Reprojection

Reprojection

First each point $\vec{X}$ in 3d coordinate space is translated into the camera's reference frame by subtracting $\vec{C}$

$$\vec{X}_c = \vec{X} - \vec{C}.$$

It is then transformed under the camera's rotation matrix $\underline{R}$

$$\vec{X}_c' = \underline{R}\,\vec{X}_c = \begin{bmatrix} X_x \\ X_y \\ X_z \end{bmatrix}.$$

$$\begin{aligned} X_x &= [R_{11} \quad R_{12} \quad R_{13}](\vec{X} - \vec{C}) \\ X_y &= [R_{21} \quad R_{22} \quad R_{23}](\vec{X} - \vec{C}) \\ X_z &= [R_{31} \quad R_{32} \quad R_{33}](\vec{X} - \vec{C}) \end{aligned}$$

Where the rotation matrix is given by 3 Euler angles (α, β, γ)

$$\underline{R} = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}.$$

The coordinates are then taken into the camera plane by division by the z component of $\vec{X}_c'$. They are scaled by f_l representing the focal length of the simulated camera.

$$\vec{U} = f_l \begin{bmatrix} X_x/X_z \\ X_y/X_z \end{bmatrix}.$$

In the normalised camera plane, these coordinates are subjected to radial and tangential distortions. The quantity $r_{sq} = \left(\left|\vec{U}\right|^2\right)^2$ is of use for these.

$$\vec{U}_r = \vec{U}\left(k_1 r_{sq} + k_2 r_{sq}^2 + k_3 r_{sq}^3\right),$$

$$\vec{U}_t = \begin{bmatrix} 2p_1 U_x U_y + p_2(r_{sq} + 2U_x^2) \\ p_1\left(r_{sq} + 2U_y^2\right) + 2p_2 U_x U_y \end{bmatrix}.$$

Adding these distortion effects to $\vec{U}$ yields the normalised distorted coordinates, which can finally be scaled to the final image resolution and offset to centre $(0, 0)$ at the middle of the image.

$$\vec{P} = \vec{U} + \vec{U}_r + \vec{U}_t + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} p_x/w \\ p_y/h \end{bmatrix}$$

In all, the reprojection function $\vec{f}\left(\underline{R}, \vec{C}, \vec{X}\right) = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$.

The Jacobian

The Jacobian of $\vec{f}$ is a (2×9) matrix for a single point in space and a single camera.

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial E} & \frac{\partial f_x}{\partial C} & \frac{\partial f_x}{\partial X} \\ \frac{\partial f_y}{\partial E} & \frac{\partial f_y}{\partial C} & \frac{\partial f_y}{\partial X} \end{bmatrix}$$

where $E = [\alpha \quad \beta \quad \gamma]^T$ is the Euler angle vector, $C = [C_x \quad C_y \quad C_z]^T$ is the camera's position, and $X = [X_x \quad X_y \quad X_z]^T$ is the point's position.

Point Position

First consider the effect of the point's position X.

$$\frac{\partial f}{\partial X} = \frac{\partial}{\partial X}\left(\vec{U} + \vec{U}_r + \vec{U}_t + 0.5\right) \cdot \begin{cases} w & \text{if } x \\ h & \text{if } y \end{cases}$$

$$\frac{\partial \vec{U}}{\partial X} = f_l \frac{\partial}{\partial X}\begin{bmatrix} X_x/X_z \\ X_y/X_z \end{bmatrix} = f_l \frac{1}{X_z^2}\begin{bmatrix} \partial_X(X_x)X_z - X_x\,\partial_X(X_z) \\ \partial_X(X_y)X_z - X_y\,\partial_X(X_z) \end{bmatrix}$$

$$\partial_X(X_x) = [R_{11} \quad R_{12} \quad R_{13}]$$
$$\partial_X(X_y) = [R_{21} \quad R_{22} \quad R_{23}]$$
$$\partial_X(X_z) = [R_{31} \quad R_{32} \quad R_{33}]$$

$$\frac{\partial \vec{U}_r}{\partial X} = \frac{\partial \vec{U}}{\partial X}\left(k_1 r_{sq} + k_2 r_{sq}^2 + k_3 r_{sq}^3\right) + \vec{U}\partial_X\left(k_1 r_{sq} + k_2 r_{sq}^2 + k_3 r_{sq}^3\right)$$

$$\partial_X(r_{sq}) = \partial_X\left(\vec{U}^2\right)^2 = 4\left|\vec{U}\right|^2\left(\vec{U} \cdot \frac{\partial \vec{U}}{\partial X}\right)$$

$$\partial_X(r_{sq}^n) = n\, r_{sq}^{n-1}\partial_X(r_{sq})$$

$$\frac{\partial \vec{U}_t}{\partial X} = \partial_X\begin{bmatrix} 2p_1U_xU_y + p_2(r_{sq} + 2U_x^2) \\ p_1(r_{sq} + 2U_y^2) + 2p_2U_xU_y \end{bmatrix} = \begin{bmatrix} 2p_1\left(\partial_X(U_x)U_y + U_x\partial_X(U_y)\right) + p_2\left(\partial_X(r_{sq}) + 4U_x\partial_X(U_x)\right) \\ p_1\left(\partial_X(r_{sq}) + 4U_y\partial_X(U_y)\right) + 2p_2\left(\partial_X(U_x)U_y + U_x\partial_X(U_y)\right) \end{bmatrix}$$

Camera Position

$$\frac{\partial \vec{f}}{\partial C} = \frac{\partial}{\partial C}\left(\vec{U} + \vec{U}_r + \vec{U}_t + 0.5\right) \cdot \begin{cases} w & \text{if } x \\ h & \text{if } y \end{cases}$$

$$\frac{\partial \vec{U}}{\partial C} = f_l \frac{\partial}{\partial C}\begin{bmatrix} X_x/X_z \\ X_y/X_z \end{bmatrix} = f_l \frac{1}{X_z^2}\begin{bmatrix} \partial_C(X_x)X_z - X_x\,\partial_C(X_z) \\ \partial_C(X_y)X_z - X_y\,\partial_C(X_z) \end{bmatrix} = -\frac{\partial \vec{U}}{\partial X}$$

$$\partial_C(X_x) = -[R_{11} \quad R_{12} \quad R_{13}]$$
$$\partial_C(X_y) = -[R_{21} \quad R_{22} \quad R_{23}]$$
$$\partial_C(X_z) = -[R_{31} \quad R_{32} \quad R_{33}]$$

$$\frac{\partial \vec{U}_r}{\partial C} = \frac{\partial \vec{U}}{\partial C}\left(k_1 r_{sq} + k_2 r_{sq}^2 + k_3 r_{sq}^3\right) + \vec{U}\partial_C\left(k_1 r_{sq} + k_2 r_{sq}^2 + k_3 r_{sq}^3\right) = -\frac{\partial \vec{U}_r}{\partial X}$$

$$\partial_C(r_{sq}) = \partial_C\left(\vec{U}^2\right)^2 = 4\left|\vec{U}\right|^2\left(\vec{U} \cdot \frac{\partial \vec{U}}{\partial C}\right) \qquad \left(\ldots = -4\left|\vec{U}\right|^2\left(\vec{U} \cdot \frac{\partial \vec{U}}{\partial X}\right) = -\partial_X(r_{sq})\right)$$

$$\partial_C(r_{sq}^n) = n\, r_{sq}^{n-1}\partial_C(r_{sq})$$

$$\frac{\partial \vec{U}_t}{\partial C} = \partial_C\begin{bmatrix} 2p_1U_xU_y + p_2(r_{sq} + 2U_x^2) \\ p_1(r_{sq} + 2U_y^2) + 2p_2U_xU_y \end{bmatrix} = \begin{bmatrix} 2p_1\left(\partial_C(U_x)U_y + U_x\partial_C(U_y)\right) + p_2\left(\partial_C(r_{sq}) + 4U_x\partial_C(U_x)\right) \\ p_1\left(\partial_C(r_{sq}) + 4U_y\partial_C(U_y)\right) + 2p_2\left(\partial_C(U_x)U_y + U_x\partial_C(U_y)\right) \end{bmatrix}$$

Quite naturally the derivatives with respect to the camera's coordinates are the negative of those for the point's coordinates, as moving the point by $\overrightarrow{dX}$ is the same as moving the camera by $-\overrightarrow{dX}$.

Camera Rotation

$$\frac{\partial \vec{f}}{\partial E} = \frac{\partial f}{\partial R}\frac{\partial R}{\partial E} = \left(\frac{\partial}{\partial R}\left(\vec{U} + \vec{U}_r + \vec{U}_t + 0.5\right) \cdot \begin{cases} w & \text{if } x \\ h & \text{if } y \end{cases}\right)\frac{\partial R}{\partial E}$$

$\frac{\partial f}{\partial E}$ represents the Jacobian of the 2d vector $\vec{f}$ with respect to the 3 Euler coordinates, hence it is a (2×3) matrix. This can be broken into $\frac{\partial f}{\partial R}$, a (2×9) matrix and $\frac{\partial R}{\partial E}$, a (9×3) matrix, whose product gives the (2×3) matrix expected.

$$\frac{\partial \vec{f}}{\partial R} = \begin{bmatrix} \partial_{R_{11}}(f_x) & \partial_{R_{12}}(f_x) & \cdots & \partial_{R_{33}}(f_x) \\ \partial_{R_{11}}(f_y) & \partial_{R_{12}}(f_y) & \cdots & \partial_{R_{33}}(f_y) \end{bmatrix}$$

$$\frac{\partial \vec{f}}{\partial R_{ij}} = \partial_{R_{ij}}\left(\vec{U} + \vec{U}_r + \vec{U}_t\right)$$

$$\frac{\partial \vec{U}}{\partial R_{ij}} = f_l \frac{1}{X_z^2}\begin{bmatrix} \partial_{R_{ij}}(X_x)X_z - X_x\,\partial_{R_{ij}}(X_z) \\ \partial_{R_{ij}}(X_y)X_z - X_y\,\partial_{R_{ij}}(X_z) \end{bmatrix}$$

$$\partial_{R_{ij}}(X_x) = \begin{cases} \left(\vec{X} - \vec{C}\right)_j & i = 1 \\ \vec{0} & else \end{cases}$$

$$\partial_{R_{ij}}(X_y) = \begin{cases} \left(\vec{X} - \vec{C}\right)_j & i = 2 \\ \vec{0} & else \end{cases}$$

$$\partial_{R_{ij}}(X_z) = \begin{cases} \left(\vec{X} - \vec{C}\right)_j & i = 3 \\ \vec{0} & else \end{cases}$$

$$\frac{\partial \vec{U}_r}{\partial R_{ij}} = \frac{\partial \vec{U}}{\partial R_{ij}}\left(k_1 r_{sq} + k_2 r_{sq}^2 + k_3 r_{sq}^3\right) + \vec{U}\partial_{R_{ij}}\left(k_1 r_{sq} + k_2 r_{sq}^2 + k_3 r_{sq}^3\right)$$

$$\partial_{R_{ij}}(r_{sq}) = \partial_{R_{ij}}\left(\vec{U}^2\right)^2 = 4\left|\vec{U}\right|^2\left(\vec{U} \cdot \frac{\partial \vec{U}}{\partial R_{ij}}\right)$$

$$\partial_{R_{ij}}(r_{sq}^n) = n\, r_{sq}^{n-1}\partial_{R_{ij}}(r_{sq})$$

$$\frac{\partial \vec{U}_t}{\partial R_{ij}} = \partial_{R_{ij}}\begin{bmatrix} 2p_1 U_x U_y + p_2\left(r_{sq} + 2U_x^2\right) \\ p_1\left(r_{sq} + 2U_y^2\right) + 2p_2 U_x U_y \end{bmatrix} =$$

$$\begin{bmatrix} 2p_1\left(\partial_{R_{ij}}(U_x)U_y + U_x\partial_{R_{ij}}(U_y)\right) + p_2\left(\partial_{R_{ij}}(r_{sq}) + 4U_x\partial_{R_{ij}}(U_x)\right) \\ p_1\left(\partial_{R_{ij}}(r_{sq}) + 4U_y\partial_{R_{ij}}(U_y)\right) + 2p_2\left(\partial_{R_{ij}}(U_x)U_y + U_x\partial_{R_{ij}}(U_y)\right) \end{bmatrix}$$

Also need

$$\frac{\partial R}{\partial E} = \begin{bmatrix} \partial_\alpha(R_{11}) & \partial_\beta(R_{11}) & \partial_\gamma(R_{11}) \\ \partial_\alpha(R_{12}) & \partial_\beta(R_{12}) & \partial_\gamma(R_{12}) \\ \vdots & \vdots & \vdots \\ \partial_\alpha(R_{33}) & \partial_\beta(R_{33}) & \partial_\gamma(R_{33}) \end{bmatrix}$$