Translating Observer

Suppose the observer translates in direction \((T_x, T_y, T_z)\) where this vector is in eye coordinates.

The path of the scene point over time, seen relative to the observer's eye coordinate system is:

\[
(x(t), y(t), z(t)) = (x_0 - T_x \cdot t, y_0 - T_y \cdot t, z_0 - T_z \cdot t)
\]

Thus, the (relative) 3D velocity of scene point \((x_0, y_0, z_0)\) is \((-T_x, -T_y, -T_z)\).

What is the (relative) 3D velocity of scene point \((x_0, y_0, z_0)\)?

What is the image velocity of this scene point?

The path in the image is as follows:

(similar equation for \(y\) coordinate)

\[
\frac{x(t)}{z(t)} = \frac{x_0 - T_x \cdot t}{z_0 - T_z \cdot t}
\]

The \(x\) component of image velocity at \(t = 0\) is:

\[
v_x = \frac{1}{z_0} \frac{dx}{dt} \bigg|_{t=0} = \frac{-(T_x x_0 + T_y z_0)}{z_0^2}
\]

Lateral Component \((T_z = 0)\)

\[(V_x, V_y) = \frac{1}{z_0}(-T_x, -T_y)\]

Lateral Component \((T_z = 0)\)

\[(V_x', V_y') = \frac{1}{z_0'}(-T_x, -T_y)\]

Heads up: there is an ‘upside down and backwards’ issue here. I draw the motion fields as a function of visual angle position in front of the eye (not upside down and backwards, as on the retina).

Lateral motion and balance

(difficult
(no retinal motion since points are at infinity)
easy
(large velocity for nearby points on ground)
Fear of heights - perceptual factors

Normally when we look down, the ground is close and so any swaying of the body will produce small image velocities. When you are on the edge of a precipice, the ground is far away and so the image velocity is near zero. This messes up the body’s posture control system which relies in part on visual input.

Motion Parallax

The term ‘motion parallax’ refers specifically to the difference in velocity of two nearby points, which is due to their differences in depth and to the eye’s changing position over time. In a nutshell, when the eye moves, objects that are closer move faster.

Motion Parallax

\[ \Delta (v_x, v_y) = \left( \frac{1}{z_0} - \frac{1}{z_1} \right) \left( -T_x, -T_y \right) \]

Binocular disparity

Disparity = \[ T_x \left( \frac{1}{z_0} - \frac{1}{z_1} \right) \]

Note illegal use of symbol \( T_x \) to mean two things.

Forward Motion (\( T_x = T_y = 0, \ T_z \neq 0 \))

What does a pilot see when approaching the runway? (from JJ Gibson 1950)

"Optical flow": term originally referred to the motion field seen by a moving observer. (In computer vision, it now refers to any motion field e.g. including the motion due to moving scene objects.)

Forward Motion (\( T_x = T_y = 0, \ T_z \neq 0 \))

\( \left( \frac{\alpha}{\beta}, \frac{\gamma}{\delta} \right) \)

visual angles (\( \theta_x, \theta_y \))

in radians

In case of a wall, \( z_0 \) is constant.

General Translation (\( T_x, T_y, T_z \)), where \( T_z \neq 0 \)

\[ \left( v_x, v_y \right) = \frac{T_z}{z_0} \left( \frac{x}{T_z}, \frac{y}{T_z} \right) \]

careful

\[ \left( \frac{x}{T_z}, \frac{y}{T_z} \right) \]

where \( \left( \frac{T_x}{T_z}, \frac{T_y}{T_z} \right) \) is the “direction of heading” or “focus of expansion”

Direction of Heading

\[ \left( v_x, v_y \right) = \frac{T_z}{z_0} \left( \frac{x}{T_z}, \frac{y}{T_z} \right) \]

ASIDE: Optical Flow and Motion Blur

Motion blur arises from the finite "integration time" of photoreceptors.

The local orientation patterns that are due to motion blur depend on the motion field (which depends on the observer’s motion)

Computational Problem (translating observer)

1) Estimate image velocity at each \( (x,y) \).
We sketched this out last class.

2) Estimate the direction of heading
Details omitted because it would take an entire lecture to explain.

3) Estimate the distance at each \( (x,y) \).
Details omitted because it would take an entire lecture to explain.
Motion due to Observer Rotation

It is not obvious why the field curves as shown. Equations of the velocity field would be covered in a computer vision course.

For pan and tilt, motion field is approximately constant within 20 degrees of fovea. Recall rotation geometry discussed at start of lecture 10.

Eye Rotations (called "eye movements")

Types of eye movements

1) saccades
2) smooth pursuit
3) VOR (vestibulo-ocular reflex)
4) OKN (optokinetic nystagmus)

Saccades

- very fast rotation speed
  (vision system ignores the motion blur during this rotation)
- "fixations" last about 1/3 second
  i.e. 3 saccades/sec on average
- saccade targets and sequence depend on many factors

"Optimal eye movement strategies in visual search", Nature 2005

http://www.utexas.edu/cdot/files/1518227

"The Unexpected Visitor" classic experiment by Yarbus
"Smooth Pursuit"
- follow a moving object (ball, person, car)
- keep the fovea on the object to get high resolution detail
- cancel the motion (if possible)

\[
\begin{pmatrix}
    V_x \\
    V_y
\end{pmatrix}
+ \begin{pmatrix}
    V_x' \\
    V_y'
\end{pmatrix} = 0
\]
due to object due to observer

VOR (eye rotations due to head movement)
- not driven by visual input (works with eye's closed!)
- much faster than smooth pursuit
(too fast and too accurate to depend on cortical processing of image motion)

What is the mechanism of VOR?

IMU - inertial measurement unit (term used in robotics)

Measures
- linear i.e. translation acceleration of head
  \[ \frac{d}{dt} (T_x, T_y, T_z) \]
- rotational i.e. angular acceleration of head
  \[ \frac{d}{dt} (\text{pan, tilt, roll}) \text{ rate} \]

Vestibular System (in the inner ear)
Semi-circular tubes (canals) that contain fluid, which creates a drag.
Otoliths (little stones)

These mechanical motions are sensed by small hairs that are attached to the sensory neurons, which send spike trains to the brain.

Putting it all together: computational problem for a general moving observer

Elements of the solution (sketch only):
1) Estimate image velocity at each (x,y). (sketched this out last lecture)
2) Estimate self translation and rotation (IMU = vestibular)

Estimate self translation and rotation (visual)
Computer vision tells us this is a nasty but solvable problem, details of how the brain solves it are not so well understood.

3) Estimate the distance to each scene point (x,y) based on estimates from steps (1) and (2) and equations that we saw earlier.