INSTITUT DES SCIENCES ET TECHNIQUES DES YVELINES

UVSQ université PARIS-SACLAY

ISTY
Institut des Sciences et Techniques des Yvelines
CAMPUS DE MANTES EN YVELINES
CAMPUS DE SAINT-QUENTIN-EN-YVELINES

Capteurs pour Véhicules Autonomes

Patrick J Bonnin

Un Certain Nombre de Véhicules

Carnegie Mellon Univiersity:

- ✓ Grand, puis Urban Challange DARPA

Boss, l'engin de l'équipe de Carnegie Mellon au Urban Challenge (2007) de la Darpa. Bardé d'énormes capteurs, dont le fameux lidar Velodyne. ©Velodyne

Un Certain Nombre de Véhicules

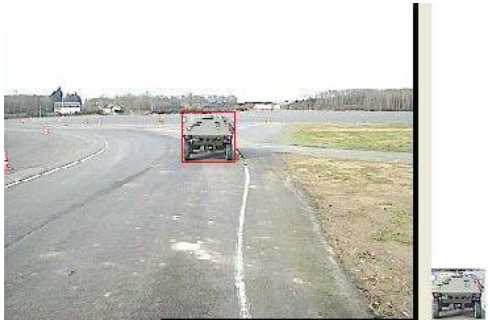
- **Institut Français des Sciences et Technologies des Transports et Aménagement Réseau:**
  - ✓ **Université Gustave Eiffel**

Grand Challenge DARPA

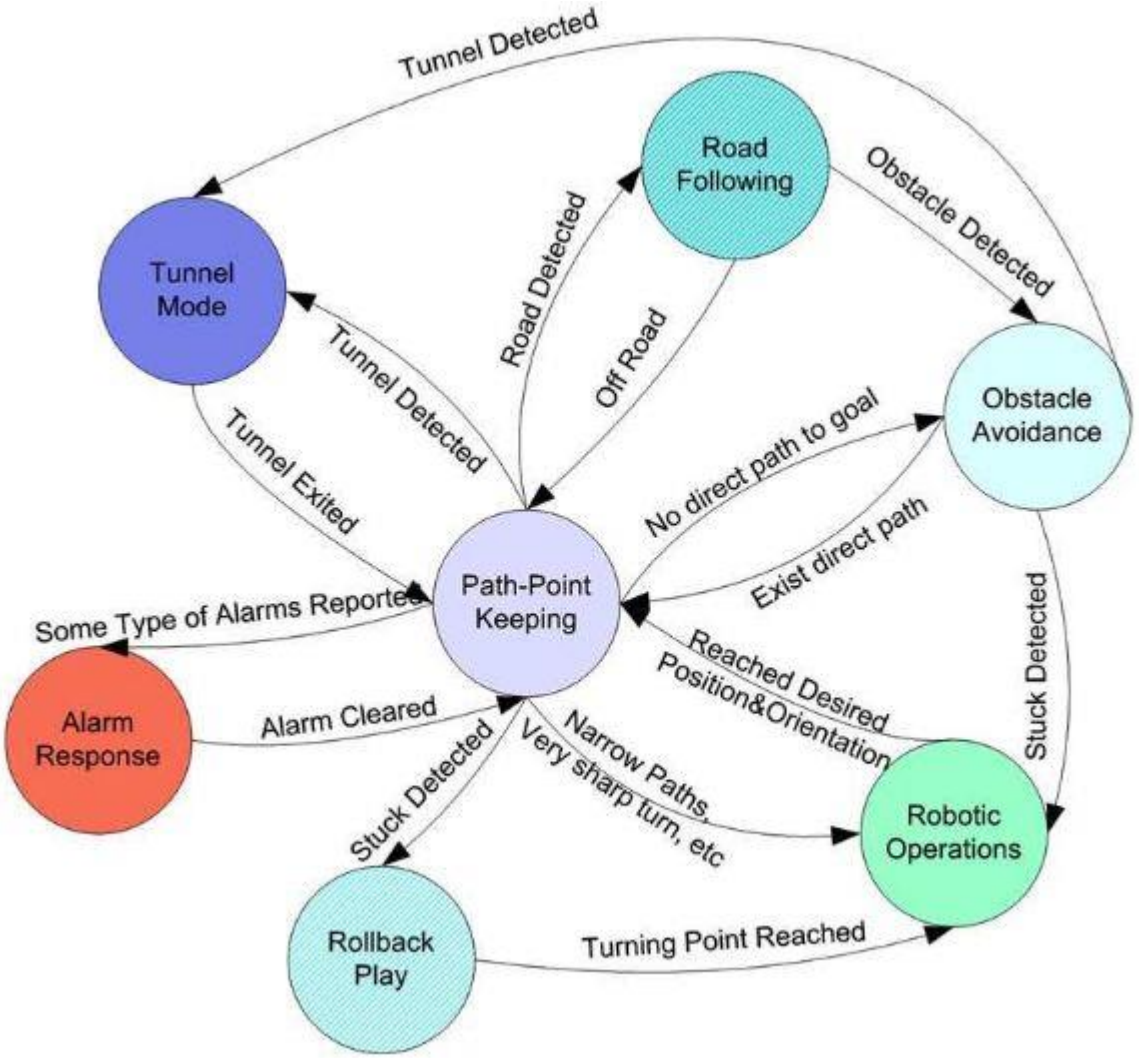
- **Véhicules Autonomes :**
  - ✓ **Editions 2004 et 2005 ;**
  - ✓ **Off Road;**
  - ✓ **Militaire;**

Un Certain Nombre de Véhicules Militaires

- **Transport pour Militaires :**
  - ✓ **R-Trooper : Thalès : PEA Tarot DGA : Suivi de Véhicules, Ralliement d'Amers : Autonomie Supervisée ;**
  - ✓ **Concurrent allemand …Velodyne …**



Partie Intelligente : Comportements : Grand Challenge Darpa



Divers Capteurs

- **Un certain nombre de capteurs :**
  - ✓ **De différentes natures ;**
  - ✓ **Lidar, Radar, Caméras Visible, IR, US …**

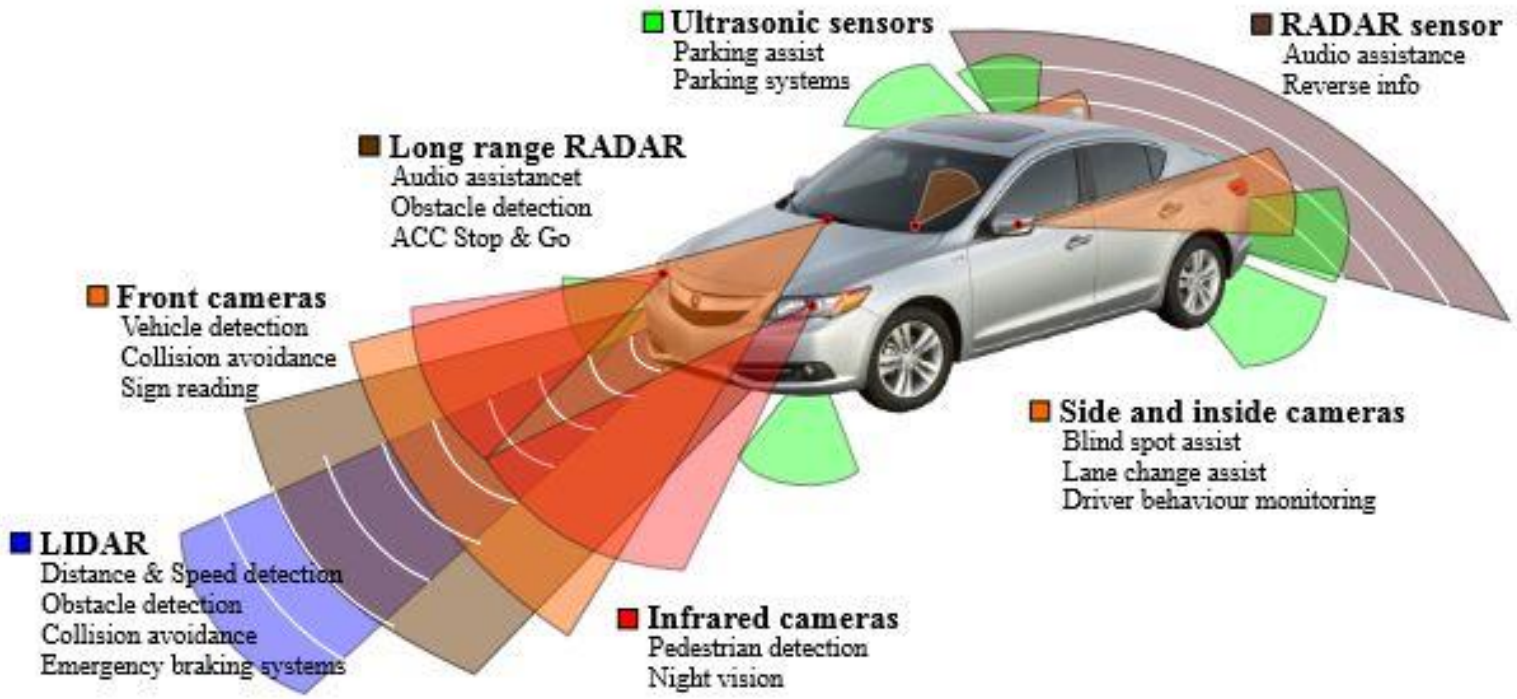
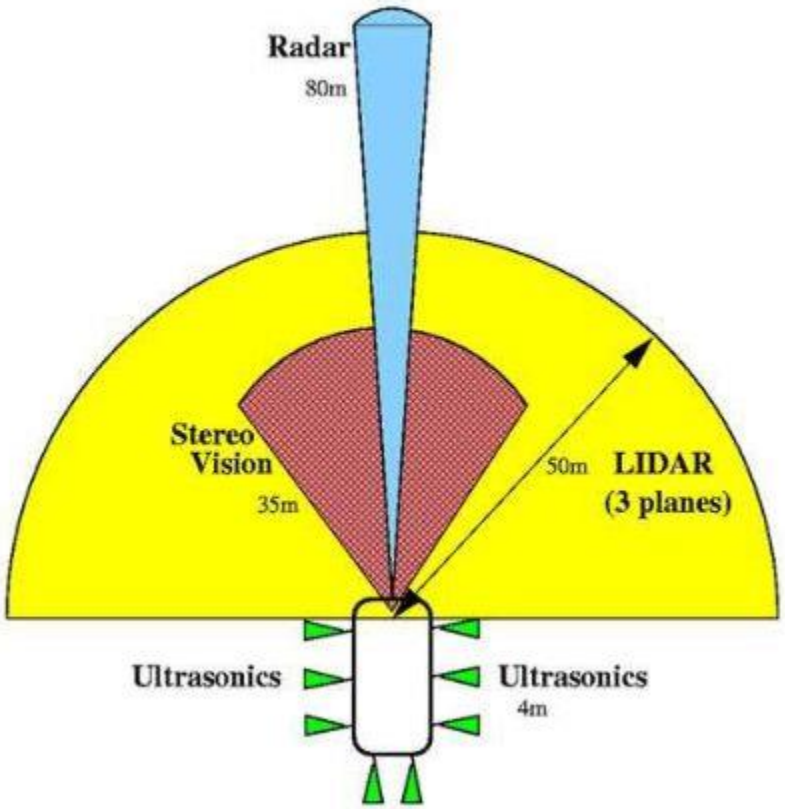


Figure 1.1: Typical types of sensors for ADAS.

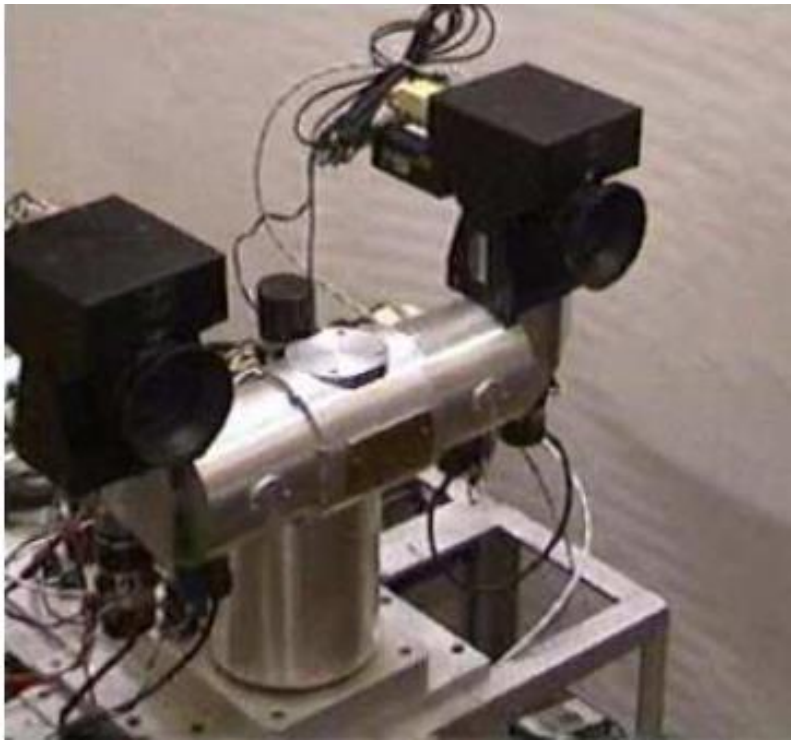
Capteurs Grand Challenge

- **Un certain nombre de capteurs : les mêmes**
  - ✓ **De différentes natures : complémentarité ;**
  - ✓ **Lidar, Radar, Caméras Visible, IR, US …**



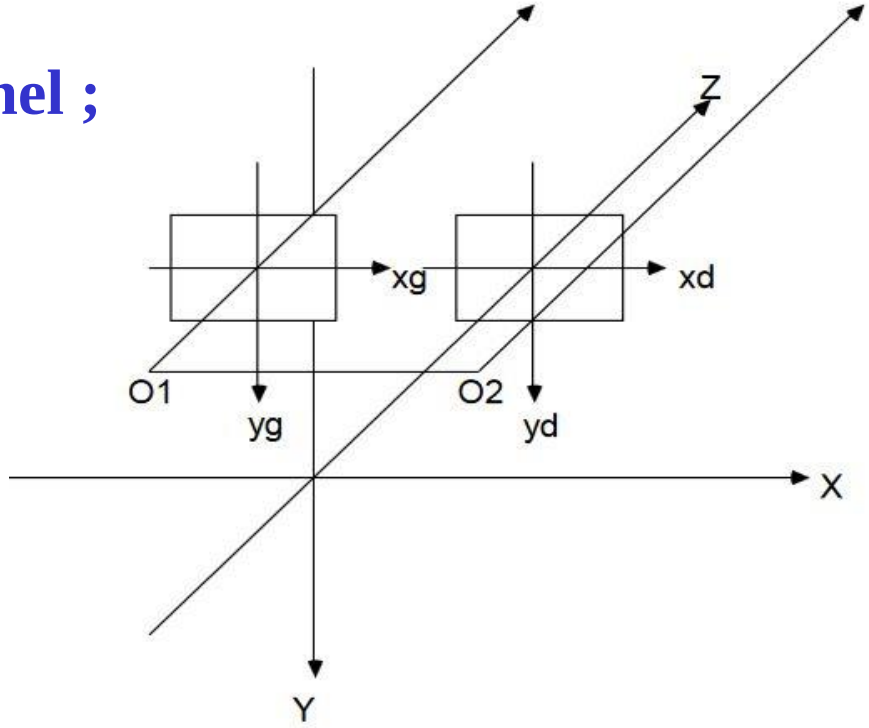
Capteurs : Stéréovision

- **Stéréovision :**
  - ✓ pour l'homme ;
  - ✓ pour un robot …
- **Elément important :**
  - ✓ la base : distance entre les caméras : pour voir des différences !



Capteurs : Stéréovision

- **Modélisation du Banc :**
  - ✓ **notation et repères personnel ;**
  - ✓ **origine au centre …**



Les diverses notations sont:

O1 (-Δ/2, -h, 0); O2 (Δ/2, -h, 0)

h: hauteur des caméras,

f: focale des caméras,

Δ: distance entre les caméras O1O2

δx, δy : taille en x et y du pixel sur le plan focal

nlig et ncol nombre de lignes et colonnes

ngx et ndx coordonnées pixel en X sur les images gauche et droite

ngy coordonnées pixel en Y sur l'image gauche

dx = ngx - ndx disparité en pixels

(xg,yg) , (xd,yd) coordonnées dans les plans images des points Mg et Md projection du point 3D M(X,Y,Z)

Les centres optiques des deux caméras ont pour position: $O_1 = \begin{pmatrix} -\Delta/2 \\ -h \\ 0 \end{pmatrix}$ $O_2 = \begin{pmatrix} \Delta/2 \\ -h \\ 0 \end{pmatrix}$

Capteurs : Stéréovision

- **Formules :**
  - ✓ **Projection 3D – 2D;**
  - ✓ **Reconstruction 3D distance inverses de la disparité**

p Formules de passage 2D -> 3D (TRIANGULATION)

** En fonction des coordonnées sur les plans images:*

$X = \Delta\, xg / (xg - xd) - \Delta/2$

$Y = \Delta\, yg / (xg - xd) - h$

$Z = \Delta\, f / (xg - xd)$

** En fonction des coordonnées pixel image*

$X = (ngx - (ncol/2 - 1)\,)\, \delta x\, \Delta / (dx\delta x) - \Delta/2$

$Y = (ngy - (nlig/2 - 1))\, \delta y\, \Delta / (dx\delta x) - h$

$Z = \Delta f / (dx\delta x)$

p Formules de passage 3D -> 2D (PROJECTIONS)

** En fonction des coordonnées sur les plans images:*

$xg = (X + \Delta/2)\, f / Z, \quad xg = (X - \Delta/2)\, f / Z$

$yg = yd = (Y + h)\, f / Z$

** En fonction des coordonnées pixel image*

$ngx = (X + \Delta/2)\, f / (Z\delta x\,) + (ncol/2 - 1)$

$ndx = (X - \Delta/2)\, f / (Z\delta x\,) + (ncol/2 - 1)$

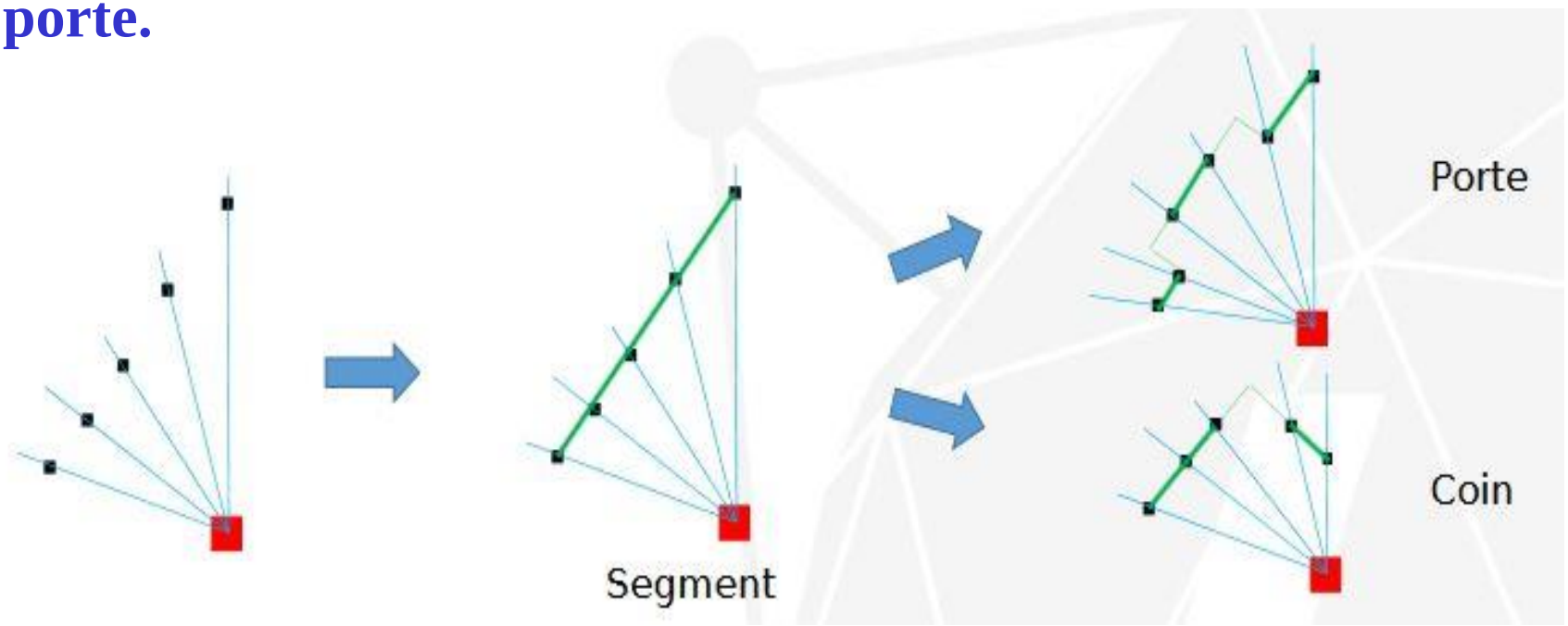
$ngy = ndy = (Y + h)\, f / (Z\delta y\,) + (nlig/2 - 1)$

Capteurs : Lidar

- **Lidar : pour la robotique autonome**
  - ✓ **Mono nappe : nuage de points dans un plan horizontal ;**
  - ✓ **Hokuyo : 10 m, 40 Hz, 130 g (1900 €)**
  - ✓ **YD Lidar : 10 m, 5 Hz, (100 €)**

Principe du Lidar

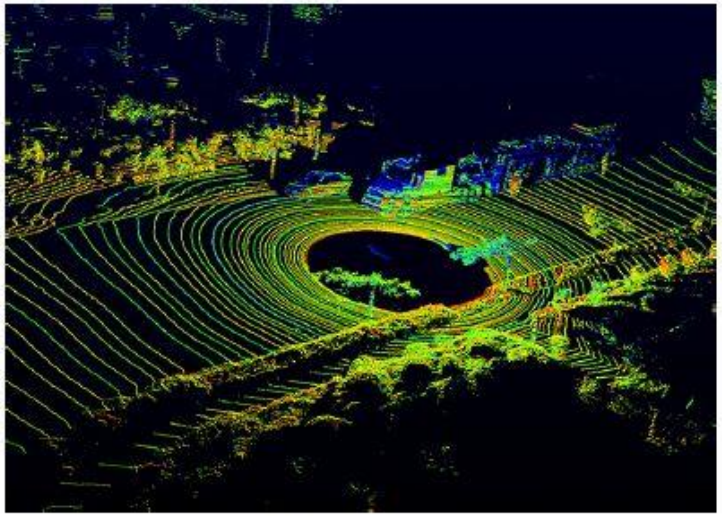
- **Principe**
  - ✓ **Lancer des rayons laser par rotation ;**
  - ✓ **Donnée : distance en fonction de l'angle ;**
  - ✓ **Effet sur un mur, un coin de mur et un renfoncement de porte.**



Capteurs : Velodynes

- **Velodyne :**
  - ✓ **Lidar : 64 nappes ;**
  - ✓ **Points 3D : Intersections avec les objets de la scène ;**
  - ✓ **Nouveaux sans pièce mobile ;**
  - ✓ **Qq années 50 k€, attente de nouveaux à moins de 100 € pour le monde automobile.**

(A) Velodyne HDL-64E



(B) Nuage de points

Capteurs : Velodynes

- **Velodyne :**
  - ✓ **Nouveaux capteurs .**

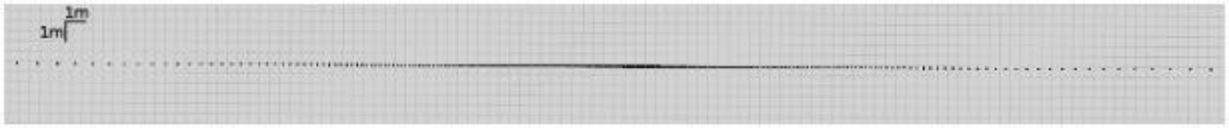
- ✔ Best horizontal (360°) and vertical (40°) long-range sensor
  - ✔ 10% targets >300m typical
  - ✔ 5% targets >180m typical
  - ✔ Ground plane hits >90m typical
- ✔ High resolution (0.2° x 0.1°) and point density at full frame rate

Effet du Mouvement : Caméra et Lidar

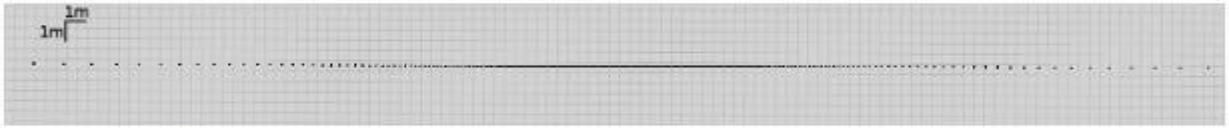
- **Mouvement et Fréquence Capteur**
  - ✓ **Caméra : Rolling et Global Shutter ;**
  - ✓ **Lidar : Vitesse de Rotation.**

(A) Images prises en global shutter

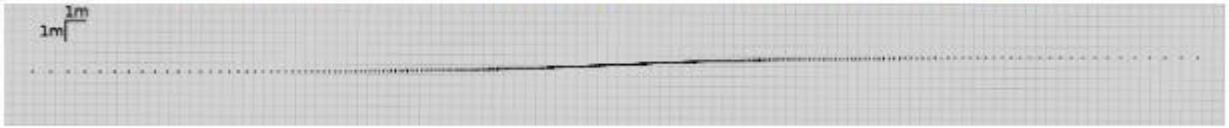
(B) Images prises en rolling shutter



(A) Trame RP-lidar pour une vitesse nulle



(B) Déformation d'une trame RP-lidar pour une vitesse de 1m/s

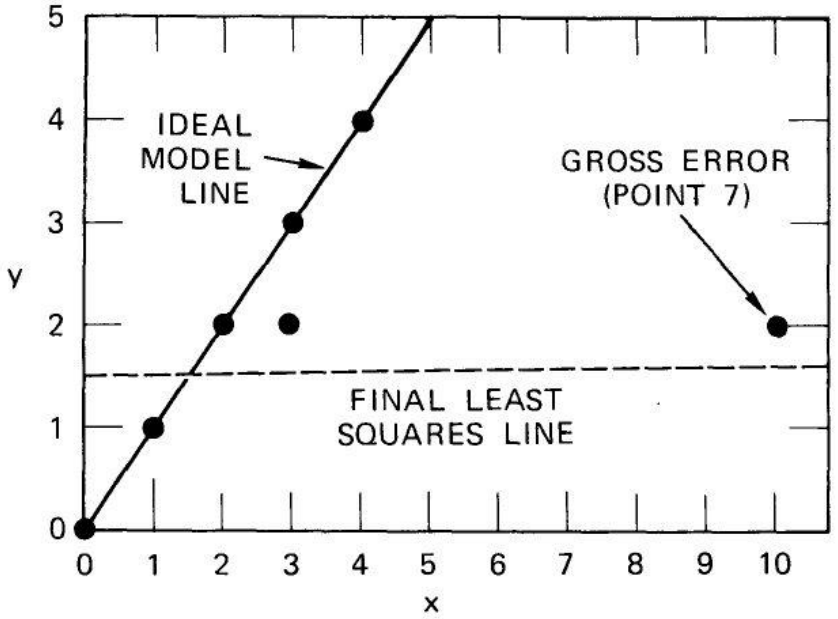


(C) Déformation d'une trame RP-lidar pour une vitesse de 10m/s

FIGURE 2.18: Effet de la vitesse sur un capteur RP-lidar

Traitement des Données du Lidar

- **Données Lidar :**
  - ✓ **Nuage « ordonné » de Points;**
  - ✓ **Traitements Classiques (ne tenant pas compte du côté ordonné) : Méthodes globales : Moindres Carrés, Hough, RANSAC**
- **Moindres Carrés : problème avec les Données Bruitées**
  - ✓**Cf exemple …**



Caractéristiques de la Méthode RANSAC

- **Intérêt : Prise en Considération des Données Erronées**
  - ✓ **Données satisfaisant au modèle,**
  - ✓ **Données ne satisfaisant pas au modèle.**
- **Méthode Itérative : Prédiction / Vérification d'Hypothèses**
  - ✓ **Tir aléatoire de Données : Modélisation Initiale,**
  - ✓ **Agrégation Rejet de nouvelles données,**
  - ✓ **Raffinement du Modèle Initial,**
  - ✓ **Quantification du Modèle trouvé à partir du nombre de données agrées.**

Algorithme de la Méthode RANSAC

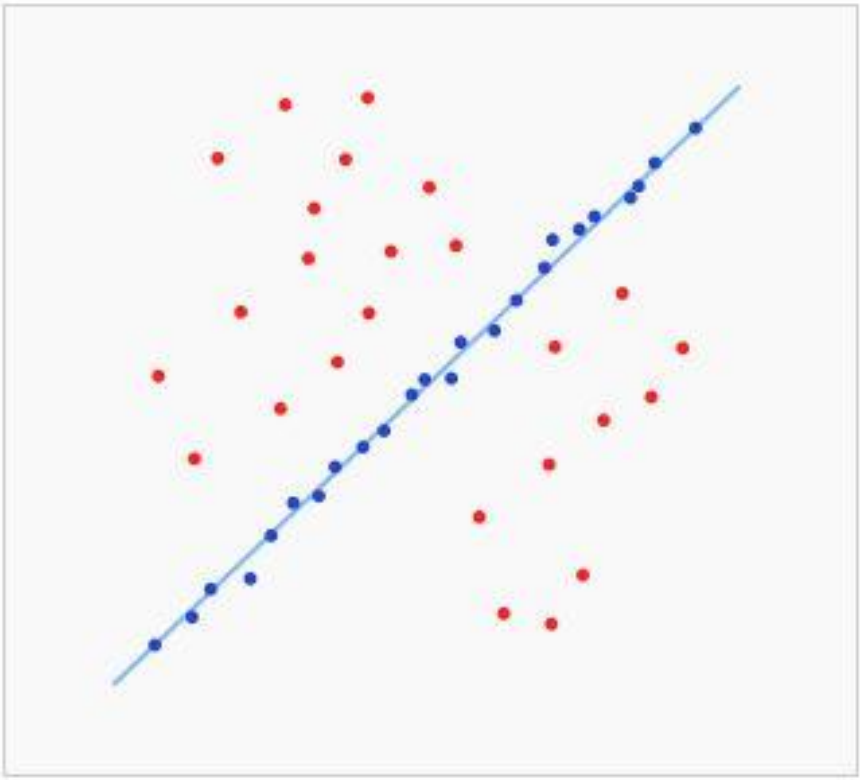
```
sorties :
    meilleur_modèle - les paramètres du modèle qui correspondent le mieux aux données (ou zéro si aucun bon modèle a été trouvé)
    meilleur_ensemble_points - données à partir desquelles ce modèle a été estimé
    meilleure_erreur - l'erreur de ce modèle par rapport aux données

itérateur := 0
meilleur_modèle := aucun
meilleur_ensemble_points := aucun
meilleure_erreur := infini
tant que itérateur < k
    points_aléatoires := n valeurs choisies au hasard à partir des données
    modèle_possible := paramètres du modèle correspondant aux points_aléatoires
    ensemble_points := points_aléatoires

    Pour chaque point des données pas dans points_aléatoires
        si le point s'ajuste au modèle_possible avec une erreur inférieure à t
            Ajouter un point à ensemble_points

    si le nombre d'éléments dans ensemble_points est > d
        (ce qui implique que nous avons peut-être trouvé un bon modèle,
        on teste maintenant dans quelle mesure il est correct)
        modèle_possible := paramètres du modèle réajusté à tous les points de ensemble_points
        erreur := une mesure de la manière dont ces points correspondent au modèle_possible
        si erreur < meilleure_erreur
            (nous avons trouvé un modèle qui est mieux que tous les précédents,
            le garder jusqu'à ce qu'un meilleur soit trouvé)
            meilleur_modèle := modèle_possible
            meilleur_ensemble_points := ensemble_points
            meilleure_erreur := erreur

    incrémention de l'itérateur

retourne meilleur_modèle, meilleur_ensemble_points, meilleure_erreur
```

Source : Wikipédia

Méthode RANSAC : Résultats pour la Recherche de Droites :

A partir des Points de Contour:

Un jeu de données avec de nombreuses valeurs aberrantes pour lequel une ligne doit être ajustée.



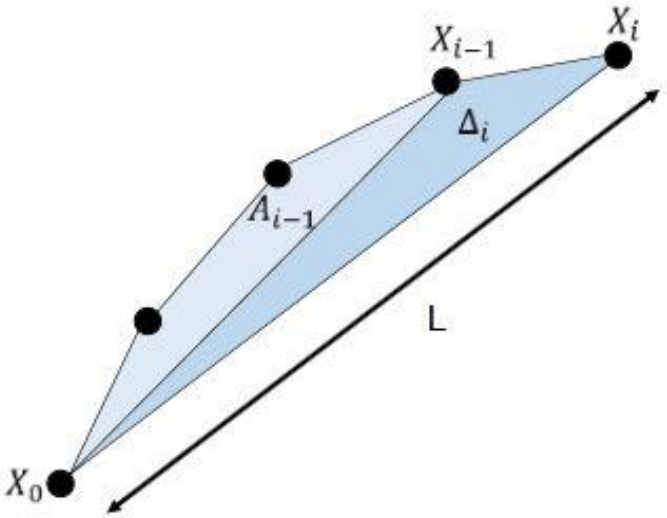
Ligne ajustée avec la méthode RANSAC, les valeurs aberrantes n'ont aucune influence sur le résultat.

Source : Wikipédia

Détection de Segments dans des Données Lidar (G.Burtin)

- **Détection de Segments**
  - ✓ **Méthode Traitement d'Image : Wall et Danielson,**
  - ✓ **Post traitement des segments morcelés.**

Principe de fonctionnement : les points sont ajoutés itérativement dans l'ordre en vérifiant la condition d'erreur acceptable



L'erreur d'approximation est la surface calculée Ai (bleue), formée entre l'arc et la corde, divisée par la longueur de l'arc

$$A_i = A_{i-1} + \Delta_i$$

On ajoute le point X_i au segment si :

$$\frac{A_i}{L} \leq Seuil$$

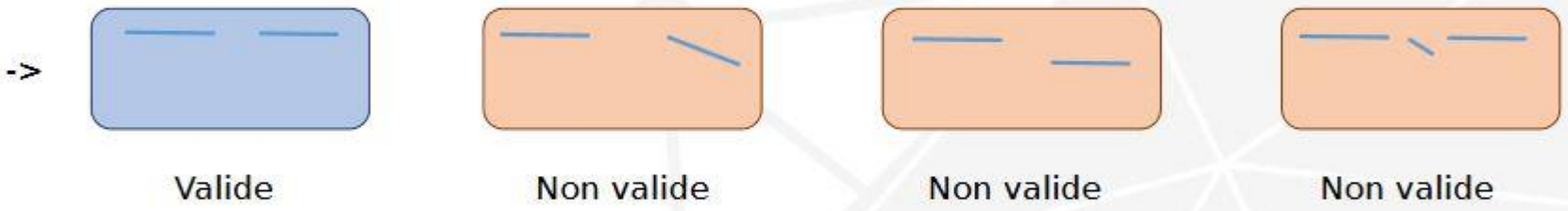
Détection de Segments dans des Données Lidar (G.Burtin)

- **Post Traitement**
  - ✓ **Segment morcelés ;**
  - ✓ **Méthode en 3 itérations**

Post-traitement : fusion - érosion - fusion

• Processus de fusion de segments

Segments consécutifs et portés par la même droite : considérés séparés à tord



Valide | Non valide | Non valide | Non valide

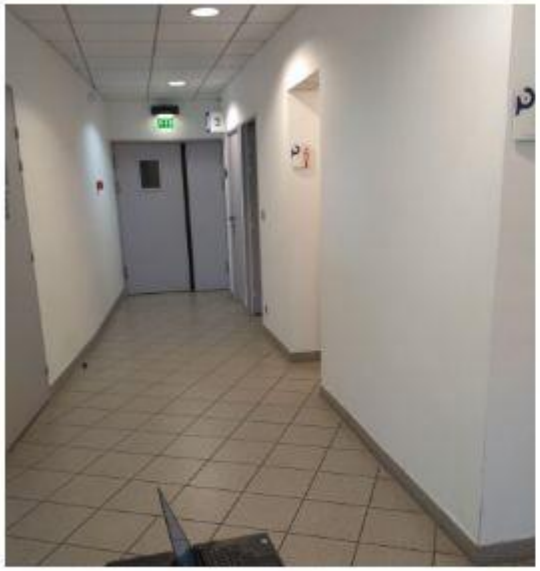
• Processus d'érosion

Segment de faible longueur : sensible au bruit angulaire (paramètre θ), considéré peu « intéressant »

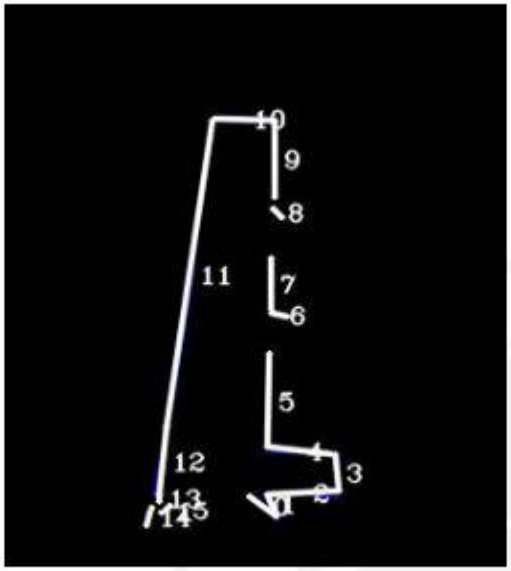
-> Suppression de tout segment inférieur à une longueur déterminée

Détection de Segments dans des Données Lidar (G.Burtin)

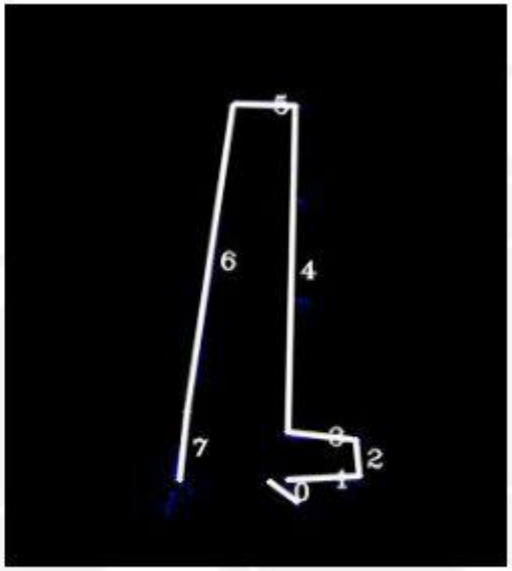
- **Résultats**
  - ✓ Expérimentations dans un couloir ;
  - ✓ Odométrie obtenue par recalage des cartes ;
  - ✓ Amers obtenus pour le recalage : fiables



Scène originale



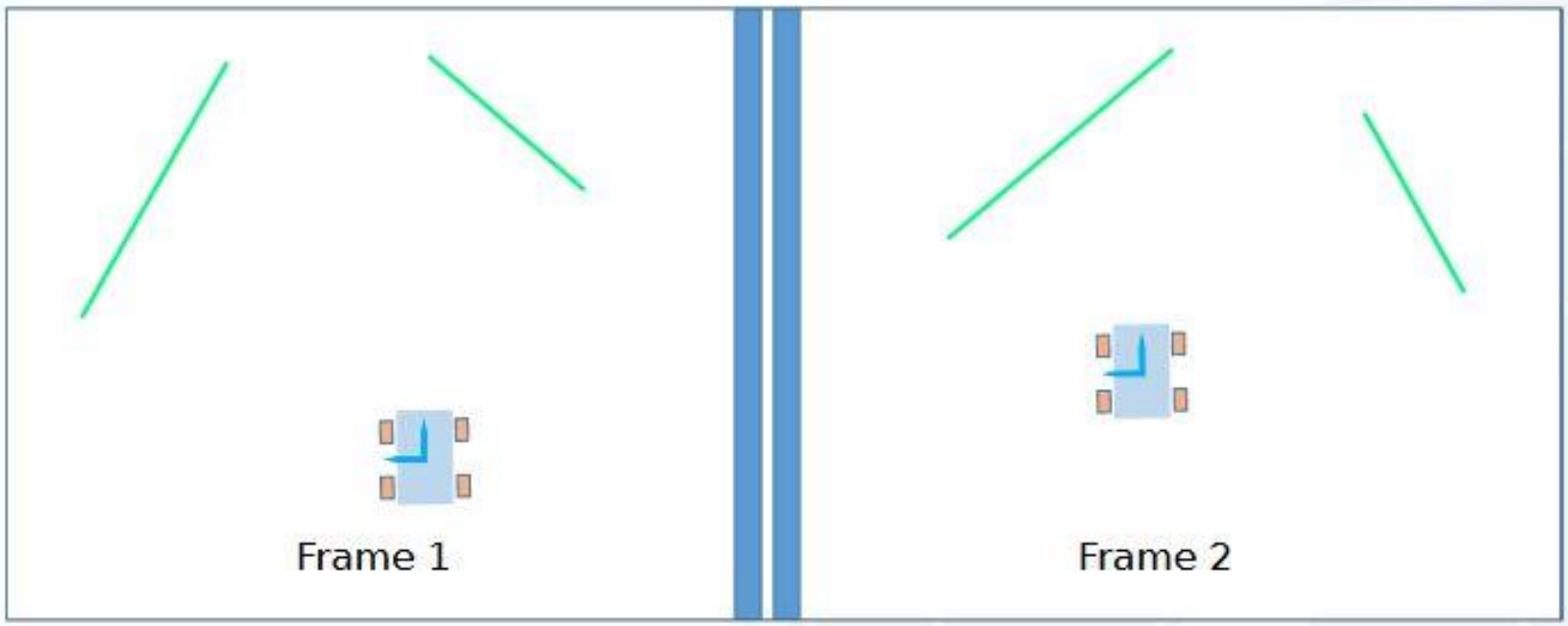
Segments initiaux



Amers finaux

Recalage à partir des Données Lidar : SLAM (G.Burtin)

Observation des amers lidar

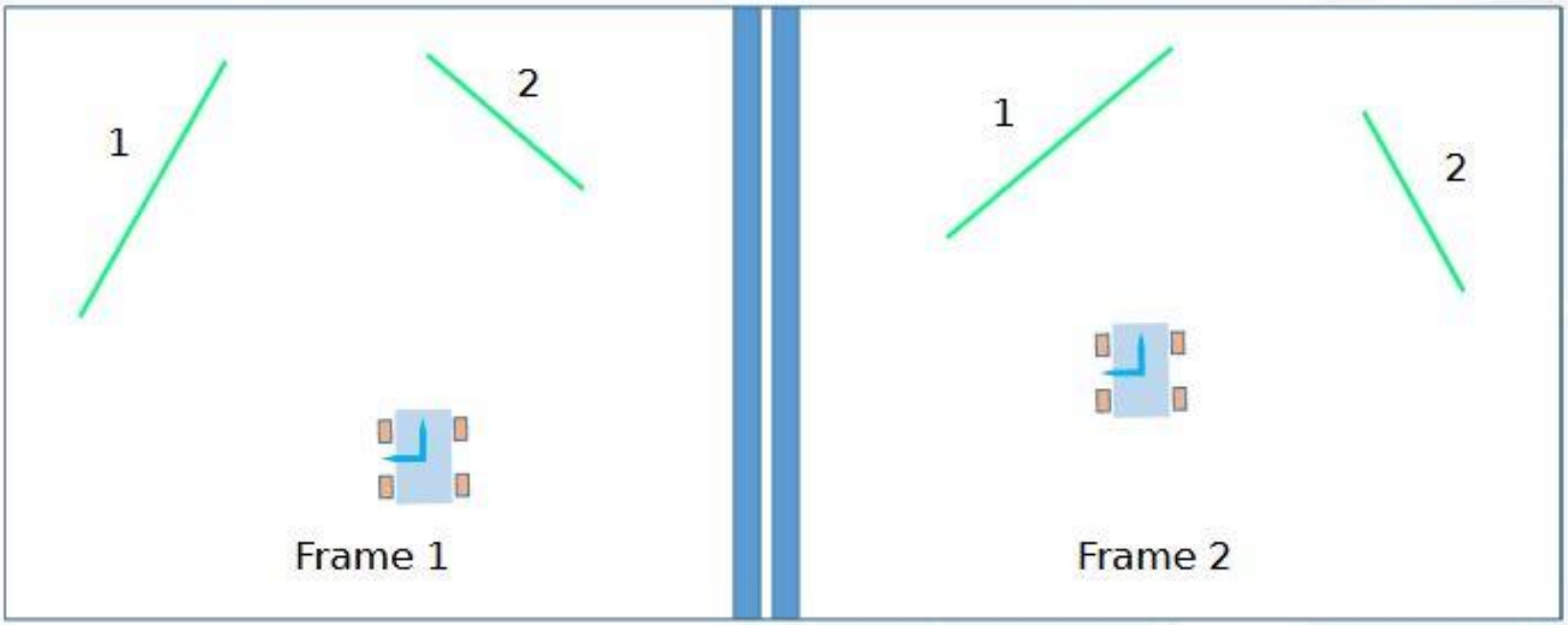


Recalage à partir des Données Lidar : SLAM (G.Burtin)

Appariement des amers d'une frame à l'autre :

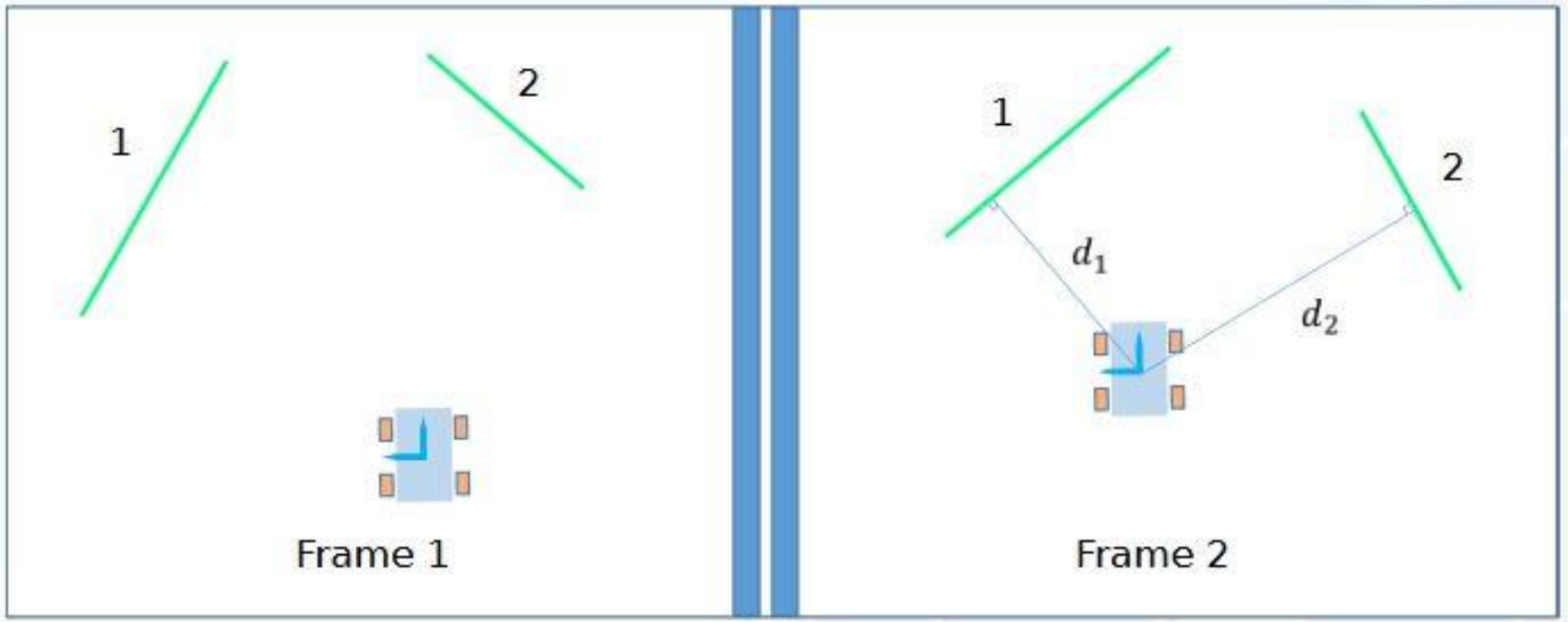
Critères de minimisation (longeur , angle, distance)

- Prédiction de la position estimée
- Optimisation de l'estimation de la matrice de pondération



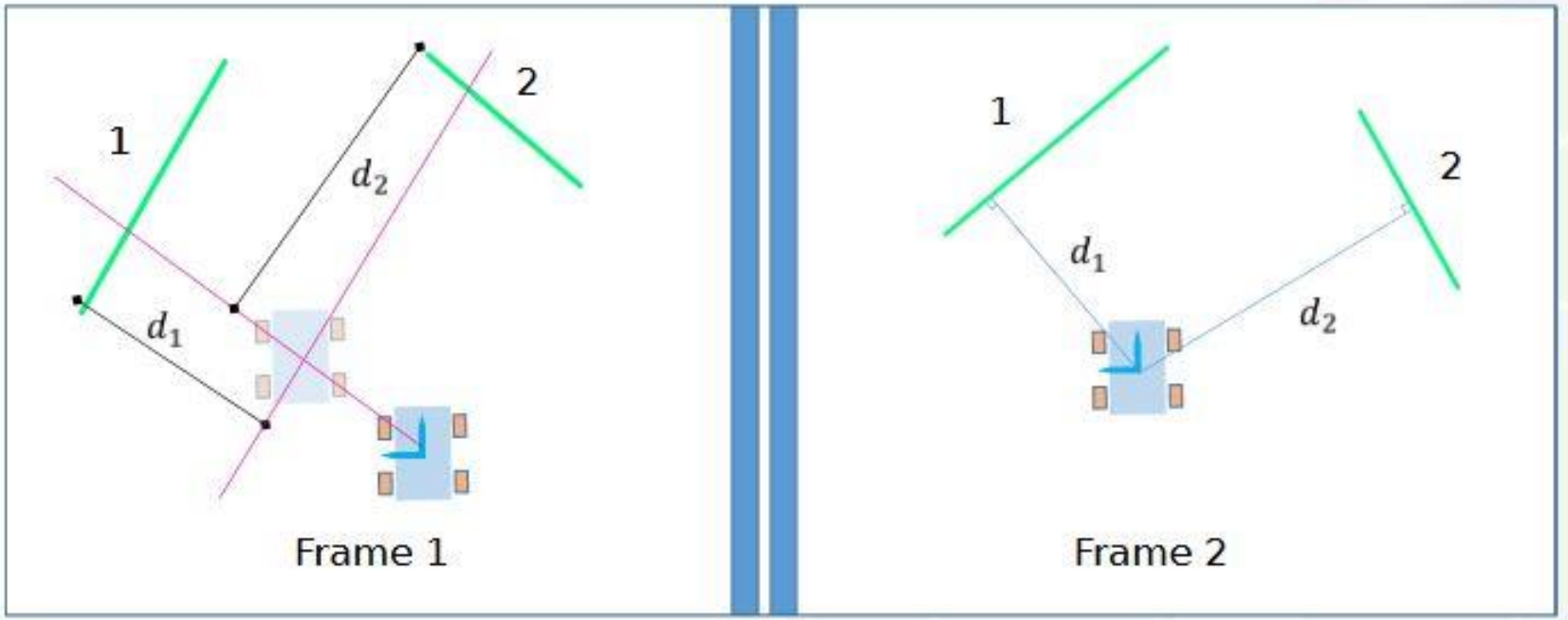
Recalage à partir des Données Lidar : SLAM (G.Burtin)

Calcul des distances minimales



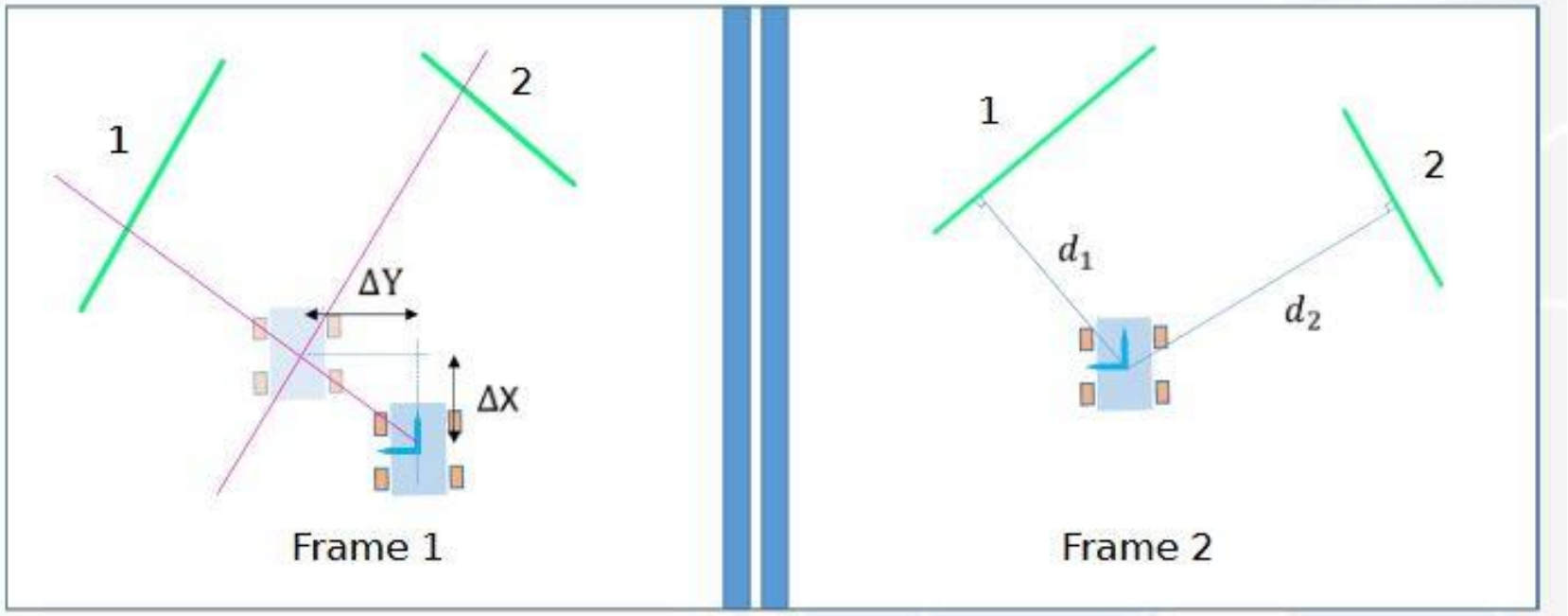
Recalage à partir des Données Lidar : SLAM (G.Burtin)

Reprojection dans la frame initiale du premier amer



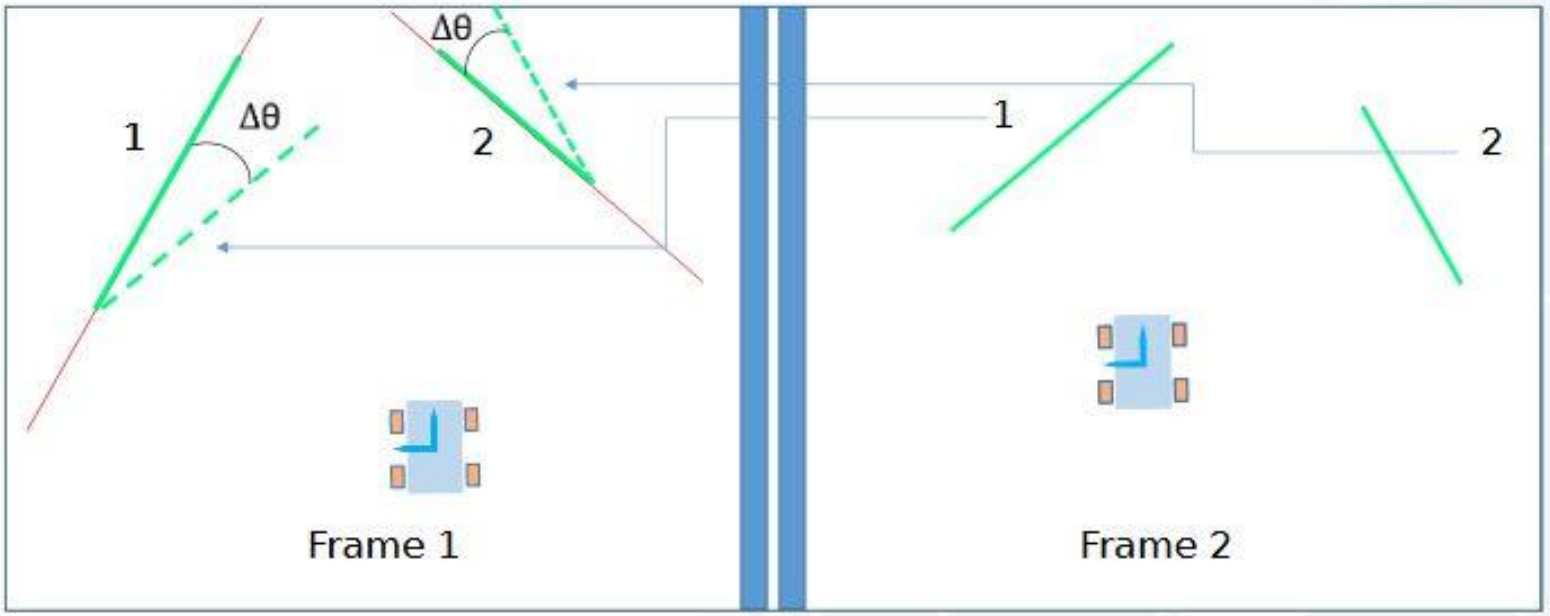
Recalage à partir des Données Lidar : SLAM (G.Burtin)

Résolution du déplacement entre les frames



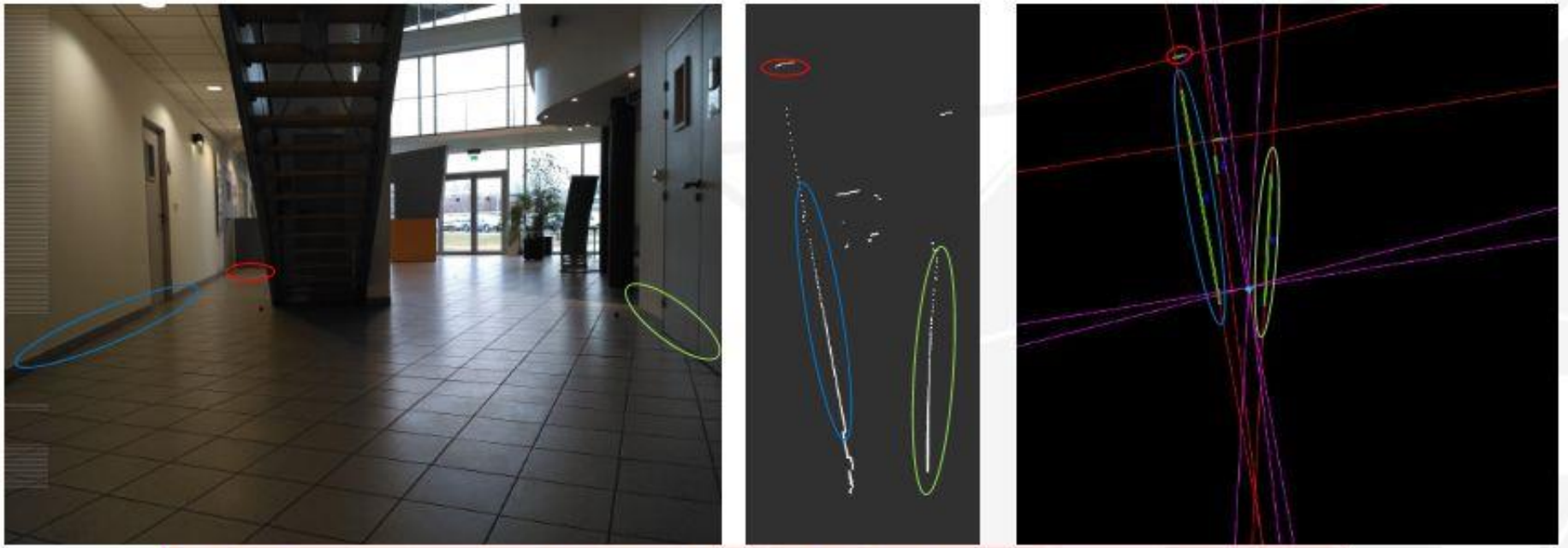
Recalage à partir des Données Lidar : SLAM (G.Burtin)

Le calcul du changement de l'orientation se fait avec les angles perçus



Recalage à partir des Données Lidar : SLAM (G.Burtin)

Recalage en environnement réel (RdC de Pascalis, locaux de 4D-Virtualiz)
- Mise en évidence des surfaces détectées (réel - lidar - recalage)
- Estimation du déplacement par l'intersection (pointe de la croix cyan)



Ici aussi, aucune modification de l'algorithme en passant du simulé au réel

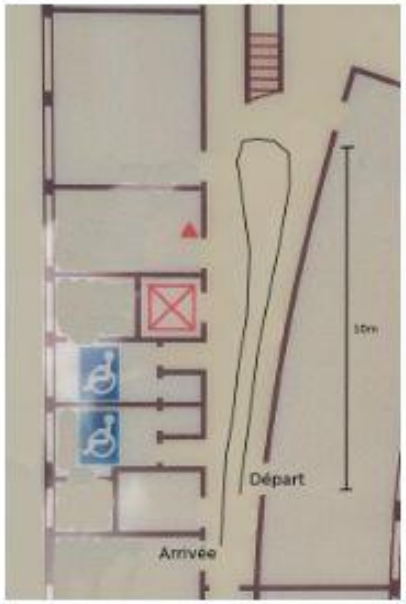
Recalage à partir des Données Lidar : SLAM (G.Burtin)

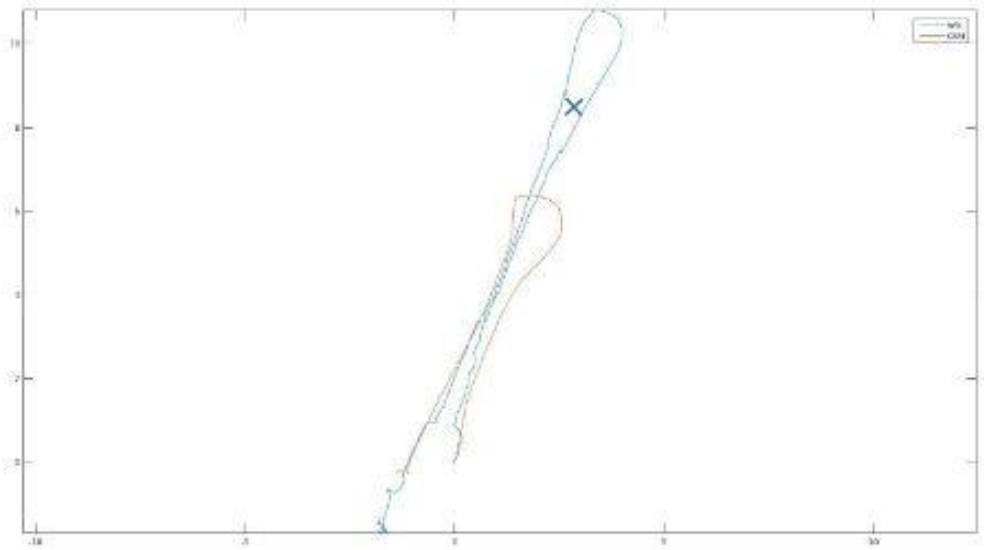
Test effectué a Pascalis dans le hall de ce bâtiment

Pas de possibilité d'avoir une vérité terrain

Conservation de la forme de la trajectoire

Le CSM sous-estime le déplacement alors que notre méthode le sur-estime

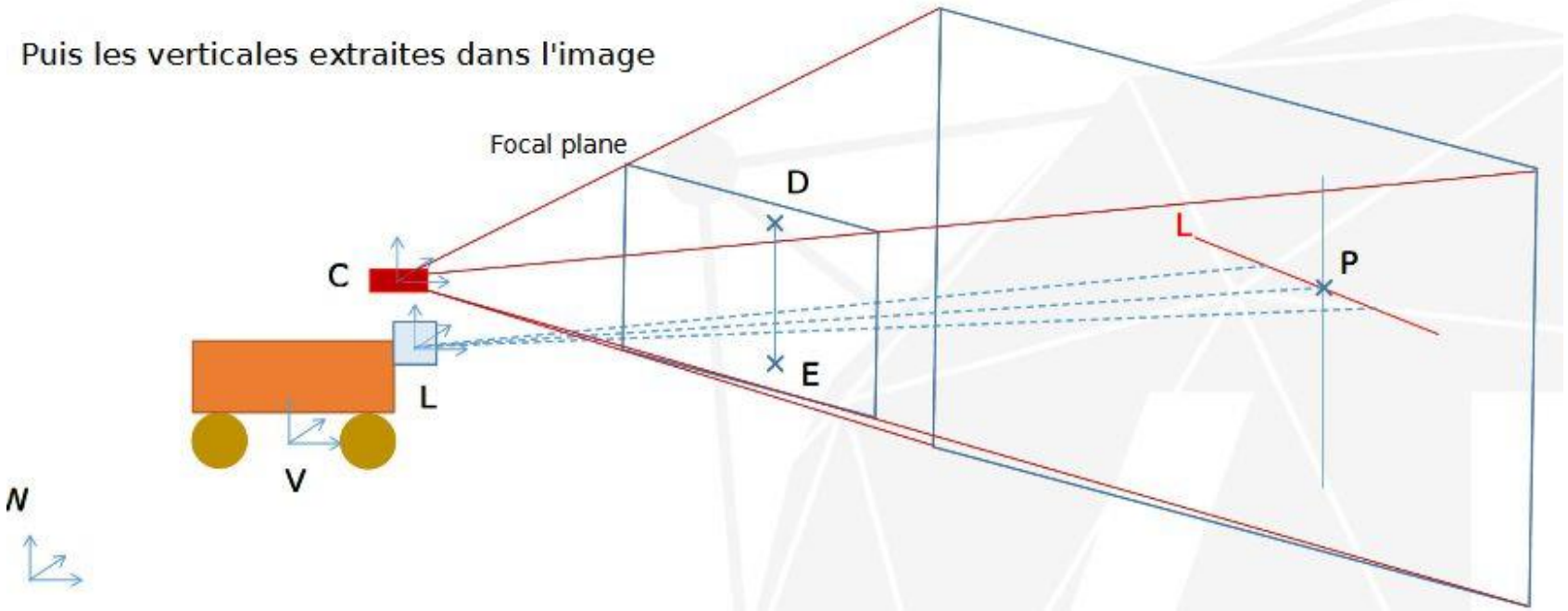




3D par fusion Lidar 2D Image 2D (G.Burtin)

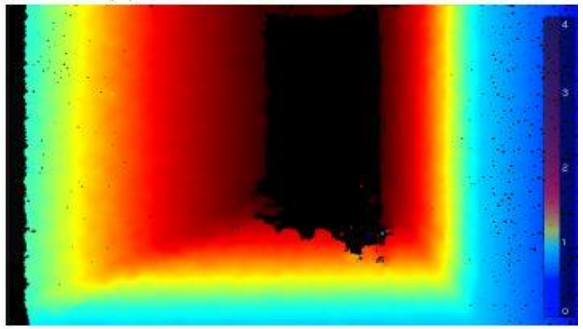
- **Hypothèse : Environnement humain**
  - ✓ Composé d'horizontales et de verticales 3D.
- **Principe :**
  - ✓ Détection d'un Amer dans le Plan Lidar : point P (3D);
  - ✓ Détection d'une Ligne Verticale dans l'Image ; D-E ;
  - ✓ Projection de la ligne DE dans l'espace 3D sur P

On cherche à détecter des points particulers (P) en utilisant à la fois la caméra RGB et la nappe laser : **exploitation de la complémentarité des capteurs**



Comparaison avec le Real Sense (G.Burtin)

Real Sense :



(B) Caméra de profondeur du RealSense D435

(A) Caméra RGB du RealSense D435

Fusion Lidar – Vision :

(C) Lidar URG-04LX

(E) Re-projection des points de contours filtrés sur l'image originale