Analyse de la vidéo

Chapitre 4.1 - La modélisation pour le suivi d'objet

10 mars 2015

Plan de la présentation

Introduction

Applications du suivi d'objet :

- **Reconnaissance par le mouvement** : Classification d'objets basé sur le comportement
- **Surveillance automatique** : Détection de dangers ou évènements suspects, traffic automobile
- **Interaction visuelle** : Interface humain-machine (ex : kinect)
- **Navigation automatique** : Plannification de trajet par la vidéo (ex : google car)

Introduction

Le suivi d'objet :

- On veut estimer la trajectoire d'un objet dans le plan image alors qu'il bouge dans une scène
  - Un moniteur d'objet (tracker) assigne un étiquette à l'objet suivi dans toutes les images de la vidéo
  - Un moniteur peut aussi fournir des informations de forme, d'action, d'orientation, de mémoire temporelle, etc.
- La tâche se sépare en deux catégories :
  - La modélisation de l'objet à suivre ;
  - L'algorithme de suivi basé sur le modèle.

Introduction

Peut être complexe :

- Basé sur le mouvement apparent ;
- Bruit, changement d'illumination ;
- Déformation d'objets, occlusion ;
- Doit être en temps réel.

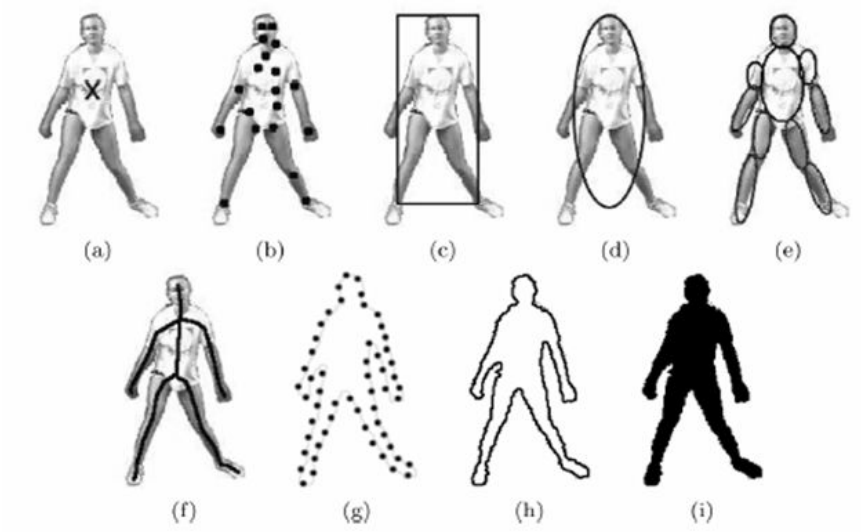
Mais on peut imposer des contraintes :

- Petits mouvements ;
- Changements de vitesses non-brusques ;
- Connaissance *a priori* de notre objet.

Représentation des objets

On classe les types de modélisation en deux classes distinctes :

- **Modèle géométrique (de forme)** :



- **Modèle iconique (d'apparence)** : Description du contenu en terme de couleur, de texture, de mouvement, ...

Plan de la présentation

Modèle d'approximation géométriques

Approximation par formes géométriques

- Les objets sont représentés par des triangles, élipses, carrés, ...
- Le mouvement est considéré constant pour tout l'objet et est facilement paramétrisable (affine, projection, ...)
- Peut être utilisé pour suivre des objets déformables, même s'il n'est pas adapté pour ça.

Modèle d'approximation géométriques

Modèle de contours/silhouettes Le modèle des contours/silhouettes est très utile pour décrire l'**allure du mouvement d'objets non rigides**.

- *Contour* d'un objet $\rightarrow$ Frontière de l'objet
- *Silhouette* d'un objet $\rightarrow$ Intérieur de l'objet

$\rightarrow$ Permet s'adapter aux changements de la taille des différentes composantes des objets suivis.

Modèle articulé

Modèle articulé

Ce modèle permet d'associer une primitive géométrique pour chaque membre d'un objet articulé en mouvement.

La relation entre les différentes parties, ainsi que les articulations les reliant, sont gouvernées par des lois de la mécanique du mouvement.

Modèle squelettique

Modèle squelettique

Le squelette d'un objet peut être généré en appliquant la transformation de l'axe médian pour la silhouette de l'objet.

$\rightarrow$ Modèle intermédiaire entre le modèle des contours/silhouettes et le modèle articulé.$\rightarrow$ Peut être analysé pour la reconnaissance des actions d'objets, et aussi pour la synthèse du mouvement.

Exemple de modélisation géométrique : Kinect

Kinect

La kinect permet d'obtenir facilement un carte de profondeur de la scène à l'aide d'une caméra à infra-rouge.

La carte de profondeur nous permet d'obtenir la silhouette des joueurs (6 personnes détecté, 2 que l'on peut suivre).

Exemple de modélisation géométrique : Kinect

Kinect

La kinect permet d'obtenir facilement un carte de profondeur de la scène à l'aide d'une caméra à infra-rouge.

La carte de profondeur nous permet d'obtenir la silhouette des joueurs (6 personnes détectés, 2 que l'on peut suivre).

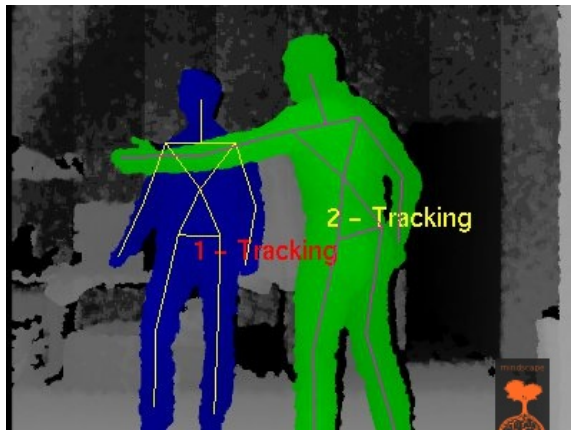
Exemple de modélisation géométrique : Kinect

Figure 1. An intensity image and grayscale-coded depth image of a person. Since the background is similar color to the person's shirt, it is difficult for the computer to segment the person from the background using the intensity image. The segmentation is trivial using the depth image.

Exemple de modélisation géométrique : Kinect

Kinect

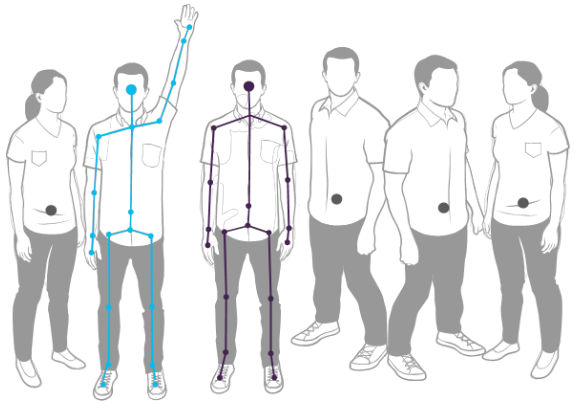
Des 6 joueurs détectables, on peut tracker que deux joueurs en simultané. Pour ce faire on utilise un hybride du modèle articulé et modèle squelettique.



Exemple de modélisation géométrique : Kinect

Kinect

Des 6 joueurs détectables, on peut tracker que deux joueurs en simultané. Pour ce faire on utilise un hybride du modèle articulé et modèle squelettique.



Exemple de modélisation géométrique : Kinect

Kinect

On utilise le modèle articulé afin de pouvoir classifier les mouvements de chaque joueur : saut, bras levé, accroupi, etc.

On observe principalement l'angle entre les articulations.

Représentation par les points d'intérêts

Points d'intérêt
Les points d'intérêt peuvent être des coins d'objets, des points de fuites, etc. Ces points peuvent être détectés en utilisant, par exemple, le **détecteur de Harris** ou **SIFT**.

En général, pour chaque frame :

1. On détecte les points d'intérêt de la scène ;
2. On fait une mise en correspondance frame par frame en des points de l'objet avec les points de la scène.

Avantages/Inconvénients des points d'intérêts

Avantages :

- Sélection partielle de l'objet.
- Robustesse aux transformations de l'image ou de l'objet.
- Robustesse aux occlusions.

Inconvénients :

- Temps de calcul de l'extraction de point.
- Temps de calcul de la mise en correspondance.

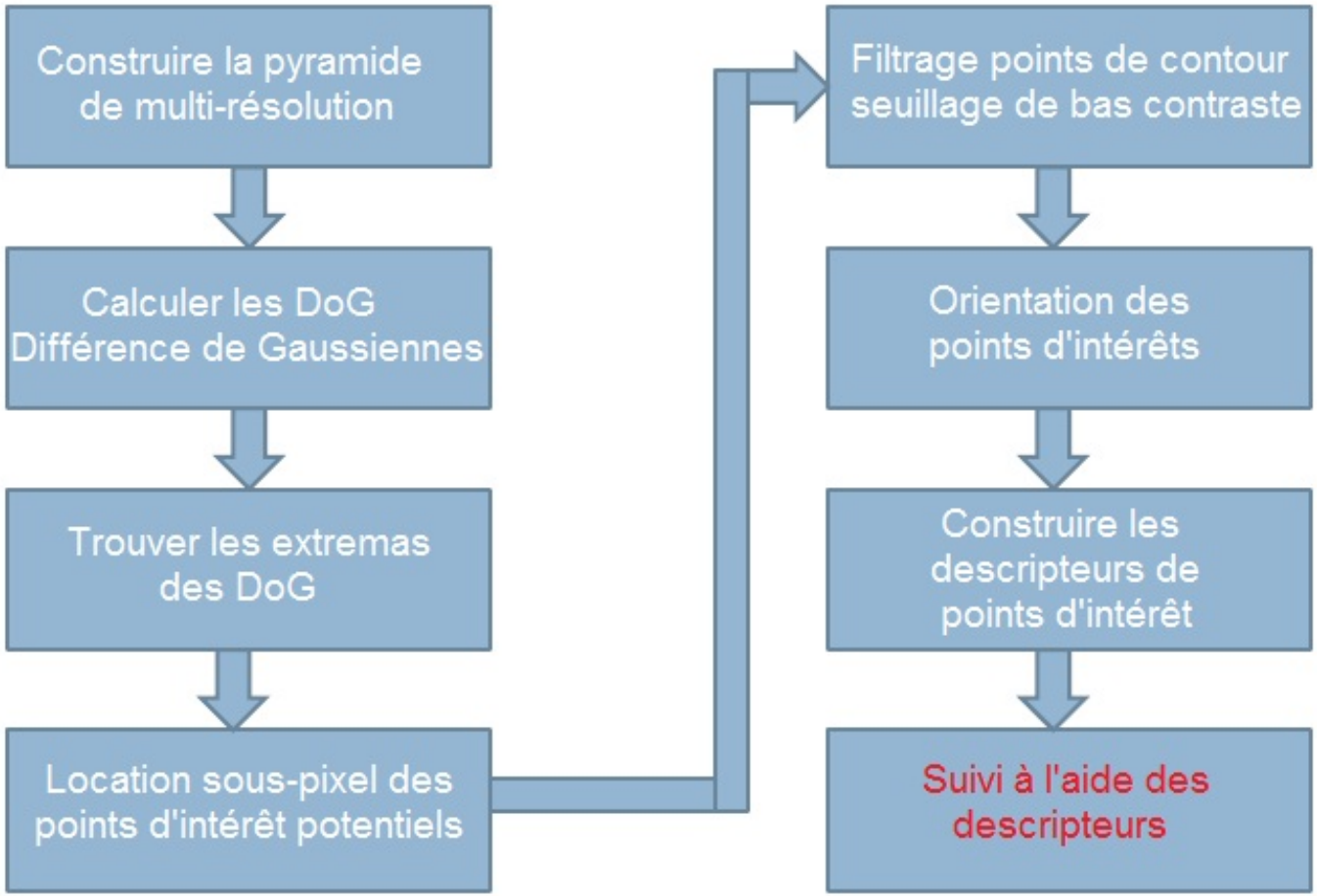
Plan de la présentation

Motivation

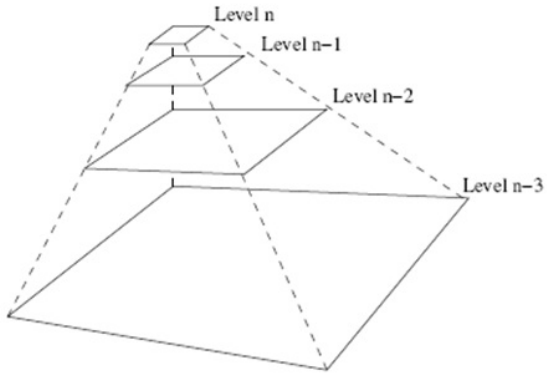
Pour fin de mise en correspondance de points d'intérêt, on voudrait trouver une primitive qui possèderait les caractéristiques suivantes :

- Invariant à la mise à échelle ;
- Invariant à la rotation ;
- Invariant à l'illumination ;
- Invariant au point de vue.

Survol de l'algorithme



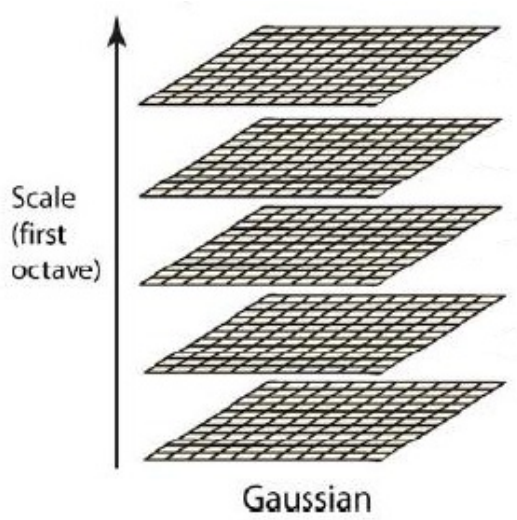
Espace multi-résolution



Pyramide d'image multi-résolution

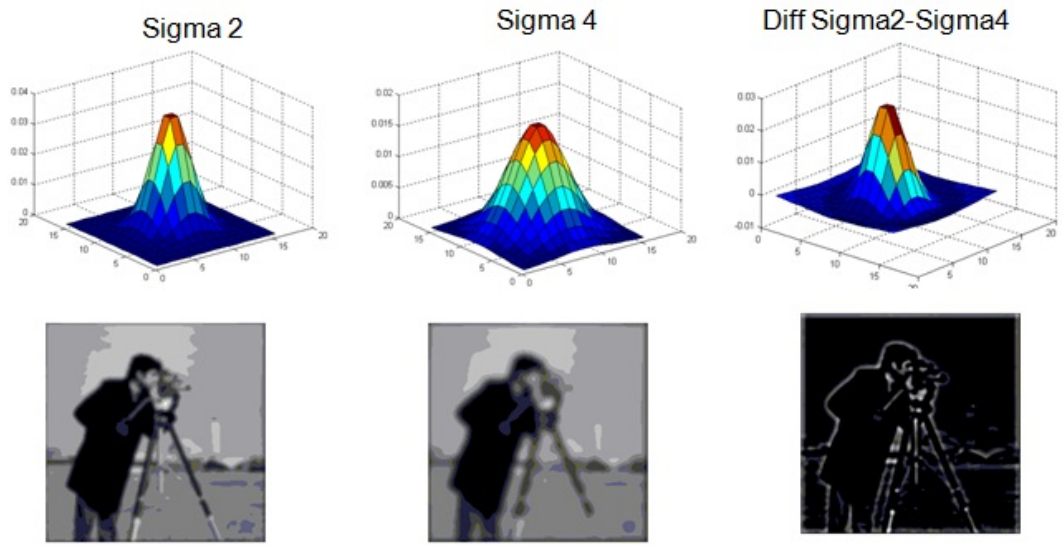
Espace multi-résolution

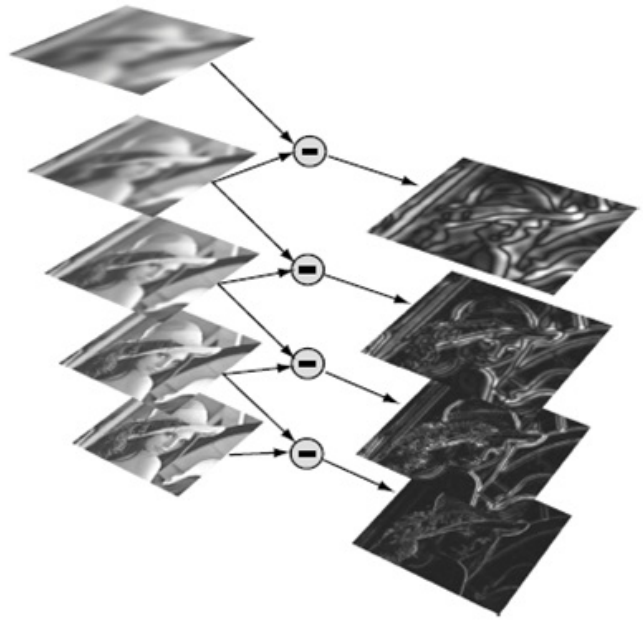
À chaque résolution (octave), on applique un flou gaussiens aux images **avec des valeur d'écart-type différentes**. Par exemple, pour un octave, on peut généré huit images flous :



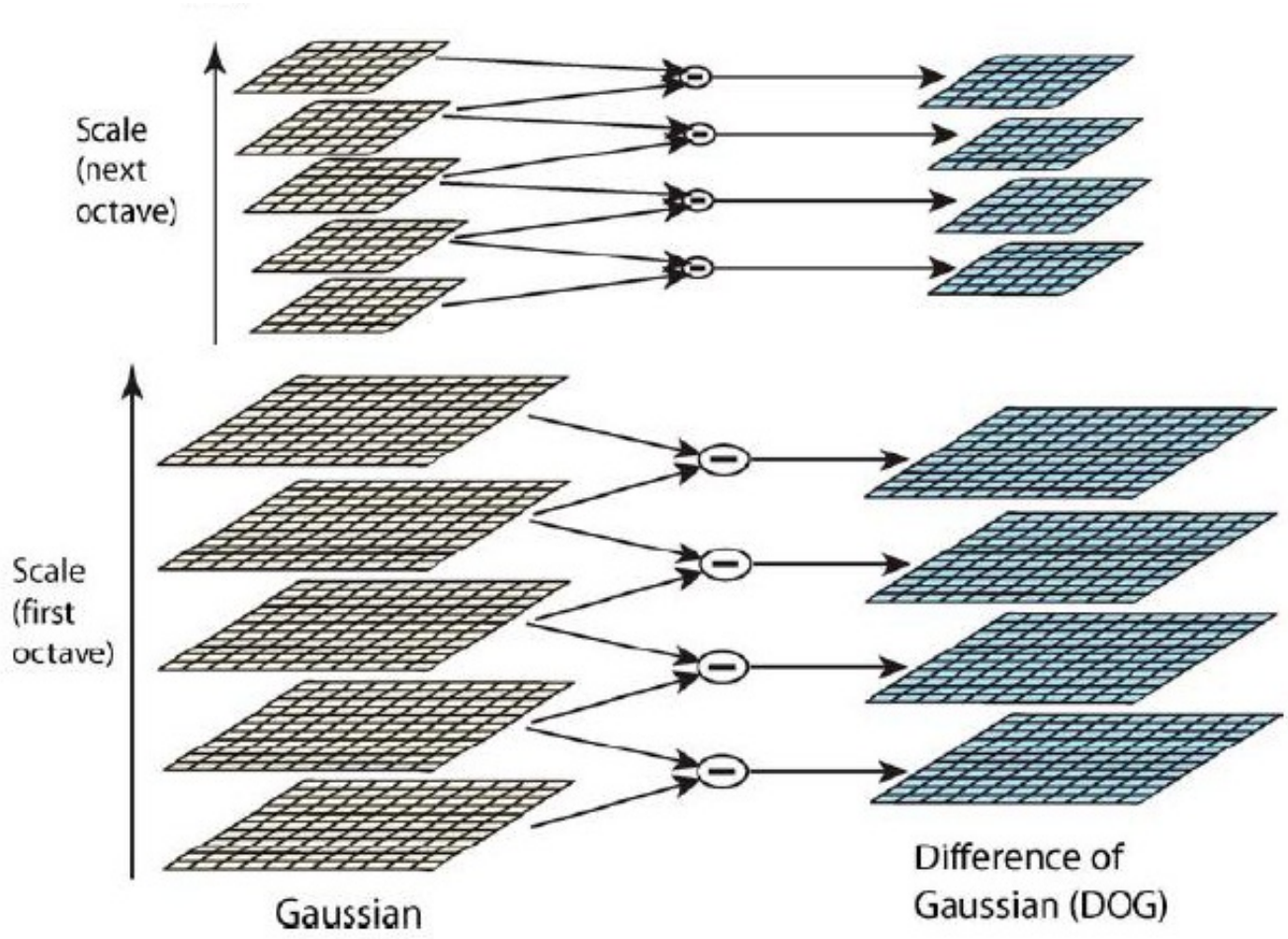
Différence de Gaussienne (DoG)

Pour trouver les extremums de ces gaussiennes, on doit trouver le laplacien des gaussiennes. Une bonne approximation est la différence de gaussienne, qui permet de trouver les contours de notre image :





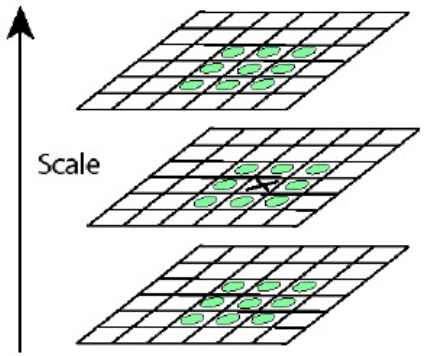
Pyramide DoG

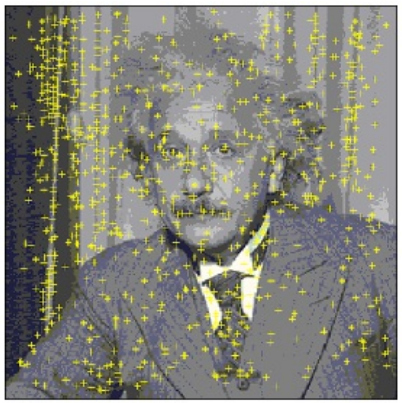


Pyramide DoG

Extremums à partir des DoG

Pour localiser des points d'intérêt potentiel, on trouve les points extrêmes dans un voisinage 3x3x3 des DoG : 3x3 au DoG courant, 3x3 au DoG précédent et 3x3 au DoG suivant. On effectue cette opération un octave à la fois, pour tous les octaves.

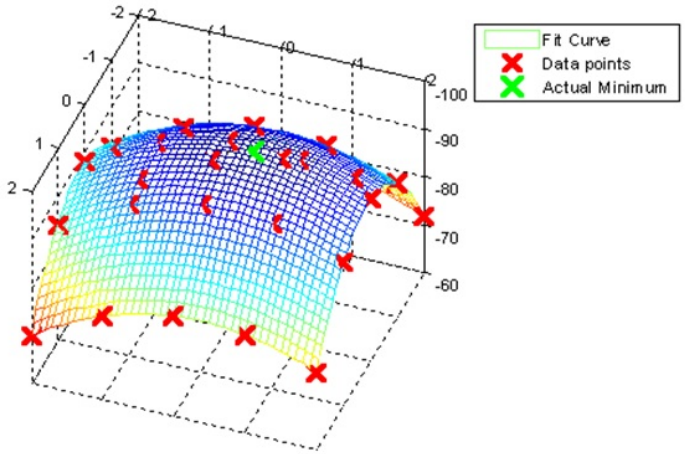


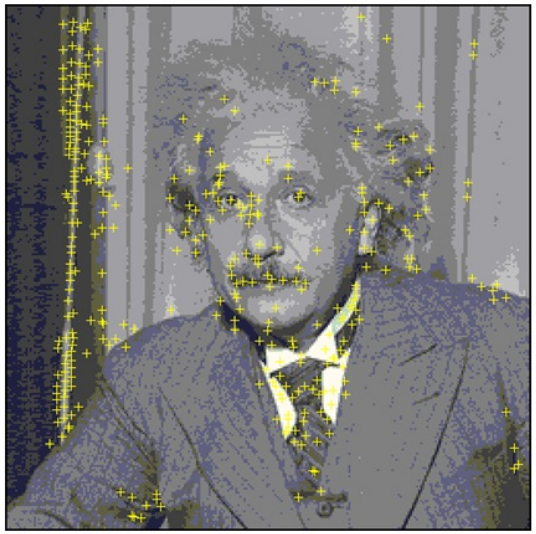


Raffinement de la position des extremums

Pour un octave, les DoG me nous donneront pas exactement les même points d'intérêts potentiel.En fittant les points du cube 3x3x3 dans une

fonction quadratique, on peut arriver à identifier l'emplacement sous-pixel du point d'intérêt extremum des DoG au pixel (x, y) :

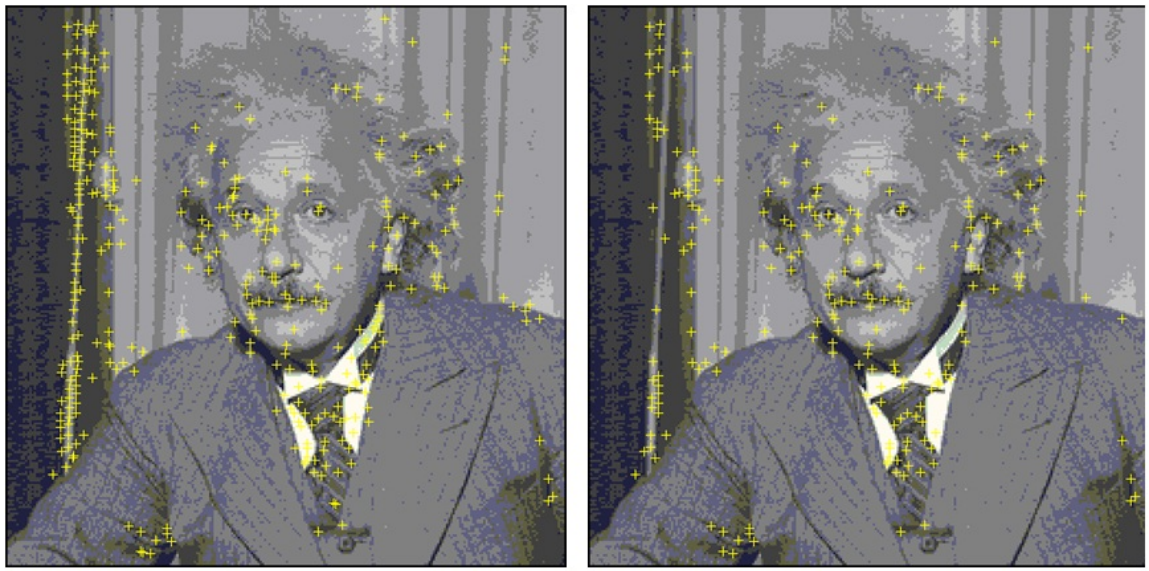




Élimination des faux extremums

De la même façon que le détecteur de Harris (Chapitre 4.1), on identifie, à l'aide du Hessien, les points qui sont des coins et on élimine les contours.

De plus, on effectue un seuil pour les extremums ayant un bas niveau de contraste dans l'espace multi-résolution (et non l'espace DoG).



Orientation

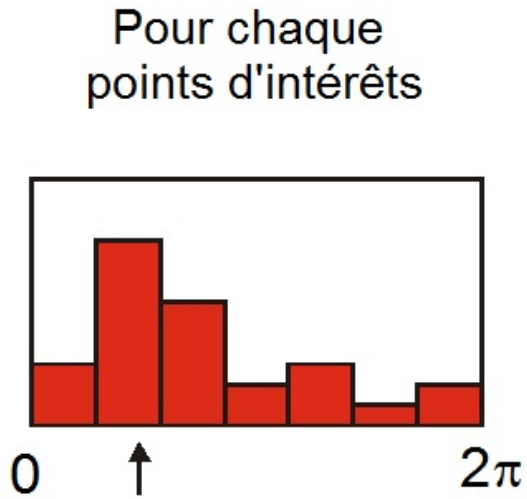
On détermine l'orientation principale d'un point d'intérêt en utilisant le **gradient des DoG** pour chaque octave :

$$\begin{aligned} m(x,y) &= \sqrt{(D(x+1,y)-D(x-1,y))^2+(D(x,y+1)-D(x,y-1))} \\ \theta(x,y) &= \tan^{-1}\left(\frac{D(x+1,y)-D(x-1,y)}{D(x,y+1)-D(x,y-1)}\right) \end{aligned}$$

En utilisant une fenêtre centrée sur le point d'intérêt, on calcule, pour chaque point, l'angle et l'intensité du gradient, que l'on intègre à un **histogramme d'angle binarisé**.

Orientation

On ajoute le gradient à l'histogramme **seulement si la norme dépasse un certain seuil**. On garde alors la meilleur orientation principale de l'histogramme.



Résumé partiel

Donc jusqu'à maintenant, on a :

- Des points d'intérêts ;
- Leur orientation principale.

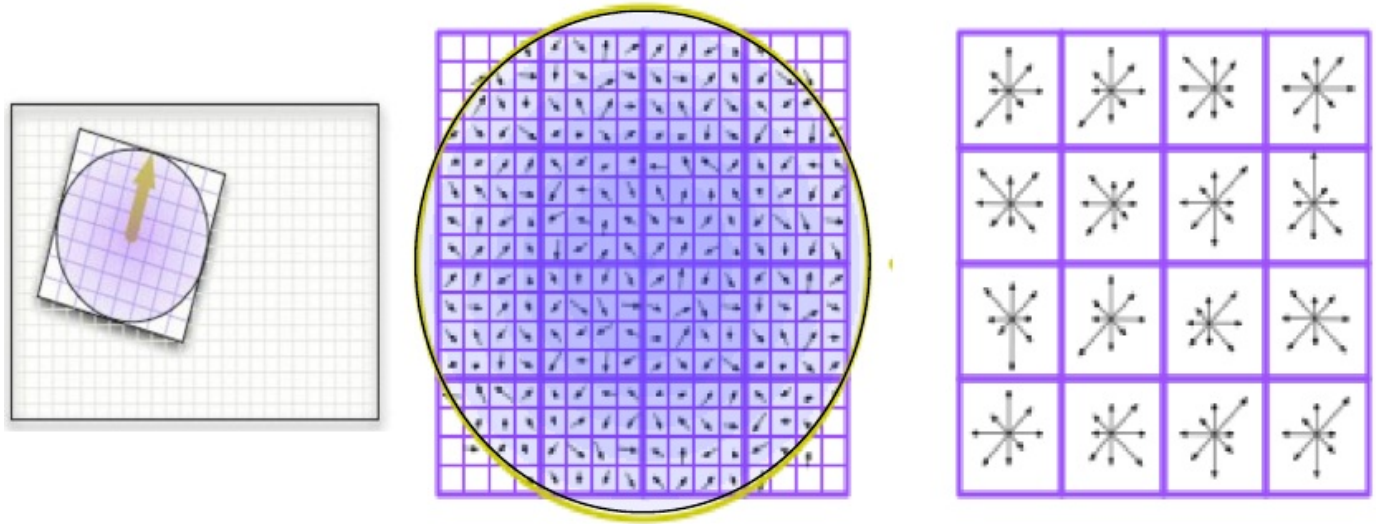
On voudrait alors obtenir les même points d'intérêt, même avec des mises à échelle, rotation, illumination différentes.

Descripteur local (signature)

Prenons l'image originale I_s dans l'espace multi-résolution. On effectue alors les opérations suivantes :

1. On échantillonne les gradients de I_s dans une matrice $16x16$;
2. On crée, de la même façon que précédemment, des histogrammes d'orientation (8 bins) pour chaque partie $4x4$ de la matrice.
3. 8 orientations x 4x4 vecteur d'histogramme = 128 dimensions !

Comment tenir compte des changements d'illumination ? On normalise les vecteurs de descripteurs d'orientation !



SURF

SURF vs SIFT

Les points SURF sont une variante plus rapide (et plus robuste) des points SIFT. En résumé :

- On remplace les informations de gradient par les ondelettes de Haar du voisinage ;
- On approxime la matrice hessienne de la détection de point par des opération entières.

Plan de la présentation

Modèle de l'apparence

Les modèles d'apparence d'objets permettent de décrire le contenu des objets en terme de **primitives basées sur l'intensité** : couleur, texture, mouvement etc.

La plupart des modèles se basent sur la distribution spatiale de la caractéristique visuelle considérée pour modéliser l'apparence.

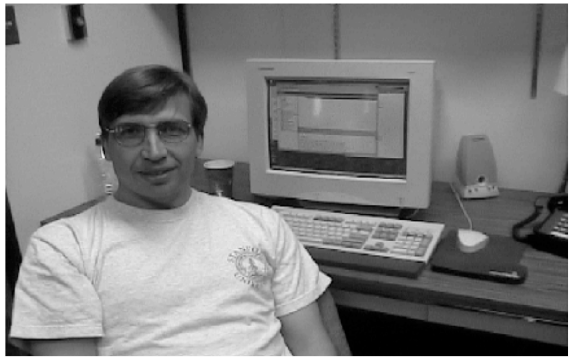
Modèle de l'apparence

Quelques modèles d'apparence :

- **Patche/modèle** : Représentation directe en couleur ou niveau de gris de l'objet à suivre.
- **Histogramme (probabiliste)** : Représente la probabilité d'obtenir la valeur de la primitive dans l'objet. (Peut être approximé par les mixtures de gaussienne).
- **Matrice de co-occurence (texture)** : Matrice qui, pour chaque pixel, représente l'information de son voisinage.

Patche/modèle

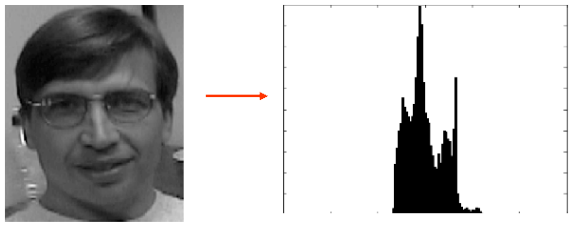
Patche/modèle :
Utile lorsqu'apparié à un modèle "bounding-box", on utilise directement l'image à l'intérieur de celle-ci pour la comparer au modèle précédent ou à une banque d'image.



Histogramme (probabiliste)

Histogramme et densité de probabilité :
Au lieu d'utiliser directement la couleur, on peut observer la distribution de celle-ci dans son ensemble, sans tenir compte de l'arrangement spatial.

On peut utiliser un modèle non-paramétrique, l'histogramme, ou une fonction de densité probabiliste (PDF) tel les mixtures de gaussiennes.

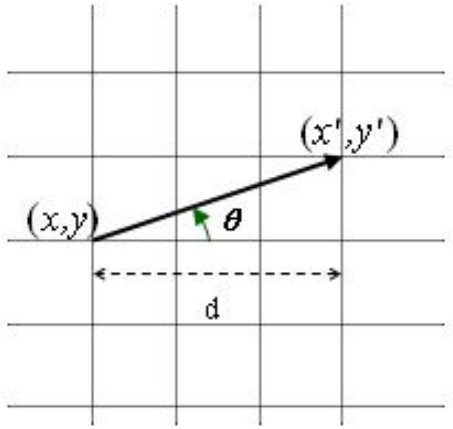


Matrice de co-occurence

Matrice de co-occurence (texture)

Outre l'analyse du domaine de Fourier, La matrice de co-occurrence peut être calculée pour chaque voisinage d'un pixel à l'intérieur de l'objet suivi.

Supposons que la valeur d'intensité des pixels est dans l'intervalle $\{0, 1, ..., m\}$. On peut calculer la matrice de co-occurence de distance d et une orientation θ représentant le voisinage observé :



Matrice de co-occurence

La valeur de $p(i,j)$ dans la matrice **C** contient la fréquence d'occurrence des deux intensités i et j sur des pixels (x,y), (x',y').

$$\mathbf{C}(d,\theta) = \begin{pmatrix} p(0,0) & p(0,1) & \dots & p(0,j) & \dots & p(0,m) \\ p(1,0) & p(1,1) & \dots & p(1,j) & \dots & p(1,m) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p(i,0) & p(i,1) & \dots & p(i,j) & \dots & p(i,m) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p(m,0) & p(m,1) & \dots & p(m,j) & \dots & p(m,m) \end{pmatrix}$$

Matrice de co-occurence

En principe, on calcule la matrice pour 4 ou 8 direction, à une distance de 1 et 2 pixels. On combine ensuite les matrices pour en former qu,une seule. **C**. Plusieurs caractéristiques de texture peuvent être extraites.

Voici quelques exemples :

- Energie $= \sum_i \sum_j [p(i,j)]^2$
- Entropie $= -\sum_i \sum_j p(i,j) log\,(p(i,j))$
- Constrast $= \sum_i \sum_j (i-j)^2 p(i,j)$
- Moment d'inertie $= \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j)$

Descripteur locaux

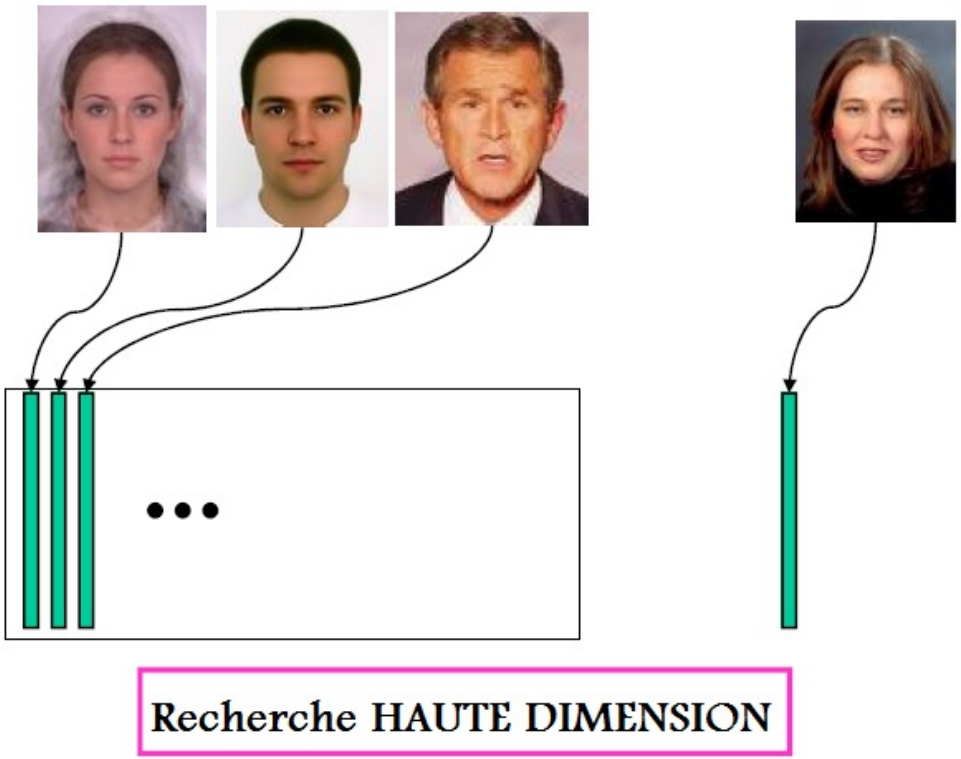
Descripteur locaux

Suite à une segmentation de l'image, on peut décrire chaque objet ou chaque partie d'objets segmentés par une des primitives précédentes.

Particulièrement adapté au modèle articulé.

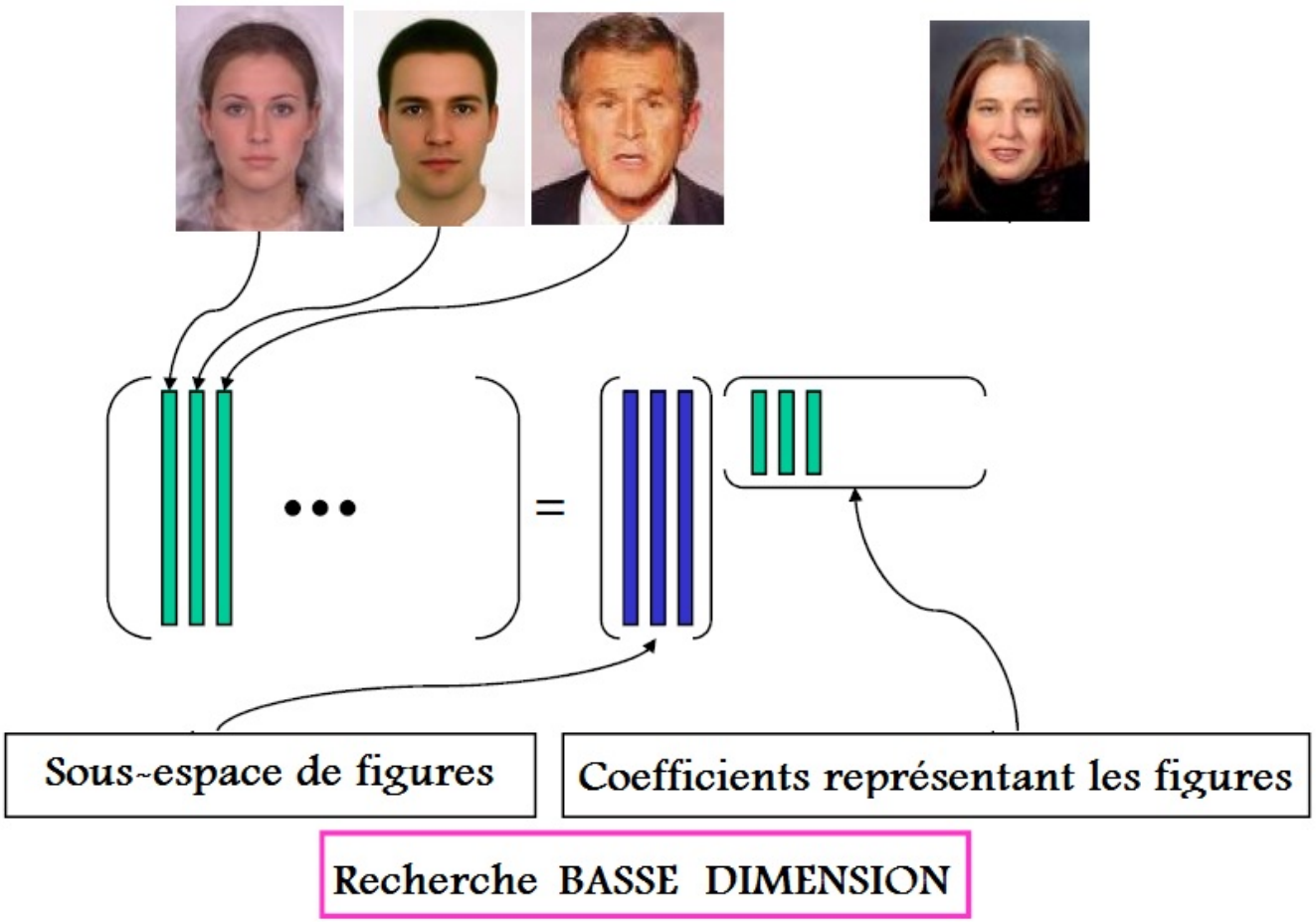
Figures propres

Supposons que nous avons une base de donnée d'images (ex : des images de visage).
On cherche à classifier un objets **en se basant sur cette base d'images** :



Figures propres

Une recherche exhaustive parmi cette base de donnée serait trop longue. On voudrait **réduire l'espace de recherche** :



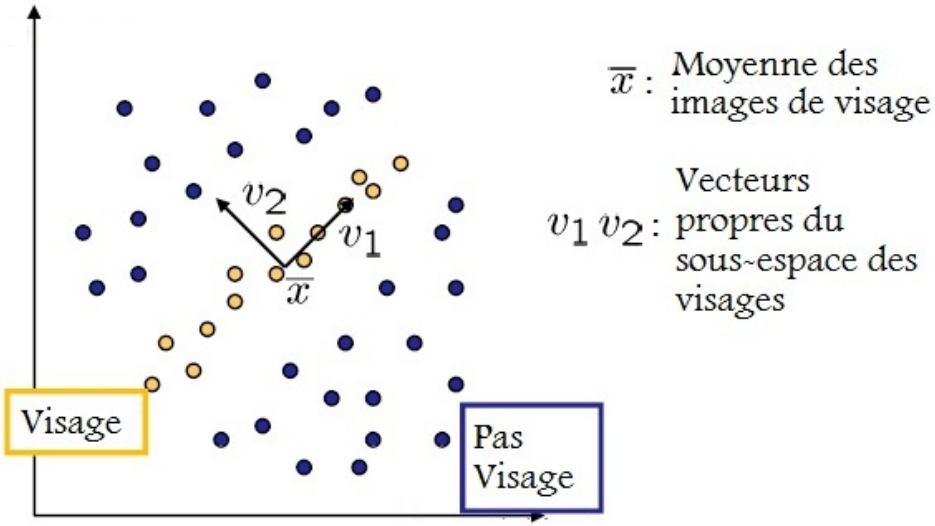
Nous **exprimerons notre requête** dans une **sous-espace** de la base d'image.

Nous utiliserons l'**analyse en composante principale** !

Figures propres

ACP (cas général à deux dimensions)

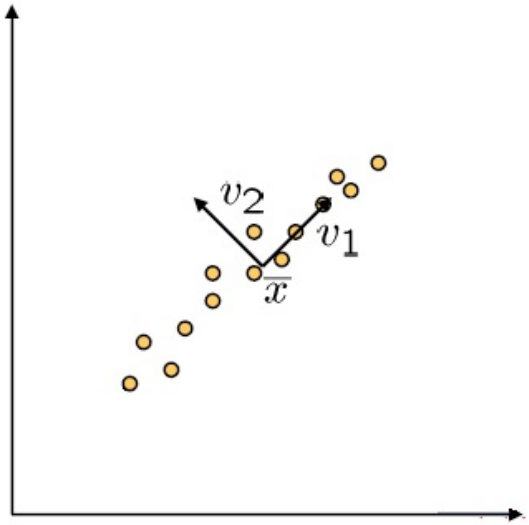
Soit un très grand ensemble d'images de *deux pixels (dimensions)* **sans lien sémantique** (bleu). Supposons que nous voulons effectuer une recherche de visage. Nous ne choisirons que les images de visage (fort lien sémantique) :



Figures propres

ACP (cas général à deux dimensions)

L'**ACP** nous permet d'effectuer un changement de base représentatif des données en utilisant les **valeurs et vecteurs propres** de l'ensemble de donnée.



La **force du lien** entre les images nous permet d'**abandonner** la composante (dimension) faible (v_2).

- Moins de données à traiter ;
- Très efficace dans un cas à N dimensions.

Figures propres

Algorithme de l'ACP (N-dimensions)

On cherche la base orthonormée $\vec{v}$ représentant le mieux les données :

$$\begin{aligned} \vec{v} &= \frac{1}{M}\sum_{r=1}^{M}\left(u_k^T \cdot (x - \bar{x})\right)^2 \\ &= u_k^T \cdot \frac{AA^T}{M} \cdot u_k \end{aligned} \tag{1}$$

On construit une matrice contenant toutes les images
$S \leftarrow$ Les M images x (vecteurs colonne de N pixels);

On calcule l'image moyenne de toutes les images
$\bar{x} \leftarrow$ Moyenne des M images;

On soustrait l'image moyenne de toutes les images
$A \leftarrow$ Les M images $x - \bar{x}$;

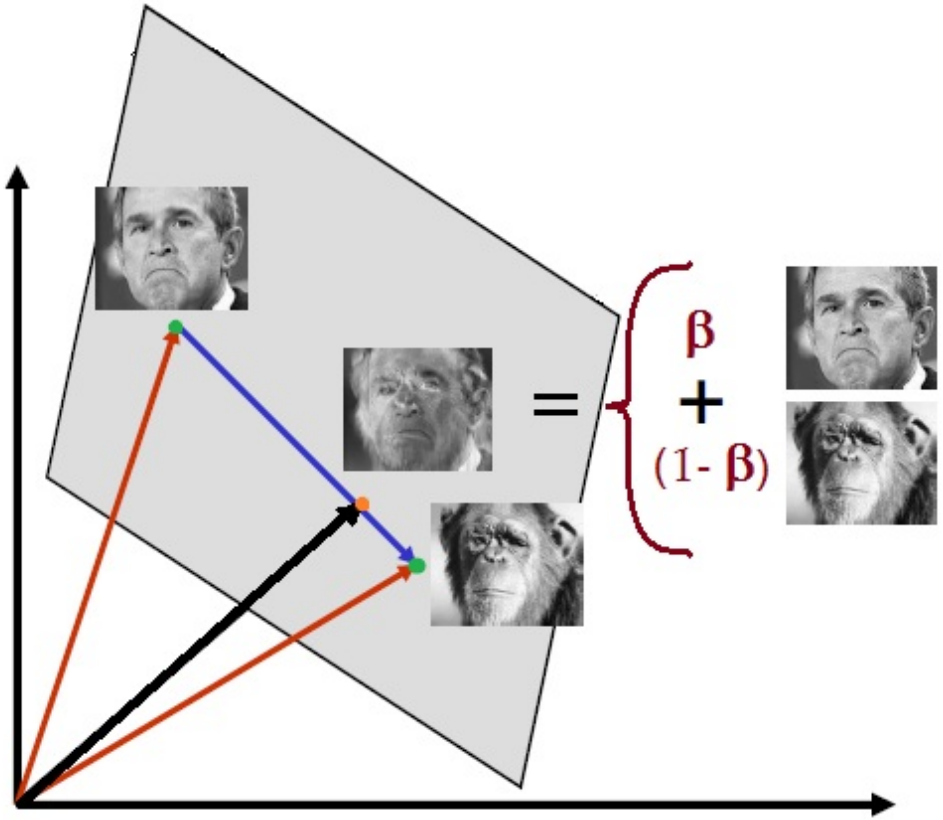
On calcule la matrice de covariance AA^T
$E \leftarrow \frac{AA^T}{M}$;

Extraction des vecteurs (v) et valeurs (λ) propres par SVD
$\vec{v}, \lambda \leftarrow \text{SVD}(E)$;

En seuillant les λ, on peut ne conserver **que les $\vec{v}$ principaux** représentant l'ensemble de donnée.

Figures propres

ACP (cas dans l'espace d'image)



- Images $N \times M \in R^{N \times M}$;
- Une image $\rightarrow$ Un point dans l'espace $(I(0,0) \times I(0,1) \times I(1,1) \times \dots)$.

1. L'ACP nous permet d'exprimer les images **par une combinaison d'images clés**.
2. L'ACP nous donne **des images représentatives de l'espace**
3. Filtrer les vecteurs propres nous permet d'obtenir **un sous-espace réduit**.

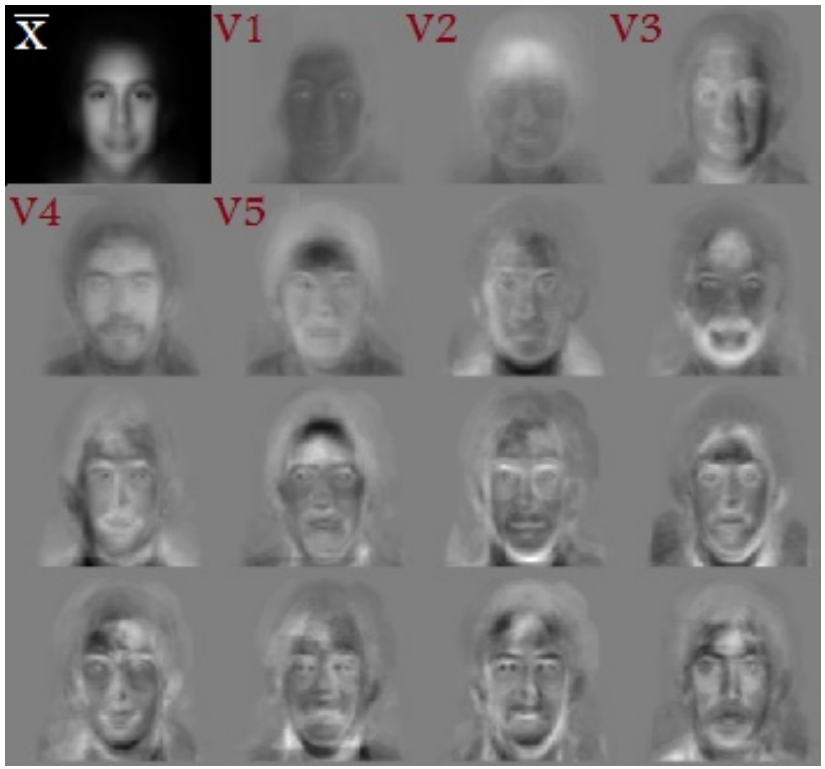
Figures propres

ACP et figures propres

$$\vec{v} = u_k^T \cdot \frac{AA^T}{M} \cdot u_k$$

L'ACP extrait les vecteurs propres ($\vec{v_1}, \vec{v_2}, \ldots, \vec{v_K}$) de $\frac{AA^T}{M}$.

Chacun de ces vecteurs est **une direction dans le sous-espace de figures**.



Figures propres

Projection d'une image dans le sous-espace de figures propres

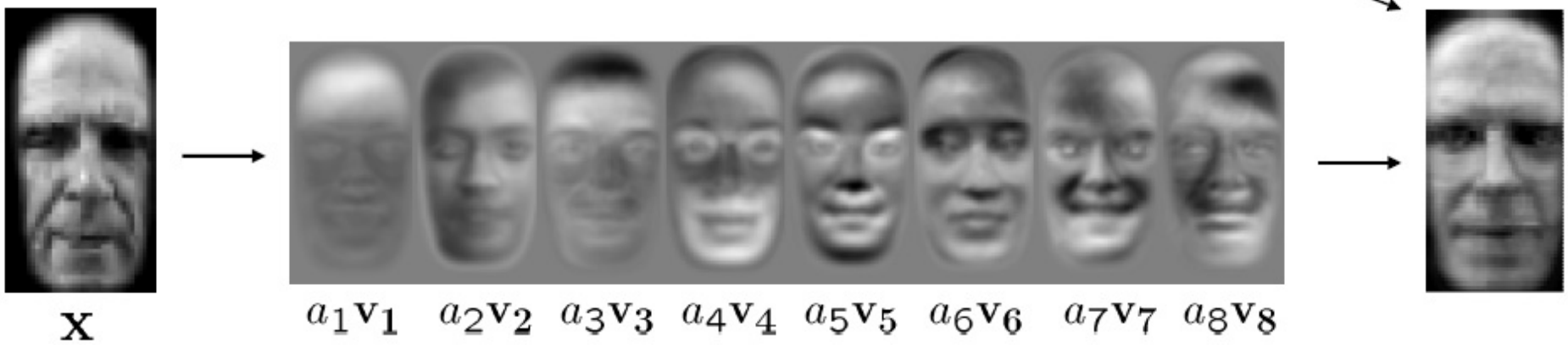
Soit x une image d'entrée.

On **projette** alors l'image centrée-réduite $(x - \bar{x})$ sur **chacun des vecteurs propres** $(\vec{v})$.

On forme alors une **combinaison linéaire** de **figures propres** nous permettant de calculer la **projection** de x sur le **sous-espace de figures propres**.

$$\mathbf{x} \rightarrow (\underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_1}}_{a_1}, \underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_2}}_{a_2}, \ldots, \underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_K}}_{a_K})$$

$$\mathbf{x} \approx \overline{\mathbf{x}} + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_K\mathbf{v_K}$$



Figures propres

Projection d'une image dans le sous-espace de figures propres

---

Entrées: $\boxed{x, y^r, T_1, T_2}$: Image à classée x, r images de la base de donnée y, Seuil d'appartenance T_1, Seuil de ressemblance T_2

On construit le sous-espace des figures propres $\vec{v}$ et son image moyenne $\bar{x}$

$\vec{v}, \bar{x} \leftarrow$ PCA en utilisant les r images y;

On calcule les K coefficient a_i^r des vecteurs propres $\vec{v}$ en projetant l'image $y^r - \bar{x}$ (y^r centrée-réduite)

pour tous les *Images y^r de la base de donnée y* **faire**

$a^r \leftarrow \{(y^r - \bar{x}) \cdot v_1, \ldots, (y^r - \bar{x}) \cdot v_K\}$;

Soit x l'image à classifier, on calcule les K coefficient a_i des vecteurs propres $\vec{v}$ en projetant l'image $x - \bar{x}$ (x centrée-réduite)

$a \leftarrow \{(x - \bar{x}) \cdot v_1, \ldots, (x - \bar{x}) \cdot v_K\}$;

On détermine s'il fait parti du sous-espace des figures

si $||x - (\bar{x} + a_1 \cdot v_1 + \cdots + a_K \cdot v_K)|| < T_1$ **alors**

C'est une figure. Est-elle très ressemblante à une figure y^r ?

pour tous les *Images y^r de la base de donnée y* **faire**

si $||a - a_K|| < T_2$ **alors**

x est associé à la figure y^r;

Plan de la présentation

Le choix du modèle

Le choix du modèle géométrique

On choisit le modèle de forme selon l'application :

- Une représentation par points est idéal pour suivre des objets petit dans une grande scène ou qui a beaucoup d'occlusion. (ex : oiseau)
- Une représentation par forme géométrique est plus approprié à de gros objets à déformation faible (ex : voiture)
- Une représentation par silhouette, contour, modèle articulé est adapté à des mouvemenents complexes ou de la reconnaissance d'actions (ex : kinect)

Le choix du modèle

Le choix du modèle d'apparence

On détermine le modèle d'apparence en fonction de la facilité à distinguer l'objet des autres ou de la scène par rapport à celle-ci.

- On choisit l'espace de couleur qui nous facilite la tâche.
- On choisit un modèle de texture lorque celle-ci se démarque des autres.
- Une représentation par points augmentés (ex : SIFT, SURF) permet de combiner à la fois des aspects d'apparence et de position.

La détection de l'objet

La détection de l'objet par rapport à la scène

- On utilise les images directements, sans traitement ;
- On apprend le background pour segmenter facilement les objets ;
- On segmente la scène (depth map ou segmentation spatiale) afin de détecter des objets à suivre ;
- On utilise une représententation en points des objets et la scène au complet.