Image-to-world transforms in `ips_cam`

Stuart Johnson

stuart.g.johnson@gmail.com

September 23, 2024

`ips_cam` assumes that it is viewing multiple ARUCO tags, each at a given height above the floor, and parallel to the plane of the floor. Furthermore, it assumes it is given the camera intrinsic (K) and extrinsic matrix (E). The intrinsic matrix is the result of camera monocular calibration, and the extrinsic matrix is obtained from an image of a chessboard pattern which defines the world coordinate system (AKA the Indoor Coordinate System or ICS). A step in finding ARUCO tags in the ICS is a conversion of image points to world coordinates.

Writing the full camera projection matrix:

$$P = KE \tag{1}$$

we have the pinhole camera model:

$$P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{2}$$

where $[X, Y, Z]$ are world (ICS) coordinates and $[u, v]$ are image (pixel) coordinates. Note that $P \in \mathbb{R}^{3\times4}$ and λ is a scale factor to be determined by perspective divide. We can extract the third column of P as P_z, leaving the remainder of the matrix as P_{xy}. We now have:

$$P_{xy} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} + P_z Z = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{3}$$

$P_{xy} \in \mathbb{R}^{3\times3}$ is quite invertible, so we can write, after multiplying on the left by P_{xy}^{-1}:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} + P_{xy}^{-1} P_z = \lambda P_{xy}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{4}$$

We now note that λ is entirely determined by the third row of this equation:

$$1 + \left(P_{xy}^{-1} P_z\right)_{3,:} = \lambda \left(P_{xy}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \right)_{3,:} \tag{5}$$

where the notation $(M)_{3,:}$ refers to the third row of matrix M. Note that while this example yields a scalar equation in λ, we typically are solving for 4 corners of an ARUCO tag at once, and so we have an expression for λ for each point (a row vector equation, if you will). After solving for each λ we then solve for world coordinates using a version of equation 4:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \lambda P_{xy}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - P_{xy}^{-1} P_z \tag{6}$$

Which is to say, we are correcting for an error caused by the perspective shift of ARUCO tags at non-zero Z (height above the floor).

All calculations in the code are carried out with `OPENCV`'s `cv::Mat` class - with some custom extensions for matrix row and column surgery.