CLAB3: Theory and Applications

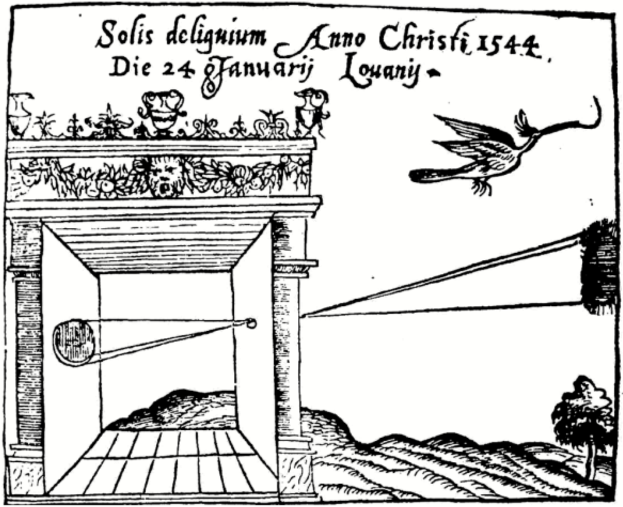
Ryan Pike

Research School of Engineering

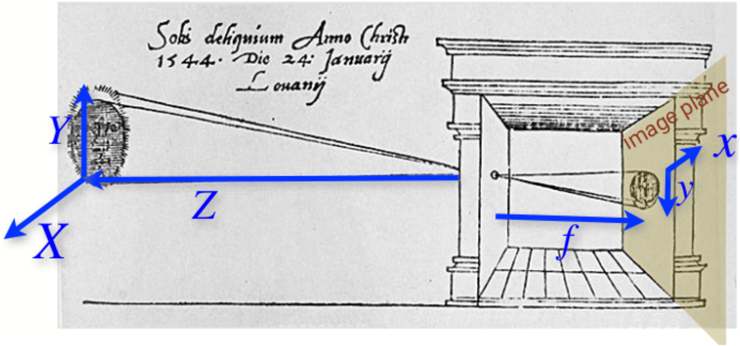
Australian National University

Slides adopted from RVSS Summer School 2020, Corke. P

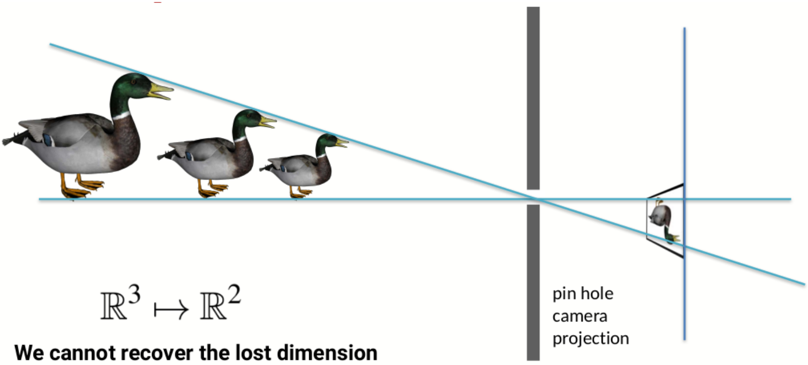
The Pinhole Camera



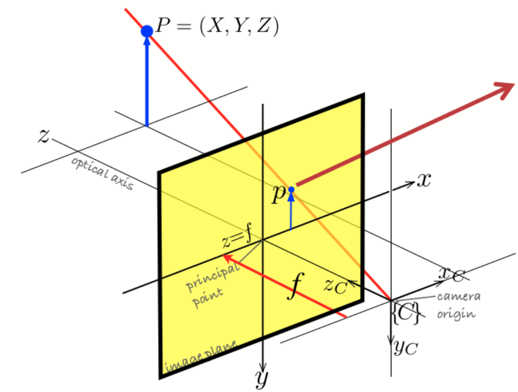
The Pinhole Camera (Cont.)



Uniqueness Concerns



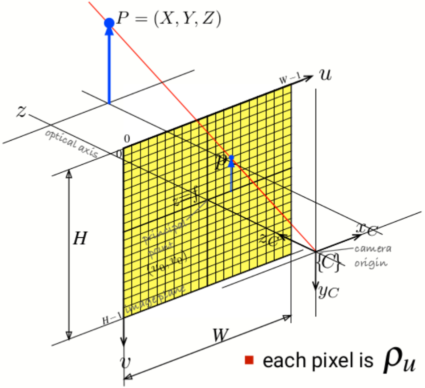
Pinhole Camera Model



Corke, P. | Reproduced with permission from Springer Science, & Business Media

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Incorporating Pixels



Corke, P. | Reproduced with permission from Springer Science, & Business Media

- each pixel is $\rho_u \times \rho_v$

- scale point from metres to pixels
- shift the origin to top left corner

$$u = \frac{x}{\rho_u} + u_0$$

$$v = \frac{y}{\rho_v} + v_0$$

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} \frac{1}{\rho_u} & 0 & u_0 \\ 0 & \frac{1}{\rho_v} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix}$$

$$p = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \tilde{u}/\tilde{w} \\ \tilde{v}/\tilde{w} \end{pmatrix}$$

The Complete Camera Model

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} \frac{1}{\rho_u} & 0 & u_0 \\ 0 & \frac{1}{\rho_v} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathrm{R} & t \\ 0_{1\times 3} & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

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

$$u = \frac{\tilde{u}}{\tilde{w}}, v = \frac{\tilde{v}}{\tilde{w}} \tag{1}$$

Using the scale invariance

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$u = \frac{\tilde{u}}{\tilde{w}}, v = \frac{\tilde{v}}{\tilde{w}} \tag{2}$$

$$\implies u_i = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + 1} \tag{3}$$

DLT Algorithm (See P. 16 of Lecture 7-B)

Direct linear transformation (DLT) is an algorithm which solves a set of variables from a set of similarity relations.

1. Remove scale factor and get it into a linear form
2. Rearrange terms to make it a linear system of equations for one constraint
3. Stack the linear system for n constraints

Rule of thumb: Number of constraints must equal or exceed the number of DOFs of the transformation

DLT Algorithm

The following two equations is merely computing u, v from the previous slides.

$$u_i = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + 1} \tag{4}$$

$$v_i = \frac{p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + 1} \tag{5}$$

$$u_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + 1) = p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14} \tag{6}$$

To use DLT, just set equation (6) to equal zero such that you get the following matrix A_i.

DLT (Single 3D-2D correspondence)

$$\begin{pmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i & -v_i \end{pmatrix}$$

$$A_i h = (0,0)^\top, h = [p_{11}, p_{12}, \dots p_{33}]^\top, \tag{7}$$

Form A by concatenating each A_i row wise. Each A_i represents a correspondence between the image point $(u_i, v_i)^\top$ and the 3D coordinate $(X_i, Y_i, Z_i)^\top$. This will give you an overconstrained equation if you select 12 points (from assignment, N is No. of calibration points).;

$$Ah = 0, \qquad A \in \mathbb{R}^{2N\times 11}, h \in \mathbb{R}^{11\times 1}, 0 \in \mathbb{R}^{2\mathrm{N}\times 1} \tag{8}$$

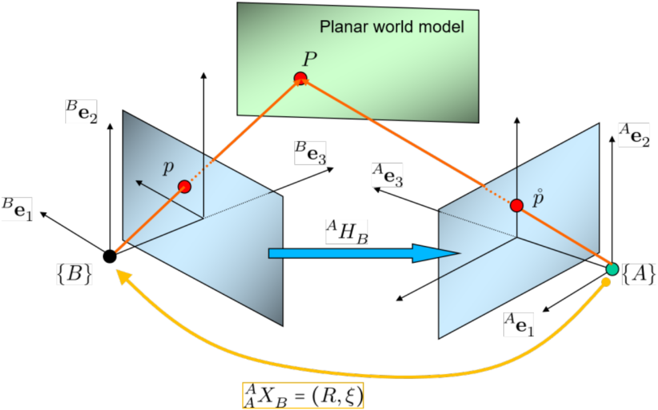
$$\text{Solution (Find parameter vector h):} \qquad \underset{\|h\|=1}{\arg\min} \|Ah\| \tag{9}$$

Homography (EECI Adopted)

A homography is an invertible mapping relating two images of the same planar scene. Any two images of the same planar surface in space are related by a homography.

- Image rectification
- Image registration
- Stereo vision
- Computation of camera motion between two images

Homography (Graphically)



$$Hp_i \cong \mathring{p}_i \tag{10}$$

The matrix H;

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

defines a homography mapping from one image B to A.

$$Hp \cong H \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \cong \begin{pmatrix} \mathring{u} \\ \mathring{v} \\ 1 \end{pmatrix} \cong \mathring{p} \tag{12}$$

Since scale is not considered then $H' = \sigma H, \sigma \neq 0$ will generate the same homography mapping. The set of possible homographies corresponds to an 8-dimensional matrix sub-manifold of $\mathbb{R}^{3\times3}$

DLT (2D-2D Correspondence)

Similar to the projection matrix approach you merely transform the following equations so that it can be represented as a linear matrix;

$$u = \frac{h_{11}\mathring{u} + h_{12}\mathring{v} + h_{13}}{h_{31}\mathring{u} + h_{32}\mathring{v} + 1} \tag{13}$$

$$v = \frac{h_{21}\mathring{u} + h_{22}\mathring{v} + h_{23}}{h_{31}\mathring{u} + h_{32}\mathring{v} + 1} \tag{14}$$

Now instead of having a RHS element as was the case for the DLT method we can use all available variables in the A_i matrix to produce the following;

$$Ah = 0 \tag{15}$$

where $h = [h_{11}, \dots, h_{32}, 1]^\top$

Homography DLT

Given $\{p_i, \mathring{p}_i\}$ solve for H such that $\mathring{p}_i = Hp_i$

1. Apply normalisation to points in left and right image (see below)
2. For each correspondence create, a 2×9 matrix A_i
3. Concatenate into a single $2n \times 9$ matrix A
4. Compute SVD of $A = U\Sigma V^\top$
5. Store eigenvector with the smallest eigenvalue $\tilde{h} = v_{\lambda_{min}}$
6. Reshape to get H

Each column of V represents a solution for $Ah = 0$ where the eigenvalues for the respective column represent the reprojection error (try prove this)!

Normalisation (Essential in DST!, Taken from Hartleys Book S4.1)

Given $n \geq 4$ 2D to 2D point correspondences $\{x_i \leftrightarrow x_i'\}$, determine the 2D homography matrix H such that $x_i' = Hx_i$

Algorithm

1. Normalization of x: Compute a similarity transformation T, consisting of a translation and scaling, that takes points x_i to a new set of points $\tilde{x}_i$ such that the centroid of the points $\tilde{x}_i$ is the coordinate origin $(0,0)^\top$, and their average distance from the origin is $\sqrt{2}$.
2. Normalization of x': Compute a similar transformation T' for the points in the second image, transforming points x_i' to $\tilde{x}_i'$
3. Apply algorithm from above
4. Set $H = T'^{-1}\tilde{H}T$

Thanks for listening (things to keep in mind)

- Camera calibration is one of the most important procedures in computer vision. If you do it poorly, nothing works!
- The DLT is simply a method to transform a set of relationships so that they are linear in the variables. Then you just setup a large matrix and solve for the coefficients.
- Consider exploring different optimisation techniques for finding the solution, the least squares estimator, and the SVD does not do well with outliers. Try it out!
- See if there are other optimisation algorithms you can implement that are more robust to outliers
- On a planar scene atleast 4 points are required to compute a homography. See how the estimation changes from choosing 4 pairs to 6 pairs, and see even beyond that.