

Projective Geometry and Transformations in Space

Bakr Albatran

July 1, 2005

1 Homogeneous coordinates

1.1 Introduction

In projective n -space p^n a point X with cartesian coordinates $(x_1, \dots, x_n)^T$ is represented by a vector $(wx_1, \dots, wx_n, w)^T$ of homogeneous coordinates, usually with $w = 1$.

Two homogeneous vectors X and Y are called equivalent $X \sim Y$ if a nonzero scalar λ exists such that $x_i = \lambda y_i \forall i \in \{1, \dots, n+1\}$ and $X = (x_1, \dots, x_n, x_{n+1})^T$ and $Y = (y_1, \dots, y_n, y_{n+1})^T$. All vectors equivalent to each other form an equivalence class.

From now on we are going to identify a vector X by its class of homogeneous vectors and we are going to write " \sim " instead of " $=$ " such as $X \sim (x_1, x_2, 1)^T$

1.2 Reasons for using homogeneous coordinates

There are many reasons for using homogeneous coordinates. One of these reasons is to simplify transformation combinations as rotation, scaling and shearing are all multiplicative transforms while only translation is an additive transform. Homogeneous coordinates allow us to apply all four transformations multiplicatively (figure 1). This is suitable for computer application using these transformations.

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} x_1 + a \\ x_2 + b \end{pmatrix} \xrightarrow{\text{becomes}} \begin{pmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = \begin{pmatrix} x_1 + a \\ x_2 + b \\ 1 \end{pmatrix}$$

Figure 1: Changing translation from additive transform into multiplicative transform using homogeneous coordinates.

It gives us the opportunity to perform certain nonlinear operations, e.g. operations that are not affinities in the underlying \mathbb{R}^n , by using linear operators in p^n . Furthermore homogeneous vectors are highly related to central

projection.

We have a representation for points that are at infinity, these are the points where $w = 0$, known as ideal points. Therefore we can calculate with them as normal points which is impossible in inhomogeneous coordinates which is taught at the school, e.g. $X \sim (a, b, 0)^T$ is a point at infinity with cartesian coordinates $(a/0, b/0)$.

1.3 Points and lines in projective 2 – space p^2 (2D homogeneous coordinates)

1.3.1 Points:

In p^2 a point X is identified by a vector $(x_1, x_2, w)^T$ and represented by a line which goes through the origin and intersects the plane π which is at a distance w from the origin and is parallel to the xy – plane (figure 2).

Transformation: the point X can be transformed under a 3×3 – matrix H . We obtain $X' \sim HX$.

Degrees of freedom: The point X has 2 degrees of freedom, the three elements of a homogeneous 3-vector less one for scale.

1.3.2 Lines:

In p^2 a line is represented by a plane which goes through the origin and intersects the plane π in a line $L \sim (a, b, c)^T$, therefore the specific line is identified by the vector $(a, b, c)^T$ (figure 2).

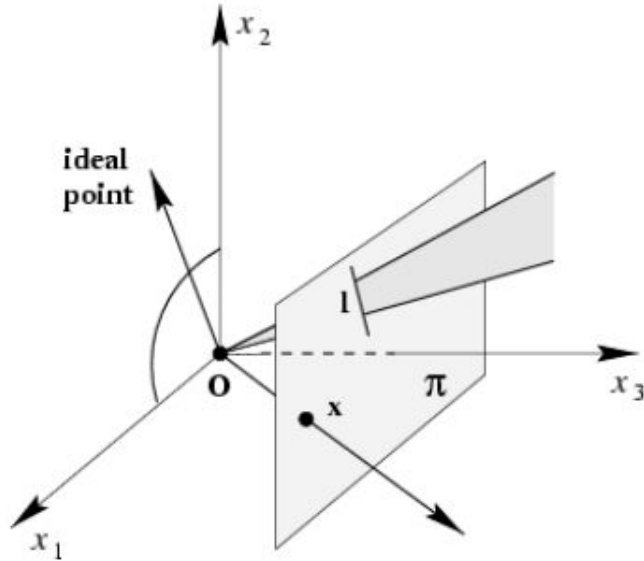


Figure 2: points and lines in p^2 .

From school geometry we know that a point $X \sim (x, y)^T$ lies on a line L with the equation $ax_1 + bx_2 + c$ if $ax + by + c = 0$. Using homogeneous coordinates we can write this equation as a scalar product of X and L . X lies on the line L if $X^T L = (x, y, 1)^T (a, b, c) = 0$

Note that this equation remains valid, even if we scale the vector X or the vector L with an arbitrary scalar $\lambda \neq 0$

Degrees of freedom: a line L has 2 degrees of freedom, the three elements of a homogeneous 3-vector less one for scale.

Intersection of two lines: The intersection of two lines L and L' defines a point X and can be calculated as the cross product of L and L' $X \sim l \times l'$.

Join of two points: The join of two points X and X' defines a line L and can be calculated as $L \sim X \times X'$.

Transformation: The line L can be transformed under transformation matrix H such as $L' \sim H^{-T}L$ because if $X \sim (x_1, x_2, 1)^T$ and $L \sim (a, b, c)^T$ we obtain $L'^T X' = (H^{-T}L)^T HX = L^T H^{-1}HX = L^T X = 0$.

1.4 Points and planes in projective 3 – space p^3 (3D homogeneous coordinates)

1.4.1 Points

Points in projective 3 – space are represented by the equivalence classes of homogeneous 4 – vectors and have thus 3 degrees of freedom. A homogeneous point $X \sim (x_1, x_2, x_3, x_4) \in p^3$ corresponds to an inhomogeneous point $(x_1, x_2, x_3) \in \mathbb{R}^3$ if $x_4 \neq 0$ and $x_1 = x_1/x_4, x_2 = x_2/x_4, x_3 = x_3/x_4$.

Transformation: a point X can be transformed by a 4×4 – matrix H as shown in the following equation: $X' \sim HX$.

Degrees of freedom: a point X has 3 degrees of freedom, the four elements of a homogeneous 4-vector less one for scale.

1.4.2 planes

In homogeneous coordinates a plane π is represented by the vector $(a, b, c, d)^T$. From school mathematics we know the equation for planes in 3D. A 3D point $X = (x, y, z)^T$ lies on the plane π with the equation $ax_1 + bx_2 + cx_3 + d$ if $ax + by + cz + d = 0$. Using homogeneous coordinates we can write this equation as a scalar product of X and π . X lies on the plane π if $X^T \pi = (x, y, z, 1)^T (a, b, c, d) = 0$

From this equation we can see that planes play the same role in projective 3-space as lines did in the 2D case.

Degrees of freedom: a plane π has 3 degrees of freedom, the four elements of a homogeneous 4-vector less one for scale.

Three points define a plane: The join of three noncolinear points X_1, X_2, X_3 defines a plane and is computed by the solution of the following matrix equation:

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} \pi = M\pi = 0$$

It follows that π is the one-dimensional right null-space of the matrix M . $\pi = \text{Null}(M)$ can be computed using singular-value decomposition (SVD).

Remember $X_i = (x_1, x_2, x_3, 1)$!

Three planes define a point: The intersection of three planes π_1, π_2, π_3 in general positions defines a point X which can be computed by the following:

$$\begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{pmatrix} X = MX = 0$$

The resulting point X is the one-dimensional right null-space of the matrix M . $X = \text{Null}(M)$ can be computed using singular-value decomposition (SVD).

Remember $\pi_i = (a, b, c, d)$!

Transformation: The plane π can be transformed by the transformation matrix H as shown in the following equation: $\pi' = H^{-T}\pi$.

1.5 ideal points, lines and planes:

1.5.1 ideal points:

In p^2 the intersection of two parallel lines $L \sim (a, b, c)^T$ and $L' \sim (a, b, c')^T$ is $(c' - c)(b, -a, 0)^T$ and is not representable in inhomogeneous coordinates. It is an ideal point and lies at infinity, which is what we would expect from intersecting parallel lines (figure 2). In p^3 an ideal point p is defined as $p \sim (a, b, c, 0)$ as mentioned above.

1.5.2 ideal lines:

In p^2 All ideal points are lying on a line, called the line at infinity, and its

canonical position is $L_\infty \sim (0, 0, 1)$. We verify that $(0, 0, 1)^T(x_1, x_2, 0) = 0$.

Remember: in p^2 The line at infinity is represented by a plane which doesn't intersect the projective plane. This plane is the xy -plane (figure 2)!

1.5.3 ideal planes:

The plane at infinity has the canonical form $\pi_\infty \sim (0, 0, 0, 1)^T$ in an affine coordinate frame. It contains all directions $D \sim (x_1, x_2, x_3, 0)^T$ that means parallel entities intersect on the plane at infinity. For example all lines parallel to each other intersect in a single point on the plane at infinity, and all planes parallel to each other intersect in a single line at the plane at infinity.

2 Transformations in general:

Remember: We can transform a point by multiplying the point with a transformation matrix as mentioned above!

2.1 Translation:

We intend to translate a point to a new place on the xy -plane (in 2D) by adding values to the x and y coordinates of this point which is clear in the following equation:

$$\begin{pmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = \begin{pmatrix} x_1' \\ x_2' \\ 1 \end{pmatrix}.$$

or in space (in D3) by adding values to the x, y and z coordinates of this point

$$\begin{pmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} x_1' \\ x_2' \\ x_3' \\ 1 \end{pmatrix}.$$

Translation has two degrees of freedom in 2D and three in 3D depending on the number of the independent variables in the translation matrix.

2.2 Scale:

Scaling transformations stretch or shrink a given object and as a result the

lengths and angles will be changed. The meaning of scaling is to make the new scale of a coordinate direction D times larger or smaller. It can be applied to all axis, each with a different scaling factor as shown here:

$$\begin{pmatrix} p & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{pmatrix}.$$

2.3 Rotation:

if a point $(x_1, x_2, 1)$ is rotated an angle α about the coordinate origin it becomes a new point $(x'_1, x'_2, 1)$, the relationship can be described as follows:

$$\begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{pmatrix}.$$

Rotation in space are more complex, because we can either rotate about the x-axis, the y-axis or the z-axis. When rotating about the z-axis, only coordinates of x and y will change and the z-coordinate will be the same.

The rotating equation about the $z - axis$ an angle α is:

$$\begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{pmatrix}.$$

let $\alpha = 90^\circ$ the x-axis rotates to the y-axis and the y-axis rotates to the negative direction of the x-axis.

Based on the same idea, rotating about the $x - axis$ an angle α is the following:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{pmatrix}.$$

let $\alpha = 90^\circ$ the y-axis rotates to the z-axis and the z-axis rotates to the negative direction of the y-axis.

Let $\alpha = 90^\circ$ again. Rotating about the y-axis is different! it is because the way of measuring angles. In a right handed-system, if your right hand holds a coordinate axis with your thumb pointing to the positive direction, your other four fingers give the positive direction of angle measuring. More precisely, the positive direction for measuring angles is from the z-axis to x-axis. However, traditionally the angle measure is from the x-axis to the z-axis. As a result rotating an angle α about the y-axis in the sense of a right-handed system is equivalent to rotating an angle $-\alpha$ measuring from the x-axis to the z-axis. Therefore, the rotation equations are:

$$\begin{pmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} x_1' \\ x_2' \\ x_3' \\ 1 \end{pmatrix}.$$

2.4 Shear:

The effect of shear transformation looks like pushing a geometric object in a direction that is parallel to a coordinate plane (3D) or a coordinate axis (2D)(figure 3).

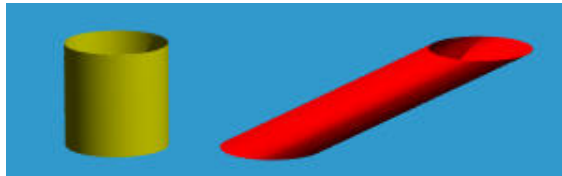


Figure 3: Under shearing the green cylinder becomes the red cylinder.

How far a direction is pushed is determined by its shearing factor. On the xy-plane we can push in the x-direction, positive or negative, and keep the y-direction unchanged.

$$\begin{pmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = \begin{pmatrix} x_1' \\ x_2' \\ 1 \end{pmatrix}.$$

Or push in the y-direction and keep the x-direction unchanged.

$$\begin{pmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = \begin{pmatrix} x_1' \\ x_2' \\ 1 \end{pmatrix}.$$

In the space we can push in the xy-direction and keep the z-direction unchanged.

$$\begin{pmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{pmatrix}.$$

Or we can push in the xz-direction and keep the y-direction unchanged.

$$\begin{pmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{pmatrix}.$$

Or we can push in the yz-direction and keep the x-direction unchanged.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ b & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{pmatrix}.$$

2.5 Combination of transformations:

In many cases, we may need several transformations to bring an object to its desired position these several transformations can be summarized into a single transformation represented by the product of all involved matrices.

For example, we may need a transformation $q = Ap$ bringing p to q followed by a second transformation $r = Bq$ bringing q to r , followed by yet another transformation $s = Cr$ bringing r to s these all transformation can be summarized in the following transformation $s = CBAp = Dp$.

Remember: A, B, C, D are all transformation matrices!

3 Hierarchy of transformations:

3.1 Euclidean transformations:

Euclidean transformation is a Rotation followed by a translation. This transformation preserves length, angles and Area in 2D and volume in 3D (figure 4). There are 3 degrees of freedom in 2D and 4 degrees of freedom in 3D which depends of the number of the independent variables in the

transformation matrix. The corresponding matrices of euclidean transformations are:

$$\text{in general } \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}, \text{ in } 2D \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ 0 & 0 & 1 \end{pmatrix} \text{ and in } 3D \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Remember: R is a Rotation matrix in inhomogeneous form!

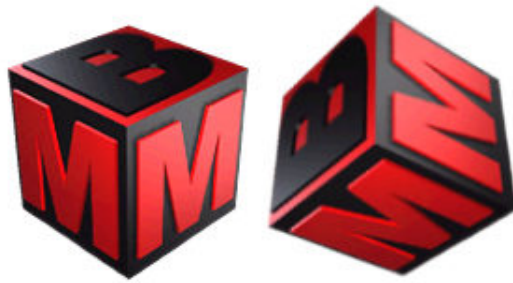


Figure 4: Euclidean transformation in 3D

3.2 Similarity transformations :

These are Euclidean transformations (translation and rotation) with an additional isotropic scaling. Normally this is the highest level of structure we can achieve in a reconstruction where no absolute measure is available.

Similarity transformations preserve colinearity, ratio of lengths and Area but not absolute lengths (figure 5).

In homogeneous coordinates this transformation can be expressed by a single matrix multiplication. The corresponding transformation matrix is:

$$\text{in general } \begin{pmatrix} SR & T \\ 0 & 1 \end{pmatrix}, \text{ in } 2D \begin{pmatrix} sr_{11} & sr_{12} & t_1 \\ sr_{21} & sr_{22} & t_2 \\ 0 & 0 & 1 \end{pmatrix} \text{ and in } 3D \begin{pmatrix} sr_{11} & sr_{12} & sr_{13} & t_1 \\ sr_{21} & sr_{22} & sr_{23} & t_2 \\ sr_{31} & sr_{32} & sr_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

There are 4 degrees of freedom in 2D and 7 degrees of freedom in 3D. These are 3 for orientation, 3 for translation and 1 for scaling.

Remember: R is a Rotation matrix in inhomogeneous form!



Figure 5: Similarity transformation in 3D

3.3 Affine transformations:

Affine transformation is a non-singular linear transformation followed by a translation. This can be compactly written by one single matrix multiplication, where the transformation matrix is:

$$\text{in general } \begin{pmatrix} A & T \\ 0 & 1 \end{pmatrix}, \text{ in 2D } \begin{pmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{pmatrix} \text{ and in 3D } \begin{pmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ a_{31} & a_{32} & a_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

This counts to 5 and 12 degrees of freedom for the 2D and 3D cases respectively. We distinguish further two types of affine transformations, orientation-preserving and orientation-reversing transformations. Algebraically characterized by $\det A > 0$ for orientation preserving and $\det A < 0$ for orientation-reversing.

The invariants of an affine transformation are parallelism, ratio of areas and volumes, ratio of length along a certain direction, and linear combinations of vectors (e.g. centroids)(figure 6). The line and the plane at infinity are preserved in affine transformations. It can easily be verified that a 2D affine transformation leaves the line at infinity in its canonical representation $L_\infty \sim (0, 0, 1)^T$ unchanged as shown in the following equation: $L_\infty = H^{-T}L_\infty$. Analogously the plane at infinity $\pi \sim (0, 0, 0, 1)^T$ is unchanged under a 3D affine transformation. The inversion is also true, the line and the plane at infinity are preserved only under affine transformations.



Figure 6: Affine transformation preserved parallelism

3.4 Projective transformations:

The projective transformation is the most general transformation, allowing the full range of projective transformations. Projective transformation in n -space can be obtained by multiplication with a real invertible $(n + 1) \times (n + 1)$ matrix H which is defined up to scale. This results in $((n + 1)(n + 1)) - 1$ degrees of freedom. For the 2D (3D) case we have a 3×3 (4×4) invertible matrix with 8 (15) degrees of freedom.

The projective transformation has the largest group of transformations associated with it therefore it has the least number of invariants. They are collinearity, incidence, intersection, tangency, and cross ratio (figure 7).

The projective transformation matrix is:

$$\text{in 2D } \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \text{ and in 3D } \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & 1 \end{pmatrix}.$$



Figure 7: projective transformation preserved colinearity

4 Application:

4.1 A Robot has to go on the street:

We built a robot which has to go on the road. We paint large squares on the road to make it easier for the robot. The question is how to find the road shape without perspective distortion from the image?

We have to find the matrix of projectivity which has 8 degrees of freedom (Figure 8). The mapping between 2 images can be computed if we have the coordinates of 4 points on one image, and know where they are mapped in the other image. Each point of the one image provides 2 independent equations (Figure 9), therefore we can compute the unknown 8 variables of the matrix of projectivity.

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix}.$$

Figure 8: shows The multiplication of a point with the matrix of projectivity which provides a new transformed point.

$$x'_1 = \frac{x_1}{x_3} = \frac{h_{11}x_1 + h_{12}x_2 + h_{13}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3}.$$

$$x'_2 = \frac{x_2}{x_1} = \frac{h_{21}x_1 + h_{22}x_2 + h_{23}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3}.$$

Figure 9: every point provides two independent equation.

The corners of squares provide 4 points. The coordinates of These 4 points allow us to compute the matrix of projectivity. Then we use the inverse of the matrix of projectivity to compute road shape from Birds-eye-view. With this method we can change a point at infinity in a normal point which is impossible in the school geometry (Figure 10 and 11).

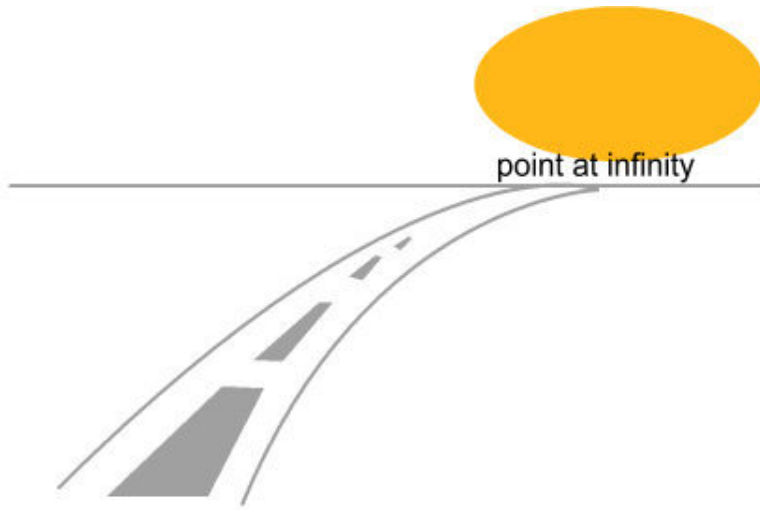


Figure 10: The road in robot perspective.

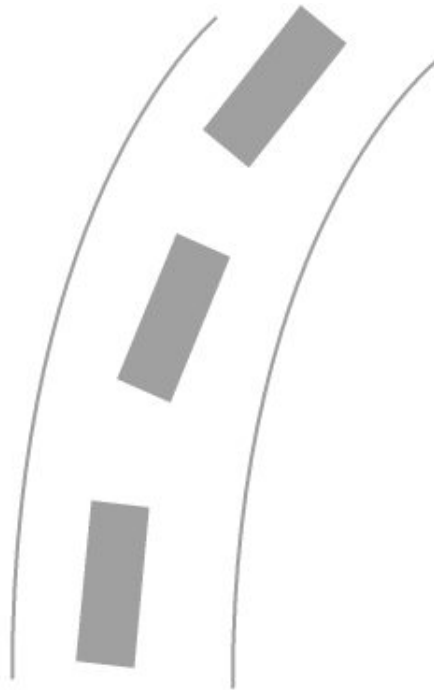


Figure 11: The road in Birds-eye-view.

4.2 Image registration:

We have two different image data A and B of the same patient which contain complementary information taken at two different times by two different methods which provide 3D image data of the body. The image A is provided by a MRI and B by a fMRI.

The question is how to determine possible Changes of the patient's state between the two points of time?

We will use 3D-3D image registration to solve this problem. This means to bring these two images into spatial alignment, or to establish a common geometric reference frame. Therefore we define a transformation T on one of these two images e.g. on B , and compare A with TB using the full image content. We refine T and repeat the procedure until convergence criteria reached. This procedure can be achieved by using image registration algorithms. The transformation T is a 4×4 matrix which provides euclidean or Similarity transformation what means that we just have to rotate, translate and maybe scale one of the images which depends of the position of the patient in the medical device at the two points of time (figure 12 and figure 13).

$$\begin{aligned}
 T_P &= T(t_x, t_y, t_z) \cdot R_z(\theta_z) \cdot R_y(\theta_y) \cdot R_x(\theta_x) \\
 &= \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_z & -s_z & 0 & 0 \\ s_z & c_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_y & 0 & s_y & 0 \\ 0 & 1 & 0 & 0 \\ -s_y & 0 & c_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_x & -s_x & 0 \\ 0 & s_x & c_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 &= \begin{pmatrix} c_y c_z & (s_x s_y c_z - c_x s_z) & (c_x s_y c_z + s_x s_z) & t_x \\ c_y s_z & (s_x s_y s_z + c_x c_z) & (c_x s_y s_z - s_x c_z) & t_y \\ -s_y & s_x c_y & c_x c_y & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 & \quad s_x = \sin(\theta_x), c_x = \cos(\theta_x), s_y = \sin(\theta_y), c_y = \cos(\theta_y), s_z = \sin(\theta_z), c_z = \cos(\theta_z)
 \end{aligned}$$

Figure 12: The transformation matrix T which represents an euclidean transformation is a combination of translation and rotation about all axis.

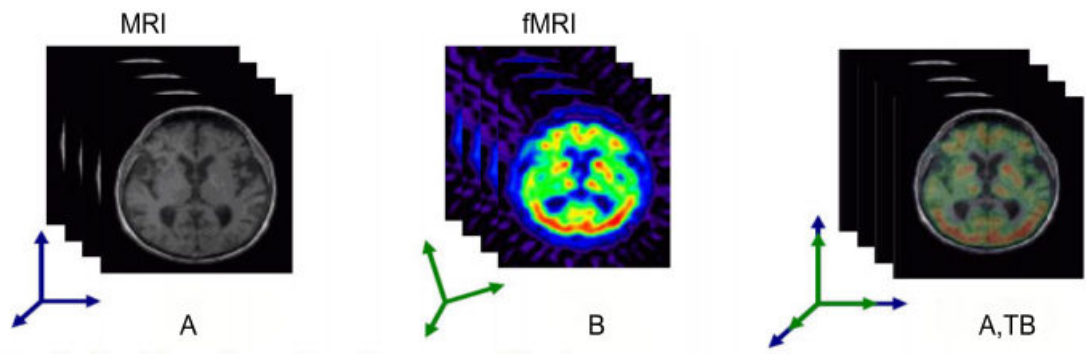


Figure 13: Image registration in 3D