Reconstruction stéréoscopique

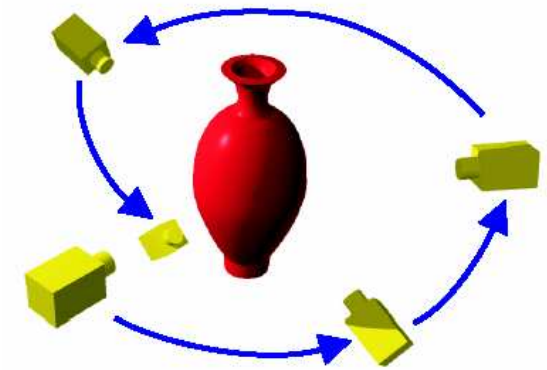
Marie-Odile Berger, INRIA Nancy Grand Est

February 20, 2017

Objectifs

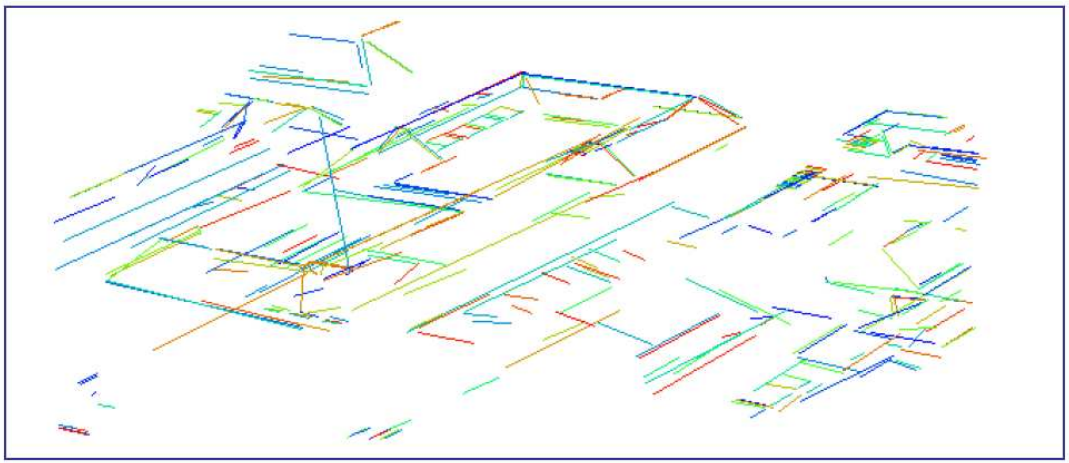
Etant données deux ou plusieurs caméras calibrées, reconstruire un modèle de la scène observée.

Intérêt de la vision stéréoscopique: peu cher, sans contrainte sur l'environnement, reconstruction dynamique possible



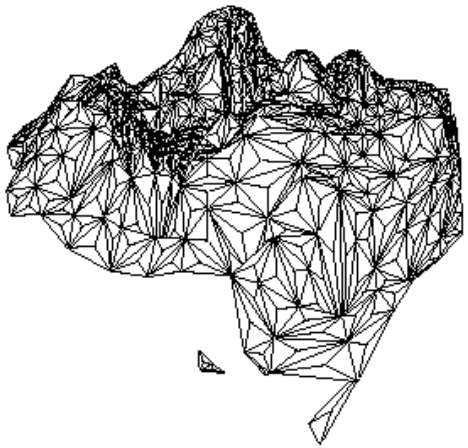
Exemples: reconstruction à partir d'images aériennes (IGN)




Exemples: reconstruction de visages



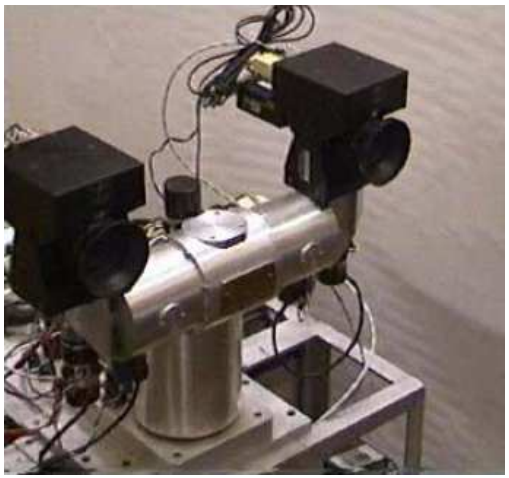



Reconstruction dynamique: virtualized reality (Kanade CMU)

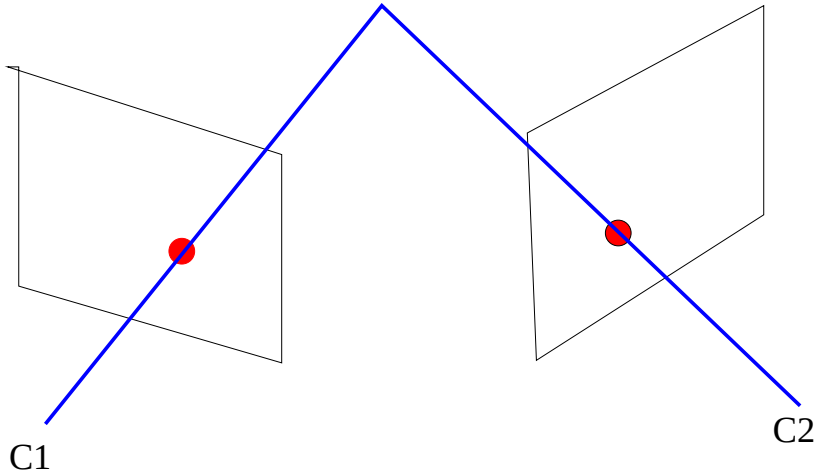
La stéréovision en pratique

Les vues peuvent être acquises

- simultanément (banc stéréoscopique, caméras synchronisés) $\rightarrow$ reconstruction possible de scènes dynamiques
- séquentiellement par une caméra en mouvement $\rightarrow$ reconstruction de scènes statiques



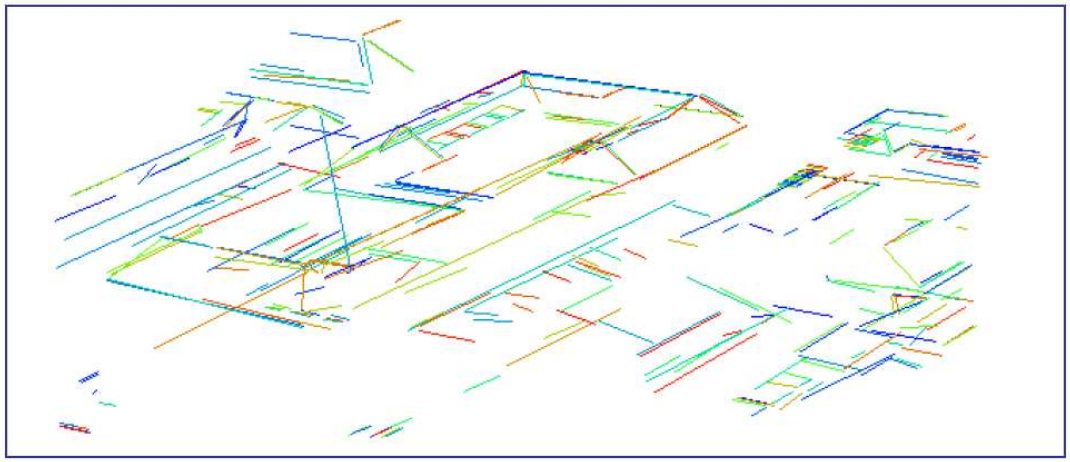
Principe de la stéréovision: la triangulation

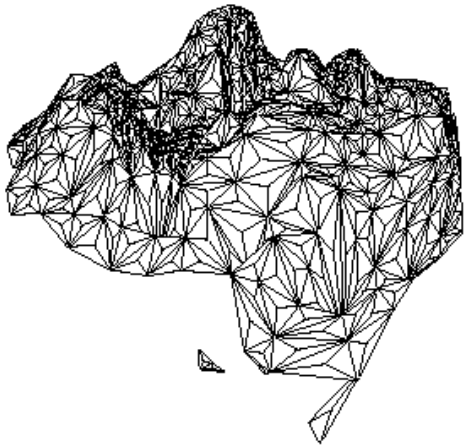


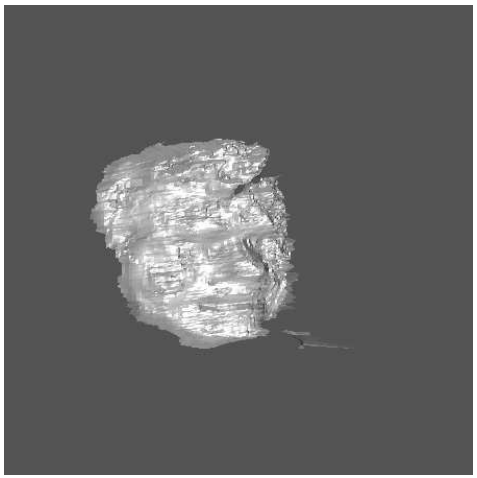
Qu'est ce qu'un modèle?

- Un ensemble de points 3D non structurés
- Un ensemble de facettes ou de primitives plus complexes que le point
- Une surface
- Un ensemble de voxels

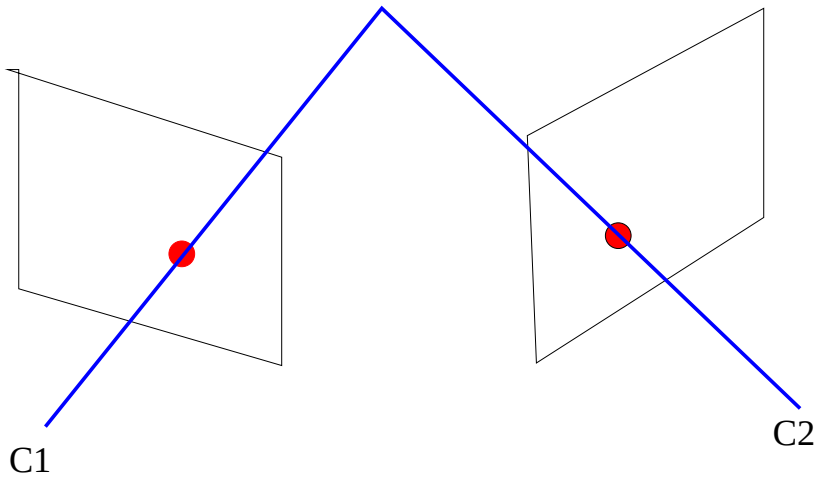
Exemples de modèles?







Principe de la stéréovision: la triangulation



- Calibrer les caméras
- Déterminer des points en correspondance entre les deux images
- Reconstruire par triangulation

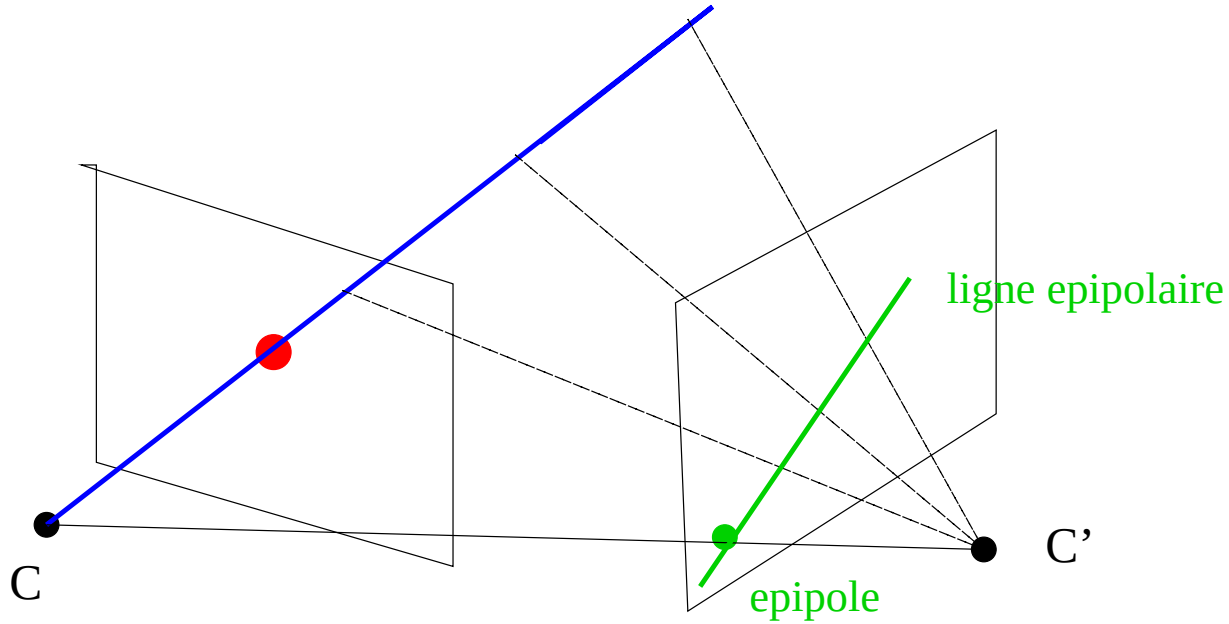
L'étape de mise en correspondance est, de loin,la plus difficile. Elle est facilité par la contrainte épipolaire qui contraint la recherche du correspondant sur une droite et pas sur toute l'image..

Part I

La géométrie épipolaire

Géométrie épipolaire

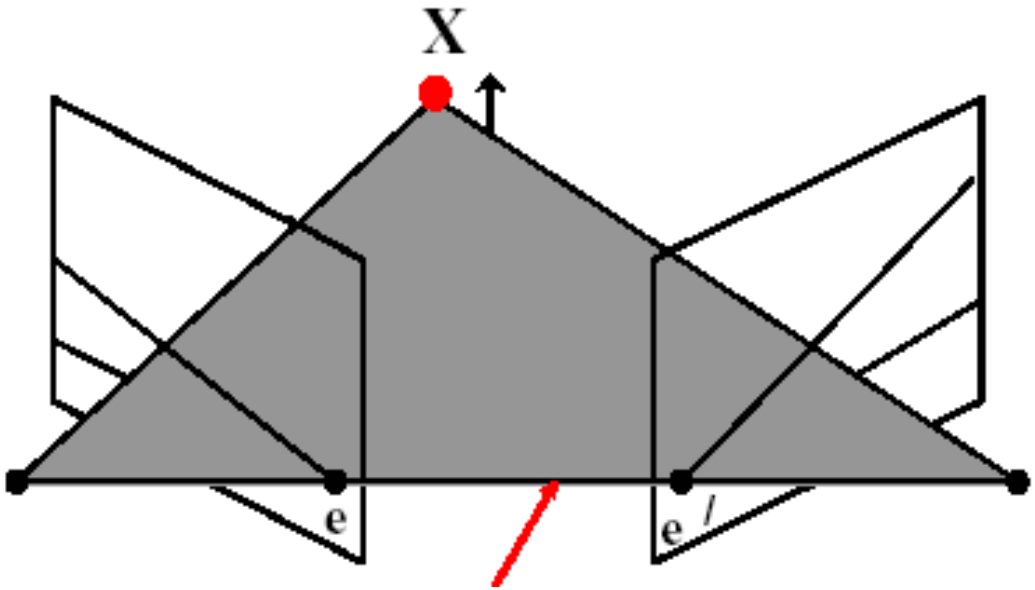
Etant donné un point dans une image, où est son correspondant dans l'autre image?



contrainte épipolaire: le correspondant d'un point appartient à une droite dans l'autre vue. La recherche du correspondant est un problème 1D.

Le faisceau épipolaire

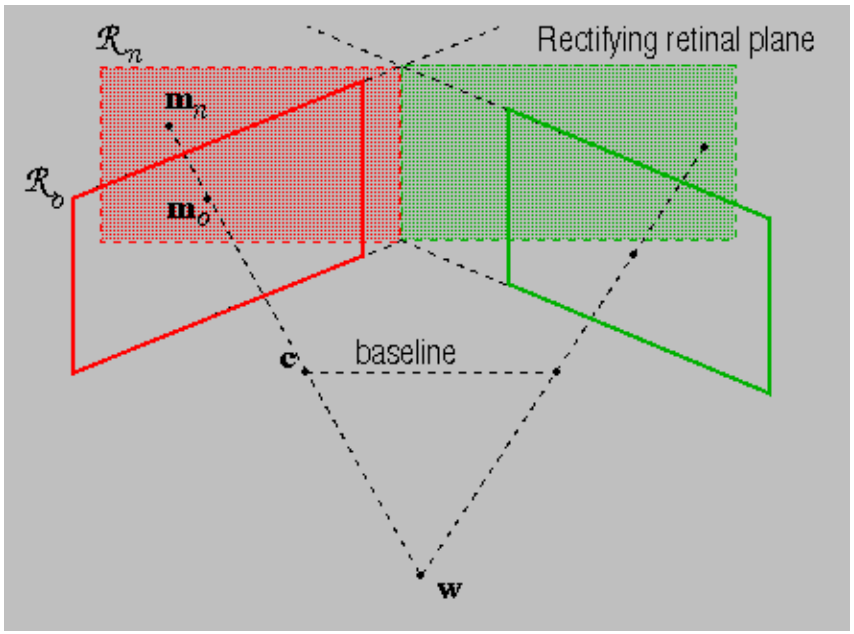
Quand X varie dans la scène, les épipolaires forment un faisceau dont de base e et e'.



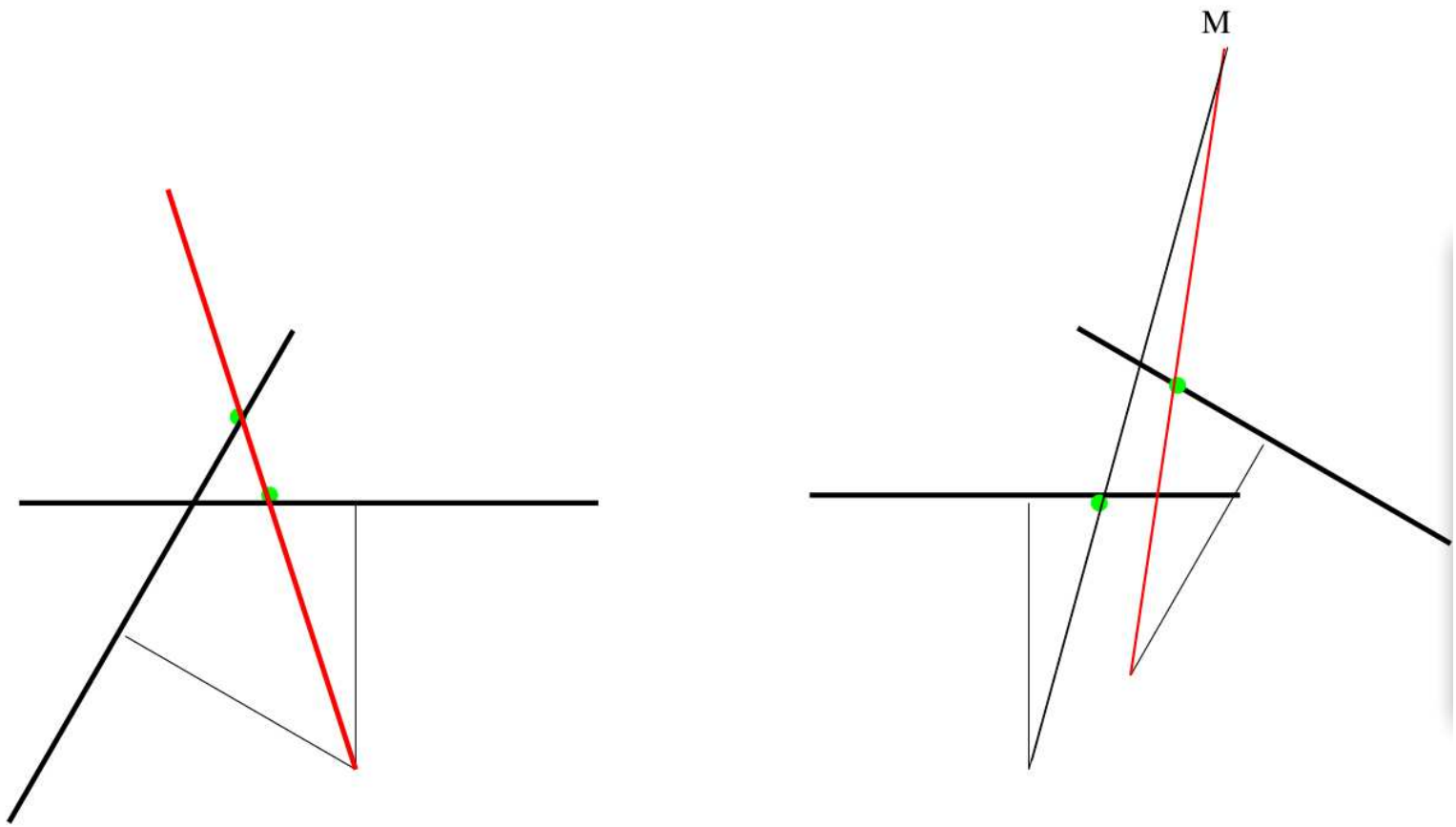
Des lignes épipolaires

Rectification d'images

Les lignes épipolaires n'ont aucune raison d'être parallèles. Mais cela facilite la recherche du correspondant.
On peut rectifier les images, c'est à dire avoir des épipolaires parallèles, si on sait calculer les images correspondant à la scène avec des plans images parallèles



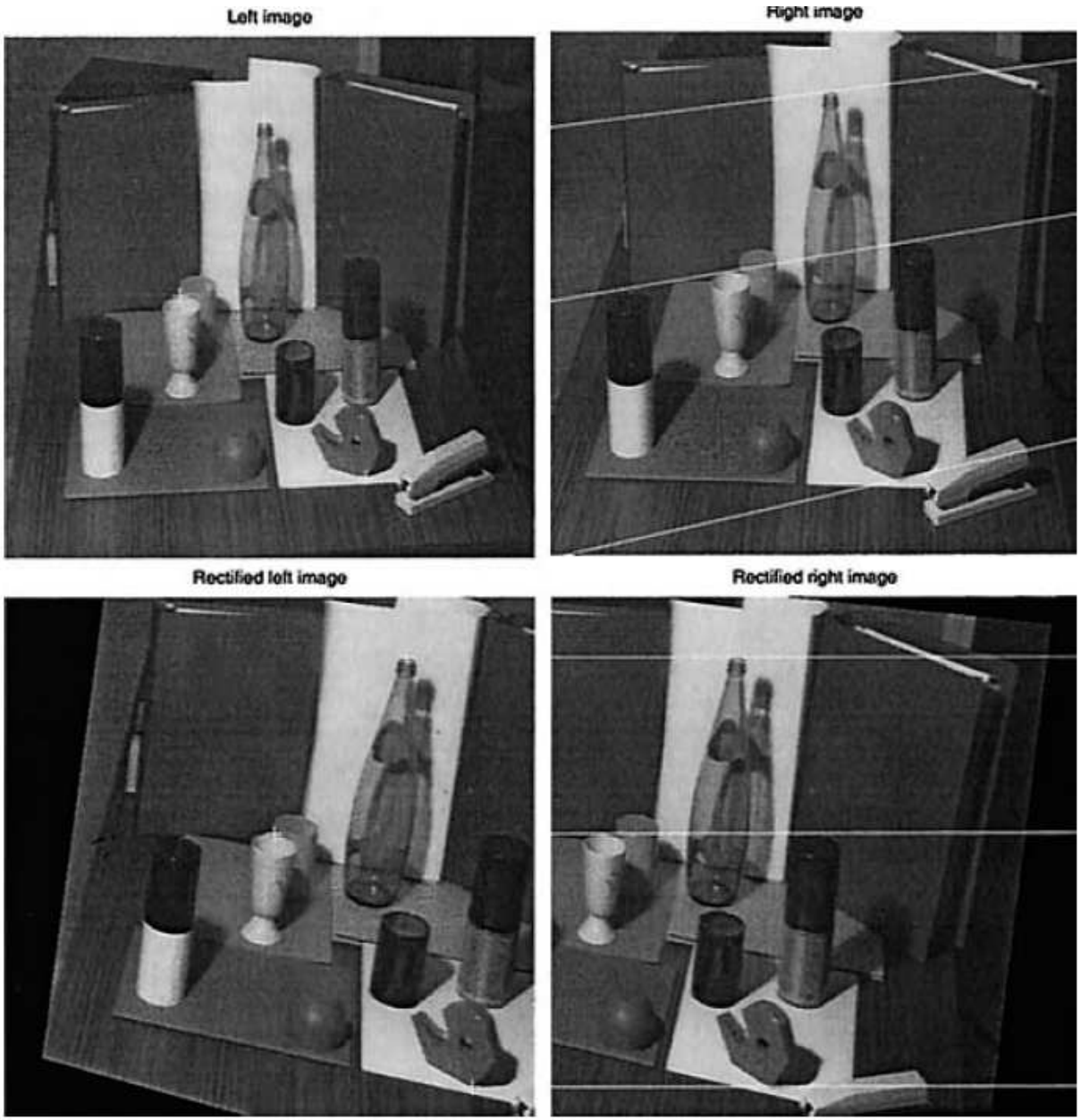
Rectification d'images



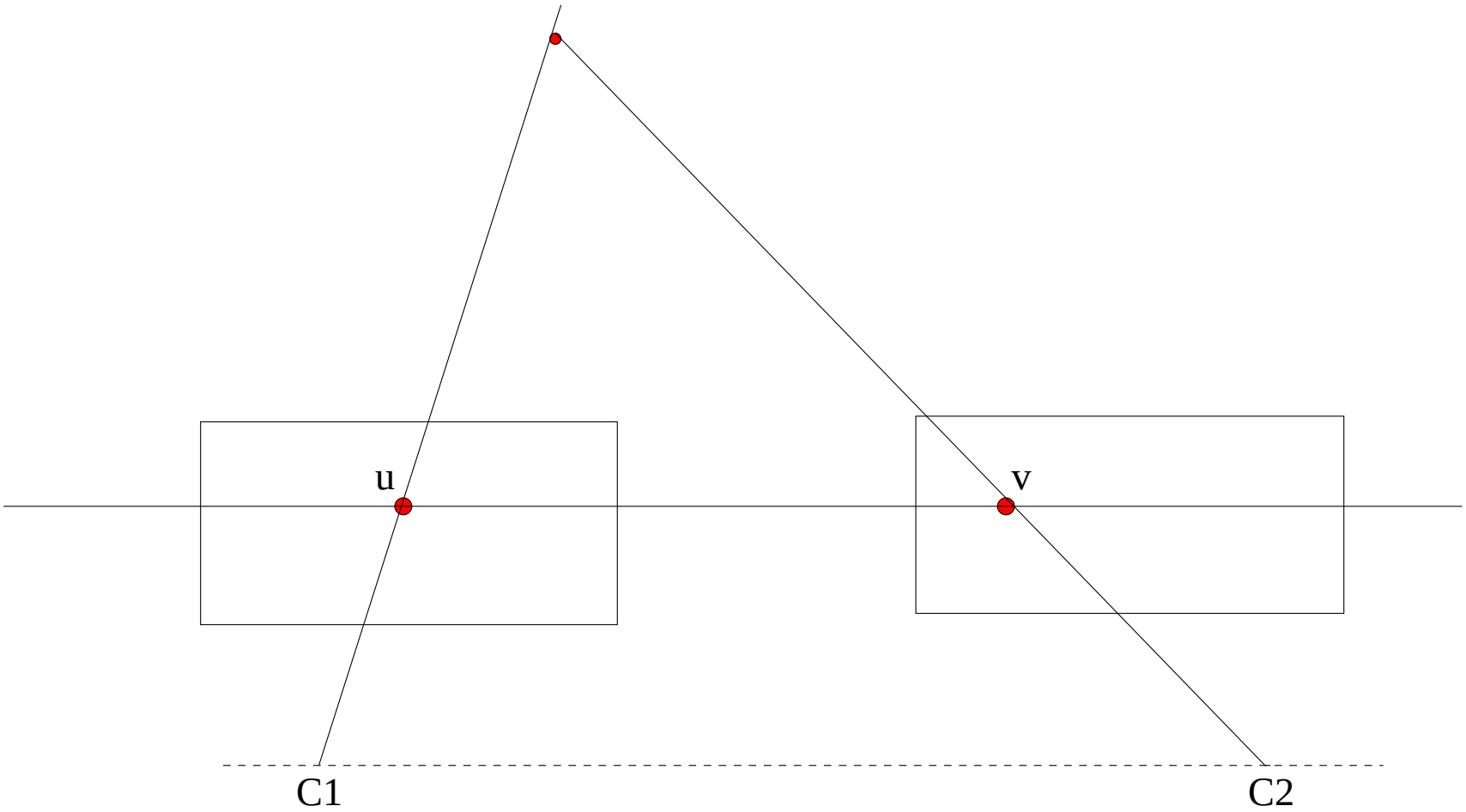
On peut générer une image d'une scène par rotation autour du centre optique sans connaitre la scène

$\Rightarrow$ Projeter les images sur un plan parallèle à la ligne de base afin d'avoir des vues rectifiées.

Rectification d'images



Géométrié rectifiée et disparité



disparité$=v-u$

Les objet proches des caméras ont une disparité forte, les objets lointains une disparité faible.

Part II

Le problème de la mise en correspondance

Quelles primitives?

On peut chercher

- A avoir une reconstruction **dense**: i.e une profondeur est calculée pour chaque pixel
- A avoir une reconstruction éparse de primitives (points, droites ou primitives spécifiques à l'environnement)

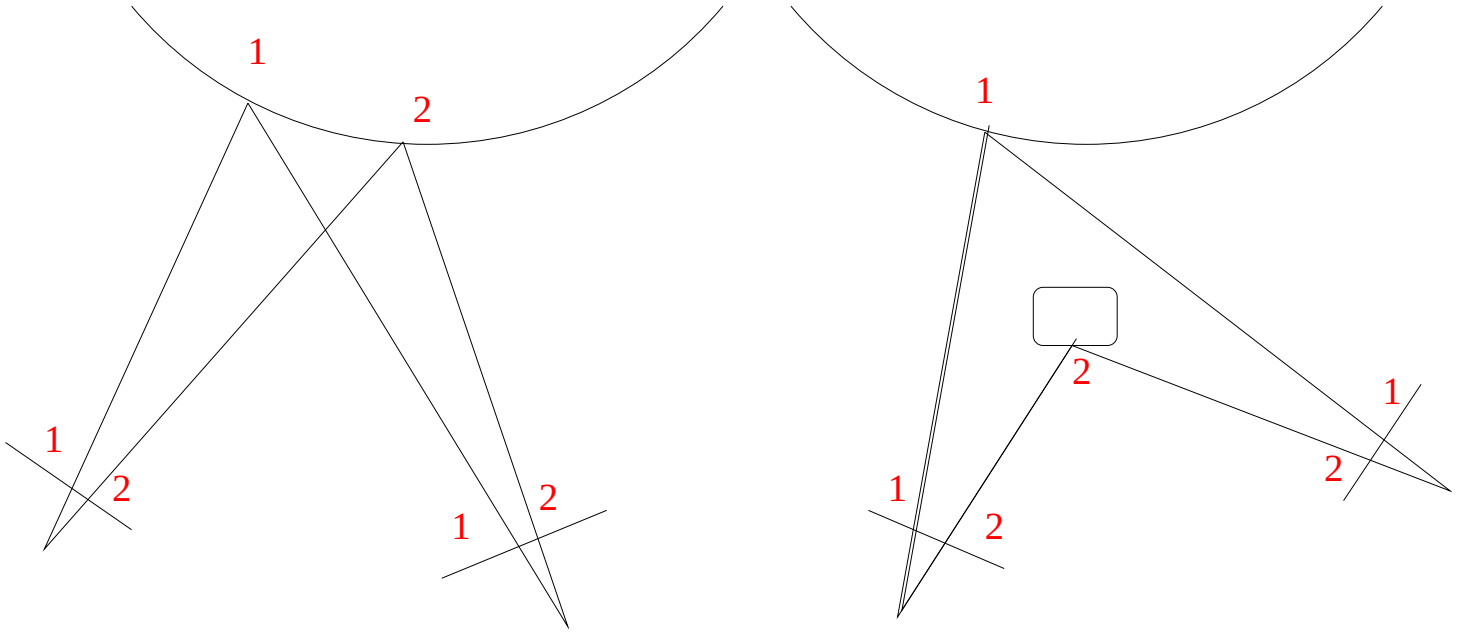
Les algorithmes de mise en correspondance dense

Etant donné un point x, x' appartient à l'épipolaire
On utilise ensuite un ensemble de contraintes faibles, le plus souvent vérifier pour aboutir à un seul appariement

- utilisation de la photométrie: x et x' ont des intensités proches (hypothèse de surface lambertienne)
- Contraintes de voisinage: des points voisins sur l'image gauche ont des correspondant voisins sur l'image droite
- contrainte d'ordre: l'ordre des points est respecté sur les épipolaires

Que penser des contraintes faibles?

Les contraintes d'ordre sont souvent vérifiées presque partout... mais problème au niveau des discontinuités de profondeur.



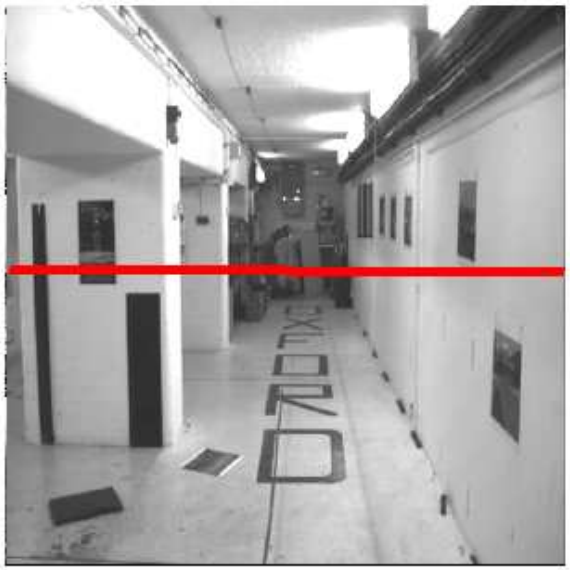
Que penser des contraintes faibles?

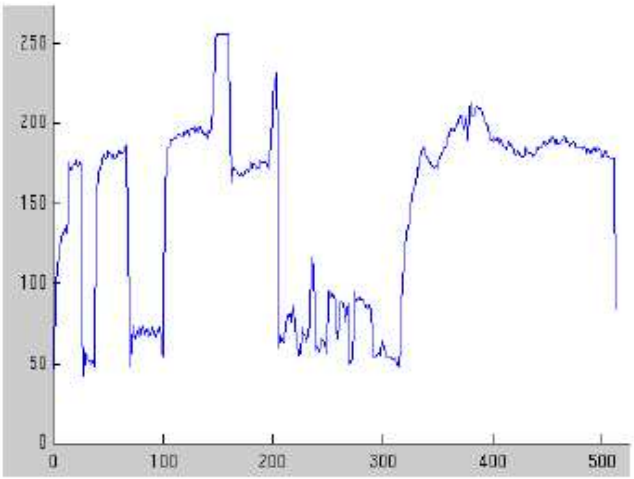
La contrainte d'appariement unique:

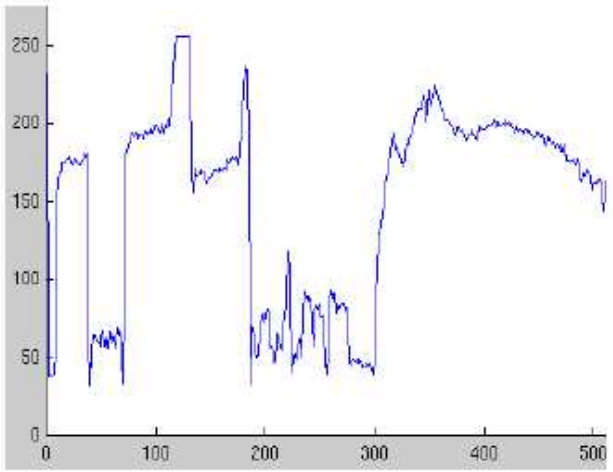
- plusieurs appariements peuvent exister (alignements fortuits)
- il peut n'exister aucun appariement (occultation)

Analyse des correspondances

Considérons des images rectifiées et le profil d'intensité sur des épipolaires homologues







La mise en correspondance par corrélation

principe: étant donné un point m_1, déterminer le point m_2 de l'image 2 tel que les intensités dans un voisinage (fenêtre rectangulaire) de m_1 et de m_2 soient semblables. Il suffit de calculer la coorélation le long de l'épiplaire. Pour des images rectifiées:

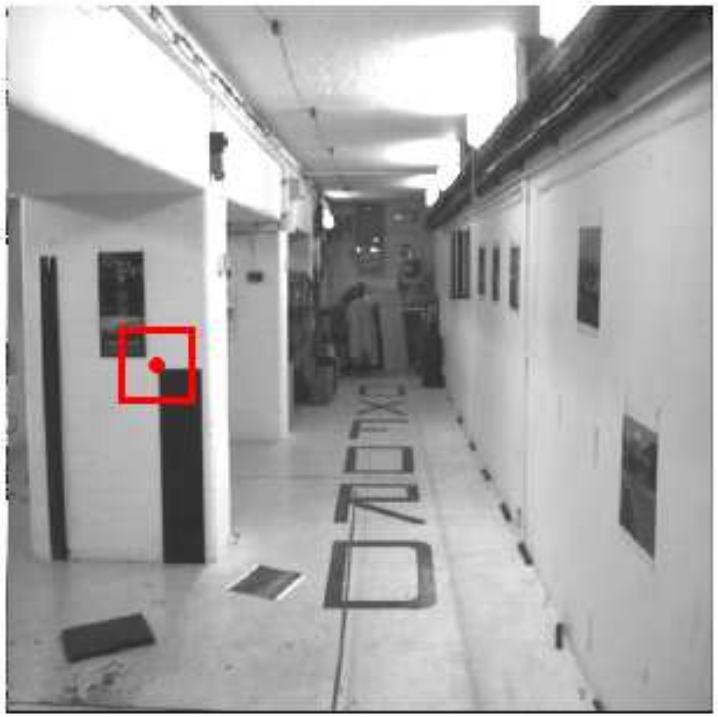
$$C(\tau) = \frac{1}{K} \sum_{i=-N}^{i=+N} \sum_{j=-P}^{j=+P} (I_1(u_0+i, v_0+j) - \overline{I_1(u_0, v_0)})(I_2(u_0+i+\tau, v_0+j) - \overline{I_2(u_0+\tau, v_0)})$$

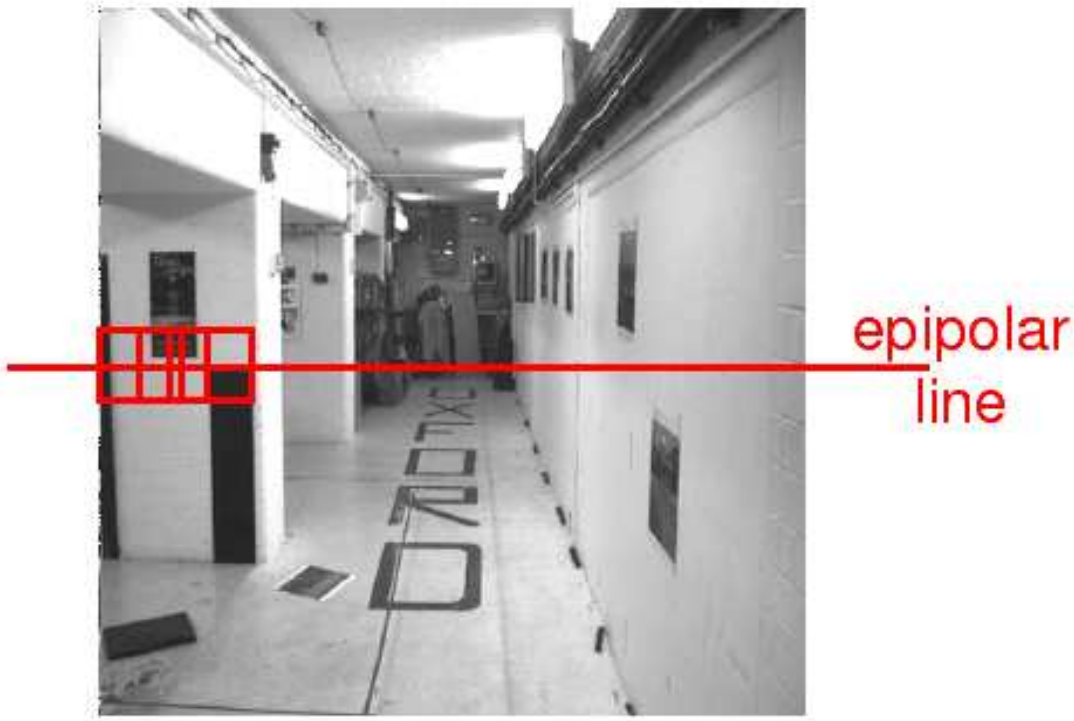
avec $K = (2N+1) * (2P+1) * \sigma_1(u_0, v_0)\sigma_2(u_0, v_0)$.

C passe par un maximum pour $\tau = \tau_0 = \textit{disparite}$

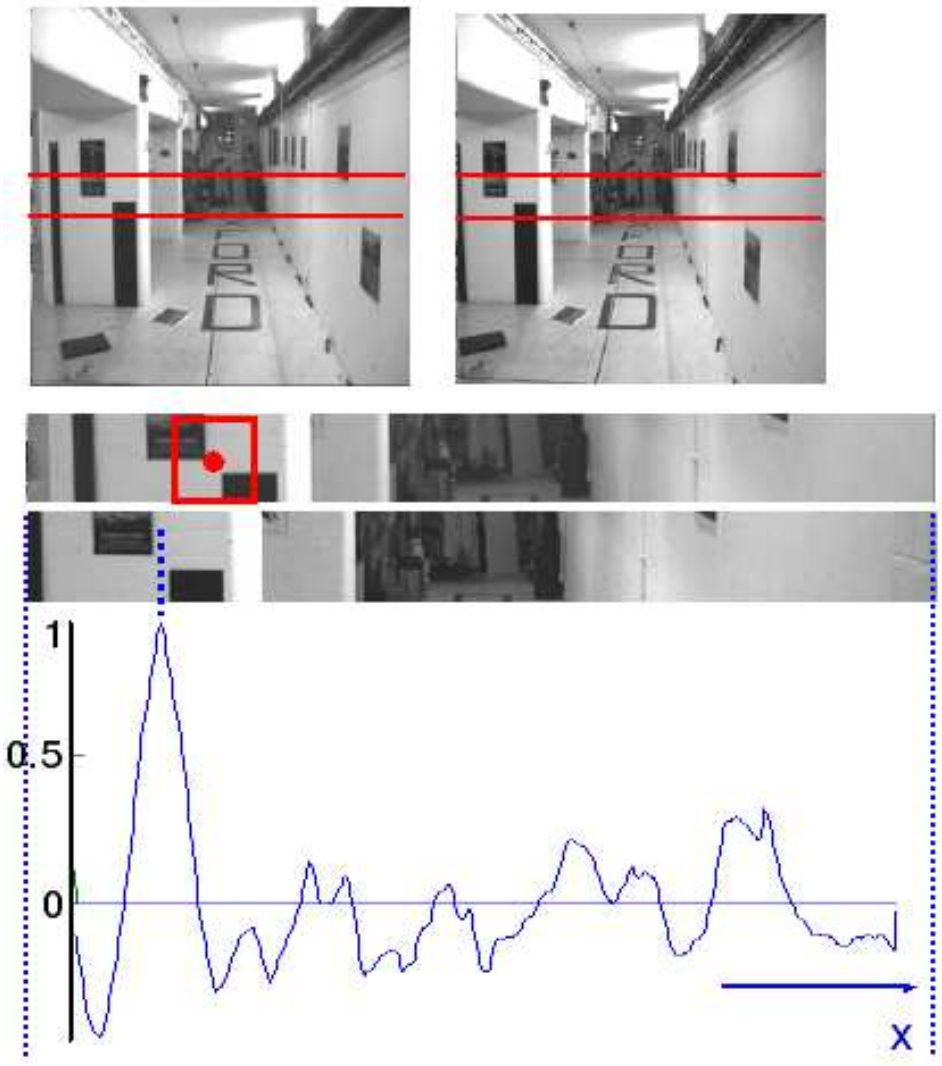
RQ: la correlation est invariante à un changement affine d'intensité.

La mise en correspondance par corrélation

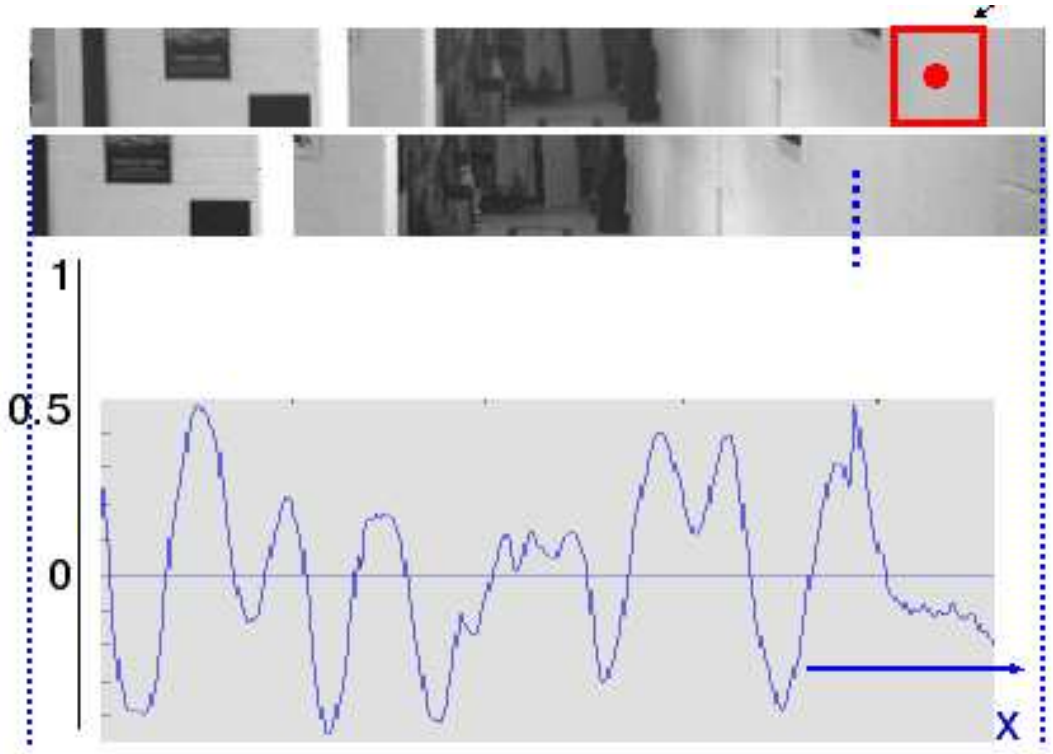




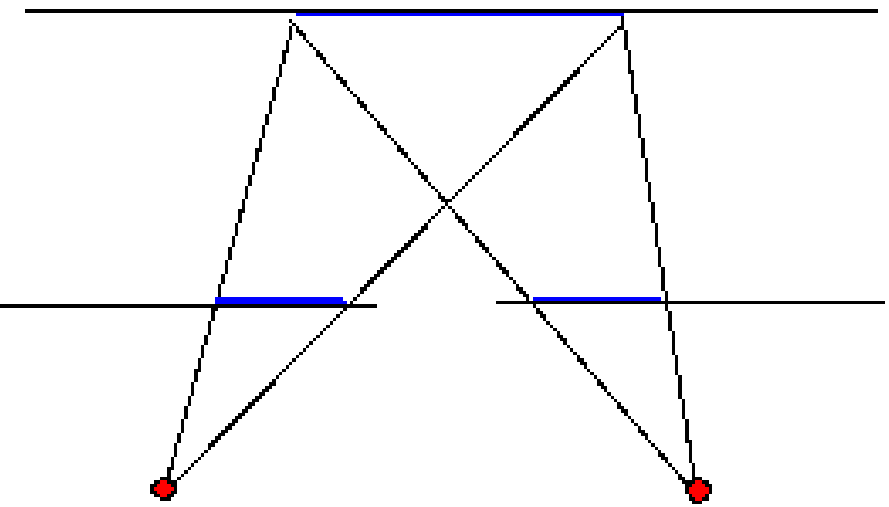
La mise en correspondance par corrélation

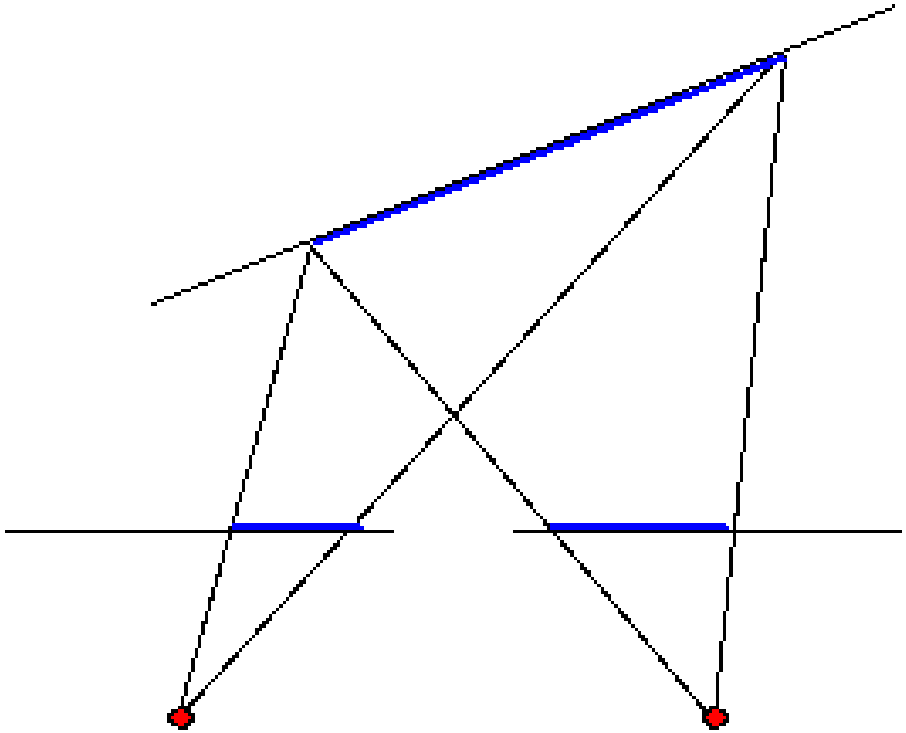


Les limites de la corrélation



Les limites de la corrélation





Influence de la taille de la fenêtre

Exemples de calcul de disparité avec deux fenêtres de taille 3 et 20.

- petite fenêtre: plus de détails
- grande fenêtre: moins d'erreurs isolées

Les limites de la corrélation

Limitations de la méthode:

- peu précis si le maximum de τ est épais
- suppose qu'il n'y a pas de déformation entre les deux images (vrai si la partie observée correspondant à la scène est parallèle au plan image). Mais des travaux d'adaptation de formes de fenêtres existent [Devernay94] avec des paramètres de déformation à optimiser en plus.
- problèmes aux frontières de discontinuité.

Algorithme de mise en correspondance

- Pour chaque point de l'image gauche, calculer un point dans l'image droite appartenant à l'épipolaire et présentant le score de corrélation le plus haut
- paramètres à adapter: taille de la fenêtre, disparité maximale
- d'autres contraintes peuvent ensuite être utilisées pour s'assurer des contraintes d'ordre, ou d'un champ de disparité assez régulier ou éviter la contrainte d'unicité (utilisation d'algorithme de relaxation ou de programmation dynamique permettant d'intégrer des contraintes à partir des détections initiales)
- bien adapté à des scènes relativement fronto parallèles ne présentant pas de trop nombreuses discontinuités de profondeurs

Reconstruction stereo: [Yuille & Geiger]

Part III

Reconstruction discrète et mis en correspondance

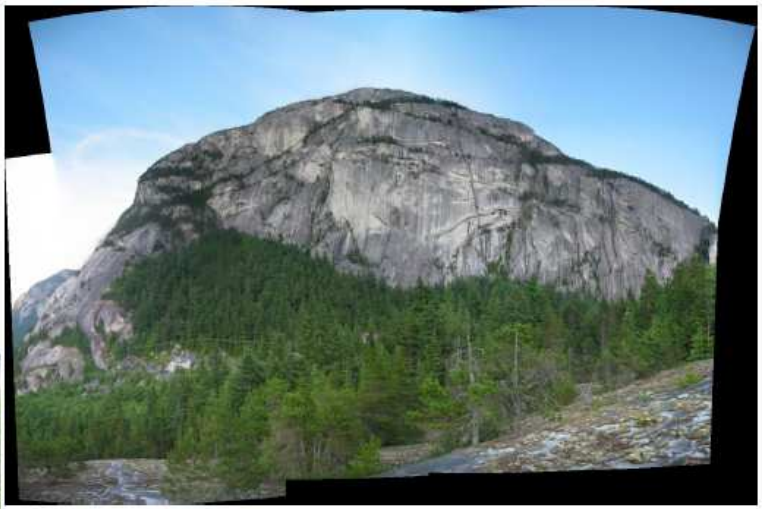
Les points d'intéret

- un indice apparu plus tardivement que le contour mais qui a produit d'excellents résultats
- Extraire des points sans sémantique mais très reproductibles (i.e faciles à mettre en correspondance entre deux images)
- fournit une information non dense mais sure
- Exemple de video

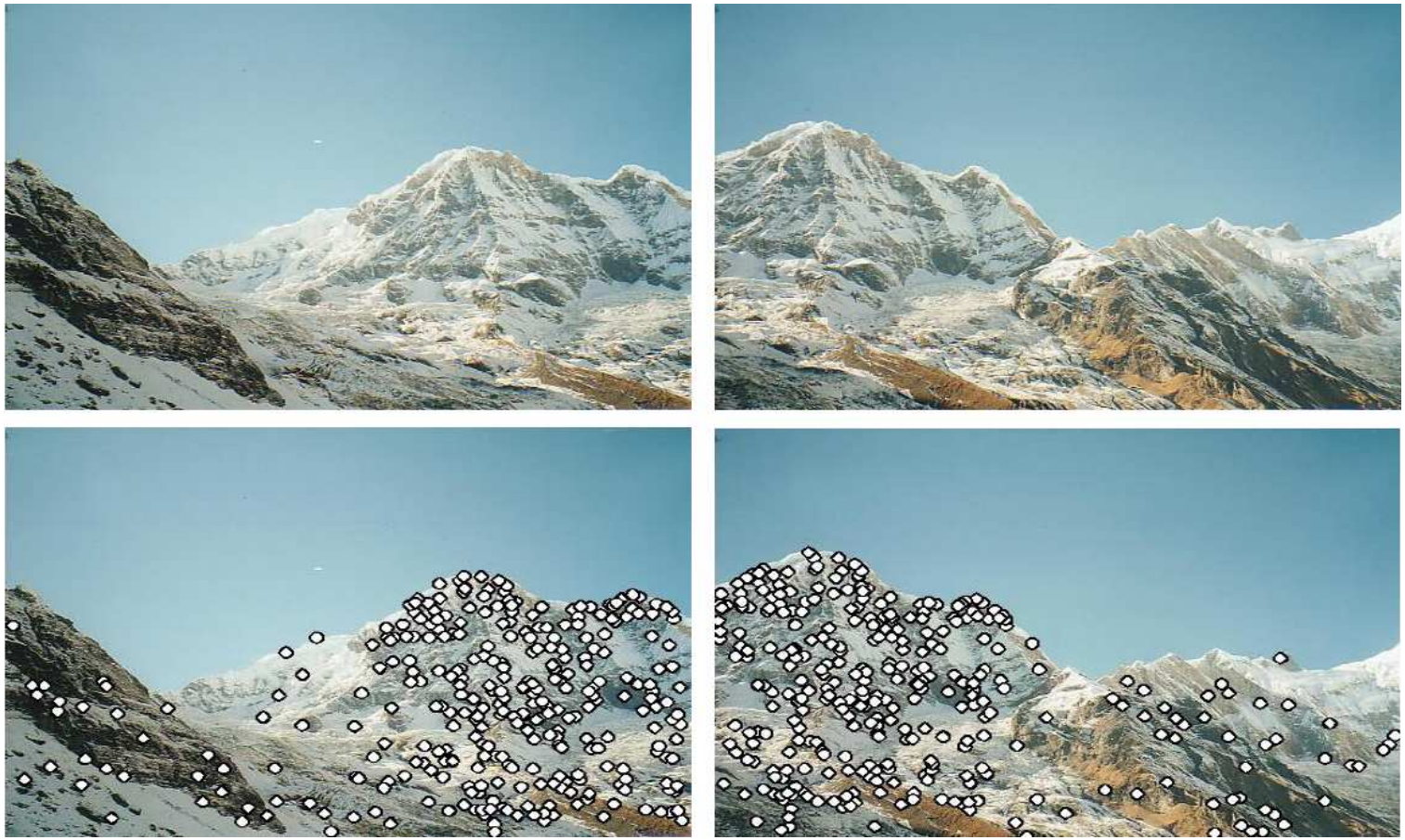
Exemple d'utilisation des points d'intéret

Fabriquer un panoramique

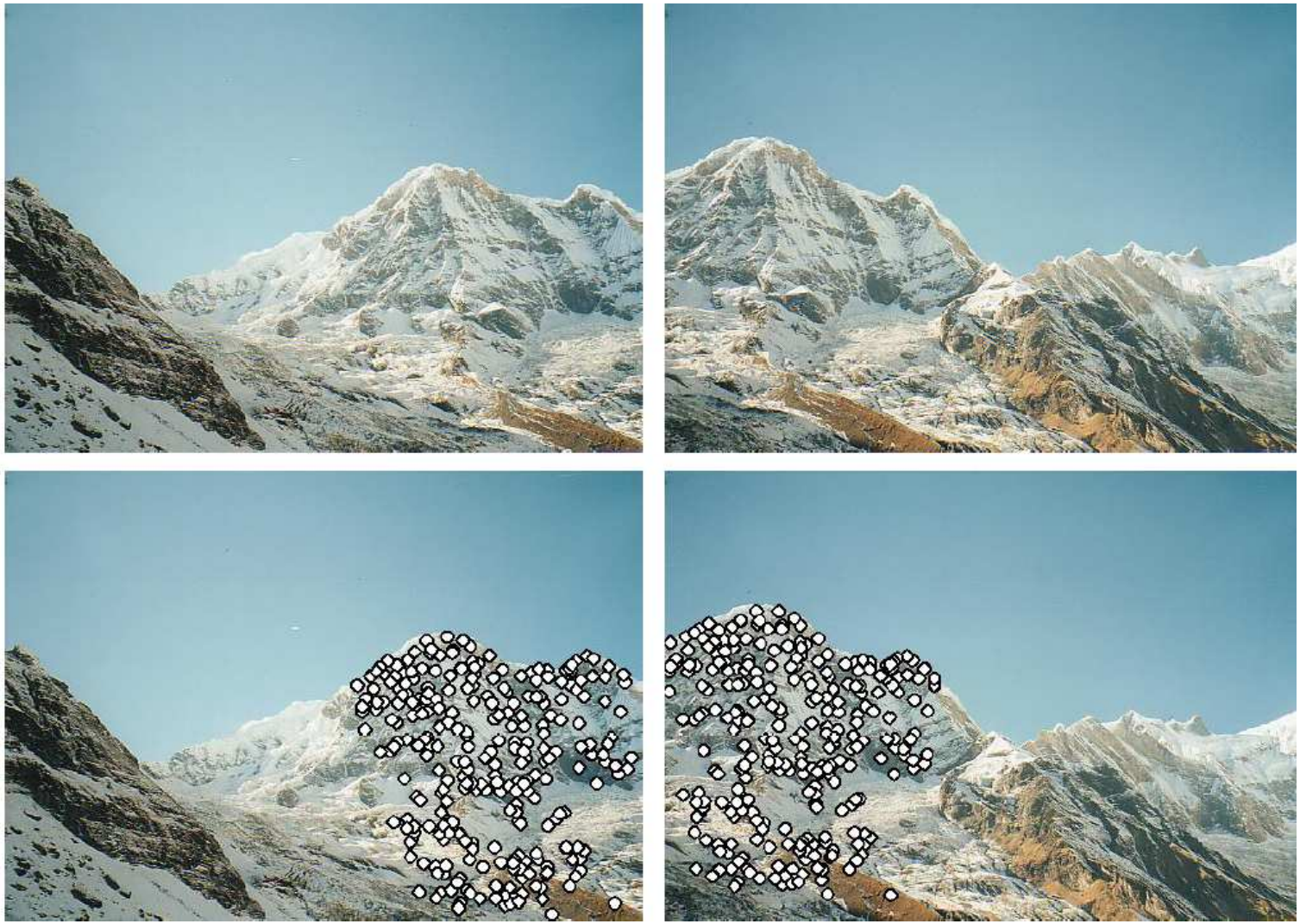


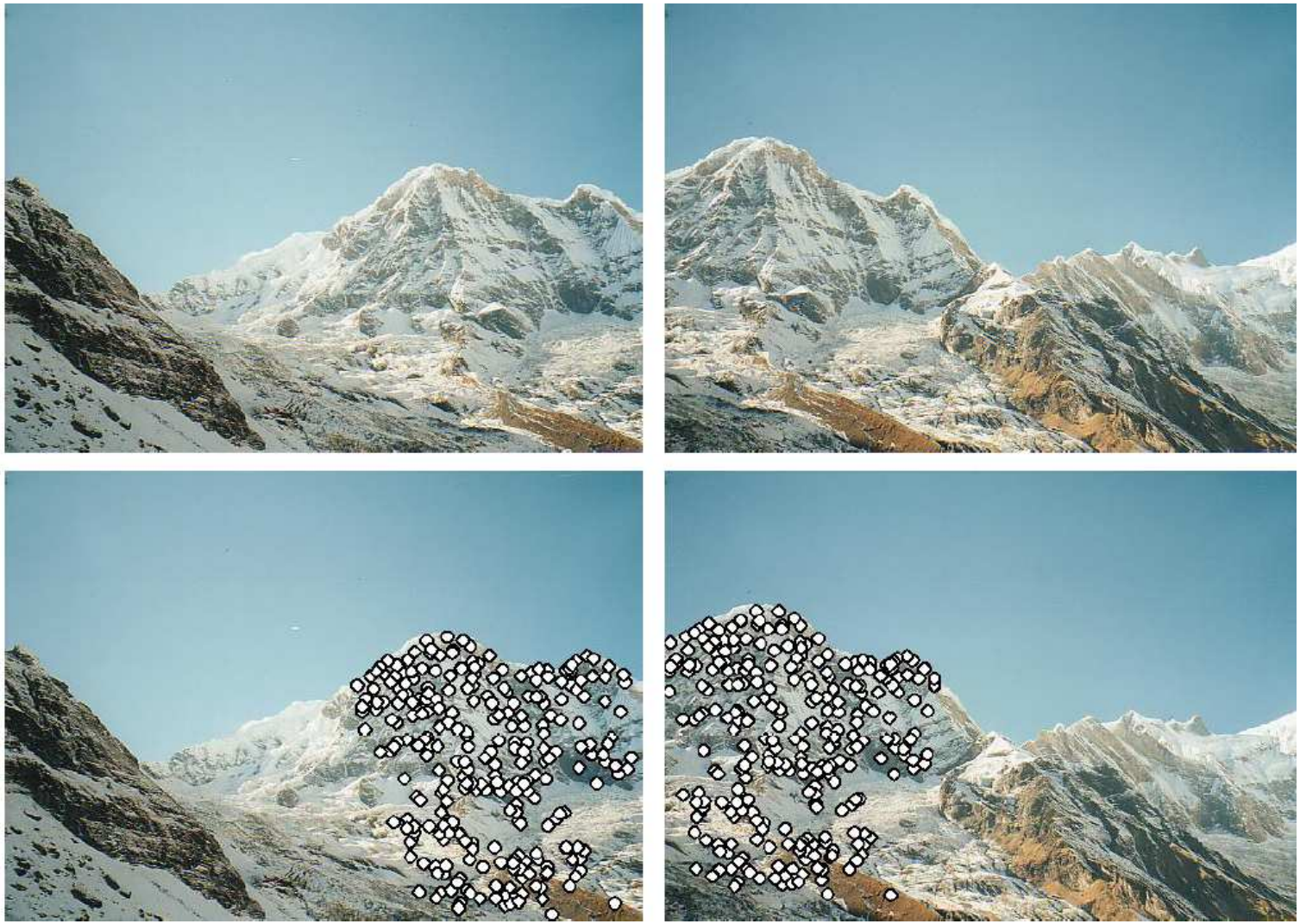

Extraction des points d'intéret



Mise en correspondance sous contrainte homographique

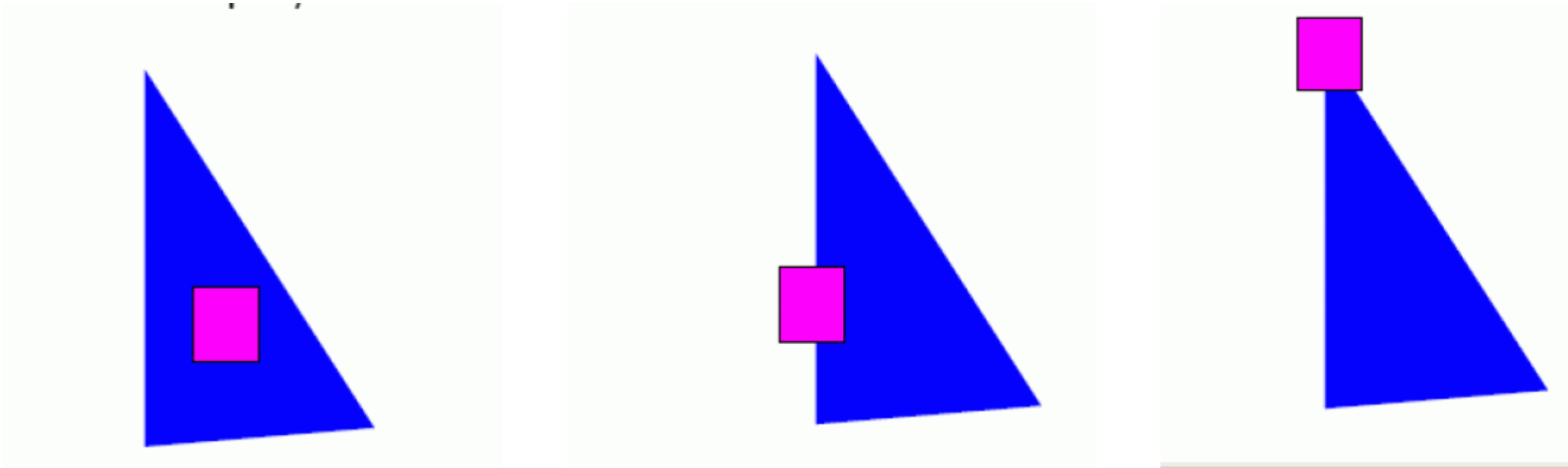


Recollement



Extraire les points d'intéret: les points de Moravec (1977)

- Idée de base: observer les changements d'intensité en déplacant localement une fenetre dans les 8 directions principales (axes + diagonale) [Moravec80]



a b c

- a: aucun changement; b: pas de changement dans la ditection du contour; c: changement dans toutes les directions.

Les points de Moravec

- Calculer le changement induit par une translationn u, v:

$$V_{u,v}(x,y) = \sum_{a,b \in fenetre}^{h} (I(x+u+a, y+u+b) - I(x+a, y+b)^2$$

$$caracteristiqueV(x,y) = min_{u,v} V_{u,v}(x,y)$$

- Détection des changements limités aux axes
- Extension de Harris: étendre cette méthode à des directions de variation quelconques

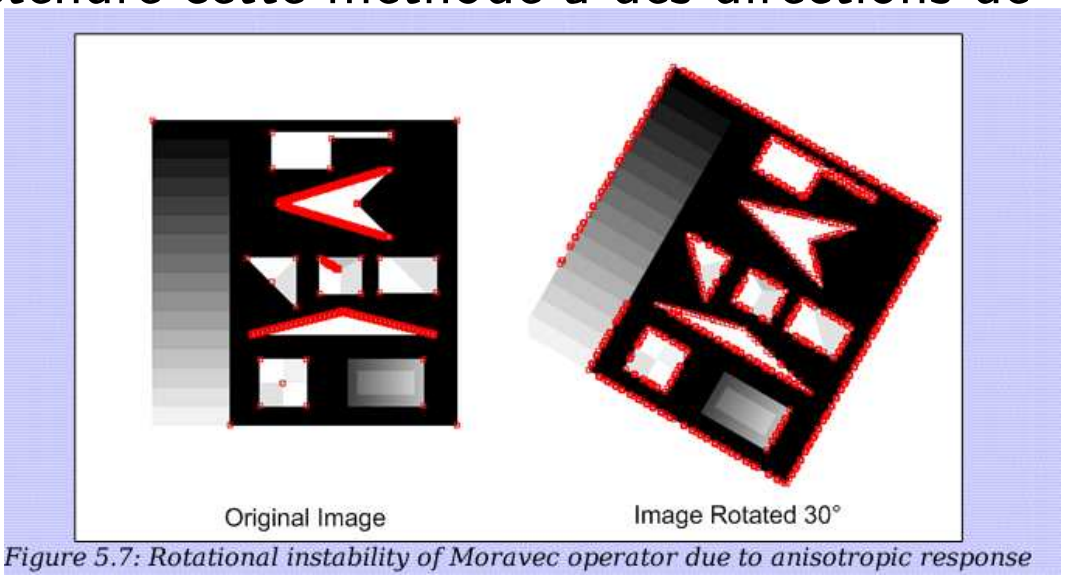


Figure 5.7: Rotational instability of Moravec operator due to anisotropic response

Le détecteur de Harris

Calcul des variations locales d'intensité dans des directions quelconques [Harris88]:

$$\begin{aligned} V & \sum_{fenetre}(I(x+u,y+v)-I(x,y))^2 \\ & \approx \sum_{fenetre}(uI_x+vI_y)^2 \\ & \approx \sum_{fenetre}(u^2I_x^2+2uvI_xI_y+v^2I_y^2) \\ & = (uv)M\begin{pmatrix} u \\ v \end{pmatrix} \end{aligned}$$

avec $M = \begin{pmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{pmatrix}$

Soient λ_1 et λ_2 les valeurs propres de M. Alors V est une ellipse dont les axes ont pour taille $\frac{1}{\sqrt{\lambda_1}}$ et $\frac{1}{\sqrt{\lambda_2}}$.

Le détecteur de Harris

Critere retenu en pratique:

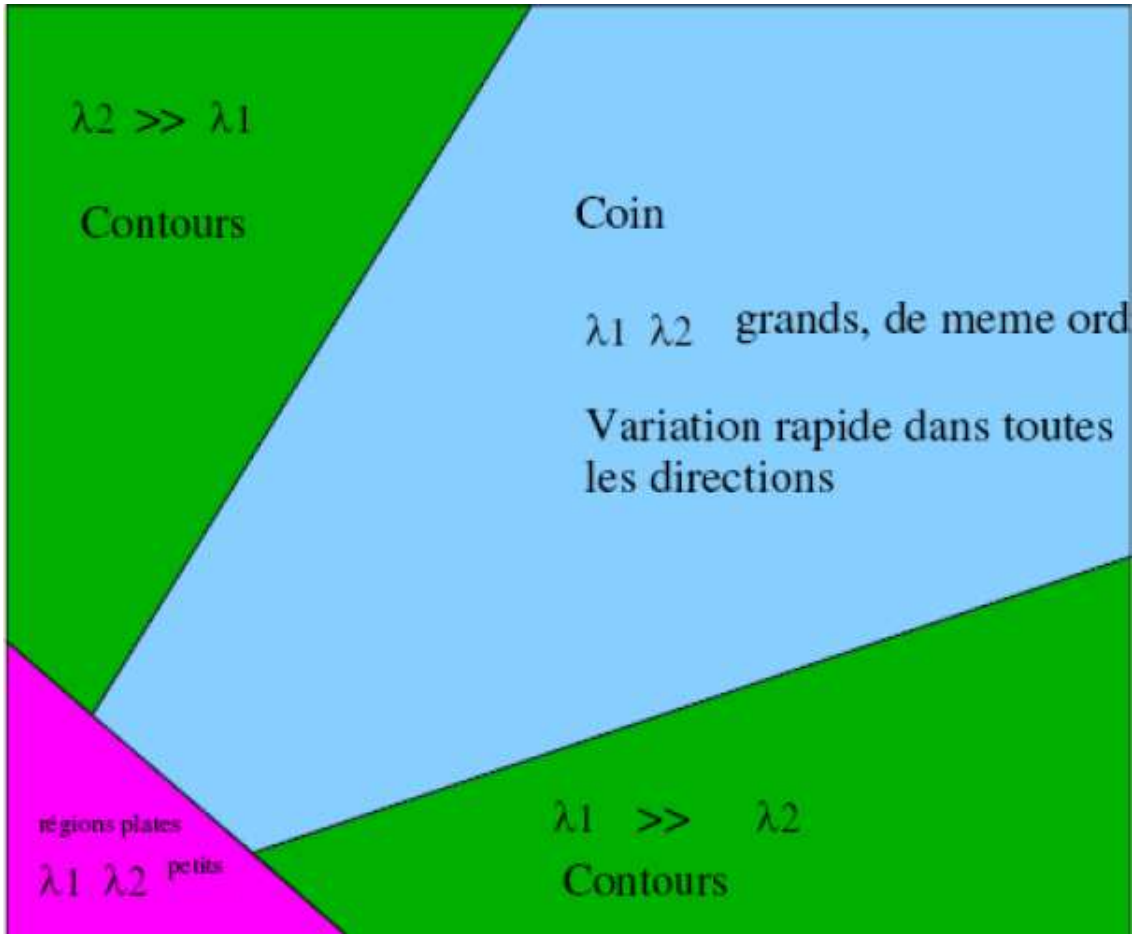
$$C(x,y) = det(M) - k \times trace((M))^2$$

$$det(M) = \lambda_1 * \lambda_2$$

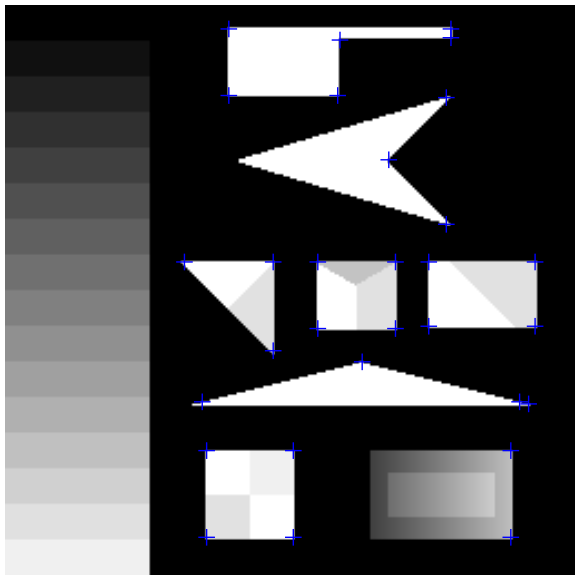
$$trace(M) = \lambda_1 + \lambda_2$$

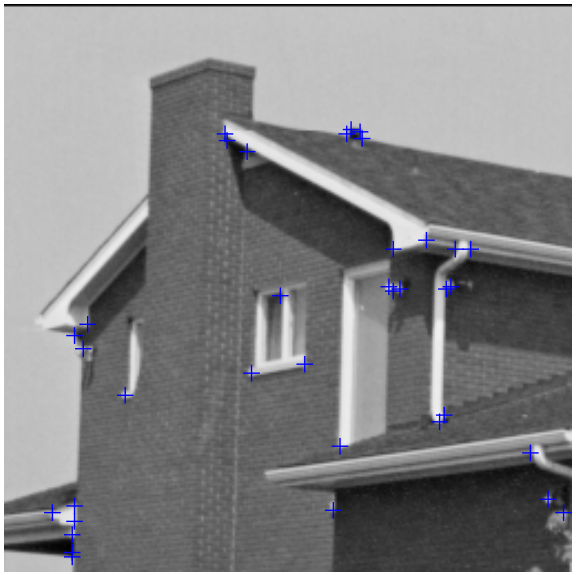
$$k = constante$$

Le détecteur de Harris: classification des points



Le détecteur de Harris: exemples





Le détecteur de Harris: exemples

Mise en correspondance

- Basée sur la ressemblance du voisinage des deux points (corrélation)
- fonctionne à condition qu'il n'y ait pas trop de déformation perspective

Le détecteur SIFT [D. Lowe 99]

- Un détecteur similitude invariant
  - scaling
  - rotation
  - translation
- Capable de mettre en correspondances des points distants avec des variations de caméra assez importantes

SIFT: extraction des points d'intéret

- Considérer $L(x, y, \sigma) = I(x, y) * G_\sigma$
- considérer comme points intéressants les extrema de $L(x, y, k\sigma) - L(x, y, \sigma)$
- Elimination des points à faible contraste ou localisés sur des contours
- Affecter une orientation aux points d'intéret en utilisant le gradient de l'image
- construire un descripteur invariant aux similitudes

SIFT:le descripteur

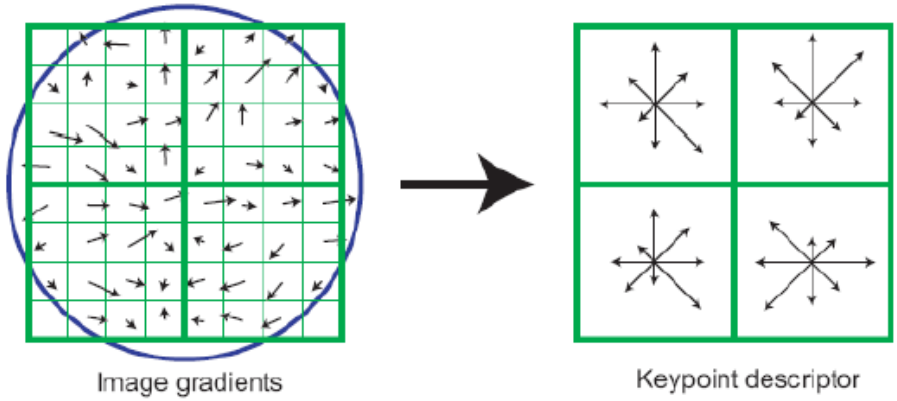


Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

descripteur= histogramme des gradients calculé par rapport à l'orientation du point d'intéret $\rightarrow$ invariance rotationnelle.
En pratique, on utilise 16 histogrammes, soit un descripteur à 128 composantes.

SIFT:résultats

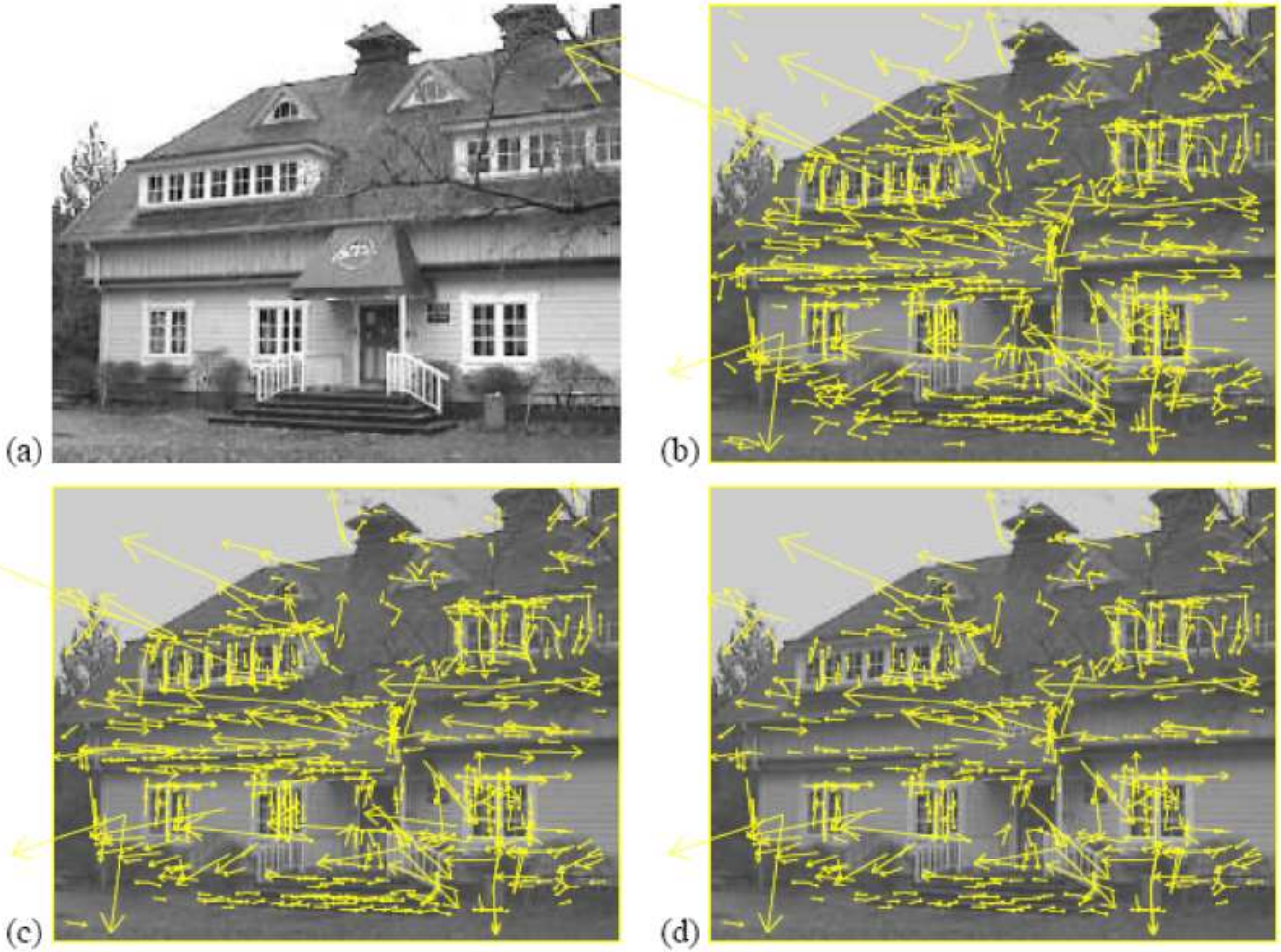


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

SIFT: application à la reconnaissance

Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affine transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

Comparaison des détecteurs de points d'intéret

objectifs: comparer plusieurs détecteurs de points d'intéret (Harris, SURF)

- Récupérer l'archive stereo.zip. Nous allons travailler ici avec les images rdimage.000.ppm et rdimage.001.ppm
- En vous basant sur les exemples de la documentation de *matchFeatures*, ecrivez une suite de commandes permettant d'extraire des points de type Harris dans les images. Afficher les avec *afficherPoints(I, points)*
- Mettez ensuite en en correspondance l'image 1 avec l'image 2. Afficher ces points mis en correspondance.
- Meme question avec le détecteur SURF. Comparer les résultats obtenus avec Harris et SURF.

TP stéréovision

Objectif: Faire une reconstruction stereo par triangulation à partir de points d'intéret en correspondance.

- Télécharger les données stereo sur ma page web (stereo.zip). Le couple d'image stereo est rdimage.000.ppm et rdimage1.000.ppm. Charger la calibration par la commande *load(' calib.mat')*. Afficher le couple d'images. Vous devez voir afficher dans le workspace un fichier stereoParams qui contient la calibration.
- En vous basant sur les exemples de la documentation de *matchFeatures*, ecrivez une suite de commandes permettant d'extraire des points de type SURF dans les images et de les mettre en correspondance. Afficher ces points mis en correspondance.
- Utiliser la fonction visu_epipolaire pour visualiser la ligne épipolaire associée à des points que vous selectionnez sur l'image 1 (cliquer sur des points avec le bouton gauche puis taper sur *enter*. Vérifier que les lignes épipolaires passent par le point correspondant dans l'image 2.

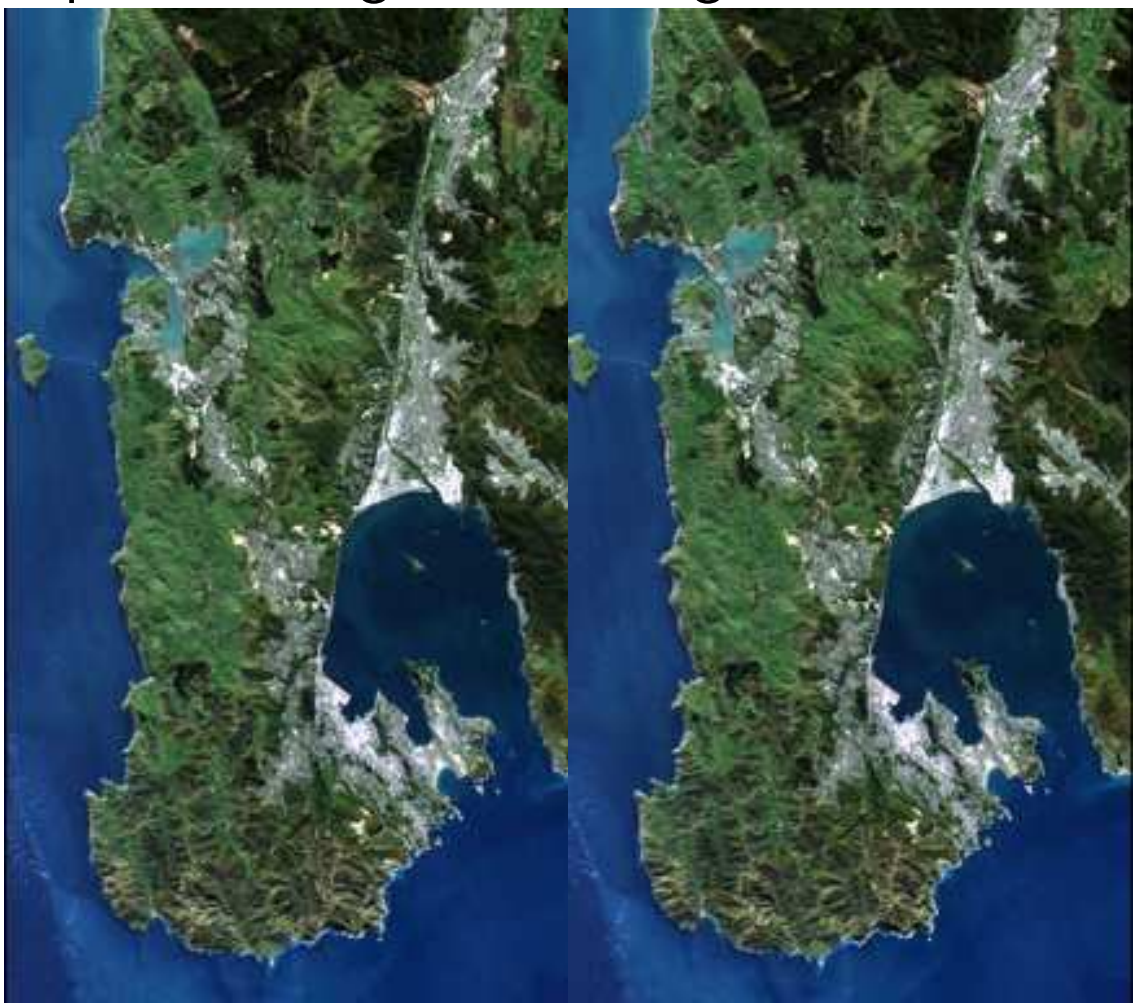
TP stéréovision (suite)

Objectif: Faire une reconstruction stereo par triangulation à partir de points d'intéret en correspondance.

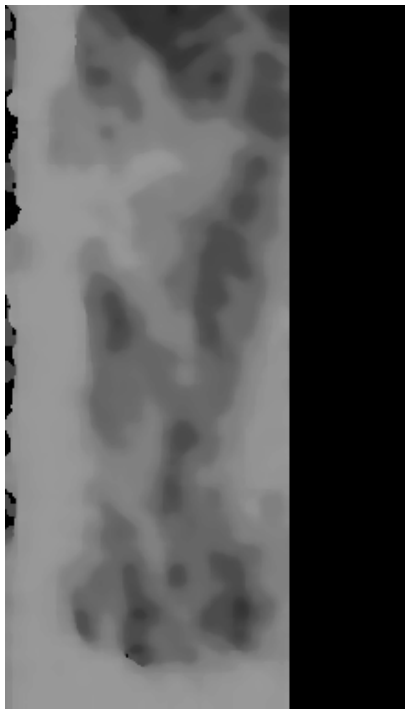
- Nous allons maintenant utiliser la contrainte épipolaire pour ne conserver que les points en correspondance qui satisfont la contrainte. En utilisant la fonction *pointLineDistance* qui mesure la distance d'un point à une droire, sélectionner dans les matchs ceux $(x, x') =$ pour lesquels la distance de x' à la ligne épipolaire Fx est inférieure à 2 pixels.
- Utiliser la fonction *triangulate* pour reconstruire l'ensemble des points 3D correspondant aux points mis en correspondance. Visualiser ce nuage de points avec la fonction *plot3*. Que pensez vous du résultat?
- Il existe sous matlab une fonction *reconstructScene* permettant la reconstruction dense d'une scène. Inspirez vous du code donné dans la documentation de la fonction pour obtenir une reconstruction dense de la scene.

TP stéréovision: reconstruction aérienne

A partir d'images de Wellington



Stereovision dense: reconstruction aérienne



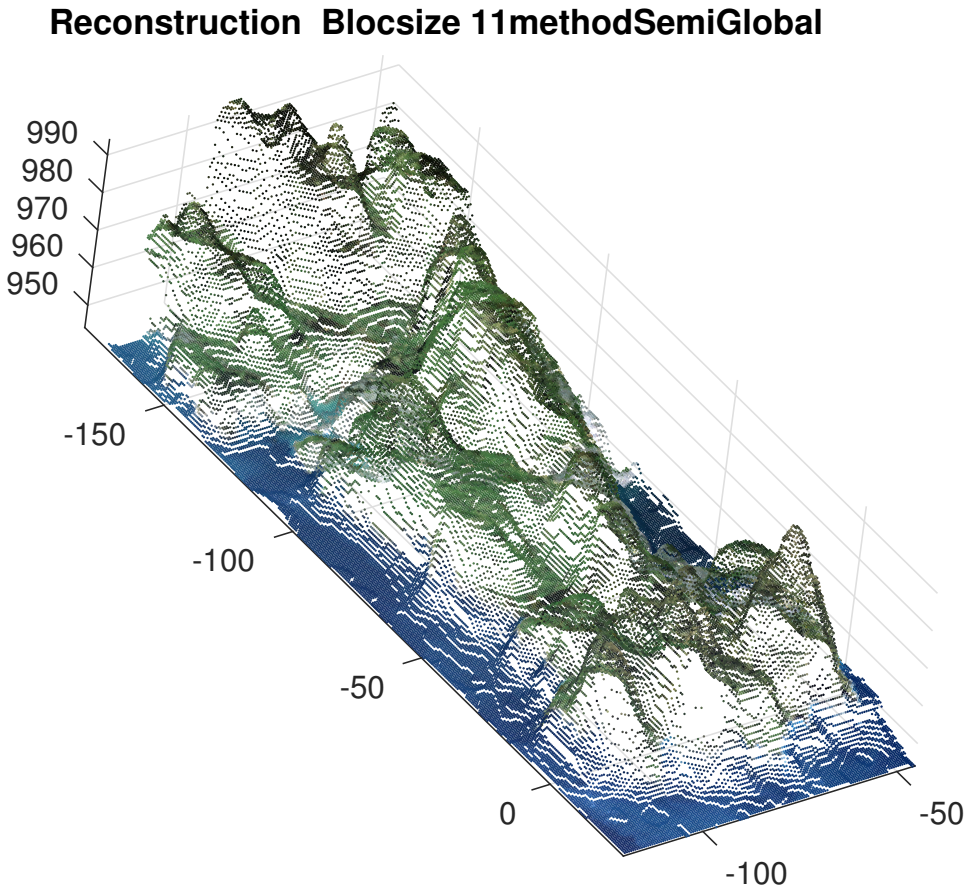


Image de disparité et reconstruction 3D.

TP stéréovision: reconstruction aérienne

- récuper les images wellingtonLeft.png et wellingtonRight.png
- la fonction *reconst_well*() vous fournit une carte de points 3D et visualise ce nuage en couleurs. Si votre machine peine à calculer, utiliser une valeur de pas>1 (pas=1 signifie qu'on calcule les corrdonnées 3D en tout pixel de l'image, pas=2 en un pixel sur 2 etc...
- A partir de ces donnés, déterminer le point culminant de la presqu'ile et afficher ce point sur l'image. A noter que la fonction [u,v]=max (tab) vous donne en u a valeur maximale du tableau et en v l'indice de cette valeur dans le tableau. Le max est ainsi atteint en tab(v)=u.
- Quel est le point de hauteur minimale de l'image? Le localiser sur la carte. en déduire la hauteur maximale de la presqu'ile au dessus du niveau de la mer.