Camera model and homographies

Camera parameters; lens distortion; homography estimation

Morten R. Hannemose, mohan@dtu.dk

February 11, 2022

02504 Computer vision course lectures,
DTU Compute, Kgs. Lyngby 2800, Denmark

Learning objectives

After this lecture you should be able to:

- explain the phenomelogical origin of lens distortion
- compensate for lens distortion in images
- explain the phenomelogical origin of the pinhole camera parameters
- derive the homography matrix
- explain the use of homographies in computer vision
- perform the linear estimation of a homography matrix
- use singular value decomposition (SVD) to solve linear systems

Presentation topics

- Pinhole camera
  - Intrinsics
- Lens distortion
  - Radial distortion
- Homographies
  - Application: image stitching (panorama)
- Homography estimation
  - Homogeneous least-squares solution

Pinhole camera

Recap last week

$$\boldsymbol{K} = \begin{bmatrix} f & 0 & \delta_x \\ 0 & f & \delta_y \\ 0 & 0 & 1 \end{bmatrix}$$

Camera intrinsics

- Focal length f
- Principal point $\delta x, \delta y$
- Skew parameters α and β

Focal length f

- The focal length is the distance from the pinhole to the image plane *in pixels*.
- This corresponds to the zoom level / field of view
  - Longer focal length = more zoomed in.

Dolly zoom/vertigo effect

What does the focal length do?

Moving camera closer while zooming out to make one object have a constant size is a popular effect

Real-life example

Simple example

Why $\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$ **and not** $\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & f \end{bmatrix}$**?**

Skew parameters α and β

Non-square pixels (α), and non-rectangular pixels (β)

$$\boldsymbol{K} = \begin{bmatrix} f & \beta f & \delta_x \\ 0 & \alpha f & \delta_y \\ 0 & 0 & 1 \end{bmatrix}$$

Orthographic projection

If all projection lines are parallel, then magnification $m = 1$

$$\boldsymbol{p} = \begin{bmatrix} P_x \\ P_y \end{bmatrix}$$

The pinhole model goes towards orthographic projection as the focal length goes to ∞

Projection matrix: Degrees of freedom

$$\mathcal{P} = \underbrace{\begin{bmatrix} f & \beta f & \delta_x \\ 0 & \alpha f & \delta_y \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{K}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{\begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix}}$$

Projection matrix: Degrees of freedom

$$\mathcal{P} = \underbrace{\begin{bmatrix} f & \beta f & \delta_x \\ 0 & \alpha f & \delta_y \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{K}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{\begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix}}$$

$\mathcal{P}$ is $3 \times 4 = 12$ numbers.

$5 + 3 + 3 + 1$ (scale) $= 12$ degrees of freedom.

Which physical properties do I need to derive the intrinsics: f, δx, δy, α, β?

Speculate; Given an unknown system, how would you estimate all parameters in $\mathcal{P}$?

Lens distortion

Help! My camera is not a pinhole camera?

Welcome to the real world!

Real lenses

Lenses are:

- complicated by many effects, intentional or non-intentional
- imperfect with, for example, defects
- different for each camera/lens

Almost all lenses have lens distortion!

Radial lens distortion

We distort points by a polynomial in the distance from the principal point (the radius $\boldsymbol{r}$)

Radial distortion equations

$$\boldsymbol{p}_\mathrm{d} = \boldsymbol{p}\left(1 + \Delta r(\|\boldsymbol{p}\|_2)\right)$$
$$\Delta r(r) = k_3 r^2 + k_5 r^4 + \dots$$

Barrel distortion

For example:

$$\Delta r(r) = -0.5r^2$$

Pincushion distortion

For example:

$$\Delta r(r) = 0.5r^2$$

Radial lens distortion in practice

Overfitting / extrapolation errors

$$\Delta r(r) = -0.1r^2 + \delta_{error}$$

Almost all lenses have lens distortion!

Projection under radial distortion

Projection:

$$\boldsymbol{q} = \boldsymbol{K} \cdot \Pi^{-1} \left[\text{dist}\left[\Pi\left([\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{Q}\right)\right]\right] \text{ , where}$$

$$\text{dist}\left(\boldsymbol{p}\right) = \boldsymbol{p} \cdot \left(1 + \Delta r(\|\boldsymbol{p}\|)\right).$$

This follows the math in OpenCV unlike the Lecture Notes

Undistorting an image

- Given a distorted image and the distortion coefficients, how do you undistort it?
- Does it involve inverting the distortion function?
- Discuss with your partner

Undistorting an image

- Undistorting the image, is equivalent to finding the intensity in every undistorted (integer) pixel location $\boldsymbol{p}_{ij}$
- Our distortion function tells us where to sample $\boldsymbol{p}_{ij}$ in the distorted image ($\boldsymbol{p}_{d,ij}$). The intensity can be computed using bilinear interpolation.
- Not necessary to invert the distortion function!

Questions? Short break

Homographies

Practical example

- No volunteers needed 😉

Practical example

- No volunteers needed 😉
- Knowing that a point seen with a camera lies on a plane lets us know exactly where on the plane it is.

The homography

Consider the following plane:

$$\begin{aligned} \boldsymbol{P} &= a\boldsymbol{A} + b\boldsymbol{B} + \boldsymbol{C}\,, \\ &= \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{C} \end{bmatrix} \begin{bmatrix} a \\ b \\ 1 \end{bmatrix}\,. \end{aligned}$$

$\boldsymbol{C}$ is a point on the plane, and $\boldsymbol{A}$ and $\boldsymbol{B}$ are vectors that span the plane.

The homography

Projections of points on a plane:

$$\boldsymbol{q} = \mathcal{P}\boldsymbol{P}_h\,,$$

$$= \mathcal{P}\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{C} \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} a \\ b \\ 1 \end{bmatrix}\,,$$

The homography

Projections of points on a plane:

$$\boldsymbol{q} = \mathcal{P}\boldsymbol{P}_h\,,$$

$$= \underbrace{\mathcal{P}\begin{bmatrix}\boldsymbol{A} & \boldsymbol{B} & \boldsymbol{C}\\ 0 & 0 & 1\end{bmatrix}}_{\boldsymbol{H}}\begin{bmatrix}a\\ b\\ 1\end{bmatrix}\,,$$

$\boldsymbol{H}$ is a 3×3 matrix called a homography.

The homography

H

Homographies between images

Two cameras photographing the same plane let's us establish pixel correspondence:

$$\boldsymbol{q}_1 = \boldsymbol{H}_1 \boldsymbol{Q}, \quad \boldsymbol{q}_2 = \boldsymbol{H}_2 \boldsymbol{Q}$$

$$\begin{aligned} \boldsymbol{q}_1 &= \boldsymbol{H}_1 \boldsymbol{H}_2^{-1} \boldsymbol{q}_2 \\ &= \boldsymbol{H} \boldsymbol{q}_2 \end{aligned}$$

Two view geometry without baseline

If two cameras are at the same point we can use a homography to relate their pixels

Image stitching

You get to implement this in week 9.

Homography multiplication

If you multiply a homography $\boldsymbol{H}$ by a scalar, how does that change the mapping it describes?

Homography multiplication

If you multiply a homography $\boldsymbol{H}$ by a scalar, how does that change the mapping it describes?

It doesn't!

$$\boldsymbol{q}_2 = \boldsymbol{H}\boldsymbol{q}_1$$
$$s\boldsymbol{q}_2 = (s\boldsymbol{H})\boldsymbol{q}_1$$

$\boldsymbol{q}_2$ and $s\boldsymbol{q}_2$ are the same point.

Homography degrees of freedom

- 3×3 matrix has 9 numbers
- Scale is arbitrary
  - A homography has 8 degrees of freedom
- We need 4 pairs of points to estimate a homography
  - Each point pair imposes two constraints (x and y)

Any two images of the same plane are related by a homography

Homography estimation

Homography estimation

Consider a number of points $\boldsymbol{q}_{1i}$ and $\boldsymbol{q}_{2i}$. The homography relates them as follows:

$$\boldsymbol{q}_{1i} = \boldsymbol{H}\boldsymbol{q}_{2i}$$

Homography rewritten i

The cross product of two parallel vectors is the zero vector:

$$\begin{aligned} \boldsymbol{q}_{1i} &= \boldsymbol{H}\boldsymbol{q}_{2i} \Rightarrow \\ \mathbf{0} &= \boldsymbol{q}_{1i} \times \boldsymbol{H}\boldsymbol{q}_{2i}\,, \end{aligned}$$

we can isolate $\boldsymbol{H}$ in this expression (next slide) ...

Homography rewritten ii

$$
\begin{aligned}
\mathbf{0} &= \boldsymbol{q}_{1i} \times \boldsymbol{H}\boldsymbol{q}_{2i}\,, \\
&= \boldsymbol{B}^{(i)}\mathrm{flatten}(\boldsymbol{H}^{\mathsf{T}})\,,
\end{aligned}
$$

$$
\boldsymbol{B}^{(i)} = \begin{bmatrix}
0 & -x_{2i} & x_{2i}y_{1i} & 0 & -y_{2i} & y_{2i}y_{1i} & 0 & -1 & y_{1i} \\
x_{2i} & 0 & -x_{2i}x_{1i} & y_{2i} & 0 & -y_{2i}x_{1i} & 1 & 0 & -x_{1i} \\
-x_{2i}y_{1i} & x_{2i}x_{1i} & 0 & -y_{2i}y_{1i} & y_{2i}x_{1i} & 0 & -y_{1i} & x_{1i} & 0
\end{bmatrix},
$$

$$
\mathrm{flatten}(\boldsymbol{H}^{\mathsf{T}}) = \begin{bmatrix} H_{11} & H_{21} & H_{31} & H_{12} & H_{22} & H_{32} & H_{13} & H_{23} & H_{33} \end{bmatrix}^{\mathsf{T}}
$$

Defining $\boldsymbol{B}^{(i)}$

The $\boldsymbol{B}^{(i)}$ matrices can be implemented with the Kronecker product:

$$\begin{aligned}\boldsymbol{B}^{(i)} &= \boldsymbol{q}_{2i}^{\mathsf{T}} \otimes [\boldsymbol{q}_{1i}]_{\times} \\ &= \boldsymbol{q}_{2i}^{\mathsf{T}} \otimes \begin{bmatrix} 0 & -1 & y_{1i} \\ 1 & 0 & -x_{1i} \\ -y_{1i} & x_{1i} & 0 \end{bmatrix}\end{aligned}$$

Homography matrix solution

Given n corresponding points, we define $\boldsymbol{B}$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}^{(1)} \\ \boldsymbol{B}^{(2)} \\ \vdots \\ \boldsymbol{B}^{(n)} \end{bmatrix}$$

$$\mathbf{0} = \boldsymbol{B}\ \text{flatten}(\boldsymbol{H}^{\mathsf{T}})$$

How do we solve this linear system of equations?

Homogeneous least-squares problem

- Problem of type: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$
- When noise is present, cannot be satisfied exactly
- Equivalent to $\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x}\|_2^2$
- Trivial solution $\boldsymbol{x} = \boldsymbol{0}$ (not useful)
- $\boldsymbol{x}$ can be arbitrarily scaled
  - We impose $||\boldsymbol{x}||_2 = 1$

Homogeneous least-squares solution

Rewrite problem $||\boldsymbol{A}\boldsymbol{x}||_2^2 = \boldsymbol{x}^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x}$;

Assume $\boldsymbol{x}$ is an eigenvector of the square matrix $\boldsymbol{A}^\top \boldsymbol{A}$.

Homogeneous least-squares solution

Which eigenvector?

Eigen-decomposition of $\boldsymbol{A}^\top \boldsymbol{A}$:

$$\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{V} \boldsymbol{\Lambda} \boldsymbol{V}^\top,$$
$$\boldsymbol{v}_i^\top \boldsymbol{A}^\top \boldsymbol{A} \boldsymbol{v}_i = \boldsymbol{v}_i^\top \boldsymbol{V} \boldsymbol{\Lambda} \boldsymbol{V}^\top \boldsymbol{v}_i = \lambda_i \,.$$

So, the minimal solution is the eigenvector with the smallest eigenvalue.

Any other vector will give a larger error.

Least-squares solution with SVD

Using the SVD (`np.linalg.svd`)

$$
\begin{aligned}
\boldsymbol{A}^\top \boldsymbol{A} &= \left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\right)^\top \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \\
&= \boldsymbol{V}\boldsymbol{\Sigma}^\top \boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \\
&= \boldsymbol{V}\underbrace{\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}}_{\Lambda}\boldsymbol{V}^\top
\end{aligned}
$$

$\boldsymbol{V}$ are the eigenvectors of $\boldsymbol{A}^\top\boldsymbol{A}$, and the minimum solution has the smallest singular value.

The solution of $\boldsymbol{Ax} = \mathbf{0}$ where $||\boldsymbol{x}||^2 = 1$ is found using SVD.

$\boldsymbol{x} = \boldsymbol{v}$ where the singular vector $\boldsymbol{v}$ corresponds to the smallest singular value.

Normalization

For numerical stability, it is recommended to give both sets of points mean $\mathbf{0}$ and standard deviation $\mathbf{1}$ before estimating the homography.

Learning objectives

After this lecture you should be able to:

- explain the phenomelogical origin of lens distortion
- compensate for lens distortion in images
- explain the phenomelogical origin of the pinhole camera parameters
- derive the homography matrix
- explain the use of homographies in computer vision
- perform the linear estimation of a homography matrix
- use singular value decomposition (SVD) to solve linear systems

Comment about exercises

- If you need for-loops, you're probably not doing it the easy way.
  - No exercises today need for-loops
- Matrix multiplication lets you project many points at once
- Converting from homogeneous coordinates to regular
  - `p = ph[:-1]/ph[-1]`
- Ask the TAs 🙂

Exercise time!