Lecture 7
Multi-view geometry

• The SFM problem
• Affine SFM
• Perspective SFM
• Self-calibration
• Applications

Reading:
[HZ] Chapter 10 “3D reconstruction of cameras and structure”
Chapter 18 “N-view computational methods”
Chapter 19 “Auto-calibration”

[FP] Chapter 13 “projective structure from motion”
[Szelisky] Chapter 7 “Structure from motion”
Affine structure from motion
(simpler problem)

From the $m \times n$ observations $x_{ij}$, estimate:
• $m$ projection matrices $M_i$ (affine cameras)
• $n$ 3D points $X_j$
Affine structure from motion
(simpler problem)

For the affine case (in Euclidean space)

\[ x_{ij} = A_i X_j + b_i \]  
[Eq. 4]

Image 1

World point \( X_i \)

Image i

2x1  2x3  3x1  2x1
The Affine Structure-from-Motion Problem

Two approaches:

- Algebraic approach (affine epipolar geometry; estimate F; cameras; points)
- Factorization method
A factorization method – Tomasi & Kanade algorithm


- Data centering
- Factorization
A factorization method - Centering the data

Centering: subtract the centroid of the image points

[Eq. 5] \[ \bar{x}_i = \frac{1}{n} \sum_{k=1}^{n} x_{ik} \]

[Eq. 6] \[ \hat{x}_{ij} = x_{ij} - \frac{1}{n} \sum_{k=1}^{n} x_{ik} \]
A factorization method - Centering the data

Centering: subtract the centroid of the image points

$$\hat{x}_{ij} = x_{ij} - \frac{1}{n} \sum_{k=1}^{n} x_{ik} = A_i X_j + b_i - \frac{1}{n} \sum_{k=1}^{n} A_i X_k - \frac{1}{n} \sum_{k=1}^{n} b_i$$  \[Eq. 6\]

$$x_{ik} = A_i X_k + b_i$$  \[Eq. 4\]

$$\bar{x}_i = \frac{1}{n} \sum_{k=1}^{n} x_{ik}$$  \[Eq. 5\]
A factorization method  - Centering the data

Centering: subtract the centroid of the image points

\[ \hat{x}_{ij} = x_{ij} - \frac{1}{n} \sum_{k=1}^{n} x_{ik} = A_i X_j + b_i - \frac{1}{n} \sum_{k=1}^{n} A_i X_k - \frac{1}{n} \sum_{k=1}^{n} b_i \]

\[ x_{ik} = A_i X_k + b_i \]

\[ \bar{X} = \frac{1}{n} \sum_{k=1}^{n} X_k \]  [Eq. 7]

Centroid of 3D points
A factorization method - Centering the data

Thus, after centering, each normalized observed point is related to the 3D point by

$$\hat{X}_{ij} = A_i \hat{X}_j \quad [\text{Eq. 8}]$$

$$\bar{X}_i = \frac{1}{n} \sum_{k=1}^{n} X_{ik}$$

Centroid of 3D points

$$\bar{X} = \frac{1}{n} \sum_{k=1}^{n} X_k \quad [\text{Eq. 7}]$$
A factorization method - Centering the data

If the centroid of points in 3D = center of the world reference system

\[ \hat{X}_{ij} = A_i \hat{X}_j = A_i X_j \]  

[Eq. 9]

\[ \bar{X}_i = \frac{1}{n} \sum_{k=1}^{n} X_{ik} \]

Centroid of 3D points

\[ \bar{X} = \frac{1}{n} \sum_{k=1}^{n} X_k \]  

[Eq. 7]
A factorization method - factorization

Let’s create a \(2m \times n\) data (measurement) matrix:

\[
\mathbf{D} = \begin{bmatrix}
\hat{x}_{11} & \hat{x}_{12} & \cdots & \hat{x}_{1n} \\
\hat{x}_{21} & \hat{x}_{22} & \cdots & \hat{x}_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{x}_{m1} & \hat{x}_{m2} & \cdots & \hat{x}_{mn}
\end{bmatrix}
\]

Each \(\hat{x}_{ij}\) entry is a 2x1 vector!
A factorization method - factorization

Let’s create a $2m \times n$ data (measurement) matrix:

$$D = \begin{bmatrix}
\hat{X}_{11} & \hat{X}_{12} & \cdots & \hat{X}_{1n} \\
\hat{X}_{21} & \hat{X}_{22} & \cdots & \hat{X}_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{X}_{m1} & \hat{X}_{m2} & \cdots & \hat{X}_{mn}
\end{bmatrix} = \begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_m
\end{bmatrix} \begin{bmatrix}
X_1 \\
X_2 \\
\vdots \\
X_n
\end{bmatrix}$$

(points $3 \times n$)

cameras

(2 $m \times 3$)

Each $\hat{X}_{ij}$ entry is a 2x1 vector!

$A_i$ is 2x3 and $X_i$ is 3x1

The measurement matrix $D = M S$ has rank 3

(it’s a product of a 2mx3 matrix and 3xn matrix)
Factorizing the Measurement Matrix

How to factorize $D$?

$$D = MS$$
Factorizing the Measurement Matrix

- By computing the Singular value decomposition of $D$!

\[ D = U \times W \times V^T \]
Since rank \((D) = 3\), there are only 3 non-zero singular values \(\sigma_1\), \(\sigma_2\) and \(\sigma_3\).

Factorizing the Measurement Matrix

Where

\[
W_3 = \begin{bmatrix}
\sigma_1 & 0 & 0 \\
0 & \sigma_2 & 0 \\
0 & 0 & \sigma_3
\end{bmatrix}
\]  

[Eq. 11]
Factorizing the Measurement Matrix

\[ D = U_3 \times W_3 \times V_3^T \]
Factorizing the Measurement Matrix

\[ D = U_3 \begin{pmatrix} W_3 & V_3^T \end{pmatrix} = U_3 \begin{pmatrix} W_3 & V_3^T \end{pmatrix} = M S \quad [\text{Eq. 12}] \]
Factorizing the Measurement Matrix

\[ D = U_3 \, W_3 \, V_3^T = U_3 \, (W_3 \, V_3^T) = M \, S \]  [Eq. 12]

What is the issue here? \( D \) has rank>3 because of:

- measurement noise
- affine approximation

Theorem: When \( D \) has a rank greater than 3, \( U_3 \, W_3 \, V_3^T \) is the best possible rank-3 approximation of \( D \) in the sense of the Frobenius norm.

\[
\begin{align*}
D &= U_3 \, W_3 \, V_3^T \\
&= \begin{cases} 
M \approx U_3 \\
S \approx W_3 \, V_3^T 
\end{cases}
\end{align*}
\]

\[
\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\sum_{i=1}^{\min\{m, n\}} \sigma_i^2}
\]
Reconstruction results

Affine Ambiguity

\[ D = M S \]
Affine Ambiguity

- The decomposition is not unique. We get the same \( D \) by applying the transformations:

\[
\begin{align*}
M^* &= M H \\
S^* &= H^{-1} S
\end{align*}
\]

where \( H \) is an arbitrary 3x3 matrix describing an affine transformation.

- Additional constraints must be enforced to resolve this ambiguity.
Affine Ambiguity

Affine

\[ S^* = H^{-1}S \]

\[ A^* = AH \]

\[ A'^* = A'H \]
The Affine Structure-from-Motion Problem

Given $m$ images of $n$ fixed points $X_j$ we can write

$$x_{ij} = A_i X_j + b_i$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, n$

N. of cameras N. of points

Problem: estimate $m$ matrices $A_i$, $m$ matrices $b_i$

and the $n$ positions $X_j$ from the $m \times n$ observations $x_{ij}$.

How many equations and how many unknown?

$2m \times n$ equations in $8m + 3n - 8$ unknowns
Similarity Ambiguity

• The scene is determined by the images only up a similarity transformation (rotation, translation and scaling)

• This is called **metric reconstruction**

• The ambiguity exists even for (intrinsically) calibrated cameras
• For calibrated cameras, the similarity ambiguity is the **only** ambiguity

[Longuet-Higgins ’81]
Similarity Ambiguity

- It is impossible, based on the images alone, to estimate the absolute scale of the scene
Resolving the similarity ambiguity

While calibrating a camera, we make assumptions about the geometry of the world.
Lecture 7
Multi-view geometry

• The SFM problem
• Affine SFM
• Perspective SFM
• Self-calibration
• Applications
Structure from motion problem

From the \( m \times n \) observations \( x_{ij} \), estimate:

- \( m \) projection matrices \( M_i \)
- \( n \) 3D points \( X_j \)

\( \text{motion} \)

\( \text{structure} \)
Structure from motion problem

$m$ cameras $M_1...M_m$

$$M_i = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & 1 \end{bmatrix}$$
Structure from Motion Ambiguities

In the general case (nothing is known) the ambiguity is expressed by an arbitrary 4X4 projective transformation.

\[ x_j = M_i X_j \]

\[ M_i = K_i [R_i \quad T_i] \]

\[ H X_j \]

\[ M_j H^{-1} \]

\[ x_j = M_i X_j = \left( M_i H^{-1} \right) \left( H X_j \right) \]
The Structure-from-Motion Problem

Given $m$ images of $n$ fixed points $X_j$ we can write

$$x_{ij} = M_i X_j$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

**Problem:** estimate $m$ $3 \times 4$ matrices $M_i$ and $n$ positions $X_j$ from $m \times n$ observations $x_{ij}$.

- If the cameras are not calibrated, cameras and points can only be recovered up to a $4 \times 4$ projective (where the $4 \times 4$ projective is defined up to scale)
- Given two cameras, how many points are needed?
- How many equations and how many unknowns?

$2m \times n$ equations in $11m + 3n - 15$ unknowns
Projective Ambiguity

$S =$
Metric reconstruction (upgrade)

- The problem of recovering the metric reconstruction from the perspective one is called **self-calibration**
Structure-from-Motion methods

1. Recovering structure and motion up to perspective ambiguity
   - Algebraic approach (by fundamental matrix)
   - Factorization method (by SVD)
   - Bundle adjustment

2. Resolving the perspective ambiguity
Algebraic approach (2-view case)

1. Compute the fundamental matrix $F$ from two views
2. Use $F$ to estimate projective cameras
3. Use these cameras to triangulate and estimate points in 3D
Algebraic approach (2-view case)

From at least 8 point correspondences, compute $F$ associated to camera 1 and 2.

For $j = 1, \ldots, n$

\[ x_{1j} = M_1 X_j \]
\[ x_{2j} = M_2 X_j \]
1. Compute the fundamental matrix $F$ from two views (eg. 8 point algorithm)
2. Use $F$ to estimate projective cameras
3. Use these cameras to triangulate and estimate points in 3D
Algebraic approach (2-view case)

Because of the projective ambiguity, we can always apply a projective transformation $H$ such that:

$$M_1 H^{-1} = \begin{bmatrix} I & 0 \end{bmatrix}$$  \hspace{2cm} [Eq. 3]  

Canonical perspective camera

$$M_2 H^{-1} = \begin{bmatrix} A & b \end{bmatrix}$$  \hspace{2cm} [Eq. 4]

$$x_{1j} = M_1 X_j$$

$$x_{2j} = M_2 X_j$$

For $j = 1, \ldots, n$

N. of points
Algebraic approach (2-view case)

- Call $X$ a generic 3D point $x_{ij}$
- Call $x$ and $x'$ the corresponding observations to camera 1 and respectively

\[
\begin{align*}
\tilde{M}_1 &= M_1 H^{-1} = \begin{bmatrix} 1 & 0 \\ \end{bmatrix} \quad \text{x} = M_1 X = M_1 H^{-1} H X = [I|0]\tilde{X} \quad \text{[Eq. 6]} \\
\tilde{M}_2 &= M_2 H^{-1} = \begin{bmatrix} A & b \end{bmatrix} \quad \text{x'} = M_2 X = M_2 H^{-1} H X = [A|b]\tilde{X} \\
\tilde{X} &= H X \\
x' = [A|b]\tilde{X} = [A|b] \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \tilde{X}_3 \\ 1 \end{bmatrix} = A[I|0] \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \tilde{X}_3 \\ 1 \end{bmatrix} + b = A[I|0] \tilde{X} + b = A x + b \quad \text{[Eq. 7]} \\
x' \times b = (A x + b) \times b = A x \times b \quad \text{[Eq. 8]} \\
x'^T \cdot (x' \times b) = x'^T \cdot (A x \times b) = 0 \quad \text{[Eq. 9]} \\
x'^T (b \times A x) = 0 \quad \text{[Eq. 10]}
\end{align*}
\]

\[\text{[Eqs. 5]}\]
Cross product as matrix multiplication

\[ \mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a} \times] \mathbf{b} \]
Algebraic approach (2-view case)

\[
\begin{align*}
\tilde{M}_1 &= M_1 H^{-1} = \begin{bmatrix} I & 0 \end{bmatrix} \\
\tilde{M}_2 &= M_2 H^{-1} = \begin{bmatrix} A & b \end{bmatrix} \\
\tilde{X} &= H X
\end{align*}
\]

\[x = M_1 H^{-1} H X = [I | 0] \tilde{X} \quad \text{[Eq. 6]}\]

\[x' = M_2 H^{-1} H X = [A | b] \tilde{X} \]

\[\begin{align*}
x'^T (b \times A x) &= 0 \quad \text{[Eq. 10]} \\
x'^T \begin{bmatrix} b \times \end{bmatrix} A x &= 0 \quad \text{is this familiar?} \\
F &= [b \times] A
\end{align*}\]

\[x'^TFx = 0 \quad \text{fundamental matrix!}\]
Compute cameras

\[ x'^T F x = 0 \quad F = [b_x] A = b \times A \quad \text{[Eq. 11]} \]

Compute \( b \):

• Let’s consider the product \( F b \)

\[ F \cdot b = [b_x] A \cdot b = b \times A \cdot b = 0 \quad \text{[Eq. 12]} \]

• Since \( F \) is singular, we can compute \( b \) as least sq. solution of \( F b = 0 \), with \(|b| = 1\) using SVD

• Using a similar derivation, we have that \( b^T F = 0 \) \( \text{[Eq. 12-bis]} \)
Compute cameras

\[ x'^T F x = 0 \quad F = [b_x] A \]

\[ \begin{cases} F b = 0 \quad [\text{Eq. 12}] \\ b^T F = 0 \quad [\text{Eq. 12-bis}] \end{cases} \]

Compute \( A \):

- Define: \( A' = -[b_x] F \)

- Let’s verify that \([b_x] A' \) is equal to \( F \):

Indeed:

\[ [b_x] A' = -[b_x] [b_x] F = -(b b^T - |b|^2 I) F = -b b^T F + |b|^2 F = 0 + 1 \cdot F = F \]

- Thus, \( A = A' = -[b_x] F \)

\[ \tilde{M}_1 = \begin{bmatrix} I & 0 \end{bmatrix} \quad \tilde{M}_2 = \begin{bmatrix} - & [b_x] F & b \end{bmatrix} \]

[Eq. 13]
Interpretation of $\mathbf{b}$

\[ \mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad \mathbf{F} = [\mathbf{b} \times] \mathbf{A} \]

[Eq. 11]

\[ \begin{cases} \mathbf{F} \mathbf{b} = 0 \quad [\text{Eq. 12}] \\ \mathbf{b}^T \mathbf{F} = 0 \quad [\text{Eq. 12-bis}] \end{cases} \]

What's $\mathbf{b}$??
F x₂ is the epipolar line associated with x₂ (l₁ = F x₂)
Fᵀ x₁ is the epipolar line associated with x₁ (l₂ = Fᵀ x₁)
F is singular (rank two)
\[ F e₂ = 0 \quad \text{and} \quad Fᵀ e₁ = 0 \]
F is 3x3 matrix; 7 DOF
Interpretation of $\mathbf{b}$

\[ \mathbf{x'}^T \mathbf{F} \mathbf{x} = 0 \quad \mathbf{F} = [\mathbf{b}_x] \mathbf{A} \]

\[ \begin{cases} \mathbf{F} \mathbf{b} = 0 \\ \mathbf{b}^T \mathbf{F} = 0 \end{cases} \]  

[Eq. 11]

\[ \mathbf{b} \text{ is an epipole!} \]

\[ \tilde{\mathbf{M}}_1 = \begin{bmatrix} \mathbf{I} & 0 \end{bmatrix} \quad \tilde{\mathbf{M}}_2 = \begin{bmatrix} - [\mathbf{b}_x] \mathbf{F} & \mathbf{b} \end{bmatrix} \]

\[ \tilde{\mathbf{M}}_1 = \begin{bmatrix} \mathbf{I} & 0 \end{bmatrix} \quad \tilde{\mathbf{M}}_2 = \begin{bmatrix} - [\mathbf{e}_x] \mathbf{F} & \mathbf{e} \end{bmatrix} \]

[Eq. 15]  

[Eq. 16]
Algebraic approach (2-view case)

1. Compute the fundamental matrix $F$ from two views (e.g., 8 point algorithm)
2. Use $F$ to estimate projective cameras
3. Use these cameras to triangulate and estimate points in 3D
Triangulation

For \( j = 1, \ldots, n \)

\[
X_j^1 = \tilde{M}_2 \tilde{X}_j
\]

\[
X_j^2 = \tilde{M}_1 \tilde{X}_j
\]

\[
\tilde{M}_1 = \begin{bmatrix} I & 0 \end{bmatrix}
\]

\[
\tilde{M}_2 = \begin{bmatrix} -[e_x]F & e \end{bmatrix}
\]

3D points can be computed from camera matrices via SVD (see page 312 of HZ for details)
Algebraic approach: the N-views case

- From $I_k$ and $I_h \rightarrow \tilde{M}_k, \tilde{M}_h, \tilde{X}_{[k,h]}$

- Pairwise solutions may be combined together using bundle adjustment

3D points associated to point correspondences available between $I_k$ and $I_h$
Structure-from-Motion Algorithms

• Algebraic approach (by fundamental matrix)
• Factorization method (by SVD)
• Bundle adjustment
Limitations of the approaches so far

• Factorization methods assume all points are visible. This not true if:
  • occlusions occur
  • failure in establishing correspondences

• Algebraic methods work with 2 views
Bundle adjustment

- Non-linear method for refining structure and motion
- Minimizes re-projection error

\[
E(M, X) = \sum_{i=1}^{m} \sum_{j=1}^{n} D(x_{ij}, M_i X_j)^2
\]

Reconstructed \( X_j \)

ground truth \( X_j \)
General Calibration Problem

\[ E(M, X) = \sum_{i=1}^{m} \sum_{j=1}^{n} D(x_{ij}, M_i X_j)^2 \]

- **Newton Method**
- **Levenberg-Marquardt Algorithm**
  - Iterative, starts from initial solution
  - May be slow if initial solution far from real solution
  - Estimated solution may be function of the initial solution
  - Newton requires the computation of J, H
  - Levenberg-Marquardt doesn't require the computation of H
Bundle adjustment

• **Advantages**
  • Handle large number of views
  • Handle missing data

• **Limitations**
  • Large minimization problem (parameters grow with number of views)
  • Requires good initial condition

• Used as the final step of SFM (i.e., after the factorization or algebraic approach)
• Factorization or algebraic approaches provide a initial solution for optimization problem
Lecture 7
Multi-view geometry

- The SFM problem
- Affine SFM
- Perspective SFM
- Self-calibration
- Applications
Self-calibration

- **Self-calibration** is the problem of recovering the metric reconstruction from the perspective (or affine) reconstruction
- We can self-calibrate the camera by making some assumptions about the cameras
Self-calibration

Several approaches:

- Use single-view metrology constraints (lecture 4)
- Direct approach (Kruppa Eqs) for 2 views
- Algebraic approach
- Stratified approach

[HZ] Chapters 19 “Auto-calibration”
Inject information about the camera during the bundle adjustment optimization

For calibrated cameras, the similarity ambiguity is the only ambiguity [Longuet-Higgins '81]
Lecture 7

Multi-view geometry

- The SFM problem
- Affine SFM
- Perspective SFM
- Self-calibration
- Applications
Structure from motion problem

Lucas & Kanade, 81
Chen & Medioni, 92
Debevec et al., 96
Levoy & Hanrahan, 96
Fitzgibbon & Zisserman, 98
Triggs et al., 99
Pollefeys et al., 99
Kutulakos & Seitz, 99
Levoy et al., 00
Hartley & Zisserman, 00
Dellaert et al., 00
Rusinkiewic et al., 02
Nistér, 04
Brown & Lowe, 04
Schindler et al., 04
Lourakis & Argyros, 04
Colombo et al. 05
Golparvar-Fard, et al. JAEI 10
Pandey et al. IFAC , 2010
Pandey et al. ICRA 2011
Microsoft’s PhotoSynth
Snavely et al., 06-08
Schindler et al., 08
Agarwal et al., 09
Frahm et al., 10
Reconstruction and texture mapping

M. Pollefeys et al 98–
Incremental reconstruction of construction sites

Initial pair – 2168 & Complete Set 62,323 points, 160 images

Golparvar-Fard, Pena-Mora, Savarese 2008
The registration of images (08.27.08) within the reconstructed scene + Site photos of the Student Dining and Residence Hall project in Champaign, IL. Images courtesy of Turner Construction.
Reconstructed scene + Site photos
Results and applications

Next lecture

• Fitting and Matching
Direct approach

We use the following results:

1. A relationship that maps conics across views
2. Concept of absolute conic and its relationship to K
3. The Kruppa equations
Projections of conics across views

\[ X^T C_w X = 0 \quad \text{[Eq. 1]} \]

\[ X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix} \]

\[ [e']_x C'^{-1} [e']_x = F C^{-1} F^T \quad \text{[Eq. 2]} \]
Projection of absolute conics across views

From lecture 4, [HZ] page 210, sec. 8.5.1

\[ [e']_x \omega^{-1} [e']_x = F \omega^{-1} F^T \]

[Eq. 3]

\[ \omega = (K K^T)^{-1} \]

[Eq. 4]

\[ \omega' = (K' K'^T)^{-1} \]

[Eq. 5]
Kruppa equations

\[
\begin{pmatrix}
  u_2^T K' K'^T u_2 \\
  -u_1^T K' K'^T u_2 \\
  u_1^T K' K'^T u_1
\end{pmatrix}
\times
\begin{pmatrix}
  \sigma_1^2 v_1^T K K^T v_1 \\
  \sigma_1 \sigma_2 v_1^T K K^T v_2 \\
  \sigma_2^2 v_2^T K K^T v_2
\end{pmatrix} = 0
\]  

[Eq. 6]

where \( u_i, v_i \) and \( \sigma_i \) are the columns and singular values of SVD of \( F \)

These give us two independent constraints in the elements of \( K \) and \( K' \)
Kruppa equations

Let's make the following assumption:

\[
\begin{pmatrix}
u_2^T K' K'^T u_2 \\
-u_1^T K' K'^T u_2 \\
u_1^T K' K'^T u_1
\end{pmatrix}
\times
\begin{pmatrix}
\sigma_1^2 v_1^T K K^T v_1 \\
\sigma_1 \sigma_2 v_1^T K K^T v_2 \\
\sigma_2^2 v_2^T K K^T v_2
\end{pmatrix} = 0
\]

\[
\frac{u_2^T K K^T u_2}{\sigma_1^2 v_1^T K K^T v_1} = \frac{-u_1^T K K^T u_2}{\sigma_1 \sigma_2 v_1^T K K^T v_2} = \frac{u_1^T K K^T u_1}{\sigma_2^2 v_2^T K K^T v_2}
\]

[Eq. 7]

Let’s make the following assumption: \( K' = K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \)

[Eq. 8]

[Eq. 9] \( \alpha f^2 + \beta f + \gamma = 0 \quad \Rightarrow \quad f \)
Kruppa equations

[Faugeras et al. 92]

- Powerful if we want to self-calibrate 2 cameras with unknown focal length

- Limitations:
  - Work on a camera pair
  - Don’t work if $R=0$

\[
\begin{align*}
\text{[Eq. 10]} \quad [e']_\times \omega^{-1} [e']_\times &= F \omega^{-1} F^T \quad \text{becomes trivial} \\
\text{Since: } \quad F &= [e']_\times
\end{align*}
\]
Self-calibration

Several approaches:

- Use single-view metrology constraints (lecture 4)
- Direct approach (Kruppa Eqs) for 2 views
- Algebraic approach
- Stratified approach
Suppose we have a projective reconstruction \( \{ \tilde{M}_i, \tilde{X}_j \} \)

Let \( H \) be a homography such that:

\[
\begin{cases}
\text{First perspective camera is canonical: } & \tilde{M}_1 = \begin{bmatrix} I & 0 \end{bmatrix} \quad [\text{Eq. 11}] \\
\text{i}^{th} \text{ perspective reconstruction of the camera (known): } & \tilde{M}_i = \begin{bmatrix} A_i & b_i \end{bmatrix} \quad [\text{Eq. 12}] 
\end{cases}
\]

\[
[A_i - b_i p^T] K_1 K_1^T (A_i - b_i p^T)^T = K_i K_i^T \quad i=2...m
\quad [\text{Eq. 13}]
\]

\[
H = \begin{bmatrix}
K_1 & 0 \\
-p^T K_1 & 1
\end{bmatrix}
\quad p \text{ is an unknown 3x1 vector}
\]

\[
K_1...K_m \text{ are unknown}
\]
Algebraic approach \hspace{1cm} Multi-view approach

Suppose we have a projective reconstruction

Let $\mathbf{H}$ be a homography such that:

\[
\left\{ \begin{array}{l}
\text{First perspective camera is canonical: } \mathbf{\tilde{M}}_1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \quad \text{[Eq. 11]} \\
\text{i}^{\text{th}} \text{ perspective reconstruction of the camera (known): } \mathbf{\tilde{M}}_i = \begin{bmatrix} \mathbf{A}_i & \mathbf{b}_i \end{bmatrix} \\
\end{array} \right.
\]

\text{[Eq. 12]}

\[
\begin{align*}
\left( \mathbf{A}_i - \mathbf{b}_i \mathbf{p}^T \right) \mathbf{K}_1 \mathbf{K}_1^T \left( \mathbf{A}_i - \mathbf{b}_i \mathbf{p}^T \right)^T &= \mathbf{K}_i \mathbf{K}_i^T \\
\end{align*}
\]

\text{[Eq. 13]}

How many unknowns?
- 3 from $\mathbf{p}$
- 5 $m$ from $\mathbf{K}_1...\mathbf{K}_m$

How many equations? 5 independent equations [per view]
Suppose we have a projective reconstruction

Let $H$ be a homography such that:

\[
\begin{aligned}
 \text{First perspective camera is canonical: } & \quad \begin{bmatrix} \tilde{M}_1 \\ \end{bmatrix} = \begin{bmatrix} I & 0 \\ \end{bmatrix} \quad \text{[Eq. 11]} \\
 \text{i^{th} perspective reconstruction of the camera (known): } & \quad \begin{bmatrix} \tilde{M}_i \\ \end{bmatrix} = \begin{bmatrix} A_i & b_i \\ \end{bmatrix} \quad \text{[Eq. 12]}
\end{aligned}
\]

Assume all camera matrices are identical: $K_1 = K_2 \ldots = K_m$

\[
\begin{aligned}
 & \quad \text{[Eq. 15]} \quad \left( A_i - b_i p^T \right) K K^T \left( A_i - b_i p^T \right)^T = K K^T \\
 \text{How many unknowns?} & \quad \text{3 from } p \\
 & \quad \text{5 from } K \\
 \text{How many equations?} & \quad 5 \text{ independent equations [per view]}
\end{aligned}
\]

We need at least 3 views to solve the self-calibration problem
Algebraic approach

Art of self-calibration:
Use assumptions on $K$s to generate enough equations on the unknowns

<table>
<thead>
<tr>
<th>Condition</th>
<th>$N. \text{ Views}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Constant internal parameters</td>
<td>3</td>
</tr>
<tr>
<td>• Aspect ratio and skew known</td>
<td>4</td>
</tr>
<tr>
<td>• Focal length and offset vary</td>
<td></td>
</tr>
<tr>
<td>• Skew =0, all other parameters vary</td>
<td>8</td>
</tr>
</tbody>
</table>

Issue: the larger is the number of view, the harder is the correspondence problem

Bundle adjustment helps!
SFM problem - summary

1. Estimate structure and motion up perspective transformation
   1. Algebraic
   2. factorization method
   3. bundle adjustment

2. Convert from perspective to metric (self-calibration)

3. Bundle adjustment

** or **

1. Bundle adjustment with self-calibration constraints