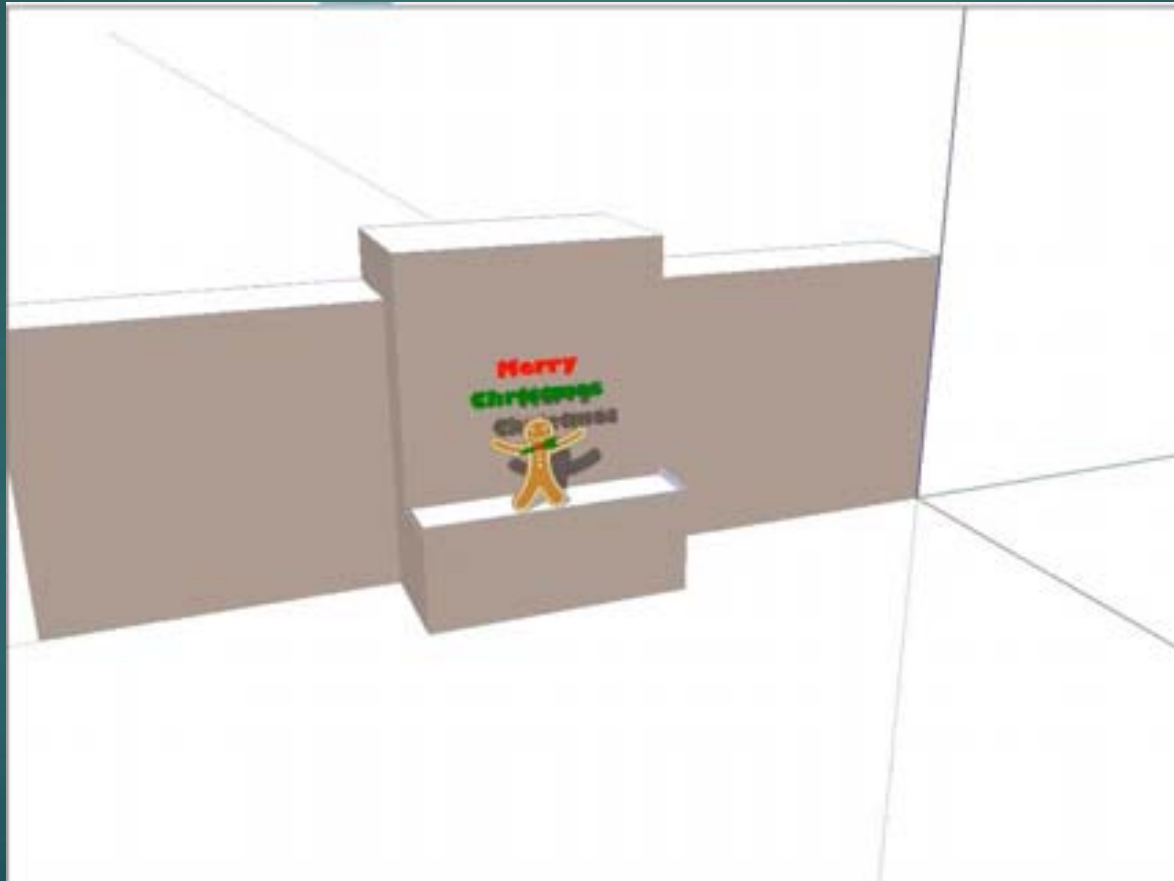


La réalité augmentée

3. CALCUL DE POSE

- 3.1 Utilisation de capteurs physiques
- 3.2 Exploitation des points de fuite
- 3.3 Utilisation d'indices visuels
- 3.4 Régression par un CNN

Le problème du recalage 3D-2D



Le problème du recalage 3D-2D

- ▶ Requiert de disposer d'un modèle de scène pour
 1. Positionner les objets virtuels
 2. Positionner la caméra virtuelle
- ▶ Le principal problème de la RA consiste à aligner la scène virtuelle (modèle ou caméra) avec la scène réelle
- ▶ Cela revient à
 1. Appliquer les paramètres intrinsèques de la caméra réelle à la caméra virtuelle
 2. mesurer ou calculer la transformation rigide ou pose (rotation + translation) entre le repère monde et le repère de la caméra
- ▶ Mesurer = solutions basées sur des capteurs physiques d'orientation et de position de la caméra
- ▶ Calculer = solutions basées sur l'analyse des images vidéo (vision par ordinateur)
 - ▶ Exploitation des points de fuite
 - ▶ Utilisation d'indices visuels
 - ▶ Inférence par réseau de neurones convolutif (CNN)

3. Calcul de pose

3.1 UTILISATION DE CAPTEURS PHYSIQUES

Positionnement basé capteurs

- ▶ Position : GPS
- ▶ Orientation : centrale inertielle



© Creative commons



© Application Métro Paris sur iPhone et iPod Touch

Positionnement basé capteurs

- ▶ Pour l'intérieur : triangulation wifi ou autres capteurs basés sur des ondes électro-magnétiques ou acoustiques (+ centrale inertielle)



Positionnement basé capteurs

Précision
insuffisante pour
certaines
applications



Positionnement basé capteurs

Précision
insuffisante pour
certaines
applications

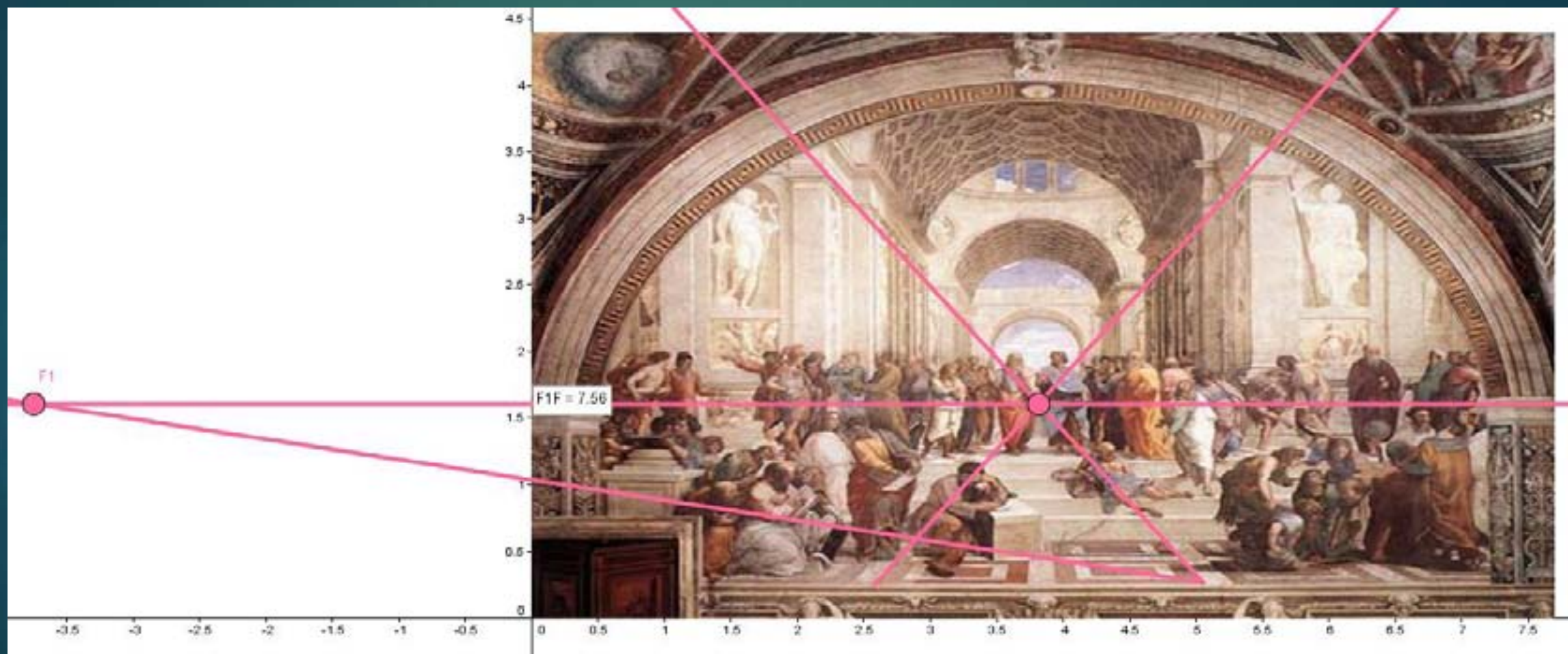


3. Calcul de pose

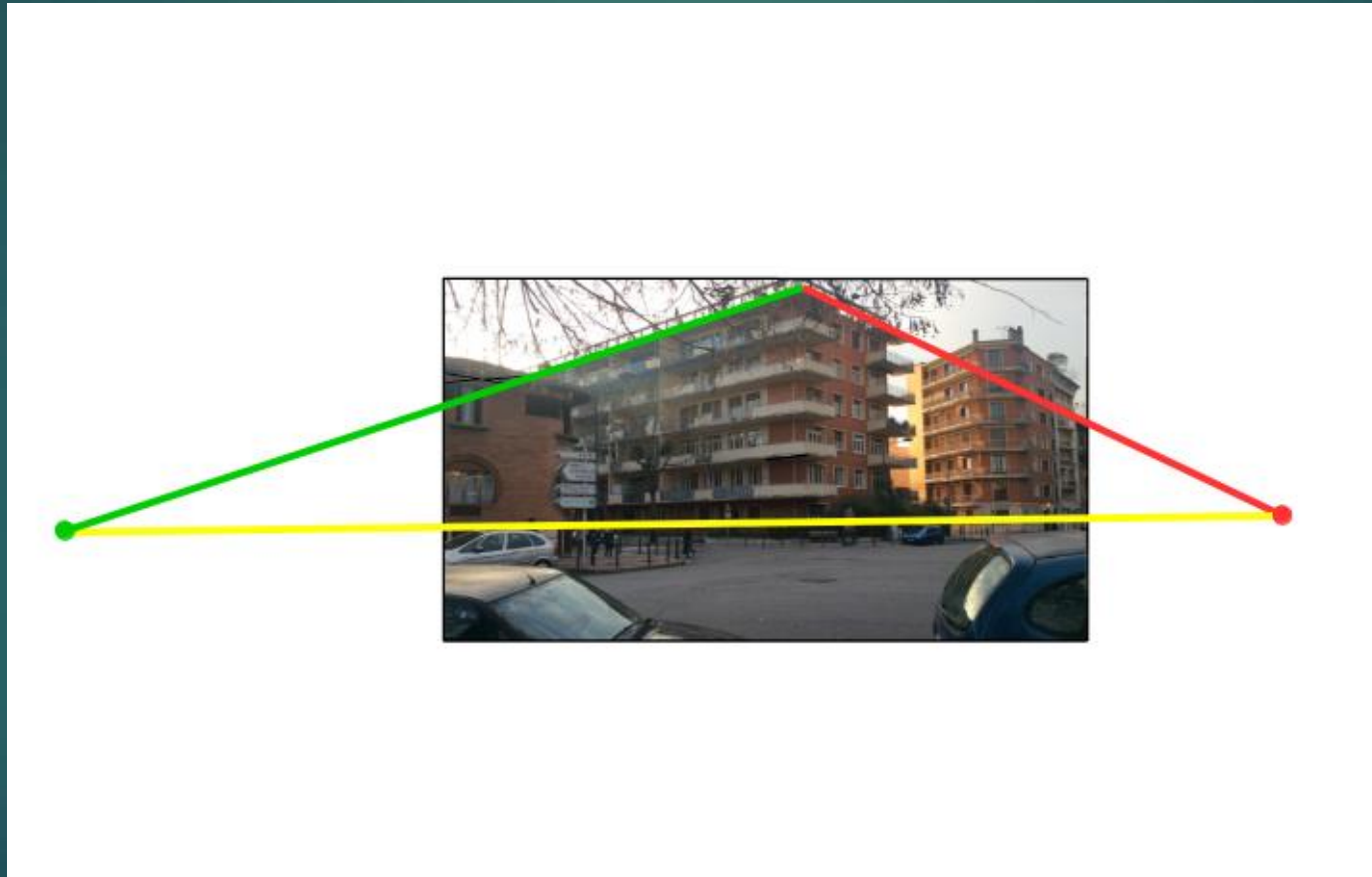
3.2 EXPLOITATION DES POINTS DE FUITE

Des points de fuite au point de vue

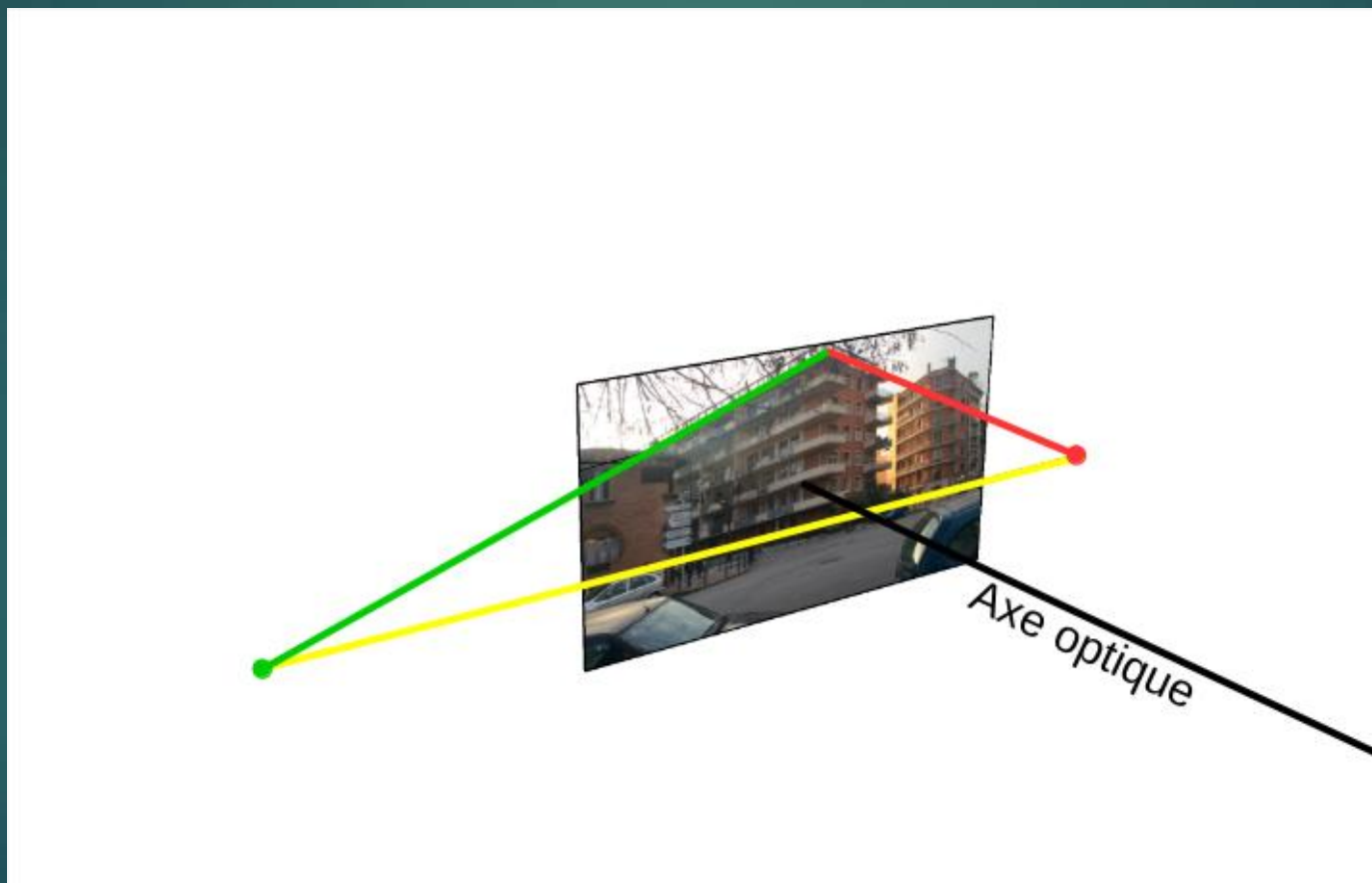
11



Des points de fuite au point de vue

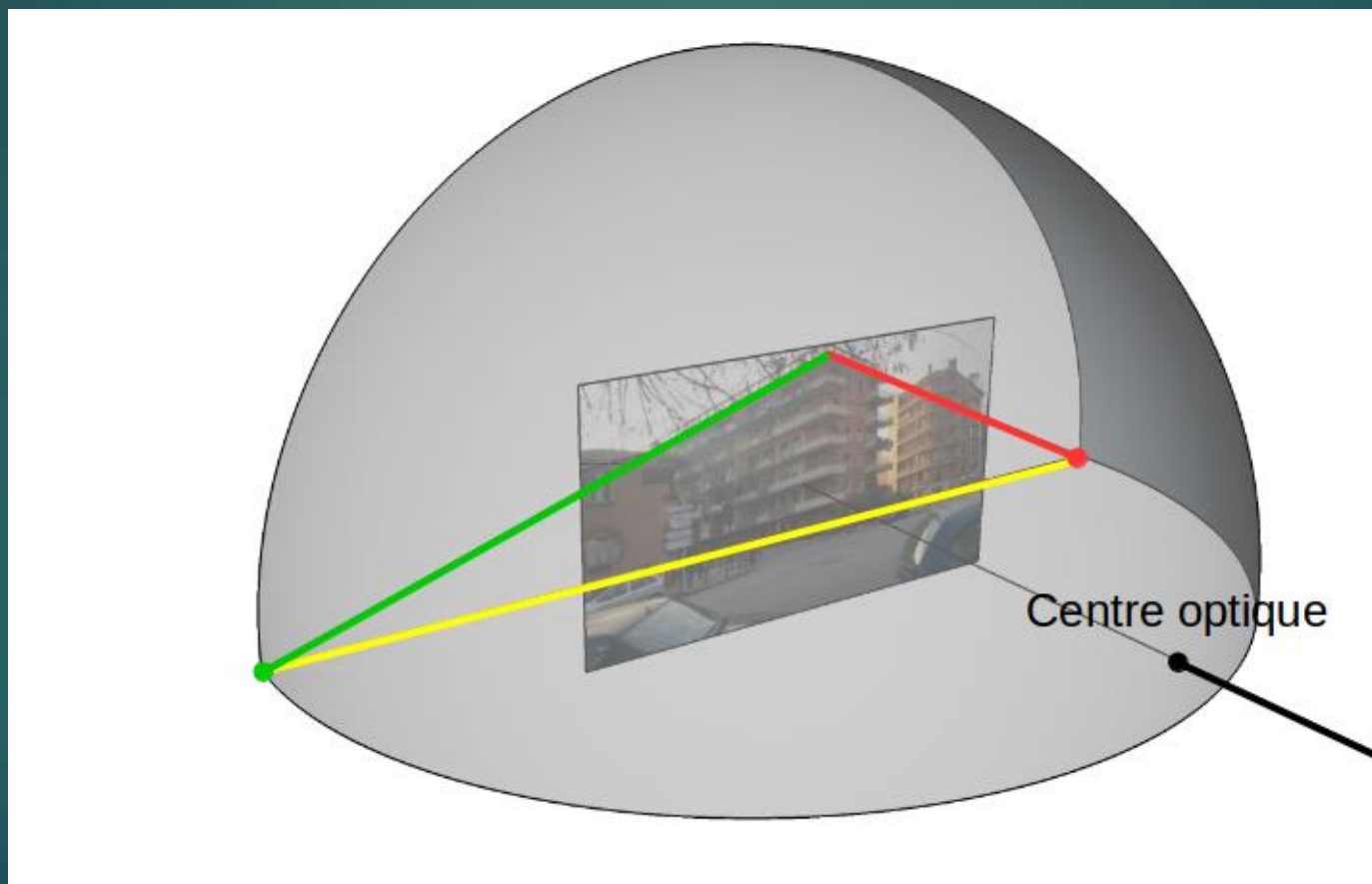


Des points de fuite au point de vue



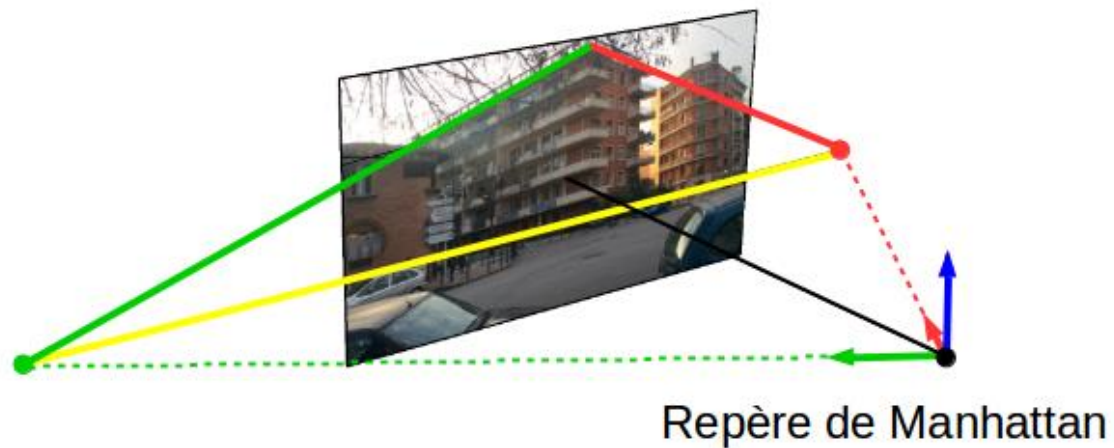
Des points de fuite au point de vue

14



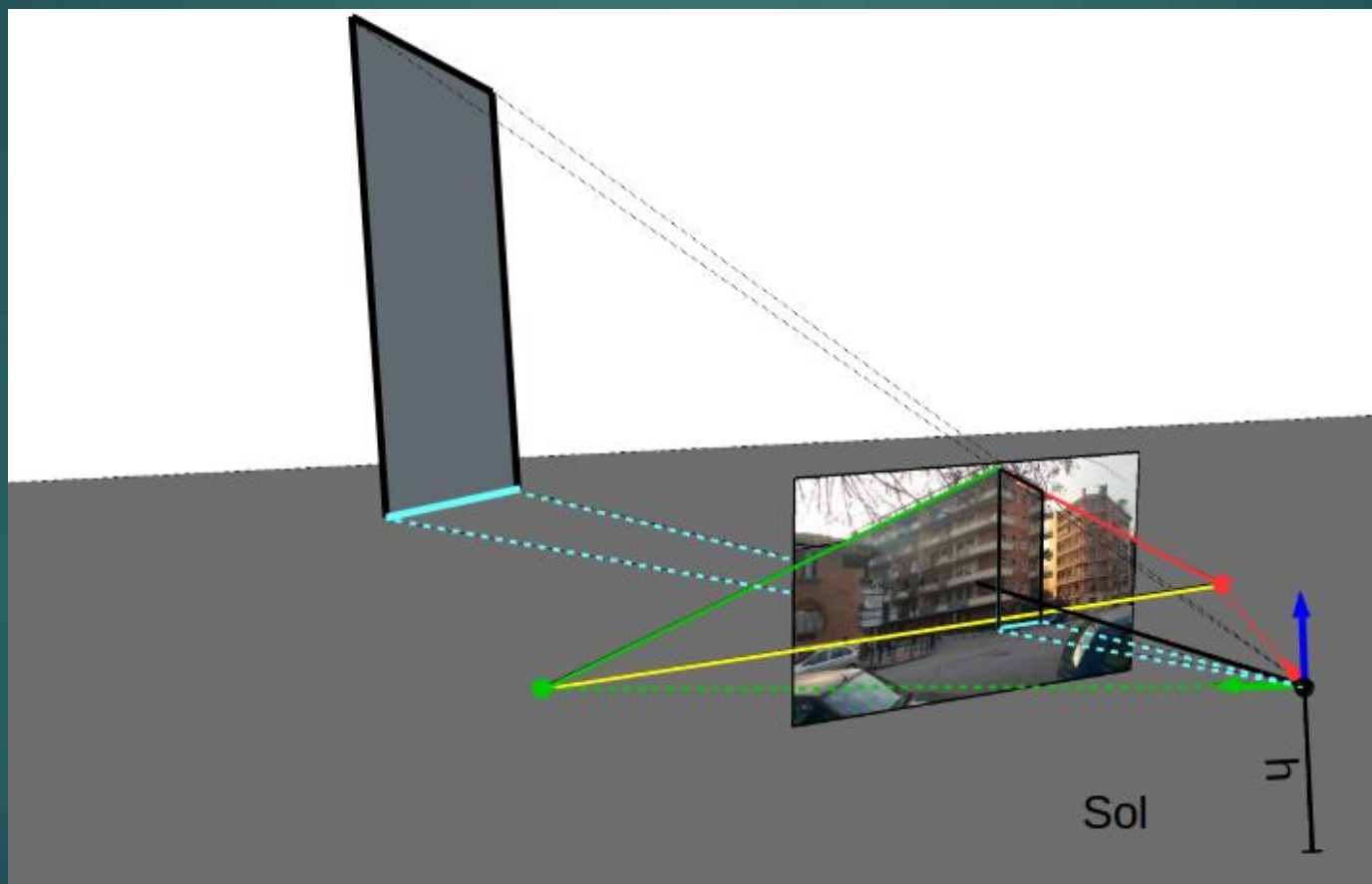
Des points de fuite au point de vue

15

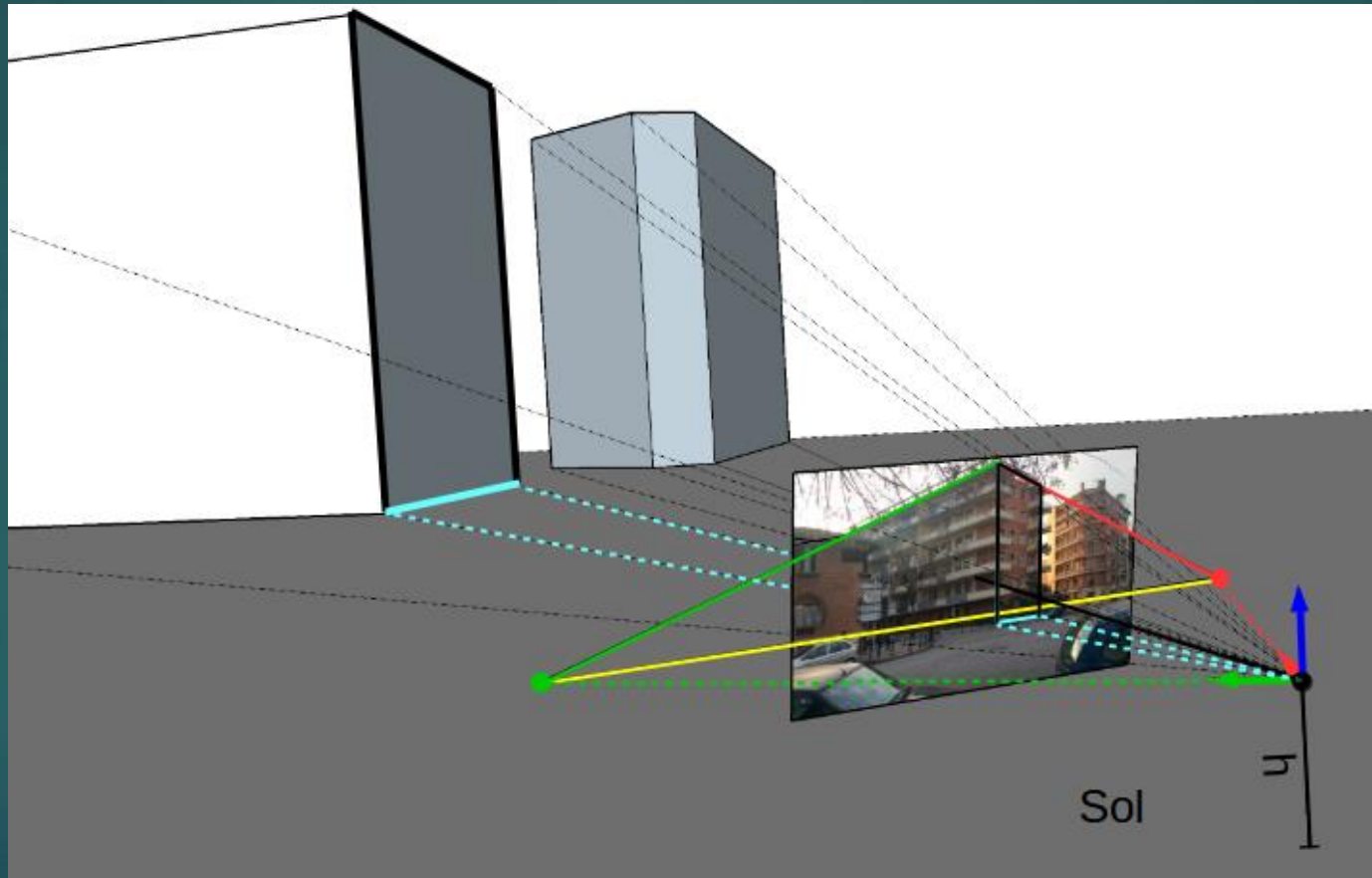


Des points de fuite au point de vue

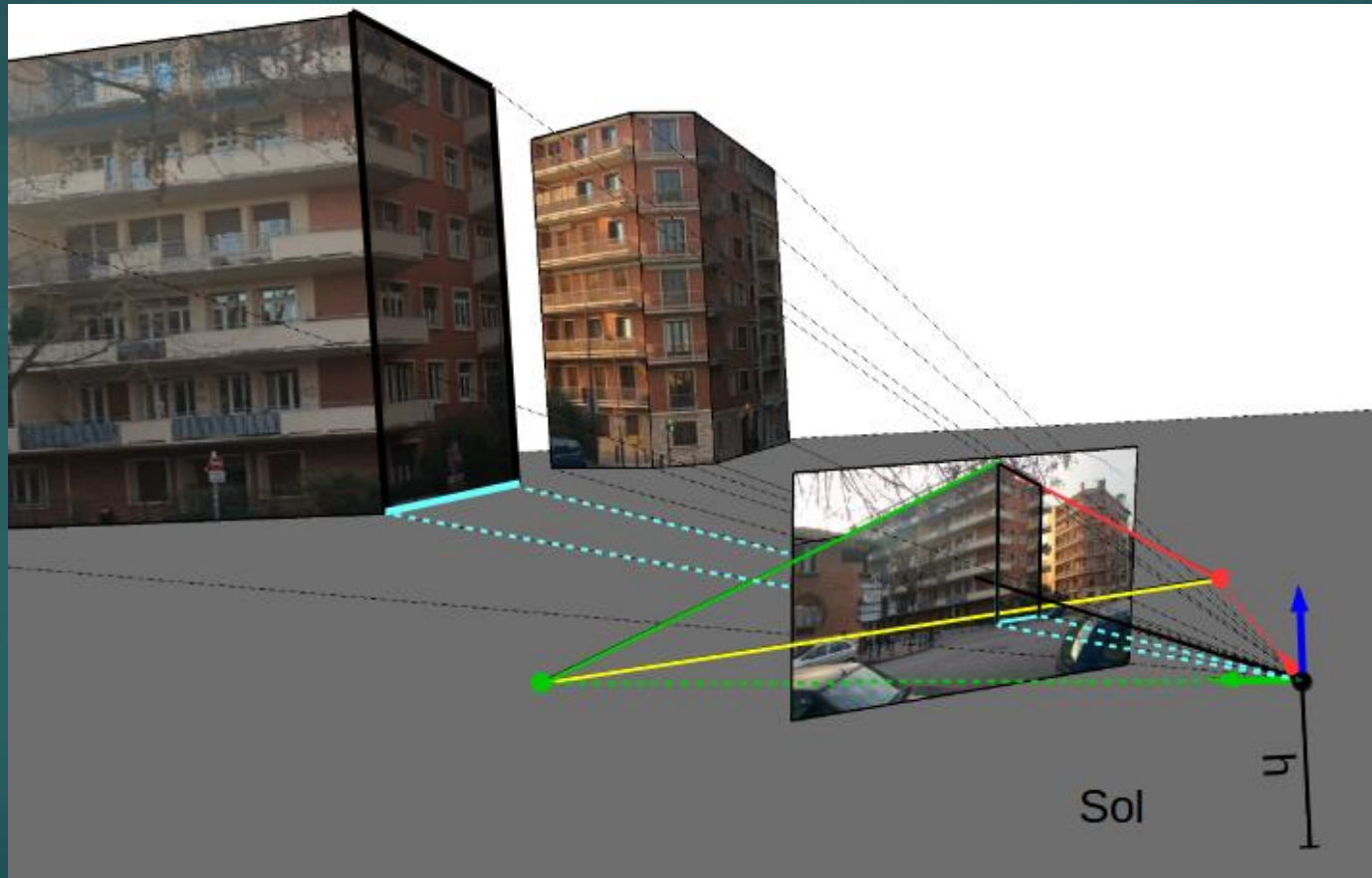
16



Des points de fuite au point de vue



Des points de fuite au point de vue



Calcul de la focale

- ▶ $\mathbf{v}_1, \mathbf{v}_2$ mesurés en coordonnées homogènes dans l'image :
 - $\mathbf{V}_1 = \mathbf{K}^{-1}\mathbf{p}_1$: vecteur 3D parallèle à la direction du point de fuite p_1 dans le repère caméra
 - $\mathbf{V}_2 = \mathbf{K}^{-1}\mathbf{p}_2$: vecteur 3D parallèle à la direction du point de fuite p_2 dans le repère caméra
- ▶ Les deux directions sont perpendiculaires $\Leftrightarrow \mathbf{V}_1^T \mathbf{V}_2 = 0$
 - $\Rightarrow (\mathbf{K}^{-1}\mathbf{p}_1)^T \mathbf{K}^{-1}\mathbf{p}_2 = 0$
 - $\Rightarrow \mathbf{p}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{p}_2 = 0$
 - $\Rightarrow \mathbf{p}_1^T (\mathbf{K}\mathbf{K}^T)^{-1} \mathbf{p}_2 = 0$

$$\mathbf{K} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \approx \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \Rightarrow \omega = (\mathbf{K}\mathbf{K}^T)^{-1} = \begin{pmatrix} \frac{1}{f^2} & 0 & 0 \\ 0 & \frac{1}{f^2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \Rightarrow f^2 = -(u_v u_w + v_v v_w) \text{ si } \mathbf{p}_1 = (u_v, v_v, 1) \text{ et } \mathbf{p}_2 = (u_w, v_w, 1)$$

Calcul de la rotation

$$\mathbf{r}_1 = \mathbf{K}^{-1}\mathbf{p}_1 / \|\mathbf{K}^{-1}\mathbf{p}_1\|$$

$$\mathbf{r}_2 = \mathbf{K}^{-1}\mathbf{p}_2 / \|\mathbf{K}^{-1}\mathbf{p}_2\|$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

$$\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$$

Exemple : photomatch (Sketchup)

21



Exemple : photomatch (Sketchup)

22



Exemple : photomatch (Sketchup)

23



Exemple : photomatch (Sketchup)

24

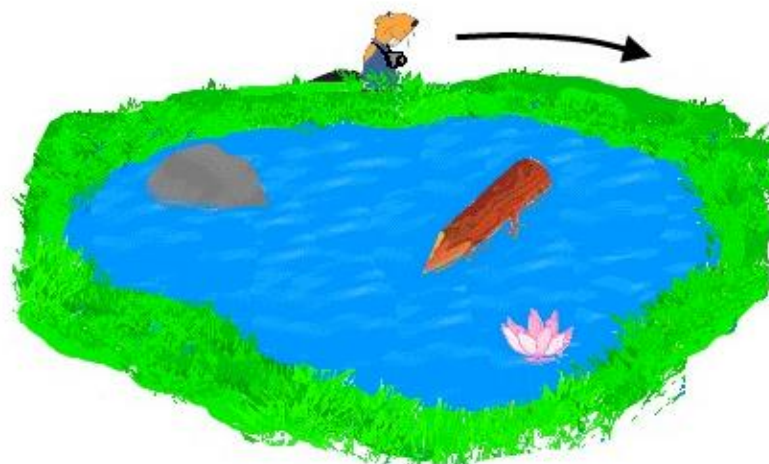


3. Calcul de pose

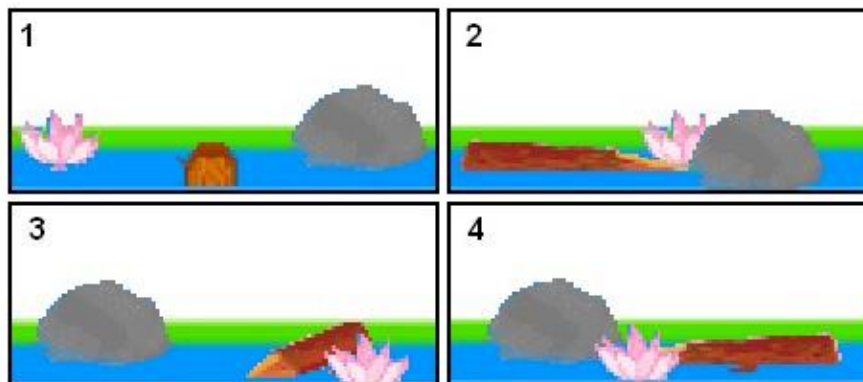
3.3 UTILISATION D'INDICES VISUELS

Illustration

Castor se promène autour d'un étang. Il commence sa promenade à l'endroit où vous le voyez sur le dessin et part dans la direction de la flèche.



Castor prend quatre photographies au cours de sa promenade.

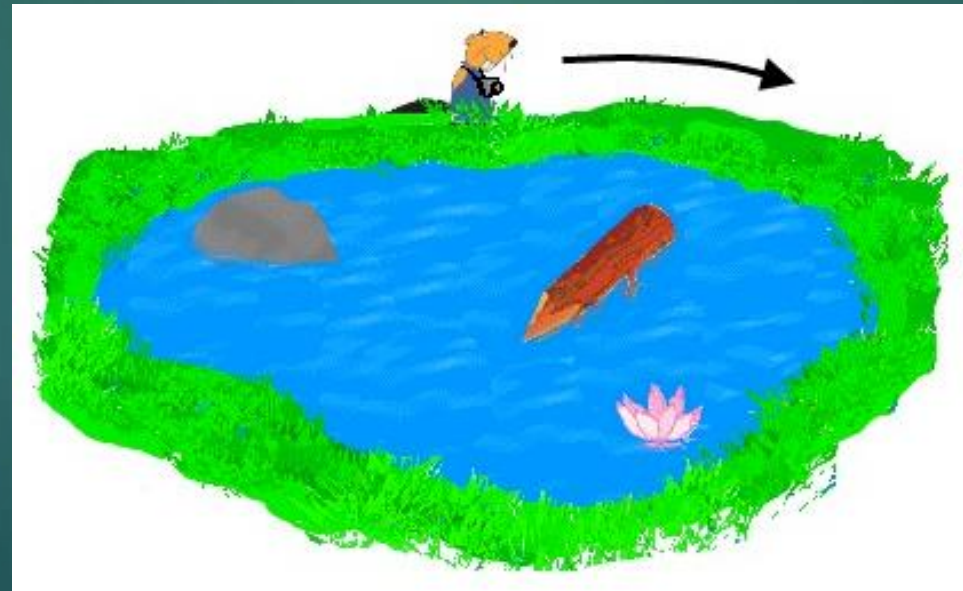


Dans quel ordre a-t-il pris les photographies ?

Illustration

27

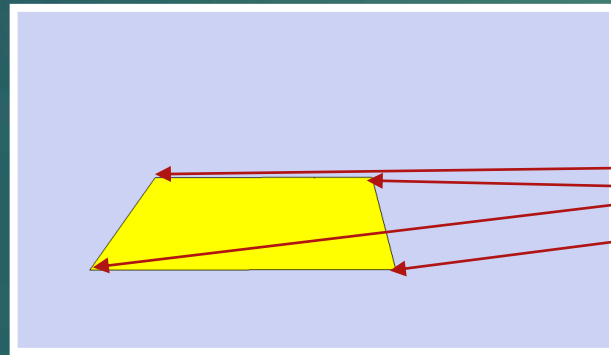
- Pour répondre à cette question, nous avons considéré des indices visuels (tronc d'arbre, nénuphar, rocher) dont les positions relatives sont connues dans un modèle de scène (plan de la mare)



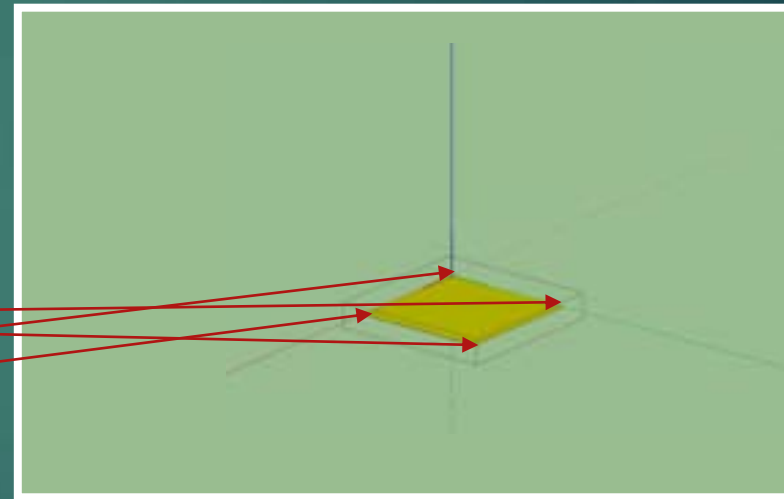
Positionnement visuel

28

- ▶ De même, les approches classiques de positionnement visuel reposent sur un **appariement d'indices visuels** entre un modèle 3D et les images acquises
- ▶ En particulier, on peut montrer qu'il suffit d'apparier quatre points (coplanaires ou non) pour calculer la pose, si les intrinsèques de la caméra sont connus



Image

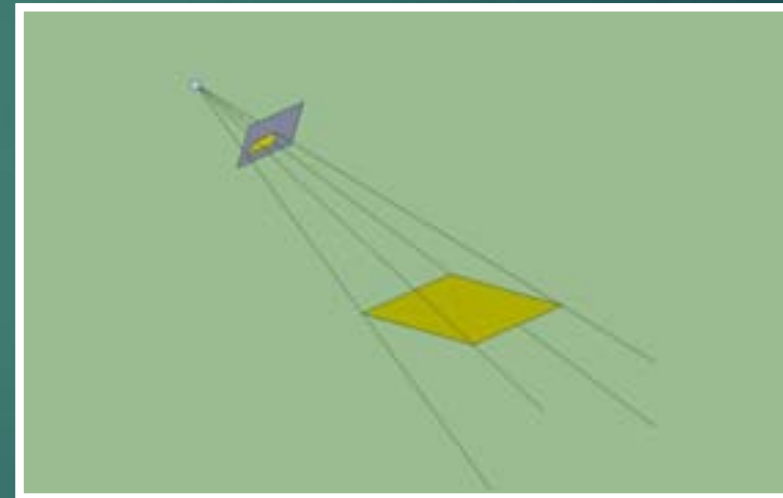
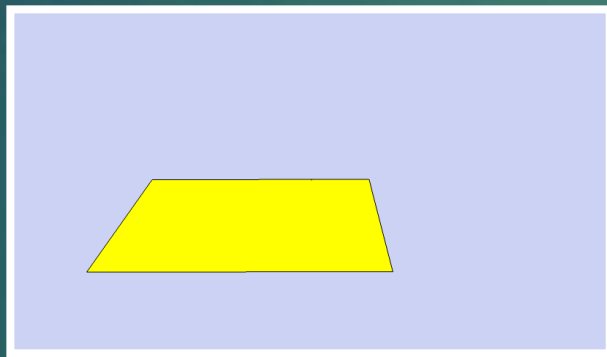


Modèle 3D

Positionnement visuel

29

- ▶ De même, les approches classiques de positionnement visuel reposent sur un **appariement d'indices visuels** entre un modèle 3D et les images acquises
- ▶ En particulier, on peut montrer qu'il suffit d'apparier quatre points (coplanaires ou non) pour calculer la pose, si les intrinsèques de la caméra sont connus



- ▶ A priori plus précis pour la RA que les méthodes basées capteurs (ancrage de la scène virtuelle à l'image réelle)

La méthode DLT

30

Chaque correspondance de points
3D/2D donne lieu à deux équations :

$$\tilde{\mathbf{m}}_i = \mathbf{P}\tilde{\mathbf{M}}_i \quad \left\{ \begin{array}{l} u_i = \frac{P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} \\ v_i = \frac{P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} \end{array} \right.$$

$$\begin{cases} P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14} - P_{31}X_i u_i - P_{32}Y_i u_i - P_{33}Z_i u_i - P_{34} u_i = 0 \\ P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24} - P_{31}X_i v_i - P_{32}Y_i v_i - P_{33}Z_i v_i - P_{34} v_i = 0 \end{cases}$$

$$\begin{pmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -X_i u_i & -Y_i u_i & -Z_i u_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -X_i v_i & -Y_i v_i & -Z_i v_i & -v_i \end{pmatrix} \begin{pmatrix} P_{11} \\ \dots \\ P_{34} \end{pmatrix} = \mathbf{0}$$

- ▶ 11 coefficients à estimer
- ▶ Chaque correspondance 3D-2D donne 2 equations
→ 6 correspondances doivent être connues

L'algorithmme de calibration DLT

31

► Extraction de la matrice \mathbf{K}

On considère la matrice \mathbf{P}_3 constituée des 3 premières colonnes de \mathbf{P} :

$$\mathbf{P} = (\mathbf{P}_3 \mid \mathbf{c}_4)$$
$$\mathbf{P}_3 = \mathbf{K}\mathbf{R}$$

On calcule

$$\mathbf{P}_3\mathbf{P}_3^T = (\mathbf{K}\mathbf{R})(\mathbf{K}\mathbf{R})^T = \mathbf{K}\mathbf{R}\mathbf{R}^T\mathbf{K}^T = \mathbf{K}\mathbf{K}^T$$

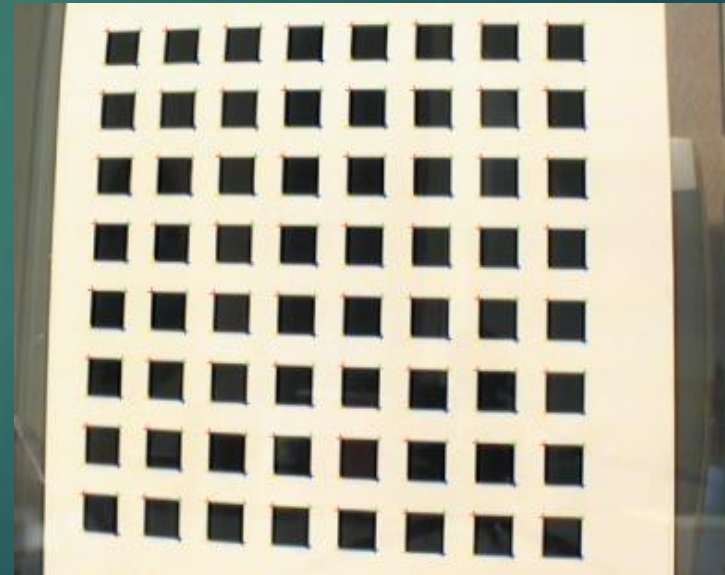
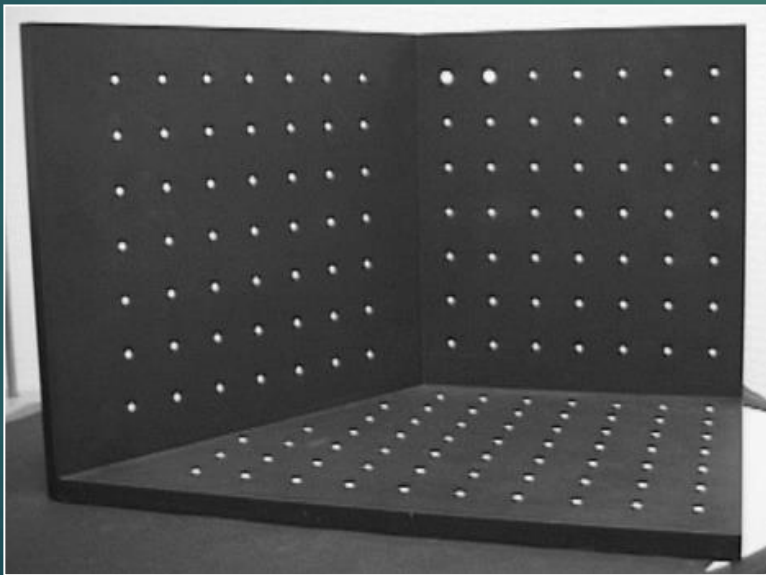
\mathbf{K} étant une matrice triangulaire (supérieure), elle peut être calculée en utilisant la méthode de factorization de Cholesky

La matrice $\omega = (\mathbf{K}\mathbf{K}^T)^{-1}$ est appelée image de la conique absolue

Calibration de la caméra

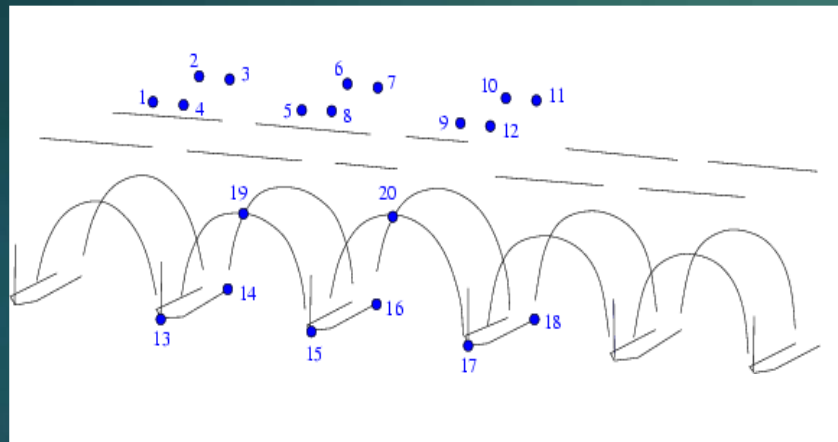
32

- ▶ Les paramètres intrinsèques de la caméra restent fixes pour un dispositif de RA donné
- ▶ Ils sont donc généralement calculés en amont, à l'aide de la méthode DLT utilisant une « mire de calibration »

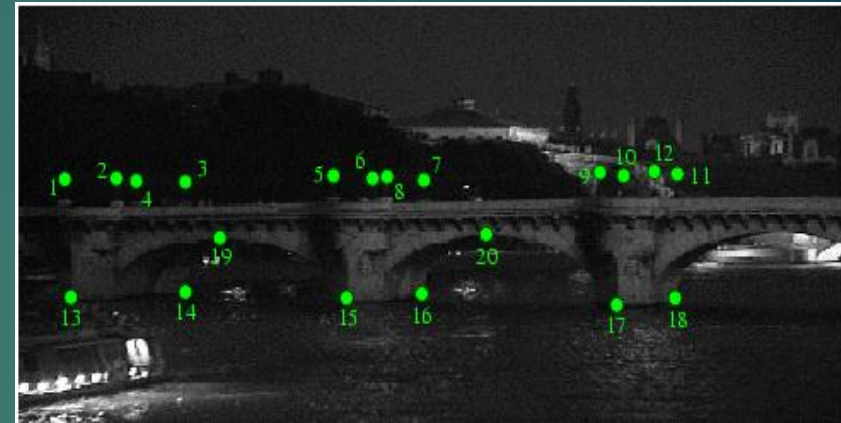


Le problème PnP

- ▶ PnP (Perspective-n-Points) : estimation de l'orientation \mathbf{R} et de la position \mathbf{T} du repère monde par rapport au repère caméra (calibrée), à partir de n correspondances de points 3D/2D

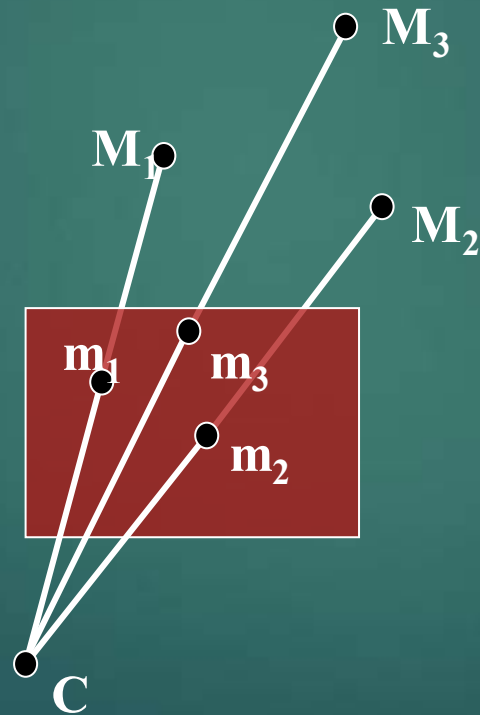


↔
Correspondances
3D/2D



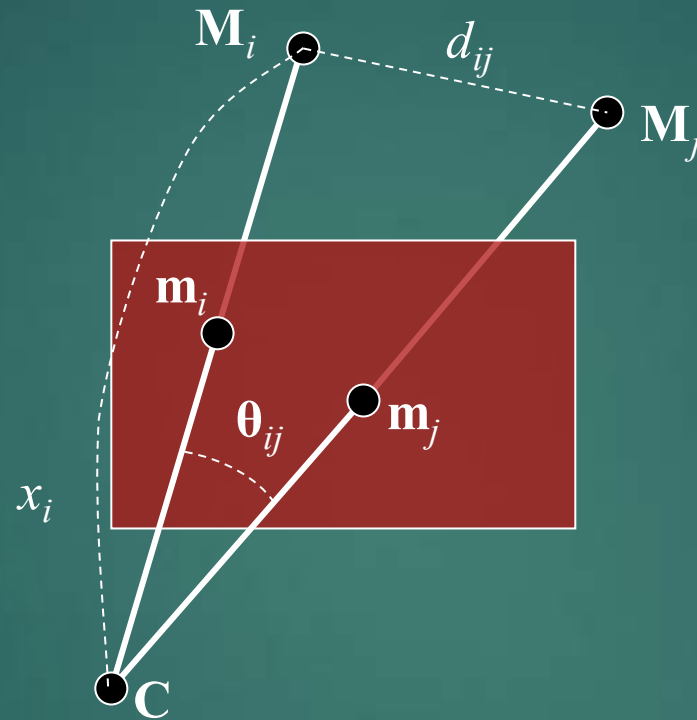
La méthode P3P

- ▶ P3P : suffisant en théorie mais 4 solutions \rightarrow ajout de contrainte nécessaire



La méthode P3P

35



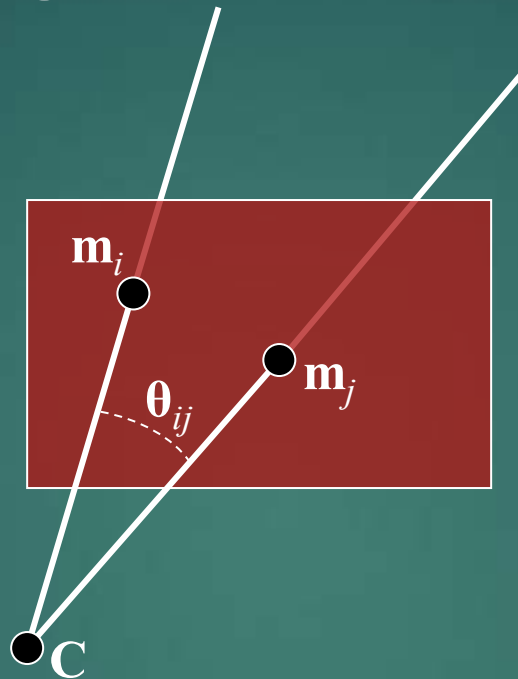
Chaque paire de correspondances $\mathbf{M}_i \leftrightarrow \mathbf{m}_i$ et $\mathbf{M}_j \leftrightarrow \mathbf{m}_j$ donne une contrainte sur les distances (inconnues) $x_i = ||\mathbf{M}_i - \mathbf{C}||$ et $x_j = ||\mathbf{M}_j - \mathbf{C}||$:

$$d_{ij}^2 = x_i^2 + x_j^2 - 2 x_i x_j \cos \theta_{ij},$$

$d_{ij} = ||\mathbf{M}_i - \mathbf{M}_j||$ est la distance (connue) entre \mathbf{M}_i et \mathbf{M}_j

θ_{ij} est l'angle (connu) entre $(\mathbf{C}\mathbf{M}_i)$ et $(\mathbf{C}\mathbf{M}_j)$

La méthode P3P



$\cos\theta_{ij}$ depend uniquement des positions 2D (connues) \mathbf{m}_i et \mathbf{m}_j .

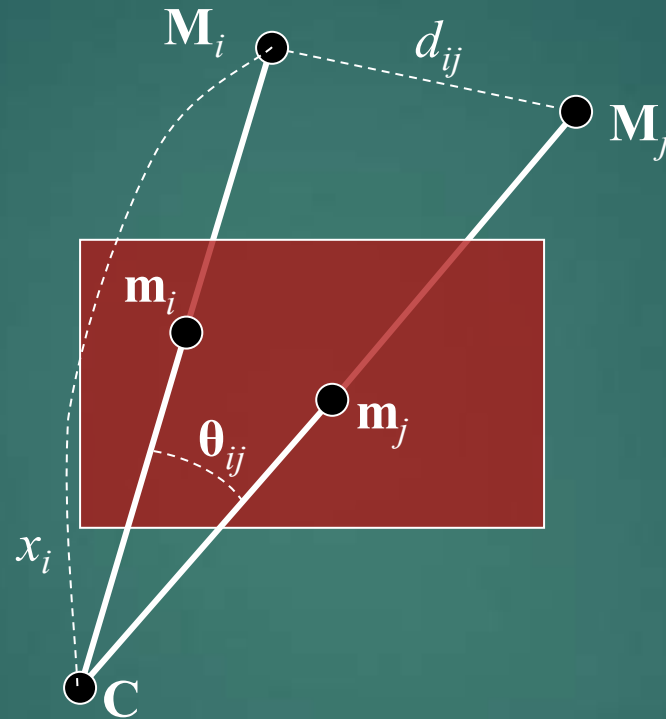
$$\cos\theta_{ij} = \frac{(\overrightarrow{\mathbf{Cm}_j})^T (\overrightarrow{\mathbf{Cm}_i})}{\|\overrightarrow{\mathbf{Cm}_j}\| \|\overrightarrow{\mathbf{Cm}_i}\|} = \frac{(\overrightarrow{\mathbf{Cm}_j})^T (\overrightarrow{\mathbf{Cm}_i})}{\sqrt{(\overrightarrow{\mathbf{Cm}_j})^T (\overrightarrow{\mathbf{Cm}_j})} \sqrt{(\overrightarrow{\mathbf{Cm}_i})^T (\overrightarrow{\mathbf{Cm}_i})}}$$

$$\overrightarrow{\mathbf{Cm}_i} = \mathbf{K}^{-1} \mathbf{m}_i$$

$$\cos\theta_{ij} = \frac{\mathbf{m}_j^T \omega \mathbf{m}_i}{(\mathbf{m}_j^T \omega \mathbf{m}_j)^{1/2} (\mathbf{m}_i^T \omega \mathbf{m}_i)^{1/2}} \text{ avec } \omega = (\mathbf{K}\mathbf{K}^T)^{-1} \text{ l'image de la conique absolue}$$

La méthode P3P

37



Algorithme :

1. Résoudre les distances x_i
2. Calculer les positions \mathbf{M}_i^C des points \mathbf{M}_i dans le repère camera
3. Calculer \mathbf{R} et \mathbf{T} comme la transformation rigide entre les \mathbf{M}_i et les \mathbf{M}_i^C

Méthodes PnP

- ▶ RANSAC P3P ou autres méthodes directes
- ▶ Affinement utilisant une optimisation non linéaire (Gauss-Newton, Levenberg Marquardt, ...) tenant compte de tous les appariements disponibles :

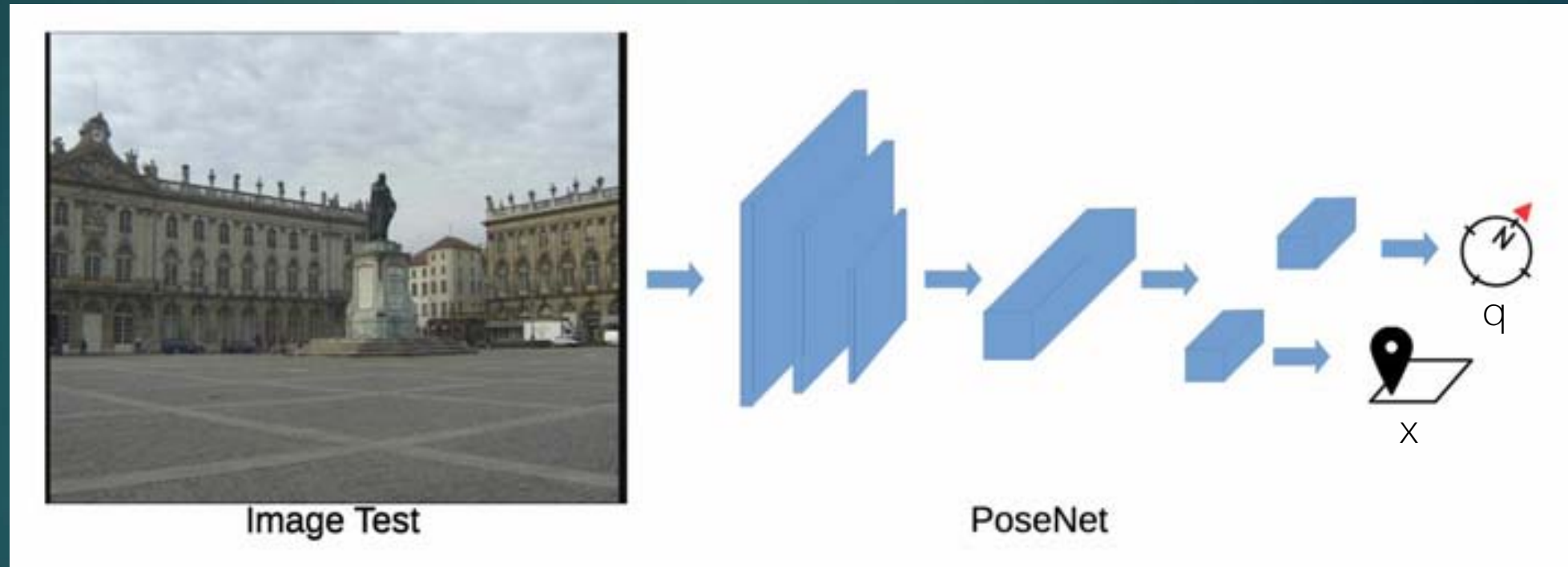
$$\min_{\mathbf{R}, \mathbf{T}} \sum_i \|\mathbf{K}[\mathbf{R} \mid \mathbf{T}]\mathbf{M}_i - \mathbf{m}_i\|^2$$

3. Calcul de pose

3.4 RÉGRESSION PAR UN CNN

PoseNet

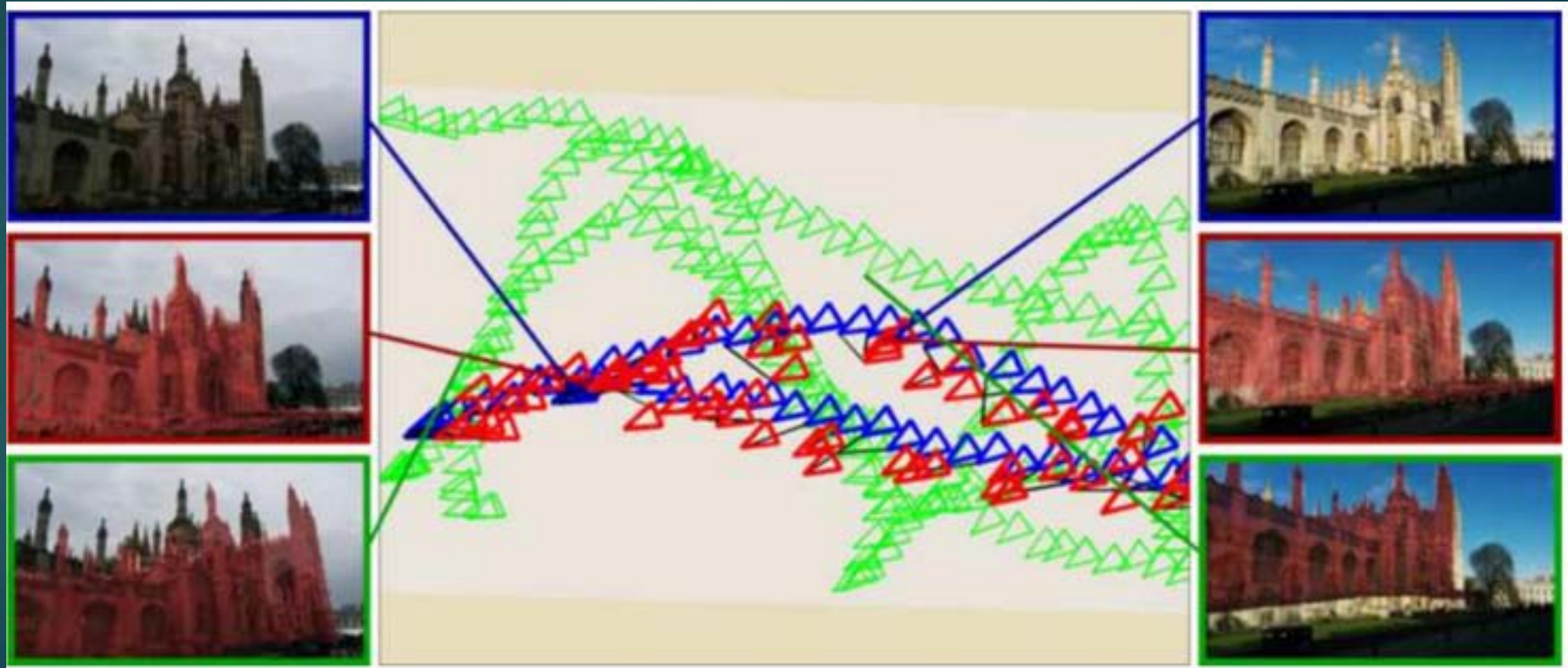
- PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. Kendall et al., ICCV'2015



$$loss(I) = \|\hat{\mathbf{x}} - \mathbf{x}\|_2 + \beta \left\| \hat{\mathbf{q}} - \frac{\mathbf{q}}{\|\mathbf{q}\|} \right\|_2$$

PoseNet

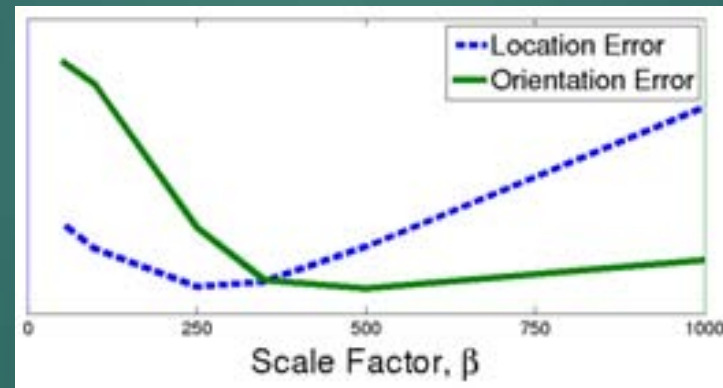
- ▶ PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. Kendall et al., ICCV'2015



PoseNet

42

- ▶ PoseNet: *A Convolutional Network for Real-Time 6-DOF Camera Relocalization*. Kendall et al., ICCV'2015
- ▶ Problèmes :
 - ▶ Problème du choix de β



- ▶ Généralement moins précis qu'un calcul PnP
- ▶ Problème de la représentativité des données (équivalent à une recherche par plus proche voisin, cf. [Sattler 2019])
- ▶ Doit être réappris pour chaque nouvelle scène