

INF555 TD4

Collinéation (Homographie)

Frank NIELSEN
nielsen@lix.polytechnique.fr

5 Octobre 2011

1 Calcul de l'homographie entre deux ensembles de points 2D

On rappelle qu'une homographie est une transformation linéaire de l'espace projectif qui permet d'apparier les points d'une surfaces planaire observée par deux points de vue perspective. Une homographie est appelée encore collinéation puisqu'elle préserve la notion d'incidence (collinéarité). Enfin, l'homographie est également la transformation qui permet de coller (*stitcher* en anglais) une image sur une autre à condition que ces images soient prises du même centre optique par projection perspective (et sans autres distortions).

- Calculer l'homographie (collinéation) qui apparie deux ensembles de 4 points étiquetés: $p_i \leftrightarrow p'_i$. Posez le problème sous la forme $Ah = b$ et résoudre (avec Jama) ce système pour obtenir les 8 coefficients de h (on suppose que $h_{33} = 1$).

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x'_4 & -y_4x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4y'_4 & -y_4y'_4 \end{bmatrix} \times \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{bmatrix}$$

- Calculer en utilisant la méthode des *moindres carrés* l'homographie définie par un ensemble de n points $\{p_i\}_i$ et $\{p'_i\}_i$. Utilisez la matrice pseudo-inverse $A^+ = (A^T A)^{-1} A^T$.
- Utilisez votre programme pour mapper une image sur une autre en utilisant une homographie. On utilisera le **forward** et **backward** mapping (avec une interpolation bilinéaire). Le forward mapping transforme les pixels p de l'image source en $Hp = q$. Le backward mapping calcule $p = H^{-1}q$ et ensuite trouve le pixel dans l'image source.
<http://www.sonycsl.co.jp/person/nielsen/visualcomputing/programs/BlendHomography.cpp>
- Implanter en Java/Processing le programme **unperspective** qui permet de rectifier des parties planes d'une prise de vue perspective en des images 2D Euclidiennes.
<http://www.sonycsl.co.jp/person/nielsen/visualcomputing/programs/Unperspective.cpp>
Observer dans le programme ci-dessus la méthode pour calculer l'homographie. Comparez avec la votre. (Les images sont disponibles sur la page du TD)
- En utilisant la dualité point/droite de l'espace projectif, en déduire une interface utilisateur stable pour sélectionner des droites qui s'apparient dans deux images.