Structure from Motion

Projections and image formation

Problem statement

Rank theorem

Factorization method
Perspective Projection

- Projectors converge at a single point: center of projection.
Parallel Projection

- Projectors are parallel to each other: center of projection at infinity.
Orthographic Projection

- Projectors are parallel to each other and perpendicular to projection surface:
Orthographic Image Formation

- Model the position of an object with respect to the camera by a rigid transformation (rotation + translation).
- Apply projection matrix (typically, drop one coordinate).

\[
\begin{bmatrix}
  x_{\text{img}} \\
  y_{\text{img}}
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  0 & 0 & 0
\end{bmatrix}
\begin{pmatrix}
  R \\
  x \\
  y \\
  z
\end{pmatrix} + t
\]
Questions

- Is it possible to recover the original (3D) scene coordinates (structure)?
- Is it possible to recover the relative transformation (motion)?

Structure from Motion: yes* to both questions, if we track a number of points in a sequence of images (frames)!

*Certain restrictions may apply!
Definition: the motion field is the 2D vector field of velocities of the image points, induced by the relative motion between the viewing camera and the observed scene.

- Dense motion field (optical flow): a motion vector at (nearly) every pixel.
- Sparse motion field: a motion vector for a sparse set of landmarks in the image.
Structure from Motion

- **Problem statement**: Given the 2D motion field estimated from an image sequence, compute the shape, or 3D structure, of the visible objects, and their 3D motion with respect to the viewing camera.

Assumptions

- Orthographic camera model.

- The position of $n$ image points, corresponding to $n$ scene points: $P_1, \ldots, P_n$, not all coplanar, have been tracked in $N$ frames ($N \geq 3$).
Measurement Matrix

- The j-th image point \((j = 1, \ldots, n)\) in the i-th frame \((i = 1, \ldots, N)\):

\[
p_{ij} = [x_{ij} \ y_{ij}]^T
\]

- Measurement matrix \((2N \times n)\):

\[
W = \begin{bmatrix}
X = \{x_{ij}\} \\
Y = \{y_{ij}\}
\end{bmatrix}
\]
Registered Measurement Matrix

\[ \tilde{x}_i = \frac{1}{n} \sum_{j=1}^{n} x_{ij} \quad \tilde{y}_i = \frac{1}{n} \sum_{j=1}^{n} y_{ij} \]

\[ \tilde{x}_{ij} = x_{ij} - \tilde{x}_i \quad \tilde{y}_{ij} = y_{ij} - \tilde{y}_i \]

\[ \tilde{W} = \begin{bmatrix} \tilde{X} = \{ \tilde{x}_{ij} \} \\ \tilde{Y} = \{ \tilde{y}_{ij} \} \end{bmatrix} \]
The registered measurement matrix $W$ (without noise) has at most rank 3.

Proof idea: the registered measurement matrix may be decomposed into a product of two matrices, rotation $R$, and shape $S$, each of which has rank 3.
The Rank Theorem - Proof

\( P_j \) - the j-th tracked object point
(3D point in the world reference frame)

\( i_i, j_i \) - the reference frame (axes) of the i-th image in the sequence (3D vectors in the world reference frame).

\( T_i \) - the vector from the world origin to the origin of the i-th image reference frame
The Rank Theorem - Proof

\[ x_{ij} = i_i^T (P_j - T_i) \]
\[ y_{ij} = j_i^T (P_j - T_i) \]
The Rank Theorem - Proof

\[ \tilde{x}_{ij} = i_i^T (P_j - T_i) - \frac{1}{n} \sum_{m=1}^{n} i_i^T (P_m - T_i) \]

\[ = i_i^T (P_j - T_i) - i_i^T \left( \frac{1}{n} \sum_{m=1}^{n} P_m - T_i \right) \]

by definition:

\[ \frac{1}{n} \sum_{m=1}^{n} P_m = 0 \]

\[ \tilde{x}_{ij} = i_i^T P_j \]

\[ \tilde{y}_{ij} = j_i^T P_j \]

Tuesday, June 5, 2007
The Rank Theorem - Proof

\[
\tilde{W} = \begin{bmatrix}
  i_1^T \\
  \vdots \\
  i_N^T \\
  j_1^T \\
  \vdots \\
  j_N^T 
\end{bmatrix}
\begin{bmatrix}
P_1 \\
P_2 \\
\vdots \\
P_n
\end{bmatrix} = R S
\]

$2N \times n$  
$3 \times n$  
$2N \times 3$
The factorization is not unique. Let $Q$ be any invertible 3x3 matrix, then:

$$\tilde{W} = R\, S = R\, QQ^{-1}\, S = (RQ)(Q^{-1}S)$$

Add constraints:
- Rows of $R$ must have unit norm;
- First $n$ rows of $R$ must be orthogonal to last (corresponding) $n$ rows.
Consider the singular value decomposition:

$$\tilde{W} = U D V^T$$

Rank 3 implies that only 3 singular values are non-zero!

$$\tilde{W} = \begin{bmatrix} u_1 & \cdots & u_{2N} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}^T$$

$$2N \times 2N \quad 2N \times n \quad n \times n$$
Factorization Algorithm

- Rewrite the decomposition as:

\[
\begin{align*}
\tilde{W} &= \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}^T \\
&= U' D' V'^T
\end{align*}
\]

Where:
- \( \tilde{W} \): The factorization matrix.
- \( U' \): The left singular vectors.
- \( D' \): The diagonal singular values.
- \( V'^T \): The right singular vectors.

\[
\begin{align*}
2N \times 3 & \quad 3 \times 3 & \quad 3 \times n \\
U' & \quad D' & \quad V'^T
\end{align*}
\]

\[
\hat{R} = U' D'^{1/2} \quad \hat{S} = D'^{1/2} V'^T
\]
Factorization Algorithm

- We get a decomposition with the desired dimensions:

\[ \tilde{W} = \hat{R} \hat{S} = (U' D'^{1/2}) (D'^{1/2} V'T) \]

- Look for Q, s.t.:

\[ \hat{i}^T_i Q Q^T \hat{i}_i = 1 \]
\[ \hat{j}^T_i Q Q^T \hat{j}_i = 1 \]
\[ \hat{i}^T_i Q Q^T \hat{j}_i = 0 \]

\[ \tilde{W} = RS = (\hat{R} Q) (Q^{-1}\hat{S}) \]
Solving for Q: method A

- Use a non-linear solver (e.g., Newton’s method) to directly solve for the variables in the non-linear (quadratic) system of equations:

\[
\hat{i}_i^T Q Q^T \hat{i}_i = 1 \\
\hat{j}_i^T Q Q^T \hat{j}_i = 1 \\
\hat{i}_i^T Q Q^T \hat{j}_i = 0
\]
Solving for Q: method B

- Solve the linear system of equations:
  \[ \hat{i}^T_i C \hat{i}_i = 1 \]
  \[ \hat{j}^T_i C \hat{j}_i = 1 \]
  \[ \hat{i}^T_i C \hat{j}_i = 0 \]

- Use Cholesky to factor the matrix C:
  \[ QQ^T = C \]

- But beware: C might not be s.p.d!