Chapitre 4 Capteurs visuels

Table des matières

Introduction

La vision prend de plus en plus d'importance en robotique mobile. En effet la localisation d'un robot à partir de capteurs de vision peut être considérée comme l'estimation de la trajectoire d'une caméra par rapport à un repère initial. l'odométrie visuelle est une des techniques de localisation par la vision.

- Ces deux étapes s'inscrivent dans le cycle global "perception/action" du processus de vision

- Perception initiale de l'environnement Formation des images/acquisition
- Traitement des images et reconnaissance Repérer les objets de l'aire de travail dans l'image
- Analyse des données Calculer la position des objets et du robot
- Planification et prise de décision Déterminer la prochaine commande à transmettre au robot
- Exécution Transmission de la commande au robot

I.Principe de base de capteur visuel

Le but des capteurs d'images est de retranscrire, le plus fidèlement possible, l'image d'un objet éclairé, ou d'une source lumineuse, formée à leur surface par un système optique adéquat.

Nous nous attacherons principalement à décrire les deux grandes familles de capteurs optico-électoniques utilisés de nos jours : les capteurs **CCD** et les capteurs **CMOS** :

- **CCD** (*«Charge Coupled Device»*),
- **CMOS** (*«Complimentary Metal Oxyde Semiconductor»*),

1. Matrices CCD

Les capteurs CCD transforme les photons lumineux qu'il reçoit en paires électron-trou par effet photoélectrique dans le substrat semi-conducteur, puis collecte les électrons dans le puits de potentiel maintenu au niveau de chaque photosite. Le nombre d'électrons collectés est proportionnel à la quantité de lumière reçue.

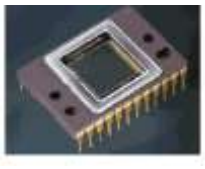


FIGURE 1 CCD

Echantillonnage

La capture d'une image à partir d'un capteur CCD est un échantillonnage de l'image.

Capteur image CCD + filtre Bayer

- Placé sur plan image (en arrière)

- Pixel CCD nu : sensible à toutes les couleurs

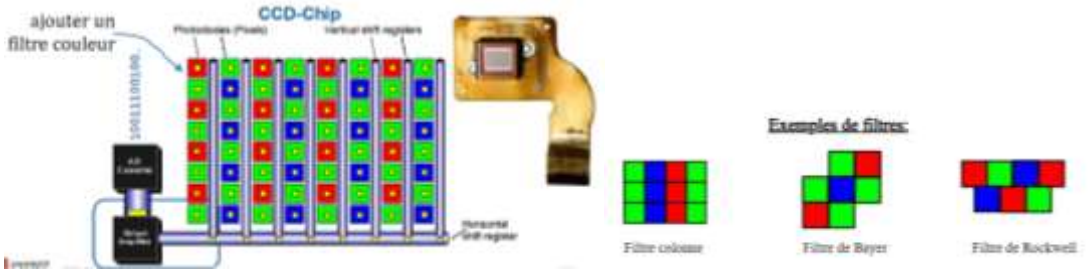


FIGURE 2CAPTEUR IMAGE CCD+FILTRE DE BAYER

2. CMOS

Un capteur CMOS (« complementary metal-oxide-semiconductor ») est composé de photodiodes, à l'instar d'un CCD, où chaque photosite possède son propre convertisseur charge/tension et amplificateur.

Leur consommation électrique, beaucoup plus faible que celle des capteurs CCD, leur vitesse de lecture et le plus faible coût de production sont les principales raisons de leur grande utilisation. De la même façon que beaucoup de CCD, les capteurs CMOS pour image couleur sont associés à un filtre coloré et un réseau de lentilles.

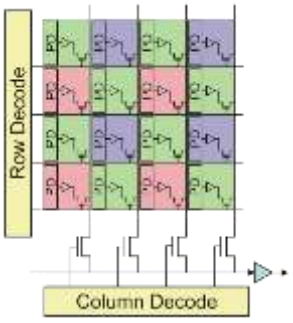


FIGURE 3

II.Modélisation et calibrage d'une caméra

1. Le modèle de la caméra

Le calibrage géométrique d'une caméra consiste à déterminer la relation mathématique existant entre les coordonnées des points 3D de la scène observée et les coordonnées 2D de leur projection dans l'image (points-image). Cette étape de calibrage constitue le point initial pour plusieurs applications de la vision artificielle exemple la reconnaissance et la localisation d'objets, le contrôle dimensionnel de pièces, la reconstruction de l'environnement pour la navigation d'un robot mobile, etc.

FIGURE 4

Calibrer une caméra, c'est choisir un modèle de caméra a priori et déterminer ensuite les paramètres de ce modèle.

1.1. Image processus de capture

• Il ne suffit pas de simplement avoir une surface photosensible :

FIGURE 5

Il faut plutôt bloquer la majorité des rayons :

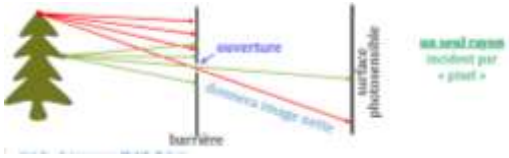


FIGURE 6

1.2. Modèle caméra à sténopé (pinhole)

Une caméra à sténopé est constituée d'une face opaque parallèle à un capteur, dans laquelle est située une petite ouverture faisant le diamètre d'un point, ne laissant passer qu'un seul rayon provenant de chaque point de la scène. Les autres rayons sont arrêtés par la barrière opaque de la caméra. Il n'y a donc qu'un seul rayon qui touche chaque point de l'image. L'image de la scène est inversée sur le capteur.

- **Le sténopé inverseur (à un repère de coordonnées)**

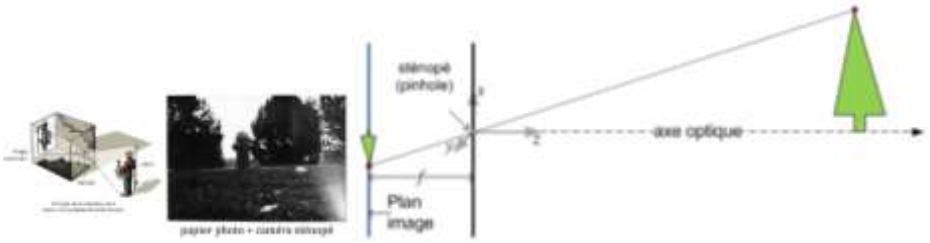


FIGURE 7 FIGURE 8 CAMERA PHYSIQUE IMAGE FORMEE A L'ENVERS

- Déplacer le plan image en avant du trou : image à l'endroit (mais non-réalisable physiquement), plus commode sur le plan mathématique car aucune inversion d'image

- Le sténopé non-inverseur (à un repère de coordonnées)

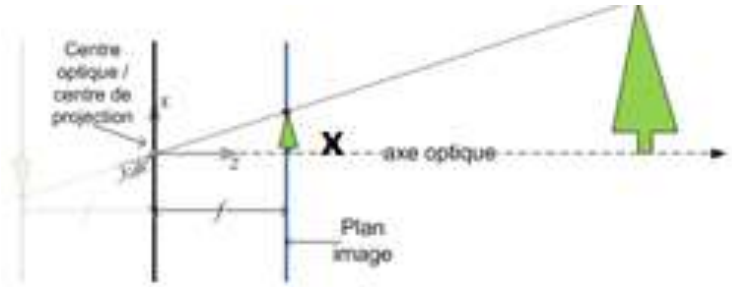


FIGURE 9 (PAR CONVENTION, LES AXES DU PLAN IMAGE SONT NOMMEES X ET Y)

Le modèle sténopé (« pinhole » en anglais) modélise une caméra par une projection perspective.

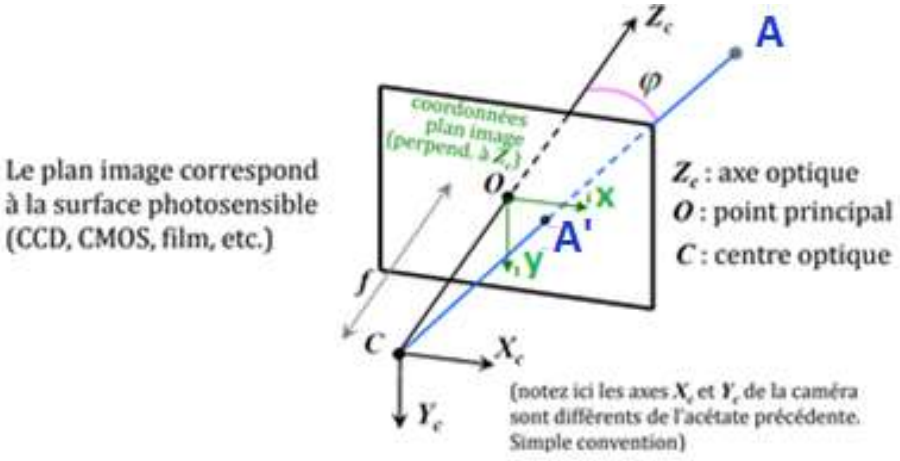


FIGURE 10 NOTEZ ICI LES AXES XC ET YC DE LA CAMERA SONT DIFFERENTS DE L'ACETATE PRECEDENTE. SIMPLE CONVENTION

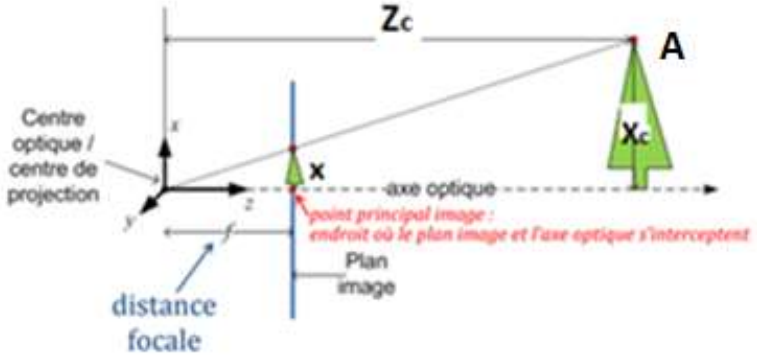
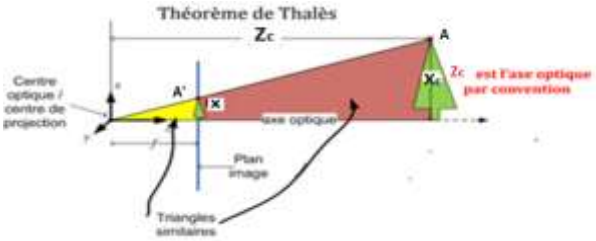


Figure 11



$$\frac{x}{f} = \frac{X_c}{Z_c}, \quad \frac{y}{f} = \frac{Y_c}{Z_c}$$

$$\underbrace{\begin{pmatrix} x \\ y \end{pmatrix}}_{\text{coordonnées plan image}} = \frac{f}{Z_c}\begin{pmatrix} X_c \\ Y_c \end{pmatrix}$$

FIGURE 12

Application : Exemple de projection

Les coordonnées de la caméra, avec centre optique C à $(0,0,0)$ **et axe optique =** Z_c

en supposant que l'origine de (x, y) **est au point principal O**

- Vous avez un point situé aux coordonnées ($X_c = 3, Y_c = 0, Z_c = 20$), en mètres
- La distance focale f de la caméra est de 50 mm
- À quelles coordonnées (x, y) du plan image, en mm, ce point apparaîtra-t-il?

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{f}{Z_c}\begin{pmatrix} X_c \\ Y_c \end{pmatrix} = \frac{50mm}{20m}\begin{pmatrix} 3m \\ 0m \end{pmatrix} = \begin{pmatrix} 7.5 \\ 0 \end{pmatrix} mm$$

2. Point de fuite et ligne d'horizon

Le point de fuite est un point de l'image où toutes les droites parallèles selon une orientation en 3D, convergent. C'est donc l'image d'un point à l'infini: x = (a, b, 0).

FIGURE 13

3. Caméra 2D : perte de 3D

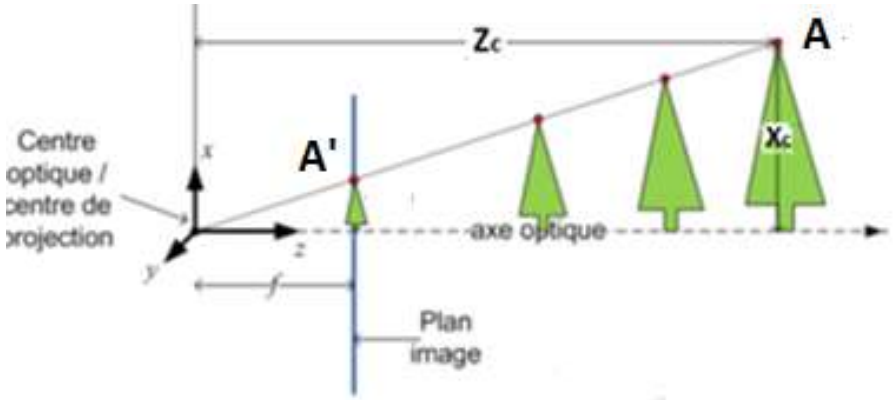


FIGURE 14

On considère la droite passant par A et C comme étant le trajet de la lumière perçue par la caméra. La projection du point A de la scène sur le plan image est A'.

4. Capteur numérique : plan image en pixel

Pour une image numérique, l'origine (u, v) est souvent le coin supérieur gauche. Le point principal O sera situé à la coordonnée (u_0, v_0). Il faudra en tenir compte dans les équations :

$$\boldsymbol{u = u_0 + f\frac{X_c}{Z_c}} \qquad \boldsymbol{v = v_0 + f\frac{Y_c}{Z_c}}$$

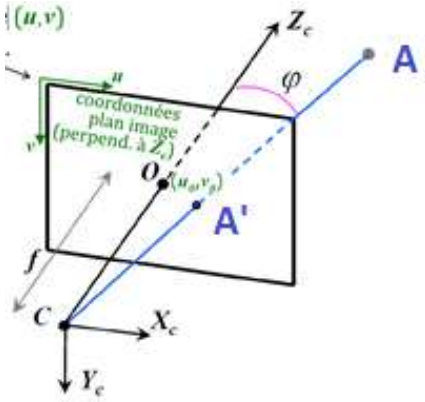
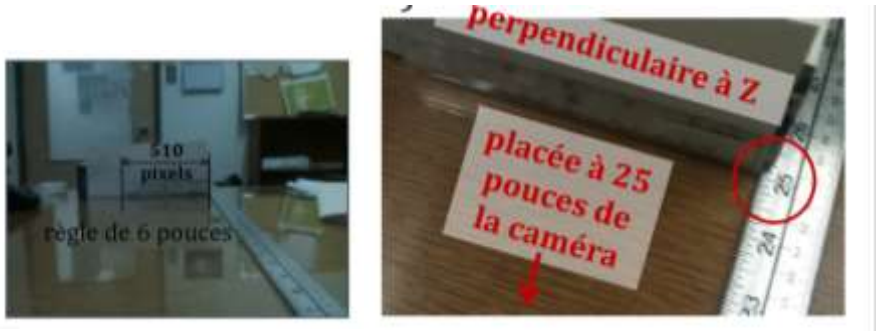


FIGURE 15 (AVEC f DEFINI EN PIXEL)

5. Valeur de la focale f

La valeur de la focale change d'une caméra à l'autre. Elle sera constante (sauf si zoom optique) et on peut l'identifier avec une calibration

• Exemple Calibration rudimentaire :



$$\frac{510pixels}{f} = \frac{6pouces}{25pouces}$$

$f = \frac{25}{6}510pixels = 2125pixels$ Note: on assume ici des pixels carrés sur la cellule

6. Utilisation des coordonnées homogènes et généralisation du modèle de projection

En vision par ordinateur, on utilise souvent les coordonnées homogènes

En 2D

$$A' = \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\text{coordonnées euclidienne}} \rightarrow \tilde{A}' = \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\text{coordonnées homogènes}}$$

En 3D

$$A = \underbrace{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}}_{\text{coordonnées euclidienne}} \rightarrow \tilde{A} = \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\text{coordonnées homogènes}}$$

Parmi les avantage, exprimer le modèle sténopé par une relation linéaire.

homogène ↔cartésien

• Représentation point 2D avec 3 composantes: $\boldsymbol{x} = (x_1, x_2, x_3) \leftrightarrow (x_1/x_3,\ x_2/x_3)$

$(3,2) \rightarrow (3,2,1)$ *ou* $(6,4,2)$ ou… en homogène

•Représentation point 3D avec 4composantes: $\boldsymbol{x} = (x_1, x_2, x_3, x_4) \leftrightarrow (x_1/x_4,\ x_2/x_4,\ x_3/x_4)$

$(2,5,7) \rightarrow (2,5,7,1)$ *ou* $(6,15,21,3)$ ou … en homogène

(Sera très pratique pour rotations et translation, un peu plus tard…)

~ : Signifie identique à un facteur d'échelle

$$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix} \sim \begin{bmatrix}2 & 4\\ 6 & 8\end{bmatrix}$$

7. Généralisation du modèle de projection

L'équation qui traduit la projection perspective s'écrit :

$$\boldsymbol{x = f\frac{X_c}{Z_c}} \qquad \boldsymbol{y = f\frac{Y_c}{Z_c}}$$

Ces équations sont non-linéaires. L'utilisation des coordonnées homogènes permet d'écrire la projection perspective (et le modèle sténopé complet) sous forme linéaire.

L'équation de projection perspective en matricielle s'écrit :

$$\boldsymbol{A' = PA}$$

$\boldsymbol{A'}$: coordonée plan image

$\boldsymbol{A}$: coordonnées d'un point dans l'espace en 3D

$\boldsymbol{P}$: Modèle de la caméra

$$\begin{bmatrix}x\\ y\\ 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & \frac{1}{f} & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{X_c}\\ \boldsymbol{Y_c}\\ \boldsymbol{Z_c}\\ 1\end{bmatrix} \text{ ou } \begin{bmatrix}x\\ y\\ 1\end{bmatrix} = \begin{bmatrix}f & 0 & 0 & 0\\ 0 & f & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{X_c}\\ \boldsymbol{Y_c}\\ \boldsymbol{Z_c}\\ 1\end{bmatrix}$$

$$\begin{matrix}A' & \sim & P & A\end{matrix}$$
$$\begin{bmatrix}x\\ y\\ 1\end{bmatrix} \sim \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & \frac{1}{f} & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{X_c}\\ \boldsymbol{Y_c}\\ \boldsymbol{Z_c}\\ 1\end{bmatrix} \} \textit{coordonnées homogénes}$$

$$\begin{bmatrix}x\\ y\\ 1\end{bmatrix} = \overset{Thalès}{\begin{bmatrix}f\frac{\boldsymbol{X_c}}{\boldsymbol{Z_c}}\\ f\frac{\boldsymbol{Y_c}}{\boldsymbol{Z_c}}\\ 1\end{bmatrix}} \overset{factorise}{=} \frac{f}{\boldsymbol{Z_c}}\begin{bmatrix}\boldsymbol{X_c}\\ \boldsymbol{Y_c}\\ \frac{\boldsymbol{Z_c}}{f}\end{bmatrix} \overset{échelle}{\sim} \begin{bmatrix}\boldsymbol{X_c}\\ \boldsymbol{Y_c}\\ \frac{\boldsymbol{Z_c}}{f}\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & \frac{1}{f} & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{X_c}\\ \boldsymbol{Y_c}\\ \boldsymbol{Z_c}\\ 1\end{bmatrix}$$

- Exemple

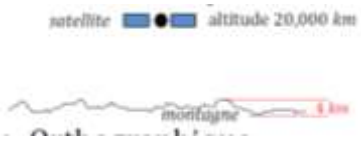
$$\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & \frac{1}{f} & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{X_c}\\ \boldsymbol{Y_c}\\ \boldsymbol{Z_c}\\ 1\end{bmatrix}$$

Point dans l'espace à $[4 \quad 6 \quad 2]^T \rightarrow [4 \quad 6 \quad 2 \quad 1]^T$, Focal $f = 0.1$

$$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 10 & 0\end{bmatrix}\begin{bmatrix}4\\ 6\\ 2\\ 1\end{bmatrix} = \begin{bmatrix}4\\ 6\\ 20\end{bmatrix}$$

Passe d'homogène en cartésien dans plan image $[4 \quad 6 \quad 20]^T \rightarrow [0.2 \quad 0.3]^T$

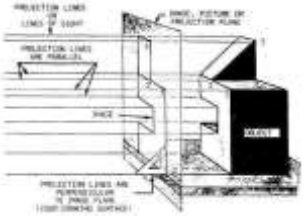
- Perspective faible: distance moyenne $\bar{Z}$



$$A' \sim P \quad A$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{\bar{Z}}{f} \end{bmatrix} \begin{bmatrix} \boldsymbol{X_c} \\ \boldsymbol{Y_c} \\ \boldsymbol{Z_c} \\ 1 \end{bmatrix}$$

- Orthographique : rayons parallèles

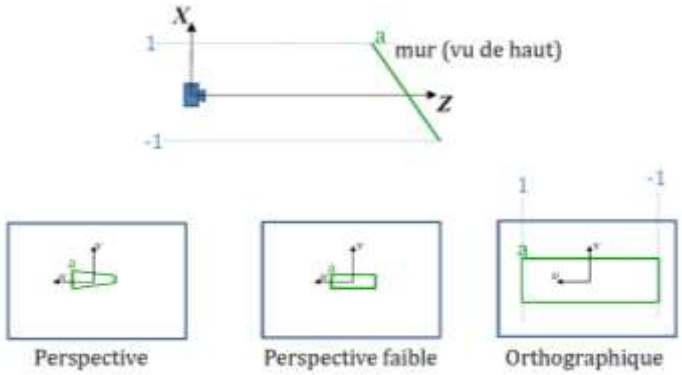


$$A' \sim P \quad A$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X_c} \\ \boldsymbol{Y_c} \\ \boldsymbol{Z_c} \\ 1 \end{bmatrix}$$

distance Z_c n'intervient plus

- Exemple de projection



8. Localisation en 2D par caméra

Pour simplifier le problème, on fait l'hypothèse que le robot est sur un plancher plat, que l'axe optique Z est parallèle au sol, l'axe X de la caméra est parallèle au sol et que tous les points de repère sont à la hauteur de la caméra du robot (sur une ligne horizontale passant par le point principal). On fait l'hypothèse que la carte est donné. avec repères l_i connues

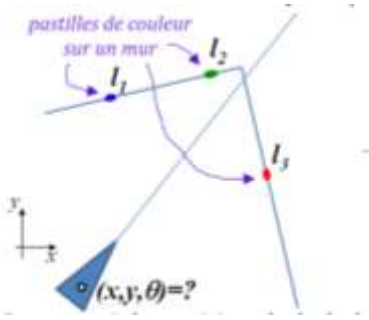


Figure 16

On dispose d'un robot tels que sa pose (x, y, θ) est inconnue. On cherche à retrouver cette pose sachant que l'information qu'on posséde est une photo de l'environnement. A partir de cette photo le robot doit estimer sa position sur la carte. Premier traitement qu'on doit faire est de retrouver les points de repères visuels dans l'image (data association) et leurs associer des étiquettes. On peut donc retrouver x, y et θ à partir des positions en pixels de l_1, l_2, l_3 Pour notre problème, on se fie sur le fait que chacune des poses génère une image différente

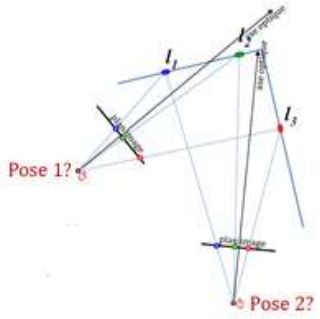


FIGURE 17

- Cela va nous permettre de retrouver la pose de la caméra

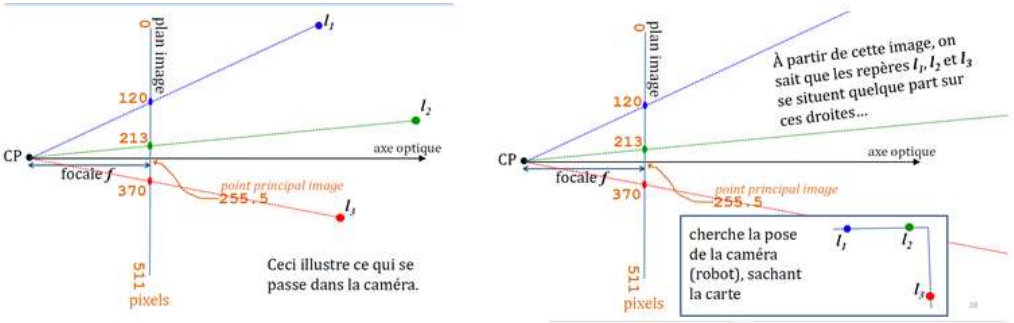


FIGURE 18

8.1. Approches pour trouver la pose d'une caméra

- Méthodes directes
  - Trouver la pose directement à partir des paramètres du problème (image, position des repères, paramètre de la caméra), Souvent basé sur des solutions analytiques Ex : solution géométrique, perspective_n_Point (PnP)
- Méthodes par optimisation
  - Minimiser l'erreur de projection
  - Besoin d'une initialisation relative proche de la réponse

8.2. Méthode direct Le problème Snellius-Pothenot

On va considérer que le robot mobile est un arpentage mobile. Soient 3 points, A B et C, avec coordonnées connues sur une carte et deux angles mesurés α et β L'objectif est de trouver position de P, Commençons tout d'abord par trouver α et β.

FIGURE 19

On va calculer des angles à partir des pixels de l'image

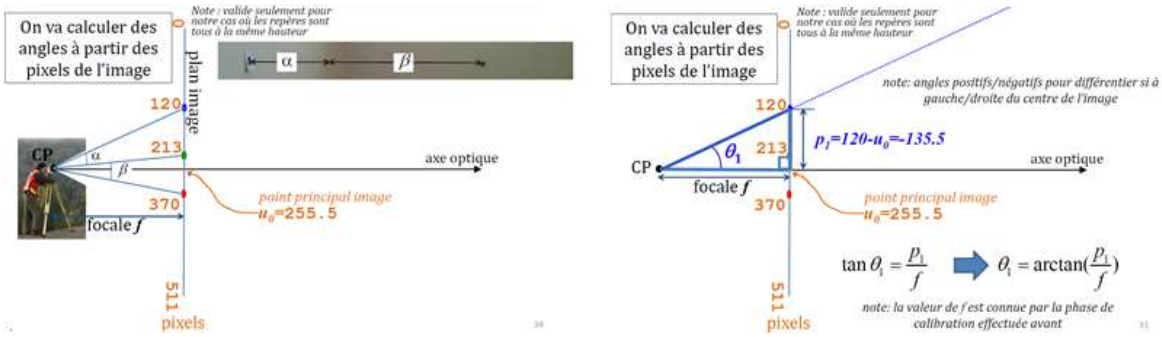


FIGURE 20

Avec cette méthode on peut calculer les 3 angles $\theta_1, \theta_2\ et\ \theta_3$ et par la suite les angles α et β

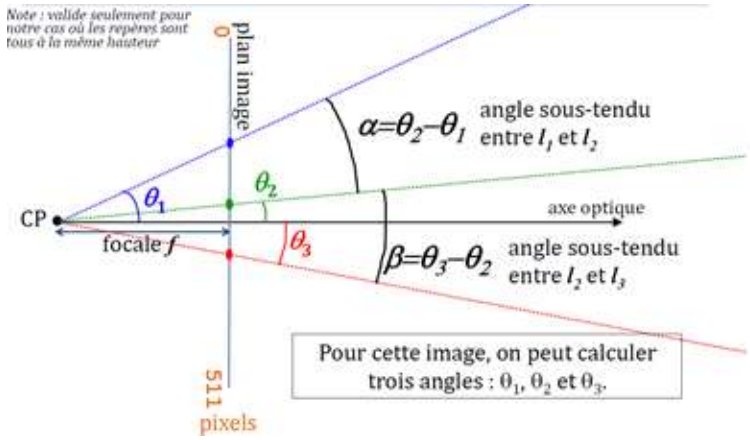


FIGURE 21

La deuxième approche repose sur le produit scalaire

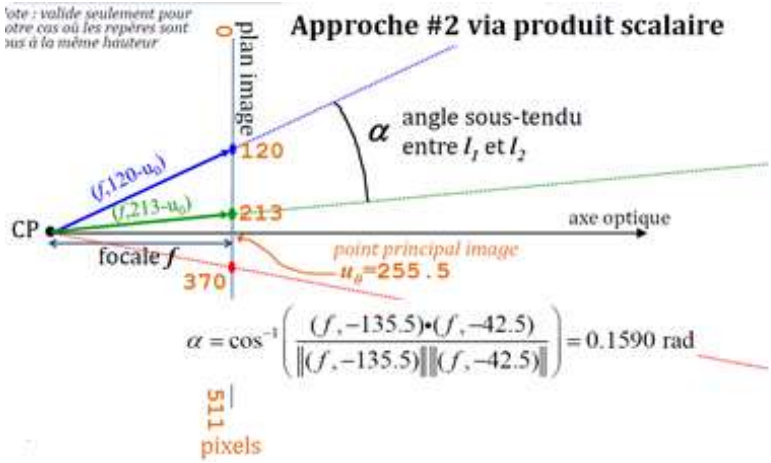


FIGURE 22

A partir des angles α et β on peut retrouver la pose de la caméra à savoir $(\hat{x} \quad \hat{y} \quad \hat{\theta})$

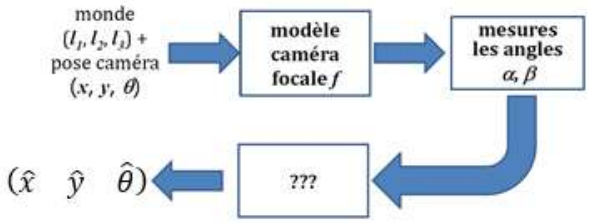
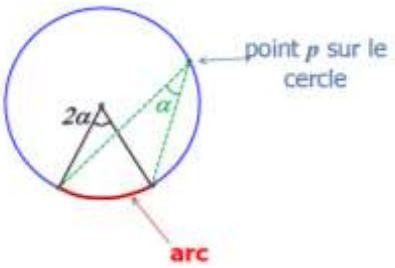


FIGURE 23

Dans la figure 23, on a la carte du monde la position de la caméra qui est inconnue. Lorsqu'on prend la photo d'une caméra avec une focale f sur cette image. A partir de l'image, on mesure les angles α et β. A partir de ces angles et de la carte du monde à savoir les coordonnées des points de repères $(l_1 \quad l_2 \quad l_3)$ on peut estimer la potion de la caméra $(\hat{x} \quad \hat{y} \quad \hat{\theta})$

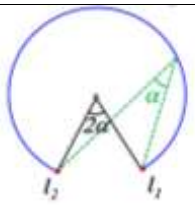
Théorème de l'angle inscrit et de l'angle au centre

Pour un cercle, l'angle au centre mesure le double d'un angle inscrit interceptant le même arc.



Corollaire : cercle unique pour l_1, l_2 et α

- Si on connait la position de l_1, l_2 sur ma carte
- Si on connait α, à partir de l'image



Exemple

La caméra se trouve quelque part sur l'arc de cercle bleu

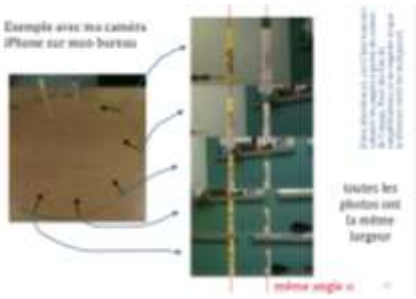
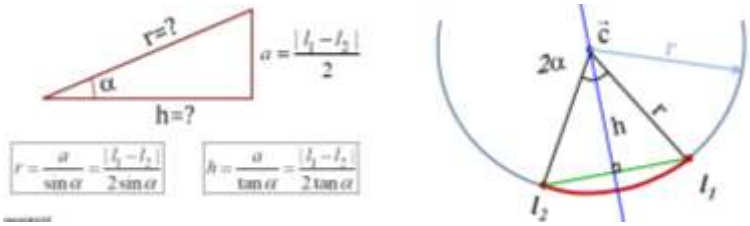


FIGURE 24

<u>Méthode permettant d'identifier le cercle pour l_1, l_2 et α</u>

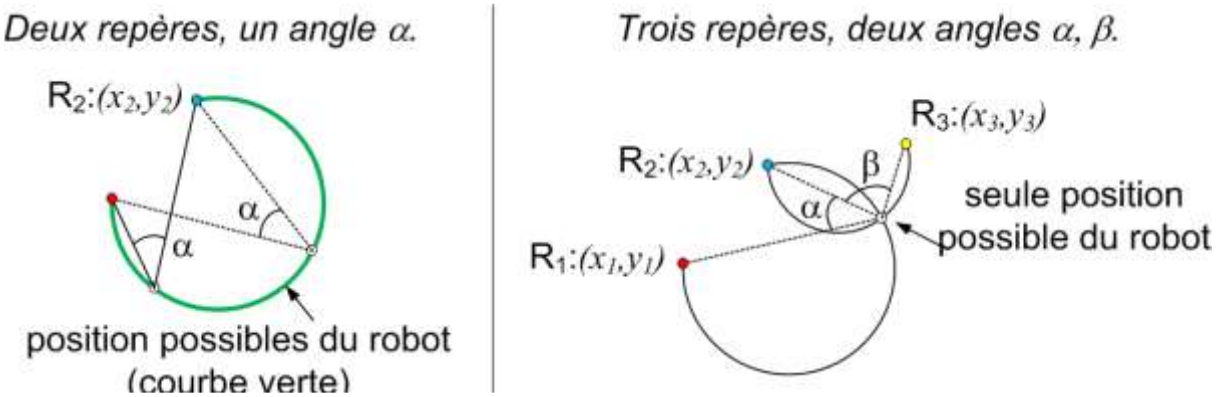
Un cercle se caractérise par la position du centre $\vec{c} = (c_x, c_y)$ et rayon r. $\vec{c}$ est situé sur la médiatrice de la corde l_1, l_2

• Triangle rectangle

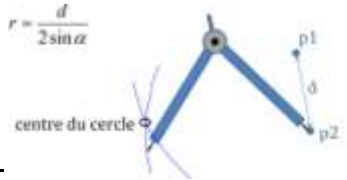


<u>Première méthode Localisation 2D par triangulation (3 angles)</u>

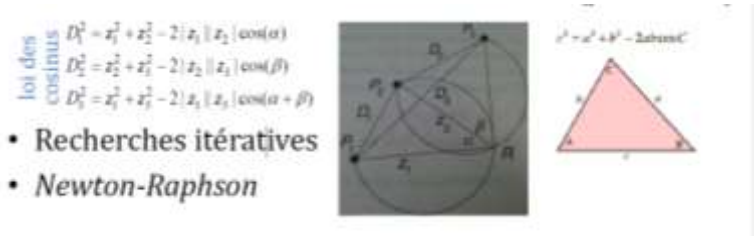
•On dispose des angles α et β entre repères dans l'image de la caméra et on connait les points de repères l_1, l_2 et l_3



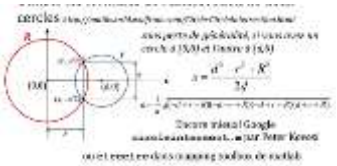
Sur papier Problème localisation, pour trouver centre du cercle $r = \frac{d}{2sin\alpha}$



<u>Autre solution l'inconnue est la position du point de repère *R*</u>

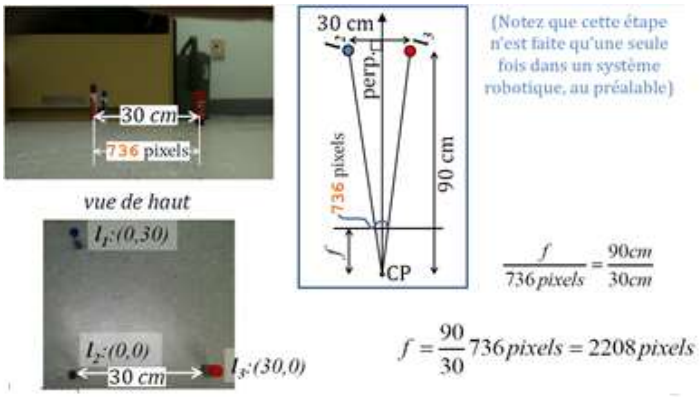


L'approche analytique permettant de trouver le centre et rayon des deux cercles en Utilisant les formules de l'intersection de deux cercles

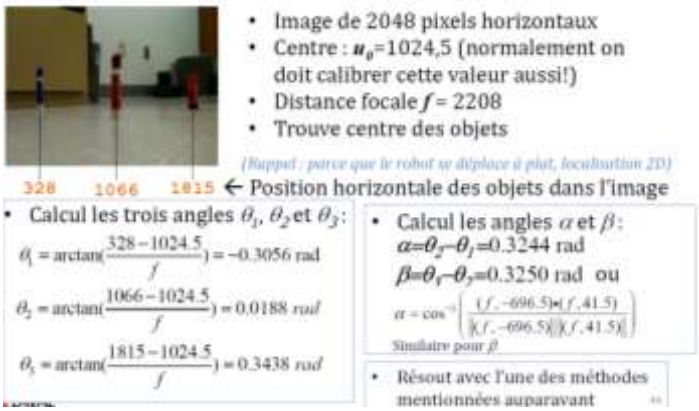


Exemple de localisation

Etape 1 calibration de f

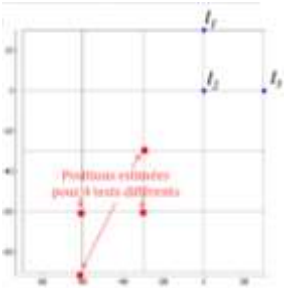


Etape 2 calcul des angles



Etape 3 résultats

- Les positions calculées sont très près des positions réelles (coins des tuiles)
- Mais la précision diminue grandement à mesure que l'on s'éloigne des points de repère ou pour certaines positions



III. Transformations géométriques : rotation et translation

1. Principe

Ce modèle transforme un point 3D de l'espace *A* en un point-image A' et peut se décomposer en trois transformations élémentaires successives (cf. figure 25) :

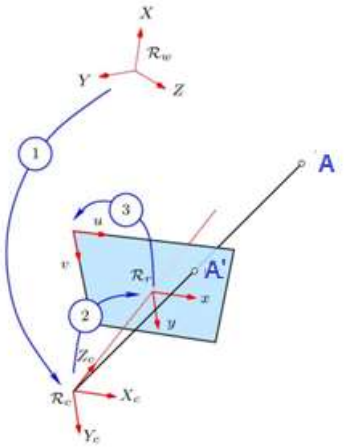


FIGURE 25 LES TROIS TRANSFORMATIONS ELEMENTAIRES DU MODELE STENOPE ET LES REPERES ASSOCIES

La projection du point A de l'espace sur le plan image peut être décomposée en trois transformations élémentaires successives :

- Transformation rigide T, Projection perspective P, Transformation affine M

2. Transformation entre le repère caméra et le repère capteur (plan rétinien)

La deuxième transformation, notée 2 sur la figure 25 relie le repère caméra R_c au repère capteur R_r (plan rétinien). C'est une projection perspective (matrice 3×4, notée P) qui transforme un point 3D $(X_C \quad Y_C \quad Z_C)$ en un point-image A' $(x \quad y)$ (en unité métrique).

L'équation qui traduit la projection perspective s'écrit :

$$\boldsymbol{x = f\frac{X_c}{Z_c} \qquad y = f\frac{Y_c}{Z_c}}$$

Ces équations sont non-linéaires. L'utilisation des coordonnées homogènes permet d'écrire la projection perspective (et le modèle sténopé complet) sous forme linéaire. On obtient la projection perspective qui s'écrit :

$$\boldsymbol{A' = PA}$$

$\boldsymbol{A'}$: coordonée plan image

$\boldsymbol{A}$: coordonnées d'un point dans l'espace en 3D

$\boldsymbol{P}$: Modèle de la caméra

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{X_c} \\ \boldsymbol{Y_c} \\ \boldsymbol{Z_c} \\ 1 \end{bmatrix} \text{ ou } \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{X_c} \\ \boldsymbol{Y_c} \\ \boldsymbol{Z_c} \\ 1 \end{bmatrix}$$

$$A' \sim P A$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{X_c} \\ \boldsymbol{Y_c} \\ \boldsymbol{Z_c} \\ 1 \end{bmatrix} \} \text{coordonnées homogénes}$$

3. Transformation entre le repère du monde et le repère caméra

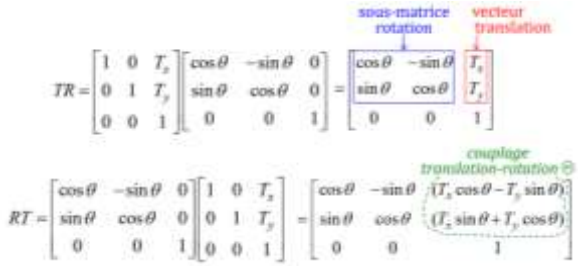
La figure 25, représente une transformation (1) entre le repère du monde R_w et le repère caméra R_C((dont l'origine est située au centre optique de la caméra). Cette transformation rigide peut se décomposer en une rotation $[R]$ et une translation $[t]$. Les paramètres de cette transformation sont appelés paramètres extrinsèques de la caméra.

$$\begin{bmatrix} \boldsymbol{X_c} \\ \boldsymbol{Y_c} \\ \boldsymbol{Z_c} \\ 1 \end{bmatrix} = [R] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} + t = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = [T] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

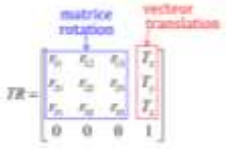
$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \rightarrow T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- <u>Homogène : chaînage des opérations</u>

Plus naturel de faire TR que de faire RT



- <u>Homogène : transformation 3D</u>

rotation autour axe x $R_x(A) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cosA & -sinA & 0 \\ 0 & sinA & cosA & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	rotation autour axe y $R_y(A) = \begin{bmatrix} cosA & 0 & sinA & 0 \\ 0 & 1 & 0 & 0 \\ -sinA & 0 & cosA & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
rotation autour axe z $R_z(A) = \begin{bmatrix} cosA & -sinA & 0 & 0 \\ sinA & cosA & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	<u>Translation</u> $T = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_x \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	

4. Transformation entre le repère capteur et le repère image

La troisième et dernière transformation, notée 3 sur la figure 25, décrit l'opération de conversion des coordonnées images $(x \quad y)$ (en unité métrique) en coordonnées images discrètes $(u \quad v)$ (pixels).

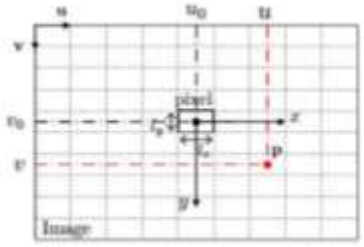


FIGURE 26 LES COORDONNEES PIXELLIQUES

Tels que $u = u_0 + k_x x \; et \; v = v_0 + k_y y$

on pose $k_x = \frac{1}{l_x}$, $k_y = \frac{1}{l_y}$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & k_x cos\theta & u_0 + v_0 cos\theta \\ 0 & k_y / sin\theta & v_0 / sin\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Avec l_x : Largeur d'un pixel et l_y : Hauteur d'un pixel où :

- $u_0 \; et \; v_0$ (en pixels) désignent les coordonnées de l'intersection de l'axe optique avec le plan image (théoriquement au centre de l'image)
- $k_x \; et \; k_y$ désignent le nombre de pixels par unité de longueur suivant les directions x et y du capteur respectivement $k_x = k_y$ dans le cas de pixels carrés)
- θ traduit la non orthogonalité éventuelle des lignes et colonnes de l'image. En pratique, θ est très proche de $\frac{\pi}{2}$. Ce paramètre est désigné par « skew factor » en anglais.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & k_x & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = A_{simplifié} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

5. Modèle sténopé complet

La composition des trois transformations, et peut être résumée par le schéma de la figure 25.

$$(\boldsymbol{X} \quad \boldsymbol{Y} \quad \boldsymbol{Z}) \xrightarrow{\boldsymbol{T}} (\boldsymbol{X_c} \quad \boldsymbol{Y_c} \quad \boldsymbol{Z_c}) \xrightarrow{\boldsymbol{P}} (\boldsymbol{x} \quad \boldsymbol{y}) \xrightarrow{\boldsymbol{M}} (\boldsymbol{u} \quad \boldsymbol{v})$$

Cela conduit à l'équation du modèle sténopé :

point image ***point objet***

$$\boldsymbol{A'} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \boldsymbol{A(R_c)} = \begin{bmatrix} \boldsymbol{X_c} \\ \boldsymbol{Y_c} \\ \boldsymbol{Z_c} \\ 1 \end{bmatrix} \quad \boldsymbol{A(R_w)} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\widetilde{\boldsymbol{A}'} = \underbrace{\boldsymbol{MP}}_{\boldsymbol{K}} \boldsymbol{T}\widetilde{\boldsymbol{A}}(\boldsymbol{R_w})$$

$$\boldsymbol{K} = \boldsymbol{MP} = \begin{bmatrix} k_x & k_x cos\theta & u_0 + v_0 cos\theta \\ 0 & k_y/sin\theta & v_0/sin\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} fk_x & fk_x cos\theta & u_0 + v_0 cos\theta & 0 \\ 0 & fk_y/sin\theta & v_0/sin\theta & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

où $f_x = fk_x$ et $f_y = fk_y$ désignent la focale de la caméra en pixels suivant les directions x et y respectivement.

Les 5 paramètres $(u_0 \quad v_0 \quad f_x \quad f_y \quad \theta)$de la matrice K sont appelés paramètres intrinsèques de la caméra.

Finalement, le modèle sténopé est décrit par 5 paramètres intrinsèques $(u_0 \quad v_0 \quad f_x \quad f_y \quad \theta)$ et des paramètres extrinsèques (pour la rotation et pour la translation).

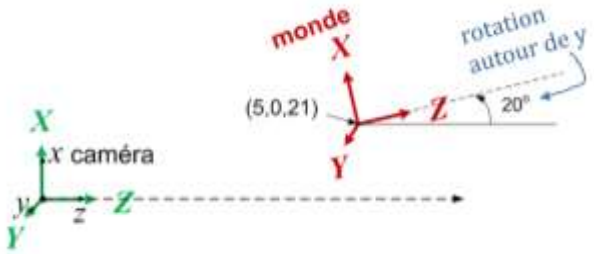
Dans le cas où le « skew factor » est négligé, le modèle sténopé, qui relie les coordonnées 3D $(X \quad Y \quad Z)$ d'un point exprimé dans le repère du monde aux coordonnées 2D $(u \quad v)$ de sa projection dans le plan image (point-image = pixel), est souvent écrit de la façon suivante :

$$K = \begin{bmatrix} fk_x & 0 & u_0 & 0 \\ 0 & fk_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11}X + r_{12}Y + r_{13}Z + t_x \\ r_{21}X + r_{22}Y + r_{23}Z + t_y \\ r_{31}X + r_{32}Y + r_{33}Z + t_z \\ 1 \end{bmatrix}$$

Application : Exemple Transformation monde → caméra

• Pour calculer l'endroit où un point dans le monde va se situer par rapport à la caméra, on trouve la transformation entre les deux référentiels



Transférer le point $m_P = (-3,0,2)$ dans le référentiel du monde, vers le référentiel de la caméra

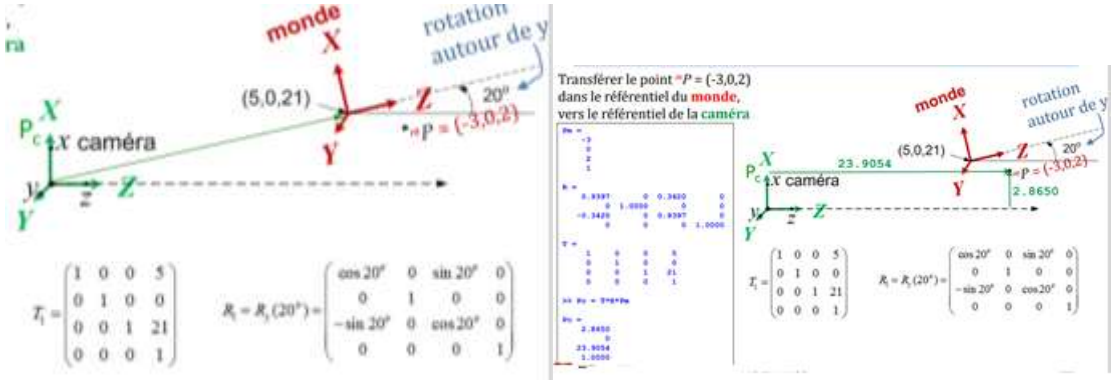


Figure 27

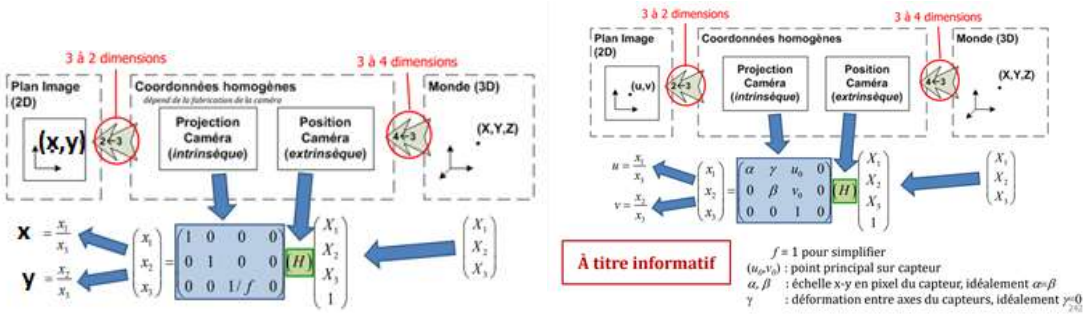


FIGURE 28 MODELE DE CAMERA COMPLET

IV.Modélisation d'un capteur de vision stéréoscopique

1. Principe

Nous nous intéressons à la modélisation d'un capteur composé de deux caméras liées rigidement : un **capteur de vision stéréoscopique**, appelé aussi **capteur de stéréovision**.

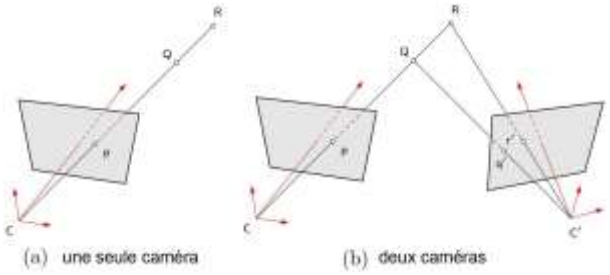


FIGURE 29 RETROUVER LA TROISIEME DIMENSION PAR L'EMPLOI DE DEUX CAMERAS

Si l'on se place d'un point de vue géométrique, une caméra est un capteur qui transforme tout « point visible» de l'espace tridimensionnel en point dans l'espace bidimensionnel de l'image. Cette transformation supprime donc la troisième dimension et est, par conséquent, irréversible. Il est possible de déterminer la position tridimensionnelle du point par **triangulation**. La triangulation consiste donc à déterminer l'intersection dans l'espace des deux droites projectives. Par conséquent, il est nécessaire d'exprimer ces deux droites par rapport à un référentiel commun, par exemple celui de la caméra gauche.

On considère deux caméras placés à une baseline b=5 cm distance, dont les axes sont optiques parallèles. Le baseline est la droite joignant les centres de projection des deux caméras.

FIGURE 30

Le phénomène étudié est la disparité pour estimer la distance par rapport à la caméra

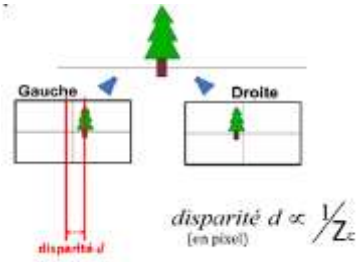


FIGURE 31 A PARTIR DE DEUX IMADE 2D ON RETROUVE 3D

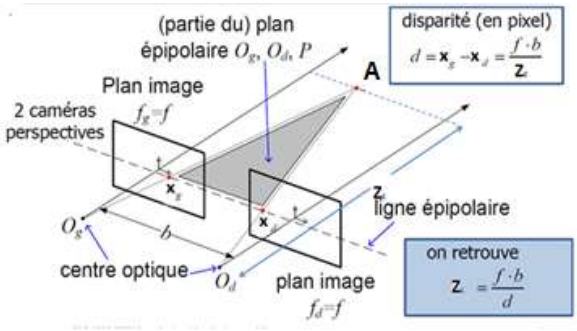


FIGURE 32 CAMERA STEREO

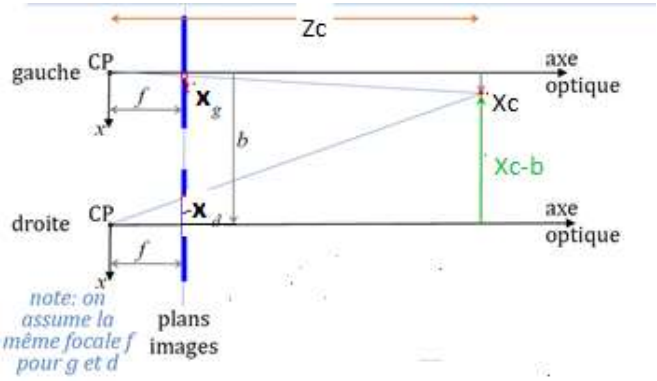


FIGURE 33 FIGURE 25 CAMERA STEREO VUE D'EN HAUT

$$x_g = \frac{f}{Z_c}X_c, x_d = \frac{f}{Z_c}(X_c - b) \rightarrow d = x_g - x_d = \frac{f}{Z_c}b$$

On retrouve donc la profondeur du point $Z_c = \frac{fb}{d}$

- Pour chaque point visible i, on doit :

- faire la correspondance entre les deux images (basé sur l'apparence) (data association)
- estimer la disparité d_i

• On pourra ainsi retrouver la profondeur Z_c pour chaque point i

Cas simpliste, 4 points visibles devant la caméra

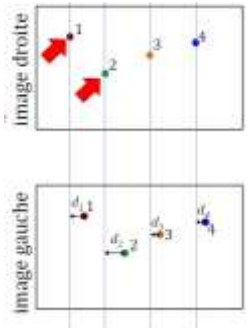


FIGURE 34

2. Définitions

La Stéréovision: La stéréovision consiste à utiliser deux caméra séparées l'une de l'autre d'une certaine distance connue et observant la même scène. Lorsqu'on utilise deux caméras, c'est le système binoculaire, il est suffisant pour reconstruire en 3D la scène observée ou retrouver l'information de la profondeur par triangulation qui peut être utilisée pour l'évitement d'obstacles ou pour la cartographie.

ligne épipolaire :Ligne résultante de l'intersection du plan épipolaire avec les plans image des deux caméras. Dans chaque plan image, il n'y a qu'une seule droite épipolaire par point objet.

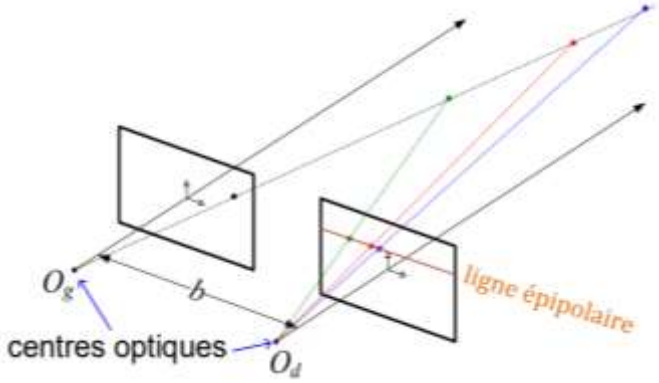


FIGURE 35

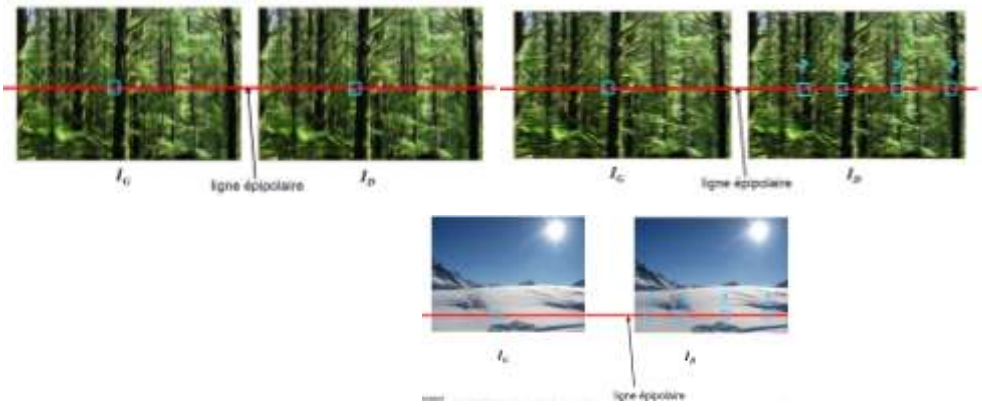


FIGURE 36 PROBLEME DE CORRESPONDANCE

La contrainte épipolaire cette contrainte établit une correspondance entre les points de l'image gauche et les droites de l'image droite et vice versa.

3. Différent type de caméra active 3D

3.1. Kinect 1

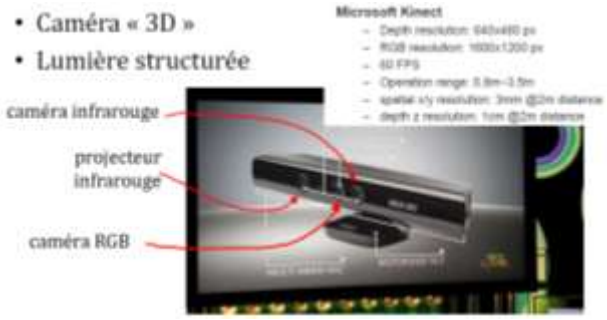


FIGURE 37 CAMÉRA ACIVE KINECT 1 : STÉRÉO ACTIVE

On projette un infrarouge localement distinct nous permettant de retirer plus d'informations sur l'environnement et *Observe* la «disparité» d avec la caméra infrarouge.

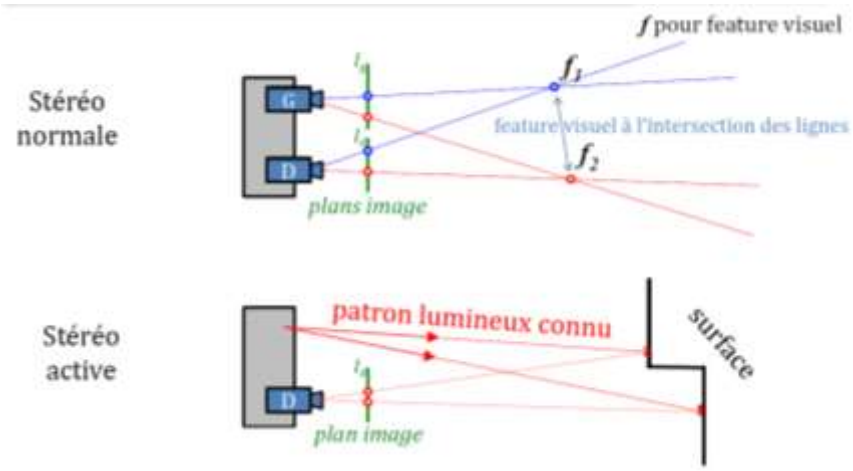


FIGURE 38 KINECT 1 : STEREO ACTIVE

•Retourne une image de profondeur (depthimage)

•Pour chaque pixel, on aura la distance en Z(ou rien)

•640x480 pixels, 30 Hz

•Précision dépend de la distance

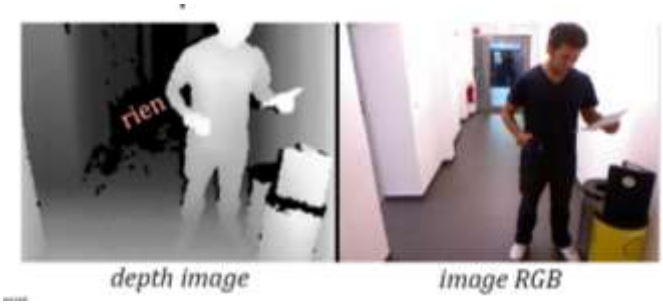


Figure 39 Kinect 1 : depthimage

3.2. Intel Real Sence

D435/D435i (IMU)

–Stéréo normale + active

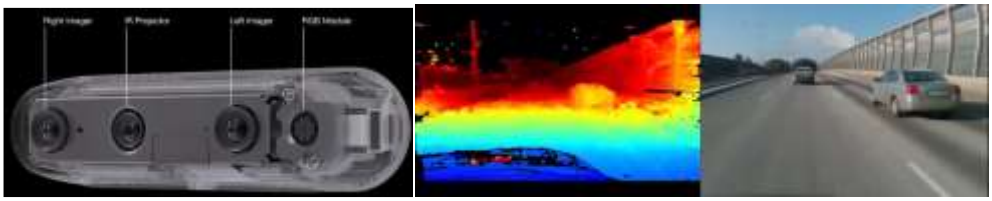


FIGURE 40 D435/D435I (IMU)

4. Autres type d'image : Caméra omnidirectionnelles

Plusieurs techniques ont été développées pour augmenter le champ de vue panoramique et omnidirectionnel que l'on peut classer en trois catégories : l'utilisation de plusieurs images pour former un panorama, L'utilisation d'objectifs grands angles, L'utilisation d'un miroir.

FIGURE 41 CAMERA STANDARD + MIROIR CONVEXE = VUE 360O

FIGURE 42 LADYBUGDE POINTGREY COMBINE 6 CAMERAS ENSEMBLE: VUE 360°

V.Espaces des couleurs

1. Représentation de couleur RGB

- On représente la couleur dans espace en 3 dimensions RGB (Red, Green, Blue)
- Teintes de gris ➔ R=G=B le long de la diagonale

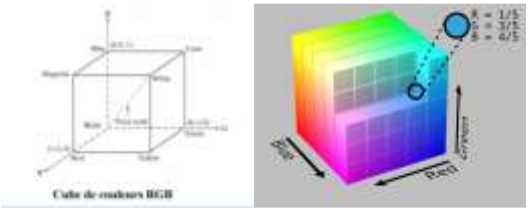


FIGURE 43 RGB

- Problème: L'espace RGB n'est pas intuitif Exemple: On veut baisser la saturation du disque orange de 50% c'est-à-dire l'intensité lumineuse change la couleur va changer.

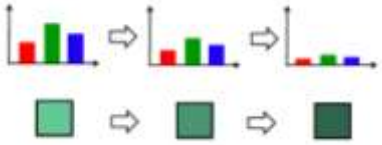


FIGURE 44 MEME TEINTE MAIS PLUS FAIBLE INTENSITE

2. Représentation de couleur HSV

- HSV ➔ Hue (H), Saturation (S), Value (V)
- Séparation de la teinte, de la saturation et de l'intensité
  - Plus intuitif pour identifier et spécifier les couleurs
  - Traitement des ombrages plus facile lors de la segmentation Ombrage ➔même teinte mais intensité différente.
  - encodage plus près de la perception humaine

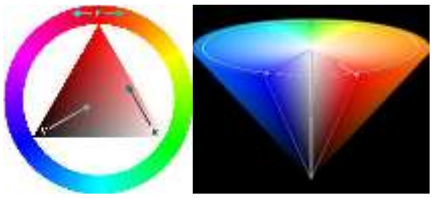


FIGURE 45 HSV

3. HSV vs RGB

Si on compare HSV vs RGB, on voit très bien que lorsque l'intensité lumineuse, les 3 vecteurs de RGB sont affectés alors l'encodage avec HSV, le H et le S ne vont pas varié alors que le V varie car il vient capturer la quantité de lumière incidente sur mon objet.

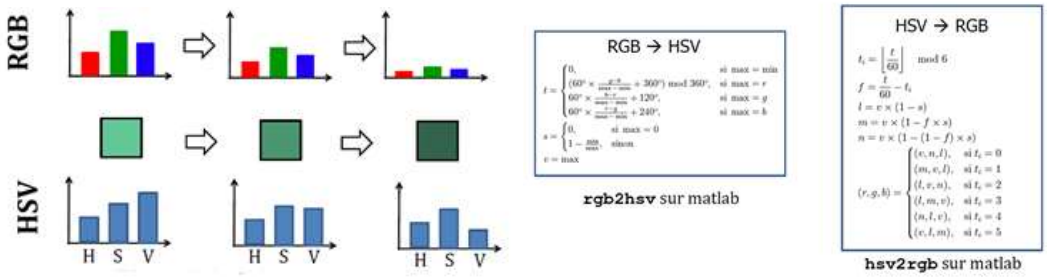


FIGURE 46 RGB VS HSV

Si on veut travailler en caméra on conseille plutôt d'utiliser le HRV

VI. Traitement d'image

Le traitement des images dans un contexte de vision artificielle est utile pour:

- Restaurer le contenu (e.g . atténuer les effets du bruit)

- Rehausser certains éléments dans les images (e.g . Mettre en évidence les contours (discontinuités d'illuminance Compresser le contenu des images en supprimant les informations redondantes (moins important pour le cours de vision)

1. Filtrage

1.1. Filtre pass bas filtre moyenneur

L'hypothèse fondamentale derrière le filtrage linéaire est que la moyenne de plusieurs échantillons devrait réduire le bruit (i.e. l'écart type du signal résultat du moyennage de N échantillons devrait être plus faible que celui de la distribution de laquelle proviennent ceux-ci.

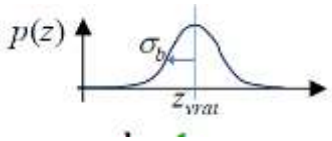


FIGURE 47

• De façon générale, l'écart type pour ***N*** mesures moyennées $\sigma_{moy}^2 = \frac{1}{N}\sigma_b^2 \rightarrow \sigma_{moy} = \frac{1}{\sqrt{N}}\sigma_b$

En utilisant les identités suivantes : $E(nx) = E(x)$

Qui dit qu'en moyennant N valeurs d'illuminance en un pixel, la moyenne demeure la même, mais la variance est réduite de $\frac{1}{N}$, déduisant ainsi l'importance du bruit.

On appelle ce filtre un opérateur de convolution qui prend la forme d'n masque ou noyau ("kernel") de convolution. La convolution est une opération linéaire:

$$commutative : D * E = E * D \qquad Associative : D * (G * E) = (D * G) * E$$

La convolution consiste à balayer l'image avec le masque et tel qu'illustré dans la figure ci-dessous

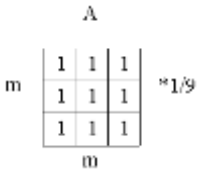


FIGURE 48 EXEMPLE DE MASQUE DE CONVOLUTION: LE FILTRE MOYENNEUR

• Exemple de convolution d'image avec un filtre moyenneur

- Chaque pixel de l'image résultat prend comme valeur la somme des pixels voisins dans le masque.
- L'équation de convolution pour le filtrage linéaire d'une image $I(x, y)$ avec un filtre de noyau $A(h, k)$ est la suivante pour chaque pixel d'illuminance $I(\ i, j)$

$$I_{filtre} = A(h,k) * I(i,j) = \frac{1}{m^2} \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} A(h,k)\,.I(i-h, j-k)$$

- Pour chaque pixel (i,j), les valeurs d'illuminance $I(\ i,j\ )$ des pixels couverts par le masque $A(h,k)$ sont multipliées par les valeurs du masque et additionnées pour produire la moyenne en multipliant par la taille du masque $(1/m^2)$. Il est plus pratique d'avoir des masques m impair
- - plus m augmente, plus le moyennage s'effectue sur une grande région autour du pixel $(\ i,j\ )$ et plus le filtrage est important.

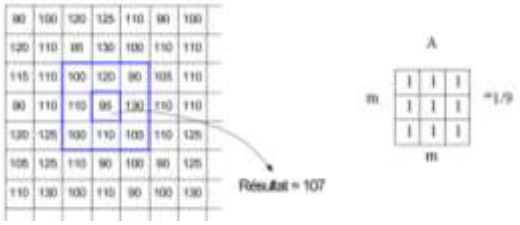


FIGURE 49

$$\frac{1}{5}\begin{bmatrix}0 & 1 & 0\\ 1 & 1 & 1\\ 0 & 1 & 0\end{bmatrix} \quad \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$carré$ $octogonal$

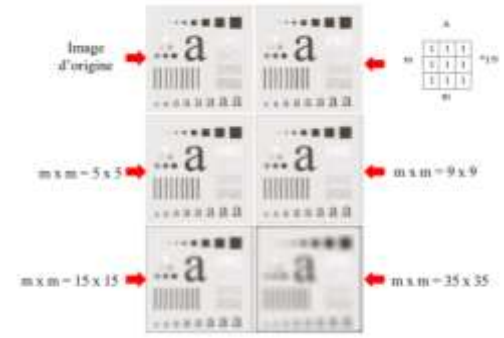


FIGURE 50

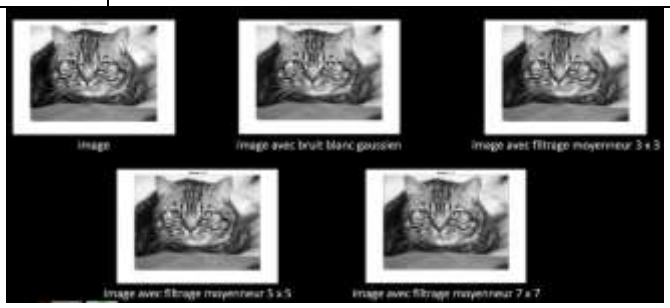


FIGURE 51 ON CONSTATE QUE SI LE BRUIT EST REDUIT, L'IMAGE DEVIENT DE PLUS EN PLUS FLOUE QUAND m AUGMENTE

1.2. Filtrage linéaire pass-bas filtre gaussien

- Le filtre gaussien est un type de filtre qui utilise une fonction gaussienne pour calculer la transformation à appliquer à chaque pixel de l'image. La formule d'une fonction gausienne à deux dimensions est $\mathrm{G}(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$

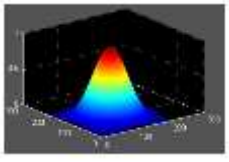


FIGURE 52

Ou x est la distance à l'origine sur l'axe horizontal, y est la distance à l'origine sur l'axe vertical et σ est l'écart type de la distribution gausienne

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

Masque filtre gaussien 3x3 $\sigma = 0,8$

1.3. Filtrage non linéaire

Le résultat n'est pas une combinaison linéaire des pixels de l'image à traiter mais plutôt une fonction non-linéaire telle qu'un min, un max ou la médiane. Le filtre étudié est le filtre médian : Le filtre médian est bien adapté au filtrage du bruit impulsionnel. Le filtre médian utilise aussi un noyau sur lequel on effectue les opérations suivantes

- trier les valeurs d'illuminance des pixels couverts par le masque
- extraire la médiane des données triées
- remplacer la valeur du pixel central par la médiane

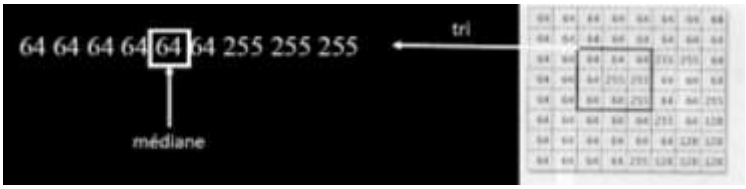


FIGURE 53

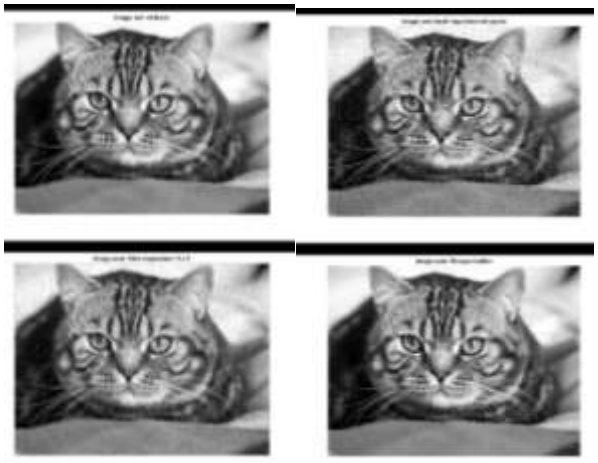


FIGURE 54

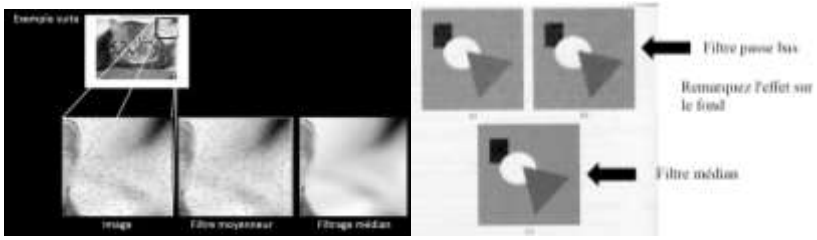


FIGURE 55

2. Segmentation : détection d'un bord (edge detection)

2.1. principe

Les bords possèdent beaucoup «d'information»

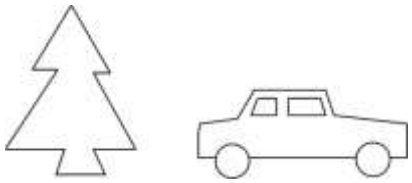


FIGURE 56

Certains bords sont plus informatifs que d'autres…

FIGURE 57

La première image est plus informative que la deuxième. La différence entre les deux est que la première présente des coins et la deuxième des lignes. Les coins sont plus significatifs. Nous allons d'abord définir des bordures car un coin est une jonction de bordures.

Définition d'une bordure : variation spatiale rapide d'intensité lumineuse dans l'image I

$$grand\ gradient\ \nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$$

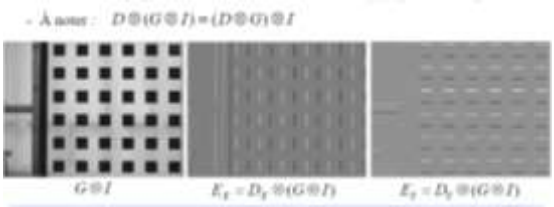
Le gradient est une quantité vectorielle ayant une amplitude et une orientation

Le gradient est difficile à extraire *parfaitement*

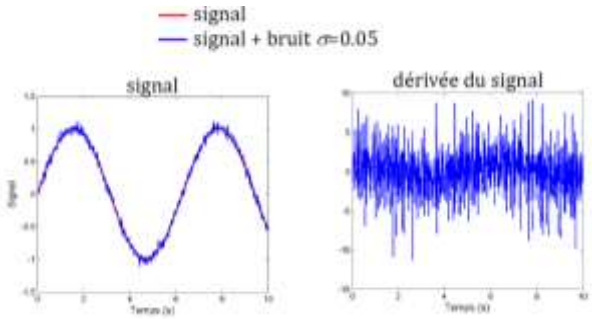
- Notamment à cause du bruit dans l'image
- En effet l'opération de dérivée **AUGMENTE** l'impact du bruit

Pour limiter l'effet du bruit, un lissage est compris dans le calcul (filtre moyenne pour Prewitt, et filtre gaussien pour Sobel)

La détection des ***arêtes par*** dérivée première :

Opérateur de ***Prewitt*** 	Gradient horizontal : $D_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ Gradient vertical : $D_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Opérateur de ***Sobel***	$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

Prenons un exemple d'un Signal corrompu avec bruit gaussien $N(0,\sigma^2)$. On peut remarquer quand on dérive la partie dominante est le bruit.



Comme la dérivée est sensible au bruit, il est recommandé de passer par un filtre gaussien.

On constate que si le bruit est réduit, l'image devient de plus en plus floue quand m augmente

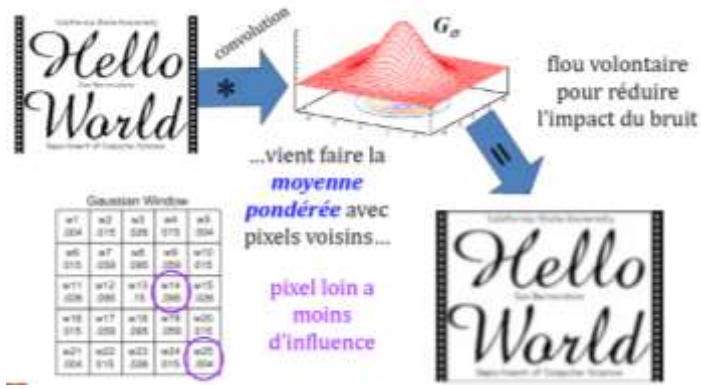


FIGURE 58

Exemple : Sobel edge detector avec le bruit

Pour bien voir l'importance d'appliquer un filtre, on va chercher les bordures en calculant des gradients dans l'image. On a une image 7×3 pixels reproduisant une intensité lumineuse (claire/foncée). Deux opérateurs pour calculer le gradient Δ_1 et Δ_2 sont utilisées. Puis un produit de convolution est appliqué. On sait, en présence d'une grande valeur absolue du gradient on a une présence de bordure. Le signe va indiquer la direction de la bordure.

$$\text{approxime le gradient}$$

$$\Delta_1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \Delta_2 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad sobel = \sqrt{\Delta_1^2 + \Delta_2^2}$$

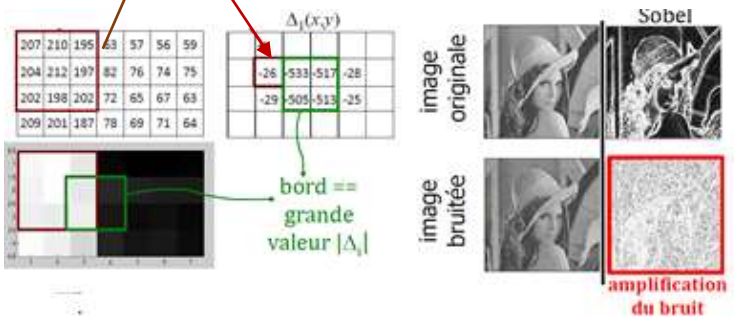


FIGURE 59

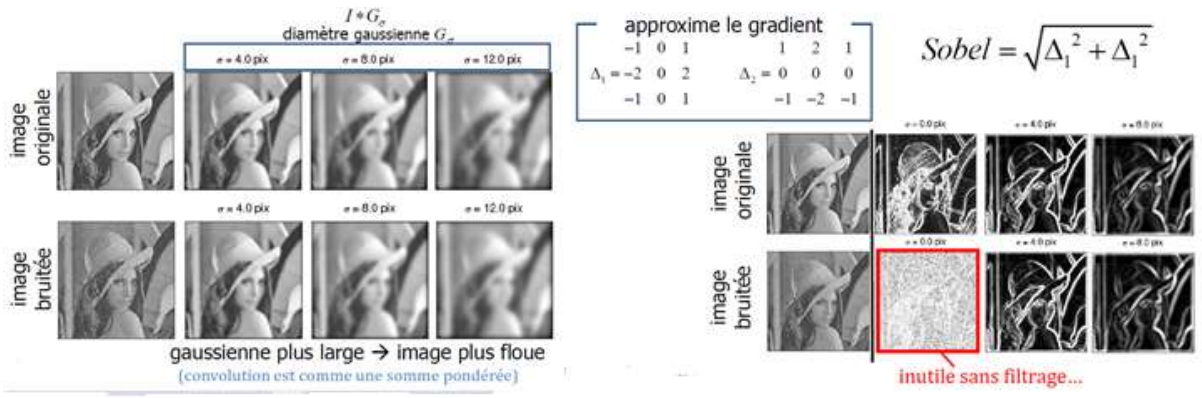


FIGURE 60

2.2. Extraction de contours (Filtre de Canny)

Etape 1 : Appliquer un filtre gaussien pour lisser l'image afin de supprimer le bruit (on applique Grascale $Y = 0.299R + 0.587G + 0.114B$ si l'image est de couleur)

Etape 2 : trouver les gradients de l'intensité de l'image (Sobel ou Prewitt, horizontal et vertical): Calculer les gradients horizontal et vertical ($I_X$et I_Y). Calculer les images : $I_s = \sqrt{I_X^2 + I_Y^2}$ $I_\theta = \tan^{-1}\frac{I_Y}{I_X}$

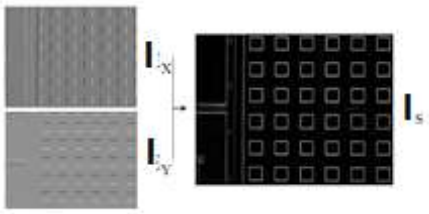


FIGURE 61

Etape 3 : Éliminer les réponses non-maximales Un pixel a une réponse maximale si ses deux voisins, situés dans l'axe de sa normale, ont une réponse inférieure. Un pixel non-max est mis à zéro. Un maximum local est présent sur les extrema du gradient, c'est-à-dire là où sa dérivée selon les lignes de champs du gradient s'annule.

FIGURE 62

Etape 4 : Seuillage par hystérésis (2 seuils) Hysteresis thresholding

- Inférieur au seuil bas, le point est rejeté ;
- Supérieur au seuil haut, le point est accepté comme formant un contour ;
- Entre le seuil bas et le seuil haut, le point est accepté s'il est connecté à un point déjà accepté.

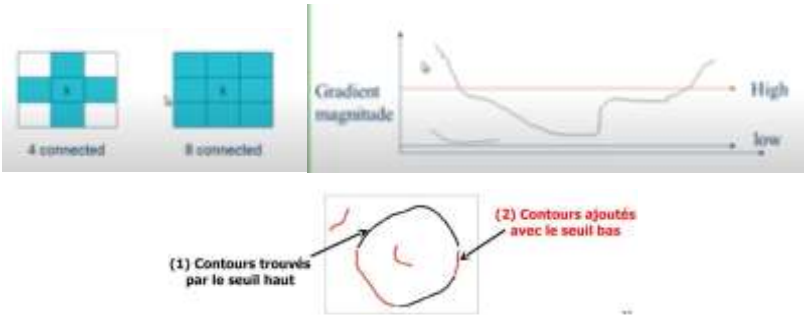


FIGURE 63

Etape 5 : Sortie du détecteur de contours de Canny: ➔ Image de contours (binaire)

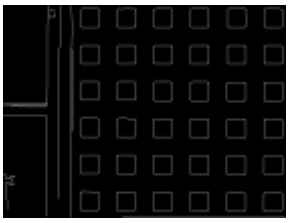


FIGURE 64

Exemple

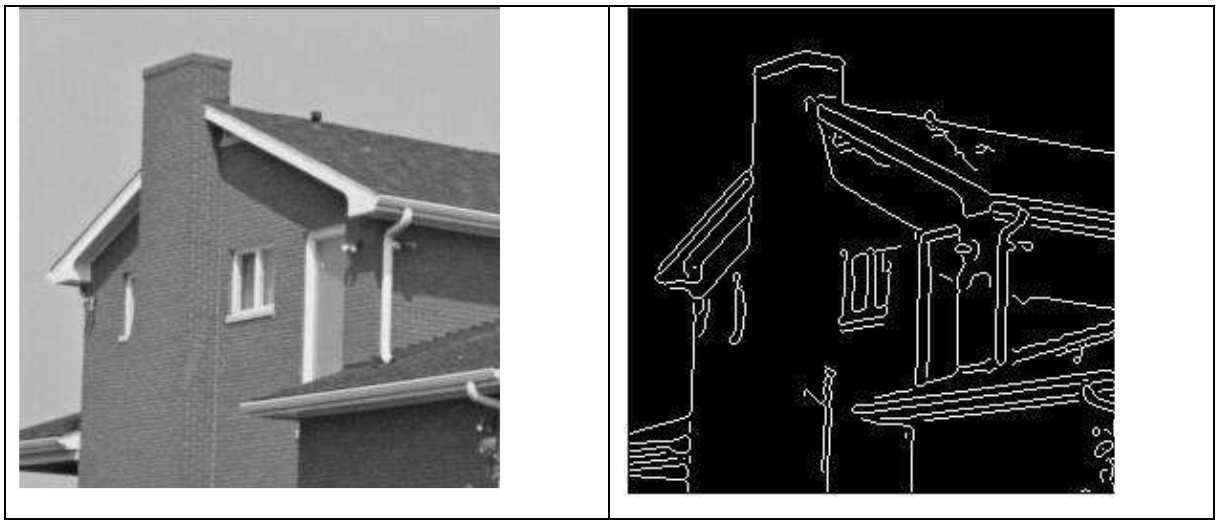


Figure 65

Algorithme de détection de contours par *thinning*

1: Convoluer l'image d'entrée à l'aide d'un filtre gaussien

$$g = G_\sigma * f$$

2: Calculer la magnitude du gradient en chaque point de l'image:

$$|\nabla g(x,y)| = \sqrt{\left(\frac{\partial g(x,y)}{\partial x}\right)^2 + \left(\frac{\partial g(x,y)}{\partial y}\right)^2}$$

3: Calculer l'orientation de la normale du gradient en chaque point de l'image

$$\theta(x,y) = \arctan\left(\frac{\partial g(x,y)}{\partial y} \Big/ \frac{\partial g(x,y)}{\partial x}\right)$$

4: À l'aide de $|\nabla g(x,y)|$ et $\theta(x,y)$ supprimer les non-maximums

$$h(x,y) = \text{NonMaxSupp}(|\nabla g(x,y)|, \theta(x,y))$$

5: Appliquer un seuil

$$c(x,y) = \begin{cases} 1 & \text{si } h(x,y) > \text{seuil} \\ 0 & \text{sinon} \end{cases}$$

FIGURE 66

VII.Repère visuelle naturels

1. Repères naturels

- Pas tous les endroits dans une image sont utiles pour la localisation
- Détecter des endroits «saillants» dans l'image, ex. coins : **keypoint**
- Calculer une signature visuelle autour de ce point (**descripteur**)
- *keypoint*+ descripteur = *feature*

2. Détecteurs de coins (keypoint detectors)

2.1. Propriétés d'un keypoint

- Diffère de ses voisins en terme de texture, couleur et/ou intensité (distinctiveness)
- Détection répétable (repeatability) d'une image à l'autre malgré des changements
  - de points de vue (rotation/translation/affine1) **ou** d'illumination
- Localisé (occupe espace restreint) • Rapide à calculer (on parcourt toute l'image)

2.2. Détection des coins (keypoint)

Constat: les détecteurs d'arêtes se comportent moins bien près des coins

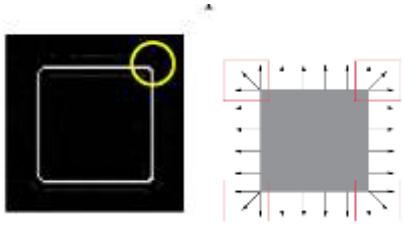


FIGURE 67

- Proposition: détecter les endroits où le gradient est fort dans plus d'une direction.
- Réalisation: détecteur de Moravec/Harris / Fast

a. Exemple : *Moravec interest operator*

• Utilise Sum-of-Squared Difference (SSD) comme mesure de similarité entre deux patches I_a et I_b :

$$SSD = \sum_{i,j \in patch} \left(I_a(i,j) - I_b(i,j)\right)^2$$

Cherche un endroit où le SSD par rapport aux patches voisines est localement maximal :

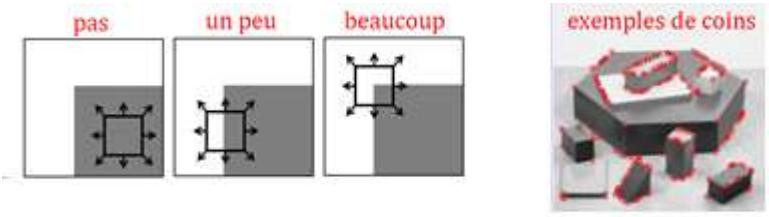


FIGURE 68

b. Exemple : détecteur de Harris

- Etape 1 Calculer les gradients horizontal et vertical ($I_x(i,j)$ et $I_y(i,j)$)

Approximation par Taylor : gradients $I_x(i,j) = \frac{\partial}{\partial x} I(i,j)$, $I_y(i,j) = \frac{\partial}{\partial y} I(i,j)$

- Etape 2 pour tous les points de l'image de gradient, calculer la matrice de covariance du gradient et en extraire les valeurs propres λ_1 et λ_2 ($\lambda_2 \leq \lambda_1$) Pour visualiser cette étape, on peut prendre un pixel (u,v) et un voisinage $(\Delta x, \Delta y)$. Si on approxime $I(u+\Delta x, v+\Delta y)$ par sa série de Taylor, en se limitant aux 2 premiers termes, on obtient:

$$I(u+\Delta x, v+\Delta y) = I(u,v) + I_x(u,v)\Delta x + I_y(u,v)\Delta y$$

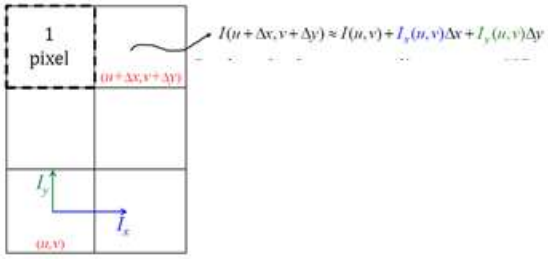


FIGURE 69

Basé sur le changement d'apparence SSD, pour une paire de pixels :

$$\big(I(u+\Delta x, v+\Delta y) - I(u,v)\big)^2 \approx \Big(I(u,v) + I_x(u,v)\Delta x + I_y(u,v)\Delta y - I(u,v)\Big)^2$$

$$\approx \big(I_x(u,v)\Delta x + I_y(u,v)\Delta y\big)^2 \approx I_x(u,v)^2\Delta x^2 + I_y(u,v)^2\Delta y^2 + 2I_x(u,v)I_y(u,v)\Delta x\Delta y$$

Comme pour Moravec, on regarde la distribution des SSD avec les patches voisines

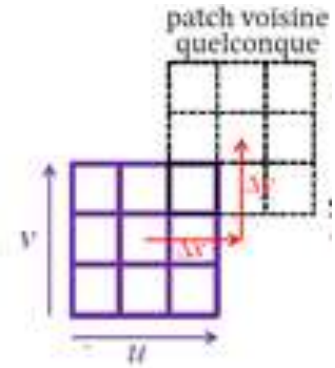
Soit la représentation matricielle SSD1 pour une paire pixel :

$$\big(I(u+\Delta x, v+\Delta y) - I(u,v)\big)^2 \approx [\Delta x \quad \Delta y] \begin{bmatrix} I_x(u,v)^2 & I_x(u,v)I_y(u,v) \\ I_x(u,v)I_y(u,v) & I_y(u,v)^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

La fonction d'autocorrélation de l'image en $(\Delta x, \Delta y)$ est donnée par:

$$C(\Delta x, \Delta y) = \sum_{u,v} [I(u,v) - I(u+\Delta x, v+\Delta y)]^2$$

Qui constitue le SSD d'une patch au complet décalée par Δx, Δy est :



$$SSD_{patch}(\Delta x, \Delta y)$$

$$\approx [\Delta x \quad \Delta y] \left(\sum_{u,v} \begin{bmatrix} I_x(u,v)^2 & I_x(u,v)I_y(u,v) \\ I_x(u,v)I_y(u,v) & I_y(u,v)^2 \end{bmatrix} \right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$SSD_{patch}(\Delta x, \Delta y) \approx [\Delta x \quad \Delta y] M \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

La matrice:

$$M = \sum_{u,v} \begin{bmatrix} I_x(u,v)^2 & I_x(u,v)I_y(u,v) \\ I_x(u,v)I_y(u,v) & I_y(u,v)^2 \end{bmatrix}$$ Nous donne une idée de la variabilité locale d'une patch

Ses valeurs propres λ_1 et λ_2 décrivent l'étalement des valeurs du gradient dans deux directions orthogonales (et peuvent être associées aux "courbures" principales de la fonction d'autocorrélation).

On calcule la valeur R (la réponse de l'opérateur) définie comme:

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

où $det(M)$ est le déterminant $de\ M$ qui est égal à $\lambda_1 . \lambda_2$

et $trace(M$est la trace de M qui vaut $\lambda_1 + \lambda_2$. (en assumant que $\lambda_1 > \lambda_2$) on a :

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad 0.4 \leq k \leq 0.6$$

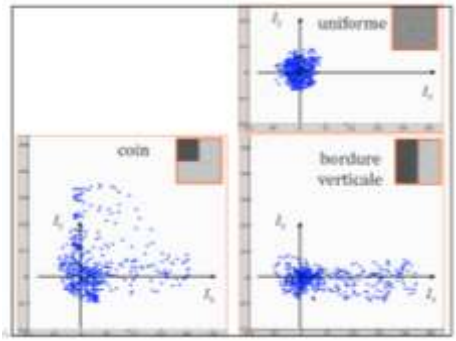


FIGURE 70 DISTRIBUTION DES DERIVEES I_x , I_Y

• Distribution des valeurs propres λ_1 , λ_2 de M

$$M = R^{-1}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

$$(car\ M\ est\ symétrique)$$

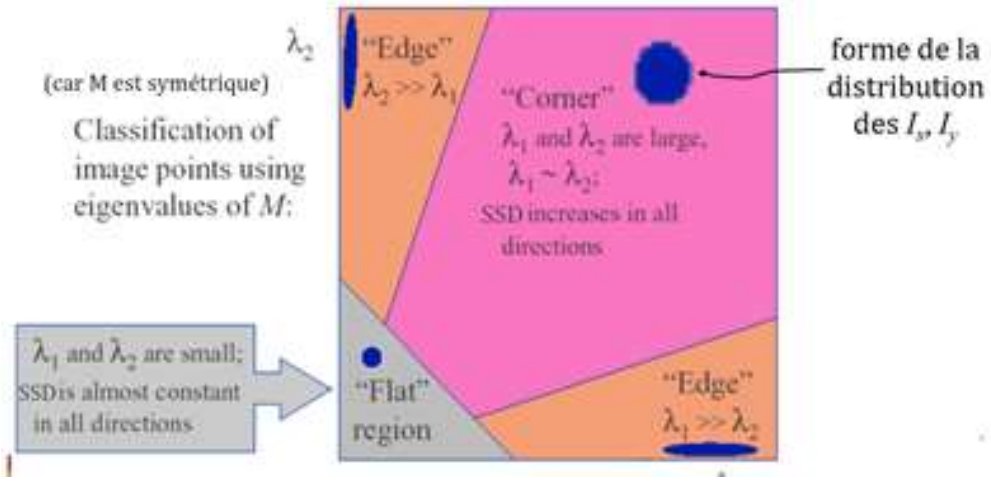


FIGURE 71 DISTRIBUTION DES VALEURS PROPRES Λ_1 , Λ_2 DE M

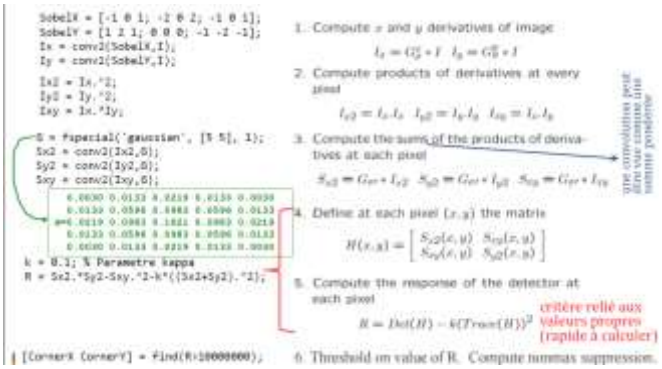


FIGURE 72 HARRIS CORNER DETECTION : ALGORITHME (VERSION MATLAB)

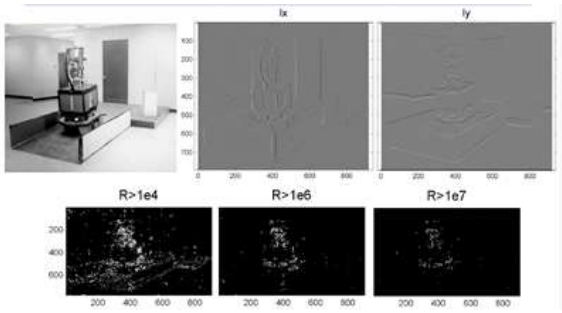


FIGURE 73 RESULTATS HARRIS (SANS NONMAX SUPPR.)

FIGURE 74 RESULTATS HARRIS

Une opération supplémentaire appelé suppression des non maxima locaux est appliqué. En effet, pour une région ou on a des coins qui sont situés un a côté de l'autre on va conserver un seul des coins qui a la valeur R la plus grande on élimine les autres)

c. FAST : Features from Accelerated Segment Test

L'objectif est de faire une technique encore plus rapide que la technique Harris

- Notre définition de coin est un **arc continu** de N ou + pixels
- un peu plus intense que le point central p: $I(x) > (I(p) + t)$ $(I(x) point\ sur\ l'arc)$ ou
- un peu moins intense que le point central p: $I(x) < (I(p) - t)$

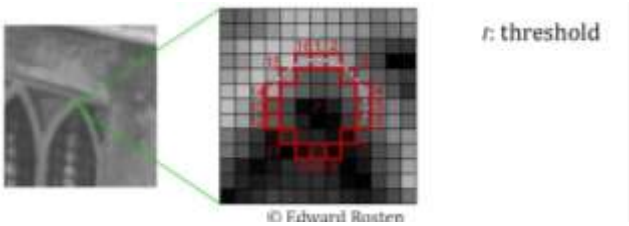


FIGURE 75

On un cercle de rayon 3 pixel satisfait un critère ou un autre

- Pour $N \geq 12$, sur rayon de 3 pixels, rapidité!
- on test si 1,9 respecte l'un des critères
- puis 5…puis 13.
- Si on a trois points qui respectent le critère, on test tous les 12 autres points restants
- Suppression des non-maxima locaux du score

$$V = max \begin{cases} \sum(pixel\ values - p) & if\ (value - p) > t \\ \sum(p - pixel\ values) & if\ (p - value) > t \end{cases}$$

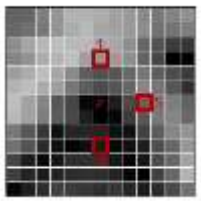


FIGURE 76 ***(SOMME DES DIFFERENCES ABSOLUES ENTRE LE PIXEL P ET TOUS LES PIXELS DE L'ARC CONTINU)***

FAST est 20x plus rapide que Harris

3. Descripteur

a. principe

- Etape 1 : identifier les keypoints (coins)
- Etape 2 : leur attribuer une signature=descripteur qui va réindentifier une image à une autre

(un « sommaire »)

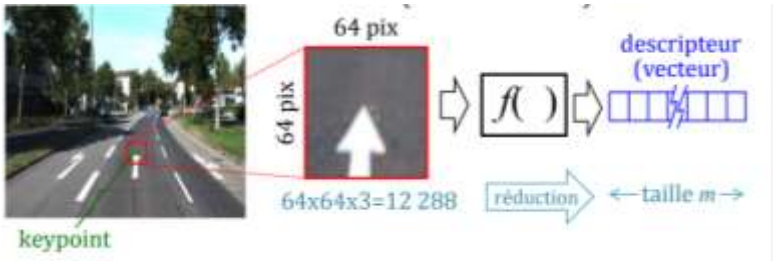


FIGURE 77

- Au final, feature = position (u, v) dans l'image + descripteur

Ex1 : Exemple simplifié : Vecteur des moyenne d'intensités 2×2

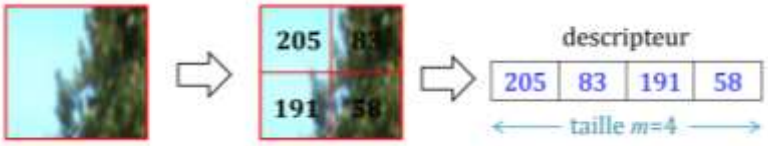


FIGURE 78

On va prendre une image on va la diviser en quatre région ou on va faire la moyenne des intensités lumineuses dans chacune des quatre régions et ça va ainsi donner le descripteur

Ex. 2 : Histogramme de couleurs

FIGURE 79

b. Propriétés désirables d'un descripteur :

- Compact
- dimension m de taille plus faible que la patch
- ex: patch de 64×64 pixels → vecteur taille $m = 128$
- Rapide à calculer
- Distinctif (si deux images sont différentes leurs descripteurs doivent être différent)
- Répétable : robuste aux changements de points de vue (rotation, translation, échelle) et d'illumination

4. Appariement par force brute

Une fonction de distance $s(f_i, f_j)$ qui compare les deux descripteurs f_i et f_j afin de faire la correspondance (Data association)

$$s(f_i, f_j) = \|f_i - f_j\|_2 = \sqrt{\sum_{a=1}^{m} (f_i{}^a - f_j{}^a)^2}$$

FLANN: Fast Library for Approximate Nearest Neighbors KD-trees

Distance cosine $s(f_i, f_j) = f_i . f_j$

- Pour tous descripteurs f_i dans l'image 1
- calculer $s(f_i, f_j)$ pour tous descripteurs f_j dans image 2
- conserver le descripteur f_j le plus proche de f_i

- Seuil maximal sur la distance $s(f_i, f_j)$
- Mutually-best match
  - f_a est le plus proche de f_j
  - f_j est le plus proche de f_a

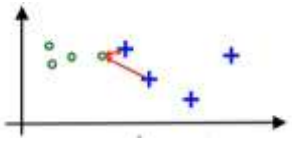


FIGURE 80 ESPACE DESCRIPTEUR

- Test de Lowe:
  - distance s_1 du plus similaire selon $s(f_i, f_j)$
  - distance s_2 du deuxième plus similaire selon $s(f_i, f_j)$
  - ratio des distances s_1/s_2
  - conserve si ce ratio est inférieur à p (typique 0.6-0.8)

5. Vérification lowe/géom

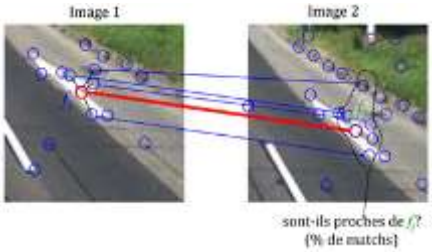


FIGURE 81

Répartition sur l'image

- Diviser l'image d'entrée en plusieurs zones
- Conserver les n features plus « forts »
- Évite de concentrer les features dans une seule région : réduit l'erreur de localisation

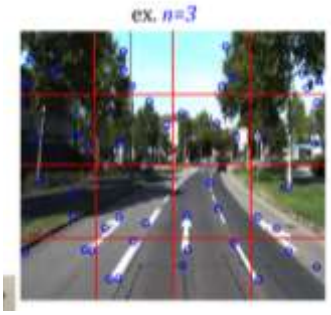


FIGURE 82

3. Exemple de descripteur SIFT

Il existe plusieurs SIFT, SURF, BRIEF, ORB nous allons nous intéresser à SIFT Scale Invariant Feature Transform dont l'bjectif de l'algorithme SIFT est de détecter et d'identifier les éléments similaires entre des images numériques avec une invariance:

- à l'échelle; au cadrage; à l'angle d'observation et à la luminosité.

FIGURE 83

Image du dessin observé sur un écran d'ordinateur avec une orientation, une luminosité et une échelle différentes de celles du dessin original

La stratégie consiste à trouver un descripteur (le descripteur SIFT) pour des pixels qui soit stable en fonction de l'échelle de l'image. Ce descripteur doit être robuste aux changements d'orientation, à la luminosité et au cadrage

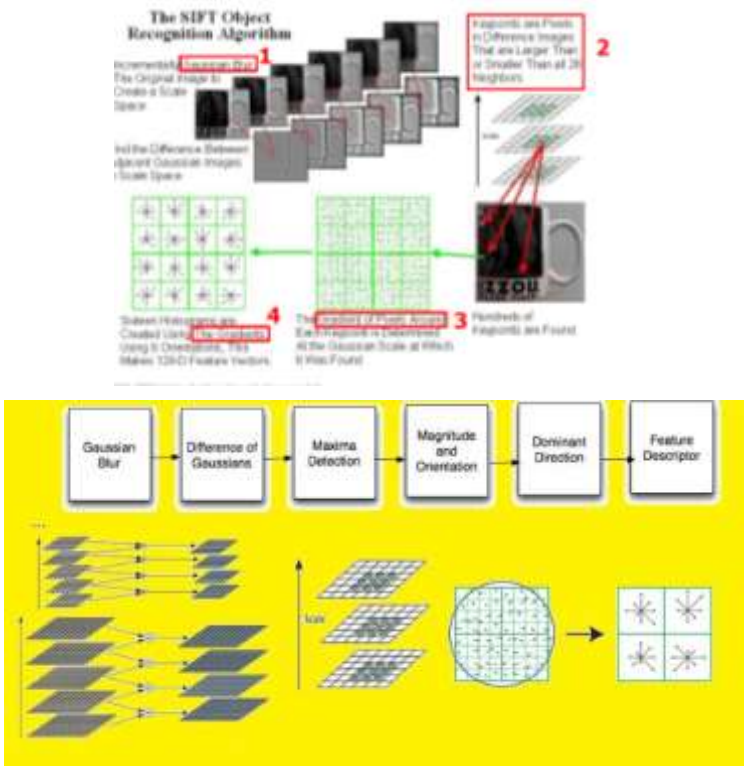


FIGURE 84

ETAPE 1 : Construction de la pyramide d'images filtrées par des gaussiennes : Construire l'espace d'échelles $L(x, y, \sigma)$ de l'image en Filtrant l'image d'origine avec un filtre gaussien d'écart-type (échelle) σ telsque $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$

- Générant chaque nouvelle échelle comme un facteur constant de l'échelle précédente ($\sigma_0 = K\sigma_0$)
- On construit une pyramide en soustrayant les images de deux échelles voisines (différences de gaussiennes $DoG \rightarrow D(x, y, \sigma)) = L(x, y, k\,\sigma) - L(x, y, \sigma)$

On effectue cette opération sur plusieurs octaves (i.e. un octave est situé à 2σ de l'échelle précédente c'est-à-dire en divisant la résolution de l'image par 2)

- D'après les analyses expérimentales menées par Lowe, les paramètres optimaux de l'approche pyramidale sont les suivants: nombre d'octaves : 4, nombre d'échelles par octave $n_e = 5$, nombre d'intervalles dans un octave $s = 2$, nombre d'images par octave $n_i = s + 3 = 5$, facteur multiplicatif de l'écart-type d'une échelle $k = \sqrt{2}$

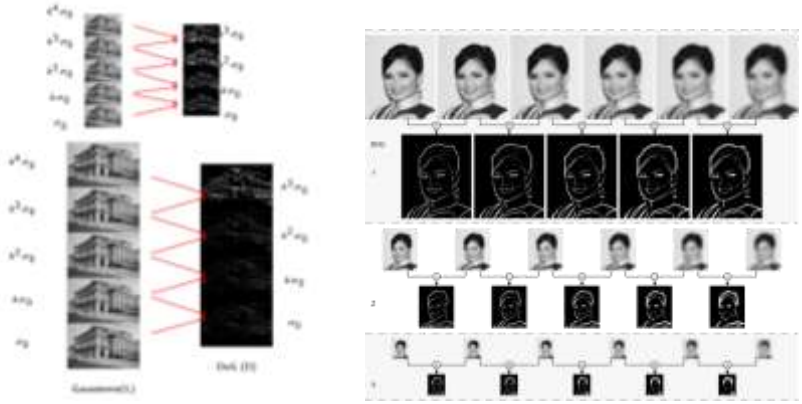


FIGURE 85

ETAPE 2 : Détection des extrémums de différences de gaussiennes dans l'espace des échelles

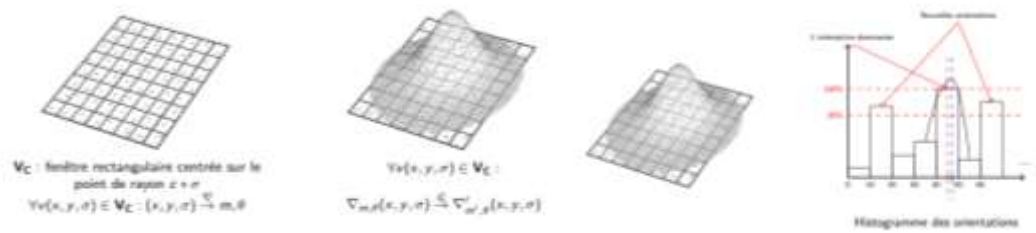
Élimination des pixels d'intérêt de faible contraste

$|D(\hat{x})| < 0.03$ permet d'éliminer ceux qui sont instable et situés dans les zones de faible contraste

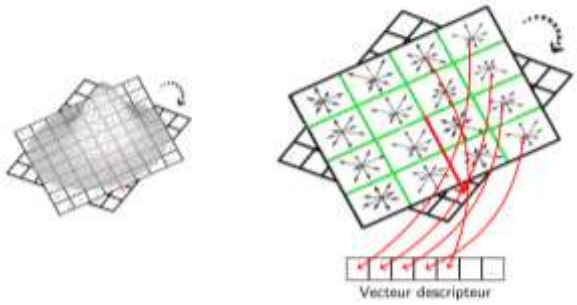
Élimination des pixels d'intérêt situés sur les arêtes d'illuminance

ETAPE 3: Assignation d'orientation privilégiée aux pixels d'intérêt (16×16)

L'histogramme d'orientations comporte 36 alvéoles ("bins") couvrant la plage complète des 360° d'orientations possibles au pixel d'intérêt



ETAPE 4 : Construction du descripteur SIFT (8 × 8)



Le descripteur est formé en concaténant les entrées des 8 alvéoles des 4 x 4 sous-regions pour former un vecteur de 4 x 4 x 8 = 128 éléments

VIII. Odémétrie visuelle

1. Introduction

- Odométrie : estime les déplacements incrémentaux du robot en fonction mouvement des actionneurs

FIGURE 86

- Difficile si : robot à pattes, sol très accidenté, sol glissant (sable)
- Avantages : n'est pas sujet au glissement des roues, trajectoire estimée plus précise qu'avec les roues
- Désavantages : propreté des lentilles, sensibilité illumination (capteur passif), position dérive avec le temps, 1 seule caméra : on n'a pas l'échelle ➔d'où l'utilisation de la stéréo

2. Odométrie visuelle (visual odometry: VO)

- Utiliser les caméras pour mesurer les déplacements relatifs (incrémentaux) du robot entre les images
- Peut utiliser différentes configurations de caméras
- 1 seule
- paire stéréo ← configuration étudiée
- caméra omnidirectionnelle

Exemple VO avec stéréo

- Identifier des « features » f_i dans les images I_G, I_D

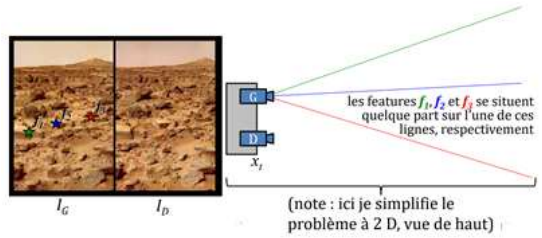


Figure 87

- Trouve les positions des f_i dans l'environnement avec la stéréo (ici, l'intersection des rayons) (rotation) (translation)

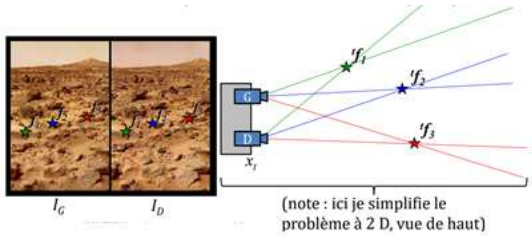


FIGURE 88

Cherche transformation R et T entre les deux poses x_t et x_{t+1}

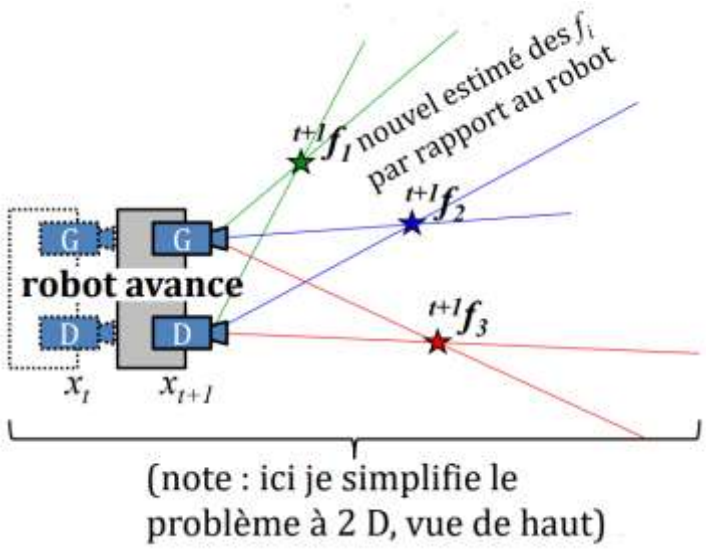


FIGURE 89

Transformation

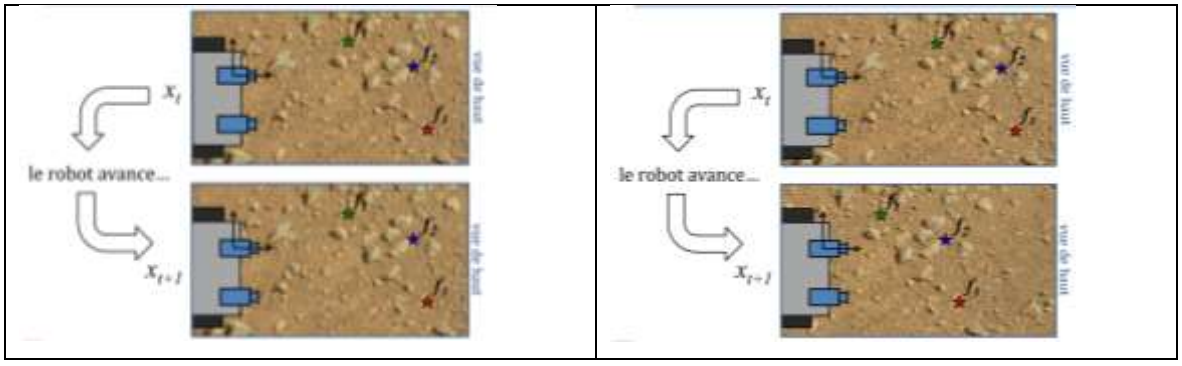


FIGURE 90

ici les features f_i sont plus proches car le robot a avancé

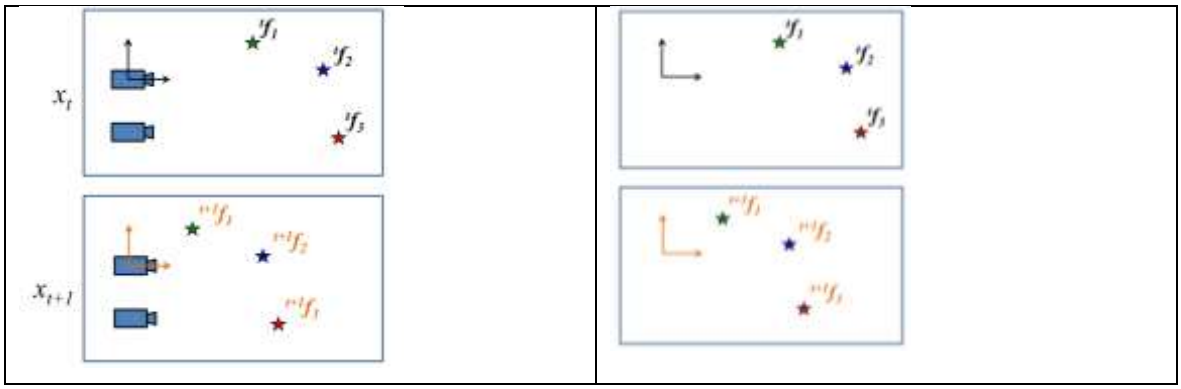


FIGURE 91

Opération de recalage : Trouver R et T pour faire matcher les ${}^{t+1}f_i$ avec les ${}^{t}f_i$

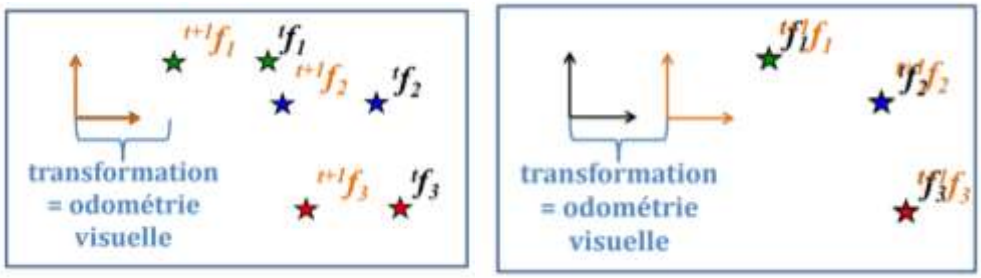


FIGURE 92

L'odemetrie visuelle consiste de chercher les opérations géométriques

• Dans la réalité, il va y avoir des incertitudes

- position (pixel)→angles (variation approx. normale)
- erreur dans l'appariement des features f_i: données aberrantes (outliers)

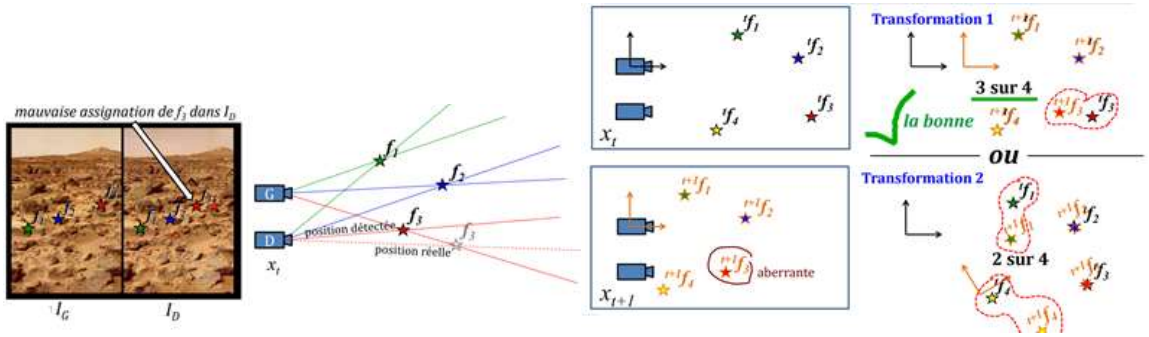


FIGURE 93 CAS AVEC 1/4 DONNEE ABERRANTE

3. RANSAC : RANdom SAmple Consensus

- Proposé par Fischler et Bolles en 1981
- Méta-algorithme probabiliste
- Permet des régressions (fit) très robuste de modèle, malgré la présence de nombreuses n données aberrantes
- Connait le modèle et le nombre N_{min} de points requis pour un fit
- ligne : 2 points

Algorithm 1 RANSAC

1: Select randomly the minimum number of points required to determine the model parameters N_{min}
2: Solve for the parameters of the model.
3: Determine how many points from the set of all points fit with a predefined tolerance ε_{Tol}.
4: If the fraction of the number of inliers over the total number points in the set exceeds a predefined threshold τ, re-estimate the model parameters using all the identified inliers and terminate.
5: Otherwise, repeat steps 1 through 4 (maximum of N times). $N \approx \frac{1}{w^{N_{min}}}$ w = prob. inlier

Variante : on peut aussi faire toutes les N itérations, et garder le modèle avec le plus grand nombre d'*inliers*

FIGURE 94 ALGORITHME RANSAC

- Variante : on peut aussi faire toutes les N itérations, et garder le modèle avec le plus grand nombre d'inliers
- Tester toutes les combinaisons possibles :

$$k\ parmi\ n = \binom{n = \#éléments}{k = N_{min}} = \frac{n!}{k!(n-k)!} : trop\ grand!$$

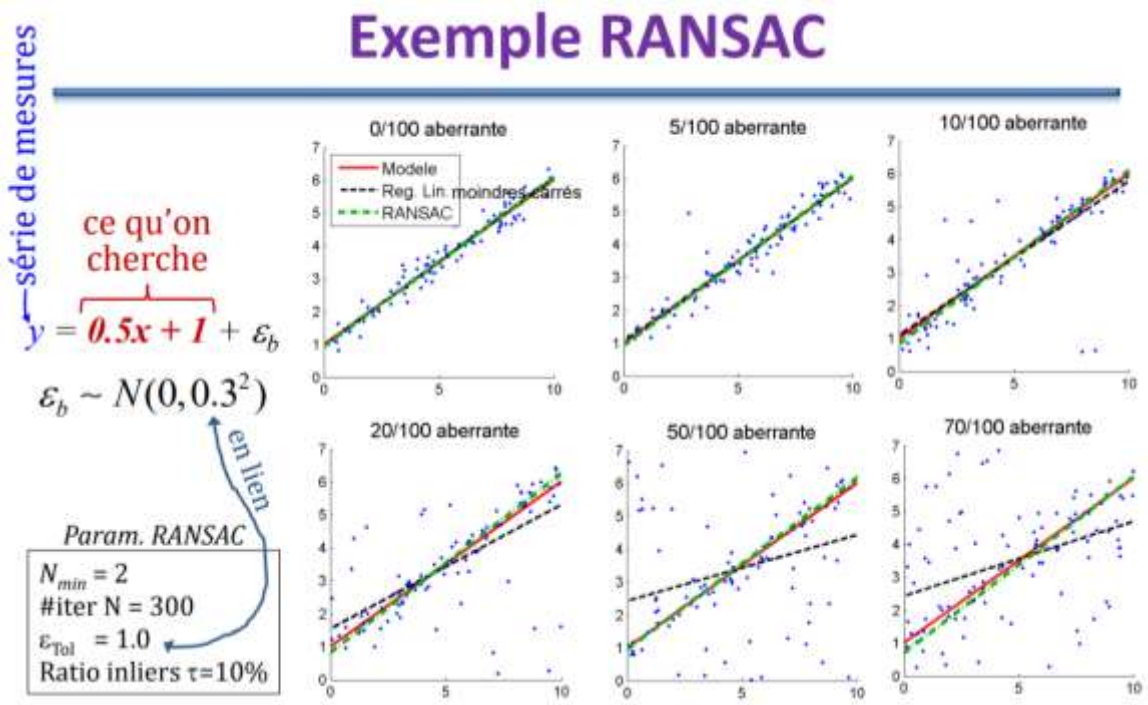


FIGURE 95 EXEMPLE RANSAC

Note : hypothèse sous-jacente des moindres carrés : bruit gaussien.

RANSAC pour VO : Essai #1

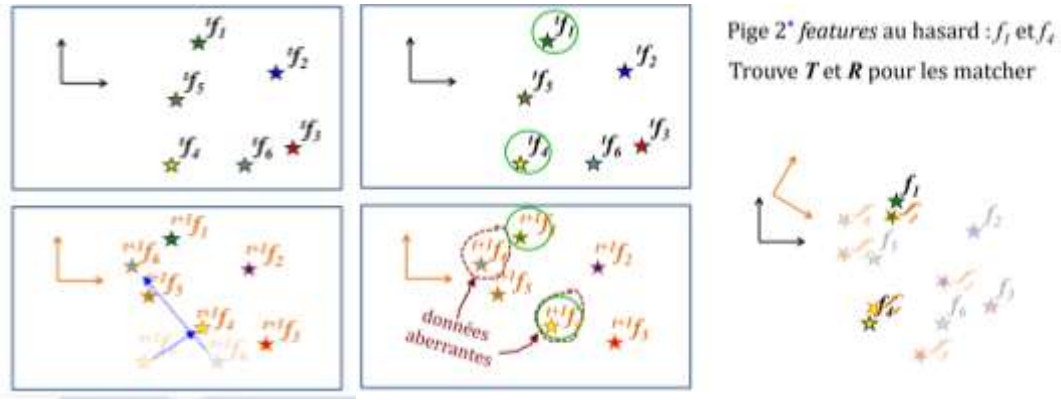


FIGURE 96 PIGE 2* FEATURES AU HASARD : F_1 ET F_4

Trouve T et R pour les matcher

* besoin de seulement 2, car les f_i ne sont pas anonymes

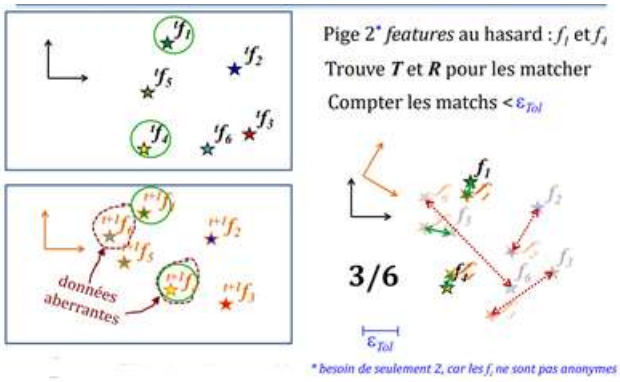


FIGURE 97

RANSAC pour VO : Essai #2

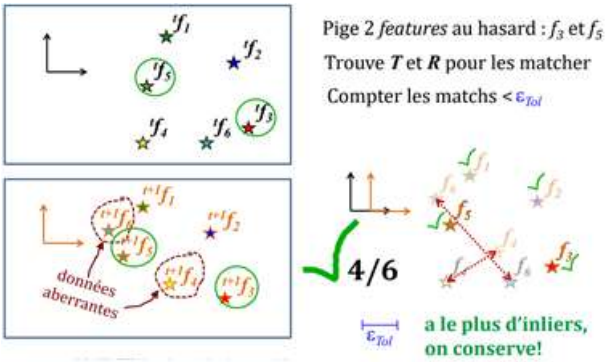


FIGURE 98

<u>Pipeline feature visuel typique</u>

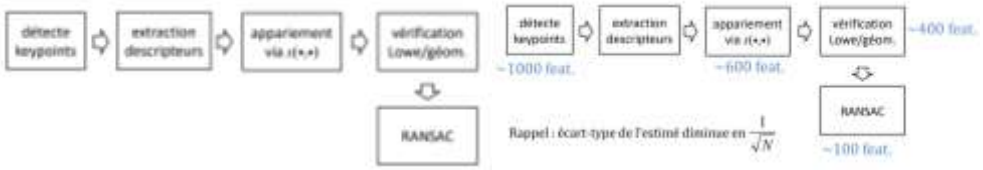


FIGURE 99 APPROCHE CASCADE

<u>Importance d'avoir BEAUCOUP de features en partant!</u>