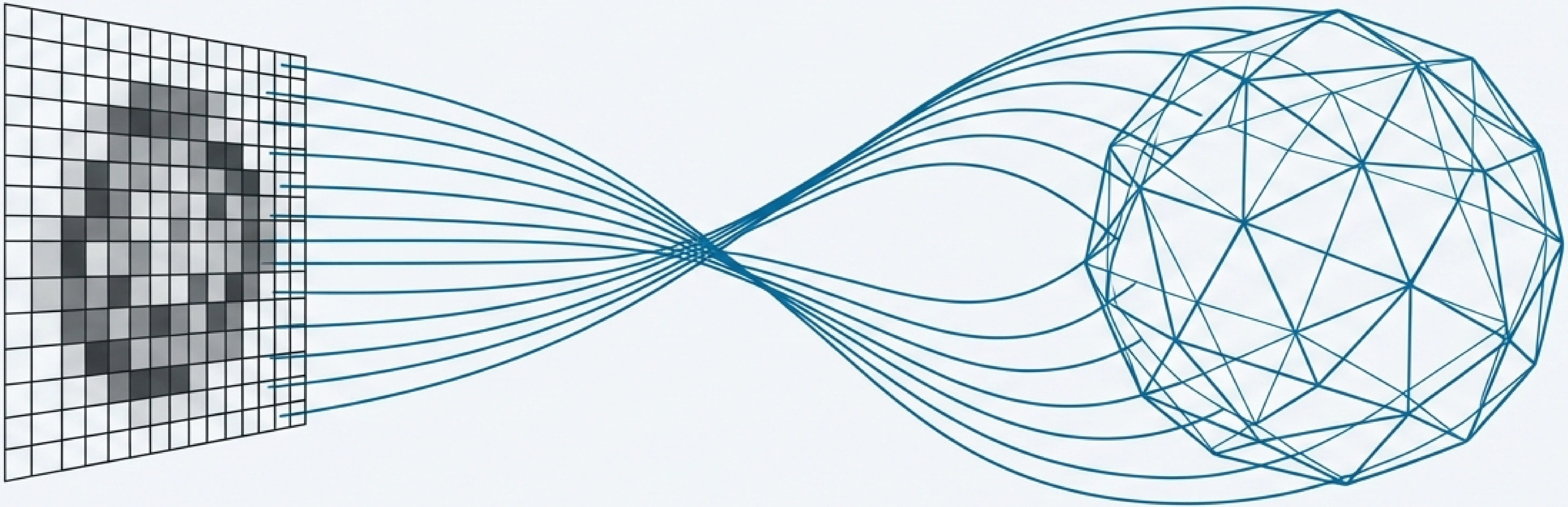
From Pixels to Reality: A Practical Guide to 3D Computer Vision

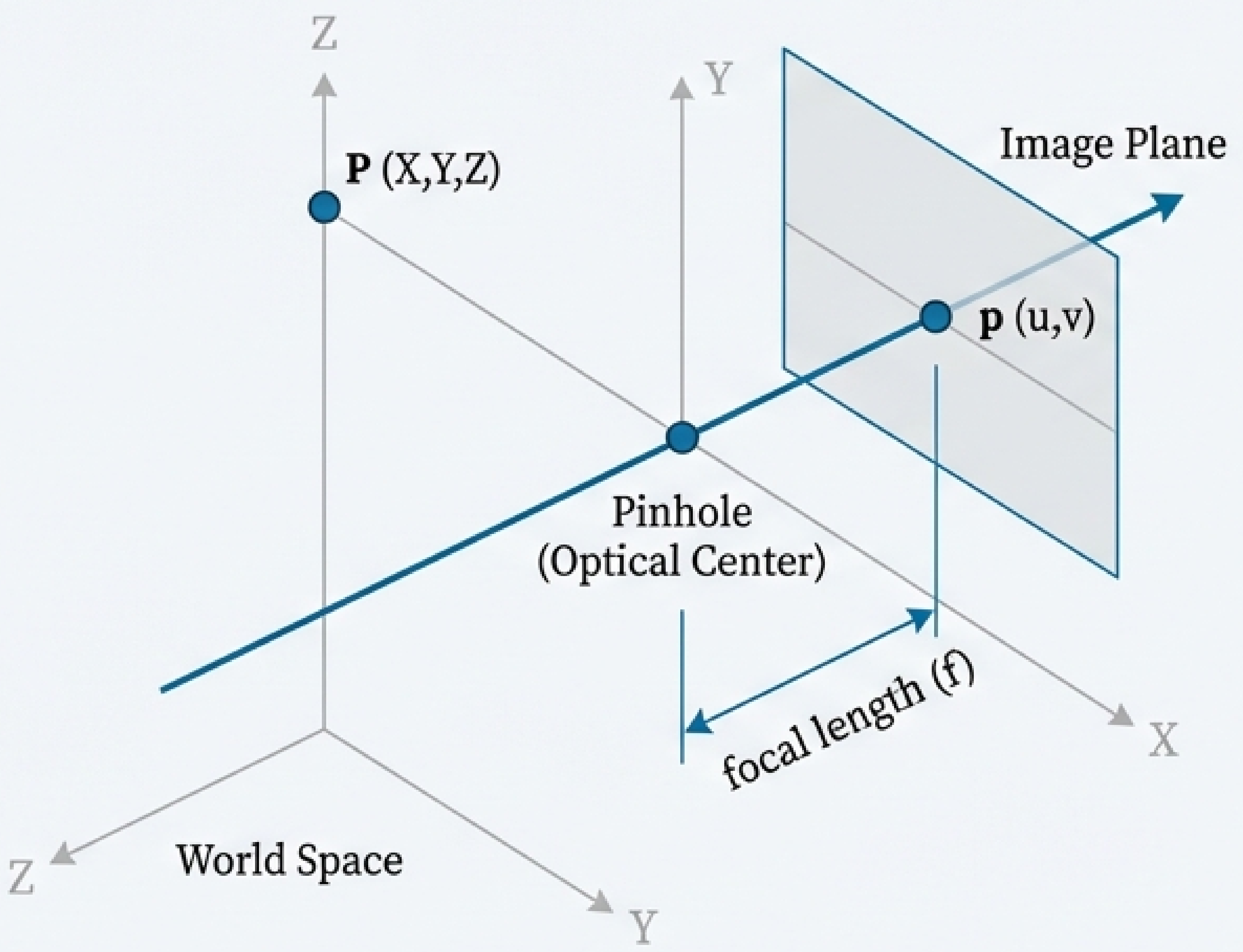
Mastering Camera Calibration to Unlock 3D Perception



This presentation outlines the fundamental journey of 3D computer vision: starting with an imperfect camera image and ending with a precise 3D reconstruction of the world. We will explore the models, correct for real-world flaws, and unlock the applications that allow machines to perceive depth and structure.

Step 1: Modeling the Perfect Camera

The Pinhole Model: Projecting a 3D World onto a 2D Plane



The pinhole model is the simplest mathematical representation of a camera. It describes how light rays from a 3D point pass through a single aperture to form an image. This gives us a direct geometric relationship between the 3D world and the 2D image.

$$s \times \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}^{T} = \mathbf{K} \times [\mathbf{R}|\mathbf{t}] \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}^{T}$$

Scale factor

Intrinsic Matrix

Extrinsic Matrix

Deconstructing the Model: The Camera's Identity

Intrinsic Parameters (K) - The Camera's DNA

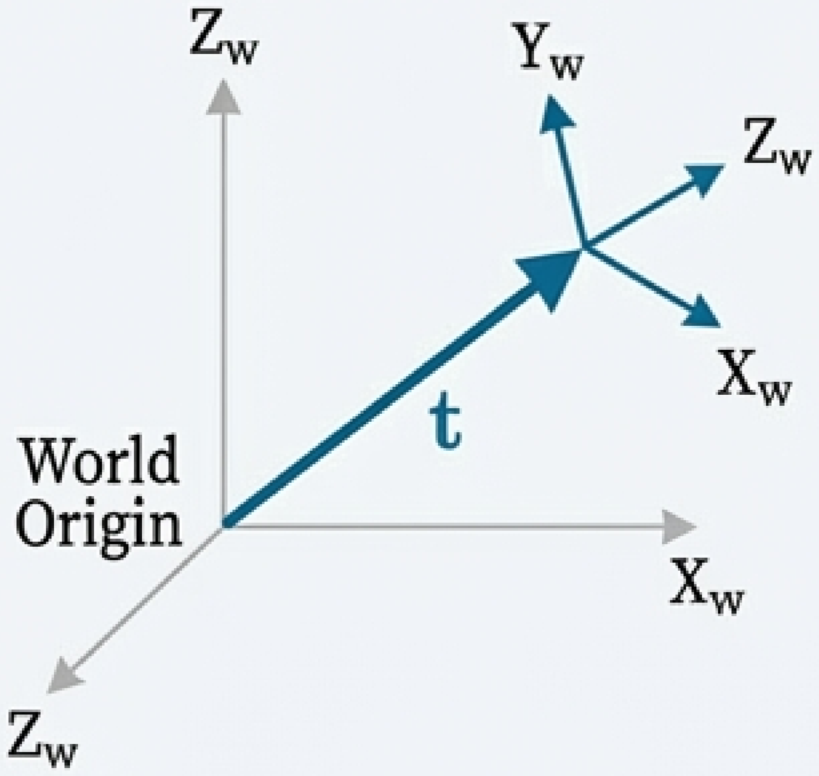
These parameters are internal to the camera and define its optical properties. They do not change unless the camera's lens or sensor configuration configuration changes.

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

f_x, f_y	Focal length in pixel units.
c_x, c_y	Principal point, the optical center of the image sensor.

Extrinsic Parameters [R|t] - The Camera's Place in the World

These parameters define the camera's position and orientation relative to a world coordinate system. They change every time the camera moves.

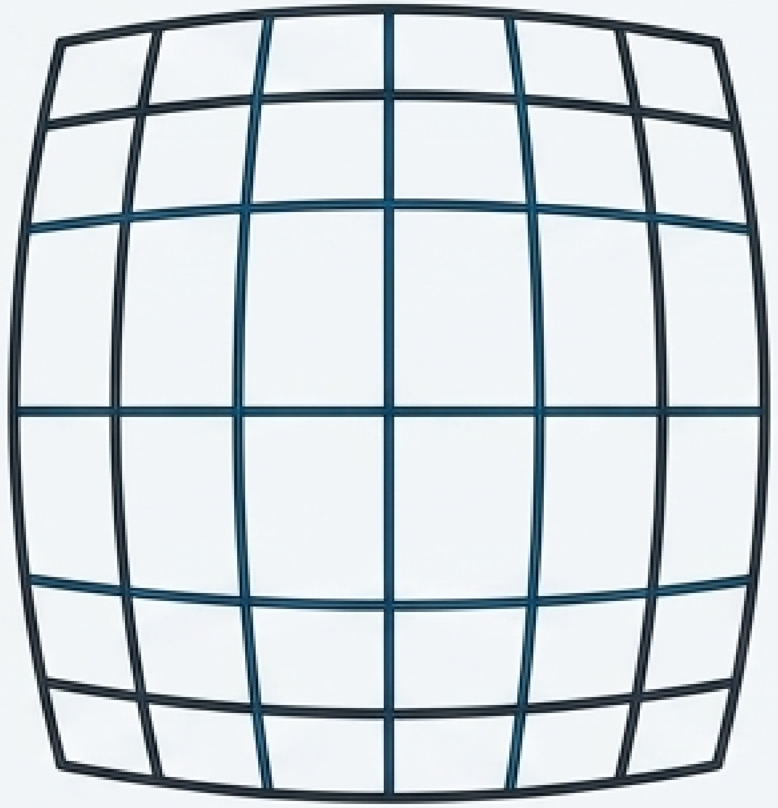


R	3x3 Rotation Matrix (orientation).
t	3x1 Translation Vector (position).

Confronting Reality: The Imperfections of Physical Lenses

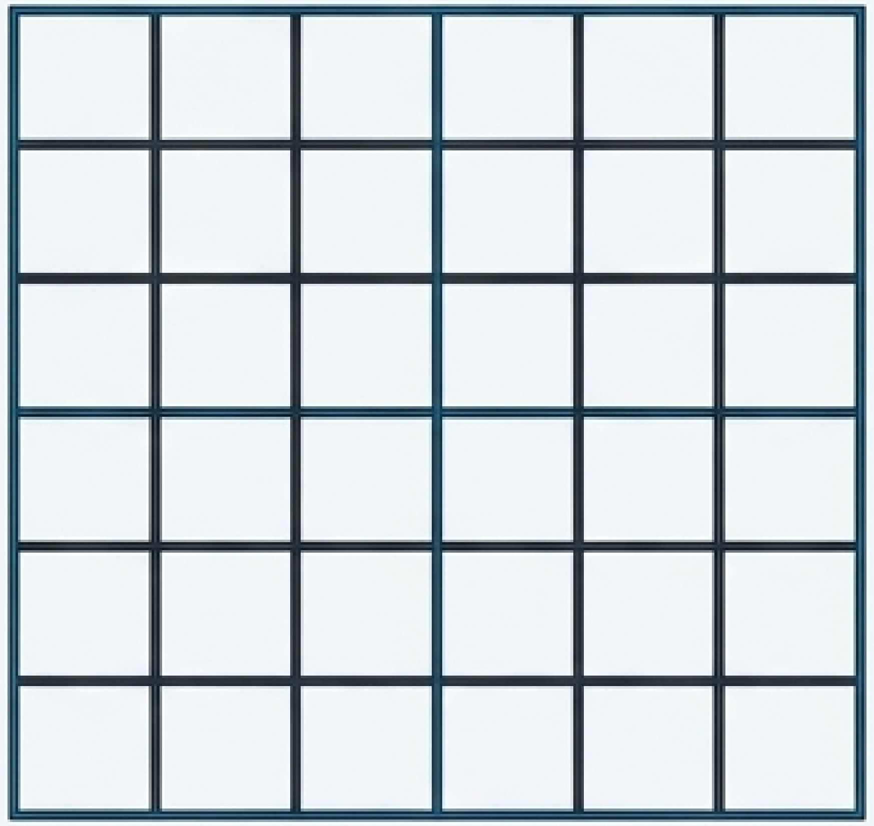
Real lenses do not follow the perfect pinhole model. They bend light rays incorrectly, causing predictable geometric distortions in the image.

Barrel Distortion ($k_1 > 0$)

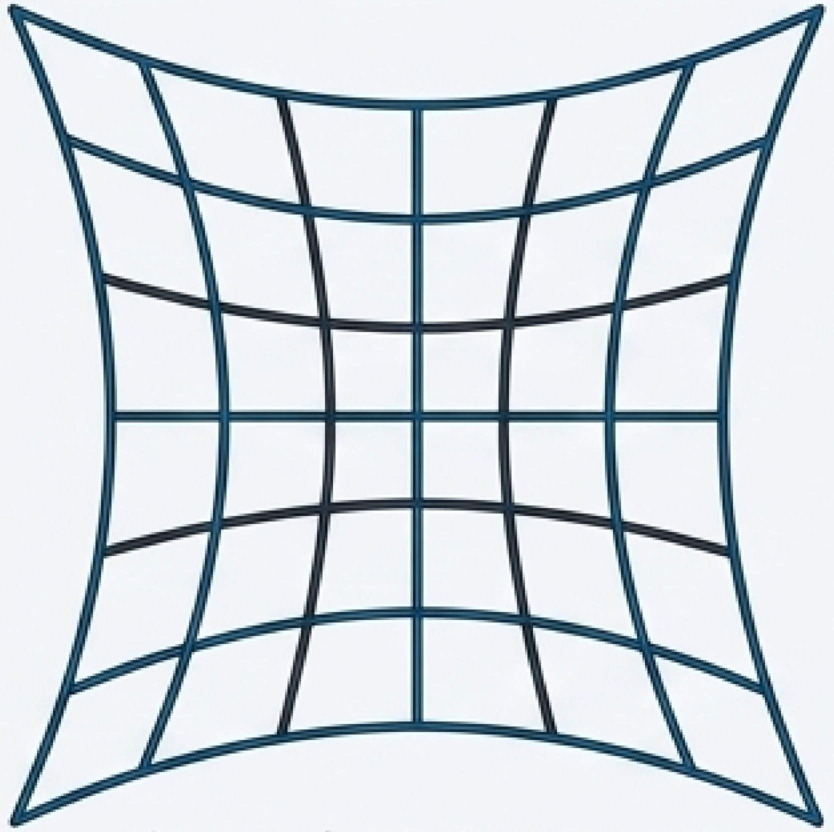


Common in wide-angle lenses.

No Distortion (Ideal)



Pincushion Distortion ($k_1 < 0$)



Common in telephoto lenses.

The Math of Distortion

Radial Distortion

The most significant type, causing lines to curve.

$$x_{\text{distorted}} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{\text{distorted}} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

Where $r^2 = x^2 + y^2$.

Tangential Distortion

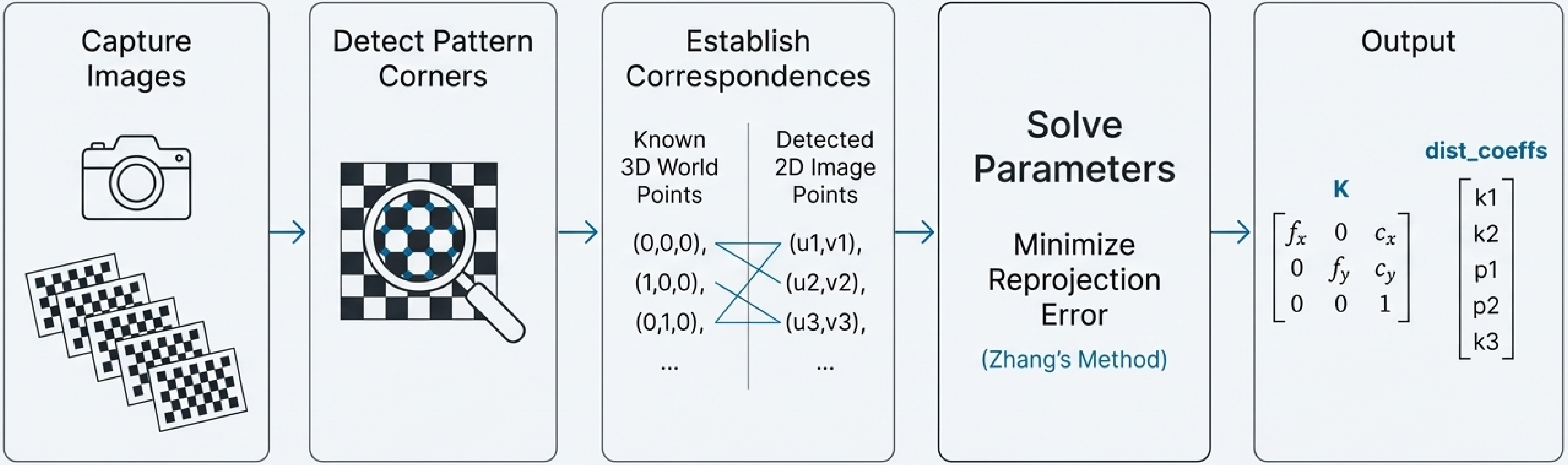
Occurs when the lens is not perfectly parallel to the sensor.

$$x_{\text{distorted}} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$
$$y_{\text{distorted}} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$

The Goal: We must find the distortion coefficients $[k_1, k_2, p_1, p_2, k_3]$ to reverse these effects.

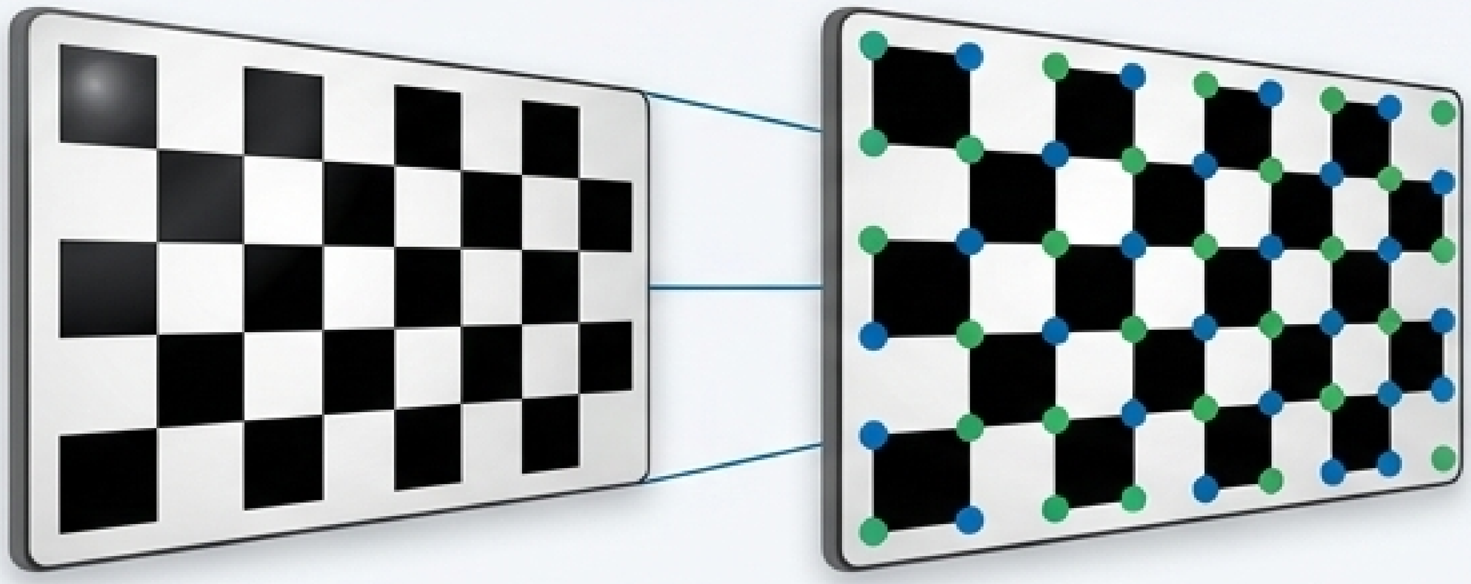
The Solution: A Systematic Calibration Process

The objective of calibration is to find the camera's intrinsic matrix **K** and its distortion coefficients `dist_coeffs` by analyzing images of a known pattern.



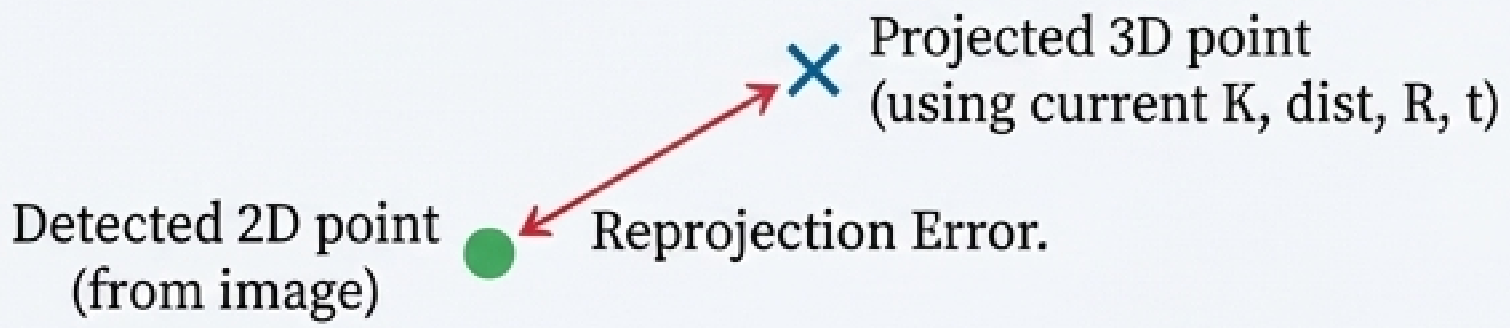
Calibration in Practice: From Checkerboards to Code

Corner Detection and Sub-Pixel Refinement



Use `findChessboardCorners()` to locate initial points, then `cornerSubPix()` to refine them to sub-pixel accuracy.

Reprojection Error & Quality Check



$$\text{error} = \frac{1}{N} \times \sum ||\text{projected\_point} - \text{detected\_point}||^2$$

A good calibration has a reprojection error of less than 0.5 pixels.

Core OpenCV Function

```
# The final step: solving for all parameters
ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(
    objpoints,                          # List of 3D world points for each image
    imgpoints,                          # List of corresponding 2D image points
    imageSize,
    None, None
)
```

is the intrinsic matrix K

is the distortion vector.

NotebookLM

The First Reward: Achieving a Perfected View

Using the calculated parameters to undistort any image from the camera.

Original Distorted Image

Apply **K** and **dist_coeffs**

Undistorted Image

Method 1: Direct (Simple)

Simple one-line function call, suitable for single images.

```
undistorted = cv2.undistort(image, mtx, dist)
```

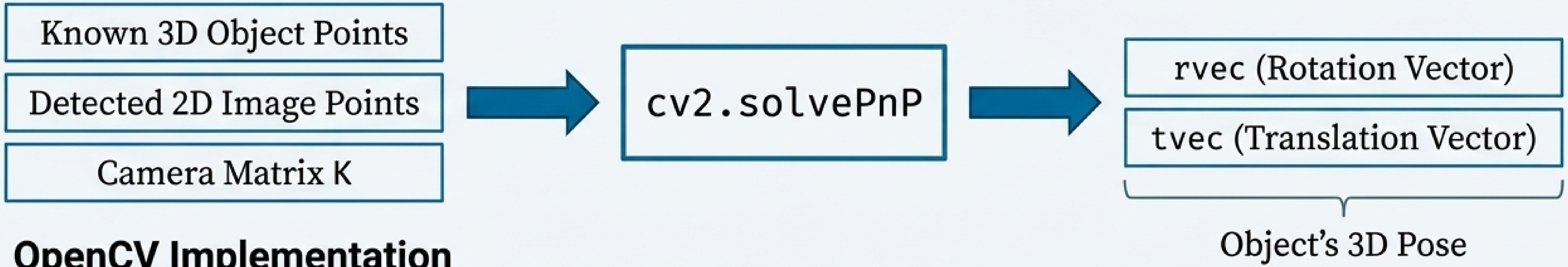
Method 2: Remapping (Fast for Video)

More efficient for real-time applications. A mapping is pre-computed once and applied rapidly to each new frame.

```
# One-time setup
mapx, mapy = cv2.initUndistortRectifyMap(...)
# Fast, per-frame application
undistorted = cv2.remap(image, mapx, mapy, cv2.INTER_LINEAR)
```

Unlocking 3D Perception: Estimating Object Pose

The Perspective-n-Point problem: Given a calibrated camera, how do we find the 3D pose (**Rotation** and **Translation**) of an object if we know the correspondence between its 3D points and their 2D projections in the image?



OpenCV Implementation

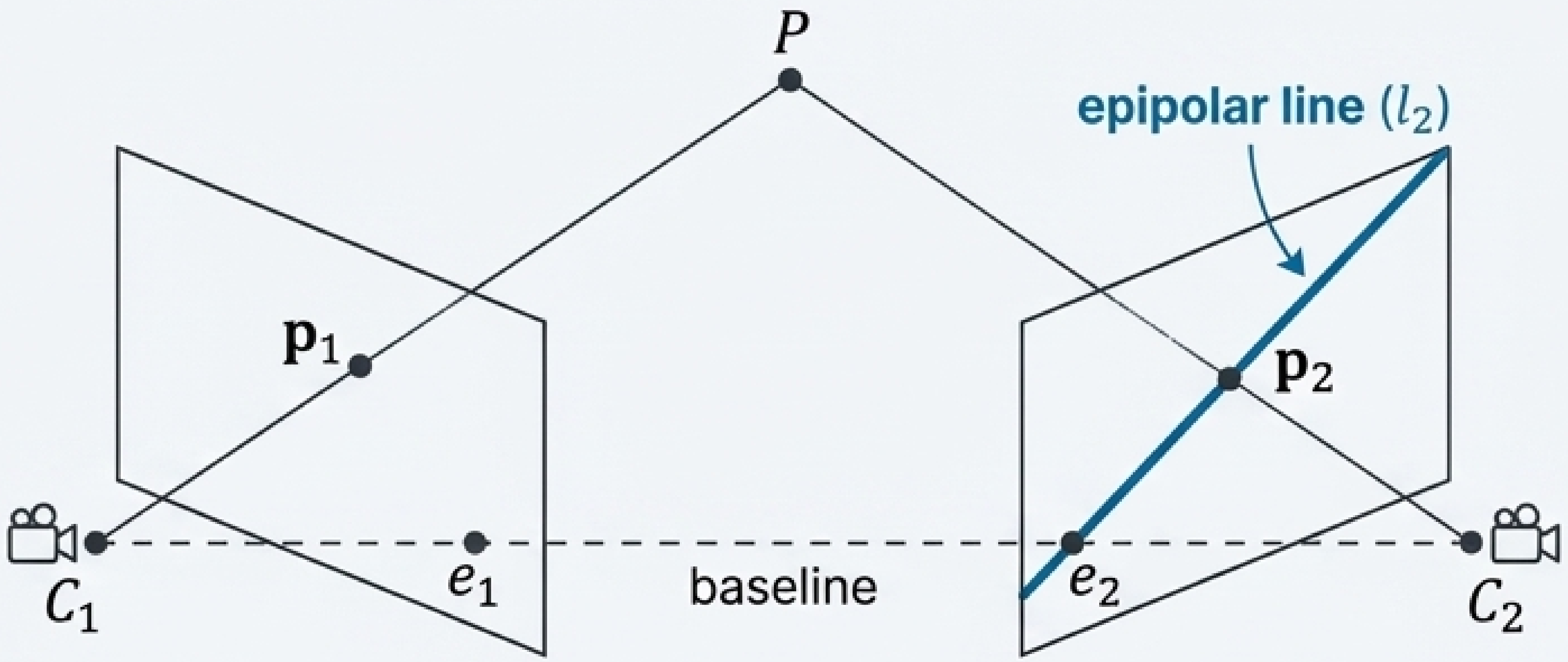
```
# We provide the known 3D points and their 2D locations
success, rvec, tvec = cv2.solvePnP(
    obj_points, img_points,
    camera_matrix, dist_coeffs
)

# Convert rotation vector to a more usable 3x3 matrix
R, _ = cv2.Rodrigues(rvec)
```

Note: For robustness against incorrect point matches, `cv2.solvePnPRansac` is often preferred.

The Geometry of Two Views: Epipolar Constraint

When a 3D point P is viewed by two cameras, its projection $\mathbf{p}_1$ in the first image provides a powerful constraint on where to find its projection $\mathbf{p}_2$ in the second image.



Given point $\mathbf{p}_1$, its corresponding point $\mathbf{p}_2$ **must** lie on the epipolar line l_2.
This reduces the search for matches from a 2D area to a 1D line.

Fundamental Matrix (F)

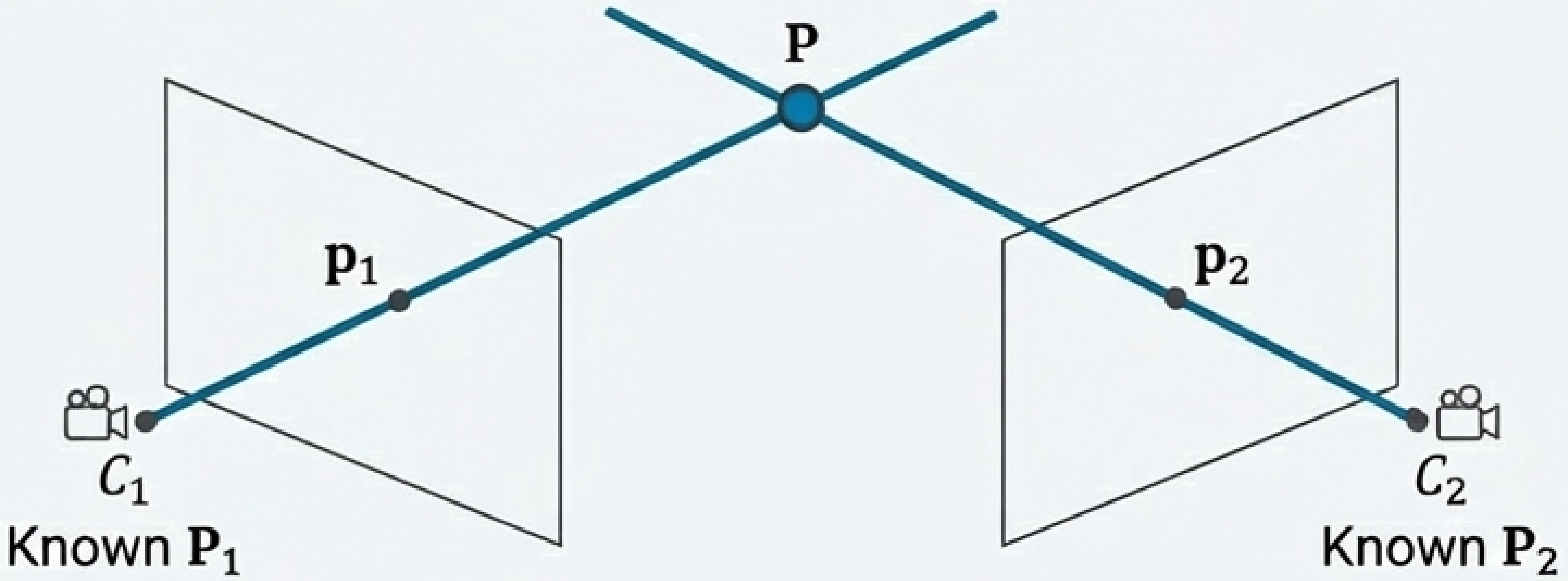
Relates pixel coordinates. $\mathbf{p}_2^{\mathrm{T}}\mathbf{F}\mathbf{p}_1 = 0.$
Use for uncalibrated cameras.

Essential Matrix (E)

Relates normalized camera coordinates. $\mathbf{x}_2^{\mathrm{T}}\mathbf{E}\mathbf{x}_1 = 0.$
Requires camera intrinsic matrix $\mathbf{K}$. $\mathbf{E}$ directly contains the relative rotation and translation between the cameras.

Reconstructing the World: Triangulation

Triangulation is the process of determining a 3D point's location by finding the intersection of two rays originating from the camera centers and passing through the point's 2D projections.



OpenCV Implementation

```
# P1 and P2 are the 3x4 projection matrices for each camera
# P1 = K @ [R1|t1], P2 = K @ [R2|t2]


points_4d = cv2.triangulatePoints(P1, P2, pts1, pts2)

# Convert from homogeneous (4D) to 3D coordinates
points_3d = (points_4d[:3] / points_4d[3]).T
```

A Complete Application: The Stereo Vision Pipeline

Goal: To estimate the depth of every pixel in a scene using a calibrated stereo camera pair.

$$disparity = x_{left} - x_{right}$$

$$depth = \frac{focal\_length \times baseline}{disparity}$$

[Step 1] Stereo Calibrate

Determine intrinsics for both cameras and the extrinsics (`R`, `t`) between them.

[Step 2] Stereo Rectify

Warp both images so that epipolar lines become horizontal. This makes matching points trivial, as they will lie on the same row.

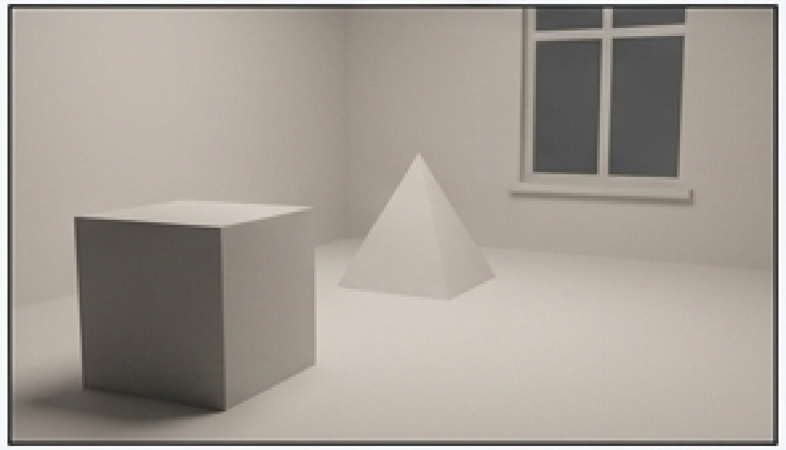
[Step 3] Compute Disparity Map

For each pixel in the left image, find the matching pixel on the same row in the right image. The horizontal distance is the disparity. (Use `StereoBM` or `StereoSGBM`).

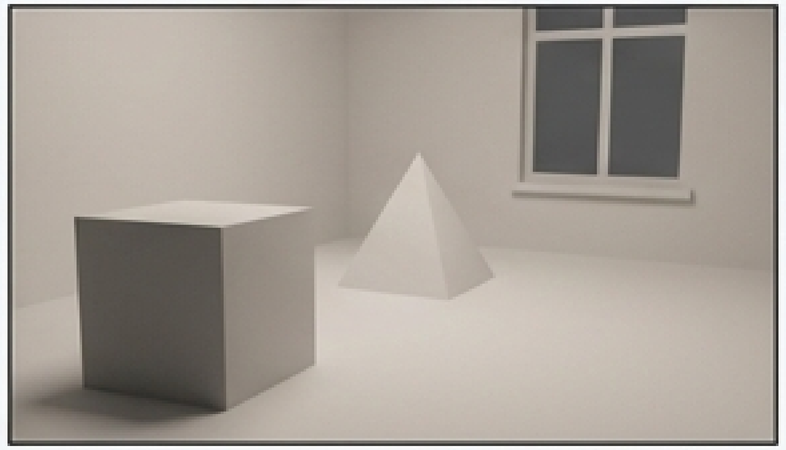
[Step 4] Calculate Depth Map

Use the core equation to convert the disparity map into a depth map.

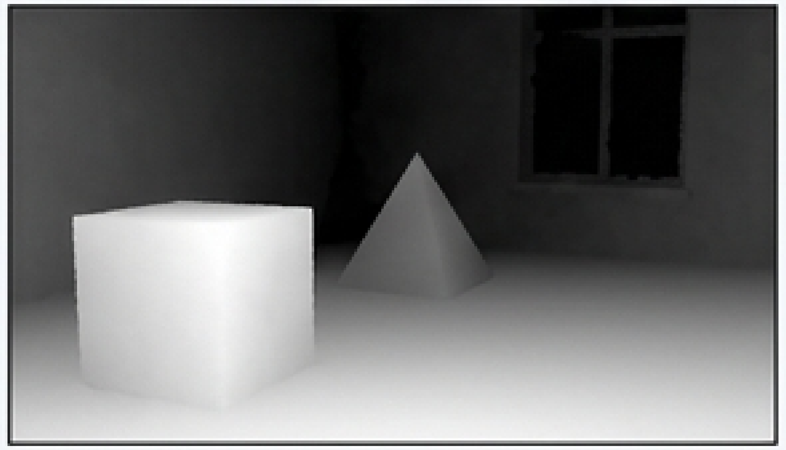
Left Image



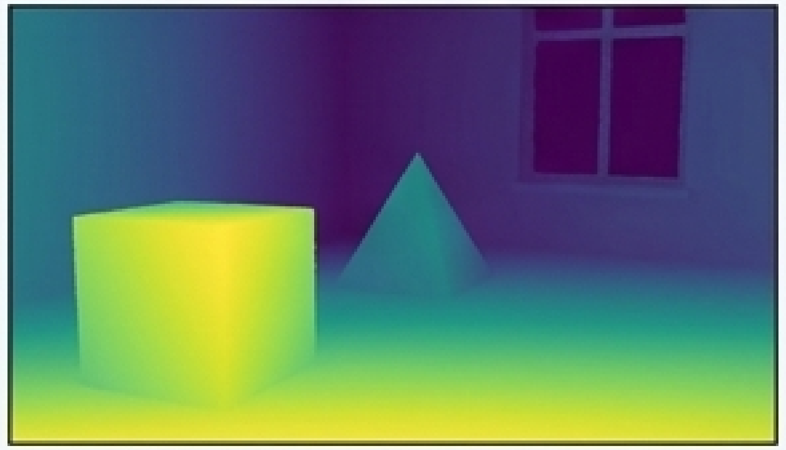
Right Image



Disparity Map



Depth Map



The Grand Synthesis: Structure from Motion

SFM simultaneously reconstructs the 3D structure of a scene (Structure) and estimates the camera poses for all input images (Motion) from an unordered image collection.

The SFM Pipeline

Feature Detection & Matching
Find **keypoints** (SIFT, ORB) in all images and match them across pairs.

↓

Geometric Verification
Estimate the **Fundamental Matrix (F)** between image pairs using RANSAC to filter out bad matches. If the camera is calibrated, compute the **Essential Matrix (E)**.

↓

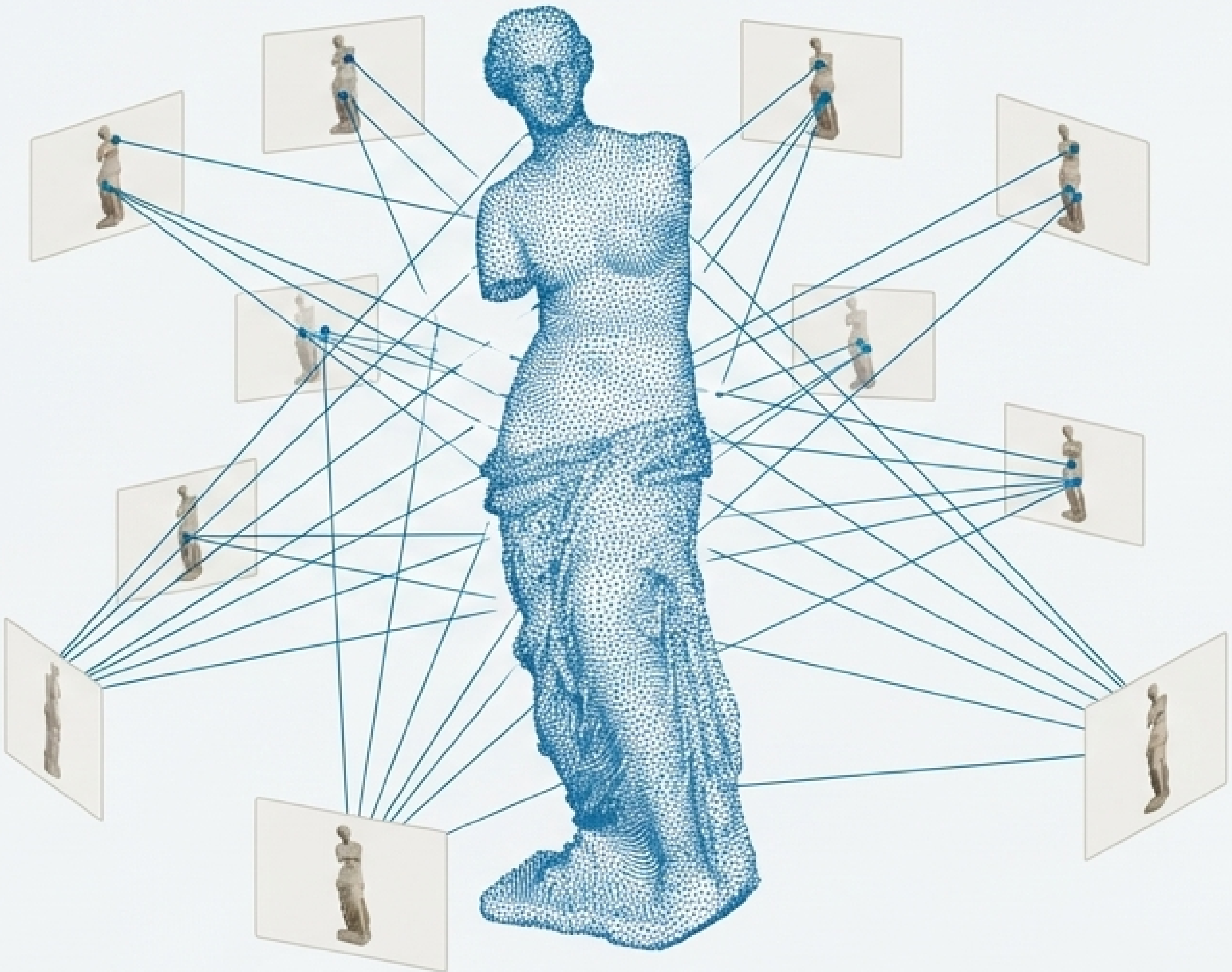
Initial Pair Reconstruction
Recover Pose: Decompose the Essential Matrix **E** to get the relative **R** and **t** for the first two cameras.
Triangulate: Create the initial 3D point cloud.

↓

Incremental Reconstruction
For each new image, find its pose using **solvePnP** (PnP).
Triangulate new 3D points.

↓

Refinement (Bundle Adjustment)
A global optimization step that minimizes the reprojection error across *all* cameras and *all* 3D points simultaneously, creating a highly accurate and consistent model.



The Computer Vision Toolkit: A Function Reference

Calibration & Undistortion	Pose Estimation	Multi-View Geometry	Stereo Vision
• cv2.findChessboardCorners() • cv2.cornerSubPix() • cv2.calibrateCamera() • cv2.undistort() • cv2.initUndistortRectifyMap()	• cv2.solvePnP() • cv2.solvePnPRansac() • cv2.Rodrigues() • cv2.projectPoints()	• cv2.findFundamentalMat() • cv2.findEssentialMat() • cv2.recoverPose() • cv2.triangulatePoints() • cv2.computeCorrespondEpilines()	• cv2.stereoCalibrate() • cv2.stereoRectify() • cv2.StereoSGBM_create() • cv2.reprojectImageTo3D()

Continuing the Journey: Key Resources

Foundational Papers & Books

- "A Flexible New Technique for Camera Calibration" - Z. Zhang (The method behind `calibrateCamera`).
- "Multiple View Geometry in Computer Vision" - Hartley & Zisserman (The definitive academic reference).

Official Tutorials

- OpenCV Camera Calibration Tutorial
- OpenCV Epipolar Geometry Tutorial
- OpenCV Pose Estimation Tutorial

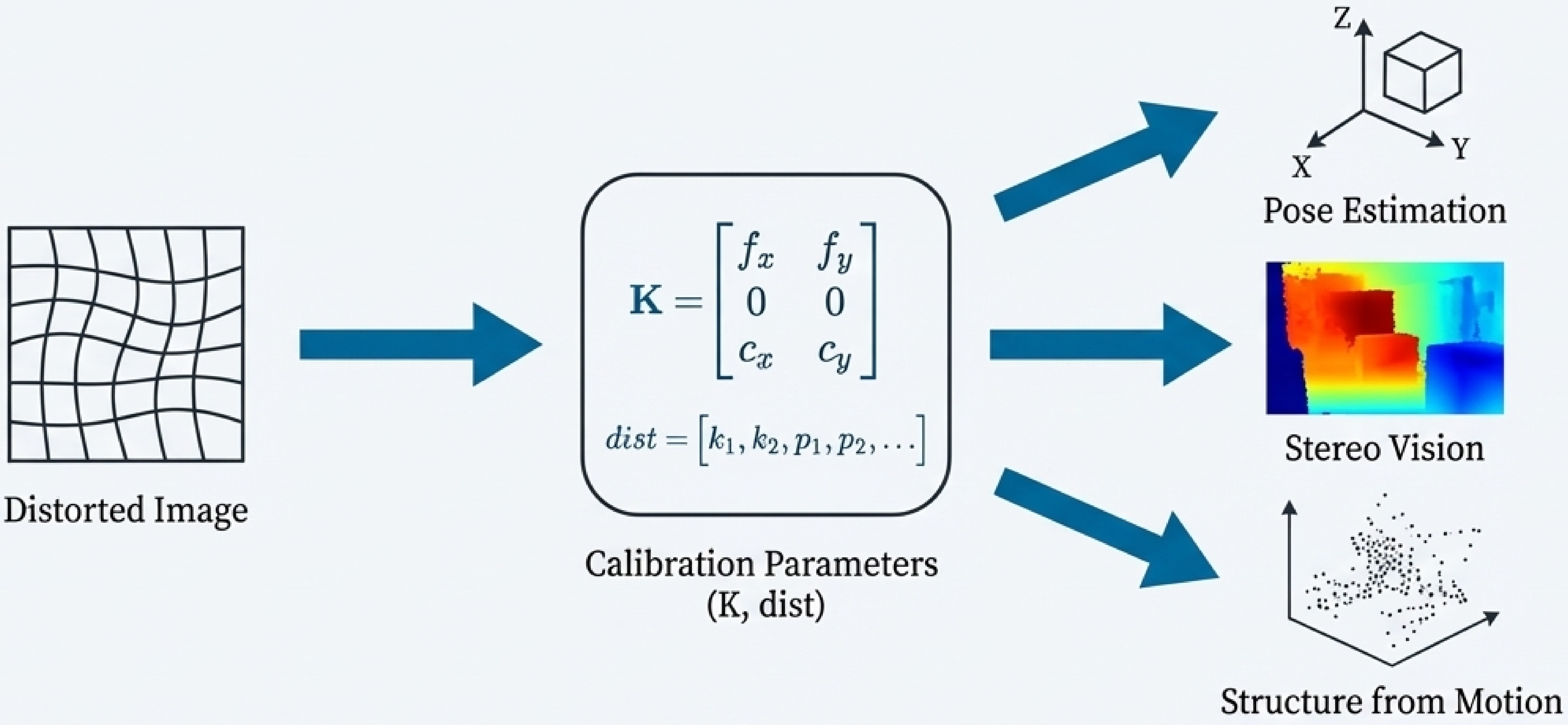
State-of-the-Art Tools & Libraries

- COLMAP: General-purpose Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipeline.
- OpenMVG: "Open Multiple View Geometry" library for the SFM community.
- Meshroom: A free, open-source 3D Reconstruction Software based on the AliceVision framework.

Standard Datasets

- Middlebury Stereo: For benchmarking stereo algorithms.
- ETH3D: High-resolution multi-view stereo datasets.

From Correction to Perception: The Power of a Calibrated System



The journey from a 2D image to 3D understanding begins with acknowledging and correcting the imperfections of our sensors. **Camera calibration is the foundational act of translation**—it turns unreliable pixel coordinates into precise, metric measurements of light rays in space. By mastering this single process, **we transform a simple camera from** a picture-taking device into a powerful instrument for measuring and reconstructing the three-dimensional world.