Cognitive Computer Vision

Projective Geometry and 3D data

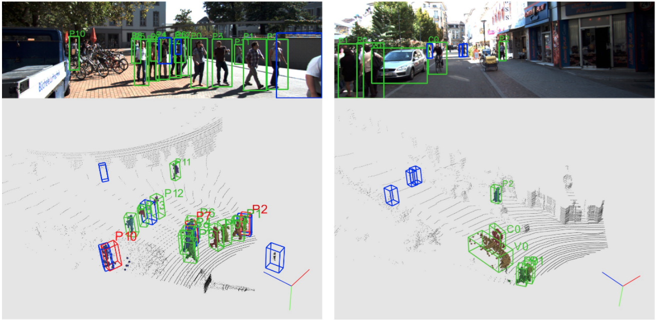
Dr. Mikko Lauri

Department of Informatics
Computer vision group

SS 2018

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Motivation



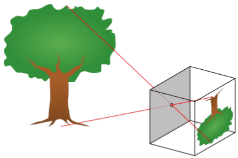
- 3D data is increasingly important in computer vision applications
- Understanding projective geometry and how images are formed is key to understanding 3D data.

Figure: Qi et al. [2018]

Contents

Contents

Pinhole camera



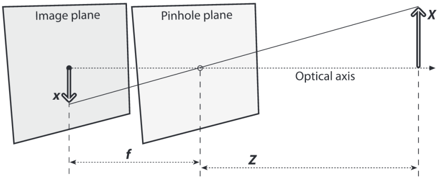
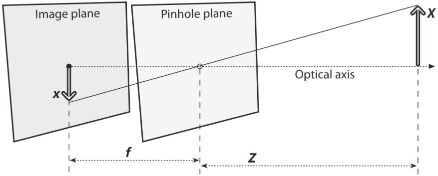


Figure (upper): https://en.wikipedia.org/wiki/Pinhole_camera_model
Figure (lower): Bradski and Kaehler [2008]

Pinhole camera



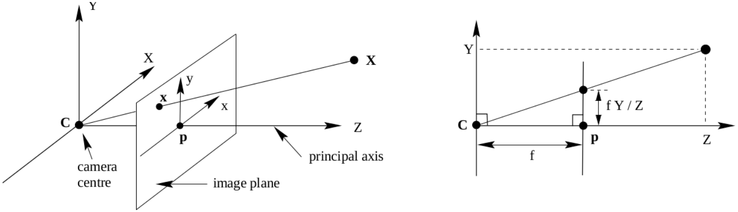
- Light rays projected from scene onto image plane via pinhole.

$$-\frac{x}{f} = \frac{X}{Z}, \quad -x = f\frac{X}{Z} \tag{1}$$

- f is called the focal length of the camera.

Figure: Bradski and Kaehler [2008]

Pinhole camera

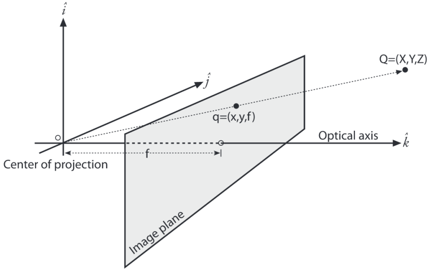


- Switching position of pinhole and image plane results in more convenient mathematical model.

$$x = f\frac{X}{Z} \quad y = f\frac{Y}{Z} \tag{2}$$

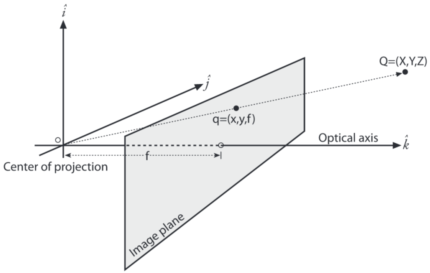
Figure: Wikipedia

Pinhole camera



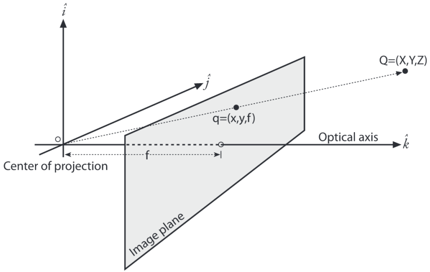
- Camera projects point Q onto the image plane at q.
- Point at the intersection of optical axis and image plane is the principal point.
- Principal point not always aligned to center of image plane: use c_x and c_y to model displacement from image plane origin.

Pinhole camera



- In reality, the focal length is product of true physical focal length F and size of an individual image element or *pixel*.
- Rectangular pixel with sides of length s_x and s_y: use $f_x = Fs_x$ and $f_y = Fs_y$

Pinhole camera: projection to image plane



Pinhole camera projection

A pinhole camera with parameters (f_x, f_y, c_x, c_y) projects a point $Q = (X, Y, Z)$ is onto the image plane as $q = (x, y)$, where

$$x = f_x \frac{X}{Z} + c_x, \quad y = f_y \frac{Y}{Z} + c_y.$$

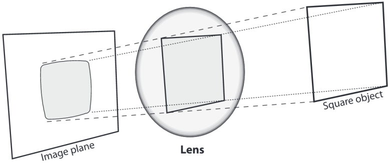
Contents

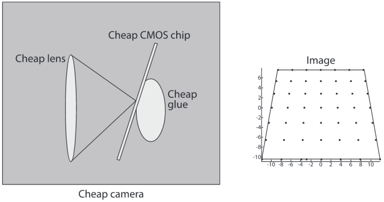
Why lenses?

- Only a small amount of light can enter through a pinhole over a fixed time: imaging is slow!
- Use lens to capture and focus more light to the image plane: increased speed
- The price to pay: lens distortion, as the perfect lens shape is difficult to achieve.

Two types of lens distortion

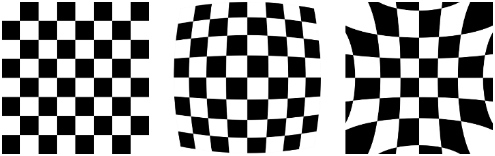
- Radial distortion
  - Due to lens shape (spherical instead of parabolic): rays farther from the center of the lens bent more than those near the center.
  - The barrel or fish-eye effect especially noticeable at edges of image
- Tangential distortion
  - Due to assembly process of camera: lens not fully parallel to image plane.



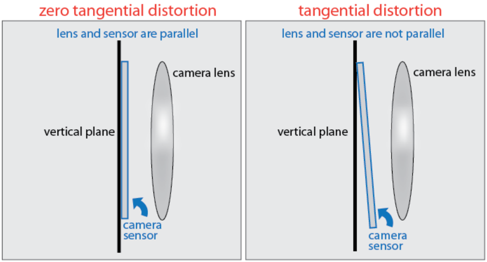


Figures: Bradski and Kaehler [2008]

Illustration of lens distortion



No distortion | Positive radial distortion (Barrel distortion) | Negative radial distortion (Pincushion distortion)



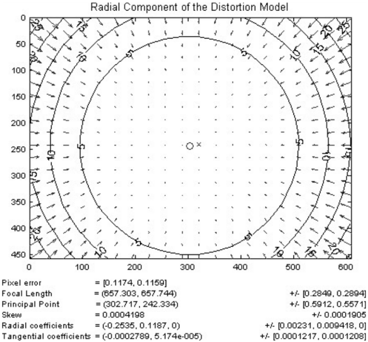
Correcting radial distortion

- Radial distortion corrected by changing pixel location (x, y) on image plane to (x_c, y_c):

$$
\begin{aligned}
x_c &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \\
y_c &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6),
\end{aligned}
\tag{3}
$$

  where k_i are the distortion correction parameters, and r is the pixel distance from center of image plane.

- Often only k_1, k_2 used, k_3 only for highly distorted cameras (e.g., fish-eye lens)



Figures: Bradski and Kaehler [2008]

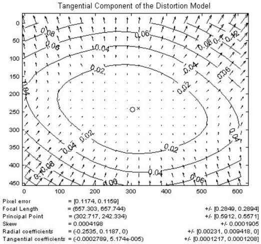
Correcting tangential distortion

- Tangential distortion corrected by changing pixel location (x, y) on image plane to (x_c, y_c):

$$\begin{aligned} x_c &= x + 2p_1 y + p_2(r^2 + 2x^2), \\ y_c &= y + p_1(r^2 + 2y^2) + 2p_2 x \end{aligned} \tag{4}$$

  where p_i are the distortion correction parameters, and r is the pixel distance from center of image plane.

- Points displaced elliptically as function of location and r.



Figures: Bradski and Kaehler [2008]

How to get the parameters?

- The camera intrinsics characterized by c_x, c_y, f_x, and f_y.
- Radial and tangential distortion characterized by k_1, k_2, k_3, p_1, and p_2.
- *Camera calibration* is applied to identify these parameters.
- In practice: Image a physical object with known properties from many views, write out the camera equations and solve for the camera instrinsics and distortion parameters.

Known object: The chessboard

- Corners of a chessboard form a regular pattern with known spacings.
- Extract chessboard corner locations from image.
- Intrinsic parameters tied to 2D/3D geometry of scene: corner locations provide information on them.
- Distortion parameters identified by examining how the pattern of the corners is distorted in the image plane.

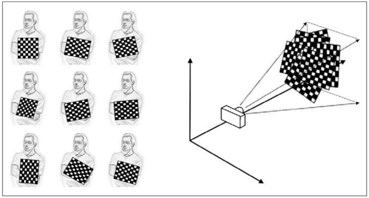


Figure 11-9. Images of a chessboard being held at various orientations (left) provide enough information to completely solve for the locations of those images in global coordinates (relative to the camera) and the camera intrinsics

---

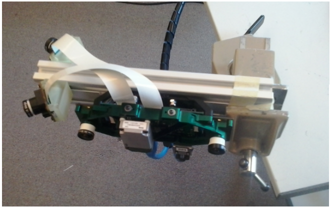
Figures: Bradski and Kaehler [2008]

Effect of calibration

- Undistorting an image: straight lines recovered

Advanced calibration

- Modern imaging systems becoming more complex
  - Multi-camera systems: need to know relative poses of cameras
  - Multiple sensor modalities: spatial and temporal calibration of cameras and inertial measurement units
- From checkerboard patterns to tag patterns: robust detection [Richardson et al., 2013]



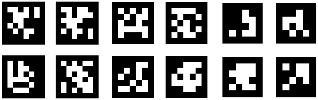
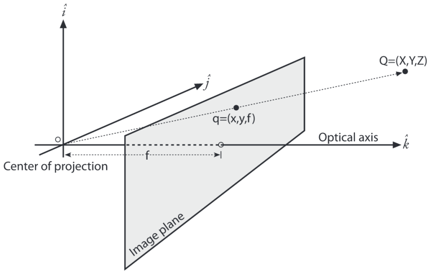


Figure: `https://github.com/ethz-asl/kalibr`, `https://april.eecs.umich.edu/wiki/AprilTags`

Contents

Depth imaging

- Color cameras record for each pixel the proportions of red (R), green (G), and blue (B) color.
- Depth cameras record for each pixel the distance of an object that reflects light along the vector from the centre of projection through that pixel on the image plane.



Some types of depth cameras

- Structured light: project a known pattern of light onto the scene. Record the pattern and recover depth by observing how pattern is distorted.
  - Example: ASUS XTion, Microsoft Kinect v1; infrared (IR) pattern and IR camera.
- Time-of-Flight: measure the time that it takes for emitted light to travel to an object and back to the sensor.
  - Example: Microsoft Kinect v2, laser range finders / Light Detection and Ranging (LIDAR) devices
- Stereo camera: observe the scene with two cameras with precisely known displacement; similar to human depth vision

Kinect v1

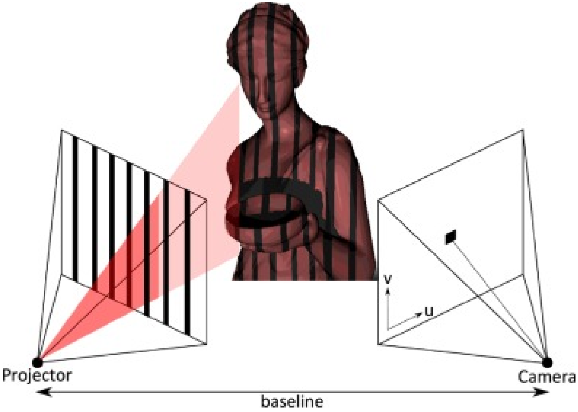
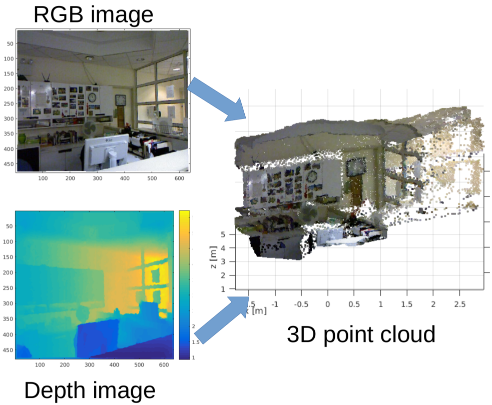


Image from Sarbolandi et al. [2015]

RGB-D images

- Typically, the depth camera is operated together with a regular RGB camera.
- If displacement between the cameras, and the intrinsic parameters of the cameras are known, the images can be mathematically *registered* to be effectively on the same image plane.
- In registered images, a pixel q in the RGB image has distance from the centre of projection denoted by pixel q of the depth image.

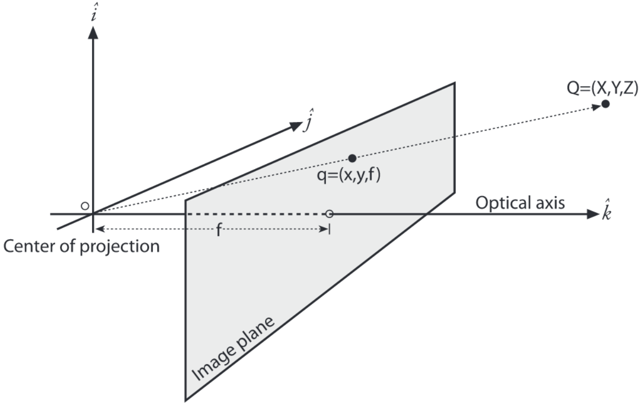
Point clouds



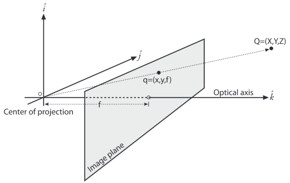
- Point clouds provide a true three-dimensional representation of the scene.

Recall: projective geometry

- Projection of a real-world point (X, Y, Z) onto the image plane loses information of the distance of objects: $x = f_x(X/Z) + c_x$, $y = f_y(Y/Z) + c_y$.



Obtaining a point cloud



- Given: depth image, parameters f_x, f_y, c_x, c_y.

Pinhole camera projection: Reversed

Reconstruct the real-world 3D point (X, Y, Z) for any pixel $q = (x, y)$ with corresponding depth $Z(q)$:

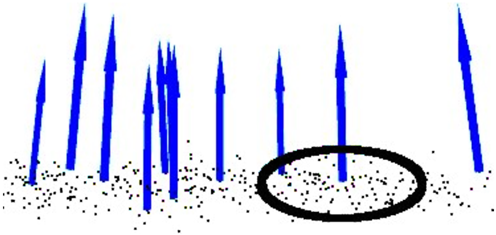
$$X = \frac{x - c_x}{f_x} Z(q) \quad Y = \frac{y - c_y}{f_y} Z(q) \quad Z = Z(q)$$

Point clouds

- Generally, a point cloud P is just a finite subset of any metric space $\mathcal{M}$.
- Example: geometry of a scene can be represented by a three-dimensional point cloud

$$P := \left\{x_i \in \mathbb{R}^3 \mid i = 1, \ldots, N\right\} \subset \mathbb{R}^3$$

- Example: $P \subset \mathbb{R}^6$ can additionally have R, G, B, information.
- Example: $P \subset \mathbb{R}^9$ additional information of the direction (n_x, n_y, n_z) of the normal vector



Point clouds: examples

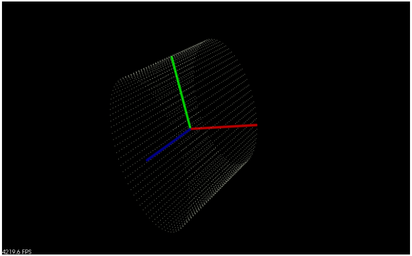


Figure: Left: XYZ only. Right: XYZ + RGB

Point clouds: Applications

- Segmentation, object recogntion, feature extraction, ...
- Reconstruction of 3D models of objects and/or scenes

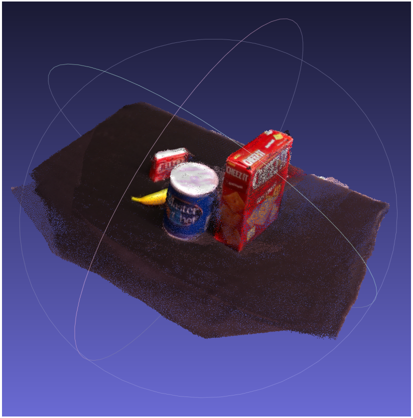


Figure: Reconstruction of objects on table from multiple RGB-D images

Example: reconstruction and labeling

- RGBD camera attached to robot arm
- 3D reconstruction from images, alignment of 3D Object model[1] to the scene
- Backprojection of information to individual frames for image labeling[2]
- Use cases: training of neural networks, pose estimation
- Video

[1] YCB Object set: http://ycbbenchmarks.com/
[2] With LabelFusion tool http://labelfusion.csail.mit.edu/

Contents

Conclusion

- Camera projects points from the real, three-dimensional world, onto the two-dimensional image plane.
- A camera model such as the pinhole camera leads to projective camera geometry that mathematically models the imaging process.
- Camera calibration is applied to reduce the effect of errors on the imaging process
- Access to depth images allows to reconstruct the 3D geometry by reversing the projective geometry
- 3D data has many applications in computer vision: segmentation, object detection, reconstruction, pose estimation, . . .

Bibliography I

Gary Rost Bradski and Adrian Kaehler. *Learning OpenCV*. O'Reilly Media, Inc., first edition, 2008. ISBN 9780596516130.

Charles Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data. In *CVPR*, 2018.

Andrew Richardson, Johannes Strom, and Edwin Olson. AprilCal: Assisted and repeatable camera calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013.

Hamed Sarbolandi, Damien Lefloch, and Andreas Kolb. Kinect range sensing: Structured-light versus Time-of-Flight Kinect. *Computer Vision and Image Understanding*, 139:1–20, 2015.