EE655: Computer Vision & Deep Learning

Lecture 19

Koteswar Rao Jerripothula, PhD
Department of Electrical Engineering
IIT Kanpur

Credits: Prof. Shree K. Nayar

Structure From Motion

Compute 3D scene structure and camera motion from a sequence of frames.

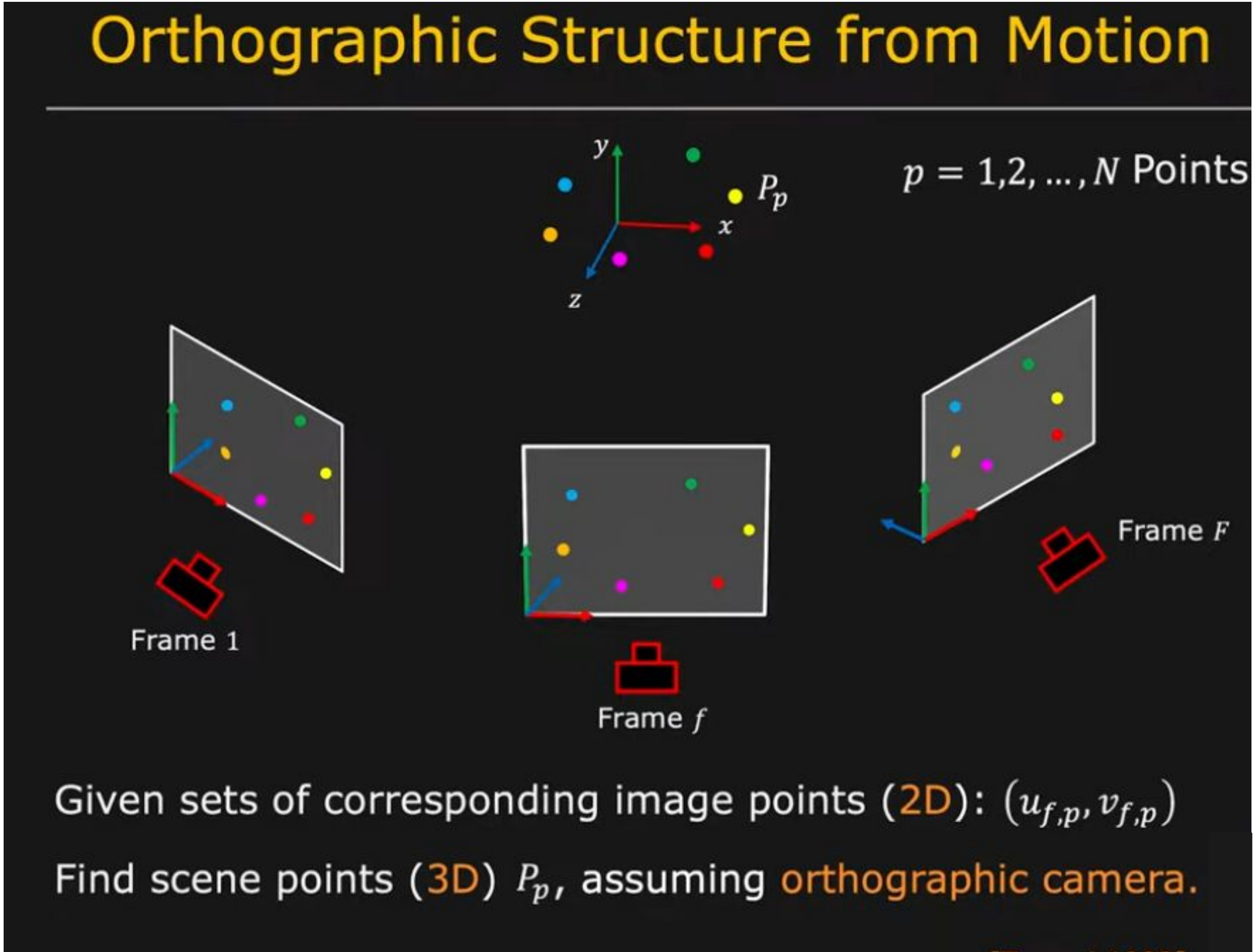
Topics:

(1) Structure from Motion Problem

(2) SFM Observation Matrix

(3) Rank of Observation Matrix

(4) Tomasi-Kanade Factorization



Orthographic Structure from Motion

$p = 1,2, \dots, N$ Points

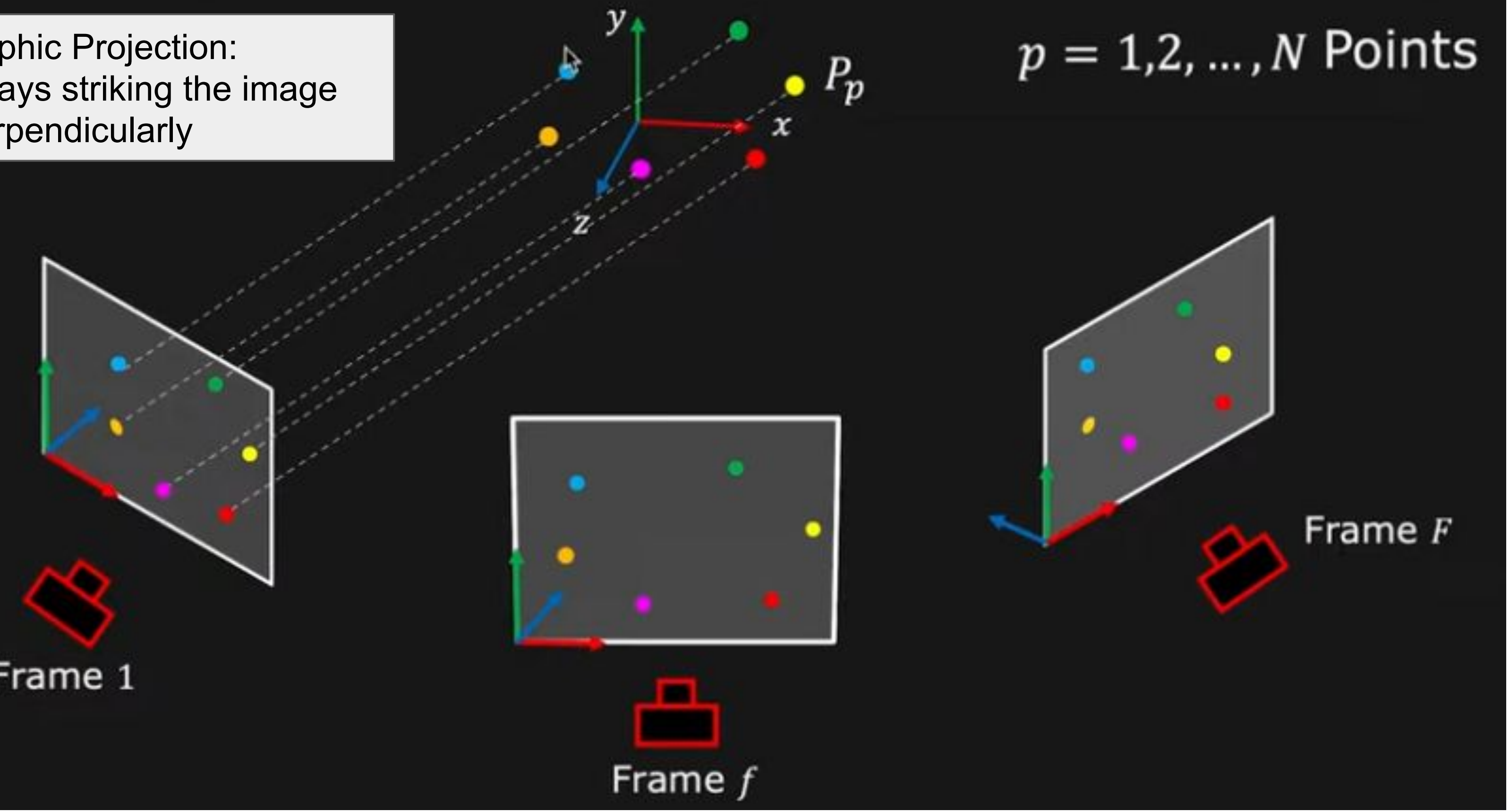
Given sets of corresponding image points (2D): $(u_{f,p}, v_{f,p})$

Find scene points (3D) P_p, assuming orthographic camera.

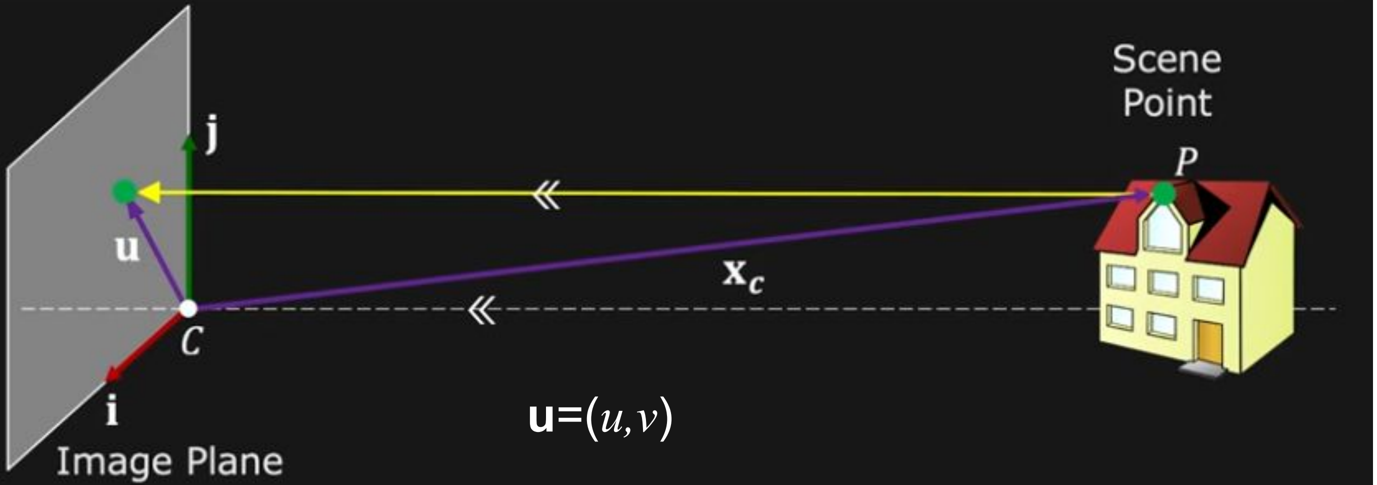
Orthographic Structure from Motion

Orthographic Projection:
Parallel rays striking the image plane perpendicularly

$p = 1,2,\ldots,N$ Points



From 3D to 2D: Orthographic Projection



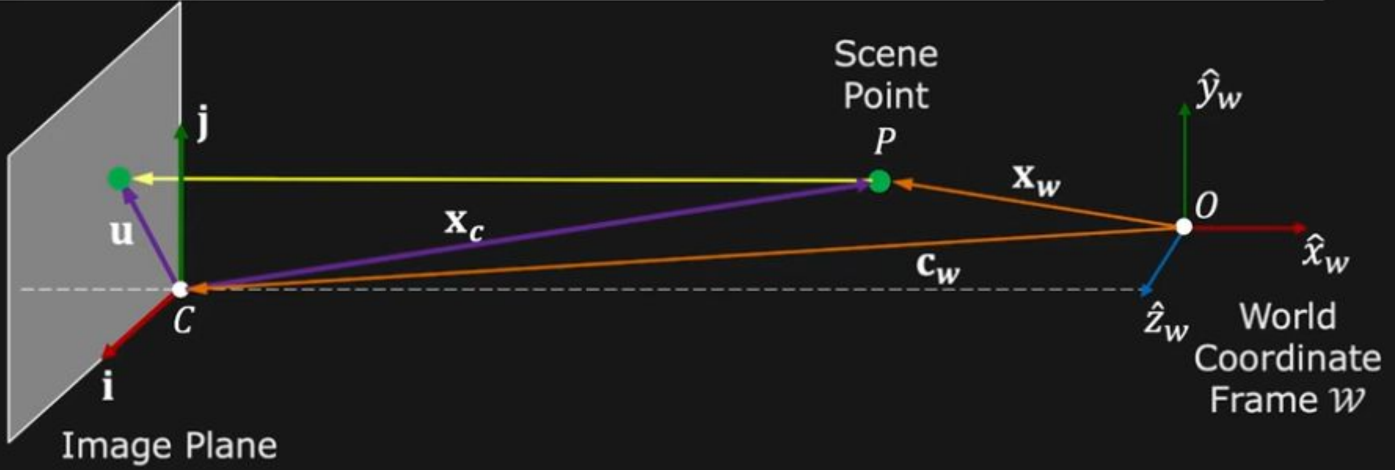
$\mathbf{u}=(u,v)$

$$u = \mathbf{i} \cdot \mathbf{x}_c = \mathbf{i}^T \mathbf{x}_c$$

$$v = \mathbf{j} \cdot \mathbf{x}_c = \mathbf{j}^T \mathbf{x}_c$$

Perspective cameras exhibit orthographic projection when distance of scene from camera is large compared to depth variation within scene (magnification is nearly constant).

From 3D to 2D: Orthographic Projection

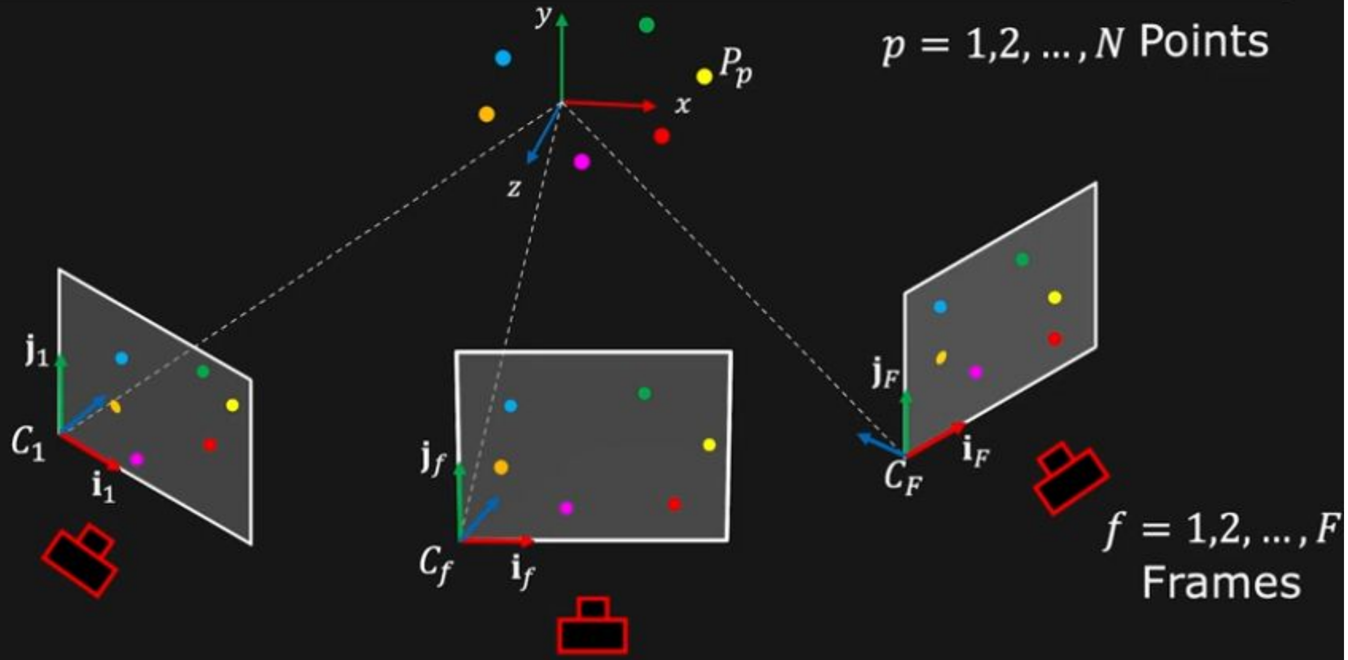


$$u = \mathbf{i}^T \mathbf{x}_c = \mathbf{i}^T(\mathbf{x}_w - \mathbf{c}_w) = \mathbf{i}^T(P - C)$$

$$v = \mathbf{j}^T \mathbf{x}_c = \mathbf{j}^T(\mathbf{x}_w - \mathbf{c}_w) = \mathbf{j}^T(P - C)$$

$$\boxed{\begin{aligned} u &= \mathbf{i}^T(P - C) \\ v &= \mathbf{j}^T(P - C) \end{aligned}}$$

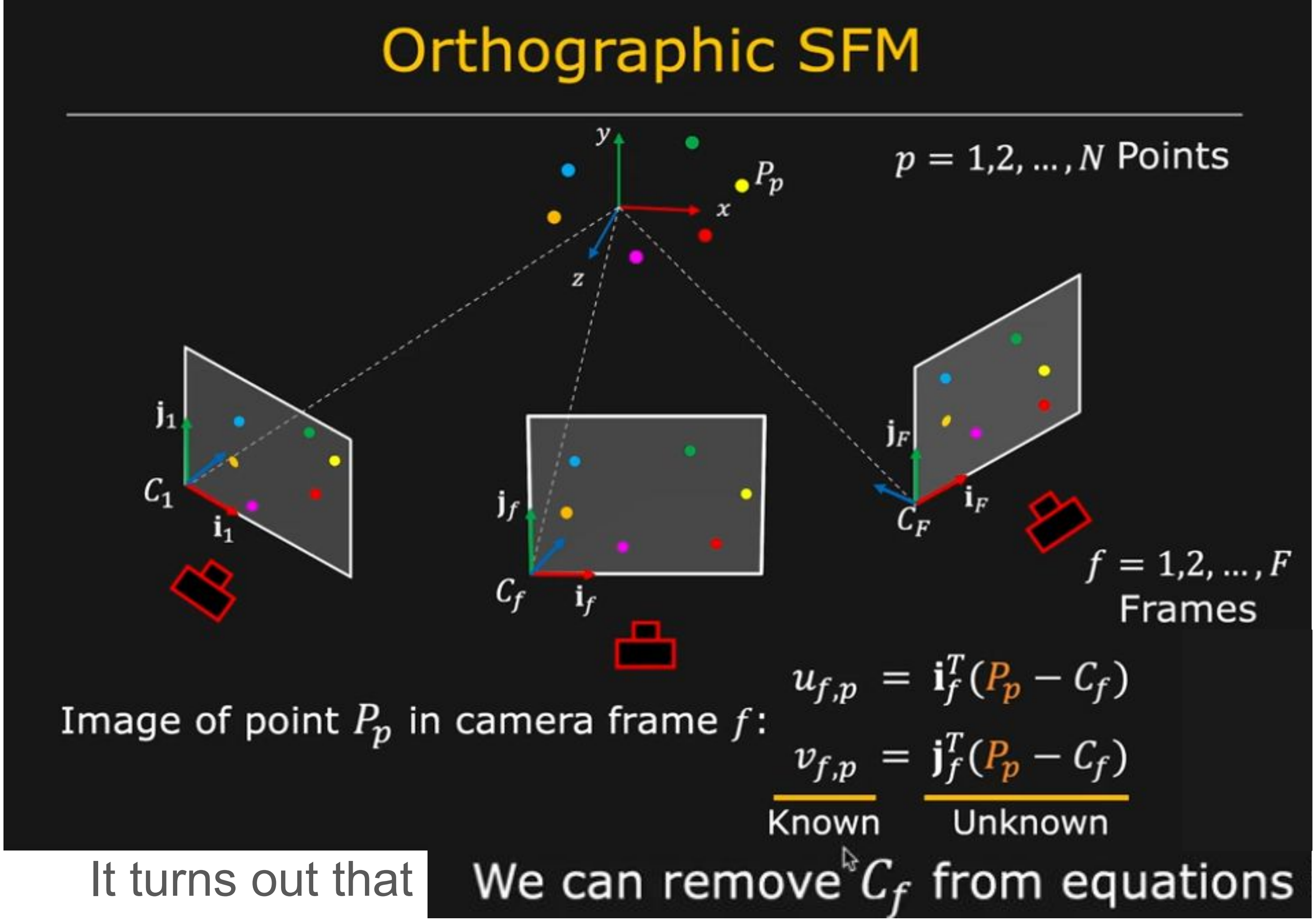
Orthographic SFM



Given corresponding image points (2D) $(u_{f,p}, v_{f,p})$

Find scene points $\{P_p\}$.

Camera Positions $\{C_f\}$, camera orientations $\{(\mathbf{i}_f, \mathbf{j}_f)\}$ are unknown



Orthographic SFM

$p = 1,2, \dots, N$ Points

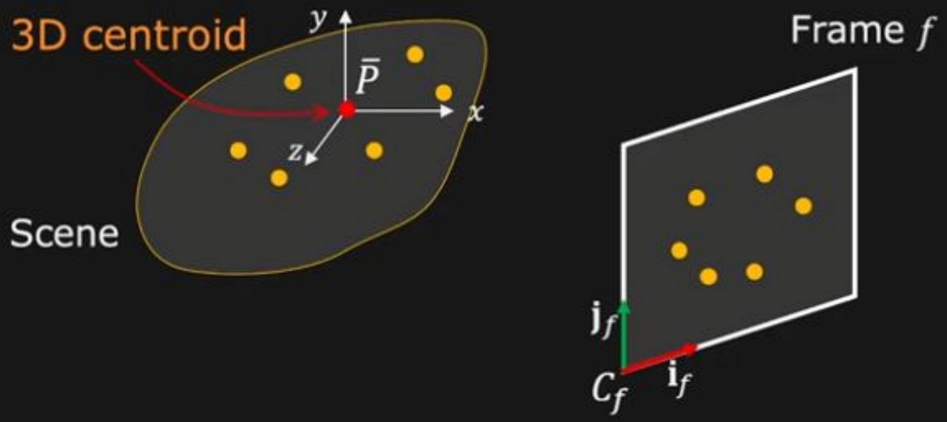
$f = 1,2, \dots, F$ Frames

Image of point P_p in camera frame f:

$$\underbrace{\begin{matrix} u_{f,p} \\ v_{f,p} \end{matrix}}_{\text{Known}} \begin{matrix} = \\ = \end{matrix} \underbrace{\begin{matrix} \mathbf{i}_f^T(P_p - C_f) \\ \mathbf{j}_f^T(P_p - C_f) \end{matrix}}_{\text{Unknown}}$$

It turns out that We can remove C_f from equations

Centering Trick

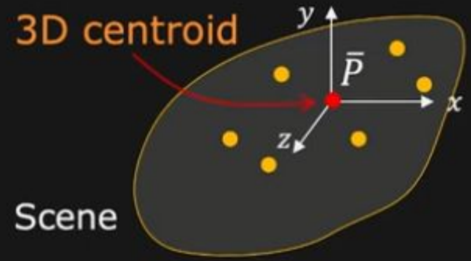


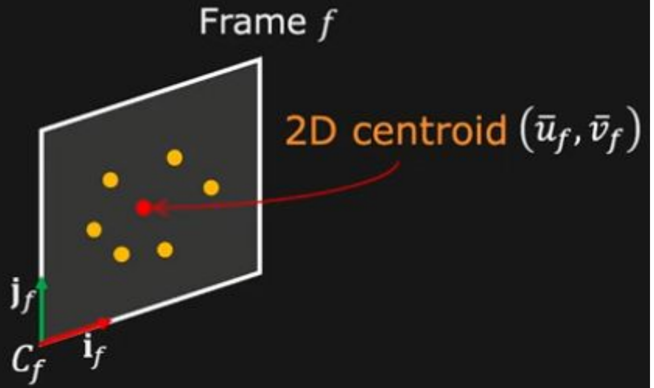
Assume origin of world at centroid of scene points:

$$\frac{1}{N}\sum_{p=1}^{N} P_p = \bar{P} = \mathbf{0}$$

We will compute scene points w.r.t their centroid!

Centering Trick





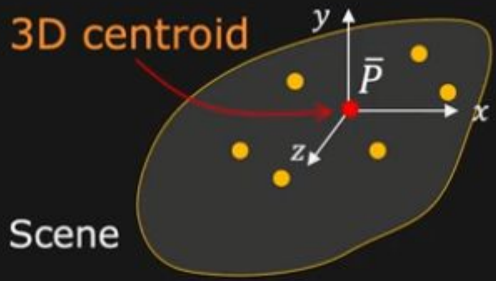
Centroid $(\bar{u}_f, \bar{v}_f)$ of the image points in frame f:

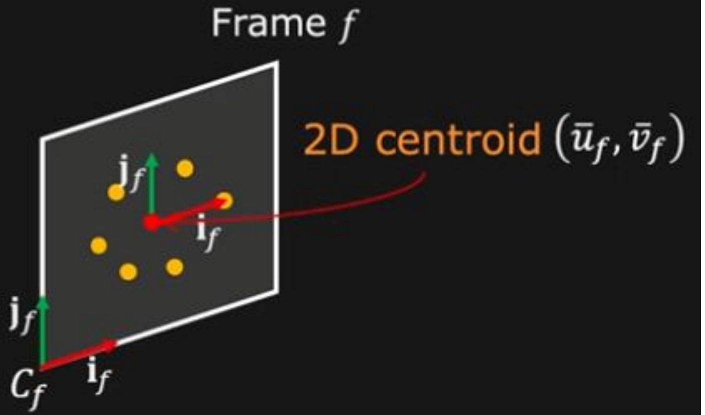
$$\bar{u}_f = \frac{1}{N}\sum_{p=1}^{N} u_{f,p} = \frac{1}{N}\sum_{p=1}^{N} \mathbf{i}_f^T (P_p - C_f) \qquad \bar{v}_f = \frac{1}{N}\sum_{p=1}^{N} v_{f,p} = \frac{1}{N}\sum_{p=1}^{N} \mathbf{j}_f^T (P_p - C_f)$$

$$\bar{u}_f = \frac{1}{N}\mathbf{i}_f^T \cancel{\sum_{p=1}^{N}} P_p - \frac{1}{N}\sum_{p=1}^{N} \mathbf{i}_f^T C_f \qquad \bar{v}_f = \frac{1}{N}\mathbf{j}_f^T \cancel{\sum_{p=1}^{N}} P_p - \frac{1}{N}\sum_{p=1}^{N} \mathbf{j}_f^T C_f$$

$$\boxed{\bar{u}_f = -\mathbf{i}_f^T C_f} \qquad \boxed{\bar{v}_f = -\mathbf{j}_f^T C_f}$$

Centering Trick





Shift camera origin to the centroid $(\bar{u}_f, \bar{v}_f)$.

Image points w.r.t. $(\bar{u}_f, \bar{v}_f)$:

$$\tilde{u}_{f,p} = u_{f,p} - \bar{u}_f \qquad \tilde{v}_{f,p} = v_{f,p} - \bar{v}_f$$

$$= \mathbf{i}_f^T (P_p - C_f) + \mathbf{i}_f^T C_f \qquad = \mathbf{j}_f^T (P_p - C_f) + \mathbf{j}_f^T C_f$$

$$\boxed{\tilde{u}_{f,p} = \mathbf{i}_f^T P_p} \qquad \boxed{\tilde{v}_{f,p} = \mathbf{j}_f^T P_p}$$

Observation Matrix W

$$\tilde{u}_{f,p} = \mathbf{i}_f^T P_p \qquad \tilde{v}_{f,p} = \mathbf{j}_f^T P_p \quad \Rightarrow \quad \begin{bmatrix} \tilde{u}_{f,p} \\ \tilde{v}_{f,p} \end{bmatrix} = \begin{bmatrix} \mathbf{i}_f^T \\ \mathbf{j}_f^T \end{bmatrix} P_p$$

$$\begin{array}{r|cccc|}
 & \text{Point 1} & \text{Point 2} & & \text{Point N} \\
\text{Image 1} & \tilde{u}_{1,1} & \tilde{u}_{1,2} & \cdots & \tilde{u}_{1,N} \\
\text{Image 2} & \tilde{u}_{2,1} & \tilde{u}_{2,2} & \cdots & \tilde{u}_{2,N} \\
 & \vdots & \vdots & \vdots & \vdots \\
\text{Image F} & \tilde{u}_{F,1} & \tilde{u}_{F,2} & \cdots & \tilde{u}_{F,N} \\
\hline
\text{Image 1} & \tilde{v}_{1,1} & \tilde{v}_{1,2} & \cdots & \tilde{v}_{1,N} \\
\text{Image 2} & \tilde{u}_{2,1} & \tilde{u}_{2,2} & \cdots & \tilde{v}_{2,N} \\
 & \vdots & \vdots & \vdots & \vdots \\
\text{Image F} & \tilde{v}_{F,1} & \tilde{v}_{F,2} & \cdots & \tilde{v}_{F,N}
\end{array}
=
\begin{bmatrix} \mathbf{i}_1^T \\ \mathbf{i}_2^T \\ \vdots \\ \mathbf{i}_F^T \\ \hline \mathbf{j}_1^T \\ \mathbf{j}_2^T \\ \vdots \\ \mathbf{j}_F^T \end{bmatrix}
\begin{array}{cccc}
\text{Point 1} & \text{Point 2} & & \text{Point N} \\
{[P_1} & P_2 & \cdots & P_N]
\end{array}$$

$W_{2F\times N}$

Centroid-Subtracted Feature Points (Known)

$M_{2F\times 3}$

Camera Motion (Unknown)

$S_{3\times N}$

Scene Structure (Unknown)

Observation Matrix W

$$\begin{array}{c}
\\
\text{Image 1} \\
\text{Image 2} \\
\\
\text{Image F} \\
\text{Image 1} \\
\text{Image 2} \\
\\
\text{Image F}
\end{array}
\begin{array}{cccc}
\text{Point 1} & \text{Point 2} & & \text{Point N} \\
\tilde{u}_{1,1} & \tilde{u}_{1,2} & \cdots & \tilde{u}_{1,N} \\
\tilde{u}_{2,1} & \tilde{u}_{2,2} & \cdots & \tilde{u}_{2,N} \\
\vdots & \vdots & \vdots & \vdots \\
\tilde{u}_{F,1} & \tilde{u}_{F,2} & \cdots & \tilde{u}_{F,N} \\
\tilde{v}_{1,1} & \tilde{v}_{1,2} & \cdots & \tilde{v}_{1,N} \\
\tilde{u}_{2,1} & \tilde{u}_{2,2} & \cdots & \tilde{v}_{2,N} \\
\vdots & \vdots & \vdots & \vdots \\
\tilde{v}_{F,1} & \tilde{v}_{F,2} & \cdots & \tilde{v}_{F,N}
\end{array}
=
\begin{bmatrix}
\mathbf{i}_1^T \\
\mathbf{i}_2^T \\
\vdots \\
\mathbf{i}_F^T \\
\mathbf{j}_1^T \\
\mathbf{j}_2^T \\
\vdots \\
\mathbf{j}_F^T
\end{bmatrix}
\begin{array}{cccc}
\text{Point 1} & \text{Point 2} & & \text{Point N} \\
{[P_1} & P_2 & \cdots & P_N]
\end{array}$$

$W_{2F\times N}$

Centroid-Subtracted Feature Points (Known)

$M_{2F\times 3}$

Camera Motion (Unknown)

$S_{3\times N}$

Scene Structure (Unknown)

Can we find M and S from W?

Linear Independence of Vectors

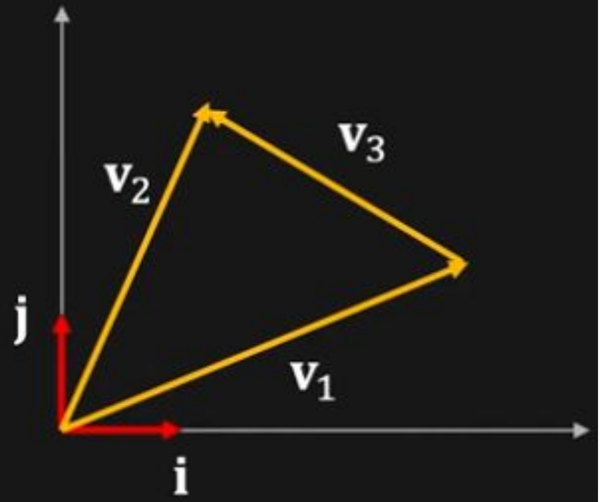
A set of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ is said to be linearly independent if no vector can be represented as a weighted linear sum of the others.

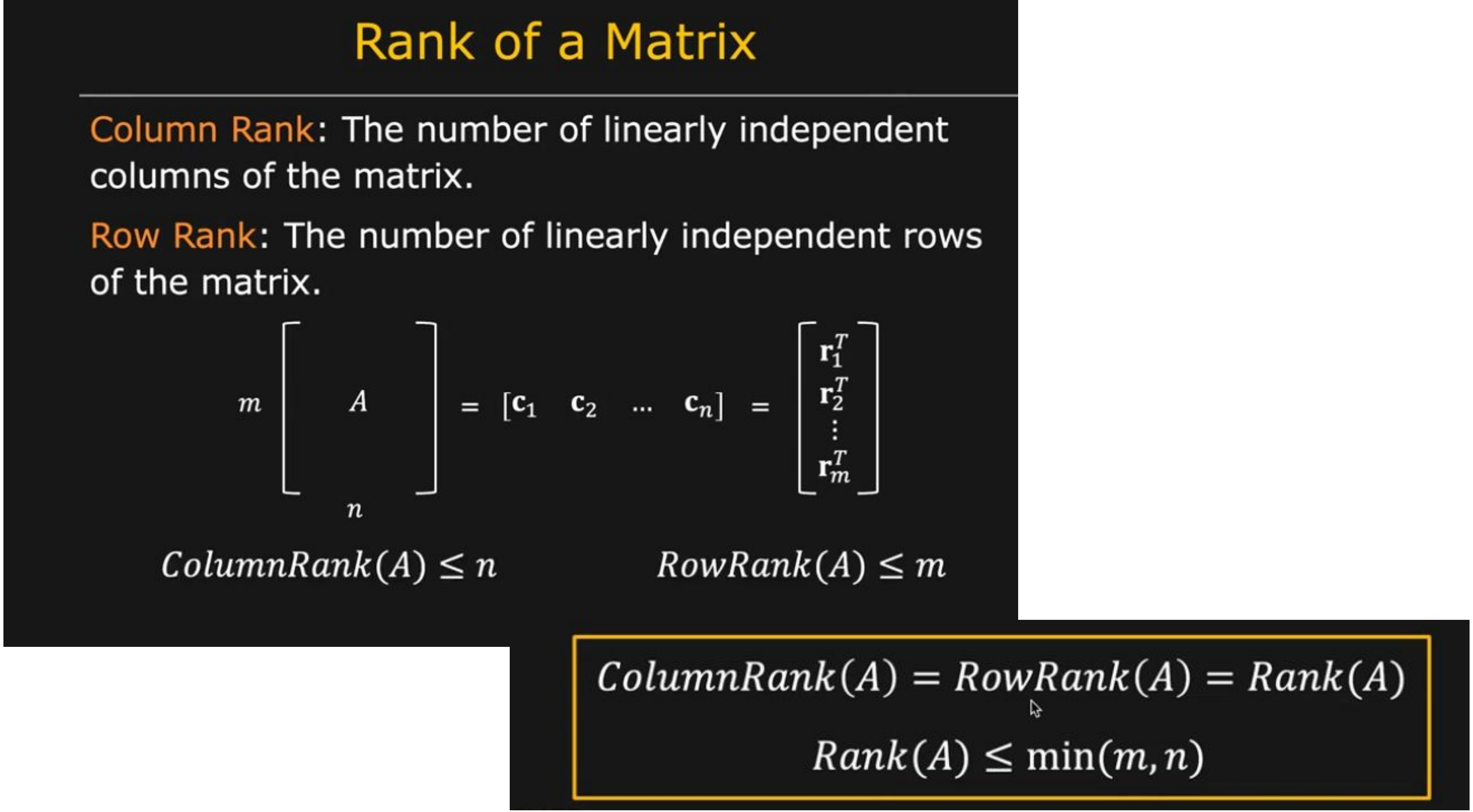
$\{\mathbf{i}, \mathbf{j}\}$ is linearly independent.

$\{\mathbf{i}, \mathbf{j}, \mathbf{v}_1\}$ is linearly dependent.

$\{\mathbf{i}, \mathbf{j}, \mathbf{v}_3\}$ is linearly dependent.

$\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is linearly dependent.





Rank of a Matrix

Column Rank: The number of linearly independent columns of the matrix.

Row Rank: The number of linearly independent rows of the matrix.

$$m\left[\begin{array}{c} \quad A \quad \\ \\ \end{array}\right]_{n} = [\mathbf{c}_1 \quad \mathbf{c}_2 \quad \dots \quad \mathbf{c}_n] = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \vdots \\ \mathbf{r}_m^T \end{bmatrix}$$

$$ColumnRank(A) \leq n \qquad RowRank(A) \leq m$$

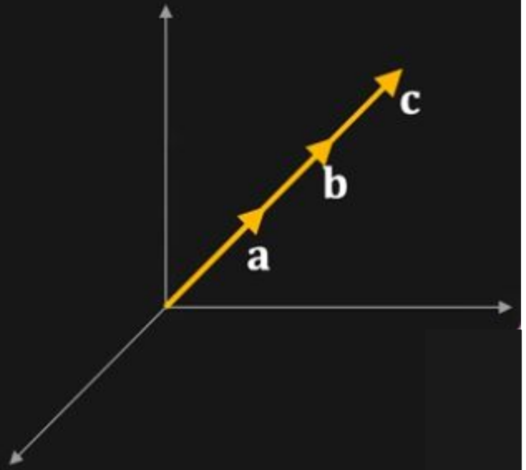
$$\boxed{\begin{array}{c} ColumnRank(A) = RowRank(A) = Rank(A) \\ Rank(A) \leq \min(m, n) \end{array}}$$

Geometric Meaning of Matrix Rank

Rank is the dimensionality of the space spanned by column or row vectors of the matrix.

$$A = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} = [\mathbf{a} \quad \mathbf{b} \quad \mathbf{c}]$$

$$Rank(A) = 1 \quad \Longleftrightarrow$$

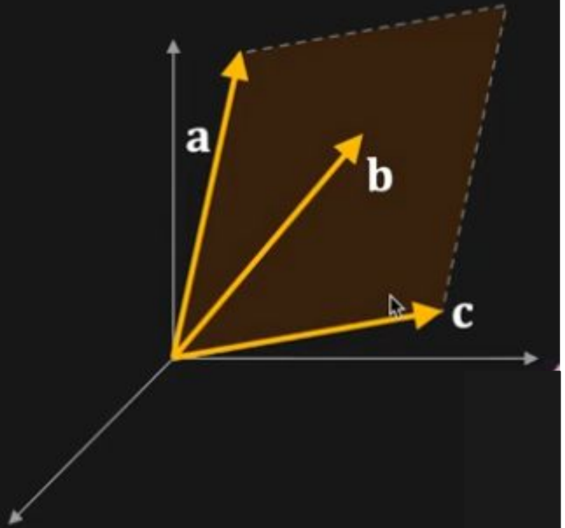


Geometric Meaning of Matrix Rank

Rank is the dimensionality of the space spanned by column or row vectors of the matrix.

$$A = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} = [\mathbf{a} \quad \mathbf{b} \quad \mathbf{c}]$$

$Rank(A) = 2$ ⟺

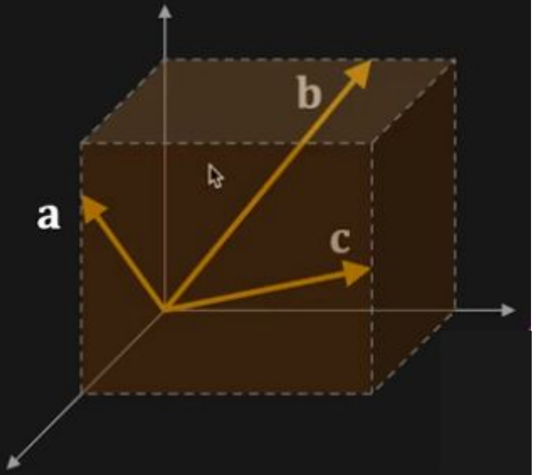


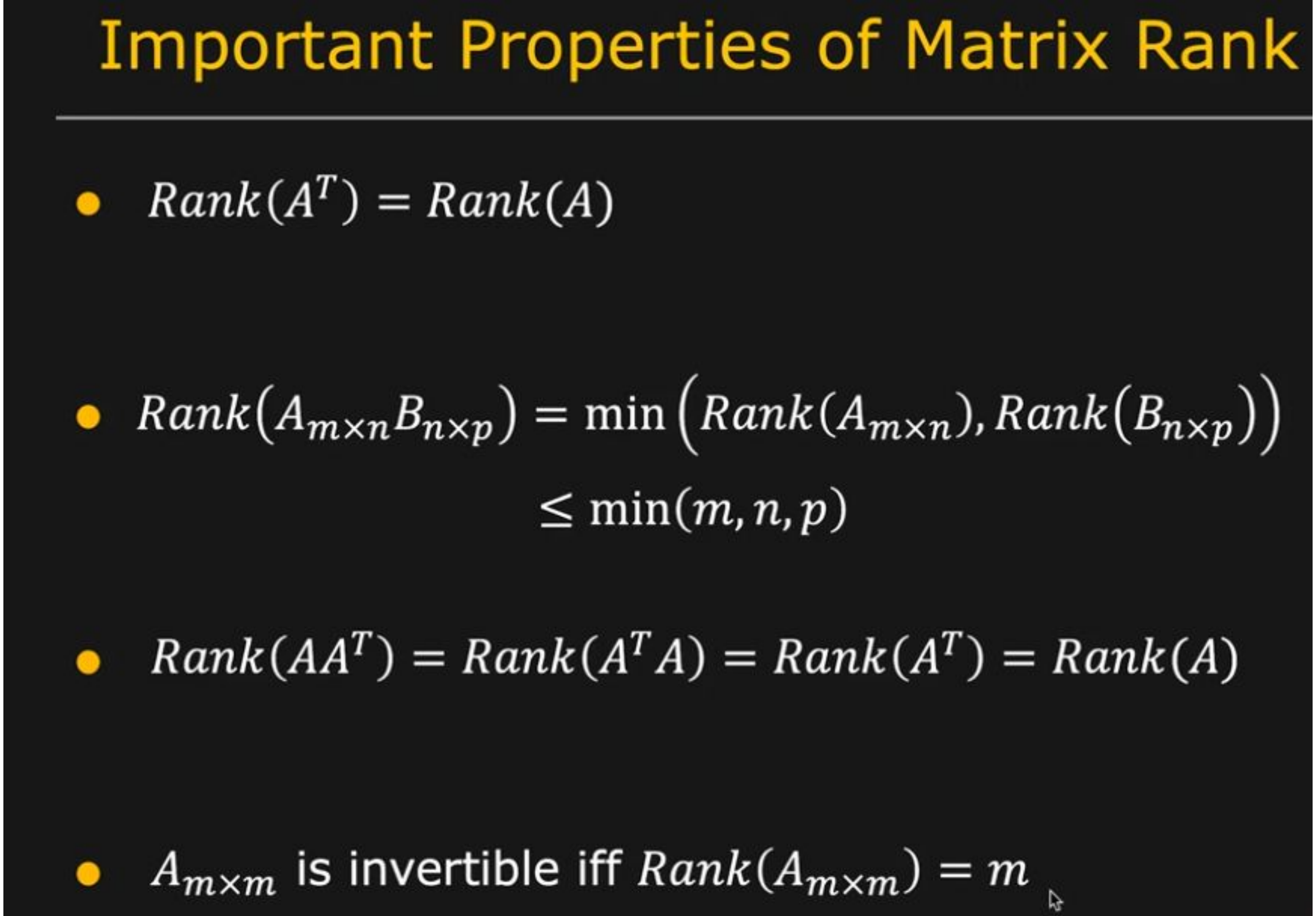
Geometric Meaning of Matrix Rank

Rank is the dimensionality of the space spanned by column or row vectors of the matrix.

$$A = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} = [\mathbf{a} \quad \mathbf{b} \quad \mathbf{c}]$$

$Rank(A) = 3$ ⟺





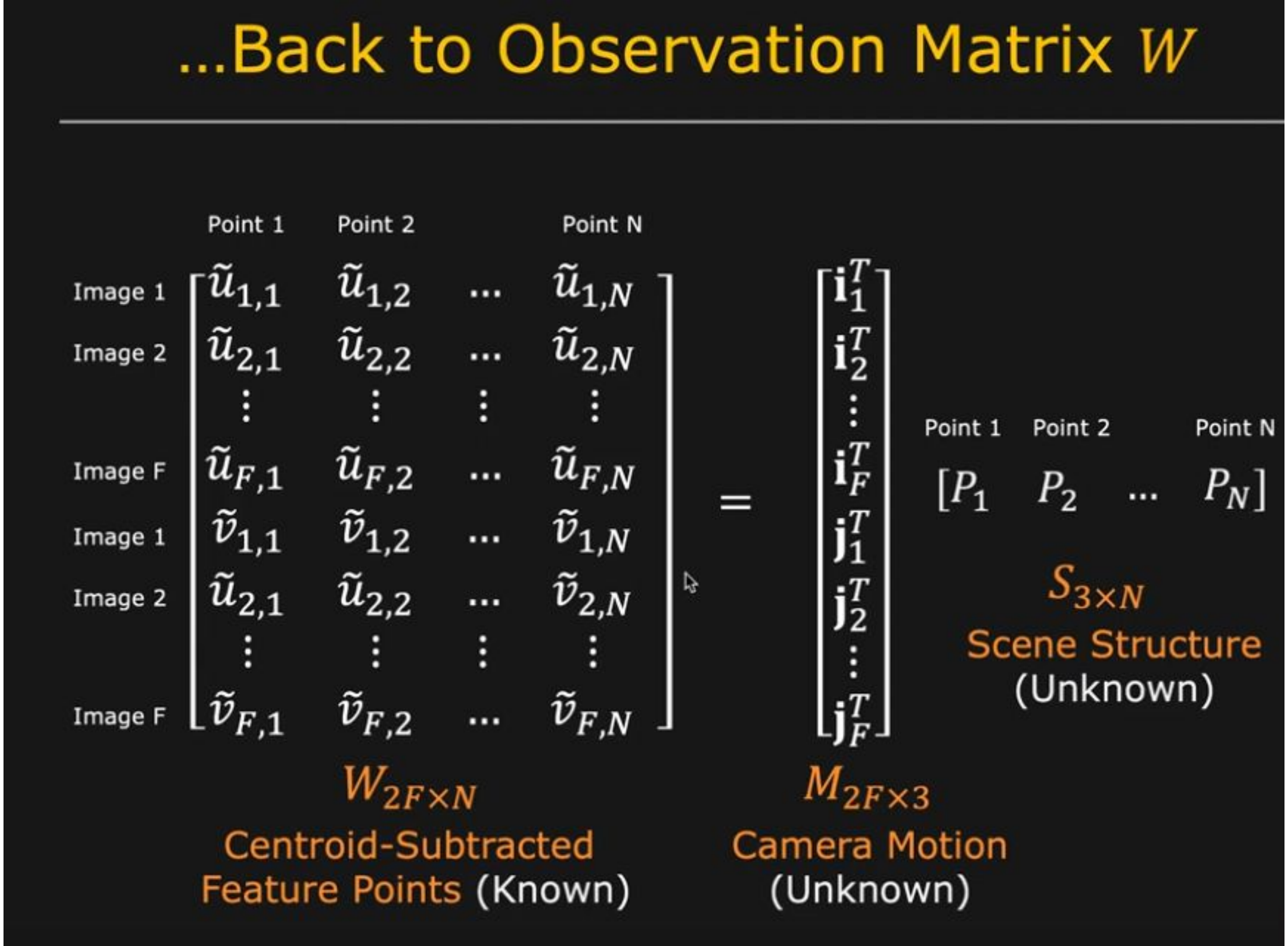
Important Properties of Matrix Rank

- $Rank(A^T) = Rank(A)$

- $$Rank(A_{m\times n}B_{n\times p}) = \min\left(Rank(A_{m\times n}), Rank(B_{n\times p})\right)$$ $$\leq \min(m, n, p)$$

- $Rank(AA^T) = Rank(A^TA) = Rank(A^T) = Rank(A)$

- $A_{m\times m}$ is invertible iff $Rank(A_{m\times m}) = m$



...Back to Observation Matrix W

$$
\begin{array}{rc}
 & \begin{array}{cccc} \text{Point 1} & \text{Point 2} & & \text{Point N} \end{array} \\
\begin{array}{r}
\text{Image 1} \\ \text{Image 2} \\ \\ \text{Image F} \\ \text{Image 1} \\ \text{Image 2} \\ \\ \text{Image F}
\end{array} &
\begin{bmatrix}
\tilde{u}_{1,1} & \tilde{u}_{1,2} & \cdots & \tilde{u}_{1,N} \\
\tilde{u}_{2,1} & \tilde{u}_{2,2} & \cdots & \tilde{u}_{2,N} \\
\vdots & \vdots & \vdots & \vdots \\
\tilde{u}_{F,1} & \tilde{u}_{F,2} & \cdots & \tilde{u}_{F,N} \\
\tilde{v}_{1,1} & \tilde{v}_{1,2} & \cdots & \tilde{v}_{1,N} \\
\tilde{u}_{2,1} & \tilde{u}_{2,2} & \cdots & \tilde{v}_{2,N} \\
\vdots & \vdots & \vdots & \vdots \\
\tilde{v}_{F,1} & \tilde{v}_{F,2} & \cdots & \tilde{v}_{F,N}
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
\mathbf{i}_1^T \\ \mathbf{i}_2^T \\ \vdots \\ \mathbf{i}_F^T \\ \mathbf{j}_1^T \\ \mathbf{j}_2^T \\ \vdots \\ \mathbf{j}_F^T
\end{bmatrix}
\begin{array}{cccc}
\text{Point 1} & \text{Point 2} & & \text{Point N} \\
[P_1 & P_2 & \cdots & P_N]
\end{array}
$$

$W_{2F\times N}$

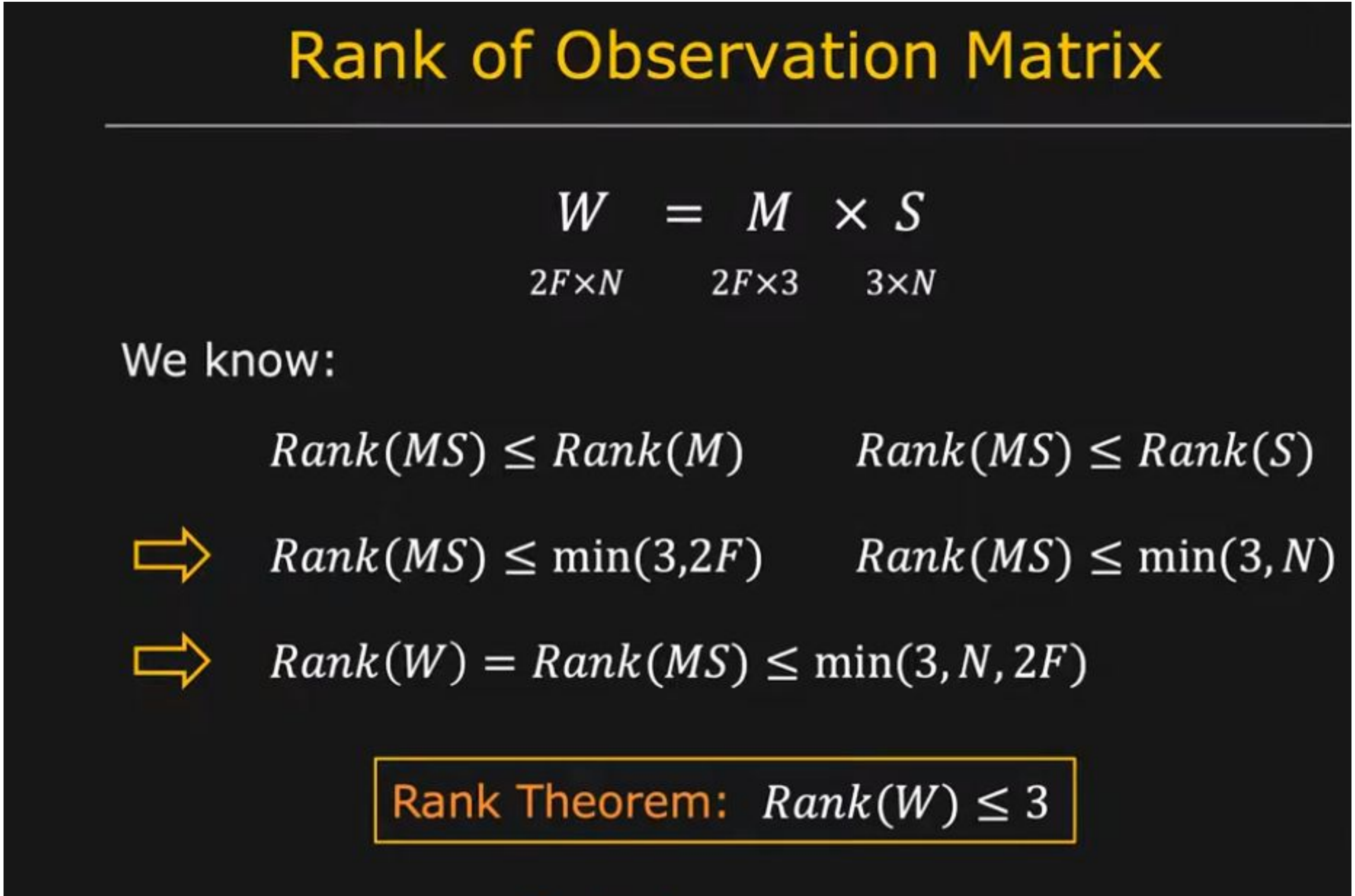
Centroid-Subtracted Feature Points (Known)

$M_{2F\times 3}$

Camera Motion (Unknown)

$S_{3\times N}$

Scene Structure (Unknown)



Rank of Observation Matrix

$$\underset{2F\times N}{W} = \underset{2F\times 3}{M} \times \underset{3\times N}{S}$$

We know:

$$Rank(MS) \leq Rank(M) \qquad Rank(MS) \leq Rank(S)$$

⇨ $$Rank(MS) \leq \min(3{,}2F) \qquad Rank(MS) \leq \min(3, N)$$

⇨ $$Rank(W) = Rank(MS) \leq \min(3, N, 2F)$$

Rank Theorem: $\boxed{Rank(W) \leq 3}$

Singular Value Decomposition (SVD)

For any matrix A there exists a factorization:

$$A_{M\times N} = U_{M\times M} \cdot \Sigma_{M\times N} \cdot {V^T}_{N\times N}$$

where U and V^T are orthonormal and Σ is diagonal.

MATLAB: `[U,S,V] = svd(A)`

$$\Sigma_{M\times N} = \begin{pmatrix} \sigma_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & 0 & \dots & 0 \\ 0 & 0 & 0 & \sigma_4 & \dots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \dots & \sigma_N \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \end{pmatrix}$$

$\sigma_1, \dots, \sigma_N$: Singular Values

If $Rank(A) = r$ then A has r non-zero singular values.

Enforcing Rank Constraint

Using SVD: $W = U\,\Sigma\,V^T$

$$= \underset{2F\times 2F}{\begin{pmatrix} \underset{3}{U_1} & \underset{2F-3}{U_2} \end{pmatrix}} \underset{2F\times N}{\begin{pmatrix} \sigma_1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \end{pmatrix}} \underset{N\times N}{\begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}} \begin{matrix} 3 \\ N-3 \end{matrix}$$

Since $Rank(W) \leq 3$, $Rank(\Sigma) \leq 3$.

Submatrices U_2 and V_2^T do not contribute to W.

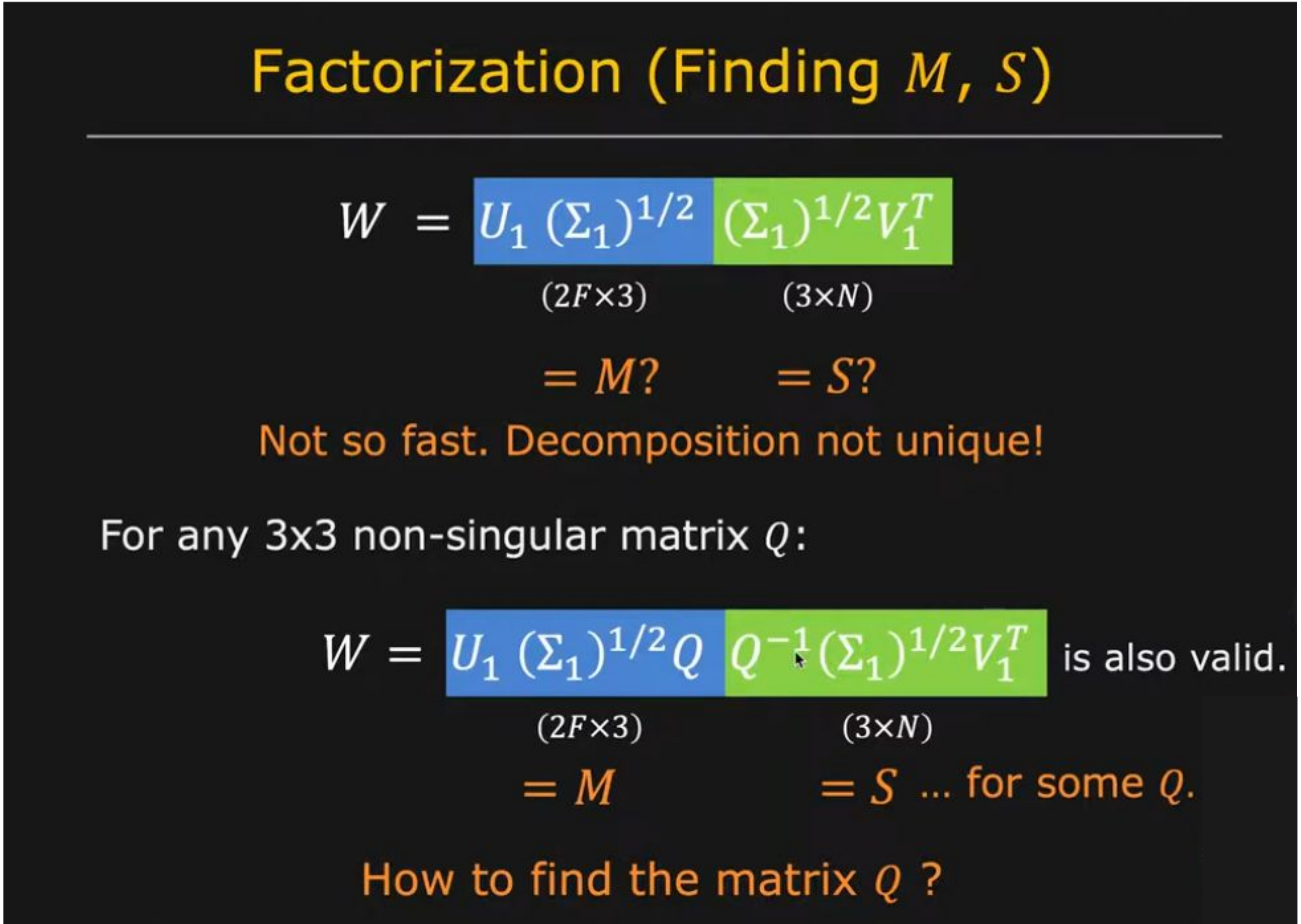
Enforcing Rank Constraint

Using SVD: $W = U \Sigma V^T$

$$= \underset{2F\times 2F}{\begin{pmatrix} \underset{3}{U_1} & \underset{2F-3}{U_2} \end{pmatrix}} \underset{2F\times N}{\begin{pmatrix} \sigma_1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \end{pmatrix}} \underset{N\times N}{\begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}\begin{matrix} 3 \\ N-3 \end{matrix}}$$

$$\boxed{W = U_1 \ \ \Sigma_1 \ \ V_1^T}$$

$$(2F\times 3)(3\times 3)(3\times N)$$



Factorization (Finding M, S)

$$W = \underbrace{U_1 \, (\Sigma_1)^{1/2}}_{(2F\times 3)} \; \underbrace{(\Sigma_1)^{1/2} V_1^T}_{(3\times N)}$$

$= M?$ $= S?$

Not so fast. Decomposition not unique!

For any 3x3 non-singular matrix Q:

$$W = \underbrace{U_1 \, (\Sigma_1)^{1/2} Q}_{(2F\times 3)} \; \underbrace{Q^{-1} (\Sigma_1)^{1/2} V_1^T}_{(3\times N)}$$ is also valid.

$= M$ $= S$ ... for some Q.

How to find the matrix Q ?

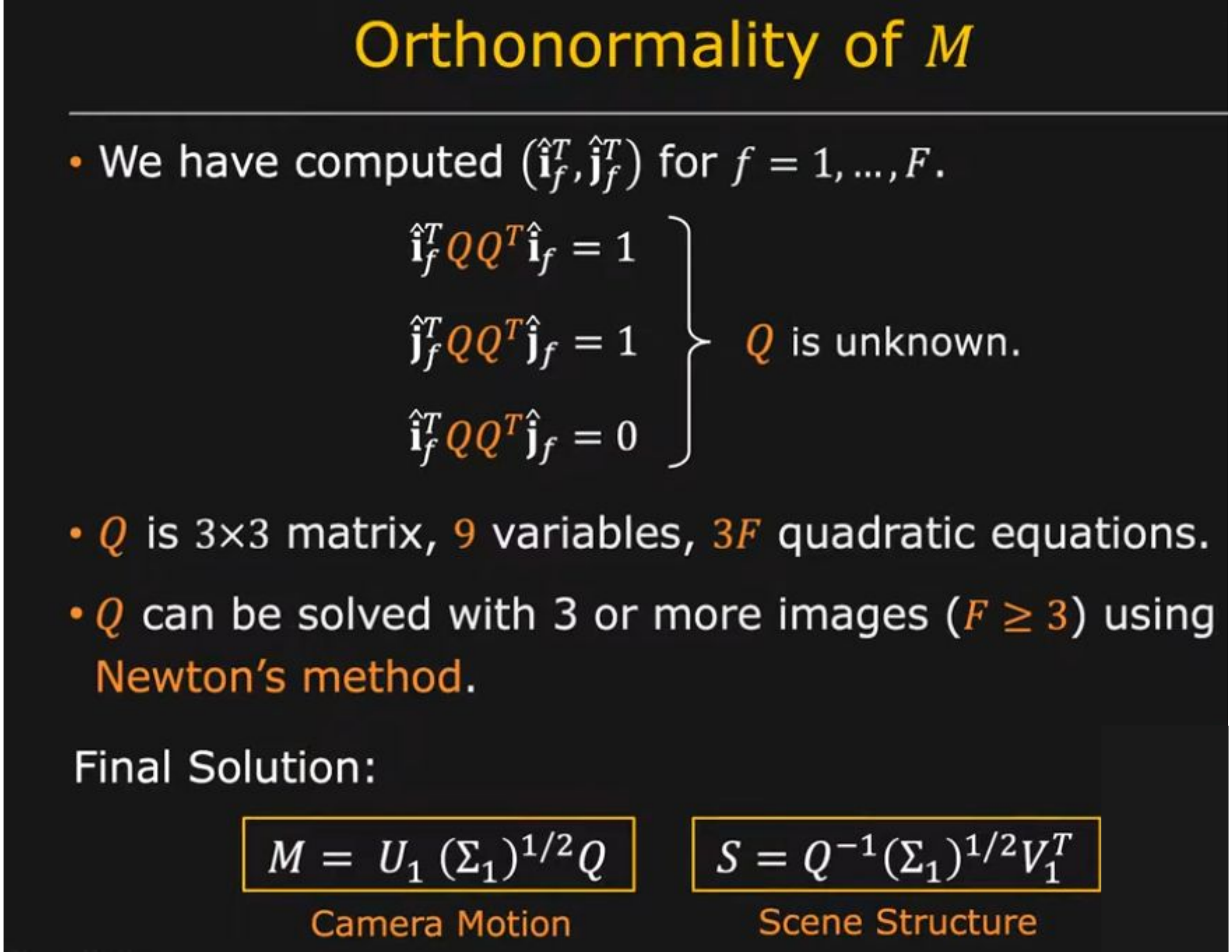
Orthonormality of M

The Motion Matrix M:

$$M = \begin{bmatrix} \mathbf{i}_1^T \\ \vdots \\ \mathbf{i}_F^T \\ \mathbf{j}_1^T \\ \vdots \\ \mathbf{j}_F^T \end{bmatrix} = \underbrace{U_1(\Sigma_1)^{1/2}}_{\text{Computed}} Q = \underbrace{\begin{bmatrix} \hat{\mathbf{i}}_1^T \\ \vdots \\ \hat{\mathbf{i}}_F^T \\ \hat{\mathbf{j}}_1^T \\ \vdots \\ \hat{\mathbf{j}}_F^T \end{bmatrix}}_{\text{Computed}} Q = \begin{bmatrix} \hat{\mathbf{i}}_1^T Q \\ \vdots \\ \hat{\mathbf{i}}_F^T Q \\ \hat{\mathbf{j}}_1^T Q \\ \vdots \\ \hat{\mathbf{j}}_F^T Q \end{bmatrix}$$

Orthonormality Constraints:

$$\boxed{\begin{aligned} \mathbf{i}_f \cdot \mathbf{i}_f &= \mathbf{i}_f^T \mathbf{i}_f = 1 \\ \mathbf{j}_f \cdot \mathbf{j}_f &= \mathbf{j}_f^T \mathbf{j}_f = 1 \\ \mathbf{i}_f \cdot \mathbf{j}_f &= \mathbf{i}_f^T \mathbf{j}_f = 0 \end{aligned}} \Rightarrow \boxed{\begin{aligned} \hat{\mathbf{i}}_f^T Q Q^T \hat{\mathbf{i}}_f &= 1 \\ \hat{\mathbf{j}}_f^T Q Q^T \hat{\mathbf{j}}_f &= 1 \\ \hat{\mathbf{i}}_f^T Q Q^T \hat{\mathbf{j}}_f &= 0 \end{aligned}}$$



Orthonormality of M

- We have computed $(\hat{\mathbf{i}}_f^T, \hat{\mathbf{j}}_f^T)$ for $f = 1, \ldots, F$.

$$\left.\begin{aligned} \hat{\mathbf{i}}_f^T Q Q^T \hat{\mathbf{i}}_f &= 1 \\ \hat{\mathbf{j}}_f^T Q Q^T \hat{\mathbf{j}}_f &= 1 \\ \hat{\mathbf{i}}_f^T Q Q^T \hat{\mathbf{j}}_f &= 0 \end{aligned}\right\} \; Q \text{ is unknown.}$$

- Q is 3×3 matrix, 9 variables, $3F$ quadratic equations.
- Q can be solved with 3 or more images ($F \geq 3$) using Newton's method.

Final Solution:

$$\boxed{M = U_1 \left(\Sigma_1\right)^{1/2} Q} \qquad \boxed{S = Q^{-1} \left(\Sigma_1\right)^{1/2} V_1^T}$$

Camera Motion

Scene Structure

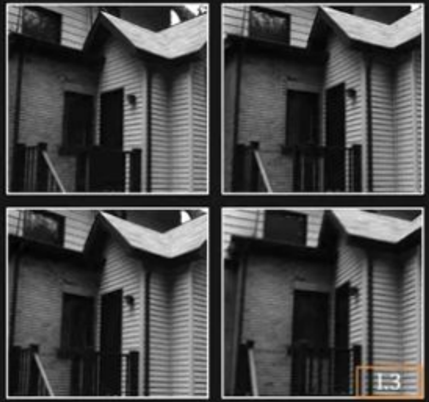
Summary: Orthographic SFM

1. Detect and track feature points.
2. Create the centroid subtracted matrix W of corresponding feature points.
3. Compute SVD of W and enforce rank constraint.

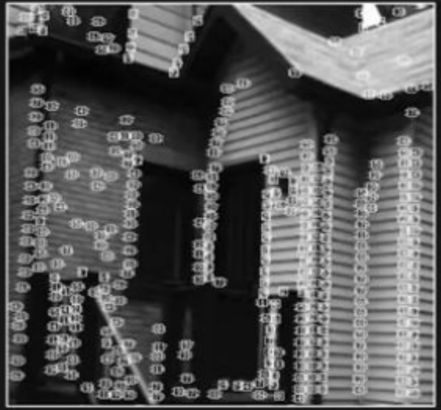
$$W = \quad U\,\Sigma\,V^T \; = \underset{(2F\times 3)}{U_1}\;\underset{(3\times 3)}{\Sigma_1}\;\underset{(3\times N)}{V_1^T}$$

4. Set $M = U_1\,(\Sigma_1)^{1/2}Q$ and $S = Q^{-1}(\Sigma_1)^{1/2}V_1^T$.
5. Find Q by enforcing the orthonormality constraint.

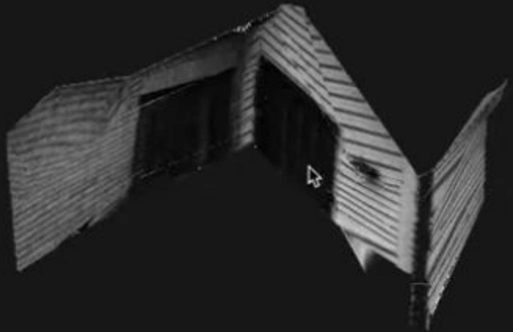
Results



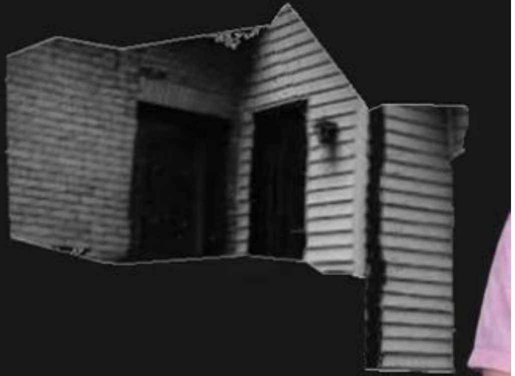
Input Image Sequence



Tracked Features



3D Reconstruction



3D Reconstruction

References

https://www.youtube.com/watch?v=oIvg7sbJRIA&list=PL2zRqk16wsdoYzrWStffqBAoUY8XdvatV&index=7

https://www.youtube.com/watch?v=JIOzyyhk1v0&list=PL2zRqk16wsdoYzrWStffqBAoUY8XdvatV&index=8

https://www.youtube.com/watch?v=Uhkb8Zq-dnM&list=PL2zRqk16wsdoYzrWStffqBAoUY8XdvatV&index=9

https://www.youtube.com/watch?v=Lyd7cf0agvI&list=PL2zRqk16wsdoYzrWStffqBAoUY8XdvatV&index=10

https://www.youtube.com/watch?v=0BVZDyRrYtQ&list=PL2zRqk16wsdoYzrWStffqBAoUY8XdvatV&index=11