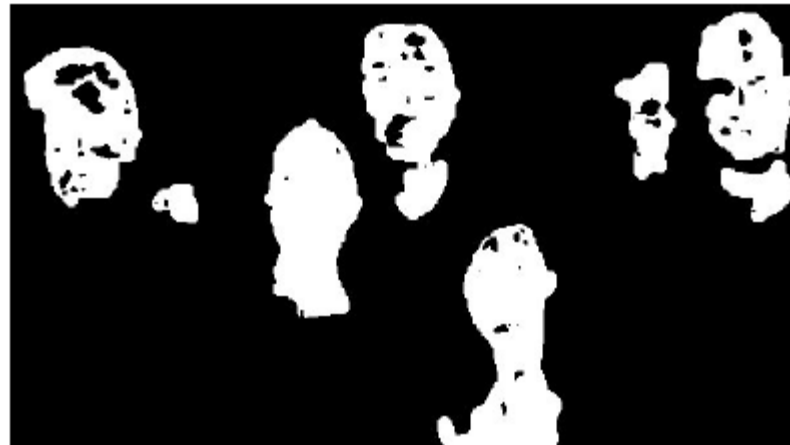


Fig. 9. After Euler Test

7. Calcul factor de alungire (aspect ratio - AR)

- $AR = \text{lg. axa majora} / \text{lg. axa minora}$

- se retin obiecte cu $AR = 1 \dots 3,5$



Fig. 10. After Aspect Ration Test

8. Template matching

- template: **imaginea medie la N fete** (grayscale)
- pt. fiecare regiune candidat se calculeaza dimensiunea si orientarea (axa de alungire)
- se compara (cross-correlation) cu img. template-ului scalata si rotita corespunzator

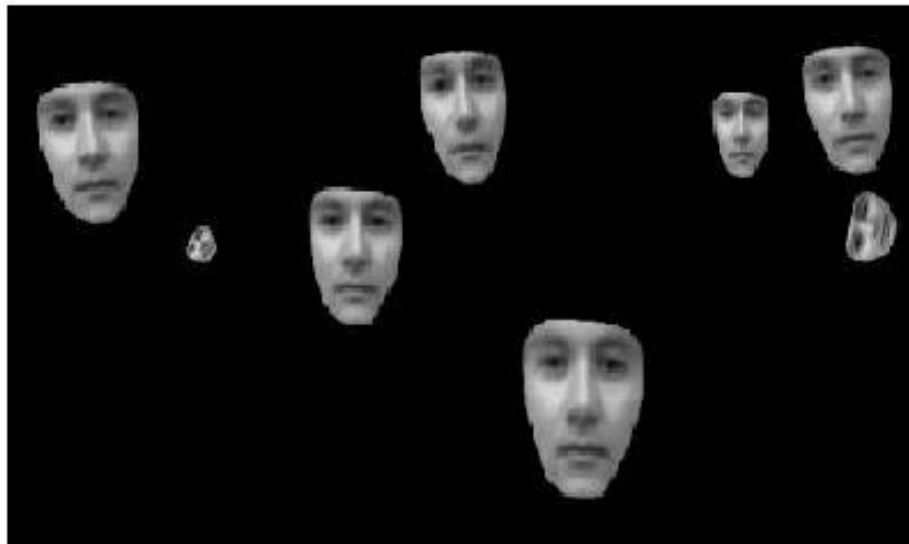


Fig. 11. Template Matching



Metoda 1

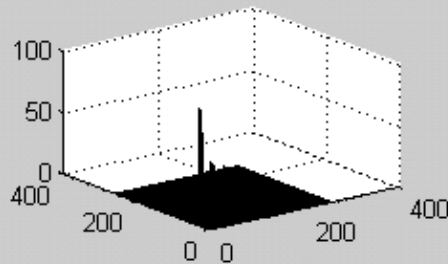
9. Rezultate



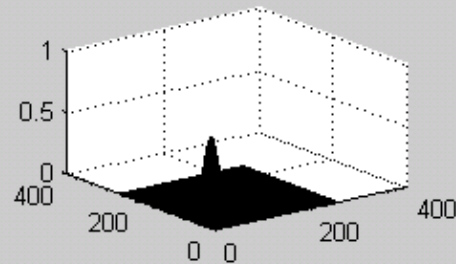
Fig. 12. Face Detection



2D Chromatic Histogram Model



Joint Gaussian Distribution Model



Original RGB Image



Exemple rulare:

face_detection

('testset/blackgirl.jpg')

Skin Likelyhood Image



Skin Segmented Image



After Erosion
(disk size: 10)



After Dilation
(disk size: 8)



Labeled Regions
(5 regions)



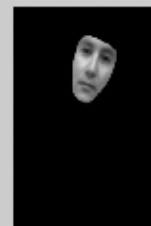
After Euler Test
(1 regions)



After Aspect Ratio Test
(1 regions)



Template Matching



Final Detection

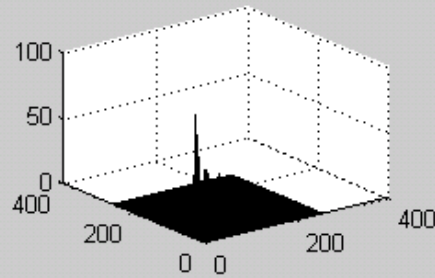




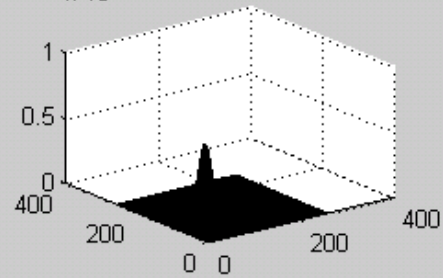
Exemple rulare:

face_detection
(`'testset/family.jpg'`)

2D Chromatic Histogram Model



Joint Gaussian Distribution Model



Original RGB Image



Skin Likelihood Image



Skin Segmented Image



After Erosion
(disk size: 10)



After Dilation
(disk size: 8)



Labeled Regions
(5 regions)



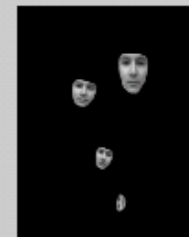
After Euler Test
(4 regions)



After Aspect Ratio Test
(4 regions)



Template Matching



Final Detection





Metoda 2

**“A Real-Time Vision Interface Based on Gaze Detection — EyeKeys”,
John J. Magee**

Referinte

B. Kisacanin, V. Pavlovic, T.S. Huang, Real-Time Vision for Human-Computer Interaction, *Springer 2005*, pp. 141-157.

- **Metoda de detectie a feței bazata pe culoare si potrivire de sabloane**

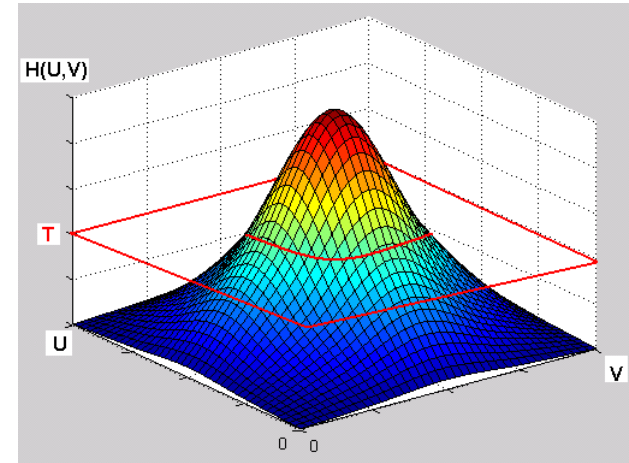


Metoda 2

Scop: gasirea unei masti pt. fata (zonei cu piele)

Spatiul de culoare YUV:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



Gasirea mastii:

- Se considera un set de 15 imagini de antrenare (aceiasi persoana, conditii similare de iluminare). La schimbarea conditiilor \Rightarrow reantrenare
- Se selecteaza manual zone de pe fata (piele) si realizeaza histograma UV pt. aceste zone si apoi se face histograma cumulativa pt. 15 imagini.
- Se stabileste o valoare de prag T
- Pentru fiecare pixel din imaginea curenta, daca valoarea (U,V) corespunde unui punct din histograma care depaseste valoarea T se marcheaza ca punct de fata/masca
- Masca va fi o imagine binara (alb – fata, negru – in rest)

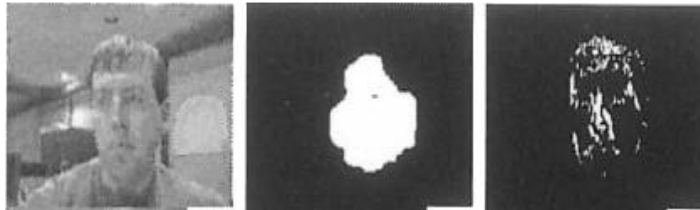




Metoda 2

Diferente intre cadre succesive (camera fixa)

- Daca exista miscare \Rightarrow diferentele sunt evidentiata prin tresholding (in special in zonele cu gradient nenul) \Rightarrow masca a miscarii:



Masca
culoare

Masca
miscare

-Daca nu exista miscare \Rightarrow se foloseste ca masca zona detectata a fetei din imaginile precedente

Detectia fetelor

– potrivire de sabloane prin cautare piramidala

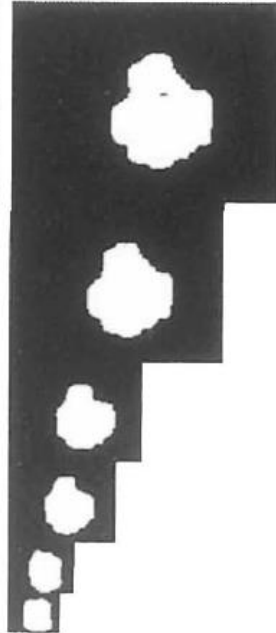


Metoda 2

Scalarea imaginilor si a sabloanelor pt. detectia fetei



P_Y



P_{color_mask}



P_{motion_mask}



$P_{correlation}$



$P_{correlation} \cap P_{color_mask} \cap P_{motion_mask}$

Nivel	Latime	Inaltime	Factor de scalare
0	640	480	1
1	320	240	2
2	160	120	4
3	128	96	5
4	80	60	8
5	64	48	10
6	40	30	16
7	32	24	20

Sablon:

- dim. fixa (12x16)
- medie a intensitatii (Y) pt. un set de antrenare de 8 fete



Metoda 2

Generarea unei piramide gaussiene (factor de decimare 2)

Convolutie cu gaussian de dimensiune fixa

Original Image
(Level 0)

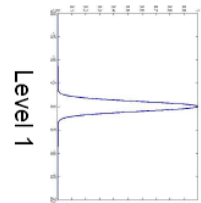
$$G_l(i,j) = \sum_m \sum_n w(m,n) G_{l-1}(2i+m, 2j+n)$$

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1



*

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

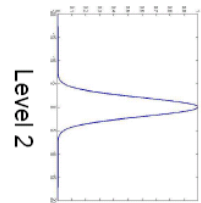


Level 1

Decimation
(Level 1)



Filter

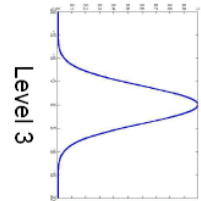


Level 2

Decimation
(Level 2)



Subsample



Level 3

Decimation
(Level 3)



Technical Univer
Computer S

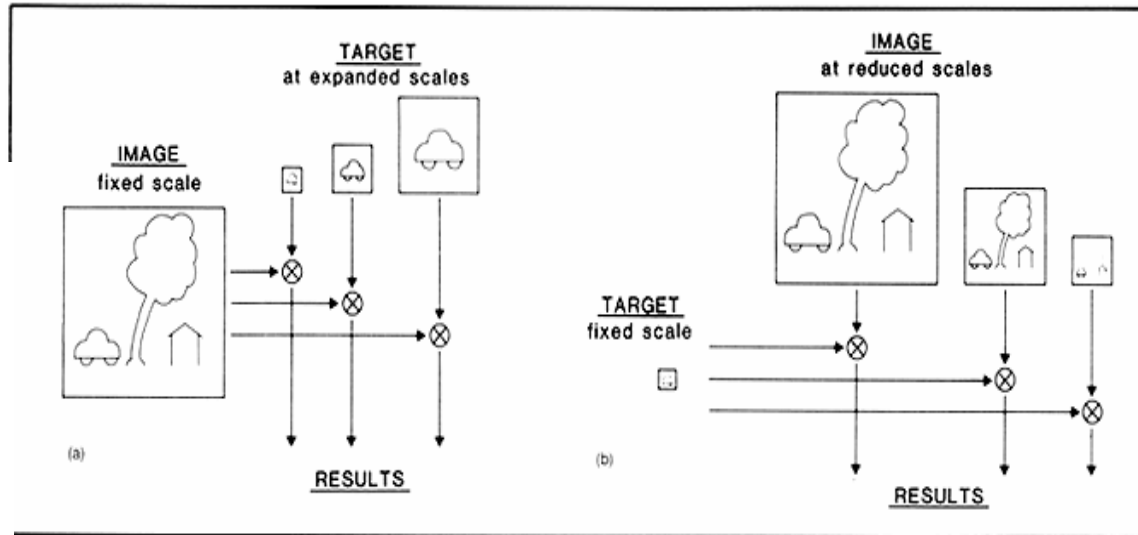
a
it



Metoda 2

Scalarea imaginilor si a sabloanelor

Template



Search Region

Original Image



Metrica de corelatie

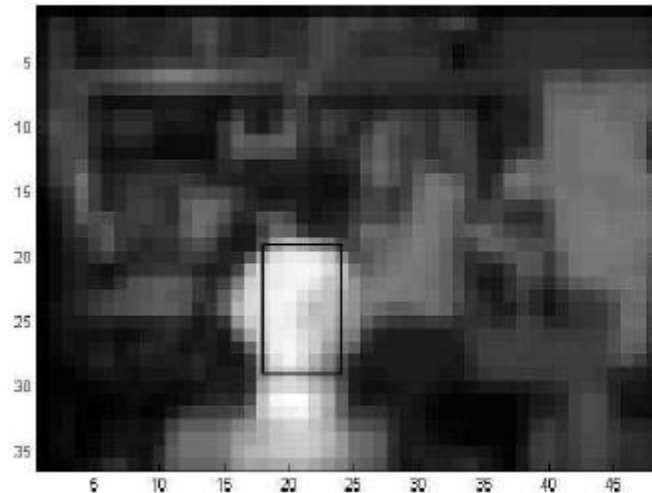
- SAD, SSD etc

$$SAD(x, y_R) = \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} |I(x+i, y+j) - T(i, j)|$$

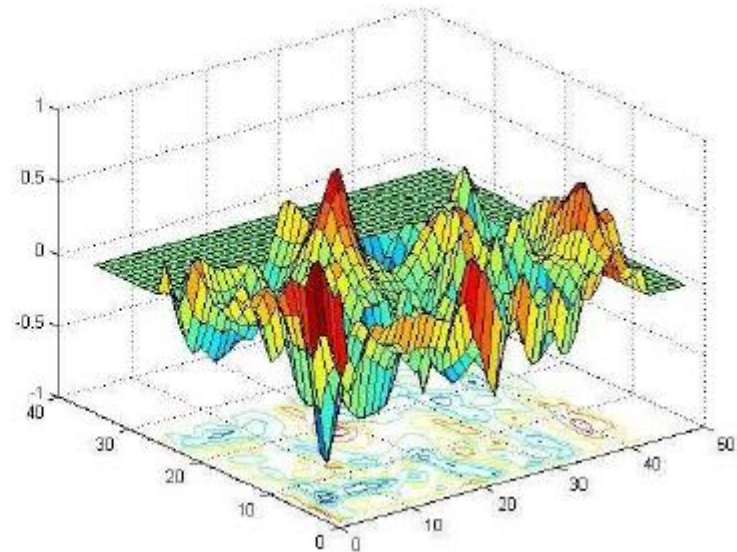
Potrivirea sablonului fetei

Reducerea spatiului de cautare:

- Cautarea incepe de la nivelul cel mai de jos al piramidei (ex. Level 3)
- La acest nivel se incerca corelarea templetului corespunzator peste intreaga imagine



Pozitia (xm, ym) din imagine
corespunzatoare Min(SAD)



SAD

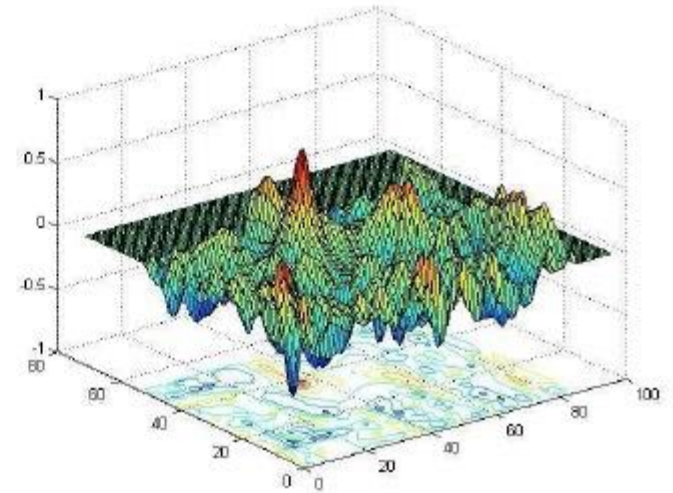
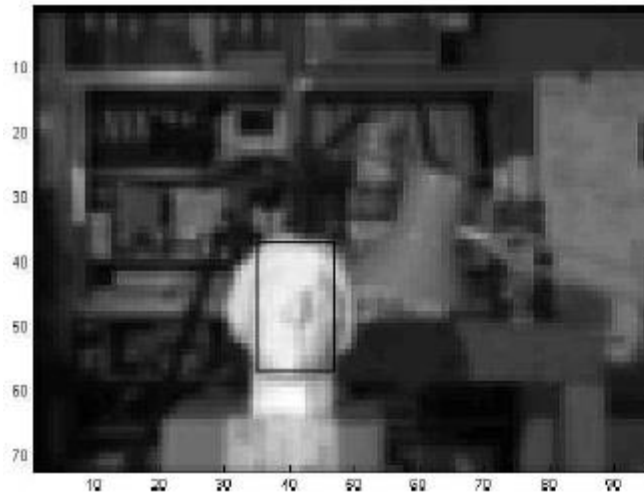


Metoda 2

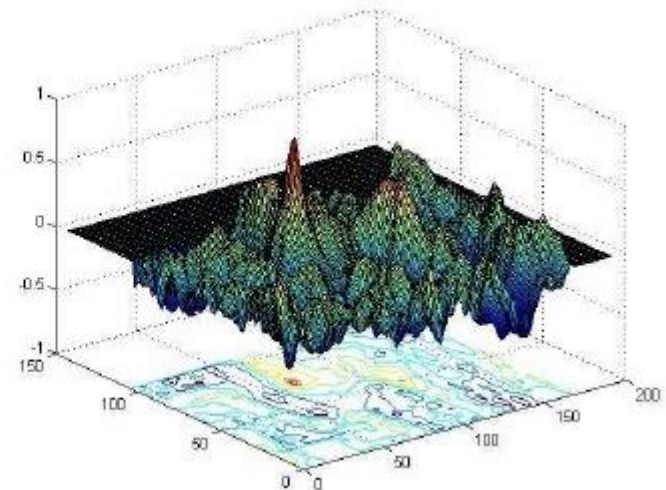
Potrivirea sablonului fetei

Cautarea in nivelul urmator (ex. Level 2) se face doar intr-o vecinatate a pozitiei minimului de la nivelul de mai jos:

Nivel 2



Nivel 1





Detectia fetelor in OpenCV

Resurse

http://docs.opencv.org/2.4.11/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html#cascade-classifier

<http://www.mathworks.com/matlabcentral/fileexchange/19912> (Matlab)

<http://www.cs.princeton.edu/courses/archive/fall08/cos429/CourseMaterials/Precept1/facedetect.pdf>

<http://note.sonots.com/SciSoftware/haartraining.html> (training)

Ming-Hsuan Yang, David J. Kriegman, Narendra Ahuja, Detecting Faces in Images: A Survey, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 1, JANUARY 2002 <http://vision.ai.illinois.edu/publications/pami02a.pdf>

Demo

<http://vimeo.com/12774628> (Haar cascade classifier demo)

Bibliografie

<https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>

<http://cs.nyu.edu/~eugenew/publications/viola-facedet04-talk.pdf>

[http://cs.nyu.edu/courses/fall12/CSCI-GA.2560-](http://cs.nyu.edu/courses/fall12/CSCI-GA.2560-001/FaceRecognitionBoosting.pdf)

[001/FaceRecognitionBoosting.pdf](http://cs.nyu.edu/courses/fall12/CSCI-GA.2560-001/FaceRecognitionBoosting.pdf)



Metoda Viola & Jones

Etape/caracteristici:

1. Extragere trasaturi rectangulare
2. Clasificare prin “boosting”
3. Algoritm de detectie multi-rezolutie

Trasaturi:

- rectangulare (grayscale) \approx Haar
- “image integrale” (marirea vitezei de calcul)

Clasificare:

- trasaturile \Rightarrow clasificatori slabi
- cascada de clasificatori slabi (concatenare seriala)
- Selectie trasaturi: “boosting” anumite trasaturi relevante (AdaBoost) \Rightarrow detector suficient de bun

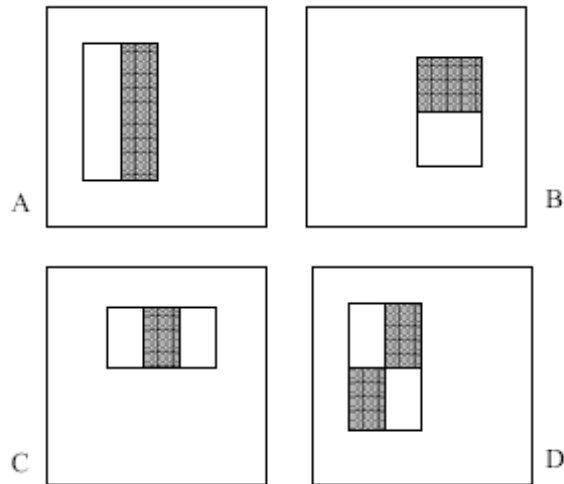
Viteza: 700Mhz Pentium III, 384x288 \approx 0.067 sec.



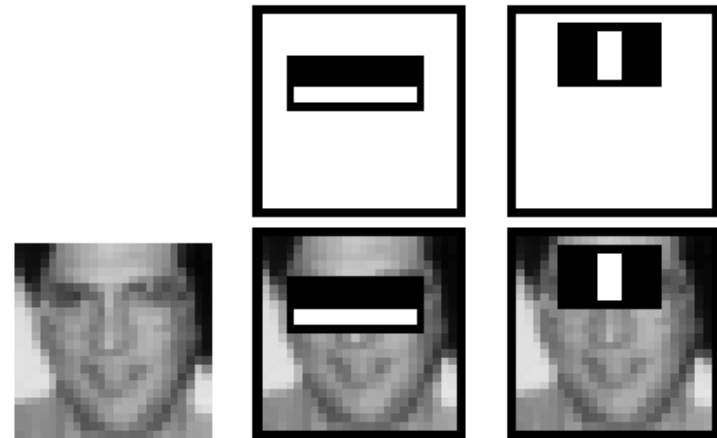
Metoda Viola & Jones

Trasaturi rectangulare

- 4 tipuri de masti generale:



Cele mai relevante::



Calcul:

- Se suprapun mastile in diverse pozitii / rezolutii peste imaginea grayscale
- Pt. fiecare pozitie/sablon se calculeaza:

$$Value = \sum (intensity\ in\ white\ area) - \sum (intensity\ in\ black\ area)$$

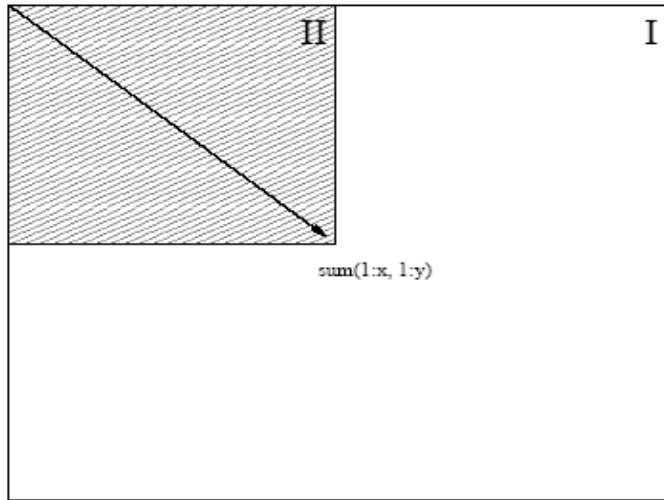
- La scala minima pt. o fata (24 x 24 pixeli) \Rightarrow set trasaturi rectangulare
~160,000 !!! \Rightarrow selectie de trasaturi relevante



Metoda Viola & Jones

Optimizare calcul trasaturi rectangulare

- Imagine INTEGRALA (a imaginii sursa / de intrare)



$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Imaginea integrala se poate obtine prin parcurgerea (o singura data) a imaginii initiale

$$s(x, y) = s(x, y - 1) + i(x, y)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y)$$

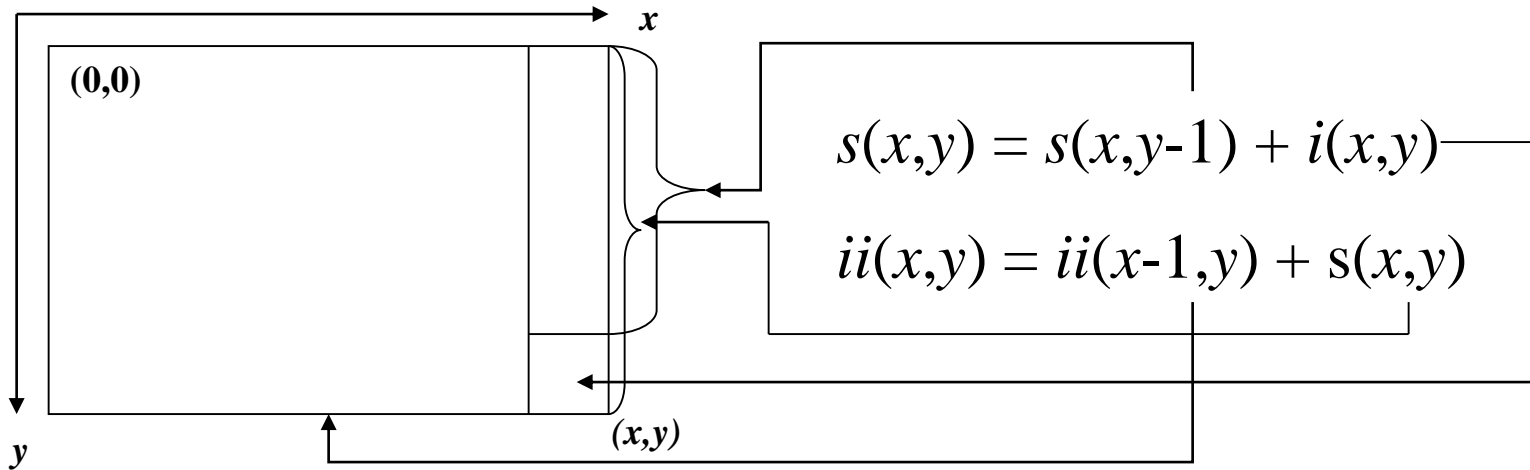
$$s(x, -1) = 0$$

$$ii(-1, y) = 0$$



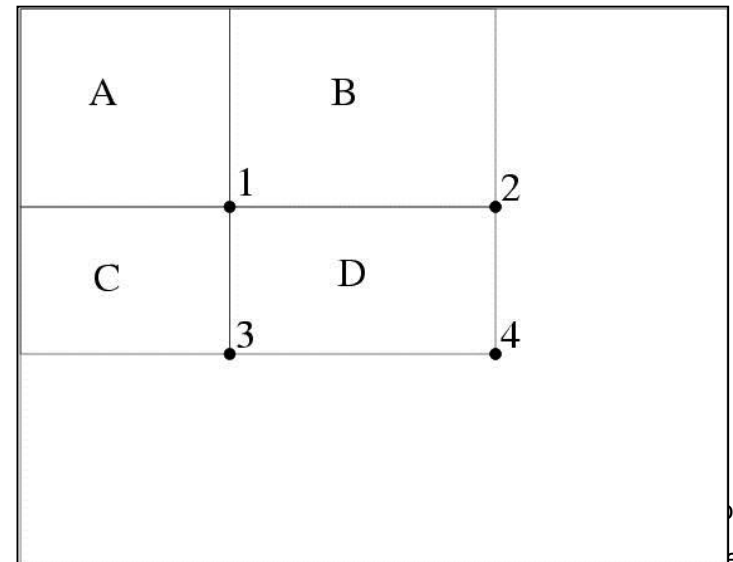
Metoda Viola & Jones

Calculul imaginii INTEGRALE



Calculul intensitatii pixelilor din dreptunghiul D (se cunosc A, A+B si A+C = imaginile integrale din pixelii 1, 2 respectiv 3):

$$\begin{aligned} D &= ii(4) - (A+B+C) \\ &= ii(4) - \{ii(1) + [ii(2)-ii(1)] + [ii(3)-ii(1)]\} \\ &= ii(4) + ii(1) - ii(2) - ii(3) \end{aligned}$$



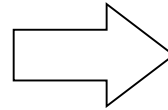


Metoda Viola & Jones

Imaginea INTEGRALA

Original

0	1	1	1
1	2	2	3
1	2	1	1
1	3	1	0



INTEGRALA

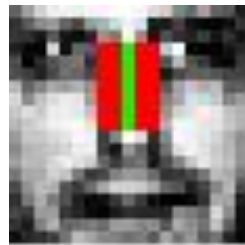
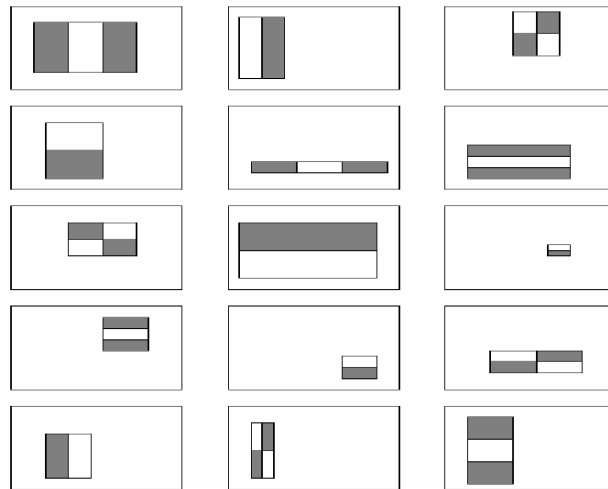
0	1	2	3
1	4	7	11
2	7	11	16
3	11	16	21



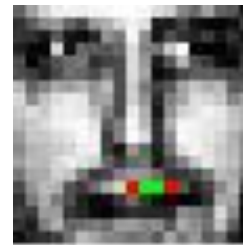
Metoda Viola & Jones

Trasaturi rectangulare

- La scala minima pt. o fata (24 x 24 pixeli) \Rightarrow set trasaturi rectangulare
~160,000 !!!



Trasatura relevanta



Trasatura nerelevanta



Metoda Viola & Jones

Selectia trasaturilor (relevante)

Selectia unui subset de trasaturi relevante \Rightarrow ADAptiveBOOSTing:

- Construirea unui clasificator "puternic" combinarea linear-ponderata a unor clasificatori simplii ("slabi")

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \dots$$

Clasificator
puternic

Pondere

Clasificator slab

Trasaturi rectangulara calculate
pe exemple + sau -

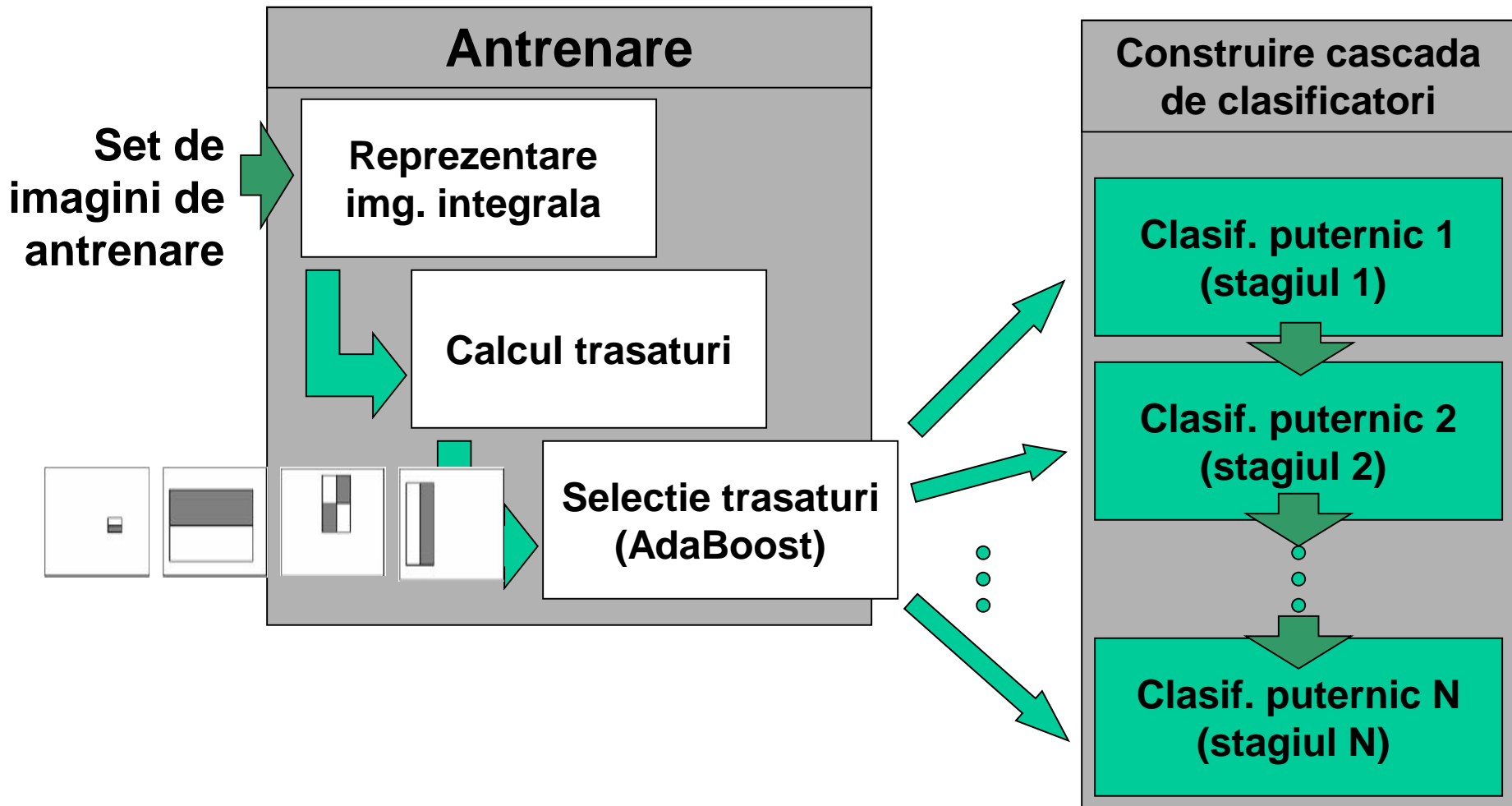
Proces iterativ de selectie trasaturilor rectangulare (pozitie, forma, scala) si a ponderilor \Rightarrow clasificator puternic format din T clasificatori slabi:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases} \quad h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$



Metoda Viola & Jones

Realizarea unei cascade de clasificatori

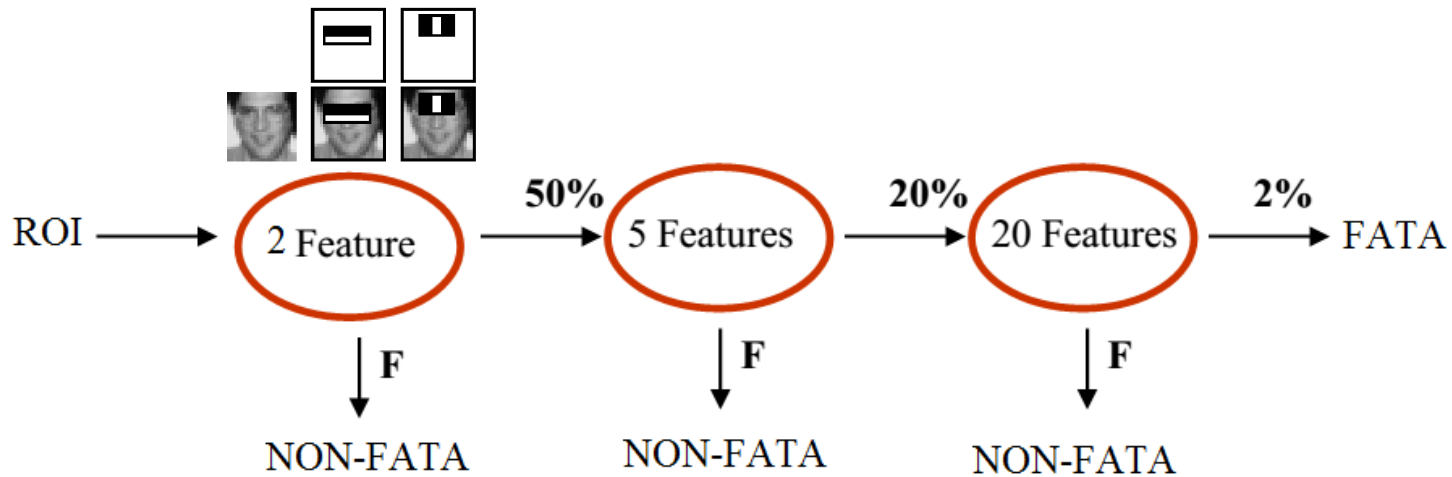




Metoda Viola & Jones

Realizarea unei cascade de clasificatori

- Foarte putine sub-regiuni dintr-o imagine contin fete
- Se insista cautarea doar in regiunile care potential contin fete. Restul se rejecteaza de la inceput (stadiu incipient)



- Cu un clasificator puternic cu cele mai simple dar relevante 2 trasaturi se poate atinge o rata TP=100% si FP=50% (stagiul 1)
- Clasificatorul de pe stagiul 2 (ex. 5 trasaturi) va trebui sa elimine din cazurile dificile (FP) de la stagiul anterior obtinand un FP = 40% (20% acumulat)
- Clasificatorul de pe stagiul 3 (ex. 20 trasaturi) va trebui sa elimine din cazurile si mai dificile (FP) de la stagiul anterior obtinand un FP = 10% (2% acumulat)



Metoda Viola & Jones

Rezultate si demo

- Strategia de cautare: multirezolutie (scale diferite)
- Se detecteaza multe instante/detectii pozitive intr-o regiune
- Pozitia finala a fetei \Rightarrow non-maxima supresion pe rezultatul clasificarii pe fiecare instana/detectie pozitiva

<http://vimeo.com/12774628> (demo)

<https://medium.com/swlh/haar-cascade-classifiers-in-opencv-explained-visually-f608086fc42c>

Imbunatatiri / extensii:

- Rotatii: in plan si in afara planului (“multi-pose”)
- Ocluzii
- Componente faciale si parti coprorale (OpenCV)



Metoda Viola & Jones - OpenCV

Clasificatori HAAR in OpenCV pt. detectia fetei, componentelor faciale sau parti corporale:

`%OPENCV_DIR%\data\haarcascades\`

`haarcascade_eye.xml`

`haarcascade_eye_tree_eyeglasses.xml`

`haarcascade_frontalface_alt.xml`

`haarcascade_frontalface_alt_tree.xml`

`haarcascade_frontalface_alt2.xml`

`haarcascade_frontalface_default.xml`

`haarcascade_fullbody.xml`

`haarcascade_lefteye_2splits.xml`

`haarcascade_lowerbody.xml`

`haarcascade_mcs_eyepair_big.xml`

`haarcascade_mcs_eyepair_small.xml`

`haarcascade_mcs_leftear.xml`

`haarcascade_mcs_lefteye.xml`

`haarcascade_mcs_mouth.xml`

`haarcascade_mcs_nose.xml`

`haarcascade_mcs_rightear.xml`

`haarcascade_mcs_righteye.xml`

`haarcascade_mcs_upperbody.xml`

`haarcascade_profileface.xml`

`haarcascade_righteye_2splits.xml`

`haarcascade_upperbody.xml`

Alti clasificatori pt. detectia de obiecte cu ajutorul metodei `detectMultiScale ()`:

`%OPENCV_DIR%\data\lbpascades :`

`lbpascade_frontalface.xml`

http://en.wikipedia.org/wiki/Local_binary_patterns

`%OPENCV_DIR%\data\hogcascades`

`hogcascade_pedestrians.xml`

http://en.wikipedia.org/wiki/Histogram_of_oriented_gradients



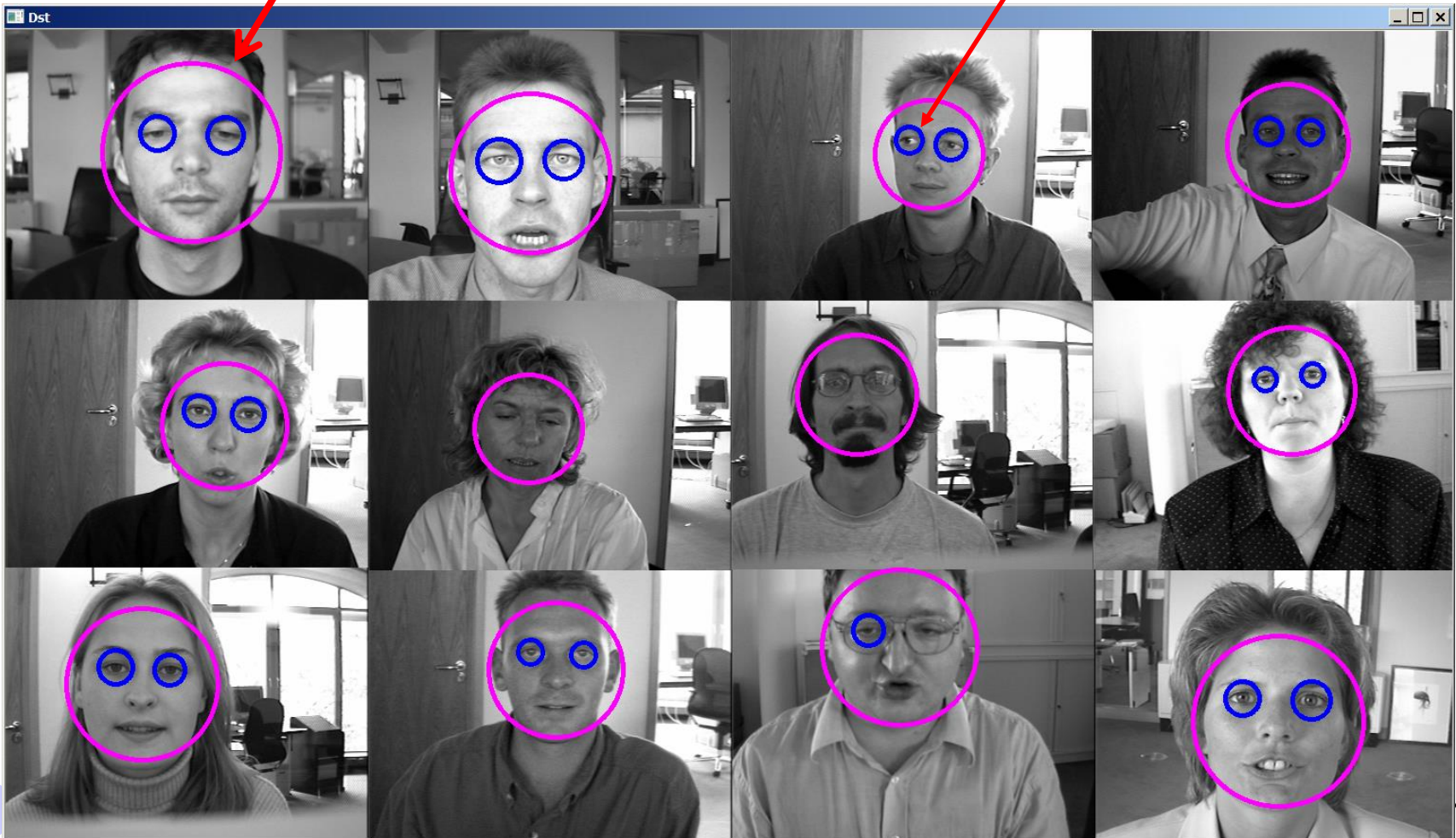
Ex. OpenCV: detectMultiScale()

Sursa imaginii: BioID Face Database

(<http://www.bioid.com/downloads/software/bioid-face-database.html>)

haarcascade_frontalface_alt.xml

haarcascade_eye_tree_eyeglasses.xml





Detectia de compnente faciale

Detectia ochilor si urmarirea directiei privirii



Detectia de compnente faciale

Detectia ochilor si urmarirea directiei privirii

Cerinte:

hardware minimal (Web-cam +PC)

resurse de calcul minimale (timp de procesare)

Aplicatii:

- interfata pentru persoane cu dizabilitati/paralizie

Referinte

B. Kisacanin, V. Pavlovic, T.S. Huang, Real-Time Vision for Human-Computer Interaction, *Springer 2005*, pp. 141-157, “A Real-Time Vision Interface Based on Gaze Detection — EyeKeys”, John J. Magee



Detectia de componente faciale

Detectia ochilor si urmarirea directiei privirii

Metode alternative:

- Camere montate pe cap sau ochelari care urmaresc miscarea iris-ului
⇒ simplifica detectia
- Electrozi montati pe fata (in jurul ochilor) care detecteaza impulsurile nervoase ale muschilor globilor oculari
- Iluminare cu lumina IR intermitenta (imagini diferenta intre iluminare normala si iluminare IR) ⇒ simplifica detectia

Dezavantaje metode alternative

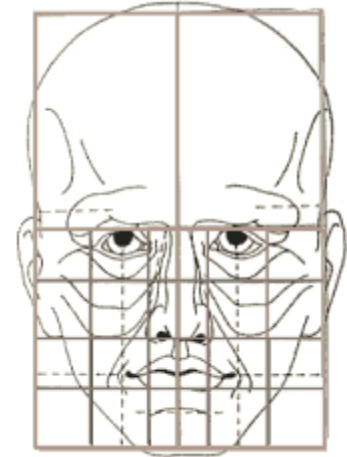
- Invazive
- Hardware specializat / costisor
- Procedura de calibrare a parametrilor elaborata



Analiza directiei privirii

1. Obținerea celor 2 imagini ale ochilor pe baza trasaturilor antropomorfe (regiunea ochilor trebuie să aibă înălțimea $1/8$ din înălțimea totală a feței și lățimea $1/5$ din lățimea totală a feței) \Rightarrow 2 sub-imagini care conțin ochii din imaginea cu rezoluția cea mai mare (nivel 0).

- Dimensiunea subimaginei depinde de scala la care s-a găsit fața \Rightarrow scalare la dimensiune fixă (60x80) pixeli prin interpolare bi-liniară.



2. Stabilizarea imaginilor ochilor (detectia și tracking-ul feței nu este suficient de precis pt. miscari ale capului de cativa pixeli)

- Diferența dintre cadre succesive pentru a crea imagini binare de mișcare (background subtraction)

- Calculul momentelor de ordin 1 (centru de masa). Aceste puncte centroid sunt folosite pentru a estima locația ochilor în imaginea feței (precizie bună pt. imagini de rezoluție mică)





Analiza directiei privirii

3. Comparatie intre ochiul stang si drept

Ochiul stâng și ochiul drept sunt comparați pentru a determina unde se uită utilizatorul:

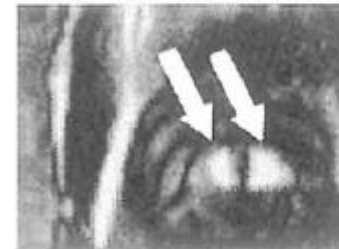
- Imaginea ochiului stâng este pusă în oglindă (b) și este scăzută din imaginea ochiului drept (a) \Rightarrow (c & d).
- Dacă utilizatorul se va uita direct la cameră diferența este mică (d). Dacă utilizatorul se uită în partea stângă, atunci ochiul pus în oglindă va părea ca se uită în partea dreaptă (b) iar diferența (c) este evidentă.
- Diferențele de intensitate (cu semn) dintre imaginea ochiului drept și cel stâng oglindit sunt proiectate (insumate) pe verticală \Rightarrow măsurarea direcției privirii



a)



b)



c)



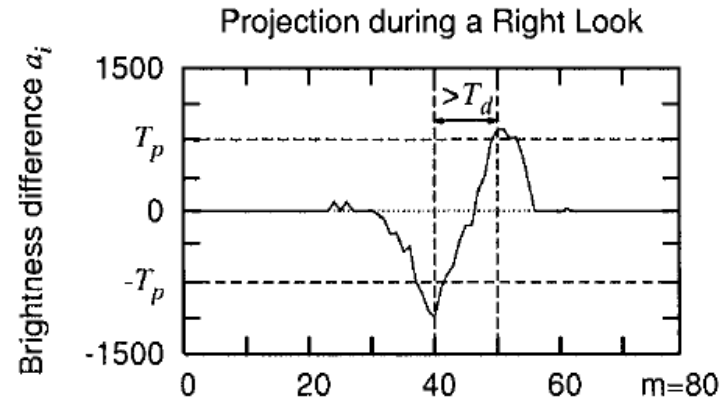
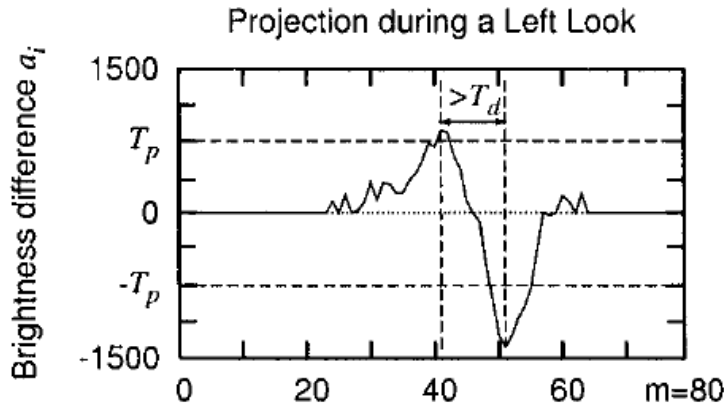
d)

- a) Ochiul drept (direcția de privire = stânga).
- b) Imagine în oglindă a ochiului stâng.
- c) Img. diferența (fara semn): la privire îndreptată în stânga.
- d) Img. Diferența (fara semn): la privire îndreptată înainte.



Analiza directiei privirii

4. Masurarea directiei privirii



$$a_i = \sum_{j=1}^n (I_r(i, j) - I_l(m - i, j))$$

$$a_{\min} = \min_{i=\{1, \dots, m\}} (a_i)$$

$$i_{\min} = \arg \min_{i=\{1, \dots, m\}} (a_i)$$

$$a_{\max} = \max_{i=\{1, \dots, m\}} (a_i)$$

$$i_{\max} = \arg \max_{i=\{1, \dots, m\}} (a_i)$$

Exista miscare a ochilor: $a_{\max} > T_p$

$$a_{\min} < -T_p$$

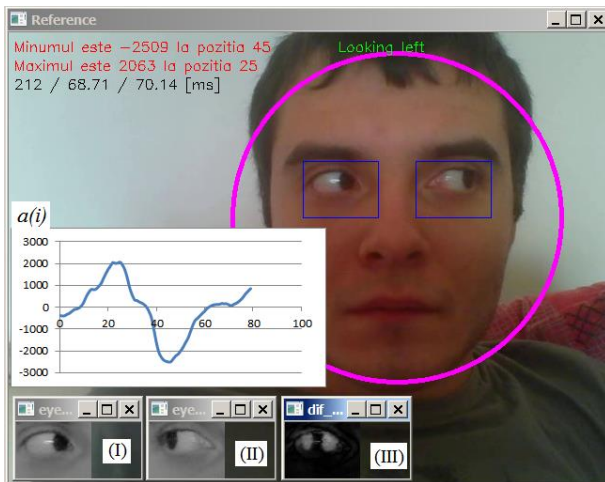
Directia miscarii: $i_{\max} - i_{\min} > T_d \Rightarrow$ 'right motion'

$i_{\max} - i_{\min} < -T_d \Rightarrow$ 'left motion'

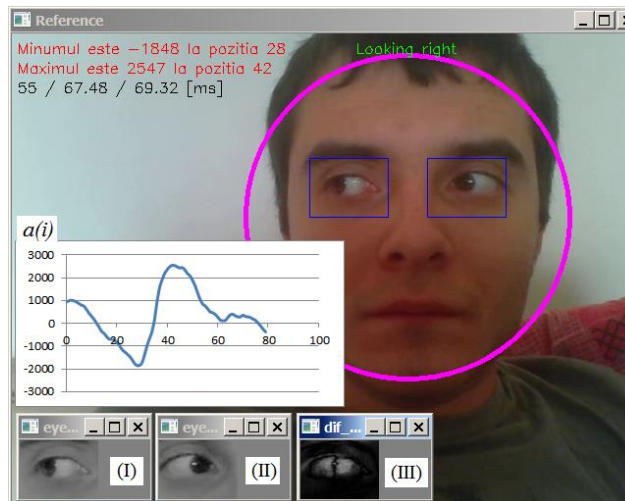


Analiza directiei privirii

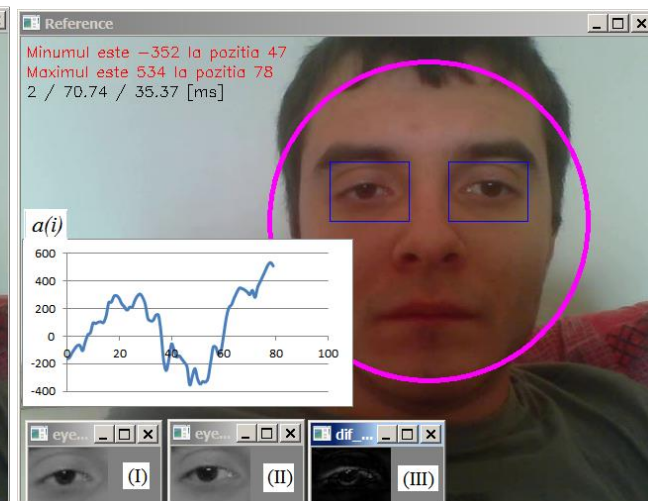
Exemple:



a.



b.



c.

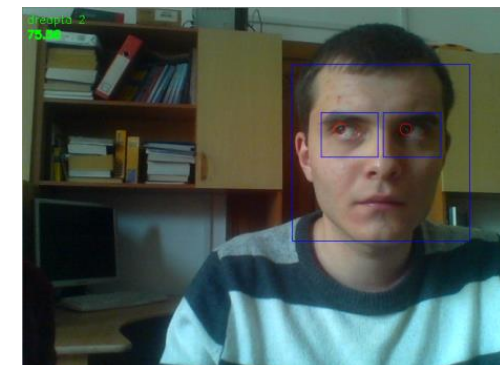
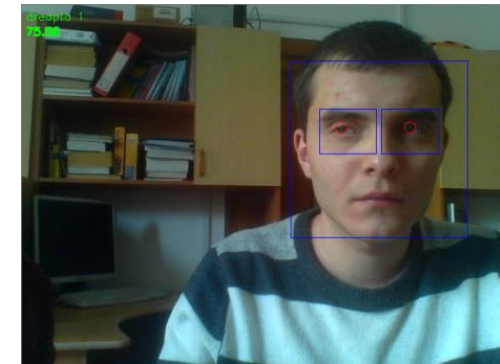
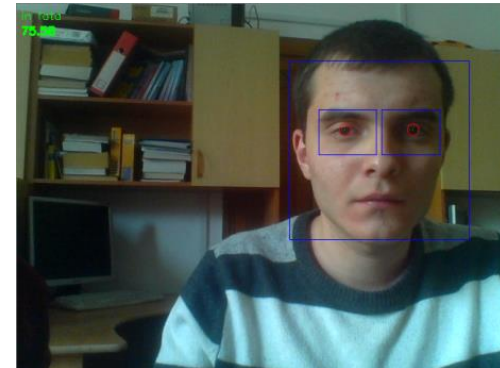
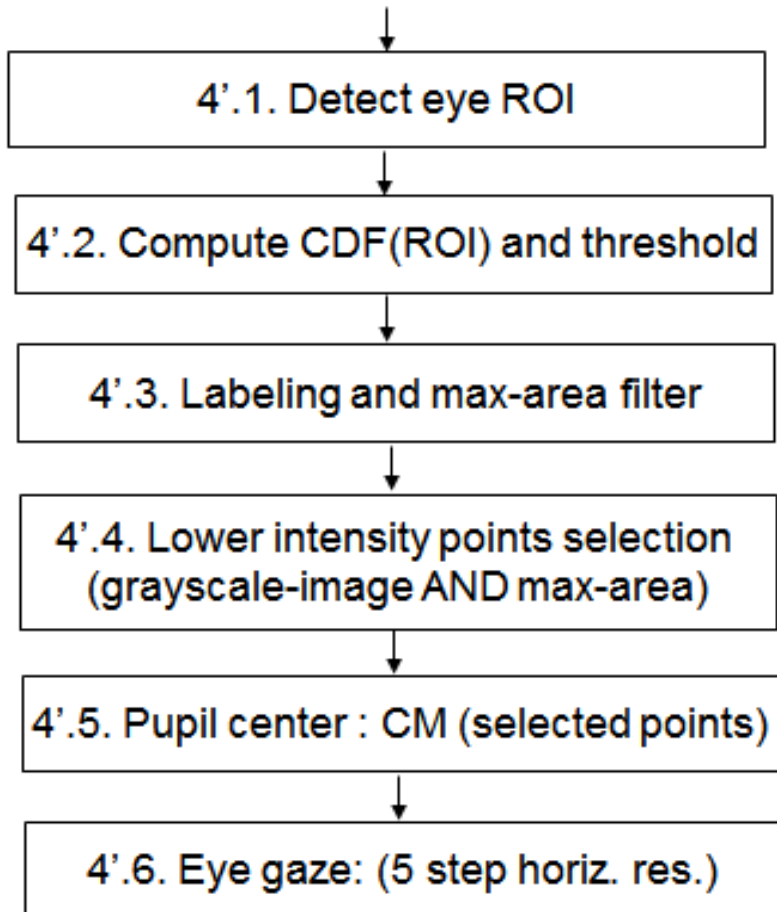
Ilustrarea rezultatelor detectiei privirii: (a) spre stanga, (b) spre dreapta (c) neutra/fata.



Estimarea directiei privirii

M. Ciesla Michal, P. Koziol, *Eye Pupil Location Using Webcam*, <http://arxiv.org/abs/1202.6517>

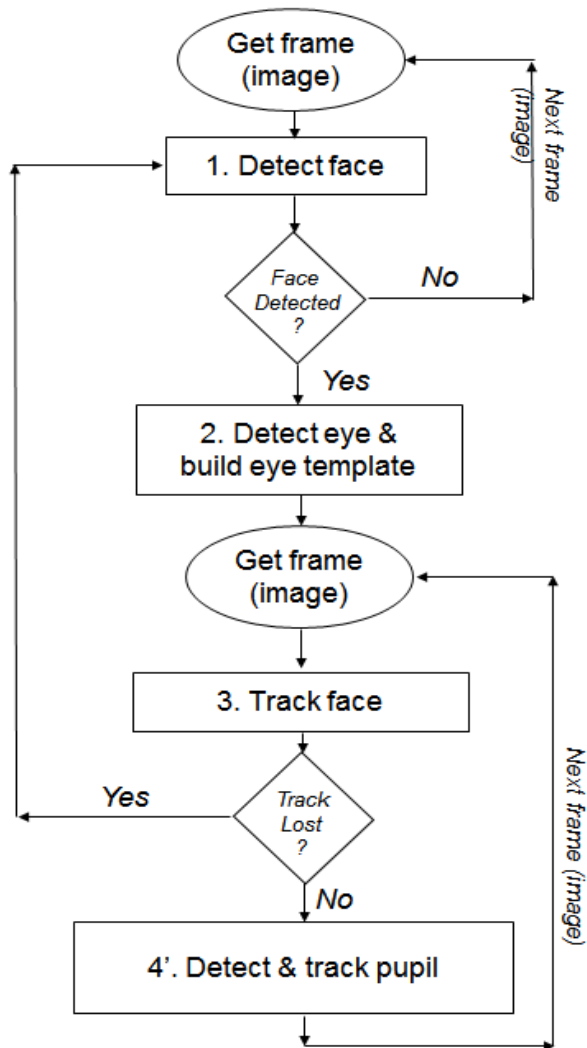
Algoritm de localizarea precisa a pupilei





Estimarea directiei privirii

Detectia si urmarirea fetei + detectia directiei ochilor



1. Detectie fete (Viola Jones / OpenCV)
2. Detectie pozitie ochi (Viola Jones) + validare trasaturi antropomorfe. Initializare sablon ochi.
3. Urmarire fata (tracking prin potrivire de sabloane (MatchTemplate / OpenCV)
4. Detectie si urmarire pupila

Alternativa la pasul 3: CAMSHIFT / OpenCV



Detectia de compnente faciale

Decti si urmarirea ochilor.
Detectia clipitului.



Referinte

M. Chauand, M. Betke, Real Time Eye Tracking and Blink Detection with USB Cameras, Boston University, Computer Science, Technical Report No. 2005-12.

http://nashruddin.com/Real_Time_Eye_Tracking_and_Blink_Detection

K.Grauman, M.Betke, J.Gips, G.Bradski, Communication via eye blinks detection and duration analysis in real time. Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR2001), Vol.2, pages1010–1017, Kauai, Hawaii, December 2001.

Alte resurse:

OpenCV Eye Tracking

<http://nashruddin.com/eyetracking-track-user-eye.html>

OpenCV Eye Detection

http://nashruddin.com/OpenCV_Eye_Detection



Aplicabilitate

Interfete pentru persoane cu dizabilitati

- detectia sablonului de clipire \Rightarrow interpretare actiuni

Sistem de detectie a gradului de atentie

- Aplicatii de asistenta a conducerii

Liveness detection

- Sisteme de securitate biometrice (bazate pe detectia si recunoasterea fetelor)

Performante

- Hardware minimal (WebCam)

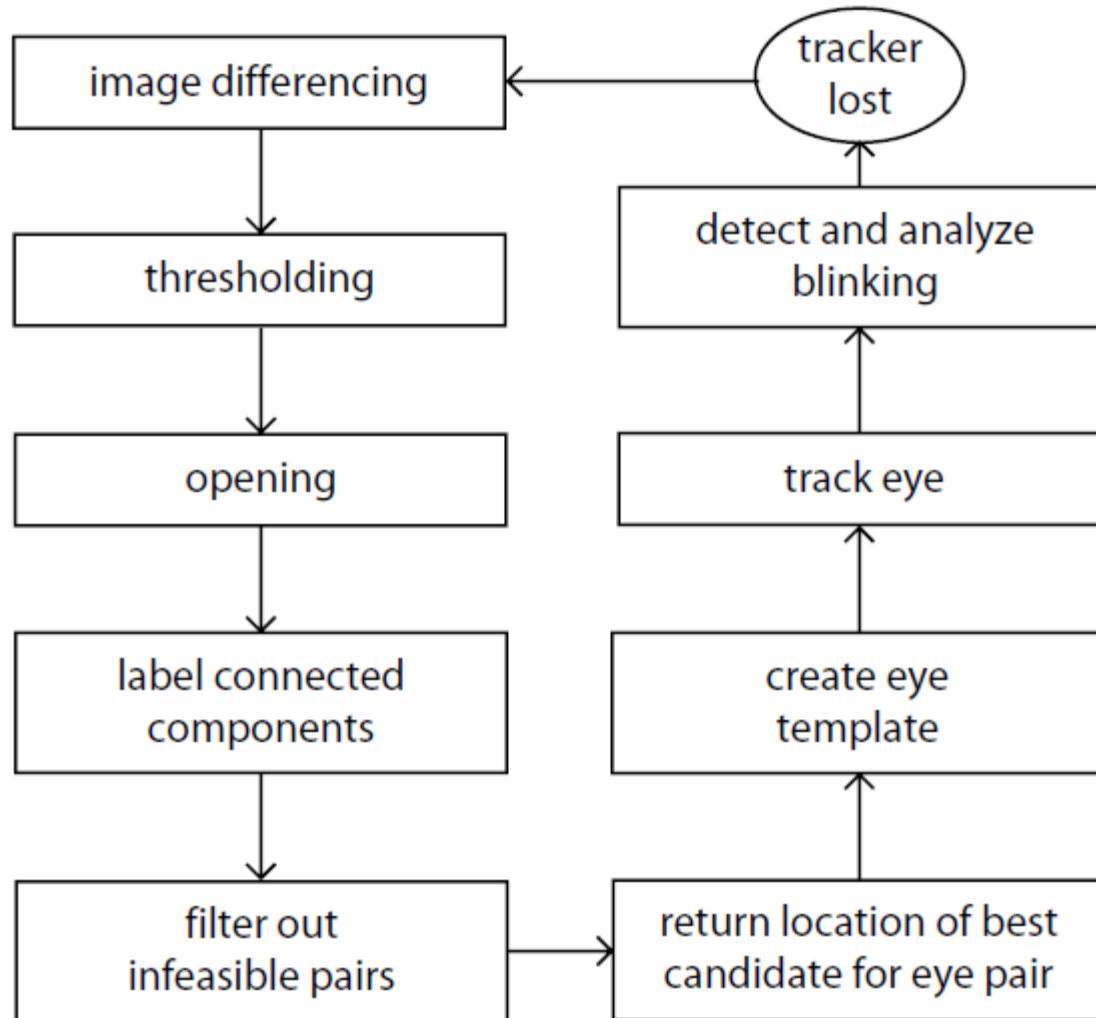
- Performante de timp real:

30 fps, 320x240, P4 2.8GHz

implementare cu functii OpenCVL



Scama bloc a aplicatiei





Initializare

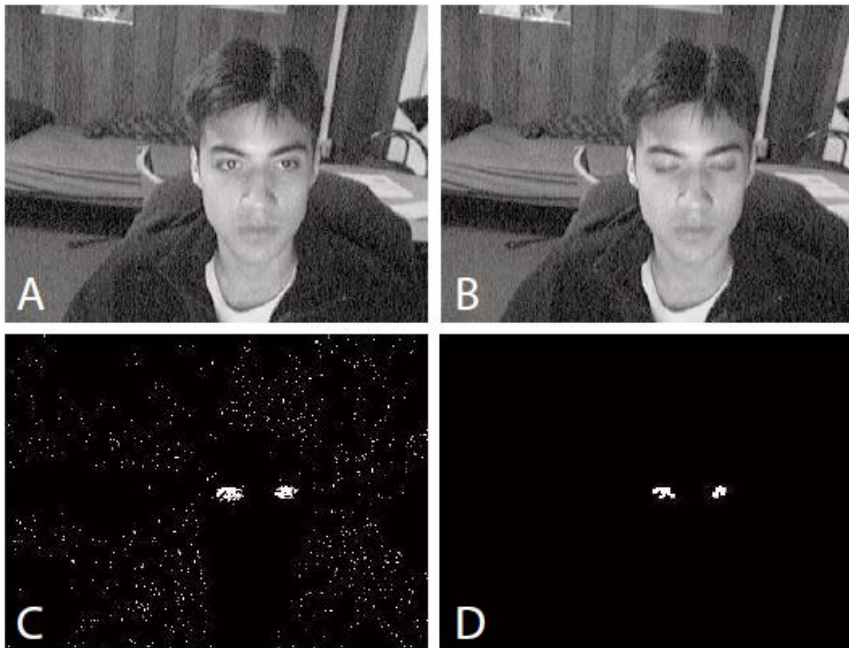
Detectia pozitiei ochilor (clipit involuntar)

1. Imaginea diferenta (B-A) \Rightarrow binarizare: C

```
cvSub(gray, prev, diff, NULL);  
cvThreshold(diff, diff, 5, 255, CV_THRESH_BINARY);
```

2. Eliminare zgomote \Rightarrow deschidere: D

```
IplConvKernel* kernel;  
kernel = cvCreateStructuringElementEx(3, 3, 1, 1, CV_SHAPE_CROSS, NULL);  
cvMorphologyEx(diff, diff, NULL, kernel, CV_MOP_OPEN, 1);
```





Initializare

- 3. Etichetare** \Rightarrow detectie componente conexe
 - dc. nr. componente conexe este prea mare (miscare pronuntata a capului) \Rightarrow reinitializare

- 4. Filtrare** suplimentara a componentelor conexe \Rightarrow perechi cu urmatoarele proprietati (ochii):
 - latime/inaltime asemenatoare
 - deplasament pe verticala mic
 - deplasamanet pe orizontala proportional cu dimensiunea componentelor conexe (trasaturi antropomorfe)

- 5. Determinare** coordonate dreptunghi circumscris la cea mai mare eticheta din pereche (ochiul cel mai bine vizibil) \Rightarrow ROI



Generare template

Template-ul pt. ochi se creeaza on-line in faza de initializare

Template-ul se copiaza din regiunea de interes (ROI) selectata la pasul precedent la momentul $t+\Delta t$ (t momentul in care se calculeaza segmentarea bazata pe clipire)

Intarzirea Δt este necesara pt. stabilizarea imaginii ochiului (ochi deschis). Valoarea ei trebuie sa fie mai mica decat intervalul de timp dintre 2 clipiri consecutive involuntare pt. a achizitiona imaginea ochiului deschis

Template-ul ochiului deschis se va folosi in tracking-ul ochilor.

```
cvWaitKey(250);  
cvSetImageROI(gray, rect_eye);  
cvCopy(gray, tpl, NULL);  
cvResetImageROI(gray);
```





Urmărirea ochilor

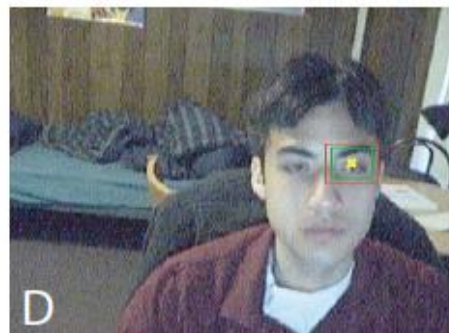
Rafinarea pozitiei ochilor in fiecare frame pe baza de template-matching

- nu se foloseste nici o metoda de filtrare (ex. Kalman) !

Metrica de corelatie: corelatia normalizata (robusta la variatii de iluminare)

$$\frac{\sum_{x,y} [f(x,y) - \bar{f}_{u,v}] [t(x-u, y-v) - \bar{t}]}{\sqrt{\sum_{x,y} [f(x,y) - \bar{f}_{u,v}]^2 \sum_{x,y} [t(x-u, y-v) - \bar{t}]^2}}$$

Potrivire $\rightarrow 1$

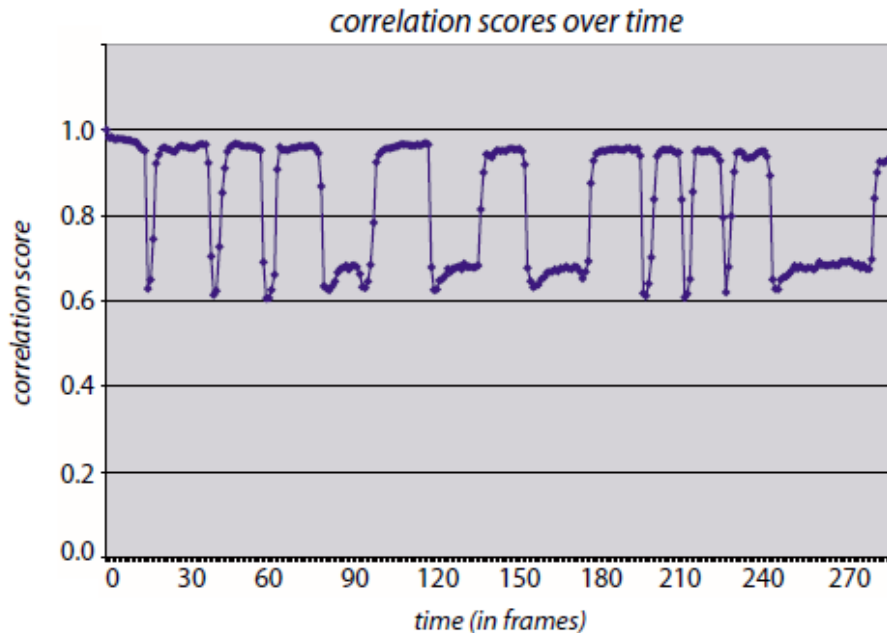


```
/* get the centroid of eye */
point = cvPoint(
    rect_eye.x + rect_eye.width / 2,
    rect_eye.y + rect_eye.height / 2
);
/* setup search window */
window = cvRect(
    point.x - WIN_WIDTH / 2,
    point.y - WIN_HEIGHT / 2,
    WIN_WIDTH,
    WIN_HEIGHT
);
/* locate the eye with template matching */
cvSetImageROI(gray, window);
cvMatchTemplate(gray, tpl, res,
    CV_TM_SQDIFF_NORMED);
cvMinMaxLoc(res, &minval, &maxval, &minloc,
    &maxloc, 0);
cvResetImageROI(gray);
```



Detectia clipirii

Calculeaza scorul de corelatie in fiecare frame



Masurarea frecventei de clipire:

-masurarea timpului T_D (ochi deschis si T_I (ochi inchis)

-masurarea se face prin declanșarea de timere cand functia de corelatie coboara/crește sub/peste pragurile stabilite

Pragurile \Rightarrow statistici pt. mai multi utilizatori (valabile doar daca capul ramane nemiscat)

- 0.85 ... 1: ochi deschis

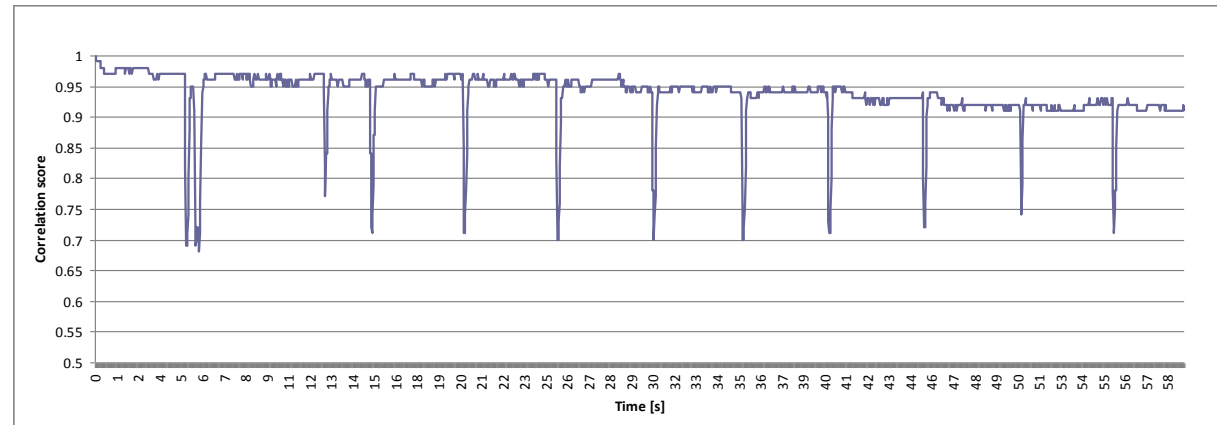
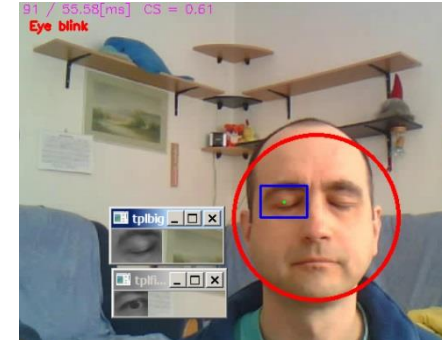
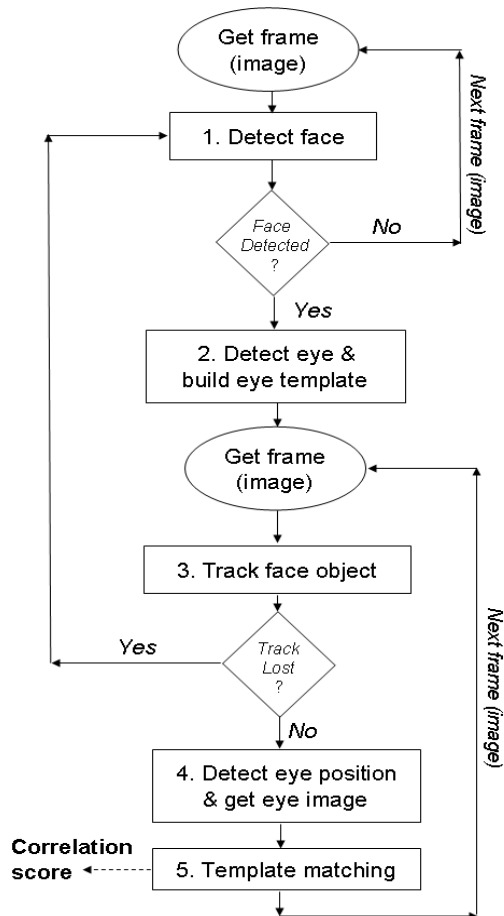
- 0.6 ... 0.7 ochi inchis

- sub 0.45: tracking pierdut \Rightarrow reinitializare tracking



Detectia clipirii

Varianta a metodei folosind detectia fetei (Viola-Jones) si a pozitiei ochilor pe baza trasaturilor antropomorfe:



Ilustrare a functiei de corelatie pentru un sablon de clipire normal (≈ 12 clipiri/min).



Postprocesare detectie fete

Validare detectie

Se poate face pe baza urmatoarelor trasaturi

- culoare
- trasaturi antropomorfe (pozitie relativa trasaturi faciale)
- factor de simetrie

Detectie orientare

- Rotatie in planul imaginii



Detectia axei de simetrie si orientarii

X. Chen, P.J. Flynn, K.W. Bowyer, "Fully Automated Facial Symmetry Axis Detection in Frontal Color Images", Department of Computer Science and Engineering University of Notre Dame, Notre Dame, IN 46556 USA,
http://nd.edu/~flynn/papers/ChenFlynnBowyer_AutoID2005.pdf

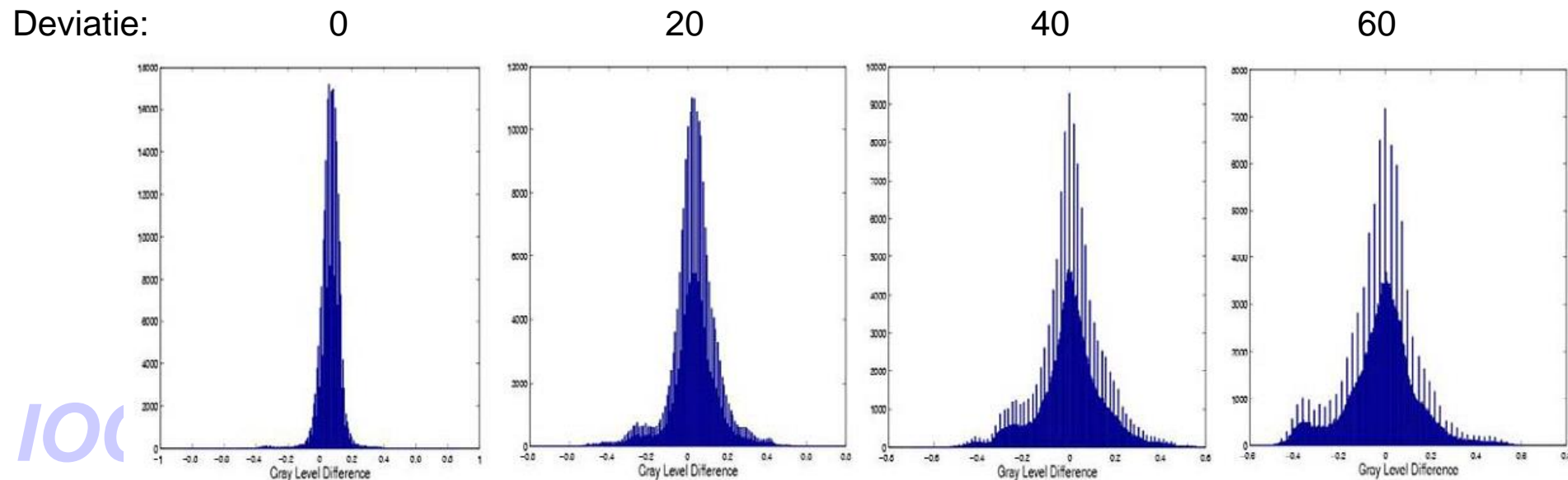
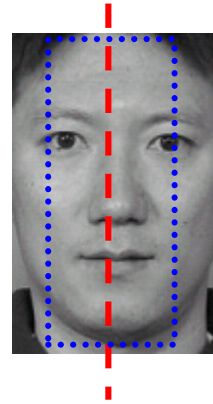
Metoda de detectie a fetelor – similara cu metoda 1





Detectia axei de simetrie

1. Se considera o axa de simetrie initiala (ex. axa de alungire a fetei segmentate sau axa verticala ce trece prin centrul de masa)
2. Se calculeaza gray level difference histogram (GLDH) in jurul axei considerate:
 - calculeaza diferentele de intensitate intre pixeli simetrici in jurul axei \Rightarrow histograma (GLDH)
3. Se ajusteaza discret pozitia axei de simetrie: in pozitia corecta media (varful histogramei) este maxima si deviata standard este minima

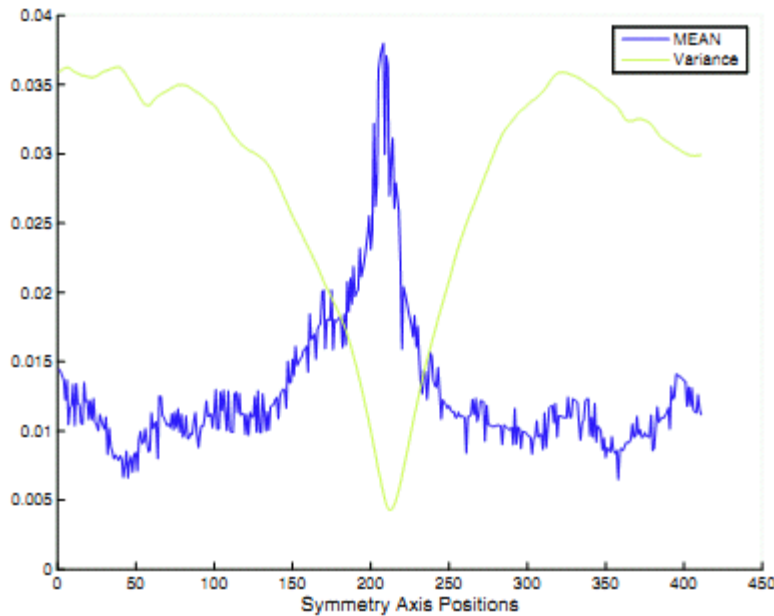




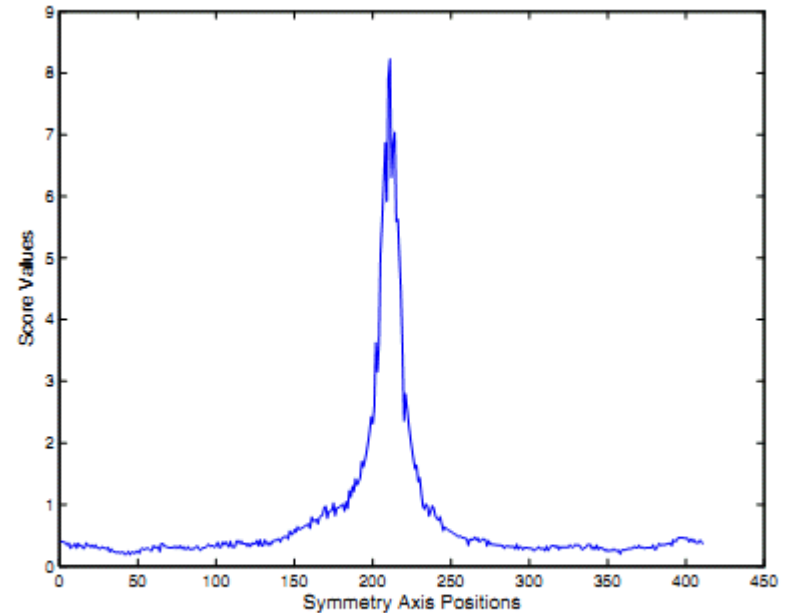
Detectia axei de simetrie

Pt. gasirea pozitiei optime (media (varful histogramei) este maxima si deviata standard este minima) se calculeaza Y -score si se maximizeaza in functie de pozitie:

$$Y = \frac{MEAN}{Variance}$$



(a) MEAN and variance

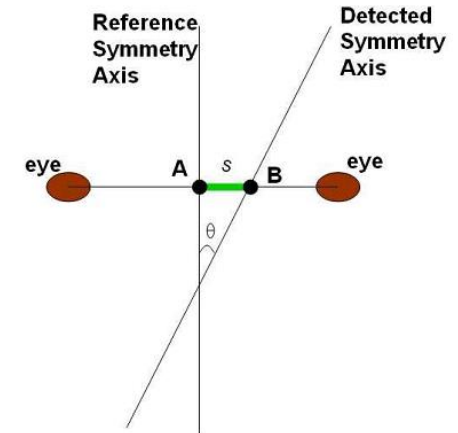
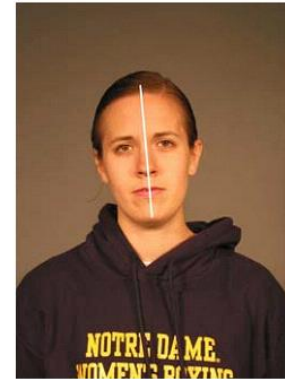
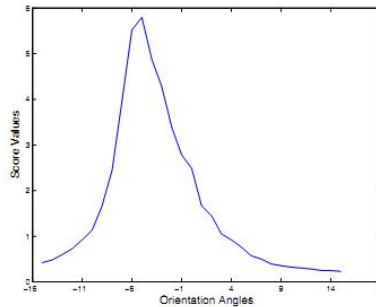
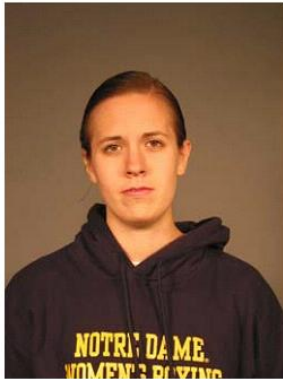


(b) Y score

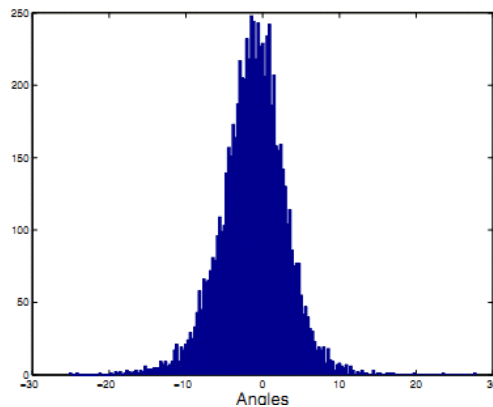


Detectia orientarii

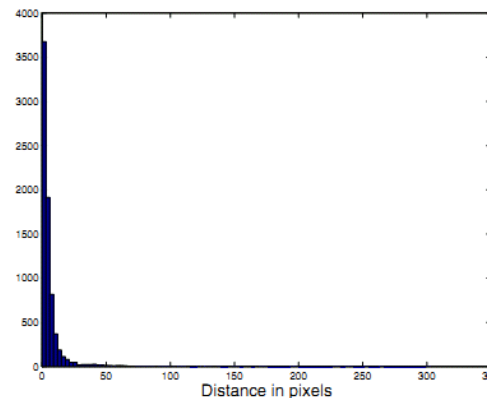
Se calculeaza *Y-score* pt. orientari in intervalul $-15 \dots 15$ grade cu pas de 1 grad.



Rezultate (7500 imagini) – comparatie cu GT selectat manual



Eroare unghi



Eroare pozitie