Outline

- Introduction
- Motion field vs. optical flow
- Brightness constancy
- Gradient-based optical flow estimation
- Finite displacement and feature-based methods
- **3D Structure and motion**
- Summary

3D structure and motion: Approaches

Problem statement

- Given image motion
  - Optical flow-based
  - Finite displacement-based
- Seek to recover
  - 3D motion (translation and rotation)
  - 3D structure (geometric layout of environment)

3D structure and motion: Approaches

Problem statement

- Given image motion
  - Optical flow-based
  - Finite displacement-based
- Seek to recover
  - 3D motion (translation and rotation)
  - 3D structure (geometric layout of environment)

Finite displacement approach

- Seeks to recover
  - full 3D rotation and translation
  - 3D structure
- Typically used in conjunction with finite displacement image motion estimates.

Infinitesimal approach

- Seeks to recover
  - 3D rotational and translation velocity (temporal derivatives of full rotation and translation)
  - 3D structure
- Typically used in conjunction with optical flow image motion estimates.

3D structure and motion: Approaches

Problem statement

- Given image motion
  - Optical flow-based
  - Finite displacement-based
- Seek to recover
  - 3D motion (translation and rotation)
  - 3D structure (geometric layout of environment)

Finite displacement approach

- Seeks to recover
  - full 3D rotation and translation
  - 3D structure
- Typically used in conjunction with finite displacement image motion estimates.

Infinitesimal approach

- Seeks to recover
  - 3D rotational and translation velocity (temporal derivatives of full rotation and translation)
  - 3D structure
- Typically used in conjunction with optical flow image motion estimates.

3D structure and motion: 3D recovery from 2D motion

Where we will go

- We now will develop an approach to recovering 3D structure and motion.
- Here, we will assume
  - arbitrary 3D motion
  - orthographic model of image formation
  - as input we have extracted the position n image points, corresponding to scene points $\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_n$, not all coplanar, and they have been tracked across N >=3 frames, For example, the points tracks might be acquired from the feature-based, finite displacement algorithm described earlier as it operates across an entire video sequence.
- The overall approach is known as the factorization approach, as it relies on a clever observation about how we can factor matrices of feature correspondences into 2 matrices.
  - One matrix captures the 3D structure.
  - The other matrix captures the 3D motion.

3D structure and motion: 3D recovery from 2D motion

Notation

- Let

$$\boldsymbol{p}_{ij} = (x_{ij}, y_{ij})^{\mathbf{T}}$$

  denote the jth image point ($j = 1, \ldots, n$) at the ith frame, ($i=1, \ldots, N$).
- Think of the x_{ij} and y_{ij} as entries of two N x n matrices **X** and **Y**.

3D structure and motion: 3D recovery from 2D motion

Notation

- Let

$$\boldsymbol{p}_{ij} = (x_{ij}, y_{ij})^{\mathbf{T}}$$

  denote the jth image point ($j = 1, \ldots, n$) at the ith frame, ($i=1, \ldots, N$).
- Think of the x_{ij} and y_{ij} as entries of two N x n matrices **X** and **Y**.
- Form the 2N x n measurement matrix

$$\mathbf{W} = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix}$$

3D structure and motion: 3D recovery from 2D motion

Notation

- Let

$$\boldsymbol{p}_{ij} = (x_{ij}, y_{ij})^{\mathbf{T}}$$

  denote the jth image point ($j = 1, \ldots, n$) at the ith frame, ($i=1, \ldots, N$).
- Think of the x_{ij} and y_{ij} as entries of two N x n matrices **X** and **Y**.
- Form the $2N$ x n measurement matrix

$$\mathbf{W} = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix}$$

- Now, subtract the mean of the entries on the same row from each x_{ij} and y_{ij}

$$\bar{x}_{ij} = x_{ij} - \bar{x}_i \qquad \bar{y}_{ij} = y_{ij} - \bar{y}_i$$

  where

$$\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij} \qquad \bar{y}_i = \frac{1}{n}\sum_{j=1}^{n} y_{ij}$$

  are the coordinates of the centroid of the image points in the ith frame.

3D structure and motion: 3D recovery from 2D motion

Notation

- Let

$$\boldsymbol{p}_{ij} = (x_{ij}, y_{ij})^{\mathbf{T}}$$

denote the jth image point ($j = 1, \ldots, n$) at the ith frame, ($i = 1, \ldots, N$).
- Think of the x_{ij} and y_{ij} as entries of two N x n matrices **X** and **Y**.
- Form the $2N$ x n measurement matrix

$$\mathbf{W} = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix}$$

- Now, subtract the mean of the entries on the same row from each x_{ij} and y_{ij}

$$\bar{x}_{ij} = x_{ij} - \bar{x}_i \qquad \bar{y}_{ij} = y_{ij} - \bar{y}_i$$

where

$$\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij} \qquad \bar{y}_i = \frac{1}{n}\sum_{j=1}^{n} y_{ij}$$

are the coordinates of the centroid of the image points in the ith frame.
- Now stack these registered points analogously to our previous construction to form the registered measurement matrix

$$\overline{\mathbf{W}} = \begin{bmatrix} \overline{\mathbf{X}} \\ \overline{\mathbf{Y}} \end{bmatrix}$$

3D structure and motion: 3D recovery from 2D motion

The rank theorem (statement)

- The factorization approach is based on the proof of a straightforward, but fundamental result: The registered measurement matrix has at most rank 3.

3D structure and motion: 3D recovery from 2D motion

The rank theorem (statement)

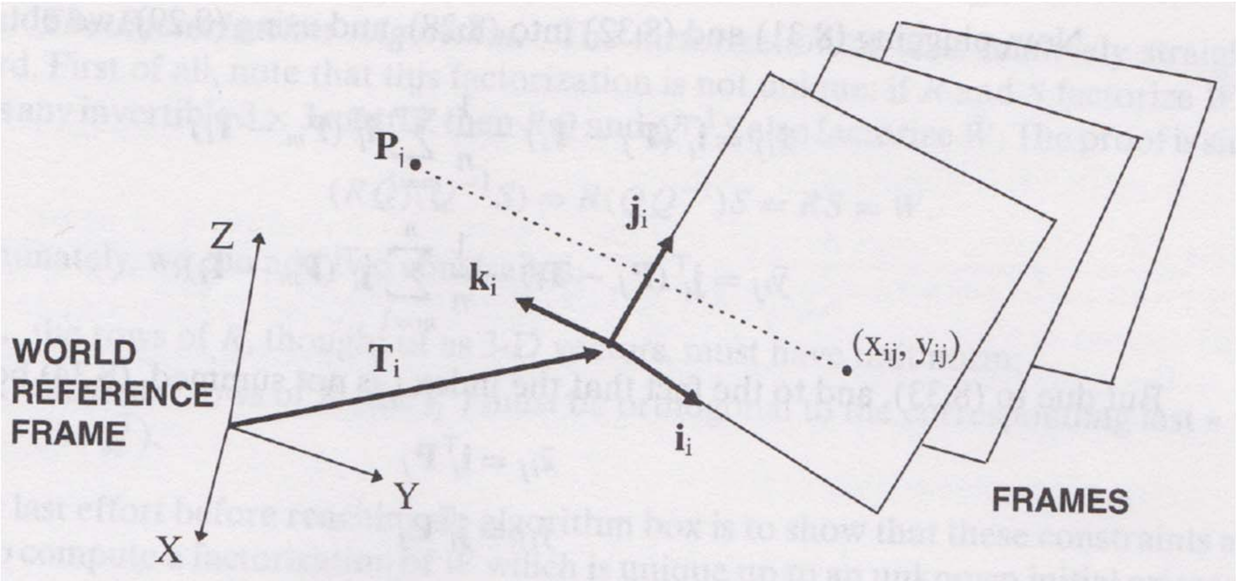
- The factorization approach is based on the proof of a straightforward, but fundamental result: The registered measurement matrix has at most rank 3.
- The proof is based on a decomposition (factorization) of the matrix in the product of a $2N$ x 3 matrix, **R**, and a 3 x n matrix **S**.
- **R** describes the frame-to-frame motion (rotation) of the camera with respect to the points $\boldsymbol{P}_j$.
- **S** describes the 3D structure (or shape) of the points.

3D structure and motion: 3D recovery from 2D motion

The rank theorem (proof)

- Consider all quantities expressed in an object-centred reference frame with the origin at the centroid of $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$. Thus,

$$\frac{1}{n}\sum_{j=1}^{n}\boldsymbol{P}_j = 0$$



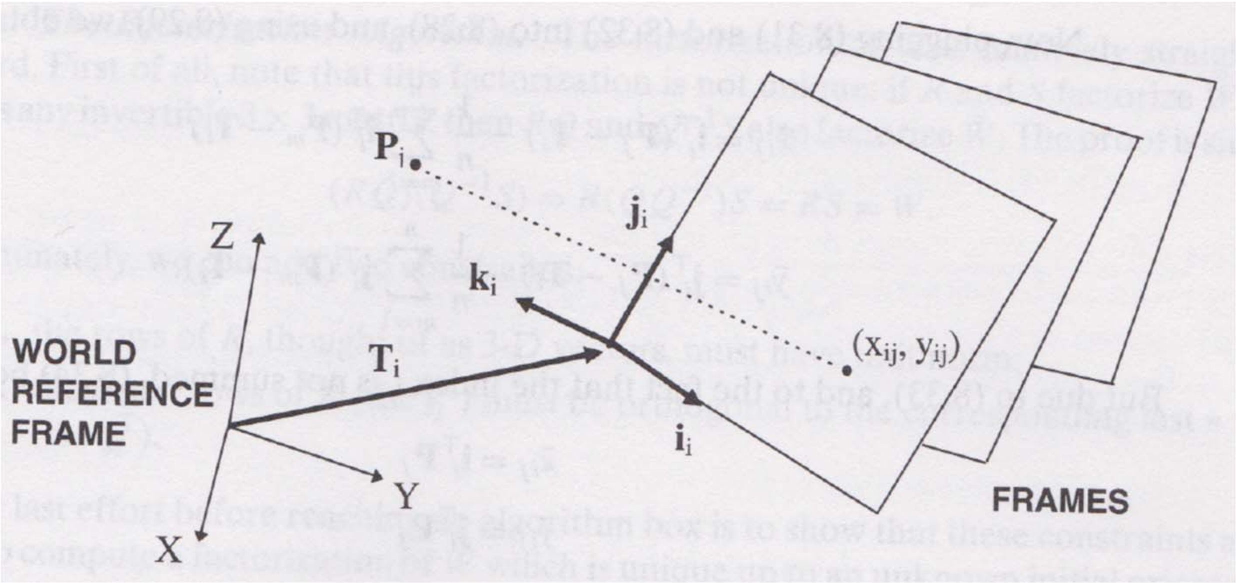
3D structure and motion: 3D recovery from 2D motion

The rank theorem (proof)

- Consider all quantities expressed in an object-centred reference frame with the origin at the centroid of $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$. Thus,

$$\frac{1}{n}\sum_{j=1}^{n}\boldsymbol{P}_j = 0$$

- Let $\mathbf{i}_i$ and $\mathbf{j}_i$ denote the unit vectors that define the image reference frame, expressed in the world reference frame at time i.



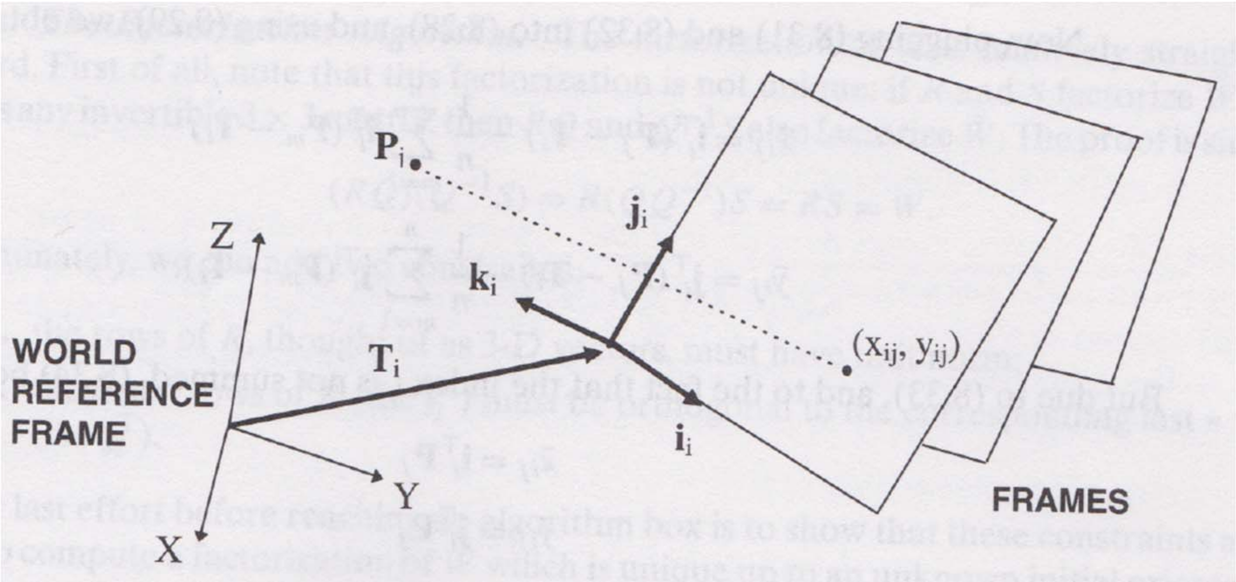
3D structure and motion: 3D recovery from 2D motion

The rank theorem (proof)

- Consider all quantities expressed in an object-centred reference frame with the origin at the centroid of $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$. Thus,

$$\frac{1}{n}\sum_{j=1}^{n} \boldsymbol{P}_j = 0$$

- Let $\mathbf{i}_i$ and $\mathbf{j}_i$ denote the unit vectors that define the image reference frame, expressed in the world reference frame at time i.
- The direction of the optical axis is then given as $\mathbf{k}_i = \mathbf{i}_i \times \mathbf{j}_i$.



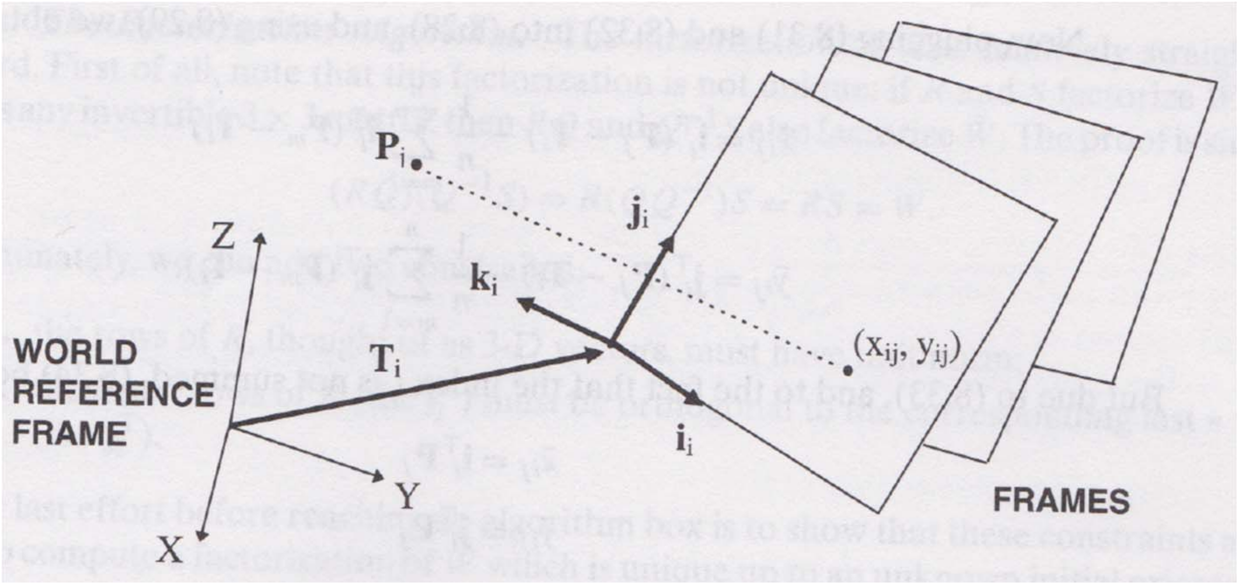
3D structure and motion: 3D recovery from 2D motion

The rank theorem (proof)

- Consider all quantities expressed in an object-centred reference frame with the origin at the centroid of $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$. Thus,

$$\frac{1}{n}\sum_{j=1}^{n}\boldsymbol{P}_j = 0$$

- Let $\mathbf{i}_i$ and $\mathbf{j}_i$ denote the unit vectors that define the image reference frame, expressed in the world reference frame at time i.
- The direction of the optical axis is then given as $\mathbf{k}_i = \mathbf{i}_i \times \mathbf{j}_i$.
- Let $\boldsymbol{T}$, be the vector from the world origin to the origin of the ith image frame.



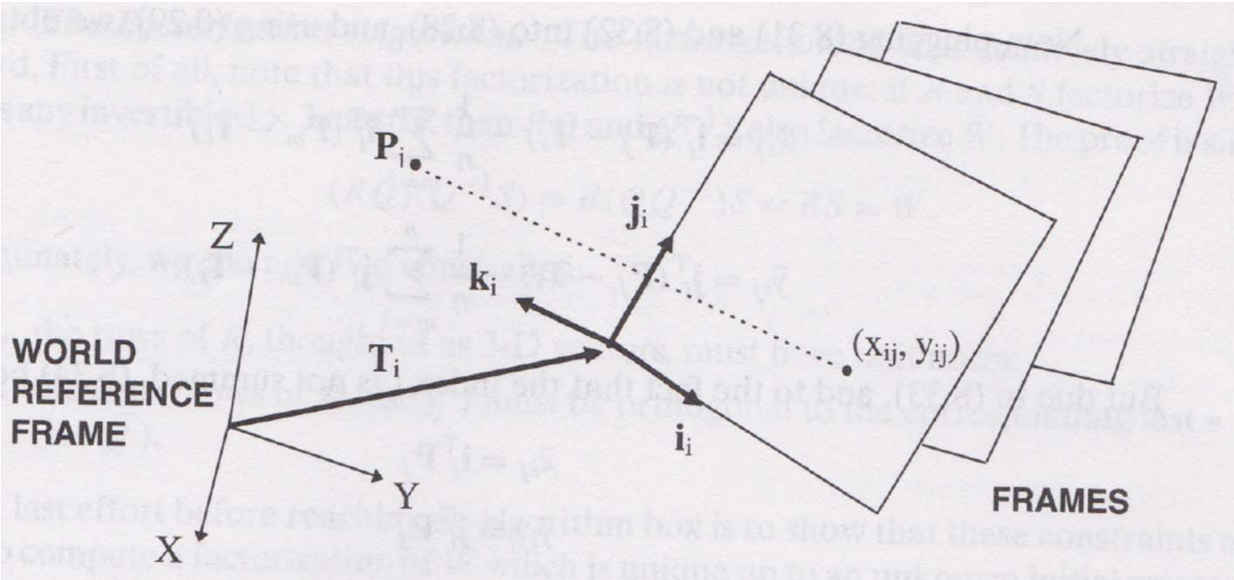
3D structure and motion: 3D recovery from 2D motion

The rank theorem (proof)

- Consider all quantities expressed in an object-centred reference frame with the origin at the centroid of $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$. Thus,

$$\frac{1}{n}\sum_{j=1}^{n}\boldsymbol{P}_j = 0$$

- Let $\mathbf{i}_i$ and $\mathbf{j}_i$ denote the unit vectors that define the image reference frame, expressed in the world reference frame at time i.
- The direction of the optical axis is then given as $\mathbf{k}_i = \mathbf{i}_i \times \mathbf{j}_i$.
- Let $\boldsymbol{T}$, be the vector from the world origin to the origin of the ith image frame.
- We have $x_{ij} = \mathbf{i}_i^{\mathrm{T}}(\boldsymbol{P}_j - \boldsymbol{T}_i) \qquad y_{ij} = \mathbf{j}_i^{\mathrm{T}}(\boldsymbol{P}_j - \boldsymbol{T}_i)$



3D structure and motion: 3D recovery from 2D motion

The rank theorem (proof)

- Now, the registered points can be written as

$$\bar{x}_{ij} = \mathbf{i}_i^{\mathbf{T}}(\boldsymbol{P}_j - \boldsymbol{T}_i) - \frac{1}{n}\sum_{m=1}^{n}\mathbf{i}_i^{\mathbf{T}}(\boldsymbol{P}_m - \boldsymbol{T}_i)$$

$$\bar{y}_{ij} = \mathbf{j}_i^{\mathbf{T}}(\boldsymbol{P}_j - \boldsymbol{T}_i) - \frac{1}{n}\sum_{m=1}^{n}\mathbf{j}_i^{\mathbf{T}}(\boldsymbol{P}_m - \boldsymbol{T}_i)$$

3D structure and motion: 3D recovery from 2D motion

The rank theorem (proof)

- Now, the registered points can be written as

$$\bar{x}_{ij} = \mathbf{i}_i^{\mathbf{T}}(\boldsymbol{P}_j - \boldsymbol{T}_i) - \frac{1}{n}\sum_{m=1}^{n}\mathbf{i}_i^{\mathbf{T}}(\boldsymbol{P}_m - \boldsymbol{T}_i)$$

$$\bar{y}_{ij} = \mathbf{j}_i^{\mathbf{T}}(\boldsymbol{P}_j - \boldsymbol{T}_i) - \frac{1}{n}\sum_{m=1}^{n}\mathbf{j}_i^{\mathbf{T}}(\boldsymbol{P}_m - \boldsymbol{T}_i)$$

- Further, recalling that

$$\frac{1}{n}\sum_{j=1}^{n}\boldsymbol{P}_j = 0$$

and the fact that the index i is not summed, we have

$$\bar{x}_{ij} = \mathbf{i}_i^{\mathbf{T}}\boldsymbol{P}_j \qquad\qquad \bar{y}_{ij} = \mathbf{j}_i^{\mathbf{T}}\boldsymbol{P}_j$$

3D structure and motion: 3D recovery from 2D motion

The rank theorem (proof)

- Now, the registered points can be written as

$$\bar{x}_{ij} = \mathbf{i}_i^{\mathbf{T}}(\boldsymbol{P}_j - \boldsymbol{T}_i) - \frac{1}{n}\sum_{m=1}^{n}\mathbf{i}_i^{\mathbf{T}}(\boldsymbol{P}_m - \boldsymbol{T}_i)$$

$$\bar{y}_{ij} = \mathbf{j}_i^{\mathbf{T}}(\boldsymbol{P}_j - \boldsymbol{T}_i) - \frac{1}{n}\sum_{m=1}^{n}\mathbf{j}_i^{\mathbf{T}}(\boldsymbol{P}_m - \boldsymbol{T}_i)$$

- Further, recalling that

$$\frac{1}{n}\sum_{j=1}^{n}\boldsymbol{P}_j = 0$$

and the fact that the index i is not summed, we have

$$\bar{x}_{ij} = \mathbf{i}_i^{\mathbf{T}}\boldsymbol{P}_j \qquad \bar{y}_{ij} = \mathbf{j}_i^{\mathbf{T}}\boldsymbol{P}_j$$

- So, we define

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{i}_N^{\mathbf{T}} \\ \mathbf{j}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{j}_N^{\mathbf{T}} \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} \boldsymbol{P}_1 & \boldsymbol{P}_2 & \cdots & \boldsymbol{P}_n \end{bmatrix}$$

- And write $\overline{\mathbf{W}} = \mathbf{RS}$

3D structure and motion: 3D recovery from 2D motion

The rank theorem (proof)

- Finally, we note that given the constructions

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{i}_N^{\mathbf{T}} \\ \mathbf{j}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{j}_N^{\mathbf{T}} \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} \boldsymbol{P}_1 & \boldsymbol{P}_2 & \cdots & \boldsymbol{P}_n \end{bmatrix}$$

R is $2N$ x 3 of rank 3, because N >= 3; further, **S** is 3 x n and also is of rank 3, because the n points in 3D space are not all coplanar.

3D structure and motion: 3D recovery from 2D motion

The rank theorem (proof)

- Finally, we note that given the constructions

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{i}_N^{\mathbf{T}} \\ \mathbf{j}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{j}_N^{\mathbf{T}} \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} \boldsymbol{P}_1 & \boldsymbol{P}_2 & \cdots & \boldsymbol{P}_n \end{bmatrix}$$

R is $2N$ x 3 of rank 3, because N >= 3; further, **S** is 3 x n and also is of rank 3, because the n points in 3D space are not all coplanar. $\square$

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Factorization of the (registered) measurement matrix can be accomplished via application of the SVD (recall Unit 5 interlude), i.e.,

$$\overline{\mathbf{W}} = \mathbf{U}\mathbf{D}\mathbf{V}^{T}$$

- Here, for $\overline{\mathbf{W}}$ having dimensions $2N$ x n, **U** is $2N$ x $2N$, **V** is n x n and **D** is $2N$ x n.

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Factorization of the (registered) measurement matrix can be accomplished via application of the SVD (recall Unit 5 interlude), i.e.,

$$\overline{\mathbf{W}} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathbf{T}}$$

- Here, for $\overline{\mathbf{W}}$ having dimensions $2N$ x n, **U** is $2N$ x $2N$, **V** is n x n and **D** is $2N$ x n.
- Owing to the rank constraint on $\overline{\mathbf{W}}$, however, only the first 3 singular values of **D** will be non-zero.
  - At least for the ideal case.

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Factorization of the (registered) measurement matrix can be accomplished via application of the SVD (recall Unit 5 interlude), i.e.,

$$\overline{\mathbf{W}} = \mathbf{UDV}^{\mathrm{T}}$$

- Here, for $\overline{\mathbf{W}}$ having dimensions $2N$ x n, **U** is $2N$ x $2N$, **V** is n x n and **D** is $2N$ x n.
- Owing to the rank constraint on $\overline{\mathbf{W}}$, however, only the first 3 singular values of **D** will be non-zero.
  - At least for the ideal case.
- For non-ideal (e.g., noise corrupted) cases, we can enforce the rank 3 constraint by setting all but the 3 largest singular values of **D** to 0 and similarly constraining **U** and **V**.

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Factorization of the (registered) measurement matrix can be accomplished via application of the SVD (recall Unit 5 interlude), i.e.,

$$\overline{\mathbf{W}} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

- Here, for $\overline{\mathbf{W}}$ having dimensions $2N$ x n, **U** is $2N$ x $2N$, **V** is n x n and **D** is $2N$ x n.
- Owing to the rank constraint on $\overline{\mathbf{W}}$, however, only the first 3 singular values of **D** will be non-zero.
  - At least for the ideal case.
- For non-ideal (e.g., noise corrupted) cases, we can enforce the rank 3 constraint by setting all but the 3 largest singular values of **D** to 0 and similarly constraining **U** and **V**.
- Let
  - **D**' be the 3 x 3 top left submatrix of **D** corresponding to its 3 largest singular values.
  - **U**' be the $2N$ x 3 submatrix of **U** formed by its columns corresponding to the 3 largest singular values
  - **V**' be the n x 3 submatrix of **V** formed by its columns corresponding to the 3 largest singular values

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Factorization of the (registered) measurement matrix can be accomplished via application of the SVD (recall Unit 5 interlude), i.e.,

$$\overline{\mathbf{W}} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathbf{T}}$$

- Here, for $\overline{\mathbf{W}}$ having dimensions $2N$ x n, **U** is $2N$ x $2N$, **V** is n x n and **D** is $2N$ x n.
- Owing to the rank constraint on $\overline{\mathbf{W}}$, however, only the first 3 singular values of **D** will be non-zero.
  - At least for the ideal case.
- For non-ideal (e.g., noise corrupted) cases, we can enforce the rank 3 constraint by setting all but the 3 largest singular values of **D** to 0 and similarly constraining **U** and **V**.
- Let
  - **D**' be the 3 x 3 top left submatrix of **D** corresponding to its 3 largest singular values.
  - **U**' be the $2N$ x 3 submatrix of **U** formed by its columns corresponding to the 3 largest singular values
  - **V**' be the n x 3 submatrix of **V** formed by its columns corresponding to the 3 largest singular values
- Then we take

$$\tilde{\mathbf{R}} = \mathbf{U}'\mathbf{D}'^{1/2}$$

$$\tilde{\mathbf{S}} = \mathbf{D}'^{1/2}\mathbf{V}'^{\mathbf{T}}$$

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Factorization of the (registered) measurement matrix can be accomplished via application of the SVD (recall Unit 5 interlude), i.e.,

$$\overline{\mathbf{W}} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathbf{T}}$$

- Here, for $\overline{\mathbf{W}}$ having dimensions $2N$ x n, **U** is $2N$ x $2N$, **V** is n x n and **D** is $2N$ x n.
- Owing to the rank constraint on $\overline{\mathbf{W}}$, however, only the first 3 singular values of **D** will be non-zero.
  - At least for the ideal case.
- For non-ideal (e.g., noise corrupted) cases, we can enforce the rank 3 constraint by setting all but the 3 largest singular values of **D** to 0 and similarly constraining **U** and **V**.
- Let
  - **D**' be the 3 x 3 top left submatrix of **D** corresponding to its 3 largest singular values.
  - **U**' be the $2N$ x 3 submatrix of **U** formed by its columns corresponding to the 3 largest singular values
  - **V**' be the n x 3 submatrix of **V** formed by its columns corresponding to the 3 largest singular values
- Then we take

$$\tilde{\mathbf{R}} = \mathbf{U}'\mathbf{D}'^{1/2}$$

$$\tilde{\mathbf{S}} = \mathbf{D}'^{1/2}\mathbf{V}'^{\mathbf{T}}$$

- But, we are not quite done …

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Recall that the matrix **R** was constructed as

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{i}_N^{\mathbf{T}} \\ \mathbf{j}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{j}_N^{\mathbf{T}} \end{bmatrix}$$

  such that the vectors $\mathbf{i}_i$ and $\mathbf{j}_i$ define the coordinate systems in the ith image.

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Recall that the matrix **R** was constructed as

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{i}_N^{\mathbf{T}} \\ \mathbf{j}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{j}_N^{\mathbf{T}} \end{bmatrix}$$

  such that the vectors $\mathbf{i}_i$ and $\mathbf{j}_i$ define the coordinate systems in the ith image.
- Correspondingly, two sets of constraints must be upheld
  1. The rows of **R** must have unit norm.
  2. The first n rows of **R** must be orthogonal to the corresponding last n rows of **R**.

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Recall that the matrix **R** was constructed as

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{i}_N^{\mathbf{T}} \\ \mathbf{j}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{j}_N^{\mathbf{T}} \end{bmatrix}$$

  such that the vectors $\mathbf{i}_i$ and $\mathbf{j}_i$ define the coordinate systems in the ith image.
- Correspondingly, two sets of constraints must be upheld
  1. The rows of **R** must have unit norm.
  2. The first n rows of **R** must be orthogonal to the corresponding last n rows of **R**.
- The problem arises as that there is no reason to expect

$$\overline{\mathbf{R}} = \mathbf{U}'\mathbf{D}'^{1/2}$$

  to respect these constraints, in general.

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Recall that the matrix **R** was constructed as

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{i}_N^{\mathbf{T}} \\ \mathbf{j}_1^{\mathbf{T}} \\ \vdots \\ \mathbf{j}_N^{\mathbf{T}} \end{bmatrix}$$

  such that the vectors $\mathbf{i}_i$ and $\mathbf{j}_i$ define the coordinate systems in the ith image.
- Correspondingly, two sets of constraints must be upheld
  1. The rows of **R** must have unit norm.
  2. The first n rows of **R** must be orthogonal to the corresponding last n rows of **R**.
- The problem arises as that there is no reason to expect

$$\overline{\mathbf{R}} = \mathbf{U}'\mathbf{D}'^{1/2}$$

  to respect these constraints, in general.
- Indeed, the SVD factorization is not unique: If **R** and **S** factorize **W** and **Q** is any invertible 3 x 3 matrix, then **RQ** and $\mathbf{Q}^{-1}$**S** also factorizes **W**

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

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Happily, we can take advantage of the fact that if **R** and **S** factorize **W** and **Q** is any invertible 3 x 3 matrix, then **RQ** and $\mathbf{Q}^{-1}\mathbf{S}$ also factorizes **W**

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

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Happily, we can take advantage of the fact that if **R** and **S** factorize **W** and **Q** is any invertible 3 x 3 matrix, then **RQ** and $\mathbf{Q}^{-1}\mathbf{S}$ also factorizes **W**

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

- We construct a particular **Q** that leads to the orthonormality constraints on **R** to be observed.

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Happily, we can take advantage of the fact that if **R** and **S** factorize **W** and **Q** is any invertible 3 x 3 matrix, then **RQ** and $\mathbf{Q}^{-1}\mathbf{S}$ also factorizes **W**

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

- We construct a particular **Q** that leads to the orthonormality constraints on **R** to be observed.
- In particular, we look for a **Q** such that

$$\tilde{\mathbf{i}}_i^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\tilde{\mathbf{i}}_i = 1$$
$$\tilde{\mathbf{j}}_i^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\tilde{\mathbf{j}}_i = 1$$
$$\tilde{\mathbf{i}}_i^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\tilde{\mathbf{j}}_i = 0$$

3D structure and motion: 3D recovery from 2D motion

The factorization algorithm

- Happily, we can take advantage of the fact that if **R** and **S** factorize **W** and **Q** is any invertible 3 x 3 matrix, then **RQ** and $\mathbf{Q}^{-1}\mathbf{S}$ also factorizes **W**

$$(\mathbf{RQ})(\mathbf{Q}^{-1}\mathbf{S}) = \mathbf{R}(\mathbf{Q}\mathbf{Q}^{-1})\mathbf{S} = \mathbf{RS} = \mathbf{W}$$

- We construct a particular **Q** that leads to the orthonormality constraints on **R** to be observed.
- In particular, we look for a **Q** such that

$$\begin{aligned} \tilde{\mathbf{i}}_i^{\mathbf{T}}\mathbf{Q}\mathbf{Q}^{\mathbf{T}}\tilde{\mathbf{i}}_i &= 1 \\ \tilde{\mathbf{j}}_i^{\mathbf{T}}\mathbf{Q}\mathbf{Q}^{\mathbf{T}}\tilde{\mathbf{j}}_i &= 1 \\ \tilde{\mathbf{i}}_i^{\mathbf{T}}\mathbf{Q}\mathbf{Q}^{\mathbf{T}}\tilde{\mathbf{j}}_i &= 0 \end{aligned}$$

and then define new matrices

$$\begin{aligned} \mathbf{R} &= \tilde{\mathbf{R}}\mathbf{Q} \\ \mathbf{S} &= \mathbf{Q}^{-1}\tilde{\mathbf{S}} \end{aligned}$$

which will still factorize **W**, but with the orthonormality constraints on **R** enforced.

3D structure and motion: 3D recovery from 2D motion

Factorization: A few loose ends

- How do we actually recover **Q**?

3D structure and motion: 3D recovery from 2D motion

Factorization: A few loose ends

- How do we actually recover **Q**?
  - The constraint equations for **Q** are non-linear, but over constrained & otherwise unremarkable.
  - Any reliable, standard non-linear solver could be applied (e.g., Newton's method).

3D structure and motion: 3D recovery from 2D motion

Factorization: A few loose ends

- How do we actually recover **Q**?
  - The constraint equations for **Q** are non-linear, but overconstrained & otherwise unremarkable.
  - Any reliable, standard non-linear solver could be applied (e.g., Newton’s method).
- Isn’t there still an ambiguity in the definition of **Q**? For any rotation matrix **G**, isn’t it still the case that we could write

$$(\mathbf{RQG})(\mathbf{G}^{-T}\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQG})(\mathbf{G}^{-1}\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQ})(\mathbf{GG}^{-1})(\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQ})(\mathbf{Q}^{-1}\mathbf{S}) = \mathbf{R}(\mathbf{QQ}^{-1})\mathbf{S} = \mathbf{RS} = \mathbf{W}$$

3D structure and motion: 3D recovery from 2D motion

Factorization: A few loose ends

- How do we actually recover **Q**?
  - The constraint equations for **Q** are non-linear, but overconstrained & otherwise unremarkable.
  - Any reliable, standard non-linear solver could be applied (e.g., Newton's method).
- Isn't there still an ambiguity in the definition of **Q**? For any rotation matrix **G**, isn't it still the case that we could write

$$(\mathbf{RQG})(\mathbf{G}^{-T}\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQG})(\mathbf{G}^{-1}\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQ})(\mathbf{G}\mathbf{G}^{-1})(\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQ})(\mathbf{Q}^{-1}\mathbf{S}) = \mathbf{R}(\mathbf{Q}\mathbf{Q}^{-1})\mathbf{S} = \mathbf{RS} = \mathbf{W}$$

  - Yes! The ambiguity can be interpreted as a lack of knowledge of the original rotation that aligns the image and world coordinate systems at the first frame. We might choose to enforce that two coordinates systems are aligned at the first frame.
  - In any case, the relative frame-to-frame motions following the first frame are unambiguously specified.
  - Notice, BTW, that **G** must be a rotation matrix; otherwise, it would destroy the orthnormality imposed by **Q**.

3D structure and motion: 3D recovery from 2D motion

Factorization: A few loose ends

- How do we actually recover **Q**?
  - The constraint equations for **Q** are non-linear, but overconstrained & otherwise unremarkable.
  - Any reliable, standard non-linear solver could be applied (e.g., Newton's method).
- Isn't these still an ambiguity in the definition of **Q**? For any rotation matrix **G**, isn't it still the case that we could write

$$(\mathbf{RQG})(\mathbf{G}^{-T}\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQG})(\mathbf{G}^{-1}\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQ})(\mathbf{G}\mathbf{G}^{-1})(\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQ})(\mathbf{Q}^{-1}\mathbf{S}) = \mathbf{R}(\mathbf{Q}\mathbf{Q}^{-1})\mathbf{S} = \mathbf{RS} = \mathbf{W}$$

  - Yes! The ambiguity can be interpreted as a lack of knowledge of the original rotation that aligns the image and world coordinate systems at the first frame. We might choose to enforce that two coordinates systems are aligned at the first frame.
  - In any case, the relative frame-to-frame motions following the first frame are unambiguously specified.
  - Notice, BTW, that **G** must be a rotation matrix; otherwise, it would destroy the orthnormality imposed by **Q**.
- Why do we refer to **R** as the motion matrix?

3D structure and motion: 3D recovery from 2D motion

Factorization: A few loose ends

- How do we actually recover **Q**?
  - The constraint equations for **Q** are non-linear, but overconstrained & otherwise unremarkable.
  - Any reliable, standard non-linear solver could be applied (e.g., Newton's method).
- Isn't these still an ambiguity in the definition of **Q**? For any rotation matrix **G**, isn't it still the case that we could write

$$(\mathbf{RQG})(\mathbf{G}^{-T}\mathbf{Q}^{-1}\mathbf{S})=(\mathbf{RQG})(\mathbf{G}^{-1}\mathbf{Q}^{-1}\mathbf{S})=(\mathbf{RQ})(\mathbf{GG}^{-1})(\mathbf{Q}^{-1}\mathbf{S})=(\mathbf{RQ})(\mathbf{Q}^{-1}\mathbf{S})=\mathbf{R}(\mathbf{QQ}^{-1})\mathbf{S}=\mathbf{RS}=\mathbf{W}$$

  - Yes! The ambiguity can be interpreted as a lack of knowledge of the original rotation that aligns the image and world coordinate systems at the first frame. We might choose to enforce that two coordinates systems are aligned at the first frame.
  - In any case, the relative frame-to-frame motions following the first frame are unambiguously specified.
  - Notice, BTW, that **G** must be a rotation matrix; otherwise, it would destroy the orthnormality imposed by **Q**.
- Why do we refer to **R** as the motion matrix?
  - The first N rows of **R** can be interpreted as the first rows of the rotation matrices that transform to the camera coordinate systems; the second N rows of **R** can be interpreted as the second rows of these transformations.

$$\boldsymbol{p}_{\boldsymbol{ij}}=\begin{pmatrix}x_{ij}\\ y_{ij}\end{pmatrix}=\mathbf{R}^{2x3}\boldsymbol{P}_j=\begin{pmatrix}\mathbf{i}_i^{\mathbf{T}}\\ \mathbf{j}_i^{\mathbf{T}}\end{pmatrix}\boldsymbol{P}_j$$

3D structure and motion: 3D recovery from 2D motion

Factorization: A few loose ends

- How do we actually recover **Q**?
  - The constraint equations for **Q** are non-linear, but overconstrained & otherwise unremarkable.
  - Any reliable, standard non-linear solver could be applied (e.g., Newton's method).
- Isn't these still an ambiguity in the definition of **Q**? For any rotation matrix **G**, isn't it still the case that we could write

$$(\mathbf{RQG})(\mathbf{G}^{-T}\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQG})(\mathbf{G}^{-1}\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQ})(\mathbf{GG}^{-1})(\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQ})(\mathbf{Q}^{-1}\mathbf{S}) = \mathbf{R}(\mathbf{QQ}^{-1})\mathbf{S} = \mathbf{RS} = \mathbf{W}$$

  - Yes! The ambiguity can be interpreted as a lack of knowledge of the original rotation that aligns the image and world coordinate systems at the first frame. We might choose to enforce that two coordinates systems are aligned at the first frame.
  - In any case, the relative frame-to-frame motions following the first frame are unambiguously specified.
  - Notice, BTW, that **G** must be a rotation matrix; otherwise, it would destroy the orthnormality imposed by **Q**.
- Why do we refer to **R** as the motion matrix?
  - The first N rows of **R** can be interpreted as the first rows of the rotation matrices that transform to the camera coordinate systems; the second N rows of **R** can be interpreted as the second rows of these transformations.

$$\boldsymbol{p_{ij}} = \begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} = \mathbf{R}^{2x3}\boldsymbol{P_j} = \begin{pmatrix} \mathbf{i}_i^{\mathbf{T}} \\ \mathbf{j}_i^{\mathbf{T}} \end{pmatrix}\boldsymbol{P_j}$$

- What about translation?

3D structure and motion: 3D recovery from 2D motion

Factorization: A few loose ends

- How do we actually recover **Q**?
  - The constraint equations for **Q** are non-linear, but overconstrained & otherwise unremarkable.
  - Any reliable, standard non-linear solver could be applied (e.g., Newton's method).
- Isn't these still an ambiguity in the definition of **Q**? For any rotation matrix **G**, isn't it still the case that we could write

$$(\mathbf{RQG})(\mathbf{G}^{-T}\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQG})(\mathbf{G}^{-1}\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQ})(\mathbf{GG}^{-1})(\mathbf{Q}^{-1}\mathbf{S}) = (\mathbf{RQ})(\mathbf{Q}^{-1}\mathbf{S}) = \mathbf{R}(\mathbf{QQ}^{-1})\mathbf{S} = \mathbf{RS} = \mathbf{W}$$

  - Yes! The ambiguity can be interpreted as a lack of knowledge of the original rotation that aligns the image and world coordinate systems at the first frame. We might choose to enforce that two coordinates systems are aligned at the first frame.
  - In any case, the relative frame-to-frame motions following the first frame are unambiguously specified.
  - Notice, BTW, that **G** must be a rotation matrix; otherwise, it would destroy the orthnormality imposed by **Q**.
- Why do we refer to **R** as the motion matrix?
  - The first N rows of **R** can be interpreted as the first rows of the rotation matrices that transform to the camera coordinate systems; the second N rows of **R** can be interpreted as the second rows of these transformations.

$$\boldsymbol{p}_{ij} = \begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} = \mathbf{R}^{2x3}\boldsymbol{P}_j = \begin{pmatrix} \mathbf{i}_i^{\mathbf{T}} \\ \mathbf{j}_i^{\mathbf{T}} \end{pmatrix} \boldsymbol{P}_j$$

- What about translation?
  - The X, Y components of the translations (i.e., those parallel to the image planes) are captured by the shifts of the centroids that we remove during registration. The Z components cannot be recovered under orthographic projection.

3D structure and motion: Approaches

Problem statement

- Given image motion
  - Optical flow-based
  - Finite displacement-based
- Seek to recover
  - 3D motion (translation and rotation)
  - 3D structure (geometric layout of environment)

Finite displacement approach

- Seeks to recover
  - full 3D rotation and translation
  - 3D structure
- Typically used in conjunction with finite displacement image motion estimates.

Infinitesimal approach

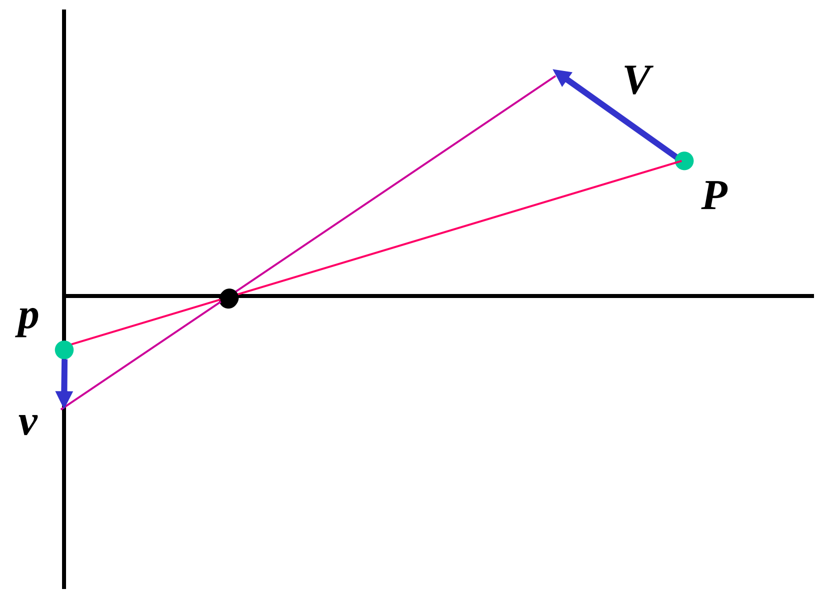
- Seeks to recover
  - 3D rotational and translation velocity (temporal derivatives of full rotation and translation)
  - 3D structure
- Typically used in conjunction with optical flow image motion estimates.

3D structure and motion: Motivation

Where we are

- Earlier, we suggested how the motion of a point in the world yields corresponding motion in the image.
  - Subject to differences between the motion field and the optical flow.
- Since then, we have developed methods to recover the image motion in terms of optical flow.
  - Gradient-based
  - Finite displacement-based.
- We also have an approach to recovering structure and motion using finite displacements.

.



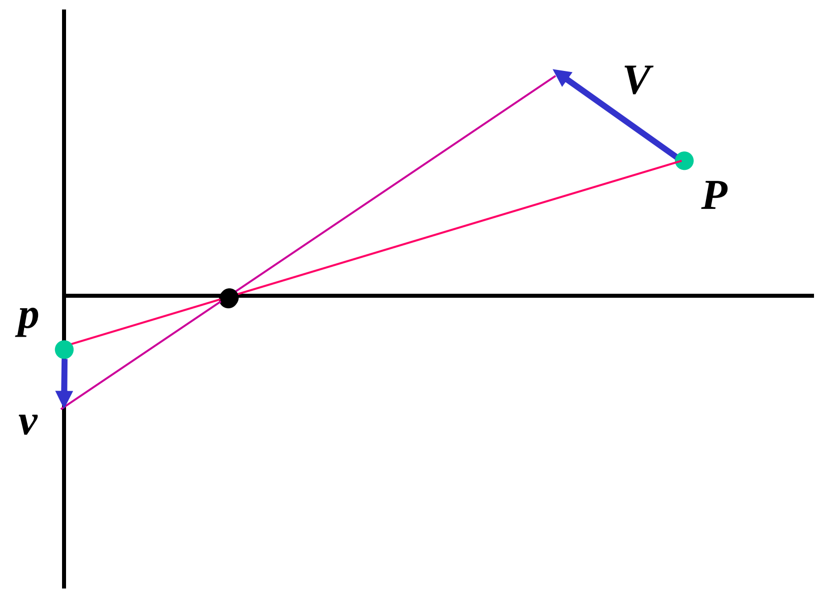
3D structure and motion: Motivation

Where we are

- Earlier, we suggested how the motion of a point in the world yields corresponding motion in the image.
  - Subject to differences between the motion field and the optical flow.
- Since then, we have developed methods to recover the image motion in terms of optical flow.
  - Gradient-based
  - Finite displacement-based
- We also have an approach to recovering structure and motion using finite displacements.

Where we shall go

- Given
  - Our ability to recover motion in the image in terms of infinitesimals
  - And the fact that the image motion arose from 3D structure and motion in the scene
- We now ask how we can “invert” the process to recover the 3D scene parameters from the image motion, all in terms of infinitesimals..



3D structure and motion: Basics

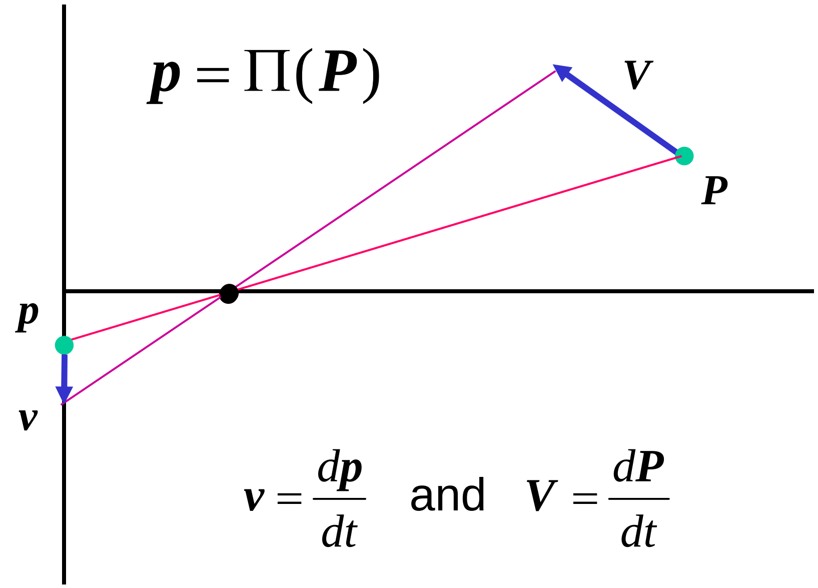
Plan of attack

- We seek to recover estimates of
  - 3D velocity (rotational and translational)
  - 3D structure (distance, Z)

  from image motion
  - Optical flow
- We must derive an explicit relationship that relates the variables of interest.
- Then we can exploit this relationship in a recovery process.
- As a point of departure, we have the diagram at the right.
- Now we must be explicit about

$$\frac{d\boldsymbol{p}}{dt} = \frac{d\Pi(\boldsymbol{P})}{dt}$$

- This requires
  1. Being precise about 3D structure and motion
  2. Deriving their image correlates



$$\boldsymbol{v} = \frac{d\boldsymbol{p}}{dt} \quad \text{and} \quad \boldsymbol{V} = \frac{d\boldsymbol{P}}{dt}$$

3D structure and motion: Basics

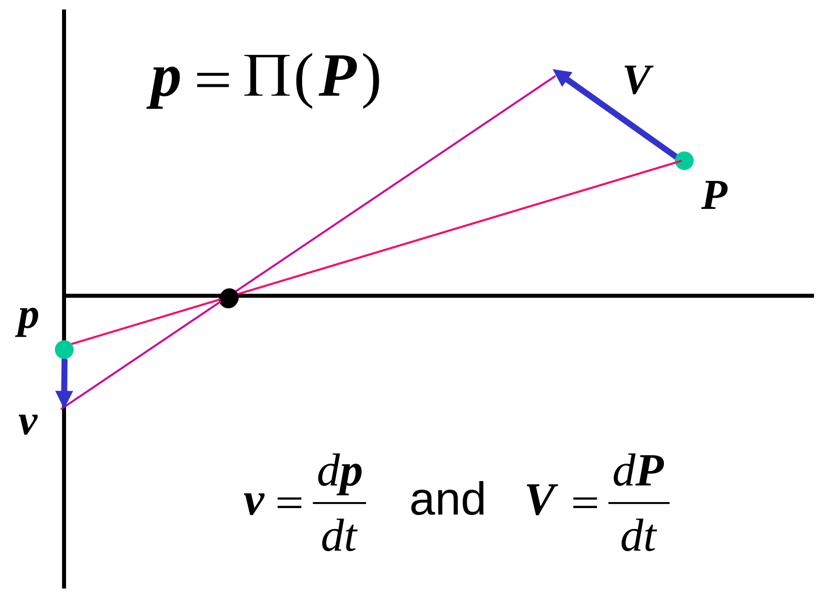
Plan of attack

- We seek to recover estimates of
  - 3D velocity (rotational and translational)
  - 3D structure (distance, Z)

  from image motion
  - Optical flow
- We must derive an explicit relationship that relates the variables of interest.
- Then we can exploit this relationship in a recovery process.
- As a point of departure, we have the diagram at the right.
- Now we must be explicit about

$$\frac{d\boldsymbol{p}}{dt} = \frac{d\Pi(\boldsymbol{P})}{dt}$$

- This requires
  1. Being precise about 3D structure and motion
  2. Deriving their image correlates



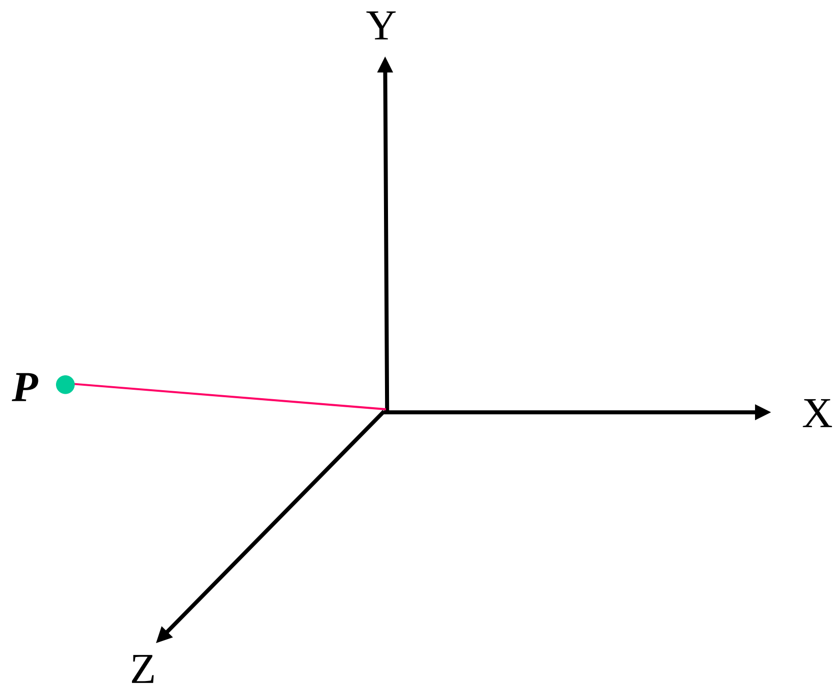
$$\boldsymbol{p} = \Pi(\boldsymbol{P})$$

$$\boldsymbol{v} = \frac{d\boldsymbol{p}}{dt} \quad \text{and} \quad \boldsymbol{V} = \frac{d\boldsymbol{P}}{dt}$$

3D structure and motion: Basics

3D motion of a point in the scene

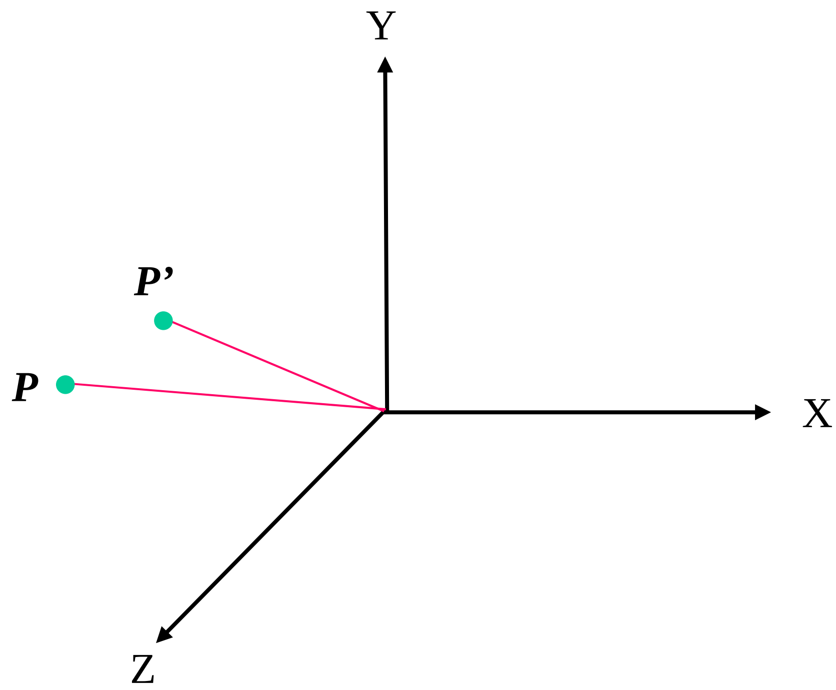
- Given a point $\boldsymbol{P} = (X,Y,Z)^{\top}$ in space



3D structure and motion: Basics

3D motion of a point in the scene

- Given a point $\boldsymbol{P} = (X,Y,Z)^{\mathrm{T}}$ in space
- Its motion to $\boldsymbol{P'} = (X',Y',Z')^{\mathrm{T}}$ can be decomposed into 2 parts



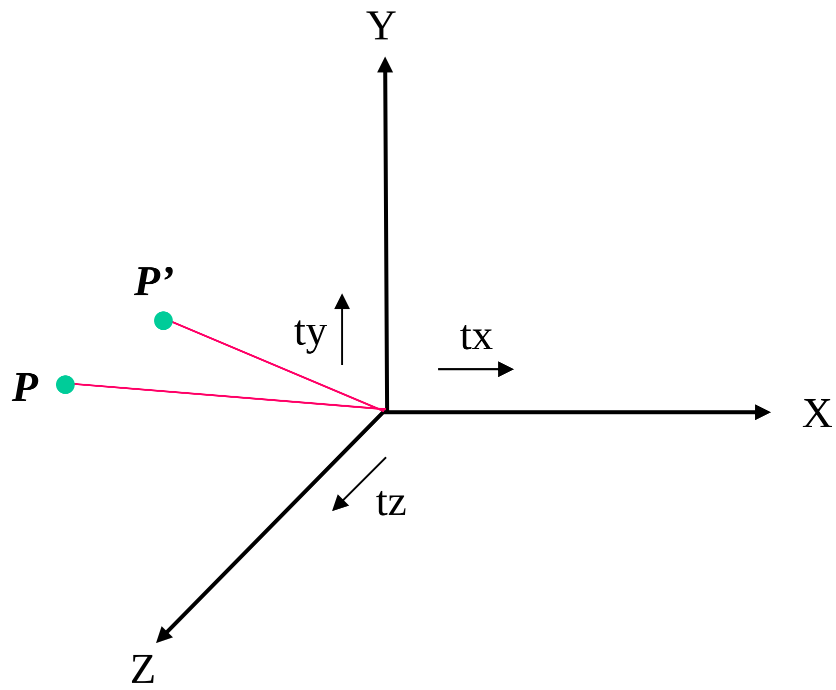
3D structure and motion: Basics

3D motion of a point in the scene

- Given a point $\boldsymbol{P} = (X,Y,Z)^{\mathsf{T}}$ in space
- Its motion to $\boldsymbol{P'} = (X',Y',Z')^{\mathsf{T}}$ can be decomposed into 2 parts

1. Translation about the three coordinate axes

$$\mathbf{T} = (t_x, t_y, t_z)^{\mathsf{T}}$$



3D structure and motion: Basics

3D motion of a point in the scene

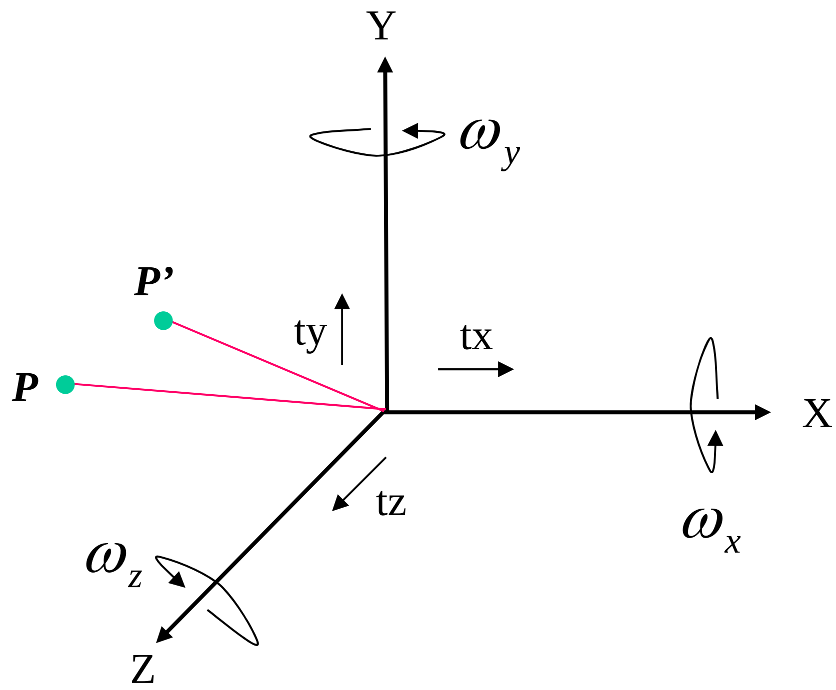
- Given a point $\boldsymbol{P} = (X,Y,Z)^{\mathsf{T}}$ in space
- Its motion to $\boldsymbol{P'} = (X',Y',Z')^{\mathsf{T}}$ can be decomposed into 2 parts

1. Translation about the three coordinate axes

$$\mathbf{T} = (t_x, t_y, t_z)^{\mathsf{T}}$$

2. Rotation about the three coordinates axes

$$\mathbf{\Omega} = (\omega_x, \omega_y, \omega_z)^{\mathsf{T}}$$



3D structure and motion: Basics

3D motion of a point in the scene

- Given a point $\boldsymbol{P} = (X,Y,Z)^{\mathsf{T}}$ in space
- Its motion to $\boldsymbol{P'} = (X',Y',Z')^{\mathsf{T}}$ can be decomposed into 2 parts

1. Translation about the three coordinate axes

$$\mathbf{T} = (t_x, t_y, t_z)^{\mathsf{T}}$$

2. Rotation about the three coordinates axes

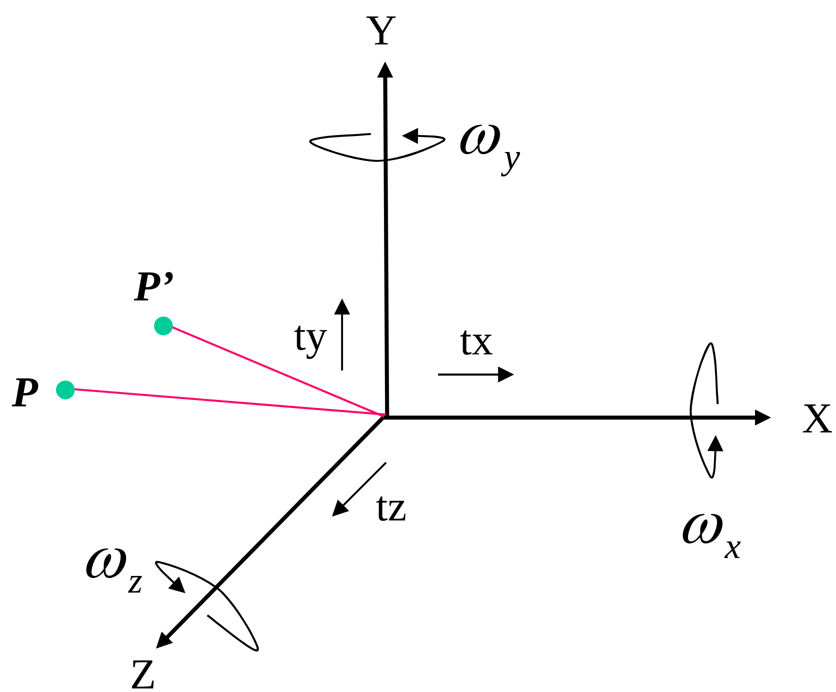
$$\mathbf{\Omega} = (\omega_x, \omega_y, \omega_z)^{\mathsf{T}}$$

- We have

$$\boldsymbol{P'} = \mathbf{R}(\Omega)\boldsymbol{P} + \mathbf{T}$$

with

– $\mathbf{R}(\mathbf{\Omega})$ the rotation matrix corresponding to $\mathbf{\Omega}$



3D structure and motion: Basics

Rotation

- An arbitrary rotation is given as

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega_x & -\sin\omega_x \\ 0 & \sin\omega_x & \cos\omega_x \end{pmatrix} \begin{pmatrix} \cos\omega_y & 0 & \sin\omega_y \\ 0 & 1 & 0 \\ -\sin\omega_y & 0 & \cos\omega_y \end{pmatrix} \begin{pmatrix} \cos\omega_z & -\sin\omega_z & 0 \\ \sin\omega_z & \cos\omega_z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos\omega_y \cos\omega_z & -\cos\omega_y \sin\omega_z & \sin\omega_y \\ \sin\omega_x \sin\omega_y \cos\omega_z + \cos\omega_x \sin\omega_z & -\sin\omega_x \sin\omega_y \sin\omega_z + \cos\omega_x \cos\omega_z & -\sin\omega_x \cos\omega_y \\ -\cos\omega_x \sin\omega_y \cos\omega_z + \sin\omega_x \sin\omega_z & \cos\omega_x \sin\omega_y \sin\omega_z + \sin\omega_x \cos\omega_z & \cos\omega_x \cos\omega_z \end{pmatrix}$$

3D structure and motion: Basics

Rotation

- An arbitrary rotation is then given as

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega_x & -\sin\omega_x \\ 0 & \sin\omega_x & \cos\omega_x \end{pmatrix} \begin{pmatrix} \cos\omega_y & 0 & \sin\omega_y \\ 0 & 1 & 0 \\ -\sin\omega_y & 0 & \cos\omega_y \end{pmatrix} \begin{pmatrix} \cos\omega_z & -\sin\omega_z & 0 \\ \sin\omega_z & \cos\omega_z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Since we are dealing with velocity, we take the infinitesimal approximations

$$\cos\theta = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \cdots \approx 1$$

$$\sin\theta = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \cdots \approx \theta$$

3D structure and motion: Basics

Rotation

- An arbitrary rotation is then given as

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega_x & -\sin\omega_x \\ 0 & \sin\omega_x & \cos\omega_x \end{pmatrix} \begin{pmatrix} \cos\omega_y & 0 & \sin\omega_y \\ 0 & 1 & 0 \\ -\sin\omega_y & 0 & \cos\omega_y \end{pmatrix} \begin{pmatrix} \cos\omega_z & -\sin\omega_z & 0 \\ \sin\omega_z & \cos\omega_z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Since we are dealing with velocity, we take the infinitesimal approximations

$$\cos\theta \approx 1$$

$$\sin\theta \approx \theta$$

- So that we have

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\omega_x \\ 0 & \omega_x & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & \omega_y \\ 0 & 1 & 0 \\ -\omega_y & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -\omega_z & 0 \\ \omega_z & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \approx \begin{pmatrix} 1 & -\omega_z & \omega_y \\ \omega_z & 1 & -\omega_x \\ -\omega_y & \omega_x & 1 \end{pmatrix}$$

3D structure and motion: Basics

Now we can explicitly calculate a change in 3D position

- The new position $\boldsymbol{P'}$ following infinitesimal rotation and translation of $\boldsymbol{P}$ is

$$\mathbf{R}\boldsymbol{P} + \boldsymbol{T} = \begin{pmatrix} 1 & -\omega_z & \omega_y \\ \omega_z & 1 & -\omega_x \\ -\omega_y & \omega_x & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \begin{pmatrix} X - \omega_z Y + \omega_y Z + t_x \\ \omega_z X + Y - \omega_x Z + t_y \\ -\omega_y X + \omega_x Y + Z + t_z \end{pmatrix} = \boldsymbol{P}'$$

3D structure and motion: Basics

Now we can explicitly calculate a change in 3D position

- The new position P' following infinitesimal rotation and translation of P is

$$\mathbf{R}\boldsymbol{P}+\boldsymbol{T}=\begin{pmatrix}1 & -\omega_z & \omega_y\\ \omega_z & 1 & -\omega_x\\ -\omega_y & \omega_x & 1\end{pmatrix}\begin{pmatrix}X\\ Y\\ Z\end{pmatrix}+\begin{pmatrix}t_x\\ t_y\\ t_z\end{pmatrix}=\begin{pmatrix}X-\omega_z Y+\omega_y Z+t_x\\ \omega_z X+Y-\omega_x Z+t_y\\ -\omega_y X+\omega_x Y+Z+t_z\end{pmatrix}=\boldsymbol{P}'$$

3D velocity also is at hand

- Since velocity is the infinitesimal change in position with time we have

$$\boldsymbol{P}-\boldsymbol{P}'=\begin{pmatrix}X\\ Y\\ Z\end{pmatrix}-\begin{pmatrix}X-\omega_z Y+\omega_y Z+t_x\\ \omega_z X+Y-\omega_x Z+t_y\\ -\omega_y X+\omega_x Y+Z+t_z\end{pmatrix}=\begin{pmatrix}\omega_z Y-\omega_y Z-t_x\\ -\omega_z X+\omega_x Z-t_y\\ \omega_y X-\omega_x Y-t_z\end{pmatrix}=\boldsymbol{V}$$

with $\boldsymbol{V}=(U,V,W)$ 3D velocity.

3D structure and motion: Basics

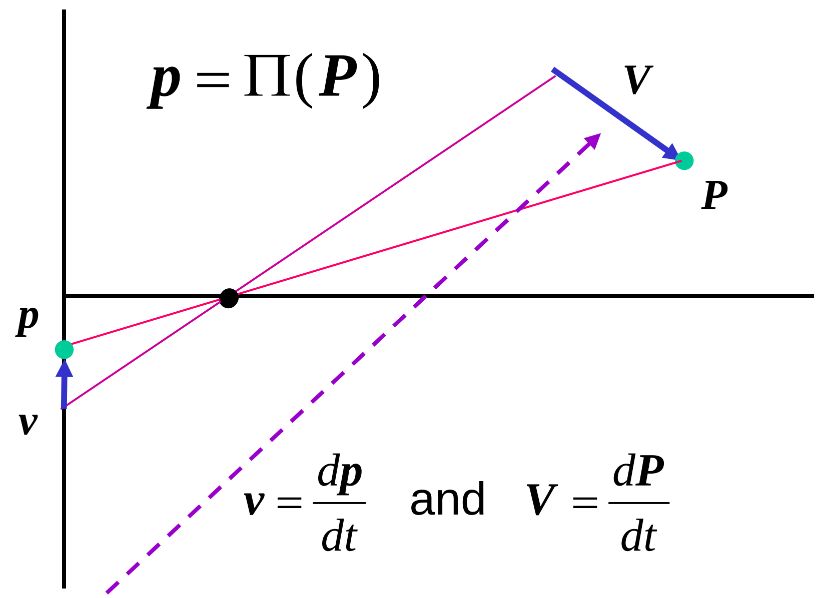
Where we stand

- We seek to recover estimates of
  - 3D velocity (rotational and translational)
  - 3D structure (distance, Z)

  from image motion
  - Optical flow
- We must derive an explicit relationship that relates the variables of interest.
- Then we can exploit this relationship in a recovery process.
- As a point of departure, we have the diagram at the left.
- Now we must be explicit about

$$\frac{d\boldsymbol{p}}{dt} = \frac{d\Pi(\boldsymbol{P})}{dt}$$

- This requires
  1. Being precise about 3D structure and motion
  2. Deriving their image correlates



$$\boldsymbol{v} = \frac{d\boldsymbol{p}}{dt} \quad \text{and} \quad \boldsymbol{V} = \frac{d\boldsymbol{P}}{dt}$$

3D structure and motion: Basics

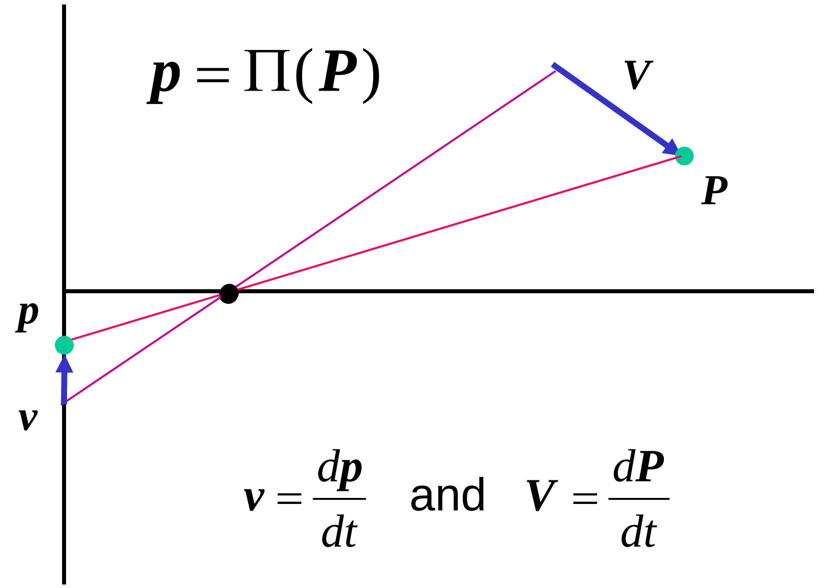
Where we stand

- We seek to recover estimates of
  - 3D velocity (rotational and translational)
  - 3D structure (distance, Z)

  from image motion
  - Optical flow
- We must derive an explicit relationship that relates the variables of interest.
- Then we can exploit this relationship in a recovery process.
- As a point of departure, we have the diagram at the left.
- Now we must be explicit about

$$\frac{d\boldsymbol{p}}{dt} = \frac{d\Pi(\boldsymbol{P})}{dt}$$

- This requires
  1. Being precise about 3D structure and motion
  2. Deriving their image correlates



3D structure and motion: Basics

Image correlate of 3D structure and motion

- We seek an expression for

$$\frac{d\boldsymbol{p}}{dt} = \begin{pmatrix} dx/dt \\ dy/dt \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix}$$

where we follow Newton's "dot" convention for a temporal derivative.

3D structure and motion: Basics

Image correlate of 3D structure and motion

- We seek an expression for

$$\frac{d\boldsymbol{p}}{dt} = \begin{pmatrix} dx/dt \\ dy/dt \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix}$$

  where we follow Newton's "dot" convention for a temporal derivative.

- Let us follow perspective projection with f=1, hence

$$x = \frac{X}{Z} \qquad y = \frac{Y}{Z}$$

3D structure and motion: Basics

Image correlate of 3D structure and motion

- We seek an expression for

$$\frac{d\mathbf{p}}{dt} = \begin{pmatrix} dx/dt \\ dy/dt \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix}$$

  where we follow Newton's "dot" convention for a temporal derivative.

- Let us follow perspective projection with f=1, hence

$$x = \frac{X}{Z} \qquad y = \frac{Y}{Z}$$

- So, we have

$$\dot{x} = u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} \qquad \dot{y} = v = \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2}$$

3D structure and motion: Basics

Image correlate of 3D structure and motion

- We have

$$\dot{x} = u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} \qquad \dot{y} = v = \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2}$$

- As well as

$$\boldsymbol{V} = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} \omega_z Y - \omega_y Z - t_x \\ -\omega_z X + \omega_x Z - t_y \\ \omega_y X - \omega_x Y - t_z \end{pmatrix}$$

3D structure and motion: Basics

Image correlate of 3D structure and motion

- We have

$$\dot{x} = u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} \qquad \dot{y} = v = \frac{\dot{Y}}{Z} - \frac{X\dot{Z}}{Z^2}$$

- As well as

$$\boldsymbol{V} = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} \omega_z Y - \omega_y Z - t_x \\ -\omega_z X + \omega_x Z - t_y \\ \omega_y X - \omega_x Y - t_z \end{pmatrix}$$

- Upon substitution we find that

$$u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = \frac{1}{Z}(\omega_z Y - \omega_y Z - t_x) - \frac{X}{Z^2}(\omega_y X - \omega_x Y - t_z)$$

3D structure and motion: Basics

Image correlate of 3D structure and motion

- We have

$$\dot{x} = u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} \qquad \dot{y} = v = \frac{\dot{Y}}{Z} - \frac{X\dot{Z}}{Z^2}$$

- As well as

$$\boldsymbol{V} = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} \omega_z Y - \omega_y Z - t_x \\ -\omega_z X + \omega_x Z - t_y \\ \omega_y X - \omega_x Y - t_z \end{pmatrix}$$

- Upon substitution we find that (dividing through by Z and simplifying X/Z, Y/Z, Z/Z)

$$u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = \frac{1}{Z}(\omega_z Y - \omega_y Z - t_x) - \frac{X}{Z^2}(\omega_y X - \omega_x Y - t_z)$$

$$= (\omega_z y - \omega_y - \frac{t_x}{Z}) - \frac{X}{Z}(\omega_y x - \omega_x y - \frac{t_z}{Z})$$

3D structure and motion: Basics

Image correlate of 3D structure and motion

- We have

$$\dot{x} = u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} \qquad \dot{y} = v = \frac{\dot{Y}}{Z} - \frac{X\dot{Z}}{Z^2}$$

- As well as

$$\boldsymbol{V} = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} \omega_z Y - \omega_y Z - t_x \\ -\omega_z X + \omega_x Z - t_y \\ \omega_y X - \omega_x Y - t_z \end{pmatrix}$$

- Upon substitution we find that (simplifying X/Z)

$$u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = (\omega_z y - \omega_y - \frac{t_x}{Z}) - \frac{X}{Z}(\omega_y x - \omega_x y - \frac{t_z}{Z})$$

$$= (\omega_z y - \omega_y - \frac{t_x}{Z}) - x(\omega_y x - \omega_x y - \frac{t_z}{Z})$$

3D structure and motion: Basics

Image correlate of 3D structure and motion

- We have

$$\dot{x} = u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} \qquad \dot{y} = v = \frac{\dot{Y}}{Z} - \frac{X\dot{Z}}{Z^2}$$

- As well as

$$\boldsymbol{V} = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} \omega_z Y - \omega_y Z - t_x \\ -\omega_z X + \omega_x Z - t_y \\ \omega_y X - \omega_x Y - t_z \end{pmatrix}$$

- Upon substitution we find that (rearranging)

$$u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = (\omega_z y - \omega_y - \frac{t_x}{Z}) - x(\omega_y x - \omega_x y - \frac{t_z}{Z})$$

$$= \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2 + 1) + \omega_z(y)$$

3D structure and motion: Basics

Image correlate of 3D structure and motion

- We have

$$\dot{x} = u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} \qquad \dot{y} = v = \frac{\dot{Y}}{Z} - \frac{X\dot{Z}}{Z^2}$$

- As well as

$$\boldsymbol{V} = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} \omega_z Y - \omega_y Z - t_x \\ -\omega_z X + \omega_x Z - t_y \\ \omega_y X - \omega_x Y - t_z \end{pmatrix}$$

- Upon substitution we find that (and similarly for *v*)

$$u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v = \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

3D structure and motion: Basics

We have achieved our (interim) goal

- We have an explicit relationship that relates
  - image velocity $(u(x,y),v(x,y))$
  - 3D parameters of structure, $Z(x,y)$
  - 3D motion, $(\omega_x \omega_y, \omega_z), (t_x, t_y, t_z)$

$$u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v = \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

3D structure and motion: Basics

We have achieved our (interim) goal

- We have an explicit relationship that relates
  - image velocity ($u(x,y), v(x,y)$)
  - 3D parameters of structure, $Z(x,y)$
  - 3D motion, $(\omega_x \omega_y, \omega_z), (t_x, t_y, t_z)$

$$\left.\begin{aligned} u &= \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y) \\ v &= \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x) \end{aligned}\right\} \text{Equations of the Motion field}$$

3D structure and motion: Basics

We have achieved our (interim) goal

- We have an explicit relationship that relates
  - image velocity ($u(x,y), v(x,y)$)
  - 3D parameters of structure, $Z(x,y)$
  - 3D motion, $(\omega_x \omega_y, \omega_z), (t_x, t_y, t_z)$

$$
\left.
\begin{aligned}
u &= \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2 + 1) + \omega_z(y) \\
v &= \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2 + 1) - \omega_y(xy) - \omega_z(x)
\end{aligned}
\right\} \text{Equations of the Motion field}
$$

Observations

- The rotation and translation components do not directly interact.
- The structure component interacts directly only with the translation component.
- If we know the 3D motion parameters and have recovered the flow (u,v), then it is trivial to recover Z (modulo noise)
- However, we often have little (or no) knowledge of the 3D motion.

3D structure and motion: 3D recovery from 2D motion

Where we stand

- We have developed methods to recover image motion (optical flow) from a time varying image sequence.
- We have derived an explicit relationship to relate the image motion to the corresponding 3D structure and motion.

3D structure and motion: 3D recovery from 2D motion

Where we stand

- We have developed methods to recover image motion (optical flow) from a time varying image sequence.
- We have derived an explicit relationship to relate the image motion to the corresponding 3D structure and motion.

Where we will go

- We seek an approach to the recovery of 3D structure and motion from the estimated image motion.
- In general, this is a very difficult problem.
  - Although approaches have been developed
  - With varying success
- We will consider 2 special cases.

1. Recovery of 3D structure, Z(x,y), given motion parameters
2. Recovery of 3D rotation, assuming no translation.

3D structure and motion: 3D recovery from 2D motion

Where we stand

- We have developed methods to recover image motion (optical flow) from a time varying image sequence.
- We have derived an explicit relationship to relate the image motion to the corresponding 3D structure and motion.

Where we will go

- We seek an approach to the recovery of 3D structure and motion from the estimated image motion.
- In general, this is a very difficult problem.
  - Although approaches have been developed
  - With varying success
- We will consider 2 special cases.

1. Recovery of 3D structure, Z(x,y), given motion parameters
2. Recovery of 3D rotation, assuming no translation.

3D structure and motion: 3D structure

We assume that the 3D motion parameters are known

- We have two equations in one unknown!

$$u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v = \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

3D structure and motion: 3D structure

We assume that the 3D motion parameters are known

- We have two equations in one unknown!

$$u = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v = \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

- We “derotate” the optical flow

$$u - \omega_x(xy) + \omega_y(x^2+1) - \omega_z(y) = \frac{1}{Z}(xt_z - t_x)$$

$$v - \omega_x(y^2+1) + \omega_y(xy) + \omega_z(x) = \frac{1}{Z}(yt_z - t_y)$$

3D structure and motion: 3D structure

We assume that the 3D motion parameters are known

- From the derotated flow we can calculate Z, perhaps averaging the two recovered values

$$u-\omega_x(xy)+\omega_y(x^2+1)-\omega_z(y)=\frac{1}{Z}(xt_z-t_x)$$

$$v-\omega_x(y^2+1)+\omega_y(xy)+\omega_z(x)=\frac{1}{Z}(yt_z-t_y)$$

$$\Rightarrow$$

$$Z(x,y)=\frac{xt_z-t_x}{u-\omega_x(xy)+\omega_y(x^2+1)-\omega_z(y)}$$

$$Z(x,y)=\frac{yt_z-t_y}{v-\omega_x(y^2+1)+\omega_y(xy)+\omega_z(x)}$$

3D structure and motion: 3D recovery from 2D motion

Where we stand

- We have developed methods to recover image motion (optical flow) from a time varying image sequence.
- We have derived an explicit relationship to relate the image motion to the corresponding 3D structure and motion.

Where we will go

- We seek an approach to the recovery of 3D structure and motion from the estimated image motion.
- In general, this is a very difficult problem.
  - Although approach have been developed
  - With varying success
- We will consider 2 special cases.

1. Recovery of 3D structure, Z(x,y), given motion parameters
2. Recovery of 3D rotation, assuming no translation.

3D structure and motion: 3D rotation

We assume purely 3D rotation

- We specialize the general case

$$u = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

to

$$u = \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v = \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

3D structure and motion: 3D rotation

We assume purely 3D rotation

- We specialize the general case

$$u = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2 + 1) + \omega_z(y)$$

$$v = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2 + 1) - \omega_y(xy) - \omega_z(x)$$

to

$$u = \omega_x(xy) - \omega_y(x^2 + 1) + \omega_z(y)$$

$$v = \omega_x(y^2 + 1) - \omega_y(xy) - \omega_z(x)$$

Remarks

- We recall that flow due to purely 3D rotation is independent of 3D scene structure.
- We see that given 3 measurements alone, we could recover the rotational velocity
- But this would be naively prone to noise sensitivity
  - So we follow another path.

3D structure and motion: 3D rotation

Error formulation

- We seek rotational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of rotational flow

$$u_{rot} = \omega_x (xy) - \omega_y (x^2 + 1) + \omega_z (y)$$

$$v_{rot} = \omega_x (y^2 + 1) - \omega_y (xy) - \omega_z (x)$$

3D structure and motion: 3D rotation

Error formulation

- We seek rotational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of rotational flow

$$u_{rot} = \omega_x (xy) - \omega_y (x^2 + 1) + \omega_z (y)$$

$$v_{rot} = \omega_x (y^2 + 1) - \omega_y (xy) - \omega_z (x)$$

- We accumulate this error over the entire image domain, I, so that we consider

$$\min_{\omega_x, \omega_y, \omega_z} \iint_I [(u - u_{rot})^2 + (v - v_{rot})^2] dxdy$$

3D structure and motion: 3D rotation

Error formulation

- We seek rotational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of rotational flow

$$u_{rot} = \omega_x (xy) - \omega_y (x^2 + 1) + \omega_z (y)$$

$$v_{rot} = \omega_x (y^2 + 1) - \omega_y (xy) - \omega_z (x)$$

- We accumulate this error over the entire image domain, I, so that we consider

$$\min_{\omega_x, \omega_y, \omega_z} \iint_I [(u - u_{rot})^2 + (v - v_{rot})^2] dxdy$$

- Differentiation WRT ω_x yields

$$\iint_I [-2(u - u_{rot})xy - 2(v - v_{rot})(y^2 + 1)] dxdy$$

3D structure and motion: 3D rotation

Error formulation

- We seek rotational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of rotational flow

$$u_{rot} = \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v_{rot} = \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

- We accumulate this error over the entire image domain, I, so that we consider

$$\min_{\omega_x,\omega_y,\omega_z} \iint_I [(u-u_{rot})^2 + (v-v_{rot})^2]dxdy$$

- Differentiation WRT ω_x and setting to zero yields

$$\iint_I [-2(u-u_{rot})xy - 2(v-v_{rot})(y^2+1)]dxdy = 0$$

3D structure and motion: 3D rotation

Error formulation

- We seek rotational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of rotational flow

$$u_{rot} = \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v_{rot} = \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

- We accumulate this error over the entire image domain, I, so that we consider

$$\min_{\omega_x,\omega_y,\omega_z} \iint_I [(u-u_{rot})^2 + (v-v_{rot})^2]dxdy$$

- Differentiation WRT ω_x and setting to zero yields

$$\iint_I [-\cancel{2}(u-u_{rot})xy - \cancel{2}(v-v_{rot})(y^2+1)]dxdy = 0$$

3D structure and motion: 3D rotation

Error formulation

- We seek rotational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of rotational flow

$$u_{rot} = \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v_{rot} = \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

- We accumulate this error over the entire image domain, I, so that we consider

$$\min_{\omega_x,\omega_y,\omega_z} \iint_I [(u-u_{rot})^2 + (v-v_{rot})^2]dxdy$$

- Differentiation WRT ω_x (and setting to zero) yields

$$\iint_I [(u-u_{rot})xy + (v-v_{rot})(y^2+1)]dxdy = 0$$

3D structure and motion: 3D rotation

Error formulation

- We seek rotational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of rotational flow

$$u_{rot} = \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v_{rot} = \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

- We accumulate this error over the entire image domain, I, so that we consider

$$\min_{\omega_x,\omega_y,\omega_z} \iint_I [(u-u_{rot})^2 + (v-v_{rot})^2]dxdy$$

- Differentiation WRT ω_y (and setting to zero) yields

$$\iint_I [(u-u_{rot})(x^2+1) + (v-v_{rot})xy]dxdy = 0$$

3D structure and motion: 3D rotation

Error formulation

- We seek rotational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of rotational flow

$$u_{rot} = \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v_{rot} = \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

- We accumulate this error over the entire image domain, I, so that we consider

$$\min_{\omega_x,\omega_y,\omega_z} \iint_I [(u-u_{rot})^2 + (v-v_{rot})^2]dxdy$$

- Differentiation WRT ω_z (and setting to zero) yields

$$\iint_I [(u-u_{rot})y - (v-v_{rot})x]dxdy = 0$$

3D structure and motion: 3D rotation

Equation counting

- We have 3 equations in 3 unknowns

$$\iint_I [(u - u_{rot})xy + (v - v_{rot})(y^2 + 1)]dxdy = 0$$

$$\iint_I [(u - u_{rot})(x^2 + 1) + (v - v_{rot})xy]dxdy = 0$$

$$\iint_I [(u - u_{rot})y - (v - v_{rot})x]dxdy = 0$$

3D structure and motion: 3D rotation

Equation counting

- We have 3 equations in 3 unknowns

$$\iint_I [(u-u_{rot})xy+(v-v_{rot})(y^2+1)]dxdy=0$$

$$\iint_I [(u-u_{rot})(x^2+1)+(v-v_{rot})xy]dxdy=0$$

$$\iint_I [(u-u_{rot})y-(v-v_{rot})x]dxdy=0$$

- So, we proceed to isolate the rotational parameters of interest

$$\iint_I [uxy+v(y^2+1)]dxdy=\iint_I [u_{rot}xy+v_{rot}(y^2+1)]dxdy$$

$$\iint_I [u(x^2+1)+vxy]dxdy=\iint_I [u_{rot}(x^2+1)+v_{rot}xy]dxdy$$

$$\iint_I [uy-vx]dxdy=\iint_I [u_{rot}y-v_{rot}x]dxdy$$

3D structure and motion: 3D rotation

Equation counting

- We have 3 equations in 3 unknowns

$$\iint_I [(u-u_{rot})xy+(v-v_{rot})(y^2+1)]dxdy=0$$

$$\iint_I [(u-u_{rot})(x^2+1)+(v-v_{rot})xy]dxdy=0$$

$$\iint_I [(u-u_{rot})y-(v-v_{rot})x]dxdy=0$$

- So, we proceed to isolate the rotational parameters of interest

$$\iint_I [uxy+v(y^2+1)]dxdy=\iint_I [u_{rot}xy+v_{rot}(y^2+1)]dxdy$$

Let's introduce notation for the LHS

$$\iint_I [u(x^2+1)+vxy]dxdy=\iint_I [u_{rot}(x^2+1)+v_{rot}xy]dxdy$$

$$\iint_I [uy-vx]dxdy=\iint_I [u_{rot}y-v_{rot}x]dxdy$$

3D structure and motion: 3D rotation

Equation counting

- We have 3 equations in 3 unknowns

$$\iint_I [(u-u_{rot})xy + (v-v_{rot})(y^2+1)]dxdy = 0$$

$$\iint_I [(u-u_{rot})(x^2+1) + (v-v_{rot})xy]dxdy = 0$$

$$\iint_I [(u-u_{rot})y - (v-v_{rot})x]dxdy = 0$$

- So, we proceed to isolate the rotational parameters of interest

$$a = \iint_I [u_{rot}xy + v_{rot}(y^2+1)]dxdy$$

$$b = \iint_I [u_{rot}(x^2+1) + v_{rot}xy]dxdy$$

$$c = \iint_I [u_{rot}y - v_{rot}x]dxdy$$

3D structure and motion: 3D rotation

Isolating the rotational parameters

- Let us expand the first equation

$$a = \iint_I [u_{rot} xy + v_{rot} (y^2 + 1)] dxdy$$

in terms of

$$u_{rot} = \omega_x (xy) - \omega_y (x^2 + 1) + \omega_z (y)$$

$$v_{rot} = \omega_x (y^2 + 1) - \omega_y (xy) - \omega_z (x)$$

3D structure and motion: 3D rotation

Isolating the rotational parameters

- Let us expand the first equation

$$a = \iint_I [u_{rot} xy + v_{rot}(y^2+1)]dxdy$$

in terms of

$$u_{rot} = \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v_{rot} = \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

- We find

$$a = \iint_I [\omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)]xy + [\omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)](y^2+1)]dxdy$$

3D structure and motion: 3D rotation

Isolating the rotational parameters

- Let us expand the first equation

$$a = \iint_I [u_{rot} xy + v_{rot}(y^2+1)]dxdy$$

in terms of

$$u_{rot} = \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v_{rot} = \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

- We find

$$a = \iint_I [\omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)]xy + [\omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)](y^2+1)]dxdy$$

- Grouping by the rotational parameters yields

$$a = \iint_I \omega_x[x^2y^2 + (y^2+1)^2] - \omega_y[xy(x^2+y^2+2)] - \omega_z[x]dxdy$$

3D structure and motion: 3D rotation

Isolating the rotational parameters

- Let us expand the first equation

$$a = \iint_I [u_{rot} xy + v_{rot}(y^2+1)]dxdy$$

in terms of

$$u_{rot} = \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v_{rot} = \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

- We find

$$a = \iint_I [\omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)]xy + [\omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)](y^2+1)]dxdy$$

- Grouping by the rotational parameters yields

$$a = \iint_I \omega_x[x^2y^2 + (y^2+1)^2] - \omega_y[xy(x^2+y^2+2)] - \omega_z[x]dxdy$$

- Since the rotation is constant over the image domain, we pull $\omega_x, \omega_y, \omega_z$ outside the integrals

$$a = \omega_x \iint x^2y^2 + (y^2+1)^2 dxdy - \omega_y \iint xy(x^2+y^2+2)dxdy - \omega_z \iint xdxdy$$

3D structure and motion: 3D rotation

3 equations in 3 unknowns ready for solution

- For compactness of notation let us write

$$a = \omega_x \iint x^2 y^2 + (y^2 + 1)^2 \, dxdy - \omega_y \iint xy(x^2 + y^2 + 2) \, dxdy - \omega_z \iint x \, dxdy$$

3D structure and motion: 3D rotation

3 equations in 3 unknowns ready for solution

- For compactness of notation let us write

$$a = \omega_x \boxed{\iint x^2 y^2 + (y^2 + 1)^2 dxdy} - \omega_y \boxed{\iint xy(x^2 + y^2 + 2) dxdy} - \omega_z \boxed{\iint xdxdy}$$

as

$$a = \omega_x (j) \quad + \omega_y (k) \quad + \omega_z (l)$$

3D structure and motion: 3D rotation

3 equations in 3 unknowns ready for solution

- For compactness of notation let us write

$$a = \omega_x \boxed{\iint x^2 y^2 + (y^2+1)^2 dxdy} - \omega_y \boxed{\iint xy(x^2+y^2+2) dxdy} - \omega_z \boxed{\iint xdxdy}$$

as

$$a = \omega_x (j) \quad + \omega_y (k) \quad \omega_z (l)$$

- Similarly, we can derive expressions for the other two constraint equations along the lines of

$$b = \omega_x (m) + \omega_y (n) + \omega_z (o)$$

$$c = \omega_x (p) + \omega_y (q) + \omega_z (r)$$

3D structure and motion: 3D rotation

Matrix formulation

- From

$$a = \omega_x(j) + \omega_y(k) + \omega_z(l)$$

$$b = \omega_x(m) + \omega_y(n) + \omega_z(o)$$

$$c = \omega_x(p) + \omega_y(q) + \omega_z(r)$$

3D structure and motion: 3D rotation

Matrix formulation

- From

$$a = \omega_x(j) + \omega_y(k) + \omega_z(l)$$

$$b = \omega_x(m) + \omega_y(n) + \omega_z(o)$$

$$c = \omega_x(p) + \omega_y(q) + \omega_z(r)$$

- We derive the matrix form

$$\begin{pmatrix} j & k & l \\ m & n & o \\ p & q & r \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

3D structure and motion: 3D rotation

Solution

- From the matrix form

$$\begin{pmatrix} j & k & l \\ m & n & o \\ p & q & r \end{pmatrix}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

3D structure and motion: 3D rotation

Solution

- From the matrix form

$$\begin{pmatrix} j & k & l \\ m & n & o \\ p & q & r \end{pmatrix}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

- We have solution via the inverse operation

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} j & k & l \\ m & n & o \\ p & q & r \end{pmatrix}^{-1}\begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

3D structure and motion: 3D rotation

By the way

- Straightforward calculation yields the coefficient expansions we skimmed over

$$m = -k$$

$$n = -\iint (x^2 + 1)^2 + x^2 y^2 dxdy$$

$$o = \iint y dxdy$$

$$p = l$$

$$q = -o$$

$$r = \iint x^2 + y^2 dxdy$$

3D structure and motion: 3D recovery from 2D motion

Where we stand

- We have developed methods to recover image motion (optical flow) from a time varying image sequence.
- We have derived an explicit relationship to relate the image motion to the corresponding 3D structure and motion.

Where we will go

- We seek an approach to the recovery of 3D structure and motion from the estimated image motion.
- In general, this is a very difficult problem.
  - Although approach have been developed
  - With varying success
- We will consider 2 special cases.

1. Recovery of 3D structure, Z(x,y), given motion parameters
2. Recovery of 3D rotation, assuming no translation.

Remark

- In a similar spirit we could attack
  - The pure translation case
  - The general case (unknown structure, rotation and translation)

3D structure and motion: 3D recovery from 2D motion

Where we stand

- We have developed methods to recover image motion (optical flow) from a time varying image sequence.
- We have derived an explicit relationship to relate the image motion to the corresponding 3D structure and motion.

Where we will go

- We seek an approach to the recovery of 3D structure and motion from the estimated image motion.
- In general, this is a very difficult problem.
  - Although approach have been developed
  - With varying success
- We will consider 2 special cases.

1. Recovery of 3D structure, Z(x,y), given motion parameters
2. Recovery of 3D rotation, assuming no translation.

Remark

- In a similar spirit we could attack
  - The pure translation case
  - The general case (unknown structure, rotation and translation)

3D structure and motion: 3D translation

We assume purely 3D translation

- We specialize the general case

$$u = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

to

$$u = \frac{1}{Z}(xt_z - t_x)$$

$$v = \frac{1}{Z}(yt_z - t_y)$$

Remarks

- We recall that flow due to purely 3D translation is intertwined with 3D scene structure. is independent of 3D scene structure.
- Z varies (potentially) with each image location.
- Translation parameters are constant across the image.
- Notice the scale ambiguity between scene structure, Z, and translation, (t_x, t_y, t_z)

3D structure and motion: 3D translation

Error formulation

- We seek scene structure and translation parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of translational flow

$$u_{trans} = \frac{1}{Z}(xt_z - t_x)$$

$$v_{trans} = \frac{1}{Z}(yt_z - t_y)$$

3D structure and motion: 3D translation

Error formulation

- We seek scene structure and translational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of translational flow

$$u_{trans} = \frac{1}{Z}(xt_z - t_x)$$

$$v_{trans} = \frac{1}{Z}(yt_z - t_y)$$

- We accumulate this error over the entire image domain, I, so that we consider

$$\min_{t_x, t_y, t_z, Z} \iint_I [(u - u_{trans})^2 + (v - v_{trans})^2]\, dxdy$$

3D structure and motion: 3D translation

Error formulation

- We seek scene structure and translational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of translational flow

$$u_{trans} = \frac{1}{Z}(xt_z - t_x)$$

$$v_{trans} = \frac{1}{Z}(yt_z - t_y)$$

- We accumulate this error over the entire image domain, I, so that we consider

$$\min_{t_x, t_y, t_z, Z} \iint_I [(u - u_{trans})^2 + (v - v_{trans})^2] dxdy$$

- A solution can be had by pursuing a path analogous to that followed for recovery of rotational parameters.
- A new wrinkle is in the apparent need to decouple the translation and scene structure; various approaches have been developed, including
  - Iterating between recovering translation and structure, while holding the other constant.
  - Initially eliminating Z and solving for (t_x, t_y, t_z) and subsequently recovering Z.

3D structure and motion: 3D recovery from 2D motion

Where we stand

- We have developed methods to recover image motion (optical flow) from a time varying image sequence.
- We have derived an explicit relationship to relate the image motion to the corresponding 3D structure and motion.

Where we will go

- We seek an approach to the recovery of 3D structure and motion from the estimated image motion.
- In general, this is a very difficult problem.
  - Although approach have been developed
  - With varying success
- We will consider 2 special cases.

1. Recovery of 3D structure, Z(x,y), given motion parameters
2. Recovery of 3D rotation, assuming no translation.

Remark

- In a similar spirit we could attack
  - The pure translation case
  - The general case (unknown structure, rotation and translation)

3D structure and motion: General case

We assume no more than instantaneous motion

- We have our general case

$$u = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

and make no specializations.

3D structure and motion: General case

Error formulation

- We seek scene structure, translational and rotational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of the general visual motion field

$$u_{gen} = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v_{gen} = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

3D structure and motion: General case

Error formulation

- We seek scene structure, translational and rotational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of the general visual motion field

$$u_{gen} = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v_{gen} = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

- We accumulate this error over the entire image domain, I, so that we consider

$$\min_{t_x, t_y, t_z, \omega_x, \omega_y, \omega_z, Z} \iint_I [(u - u_{gen})^2 + (v - v_{rot})^2] dxdy$$

3D structure and motion: General case

Error formulation

- We seek scene structure, translational and rotational parameters that minimize the squared error between
  - The observed flow (u,v)
  - And the supposed parametric model of the general visual motion field

$$u_{gen} = \frac{1}{Z}(xt_z - t_x) + \omega_x(xy) - \omega_y(x^2+1) + \omega_z(y)$$

$$v_{gen} = \frac{1}{Z}(yt_z - t_y) + \omega_x(y^2+1) - \omega_y(xy) - \omega_z(x)$$

- We accumulate this error over the entire image domain, I, so that we consider

$$\min_{t_x, t_y, t_z, \omega_x, \omega_y, \omega_z, Z} \iint_I [(u - u_{gen})^2 + (v - v_{gen})^2] dxdy$$

- Again, analogous methods to those used so far can be brought to bear to yield a solution.

3D structure and motion: 3D recovery from 2D motion

Where we stand

- We have developed methods to recover image motion (optical flow) from a time varying image sequence.
- We have derived an explicit relationship to relate the image motion to the corresponding 3D structure and motion.

Where we will go

- We seek an approach to the recovery of 3D structure and motion from the estimated image motion.
- In general, this is a very difficult problem.
  - Although approach have been developed
  - With varying success
- We will consider 2 special cases.

1. Recovery of 3D structure, Z(x,y), given motion parameters
2. Recovery of 3D rotation, assuming no translation.

Remark

- In a similar spirit we could attack
  - The pure translation case
  - The general case (unknown structure, rotation and translation)

3D structure and motion: 3D recovery from 2D motion

Where we stand

- We have developed methods to recover image motion (optical flow) from a time varying image sequence.
- We have derived an explicit relationship to relate the image motion to the corresponding 3D structure and motion.

Where we will go

- We seek an approach to the recovery of 3D structure and motion from the estimated image motion.
- In general, this is a very difficult problem.
  - Although approach have been developed
  - With varying success
- We will consider 2 special cases.

1. Recovery of 3D structure, Z(x,y), given motion parameters
2. Recovery of 3D rotation, assuming no translation.

Remark

- In a similar spirit we could attack
  - The pure translation case
  - The general case (unknown structure, rotation and translation)

But, instead of developing the details we look at some empirical results derived from a general solution.

3D structure and motion: 3D recovery from 2D motion

Empirical examples presented in lecture.

3D structure and motion: Final remarks

Flow-based

- Works with minimal number of frames (n=2).
- Assumes infinitesimal 3D motion and full perspective projection.
- Provides dense 3D distance estimates.
- Can be numerically sensitive

Factorization-based

- Works with larger temporal streams of frames (n>2).
- Assumes arbitrary 3D motion and orthographic projection (but extensions to more general projection models have been developed).
- Provides 3D shape estimates only at tracked feature points.
- Reasonably numerically stable; at least for orthographic cases...

Summary

- Motion field vs. optical flow
- Brightness constancy
- Gradient-based optical flow estimation
- Finite displacement and feature-based methods
- 3D Structure and motion