LA VISION PAR ORDINATEUR : DE LA VISION 2D À LA VISION 3D.

Cédric Demonceaux

26 avril 2017

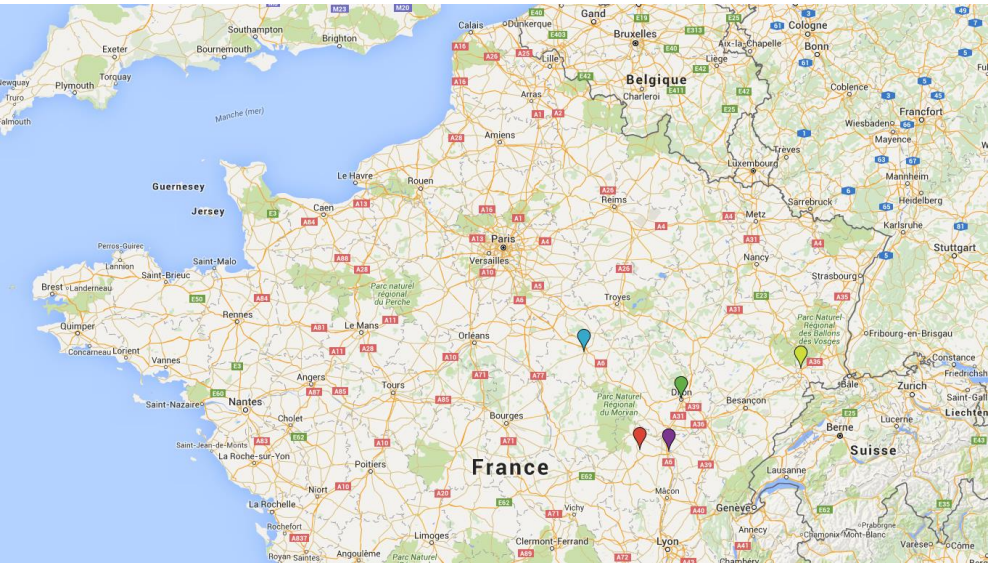
LIRMM – Séminaire Image

Laboratoire Électronique Informatique et Image

cnrs

UBFC UNIVERSITE BOURGOGNE FRANCHE-COMTE

Le2i en chiffres



5 Sites

- Auxerre
- Belfort
- Chalon sur Saône
- Dijon
- Le Creusot

110 Enseignants Chercheurs

7 pôles de recherche

Le2i

- P#1 : Modélisation Géométrique et Immersion Virtuelle
- P#2 : Combinatoire, Réseaux et Science des Données
- P#3 : Environnements Intelligents
- P#4 : Signal et Instrumentation
- P#5 : Systèmes de Vision et Méthodes d'Imagerie
- P#6 : Vision pour la Robotique
- P#7 : Imagerie Médicale et Santé

Le2i – Site Le Creusot

- 13 Enseignants Chercheurs (6 PR, 7 MCF)
- 1 IR CNRS
- 1 secrétaire CNRS
- 1 Tech Ub (30%)
- 3 pôles sont représentés

Systèmes de Vision
et Méthodes
d'Imagerie
7 EC

Vision pour la
Robotique
5 EC

Imagerie Médicale
et Santé
1 EC

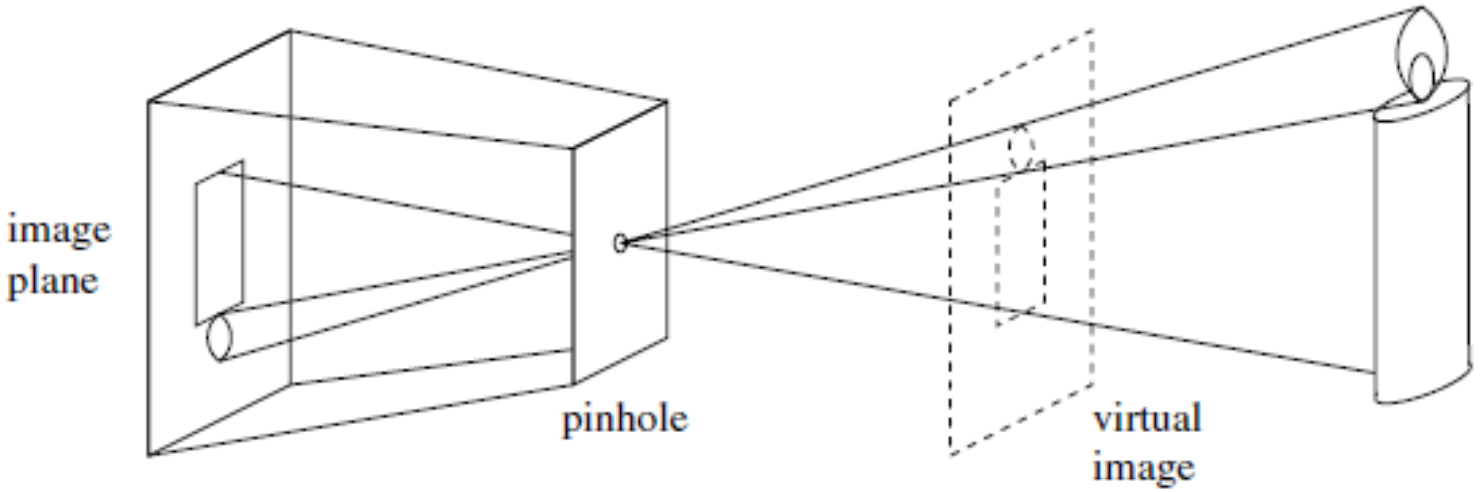
Plan

1. La vision par ordinateur
   I. Caméras : perspectives, catadioptrique, fish-eye, multi-caméras
   II. Calibrage : modèle unifié (SVP) & modèle générique (NSVP)
2. De la 2D à la 3D
   I. La 3D à partir d'une seule image (SVP & NSVP)
   II. La 3D à partir d'un système de vision stéréoscopique
   III. La 3D à partir du « Structure From motion »
3. Les caméras 3D
   I. La lumière structurée
   II. Les caméras temps de vol
4. Applications et travaux de notre équipe

Comment à partir de connaissances a priori sur la scène et/ou les caméras, pouvons-nous améliorer les approches de la littérature?

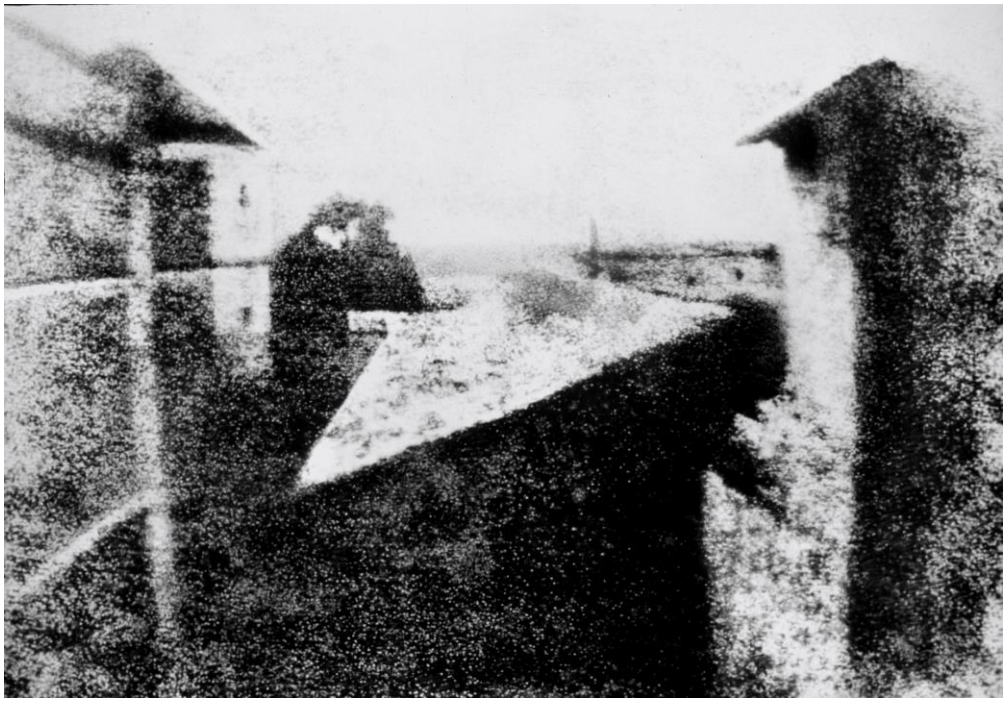
*certaines planches proviennent du cours de D. Sidibé : 3D reconstruction from Images, Master In Computer Vision, Université de Bourgogne

Modèle sténopé : caméra *pinhole*



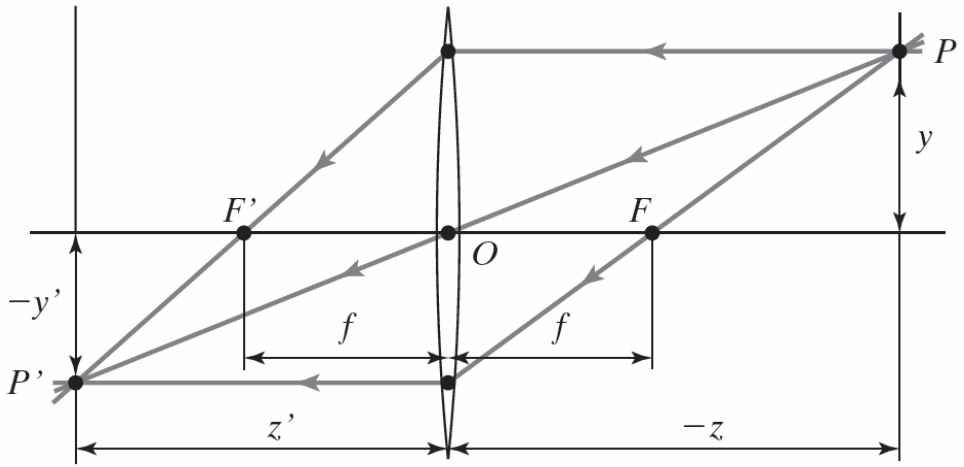
- Modèle sténopé (chambre noire – *camera obscura*) : centre de projection+surface photo-sensible
- Un point 3D est projeté le long d'une ligne de vue passant par le "trou d'épingle"
- Une image est composée de l'intersection de ces lignes de vue sur la surface photo-sensible (rétine)
- Une image est issue d'une projection perspective!

Modèle sténopé : caméra *pinhole*

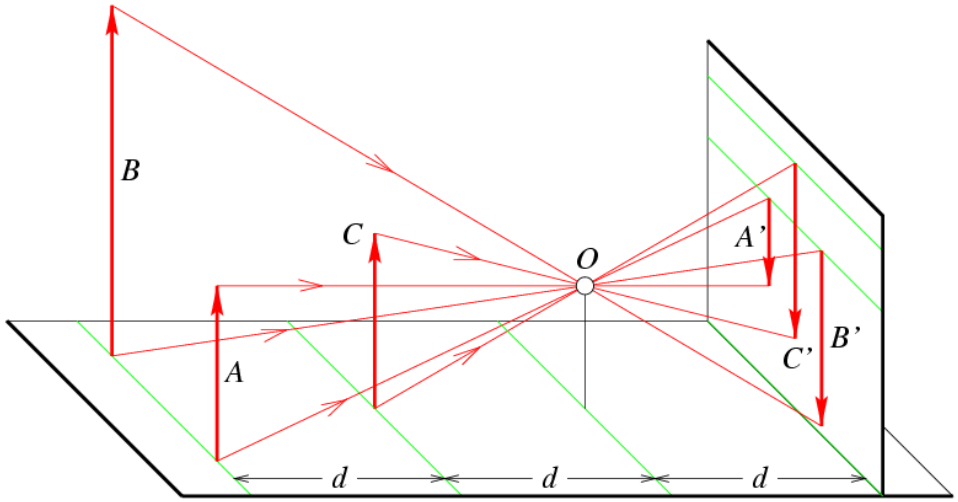


1826 : Point de vue du gras (Nicéphore Niepce)

Caméra perspective

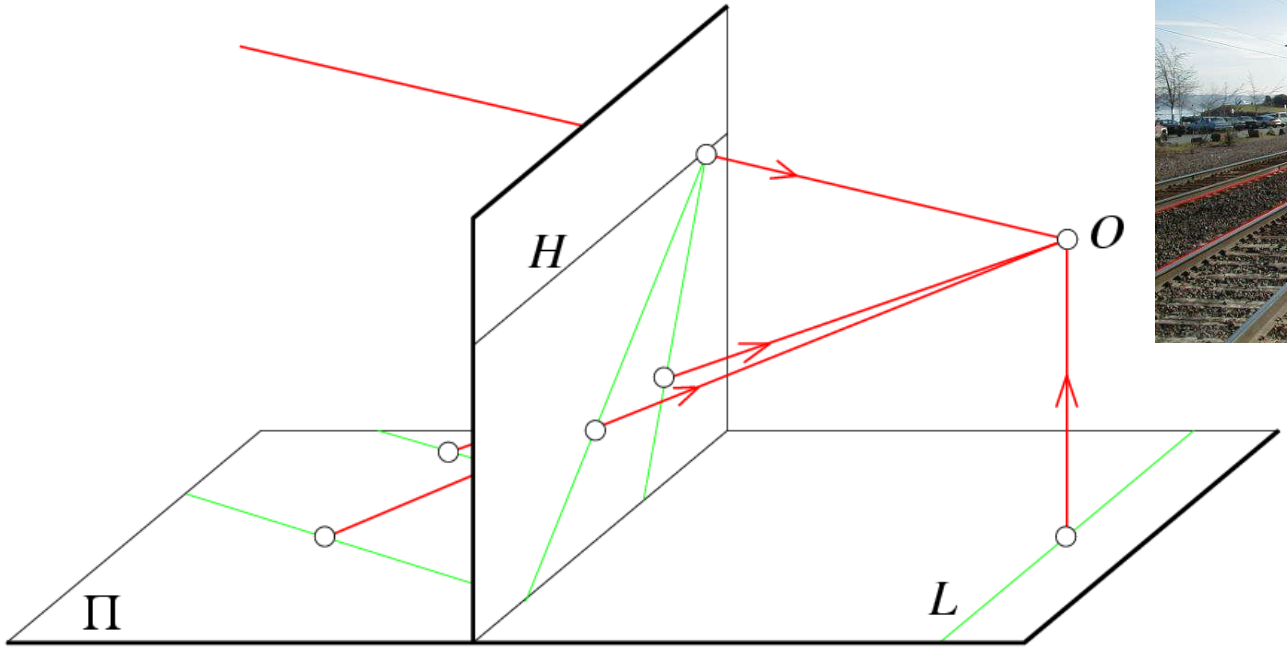


Caméra perspective



- La taille d'un objet sur l'image dépend de sa distance relative par rapport au centre optique.

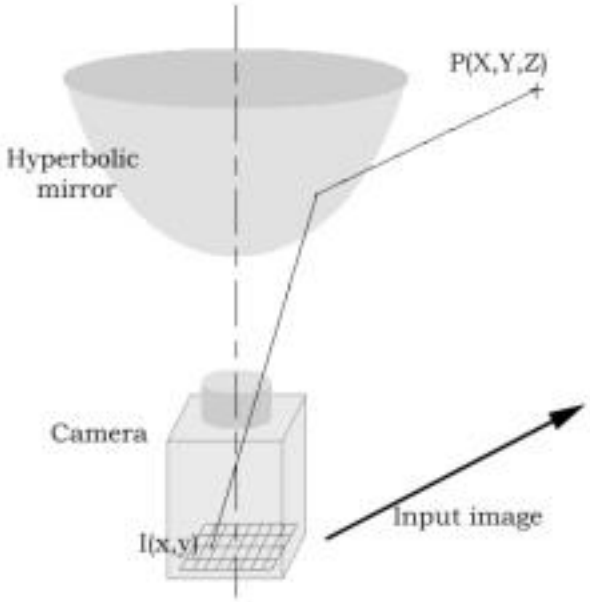
Caméra perspective

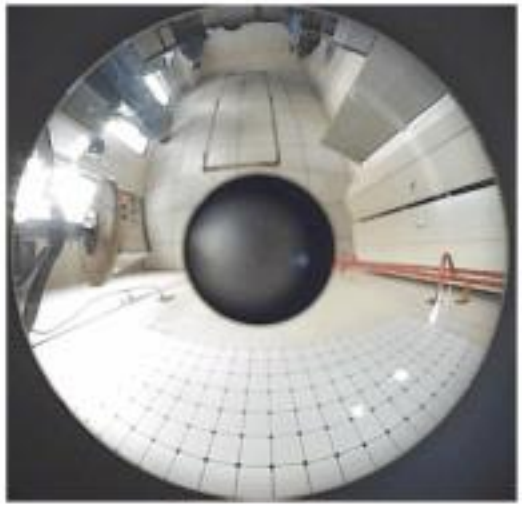


- 2 droites parallèles s'interceptent : point de fuites

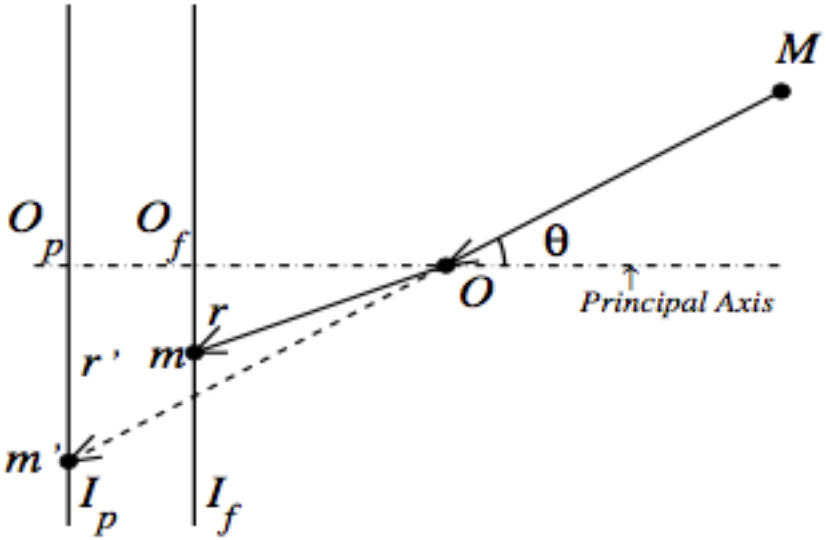
Champ de vue très limité

Caméra grand angle : catadioptrique





Caméra grand angle : Fish-eye



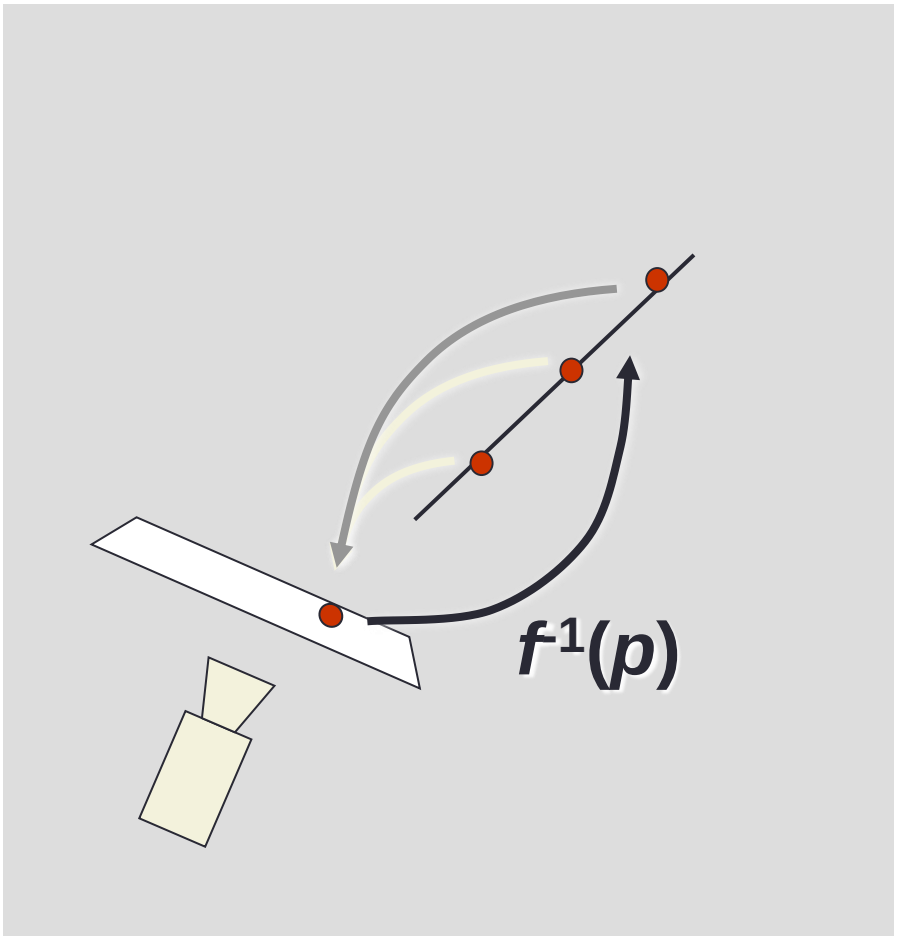
Caméra à large champ de vues : multi-caméra

LadyBug

Spherical GoPro

eleVRant

Samsung Gear 360



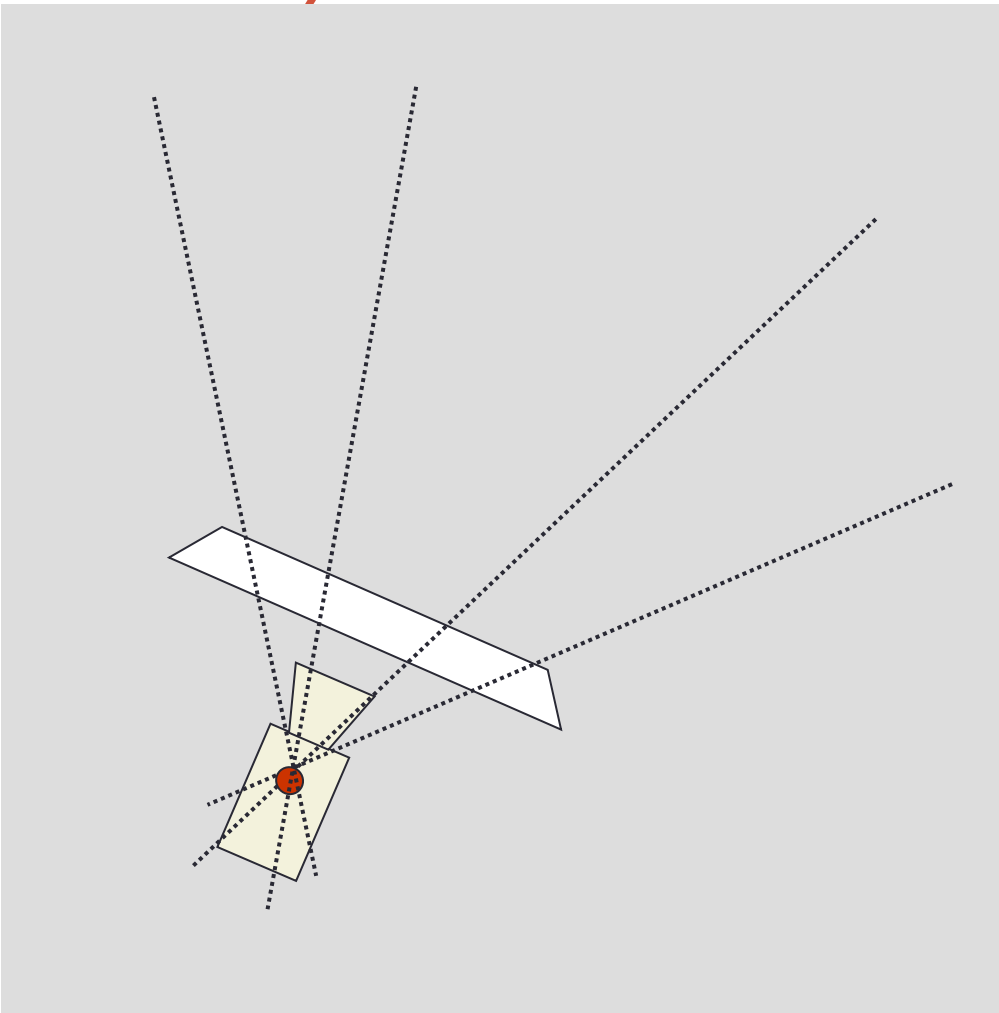
The projection induced by a camera is the function from space to the image plane, e.g.

$$f: \mathbb{P}^3 \rightarrow \mathbb{P}^2$$

The least restrictive assumption that can be made about any camera model is that the inverse image of a point is a line in space

[1]C. Geyer, T. Pajdla, K. Daniilidis, Courses on Omnidirectional Vision. In *ICCV'03*.

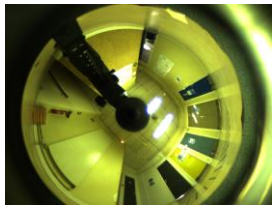
Caméra centrale (SVP – Single View Point)



If all the lines intersect in a single point, the camera has a single viewpoint and called central camera

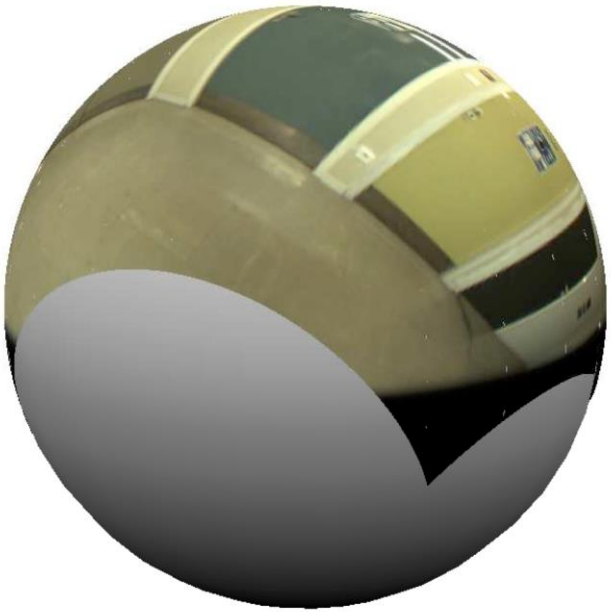
[1]C. Geyer, T. Pajdla, K. Daniilidis, Courses on Omnidirectional Vision. In *ICCV'03*.

Caméra centrale



1,2,3

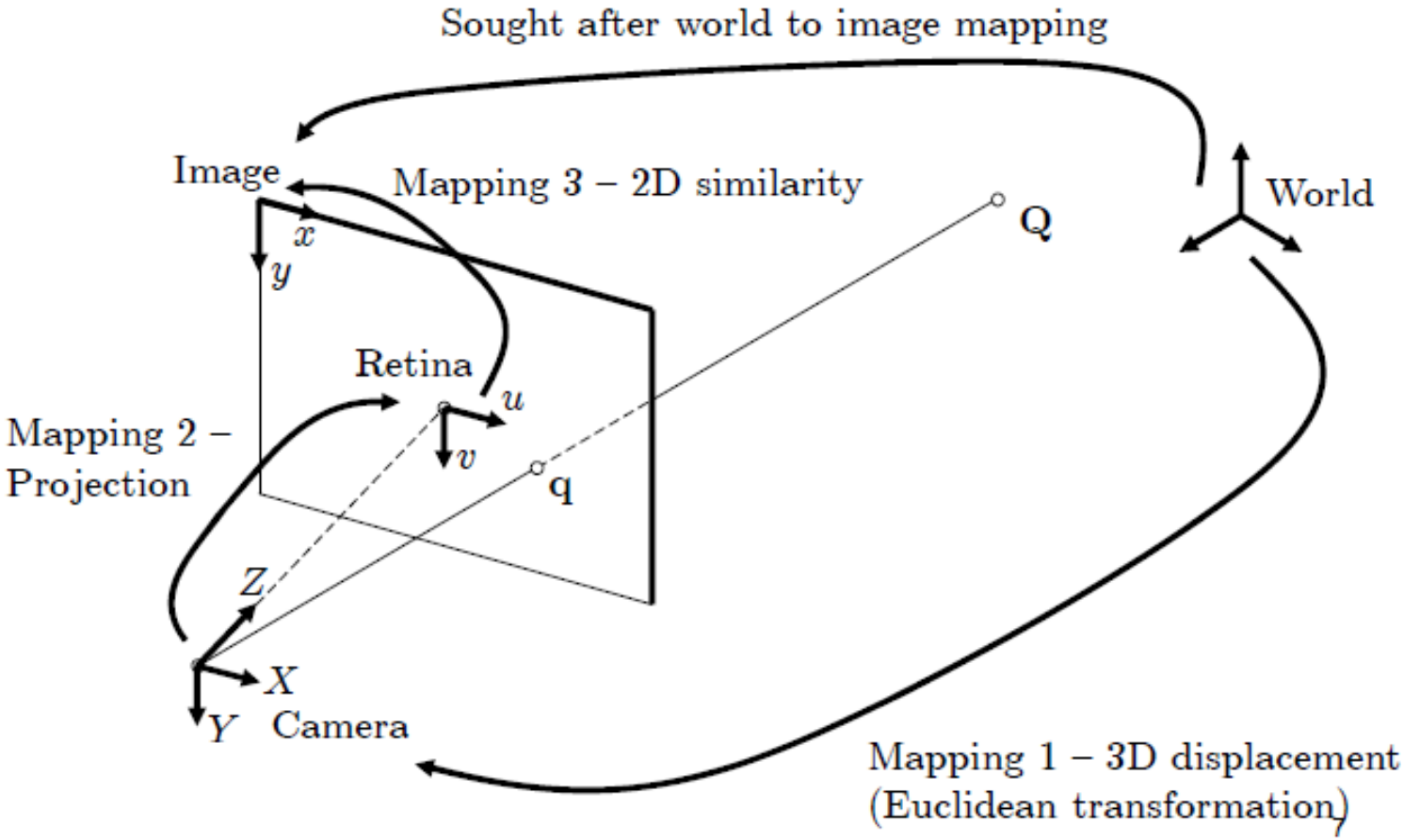
P



[1]C. Geyer and K. Daniilidis. Catadioptric projective geometry. IJCV 2001.
[2]J. Courbon, Y. Mezouar, L. Eck, and P. Martinet. A generic fisheye camera model for robotic applications. IROS 2007.
[3]X. Ying and Z. Hu. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. ECCV 2004.

Modélisation (ex : caméra perspective)

Mapping a 3D point to a 2D image point : 3 mappings



Modélisation (ex : caméra perspective)

First mapping : From 3D world to Camera

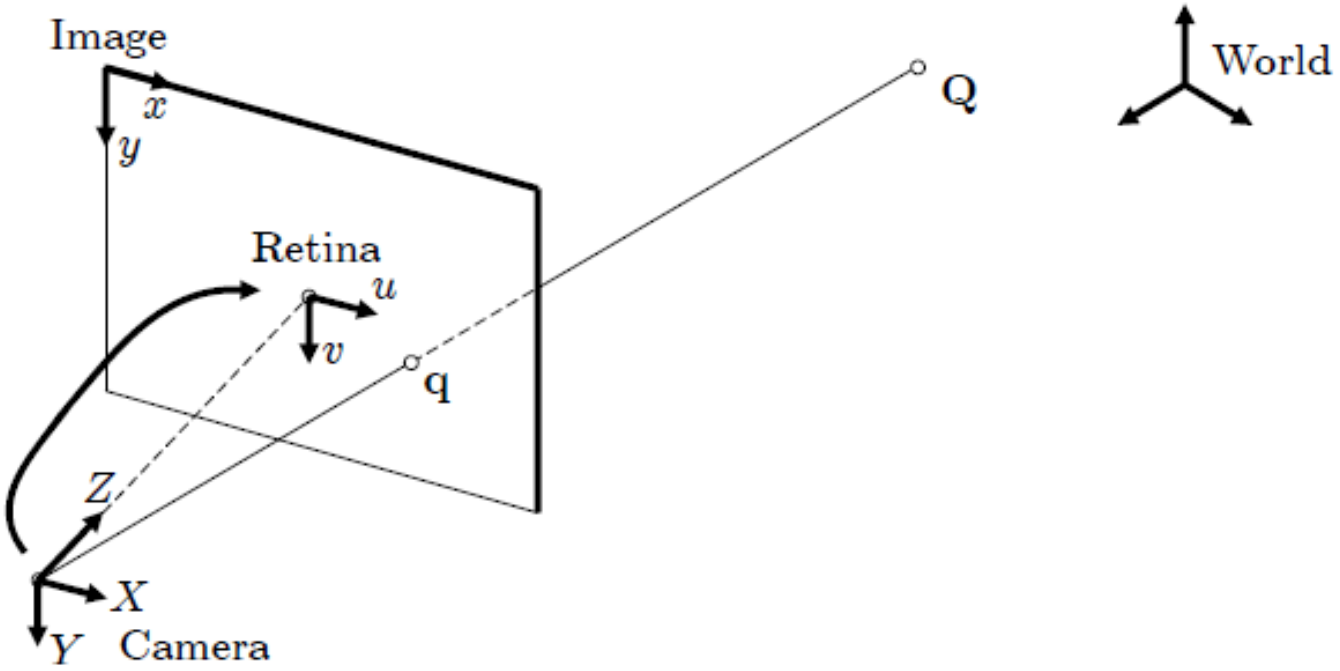
- ❑ Models camera displacements : position and orientation
- ❑In homogeneous coordinates:

$$Q_c = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} Q$$

- **R** is a (3 x 3) rotation matrix : $R^T R = I$ and det(R)= 1
- **t** is a (3 x 1) translation vector
- **Q** is the world homogeneous coordinates of the 3D point $Q \sim \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$

Modélisation (ex : caméra perspective)

Second mapping : From Camera to Retina



$$\text{Maps } \mathbb{R}^3 \text{ to } \mathbb{R}^2$$

Modélisation (ex : caméra perspective)

Second mapping : From Camera to Retina

- Camera-centered 3D point coordinates : $Q_c \sim (X_c \;\; Y_c \;\; Z_c \;\; 1)^T$
- Retina-centered coordinates:

$$u = f\frac{X_c}{Z_c} \quad \text{and} \quad v = f\frac{Y_c}{Z_c}$$

in homogeneous coordinates

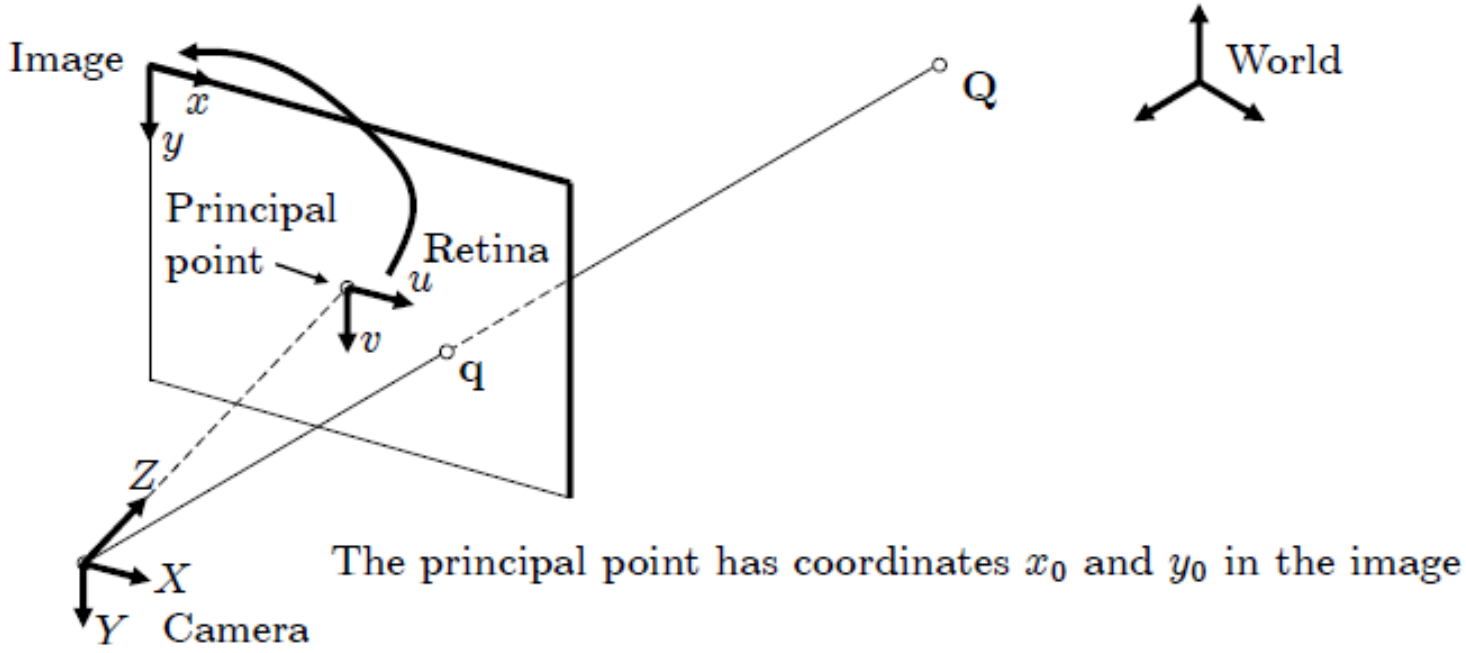
$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX_c \\ fY_c \\ Z_c \end{pmatrix}$$

- In matrix form, the projection can be written as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

Modélisation (ex : caméra perspective)

Third mapping : From Retina to Image



The principal point has coordinates x_0 and y_0 in the image

$$\text{Maps } \mathbb{R}^2 \text{ to } \mathbb{R}^2$$

Modélisation (ex : caméra perspective)

Third mapping : From Retina to Image

- k_x, k_y : are the density of pixels along *u* and *v*, e.g. in number of pixels per mm
- We have:

$$x = k_x u + x_0 \quad \text{and} \quad y = k_y v + y_0$$

which in matrix form gives:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} k_x & 0 & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

Modélisation (ex : caméra perspective)

Mapping 2+3: Camera to Image

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} k_x & 0 & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

$$= \underbrace{\begin{pmatrix} f k_x & 0 & x_0 \\ 0 & f k_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}}_{K} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

- K is the **camera calibration matrix**
- K contains the ‘internal’ or ‘intrinsic’ camera parameters

Modélisation (ex : caméra perspective)

Mapping 1+2+3: World to Image

$$q \sim \underbrace{\begin{pmatrix} fk_x & 0 & x_0 \\ 0 & fk_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}}_{K} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} & R & & t \\ 0 & 0 & 0 & 1 \end{pmatrix} Q$$

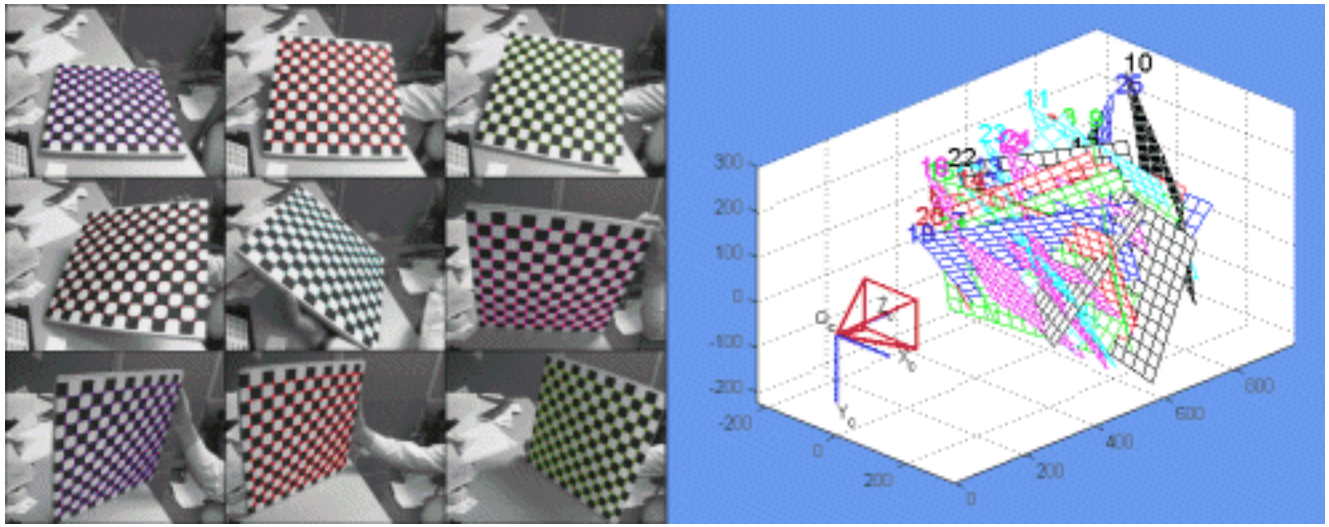
or

$$q \sim \underbrace{(KR \;\; Kt)}_{P} Q$$

- **P** is the **perspective projection matrix**: P is a (3 x 4) matrix
- **K** contains the ‘internal’ or ‘intrinsic’ camera parameters
- **R** and **t** are the ‘extinsic’ or ‘external’ camera parameters, also called the pose of the camera

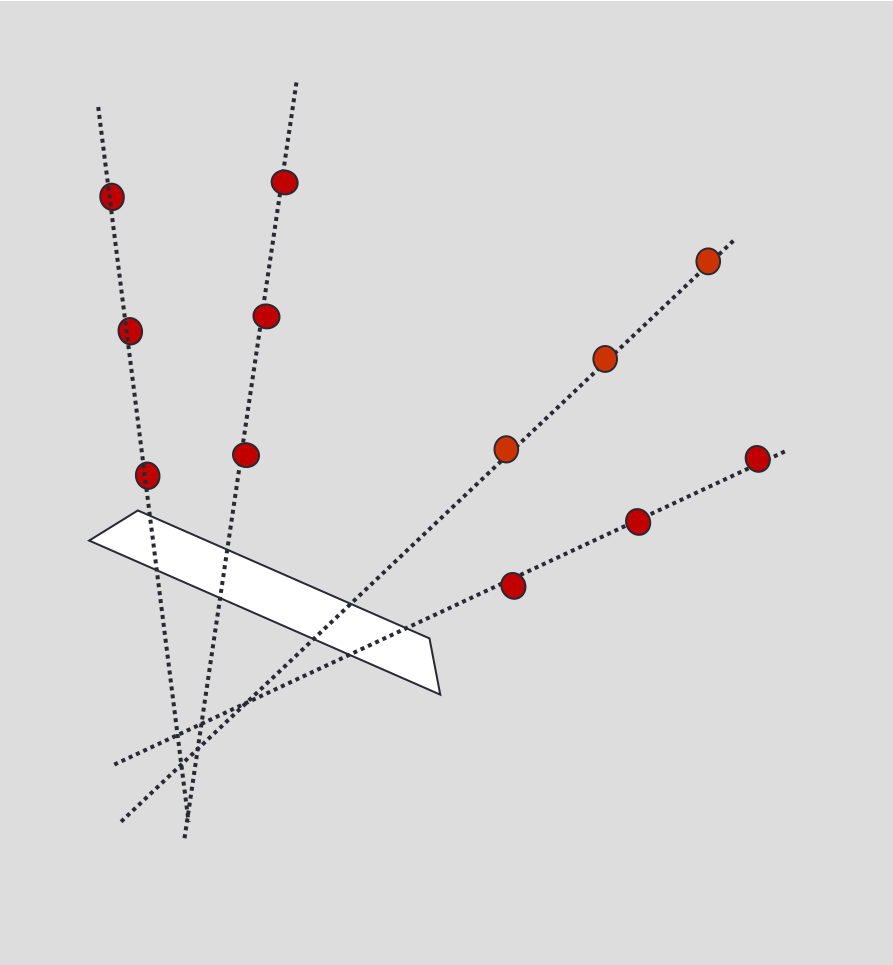
Modélisation (ex : caméra perspective)

$$q \sim \underbrace{(KR \;\; Kt)}_{P} Q$$



Camera calibration Toolbox for Matlab (Bouguet)

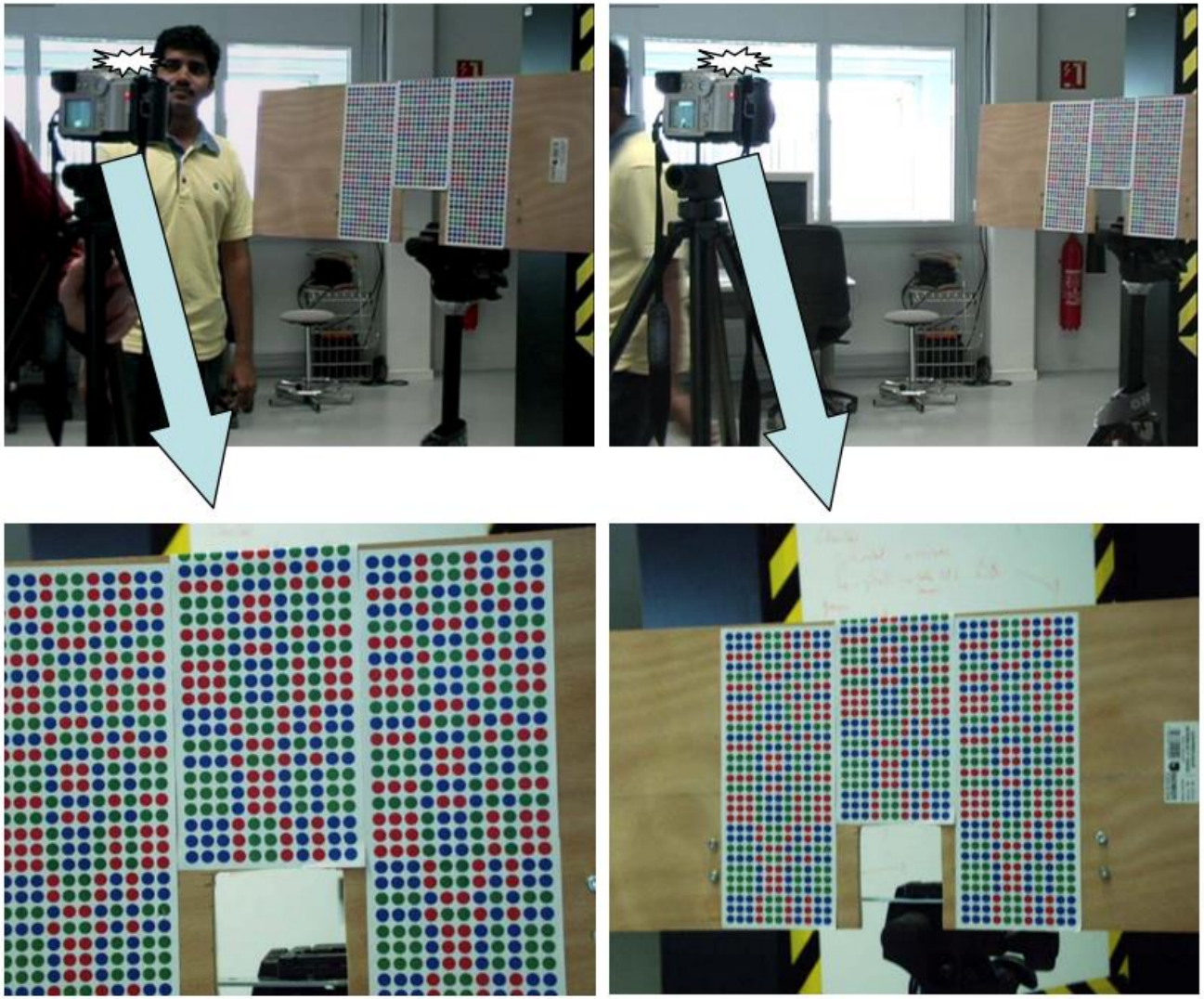
Caméra non centrale (NSVP – Non Single View Point)



For many cameras, all such lines do not necessarily intersect in a single point. We call this camera : Non-Central Camera

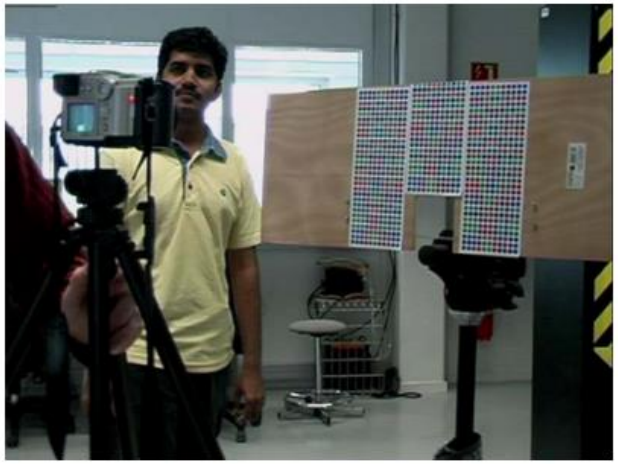
[1]C. Geyer, T. Pajdla, K. Daniilidis, Courses on Omnidirectional Vision. In *ICCV'03*.

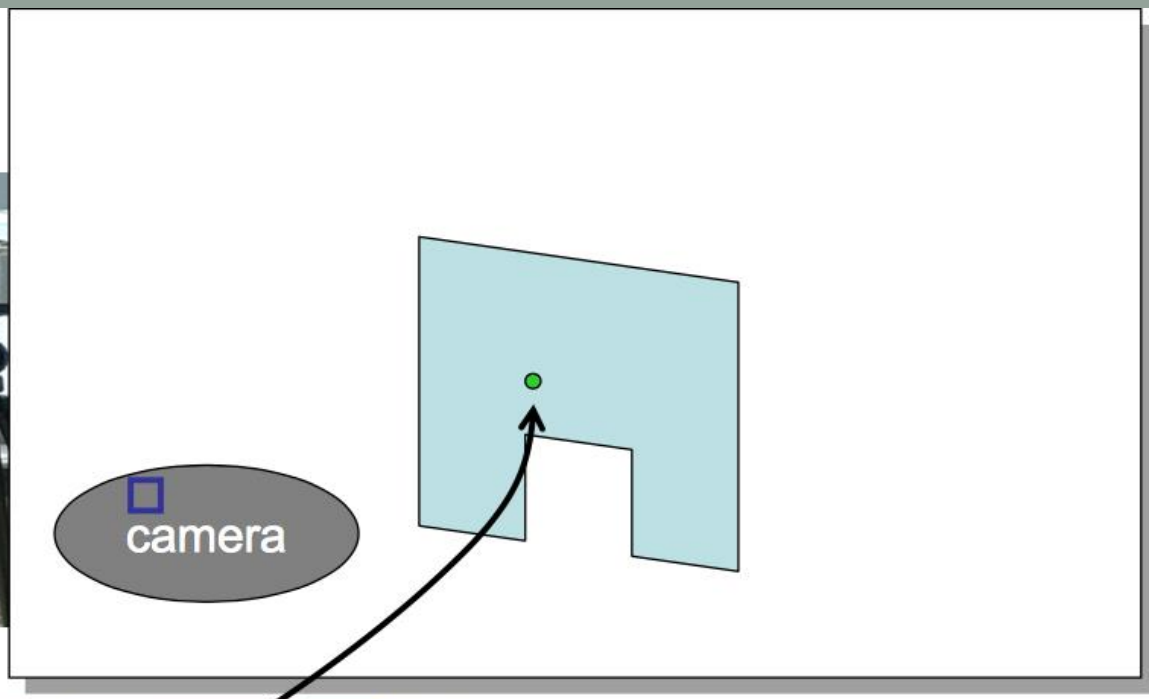
Approach using known motion: [Gremban-etal-ICRA'88,Champleboux-etal-ICRA'92, Grossberg-Nayar-ICCV'01]

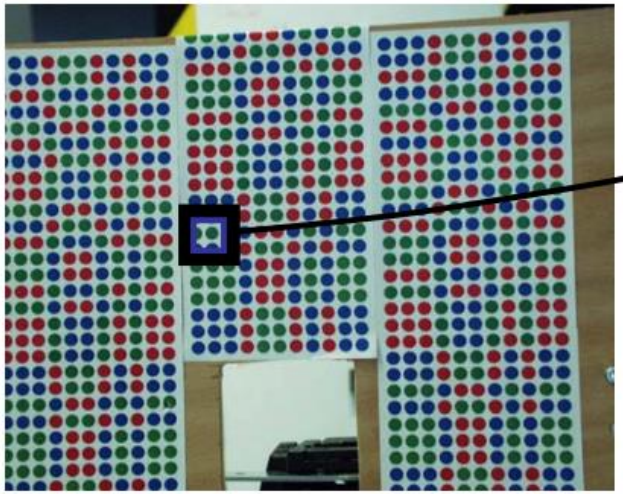


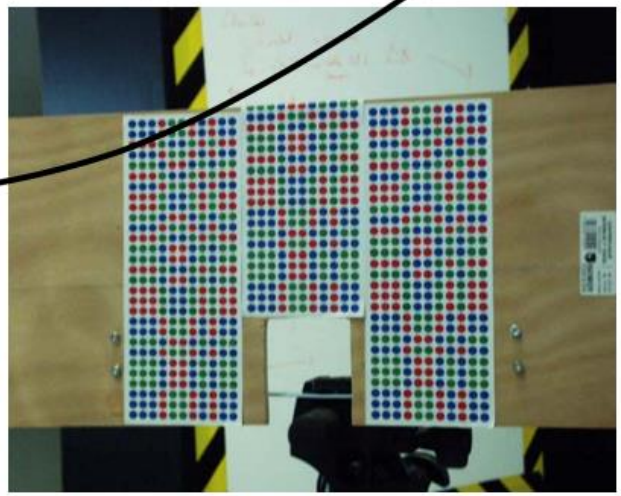
[1]From: PCV'06 - Tutorial on Modeling and Analysing Images of Generic Cameras, P. Sturm

Approach using known motion:



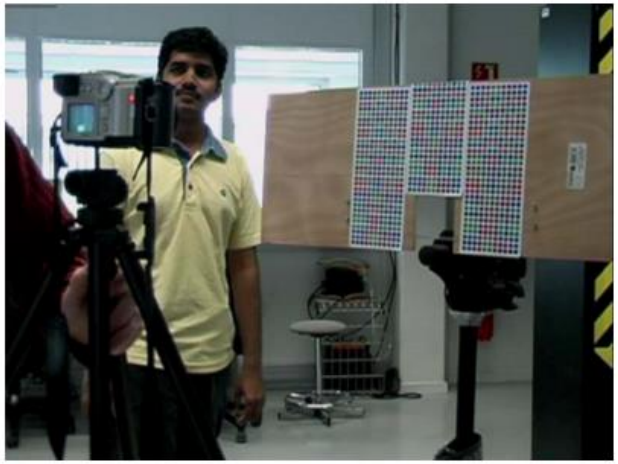


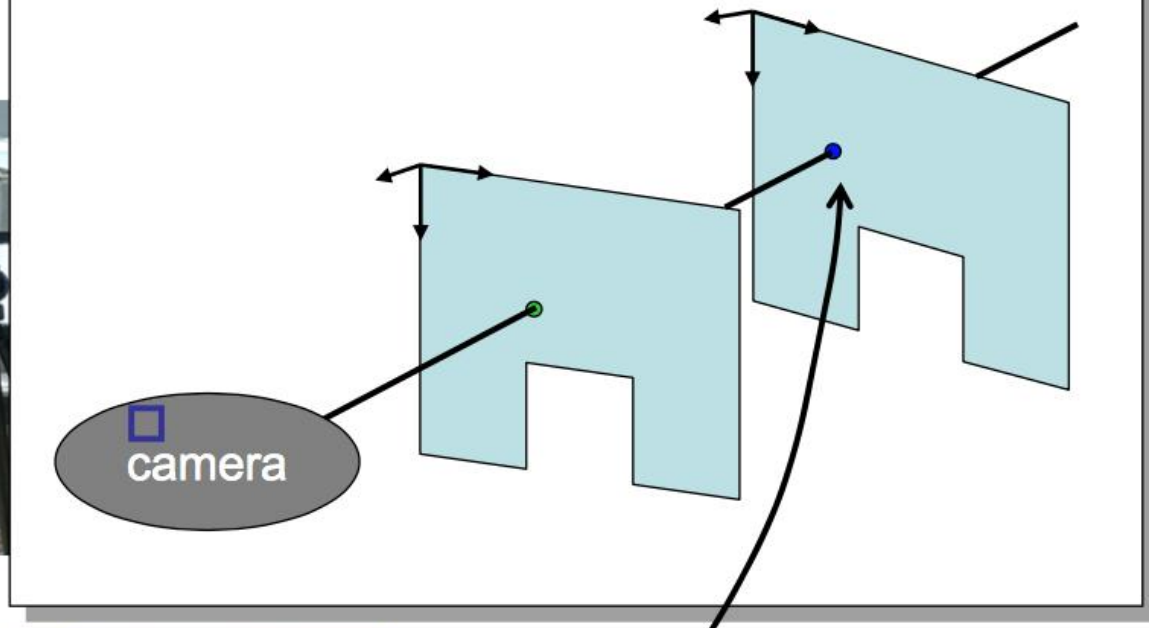


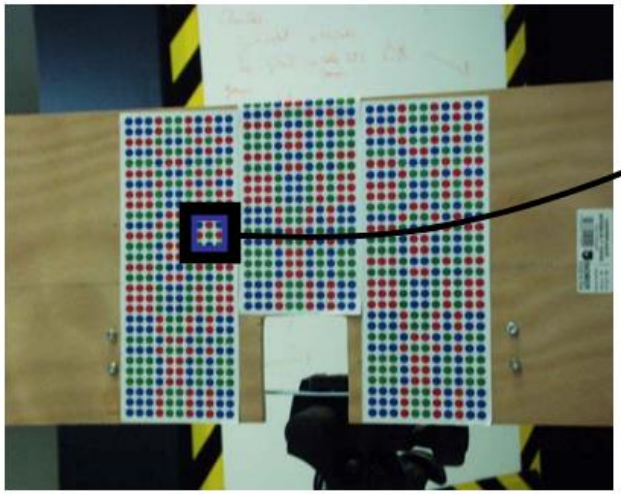


[1]From: PCV'06 - Tutorial on Modeling and Analysing Images of Generic Cameras, P. Sturm

Approach using known motion:







With unknown motion : see P. Sturm, S. Ramalingam, ECCV'04

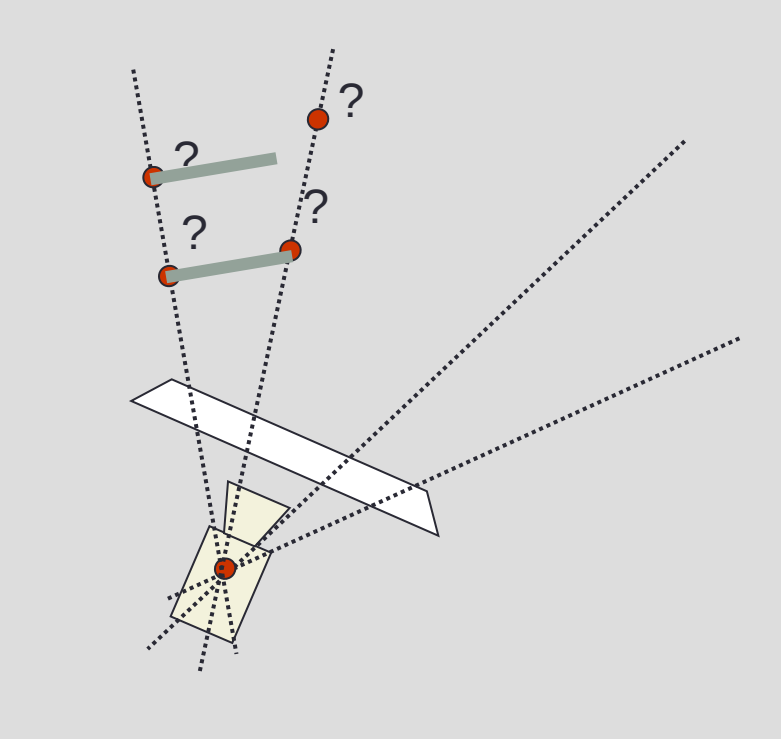
[1]From: PCV'06 - Tutorial on Modeling and Analysing Images of Generic Cameras, P. Sturm

DE LA 2D A LA 3D

I. La 3D à partir d'une seule image (SVP & NSVP)
II. La 3D à partir d'un système de vision stéréoscopique
III. La 3D à partir du « Structure From motion »

Cas SVP

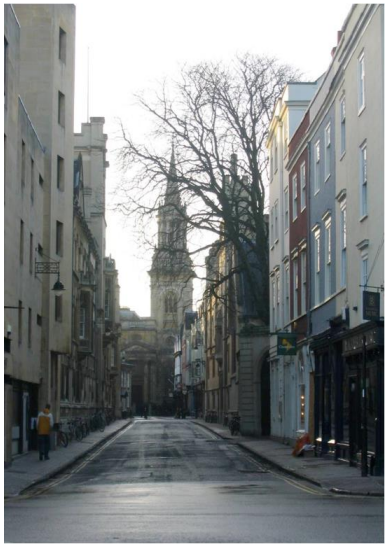
Problème mal posé sans connaissance a priori sur les objets



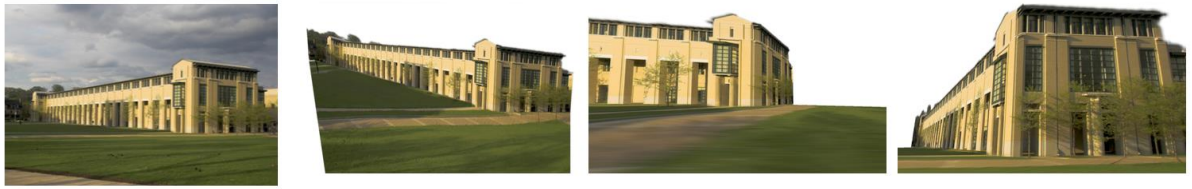
Si objet de taille connue : ok

Cas SVP

Problème mal posé sans connaissance a priori sur les objets



Scène constituées de plans, points de fuites… ok

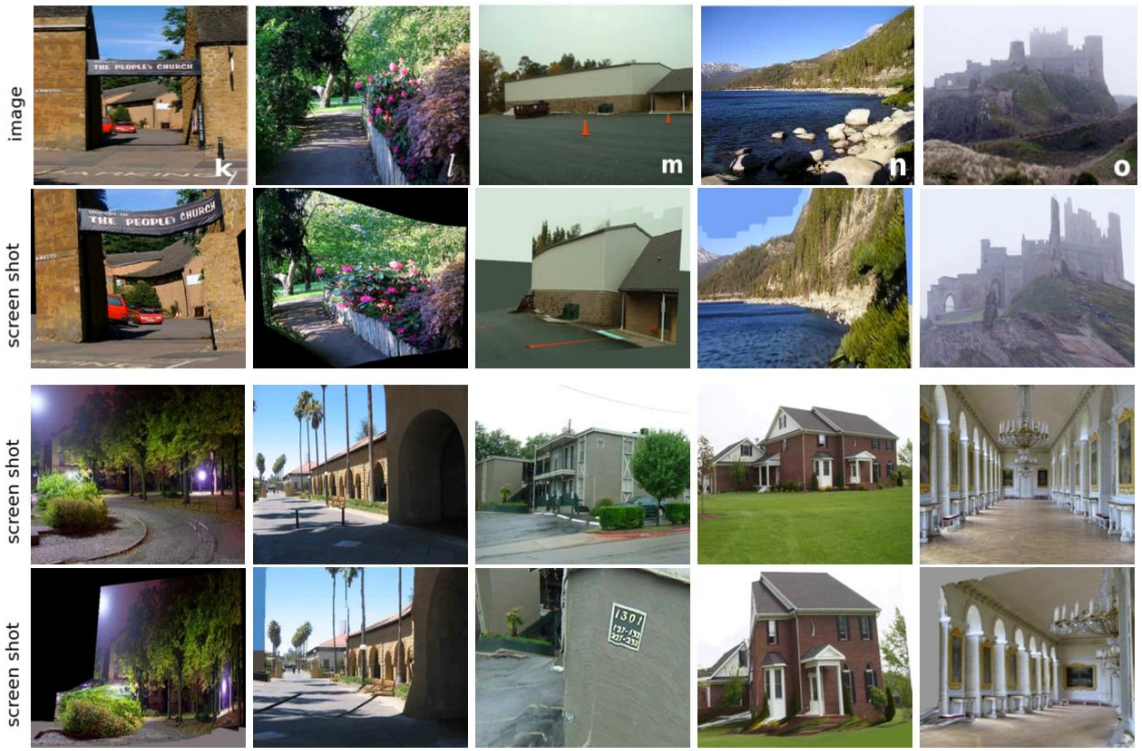


D. Hoiem, A.A.Efros, M. Hebert, « Automatic Photo Pop-up », SIGGRAPH 2005

Cas SVP

Problème mal posé sans connaissance a priori sur les objets : Learning

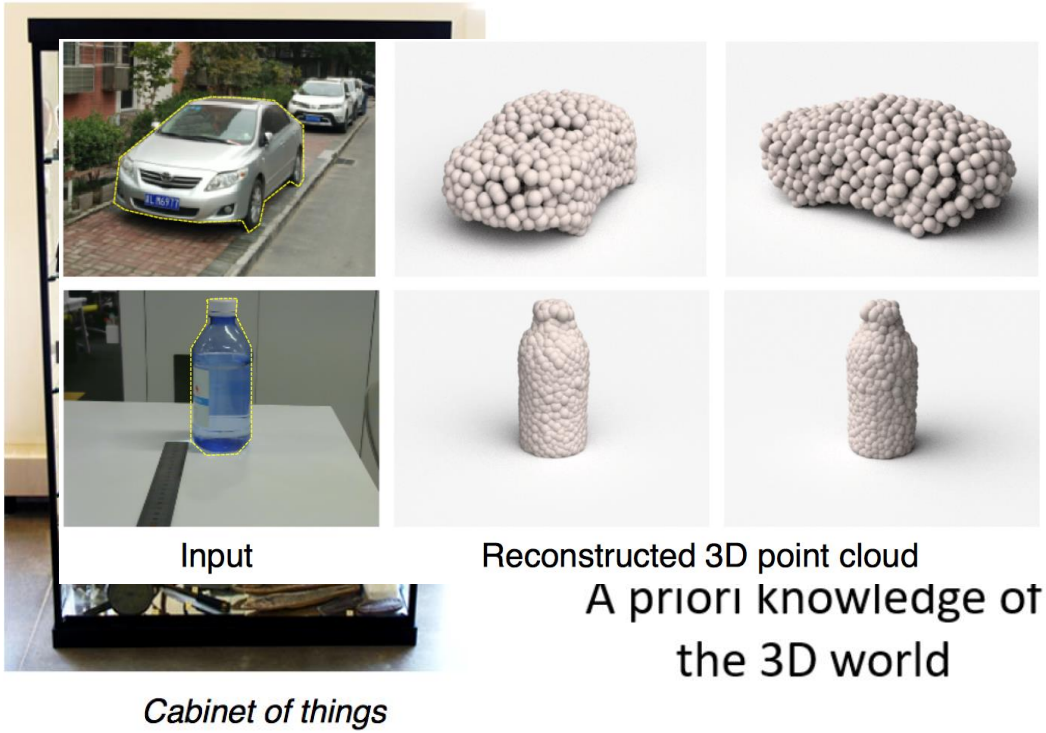
Scène constituées de plans



A. Saxena, M. Sun, A.Y. Ng « Make 3D: Learning 3D scene Structure From a Single Still Image », PAMI 2009

Cas SVP

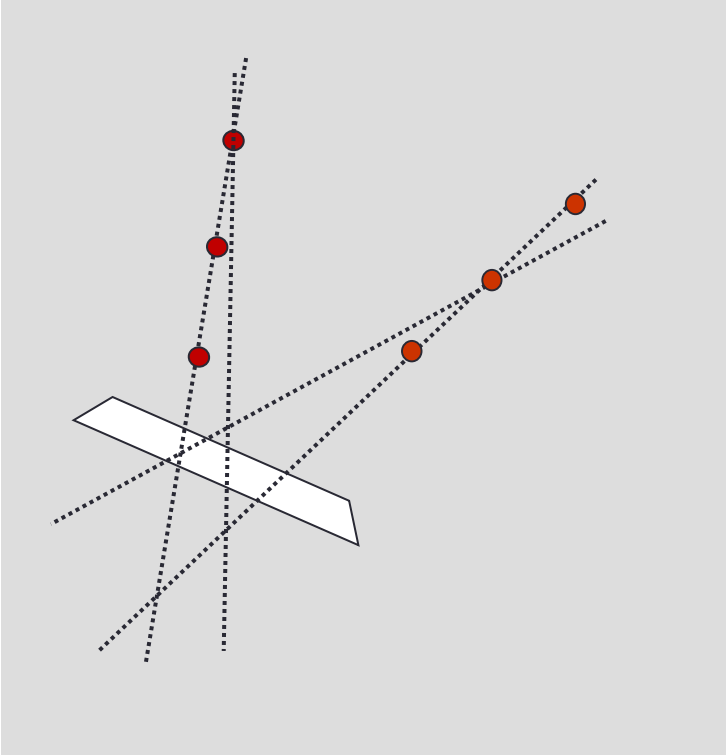
Problème mal posé sans connaissance a priori sur les objets : Deep Learning



H. Su, H. Fan, L. Guibas, « A point Set Generation Network for 3D Object Reconstruction from a single Image », CVPR 2017

Cas NSVP

Un point 3D peut être vu par 2 lignes de vue différentes



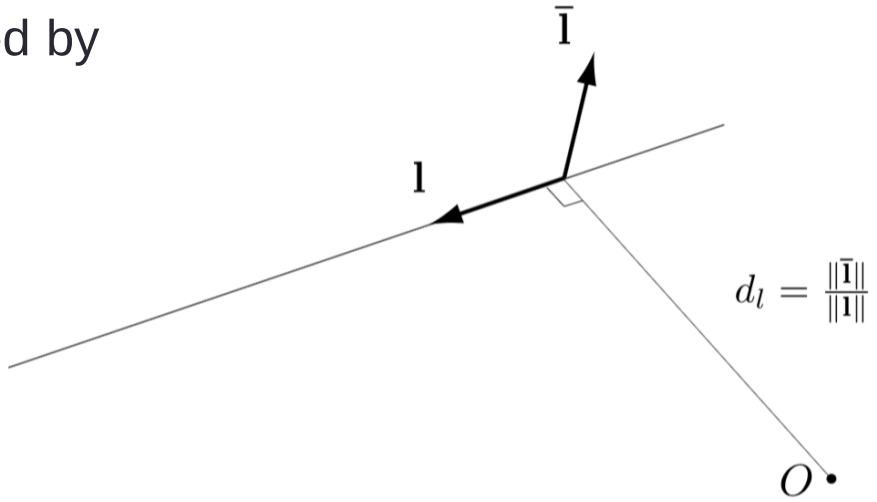
Backgrounds : Plücker coordinates

A 3D line $\boldsymbol{L} \in \mathbb{P}^5$ can be represented by 2 vectors :

$$\boldsymbol{L} = \left(l^T, \bar{l}^T\right)^T$$

$l \in \mathbb{R}^3$ direction vector

$\bar{l} \in \mathbb{R}^3$ moment vector.



Properties :

- A point in $\mathbb{P}^5$ is a line iif $l^T.\bar{l} = 0$: Plücker Identity
- 2 3d lines $\boldsymbol{L_1}$ and $\boldsymbol{L_2}$ intersect iif :

$$side(\boldsymbol{L_1}, \boldsymbol{L_2}) = l_1^T.\bar{l}_2 + l_2^T.\bar{l}_1 = \boldsymbol{L_1^T W L_2} = 0$$

$$W = \begin{pmatrix} 0_{3\times3} & I_{3\times3} \\ I_{3\times3} & 0_{3\times3} \end{pmatrix}$$

3D line reconstruction 4 pt algorithm

An image point ε defined a 3D line :

$$\boldsymbol{\Xi} = \left(\xi^T, \bar{\xi}^T\right)^T$$

If this image point corresponds to a 3D point on $\boldsymbol{L}$:

$$side(\boldsymbol{L}, \boldsymbol{\Xi}) = \boldsymbol{L}^T W \boldsymbol{\Xi} = 0$$

A 3D line $\boldsymbol{L}$ has 4 dof

Using 4 image points, $\boldsymbol{L}$ is solution of :

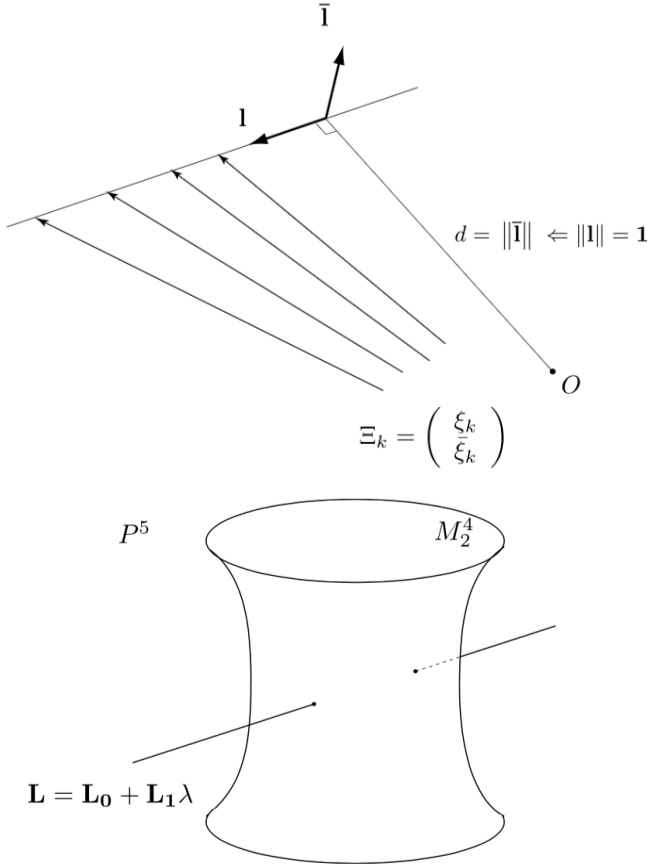
$$A\boldsymbol{L} = \begin{pmatrix} \bar{\xi}_1^T & \xi_1 \\ \vdots & \vdots \\ \bar{\xi}_4^T & \xi_4 \end{pmatrix} \boldsymbol{L} = 0$$

A has rank 4 $\Longrightarrow$

$$\boldsymbol{L} = \boldsymbol{L_0} + \lambda \boldsymbol{L_1}$$

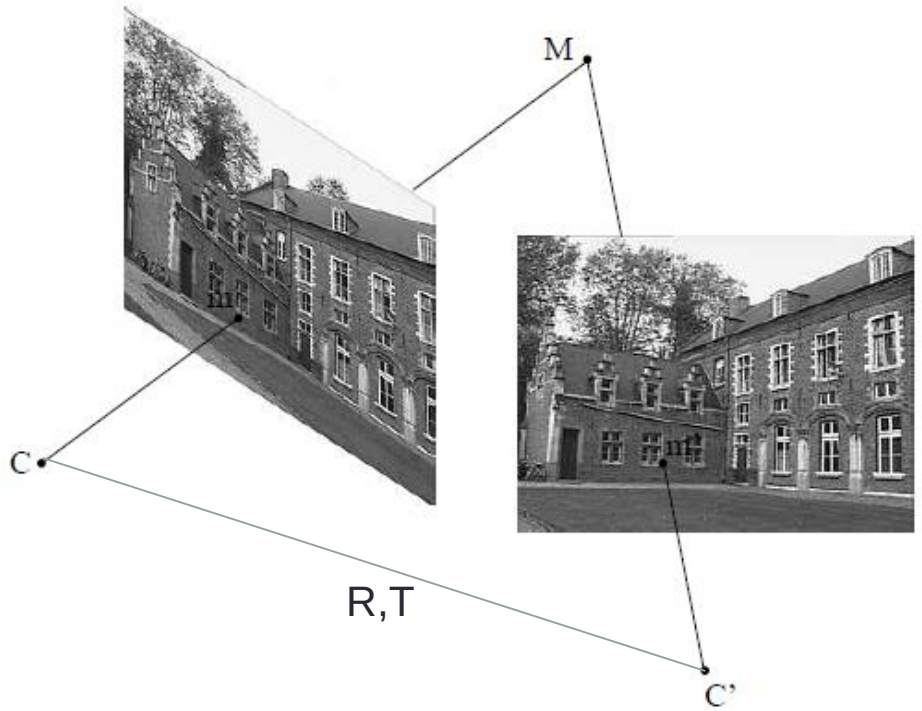
λ is obtained thanks to Plücker Identity :

$$side(\boldsymbol{L}, \boldsymbol{L}) = 0$$



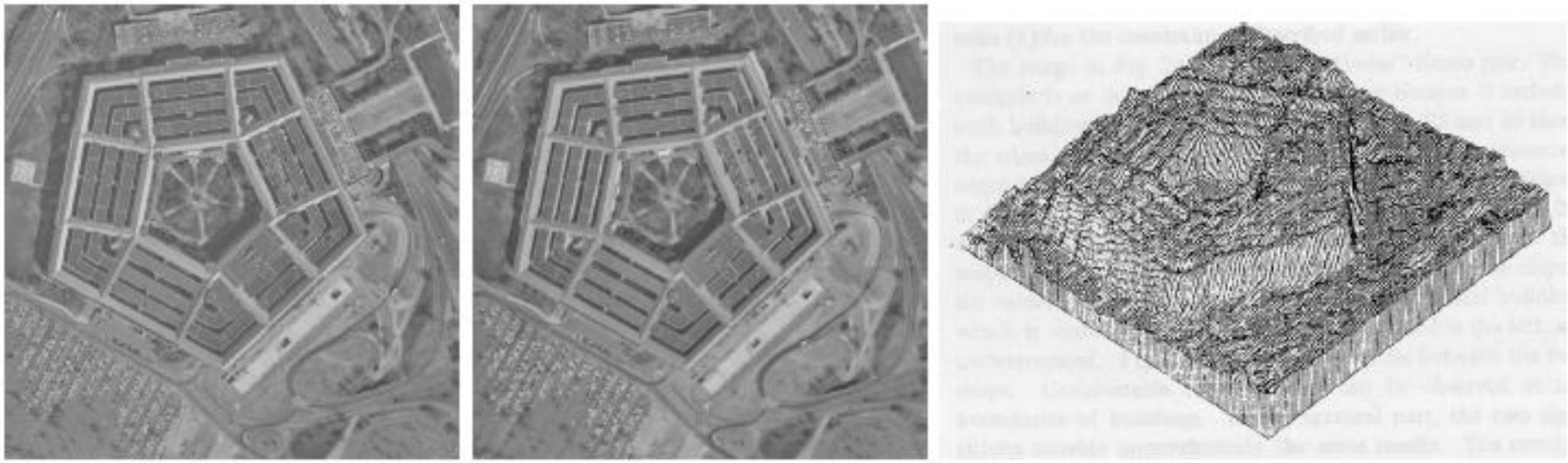
S. Teller, M. Hohmeyer. Determining the lines through four lines. *Journal of graphics tools*, *1999*
S. Gasparini, V. Caglioti. Line localization from single catadioptric images. *IJCV 2011*

Système stéréoscopique



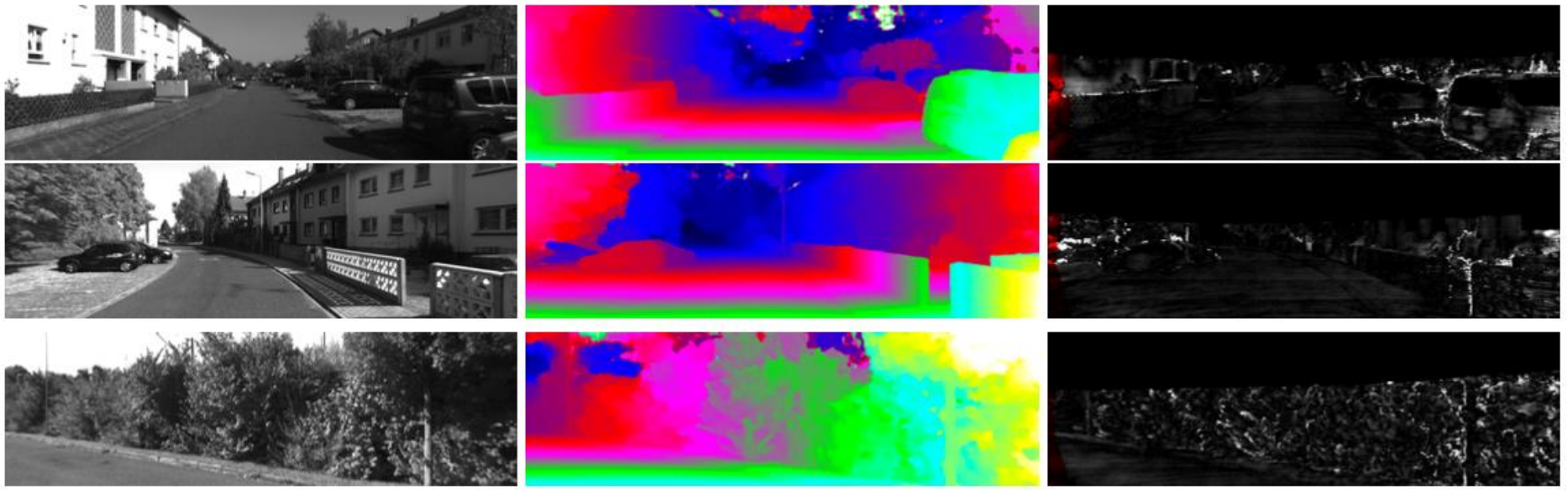
- Avec plus d'une image, nous pouvons reconstruire les points 3D par triangulation
- Avec 2 images : stéréoscopie

Système stéréoscopique



Y. Otha, T. Kanade, Stereo by intra and inter-scanline search using dynamic programming. IEEE PAMI 1985

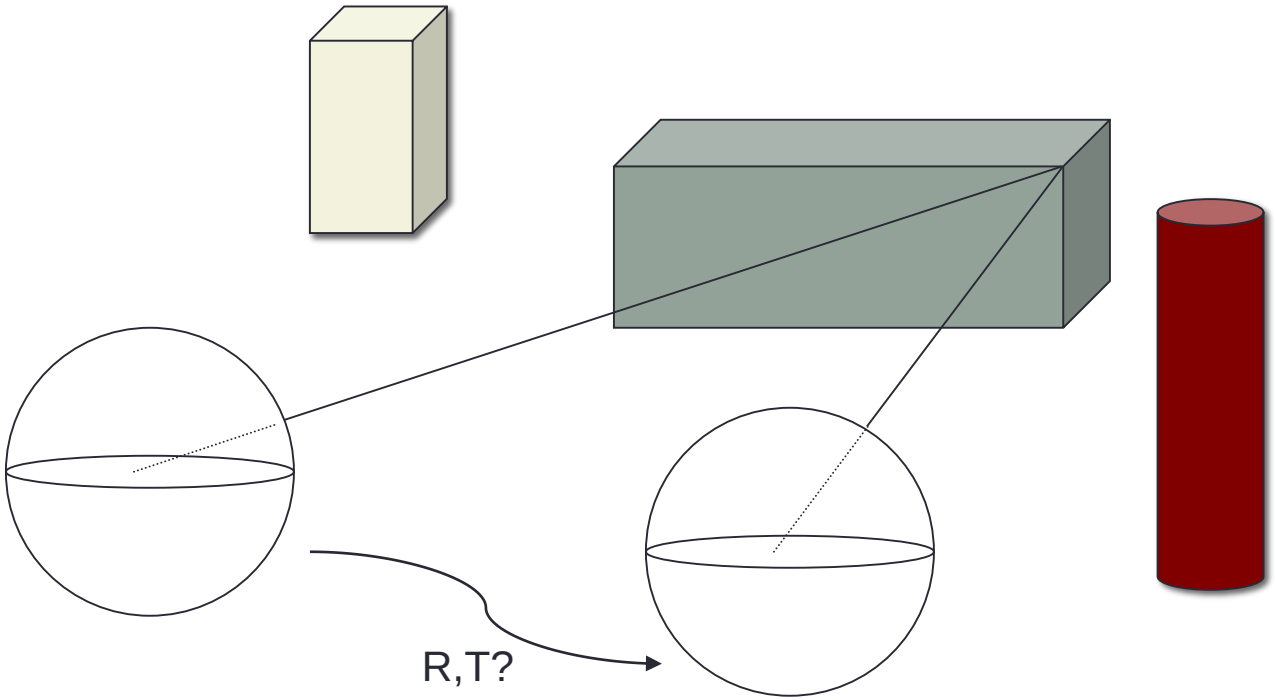
Système stéréoscopique



(a) KITTI 2012 test data qualitative results. From left: left stereo input image, disparity prediction, error map.

A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, A. Bry : End to end learning of geometry and contect for deep stereo regression. Arxiv.org 2017

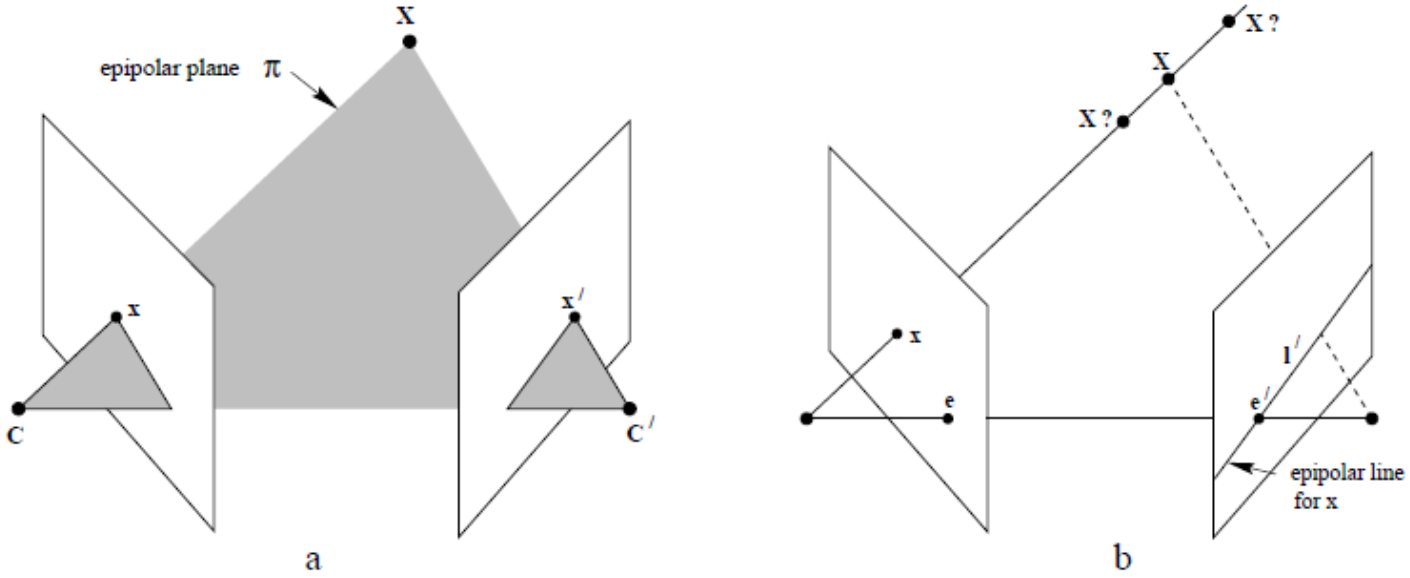
3D à partir du mouvement



Géométrie épipolaire

❑ The **epipolar geometry** is the intrinsic projective geometry between two views.

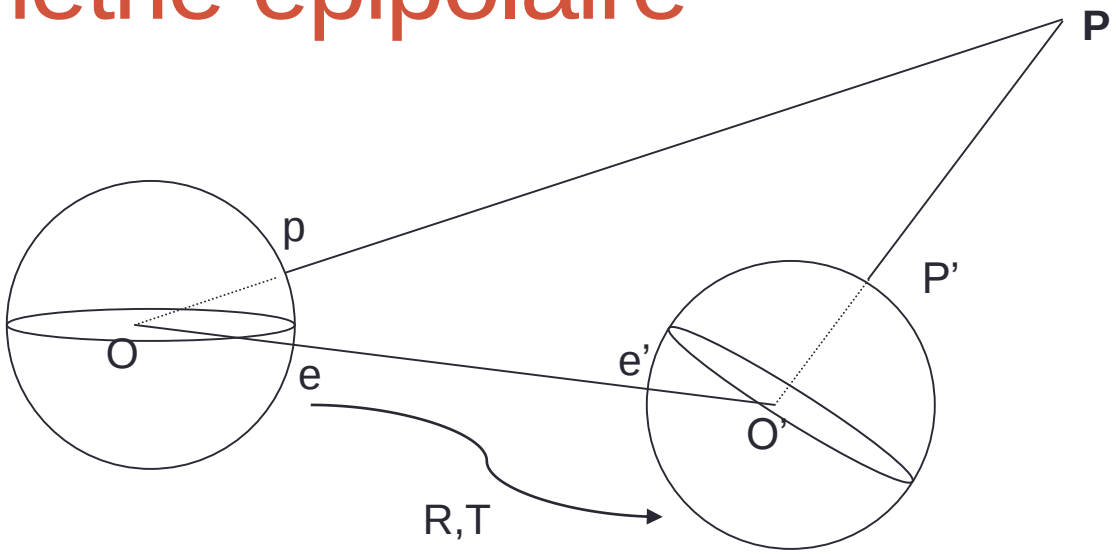
❑ It is independent of scene structure, and only depends on the cameras' internal parameters and relative pose.



❖The camera centres, 3D point X, and its images x and x′ lie in a common plane

❖The image of X in the second view must lie on l′.

Géométrie épipolaire



Op, OO', O'p' are co-planar → $p' \cdot \left[T \times Rp\right] = 0$

$$p' E p = 0 \qquad E = \left[T_\times\right] R \qquad \left[T_\times\right] = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$$

Matrice Essentielle
(Longuet-Higgins, 1981)

Matrice essentielle : propriétés

$E = [T]_X R$

Donc

- E est de rang 2

- $\det(E) = 0$
- $E = U \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$
- $EE^T E - \frac{1}{2}\mathrm{Trace}(EE^T)E = 0$

Algorithme des 8 points

$$p' E p = 0 \qquad E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}$$

$$p = (x, y, 1) \quad p' = (x', y', 1)$$

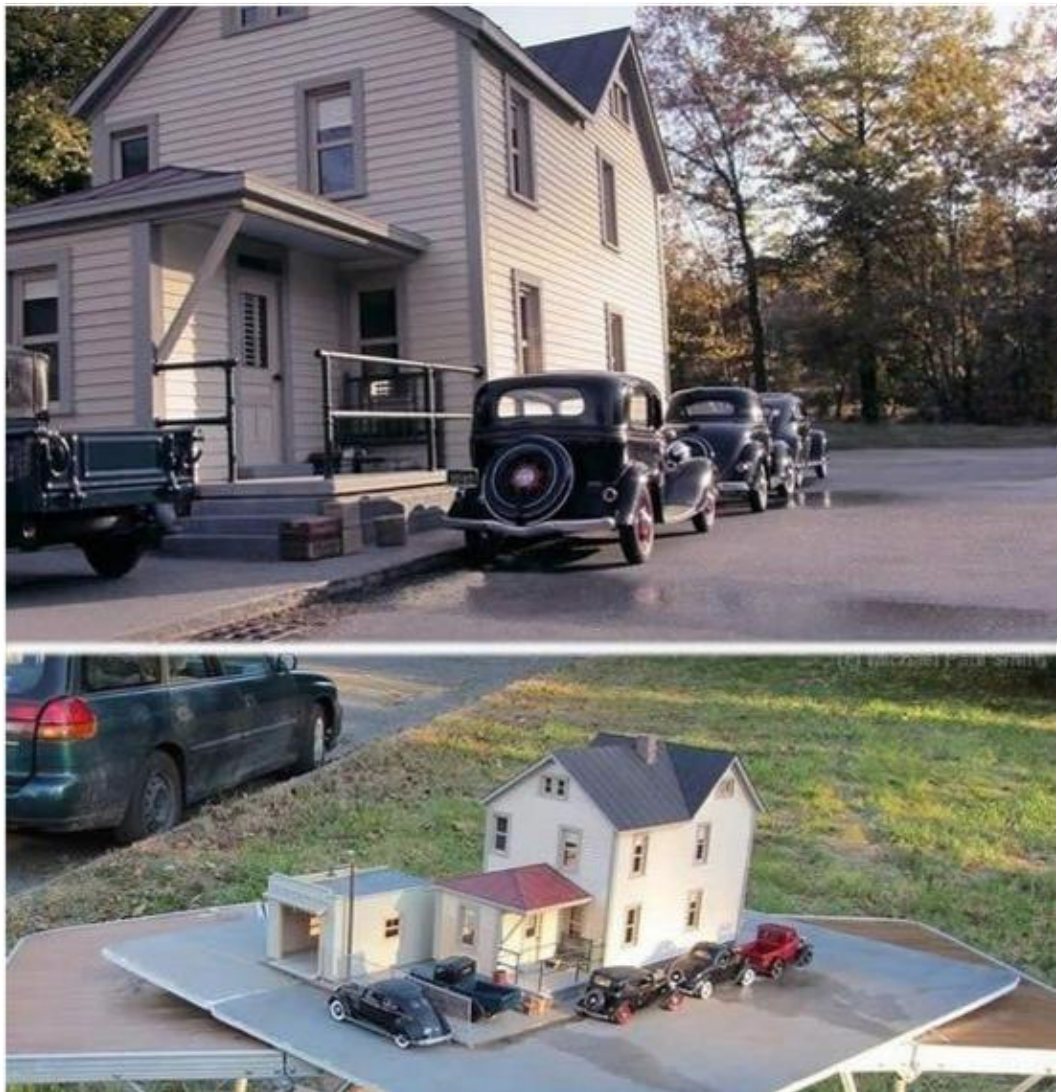
$$x'xe_{11} + x'ye_{12} + x'e_{13} + y'xe_{21} + y'ye_{22} + y'e_{23} + xe_{31} + ye_{12} + e_{33} = 0$$

Pour n points en correspondance:

$$Ae = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} e = 0$$

Algorithme des 8 points

- Nous disposons d'un système homogène Ae=0
- e ne peut être déterminé qu'à l'échelle près,



Algorithme des 8 points

- Nous disposons d'un système homogène Ae=0
- e ne peut être déterminé qu'à l'échelle près,nous avons donc que 8 inconnues à déterminer.
- 8 points en correspondance sont nécessaires
- En pratique, nous utilisons plus de 8 points et résolvons :

$$\min_{e} \sum_{i=1}^{n} \|Ae\|^2 \quad \text{such as} \quad \|e\| = 1$$

→ La solution étant la plus petite valeur singulière de A

Algorithme des 5 points

$$Ae = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} e = 0$$

Avec n = 5, le problème est sous-déterminé (matrice 5 *9) :

$$E = xX + yY + zZ + wW$$

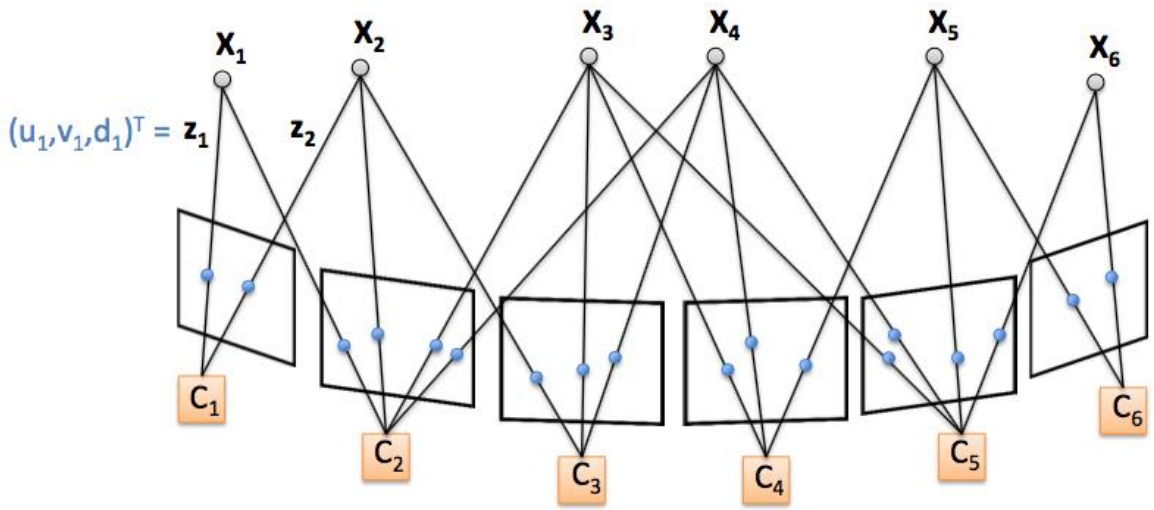
mais

$$\det(E) = 0$$

$$EE^TE - \frac{1}{2}\mathrm{Trace}(EE^T)E = 0$$

D. Nister, « An efficient solution to the five-point relative pose problem », PAMI 2004

Ajustement de faisceaux



$$\min_{C_{i_k}, \mathbf{X}_{j_k}} \sum_{k=1}^{K} ||\pi(\mathcal{T}^{-1}(C_{i_k}, \mathbf{X}_{j_k})) - (u_k, v_k)^\top||^2$$

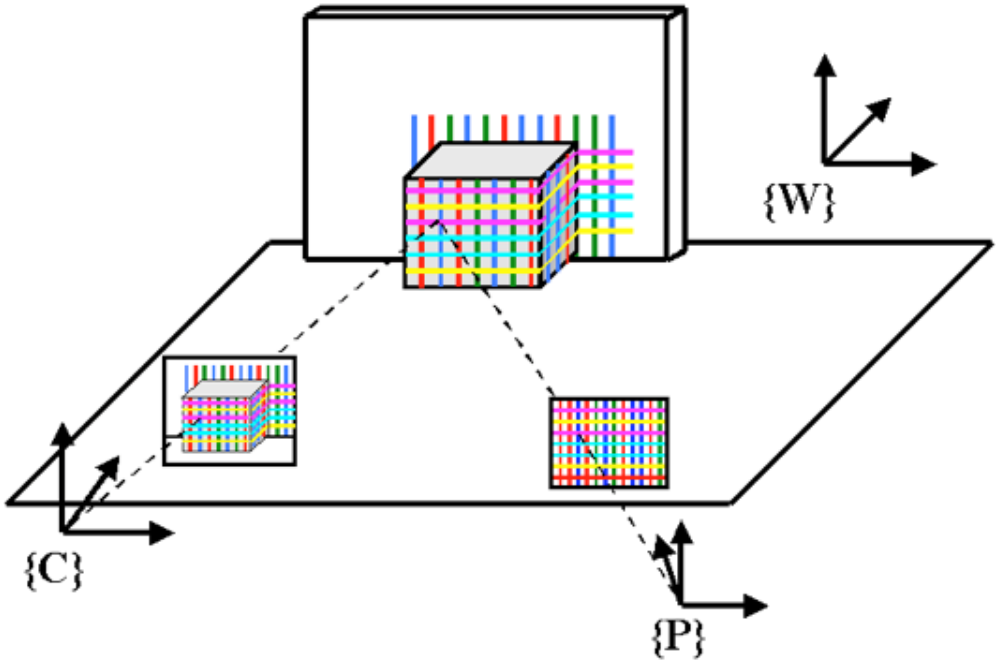
R. Maier, J. Sturm, D. Cremers, Submap-based bundle adjustment for 3D reconstruction form RGB-D data, GCPR 2014

Y. Furukawa, J. Ponce, Accurate, Dense, and Robust Multi-View Stereopsis, PAMI 2010

LES CAMÉRAS 3D

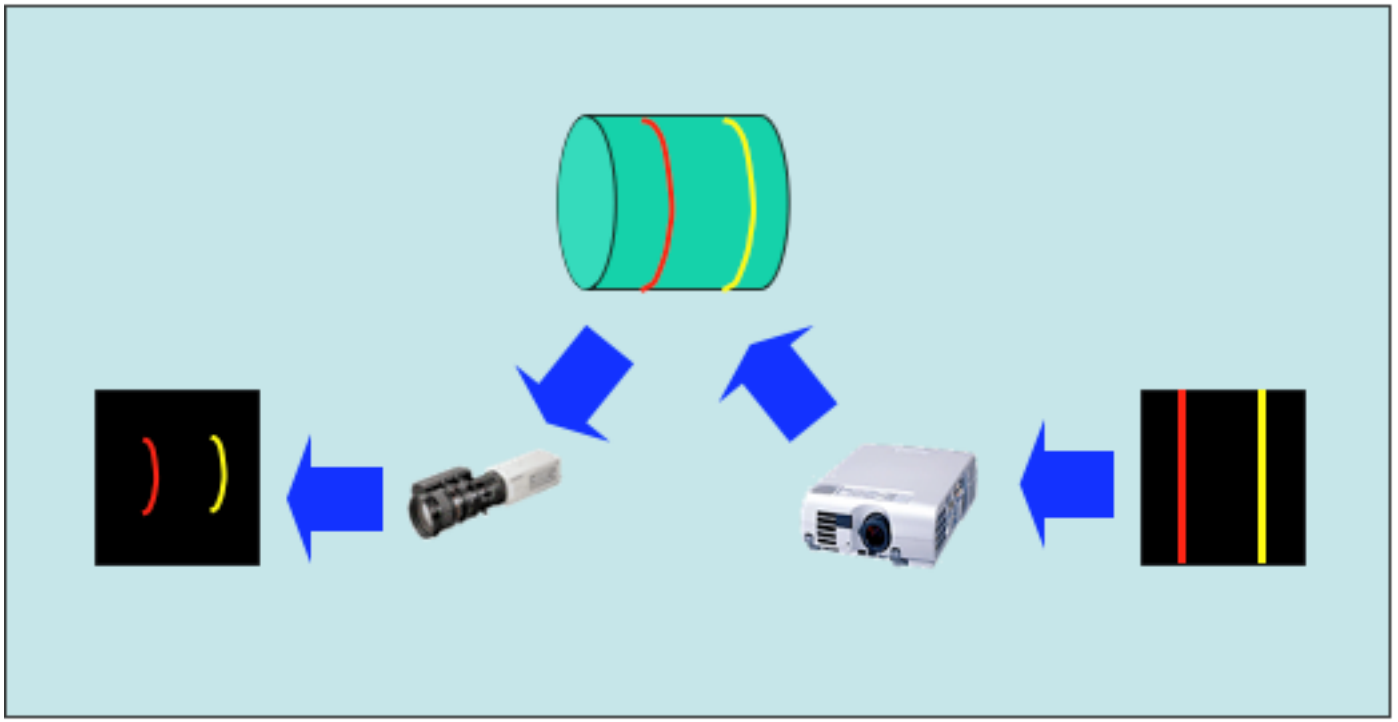
I. La lumière structurée
II. Les caméras temps de vol

Principe



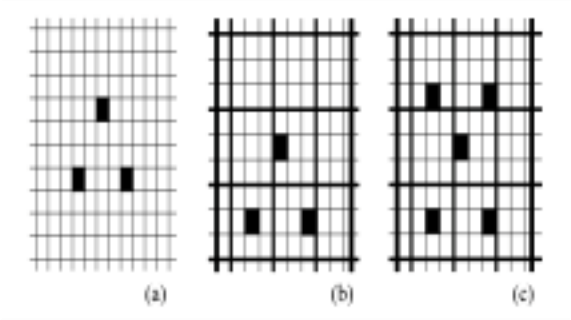
- L'une des caméras est remplacée par la projection d'un patron structuré sur la scène

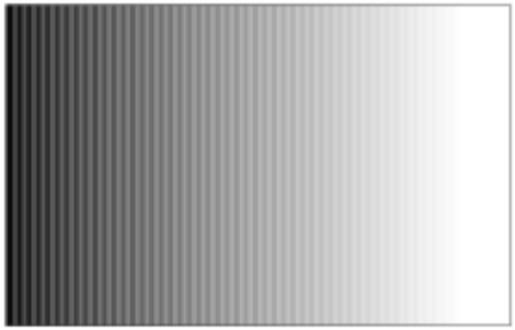
Principe

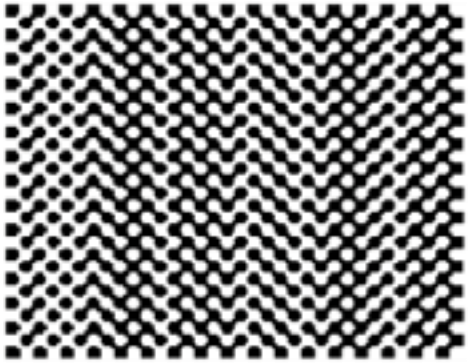


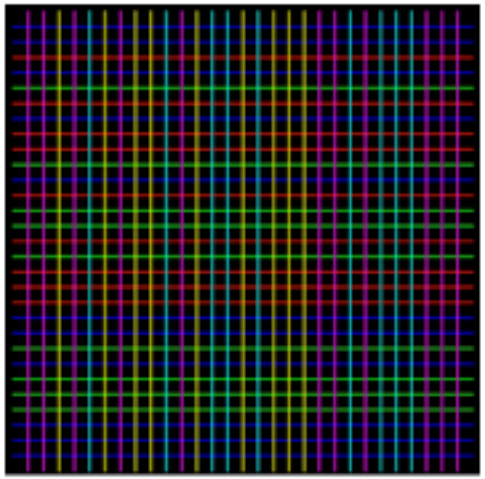
- Pour faciliter la mise en correspondance, le motif projeté est codé pour le reconnaître facilement dans l'image

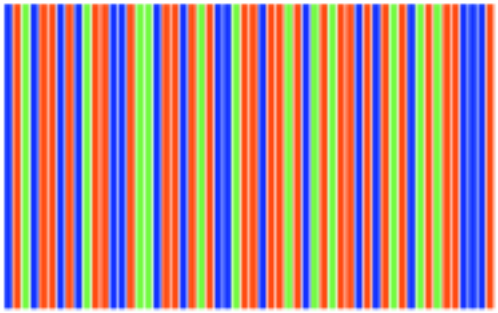
Principe

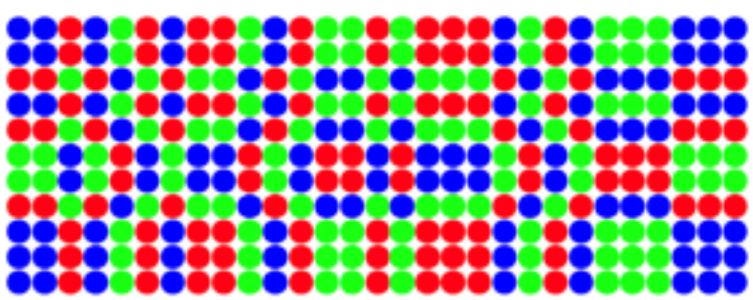




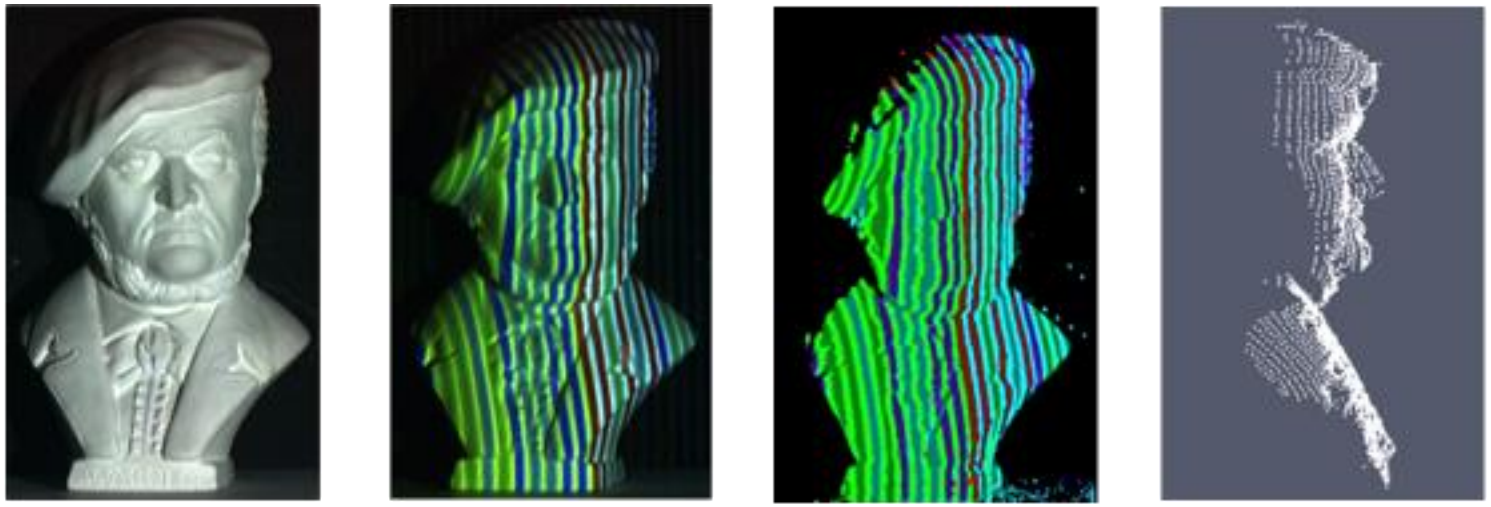








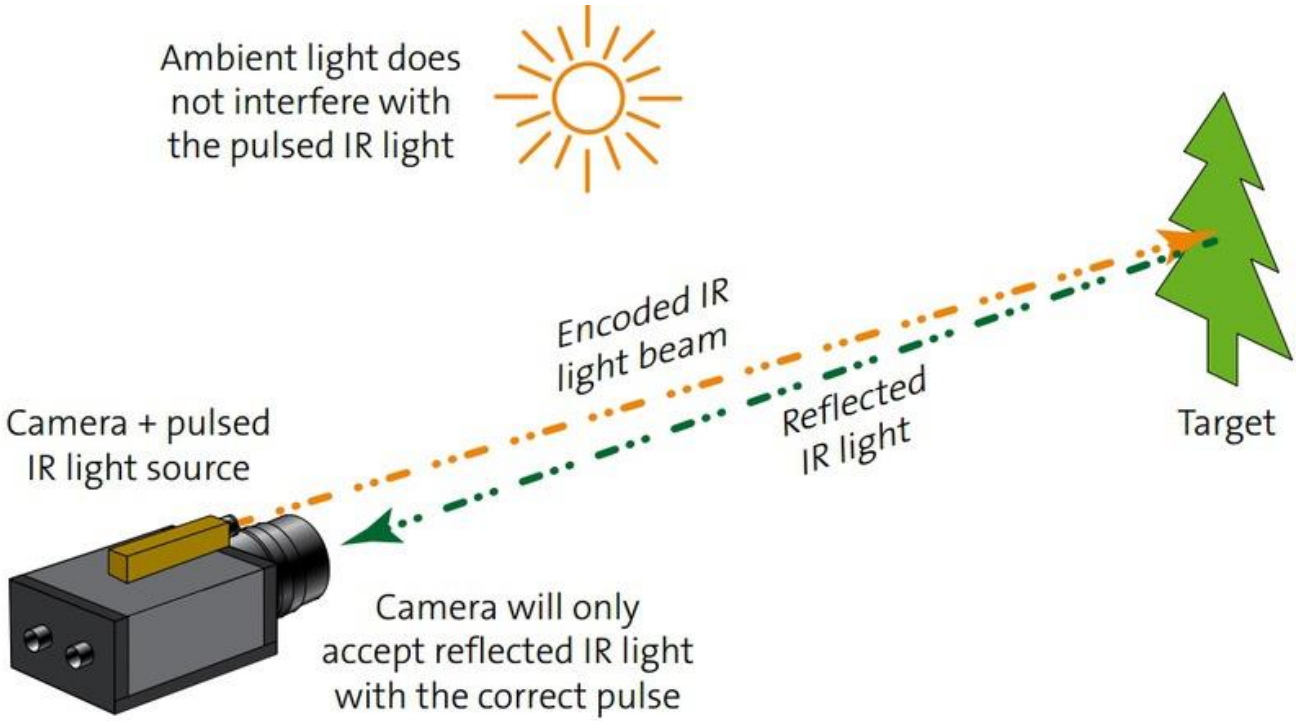
Résultats



J. Pages, J. Salvi, C. Collewet, J. Forest, Optimised De Bruijn patterns for one-shot shape acquisition, IVC 2005

Kinect 1
RGB : 640*480
Depth : 320*240

Principe



Stemmer-Imaging

SwissRanger 4000
176*144

Principe

Kinect 2
RGB : 1920*1080
Depth : 512*424

Kinect For Windows 1

Processed Image From Kinect

Kinect For Windows 2

Image via http://blogs.msdn.com

LES TRAVAUX DU LE2I

Travaux au sein du Le2i : contexte

- *Aujourd'hui, les méthodes de vision par ordinateur sont bien maîtrisées*
  - *SFM,*
  - *Reconstrcution 3D dense,*
  - *Estimation du mouvement,*
  - *Segmentation,*
  - *Reconnaissance d'objets,*
  - *…*

- *Pouvons-nous aller plus loin en utilisant d'autres informations?*
  - *Pouvons-nous améliorer l'estimation de pose et la segmentation 3D si nous connaissons : :*
    - *La structure 3D de la scène (caméra RGB-D)*
    - *L'attitude de la caméra (capteur inertiel)*
  - *Pouvons-nous améliorer la reconstruction 3D de la scène si nous savons que :*
    - *Les lignes sont parallèles/orthogonales*
    - *L'attitude de la caméra est connue*

Travaux au sein du Le2i

- 3D à partir d'une seule image (NSVP) + verticale connue



- 3D à partir d'homographie + verticale connue



- 3D à partir du « structure from motion » +3D frustre

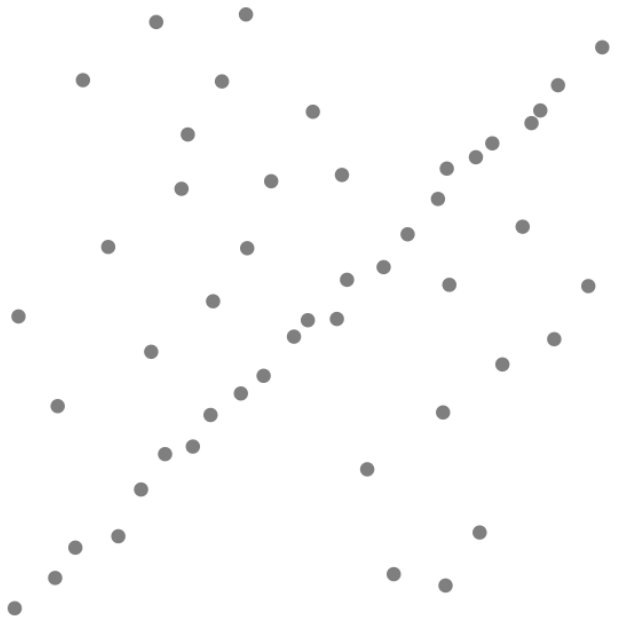


- 2D + 3D pour l'analyse de scènes 3D dynamiques

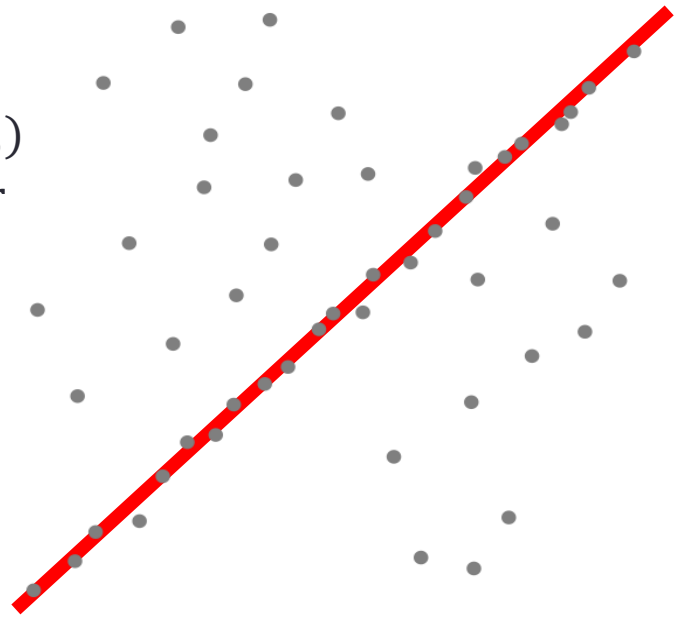

RANSAC (RANdom Sample Consensus)

$\boldsymbol{X} = (X, X_2 \dots X_M)$
M données

$\boldsymbol{\Theta} \leftarrow f(X_{i_1}, X_{i_2} \dots X_{i_n})$
n points pour calculer $\boldsymbol{\Theta}$



$\boldsymbol{\Theta} = (\Theta_1, \Theta_2 \dots \Theta_p)$
p variables pour
Décrire **X**



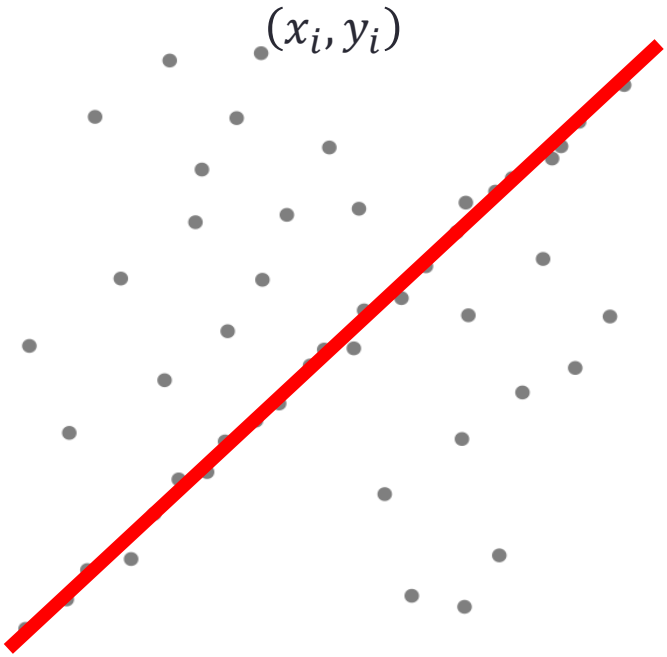
RANSAC (RANdom Sample Consensus)

Exemple : *line fitting*

$\boldsymbol{X} = (X_i = (x_i, y_i))_{i=1\ldots M}$

$ax_i + by_i = c \quad \forall\, (x_i, y_i)$

$\boldsymbol{\Theta} = (a, b, c)$



Combien de points faut – il pour calculer $\boldsymbol{\Theta}$?

RANSAC (RANdom Sample Consensus)

Beaucoup de problèmes en Vision par Ordinateur peuvent être modélisés par :

$$\Theta = f(x_{i1}, x_{i2}, \cdots x_{in})$$

Avec $X = (x_1 \cdots x_M)$, M observations

n : nombre de données nécessaires pour estimer Θ

Si toutes les données sont bonnes, nous pouvons choisir n observations quelconques.

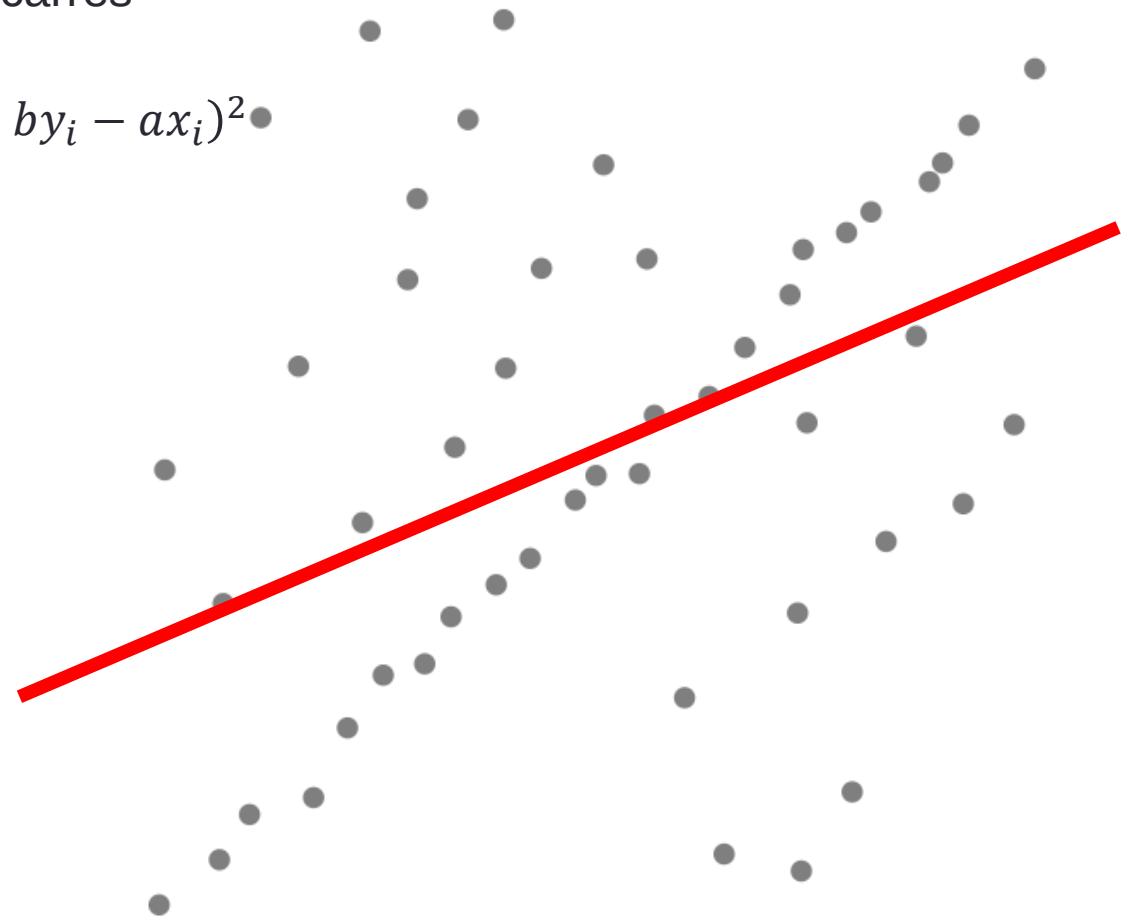
En pratique, des données sont souvent corrompues !

Comment s'assurer de la robustesse de la méthode?

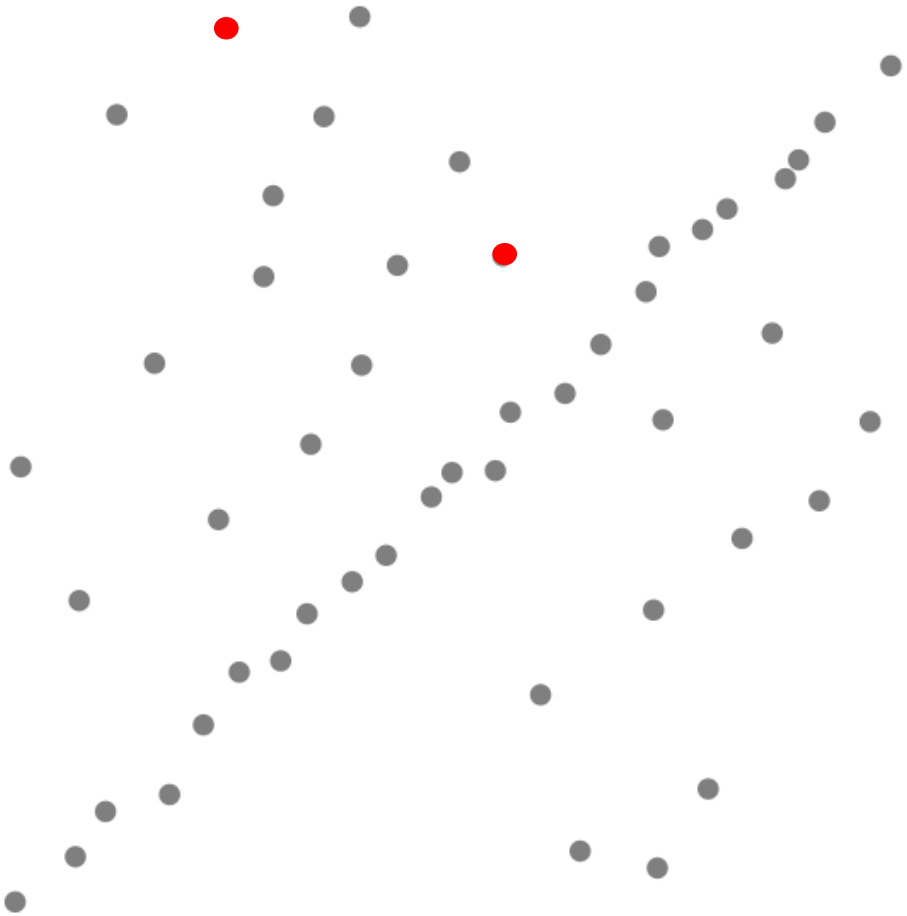
RANSAC (RANdom Sample Consensus)

Moindres carrés

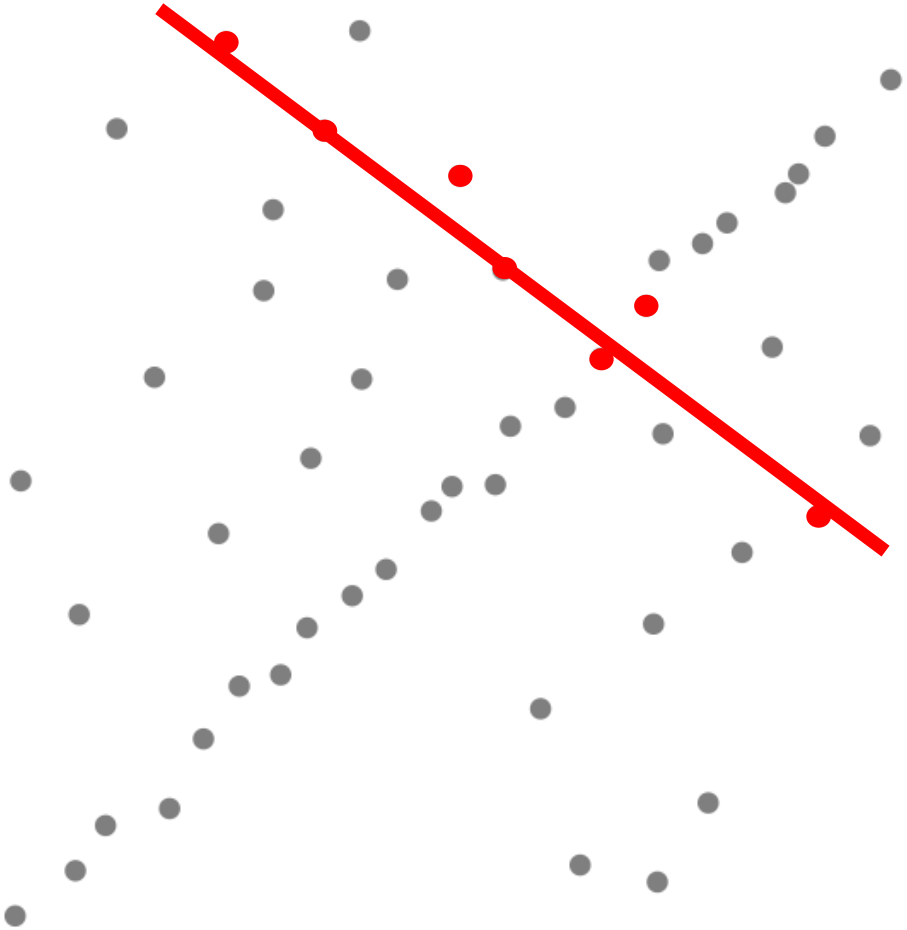
$$\Theta = \min_{(a,b)} \sum_{i=1}^{M} (1 - by_i - ax_i)^2$$



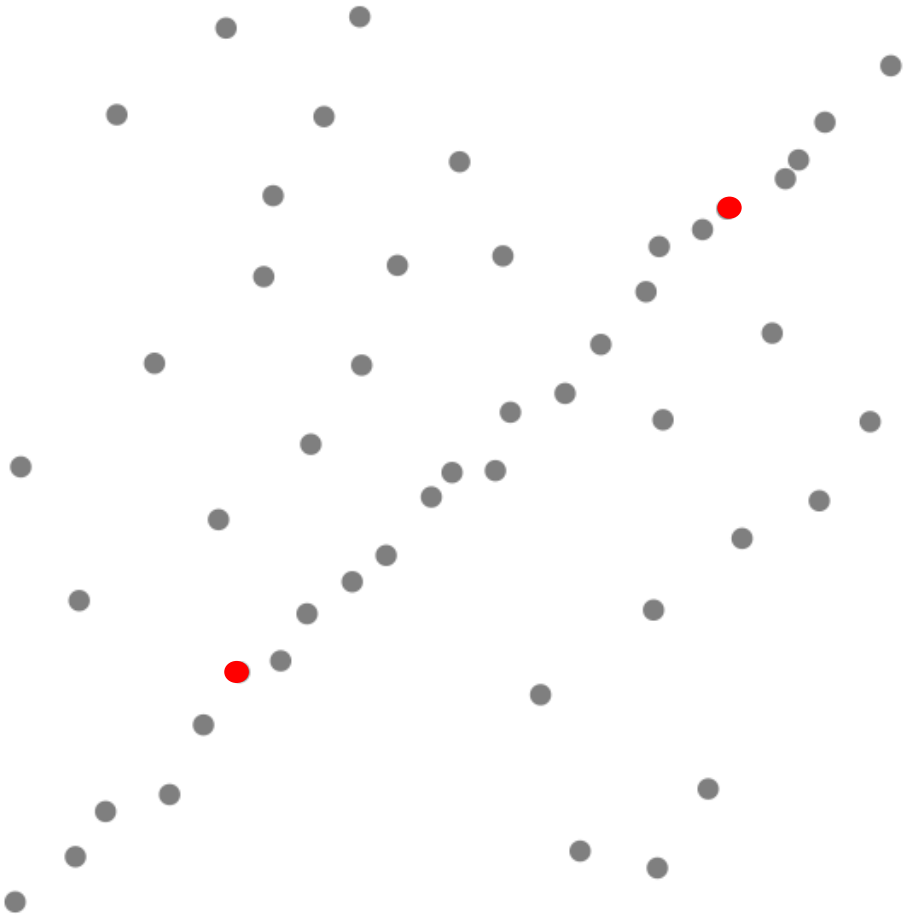
RANSAC (RANdom Sample Consensus)



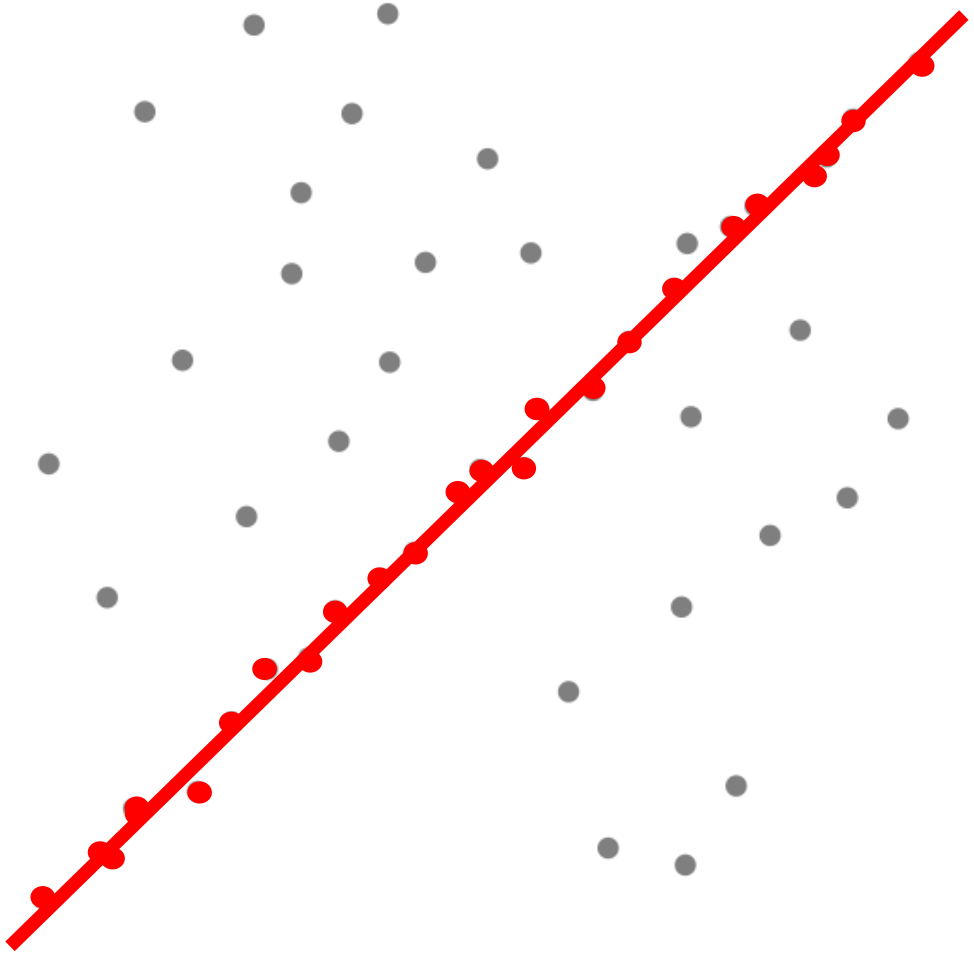
RANSAC (RANdom Sample Consensus)



RANSAC (RANdom Sample Consensus)



RANSAC (RANdom Sample Consensus)



RANSAC (RANdom Sample Consensus)

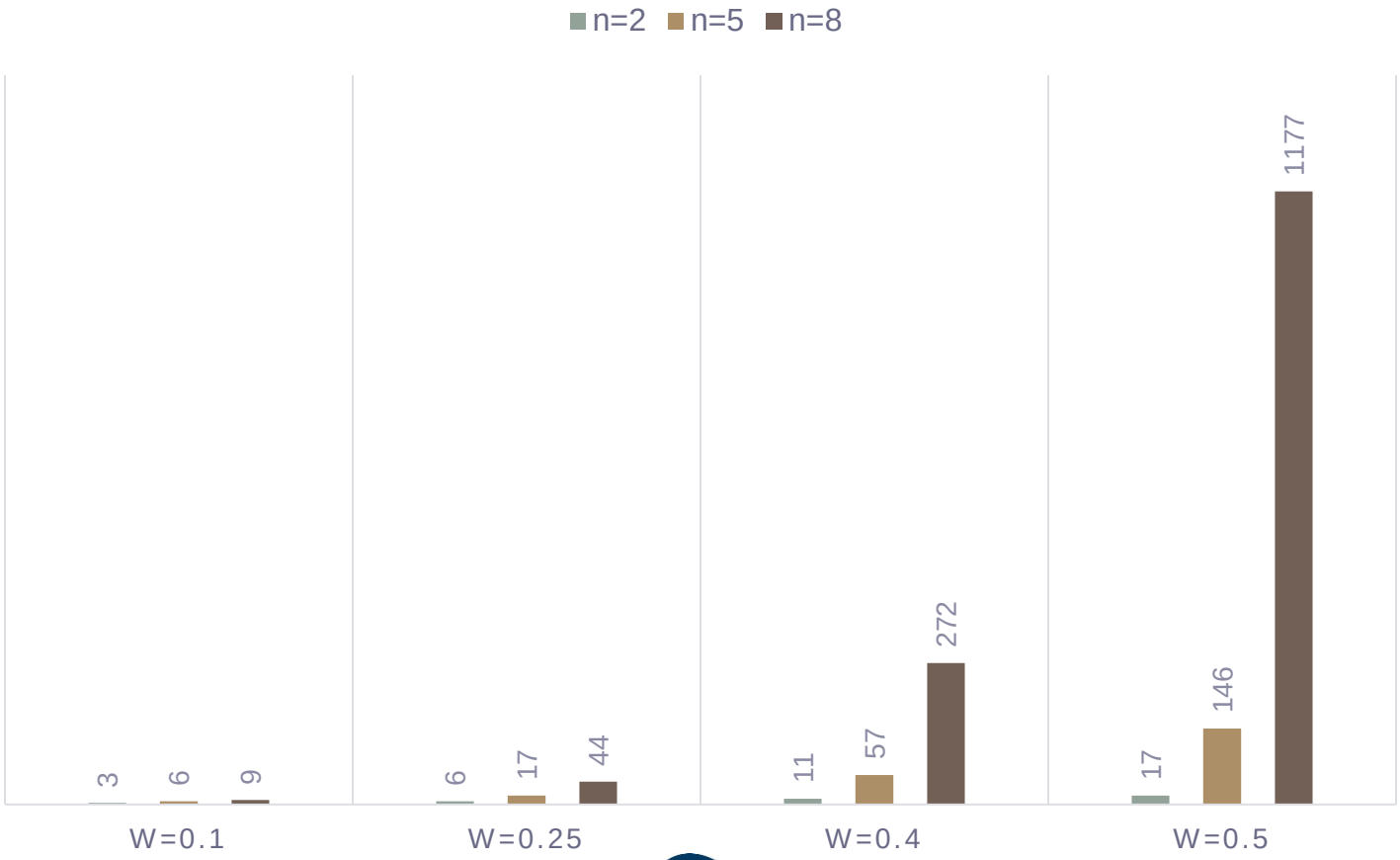
Pour assurer la convergence de l'algorithme, nous avons besoin de L itérations où:

$$\mathrm{L} = \frac{\log(1 - p_r)}{\log(1 - w^n)}$$

p_r = success probability
w = ratio of outliers
n = number of observations needed for estimating $\boldsymbol{\Theta}$

RANSAC (RANdom Sample Consensus)

Avec $p_r = 0.99\%$,



Théorème du rang

Let us consider a linear problem :

$$M\Theta = P$$

M, P : known matrices of size resp. $p \times n$ and $1 \times n$
Θ : unknown vector of size $1 \times p$

Theorem :
If $n = p$ and M is invertible (rank $M = p$) :

$$\Theta = M^{-1}P$$

If rank $M = q \leq p$:

$$\Theta = Y + \lambda_1 e_1 + \lambda_2 e_2 + \cdots + \lambda_{p-q} e_{p-q}$$

where $E = (e_1, e_2, \ldots e_{p-q})$ is a basis of the kernel of M and Y is solution of $MY = P$

Rq : Y and E can be obtained by a SVD decomposition of M

$$M = UDV^T, Y = VD^{-1}U^T P,$$

E : last columns of V which correspond to the null singular values of M

RECONSTRUCTION 3D A PARTIR D'UNE CAMÉRA NON CENTRALE ET D'UN CAPTEUR INERTIEL

J. Bermudez-Cameo, J.P. Barreto, G. Lopez-Nicolas, J. J. Guerrero, **Minimal solution for computing pairs of lines in non-central cameras**. In 12th Asian Conference on Computer Vision (ACCV 2014), November 2014, Singapour.

J. Bermudez-Cameo, C. Demonceaux, J.P. Barreto, G. Lopez-Nicolas, J. J. Guerrero, **Minimal solution for 3D line reconstruction in non-central systems.** Technical Report

Non central camera

LadyBug

Spherical GoPro

eleVRant

Samsung Gear 360

3D line reconstruction 4 pt algorithm

An image point ε defined a 3D line :

$$\boldsymbol{\Xi} = \left(\xi^T, \bar{\xi}^T\right)^T$$

If this image point corresponds to a 3D point on $\boldsymbol{L}$:

$$side(\boldsymbol{L}, \boldsymbol{\Xi}) = \boldsymbol{L}^T W \boldsymbol{\Xi} = 0$$

A 3D line $\boldsymbol{L}$ has 4 dof

Using 4 image points, $\boldsymbol{L}$ is solution of :

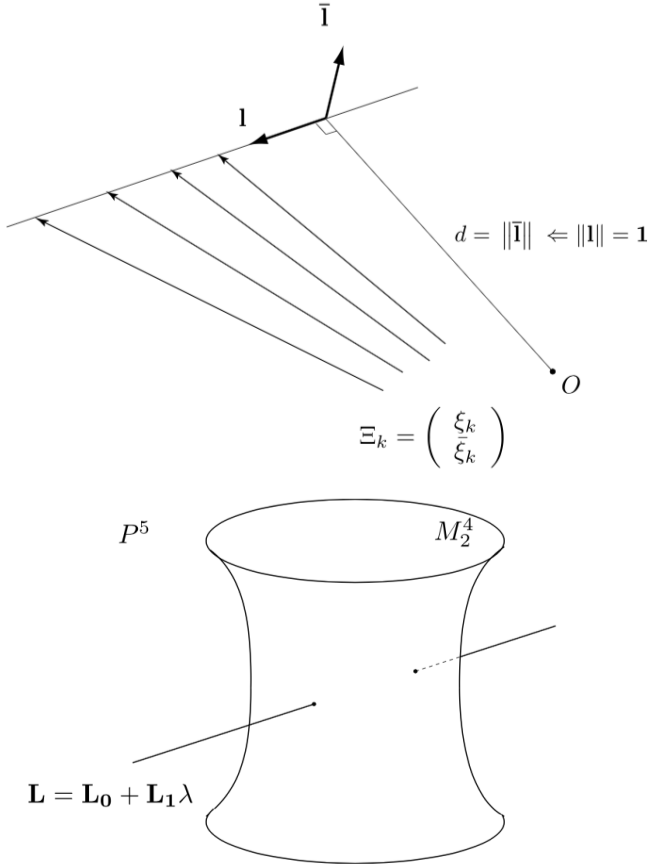
$$A\boldsymbol{L} = \begin{pmatrix} \bar{\xi}_1^T & \xi_1 \\ \vdots & \vdots \\ \bar{\xi}_4^T & \xi_4 \end{pmatrix} \boldsymbol{L} = 0$$

A has rank 4 $\Longrightarrow$

$$\boldsymbol{L} = \boldsymbol{L_0} + \lambda \boldsymbol{L_1}$$

λ is obtained thanks to Plücker Identity :

$$side(\boldsymbol{L}, \boldsymbol{L}) = 0$$



S. Teller, M. Hohmeyer. Determining the lines through four lines. *Journal of graphics tools*, *1999*
S. Gasparini, V. Caglioti. Line localization from single catadioptric images. *IJCV 2011*

3D line reconstruction 3 pt algorithm

Hypothesis : The 3D line $\boldsymbol{L}$ is parallel to a given plane $\boldsymbol{U} = (u_0, u^T)^T$

3 image points give us 3 linear equations:

$$\boldsymbol{\Xi}_k^T W \boldsymbol{L} = 0,\ k = 1..3$$

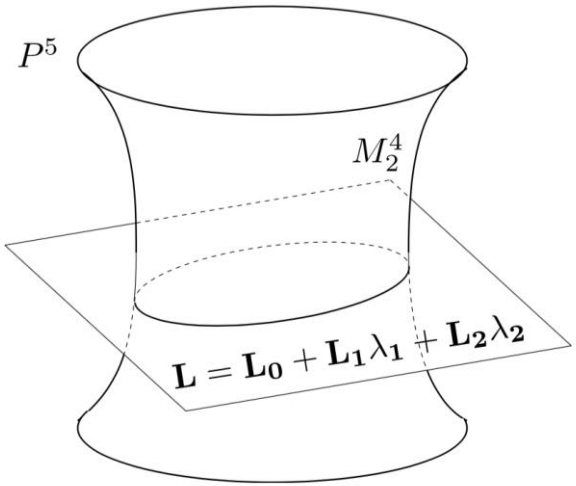
A has rank 3 $\Longrightarrow$

$$\boldsymbol{L} = \boldsymbol{L_0} + \lambda_1 \boldsymbol{L_1} + \lambda_2 \boldsymbol{L_2}$$

We need 2 other constraints

$side(\boldsymbol{L}, \boldsymbol{L}) = 0$ Plücker Identity

$u^T l = 0$



3D line reconstruction 2 pt algorithm

Hypothesis : The 3D line $\boldsymbol{L}$ reconstruction with known direction l

2 image points give us 2 linear equations:

$$\xi_k^T \bar{l} = -\bar{\xi}_k^T l,\ k = 1,2 \text{ and } side(\boldsymbol{L}, \boldsymbol{L}) = 0$$

$$\Rightarrow \begin{bmatrix} \xi_1^T \\ \xi_2^T \\ l^T \end{bmatrix} \bar{l} = \begin{pmatrix} -\bar{\xi}_1^T l \\ -\bar{\xi}_2^T l \\ 0 \end{pmatrix}$$

$$\Rightarrow \bar{l} = \begin{bmatrix} \xi_1^T \\ \xi_2^T \\ l^T \end{bmatrix}^{-1} \begin{pmatrix} -\bar{\xi}_1^T l \\ -\bar{\xi}_2^T l \\ 0 \end{pmatrix}$$

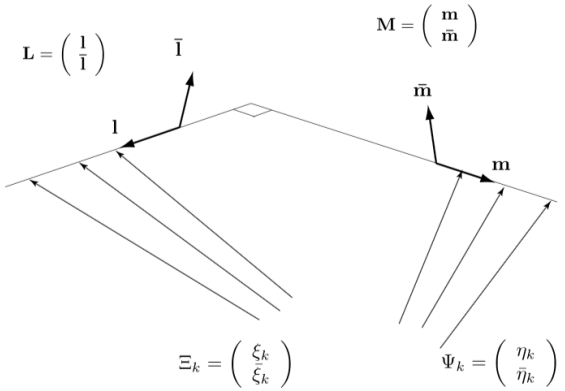
3D lines reconstruction 6 pt algorithm

Hypothesis : The 3D lines $\boldsymbol{L}$ and $\boldsymbol{M}$ are orthogonal

3 image points of $\boldsymbol{L}$ and 3 of $\boldsymbol{M}$ give us 6 linear equations:

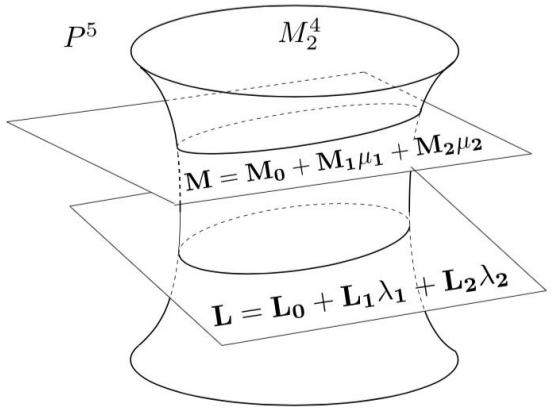
$$\boldsymbol{\Xi}_k^T W \boldsymbol{L} = 0, \boldsymbol{\Psi}_k^T W \boldsymbol{M} = 0 \; k = 1..3$$

$$\Rightarrow \begin{cases} \boldsymbol{L} = \boldsymbol{L_0} + \lambda_1 \boldsymbol{L_1} + \lambda_2 \boldsymbol{L_2} \\ \mathbf{M} = \boldsymbol{M_0} + \mu_1 \boldsymbol{M_1} + \mu_2 \boldsymbol{M_2} \end{cases}$$



We need 4 other constraints

$$side(\boldsymbol{L}, \boldsymbol{L}) = 0$$
$$side(\boldsymbol{M}, \boldsymbol{M}) = 0$$
$$l^T m = 0$$
$$\boldsymbol{L}^T W \boldsymbol{M} = 0$$



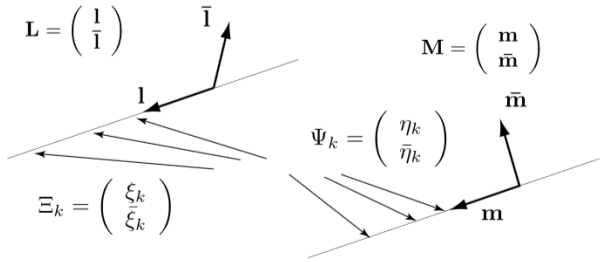
3D lines reconstruction 6 pt algorithm

Hypothesis : The 3D lines $\boldsymbol{L}$ and $\boldsymbol{M}$ are parallel

3 image points of $\boldsymbol{L}$ and 3 of $\boldsymbol{M}$ give us 6 linear equations:

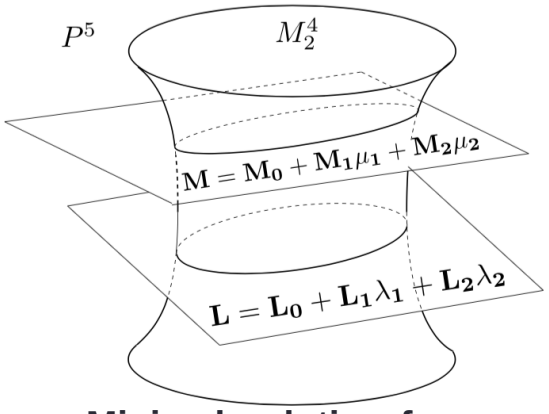
$$\boldsymbol{\Xi}_k^T W \boldsymbol{L} = 0,\ \boldsymbol{\Psi}_k^T W \boldsymbol{M} = 0\ k = 1..3$$

$$\Rightarrow \begin{cases} \boldsymbol{L} = \boldsymbol{L_0} + \lambda_1 \boldsymbol{L_1} + \lambda_2 \boldsymbol{L_2} \\ \mathbf{M} = \boldsymbol{M_0} + +\mu_1 \boldsymbol{M_1} + \mu_2 \boldsymbol{M_2} \end{cases}$$



We need 4 other constraints

$$side(\boldsymbol{L}, \boldsymbol{L}) = 0$$
$$side(\boldsymbol{M}, \boldsymbol{M}) = 0$$
$$l = l_0 + \lambda_1 l_1 + \lambda_2 l_2 = K(m_0 + \mu_1 m_1 + \mu_2 m_2)$$



J. Bermudez-Cameo, J.P. Barreto, G. Lopez-Nicolas, J. J. Guerrero, **Minimal solution for computing pairs of lines in non-central cameras**. ACCV 2014

3D lines reconstruction 7 pt algorithm

Hypothesis : The 3D line $\boldsymbol{L}$ intersects the 3D line $\boldsymbol{M}$

3 image points of $\boldsymbol{L}$ and and 4 image points of $\boldsymbol{M}$ give us 7 linear equations:

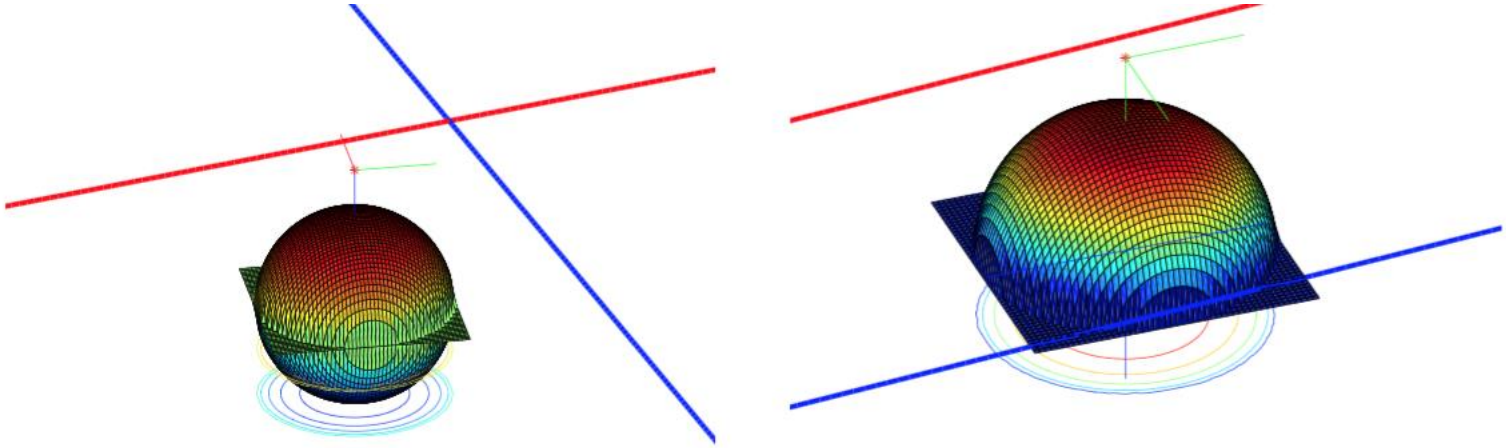
$$\boldsymbol{\Xi}_i^T W \boldsymbol{L} = 0, \boldsymbol{\Psi}_j^T W \boldsymbol{M} = 0 \text{ i} = 1..3, \text{j} = 1..4$$

$$\Rightarrow \begin{cases} \boldsymbol{L} = \boldsymbol{L_0} + \lambda_1 \boldsymbol{L_1} + \lambda_2 \boldsymbol{L_2} \\ \mathbf{M} = \boldsymbol{M_0} + \mu_1 \boldsymbol{M_1} \end{cases}$$

We need 3 other constraints

$$side(\boldsymbol{L}, \boldsymbol{L}) = 0$$
$$side(\boldsymbol{M}, \boldsymbol{M}) = 0$$
$$side(\boldsymbol{M}, \boldsymbol{L}) = 0$$

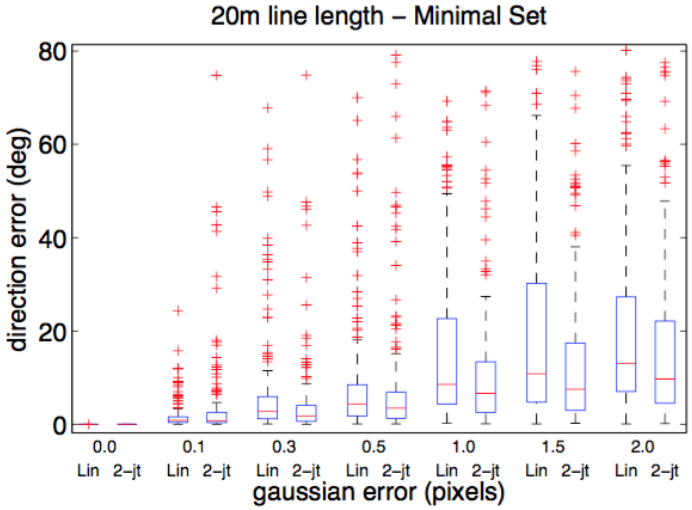
Experiments

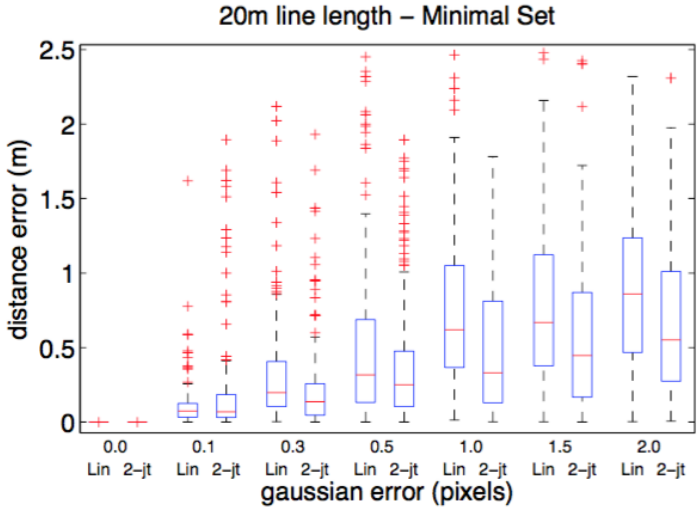


(a) (b)

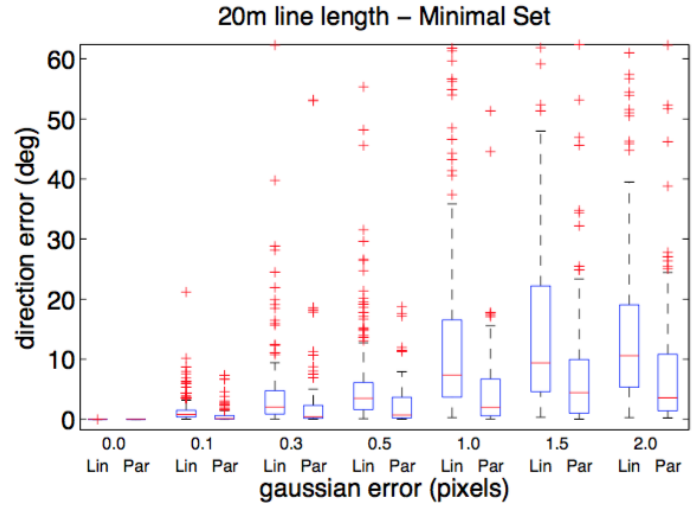
Synthetic catadioptric system composed of a spherical mirror and a perspective camera. (a) orthogonal lines (b) parallel lines.

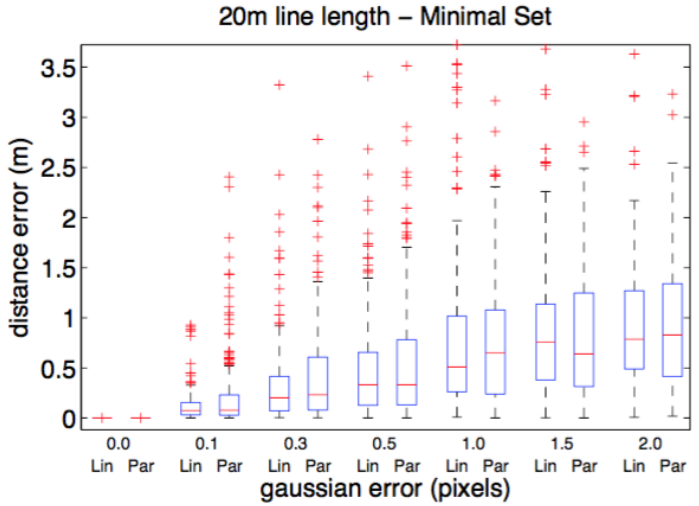
Experiments





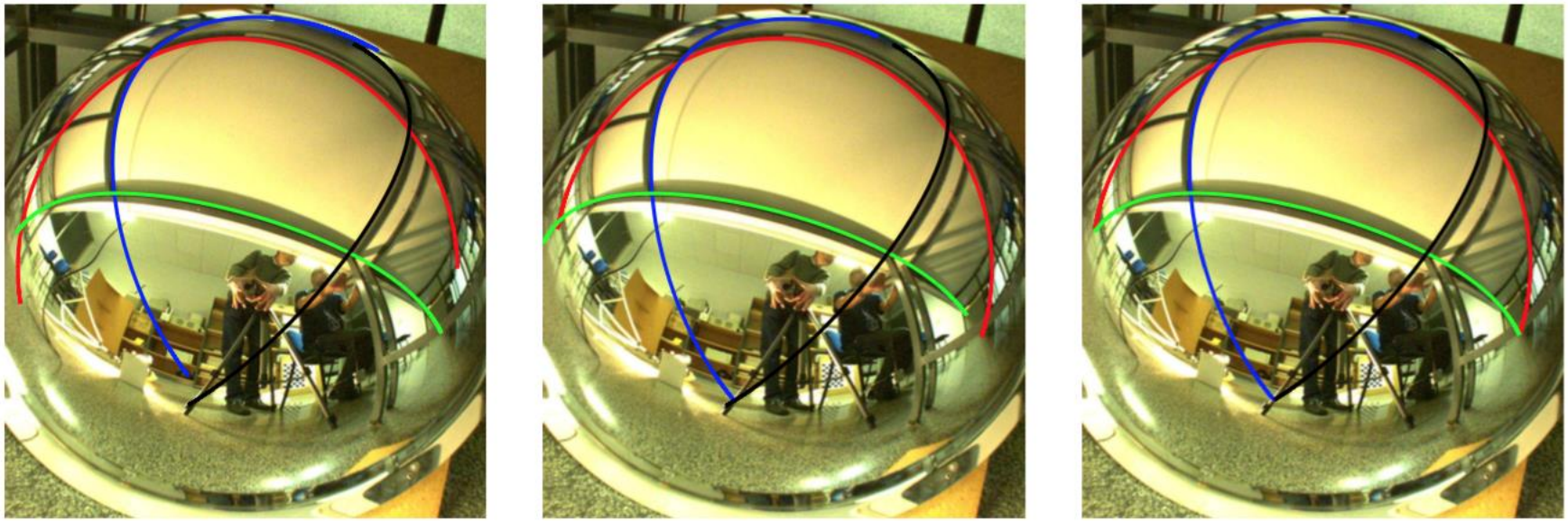
Orthogonal case



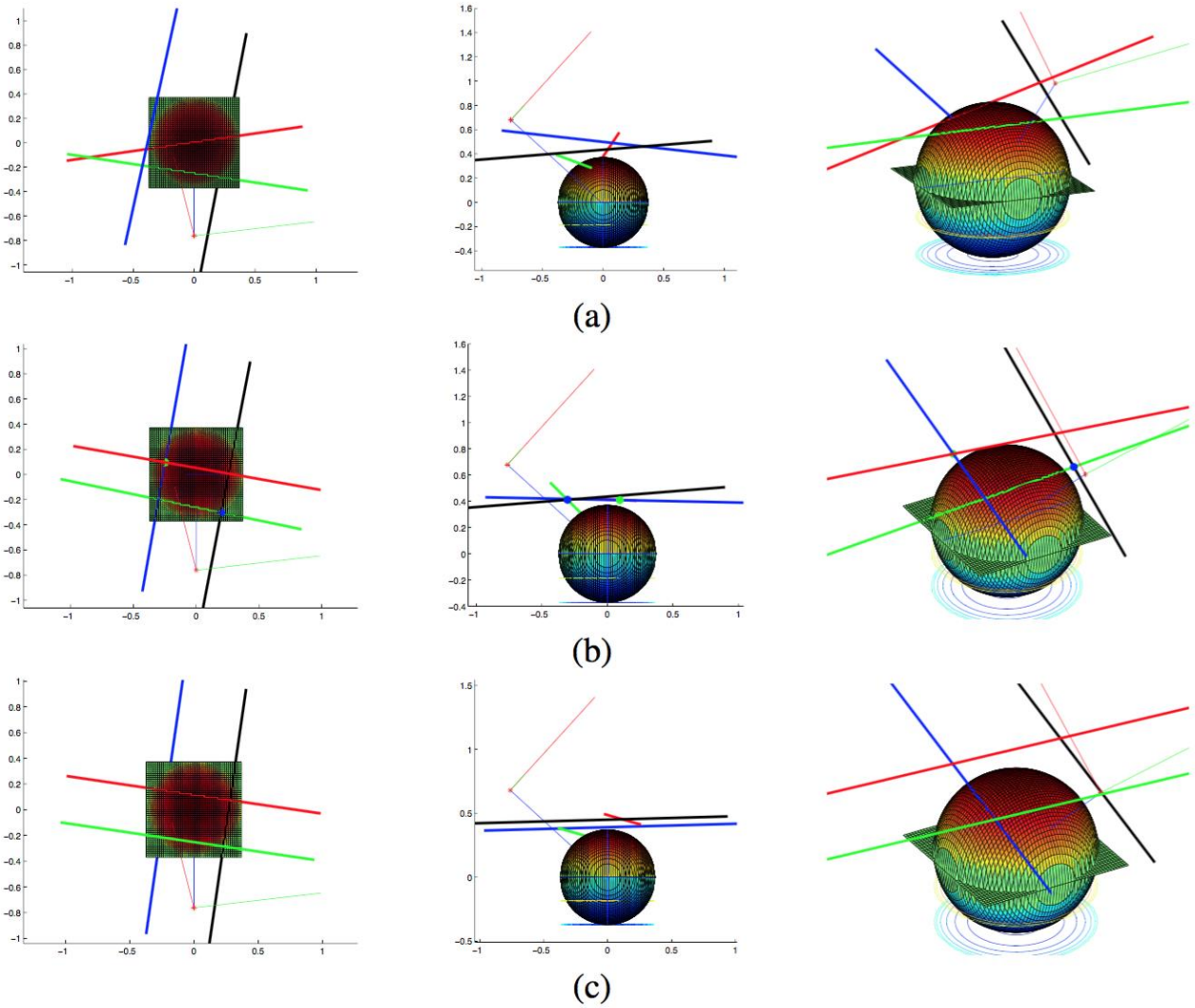


Parallel case

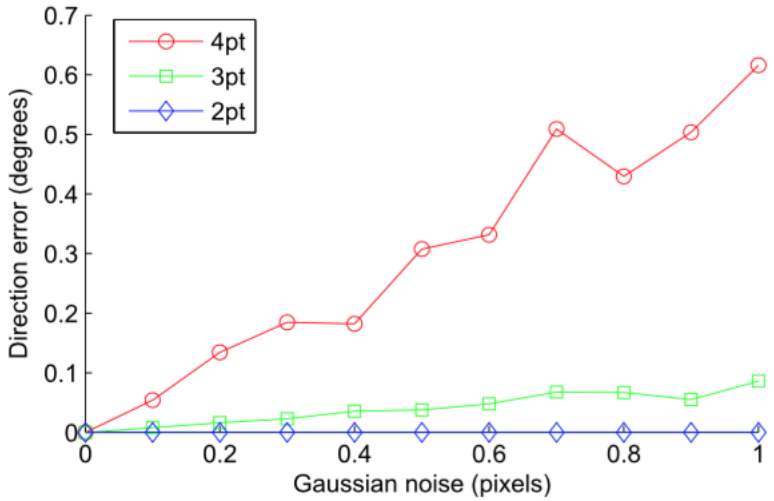
Experiments

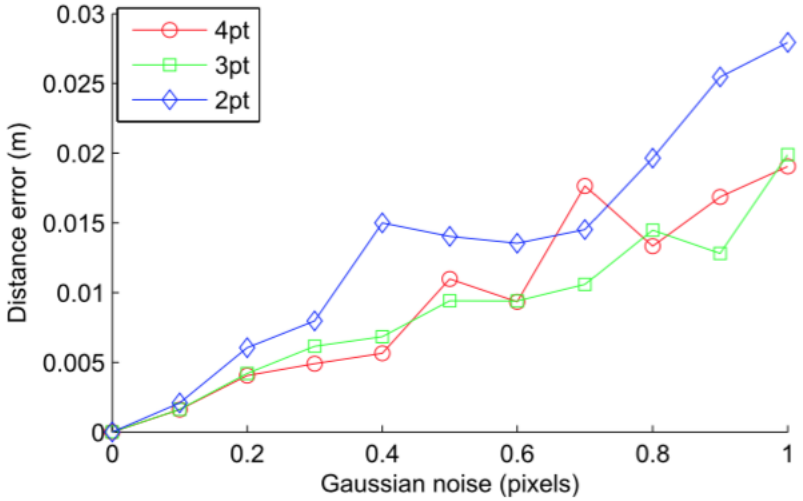


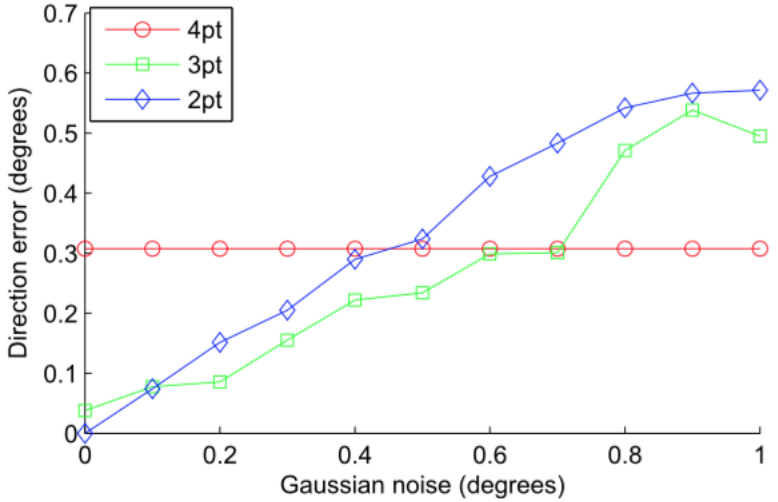
Real camera : Spherical mirror + perspective camera. Lines reprojection after 3D reconstruction. (a) 4 pt, (b) 3 pt+ortho, (c) 3pt+parallel.

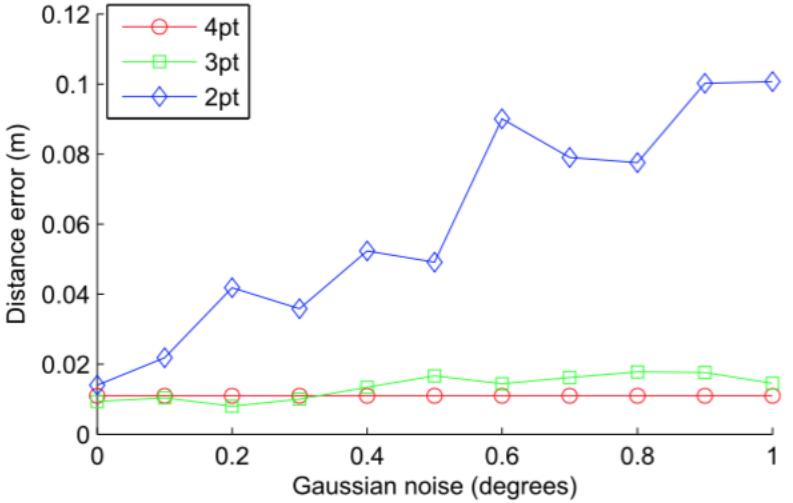


3D lines reconstruction (a) 4 pt, (b) 3 pt+ortho, (c) 3pt+parallel.



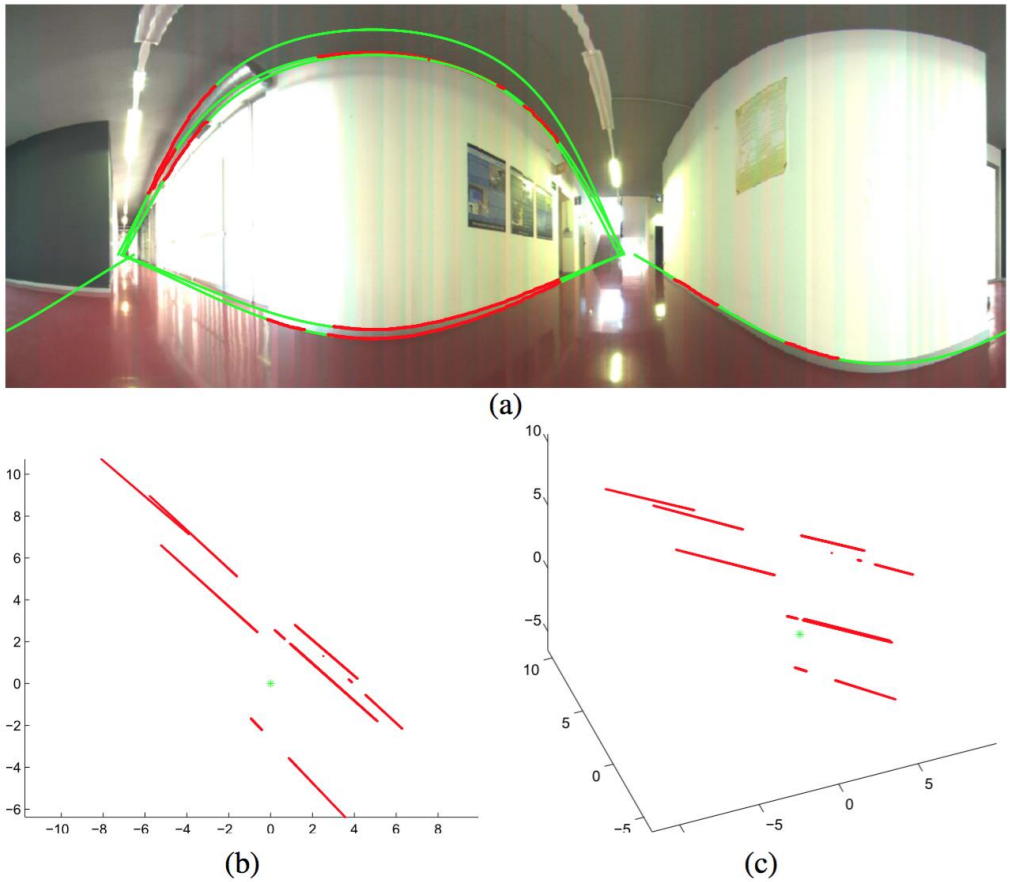






Synthetic data : 4pt vs 3pt vs 2pt. First row : with gaussian noise in pixels and without error in prior information. Second row : gaussian noise in pixels (σ=0.5) and Gaussian noise in prior information.

Experiments



First experiment with real non central camera

Conclusion

For 3D line reconstruction, we need[1]:

4 points

If the normal direction of 3D line is known ($\boldsymbol{L}$ belongs to a known plane):

3 points

If the direction of the 3D line is known:

2 points

For 2 3D lines:

8 points

If the 2 lines intersect:

7 points

If the 2 lines are orthogonal or parallel:

6 points,

[1]J. Bermudez-Cameo, C. Demonceaux, J.P. Barreto, G. Lopez-Nicolas, J. J. Guerrero, **Minimal solution for 3D line reconstruction in non-central systems.** Technical Report (in progress…)

3D À PARTIR D'HOMOGRAPHIE + VERTICALE CONNUE

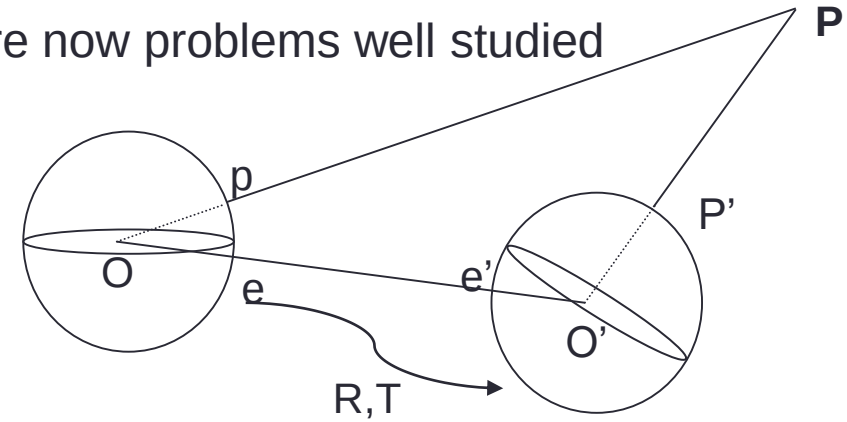
O. Saurer, P. Vasseur, C. Demonceaux, F. Fraundorfer. **A homography formulation to the 3pt plus a common direction relative pose problem**. In 12th Asian Conference on Computer Vision (ACCV 2014), November 2014, Singapour.

O. Saurer, P. Vasseur, R. Boutteau, C. Demonceaux, M. Pollefeys, F. Fraundorfer. **Homography Based Egomotion Estimation with a Common Direction**. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, To appear.

Introduction

Pose Estimation / Structure from Motion are now problems well studied

- Uncalibrated case[1] : 8 pts
- Calibrated case[2] : 5 pts
- Gravity known[3] : 3 pts
- Rotation known[4] : 2 pts



[1]R. Hartley, A. Zisserman. **Multiple View Geometry in Computer Vision**. Cambridge University Press, 2004
[2]D. Nister. **An efficient solution to the five-point relative pose problem**. PAMI 2004.
[3]F. Fraundorfer, P. Tanskanen, M. Pollefeys. **A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles**. ECCV 2010
[4]J.C. Bazin, C. Demonceaux, P. Vasseur, I.S. Kweon. **Motion Estimation by Decoupling Rotation and Translation in Catadioptric Vision**. CVIU 2010

Can we reduce the number of points needed for pose estimation when the 3D points belong to a plane?

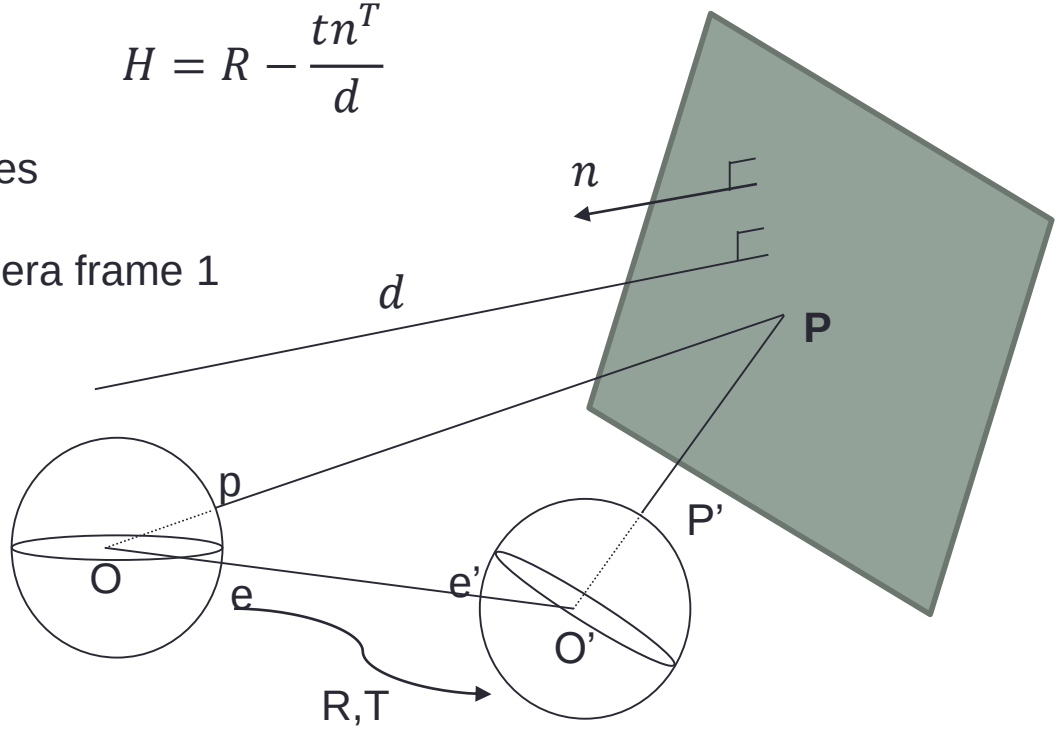
Homography estimation : 4 pt algo

Hypothesis : points belong on a 3D plane

Theorem : it exists an homography H such as :

$$H = R - \frac{tn^T}{d}$$

(R, t) rotation between 2 images
n normal vector of the plane
d distance to the plane in camera frame 1



Homography estimation : 4 pt algo

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \qquad p = \begin{pmatrix} x \\ y \\ w \end{pmatrix} \longleftrightarrow p' = \begin{pmatrix} x' \\ y' \\ w' \end{pmatrix}$$

$$p' \sim Hp \Rightarrow p' \times Hp = 0 \Rightarrow \begin{bmatrix} 0^T & -w'p^T & y'p^T \\ w'p^T & 0^T & -x'p^T \\ -y'p^T & x'p^T & 0^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0$$

This matrix has rank 2 (only 2 rows are linearly independent)

⇒ 4 pts are needed for estimating the 8 dof of H[1]

[1]R. Hartley, A. Zisserman. **Multiple View Geometry in Computer Vision**. Cambridge University Press, 2004

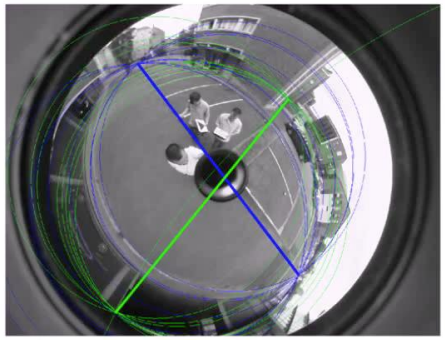
Homography estimation with known direction

Hypothesis : points belong on a 3D plane + gravity direction known

The vertical can be obtained by :

[1]IMU sensor

[2]C. Demonceaux, P. Vasseur, C. Pégard. **Robust Attitude Estimation with Catadioptric Vision.** IROS 2008



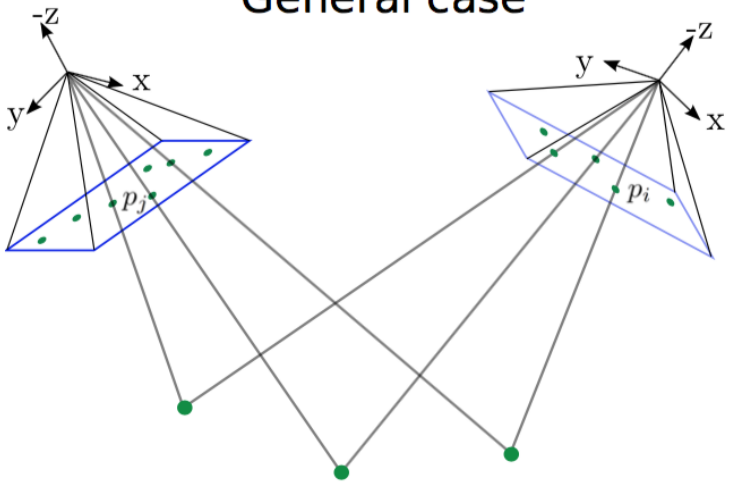
[3]J.C. Bazin, C. Demonceaux, P. Vasseur, I.S. Kweon. **Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment.** IJRR 2012

[4]J.C. Bazin, Y.D. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I.S. Kweon, M. Pollefeys. **Globally Optimal Line Clustering and Vanishing Point Estimation in Manhattan World.** CVPR 2012

Homography estimation with known direction

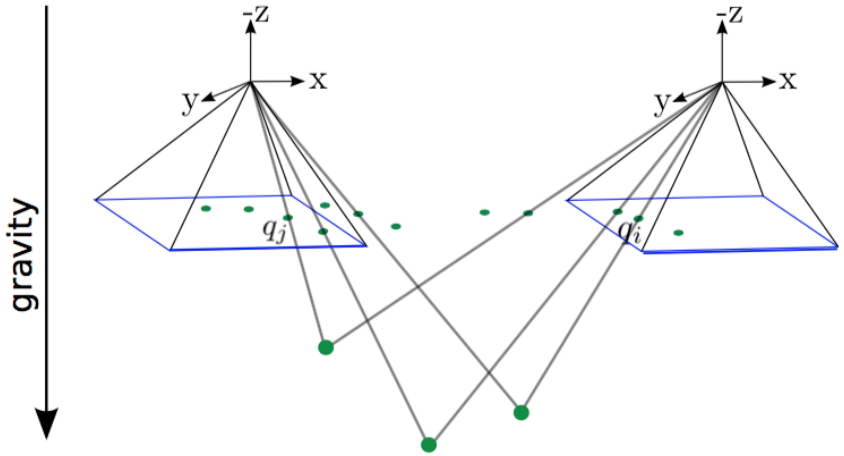
Hypothesis : points belong on a 3D plane + gravity direction known

General case



$$H = R - \frac{tn^T}{d}$$

Aligned with gravity direction



$$H = R_z - \frac{tn^T}{d}$$

Homography estimation : 3pt algorithm

Hypothesis : points belong on a 3D plane + gravity direction known

$$R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad t = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \qquad n = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}$$

$$H = R_z - \frac{tn^T}{d} = \begin{bmatrix} \cos\theta - t_x n_x & -\sin\theta - t_x n_y & -t_x n_z \\ \sin\theta - t_y n_x & \cos\theta - t_y n_y & -t_y n_z \\ -t_z n_x & -t_z n_y & 1 - t_z n_z \end{bmatrix}$$

H has 8 unknowns : $t, n, \cos\theta, \sin\theta$

Homography estimation : 3pt algorithm

$$p = \begin{pmatrix} x \\ y \\ w \end{pmatrix} \longleftrightarrow p' = \begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} \qquad p' \sim Hp \Rightarrow p' \times Hp = 0$$

1 point gives us 2 independent equations :

$$at_y - bt_z - w'x\sin\theta - w'y\cos\theta + y'w = 0$$
$$-at_x + ct_z + w'x\cos\theta - w'y\sin\theta - x'w = 0$$

where

$$a = w'xn_x + w'yn_y + w'wn_z$$
$$\mathrm{b} = y'xn_x + y'yn_y + y'wn_z$$
$$c = x'xn_x + x'yn_y + x'wn_z$$

Homography estimation : 3pt algorithm

3 points give us 6 independent equations

$$at_y - bt_z - w'x\sin\theta - w'y\cos\theta + y'w = 0$$
$$-at_x + ct_z + w'x\cos\theta - w'y\sin\theta - x'w = 0$$

A has a rank 6, $h = Y + \lambda_1 e_1 + \lambda_2 e_2$
+ 2 additional constraints

$$\cos^2\theta + \sin^2\theta = 1$$
$$\|n\|^2 = n_x^2 + n_y^2 + n_z^2 = 1$$

⇒ 3 points are sufficient for estimating H [1]

[1]O. Saurer, P. Vasseur, C. Demonceaux, F. Fraundorfer. **A homography formulation to the 3pt plus a common direction relative pose problem**. ACCV 2014.

Homography estimation : 2.5 pt algorithm

Hypothesis : points belong on a vertical plane + gravity direction known

$$R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad t = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \qquad n = \begin{pmatrix} n_x \\ n_y \\ 0 \end{pmatrix}$$

$$H = R_z - \frac{tn^T}{d} = \begin{bmatrix} \cos\theta - t_x n_x & -\sin\theta - t_x n_y & 0 \\ \sin\theta - t_y n_x & \cos\theta - t_y n_y & 0 \\ -t_z n_x & -t_z n_y & 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & 0 \\ h_3 & h_4 & 0 \\ h_5 & h_6 & 1 \end{bmatrix}$$

H has 6 unknowns

Homography estimation : 2.5 pt algorithm

2 points + 1 equation of a third point, give us 5 linear independent equations

$$Ah = B, \text{with } h = (h_1, h_2, \ldots, h_6)^T$$

A has a rank 5, $h = Y + \lambda e_1$

But, $\begin{cases} \cos^2\theta + \sin^2\theta = 1 \\ \|n\|^2 = n_x^2 + n_y^2 = 1 \end{cases} \Rightarrow$

$$h_1^2 h_6^2 - 2h_1 h_2 h_5 h_6 + h_2^2 h_5^2 + h_3^2 h_6^2 - 2h_3 h_4 h_5 h_6 + h_4^2 h_5^2 - h_5^2 - h_6^2 = 0$$

⇒ 2.5 points are sufficient for estimating H when we are watching an unknown vertical plane

Homography estimation : 2.5 pt algorithm

Remarks :

- the sixth equation of the third point can be used to see if these 3 points are inliers.
- the camera pose and scene structure can be easily retrieve :

$$t_z = \sqrt{h_5^2 + h_6^2}\,,\quad n_x = -\frac{h_5}{t_z},\quad n_y = -\frac{h_6}{t_z},$$

$$t_x = n_x(h_4 - h_1) - n_y(h_2 + h_3)$$

$$t_y = n_y(h_1 - h_4) - n_x(h_2 + h_3)$$

$$\cos\theta = h_1 + n_x t_x$$

Homography estimation : 2 pt algorithm

Hypothesis : points belong on the ground plane + gravity direction known

$$R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad t = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \qquad n = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$H = R_z - \frac{tn^T}{d} = \begin{bmatrix} \cos\theta & -\sin\theta & -t_x \\ \sin\theta & \cos\theta & -t_y \\ 0 & 0 & 1 - t_z \end{bmatrix} = \begin{bmatrix} h_1 & -h_2 & h_3 \\ h_2 & h_1 & h_4 \\ 0 & 0 & h_5 \end{bmatrix}$$

H has 5 unknowns

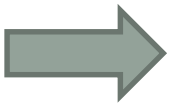
Homography estimation : 2pt algorithm

2 points give us 4 linear independent equations

$$Ah = 0, \text{ with } h = (h_1, h_2, \ldots, h_5)^T$$

A has a rank 4, $h = \lambda e_1$

but, $\cos^2\theta + \sin^2\theta = 1 \implies h_1^2 + h_2^2 = 1$

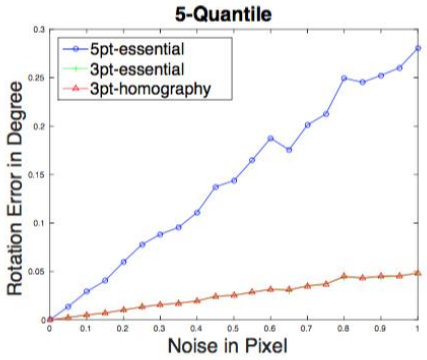


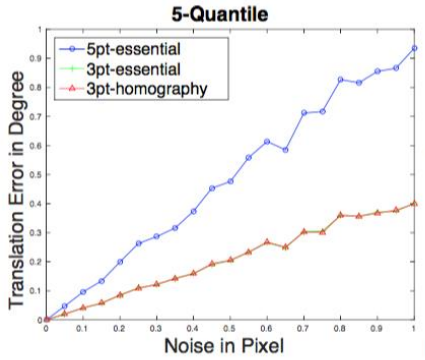
2 points are sufficient for estimating H when we are watching the ground plane

Experiments : Synthetic evaluation

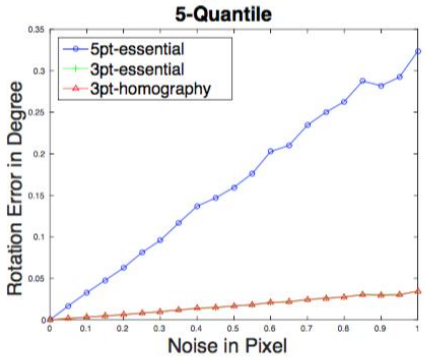
The synthetic scene is composed of 2 planes (vertical, ground). Each plane is randomly sampled with 200 points.

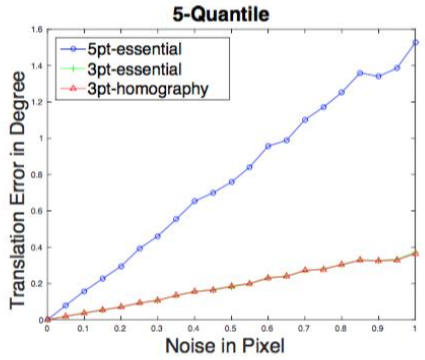
Sideways motion





Forward motion

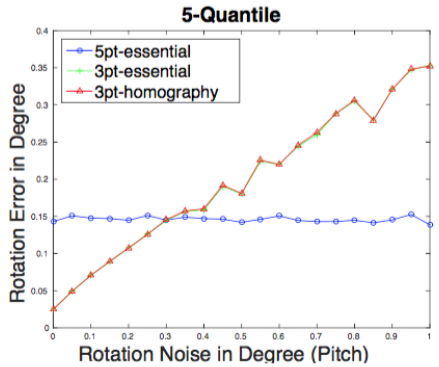


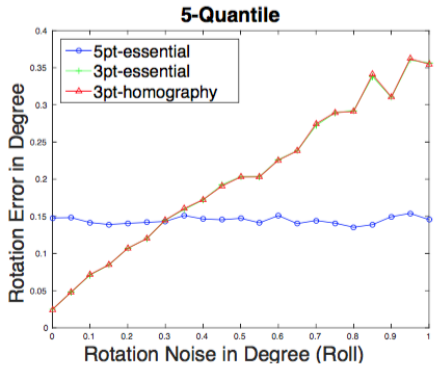


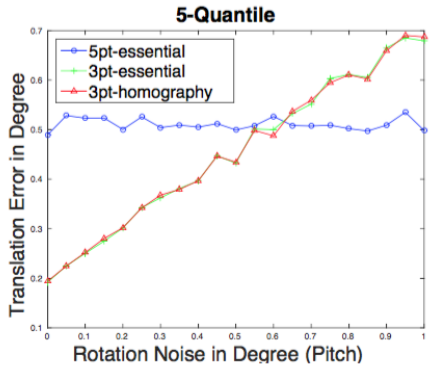
3pt approach vs classical methods with varying image noise

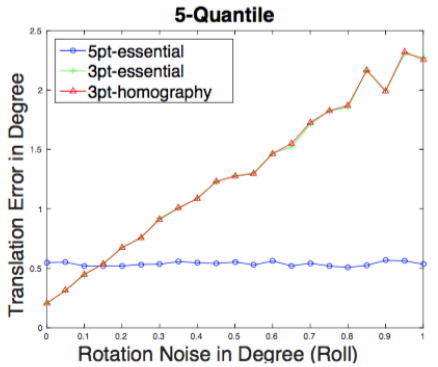
Experiments : Synthetic evaluation

Sideways motion

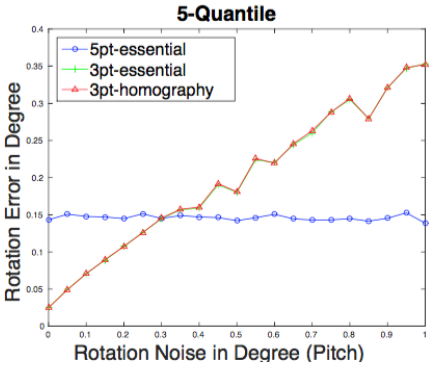


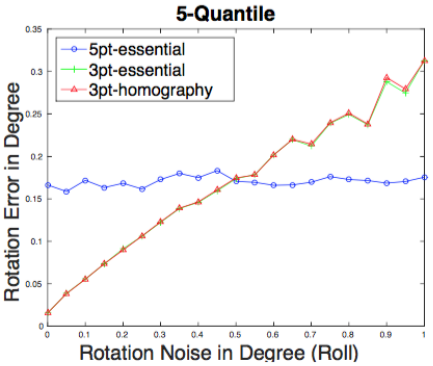


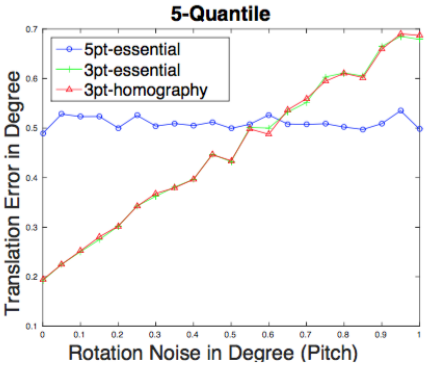


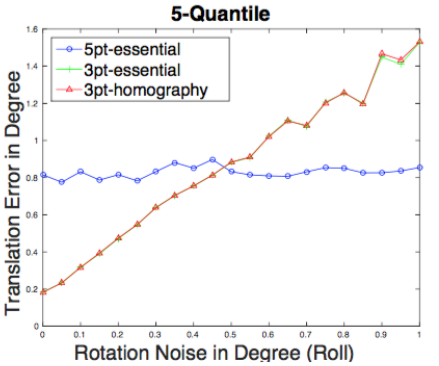


Forward motion







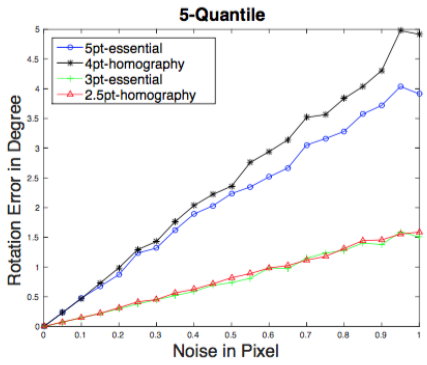


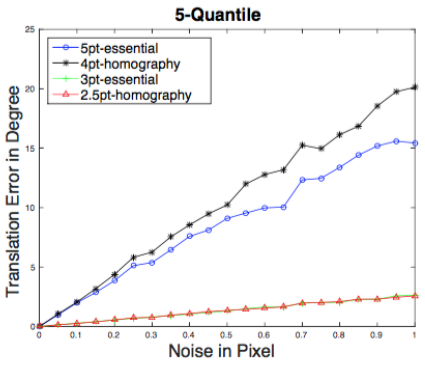
3 pt approach vs classical methods with IMU noise and constant image noise of 0.5 pixels.

Experiments : Synthetic evaluation

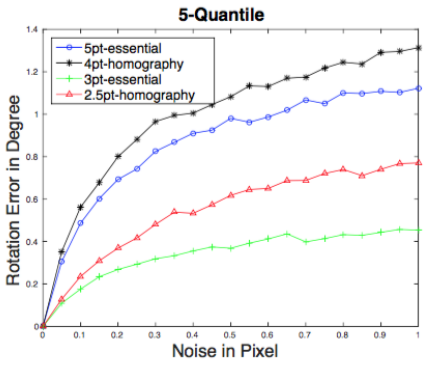
The synthetic scene is composed of 2 planes (vertical, ground). Each plane is randomly sampled with 200 points.

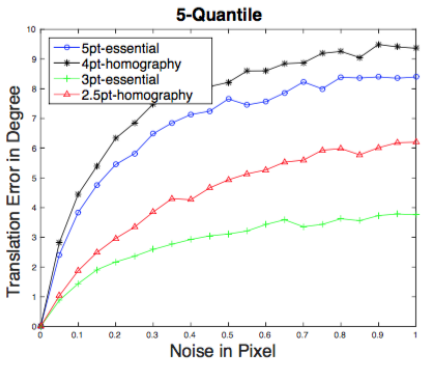
Sideways motion





Forward motion

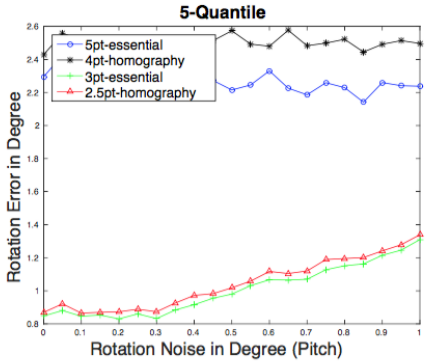


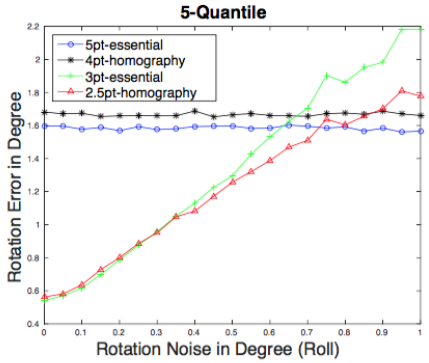


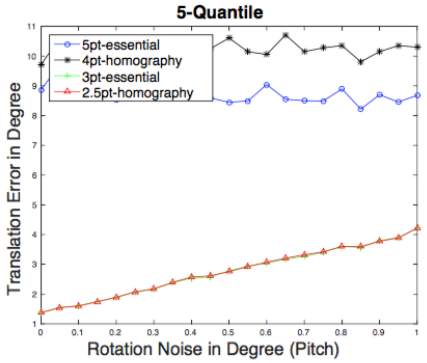
2.5 pt approach vs classical methods with varying image noise

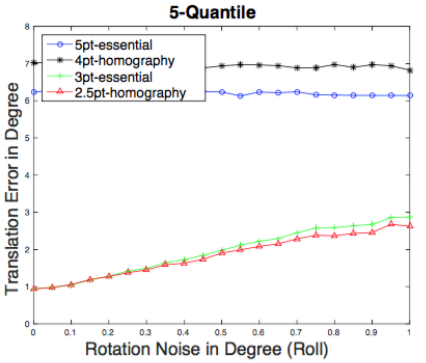
Experiments : Synthetic evaluation

Sideways motion

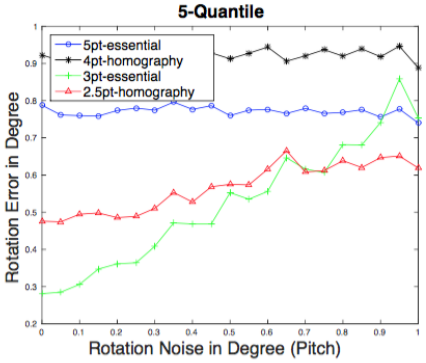


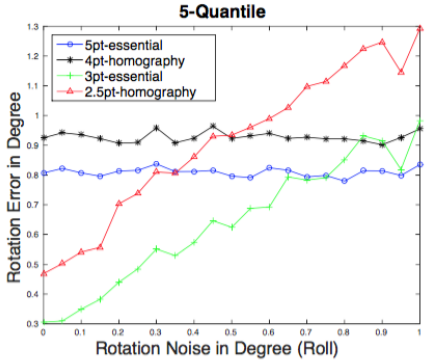


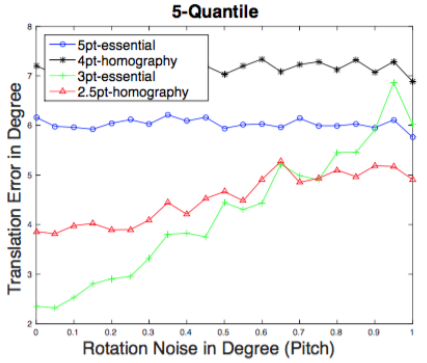


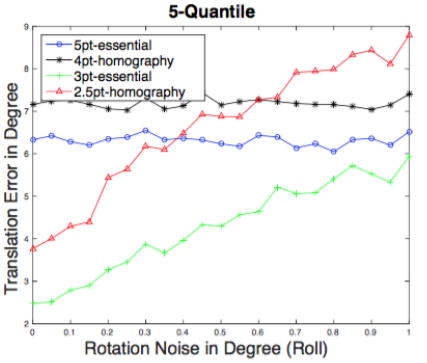


Forward motion







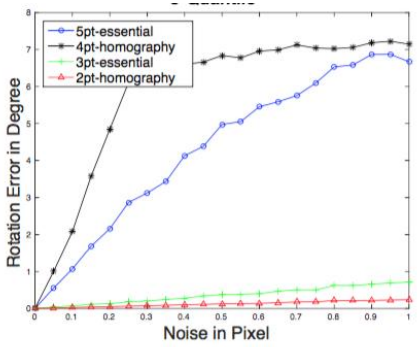


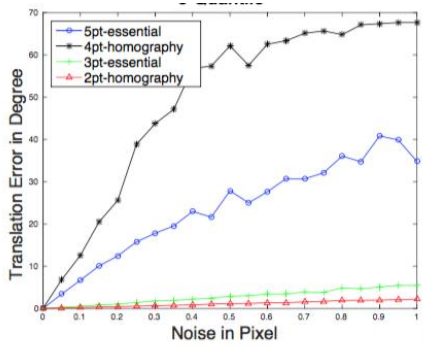
2.5 pt approach vs classical methods with IMU noise and constant image noise of 0.5 pixels.

Experiments : Synthetic evaluation

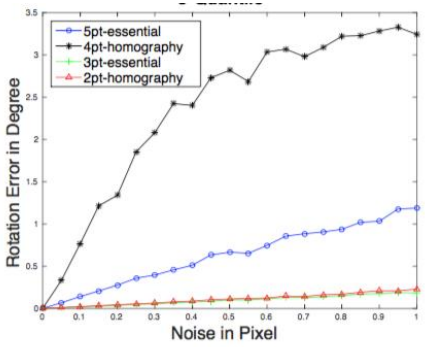
The synthetic scene is composed of 2 planes (vertical, ground). Each plane is randomly sampled with 200 points.

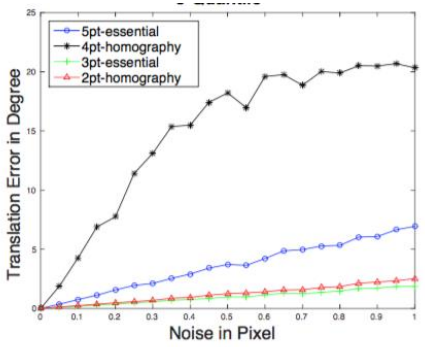
Sideways motion





Forward motion

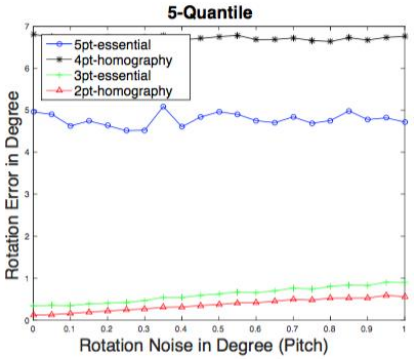


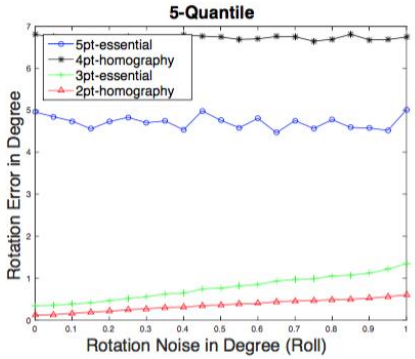


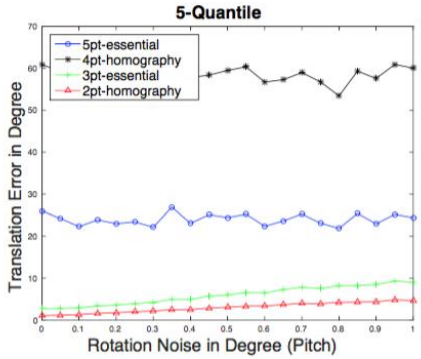
2pt approach vs classical methods with varying image noise

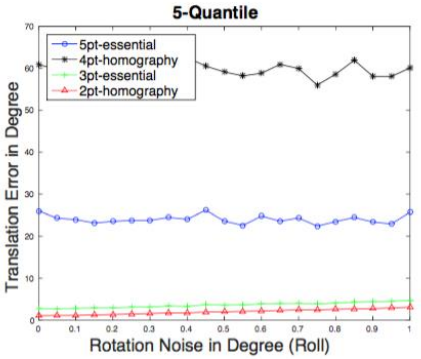
Experiments : Synthetic evaluation

Sideways motion

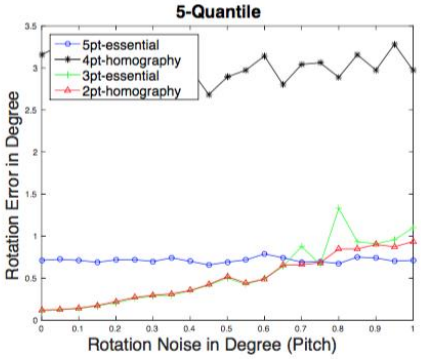


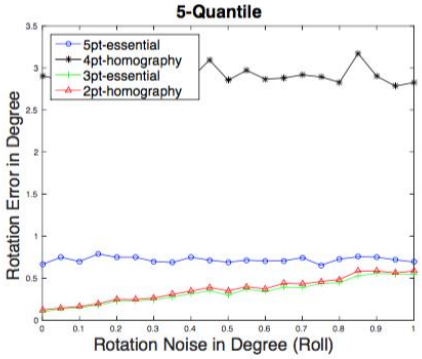


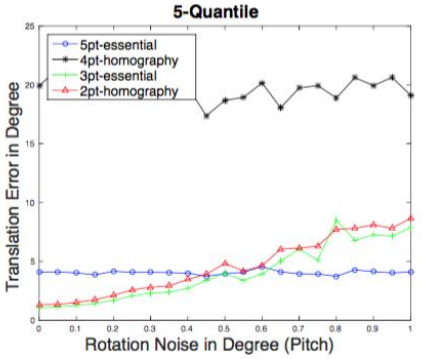


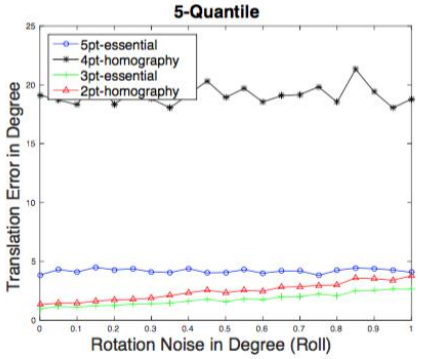


Forward motion









2pt approach vs classical methods with IMU noise and constant image noise of 0.5 pixels.

Experiments : Timings

Method	Hypothesis Estimation(ms)	RANSAC 1 Iteration(ms)
2pt	0.09	8.31
2.5pt	0.22	33.45
3pt-homography	27.28	55.17
3pt-essential	0.49	25.02
4pt-homography	0.18	8.65
5pt-essential	0.42	64.33

Run-time comparison of different pose estimation algorithms
(Intel i7 3.4GHz with Matlab)

Experiments : Real data

(a)

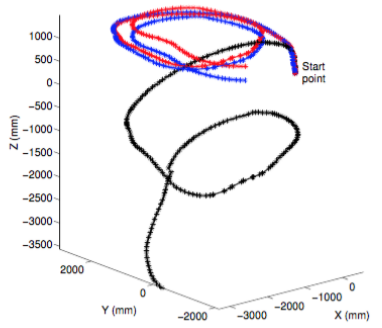
(b)

(c)

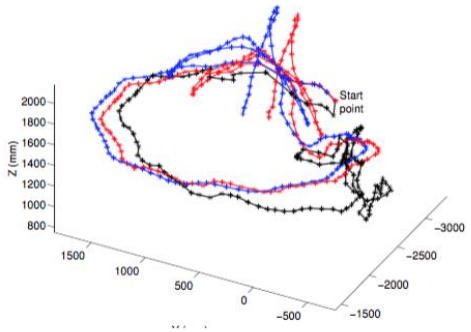
(a) Vicon arena used to record the ground-truth dataset, (b) teleoperated Segway mobile robot capturing data, (c) sample image captured by the robot.

Experiments : Real data

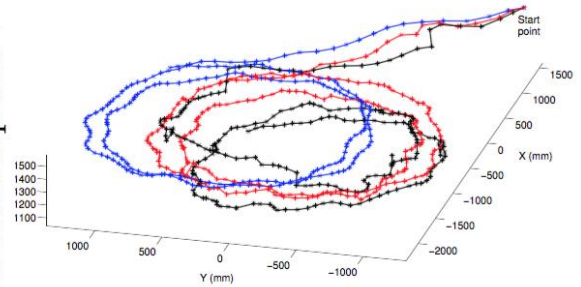
Ground Sequence I



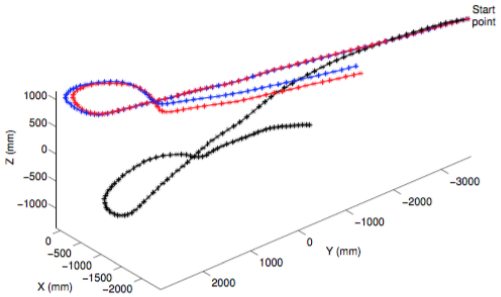
Ground Sequence III



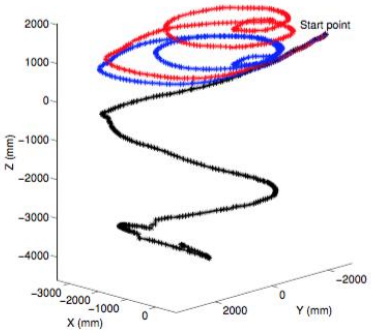
Ground Sequence V



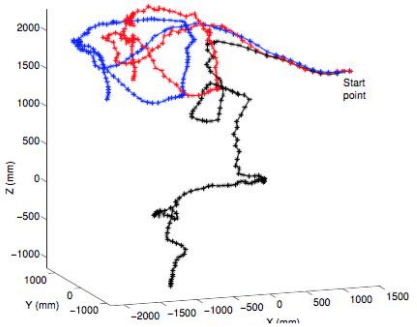
Ground Sequence II



Ground Sequence IV

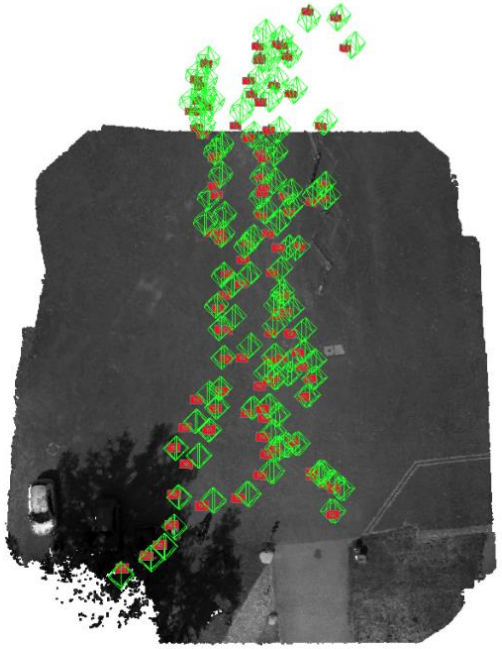


Ground Sequence VI

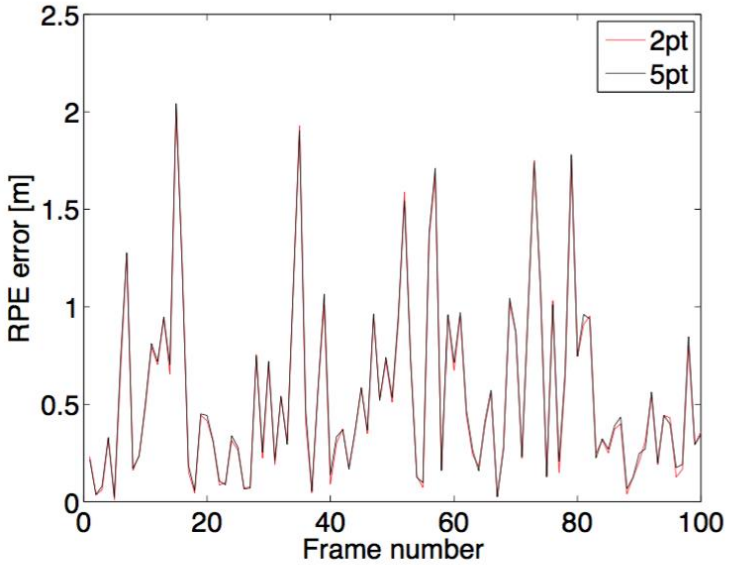


Evaluation of 2pt. Red : 2pt, Black : 5pt, Blue : Grounth-truth

Experiments : Real data



(a)



(b)

Results of an incremental SFM pipeline using the 2pt algorithm. (a) Resulting 3D point cloud, camera positions (red) and GPS positions (green). (b) RPE error plot when using 5pt or 2pt within the SFM pipeline. The initial solution of the 5pt and 2pt are similar enough for bundle adjustment to converge to almost the same final solution with a gained speed-up for 2 pt.

Conclusion and perspective

We can reduce the number of points we need to estimate an homography if we suppose[1]:

- vertical known : 3 points
- vertical known + vertical plane : 2.5 points
- vertical known + ground plane : 2 points

Advantages :

- real time application (less time consuming because we reduce the number of iteration)
- robustness!

[1]O. Saurer, P. Vasseur, R. Boutteau, C. Demonceaux, M. Pollefeys, F. Fraundorfer. **Homography Based Egomotion Estimation with a Common Direction**. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, To appear.

Conclusion and perspective

Can we go further?
Vertical known + ground plane : 5 unknowns
1.5 point give us 3 equations

$$Ah = 0, \text{with } h = (h_1, h_2, \ldots, h_5)^T$$

A has a rank 3, $h = \alpha_1 e_1 + \alpha_2 e_2$
The 3 singular values $\lambda_1 < \lambda_2 < \lambda_3$ of H verify[1] :

$$\|t_1\| = \|R^T t\| = \lambda_1 - \lambda_3$$
$$n^T t_1 = \lambda_1 \lambda_3 - 1$$
$$\lambda_2 = 1$$

[1]E. Malis, M. Vargas. Deeper understanding of the homography decomposition for vision-based control. Research Report INRIA 2007

Conclusion and perspective

$$h = \alpha_1 e_1 + \alpha_2 e_2$$
$$h_3^2 + h_4^2 + (1 - h_5)^2 = \lambda_1 - \lambda_3$$
$$h_5 = \lambda_1 \lambda_3$$
$$\lambda_2 = 1$$

with

$$\lambda_1 = \frac{\sqrt{h_1^2+h_2^2+h_3^2+h_4^2+h_5^2+(h_1^4+2h_1^2h_2^2+2h_1^2h_3^2+2h_1^2h_4^2-2h_1^2h_5^2+h_2^4+2h_2^2h_3^2+2h_2^2h_4^2-2h_2^2h_5^2+h_3^4+2h_3^2h_4^2+2h_3^2h_5^2+h_4^4+2h_4^2h_5^2+h_5^4)}}{2}$$

$$\lambda_2 = h_1^2 + h_2^2$$

$$\lambda_3 = \frac{\sqrt{h_1^2+h_2^2+h_3^2+h_4^2+h_5^2-(h_1^4+2h_1^2h_2^2+2h_1^2h_3^2+2h_1^2h_4^2-2h_1^2h_5^2+h_2^4+2h_2^2h_3^2+2h_2^2h_4^2-2h_2^2h_5^2+h_3^4+2h_3^2h_4^2+2h_3^2h_5^2+h_4^4+2h_4^2h_5^2+h_5^4)}}{2}$$

(9)

3D MOTION ANALYSIS USING PRIOR KNOWLEDGE

C. Jiang, D. P. Paudel, Y. Fougerolle, D. Fofi, C. Demonceaux. **Static-map and Dynamic Object Reconstruction in Outdoor Environments using 3D Motion Segmentation**. In *IEEE Robotics and Automation Letters*, Vol. 1, Issue 1, January 2016, pp 324-331.

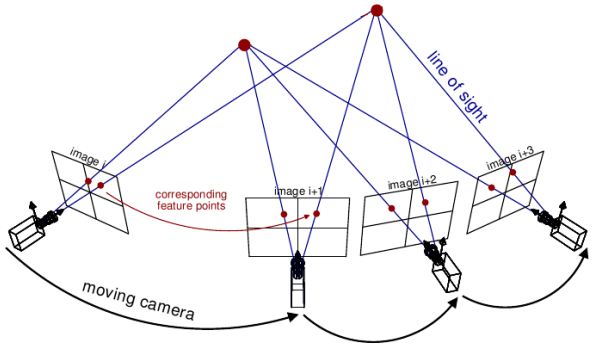
D.P. Paudel, A. Habed, C. Demonceaux, P. Vasseur. **Robust and Optimal SoS-based Point-to-Plane Registration of Image Sets and Structured Scenes.** In ICCV 2015 (oral presentation)

D.P. Paudel, A. Habed, C. Demonceaux, P. Vasseur. **LMI-based 2D-3D Registration: from Uncalibrated Images to Euclidean Scene.** In CVPR 2015

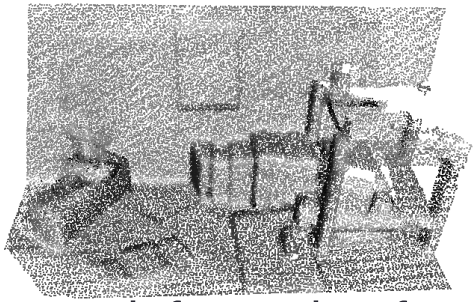
D.P. Paudel, C. Demonceaux, A. Habed, P. Vasseur, I.S. Kweon. **2D-3D Camera fusion for Visual Odometry in outdoor environments.** In IROS 2014

Problem statement

- *Nowadays, we can obtain the 3D structure of a scene really easily.*
  - *Structure from motion*



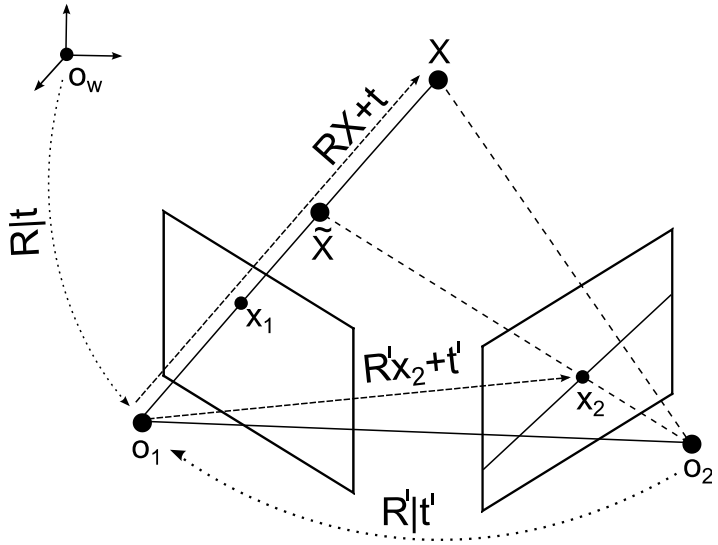
  - *RGB-D Camera*



  - Can we use the scene information for motion refinement?
  - Can we take the advantage from 2D and 3D for a better SLAM?

Problem formulation

- 3D – 2D/2D relationships
  - Knowns :
    - $X_k, k = 1 \ldots p$, 3D points
    - 2 calibrated cameras C_1, C_2
    - $x_j^1 \leftrightarrow x_j^2, j = 1 \ \ldots n$, 2D points
  - Unknowns :
  - R_i, t_i cameras position in O_w
  - R', t'camera of C_2 in C_1
  - 2D-3D matching ϕ



Problem formulation

- 3D – 2D/2D relationships
  - 1st Camera Localization

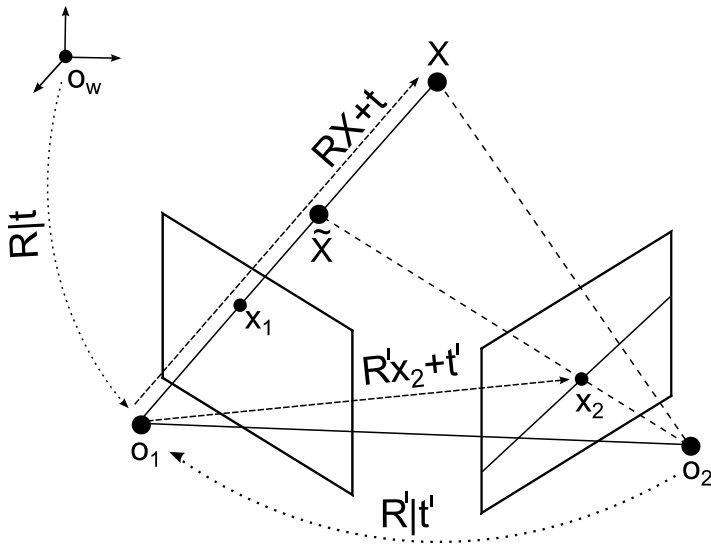
$$x_j^1 = P_1(R_1, t_1, X_{\phi_1(j)})$$

  - 2nd Camera Localization

$$x_j^2 = P_2(R_1 R', R' t_1 + t', X_{\phi_1(j)})$$

  - Epipolar Constraint

$$(x_j^2)^T E_1(R', t') x_j^1 = 0.$$

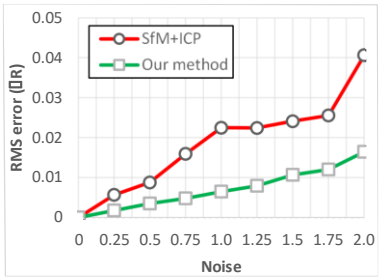


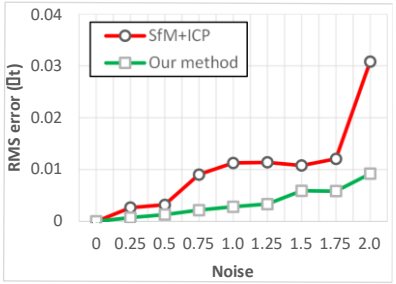
Optimization framework

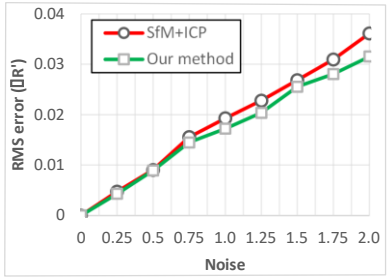
$$\min_{q_1,t_1,q',t',\phi} \sum_{j=1}^{n} ||x_j^2 - P(R_1R', R't_1 + t', X_{\phi_1(j)})||^2$$
$$\text{subject to } ||x_j^1 - P(R_1, t_1, X_{\phi_1(j)})||^2 = 0$$
$$(x_j^2)^T E_1(R', t')x_j^1 = 0$$
$$||q_1||^2 = 1, ||q'||^2 = 1$$

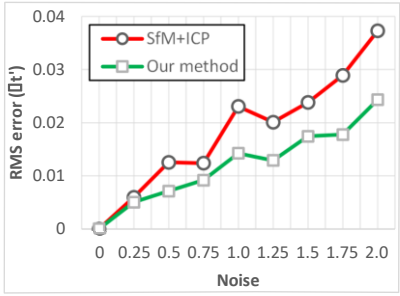
Synthetic data

- We generated 800 random 3D points on a $[-10,10]^3$ cube
- Cameras were placed about 20 units aways from the origin with random rotations (looking towards the center).
- 256 × 256 images
- 2D data (400 points randomly chosen from 800 3D points) obtained by adding with noise to the pixel coordinates
- 100 tests per noise.



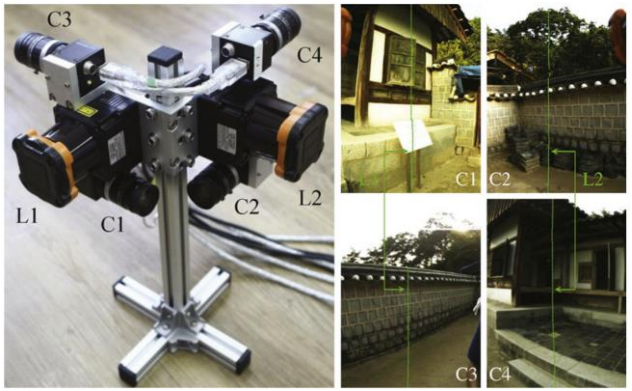


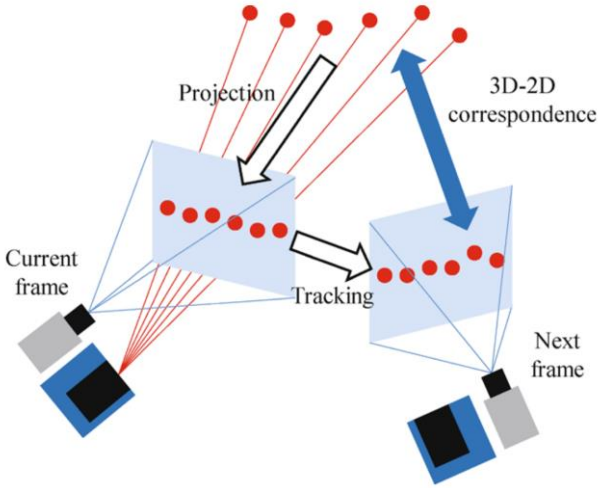




KAIST dataset

- System setup
  - Two laser line scanners (L1 and L2)
  - Four perspective camera (C1 – C4)
  - Laser/camera are calibrated and synchronized
- Motion estimation (Bok et al. IJCV'11)
  - Optical flow based correspondences
  - Relative motion from SFM (2D)
  - Scale recovery from known 2D-3D correspondences.





KAIST dataset

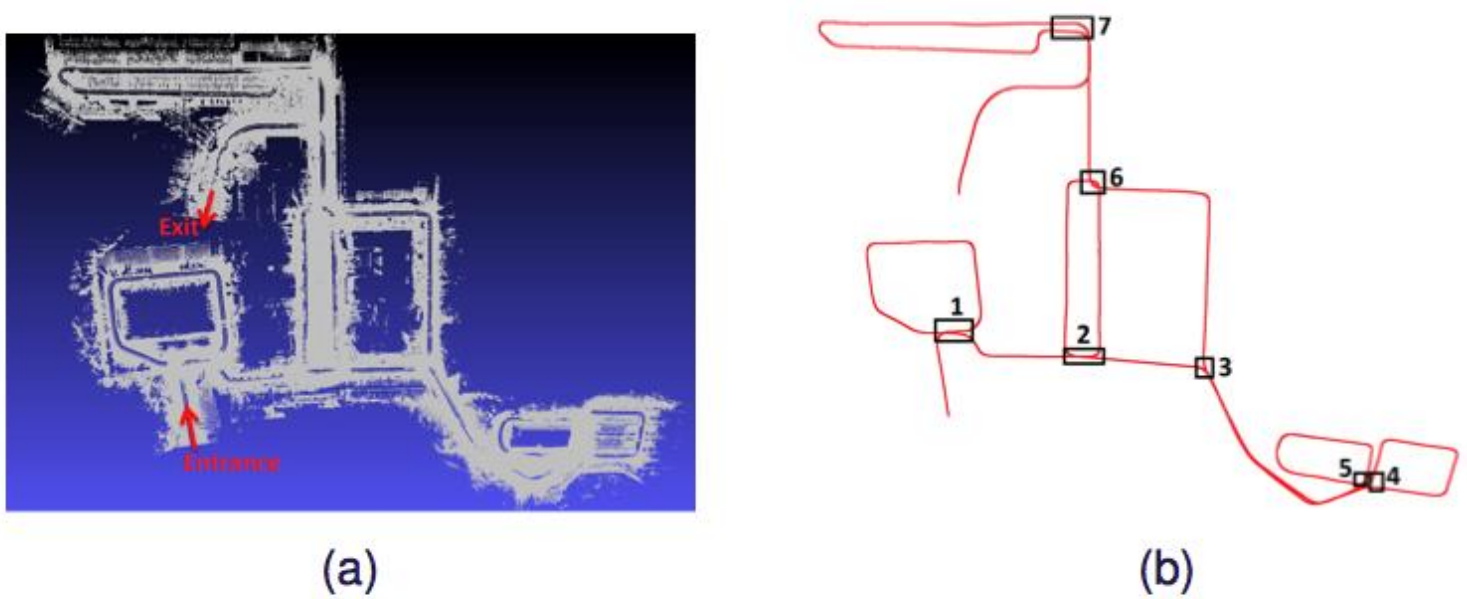


Figure: (a) A large map (3.5 km) reconstructed. (b) Closed loops made during the travel. Boxes shown are the loop closing locations of seven different loops.

KAIST dataset

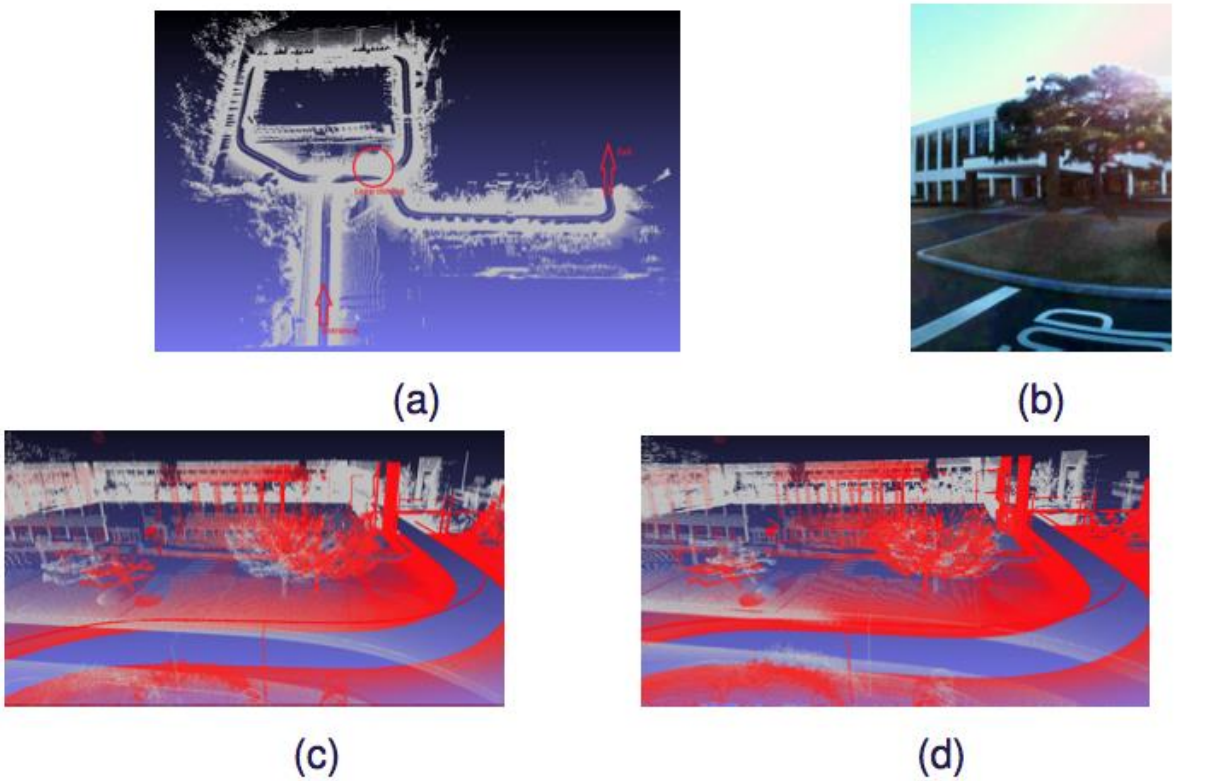


Figure: (a) Map built around Loop 1. (b) Image taken at loop closing point. (c) Map built before (red) and after (white) the visit around the loop before refinement. (d) Map refined using our method.

KAIST dataset

Loop	Size	Before refinement	After refinement
1	351.76	4.063	1.548
2	386.38	4.538	1.469
3	224.37	4.765	4.398
4	242.87	1.696	1.077
5	931.14	3.884	2.858
6	1496.4	7.182	6.381
7	546.05	5.502	2.115

Table: Loop size and loop closing errors in meters.

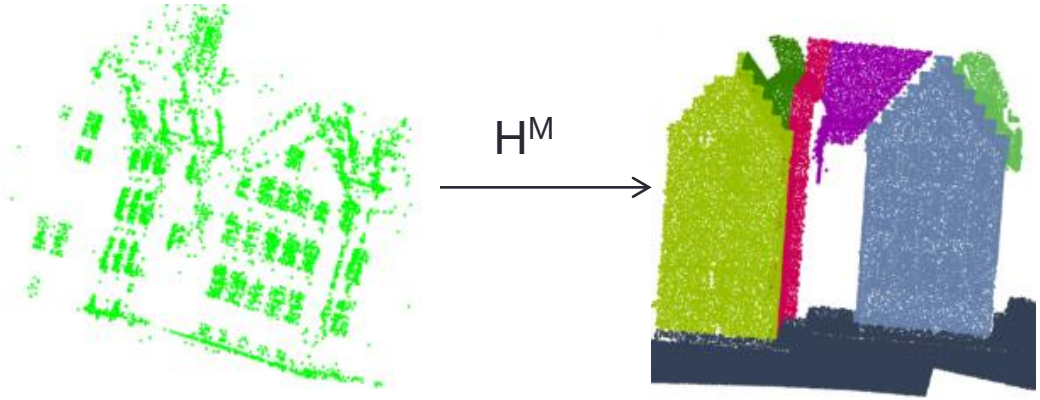
D.P. Paudel, A. Habed, C. Demonceaux, P. Vasseur. **LMI-based 2D-3D Registration: from Uncalibrated Images to Euclidean Scene.** In CVPR 2015

Part I Conclusion 1

- If we know 3D structure of the scene or if we use RGB-D cameras, we can improve classical technics for pose estimation
- Can we go further if we have more information about the 3D structure?
  - Ex : The scene is composed of different planes

3D scene is structured by planar patches : a global optimization framework

- Problem
  - Calibrated camera
  - The scene is composed of at least 4 distinct planes
  - The scene has been scanned by a 3D sensor



3D scene is structured by planar patches : a global optimization framework

- Set $\mathcal{P}$ of distinct planes in the scene
- $Y \in \mathcal{Y}$ 3D points
- $\mathcal{A} \subset \mathcal{Y} \times \mathcal{P}$ set of putative point to plane assignments

$$a = (Y, \Pi) \in \mathcal{A}$$

$$\mathbf{x} = (\mathbf{q}^\mathsf{T}, \mathbf{t}^\mathsf{T})^\mathsf{T} \qquad f_a(\mathbf{x}) := \pi^\mathsf{T}(\mathbf{QY} + \mathbf{t}) - d.$$

- Consensus set maximization

$$\max_{\mathcal{Z},\, \mathbf{x}} \quad \sum_{a \in \mathcal{A}} z_a$$

$$\text{subject to} \quad z_a f_a(\mathbf{x}) = 0, \quad \text{for all } a \in \mathcal{A},$$

$$z_a \in \{0, 1\}, \quad \text{for all } a \in \mathcal{A}.$$

- This problem is solved in a global optimization formalism by Branch-and-Bound paradigm using Sum-of-Squares (SoS)

Experiments

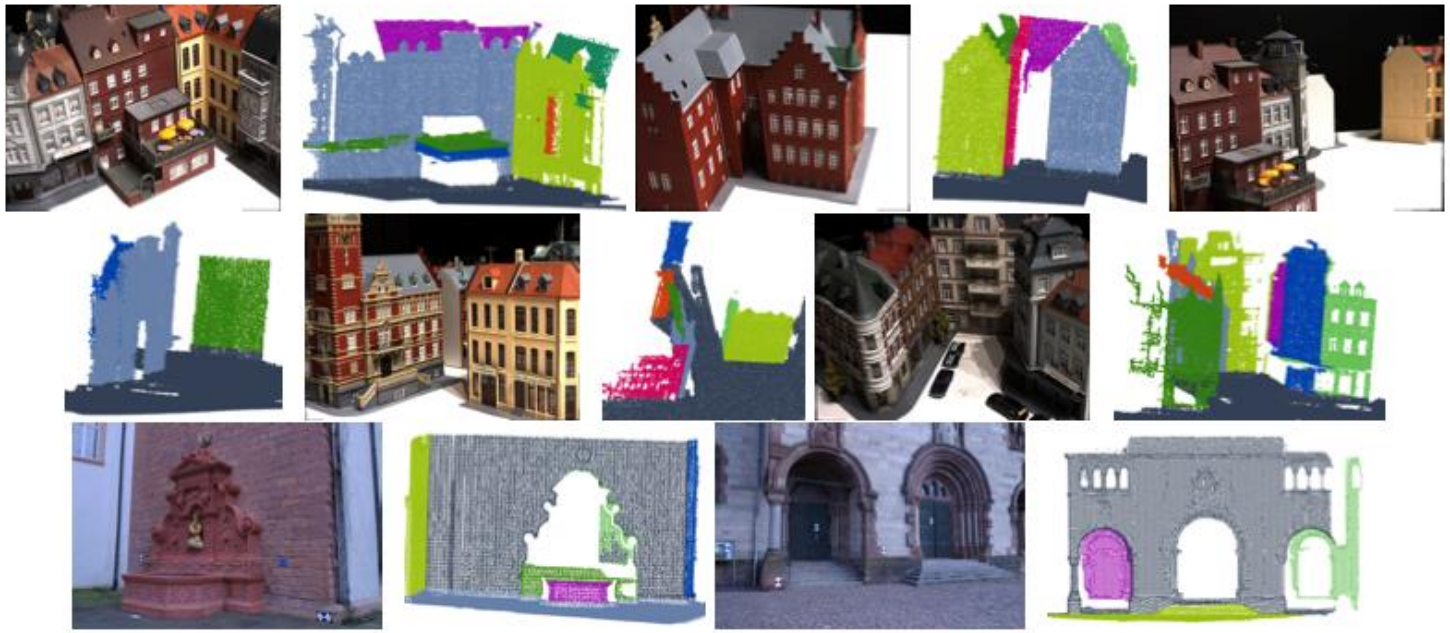


FIGURE 5.2 – Sample image and corresponding segmented scene next to each other shown in different colors for each plane (In order : Scene23, Scene24, Scene27, Scene29, Scene73, Fountain, and Herz-Jesu).

Experiments

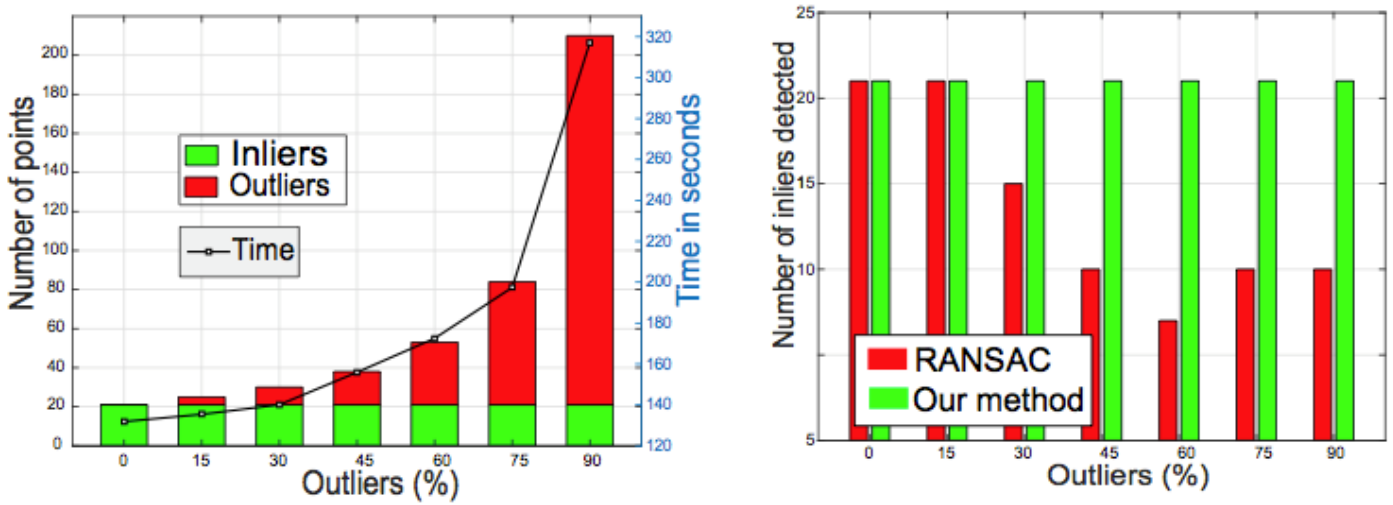


FIGURE 5.3 – Experiments on Scene73 with correspondences and no camera bounds. Left : no. of processed points and time taken for various levels of outliers. Right : no. of detected inliers using RANSAC and our method.

Results

Scene	Method	Time (sec)	ΔR	$\Delta T(\%)$	3D error
Fountain	RISAG	805.680	8.6825	14.08	0.3275
	Go-ICP	529.415	0.7225	1.63	0.0348
	Our method	55.730	2.8639	3.18	0.0570
Herz-Jesu	RISAG	160.064	17.6378	5.70	0.1830
	Go-ICP	31.254	3.2618	16.9	0.0725
	Our method	137.766	7.1958	4.02	0.0464

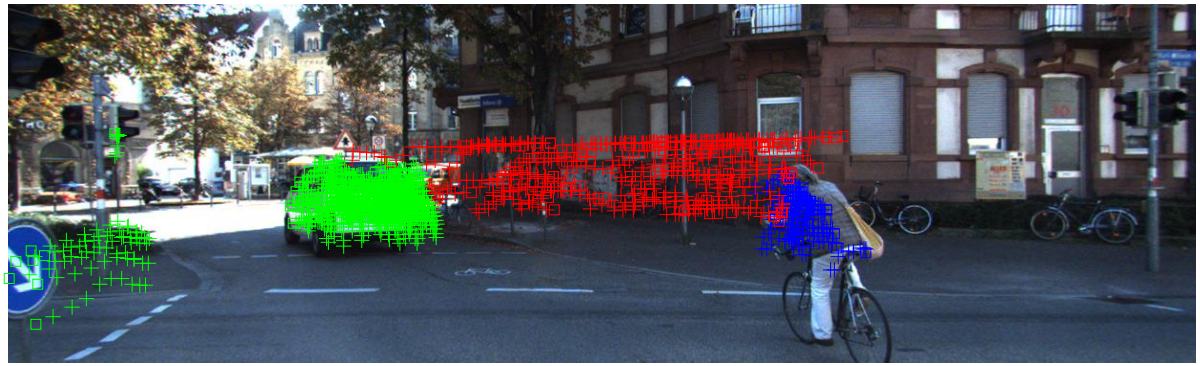
Comparison between our method[1], RISAG[2] and GO-ICP[2]

[1]D.P. Paudel, A. Habed, C. Demonceaux, P. Vasseur. **Robust and Optimal SoS-based Point-to-Plane Registration of Image Sets and Structured Scenes.** In ICCV 2015
[2]M. Corsini, M. Dellepiane, F. Ganovelli, R. Gherardi, A. Fusiello, R. Scopigno. Fully automatic registration of image sets on approximate geometry. IJCV 2013
[3]J. Yang, H. Li, and Y.Jia. Go-icp : Solving 3d registration efficiently and globally optimally. ICCV 2013

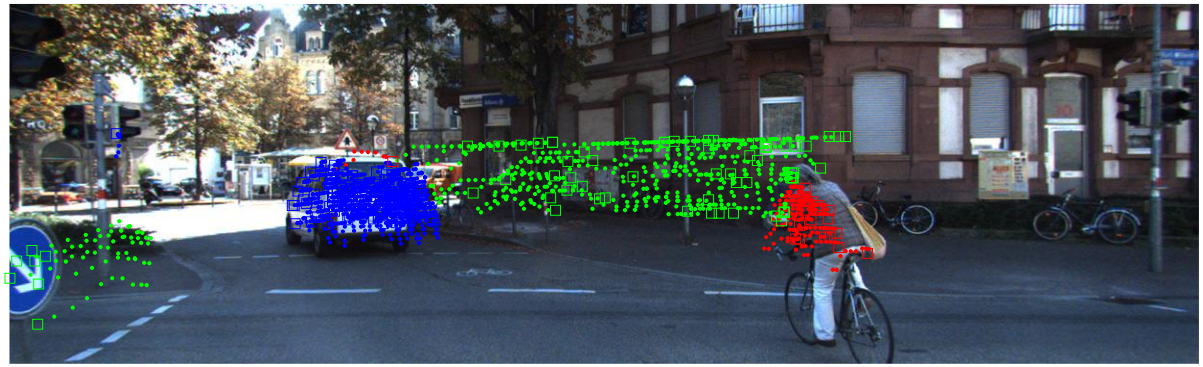
Part 2 Conclusion 2

- If we know 3D structure of the scene or if we use RGB-D cameras, we can improve classical technics for pose estimation.
- If we suppose that the scene is composed of planes, the problem becomes a global optimization problem
- Can we use these technics for 3D motion analysis using 3D/2D information ?

3D motion analysis with prior knowledge

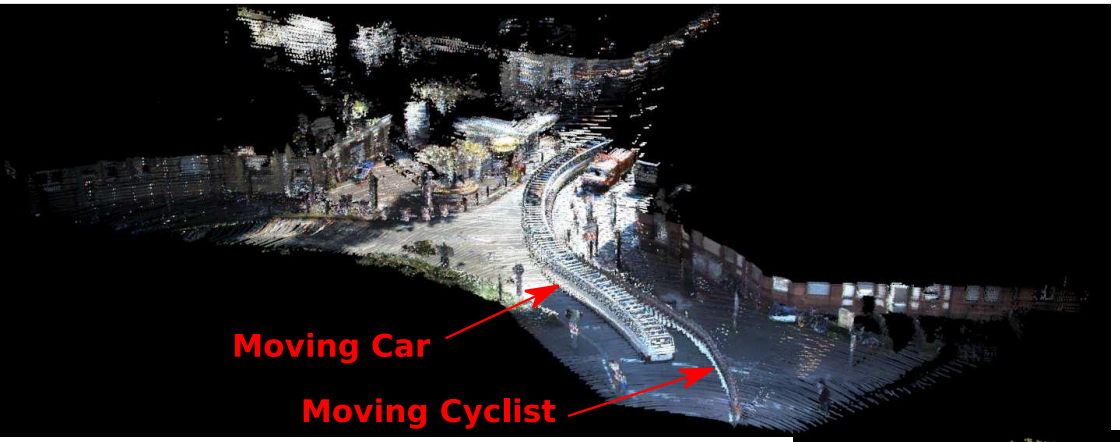


MS using 2D feature trajectories.

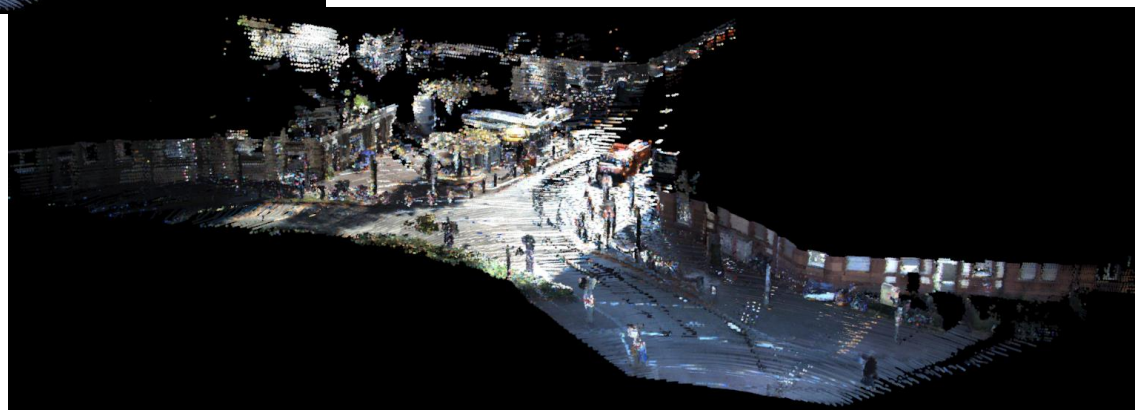


MS using 3D feature trajectories.

3D motion analysis with prior knowledge



Reconstructed static map

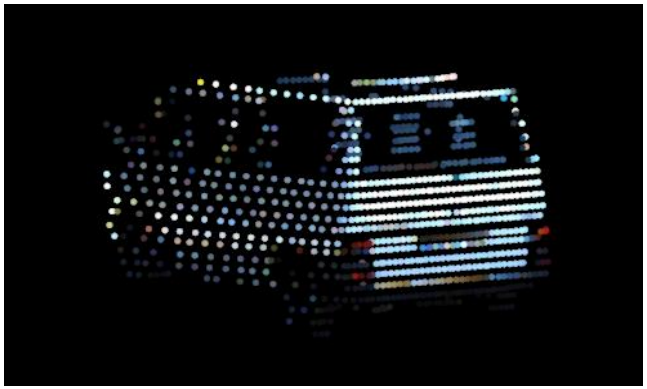


C. Jiang, D. P. Paudel, Y. Fougerolle, D. Fofi, C. Demonceaux. **Static-map and Dynamic Object Reconstruction in Outdoor Environments using 3D Motion Segmentation**. In *IEEE Robotics and Automation Letters*, Vol. 1, Issue 1, January 2016, pp 324-331.

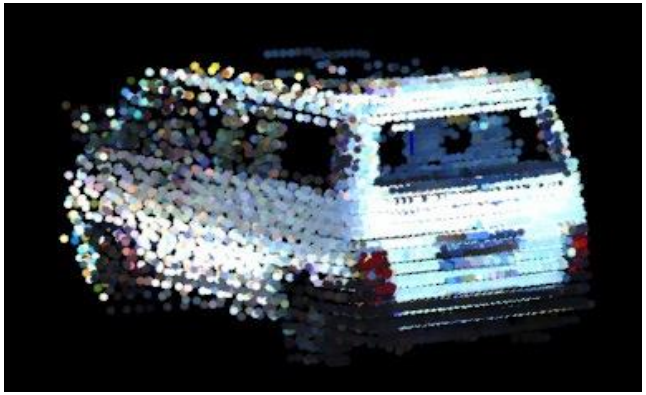
3D motion analysis with prior knowledge

Right View

Right View Reconstruction



Left View



Left View Reconstruction

3D motion analysis with prior knowledge

Thanks to :

Joao Barreto
(University of Coimbra)

Jesus Bermudez-Cameo
(University of Zaragoza)

Remi Boutteau
(Esigelec, Rouen)

Friedrich Fraundorfer
(Graz University of Technology)

Josechu Guerrero
(University of Zaragoza)

Adlane Habed
(Icube)

Gonzalo Lopez-Nicolas
(University of Zaragoza)

Cansen Jiang
(Le2i)

Danda Pani Paudel
(Le2i – ETH Zurich)

Marc Pollefeys
(ETH Zurich)

Olivier Saurer
(ETH Zurich)

Pascal Vasseur
(University of Rouen)

Related publications

Journals

O. Saurer, P. Vasseur, R. Boutteau, C. Demonceaux, M. Pollefeys, F. Fraundorfer. **Homography Based Egomotion Estimation with a Common Direction**. In *PAMI*, Issue 2, February 2017, pp 327-341.

C. Jiang, D. P. Paudel, Y. Fougerolle, D. Fofi, C. Demonceaux. **Static-map and Dynamic Object Reconstruction in Outdoor Environments using 3D Motion Segmentation**. In *IEEE Robotics and Automation Letters*, Vol. 1, Issue 1, January 2016, pp 324-331.

J.C. Bazin, H. Li, I.S. Kweon, C. Demonceaux, P. Vasseur, K. Ikeuchi. **A Branch and Bound Approach to Correspondence and Grouping Problems**. In *PAMI*, Vol 35, Issue 7, Juillet 2013, pp.1565-1576.

J.C. Bazin, C. Demonceaux, P. Vasseur, I.S. Kweon. **Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment.** In *IJRR,* Vol 31, Issue 1, Janvier 2012, pp. 63-81.

Conferences

J. Bermudez-Cameo, C. Demonceaux, G. Lopez-Nicolas, J. Guerrero. **Line reconstruction using prior knowledge in single non-central view**. In BMVC 2016

D.P. Paudel, A. Habed, C. Demonceaux, P. Vasseur. **Robust and Optimal SoS-based Point-to-Plane Registration of Image Sets and Structured Scenes.** In ICCV 2015

D.P. Paudel, A. Habed, C. Demonceaux, P. Vasseur. **LMI-based 2D-3D Registration: from Uncalibrated Images to Euclidean Scene.** In CVPR 2015

O. Saurer, P. Vasseur, C. Demonceaux, F. Fraundorfer. **A homography formulation to the 3pt plus a common direction relative pose problem**. In ACCV 2014

D.P. Paudel, C. Demonceaux, A. Habed, P. Vasseur, I.S. Kweon. **2D-3D Camera fusion for Visual Odometry in outdoor environments.** In IROS 2014

J.C. Bazin, Y.D. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I.S. Kweon, M. Pollefeys. **Globally Optimal Line Clustering and Vanishing Point Estimation in Manhattan World.** In CVPR 2012

C. Demonceaux, P. Vasseur, C. Pégard. **UAV Attitude Computation by Omnidirectional Vision in Urban Environment.** In ICRA 2007.

C. Demonceaux, P. Vasseur, C. Pégard. **Robust Attitude Estimation with Catadioptric Vision.** In IROS 2007.

?

cedric.demonceaux@u-bourgogne.fr