Lecture 14: CS677

October 5, 2017

Admin

- HW3, due tomorrow, Oct 6, noon5
- Exam 1, October 12: Class period (9-10:50AM)
  - Topic slides will be updated this afternoon
- Class schedule
  - Oct 6, 3:30-5:00 pm, OHE 136 in lieu of Oct 10 class
  - Oct 10, no class
  - Oct 12, Exam1
  - Oct 13, 3:30-5:00 pm, OHE 136, extra class
  - Oct 17, 19 regular class
  - Oct 20, 3:30-5:00 pm, OHE 136, extra class
  - Oct 24, 26: No class
  - Oct 31, resume normal schedule

Review

- Previous class
  - Reconstruction from two calibrated cameras
  - Affine structure from multiple images
- Today's objective
  - Upgrade from affine to Euclidean
  - Projective reconstruction

Affine Ambiguity

- Solution is ambiguous up to an affine transformation

If M_i and P_i are solutions to $\boldsymbol{p}_{ij} = \mathcal{M}_i \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix} = \mathcal{A}_i \boldsymbol{P}_j + \boldsymbol{b}_i$

then so are M'_i and P'_j, where

$$\mathcal{M}'_i = \mathcal{M}_i \mathcal{Q} \quad \text{and} \quad \begin{pmatrix} \boldsymbol{P}'_j \\ 1 \end{pmatrix} = \mathcal{Q}^{-1} \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix}$$

and Q is an arbitrary affine transformation matrix. $\mathcal{Q} = \begin{pmatrix} \mathcal{C} & \boldsymbol{d} \\ \boldsymbol{0}^T & 1 \end{pmatrix}$

where C is a non-singular 3 × 3 matrix and d is a vector in R3

Affine to Euclidean Shape

- Consider *weak perspective* cameras (more restricted than general affine camera)
- Projection matrix can be written as:

$$\mathcal{M} = \frac{1}{Z_r}\begin{pmatrix} k & s \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \mathcal{R}_2 & t_2 \end{pmatrix}$$

- If intrinsic parameters are known, we can set *k* = 1, *s*= 0.

  Rewrite projection matrix as: $\hat{\mathcal{M}} = \begin{pmatrix} \hat{\mathcal{A}} & \hat{b} \end{pmatrix} = \frac{1}{Z_r}\begin{pmatrix} \mathcal{R}_2 & t_2 \end{pmatrix}$

- $\hat{\mathcal{A}}$ is the 2 x 3 matrix formed by the first two rows of rotation matrix R. It follows its row vectors $\hat{a}_1^T$ and $\hat{a}_2^T$ are orthogonal to each other and have the same norm, i.e

$$\hat{a}_1 \cdot \hat{a}_2 = 0 \quad \text{and} \quad ||\hat{a}_1||^2 = ||\hat{a}_2||^2.$$

- We want to find matrix $\hat{\mathcal{M}}_i = \mathcal{M}_i \mathcal{Q}$ where $\mathcal{Q} = \begin{pmatrix} \mathcal{C} & \boldsymbol{d} \\ \mathbf{0}^T & 1 \end{pmatrix}$ such that above equations hold.

  Only constraint on C is that it is a non-singular 3 x 3 matrix

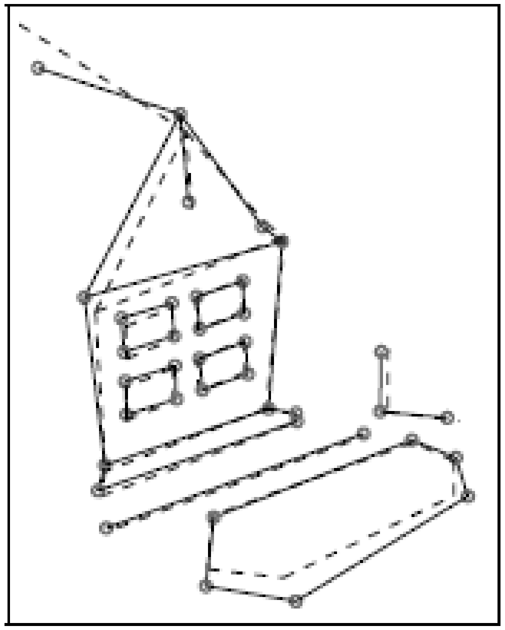
Euclidean Upgrade: continued

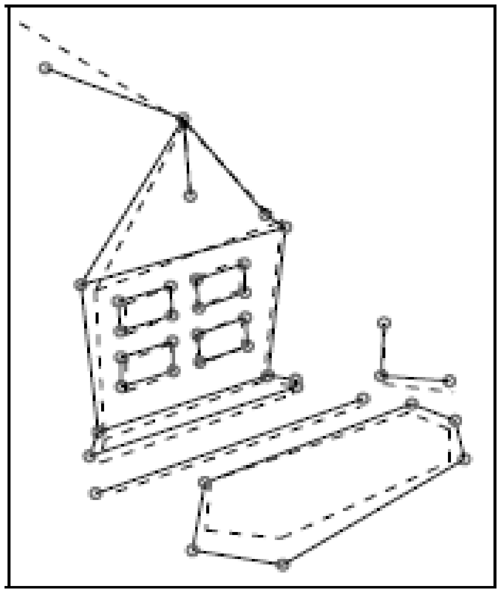
- This results in following equations

$$\begin{cases} \hat{a}_{i1} \cdot \hat{a}_{i2} = 0, \\ \|\hat{a}_{i1}\|^2 = \|\hat{a}_{i2}\|^2, \end{cases} \Longleftrightarrow \begin{cases} a_{i1}^T \mathcal{C}\mathcal{C}^T a_{i2} = 0, \\ a_{i1}^T \mathcal{C}\mathcal{C}^T a_{i1} = a_{i2}^T \mathcal{C}\mathcal{C}^T a_{i2}, \end{cases} \quad \text{for} \quad i = 1, \dots, m,$$

- Let matrix $D = CC^T$ (D is not same as data matrix defined earlier; **another bad choice of notation** in the book).
- C is 3x3, so D is also 3 x 3
  - 8 independent parameters in D; two equations per camera
- Can be solved given four or more cameras by linear least squares
- Compute C ("square root" of D) by "*Cholesky decomposition*"
  - Requires D to be positive definite, may not hold with noise
- ***d*** in matrix Q can be set to 0
- New point positions can be computed by multiplying computed point positions by Q^{-1}.

Results (Figs 8.8 and 8.9 from FP book)

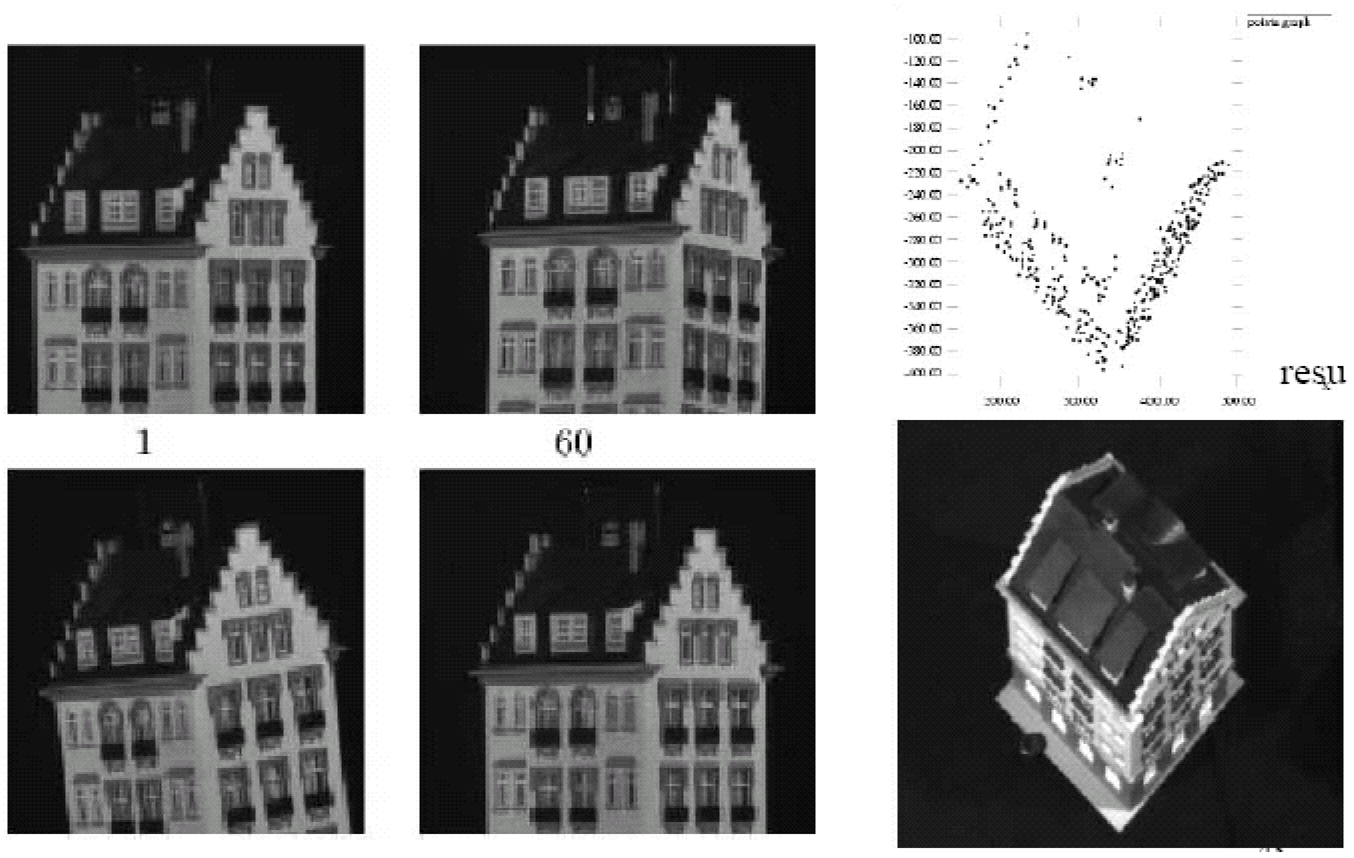

Affine


Euclidean

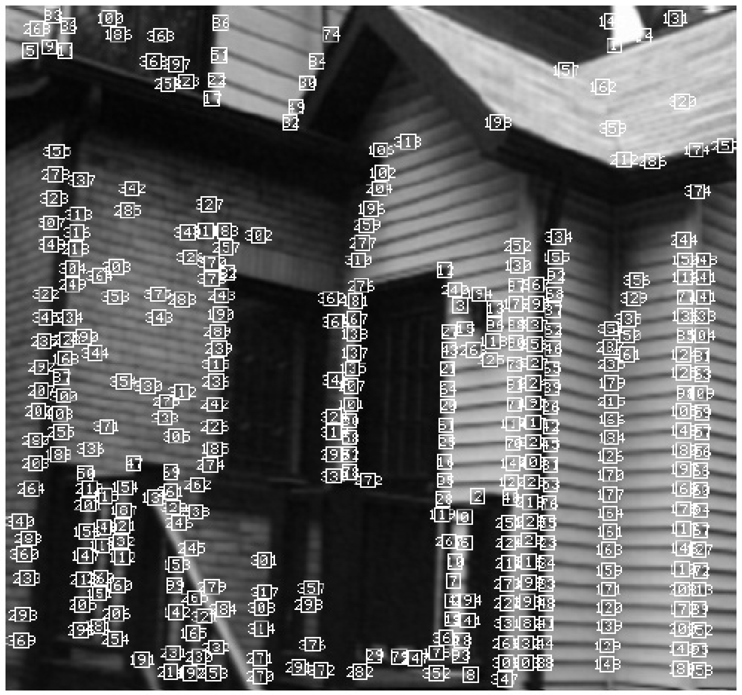
- Affine and Euclidean look similar, after projection, but affine could actually be much more distorted
- Point-wise errors in Euclidean are somewhat higher (more constraints are placed on the solution)

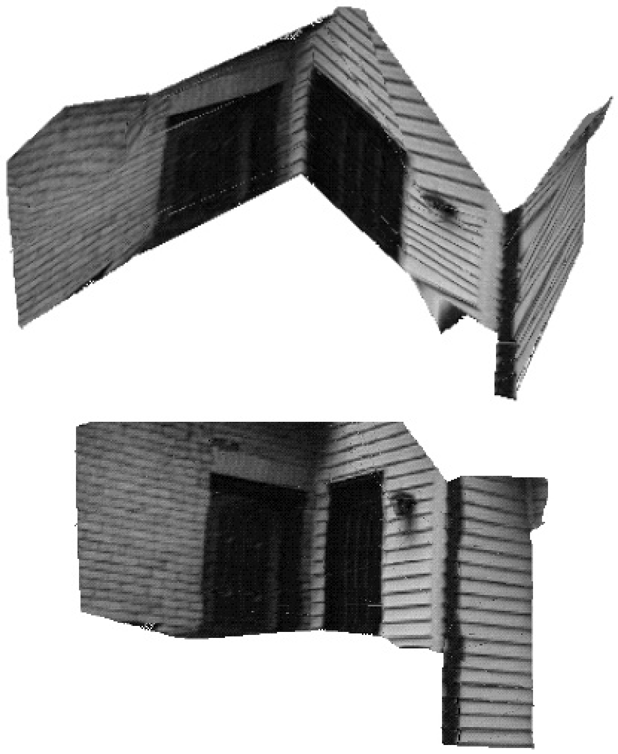
Example

- FP, fig 12.7



Results





Projective Cameras

- Perspective camera projection can be written as

$$\begin{cases} x_{ij} = \dfrac{\boldsymbol{m}_{i1} \cdot \boldsymbol{P}_j}{\boldsymbol{m}_{i3} \cdot \boldsymbol{P}_j} \\ y_{ij} = \dfrac{\boldsymbol{m}_{i2} \cdot \boldsymbol{P}_j}{\boldsymbol{m}_{i3} \cdot \boldsymbol{P}_j} \end{cases} \quad \text{for} \quad i = 1, \dots, m \quad \text{and} \quad j = 1, \dots, n,$$

- M is a 3 x 4 matrix, which can be also written as M = [A b] where A is non-singular 3 x 3 matrix, b is an arbitrary 3-vector
- In a *projective* camera, we remove restrictions on A so M is arbitrary 3 x 4 matrix of rank 3
  - A perspective camera is also a projective camera but not necessarily the reverse
- Projective camera introduces additional ambiguity in reconstruction but equations become simpler
- As before, given $\boldsymbol{p}_{\mathrm{ij}}$, the task is to estimate $\boldsymbol{P}_{\mathrm{j}}$ and M_{i}.

Projective Cameras

- Projective structure from multiple images follows procedure very similar to the affine case
  - We want to get the best solution to D = M P (homogeneous coordinates this time; Z_{ij} is included in D),
  - D is the data matrix as before, M is the camera (motion) matrix and P has the set of scene points. Matrix D has rank 4 (compared to 3 for affine cameras)

$$\mathcal{D} = \begin{pmatrix} Z_{11}p_{11} & Z_{12}p_{12} & \dots & Z_{1n}p_{1n} \\ Z_{21}p_{21} & Z_{22}p_{22} & \dots & Z_{2n}p_{2n} \\ \dots & \dots & \dots & \dots \\ Z_{m1}p_{m1} & Z_{m2}p_{m2} & \dots & Z_{mn}p_{mn} \end{pmatrix}, \; \mathcal{M} = \begin{pmatrix} \mathcal{M}_1 \\ \mathcal{M}_2 \\ \dots \\ \mathcal{M}_m \end{pmatrix} \text{ and } \mathcal{P} = \begin{pmatrix} P_1 & P_2 & \dots & P_n \end{pmatrix}$$

- Minimize E given by:

$$E = \sum_{i,j} ||Z_{ij}p_j - \mathcal{M}_i P_j||^2 = ||\mathcal{D} - \mathcal{M}\mathcal{P}||_F^2$$

Projective Reconstruction

- Similar to the case of affine reconstruction but data matrix *D* contains image coordinates multiplied by Z_{ij}, these parameters are not known prior to solution.
- ***Given*** Z_{ij}: we can compute motion and structure as for the affine case: *i.e.* by factorizing, using SVD.
  - This time, matrix is of rank 4 (compared to rank 3 for affine)
- ***Given*** motion and structure: we can compute Z_{ij}.
- We can iterate between the two steps until convergence.
  - Starting point determines quality of result: a reasonable initial condition is to set $Z_{ij} = 1$, corresponds to weak perspective solution for initialization.
- Solution can degenerate to one where all depths are zero!
  - Non-linear optimization may help overcome this difficulty (see next slide)

Bundle Adjustment

- Determine the camera matrices M_i and points P_j, such that the mean squared error between the actual and predicted image points is minimized, *i.e.* minimize:

$$E = \frac{1}{mn}\sum_{i,j}[(u_{ij} - \frac{\boldsymbol{m}_{i1}\cdot \boldsymbol{P}_j}{\boldsymbol{m}_{i3}\cdot \boldsymbol{P}_j})^2 + (v_{ij} - \frac{\boldsymbol{m}_{i2}\cdot \boldsymbol{P}_j}{\boldsymbol{m}_{i3}\cdot \boldsymbol{P}_j})^2].$$

- This is a physically meaningful quantity and likely to give good results. However, it requires non-linear optimization in a large number of variables. Initial estimates of the camera matrices M_i and points P_j, from factorization can provide good initial conditions.
- *Bundle adjustment* procedures have been in use in the field of *photogrammetry* for a long time but initial estimates are typically derived from knowledge of some 3-D *control* points and cameras are assumed to be calibrated.
- Ambiguity will depend on the restrictions placed on matrix M.

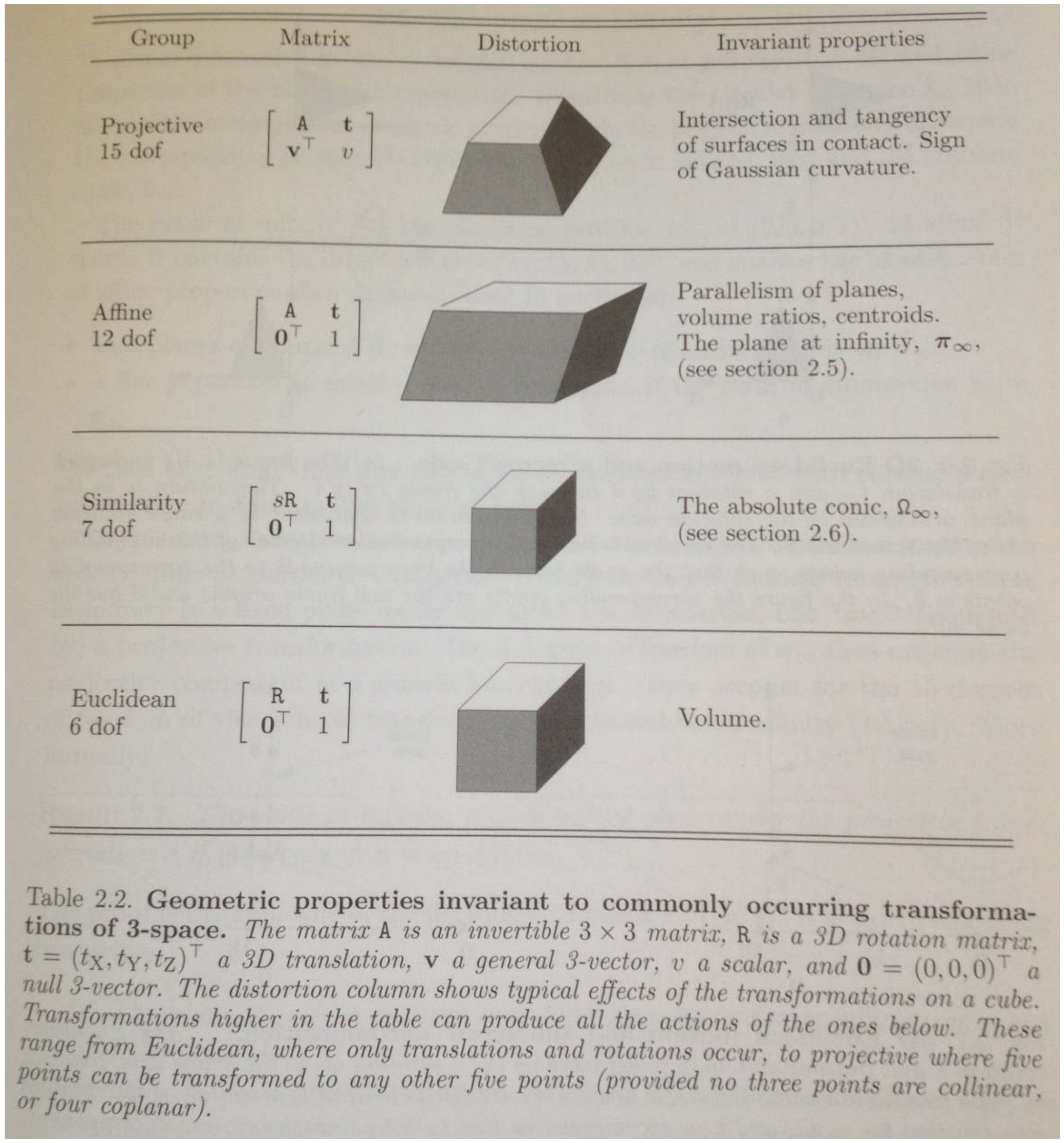
Bilinear Formulation (optional)

- As before, minimize $E = \sum_{i,j} ||Z_{ij}\boldsymbol{p}_j - \mathcal{M}_i \boldsymbol{P}_j||^2$
- Derivative of E with respect to Z_{ij} is zero at the minimum, it can be shown that the minimum,

$$E = \sum_{ij} ||\boldsymbol{p}_{ij} \times (\mathcal{M}_i \boldsymbol{P}_j)||^2$$

- E can be minimized iteratively by keeping P_j fixed or keeping M_i fixed
- Each step requires a linear solution
- Error is guaranteed to keep decreasing at each iteration but solution can degenerate after many iterations
- Has been used in some early commercial systems

3D Transformations (from Hartley-Zisserman Book)



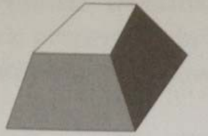
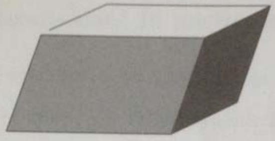
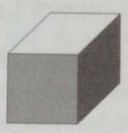
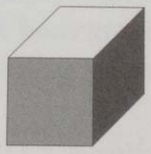
Group	Matrix	Distortion	Invariant properties
Projective 15 dof	$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\top & v \end{bmatrix}$		Intersection and tangency of surfaces in contact. Sign of Gaussian curvature.
Affine 12 dof	$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$		Parallelism of planes, volume ratios, centroids. The plane at infinity, $\boldsymbol{\pi}_\infty$, (see section 2.5).
Similarity 7 dof	$\begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$		The absolute conic, Ω_∞, (see section 2.6).
Euclidean 6 dof	$\begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$		Volume.

Table 2.2. **Geometric properties invariant to commonly occurring transformations of 3-space.** *The matrix* $\mathtt{A}$ *is an invertible* 3×3 *matrix,* $\mathtt{R}$ *is a 3D rotation matrix,* $\mathbf{t} = (t_X, t_Y, t_Z)^\top$ *a 3D translation,* $\mathbf{v}$ *a general 3-vector,* v *a scalar, and* $\mathbf{0} = (0,0,0)^\top$ *a null 3-vector. The distortion column shows typical effects of the transformations on a cube. Transformations higher in the table can produce all the actions of the ones below. These range from Euclidean, where only translations and rotations occur, to projective where five points can be transformed to any other five points (provided no three points are collinear, or four coplanar).*

Projective to Euclidean Shape

- Possible under some conditions/assumptions
- Ambiguity equations: $M_i\hat{} = M_i Q$ and $P_j\hat{} = Q^{-1} P_j$;
  - $Q = (Q_3, \mathbf{q}_4)$ where Q_3 is a (4 x 3) matrix, $\mathbf{q}_4$ is a 4 vector
- $M_i\hat{} = \rho_i \, Ki \; (Ri \; \boldsymbol{t}_i)$ (perspective projection matrix, scale not fixed)
- $M_i \, Q_3 = \rho_i K_i R_i \, Q_3$
- If intrinsic parameters are known, K_i is known; above equation gives sufficient constraints to estimate Q_3 as $M_i \, Q_3$ needs to be a scaled rotation matrix (dot product of rows is zero, norm is the same).

$$\begin{cases} m_{i1}^T A m_{i2} = 0, \\ m_{i2}^T A m_{i3} = 0, \\ m_{i3}^T A m_{i1} = 0, \\ m_{i1}^T A m_{i1} - m_{i2}^T A m_{i2} = 0, \\ m_{i2}^T A m_{i2} - m_{i3}^T A m_{i3} = 0, \end{cases} \qquad A = Q_3 Q_3^T$$

- Equations can be solved to recover Q, by linear or non-linear methods (see book for details): A is a symmetric matrix, so has 10 independent coefficients

Projective to Euclidean Shape: other constraints

- Partial knowledge of intrinsic parameters may suffice

$$\mathcal{M}_i \mathcal{A} \mathcal{M}_i^T = \rho_i^2 \mathcal{K}_i \mathcal{K}_i^T$$

- If center of image is known, $x_0 = y_0 = 0$; it follows that

$$\mathcal{K}_i \mathcal{K}_i^T = \begin{pmatrix} \alpha_i^2 \frac{1}{\sin^2 \theta_i} & -\alpha_i \beta_i \frac{\cos \theta_i}{\sin^2 \theta_i} & 0 \\ -\alpha_i \beta_i \frac{\cos \theta_i}{\sin^2 \theta_i} & \beta_i^2 \frac{1}{\sin^2 \theta_i} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

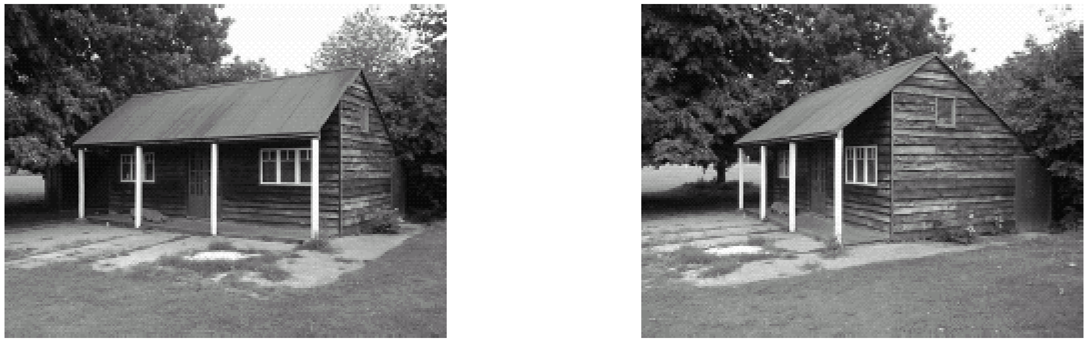
- The two zero entries provide two equations for each camera
- Five views suffice to recover K and Q
- It is easy to add other constraints, e.g. skew is zero ($\theta_i = 90^0$) and/or aspect ratio is one
- Many such special cases studied in literature

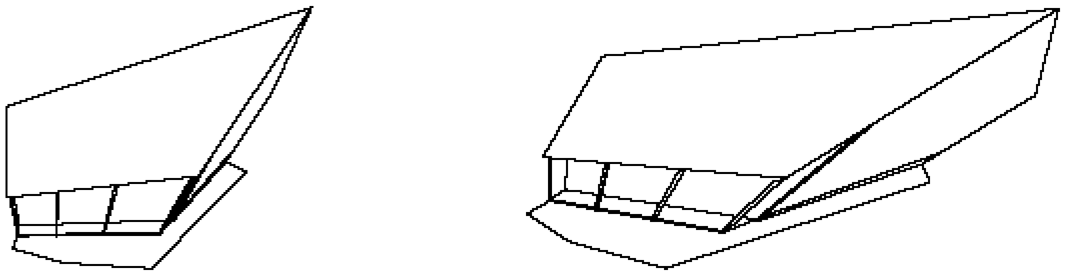
Stratified Reconstruction

- Upgrade in steps
- Each step takes advantage of some observations, e.g.
  - Parallelism of certain lines (indicated by vanishing points)
  - Vanishing (horizon) line
  - Orthogonality of certain lines
- Details beyond scope of cs574
  - See Hartley-Zisserman book for details

Projective Reconstruction Example

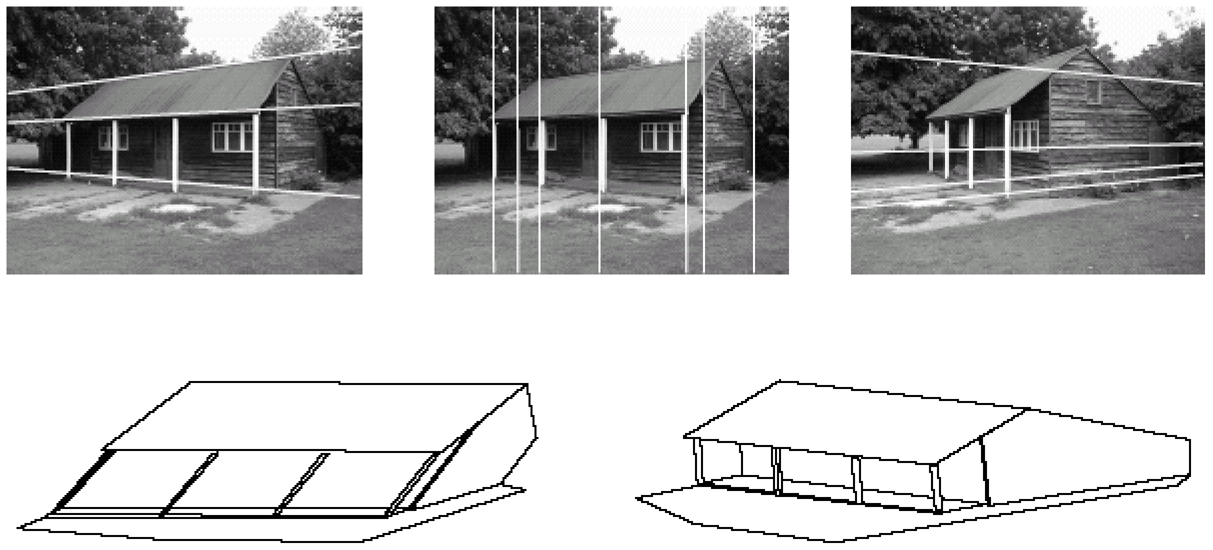
- From Hartley-Zisserman book





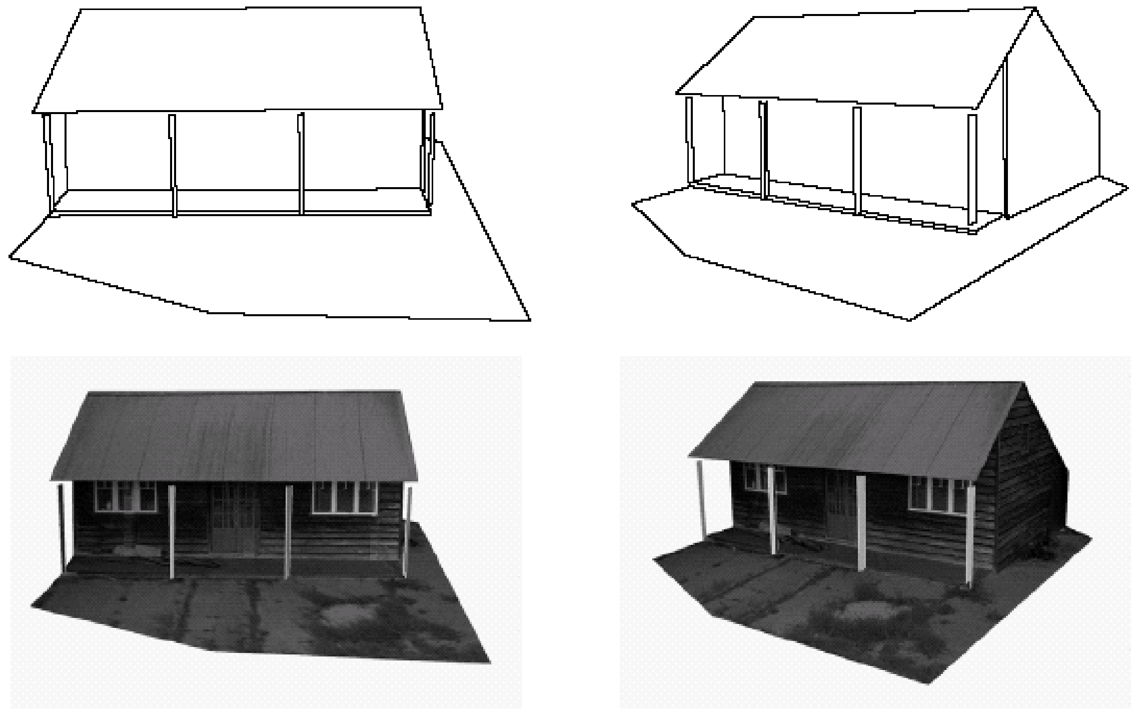
Affine Upgrade

- Based on assumptions of three parallel sets of lines (indicated by vanishing points)

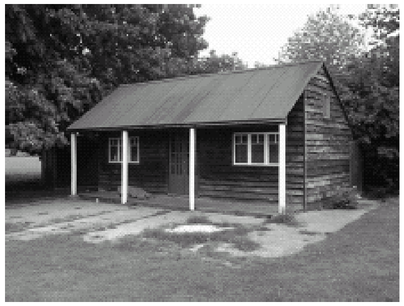
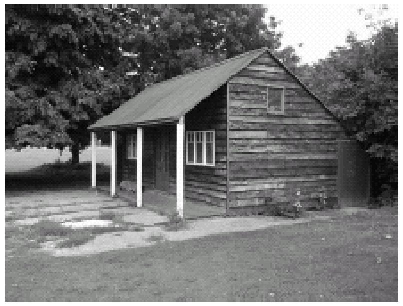

Affine reduction using scene constraints - parallel lines

Metric (similarity) Upgrade

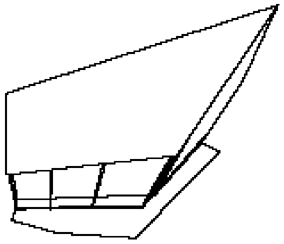
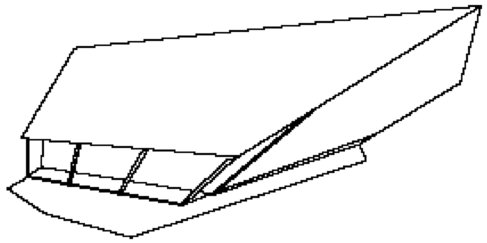
- Based on orthogonality of convergent lines **and** assumption that pixels are square



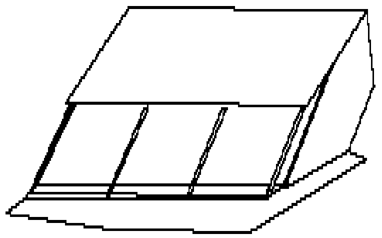
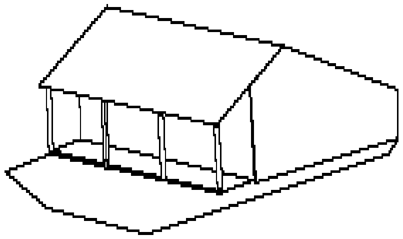
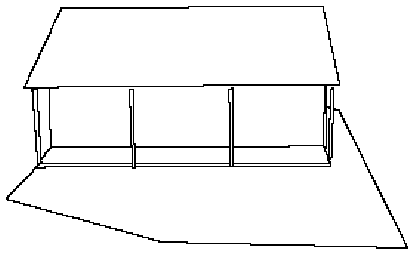
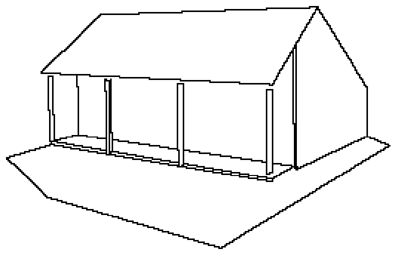
Stratified Reconstruction

Two Images

Projective Reconstruction

Affine

Metric

Multiple Views: Alternate Global Pipeline

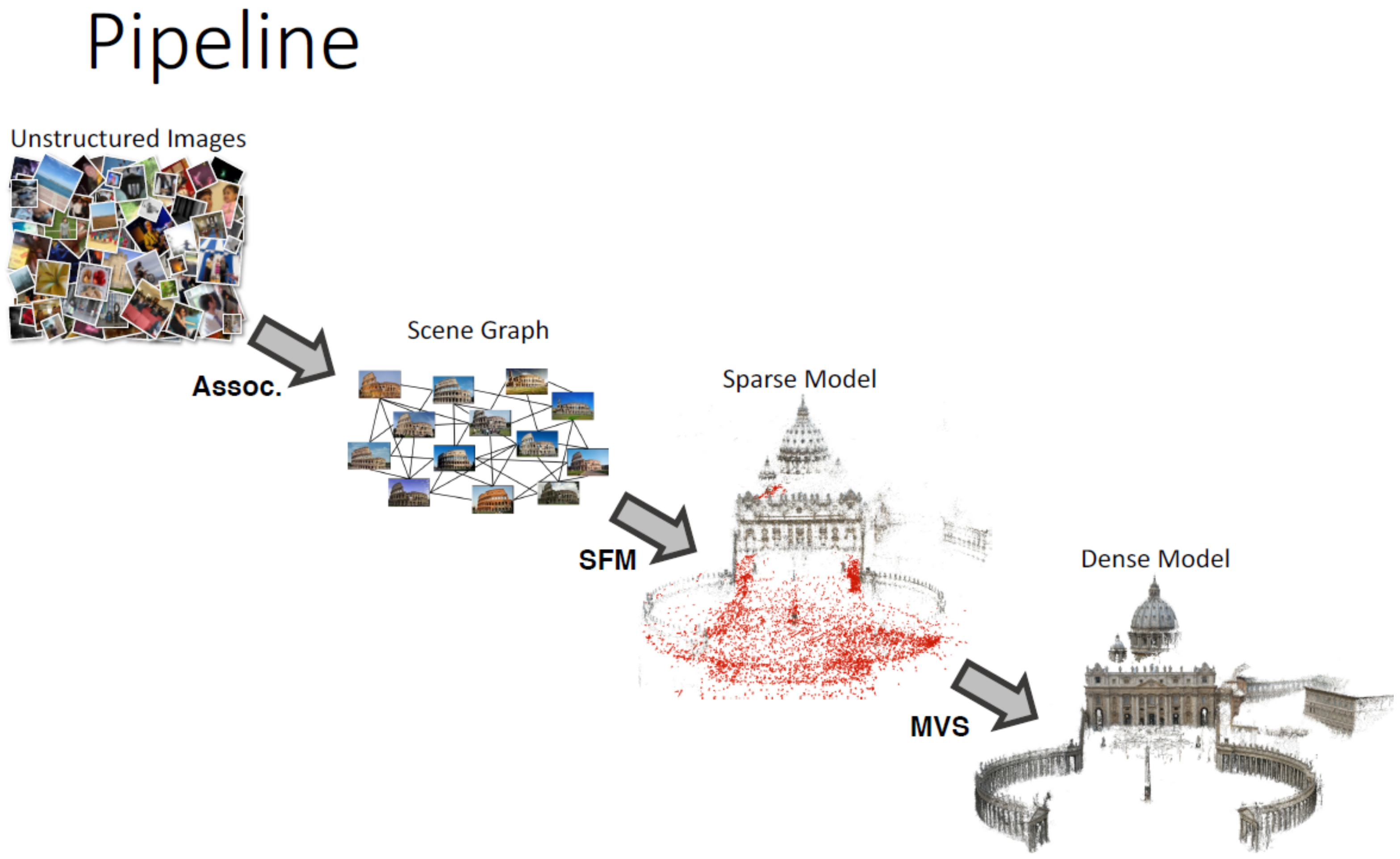


Figure from CVPR2017 Tutorial: Large-scale 3D modeling…

Newer Global Methods

- Assume calibrated cameras (known intrinsic parameters)
- Estimate relative (R, t) between all (many) pairs
- Construct a graph with each view as a node, links between nodes give relative [R,t] for the pair
  - How to choose pairs?
- Find optimal "global" rotations (w.r.t. to some local coordinate system)
  - Filter pairs with poor estimates
- Find global translations, filter poor estimates
- Can iterate between translations and rotations
- Final step is bundle adjustment, above used for initialization
- Is this approach better than factorization approach?
  - Seems more popular in recent literature and for SLAM
- Following slides taken from CVPR2017 tutorial, cited earlier

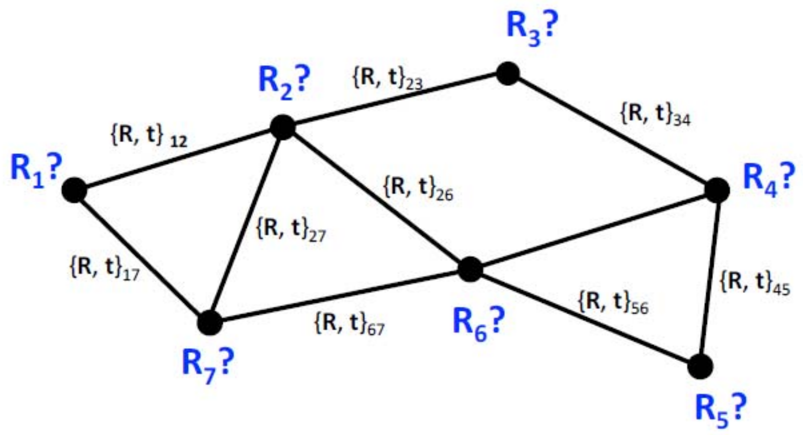
Global SfM

● Image

— Two-View Geometry

1. Estimate global rotations: $\min_{R} \|R_{ij} - R_j R_i^T\|$

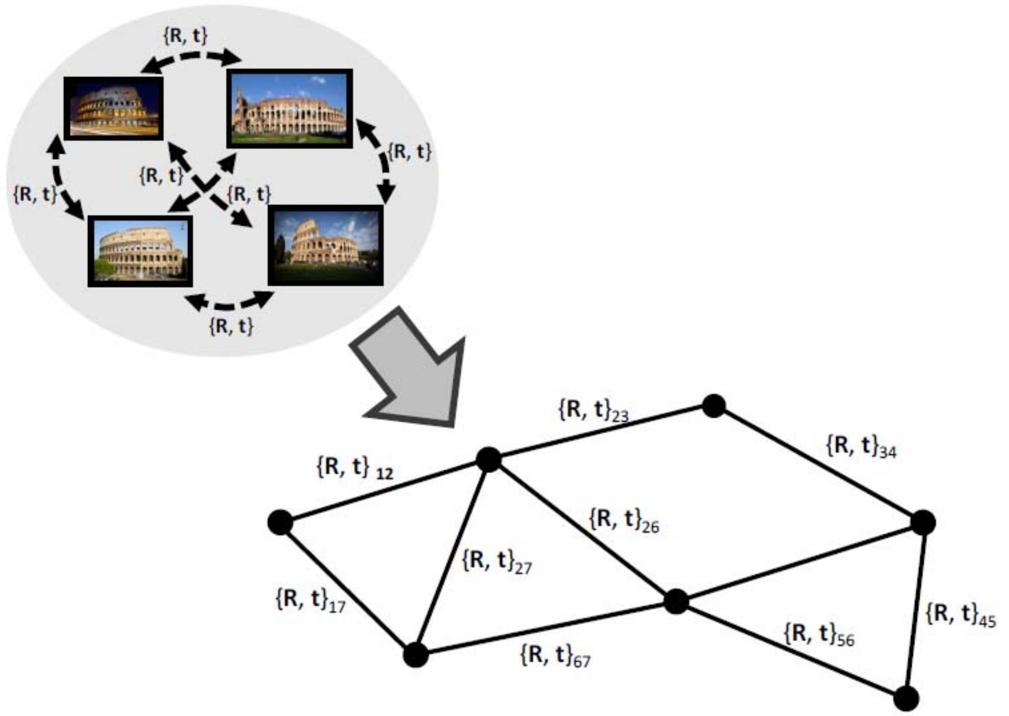
Chatterje and Govindu 2013, "Efficient and Robust Large-Scale Rotation Averaging"



Global SfM

● Image

— Two-View Geometry



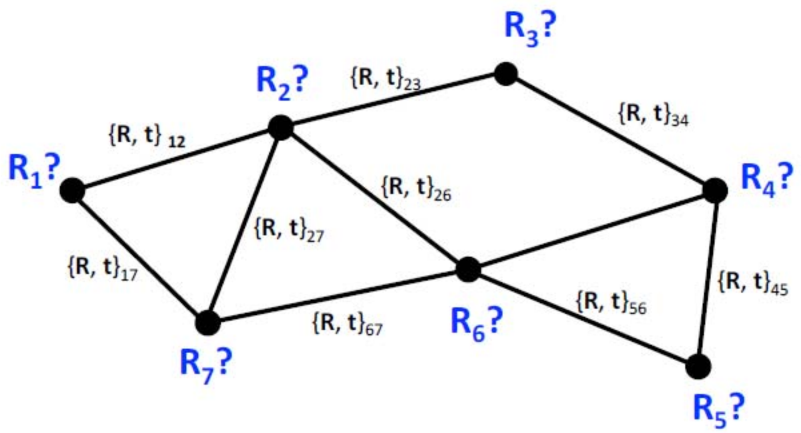
Global SfM

● Image

— Two-View Geometry

1. Estimate global rotations: $\min_{R} \| R_{ij} - R_j R_i^T \|$

Chatterje and Govindu 2013, "Efficient and Robust Large-Scale Rotation Averaging"



ETH zürich Microsoft URCV Large-scale 3D Modeling from Crowdsourced Data 50

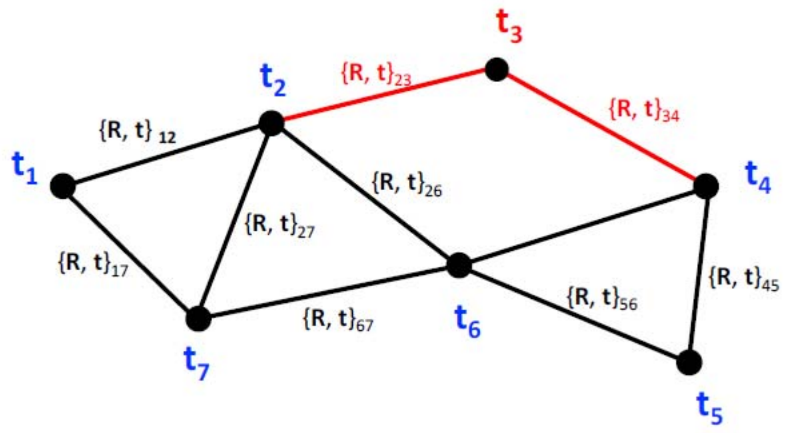
Global SfM

● Image

— Two-View Geometry

1. Estimate global rotations: $\min_{R} \|R_{ij} - R_j R_i^T\|$

   Filter relative rotations: $\|R_{ij} - R_j R_i^T\| > \epsilon$

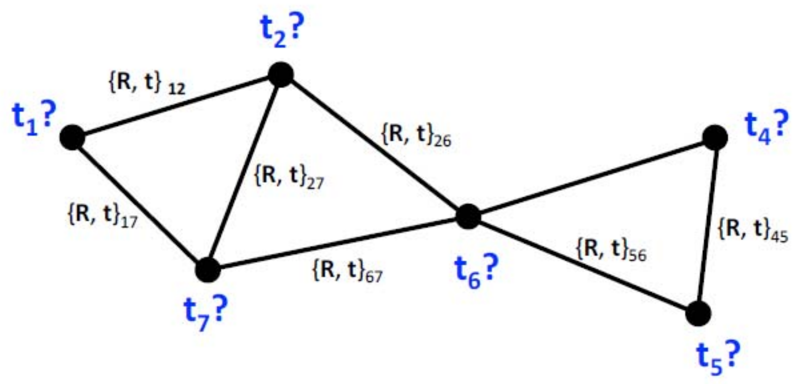


Global SfM

● Image

— Two-View Geometry

1. Estimate and filter global rotations
2. Estimate global translations: $\min_{t} \left\| t_{ij} - \frac{t_i - t_j}{\|t_i - t_j\|} \right\|$



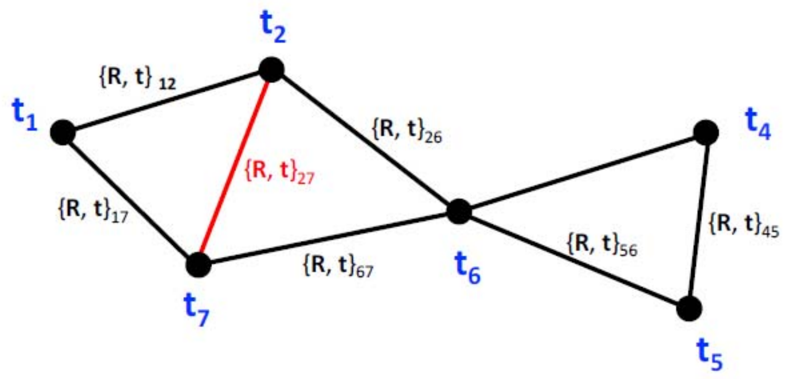
Global SfM

● Image

— Two-View Geometry

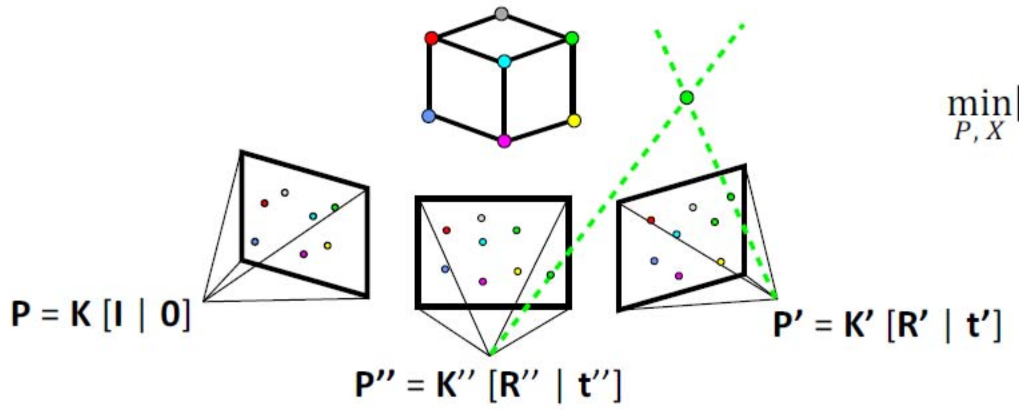
1. Estimate and filter global rotations
2. Estimate global translations: $\min_R \left\| t_{ij} - \frac{t_i - tj}{\|t_i - tj\|} \right\|$

Filter relative translations: $\left\| t_{ij} - \frac{t_i - t_j}{\|t_i - t_j\|} \right\| > \epsilon$



Global SfM

1. Estimate and filter global rotations
2. Estimate and filter global translations
3. Triangulate and refine with bundle adjustment



$$\min_{P,X} \|x - \pi(P, X)\|$$