Détection et mise en correspondance de points d'intérêt
Invariants locaux – mesures invariantes

Arnaud LE BRIS
arnaud.le-bris@ign.fr

28/02/2012

Introduction

Qu'est-ce qu'un point d'intérêt ?

La détection et la mise en correspondance de points d'intérêts est une étape nécessaire pour de nombreux processus de vision par ordinateur.
Ces points sont des points "remarquables" qui correspondent à des doubles discontinuités de la fonction d'intensité de l'image (celles-ci pouvant être dues à des variations de la radiométrie (texture) des objets photographiés ou à des discontinuités de profondeur…). Les points d'intérêt vont donc correspondre à des points isolés, des coins, des intersections, des "taches" ou des "blobs". Il y aura donc de bonnes chances de détecter ces mêmes points dans plusieurs images d'une même scène.

Intérêt des points d'intérêt

Les points d'intérêt présentent sur d'autres primitives tels que les contours plusieurs avantages pour la mise en correspondance :
- Ils présentent une double discontinuité de la fonction intensité de l'image contre une simple discontinuité pour les contours. (davantage de contrainte)
- Ils sont soit complètement occultés, soit visibles.
- Ils sont présents dans la majorité des images.
(- Ils ne nécessitent pas d'opérations de chaînage.)

Qualités d'un détecteur de points d'intérêt

- La **répétabilité** est la qualité d'un détecteur de points d'intérêt capable pour une scène donnée de détecter les points d'intérêt correspondant aux mêmes détails dans des images différentes, même si les conditions de prise de vue varient (éclairement, point de vue, échelle, rotation, …). Un détecteur de points d'intérêt est d'autant plus répétable qu'il produit les mêmes ensembles de points pour une même scène, malgré les variations des conditions de prise de vue.
  Définition : Soit P un point 3D de l'espace objet. Si l'image de P est détectée comme point d'intérêt dans l'image 1, alors P est "**répété**" dans l'image *i* si et seulement si l'image de P est également détectée dans l'image *i*.

- Un détecteur de point d'intérêt doit également faire preuve de **précision**. En effet, les points qu'il détecte ne doivent pas seulement correspondre à un même détail, mais aussi être précis géométriquement (i.e. au niveau de leurs coordonnées).

Les deux principales qualités d'un détecteur de points d'intérêt vont donc être sa **répétabilité** (aptitude à détecter des points correspondant aux mêmes détails) et sa **précision** (au niveau des coordonnées du point détecté).
Ces deux critères impliquent donc au détecteur de points d'intérêt d'être :
- **invariant** (à la translation,) à la rotation (2D), au changement d'échelle et aux autres déformations dues aux changements de point de vue (ex : affinités pas trop fortes…)
- **robuste** au bruit, aux conditions d'acquisition de l'image, à la compression, …
- **discriminant**, c'est-à-dire permettant d'obtenir quelques points correspondant à des détails spécifiques (en vue de l'appariement)

En fonction de l'application visée, le détecteur de points d'intérêt devra également respecter les deux critères suivants :

- **Quantité** : aptitude du détecteur à extraire beaucoup de points caractéristiques.
- **Efficacité / rapidité** : le calcul doit être rapide (critère important pour les applications temps réel)

Approches d'extraction de points d'intérêt

Il va donc s'agir d'abord d'extraire ces points particuliers des images, puis de les mettre en correspondance i.e. de retrouver leurs homologues dans d'autres images de la même scène.
La détection de ces points ainsi que les mesures utilisées pour leur mise en correspondance doivent donc être invariantes au bruit, aux variations d'illumination et aux changements de point de vue, i.e. aux rotations, aux variations d'échelle ainsi qu'aux déformations locales de l'image.

De nombreuses méthodes pour détecter des points d'intérêt ont été proposées. Elles peuvent se classer en 3 familles :
1. *Approches contours* : il s'agit de détecter les contours dans une image puis d'extraire les points d'intérêt le long des contours en considérant les points de courbures maximales ainsi que les intersections de contours.
2. *Approches intensité* : l'idée est de s'intéresser directement à la fonction d'intensité des images pour en extraire les points de discontinuité.
3. *Approches à base de modèles* : les points d'intérêts sont identifiés dans l'image par mise en correspondance de la fonction d'intensité de l'image avec un modèle théorique de cette fonction au voisinage des point d'intérêts recherchés. Précis mais problème de répétabilité (approche moins générique)…

Les approches « intensité » sont celles utilisées généralement. Les raisons en sont leur indépendance vis à vis d'une première étape de détection de contours (stabilité) ainsi que vis à vis du type de points d'intérêts (méthodes plus générales).

Les détecteurs de coins : de Moravec à Harris

Le détecteur de Moravec (1980)

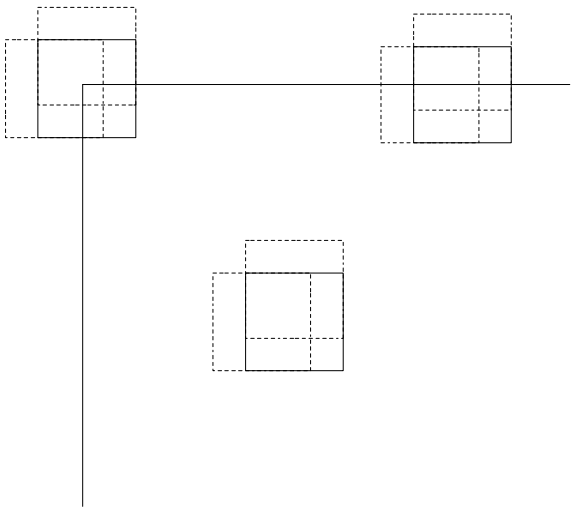
L'idée du détecteur de Moravec est de considérer le voisinage d'un pixel (i.e. une fenêtre) et de déterminer les changements moyens de l'intensité de l'image dans ce voisinage pour des petits déplacements de la fenêtre dans différentes directions.
Notons $I(x,y)$ l'intensité de l'image en (x ;y). On va en fait s'intéresser à la fonction $E_{dx,dy}(w)$ liée au changement moyen d'intensité produit par un déplacement local (dx, dy) de la fenêtre dans le voisinage rectangulaire défini par w (fonction d'appartenance au voisinage dont les valeurs sont soit 0, soit 1).

$$E_{dx,dy}(w) = \sum_{x,y} w(x,y).\left|I(x+dx, y+dy) - I(x,y)\right|^2$$

Les trois cas suivants vont alors se présenter :

- Si l'intensité de l'image est à peu près constante dans la zone de l'image considérée, alors $E_{dx,dy}(w)$ sera faible pour des déplacements (dx,dy) de la fenêtre dans toutes les directions.
- Si un contour rectiligne passe par la zone de l'image considérée, alors $E_{dx,dy}(w)$ sera faible pour des déplacements (dx,dy) de la fenêtre le long de cette ligne et fort pour des déplacements perpendiculaires à cette ligne.
- Si la zone de l'image considérée contient un coin ou un point isolé, alors $E_{dx,dy}(w)$ sera fort pour des déplacements (dx,dy) de la fenêtre dans toutes les directions.

Le détecteur de coins de Moravec consiste donc simplement à rechercher dans l'image les maxima locaux de $\min_{dx,dy\in\{-1;0;1\}}\left(E_{dx,dy}(u,v)\right)$supérieurs à un certain seuil. (avec $E_{dx,dy}(u,v) = E_{dx,dy}(w)$ pour w fenêtre voisinage du point (u;v))



Le détecteur de Harris (- Stephens) (1988)

Le détecteur de Moravec est limité par un certain nombre d'éléments : sa réponse est anisotropique, bruitée et trop forte aux contours.

Sa réponse est anisotropique puisqu'on ne s'intéresse en fait qu'à un ensemble discret de déplacements : un déplacement tous les 45°.
Appliquons à l'intensité de l'image un développement de Taylor au voisinage du pixel (x ;y) :

$$I(x+dx, y+dy) = I(x,y) + \frac{\partial I}{\partial x}(x,y).dx + \frac{\partial I}{\partial y}(x,y).dy + O(dx^2, dy^2)$$

Or $E_{dx,dy}(w) = \sum_{x,y} w(x,y).\left(I(x+dx, y+dy) - I(x,y)\right)^2$

On a donc, pour de petits déplacement (*dx,dy*) :

$$E_{dx,dy}(w) = \sum_{x,y} w(x,y).\left(\frac{\partial I}{\partial x}(x,y).dx + \frac{\partial I}{\partial y}(x,y).dy + O(dx^2, dy^2)\right)^2$$

soit :

$$E_{dx,dy}(w) = \sum_{x,y} w(x,y).\left[\left(\frac{\partial I}{\partial x}(x,y)\right)^2.dx^2 + \left(\frac{\partial I}{\partial y}(x,y)\right)^2.dy^2 + 2.\left(\frac{\partial I}{\partial x}(x,y).\frac{\partial I}{\partial y}(x,y)\right)dx.dy + O(dx^2, dy^2)\right]$$

où les dérivées premières sont approximées par :

$\frac{\partial I}{\partial x} = I_x = I * \begin{bmatrix}-1 & 0 & 1\end{bmatrix}$ et $\frac{\partial I}{\partial y} = I_y = I *^t \begin{bmatrix}-1 & 0 & 1\end{bmatrix}$

On obtient donc :

$$E_{dx,dy}(w) = \sum_{x,y} w(x,y).\left(I_x{}^2(x,y).dx^2 + I_y{}^2(x,y).dy^2 + 2.\left((I_x.I_y)(x,y)\right).dx.dy\right) + O(dx^2, dy^2)$$

$$E_{dx,dy}(w) = \left(\sum_{x,y} w(x,y).I_x{}^2(x,y)\right).dx^2 + \left(\sum_{x,y} w(x,y).I_y{}^2(x,y)\right).dy^2 + \left(2.\sum_{x,y} w(x,y).(I_x.I_y)(x,y)\right).dx.dy + O(dx^2,$$

On néglige désormais le terme en $O(dx^2, dy^2)$.

La fonction *w*(*u,v*) définit une zone de l'image et même plus précisément un voisinage autour du pixel (*u,*v). Cet élément peut donc être considéré comme un masque de convolution.
Par ailleurs, dans le cas du détecteur de Moravec, la réponse est bruitée en raison du voisinage considéré : le filtre utilisé est en effet binaire (1 ou 0) et appliqué sur un voisinage rectangulaire. Pour améliorer cela, Harris et Stephen proposent d'utiliser un filtre permettant de lisser sur une fenêtre circulaire comme par exemple le filtre gaussien.

$$w(x,y) = \frac{1}{2.\pi.\sigma^2}.e^{-\frac{(x-u)^2+(y-v)^2}{2.\sigma^2}}$$

On obtient alors :

$$E_{dx,dy}(u,v) = \left(\left(w * I_x^2\right)(u,v)\right).dx^2 + \left(\left(w * I_y^2\right)(u,v)\right).dy^2 + 2.\left(\left(w * (I_x.I_y)\right)(u,v)\right).dx.dy$$

soit : $E_{dx,dy}(u,v) = \begin{bmatrix}dx & dy\end{bmatrix} M(u,v) . \begin{bmatrix}dx \\ dy\end{bmatrix}$ avec $\boxed{M = \begin{bmatrix} w * I_x^2 & w * (I_x.I_y) \\ w * (I_x.I_y) & w * I_y^2 \end{bmatrix}}$

Notons $F_{u,v}(dx,dy) = E_{dx,dy}(u,v)$

Le détecteur de Moravec répond de manière trop forte aux contours en raison du fait que seul le minimum de $F_{u,v}$ est pris en compte en chaque pixel. Il faut donc prendre en compte le comportement général de la fonction localement.
Or la matrice *M* caractérise le comportement local de la fonction $F_{u,v}$ au voisinage de (*u*,*v*). Cette fonction est liée à la fonction locale d'autocorrélation de l'image (au voisinage de (*u*,*v*)). *M* décrit par conséquent également la forme et le comportement de cette fonction au voisinage de (*u*,*v*). Les valeurs propres de cette matrice sont proportionnelles aux courbures principales de cette fonction locale d'autocorrélation et forment donc une description invariante à la rotation de *M*

Trois cas de figure se présentent :

- Si les deux courbures sont de faibles valeurs, alors la région considérée a une intensité approximativement constante.
- Si une des courbures est de forte valeur alors que l'autre est de faible valeur, alors la région contient un contour.
- Si les deux courbures sont de fortes valeurs alors l'intensité de l'image varie fortement dans toutes les directions : on a un point d'intérêt.

Harris et Stephen ont donc introduit la mesure suivante (qui permet de comparer les valeurs propres de *M* sans les calculer explicitement):

$$R = \det(M) - k.(trace(M))^2$$

On prendra par exemple *k=0,05*.

Les valeurs de R sont positives au voisinage d'un point d'intérêt, négatives au voisinage d'un contour et faibles dans une région d'intensité constante.

(*u*,*v*) sera considéré comme un point d'intérêt si *R*(*u*,*v*)>*seuil* et si *R*(*u*,*v*) est un maximum local (par rapport aux 8 pixelsvoisins de (*u*,*v*)).

Référence :
Harris, C. et Stephens, M. A combined corner and edge detector. In *Alvey Vision Conference*, pp 147-151, 1988

	Image initiale I
	Mesure de Harris R
	Points détectés

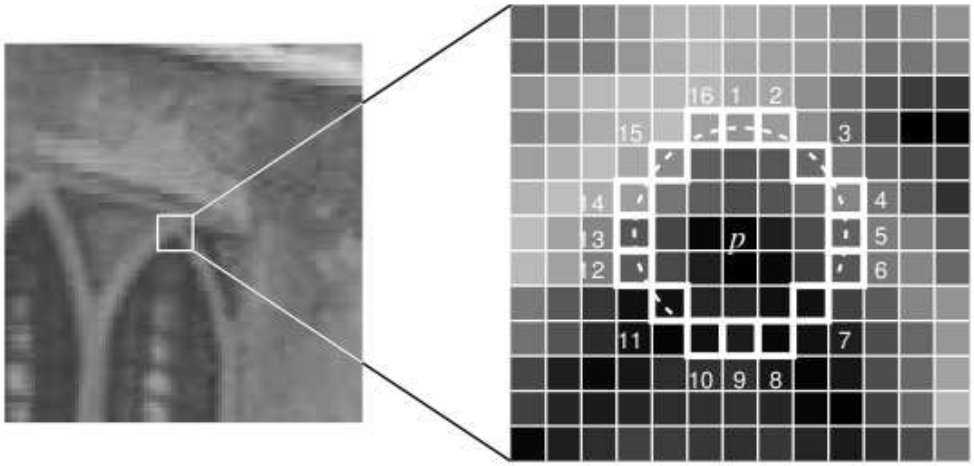
Le détecteur de Förstner (1987)

Ce détecteur de point reposant sur plusieurs critères et garantissant une bonne précision des points détectés a été très utilisé dans les logiciels de photogrammétrie.

Référence :
Förstner, W. et Gülch, E. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. of the ISPRS Intercommission Workshop*. Interlaken. 1987.

Un détecteur de coins différent : FAST (2005)

FAST (*Features from Accelerated Segment Test*) peut également être considéré comme un détecteur de coin.
Il est assez populaire ces dernières années, notamment du fait de la grande rapidité dont il fait preuve.



L'idée originale de FAST consiste à considérer un pixel p comme un coin si sa valeur est plus sombre ou plus claire (à un seuil près) que n pixels contigus appartenant au cercle de rayon R de centre le pixel p.

L'algorithme original est le suivant :

- On s'intéresse à un cercle de 16 pixels autour du pixel $p(x_p,y_p)$.
- On teste alors si n=12 pixels contigus de ce cercle ont tous une valeur supérieure à $I(p)+t$ ou si n=12 pixels contigus de ce cercle ont tous une valeur inférieure à $I(p)-t$.
- Si l'une de ces conditions est vérifiée, p est détecté comme coin.

Or, il est possible d'aller plus vite en rejetant d'emblée certaines configurations pour lesquelles la condition du test ne pourra être vérifiée. L'algorithme devient donc le suivant :

- On s'intéresse à un cercle de 16 pixels autour du pixel $p(x_p,y_p)$.

- Si le pixel 1 et le pixel 9 du cercle ont tous deux des valeurs comprises entre I(*p*)-*t* et I(*p*)-*t*, la condition de FAST pour que *p* soit un coin ne pourra être vérifiée. Fin du test pour *p*.
- On s'intéresse alors également aux pixels 5 et 13 du cercle. Si les pixels 1, 5, 9 et 13 du cercle ne sont pas au moins 3 à avoir tous une valeur supérieure à I(*p*)+*t* ou inférieure à I(*p*)-*t*, alors la condition de FAST pour que *p* soit un coin ne pourra être vérifiée. Fin du test pour *p*.
- On teste alors si $n=12$ pixels contigus de ce cercle ont tous une valeur supérieure à I(*p*)+*t* ou si $n=12$ pixels contigus de ce cercle ont tous une valeur inférieure à I(*p*)-*t*.
- Si l'une de ces conditions est vérifiée, *p* est détecté comme coin.

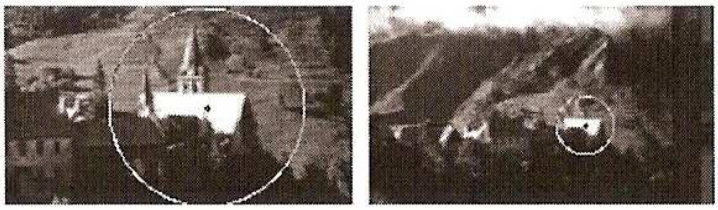
Cet algorithme souffrait néanmoins de quelques limitations. FAST a depuis fait l'objet de certaines améliorations (utilisant notamment des notions d'apprentissage (machine learning) pour définir ce qu'est un bon coin), le rendant plus générique et encore plus rapide..

Références :

Rosten, E. et Drummond, T. Fusing points and lines for high performance tracking.
In *International Conference on Computer Vision* (ICCV'05). Vol. 2. 2005.
Rosten, E. et Drummond, T. Machine learning for high-speed corner detection.
In *European Conference on Computer Vision* (ECCV'06). 2006.

Les détecteurs multi-échelle

Le détecteur de Harris permet de garantir (dans une certaine mesure) la répétabilité face aux rotations (2D) de l'image, mais pas aux variations d'échelle.
Pour être également invariant aux changements d'échelle, les détecteurs de points d'intérêt vont donc devoir prendre en compte la notion d'échelle, de niveau de détail.



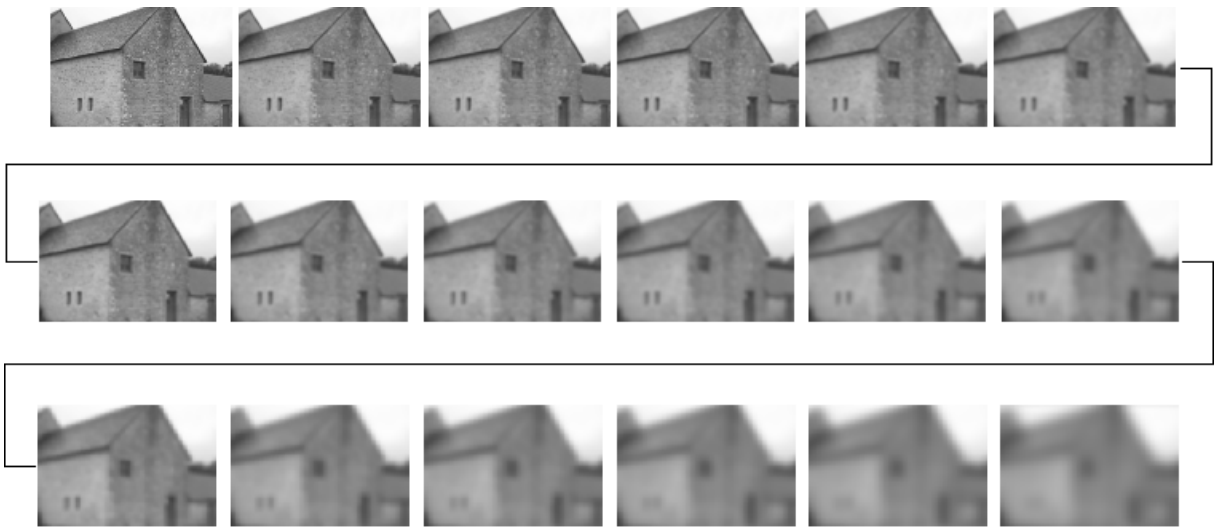
Les "coins" vont maintenant être recherchés dans un **espace d'échelles (*scale space*)** gaussien L construit à partir de l'image I :

Notation :

$$L(x, y, \sigma) = \big(L(\sigma)\big)(x, y) = (G_\sigma * I)(x, y)$$

$$L(\sigma) = G_\sigma * I \quad \text{avec } G_\sigma \text{ gaussienne d'écart type } \sigma$$

Le rôle d'un espace d'échelles est de s'intéresser à différents niveaux de détails, d'échelles (comme dans l'illustration suivante).



Les points qui vont être détectés ne seront donc plus de simples "coins"/points isolés, mais plus généralement des centres de taches, de "blobs" (comme dans les exemples ci-dessous).

Harris – Laplace (2004)

La mesure de "*cornerness*" (détection de "coins") devient :

$$R(x,y,\sigma_D,\sigma_I) = \det\big(M(x,y,\sigma_D,\sigma_I)\big) - k.\big(trace\big(M(x,y,\sigma_D,\sigma_I)\big)\big)^2$$

$$\text{avec } M(x,y,\sigma_D,\sigma_I) = \sigma_D^2.\begin{bmatrix} \big(G_{\sigma_I} * L(\sigma_D)_x^2\big)(x,y) & \big(G_{\sigma_I} * (L(\sigma_D)_x.L(\sigma_D)_y)\big)(x,y) \\ \big(G_{\sigma_I} * (L(\sigma_D)_x.L(\sigma_D)_y)\big)(x,y) & \big(G_{\sigma_I} * L(\sigma_D)_y^2\big)(x,y) \end{bmatrix}$$

σ_I *integration scale,* σ_D *differentiation scale*

avec : $f_x = \dfrac{\partial f}{\partial x}$, $f_{xy} = \dfrac{\partial^2 f}{\partial x \partial y}$, …

On va également utiliser une autre mesure pour déterminer l'échelle caractéristique des points détectés : il s'agit du ***LoG* ("*Laplacian-of-Gaussians*")** donné par la formule suivante :

$$\big|LoG(x,y,\sigma)\big| = \sigma^2.\big|L_{xx}(x,y,\sigma) + L_{yy}(x,y,\sigma)\big|$$

(Cette mesure atteint un extremum au niveau des structures en "blobs". (i.e. pour (x,y) centre d'une telle structure d'une taille correspondant à l'échelle σ …))

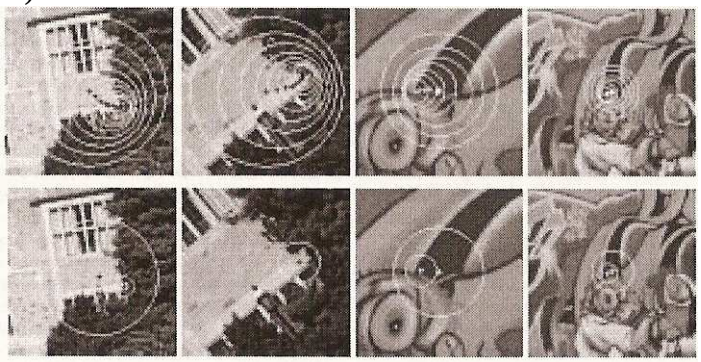
L'algorithme du détecteur de Harris-Laplace est le suivant :

1) Détection des points de Harris à chaque niveau σ_n de l'espace d'échelles (i.e. sur l'image $L(\sigma_n)$). On ne parcourt bien entendu pas toutes les échelles possibles, mais seulement les échelles $\sigma_n = \xi^n.\sigma_0$ (avec *s=1.4*). La mesure de Harris $R(x,y,\sigma_D,\sigma_I)$ définie précédemment est alors calculée pour $\sigma_I = \sigma_n$ et $\sigma_D = s.\sigma_I$ (avec *s=0.7*)
2) Pour chaque point $m^{(0)}(x^{(0)},y^{(0)},\sigma^{(0)})$ détecté à l'étape précédente, on itère les actions suivantes tant que $m^{(k)} \neq m^{(k+1)}$:
   a) Recherche de l'extremum local du *LoG* au point $(x^{(k)},y^{(k)})$ parmi les échelles au voisinage de $\sigma^{(k)}$ (Le domaine de recherche est limité au $\sigma(t) = t.\sigma^{(k)}$ avec $t \in [0.7,...,1.4]$.) Il s'agit de sélectionner l'échelle caractéristique du point d'intérêt. Si aucun extremum n'est rencontré, le point est rejeté.
   Sinon : $\sigma^{(k+1)} \leftarrow \sigma_m$ avec σ_m échelle de l'intervalle de recherche pour laquelle cet extremum est atteint.
   b) Recherche du maximum de la mesure de Harris le plus proche de $(x^{(k)}, y^{(k)})$ pour $\sigma_I = \sigma^{(k+1)}$ (et $\sigma_D = s.\sigma_I$). Soit (x_m, y_m) les coordonnées image de ce maximum. Elles deviennent les nouvelles coordonnées du point courant : $(x^{(k+1)}, y^{(k+1)}) \leftarrow (x_m, y_m)$

Une version simplifiée du détecteur de Harris-Laplace existe également avec une étape (2) consistant uniquement à éliminer les points détectés pour lesquels le *LoG* n'atteint pas d'extremum (à leur échelle) ou renvoie une valeur inférieure à un certain seuil.

Le détecteur de Harris-Laplace permet donc pour chaque point d'intérêt détecté de déterminer non seulement sa position 2D (*x*,*y*), mais aussi son échelle caractéristique σ.
Par ailleurs, comme on peut le constater sur la figure ci-dessous, le détecteur de Harris-Laplace permet une détection de points plus stable que celle fournie par un simple Harris

multi-échelle (c'est-à-dire la simple application du détecteur de Harris aux différents niveaux de l'espace d'échelles).



En haut : points détectés simplement par Harris multi-échelle.
En bas : points détectés par Harris-Laplace.

<u>*Référence*</u> :
Mikolajczyk, K. et Schmid, C. Scale and affine invariant point detectors. In *International Journal of Conference Vision (IJCV)*, n°60(1), pp 63-86, 2004

Hessian – Laplace (2004)

Il s'agit d'une variante de ce qui précède.

Cette fois, le critère de détection de "coins" (équivalent de "*R*" du détecteur de Harris-Laplace) de la méthode est le déterminant de la hessienne (dont on va rechercher les maxima locaux), c'est-à-dire :

$$\det(H) \text{ avec } H(x,y,\sigma) = \sigma^2 . \begin{bmatrix} (L(\sigma))_{xx}(x,y) & (L(\sigma))_{xy}(x,y) \\ (L(\sigma))_{xy}(x,y) & (L(\sigma))_{yy}(x,y) \end{bmatrix}$$

Les dérivées secondes de $(L(\sigma))$ renvoient en effet des réponses fortes sur les "blobs" et les crêtes.
Les points détectés à chaque échelle σ correspondent donc aux extrema locaux à la fois du déterminant et de la trace de la matrice $H(x,y,\sigma)$.
En choisissant des points qui maximisent le déterminant de la hessienne, cette mesure pénalise les structures plus allongées qui ont une dérivée seconde plus faible dans une direction.
(Notons au passage que la trace de $H(x,y,\sigma)$ est identique au *LoG*.)

L'algorithme de Hessian-Laplace est similaire à celui de Harris-Laplace, avec une première étape de détection de coins dans l'espace d'échelles construit à partir de l'image et une seconde étape (itérative) de sélection et de recherche de la position précise du point.

SIFT (et le *DoG*) (2004) et ses variantes, SURF (2006)

Ces méthodes comportent à la fois un détecteur de point d'intérêt et une méthode d'appariement, i.e. un descripteur invariant caractérisant le comportement de l'image au voisinage des points d'intérêt détectés.
Aussi, des parties spécifiques leur sont consacrées plus bas dans ce document.
(cf parties SIFT et SURF)

ORB (Oriented FAST and Rotated BRIEF)

ORB comprend un détecteur de point d'intérêt qui est une extension du détecteur FAST au cas multi-échelle :

- Les coins FAST sont détectés à chaque niveau d'un espace d'échelle
- On vérifie ensuite s'ils vérifient le critère de Harris (afin d'éviter d'avoir trop de points situés sur des contours rectilignes)

Remarque :
Tout comme SIFT et SURF (ainsi qu'on le verra plus bas), ORB est constitué à la fois d'un détecteur de point d'intérêt et d'une méthode d'appariement, i.e. d'un descripteur invariant caractérisant le comportement de l'image au voisinage des points d'intérêt détectés. Dans ce cas précis, le descripteur BRIEF est utilisé.

Référence :
Rublee , E., Rabaud, V., Konolige, K. et Bradski, G. ORB : an efficient alternative to SIFT and SURF. In *proc. of the International Conference on Computer Vision* (*ICCV'11*). 2011.

Les détecteurs invariants aux affinités

On a vu précédemment certains détecteurs réellement conçus de manière à être invariants aux rotations (2D dans le plan de l'image) et aux variations d'échelles. Ces détecteurs sont généralement également invariants dans une certaine mesure à des déformations affines de l'image.
Néanmoins, il existe également des détecteurs réellement invariants aux affinités et conçus dans ce but.

Harris–affine (2004)

La mesure de "*cornerness*" (détection de "coins") devient :

$$R(x, y, \Sigma_D, \Sigma_I) = \det(\Sigma_D).G_{\Sigma_I} * \left((\nabla L)(x, \Sigma_D).^t (\nabla L)(x, \Sigma_D)\right)$$

avec Σ_D et Σ_I matrices de covariance des noyaux gaussiens respectivement de différenciation et d'intégration.

Dans le cas isotropique (i.e. si $\Sigma_I = \begin{bmatrix} \sigma_I^2 & 0 \\ 0 & \sigma_I^2 \end{bmatrix}$ et $\Sigma_D = \begin{bmatrix} \sigma_D^2 & 0 \\ 0 & \sigma_D^2 \end{bmatrix}$), cette mesure est exactement la même que celle du détecteur de Harris-Laplace.

Référence :
Mikolajczyk, K. et Schmid, C. Scale and affine invariant point detectors. In *International Journal of Conference Vision (IJCV)*, n°60(1), pp 63-86, 2004

Hessian–affine

(même approche que pour Harris – affine)

MSER (Maximum Stable Extremal Region) (2002)

Cette fois, on va rechercher des régions.

Le terme "*extremal*" signifie que tous les pixels à l'intérieur d'une région MSER ont une valeur supérieure (ou inférieure) à celle des pixels de sa frontière. On recherche en effet les "régions extrémales" (i.e. les régions les plus claires et les plus sombres) de l'image.
Autrement dit, les régions *R* recherchées sont telles que :

$$\max_{p \in R} I(p) \leq \min_{p \in \partial R} I(p) \text{ ou } \min_{p \in R} I(p) \geq \max_{p \in \partial R} I(p)$$ (avec ∂R désignant la frontière de *R*)

Le terme "*maximum stable*" renvoie au fait qu'on recherche les régions les plus stables : une région MSER va être une région de l'image qui reste stable par la binarisation pour des seuils variant dans un certain intervalle. Autrement dit, la surface d'une région MSER doit peu varier si l'image est binarisée pour des seuils différents, et doit varier le moins possible pour des seuils variant dans un intervalle donné.

L'algorithme est le suivant (avec d'abord une détection des régions MSER puis une détermination de leurs caractéristiques) :

1/ Recherche des régions extrémales

- Tri des pixels de l'image en fonction de leur intensité et tri des pixels par régions : construction d'un arbre
- Calcul de l'aire des régions
- Détection des régions extrémales (suppression des régions incluses dans une région plus grande correspondant à un même seuil)

2/ Recherche des régions les plus stables

- Calcul de la variation d'aire relative pour une variation du seuil de binarisation
- Sélection des régions les plus stables (i.e. celles pour lesquelles la variation d'aire est plus petite que celle de leur parent et de leurs enfants)

3/ Filtrage : élimination de certaines régions

- Elimination des régions trop petites ou trop grandes
- Elimination des régions dont l'aire varie trop (selon les binarisations)
- Elimination des régions quasiment semblables à une autre région détectée dans laquelle elles sont incluses (i.e. si la variation d'aire relative entre la région et sa première région parente conservée est trop faible)

4/ Détermination des caractéristiques (centre, échelle, moment) des régions

- Modélisation de chaque région par une ellipse (i.e. lissage de la région) : calcul du centre et du moment de la région (c'est le centre de la région qui constituera le point d'intérêt)
- Estimation de l'échelle associée à la région

Référence :

Matas, J., Chum, O., Urba, M. et Pajdla, T. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC'2002*, pp 384-396, 2002

ASIFT (Affine SIFT) (2009)

(voir plus bas)

Site présentant plusieurs détecteurs de points d'intérêt :

http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html

Appariement – mesures invariantes

On souhaite maintenant apparier des points d'intérêt extraits d'une première image avec ceux extraits d'une seconde image. Cette tâche nécessite d'utiliser des "descripteurs" décrivant le comportement de l'image au voisinage de ces points ainsi que des mesures permettant de les comparer.
Ces descripteurs et mesures doivent donc être :

- **Invariants** (aux translations,) aux rotations (2D), aux changements d'échelle et autres déformations locales liées aux changements de points de vue (ex : affinités pas trop fortes…)
- **Robustes** au bruit, aux conditions d'acquisition de l'image, à la compression, …
- **Discriminants**, c'est-à-dire permettant de bien caractériser les points homologues. Ils doivent également permettre d'extraire une certaine **quantité** de points homologues.

L'utilisation de descripteurs **locaux** et de mesures calculées **localement** permettra de vérifier ces critères d'invariance et d'éviter d'être trop sensible aux occlusions.

En fonction de l'application visée (et notamment pour les applications en temps réel), l'**efficacité /** la **rapidité** pourra également être un critère important.

Coefficient de corrélation

L'appariement (i.e. la mise en correspondance) des points d'intérêt détectés peut se faire par corrélation.

$$\frac{\sum_{u,v}\left(I_1(x_1+u,x_1+v)-\bar{I}_1\right).\left(I_2(x_2+u,y_2+v)-\bar{I}_2\right)}{\sqrt{\sum_{u,v}\left(I_1(x_1+u,x_1+v)-\bar{I}_1\right)^2}.\sqrt{\sum_{u,v}\left(I_2(x_2+u,y_2+v)-\bar{I}_2\right)^2}}$$

Cette mesure présente toutefois des problèmes de robustesse face aux rotations et aux changements d'échelle. Un prédicteur peut donc s'avérer nécessaire.

Invariants du signal : invariants différentiels

D'autres mesures ont alors été proposées : notamment des mesures invariantes du signal, comme par exemple des invariants différentiels, calculés au voisinage d'un point d'intérêt et décrivant la géométrie locale (i.e. le comportement de l'image) au sein de ce voisinage.

En voici un exemple (invariance aux rotations) :

$$V=\begin{bmatrix} L \\ L_x.L_x+L_y.L_y \\ L_{xx}.L_x.L_x+2L_{xy}.L_x.L_y+L_{yy}.L_y.L_y \\ L_{xx}+L_{yy} \\ L_{xx}.L_{xx}+2.L_{xy}.L_{xy}+L_{yy}.L_{yy} \end{bmatrix} \quad \text{avec} \quad L_x=\frac{\partial L}{\partial x} \quad L_{xy}=\frac{\partial^2 L}{\partial x \partial y} \quad L=G_\sigma * I$$

L désigne la convolution de l'image I par une gaussienne
(L'écart type de la gaussienne permet de caractériser une fonction à plusieurs niveaux d'échelle ou d'adapter l'échelle à l'image considérée.)

(D'autres mesures du même type peuvent être proposées.)

On cherche alors pour un point (de vecteur d'invariants différentiels *V1*) de l'image 1 le point (de vecteur d'invariants différentiels *V2*) de l'image 2 qui minimise la distance euclidienne *d(V1,V2)*.

Référence :
Schmid, C. et Mohr, R. Mise en correspondance par invariants locaux in *Traitement du Signal* – volume 13 – n°6 , pp 591- 605, 1996

Invariants basés sur des moments colorimétriques

Soit une image RVB. On a donc cette fois $I(x,y)=\{R(x,y)\ V(x,y)\ B(x,y)\}$.
On définit alors le moment colorimétrique d'ordre $p+q$ et de degré $a+b+c$ au sein de la région de l'image Ω :

$$M_{pq}^{rvb}(\Omega) = \iint_{\Omega} x^p . y^q . [R(x,y)]^r . [V(x,y)]^v . [B(x,y)]^b . dx.dy$$

Si l'on suppose que la déformation de la texture entre les image I et I' peut être modélisée localement par une affinité :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_x & a_y \\ b_x & b_y \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_o \\ b_o \end{pmatrix}$$

et les variations radiométriques peuvent être modélisées par :

$$R'(x,y) = s_R . R(x,y) + u_R \quad V'(x,y) = s_V . V(x,y) + u_V \quad B'(x,y) = s_B . B(x,y) + u_B$$

alors :

$$\left(M_{pq}^{rvb}(\Omega)\right)' = \iint_{\Omega} \frac{[a_x x + a_y y + a_o]^p . [b_x x + b_y y + b_o]^q}{[s_R R(x,y) + u_R]^r . [s_V V(x,y) + u_V]^v . [s_B B(x,y) + u_B]^b} . abs(|A|) . dx.dy$$

$$\begin{pmatrix} \left(M_{10}^{rvb}\right)' \\ \left(M_{01}^{rvb}\right)' \\ \left(M_{00}^{rvb}\right)' \end{pmatrix} = s_R^r . s_V^v . s_B^b . |A| . \begin{pmatrix} a_x & a_y & a_o \\ b_x & b_y & b_o \\ 0 & 0 & 1 \end{pmatrix} . \begin{pmatrix} M_{10}^{rvb} \\ M_{01}^{rvb} \\ M_{00}^{rvb} \end{pmatrix} \quad \forall (r,v,b) \in \begin{Bmatrix} (0,0,0);(1,0,0);(0,1,0);(0,0,1);(2,0,0); \\ (0,2,0);(0,0,2);(1,1,0);(1,0,1);(0,1,1) \end{Bmatrix}$$

Il devient alors possible d'en dériver des invariants. Ainsi, si l'on a une seule bande (M_{pq}^{i} ou dans les cas $M_{pq}^{r00}, M_{pq}^{0v0}, M_{pq}^{00b}$) :

$$B_{02} = \frac{M_{00}^2 M_{00}^0}{\left(M_{00}^1\right)^2} \text{ (ordre 0, degré 2)}$$

$$B_{12} = \frac{M_{10}^2 M_{01}^0 M_{00}^1 + M_{10}^1 M_{01}^2 M_{00}^0 + M_{10}^0 M_{01}^1 M_{00}^2}{M_{00}^2 M_{00}^1 M_{00}^0} - \frac{M_{10}^2 M_{01}^1 M_{00}^0 + M_{10}^1 M_{01}^0 M_{00}^2 + M_{10}^0 M_{01}^2 M_{00}^1}{M_{00}^2 M_{00}^1 M_{00}^0}$$

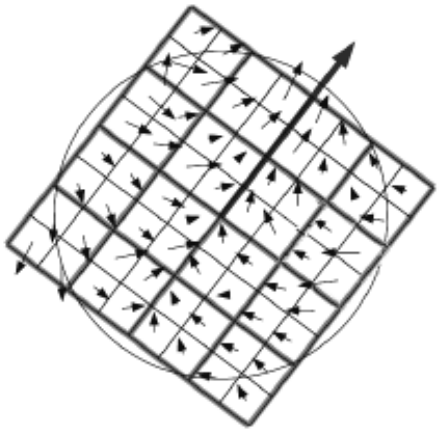
(ordre 0, degré 2)

……

<u>Référence</u> :
Mindru, F., Moons, T. et Van Gool, L. Recognizing Color Patterns Irrespective of Viewpoint and Illumination. In *CVPR'99*, pp 368-373, 1999

Le descripteur SIFT et ses variantes

(voir parties spécifiques)
Il s'agit là encore de descripteurs (vecteurs de "grandes" dimension) décrivant le comportement local de l'image au voisinage du point d'intérêt (à son échelle).
Les variantes plus ou moins inspirées du principe de **SIFT** les plus connues sont notamment **SURF**, **GLOH**, **PCA-SIFT**, **DAISY** et **BRIEF**.



L'appariement des points d'intérêt s'effectue alors dans l'espace des descripteurs (par la recherche du plus proche voisin au sens d'une distance entre les descripteurs).

SIFT (Scale Invariant Feature Transform)

Proposée par D. Lowe en 1999 puis en 2004, la méthode *SIFT* (*Scale Invariant Feature Transform*) est à la fois une méthode d'extraction (aussi appelée ***Difference-of-Gaussian*** ou ***DoG***) de points d'intérêt (au voisinage desquels on va décrire le comportement de l'image au moyen d'un descripteur) et de mise en correspondance de ces points (dans l'espace de ces descripteurs).

Le détecteur de points est aussi connu sous le nom de *DoG*. Il s'agit d'un détecteur de points d'intérêt, ou plus exactement de "points clés" ou "key points" puisque les points détectés ne correspondent pas nécessairement à des détails ponctuels de l'image, mais aussi à des régions.

En pratique, *SIFT* fait preuve d'une grande robustesse notamment face aux variations d'échelle, aux rotations, au bruit et aux variations d'éclairement ainsi qu'à des transformations affines limitées mais est en revanche assez sensible au phénomène de diachronisme (i.e. lorsque les images n'ont pas été acquises à la même période sous les mêmes conditions d'illumination et présentent un aspect différent).
Les grandes qualités et la robustesse de la partie descripteur et appariement de *SIFT* ont été montrées par (Mikolajczyk et al., 2005). Le détecteur n'est initialement conçu que pour être invariant aux variations d'échelle ((Morel et Yu, 2008) montre d'ailleurs qu'on ne peut pas vraiment faire mieux à ce niveau), aux rotations 2D et aux variations d'illumination (il faut toutefois faire attention aux paramètres d'élimination des points les moins contrastés), mais se comporte en pratique très bien dans le cas de déformations affines restant limitées. Mentionnons toutefois l'existence de la méthode *ASIFT* conçue pour être invariante aux affinités et présentée dans plus loin.

Du fait de son caractère multi-échelle, les points homologues fournis par *SIFT* ne correspondent pas tous à des détails ponctuels de l'image, mais aussi à des régions. Les conséquences de ce fait sur leur précision et la manière de l'améliorer est notamment étudiée dans (Remondino, 2006) dans le cadre d'applications photogrammétriques.

SIFT fonctionne également en deux étapes distinctes :

A/ Extraction des points et calcul de leurs descripteurs

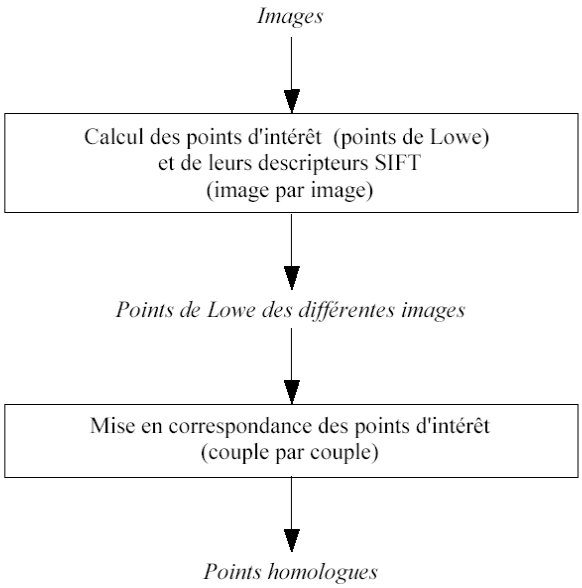
1- Les points d'intérêts (points de Lowe) sont d'abord extraits de chaque image à l'aide du détecteur *DoG*

2- Un descripteur *SIFT* leur est ensuite associé : il va décrire le comportement de l'image au voisinage de ces points. Il s'agit d'un vecteur de dimension 128.

B/ Appariement

Pour chaque paire d'images, l'étape de mise en correspondance revient alors à rechercher pour chaque point d'intérêt de la première image son plus proche voisin dans l'espace des descripteurs parmi les points détectés dans la seconde image.

Remarque : Cette approche peut être mise en œuvre pour un détecteur de points d'intérêt différent du *DoG*, comme par exemple Harris-Laplace ou Harris-Affine. Un descripteur SIFT peut en effet être associé à des points d'intérêt extraits de l'image par n'importe quel détecteur.



Extraction des points d'intérêt (le détecteur DoG) et calcul des descripteurs

Dans un premier temps, les points de Lowe (points d'intérêt) sont extraits de chaque image. Un descripteur leur est ensuite associé : il décrit le comportement de l'image dans leur voisinage et va être utilisé lors de l'étape suivante de mise en correspondance des points de Lowe homologues.

Ce calcul s'effectue en 2 étapes principales :

A. Extraction des points d'intérêt
   1. Extraction des points d'intérêt potentiels (invariant aux changements d'échelle et à l'orientation) : ces points sont sélectionnés parmi les extrema dans un espace d'échelles (*scale*-space) d'une fonction différence de gaussienne convoluée avec l'image initiale.
   2. Affinage de la localisation des points détectés à l'étape précédente par interpolation. Rejet ou conservation des points détectés en fonction de mesures de leur stabilité (les points les plus contrastés et non situés sur une arête sont conservés.)

B. Calcul des orientations et des descripteurs
   3. Calcul de l'orientation (ou des orientations) associée(s) à chaque point d'intérêt.
   4. Calcul du descripteur *SIFT* associé à chaque point d'intérêt détecté

Algorithme d'extraction des points de Lowe

A – Extraction des points de Lowe

0 - Sur-échantillonnage [facultatif] de l'image d'un facteur 2 (interpolation linéaire) puis convolution par une gaussienne.

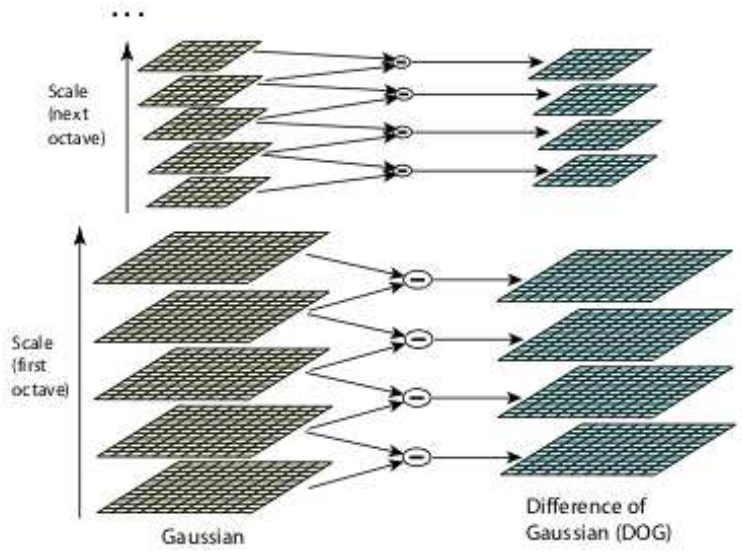
1 - Construction de l'espace d'échelles (« *scale-space* »)

Le rôle d'un espace d'échelles est de s'intéresser à différents niveaux de détails, d'échelles (comme dans l'illustration suivante).

Notations :

On désigne par *I* l'image à traiter et par G_σ la gaussienne de d'écart type σ.

Chaque octave de l'espace d'échelles est divisée en *Ni* intervalles. Soit $k = 2^{1/Ni}$.



$\sigma = \sigma_0$

$I_o \leftarrow I$

Pour *o* de *0* à *No* **Faire**

$\sigma_t = \sigma_0$

Pour *i* de *1* à *Ni+3* **Faire**

$L(x, y, \sigma) = (L(\sigma))(x, y) = (G_{\sigma_t} * I_o)(x, y)$

Différences de gaussiennes convoluées avec l'image :

$$D(x, y, \sigma) = (L(k.\sigma_t))(x, y) - (L(\sigma_t))(x, y) = ((G_{k.\sigma_t} - G_{\sigma_t}) * I)(x, y)$$

$\sigma = k.\sigma$

$\sigma_t = k.\sigma_t$

FinPour

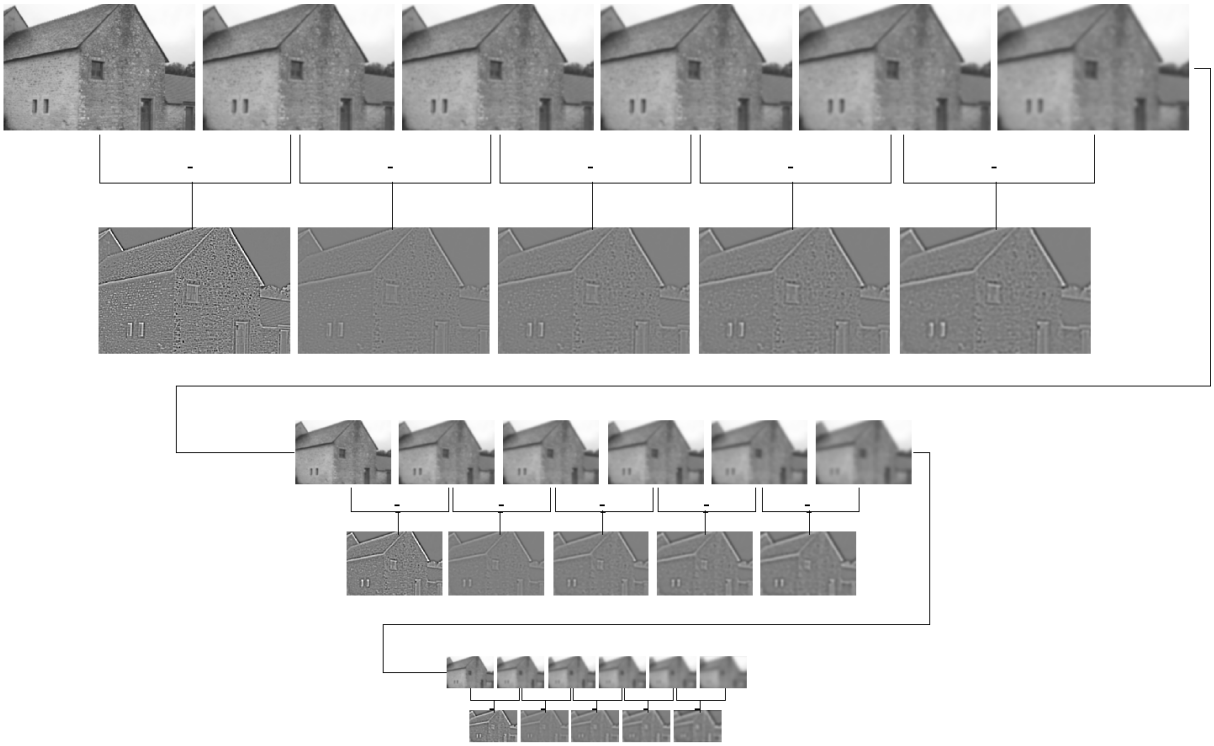
L'image $(L(2^{o+1}.\sigma_0))$ est sous-échantillonnée d'un facteur 2.

→ L'image résultante devient la nouvelle image I_o.

FinPour

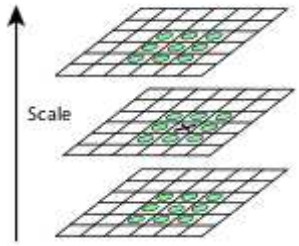
La différence de gaussienne (*Difference of Gaussian* (*DoG*)) revient en fait à une approximation du *Laplacian of Gaussian* (*LoG*).

Remarque : Le fait de sous-échantillonner l'image à la fin de chaque octave est une approximation permettant de réduire les temps de calcul.



2 - Recherche des extrema (minima et maxima) locaux de $D(x, y, \sigma)$

(Un tel point est non seulement inférieur ou supérieur à ses 8 voisins à la même échelle mais également à ses 9 voisins des échelles supérieure et inférieure.)



3 - Affinage de la localisation des points détectés à l'étape précédente par interpolation à partir de la fonction D : recherche de la position interpolée de l'extremum.

Soit $P(x; y; \sigma)$ un point détecté précédemment. Soit la matrice $X = {}^t[x \quad y \quad \sigma]$.

On recherche l'extremum $\hat{X} = X + d\hat{X}$ de D au voisinage de X :

$$\begin{bmatrix} d\hat{x} \\ d\hat{y} \\ d\hat{\sigma} \end{bmatrix} = d\hat{X} = -\left(\frac{\partial^2 D}{\partial X^2}(X)\right)^{-1} . \frac{\partial D}{\partial X}(X)$$

(qui correspond au passage par 0 de la dérivée de

$D(X + dX) = D(X) + {}^t\left(\frac{\partial D}{\partial X}(X)\right).dX + \frac{1}{2}.{}^t dX.\left(\frac{\partial^2 D}{\partial X^2}(X)\right).dX$ (développement de Taylor de la fonction D au voisinage de X).)

Si $\max(d\hat{x}, d\hat{y}, d\hat{\sigma}) > 0{,}5$ **Alors**

On se trouve en fait face à un nouveau point de coordonnées $X + d\hat{X}$ pour lequel on effectue une nouvelle interpolation.

Sinon

Le point *P* reçoit les nouvelles coordonnées $X + d\hat{X}$.

FinSi

4 - Filtrage des points détectés : élimination des points les moins contrastés

Les points les plus contrastés sont privilégiés car ce sont ceux que l'on aura vraisemblablement le plus de chance de détecter sur les différentes images. Les points les moins contrastés sont donc éliminés par la méthode suivante :

Si $\left| D(X) + \frac{1}{2}.{}^{t}\left(\frac{\partial D}{\partial X}(X)\right).d\hat{X} \right| < S_{constraste}$ (avec les mêmes notations qu'à l'étape précédente)

i.e. $\left| D(X + d\hat{X}) \right| < S_{contraste}$ **Alors**

Le point *P* est éliminé.

Sinon

Le point *P* est conservé.

FinSi

5 - Filtrage des points détectés : élimination des points situés sur une arête

On souhaite récupérer de véritables points (pour lesquels l'image présente une double discontinuité) et non des points situés sur une arête (et pour lesquels l'image ne présente une discontinuité que dans une seule direction). Les points situés sur une arête ne fixe donc une contrainte que dans une seule direction et sont donc éliminés par la méthode suivante :

Soit la matrice hessienne :

$$H(x,y,\sigma) = \begin{bmatrix} \frac{\partial^2 D}{\partial x^2}(x,y,\sigma) & \frac{\partial^2 D}{\partial x \partial y}(x,y,\sigma) \\ \frac{\partial^2 D}{\partial y \partial x}(x,y,\sigma) & \frac{\partial^2 D}{\partial y^2}(x,y,\sigma) \end{bmatrix}$$ au point $P(x;y;\sigma)$.

Si $\frac{Tr\left(H(x,y,\sigma)\right)^2}{Det\left(H(x,y,\sigma)\right)} < \frac{(S_{edge}+1)^2}{S_{edge}}$ (avec *Tr(M)* et *Det(M)* respectivement trace et déterminant d'une matrice *M)* **Alors**

Le point *P* est conservé.

Sinon

Le point *P* est rejeté.

FinSi

B – Calcul de leur orientation et de leur descripteur SIFT (Scale Invariant Feature Transform)

1 - Pour chaque point d'intérêt, calcul de son orientation

- Calcul de l'histogramme des orientations du gradient des pixels voisins du point d'intérêt (x,y,σ) (dans l'image $(L(\sigma))$ correspondant à l'échelle σ du point d'intérêt). L'influence de l'orientation θ du gradient d'un de ces pixels voisins (u,v) lors du calcul de cet histogramme est pondérée par un poids fonction du module *m* du gradient en

(u,v) et de la distance de (u,v) au point d'intérêt (x,y) ($poids(u,v) = \frac{1}{2.\pi.(1{,}5.\sigma)^2}.e^{-\frac{(x-u)^2+(y-v)^2}{2.(1{,}5.\sigma)^2}}$).

$$m(u,v) = \sqrt{\left(\left(L(\sigma)\right)(u+1,v) - \left(L(\sigma)\right)(u-1,v)\right)^2 + \left(\left(L(\sigma)\right)(u,v+1) - \left(L(\sigma)\right)(u,v-1)\right)^2}$$

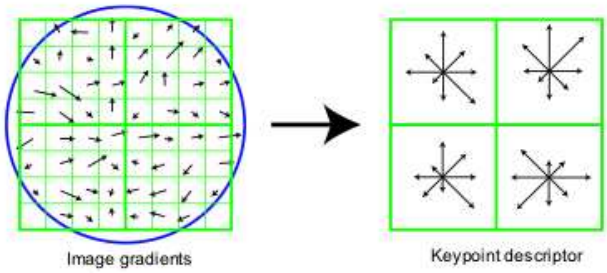
$$\theta(u,v) = \tan^{-1}\left(\frac{\left(L(\sigma)\right)(u,v+1) - \left(L(\sigma)\right)(u,v-1)}{\left(L(\sigma)\right)(u+1,v) - \left(L(\sigma)\right)(u-1,v)}\right)$$

- L'orientation du point d'intérêt est celle qui correspond au plus haut pic de l'histogramme. Si celui-ci comporte d'autres pics importants (>80 % du plus haut pic), on dédouble le point d'intérêt.

Tous les calculs qui suivent sont effectués relativement à l'orientation, l'échelle et les coordonnées 2D du point d'intérêt afin de garantir l'invariance par rapport à ces paramètres.

2 - Pour chaque point d'intérêt, calcul de son descripteur SIFT

- Calcul de l'orientation (relative à l'orientation du point d'intérêt calculée à l'étape précédente) du gradient des pixels voisins du point d'intérêt (dans l'image $\left(L(\sigma)\right)$ correspondant à l'échelle σ du point d'intérêt).
- Calcul des histogrammes de ces orientations au sein de fenêtres (plus nombreuses que les 4 fenêtres de l'illustration suivante) situées de part et d'autres du point d'intérêt détecté. (L'influence de ces orientations lors du calcul de l'histogramme est pondérée comme lors de l'étape de détermination de l'orientation du point d'intérêt.)



- Le descripteur SIFT du point d'intérêt est le vecteur (de dimension 128) contenant la concaténation des valeurs de ces histogrammes. (Ce vecteur est ensuite normalisé afin de garantir l'invariance aux variations d'éclairement.)

De par sa construction, ce descripteur constitue donc une mesure invariante à l'échelle, aux rotations, aux variations d'éclairement et, de manière empirique, il est également invariant à certaines déformations locales (affinités "légères").

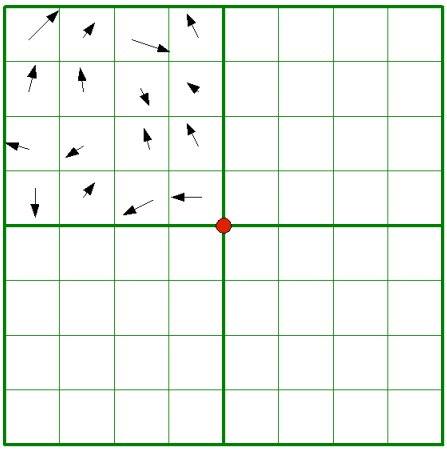
<u>Plus de détails sur le calcul de l'orientation et du descripteur…</u>

Voyons plus en détail et étape par étape la détermination de l'orientation et du descripteur du point détecté (et plus particulièrement le calcul des histogrammes utilisés).
Les calculs de l'orientation et du descripteur se font à l'échelle σ du point d'intérêt (x,y,σ), autrement dit sur l'image $L(\sigma)$ de l'espace d'échelles calculé à partir de l'image I.

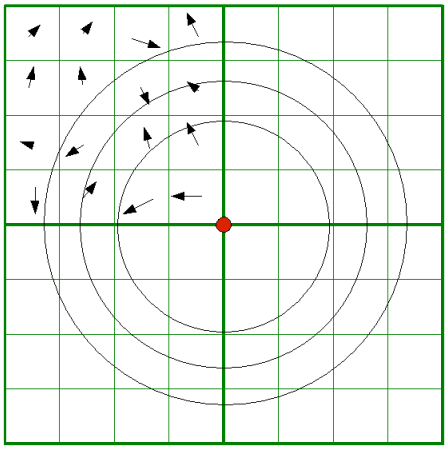
On note respectivement $m(u,v)$ et $\theta(u,v)$ le module et l'argument du gradient de $L(\sigma)$ en (u,v).

$$m(u,v) = \sqrt{\left(\left(L(\sigma)\right)(u+1,v) - \left(L(\sigma)\right)(u-1,v)\right)^2 + \left(\left(L(\sigma)\right)(u,v+1) - \left(L(\sigma)\right)(u,v-1)\right)^2}$$

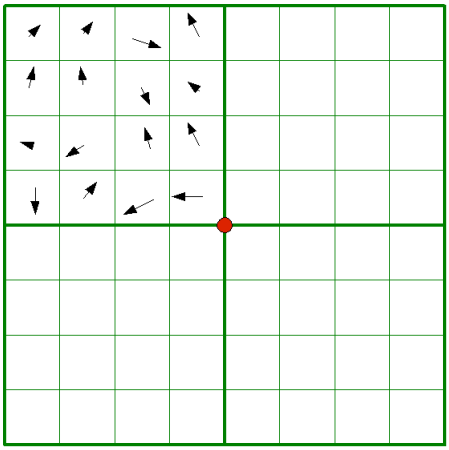
$$\theta(u,v) = \tan^{-1}\left(\frac{\left(L(\sigma)\right)(u,v+1) - \left(L(\sigma)\right)(u,v-1)}{\left(L(\sigma)\right)(u+1,v) - \left(L(\sigma)\right)(u-1,v)}\right)$$



On va s'intéresser aux pixels voisins du point d'intérêt détecté précédemment (figuré ici par un point rouge).
On calcule le gradient de l'image en chacun des pixels situés dans un certain voisinage du point d'intérêt. Sur l'illustration ci-contre, on a représenté par des flèches noires ces vecteurs gradients dans la partie supérieure gauche de ce voisinage.

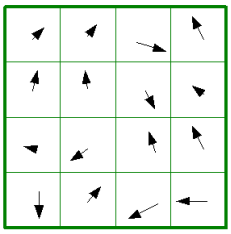


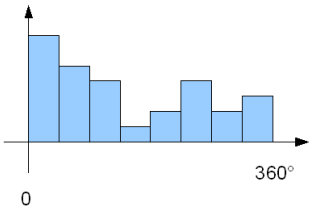
On convolue les modules de ces gradients par une gaussienne centrée sur le point d'intérêt. (Cela va permettre de donner plus d'influence aux pixels les plus proches du point d'intérêt lors du calcul de l'histogramme de répartition des orientations des gradients.)

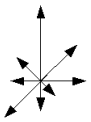


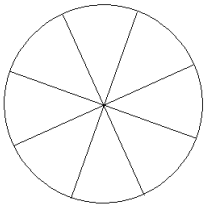
On obtient $m_{pond}(u,v) = poids(u,v).m(u,v)$

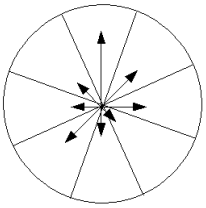
avec : $poids(u,v) = \frac{1}{2.\pi.(1{,}5.\sigma)^2}.e^{-\frac{(x-u)^2+(y-v)^2}{2.(1{,}5.\sigma)^2}}$







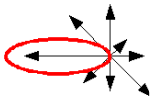




On va calculer, pour une discrétisation donnée, l'histogramme *h* décrivant la distribution de la direction de ces gradients (pondérée par leur module et leur éloignement par rapport au point d'intérêt). Autrement dit :

$$h(\theta_i, \theta_{i+1}) = \sum_{\substack{(u,v)\in V(x,y) \\ \theta_i \le \theta(u,v) < \theta_{i+1}}} m_{pond}(u,v)$$

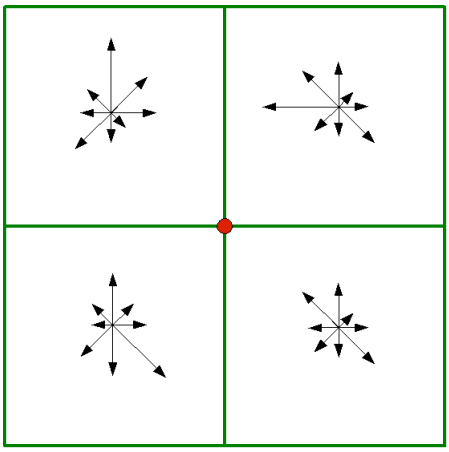
(avec V(x,y) , voisinage 2D du point de (x,y))
soit : $h(\theta_i, \theta_{i+1}) = \sum_{\substack{(u,v)\in V(x,y) \\ \theta_i \le \theta(u,v) < \theta_{i+1}}} poids(u,v,x,y).m(u,v)$



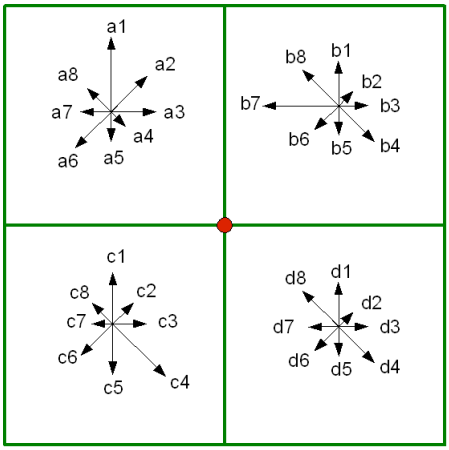
<u>Dans le cas du calcul de l'orientation</u> du point d'intérêt, on calcule un seul histogramme pour tout le voisinage du point d'intérêt. On en détermine ensuite le maximum : il correspondra à l'orientation qui va être attribué au point d'intérêt.

$$\theta_{SIFT} = \arg\max_i h(\theta_i, \theta_{i+1})$$

Au cas où l'histogramme présente un autre pic important (>80% du pic correspondant à θ_{SIFT}), on dédouble le point.



<u>Dans le cas du calcul du descripteur</u> (représenté ci-contre), on découpe le voisinage du point d'intérêt en plusieurs cases (situés de part et d'autres du point). (En pratique, la grille comporte bien entendu plus de cases que les 4 représentés ci-contre.)
On calcule un histogramme au sein de chacune de ces cases.



Les valeurs de ces histogrammes sont ensuite concaténées en un vecteur :
V=[a1 a2 a3 a4 a5 a6 a7 a8 b1 b2 b3 b4 b5 b6 b7 b8 c1 c2 c3 c4 c5 c6 c7 c8 d1 d2 d3 d4 d5 d6 d7 d8]

On normalise ensuite ce vecteur et l'on obtient alors le descripteur SIFT :

$$V_{SIFT} = \frac{V}{\|V\|}$$

Appariement (dans l'espace des descripteurs)

Notation : $d_{SIFT}(Pa, Pb)$ désigne la distance euclidienne entre les descripteurs SIFT des points *Pa* et *Pb*.
La deuxième étape consiste donc, pour chaque paire d'image, à mettre en correspondance les points de Lowe homologues. Il s'agit donc de rechercher pour chaque point de Lowe extrait de la première image son homologue (s'il existe) parmi les points de Lowe extraits de la seconde image. Lors de l'étape précédente, un descripteur a été associé à chaque point d'intérêt détecté : l'appariement va s'effectuer dans l'espace de ces descripteurs. (d'où le terme de transformation dans « Scale Invariant Feature Transform »).
La mesure utilisée pour ces mises en correspondance n'est donc pas un coefficient de corrélation, mais la distance euclidienne entre les descripteurs *SIFT* des deux points : on va chercher à minimiser cette distance. On va donc rechercher pour chaque point de Lowe de la première image son plus proche voisin parmi les points de Lowe de la seconde image au sens de la distance euclidienne entre leurs descripteurs *SIFT*. Par ailleurs, afin d'éviter des erreurs de mises en correspondance (notamment dans le cas où le point de la première image ne compte pas d'homologue parmi les points de Lowe de la seconde image), une sécurité est prise en s'intéressant, pour chaque point P de Lowe de la première image, aux deux plus proches voisins P1 et P2 parmi les points de Lowe de la seconde image : si le plus proche voisin P1 n'est pas un bien plus proche voisin (et donc un bien meilleur candidat) que le second plus proche voisin P2, c'est-à-dire si le ratio $\frac{d_{SIFT}(P,P_1)}{d_{SIFT}(P,P_2)}$ n'est pas inférieur à un certain seuil, alors on rejette la mise en correspondance. Dans le cas contraire, on considère qu'il n'y a aucune ambiguïté et la mise en correspondance est acceptée.

Par ailleurs, du fait du nombre de points de Lowe détectés et de la dimension du descripteur *SIFT*, la recherche des plus proches voisins peut être effectuée en un temps raisonnable uniquement en utilisant un algorithme de recherche approchée des plus proches voisins nommé *best-bin-first* et basé sur les *k-d-tree*.
Ainsi, on commence la phase de mise en correspondance par la construction d'un *k-d-tree* (dans l'espace *SIFT*) dans lequel les points de Lowe de la deuxième image sont triés en fonction de leur descripteur *SIFT*. Ensuite, pour chaque point de Lowe de la première image, on visite les éléments de ce k-d-tree dans l'ordre de leur plus proche distance (au sens de

SIFT) à partir de ce point (de la première image) dont on recherche les plus proches voisins. Par ailleurs, on considère que les deux plus proches voisins font partie des *n* premiers points visités et on ne parcourt donc pas l'ensemble du *k-d-tree* (d'où le terme de "recherche approchée").

Algorithme d'appariement SIFT

1 – Construction d'un k-d tree pour les descripteurs SIFT des points de l'image de référence

2 – Pour chaque point de Lowe de l'image 2, recherche de son homologue parmi les points de Lowe de l'image 1

Pour Tout point d'intérêt *P* de l'image 2 **Faire**

Recherche par l'algorithme *best-bin-first* dans le *k-d-tree* des deux points *P1* et *P2* de l'image 1 les plus proches de *P* au sens de la distance euclidienne dans l'espace SIFT (128 dimensions) avec $d_{SIFT}(P1,P) < d_{SIFT}(P2,P)$.

(L'algorithme de recherche des plus proches voisins utilisé utilise un ordre de visite du *k-d tree* modifié (visite des éléments du k-d-tree dans l'ordre de leur plus proche distance au point dont on recherche les plus proches voisins). Il s'agit en outre d'une recherche approchée puisqu'on ne parcourt pas l'ensemble du *k-d-tree* mais que l'on considère que les deux plus proches voisins font partie des *maxPtsAVisiter* (auquel on attribue la valeur 200) premiers points visités.)

Si $\frac{d_{SIFT}(P1,P)}{d_{SIFT}(P2,P)} < 0{,}8$ **Alors**

P1 est l'homologue de *P* dans l'image de référence.

Sinon

P n'a pas d'homologue parmi les points de Lowe de l'image de référence.

FinSi

FinPour

[3 – Eventuellement filtrage des mises en correspondance fausses par l'une des deux méthodes suivantes :

- par considération du voisinage (dans l'espace image 2D) des points dans les deux images :

Pour Toutes les mises en correspondance obtenues à l'étape précédente **Faire**

Si parmi les points mis en correspondance, la proportion de points parmi les *n* plus proches voisins du point courant dans l'image à recaler qui sont les homologues des *n* plus proches voisins de l'homologue de ce point dans l'image de référence est supérieure à un certain seuil **Alors**

La mise en correspondance est considérée comme valide.

Sinon

La mise en correspondance est rejetée.

FinSi

FinPour

- par élimination selon les résidus obtenus à l'issue de l'estimation des paramètres d'une transformation entre les deux images.

Références :
http://www.cs.ubc.ca/~lowe/keypoints
David. G. Lowe Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision. 2004.*

Mikolajczyk, K. et C. Schmid, 2005, *A performance evaluation of local descriptors*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 27(10):1615-1630.
Morel, J.M. et G.Yu, 2008, *On the consistency of the SIFT Method*, Rapport technique du Centre de Mathématiques et de Leurs Applications (CMLA), CMLA 2008-26.
Remondino, F., 2006, *Detectors and descriptors for photogrammetric applications*, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXVI(3):49-54.

Quelques exemples :

Les tests ont montré que la mise en correspondance de points de Lowe par la méthode décrite donne généralement de bons résultats (et ceci sans aucune connaissance géométrique *a priori* (prédicteur)). Elle est notamment peu sensible aux rotations et aux variations d'échelles ainsi qu'au bruit, aux variations d'éclairement, aux changements de points de vue et à des transformations affines limitées.
SIFT est néanmoins sensible au diachronisme (i.e. au fait que les images n'ont pas été acquises au même moment et ne se ressemblent pas), ce qui peut être une limite dans certains cas (pas de points homologues…), mais aussi une force dans d'autres (aucune mise en correspondance plus souhaitable que des appariements majoritairement faux)….

Figure 1. Mises en correspondance entre images présentant un changement de point de vue 3D.

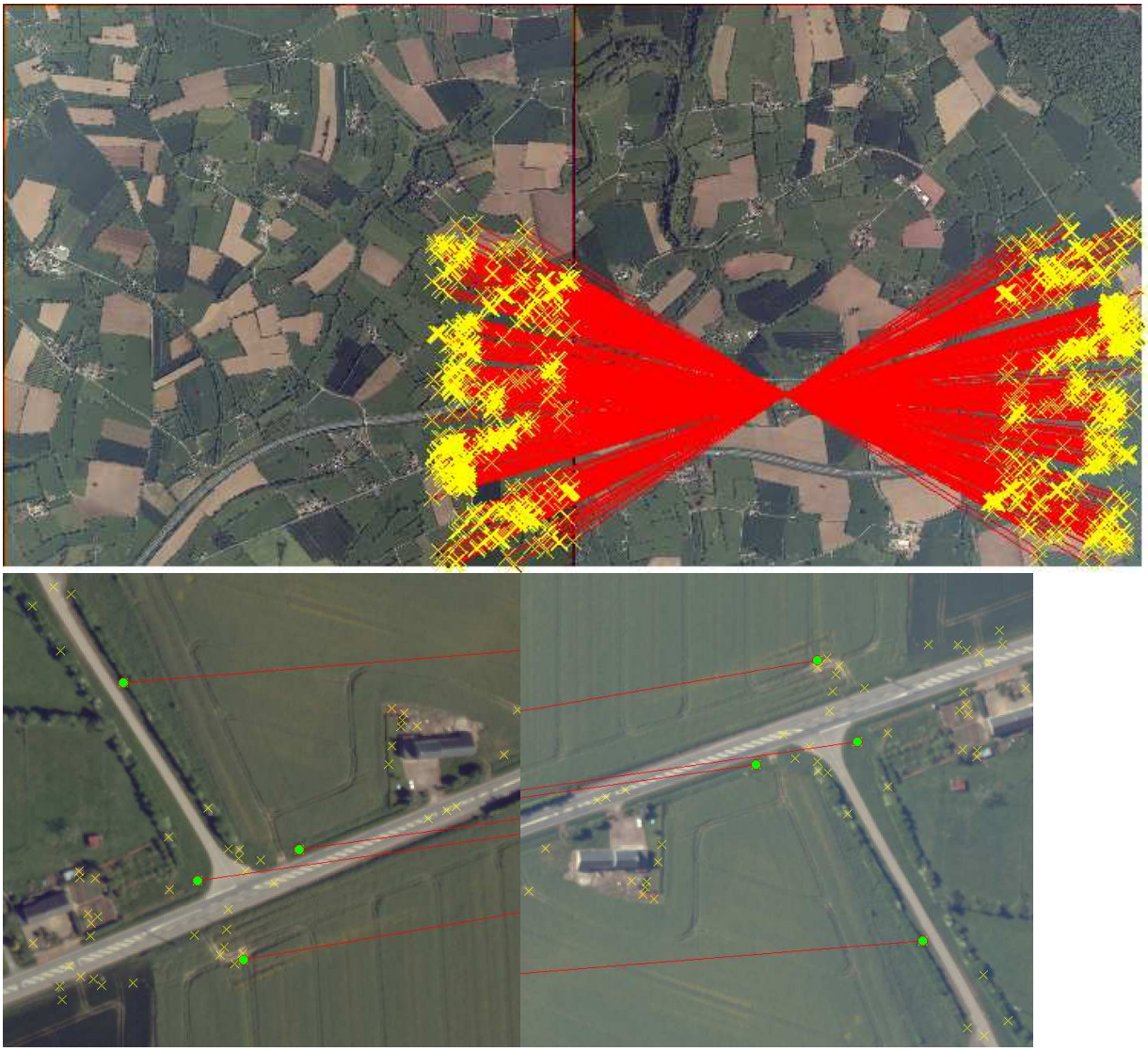


Figure 2. Mises en correspondance entre images ayant subi une rotation (image globale et détail)

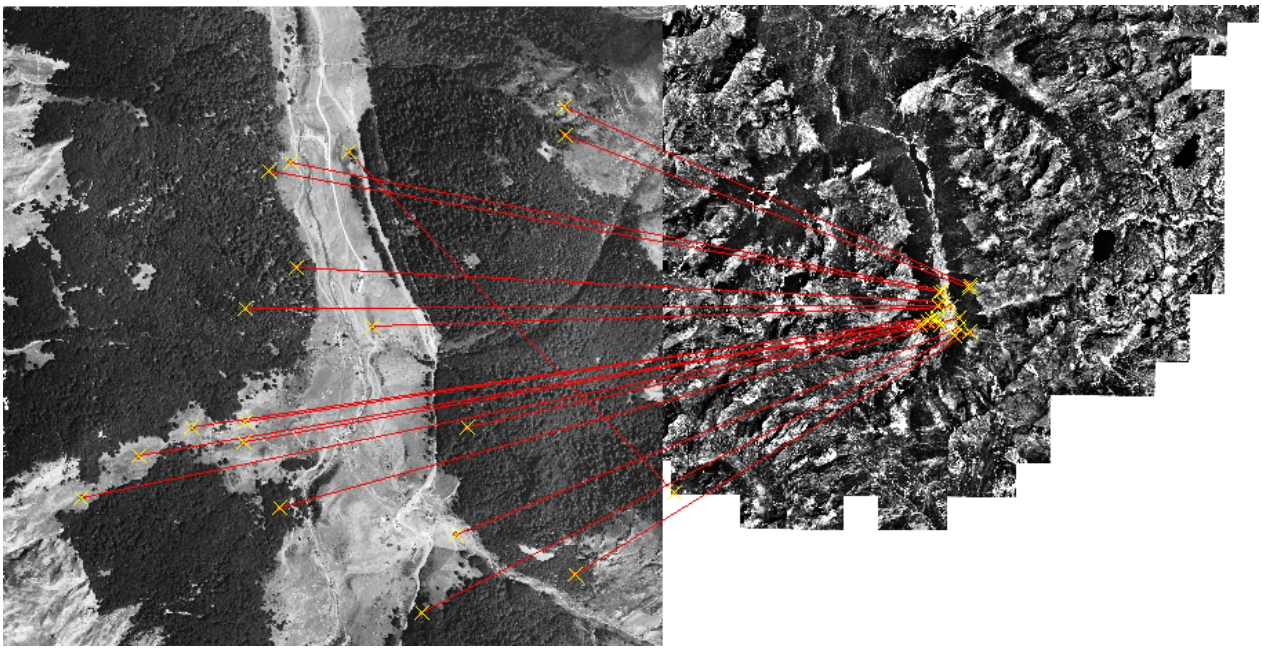


Figure 3. L'image de gauche a une résolution de 2m, celle de droite une résolution de 10m (obtenue par sous-échantillonnage de la première)

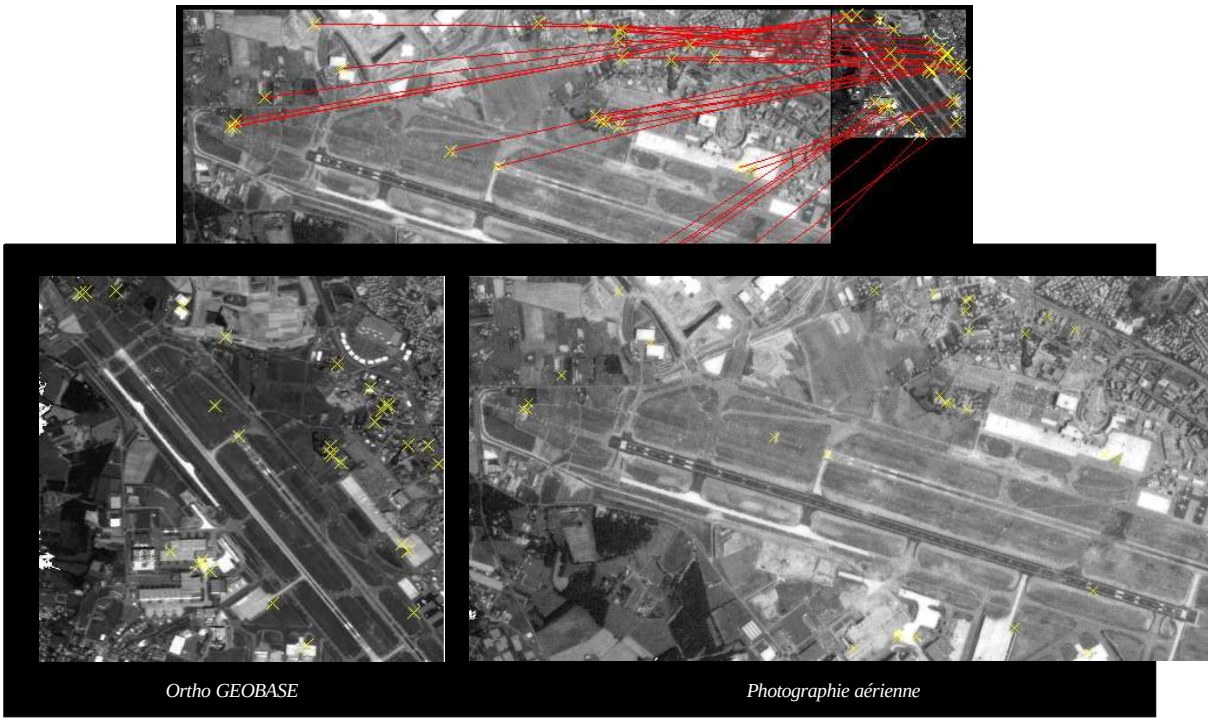


Figure 4. Mise en correspondance entre images d'échelles différentes.

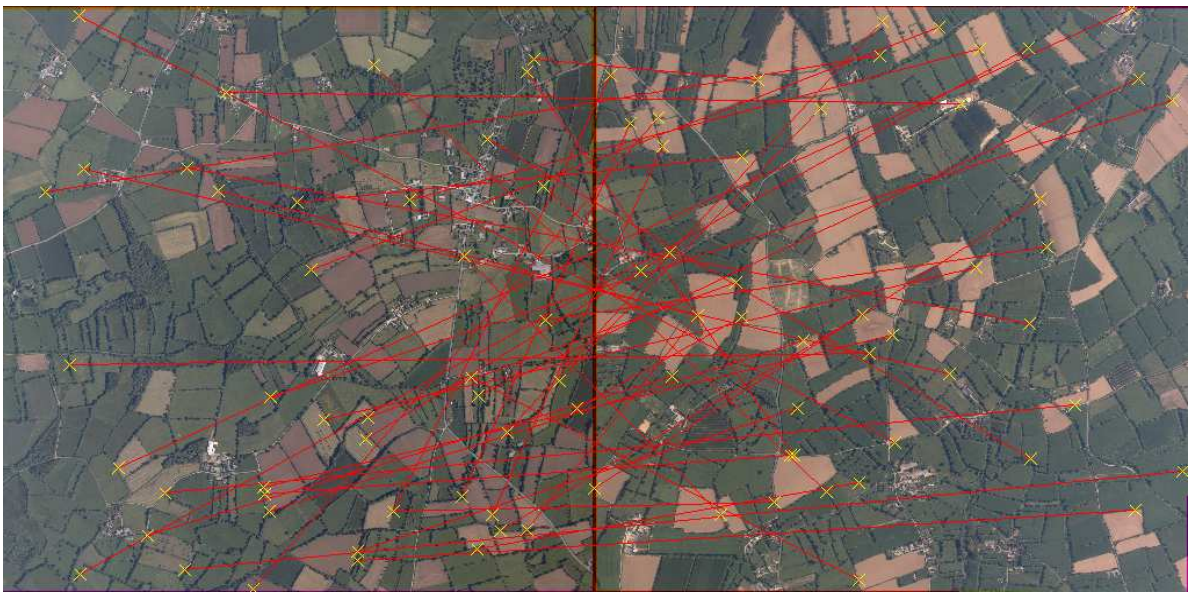


Figure 5. Echec lors de la mise en correspondance entre images acquises à un mois d'intervalle.

Le descripteur SIFT a également été utilisé comme un canal de texture pour des applications de classification.

SURF (Speeded-Up Robust Features)

Proposée en 2006, la méthode *SURF* (*Speeded Up Robust Features*) peut être considéré dans une certaine mesure comme une variante à la fois de SIFT (pour la partie descripteur) et du détecteur hessienne (pour la partie détecteur de points d'intérêt) pour laquelle un certain nombre d'approximations sont effectuées afin de permettre de réduire fortement les temps de calcul. SURF est donc également à la fois une méthode d'extraction de points d'intérêt (au voisinage desquels on va décrire le comportement de l'image au moyen d'un descripteur) et de mise en correspondance de ces points (dans l'espace de ces descripteurs). Les points fournis par SURF sont généralement moins nombreux que ceux fournis par SIFT.

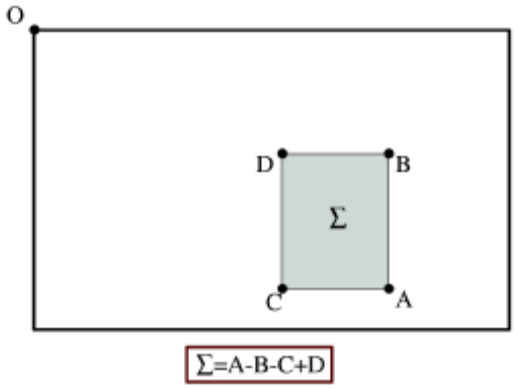
Tout comme SIFT, SURF fonctionne aussi en deux étapes :
- Les points d'intérêts sont d'abord extraits de chaque image. Un descripteur leur est ensuite associé : il va décrire le comportement de l'image au voisinage de ces points. Il s'agit d'un vecteur (de dimension inférieure à celle du descripteur SIFT).
- Pour chaque paire d'images, l'étape de mise en correspondance revient alors à rechercher pour chaque point d'intérêt de la première image son plus proche voisin dans l'espace des descripteurs parmi les points détectés dans la seconde image.

Extraction des points d'intérêt : hessienne rapide

L'**image intégrale** I_Σ d'une image I se calcule par la formule suivante :

$$I_\Sigma(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j)$$

Ainsi que le montre l'illustration suivante, la somme des valeurs des pixels au sein d'une zone rectangulaire de l'image se calcule donc simplement et rapidement à partir de l'image intégrale (en seulement 3 opérations) au lieu d'une double boucle sur les pixels de la zone si l'on travaille directement à partir de l'image. Les temps de calcul d'une telle intégrale deviennent donc indépendants de la taille de la zone d'intégration.



A – Extraction des points d'intérêt

1 – Détection des points d'intérêt

En théorie, les points recherchés sont les extrema locaux du déterminant de la hessienne de l'image convoluée par une gaussienne, c'est-à-dire avec les notations précédentes les extrema de *det(H)* avec :

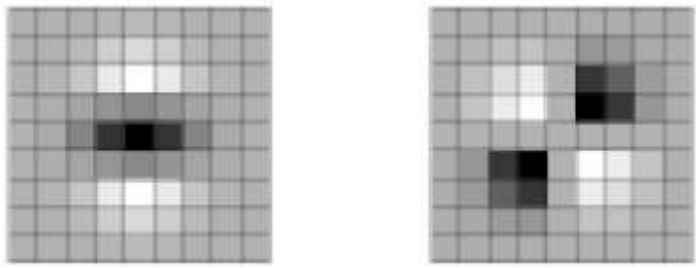
$$H(x,y,\sigma)=\begin{bmatrix} L_{xx}(x,y,\sigma) & L_{xy}(x,y,\sigma) \\ L_{xy}(x,y,\sigma) & L_{yy}(x,y,\sigma) \end{bmatrix}$$

et $L_{xy}(x,y,\sigma)=\left(\frac{\partial^2(G_\sigma * I)}{\partial x \partial y}\right)(x,y)=\left(\frac{\partial^2 G_\sigma}{\partial x \partial y} * I\right)(x,y)$

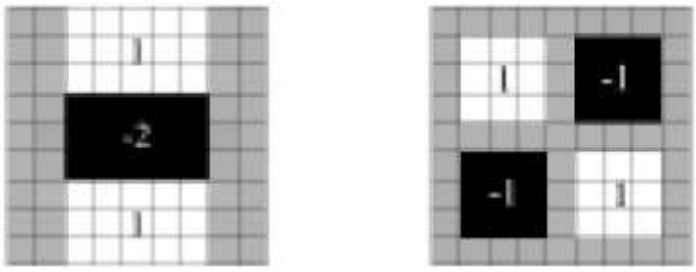
On recherche donc les extrema de $\det\big(H(x,y,\sigma)\big)=L_{xx}(x,y,\sigma)\cdot L_{yy}(x,y,\sigma)-\big(L_{xy}(x,y,\sigma)\big)^2$

Afin de réduire les temps de calcul, les $\frac{\partial^2 G_\sigma}{\partial x \partial y}$, $\frac{\partial^2 G_\sigma}{\partial x \partial x}$ et $\frac{\partial^2 G_\sigma}{\partial y \partial y}$ sont approximés par les noyaux de convolution suivants permettant l'utilisation d'images intégrales. Les temps de calcul deviennent alors indépendants de la taille du masque de convolution (et donc de la valeur de σ).

Noyaux de convolutions de $\frac{\partial^2 G_\sigma}{\partial y \partial y}$ et $\frac{\partial^2 G_\sigma}{\partial x \partial y}$



Approximations des noyaux de convolutions de $\frac{\partial^2 G_\sigma}{\partial y \partial y}$ et $\frac{\partial^2 G_\sigma}{\partial x \partial y}$



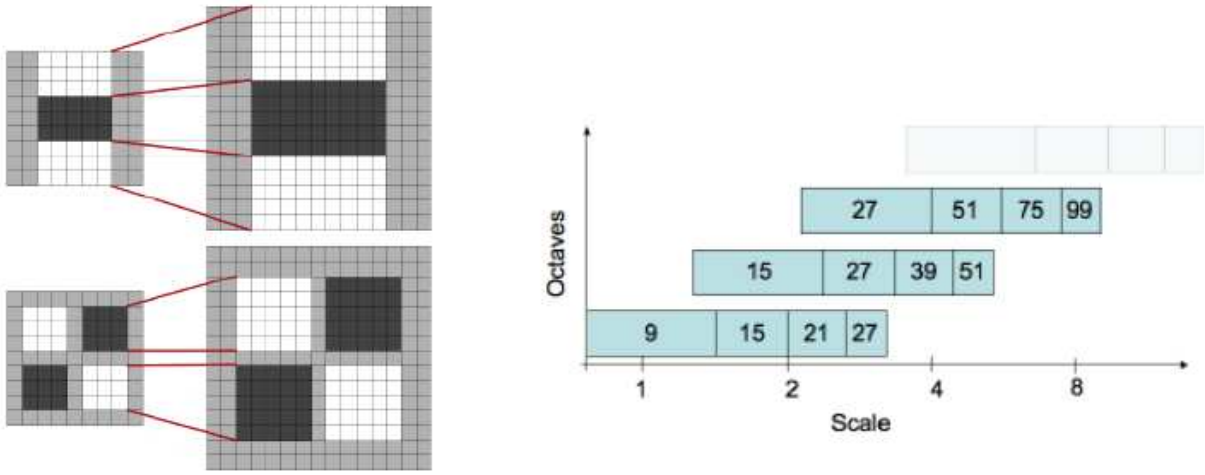
Notons respectivement D_{xx}, D_{xy} et D_{yy} les équivalents de L_{xx}, L_{xy} et L_{yy} en utilisant ces masques de convolution approximés.

Le détecteur de points d'intérêt consiste donc finalement à rechercher les maxima de la fonction suivante :

$$\det H_{app}(x,y,\sigma)=D_{xx}(x,y,\sigma)\cdot D_{yy}(x,y,\sigma)-\big(w\cdot D_{xy}(x,y,\sigma)\big)^2 \qquad \text{(avec } w = 0{,}9\text{)}$$

c'est-à-dire les maxima d'une approximation de *det(H)*.

Néanmoins, pour une telle approximation, à plusieurs valeurs de σ vont correspondre les mêmes masques de convolution approximés et donc les mêmes valeurs de D_{xx}, D_{xy} et D_{yy} (ainsi que l'illustre le diagramme ci-dessous).
La notion d'échelle est liée à la taille du masque de convolution.
La notion d'octave est conservée, mais au lieu de sous-échantillonner les images d'un facteur 2 lors d'un changement d'octave (comme pour SIFT), c'est la "vitesse" d'augmentation de la taille des masques de convolution qui augmente lors d'un changement d'octave avec SURF.



A gauche, augmentation de la taille des masques de convolution lors d'un changement de niveau (deux masques consécutifs) dans la première octave.
A droite, lien entre la taille du masque de convolution, l'octave et l'échelle.

1 – Interpolation : affinage de la position (x,y,σ) des points détectés

La méthode d'interpolation est la même que celle mise en œuvre avec SIFT.
Cette interpolation est d'autant plus nécessaire ici que plusieurs σ correspondent à un même masque de convolution approché…

B – Calcul de l'orientation et du descripteur SURF des points détectés

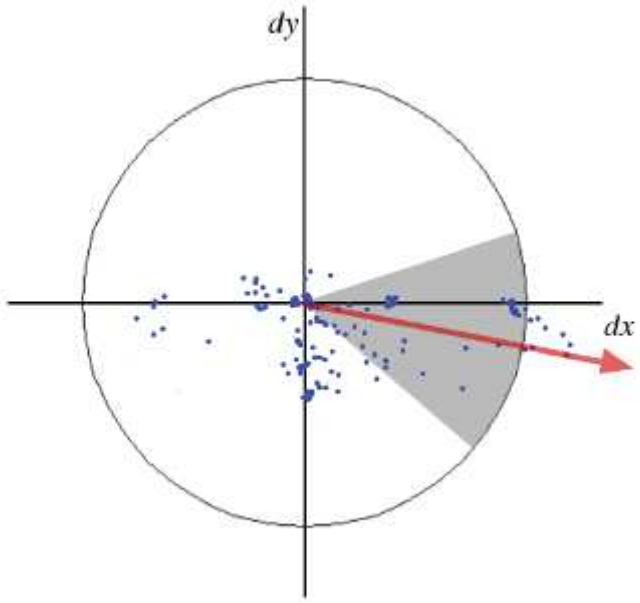
Masques de convolution des filtres de Haar ou ondelettes de Haar
en x (horizontale) et en y (verticale)

La réponse à un tel filtre peut ici encore être calculée rapidement en effectuant les calcul avec une image intégrale.

1 – Calcul de l'orientation des points détectés

Soit le point d'intérêt P d'échelle σ.
Au sein d'un voisinage circulaire (de rayon $6.\sigma$), les réponses *dx* et *dy* des ondelettes de Haar (de dimension $4.\sigma$) en x et en y sont calculées avec un pas d'échantillonnage fixé à σ, puis pondérées par une gaussienne (centrée au point d'intérêt et de sigma = $2.\sigma$).

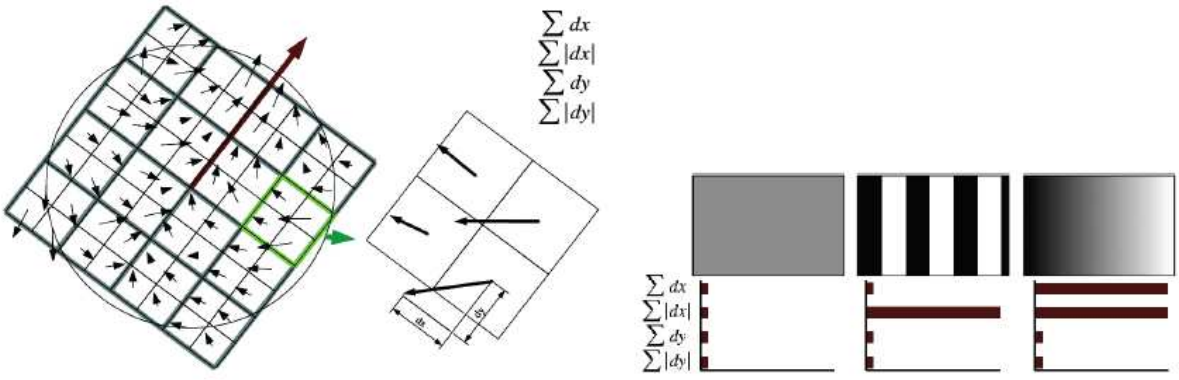
Au sein d'une fenêtre (angulaire) glissante (d'angle $\pi/3$), on calcule les sommes Σdx et Σdy de ces réponses *dx* et *dy*, puis le vecteur V de coordonnées (Σdx ; Σdy).
L'orientation attribuée à P va être l'orientation du plus long de ces vecteurs.



2 – Calcul du descripteur SURF (Speeded-Up Robust Features)
Le descripteur va être calculé relativement à l'orientation calculée précédemment.

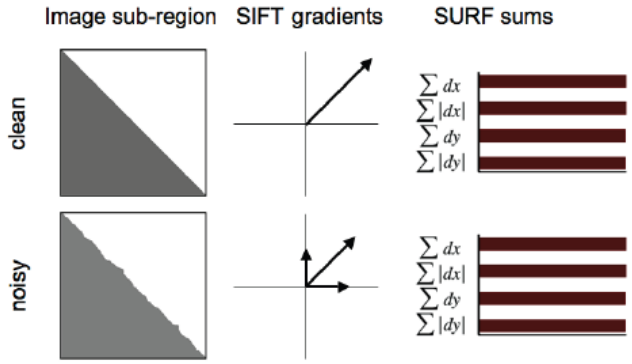
Le calcul du descripteur s'effectue au sein d'une région carrée de côté $20.\sigma$ (dirigée suivant l'orientation calculée précédemment) centrée sur le point d'intérêt. Cette région est divisée en 4x4 sous-régions (de côté $5.\sigma$) situées de part et d'autre du point détecté.
Au sein de chacune de ces sous-régions :

- Les réponses respectives aux ondelettes de Haar (de largeur $2.\sigma$) en x et en y sont calculées en 5x5 points répartis de manière régulière.
- Ces réponses dx et dy sont pondérées par une gaussienne de sigma = $3{,}3.\sigma$ centrée au point d'intérêt.
- Les sommes Σdx, Σdy, $\Sigma|dx|$ et $\Sigma|dy|$ sont ensuite calculées au sein de la case. On en déduit le vecteur $\mathbf{V}(\Sigma dx ; \Sigma dy ; \Sigma|dx| ; \Sigma|dy|)$, descripteur du comportement de l'image au sein de la sous-région.



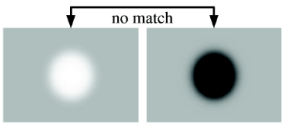
Les vecteurs **V**(Σdx ; Σdy ; $\Sigma|dx|$; $\Sigma|dy|$) des 4x4 sous-régions sont ensuite concaténées en un seul vecteur de dimension 64. Le descripteur SURF du point d'intérêt est le vecteur obtenu en normalisant ce vecteur (afin de le rendre invariant au contraste).

Un intérêt possible du descripteur SURF par rapport au descripteur SIFT est d'intégrer l'information de gradient au sein d'un voisinage (cf utilisation du filtre de Haar en plusieurs points au lieu de calculer les gradients de tous les pixels du voisinages) : cela rend l'information de SURF moins sensible au bruit (voir l'illustration suivante).



Appariement des points d'intérêt dans l'espace des descripteurs

Tout comme dans le cas de SIFT, l'appariement des points d'intérêt SURF s'effectue par une recherche de plus proche voisin au sens de la distance euclidienne dans l'espace des descripteurs.
L'utilisation du signe du *LoG* (i.e. de la trace de la matrice hessienne approximée) permet d'accélérer l'appariement. En effet, si le contraste entre deux points d'intérêt est différent (point sombre sur fond clair VS point clair sur fond sombre), alors l'appariement ne sera pas valable.



Références :
http://www.vision.ee.ethz.ch/~surf/papers.html
Bay, H., Tuytelaars, T. et Van Gool, L., SURF : Speeded Up Robust Features, *ECCV'2006*, 2006.

ASIFT (Affine SIFT) (2009)

Modèle de déformation

La déformation de l'image due à un changement de point de vue est modélisable localement par une affinité 2D (en considérant la scène localement plane) avec un modèle de caméra affine :

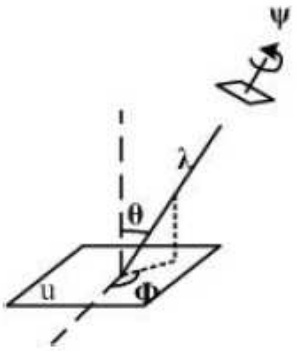
$$u(x,y) \rightarrow \quad u(ax+by+e, cx+dy+f)$$

$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ de déterminant positif

(modèle local de déformation de l'image = affinité 2D)

Il existe une décomposition unique de A :

$$A = \lambda R(\psi) T_t R(\phi) = \lambda \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$



Ψ = "camera spin" = angle de rotation dans le plan de la caméra
ϕ et θ = angles donnant le point de vue de la caméra
t appelé "*tilt*", t>=1
θ=arccos(1/t)
λ>0 avec λ.t déterminant de A
ϕ dans [0,180[

Simulations de déformations et SIFT

On dispose d'une paire d'images à apparier.

ASIFT va simuler les déformations possibles pour ces 2 images en utilisant la formule précédente avec un échantillonnage des paramètres ϕ et *t* :

Tests pour un nombre fini et réduit de configurations

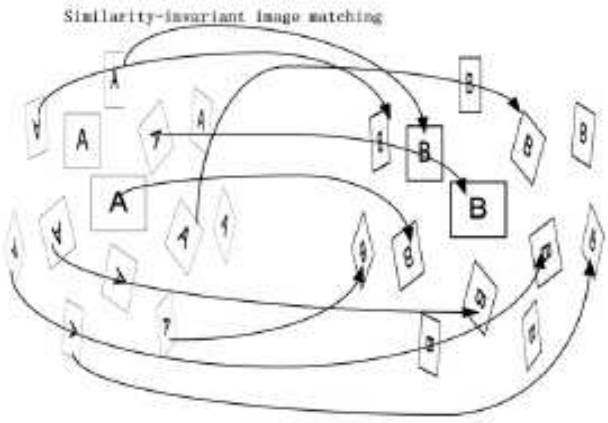
$t \in \{1, a, a^2, ..., a^n\}$ avec $a = \sqrt{2}$

Pour chaque *t*, $\phi \in \left\{0, \frac{b}{t}, ..., k.\frac{b}{t}\right\}$ avec *b*=72° et $k = Arg\max_L \left(L.\frac{b}{t} < 180° \right)$

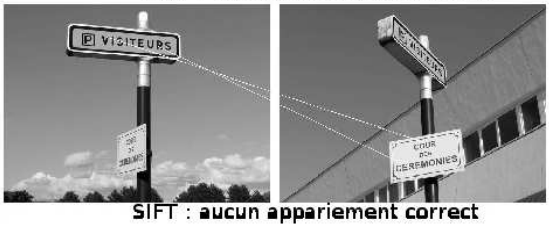
Pour chaque paire d'images simulées :

extraction et appariement des points SIFT

Les temps de calcul sont réduits en appliquant d'abord l'algorithme à des images sous-échantillonnées afin de déterminer les configurations intéressantes.



ASIFT : 50 appariements corrects



SIFT : aucun appariement correct

Références :

http://mw.cmla.ens-cachan.fr/demo/asift

Morel, J.M. et G.Yu, 2009, *ASIFT: A new framework for fully affine invariant image comparison*, SIAM Journal on Imaging Sciences, 2(2):438-469.

Conclusion

Il existe de nombreux détecteurs de points d'intérêt, et de nombreux descripteurs (invariants locaux) permettant de caractériser le comportement de l'image dans leur voisinage et pouvant être utilisés afin de les mettre en correspondance avec leurs homologues sur d'autres images.

Certaines approches proposent directement un détecteur de points d'intérêt et un descripteur comme dans le cas de SIFT. Néanmoins, rien n'empêche d'associer un détecteur de points d'intérêt avec un descripteur d'une autre méthode. Ainsi, on pourra par exemple associer des descripteurs SIFT à des points d'intérêt détectés par la méthode de Harris-Laplace.

Certaines approches ont été conçues pour être invariantes aux affinités. Néanmoins, dans la majorité des cas (affinités et déformations locales de l'image pas trop importantes), les approches simplement invariantes aux rotations 2D et aux variations d'échelles restent sufisantes.