EECS C106B / 206B Robotic Manipulation and Interaction **Spring 2020**

Lecture 19: (Geometry and Computer Vision Overview)

Scribes: Eric Liu, Derek Pan

19.1 Overview

This lecture provides an introduction to computer vision and geometry for structure from motion. In this lecture, we tackle the problem statement, example applications, and an introduction to the theory we'll be looking at in future lectures. The main challenge in computer vision is recovering the geometric structure and texture of an object from a variety of images.

19.2 Example Applications

19.2.1 Autonomous Highway Vehicles

Autonomous highway vehicles can use computer vision to do things like detect lane lines among other things. The image to the right is the reconstruction of the lanes, and small artifacts that weren't in the original image can be seen as small line segments to the side of the lanes.

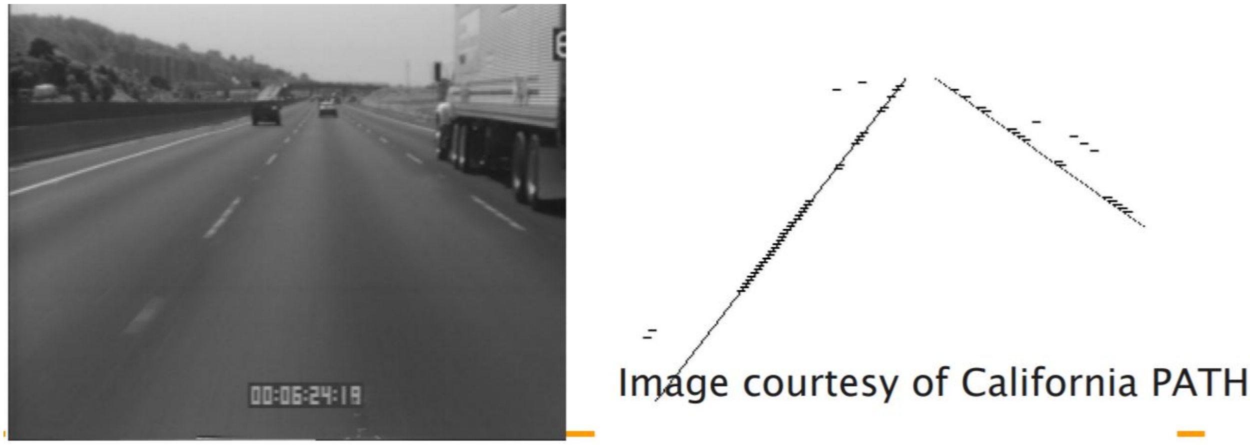


Image courtesy of California PATH

Figure 19.1: Highway lane lines from CV (from lecture slides)

19.2.2 UAVs

An example of this example given in the lecture was the autonomous landing of a helicopter onto a moving ship. Since a ship moves significantly in the waves, it can be difficult to consistently locate and land on the designated spot. One way that vision can help is to consistently find the designated spot and obtain its pose estimate. A controller can then be used to track the desired pose. Here, the heaving ship is simulated with a Stewart platform on a trailer, and an algorithm tracked so(3) for landing on the AR tag precursor.

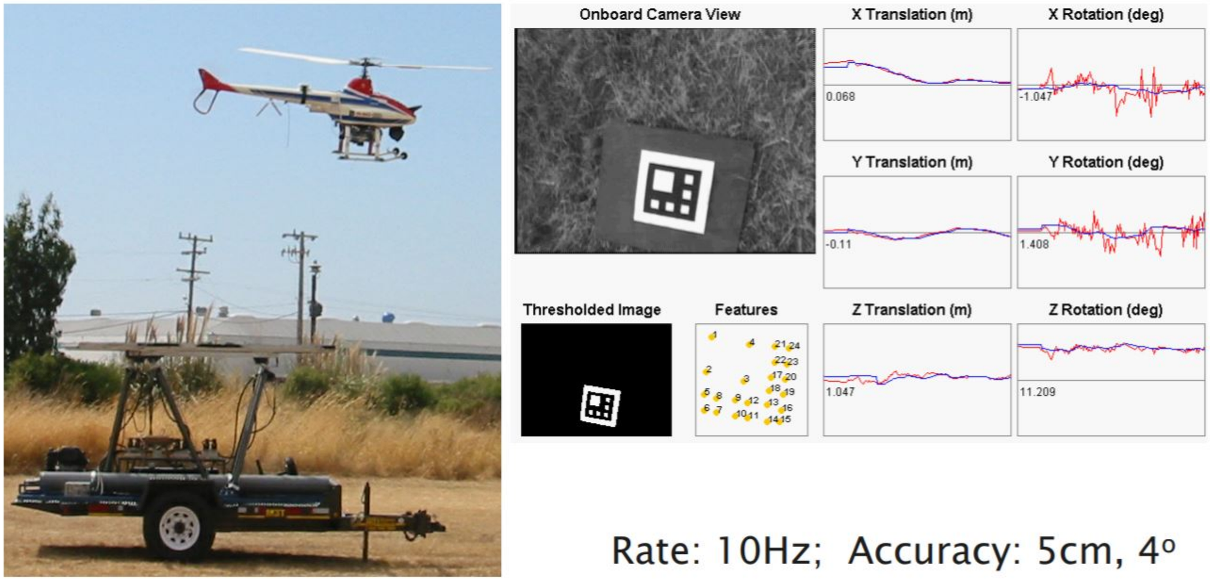


Figure 19.2: UAV landing example (from lecture slides)

19.2.3 Augmented Reality

A well known example of 3D vision is augmented reality. This is seen in well known applications like Pokemon Go and the NFL. In the NFL, for example, vision is used to superimpose first down lines on the field as well as ads.

19.2.4 Image based modeling and rendering

Another application is modeling and rendering from images. The question posed is how do you get from real life, extremely complicated models, to geometric models? There are issues with estimating what's not shown on the photo, dealing with background, noise, environment, etc.

19.2.5 Mosaicing

Mosaicing is the process of taking multiple photos and stitching them together into one. This process is used in applications like panoramic photos.

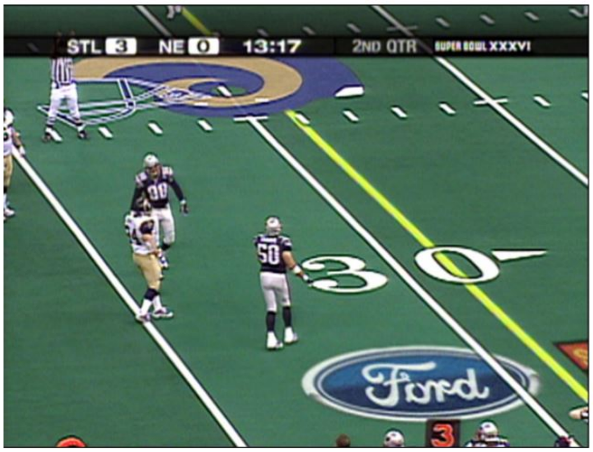


Figure 19.3: Football first down lines and ads example (from lecture slides)

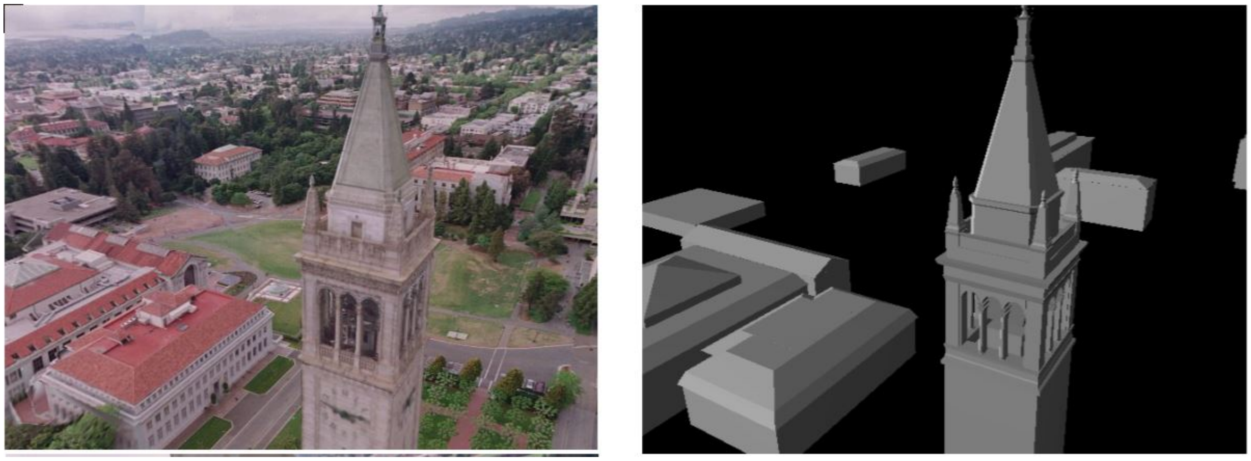


Figure 19.4: Imaging example (from lecture slides)

19.3 Computer Vision History

19.3.1 CV and art

The theory behind computer vision originated from painters who wanted to make realistic art. Painters needed to understand how light reflects off objects and falls incident to the eye. Euclid of Alexandria and Anaxagoras were perhaps the first to discuss this, and fresco artists of ancient Rome tried to incorporate it in their painting. Painting realistic shadows, shading, lighting, etc required a good understanding of structures and perspective. One example of a deep understanding of perspective geometry comes through the Scholar of Athens painting by Raphael done in 1518.

Here, Raphael shows his precise and rigorous calculation of the vanishing point. He understood that when there are multiple vanishing points, the eye will pick out the inconsistencies.

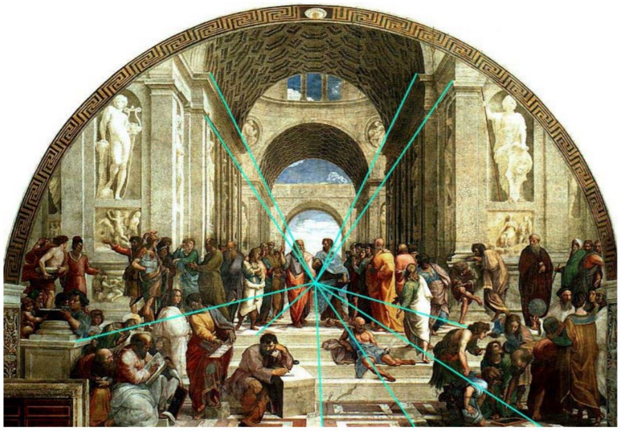


Figure 19.5: ”Scholar of Athens”, Raphael, 1518 (from lecture slides)

19.4 Camera Model

Also known as a Pinhole Camera Model, from the earliest pinhole cameras. In the figure below the bottom equation ($\mathbf{X}$) is known as the ”perspective equation,” and $[X; \ Y; \ Z]$ are the world coordinates, with Z the unknown depth, also designated λ elsewhere. The coordinates of $[x; \ y]$ can be thought of as ratios of the corresponding world coordinates and the depth Z.

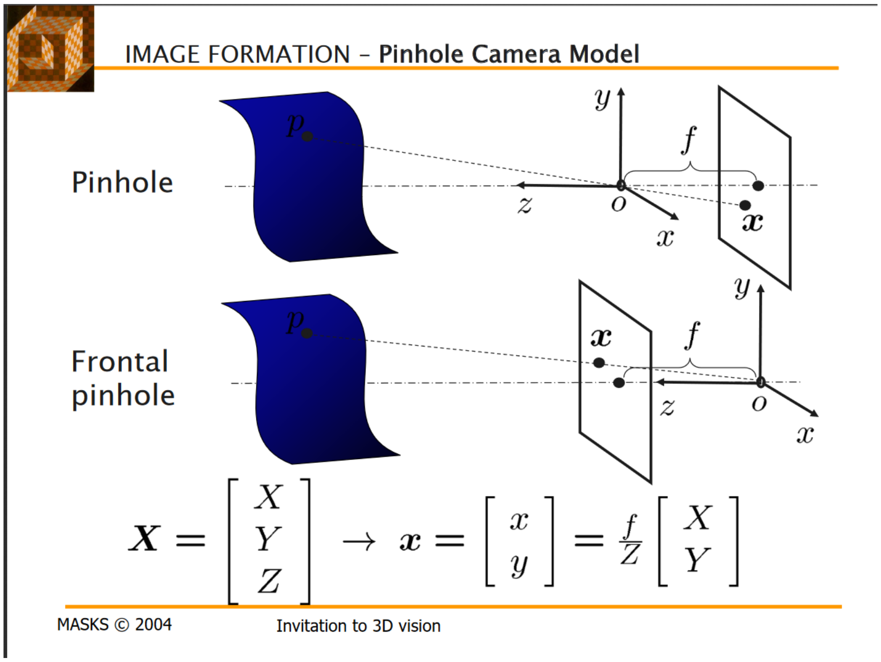


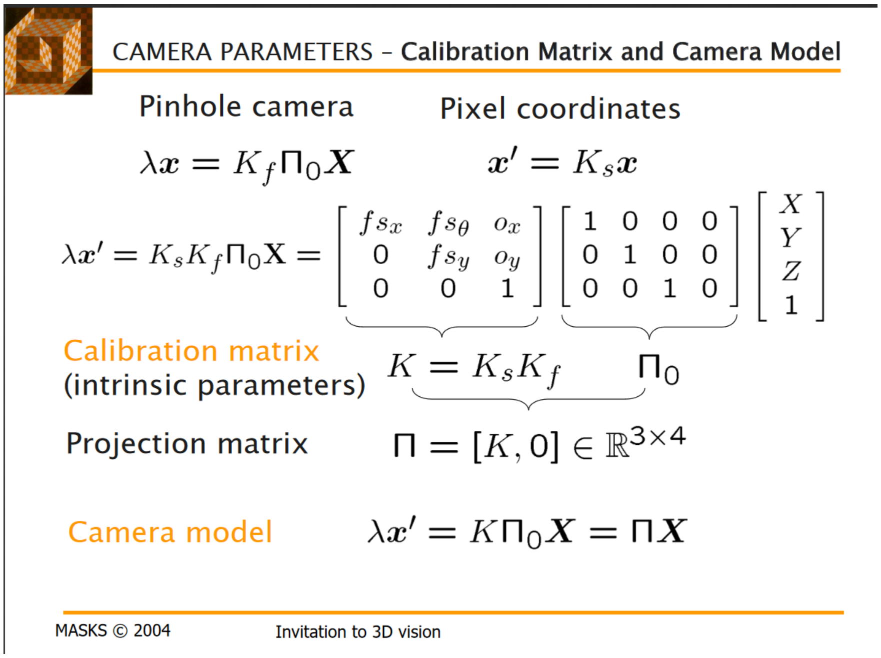
IMAGE FORMATION – **Pinhole Camera Model**

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \to \boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}$$

MASKS © 2004 Invitation to 3D vision

Figure 19.6: Pinhole camera model (from lecture slides)

The model is developed further in the slides and concludes with the following after incorporating camera calibration:



CAMERA PARAMETERS – **Calibration Matrix and Camera Model**

Pinhole camera Pixel coordinates

$\lambda \boldsymbol{x} = K_f \Pi_0 \boldsymbol{X}$ $\boldsymbol{x}' = K_s \boldsymbol{x}$

$$\lambda \boldsymbol{x}' = K_s K_f \Pi_0 \mathbf{X} = \underbrace{\begin{bmatrix} f s_x & f s_\theta & o_x \\ 0 & f s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}_{K = K_s K_f} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\Pi_0} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Calibration matrix (intrinsic parameters) $K \underbrace{= K_s K_f \qquad \Pi_0}$

Projection matrix $\Pi = [K, 0] \in \mathbb{R}^{3\times 4}$

Camera model $\lambda \boldsymbol{x}' = K \Pi_0 \boldsymbol{X} = \Pi \boldsymbol{X}$

MASKS © 2004 Invitation to 3D vision

Figure 19.7: Camera model and calibration matrix (from lecture slides)

where

$$\begin{cases} K_f & \text{camera parameter matrix (Kruppa)} \\ s_{x,y} & \text{relative sizes of pixels} \\ s_\theta & \text{pixel skew from rotation} \\ o_{x,y} & \text{origin of pixel grid} \\ f & \text{focal length} \end{cases}$$

19.5 Primitives and Correspondence

The human eye uses local attributes and global correspondences to identify a specific point of an object and be able to track it in spite of changes in perspective and illumination. Getting a computer to perform the same task is very difficult and an active area of research. The example below with the Egyptian queen illustrates the challenge. Illumination of the point below the queen's eye changes with the perspective, and global correspondence points are easily confused by the computer. This will be the topic of the next lectures.

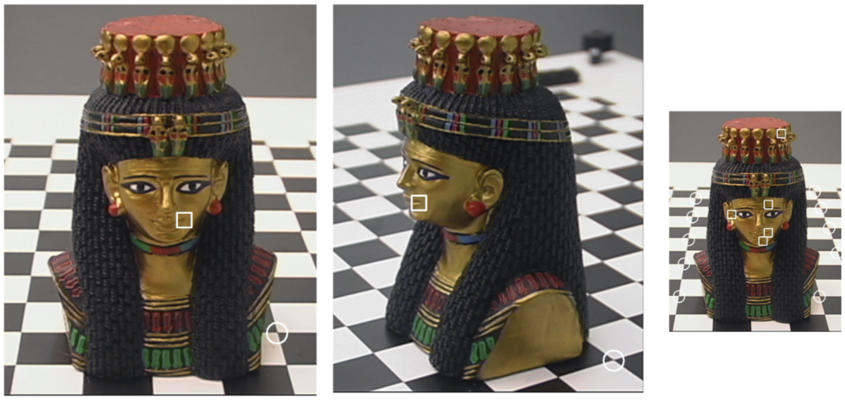


Figure 19.8: Difficulty in identifying a point on object after rotation (from lecture slides)

19.6 Additional Resources

Professor Yi Ma teaches "Geometry and Learning for 3D Vision" CS 294/167