Computer Vision

Levente Hajder, Dmitry Chetverikov

Eötvös Loránd University, Faculty of Informatics

Camera Models and Calibration

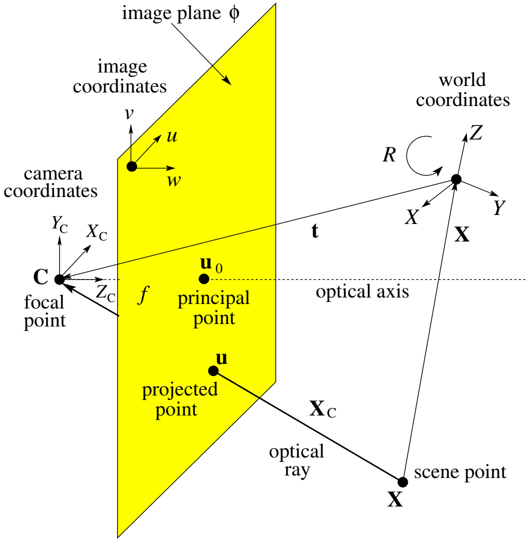
Outline

Outline

Gemoetric Imaging Models

- We introduce different **geometric** models
  - General perspective camera
  - Simplified camera models
- Perspective camera model equivalent to **pin-hole camera.**
  - camera obscura
- Pin-hole camera is close to real optics
  - → simple model of a thin optics
  - → Physical models are significantly complicated.
- However, a perspective camera is a very good **geometric approximation.**
- We address separately the following issues:
  - radiometric properties (brightness, colors)
  - geometric distortions

Perspective camera model



Notations: coordinates and transformations

- Coordinates

$$\mathbf{X} = [X, Y, Z]^T \qquad \text{world}$$
$$\mathbf{X}_c = [X_c, Y_c, Z_c]^T \qquad \text{camera}$$
$$\mathbf{u} = [u, v]^T \qquad \text{image plane}$$

- Homogeneous coordinates

$$\mathbf{X} = [X, Y, Z, 1]^T \qquad \text{world}$$
$$\mathbf{X}_c = [X_c, Y_c, Z_c, 1]^T \qquad \text{camera}$$
$$\mathbf{u} = [u, v, 1]^T \qquad \text{image plane}$$

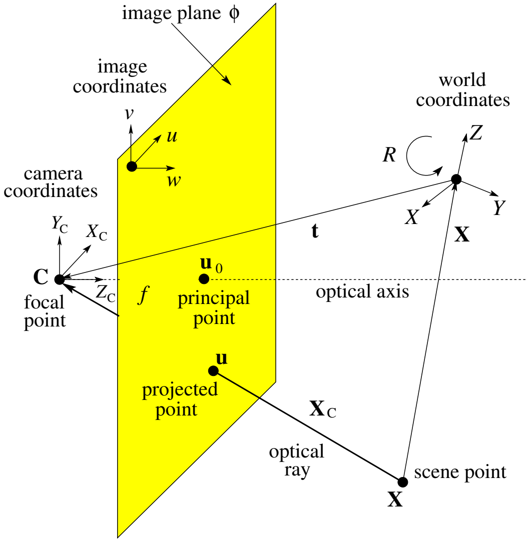
- Transformations
  - **R**: rotation (matrix)
  - **t**: translation (vector)

Notations: camera

$\mathbf{C}$	ϕ	f	$\mathbf{u}_0 = [u_0, v_0]^T$
focal point	*image plane*	*focal length*	*principal point*

- **C** focal point: central projection
- Optical ray: it connects a 3D point and focal point **C**
- Optical axis: Contains the focal point **C** and perpendicular to image plane ϕ
- Focal length: distance between **C** and ϕ.
- Principal point: the point in image plane where optical axis intersects ϕ

Perspective camera model



Translation and rotation

- World $\longrightarrow$ Camera
- Euclidean coordinates

$$\mathbf{X}_c = R(\mathbf{X} - \mathbf{t}) \tag{1}$$

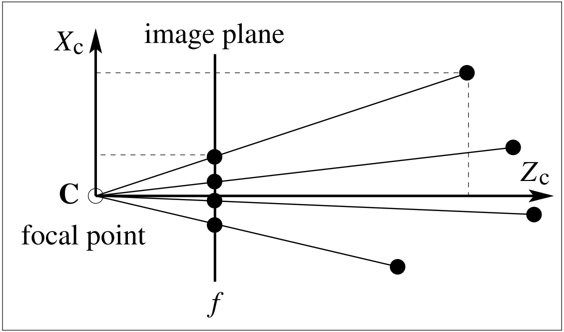
- Homogeneous coordinates

$$\mathbf{X}_c = R\,[\mathrm{I}| - \mathbf{t}] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \tag{2}$$

  - I is a 3×3- **identity matrix**
  - $[\mathrm{I}| - \mathbf{t}]$ is a 3×4 -matrix

  $\rightarrow$ I completed by colums $-\mathbf{t}$

Projection to an image plane



$$u = \frac{fk_u}{Z_c} X_c + u_0 \qquad (3)$$

$$v = \frac{fk_v}{Z_c} Y_c + v_0 \qquad (4)$$

- k_u, k_v is the horizontal/vertical pixel size.

→ their unit is *pixel/length*.

- Usually, $k_u = k_v = k$.

Projection using homogeneous coordinates

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \mathbf{K}\mathbf{X}_c \tag{5}$$

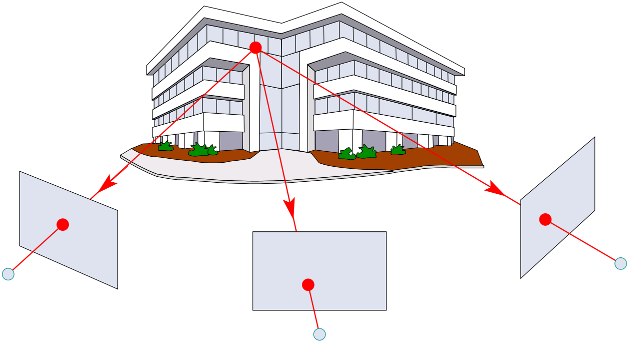
- $\sim$ homogeneous division yields scale ambiguity
- K is the (intrinsic) **calibration matrix**

$$K = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

  - upper triangular matrix
  - consists of 5 parameters, but only four are realistic

$\rightarrow$ fk_u, fk_v, u_0, v_0

Multi-view projection of a spatial point



- Locations of the same spatial point differ in images.
- Locations should be detected and/or tracked in the images.
  - $\rightarrow$ They are called *correspondences*.

Perspective camera model

- Goal: to determine the location of the projected 3D points in camera images.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \mathbf{KR}\left[\mathbf{I} | -\mathbf{t}\right] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = P \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \tag{7}$$

- $P \doteq \mathbf{KR}\left[\mathbf{I} | -\mathbf{t}\right]$ is the **projection matrix**
  - consists of 11 parameters

  $\rightarrow$ 5 in K, 3 in R, another 3 in $\mathbf{t}$.

Outline

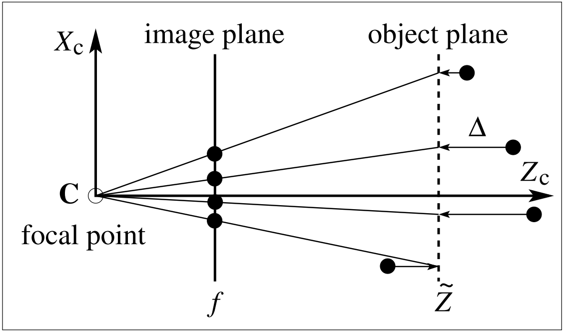
1. Camera Models
   - Perspective (pin-hole) camera
   - Weak-perspective camera
   - Comparison of camera models
   - Back-projection to 3D space
2. Homography
   - Homography estimation
   - Non-linear estimation by minimizing geometric error
3. Camera Calibration
   - Calibration by a spatial object
   - Calibration using a chessboard
   - Radial distortion
4. Summary

Weak-perspective projection 1/2

- It is assumed that the object is not 'too close' from the camera
  - change in depth is significantly smaller than the camera-object distance
- Object plane is parallel to the image plane
  - it is ideal if object center contains the center of gravity of the object.
- Objects are orthogonally projected into the object plane
- Then perspective projection is applied
  - as there is no difference in depth, location of principal point does not matter.

  → for the sake of simplicity, $u_0 = v_0 = 0$.

Weak-perspective projection 2/2



$$u = \frac{fk}{\widetilde{Z}_c} X_c + u_0 \qquad (8)$$

$$v = \frac{fk}{\widetilde{Z}_c} Y_c + v_0 \qquad (9)$$

- If pixel is a square, $k_u = k_v = k$
- It is also assumed that $Z_c \gg \Delta$

$\rightarrow$ $Z_c \approx \widetilde{Z}_c$, where $\widetilde{Z}_c$ is the common depth

$\rightarrow$ scaled orthographic projection

Weak-perspective camera model 1/2

- Translation and rotation in conjunction with weak-perspective projection:

$$u = q\mathbf{r}_1^T(\mathbf{X} - \mathbf{t}) + u_0 \tag{10}$$

$$v = q\mathbf{r}_2^T(\mathbf{X} - \mathbf{t}) + v_0, \quad \text{where} \tag{11}$$

$$q \doteq \frac{fk}{\widetilde{Z}_c}$$

- $\mathbf{r}_1^T$ and $\mathbf{r}_2^T$ are the first and second row vectors of rotation matrix R.
- $\mathbf{u}_0$ represents offset: $\longrightarrow u_0 = v_0 = 0$

$$u = q\mathbf{r}_1^T(\mathbf{X} - \mathbf{t}) \tag{12}$$

$$v = q\mathbf{r}_2^T(\mathbf{X} - \mathbf{t}) \tag{13}$$

Weak-perspective camera model 2/2

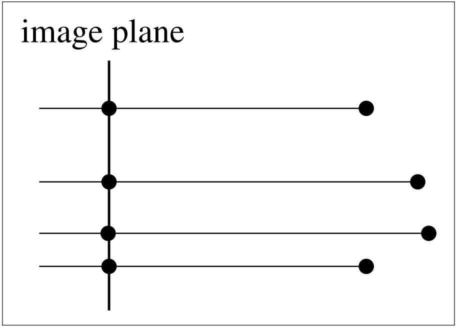
- Projection can be written with the help of a weak-perspective camera matrix:

$$\begin{bmatrix} u \\ v \end{bmatrix} = [M|\mathbf{b}] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}, \quad \text{where} \tag{14}$$

$$M \doteq q \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \end{bmatrix}, \qquad \mathbf{b} \doteq - \begin{bmatrix} q\mathbf{r}_1^T\mathbf{t} \\ q\mathbf{r}_2^T\mathbf{t} \end{bmatrix}$$

- Model has 6 degree of freedom (DoF)
    - if $k_u \neq k_v$, DOF=7
- There is no scale ambiguity.

orthographic projection



- Orthogonal projection can be applied if object
  - is far from the camera
  - depth is relatively static
- Model has 5 degree of freedom (DoF)
  - R, t_1, t_2

Outline

Affine camera

- **General affine camera**

$$\mathbf{u} = M_{2\times3}\mathbf{X} + \mathbf{t}$$

  - 8 degrees of freedom
  - $M_{2\times3}$ is a 2×3matrix with rank two

- Hierarchy of affine cameras
  - general affine camera

$$\Downarrow$$

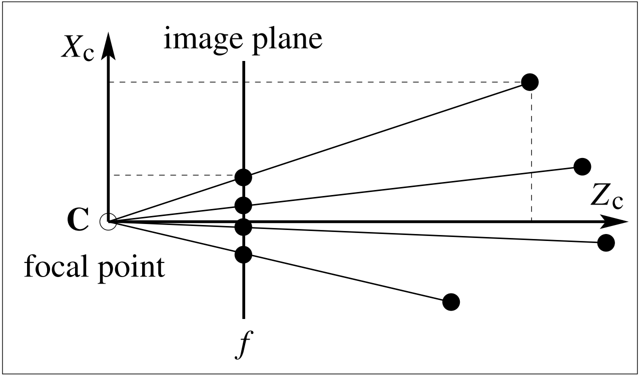
  - more constraints,
  - less DoFs

Herarchy of affine camera models

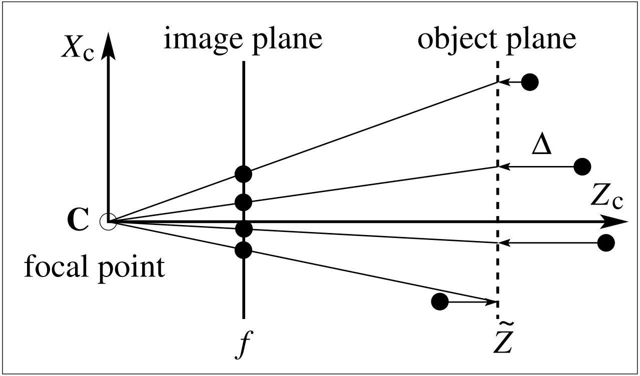
- **Weak-perspective projection**
  - 7 degrees of freedom ($k_u \neq k_v$)
- **Scaled orthographic projection**
  - six degrees of freedom
  - orthogonal projection + isotropic scale

  $\rightarrow$ if $k_u = k_v$, it is a scaled orthographic projection
- **Orthogonal projection**
  - five degrees of freedom

Perspective projection



Weak-perspective projection



Applicability of weak-perspective projection

- Projection error of weak-perspective projection

$$\mathbf{X}_c^{\text{weak}} - \mathbf{X}_c^{\text{proj}} = \frac{\Delta}{\widetilde{Z}_c}\mathbf{X}_c^{\text{proj}}$$

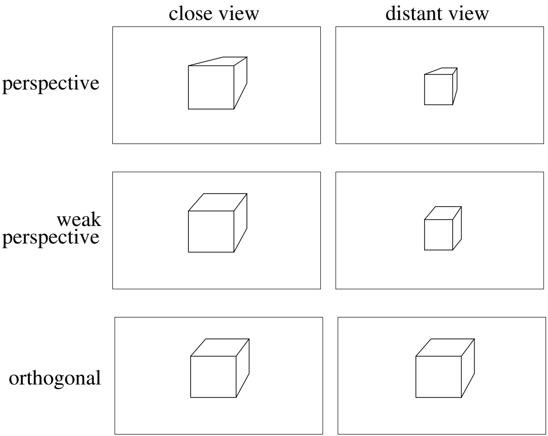
→ with respect to real location
  - Δ: distance betwen point and object plane
  - $\widetilde{Z}_c$ mean of depth values

→ Weak-perspective projection applicable if
  - $\Delta \ll \widetilde{Z}_c$

Benefits & disadvantages of weak-persective projection

- Advantages over real perspective projection
  - no scale ambiguity
  - less parameters to be estimated
  - → accuracy of estimation can be better
  - simpler
  - → closed-form solutions exist for several problems
- Disadvantages
  - It is only an approximation of real projection
  - → less accurate if conditions do not hold

Comparison of the projection models



effect	perspective	weak-persp.	orthogonal
change in sizes	yes	yes	no
persp. distortion	yes	no	no

Outline

Back-projection of a point 1/2

- Projection

$$u = \frac{fk_u}{Z_c} X_c + u_0$$
$$v = \frac{fk_v}{Z_c} Y_c + v_0$$

- Back projection by expressing spatial coordinates:

$$X_c = \frac{Z_c}{fk_u}(u - u_0)$$
$$Y_c = \frac{Z_c}{fk_v}(v - v_0)$$
$$Z_c = Z_c$$

Back-projection of a point 2/2

- Matrix form

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = Z_c \begin{bmatrix} \frac{1}{fk_u} & 0 & -\frac{u_0}{fk_u} \\ 0 & \frac{1}{fk_v} & -\frac{v_0}{fk_v} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = Z_c K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (15)$$

  - where the calibration matrix is as follows:

$$K = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

- a 3D point is ambiguous w.r.t. depth
  $\rightarrow$ a point in image represents a line in 3D space

Back projection by homogeneous coordinates

- Projection of 3D point to an image plane:

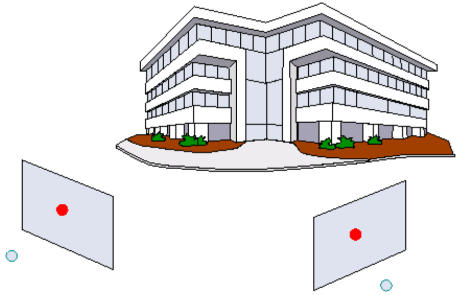
$$\mathbf{u} = P\mathbf{X}$$

- Back-projection yields a line:

$$\mathbf{X}(\lambda) = (1-\lambda)P^{+}\mathbf{u} + \lambda\mathbf{C}$$

  - It is a line written by a parameter λ
- P^{+} is the pseudo-inverse of P
  - $PP^{+} = \mathrm{I}$ (I : identity matrix)
  - $P^{+} = P^{\mathrm{T}}\left(PP^{\mathrm{T}}\right)^{-1}$
- The line contains
  - point $P^{+}\mathbf{u}$ ($\lambda = 0$)
  - Focal point **C** of camera
    - **C** is the null-vector of matrix P

Back projection and triangulation



- To estimate a 3D points
  - at least two calibrated cameras
  - and two corresponding points in the images are required.
- Estimation of 3D coordinates is called **triangulation** in computer vision.

Outline

Homography

- General, $(n+1)$-dimension case
  - P^n: n-dimensional space
  - $R^{(n+1)}$ extended space of P^n
  - transformation $P^n \rightarrow P^d$ is a homography, it is a linear transformation $R^{(n+1)}$
  - It is applied using homogeneous coordinates:

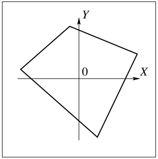
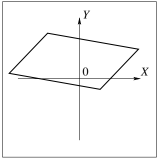
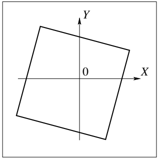
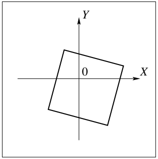
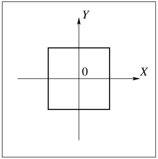
$$\mathbf{u}' \sim H\mathbf{u}$$

$\rightarrow$ H is a non-singular $(n+1) \times (n+1)$ matrix

- 3D case: $n+1=3$
  - P^2 is a plane in R^3
  - A homography is a projective transormation between two planes

  $\rightarrow$ it is unequivocal
  - Lines remain lines after homographic transformation.

Special cases of a homography

projective	affine	similarity	metric	identity
				
$\det H \neq 0$	$H = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	$H = \begin{bmatrix} sR & -R\mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	$H = \begin{bmatrix} R & -R\mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	$H = E$
	$\det A \neq 0$	$R^T R = E$	$R^T R = E$	
	$\det R = 1$	$\det R = 1$		
		$s > 0$		

Plane-plane homography (1)

- A planar pattern is given in 3D space.
- Two different images are taken:

$$\mathbf{u} = \lambda_1 \mathbf{P}_1 \mathbf{X} \quad \mathbf{u}' = \lambda_2 \mathbf{P}_2 \mathbf{X}$$

- Origin and oriantation of coordinate system can be freely selected
  - Let plane $Z = 0$ be the plane of the pattern
  - Then an arbitrary point within the pattern is $\mathbf{X_i} = [X_i, Y_i, 0, 1]^T$.
- Projection is more simple:

$$\mathbf{u}_i = \lambda_1 \tilde{\mathbf{P}}_1 \tilde{\mathbf{X}}_i \quad \mathbf{u'}_i = \lambda_2 \tilde{\mathbf{P}}_2 \tilde{\mathbf{X}}_i$$

- where $\tilde{\mathbf{X}}_i = [X_i, Y_i, 1]^T$. Matrices $\tilde{\mathbf{P}}_1$ and $\tilde{\mathbf{P}}_2$ is the original $\mathbf{P_1}$ and $\mathbf{P_2}$ matrices, removing the third column.

Application of homography (1)

- $\tilde{\mathbf{P}}_1$ and $\tilde{\mathbf{P}}_2$ are 3×3 square matrices
  - They are invertible.
- Spatial points:

$$\frac{1}{\lambda_1}\tilde{\mathbf{P}}_1^{-1}\mathbf{u} = \mathbf{X} \quad \frac{1}{\lambda_2}\tilde{\mathbf{P}}_2^{-1}\mathbf{u}' = \mathbf{X}$$

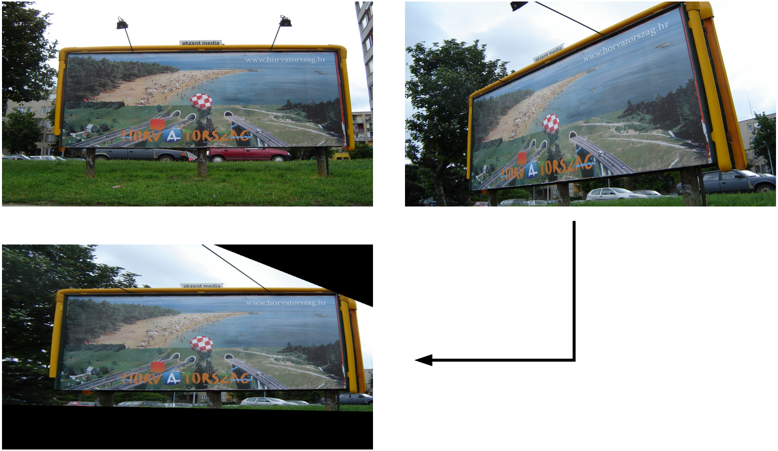
- Image coordinates can be computed from each other:

$$\mathbf{u}' = \frac{\lambda_2}{\lambda_1}\tilde{\mathbf{P}}_2\tilde{\mathbf{P}}_1^{-1}\mathbf{u}$$

- Transformation is given by the 3×3 matrix $\frac{\lambda_2}{\lambda_1}\tilde{\mathbf{P}}_2\tilde{\mathbf{P}}_1^{-1}$.

  $\rightarrow$ This is a homography.

Application of homography (1) : transformation of planar patterns



Application of homography (1/b) : Inverse Perspective Mapping

- For vision system of autonomous vehicles, the road is one of the main focuses of attention.
  - The road is a planar surface.
  - $\rightarrow$ It can be rectified by a homography.
- Objects can be more accurately detected in rectified images.
  - The distances can also be measured and visualized.

Application of homography (1/b) : Inverse Perspective Mapping

Left: Original image. Right: Rectified image.

Application of homography (2)

- A 3D world is given, two images are taken from the same focal point. Only camera orientations differ.
  - $\rightarrow$ Input for a panoramic image.
- The origin is selected as the common focal points of the images.
- Camera projection matrices: $\mathbf{P}_1 = \mathbf{K}_1[\mathbf{R}_1|0]$ and $\mathbf{P}_2 = \mathbf{K}_2[\mathbf{R}_2|0]$
- Projection: $\mathbf{u} = \mathbf{K}[\mathbf{R}|0][X, Y, Z, 1]^T$. Homogeneous (last) coordinate does not effect result.
- Transformation between two corresponding image locations:

$$\mathbf{u}' = \mathbf{K}_2\mathbf{R}_2{\mathbf{R}_1}^T{\mathbf{K}_1}^{-1}\mathbf{u}$$

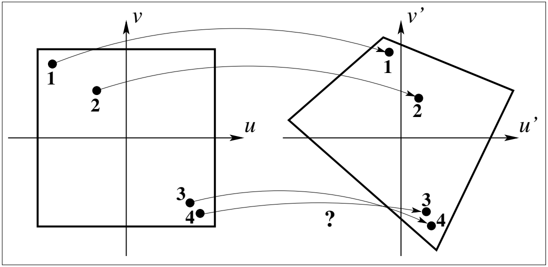
- Transformation is represented by 3×3 matrix $\mathbf{K}_2\mathbf{R}_2{\mathbf{R}_1}^T{\mathbf{K}_1}^{-1}$.
  - $\rightarrow$ This is a homography as well.

Application of homography (2) : panoramic imaging

Outline

1. Camera Models
   - Perspective (pin-hole) camera
   - Weak-perspective camera
   - Comparison of camera models
   - Back-projection to 3D space
2. Homography
   - Homography estimation
   - Non-linear estimation by minimizing geometric error
3. Camera Calibration
   - Calibration by a spatial object
   - Calibration using a chessboard
   - Radial distortion
4. Summary

Point-correspondence-based homography estimation



- m point correspondences are given. $\mathbf{u}_i \rightarrow \mathbf{u}'_i$:
  $\mathbf{u}'_i \sim H\mathbf{u}_i, \quad i = 1, \ldots, m$
- Task: estimate H
  - at least $m = n + 2$ correspondences are required
    - planar homography: at least four points needed.
  - For more points, problem is over-determined.
- In case of outliers: robust estimation
  - Robustification requires more corresponences.

Homography written by point locations

$$\alpha \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix},$$

where $\alpha \neq 0$ is an unknown scale factor.

Transformation yields

$$u' = \frac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + h_{33}} = \frac{\mathbf{h}_1^T\mathbf{u}}{\mathbf{h}_3^T\mathbf{u}}, \quad (16)$$

$$v' = \frac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + h_{33}} = \frac{\mathbf{h}_2^T\mathbf{u}}{\mathbf{h}_3^T\mathbf{u}}, \quad (17)$$

where $\mathbf{h}_i$ is the i-th row of matrix $\mathbf{H}$.

Linear estimation of planar homography 1/2

Equations are multiplied by the common denominator

$$(h_{31}u + h_{32}v + h_{33})u' = h_{11}u + h_{12}v + h_{13} \tag{18}$$

$$(h_{31}u + h_{32}v + h_{33})v' = h_{21}u + h_{22}v + h_{23} \tag{19}$$

For the i-th point, two homogeneous equations are obtained as $A_i\mathbf{h} = 0$, where

$$A_i = \begin{bmatrix} u_i & v_i & 1 & 0 & 0 & 0 & -u_iu_i' & -v_iu_i' & -u_i' \\ 0 & 0 & 0 & u_i & v_i & 1 & -u_iv_i' & -v_iv_i' & -v_i' \end{bmatrix}, \tag{20}$$

$$\mathbf{h} = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}]^T \tag{21}$$

For all points, $A\mathbf{h} = \mathbf{0}$ linear system of equations should be solved, where

$$A = [A_1, A_2, \ldots, A_m]^T$$

Linear estimation of planar homography 2/2

- Trivial solution $\mathbf{h} = \mathbf{0}$ is discarded
  - $\mathbf{h}$ can be determined up to a scale
  - $\rightarrow$ the norm is fixed: $\|\mathbf{h}\| = 1$
- If there are $m = 4$ correspondences; or $m > 4$, but data are noisy-free
  - if rank of A equals 8, exact solution can be obtained.
- If $m > 4$ and data are contaminated
  - only an estimate can be computed,
  - by minimizing $\|A\mathbf{h}\|$, subject to $\|\mathbf{h}\| = 1$.
  - $\rightarrow$ Optimal solution in the least squares sense is the eigenvalue of $A^T A$ corresponding to the smallest eigenvalue.

Properties of linear estimation 1/2

- Linear method
  $\rightarrow$ unequivocal, clear solution
- Low computational demand
  $\rightarrow$ fast execution
- The cost function of the estimation is determined by ϵ

$$\epsilon = \|A\mathbf{h}\|$$

- ϵ-is an algebraic distance
  - no direct geometric meaning
  $\rightarrow$ minimization of geometric distance(s) preferred

Properties of linear estimation 2/2

- Not robust
  - noisy correspondences
  - Works well if there are no outliers
  - → one outlier can destroy the good result
  - → (*breakdown point*) is very low
- Due to numerical computation, data normalization required
  - elements in coefficient should be in the same order of magnitude
  - → translation: origo should be at the center of gravity
  - → scale: spread should be set to $\sqrt{2}$
- Numerical optimization is usually applied, linear method yields initial value

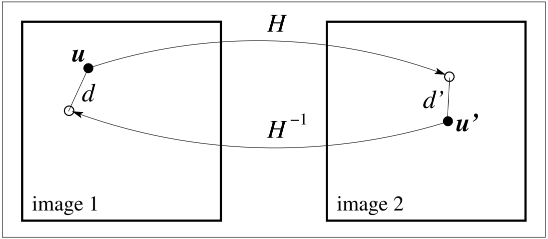
Data normalization

- Coordinate system can be freely selected.
- Original homography: $[u_2, v_2, 1]^T \sim \mathbf{H}[u_1, v_1, 1]^T$
  - Modified coordinates: $[u'_1, v'_1, 1]^T = \mathbf{T}_1[u_1, v_1, 1]^T$ and $[u'_2, v'_2, 1]^T = \mathbf{T}_2[u_2, v_2, 1]^T$
    - where $\mathbf{T}_1$ and $\mathbf{T}_2$ are affine transformations (translation + scale)
- Projection by the modified homography: $[u'_2, v'_2, 1]^T \sim \mathbf{H'}[u'_1, v'_1, 1]^T$
- After substitution: $\mathbf{T}_2[u_2, v_2, 1]^T \sim \mathbf{H'}\mathbf{T}_1[u_1, v_1, 1]^T$
- Then: (multiplication by $\mathbf{T}_2^{-1}$ from the left): $[u_2, v_2, 1]^T \sim \mathbf{T}_2^{-1}\mathbf{H'}\mathbf{T}_1[u_1, v_1, 1]^T$
- Thus, $\mathbf{H} = \mathbf{T}_2^{-1}\mathbf{H}'\mathbf{T}_1$ or $\mathbf{H}' = \mathbf{T}_2\mathbf{H}\mathbf{T}_1^{-1}$:

Outline

1. Camera Models
   - Perspective (pin-hole) camera
   - Weak-perspective camera
   - Comparison of camera models
   - Back-projection to 3D space
2. Homography
   - Homography estimation
   - Non-linear estimation by minimizing geometric error
3. Camera Calibration
   - Calibration by a spatial object
   - Calibration using a chessboard
   - Radial distortion
4. Summary

Minimization of projection error



- Projection error

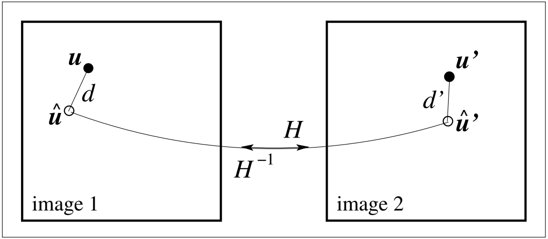
$$\sum_i \left[d(\mathbf{u}_i', H\mathbf{u}_i)\right]^2$$

- Symmetric projection error

$$\sum_i \left(\left[d(\mathbf{u}_i', H\mathbf{u}_i)\right]^2 + \left[d(\mathbf{u}_i, H^{-1}\mathbf{u}_i')\right]^2\right)$$

- Refinement of homography **H**: 9 variables, 8 DoFs

Minimization of reprojection error



- If both **H** and $\mathbf{u}_i$ are refined: $9+2m$ variable
- Homography $\widehat{H}$ and points $\hat{\mathbf{u}}_i, \hat{\mathbf{u}}'_i$ should be refined by minimizing the re-projection error

$$\sum_i \left([d(\mathbf{u}_i, \hat{\mathbf{u}}_i)]^2 + \left[d(\mathbf{u}'_i, \hat{\mathbf{u}}'_i)\right]^2 \right) \text{ subject to } \hat{\mathbf{u}}'_i = \widehat{H}\hat{\mathbf{u}}_i \ \forall i$$

Non-linear estimation of a homography

- Cost function is non-convex
  - → Global minimum cannot be guaranteed
  - → Good initilal value required
- Two-step approach
  1. Linear estimation first,
  2. then numerical optimization, e.g. Levenberg-Marquardt applied
- For outlier handling, robustification is required
  - outlier filtering, robust statistics
  - RANSAC, M-estimation, median, ...

Properties of non-linear estimation

- Benefits
  - Geometric error (meaningful) can be applied
  - Accurate
  - Can be straightforwardly robustified
- Disadvantages
  - Numerical methods $\longrightarrow$ local minima can exist
  - $\rightarrow$ Results depend on initial values
  - Higher time demand

Outline

Goals of Camera calibration 1/2

- Camera calibration is an important pre-step of 3D reconstruction
  - Estimation of intrinsic parameters
  - Extrinsic parameters (position, orientation)
- Two main types of calibration
  - Photogrammetric calibration, separate process
  - Auto-calibration: joint estimation of camera parameters + 3D scenes

Goals of Camera calibration 2/2

- Better camera parameters $\rightarrow$ better 3D vision
  - Stereo (two-view) calibration is possible if cameras are pre-calibrated
- Calibration requires known positions of
  - Feature points and
  - Lines
- **Auto-calibration**
  - More difficult
  - Less accurate
  - Applied if pre-calibration is impossible

  $\rightarrow$ This course does not deal with auto-calibration.

Camera calibration with calibration object

- Know 3D scene required
  - partally or fully known geometry
  - detectable features on the images
  - manual intervention can be applied
- Partially-known geometry
  - parallel lines
  - perpendicular edges
  - → e.g. a building
- Known objects
  - static 3D point cloud
  - → e.g. calibration cube
  - → or calibration chessboard

Outline

Calibration by a spatial object

- Given m point correspondence between 3D scene and images plane $\mathbf{X}_i \rightarrow \mathbf{u}_i$: $\mathbf{u}_i \sim P\mathbf{X}_i, \quad i = 1, \ldots, m$
- Task: estimation of $P = KR\,[I| - t]$.
  - At least 6 correspondences required
  - Over-determined system
- Wrong correspondence $\longrightarrow$ robust methods
  - Many correspondences $\longrightarrow$ outlier detection

Calibration by Cartesian coordinates

$$\alpha \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$

where $\alpha \neq 0$ is an arbitrary scale factor.

Equations can be rewritten as

$$u = \frac{P_{11}X + P_{12}Y + P_{13}Z + P_{14}}{P_{31}X + P_{32}Y + P_{33}Z + P_{34}} = \frac{\mathbf{p}_1^T \mathbf{X}}{\mathbf{p}_3^T \mathbf{X}}, \quad (22)$$

$$v = \frac{P_{21}X + P_{22}Y + P_{23}Z + P_{24}}{P_{31}X + P_{32}Y + P_{33}Z + P_{34}} = \frac{\mathbf{p}_2^T \mathbf{X}}{\mathbf{p}_3^T \mathbf{X}}, \quad (23)$$

where $\mathbf{p}_i$ is the i-th row of projection matrix P.

Linear estimation of a projection matrix 1/2

Equations are multiplied by the common denominator:

$$(P_{31}X + P_{32}Y + P_{33}Z + P_{34})u = P_{11}X + P_{12}Y + P_{13}Z + P_{14} \quad (24)$$

$$(P_{31}X + P_{32}Y + P_{33}Z + P_{34})v = P_{21}X + P_{22}Y + P_{23}Z + P_{24} \quad (25)$$

For the i-th point, $A_i\mathbf{p} = 0$, where

$$A_i = \begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i & -v_i \end{bmatrix} \quad (26)$$

$$\mathbf{p} = [P_{11}, P_{12}, P_{13}, P_{14}, P_{21}, P_{22}, P_{23}, P_{24}, P_{31}, P_{32}, P_{33}, P_{34}]^T \quad (27)$$

For all the points, a homogeneous linear system of equation obtained in the form $A\mathbf{p} = \mathbf{0}$ where

$$A = [A_1, A_2, \dots, A_m]^T$$

Linear estimation of a projection matrix 2/2

- $\mathbf{p} = \mathbf{0}$ trivial solution omitted.
  - estimation obtained up to a scale
  - $\rightarrow$ norm is fixed as $\|\mathbf{p}\| = 1$
- For noiseless case
  - rank of **A** is 11, perfect solution is obtained
- For over-determined and noisy case,
  - only estimation can be computed
  - minimization of $\|A\mathbf{p}\|$ subject to: $\|\mathbf{p}\| = 1$.
  - $\rightarrow$ optimal solution if the eigenvector of $A^T A$ corresponding to the least eigenvalue.
  - solution can be obtained by Singular Value Decomposition (SVD) as well.

Decomposition of a projection matrix

- Structure of a projection matrix:

$$\mathbf{P} = \mathbf{K}\mathbf{R}\left[\mathbf{I}| - \mathbf{t}\right] \tag{28}$$

- First three columns of matrix $\mathbf{P}$: $\mathbf{P}_{3\times 3} = \mathbf{K}\mathbf{R}$
  - Decomposition can be obtained by RQ - decomposition
  - It decomposes $\mathbf{P}$ into product of an upper triangular and an othonormal matrices
- Last column of matrix $\mathbf{P}$:

$$\mathbf{p}_4 = -\mathbf{K}\mathbf{R}\mathbf{t} \tag{29}$$

- Thus,

$$\mathbf{t} = -\mathbf{R}^T\mathbf{K}^{-1}\mathbf{p}_4 \tag{30}$$

Data normalization

- Point coordinates can be normalized similarly to homography estimation
- Original transformation: $[u, v, 1]^T \sim \mathbf{P}[X, Y, Z, 1]^T$
- Normalizing transformations: $[u', v', 1]^T = \mathbf{T}_{2D}[u, v, 1]^T$ and $[X', Y', Z', 1]^T = \mathbf{T}_{3D}[X, Y, Z, 1]^T$
  - $\mathbf{T}_{2D}$ 2D transformation(s) (size: 3×3)
  - $\mathbf{T}_{3D}$ 3D transformation(s) (size: 4×4)
- Projection by normalized coordinates: $[u', v', 1]^T \sim \mathbf{P}'[X', Y', Z', 1]^T$
- Solution applied normalized coordinates:
  - $\mathbf{P} = \mathbf{T}_{2D}^{-1}\mathbf{P}'\mathbf{T}_{3D}$ or $\mathbf{P}' = \mathbf{T}_{2D}\mathbf{P}\mathbf{T}_{3D}^{-1}$.

Outline

1. Camera Models
   - Perspective (pin-hole) camera
   - Weak-perspective camera
   - Comparison of camera models
   - Back-projection to 3D space
2. Homography
   - Homography estimation
   - Non-linear estimation by minimizing geometric error
3. Camera Calibration
   - Calibration by a spatial object
   - Calibration using a chessboard
   - Radial distortion
4. Summary

Chessboard-based camera calibration

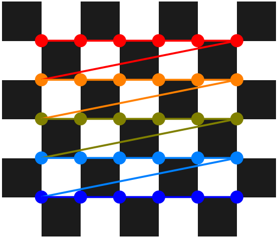
- Z. Zhang, Microsoft Research, 1998.
- Easy and accurate method
- Frequently-used
- Non-perspective distortion can be handled
- Chessboard can be easily printed
- Efficient implementations available, e.g. in OpenCV
- See demos on Youtube
- Disadvantages
  - Multiple images required
  - Avoid checked patterns on shirts :(

Box-based camera calibration

- R.Y. Tsai, 1986.
- Non-perspective distortion can be handled
- Less user-friendly than chessboard-based one
  $\rightarrow$ not frequently used
- Hard to manufacture a precize calibration box
  - especially in large dimensions

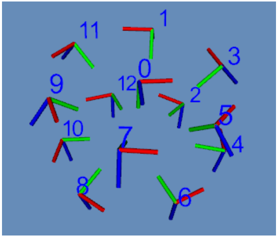
  $\rightarrow$ it is difficult to calibrate a camera using a small cube
- Benefits
  - One static image is satisfactory

Calibration of a camera-system using chessboards



Detected corners

Chessboard



Extrinsic camera params

- Chessboard patterns are assymetric $\longrightarrow$ corner detection unambiguous.
- Orientations of chessboard in images should differ.
- Extrinsic parameters can also be retrieved.

Chessboard-based calibration

- Main steps of calibration
  1. Homography exists between the calibration plane and an image
  2. Camera intrinsics in matrix K can be computed from homographies

- World coordinate is fixed to the board
  - Axis Z is perpendicular to the board $\longrightarrow Z = 0$ is the board plane

$$\alpha \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \quad \text{where} \tag{31}$$

  - $\mathbf{r}_1$ and $\mathbf{r}_2$ are the first two rows of rotation $\mathbf{R}$
  - $\mathbf{H} \doteq \mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}$

- Task is to (1) estimate **H**, then (2) computate intrinsic matrix **K**

Chessboard-based calibration

- Corners of chessboard fields can be easily detected.
  - chessboard -> image correspondences $\mathbf{x}_i \rightarrow \mathbf{u}_i$ used
  - at least 4 required
  - $\rightarrow$ More correspondences needed for contaminated data
  - subpixel corner detection $\rightarrow$ improved accuracy
  - $\rightarrow$ intersections of lines
- Estimation of homography $\mathbf{H}$
  - linear estimation minimizing algebraic error
  - non-linear estimation considering geometric error
- Homography $\mathbf{H}$ can be estimated up to an unknown scale
- Let $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3$ denote the columns of $\mathbf{H}$: $\mathbf{H} \doteq \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{bmatrix}$

Computation of intrinsic parameters 1/3

- For homography matrix, the following equations are valid:

$$\begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix} \sim \mathbf{K}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}$$
$$\begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2\end{bmatrix} \sim \mathbf{K}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2\end{bmatrix}$$
$$\mathbf{K}^{-1}\begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2\end{bmatrix} \sim \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2\end{bmatrix}$$

- $\mathbf{r}_1$ and $\mathbf{r}_2$ are orthonormal, therefore

$$\mathbf{r}_1^T\mathbf{r}_2 = \mathbf{h}_1^T\mathbf{S}\mathbf{h}_2 = 0, \tag{32}$$
$$\|\mathbf{r}_1\|^2 - \|\mathbf{r}_2\|^2 = \mathbf{h}_1^T\mathbf{S}\mathbf{h}_1 - \mathbf{h}_2^T\mathbf{S}\mathbf{h}_2 = 0, \tag{33}$$

  - where $\mathbf{S} \doteq \mathbf{K}^{-T}\mathbf{K}^{-1}$, $\mathbf{K}^{-T} \doteq (\mathbf{K}^{-1})^T$
- This is a linear problem w.r.t. the elements of **S**. $\rightarrow$ They can be optimally estimated.

Computation of intrinsic parameters 2/3

- Elements of calibration matrix **K**:

$$\mathbf{K} = \begin{bmatrix} fk_u & s & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

- $\mathbf{S} = \lambda \mathbf{K}^{-T}\mathbf{K}^{-1}$

$$\frac{\mathbf{S}}{\lambda} = \begin{bmatrix} \frac{1}{(fk_u)^2} & -\frac{s}{(fk_u)^2 fk_v} & \frac{u_0 fk_v - v_0 s}{(fk_u)^2 fk_v} \\ -\frac{s}{(fk_u)^2 fk_v} & \frac{s^2}{(fk_u)^2 (fk_v)^2} + \frac{1}{(fk_v)^2} & \frac{-s(u_0 fk_v - v_0 s)}{(fk_u)^2 (fk_v)^2} + \frac{v_0}{(fk_v)^2} \\ \frac{u_0 fk_v - v_0 s}{(fk_u)^2 fk_v} & \frac{-s(u_0 fk_v - v_0 s)}{(fk_u)^2 (fk_v)^2} + \frac{v_0}{(fk_v)^2} & 1 + \frac{v_0^2}{fk_v^2} + \frac{(u_0 fk_v - v_0 s)^2}{(fk_u)^2 (fk_v)^2} \end{bmatrix}$$

- Matrix **S** has 5 parameters to be estimated: fk_u, fk_v, u_0, v_0, s
- Each chessboard image yields 2 equations $\longrightarrow$ at least 3 images required
- More images $\longrightarrow$ overdetermined system
- Robustification also requires many images

Computation of intrinsic parameters 3/3

- Matrix **S** $\longrightarrow$ closed-form solution for intrinsic parameters(**K**)

$$
\begin{aligned}
v_0 &= \frac{(S_{11}S_{23}-S_{21}S_{13})}{S_{11}S_{22}-S_{12}^2} \\
\lambda &= S_{33} - \frac{S_{13}^2+v_0(S_{12}S_{13}-S_{11}S_{23})}{S_{11}} \\
fk_u &= \sqrt{\frac{\lambda}{S_{11}}} \\
fk_v &= \sqrt{\lambda S_{11}/(S_{11}S_{22}-S_{12}^2)} \\
s &= -S_{12}fk_u^2 fk_v/\lambda \\
u_0 &= sv_0/fk_v - S_{13}{fk_v}^2/\lambda
\end{aligned}
$$

- Check: homework...

Computation of extrinsic parameters

$$\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} = \mathbf{K}^{-1} H \tag{34}$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \tag{35}$$

- For detailed description: Z. Zhang, Technical Report
- Implementation available in OpenCV, C++
- Matlab toolbox also exists
  - `http://sourceforge.net/projects/opencvlibrary/`

Outline

1. Camera Models
   - Perspective (pin-hole) camera
   - Weak-perspective camera
   - Comparison of camera models
   - Back-projection to 3D space
2. Homography
   - Homography estimation
   - Non-linear estimation by minimizing geometric error
3. Camera Calibration
   - Calibration by a spatial object
   - Calibration using a chessboard
   - Radial distortion
4. Summary

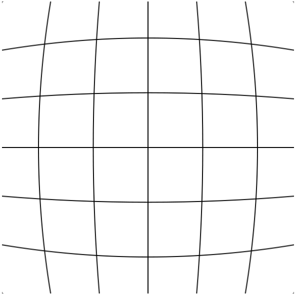
Radial distortion

- Non-perspective distorton is common for cheap or wide FoV (field of view) lenses
- Perspective model is only an approximation for real projection
  - e.g. projection of a line should be a straight line due to perspectivity

  → This is not always true for real cameras
- Usually, type of distortion is **radial distortion**
  - Two subtypes: **Barrel/pillow** distortion

  → barellel is more frequent
- Radial distortion has to be undistorted
  - Especially, when accurate 3D reconstruction should be achieved.
- Undistortion is usual part of camera calibration
  - It is included e.g. in OpenCV's calibration, in final numerical optimization

Radial distortion



barrell

pillow

- Straight lines become curves
- It is usual for wide FoV (small focal length)

Source of images: Wikipedia

Correction of radial distortion

$$\hat{\mathbf{u}} = \mathbf{u}_c + L(r)(\mathbf{u} - \mathbf{u}_c), \quad \text{ahol} \tag{36}$$

- $\mathbf{u}$: measured, $\hat{\mathbf{u}}$: corrected coordinates
- $\mathbf{u}_c$ center of distortion
  - it is usually assumed that $\mathbf{u}_c$ coincides with principal point $\mathbf{u}_0$.
- $L(r)$ is a cubic polynomial in r^2

$$L(r) = 1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6$$

  - $r = \|\mathbf{u} - \mathbf{u}_c\|$ is the distance from $\mathbf{u}_c$.
  - $L(r)$ is a Taylor approximation of the real distortion function

  $\rightarrow$ $\kappa_1, \kappa_2, \kappa_3$ are small real numbers

- Model is built in the non-linear homography estimation

  $\rightarrow$ Parameters $\kappa_1, \kappa_2, \kappa_3$ are stimated based on 2D geometric error

OpenCV: tangential distortion

$$\hat{x} = y_c + L_1(x, y)(x - x_c)$$
$$\hat{y} = y_c + L_2(x, y)(y - y_c)$$

- $L_{\{1,2\}}(x, y)$ are products of polynomials

$$L_1(x, y) = 1 + 2p_1 xy + p_2 \left(r^2 + 2x^2\right)$$
$$L_2(x, y) = 1 + 2p_2 xy + p_1 \left(r^2 + 2y^2\right)$$

  - $r = x^2 + y^2$ is the distance from the optical axis
  - p_1, p_2 are small real numbers
  - It is not mandatory to use all tangential parameters.
  - Tangential distortion does complete and not substitutes radial distortion.

Outline

1. Camera Models
   - Perspective (pin-hole) camera
   - Weak-perspective camera
   - Comparison of camera models
   - Back-projection to 3D space
2. Homography
   - Homography estimation
   - Non-linear estimation by minimizing geometric error
3. Camera Calibration
   - Calibration by a spatial object
   - Calibration using a chessboard
   - Radial distortion
4. Summary

Camera models and Calibration

1. Camera Models
   - Perspective (pin-hole) camera
   - Weak-perspective camera
   - Comparison of camera models
   - Back-projection to 3D space
2. Homography
   - Homography estimation
   - Non-linear estimation by minimizing geometric error
3. Camera Calibration
   - Calibration by a spatial object
   - Calibration using a chessboard
   - Radial distortion
4. Summary

References

- R.Hartley, A.Zisserman: "Multiple View Geometry in Computer Vision", Cambridge University Press
- M.Sonka, V.Hlavac, R.Boyle: "Image Processing, Analysis and Machine Vision", Thomson
- Y. Ma, S. Soatto, J. Kosecka, S. Shankar Sastry: "An Invitation to 3-D Vision", Springer
- Z. Zhang, "A Flexible New Technique for Camera Calibration", Microsoft Research, Technical Report MSR-TR-98-71, 1998
- Wikipedia: "Radial distortion (optics)"