

[Open in app](#)[Get started](#)

Published in Towards Data Science



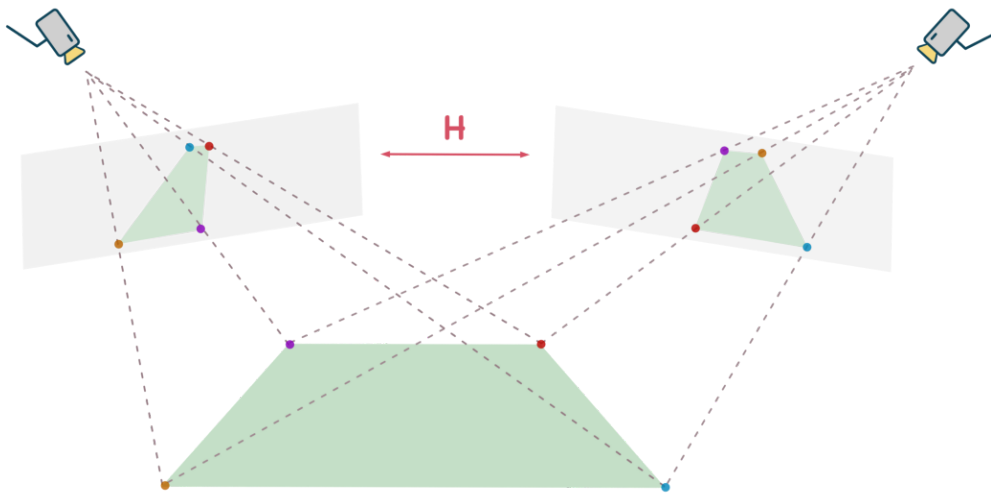
Yalda Shankar

[Follow](#)

Apr 24, 2021 · 7 min read · [Listen](#)



Save



Homography Concept Visualization [ Image by author ]

[ LET'S KNOW SERIES ] — #1

# Homography Estimation

*This short piece describes equations for estimating a  $3 \times 3$  homography matrix. We first discuss the computation from intrinsic and extrinsic camera parameters; and wherever necessary, connect the formulation to the real-world camera specifications. We next present the computation methodology using two tuples of corresponding points, that are co-planar in their respective planes and avoid collinear degenerations.*

## 1| HOMOGRAPHY FROM CAMERA PARAMETERS

### a. Basic Setup





The 3D world-view is captured onto a 2D image through a camera, placed in the world-view coordinates.

We can then map this 3D point to a point in an *arbitrary space*, as follows:

$$\begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} = C_{int} C_{ext} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Eq. (1)

where  $C_{int}$  is the *intrinsic* and  $C_{ext}$  is the *extrinsic* camera matrix respectively. The point  $(x_a, y_a, z_a)$  in the arbitrary space, can be mapped to the 2D image space by involving a scale factor as follows:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{z_a} \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix}$$

Eq. (2)

Thus, once we have a point in the arbitrary space, we can simply scale its coordinates to get the 2D coordinates in the (captured) image space.

## b. The Intrinsic Matrix

Let us now see the form of  $C_{int}$ . Consider a camera with a focal length  $f$  (in mm), with actual sensor size  $(x_S, y_S)$  (in mm), and the width and the height of the captured image (effective sensor size) as  $(w, h)$  (in pixels).

The *optical centre*  $(o_x, o_y)$  of the camera is then  $(w/2, h/2)$ . We are now in a state to specify  $C_{int}$  as follows:

$$C_{int} = \begin{bmatrix} \frac{fw}{x_S} & 0 & o_x \\ 0 & \frac{fh}{y_S} & o_y \end{bmatrix} = \begin{bmatrix} \frac{fw}{x_S} & 0 & w/2 \\ 0 & \frac{fh}{y_S} & h/2 \end{bmatrix}$$





One may thus observe that all entries in  $C_{int}$  are in the units of pixels. The following may be seen as the effective focal lengths in pixels in  $x$  and  $y$  directions respectively

$$\frac{fw}{x_S}, \frac{fh}{y_S}$$

In practice, they are taken the same, since when it is calculated from the camera specifications, they have a very slight difference ( $\approx 0.2$  to  $0.5$  %)

### c. The Extrinsic Matrix

$C_{ext}$  consists of a rotation matrix  $R$  and a translation matrix  $T$  as follows:

$$C_{ext} = [R | T] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix}$$

Eq. (4)

The tuple  $(T_x, T_y, T_z)$  indicates the translation of the camera in the world-space coordinates. Typically, we can consider that the camera has no  $x$  and  $y$  translation ( $T_x = T_y = 0$ ) and the height of the camera position from the ground (in mm) equates to  $T_z$ .

If  $\theta, \phi, \psi$  be the orientation of the camera with respect to  $x, y$  and  $z$  axes respectively (as angles in radians), we may get  $r_{ij}; i, j \in \{1, 2, 3\}$  as follows:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Eq. (5)

### d. The Homography

The homography matrix  $H$  to be estimated is a  $3 \times 3$  matrix, and encompasses the parts of both *intrinsic* and *extrinsic* camera matrices as follows:





The above can be directly established from the fact, that when we are looking for a planar surface in the world-view to compute the homography,  $Z_w = 0$ , and thus,

$$\begin{bmatrix} \frac{fw}{x_s} & 0 & w/2 \\ 0 & \frac{fh}{y_s} & h/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w (= 0) \\ 1 \end{bmatrix} \\ = \begin{bmatrix} \frac{fw}{x_s} & 0 & w/2 \\ 0 & \frac{fh}{y_s} & h/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & T_x \\ r_{21} & r_{22} & T_y \\ r_{31} & r_{32} & T_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

Eq. (7)

Hence, the homography  $H$  will map a world-view point in the arbitrary space. This space suffices well, in case we just need to compute the distances between any two given points. However, in reality, the coordinates in the pixel space will be calculated by considering the scale factor as specified in Eq. (2).

## 2] HOMOGRAPHY FROM CO-PLANAR POINTS

### a. Basic Setup

Homography lets us relate two cameras viewing the same planar surface; Both the cameras and the surface that they view (generate images of) are located in the world-view coordinates. In other words, two  $2D$  images are related by a *homography*  $H$ , if both view the same plane *from a different angle*. The homography relationship does not depend on the scene being viewed.

Consider two such images viewing the same plane in the world-view.

Let  $(x_1, y_1)$  be a point in the first image, and  $(x_2, y_2)$  be the corresponding point in the second image. Then, these points are related by the *estimated* homography  $H$ , as follows:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = H_{2 \times 2} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$





$$\begin{bmatrix} \hat{x}_1 \\ \hat{y}_1 \\ 1 \end{bmatrix} = \frac{1}{z_a} \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix}$$

Eq. (9)

Thus, any point in the first image may be mapped to a corresponding point in the second image through homography, and the operation may be seen as that of an image warp.

### b. The Homography

Let us parametrize the  $3 \times 3$  homography matrix  $H$  as follows:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

Eq. (10)

Thus, the estimation of  $H$  requires the estimation of 9 parameters. In other words,  $H$  has 9 degrees of freedom. If we choose two tuples of corresponding points,  $\mathbb{P}[\text{co-planar}]$  in their respective planes, as follows:

$$T_1 = ((x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4))$$

$$\hat{T}_1 = ((\hat{x}_1, \hat{y}_1), (\hat{x}_2, \hat{y}_2), (\hat{x}_3, \hat{y}_3), (\hat{x}_4, \hat{y}_4))$$

Eq. (11)

**[co-planar]** The homography relation is provable only under the co-planarity of the points, since everywhere, we are assuming that the  $z$ -coordinate of any point in any image is 1. In practice, for instance, one may thus choose four points on a floor, or a road, which indicate a *nearly* planar surface in the scene.





$$\begin{bmatrix} \hat{x}_i z_a \\ \hat{y}_i z_a \\ z_a \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

Eq. (12)

Where  $(\hat{x}_i, \hat{y}_i) \in T^*_1$  and  $(x_i, y_i) \in T_1$  for  $i, j \in \{1, 2, 3, 4\}$ . This will then translate to the following system of equations to be solved:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1\hat{x}_1 & -y_1\hat{x}_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2\hat{x}_2 & -y_2\hat{x}_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3\hat{x}_3 & -y_3\hat{x}_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4\hat{x}_4 & -y_4\hat{x}_4 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1\hat{y}_1 & -y_1\hat{y}_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2\hat{y}_2 & -y_2\hat{y}_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3\hat{y}_3 & -y_3\hat{y}_3 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4\hat{y}_4 & -y_4\hat{y}_4 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = h_{33} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \hat{y}_4 \end{bmatrix}$$

Eq. (13)

We now have 8 equations, which can be used to estimate the 8 degrees of freedom of  $H$  (except  $h_{33}$ ). For this, we would require that the  $8 \times 8$  matrix above has a full rank (no redundant information), in the sense that none of the rows are linearly dependent. This implies that *no three points* in either of  $T_1$  or  $T^*_1$  should be collinear.

We then need to tackle  $h_{33}$ . Note that in Eq. (13), if  $h_{33}$  is pre-equated to 1, we would simply shift the entire set of  $h_{ij}$  hyperplanes to another reference frame, but their directions would not change. Practically, we would thus, simply see a different  $z_a$  value, while mapping a 2D image coordinate according to Eq. (8), which would subsequently get divided out in Eq. (9). Hence, we keep  $h_{33} = 1$  in  $H$ , and Eq. (13) may then be solved using least-squares estimation.

In *OpenCV*, one can use the function ***findHomography***, which does precisely the same, as delineated above. It accepts two tuples of four corresponding points, and computes the homography  $H$  with  $h_{33}$  always and strictly 1. Any 2D image point will then be mapped to a  $z_a$  amplified version of the corresponding point in the other plane.





and compute a homography matrix, which can project any point in the observed scene to the plane of the image captured by  $C$ .

Imagining  $C$  in the bird's eye (top view) is a **[popular choice]** In such a case, one may choose  $T_1$  with four co-planar points in the observed scene, while the corresponding tuple  $T^*_1$  can simply have four points as corners of a hypothetical rectangle, with a Euclidean coordinate system, centered around  $(0, 0)$ . Any point in the scene can then be mapped to its bird's eye view, i.e. how may it look from the top.

**[popular choice]** There has been a recent surge of research papers, which exploit the bird's eye view (BEV) for behavioural prediction and planning in autonomous driving.

Note here that homography-based mapping is only a warped version of the observed image, and that no new information in the scene is being synthesized. For instance, if we have only observed the frontal view of a person in the scene, its birds-eye view, won't really start telling as to how the person's hair is from the top; but it will only warp the frontal-view visible portion of his head in a way that it would roughly look like a top-view.

#### d. Negative values in projections with Homography

Note that while solving for  $H$ , there is no constraint that the projection points in the arbitrary space should be positive, i.e.  $x_a, y_a$  and  $z_a$  may be negative. Once scaled out by  $z_a$ , this would mean that a mapped point  $(x^*_i, y^*_i)$  may be negative.

This may look intuitively undesirable since the image coordinates are generally taken to be positive. However, this may be seen as only a reference axis shift, and after mapping the entire image, the amount of shift may be appropriately decided.



**Remark** - The treatment presented here, *may not be* akin to 3D reconstruction procedures, which may involve estimation of multiple-view homographies', sometimes via a hypothesized view projection. Multi-view homographies, have been shown to possess specific algebraic structures, but 3D reconstruction from 2D scenes largely remains an unsolved problem.





Open in app

Get started

---

## Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. [Take a look.](#)

Get this newsletter

