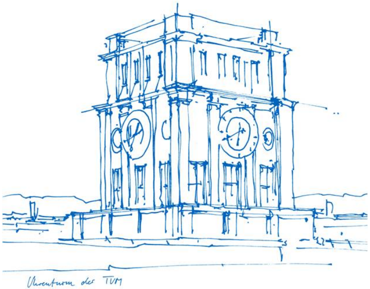
Computer Vision II: Multiple View Geometry (IN2228)

Chapter 09 Single-view Geometry

Dr. Haoang Li

28 June 2023 12:00-13:30



Announcements before Class

➢ Email/Post Reply

- ✓ Recently, some students asked questions about our course content via Moodle or email. I may fail to reply to all the questions in time due to a submission deadline of an academic work.
- ✓ I will reply to all your questions by weekend.

Announcements before Class

➢ Document for Knowledge Review

✓ This week, we will finalize the core part of our course.
✓ I will upload a new document for knowledge review.

Thu 01.06.2023 Chapter 06: 2D-2D Geometry (Part 1)

Wed 07.06.2023 Chapter 06: 2D-2D Geometry (Part 2)
Thu 08.06.2023 No lecture (Public Holiday)

Wed 14.06.2023 Chapter 06: 2D-2D Geometry (Part 3)
Thu 15.06.2023 Chapter 06: 2D-2D Geometry (Part 4)

Wed 21.06.2023 Chapter 07: 3D-2D Geometry
Thu 22.06.2023 Chapter 08: 3D-3D Geometry

Wed 28.06.2023 Chapter 09: Single-view Geometry
Thu 29.06.2023 Chapter 10: Combination of Different Configurations

Core part

Announcements before Class

➢ Reminder of Exam Registration

✓ Summer Semester Exam

- Our exam will tentatively take place on 04 August from 8:00 am to 10:00 am.
- Registration for our exam is possible between 22 May and 30 June.
- Deadline for grading of exams: 06 September 2023.

✓ Winter Semester Exam (Repeat Exam)

- You can skip the Summer Semester Exam and directly register for the Winter Semester Exam.
- We will provide any update in time.

Outline

- Overview
- Background Knowledge
- Vanishing Point Expression
- Vanishing Point Estimation
- Application: Camera Calibration

Overview

➢ Extract Geometric information from a single image

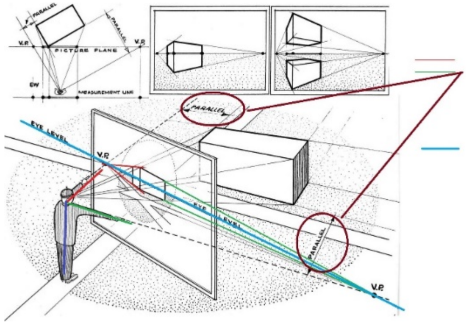
✓ General requirements of an algorithm for single-view geometry

- No longer needs point correspondences
- Can recover camera pose and/or 3D structure

✓ This sounds very difficult. We need some other clues:

- Structural regularity such as orthogonality and parallelism
- "Structured" lines and planes
- High-level geometric features such as vanishing point

…



Overview

➢ Some Applications of Single-view Geometry

✓ Single-view 3D reconstruction

Combining vanishing points and 3D box priors

Extracted vanishing points and contour primitives from a single image

Reconstructed 3D structure

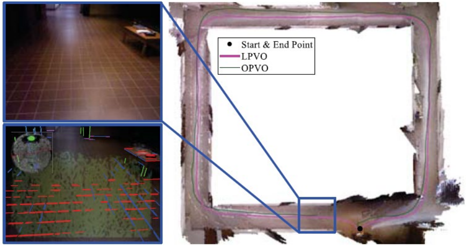
Y. Li, J. Mao, B. Freeman, J. Tenenbaum, N. Snavely, and J. Wu, “Multiplane program induction with 3D box priors,” in Neural Inf. Process. Syst., 2020.

Overview

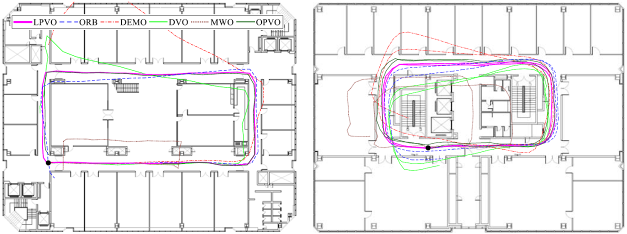
➢ Some Applications of Single-view Geometry

✓ Camera pose estimation and optimization (details will be introduced tomorrow)

Vanishing point encodes the rotation information of camera



Consistent map



Reducing the drift error of trajectory

Pyojin Kim, Brian Coltin, H. Jin Kim, ”Low-Drift Visual Odometry in Structured Environments by Decoupling Rotational and Translational Motion”, ICRA, 2018

Overview

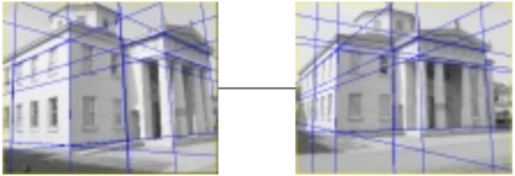
➢ Some Applications of Single-view Geometry

✓ Camera calibration (details will introduced later)

Estimate intrinsic parameters

Original uncalibrated photographs

Finding vanishing points and camera calibration

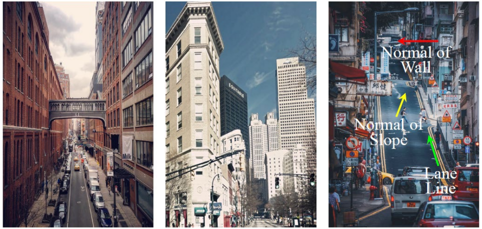


R. Cipolla, T. Drummond and D. Robertson, “Camera calibration from vanishing points in images of architectural scenes”, BMVC, 1999

Background Knowledge

➢ Representative Cities

✓ Man-made environments typically exhibit structural regularity



(a) Manhattan (b) Atlanta (c) Hong Kong

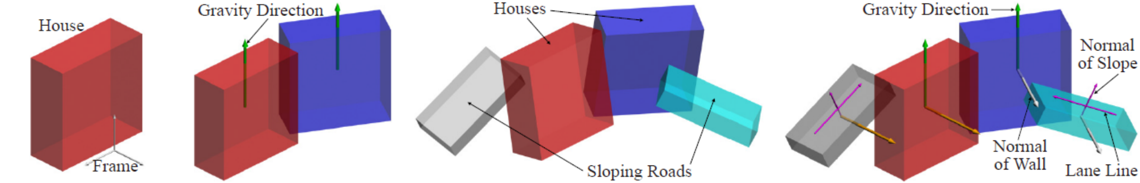
Representative cities exhibiting various structural regularities.

(a) Manhattan with a vertical dominant direction (DD) and two horizontal DDs.
(b) Atlanta with a vertical DD and multiple horizontal DDs.
(c) Hong Kong with a vertical DD, multiple horizontal DDs (see red arrow), and multiple sloping DDs (see yellow and green arrows). Red, yellow, and green arrows are mutually orthogonal.

Background Knowledge

➢ Common Structural Models

✓ Man-made environments can be abstracted by various structural models.



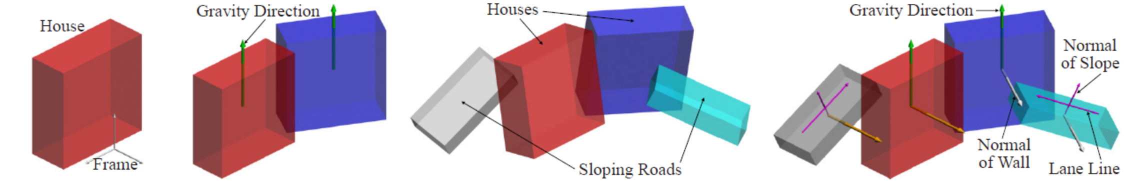
(a) Manhattan World (b) Atlanta World (c) A Mixture of Manhattan Worlds (d) Hong Kong World

[1] J. M. Coughlan and A. L. Yuille, “Manhattan world: Compass direction from a single image by Bayesian inference,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), vol. 2, 1999, pp. 941–947.
[2] G. Schindler and F. Dellaert, “Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), vol. 1, 2004, pp. 203–209.
[3] J. Straub, O. Freifeld, G. Rosman, J. J. Leonard, and J. W. Fisher, “The Manhattan frame model—Manhattan world inference in the space of surface normals,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 1, pp. 235–249, 2017.
[4] H. Li et al., "Hong Kong World: Leveraging Structural Regularity for Line-based SLAM," IEEE Transactions on Pattern Analysis and Machine Intelligence, Early access, 2023.

Background Knowledge

➢ Common Structural Models

✓ Man-made environments can be abstracted by various structural models.



(a) Manhattan World (b) Atlanta World (c) A Mixture of Manhattan Worlds (d) Hong Kong World

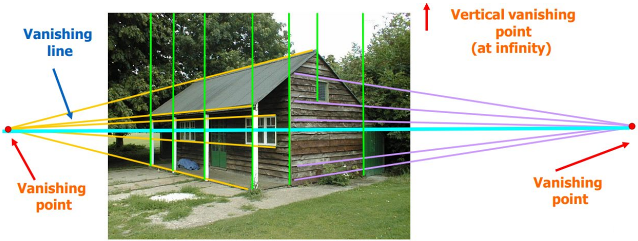
(a) Manhattan world [1] corresponds to a single block or frame. (In our class, we also consider this model)
(b) Atlanta world [2] corresponds to multiple blocks sharing a common vertical DD, e.g., gravity direction.
(c) A mixture of independent Manhattan worlds [3] corresponds to multiple unaligned and unrelated blocks.
(d) In Hong Kong world [4], {red, blue} blocks share a common vertical DD, e.g., gravity direction. {Blue, cyan} or {red, gray} blocks share a common horizontal DD, e.g., a normal of wall.

Background Knowledge

➢ Recap on Vanishing Points in 2D

✓ Definition and properties

- 2D lines projected from parallel 3D lines intersect at a "vanishing point" in the image.
- Vanishing points can lie both inside or outside the image.
- The connection between two horizontal vanishing points is the horizon.

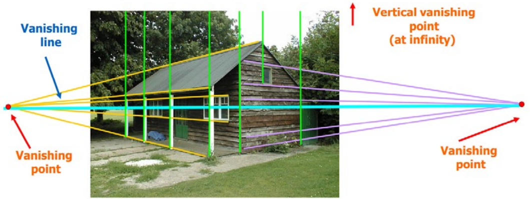


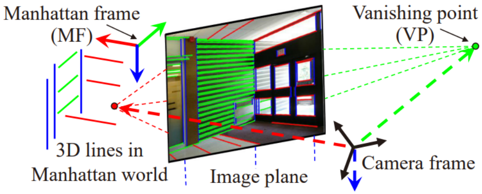
Background Knowledge

➢ Recap on Vanishing Directions in 3D

✓ Definition and properties

- A vanishing direction is defined by the connection between a vanishing point and camera center.
- **Vanishing direction** is parallel to a 3D **dominant direction**. We thus do not differentiate between them.



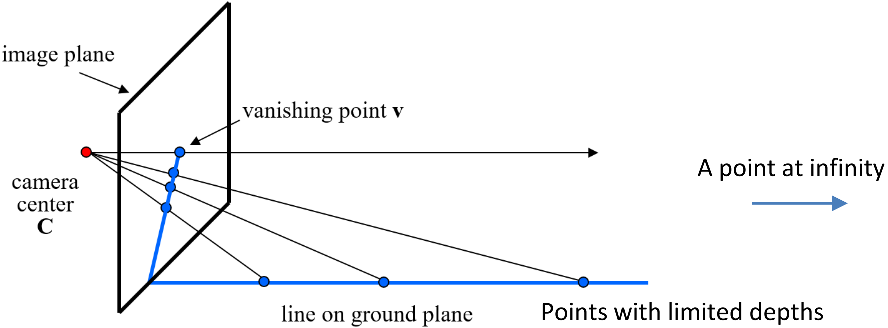


Vanishing Point Expression

➢ Intuitive Illustration

✓ Vanishing point **v** can be treated as a projection of a point at infinity. (Mathematical explanation will be given later)

✓ The ray from the camera center **C** through **v** is parallel to the 3D line.



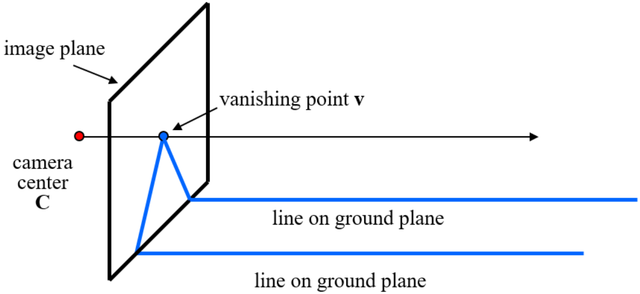
Vanishing Point Expression

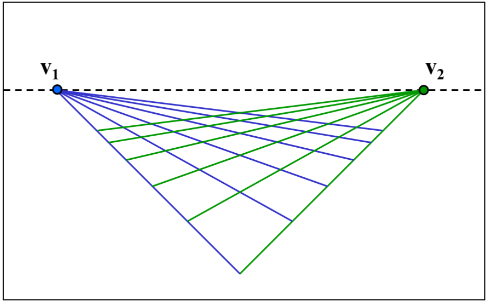
- Intuitive Illustration

✓ Lines and vanishing points

Any two parallel 3D lines have the same vanishing point v (see the left image)

An image may have more than one vanishing point (see the right image)

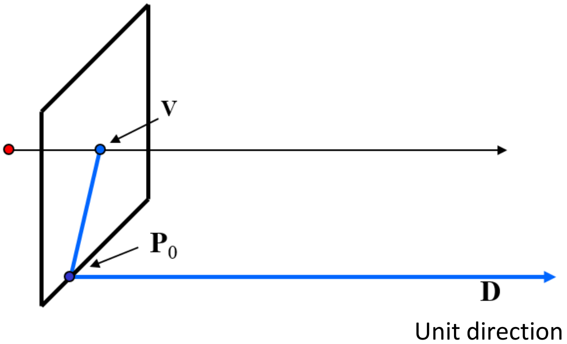




Vanishing Point Expression

➢ Mathematical Representation

✓ Expression of a 3D point with limited depth



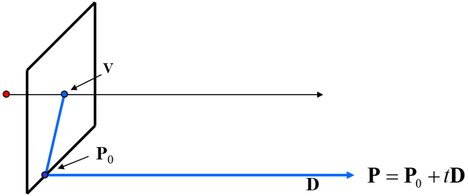
$$\mathbf{P} = \mathbf{P}_0 + t\mathbf{D}$$

$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix}$$

Homogeneous coordinates

Vanishing Point Expression

➢ Mathematical Representation



✓ Expression of a 3D point at infinity

- $\mathbf{P}_\infty$ denotes a point at *infinity*, and **v** is its projection.
- They depend only on line *direction*

$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} P_X/t + D_X \\ P_Y/t + D_Y \\ P_Z/t + D_Z \\ 1/t \end{bmatrix}$$

Homogeneous coordinates

If $t \to \infty$ ⇨ $\mathbf{P}_\infty \cong \begin{bmatrix} D_X \\ D_Y \\ D_Z \\ 0 \end{bmatrix}$

A special scalar

Considering non-homogeneous coordinates ⇨

Up-to-scale

$\mathbf{v} = \mathbf{K}\mathbf{P}_\infty = \mathbf{K}\mathbf{D}$

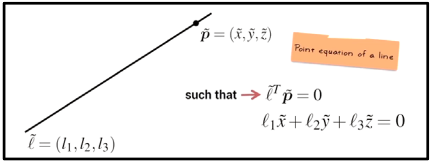
Perspective projection

$\mathbf{K}^{-1}\mathbf{v} = \mathbf{P}_\infty = \mathbf{D}$

Image normalization

Vanishing Point Expression

➢ Mathematical Representation



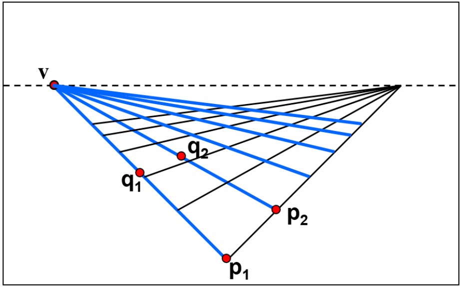
✓ Computation based on 2D lines

Intersect p_1q_1 with p_2q_2

Homogeneous coordinates of **2D intersection point**

$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

Homogeneous coordinates of **2D line**



✓ Least-squares version: Better to use more than two lines and compute the "closest" point of real intersection.

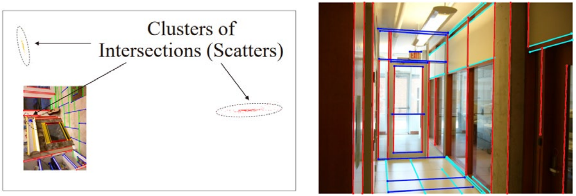
Vanishing Point Expression

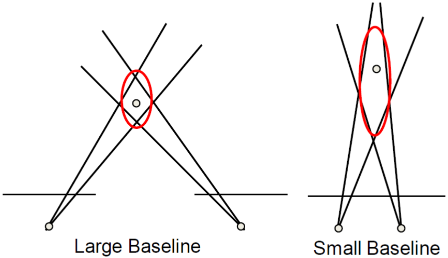
➤ Representation on Sphere

✓ Motivation

The intersection (vanishing point) may be far from the image center since image lines may be roughly parallel.

This is analogous to the case of short baseline in 3D reconstruction (two nearly parallel rays intersect at infinity).





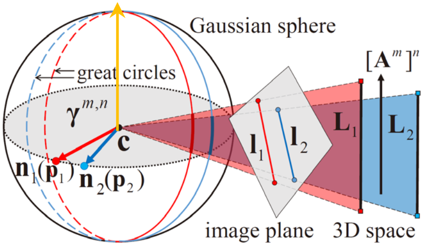
Vanishing Point Expression

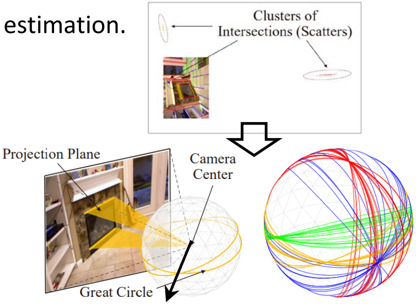
➢ Representation on Sphere

✓ Mapping image lines into great circles on sphere

- A projection plane intersect with sphere, forming a great circle.
- A set of great circles intersect at the same point. This point and sphere center (camera center) define the **vanishing direction**.
- We can reformulate vanishing point estimation as vanishing direction estimation.

Vanishing direction





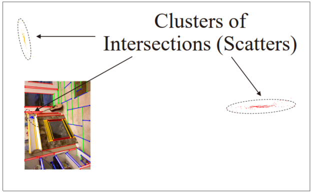
Vanishing direction

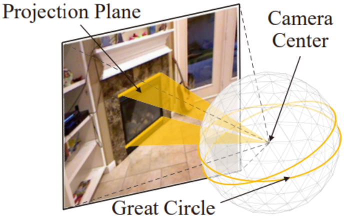
Vanishing Point Expression

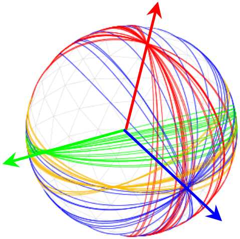
➢ Representation on Sphere

✓ Comparison with image representation and sphere representation

- Image is an unbounded space, while unit sphere is a bounded space.
- Image can hardly encode the orthogonality of vanishing directions. Unit sphere encodes the orthogonality constraint in 3D.







Vanishing directions

Vanishing Point Expression

➢ Transformation of Vanishing Direction (used later)

✓ 3D transformation between Manhattan and camera frames

- Manhattan frame's axes are aligned to the dominant directions

$$\mathbf{d}_1^{\mathcal{M}} = [1, 0, 0]^\top, \mathbf{d}_2^{\mathcal{M}} = [0, 1, 0]^\top \quad \mathbf{d}_3^{\mathcal{M}} = [0, 0, 1]^\top$$

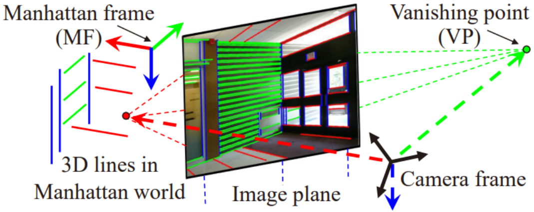
Known

- Rotation from Manhattan frame to camera frame

$$\mathbf{M}[\mathbf{d}_1^{\mathcal{M}}, \mathbf{d}_2^{\mathcal{M}}, \mathbf{d}_3^{\mathcal{M}}] = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3]$$

$$\mathbf{M} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3]$$

Unknown

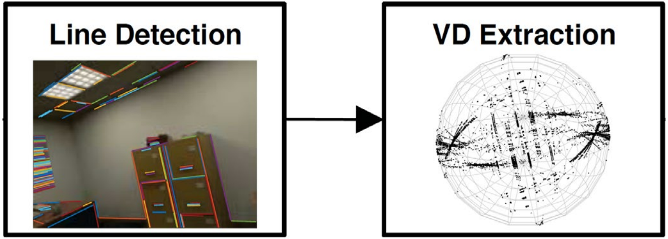


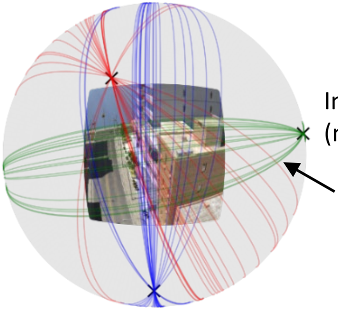
Vanishing Point Estimation

➢ Three Dominant Strategies

✓ 1. Census-based methods (old-fashioned)

- Let us consider estimation in 3D for example, i.e., compute vanishing directions instead of vanishing points.
- Due to noise, intersection of a pair of inlier circles slightly deviate from the ground truth position.
- A small area associated with a **high density of noisy intersections** corresponds to a vanishing point.
- Problem: sensitive to outliers.



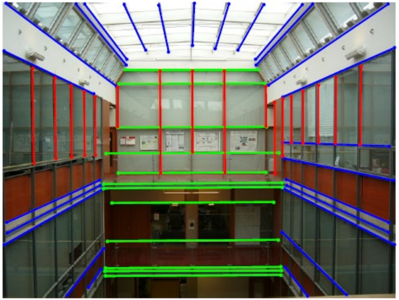


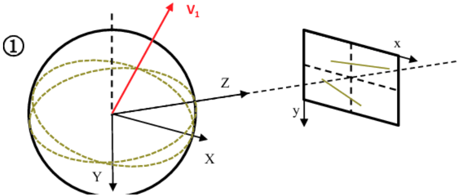
Vanishing Point Estimation

➢ Three Dominant Strategies

✓ 2. Sampling-based method (efficient)

- Let us consider estimation in 3D for example.
- Sample three image lines to compute three great circles
- Assume that the first two lines are associated with the same vanishing point, e.g., v_1; the third line is associated with another vanishing point (e.g., v2)
- We have to perform sampling several times to guarantee at least one sampling is valid.



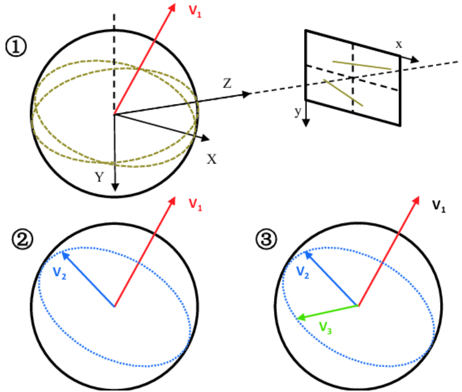


Vanishing Point Estimation

➢ Three Dominant Strategies

✓ 2. Sampling-based method (efficient)

- Use two circles to determine the first vanishing direction by computing the intersection.
- The second vanishing direction has two constraints: 1) lies within the third circle (reason: this direction is aligned to a 3D line lying within this circle); 2) orthogonal to the first dominant direction.
- The third direction is orthogonal to both first and second directions.



✓ 3. Search-based method (accurate)

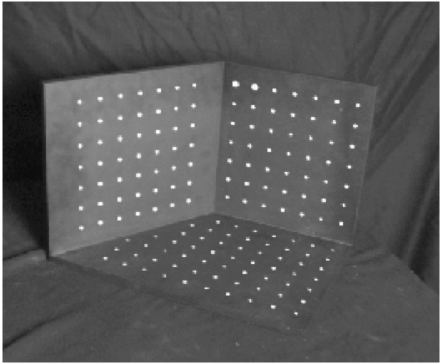
- Typically use branch and bound. It is relatively difficult to understand, and will not be introduced in our class.

Application: Camera Calibration

➢ Motivation

✓ Recap on correspondence-based calibration
- Place a known object in the scene
- identify correspondence between image and scene
- compute mapping from scene to image

✓ Issues
- must know geometry very accurately
- must know 3D-2D correspondences

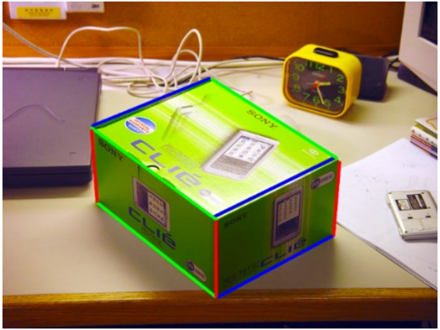


Application: Camera Calibration

➢ Calibration Based on Vanishing Points

✓ Advantages
- No need to establish point correspondences.
- Only a single image is enough.

✓ Setup: Let's assume that the camera is reasonable
- Square pixels
- Image plane parallel to sensor plane
- Principle point in the center of the image
- We only estimate the focal length.



Application: Camera Calibration

➢ Preliminary

✓ Perspective projection

- Recap on intrinsic matrix and projection matrix

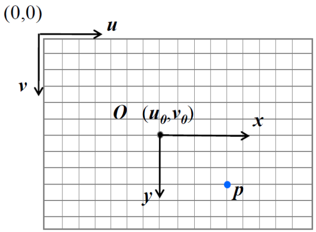


Image plane

$$u = u_0 + \frac{\alpha_u X_C}{Z_C}$$

$$v = v_0 + \frac{\alpha_v Y_C}{Z_C}$$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Intrinsic/Calibration matrix

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \left[R \mid T \right] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Projection Matrix (M)

Application: Camera Calibration

➢ Preliminary

✓ Perspective projection

- **Simplification**: We assume that principle point in the center of the image, and thus only focus on focal length.
- We define a **new image coordinate system** whose origin is located at (u_0, v_0).
- Accordingly, we define a new projection matrix

$$\begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} (R \mid t) \cong \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31}/f & r_{32}/f & r_{33}/f & t_3/f \end{bmatrix}$$

Projection result is up-to-scale, so we divide the projection matrix by *f*.

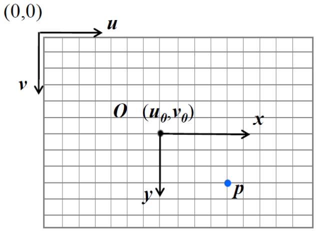
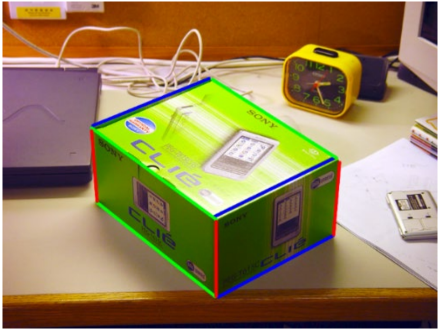


Image plane

Application: Camera Calibration

- Procedures

✓ Step 1: Calculate the vanishing points.

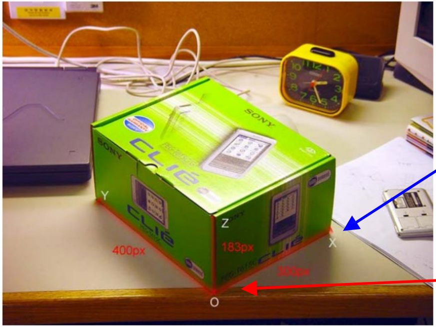


Marking the parallel lines in x (blue), y (green), z(red) directions

Application: Camera Calibration

➢ Procedures

✓ Step 2: Define **3D points at infinity** along x, y, and z axes, as well as origin in Manhattan frame.



3D Point at infinity along x axis:

$$\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$$

Infinity

2D projection (vanishing point has been **computed**)

3D Origin of Manhattan frame:

$$\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T$$

Finite

2D projection o can be **measured**

Application: Camera Calibration

➢ Procedures

✓ Step 3: Calculate the projection matrix (up-to-scale)

The projection matrix maps the 3D co-ordinates onto the 2D plane.

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

Homogeneous coordinates of 2D point

Projection matrix from the **Manhattan** frame to the image

Homogeneous coordinates of 3D point **in Manhattan frame (not world frame)**

$$\mathbf{\Pi} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} = \begin{bmatrix} \boldsymbol{\pi}_1 & \boldsymbol{\pi}_2 & \boldsymbol{\pi}_3 & \boldsymbol{\pi}_4 \end{bmatrix}$$

$\boldsymbol{\pi}_1$ $\boldsymbol{\pi}_2$ $\boldsymbol{\pi}_3$ $\boldsymbol{\pi}_4$

We aim to estimate each column of projection matrix

Application: Camera Calibration

➢ Procedures

✓ Step 3: Calculate the projection matrix (up-to-scale)

Estimated columns

Definition of projection

$$\boldsymbol{\pi}_1 = \boldsymbol{\Pi}\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T = \mathbf{v}_X$$

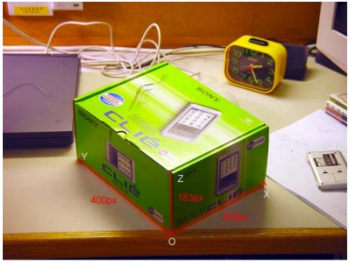
Known vanishing point

Unknown first column of $\boldsymbol{\Pi}$

3D point along x-axis at infinity in the Manhattan frame

Similarly, we have $\boldsymbol{\pi}_2 = \mathbf{v}_Y,\ \boldsymbol{\pi}_3 = \mathbf{v}_Z$

Application: Camera Calibration



- Procedures

✓ Step 3: Calculate the projection matrix (up-to-scale)

Estimated columns

$$\boldsymbol{\pi}_4 = \boldsymbol{\Pi}\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T$$ = Measured origin projection

Unknown fourth column of $\boldsymbol{\Pi}$

3D origin of the Manhattan frame

Estimated projection matrix: $\boldsymbol{\Pi} = \begin{bmatrix}\mathbf{v}_X & \mathbf{v}_Y & \mathbf{v}_Z & \mathbf{o}\end{bmatrix}$ Up-to-scale

Application: Camera Calibration

➢ Procedures

✓ Step 4: determine the scale and compute focal length

We only know vanishing point **v** and projection of origin **o** up to scale (for 2D points in the homogeneous coordinates, the last element is 1). Thus, we introduced unknown {a,b,c,d}.

$$\mathbf{\Pi} = \begin{bmatrix} a\,\mathbf{v}_X & b\mathbf{v}_Y & c\mathbf{v}_Z & d\mathbf{o} \end{bmatrix}$$

Simplified projection matrix defined before

$$\begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} (R \mid t) \simeq \begin{bmatrix} r_{11} & r_{12} & r_{13} & t'_1 \\ r_{21} & r_{22} & r_{23} & t'_2 \\ r_{31}/f & r_{32}/f & r_{33}/f & t'_3/f \end{bmatrix}$$

Application: Camera Calibration

$$\Pi = [a\mathbf{v}_X \quad b\mathbf{v}_Y \quad c\mathbf{v}_Z \quad d\mathbf{o}]$$

$$\begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}(R|t) \cong \begin{bmatrix} r_{11} & r_{12} & r_{13} & t'_1 \\ r_{21} & r_{22} & r_{23} & t'_2 \\ r_{31}/f & r_{32}/f & r_{33}/f & t'_3/f \end{bmatrix}$$

- Procedures

✓ Step 4: determine the scale and compute focal length

$$\begin{bmatrix} v_{x1} & v_{y1} & v_{z1} & o_1 \\ v_{x2} & v_{y2} & v_{z2} & o_2 \\ v_{x3} & v_{y3} & v_{z3} & o_3 \end{bmatrix} = \begin{bmatrix} r_{11}/a & r_{12}/b & r_{13}/c & t'_1/d \\ r_{21}/a & r_{22}/b & r_{23}/c & t'_2/d \\ r_{31}/af & r_{32}/bf & r_{33}/cf & t'_3/df \end{bmatrix}$$

Known — a and f are both Unknown

f

$$\begin{bmatrix} v_{x1} & v_{y1} & v_{z1} & o_1 \\ v_{x2} & v_{y2} & v_{z2} & o_2 \\ fv_{x3} & fv_{y3} & fv_{z3} & fo_3 \end{bmatrix} = \begin{bmatrix} r_{11}/a & r_{12}/b & r_{13}/c & t'_1/d \\ r_{21}/a & r_{22}/b & r_{23}/c & t'_2/d \\ r_{31}/a & r_{32}/b & r_{33}/c & t'_3/d \end{bmatrix}$$

Orthogonal vectors — Orthogonal vectors

$$v_{x1}v_{y1} + v_{x2}v_{y2} + f^2 v_{x3}v_{y3} = 0 \quad \Rightarrow \quad f = \sqrt{\frac{v_{x1}v_{y1} + v_{x2}v_{y2}}{-v_{x3}v_{y3}}}$$

Summary

- Overview
- Background Knowledge
- Vanishing Point Expression
- Vanishing Point Estimation
- Application: Camera Calibration

Thank you for your listening!

If you have any questions, please come to me :-)