

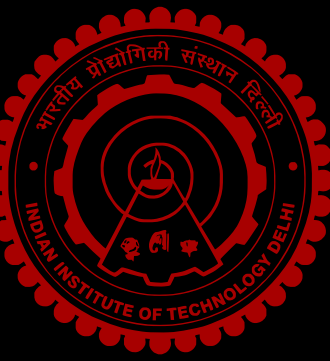


Zhang's Method

Compute Homography

- If all P_i lie on a plane, we can arrange that all points are on the plane $Z_w = 0$

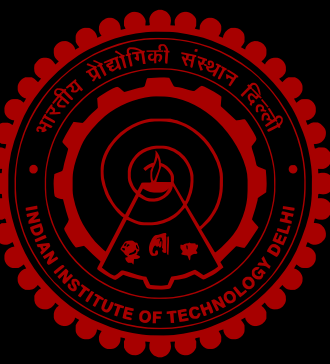
$$\begin{aligned}
 \begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{w} \end{bmatrix} &= \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ \boxed{0} \\ 1 \end{bmatrix} \\
 &\quad \text{Zeroed} \quad \text{Homography matrix} \\
 &\quad \text{Projective mapping} \\
 &\quad \text{from plane to plane} \\
 \begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{w} \end{bmatrix} &= \begin{bmatrix} m_{00} & m_{01} & m_{03} \\ m_{10} & m_{11} & m_{13} \\ m_{20} & m_{21} & m_{23} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} = \begin{bmatrix} H_0^T \\ H_1^T \\ H_2^T \end{bmatrix} P \\
 u &= \frac{H_0^T P}{H_2^T P} \Rightarrow P^T H_0 + 0H_1 - uP^T H_2 = 0 \\
 v &= \frac{H_1^T P}{H_2^T P} \Rightarrow 0H_0 + P^T H_1 - vP^T H_2 = 0 \\
 \Rightarrow \text{Solve} &\quad \begin{bmatrix} P^T & 0^T & -uP^T \\ 0^T & P^T & -vP^T \end{bmatrix} \begin{bmatrix} H_0 \\ H_1 \\ H_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
 &\quad \text{9 var, 8 DOF} \quad \text{Use the DLT method,} \\
 &\quad \text{actually computing } H \text{ instead of full } M
 \end{aligned}$$



Stability of Homography Computation

Hartley pre-conditioning

- Small change in P and p can make a large impact in Homography computed
- Pre-condition the matrix by normalizing P
 - Translate center of mass to the origin: $P_i - \text{mean}$
 - Scale the point so that average distance of points from origin is $\sqrt{2}$



Zhang's Method

Computing Intrinsic matrix

- We compute H up to a scale, one for each view: $\begin{bmatrix} m_{00} & m_{01} & m_{03} \\ m_{10} & m_{11} & m_{13} \\ m_{20} & m_{21} & m_{23} \end{bmatrix} = [\tilde{H}_0 \quad \tilde{H}_1 \quad \tilde{H}_2]$
- Recall $H = K[R_0|R_1|T]$ $\underbrace{\begin{bmatrix} f_x & s & O_x \\ 0 & f_y & O_y \\ 0 & 0 & 1 \end{bmatrix}}_K \underbrace{\begin{bmatrix} r_{00} & r_{01} & t_0 \\ r_{10} & r_{11} & t_1 \\ r_{20} & r_{21} & t_2 \end{bmatrix}}_{\begin{smallmatrix} \uparrow \\ R_0 \end{smallmatrix} \quad \begin{smallmatrix} \uparrow \\ R_1 \end{smallmatrix} \quad \begin{smallmatrix} \uparrow \\ T \end{smallmatrix}}$ with $R_0^T R_0 = R_1^T R_1 = 1$ and $R_0^T R_1 = 0$
and $R_0 = \lambda K^{-1} \tilde{H}_0$ and $R_1 = \lambda K^{-1} \tilde{H}_1$

$$\Rightarrow \tilde{H}_0^T \boxed{K^{-T} K^{-1}} \tilde{H}_0 = \tilde{H}_1^T K^{-T} K^{-1} \tilde{H}_1 \quad \text{and} \quad \tilde{H}_0^T K^{-T} K^{-1} \tilde{H}_1 = 0$$

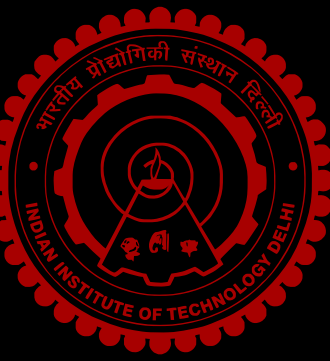
- Rewrite as $\tilde{H}_0^T B \tilde{H}_0 - \tilde{H}_1^T B \tilde{H}_1 = 0$ and $\tilde{H}_0^T B \tilde{H}_1 = 0$ (Each H gives two equations for unknown B)
(Six non-zero values of B)

Cholesky decomposition:

$$B = LL^T \Rightarrow L^{-T} \text{ is } K$$

← Solve for B (DLT method) ←

$$\begin{bmatrix} f(\tilde{H}_0, \tilde{H}_1) \\ g(\tilde{H}_0, \tilde{H}_1) \\ \vdots \end{bmatrix} \swarrow B_c = \begin{bmatrix} 0 \\ 0 \\ \vdots \end{bmatrix}$$



Zhang's Method

Computing Extrinsic matrix

- We compute K up to a scale
- Recall: $R_0 = \lambda K^{-1} \tilde{H}_0$ and $R_1 = \lambda K^{-1} \tilde{H}_1$
 - and $\|R_0\| = \|R_1\| = 1$ and
- Initialize: $\tilde{R}_0 = K^{-1} \tilde{H}_0$ and $\lambda = \frac{1}{\|\tilde{R}_0\|}$
 $\Rightarrow R_0 = \lambda \tilde{R}_0$
- Similarly, $R_1 = \lambda K^{-1} \tilde{H}_1$, $R_2 = R_0 \times R_1$, and $T = \lambda K^{-1} \tilde{H}_2$
- Finally, make these more accurate: treat as starting guess and refine further, incorporating lens distortion

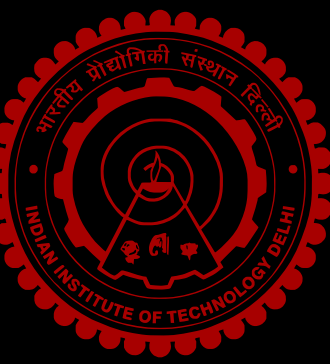
$$\begin{bmatrix} \tilde{H}_0 & \tilde{H}_1 & \tilde{H}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & s & O_x \\ 0 & f_y & O_y \\ 0 & 0 & 1 \end{bmatrix}}_K \begin{bmatrix} r_{00} & r_{01} & t_0 \\ r_{10} & r_{11} & t_1 \\ r_{20} & r_{21} & t_2 \end{bmatrix}$$

$\begin{matrix} \uparrow & \uparrow & \uparrow & \uparrow \\ K & R_0 & R_1 & T \end{matrix}$

Minimize

$$\sum_{i=0}^{n-1} \sum_{c=0}^{m-1} \|p_{ic} - f(K, R_i, T_i, P_c)\|^2$$

(Bundle Adjustment)



Review Zhang's Camera Calibration

- Image chessboard with corner positions at P_i from multiple camera locations
 - Detect precise corners in each image, p_j
- Compute one Homography H , for each image
- Computer matrix $B = K^{-T}K^{-1}$ from multiple homographies
- Factorize B to get a good estimate of K
- From each H and (a single) K compute (R, T) , for each camera
- Perform Levenberg-Marquardt Optimization to refine K , all R and T

Sub-pixel Accuracy of Corners

$$(p_i - q) \cdot \nabla(p_i) \approx 0$$

$$\text{Solve } [\nabla(p_i)]^\top (p_i - q) = 0$$

$$[\nabla(p_i)]^\top q = [\nabla(p_i)]^\top p_i \quad \rightarrow \quad Aq = b$$

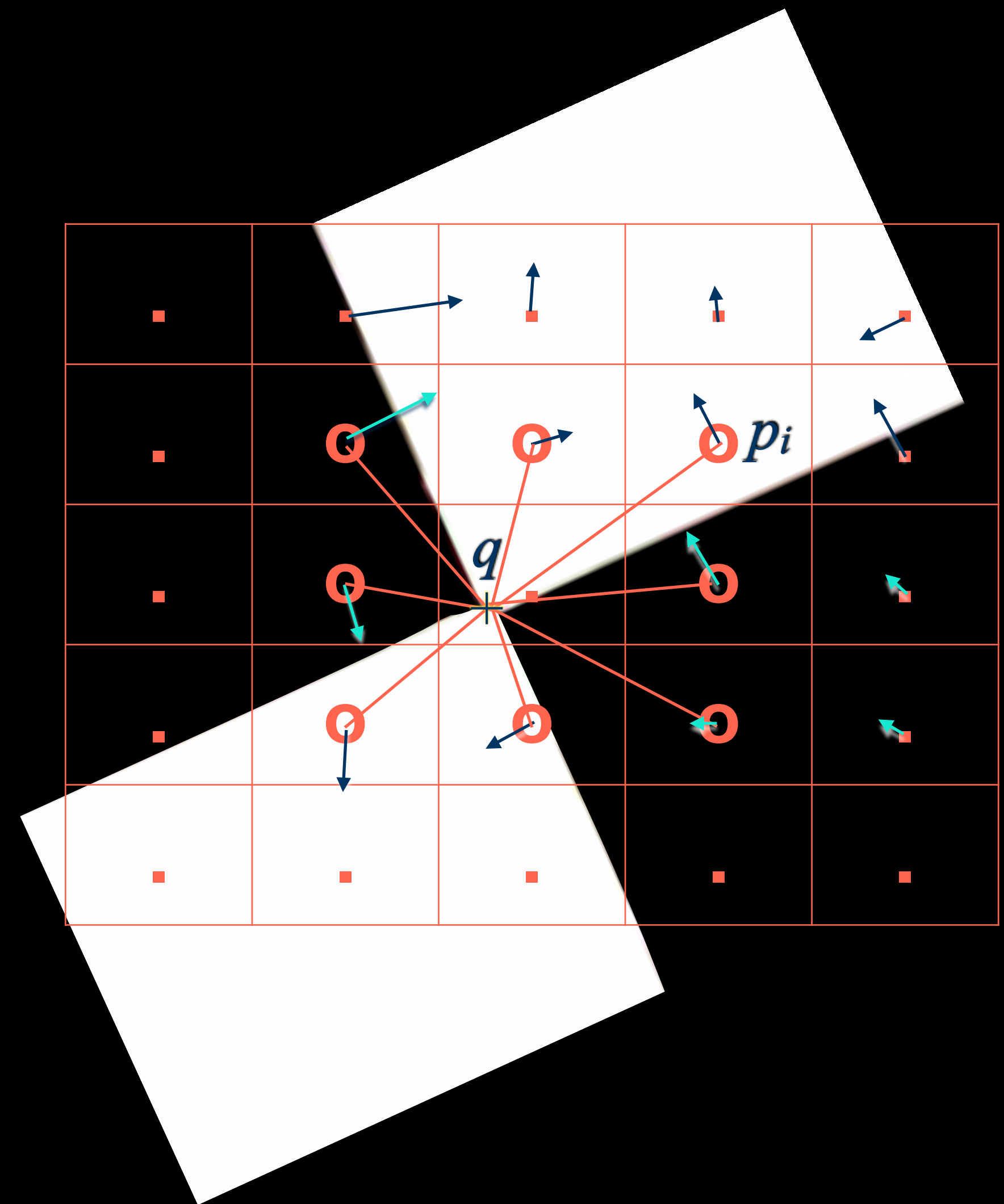
To minimize $\sum \|\nabla(p_i)^\top (p_i - q)\|^2$:

$$\sum [\nabla(p_i)][\nabla(p_i)]^\top q = \sum [\nabla(p_i)][\nabla(p_i)]^\top p_i$$

Structure tensor (gradient auto-correlations)

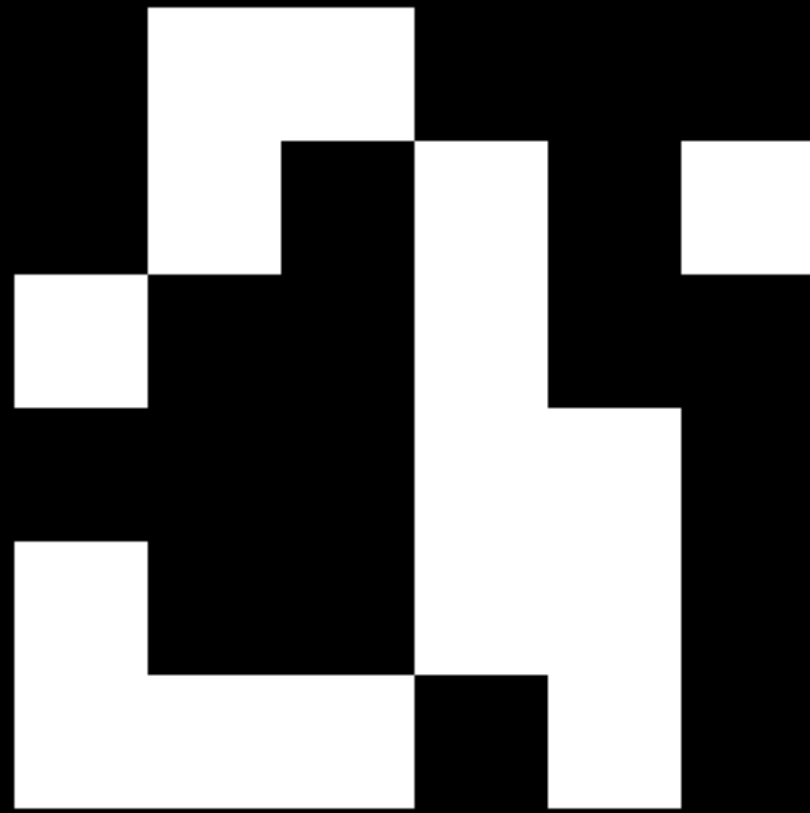
Actually, also weight each p_i by the Gaussian:

$$\sum w_i [\nabla(p_i)][\nabla(p_i)]^\top q = \sum [\nabla(p_i)][\nabla(p_i)]^\top p_i$$

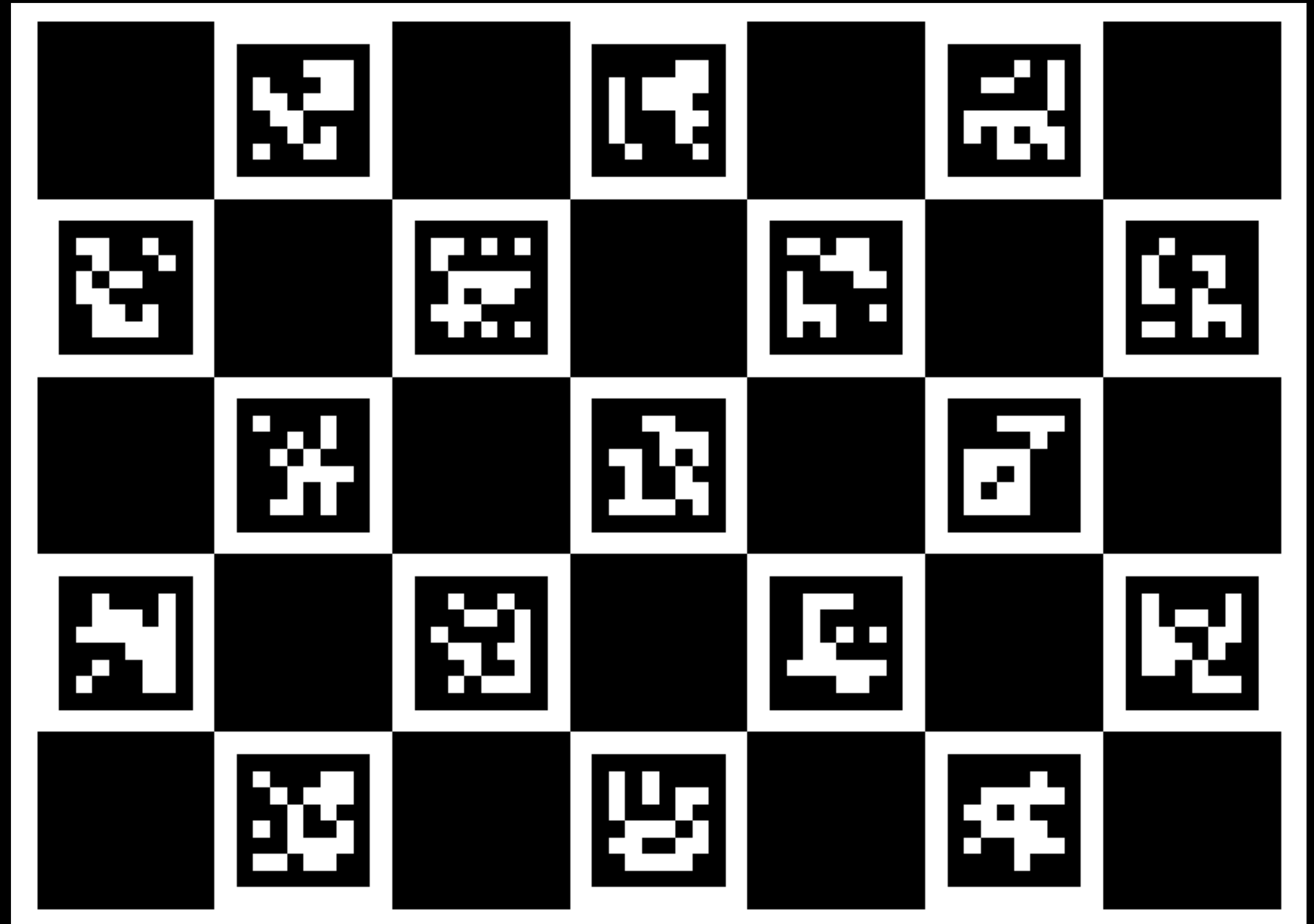


Aruco Markers

For more precision than a chessboard



Example of Aruco marker



ChAruco marker



Recovering Parameters of Model

Model belongs to a known class

- If noisy data fits a model W , *e.g.*, $O = W(P)$. Say, W is represented by $M \in \mathbb{R}^m$
 O and P are vectors
(M symbolizes W)

i.e., $O_i \approx W(P_i)$

- Best model fit:

$$\arg \min_{M \in \mathbb{R}^m} E = \sum_i \text{Diff}(O_i, W(P_i)), \text{ e.g., } \arg \min_{M \in \mathbb{R}^m} \sum_i \|O_i - W(P_i)\| \text{ (L}_2\text{: least squares)}$$

- To minimize E , solve:

If M denotes matrix multiplication, the least squares solution can be found analytically

$$\frac{\partial E}{\partial M_j} = 0 \text{ for all components } M_j \text{ of } M$$

Similarly, for restricted DOF matrix transformations

In general, use Levenberg-Marquardt method (bundle adjustment)