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

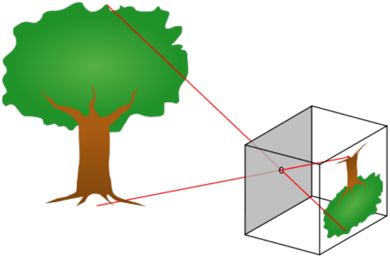
Pascal Monasse (pascal.monasse@enpc.fr)

IMAGINE/LIGM, École nationale des ponts et chaussées

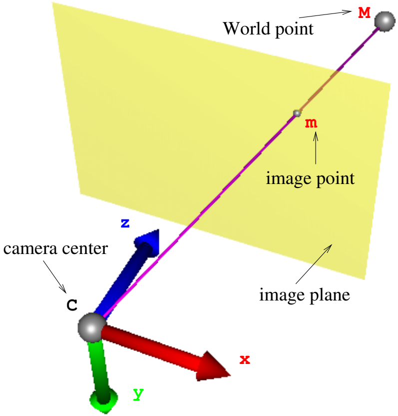
Contents

Contents

The "pinhole" camera model



Projection (Source: Wikipedia)



Model

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: $\vec{Cm} = \lambda \vec{CM}$
- 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:

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

- 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, its equation is $aX + bY + cZ = 0$. It corresponds to a line in $\mathbb{P}^2$ represented in homogeneous coordinates by $\begin{pmatrix} a & b & c \end{pmatrix}$. Its equation is:

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

Projective plane

- Line through points x_1 and x_2:

$$\ell = \mathrm{x}_1 \times \mathrm{x}_2 \text{ since } (\mathrm{x}_1 \times \mathrm{x}_2)^\top \mathrm{x}_\mathrm{i} = |\mathrm{x}_1 \quad \mathrm{x}_2 \quad \mathrm{x}_\mathrm{i}| = 0$$

- Intersection of two lines ℓ_1 and ℓ_2:

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

- Line at infinity:

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

- 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:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f\,X/Z + c_x \\ f\,Y/Z + c_y \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} f\,X + c_x Z \\ f\,Y + c_y Z \end{pmatrix}$$

  (replacing αf by f)
- We rewrite:

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

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

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

Calibration matrix

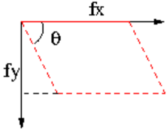
- The (internal) calibration matrix (3×3) is:

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

- 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 $T = K^{-1} P_4$.

Projective plane (perspective effect)

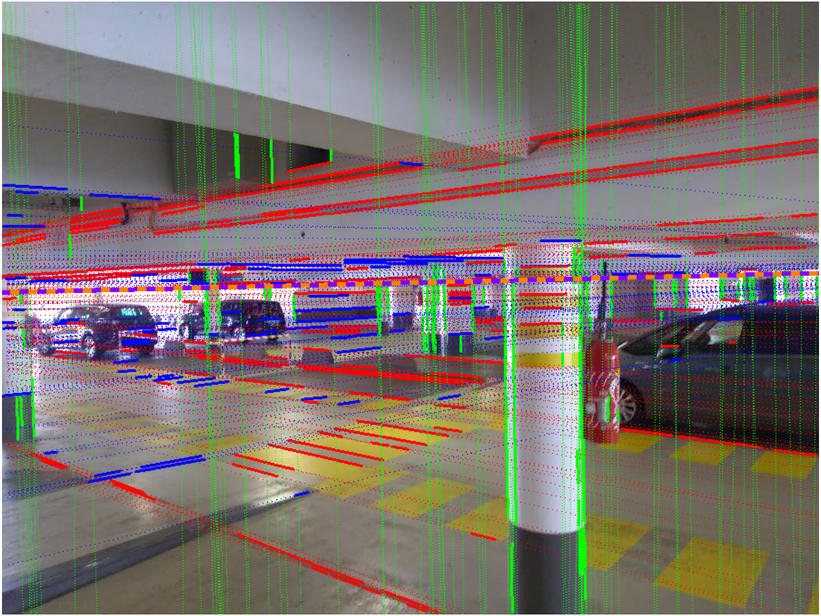
- Lines parallel in space project to a line bundle (set of lines parallel or concurrent). Let d be a fixed direction vector:

$$\begin{aligned} K(\mathsf{X} + \lambda \mathsf{d}) &= K\mathsf{X} + \lambda K\mathsf{d} \\ \ell_{\mathsf{X}} &= (K\mathsf{X}) \times (K\mathsf{d}) \\ \forall \mathsf{X}, \ell_{\mathsf{X}}^\top \mathsf{v} &= 0 \quad \text{for } \mathsf{v} := K\mathsf{d} \end{aligned}$$

- v is the vanishing point of lines of direction d.
- If $\mathsf{v}_1 = K\mathsf{d}_1$ and $\mathsf{v}_2 = K\mathsf{d}_2$ are vp of "horizontal" lines, another set of horizontal lines has direction $\alpha \mathsf{d}_1 + \beta \mathsf{d}_2$, hence its vp $\alpha \mathsf{v}_1 + \beta \mathsf{v}_2$, which belongs to line $\mathsf{v}_1 \times \mathsf{v}_2$, the "horizon".

Projective plane (perspective effect)

Projective plane (perspective effect)

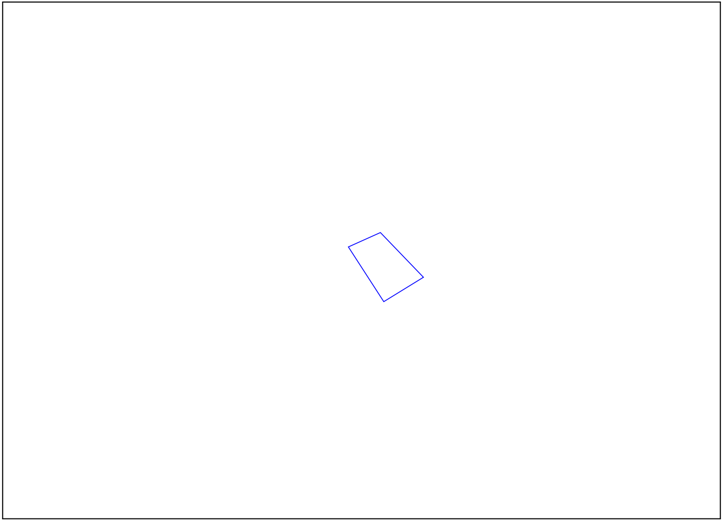


Projective plane (perspective effect)

How to draw a tiled floor like Renaissance painters:

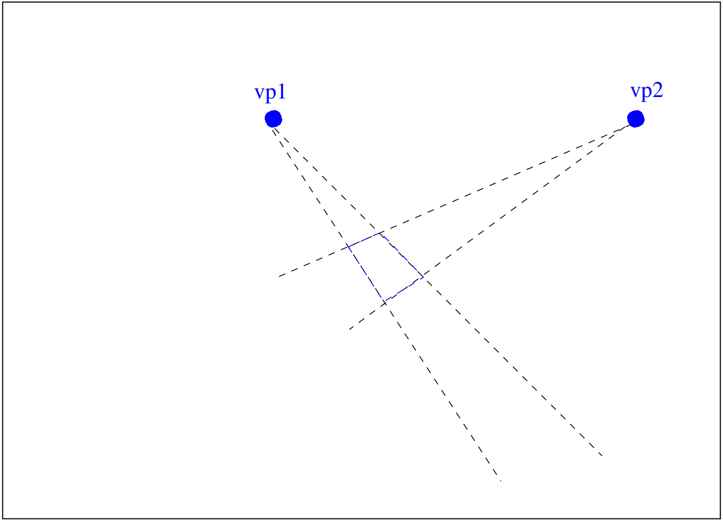
Bartholomeus van Bassen
The king and queen of Bohemia dining in public (1634)

Projective plane (perspective effect)



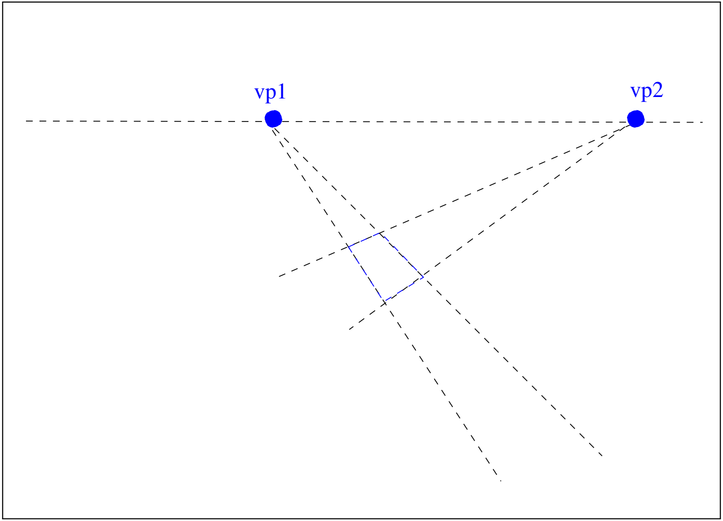
Initial tile

Projective plane (perspective effect)



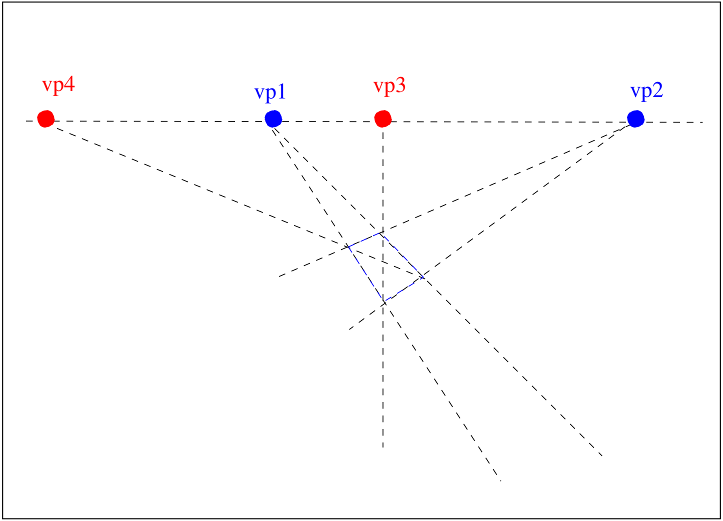
Find vp of opposite sides

Projective plane (perspective effect)



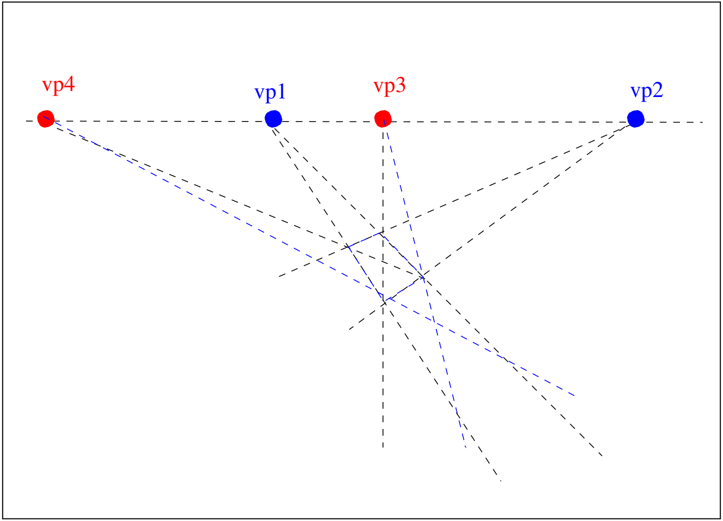
Find horizon by joining vp

Projective plane (perspective effect)



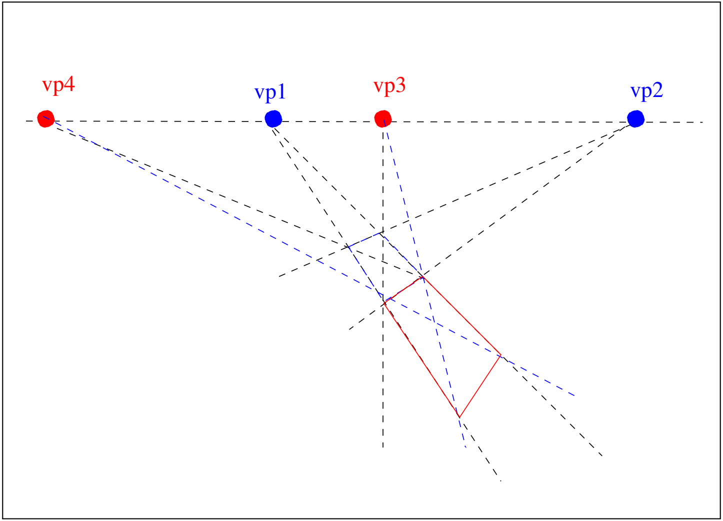
Find vp of diagonals (intersect with horizon)

Projective plane (perspective effect)



Take one side, intersect lines joining endpoints and vp3 and vp4 with adjacent sides

Projective plane (perspective effect)



We get the two diagonals of the adjacent tile

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).

$$x' = K'RK^{-1}x := Hx$$

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

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

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

Property: a homography preserves alignment. If x_1, x_2, x_3 are aligned, then

$$|Hx_1 \quad Hx_2 \quad Hx_3| = |H||x_1 \quad x_2 \quad x_3| = 0$$

Homographies

Type	Matrix	Invariants
Rigid (Rot.+Trans.)	$H = \begin{pmatrix} c & -s & t_x \\ s & c & t_y \\ 0 & 0 & 1 \end{pmatrix}$ $(c^2 + s^2 = 1)$	distances
Similarity	$H = \begin{pmatrix} c & -s & t_x \\ s & c & t_y \\ 0 & 0 & 1 \end{pmatrix}$ $(c^2 + s^2 \neq 0)$	|angles|, ratio of distances
Affine	$H = \begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix}$ $(ad - bc \neq 0)$	parallelism
Homography	H invertible $(\lvert H \rvert \neq 0)$	cross-ratio of 4 aligned points

Given 4 aligned points A, B, C, D, their cross-ratio is:

$$(A, B; C, D) = \frac{AC}{BC} : \frac{AD}{BD}$$

Homographies: estimation from point correspondences

Theorem Let $e_1, \cdots, e_{d+1}, f_1, \cdots, f_{d+1} \in \mathbb{R}^d$ such that any d vectors e_i (resp. f_i) are linearly independent. Then there are (up to scale) a unique isomorphism H and a unique set of scalars $\lambda_i \neq 0$ so that $\forall i, He_i = \lambda_i f_i$.

1. Analysis: writing $e_{d+1} = \sum_{i=1}^d \mu_i e_i$ and $f_{d+1} = \sum_{i=1}^d \nu_i f_i$,

   $$\sum_{i=1}^{d} \nu_i \lambda_{d+1} f_i = \lambda_{d+1} f_{d+1} = He_{d+1} = \sum_{i=1}^{d} \mu_i \lambda_i f_i$$

   so that $\forall i = 1, \ldots, d : \mu_i \lambda_i = \nu_i \lambda_{d+1}$.
2. $\forall i, \mu_i \neq 0$ and $\nu_i \neq 0$, since $\mu_i = 0 \Rightarrow \{e_j\}_{j \neq i}$ are linearly dependent.
3. Therefore we get $\lambda_i = \frac{\nu_i}{\mu_i} \lambda_{d+1}$.
4. Synthesis: fix $\lambda_{d+1} = 1$ and $\forall i : \lambda_i = \frac{\nu_i}{\mu_i} \neq 0$.
5. There is a unique H mapping basis $\{e_i\}_{i=1,\ldots,d}$ to basis $\{\lambda_i f_i\}_{i=1,\ldots,d}$

Application: given $n+2$ pairs $(\mathrm{x_i}, \mathrm{x_i'}) \in \mathbb{P}^n$, there is a unique homography mapping $\mathrm{x_i}$ to $\mathrm{x_i'}$.

Homography or not homography?

Look around depth discontinuities whether occlusions are present.

OK, homography

Homography or not homography?

Look around depth discontinuities whether occlusions are present.

Not OK, no homography

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:

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

  which amounts to 2 independent linear equations per correspondence $(\mathrm{x}, \mathrm{x}')$.
- 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

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

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

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

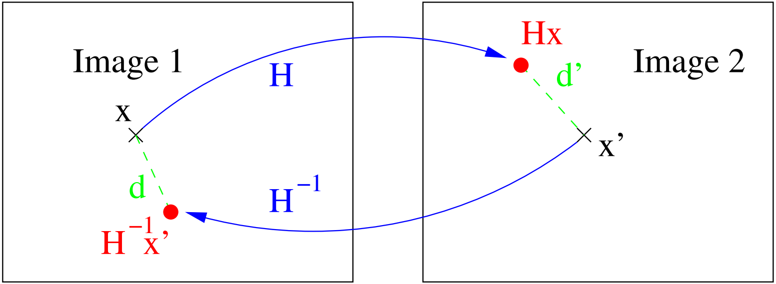
$$A_i = \begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i' x_i & -x_i' y_i & -x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -y_i' x_i & -y_i' y_i & -y_i' \\ -x_i y_i' & -y_i y_i' & -y_i' & x_i' x_i & x_i' y_i & x_i' & 0 & 0 & 0 \end{pmatrix}$$

- 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

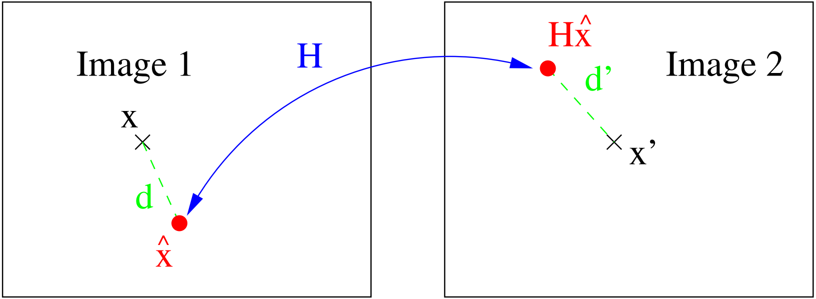
- When we have more than 4 correspondences, we minimize the algebraic error
$$\epsilon = \sum_i \|\mathsf{x}_i' \times (H\mathsf{x}_i)\|^2,$$
but it has no geometric meaning.
- A more significant error is geometric:



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

Gold standard error/Maximum likelihood estimator

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



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

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

Sampson error

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

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

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

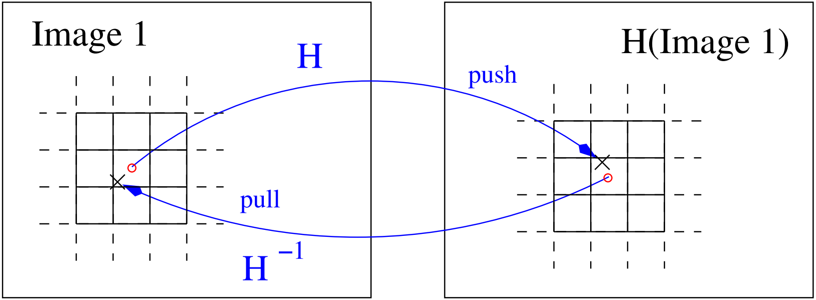
$$\|\mathsf{x} - \hat{\mathsf{x}}\|^2 = \epsilon^\top (JJ^\top)^{-1} \epsilon \tag{1}$$

- Here, $\epsilon_i = A_i h = \mathsf{x}_\mathsf{i}' \times (H\mathsf{x}_\mathsf{i})$ is a 3-vector.
- For each i, there are 4 variables $(\mathsf{x}_\mathsf{i}, \mathsf{x}_\mathsf{i}')$, so J is 3×4.
- This is almost the algebraic error $\epsilon^\top \epsilon$ 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

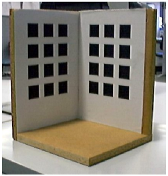
Camera calibration by resection

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

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

- 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:

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

- With 3 views, we have 6 equations for the 5 parameters of $K^{-\top} K^{-1}$; 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:

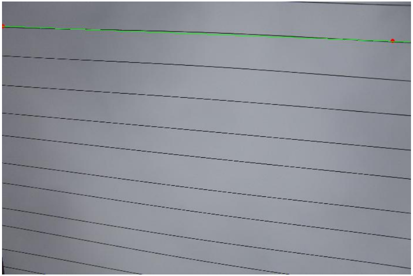


Photo: 5600×3700 pixels

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

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

- The gradient of ϵ can be easily computed:

$$\nabla\epsilon(\mathsf{X}) = 2(A^\top A\mathsf{X} - A^\top B)$$

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

$$\mathsf{X} = (A^\top A)^{-1}A^\top B$$

- Remark 1: this is correct only if $A^\top A$ is invertible, that is A has full rank.
- Remark 2: if A is square, it is the standard solution $\mathsf{X} = A^{-1}B$
- Remark 3: $A^{(-1)} = (A^\top A)^{-1}A^\top$ 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 $f(\mathsf{X}) = 0$ with f non-linear. We thus minimize

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

- Let us compute the gradient of ϵ:

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

- Gradient descent: we iterate until convergence

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

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

$$\epsilon(\mathsf{X}_0) \sim \epsilon(\mathsf{X}) + \nabla\epsilon(\mathsf{X})^\top(\triangle\mathsf{X}) + (\triangle\mathsf{X})^\top(\nabla^2\epsilon)(\triangle\mathsf{X})/2$$

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

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

$$\nabla\epsilon(\mathsf{X}) = 2J^\top f(\mathsf{X})$$

  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

$$H = 2J^\top J$$

- Levenberg-Marquardt iteration:

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

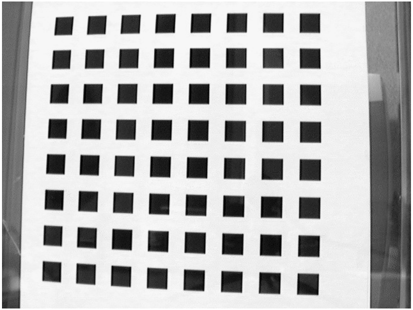
Levenberg-Marquardt algorithm

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

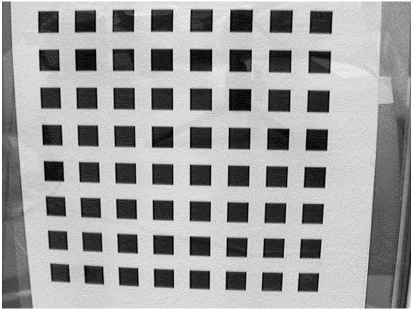
1. Start from initial X and $\lambda = 10^{-3}$.
2. Compute
$$\triangle X = -(J^\top J + \lambda I)^{-1} J^\top f(X), \lambda > 0$$
3. Compare $\epsilon(X + \triangle X)$ and $\epsilon(X)$:
   3a If $\epsilon(X + \triangle X) \sim \epsilon(X)$, finish.
   3b If $\epsilon(X + \triangle X) < \epsilon(X)$,
$$X \leftarrow X + \triangle X \quad \lambda \leftarrow \lambda/10$$
   3c If $\epsilon(X + \triangle X) > \epsilon(X)$, $\lambda \leftarrow 10\lambda$
4. Go to step 2.

Example of distortion correction

Results of Zhang:



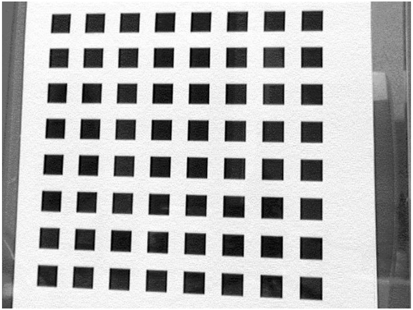
Snapshot 1



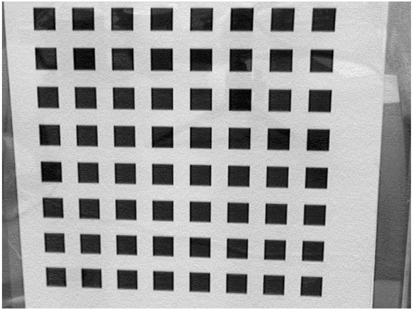
Snapshot 2

Example of distortion correction

Results of Zhang:



Corrected image 1



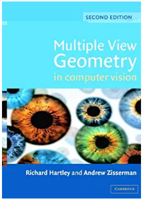
Corrected image 2

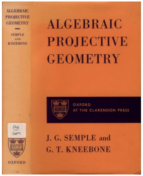
Contents

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)

References





Hartley & Zisserman (2004)

- Chapter 2: Projective Geometry and Transformations of 2D
- Chapter 4: Estimation–2D Projective Transformations
- Chapter 6: Camera Models
- Chapter 7: Computation of the Camera Matrix P

Semple & Kneebone (1962)

- Chapter IV: Projective Geometry of Two Dimensions
- Appendix: Two Basic Algebraic Theorems

Appendix: QR decomposition

Let A be an $n \times n$ matrix. Then we can decompose $A = QR$ with $Q \in O(n)$ ($Q^\top Q = I$) and R upper triangular with $\min \operatorname{diag}(R) \geq 0$. Moreover

$$A \text{ invertible} \Leftrightarrow \min \operatorname{diag}(R) > 0 \Leftrightarrow (Q, R) \text{ unique.}$$

Variants:

- $A = LQ$ with L lower triangular: write $A^\top = Q_1 R_1$, then $L = R_1^\top$, $Q = Q_1^\top$.
- $A = QL$: write $JAJ = Q_1 R_1$ with $J_{ij} = \delta_{i+j=n+1}$, then $Q = JQ_1$, $L = R_1 J$. (Notice $J = J^\top = J^{-1} \in O(n)$)
- $A = RQ$: write $A^\top = Q_1 L_1$, then $R = L_1^\top$ and $Q = Q_1^\top$.

Appendix: Invariance of cross-ratio through homography

- 1D homography: $x \to h(x) = \frac{ax+b}{cx+d}$ $(ad - bc \neq 0)$.
- Cross ratio: $(x_1, x_2 : x_3, x_4) = \frac{x_1 - x_3}{x_1 - x_4} / \frac{x_2 - x_3}{x_2 - x_4}$.
- Write:

$$h(x) = \begin{cases} \frac{b-ad/c}{cx+d} + \frac{a}{c} = \mathcal{A}_{b-ad/c,a/c} \circ \mathcal{I} \circ \mathcal{A}_{c,d}(x) & (c \neq 0) \\ \mathcal{A}_{a/d,b/d}(x) & (c = 0 \Rightarrow d \neq 0) \end{cases}$$

with

$$\mathcal{A}_{a,b}(x) = ax + b, \qquad \mathcal{I}(x) = 1/x$$

- Check $(\mathcal{A}(x_1), \mathcal{A}(x_2) : \mathcal{A}(x_3), \mathcal{A}(x_4)) = (x_1, x_2 : x_3, x_4)$.
- Compute:

$$\begin{aligned}(\mathcal{I}(x_1), \mathcal{I}(x_2) : \mathcal{I}(x_3), \mathcal{I}(x_4)) &= \frac{(x_3 - x_1)/(x_1 x_3)}{(x_4 - x_1)/(x_1 x_4)} / \frac{(x_3 - x_2)/(x_2 x_3)}{(x_4 - x_2)/(x_2 x_4)} \\ &= \frac{x_4(x_1 - x_3)}{x_3(x_1 - x_4)} / \frac{x_4(x_2 - x_3)}{x_3(x_2 - x_4)} \\ &= (x_1, x_2 : x_3, x_4)\end{aligned}$$

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 $Ax = b$ and find 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.

Useful Imagine++ types/functions: `Matrix`, `Vector`, `Image`, `anyGetMouse`, `linSolve`