Robotique et Agents Autonomes (2014-2015)
Bases de la vision robotique

Emanuel Aldea <emanuel.aldea@u-psud.fr>
http://hebergement.u-psud.fr/emi

Département Systèmes Autonomes, IEF

2014-2015

(version du 19 décembre 2014)

Plan du cours

- Apprentissage, robotique et vision
- Généralités
- Invariance et points d'intérêt
- Analyse de mouvement
  - Les différents contextes
  - Notions de géométrie projective
  - Transformations 2D
  - Estimation robuste
  - Applications
- Stéréo et profondeur
- Navigation

Représentation des points 3D

Dans l'espace 3D :

$$\underbrace{\mathbf{p} = (X, Y, Z)^T = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}}_{\text{point initial}} \quad \underbrace{\mathbf{p}' = (X', Y', Z')^T = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix}}_{\text{point transformé}}$$

- six degrés de liberté (trois rotations, trois translations)
- on abandonne les ˜ pour simplifier les notations, mais les variables sont homogènes

Représentation des points 3D

Dans l'espace 3D :

$$\underbrace{\mathbf{p} = (X, Y, Z)^T = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}}_{\text{point initial}} \quad \underbrace{\mathbf{p}' = (X', Y', Z')^T = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix}}_{\text{point transformé}}$$

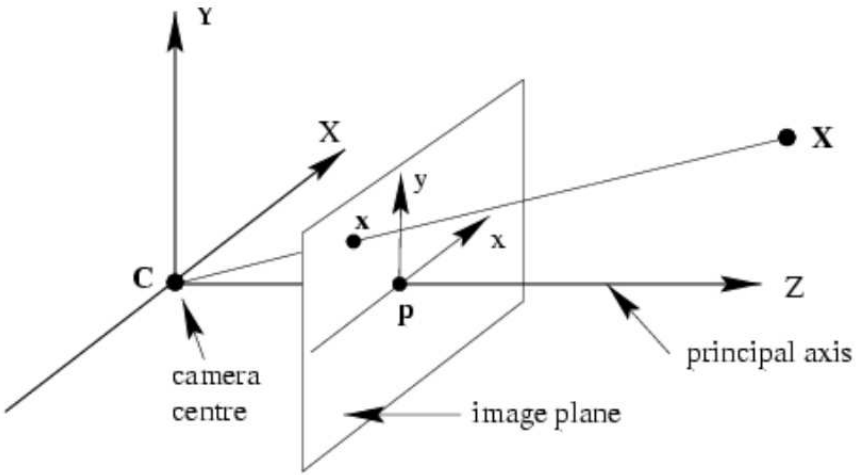
Une transformation Euclidienne avec les versions homogènes :

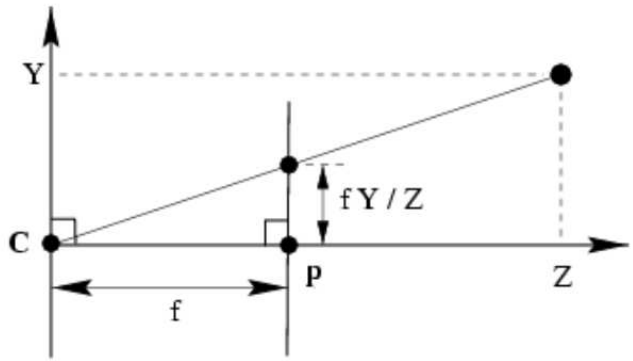
$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

ou bien $\mathbf{p}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{p}$, avec $\mathbf{R}^T\mathbf{R} = \mathbf{I}$, $\det \mathbf{R} = 1$

- six degrés de liberté (trois rotations, trois translations)
- on abandonne les ˜ pour simplifier les notations, mais les variables sont homogènes

Le modèle de camera pinhole





Projection 3D $\Rightarrow$ 2D par une projection centrale

- Dans le plan focal 3D : $(X, Y, Z)^T \Rightarrow (fX/Z, fY/Z, f)^T$
- Dans le plan 2D de l'image : $(X, Y, Z)^T \Rightarrow (fX/Z, fY/Z) = (x, y)$

Le modèle de camera pinhole

La projection dans le plan image $(fX/Z, fY/Z)$ en coordonnées homogènes donne :

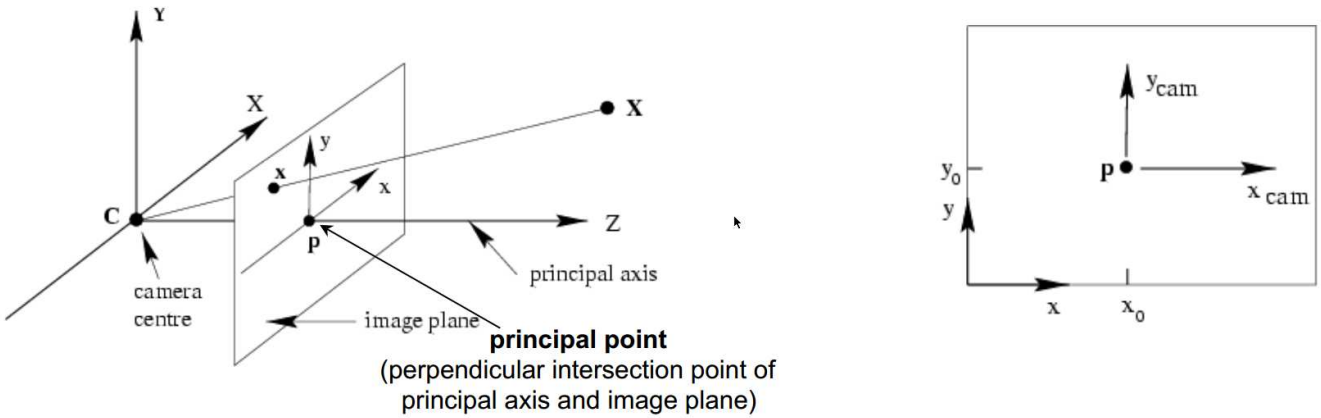
$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathrm{diag}(f, f, 1)[\mathbf{I}|\mathbf{0}]\mathbf{X}$$

Le modèle de camera pinhole

La projection dans le plan image $(fX/Z, fY/Z)$ en coordonnées homogènes donne :

$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathrm{diag}(f, f, 1)[\mathbf{I}|\mathbf{0}]\mathbf{X}$$

La référence choisie dans le plan image n'est pas la projection de l'axe optique :



Dans le système de référence qu'on utilise habituellement :

$$(X, Y, Z) \Rightarrow (fX/Z + p_x, fY/Z + p_y)$$

Le modèle de camera pinhole

La projection dans le plan image $(fX/Z, fY/Z)$ en coordonnées homogènes donne :

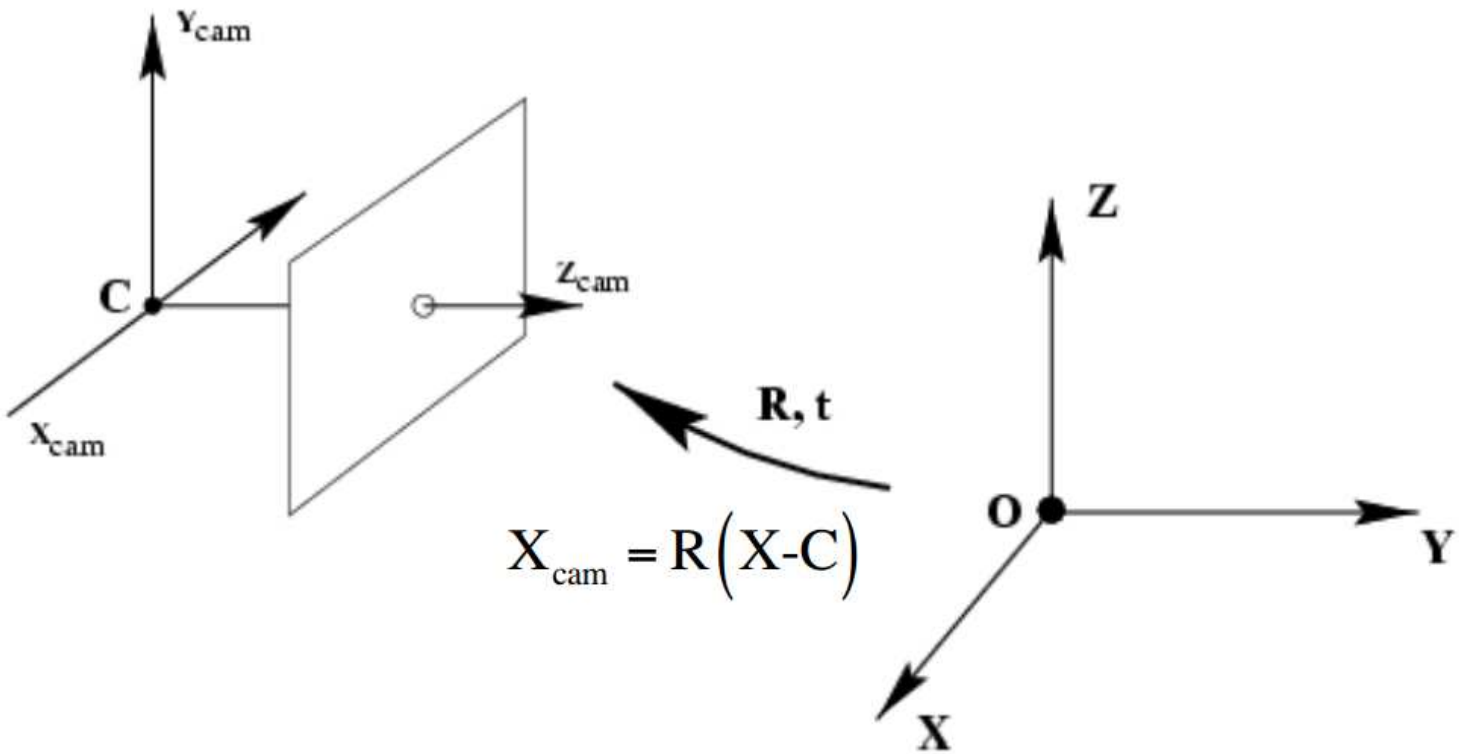
$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \cdot \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathrm{diag}(f, f, 1)[\mathbf{I}|\mathbf{0}]\mathbf{X}$$

K - matrice de calibration intrinsèque

- constante tant que l'optique de la camera n'est pas ajustée
- nécessaire pour faire le passage $2D \Leftrightarrow 3D$
- habituellement, estimée en utilisant des objets (mires) de calibration

Passage dans un repère inertiel (fixe)

Dernière étape de la modélisation : on exprime les variables dans un repère qui n'est pas solidaire de la camera et qui est fixe (typiquement pour la robotique mobile) :



Passage dans un repère inertiel (fixe)

Dernière étape de la modélisation : on exprime les variables dans un repère qui n'est pas solidaire de la camera et qui est fixe (typiquement pour la robotique mobile) : En notant par $\mathbf{C}$ le centre de la camera en coordonnées "world", le passage monde vers camera s'écrit

$$\mathbf{X}_{cam} = \mathbf{R}(\mathbf{X} - \mathbf{C}) = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ \mathbf{0}^T & 1 \end{bmatrix} \tilde{\mathbf{X}}$$

et donc à la place de la projection des coordonnées camera vers le plan image :

$$\mathbf{x} = \mathbf{K}\big[\mathbf{I}|\mathbf{0}\big]\mathbf{X}_{cam}$$

on utilise la projection des coordonnées world vers le plan image :

$$\mathbf{x} = \mathbf{KR}\big[\mathbf{I}|-\mathbf{C}\big]\mathbf{X}$$

Qu'est-ce qu'on peut déterminer là-dedans ?

Il y a d'autres problèmes à régler concernant la projection 3D $\Rightarrow$ 2D :

FIGURE: FOV étroit

Qu'est-ce qu'on peut déterminer là-dedans ?

Il y a d'autres problèmes à régler concernant la projection 3D $\Rightarrow$ 2D :

FIGURE: FOV moyen

Qu'est-ce qu'on peut déterminer là-dedans ?

Il y a d'autres problèmes à régler concernant la projection 3D $\Rightarrow$ 2D :

FIGURE: FOV large

Qu'est-ce qu'on peut déterminer là-dedans ?

Distorsions : visibles surtout pour des optique a FOV large, doivent être corrigées en s'appuyant sur des modèles de distorsion.

Stratégies pour déterminer les paramètres de la projection

Qu'est-ce qu'on peut déterminer là-dedans ?

Distorsions : visibles surtout pour des optique a FOV large, doivent être corrigées en s'appuyant sur des modèles de distorsion.

Stratégies pour déterminer les paramètres de la projection

- On utilise des objets avec des configurations connues, et on minimise une erreur de projection sur un nombre suffisant d'exemples :

Qu'est-ce qu'on peut déterminer là-dedans ?

Distorsions : visibles surtout pour des optique a FOV large, doivent être corrigées en s'appuyant sur des modèles de distorsion.

Stratégies pour déterminer les paramètres de la projection

- On utilise des objets avec des configurations connues, et on minimise une erreur de projection sur un nombre suffisant d'exemples :
- On peut utiliser directement la scène (auto-calibration)

- Z. Zhang, *A flexible new technique for camera calibration*, PAMI, 2000
- M Pollefeys, R Koch, L Van Gool, *Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters*, IJCV, 1999
- A. Fusiello, *Uncalibrated Euclidean reconstruction : a review*, Image and Vision Computing , 2000

Stéréo

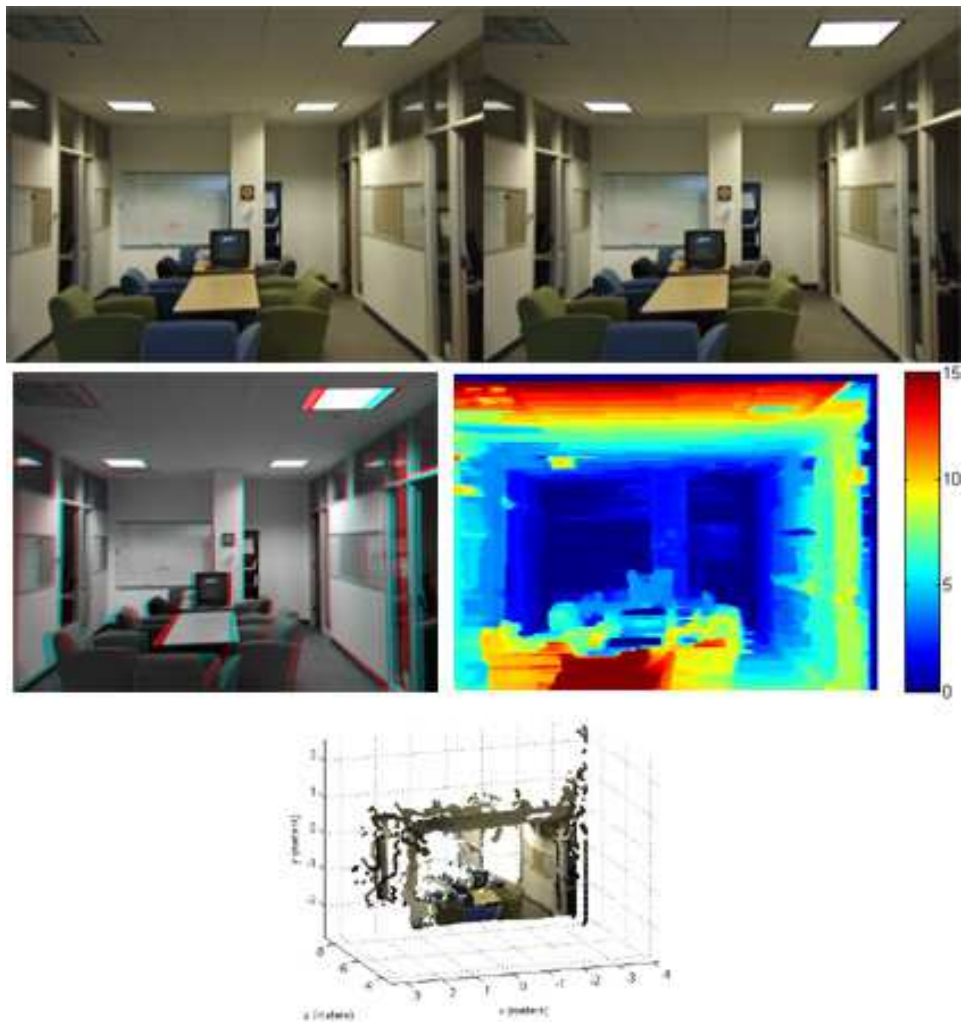
- Avec la projection, on perd l'information de profondeur.

Stéréo

- Avec la projection, on perd l'information de profondeur.
- Mais avec deux vues, on peut remonter à la valeur de Z

Stéréo

- Avec la projection, on perd l'information de profondeur.
- Mais avec deux vues, on peut remonter à la valeur de Z



Stéréo

- Avec la projection, on perd l'information de profondeur.
- Mais avec deux vues, on peut remonter à la valeur de Z

Pour simplifier, supposons que la rotation entre les deux cameras est nulle, et que la translation est purement latérale. En prenant b comme la distance entre les cameras, on obtient rapidement :

$$x_1 = \frac{fX_1}{Z};\quad y_1 = \frac{fY_1}{Z};\quad x_2 = \frac{fX_2}{Z};\quad y_2 = \frac{fY_2}{Z};\quad Y_1 = Y_2;\quad X_1 = X_2 + b$$

D'ici, on obtient deux résultats fondamentaux : $y_1 = y_2$ (les points 3D de la scène se projettent sur des lignes correspondantes) et en notant $d = x_2 - x_1$ la **disparité** :

$$\boxed{d = \frac{fb}{Z} \Leftrightarrow Z = \frac{fb}{d}}$$

Estimation de l'imprécision

Un calcul de propagation de l'erreur de disparité ϵ_d nous montre que :

$$\epsilon_Z \approx \frac{Z^2}{bf} \cdot \epsilon_d$$

Estimation de l'imprécision

Un calcul de propagation de l'erreur de disparité ϵ_d nous montre que :

$$\epsilon_Z \approx \frac{Z^2}{bf} \cdot \epsilon_d$$

Problème : la fonction d'erreur d'estimation de Z est quadratique en Z. Par exemple, si on veut faire de la reconstruction avec un dispositif de la taille d'un téléphone :

- $b = 8cm$, $f = 750px$, $\epsilon_d \approx 2px$
- on obtient $\epsilon_Z \approx 13cm$ pour $Z = 2m$ et $\epsilon_Z \approx 83cm$ pour $Z = 5m$

Solutions rapides pour un gain limité en précision :

Estimation de l'imprécision

Un calcul de propagation de l'erreur de disparité ϵ_d nous montre que :

$$\epsilon_Z \approx \frac{Z^2}{bf} \cdot \epsilon_d$$

Problème : la fonction d'erreur d'estimation de Z est quadratique en Z. Par exemple, si on veut faire de la reconstruction avec un dispositif de la taille d'un téléphone :

- $b = 8cm$, $f = 750px$, $\epsilon_d \approx 2px$
- on obtient $\epsilon_Z \approx 13cm$ pour $Z = 2m$ et $\epsilon_Z \approx 83cm$ pour $Z = 5m$

Solutions rapides pour un gain limité en précision :

- augmenter f ... mais on réduit le FOV de travail ; en pratique en navigation robotique on fait exactement le contraire (FOV $\geq$ 135 deg)

Estimation de l'imprécision

Un calcul de propagation de l'erreur de disparité ϵ_d nous montre que :

$$\epsilon_Z \approx \frac{Z^2}{bf} \cdot \epsilon_d$$

Problème : la fonction d'erreur d'estimation de Z est quadratique en Z. Par exemple, si on veut faire de la reconstruction avec un dispositif de la taille d'un téléphone :

- $b = 8cm$, $f = 750px$, $\epsilon_d \approx 2px$
- on obtient $\epsilon_Z \approx 13cm$ pour $Z = 2m$ et $\epsilon_Z \approx 83cm$ pour $Z = 5m$

Solutions rapides pour un gain limité en précision :

- augmenter f ... mais on réduit le FOV de travail ; en pratique en navigation robotique on fait exactement le contraire (FOV $\geq$ 135 deg)
- augmenter b ... mais c'est peu pratique au delà d'une certaine valeur (10cm pour un outil de poche, 1m pour une voiture etc.) et cela réduit le FOV commun à de petites distances

Les données d'entrée

On a besoin de ϵ_d en entrée de l'algorithme, mais cette disparité peut être estimée directement uniquement pour des points d'intérêt :

- on peut faire de la reconstruction 3D sparse

Les données d'entrée

On a besoin de ϵ_d en entrée de l'algorithme, mais cette disparité peut être estimée directement uniquement pour des points d'intérêt :

- on peut faire de la reconstruction 3D sparse
- on peut utiliser des méthodes d'optimisation pour associer tous les pixels d'une image à l'autre en minimisant la différence de couleur entre les pixels correspondants et en régularisant le champ de disparité résultant

Voir `http://vision.middlebury.edu/stereo/eval/` pour une comparaison très large des méthodes existantes

Les données d'entrée

On a besoin de ϵ_d en entrée de l'algorithme, mais cette disparité peut être estimée directement uniquement pour des points d'intérêt :

- on peut faire de la reconstruction 3D sparse
- on peut utiliser des méthodes d'optimisation pour associer tous les pixels d'une image à l'autre en minimisant la différence de couleur entre les pixels correspondants et en régularisant le champ de disparité résultant
- mais l'imprécision sera supérieure aux valeurs vues précédemment, et les points qui sont aux bords des objets peuvent ne pas avoir des correspondants

Les données d'entrée

On a besoin de ϵ_d en entrée de l'algorithme, mais cette disparité peut être estimée directement uniquement pour des points d'intérêt :

- on peut faire de la reconstruction 3D sparse
- on peut utiliser des méthodes d'optimisation pour associer tous les pixels d'une image à l'autre en minimisant la différence de couleur entre les pixels correspondants et en régularisant le champ de disparité résultant
- mais l'imprécision sera supérieure aux valeurs vues précédemment, et les points qui sont aux bords des objets peuvent ne pas avoir des correspondants

On a également besoin de b. En pratique, pour des paires stéréo, on estime précisément $\mathbf{C}$ et $\mathbf{R}$, qui sont proches d'un vecteur aligné à l'axe X et l'identité respectivement, mais pas identiques.

Les données d'entrée

On a besoin de ϵ_d en entrée de l'algorithme, mais cette disparité peut être estimée directement uniquement pour des points d'intérêt :

- on peut faire de la reconstruction 3D sparse
- on peut utiliser des méthodes d'optimisation pour associer tous les pixels d'une image à l'autre en minimisant la différence de couleur entre les pixels correspondants et en régularisant le champ de disparité résultant
- mais l'imprécision sera supérieure aux valeurs vues précédemment, et les points qui sont aux bords des objets peuvent ne pas avoir des correspondants

On a également besoin de b. En pratique, pour des paires stéréo, on estime précisément $\mathbf{C}$ et $\mathbf{R}$, qui sont proches d'un vecteur aligné à l'axe X et l'identité respectivement, mais pas identiques.
Nouveau problème : dans ce cas, le rayons qui remontent au point 3D par ses deux projections ne s'intersectent pas forcement $\Rightarrow$ on utilise des algorithmes de triangulation qui cherchent une solution optimale par rapport à un certain critère, i.e. le point en 3D se trouvant à mi-chemin entre les deux rayons.

Les données d'entrée

On a besoin de ϵ_d en entrée de l'algorithme, mais cette disparité peut être estimée directement uniquement pour des points d'intérêt :

- on peut faire de la reconstruction 3D sparse
- on peut utiliser des méthodes d'optimisation pour associer tous les pixels d'une image à l'autre en minimisant la différence de couleur entre les pixels correspondants et en régularisant le champ de disparité résultant
- mais l'imprécision sera supérieure aux valeurs vues précédemment, et les points qui sont aux bords des objets peuvent ne pas avoir des correspondants

On a également besoin de b. En pratique, pour des paires stéréo, on estime précisément $\mathbf{C}$ et $\mathbf{R}$, qui sont proches d'un vecteur aligné à l'axe X et l'identité respectivement, mais pas identiques.

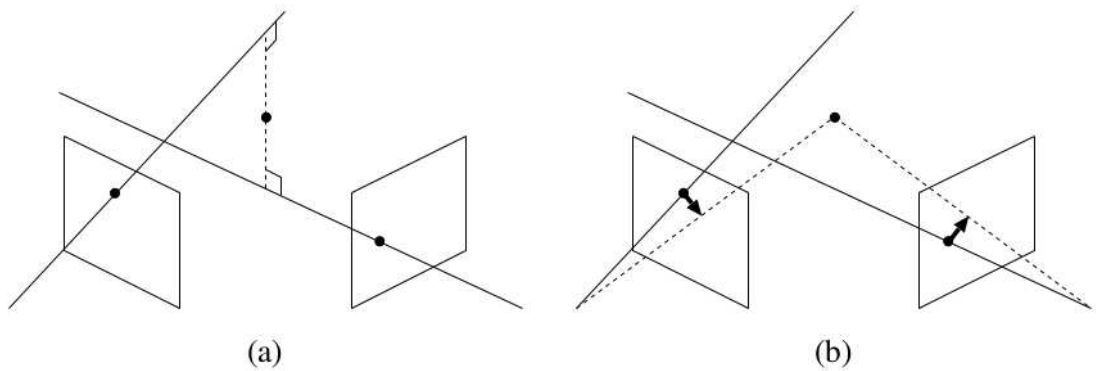


Figure 1: Triangulation. (a) The mid-point method. (b) Optimal correction.

Les données d'entrée

On a besoin de ϵ_d en entrée de l'algorithme, mais cette disparité peut être estimée directement uniquement pour des points d'intérêt :

- on peut faire de la reconstruction 3D sparse
- on peut utiliser des méthodes d'optimisation pour associer tous les pixels d'une image à l'autre en minimisant la différence de couleur entre les pixels correspondants et en régularisant le champ de disparité résultant
- mais l'imprécision sera supérieure aux valeurs vues précédemment, et les points qui sont aux bords des objets peuvent ne pas avoir des correspondants

On a également besoin de b. En pratique, pour des paires stéréo, on estime précisément $\mathbf{C}$ et $\mathbf{R}$, qui sont proches d'un vecteur aligné à l'axe X et l'identité respectivement, mais pas identiques.

- Richard I. Hartley, and Peter F. Sturm., *Triangulation*, CVIU, 1997
- Kenichi Kanatani, Yasuyuki Sugaya and Hirotaka Niitsuma, *Triangulation from Two Views Revisited : Hartley-Sturm vs. Optimal Correction*, BMVC, 2008

Vers le SLAM

Avec deux vues de la scène 3D, nous sommes très limités en précision.

Deux besoins majeurs en robotique

- se localiser dans un environnement inconnu, mais on ne veut pas forcement une cartographie précise de l'environnement

Vers le SLAM

Avec deux vues de la scène 3D, nous sommes très limités en précision.

Deux besoins majeurs en robotique

- se localiser dans un environnement inconnu, mais on ne veut pas forcement une cartographie précise de l'environnement
- obtenir une reconstruction 3D précise, sans s'intéresser à la manière dont elle a été obtenue

Vers le SLAM

Avec deux vues de la scène 3D, nous sommes très limités en précision.

Deux besoins majeurs en robotique

- se localiser dans un environnement inconnu, mais on ne veut pas forcement une cartographie précise de l'environnement
- obtenir une reconstruction 3D précise, sans s'intéresser à la manière dont elle a été obtenue

SLAM

- la manière la plus robuste de répondre à une de ces deux questions est de traiter les deux au même temps

Vers le SLAM

Avec deux vues de la scène 3D, nous sommes très limités en précision.

Deux besoins majeurs en robotique

- se localiser dans un environnement inconnu, mais on ne veut pas forcement une cartographie précise de l'environnement
- obtenir une reconstruction 3D précise, sans s'intéresser à la manière dont elle a été obtenue

SLAM

- la manière la plus robuste de répondre à une de ces deux questions est de traiter les deux au même temps
- quand le robot est mobile, **C** et **R** deviennent inconnus au bout de chaque petit déplacement ...

Vers le SLAM

Avec deux vues de la scène 3D, nous sommes très limités en précision.

Deux besoins majeurs en robotique

- se localiser dans un environnement inconnu, mais on ne veut pas forcement une cartographie précise de l'environnement
- obtenir une reconstruction 3D précise, sans s'intéresser à la manière dont elle a été obtenue

SLAM

- la manière la plus robuste de répondre à une de ces deux questions est de traiter les deux au même temps
- quand le robot est mobile, **C** et **R** deviennent inconnus au bout de chaque petit déplacement ...
- mais on peut les estimer en s'appuyant sur un ensemble existant de points 3D cartographiés...

Vers le SLAM

Avec deux vues de la scène 3D, nous sommes très limités en précision.

Deux besoins majeurs en robotique

- se localiser dans un environnement inconnu, mais on ne veut pas forcement une cartographie précise de l'environnement
- obtenir une reconstruction 3D précise, sans s'intéresser à la manière dont elle a été obtenue

SLAM

- la manière la plus robuste de répondre à une de ces deux questions est de traiter les deux au même temps
- quand le robot est mobile, **C** et **R** deviennent inconnus au bout de chaque petit déplacement ...
- mais on peut les estimer en s'appuyant sur un ensemble existant de points 3D cartographiés...
- et par la suite ajouter de nouvelles observations pour affiner la position 3D de ces points 3D

Vers le SLAM

Avec deux vues de la scène 3D, nous sommes très limités en précision.

Deux besoins majeurs en robotique

- se localiser dans un environnement inconnu, mais on ne veut pas forcement une cartographie précise de l'environnement
- obtenir une reconstruction 3D précise, sans s'intéresser à la manière dont elle a été obtenue

SLAM

- la manière la plus robuste de répondre à une de ces deux questions est de traiter les deux au même temps
- quand le robot est mobile, **C** et **R** deviennent inconnus au bout de chaque petit déplacement ...
- mais on peut les estimer en s'appuyant sur un ensemble existant de points 3D cartographiés...
- et par la suite ajouter de nouvelles observations pour affiner la position 3D de ces points 3D
- et initialiser d'autres nouveaux points

Le cadre formel

Au moment t, si on associe la pose de la camera à une variable $\mathbf{s}_t$ et si on note par Z_t l'ensemble des observations et par $\mathbf{X}$ la cartographie existante, l'objectif du SLAM est d'estimer la distribution :

$$P(\mathbf{s}_t, \mathbf{X} | \mathbf{Z}_{0:t})$$

La solution est approchée de manière récursive :

- on considère qu'on dispose d'une estimation pour $P(\mathbf{s}_{t-1}, \mathbf{X} | \mathbf{Z}_{0:t-1})$

Le cadre formel

Au moment t, si on associe la pose de la camera à une variable $\mathbf{s}_t$ et si on note par Z_t l'ensemble des observations et par $\mathbf{X}$ la cartographie existante, l'objectif du SLAM est d'estimer la distribution :

$$P(\mathbf{s}_t, \mathbf{X} | \mathbf{Z}_{0:t})$$

La solution est approchée de manière récursive :

- on considère qu'on dispose d'une estimation pour $P(\mathbf{s}_{t-1}, \mathbf{X} | \mathbf{Z}_{0:t-1})$
- on dispose d'un modèle d'observation $P(\mathbf{z}_t | \mathbf{s}_t, \mathbf{X})$: la probabilité de faire une certaine observation quand la pose ainsi que la cartographie sont connues

Le cadre formel

Au moment t, si on associe la pose de la camera à une variable $\mathbf{s}_t$ et si on note par Z_t l'ensemble des observations et par $\mathbf{X}$ la cartographie existante, l'objectif du SLAM est d'estimer la distribution :

$$P(\mathbf{s}_t, \mathbf{X} | \mathbf{Z}_{0:t})$$

La solution est approchée de manière récursive :

- on considère qu'on dispose d'une estimation pour $P(\mathbf{s}_{t-1}, \mathbf{X} | \mathbf{Z}_{0:t-1})$
- on dispose d'un modèle d'observation $P(\mathbf{z}_t | \mathbf{s}_t, \mathbf{X})$: la probabilité de faire une certaine observation quand la pose ainsi que la cartographie sont connues
- on dispose d'un modèle de transition $P(\mathbf{s}_t | \mathbf{s}_{t-1})$, indépendant des observations et de la cartographie

La mise à jour

Le modèle de transition est utilisé pour estimer la pose courante (time update) :

$$P(\mathbf{s}_t, \mathbf{X}|\mathbf{s}_{t-1}, \mathbf{Z}_{0:t-1}) = \int P(\mathbf{s}_t|\mathbf{s}_{t-1})\ P(\mathbf{s}_{t-1}, \mathbf{X}|\mathbf{Z}_{0:t-1}) d\mathbf{s}_{t-1}$$

Dans une deuxième étape, on applique une correction qui prend en compte les observations courantes (measurement update) :

$$P(\mathbf{s}_t, \mathbf{X}|\mathbf{s}_{t-1}, \mathbf{Z}_{0:t}) = \frac{P(\mathbf{z}_t|\mathbf{s}_t, \mathbf{X})\ P(\mathbf{s}_t, \mathbf{X}|\mathbf{s}_{t-1}, \mathbf{Z}_{0:t-1})}{P(\mathbf{z}_t)}$$

Pour EKF-SLAM :

- dynamique non-linéaire, bruit Gaussien pour les observations :

$$P(\mathbf{s}_t, \mathbf{X}|\mathbf{Z}_{0:t}) : \boldsymbol{\mu}_{\mathbf{s}_t,\mathbf{X}}, \boldsymbol{\Sigma}_{\mathbf{s}_t,\mathbf{X}}$$

- en pratique, la linéarisation mène à l'apparition d'inconsistances dans la cartographie
- complexité quadratique par rapport au nombre de points de la cartographie $O(M^2)$

Optimisation globale - étape préliminaire

Depuis que la puissance de calcul est plus accessible dans les systèmes autonomes, on préfère une approche où on minimise par rapport à toutes les observations disponibles un coût :

$$u(\mathbf{s}_t, \mathbf{X}, \mathbf{z}_t) = \frac{1}{M} \sum_{j=1}^{M} e(\mathbf{s}_t, x_{j,t}, z_{j,t})^T \; e(\mathbf{s}_t, x_{j,t}, z_{j,t})$$

où la fonction d'erreur est naturellement :

$$e(\mathbf{s}_t, x_{j,t}, z_{j,t}) = z_{j,t} - g(\mathbf{s}_t, x_{j,t})$$

$z_{j,t}$ étant la mesure 2D et $g(\mathbf{s}_t, x_{j,t})$ étant la projection de l'élément $x_{j,t}$ de la cartographie pour la pose $\mathbf{s}_t$.
Dans un premier pas, on va juste approcher la nouvelle pose :

$$\hat{\mathbf{s}}_t = \arg\min_{\mathbf{s}_t} u(\mathbf{s}_t, \mathbf{X}, \mathbf{z}_t)$$

Observation : l'estimation de pose ne modifie pas $\mathbf{X}$

Optimisation globale - étape finale

On calcul le MAP pour le nombre maximal d'estimations préliminaires et d'observations au moment respectif :

$$u_{BA}(\mathbf{S}_{0:t}, \mathbf{X}, \mathbf{Z}_{0:t}) = \frac{1}{M(t+1)} \sum_{\tau=0}^{T} \sum_{j=1}^{M} e(\mathbf{s}_\tau, x_{j,\tau}, z_{j,\tau})^T \; e(\mathbf{s}_\tau, x_{j,\tau}, z_{j,\tau})$$

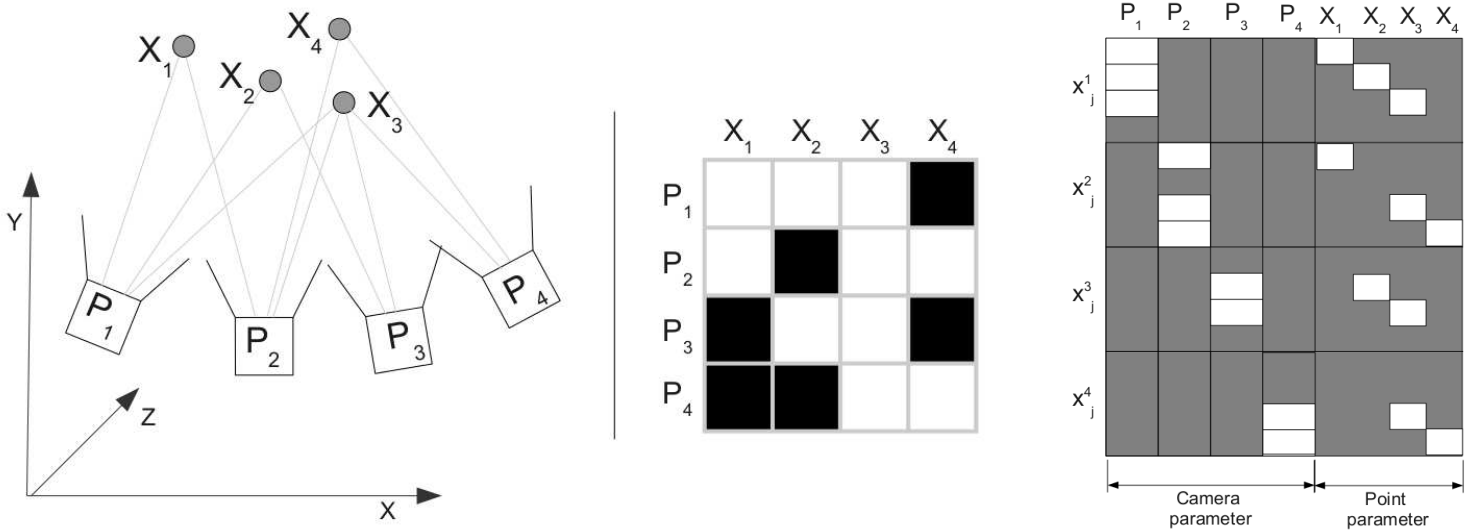
$$(\mathbf{J}^T\mathbf{J} + \lambda \,\mathrm{diag}(\mathbf{J}^T\mathbf{J}))\boldsymbol{\Delta} = -\mathbf{J}^T e$$

La solution qu'on cherche cette fois est composée par :

$$\hat{\mathbf{S}}_{0:t}, \hat{\mathbf{X}} = \underset{\mathbf{S}_{0:t}, \mathbf{X}}{\arg\min}\, u_{BA}(\mathbf{S}_{0:t}, \mathbf{X}, \mathbf{Z}_{0:t})$$

Complexité de cet algorithme une fois qu'on exploite la parcimonie de son Jacobien : $O(N^3 + MN^2)$, très intéressant puisque $M \gg N$.

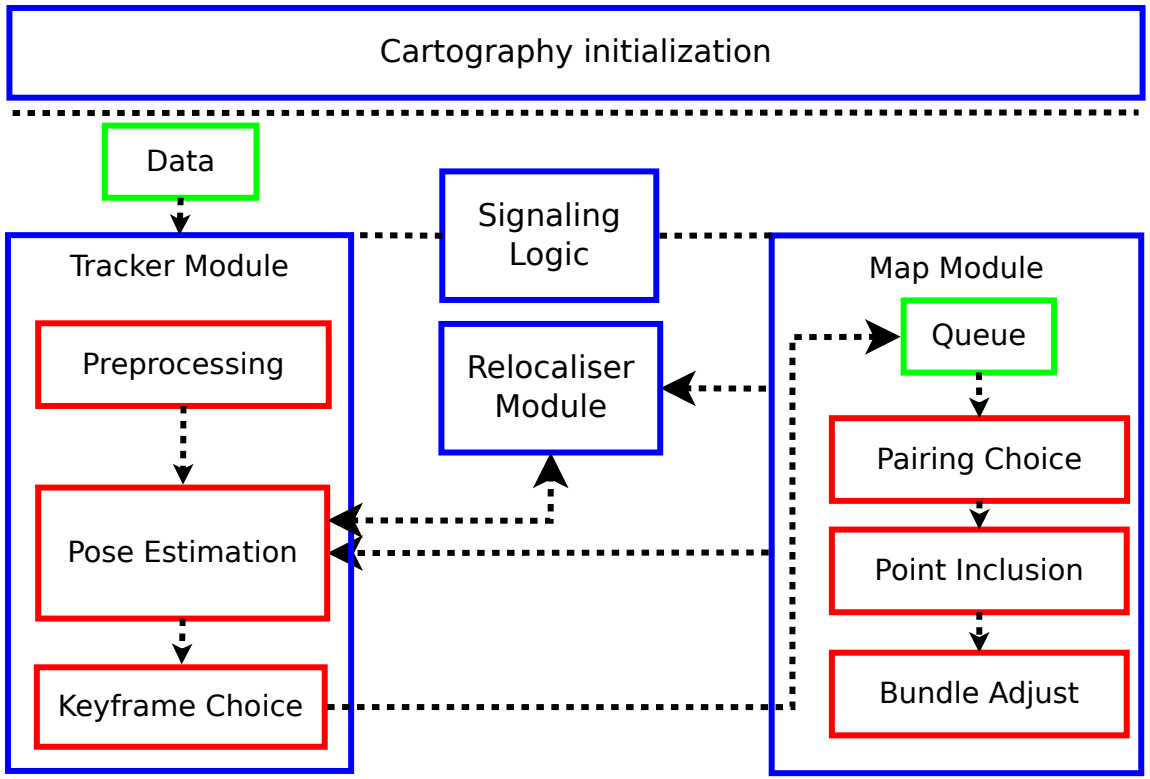
SLAM en temps réel



Un exemple de configuration : 5207 points 3D , 54 poses, 24609 projections, 15945 variables, 21 it., 7.99 sec.
Pas suffisamment rapide !

- Sélection de key-frames
- Exécution parallèle de tracking et BA
- Limiter le nombre d'itérations
- Local Bundle Adjustment

Une vue générale de l'architecture



- bénéficie pleinement d'un système multi-processor
- généralement très complexe et difficile à mettre en œuvre en assurant une navigation robuste et en évitant de surcharger le système

Références SLAM

Introduction et revue des méthodes

- Hugh Durrant-Whyte, Tim Bailey, *Simultaneous localization and mapping : Part I*, IEEE Robotics & Automation Magazine, 2006
- Tim Bailey, Hugh Durrant-Whyte, *Simultaneous localization and mapping (SLAM) : Part II*, IEEE Robotics & Automation Magazine, 2006
- S Thrun, W Burgard, D Fox, *Probabilistic robotics*, MIT Press, 2005
- le site `http://www.openslam.org/`
- Algèbre, Jacobiens : Juan Solà, *3D Algebra for Vision Systems in Robotics*, 2006

Pour aller plus en détail

- Tutoriel EKF-SLAM en 2D + code Matlab : ici
- Analyse BA : Bill Triggs, Philip Mclauchlan, Richard Hartley, Andrew Fitzgibbon, *Bundle Adjustment - A Modern Synthesis*, Vision algorithms : theory and practice, 2000
- Parcimonie du Jacobien BA : M.I. A. Lourakis and A.A. Argyros, *SBA : A Software Package for Generic Sparse Bundle Adjustment*, ACM Trans. Math. Software, 2009
- Yekeun Jeong, David Nistér, Drew Steedly, Richard Szeliski, In-So Kweon, *Pushing the Envelope of Modern Methods for Bundle Adjustment*, PAMI, 2012
- BA temps réel : G Klein, D Murray, *Parallel tracking and mapping for small AR workspaces*, ISMAR, 2007