Vision 3D artificielle
Session 1: Projective geometry, camera matrix, panorama

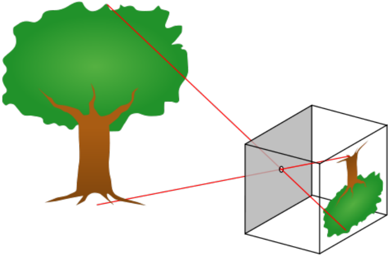
Pascal Monasse
monasse@imagine.enpc.fr

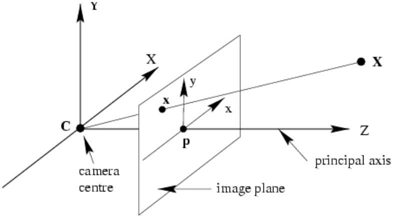
IMAGINE, École des Ponts ParisTech

Contents

Contents

The “pinhole” camera model





Model

Projection (Source: Wikipedia)

The“pinhole” camera (French: sténopé):

- Ideal model with an aperture reduced to a single point.
- No account for blur of out of focus objects, nor for the lens geometric distortion.

Central projection in camera coordinate frame

- Rays from C are the same: $\vec{Cx} = \lambda \vec{CX}$
- In the camera coordinate frame $CXYZ$:

$$\begin{pmatrix} x \\ y \\ f \end{pmatrix} = \lambda \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

- Thus $\lambda = f/Z$ and

$$\begin{pmatrix} x \\ y \end{pmatrix} = f \begin{pmatrix} X/Z \\ Y/Z \end{pmatrix}$$

- In pixel coordinates:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \alpha x + c_x \\ \alpha y + c_y \end{pmatrix} = \begin{pmatrix} (\alpha f)X/Z + c_x \\ (\alpha f)Y/Z + c_y \end{pmatrix}$$

- αf: focal length *in pixels*, (c_x, c_y): position of principal point P in pixels.

Contents

Projective plane

- We identify two points of $\mathbb{R}^3$ on the same ray from the origin through the equivalence relation:

$$\mathcal{R} : \mathbf{x}\mathcal{R}\mathbf{y} \Leftrightarrow \exists\lambda \neq 0 : \mathbf{x} = \lambda\mathbf{y}$$

- Projective plane: $\mathbb{P}^2 = (\mathbb{R}^3 \setminus O)/\mathcal{R}$
- Point $\begin{pmatrix} x & y & z \end{pmatrix} = \begin{pmatrix} x/z & y/z & 1 \end{pmatrix}$ if $z \neq 0$.
- The point $\begin{pmatrix} x/\epsilon & y/\epsilon & 1 \end{pmatrix} = \begin{pmatrix} x & y & \epsilon \end{pmatrix}$ is a point "far away" in the direction of the line of slope y/x. The limit value $\begin{pmatrix} x & y & 0 \end{pmatrix}$ is the infinite point in this direction.
- Given a plane of $\mathbb{R}^3$ through O, of equation $aX + bY + cZ = 0$. It corresponds to a line in $\mathbb{P}^2$ represented in homogeneous coordinates by $\begin{pmatrix} a & b & c \end{pmatrix}$. Its equation is:

$$\begin{pmatrix} a & b & c \end{pmatrix} \begin{pmatrix} X & Y & Z \end{pmatrix}^T = 0.$$

Projective plane

- Line through points $\mathbf{x_1}$ and $\mathbf{x_2}$:

$$\ell = \mathbf{x_1} \times \mathbf{x_2} \text{ since } (\mathbf{x_1} \times \mathbf{x_2})^T \mathbf{x_i} = |\mathbf{x_1} \quad \mathbf{x_2} \quad \mathbf{x_i}| = 0$$

- Intersection of two lines ℓ_1 and ℓ_2:

$$\mathbf{x} = \ell_1 \times \ell_2 \text{ since } \ell_i^T (\ell_1 \times \ell_2) = |\ell_i \quad \ell_1 \quad \ell_2| = 0$$

- Line at infinity:

$$\ell_\infty = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \text{ since } \ell_\infty^T \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = 0$$

- Intersection of two "parallel" lines:

$$\begin{pmatrix} a \\ b \\ c_1 \end{pmatrix} \times \begin{pmatrix} a \\ b \\ c_2 \end{pmatrix} = (c_2 - c_1) \begin{pmatrix} b \\ -a \\ 0 \end{pmatrix} \in \ell_\infty$$

Calibration matrix

- Let us get back to the projection equation:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} fX/Z + c_x \\ fY/Z + c_y \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} fX + c_x Z \\ fY + c_y Z \end{pmatrix}$$

  (replacing αf by f)
- We rewrite:

$$Z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} := \mathbf{x} = \begin{pmatrix} f & & c_x \\ & f & c_y \\ & & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

- The 3D point being expressed in another orthonormal coordinate frame:

$$\mathbf{x} = \begin{pmatrix} f & & c_x \\ & f & c_y \\ & & 1 \end{pmatrix} \begin{pmatrix} R & T \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Calibration matrix

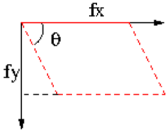
- The (internal) calibration matrix (3×3) is:

$$K = \begin{pmatrix} f & & c_x \\ & f & c_y \\ & & 1 \end{pmatrix}$$

- The projection matrix (3×4) is:

$$P = K \begin{pmatrix} R & T \end{pmatrix}$$

- If pixels are trapezoids, we can generalize K:



$$K = \begin{pmatrix} f_x & s & c_x \\ & f_y & c_y \\ & & 1 \end{pmatrix} \text{(with } s = -f_x \operatorname{cotan} \theta\text{)}$$

Theorem

Let P be a 3×4 matrix whose left 3×3 sub-matrix is invertible. There is a unique decomposition $P = K \begin{pmatrix} R & T \end{pmatrix}$.

Proof: Gram-Schmidt on rows of left sub-matrix of P starting from last row (RQ decomposition), then $T = K^{-1} P_4$.

Contents

Homographies

Let us see what happens when we take two pictures in the following particular cases:

1. Rotation around the optical center (and maybe change of internal parameters).

$$\mathbf{x}' = K'RK^{-1}\mathbf{x} := H\mathbf{x}$$

2. The world is flat. We observe the plane $Z = 0$:

$$\mathbf{x}' = K \begin{pmatrix} R_1 & R_2 & R_3 & T \end{pmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = K \begin{pmatrix} R_1 & R_2 & T \end{pmatrix} \mathbf{x} := H\mathbf{x}$$

In both cases, we deal with a 3×3 invertible matrix H, a homography.

Property: a homography preserves alignment. If $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}$ are aligned, then

$$|H\mathbf{x_1} \quad H\mathbf{x_2} \quad H\mathbf{x_3}| = |H||\mathbf{x_1} \quad \mathbf{x_2} \quad \mathbf{x_3}| = 0$$

Contents

Panorama construction

- We stitch together images by correcting homographies. This assumes that the scene is flat or that we are rotating the camera.
- Homography estimation:

$$\lambda \mathbf{x}' = H\mathbf{x} \Rightarrow \mathbf{x}' \times (H\mathbf{x}) = 0,$$

  which amounts to 2 independent linear equations per correspondence $(\mathbf{x}, \mathbf{x}')$.
- 4 correspondences are enough to estimate H (but more can be used to estimate through mean squares minimization).

Panorama from 14 photos

Algebraic error minimization

- $\mathbf{x_i'} \times (H\mathbf{x_i}) = 0$ is a system of three linear equations in H.
- We gather the unkwown coefficients of H in a vector of 9 rows

$$h = \begin{pmatrix} H_{11} & H_{12} & \dots & H_{33} \end{pmatrix}^T$$

- We write the equations as $A_i h = 0$ with

$$A_i = \begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -y_i'x_i & -y_i'y_i & -y_i' \\ -x_iy_i' & -y_iy_i' & -y_i' & x_i'x_i & x_i'y_i & x_i' & 0 & 0 & 0 \end{pmatrix}$$

- We can discard the third line and stack the different A_i in A.
- h is a vector of the kernel of A (8×9 matrix)
- We can also suppose $H_{3,3} = h_9 = 1$ and solve

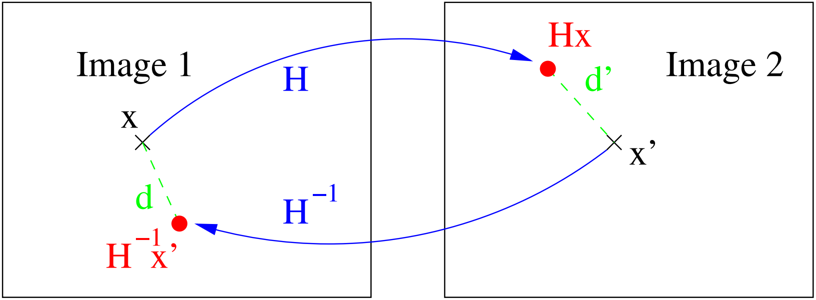
$$A_{:,1:8} h_{1:8} = -A_{:,9}$$

Geometric error

- When we have more than 4 correspondences, we minimize the algebraic error

$$\epsilon = \sum_i \|\mathbf{x_i}' \times (H\mathbf{x_i})\|^2,$$

  but it has no geometric meaning.
- A more significant error is geometric:

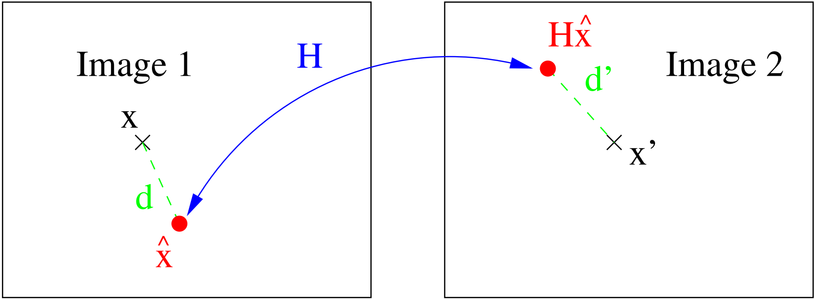


- Either $d'^2 = d(\mathbf{x}', \mathbf{Hx})^2$ (transfer error) or

  $d^2 + d'^2 = d(\mathbf{x}, H^{-1}\mathbf{x}')^2 + d(\mathbf{x}', H\mathbf{x})^2$(Symmetric transfer error)

Gold standard error

- Actually, we can consider $\mathbf{x}$ and $\mathbf{x}'$ as noisy observations of ground truth positions $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}' = H\hat{\mathbf{x}}$.



$$\epsilon(H, \hat{\mathbf{x}}) = d(x, \hat{\mathbf{x}})^2 + d(\mathbf{x}', H\hat{\mathbf{x}})^2$$

- Problem: this has a lot of parameters: $H, \{\hat{\mathbf{x}}_\mathbf{i}\}_{i=1\ldots n}$

Sampson error

- A method that linearizes the dependency on $\hat{\mathbf{x}}$ in the gold standard error so as to eliminate these unknowns.

$$0 = \epsilon(H, \hat{\mathbf{x}}) = \epsilon(H, \mathbf{x}) + J(\hat{\mathbf{x}} - \mathbf{x}) \text{ with } J = \frac{\partial \epsilon}{\partial \mathbf{x}}(H, \mathbf{x})$$

- Find $\hat{\mathbf{x}}$ minimizing $\|\mathbf{x} - \hat{\mathbf{x}}\|^2$ subject to $J(\mathbf{x} - \hat{\mathbf{x}}) = \epsilon$
- Solution: $\mathbf{x} - \hat{\mathbf{x}} = J^T (JJ^T)^{-1} \epsilon$ and thus:

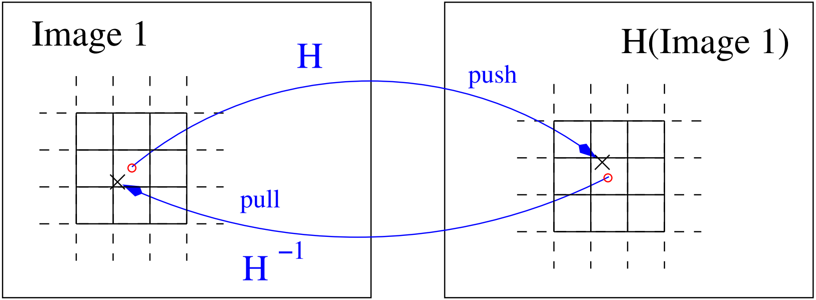
$$\|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \epsilon^T (JJ^T)^{-1} \epsilon \tag{1}$$

- Here, $\epsilon_i = A_i h = \mathbf{x_i'} \times (H\mathbf{x_i})$ is a 3-vector.
- For each i, there are 4 variables $(\mathbf{x_i}, \mathbf{x_i'})$, so J is 3×4.
- This is almost the algebraic error $\epsilon^T \epsilon$ but with adapted scalar product.
- The resolution, through iterative method, must be initialized with the algebraic minimization.

Applying homography to image

Two methods:

1. push pixels to transformed image and round to the nearest pixel center.
2. pull pixels from original image by interpolation.



Contents

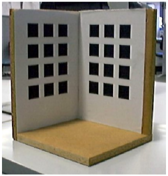
Camera calibration by resection

[R.Y. Tsai, *An efficient and accurate camera calibration technique for 3D machine vision*, CVPR'86] We estimate the camera internal parameters from a known rig, composed of 3D points whose coordinates are known.

- We have points $\mathbf{X_i}$ and their projection $\mathbf{x_i}$ in an image.
- In homogeneous coordinates: $\mathbf{x_i} = P\mathbf{X_i}$ or the 3 equations (but only 2 of them are independent)

$$\mathbf{x_i} \times (P\mathbf{X_i}) = 0$$

- Linear system in unknown P. There are 12 parameters in P, we need 6 points in general (actually only 5.5).
- Decomposition of P allows finding K.



Restriction: The 6 points cannot be on a plane, otherwise we have a degenerate situation; in that case, 4 points define the homography and the two extra points yield no additional constraint.

Calibration with planar rig

[Z. Zhang *A flexible new technique for camera calibration* 2000]

- Problem: One picture is not enough to find K.
- Solution: Several snapshots are used.
- For each one, we determine the homography H between the rig and the image.
- The homography being computed with an arbitrary multiplicative factor, we write

$$\lambda H = K \begin{pmatrix} R_1 & R_2 & T \end{pmatrix}$$

- We rewrite:

$$\lambda K^{-1} H = \lambda \begin{pmatrix} K^{-1} H_1 & K^{-1} H_2 & K^{-1} H_3 \end{pmatrix} = \begin{pmatrix} R_1 & R_2 & T \end{pmatrix}$$

- 2 equations expressing orthonormality of R_1 and R_2:

$$H_1^T (K^{-T} K^{-1}) H_1 = H_2^T (K^{-T} K^{-1}) H_2$$
$$H_1^T (K^{-T} K^{-1}) H_2 = 0$$

- With 3 views, we have 6 equations for the 5 parameters of $K^{-T} K^{-1}$; then Cholesky decomposition.

The problem of geometric distortion

- At small or moderate focal length, we cannot ignore the geometric distortion due to lens curvature, especially away from image center.
- This is observable in the non-straightness of certain lines:

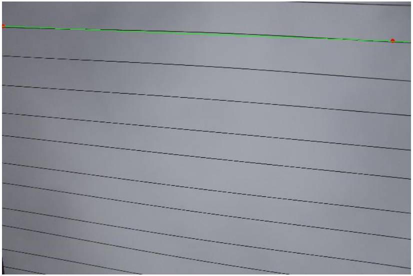


Photo: 5600×3700 pixels

Deviation of 30 pixels

- The classical model of distortion is radial polynomial:

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} - \begin{pmatrix} d_x \\ d_y \end{pmatrix} = (1 + a_1 r^2 + a_2 r^4 + \dots) \begin{pmatrix} x - d_x \\ y - d_y \end{pmatrix}$$

Estimation of geometric distortion

- If we integrate distortion coefficients as unknowns, there is no more closed formula estimating K.
- We have a non-linear minimization problem, which can be solved by an iterative method.
- To initialize the minimization, we assume no distortion ($a_1 = a_2 = 0$) and estimate K with the previous linear procedure.

Contents

Linear least squares problem

- For example, when we have more than 4 point correspondences in homography estimation:

$$A_{m\times 8}h = B_m \quad m \geq 8$$

- In the case of an overdetermined linear system, we minimize

$$\epsilon(\mathbf{X}) = \|A\mathbf{X} - B\|^2 = \|f(\mathbf{X})\|^2$$

- The gradient of ϵ can be easily computed:

$$\nabla\epsilon(\mathbf{X}) = 2(A^T A\mathbf{X} - A^T B)$$

- The solution is obtained by equating the gradient to 0:

$$\mathbf{X} = (A^T A)^{-1} A^T B$$

- Remark 1: this is correct only if $A^T A$ is invertible, that is A has full rank.
- Remark 2: if A is square, it is the standard solution $\mathbf{X} = A^{-1}B$
- Remark 3: $A^{(-1)} = (A^T A)^{-1} A^T$ is called the pseudo-inverse of A, because $A^{(-1)}A = I_n$.

Non-linear least squares problem

- We would like to solve as best we can $f(\mathbf{X}) = 0$ with f non-linear. We thus minimize

$$\epsilon(\mathbf{X}) = \|f(\mathbf{X})\|^2$$

- Let us compute the gradient of ϵ:

$$\nabla\epsilon(\mathbf{X}) = 2J^T f(\mathbf{X}) \text{ with } J_{ij} = \frac{\partial f_i}{\partial x_j}$$

- Gradient descent: we iterate until convergence

$$\triangle\mathbf{X} = -\alpha J^T f(\mathbf{X}),\ \alpha > 0$$

- When we are close to the minimum, a faster convergence is obtained by Newton's method:

$$\epsilon(\mathbf{X_0}) \sim \epsilon(\mathbf{X}) + \nabla\epsilon(\mathbf{X})^T(\triangle\mathbf{X}) + (\triangle\mathbf{X})^T(\nabla^2\epsilon)(\triangle\mathbf{X})$$

and minimum is for $\triangle\mathbf{X} = -(\nabla^2\epsilon)^{-1}\nabla\epsilon$

Levenberg-Marquardt algorithm

- This is a mix of gradient descent and quasi-Newton method (*quasi* since we do not compute explictly the Hessian matrix, but approximate it).
- The gradient of ϵ is

$$\nabla \epsilon(\mathbf{X}) = 2J^T f(\mathbf{X})$$

  so the Hessian matrix of ϵ is composed of sums of two terms:
  1. Product of first derivatives of f.
  2. Product of f and second derivatives of f.
- The idea is to ignore the second terms, as they should be small when we are close to the minimum ($f \sim 0$). The Hessian is thus approximated by

$$H = 2J^T J$$

- Levenberg-Marquardt iteration:

$$\triangle \mathbf{X} = -(J^T J + \lambda I)^{-1} J^T f(\mathbf{X}), \lambda > 0$$

Levenberg-Marquardt algorithm

- Principle: gradient descent when we are far from the solution (λ large) and Newton's step when we are close (λ small).

1. Start from initial $\mathbf{X}$ and $\lambda = 10^{-3}$.
2. Compute

$$\triangle\mathbf{X} = -(J^T J + \lambda I)^{-1} J^T f(\mathbf{X}), \lambda > 0$$

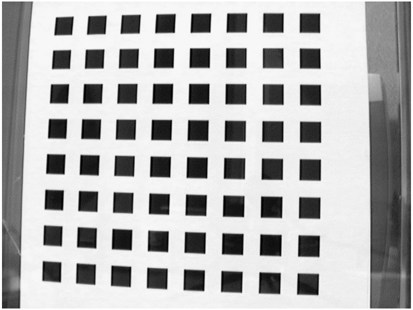
3. Compare $\epsilon(\mathbf{X} + \triangle\mathbf{X})$ and $\epsilon(\mathbf{X})$:
   - 3a If $\epsilon(\mathbf{X} + \triangle\mathbf{X}) \sim \epsilon(\mathbf{X})$, finish.
   - 3b If $\epsilon(\mathbf{X} + \triangle\mathbf{X}) < \epsilon(\mathbf{X})$,

$$\mathbf{X} \leftarrow \mathbf{X} + \triangle\mathbf{X} \quad \lambda \leftarrow \lambda/10$$

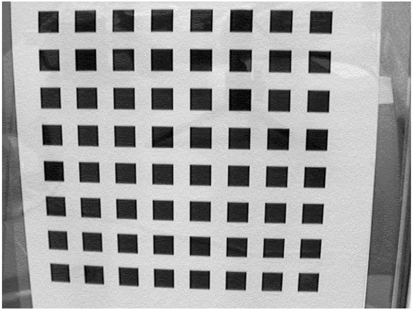
   - 3c If $\epsilon(\mathbf{X} + \triangle\mathbf{X}) > \epsilon(\mathbf{X})$, $\lambda \leftarrow 10\lambda$
4. Go to step 2.

Example of distortion correction

Results of Zhang:



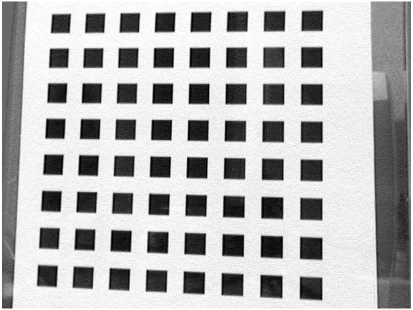
Snapshot 1



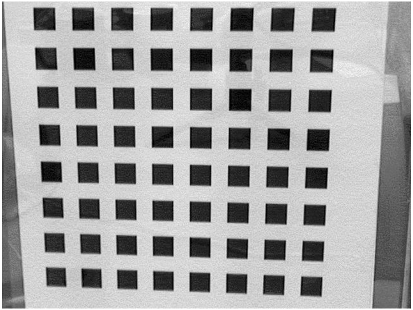
Snapshot 2

Example of distortion correction

Results of Zhang:



Corrected image 1



Corrected image 2

Contents

Conclusion

- Camera matrix K (3×3) depends only on internal parameters of the camera.
- Projection matrix P (3×4) depends on K and position/orientation.
- Homogeneous coordinates are convenient as they linearize the equations.
- A homography between two images arises when the observed scene is flat or the principal point is fixed.
- 4 or more correspondences are enough to estimate a homography (in general)

Contents

Practical session: panorama construction

Objective: the user clicks 4 or more corresponding points in left and right images. After a right button click, the program computes the homography and shows the resulting panorama in a new window.

- Install Imagine++ (`http://imagine.enpc.fr/~monasse/Imagine++/`) on your machine.
- Let the user click the matching points.
- Build the linear system to solve $Ax = b$; use `linSolve` (in `LinAlg`) to find the solution x.
- Compute the bounding box of the panorama.
- Stitch the images: on overlapping area, take the average of colors at corresponding pixels in both images.