

# Removing Projective Distortion from Images

Mickey Smith

August 2019

## 1 Homogenous Coordinates

Similar to cartesian coordinates, homogenous coordinates are a method to represent specific in a given dimension.

$$\begin{pmatrix} x \\ y \end{pmatrix}$$

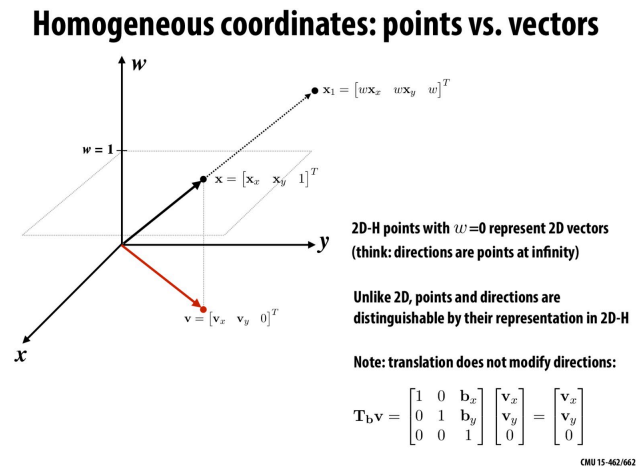
(a) Cartesian  $\mathbb{R}^2$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

(b) Homogenous  $\mathbb{R}^2$

**Figure 1:** Representing coordinates as homogenous and cartesian

Note that the only practical difference between cartesian and homogenous representations (for the purposes of this assignment) is that homogenous coordinates contain an extra dimension value that represents vector scaling. This allows a point to be further from the origin on its vector passing through the origin without altering the other coordinates.



**Figure 2:** Explanation of homogenous coordinates

## 2 Projective Transformations

When a photo is taken of a scene, the camera applies a transformation to the original scene in the form of a matrix called the *homography*, creating a new perspective captured by the photo. An original scene  $\mathbf{x}$  manipulated by a camera and resulting in a photo becomes  $\mathbf{x}'$  through the application of a projective homography transformation  $H$ .

$$\mathbf{x}' = H\mathbf{x}$$

**Figure 3:** Homography application.  $\mathbf{x}$  represents the original scene and  $\mathbf{x}'$  represents what we see in a photograph taken of the scene at an angle.

This equation is notably of the form  $Ax = B$ , which will allow us to solve for the homography matrix easily<sup>1</sup>.

In order to restore the original image's perspective we need to determine the homography matrix's inverse,  $H^{-1}$ , not just the homography matrix itself. Once we have  $H^{-1}$  we can apply it to every pixel in the image to determine where it should be placed within the new image.

In order for a program to apply these inverse transformations, 8 coordinates within  $\mathbb{R}^2$  have to be defined. 4 representing the corners of a box within the supplied image and 4 representing the destination location of those points within the new image that represents the real orthogonal world view of the image. We'll choose those locations as the corners of a rectangle within the image.

$$\begin{pmatrix} x^{ll} \\ x^{lr} \\ x^{ul} \\ x^{ur} \end{pmatrix}$$

(a) Image points

$$\begin{pmatrix} x'^{ll} \\ x'^{lr} \\ x'^{ul} \\ x'^{ur} \end{pmatrix}$$

(b) World points

**Figure 4:** User-defined parameters (where a point exists within the photograph and where it should exist in a warped view of the photograph to represent a different perspective).

Once these 8 points have been defined they can be placed into the  $A$  and  $B$  matrices for the  $Ax = B$  equation and used to solve for  $H$ . The ultimate equation will be an 8x8  $A$  matrix multiplied with an 8x1  $H$  matrix resulting in an 8x1  $B$  matrix[Figure 5].

$$\begin{pmatrix} x^i & y^i & 1 & 0 & 0 & 0 & -x^i x'^i & -x^i y'^i \\ 0 & 0 & 0 & x^i & y^i & 1 & -y^i x'^i & -y^i y'^i \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{pmatrix} = \begin{pmatrix} x'^i \\ y'^i \end{pmatrix} \quad (1)$$

**Figure 5:** Homography application

---

<sup>1</sup>For the purposes of this assignment the Python library Numpy's *linear algebra* class' *solve()* method is used to compute the solution vector  $x$ .