A Face Detection and Facial Expression Recognition Method

Dr. Nikolaos Bourbakis
ATRC
Wright State University
AIIS

I would like to thank the ICSDT-10 Organizers for the very successful meeting, but especially I would like to thank Professors Tsigritzis and Virvou for inviting me to share some of my views on Face & Facial Expressions with you

Outline

- **Introduction**
  - What-why-how
  - Related work
  - Proposed methodology
- **Skin color detection**
- **Face Detection**
  - FRG Segmentation
  - Skin Region Synthesis
  - LG Graph
- **Facial Expression Detection**
- **Applications**
- **Conclusions**

Introduction

- **What-why-how**
- **Face detection**
- **Related work (a brief overview)**
- **Motivation / Open issues**
- **Proposed methodology overview**

What-Why-How

■ What

A method for detecting and recognizing facial expressions in real environment

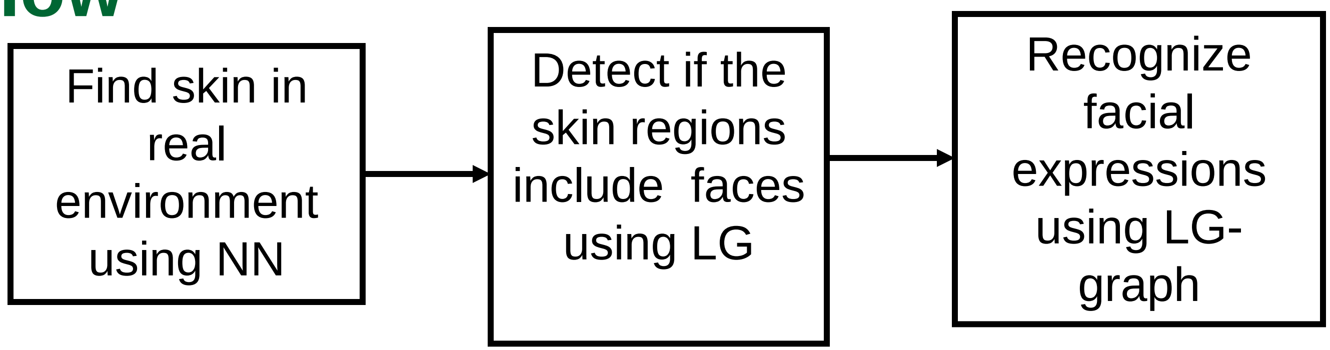
■ Why

Understanding Human Behavior and Emotions (HHI-HMI)

Security

Surveillance

■ How



Face Detection

- **Goal**
  - <u>Detect</u> and <u>locate</u> the face present in the image regardless of
    - Illuminations
    - Background
    - Occlusions
    - Facial pose, orientation and expressions

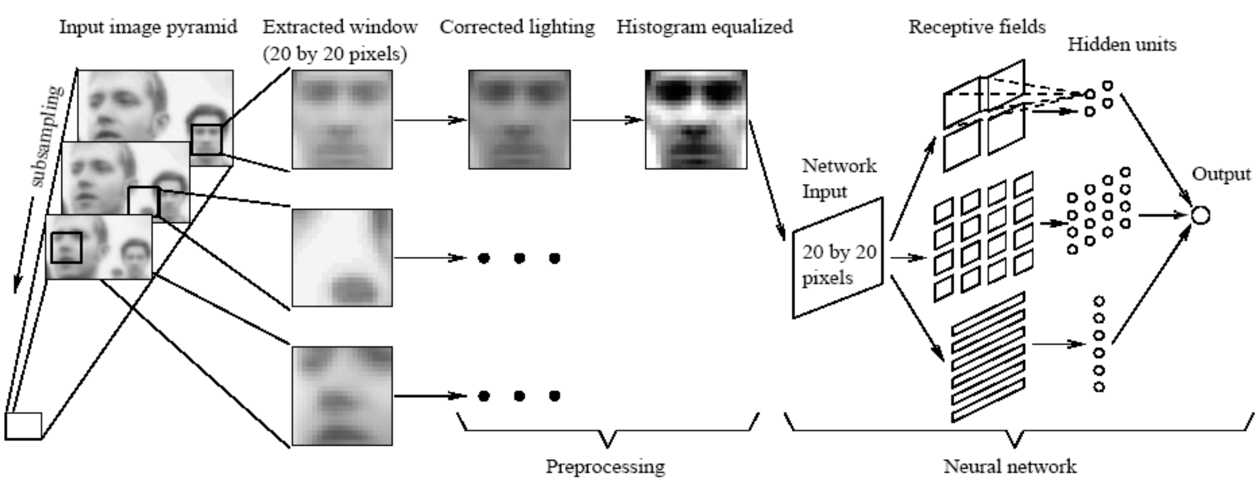
- **Why is face detection important?**
  - Primary step in
    - Face expression detection
    - Vision-based Intelligent HCI Systems
    - Secure Identification and Authentication

Appearance based approaches

- **Approach**
  - Face is recognized as a whole
  - Face and non-face patterns are learned

- **Limitations**
  - Accurate in frontal images with simple background and in well-illuminations

- **Example - NN Face Detector [Rowley et al., 96, 98]**

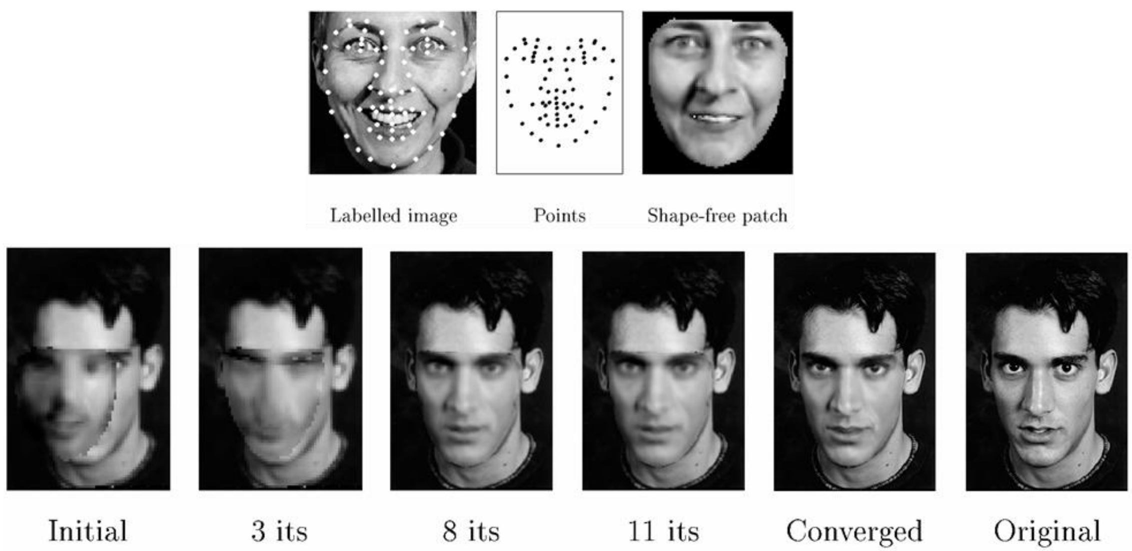


Template based approaches

- **Approach**
  - A standard face pattern is pre-defined
  - Use correlation to locate faces

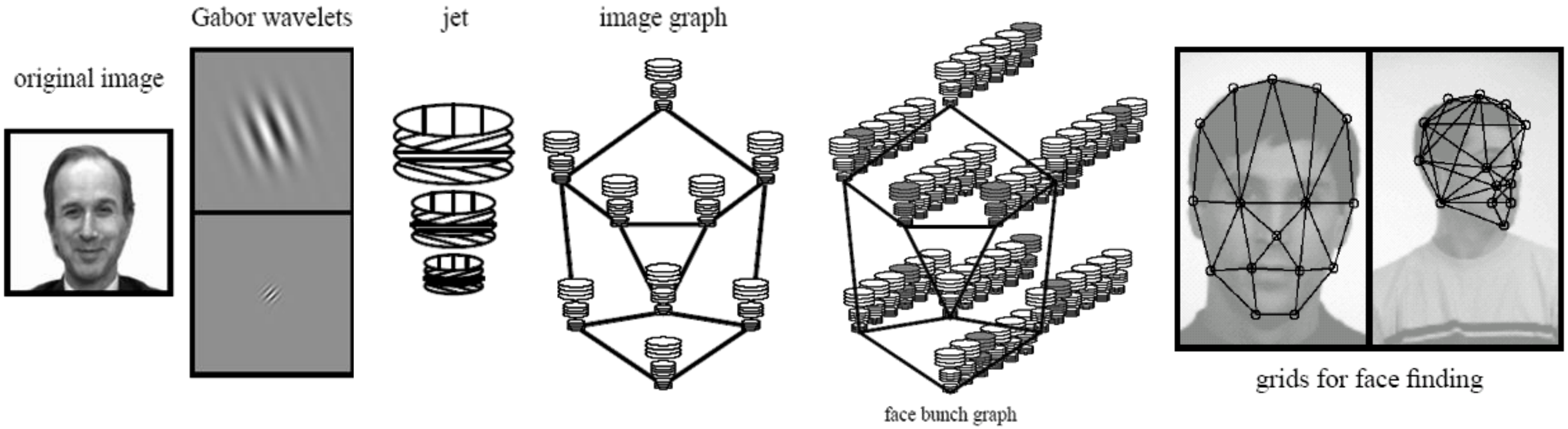
- **Limitations**
  - Difficult to extend to various poses, shapes and scale

- **Example - Active Shape / Appearance Model [Cootes & Taylor, 01]**



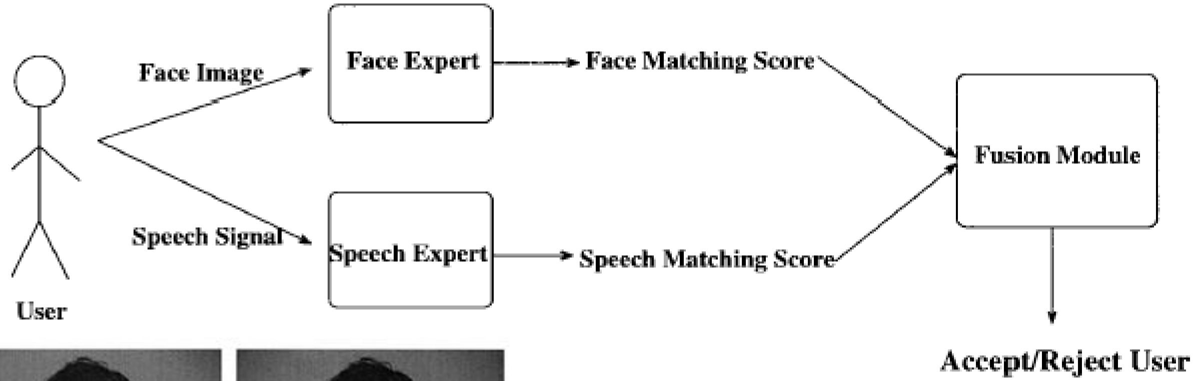
Feature based approaches

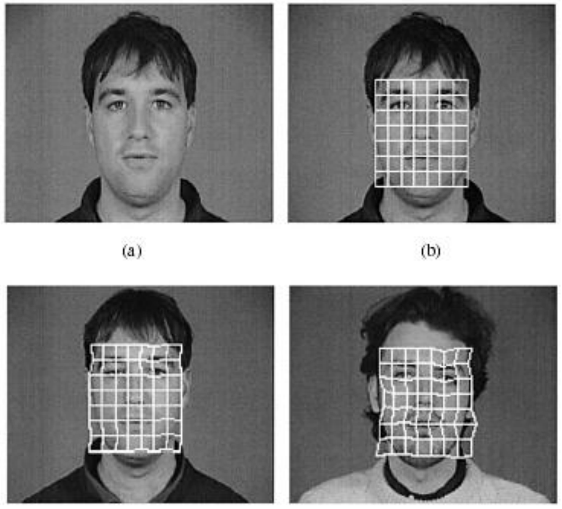
- **Approach**
  - Detect invariant facial features
  - Group features into candidate faces and verify them
- **Limitations**
  - Difficult to locate features in complex backgrounds and in various illuminations
- **Example - Elastic Bunch Graph [Wiskott et al., 99]**



Multiple Modalities Face-Detection

.Perceptron: Better performance than single modalities (Yacoub, et. al. IEEE T-NN, 1999)





(a) (b)

(c) (d)

TABLE I
PERFORMANCE OF SINGLE MODALITIES ON TEST SETS

	Conf I			Conf II		
Modality	FA	FR	EER	FA	FR	EER
Face (EGM)	8.08	8.5	8.4	7.67	7.25	7.31
Voice (Spher.)	1.6	5.00	3.25	5.53	4.25	4.75
Voice (HMM)	0.00	1.48	0.75	-	-	-

Open issues

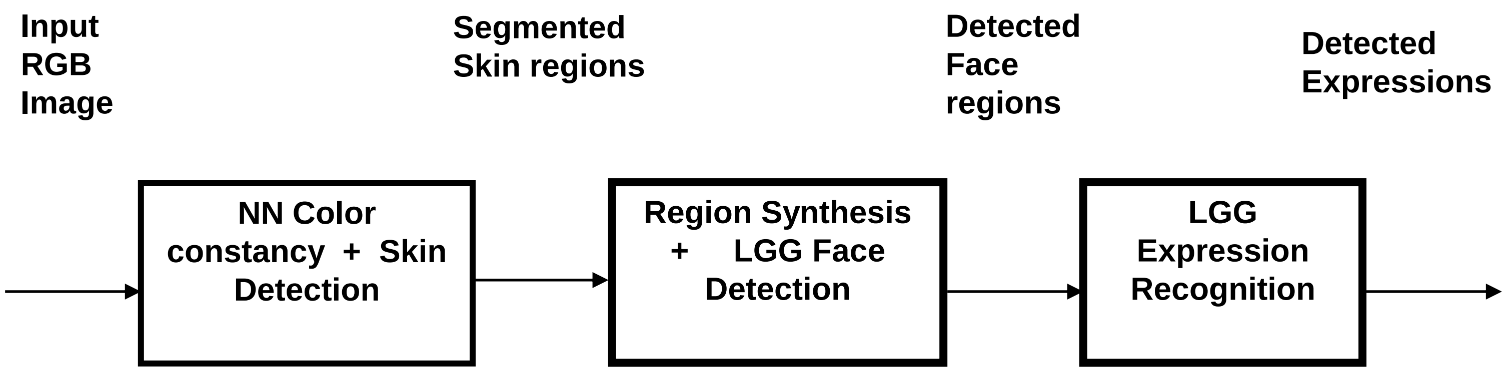
- **Current face detection methods [FRVT 2003]**
  - Low performance
    - Outdoor images with uncontrolled illuminations
    - Complex backgrounds
    - Non-frontal views

- **Realistic, unavoidable conditions!**

* **Need a robust method to satisfactorily detect faces in**
  * varying illuminations
  * complex backgrounds

The Local-Global Graph Approach

- **Overview**
  - Skin detection
  - L-G graph Face detection
  - L-G graph Expression detection



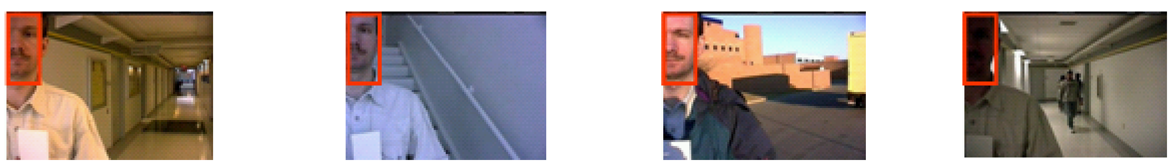
Skin-color Adaptation

- **ANN method for skin-color adaptation**
- **Skin detection results**

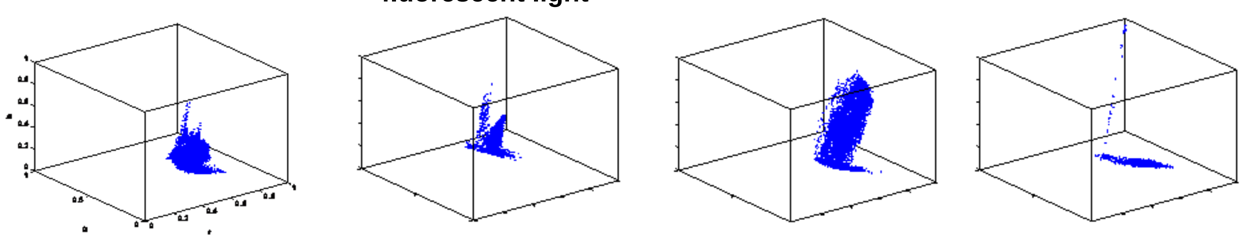
Skin-color

- **Skin-color**
  - Robust against rotations, scaling and partial occlusions
  - Challenges - ethnicity, Illuminations, background, makeup, motion

- **Effect of Illumination**
  - Skin-color varies significantly with different illuminations
  - Humans can dynamically adapt to these illumination changes - <u>color constancy</u>



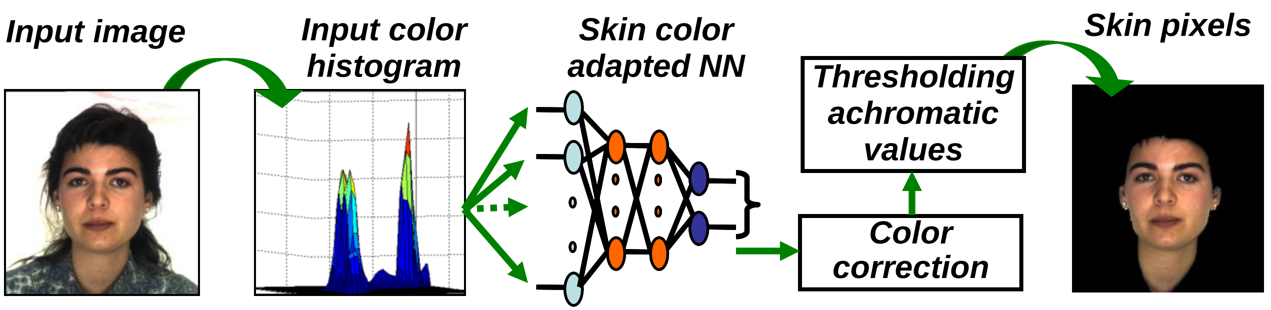
(a) indoor + neon light **(b) natural + fluorescent light** **(c) outdoor + shadow** **(d) indoor + shadow**



Skin color clusters in normalized *rg* space

Skin detection approach

- **Overview [Kakumanu-Bourbakis, *ICTAI-04*]**
  - Apply color correction
  - Estimate the illuminant
    - Train a neural network to estimate the skin color illuminant
    - Apply color correction in LMS cone space
  - Detect skin as the achromatic region

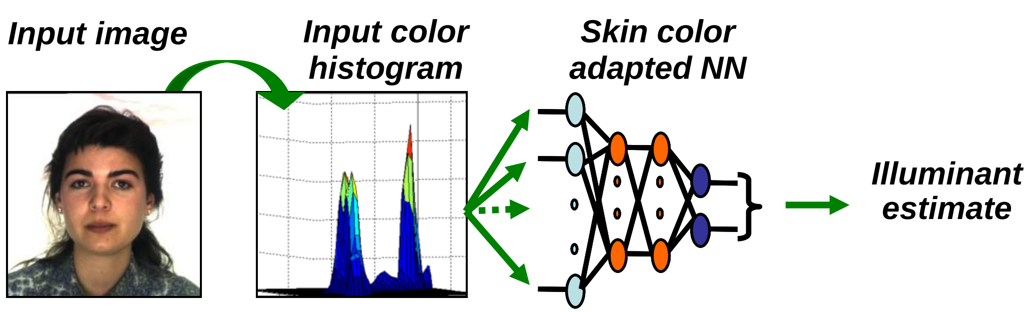


Skin detection overview

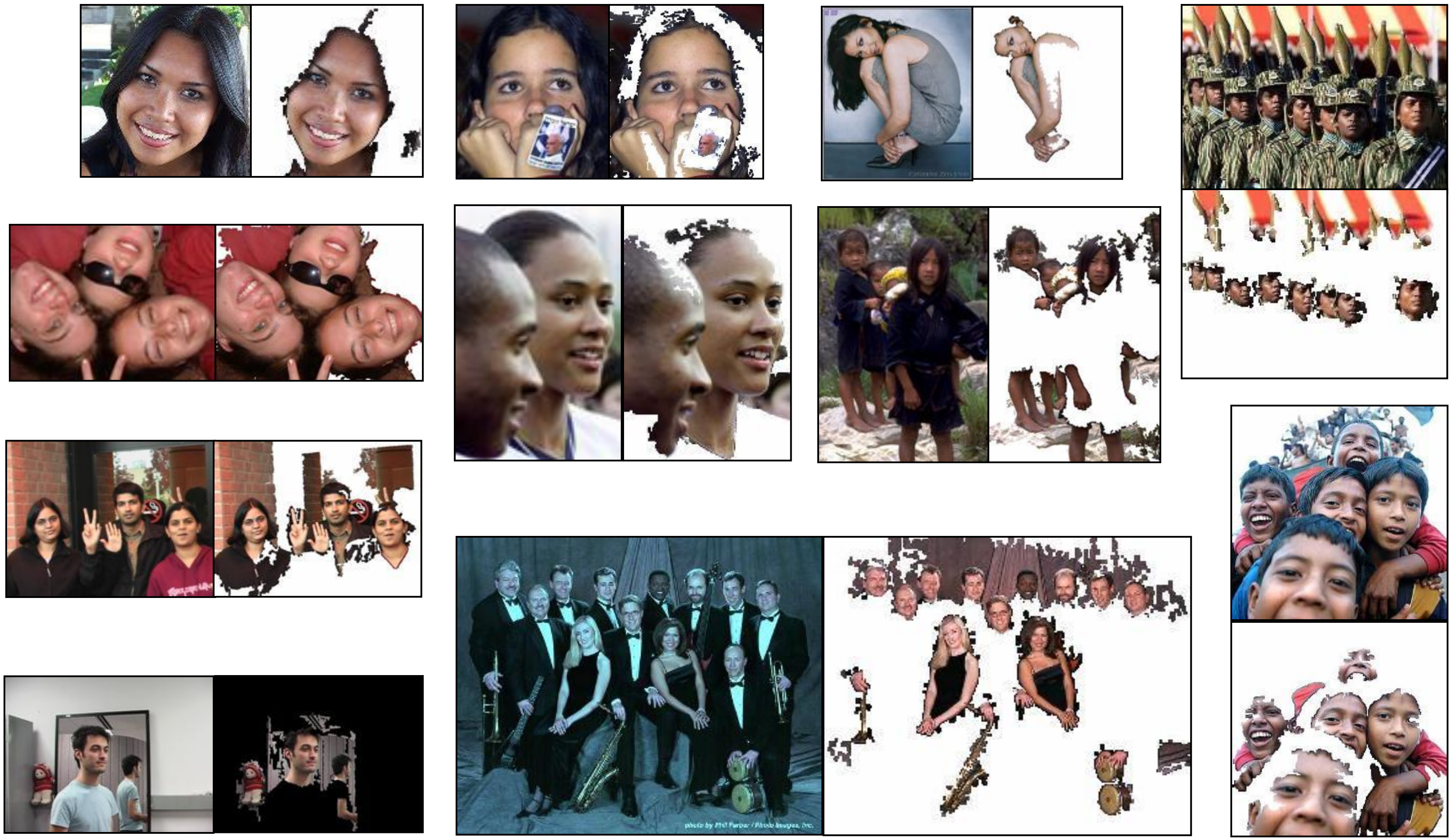
ANN for skin color adaptation

Multi-layer perceptron network

- Three layered - *1600 * 48 * 8 * 2* [Cardei, 00]
- Input
  - Input *rg* space is divided into *40 * 40* discrete bins
  - Input to the neuron is *1* or *0*
- Output
  - Expected *rg* chromaticity of the image illuminant



Skin detection results



Images from UCD Database [Sharma & Reilly, 2003]

+ Images collected at WSU

L-G Graph Face Detection

- **FRG Segmentation (**smoothing, edge detection segmentation)
- **Facial feature region representation**
- **Region synthesis**
- **Face LG Graph**
- **LG Graph matching**
- **Face detection results**

Face Detection: Smoothing

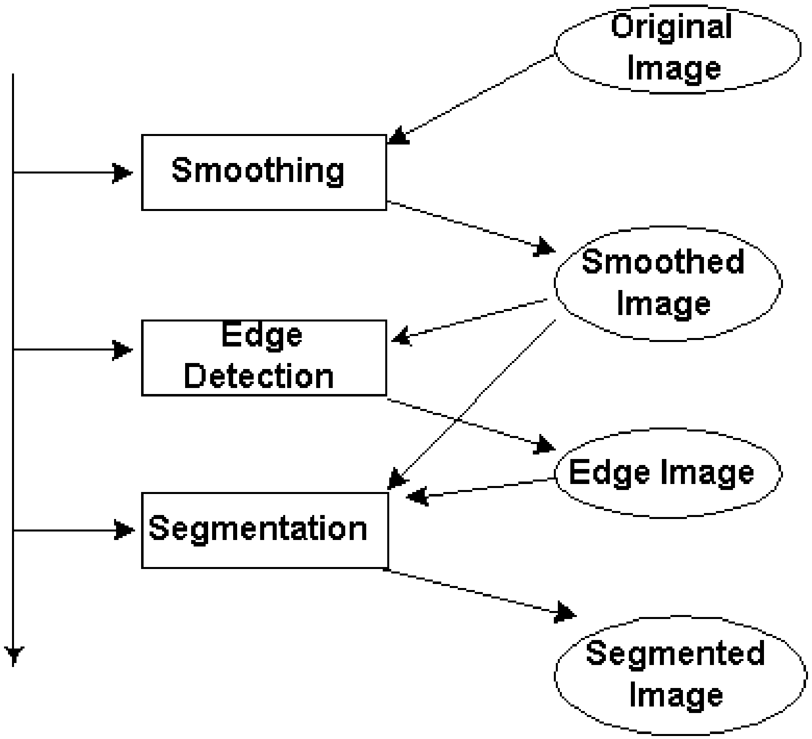
- **Smoothing is usually considered as an important preprocessing step for a segmentation operation that allows a reduction of the noise within an image**
- **It works as low pass filter by making areas more continuous in their color value**
- **It destructs, however, edges within an image**

Face Detection: Edge Detection

- Edge Detection is the important process that preserves the edges and sharp capes
- It works as high pass filter that does not change high frequencies

Face Detection: Segmentation

- **Motivation**
  - Skin regions form candidate faces - detect true faces from skin regions
- **Identify-extract key features using FRG**
  - Apply Fuzzy Region Growing (FRG) Segmentation [Moghaddamzadeh- Bourbakis, 1993]



Illustrative Example
:Original, Smoothed, Edged, Segmented

Fuzzy Region Growing (FRG) Segmentation

- Find big and crisp segments.
- Expand segments based on homogeneity criteria.
- Expand segments based on dichromatic reflection model.
- Expand segments based on degree of farness measure.
- Apply an iterative filter.
- Find medium size segments.
- Expand segments using homogeneity criteria and degree of farness
- Fill in blank regions
- Apply an iterative filter.

Face detection

- **Identify key features**
  - Apply Fuzzy Region Growing (FRG) Segmentation

Skin-segmented Image

FRG Segmented Image

Local Graph

- **Local graph [Bourbakis 1987, 2002]**
  - Represent the shape by a set of line segments
  - Use local graph to encode spatial relationships

i. The individual properties P_j of line Ln_j,

P_{j-} = {*sp*(*starting point*), *l*(*length*), *d'*(*orientation*), *cu*(*curvature*) }

where the index *j* indicates the appropriate segment.

ii. The relationships RL_{ij} among the line segments

RL_{ij} = { *c*(*connectivity*), *p*(*parallelism*), *rd*(*relative distance*), *rm*(*relative magnitude*), *sy*(*symmetry*), *etc* }

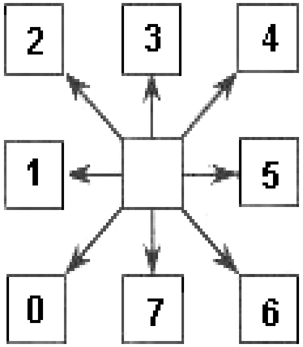
where, the sub index *ij* means the relationship between line *i* and line *j*.

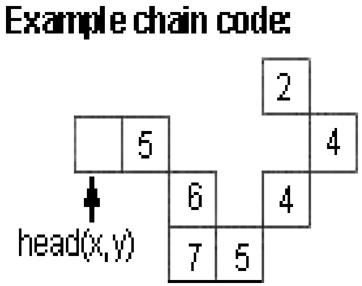
$$SH = \bigcup \{ Ln_j \cdot R^c_{j,j+1} \cdot Ln_{j+1} \}$$

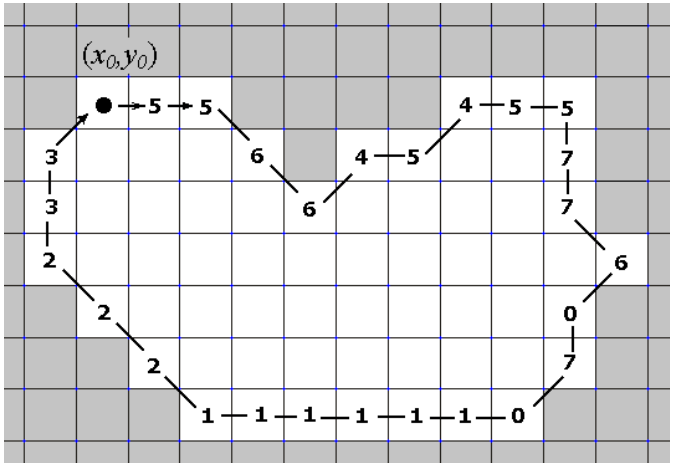
$$= Ln_1 \cdot R^c_{1,2} \cdot Ln_2 \cdot R^c_{2,3} \cdot Ln_3 \cdot R^c_{3,4} \cdot \ldots Ln_{n-2} \cdot R^c_{n-2,n-1} \cdot Ln_{n-1} \cdot R^c_{n-1,n} \cdot Ln_n$$

Chain Coding or Freeman Coding

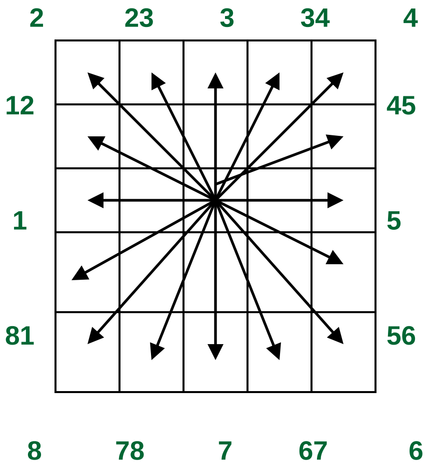
- Chain coding encodes the position of a pixel not by its actual Cartesian coordinates, but rather by its relative position to an adjoining pixel.



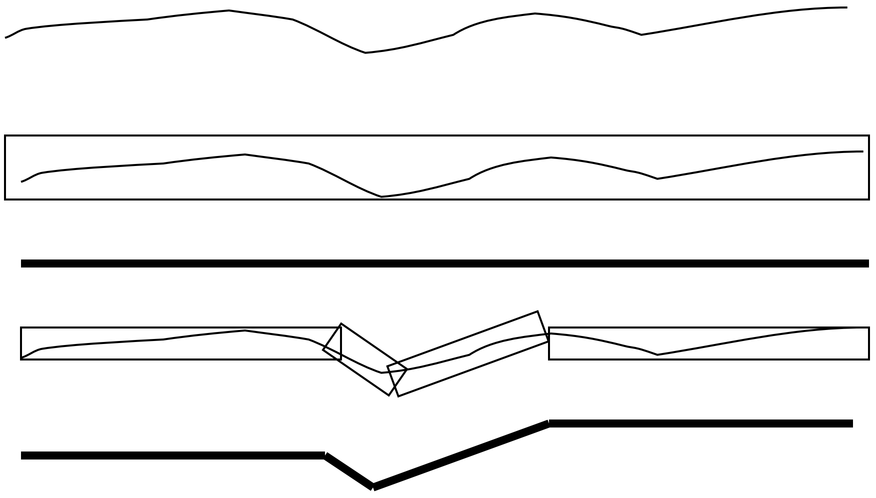


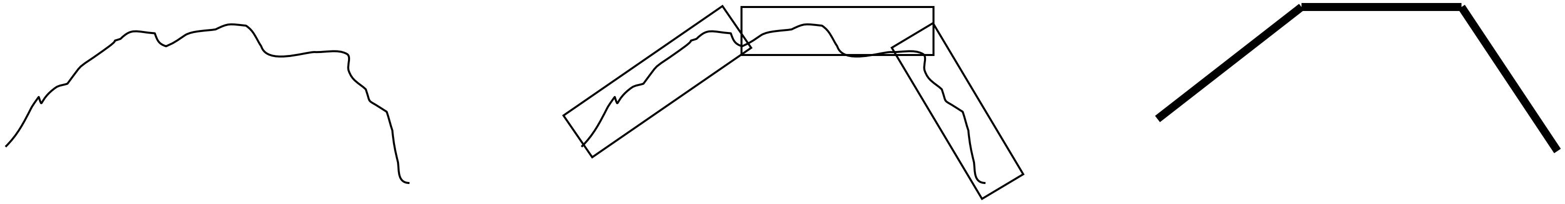


Chain Coding & Line Segments



An extended chain code





Chain Code and Line Segments

String

$$S = k_1(di)k_2(dj)k_3(dn)\ldots k_r(dm)$$

where di $\in$ {1,12,2,23,3,…,78,8} represent directions, see chain code and ki $\in$ Z

$$S = L_1 R_{12} L_2 R_{23} L_3 \ldots L_n R_{n1} L_1$$

where Li represents a line segment and Rij the connectivity relationship with the next line segment Rj

Line Graph Generation

$$S = L_1 R_{12} L_2 R_{23} L_3 \ldots L_n R_{n1} L_1$$

$$g : L \longrightarrow G$$

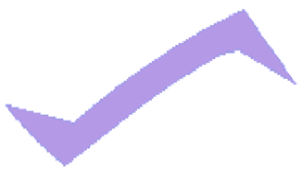
where g(Li) = Ni and g(Rij) = aij

Ni = { sp, orientation, length, curvature}

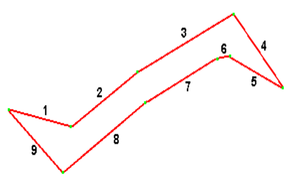
Rij = { connectivity, parallelism, symmetry, etc}

Local Region graph

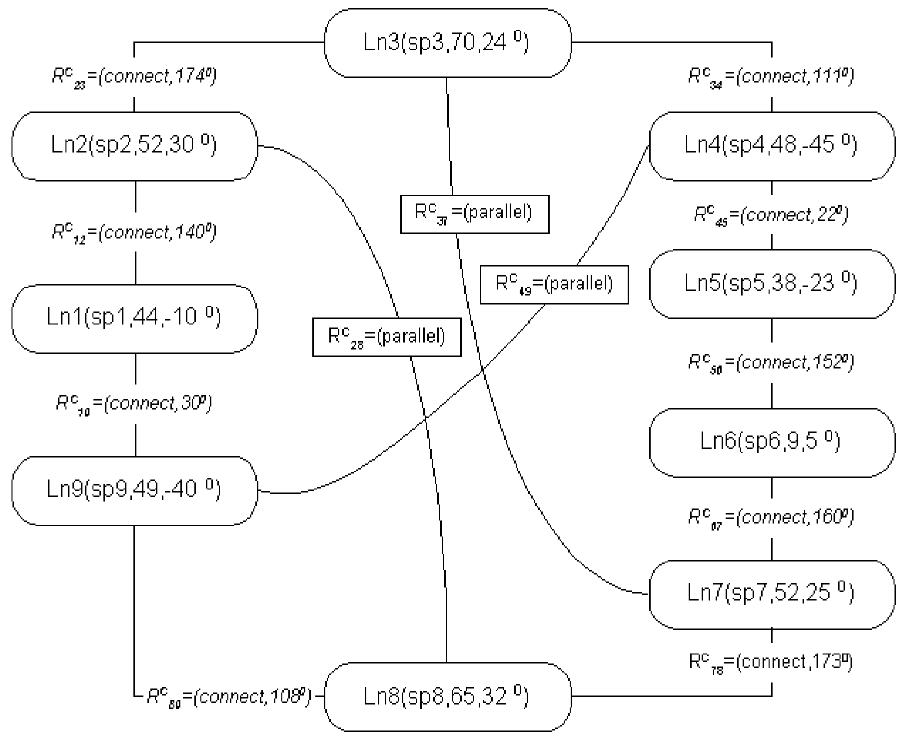
- **Example**



Original region



Line fitted region



Region local graph

Region Matching with Wavelets*: Geometry Transformation*

Single region matching using Wavelets (Yuan-Bourbakis 2002)

A region's border is represented as

$$\boldsymbol{f(t)} = \begin{bmatrix} x(t) \\ y(t) \\ 1 \end{bmatrix}, t=1,2,\dots,m$$

The general form of 2-D geometry transformation matrix is

$$\phi = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} \\ \phi_{2,1} & \phi_{2,2} & \phi_{2,3} \\ 0 & 0 & 1 \end{bmatrix} \qquad \phi = \begin{bmatrix} rs_{xx} & rs_{xy} & trs_{xx} \\ rs_{yx} & rs_{yx} & trs_{xx} \\ 0 & 0 & 1 \end{bmatrix}$$

$$g(t) = S \cdot M_\theta \cdot f(t) + T$$

$$\begin{bmatrix} x'(t) \\ y'(t) \\ 1 \end{bmatrix} = \begin{bmatrix} rs_{xx} & rs_{xy} & trs_{xx} \\ rs_{yx} & rs_{yx} & trs_{xx} \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x(t) \\ y(t) \\ 1 \end{bmatrix}$$

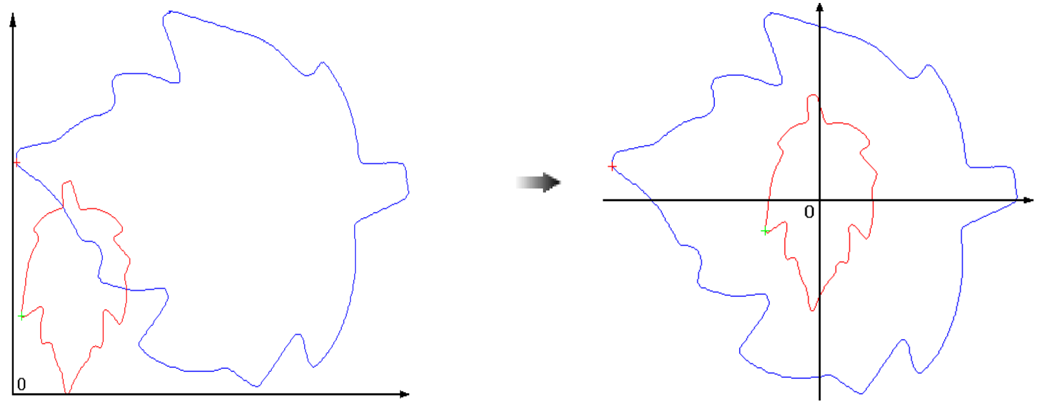
Region Matching: *Translation*

- **we set the scale factors to both *x* and *y* directions are the same, i.e. $s \equiv s_x \equiv s_y$, thus.**

$$g(t) = s \cdot M_\theta \cdot f(t) + T$$

$$\begin{aligned} F_{cen}(g(t)) &= F_{cen}(s \cdot M_\theta \cdot f(t) + T) \\ &= F_{cen}(s \cdot M_\theta \cdot f(t)) + F_{cen}(T) \\ &= F_{cen}(f(t)) + T \end{aligned}$$

$$\rightarrow \quad T = F_{cen}(g(t)) - F_{cen}(f(t))$$



Translate parameter example

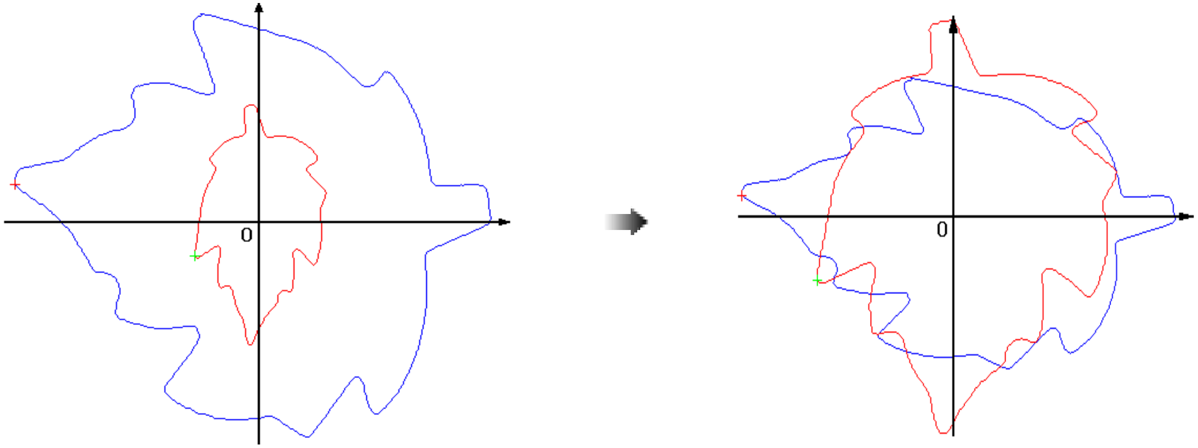
Region Matching: Scale

- **Momentum is a measure of object's mass distribution.**

$$Mom' = \frac{1}{N}\sum_{i=1}^{N} m_i \cdot \left\| p'_i - p'_{centroid} \right\|^2$$
$$= \frac{1}{N}\sum_{i=1}^{N} m_i \cdot \left\| s \cdot p_i - s \cdot p_{centroid} \right\|^2$$
$$= s^2 \cdot \frac{1}{N}\sum_{i=1}^{N} m_i \cdot \left\| p_i - p_{centroid} \right\|^2$$
$$= s^2 \cdot Mom$$

$$Mom = \frac{1}{N}\sum_{i=1}^{N} m_i \cdot \left\| p_i - p_{centroid} \right\|^2$$

$$\Rightarrow \quad s = \left(\frac{Mom'}{Mom} \right)^{1/2}$$



Scale parameter example

Region Matching: *Rotation*

The rotate matrix ***M*** **is defined as**

$$M = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

the rotated curve ***f'(t)*** **is computed by**

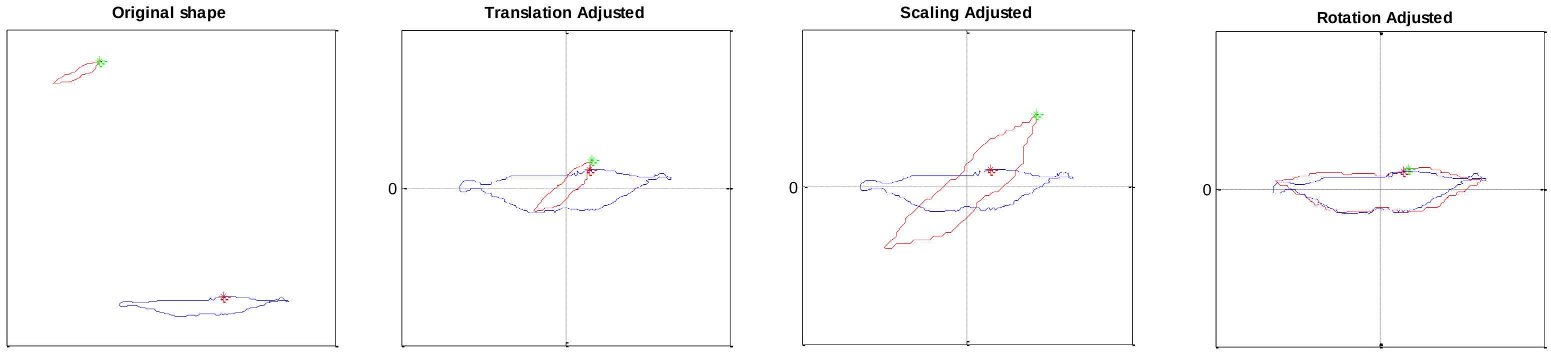
$$f'(t) = \begin{bmatrix} x(t)' \\ y(t)' \\ 1 \end{bmatrix} = M * f(t) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x(t) \\ y(t) \\ 1 \end{bmatrix}$$

Open problems are

- Rotate angle
- Point correspondence

Region matching: Lips example

- **Single region matching using Wavelets (Yuan-Bourbakis 2002)**
  - Translation
  - Scaling
  - Rotation



Example of single region matching (model region in *blue*, object region in *red*)

Region Synthesis

- **Synthesis of regions (Bourbakis 1987)**

- **Motivation**
  - For face detection, need to Identify the key facial regions and their spatial relationships.
  - Other regions - not necessary!

Region Synthesis steps

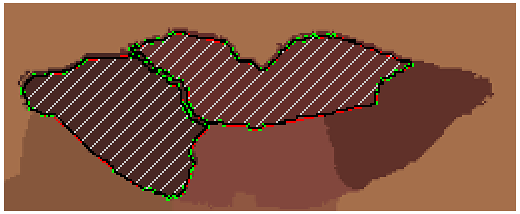
1. ***Initialize the first region which is closer to average skin color as the active skin region. The degree of closeness is calculated as the RGB color difference.***
2. ***Select the next region which is closer to average skin color. Find the common edge between this region and the active region.***
3. ***If a common edge is found, synthesize the current region and the active region. Assign new region to active region.***
4. ***If all regions have been processed, region synthesis completes; otherwise go to step 2.***

Region Synthesis

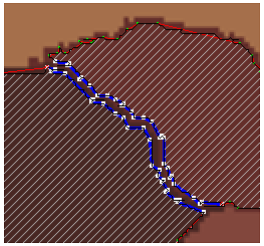
- **Skin region synthesis**
  - Merge neighbor regions into a single segment
    - Neighborhood region searching based on *skin-color similarity*
    - Find common edge using local graph

$$L_1 = Ln_1 R_{12}^c Ln_2 R_{23}^c Ln_3 R_{34}^c \ldots Ln_{n-1} R_{n-1n}^c Ln_n$$

$$L_2 = L'n_1 R_{12}^c L'n_2 R_{23}^c L'n_3 R_{34}^c \ldots L'n_{m-1} R_{m-1m}^c L'n_m$$



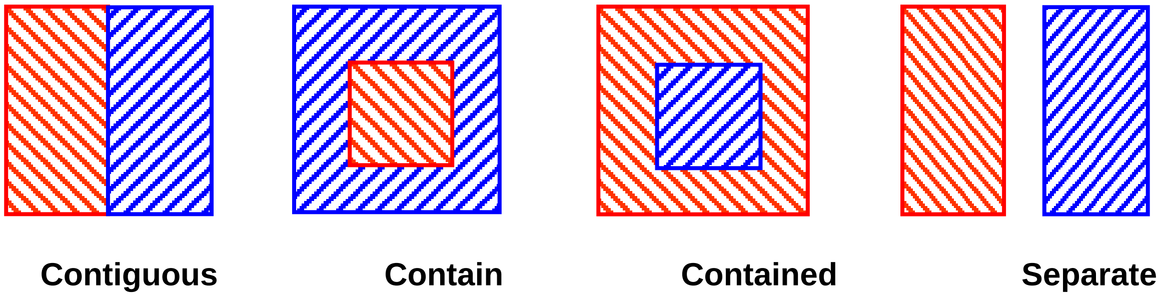
Selected regions



Detected common edge

Region Synthesis

.Relationships among regions

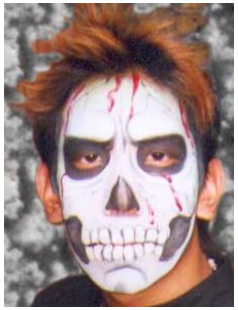


$$shape(R_{12}) = \begin{cases} shape(R_1)shape(R_2), & if\ REL(R_1, R_2) = contiguous \\ shape(R_1), & if\ REL(R_1, R_2) = contain \\ shape(R_2), & if\ REL(R_1, R_2) = contained \\ \phi, & if\ REL(R_1, R_2) = separate \end{cases}$$

Skin region synthesis

- **Skin region removal or face-lifting**
  - Remove skin region - Do not consider it further!
  - Advantage - Simplifies LG graph matching

Skin region synthesis procedure for face lifting for extracting facial features



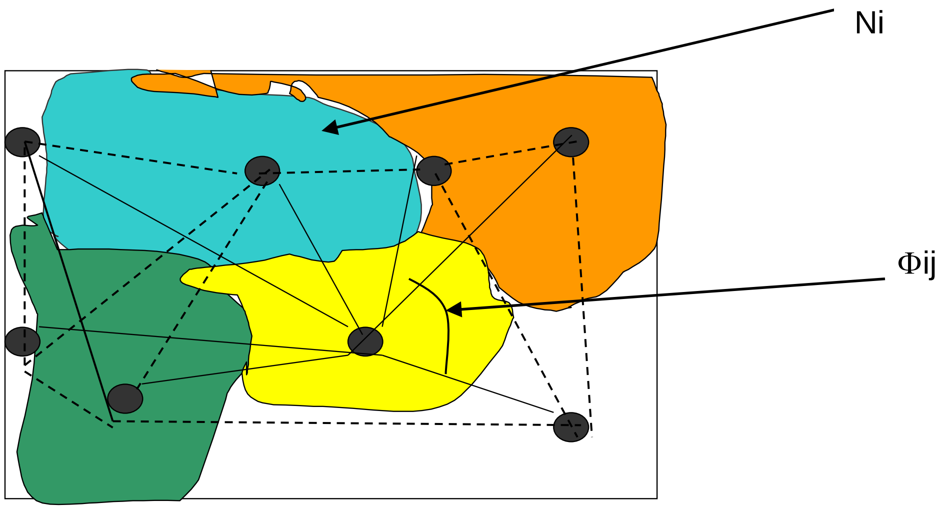
Skin removal

Global Graph

- The Global graph is a structure that carries inter-region relations. Each node of this graph represents a region, i.e. it contains the respective local graph.

$$GG(I_p) = (P_1 \mathrm{R}_{12} P_2)\Phi_{23}(P_1 \mathrm{R}_{13} P_3)\ldots$$
$$(P_1 \mathrm{R}_{1\mathrm{n}\text{-}1} P_{n-1})\Phi_{n-1n}(P_1 \mathrm{R}_{1\mathrm{n}} P_n)$$

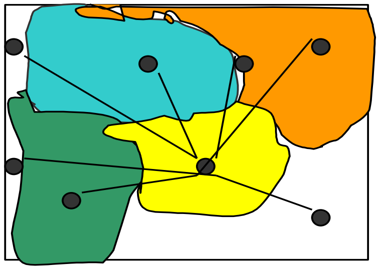
- The links of this graph represent the relations between regions



G(A(N1))=(N1R12N2)
Φ23(N1R13N3)
Φ34(N1R14N4) …
Φ67(N1R17N7)
Φ78(N1R18N8)Φ81

Image L-G Graph Comparison

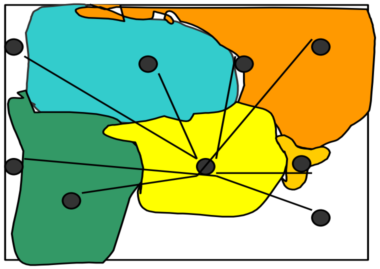
IMAGE -A
Image regions and the graph of gravity



$$G(A_{(N1)}) = (N_1R_{12}N_2)\ \Phi_{23}\ (N_1R_{13}N_3)\ \Phi_{34}\ (N_1R_{14}N_4)\ \ldots$$
$$\Phi_{67}\ (N_1R_{17}N_7)\ \Phi_{78}\ (N_1R_{18}N_8)\ \Phi_{81}$$

IMAGE -B
Image regions and the graph of gravity

$$G(A_{(N1)}) = (N_1R_{12}N_2)\ \Phi_{23}\ (N_1R_{13}N_3)\ \Phi_{34}\ (N_1R_{14}N_4)\ \ldots$$
$$\Phi_{67}\ (N_1R_{17}N_7)\ \underline{\Phi_{78}\ (N_1R_{18}N_8)\ \Phi_{89}\ (N_1R_{19}N_9)\ \Phi_{91}}$$



Comparison A and B
7/8 region relationships same
5/7 angles same

Local-Global (LG) Graph

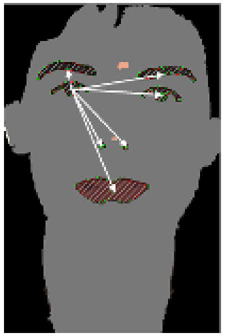
- **Image representation with LG Graph (Bourbakis 1987)**
  - Use Delaunay Triangulation
  - Graph edges hold the spatial relationships between facial features
  - Graph nodes hold information about key facial features

$$node = \{Centroid(x, y), color / texture, Local - graph(L), size, border\}$$

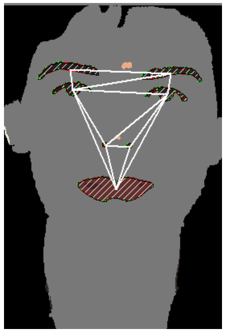
$$LG = \{NodeSet, EdgeSet\}$$

Model face image

Selected facial regions



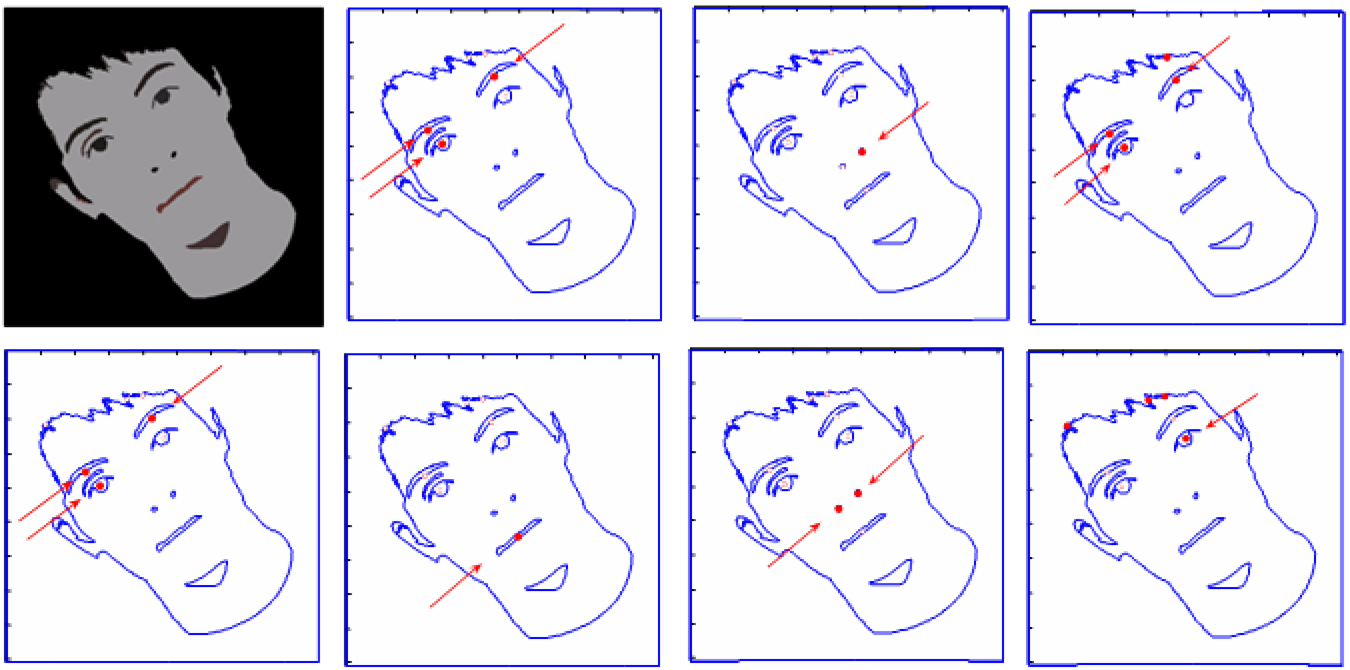
Local - Global Graph view



Delaunay graph

LG Graph matching

- **Potential Region Correspondent Pair (PCRP)**
  - Node correspondence - select regions based on *color similarity*
  - Random graph matching is not allowed!



PCRP region selection

LG Graph matching

- **Graph similarity - represented by angle**
  - If the correspondent angles between arcs are similar, the graphs are similar
  - Angular similarity

$$SIM_{LG} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} E(i,j).S_{ANGSIM}(\theta_{ij} - \theta_{i0})$$

SIM_{LG} - similarity between two graphs

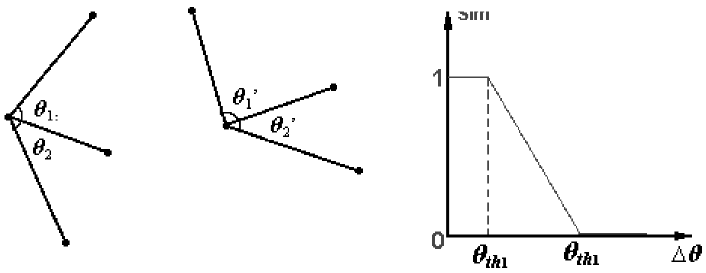
N - total number of nodes

- angle of the edge, $E(i, j)$

θ_{ij} - base angle

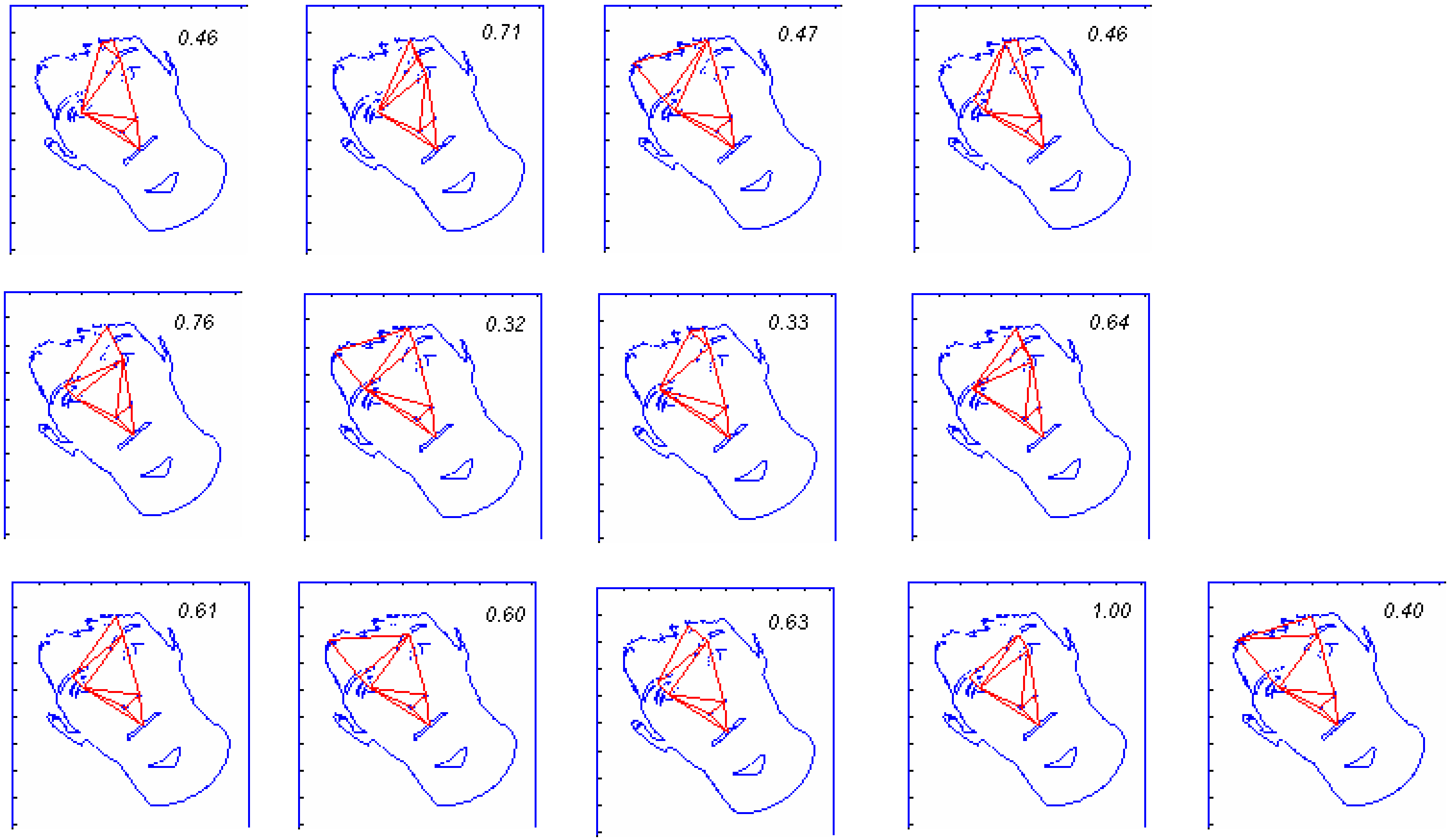
θ_{i0} - angle similarity function

$S_{ANGSIM}(\Delta\theta)$



LG Graph matching

- **PCRP graphs and angular similarity**



LG Graph matching

- **LG Graph shape constraint**
  - Nodes are not just fiduciary points!!
  - Use shape from local graph as a similarity constraint

$$SIM_{LG} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} E(i, j).S_{ANGSIM}(\theta_{ij} - \theta_{i0}).Weight(i, j)$$

- **LG Graph relationship checking**
  - Relationship similarity *(T_{RE})* - Contiguous, Contain, Contained, Separate

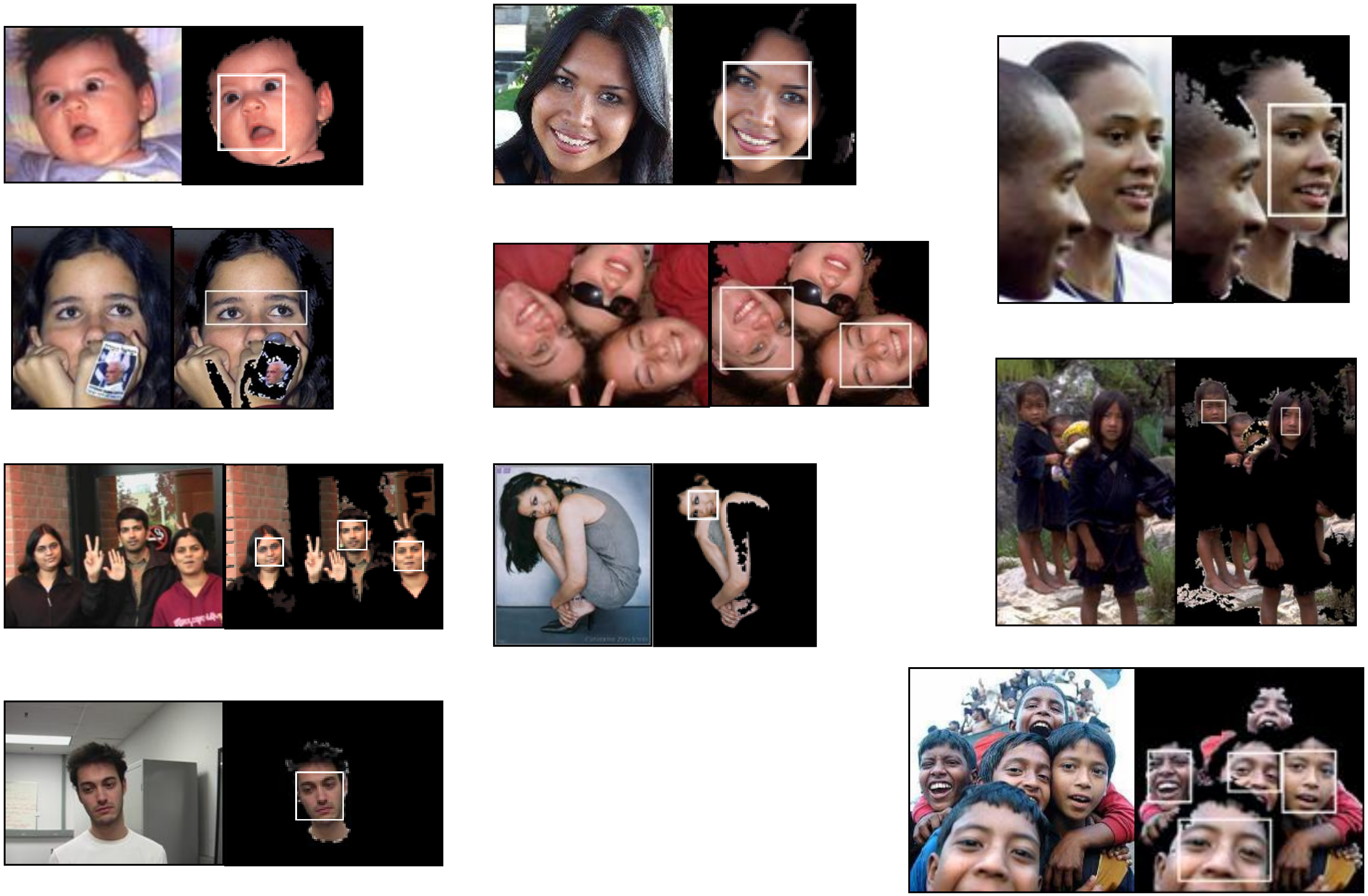
$$SIM_{REL} = \prod_{i=1}^{N} \prod_{j=1}^{N} T_{RE}(r_{i,j}, r'_{i,j})$$

* ***LG Graph error***

$$\boldsymbol{ERR_{LG} = (1 - ERR_{rel}) x (ERR_{graph} + ERR_{shape})}$$

Experimental results

- **LGG matching**



Experimental results

- **Performance on AR face database [Martinez & Benavante, 1998]**

	Proposed method	***Chiang & Huang, 2005***	
		AR	*AR'*
Number of faces	*240 (30 people *8 images / person)*	*945*	*540*
Number of correct detections	*231*	*874*	*511*
Number of misses/errors	*9*	*14/66*	*4/26*
Recall rate	***96.25***	*92.48*	*94.68*

- **Limitations**
  - Low-image size
  - Poor-image qualities
  - Profile-views (>45^0)
  - Dependence on Skin detection method

L-G Graph Expression Recognition

- **LG Expression models**
- **Expression recognition**
- **Results**

Basic facial expressions

- **Basic six facial expressions [Ekman, 1993, Kanade et al., 2000]**
  - Happy, Surprise, Sad, Anger, Disgust and Fear

Happy

Surprise

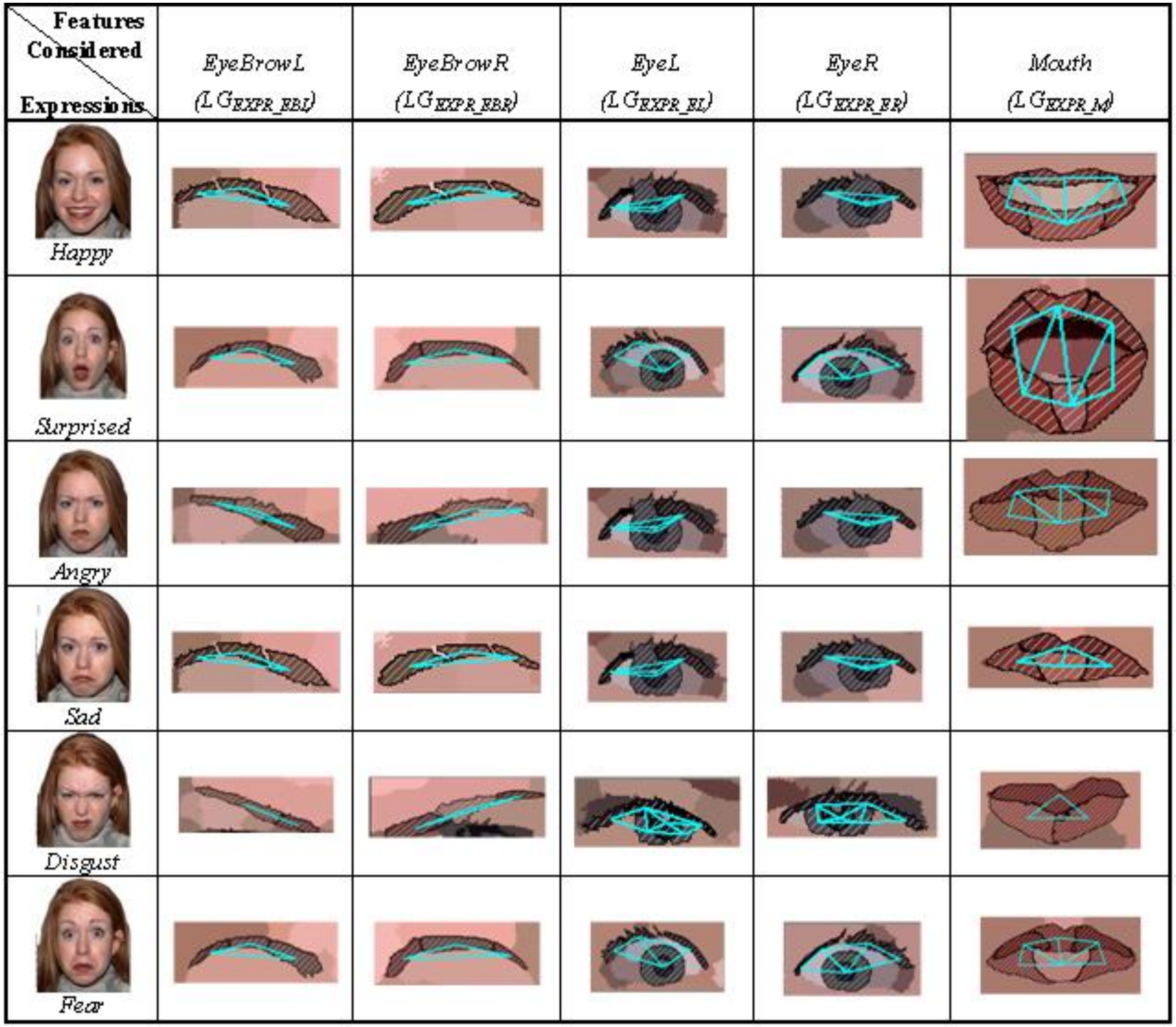
Angry

Disgust

Sad

Fear

LG Expression models



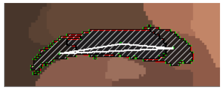
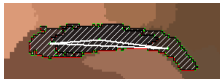
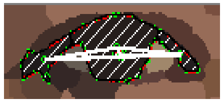
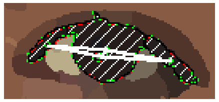
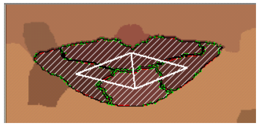
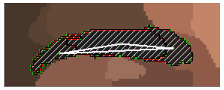
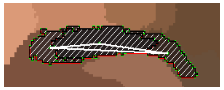
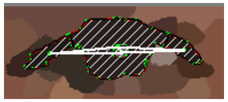
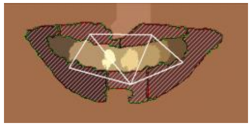
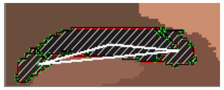
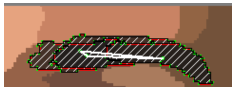
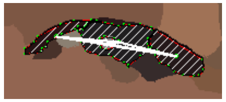
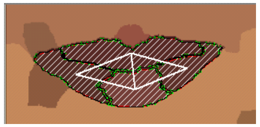
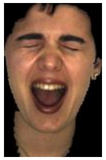
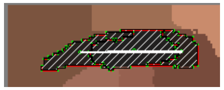
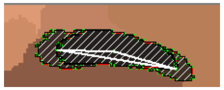
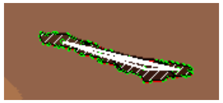
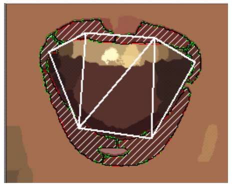
Expression recognition

- **Steps in expression recognition [Faisel & Luttin, 2003]**
  - Face detection
  - Feature recognition
    - Nodes in LG graph
  - Expression recognition
    - Construct LG expression graphs

$$node = \{Centroid(x, y), color / texture, L, size, border, LG_{EXPR1}, ..LG_{EXPRi}\}$$

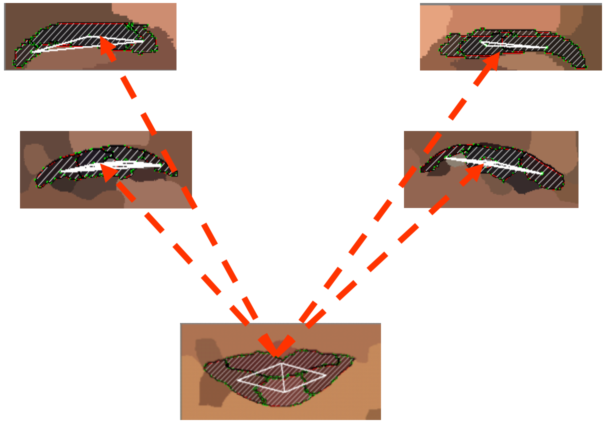
- **Features used**
  - Five facial features - eyes, eye-brows, mouth

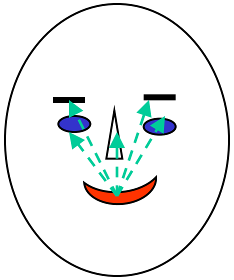
LG Expression models [Kakumanu-Bourbakis, in press]

Features Considered / **Expressions**	*EyeBrowL* (LG_{EXPR_EBL})	*EyeBrowR* (LG_{EXPR_EBR})	*EyeL* (LG_{EXPR_EL})	*EyeR* (LG_{EXPR_ER})	*Mouth* (LG_{EXPR_M})
 *Neutral*					
 *Happy*					
 *Angry*					
 *Scream*					

Facial Expression LG graph

- Example

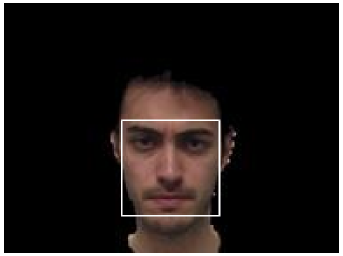




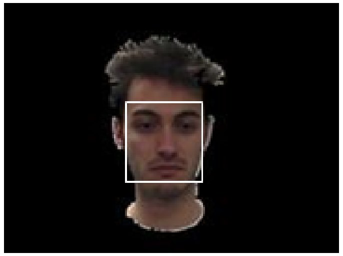
Model in GDB

Facial Expressions

- **Results**

Happy = 0.72
Sadness = 0.44
Surprise = 0.77
Disgust = 0.54
Fear = 0.52
Anger = 0.29
Expression = Angry

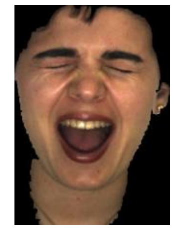



Happy = 0.82
Sadness = 0.34
Surprise = 0.78
Disgust = 0.51
Fear = 0.53
Anger = 0.59
Expression = Sad

Experimental results

- **Performance on AR database [Martinez & Benavante, 1998]**

	Neutral	**Happy**	**Angry**	**Scream**
Neutral	***0.875***	*0.000*	*0.125*	*0.000*
Happy	*0.025*	***0.900***	*0.025*	*0.050*
Angry	*0.200*	*0.025*	***0.775***	*0.000*
Scream	*0.000*	*0.050*	*0.000*	***0.950***

Expression LG Graph Errors		Neutral 	Happy 	Angry 	Scream 
	EyeL	*0.72*	*0.53*	*0.22*	***0.26***
	EyeR	*0.78*	*0.52*	*0.27*	***0.28***
	EyeBrowL	*0.33*	*0.34*	*0.40*	***0.42***
	EyeBrowR	*0.37*	*0.37*	*0.39*	***0.42***
	Mouth	*0.85*	*0.67*	*0.82*	***0.08***
	Avg. Error	*0.61*	*0.49*	*0.42*	***0.29***

Ekfrasis

- **. Definition: The Ekfrasis language is defined (or generated) by a grammar G {V_N, V_T, PR, S},**
- **where V_N, is the set of non-terminal symbols and is defined as V_N = {S, T, k, L, X}; V_T is the set of terminal symbols and is defined as $V_T = \sum \cup \{ i / i \in Z, \} \cup \{\#\}$**
- **S is the starting symbol of a sentence; T is the symbol for a terminal letter; L is the symbol for the alphabet letters; ∑ is the alphabet; # is the synthesis symbol between letters of the alphabet; and PR is the set of production rules and is defined as**
- **PR={ S → T; S → S # T; T → L_k; L→ $L_1/L_2/L_3/L_4/L_5/L_6$; $k \in Z$, $1 \le k \le 6$}; and $L_i \in \sum$,**
- **where $\sum$ = {EBL_i, EBR_i, EL_i, ER_i, N_i, M_i (UL_i,LL_i)}**

Language

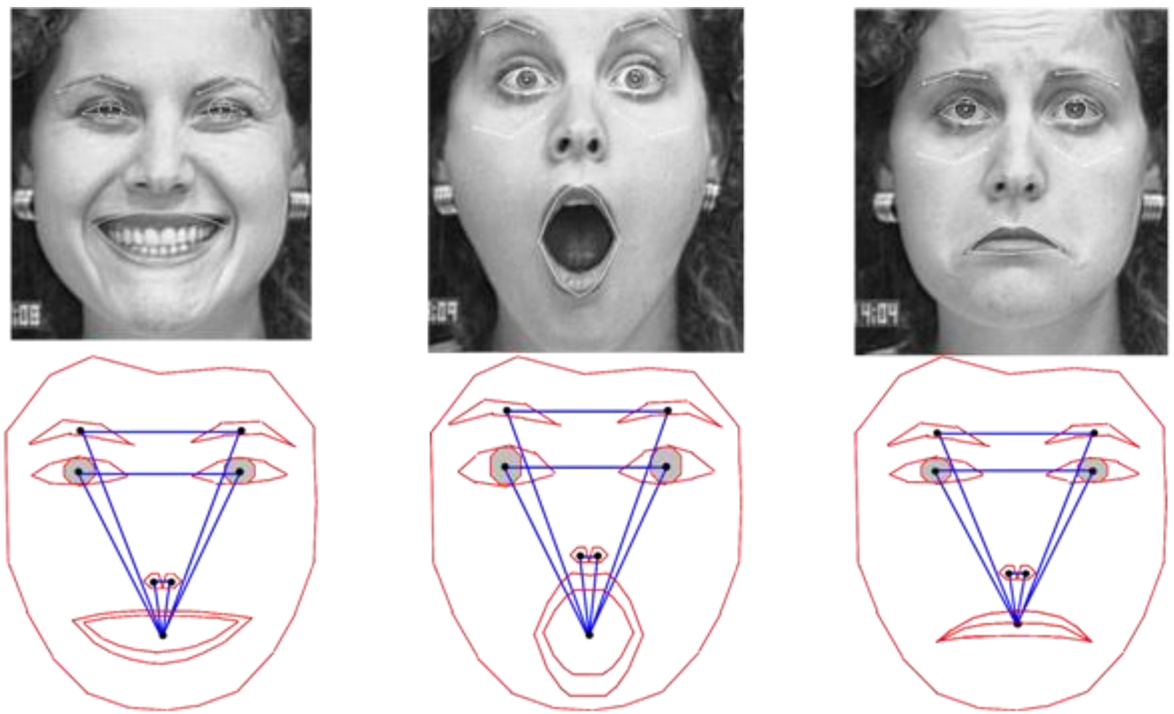
-

- ***Definition*: The Ekfrasis language (L_{EF}) is defined over the G grammar as follows:**

$$L_{EF}\,(G) = \{\, Le_i / Le_i \,,\, V_T \;:\; S \;{}_{G}\!\rightarrow Le_i \,\}$$

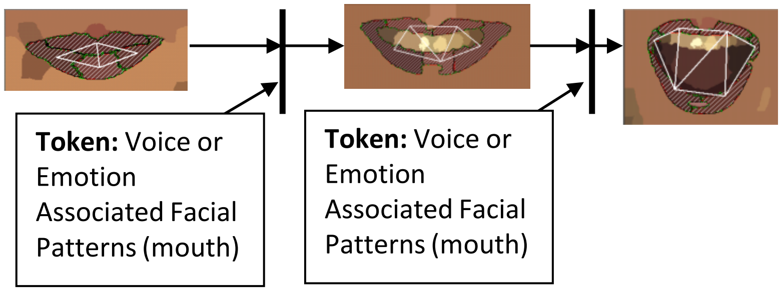
Expression recognition (II)

- **LG Expression graphs**
  - LG expression graphs for each expression to be recognized
  - Expression graph matching

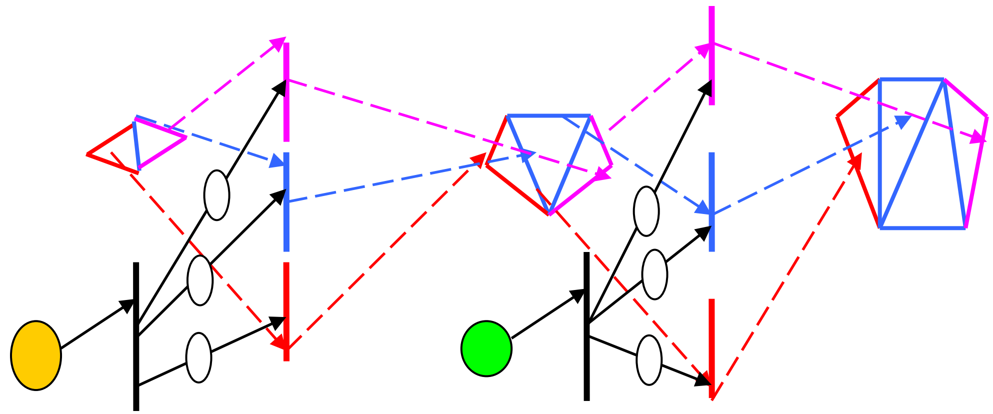


(a) Happy Face Model (b) Surprise Face Model (c) Sad Face Model

SPN



It shows a sequence of facial features (mouth) L-G graphs represent letters from the Ekfrasis language. In this case, individual letters of the alphabet can be used to associate emotional patterns

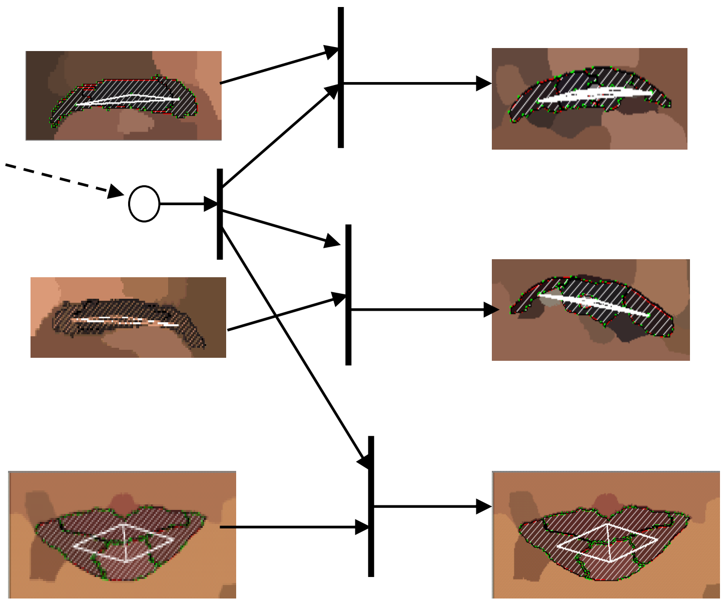


It shows the SPN association of the L-G graph of the facial features (mouth) extracted and represented from different image frames, and their activation via token (orange, green color. This case is a sequence of facial expressions related with happiness and laughter).Thus, the color token could be a joke or a happy thought or else.

SPN

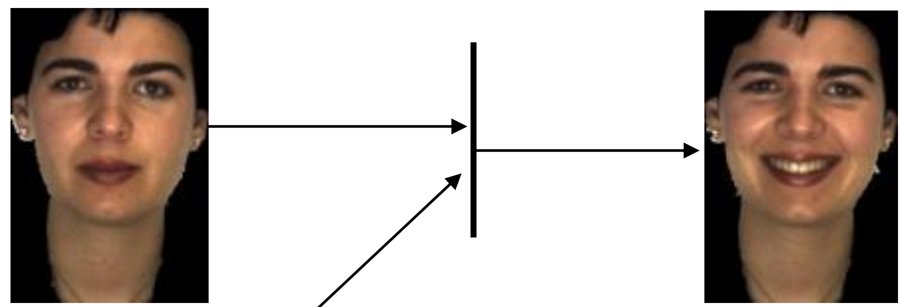
-

This is the token that has triggered the transition from one state of expressions into another.



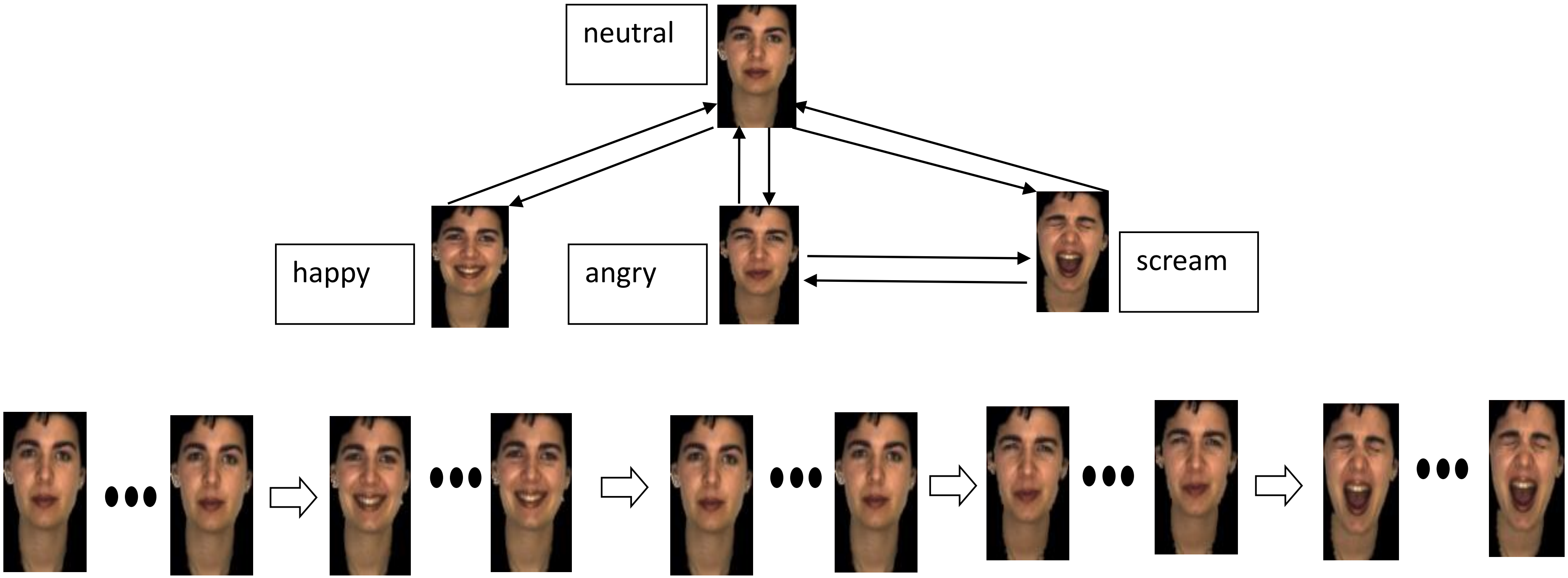
It shows the transition from the neutral into angry

SPN



Token: Voice or Emotion Associated Facial Patterns (mouth). This token triggers the transition from one facial expression into the next.

SPN transitions



It graphically shows :a) The transition from a facial expression (neutral state) into the next facial one (happy state);

b) The order of transitions of facial expressions (from a “state” into another “state”) that take place for the most frequently used facial expressions (neutral, happy, angry, scream);

c)Frames of expressions showing the sequence of transitions in facial expressions.

Applications (II)

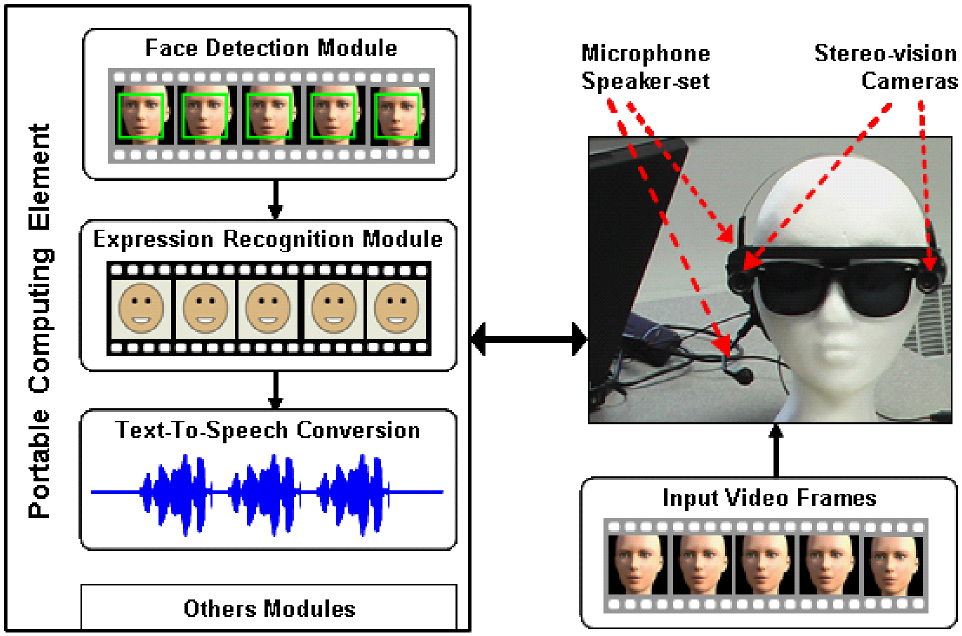
- **Biometrics**
  - Identification & Authentication

ATM Kiosk –
Face Verification

Multibiometrics - voice, face, and fingerprint recognition

Applications to People with Disabilities

- **Emotional Behavior of People with Disabilities (*collaboration with Dr. Esposito, 2003*)**
- **Hearing Impaired Individuals (*Koufos project*)**
- **Mentally Retarded Individuals (*Anapiros project*)**
- **Visually impaired individuals (*Tyflos project*)**



Summary

Face Detection LG graph Method

LG graph matching

- spatial (graph), shape and relation constraints
- Synthesis of regions
- Compact representation - local (facial features) and global (topology) info
- Invariant to translation, rotation and scale (to an extent)

Facial Expression Detection LG graph Method

Applications

Thank you for your patience

QUESTIONS?