Single View Geometry

Camera model & Orientation + Position estimation

What am I?
Ideal case:

Projection equation:
\[ x' = f \frac{X}{Z} \]
\[ y' = f \frac{Y}{Z} \]

\[ Zx' = f \times X \]
\[ Zy' = f \times Y \]
\[ Z = Z \]
Step 1: Camera projection matrix

\[
P_0 \times \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \]

\[
Zx' = fX \\
Zy' = fY \\
Z = Z
\]
Step 2: Intrinsic camera parameters: map camera coordinate to *pixel* coordinate

\[
\begin{bmatrix}
\alpha_x & s & p_x \\
0 & \alpha_y & p_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x' \\
y' \\
z'
\end{bmatrix} =
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix}
\]

\(K\) (3x3 submatrix)

\(\alpha_x, \alpha_y\) is pixel scaling factor

\(p_x, p_y\) is the principle point (where optical axis hits image plane)

\(s\) is the slant factor, when the image plane is not normal to the optical axis
Combine the Intrinsic camera parameters

\[
P_0 X = x
\]

\[
P = [K, 0] = K [I, 0]
\]

\[
\begin{bmatrix}
\alpha_x & s & p_x \\
0 & \alpha_y & p_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
f & 0 & 0 & 0 \\
0 & f & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
X_c \\
Y_c \\
Z_c \\
1
\end{bmatrix} =
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix}
\]
Step 3: External parameters: rotation and translation map the world to camera coordinates

\[ X_c = \begin{bmatrix} R_{3\times3} & t_{3\times1} \\ 0 & 1 \end{bmatrix} X \]

(R,t)

World coordinate

Camera coordinate
Combining Internal and External parameters

1) \( X_c = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} X \)

2) \( x = K_{3\times3} \begin{bmatrix} I; 0_{3\times4} \end{bmatrix} X_c \)

1) Translate the \textbf{world} coordinate into the \textbf{camera} coordinate
2) Translate the \textbf{camera} coordinate into the \textbf{pixel} coordinate
Combining Internal and External parameters

\[ X_c = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \end{bmatrix} \]

\[ x = K_{3 \times 3} \begin{bmatrix} I ; 0 \end{bmatrix}_{3 \times 4} X_c \]

After simplification:

\[ x = K \begin{bmatrix} R, t \end{bmatrix} X \]

(pixel, world)
Special case, planar world, homograph

\[ X = (x_w, y_w, Z_w=0, 1) \]

\[ x = K [R, t] X \]

Expand:

\[ x = K \begin{bmatrix} r1 & r2 & r3 \\ t \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ Z_w=0 \\ 1 \end{bmatrix} \]
Special case, planar world, homograph

\[ X = (x_w, y_w, Z_w = 0, 1) \]

After simplification:

\[ x = K \begin{pmatrix} r1 & r2 & t \\ r1 & r2 & t \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ 1 \end{pmatrix} \]

We have the homographic mapping:

\[ x = K H_{3 \times 3} X \]
Special case: rotating camera

\[ x = K \begin{bmatrix} R, t \end{bmatrix} X \]

*with* \( t = 0 \)

Expand:

\[ x = K \begin{bmatrix} R \end{bmatrix} X \]

\[ X = \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} \]

This is also a homography with \( H = R \)
Panoramas

\[ x = K [R] X \]
Visual Odometry: Overview

1) The transformation \((R, t)\) tells us how to rotate + move the **world** coordinate to the **camera** coordinate.

2) In the following slides, we will see
   1) how to get the world reference’s frame orientation+position from \((R, t)\),
   2) how to get the pan/tilt/yaw angles from \(R\)

3) If we want to know the **camera** orientation+position, we need to look at its inverse transform, which we should see is \((R^T, R^T(-t))\),
   1) The computation of the camera pan/tilt/yaw angle is the same as in the previous case
Recover Camera orientation
Case 1: single vanishing point (z-direction)
Pan/tilt/yaw angles

Pan $\alpha$: rotation around $y$-axis

Yaw $\gamma$: rotation around $z$-axis

Tilt $\beta$: rotation around $x$-axis
Rotation: Pan/Tilt/Yaw

Given the 3x3 rotation matrix R, we can recover, pan/tilt/yaw angle, and vice versa.
The basic idea is that if we can see the “north” star, we know how to oriented ourselves (minus yaw angle) (why?) And the `north” star is just the point of infinity in some direction
Seeing the vanishing point can tell us:

Camera projection equation: \( \mathbf{x} = \mathbf{K} \cdot [\mathbf{R}, \mathbf{t}] \cdot \mathbf{X} \), or

\[
\begin{bmatrix}
    x_w \\
    y_w \\
    z_w \\
    t_w
\end{bmatrix}
= \begin{bmatrix}
    x_c \\
    y_c \\
    z_c
\end{bmatrix}
\]

Columns of \( \mathbf{R} \) matrix = image of vanishing points of world-space axes!

\( r_1, r_2, r_3 \) == image of \( x, y, z \) axis vanishing points

\[
\begin{bmatrix}
r_1 & r_2 & r_3 & t
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix}
= r_3 \quad \text{(need to normalize to norm =1)}
\]

\( r_1, r_2, r_3 \) == image of \( x, y, z \) axis vanishing points

\( \mathbf{y}_c \) & \( \mathbf{z}_c \)

vanishing point
Recover Pan/Tilt angle from z-vanishing point

Geometric explanation: Pan $\alpha$  Tilt $\beta$

The alignment between z-axis of the world and image is defined only by the pan/tilt angle:

$$r_3 = \begin{bmatrix} \sin \alpha \cos \beta \\ \sin \beta \\ \cos \alpha \cos \beta \end{bmatrix}$$

Note: tilt angle with rotation around x-axis is positive when pointing down in this figure
In this example, we are walking down a track, the ground is tilted up (we are going up the hill). Assume we are $b$ meter tall. We see two parallel lines to our left and right, with $x = 1, -1$. A line on the ground is described by equation $y = az - b$, where $a$ is the slope the ground.

So we know: $\tan(\beta) = a$
As a line gets far away, $z \to \infty$. If $(1,y,z)$ is a point on this line, its image is $f(1/z,y/z)$. As $z \to \infty$, $1/z \to 0$. What about $y/z$?

$y/z = (az-b)/z = a - b/z \to a$.

So a point on the line appears at: $(0,a)$. 

![Diagram of a line on the image plane with a vanishing point at (0,a)](image)
Now, we check

\[ \tan(\beta) = a \]
Recover Pan/Tilt angle from z-vanishing point

Geometric explanation: Pan $\alpha$  Tilt $\beta$

The alignment between $z$-axis of the world and image is defined only by the pan/tilt angle:

$$
\mathbf{r}_3 = \begin{bmatrix}
\sin \alpha \cos \beta \\
\sin \beta \\
\cos \alpha \cos \beta
\end{bmatrix}
$$

$$
\alpha = \tan^{-1}\left(\frac{\mathbf{r}_3(1)}{\mathbf{r}_3(3)}\right)
$$

$$
\beta = \sin^{-1}(\mathbf{r}_3(2))
$$
Recover Camera orientation

Case 2: two vanishing points, x-y direction
In some cases, we can’t measure the z-vanishing point directly, but if we have x-y vanishing points, we can imagine what the z-vanishing point looks like.
Basic idea:

1) x-y vanishing point gives us the first two columns of $\mathbf{R}$, $\mathbf{r}_1$ and $\mathbf{r}_2$

2) z-vanishing point, the third columns of $\mathbf{R}$, can be computed by $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$

Recall: vanishing points of x,y direction, give us the, $\mathbf{r}_1$ and $\mathbf{r}_2$
Steps for Recovering $R$ from Two vanishing Point

1. Measure $X$ / $Y$ direction va
Recovering $R$ from Two vanishing Point (2)

2. Translate to the camera coordinates by intrinsic matrix $K$

$$x_{cam} \sim K^{-1} \cdot \begin{pmatrix} x_{img} & x \\ x_{img} & y \\ 1 \end{pmatrix}$$

[1.431, -0.372]

[0.020, 2.783]
Recovering $R$ from Two vanishing Point (3)

3. Get $r_1$ and $r_2$ from vanishing point $x_{vX}$ and $x_{vY}$

$$r_1 = \frac{x_{vX}}{\|x_{vX}\|}$$
$$r_2 = \frac{x_{vY}}{\|x_{vX}\|}$$

vanishing points of $x,y$ direction, give us the, $r_1$ and $r_2$

first two columns of $R$
Recovering $R$ from Two vanishing Point (4)

4. Get $r_3$

\[ r_1 = (0.8017, -0.2086, 0.5602)^T \]
\[ r_2 = (0.0067, 0.9411, 0.3382)^T \]
\[ r_3 = r_1 \times r_2 = (-0.5988, -0.2673, 0.7558)^T \]
\[ r_3 = r_1 \times r_2 \]

5. Get $R$

\[
R = \begin{pmatrix}
0.8017 & 0.0067 & -0.5977 \\
-0.2086 & 0.9411 & -0.2673 \\
0.5602 & 0.3382 & 0.7558 \\
\end{pmatrix}
\]
Example Recover Camera Pan /Tilt From $r_3$

$$\alpha = \tan^{-1}(r_3(1)/r_3(3))$$

$$\beta = \sin^{-1}(r_3(2))$$

$$r_3 = (-0.5988, -0.2673, 0.7558)^T$$

$$\alpha = -0.6691 = -0.2130\pi$$

$$\beta = +0.2706 = +0.0861\pi$$
Recover Camera Pan /Tilt From $r_3(2)$

Examples of Camera Pan/Tilt angle
Recovering all three angles: yaw + pan/tilt

\[ R = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} \]

1) Write out equation in yaw/pan/tilt angles for \( r_1 \) and \( r_2 \)

2) Solve for yaw angle given pan/tilt angles

Depends on pan/tilt

Depends on yaw/pan/tilt
Summary so far

1) If we have the vanishing point in z-direction, we can recover pan/tilt angle
   1) This is when the planar target is on the ground
2) If don’t see the z-direction vanishing point directly, you would need two vanishing points of x-y directions
3) Given x-y directions vanishing point, we can cover the full rotation matrix, therefore its z-direction (pan/tilt)
4) Given pan/tilt + x-direction vanishing point, we can recover yaw angle
Recover Camera orientation + position
Case 3: homography matrix
Recall for planar surface

\[
x = K \begin{pmatrix} r_1 & r_2 & t \\ \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ 1 \\ \end{pmatrix}
\]

We have the homographic mapping:

\[
x = K H_{3x3} X
\]
Recall for planar surface

\[ x = K \begin{pmatrix} r1 & r2 & t \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ 1 \end{pmatrix} \]

This implies if we have H,

1) We can recover the full rotation matrix, \( R \),
2) We can recover the position vector \( t \)
Example Recover $[R, t]$ from $H$ (1)

1. Get $H$ from 4 points

\[
H = \begin{pmatrix}
0.4430 & 0.0037 & -0.1071 \\
-0.1153 & 0.5216 & 0.1506 \\
0.3096 & 0.1875 & 0.5944 \\
\end{pmatrix}
\]

2. Computer the norm of first column

\[
a = \| (H_{11}, H_{21}, H_{31}) \| \]
Example Recover \([R, t]\) from \(H\) (2)

3. Computer \(r_1, r_2, t\)

\[
\begin{align*}
    t & = H(:, 3)/a = (-0.1937, 0.2726, 1.0756)^T \\
    r_1 & = H(:, 1)/a = (0.8017, -0.2086, 0.5602)^T \\
    r_2 & = H(:, 2)/a = (0.0067, 0.9439, 0.3392)^T
\end{align*}
\]

4. Computer \(r_3\)

\[
\begin{align*}
    r_3 & = r_1 \times r_2 = (-0.1937, 0.2726, 1.0756)^T
\end{align*}
\]
Procedure of recovery $R$, and $t$ from $H$

1) Compute the rotation of x-axis, set
   \[ \mathbf{r}_1 = H(:, 1); \quad \mathbf{X} = [\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}^T \quad \mathbf{x}_w \quad \mathbf{y}_w \quad \mathbf{1}] \]
   \[ a = \text{norm}(\mathbf{r}_1) \]

2) Compute the rotation of the y-axis, set
   \[ \mathbf{r}_2 = H(:, 2)/a; \]

3) Compute the translation vector $t$ (the position of the robot)
   \[ \mathbf{t} = H(:, 3)/a; \]

4) Compute the rotation of the z-axis, set
   \[ \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \quad \mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3] \]

5) Use the previous case(2), to compute the pan/tilt/yaw angle
Summary so far

1) Each vanishing point in the image gives you one rotation vector
2) Given the z-vanishing point, we can detect the pan/tilt angle.
   1) This is the case when the planar target is on the ground.
3) Given x-y vanishing point, we can compute the z-vanishing point (therefore pan/tilt angle).
   1) This is the case when we are facing the planar surface
   2) Given x-rotation vector, pan-tilt angle, we can determine the yaw angle
   3) But we can’t determine the position of the robot
4) If we are given the homograph, (using 4 corresponding points), we can recover both the rotational angles, and the position of the robot.
How to estimate the rotation and translation of the robot from the world point of view?

In the case of moving robot (rather than moving target), we need to know the orientation/position of the robot in the world

we need to how to pan/tilt the world oriented to the robot.

Note: pan/tilt of the camera is very different from the pan/tilt of the world!
Converting rotation/translation from (world→camera) to (camera→world)

\[
\begin{align*}
(R, t) & \quad \text{(world→ camera)} \\
(R^T, R^T(-t)) & \quad \text{(camera→world)}
\end{align*}
\]

With the fact: \( R^{-1} = R^T \)
Converting rotation/translation from \((\text{world-}\to\text{camera})\) to \((\text{camera-}\to\text{world})\)

What does it mean:

1) The position of the camera in the world reference frame is: \(R^T t\)

2) The pan/tilt/yaw of the world to align with camera is computed from the **rows** of the \(R\) instead of columns of \(R\)
Recovering World pan/tilt from Rotation matrix $R^T$

1. Compute the (world->camera) rotation $R$ and translation $t$

2. Take the last row of the $R$, and recover pan/tilt angle of camera

$$R = \begin{pmatrix} 0.8017 & 0.0067 & -0.5977 \\ -0.2086 & 0.9411 & -0.2673 \\ 0.5602 & 0.3382 & 0.7558 \end{pmatrix}$$

$$\alpha_{camera} = tan^{-1}(R(3, 1)/R(3, 3))$$

3) Compute the yaw angle if needed, from first two rows of $R$

4) Compute the position of the robot, by

$$\beta_{camera} = sin^{-1}(R(3, 2))$$

$$t_{camera} = R^T(-t)$$
Example Recover Pan/Tilt in *World* Frame

From World-> Camera: \([R, t]\)

\[\Rightarrow\] From Camera -> World : \([R^T, -R^Tt]\)
Example Recover Translation in *World* Frame (2)
1) There are two rotation/translation we care about:

1) $(R,t)$: world-$\to$ camera transformation. It tells us parameters of the **camera**. The columns of $R$: rotation of camera so it aligns with the world. The vector $t =$ coordinate of the world center in camera frame.

2) $(R^T, R^T(-t))$: camera-$\to$ world. It tells us parameters of the **world**. Columns of $R^T$: rotation of the world so it aligns with the camera. The vector $R^T(-t) =$ coordinate of the camera in the world frame.

3) For the visual odometry task used in project 2, you need to use the transformation $(R^T, R^T(-t))$.

4) The pan angle of the world, is the robot orientation in the world frame.