Lecture 13: CS677

October 3, 2017

SFM: Motivating Examples

- Pix4d: focus on images acquired by low flying drones

https://pix4d.com/
https://pix4d.com/mapping-christ/
https://pix4d.com/modelling-matterhorn/

- Competing technology, LIDAR
  - We will study right after SFM study
  - Then, we can compare and contrast the two

Fundamental Matrix

- Essential matrix equation applies when cameras are calibrated
- Fortunately, a similar condition holds even without knowledge of the intrinsic parameters
- Let $p = K\hat{p}$ and $p' = K'\hat{p}'$; *p* and *p*' are the image coordinates; $\hat{p}$ and $\hat{p}'$ are the normalized coordinates, *K* and *K'* are the intrinsic matrices
- Substitute in essential matrix equation $\hat{p}^{T}\boldsymbol{\varepsilon}\,\hat{p}' = 0$,we get: $\boldsymbol{p}^{T}\boldsymbol{F}\boldsymbol{p}' = 0$;

  where $F = K^{-T}\,\boldsymbol{\varepsilon}\,K'^{-1}$; ***F*** is called the *fundamental* matrix.

$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

- ***F*** is also of rank 2 (since $\boldsymbol{\varepsilon}$ is of rank 2) but the two eigenvalues are now not necessarily equal. Only 7 independent parameters even though it has 9 elements

Eigen-Decomposition

- Given an n x n square matrix, say A, it can be decomposed as:

  $\mathbf{A} = \mathbf{U}\,\mathbf{W}\,\mathbf{U}^{-1}$,

  where **W** is a diagonal matrix of eigenvalues along the diagonal; columns of **U** are the eigenvectors (in order of eigenvalues in **W**)
- Can be used to invert **A**: $\mathbf{A}^{-1} = \mathbf{U}\,\mathbf{W}^{-1}\,\mathbf{U}^{-1}$
- Can also be used to solve equations such as $A\boldsymbol{x} = \boldsymbol{b}$
- When **A** is not square (consider over determined set of linear equations), we can use singular valued decomposition (and pseudo-inverse of **A**).

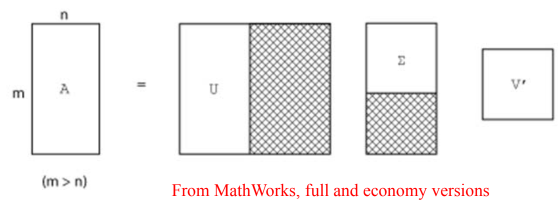
Singular Value Decomposition (SVD)

$\mathcal{A}$, with $p \geq q$, can be written as

$$\mathcal{A} = \mathcal{U}\mathcal{W}\mathcal{V}^T,$$

where

- $\mathcal{U}$ is a $p \times q$ column-orthogonal matrix—that is, $\mathcal{U}^T\mathcal{U} = \mathrm{Id}_p$;
- $\mathcal{W}$ is a diagonal matrix whose diagonal entries w_i $(i = 1, \ldots, q)$ are the singular values of $\mathcal{A}$ with $w_1 \geq w_2 \geq \ldots \geq w_q \geq 0$;
- and $\mathcal{V}$ is a $q \times q$ orthogonal matrix—that is, $\mathcal{V}^T\mathcal{V} = \mathcal{V}\mathcal{V}^T = \mathrm{Id}_q$.



From MathWorks, full and economy versions

Computing SVDs

- SVD can be computed by computing eigenvalues and eigenvectors of matrix A^TA; square roots of eigenvalues of this matrix are the singular values of A; eigenvectors give columns of V above.
- Columns of *U* are eigenvectors of AA^T corresponding to its n largest eigenvalues
- Actual implementations may use more efficient algorithms
- Functions for computing SVD exist in many numerical packages (including OpenCV and Matlab).

Applications of SVD

- Solution of A***x*** = ***y*** in the least mean squared sense is given by

$$\boldsymbol{x} = \mathcal{A}^{\dagger}\boldsymbol{y} \quad \text{with} \quad \mathcal{A}^{\dagger} \stackrel{\text{def}}{=} [(\mathcal{A}^T\mathcal{A})^{-1}\mathcal{A}^T]$$

- Also, $\mathcal{A}^{\dagger} = (\mathcal{A}^T\mathcal{A})^{-1}\mathcal{A}^T = [(\mathcal{V}\mathcal{W}^T\mathcal{U}^T)(\mathcal{U}\mathcal{W}\mathcal{V}^T)]^{-1}(\mathcal{V}\mathcal{W}^T\mathcal{U}^T) = \mathcal{V}\mathcal{W}^{-1}\mathcal{U}^T.$

- If matrix A has rank r < q, we can rewrite U, W and V^T as:

$$\mathcal{U} = \begin{array}{|c|c|}\hline \mathcal{U}_r & \mathcal{U}_{q-r} \\ \hline\end{array}, \quad \mathcal{W} = \begin{array}{|c|c|}\hline \mathcal{W}_r & 0 \\ \hline 0 & 0 \\ \hline\end{array}, \quad \text{and} \quad \mathcal{V}^T = \begin{array}{|c|}\hline \mathcal{V}_r^T \\ \hline \mathcal{V}_{q-r}^T \\ \hline\end{array}$$

Theorem 6. When $\mathcal{A}$ has a rank greater than r, $\mathcal{U}_r\mathcal{W}_r\mathcal{V}_r^T$ is the best possible rank-r approximation of $\mathcal{A}$ in the sense of the Frobenius norm.[2]

The Frobenius norm of a matrix is the square root of the sum of the squares of its entries.

Estimating F (part of Alg 8.1)

Estimate $\mathcal{F}$.

(a) Compute Hartley's normalization transformation $\mathcal{T}$ and $\mathcal{T}'$, and the corresponding points $\tilde{\boldsymbol{p}}_i$ and $\tilde{\boldsymbol{p}}'_i$.

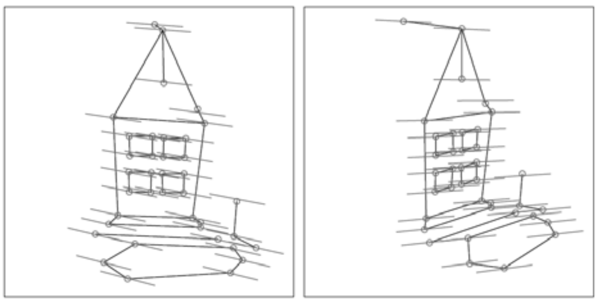
(b) Use homogeneous linear least squares to estimate the matrix $\tilde{\mathcal{F}}$ minimizing $\frac{1}{n}\sum_{i=1}^{n}(\tilde{\boldsymbol{p}}_i^T\tilde{\mathcal{F}}\tilde{\boldsymbol{p}}'_i)^2$ under the constraint $||\tilde{\mathcal{F}}||_F^2 = 1$.

(c) Compute the singular value decomposition $\mathcal{U}\mathrm{diag}(r, s, t)\mathcal{V}^T$ of $\tilde{\mathcal{F}}$, and set $\bar{\mathcal{F}} = \mathcal{U}\mathrm{diag}(r, s, 0)\mathcal{V}^T$.

(d) Output the fundamental matrix $\mathcal{F} = \mathcal{T}^T\bar{\mathcal{F}}\mathcal{T}'$.

- Hartley transformation: recommended that origin be at the average of data points and the average distance from origin be √2.

Weak Calibration Example



Two Camera Case: Given Intrinsic Parameters

- Compute essential matrix ε from fundamental matrix *F*
- Decompose ε by using singular valued decomposition. See step 2 in algorithm 8.1 (next slide)
- R and t define ε directly, going in the other direction requires some algebraic manipulation; we skip derivations, equations given with algorithm 8.1(four combinations of R and t are possible)
  - Note that the matrix W in step 3 (a) is not the matrix resulting from SVD of ε but instead one defined as:

$$\mathcal{W} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Knowing R and t, and matching points, we can compute point positions by triangulation as in the calibrated stereo case.
  - Reconstruction is possible only up to a similarity transform (up to scale and a rigid transformation)

Algorithm 8.1 (Derivations skipped)

1. Estimate $\mathcal{F}$.

(a) Compute Hartley's normalization transformation $\mathcal{T}$ and $\mathcal{T}'$, and the corresponding points $\tilde{p}_i$ and $\tilde{p}'_i$.

(b) Use homogeneous linear least squares to estimate the matrix $\tilde{\mathcal{F}}$ minimizing $\frac{1}{n}\sum_{i=1}^{n}(\tilde{p}_i^T \tilde{\mathcal{F}} \tilde{p}'_i)^2$ under the constraint $|\tilde{\mathcal{F}}|_F^2 = 1$.

(c) Compute the singular value decomposition $\mathcal{U}\text{diag}(r, s, t)\mathcal{V}^T$ of $\tilde{\mathcal{F}}$, and set $\tilde{\mathcal{F}} = \mathcal{U}\text{diag}(r, s, 0)\mathcal{V}^T$.

(d) Output the fundamental matrix $\mathcal{F} = \mathcal{T}^T \tilde{\mathcal{F}} \mathcal{T}'$.

2. Estimate $\mathcal{E}$.

(a) Compute the matrix $\hat{\mathcal{E}} = \mathcal{K}^T \mathcal{F} \mathcal{K}'$.

(b) Set $\mathcal{E} = \mathcal{U}\,\text{diag}(1, 1, 0)\mathcal{V}^T$, where $\mathcal{U}\mathcal{W}\mathcal{V}^T$ is the singular value decomposition of the matrix $\hat{\mathcal{E}}$.

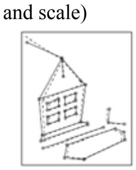
3. Compute $\mathcal{R}$ and t.

(a) Compute the rotation matrices $\mathcal{R}' = \mathcal{U}\mathcal{W}\mathcal{V}^T$ and $\mathcal{R}'' = \mathcal{U}\mathcal{W}^T\mathcal{V}^T$, and the translation vectors $t' = u_3$ and $t'' = -u_3$, where u_3 is the third column of the matrix $\mathcal{U}$.

(b) Output the combination of the rotation matrices $\mathcal{R}'$, $\mathcal{R}''$, and the translation vectors t', t'' such that the reconstructed points lie in front of both cameras.

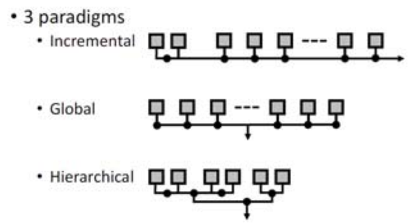
Reconstruction

- Given matching points and camera matrices, 3-d positions of these points can be computed by triangulation, as in stereo.
- Euclidean reconstruction for internally calibrated cameras
  - 7 parameter ambiguities remain (global rotation, global translation and scale)



Using Multiple Images

- Three Approaches



  – Figure from CVPR2017 Tutorial: Large-scale 3D modeling…
  – This tutorial is also a good source of current state-of-art in SFM from crowd sourced data

Incremental Approach

- Use two views to do an initial reconstruction
  – May select the two views that give the best result
    • Try all pairs and compare errors?
- Use the constructed 3-D model to estimate camera orientation of third camera
  – Refine estimates using all three cameras
    • Bundle adjustment

$$E = \frac{1}{mn}\sum_{i,j} ||\boldsymbol{p}_{ij} - \frac{1}{Z_{ij}}(\mathcal{R}_i \quad \boldsymbol{t}_i)\begin{pmatrix}\boldsymbol{P}_j \\ 1\end{pmatrix}||^2$$

  – Repeat to add more cameras

Generalizing SFM

- Use multiple images simultaneously (not just a pair at a time)
- Consider cases where internal calibration parameters are not known
  – Additional ambiguities emerge
  – We will start with the case of *affine* cameras where the number of unknowns is smaller than for perspective cameras
  – Reconstruction will not be Euclidean
    • Additional knowledge is needed for removing some of the ambiguities

Hierarchy of Transformations

- Euclidean: rotation and translation, shape and size do not change
- Similarity: allows for isotropic scale change
- Affine: preserves parallelism of lines and planes, but not angles or distances (some distance ratios preserved)
- Projective: parallelism not preserved; intersection, tangency and sign of Gaussian curvature preserved

Different constraints on the components of transformation matrix are implied for each case

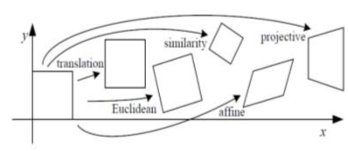
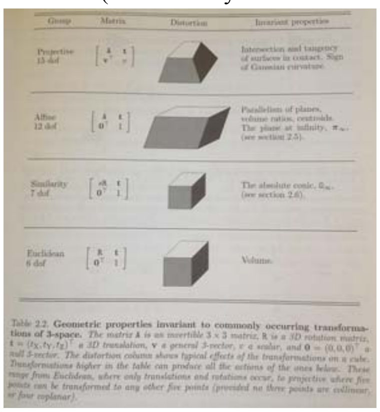


Figure 2.4: *Basic set of 2D planar transformations*

3D Transformations (from Hartley-Zisserman Book)

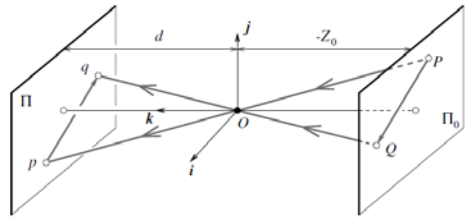


Group	Matrix	Distortion	Invariant properties
Projective 15 dof	$\begin{bmatrix} A & t \\ v^\top & v \end{bmatrix}$		Intersection and tangency of surfaces in contact. Sign of Gaussian curvature.
Affine 12 dof	$\begin{bmatrix} A & t \\ 0^\top & 1 \end{bmatrix}$		Parallelism of planes, volume ratios, centroids.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

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

Weak Perspective

Perspective projection but assume all points have the same z-value (object sizes small, compared to distance from camera)



$$x = -mX, \quad y = -mY, \quad \text{where} \quad m = -\frac{d}{Z_0}$$

Matrix form developed in next slide

Weak Perspective

- Equations become become simpler if we use homogeneous coordinates for P and non-homogeneous for image point p.
- Let Z_r be the distance of all points P; then, in normalized coordinate system

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \longrightarrow \begin{pmatrix} Z \\ Y \\ Z_r \end{pmatrix} \longrightarrow \begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix} = \begin{pmatrix} X/Z_r \\ Y/Z_r \\ 1 \end{pmatrix}$$

- In matrix form:

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix} = \frac{1}{Z_r} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & Z_r \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

- Including K, R and t

$$\boldsymbol{p} = \frac{1}{Z_r} \mathcal{K} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & Z_r \end{pmatrix} \begin{pmatrix} \mathcal{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{pmatrix} \boldsymbol{P}$$

Weak Perspective (Continued)

- Revisit *K*:

$$\mathcal{K} \stackrel{\text{def}}{=} \begin{pmatrix} \alpha & -\alpha \cot\theta & x_0 \\ 0 & \dfrac{\beta}{\sin\theta} & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Rewrite as: $\mathcal{K} = \begin{pmatrix} \mathcal{K}_2 & \boldsymbol{p}_0 \\ \boldsymbol{0}^T & 1 \end{pmatrix}$, where $\mathcal{K}_2 \stackrel{\text{def}}{=} \begin{pmatrix} \alpha & -\alpha \cot\theta \\ 0 & \dfrac{\beta}{\sin\theta} \end{pmatrix}$ and $\boldsymbol{p}_0 \stackrel{\text{def}}{=} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$

Rewrite weak perspective projection equation as:

$$\boldsymbol{p} = \mathcal{M}\boldsymbol{P}, \quad \text{where} \quad \mathcal{M} = \begin{pmatrix} \mathcal{A} & \boldsymbol{b} \end{pmatrix}$$

- Note *p* is a *non-homogeneous* coordinate vector here; M is 2x4

$$\mathcal{A} = \frac{1}{Z_r} \mathcal{K}_2 \mathcal{R}_2 \quad \text{and} \quad \boldsymbol{b} = \frac{1}{Z_r} \mathcal{K}_2 \boldsymbol{t}_2 + \boldsymbol{p}_0$$

R_2 is the sub-matrix of *R* consisting of the first two rows; t_2 contains the first two terms of vector *t*.

Note that t_3 does not appear in the projection equation.

Affine Cameras

- Affine projection matrix $p_{ij} = \mathcal{M}_i \begin{pmatrix} P_j \\ 1 \end{pmatrix} = \mathcal{A}_i P_j + b_i$
  - $\mathcal{M}_i$ is 2 x4 matrix which can be written as $\mathcal{M}_i = \begin{pmatrix} \mathcal{A}_i & b_i \end{pmatrix}$
  - Note: $\boldsymbol{p}_{ij}$ and $\boldsymbol{P}_j$ are both **non**-homogeneous coordinates
  - $\mathbf{A}_i$ is an arbitrary 2x3 matrix of rank 2, $\mathbf{b}_i$ is an arbitrary 2-vector
  - Weak perspective is a special case (FP 1.2.5) where

$$\mathcal{A} = \frac{1}{Z_r}\mathcal{K}_2\mathcal{R}_2 \quad \text{and} \quad b = \frac{1}{Z_r}\mathcal{K}_2 t_2 + p_0,$$

- Given *m* views and *n* points, 8*m*+3*n* unknowns, 2*mn* equations, we can solve given large enough *m* and *n.*

Affine Ambiguity

- Solution is ambiguous up to an affine transformation

If M_i and P_i are solutions to $p_{ij} = \mathcal{M}_i \begin{pmatrix} P_j \\ 1 \end{pmatrix} = \mathcal{A}_i P_j + b_i$

then so are $\mathrm{M'}_i$ and $\mathrm{P'}_i$, where

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

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

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

- Affine transformation is defined by 12 unknowns (in C and d above), so for affine reconstruction, equations relating unknowns become:

2*mn* >= 8*m*+3*n* – 12, for m =2, n = 4

- For 2 images, we need only 4 point match pairs for affine reconstruction

Affine Structure from a Motion Sequence

- Consider *m* cameras and *n* points $P_1,\ldots,P_n$; let P_0 be their center of mass.
- Let p_{ij} denote the image of j^{th} point in the i^{th} camera; p_{i0} is the image of P_0 in the i^{th} camera. Then,

$$p_{i0} = \mathcal{A}_i P_0 + b_i, \quad \text{and thus} \quad p_{ij} - p_{i0} = \mathcal{A}_i(P_j - P_0)$$

- Choose the world coordinate origin to be at P_0; let the i^{th} image coordinate origin be at p_{i0}, then we can rewrite above as:

$$p_{ij} = \mathcal{A}_i P_j \quad \text{for} \quad i = 1,\ldots,m \quad \text{and} \quad j = 1,\ldots,n,$$

- These *mn* equations can be written in matrix form as:

$$\mathcal{D} = \mathcal{A}\mathcal{P}, \text{ where } \mathcal{D} = \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \ldots & \ldots & \ldots \\ p_{m1} & \cdots & p_{mn} \end{pmatrix}, \mathcal{A} = \begin{pmatrix} \mathcal{A}_1 \\ \vdots \\ \mathcal{A}_m \end{pmatrix}, \text{ and } \mathcal{P} = \begin{pmatrix} P_1 & \ldots & P_n \end{pmatrix}$$

- Given D, we want to solve for A and P. If exact solution is not possible, we can minimize errors as follows:

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

Solving the Equations

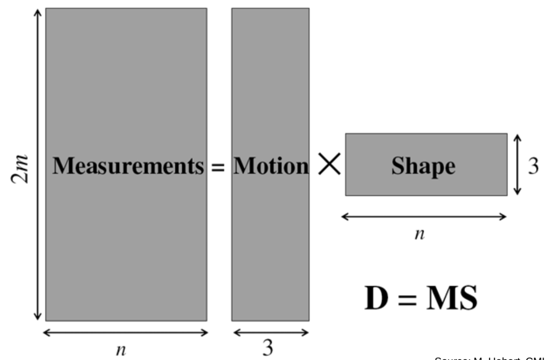
- Without noise: D is a rank-3 matrix (D is 2*m* x 3*n*, product of *A* which is 2*m* x 3 and *P* which is 3 x *n*)
  - We can decompose by using SVD to get A and P
  - Max 3 non-zero singular values
- With noise:
  - SVD may give more than 3 non-zero singular values
  - Best rank 3 approximation can be derived from SVD
- If a p x q matrix, $A = UW V^T$, has rank r < q, we can rewrite U, W and V^T as:

$$\mathcal{U} = \boxed{\mathcal{U}_r \mid \mathcal{U}_{q-r}}, \quad \mathcal{W} = \begin{array}{|c|c|} \hline \mathcal{W}_r & 0 \\ \hline 0 & 0 \\ \hline \end{array}, \quad \text{and} \quad \mathcal{V}^T = \begin{array}{|c|} \hline \mathcal{V}_r^T \\ \hline \mathcal{V}_{q-r}^T \\ \hline \end{array}$$

Theorem 6. When $\mathcal{A}$ has a rank greater than r, $\mathcal{U}_r\mathcal{W}_r\mathcal{V}_r^T$ is the best possible rank-r approximation of $\mathcal{A}$ in the sense of the Frobenius norm.[2]

- We can thus approximate D by using only the first 3 singular values and corresponding eigenvectors (see diagrams on following slides)

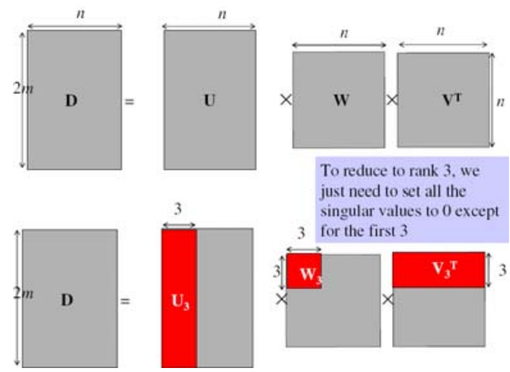
Factorizing the measurement matrix



Source: M. Hebert, CMU

25

Factorizing the measurement matrix

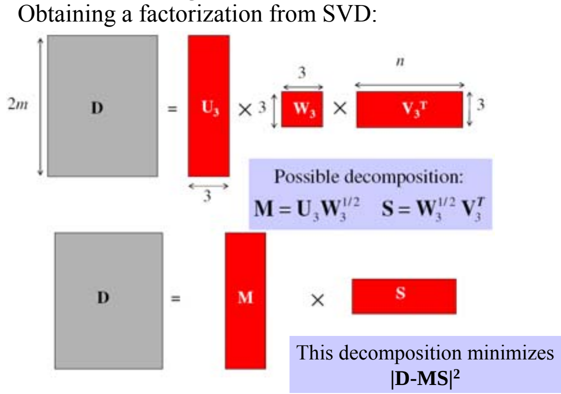


Source: M. Hebert, CMU

26

Factorizing the measurement matrix

- Obtaining a factorization from SVD:



Source: M. Hebert, CMU 27

Algorithm 8.2

1. Compute the singular value decomposition $\mathcal{D} = \mathcal{U}\mathcal{W}\mathcal{V}^T$.
2. Construct the matrices $\mathcal{U}_3$, $\mathcal{V}_3$, and $\mathcal{W}_3$ formed by the three leftmost columns of the matrices $\mathcal{U}$ and $\mathcal{V}$, and the corresponding 3×3 submatrix of $\mathcal{W}$.
3. Define

$$\mathcal{A}_0 = \mathcal{U}_3\sqrt{\mathcal{W}_3} \quad \text{and} \quad \mathcal{P}_0 = \sqrt{\mathcal{W}_3}\mathcal{V}_3^T;$$

the $2m \times 3$ matrix $\mathcal{A}_0$ is an estimate of the camera motion, and the $3 \times n$ matrix $\mathcal{P}_0$ is an estimate of the scene structure.

- Why $\sqrt{W_3}$? Actually, distribution of w_3 between and A and P is not important as we maintain an affine ambiguity.
- **Poor notation**: note that A_0 and P_0 do not refer to camera number 0 or the point number 0.

USC CS574: Computer Vision, Fall 2017

28

Algorithm 8.2

1. Compute the singular value decomposition $\mathcal{D} = \mathcal{U}\mathcal{W}\mathcal{V}^T$.
2. Construct the matrices $\mathcal{U}_3$, $\mathcal{V}_3$, and $\mathcal{W}_3$ formed by the three leftmost columns of the matrices $\mathcal{U}$ and $\mathcal{V}$, and the corresponding 3×3 submatrix of $\mathcal{W}$.
3. Define

$$\mathcal{A}_0 = \mathcal{U}_3\sqrt{\mathcal{W}_3} \quad \text{and} \quad \mathcal{P}_0 = \sqrt{\mathcal{W}_3}\mathcal{V}_3^T;$$

the $2m \times 3$ matrix $\mathcal{A}_0$ is an estimate of the camera motion, and the $3 \times n$ matrix $\mathcal{P}_0$ is an estimate of the scene structure.

- Why $\sqrt{W_3}$? Actually, distribution of w_3 between and A and P is not important as we maintain an affine ambiguity.
- **Poor notation**: note that A_0 and P_0 do not refer to camera number 0 or the point number 0.

Error Analysis

- This method requires all points to be visible and matched in all views
- If there are errors in matching, they will appear in SVD decomposition
  - Decomposition exists except under some degenerate conditions
- Error in rank 3 approximation of D matrix
  - May be able to assess if matches are incorrect
  - Note: matches s/b already consistent with epipolar lines
- Published literature does not seem to address this issue explicitly
- RANSAC could be used to select a subset of matches and then verify for others

Example from Six Images