Vision 3D artificielle
Session 2: Internal calibration, geometric distortion correction, resection

Pascal Monasse
monasse@imagine.enpc.fr

IMAGINE, École des Ponts ParisTech

http://imagine.enpc.fr/~monasse/Stereo/

Contents

Contents

Reminder on camera matrix K

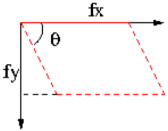
- The (internal) calibration matrix (3×3) is:

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

- The projection matrix (3×4) is:

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

- If pixels are parallelograms, 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 $T = K^{-1} P_4$.

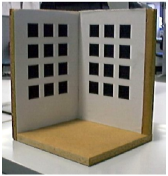
Camera calibration by resection

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

$$x_i \times (PX_i) = 0$$

- 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:

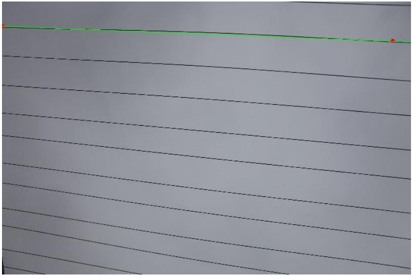


Photo: 5600×3700 pixels

Deviation of 30 pixels

- The classical model of distortion is radial polynomial:

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

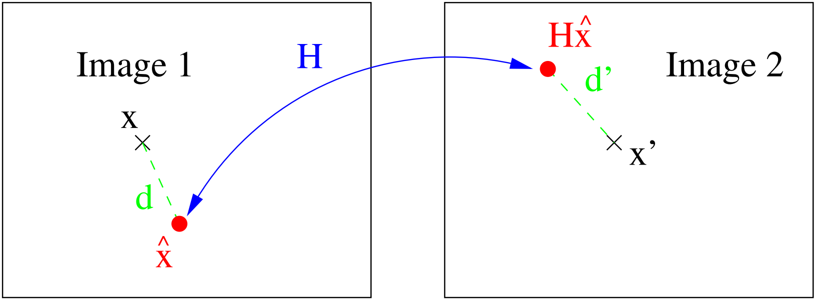
Estimation of geometric distortion

- 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

Gold standard error in homography estimation

- We 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: $H, \{\hat{x}_i\}_{i=1\ldots n}$
- The minimization is heavy in complexity and memory.

Sampson error

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

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

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

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

- Here, $\epsilon_i = A_i h = x_i' \times (Hx_i)$ is a 3-vector.
- For each i, there are 4 variables (x_i, 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.

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

$$\epsilon(X) = \|AX - B\|^2 = \|f(X)\|^2$$

- The gradient of ϵ can be easily computed:

$$\nabla\epsilon(X) = 2(A^T AX - A^T B)$$

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

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

- 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 $X = A^{-1}B$
- 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 $f(X) = 0$ with f non-linear. We thus minimize

$$\epsilon(X) = \|f(X)\|^2$$

- Let us compute the gradient of ϵ:

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

- Gradient descent: we iterate until convergence

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

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

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

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

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

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

  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:

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

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

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

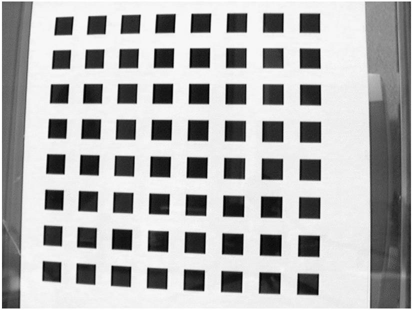
3. Compare $\epsilon(X + \triangle X)$ and $\epsilon(X)$:
   3a If $\epsilon(X + \triangle X) \sim \epsilon(X)$, finish.
   3b If $\epsilon(X + \triangle X) < \epsilon(X)$,

$$X \leftarrow X + \triangle X \quad \lambda \leftarrow \lambda/10$$

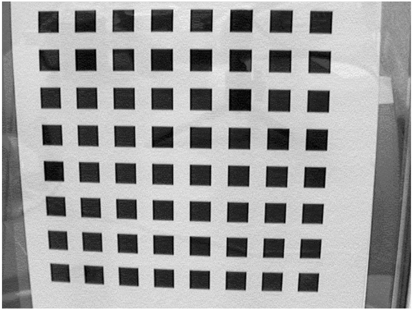
   3c If $\epsilon(X + \triangle X) > \epsilon(X)$, $\lambda \leftarrow 10\lambda$
4. Go to step 2.

Example of distortion correction

Results of Zhang:



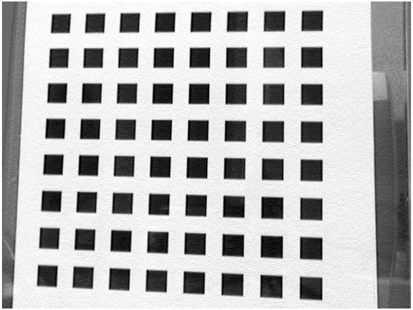
Snapshot 1



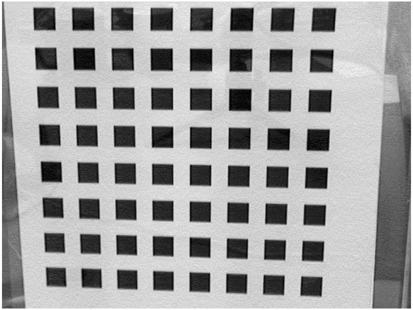
Snapshot 2

Example of distortion correction

Results of Zhang:



Corrected image 1



Corrected image 2

Conclusion

- Calibration with a 3D rig is constraining, though the algorithm is simple (resection).
- Calibration with a planar pattern is easier to implement.
- With distortion correction, we have a non-linear least squares problem, which can be initialized with a linear minimization.
- The method of choice for non-linear least squares is Levenberg-Marquardt.
- Gold standard error can be well approximated with Sampson error, at a fraction of the complexity.

Contents

Practical session: camera calibration

Objective: Implement Zhang's calibration (without distortion).
From a set of photographs of a planar pattern, recover K.

- Get initial program from the website.
- The points of the model and in the images are in files `model.txt` and `data?.txt`
- Fill the function `computeK`:
  1. Build the 10×5 linear system satisfied by the coefficients of $K^{-T}K^{-1}$, supposing its (3,3) entry is 1.
  2. Find the result by Cholesky decomposition.
- For comparison, I get:

$$K = \begin{pmatrix} 871.024 & 0.153579 & 300.682 \\ 0 & 870.678 & 220.872 \\ 0 & 0 & 1 \end{pmatrix}$$

- Zhang gets from the same data (but with distortion correction):

$$K = \begin{pmatrix} 832.5 & 0.204494 & 303.959 \\ 0 & 832.53 & 206.585 \\ 0 & 0 & 1 \end{pmatrix}$$

- You can see the distortion by looking at the snapshots.