# VISIÓN POR COMPUTADOR

**Exercise 9: Camera model.**

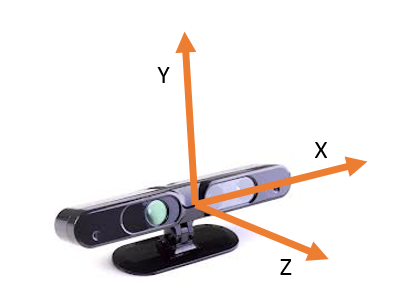
Concepts: Perspective projection matrix, spatial transformation.

1. **Homogeneous transformations:** Given a camera located at the origin of coordinates, obtain the transformations matrices (4 x 4) to:

* Move the camera 0.5 meters forward.
* Rotate the camera 35º to look to the left (yaw rotation).
* Move the camera 0.5 meters upwards and rotate it downwards (to observe the same object from a higher perspective).

Show the original and the transformed reference frames with *showTransformation(T)*, where *T* is the transformation matrix.

The reference frame of the camera is defined as follows:



1. **Camera-to-world transformation**: In this exercise we will work with images provided with an RGB-D camera. These cameras provide standard RGB images together with depth images which measure how far is the point observed by each pixel. Thus, if the intrinsic parameters are known (matrix K), we can compute the 3D coordinates of the observed points. To do that, you have to:

* Read the RGB and depth images provided and show them (you will have to re-scale the depth image to visualize it properly). To read the depth image use “imread” and divide the resulting matrix by 5000.
* Obtain the 3D coordinates of the observed points (relative to the reference frame of the camera), assuming that and .
* Plot the 3D points with their corresponding color with “plot3DScene(X,Y,Z,image\_RGB, downsample)”.

|  |  |  |
| --- | --- | --- |
|  |  |  |

1. **World-to-camera transformation:** Now we will simulate views of the same scene from different perspectives. To that end, you have to define the homogeneous transformation associated to these new perspectives and use it, together with K, to create the new images. Generate at least two images from different perspectives moving the camera around the observed points. Take into account that these images will only look “reasonable” if the camera does not get very far from its original location.

Use the function “renderNewImage(x\_proj, y\_proj, depth\_transformed, image\_RGB)”.

**Commands:**

|  |  |
| --- | --- |
| **showTransformation(T)** | Creates a visualization of a ref. frame at the origin (blue) and transformed (red) according to T (4x4). |
| **plot3DScene(X,Y,Z,image\_RGB,downsample)** | Shows a coloured 3D point cloud. X,Y,Z are matrices with the spatial coordinates of the points, “image RGB” is the colour image and “downsample” is used to reduce the number of points to draw (use downsample ~= 4). |
| **renderNewImage(x\_proj, y\_proj, depth\_transformed, image\_RGB)** | Creates a new image by rendering the scene from a different perspective. “x\_proj” and “y\_proj” are the pixel coordinates of the scene points after transforming and projecting them, “depth\_transformed” is the depth after transforming them and “image\_RGB” is the original colour image. |