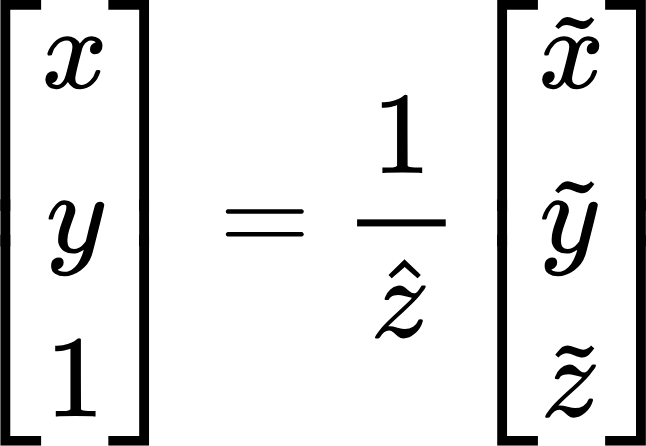
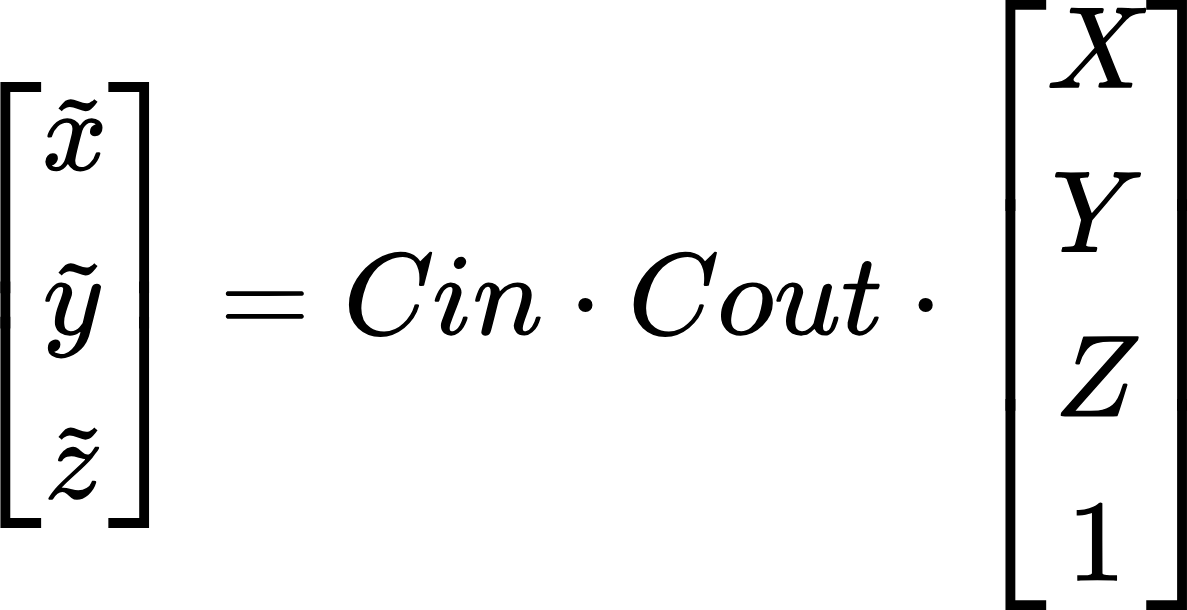
# Homogeneous coordinates

Translation of 2D pixel from image plane to 3D real world

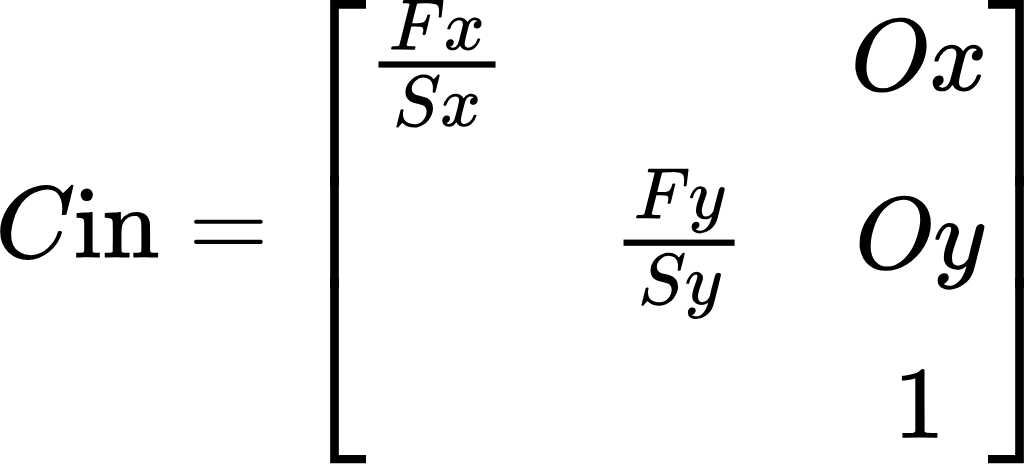


# **3D point to 2D image**

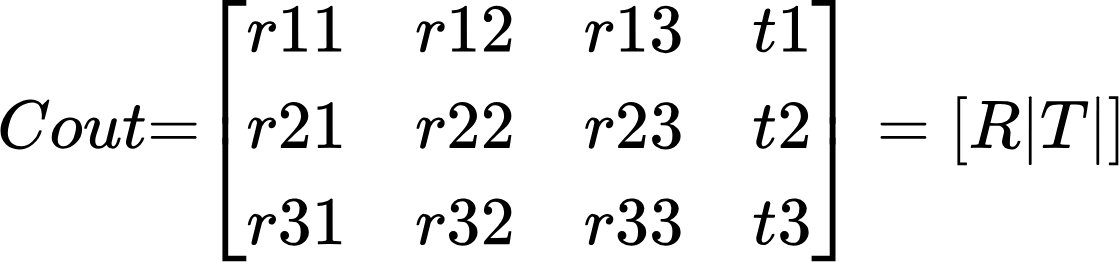
**Cin - camera intrinsic, Cout - camera extrinsic**

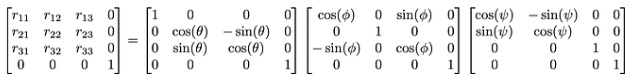
****

**Intrinsic - translate 3D point to 2D**

, F - focal length, S-sensor size, O - image center

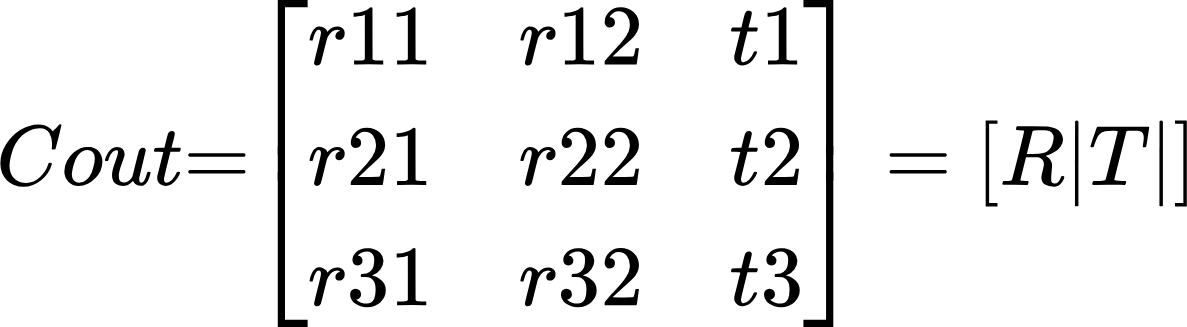
**Extrinsic - translate between coordinate frames**

****, R - rotation matrix, T-translation matrix



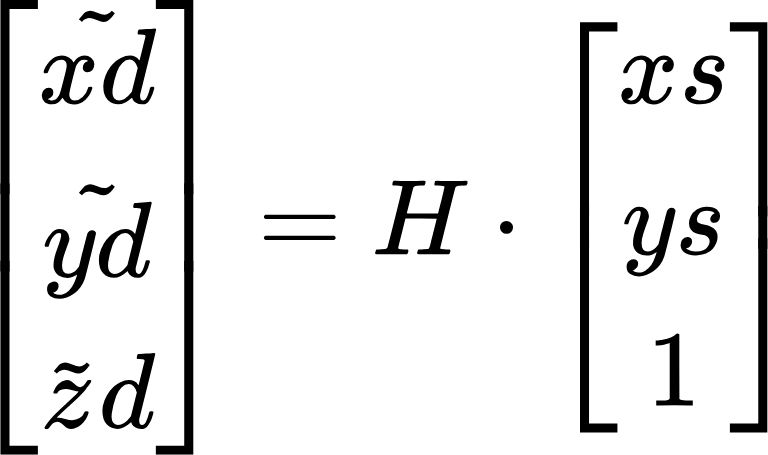
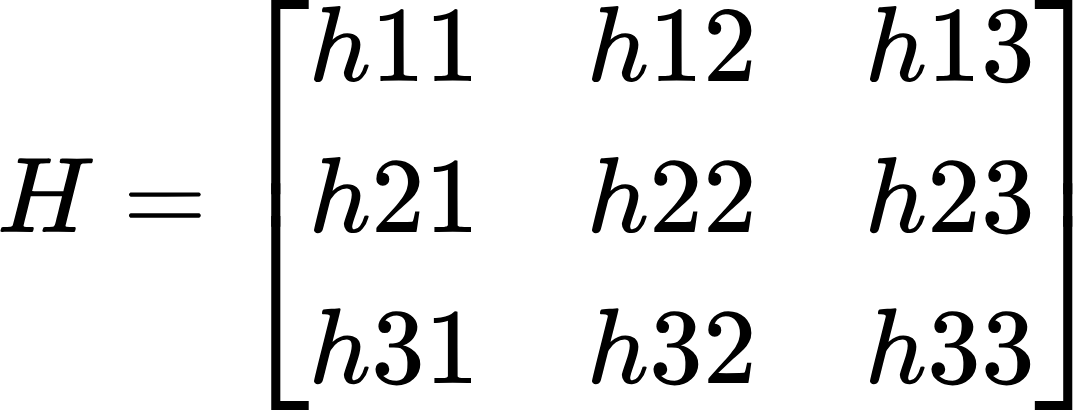
# Homography Matrix

we are looking for a planar surface in the world-view to compute the homography, ***Z***= 0



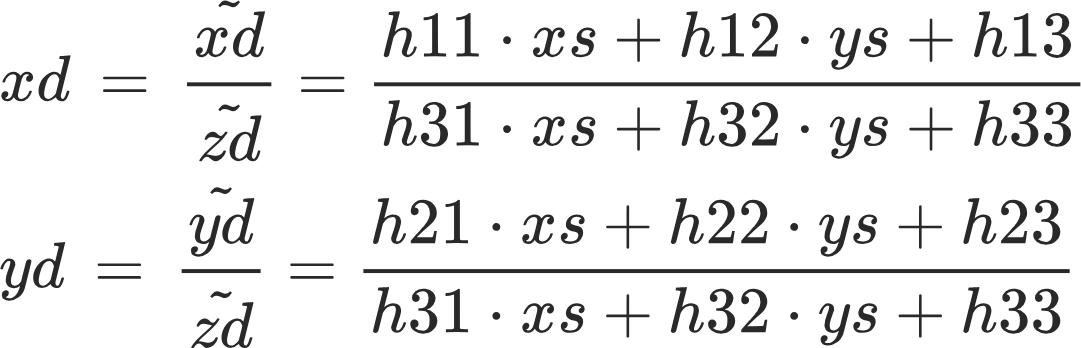
Homography matrix - translation between shots

S -source, d - destination

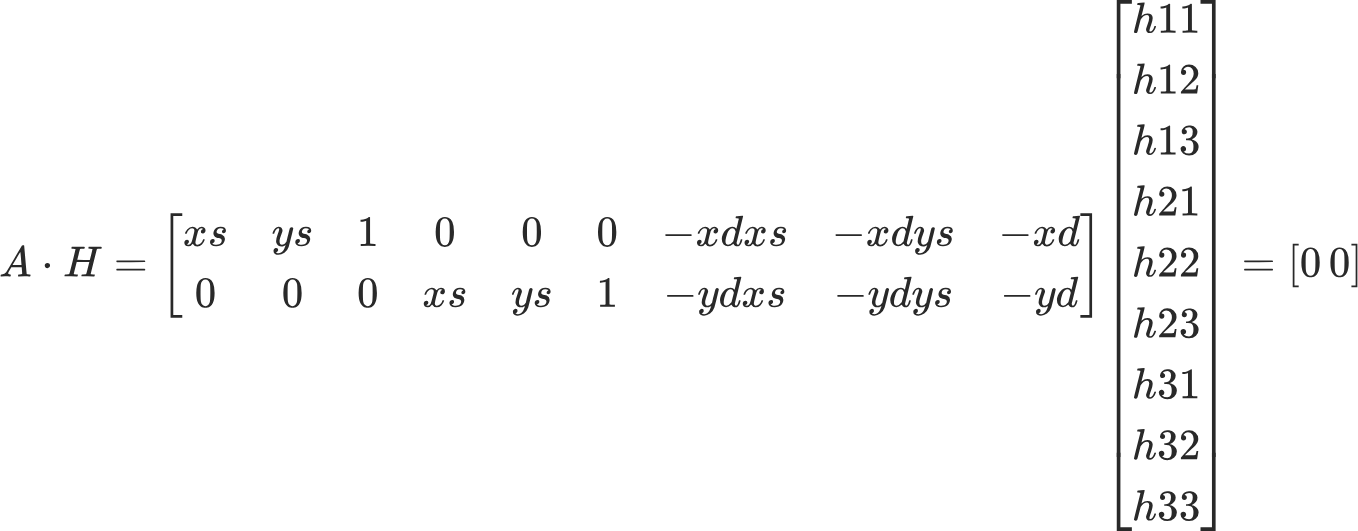
, , h33 - image scale with pixel position save, so H matrix has 8 degrees of freedom

## Computation

Every point gives 2 features, so H matrix computation needs 4 points



Multiply by denominator and rearrange all to left side



Concatenate 3 more points to the left matrix for H estimation. Find H with known A such that ||H||2=1, which is **Constrained Least Squares problem:**

### **Constrained Least Squares problem**

**min**h||A\*H||2 , ||H||2=1

||A\*H||2=(A\*H)T\*(A\*H)=HT\*AT\*A\*H, HT\*H=1

Define Loss Function:

L(H, l) = HT\*AT\*A\*H - l\*(HT\*H-1)

Take derivative of loss to find extremum:

AT\*A\*H - l\*H=0

AT\*A\*H = l\*H - **Eigenvalue problem**

**Eigenvector h with smallest eigenvalue l would give minimal Loss value**