

# Analyse et modélisation de visages



# Analyse et modélisation de visages

---

## Plan

- Introduction
- Outils d'analyse
- Phase de détection
- Phase d'alignement et de suivi
- Du modèle rigide au modèle déformable

# Analyse et modélisation de visages

---

## Plan

- Introduction
- Outils d'analyse
- Phase de détection
- Phase d'alignement et de suivi
- Du modèle rigide au modèle déformable

# Introduction

---

Pourquoi l'analyse de visages ?



# Introduction

---

Pourquoi l'analyse de visages ?



# Introduction

---

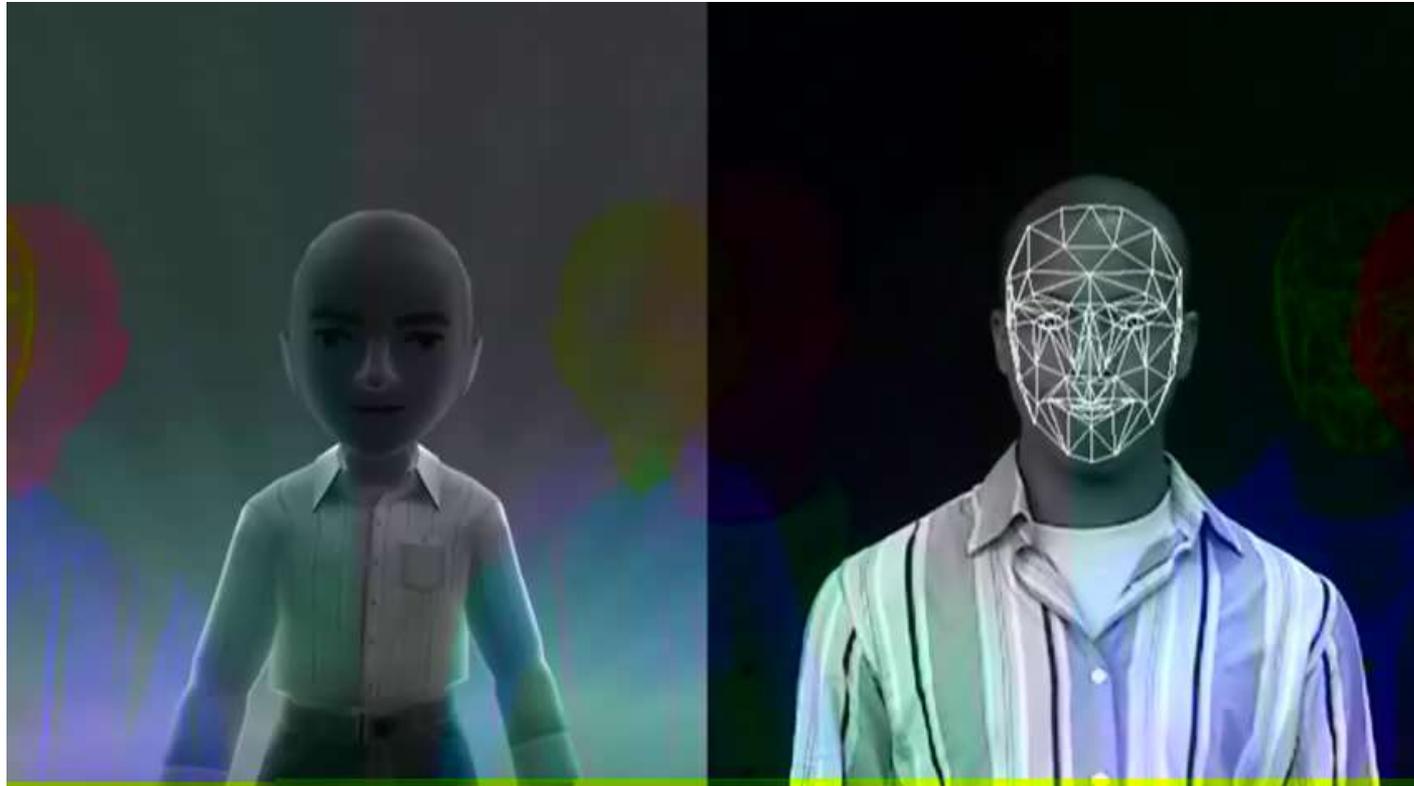
Pourquoi l'analyse de visages ?



# Introduction

---

Pourquoi l'analyse de visages ?



# Introduction

---

Pourquoi l'analyse de visages ?



# Introduction

---

Pourquoi l'analyse de visages ?



# Introduction

---

Pourquoi l'analyse de visages ?



# Introduction

---

Pourquoi l'analyse de visages ?



# Introduction

---

## Applications

- Reconnaissance
- Morphing
- Simulation de vieillissement
- Mesure de l'attention (prévention routière...)
- Analyse comportementale (réactions, émotions, ...)
- Miroir virtuel (essayage en ligne : maquillage, lunettes, ...)
- Visio-conférence, jeux vidéos
- ...

# Introduction

---

## Traitement d'images et vision artificielle

- Niveau 0 : représentation  
*(acquisition, compression)*
- Niveau 1 : transformation « image vers image »  
*(amélioration, restauration, segmentation)*
- Niveau 2 : transformation « image vers paramètres »  
*(description)*
- Niveau 3 : transformation « paramètres vers décision »  
*(reconnaissance et interprétation)*

# Introduction

---

Les phases classiques de l'analyse :

- Détection
- Alignement et suivi
- Reconnaissance/classification

# Introduction

---

## Les défis

Pose



Expression



Illumination



Résolution



# Introduction

---

## Les outils

- Description
  - Recherche de descripteurs discriminants
  - Variations globales ? Locales ? Granularité ?
- Modélisation
  - Modèles de texture, modèles de forme
  - Paramétrage
- Apprentissage
  - Bases d'exemples
  - Généralisation

# Analyse et modélisation de visages

---

## Plan

- Introduction
- Outils d'analyse
- Phase de détection
- Phase d'alignement et de suivi
- Du modèle rigide au modèle déformable

# Outils d'analyse

---

L'apprentissage automatique « permet à une machine d'évoluer grâce à un processus d'apprentissage, et de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques »

- Quelle est la bonne orthographe de ce mot ?
- Comment traduire cette phrase ?
- Le patient aura-t-il un accident cardio-vasculaire ?
- Qui est l'auteur de cette page HTML ?
- Quelle sera la taille de cet enfant à l'âge adulte ?

Nous ne souhaitons pas écrire des lois qui répondent à ces questions, mais les découvrir automatiquement, par observation d'exemples et de contre-exemples (apprentissage supervisé), dans le but de prédire, c'est-à-dire classer de nouveaux exemples.

# Outils d'analyse

---

**L'apprentissage automatique** « permet à une machine d'évoluer grâce à un processus d'apprentissage, et de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques »

- Mise en place de **raisonnements automatiques sur des faits et règles**
- Où trouver ces règles ? L'humain ne peut pas toujours expliciter simplement son expertise
- Solution : lui demander de faire ce qu'il sait faire, **l'observer**, et **apprendre**

# Outils d'analyse

---

## Les points à définir

- Ce que l'on veut prédire
- La nature et la modalité d'obtention des exemples
- Le mode d'évaluation des prédictions
- La méthode d'apprentissage
- L'étiquetage des exemples
  - Binaire (-1 ou 1, vrai ou faux)
  - Discret (classification)
  - Continu (régression)
  - Vectoriel (sortie structurée)
  - Valeur de confiance

# Outils d'analyse

---

## Les difficultés

- Nombre d'exemples
- Bruit dans les données
- Valeurs manquantes dans les descriptions
- Déséquilibre de classes
- Complexité de la description (malédiction de la dimensionnalité)
- Garantie de l'existence d'un lien description-cible (corrélation)
- Evolution du lien dans le temps
- Erreur de généralisation (sur-apprentissage)

# Outils d'analyse

---

## Les méthodes

- Réseaux de neurones
- Processus stochastiques
- Analyse statistique (ACP, LDA, ICA...)
- Arbres de décision
- Machines à vecteurs de support (SVM)
- Boosting

... mais commençons par la notion de « descripteur »

# Outils d'analyse

---

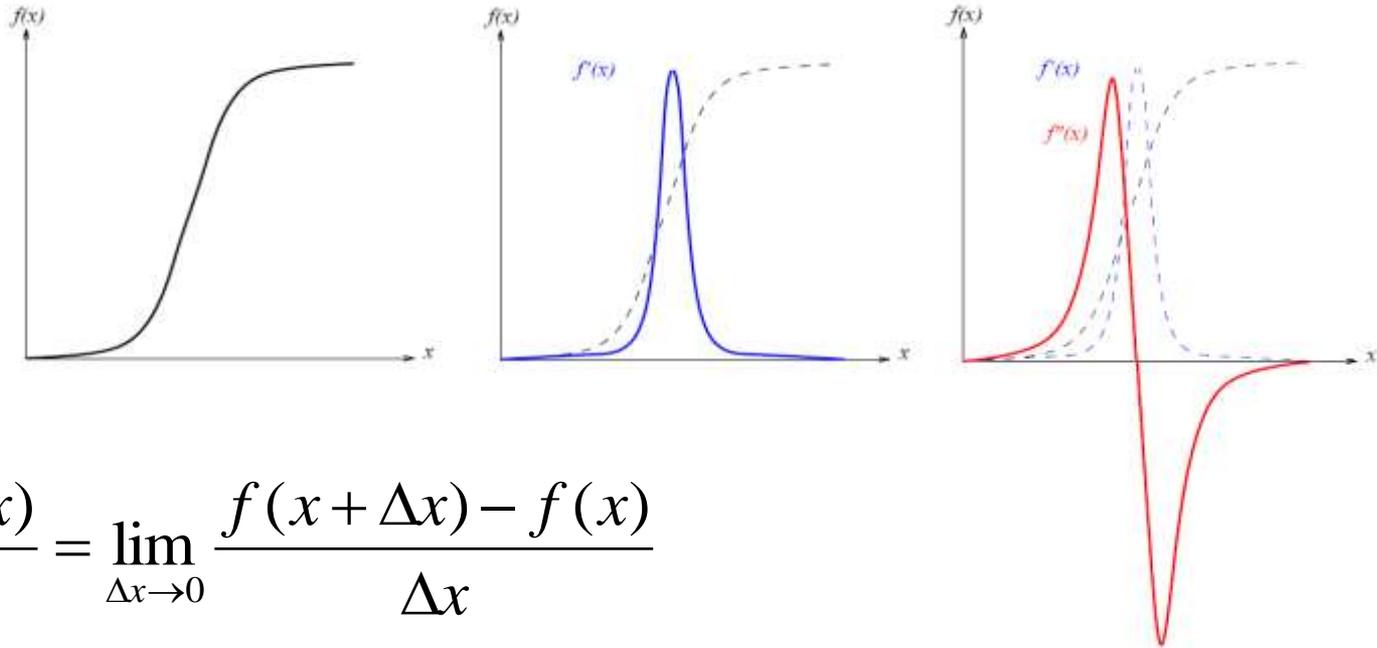
Le choix de l'espace de description consiste à rechercher des attributs significatifs de la base d'images, permettant d'en obtenir une représentation compacte (moins de données, plus de sémantique). Il pourra s'agir :

- De descripteurs de couleur
- De descripteurs de texture
- De descripteurs de forme

# Outils d'analyse

---

## Opérateurs différentiels



$$\frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

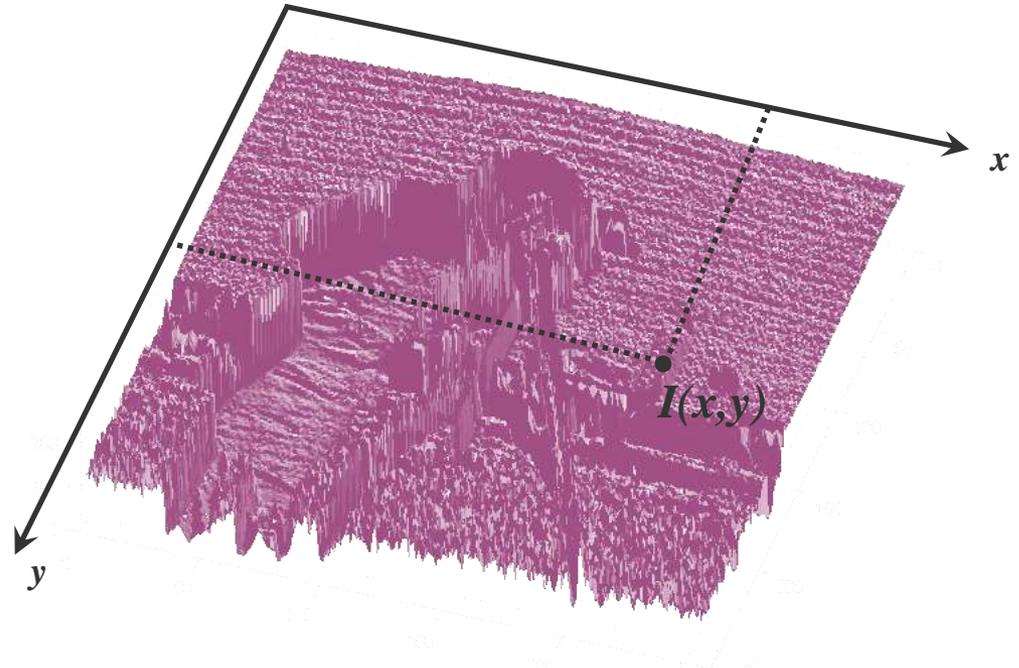
$$\frac{\tilde{d}f[i]}{di} = f[i + 1] - f[i]$$

# Outils d'analyse

## Opérateurs différentiels



*Représentation usuelle*



*Représentation surfacique*

# Outils d'analyse

## Gradient

$$\nabla I(x, y) = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}$$

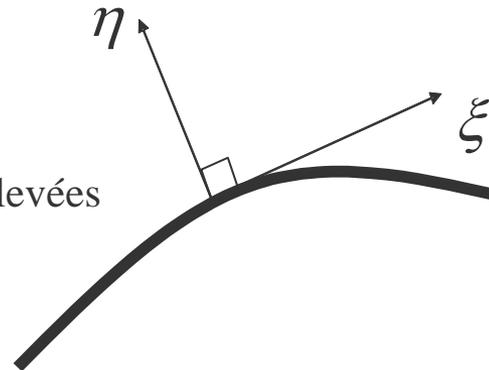
$$\|\nabla I\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

$$\arg(\nabla I) = \arctan\left(\frac{\partial I / \partial y}{\partial I / \partial x}\right)$$



Valeurs  $\|\nabla I\|$  faibles  
(zones homogènes)

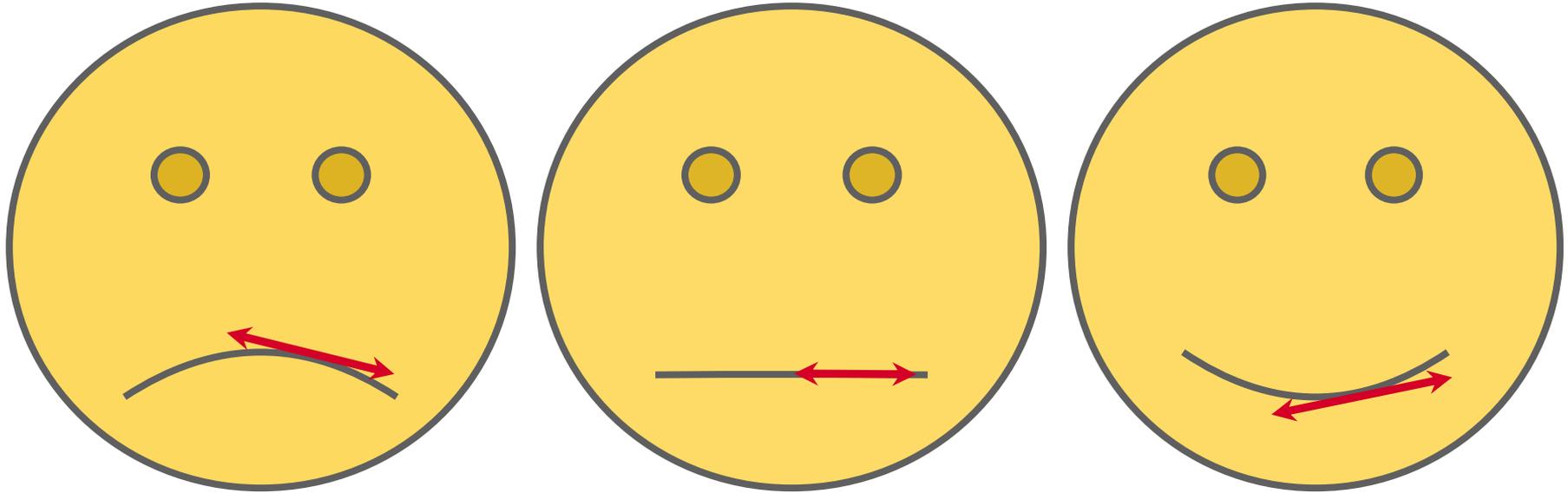
Valeurs  $\|\nabla I\|$  élevées  
(contours)



# Outils d'analyse

---

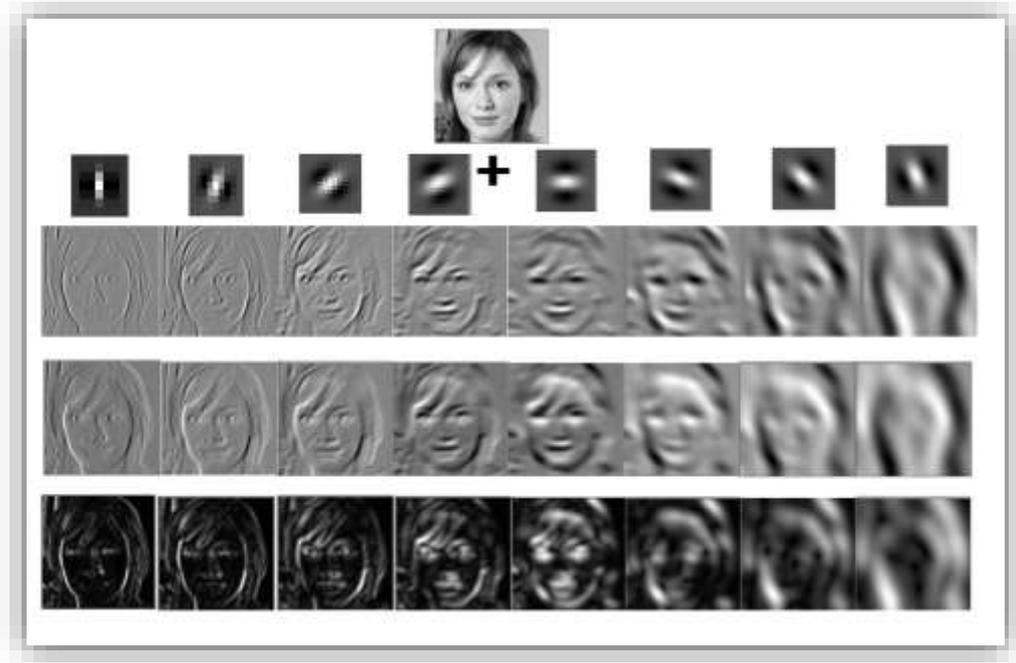
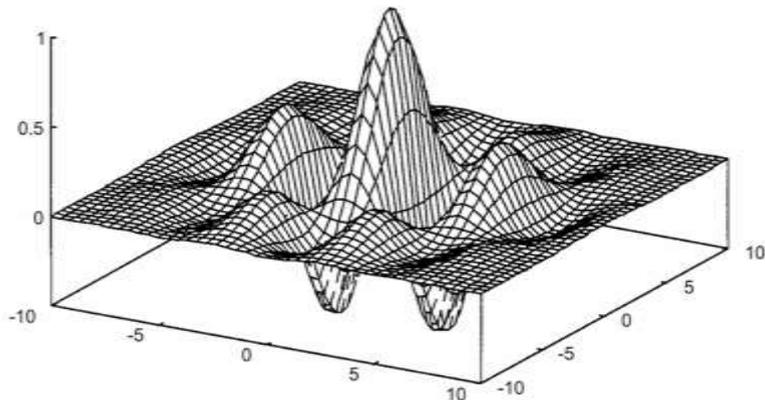
## Gradient



# Outils d'analyse

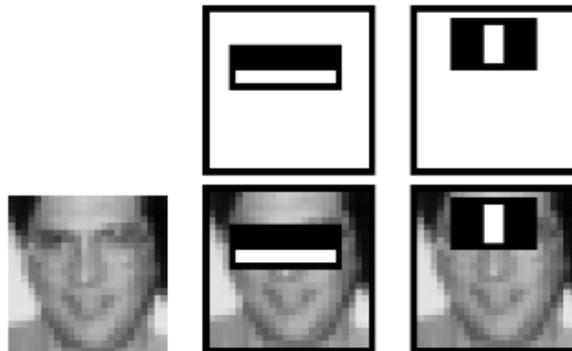
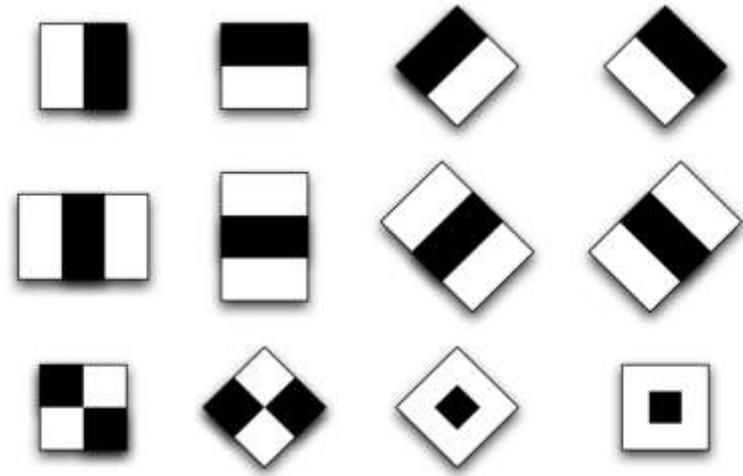
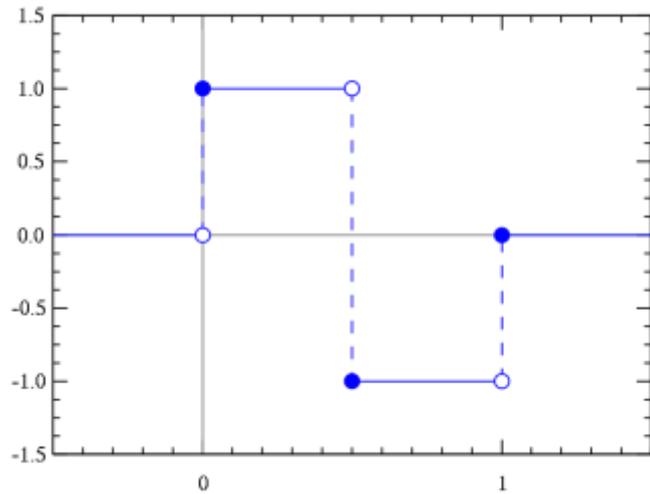
## Ondelettes

$$g(x, y, \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right) \quad \begin{array}{l} x' = x \cos(\theta) + y \sin(\theta) \\ y' = -x \sin(\theta) + y \cos(\theta) \end{array}$$



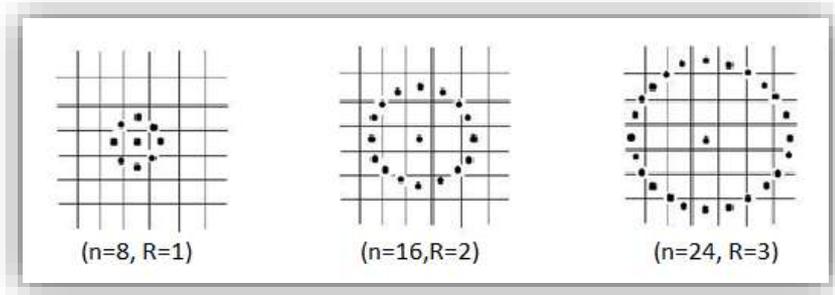
# Outils d'analyse

## Descripteurs de Haar

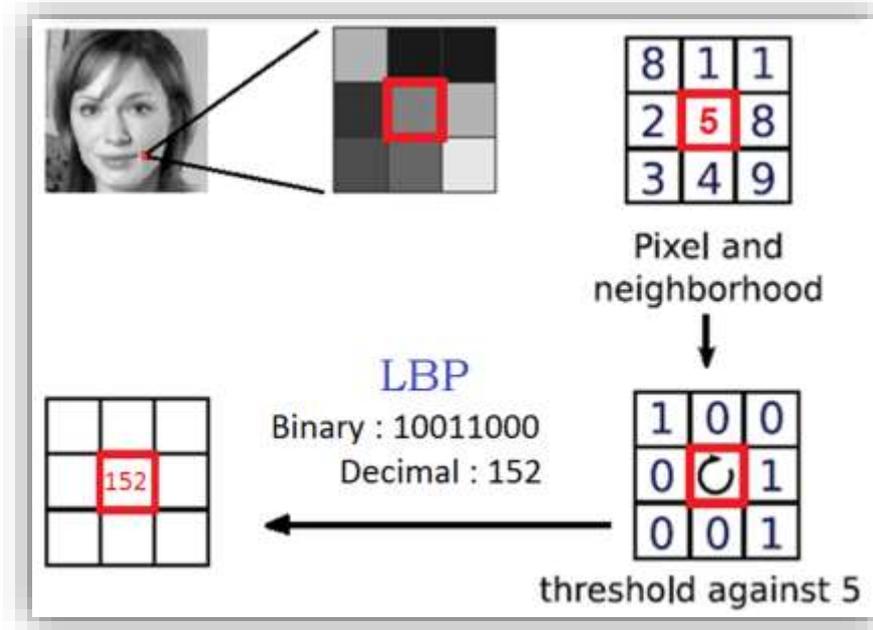


# Outils d'analyse

## Local Binary Patterns (LBP)



$$LBP(x, y) = \sum_{n=0}^7 2^n s(i_n - i_c)$$



# Outils d'analyse

---

Mais aussi...

- Transformée / moments de Fourier
- SIFT
- SURF
- *etc !*

# Analyse et modélisation de visages

---

## Plan

- Introduction
- Outils d'analyse
- Phase de détection
- Phase d'alignement et de suivi
- Du modèle rigide au modèle déformable

# Phase de détection

---

## Le Boosting : un modèle d'apprentissage PAC\*

- Il est « facile » d'avoir des règles qui sont « généralement » justes  
(*SI un objet est entouré de ciel ALORS c'est un avion*)
- Il est difficile d'avoir des règles toujours justes



\* « Probablement Approximativement Correct »

# Phase de détection

---

## Le Boosting : un modèle d'apprentissage PAC

- Des classifieurs « faibles », capables de reconnaître deux classes au moins aussi bien que le hasard ne le ferait, c'est-à-dire qui ne se trompent pas plus d'une fois sur deux en moyenne (ex : 55% de réponses correctes)
- Par itérations successives, la connaissance de différents classifieurs faibles (on parlera de « cascades ») est ajoutée au classifieur final, nommé classifieur « fort », dont la précision sera très élevée (ex : 99%)
- Cette méthode peut optimiser les performances de nombreux algorithmes qui s'appuient sur des ensembles de classifieurs binaires
- **Boosting adaptatif (AdaBoost)** : un des algorithmes les plus utilisés, qui repose sur une sélection de classifieurs faibles en fonction des difficultés rencontrées sur les exemples de la base d'apprentissage

# Phase de détection

---

## AdaBoost

... un algorithme de référence en détection de visages (Viola-Jones)



# Phase de détection

## AdaBoost

- Soit un ensemble d'apprentissage, constitué de descriptions (uniques ou multiples)  $\mathbf{x}$  et d'étiquettes  $\mathbf{y}$  (ex : visage / pas visage) :

$$(x_1, y_1), \dots (x_i, y_i), \dots (x_m, y_m) \quad x_i \in X \quad y_i \in Y = \{-1, 1\}$$

- Initialiser des indices de pondération :

$$D_1(i) = \frac{1}{m}$$

- Trouver le classifieur faible  $h_t(\mathbf{x})$  (un seuil, une droite, un plan, un hyperplan...) qui minimise l'erreur :

$$\varepsilon_t = \sum_{i=1}^m D_t(i) [y_i \neq h(x_i)] \quad h_t = \arg \min_{h \in H} (\varepsilon_t) \quad t = 1 \dots T$$
$$\varepsilon_{\min, t} < 0.5$$

- Mettre à jour les pondérations :

$$D_{t+1}(i) = \frac{D_t(i) \cdot e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$$

# Phase de détection

---

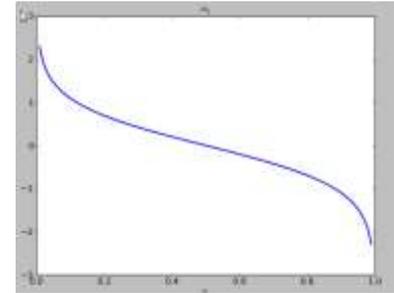
## AdaBoost

- Poids d'un classifieur faible :

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$$

- Classifieur final (prédicteur) :

$$H(x) = \text{sign} \left( \sum_{t=1}^T \alpha_t h_t(x) \right)$$

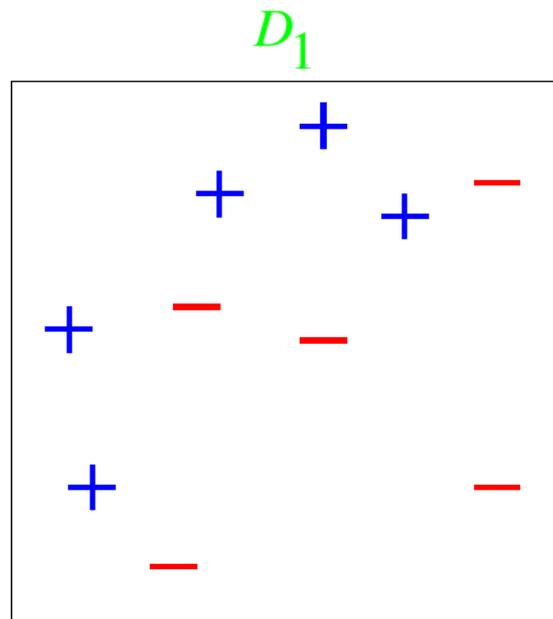


# Phase de détection

---

## AdaBoost

Ex : deux descripteurs par exemple  
des droites horizontales ou verticales comme classifieurs faibles

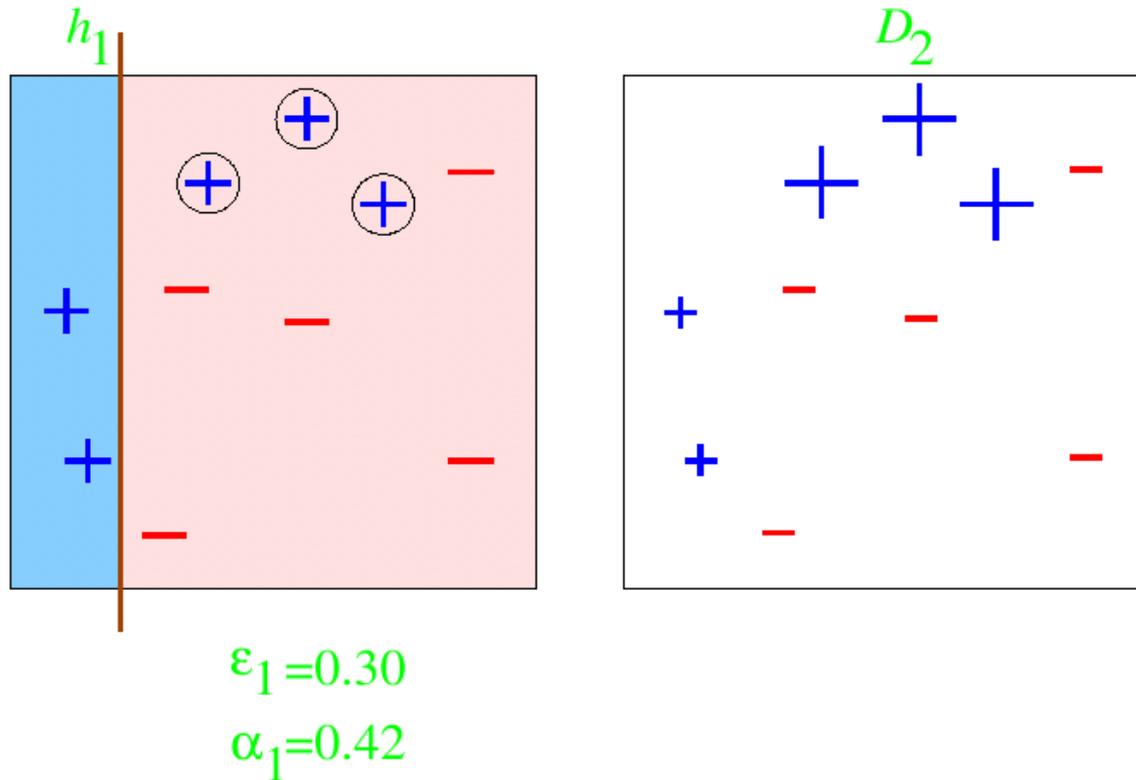


# Phase de détection

---

## AdaBoost

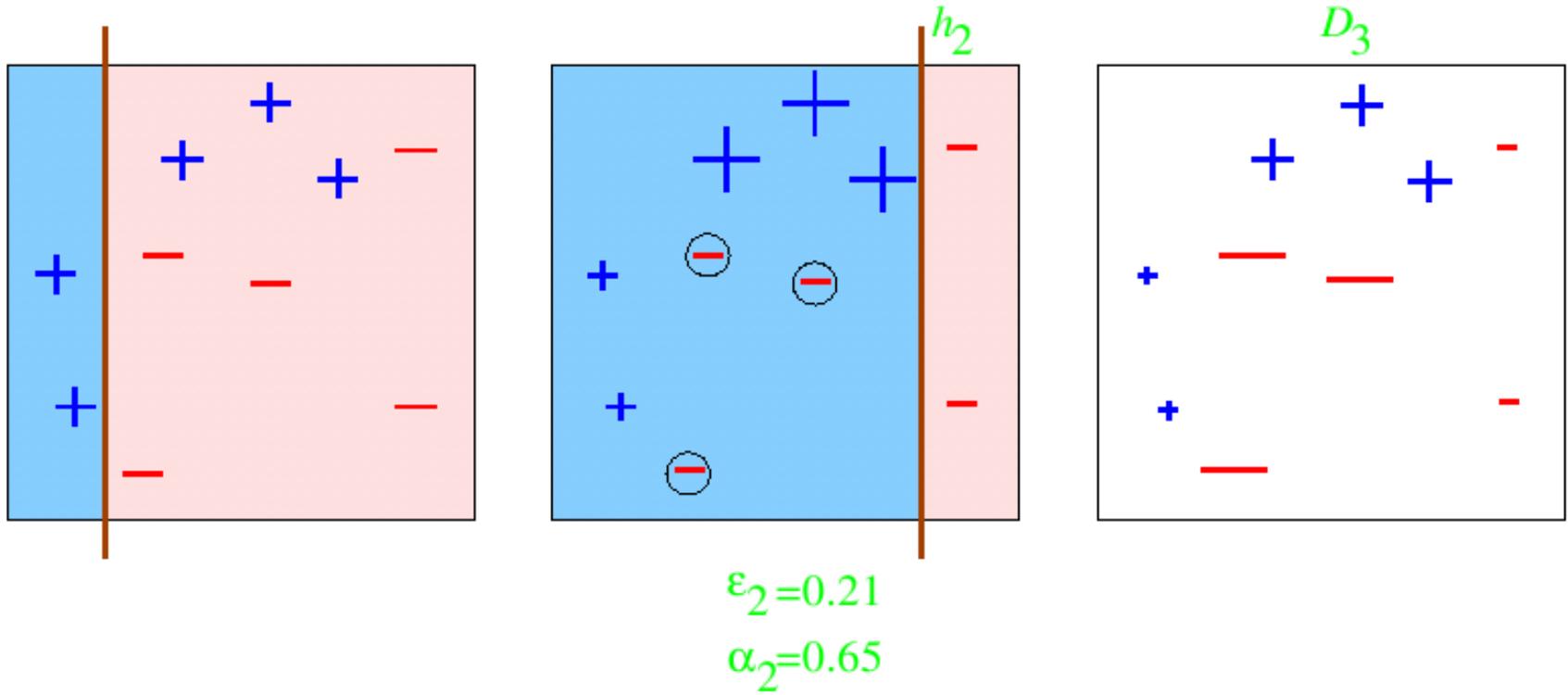
Ex : deux descripteurs par exemple  
des droites horizontales ou verticales comme classifieurs faibles



# Phase de détection

## AdaBoost

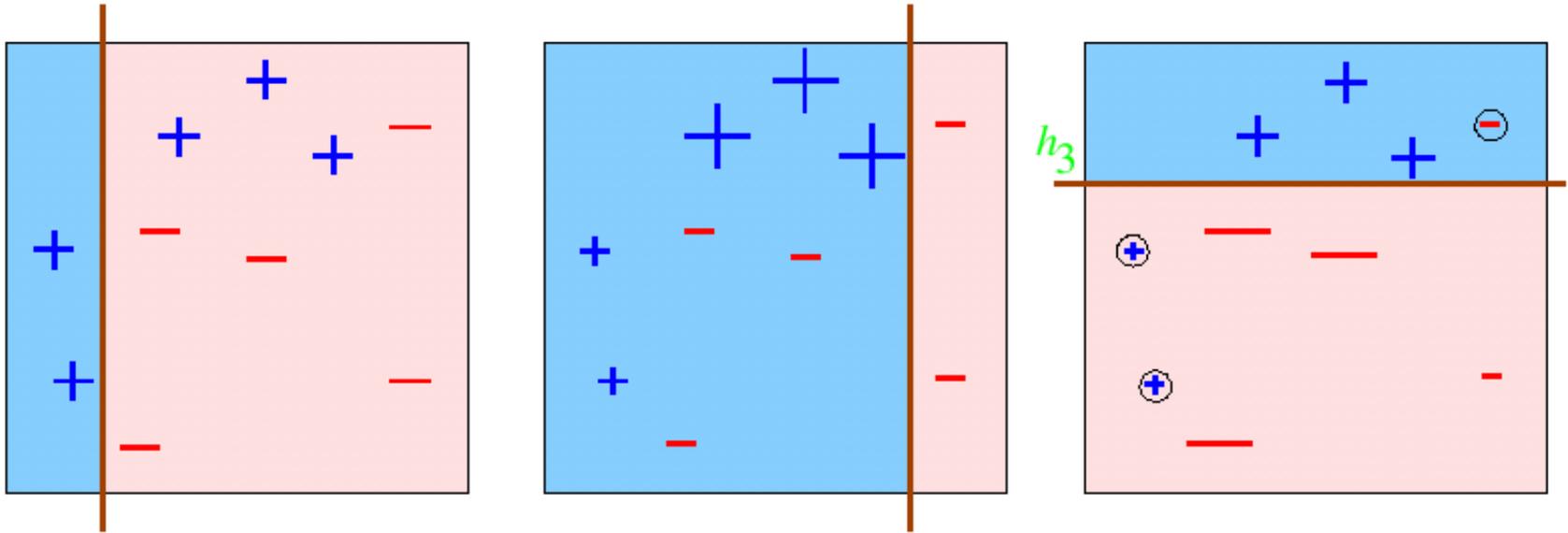
Ex : deux descripteurs par exemple  
des droites horizontales ou verticales comme classifieurs faibles



# Phase de détection

## AdaBoost

Ex : deux descripteurs par exemple  
des droites horizontales ou verticales comme classifieurs faibles



$$\epsilon_3 = 0.14$$

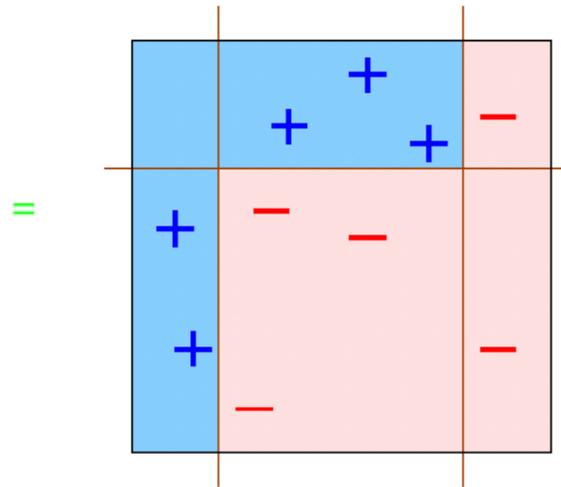
$$\alpha_3 = 0.92$$

# Phase de détection

## AdaBoost

Ex : deux descripteurs par exemple  
des droites horizontales ou verticales comme classifieurs faibles

$$H_{\text{final}} = \text{sign} \left( 0.42 \left[ \begin{array}{|c|c|} \hline \text{blue} & \text{red} \\ \hline \end{array} \right] + 0.65 \left[ \begin{array}{|c|c|} \hline \text{blue} & \text{red} \\ \hline \end{array} \right] + 0.92 \left[ \begin{array}{|c|c|} \hline \text{blue} & \text{red} \\ \hline \end{array} \right] \right)$$

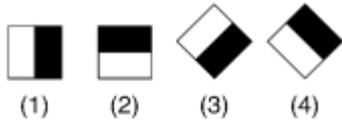


# Phase de détection

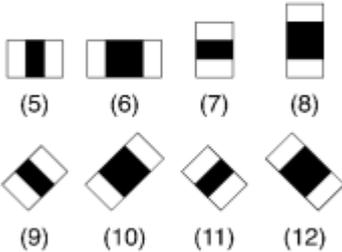
## AdaBoost

Descripteurs pour la détection de visage : Haar (algorithme de Viola-Jones)

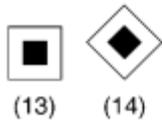
Caractéristiques de bord



Caractéristiques de ligne



Caractéristiques centre-pourtour



# Phase de détection

---

## AdaBoost

Descripteurs pour la détection de visage (algorithme de Viola-Jones)

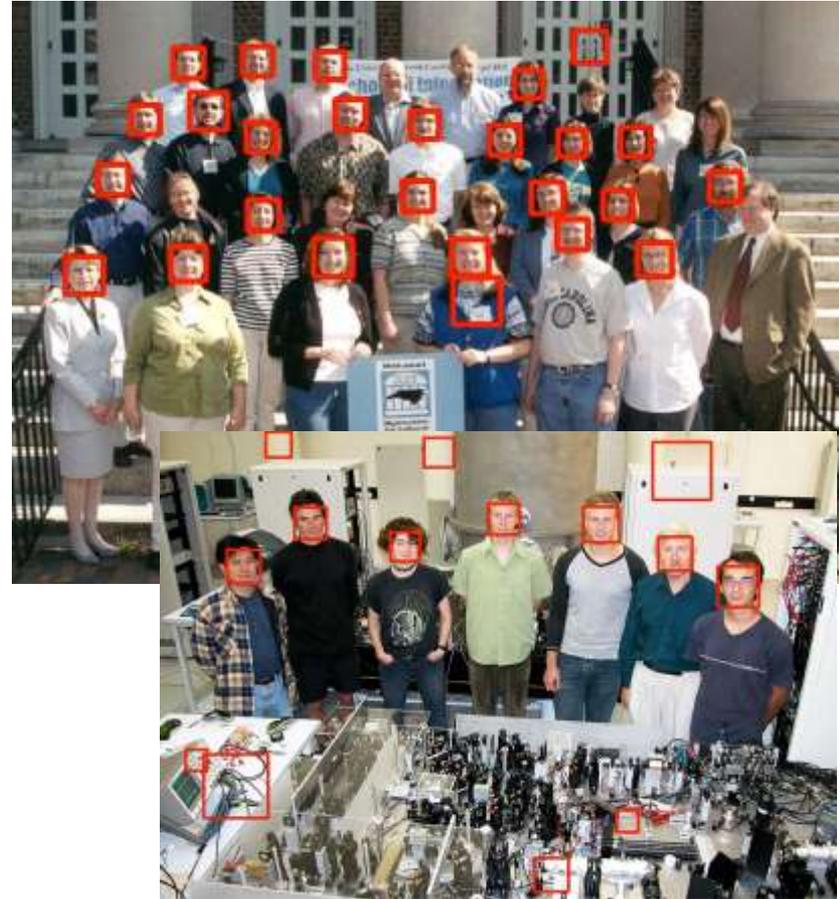
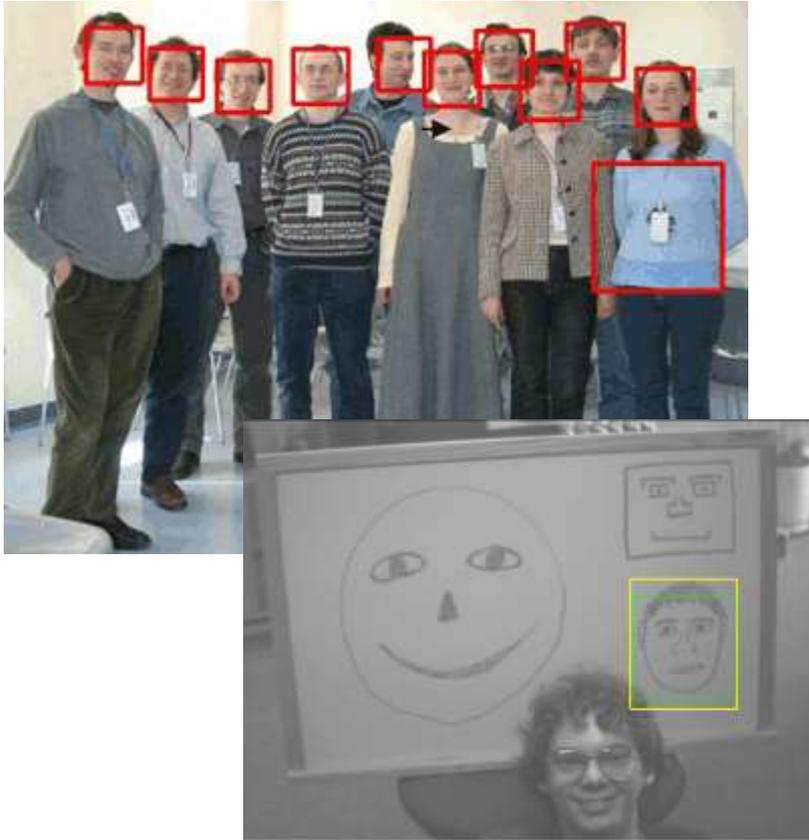


# Phase de détection

---

## AdaBoost

Un algorithme parfait ?

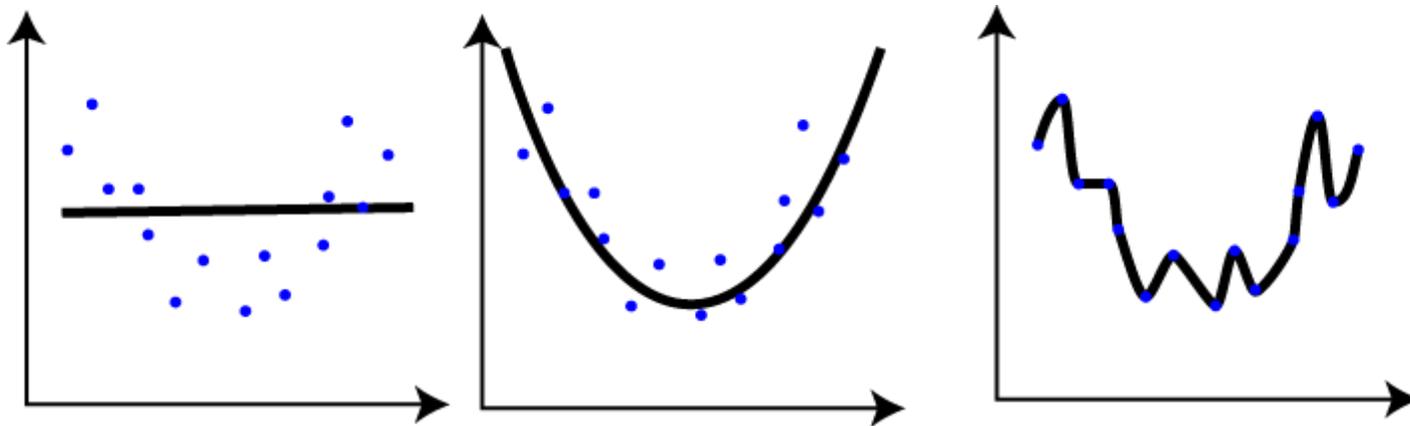


# Phase de détection

---

Un problème récurrent : le choix des exemples !

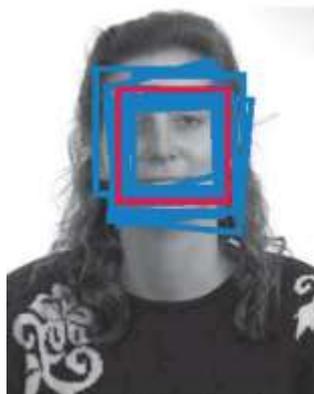
- Sous-apprentissage : les performances de l'algorithme sont médiocres
- Sur-apprentissage : l'algorithme semble très performant mais ne généralise pas suffisamment pour de nouveaux exemples
- Solution : choix d'une base d'apprentissage et d'une base de test
- En détection de visages : généricité vs. précision



# Phase de détection

---

## Détection, suivi, et alignement



subject  $i$

Random  
Crop →



image set  $X_i$

# Analyse et modélisation de visages

---

## Plan

- Introduction
- Outils d'analyse
- Phase de détection
- Phase d'alignement et de suivi
- Du modèle rigide au modèle déformable

# Phase d'alignement et de suivi

---

## Notion de flot optique

Le **flot optique** est un champ de déplacement visuel qui permet d'expliquer des variations dans une image animée en terme de déplacement de points images. L'hypothèse que l'on fait afin d'en tirer une équation simple est que l'intensité d'un point  $(x,y)$  ne dépend que du déplacement d'un objet le long de ce champ durant un temps correspondant au temps d'acquisition entre deux images :

$$I(x, y, t - 1) = I(x + u, y + v, t) \Rightarrow -\frac{\partial I}{\partial t} \approx \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} = \mathbf{v}^T \nabla I$$

# Phase d'alignement et de suivi

---

## Notion de flot optique

On cherche donc à trouver le vecteur  $(u, v)$  qui minimise une erreur pour un ou plusieurs points de l'image (algorithme de Lucas-Kanade) :

$$\xi(\mathbf{v}) = \sum_x \sum_y \left( \underbrace{I(x, y, t-1) - I(x+u, y+v, t)}_{-\delta I} \right)^2$$

Un développement en série de Taylor nous donne la solution :

$$\hat{\mathbf{v}} = \mathbf{H}^{-1} \mathbf{b}$$

Avec :

$$\mathbf{H} = \sum_x \sum_y \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

(Hessian)

$$\mathbf{b} = \sum_x \sum_y \begin{bmatrix} \delta I I_x \\ \delta I I_y \end{bmatrix}$$

(Steepest Descent)

# Phase d'alignement et de suivi

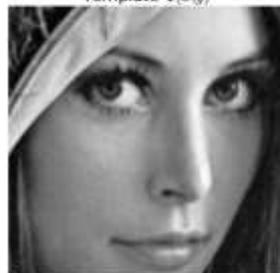
## Notion de flot optique

**ATTENTION** : l'utilisation de développements en série de Taylor nous impose l'hypothèse d'une champ de déplacement suffisamment petit ! En pratique, la solution sera estimée par un processus itératif de résolution numérique. On la suppose optimale lorsque ce processus a convergé

$$\hat{\mathbf{v}}^{n+1} = \hat{\mathbf{v}}^n + h\mathbf{H}^{n-1}\mathbf{b}^n$$



Template  $T(x,y)$



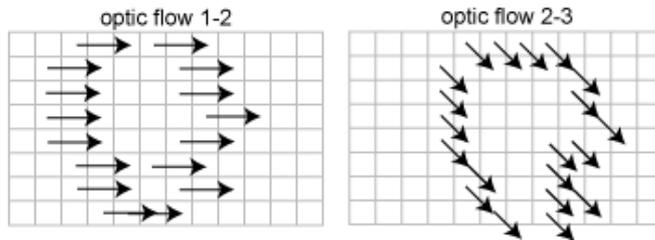
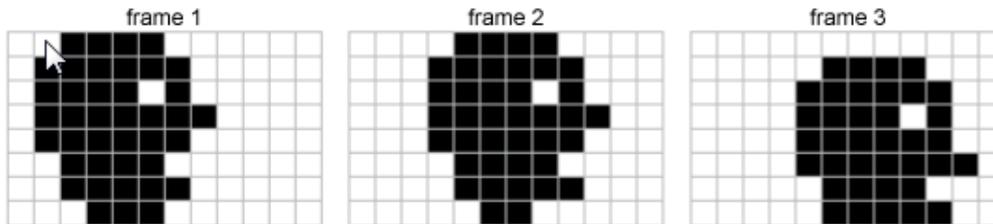
Matched  $I|W(x,y)$



# Phase d'alignement et de suivi

## Notion de flot optique

Un unique champ de déplacement en tout point d'un objet ???



# Phase d'alignement et de suivi

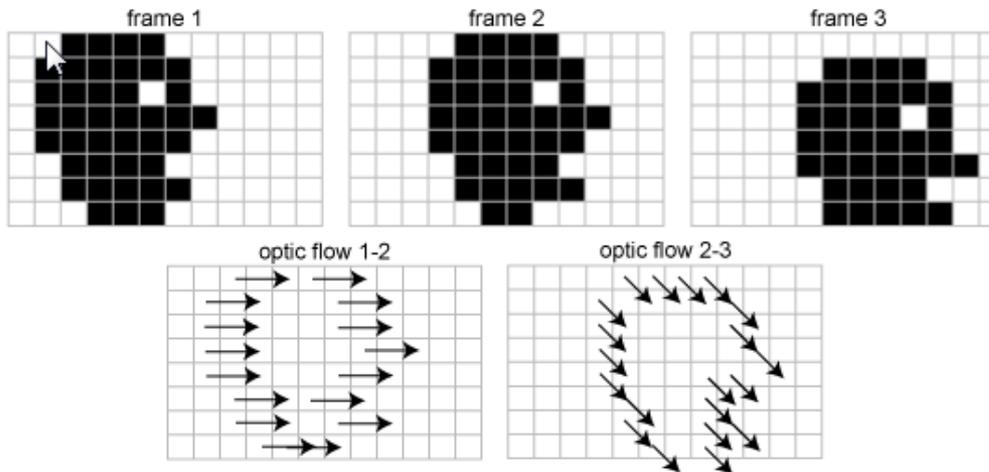
## Formalisation de Matthews-Baker (Lucas-Kanade 20 ans après)

On reprend :

$$\xi(\mathbf{W}) = \sum_x \sum_y (I^t(x, y) - I^{t-1}(\mathbf{W}(x, y)))^2$$

Exemple :

$$\mathbf{W}(x, y) = \begin{bmatrix} x + u \\ y + v \end{bmatrix} = \mathbf{W}(x, y; \mathbf{v}) = \begin{bmatrix} 1 & 0 & u \\ 0 & 1 & v \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



# Phase d'alignement et de suivi

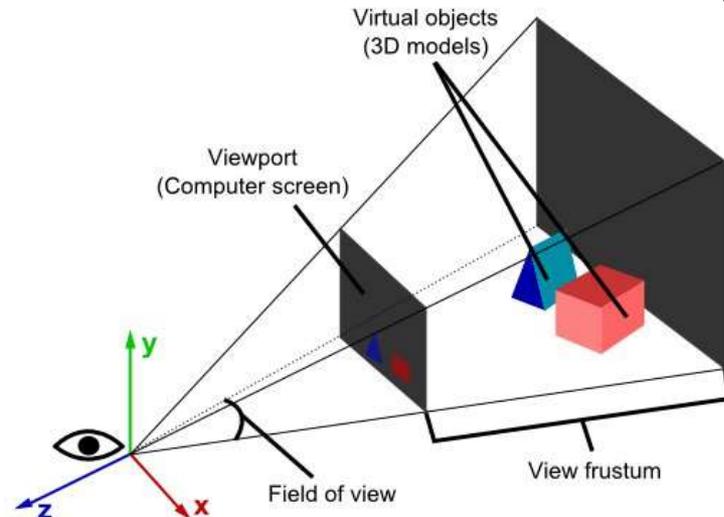
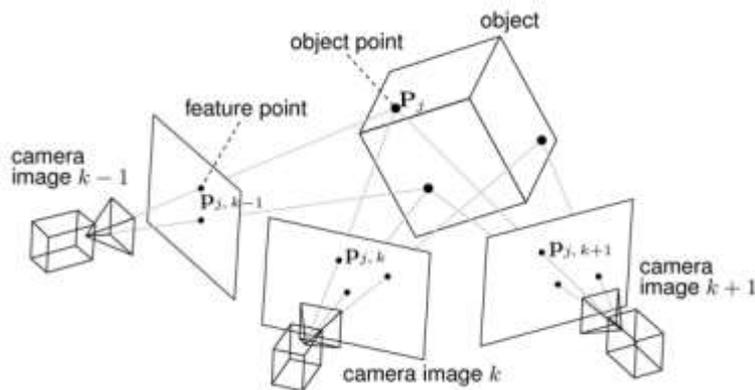
## Formalisation de Matthews-Baker (Lucas-Kanade 20 ans après)

On reprend :

$$\xi(\mathbf{W}) = \sum_x \sum_y \left( \underbrace{T(x, y)}_{\text{template}} - I \left( \underbrace{\mathbf{W}(x, y)}_{\text{warp}} \right) \right)^2$$

Autre exemple :

$$\mathbf{W}(x, y) = \begin{bmatrix} s_x x + w_z y + t_x \\ -w_z x + s_y y + t_y \end{bmatrix} = \mathbf{W}(x, y; \mathbf{p}) = \begin{bmatrix} 1 + p_1 & p_2 & p_3 \\ p_4 & 1 + p_5 & p_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



# Phase d'alignement et de suivi

---

Formalisation de Matthews-Baker (Lucas-Kanade 20 ans après)

On cherche donc à minimiser :

$$\xi(\mathbf{p}) = \sum_x \sum_y (T(x, y) - I(\mathbf{W}(x, y; \mathbf{p})))^2$$

Ou encore, de façon itérative :

$$\xi(\Delta\mathbf{p}) = \sum_x \sum_y (T(x, y) - I(\mathbf{W}(x, y; \mathbf{p} + \Delta\mathbf{p})))^2$$

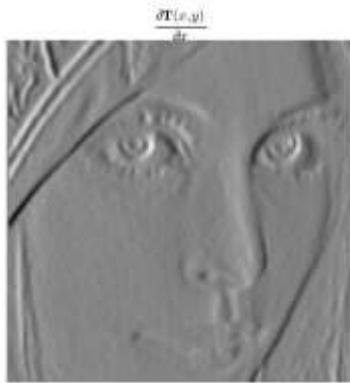
Avec des paramètres initiaux connus, par exemple (suivi d'objets) :

$$\mathbf{W}(x, y; \mathbf{0}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

# Phase d'alignement et de suivi

Formalisation de Matthews-Baker (Lucas-Kanade 20 ans après)

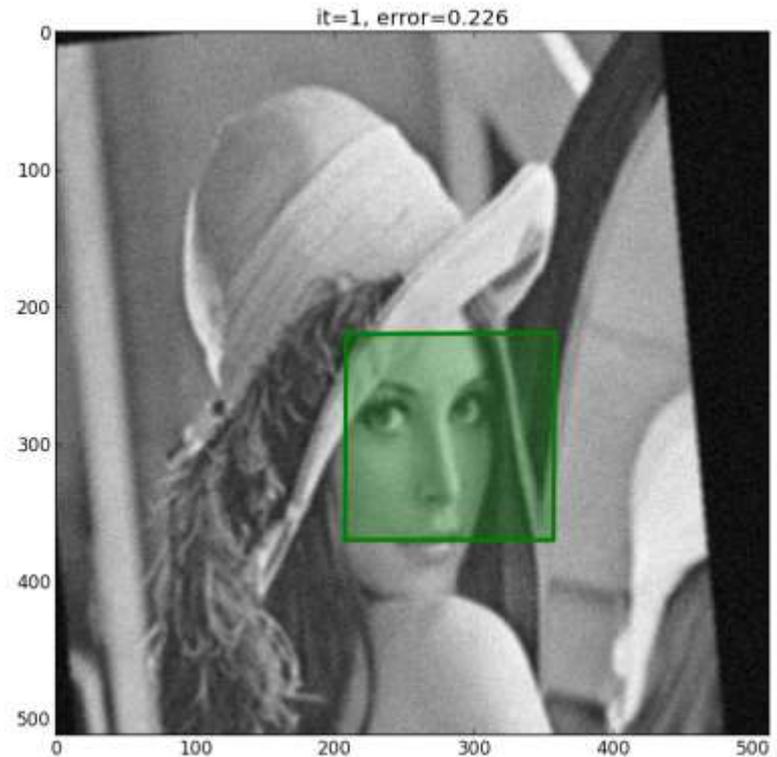
Exemple : transformée affine à 6 paramètres



Template  $T(x,y)$



Warp  $I(W(x,y))$



# Analyse et modélisation de visages

---

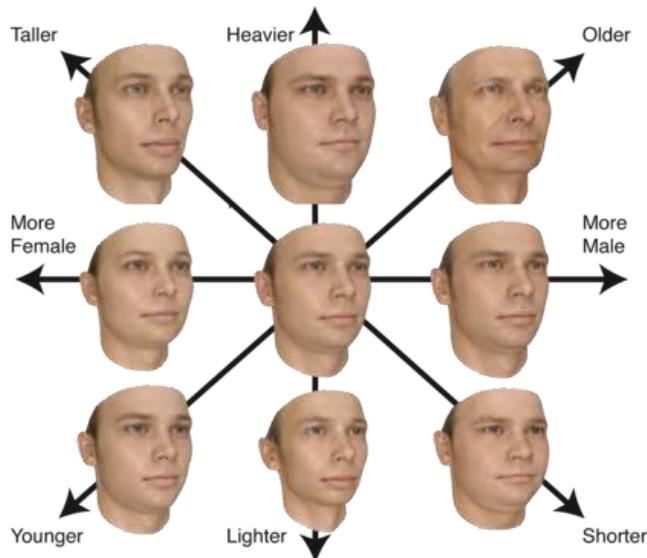
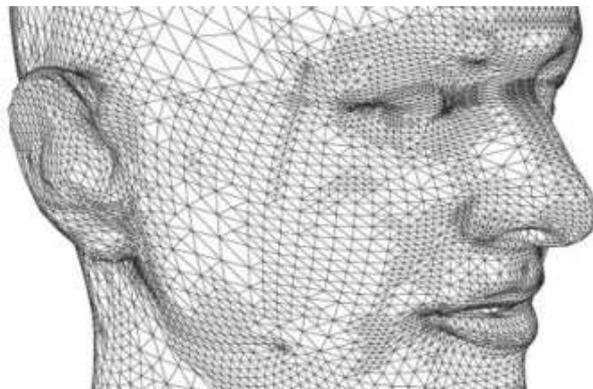
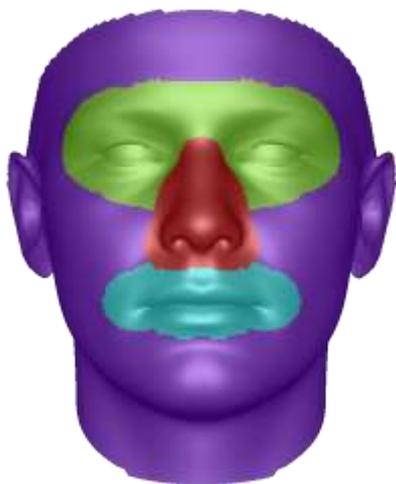
## Plan

- Introduction
- Outils d'analyse
- Phase de détection
- Phase d'alignement et de suivi
- Du modèle rigide au modèle déformable

# Du modèle rigide au modèle déformable

## Création d'un modèle de visage

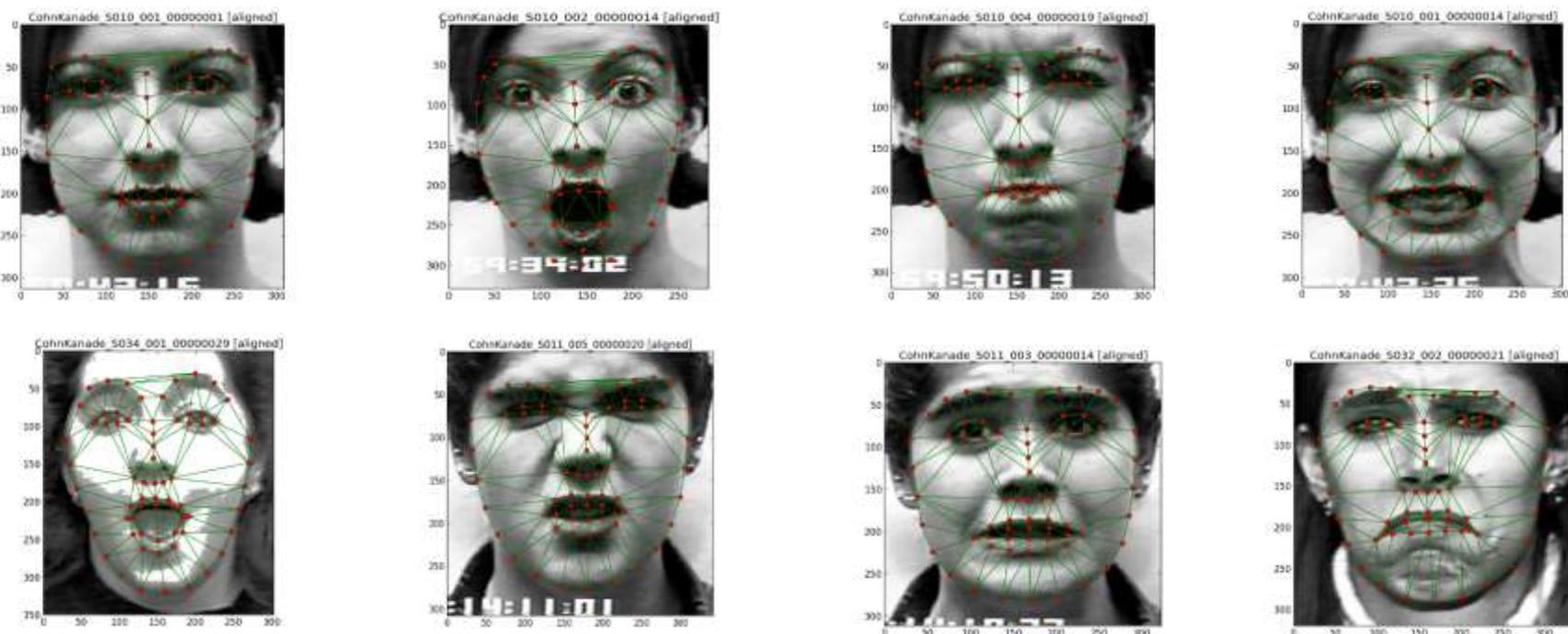
Dans le contexte de l'analyse de visage, nous souhaitons enrichir un modèle unique d'alignement par la prise en compte des changements d'identité, d'expression, ou encore d'illuminant. En vision artificielle, ces changements seront vus comme des variations locales de forme et/ou de texture.



# Du modèle rigide au modèle déformable

## Modèles actifs d'apparence

Ils permettent de synthétiser conjointement une image (« texture ») et la forme qui lui est associée. Un modèle AAM\* est construit par analyse en composantes principales (ACP) de données de texture et de forme alignées :



\*AAM : Active Appearance Model

# Du modèle rigide au modèle déformable

---

## Modèles actifs d'apparence

Ils permettent de synthétiser conjointement une image (« texture ») et la forme qui lui est associée. Un modèle AAM\* est construit par analyse en composantes principales (ACP) de données de texture et de forme alignées :



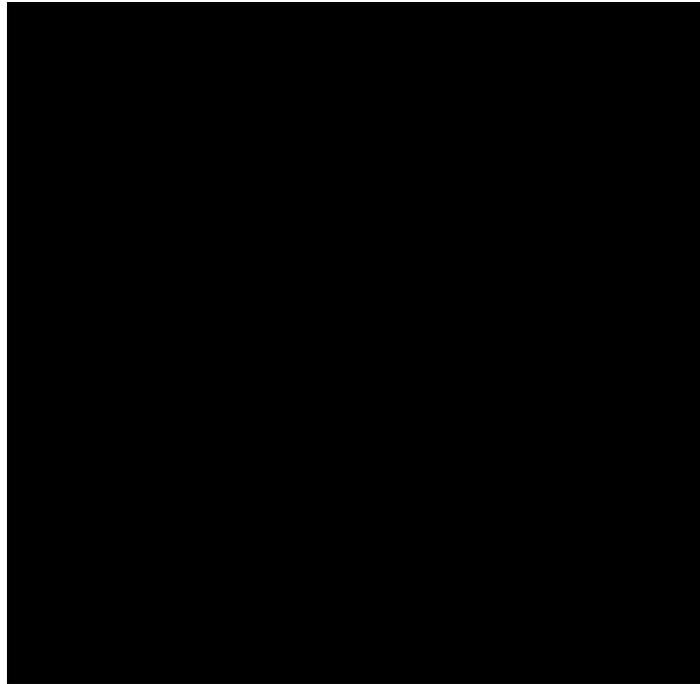
\*AAM : Active Appearance Model

# Du modèle rigide au modèle déformable

---

## Modèles actifs d'apparence

Ils permettent de synthétiser conjointement une image (« texture ») et la forme qui lui est associée. Un modèle AAM\* est construit par analyse en composantes principales (ACP) de données de texture et de forme alignées :



\*AAM : Active Appearance Model

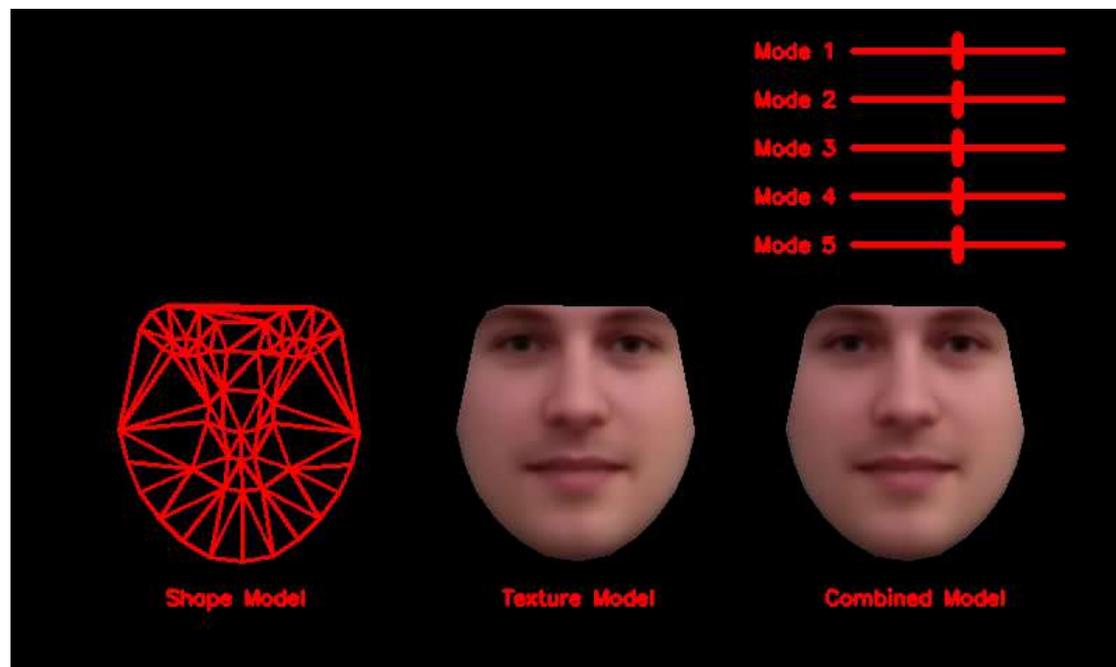
# Du modèle rigide au modèle déformable

## Modèles actifs d'apparence

Création d'un modèle de visage par association d'une forme et d'une texture :

$$\mathbf{s} = \mathbf{s}_0 + \sum_{p=1}^{P_s} \Phi_p \mathbf{s}_p$$

$$\mathbf{g} = \mathbf{g}_0 + \sum_{p=1}^{P_g} \Phi_p \mathbf{g}_p$$

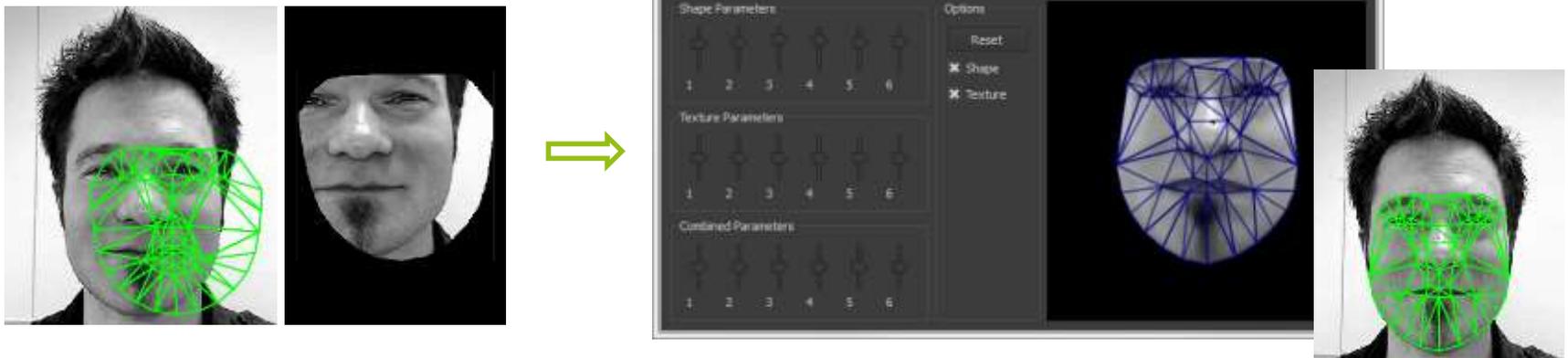


# Du modèle rigide au modèle déformable

## Modèles actifs d'apparence

Alignement du modèle sur des données observées : estimer les paramètres de déformation à partir des données observées (texture) depuis une position initiale

- Par régression linéaire (Cootes)
- Par descente de gradient (Matthews-Baker)
- Par apprentissage automatique (Cootes, Saragih...)



# Du modèle rigide au modèle déformable

---

Modèles actifs d'apparence

Alignement du modèle sur des données observées



# Du modèle rigide au modèle déformable

---

Modèles actifs d'apparence

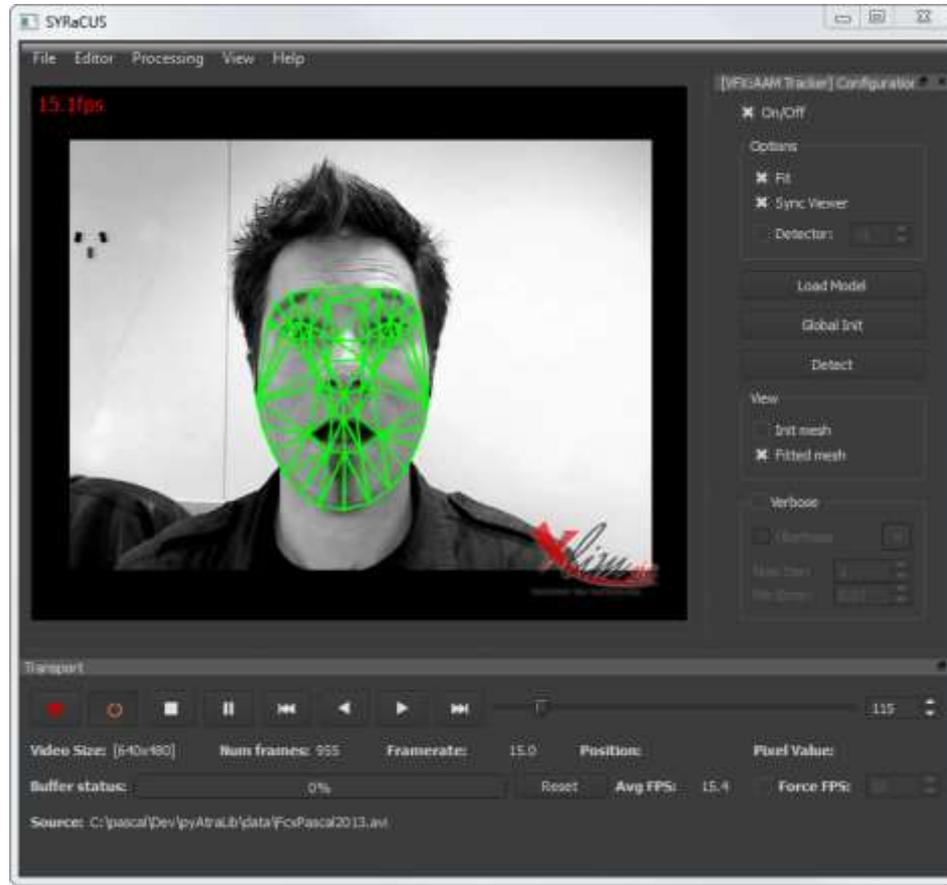
Alignement du modèle sur des données observées



# Du modèle rigide au modèle déformable

Modèles actifs d'apparence

Alignement du modèle sur des données observées



# Analyse et modélisation de visages

---

Merci ! 😊

# Analyse et modélisation de visages

---

## Plan

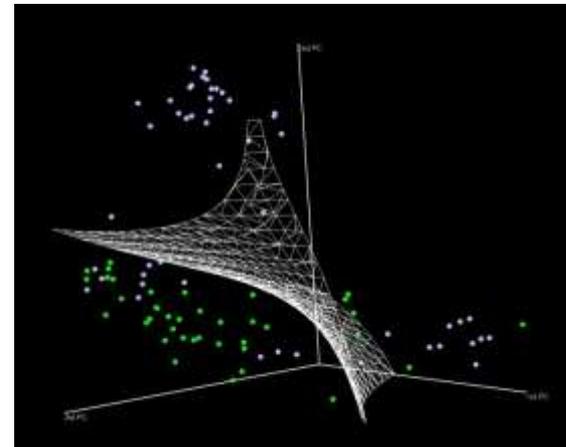
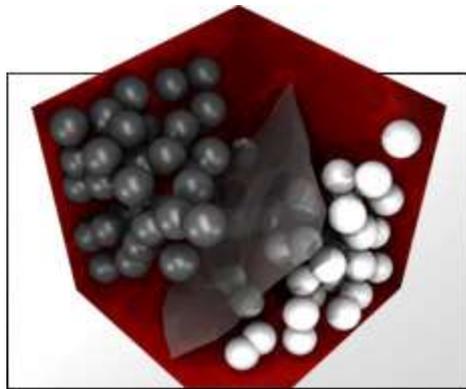
- Introduction
- Outils d'analyse
- Phase de détection
- Phase d'alignement et de suivi
- Du modèle rigide au modèle déformable
- [BONUS] Reconnaissance et classification

# Reconnaissance et classification

---

## Machines à vecteurs de support (SVM\*)

- Apprentissage supervisé
- Généralisation des classifieurs linéaires
- Séparateurs à vaste marge
- Possibilité de définir une marge souple
- Classification linéaire ou non-linéaire
- Données de grandes dimensions



\* SVM : Support Vector Machines

# Reconnaissance et classification

---

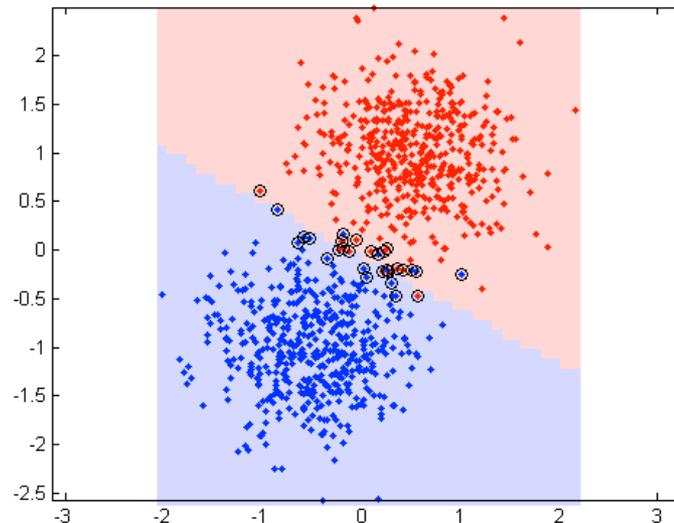
## Machines à vecteurs de support

- Recherche d'une fonction discriminante linéaire :

$$y = \text{sign}(h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0) \quad \mathbf{x} = (x_1, x_2 \dots x_p)^T \quad y \in \{-1, 1\}$$

- Hyperplan séparateur / frontière de décision :

$$h(\mathbf{x}) = 0 \quad y_k h(\mathbf{x}_k) = y_k (\mathbf{w}^T \mathbf{x}_k + w_0) \geq 0$$

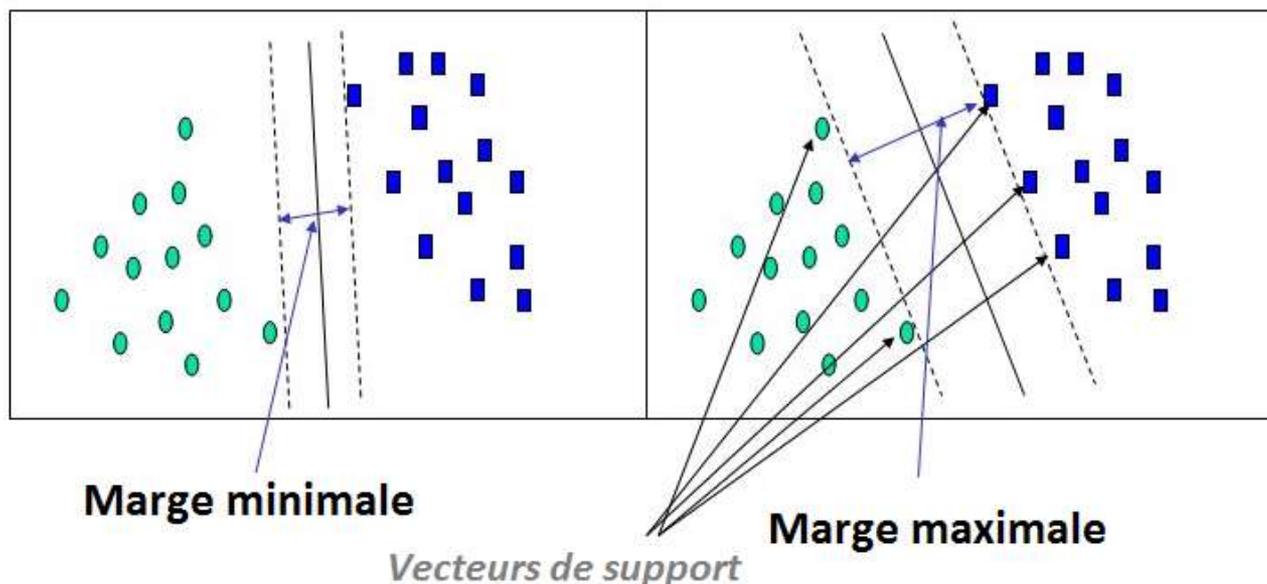


# Reconnaissance et classification

## Machines à vecteurs de support

- Notion de marge optimale

Le classifieur le plus robuste est celui défini par l'hyperplan le plus éloigné des échantillons. Cela revient à maximiser la distance des points à l'hyperplan. La « marge » est définie par des « vecteurs de support »



$$\begin{cases} \min \frac{1}{2} \|\mathbf{w}\|^2 \\ \forall k \quad y_k (\mathbf{w}^T \mathbf{x}_k + w_0) \geq 1 \end{cases}$$

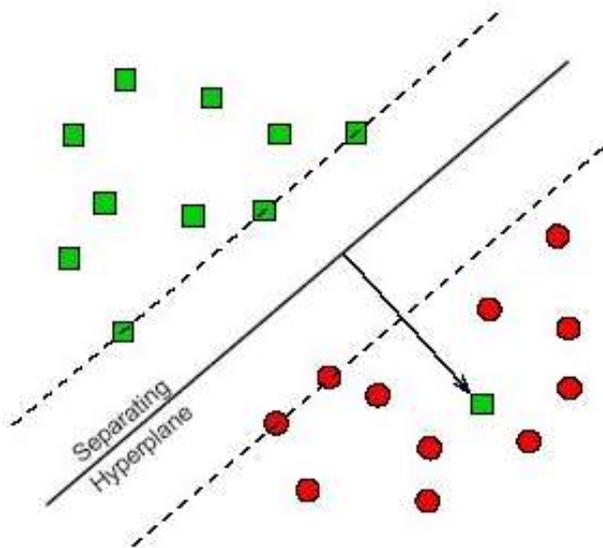
# Reconnaissance et classification

---

## Machines à vecteurs de support

- Cas non séparable : marge souple

Dans le cas non-séparable, on va relâcher les contraintes de marge en introduisant des variables « ressort »

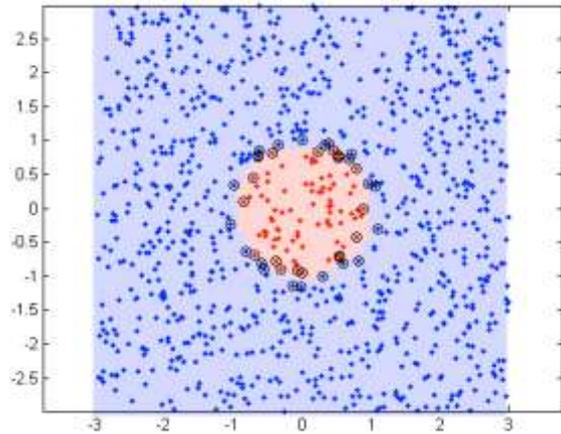


$$y_k (\mathbf{w}^T \mathbf{x}_k + w_0) \geq 1 - \xi_k \quad \xi_k \geq 0$$

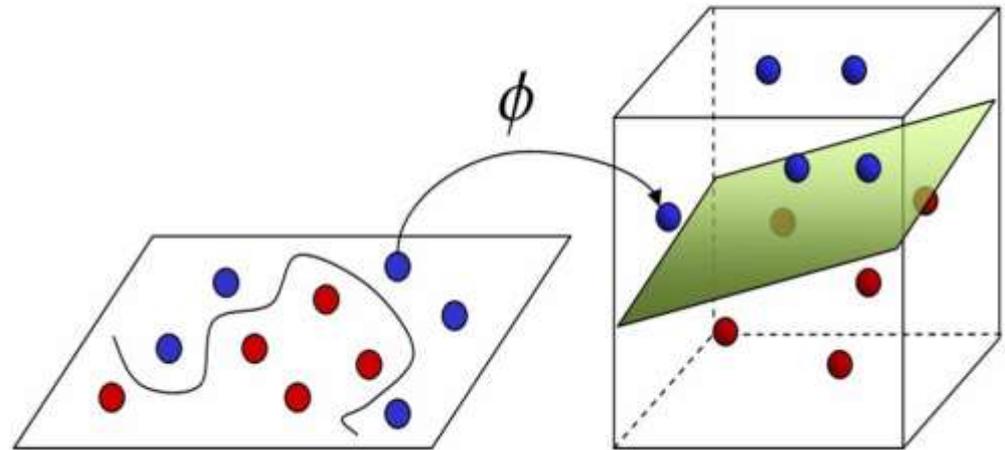
# Reconnaissance et classification

## Machines à vecteurs de support

- Cas non séparable : astuce du noyau (« kernel trick »)



$$h(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + w_0$$

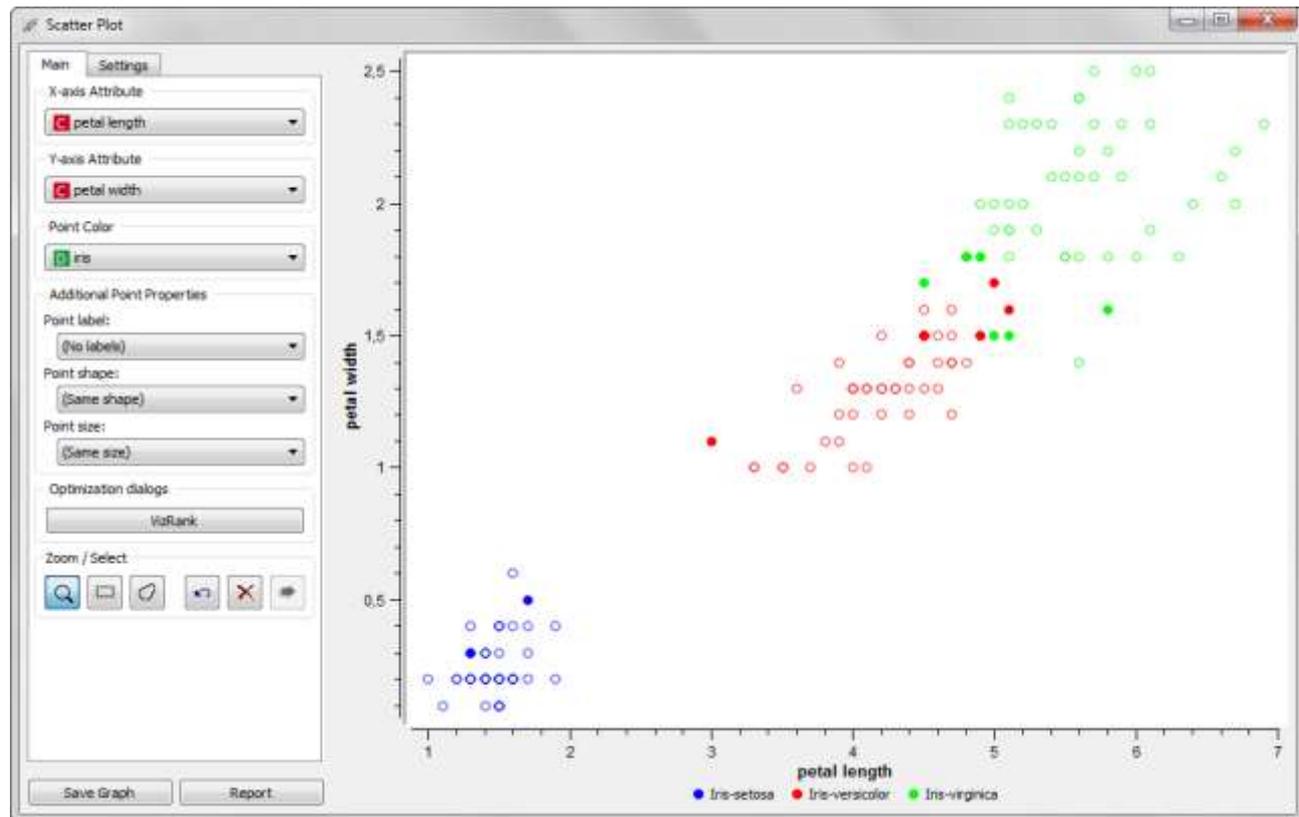


Définition d'un espace de redescription

# Reconnaissance et classification

## Machines à vecteurs de support

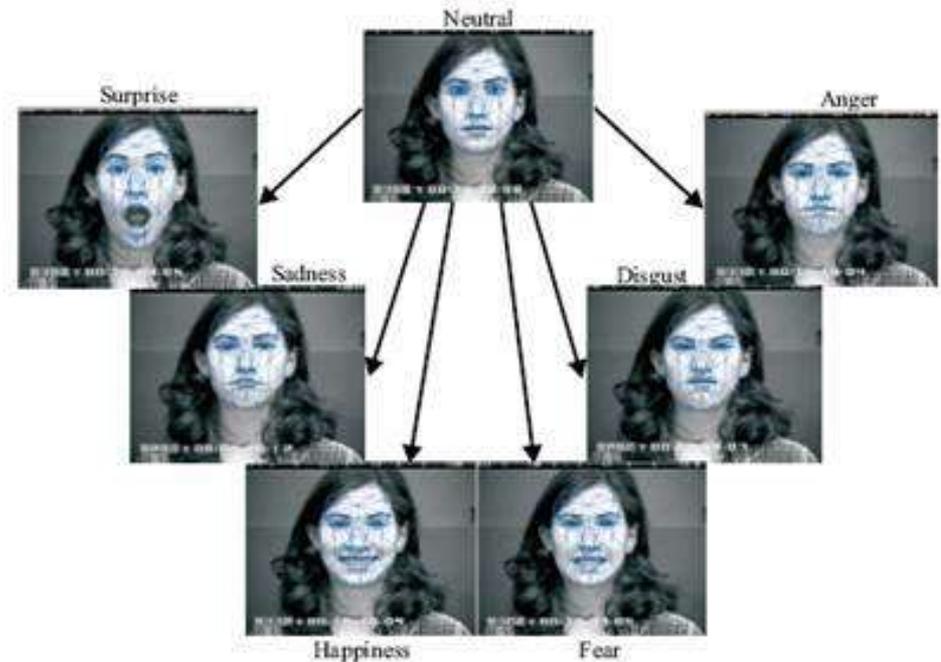
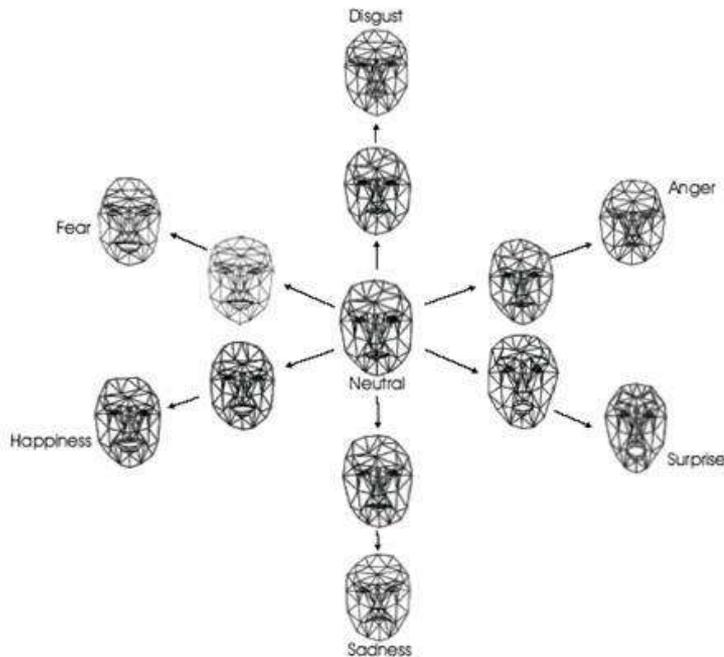
- Exemple : classification d'iris



# Reconnaissance et classification

## Machines à vecteurs de support

- Exemple : reconnaissance d'expressions faciales



# Reconnaissance et classification

## Machines à vecteurs de support

- Exemple : reconnaissance d'expressions faciales

