1 Introduction

This text aims at explaining homographies and to give a practical guide on how to use it. A homography is a transformation from one Projective plane to another. This is very useful for generating "fake" images on objects from views that were never actually used. Also inverse perspective mapping, IPM, makes use of this, by projecting a taken image on a know plane. This way one gets rid of the perspective effect on objects that are on that plane, for all others this transformation results in a distorted view. To understand homographies some background knowledge about the Projective space and camera modeling is necessary. This tutorial is not complete, yet. So far, it covers most of the necessary background knowledge and is to be extended.

2 Projective Geometry

Projective geometry is an extension of the familiar Euclidean geometry. The latter allows us to describe transformations like translations and rotations, but we cannot model more complicated transformations like perspective (or central) projections. Perspective projection is a mathematical model (description) that allows us to precisely describe how the world around us gets mapped on a plane, it is described in the section "Perspective Projection" below. Note, that there are also other projections, e.g. the orthographic projection, but we will not consider those here. Why can we not use Euclidean geometry for this transformation of perspective projection? Because in Euclidean geometry several properties are preserved under transformation, like lengths, angles and parallelism. That means, if one uses a transformation defined in the euclidean space, e.g. rotation on two parallel lines the result will also be a pair of parallel lines. (In fact that is one of the axioms for this geometry.) However, when we use the transformation of perspective projection on a pair of parallel lines the resulting pair is not supposed to be parallel anymore. Everybody knows this effect when looking at a long straight train track, (which is a projection of two parallel lines) then the track converges to a single point at the horizon, see Fig. 1. Our first goal is to
derive the mathematical model for the perspective projection. We will see, that it is very easy to do so in Euclidean space, (in fact it is more difficult to remember all the newly introduced terms) but that it results in a nonlinear equation. To transform this nonlinear equation to a linear one we can use homogeneous coordinates which live in the Projective space. Therefore, before we turn to the perspective projection we are going to have a look at homogeneous coordinates.

2.1 Projective Space and Homogeneous Coordinates

Homogeneous coordinates were invented by August Ferdinand Möbius [1]. To get an intuition let's look at Barycentric coordinates first: If you have a triangle you can describe the point $p$ as the center of gravity by the weights required at each corner as shown in Fig. 2. Thus, the coordinate vector of the two-dimensional point $p$ are the three weights $d_1, d_2, d_3$. If the triangle and all of these weights were scaled by some factor, the point would still be the center of gravity. This shows an important feature of homogeneous coordinates: $n+1$ values are needed to represent an $n$-dimensional point, which makes the point description invariant to scaling! In other words: Points in homogeneous coordinates are defined up to a scale factor.

Consequently, to convert a 2-dimensional Euclidean point $m = [x, y]^T$ into homogeneous coordinates, one must add a third dimension usually (points in homogeneous coordinates are usually denoted by adding a $\tilde{}$ to the point variable, thus, $\tilde{m}$), i.e.:

$$m = [x, y]^T \mapsto \tilde{m} = [x, y, 1]^T.$$ (1)
Since the homogeneous point $\tilde{m}$ is defined up to a scale vector we can write:

$$\tilde{m} = [x, y, 1]^T = [\alpha x, \alpha y, \alpha]^T = [X, Y, Z]^T, Z \neq 0 \quad (2)$$

Often vector notation is used:

$$\tilde{m} = \alpha [X, Y, Z]^T. \quad (3)$$

To convert the homogenous point back to Euclidean coordinates we must divide by the last coordinate:

$$\tilde{m} = [X, Y, Z]^T \mapsto m = [X/Z, Y/Z] \quad (4)$$

What happens if $Z = 0$? These points are not defined in the Euclidean space, thus, the Projective space contains more points! These points are the so called points at infinity, we give a more intuitive description soon. In fact, adding an additional dimension is the same as adding points at infinity to the space we are working in. Therefore, the Projective space is the Euclidean space plus points at infinity. Another way to think about this is to imagine that you add a reference point to the space that you are working with, through which all points and lines must pass. This is also shown below.

**Projective Line** Let's look at the Euclidean line to use the above explanations to derive at the Projective line. The following is visualized in Fig. 3. A point on a line in $\mathbb{R}^1$ in Euclidean coordinates is given by a coordinate vector of length 1, $m = [x]$. If we want to convert the Euclidean line to a Projective line we have to introduce one further dimension, i.e. we go from $\mathbb{R}^1 \mapsto \mathbb{R}^2$, yielding $\tilde{m} = [X, Y]^T$. We can now represent all points on the Euclidean line homogeneous points. It can be seen that homogeneous points are actually represented by rays. All points on a given ray correspond to the same Euclidean point, this is a consequence of the invariance to scaling. The Fig. 3 also nicely shows how we augmented the space by a reference point through which all rays pass.
The Projective Plane

If you augment the ordinary (Euclidean) plane with points at infinity (that is you add the third coordinate, that can be 0) you get the Projective plane. Again, in the Projective plane all lines and planes pass through one point. This is visualized in Fig. 4. Here, $\pi$ denotes the Euclidean plane. The sphere indicates all lines and planes that pass through the reference point. Lines or planes that intersect $\pi$ are the homogeneous lines or points in the Euclidean plane. Those lines and planes that do not intersect with $\pi$ are by definition said to intersect at infinity, they are also called ideal lines or points. Consider the example from the beginning, that two parallel line intersect in one point. That is a point at infinity. This is visualized in Fig. 5. Also note, that antipodal points represent the same homogeneous point.

Fig. 3. The extension from the Euclidean line to the Projective line. It can be seen that points on the Projective line correspond to rays.

Fig. 4. The extension from the Euclidean plane to the Projective plane. All rays in the Projective plane pass through one point. The points at infinity are those that do not get projected on the plane. The figure also shows that an Euclidean Geometry is contained within a Projective Geometry.

Fig. 5. Shows how parallel lines get mapped to lines that intersect in one point after the perspective projection. The image is from Digital image processing, or Image analysis and understanding from website:
2.2 Duality

An important concept in Projective space, is that of duality. Intuitively that means, that in statements about the Projective plane one can substitute "line" by "point" and vice versa as we will demonstrate in the following.

**Homogeneous Line Equation** A line in the Euclidean plane is defined by:

\[ ax + by + c = 0. \]  \hspace{1cm} (5)

That is, a point on the line has the Cartesian coordinates \([x, y]^T\). Considering that this point is represented in homogeneous coordinates as \([X, Y, Z]^T\), as shown in equation 3, with \(x = X/Z\) and \(y = Y/Z\) we can substitute the homogeneous coordinates and divide by \(Z\), thus:

\[
a \frac{X}{Z} + b \frac{Y}{Z} + c \Rightarrow aX + bY + cZ = \tilde{I}, \; \text{homogeneous line equation}. \]  \hspace{1cm} (6)

Formula 6 is referred to as the homogeneous line equation. Note, that it corresponds to an Euclidean plane equation through the origin.

\[ p = n(r - p) = 0. \]  \hspace{1cm} (7)

**Point on a Line** Now we come to an important consequence: For a homogeneous point \(\tilde{m}\) in order to lie on a homogeneous line the dot product (= scalar product) between both must vanish:

\[ \tilde{m} \tilde{I} = 0, \; \text{point on a line} \]  \hspace{1cm} (8)

(Recall that the dot product between two vectors \(a\) and \(b\) is equal to \(ab = |a||b| \cos(\theta)\)).

**Line intersecting two Points** To calculate the line \(\tilde{I}\) through two points \(\tilde{m}_1\) and \(\tilde{m}_2\) we use the cross-product:

\[ \tilde{I} = \tilde{m}_1 \times \tilde{m}_2 \; \text{line intersecting two points}. \]  \hspace{1cm} (9)

This is also visualized in Fig. 6.

**Intersection Point of two Lines** To calculate the intersection point \(\tilde{m}\) through two lines \(\tilde{I}_1\) and \(\tilde{I}_2\) we also (!) use the cross-product:

\[ \tilde{m} = \tilde{I}_1 \times \tilde{I}_2 \; \text{point of intersection of two lines}. \]  \hspace{1cm} (10)

As can be seen from equations 9 and 10 points and lines can be exchanged. This is the principle of duality in the Projective plane.
2.3 Perspective Projection

A perspective projection is used to model the imaging process of a pinhole camera, or more generally speaking the projection of the world around us onto a plane, where all rays of light are intersecting at one point. Fig. 7 shows how an object gets mapped up-side-down onto an image plane by the rays of light passing through a single hole. Let us now introduce some important terms: We call the point through which all rays pass the focal node and denote it with $C$. Furthermore, let us call the plane that the object, in this case the tree, is mapped to the image or retinal plane, and the plane parallel to it containing the focal node (in the image the shaded rectangle) focal plane. The distance between these two planes is $f$, the focal length. If we assume that all points in the world can be identified by a coordinate vector $[x, y, z]$ and if we place the origin of this (Cartesian) coordinate system to the focal node such that the $z$-axis points into space and $x$ and $y$ span the focal plane, we can schematically redraw the projection as done in Fig. 8. If we now want to know how a point in the world $M$ gets mapped onto the image plane $m$ we see immediately (similar triangles) that the $y$-coordinate of $M$ corresponds to $-fy/z$. (Similar triangles are those where the corresponding angles are equal, but not the size of the triangle. The ratios of corresponding sides are then also equal. Concerning this case here, that means that $y/z = y'/f \Rightarrow y' = -fy/z$. Similarly $x \Rightarrow -fx/z$.) Summing up and using vector notation:

$$M \mapsto m = [x, y, z]^T \mapsto \left[\frac{-fx}{z} - \frac{fy}{z}, -f\right]^T,$$ perspective projection. \hspace{1cm} (11)

This formula describes the projective projections from world to image coordinates, from $\mathbb{R}^3 \rightarrow \mathbb{R}^2$. The focal node is also called camera center or optical
center and the point where the z axis pierces the image plane is called principal point. The z-axis is also called principal axis. These terms are described in image 9.

**Fig. 7.** Visualizing the principle (Arbeitsweise?) of a pinhole camera.

**Fig. 8.** deriving perspective projection. xxx ERROR! the last coordinate for the image point is -f and not z!!

**Fig. 9.** Put an image where you explain all terms from above

Unfortunately equation 11 is not linear (a linear equation is a special algebraic equation. An algebraic equation is an equation between polynomials, i.e. only addition, subtraction, power to nonnegative values, and multiplication is allowed. The division is not, since it corresponds to a nonnegative power! Therefore, this equation is not a polynomial and therefore not an algebraic equation and therefore not a linear equation. personal notes:We only said that a linear equation is a special form of an algebraic equation, that means that here each term is either a constant or the product of a constant and the first power of a single variable. Note, in polynomials you can have higher powers. Linear equations can have more than one variables, if it is of two variables, x and y, it can be interpreted as a line equation. The variable y can be written as a function of x,
therefore we also speak of a linear function. A linear function is more generally a linear map.

How do we get \( y \mapsto -\frac{fx}{z} \) into a linear form, like \( y = ax + b \)? The trick is to use homogeneous coordinates, where the division is "hidden" in the extended variable. Thus, equation 11 can be rewritten linearly:

\[
\begin{align*}
\tilde{m} & \mapsto \tilde{M} \\
\begin{pmatrix} -fx \\ -fy \\ -f \\
\end{pmatrix} & \mapsto \begin{pmatrix} x \\ y \\ 1 \\
\end{pmatrix} \\
\tilde{m} & = PM
\end{align*}
\]

Note, that we transformed \( M \) into a homogeneous coordinate vector \( \tilde{M} \) by adding a 1 as additional dimension, but that we use \( m \) as is, i.e. we only interpret it as a homogeneous vector \( \tilde{m} \). This leads to the following: If we transform the interpreted \( \tilde{m} \) back into \( m \) we divide by the last coordinate and then drop it:

\[
\begin{align*}
\tilde{m} & \mapsto m \\
\begin{pmatrix} -fx \\ -fy \\ -f \\
\end{pmatrix} & \mapsto \begin{pmatrix} \frac{x}{z} \\ \frac{y}{z} \\ 1 \\
\end{pmatrix} \\
\end{align*}
\]

Finally, we denote:

\[
\begin{pmatrix} u \\ v \\
\end{pmatrix} = \begin{pmatrix} \frac{x}{z} \\ \frac{y}{z} \\
\end{pmatrix}
\]

In other words:

\[
\alpha \begin{pmatrix} u \\ v \\
\end{pmatrix} = \begin{pmatrix} \frac{x}{z} \\ \frac{y}{z} \\
\end{pmatrix}
\]

Why is it so nice, to have this equation in linear form? Because we can now concatenate many transformations into one matrix! Therefore, it can be seen that homogeneous coordinates are very helpful when dealing with image projections.

### 3 Camera Modeling

We have seen how a world point gets mapped onto a plane. However, this was a simplified scenario. Usually the projection is subdivided into three parts. The external-, projective- and internal part.
### 3.1 Internal transformation

We have denoted the transformed world point as $\tilde{m}' = [u, v]^T$ on the retinal plane, i.e. on the CCD chip. However, the origin usually lies in the upper left corner. That is, we need a translation of $u_0, v_0$. Furthermore, we must change from metric system e.g. mm into pixel coordinates. For that we need to not the size of a pixel in mm, $k_u, k_v$. This can be put into a matrix.

$$\tilde{m} = H \tilde{m}' \quad (20)$$

$$H = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (21)$$

with $\tilde{m}$ being the transformed point. Combined with the perspective projection we get:

$$\tilde{P}_{\text{new}} = H \tilde{P}_{\text{old}} \quad (22)$$

$$\tilde{P}_{\text{new}} = \begin{bmatrix} -f k_u & 0 & u_0 \\ 0 & -f k v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (23)$$

**Intrinsic Parameters** The parameters of the matrix $\tilde{P}_{\text{new}}$ are:

- focal length $f$
- pixel size in $x$ and $y$ direction $k_u$ and $k_v$
- the distance in $x$ and $y$ direction to the upper left corner of the retinal plane $u_0$ and $v_0$

These are five unknown parameters called the intrinsic parameters, since they are camera dependent.

### 3.2 External transformation

So far we assumed that the center of projection coincides with the the origin of the world coordinate system. Usually this is not the case. Therefore, we must shift the world coordinate system into the camera coordinate system which is done by a rotation and a translation:

**Extrinsic Parameters** The parameters of the matrix $\tilde{P}_{\text{new}}$ are:

- three rotation vectors
- three translation values

These are six unknown parameters called the extrinsic parameters, since they are not camera dependent.

$$\begin{pmatrix} R \\ t \\ 0^T \end{pmatrix}, \quad (24)$$
3.3 Camera Calibration

The process to estimate intrinsic and extrinsic parameters from known point correspondences is called camera calibration.

4 Homography, not ready yet

Linear transformation/map/operator (lineare Abbildung) = a function between two vector spaces that preserves the operations of vector addition and scalar multiplication. Linear transformations can be nicely described by matrices. Because a linear transformation from \( x \) to \( y = x = a^*y \) and if \( a \) is a vector, you can write all coefficients in a matrix. Eine lineare Transformation ist in der Sprache der abstrakten Algebra ein Homomorphismus. Das bedeutet, dass der map der auf einer Menge mit einem Magma (einem binären Operator) definiert ist, danach wieder eine Menge ergibt, in der man das Magma anwenden kann. Affine Abbildung = Rotation, Skalierung, Scherung. \((A^*x)\) Lineare Transformation = Affine Abbildung + Translation. \((A^*x + t)\)

5 Practical Example

5.1 Getting the Plane Equation

We first calibrated a single camera with Matlab. Those, that do not have Matlab can use the free implementation of the same calibration algorithms in OpenCV. However, here, we stick to the Matlab notation. Put the camera on the robot and take all images for calibration. Then do the calibration procedure, you get intrinsic and extrinsic parameters. The extrinsic parameters describe how you get from the origin of the world coordinate system, which is now the starting point (where you start to click the four corners) on the calibration grid, to the origin of the camera coordinate system. Thus, if you use another image, and you know, where it is, how far away from the camera, then, using the extrinsic parameters, which are a rotation and a translation vector, you can convert from image to world. If you then put an image on the ground, you can use this information to calculate a plane equation!

The first clicked point is selected to be associated to the origin point of the reference frame attached to the grid. Click 1 to 2 is y coordinate and 4 to 1 is x. z is pointing upwards (right-hand coordinate system) This is nicely shown on pictures shamelessly stolen from the camera calibration toolbox website: Do the calibration and save Calib_Results.mat. This you can then read in later and follow the instructions for calculating extrinsic parameters only, or, if a certain image was part of the calibration procedure, you can check a file in which everything is already stored. Thus, for each image, used for calibration there is already the intrinsic and extrinsic parameters given.

The extrinsic parameters are encoded in the form of a rotation matrix \((R_{c_{ext}})\) and a translation vector \((T_{c_{ext}})\). The rotation vector \(omc_{ext}\) is related to the
rotation matrix \((R_{c_{\text{ext}}})\) through the Rodrigues formula: 
\[ R_{c_{\text{ext}}} = \text{rodrigues}(\text{omc}_{\text{ext}}). \]

Let us give the exact definition of the extrinsic parameters: Let \(P\) be a point space of coordinate vector \(XX = [X; Y; Z]\) in the grid reference frame \((O, X, Y, Z)\) shown on the following figure:

Let \(XXc = [Xc; Yc; Zc]\) be the coordinate vector of \(P\) in the camera reference frame \((Oc, Xc, Yc, Zc)\). Then \(XX\) and \(XXc\) are related to each other through the following rigid motion equation:

\[ XXc = R_{c_{\text{ext}}} \times XX + T_{c_{\text{ext}}} \]

We use the image shown in Fig. ??This is what we got from it:

\[
\text{omc}_{90} = [1.934632e + 00; -1.046264e - 02; 3.405648e - 02] \quad (25)
\]

\[
T_{c90} = [-5.246784e + 01; 3.864721e + 02; 5.010625e + 02] \quad (26)
\]

\[
\text{omc}_{\text{error}}_{90} = [5.984460e - 03; 3.820616e - 03; 7.277470e - 03] \quad (27)
\]

\[
T_{c_{\text{error}}90} = [1.509933e + 00; 2.070000e + 00; 2.477613e + 00] \quad (28)
\]

We know from the tutorial page: \(T_c = \) translation vector \(\text{omc} = \) rotation vector The rotation vector is related to the rotation matrix through the Rodriguez formula: 
\[ R_{c_{\text{matrix}}} (R_{c_{\text{ext}}}) = \text{rodrigues}(\text{omc}) \]

We now know where the point is that lies in the plane, that we are interested in, but we still do not know the plane equation. However, we can calculate it the following way:

![Image to get the plane equation.](image)

**References**