# *Enhanced visualization: assignment 3*
## *Image formation*

## *I. Theoretical part*

*A. Vanishing points.*

1. Show that the projections of parallel lines which are not parallel to the image plane intersect.
2. What are the projections of parallel lines which are parallel to the image plane?
3. Show that sets of parallel lines from the same plane have vanishing points located in a line.

*B. Examples.*

1. A camera has a field of view of $90^\circ \times 90^\circ$ and image resolution of $400 \times 400$ pixels. The image center lies in the intersection between optical axis and the image plane.

   A point has coordinates $[1 \quad 2 \quad -8]$ m in camera coordinates. Find the pixel projection of the point in the image.

2. A camera views a square on a plane. The plane is parallel to the image plane.
   - Show that the width and height of the square in the image does not depend on its $x$ and $y$ coordinates.
   - Find the relationship between the width of the square and its position in the camera frame $z_c$ axis.
   - Can you measure the average speed of a camera moving forward in the direction of the square from the width in the image?

3. A CCD sensor is 10mm $\times$ 10mm and has 10 million sensor elements (pixels) covering uniformly its surface. The focal length of the camera is 6mm what is the field of view of one pixel in the center of the CCD sensor? This field of view is called the instantaneous field of view (iFOV) for one pixel.

## *II. Practical part*

In this part you are going to simulate the image formation process in Python.

1. Write a python function which implements the image formation process. More precisely write a function

   ```
   im_buffer, X_im = im_proj(X,K_wc,f,s,o,pix)
   ```

   where

   - **X** is the input matrix of 3D points coordinates, each column of the matrix specifies a point in Cartesian coordinates.
   - $\mathbf{K}_{cw}$ is the matrix describing the transformation from camera frame to world frame.
   - $f$ is the focal length of the camera (pinhole model focal length).
   - $s$ is the vector of camera key scale parameters $s_x$, $s_y$, $s_\theta$ in pixels/m.
   - $o$ is the position vector of the origin of the image plane with respect to the origin of the image buffer.
   - pix is a vector with the number of pixels in the horizontal and vertical directions.
   - $\mathrm{X_{im}}$ are the 2D coordinates of the points projected in the image buffer. Remember that these coordinates are integers.
   - $\mathrm{im_{buffer}}$ is the image buffer, it is a matrix of sizes given by pix. We will assume that when a point is observed in a given position its amplitude in the image buffer is 255. Note that points outside the image buffer cannot be displayed in the image buffer and should be discarded.

   You can fill the gaps in the code given in the Jupyter notebook "assignment_3.ipynb" which you can download from *moodle*. The gaps are indicated by " ************* ".

2. A camera $C$ is positioned at $[0 \quad 0 \quad 6]^T$ u.s. looking in the negative direction of the $z$-axis. Assume a pinhole camera model with $fs_x = fs_y = 600$ pixels ($s_\theta = 0$) and that the image size is 640 pixels wide and 480 pixels high. The origin of the image buffer is on the top-left corner of the image plane as usual.

   A cube has vertices in the world reference frame given by the columns of the matrix $X$[1]. Generate the image buffer of the camera representing the view of the vertices of the cube. Does it look like a realistic view of the 3D object? If not, explain why.

[1] $X = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$ u.s.

3. A vehicle $V$ is positioned at $[6 \quad 1 \quad 8]^{\mathrm{T}}$ u.s. with respect to world reference, where u.s. means unities of space. It is rotated by 30° about the $y$ axis, see Figure 1 for a top-down view.

   The camera $C$ is mounted on a rotational mount $M$ on the top of the vehicle. The mount is positioned at distance $= 3$ u.s. above the vehicle reference frame and it is tilted down by 30°. The camera is attached to the mount and it is positioned above it at a distance $= 1.5$ u.s.. See Figure 2 for a side view of the vehicle.

   Generate the corresponding image frame buffer representing the view of the vertices of the cube.

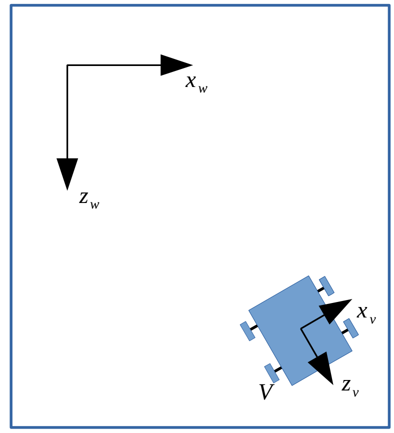


Figure 1: Position of the vehicle with respect to world frame.

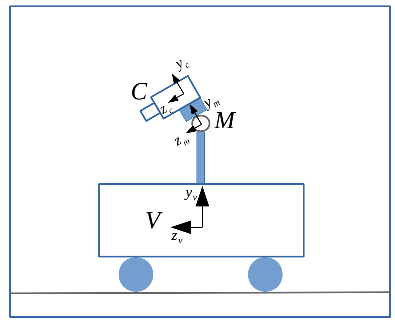


Figure 2: Position of the mount and the camera on top of the vehicle.