Stéréovision : Généralités & géométrie épipolaire

Systèmes de vision, INSA - ASI4

Sebastien.Kramm@univ-rouen.fr

LITIS - INSA Rouen

février 2016

Sommaire

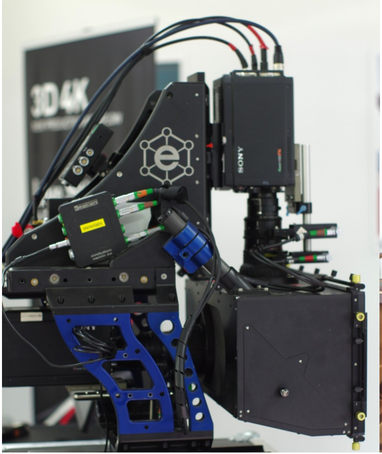
Introduction

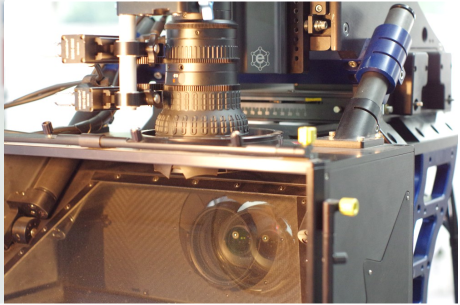
Elphel

ZED / Stereolabs

- Acquisition binoculaire :
  - Deux caméras, acquisition simultanée.
  - Ou : une caméra qui se déplace (si scène statique !)
- Intéret : acquisition de la profondeur.
- Applications : industrie, robotique, automobile (ADAS), réalité augmentée, cinéma, métrologie, cartographie,. . .

ADAS : Advanced Driving Assistance System


3D 4K
SONY



Deux types de productions

Stéréovision dense

- Production d'une carte de profondeur, donnant Z pour tout point de l'image.
    - coût calculatoire élevé (volume de données à traiter lié à la résolution),
    - nécéssité d'un post traitement pour de l'analyse de scène : segmentation et/ou localisation (techniques classiques de CV)

Deux types de productions

Stéréovision dense

- Production d'une carte de profondeur, donnant Z pour tout point de l'image.
    - coût calculatoire élevé (volume de données à traiter lié à la résolution),
    - nécéssité d'un post traitement pour de l'analyse de scène : segmentation et/ou localisation (techniques classiques de CV)

Stéréovision éparse

- La profondeur est calculée uniquement sur des points caractéristiques (angles, coins, droites, obstacles,. . .)
    - volume de données à traiter réduit,
    - mais performances liées à celle du détecteur.

Deux types de productions

Stéréovision dense

- Production d'une carte de profondeur, donnant Z pour tout point de l'image.
  - coût calculatoire élevé (volume de données à traiter lié à la résolution),
  - nécéssité d'un post traitement pour de l'analyse de scène : segmentation et/ou localisation (techniques classiques de CV)

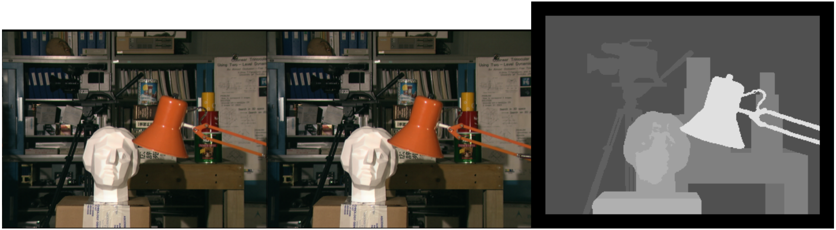
Stéréovision éparse

- La profondeur est calculée uniquement sur des points caractéristiques (angles, coins, droites, obstacles,. . .)
  - volume de données à traiter réduit,
  - mais performances liées à celle du détecteur.

- Point clé dans les 2 cas : l'appariement !

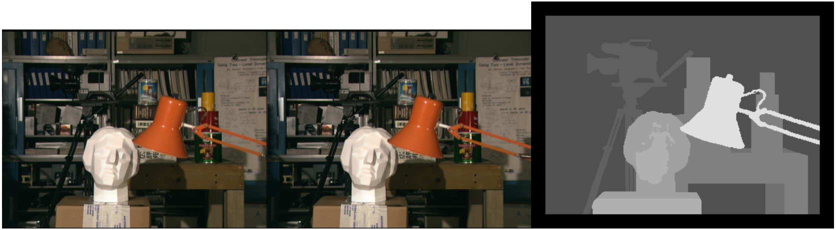
Introduction

- Stéréovision dense

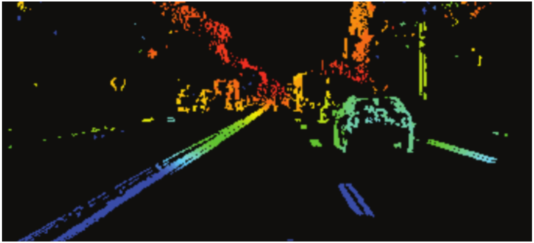


Introduction

- Stéréovision dense



- Stéréovision éparse

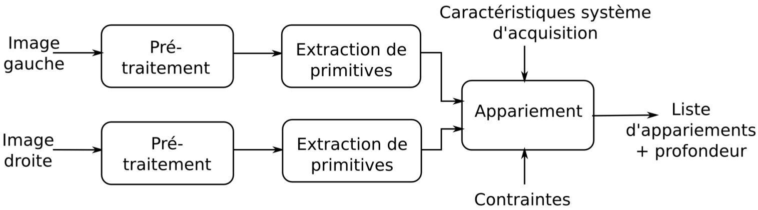


source : JP Tarel, LIVIC

Généralisation

- La stéréovision consiste à :
  1. extraire des primitives (points d'intérêts, segments de droite,. . .),
  2. les apparier,
  3. calculer pour chaque paire une valeur de profondeur ("reconstruction 3D").

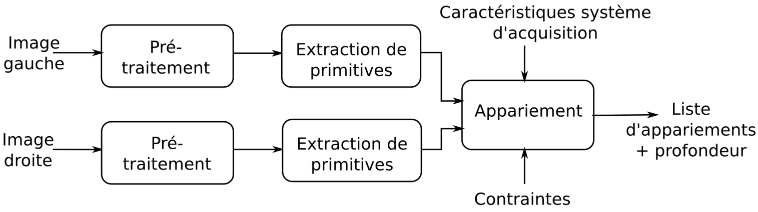
Note : pour certaines applications, on pourra se passer de l'étape 3, et se contenter de la **disparité**.



Généralisation

- La stéréovision consiste à :
  1. extraire des primitives (points d'intérêts, segments de droite,. . .),
  2. les apparier,
  3. calculer pour chaque paire une valeur de profondeur ("reconstruction 3D").

Note : pour certaines applications, on pourra se passer de l'étape 3, et se contenter de la **disparité**.

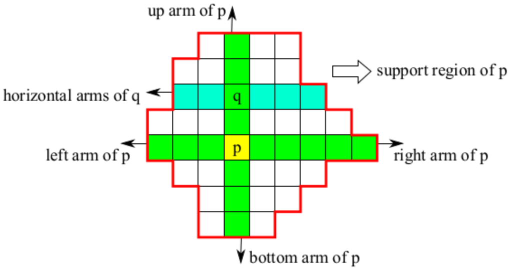


- Stéréovision dense $\Rightarrow$ primitives $=$ pixels de l'image.
- Il est fréquent de réaliser un pré-traitement des images (rectification, recadrage, seuillage,. . .)

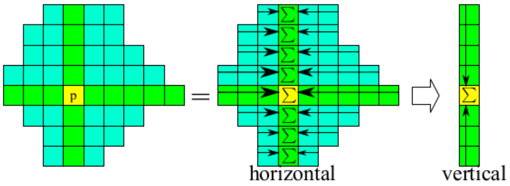
Appariement

- Les algorithmes d'appariement sont dépendant des primitives utilisées :
  - Stéréo dense : corrélation (SSD, ZSSD, ZNSSD, ZNCC, ...), algorithmes de coupure de graphe (*graph-cut*), Techniques récentes : CBA(2011) Cross Based Correlation
  - Stéréo éparse : utilisation de la caractérisation des primitives.
- Difficultés
  - Un point 2D peut **ne pas** avoir de stéréo-correspondant (occultation ou recouvrement des images insuffisant).
  - L'apparence des éléments de la scène peut être différente dans les 2 images.
- Contraintes exploitables :
  - Unicité : une primitive d'une image ne peut correspondre qu'à un seul appariement.
  - Contrainte d'ordre (si scène opaque)
- En pratique, les contraintes utilisables sont dépendantes du **contexte applicatif**.

CBA : Cross Based Aggregation

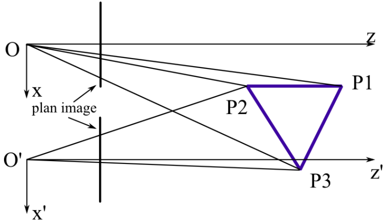


(a) Cross Construction



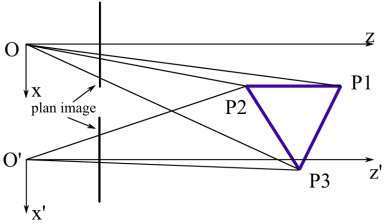
Contraintes & difficultés de l'appariement

- Occultation : si (P1,P2,P3) est un objet opaque, le point P1 n'aura pas de projection dans la caméra droite.

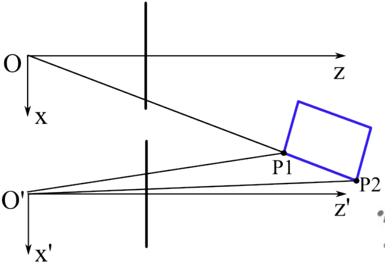


Contraintes & difficultés de l'appariement

- Occultation : si (P1,P2,P3) est un objet opaque, le point P1 n'aura pas de projection dans la caméra droite.

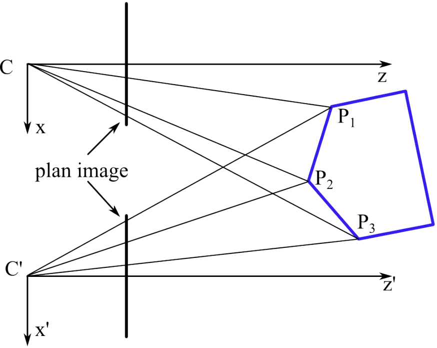


- Unicité : contrainte imposée, mais ne correspond pas toujours à la réalité.



Contraintes & difficultés de l'appariement

- Contrainte d'ordre (valable si éléments opaques) :



- Chaque point P_i se projette
  - dans (Oxz) en u_i
  - dans (O'x'z') en u'_i

- Si $u_2 > u_1$ et $u_2 < u_3$, alors on devra avoir $u'_2 > u'_1$ et $u'_2 < u'_3$

Exemple de primitive : descripteur local

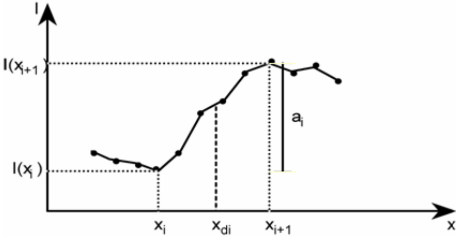
- Un descripteur local effectue deux opérations :
  - recherche de **points d'intérêts** dans l'image,
  - caractérisation de ces points par un **vecteur d'attributs**.
- Le précurseur : SIFT (*Scale-Invariant Feature Transform*) [Lowe,2004]
  - basé sur la théorie des "espaces d'échelles" (*Space scale*),
  - invariant aux rotations, changement d'illumination, etc.,
  - fournit pour chaque point un vecteur de 128 attributs.
- Avantages :
  - richesse de la caractérisation (pouvoir discriminant élevé),
  - position subpixellique.
- Inconvénient : coût.
- Domaine de recherche actif (compromis coût/performance)

Descripteur SIFT

- Exemple d'extraction

Exemple de primitive : opérateur "déclivité"

- On recherche dans l'image, ligne par ligne, les variations significatives de niveau (seuil auto-adaptatif).
- On caractérise par l'amplitude, la pente, et les moyennes des niveaux à gauche et à droite.
- Avantages :
  - coût (très rapide), peut être parallélisé,
  - position horizontale subpixellique.
- Inconvénient : caractérisation faible.

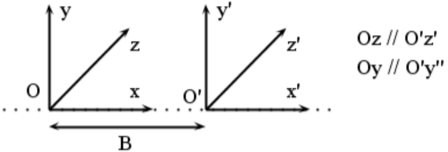


Sommaire

Sous-sommaire

Stéréovision : cas aligné

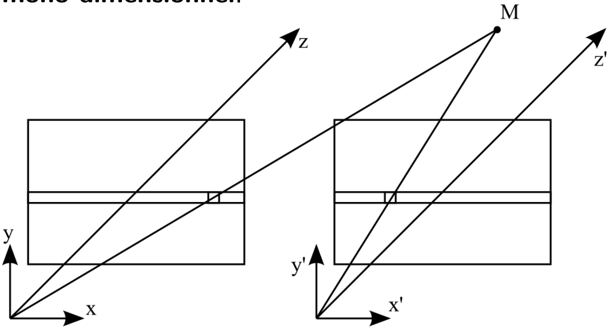
- Les calculs sont simplifiés si les 2 caméras
  - ont des paramètres intrinsèques identiques ($\mathbf{K} = \mathbf{K}'$),
  - sont alignées : axes Ox confondus et axes Oy et Oz parallèles.



- Deux niveaux de simplification
  1. La profondeur s'obtient par une simple division.
  2. Les points stéréo-correspondants sont sur la même ligne image.

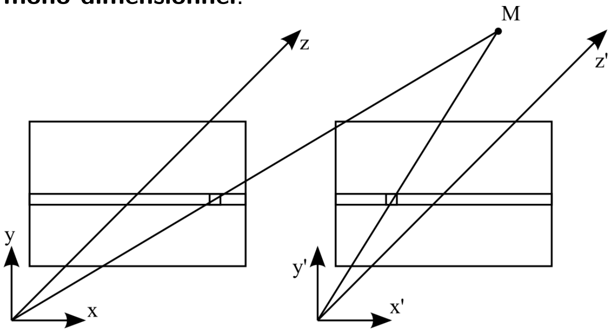
Cas aligné : simplification de l'appariement

- Les points stéréo-correspondants sont sur les **mêmes** lignes-image.
  $\Rightarrow$ L'appariement passe d'un problème **bi-dimensionnel** à un problème **mono-dimensionnel**.



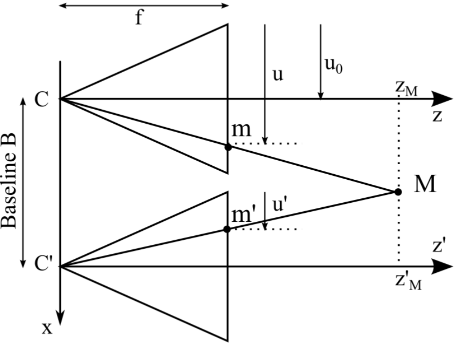
Cas aligné : simplification de l'appariement

- Les points stéréo-correspondants sont sur les **mêmes** lignes-image.
  $\Rightarrow$ L'appariement passe d'un problème **bi-dimensionnel** à un problème **mono-dimensionnel**.



- L'appariement consiste à trouver pour le point (ou pixel) m_{in} (se trouvant sur la ligne n) son stéréo-correspondant m_{jn} dans l'image I' sur la ligne n.

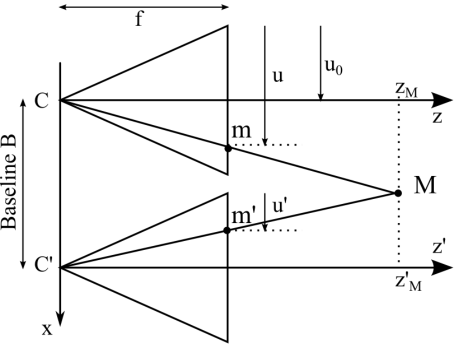
Cas aligné : calcul de la profondeur



- Un point 3D M aura pour projections :
  - $m = (x_M, y_M, z_M)$ dans le repère caméra $(Cxyz)$,
  - $m = (x'_M, y'_M, z'_M)$ dans le repère caméra $(C'x'y'z')$

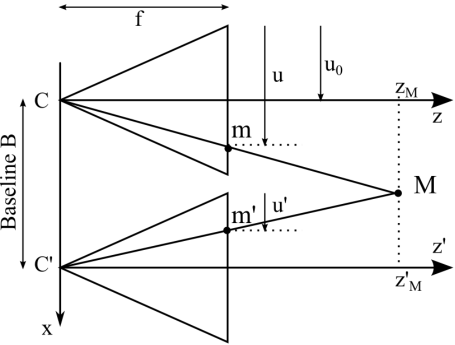
- x'_M est lié à x_M par la relation $x'_M = x_M - B$
- Remarque : dans la configuration du dessin ci-dessus, on aura $x > 0$ et $x' < 0$

Cas aligné : calcul de la profondeur



$$\begin{cases} u = \alpha_u \dfrac{x_M}{z_M} + u_0 \\ u' = \alpha_u \dfrac{x'_M}{z'_M} + u_0 \end{cases} \quad \text{et} \quad z_M = z'_M$$

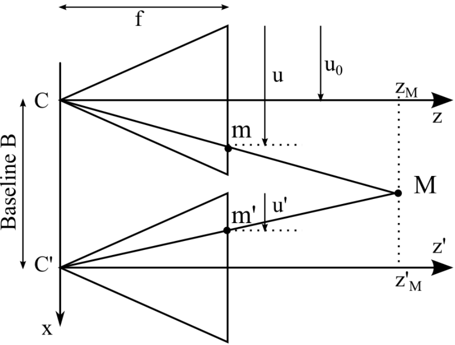
Cas aligné : calcul de la profondeur



$$\begin{cases} u = \alpha_u \dfrac{x_M}{z_M} + u_0 \\ u' = \alpha_u \dfrac{x'_M}{z'_M} + u_0 \end{cases} \quad \text{et} \quad z_M = z'_M$$

$$u - u' = \alpha_u \frac{x_M - x'_M}{z_M}$$
$$= \alpha_u \frac{B}{z_M}$$

Cas aligné : calcul de la profondeur



$$\left\{\begin{array}{l} u = \alpha_u \dfrac{x_M}{z_M} + u_0 \\ u' = \alpha_u \dfrac{x'_M}{z'_M} + u_0 \end{array}\right. \quad \text{et} \quad z_M = z'_M$$

$$u - u' = \alpha_u \frac{x_M - x'_M}{z_M} = \alpha_u \frac{B}{z_M}$$

La profondeur est inversement proportionnelle à la disparité $u - u'$

$$z_M = \frac{B\,\alpha_u}{u - u'} \qquad d = u - u' > 0 \qquad d = 0 \Leftrightarrow z_M = \infty$$

Importance de la *baseline*

- La précision est directement proportionnelle à l'écart entre les caméras.
- Il devra être choisi selon le domaine applicatif (vision à 2 m. ou à 50 m. ?)
- Les algorithmes d'appariement utilisent souvent une limite haute sur la disparité, afin de limiter les erreurs d'appariement.
  (Exemple : 1/10 de la largeur de l'image)

Importance de la *baseline*

- La précision est directement proportionnelle à l'écart entre les caméras.
- Il devra être choisi selon le domaine applicatif (vision à 2 m. ou à 50 m. ?)
- Les algorithmes d'appariement utilisent souvent une limite haute sur la disparité, afin de limiter les erreurs d'appariement.
  (Exemple : 1/10 de la largeur de l'image)
- Exercice 1 : avec cette limite et une image "VGA", quel sera la distance minimale observable ($\alpha = 500$) ?
  - pour B=20 cm ?
  - pour B=50 cm ?

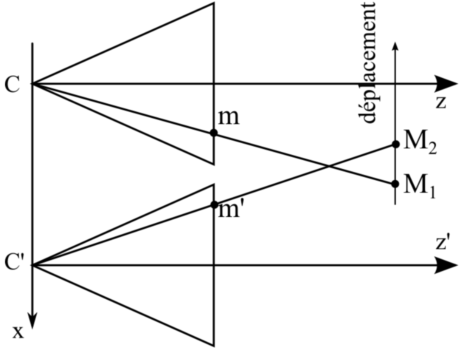
Importance de la *baseline*

- La précision est directement proportionnelle à l'écart entre les caméras.
- Il devra être choisi selon le domaine applicatif (vision à 2 m. ou à 50 m. ?)
- Les algorithmes d'appariement utilisent souvent une limite haute sur la disparité, afin de limiter les erreurs d'appariement.
  (Exemple : 1/10 de la largeur de l'image)
- Exercice 1 : avec cette limite et une image "VGA", quel sera la distance minimale observable ($\alpha = 500$) ?
  - pour B=20 cm ?
  - pour B=50 cm ?
- Exercice 2 : on souhaite avoir un système stéréo qui fournit une disparité de 100 pixels pour un objet situé à 2m. Quel devra être la valeur de B ?
  - pour $\alpha = 200$?
  - pour $\alpha = 500$?

Contrainte d'implémentation : aspect temporel

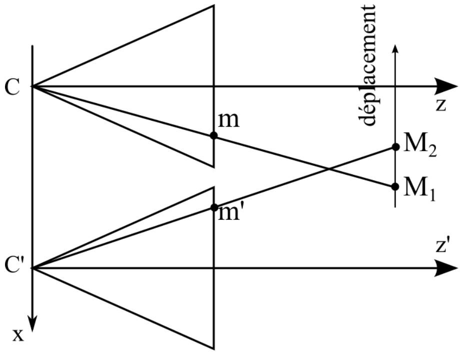
- En cas de scène dynamique, il est **impératif** que l'acquisition soit simultanée.
- Si l'objet se déplace, sa position dans l'image sera modifiée $\Rightarrow$ erreur sur la profondeur mesurée.

Exercice



- Soit un point $M = (x_M, 0, z_M)$ se deplaçant parallèlement à Ox, à une vitesse v.

Exercice



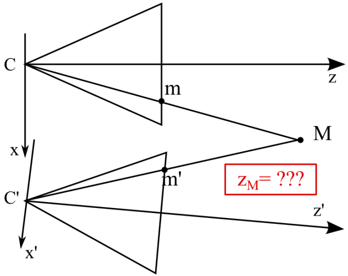
- Soit un point $M = (x_M, 0, z_M)$ se deplaçant parallèlement à Ox, à une vitesse v.

- La caméra stéréo a un framerate $fr = 20$ paires/s, et réalise l'acquisition de l'image G puis D de façon alternée avec un intervalle de temps Δ_t.
  - exprimer la profondeur observée z_ϵ si $\Delta t \neq 0$
  - exprimer l'erreur relative ϵ (%)
  - A.N. : focale $f = 8$mm, taille pixel=8μm, B=50cm, $v = 15 km/h$, $z = 15$m.

Sous-sommaire

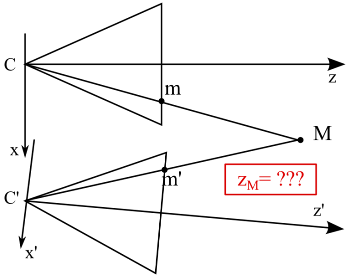
Systèmes réels

- Si les caméras ne sont pas alignées, on **ne peut plus** obtenir la profondeur par une simple division.
- L'appariement devient aussi plus complexe. . . (recherche 2D)



Systèmes réels

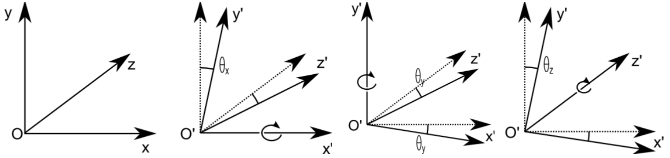
- Si les caméras ne sont pas alignées, on **ne peut plus** obtenir la profondeur par une simple division.
- L'appariement devient aussi plus complexe. . . (recherche 2D)



- Problème : pour des écartements importants, il devient délicat **d'obtenir** et de **maintenir** un alignement "parfait".

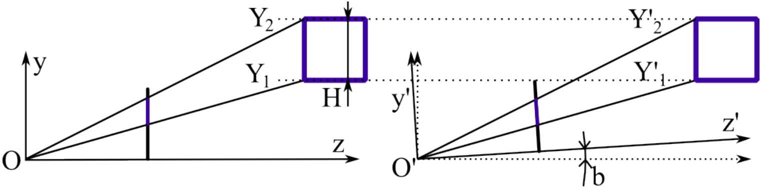
Influence du désalignement

- Les perturbations les plus importantes seront liées aux écarts en rotation entre caméras.
- L'influence sur la profondeur mesurée dépend de l'axe de rotation considéré.
  - axe horizontal (Ox) : peu d'erreur sur la profondeur, mais sur la hauteur apparente des élements observés, après appariement.
  - axe vertical (Oy) : influera directement sur la disparité et donc sur la profondeur mesurée.
  - axe optique (Oz) : combinaison des 2 précédentes erreurs, selon la position dans l'image du point considéré.
- Dans la réalité, les rotations seront évidemment combinées, rendant la correction par post-traitement très difficile.



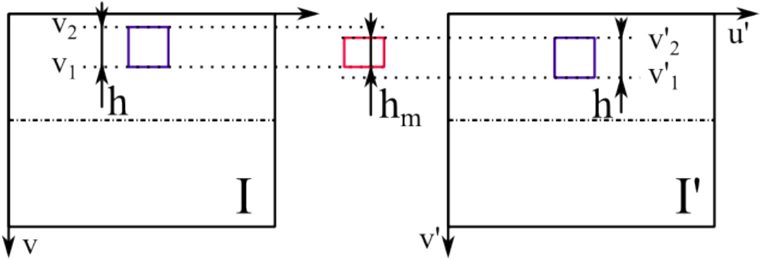
Exercice

- On suppose un système de caméras alignées, sauf sur l'axe (Ox), sur lequel un écart de b degrés existe entre la caméra gauche et droite.
- Un objet vertical (cube) de hauteur H et dont la base est à une hauteur Y_1, situé à une distance Z est observé par ce système, et se projette avec une hauteur h dans chaque image.
- Déterminer la hauteur apparente observée après appariement, et comparer avec celle obtenue par un système aligné.
- A partir de quel angle l'objet ne sera plus du tout apparié ?
- A.N. : $\alpha_v = -500$, $Z_M = 10\text{m}$, $H = 1\text{m}$
  - $b = 1^\circ$
  - $b = 2^\circ$



Exercice

- La projection de l'objet dans l'image droite sera **décalée verticalement** par rapport à l'image gauche.



- Après appariement ligne par ligne, la hauteur de l'objet reconstruit sera égale à la partie commune entre gauche et droite.

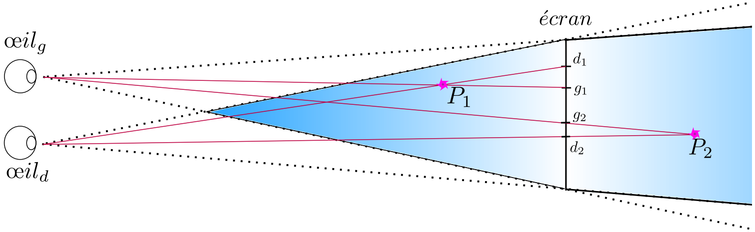
Systèmes réels

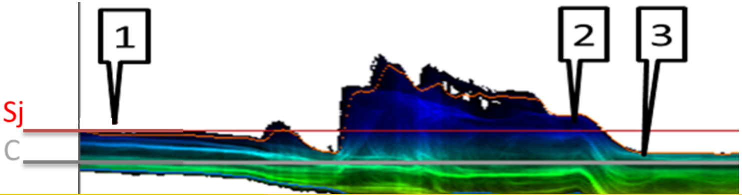
- En dehors d'applications spécifiques, le cas aligné a peu d'applications pratiques :
  - L'alignement mécanique des caméras est délicat, et risque de dériver (vibrations).
  - Difficile de quantifier l'erreur en fonction de la position réelle.

  $\Rightarrow$ nécessité d'un **calibrage stéréoscopique**.
- Vocable ambigu, peut désigner :
  - l'action d'aligner mécaniquement les deux caméras,
  - la caractérisation de la géométrie relative entre caméras.

Cinéma 3D

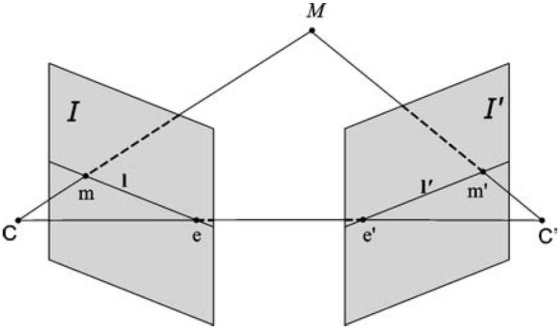
- Pour de la production de contenus 3D, pas besoin de métrique : on utilise des systèmes à convergence : axes (Oy) parallèle, rotation sur (Oy)
- On peut ainsi faire varier la disparité (et donc la perception visuelle de la profondeur) dans une plage définie par le réalisateur.





Plan épipolaire

- Le point M, et les deux foyers optiques vont déterminer un **plan épipolaire**.
- L'intersection de la droite reliant les 2 centre optiques avec les plan images va générer les **épipôles**.
- Ces épipôles peuvent être **dans** ou **hors** de l'image réelle.
- A un point dans une image correspond une droite dans l'autre image ("droite **épipolaire**").



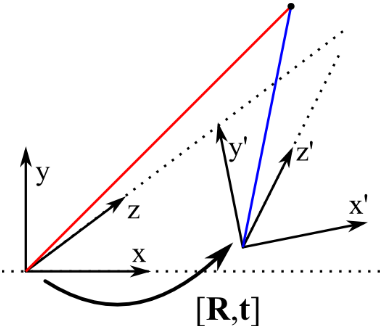
Caractérisation de la géométrie épipolaire

- La géométrie épipolaire peut-être entièrement définie par deux matrices :
  - la **matrice Essentielle**, si on connait les paramètres intrinsèques,
  - la **matrice Fondamentale**, si on ne les connait pas.
- Ces matrices permettent d'associer **un point** d'une image à **une droite** dans l'autre image
  - matrice Essentielle : dans le repère caméra (C, x, y)
  - matrice Fondamentale : dans le repère image (O, u, v)

Sous-sommaire

Modélisation de la géométrie épipolaire

- Un point de la scène peut être exprimé dans les repères des 2 caméras : $\mathbf{M} = (X, Y, Z, 1)^T$ et $\mathbf{M}' = (X', Y', Z', 1)^T$



- Les repères sont liés par une homographie 3D **A**, composée d'une rotation **R** et une translation $\mathbf{t} = (t_x, t_y, t_z, 1)^T$:

$$\mathbf{A} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{33} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Modélisation - 2

- On peut exprimer la position du point dans un repère en utilisant les coordonnées de l'autre repère :

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} \simeq \mathbf{A} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} \simeq \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{33} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

- Sous forme développée :

$$\begin{cases} X' = r_{11}X + r_{12}Y + r_{13}Z + t_x \\ Y' = r_{21}X + r_{22}Y + r_{23}Z + t_y \\ Z' = r_{31}X + r_{32}Y + r_{33}Z + t_z \end{cases}$$

Projection en 2D

- La **projection** du point 3D dans l'image droite génère le point image 2D (x', y') :

$$\begin{cases} x' = \dfrac{X'}{Z'} = \dfrac{r_{11}X + r_{12}Y + r_{13}Z + t_x}{r_{31}X + r_{32}Y + r_{33}Z + t_z} \\ y' = \dfrac{Y'}{Z'} = \dfrac{r_{21}X + r_{22}Y + r_{23}Z + t_y}{r_{31}X + r_{32}Y + r_{33}Z + t_z} \end{cases}$$

- en utilisant la relation $X = x.Z$ et $Y = y.Z$, on arrive à

$$\begin{cases} x' = \dfrac{Z(r_{11}x + r_{12}y + r_{13}) + t_x}{Z(r_{31}x + r_{32}y + r_{33}) + t_z} \\ y' = \dfrac{Z(r_{21}x + r_{22}y + r_{23}) + t_y}{Z(r_{31}x + r_{32}y + r_{33}) + t_z} \end{cases}$$

- en notant $\mathbf{r_i}$ la i-ème ligne de la matrice $\mathbf{R}$, et $\mathbf{m} = (x, y, 1)^T$:

$$x' = \frac{Z\ \mathbf{r_1}\ \mathbf{m} + t_x}{Z\ \mathbf{r_3}\ \mathbf{m} + t_z} \qquad y' = \frac{Z\ \mathbf{r_2}\ \mathbf{m} + t_y}{Z\ \mathbf{r_3}\ \mathbf{m} + t_z} \quad (1)$$

Relations entre les projections

Interprétation

La projection d'un point 3D dans l'image **droite** peut s'exprimer en fonction :

1. de la projection de ce point dans l'image **gauche**,
2. des caractéristiques de la tête stéréo (rotation et translation entre caméras),
3. de la profondeur z.

Relations entre les projections

Interprétation

La projection d'un point 3D dans l'image **droite** peut s'exprimer en fonction :

1. de la projection de ce point dans l'image **gauche**,
2. des caractéristiques de la tête stéréo
   (rotation et translation entre caméras),
3. de la profondeur z.

- On peut formaliser encore plus : à partir des expressions précédentes, on sort Z :

$$Z = \frac{t_x - x' t_z}{x' \mathbf{r_3 m} - \mathbf{r_1 m}} \qquad Z = \frac{t_y - y' t_z}{y' \mathbf{r_3 m} - \mathbf{r_2 m}}$$

- Ce qui permet d'écrire :

$$\frac{t_x - x' t_z}{x' \mathbf{r_3 m} - \mathbf{r_1 m}} = \frac{t_y - y' t_z}{y' \mathbf{r_3 m} - \mathbf{r_2 m}}$$

- Après développement, on arrive à :

$$x'(t_z \mathbf{r_2} - t_y \mathbf{r_3})\mathbf{m} - y'(t_z \mathbf{r_1} - t_x \mathbf{r_3})\mathbf{m} - (t_x \mathbf{r_2} - t_y \mathbf{r_1})\mathbf{m} = 0$$

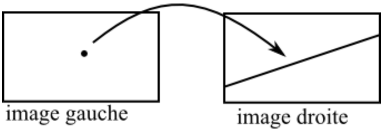
- Ceci correspond à l'équation d'une droite $\mathbf{l}'$ dans l'image droite, et peut être écrit $x'a' + y'b' + c' = 0$, avec :

$$\begin{cases} a' = (t_z \mathbf{r_2} - t_y \mathbf{r_3})\mathbf{m} \\ -b' = (t_z \mathbf{r_1} - t_x \mathbf{r_3})\mathbf{m} \\ -c' = (t_x \mathbf{r_2} - t_y \mathbf{r_1})\mathbf{m} \end{cases}$$

Relation point → droite épipolaire

Interprétation : relation point-droite

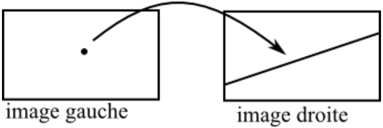
- A un **point** dans une image correspond une **droite** dans l'autre image.
- Cette droite est appellée **droite épipolaire**, et ne dépend **que** de la relation inter-caméras.
- Cette relation n'est **pas** réversible (on ne peut pas obtenir un point dans I à partir d'une droite dans I').



Relation point $\rightarrow$ droite épipolaire

Interprétation : relation point-droite

- A un **point** dans une image correspond une **droite** dans l'autre image.
- Cette droite est appellée **droite épipolaire**, et ne dépend **que** de la relation inter-caméras.
- Cette relation n'est **pas** réversible (on ne peut pas obtenir un point dans I à partir d'une droite dans I').



- Les coefficients de cette droite s'écrivent :

$$\begin{pmatrix} a' \\ b' \\ c' \end{pmatrix} = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Sous-sommaire

Matrice Essentielle - E

- Cette relation est formalisée par la **matrice Essentielle** (**E**), qui à un point $\mathbf{p} = (x, y, 1)^T$ associe une droite $\mathbf{l}' = (a', b', c')^T$, dans le repère "caméra" :

$$\begin{pmatrix} a' \\ b' \\ 1 \end{pmatrix} = \mathbf{E} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- **E** s'écrit :

$$\mathbf{E} = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = [\mathbf{t}]_\times \mathbf{R}$$

- 5 DOF : 3 rotations + 3 translations - ambiguité projective globale (quantité homogène) = 5

Propriétés de la matrice Essentielle

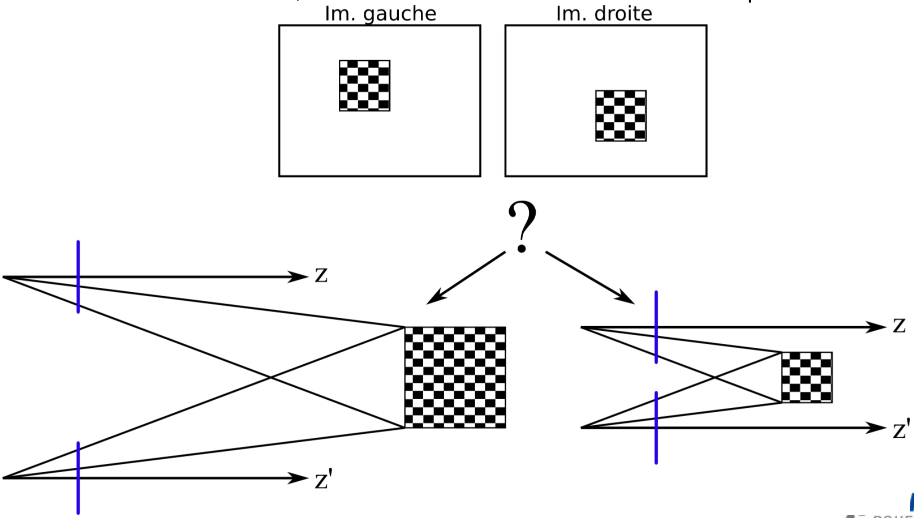
- Rang $= 2 \Rightarrow$ 2 valeurs singulières égales.
- Cette propriété permet l'extraction de $\mathbf{R}$ et $\mathbf{t}$ à partir de $\mathbf{E}$
- La décomposition SVD de $\mathbf{E}$ s'écrit

$$\mathbf{E} = \mathbf{U}.\mathrm{diag}(1, 1, 0)\mathbf{V}^T$$

- Le vecteur $\mathbf{t}$ sera connu à un facteur d'échelle près (ambiguité projective)

Concept d'ambiguité projective

Si la *baseline* est inconnue, l'observation est à un facteur d'échelle près.



Matrice Fondamentale - F

- La relation épipolaire peut aussi etre exprimée en coordonnées-image
- On utilise alors la **matrice Fondamentale** (**F**), qui associe un point à sa droite épipolaire :

$$\mathbf{l}' = \mathbf{F}\,\mathbf{m} \qquad\qquad \mathbf{l} = \mathbf{F}^T\,\mathbf{m}'$$

Matrice Fondamentale - F

- La relation épipolaire peut aussi etre exprimée en coordonnées-image
- On utilise alors la **matrice Fondamentale** (**F**), qui associe un point à sa droite épipolaire :

$$\mathbf{l}' = \mathbf{F}\,\mathbf{m} \qquad\qquad \mathbf{l} = \mathbf{F}^T\,\mathbf{m}'$$

- **F** et **E** sont liés par les paramètres intrinsèques (matrices **K** et **K**′) :

$$\mathbf{E} = \mathbf{K}'^T\,\mathbf{F}\,\mathbf{K} \qquad\qquad \mathbf{F} = \mathbf{K}'^{-T}\,\mathbf{E}\,\mathbf{K}^{-1}$$

- **F** peut-être déterminée uniquement à partir de correspondances entre points-images, sans connaissance des caméras.

Propriétés de la matrice Fondamentale

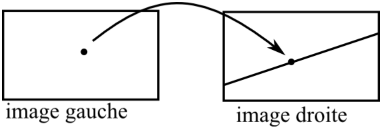
- Rang $= 2$
- 7 degrés de libertés : 9 coefficients, mais :
  - homogène, définie à un facteur d'échelle près
  - contrainte supplémentaire : $\det(F) = 0$
- Coordonnées des épipoles $\mathbf{e}$ et $\mathbf{e}'$:

$$\mathbf{Fe} = 0 \qquad\qquad \mathbf{e}'^T\mathbf{F} = 0$$

- Les coordonnées seront dans le **repère image**, et pourront être **hors** de l'image réelle, qui est de dimensions finie (u_{max}, v_{max}).
- Si les caméras sont alignées, les épipôles sont à **l'infini**.
  $\mathbf{e} = (1, 0, 0)^T$ et $\mathbf{e}' = (-1, 0, 0)^T$

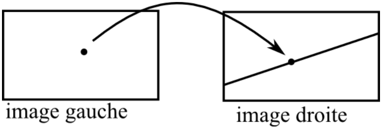
"Contrainte épipolaire"

- Tout les points stéréo-correspondants respectent cette contrainte.
- Peut s'exprimer de deux façons, selon qu'on se place dans le repère caméra ou dans le repère image.
  - En coordonnées image : $\mathbf{m'}^T.\mathbf{F}.\mathbf{m} = 0$
  - En coordonnées caméra : $\mathbf{p'}^T.\mathbf{E}.\mathbf{p} = 0$
- Interpretation : tout point droit correspondant à un point gauche doit se trouver sur la ligne épipolaire correspondante.



"Contrainte épipolaire"

- Tout les points stéréo-correspondants respectent cette contrainte.
- Peut s'exprimer de deux façons, selon qu'on se place dans le repère caméra ou dans le repère image.
  - En coordonnées image : $\mathbf{m'}^T.\mathbf{F}.\mathbf{m} = 0$
  - En coordonnées caméra : $\mathbf{p'}^T.\mathbf{E}.\mathbf{p} = 0$
- Interpretation : tout point droit correspondant à un point gauche doit se trouver sur la ligne épipolaire correspondante.



Attention

Pour deux points $\mathbf{m'}$ et $\mathbf{m}$, la valeur $v = \mathbf{m'}^T \cdot \mathbf{F} \cdot \mathbf{m}$ n'est **pas** une distance entre un des points et la droites épipolaire !

Sous-sommaire

Matrices de projection

- En prenant la caméra gauche comme référence, on construit la forme canonique des matrices de projection :

$$\mathbf{P} = \mathbf{K}\,[\mathbf{I} \mid \mathbf{0}] \qquad\qquad \mathbf{P}' = \mathbf{K}'\,[\mathbf{R} \mid \mathbf{t}]$$

$$\mathbf{P} = \mathbf{K}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad\qquad \mathbf{P}' = \mathbf{K}'\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{33} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}$$

- Les matrices de projections peuvent être reconstruites connaissant paramètres intrinsèques et géométrie inter-caméras.
- Il est nécessaire de les calculer pour faire de la **reconstruction 3D** (voir plus loin).

Cas particulier : configuration alignée

- Si les 3 angles de rotations sont nuls, et $t_y = t_z = 0$, alors la matrice Essentielle se réduit à :

$$\mathbf{E} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{pmatrix}$$

- Ces quantités étant homogènes, il est d'usage d'écrire alors :

$$\mathbf{E} = \mathbf{F} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

- La matrice de projection droite s'écrit alors :

$$\mathbf{P}' = \mathbf{K}' \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Sommaire

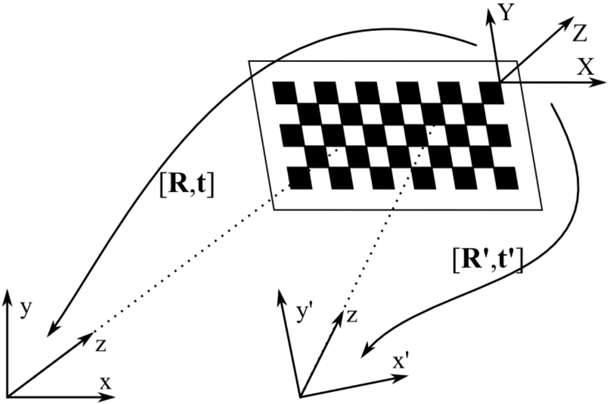
Sous-sommaire

Calibrage stéréoscopique

- Calibrer un système stéréo, c'est **estimer** sa géométrie épipolaire.
- 2 approches :
  - Calibrage **fort** : on calibre chaque caméra (K, R, T) avec une mire commune, et on déduit la relation inter-caméras.
  - Calibrage **faible** (*Weak calibration*) : on détermine uniquement la relation inter-caméras ("Géométrie épipolaire").
- Choix : selon l'application envisagée.
- Le calibrage faible est plus précis... mais observation à un facteur d'échelle près, si la *baseline* est inconnue.
- Suffisant pour de la reconnaissance de scènes (permet la rectification des images).

Calibrage fort

- On fait un calibrage simultané des 2 caméras avec la même mire.
- La matrice de rotation entre caméras va s'écrire $\mathbf{R}_0 = \mathbf{R}'^T\ \mathbf{R}$
- Le vecteur de translation entre caméras va s'écrire $\mathbf{t}_0 = \mathbf{t} - \mathbf{t}'$
- La matrice Essentielle va s'écrire $\mathbf{E} = [\mathbf{t_0}]_\times\ \mathbf{R}_0$



Sous-sommaire

Principe du calibrage faible

- On utilise un jeu de n paires de points stéréo-correspondants : (m, m')
- En développant l'expression $\mathbf{m'}^T\mathbf{Fm} = 0$:

$$f_{11}uu' + f_{12}vu' + f_{13}u' + f_{21}uv' + f_{22}vv' + f_{23}v' + f_{31}u + f_{32}v + f_{33} = 0$$

- Avec n paires de points, on peut construire le système d'équations :
  $\mathbf{A}_{nx9} \,.\, \mathbf{f}_{9x1} = 0$
  avec $f = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^T$
- **F** est homogène, on peut donc fixer une des valeurs à 1
  $\Rightarrow$ 8 inconnues

Résolution

- 8 paires de points donnent une solution unique.
- Inconvénient : **très** sensible au bruit !
- En pratique, on aura bien plus de points de correspondances. (jusqu'à plusieurs centaines)
- Solution au sens des moindres carrés. . .

Résolution

- 8 paires de points donnent une solution unique.
- Inconvénient : **très** sensible au bruit !
- En pratique, on aura bien plus de points de correspondances. (jusqu'à plusieurs centaines)
- Solution au sens des moindres carrés. . . Mais toujours très sensible au bruit ! ($\Rightarrow$ erreurs d'appariement)
- Les erreurs d'appariement (données aberrantes) vont complètement fausser le résultat !

Estimation robuste

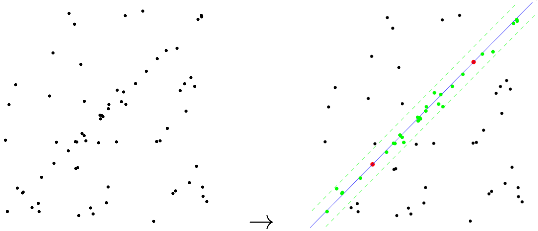
- L'appariement ne peut pas être juste à 100% $\Rightarrow$ il introduit obligatoirement des *outliers* (appariements erronés).
- Afin d'éliminer ces outliers, on utilise les techniques de l'**estimation statistique robuste**.
- Principe : ajuster des données à un modèle en éliminant les points qui s'écartent trop du modèle.
  - Implique une **métrique** adaptée.
  - Souvent basé sur des seuils absolus.

Estimation robuste

- L'appariement ne peut pas être juste à 100% $\Rightarrow$ il introduit obligatoirement des *outliers* (appariements erronés).
- Afin d'éliminer ces outliers, on utilise les techniques de l'**estimation statistique robuste**.
- Principe : ajuster des données à un modèle en éliminant les points qui s'écartent trop du modèle.
  - Implique une **métrique** adaptée.
  - Souvent basé sur des seuils absolus.
- Principaux algorithmes utilisés en CV :
  - LMEDS (*Least MEdian of Squares*) : basé sur la médiane, tolère un maximum de 50% d'outliers.
  - RANSAC (*RANdom SAmple Consensus*), et (très) nombreux dérivés (recherche active) : solution recommandée.

Exemple d'application de l'estimation robuste

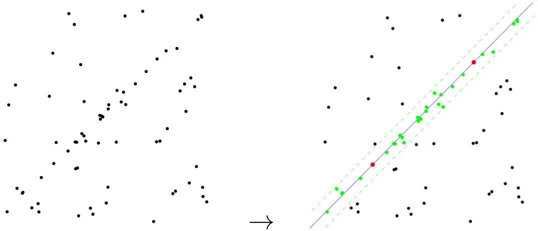
- Avec les points ci-dessous, trouver les paramètres de "la" droite.



  - *fitting* classique ("moindres carrés") $\Rightarrow$ solution abberante.
  - L'estimation robuste va éliminer les points qui ne rentrent pas dans le modèle "de la majorité" des points.

Exemple d'application de l'estimation robuste

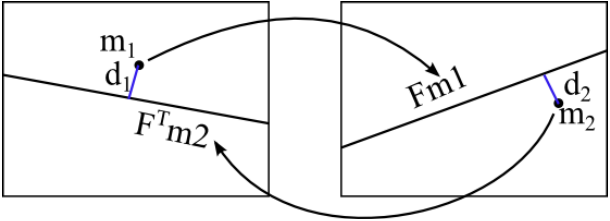
- Avec les points ci-dessous, trouver les paramètres de "la" droite.



  - *fitting* classique ("moindres carrés") ⇒ solution abberante.
  - L'estimation robuste va éliminer les points qui ne rentrent pas dans le modèle "de la majorité" des points.
- Principales étapes (algo. simplifié) :
  - Sélectionner un sous-ensemble minimal de points suffisant pour obtenir un modèle numérique.
  - Déterminer l'ensemble des n points qui "rentrent" dans le modèle trouvé (⇒ qui sont à une distance inf. à un seuil).
  - Répéter jusqu'à maximiser n, puis calculer l'estimation finale en utilisant tous ces points.

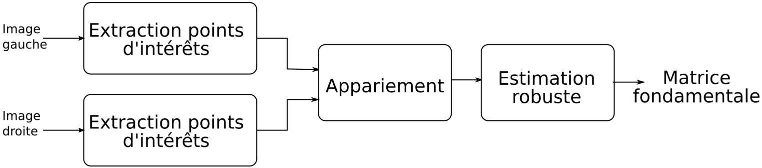
Estimation robuste de **F**

- Dans le cas de la stéréovision, il faut trouver les valeurs de **F** dont les droites épipolaires "collent" le plus aux points stéréo-correspondants fournis par l'étape de *matching*.
- On utilise l'algorithme "8-points" pour trouver un modèle à partir d'un sous-ensemble aléatoire de 8 paires.
- La métrique sera la **distance épipolaire** : moyenne des 2 distances entre droite épipolaire et point stéréo-correspondants : $d = d_1 + d_2$



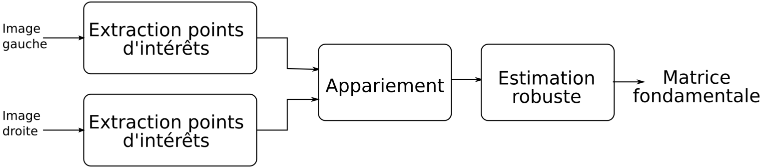
Sous-sommaire

Mise en œuvre de la calibration faible



- Solution généralement adoptée :
  - utiliser un descripteur local pour obtenir automatiquement un ensemble de points dans chaque image,
  - puis les apparier (algorithme des "plus proches voisins" avec critère d'ambiguité).

Mise en œuvre de la calibration faible

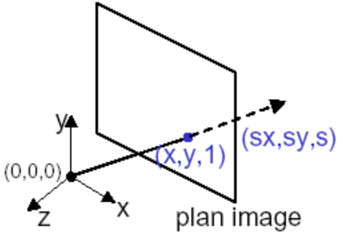


- Il faut une scène suffisament riche en contenu pour obtenir une caractérisation suffisante.
- La précision sera liée :
  - au nombre de points extraits,
    (pas assez de points ⇒ information imprécise
    trop de points ⇒ temps de calcul prohibitif)
  - a leur répartition géographique dans l'image,
    (si les points sont concentrés dans une zone, la géométrie ne sera précise que pour cette zone)
  - aux performances du descripteur local.

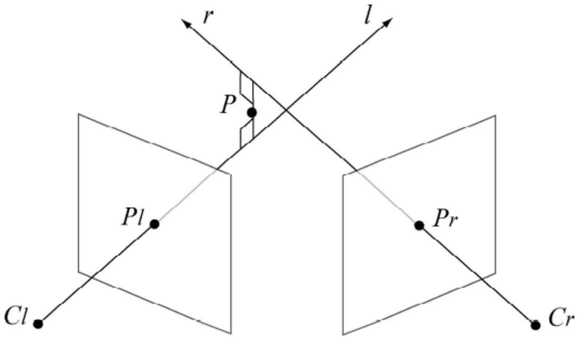
Sommaire

Reconstruction 3D

- Consiste à calculer le point 3D correspondant à 2 points 2D stéréocorrespondants.
- Necessite la connaissance complète du système :
  - Géométrie relative des caméras (matrice fondamentale **F**)
  - + paramètres intrinsèques $\mathbf{K}$ et $\mathbf{K}'$
  - = matrices de projection $\mathbf{P}$ et $\mathbf{P}'$
- Rappel (voir cours 1) : **un point 2D correspond à une droite 3D**



Reconstruction 3D par rétroprojection

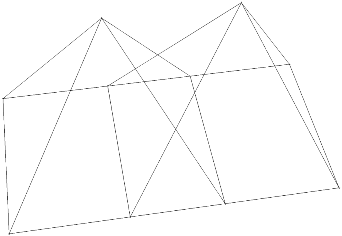


- Principe : connaissant **m** et **m**′, on cherche le point 3D correspondant, en calculant le point d'intersection des 2 droites.
- Problème : du au bruit sur les positions des points 2D et sur la calibration, ces 2 droites. . . ne se coupent pas !
- Solution généralement adoptée : on calcule le milieu du segment de droite à l'endroit ou les 2 droites sont les plus proches.

Exercice : erreur sur la profondeur

- Soit un système stéréo aligné dont l'appariement se fait au niveau pixel. Pour chaque appariement d'un pixel gauche avec un pixel droit, on aura une erreur dans l'intervalle $e = [0, 0.5]$ pixels.
- Exprimer l'erreur sur la profondeur ϵ_z en fonction de la disparité d et de l'erreur e.
- Exprimer l'erreur sur la profondeur ϵ_z pour une profondeur z_0 donnée, en considérant l'erreur maximale $e = 0,5$.
- A.N. $B = 30$cm, $\alpha = 1000$. Tracer l'allure de ϵ_z pour quelques valeurs de z_0 (1m, 10m, 100m).
- Conclusion ?
- Question subsidiaire : on souhaite limiter l'erreur à 10%. Jusqu'à quelle distance pourra-t-on mesurer avec une baseline de 20 cm / de 50 cm ?

Exercice : chevauchement d'image



- Pour que la stéréovision soit possible, il faut qu'il y ait un chevauchement minimal de la scène dans les images.
- On définit le ratio de chevauchement r comme la largeur en pixels de la partie commune (pour une profondeur z_0 donnée) par rapport à la largeur totale d'une image.
- Donner la condition sur β (FOV) et la baseline B pour qu'il y ait chevauchement ($r > 0$).
- Exprimer r en fonction de b, β et z_0