# Study Guide: Camera Models, Geometry, and Real-World Aspects

This guide covers fundamental concepts related to camera operation, image formation, and the mathematical models used to describe them.

#### **The Pinhole Camera Model (Sources 1-4)**

The pinhole camera model is a foundational concept for understanding how images are formed.

**Image Formation Basics**:

* An image is created when a **light source** illuminates an **object**, and the reflected light reaches a **sensor**.
* The model helps answer how a camera creates an image by "looking" at the real world.

**The Pinhole Concept**:

* A pinhole selectively allows light rays to reach the sensor.
* **Perfect definition** (sharp image) is achieved if only one ray per point reaches the sensor, which ideally happens if the hole is just a point. This theoretical point-sized hole gives rise to the **pinhole camera model**.
* This model has been used for centuries, exemplified by the "camera obscura".

**Modeling a Camera with Pinhole Elements**:

* The pinhole model simplifies a real camera, neglecting certain effects to provide a starting point for studying geometric projection.
* **Main elements**:
* **Optical center (O)**: The location of the pinhole.
* **Image plane (Π)**: The surface where the image is formed.
* **Optical axis (Z)**: An imaginary line perpendicular to the image plane and passing through the optical center.
* **Principal point**: The intersection of the image plane and the optical axis.
* **Focal length (f)**: The distance between the optical center and the image plane.
* **Reference systems**:
* A **camera reference system (XYZ)** describes the camera's position and orientation.
* An **image plane reference system (xy)** describes positions on the image plane.
* The **right-hand rule** is used for orientation.

#### **Projective Geometry (Sources 5-20)**

Projective geometry provides a quantitative description of the camera projection process.

**Describing Projection Quantitatively**:

* It defines the relationship between a **3D point in the world (P)** and its **2D projection on the image plane (p)**.
* For easier work, projection is often considered on a plane parallel to the image plane, located in front of the optical center at distance f, which avoids the upside-down effect while maintaining the same geometrical relation.

**Field of View (FoV)**:

* The FoV defines the limits of the world framed by the camera.
* It is described by the angle 2φ, where φ is the angle under which a point P is seen.
* FoV depends on the **sensor size (d)** and the **focal length (f)**, mathematically expressed as   
  φ = arctan(d / 2f). Horizontal and vertical FoVs are commonly provided.

**Camera Projection Equations**:

* Using the similar triangle rule, the 2D projected coordinates (xp, yp) relate to the 3D world coordinates (Xp, Yp, Zp) and focal length (f) as:
* xp = f \* Xp / Zp
* yp = f \* Yp / Zp
* **Important**: Projecting 3D points onto a 2D surface leads to the **loss of distance information (Zp)**.

**Homogeneous Coordinates and Projection Matrix**:

* **Homogeneous coordinates** are a "mathematical trick" to extend N-dimensional points into (N+1) coordinates, allowing geometric transformations to be represented as matrix multiplications.
* The projection equations can be rewritten in **matrix form** using homogeneous coordinates: m̃ ≈ P M̃.
* **P** is the **projection matrix**, which describes how the 3D world is mapped onto the image plane. For a basic pinhole model with focal length f, P = [f 0 0 0; 0 f 0 0; 0 0 1 0]. When f=1, this represents the essential perspective projection.

**Mapping to Image Coordinates (Pixels)**:

* The projected x, y coordinates (metric distances) need to be converted to pixel coordinates (u, v) for digital images.
* The origin of pixel coordinates is typically the **top-left corner**.
* Conversion factors ku = 1/w (pixel width) and kv = 1/h (pixel height) are used.
* The mapping includes the principal point coordinates (u0, v0):
* u = u0 + x\_p / w = u0 + k\_u \* x\_p
* v = v0 + y\_p / h = v0 + k\_v \* y\_p
* Combining these with the projection equations yields:
* u = u0 + f\_u \* Xp / Zp
* v = v0 + f\_v \* Yp / Zp
* where f\_u = k\_u \* f and f\_v = k\_v \* f are the **focal lengths in pixels**.

**Camera Matrix (Intrinsic Parameters)**:

* The combined projection can be expressed using the **camera matrix K** (also called the intrinsic matrix): P = K [I | 0].
* K = [fu 0 u0; 0 fv v0; 0 0 1].
* The elements of K (fu, fv, u0, v0, (and implicitly ku, kv), f) are called **intrinsic parameters**. They define the inherent projection characteristics of the camera.

**Camera vs. Real World (Extrinsic Parameters)**:

* To relate the camera's reference frame to a separate world reference frame, a **rototranslation matrix T** is used.
* T = [R t; 0 1], where R is a **rotation matrix** and t is a **translation vector**.
* The complete projection process then becomes: m̃ ≈ P T M̃.
* The parameters of T (3 for translations, 3 for rotations) are called **extrinsic parameters**. They define the relationship between the camera and the world.

**Projection Recap**:

* The full projection process involves three transformations:

1. **3D to 3D**: From world coordinates to camera coordinates (rototranslation).
2. **3D to 2D**: From camera coordinates to the sensor plane (projection).
3. **2D to 2D**: From sensor plane to sensor coordinates (pixels) (scaling and translation).

**Inverse Projection**:

* The inverse transformation (from 2D image back to 3D world) is generally **not invertible**.
* This non-invertibility is due to the **loss of the Z-dimension** during 3D-to-2D projection and **pixel quantization**.
* Inverse projection is possible if:
* The result is accepted as the **direction of the object** rather than its exact 3D position.
* **Additional constraints** (e.g., knowledge of a ground plane) provide location along the ray.
* The **quantization effect is neglected** (assuming pixel location is the same as the projected point), which is acceptable for high-resolution sensors.

#### **Cameras and Lenses (Sources 21-29)**

Real cameras use lenses to overcome the limitations of the simple pinhole model, but lenses introduce their own complexities.

**Pinhole Limitations & Adding a Lens**:

* The pinhole model has a **trade-off**: a smaller hole gives sharper images but lets in less light (low intensity).
* A **lens** provides sharp images without needing a tiny pinhole, gathering more light.

**Thin Lens Model**:

* A lens is added, centered on the pinhole, with its main axis on the optical axis.
* A "thin" lens is modeled as a 2D plane where light deviation occurs, neglecting its width.
* **Effect of a lens**: Lenses deviate light rays to **focus** them.
* Rays passing through the center of the lens are **not deviated**.
* Rays parallel to the lens axis are deviated through the **focal point**.
* **Thin Lens Equation**: This equation relates the object distance (d0), image distance (d1), and lens focal length (f): 1/d0 + 1/d1 = 1/f.

Immagine che contiene testo, diagramma, Diagramma, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

**Focus and Depth of Field (DoF)**:

* Points at a specific distance are **in focus**. Objects at other distances appear out of focus, creating a **circle of confusion**.
* Changing the distance between the lens and the sensor alters the focus characteristics.
* **Depth of Field** is the range of distances within which objects appear acceptably in focus, meaning the circle of confusion is negligible.
* **Aperture controls DoF**: A **smaller aperture (larger f-number)** increases the depth of field, allowing more of the scene to be in focus.

**Focal Length Comparison**:

* The term "focal length" has two meanings:
* In a **thin lens**, it's the distance where parallel rays intersect.
* In the **pinhole camera model**, it's the distance between the pinhole and the sensor.

**Non-Ideal Lenses: Distortion and Aberrations**:

* Real lenses are not ideal and introduce effects like **distortion** and **chromatic aberrations**.
* **Distortion** is a deviation from ideal projection behavior.
* **Radial distortion**: The amount of distortion depends on the distance of a point from the image center. It causes straight lines to appear curved. It's often modeled by a polynomial function using coefficients k1, k2, k3 (e.g., x\_corr = x \* (1 + k1\*r^2 + k2\*r^4 + k3\*r^6)).
* **Tangential distortion**: Caused by non-ideal alignment between the lens and sensor. It is usually negligible and modeled with p1, p2 parameters.
* **Chromatic aberration (dispersion)**: The refractive index of the lens depends on the light's wavelength, causing different colors to focus at different points. This results in **color fringes** near image edges.
* Camera models can be coupled with distortion estimation to compensate for these effects.

#### **Camera Calibration (Sources 30-38)**

Camera calibration is the process of estimating the intrinsic and extrinsic parameters of a camera.

**What is Camera Calibration?**:

* It's the process of **estimating camera parameters**, including **intrinsic parameters** (which define the camera's internal geometry and lens properties, like focal length, principal point, and distortion coefficients) and **extrinsic parameters** (which define the camera's position and orientation in the world).
* Calibration is crucial for quantitative applications, such as measuring projection characteristics or relating image points to the real world (e.g., determining distances).
* Camera orientation can be expressed using angles like **yaw, pitch, and roll** (or pan and tilt).

**The Calibration Process Outline**:

* **General steps**:

1. Acquire an **object of known shape and dimensions** (a **calibration pattern**, e.g., a checkerboard, where corners serve as easily recognizable points).
2. Take **multiple pictures (N images)** of this pattern from different viewpoints.
3. For each image, identify the **3D positions of the pattern's corners** in its own reference system (P\_i,j) and their corresponding **2D pixel coordinates** in the image (p\_i,j).
4. **Initialize intrinsic parameters** (fu, fv, u0, v0, and distortion coefficients k1, k2, k3, p1, p2) and **extrinsic parameters** (rototranslation T) to default values.
5. Solve a **non-linear least squares problem** to minimize the difference between the observed 2D image points and the projected 3D world points: min || K [I|0] T\_i P\_i,j - p\_i,j ||^2.

**Challenges and Improvements**:

* The minimization process can be unstable due to the large number of parameters.
* **Good initial guesses** are highly desirable.
* **Homography** can provide a good initial guess: For a planar object (like a checkerboard) and neglecting distortion, the transformation from the object to the image plane can be represented by a homography. (*Se l’oggetto che stai inquadrando è piatto (come una scacchiera), e trascuri la distorsione della lente, allora il modo in cui l’oggetto si proietta sull’immagine può essere descritto bene da una omografia (homography)).  
  (Una omografia è una matrice 3×3 che descrive come un piano si trasforma da una vista all’altra).*
* **Number of views needed**:
* Each view of a planar object (e.g., checkerboard) provides **8 constraints** (from 4 "free" corner points). More points are useful for measuring distortion.
* Neglecting distortion, a minimum of **2 views** are theoretically needed to determine the 4 or 5 intrinsic and 6 extrinsic parameters.
* In practice, a **larger number of views (e.g., 10 or more)** is needed for a stable and accurate calibration due to the instability of the minimization process.

#### **Image Mapping (Sources 39-46)**

Image mapping deals with how pixels are transformed spatially in an image.

**Geometric Transforms**:

* A geometric transform modifies the **spatial relationship among pixels**.
* It involves two steps: **coordinate transformation** (e.g., x', y' = T(x, y)) and **image mapping/resampling**.

**Forward Mapping**:

* Directly applies the transformation from **source pixels** to **destination pixels**.
* A significant drawback is that it can **leave empty pixels** in the destination image if multiple source pixels map to the same destination, or if transformed coordinates fall between integer pixel locations.

**Inverse/Backward Mapping**:

* Inverts the process: For each pixel in the **destination image**, it calculates the corresponding location in the **source image**.
* **Advantages**:
* Every pixel at the destination is visited once.
* Every pixel at the destination is covered by the transformation, ensuring no empty pixels.
* **Interpolation** is used to determine the pixel value when the calculated source location falls between discrete pixel centers. Common interpolation methods include nearest neighbor, bilinear, and bicubic.

**Look-Up Table (LUT)**:

* If the same mapping is applied to many images, the locations of the source pixels can be pre-calculated and saved into a **Look-Up Table (LUT)** (one for x-values, one for y-values). This avoids re-evaluating the transformation repeatedly.

**OpenCV Functions**:

* Libraries like OpenCV provide functions for image mapping, such as cv::warpAffine(), which handles affine transformations and manages the low-level details of backward mapping and interpolation automatically.
* These functions require a transformation matrix (e.g., a 2x3 matrix for affine transforms) as input.

#### **Real Cameras (Sources 47-58)**

This section discusses additional practical considerations and effects in real-world cameras beyond the idealized models.

**Perspective Projection Consequences**:

* Cameras fundamentally act as "dimensionality reduction machines," projecting the 3D world onto a 2D surface.
* This transformation results in the **loss of certain information**: angles, distances, and parallelism (though straight lines remain straight).

**The Role of Lenses**:

* Lenses allow cameras to **gather more light** compared to a pinhole.
* They also necessitate the concept of **focus**.

**Perspective vs. Viewpoint & Focal Length**:

* While focal length changes the subject size in the image, the **viewpoint (camera distance to subject)** also significantly affects perspective.
* One can compensate for a change in focal length by moving the viewpoint to keep the subject size constant, but this will alter the background and potentially introduce different distortions (e.g., the "Vertigo effect").
* A **large FoV (small focal length)** means the camera is close to the subject, leading to more distortion.
* A **small FoV (large focal length)** means the camera is further from the subject, leading to less distortion.

**Image Sensors**:

* Modern digital cameras use **sensors (CCD or CMOS)** to convert photons into electrons. These are essentially grayscale sensors.
* **Sensing colors**:
* **3-chip color sensors**: Use a trichroic prism to split incident light, sending R, G, and B components to separate dedicated imagers.
* **Single-chip color sensors**: Use a **Bayer Color Filter Array (CFA)** or mosaic pattern on a single imager, requiring **interpolation (demosaicing)** to provide complete color information for each pixel.
* **Direct image sensors (e.g., Foveon)**: Do not require interpolation, minimizing information loss.

**Exposure Control: Aperture and Shutter Speed**:

* **Exposure** (the total amount of light reaching the sensor) is controlled by two main parameters: **aperture** and **shutter speed**.
* **Aperture**:
* The diameter of the lens opening.
* Expressed as an **f-number (f/N)**, where A = f/N (A is aperture diameter, f is focal length, N is f-number).
* A **smaller f-number (e.g., f/2.0)** means a **larger aperture opening**.
* **F-numbers are used** because if the f-number (f/N) is constant, the amount of light gathered is constant regardless of the lens's focal length.
* Typical f-number progression (e.g., f/2.0, f/2.8, f/4) ensures that each stop roughly **halves or doubles the amount of light**.
* **Shutter Speed**:
* The **exposure time**: the duration for which the sensor is exposed to light.
* Typical values include fractions of a second (e.g., 1/60s, 1/1000s).
* **Electronic shutters** are used in video cameras, controlling exposure electronically.
* **Global shutter**: All pixels are acquired simultaneously.
* **Rolling shutter**: The image is acquired row by row, which can lead to distortion with fast motion.
* **Reciprocity**: Various combinations of shutter speed and aperture can yield the **same overall exposure**.
* **Choice of shutter speed** affects motion: **shorter exposure** (faster shutter speed) **freezes motion**, while **longer exposure** (slower shutter speed) results in **motion blur**.
* **Choice of aperture** impacts **depth of field** and diffraction effects.  
  Large aperture = Small DoF  
  Small aperture = Large DoF