# Computer Vision: 7. Camera Models (b)

I-Chen Lin, Assistant Professor
Dept. of CS, National Chiao Tung University

# Objective

- Least square methods
- Geometric camera calibration
  - Linear and non-linear methods

**Textbook:**
• David A. Forsyth and Jean Ponce, Computer Vision: A Modern Approach, Prentice Hall, New Jersey, 2003.

**Plenty of slides are modified from the reference lecture notes or project pages:**
• Prof. T. Darrell, Computer Vision and Applications, MIT.
• Prof. J. Rehg, Computer Vision, Georgia Inst. of Tech.
• Prof. D.A. Forsyth, Computer Vision, UIUC.

# Linear least-squares methods

- *P* linear equations in *q* unknowns:
- *Ux = y*

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1q} \\ u_{21} & u_{22} & \cdots & u_{2q} \\ \cdots & \cdots & \cdots & \cdots \\ u_{p1} & u_{p2} & \cdots & u_{pq} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_q \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_p \end{bmatrix}$$

- When p < q: a (q-p) dimensional vector space
- When p = q: a unique solution
- When p > q: overconstrained system

# Normal equations and pseudo-inverse

- min $E = |Ux-y|^2 = e^Te$, where e = $Ux-y$.
- The minimum $E$ occurs when the derivatives are zeros.
- We define vector $c_j$ = the j^th^ column of $U$

$$\frac{\partial e}{\partial x_i} = \partial\left[\begin{pmatrix} c_1 & \cdots & c_q \end{pmatrix}\begin{pmatrix} x_1 \\ \cdots \\ x_q \end{pmatrix} - y\right] \Big/ \partial x_i = c_i \qquad \frac{\partial E}{\partial x_i} = 2\frac{\partial e}{\partial x_i}\cdot e = 2c_i^T(Ux-y) = 0$$

$$0 = \begin{pmatrix} c_1^T \\ \cdots \\ c_q^T \end{pmatrix}(Ux-y) = U^T(Ux-y) \Leftrightarrow U^TUx = U^Ty$$

$$x = \left(U^TU\right)^{-1}U^Ty$$

*Numerical issue: QR or SVD methods are more reliable*

# min $Ux=0$, subject to $|x|=1$

- Assume $y=0$, $E=|Ux|^2 = x^TU^TUx$.
- $U^TU$ is symmetric positive semidefinite :
  - $U^TU$'s eigenvalues $0 \leq \lambda_1 \leq \ldots \leq \lambda_q$.
  - $U^TU$ can be decomposed as $QDQ^{-1}$, where Q and D consist of eigenvectors and eigenvalue respectively.

- Unit vector $x$ can be represented in terms of eigen vector $e_i$:
  - $x = \mu_1 e_1 + \mu_2 e_2 + \ldots + \mu_q e_q$ and $\mu_1^2 + \ldots. + \mu_q^2 = 1$

- $E(x) - E(e_1) = x^TU^TUx - e_1^TU^TUe_1 = \lambda_1\mu_1^2 + \ldots. + \lambda_q\mu_q^2 - \lambda_1$

$$\geq \lambda_1(\mu_1^2 + \ldots. + \mu_q^2 - 1) = 0$$

The $x$ that minimize $E$ is eigenvector $e_1$ of $U^TU$

# Nonlinear least-squares methods

- *P* general equations in *q* unknowns:

$$f_1(x_1, x_2, \cdots, x_q) = 0$$
$$f_2(x_1, x_2, \cdots, x_q) = 0$$
$$\cdots$$
$$f_p(x_1, x_2, \cdots, x_q) = 0$$

- The error function E(*x*) = $|f(x)|^2 = \sum ( f_i(x) )^2$
- Taylor expansion of $f_i$ is

$$f_i(x + \delta x) = f_i(x) + \delta x_1 \frac{\partial f_i}{\partial x_1}(x) + \cdots + \delta x_q \frac{\partial f_i}{\partial x_q}(x) + O\left((\delta x)^2\right)$$

$$\approx f_i(x) + \nabla f_i(x) \cdot \delta x$$

# Nonlinear least-squares methods (cont.)

$$f(x+\delta x) \approx f(x) + \Im_{f(x)}\delta x$$

$$\Im_{f(x)} = \begin{pmatrix} \nabla f_1^T(x) \\ \cdots \\ \nabla f_p^T(x) \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_q}(x) \\ \cdots & \cdots & \cdots \\ \frac{\partial f_p}{\partial x_1}(x) & \cdots & \frac{\partial f_p}{\partial x_q}(x) \end{pmatrix}$$

*Jacobian of f*

# Newton's method

- For $p=q$ (square system), an iterative algorithm.

- Compute perturbation $\delta x$ such that $f(x + \delta x) \approx 0$:

$$\Im_{f(x)} \delta x = -f(x)$$

# Overconstrained system (p>q)

- Gaussian-Newton method, similar to the pseudo-inverse:

$$\Im_{f(x)}{}^{T}\Im_{f(x)}\delta x = -\Im_{f(x)}{}^{T} f(x)$$

- Levenberg-Marquardt method, to avoid degenerate pseudo-inverse of the Jacobian matrix $J_f$.

$$(\Im_{f(x)}{}^{T}\Im_{f(x)} + \mu I)\delta x = -\Im_{f(x)}{}^{T} f(x)$$

# Camera calibration (linear approach)

- Evaluating the projection matrix *M* and camera parameters with known 3D positions $P_i$ and estimated 2D feature points $p_i$ $(u_i, v_i)$.
  - Using corner detection or other filtering to extract features.

$$p = \frac{1}{z} MP, \text{where } M = K\begin{pmatrix} R & t \end{pmatrix}$$

$$u = \frac{m_1 \cdot P}{m_3 \cdot P}, v = \frac{m_2 \cdot P}{m_3 \cdot P}$$

$m_i^T$ is the $i^{th}$ row of $M$

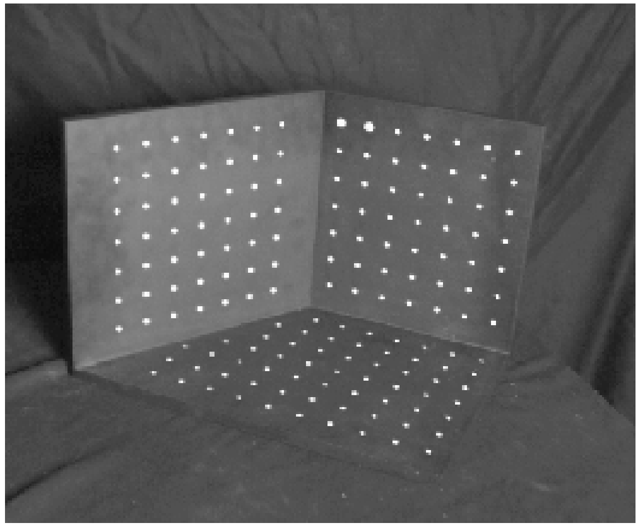


Figure from lecture note of Prof. L.Zhang, Computer Vision, U. Wisconsin-Madison.

# Camera calibration (linear approach)

- With *n* pairs of $P_i$ and $(u_i, v_i)$, we have constraints:

$$(m_1 - u_i m_3) \cdot P_i = 0$$

$$(m_2 - v_i m_3) \cdot P_i = 0$$

- Reform the matrix *V* and unknown *m*
- When n > 6, we can estimate *m* by minimizing $|Vm|^2$

$$V = \begin{bmatrix} P_1^T & 0^T & -u_1 P_1^T \\ 0^T & P_1^T & -v_1 P_1^T \\ \cdots & \cdots & \cdots \\ P_n^T & 0^T & -u_n P_n^T \\ 0^T & P_n^T & -v_n P_n^T \end{bmatrix}_{2n \times 12} \quad \text{and } m = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}_{12 \times 1}$$

# Estimating intrinsic and extrinsic param.

- *M*=(*A b*). Since our |*M*|=1, *ρ* is an unknown scale factor.

$$\rho\begin{pmatrix}A & b\end{pmatrix} = K\begin{pmatrix}R & t\end{pmatrix} \Leftrightarrow \rho\begin{pmatrix}a_1^T\\ a_2^T\\ a_3^T\end{pmatrix} = \begin{pmatrix}\alpha r_1^T - \alpha\cot\theta r_2^T + u_0 r_3^T\\ \dfrac{\beta}{\sin\theta} r_2^T + v_0 r_3^T\\ r_3^T\end{pmatrix}$$

$$\rho = \varepsilon / \|a_3\|$$

$$r_3 = \rho a_3$$

$$u_0 = \rho^2 (a_1 \cdot a_3)$$

$$v_0 = \rho^2 (a_2 \cdot a_3)$$

where $\varepsilon$ =1 or -1.

$$\rho^2 (a_1 \times a_3) = -\alpha r_2 - \alpha\cot\theta r_1 \Rightarrow \rho^2 \|a_1 \times a_3\| = \frac{\|\alpha\|}{\sin\theta}$$

$$\rho^2 (a_2 \times a_3) = \frac{\beta}{\sin\theta} r_1 \Rightarrow \rho^2 \|a_2 \times a_3\| = \frac{\|\beta\|}{\sin\theta}$$

# Estimating intrinsic and extrinsic param.

$$\cos\theta = \frac{-(a_1 \times a_3)\cdot(a_2 \times a_3)}{|a_1 \times a_3||a_2 \times a_3|}$$

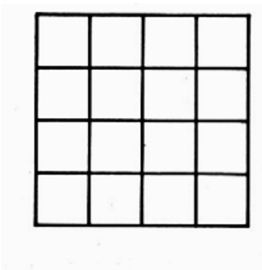
$$\alpha = \rho^2 |a_1 \times a_3| \sin\theta$$

$$\beta = \rho^2 |a_2 \times a_3| \sin\theta$$

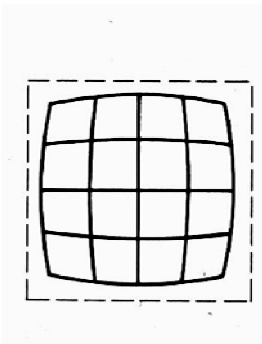
$$r_1 = \frac{(a_2 \times a_3)}{|a_2 \times a_3|}, \text{ and } r_2 = r_3 \times r_1$$

# Radial distortion

- Caused by imperfect lenses
- Deviations are most noticeable for rays that pass through the edge of the lens



No distortion

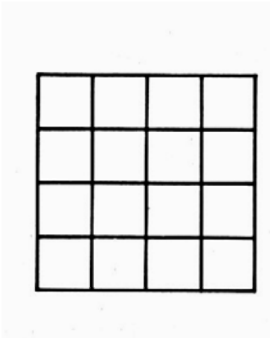


Barrel

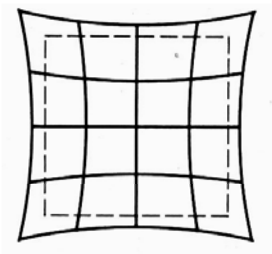
Wide Angle Lens

Figure from lecture note of Prof. L.Zhang, Computer Vision, U. Wisconsin-Madison.

# Radial distortion (cont.)



No distortion



Pin cushion

Telephoto lens

Figure from lecture note of Prof. L.Zhang, Computer Vision, U. Wisconsin-Madison.

# Radial distortion model

- If $u_0 = v_0 = 0$, we can model the distortion as function of $d$.

$$d^2 = \hat{u}^2 + \hat{v}^2$$

$$p = \frac{1}{z}\begin{pmatrix} r(d) & 0 & 0 \\ 0 & r(d) & 0 \\ 0 & 0 & 1 \end{pmatrix} MP$$

$$r(d) = 1 + \kappa_1 d^2 + \kappa_2 d^4 + \kappa_4 d^6$$

- Need non-linear least squares for general cases.

# Calibration with non-linear methods

- The multi-stage linear method can be contaminated by noises or more calibration points are required.
- Using the solution by a linear approach as initial guesses, non-linear optimization can further improve our calibration.

$$E(\xi) = \sum_{i=1}^{n} \left[ \left( \tilde{u}_i(\xi) - u_i \right)^2 + \left( \tilde{v}_i(\xi) - v_i \right)^2 \right],$$

$$\text{where } \tilde{u}_i(\xi) = \frac{m_1(\xi) \cdot P_i}{m_3(\xi) \cdot P_i} \text{ and } \tilde{v}_i(\xi) = \frac{m_2(\xi) \cdot P_i}{m_3(\xi) \cdot P_i}$$

- We can reformulate the objective function for non-linear least square evaluation.

# Calibration with non-linear methods

$$E(\xi) = \sum_{j=1}^{2n} f_j^{2}(\xi)$$

$$f_{2i-1}(\xi) = \tilde{u}_i(\xi) - u_i$$
$$f_{2i}(\xi) = \tilde{v}_i(\xi) - v_i$$

- The gradients are

$$\frac{\partial f_{2i-1}}{\partial \xi_j} = \frac{1}{\tilde{z}_i}\frac{\partial \tilde{x}_i}{\partial \xi_j} - \frac{\tilde{x}_i}{\tilde{z}_i^2}\frac{\partial \tilde{z}_i}{\partial \xi_j} = \frac{1}{z_i}\left(\frac{\partial (m_1 \cdot P_i)}{\partial \xi_j} - \tilde{u}_i \frac{\partial (m_3 \cdot P_i)}{\partial \xi_j}\right)$$

$$\frac{\partial f_{2i}}{\partial \xi_j} = \frac{1}{\tilde{z}_i}\frac{\partial \tilde{y}_i}{\partial \xi_j} - \frac{\tilde{y}_i}{\tilde{z}_i^2}\frac{\partial \tilde{z}_i}{\partial \xi_j} = \frac{1}{z_i}\left(\frac{\partial (m_2 \cdot P_i)}{\partial \xi_j} - \tilde{v}_i \frac{\partial (m_3 \cdot P_i)}{\partial \xi_j}\right)$$

$$\begin{pmatrix} \nabla f_{2i-1}^T \\ \nabla f_{2i}^T \end{pmatrix} = \frac{1}{\tilde{z}_i}\begin{pmatrix} P_i^T & 0^T & -\tilde{u}_i P_i^T \\ 0^T & P_i^T & -\tilde{v}_i P_i^T \end{pmatrix}\mathfrak{I}_m$$

$$\mathfrak{I}_f = \begin{pmatrix} \frac{1}{\tilde{z}_1}P_1^T & 0^T & \frac{-\tilde{u}_1}{\tilde{z}_1}P_1^T \\ 0^T & \frac{1}{\tilde{z}_1}P_1^T & \frac{-\tilde{v}_1}{\tilde{z}_1}P_1^T \\ \cdots & \cdots & \cdots \\ \frac{1}{\tilde{z}_n}P_n^T & 0^T & \frac{-\tilde{u}_n}{\tilde{z}_n}P_n^T \\ 0^T & \frac{1}{\tilde{z}_n}P_n^T & \frac{-\tilde{v}_n}{\tilde{z}_n}P_n^T \end{pmatrix}\mathfrak{I}_m$$

# Calibration with non-linear methods

$$\begin{pmatrix} \nabla f_{2i-1}^T \\ \nabla f_{2i}^T \end{pmatrix} = \frac{1}{\tilde{z}_i} \begin{pmatrix} P_i^T & 0^T & -\tilde{u}_i P_i^T \\ 0^T & P_i^T & -\tilde{v}_i P_i^T \end{pmatrix} \Im_m \qquad \Im_f = \begin{pmatrix} \frac{1}{\tilde{z}_1} P_1^T & 0^T & \frac{-\tilde{u}_1}{\tilde{z}_1} P_1^T \\ 0^T & \frac{1}{\tilde{z}_1} P_1^T & \frac{-\tilde{v}_1}{\tilde{z}_1} P_1^T \\ \cdots & \cdots & \cdots \\ \frac{1}{\tilde{z}_n} P_n^T & 0^T & \frac{-\tilde{u}_n}{\tilde{z}_n} P_n^T \\ 0^T & \frac{1}{\tilde{z}_n} P_n^T & \frac{-\tilde{v}_n}{\tilde{z}_n} P_n^T \end{pmatrix} \Im_m$$

- We can solve the optimization by Gauss-Newton or LM methods.
- In addition to intrinsic and extrinsic parameters, other parameter (e.g. distortions) can also be included.