Structure from motion II

Digital Visual Effects, Spring 2005

Yung-Yu Chuang

2005/5/4

with slides by Richard Szeliski, Steve Seitz, Marc Pollefyes and Daniel Martinec

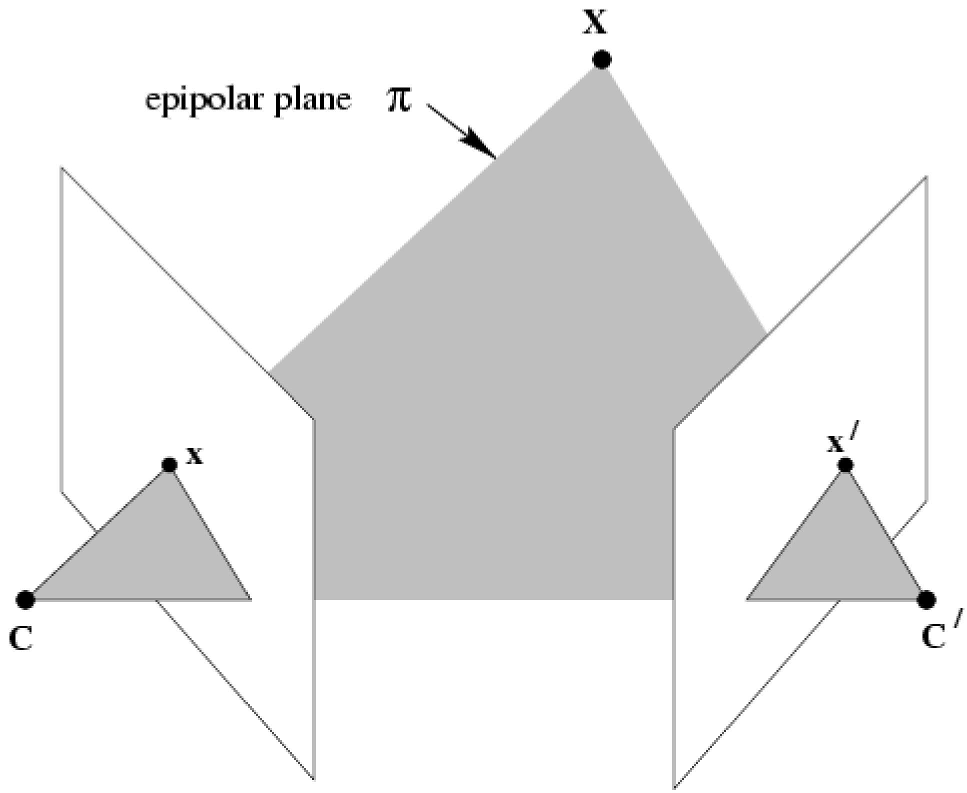
Announcements

- Project #2 artifacts voting.
- Project #3 will be online tomorrow, hopefully.
- Scribe schedule.

Outline

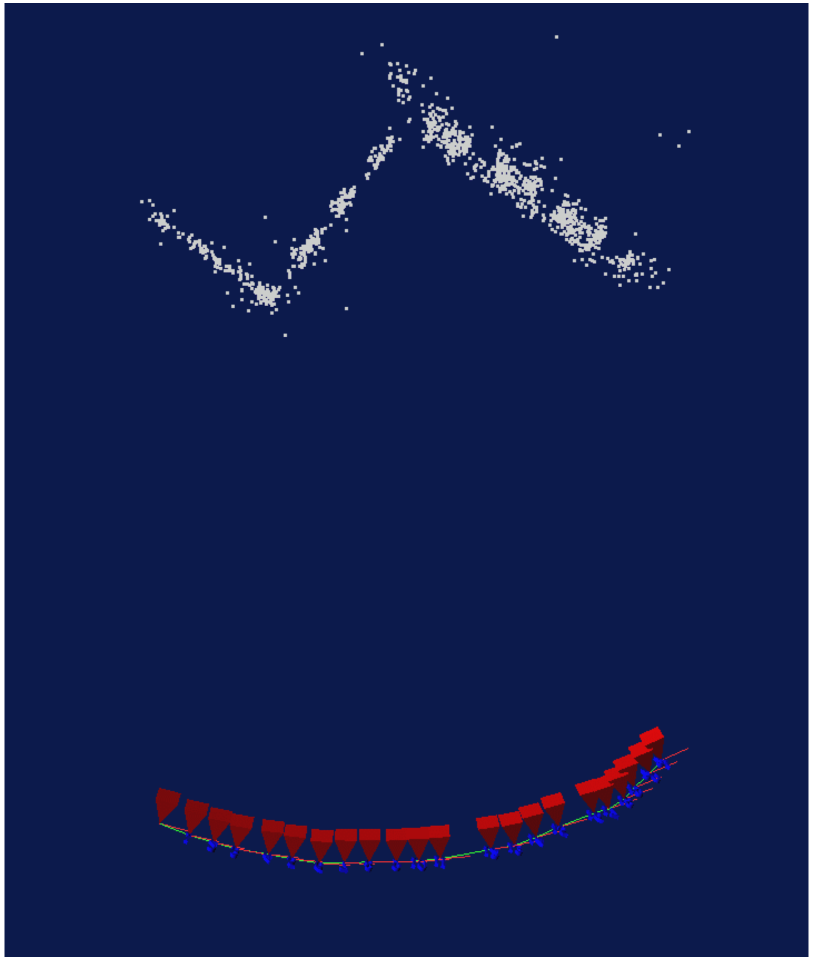
- Factorization methods
  - Orthogonal
  - Missing data
  - Projective
  - Projective with missing data
- Project #3

Recap: epipolar geometry



$$\mathbf{x}'^{\mathrm{T}}\mathbf{F}\mathbf{x} = 0$$

Structure from motion



Structure from motion

2D feature tracking → 3D estimation → optimization (bundle adjust) → geometry fitting

SFM pipeline

Factorization methods

Notations

- *n* 3D points are seen in *m* views
- **q**=*(u,v,1): 2D image point*
- **p**=*(x,y,z,1): 3D scene point*
- Π: projection matrix
- π: projection function
- q_{ij} is the projection of the *i*-th point on image *j*
- λ_{ij} projective depth of q_{ij}

$$\mathbf{q}_{ij} = \pi(\Pi_j \mathbf{p}_i)$$

$$\pi(x, y, z) = (x/z, y/z)$$

$$\lambda_{ij} = z$$

SFM under orthographic projection

$$\underbrace{\overset{\text{2D image point}}{\mathbf{q}}}_{2\times 1} = \underbrace{\overset{\text{orthographic projection matrix}}{\mathbf{\Pi}}}_{2\times 3}\underbrace{\overset{\text{3D scene point}}{\mathbf{p}}}_{3\times 1} + \underbrace{\overset{\text{image offset}}{\mathbf{t}}}_{2\times 1}$$

- Trick
  - Choose scene origin to be centroid of 3D points
  - Choose image origins to be centroid of 2D points
  - Allows us to drop the camera translation:

$$\mathbf{q} = \mathbf{\Pi}\mathbf{p}$$

factorization (Tomasi & Kanade)

projection of *n* features in one image:

$$\underset{2\times n}{\begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n\end{bmatrix}} = \underset{2\times 3}{\prod} \underset{3\times n}{\begin{bmatrix}\mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n\end{bmatrix}}$$

projection of *n* features in *m* images

$$\underset{2\mathrm{m}\times \mathrm{n}}{\begin{bmatrix}\mathbf{q}_{11} & \mathbf{q}_{12} & \cdots & \mathbf{q}_{1n}\\ \mathbf{q}_{21} & \mathbf{q}_{22} & \cdots & \mathbf{q}_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{q}_{m1} & \mathbf{q}_{m2} & \cdots & \mathbf{q}_{mn}\end{bmatrix}} = \underset{2\mathrm{m}\times 3}{\begin{bmatrix}\mathbf{\Pi}_1\\ \mathbf{\Pi}_2\\ \vdots\\ \mathbf{\Pi}_m\end{bmatrix}} \underset{3\times \mathrm{n}}{\begin{bmatrix}\mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n\end{bmatrix}}$$

W measurement **M** motion **S** shape

Key Observation: *rank*(**W**) <= 3

Factorization

$$\text{known} - \mathbf{W}_{2m\times n} = \boxed{\mathbf{M}_{2m\times 3}\ \mathbf{S}_{3\times n}} - \text{solve for}$$

- Factorization Technique
  - **W** is at most rank 3 (assuming no noise)
  - We can use *singular value decomposition* to factor **W**:

$$\underset{2m\times n}{\mathbf{W}} = \underset{2m\times 3}{\mathbf{M}'}\ \underset{3\times n}{\mathbf{S}'}$$

  - **S'** differs from **S** by a linear transformation ***A***:

$$\mathbf{W} = \mathbf{M}'\mathbf{S}' = (\mathbf{M}\mathbf{A}^{-1})(\mathbf{A}\mathbf{S})$$

  - Solve for **A** by enforcing *metric* constraints on **M**

Metric constraints

- Orthographic Camera
  - Rows of Π are orthonormal: $\Pi\,\Pi^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- Enforcing "Metric" Constraints
  - Compute **A** such that rows of **M** have these properties

$$\mathbf{M}'\mathbf{A} = \mathbf{M}$$

Trick (not in original Tomasi/Kanade paper, but in followup work)

- Constraints are linear in **AA**^{T} :

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \Pi\,\Pi^T = \Pi'\mathbf{A}(\mathbf{A}\,\Pi')^T = \Pi'\mathbf{G}\,\Pi'^T \qquad where \quad \mathbf{G} = \mathbf{A}\mathbf{A}^T$$

- Solve for **G** first by writing equations for every Π_i in **M**
- Then **G** = **AA**^{T} by SVD (since **U** = **V**)

Factorization with noisy data

$$\underset{2m\times n}{\mathbf{W}} = \underset{2m\times 3}{\mathbf{M}}\ \underset{3\times n}{\mathbf{S}} + \underset{2m\times n}{\mathbf{E}}$$

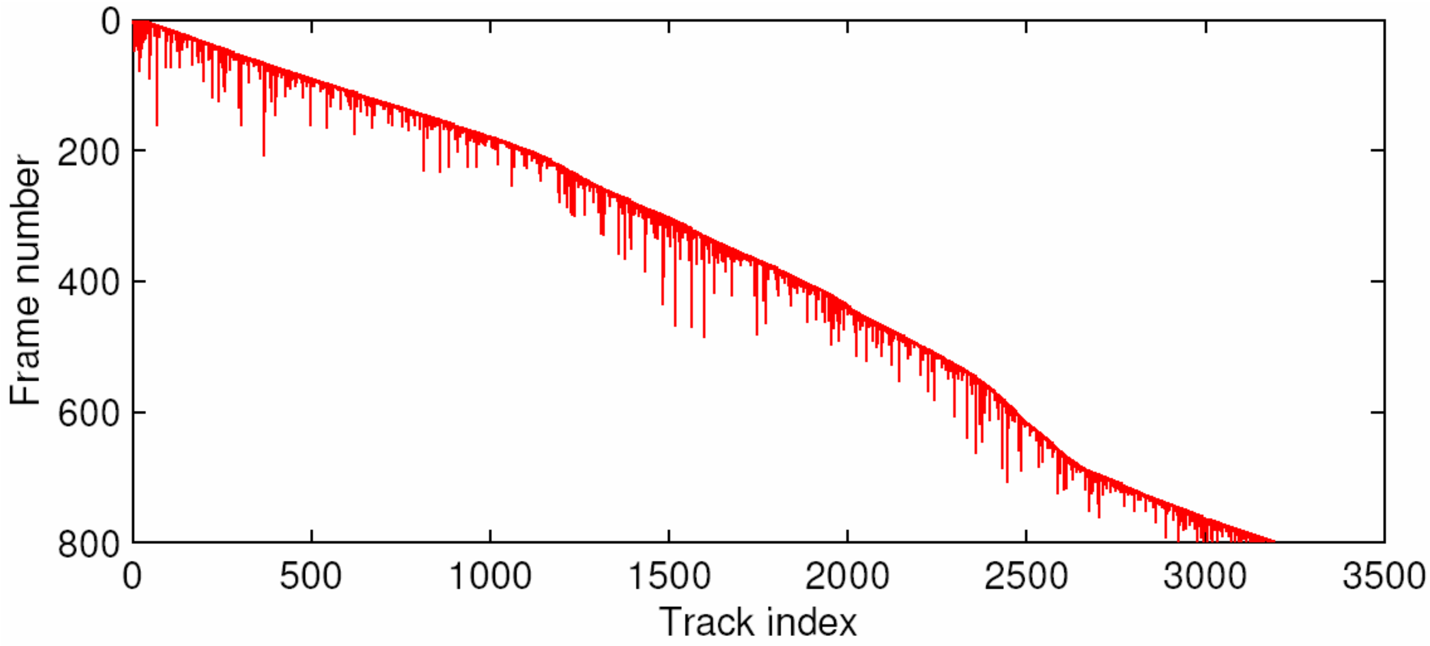
- SVD gives this solution
  - Provides optimal rank 3 approximation **W'** of **W**

$$\underset{2m\times n}{\mathbf{W}} = \underset{2m\times n}{\mathbf{W}'} + \underset{2m\times n}{\mathbf{E}}$$

- Approach
  - Estimate **W'**, then use noise-free factorization of **W'** as before
  - Result minimizes the SSD between positions of image features and projection of the reconstruction

Factorization method
with missing data

Why missing data?



- occlusions
- tracking failure

→W is only partially filled, factorization doesn't work

Tomasi & Kanade

- Hallucination/propagation

$$W = \left[\frac{U}{V}\right] = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ u_{21} & u_{22} & u_{23} & u_{24} \\ u_{31} & u_{32} & u_{33} & u_{34} \\ u_{41} & u_{42} & u_{43} & ? \\ v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & ? \end{bmatrix}$$

4 points in 3 views determine structure and motion

$$W_{6\times 4} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ u_{21} & u_{22} & u_{23} & u_{24} \\ u_{31} & u_{32} & u_{33} & u_{34} \\ v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \end{bmatrix}$$

Tomasi & Kanade

$$W_{6\times 4} = R_{6\times 3}S + \mathbf{t}_{6\times 1}\mathbf{e}_4^T$$

$$S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 & \mathbf{s}_4 \end{bmatrix}$$

$$\mathbf{t}_{6\times 1} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad \text{and} \quad R_{6\times 3} = \begin{bmatrix} \mathbf{i}_1^T \\ \mathbf{i}_2^T \\ \mathbf{i}_3^T \\ \mathbf{j}_1^T \\ \mathbf{j}_2^T \\ \mathbf{j}_3^T \end{bmatrix}$$

Tomasi & Kanade

- Solve for $\mathbf{i}_4$ and $\mathbf{j}_4$:

$$\begin{bmatrix} u'_{41} & u'_{42} & u'_{43} \end{bmatrix} = \mathbf{i}_4^T \begin{bmatrix} \mathbf{s}'_1 & \mathbf{s}'_2 & \mathbf{s}'_3 \end{bmatrix}$$
$$\begin{bmatrix} v'_{41} & v'_{42} & v'_{43} \end{bmatrix} = \mathbf{j}_4^T \begin{bmatrix} \mathbf{s}'_1 & \mathbf{s}'_2 & \mathbf{s}'_3 \end{bmatrix}$$

$$u'_{4p} = u_{4p} - a'_4$$
$$v'_{4p} = v_{4p} - b'_4$$

$$\mathbf{s}'_p = \mathbf{s}_p - \mathbf{c}$$

$$a'_4 = \frac{1}{3}(u_{41} + u_{42} + u_{43})$$
$$b'_4 = \frac{1}{3}(v_{41} + v_{42} + v_{43})$$

$$\mathbf{c} = \frac{1}{3}(\mathbf{s}_1 + \mathbf{s}_2 + \mathbf{s}_3)$$

Tomasi & Kanade

- Alternatively, first apply factorization on

$$W_{8\times 3} = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \\ u_{41} & u_{42} & u_{43} \\ v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \\ v_{41} & v_{42} & v_{43} \end{bmatrix}$$

Tomasi & Kanade

- Disadvantages
  - Finding the largest full submatrix of a matrix with missing elements is NP-hard.
  - The data is not used symmetrically, these inaccuracies will propagate in the computation of additional missing elements.

Shum, Ikeuchi & Reddy

- Treat SVD as a PCA with missing data problem which is a weighted least square problem.
- Assume that W consists of n m-d points with mean t and covariance Σ. If the rank of W is r, the problem of PCA is to find U,S,V such that $\left\| \mathbf{W} - \mathbf{e}\mathbf{t}^{\mathrm{T}} - \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}} \right\|$ is minimal.
- If W is incomplete, it becomes

$$\min \phi = \frac{1}{2} \sum_{q_{ij} \text{ is visible}} \left(\mathbf{q}_{ij} - \mathbf{t}_j - \mathbf{u}_i^{\mathrm{T}} \mathbf{v}_j \right)^2$$

Shum, Ikeuchi & Reddy

- To be solvable, the number of observable elements *c* in W must be larger than r(m+n-r)
- If we arrange W as an c-d vector w, we can rewrite it as

$$\min \phi = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{f}$$

$$\mathbf{f} = \mathbf{w} - \mathbf{t} - \mathbf{B}\mathbf{u} = \mathbf{w} - \mathbf{G}\mathbf{v}$$

- To reach minimum, u and v satisfies:

$$\begin{bmatrix} \mathbf{B}^{\mathrm{T}}\mathbf{B}\mathbf{u} - \mathbf{B}^{\mathrm{T}}(\mathbf{w} - \mathbf{t}) \\ \mathbf{G}^{\mathrm{T}}\mathbf{G}\mathbf{v} - \mathbf{G}^{\mathrm{T}}\mathbf{w} \end{bmatrix} = 0$$

Shum, Ikeuchi & Reddy

- Nonlinear, solved by iterating between "fixing v and solving u" and "fixing u and solving v"

1) initialize v
2) update $\mathbf{u} = \mathbf{B}^{+}(\mathbf{w} - \mathbf{t})$
3) update $\mathbf{v} = \mathbf{G}^{+}\mathbf{w}$
4) stop if convergence, or go back to step 2

- The above procedure can be further simplified by taking advantage of the sparse structure.

Shum, Ikeuchi & Reddy

- Disadvantages: sensitive to the starting point

Linear fitting

- Try to find a rank-r matrix $\hat{\mathbf{W}}$ so that $\left\|\hat{\mathbf{W}} - \mathbf{W}\right\|$ is minimal.
- Each column of W is an m-d vector. SVD tries to find an r-d linear space L that is closest to these n m-d vectors and projects these vectors to L.
- A matrix describes a vector space.

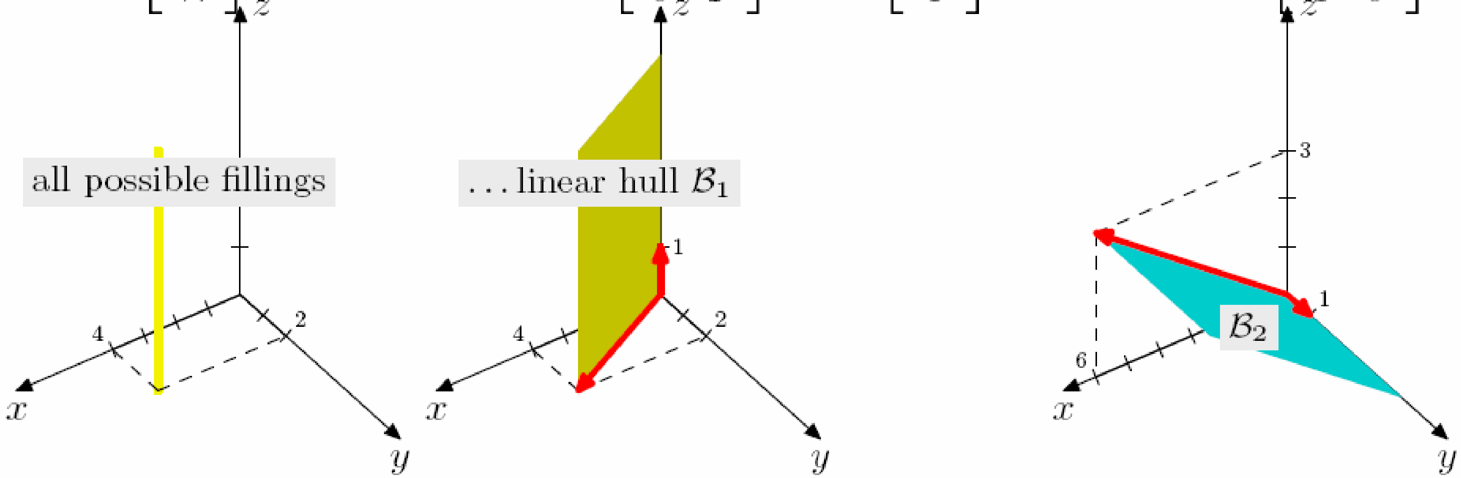
Linear fitting

- Without noise, each triplet of columns of M exactly specifies L. When there is missing data, each triplet only forms a constraint.
- For SFM, r=3, We can combine constraints to find L

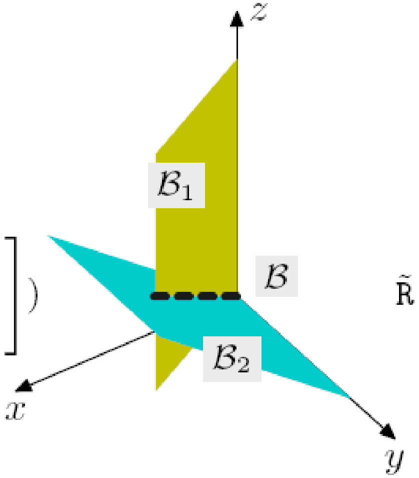
$$\mathrm{L} \subseteq \bigcap_{(i,j,k)} span(A_i, A_j, A_k)$$

Example: $\quad \mathtt{R} = \begin{bmatrix} 4 & 6 \\ 2 & \times \\ \times & 3 \end{bmatrix}$, for rank $\mathtt{R} = 1$ instead of rank $\mathtt{R} = 4$

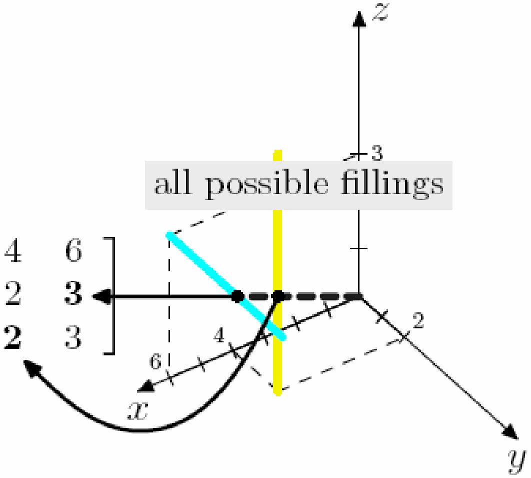
$\begin{bmatrix} 4 \\ 2 \\ \times \end{bmatrix} \ \dots \ \mathcal{B}_1 = Span(\begin{bmatrix} 4 & \mathit{0} \\ 2 & \mathit{0} \\ \mathit{0} & \mathit{1} \end{bmatrix}),$ $\qquad \begin{bmatrix} 6 \\ \times \\ 3 \end{bmatrix} \ \dots \ \mathcal{B}_2 = Span(\begin{bmatrix} 6 & \mathit{0} \\ \mathit{0} & \mathit{1} \\ \mathit{3} & \mathit{0} \end{bmatrix})$



$\mathcal{B} \subseteq \mathcal{B}_1 \cap \mathcal{B}_2 = Span(\begin{bmatrix} \mathbf{2} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix})$



$\tilde{\mathtt{R}} = \begin{bmatrix} 4 & 6 \\ 2 & \mathbf{3} \\ \mathbf{2} & 3 \end{bmatrix}$



Linear fitting

$$\mathrm{L} = \bigcap_t S_t \quad \Rightarrow \quad \overline{\mathrm{L}} = \bigcup_t \overline{S}_t$$

Let N_t denote a matrix representation of $\overline{S}_t$, that is, each column of N_t is a vector orthogonal to the space S_t.

If $N=[N_1, N_2, \ldots N_l]$, then L is the null space of N.

Because of noise, the matrix N will typically have full rank. Taking the SVD of N, and find its three least significant components. If fourth smallest singular value of this matrix is less than 0.001, the result is unreliable.

This method can be used as the initialization for Shum's method.

Factorization method with projective projection

Factorization for projective projection

projective depth

$$\lambda_{ij}\mathbf{q}_{ij} = \Pi_j \mathbf{p}_i$$

$$\underset{3\mathrm{m}\times\mathrm{n}}{\begin{bmatrix} \lambda_{11}\mathbf{q}_{11} & \lambda_{11}\mathbf{q}_{12} & \cdots & \lambda_{11}\mathbf{q}_{1n} \\ \lambda_{21}\mathbf{q}_{21} & \lambda_{22}\mathbf{q}_{22} & \cdots & \lambda_{2n}\mathbf{q}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{q}_{m1} & \lambda_{m2}\mathbf{q}_{m2} & \cdots & \lambda_{mn}\mathbf{q}_{mn} \end{bmatrix}} = \underset{3\mathrm{m}\times 4}{\begin{bmatrix} \mathbf{\Pi}_1 \\ \mathbf{\Pi}_2 \\ \vdots \\ \mathbf{\Pi}_m \end{bmatrix}} \underset{4\times\mathrm{n}}{\begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \end{bmatrix}}$$

W has rank at most 4. The problem is that we don't know λ.

Sturm & Triggs

For the p-th point, its projective depths for the i-th and j-th images are related by

$$\lambda_{ip} = \frac{(\boldsymbol{e}_{ij} \wedge \boldsymbol{q}_{ip}) \cdot (\boldsymbol{F}_{ij}\boldsymbol{q}_{jp})}{\|\boldsymbol{e}_{ij} \wedge \boldsymbol{q}_{ip}\|^2} \lambda_{jp}$$

1. Normalize the image coordinates, by applying transformations $\boldsymbol{T}_i$.
2. Estimate the fundamental matrices and epipoles with the method of [Har95].
3. Determine the scale factors λ_{ip} using equation (3).
4. Build the rescaled measurement matrix $\boldsymbol{W}$.
5. Balance $\boldsymbol{W}$ by column-wise and "triplet-of-rows"-wise scalar mutliplications.
6. Compute the SVD of the balanced matrix $\boldsymbol{W}$.
7. From the SVD, recover projective motion and shape.
8. Adapt projective motion, to account for the normalization transformations $\boldsymbol{T}_i$ of step 1.

Sturm & Triggs

DigiVFX

Compute an initial estimate of the projective depths z_{ij}, with $i = 1, .., m$ and $j = 1, .., n$.
Repeat:
(1) normalize each row of the data matrix $\mathcal{I}$, then normalize each one of its columns;
(2) use singular value decomposition to compute the matrices $\mathcal{M}$ and $\mathcal{P}$ minimizing $|\mathcal{I} - \mathcal{M}\mathcal{P}|^2$;
(3) for $i = 1, .., m$ and $j = 1, .., n$, find the value of z_{ij} minimizing $|z_{ij}\boldsymbol{p}_{ij} - \mathcal{M}_i \boldsymbol{P}_j|^2$ using linear least squares;
until convergence.

Factorization method with projective projection and missing data

Mahamud *et. al.*

$$E_{ij}=\sum_{ij}|z_{ij}q_{ij}-M_iP_j|^2=\sum_{ij}|q_{ij}\times(M_iP_j)|^2$$

$$E_i^{(P)}=\sum_{j=1}^{n}\left|q_{ij}\times(M_iP_j)\right|^2=\left|C_im_i\right|^2$$

$$\begin{pmatrix}
-w_{i1}\boldsymbol{P}_1^T & \mathbf{0}^T & u_{i1}\boldsymbol{P}_1^T \\
\mathbf{0}^T & -w_{i1}\boldsymbol{P}_1^T & v_{i1}\boldsymbol{P}_1^T \\
-v_{i1}\boldsymbol{P}_1^T & u_{i1}\boldsymbol{P}_1^T & \mathbf{0}^T \\
\cdots & \cdots & \cdots \\
-w_{in}\boldsymbol{P}_n^T & \mathbf{0}^T & u_{in}\boldsymbol{P}_n^T \\
\mathbf{0}^T & -w_{in}\boldsymbol{P}_n^T & v_{in}\boldsymbol{P}_n^T \\
-v_{in}\boldsymbol{P}_n^T & u_{in}\boldsymbol{P}_n^T & \mathbf{0}^T
\end{pmatrix}$$

Mahamud *et. al*.

Compute an initial estimate of the vectors $\boldsymbol{P}_1 \ldots, \boldsymbol{P}_n$ and normalize these vectors.

Repeat:

(1) for $i = 1$ to m, compute the unit vector $\boldsymbol{m}_i$ that minimizes $|\mathcal{C}_i \boldsymbol{m}_i|^2$;

(2) for $j = 1$ to n, compute the unit vector $\boldsymbol{P}_j$ that minimizes $|\mathcal{D}_j \boldsymbol{P}_j|^2$;

until convergence.

Project #3 Matchmove

- Assigned: 5/4
- Due: 11:59pm 5/24
- Work in pairs
- Implement Tomasi/Kanade factorization method.
- Some matlab implementations are provided as reference for implementation details.

Bells & whistles

- Tracking
- Extensions of factorization methods (Jacobs, Mahamud are recommended)
- Bundle adjustment
- Better graphics composition

Artifacts

- Take your own movie and insert some objects into it.
- Sony TRV900, progressive mode, 15fps
- Capturing machine in 219
- Demo of how to capturing video

Submission

- You have to turn in your complete source, the executable, a html report and an artifact.
- Report page contains:

  description of the project, what do you learn, algorithm, implementation details, results, bells and whistles…
- Artifacts must be made using your own program.

  artifacts voting on forum.

Reference software

- Famous matchmove software include 3D-Equalizer, boujou, REALVIS MatchMover, PixelFarm PFTrack... Most are very expensive
- We will use Icarus, predecessor of PFTrack. It will be available at project's page (id/password).

ICARUS

- Three main components:
  - Distortion
  - Calibration
  - Reconstruction
- Capturing video
  - Enough depth variance
  - Fixed zoom if possible
  - Static scene if possible

Reference

- Heung-Yeung Shum, Katsushi Ikeuchi and Raj Reddy, Principal Component Analysis with Missing Data and Its Application to Polyhedral Object Modeling, PAMI 17(9), 1995.
- David Jacobs, Linear Fitting with Missing Data for Structure from Motion, Computer Vision and Image Understanding, 2001
- Peter Sturm and Bill Triggs, A factorization Based Algorithm for Multi-Image Projective Structure and Motion, ECCV 1996.
- Shyjan Mahamud, Martial Hebert, Yasuhiro Omori and Jean Ponce, Provably-Convergent Iterative Methods for Projective Structure from Motion, ICCV 2001.