A Discussion on Camera Models

Kai Zhang

September 2018

1 Notation

Note that in projective space, we always use homogeneous coordinates, while in Euclidean space, Cartesian coordinate system is employed. Unless noted, the coordinate system should be inferred from the context. We also use $\mathcal{P}^n(\mathcal{R})$ to denote n dimensional projective space. $< \cdot, \cdot >$ stands for inner product of two vectors; for notational convenience, one of the two vectors can be a row vector while the other is a column vector.

2 Background

These ingredients are important in the imaging process of a digital image,

- scene coordinate frame
- camera coordinate frame
- projection into a plane
- sensor grid
- image grid
- rounding

The first three steps work in the Euclidean space with the distance unit being meter, centimeter, or others, while the following two steps achieves changing the distance unit to be pixel. The last step is to get the actual pixel coordinate.

Mathematically, a camera is a function that models the first five steps. The function's domain is the 3D Euclidean space $\mathcal{R}^3$, and its range is the image plane, which is an affine space. As a function, we care about its linearity; as a geometric transformation,[1] we care about its affinity. Linearity always implies affinity.

[1] when we talk about geometric transformations, the camera's domain, which is a Euclidean space, should also be thought of as an affine space.

3 Introduction

In projective space, image pixel coordinate $x \in \mathcal{P}(\mathcal{R})$ and object coordinate $X \in \mathcal{P}^2(\mathcal{R})$ are typically related by a matrix $P \in \mathcal{R}^{3\times 4}$, namely, $x = PX$. P is referred to as the camera matrix.

Written in Euclidean space, the equation would look like the following,

$$
\begin{aligned}
u &= \frac{< P_{(1,1:3)}, X > + P_{(1,4)}}{< P_{(3,1:3)}, X > + P_{(3,4)}} \\
v &= \frac{< P_{(2,1:3)}, X > + P_{(2,4)}}{< P_{(3,1:3)}, X > + P_{(3,4)}}
\end{aligned}
$$

, where $x = (u, v)$ are the pixel coordinate, and $P_{(i_1:i_2, j_1:j_2)}$ denotes the submatrix consisting of P's $i_1 \sim i_2$ rows and $j_1 \sim j_2$ columns.

Remark

We could easily see from the equation in Euclidean space that, a camera model $x = f(X)$ is in general non-linear. The fact that the equation is linear in projective space is more like an algebraic trick that enables fast computation by representing all geometric transformations as a matrix-vector multiplication.

Geometric transformations

In Euclidean space, some geometric transformations, e.g., scaling, rotation, shearing, reflection, orthographic projection, etc., are linear, while others, like translation, perspective projection and so on, are not. The good news is that they are all linear in projective space, which means they can be represented by a matrix.

In projective space, try to check the linearity of these geometric transformations: scaling, rotation, reflection, shearing, translation, orthographic projection, perspective projection.

An affine space is nothing but an Euclidean space plus a translation vector in this Euclidean space. This implies that any Euclidean space is also an affine space as we can choose the translation vector to be $\vec{0}$. In a general affine space, there's no concept of addition between two vectors; only subtraction makes sense. Geometric transformations between two affine spaces that preserve the geometric concepts—straight lines, planes, line parallelism, are especially interesting, and as a result, they are called affine transformations. Mathematically, any affine transformation should take the form $x = MX + b$ (x, X) in order to preserve the three geometric concepts. Apparently, $b = \vec{0}$ implies linear transformations in Euclidean space; $b \neq \vec{0}$ implies translation.

4 Decomposition of P for the perspective camera

The camera matrix P provides no insights into the physical imaging process. For the common perspective camera, people have derived a mathematical model from the imaging process.

From the very high-level perspective, the physics-based model implies that $P = K[R, t]$ holds for the perspective camera, with $K \in \mathcal{R}^{3\times 3}$ being an upper-triangular matrix that has positive diagonal entries, $R \in \mathcal{R}^{3\times 3}$ being an orthonormal matrix, $t \in \mathcal{R}^3$ is a vector. There are physical interpretations for the three parameters, K, R, t; K is called the intrinsic matrix, while $[R, t]$ is the extrinsic that aligns the scene coordinate frame with the camera coordinate frame; R represents a 3D rotation, and t is a 3D translation.

Besides the physical view, we can also try understanding $P = K[R, t]$ from a purely mathematical perspective. We are interested in how to decompose P into $K[R, t]$. First, notice that $P = K[R, t] = [KR, Kt]$. This means $P_{(1:3,1:3)} = KR$, and further implies that K, R can be recovered by QR-decomposition of the submatrix $P_{(1:3,1:3)}$.[2] After we have K, R in hand, $t = K^{-1}P_{(:,4)}$ can be applied to retrieve t. Note that the existence and uniqueness of such QR-decomposition requires the sub-matrix $P_{(1:3,1:3)}$ to be invertible.[3]

5 Perspective camera with a large focal length

There are mainly two ways to achieve a very narrow field of view; one is to increase the focal length, while the other is to shrink the sensor array size. Here we consider the influence of a large focal length.

Let N be a very large real number. For simplicity, we assume $f_x = f_y$. Then the perspective camera with the following intrinsic

$$K = \begin{bmatrix} Nf & Ns & Nx_0 \\ 0 & Nf & Ny_0 \\ 0 & 0 & 1 \end{bmatrix}$$

maps a point X to the pixel coordinate (Nu, Nv), with (u, v) being the pixel coordinate mapped by the perspective camera with the following intrinsic

$$K = \begin{bmatrix} f & s & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In words, a perspective camera with a large focal length is equivalent to a small-focus camera followed by a scale of the image.

[2] In pratice, we would actually first QR-decompose $[P_{(1:3,1:3)}]^{-1} = R^{-1}K^{-1}$, then inverse both sides to have $P_{(1:3,1:3)} = KR$.

[3] A matrix $A \in \mathcal{R}^{m\times n}, m \geq n$ is QR-decomposable if and only if it's full column rank. For a square matrix, full column rank is also equivalent to invertibility.

6 Linear approximation of the perspective camera

In last section, we have mentioned that the sub-matrix $P_{(1:3,1:3)}$ is invertible for the perspective camera. This fact implies that $P_{3,1:3}$ can't be a zero vector. Looking back at the projection equation in Euclidean space, we would see that the perspective camera is non-linear.

But is it affine? The answer is not. To see this more clearly, we re-write the camera matrix $P = K[R, t]$ in a slightly different way,

$$P_{3\times4} = K_{3\times3}O_{3\times4}\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}_{4\times4}.$$

, with

$$O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

being the perspective projection matrix. K, R, t are all affine transformations while O is not. Therefore, the perspective camera is also non-affine.

One direct result of the perspective camera's non-linearity is that line parallelism is not preserved. It can be further shown that a set of parallel lines will converge to a common point called the vanishing point after perspective projection. And a plane will converge to a vanishing line. This result is not that surprising if we take a second look at the projection equation. A line in a 3D Euclidean space is the set of points $\{X_0 + k\vec{n} : k \in \mathcal{R}\}$, with X_0 being an arbitrary point on the line, and $\vec{n}$ being the line's directional vector. Substitute $x_0 + k\vec{n}$ for X in the projection equation, then take limit with respect to k, and we would have

$$\begin{aligned} \lim_{k\to\infty} u &= \lim_{k\to\infty} \frac{< P_{(1,1:3)}, X_0 + k\vec{n} > + P_{(1,4)}}{< P_{(3,1:3)}, X_0 + k\vec{n} > + P_{(3,4)}} \\ &= \frac{< P_{(1,1:3)}, \vec{n} >}{< P_{(3,1:3)}, \vec{n} >} \\ \lim_{k\to\infty} v &== \frac{< P_{(2,1:3)}, \vec{n} >}{< P_{(3,1:3)}, \vec{n} >} \end{aligned}$$

. The above gives a proof to the existence of the vanishing point. We could also see that given a plane with the unit normal vector $\vec{d}$, let $X_0 + k\vec{n}$ be a line on the plane. Then obviously we have $< \vec{d}, \vec{n} >= 0$. Since $P_{(1:3,1:3)}$ is a full-rank matrix, we can have $\vec{d}^T = aP_{(1,1:3)} + bP_{(2,1:3)} + cP_{(3,1:3)}$, where at least one of a, b, c is non-zero. Then it's very easy to check $a \cdot (\lim_{k\to\infty} u) + b \cdot (\lim_{k\to\infty} v) + c = 0$. This justifies the existence of the vanishing line.

Because of the perspective camera's non-linearity, we would like to approximate it by a linear function for better mathematical tractability. The linear approximation consists of two cases—zero-order approximation and first-order approximation.

Taylor expansion

Look at the camera matrix $K[R,t]$ of a perspective camera again; the extrinsic $[R,t]$ is responsible for aligning the object coordinate frame with the camera coordinate frame. Rotation is linear, while translation is non-linear but affine.

Thus we don't have to approximate it, but instead we assume the object coordinate frame is already aligned with the camera coordinate frame. K is a 3×3 upper-triangular matrix, and as we are working in projective space, we further assume $K_{(3,3)} = 1$. There are also physical meanings for the remaining five parameters in K; but I will not go further here.

$$K = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Let (X,Y,Z) be a point in the camera coordinate frame. Left-multiply it with K, then get back to the pixel coordinate, and we have

$$\begin{aligned} u &= \frac{f_x X + sY + x_0 Z}{Z} \\ v &= \frac{f_y Y + y_0 Z}{Z} \end{aligned}$$

. If $Z \in [\bar{Z} - \frac{\hat{Z}}{2}, \bar{Z} - \frac{\hat{Z}}{2}]$ with $\bar{Z} >> \hat{Z}$, we could Taylor-expand it along the Z axis at $Z = \bar{Z}$.

$$\begin{aligned} u &= \frac{f_x X + sY}{\bar{Z} + \Delta Z} + x_0 \\ &= \frac{f_x X + sY}{\bar{Z}} \cdot \frac{1}{1 + \frac{\Delta Z}{\bar{Z}}} + x_0 \\ &= \frac{f_x X + sY}{\bar{Z}} \cdot (1 - \frac{\Delta Z}{\bar{Z}}) + x_0 + o((\frac{\Delta Z}{\bar{Z}})^2) \\ v &= \frac{f_y Y}{\bar{Z} + \Delta Z} + y_0 \\ &= \frac{f_y Y}{\bar{Z}} \cdot (1 - \frac{\Delta Z}{\bar{Z}}) + y_0 + o((\frac{\Delta Z}{\bar{Z}})^2) \end{aligned}$$

In the above, $\Delta Z = Z - \bar{Z}$. In words, the condition '$Z \in [\bar{Z} - \frac{\hat{Z}}{2}, \bar{Z} - \frac{\hat{Z}}{2}]$ with $\bar{Z} >> \hat{Z}$' means the scene depth $\hat{Z}$ is much smaller than the average object-camra distance $\bar{Z}$.

Zero-order approximation

The zero-order approximation is also called the weak-perspective camera, which is essentially the constant term in the Taylor expansion.

$$\begin{aligned} u &\approx \frac{f_x X + sY}{\bar{Z}} + x_0 \\ v &\approx \frac{f_y Y}{\bar{Z}} + y_0 \end{aligned}$$

Written in matrix form, the approximate camera is

$$\hat{P}_{3\times 4} = \hat{K}_{3\times 3}\hat{O}_{3\times 4} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}_{4\times 4}$$

, with

$$\hat{K} = \begin{bmatrix} f_x/\bar{Z} & s/\bar{Z} & x_0 \\ 0 & f_y/\bar{Z} & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \hat{O} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

. It's easy to see that $\hat{O}$ is linear in the Euclidean space, thus $\hat{P}$ is indeed linear in the Euclidean space.

First-order approximation

The first-order approximation is also called the para-perspective camera, which is essentially the constant term and the first-order term.

$$\begin{aligned} u &\approx \frac{f_x X + sY}{\bar{Z}} \cdot (1 - \frac{\Delta Z}{\bar{Z}}) + x_0 \\ &= \frac{f_x X + sY}{\bar{Z}} \cdot (1 - \frac{Z - \bar{Z}}{\bar{Z}}) + x_0 \\ &= \frac{(f_x X + sY) \cdot (2\bar{Z} - Z)}{\bar{Z}^2} + x_0 \\ v &\approx \frac{f_y Y \cdot (2\bar{Z} - Z)}{\bar{Z}^2} + y_0 \end{aligned}$$

7 Orthographic camera

For orthographic camera, the camera matrix is of the form

$$P_{3\times 4} = K_{3\times 3}O_{3\times 4} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}_{4\times 4}$$

. In the above, R, t represents the 3D rotation and translation, respectively; O represents the orthographic projection; K is the intrinsic.

$$K = \begin{bmatrix} p_x & s & x_0 \\ 0 & p_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}, O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

There is no concept of focal length in orthographic camera; thus to avoid confusion with the perspective camera, I use p_x, p_y in the intrinsic matrix instead of f_x, f_y.

The camera matrix of the orthographic camera would look like

$$P = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

8 Affine camera

Affine camera model is different from the perspective camera in the sense that it is linear in the Euclidean space. The camera matrix matrix of an affine camera takes the following form,

$$P = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

. We note that the sub-matrix $P_{(1:3,1:3)}$ is not invertible, while the perspective camera requires this sub-matrix to be invertible.

Mathematically, we can decompose P in such a way,

$$P_{3\times4} = K_{3\times3} O_{3\times4} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}_{4\times4}$$

, with

$$K = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ 0 & 0 & 1 \end{bmatrix}, O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and R, t represents 3D rotation and translation, respectively.

8.1 Orthographic camera v.s. affine camera

At first glance, these two cameras appears to the same. The difference actually lies in whether the sub-matrix $P_{1:2,1:3}$ is QR-decomposable or not.[4] For orthographic camera, the sub-matrix should be QR-decomposable, while it's not necessarily true for the affine camera. Thus, the orthographic camera is a special case of the affine camera.

[4] In practice, we would actually consider QR-decomposition of its pseudo-inverse. If $A \in \mathcal{R}^{m\times n}$, with $n > m$, and $(A^TA)^{-1}A^T$ is QR-decomposable, i.e., $(A^TA)^{-1}A^T = Q_{n\times m}R_{m\times m}$, then we have $A = R^{-1}Q^T$.

9 Putting all together

- perspective camera is both non-linear and non-affine.
- affine camera is linear and thus affine.
- orthographic camera is a special case of the affine camera. Their difference lies in whether the sub-matrix $P_{(1:2,1:3)}$ is QR-decomposable or not.
- when the scene depth is much smaller than the average distance between the scene and camera, perspective camera reduces to a weak-perspective camera through zero-order approximation; weak-perspective camera is a special case of the orthographic camera.