Introduction à la géométrie projective & Modélisation de caméras

Ecole Doctorale MIIS - 2017

Sebastien Kramm

LITIS Rouen

28 juin 2017

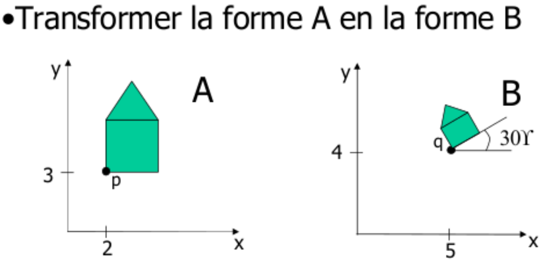
Introduction : paradoxe du parallélisme

Sommaire

Sous-sommaire

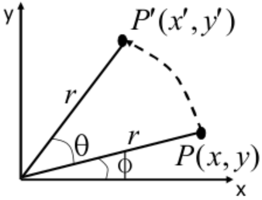
Transformation 2D en infographie

- En infographie, une transformation 2D s'applique à un ensemble de points
  - Translation
  - Rotation
  - Changement d'échelle (*scaling*)
- Par exemple :



Decompositions des transformations - 1

- Rotation

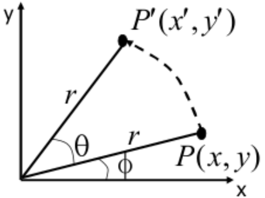


$$\begin{cases} x' = x\cos\theta - y\sin\theta \\ y' = x\sin\theta + y\cos\theta \end{cases}$$

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

Decompositions des transformations - 1

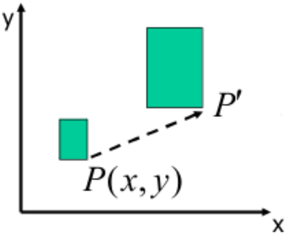
- Rotation



$$\begin{cases} x' = x\cos\theta - y\sin\theta \\ y' = x\sin\theta + y\cos\theta \end{cases}$$

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

- Multiplication par un facteur d'échelle

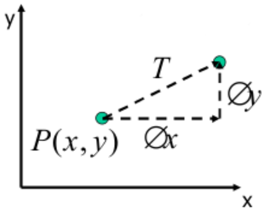


$$\begin{cases} x' = k_x \cdot x \\ y' = k_y \cdot y \end{cases}$$

$$\Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Decompositions des transformations - 2

- Translation

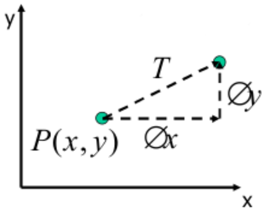


$$\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases}$$

$$\Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Decompositions des transformations - 2

- Translation



$$\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases}$$

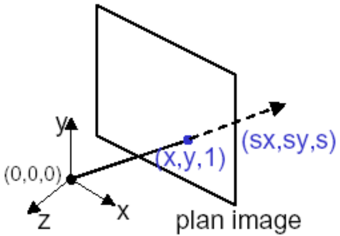
$$\Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

- Inconvénient de cette notation
  - Rotation et *scaling* : produit matriciel
  - Translation : addition matricielle
- Solution : remplacer les **coordonnées euclidiennes** par les **coordonnées homogènes**.

Sous-sommaire

Coordonnées homogènes

- Les coordonnées homogènes permettent d'avoir un formalisme mathématique homogène.
- Toute combinaison de transformation $[A]$,$[B]$,$[C]$ pourra se mettre sous la forme $[T] = [A].[B].[C]$, $\forall$ le type de la tranformation.
- Permettent de représenter un point à l'infini.
- Permettent de formaliser la projection 3D $\Rightarrow$ 2D.
- Utilisables en 2D ou 3D.
- Très utilisées en infographie (OpenGL/Direct3D).



Point 2D en coordonnées homogènes

- Un point 2D (u, v) est représenté par un vecteur 3×1
  $\mathbf{m} = (u, v, 1)^T = (su, sv, s)^T = (m_1, m_2, m_3)^T$, avec $s \neq 0$

Point 2D en coordonnées homogènes

- Un point 2D (u, v) est représenté par un vecteur 3×1 $\mathbf{m} = (u, v, 1)^T = (su, sv, s)^T = (m_1, m_2, m_3)^T$, avec $s \neq 0$
- Seul le **ratio** des coefficients est significatif : les points $(3, 2, 1)$ et $(6, 4, 2)$ sont les mêmes.

Point 2D en coordonnées homogènes

- Un point 2D (u, v) est représenté par un vecteur 3×1 $\mathbf{m} = (u, v, 1)^T = (su, sv, s)^T = (m_1, m_2, m_3)^T$, avec $s \neq 0$
- Seul le **ratio** des coefficients est significatif : les points $(3, 2, 1)$ et $(6, 4, 2)$ sont les mêmes.
- On obtient les coordonnées euclidiennes par division : $u = m_1/m_3 \quad v = m_2/m_3$

Point 2D en coordonnées homogènes

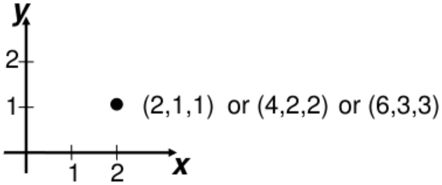
- Un point 2D (u, v) est représenté par un vecteur 3×1
  $\mathbf{m} = (u, v, 1)^T = (su, sv, s)^T = (m_1, m_2, m_3)^T$, avec $s \neq 0$
- Seul le **ratio** des coefficients est significatif : les points $(3, 2, 1)$ et $(6, 4, 2)$ sont les mêmes.
- On obtient les coordonnées euclidiennes par division :
  $u = m_1/m_3 \quad v = m_2/m_3$
- Un point à l'infini sera représenté par $(u, v, 0)$
  $\rightarrow$ coord. euclidiennes $= (u/0, v/0)$

Point 2D en coordonnées homogènes

- Un point 2D (u, v) est représenté par un vecteur 3×1
  $\mathbf{m} = (u, v, 1)^T = (su, sv, s)^T = (m_1, m_2, m_3)^T$, avec $s \neq 0$
- Seul le **ratio** des coefficients est significatif : les points $(3, 2, 1)$ et $(6, 4, 2)$ sont les mêmes.
- On obtient les coordonnées euclidiennes par division :
  $u = m_1/m_3 \quad v = m_2/m_3$
- Un point à l'infini sera représenté par $(u, v, 0)$
  $\rightarrow$ coord. euclidiennes $= (u/0, v/0)$
- Symbole $\simeq$: "congruent à" : égalité à un facteur d'échelle près. (souvent remplacé abusivement par '=')

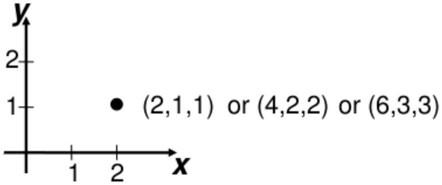
Interprétation graphique

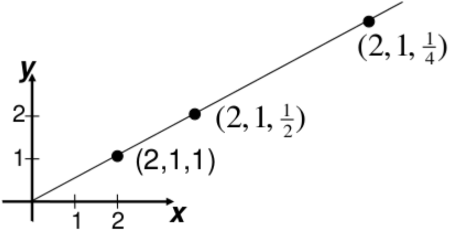
$$\mathbf{m} = (m_1, m_2, m_3)$$



Interprétation graphique

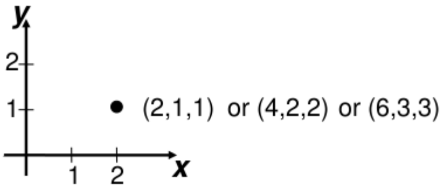
$$\mathbf{m} = (m_1, m_2, m_3)$$

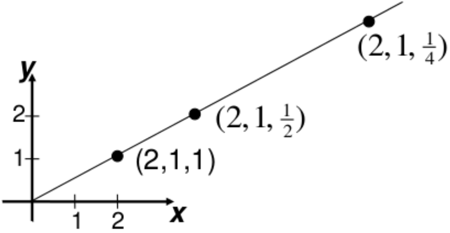




Interprétation graphique

$$\mathbf{m} = (m_1, m_2, m_3)$$





- Le ratio m_1/m_2 indique une **direction**.
- m_3 indique la **distance** à l'origine.

Droite 2D en coordonnées homogènes

- Une droite est classiquement modélisée par $y = ax + b$

Droite 2D en coordonnées homogènes

- Une droite est classiquement modélisée par $y = ax + b$
- Problème : une droite verticale implique $a = \infty \Rightarrow$ difficilement implémentable...

Droite 2D en coordonnées homogènes

- Une droite est classiquement modélisée par $y = ax + b$
- Problème : une droite verticale implique $a = \infty \Rightarrow$ difficilement implémentable...
- Solution : droite en coordonnées homogènes : $ax + by + c = 0$
  $\rightarrow$ Une droite est représentée par un vecteur 3x1 : $l = (a, b, c)^T$

Droite 2D en coordonnées homogènes

- Une droite est classiquement modélisée par $y = ax + b$
- Problème : une droite verticale implique $a = \infty \Rightarrow$ difficilement implémentable...
- Solution : droite en coordonnées homogènes : $ax + by + c = 0$
  $\rightarrow$ Une droite est représentée par un vecteur 3x1 : $l = (a, b, c)^T$
  - droite horizontale : $(0, 1, c)^T \Rightarrow y = -c$

Droite 2D en coordonnées homogènes

- Une droite est classiquement modélisée par $y = ax + b$
- Problème : une droite verticale implique $a = \infty \Rightarrow$ difficilement implémentable...
- Solution : droite en coordonnées homogènes : $ax + by + c = 0$
  $\rightarrow$ Une droite est représentée par un vecteur 3x1 : $l = (a, b, c)^T$
  - droite horizontale : $(0, 1, c)^T \Rightarrow y = -c$
  - droite verticale : $(1, 0, c)^T \Rightarrow x = -c$

Droite 2D en coordonnées homogènes

- Une droite est classiquement modélisée par $y = ax + b$
- Problème : une droite verticale implique $a = \infty \Rightarrow$ difficilement implémentable...
- Solution : droite en coordonnées homogènes : $ax + by + c = 0$
  $\rightarrow$ Une droite est représentée par un vecteur 3x1 : $l = (a, b, c)^T$
  - droite horizontale : $(0, 1, c)^T \Rightarrow y = -c$
  - droite verticale : $(1, 0, c)^T \Rightarrow x = -c$
- Comme pour les points, seul le **ratio** est significatif.

Normalisation

- Lors du stockage, il est d'usage de **normaliser** les grandeurs homogènes
  - points : avec $p = (su, sv, s)^T$, on divise tous les membres par s, pour arriver à $p = (u, v, 1)^T$.
  - droites : soit une droite $l = (a, b, c)$, on normalise pour avoir $a^2 + b^2 = 1$.
  - matrices : on normalise par l'un des éléments (h_{33} par exemple).
- Remarque : on ne peut normaliser que des éléments "réels" ($\neq \infty$)

Dualité points - droites

- Deux points $\mathbf{p_1}$ et $\mathbf{p_2}$ définissent une droite unique $\mathbf{l}$

Dualité points - droites

- Deux points $\mathbf{p_1}$ et $\mathbf{p_2}$ définissent une droite unique $\mathbf{l}$

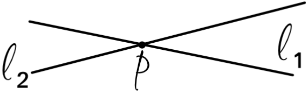
$$\Leftrightarrow \quad \mathbf{p_1} \times \mathbf{p_2} = \mathbf{l}$$

Dualité points - droites

- Deux points $\mathbf{p_1}$ et $\mathbf{p_2}$ définissent une droite unique $\mathbf{l}$

$$\Leftrightarrow \quad \mathbf{p_1} \times \mathbf{p_2} = \mathbf{l}$$

- L'intersection de deux droites $\mathbf{l_1}$ et $\mathbf{l_2}$ définit un point unique $\mathbf{p}$

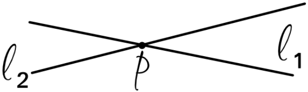


Dualité points - droites

- Deux points $\mathbf{p_1}$ et $\mathbf{p_2}$ définissent une droite unique $\mathbf{l}$

$$\Leftrightarrow \quad \mathbf{p_1} \times \mathbf{p_2} = \mathbf{l}$$

- L'intersection de deux droites $\mathbf{l_1}$ et $\mathbf{l_2}$ définit un point unique $\mathbf{p}$



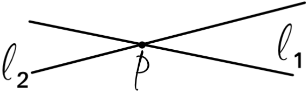
$$\Leftrightarrow \quad \mathbf{l_1} \times \mathbf{l_2} = \mathbf{p}$$

Dualité points - droites

- Deux points $\mathbf{p_1}$ et $\mathbf{p_2}$ définissent une droite unique $\mathbf{l}$

$$\Leftrightarrow \quad \mathbf{p_1} \times \mathbf{p_2} = \mathbf{l}$$

- L'intersection de deux droites $\mathbf{l_1}$ et $\mathbf{l_2}$ définit un point unique $\mathbf{p}$



$$\Leftrightarrow \quad \mathbf{l_1} \times \mathbf{l_2} = \mathbf{p}$$

- Un point est sur une ligne SSI :

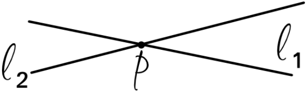
$$\mathbf{l} \cdot \mathbf{p} = 0 \quad \text{ou} \quad \mathbf{l}^T \cdot \mathbf{p} = 0 \quad \text{ou} \quad \mathbf{p}^T \cdot \mathbf{l} = 0$$

Dualité points - droites

- Deux points $\mathbf{p_1}$ et $\mathbf{p_2}$ définissent une droite unique $\mathbf{l}$

$$\Leftrightarrow \quad \mathbf{p_1} \times \mathbf{p_2} = \mathbf{l}$$

- L'intersection de deux droites $\mathbf{l_1}$ et $\mathbf{l_2}$ définit un point unique $\mathbf{p}$



$$\Leftrightarrow \quad \mathbf{l_1} \times \mathbf{l_2} = \mathbf{p}$$

- Un point est sur une ligne SSI :

$$\mathbf{l} \cdot \mathbf{p} = 0 \quad \text{ou} \quad \mathbf{l}^T \cdot \mathbf{p} = 0 \quad \text{ou} \quad \mathbf{p}^T \cdot \mathbf{l} = 0$$

Attention : Soit un point $\mathbf{p}$ et une droite $\mathbf{l}$. La valeur $d = \mathbf{l} \cdot \mathbf{p}$ n'est **pas** la distance entre le point et la droite !

Rappel : produit vectoriel

- Noté $\times$ dans le monde entier (et $\wedge$ en France...)
- Définition pour un vecteur 3x1 :

$$u \times v = \begin{pmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{pmatrix}$$

- Peut-être transformé en un produit matriciel en utilisant la notion de "**Matrice antisymétrique**", notée $[\mathbf{a}]_\times$

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \, . \, \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} . \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Sommaire

Matrices 2x2 ⇒ matrices 3x3

- Si les points sont en coordonnées homogènes, alors on pourra écrire toutes les transformations sous formes de matrices 3x3 :

Matrices 2x2 ⇒ matrices 3x3

- Si les points sont en coordonnées homogènes, alors on pourra écrire toutes les transformations sous formes de matrices 3x3 :
- Translation

$$\begin{cases} s\,x' = 1.x + 0.y + t_x \\ s\,y' = 0.x + 1.y + t_y \\ s = 0.x + 0.y + 1 \end{cases} \Rightarrow \quad \mathbf{T} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Matrices 2x2 $\Rightarrow$ matrices 3x3

- Rotation

$$\begin{cases} s\,x' = \cos\theta.x - \sin\theta.y + 0 \\ s\,y' = \sin\theta.x + \cos\theta.y + 0 \\ \quad s = 0.x + 0.y + 1 \end{cases} \quad \Rightarrow \quad \mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrices 2x2 $\Rightarrow$ matrices 3x3

- Rotation

$$\begin{cases} s\,x' = \cos\theta.x - \sin\theta.y + 0 \\ s\,y' = \sin\theta.x + \cos\theta.y + 0 \\ s = 0.x + 0.y + 1 \end{cases} \Rightarrow \quad \mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Scaling

$$\begin{cases} s\,x' = \alpha_x.x & +0.y + 0 \\ s\,y' = 0.x & +\alpha_y.y + 0 \\ s = 0.x & +0.y + 1 \end{cases} \Rightarrow \quad \mathbf{S} = \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Combinaison de transformation

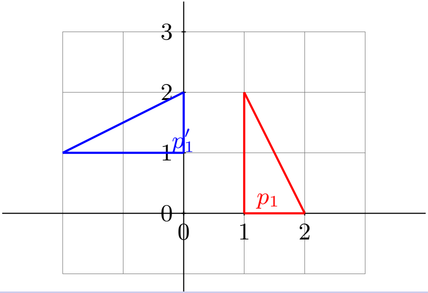
- On pourra construire une transformation composée par produit des différentes matrices de transformation.
- Exemple : pour appliquer une transformation $\mathbf{H}_1$ puis une transformation $\mathbf{H}_2$ à un ensemble de points p_i, on pourra faire :
  $p'_i = \mathbf{H}_1 \cdot p_i$, puis $p''_i = \mathbf{H}_2 \cdot p'_i$ (2 multiplications pour chaque point)
  Ou, plus simplement :
  $p''_i = \mathbf{H}_{12} \cdot p_i$ avec $\mathbf{H}_{12} = \mathbf{H}_2 \cdot \mathbf{H}_1$
- Attention à l'ordre : $\mathbf{A} \cdot \mathbf{B} \neq \mathbf{B} \cdot \mathbf{A}$

Combinaison de transformation : rotation

- Exemple : on souhaite appliquer une rotation de 90 ˚à une forme géométrique.
- Si on applique directement la matrice :

$$\mathbf{R} = \begin{bmatrix} \cos 90 & -\sin 90 & 0 \\ \sin 90 & \cos 90 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Alors la rotation s'applique par rapport à l'origine, et on obtiendra cette transformation :



$$p'_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Combinaison de transformation : rotation

- Pour obtenir une rotation d'un angle θ au point (u, v), il faudra :
  1. Appliquer une translation en (u, v)
  2. Appliquer une rotation de θ degrés
  3. Appliquer une translation de $(-u, -v)$

Combinaison de transformation : rotation

- Pour obtenir une rotation d'un angle θ au point (u, v), il faudra :
  1. Appliquer une translation en (u, v)
  2. Appliquer une rotation de θ degrés
  3. Appliquer une translation de $(-u, -v)$
- Mais on pourra aussi appliquer la matrice $\mathbf{H} = \mathbf{T}_1.\mathbf{R}.\mathbf{T}_2$, avec :

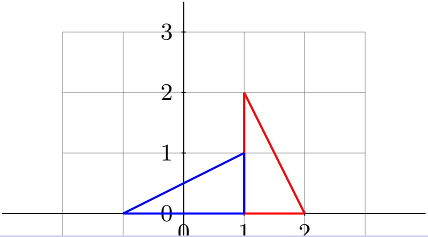
$$\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & u \\ 0 & 1 & v \\ 0 & 0 & 1 \end{bmatrix}$$

Translation positive

$$\mathbf{T}_2 = \begin{bmatrix} 1 & 0 & -u \\ 0 & 1 & -v \\ 0 & 0 & 1 \end{bmatrix}$$

Translation négative

Ceci donnera :



Sous-sommaire

Généralisation : notion d'homographie

$$\mathbf{m}_2 = \mathbf{H}.\mathbf{m}_1 \qquad \mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

- Formellement : "Application projective bijective"
  $\Rightarrow$ transformation 2D (ou 3D).
  - *Computer Graphics* (Infographie) : appliquée à un ensemble de points représentant un "modèle".
  - *Computer Vision* : appliquée à l'ensemble des pixels d'une image.
- Matrice homogène $\Rightarrow$ 8 degrés de liberté
  (on peut poser $h_{33} = 1$).

Propriétés

La multiplication de matrices n'est pas commutative

$\mathbf{H}_1 = \mathbf{H}_a \mathbf{H}_b$ est différent de $\mathbf{H}_2 = \mathbf{H}_b \mathbf{H}_a$

Propriétés

La multiplication de matrices n'est pas commutative

$\mathbf{H}_1 = \mathbf{H}_a \mathbf{H}_b$ est différent de $\mathbf{H}_2 = \mathbf{H}_b \mathbf{H}_a$

La transformation est inversible

Si $\mathbf{m}_2 = \mathbf{H}\ \mathbf{m}_1$, alors $\mathbf{m}_1 = \mathbf{H}^{-1}\mathbf{m}_2$

Propriétés

La multiplication de matrices n'est pas commutative

$\mathbf{H}_1 = \mathbf{H}_a\mathbf{H}_b$ est différent de $\mathbf{H}_2 = \mathbf{H}_b\mathbf{H}_a$

La transformation est inversible

Si $\mathbf{m}_2 = \mathbf{H}\ \mathbf{m}_1$, alors $\mathbf{m}_1 = \mathbf{H}^{-1}\mathbf{m}_2$

Transformation de droites

Si un point $\mathbf{m}$ est sur une droite $\mathbf{d}$ (soit $\mathbf{d}^T\ \mathbf{m} = 0$), alors le point $\mathbf{m}' = \mathbf{H}\ \mathbf{m}$ sera sur la droite $\mathbf{d}' = \mathbf{H}^{-T}\ \mathbf{d}$

Homographie : interprétation algébrique

- Le point $\mathbf{m} = (s\,u, s\,v, s)^T$ est projeté vers le point $\mathbf{m}' = (s\,u', s\,v', s)^T$

$$\begin{cases} s\,u' = h_{11}u + h_{12}v + h_{13} \\ s\,v' = h_{21}u + h_{22}v + h_{23} \\ s = h_{31}u + h_{32}v + h_{33} \end{cases}$$

$$\Rightarrow u' = \frac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + h_{33}} \quad v' = \frac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + h_{33}}$$

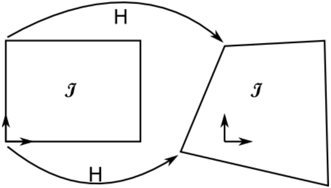
Homographie : interprétation algébrique

- Le point $\mathbf{m} = (s\,u, s\,v, s)^T$ est projeté vers le point $\mathbf{m}' = (s\,u', s\,v', s)^T$

$$\begin{cases} s\,u' = h_{11}u + h_{12}v + h_{13} \\ s\,v' = h_{21}u + h_{22}v + h_{23} \\ s = h_{31}u + h_{32}v + h_{33} \end{cases}$$

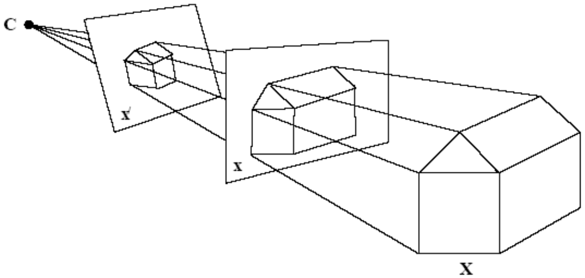
$$\Rightarrow u' = \frac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + h_{33}} \quad v' = \frac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + h_{33}}$$

- Projection de type **perspective** (non linéaire) : l'image n'est plus rectangulaire :



Homographie : interprétation optique

- Intersection sur un plan du faisceau optique générant l'image.



- Problème : difficile de faire un lien entre l'orientation du plan et le résultat.

Utilisation en vision par ordinateur

- Dans ce contexte, on traite tous les pixels d'une image
- Applications : suppression de perspective, rectification pour la stéréovision, etc.

Exemples

Source :

Rotation :

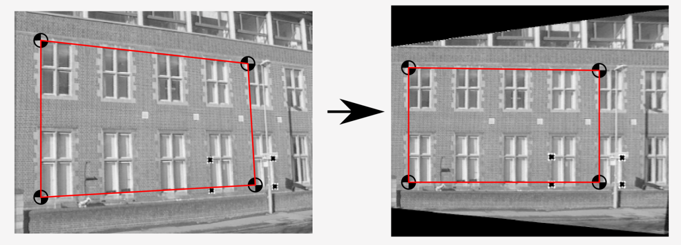
Transformation affine :

Transformation perspective :

Sous-sommaire

Détermination à partir de points 2D

- Dans certaines situation, on ne connait pas la transformation sous forme paramétrique, mais sous forme de "mapping" de points.

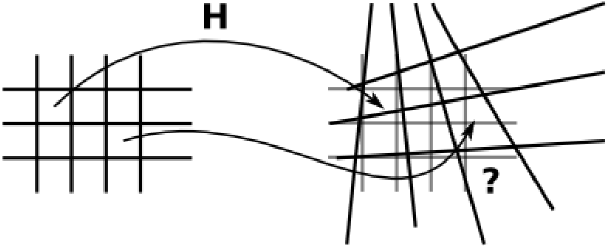


- Deux jeux de quatre points (non-colinéaires) définissent une homographie unique.
- Si plus de 4 points, il faudra utiliser un **estimateur robuste** (RanSaC ou autre) pour trouver une solution optimale.

Sous-sommaire

Implémentation pratique - 1

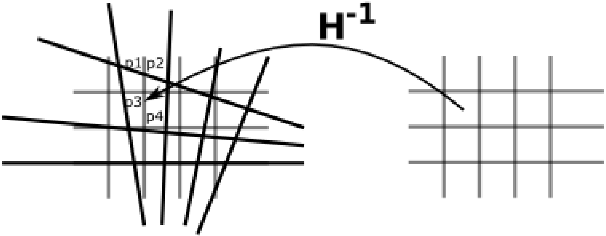
- L'application directe de la transformation à chaque pixel source peut amener des "trous" dans l'image destination.



- Question : quels pixels doit on remplir avec la valeur du pixel source parmi les 4 possibles ?

Implémentation pratique - 2

- On préfère énumérer les pixels destinations et les remplir par interpolation sur les pixels source :

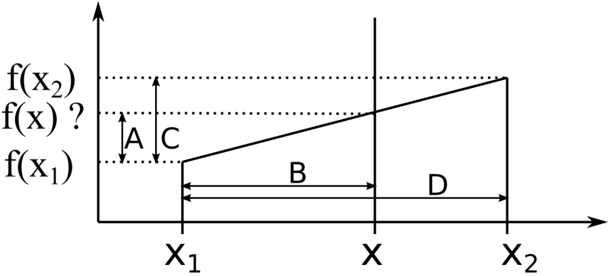


$\Rightarrow$ Pour chaque pixel destination, on va fixer son niveau de gris par interpolation entre les valeurs des 4 pixels source.

- En pratique, les bibliothèques de traitement d'image (OpenCv) fournissent des fonctions optimisées.

Interpolation : cas monodimensionnel

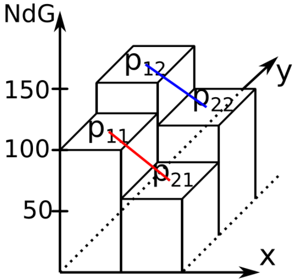
- Interpolation linéaire :
  déterminer l'expression de $f(x)$ à partir des valeurs (connues) de $x_1, x_2, f(x_1), f(x_2)$.

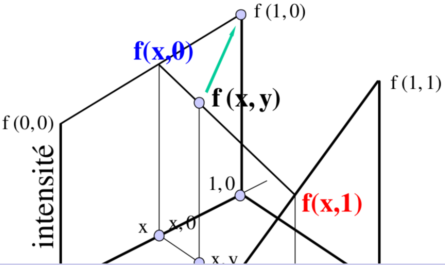


$$A/B = C/D$$
$$A = f(x) - f(x_1)$$
$$B = x - x_1$$
$$C = f(x_2) - f(x_1)$$
$$D = x_2 - x_1$$

$$\Rightarrow f(x) = \frac{\quad}{\quad} f(x_1) + \frac{\quad}{\quad} f(x_2)$$

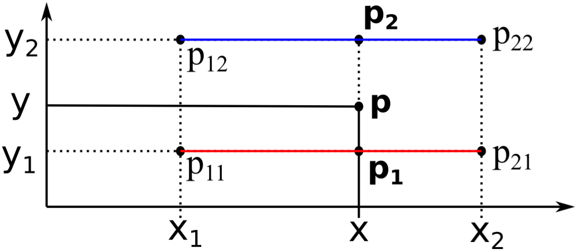
Interpolation bilinéaire





Interpolation bilinéaire

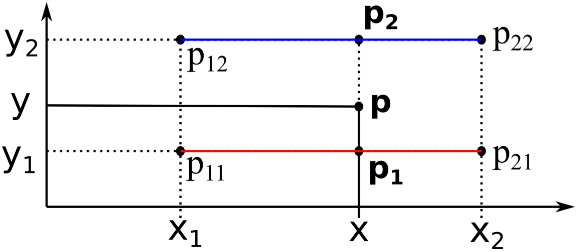
- Interpolation bilinéaire, cas bi-dimensionnel :
  - x_1, x_2, y_1, y_2 sont les coordonnées (entières) des pixels,
  - p_{ij} est la valeur du pixel x_i, y_j.
  - On cherche la valeur de p, de coordonnées (x, y).



- Deux étapes :
  - calcul sur l'axe Ox des points p_1 et p_2 par interpolation,
  - calcul de p par interpolation de p_1 et p_2.
- Ratio effet visuel / coût le plus élevé.

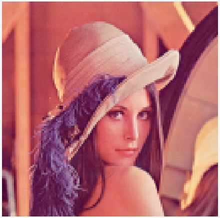
Interpolation bilinéaire

- Interpolation bilinéaire, cas bi-dimensionnel :
  - x_1, x_2, y_1, y_2 sont les coordonnées (entières) des pixels,
  - p_{ij} est la valeur du pixel x_i, y_j.
  - On cherche la valeur de p, de coordonnées (x, y).

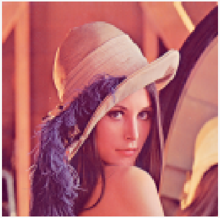


- Deux étapes :
  - calcul sur l'axe Ox des points p_1 et p_2 par interpolation,
  - calcul de p par interpolation de p_1 et p_2.
- Ratio effet visuel / coût le plus élevé.
- Exercice : exprimer p en considérant $x_1 = y_1 = 0$ et $x_2 = y_2 = 1$

Rotation avec interpolation au plus proche voisin



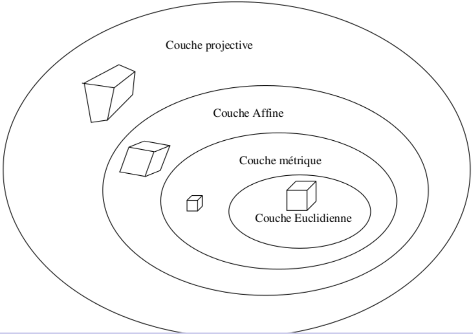
Rotation avec interpolation bilinéaire



Sommaire

Stratification des géométries

- Les différentes transformations possibles s'insèrent dans différents types de géométrie.
- Pour chaque couche on peut définir
  - une **forme canonique** de la matrice d'homographie.
  - des **invariants** : "concepts géométriques" non modifiés par une homographie appartenant à cette couche.
- Chaque couche englobe la précédente.



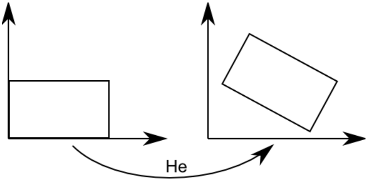
1 : Transformation euclidienne

- Le monde qui nous entoure est **euclidien** : les objets rigides peuvent subir uniquement des rotations et translations.
  $\Rightarrow$ on parle alors de géométrie euclidienne (notion de "transformation rigide").
- 3 DOF (*Degree Of Freedom*) : 2 translations + 1 angle de rotation

- En 2D, l'homographie associée sera de la forme :

$$\mathbf{H} = \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

(r_{ij} : coefficients d'une matrice de rotation)

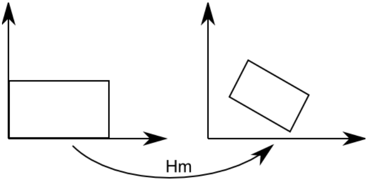


- Invariants : longueur des segments, surfaces, angles entre segments.

2 : Transformation métrique

- Permet l'ajout d'un facteur d'échelle k.
- 4 DOF : 2 translations + 1 rotation + 1 facteur d'échelle.
- En 2D, l'homographie associée sera de la forme :

$$\mathbf{H} = \begin{bmatrix} k.r_{11} & k.r_{12} & t_x \\ k.r_{21} & k.r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



- Invariants : rapport des longueurs, angles entre segments.

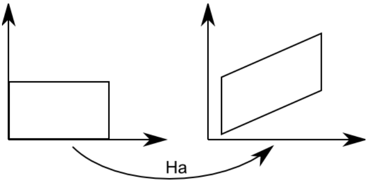
3 : Transformation affine

- Linéaire, préserve le parallélisme.
- 6 DOF

- En 2D, l'homographie associée sera de la forme

$$\mathbf{H} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

(avec a_{ij} quelconque)



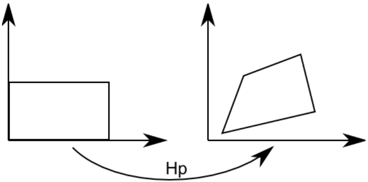
- Invariants : parallélisme, rapport des surfaces, . . .

4 : Transformation projective

- Cas le plus général, englobe toutes les autres géométries.
- 8 DOF

- En 2D, l'homographie associée sera de la forme :

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}$$

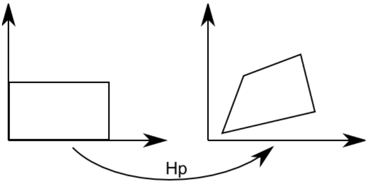


4 : Transformation projective

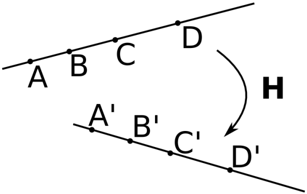
- Cas le plus général, englobe toutes les autres géométries.
- 8 DOF

- En 2D, l'homographie associée sera de la forme :

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}$$



- Invariants : colinéarité, birapport.

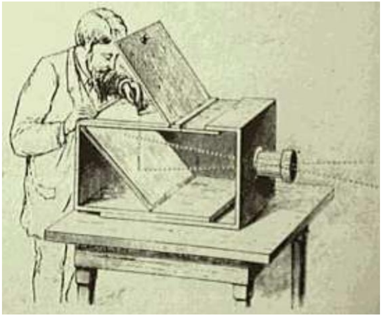


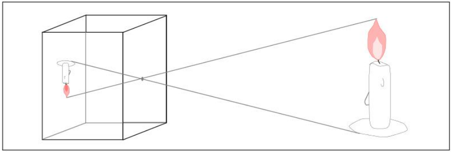
$$\frac{AC/BC}{AD/BD} = \frac{A'C'/B'C'}{A'D'/B'D'}$$

Sommaire

Modèle sténopé d'une caméra (*pin-hole*)

- Un modèle parmi d'autres.
- Le plus utilisé.
- Ce modèle réalise une **projection** sur un **plan** (2D) de points 3D.



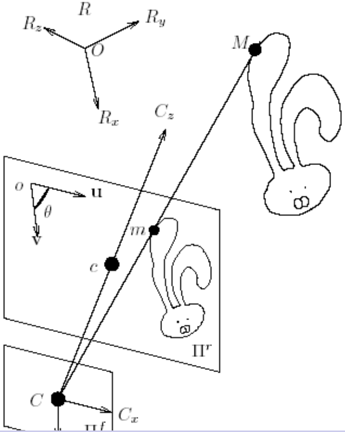


- Note : il est d'usage de considérer le plan image **devant** le foyer optique.

Projection d'un point dans une image

- 3 systèmes de coordonnées :
  - repère "monde" : X, Y, Z
  - repère caméra : x, y, z
  - repère image : u, v

- 1 point-image $\leftrightarrow$ infinité de points réels.
- 1 point-image est un **rayon** de l'espace (droite 3D).



Coordonnées homogènes 3D

- On ajoute un 4ème élement au point (vecteur 4x1) : $\mathbf{M} \simeq (X, Y, Z, 1)^T$
- La projection d'un point 3D $M = (X, Y, Z, 1)$ vers un point 2D $m = (u, v, 1)$ sur le plan image 2D se résume à l'opération matricielle :

$$\mathbf{m}_{3x1} \simeq \mathbf{P}_{3x4}.\mathbf{M}_{4x1}$$

- $\mathbf{P}$ s'appelle la **matrice de projection** (3 lignes x 4 colonnes).

Coordonnées homogènes 3D

- On ajoute un 4ème élement au point (vecteur 4x1) : $\mathbf{M} \simeq (X, Y, Z, 1)^T$
- La projection d'un point 3D $M = (X, Y, Z, 1)$ vers un point 2D $m = (u, v, 1)$ sur le plan image 2D se résume à l'opération matricielle :

$$\mathbf{m}_{3x1} \simeq \mathbf{P}_{3x4}.\mathbf{M}_{4x1}$$

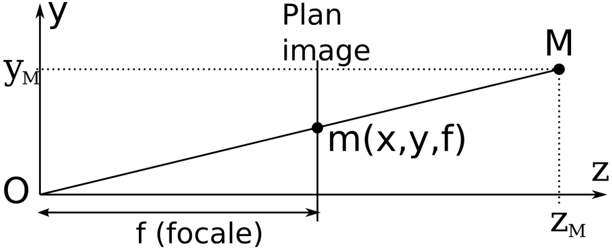
- $\mathbf{P}$ s'appelle la **matrice de projection** (3 lignes x 4 colonnes).

Attention !

Une droite 3D **n'est pas** représentée par un vecteur 4x1 !
$aX + bY + cZ + d = 0$: equation d'un plan 3D

Etapes de la projection

- Dans le repère "caméra", en considérant l'axe Oy :
  (raisonnement similaire pour l'axe Ox)



- Le point image m a pour coordonnées euclidiennes :

$$x = f\,\frac{x_M}{z_M} \qquad\qquad y = f\,\frac{y_M}{z_M}$$

Etapes de la projection

- Afin d'obtenir l'expression matricielle, on écrit :

$$\begin{cases} s\,x = f.x_M + 0.y_M + 0.z_M \\ s\,y = 0.x_M + f.y_M + 0.z_M \\ s = 0.x_M + 0.y_M + 1.z_M \end{cases}$$

- En notation matricielle :

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \simeq \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} . \begin{bmatrix} x_M \\ y_M \\ z_M \\ 1 \end{bmatrix}$$

Attention !

On est pour l'instant en coordonnées **métriques** ! (m.)

Sous-sommaire

Paramètres intrinsèques

- Pour s'affranchir de la focale, on introduit des coordonnées sans dimension : $x_c = \frac{x_M}{z_M}$ et $y_c = \frac{y_M}{z_M}$
- Comme précédemment, on obtient :

$$\begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} \simeq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} . \begin{bmatrix} x_M \\ y_M \\ z_M \\ 1 \end{bmatrix}$$

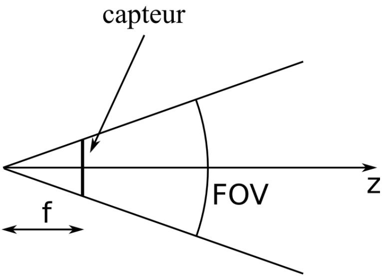
- On passe des coordonnées caméra aux coordonnées "image" via un facteur d'échelle et une translation :

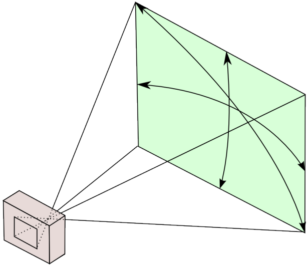
$$u = \alpha_u \cdot x_c + u_0 \qquad \alpha_u = k_u \cdot f$$
$$v = \alpha_v \cdot y_c + v_0 \qquad \alpha_v = -k_v \cdot f$$

- Avec :
  - k_u : facteur d'échelle horizontal (pixel / m)
  - k_v : facteur d'échelle vertical (pixel / m)
  - u_0, v_0 : coordonnées de l'origine du repère caméra dans le repère image

Aspects technologiques : FOV

- Dans certaines situations, on a parfois besoin de connaître l'**angle de champ**, ou **FOV** (*Field of View*), exprimé en degrés.





- On peut le déduire facilement des caractéristiques de la caméra (définition capteur et taille des pixels) et de l'objectif (focale f).
- Pour des pixels carrés, cet angle sera en général différent sur les deux axes (FOV horizontal et vertical).

Matrice des paramètres intrinsèques

- Ces paramètres caractérisent la caméra seule, et s'appellent les **paramètres intrinsèques**.
- On les regroupe dans une matrice $\mathbf{K}$, qui permet de passer des **coordonnées caméras** (x, y, z) aux **coordonnées images** (u, v) :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \mathbf{K}.\begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} \qquad \text{avec} \qquad \mathbf{K} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

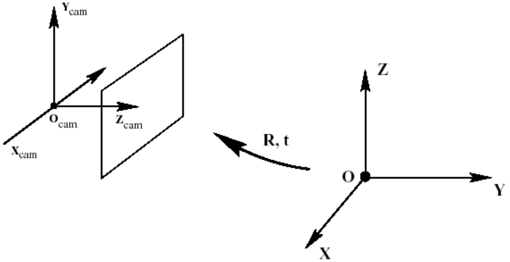
- Le passage des coordonnées caméra aux coordonnées image s'écrit :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Sous-sommaire

Paramètres extrinsèques

- Caractérisent la relation entre le repère "monde" et le repère "caméra".
- Se composent d'une rotation 3D (3 axes) et d'une translation 3D :



$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \simeq \begin{bmatrix} R_{3x3} & t_{3x1} \\ 0^T & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Matrice de projection $\mathbf{P}$

- Combine paramètres intrinsèques et extrinsèques.
- Permet d'exprimer la position d'un point image 2D $\mathbf{m} = (u, v, 1)$ en fonction d'un point 3D $\mathbf{M} = (X, Y, Z, 1)$

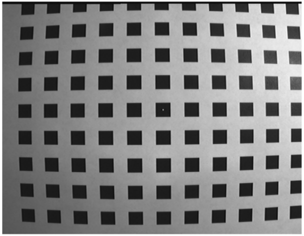
$$\mathbf{m} \simeq \mathbf{PM} \qquad \mathbf{P} = \mathbf{K}. \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} . \begin{bmatrix} R_{3x3} & t_{3x1} \\ 0^T & 1 \end{bmatrix}$$

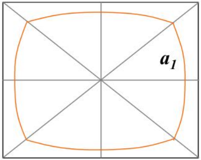
- L'inverse est impossible ! (infinité de solutions. . .)
- On peut l'écrire sous la forme : $\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]$
- Rang 3, 10 DOF : 3 rotations 3D, 3 translations 3D, facteur d'échelle H et V, plus (u_0, v_0))

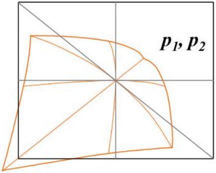
Sous-sommaire

Distorsion optique

- Cette modélisation ne prend pas en compte l'éventuelle distorsion radiale dues au lentilles de l'objectif.

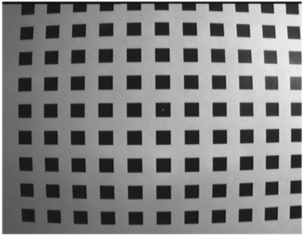


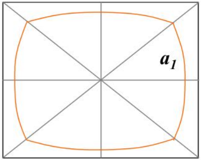


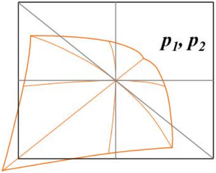


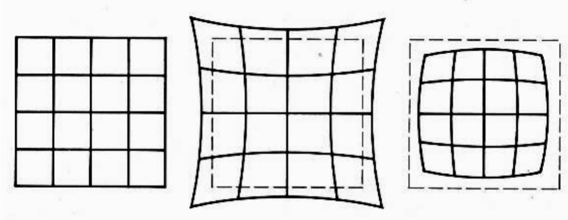
Distorsion optique

- Cette modélisation ne prend pas en compte l'éventuelle distorsion radiale dues au lentilles de l'objectif.









Modélisation de la distorsion radiale

- On modélise cette distorsion radiale par un terme correctif δ, fonction de la distance au centre de l'image et d'un polynôme (Modèle de *Brown*) :

$$\begin{cases} x' = x + \delta_x(x,y) & = x + (x - x_c)\, p(x,y) \\ y' = y + \delta_y(x,y) & = y + (y - y_c)\, p(x,y) \end{cases}$$

- Le terme correctif est une fonction du carré de la distance au centre de distorsion (en général, le centre de l'image) :

$$p(x,y) = k_1 r^2 + k_2 r^4 + \dots$$

- Avec :

$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ $\quad (x_c, y_c)$: position du centre de distorsion

- La calibration complète implique l'estimation des paramètres k_1, k_2,...
- En général 2 coefficients sont suffisants

Correction de la distorsion radiale

- Une fois ces paramètres estimés, un post-traitement permet ensuite la correction de l'image :

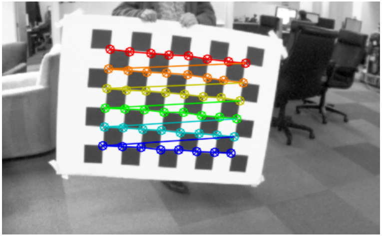
Sous-sommaire

Calibrage monoculaire

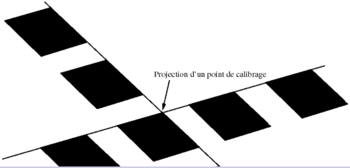
- L'opération de calibrage consiste à **estimer** les paramètres du modèle,
  - soit uniquement les valeurs de la matrice de projection $\mathbf{P}$,
  - soit séparemment paramètres intrinsèques $\mathbf{K}$ et extrinsèques $(\mathbf{R}, \mathbf{T})$.
- De nombreuses techniques publiées, selon les besoins et les données d'entrées, parmi lesquelles :
  - DLT (*Direct Linear Transform*) : solution algébrique directe
  - Algorithme de Tsai (1987)
  - Méthode de Zhang (2000)
- Cette étape implique généralement une phase d'acquisition dédiée, en utilisant une **mire**.

Obtention des points de calibrage

- On utilise une mire type *ckeckerboard*, dont les caractéristiques dimensionnelles sont connues.



- A partir de l'image de la mire, on détermine la position des points 2D par intersection des droites (précision sub-pixellique).



Méthode linéaire directe - 1

- Principe : avec suffisamment de points 3D connus, dont on connait la projection en 2D, on peut en déduire $\mathbf{P}$.
- Chaque point de mesure, composé d'un point 3D $M = (X, Y, Z)$ et d'un point 2D $m_i = (u, v)$ donne un vecteur $\mathbf{x}_i = (X_i, Y_i, Z_i, u_i, v_i)^T$
- En exprimant (u, v) algébriquement à partir de $\mathbf{m} = \mathbf{PM}$:

$$\begin{cases} u = \dfrac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \\ v = \dfrac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \end{cases}$$

Méthode linéaire directe - 2

- Sous forme d'un système linéaire :

$$\begin{cases} p_{11}X + p_{12}Y + p_{13}Z + p_{14} - u\,p_{31}X - u\,p_{32}Y - u\,p_{33}Z - u\,p_{34} = 0 \\ p_{21}X + p_{22}Y + p_{23}Z + p_{24} - v\,p_{31}X - v\,p_{32}Y - v\,p_{33}Z - v\,p_{34} = 0 \end{cases}$$

Méthode linéaire directe - 2

- Sous forme d'un système linéaire :

$$\begin{cases} p_{11}X + p_{12}Y + p_{13}Z + p_{14} - u\,p_{31}X - u\,p_{32}Y - u\,p_{33}Z - u\,p_{34} = 0 \\ p_{21}X + p_{22}Y + p_{23}Z + p_{24} - v\,p_{31}X - v\,p_{32}Y - v\,p_{33}Z - v\,p_{34} = 0 \end{cases}$$

- Afin d'éviter la solution triviale nulle, on fixe p_{34} (par exemple $p_{34} = 1$) :

$$\begin{cases} p_{11}X + p_{12}Y + p_{13}Z + p_{14} - u\,p_{31}X - u\,p_{32}Y - u\,p_{33}Z = u\,p_{34} \\ p_{21}X + p_{22}Y + p_{23}Z + p_{24} - v\,p_{31}X - v\,p_{32}Y - v\,p_{33}Z = v\,p_{34} \end{cases}$$

Méthode linéaire directe - 2

- Sous forme d'un système linéaire :

$$\begin{cases} p_{11}X + p_{12}Y + p_{13}Z + p_{14} - u\,p_{31}X - u\,p_{32}Y - u\,p_{33}Z - u\,p_{34} = 0 \\ p_{21}X + p_{22}Y + p_{23}Z + p_{24} - v\,p_{31}X - v\,p_{32}Y - v\,p_{33}Z - v\,p_{34} = 0 \end{cases}$$

- Afin d'éviter la solution triviale nulle, on fixe p_{34} (par exemple $p_{34} = 1$) :

$$\begin{cases} p_{11}X + p_{12}Y + p_{13}Z + p_{14} - u\,p_{31}X - u\,p_{32}Y - u\,p_{33}Z = u\,p_{34} \\ p_{21}X + p_{22}Y + p_{23}Z + p_{24} - v\,p_{31}X - v\,p_{32}Y - v\,p_{33}Z = v\,p_{34} \end{cases}$$

- On obtient 1 paire d'équations par point $\mathbf{x}_i = (X_i, Y_i, Z_i, u_i, v_i)^T$ (correspondance 3D $\leftrightarrow$ 2D)

Méthode linéaire directe - 3

- Avec n points de mesure, on obtient $2n$ équations, qu'on peut écrire sous forme matricielle :

$$\mathbf{A}_{2n\times 11}.\mathbf{p}_{11\times 1} = \mathbf{B}_{2n\times 1}$$

- Avec :
  - $\mathbf{B}$ contenant les positions 2D

$$\mathbf{B} = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^T$$

Méthode linéaire directe - 3

- Avec n points de mesure, on obtient $2n$ équations, qu'on peut écrire sous forme matricielle :

$$\mathbf{A}_{2n\times 11}.\mathbf{p}_{11\times 1} = \mathbf{B}_{2n\times 1}$$

- Avec :
  - $\mathbf{B}$ contenant les positions 2D

    $$\mathbf{B} = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^T$$

  - $\mathbf{p}$ un vecteur regroupant les 11 éléments de $\mathbf{P}$ à estimer :

    $$\mathbf{p} = (p11, p12, p13, p14, p21, p22, p23, p24, p31, p32, p33)^T$$

    (on fixe $p_{34} = 1$)
  - Chaque paire de lignes i de $\mathbf{A}$ sera construite comme suit :

$$\begin{array}{ccccccccccc} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_i.X_i & -u_i.Y_i & -u_i.Z_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_i.X_i & -v_i.Y_i & -v_i.Z_i \end{array}$$

Résolution du système

$$\mathbf{A} \cdot \mathbf{p} = \mathbf{B}$$

- Minimum 6 points de mesure (12 équations pour 11 paramètres), mais en pratique on en prendra bien plus.
- Si système surdimensionné (plus d'équations que d'inconnues), pas de solution exacte, mais une solution au sens des moindres carrés, donnée par $\mathbf{p} = \mathbf{A}^{+}\mathbf{B}$

Résolution du système

$$\mathbf{A} \cdot \mathbf{p} = \mathbf{B}$$

- Minimum 6 points de mesure (12 équations pour 11 paramètres), mais en pratique on en prendra bien plus.
- Si système surdimensionné (plus d'équations que d'inconnues), pas de solution exacte, mais une solution au sens des moindres carrés, donnée par $\mathbf{p} = \mathbf{A}^{+}\mathbf{B}$
- $\mathbf{A}^{+}$ est la matrice *pseudo-inverse* de $\mathbf{A}$, obtenue par décomposition SVD ("décomposition en valeurs singulières") :
  $\mathbf{A} = U\ D\ V^T \Rightarrow \mathbf{A}^{+} = V\ D^{-1}\ U^T$
- On peut ensuite affiner en utilisant une minimisation non-linéaire utilisant la première estimation comme initialisation.

Autre méthodes

- Inconvénient de cette méthode : il faut avoir un objet de calibrage
  - dont on connait parfaitement la position par rapport à la caméra,
  - de géométrie parfaite,
  - dont les points projetés sont sans ambiguité.
  - **non coplanaire** !

$\Rightarrow$ méthode peu utilisée en pratique. . .

Autre méthodes

- Inconvénient de cette méthode : il faut avoir un objet de calibrage
  - dont on connait parfaitement la position par rapport à la caméra,
  - de géométrie parfaite,
  - dont les points projetés sont sans ambiguité.
  - **non coplanaire** !

  $\Rightarrow$ méthode peu utilisée en pratique...
- D'autre méthodes permettent de s'affranchir de cette contrainte de positionnement.

Autre méthodes

- Méthode de Tsai (1987)
  - Basée sur un objet de calibrage composé de plans à angle droit
  - Prise en compte de la distorsion radiale.
  - Algorithme en deux étapes.

Autre méthodes

- Méthode de Tsai (1987)
  - Basée sur un objet de calibrage composé de plans à angle droit
  - Prise en compte de la distorsion radiale.
  - Algorithme en deux étapes.

- Méthode de Zhang (2000)
  - Basée sur une séquence d'images de mire planaire