Vedere Artificială
(Computer Vision)

Bogdan Alexe

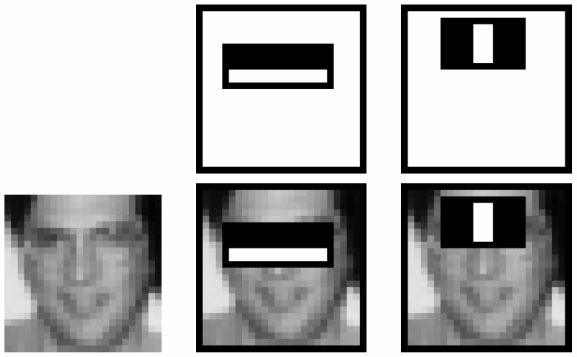
bogdan.alexe@fmi.unibuc.ro

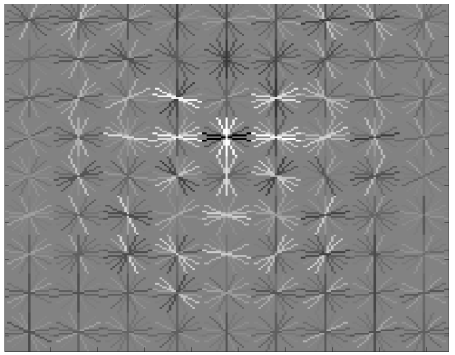
anul 2, master Informatică, semestrul I, 2019-2020

Recapitulare – cursul trecut

- Detectarea de obiecte în imagini cu R-CNN

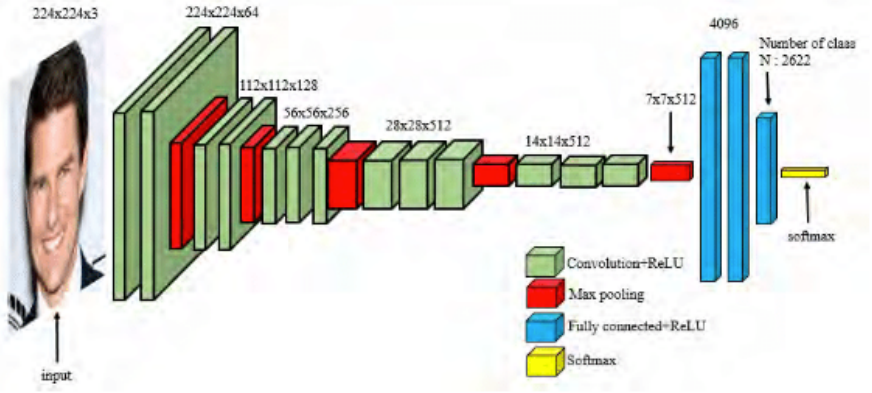
- Detectare facială
  - Viola-Jones
  - HOG



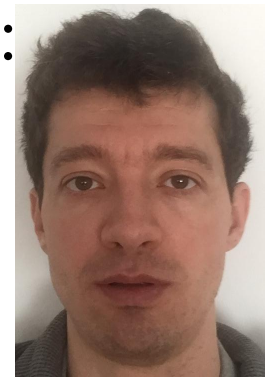
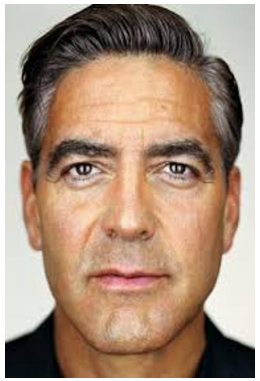


Cursul de azi

- Detectare facială
  - CNN-uri



- Tema 3 – Detectare facială pentru:
  - transformarea fețelor în alte fețe
  - recunoașterea de acțiuni


Dificultăți în detectarea facială

- performanța în detectarea facială a tot crescut

- detectarea fețelor mici este încă o problemă grea

- baza de date WIDER, conținând un număr mare de imagini - fiecare cu multe fețe – arată diferența dintre performanța umană și detectorii faciali cei mai buni.

S. Yang, P. Luo, C. C. Loy, X. Tang: WIDER Face*: a face detection benchmark detector*, CVPR 2016

WIDER FACE: A Face Detection Benchmark

Shuo Yang Ping Luo Chen Change Loy Xiaoou Tang

Department of Information Engineering, The Chinese University of Hong Kong

{ys014, pluo, ccloy, xtang}@ie.cuhk,edu.hk

S. Yang, P. Luo, C. C. Loy, X. Tang: WIDER Face: *a face detection benchmark detector*, CVPR 2016

WIDER FACE: A Face Detection Benchmark

Shuo Yang Ping Luo Chen Change Loy Xiaoou Tang

Department of Information Engineering, The Chinese University of Hong Kong

{ys014, pluo, ccloy, xtang}@ie.cuhk,edu.hk

S. Yang, P. Luo, C. C. Loy, X. Tang: WIDER Face*: a face detection benchmark detector*, CVPR 2016

WIDER FACE: A Face Detection Benchmark

Shuo Yang Ping Luo Chen Change Loy Xiaoou Tang

Department of Information Engineering, The Chinese University of Hong Kong

{ys014, pluo, ccloy, xtang}@ie.cuhk,edu.hk

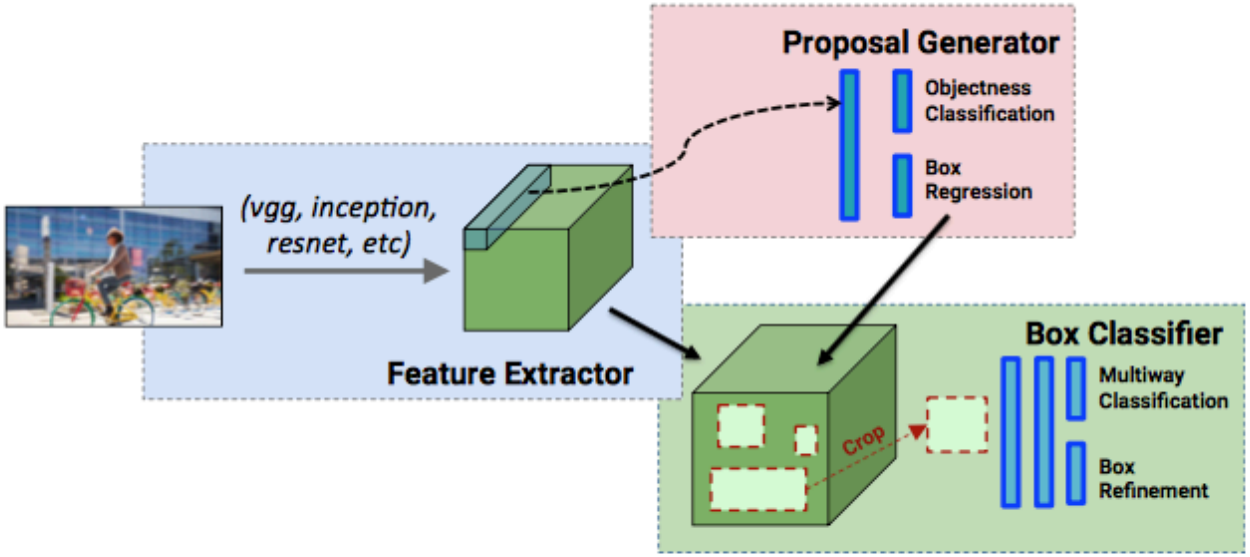
Table 1. Comparison of face detection datasets.

Dataset	Training		Testing		Height			Properties		
	#Image	#Face	#Image	#Face	10-50 pixels	50-300 pixels	≤ 300 pixels	Occlusion labels	Event labels	Pose labels
AFW [30]	-	-	0.2k	0.47k	12%	70%	18%	-	-	✓
FDDB [12]	-	-	2.8k	5.1k	8%	86%	6%	-	-	-
PASCAL FACE [24]	-	-	0.85k	1.3k	41%	57%	2%	-	-	-
IJB-A [13]	16k	33k	8.3k	17k	13%	69%	18%	-	-	-
MALF [26]	-	-	5.25k	11.9k	N/A	N/A	N/A	✓	-	✓
WIDER FACE	16k	**199k**	16k	**194k**	50%	43%	7%	✓	✓	✓

box is denoted in green color. The WIDER FACE dataset consists of 393, 703 labeled face bounding boxes in 32, 203 images (**Best view in color).**

AFW – Annotated Faces in the Wild
FDDB – Face Detection Data set and Benchmark
IJBA - IARPA Janus Benchmark A dataset
MALF - Multi-Attribute Labelled Faces

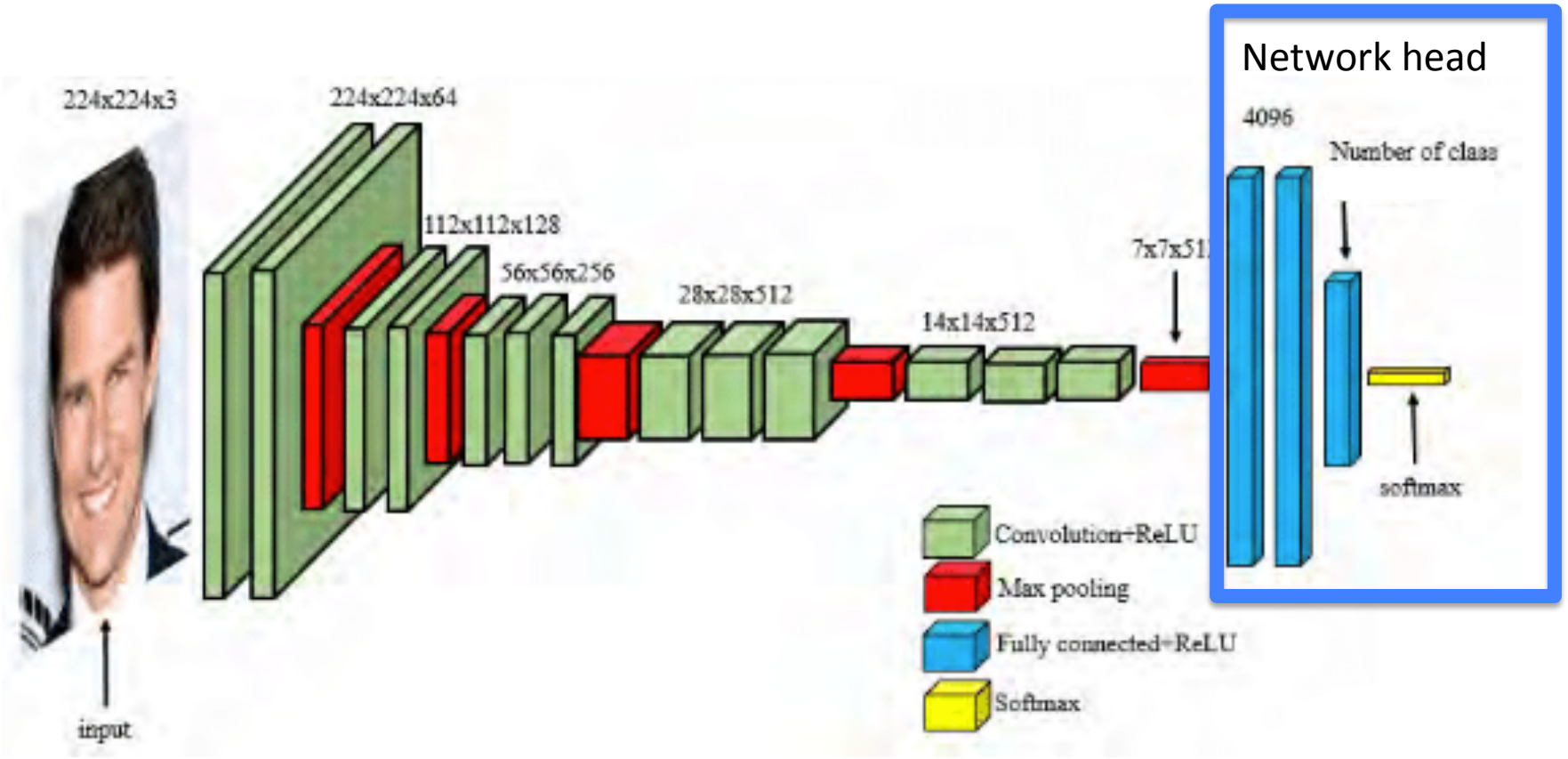
Detectori faciali pe bază de CNN-uri



-folosesc paradigma Faster R-CNN, detector în două etape.
Caracteristicle unui CNN (VGG, RESNET, INCEPTION) folosite pentru

1. a găsi ferestre candidat (poziția din imagine unde pot fi fețele) – învață un RPN (region proposal network)
2. clasificarea ferestrelor candidat folosind o rețea
   - FC layers ➔ network head (120M de parametri în VGG16)

VGG-16 pentru detectare facială

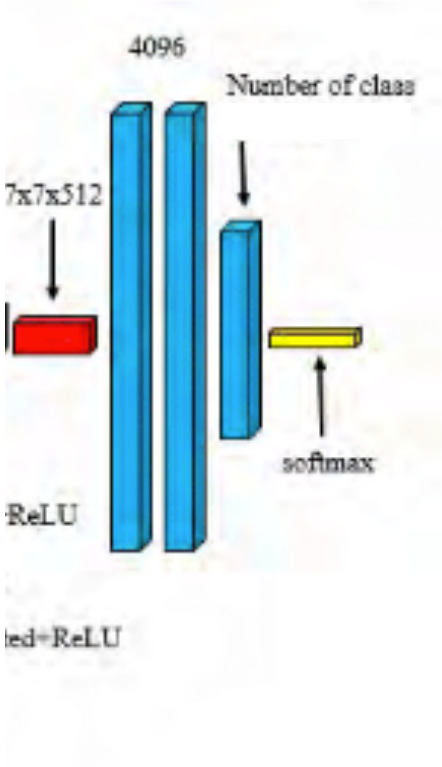


Numărul de parametri în VGG-16

```
INPUT:     [224x224x3]    memory:  224*224*3=150K   weights: 0
CONV3-64:  [224x224x64]   memory:  224*224*64=3.2M  weights: (3*3*3)*64 = 1,728
CONV3-64:  [224x224x64]   memory:  224*224*64=3.2M  weights: (3*3*64)*64 = 36,864
POOL2:     [112x112x64]   memory:  112*112*64=800K  weights: 0
CONV3-128: [112x112x128]  memory:  112*112*128=1.6M weights: (3*3*64)*128 = 73,728
CONV3-128: [112x112x128]  memory:  112*112*128=1.6M weights: (3*3*128)*128 = 147,456
POOL2:     [56x56x128]    memory:  56*56*128=400K   weights: 0
CONV3-256: [56x56x256]    memory:  56*56*256=800K   weights: (3*3*128)*256 = 294,912
CONV3-256: [56x56x256]    memory:  56*56*256=800K   weights: (3*3*256)*256 = 589,824
CONV3-256: [56x56x256]    memory:  56*56*256=800K   weights: (3*3*256)*256 = 589,824
POOL2:     [28x28x256]    memory:  28*28*256=200K   weights: 0
CONV3-512: [28x28x512]    memory:  28*28*512=400K   weights: (3*3*256)*512 = 1,179,648
CONV3-512: [28x28x512]    memory:  28*28*512=400K   weights: (3*3*512)*512 = 2,359,296
CONV3-512: [28x28x512]    memory:  28*28*512=400K   weights: (3*3*512)*512 = 2,359,296
POOL2:     [14x14x512]    memory:  14*14*512=100K   weights: 0
CONV3-512: [14x14x512]    memory:  14*14*512=100K   weights: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512]    memory:  14*14*512=100K   weights: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512]    memory:  14*14*512=100K   weights: (3*3*512)*512 = 2,359,296
POOL2:     [7x7x512]      memory:  7*7*512=25K      weights: 0
FC:        [1x1x4096]     memory:  4096             weights: 7*7*512*4096 = 102,760,448
FC:        [1x1x4096]     memory:  4096             weights: 4096*4096 = 16,777,216
FC:        [1x1x1000]     memory:  1000             weights: 4096*1000 = 4,096,000

TOTAL memory: 24M * 4 bytes ~= 93MB / image (only forward! ~*2 for bwd)
TOTAL params: 138M parameters
```

Network head



For face detection fc3 has 4096 x 2 classes = 8192 params

SSH: Single Stage Headless Face Detector

Mahyar Najibi* Pouya Samangouei* Rama Chellappa Larry S. Davis
University of Maryland
najibi@cs.umd.edu {pouya,rama,lsd}@umiacs.umd.edu

Abstract

We introduce the Single Stage Headless (SSH) face detector. Unlike two stage proposal-classification detectors, SSH detects faces in a single stage directly from the early convolutional layers in a classification network. SSH is headless. That is, it is able to achieve state-of-the-art results while removing the "head" of its underlying classification network – i.e. *all fully connected layers in the VGG-16 which contains a large number of parameters. Additionally, instead of relying on an image pyramid to detect faces with various scales, SSH is scale-invariant by design. We simultaneously detect faces with different scales in a single forward pass of the network, but from different layers. These properties make SSH fast and light-weight. Surprisingly, with a headless VGG-16, SSH beats the ResNet-101-based state-of-the-art on the WIDER dataset. Even though, unlike the current state-of-the-art, SSH does not use an image pyramid and is* $5X$ *faster. Moreover, if an image pyramid is deployed, our light-weight network achieves state-of-the-art on all subsets of the WIDER dataset, improving the AP by* 2.5%. *SSH also reaches state-of-the-art results on the FDDB and Pascal-Faces datasets while using a small input size, leading to a runtime of* 50 *ms/image on a GPU. The code is available at* https://github.com/mahyarnajibi/SSH.

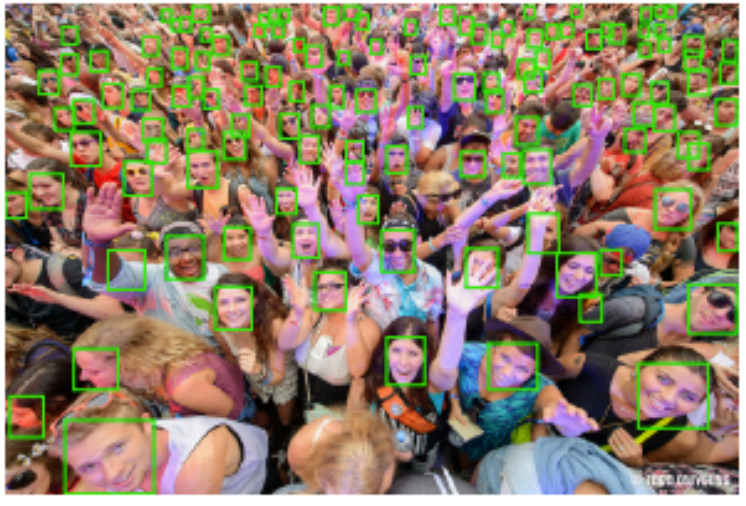


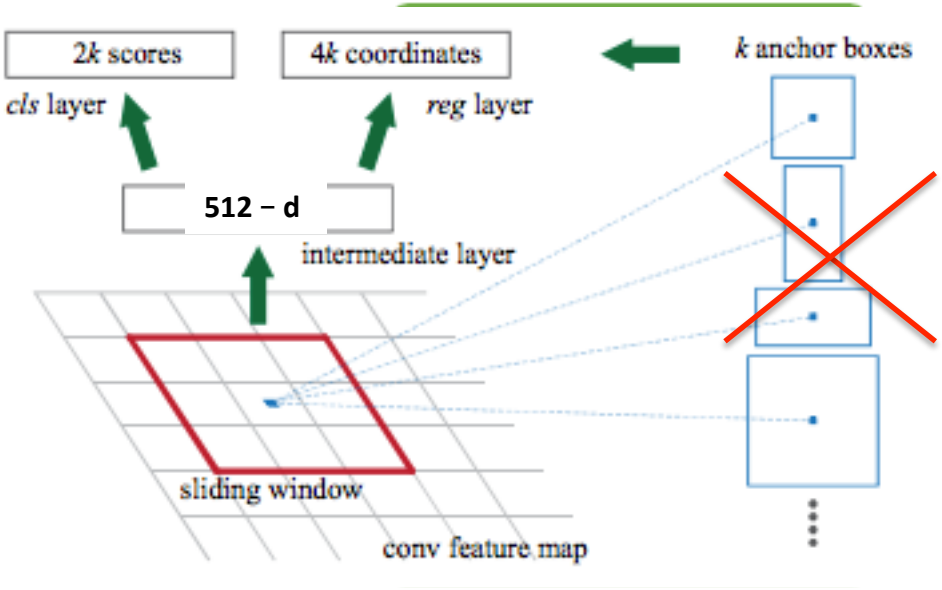
Figure 1: *SSH* detects various face sizes in a single CNN forward pass and without employing an image pyramid in $\sim$ 0.1 second for an image with size 800×1200 on a GPU.

ing face detectors are usually slow and have high memory foot-prints (*e.g.* [7] takes more than 1 second to process an image, see Section 4.5) partly due to the huge number of parameters as well as the way robustness to scale or incorporation of context are addressed.

State-of-the-art CNN-based detectors convert image classification networks into two-stage detection systems [4, 24]. In the first stage, early convolutional feature maps

Anchor boxes

SSH regresses a set of predefined bounding boxes called *anchors*, to the ground-truth faces. We employ a similar strategy to the *RPN* [24] to form the anchor set. We define the anchors in a dense overlapping sliding window fashion. At each sliding window location, K anchors are defined which have the same center as that window and different scales. However, unlike *RPN*, we only consider anchors with aspect ratio of one to reduce the number of anchor boxes. We noticed in our experiments that having various aspect ratios does not have a noticeable impact on face detection precision. More formally, if the feature map connected to the detection module $\mathcal{M}_i$ has a size of $W_i \times H_i$, there would be $W_i \times H_i \times K_i$ anchors with aspect ratio one and scales $\{S_i^1, S_i^2, \dots S_i^{K_i}\}$.



Only one scale = square bonding boxes

M. Najibi, P. Samangouei, R. Chellappa, LS. Davis: *SSH: Single Stage Headless Face Detector*, ICCV 2017

Soluția propusă

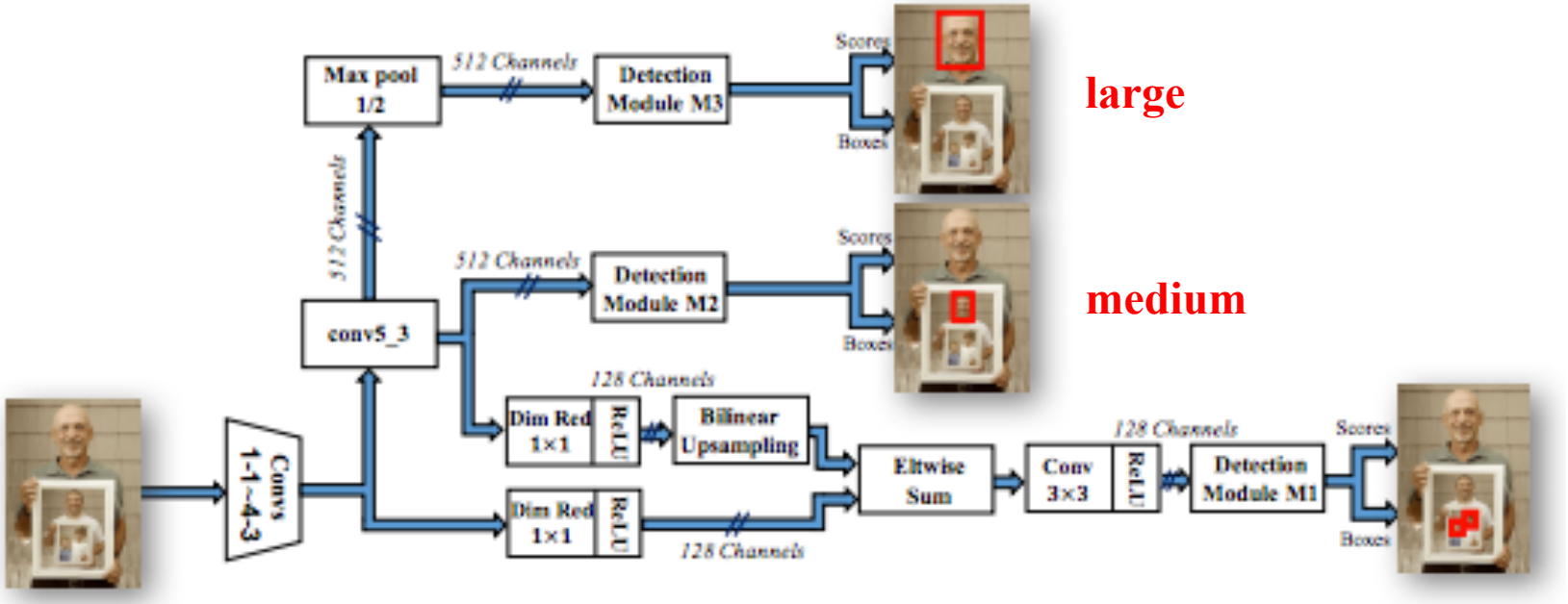
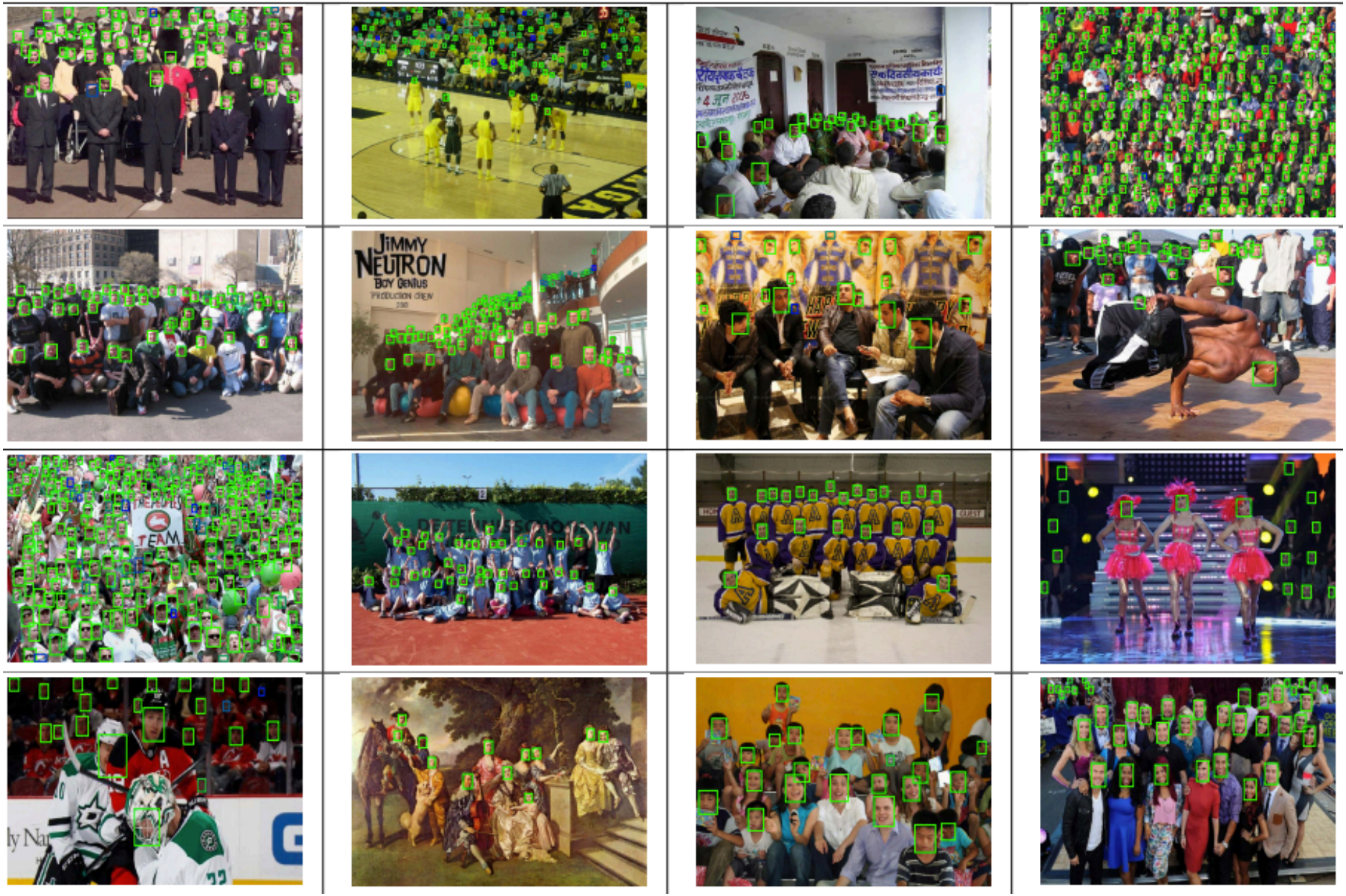


Figure 2 shows the general architecture of *SSH*. It is a fully convolutional network which localizes and classifies faces early on by adding a *detection module* on top of feature maps with strides of 8, 16, and 32, depicted as $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ respectively. The *detection module* consists of a convolutional binary classifier and a regressor for detecting faces and localizing them respectively.

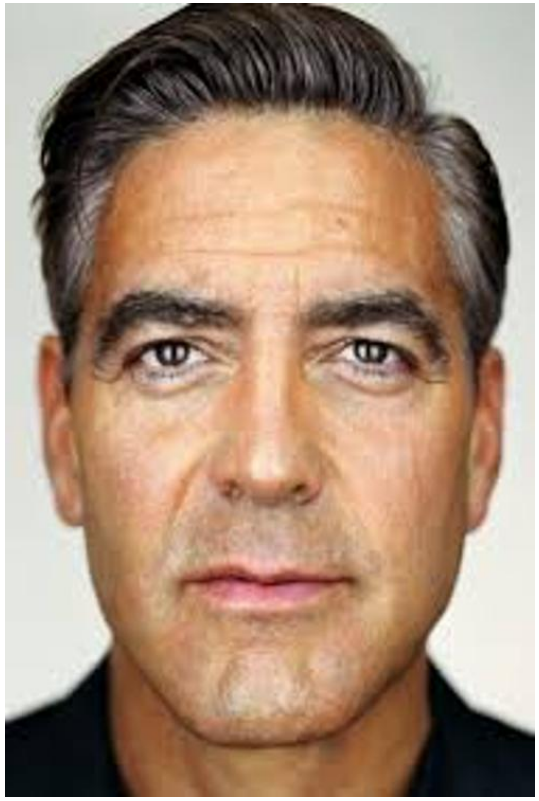
M. Najibi, P. Samangouei, R. Chellappa, LS. Davis: *SSH: Single Stage Headless Face Detector*, ICCV 2017

Rezultate calitative



Rezultate calitative

Tema 3: Detectare facială pentru transformarea fețelor în alte fețe



Tema 3: transformarea fețelor în alte fețe

Tema 3: transformarea fețelor în alte fețe

https://www.youtube.com/watch?v=nUDIoN-_Hxs

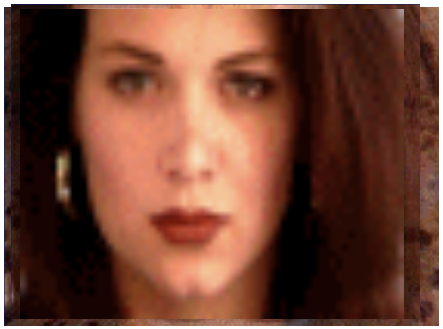
Tema 3: transformarea fețelor în alte fețe

http://youtube.com/watch?v=3ZHtL7CirJA

Tema 3: transformarea fețelor în alte fețe

https://www.youtube.com/watch?v=XHb7Ig3yPgI

Tranziția între obiecte


Realizează o tranziție treptată, în timp, de la obiectul sursă la obiectul țintă

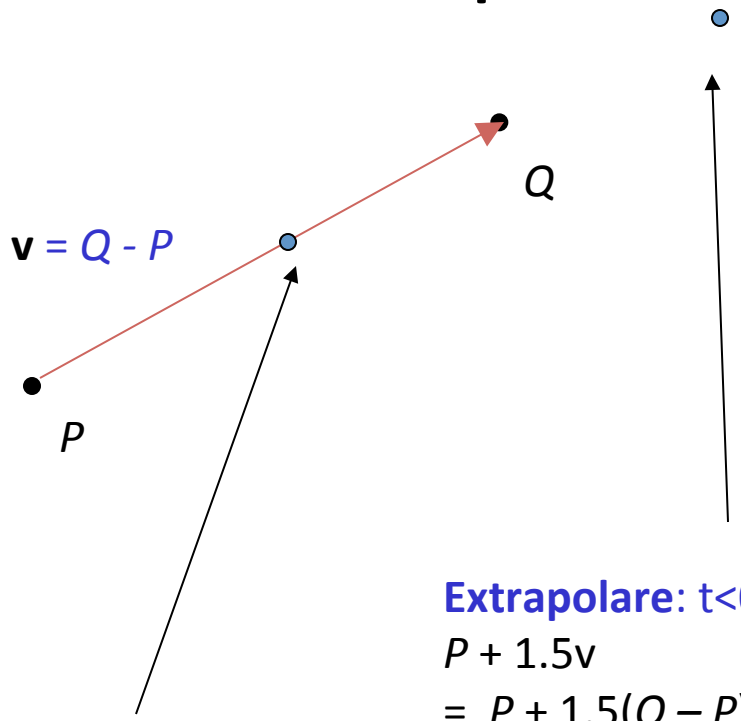
- realizează o "medie ponderată" a obiectelor în timp
- timpul $T = \{t_0=0, t_1, t_2, t_3, \ldots, t_n=1\}$
- pentru $t_0 = 0$, obținem obiectul sursă
- pentru $t_n = 1$, obținem obiectul țintă
- pentru $t_{n/2} = 1/2$, obținem obiectul "mediu"

Scopul este de a găsi "o medie" între două obiecte

- NU este media imaginilor ce conțin obiectele…
- …este imaginea obiectului "mediu"

Media ponderată a două puncte

Care este media punctelor P și Q?



Extrapolare: t<0 or t>1

$P + 1.5v$

$= P + 1.5(Q - P)$

$= -0.5P + 1.5\ Q$ *(t=1.5)*

$P + 0.5v$

$= P + 0.5(Q - P)$

$= 0.5P + 0.5\ Q$

Interpolare liniară

Obținem un punct nou:

(1-t)P + tQ, 0<t<1

- P și Q pot fi orice:
  - puncte în plan (2D) sau în spațiu (3D)
  - culori în spațiul RGB (3D)
  - imagini … etc.

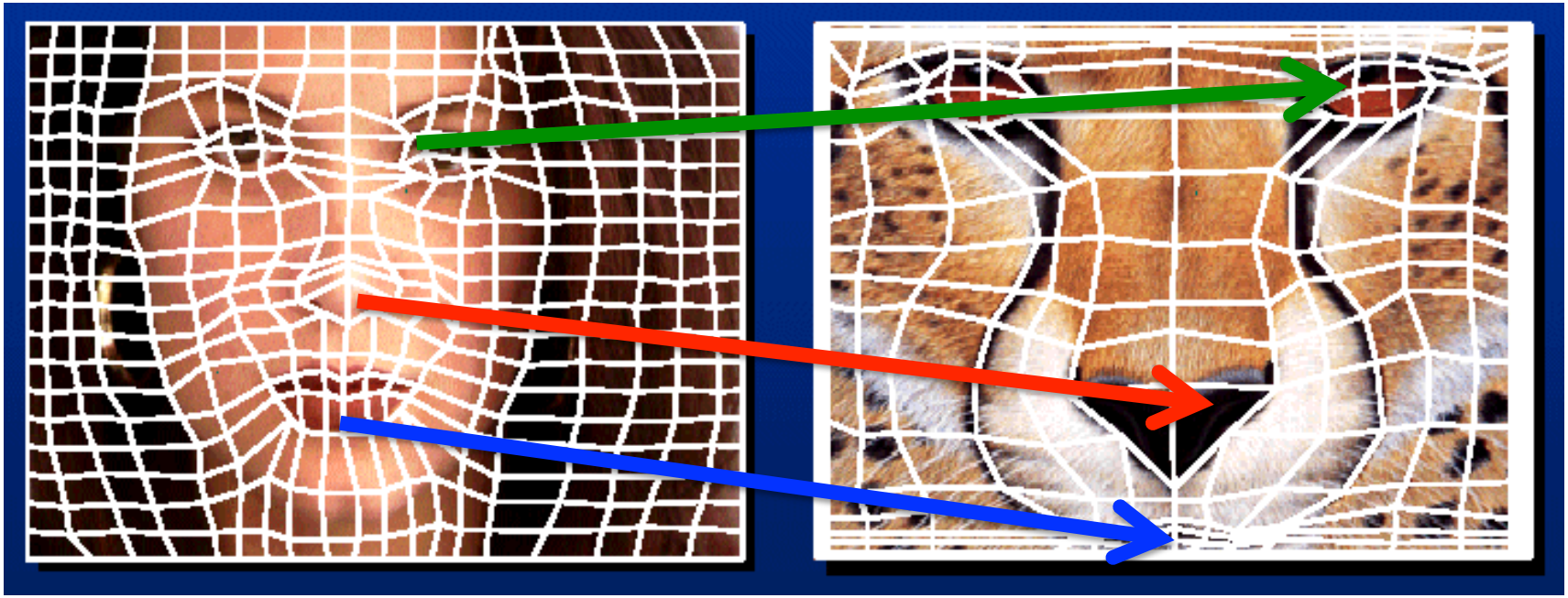
Tranziția globală de la o imagine la alta

- interpolare la nivel de imagini (P și Q sunt imagini):
- $Imagine_t = (1-t)*Imagine_1 + t*Imagine_2$
- tranziție în culoare + formă
- merge bine cât timp imaginile sunt aliniate în formă (contururile se suprapun foarte bine)
- ce se întâmplă dacă nu avem imagini aliniate?

Aliniere + tranziția globală

- mai întâi aliniază imaginile, apoi realizează tranziția
  - găsește transformarea globală geometrică dintre imagini (fiecare punct din imaginea 1 se transformă într-un punct din imaginea 2)

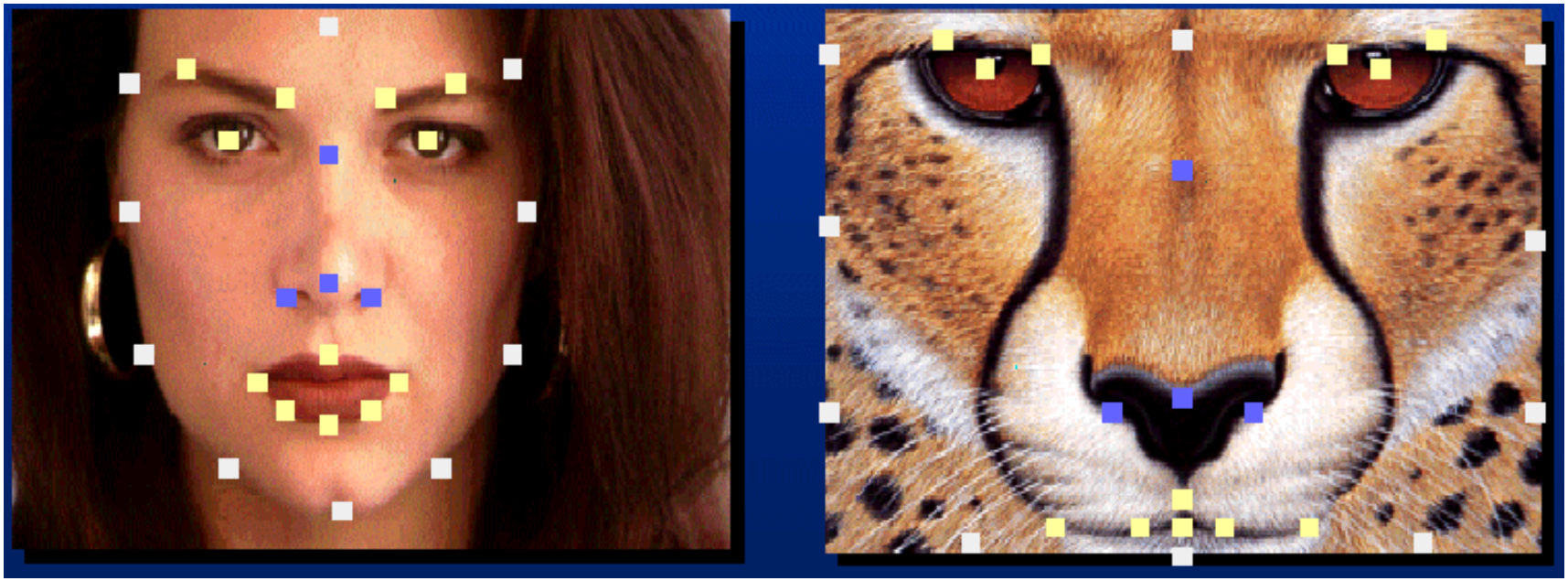
Alinierea obiectelor – varianta 1



Pe bază de caroiaj cu regiuni corespondente:

- fiecare regiune din imaginea sursă va suferi transformări geometrice + fotometrice (de culoare) pentru a ajunge la regiunea corespondentă din imaginea țintă

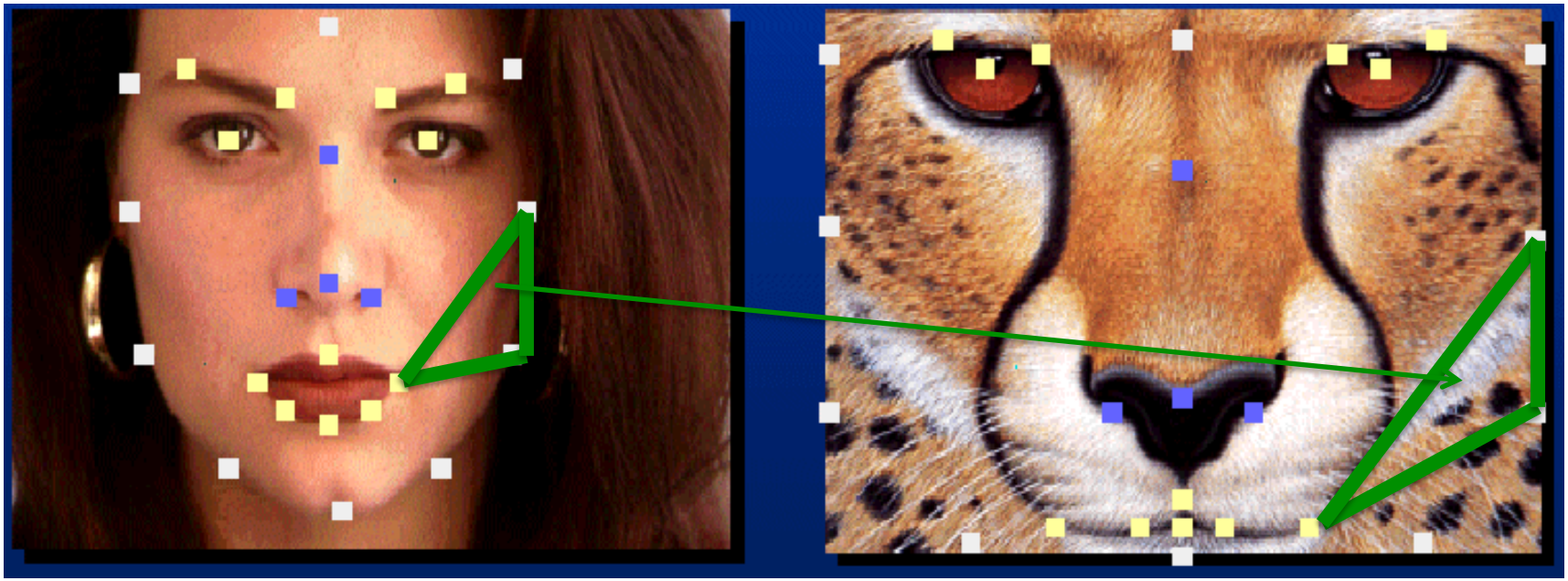
Alinierea obiectelor – varianta 2



Pe bază de puncte corespondențe:

- obținem regiuni = triunghiuri folosind un algoritm de triangulare (Delaunay). Toate regiunile formează o rețea de triunghiuri.
- fiecare regiune (triunghi) din imaginea sursă va fi transformată geometric + fotometric pentru a ajunge la regiunea corespondentă (triunghiul corespondent) din imaginea țintă

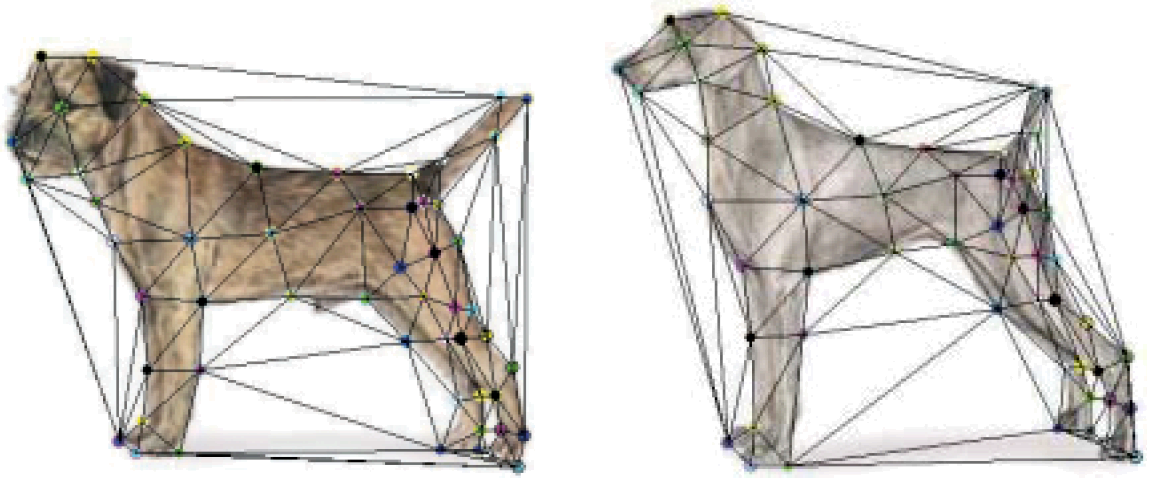
Alinierea obiectelor – varianta 2



Pe bază de puncte corespondențe:

- obținem regiuni = triunghiuri folosind un algoritm de triangulare (Delaunay). Toate regiunile formează o rețea de triunghiuri.
- fiecare regiune (triunghi) din imaginea sursă va fi transformată geometric + fotometric pentru a ajunge la regiunea corespondentă (triunghiul corespondent) din imaginea țintă

Rețea de triunghiuri

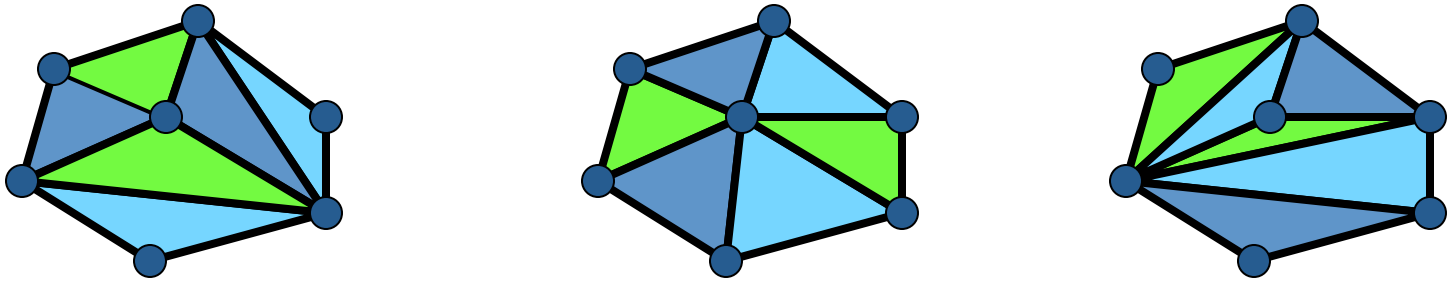


1. Determinăm o rețea de triunghiuri pe baza punctelor inițiale
   - aceeași rețea în ambele imagini
   - avem corespondențe între triunghiuri
2. Transformăm fiecare triunghi separat din imaginea sursă în imaginea țintă
   - transformare geometrică = transformare afină
   - transformare fotometrică = tranziție de culoare

Triangulare

• O *triangulare* a unui set de puncte în planul 2D este o *partiție* a acoperirii convexe a setului de puncte în triunghiuri ale căror vârfuri sunt punctele și nu conțin nici un alt punct în interior (triunghiurile nu se “taie”)

• Există un număr exponențial de triangulări ale unui set de puncte

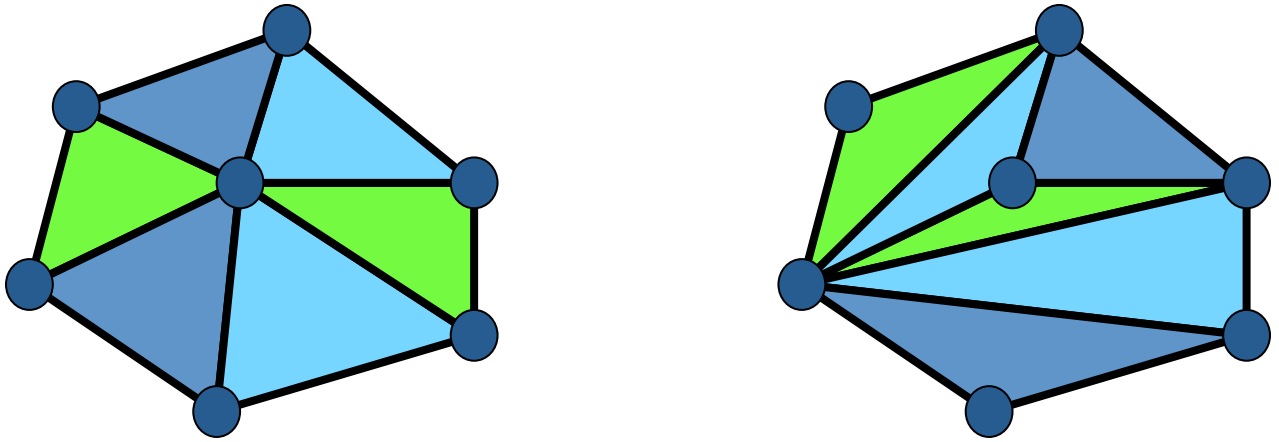


Criteriu de a măsura triangulările

• Fie $\alpha(T_i) = (\alpha_{i1}, \alpha_{i2}, .., \alpha_{in})$ vectorul cu unghiurile ordonate crescător din toate triunghirile unei triangulări T_i

• O triangulare T_1 este "mai bună" decât T_2 dacă cel mai mic unghi al lui T_1 este mai mare decât cel mai mic unghi al lui T_2

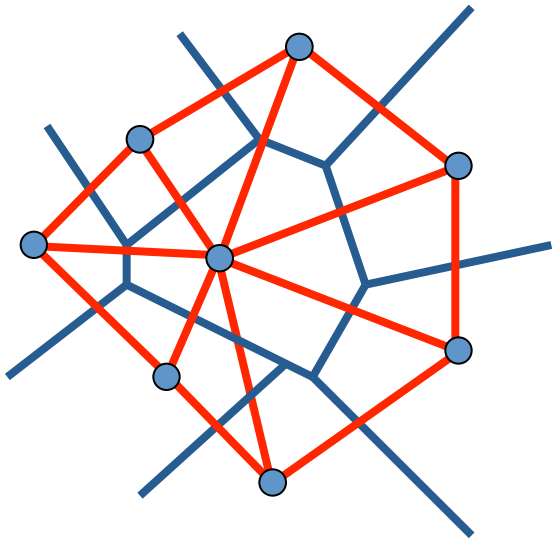
• Triangulările Delaunay sunt cele mai "bune" (maximizează cele mai mici unghiuri)



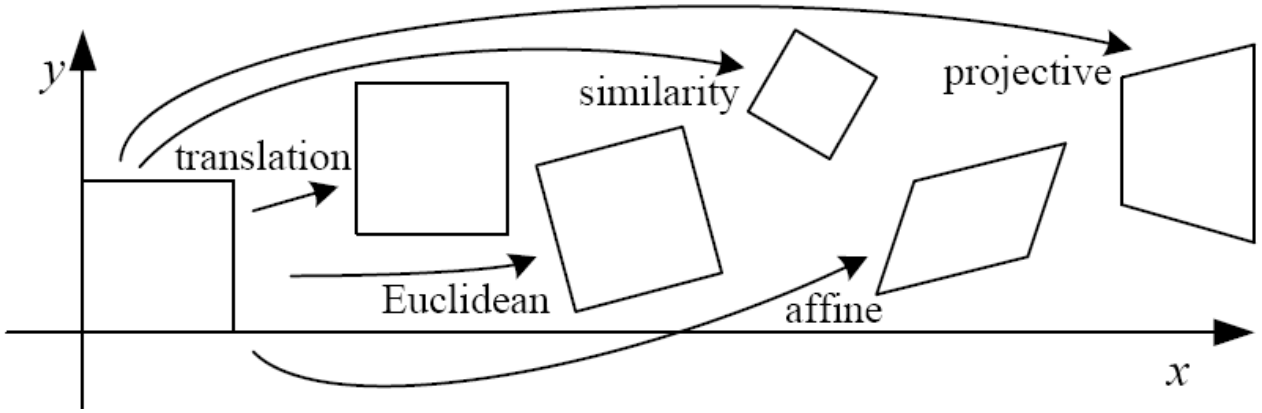
triangularea din stânga este "mai buna" decât cea din dreapta

Algoritmul de triangulare a lui Delauney

- găsește diagrama Voronoi pe baza setului de puncte 2D;
- conectează toate regiunile vecine
- în limbaje de programare aveți o asemenea funcție (în MATLAB – funcția delaunay)



Transformări geometrice 2D



Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\left[\boldsymbol{I} \mid \boldsymbol{t} \right]_{2\times 3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\left[\boldsymbol{R} \mid \boldsymbol{t} \right]_{2\times 3}$	3	lengths $+\cdots$	
similarity	$\left[s\boldsymbol{R} \mid \boldsymbol{t} \right]_{2\times 3}$	4	angles $+\cdots$	
affine	$\left[\boldsymbol{A} \right]_{2\times 3}$	6	parallelism $+\cdots$	
projective	$\left[\tilde{\boldsymbol{H}} \right]_{3\times 3}$	8	straight lines	

- translație, translație + rotație (euclidiană), translație + rotație + scalare (similaritate), afină, proiectivă

Transformări (globale) parametrice

Exemple de transformări parametrice:

translație

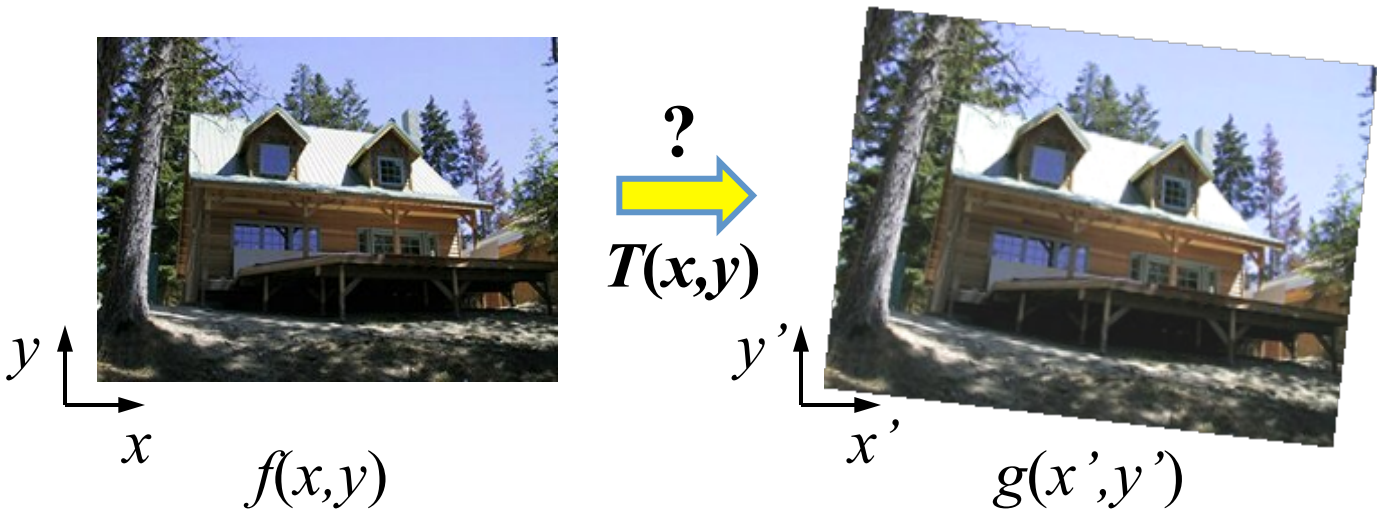
rotație

scalare

afină

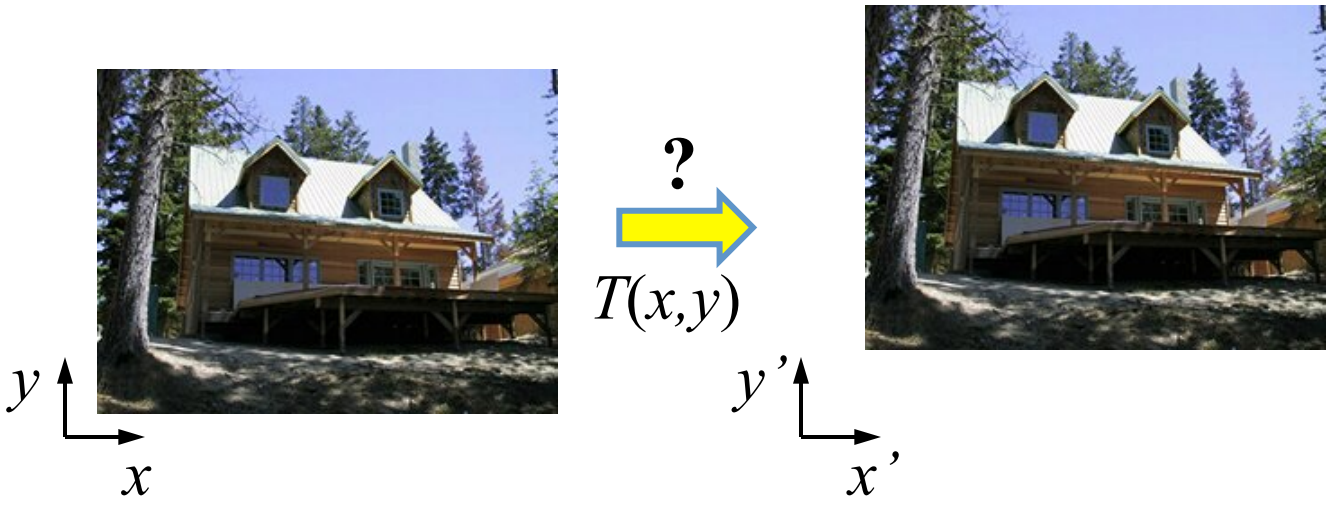
perspectivă

Calculul transformărilor



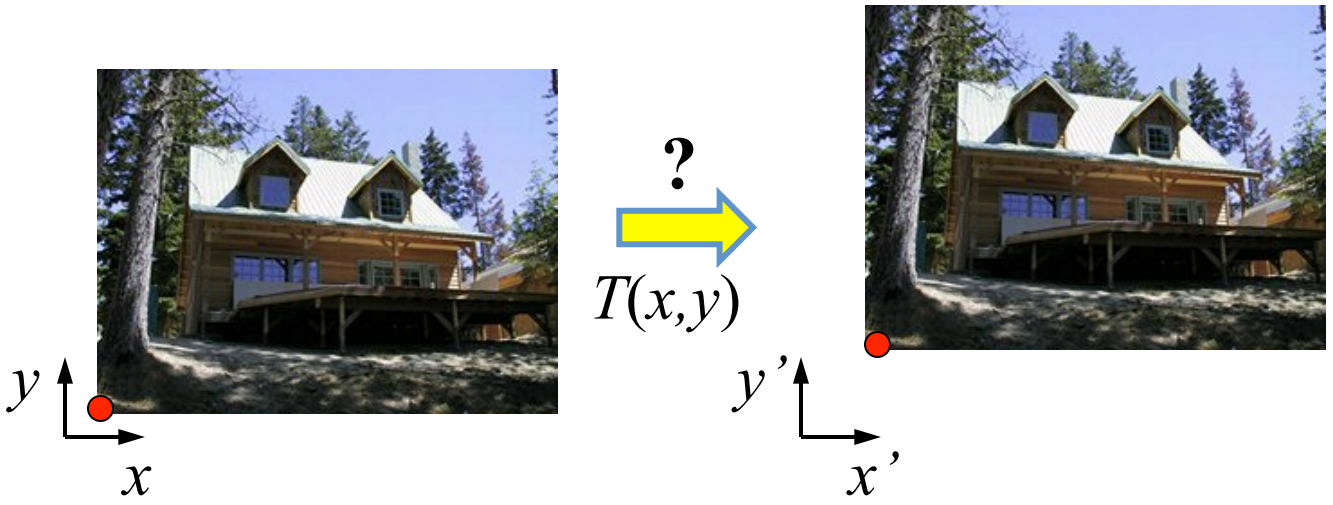
- Dacă știu *f* și *g* cum calculez transformarea T?
  - presupunem că avem perechi de puncte corespondente (date de utilizator sau calculate automat)
    - de câte asemenea perechi este nevoie?

Translație: # corespondențe?



- Câte grade de libertate are o asemenea transformare?
- De câte corespondențe am nevoie?
- Care este matricea transformării?

Translație: # corespondențe?



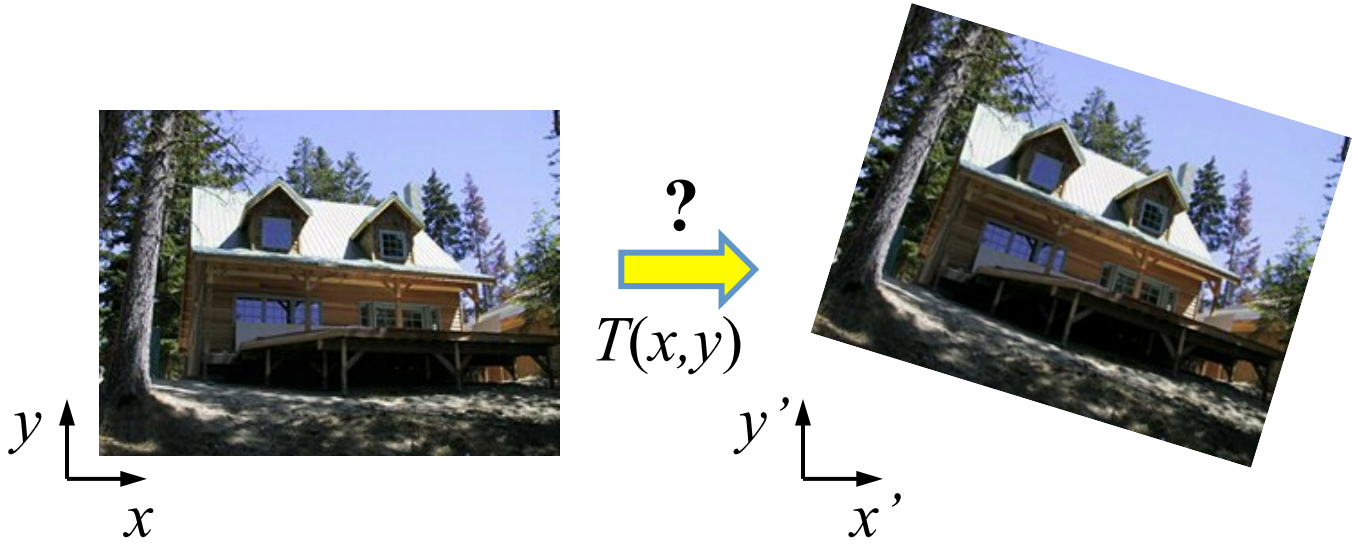
O singură pereche de puncte corespondente este suficientă pentru a găsi translația (2 grade de libertate).

Coordonate omogene

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

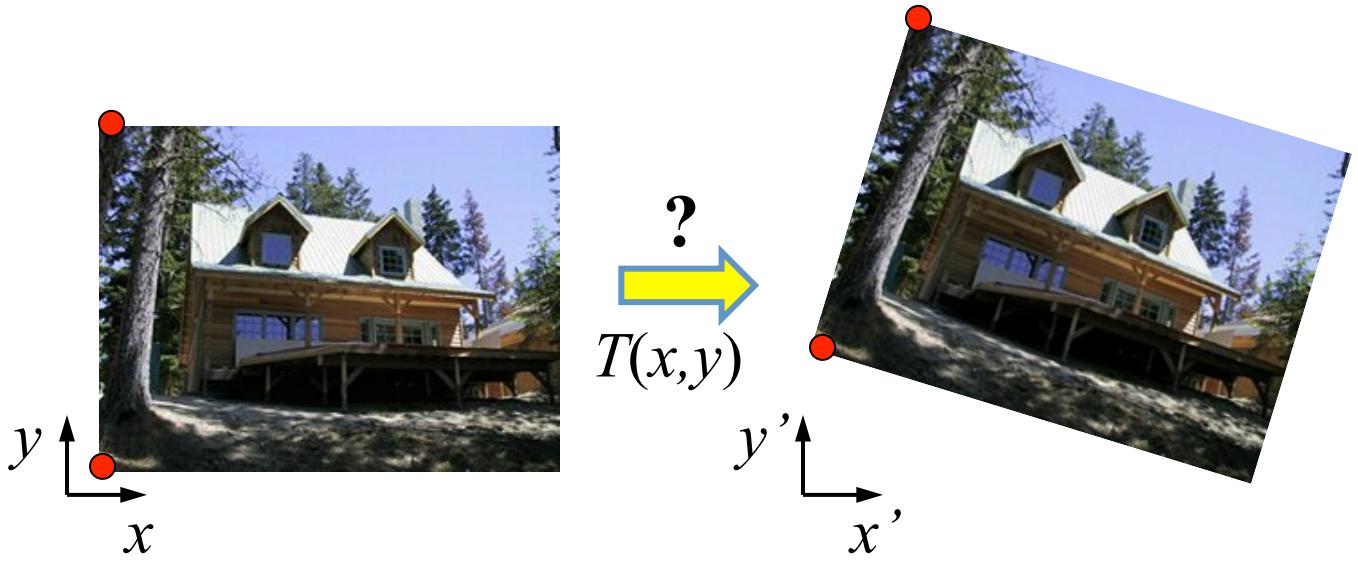
$$\mathbf{M} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Transformarea euclidiană (translație + rotație): # corespondențe?



- Câte grade de libertate are o asemenea transformare?
- De câte corespondențe am nevoie?
- Care este matricea transformării?

Transformarea euclidiană (translație + rotație): # corespondențe?



Două perechi de puncte corespondente sunt suficiente pentru a găsi transformarea euclidiană (3 grade de libertate).

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

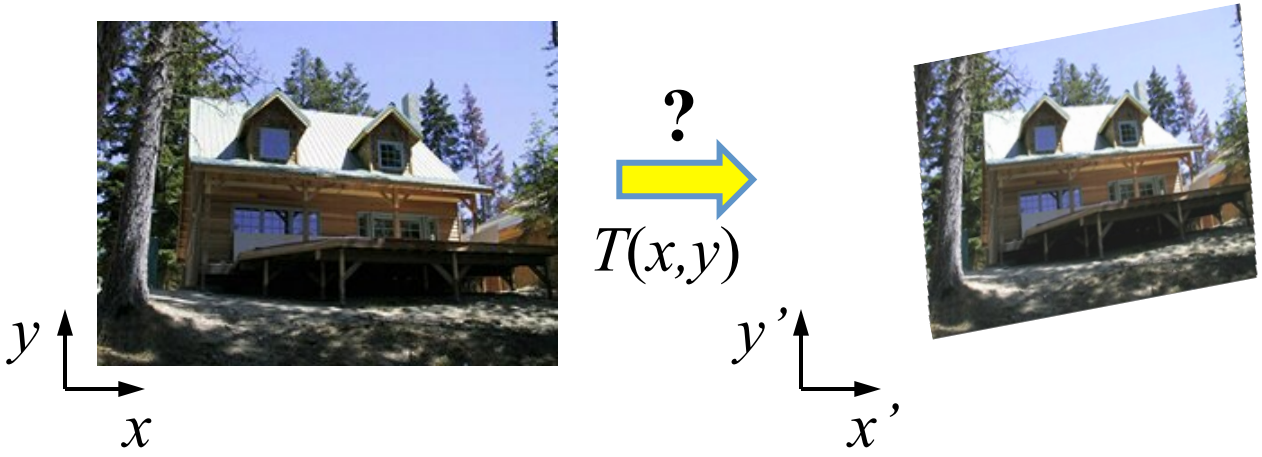
translație

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

rotație

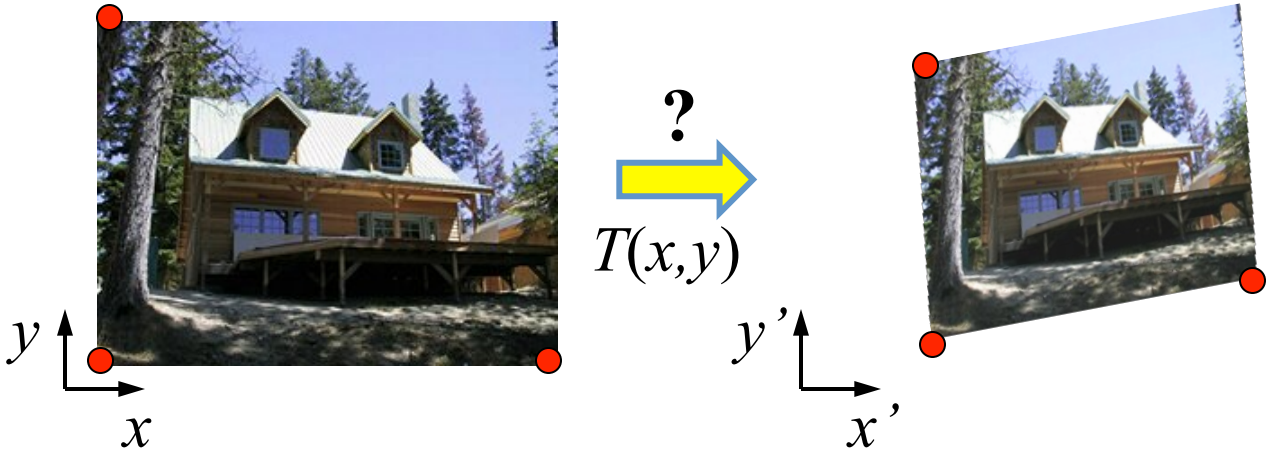
$$\mathbf{M} = \begin{bmatrix} \cos\Theta & -\sin\Theta & t_x \\ \sin\Theta & \cos\Theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Transformarea afină: # corespondențe?



- Câte grade de libertate are o asemenea transformare?
- De câte corespondențe am nevoie?
- Care este matricea transformării?

Transformarea afină: # corespondențe?



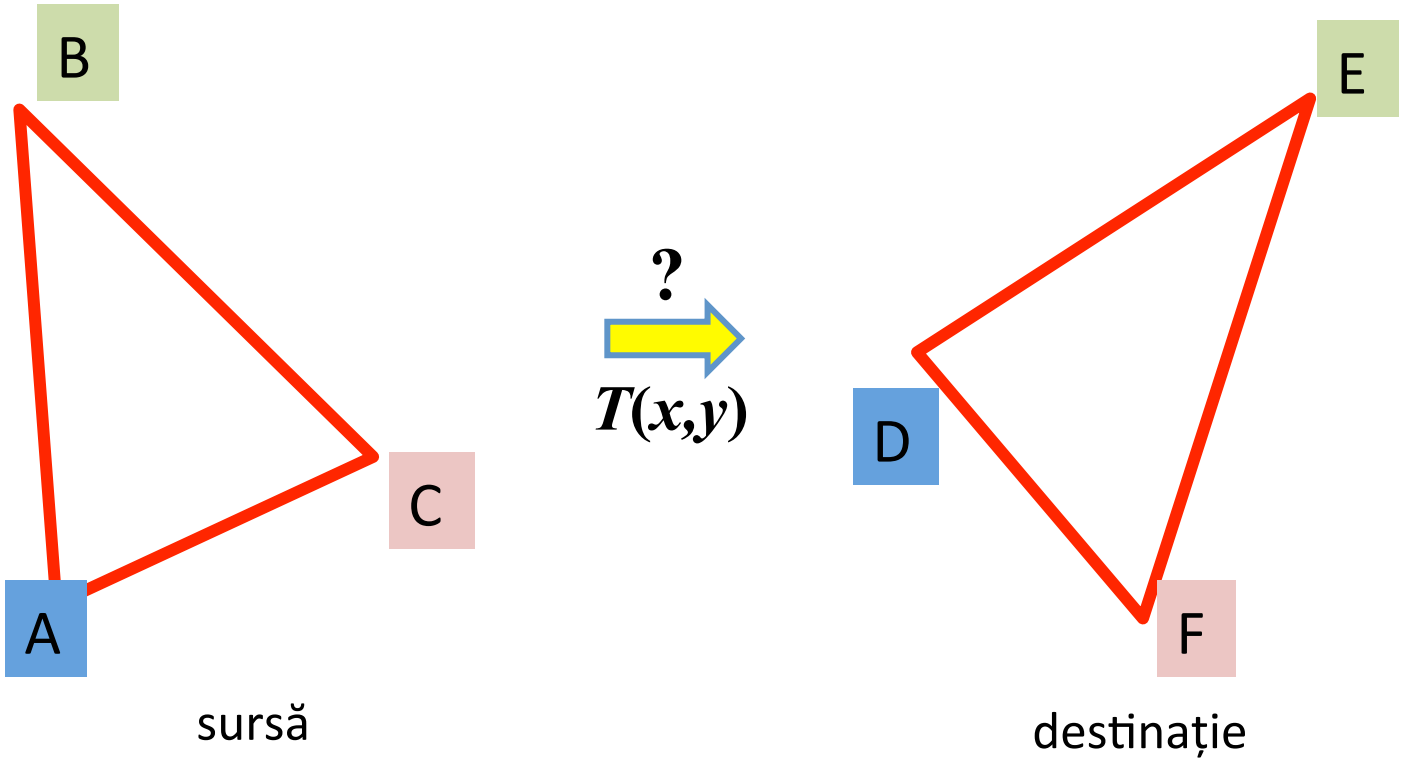
Trei perechi de puncte corespondente sunt suficiente pentru a găsi transformarea afină (6 grade de libertate).

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$$

Transformări afine aplicate triunghiurilor

Pentru două triunghiuri ABC și DEF pot găsi transformarea afină care îmi transformă A în D, B în E și C în F rezolvând un sistem de 6 ecuații cu 6 necunoscute



Transformări afine aplicate triunghiurilor

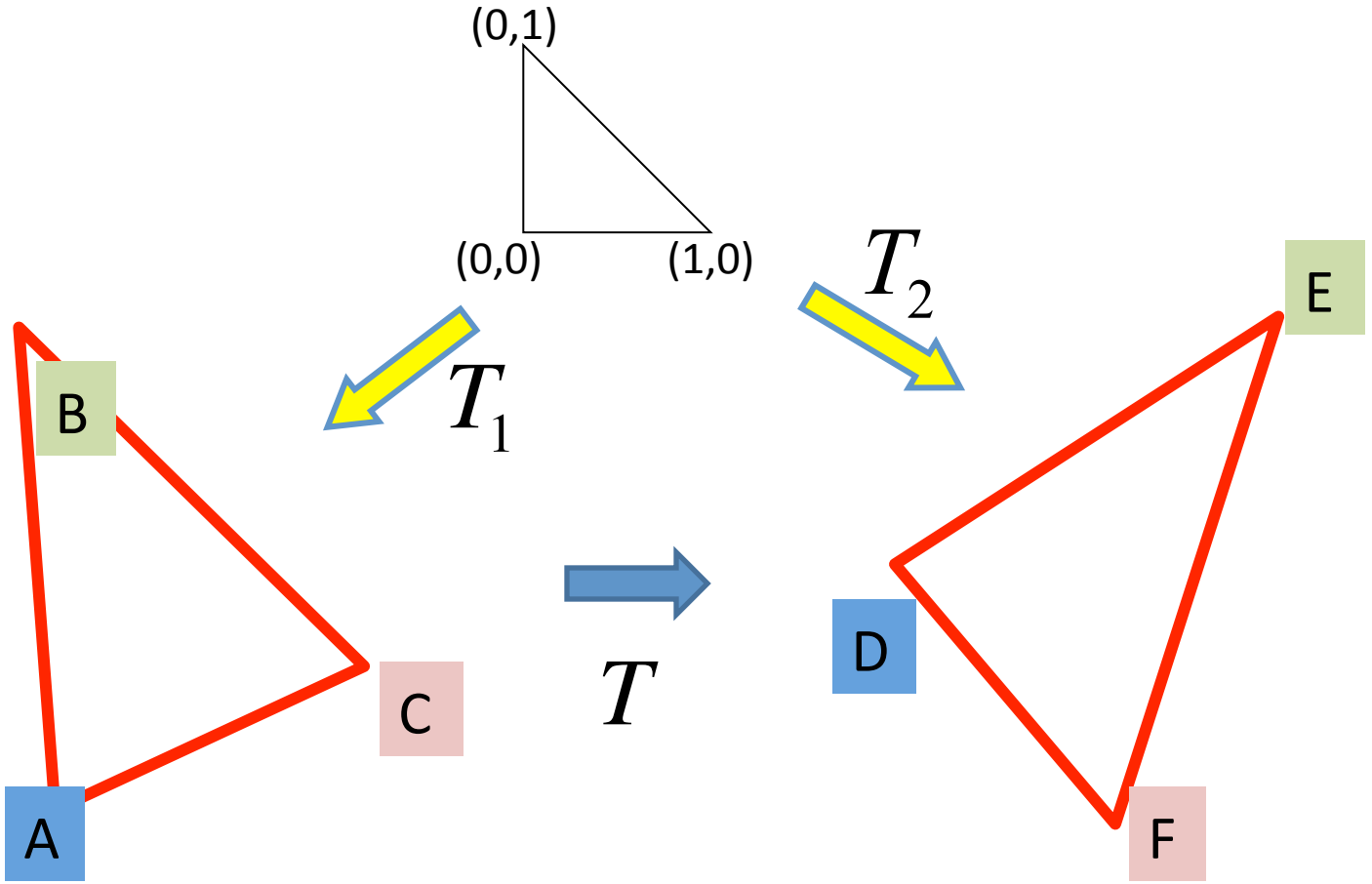
Pentru două triunghiuri ABC și DEF pot găsi transformarea afină care îmi transformă A în D, B în E și C în F rezolvând un sistem de 6 ecuații cu 6 necunoscute

$$\begin{bmatrix} x_D \\ y_D \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_A \\ y_A \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x_F \\ y_F \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_C \\ y_C \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_E \\ y_E \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ 1 \end{bmatrix}$$

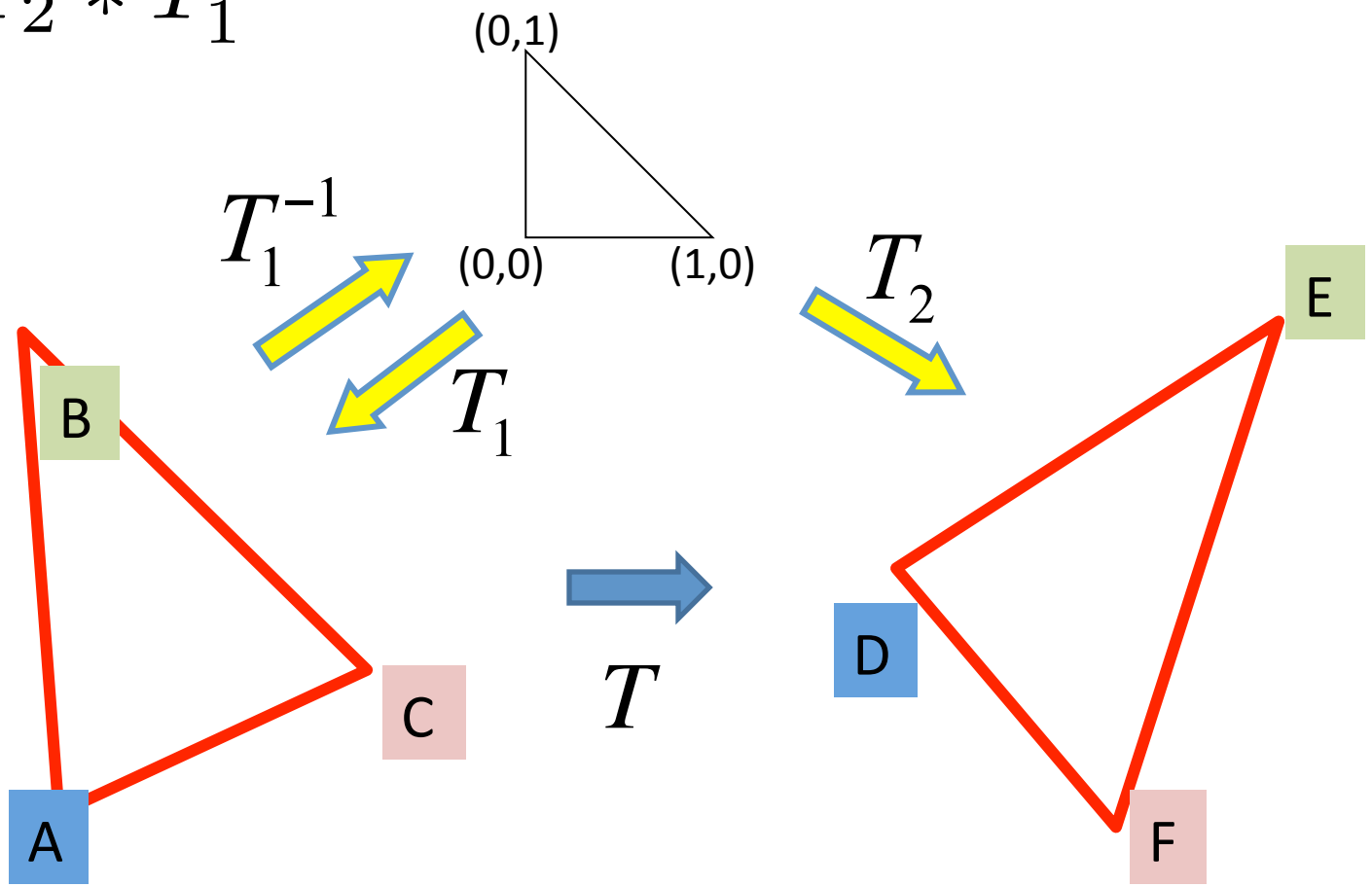
Abordare echivalentă: compunere de transformări afine

Folosesc un triunghi cu coordonate “convenabile” (sistemul de 6 ecuații și 6 necunoscute este ușor de rezolvat) pentru a calcula T_2, T_1. Pe baza lor aflu T.



Abordare echivalentă: compunere de transformări afine

$$T = T_2 * T_1^{-1}$$



Exemplu: Calculul lui T_2

Sistem de 6 ecuații cu 6 necunoscute, mult mai simplu:

$$\begin{bmatrix} x_D \\ y_D \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x_F \\ y_F \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

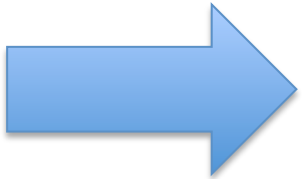
$$\begin{bmatrix} x_E \\ y_E \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Exemplu: Calculul lui T_2

Sistem de 6 ecuații cu 6 necunoscute, mult mai simplu:

$$\begin{bmatrix} x_D \\ y_D \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x_F \\ y_F \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

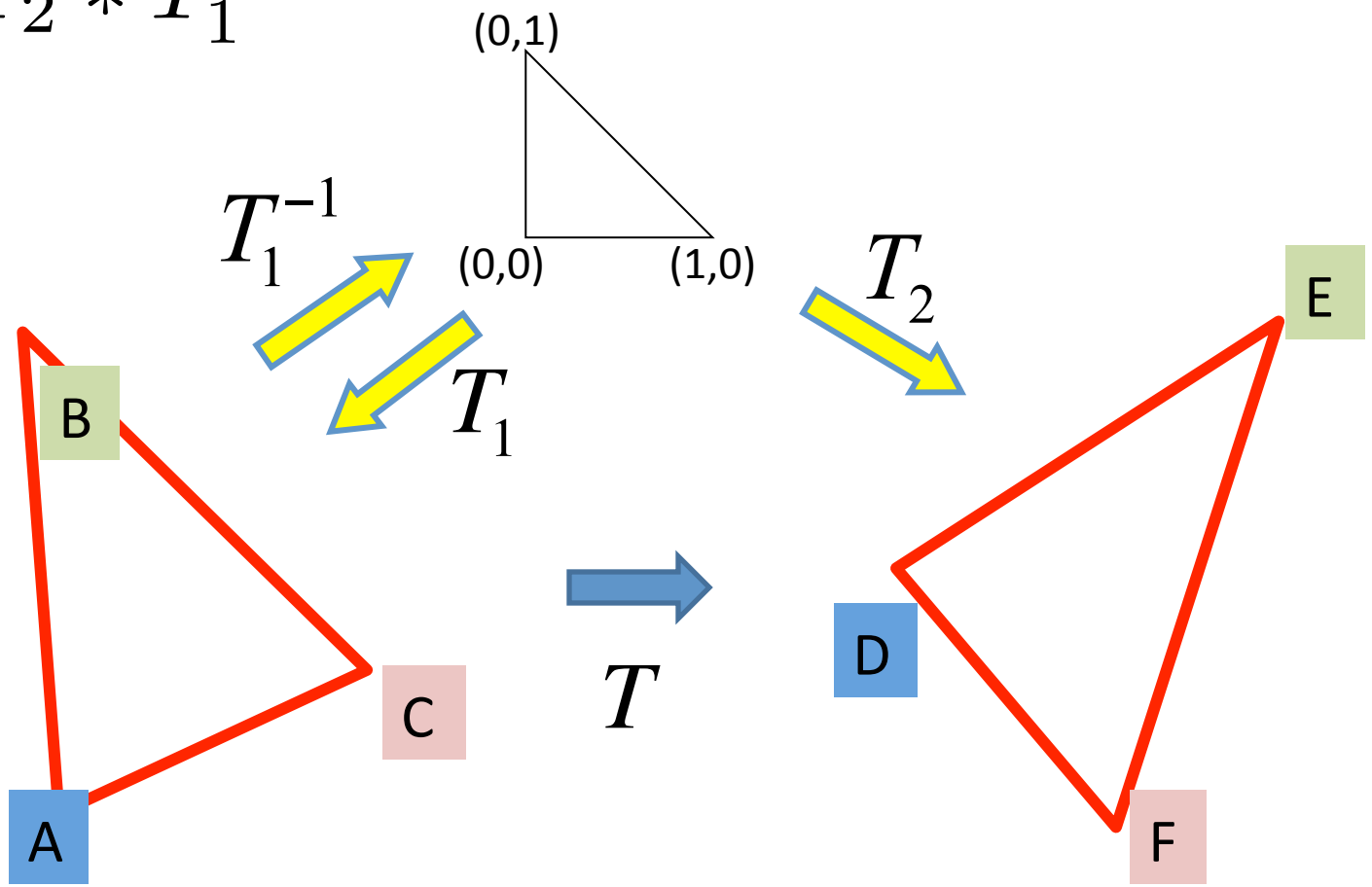
$$\begin{bmatrix} x_E \\ y_E \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$



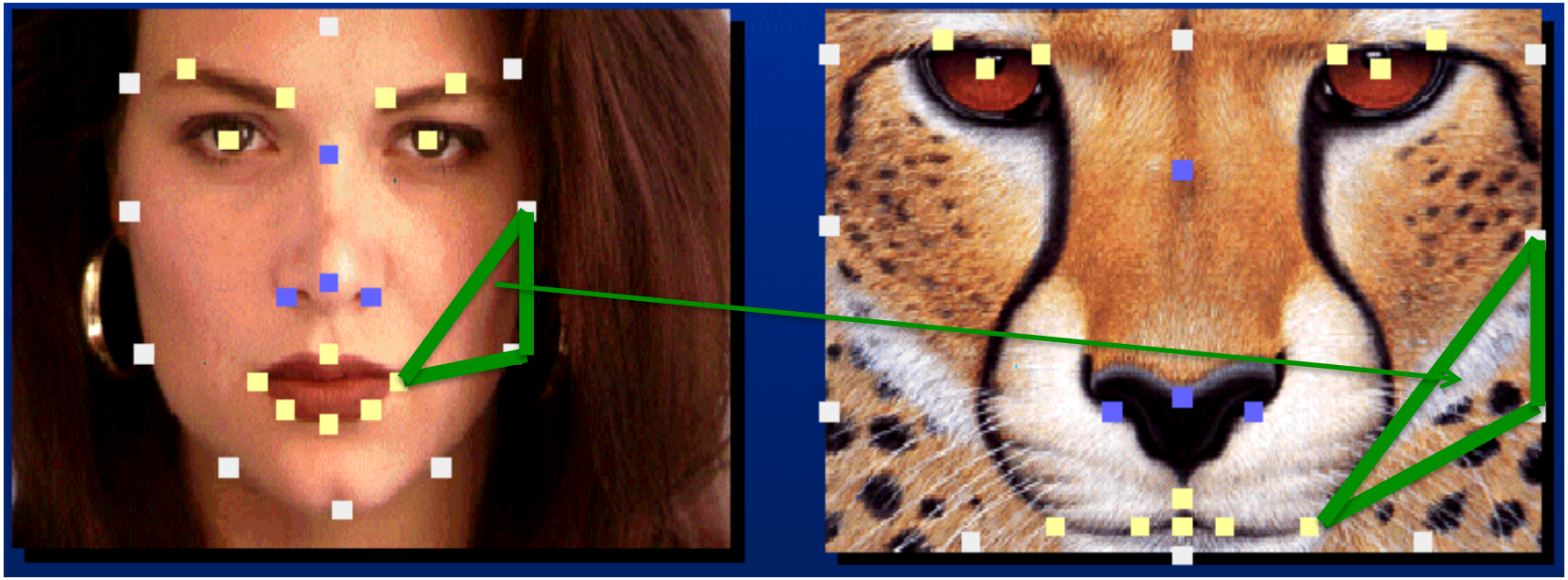
$$c = x_D$$
$$f = y_D$$
$$a = x_F - x_D$$
$$d = y_F - y_D$$
$$b = x_E - x_D$$
$$e = y_E - y_D$$

Abordare echivalentă: compunere de transformări afine

$$T = T_2 * T_1^{-1}$$

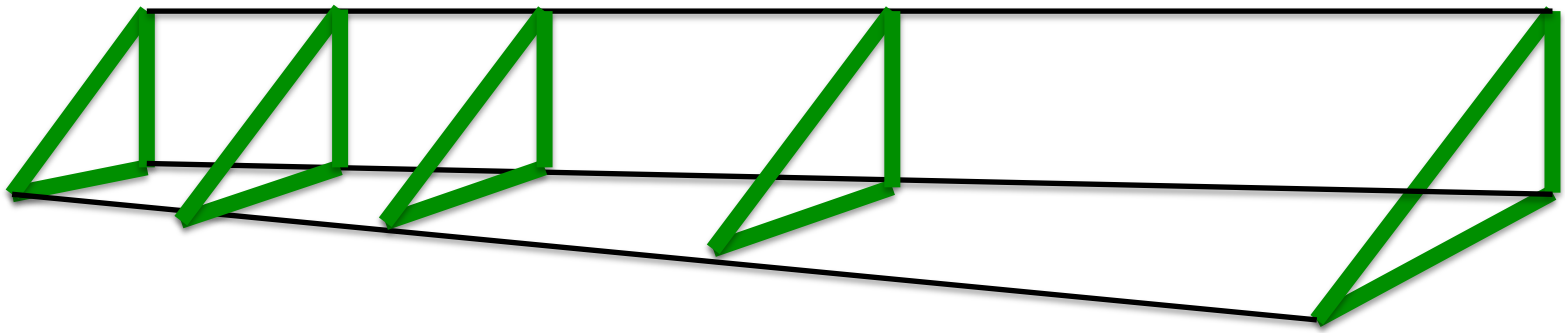


Tranziția geometrică și fotometrică între două triunghiuri



- fiecare regiune (triunghi) din imaginea sursă va fi transformată geometric + fotometric pentru a ajunge la regiunea corespondentă (triunghiul corespondent) din imaginea țintă

Tranziția geometrică și fotometrică între două triunghiuri



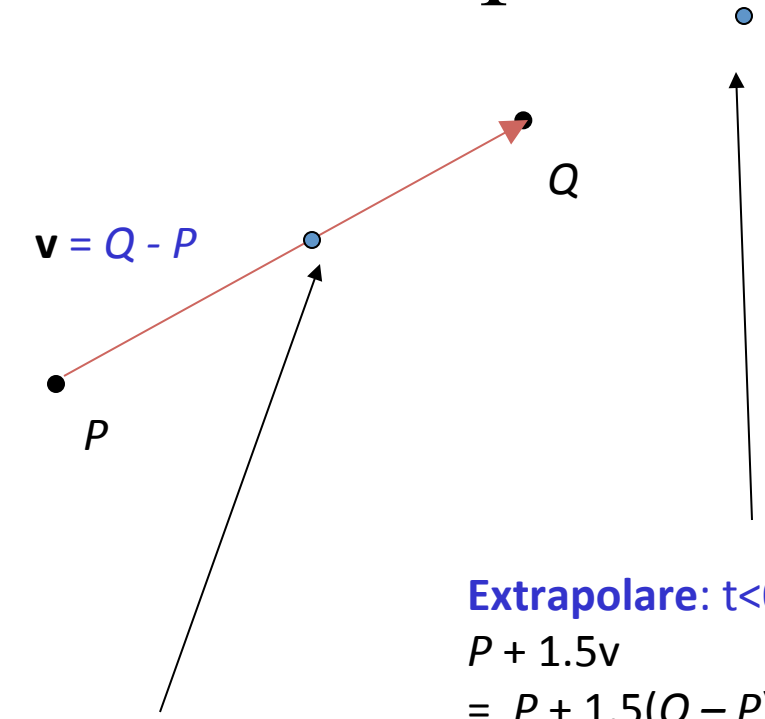
Realizează o tranziție treptată, în timp, de la triunghiul sursă T_0 la triunghiul țintă T_1 . La pasul t:

- geometric: $T_t = (1-t)*T_0 + t*T_1$ (“medie ponderată” a lui T_0 și T_1)
- fotometric: pentru orice pixel (x,y) din triunghiul T_t găsește pixelii corespondenți (x_0,y_0) din T_0 și (x_1,y_1) T_1 (pe baza trasformării afine găsite).

$RGB_{(x,y)} = (1-t)*RGB_{(x0,y0)} + t*RGB_{(x1,y1)}$ (“medie ponderată” a culorilor RGB din T_0 și T_1)

Media ponderată a două puncte

Care este media punctelor P și Q?



Extrapolare: t<0 or t>1

$P + 1.5v$

$= P + 1.5(Q - P)$

$= -0.5P + 1.5\ Q\ (t=1.5)$

$P + 0.5v$

$= P + 0.5(Q - P)$

$= 0.5P + 0.5\ Q$

Interpolare liniară

Obținem un punct nou:

$(1-t)P + tQ,\ 0<t<1$

- P și Q pot fi orice:
  - puncte în plan (2D) = varfurile corespondente ale triunghiurilor
  - Culori RGB

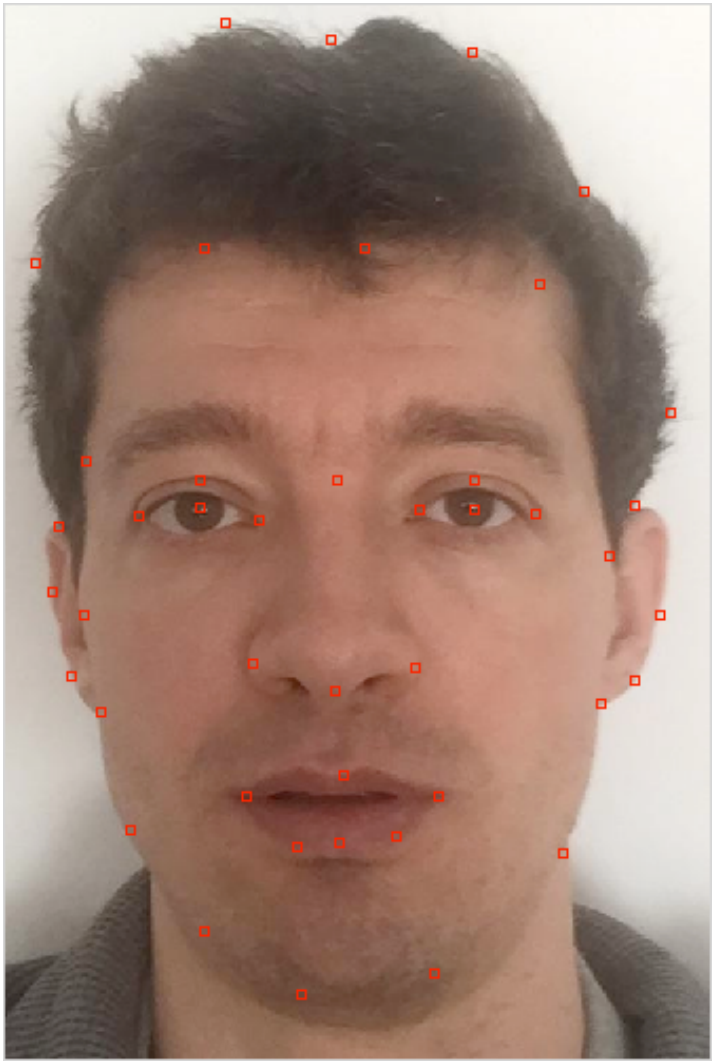
Algoritm pentru transformarea obiectelor

1. Obținem perechi de puncte corespondente de pe cele două obiecte (imagini) – ideal se obține automat

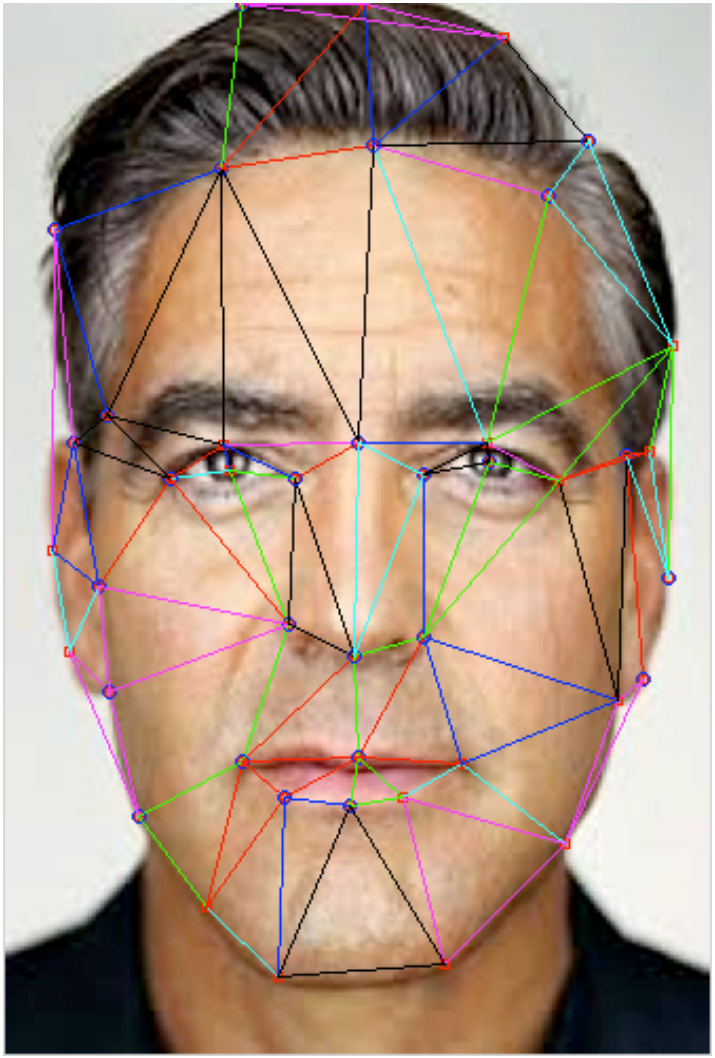
2. Obține o triangulare a setului de puncte inițial într-o imagine (folosește aceeași triangulare pentru ambele imagini, funcția delaunay)

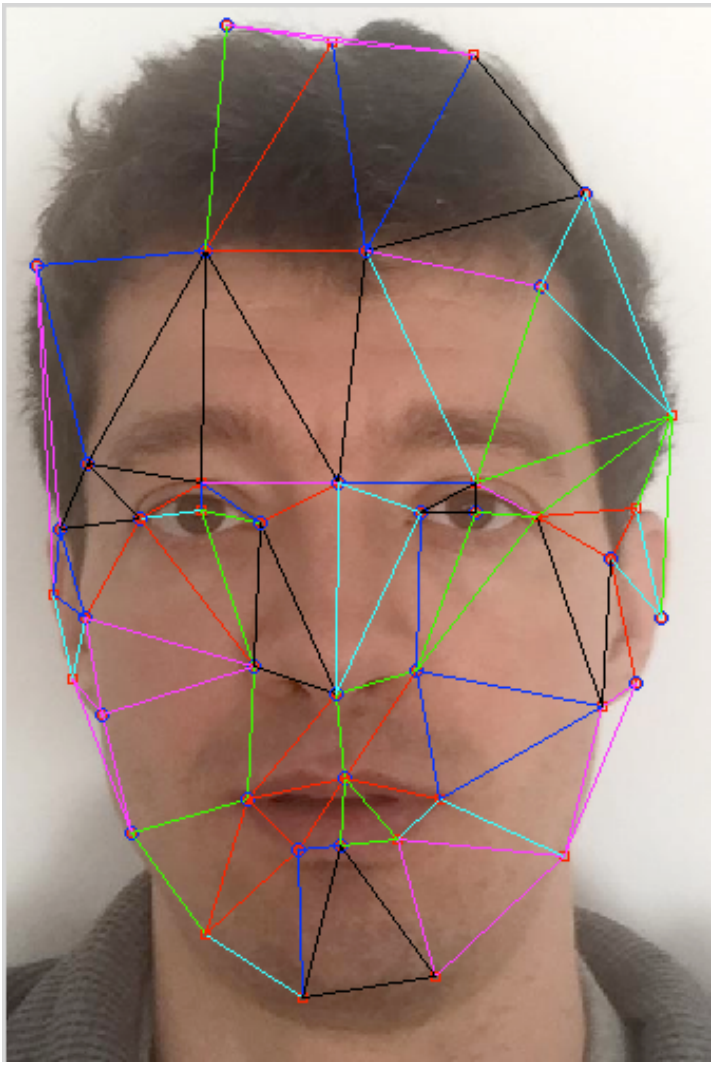
3. Pentru t = 0:1/N:1 (N+1 imagini de tranziție)
   a. calculează fiecare triunghi mediu ponderat pe baza punctelor corespondente
   b. pentru fiecare triunghi mediu
      - calculeaza transformare afina dintre el si triunghiurile corespondente
      - pentru fiecal pixel in triunghi, calculeaza punctele corespondente din fiecare imagine si determina media ponderata a culorilor
   c. Salveaza imaginea intermediara (in total N+1)

Plotarea punctelor corespondente



Plotarea triangulărilor Delaunay

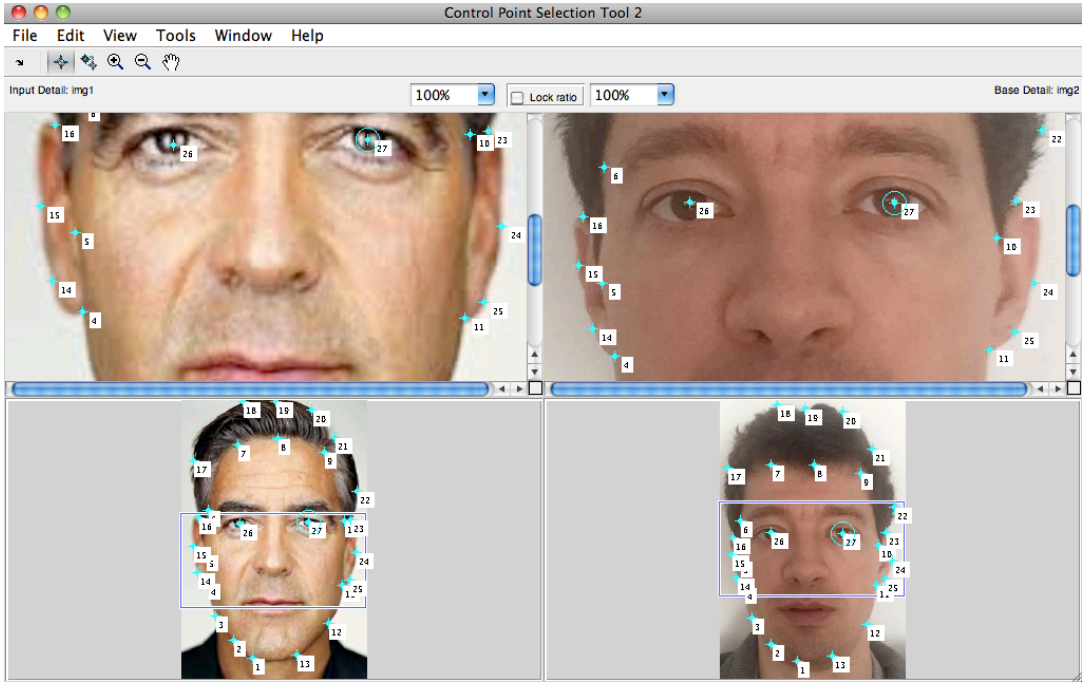




Transformarea unei fețe în altă față

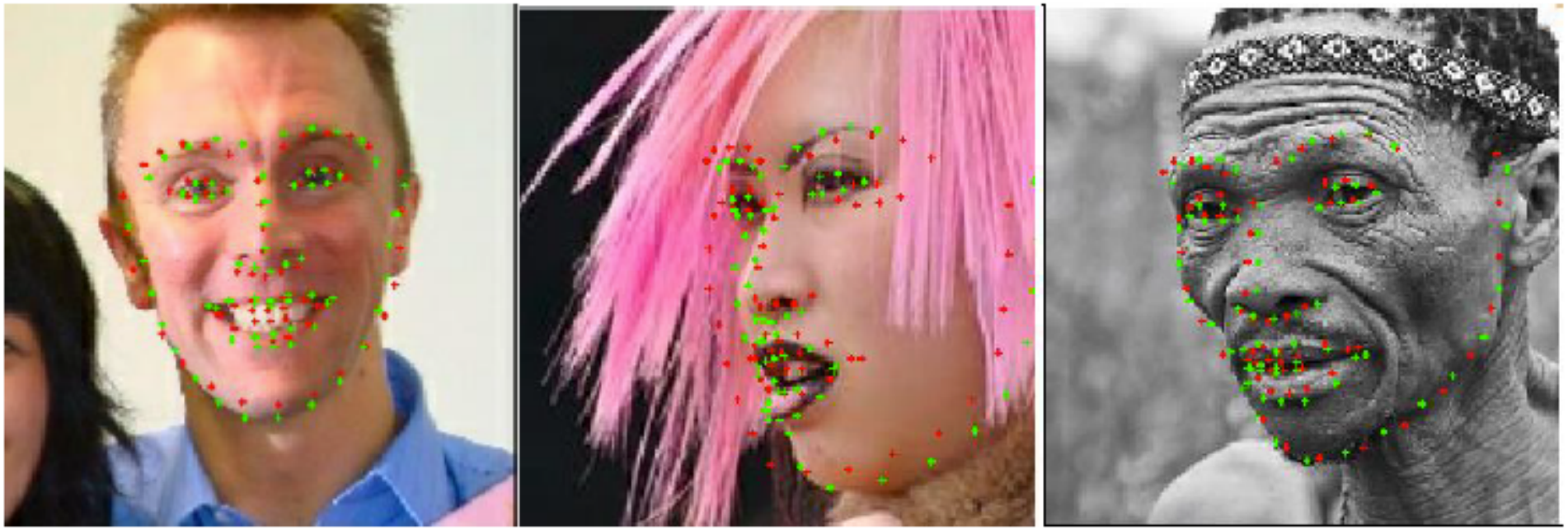
Găsirea de puncte corespondente din cele două obiecte

1. Manual (plicitisitor, consumator de timp)



Găsirea de puncte corespondente din cele două obiecte

1. Manual (plicitisitor, consumator de timp)
2. Automat, prin algoritmi de învățare automată



Verde – puncte faciale de interes (facial landmarks) adnotate
Roșu – puncte faciale de interes (facial landmarks) regresate

300 Faces in-the-Wild Challenge: The first facial landmark localization Challenge

Christos Sagonas[1], Georgios Tzimiropoulos[1,2], Stefanos Zafeiriou[1] and Maja Pantic[1,3]

Template de 68 de puncte

300 Faces in-the-Wild Challenge: The first facial landmark localization Challenge

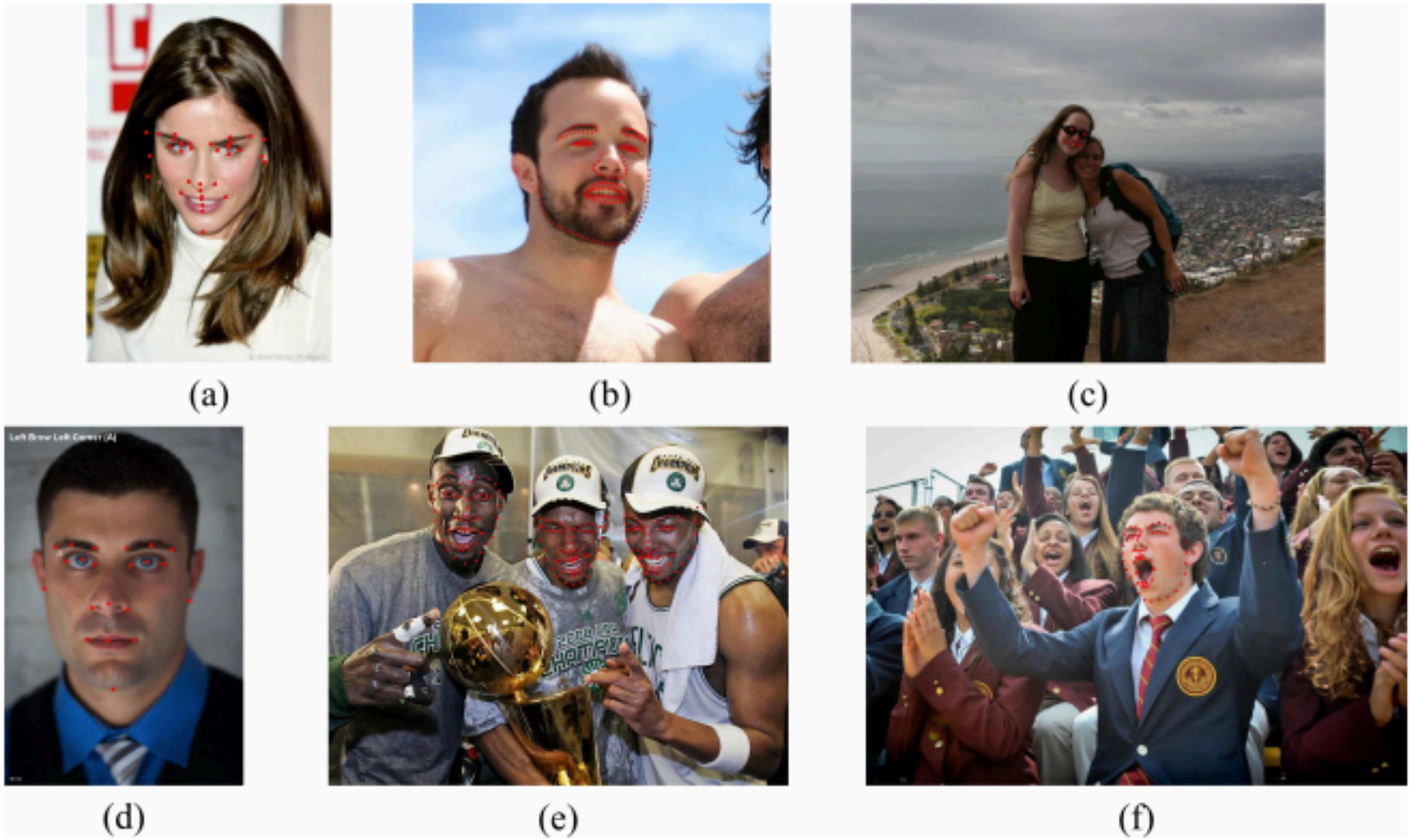


Figure 1. Annotated images from (a) LFPW, (b) HELEN, (c) AFW, (d) AFLW, (e) 300-W 'Indoor' and (f) 300-W 'Outdoor'.

Seturi de date cu puncte faciale de interes adnotate

1. 300 Faces in the wild challenge
2. The Annotated Faces in the Wild
3. The Caltech Occluded Faces în the Wild
4. Face Recognition Grand Challenge v2
5. HELEN
6. iBUG
7. The Labeled Face Parts în the Wild
8. MENPO
9. XM2VTS

Algoritmi de învățare pentru regresia punctelor de interes faciale

1. Cascade de arbori de regresie (implementat în DLIB)

2. Rețele neuronale convoluționale pentru regresie

One Millisecond Face Alignment with an Ensemble of Regression Trees

Vahid Kazemi and Josephine Sullivan
KTH, Royal Institute of Technology
Computer Vision and Active Perception Lab
Teknikringen 14, Stockholm, Sweden
{vahidk,sullivan}@csc.kth.se

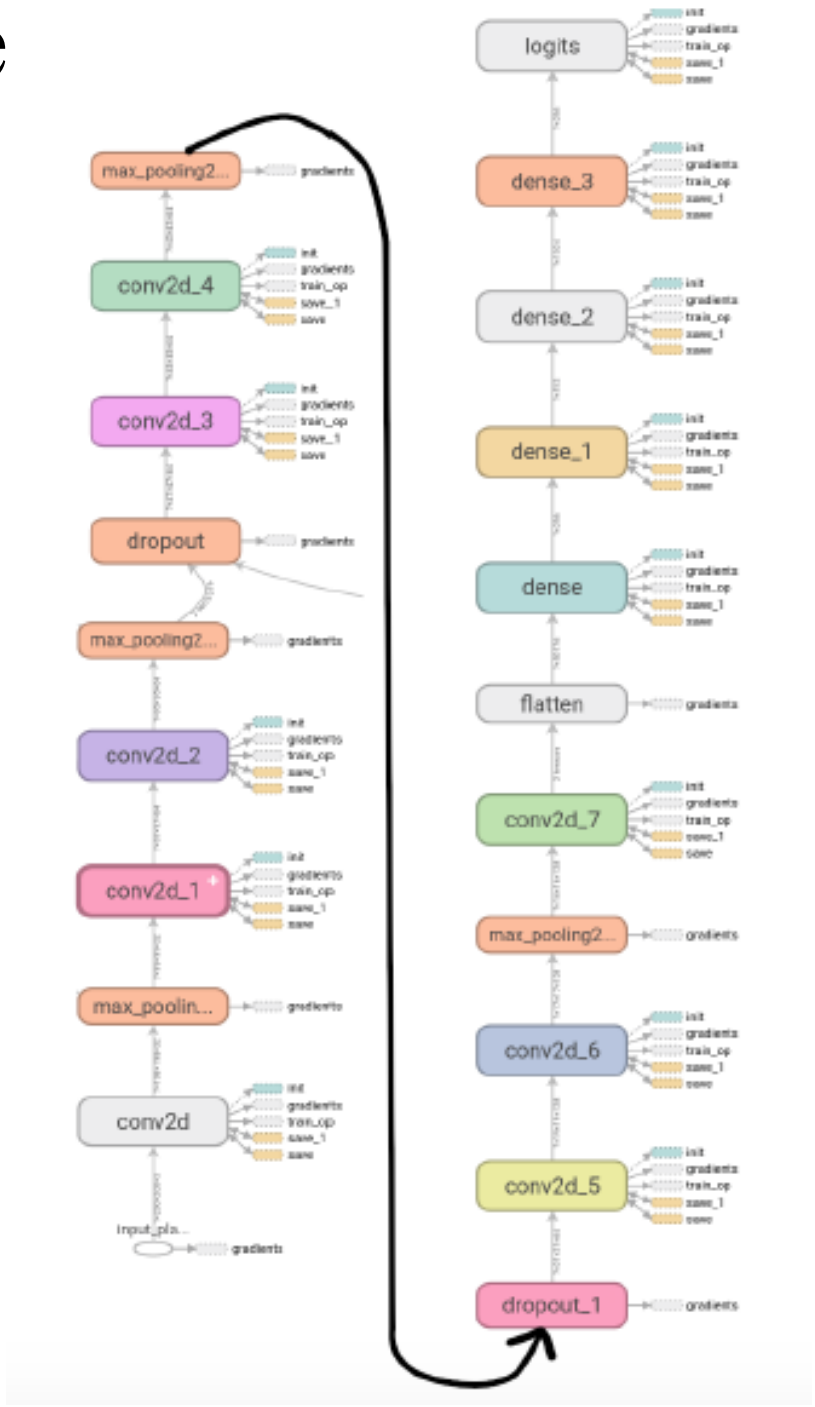
Abstract

This paper addresses the problem of Face Alignment for a single image. We show how an ensemble of regression trees can be used to estimate the face's landmark positions directly from a sparse subset of pixel intensities, achieving super-realtime performance with high quality predictions. We present a general framework based on gradient boosting for learning an ensemble of regression trees that optimizes the sum of square error loss and naturally handles missing or partially labelled data. We show how using appropriate priors exploiting the structure of image data helps with efficient feature selection. Different regularization strategies and its importance to combat overfitting are also investigated. In addition, we analyse the effect of the quantity of training data on the accuracy of the predictions and explore the effect of data augmentation using synthesized data.

Figure 1. Selected results on the HELEN dataset. An ensemble of randomized regression trees is used to detect 194 landmarks on face from a single image in a millisecond.

1. Introduction

Rețele convoluționale pentru regresie



Algoritm pentru transformarea obiectelor

1. Obținem perechi de puncte corespondente de pe cele două obiecte (imagini) – ideal se obține automat

2. Obține o triangulare a setului de puncte inițial într-o imagine (folosește aceeași triangulare pentru ambele imagini, funcția delaunay)

3. Pentru t = 0:1/N:1 (N+1 imagini de tranziție)
   a. calculează fiecare triunghi mediu ponderat pe baza punctelor corespondente
   b. pentru fiecare triunghi mediu
      - calculeaza transformare afina dintre el si triunghiurile corespondente
      - pentru fiecal pixel in triunghi, calculeaza punctele corespondente din fiecare imagine si determina media ponderata a culorilor
   c. Salveaza imaginea intermediara (in total N+1)

Transformarea unei fețe în altă față

Animație GIF

Verificarea identității

Verificare identității prin recunoașterea de acțiuni simple

- Multe companii (Revolut, Uber, bănci, etc) efetuează o verificare online a identității pe baza unor fotografii cu documente + selfie

- Măsură suplimentară: video cu persoana realizând o acțiune simplă (întoarce capul în partea stângă/dreaptă, zâmbește, clipește din ochi)

Verificare identității prin recunoașterea de acțiuni simple

- Măsură suplimentară: video cu persoana realizând o acțiune simplă (întoarce capul în partea stângă/dreaptă, zâmbește, clipește din ochi)

- Clasifică asemenea acțiuni (4 clase) folosind euristici pe baza punctelor facile de interes regresate din video scut (5 secunde)

Verificare identității prin recunoașterea de acțiuni simple

Cerințe tema 3

1. GIF cu transformarea unei fețe în altă față folosind
   - adnotări manuale (3 puncte)
   - puncte faciale de interes regresate automat (2 puncte)

2. Clasificarea de video-uri (4 puncte) într-un din cele 4 clase

Oficiu – 1 punct

BONUS: antrenare regresor propriu (rețea convoluțională) și comparație cu DLIB - 2 puncte

DLIB

Not Secure | dlib.net/face_landmark_detection.py.html

```
#!/usr/bin/python
# The contents of this file are in the public domain. See LICENSE_FOR_EXAMPLE_PROGRAMS.txt
#
#   This example program shows how to find frontal human faces in an image and
#   estimate their pose.  The pose takes the form of 68 landmarks.  These are
#   points on the face such as the corners of the mouth, along the eyebrows, on
#   the eyes, and so forth.
#
#   The face detector we use is made using the classic Histogram of Oriented
#   Gradients (HOG) feature combined with a linear classifier, an image pyramid,
#   and sliding window detection scheme.  The pose estimator was created by
#   using dlib's implementation of the paper:
#      One Millisecond Face Alignment with an Ensemble of Regression Trees by
#      Vahid Kazemi and Josephine Sullivan, CVPR 2014
#   and was trained on the iBUG 300-W face landmark dataset (see
#   https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/):
#      C. Sagonas, E. Antonakos, G, Tzimiropoulos, S. Zafeiriou, M. Pantic.
#      300 faces In-the-wild challenge: Database and results.
#      Image and Vision Computing (IMAVIS), Special Issue on Facial Landmark Localisation "In-The-Wild". 2016.
#   You can get the trained model file from:
#   http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2.
#   Note that the license for the iBUG 300-W dataset excludes commercial use.
#   So you should contact Imperial College London to find out if it's OK for
#   you to use this model file in a commercial product.
#
#
#   Also, note that you can train your own models using dlib's machine learning
```

Evaluarea în sesiune

- Evaluarea temelor are termen limită: joi, 6 februarie

- *Rugăminte: toți studenții de la 507 care vor să prezinte teme să îmi dea un email până pe 31 ianuarie astfel încât să stabilesc cu fiecare data prezentării*

- Examen scris: vineri, 7 februarie, ora 9:00, fără materiale, 2 ore, greu ☹