Image motion seen by moving camera

Let's next consider what happens when the viewer moves. The motion can be either a translation or a rotation (or both). Motion causes the viewer to see the scene from different 3D positions and in different directions, and the result is that scene points project to different positions in the image. Later in the course we will see that if a vision system can measure the changes in image positions over time (in Part 2 of the course), then it is possible to compute the 3D positions of the points in the scene *from the changing image positions alone* (in Part 3 of the course).

Translating the viewer

We begin by looking at translation. Suppose the camera translates with 3D velocity (T_x, T_y, T_z). For example, forward camera motion is 3D velocity $(0, 0, 1)$. Rightward camera motion is 3D velocity $(1, 0, 0)$. Upward camera motion is 3D velocity $(0, 1, 0)$. When the camera translates, the position of any visible point varies over time. In the camera's coordinate system, the position of the point moves in the 3D direction and speed opposite to the camera. If the camera coordinates of a point at time $t = 0$ are (X_0, Y_0, Z_0), then at time t the point will be at $(X_0 - T_x t,\ Y_0 - T_y\ t,\ Z_0 - T_z\ t)$ in camera coordinates.

Now let's project the 3D point into the image plane. How does the image position of this point in the image vary with time? The image coordinate of the point is a function of t, namely,

$$(x(t), y(t)) \ = \ (\ \frac{X_0 - T_x\ t}{Z_0 - T_z\ t}, \frac{Y_0 - T_y\ t}{Z_0 - T_z\ t})\ f$$

Taking the derivative with respect to t at $t = 0$ yields an *image velocity vector* (v_x, v_y) :

$$(v_x, v_y) = \frac{d}{dt}(x(t), y(t))\ |_{t=0} \ = \ \frac{f}{{Z_0}^2}(-\ T_x\ Z_0\ +\ T_z\ X_0, -\ T_y\ Z_0\ +\ T_z\ Y_0). \qquad (1)$$

We will sometimes speak of the *motion field* (v_x, v_y) or *image velocity vector field* to be the 2D vector function, defined in the image plane. As we will see next, the velocity field depends on image position (x, y) and on the depth Z_0.

Lateral translation

Consider the case that $T_z = 0$. This means the camera is moving in a direction perpendicular to the optical axis. One often refers to this as *lateral motion.* It could be left/right motion, or up/down motion, or some combination of the two. Plugging $T_z = 0$ into the above equation yields:

$$(v_x, v_y) = \frac{f}{Z_0}(-T_x, \ \ -T_y) \ \ .$$

Note that the direction of the image velocity is the same for all points, and the magnitude (speed) depends on inverse depth.

Let's look at a few examples. Recall last class that if we have a plane

$$aX + bY + cZ = d$$

then

$$ax + by + cf = \frac{fd}{Z}$$

Dividing by d and then substituting f/Z into the velocity field expression above, we see that the image velocity field has a very simple form, namely

$$(v_x, v_y) = \frac{ax + by + cf}{d} \, (-T_x, \ -T_y).$$

A specific example is the case $T_x \neq 0$, but $T_y = T_z = 0$. The motion field corresponds to the camera pointing out the side window of the (passenger!) seat of the car, as the car drives forward. If we restrict the scene to be a single ground plane $Y = h$, you can observe from the ground plane equation from last class, the image velocity is

$$(v_x, v_y) = -\frac{T_x}{h}(y, 0).$$

This produces a *shear field*, where the x-velocity is 0 at $y = 0$ (the horizon) and increases linearly with y. You have seen this motion pattern many times in your life when looking out the side window of the car or train.

Forward translation

Next take the case of forward translation ($T_x = T_y = 0$ but $T_z > 0$). In this case Eq. (1) reduces to

$$(v_x, v_y) \ = \ \frac{T_z}{Z_0} \, (x \ , \ y) \tag{2}$$

You can verify that this field points away from the origin $(x, y) = (0, 0)$. Also, the image speed (the length of the velocity vector) is

- proportional to the image distance from the origin i.e. $|(x, y)|$,
- inversely proportional to the depth Z
- proportional to the forward speed of the camera T_z

In the case of a ground plane ($y \ = \ \frac{hf}{Z}$), we get

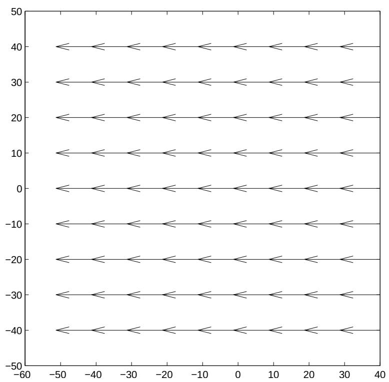
$$(v_x, v_y) \ = \ \frac{yT_z}{hf} \, (x \ , \ y) = \frac{T_z}{hf} \, (xy \ , \ y^2)$$

See figure below and to the right.

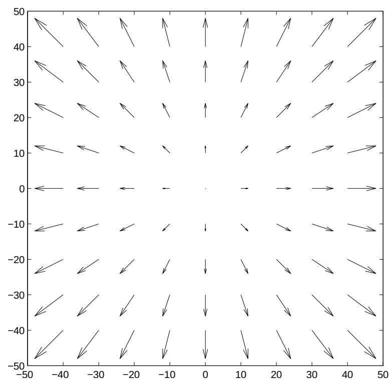
General translation

Returning to Eq. (1), let's take the case of a general non-lateral translation direction i.e. $T_z \neq 0$ and either T_x or T_y are not zero. We can easily rewrite Eq. (1) as:

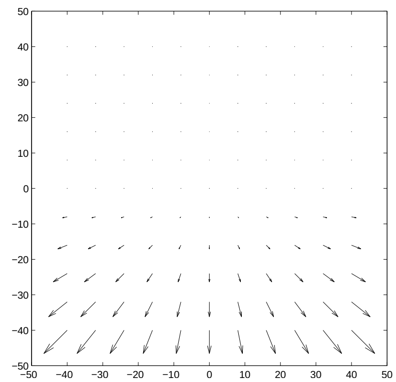
$$(v_x, v_y) \ = \ \frac{T_z}{Z_0} \, (x - f\frac{T_x}{T_z}, \ y - f\frac{T_y}{T_z})$$



$\mathbf{T} = (1, 0, 0)$
lateral + 'Z = constant'



$\mathbf{T} = (0, 0, 1)$
forward, and Z = constant



$\mathbf{T} = (0, 0, 1)$
forward and ground plane

There are a number of important properties of this general translation field. As we saw earlier, image speed is proportional to T_z and inversely proportional to depth. In addition, the image velocities point away from a particular image position,

$$(f\frac{T_x}{T_z},\ f\frac{T_y}{T_z})$$

which is sometimes called the *direction of heading*. Notice that the point $(x, y) = (f\frac{T_x}{T_z},\ f\frac{T_y}{T_z})$, is on the projection plane $Z = f$, and so we put back this third dimension, we get the vector $(f\frac{T_x}{T_z},\ f\frac{T_y}{T_z}, f)$ which is obviously parallel to (T_x, T_y, T_z), namely the direction of translation (heading). This should be no surprise...

Rotating the viewer

Often the camera will rotate as well as translate. For example, moving cameras may pan (side to side), pitch (up and down), or roll (rotate about optical axis).

Rotation about y axis

Consider first the case that we *rotate the camera coordinate system* around the Y axis, with an angular velocity of Ω radians per second. Any point (X_0, Y_0, Z_0) will sweep out a circle centered on the Y axis, when viewed in the camera's coordinate system. At time t the point would be at $(X(t), Y(t), Z(t))$,

$$\begin{bmatrix} X_0(t) \\ Y_0(t) \\ Z_0(t) \end{bmatrix} = \begin{bmatrix} \cos(\Omega t) & 0 & \sin(\Omega t) \\ 0 & 1 & 0 \\ -\sin(\Omega t) & 0 & \cos(\Omega t) \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix}$$

As in the translation case, we are interested in the velocity of the corresponding projected image point at time $t = 0$, i.e.

$$(v_x, v_y) = \frac{d}{dt}(x(t), y(t))|_{t=0}$$

So,

$$\begin{aligned} v_x &= f\frac{d}{dt}(\frac{X_0(t)}{Z_0(t)}) \\ &= f\frac{(\Omega Z_0^2 + \Omega X_0^2)}{Z_0^2}, \text{ since } \ \frac{d}{dt}X(t)|_{t=0} = \Omega Z_0, \ \ \frac{d}{dt}Z(t)|_{t=0} = \Omega X_0 \\ &= f\Omega(1 + (\frac{x}{f})^2) \end{aligned}$$

So, we see that the rotation gives us two components to the x velocity. There is a constant motion, plus there is a second order motion. The first term is intuitively easy to understand: when we pan the camera, we move all the points in the same direction. The second component is less obvious. It arises because we are projecting onto a plane. 3D points that project to large positive or negative x values will not have the same instantaneous x velocity on the projection plane, as points that are near the optical axis ($x \approx 0$).

Next, we calculate v_y. You might think that rotating about the Y axis gives no y component to the velocity, but this intuition turns out to be incorrect.

$$\begin{aligned} v_y &= f\frac{d}{dt}(\frac{Y_0(t)}{Z_0(t)}) \ |_{t=0} \\ &= f\frac{\Omega X_0 Y_0}{Z_0^2} \\ &= \Omega\frac{xy}{f} \end{aligned}$$

Again, we see that there is a second order component to the rotation. We will see an example below (see plots).

Rotation about x axis

Rotating about the x axis (called "pitch" or "praying" or "nodding") leads to similar equations, except that the x and y axes are swapped.

$$(v_x, v_y) = \Omega(\frac{xy}{f}, f(1 + (\frac{y}{f})^2)\ )$$

The dominant component is the constant y component Ωf, and there are second order components there as well. These second order components are more significant away from the optical axis (the origin of the image).

Rotation about z axis

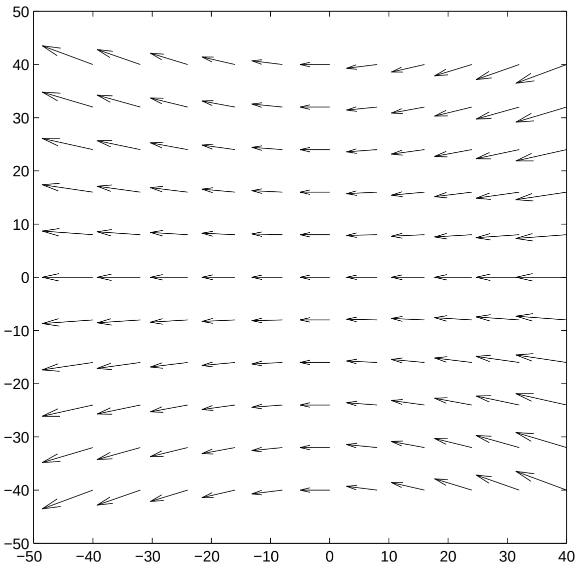
Finally, we consider the camera rotation about the optical axis. Again let the angular velocity be Ω. Then,

$$\begin{bmatrix} X_0(t) \\ Y_0(t) \\ Z_0(t) \end{bmatrix} = \begin{bmatrix} \cos(\Omega t) & -\sin(\Omega t) & 0 \\ \sin(\Omega t) & \cos(\Omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix}$$

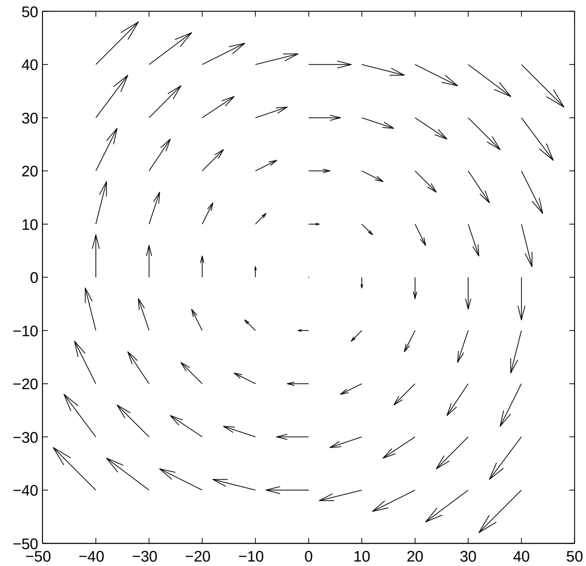
Cranking through the calculations, we get:

$$(v_x, v_y) = \Omega(-y, x)$$

This defines a vector field in the motion is along circles and the image speed of a point is proportional to the radius of the circle on which the point lies.



$\mathbf{\Omega} = (0, 1, 0)$
pan



$\mathbf{\Omega} = (0, 0, 1)$
roll

Final points

A key difference between the motion fields that are caused by camera translation versus camera rotation is that the translation fields depend on the depth of points in the scene, whereas the rotation fields do not. This is interesting. Panning or rolling the camera gives you motion, but the motion tells you nothing about how far away the scene points are. To get information about the distance to scene points from motion, you need to translate the camera.

At the very end of the lecture I sketched out a general expression for a rotation about an arbrary axis (not just X, Y, Z) – called "Rodriguez'es formula for rotation". There wasn't quite enough time to go into all the details (or to derive the vector field from this formula). I will return to this general rotation problem in Part 3 of the course, when we look at how to estimate the motion parameters of the camera, and how to estimate the depths Z at each point (x, y) in the image.