Averaging Essential and Fundamental Matrices

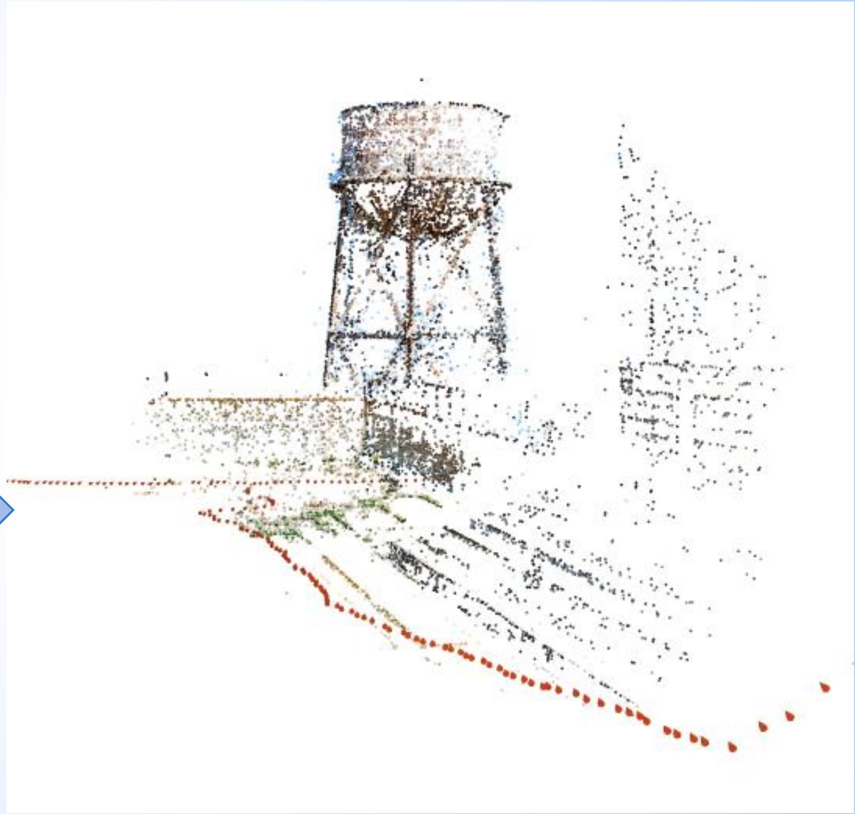
+ Robust Projective Factorization

Ronen Basri

מכון ויצמן למדע

WEIZMANN INSTITUTE OF SCIENCE

Multiview structure from motion

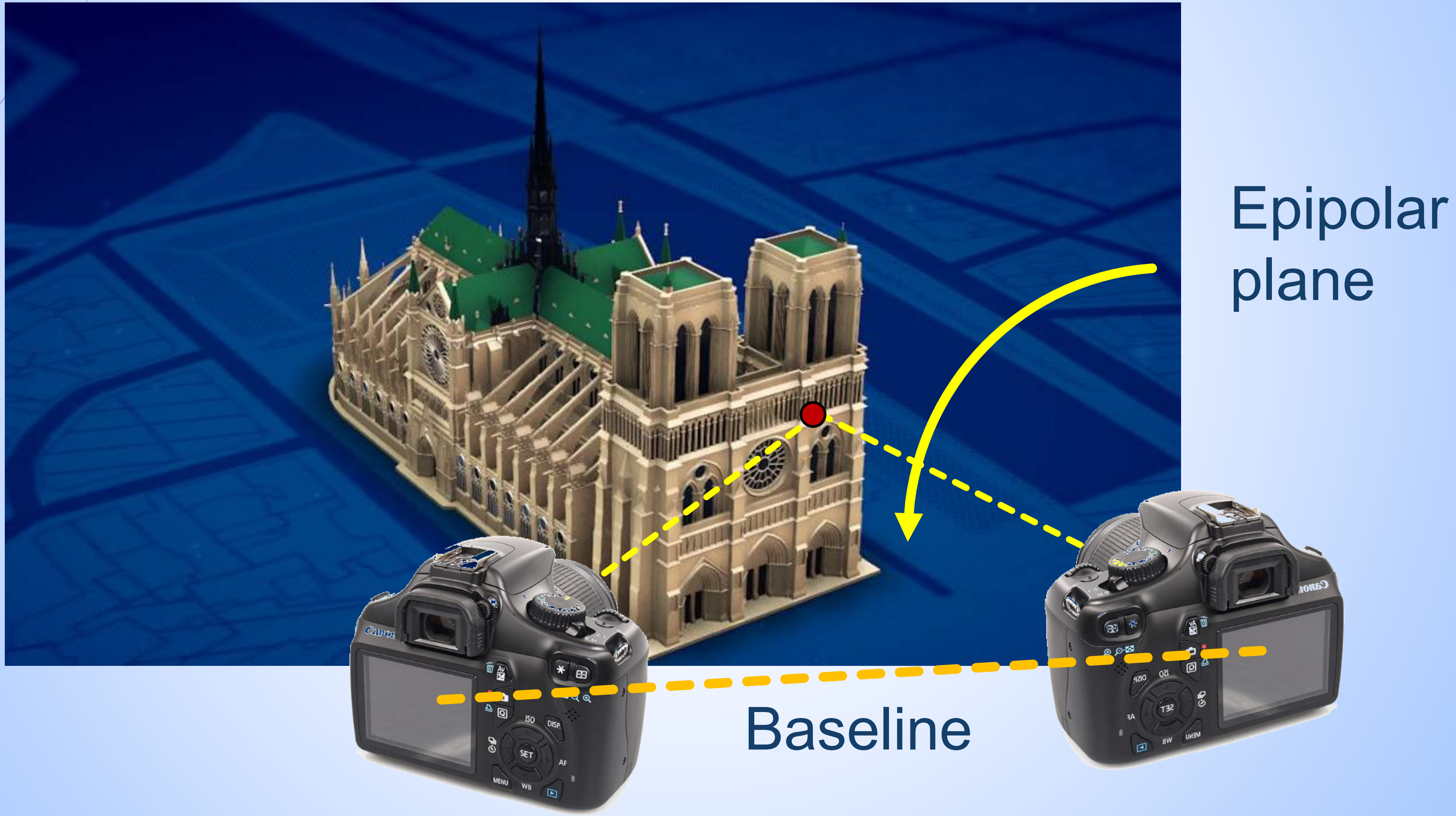


Camera matrices: $P_1, \ldots, P_m$
3D scene points: $X_1, \ldots, X_n$

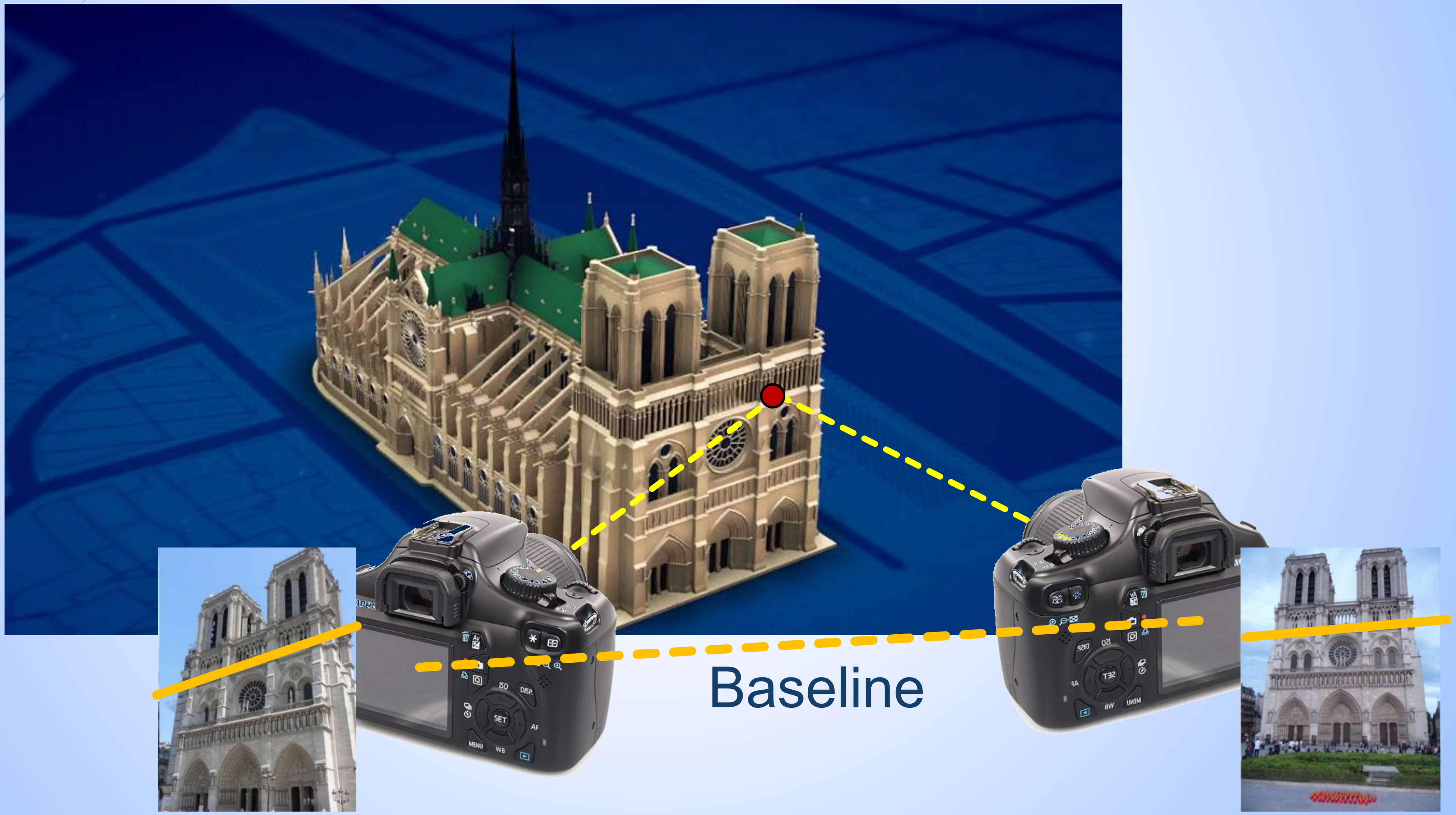
Outline

- Two-view geometry
- Solution to the problem of averaging (aka. synchronization) of two-view, relational camera measurements
- Projective factorization: a NN solution

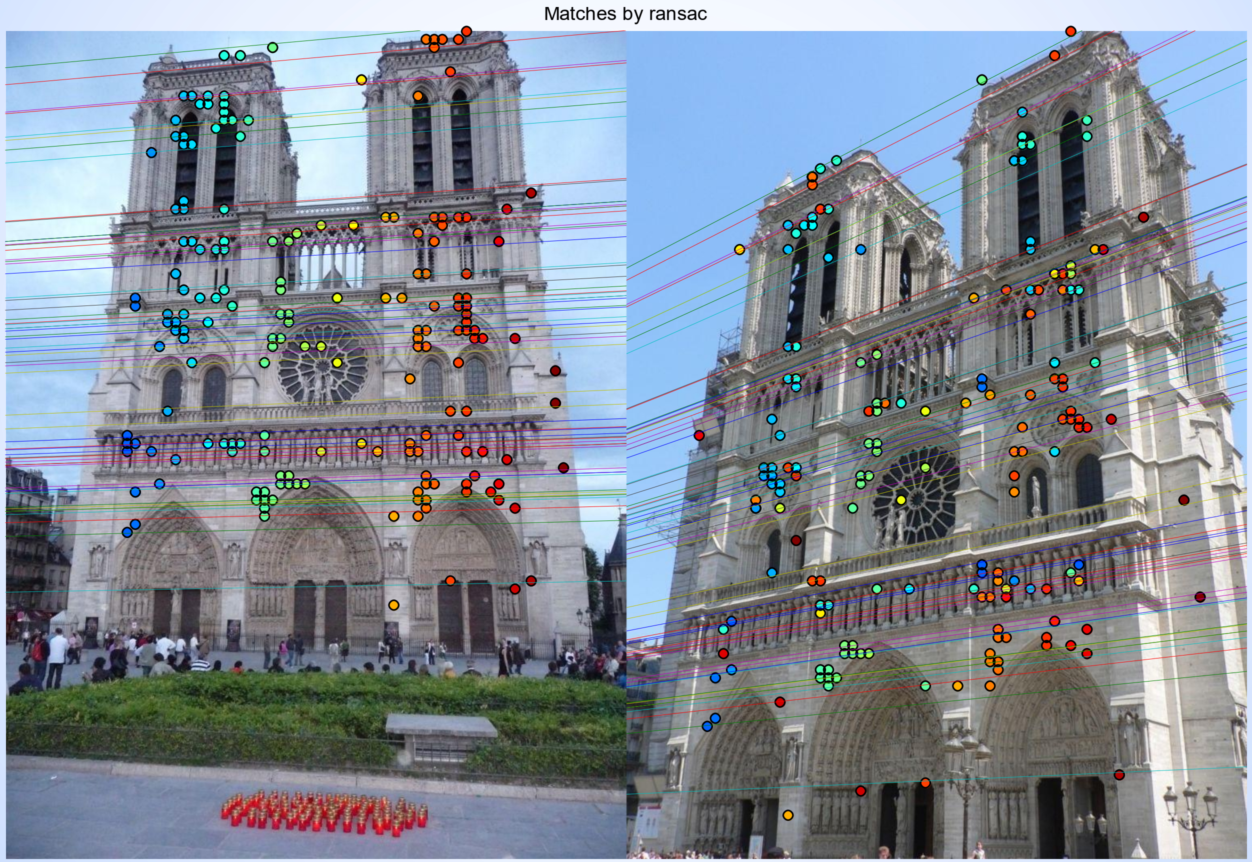
Two-view geometry



Two-view geometry



The essential matrix



$$p^T E q = 0$$

The essential and fundamental matrices

- Internally calibrated images: essential matrix

$$p^T E q = 0$$

- Decomposes to

$$E = RT, \qquad T = [\boldsymbol{t}]_\times$$

- E is 3×3 rank 2 with singular values $\sigma, \sigma, 0$. $\boldsymbol{t}$ can be recovered only up to scale

- Uncalibrated images: fundamental matrix

$$p^T F q = 0$$

- Decomposes to

$$F = K_1^T R T K_2, \qquad T = [\boldsymbol{t}]_\times$$

- F is 3×3 rank 2. $\boldsymbol{t}$ can be recovered only up to scale

Sequential methods

- Photo-Tourism → Building Rome in a Day → COLMAP

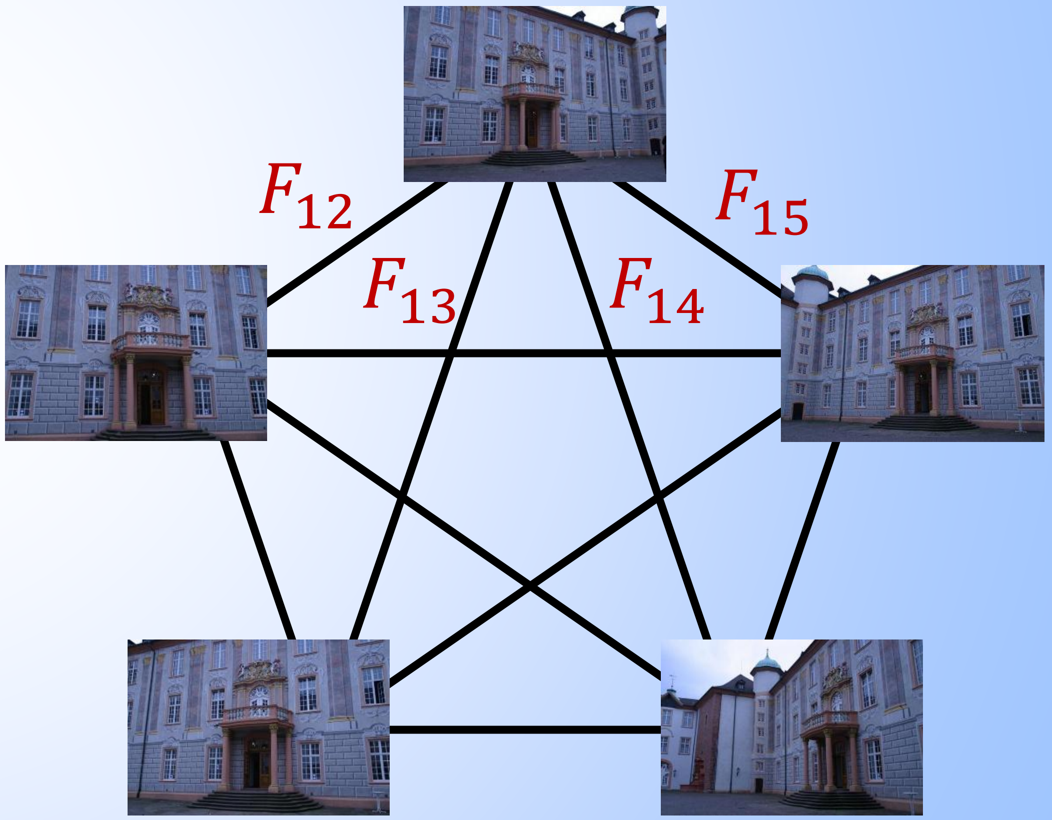
  (Snavely et al. 2006, Agarwal et al. 2009, Schöenberger et al. 2016)

- Feature point detection and matching: SIFT (Lowe, 1999)
- Essential/Fundamental matrix estimation (Longuet-Higgins, 1981)
- RANSAC (Fischler and Bolles, 1981)
- Perspective-n-Point (PnP) (Fischler and Bolles, 1981)
- Bundle adjustment (BA)

- Sequential NNs: CUT3R (Wang et al., 2025)

Averaging techniques

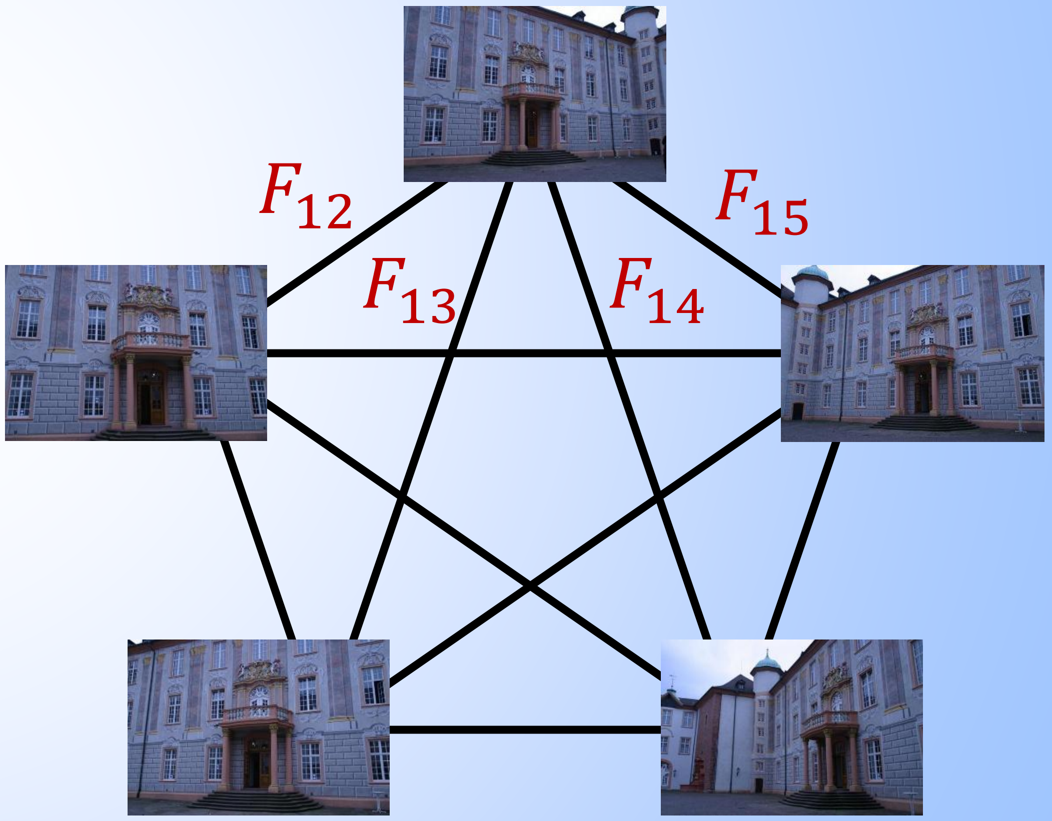
- Recover global camera poses from relative pairwise camera measurements
- Rotation averaging (Govindu 2004; Martinec and Padjla 2007)
- Translation + scale (Ozyesil and Singer 2015)
- Glomap (Pan et al. 2024)



Averaging techniques

- Recover global camera poses from relative pairwise camera measurements
- Rotation averaging (Govindu 2004; Martinec and Padjla 2007)
- Translation + scale (Ozyesil and Singer 2015)
- Glomap (Pan et al. 2024)

Can we directly average essential/fundamental matrices?



Averaging essential and fundamental matrices

- Arie-Nachimson, Kovalsky, Kemelmacher-Shlizerman, Singer, Basri: Global motion estimation from point matches. 3DPVT 2012
- Sengupta, Amir, Galun, Goldstein, Jacobs, Singer, Basri. A new rank constraint on multiview fundamental matrices, and its application to camera location recovery. CVPR 2017
- Kasten, Geifman, Galun, Basri: Algebraic characterization of essential matrices and their averaging in multiview settings. ICCV 2019
- Kasten, Geifman, Galun, Basri: GPSFM: Global projective SFM using algebraic constraints on multi-view fundamental matrices. CVPR 2019
- Geifman, Kasten, Galun, Basri: Averaging essential and fundamental matrices in collinear camera settings. CVPR 2020
- Khatib, Galun, Basri: Learning global camera poses from noisy view-graphs for structure from motion. ECCV 2026

Essential matrices in a canonical frame

- Essential Matrix decomposes to the relative motion ($P_i = [I \ 0], P_j = [R_{ij} \ T_{ij}]$)

$$E_{ij} = R_{ij}T_{ij}$$

Essential matrices in a canonical frame

- Essential Matrix decomposes to the relative motion $(P_i = [I\ 0], P_j = [R_{ij}\ t_{ij}])$

$$E_{ij} = R_{ij}T_{ij}$$

- Can be expressed with respect to canonical cameras $(P_i = R_i[I\ \ -t_i], P_j = R_j[I\ \ -t_j])$

$$E_{ij} = R_i^T(T_i - T_j)R_j$$

Essential matrices in a canonical frame

- Essential Matrix decomposes to the relative motion

$$E_{ij} = R_{ij}T_{ij}$$

- Can be expressed with respect to canonical cameras

$$E_{ij} = R_i^T(T_i - T_j)R_j$$

- $R_i^T T_i R_j$ with $i, j \in [n]$:

$$\mathrm{A} = \begin{bmatrix} R_1^T T_1 \\ \vdots \\ R_n^T T_n \end{bmatrix}_{3n\times 3} \begin{bmatrix} R_1 & \dots & R_n \end{bmatrix}_{3\times 3n}$$

Essential matrices in a canonical frame

- Essential Matrix decomposes to the relative motion

$$E_{ij} = R_{ij}T_{ij}$$

- Can be expressed with respect to canonical cameras

$$E_{ij} = R_i^T\left(T_i - T_j\right)R_j$$

- $R_i^T T_i R_j$ with $i, j \in [n]$:

$$\mathrm{A} = \begin{bmatrix} R_1^T T_1 \\ \vdots \\ R_n^T T_n \end{bmatrix}_{3n\times 3} \begin{bmatrix} R_1 & \dots & R_n \end{bmatrix}_{3\times 3n}$$

- Therefore, for n views, E is rank ≤ 6

$$E = A + A^T = \begin{bmatrix} 0 & E_{12} & \cdots & E_{1n} \\ E_{21} & 0 & & E_{2n} \\ \vdots & & \ddots & \vdots \\ E_{n1} & E_{n2} & \dots & 0 \end{bmatrix}_{3n\times 3n}$$

Fundamental matrices in a canonical frame

- Fundamental Matrix decomposes to the relative motion

$$F_{ij} = K_i^T R_{ij} T_{ij} K_j$$

- Can be expressed with respect to canonical cameras

$$F_{ij} = K_i^T R_i^T (T_i - T_j) R_j K_j$$

- $R_i^T T_i R_j$ with $i, j \in [n]$:

$$\mathrm{A} = \begin{bmatrix} K_1^T R_1^T T_1 \\ \vdots \\ K_n^T R_n^T T_n \end{bmatrix}_{3n\times 3} \quad [R_1 K_1, \quad \dots, \quad R_n K_n]_{3\times 3n}$$

- Therefore, for n views, F is rank ≤ 6

$$F = A + A^T = \begin{bmatrix} 0 & F_{12} & \cdots & F_{1n} \\ F_{21} & 0 & & F_{2n} \\ \vdots & & \ddots & \vdots \\ F_{n1} & F_{n2} & \dots & 0 \end{bmatrix}_{3n\times 3n}$$

Scales

Note: we assume all fundamental/essential matrices are scaled properly according to the canonical frame

So $\hat{F}_{ij} \propto F_{ij}$ up to measurement error,

and Rank $\begin{bmatrix} 0 & \hat{F}_{12} & \cdots & \hat{F}_{1n} \\ \hat{F}_{21} & 0 & & \hat{F}_{2n} \\ \vdots & & \ddots & \vdots \\ \hat{F}_{n1} & \hat{F}_{n2} & \dots & 0 \end{bmatrix}$ in general is not 6

Definition: n-view fundamental/essential matrix

$$F = \begin{bmatrix} 0 & F_{12} & \cdots & F_{1n} \\ F_{21} & 0 & & F_{2n} \\ \vdots & & \ddots & \vdots \\ F_{n1} & F_{n2} & \cdots & 0 \end{bmatrix}_{3n \times 3n}$$

- $3n \times 3n$
- Symmetric
- Zero block diagonal
- Each 3×3 off-diagonal block has rank 2

- For essentials, the two non-zero singular values of each 3×3 block are equal

Consistent n-view fundamental matrices

$$F = \begin{bmatrix} 0 & F_{12} & \cdots & F_{1n} \\ F_{21} & 0 & & F_{2n} \\ \vdots & & \ddots & \vdots \\ F_{n1} & F_{n2} & \cdots & 0 \end{bmatrix}_{3n\times 3n}$$

With non-colinear cameras

- An n-view Fundamental matrix is *consistent* <u>if and only if</u>:
  - Rank(F) =6 and
  - 3 positive and 3 negative eigenvalues
    $F = A + A^T$ and Rank(A) =3 $\Leftrightarrow$ $F = XX^T - YY^T$, i.e., difference of two rank 3 PSD matrices
  - Columns of fundamental matrices are full rank (rank 3)

Consistent n-view essential matrices

$$E = \begin{bmatrix} 0 & E_{12} & \cdots & E_{1n} \\ E_{21} & 0 & & E_{2n} \\ \vdots & & \ddots & \vdots \\ E_{n1} & E_{n2} & \dots & 0 \end{bmatrix}_{3n \times 3n}$$

- Additionally,
  - Eigenvalues come in pairs of equal absolute values

$$\lambda_1, \lambda_2, \lambda_3, -\lambda_1, -\lambda_2, -\lambda_3$$

  - The sum of the eigenvector matrices is a block rotation matrix:

$$\frac{1}{\sqrt{2}}(X + Y) = \frac{1}{\sqrt{n}}\begin{bmatrix} R_1^T & R_2^T & \dots & R_n^T \end{bmatrix}^T$$

Averaging fundamental matrices

- Each F_{ij} is estimated only up to scale $\alpha_{ij}\hat{F}_{ij}$
  - We need to estimate up to $\binom{n}{2}$ scale factors

$$\min_{F,\,\alpha_{ij}} \left\| \begin{bmatrix} 0 & F_{12} & \cdots & F_{1n} \\ F_{21} & 0 & & F_{2n} \\ \vdots & & \ddots & \vdots \\ F_{n1} & F_{n2} & \dots & 0 \end{bmatrix} - \begin{bmatrix} 0 & \alpha_{12}\hat{F}_{12} & \cdots & \alpha_{1n}\hat{F}_{1n} \\ \alpha_{21}\hat{F}_{21} & 0 & & \alpha_{2n}\hat{F}_{2n} \\ \vdots & & \ddots & \vdots \\ \alpha_{n1}\hat{F}_{n1} & \alpha_{n2}\hat{F}_{n2} & \dots & 0 \end{bmatrix} \right\|_{Fro}^2$$

s.t., rank(F) = 6, $F = F^T$, $F_{ii} = 0$,

~~3 positive and 3 negative eigenvalues, and F_{ij} is rank 2~~

- Solve with ADMM, allowing for missing data

Dealing with scales

- Each F_{ij} is estimated only up to scale
  - We need to estimate up to $\binom{n}{2}$ scale factors
- Perspective cameras are not affected by scale
  - We are seeking to estimate n cameras

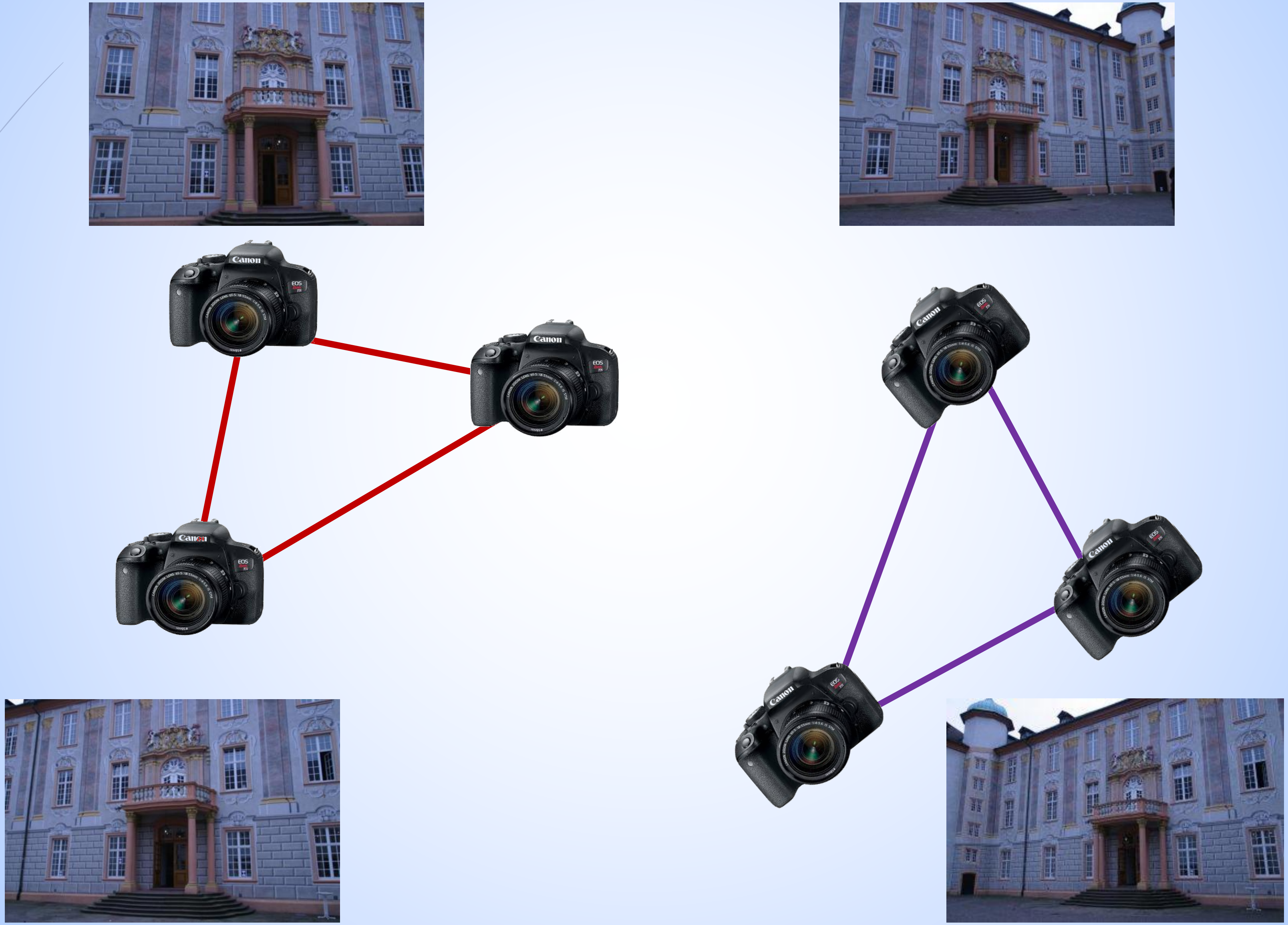
Dealing with scales

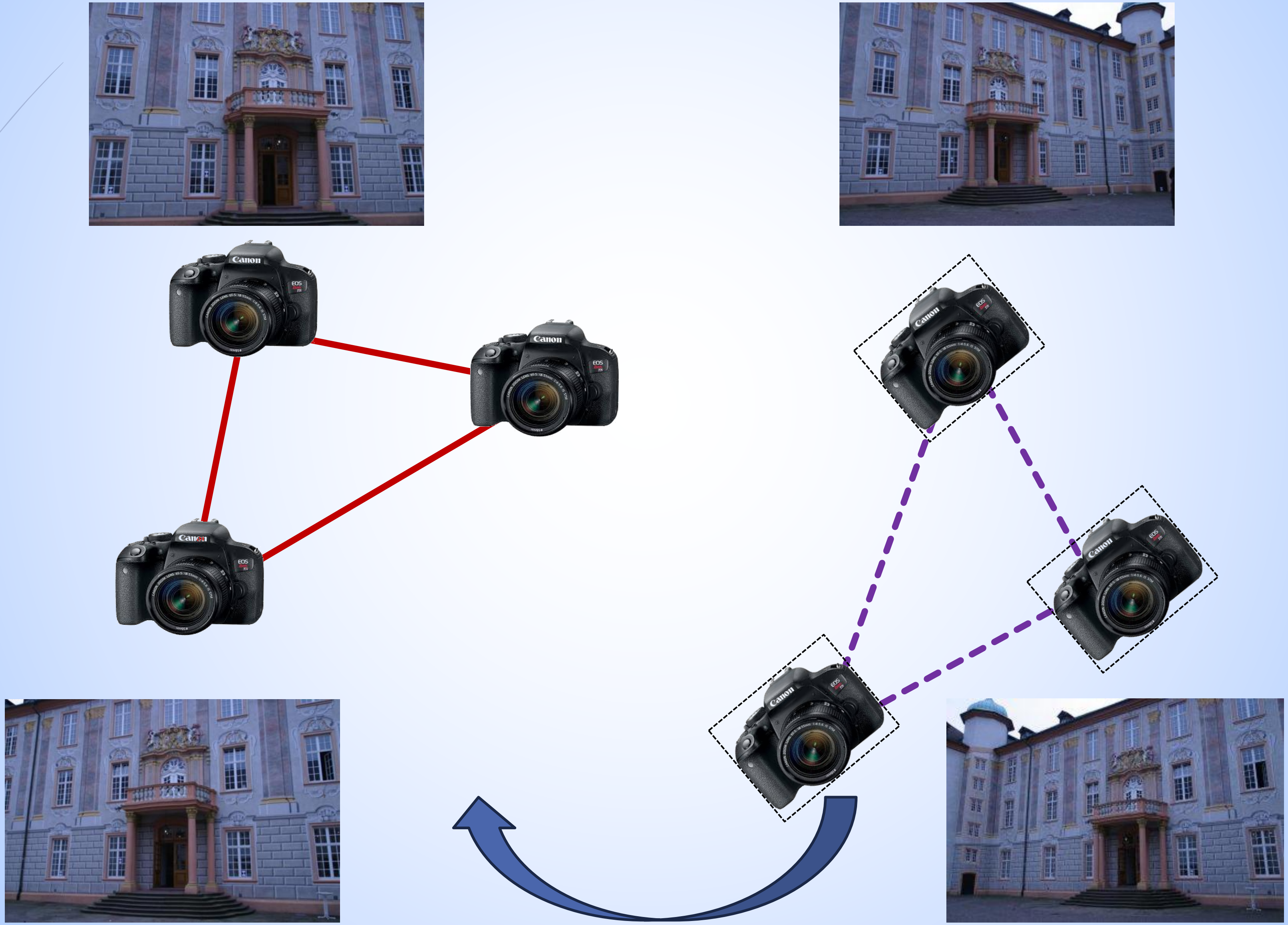
- Each F_{ij} is estimated only up to scale
  - We need to estimate up to $\binom{n}{2}$ scale factors
- Perspective cameras are not affected by scale
  - We are seeking to estimate n cameras
- So, we only need to solve for $\binom{n}{2} - n$ scale factors

Dealing with scales

- Each F_{ij} is estimated only up to scale
  - We need to estimate up to $\binom{n}{2}$ scale factors
- Perspective cameras are not affected by scale
  - We are seeking to estimate n cameras
- So, we only need to solve for $\binom{n}{2} - n$ scale factors

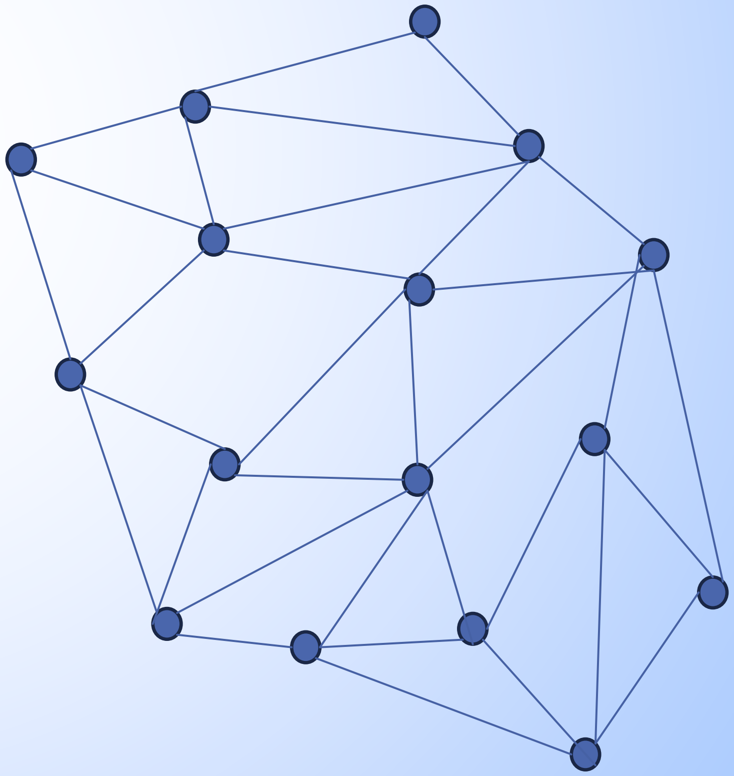
- Note: $\binom{3}{2} - 3 = 0 \Longrightarrow$
  for 3 views, solutions are equivalent with any scale factors





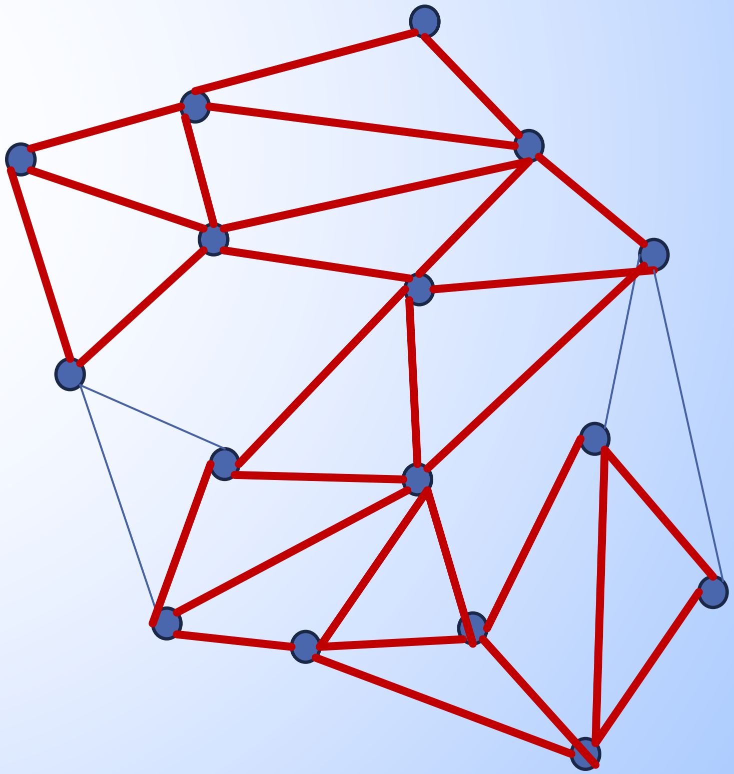
Triangle cover

- Given a view graph



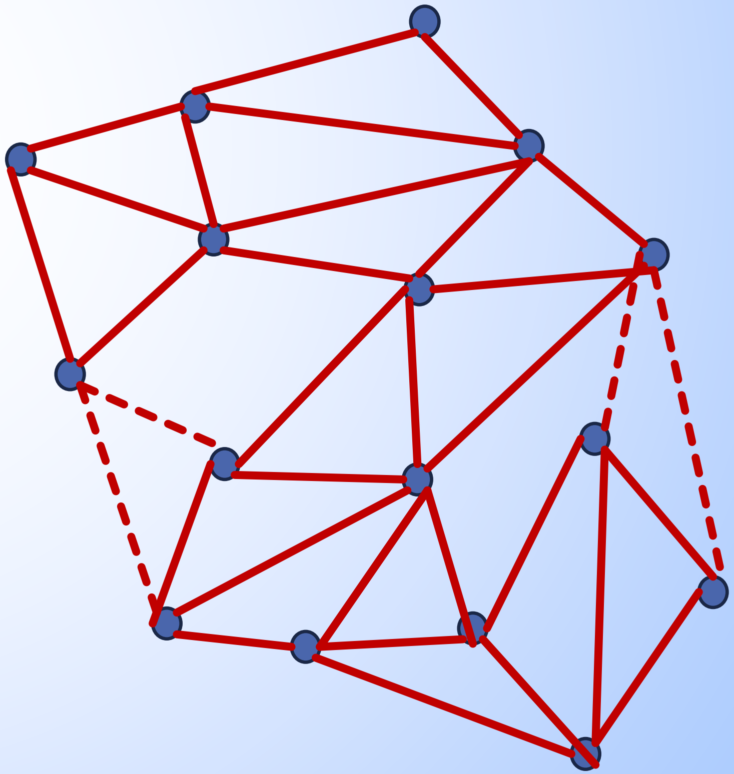
Triangle cover

- Given a view graph
- Cover the nodes with triangles, connected through their edges
- A spanning tree suffices for efficiency



Triangle cover

- Given a view graph
- Cover the nodes with triangles, connected through their edges
- A spanning tree suffices for efficiency
- Additional triangles add redundancy



Algorithm

- Algrithm:
  - enforce rank 6 on camera triplets
  - Global consistency is ensured by the proper triangle cover

$$\min_{F,\alpha_{ij}} \left\| \begin{bmatrix} 0 & F_{12} & \cdots & F_{1n} \\ F_{21} & 0 & & F_{2n} \\ \vdots & & \ddots & \vdots \\ F_{n1} & F_{n2} & \dots & 0 \end{bmatrix} - \begin{bmatrix} 0 & \hat{F}_{12} & \cdots & \hat{F}_{1n} \\ \hat{F}_{21} & 0 & & \hat{F}_{2n} \\ \vdots & & \ddots & \vdots \\ \hat{F}_{n1} & \hat{F}_{n2} & \dots & 0 \end{bmatrix} \right\|_{Fro}^{2}$$

Algorithm

- Algrithm:
  - enforce rank 6 on camera triplets
  - Global consistency is ensured by the proper triangle cover

$$\begin{bmatrix} I & F_{12} & F_{13} & F_{14} & F_{15} & F_{16} \\ F_{21} & I & F_{23} & F_{24} & F_{25} & F_{26} \\ F_{31} & F_{32} & I & F_{34} & F_{35} & F_{36} \\ F_{41} & F_{42} & F_{43} & I & F_{45} & F_{46} \\ F_{51} & F_{52} & F_{53} & F_{54} & I & F_{56} \\ F_{61} & F_{62} & F_{63} & F_{64} & F_{65} & I \end{bmatrix}_{3n \times 3n}$$

- Camera matrices are recovered uniquely from a solution
- **No scale needs to be recovered!**

Optimization

- Solve

$$\min_F \sum_{k=1}^{m} \left\| F - \hat{F} \right\|_{\mathrm{Fro}}^2$$

s.t.

$$F = F^T$$
$$F_{ii} = 0$$
$$\mathrm{rank}\left(F_{\tau(k)}\right) = 6$$

- Solve with ADMM
- Works with vanilla initialization

Optimization: calibrated

- Solve

$$\min_{F} \sum_{k=1}^{m} \left\| E - \hat{E} \right\|_{\mathrm{Fro}}^{2}$$

s.t.

$$\mathrm{rank}\left(E_{\tau(k)}\right) = 6$$

$$\lambda_i\left(E_{\tau(k)}\right) = -\lambda_{-i}\left(E_{\tau(k)}\right)$$

$X\left(E_{\tau(k)}\right) + Y\left(E_{\tau(k)}\right)$ is block-rotation

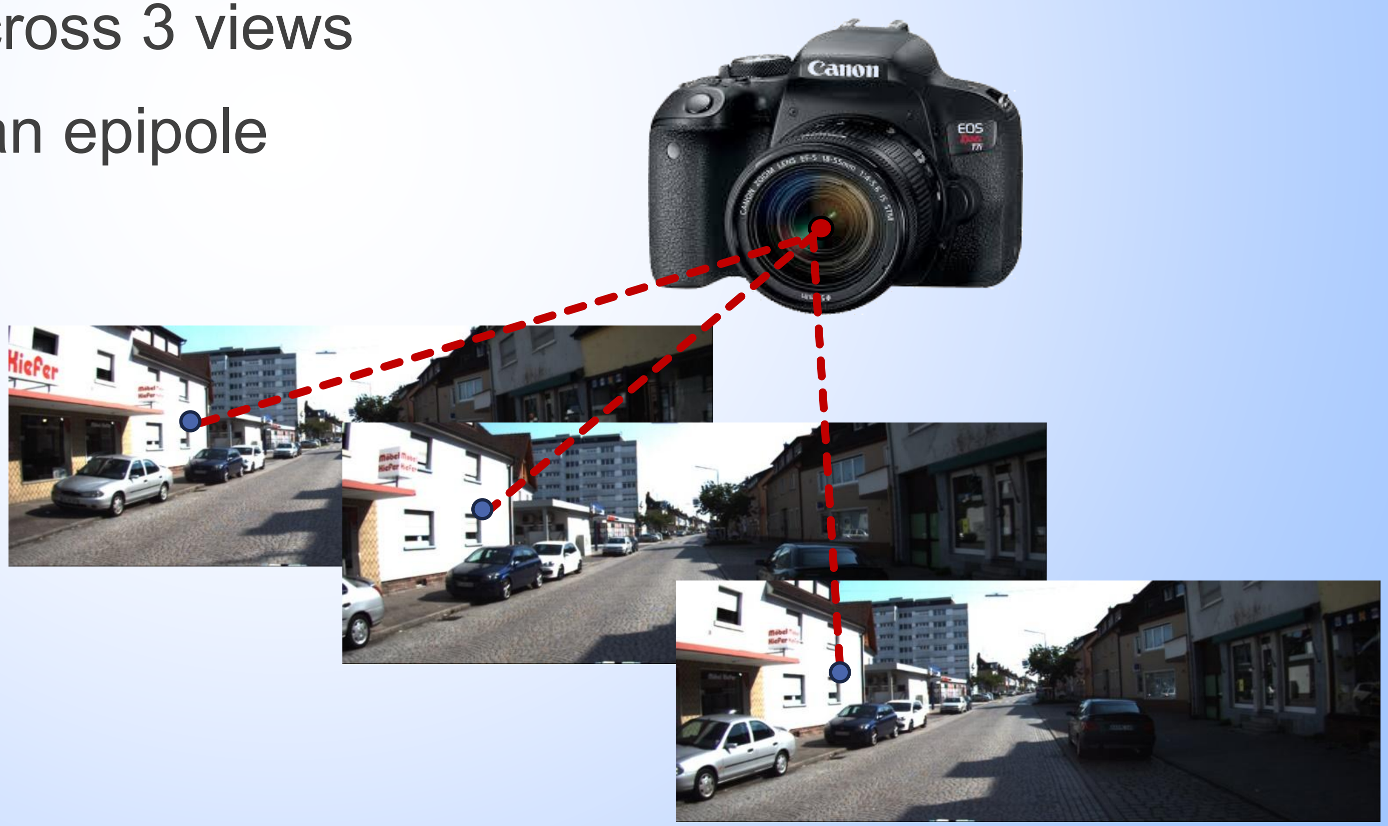
(X, Y eigenvector matrices)

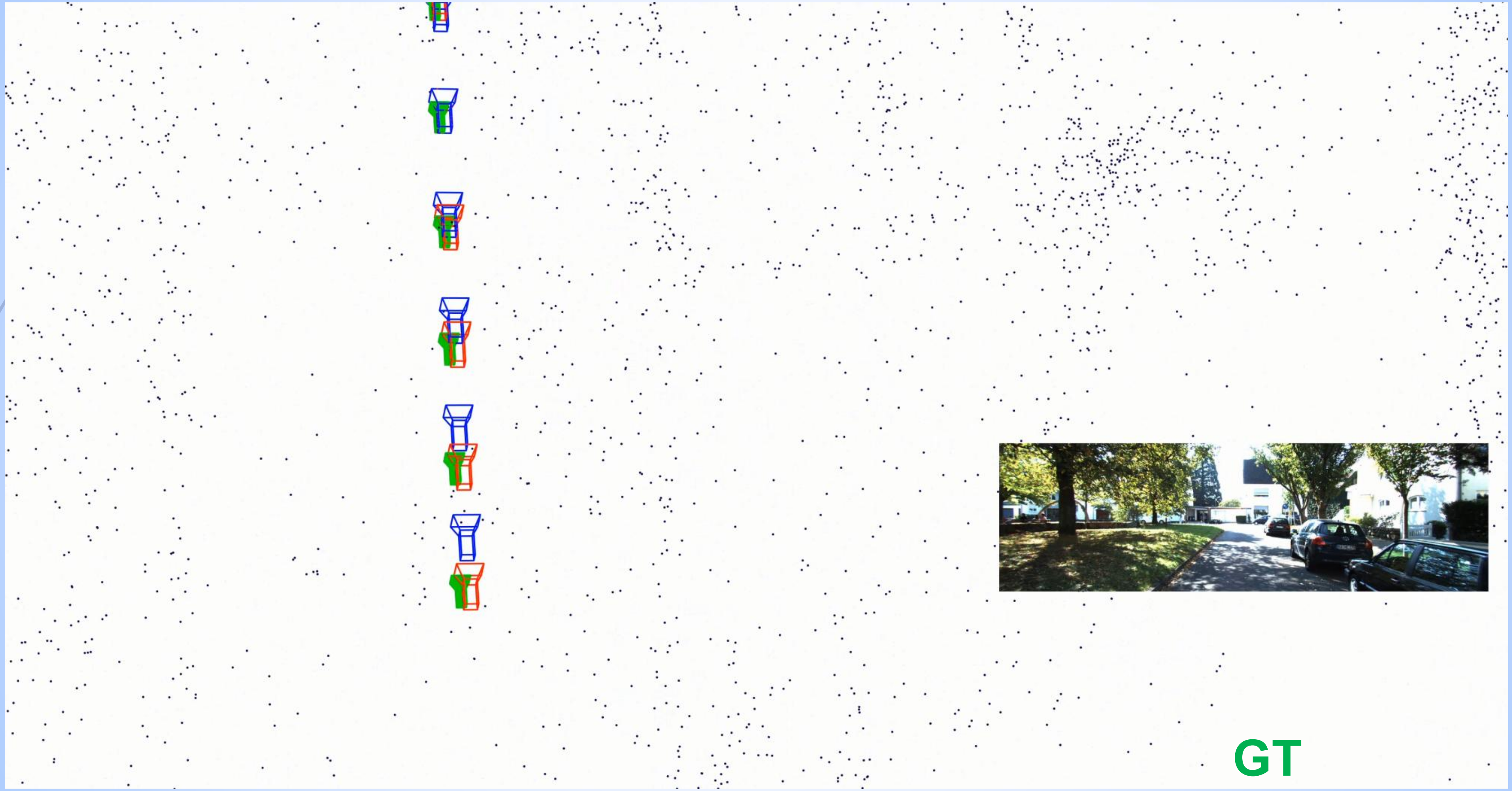
Results

Notre Dame (Montreal): 415 cameras, 161070 points

Collinear cameras

- Collinear cameras introduce a degeneracy, $\mathrm{rank}(F) = 4$
- We can still optimize by adding virtual cameras:
  - Use correspondences across 3 views
  - Each 2D point provides an epipole





GT
LUD
Ours

VGPA: Global averaging with GNN

- Input: a view graph with essential matrices
- Output: camera matrices
- Inference using edge conditioned message passing + robust BA
- Self-supervised training
- Competitive accuracies, faster than classical methods

Scene	N_c	Outliers%	Ours			RESfM			Theia			GLOMAP		
			N_r	Rot	Trans	N_r	Rot	Trans	N_r	Rot	Trans	N_r	Rot	Trans
0238	522	44.6%	486	1.06	0.285	283	2.61	0.325	506	1.21	0.334	497	0.74	0.349
0060	528	41.6%	518	0.10	0.026	503	0.29	0.029	525	0.85	0.124	520	0.11	0.048
0197	870	40.7%	718	0.58	0.108	667	4.22	0.333	855	1.16	0.227	813	0.43	0.130
0094	763	40.1%	659	0.81	0.116	537	3.77	0.750	742	0.75	0.160	711	0.88	3.907
0265	571	38.8%	372	5.30	1.433	346	1.25	0.389	554	5.83	2.216	554	7.46	2.839
0083	635	31.3%	622	0.32	0.062	596	0.64	0.058	632	0.37	0.372	614	0.08	0.016
0076	558	30.5%	547	0.09	0.037	524	0.37	0.094	549	0.78	0.120	540	0.17	0.042
0185	368	30.0%	364	0.12	0.021	350	0.06	0.010	365	0.41	0.094	365	0.16	0.051
0048	512	24.2%	501	0.10	0.011	474	4.69	0.178	507	0.41	0.105	505	0.15	0.224
0024	356	23.0%	328	3.59	1.122	309	2.03	0.398	355	0.56	0.219	338	0.15	0.104
0223	214	17.0%	211	3.45	0.285	204	3.76	0.510	212	3.34	0.519	213	1.75	0.275
5016	28	16.9%	28	0.08	0.015	28	0.12	0.016	28	0.10	0.061	28	0.08	0.046
0046	440	14.6%	438	0.47	0.073	399	0.95	0.043	434	0.25	0.112	440	0.03	0.007

Summary up to here

- Algebraic characterization of n-view fundamental and essential matrices
- Algorithms for averaging fundamental and essential matrices
- Handling colinear cameras via virtual cameras

Projective factorization

- Sturm and Triggs, 1996:

$$\begin{bmatrix} \boldsymbol{x}_{11} & \boldsymbol{x}_{12} & \cdots \\ \boldsymbol{x}_{21} & \boldsymbol{x}_{11} & \\ \vdots & & \end{bmatrix} \rightarrow \begin{bmatrix} z_{11}\boldsymbol{x}_{11} & z_{12}\boldsymbol{x}_{12} & \cdots \\ z_{21}\boldsymbol{x}_{21} & z_{22}\boldsymbol{x}_{11} & \\ \vdots & & \end{bmatrix} \rightarrow \begin{bmatrix} P_1 \\ P_2 \\ \vdots \end{bmatrix} [\boldsymbol{X}_1 \quad \boldsymbol{X}_2 \quad \ldots]$$

Estimate depth values z_{ij} SVD factorization

Repeat iteratively

- Requires good initialization of depth
- Handling missing data complicates the algorithm
- Originally proposed for uncalibrated cameras

Projective factorization

- Perspective rule: $\boldsymbol{x}_{ij} \propto P_i \boldsymbol{X}_j$, in matrix form:

$$\begin{bmatrix} \boldsymbol{x}_{11} & \dots & \boldsymbol{x}_{1n} \\ \vdots & & \vdots \\ \boldsymbol{x}_{m1} & \dots & \boldsymbol{x}_{mn} \end{bmatrix} = \Psi\left(\begin{bmatrix} P_1 \\ \vdots \\ P_m \end{bmatrix} [\boldsymbol{X}_1 \quad \cdots \quad \boldsymbol{X}_n]\right)$$

Projective factorization

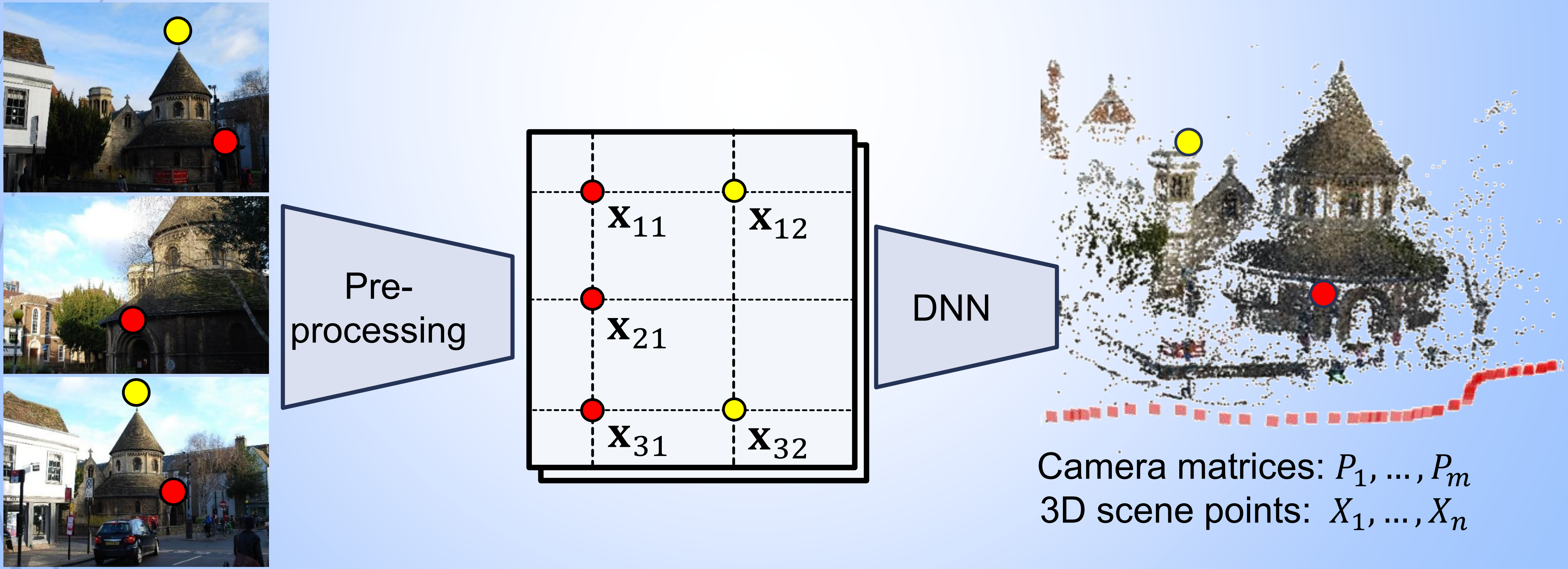
$$\begin{bmatrix} \boldsymbol{x}_{11} & \boldsymbol{x}_{12} & \cdots \\ \boldsymbol{x}_{21} & \boldsymbol{x}_{11} & \\ \vdots & & \end{bmatrix} \dashrightarrow \xcancel{\begin{bmatrix} z_{11}\boldsymbol{x}_{11} & z_{12}\boldsymbol{x}_{12} & \cdots \\ z_{21}\boldsymbol{x}_{21} & z_{22}\boldsymbol{x}_{11} & \\ \vdots & & \end{bmatrix}} \dashrightarrow \begin{bmatrix} P_1 \\ P_2 \\ \vdots \end{bmatrix} [\boldsymbol{X}_1 \quad \boldsymbol{X}_2 \quad \ldots]$$

Neural network?

- Moran, Koslowsky, Kasten, Maron, Galun, B, CVPR 2021
- Khatib, Kasten, Moran, Galun, B, ICLR 2025

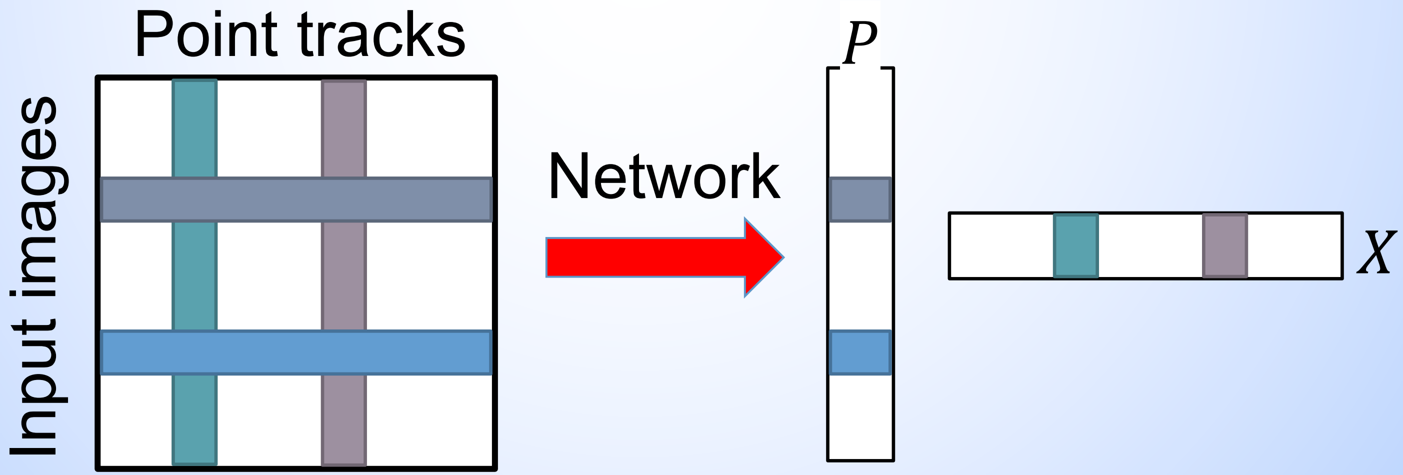
Factorization with DNN

- Preprocess: matching, RANSAC, create point tracks tensor
- DNN: factor tensor to recover cameras and scene points



Deep equivariant network

- We need a network that is equivariant to:
  - permutations of rows (reordering the cameras)
  - permutations of columns (reordering the tracks/3D points)



Deep Sets[2]

- Linear equivariant operations are spanned by 4 operations (Hartford et al., 2018):

Identity | Column sum | Row sum | Matrix sum

$$L(\widetilde{M})_{ij} = W_1 \widetilde{M}_{ij} + W_2 \sum_i \widetilde{M}_{ij} + W_3 \sum_j \widetilde{M}_{ij} + W_4 \sum_{i,j} \widetilde{M}_{ij} + b$$

- Entry-wise activation function maintains equivariance
- Main network: $L_k \circ \sigma \circ L_{k-1} \cdots \circ \sigma \circ L_1$
- Replace sums with averages to handle missing entries

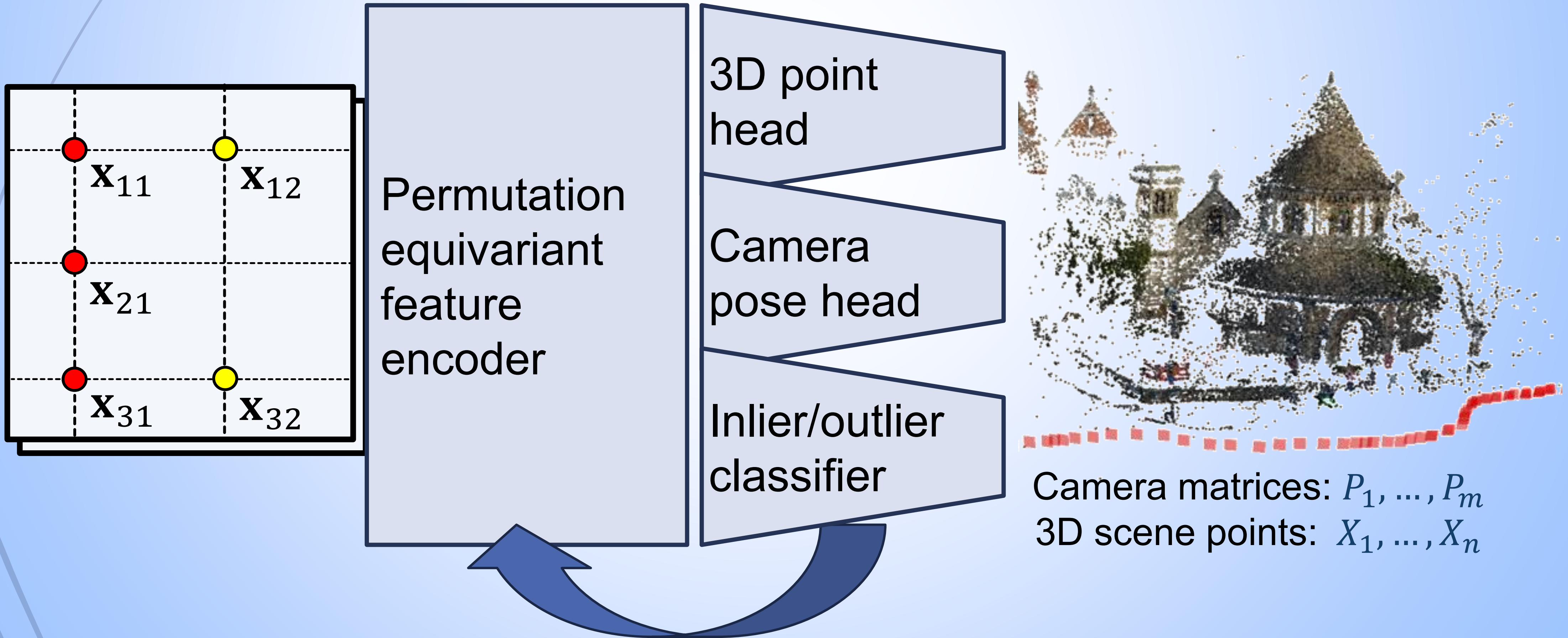
Benefits of equivariance

1. Equivariant architectures encode the structure of the task into the model, providing a strong inductive bias.
2. Parameter sharing yields considerably fewer parameters (number of parameters is independent of matrix size)
3. Improved efficiency
4. Better generalization
5. Handling inputs of variable size and variable missing data patterns
6. Architecture is universal (Ravanbakhsh et al., 2017)

Equivariance can be achieved also with graph transformers (Brynte et al.,2023)

Architecture

- Loss: reprojection + cross entropy (for inlier/outlier)



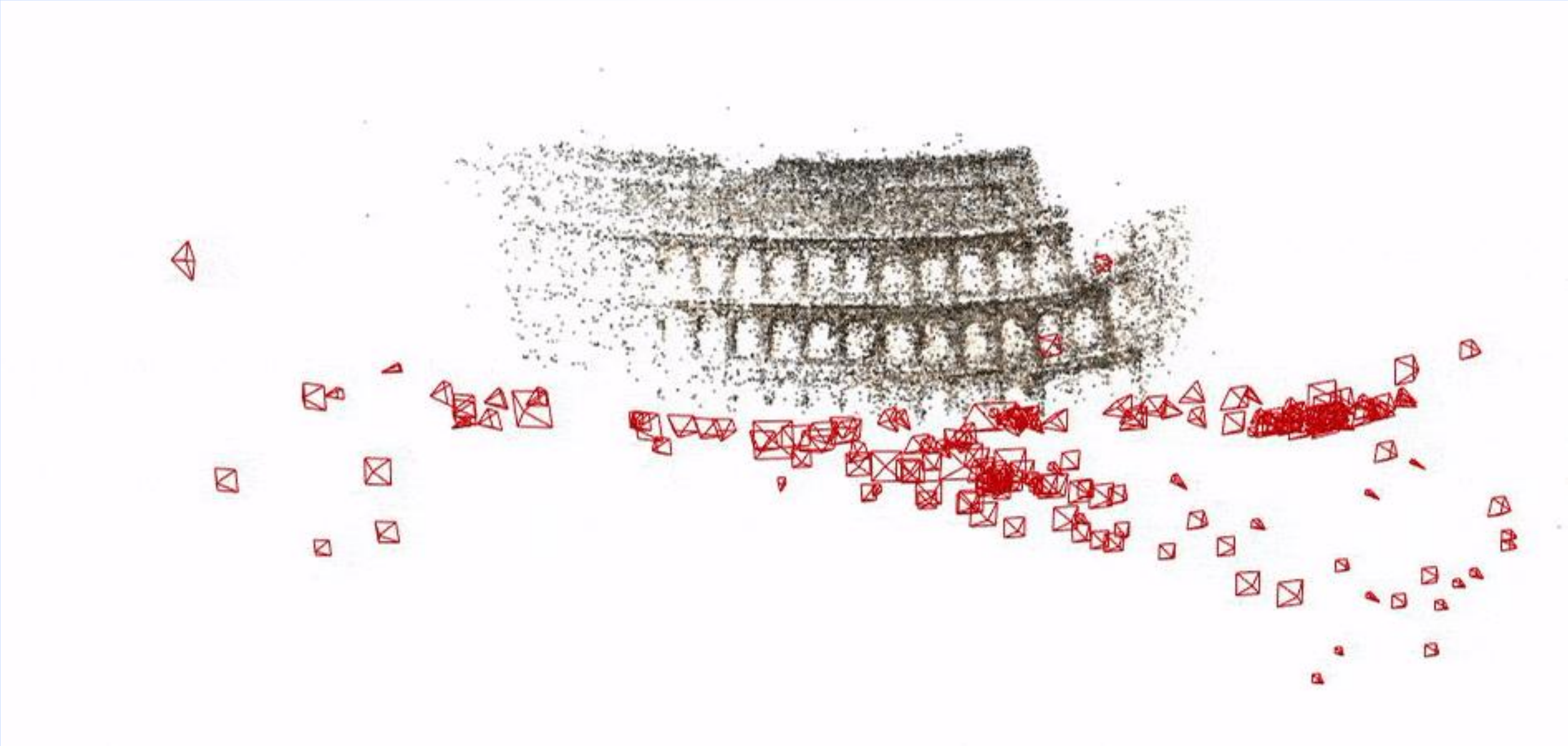
Results

- Feed forward + per-scene finetuning + robust BA
- Works also in single-scene optimization mode
- Competitive results on MegaDepth and 1DSFM datasets
- Some failures with circular camera trajectories

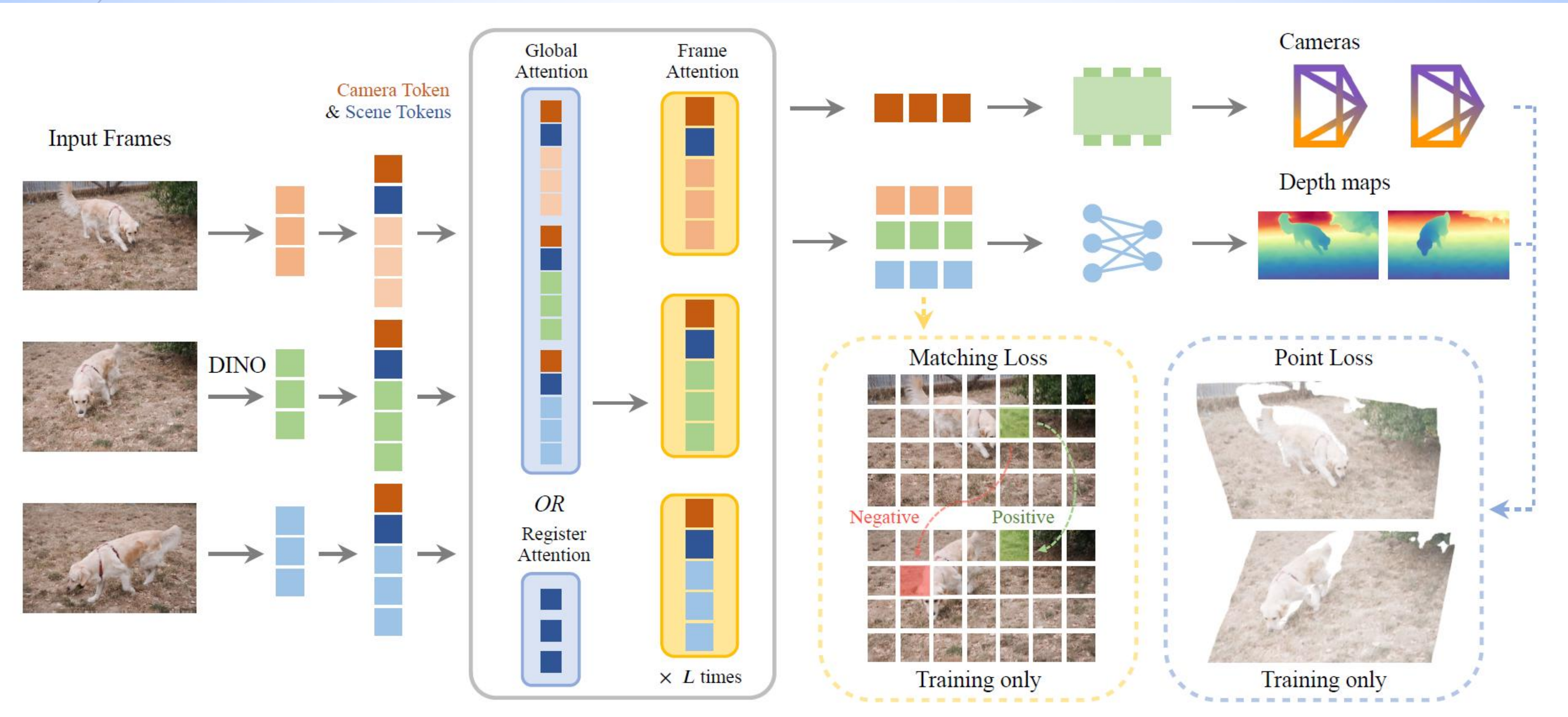
Scene	N_c	Outliers%	Ours		
			N_r	Rot	Trans
0238	522	44.6	283	2.61	**0.325**
0060	528	41.6	503	0.29	**0.029**
0197	870	40.7	667	4.22	0.333
0094	763	40.1	537	3.77	0.750
0265	571	38.8	346	**1.25**	**0.389**
0083	635	31.3	596	0.64	0.058
0076	558	30.5	524	0.37	0.094
0185	368	30.0	350	**0.06**	**0.010**
0048	512	24.2	474	4.69	0.178
0024	356	23.0	309	2.03	0.398
0223	214	17.0	204	3.76	0.510
5016	28	0.2	**28**	0.12	0.016
0046	440	14.6	399	0.95	0.043

Scene	N_c	Outliers%	Ours		
			N_r	Rot	Trans
Alamo	573	32.6	484	3.66	0.515
Ellis Island	227	25.1	214	0.78	**0.138**
Madrid Metropolis	333	39.4	244	8.42	0.827
Montreal Notre Dame	448	31.7	419	1.35	**0.168**
Notre Dame	549	35.6	518	**0.78**	**0.155**
NYC Library	330	33.6	224	3.96	0.429
Piazza del Popolo	336	33.1	249	2.20	**0.186**
Tower of London	467	27.0	94	**0.67**	**0.026**
Vienna Cathedral	824	31.4	483	**0.98**	**0.076**
Yorkminster	432	29.0	331	14.54	1.468

Reconstruction example



VGGT-Ω



Input Frames
DINO
Camera Token
& Scene Tokens
Global Attention
Frame Attention
OR
Register Attention
× L times
Cameras
Depth maps
Matching Loss
Negative
Positive
Training only
Point Loss
Training only

Conclusion

- Algebraic characterizations and algorithms for averaging fundamental and essential matrices
- Equivariant deep network architectures can solve projective factorization
- Future work:
  - Numerical techniques for projective factorization
  - Averaging for general solvable graphs (see Madhavan and Arrigoni 2025)
  - Incorporate structure recovery within fundamental/essential averaging