Lecture 04/06

April 6, 2020

1 Transformations in homogenous coordinates

In the previous lecture we discussed homogenous coordinates, and showed how several transformations can be expressed as linear transformations in homogenous coordinates. In particular, we showed the following:

1. **Rotations**: In Euclidean coordinates, rotation is represented as $\mathbf{x}' = R\mathbf{x}$. In homogenous coordinates, rotation is represented as $\vec{\mathbf{x}'} \equiv \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \vec{\mathbf{x}}$.

2. **Translations**: In Euclidean coordinates, translation is represented as $\mathbf{x}' = \mathbf{x} + \mathbf{t}$. In homogenous coordinates, this is $\vec{\mathbf{x}'} \equiv \begin{bmatrix} I & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \vec{\mathbf{x}}$.

However, these are not the only transformations that we are interested in. Here are some more:

1. **Linear transformations in Euclidean coordinates** are performed through a matrix multiplication: $\mathbf{x}' = M\mathbf{x}$. In homogenous coordinates, such transformations are represented in a manner similar to rotation: $\vec{\mathbf{x}'} \equiv \begin{bmatrix} M & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \vec{\mathbf{x}}$

2. **Euclidean transformations**, also called **Rigid transformations** are a combination of rotations and translations: $\mathbf{x}' = R\mathbf{x} + \mathbf{t}$. In homogenous coordinates, such transformations are expressed as: $\vec{\mathbf{x}'} \equiv \begin{bmatrix} I & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \vec{\mathbf{x}} \equiv \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \vec{\mathbf{x}}$.

3. **Similarity transformations** involve a global scaling in addition to rotations and translations: $\mathbf{x}' = \alpha R\mathbf{x} + \mathbf{t}$. Here α is a scalar. In homogenous coordinates, similarity transformations are similar to Euclidean transformations : $\vec{\mathbf{x}'} \equiv \begin{bmatrix} \alpha R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \vec{\mathbf{x}}$.

4. **Affine transformations** are even more general than similarity transformations, and combine a general linear transformation with a translation: $\mathbf{x}' = M\mathbf{x} + \mathbf{t}$. These are represented in homogenous coordinates as $\vec{\mathbf{x}'} \equiv \begin{bmatrix} M & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \vec{\mathbf{x}}$.

Finally, **projective** transformations are the class of transformations that can be expressed as a linear transformation in homogenous coordinates. They include all the transformations above, and then some. Figure 1 shows some transformations.

2 Figuring out R and $\mathbf{t}$ for a camera

As discussed in the previous lecture, the action of a pinhole camera can be written as:

$$\vec{\mathbf{q}} \equiv K \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \vec{\mathbf{Q}} \tag{1}$$

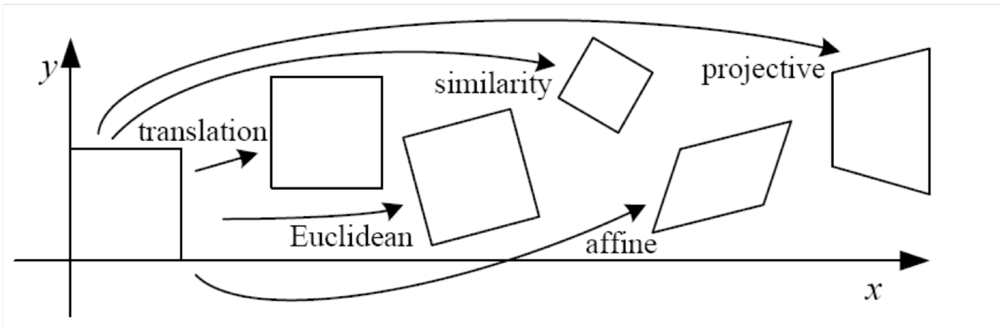

Figure 1: Different kinds of transformations. All of these can be expressed as a matrix multiplication in homogenous coordinates.

R and $\mathbf{t}$ depend on the location and orientation of the camera. But how can we figure it out?

Suppose that in the world coordinate system, the camera is at location $\mathbf{c}$. As for the orientation of the camera, suppose the viewing direction of the camera (it's internal Z axis) is along the vector $\mathbf{n}_z$. The X and Y axes of the camera are along the vectors $\mathbf{n}_x$ and $\mathbf{n}_y$ respectively. Given this information, what should R and t be?

Let us figure out R first. Here R is intended to align the world coordinate axes with the camera coordinate axes. Thus, the ccamera viewing direction $\mathbf{n}_z$ after rotation by R should become the Z axis, $\begin{bmatrix}0\\0\\1\end{bmatrix}$. Similarly, $\mathbf{n}_x$ and $\mathbf{n}_y$ after rotation R should align with X and Y axes. This gives:

$$R\mathbf{n}_x = \begin{bmatrix}1\\0\\0\end{bmatrix} \tag{2}$$

$$R\mathbf{n}_y = \begin{bmatrix}0\\1\\0\end{bmatrix} \tag{3}$$

$$R\mathbf{n}_z = \begin{bmatrix}0\\0\\1\end{bmatrix} \tag{4}$$

$$\Rightarrow R\begin{bmatrix}\mathbf{n}_x & \mathbf{n}_y & \mathbf{n}_z\end{bmatrix} = I \tag{5}$$

$$\tag{6}$$

where the last line just stacks the previous equations together, and I is the identity.

Denote the matrix $\begin{bmatrix}\mathbf{n}_x & \mathbf{n}_y & \mathbf{n}_z\end{bmatrix}$ as R'. Observe that the columns of R' are unit length and are perpendicular to each other (because the axes of the camera are perpendicular to each other). Thus R' is an orthonormal matrix and $R'^{-1} = R'^T$.

Thus:

$$RR' = I \tag{7}$$

$$\Rightarrow R = R'^T \tag{8}$$

.

Next let us figure out the translation $\mathbf{t}$. It must be the case that after rotation and translation, the camera center is at the origin. Thus:

$$R\mathbf{c} + \mathbf{t} = \mathbf{0} \tag{9}$$

$$\Rightarrow \mathbf{t} = -R\mathbf{c} \tag{10}$$

Thus, given a camera at location $\mathbf{c}$ and with axes pointed along $\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z$ in the world coordinate frame, R and $\mathbf{t}$ should be given by:

$$\boxed{\begin{aligned} R &= \begin{bmatrix} \mathbf{n}_x & \mathbf{n}_y & \mathbf{n}_z \end{bmatrix}^T && (11) \\ \mathbf{t} &= -R\mathbf{c} && (12) \end{aligned}}$$

3 Camera calibration

What if we don't know where the camera is ? How can we find out the parameters of the camera?

The problem of estimating the unknown camera parameters K, R and $\mathbf{t}$, from an image is called **camera calibration**. This is essential for at least two applications:

1. As can be seen above, R and $\mathbf{t}$ are related to where the camera is in the world and how it is oriented. Knowing R and $\mathbf{t}$ thus tells us where the camera is and where it is looking. Alternatively, it tells us how objects in the world are oriented and located relative to the camera.

2. Knowing the camera parameters allows us to "render" new objects into the scene. This would be useful in augmented reality/virtual reality.

To estimate the parameters, let us rewrite the projection equation as follows:

$$\vec{\mathbf{q}} \equiv P\vec{\mathbf{Q}} \tag{13}$$

where $P = K\begin{bmatrix} R & \mathbf{t} \end{bmatrix}$. We will first estimate P, then decompose it into K, R and $\mathbf{t}$.

To estimate P, we assume that we know a few world points $\vec{\mathbf{Q}}_1, \ldots, \vec{\mathbf{Q}}_n$ and their corresponding image positions $\vec{\mathbf{q}}_1, \ldots, \vec{\mathbf{q}}_n$. Thus, we have a set of equations:

$$\vec{\mathbf{q}}_i \equiv P\vec{\mathbf{Q}}_i \quad i = 1, \ldots, n \tag{14}$$

Note that if P is a solution to these equations, αP is also a solution for $\alpha \neq 0$. This is because $P\vec{\mathbf{Q}}_i \equiv \alpha P\vec{\mathbf{Q}}_i$ (multiplying by a scalar in homogenous coordinates has no effect). Thus this set of equations will not identify a single solution, but in fact a *family* of solutions, all scalar multiples of each other.

To isolate one solution from this family, we need a way to fix the scale factor. We add the following constraint:

$$\|P\|_F = 1 \tag{15}$$

where $\|P\|_F = \sqrt{P_{11}^2 + P_{12}^2 + \ldots + P_{34}^2}$ is the Frobenius norm of P. Thus we now have the following system:

$$\begin{aligned} \vec{\mathbf{q}}_i &\equiv P\vec{\mathbf{Q}}_i \quad i = 1, \ldots, n && (16) \\ \|P\|_F &= 1 && (17) \end{aligned}$$

How do we deal with the first set of equivalences? Let us consider the first equivalence:

$$\vec{\mathbf{q}}_1 \equiv P\vec{\mathbf{Q}}_1 \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \equiv P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{18}$$

where $\vec{\mathbf{q}}_1 \equiv \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ and $\vec{\mathbf{Q}}_1 \equiv \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$. The equivalence means that the left hand side is a scaled copy of the right hand side. Thus, there exists $\lambda \neq 0$ such that:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{19}$$

Multiplying the matrices out, we get:

$$\lambda x = P_{11}X + P_{12}Y + P_{13}Z + P_{14} \tag{20}$$
$$\lambda y = P_{21}X + P_{22}Y + P_{23}Z + P_{24} \tag{21}$$
$$\lambda = P_{31}X + P_{32}Y + P_{33}Z + P_{34} \tag{22}$$

Substituting the last in the other two we get:

$$(P_{31}X + P_{32}Y + P_{33}Z + P_{34})x = P_{11}X + P_{12}Y + P_{13}Z + P_{14} \tag{23}$$
$$(P_{31}X + P_{32}Y + P_{33}Z + P_{34})y = P_{21}X + P_{22}Y + P_{23}Z + P_{24} \tag{24}$$

Observe that these are *two equations*, and they are *linear in the unknowns* (the entries of P). Thus, these equivalences lead to a system of linear equations! Writing out the entries of P as a vector of unknowns $\mathbf{p} = \begin{bmatrix} P_{11} \\ P_{12} \\ \vdots \\ P_{34} \end{bmatrix}$, we can thus write out this system of linear equations in matrix form:

$$A\mathbf{p} = \mathbf{0} \tag{25}$$

A is a matrix of coefficients (**write down what they are!**)

Together with the Frobenius norm constraint, we have:

$$A\mathbf{p} = \mathbf{0} \tag{26}$$
$$\|\mathbf{p}\|_2 = 1 \tag{27}$$

where the last line refers to the $L2$ norm. Usually, there might be some noise in the image locations $\mathbf{q}_i$, so we don't want to solve these equations exactly. Instead, we try to solve:

$$\min_{\mathbf{p}} \|A\mathbf{p}\|_2 \tag{28}$$
$$\text{such that} \tag{29}$$
$$\|\mathbf{p}\|_2 = 1 \tag{30}$$

In the next lecture we will solve this optimization.