CS 188 Robotics Week 3

Aidan Jan

April 15, 2025

Model Predictive Control

[FILL 10] **Model predictive control (MPC)** is an optimal control technique in which the calculated control actions minimize a cost function for a constrained dynamical system over a finite, receding, horizon.

- Predicts future system behavior using a **model**
- Solves an **optimization** problem at each step.
- Applies only the first control input at each step (iterative).
- Repeats this process continuously (receding horizon).
- Handles input and output **constraints** directly.

System Identification

. . . building a mathematical model of a dynamic system **from measured data**
[FILL 13]

Model Predictive Contouring Control (MPCC++)

[FILL 14]

Cameras and 2D Perception

Color Camera

- Cameras are the primary sensor for many robotic platforms
- One of the cheapest and richest sensors is a camera
- Many other sensors are built on top of the color camera

Images Representation

- An image is basically a 2D array of intensity/color values
- Image types: [FILL 17, incl bottom text]

Grayscale Images

A grid (2D matrix) of intensity values: [FILL 18]

- Pixel: A "picture element" that contains the light intensity at some location (x, y) in the image Referred to as $I(x, y)$
- Image Resolution: expressed in terms of Width and Height of the image

Camera and Image Formation

- How we get the image (Image formation)
- Pinhole Camera Model
- 2D computer vision tasks and challenges

Image Formation

[FILL 20]

Pinhole Camera Model

Pinhole image: Natural phenomenon, known during classical period in China and Greece (e.g., 470 BCE to 390 BCE).

- Used for art creation and religious ceremony in the ancient times.
- Expensive to record the image (drawing)

[FILL 21, 22]

- Light sensitive material was used as film.
- Hard to store, loses color after a while

Today: photon sensors are CCD, CMOS, etc.

Human Vision

[FILL 24, im9]

Why do we need a pinhole?

[FILL 25-1] Light rays from many different parts of the scene strike the same point on the paper. [FILL 25-2] Each point on the image plane sees light from only one direction — the one that passes through the pinhole.

Problem with Pinhole Camera

The pinhole size:

- If large, blurry
- If small, not enough light
- When the pinhole size is extremely small, we will see the diffraction effect through the pinhole, resulting in the blurry image

Solution: refraction (lenses)

- Essentially add multiple pinhole images
- Shift them to align using the light **refraction**
- However, this alignment works only for one depth (need the object and image plane to stay in focus.)

[FILL 27]

Lenses Issues (depth of field)

Only objects on focus plane are in "perfect" focus [FILL 29, incl text.]

$$\frac{1}{D} + \frac{1}{D'} = \frac{1}{f}$$

where D is the distance of a focus plane to the lens plane, D' is the distance of the image plane to the lens plane, and f is the focal length of the lens.

- Objects close to the focus plane are in better focus
- Objects further away are not.

Camera Terminology

These terms will be defined below.

- Focal length
- Field of view
- Aperture
- Camera intrinsic
- Camera extrinsic

Pinhole Camera Geometry

Motivation

- Physics of real cameras are all different (too tedious to model all of them).
- But they all try their best to approximate pinhole camera.
- So **in most of computer vision subjects, we model all cameras mathematically as a pinhole camera.**

Field of View (FOV)

[FILL 34]

$$\alpha = 2 \arctan \frac{d}{2f}$$

- The unit of FoV α is a degree.
- Each camera has two FoV: vertical and horizontal.

Focal Length

[FILL 35, full]

Camera Projection

[FILL 37, 38 (combined)]

$$\frac{u}{f} = \frac{X}{Z}$$
$$\frac{v}{f} = \frac{Y}{Z}$$

In camera coordinates, the camera center is the origin.

$$p_{2d} = \begin{bmatrix} u \\ v \end{bmatrix} \qquad\qquad p_{3d} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

[FILL 40, all] [FILL 41, all]

Image Coordinate

[FILL 43] Until now, we use 2D coordinate conventions that are **consistent** with the 3D camera coordinate. However, if your application uses a different 2D coordiante, you'll need to further transform the (u, v).

For example, consider the following cases where we change the direction of the axes and the position of the origin. [FILL 44, 45]

Popular Camera Coordinate Systems

[FILL 46, full]

Camera Projection:

$$\lambda \begin{bmatrix} u \\ v \\ f \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

[FILL 47]

Computer Vision

A quick overview: what is computer vision? [FILL 49]

What Makes 2D Computer Vision Hard?

[FILL 50] Variation: same cat, different poses, view points, ... [FILL 51] More variation: different cats, different shapes, colors, textures, ...

Other factors:

- Illumination
- Occlusion: partial observation
- Ambiguity: some objects may look a lot like others; different perspectives may look like different objects

Human Vision

[FILL 57]

3D Vision

How do humans and animals perceive depth?

- Binocular vision: 2 eyes instead of 1
- Structure from Motion (SIM): walking around an object allows you to build a 3D model.

How do robots perceive depth?

- Stereo camera
- Time of flight
- Structured light

2D to 3D Projection

[FILL 63]

- Knowing just 2D coordinate (u, v), we don't have enough information to compute the 3D point location (X, Y, Z)
- However, with an additional depth channel we can. (RGB-D image).
  - The image on the right is an RGB-D image. Each pixel records the depth value Z (in meter or millimeter)

We can combine the two images to form a single, 3D image. [FILL 65]

- Depth image $\rightarrow$ 3D point clouds:
- A pixel with
  - image coordinate (u, v)
  - Depth value $= Z$
  - Focal length f
- Its 3D location (X, Y, Z) in camera coordinate can be computed by:

$$X = \frac{u}{f} \cdot Z \qquad\qquad Y = \frac{v}{f} \cdot Z$$

Summary:

[FILL 66, full]

World Coordinate to Camera Coordinate

- In order to apply the camera model we described so far, the 3D point (X, Y, Z) must be expressed in camera coordinates (i.e.; centered at the camera origin)
- However, the world coordinate can be different from the *camera coordinates*.
- Requires an additional transformation

[FILL (combine 68-70)]

Camera: Putting Everything Together

[FILL (combine 72-73)]

$$x = K[R|t]X$$

- Map a 3D point X into a 2D coordinate in image x
- How to describe its *pose* in the world? (extrinsic matrix)
- How to describe its internal parameters? (intrinsic matrix)

Camera: Calibration

Goal: estimate the camera parameters

- Version 1: solve for projection matrix

$$X = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = PX$$

- Version 2: solve for camera parameters separately
  - Intrinsic (focal length, principle point, pixel size)
  - Extrinsic (rotation angles, translation)

To calibrate:

1. Identify correspondance between image and scene
2. Compute mapping from scene to image

Requirement

1. Must know geometry very accurately
2. Must know correspondance

$$x_i = PX_i$$

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Robust camera calibration is still an open challenge!