1 Introduction

In computer vision, the camera matrix is a 3×4 matrix that encapsulates the mapping performed by a pinhole camera, projecting 3D points in the real world onto 2D points in an image. This "mini"-project focuses on understanding and implementing the simplest camera model, the basic pinhole camera.

We will begin by describing the projection of points in 3D space onto a 2D plane using a series of matrices that we will define. The majority of this report is based on the explanations provided in *Multiple View Geometry in Computer Vision* by Richard Hartley and Andrew Zisserman. Finally, we will discuss the implementation of a specific algorithm in MATLAB to compute the camera matrix.

1.1 TODO: Notational note

To remain consistent with the textbook by Hartley and Zisserman, we will keep our notations largely similar. To emphasize a couple of matrices, I am going to

2 The Pinhole Camera Model

The pinhole camera model is a simplified representation of how a camera projects three-dimensional (3D) points in space onto a two-dimensional (2D) image plane. Under this model, we aim to map a 3D point, denoted by $\mathbf{X} = \begin{pmatrix} x & y & z \end{pmatrix}^\top$, to its corresponding point on the image plane. This point is the intersection of the line passing through $\mathbf{X}$ and the center of projection (the camera's optical center) with the image plane.

The geometry of the setup defines the center of projection at the origin of the camera coordinate system, with the image plane located at a fixed distance f (the focal length) from the origin. The projection follows the principle of similar triangles.

[[TODO: INSERT DIAGRAM HERE]]

The equations governing this projection are given by:

$$u = \frac{fx}{z}, \quad v = \frac{fy}{z}.$$

Now, note that these perspective projections are nonlinear due to the division by z. To simplify our operations, we want to transform our system using a linear model. In our case, we can accomplish this by using homogeneous coordinates.

2.1 Linearizing and using homogenous coordinates

In the homogeneous form, the coordinates of our point $\mathbf{X}$ are expressed as $\begin{pmatrix} x & y & z & 1 \end{pmatrix}^\top$, and the corresponding point on the image plane becomes $\begin{pmatrix} u & v & z \end{pmatrix}^\top$. Then, we can write

our projection in matrix form as:

$$\begin{pmatrix} u \\ v \\ z \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}.$$

This homogeneous representation serves as the foundation for deriving the complete camera matrix, which incorporates additional parameters to account for intrinsic and extrinsic camera properties.

2.2 Intrinsic Matrix

In real-world cameras, the optical center does not necessarily coincide with the center of the image plane. Instead, the projection may be offset by a certain amount. This offset is characterized by the principal point, denoted by p_x and p_y for the horizontal and vertical displacements, respectively.

Thus, the projection equations are modified to account for this offset:

$$u = \frac{fx}{z} + p_x, \quad v = \frac{fy}{z} + p_y.$$

In the homogeneous representation, this is expressed as:

$$\begin{pmatrix} u \\ v \\ z \end{pmatrix} = \begin{pmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}.$$

Here, the 3×3 submatrix

$$\mathbf{K} = \begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix}$$

is known as the *camera calibration matrix*, which encapsulates the *intrinsic parameters* of the camera, such as the focal length f and the principal point offsets, p_x and p_y.

The entire matrix is referred to as the *intrinsic matrix*. Now, our 2D coordinate $\mathbf{x}$ can be expressed as

$$\mathbf{x} = \mathbf{K} \begin{bmatrix} I & | & \vec{\mathbf{0}} \end{bmatrix} \mathbf{x}_{\text{cam}}$$

where $\mathbf{x}_{\text{cam}} = \begin{pmatrix} x & y & z & 1 \end{pmatrix}^\top$ is our camera at the origin of a Euclidean coordinate system.

2.3 Extrinsic Matrix

In addition to the intrinsic properties of a camera, we must account for its position and orientation in the world. These are described by the *extrinsic parameters*, which define the relationship between the *camera coordinate frame* and the *world coordinate frame*.

The transformation between the world coordinate frame and the camera coordinate frame involves two components:

- A 3×3 rotation matrix $\mathbf{R}$, which describes the orientation of the camera with respect to the world coordinate frame. Note that this matrix is orthogonal.
- A 3×1 translation vector $\mathbf{t}$, which specifies the position of the camera's optical center in the world coordinate frame.

The matrix $\mathbf{R}$ has three rows, each representing a basis vector of the camera coordinate frame expressed in the world coordinate frame:

- The first row of $\mathbf{R}$ represents the direction of the camera's x-axis in the world coordinate frame.
- The second row of $\mathbf{R}$ represents the direction of the camera's y-axis in the world coordinate frame.
- The third row of $\mathbf{R}$ represents the direction of the camera's optical axis (or z-axis) in the world coordinate frame.

We again express the system in terms of a homogeneous coordinate to transform a point $\mathbf{X}_\text{w}$ in the world coordinate frame to the camera coordinate frame, we apply the rotation and translation as follows:

$$\mathbf{x}_\text{cam} = \mathbf{R}\mathbf{X}_\text{w} + \mathbf{t} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_\text{w} \\ y_\text{w} \\ z_\text{w} \\ 1 \end{pmatrix}.$$

Here, the 4×4 matrix $\begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix}$ is known as the *extrinsic matrix*, combining rotation and translation into a single transformation.

3 The Camera Matrix

The camera matrix combines the intrinsic and extrinsic transformations to map a 3D point in the world coordinate frame directly onto the 2D image plane.

The intrinsic matrix $\mathbf{K}$ maps a point from the camera coordinate system to the 2D image plane. The extrinsic matrix $\begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix}$ transforms a point from the world coordinate frame to the camera coordinate frame, where the rotation matrix $\mathbf{R}$ encodes the orientation of the camera and the translation vector $\mathbf{t}$ specifies its position.

Together, the intrinsic and extrinsic matrices form our 3×4 camera matrix, $\mathbf{P}$:

$$\begin{aligned}\mathbf{P} &= \mathbf{K}\begin{bmatrix}\mathbf{R} & | & \mathbf{t}\end{bmatrix} \\ &= \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}.\end{aligned}$$

The transformation can now be expressed compactly as:

$$\mathbf{x}_{\text{image}} = \mathbf{P}\mathbf{X}_{\text{w}},$$

where $\mathbf{X}_{\text{w}}$ is a 3D point in homogeneous world coordinates, and $\mathbf{x}_{\text{image}}$ is its corresponding 2D point in homogeneous image coordinates.

Now that we have a complete overview of what the camera matrix $\mathbf{P}$ is composed of, we can compute an initial estimate for $\mathbf{P}$ using the Gold Standard algorithm outlined in the textbook (Hartley and Zisserman 181). The remainder of this report will be dedicated to computing each of the following steps.

3.1 Steps of the Gold Standard Algorithm

(1) Normalization We first normalize both the image points and the 3D space points:

- Apply a similarity transformation $\mathbf{T}$ to normalize the image points. This involves translating and scaling the points so that their centroid is at the origin and their average distance from the origin is $\sqrt{2}$.
- Similarly, apply a similarity transformation $\mathbf{U}$ to normalize the 3D space points so that their centroid is at the origin, and their average distance from the origin is $\sqrt{3}$.

(2) Direct Linear Transformation (DLT) The next step is to form a linear system of equations:

- For each correspondence $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$, use the normalized coordinates $\tilde{\mathbf{X}}_i$ and $\tilde{\mathbf{x}}_i$ to generate two equations based on the projection model $\tilde{\mathbf{x}}_i = \mathbf{P}\tilde{\mathbf{X}}_i$.
- Stack all the equations into a $2n \times 12$ matrix $\mathbf{A}$, where n is the number of correspondences.
- Write the entries of $\mathbf{P}$ as a 12-dimensional vector $\mathbf{p}$. The system $\mathbf{A}\mathbf{p} = 0$ is then solved subject to the constraint $\|\mathbf{p}\| = 1$.
- The solution is obtained from the unit singular vector of $\mathbf{A}$ corresponding to its smallest singular value.

(3) Denormalization Finally, denormalize the estimated $\mathbf{P}$ to obtain the camera matrix in the original coordinate system:

$$\mathbf{P} = \mathbf{T}^{-1}\tilde{\mathbf{P}}\mathbf{U}.$$

Here, $\mathbf{T}$ and $\mathbf{U}$ are the inverse similarity transformations used for normalization, and $\tilde{\mathbf{P}}$ is the intermediate result obtained from the DLT step.

4 Normalization

Normalization is a crucial preprocessing step in the Gold Standard Algorithm, designed to improve numerical stability when estimating the camera matrix. By scaling and centering the points appropriately, we mitigate the effects of numerical inaccuracies in the subsequent computations.

Data normalization is an essential step in the DLT algorithm and must not be considered optional (ibid. 108).

4.1 Normalizing Image Points

Given a set of image points $\mathbf{x}_i = \begin{pmatrix} u_i & v_i \end{pmatrix}^\top$ in homogeneous coordinates, the goal is to apply a similarity transformation $\mathbf{T}$ such that:

- The centroid of the points is shifted to the origin.
- The average distance of the points from the origin is $\sqrt{2}$.

To achieve this, the transformation matrix $\mathbf{T}$ is defined as:

$$\mathbf{T} = \begin{pmatrix} s & 0 & -s\bar{u} \\ 0 & s & -s\bar{v} \\ 0 & 0 & 1 \end{pmatrix},$$

where:

- $\bar{u}, \bar{v}$ are the centroid coordinates of the points, computed as:

$$\bar{u} = \frac{1}{n}\sum_{i=1}^{n} u_i, \quad \bar{v} = \frac{1}{n}\sum_{i=1}^{n} v_i.$$

- s is the scaling factor, defined to ensure the average distance from the origin is $\sqrt{2}$:

$$s = \frac{\sqrt{2}}{\frac{1}{n}\sum_{i=1}^{n}\sqrt{(u_i - \bar{u})^2 + (v_i - \bar{v})^2}}.$$

4.2 Normalizing 3D Space Points

Similarly, for a set of 3D space points $\mathbf{X}_i = \begin{pmatrix} x_i & y_i & z_i \end{pmatrix}^\top$ in homogeneous coordinates, we apply a similarity transformation $\mathbf{U}$ such that:

- The centroid of the points is shifted to the origin.
- The average distance of the points from the origin is $\sqrt{3}$.

The transformation matrix $\mathbf{U}$ is defined as:

$$\mathbf{U} = \begin{pmatrix} s_x & 0 & 0 & -s_x\bar{x} \\ 0 & s_y & 0 & -s_y\bar{y} \\ 0 & 0 & s_z & -s_z\bar{z} \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where:

- $\bar{x}, \bar{y}, \bar{z}$ are the centroid coordinates of the 3D points, computed as:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i, \quad \bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i.$$

- s_x, s_y, s_z are the scaling factors for each axis, defined to ensure the average distance from the origin is $\sqrt{3}$:

$$s_x = s_y = s_z = \frac{\sqrt{3}}{\frac{1}{n}\sum_{i=1}^{n}\sqrt{(x_i-\bar{x})^2 + (y_i-\bar{y})^2 + (z_i-\bar{z})^2}}.$$

After normalization, we have that:

- the image points $\mathbf{x}_i$ are transformed into normalized coordinates $\tilde{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i$; and
- the 3D space points $\mathbf{X}_i$ are transformed into normalized coordinates $\tilde{\mathbf{X}}_i = \mathbf{U}\mathbf{X}_i$.

The transformations $\mathbf{T}$ and $\mathbf{U}$ ensure that the numerical conditioning of the problem is improved, making the DLT step more stable and accurate.

5 Performing the direct linear transformation (DLT)

5.1 Setting up the linear system

Once the points have been normalized using the transformations $\mathbf{T}$ and $\mathbf{U}$, we estimate the camera matrix $\mathbf{P}$ in the normalized coordinate system.

For each correspondence $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{X}}_i$, the relationship between the normalized 2D image point $\tilde{\mathbf{x}}_i$ and the normalized 3D space point $\tilde{\mathbf{X}}_i$ is given by:

$$\tilde{\mathbf{x}}_i = \mathbf{P}\tilde{\mathbf{X}}_i,$$

which we write in matrix form as

$$\begin{pmatrix} \tilde{u}_i \\ \tilde{v}_i \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} \tilde{X}_i \\ \tilde{Y}_i \\ \tilde{Z}_i \\ 1 \end{pmatrix}.$$

For each corresponding point $\tilde{\mathbf{x}}_i$, we can expand the matrix to give us a pair of equations:

$$\tilde{u}_i = \frac{\mathbf{p}_1^\top \tilde{\mathbf{X}}_i}{\mathbf{p}_3^\top \tilde{\mathbf{X}}_i} = \frac{p_{11}\tilde{X}_i + p_{12}\tilde{Y}_i + p_{13}\tilde{Z}_i + p_{14}}{p_{31}\tilde{X}_i + p_{32}\tilde{Y}_i + p_{33}\tilde{Z}_i + p_{34}}, \quad \text{and}$$
$$\tilde{v}_i = \frac{\mathbf{p}_2^\top \tilde{\mathbf{X}}_i}{\mathbf{p}_3^\top \tilde{\mathbf{X}}_i} = \frac{p_{21}\tilde{X}_i + p_{22}\tilde{Y}_i + p_{23}\tilde{Z}_i + p_{24}}{p_{31}\tilde{X}_i + p_{32}\tilde{Y}_i + p_{33}\tilde{Z}_i + p_{34}}.$$

Now, we can rewrite the equations again in matrix form. For n correspondences, we stack the equations into a $2n \times 12$ matrix $\mathbf{A}$, containing all the known elements. Then, writing our camera matrix $\mathbf{P}$ as a 12-dimensional vector $\mathbf{p}$, we have a homogeneous system $\mathbf{Ap} = \mathbf{0}$:

$$\begin{pmatrix} \tilde{X}_1 & \tilde{Y}_1 & \tilde{Z}_1 & 1 & 0 & 0 & 0 & 0 & -\tilde{u}_1\tilde{X}_1 & -\tilde{u}_1\tilde{Y}_1 & -\tilde{u}_1\tilde{Z}_1 & -\tilde{u}_1 \\ 0 & 0 & 0 & 0 & \tilde{X}_1 & \tilde{Y}_1 & \tilde{Z}_1 & 1 & -\tilde{v}_1\tilde{X}_1 & -\tilde{v}_1\tilde{Y}_1 & -\tilde{v}_1\tilde{Z}_1 & -\tilde{v}_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{X}_i & \tilde{Y}_i & \tilde{Z}_i & 1 & 0 & 0 & 0 & 0 & -\tilde{u}_i\tilde{X}_i & -\tilde{u}_i\tilde{Y}_i & -\tilde{u}_i\tilde{Z}_i & -\tilde{u}_i \\ 0 & 0 & 0 & 0 & \tilde{X}_i & \tilde{Y}_i & \tilde{Z}_i & 1 & -\tilde{v}_i\tilde{X}_i & -\tilde{v}_i\tilde{Y}_i & -\tilde{v}_i\tilde{Z}_i & -\tilde{v}_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{X}_n & \tilde{Y}_n & \tilde{Z}_n & 1 & 0 & 0 & 0 & 0 & -\tilde{u}_n\tilde{X}_n & -\tilde{u}_n\tilde{Y}_n & -\tilde{u}_n\tilde{Z}_n & -\tilde{u}_n \\ 0 & 0 & 0 & 0 & \tilde{X}_n & \tilde{Y}_n & \tilde{Z}_n & 1 & -\tilde{v}_n\tilde{X}_n & -\tilde{v}_n\tilde{Y}_n & -\tilde{v}_n\tilde{Z}_n & -\tilde{v}_n \end{pmatrix} \begin{pmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{pmatrix} = 0.$$

5.2 Solving the system using singular value decomposition

To solve $\mathbf{Ap} = 0$, subject to $\|\mathbf{p}\| = 1$, we use can simply utilize the singular value decomposition (SVD), $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$.

Where $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ contains the singular values such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$, we denote the solution $\tilde{\mathbf{p}}$ to be the unit singular vector corresponding to the smallest singular value in $\mathbf{\Sigma}$, i.e., the last column of $\mathbf{V}$.

Reshaping our solution $\tilde{\mathbf{p}}$ back into a 3×4 matrix gives the normalized estimate of $\mathbf{P}$, which we shall denote as $\tilde{\mathbf{P}}$.

6 Denormalization

Finally, we then denormalize the matrix $\tilde{\mathbf{P}}$ that we just computed to obtain our camera matrix $\mathbf{P}$ in the original coordinate system:

$$\mathbf{P} = \mathbf{T}^{-1}\tilde{\mathbf{P}}\mathbf{U}.$$

7 Conclusion

The camera matrix serves as a fundamental tool in computer vision, enabling us to project 3D world points onto a 2D image plane. By combining the intrinsic parameters, which map camera coordinates to image coordinates, and the extrinsic parameters, which map world coordinates to camera coordinates, the camera matrix encapsulates the entire process of image formation in the pinhole camera model. This makes it invaluable for tasks such as 3D reconstruction, camera calibration, and image-based localization.

7.1 Limitations

To maintain the scope of this project and keep it focused as a "mini" project, certain variables and complexities have been deliberately omitted from our model.

Intrinsic parameters When defining the intrinsic matrix, we assumed that the camera pixels are square. However, in many real-world applications, pixels may be rectangular. This would require differentiating between focal lengths in the x- and y-directions, introducing f_x and f_y into the model. Moreover, the pinhole camera model does not account for lens distortions, such as radial or tangential distortions, which can significantly impact the accuracy of the projection in practical scenarios.

Number of correspondences Estimating the camera matrix involves solving for 12 unknowns, which requires at least six point correspondences between 3D world points and their 2D image projections. Accurate estimation depends on these correspondences being precise and well-distributed across the scene. Insufficient points or poorly distributed correspondences can lead to inaccurate results. Additionally, if all points lie on a single plane, a degenerate configuration may arise, resulting in an ambiguous or unreliable camera matrix $\mathbf{P}$.

Error minimization This project omits the error minimization step typically included in the Gold Standard algorithm. Minimizing geometric errors, such as reprojection error, requires iterative optimization techniques like Levenberg–Marquardt. While this step improves the accuracy of the camera matrix, it falls outside the scope of this simplified approach.

7.2 Closing Remark on the Use of the Camera Matrix

Typically, when calculating the camera matrix $\mathbf{P}$, the focus is not on $\mathbf{P}$ itself but on the information it encapsulates: the intrinsic parameters, the extrinsic parameters (rotation matrix),

and the position of the camera in the world.

The composition of the camera matrix that we detailed at the start can be exploited to reconstruct these matrices. Given that $\mathbf{K}$ is an upper-right triangular matrix and $\mathbf{R}$ is orthonormal, we can decouple $\mathbf{K}$ and $\mathbf{R}$ from the 3×3 submatrix, using QR factorization.

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \mathbf{KR}$$

Having found $\mathbf{K}$, we can then recover the translation vector $\mathbf{t}$ using the last column of $\mathbf{P}$.

$$\begin{pmatrix} p_{14} \\ p_{24} \\ p_{34} \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \mathbf{Kt} \quad \Longrightarrow \quad \mathbf{t} = \mathbf{K}^{-1} \begin{pmatrix} p_{14} \\ p_{24} \\ p_{34} \end{pmatrix}$$

By decomposing $\mathbf{P}$, we can extract the calibration matrix $\mathbf{K}$, which contains intrinsic parameters such as the focal length and principal point offset, as well as the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$, which describe the camera's orientation and position in the world coordinate frame.