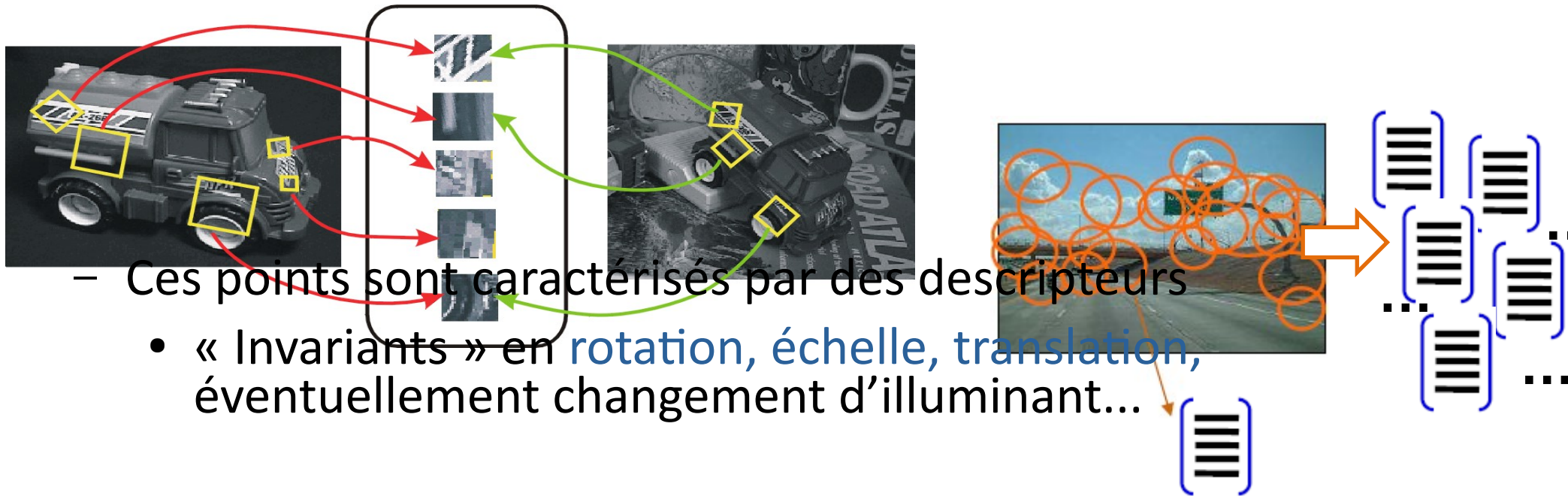


Points d'intérêt

- Une image est décrite :
 - par un ensemble de points (pixels) remarquables : POI, keypoints, points clefs
 - Souvent des coins, des blobs...



Types d'invariances



illumination

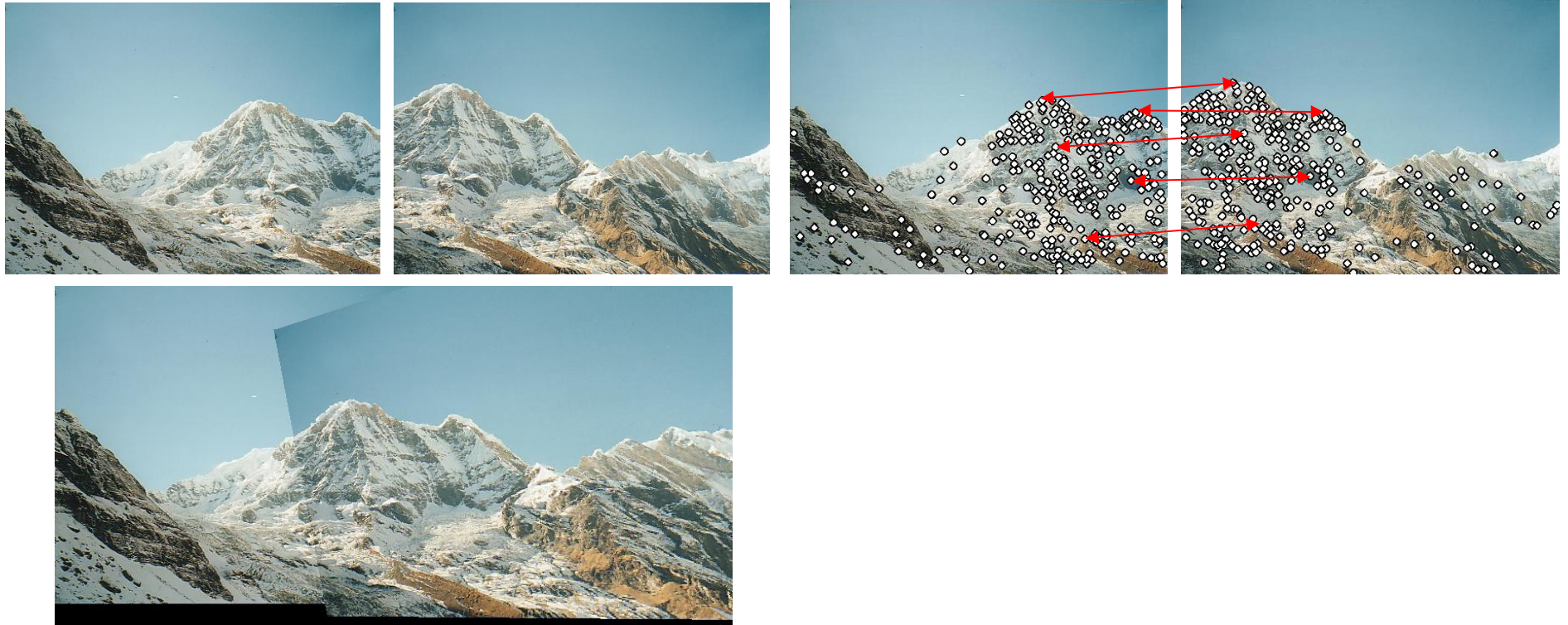


échelle



rotation

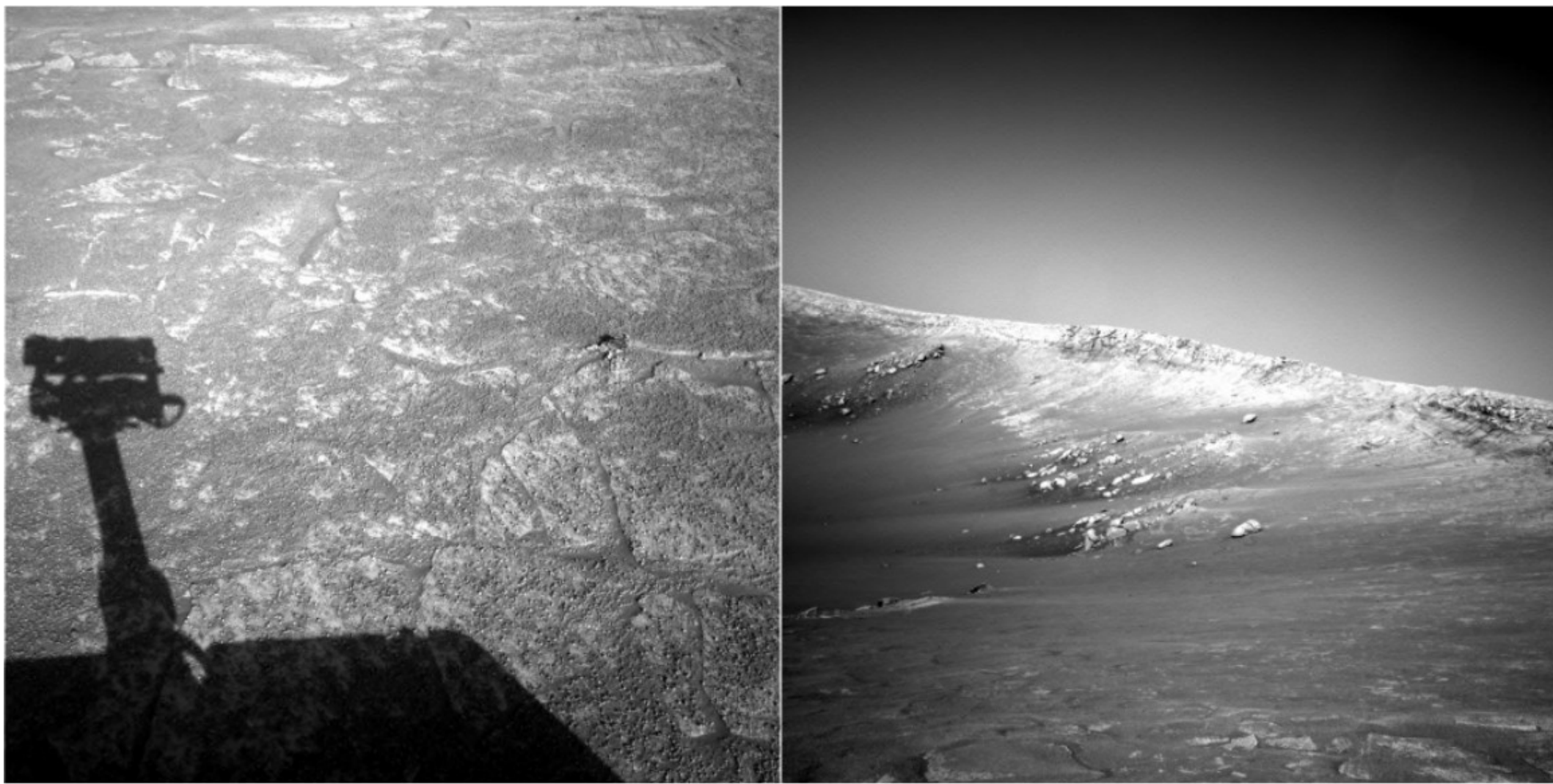
- Utilisés dans de nombreuses applications : reconnaissance d'objet, assemblage d'images (**stitching**), modélisation 3D, recherche d'image par le contenu, tracking video, etc.



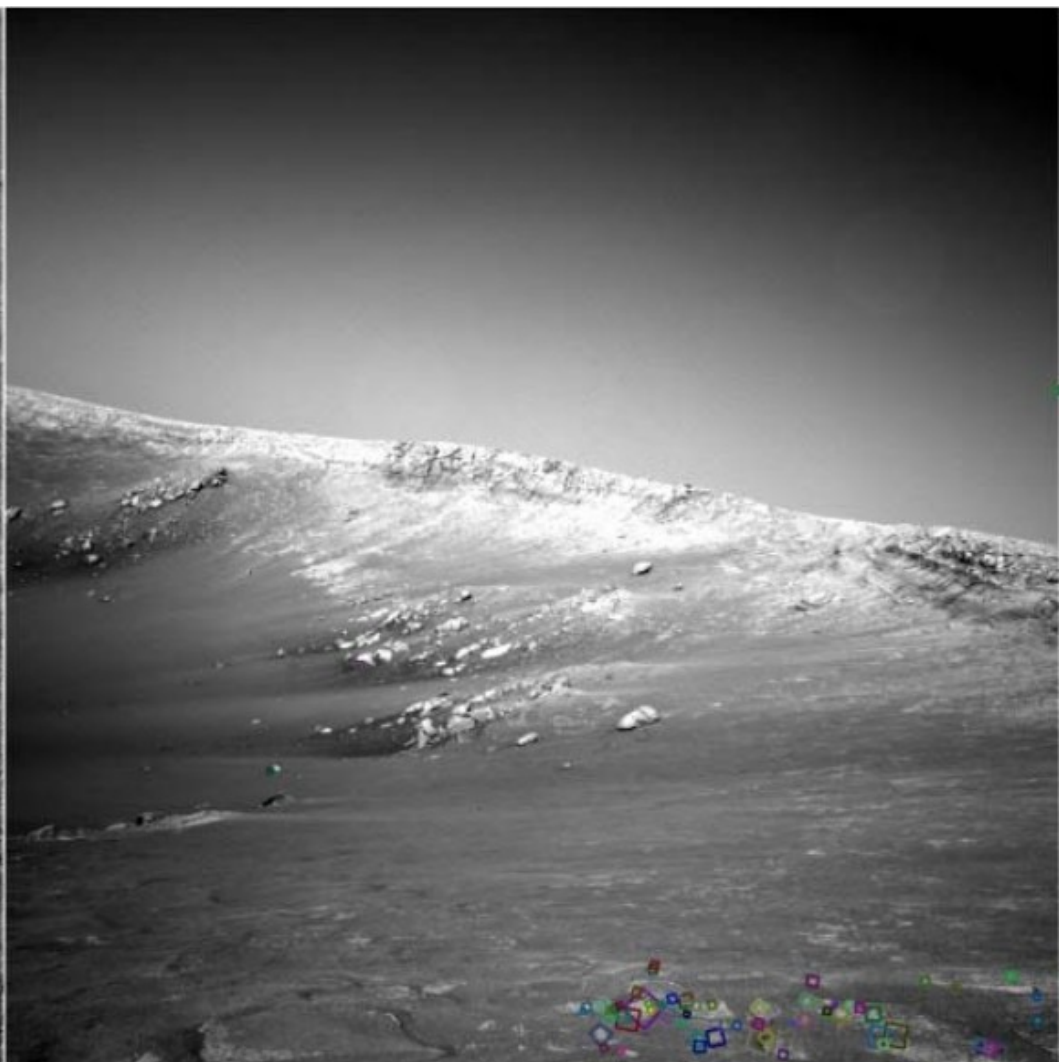
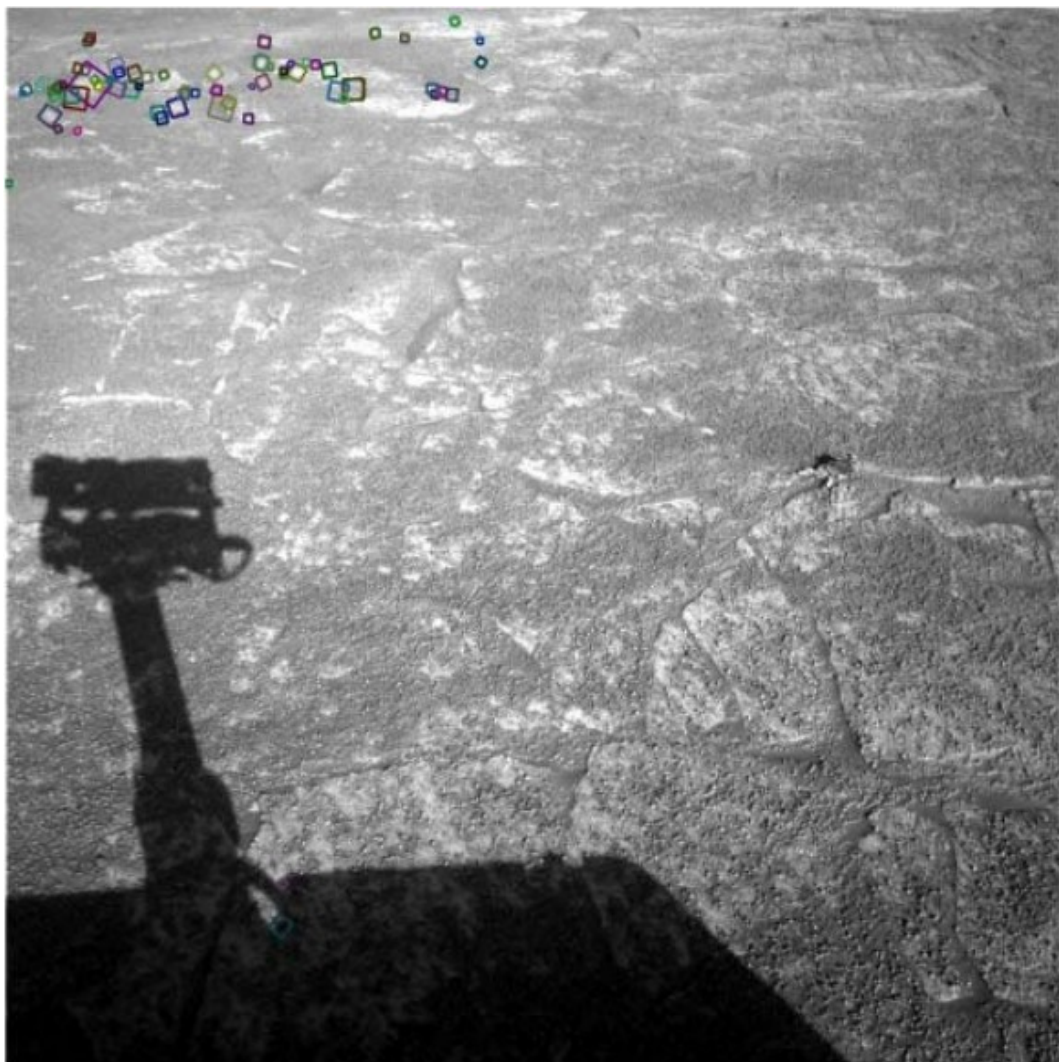
- Popularisés par **SIFT** (scale-invariant feature transform) :

David G. Lowe,

« Object recognition from local scale-invariant features », Proceedings of the International Conference on Computer Vision, vol. 2, 1999, p. 1150–1157.



NASA Mars Rover images



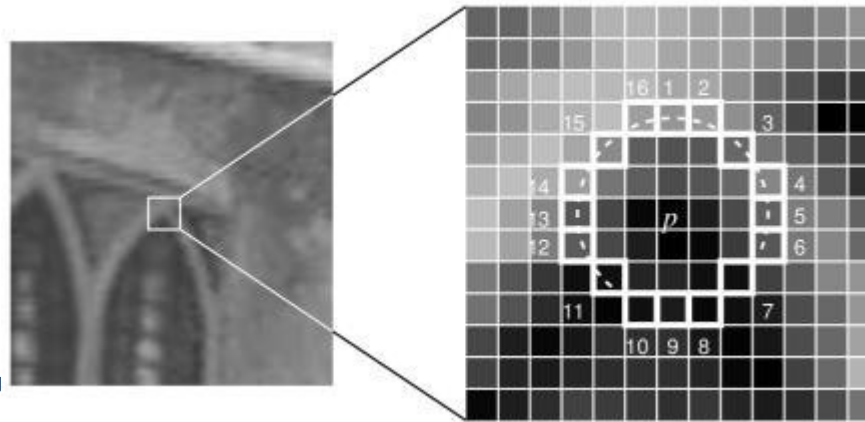
NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

ORB (Oriented FAST and Rotated BRIEF)

- OpenCV labs by Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski in 2011,
- Alternative à SIFT (et SURF) qui sont brevetés.
- ORB est aussi efficace et deux ordres de grandeur plus rapide.
- Repose sur deux algorithmes :
 - **FAST** (modifié) pour la détection des points d'intérêts (POI ou keypoints)
 - **BRIEF** pour le calcul des descripteurs

FAST (Features from Accelerated Segment Test)

- t est un seuil fixé
- Pour chaque pixel p de l'image, dont le niveau de gris est noté I_p :
 - On considère le cercle de 16 pixels autour de p (Cercle discret de rayon 3.)

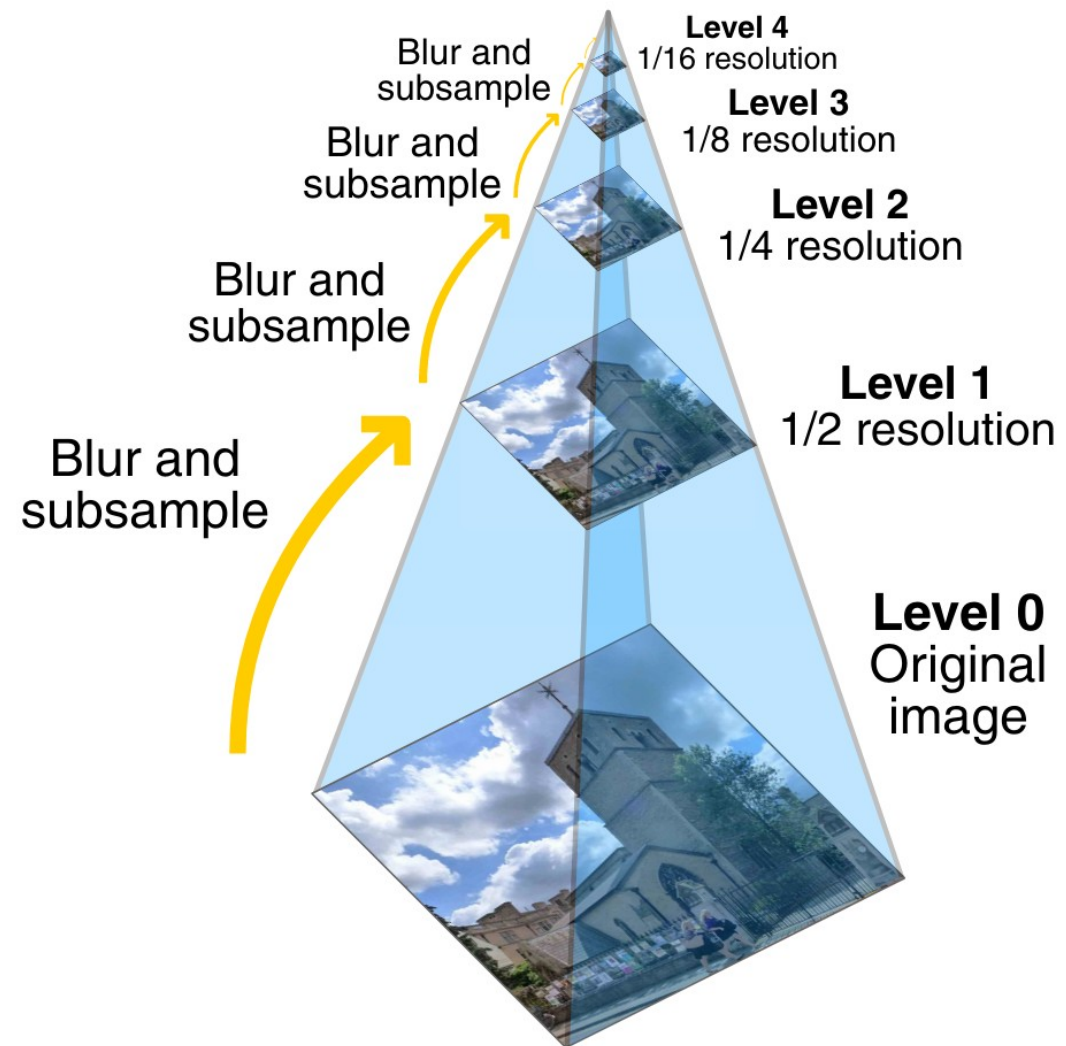


- p est un **point d'** plus clairs ou plus foncés que p :
 - Un pixel est foncé si son intensité est supérieure à $I_p + t$
 - Un pixel est clair si son intensité est inférieure à $I_p - t$

cercle consécutifs

- → invariance par changement d'illuminant
- Dans sa version originale, cet algorithme détecte trop de points, il a donc été amélioré :
 - Sélection des configurations les plus pertinentes par apprentissage (ID3)
 - Suppression des non-maxima si des POI sont trop proches :
 - Calcul de la somme des différences absolues entre un POI et les pixels du cercle
 - Seul le POI de plus grande valeur est conservé.
 - Image préalablement lissée (gaussien)

- Les points d'intérêt (POI) sont détectés à différentes **échelles** :
 - Pyramide par sous-échantillonnage et filtrage
 - Porte rectangulaire
 - Ou porte gaussienne



→ **invariance en échelle**

Rappel - Convolution 2D

généralisation de l'idée de **moyenne mobile d'une image f par un masque g**

En continu :

$$(f * g)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u, v) g(x - u, y - v) du dv$$

En discret :

$$(f * g)[x, y] = \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} f[u, v] g[x - u, y - v]$$

En pratique (l'image n'est pas de taille infinie..)

$$(f * g)[x, y] = \sum_{i=-r}^r \sum_{j=-r}^r f[i + x, j + y] g[r - i, r - j]$$

Pour une image de taille (m,n) où $(x,y) \in [0..m-1] \times [0..n-1]$
Et un masque de taille impaire $(2r+1, 2r+1)$

Rappel - Convolution 2D

Avant application sur l'image on prend le symétrique du masque sinon c'est une corrélation

En continu :

$$(f * g)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u, v) g(x - u, y - v) du dv$$

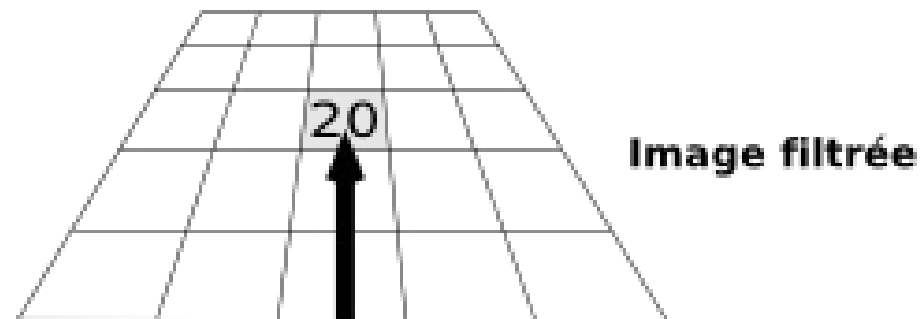

En discret :

$$(f * g)[x, y] = \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} f[u, v] g[x - u, y - v]$$

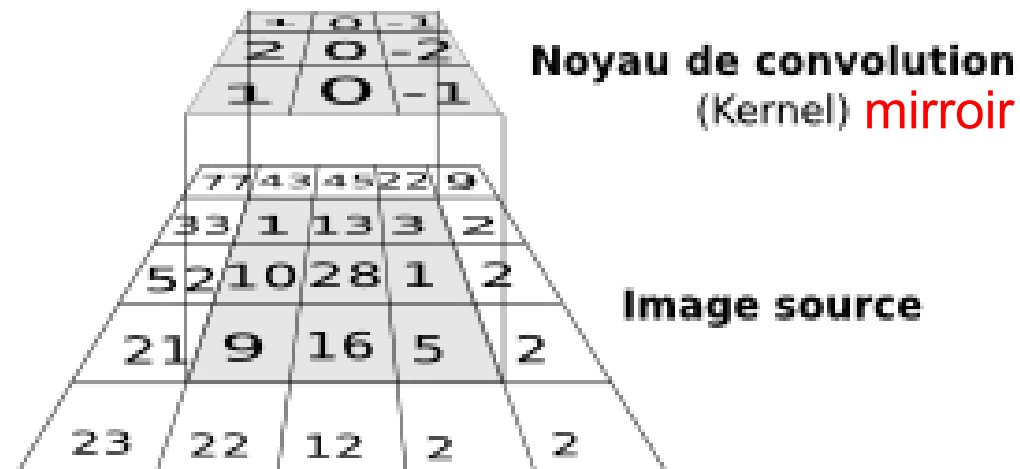
En pratique (l'image n'est pas de taille infinie..)

$$(f * g)[x, y] = \sum_{i=-r}^r \sum_{j=-r}^r f[i + x, j + y] g[r - i, r - j]$$

Pour une image de taille (m,n) où $(x,y) \in [0..m-1] \times [0..n-1]$
Et un masque de taille impaire $(2r+1, 2r+1)$



$$\begin{aligned}
 &= 1 \cdot 1 + 0 \cdot 13 + -1 \cdot 3 + \\
 &\quad 2 \cdot 10 + 0 \cdot 28 + -2 \cdot 1 + \\
 &\quad 1 \cdot 9 + 0 \cdot 16 + -1 \cdot 5
 \end{aligned}$$



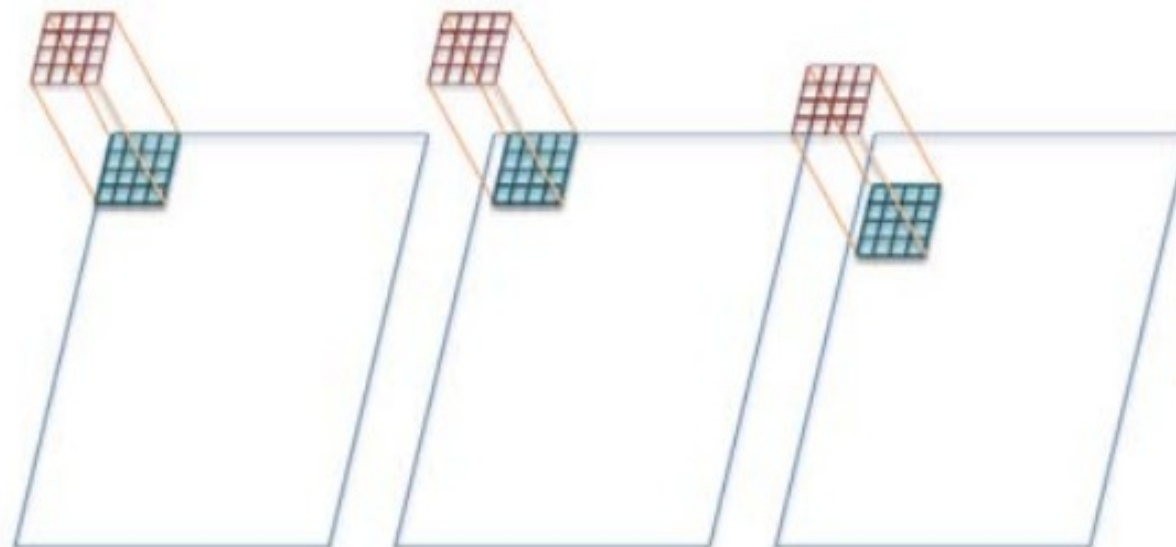


Image traitée

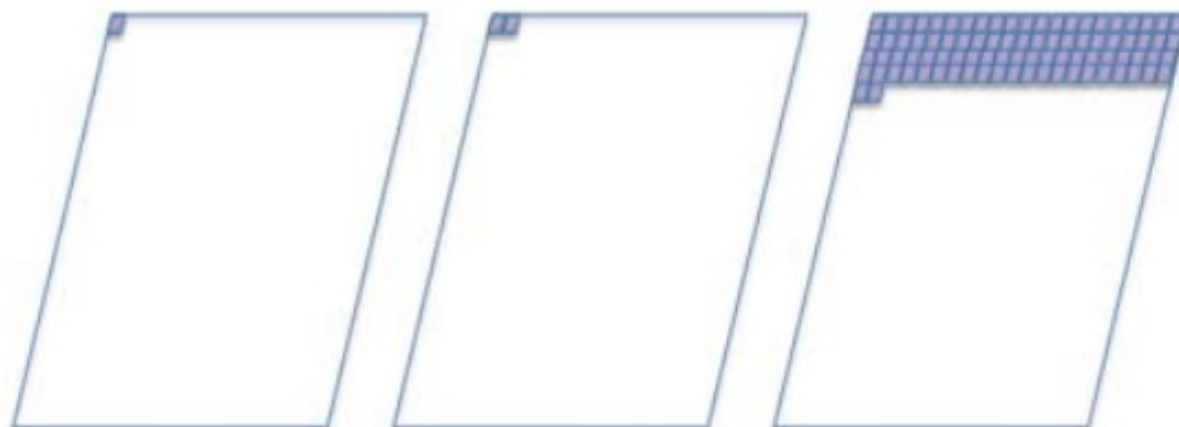


Image filtrée

Filtre-passe bas

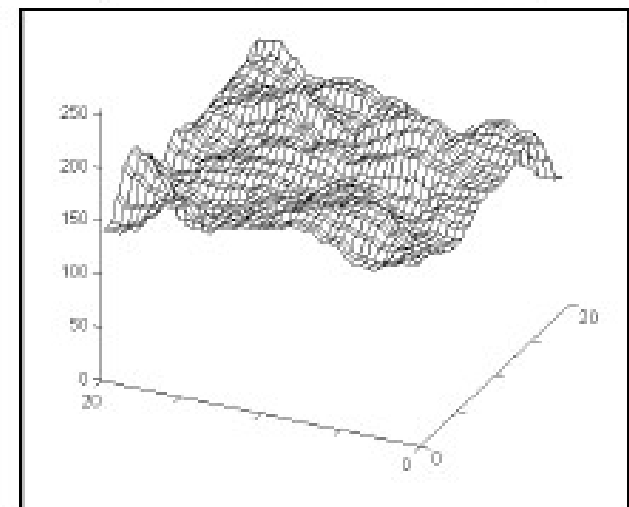
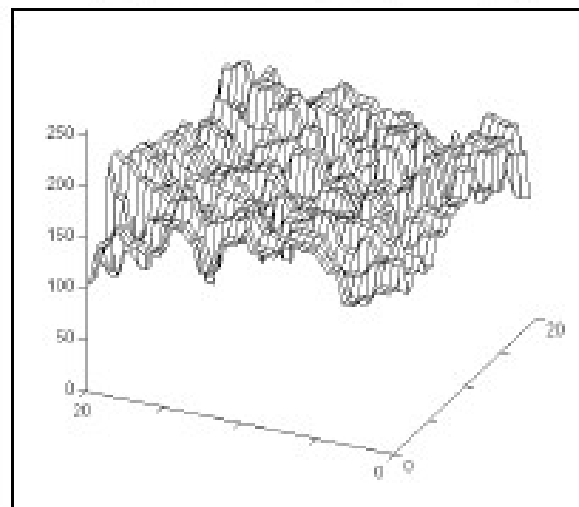
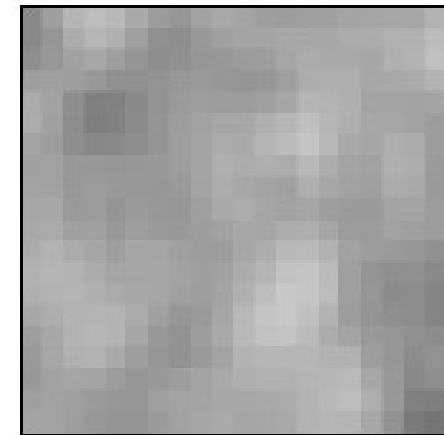
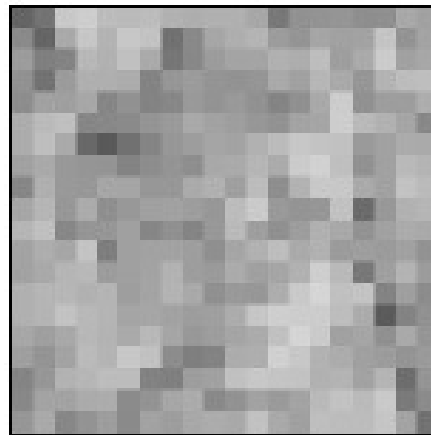
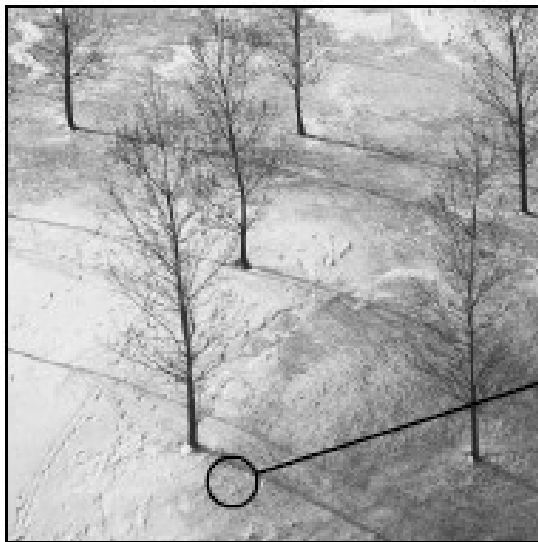
Filtre moyennneur
(blur ou box filter)

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Filtrage passe-bas (détails)

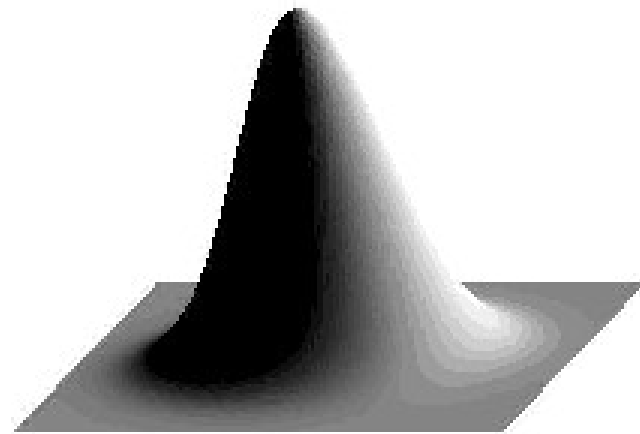
$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Filtre gaussien

- C'est le filtre passe-bas le plus utilisé
- Il est paramétrable :
 - Degré de lissage par la taille du filtre et la variance de la gaussienne

$$o(x,y) = I(x,y) \cdot \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



Filtre gaussien

■ Paramètres et calcul du noyau de convolution

$$o(x, y) = I(x, y) \cdot \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



$$W(i, j) = k \cdot \exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right)$$



$$\frac{W(i, j)}{k} = \exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right)$$

$$\sigma^2 = 2 \quad n = 7$$

		j						
		-3	-2	-1	0	1	2	3
i	-3	.011	.039	.082	.105	.082	.039	.011
	-2	.039	.135	.287	.368	.287	.135	.039
	-1	.082	.287	.606	.779	.606	.287	.082
	0	.105	.039	.779	1.000	.779	.368	.105
	1	.082	.287	.606	.779	.606	.287	.082
	2	.039	.135	.287	.368	.287	.135	.039
	3	.011	.039	.082	.105	.082	.039	.011

$$\frac{W(1,2)}{k} = \exp\left(-\frac{1^2 + 2^2}{2 \cdot 2}\right)$$

Représentation entière,
poser k=91 pour rendre
ce coefficient unitaire

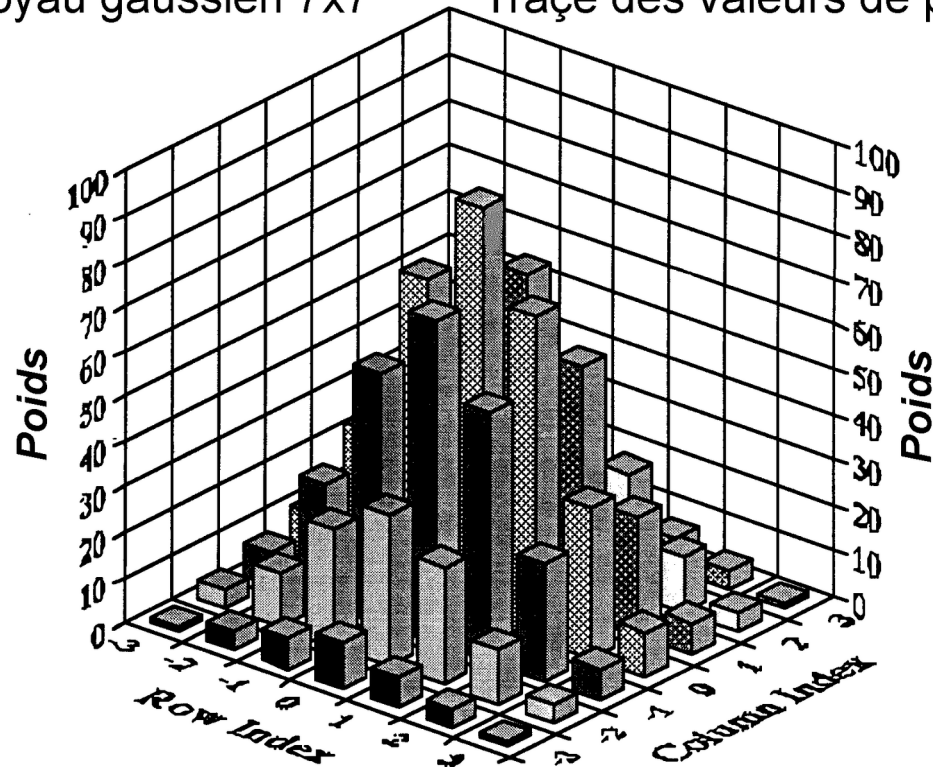
Filtre gaussien

1	4	7	10	7	4	1
4	12	26	33	26	12	4
7	26	55	71	55	26	7
10	33	71	91	71	33	10
7	26	55	71	55	26	7
4	12	26	33	26	12	4
1	4	7	10	7	4	1

$$\sum_{i=-3}^3 \sum_{j=-3}^3 W(i,j) = 1,115$$

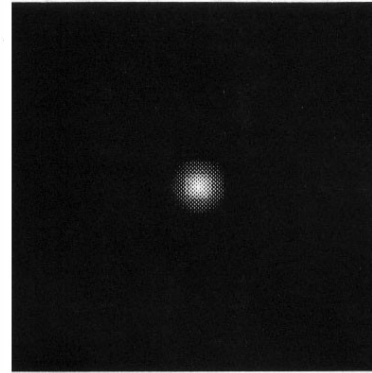
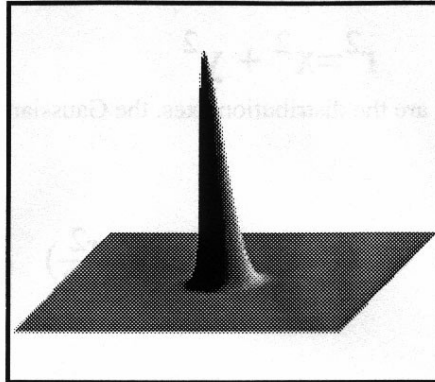
Noyau gaussien 7x7

Traçé des valeurs de poids



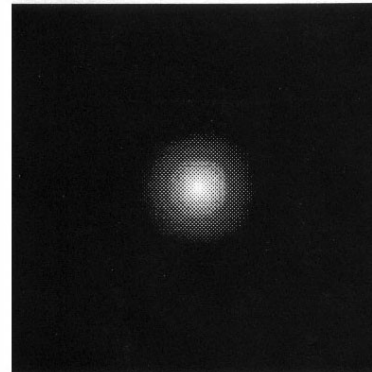
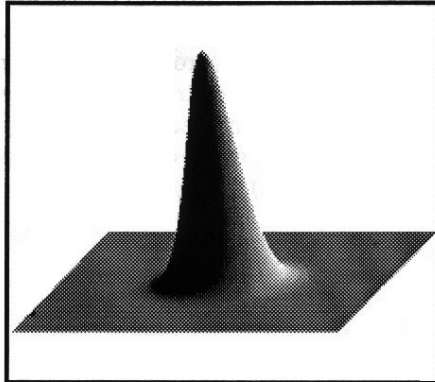
Filtre gaussien

$$\sigma^2 = .25$$



$$\begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix}$$

$$\sigma^2 = 1.0$$

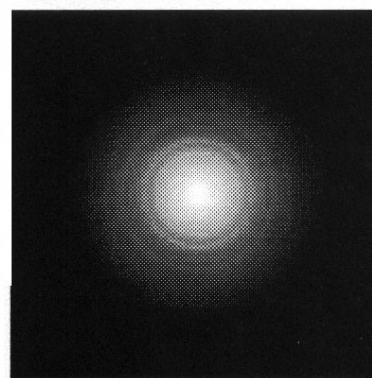
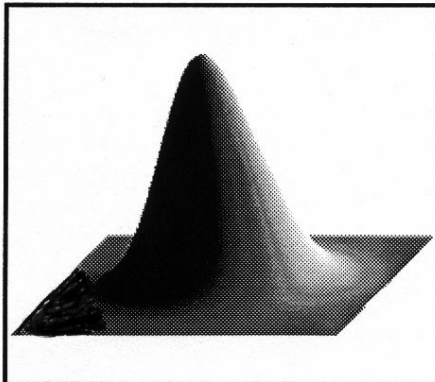


$$\begin{pmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{pmatrix}$$

Noyau
gaussien
7x7

$$\begin{bmatrix} 1 & 1 & 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 4 & 2 & 2 & 1 \\ 2 & 2 & 4 & 8 & 4 & 2 & 2 \\ 2 & 4 & 8 & 16 & 8 & 4 & 2 \\ 2 & 2 & 4 & 8 & 4 & 2 & 2 \\ 1 & 2 & 2 & 4 & 2 & 2 & 1 \\ 1 & 1 & 2 & 2 & 2 & 1 & 1 \end{bmatrix}$$

$$\sigma^2 = 4.0$$



Noyau gaussien 15x15

2	2	3	4	5	5	6	6	6	5	5	4	3	2	2
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
5	7	10	12	4	16	17	18	17	16	14	12	10	7	5
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
6	8	11	13	16	18	19	20	19	18	16	13	11	8	6
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
2	2	3	4	5	5	6	6	6	5	5	4	3	2	2



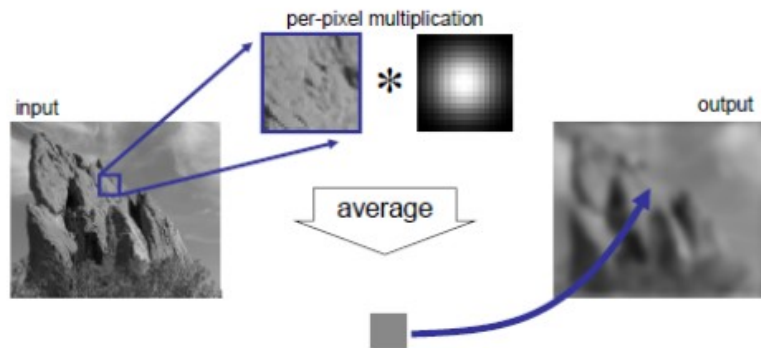
Noyau gaussien 7x7



Noyau gaussien 15x15

Gaussian Blur

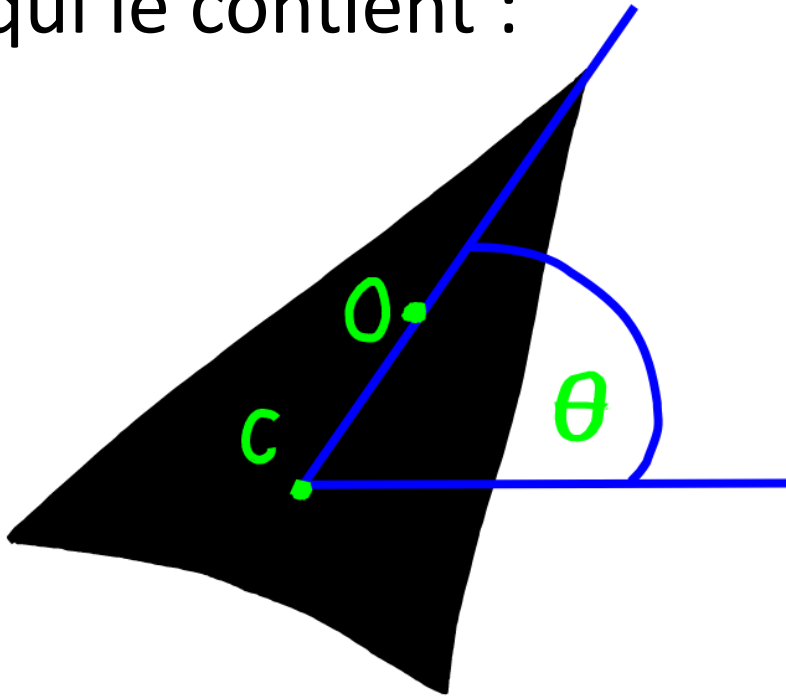
DigiVFX



Fin du rappel

Orientation

- L'idée est que le coin détecté **O** est décalé par rapport au centre de gravité **C** de l'imagette (**patch**) qui le contient :

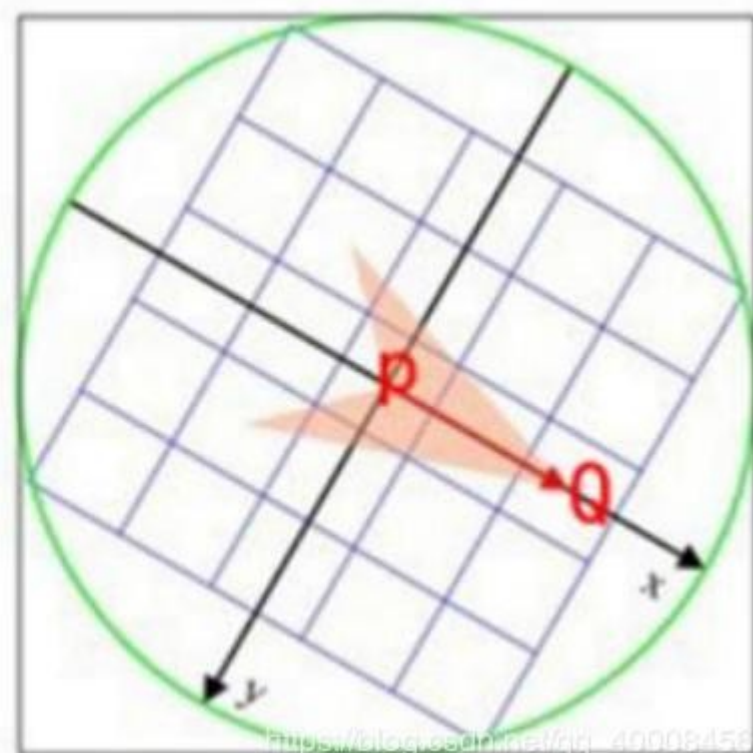
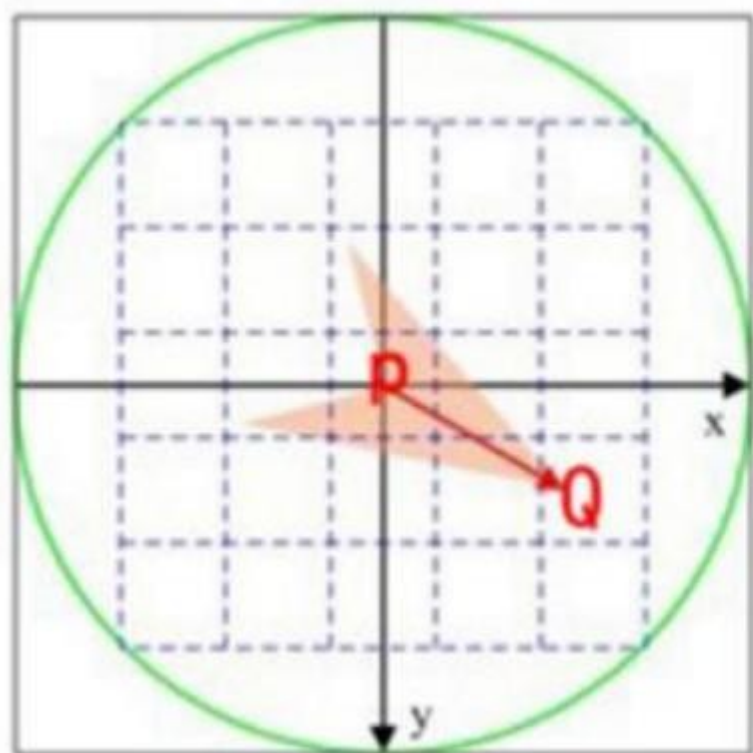


$$m_{pq} = \sum_{x,y} x^p y^q I(x, y)$$

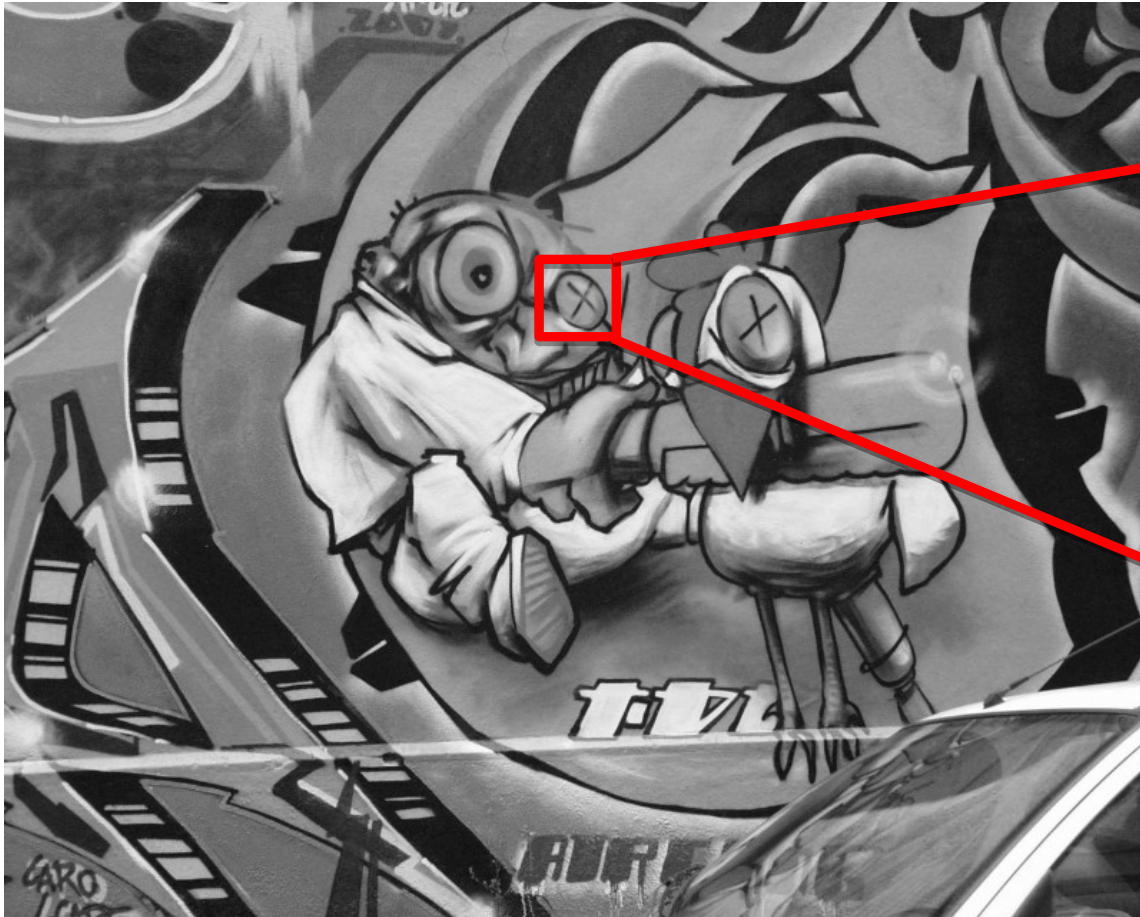
$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

$$\theta = \text{atan2}(m_{01}, m_{10})$$

- Le patch est réaligné sur un angle arbitraire avant le calcul du descripteur → **invariance en rotation**



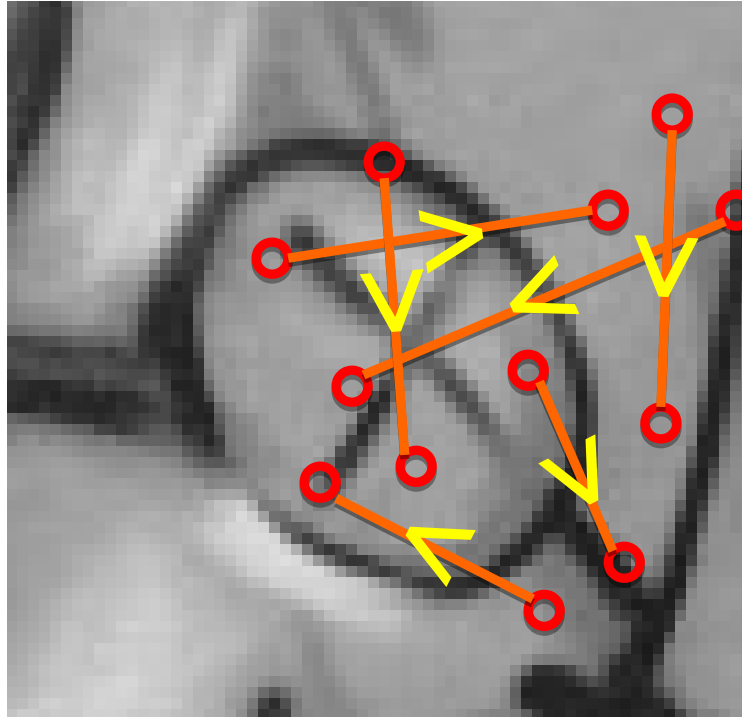
Brief (Binary robust independent elementary feature)



slide: J. Heinly

- Lissage gaussien (filtre les hautes fréquences)
- Sélection d'une paire de pixels tirés au hasard dans un voisinage (patch, ex : 37x37) autour du POI :
 - Le premier pixel est tiré aléatoirement selon une loi normale centrée sur le POI et avec un écart-type σ .
 - Le second est tiré aléatoirement selon une loi normale centrée sur le POI et avec un écart-type $\sigma/2$.
 - Si le premier point est plus clair que le second : bit à **1** sinon **0**.





Descripteur **binaire** 128 à 512 bits : **011010...**

Binary
Feature
Vectors

$V_1 = [01011100100110...$

$V_2 = [10010100110100...$

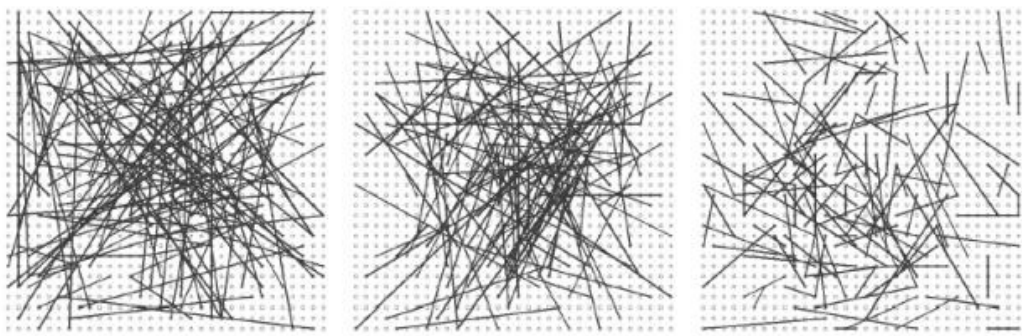
$V_3 = [11000100101110...$

$V_4 = [01011111100100...$

.

.

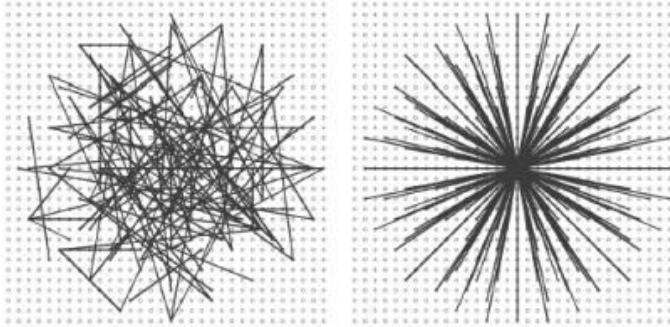
.



G I

G II

G III



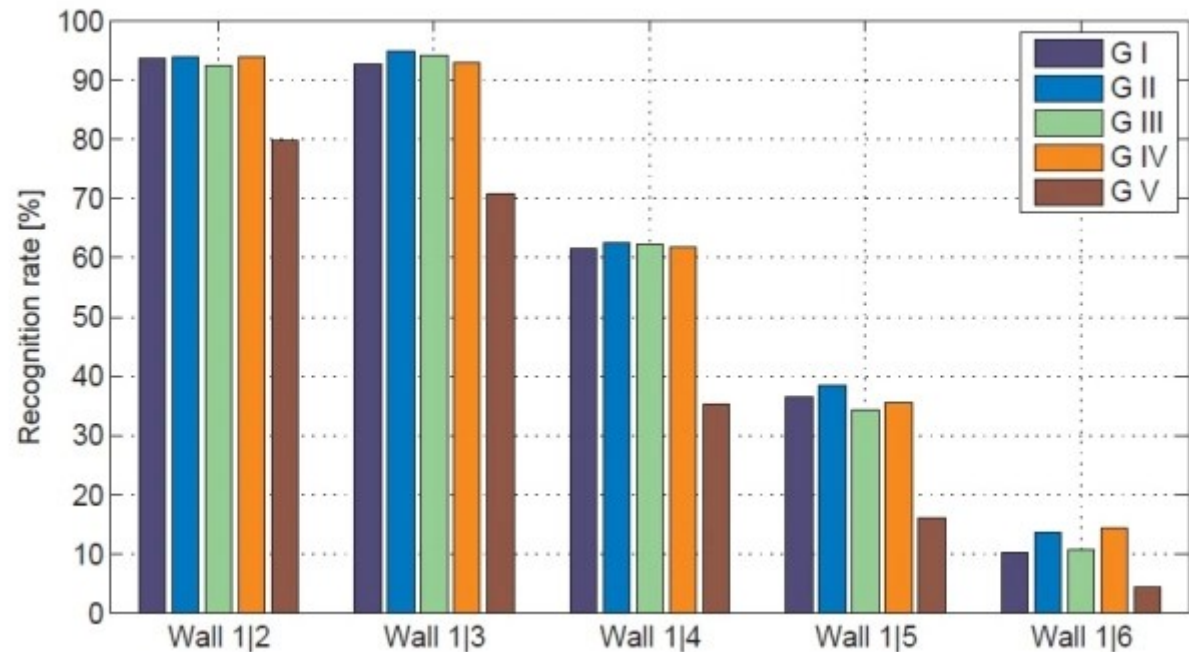
G IV

G V

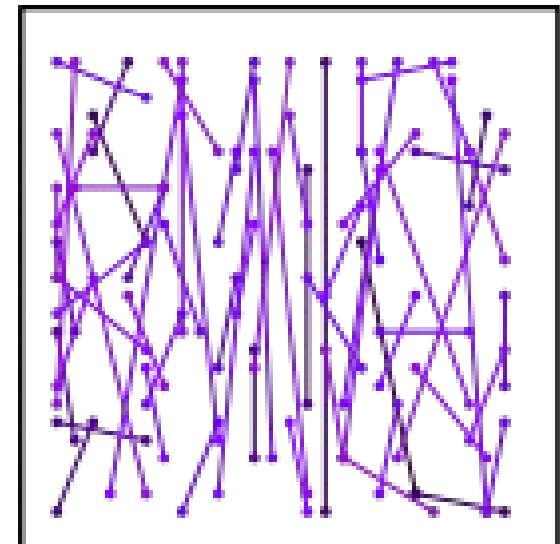
- uniforme
- gaussien (centré)
- gaussien
- grille discrète
- non aléatoire (grille fixée)

• Choix de la stratégie d'échantillonnage :

- Tests sur des base
(Pascal VOC 2006)

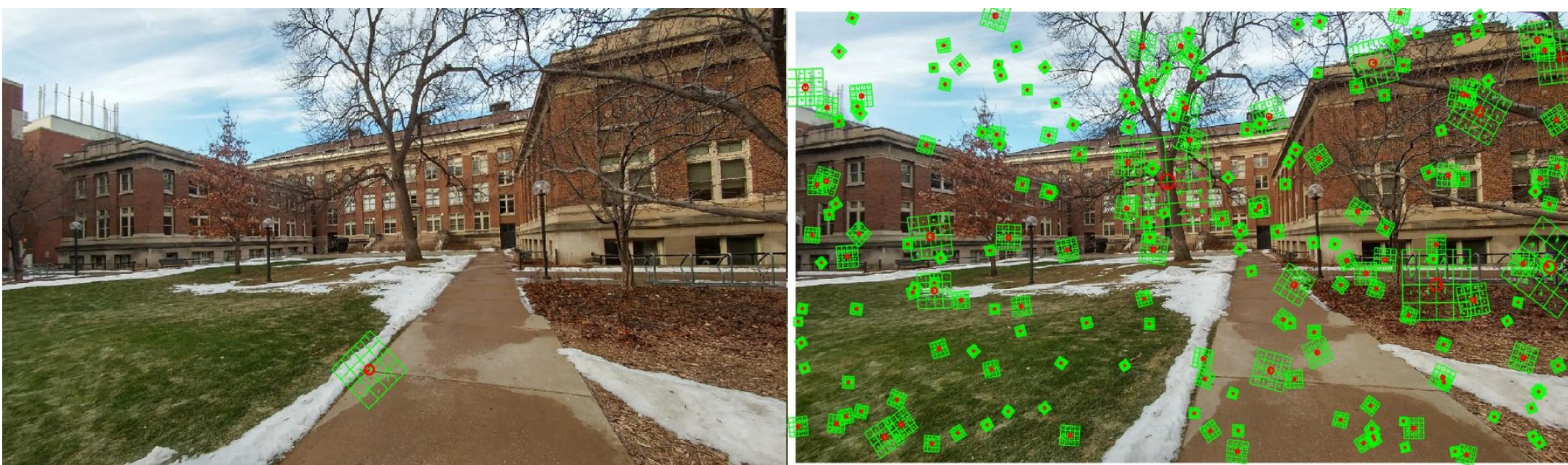


- L'algorithme précédent est exécuté 128,256, 512 fois selon la taille de la chaîne de bits que l'on veut utiliser.
- Le patch est toujours réaligné (grâce à l'angle θ)
- La sélection des paires de pixels est faite une seule fois : on garde les mêmes positions « aléatoires » pour tous les patches

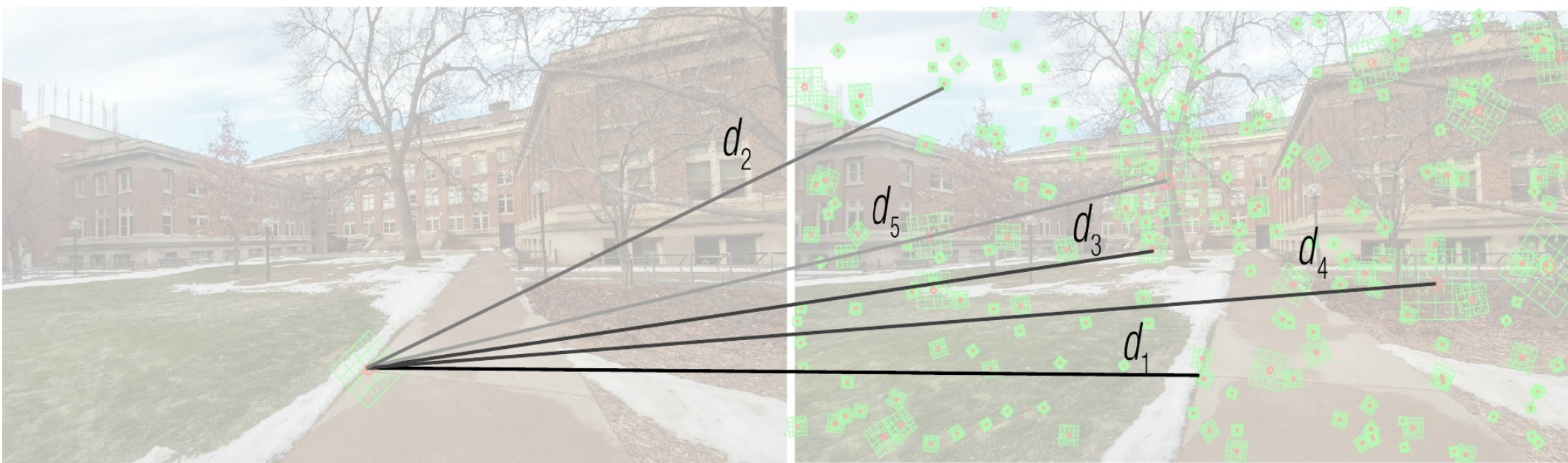


Mise en correspondance

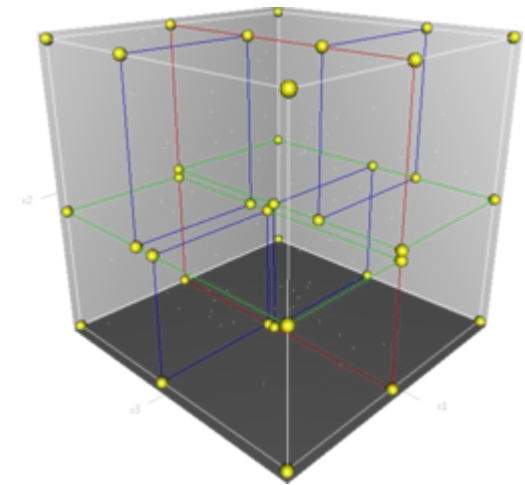
- Pour chaque POI dans une image il faut trouver le meilleur correspondant dans l'autre image



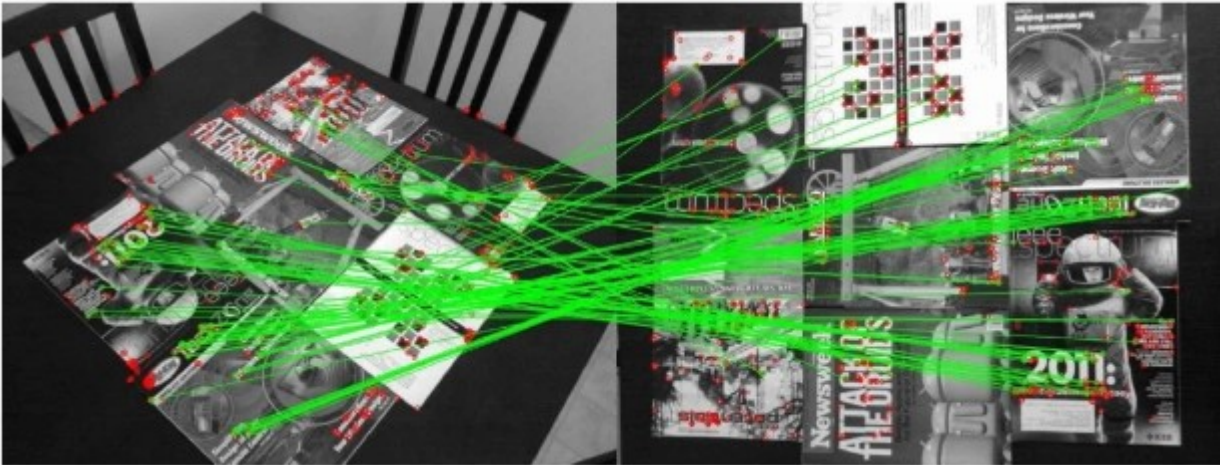
- Il faut une distance entre descripteurs. Pour ORB c'est la distance de Hamming (nombre de bits communs)



- Il faut aussi une méthode pour trouver le plus proche correspondant : Recherche du Plus Proche Voisin (**Nearest Neighbor Search**) :
 - Exhaustif : **Brute force** → coût élevé
 - Flann (**Fast Approximate Nearest Neighbor searches**) : kd-tree
 - Ransac : méthode stochastique
 - Hashing...
- Souvent associé à un filtrage des distances :
 - Exemple : $d1/d2 < 0.7$
 - *S'il y a trop de distances voisines il ne faut pas faire la mise en correspondances.*







- Algorithme efficace
- Grande variétés d'applications
- De nombreuses variantes (steered Brief, rBrief...)