structure from motion
structure from motion

- for now: “static scene + moving camera”
  - equivalently: rigidly moving scene + static camera
- limiting case of stereo with many cameras
- limiting case of multiview camera calibration with unknown target
- given n points and N camera positions, have 2nN equations and 3n+6N unknowns
approaches

- obtain point correspondences
  - optical flow
  - stereo methods: correlation, feature matching

- solving for points and camera motion
  - nonlinear minimization (bundle adjustment)
  - various approximations...
orthographic camera approximation

- simplest SFM case: camera assumed to apply orthographic projection
weak perspective

- an orthographic assumption is sometimes well approximated by a telephoto lens
consequences of orthographic projection

- scene can be recovered up to scale
- translation perpendicular to image plane can never be recovered
orthographic structure from motion

- method due to Tomasi & Kanade 1992
- assume n points in space: \( p_1, p_2, \ldots, p_n \)
- observed at N points in time at image coordinates: \((x_{ij}, y_{ij}), \quad i = 1 : N, \quad j = 1 : n\)

- feature tracking, optical flow, etc.
orthographic structure from motion

- assemble "measurement matrix":

$$D = \begin{bmatrix}
    x_{11} & \cdots & x_{1n} \\
    \vdots & \ddots & \vdots \\
    x_{N1} & \cdots & x_{Nn} \\
    y_{11} & \cdots & y_{1n} \\
    \vdots & \ddots & \vdots \\
    y_{N1} & \cdots & y_{Nn}
\end{bmatrix}$$
orthographic structure from motion

- step 1: find translation
- translation parallel to viewing direction can not be obtained
- translation perpendicular to viewing direction equals motion of average position of all points
orthographic structure from motion

- subtract average of each row to obtain “registered measurement matrix”

\[
\tilde{D} = \begin{bmatrix}
x_{11} - \bar{x}_1 & \cdots & x_{1n} - \bar{x}_1 \\
\vdots & & \vdots \\
x_{N1} - \bar{x}_N & \cdots & x_{Nn} - \bar{x}_N \\
y_{11} - \bar{y}_1 & \cdots & y_{1n} - \bar{y}_1 \\
\vdots & & \vdots \\
y_{N1} - \bar{y}_N & \cdots & y_{Nn} - \bar{y}_N \\
\end{bmatrix}
\]
orthographic structure from motion

- step 2: try to find rotation
- rotation at each frame defines local coordinate axes: $\hat{i}, \hat{j}, \hat{k}$

\[
\tilde{x}_{ij} = \hat{i}_i^T \tilde{p}_j \\
\tilde{y}_{ij} = \hat{j}_i^T \tilde{p}_j
\]

**Fig. 1.** Systems of reference used in our problem formulation.
orthographic structure from motion

can write: \( \tilde{D} = RS \)

“rotation” matrix \( R \)

“shape” matrix \( S \)

\[
R = \begin{bmatrix}
\hat{\iota}_1^T \\
\vdots \\
\hat{\iota}_N^T \\
\hat{\jmath}_1^T \\
\vdots \\
\hat{\jmath}_N^T
\end{bmatrix}
\]

\[
S = [\tilde{\mathbf{p}}_1 \cdots \tilde{\mathbf{p}}_n]
\]
orthographic structure from motion

- goal is to factor $\tilde{D}$ into $RS$
- first observation: $\text{rank}(\tilde{D}) = 3$
  - in ideal case w/o noise
- proof:
  - rank of $R$ is 3 unless no rotation
  - rank of $S$ is 3 iff noncoplanar points
  - product of 2 rank-3 matrices has rank 3
- in practice, $\text{rank}(\tilde{D}) > 3$
big surprise...SVD

\[ \tilde{D} = U W V^T \]

• if rank 3 => all but 3 singular values should be 0
• extract the largest singular values, together with corresponding columns of U and V
factorization for orthographic structure from motion

- after extracting columns, $U_3$ has dimensions $2N \times 3$ (just what we expected for $R$)
- $W_3 V_3^T$ has dimensions $3 \times n$ (just what we expected for $S$)
- so, let

\[
R^* = U_3 \\
S^* = W_3 V_3^T
\]
affine structure from motion

- the i and j entries of $R^*$ are not, in general, unit length and perpendicular
- we have found motion (and therefore shape) up to an affine transformation
- this is the best we could do if we didn’t assume orthographic camera
ensuring orthogonality

- since $\tilde{D}$ can be factored as $R^* S^*$ it can also be factored as $(R^* Q)(Q^{-1} S^*)$ for any $Q$
- so... search for $Q$ such that $R = R^* Q$ has the properties we want (i.e., orthogonality and unit-length)
ensuring orthogonality

- want: \( \hat{i}_i^* T Q Q^T \hat{i}_i^* = 1 \)
  \( \hat{j}_i^* T Q Q^T \hat{j}_i^* = 1 \)
  \( \hat{i}_i^* T Q Q^T \hat{j}_i^* = 0 \)

- let \( T = QQ^T \)

- equations for elements of \( T \) (LSQ)

- ambiguity, add constraints:

\[
Q^T \hat{i}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad Q^T \hat{j}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}
\]
ensuring orthogonality

- have found $T = QQ^T$
- find $Q$ by taking “square root” of $T$
  - Cholesky decomposition if $T$ is positive definite
  - general algorithms (e.g., sqrtm in MATLAB)
orthogonal structure from motion

- let’s recap:
  - write down matrix of observations
  - find translation from avg. position
  - subtract translation
  - factor matrix using SVD
  - write down equations for orthogonalization
  - solve using least squares, square root
orthogonal structure from motion

\* in the end, you can directly compute the shape and camera information:

\[
R = R^* Q \\
S = Q^{-1} S^*
\]

- camera positions
- 3d positions
results

- image sequence:

  ![Image 1](image1.png)
  ![Image 2](image2.png)
  ![Image 3](image3.png)
  ![Image 4](image4.png)
results

- tracked features:
results

- reconstructed shape:

[Tomasi & Kanade]
orthographic => perspective

- with orthographic or “weak perspective” cannot recover all information
- with full perspective, can recover more information (translation along optical axis)
- result: can recover geometry and full motion up to global scale factor
perspective SFM methods

- bundle adjustment (full nonlinear minimization)
- methods based on factorization
- methods based on fundamental matrices
- methods based on vanishing points
motion field for camera motion

- translation:

- motion field lines converge (possibly at $\infty$)
motion field for camera motion

- rotation:

- motion field lines do not converge
motion field for camera motion

- combined rotation and translation: motion field lines have component that converges, and component that does not
- algorithms can look for vanishing point, then determine component of motion around this point
- "focus of expansion / contraction"
- "instantaneous epipole"
finding instantaneous epipole

- observation: motion field due to translation depends on depth of points
- motion field due to rotation does not
- idea: compute difference between motion of a point, motion of neighbors
- differences point towards instantaneous epipole
SVD (yet again!!!)

- want to fit direction to all dv (differences in optical flow) within some neighborhood
- PCA on matrix of dv
- equivalently, take eigenvector of $A = \Sigma(dv)(dv)^T$ corresponding to largest eigenvalue
- gives direction of parallax $l_i$ in that patch, together with estimate of reliability
SFM algorithm

- compute optical flow
- find vanishing point (least squares solution)
- find direction of translation from epipole
- find perpendicular component of motion
- find velocity, axis of rotation
- find depths of points (up to global scale)
case study: Photo Tourism

Photo Tourism
Exploring photo collections in 3D

(a) (b) (c)

[Snavely et al.]
Photo Tourism
Exploring photo collections in 3D

Noah Snavely  Steven M. Seitz  Richard Szeliski
University of Washington  Microsoft Research

SIGGRAPH 2006