Detectia trasaturilor faciale





Introducere

- **Obiectiv**:
  - Estimarea pozitiei trasaturilor proeminente ale fetei
    - Ochi
    - Gura
    - Nas
    - Sprancene
    - Set complet de 64 puncte?



Poza luata de la http://www.luxand.com/facesdk/

Introducere

- Aplicatii posibile:
  - Detectia si urmarirea fetelor
  - Alinierea fetelor pentru recunoastere
  - Pas intermediar pentru
    - Recunoasterea expresiilor faciale si a emotiilor
    - Urmarirea privirii
      - Estimarea oboselii soferului
  - Estimarea pozitiei capului
  - Sinteza si animatie de fete virtuale
- Numeroase aplicatii => multitudine de metode cu diverse caracteristici si cerinte apriori in functie de rezultatul dorit

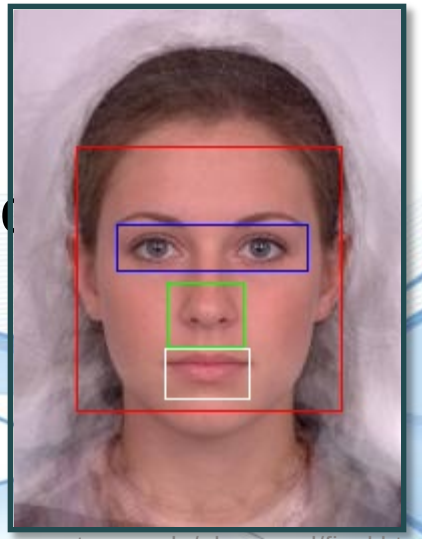
-> NU exista metoda universal valabila

Introducere

- Rezultate scontate:
  - Lista cu puncte caracteristice fiecar
    - centrul irisilor si colturile ochilor,
    - colturile si centrul sprancenelor,
    - varful nasului si/sau centrele narilor
    - centrul si colturile gurii
    - etc.
  - Zonele care delimiteaza trasaturile fa
    - Dreptunghi/elipsa de incadrare
    - Contururi (mai precis)



http://www.fantaface.com/mixer/



http://www.cs.utexas.edu/~bmccord/final.html

Criterii de performanta

- Pentru evaluarea obiectiva sunt necesare **marcaje de referinta**

- Detectia **corecta** - ***suficient de aproape*** de referinta
  - Diferite metrici
  - Diferite praguri

$$\text{Rata de detectie} = \frac{\text{Numar detectii corecte}}{\text{Numar de imagini}}$$

Criterii de performanta

[Jesorsky01]:

- Pentru detectia **centrelor irisilor**:

$$d_{eye} = \frac{\max(d_l, d_r)}{s} < T_{eye}$$

  - d_l si d_r = distantele dintre centrele detectate si cele marcate
  - s = distanta interoculara (intre cei doi irisi) marcata
  - T_{eye} = pragul de performata (ex: 0.1 sau 0.25)

- Pentru o mai buna caracterizare a performatei algoritmilor:
  - grafice detectii corecte vs T_{eye}

Criterii de performanta

[Cristinacce04]:

- Pentru *n* trasaturi:

$$m_e = \frac{1}{ns}\sum_{i=1}^{n} d_i < T_{eye}$$

  - d_i = distanta de la trasatura detectata la cea marcata
  - s = distanta interoculara (intre cei doi irisi) marcata
  - T_{eye} = pragul de performata

[Cootes98] :

- Eroarea medie absoluta intre punctele detectate si cele de referinta, normalizata la latimea fetei

Criterii de performanta

- Definite numai pentru cazul in care se cauta liste cu puncte caracteristice fiecarei trasaturi faciale

- Pentru cazul in care se cauta zona/regiunea care delimiteaza trasaturile faciale
  - Corect daca include trasatura faciala cautata
  - Corect daca se suprapune 90% peste marcajul manual
  - Index Jaccardi:

$$\frac{\text{Intersectie}}{\text{reuniune}} > 50\ \%$$

Baze de date publice

- Puse la dispozitia utilizatorilor pentru evaluarea performantelor algoritmilor
- Sunt in general gratuite pentru cercetatori din interiorul universitatilor (nu pentru industrie)
- Pot fi obtinute la cerere de catre personal de cercetare din interiorul universitatilor
- In general se cere citarea unui articol al autorilor bazei de date, in cazul amintirii acestei baze de date in publicatii de profil

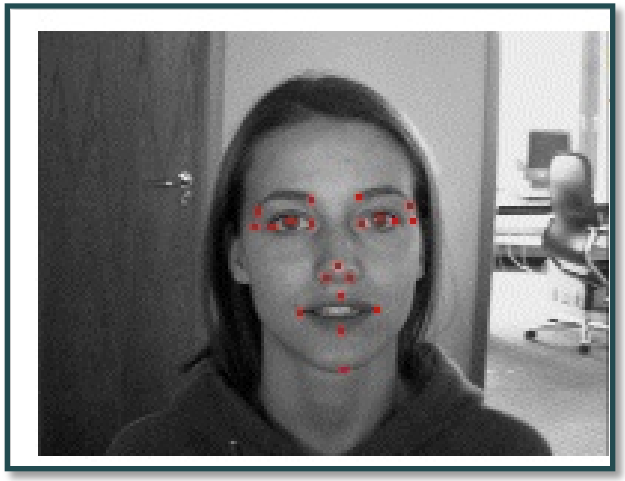
BioID

- Una dintre cele mai folosite baze de date din domeniu
- 1521 imagini
  - cu niveluri de gri
  - fete frontale de 384×286 pixeli
  - in conditii de iluminare diversa
  - cu fundal complex
- Contine si:
  - fete inclinate si rotite,
  - subiecti cu ochelari
  - cateva cazuri de subiecti cu ochi inchisi
  - cateva cazuri de subiecti care au expresii variate



BioID

- Contine marcaje pentru 20 de puncte caracteristice trasaturilor faciale:



  - Centrele irisilor – 2 puncte
  - Colturile ochilor – 4 puncte
  - Colturile sprancenelor – 4 puncte
  - Narile si varful nasului – 3 puncte
  - Colturile si punctele inferior si superior ale gurii – 4 puncte
  - Barbie – 1 punct
  - Tample – 2 puncte

FRGC

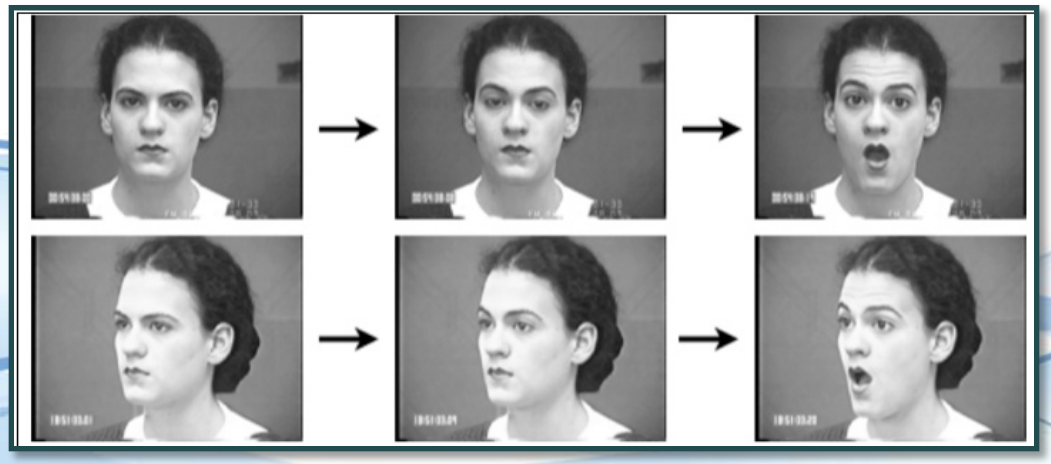
[Phillips05] **F**ace **R**ecognition **G**rand **C**hallenge

- Set de antrenare:
  - 12776 imagini color; 222 persoane
- Set de testare:
  - 4007*8=32056 imagini color; 466 persoane
- Rezolutie:1704×2272 sau 1200×1600
- Conditii de iluminare (ne)controlate
- Pozitie neutra sau zambind
- Marcaje:
  - Pozitiile centrelor ochilor, nasului si gurii
  - Identificarea subiectilor

Cohn-Kanade

[Kanade00]

- 1917 secvente video
- 182 persone cu diverse expresii
- Conditii de iluminare controlate
- 640×480 pixeli
- Marcaje de miscari faciale standard (ex: interiorul sprancenelor se ridica)



Alte baze de date

- 3-4 Baze de date de fata la fiecare conferinta importanta:
  - CVPR
  - ICCV/ECCV
  - Face and Gesture
- Baze de dimensiuni mari
    - Megaface
    - 1 Milion Tiny Faces
    - Affective faces in the Wild
  - Susceptibile pentru a fi folosite cu Deep Learning

Clasificarea metodelor

I. In functie de abordare aplicatiile folosesc:

- Geometria faciala
- Luminozitatea
- Contururi
- Culoare
- Forma

II. In functie de cunostintele apriori necesare:

- Cu fata detectata (mai robuste, fara probleme de scalare – sunt rezolvate de detectia de fete)
- Fara fata detectata (au nevoie de o estimare initiala a dimensiunii fetei => pentru aplicatii specifice in care se stie distanta de la camera la subiect)

Clasificarea metodelor

III. In functie de metoda:

- Potrivire de model/template
  - Cauta zone de imagine ale caror caracteristici sunt similare cu cele ale unui model a trasaturii faciale dorite a fi detectate. Modelele pot fi bazate pe contururi, culori, luminozitate, etc.
    - Cu rezultate mai bune
    - Mai lente
- Gasire de caracteristici
  - Se bazeaza pe proprietatile particulare ale trasaturii faciale cautate (contururi, culori, intensitati) in comparatie cu restul trasaturilor
    - Mai rapide
- Mixte (majoritatea)

METODE BAZATE PE POTRIVIRE DE MODEL

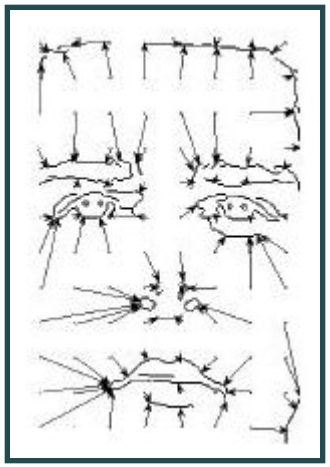




[Asteriadis06,09]

- **Campuri de vectori de distante**

  Distance Vector Fields (DVF)



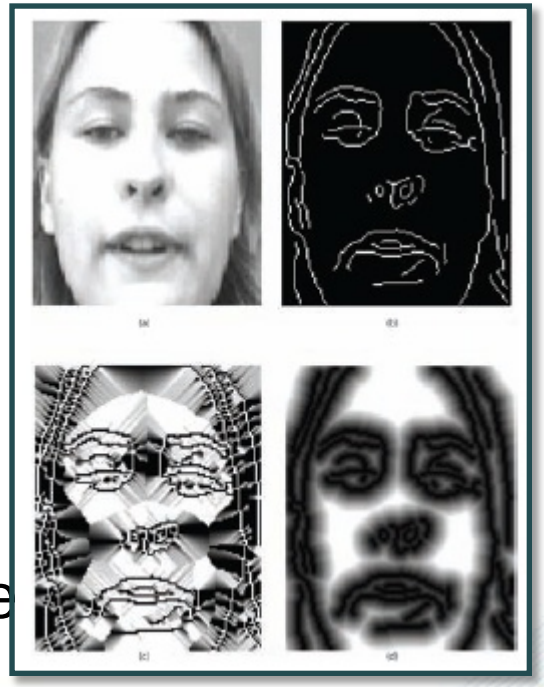
– Pentru fiecare pixel se calculeaza *vectorul* catre cel mai apropiat contur

- Descriere geometrica a trasaturilor
- Robust la variatii de iluminare si conditii slabe de iluminare

- Vector = amplitudine si unghi
  = proiecti pe orizontala si verticala

[Asteriadis06]

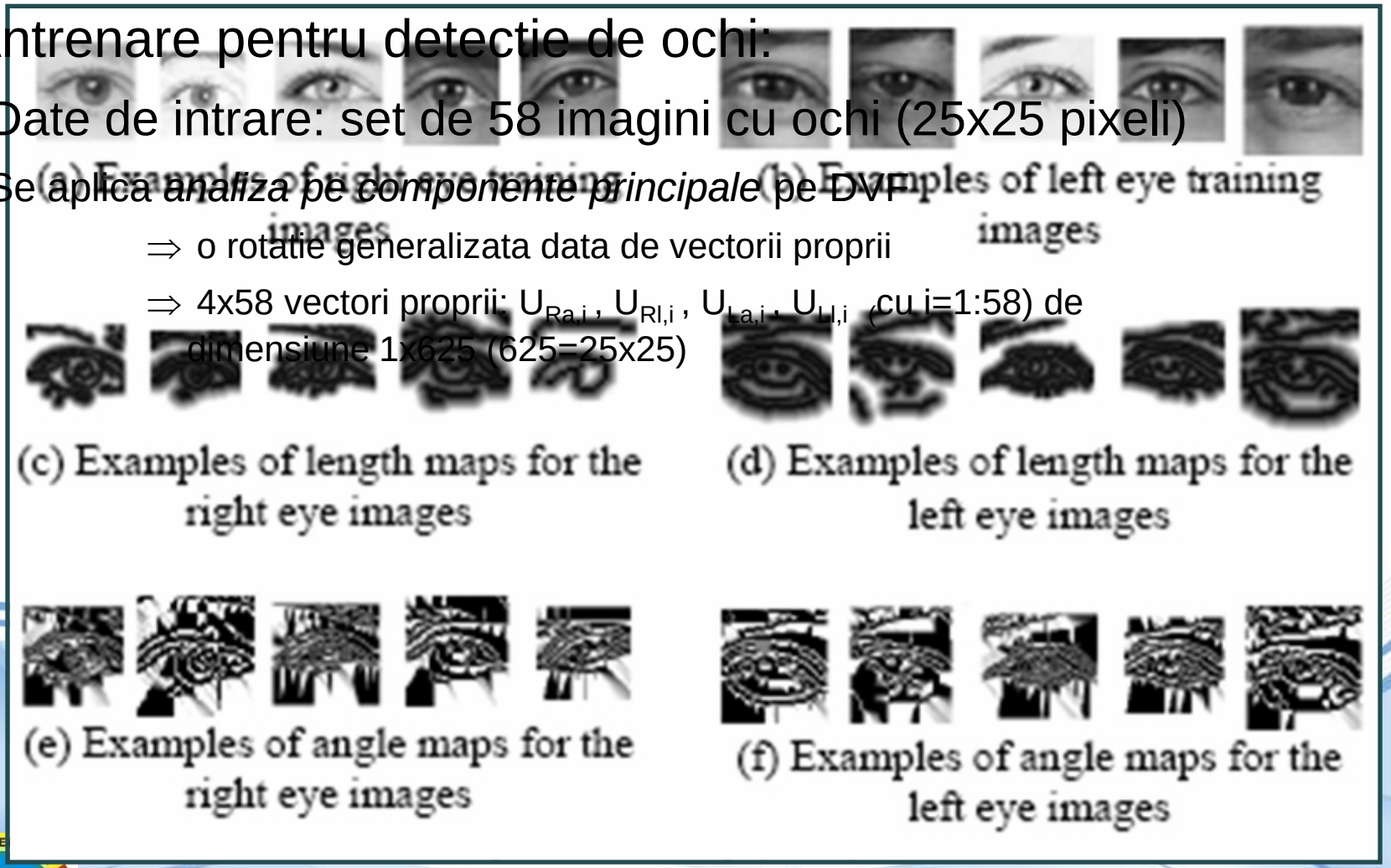
- Algoritm 1:
  - Detectie de fete
    - Scalare la 150x90 pixeli
  - Detectie de contururi
    - Canny (praguri 50 si 20 si varianta 1)
    - Se detecteaza contururile proeminente
  - Calcul de DVF = unghi + amplitudine





[Asteriadis06]

- Antrenare pentru detectie de ochi:
  - Date de intrare: set de 58 imagini cu ochi (25x25 pixeli)
  - Se aplica *analiza pe componente principale* pe DVF
    - $\Rightarrow$ o rotatie generalizata data de vectorii proprii
    - $\Rightarrow$ 4x58 vectori proprii: $U_{Ra,i}$, $U_{Rl,i}$, $U_{La,i}$, $U_{Ll,i}$ (cu i=1:58) de dimensiune 1x625 (625=25x25)



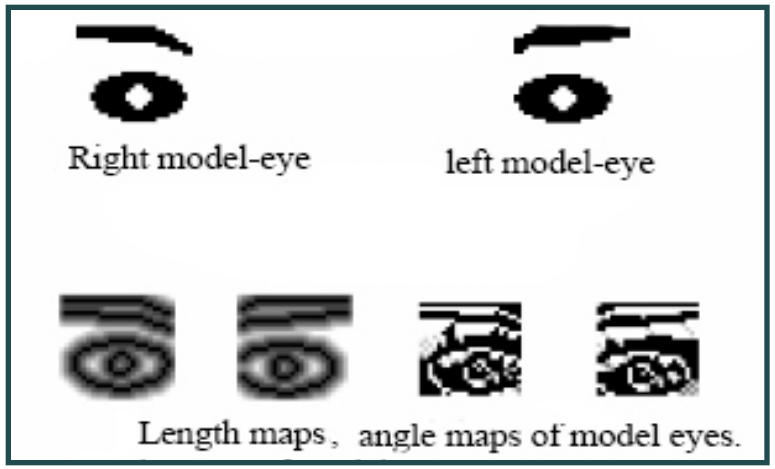
- Detectia zonei ochilor:
  - Se calculeaza DVF-ul fetei
  - Pentru toate zonele de 25x25 pixeli:
    - DVF-ul se vectorizeaza => Φ_{Ra} , Φ_{Rl} , Φ_{La} , Φ_{Ll}
    - Se proiecteaza DVF-ul pe vectorii proprii rezultati la antrenare

      =>

$$w_{Ra,i} = U_{Ra,i}^{T} \cdot \Phi_{Ra}$$
$$w_{Rl,i} = U_{Rl,i}^{T} \cdot \Phi_{Rl}$$
$$w_{La,i} = U_{La,i}^{T} \cdot \Phi_{La}$$
$$w_{Ll,i} = U_{Ll,i}^{T} \cdot \Phi_{Ll}$$

[Asteriadis06]

- Detectia zonei ochilor (continuare):
  - Se compara valorile din noul spatiu cu cele date de doua template-uri de ochi

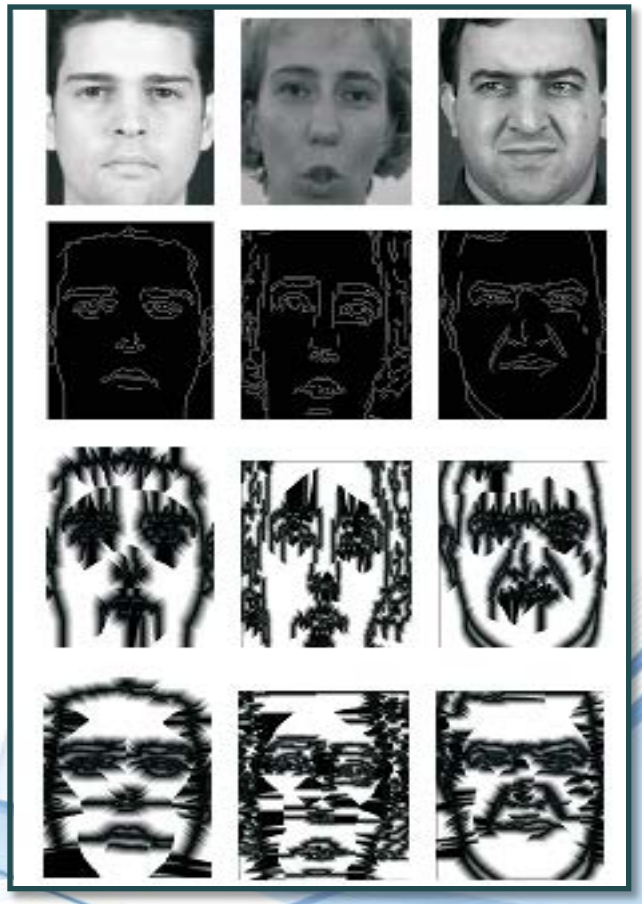


$$L_{2,R} = \| w_{Ra} - w_{Ra,Model} \| + \| w_{Rl} - w_{Rl,Model} \|$$
$$L_{2,L} = \| w_{La} - w_{La,Model} \| + \| w_{Ll} - w_{Ll,Model} \|$$

Se aleg zonele cu cea mai mica distanta

[Asteriadis09]

- Algoritm 2:
  - Detectie de fete
    - Scalare la 150x120 pixeli
  - Detectie de contururi
    - Canny (praguri 50 si 20 si varianta 1)
    - Se detecteaza contururile proeminente
  - Calcul de DVF = proiectii pe orizontala/verticala



[Asteriadis09]

- Antrenare pentru detectie de ochi:
  - Date de intrare: set de 58 imagini cu ochi (26x26 pixeli)



  - Se gaseste VDF-ul *mediu* pentru fiecare ochi si se foloseste ca model/template



- Detectia zonei ochilor:
  - Se calculeaza DVF-ul fetei
  - Pentru toate zonele de 26x26 pixeli:
    - Se compara DVF-ul zonei cu DVF-ul template-ului:

$$E_{L_2} = \sum_{i \in R_k} \| v_i - m_i \|_2$$

$\|.\|_2$ – norma L_2

v_i – DVF-ul regiunii (vectorizat)

m_i – DVF-ul template-ului (vectorizat)

- Zona care minimizeaza E_{L_2} e selectata ca zona de ochi

[Asteriadis06,09]

- Detectia centrelor irisilor:
  - Se anuleaza reflexia din zona detectata a ochilor
    - Se binarizeaza zona cu algoritmul Otsu
    - Toate zonele albe care ocupa mai putin de 1% sunt considerate reflexii si se inlocuiesc in imaginea initiala cu media pixelilor din jur



[Asteriadis06,09]

- Detectia centrelor irisilor (continuare):
  - Se considera partea de jos a zonei detectate a ochilor 15x25pixeli (se taie zona sprancenelor)
  - Obs: in aceasta zona sunt contururi puternice

  => se fac derivatele pe verticala si se proiecteaza pe orizontala

$$D_x(x,y) = I(x,y) - I(x-1,y)$$

  - Se determina cele 4 linii cu cele mai mari proiectii
  - Se alege linia *mediana* ca fiind cea a centrului irisului
  - Analog pentru coloana centrului irisului

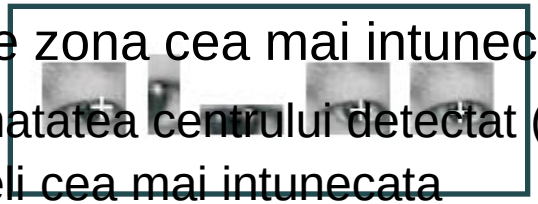
$$D_x^p(y) = \sum_x D_x(x,y)$$

[Asteriadis06,09]

- Detectia centrelor irisilor (rafinarea rezultatului)

  Pas 1. Se cauta cea mai intunecata coloana in jurul celei deja detectate (se foloseste o zona de 10x4pixeli). Analog cea mai intunecata linie.

  Pas 2. Irisul e zona cea mai intunecata

  - In vecinatatea centrului detectat (9x9pixeli) se cauta zona de 5x5pixeli cea mai intunecata
  - Centrul irisului este centrul acestei zone

[Asteriadis06]

Table I: RESULTS ON THE XM2VTS DATABASE

	Eye regions	Eye centers
People without glasses	99.7%	99.7%
People with glasses	96.2%	95.0%
Total	98.5%	98%

Table II: RESULTS ON THE BioID DATABASE

	Eye regions	Eye centers
People without glasses	99.5%	99.1%
People with glasses	98.3%	98.2%
People with closed eyes	100%	100%
Total	99.1%	98.8%



[Asteriadis09]

Eye region detection and eye centers localization results on the XM2VTS database.

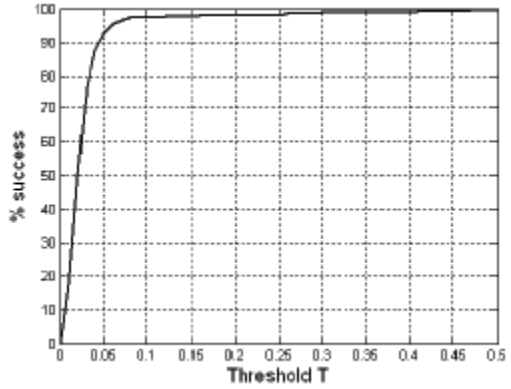
	Eye Regions	Eye Centers, T=0.25	Eye Centers, T=0.1
Proposed method, people without glasses	99.06%	99.33%	98.54%
Proposed method, people with glasses	98.7%	97.92%	97.44%
Proposed method, total	98.93%	98.85%	98.14%

Eye region detection and eye centers localization results on the BioID database.

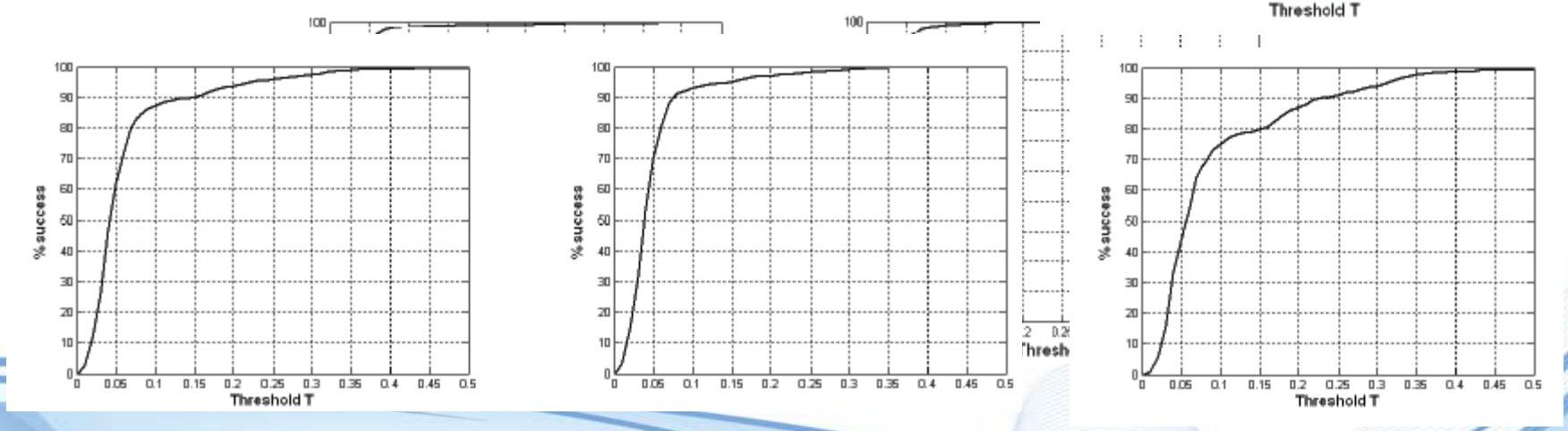
	Eye Regions	Eye Centers, T=0.25	Eye Centers, T=0.1
Proposed method, people without glasses	99.50%	98.74%	94.30%
Proposed method, people with glasses	98.3%	90.95%	75.93%
Proposed method, total	99.10%	96.00%	89.42%

[Asteriadis09]

XM2VTS



BioID



Detectii corecte pentru localizarea centrelor ochilor vs T_{eye} [Jesorsky01]
pentru toata baza de date, numai pentru subiectii fara ochelari
si respectiv numai pentru subiectii cu ochelari

Modele Active de Aparență

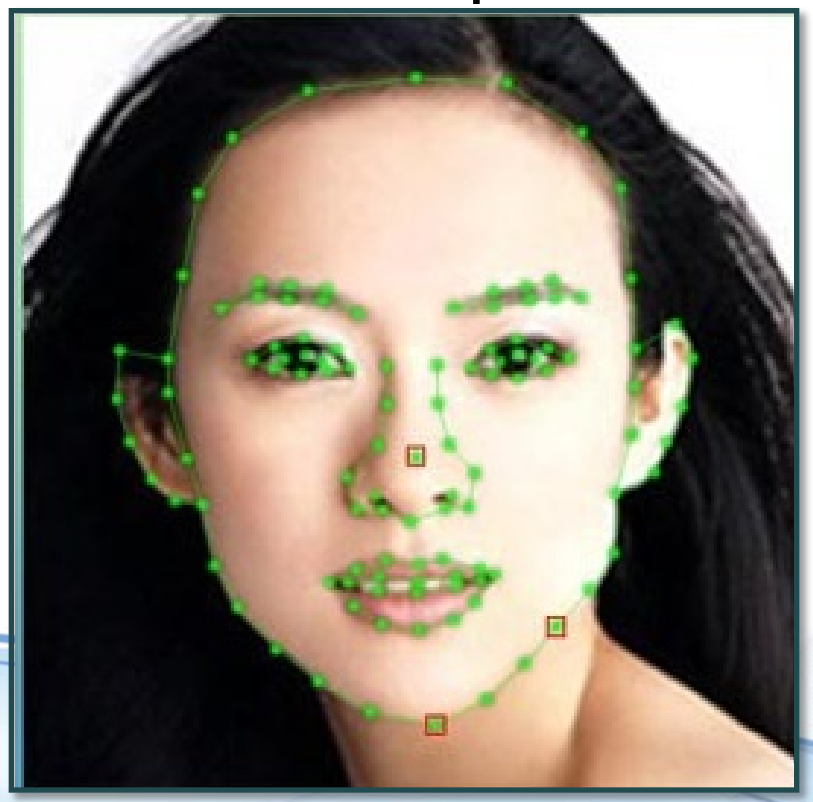
- AAM- Active Apearance models=Modele active de aparență
  - [Cootes98] T.F. Cootes, G. J. Edwards, and C. J. Taylor. *Active appearance models*. ECCV, 2:484–498, 1998
  - Extensie a ASM – Active Shape Models =Modele active de formă
    - [Cootes95] Cootes, Timothy F., et al. "Active shape models-their training and application." *Computer vision and image understanding* 61.1 (1995): 38-59.
- Scop: Formulează modelul astfel incât să “interpreteze” imaginile cu fete
    - Set de parametri ce caracterizeaza persoana, expresie, postura, etc.
    - Set compact
    - Model robust

Ideea

- Obiectele sunt modelate in raport cu forma (**shape - ASM**) si nivel de gri
  - ASM+ intensitate = AAM
  - Necesita antrenare
- Modele noi sunt sintetizate si apoi suprapuse peste imaginea noua
  - Parametri liberi care permit adaptarea
- Parametri modelului sunt modificati (optimizati) astfel incat sa se obtina **maximul potrivirii**

Active Shape Model (ASM)

- Model de distributie a punctelor



[Cootes95] Cootes, Timothy F., et al. "Active shape models-their training and application." *Computer vision and image understanding* 61.1 (1995): 38-59

ASM-model

- Forma/modelul - multimea de pct $\mathbf{X_i}$=[linie,coloana]
- Generam modele noi $x=[X_1, X_2, X_3...X_N]$

$$x = \mu + P*b$$ **Offline - antrenare**

unde

- μ – modelul mediu
- (b) combinatie liniara a componentelor principale (via PCA) P

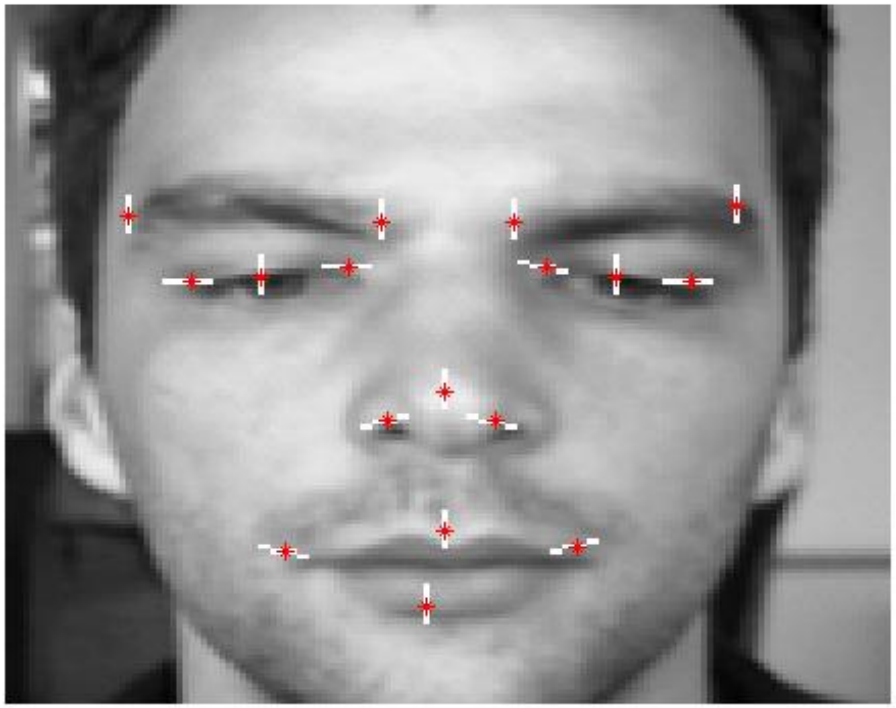
- Se potriveste I_x peste regiune din imaginea de test I_i, alterand b astfel incat $(I_x - I_i) = \Delta I$ sa fie minim **online**

ASM

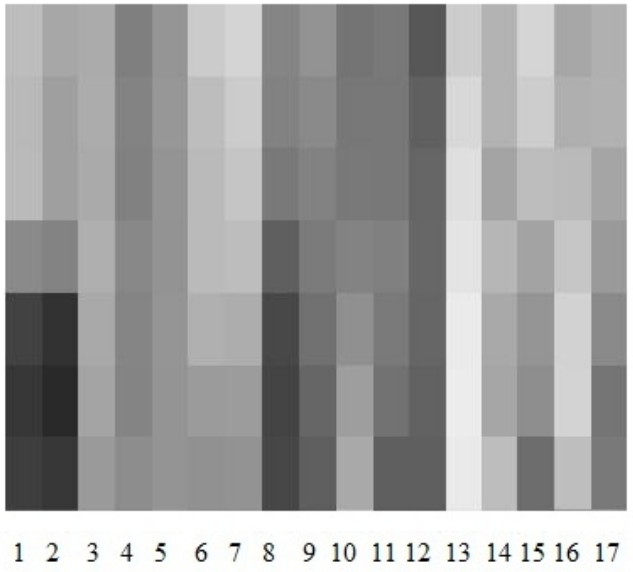
- Offline - antrenare: Aplica PCA pe imaginile de antrenament pentru a afla axele de variatie

- Online- antrenare – imagine noua
  - Se proiecteza forma medie $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Pb}$
  - Se modifica iterativ punctele pentru a se potivi pe vecinatatea locala

Generarea modelului de profil



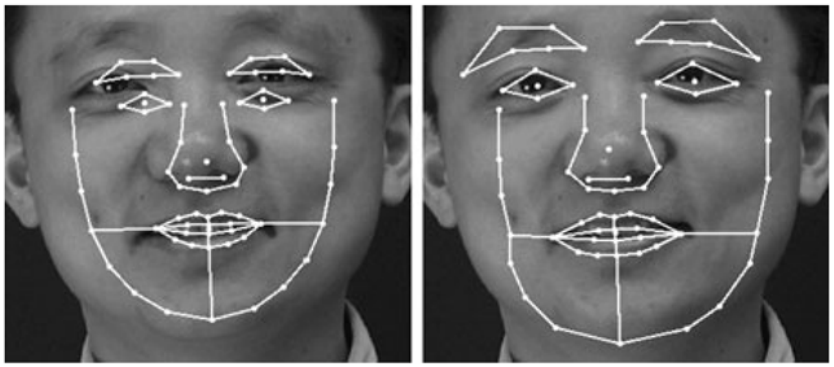
Mustăţile corespunzătoare punctelor de reper suprapuse peste imaginea subiectului



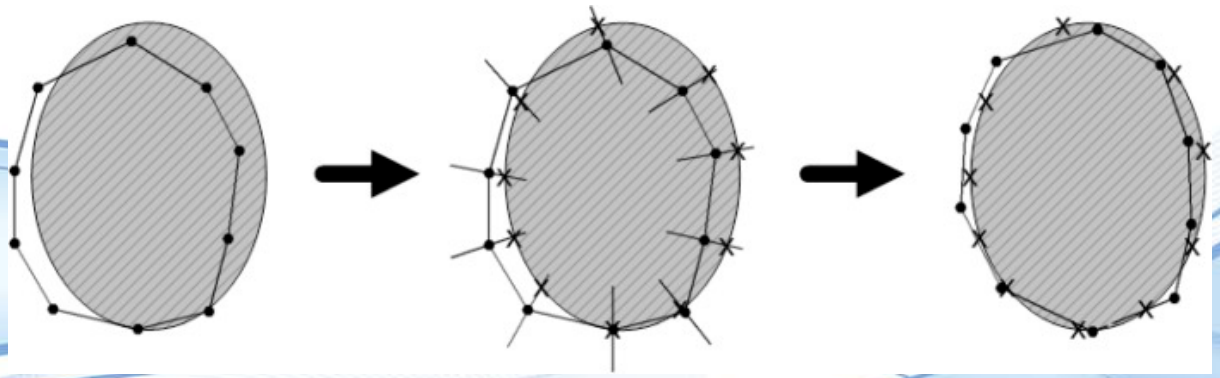
Valorile pixelilor de-a lungul mustăţilor

ASM Concluzii

- ASM e rapid
- ASM este prea simplu pentru a functiona cand noi imagini sunt introduse
- Punctul de convergenta poate sa nu fie o solutie buna (vizual)

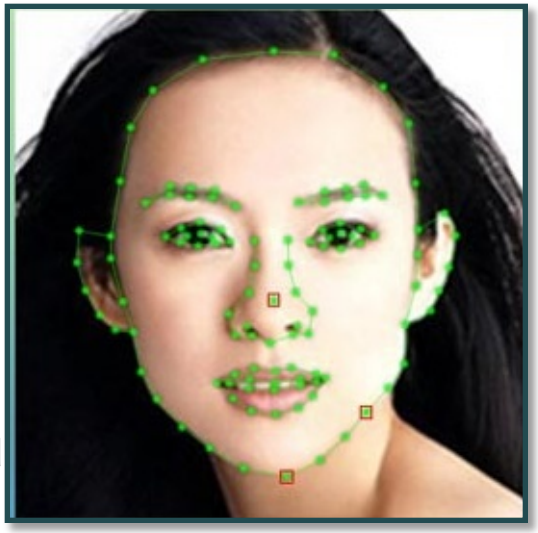


Initial Converged



Algoritmul AAM : pas cu pas

1. Baza de date cu set de landmark-uri adnotate
2. Se aliniaza formele din fiecare imagine
3. Aplica PCA
   - Se afla axele principale de variatie a formelor
4. Se normalizeaza datele astfel incat sa nu fie dependente de forma
   - Transformare liniara incat media si variata standard
5. Se normalizeaza nivelurile de gri
6. Se aplica PCA – directiile important de variatie
7. PCA pe modelul combinat



AAM=ASM+gri

- Se combina forma si variatia nivelului de gri intr-o masura statistica mai puternica
- Obiective:
  - Modelul va avea o putere de reprezentare crescuta
  - Modelul mosteneste beneficiile formei (ASM)
  - Modelul imbunatateste performanta

Detalii AAM

- Formeaza vectorul de forma **x**:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{b}_s$$

- Distorsioneaza (warp) imaginea pentru a o potrivi peste forma medie
- Esantioneaza (culege) informatie de intensitate **g**
- Aplica PCA peste **g**

Detalii AAM

- Se obtine:

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{b}_g$$

- Se concateneaza forma si parametri nivelului de gri dupa PCA
- Se aplica un nou PCA peste vectorul concatenat

AAM Detalii

- Se obtine:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Q}_s \mathbf{c}$$

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{Q}_g \mathbf{c}$$

AAM – exemplu nou

- Imaginea noua este interpretata via o problema de optimizare
- Se minimizeaza differenta dintre imaginea reala si cea sintetizata cu PCA
  - In optimizare se cauta acei paramteri ai PCA care produc cea mai buna potrivire

$$\delta \mathbf{I} = \mathbf{I_i} - \mathbf{I_m}$$

Optimizarea este dificila avand (~80 parametri).
Exista solutii post AAM

Rezultate experimentare (cont.)

- Antrenare:
  - 400 imgs, 112 puncte cheie
  - 80 AAM params
  - Variatia params contine 98% din variatia observata
- Testare:
  - 80 fete nefolosite la antrenare



Rezultate experimentare (cont.)

- Rezultate dupa , 2, 5, and 12 iterations



Rezultate experimentare (cont.)

- Reconstructii:



Experimental results (cont.)

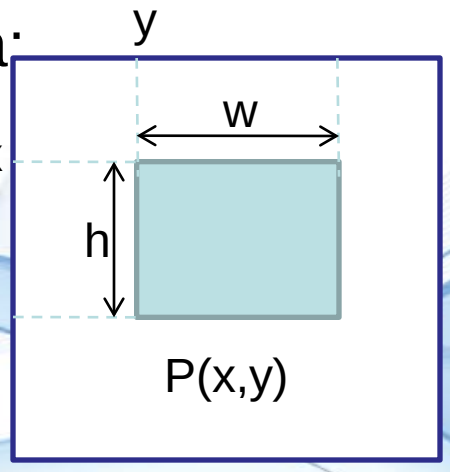


[Wu03]

- Se cauta fete pe baza pozitiei ochilor
- Detecteaza ochii pe baza luminozitatii **-> ochi sunt mai intunecati decat zona inconjuratoare**
- Cauta zone de imagine de dimensiunea ochilor care sunt mai intunecate decat zonele inconjuratoare
- Lucreaza pe imaginea de intensitate P(x,y)
- Foloseste media pixelilor dintr-o zona:

$$avg(P,x,y,h,w) = \frac{\sum_{i=x}^{x+h-1}\sum_{j=y}^{y+w-1} P(i,j)}{h*w}$$

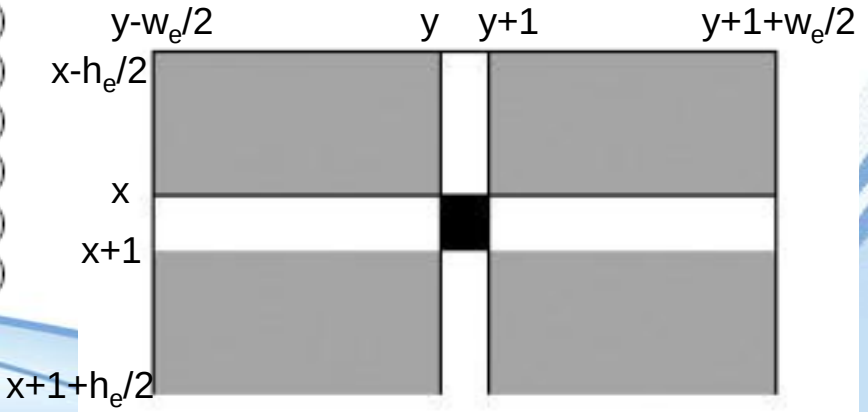
$$P(x,y) \in [0,1]$$



[Wu03]

- Se cauta ochi de dimensiunea $h_e \times w_e$
- Un pixel e marcat ca fiind posibil pixel de ochi daca indeplineste cel putin 6 dintre conditiile:

$$P(x,y) < 0.9 * \text{avg}(P, x - \lfloor h_e/2 \rfloor, y - \lfloor w_e/2 \rfloor, \lfloor h_e/2 \rfloor, \lfloor w_e/2 \rfloor)$$
$$P(x,y) < 0.9 * \text{avg}(P, x - \lfloor h_e/2 \rfloor, y, \lfloor h_e/2 \rfloor, 1)$$
$$P(x,y) < 0.9 * \text{avg}(P, x - \lfloor h_e/2 \rfloor, y + 1, \lfloor h_e/2 \rfloor, \lfloor w_e/2 \rfloor)$$
$$P(x,y) < 0.9 * \text{avg}(P, x, y - \lfloor w_e/2 \rfloor, 1, \lfloor w_e/2 \rfloor)$$
$$P(x,y) < 0.9 * \text{avg}(P, x, y + 1, 1, \lfloor w_e/2 \rfloor)$$
$$P(x,y) < 0.9 * \text{avg}(P, x + 1, y - \lfloor w_e/2 \rfloor, \lfloor h_e/2 \rfloor, \lfloor w_e/2 \rfloor)$$
$$P(x,y) < 0.9 * \text{avg}(P, x + 1, y, \lfloor h_e/2 \rfloor, 1)$$
$$P(x,y) < 0.9 * \text{avg}(P, x + 1, y + 1, \lfloor h_e/2 \rfloor, \lfloor w_e/2 \rfloor)$$



=> o imagine binara P'(x,y)

Proiectii integrale

- IPF - Integral Projection Function
- Imaginea 2D -> 2 vectori 1D
- Vectorii = suma pe linii/coloane:

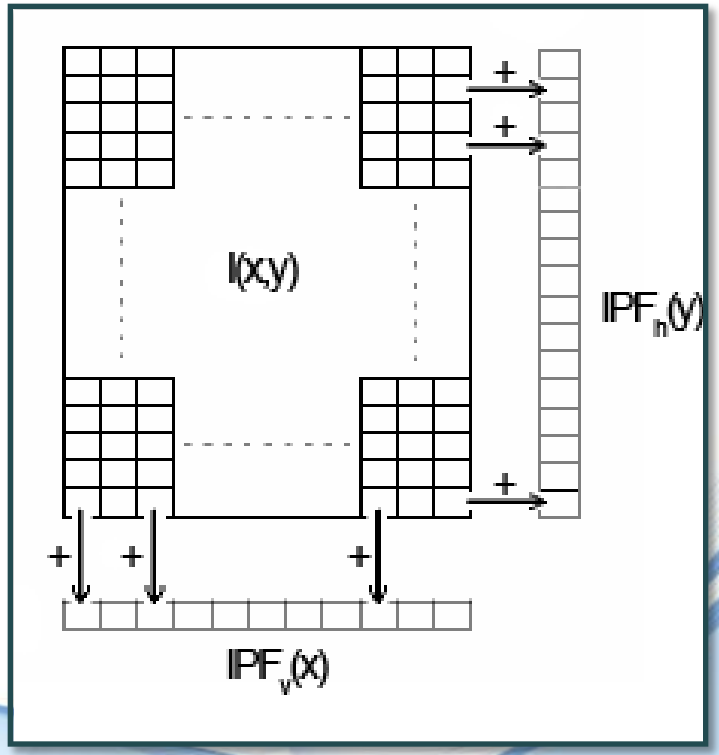
$$IPF_h(y) = \sum_{x_i = x_1}^{x_2} I(x_i, y)$$

$$IPF_v(x) = \sum_{y_j = y_1}^{y_2} I(x, y_j)$$

- Vectorii normati (medii)

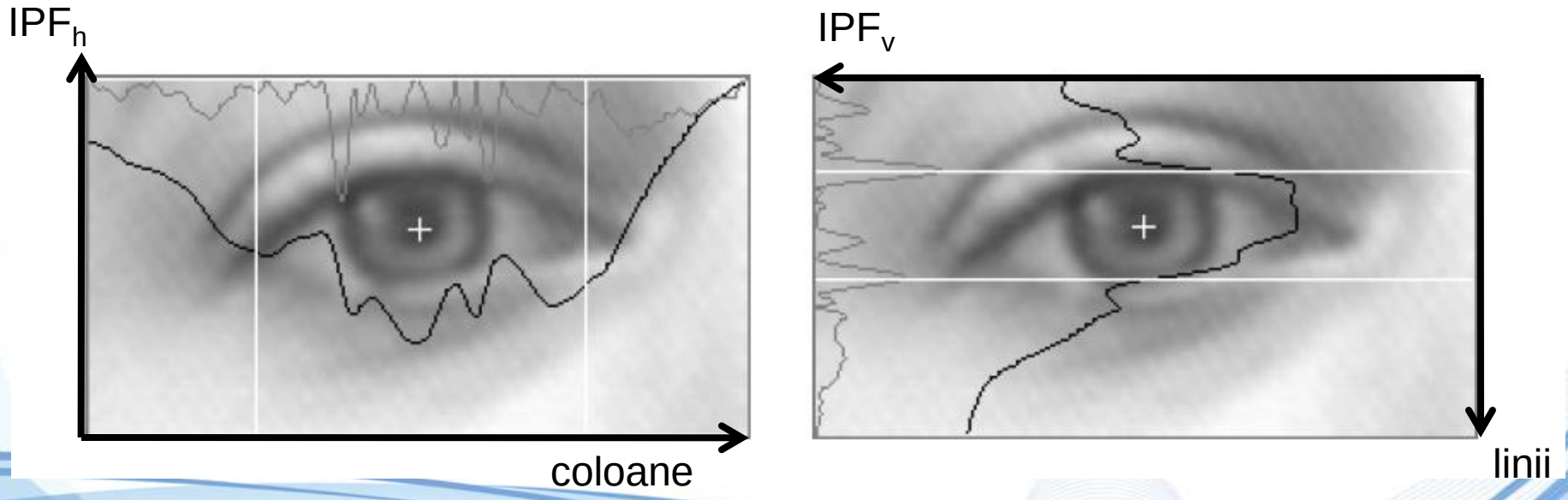
$$IPF_h'(y) = \frac{1}{x_2 - x_1} \sum_{x_i = x_1}^{x_2} I(x_i, y)$$

$$IPF_v'(x) = \frac{1}{y_2 - y_1} \sum_{y_j = y_1}^{y_2} I(x, y_j)$$



Proiectii integrale - exemplu

[Zhou04]



Proiectii de varianta

- VPF - Variance Projection Function

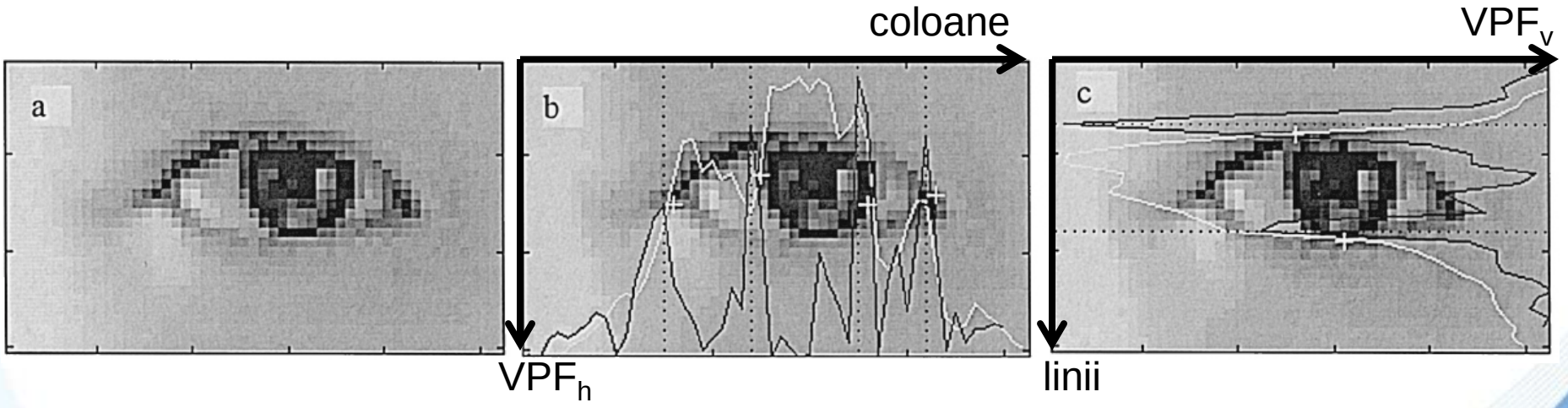
- Imaginea 2D -> 2 vectori 1D

$$VPF_h(y) = \frac{1}{x_2 - x_1} \sum_{x_i = x_1}^{x_2} [I(x_i, y) - IPF_h'(y)]^2$$

$$VPF_v(x) = \frac{1}{y_2 - y_1} \sum_{y_j = y_1}^{y_2} [I(x, y_j) - IPF_v'(x)]^2$$

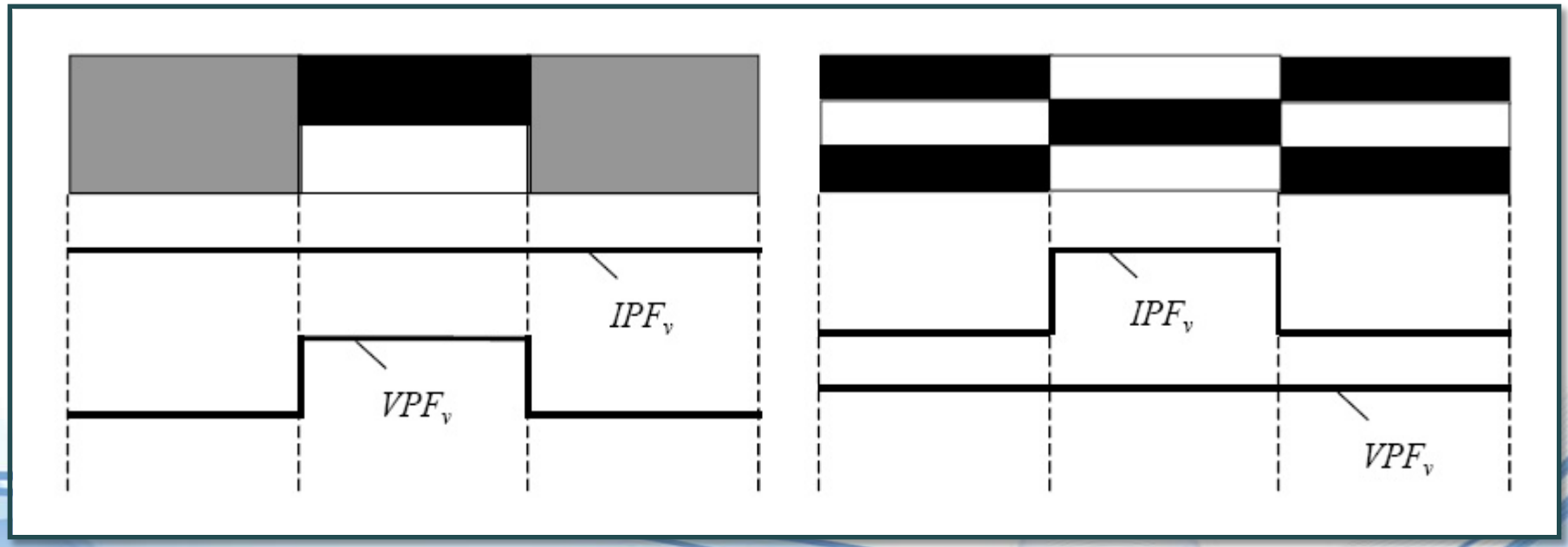
Proiectii de varianta - exemplu

[Feng98]



Proiectii - comparatie

[Zhou04]



Proiectii generalizate/hibride

[Zhou04]

- Proiectii generalizate = combinatie liniara de IPF si VPF

$$GPF_v(x) = (1-\alpha)IPF_v^{'}(x) + \alpha VPF_v(x)$$

$$GPF_h(y) = (1-\alpha)IPF_h^{'}(y) + \alpha VPF_h(y)$$

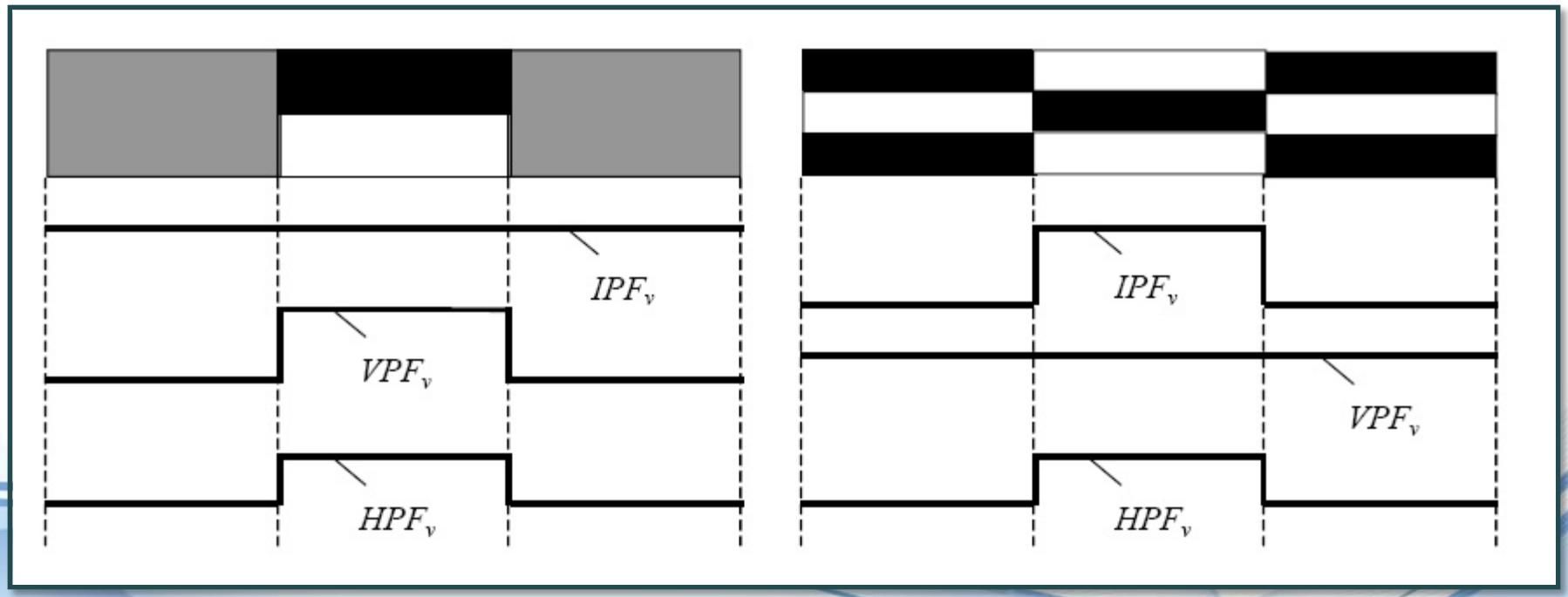
$$0 \le \alpha \le 1$$

- Proiectii hibride = caz particular de proiectii generalizate folosit pentru detectie de ochi

$$\alpha = 0.6$$

Proiectii - comparatie

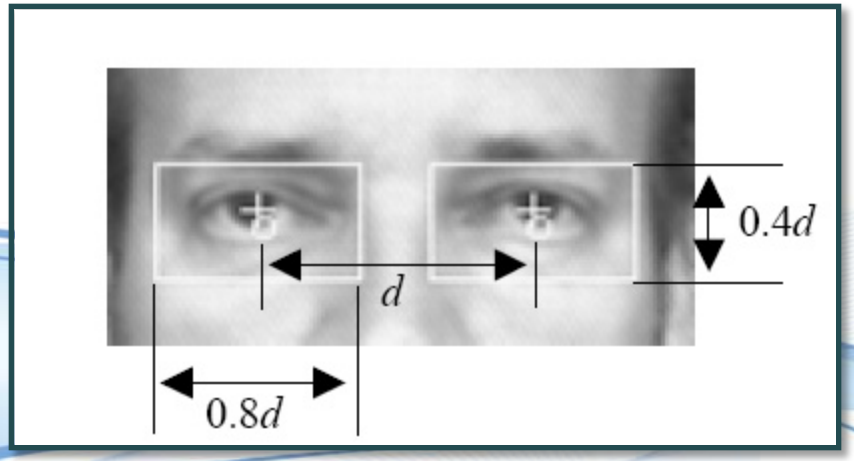
[Zhou04]



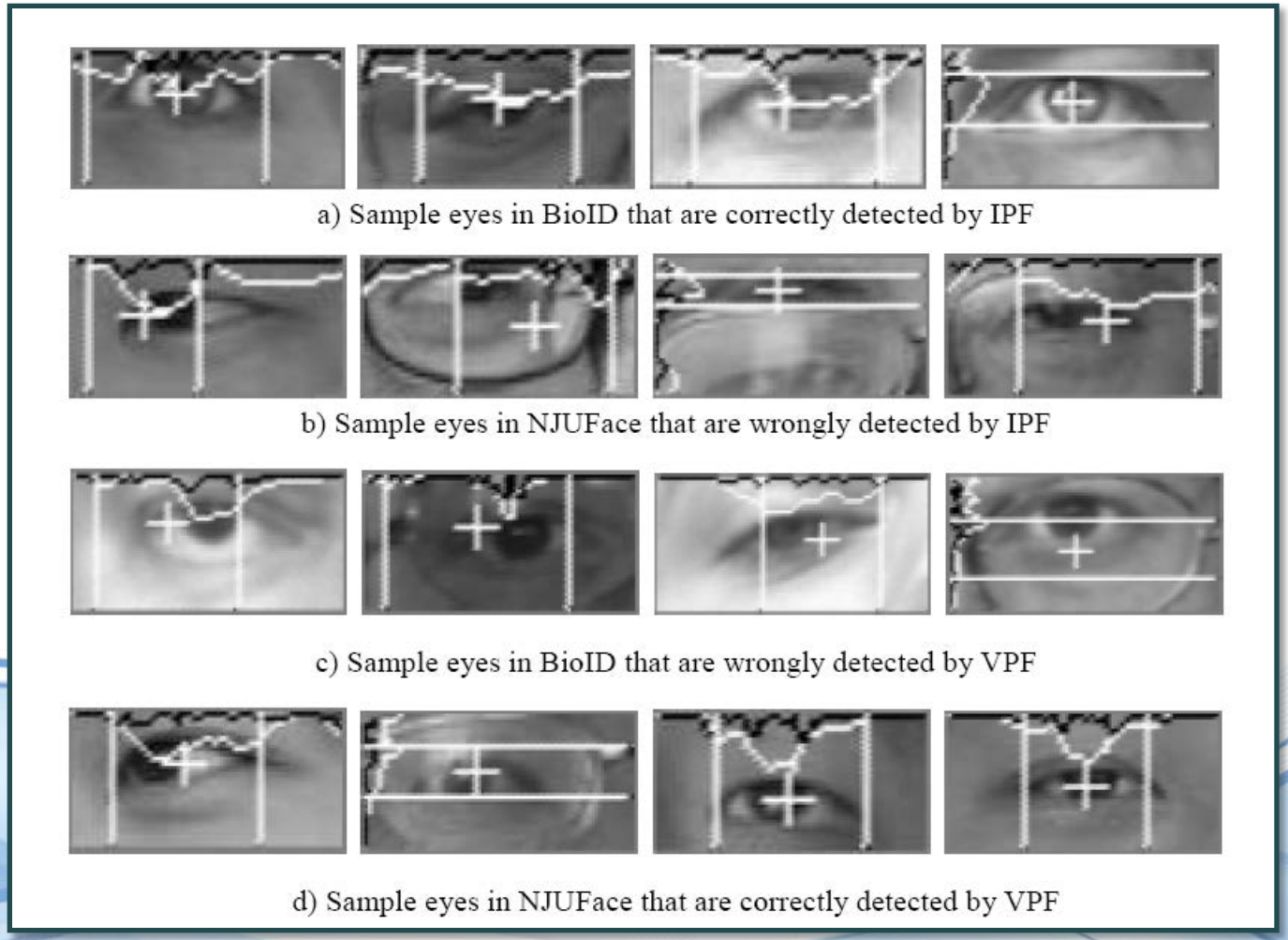
[Zhou04]

- Folosesc proiectii (si derivatele lor) pentru o rafinare a detectiei centrelor ochilor

- Pleaca de la o detectie grosiera facuta cu [Wu03]
- Cauta in zona inconjuratoare cu IPF, VPF, GPF si HPF



[Zhou04]



a) Sample eyes in BioID that are correctly detected by IPF

b) Sample eyes in NJUFace that are wrongly detected by IPF

c) Sample eyes in BioID that are wrongly detected by VPF

d) Sample eyes in NJUFace that are correctly detected by VPF

[Zhou04]

α	BioID	JAFFE	NJUFace
0.0	93.69%	96.71%	92.48%
0.1	93.82%	96.71%	92.76%
0.2	93.82%	97.18%	94.99%
0.3	94.21%	96.71%	95.26%
0.4	94.41%	97.18%	95.26%
0.5	94.41%	97.18%	95.54%
0.6	94.81%	97.18%	95.82%
0.7	91.85%	97.18%	95.54%
0.8	94.54%	97.18%	95.54%
0.9	94.48%	97.18%	95.26%
1.0	94.41%	97.18%	95.54%

func	BioID	JAFFE	NJUFace
IPF	93.69%	96.71%	92.48%
VPF	94.41%	97.18%	95.54%
HPF	94.81%	97.18%	95.82%

DLIB

Solutia pragmatica: **DLIB**

- Implementeaza *V. Kazemi, J. Sullivan "One millisecond face alignment with an ensemble of regression trees" IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1867-1874*
- Interfata usor de folosit in Python
- Antrenat pe iBUG 300-W dataset
- 64 landmark-uri
- **Robust !!!!**

Bibliografie

- [Asteriadis06] S. Asteriadis, N. Nikolaidis, A. Hajdu, I. Pitas, A novel eye detection algorithm utilizing edge-related geometrical information. *14th European Signal Processing Conf., 2006.*
- [Asteriadis09] S. Asteriadis, N. Nikolaidis, I. Pitas, Facial feature detection using distance vector fields. *Pattern Recognition, 42(7), 1388–1398, 2009.*
- [Asteriadis11] S. Asteriadis, N. Nikolaidis, I. Pitas, A Review of Facial Feature Detection Algorithms, *Zhang, Y. (Ed.), Advances in Face Image Analysis: Techniques and Technologies, pp. 42-61, 2011.*
- [Belhumeur97] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 711–720, 1997*
- [Bhuiyan03] M. A. Bhuiyan, V. Ampornaramveth, S. Muto, H. Ueno, Face detection and facial feature localization for human-machine interface, *National Institute of Informatics Journal, 5, pp. 25–39, 2003.*
- [Cootes98] T.F. Cootes, G.J. Edwards, C. Taylor, Active appearance models. *5th European Conf. on Computer Vision, Vol. 2, pp. 484-498, 1998.*
- [Cristinacce04] D. Cristinacce, T. Cootes, I. Scott, A multi-stage approach to facial feature detection. *15th British Machine Vision Conf., pp. 231-240, 2004.*
- [Feng98] G. C. Feng, P. C. Yuen. Variance projection function and its application to eye detection for human face recognition. *Pattern Recognition Letters 19(9), pp. 899-906, 1998.*

Bibliografie

- [Feris02] R.S. Feris, J. Gemmell, K. Toyama, V. Kruger, Hierarchical wavelet networks for facial feature localization. *5th Int. Conf. on Automatic Face and Gesture Recognition, pp. 125-130, 2002.*
- [Jesorsky01] O. Jesorsky, K.J. Kirchberg, R.W. Frischholz, Robust face detection using the Hausdorff distance. *3rd Int. Conf. on Audio and Video-based Biometric Person Authentication, pp. 90-95, 2001.*
- [Kanade00] T. Kanade, J. Cohn, Y.-L. Tian, Comprehensive database for facial expression analysis. *Int. Conf. on Automatic Face and Gesture Recognition, pp. 46- 53, 2000.*
- [Messer99] K. Messer, J. Matas, J. Kittler, J. Luettin, G. Maitre, XM2VTSDB: The extended M2VTS database. *2nd Int. Conf. on Audio and Video-based Biometric Person Authentication, pp. 72-77, 1999.*
- [Philips98] P. Phillips, H. Wechsler, J. Huang, P. Rauss, The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing, 16(5), 295–306, 1998.*
- [Phillips05] P. J. Phillips, P. J. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, Overview of the face recognition grand challenge, *IEEE Computer Vision and Pattern Recognition Conf., Vol. 1, pp. 947–954, 2005.*
- [Wu03] J. Wu, Z.-H. Zhou. Efficient face candidates selector for face detection. *Pattern Recognition 36(5), pp. 1175-1186, 2003.*
- [Yilmaz02] A. Yilmaz, M.A. Shah, Automatic feature detection and pose recovery for faces. *Asian Conf. on Computer Vision, pp. 23-25, 2002.*
- [Zhou04] Z.H. Zhou, X. Geng, Projection functions for eye detection. *Pattern Recognition, 37(5), pp. 1049–1056, 2004.*