ECE 417/598: Direct Linear Transform

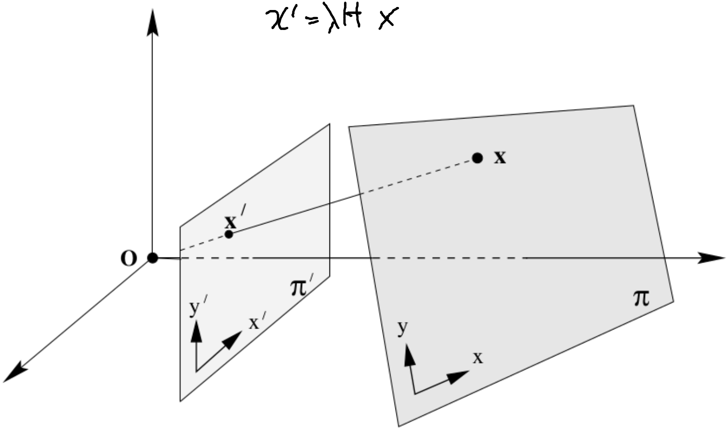
Vikas Dhiman

March 23, 2022

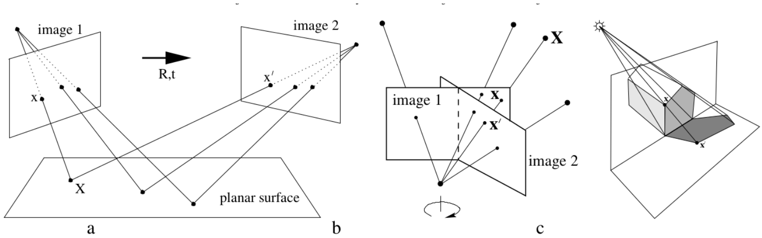
Homography

unknown

$x' = \lambda H x$

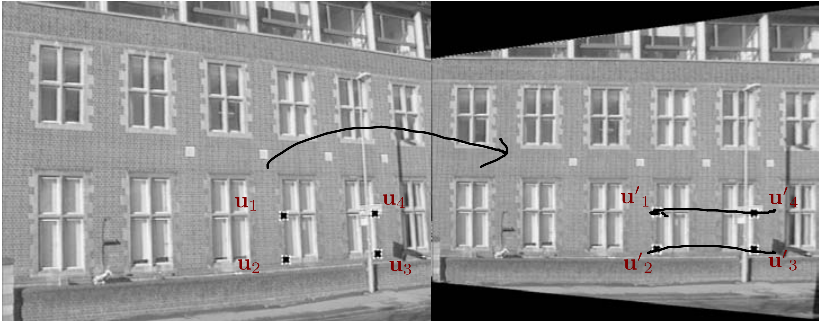


Examples of Homography




Top view
Graphic
Look! Safe to move?
Settings

Computing Homography



$$\underline{\mathbf{u}}_1 = [100, 98, 1]^\top \qquad \underline{\mathbf{u}}_2 = [102, 95, 1]^\top$$

$$\underline{\mathbf{u}}_3 = [107, 90, 1]^\top \qquad \underline{\mathbf{u}}_4 = [110, 85, 1]^\top$$

$$\underline{\mathbf{u}}'_1 = [100, 98, 1]^\top \qquad \underline{\mathbf{u}}'_2 = [102, 95, 1]^\top$$

$$\underline{\mathbf{u}}'_3 = [107, 98, 1]^\top \qquad \underline{\mathbf{u}}'_4 = [110, 95, 1]^\top$$

Find H such that $\underline{\mathbf{u}}' = \lambda H \underline{\mathbf{u}}$ for any point on one image to another image, where $\underline{\mathbf{u}}', \underline{\mathbf{u}} \in \mathbb{P}^2$

2D homography

Given a set of points $\underline{\mathbf{u}}_i \in \mathbb{P}^2$ and a corresponding set of points $\underline{\mathbf{u}}'_i \in \mathbb{P}^2$, compute the projective transformation that takes each $\underline{\mathbf{u}}_i$ to $\underline{\mathbf{u}}'_i$. In a practical situation, the points $\underline{\mathbf{u}}_i$ and $\underline{\mathbf{u}}'_i$ are points in two images (or the same image), each image being considered as a projective plane $\mathbb{P}^2$.

$$\underline{u}'_i = \lambda H \underline{u}_i \longrightarrow A\underline{x} = \underline{b}$$

Perspective space

$\lambda \in \mathbb{R}$

$\underline{u} = KX$ in perspective space

$\underline{u} = \lambda \cancel{K} X$

$$\underbrace{\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = 4\begin{bmatrix} 4 \\ 8 \\ 12 \end{bmatrix}}_{\text{Perspective space}}$$

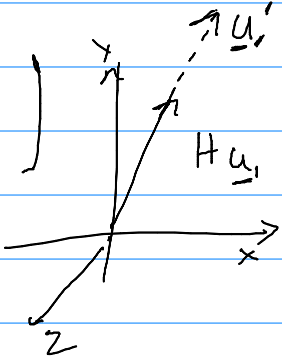
$ax + by + c = 0$

$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

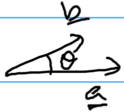
$$\underline{u}_i' \in \mathbb{P}^2 = \begin{bmatrix} x_i' \\ y_i' \\ w_i' \end{bmatrix} \qquad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

$$\underline{u}_i \in \mathbb{P}^2 = \begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix}$$

$$\underline{\underline{u}}_i' = \lambda H \underline{u}_i = \begin{bmatrix} \ \end{bmatrix}$$



$$\underline{u}_i' \times H \underline{u}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$



$$\|\underline{a} \times \underline{b}\| = \|a\|\|b\| \sin\theta$$

If $\underline{a} \parallel \underline{b}$ then $\underline{a} \times \underline{b} = \underline{0}$

$$\underline{u}_i' = \lambda H \underline{u}_i$$

$$\begin{bmatrix} x_i' \\ y_i' \\ w_i' \end{bmatrix} = \lambda \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix}$$

$$\underset{\alpha}{x_i'} = \underbrace{\lambda h_1}_{} \underset{\alpha}{x_i} + \underbrace{\lambda h_2}_{} \underset{\alpha}{y_i} + \underbrace{\lambda h_3}_{} \underset{\alpha}{w_i}$$

Quadratic

$\hookrightarrow x^2, \; y^2, \; xy$ ← unknown

Cubic

$\hookrightarrow x^3, \; xyz, \; y^3, \; z^3$ ←

$$\underline{u}_i' \times H \underline{u}_i = 0$$

$$\underbrace{\begin{bmatrix} 0 & -w_i' & y_i' \\ w_i' & 0 & -x_i' \\ -y_i' & x_i' & 0 \end{bmatrix}}_{\text{cross product matrix}} H u_i = 0$$

$$\underbrace{\underline{a} \times}\, \underline{b}$$

$$= \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -w_i' & y_i' \\ w_i' & 0 & -x_i' \\ -y_i' & x_i' & 0 \end{bmatrix} H u_i = 0$$

$$H = \begin{bmatrix} \underline{h}_1^T \\ \underline{h}_2^T \\ \underline{h}_3^T \end{bmatrix}.$$

$$\underline{h}_1^T = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix}$$

$$H\underline{u}_i = \begin{bmatrix} \underline{h}_1^T \\ \underline{h}_2^T \\ \underline{h}_3^T \end{bmatrix} \underline{u}_i$$

$$= \begin{bmatrix} \underline{h}_1^T \underline{u}_i \\ \underline{h}_2^T \underline{u}_i \\ \underline{h}_3^T \underline{u}_i \end{bmatrix}$$

$$\begin{bmatrix} 0 & -w_i' & y_i' \\ w_i' & 0 & -x_i' \\ -y_i' & x_i' & 0 \end{bmatrix} H u_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -w_i' & y_i' \\ w_i' & 0 & -x_i' \\ -y_i' & x_i' & 0 \end{bmatrix} \begin{pmatrix} \underline{u}_i^T \underline{h}_1 \\ \underline{u}_i^T \underline{h}_2 \\ u_i^T h_3 \end{pmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{pmatrix} 0 - w_i' \underline{u}_i^T h_2 + y_i' \underline{u}_i^T h_3 \\ w_i' u_i^T \underline{h}_1 + 0 - x_i' \underline{u}_i^T h_3 \\ -y_i' \underline{u}_i^T h_1 + x_i' u_i^T h_2 + 0 \end{pmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\underbrace{\begin{array}{c} \begin{array}{ccc} \xleftrightarrow{\;3\;} & \xleftrightarrow{\;3\;} & \xleftrightarrow{\;3\;} \end{array} \\ \begin{bmatrix} \underline{0}^T_{3\times 1}, & -w_i' \underline{u}_i^T, & y_i' \underline{u}_i^T \\ w_i' \underline{u}_i^T & 0 & -x_i' \underline{u}_i^T \\ -y_i' \underline{u}_i^T & x_i' \underline{u}_i^T & 0 \end{bmatrix}_{3\times 9} \end{array}}_{\substack{\text{given} \\ A}} \underbrace{\begin{bmatrix} \underline{h}_1 \\ \underline{h}_2 \\ \underline{h}_3 \end{bmatrix}_{9\times 1}}_{\substack{\text{unknown} \\ \underline{h}}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}_{3\times 1}$$

$$A \underline{h} = 0$$

$$M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \qquad N = \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix}$$

$$\underset{2\times 2}{M} \otimes \underset{2\times 2}{N} = \begin{bmatrix} m_{11} N & m_{12} N \\ m_{21} N & m_{22} N \end{bmatrix}_{4\times 4}$$

$$\left(M_{p\times q} \otimes N_{r\times s} \right)_{(pr)\times(qs)}$$

KRONECKER PRODUCT

$$\underbrace{\begin{bmatrix} 0^T_{3\times 1}, & -w_i' \underline{u}_i^T, & y_i' \underline{u}_i^T \\ w_i' \underline{u}_i^T & 0 & -x_i' \underline{u}_i^T \\ -y_i' u_i^T & x_i' u_i^T & 0 \end{bmatrix}_{3\times 9}} \begin{bmatrix} \underline{h}_1 \\ \underline{h}_2 \\ \underline{h}_3 \end{bmatrix}_{9\times 1} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}_{3\times 1}$$

$$[\underline{u}_i']_\times \otimes \underline{u}_i^T \begin{bmatrix} \underline{h}_1 \\ \underline{h}_2 \\ \underline{h}_3 \end{bmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$H = 3 \times 3 = 9$ unknowns

from each point

3 eqns

Because of $\mathbb{P}$

8 DOF

2 linearly independant equations

$8/2 = 4$ points (pair of points)

$$\underline{u}_1 \longleftrightarrow \underline{u}_1'$$
$$\underline{u}_2 \longleftrightarrow u_2'$$
$$\vdots$$
$$\underline{u}_4 \longleftrightarrow \underline{u}_4'$$

$$\begin{array}{l} \text{1st pt} \\ \\ \text{2nd pt} \\ \\ \end{array}
\begin{bmatrix} 0^T_{3\times 1}, & -\omega_1' \underline{u}_1^T, & y_1' \underline{u}_1^T \\ \omega_1' \underline{u}_1^T & 0^T & -x_1' \underline{u}_1^T \\ 0^T_{3\times 1} & -\omega_2' \underline{u}_2^T & y_2' u_2^T \\ \omega_2' \underline{u}_2^T & 0^T & -x_2' u_2^T \end{bmatrix}
\begin{pmatrix} \underline{h}_1 \\ \underline{h}_2 \\ \underline{h}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Solving for Homography

$$\begin{array}{cc} \text{8eqn} & \underbrace{\begin{bmatrix} \quad & \quad & \quad \\ \\ \\ \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \underline{h_1} \\ \underline{h_2} \\ \underline{h_3} \end{bmatrix}}_{\underline{x}} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \end{array}$$

$$= 0$$

Solving for Homography

$$A\underline{x} = 0 \quad \Big\} \text{ Nullspace}$$

$$A_{8\times 9} \qquad \text{rank}(A) = 8$$

$$\underset{8\times 9}{A} = \underset{8\times 8}{U}\,\underset{8\times 9}{\Sigma}\,\underset{9\times 9}{V^T}$$

$$U \in \mathbb{R}^{8\times 8}$$

$$V \in \mathbb{R}^{9\times 9}$$

$$V = \begin{bmatrix} \underline{v}_1 & \cdots & \underline{v}_9 \end{bmatrix}$$

$$N(A) = \underline{v}_9 = \begin{bmatrix} \underline{h}_1 \\ \underline{h}_2 \\ \underline{h}_3 \end{bmatrix}$$

$$H = \begin{bmatrix} \underline{h}_1^T \\ \underline{h}_2^T \\ \underline{h}_3^T \end{bmatrix}$$

Solving for Homography

```
Eigen::Matrix3d
findHomography(const std::vector<Eigen::Vector3d>& us,
               const std::vector<Eigen::Vector3d>& ups)
{
    Eigen::MatrixXd A(8, 9); A.setZero();
    for (int i = 0; i < us.size(); ++i) {
        // [[0ᵀ      -w'ᵢ uᵢᵀ   yᵢ' uᵢᵀ]]
        //  [wᵢ'uᵢᵀ       0ᵀ   -xᵢ uᵢᵀ]]
        A.block(2*i, 3, 1, 3) = -ups[i](2)*us[i].transpose();
        A.block(2*i, 6, 1, 3) = ups[i](1)*us[i].transpose();
        A.block(2*i+1, 0, 1, 3) = ups[i](2)*us[i].transpose();
        A.block(2*i+1, 6, 1, 3) = -ups[i](0)*us[i].transpose();
    }

    auto svd = A.jacobiSvd(Eigen::ComputeFullV);
    Eigen::Matrix3d H;
    Eigen::VectorXd nullspace = svd.matrixV().col(8);
    H.row(0) = nullspace.block(0, 0, 3, 1).transpose();
    H.row(1) = nullspace.block(3, 0, 3, 1).transpose();
    H.row(2) = nullspace.block(6, 0, 3, 1).transpose();

    return H;
}
```

Apply Homography

```
Eigen::MatrixXd
applyHomography(const Eigen::Matrix3d& H,
                const Eigen::MatrixXd& img) {
    Eigen::MatrixXd new_img(img.rows(), img.cols());
    Eigen::Vector3d u;
    Eigen::Vector3d up;
    for (int new_row = 0; new_row < new_img.rows(); ++new_row) {
        for (int new_col = 0; new_col < new_img.cols(); ++new_col) {
            u << new_col + 0.5, new_row + 0.5, 1;
            /**** Apply homography for each pixel ***/
            up = H * u;
            up /= up(2);
            /**** Apply homography for each pixel ***/
            int row = round(up(1));
            int col = round(up(0));
            if (0 <= row && row < img.rows()
                && 0 <= col && col < img.cols()) {
                new_img(new_row, new_col) = img(row, col);
            }
        }
    }
    return new_img;
}
```