

*UE Vision, Réalité Augmentée et Applications*

# **Matière : Vision**

## Partie II

### (Cours)

Sylvie CHAMBON

[schambon@toulouse-inp.fr](mailto:schambon@toulouse-inp.fr)

2 octobre 2025

## Notations

1.  $I$  : une image qui à chaque point/pixel  $\mathbf{p}$  associe un niveau de gris noté  $I(x, y)$
2.  $I_x, I_y, I_{xx}, I_{yy}, I_{xy}$  : images des dérivées premières et secondes de l'image
3.  $I_x(x, y), \dots, I_{xy}(x, y)$  : valeurs des dérivées premières et secondes au point  $\mathbf{p}$
4. Pour simplifier, on notera  $I_x$  pour  $I_x(x, y) \dots$

## Construction d'une pyramide

- But : produire à partir d'une image, un ensemble d'images de résolution de plus en plus faibles

## Construction d'une pyramide

- But : produire à partir d'une image, un ensemble d'images de résolution de plus en plus faibles
- Plusieurs techniques développées dans le domaine de la compression d'images

## Construction d'une pyramide

- But : produire à partir d'une image, un ensemble d'images de résolution de plus en plus faibles
- Plusieurs techniques développées dans le domaine de la compression d'images
- Différents types de pyramides : Pyramides **régulières** ou irrégulières

## Construction d'une pyramide régulière

- Soit  $n$  la résolution courante à calculer (qui est plus faible que la résolution  $n - 1$ )

$$I^n(i, j) = \underset{(i', j') \in ZV(i, j)}{\text{reduction}} f_r(I^{n-1}(i', j')), \quad (1)$$







## Construction d'une pyramide régulière

- *Pyramides gaussiennes* – Lissage par un filtre gaussien, puis sous-échantillonnage, en récupérant un pixel sur quatre
- *Pyramides moyennes* – Moyenne sur un voisinage  $2 \times 2$

# Plan

Notations

Construction d'une pyramide

## Détecteurs du premier ordre

Détecteurs du premier ordre : vers le multi-échelle

## Détecteurs de régions

Vers le multi-échelle : Détecteur de Kadir

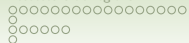
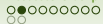
Vers le multi-échelle : EBR, *Edge-Based Regions*

Vers le multi-échelle : PCBR, *Principal Curvature-Based Regions*

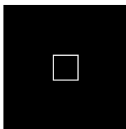
## Détecteurs du second ordre

Vers le multi-échelle : SIFT *Scale Invariant Feature Transform*

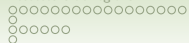
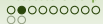
Vers une variante accélérée de SIFT : SURF



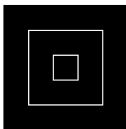
# Moravec



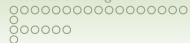
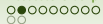
Point quelconque



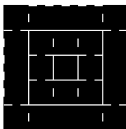
# Moravec



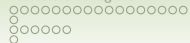
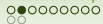
Point quelconque



# Moravec



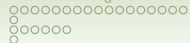
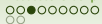
Point quelconque



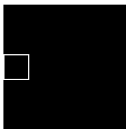
# Moravec



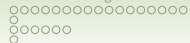
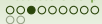
Point quelconque



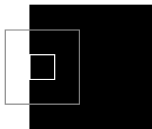
# Moravec



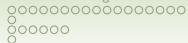
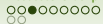
Point contour



# Moravec



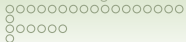
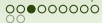
Point contour



# Moravec



Point contour



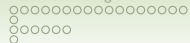
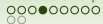
# Moravec



Point contour



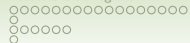
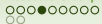
direction contour



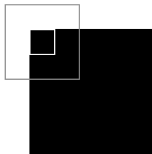
# Moravec



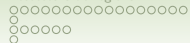
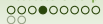
Point de type coin



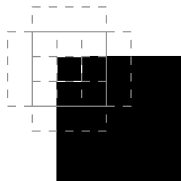
# Moravec



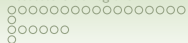
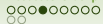
Point de type coin



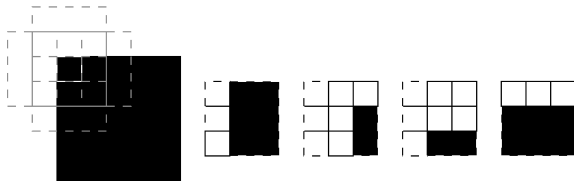
# Moravec



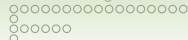
Point de type coin



# Moravec



Point de type coin



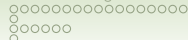
## Moravec : définition

Soit  $(u, v)$  un déplacement, l'impact de ce déplacement se mesure par :

$$E_{(u,v)}(x, y) = \sum_{x,y} w(x, y) (I(x + u, y + v) - I(x, y))^2$$

où :

1.  $w$  : poids
2.  $E_{(u,v)}(x, y) =$  **moyenne du changement** si on déplace la fenêtre



## Moravec : définition

Soit  $(u, v)$  un déplacement, l'impact de ce déplacement se mesure par :

$$E_{(u,v)}(x, y) = \sum_{x,y} w(x, y) (I(x + u, y + v) - I(x, y))^2$$

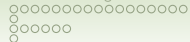
où :

1.  $w$  : poids
2.  $E_{(u,v)}(x, y) =$  **moyenne du changement** si on déplace la fenêtre

**La réponse du détecteur est donc la valeur minimale parmi toutes les directions testées :**

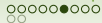
$$Mor(x, y) = \min_{(u,v)} E_{(u,v)}(x, y)$$





# Harris : améliorations de Moravec

1. **La réponse est bruitée en raison des poids binaires** : utilisation des poids Gaussiens, de moyennes nulles



## Harris : améliorations de Moravec

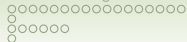
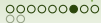
1. **La réponse est bruitée en raison des poids binaires** : utilisation des poids Gaussiens, de moyennes nulles
2. **La réponse du détecteur est anisotrope (dépendant de la direction)**

## Harris : améliorations de Moravec

1. **La réponse est bruitée en raison des poids binaires** : utilisation des poids Gaussiens, de moyennes nulles
2. **La réponse du détecteur est anisotrope (dépendant de la direction)**

ET

3. **La réponse est trop forte/ambiguë au niveau des contours**



## Harris : améliorations de Moravec

**développement de Taylor à l'ordre 2** au voisinage de  $(x, y)$

$$I(x + u, y + v) = I(x, y) + u \frac{\partial I}{\partial u} + v \frac{\partial I}{\partial v} + o(u^2, v^2)$$

## Harris : améliorations de Moravec

**développement de Taylor à l'ordre 2** au voisinage de  $(x, y)$

$$I(x + u, y + v) = I(x, y) + u \frac{\partial I}{\partial u} + v \frac{\partial I}{\partial v} + o(u^2, v^2)$$

D'où, dans cette équation :

$$E_{(u,v)}^{\text{modifiée}}(x, y) = \sum_{x,y} w(x, y) \left( u \frac{\partial I}{\partial u} + v \frac{\partial I}{\partial v} + o(u^2, v^2) \right)^2$$

## Harris : améliorations de Moravec

développement de Taylor à l'ordre 2 au voisinage de  $(x, y)$

$$I(x + u, y + v) = I(x, y) + u \frac{\partial I}{\partial u} + v \frac{\partial I}{\partial v} + o(u^2, v^2)$$

D'où, dans cette équation :

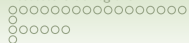
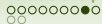
$$E_{(u,v)}^{\text{modifiée}}(x, y) = \sum_{x,y} w(x, y) \left( u \frac{\partial I}{\partial u} + v \frac{\partial I}{\partial v} + o(u^2, v^2) \right)^2$$

On obtient :

$$E_{\text{modifiée}} = Au^2 + 2Cuv + Bv^2,$$

avec  $A = \frac{\partial^2 I}{\partial u^2} \otimes w$ ,  $B = \frac{\partial^2 I}{\partial v^2} \otimes w$  et  $C = \frac{\partial I}{\partial u} \frac{\partial I}{\partial v} \otimes w$

$\otimes$  = convolution

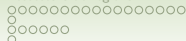
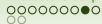


## Harris : améliorations de Moravec

1. En repartant de l'équation (2)

$$E_{(u,v)}^{\text{modifiée}}(x,y) = (u,v)M(u,v)^T,$$

$$\text{avec } M = \begin{pmatrix} A & C \\ C & B \end{pmatrix}.$$



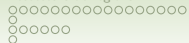
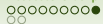
## Harris : améliorations de Moravec

1. En repartant de l'équation (2)

$$E_{(u,v)}^{\text{modifiée}}(x, y) = (u, v)M(u, v)^T,$$

avec  $M = \begin{pmatrix} A & C \\ C & B \end{pmatrix}.$

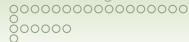
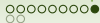
2. **Les valeurs propres de la matrice  $M$  correspondent aux courbures principales associées à  $E$** 
  - deux courbures faibles = région homogène
  - une des courbures est forte et l'autre est faible = un contour
  - deux courbures fortes = un coin, un point d'intérêt



## Harris : améliorations de Moravec

$$R(x, y) = \text{Det}(M) - k\text{Trace}(M)^2$$

avec  $\text{Det}(M) = AB - C^2$  et  $\text{Trace}(M) = A + B$

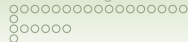
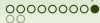


## Harris : améliorations de Moravec

$$R(x, y) = \text{Det}(M) - k\text{Trace}(M)^2$$

avec  $\text{Det}(M) = AB - C^2$  et  $\text{Trace}(M) = A + B$

1. on s'est affranchi du choix de la direction !

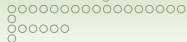
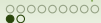


## Harris : améliorations de Moravec

$$R(x, y) = \text{Det}(M) - k\text{Trace}(M)^2$$

avec  $\text{Det}(M) = AB - C^2$  et  $\text{Trace}(M) = A + B$

1. **on s'est affranchi du choix de la direction !**
2. Les valeurs de  $R$  sont
  - positives au voisinage d'un coin
  - négatives au voisinage d'un contour
  - faibles dans une région d'intensité constante

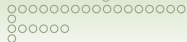
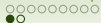


## Harris multi-échelle : Harris-Laplace

### 1. Calcul de la réponse de Harris à plusieurs échelles

Extrema locaux estimées en  $(x, y, \sigma)$

### 2. Affinement de la sélection des points d'intérêt dans l'espace échelle du laplacien



## Harris multi-échelle : Harris-Laplace

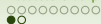
### 1. Calcul de la réponse de Harris à plusieurs échelles

Extrema locaux estimées en  $(x, y, \sigma)$

### 2. Affinement de la sélection des points d'intérêt dans l'espace échelle du laplacien

3. Algorithme itératif appliqué pour chaque candidat retenu

4. Soient  $(i_0, j_0, \sigma_0)$ , les coordonnées et l'échelle initiale



## Harris multi-échelle : Harris-Laplace

### 1. Calcul de la réponse de Harris à plusieurs échelles

Extrema locaux estimées en  $(x, y, \sigma)$

### 2. Affinement de la sélection des points d'intérêt dans l'espace échelle du laplacien

### 3. Algorithme itératif appliqué pour chaque candidat retenu

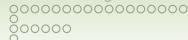
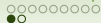
### 4. Soient $(i_0, j_0, \sigma_0)$ , les coordonnées et l'échelle initiale

4.1 Recherche d'un extremum dans l'espace échelle du laplacien pour les échelles comprises dans  $[c^{-1}\sigma_{l_k}; c\sigma_{l_k}]$ ,

4.2 Intervalle inter-échelle guidé par  $c_2 < c$

Si un extremum n'est pas trouvé alors ce candidat est rejeté

Sinon, on note  $\sigma_u$ , l'échelle à laquelle cet extremum a été trouvé



## Harris multi-échelle : Harris-Laplace

### 1. Calcul de la réponse de Harris à plusieurs échelles

Extrema locaux estimées en  $(x, y, \sigma)$

### 2. Affinement de la sélection des points d'intérêt dans l'espace échelle du laplacien

### 3. Algorithme itératif appliqué pour chaque candidat retenu

### 4. Soient $(i_0, j_0, \sigma_0)$ , les coordonnées et l'échelle initiale

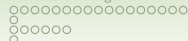
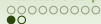
4.1 Recherche d'un extremum dans l'espace échelle du laplacien pour les échelles comprises dans  $[c^{-1}\sigma_{l_k}; c\sigma_{l_k}]$ ,

4.2 Intervalle inter-échelle guidé par  $c_2 < c$

Si un extremum n'est pas trouvé alors ce candidat est rejeté

Sinon, on note  $\sigma_u$ , l'échelle à laquelle cet extremum a été trouvé

4.3 Détection de la position  $(i_{k+1}, j_{k+1})$  du maximum de la réponse de Harris le plus proche de  $(i_k, j_k)$  à l'échelle  $\sigma_u$



## Harris multi-échelle : Harris-Laplace

### 1. Calcul de la réponse de Harris à plusieurs échelles

Extrema locaux estimées en  $(x, y, \sigma)$

### 2. Affinement de la sélection des points d'intérêt dans l'espace échelle du laplacien

### 3. Algorithme itératif appliqué pour chaque candidat retenu

### 4. Soient $(i_0, j_0, \sigma_0)$ , les coordonnées et l'échelle initiale

4.1 Recherche d'un extremum dans l'espace échelle du laplacien pour les échelles comprises dans  $[c^{-1}\sigma_{l_k}; c\sigma_{l_k}]$ ,

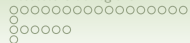
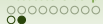
4.2 Intervalle inter-échelle guidé par  $c_2 < c$

Si un extremum n'est pas trouvé alors ce candidat est rejeté

Sinon, on note  $\sigma_u$ , l'échelle à laquelle cet extremum a été trouvé

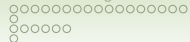
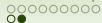
4.3 Détection de la position  $(i_{k+1}, j_{k+1})$  du maximum de la réponse de Harris le plus proche de  $(i_k, j_k)$  à l'échelle  $\sigma_u$

Nouvelle itération si  $\sigma_k \neq \sigma_{k+1}$  et  $(i_k, j_k) \neq (i_{k+1}, j_{k+1})$



## Remarques

- Une version moins coûteuse : seuillage de la réponse



## Remarques

- Une version moins coûteuse : seuillage de la réponse
- Hessien-Laplace : même algorithme appliqué au détecteur de Beudet (que nous verrons par la suite)

# Plan

Notations

Construction d'une pyramide

Détecteurs du premier ordre

Détecteurs du premier ordre : vers le multi-échelle

Détecteurs de régions

Vers le multi-échelle : Détecteur de Kadir

Vers le multi-échelle : EBR, *Edge-Based Regions*

Vers le multi-échelle : PCBR, *Principal Curvature-Based Regions*

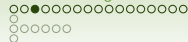
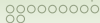
Détecteurs du second ordre

Vers le multi-échelle : SIFT *Scale Invariant Feature Transform*

Vers une variante accélérée de SIFT : SURF

## Détecteurs de régions

**Les détecteurs de type régions ne détectent pas les coins ou des angles dans l'image. Ils considèrent le centre des régions détectées comme étant le point d'intérêt.**

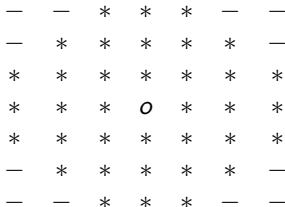


# Détecteur SUSAN (*Smallest Univalve Segment Assimilating Nucleus*)

## Les principes

1. Notion de **noyau**, *nucleus* = pixel étudié
2. Notion de région d'intérêt autour de ce noyau = **masque** circulaire
3. Notion de sous-région, *segment*, autour du noyau, photométriquement proche du noyau
4. Forme et taille de cette sous-région = caractérise le point et indique sa particularité

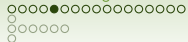
## SUSAN : masque circulaire



○ : pixel étudié

\* : pixel pris en compte dans le voisinage

— : pixel non pris en compte dans le voisinage



## SUSAN : définition de l'USAN

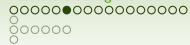
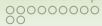
$$f : \mathbb{N}^2 \rightarrow \mathbb{R}$$

$$\mathbf{p} \rightarrow f(\mathbf{p}) = \sum_{\mathbf{p}' \in \mathcal{V}(\mathbf{p})} e^{-\left(\frac{I(\mathbf{p}) - I(\mathbf{p}')}{T_u}\right)^6}$$

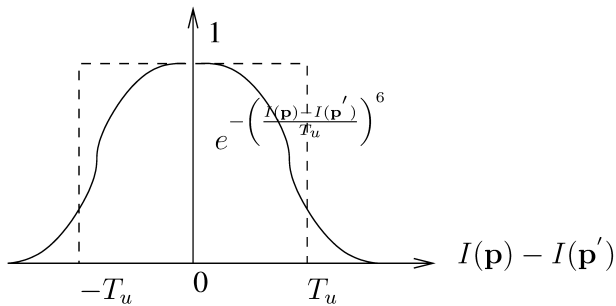
$T_u$  correspond au contraste maximal autorisé

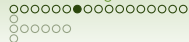
**USAN = région d'intérêt/région d'appartenance du pixel étudié**

**Le barycentre de l'USAN = le centre**



# SUSAN : Comportement de la fonction de similarité





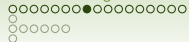
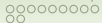
## SUSAN : définition de la réponse

$$SU(\mathbf{p}) = \begin{cases} T_g - f(\mathbf{p}) & \text{si } f(\mathbf{p}) < T_g \\ 0 & \text{sinon} \end{cases}$$

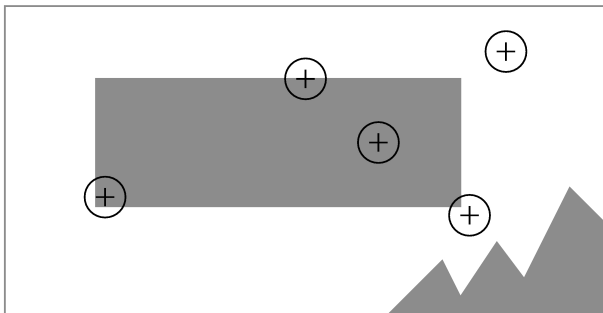
$T_g$  est un seuil géométrique.

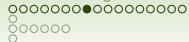
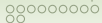




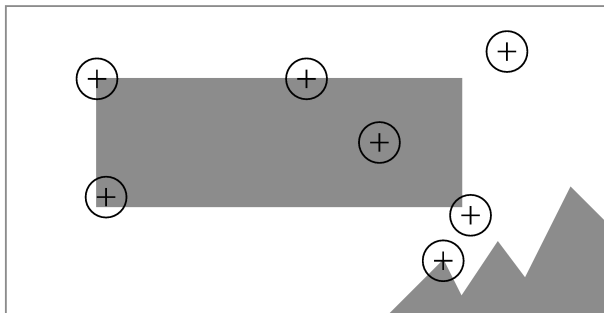


# Analyse de la surface USAN





# Analyse de la surface USAN



# Comportement du détecteur

## Comportement “idéal”

- Surface de l'USAN proche de la surface du voisinage = zone homogène
- Surface de l'USAN proche de la moitié de la surface du voisinage = contour
- Surface de l'USAN plus petite que la moitié de la surface du voisinage = point d'intérêt

## Comportement du détecteur

### Comportement “idéal”

- Surface de l'USAN proche de la surface du voisinage = zone homogène
- Surface de l'USAN proche de la moitié de la surface du voisinage = contour
- Surface de l'USAN plus petite que la moitié de la surface du voisinage = point d'intérêt

Ainsi, souvent,  $T_g = \frac{s_{max}}{2}$  où  $s_{max}$  est la surface maximale de l'USAN étant donnée la taille du voisinage considéré.

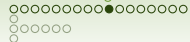
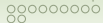
## Comportement du détecteur

### Comportement “idéal”

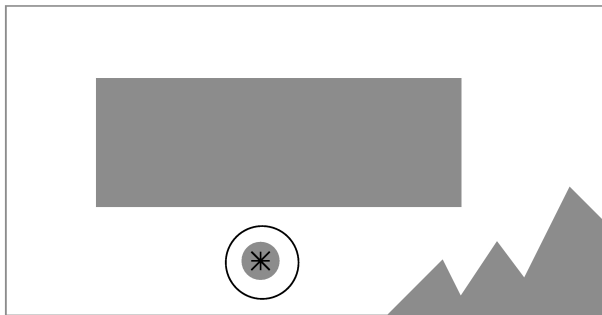
- Surface de l'USAN proche de la surface du voisinage = zone homogène
- Surface de l'USAN proche de la moitié de la surface du voisinage = contour
- Surface de l'USAN plus petite que la moitié de la surface du voisinage = point d'intérêt

Ainsi, souvent,  $T_g = \frac{s_{max}}{2}$  où  $s_{max}$  est la surface maximale de l'USAN étant donnée la taille du voisinage considéré.

Bien sûr, il y a des cas où le respect de cette contrainte n'est pas corrélé au fait que le point étudié est un point d'intérêt ← nécessité d'ajouter des contraintes



## SUSAN : Contrainte du centre



+ Noyaux

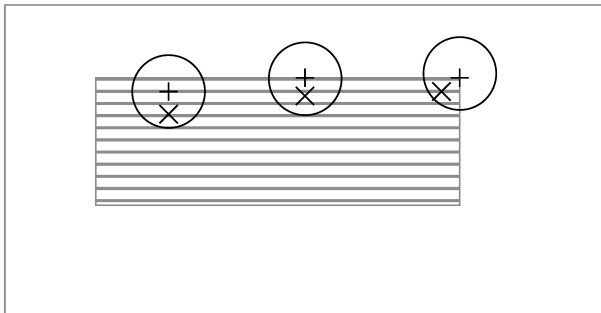
× Centres



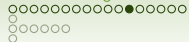
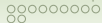


## SUSAN : Contrainte de continuité

Quelle est la différence entre le noyau et le centre d'une région ?



+ Noyaux  
× Centres



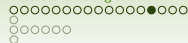
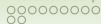
## SUSAN : Conclusion

**Plus on ajoute de contraintes, plus on élimine de “mauvais” points d’intérêt mais plus on risque d’éliminer de “bons” points d’intérêt !**

# SUSAN FAST, Features from Accelerated Segment Test

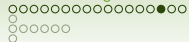
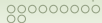
- Proche de SUSAN
- Zone circulaire : Algorithme du cercle de Bresenham de rayon 3  
→ 16 points sur le périmètre de ce cercle
- Ensembles de pixels de même intensité qui forment une région connexe
- Vérification d'une de ces hypothèses, pour  $T$  à choisir :
  1.  $\exists Np = 12$  pixels connexes tels que  $I(x', y') < I(x, y) - T$
  2.  $\exists Np = 12$  pixels connexes tels que  $I(x', y') > I(x, y) + T$
- Version plus rapide avec 4 pixels et vérification de 3/4 pixels
- Inconvénient : comment choisir les  $n$  pixels significatives ?

Solution **Apprentissage**



## Maximally Stable Extremal Regions (MSER)

- **Principe** : intensité varie rapidement aux bords des objets  
régions = composantes connexes de l'image seuillée (ligne de partage des eaux)
- **Propriétés** : invariant aux transformations monotones d'intensité et aux transformations géométriques homographiques ou non linéaires mais continues
- **Inconvénient** : sensible au flou (intensité sur les frontières mal définie)



# Intensity Based Regions (IBR)

- **Étapes**

1. Détection des extrema locaux de l'intensité lumineuse de l'image  $I$
2. Caractérisation de la région autour de ces extrema

## Intensity Based Regions (IBR)

- **Étapes**

1. Détection des extrema locaux de l'intensité lumineuse de l'image  $I$
2. Caractérisation de la région autour de ces extrema

- **Caractérisation**

1. Fonction d'intensité le long des rayons issus de cet extremum

$$f_I(t) = \frac{|I(t) - I(0)|}{\max(d, \frac{1}{t} \int_0^t |I(x) - I(0)| dx)} \quad (2)$$

## Intensity Based Regions (IBR)

- **Étapes**

1. Détection des extrema locaux de l'intensité lumineuse de l'image  $I$
2. Caractérisation de la région autour de ces extrema

- **Caractérisation**

1. Fonction d'intensité le long des rayons issus de cet extremum

$$f_I(t) = \frac{|I(t) - I(0)|}{\max(d, \frac{1}{t} \int_0^t |I(x) - I(0)| dx)} \quad (2)$$

2. Extremum chaînés entre eux

## Intensity Based Regions (IBR)

- **Étapes**

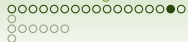
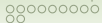
1. Détection des extrema locaux de l'intensité lumineuse de l'image  $I$
2. Caractérisation de la région autour de ces extrema

- **Caractérisation**

1. Fonction d'intensité le long des rayons issus de cet extremum

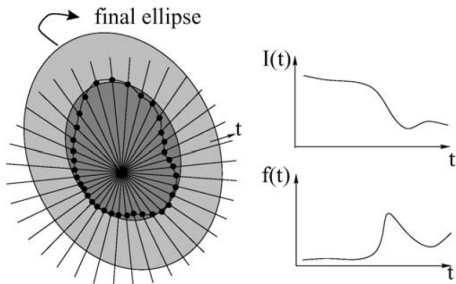
$$f_I(t) = \frac{|I(t) - I(0)|}{\max(d, \frac{1}{t} \int_0^t |I(x) - I(0)| dx)} \quad (2)$$

2. Extremum chaînés entre eux
3. Détermination de l'ellipse



# Intensity Based Regions (IBR)

## Illustration de la région



## *Intensity Based Regions (IBR)*

### **Analyse**

- Notion de blob : proche de l'approche définie dans SUSAN
- Taille de l'ellipse doublée
- Robuste sur les documents imprimés
- Invariant par les transformations affines géométriques et photométriques

## Détecteur de Kadir

- Entropie de la distribution des valeurs des pixels au voisinage du point considéré
- = Plus le voisinage est désordonné plus il est discriminant

$$\mathcal{H} : \mathbb{N}^2 \rightarrow \mathbb{R}$$

$$\mathbf{p} \mapsto - \sum_{k,l} p(\mathbf{p}) \log(p(\mathbf{p})) \quad (3)$$

avec  $(k, l) \in \mathbb{N}^2$

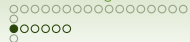
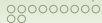
et  $(k - i)^2 + (l - j)^2 < r^2$  où  $r$  est le rayon du voisinage considéré

et probabilité  $p(\mathbf{p}) =$  distribution des valeurs des pixels du voisinage

- La réponse est donnée par :

$$\text{KA}(\mathbf{p}) = \mathcal{H}(\mathbf{p}) \frac{\sigma^2}{2\sigma - 1} \sum_{k,l} \left| \frac{\partial p(\mathbf{p})}{\partial \sigma} \right|. \quad (4)$$

- Invariant aux transformations photométriques (biais), géométriques affines
- = indépendant du point de vue



## Edge-Based Regions (EBR)

- Contours stables aux changements de point de vue et de conditions d'éclairage
- **Conséquence** : Utilisation de la géométrie locale : droites et parallélogrammes

# Edge-Based Regions (EBR)

## Étapes

1. Coins localisés avec Harris
2. Contours détectés avec Deriche
3. Pour chaque coin, analyse des contours passant par ce coin

# Edge-Based Regions (EBR)

## Principe

- Un coin  $p$  et deux points  $p_1(l_1)$  et  $p_2(l_2)$  se déplaçant le long des contours partant de ce coin

## Edge-Based Regions (EBR)

### Principe

- Un coin  $p$  et deux points  $p_1(l_1)$  et  $p_2(l_2)$  se déplaçant le long des contours partant de ce coin
- Distance curviligne  $l_{1,2}$  de chaque point  $p_{1,2}$  au coin  $p =$  surface entre le contour et la droite joignant le coin et le point  $p_{1,2}$

## Edge-Based Regions (EBR)

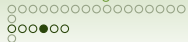
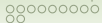
### Principe

- Un coin  $p$  et deux points  $p_1(l_1)$  et  $p_2(l_2)$  se déplaçant le long des contours partant de ce coin
- Distance curviligne  $l_{1,2}$  de chaque point  $p_{1,2}$  au coin  $p =$  surface entre le contour et la droite joignant le coin et le point  $p_{1,2}$
- Distance invariante aux transformations affines

## Edge-Based Regions (EBR)

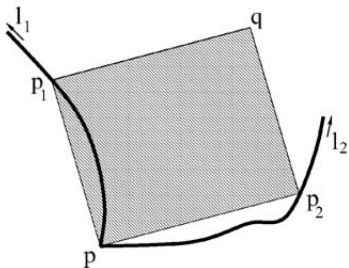
### Principe

- Un coin  $p$  et deux points  $p_1(l_1)$  et  $p_2(l_2)$  se déplaçant le long des contours partant de ce coin
- Distance curviligne  $l_{1,2}$  de chaque point  $p_{1,2}$  au coin  $p$  = surface entre le contour et la droite joignant le coin et le point  $p_{1,2}$
- Distance invariante aux transformations affines
- Un coin  $p$  et deux points  $p_1(l_1)$  et  $p_2(l_2)$  = une famille de parallélogrammes



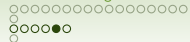
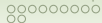
# Edge-Based Regions (EBR)

## Analyse des coins



## *Edge-Based Regions (EBR)*

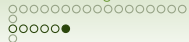
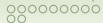
- Dans cette famille de parallélogrammes, sélection de celui qui a une valeur minimale d'une fonction photométrique



## Edge-Based Regions (EBR)

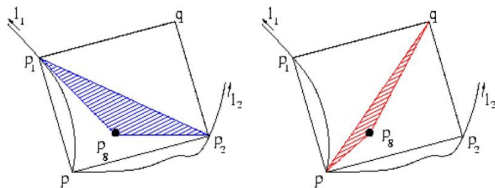
- Dans cette famille de parallélogrammes, sélection de celui qui a une valeur minimale d'une fonction photométrique
- Fonction : distance entre le centre de **gravité et les diagonales du parallélogramme**

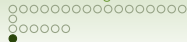
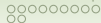




# Edge-Based Regions (EBR)

## Illustration du critère associé





## PCBR Principal Curvature-Based Regions

- Estimer les courbures principales estimées à différentes échelles
- Améliorer cette détection en utilisant
  1. des opérateurs morphologiques pour fermer les contours
  2. un seuillage par hystérésis des courbures
  3. la conservation uniquement si la courbure est élevée sur différentes échelles
- Détecter des **lignes de partages des eaux** (à partir des courbures)

# Plan

Notations

Construction d'une pyramide

Détecteurs du premier ordre

Détecteurs du premier ordre : vers le multi-échelle

Détecteurs de régions

Vers le multi-échelle : Détecteur de Kadir

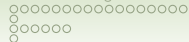
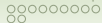
Vers le multi-échelle : EBR, *Edge-Based Regions*

Vers le multi-échelle : PCBR, *Principal Curvature-Based Regions*

Détecteurs du second ordre

Vers le multi-échelle : SIFT *Scale Invariant Feature Transform*

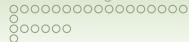
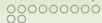
Vers une variante accélérée de SIFT : SURF



# DéTECTEUR de Beaudet

**Courbure de la surface** : minimiser le déterminant de la matrice hessienne

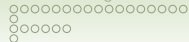
$$\text{Beaudet}(x, y) = \det \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} = I_{xx}I_{yy} - I_{xy}^2$$



# Détecteur de Kitchen et Rosenfeld

**Courbure du contour pondéré par la norme du gradient**

$$KR(x, y) = \frac{I_{xx}I_y^2 - 2I_{xy}I_xI_y + I_{yy}I_x^2}{I_x^2 + I_y^2}$$



# Courbure et espace d'échelle : *Curvature Scale Space* (CSS)

- **Principe :** recherche les extrema de la fonction courbure le long des contours en utilisant une approche multi-échelle

# Courbure et espace d'échelle : *Curvature Scale Space* (CSS)

- **Principe** : recherche les extrema de la fonction courbure le long des contours en utilisant une approche multi-échelle
- **Étapes**
  1. Détection par Canny
  2. Fusion des contours proches
  3. Calcul de la courbure des contours avec lissage par une gaussienne
  4. Détection des extrema dans cet espace

# Courbure et espace d'échelle : *Curvature Scale Space* (CSS)

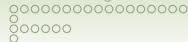
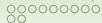
- **Principe** : recherche les extrema de la fonction courbure le long des contours en utilisant une approche multi-échelle
- **Étapes**
  1. Détection par Canny
  2. Fusion des contours proches
  3. Calcul de la courbure des contours avec lissage par une gaussienne
  4. Détection des extrema dans cet espace
- **Variante** : Distinction entre contours courts et longs (détermination automatique du seuil de décision)

# Courbure et espace d'échelle : *Curvature Scale Space* (CSS)

- **Principe** : recherche les extrema de la fonction courbure le long des contours en utilisant une approche multi-échelle
- **Étapes**
  1. Détection par Canny
  2. Fusion des contours proches
  3. Calcul de la courbure des contours avec lissage par une gaussienne
  4. Détection des extrema dans cet espace
- **Variante** : Distinction entre contours courts et longs (détermination automatique du seuil de décision)
- **Invariance** : Transformations affines géométriques et photométriques

# SIFT *Scale Invariant Feature Transform*

## 1. Détection des extremums en **échelle et en espace**

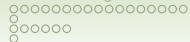
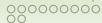


## SIFT *Scale Invariant Feature Transform*

1. Détection des extremums en **échelle et en espace**
2. Caractérisation par une **orientation principale** = gradient dominant dans le voisinage du point considéré, avec prise en compte du bruit

## SIFT *Scale Invariant Feature Transform*

1. Détection des extremums en **échelle et en espace**
2. Caractérisation par une **orientation principale** = gradient dominant dans le voisinage du point considéré, avec prise en compte du bruit
3. Ajout d'un descripteur associé prenant en compte un **histogramme local de ces orientations**



# Propriétés intéressantes

1. **Détecteur complet** : un descripteur associé permettant mettre en correspondance de manière robuste

# Propriétés intéressantes

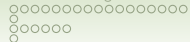
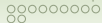
1. **Détecteur complet** : un descripteur associé permettant mettre en correspondance de manière robuste
2. De nombreuses propriétés
  - 2.1 Localisation précise
  - 2.2 robustes aux occultations et aux bruits
  - 2.3 invariance aux changements de luminosité
  - 2.4 robustesse aux changements de points de vue

## Propriétés intéressantes

1. **Détecteur complet** : un descripteur associé permettant mettre en correspondance de manière robuste
2. De nombreuses propriétés
  - 2.1 Localisation précise
  - 2.2 robustes aux occultations et aux bruits
  - 2.3 invariance aux changements de luminosité
  - 2.4 robustesse aux changements de points de vue

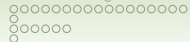
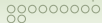
MAIS

3. **Non invariant aux transformations affines**



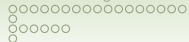
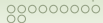
# Étapes

1. Détection des extremums en échelle et en espace
2. Localisation des points
3. Affectation d'une orientation
4. Estimation des descripteurs



# Étapes

1. Détection des extremums en échelle et en espace : Détection
2. Localisation des points : Affinement de la détection
3. Affectation d'une orientation : Caractérisation
4. Estimation des descripteurs : Pour la mise en correspondance



# Détection des extremums en échelle et en espace

- Approche par **filtres en cascade**

# Détection des extremums en échelle et en espace

- Approche par **filtres en cascade**
- Estimation de différences de gaussiennes = **approximation du laplacien**

$$\begin{aligned} DoG(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

où  $G(x, y, \sigma)$  est la fonction gaussienne

# Détection des extremums en échelle et en espace

- Approche par **filtres en cascade**
- Estimation de différences de gaussiennes = **approximation du laplacien**

$$\begin{aligned} DoG(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

où  $G(x, y, \sigma)$  est la fonction gaussienne

- Pour des valeurs successives de  $\sigma$  : laplacien normalisé en échelle

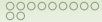
## Détection des extremums en échelle et en espace

- **Définition d'une octave** = Ensemble d'échelles  
**ensemble des images lissées par un filtre gaussien à différentes échelles pour une résolution donnée**

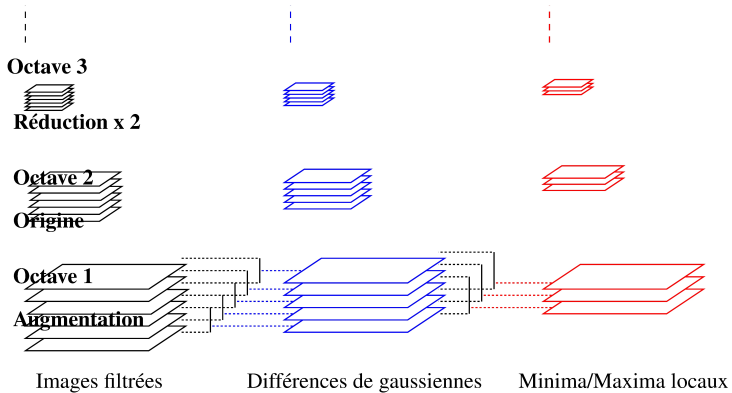
Pour passer d'une octave à l'autre = double la valeur de  $\sigma$

Donc : si on souhaite  $n$  échelles par octave, alors il faut fixer  
 $k = 2^{1/n}$





# Détection





# Affinement et correction de la localisation des points

## 1. Calcul de la position au sous-pixel

Estimation du décalage et re-calage si nécessaire

# Affinement et correction de la localisation des points

## 1. Calcul de la position au sous-pixel

Estimation du décalage et re-calage si nécessaire

## 2. Suppression des points faiblement contrastés

Différence (après filtrage) entre le point initial et le point re-localisé

# Affinement et correction de la localisation des points

## 1. Calcul de la position au sous-pixel

Estimation du décalage et re-calage si nécessaire

## 2. Suppression des points faiblement contrastés

Différence (après filtrage) entre le point initial et le point re-localisé

## 3. Suppression des points contours

Utilisation de la matrice Hessienne (comme Harris)

## Position sub-pixellique ou relocalisation

**En quoi consiste la relocalisation lorsque la détection n'est pas optimale au cours du test de la répétabilité ?**

- **But :** Contrecarrer les effets des lissages

## Position sub-pixellique ou relocalisation

**En quoi consiste la relocalisation lorsque la détection n'est pas optimale au cours du test de la répétabilité ?**

- **But :** Contrecarrer les effets des lissages
- **Position calculée au sous-pixel :** Développement de Taylor à l'ordre 2

$$D(\hat{\mathbf{p}}) = D(\mathbf{p}) + \frac{\partial D(\mathbf{p})^T}{\partial \mathbf{p}} + \frac{1}{2} \mathbf{p}^T \frac{\partial^2 D(\mathbf{p})}{\partial \mathbf{p}^2} \mathbf{p}, \quad (5)$$



## Position sub-pixellique ou relocalisation

**En quoi consiste la relocalisation lorsque la détection n'est pas optimale au cours du test de la répétabilité ?**

- **But** : Contrecarrer les effets des lissages
- **Position calculée au sous-pixel** : Développement de Taylor à l'ordre 2

$$D(\hat{\mathbf{p}}) = D(\mathbf{p}) + \frac{\partial D(\mathbf{p})^T}{\partial \mathbf{p}} + \frac{1}{2} \mathbf{p}^T \frac{\partial^2 D(\mathbf{p})}{\partial \mathbf{p}^2} \mathbf{p}, \quad (5)$$

- Localisation précise de l'extremum = passage par zéro de la dérivée suivant  $\mathbf{p}$

$$\hat{\mathbf{p}} = - \frac{\partial^2 D(\mathbf{p})^{-1}}{\partial \mathbf{p}^2} \frac{\partial D(\mathbf{p})}{\partial \mathbf{p}}. \quad (6)$$

- Si le décalage est plus grand que 0.5 dans chaque direction = relocalisation

## Filtrage des minima/maxima locaux

**Première contrainte pour la sélection des minima/maxima locaux**  
**Élimination des points dont le contraste est trop faible**

## Filtrage des minima/maxima locaux

**Première contrainte pour la sélection des minima/maxima locaux**  
**Élimination des points dont le contraste est trop faible**

- Si on utilise :

$$D(\hat{\mathbf{p}}) = D(\mathbf{p}) + \frac{1}{2} \frac{\partial D(\mathbf{p})^T}{\partial \mathbf{p}} \hat{\mathbf{p}} \quad (7)$$

## Filtrage des minima/maxima locaux

### Première contrainte pour la sélection des minima/maxima locaux Élimination des points dont le contraste est trop faible

- Si on utilise :

$$D(\hat{\mathbf{p}}) = D(\mathbf{p}) + \frac{1}{2} \frac{\partial D(\mathbf{p})^T}{\partial \mathbf{p}} \hat{\mathbf{p}} \quad (7)$$

- Expérimentalement : élimination des points tels que  $|D(\hat{\mathbf{p}})| < 0.03$   
En supposant une variation entre 0 et 1

## Filtrage des minima/maxima locaux

**Deuxième contrainte pour la sélection des minima/maxima locaux**  
**Points sur les contours**

## Filtrage des minima/maxima locaux

### Deuxième contrainte pour la sélection des minima/maxima locaux Points sur les contours

- Même chose que pour l'approche de Harris

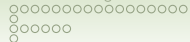
## Filtrage des minima/maxima locaux

### Deuxième contrainte pour la sélection des minima/maxima locaux Points sur les contours

- Même chose que pour l'approche de Harris
- Seuls les points où le rapport entre la plus grande et la plus petite valeur propre est inférieur à un seuil  $r$  sont conservés

$$\frac{\text{Trace}(M)^2}{\text{Det}(M)} < \frac{(r+1)^2}{r} \quad (8)$$

- $r = 10$



# Speeded Up Robust Features, SURF

- **Intérêt :** Multi-échelle moins coûteux mais équivalent à SIFT

# Speeded Up Robust Features, SURF

- **Intérêt** : Multi-échelle moins coûteux mais équivalent à SIFT
- **Réponse** : approximation du déterminant de la matrice hessienne à plusieurs échelles

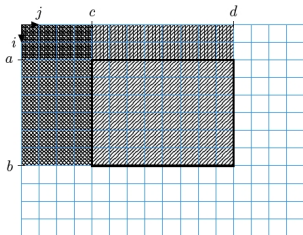
## Speeded Up Robust Features, SURF

- **Intérêt** : Multi-échelle moins coûteux mais équivalent à SIFT
- **Réponse** : approximation du déterminant de la matrice hessienne à plusieurs échelles
- **Outils**
  - Lissage gaussien réalisé par des **box filter**
  - Utilisation des **images intégrales**

## Speeded Up Robust Features, SURF

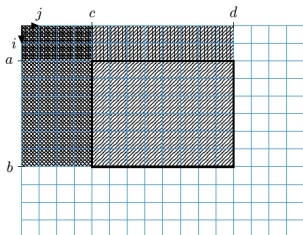
- **Intérêt** : Multi-échelle moins coûteux mais équivalent à SIFT
  - **Réponse** : approximation du déterminant de la matrice hessienne à plusieurs échelles
  - **Outils**
    - Lissage gaussien réalisé par des **box filter**
    - Utilisation des **images intégrales**
  - **En pratique**
    - Calcul par convolution est déjà une approximation
- MAIS** Résultats exploitables
- Utilisation d'une implémentation moins coûteuses

# Notion d'image intégrale



$$I_{int}(i, j) = \sum_{k=0}^{i-1} \sum_{l=0}^{j-1} I(k, l). \tag{9}$$

## Notion d'image intégrale



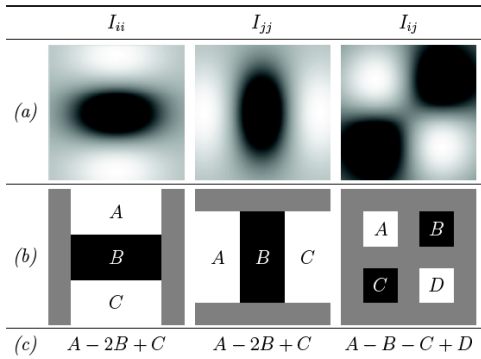
$$I_{int}(i, j) = \sum_{k=0}^{k < i} \sum_{l=0}^{l < j} I(k, l). \quad (9)$$

Permet de rendre rapide le calcul de sommes dans un rectangle donné

$$\sum_{i=a}^b \sum_{j=c}^d I(i, j) = I_{int}(b, d) - I_{int}(a, d) - I_{int}(b, c) + I_{int}(a, c). \quad (10)$$

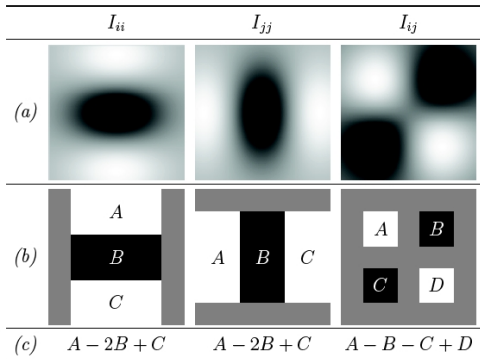
## Box filtering

- Masque des dérivées = combinaisons de sommes de valeurs de pixels situés dans différents rectangles (*box*)

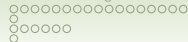
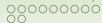


## Box filtering

- Masque des dérivées = combinaisons de sommes de valeurs de pixels situés dans différents rectangles (*box*)

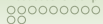


- L'utilisation de l'image intégrale réduit le nombre d'opérations arithmétiques

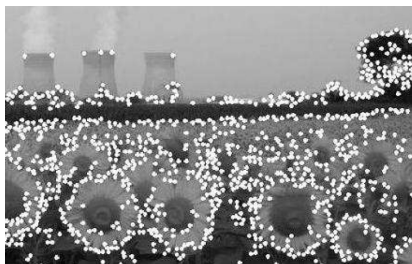


# BRISK

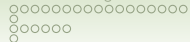
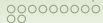
- Considération d'un voisinage circulaire
- Sélection d'un ensemble de points bien répartis dans ce voisinage
- Comparaison des intensités de ces points avec le point étudié
- Le descripteur = chaîne de bits indiquant le signe des comparaisons



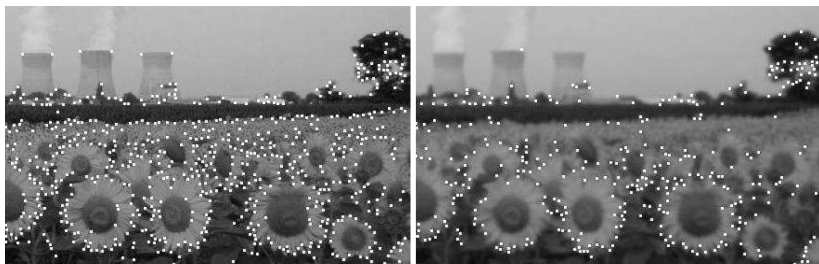
## Comparaison des différents détecteurs



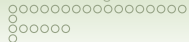
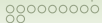
Beudet



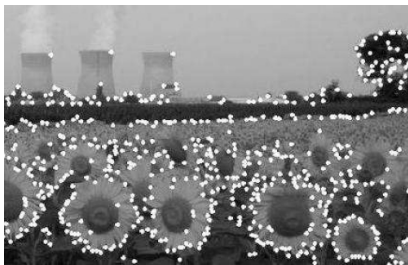
## Comparaison des différents détecteurs



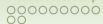
SUSAN



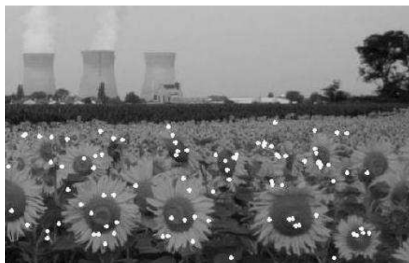
## Comparaison des différents détecteurs



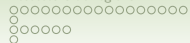
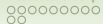
Harris



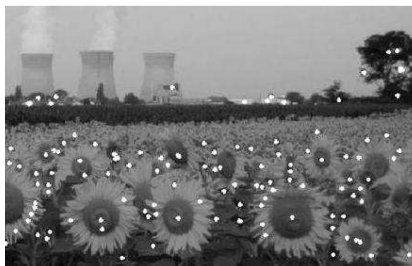
## Comparaison des différents détecteurs



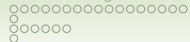
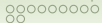
EBR



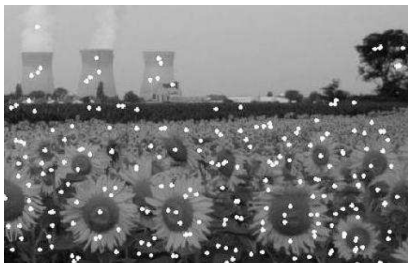
## Comparaison des différents détecteurs



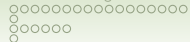
MSER



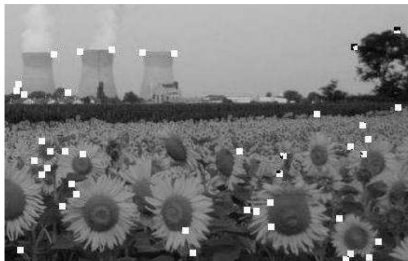
## Comparaison des différents détecteurs



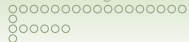
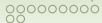
IBR



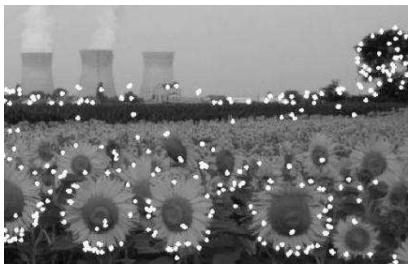
## Comparaison des différents détecteurs



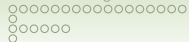
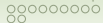
CSS



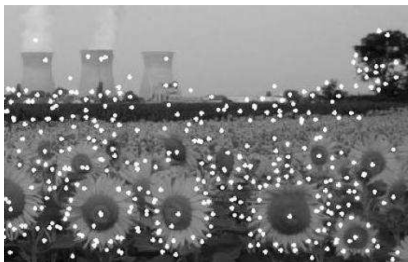
## Comparaison des différents détecteurs



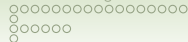
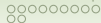
SIFT



## Comparaison des différents détecteurs



SURF



**Est-ce qu'il y a des détecteurs plus adaptés pour ensuite faire correspondre des points d'intérêt ou est-ce indépendant? Par exemple SIFT ?**

- *Certains détecteurs n'ont pas de descripteur et/ou méthode d'appariement spécifiques associés.*
- *On peut donc choisir toutes les techniques que l'on souhaite parmi celles présentées dans la partie IV, sur la mise en correspondance.*