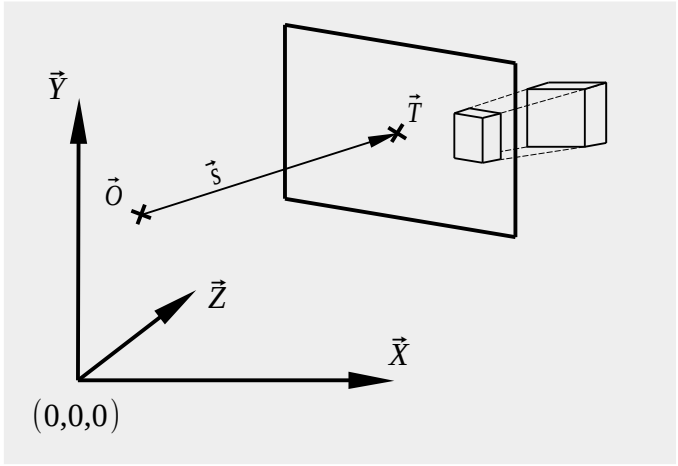
Obtain observer CS*

How can we project the 3D world onto the screen from an arbitrary perspective?

Define a virtual observer O and a virtual target T. We want the human viewer in-front of the screen to see the world as it is observed by O.



It is difficult to rotate the real viewer. Instead we rotate the virtual world to achieve the same effect.

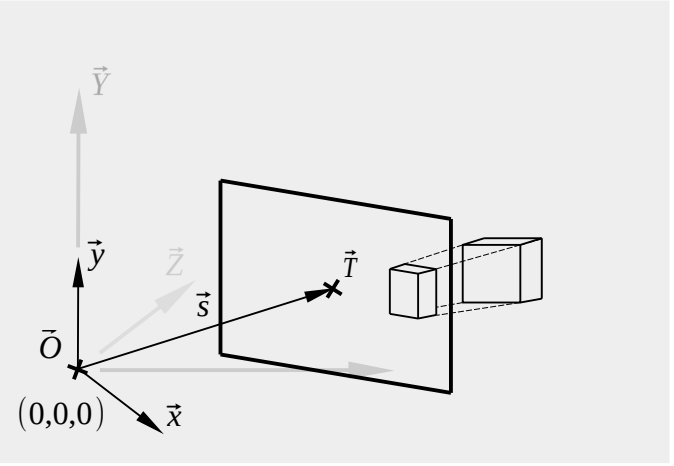
But how can the virtual world be rotated about the observer? So far all rotations were revolving about the origin (0,0,0).

Solution: Move the observer into the origin. Perform the same translation with the target and with any virtual object. The change is not noticeable from the observer's perspective but now we can conveniently rotate around the origin.

*CS = coordinate system

Obtain observer CS

Now we need to determine the local coordinate system of the viewer which will hopefully lead to the correct rotation of the world, as it is seen from the observer perspective.

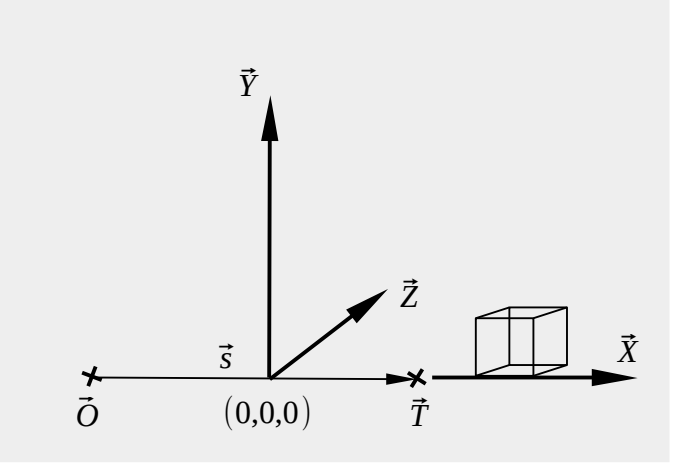


global axes are denoted with capital letter, local axes with small letters

Preliminary example

Observer O looks right into the direction of the X-axis of the world coordinate system. So his local z-axis corresponds to the global X-axis. Accordingly his local x-axis is -Z and his local y-axis is the same as the global one. As column-vectors we can write:

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

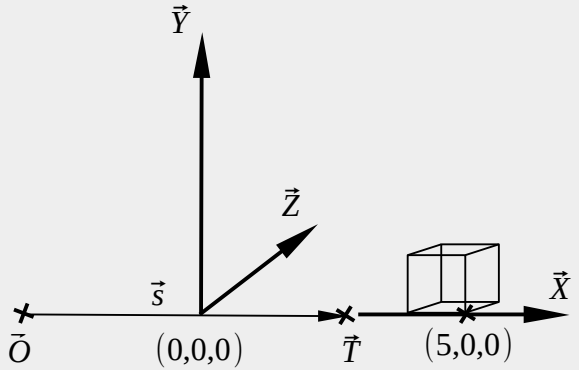


Obtain observer CS

Preliminary example

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 5 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -5 \end{pmatrix}$$

We rotate one vertex (5,0,0) of the cube by the matrix



and realize that the point is behind us. However, we expected (5,0,0) to be in front of us.

Conclusion: We rotate into the wrong direction and must thus invert the matrix. In the case of rotations we are lucky, because the transposed matrix is equal to the inverse one.

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}^T = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

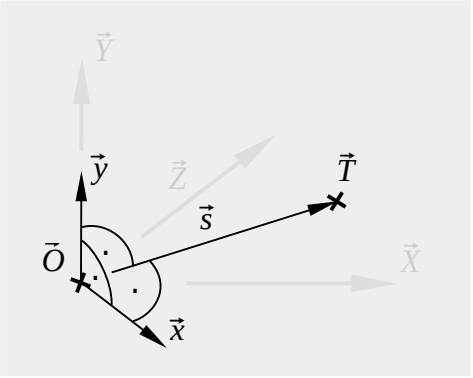
$$\begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 5 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 5 \end{pmatrix}$$

Obtain observer CS

General case

This step will try to evaluate the local x-, y-, and z-axis of the observer.

$$\begin{pmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{pmatrix}$$



The z-axis is obviously the sight-vector s between O and T: $\vec{s} = \vec{T} - \vec{O}$

$$\begin{pmatrix} x_x & y_x & s_x \\ x_y & y_y & s_y \\ x_z & y_z & s_z \end{pmatrix}$$

There are another six unknown variables. Theoretically, the problem can be solved finding six constraints.

However, it is easier to use the cross-product:

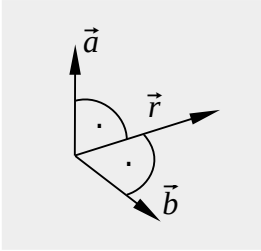
Obtain observer CS

Simple solution

We use the so-called cross-product. In 3D (and only in 3D) it produces a vector r perpendicular to another two vectors a and b:

The cross-product

The cross product of vector a x b results in a vector r which is perpendicular to a and b. It is defined as:



$$\vec{r} = \vec{a} \, x \, \vec{b} = \begin{pmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{pmatrix}$$

Note that r can be flipped into the opposite direction and will still be perpendicular to a and b. We do not yet now which version of r we obtain.

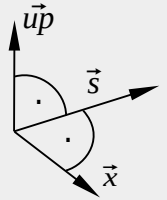
Obtain observer CS

Simple solution

Let us try to create the x-axis using the cross product. We need two vectors. So far we only have the sight vector s. Where do we get a meaningful second sector from?

Solution: We invent an up vector. For simplicity we use the global Y-axis as up.

$$\vec{x} = \vec{s} \, x \, \vec{up}$$



Now the Y-axis of our personal coordinate system is missing. Note: This is not necessarily equal to the global Y-axis or the up-vector.

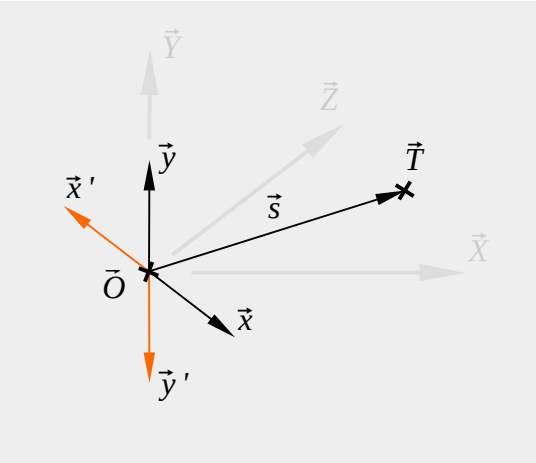
$$\vec{y} = \vec{x} \, x \, \vec{s}$$

Obtain observer CS

Distinction of cases

Our calculation can yield two different results for the x-axis. In fact we can multiply it by (-1) without violating any of the constraints.

But which case is the one sought after?



Choice for the correct x-axis

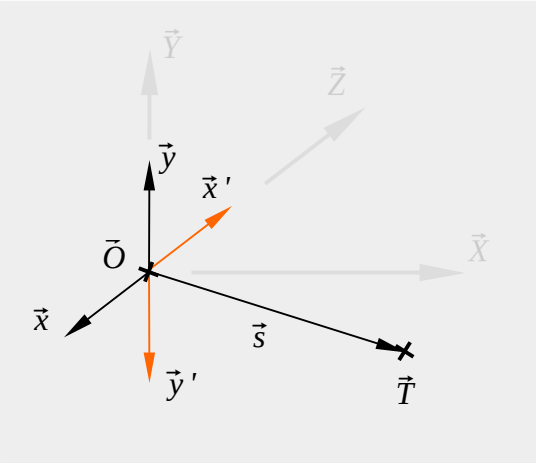
In the sketch the black x-axis is the correct one because it points to the right. More technically it holds true that x.x > 0.

This case applies only if our sight vector s points into the same half-space as the global Z-axis does. In other words, it applies if s.z > 0.

Obtain observer CS

Distinction of cases

Choice for the correct x-axis



Here the sight vector s points into the opposite half-space as the global Z-axis. In other words, s.z < 0. Here we are only happy if the local x-axis points to the left.

More technically: x.x < 0 is ok.

Otherwise the local x-axis gets swapped by 180° by multiplying it with (-1).

Choice for the correct y-axis

Again, two directions are possible. Choose the one which points into the direction of our up-vector.