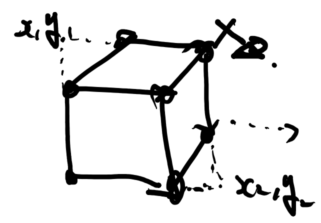
Paper behind: 3D Bounding Box Estimation Using Deep Learning and Geometry

Sunday, December 10, 2017 3:17 PM



Note

$$\underset{3\times1}{x_{img}} \approx \underset{3\times4}{K} \cdot [\underset{3\times3}{R} \mid \underset{3\times1}{T}] \underset{4\times1}{X_{world}}.$$

Rewrite

$$\approx K \cdot \begin{bmatrix} \underset{3\times3}{I} & \underset{3\times1}{RX_o} \\ \underset{1\times3}{0} & 1 \end{bmatrix} T$$

· multiplication form
· To make $\boxed{Ax=b}$ form later.

Here, unknown is $T = [t_x, t_y, t_z, 1]^T$.

— * — * — * —

we have $(x_{min}, y_{min}, x_{max}, y_{max})$ in 2D

we have 8 $(\pm w/2, \pm h/2, \pm d/2)$ in 3D.

Generally, $\underbrace{x_{min}}_{2D}$ is the projection of any $\underbrace{8 \text{ pts}}_{3D}$.

total $8^4 = 4096$ configurations.

(one can reduce this)

Let's say, $(x_{min} \ldots) \xleftarrow{\text{from}} \overset{(a,b,c,d)}{(X_a, X_b, X_c, X_d)}$

$$x_{min} \approx K \cdot \begin{bmatrix} I & RX_a \\ 0 & 1 \end{bmatrix} \cdot T \Big|_{(x)} \qquad (1)$$

homogeneous (up to scale)

* only x coord matters.

$$y_{min} \approx K \cdot \begin{bmatrix} I & RX_b \\ 0 & 1 \end{bmatrix} \cdot T \Big|_{(y)} \qquad (2)$$

x_{max}, y_{max} are similar. (3), (4)

We have 4 equations. 3 unknowns.
→ over-constraint. → least square.

$$\underset{4\times3}{A} \underset{3\times1}{x} = \underset{4\times1}{b} \qquad \text{where } x = [t_x, t_y, t_z]^T$$

$$\rightarrow x = (A^TA)^{-1} \cdot A^T \cdot b.$$

However (1),(2),(3),(4) are up to scale!!
CAREFUL!
To deal with this, for example of (1),

$$\underbrace{K \cdot \begin{bmatrix} I & RX_a \\ 0 & 1 \end{bmatrix}}_{\text{let's say this guy } M_a \ (3\times4)} \begin{bmatrix} t_x \\ t_y \\ t_z \\ 1 \end{bmatrix} = \begin{bmatrix} s \cdot x_{min} \\ \cancel{*} \\ s \end{bmatrix}$$

← DON'T CARE. DON'T KNOW.

$$M_a[0,:] \cdot \begin{bmatrix} t_x \\ t_y \\ t_z \\ 1 \end{bmatrix} = s \cdot x_{min}.$$

$$M_a[2,:] \cdot \begin{bmatrix} t_x \\ t_y \\ t_z \\ 1 \end{bmatrix} = s.$$

$$\Rightarrow M_a[0,:] \cdot \begin{bmatrix} t_x \\ t_y \\ t_z \\ 1 \end{bmatrix} = M_a[2,:] \begin{bmatrix} t_x \\ t_y \\ t_z \\ 1 \end{bmatrix} \cdot x_{min}.$$

remember. this is unknow x

$$\Rightarrow (M_a[0,:3] - x_{min} M_a[2,:3]) \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$= x_{min} \cdot M_a[2,3] - M[0,3]$$

More cleanly,

$$v := M_a[0,:] - x_{min} \cdot M_a[2,:]$$

$$\underset{1\times3}{v[0,:3]} \cdot \underset{3\times1}{x} = -\underset{1\times1}{v[0,-1]}.$$

fill first row.

$$[-]x = [\cdot]$$
$$Ax = b.$$

— * — * —

therefore, for a given configuration $\mathbb{C} = (a, b, c, d)$,

bbox	x_{min}	y_{min}	x_{max}	y_{max}
index	0	1	0	1
row	0	1	2	3

$$A[row,:] = M_{\underset{c[row]}{o}}[index, :3] - bbox[row] \cdot M_o[2, :3]$$

$$b[row] = bbox[row] \cdot M_o[2,3] - M_o[index, 3].$$

Then, solve for x. Again, $x = (A^TA)^{-1}A^T \cdot b$.