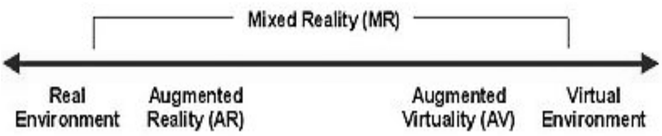
Réalité Augmentée

Jean-Baptiste Fasquel

Définition

- Concept introduit par Paul Milgram (université de Toronto, Canada) en 1994 : « Augmented Reality: A class of displays on the reality-virtuality continuum » SPIE Vol. 2351, Telemanipulator and Telepresence Technologies (1994)
- « The virtuality continuum is a continuous scale ranging between the completely virtual, a virtuality, and the completely real, reality. »
- « is said to consist of both augmented reality, where the virtual augments the real, and augmented virtuality, where the real augments the virtual. »



Quelques applications (prototypes ou existantes)

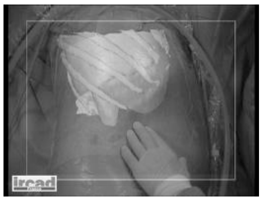
Information

Tourisme

Jeux

Quelques applications (prototypes ou existantes)

Automobile



Médecine

Ameublement

Opticien

Disciplines fondamentales

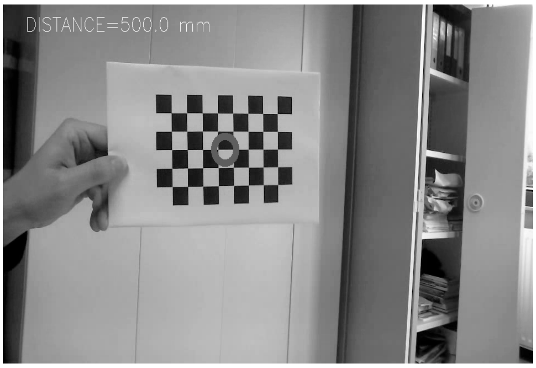
- Visualisation 3D
- Géométrie projective : pour correctement positionner une information virtuelle dans un monde réel (ou une vue du monde réel) : le monde 3D → le monde 2D (projection et affichage sur un écran 2D)
  - Référence : « Multiple View Geometry », Hartley and Zissermann, 2004
  - Supports de cours intéressants :
    - « Robotics – Projective Geometry and Camera Model », Matteo Pirotta, Politecnico di Milano (Italy), 2014
    - « Vision par ordinateur », Gilles Simon, Université de Lorraine, 2014
    - « Computer Vision », Robert Collins, Penn State University (USA), 2007
- Traitement d'images
  - Souvent requis pour détecter des points d'intérêt et recaler l'objet virtuel
  - Mise en correspondance d'images (ou de points d'intérêt)

Quelques frameworks

- ARToolWorks : http://www.artoolworks.com
  - Java, Flash, C++...
- WorldViz : http://www.worldviz.com
  - Généraliste réalité virtuelle (3D, Kinect, Cave,...), orienté C++/Python
  - Partie réalité augmentée : utilise ARToolWorks (avec bindings Python)
- OpenCV : http://opencv.org/
  - Module camera calibration and 3D reconstruction
- Vuforia : https://developer.vuforia.com
  - Orienté applications de vision pour plateformes mobiles
  - Java / C++ / Unity extension
- Metaio : http://www.metaio.com/
  - Langage de script AREL (javascript)
- …

Objectif / Périmètre du cours

- Objectif
  - Familiarisation à la réalité augmentée et plus particulièrement aux disciplines sous-jacentes que sont la géométrie projective et le traitement d'images
- Périmètre : réalisation de 2 fonctionnalités particulières
  - Estimation de la distance d'un objet à la caméra

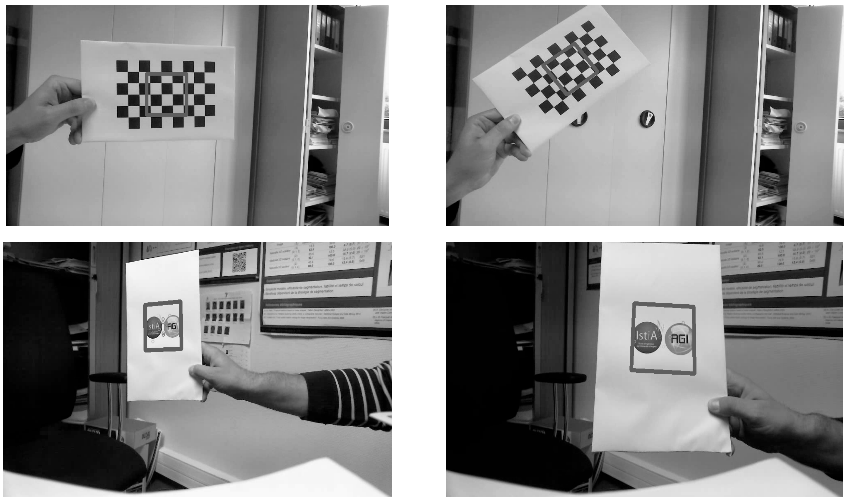


Objectif / Périmètre du cours

- Périmètre : réalisation de 2 fonctionnalités particulières
  - Recalage d'un objet « virtuel » sur la vue 2D réelle

Objectif / Périmètre du cours

- Périmètre : réalisation de 2 fonctionnalités particulières
  - En considérant deux « motifs » (damier et motif plus générique)



Plan du cours

- Vision monoculaire : on considère une seule caméra
- Modélisation de la vision monoculaire :
  - Matrice des paramètres intrinsèques : le modèle de sténopé
    - lié à la caméra (formation de l'image).
  - Matrice des paramètres extrinsèques :
    - lié au déplacement de la caméra et/ou des objets dans le monde réel.
  - Matrice de projection : matrice extrinsèque + matrice intrinsèque
- Recalage : estimation de la matrice de projection

Modèle du Sténopé

Le modèle du sténopé

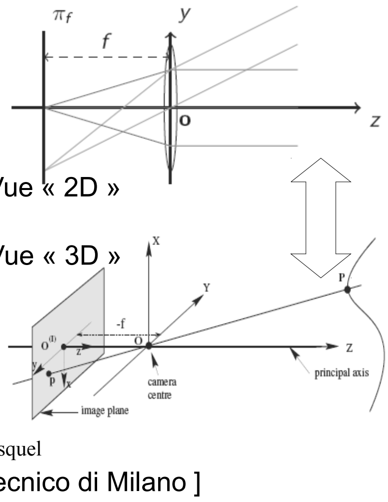
- En anglais : « pinhole model » - pinhole : « trou d'épingle »
- Modélisation simple de la formation des images au sein d'une caméra

Système optique réel : exemple

Système optique approximé par une lentille mince

Vue « 2D »

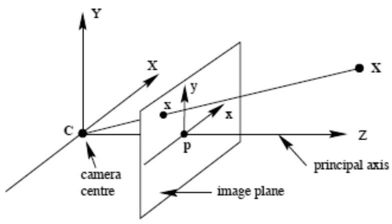
Vue « 3D »

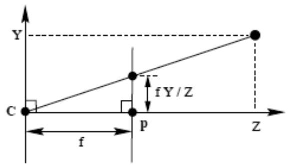


[source : Matteo Pirotta, Politecnico di Milano]

Le modèle du sténopé

- Modélisation simple de la formation des images au sein d'une caméra
  - Modèle du sténopé (modèle « frontal » : formalisme plus simple car plan image « Z positif »
    - (C,X,Y,Z) : Repère camera (centre en C)
    - f : distance focale
    - (p, x,y) : plan image « frontal », perpendiculaire à l'axe optique CZ.
    - p : point principal (intersection du plan image avec CZ, en Z=f)

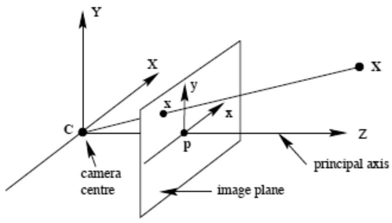


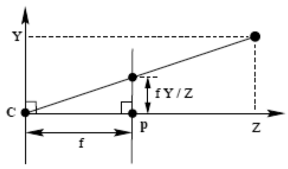


[source : Hartley and Zissermann, « Multiple View Geometry »]

Le modèle du sténopé

- Coordonnées dans le plan image (capteur)
  - Pour un point X (coordonnées (X,Y,Z)) dans le répère caméra, on obtient le point de coordonnées (x,y) dans le plan image:
    - Thalès : y/Y=f/Z (et x/X=f/Z) $\rightarrow$ x=fX/Z et y=fY/Z
    - Projection : (X,Y,Z) $\rightarrow$ (fX/Z,fY/Z)





[source : Hartley and Zissermann, « Multiple View Geometry »]

Le modèle du sténopé

- Coordonnées dans le plan image (capteur) : expression matricielle & coordonnées homogènes (voir annexes)

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}.$$

  - Le point (X,Y,Z,1) correspond au point 3D dans le repère caméra, exprimé en coordonnées homogènes (dernier terme)
  - Le point (fX,fY,Z) correspond au point 2D (dans le plan image), exprimé en coordonnées homogènes (dernier terme « Z »).
  - Pour obtenir les coordonnées « réelles » dans le plan image, on normalise par rapport au dernier terme :
    - (fX,fY,Z) → (x=fX/Z , y=fY/Z , 1) → (x=fX/Z , y=fY/Z)
- En résumé :
  - (X,Y,Z,1) → matrice → (fX,fY,Z) → normalisation → (fX/Z,fY/Z,1) → (fX/Z,fY/Z)

[source : Hartley and Zissermann, « Multiple View Geometry »]

Le modèle du sténopé

- Coordonnées dans le plan image (du capteur) : expression matricielle

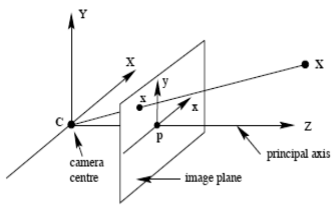
$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}.$$

- Pour simplifier, on utilisera dans la suite la matrice K = $\boxed{\begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}}$

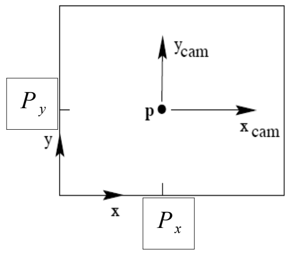
$$\begin{bmatrix} f & & & 0 \\ & f\text{c} & & 0 \\ & & 1 & 0 \end{bmatrix} = \boxed{\begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}} \times \boxed{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}$$

Le modèle du sténopé

- Changement d'origine dans le plan image (« coin de l'image »)
  - (x_cam,y_cam) → (x,y)



Repère caméra



Plan image

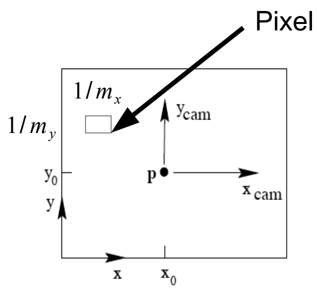
$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}$$

Matrice

  - Exercice : prenons le point 3D de coordonnées (0,0,1) (repère caméra) :
    - Coordonnées de sa projection dans le plan image (x_cam,y_cam) ?
    - Coordonnées de sa projection dans le plan image (x,y) ?

Le modèle du sténopé

- Echantillonnage spatial (« pixelisation ») : coordonnées exprimées en pixels



$$K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

$-m_x, m_y : \textit{nombre de pixels par unité de distance} (\textit{e.g. par millimètre})$

$-1/m_x, 1/m_y : \textit{taille du pixel}$

$-x_0 = m_x * p_x, y_0 = m_y * p_y : \textit{centre optique} (\textit{en pixels})$

$-\alpha_x = f * m_x, \alpha_y = f * m_y$

Le modèle du sténopé

- En résumé : le passage du monde 3D caméra à l'image numérique est caractérisé par la matrice K :

$$K = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

Coordonnées en pixels	Coordonnées homogènes normalisées	Coordonnées homogènes non normalisées	Coordonnées 3D « caméra »

$$\begin{pmatrix} x = x'/Z \\ y = y'/Z \end{pmatrix} \Longleftrightarrow \begin{pmatrix} x = x'/Z \\ y = y'/Z \\ 1 \end{pmatrix} \text{ Avec } \begin{pmatrix} x' = \alpha_x * X + x_0 * Z \\ y' = \alpha_y * Y + y_0 * Z \\ Z \end{pmatrix} = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \boxed{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}.$$

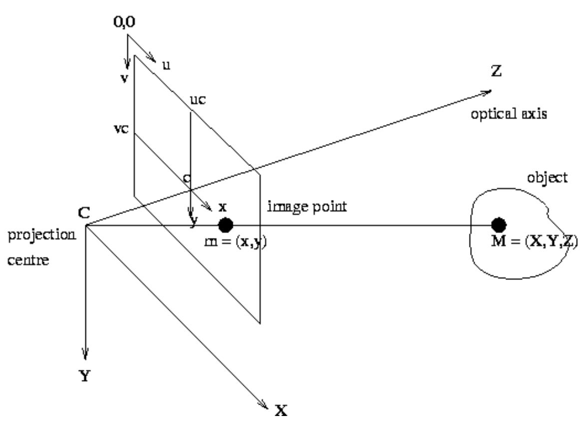
- K embarque les informations intrinsèques (propres) à la caméra :

  « matrice des paramètres intrinsèques » ou « matrice de calibration interne »

  ou « matrice de calibration intrinsèque » ou « matrice de calibration »

Le modèle du sténopé

- Remarque : le repère peuvent changer sans pour autant remettre en question la validité du modèle (i.e. formulation de la matrice K).
  - Par exemple, dans opencv, les coordonnées dans l'image « pixel » sont définies selon u (colonnes) et v (lignes), avec une origine au coin supérieure gauche
    - L'expression de K demeure inchangée, sous condition que les repères (caméra et plan image) soient adaptés en conséquence !!!



Le modèle du sténopé

$$K = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

- Estimation de la Matrice K : 2 possibilités
  - Connaissance des propriétés du système (spécifications du constructeur).
  - Estimation expérimentale (calibration).
- Méthode de calibration intuitive & simple :
  - Hypothèse simplificatrice : $\boxed{(x_0, y_0)}$ = centre de l'image numérique.
  - Estimation de $\boxed{(\alpha_x, \alpha_y)}$:
    - Considérer un objet de géométrie connue. Par exemple damier dont le nombre et dimensions des cases sont connus (mesures).
    - Effectuer une acquisition dans une configuration géométrique connue. Par exemple damier placé perpendiculairement à l'axe Z, à une distance connue (mesurée) de la caméra.
    - Mesurer les dimensions de l'objet dans l'image numérique (en pixels).

Matrice extrinsèque

Matrice extrinsèque

- On introduit un nouveau repère : le repère monde
  - Les coordonnées des objets du monde sont exprimées dans le repère monde qui n'est pas celui de la caméra.
    - e.g. coordonnées des cases du damier par rapport au damier complet
  - Pour déduire les coordonnées pixels :
    - repère monde (3D) → repère caméra (3D) → repère image (2D) → pixels (2D)

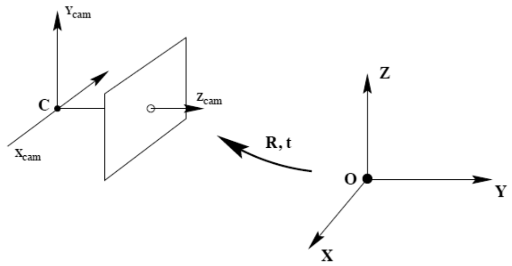


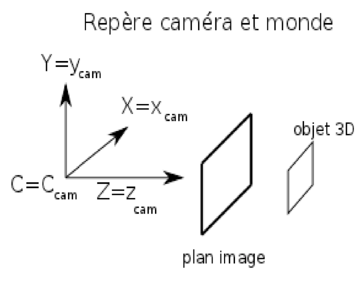
Fig. 6.3. *The Euclidean transformation between the world and camera coordinate frames.*

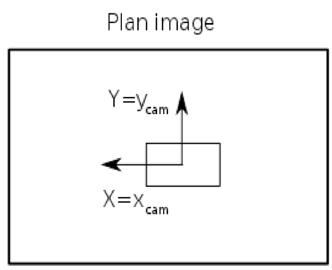
[source : Hartley and Zissermann, « Multiple View Geometry »]

Matrice extrinsèque

- Intérêt du changement de repère : exemple
  - On considère un objet 3D dont les coordonnées sont connues dans le repère monde (X;Y,Z).
  - Initialement : le repére monde de l'objet 3D est confondu avec le repère caméra

    (X,Y,Z) = (x_cam,y_cam,z_cam)





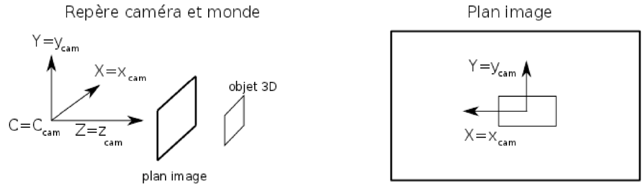
Matrice extrinsèque

- Intérêt du changement de repère : exemple
  - L'objet 3D est déplacé : on considère que ses coordonnées sont <u>inchangées</u> dans le repère monde (par rapport à l'état initial), mais que le repère monde s'est déplacé dans le repère caméra. Remarque : on peut aussi considérer que la caméra (i.e. le repère caméra) a été déplacée.
  - Pour pouvoir correctement afficher l'objet (virtuel) sur le plan image, il faut exprimer ses coordonnées dans le repère caméra (la matrice intrinsèque « prend en entrée » des coordonnées dans le repère caméra). Ceci revient à appliquer la transformation « repère monde vers repère caméra » à l'objet avant de le projeter sur l'image numérique.

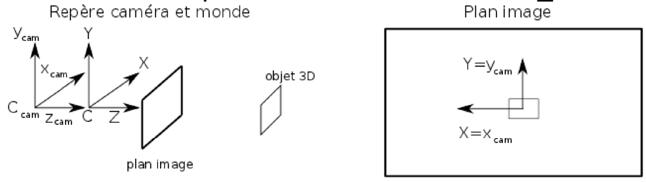
Matrice extrinsèque

- Intérêt du changement de repère : exemple
  - Nouvelle position : il faut appliquer une translation selon Z pour déduire z_cam (z_cam=Z+ distance « C_cam-C »)
    - Chacune des coordonnées « z » de l'objet va être « augmentée » avant projection et l'objet, « plus éloigné » de la caméra, apparaîtra plus « petit » dans l'image.

Position initiale



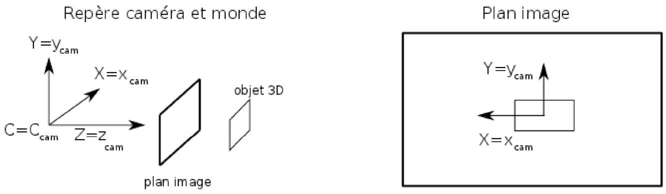
Nouvelle position : translation selon z_cam



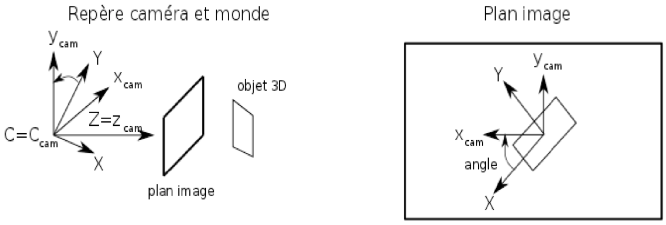
Matrice extrinsèque

- Intérêt du changement de repère : exemple
  - Nouvelle position : l'expression des coordonnées dans le repère caméra (celles dans le repère monde étant connues) revient à effectuer une rotation d'un certain angle autour de l'axe « Z=z_cam » (z reste inchangé).

Position initiale



Nouvelle position : rotation autour de Z



Matrice extrinsèque

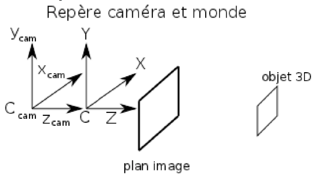
- Changement de repère : entre le repère monde 3D et le repère caméra
  - Pour déduire les coordonnées pixels :
    - repère monde (3D) → repère caméra (3D) → repère image (2D) → pixels (2D)
  - Il s'agit d'un transformation rigide (rotation + translation) :
    - M_cam= Rotation x M_monde + Translation → $\mathbf{M_{cam}} = \mathbf{RM} + \mathbf{T}$
      - R: matrice rotation de taille 3x3
      - T : vecteur de taille 3

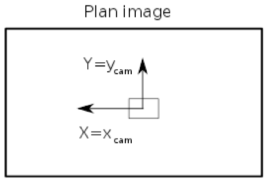
$$\mathbf{M}_{\mathrm{cam}} = \mathbf{RM} + \mathbf{T} \quad \rightarrow \quad \begin{pmatrix} X_{\mathrm{cam}} \\ Y_{\mathrm{cam}} \\ Z_{\mathrm{cam}} \end{pmatrix} = \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{T}$$

[source : Gilles Simon, « Vision par ordinateur »]

Matrice extrinsèque

- Changement de repère : entre le repère monde 3D et le repère caméra
  - Exemple :





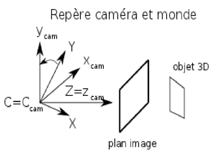
$$\mathbf{M}_{cam} = \mathbf{R}\mathbf{M} + \mathbf{T} \rightarrow \begin{pmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{pmatrix} = \mathbf{R}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{T}$$

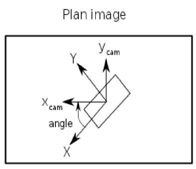
$$x_{cam} = X$$
$$y_{cam} = Y$$
$$z_{cam} = Z + \|C_{cam}C\|$$

$$\begin{pmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \|C_{cam}C\| \end{pmatrix}$$

Matrice extrinsèque

- Changement de repère : entre le repère monde 3D et le repère caméra
  - Exemple :





$$\mathbf{M}_{cam} = \mathbf{R}\mathbf{M} + \mathbf{T} \rightarrow \begin{pmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{pmatrix} = \mathbf{R}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{T}$$

$$x_{cam} = \cos(\alpha)X - \sin(\alpha)Y$$
$$y_{cam} = \sin(\alpha)X + \cos(\alpha)Y$$
$$z_{cam} = Z$$

$$\begin{pmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Astuce pratique:
On peut retrouver cette expression en exprimant (x_cam,y_cam) pour (X,Y)=(1,0) puis (X,Y)=(0,1),
et en utilisant les fonctions trigonométriques cos et sin
Remarque concernant l'angle : angle est ici « négatif » par rapport au sens trigonométrique (anti-horaire)
Si l'on passe dans le sens trogonométrique, sin → -sin
Remarque : pour une rotation autour des axes « y » ou « z » voir annexe

Matrice extrinsèque

- Changement de repère : entre le repère monde 3D et le repère caméra
  - Coordonnées homogènes : permet d'intégrer la translation dans le produit matriciel

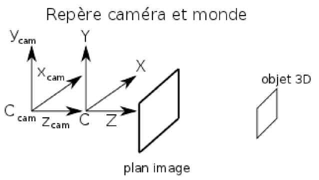
$$\mathbf{M}_{\text{cam}} = \mathbf{RM} + \mathbf{T} \rightarrow \begin{pmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \end{pmatrix} = \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{T} \quad \Longleftrightarrow \quad \begin{pmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \end{pmatrix} = (\mathbf{R} \mid \mathbf{T}) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

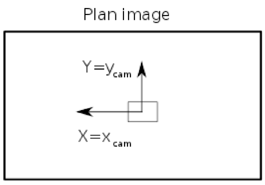
avec $$(\mathbf{R} \mid \mathbf{T}) = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{13} & \mathbf{R}_{13} & \mathbf{T}_1 \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{R}_{23} & \mathbf{T}_2 \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} & \mathbf{T}_3 \end{pmatrix}$$

[source : Gilles Simon, « Vision par ordinateur »]

Matrice extrinsèque

- Changement de repère : entre le repère monde 3D et le repère caméra
  - Coordonnées homogènes : permet d'intégrer la translation dans le produit matriciel
  - Exemple 1 :





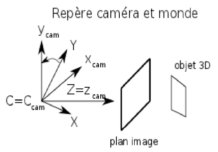
$$\begin{pmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \|C_{cam}C\| \end{pmatrix}$$

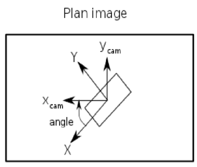
$$\Rightarrow \begin{pmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \|C_{cam}C\| \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

[source : Gilles Simon, « Vision par ordinateur »]

Matrice extrinsèque

- Changement de repère : entre le repère monde 3D et le repère caméra
  - Coordonnées homogènes : permet d'intégrer la translation dans le produit matriciel
  - Exemple 2 :





$$\begin{pmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$\begin{pmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

[source : Gilles Simon, « Vision par ordinateur »]

Matrice de projection

Matrice de projection

- Matrice de projection : matrice intrinsèque + matrice extrinsèque
- « Vue d'ensemble » de la projection « monde 3D » → « pixels 2D)

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \underbrace{\begin{pmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Matrice intrinsèque}} \underbrace{\begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{13} & \mathbf{R}_{13} & \mathbf{T}_1 \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{R}_{23} & \mathbf{T}_2 \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} & \mathbf{T}_3 \end{pmatrix}}_{\text{Matrice extrinsèque}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

= matrice P = K*[R|T]

[source : Gilles Simon, « Vision par ordinateur »]

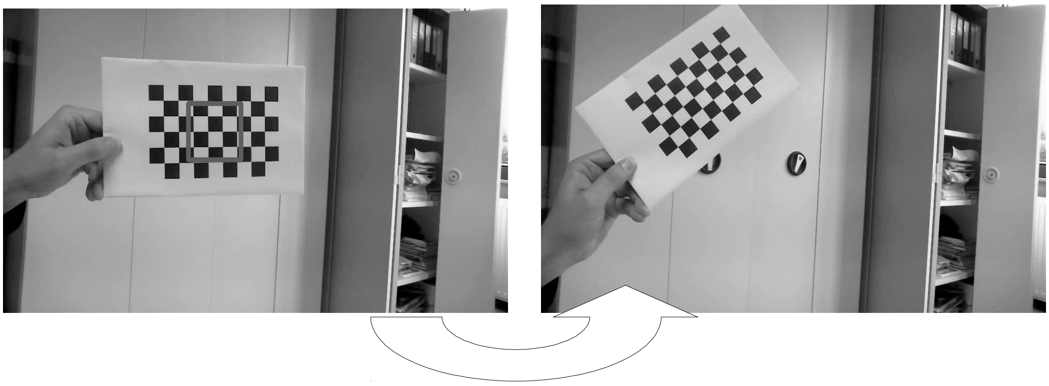
Recalage

Recalage

- Rappel du problème

Image de référence et objet virtuel (configuration acquisition connue donc position 3D connue)

Image courante (configuration acquisition inconnue)



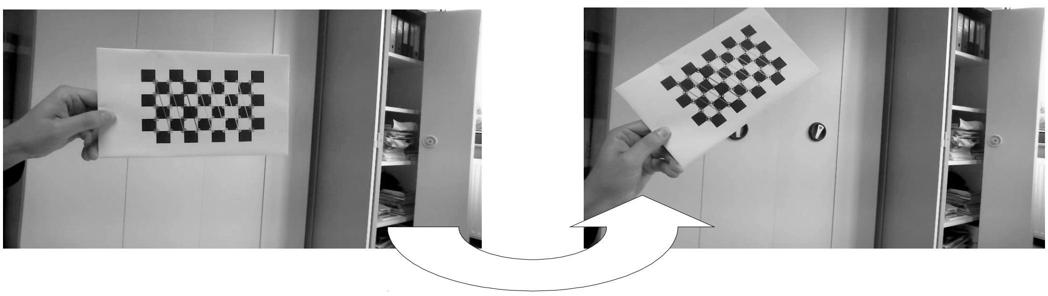
Problème : Où replacer l'objet virtuel ? Quelle pose?

Recalage

- Placement de l'objet virtuel : principe

Etape 1
Détection de marqueurs 2D
(coins des cases du damier)
coordonnées 3D connues
(configuration acquisition connue)

Etape 2
Détection de marqueurs 2D
(coins des cases du damier)



Etape 3 : estimation de la projection entre les coordonnes 3D (étape 1) et les coordonnées 2D détectées automatiquement (étape 2) → matrice extrinsèque (la matrice intrinsèque K étant connue)

<u>Hypothèse :</u>
Pour chaque marqueur de l'étape 1, on peut identifier automatiquement le marqueur correspondant dans l'image de l'étape 2 (la géométrie damier étant connue)

Recalage

- Placement de l'objet virtuel : principe

Etape 4 : Application de la projection estimée à l'objet virtuel (3D !!!!) et affichage dans l'image Courante

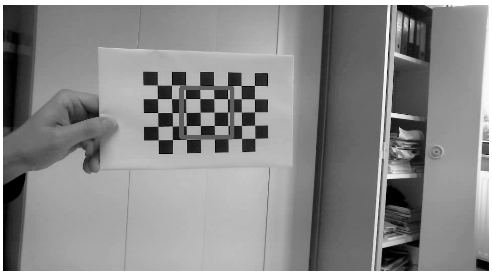


Image de référence

Image courante

Matrice de projection & Recalage

- En résumé :
  - Rappel formalisme :

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{13} & \mathbf{R}_{13} & \mathbf{T}_1 \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{R}_{23} & \mathbf{T}_2 \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} & \mathbf{T}_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

  - Matrice intrinsèque connue (estimée une seule fois = calibration)
  - Détermination de la matrice extrinsèque → Matrice de projection complète
    - On dispose d'une position de référence 3D de « marqueurs » (image de référence)
    - On détecte les marqueurs 2D dans une nouvelle image
    - On estime la transformation des marqueurs 3D → 2D : matrice de extrinsèque
  - Estimation 3D → 2D : Problème « Perspective – n - Point » (PnP)
    - Problème d'estimation des paramètres (coefficients de la matrice), connaissant plusieurs couples « entrées 3D → sorties 2D » (marqueurs mis en correspondance)
    - 6 paramètres : 3 angles de rotation + translation 3D (3 paramètres)
    - En pratique, on cherche les coefficients minimisant l'écart entre la projection 2D attendue (marqueurs détectés dans l'image courante) et celle obtenue :

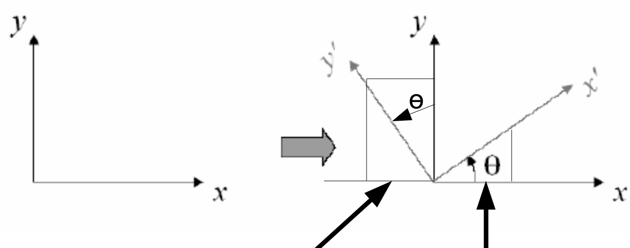
$$\min_{\mathrm{R,T}} \sum_i \left\| \mathrm{K}[\mathrm{R} \mid \mathrm{T}]\mathrm{M}_i - \mathrm{m}_i \right\|^2$$

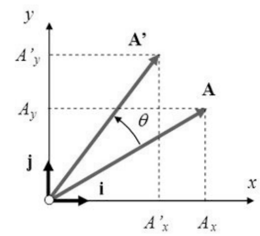
Annexes

Annexe : rotations

- Rotation autour de l'axe des « z » : x → x' ; y → y'

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$





A:(Ax,Ay) → A' :(A'x,A'y)

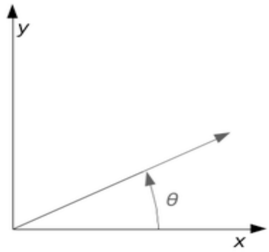
Si (x,y)=(0,y) → x'=-sin(ϴ)*y
La coordonnée en x (notée x') passe de 0 à une valeur négative fonction de y (-sin(ϴ)y).

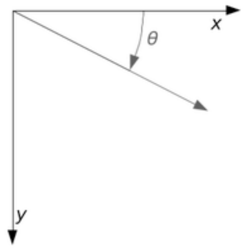
Si (x,y)=(x,0) → x'=cos(ϴ)*x
La coordonnée en x (notée x') est réduite de x à cos(ϴ)x.

→ On déduit que (x,y) → x'=cos(ϴ)x-sin(ϴ)y
→ Par un raisonnement analogue, on déduit que (x,y) → y'=sin(ϴ)x+cos(ϴ)y

Annexe : rotations

- Rotation autour de l'axe des « z » : x → x' ; y → y'
  - Cas plan orienté différemment (rotation horaire et non plus anti-horaire)



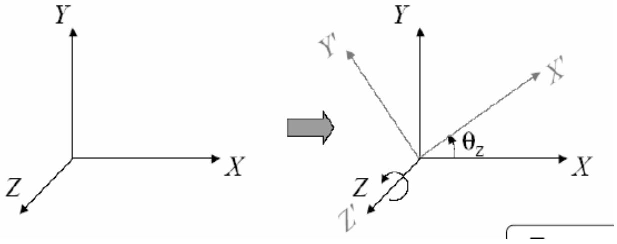


$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \qquad R(-\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

Annexe : rotations

- Rotation autour de l'axe des « z » : x → x' ; y → y'
  - Vue 3D

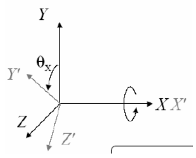
$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} \cos\theta_Z & -\sin\theta_Z & 0 \\ \sin\theta_Z & \cos\theta_Z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$



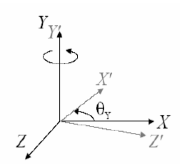
Annexe : rotations

- Rotation autour des « x », « y » et « z » (coordonnées homogènes)

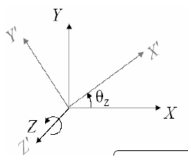
Rotation d'angle α autour de $\vec{i}$: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$



Rotation d'angle β autour de $\vec{j}$: $\begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

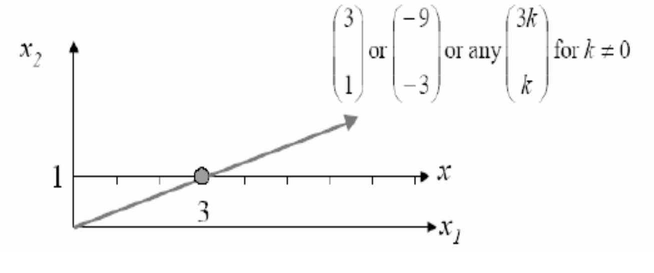


Rotation d'angle γ autour de $\vec{k}$: $\begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$



Annexes : coordonnées homogènes

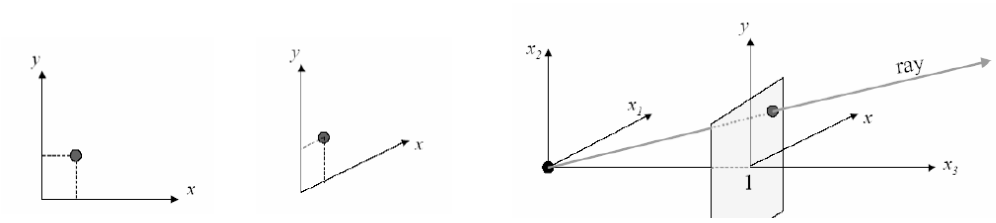
- Cas 1D :
  - Normalement, dans un système non-homogène de coordonnées, pour le point P une seule dim. : x=3.
  - Dans un système homogène, un point 1D est représenté par un vecteur de dimension 2



[source : CLADY http://www.isir.upmc.fr/]
[Institut des Systèmes Intelligents et de robotique]

Annexes : coordonnées homogènes

- Cas 2D → 3D
  - Dans un système non-homogène de coordonnées, le vecteur coord. du point P a 2 dimensions.
  - Dans un système homogène, le vecteur coord. Du point P a 3 dimensions.



[source : CLADY http://www.isir.upmc.fr/]
[Institut des Systèmes Intelligents et de robotique]