Computer Vision - Lecture 21

Structure-from-Motion

29.01.2015

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de

leibe@vision.rwth-aachen.de

Many slides adapted from Svetlana Lazebnik, Martial Hebert, Steve Seitz
Announcements

• Exam
  - **1st Date:** Monday, 23.02., 13:30 - 17:30h
  - **2nd Date:** Thursday, 26.03., 09:30 - 12:30h
  - Closed-book exam, the core exam time will be 2h.
  - **Admission requirement:** 50% of the exercise points or passed test exam
  - We will send around an announcement with the exact starting times and places by email.

• Test exam
  - **Date:** Thursday, 05.02., 09:15 - 10:45h, room UMIC 025
  - Core exam time will be 1h
  - **Purpose:** Prepare you for the questions you can expect.
  - **Possibility to collect bonus exercise points!**
Announcements (2)

• Last lecture next Monday: Repetition
  - Summary of all topics in the lecture
  - “Big picture” and current research directions
  - Opportunity to ask questions

  Please use this opportunity and prepare questions!
Course Outline

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
  - Epipolar Geometry and Stereo Basics
  - Camera calibration & Uncalibrated Reconstruction
  - Active Stereo
- Motion
  - Motion and Optical Flow
- 3D Reconstruction (Reprise)
  - Structure-from-Motion
Recap: Estimating Optical Flow

- Given two subsequent frames, estimate the apparent motion field $u(x,y)$ and $v(x,y)$ between them.

- Key assumptions
  - **Brightness constancy**: projection of the same point looks the same in every frame.
  - **Small motion**: points do not move very far.
  - **Spatial coherence**: points move like their neighbors.
Recap: Lucas-Kanade Optical Flow

- Use all pixels in a $K \times K$ window to get more equations.
- Least squares problem:

  $$
  \begin{bmatrix}
  I_x(p_1) & I_y(p_1) \\
  I_x(p_2) & I_y(p_2) \\
  \vdots & \vdots \\
  I_x(p_{25}) & I_y(p_{25})
  \end{bmatrix}
  \begin{bmatrix}
  u \\
  v
  \end{bmatrix}
  =
  -
  \begin{bmatrix}
  I_t(p_1) \\
  I_t(p_2) \\
  \vdots \\
  I_t(p_{25})
  \end{bmatrix}
  \begin{bmatrix}
  A \\
  d = b
  \end{bmatrix}
  $$

- Minimum least squares solution given by solution of

  $$(A^T A) \quad d = A^T b$$

  $$
  \begin{bmatrix}
  \sum I_x I_x & \sum I_x I_y \\
  \sum I_x I_y & \sum I_y I_y
  \end{bmatrix}
  \begin{bmatrix}
  u \\
  v
  \end{bmatrix}
  =
  -
  \begin{bmatrix}
  \sum I_x I_t \\
  \sum I_y I_t
  \end{bmatrix}
  \begin{bmatrix}
  A^T A \\
  A^T b
  \end{bmatrix}
  $$

Recall the Harris detector!
Recap: Iterative Refinement

- Estimate velocity at each pixel using one iteration of LK estimation.
- Warp one image toward the other using the estimated flow field.
- Refine estimate by repeating the process.

- Iterative procedure
  - Results in subpixel accurate localization.
  - Converges for small displacements.
Recap: Coarse-to-fine Estimation

- Gaussian pyramid of image 1:
  - $u=10$ pixels
  - $u=5$ pixels
  - $u=2.5$ pixels
  - $u=1.25$ pixels

- Gaussian pyramid of image 2:
Recap: Coarse-to-fine Estimation

- Gaussian pyramid of image 1
- Run iterative L-K
- Warp & upsample
- Run iterative L-K
- Gaussian pyramid of image 2

Slide credit: Steve Seitz
Topics of This Lecture

• Structure from Motion (SfM)
  - Motivation
  - Ambiguity

• Affine SfM
  - Affine cameras
  - Affine factorization
  - Euclidean upgrade
  - Dealing with missing data

• Projective SfM
  - Two-camera case
  - Projective factorization
  - Bundle adjustment
  - Practical considerations

• Applications
Structure from Motion

- Given: \( m \) images of \( n \) fixed 3D points
  \[ x_{ij} = P_i X_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \]

- Problem: estimate \( m \) projection matrices \( P_i \) and \( n \) 3D points \( X_j \) from the \( mn \) correspondences \( x_{ij} \)

Slide credit: Svetlana Lazebnik
What Can We Use This For?

- E.g. movie special effects
Structure from Motion Ambiguity

- If we scale the entire scene by some factor $k$ and, at the same time, scale the camera matrices by the factor of $1/k$, the projections of the scene points in the image remain exactly the same:

$$x = PX = \left(\frac{1}{k}P\right)(kX)$$

$\implies$ It is impossible to recover the absolute scale of the scene!
Structure from Motion Ambiguity

- If we scale the entire scene by some factor $k$ and, at the same time, scale the camera matrices by the factor of $1/k$, the projections of the scene points in the image remain exactly the same.

- More generally: if we transform the scene using a transformation $Q$ and apply the inverse transformation to the camera matrices, then the images do not change.

\[ x = PX = (PQ^{-1})QX \]
Reconstruction Ambiguity: Similarity

\[ x = PX = (PQ_S^{-1})Q_SX \]
Reconstruction Ambiguity: Affine

\[ x = PX = (PQ_A^{-1})Q_AX \]
Reconstruction Ambiguity: Projective

\[ x = PX = (PQ_P^{-1})Q_PX \]
Projective Ambiguity
From Projective to Affine

Images from Hartley & Zisserman
From Affine to Similarity

Images from Hartley & Zisserman
Hierarchy of 3D Transformations

- **Projective**
  - 15dof
  - \[
  \begin{bmatrix}
  A & t \\
  v^T & v
  \end{bmatrix}
  \]
  - Preserves intersection and tangency

- **Affine**
  - 12dof
  - \[
  \begin{bmatrix}
  A & t \\
  0^T & 1
  \end{bmatrix}
  \]
  - Preserves parallellism, volume ratios

- **Similarity**
  - 7dof
  - \[
  \begin{bmatrix}
  sR & t \\
  0^T & 1
  \end{bmatrix}
  \]
  - Preserves angles, ratios of length

- **Euclidean**
  - 6dof
  - \[
  \begin{bmatrix}
  R & t \\
  0^T & 1
  \end{bmatrix}
  \]
  - Preserves angles, lengths

- With no constraints on the camera calibration matrix or on the scene, we get a *projective* reconstruction.
- Need additional information to *upgrade* the reconstruction to affine, similarity, or Euclidean.

Slide credit: Svetlana Lazebnik
Topics of This Lecture

- **Structure from Motion (SfM)**
  - Motivation
  - Ambiguity

- **Affine SfM**
  - Affine cameras
  - Affine factorization
  - Euclidean upgrade
  - Dealing with missing data

- **Projective SfM**
  - Two-camera case
  - Projective factorization
  - Bundle adjustment
  - Practical considerations

- **Applications**
Structure from Motion

- Let’s start with *affine cameras* (the math is easier)
Orthographic Projection

• Special case of perspective projection
  - Distance from center of projection to image plane is infinite

  Projection matrix:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= 
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\Rightarrow (x, y)
\]
Affine Cameras

Orthographic Projection

Parallel Projection

Slide credit: Svetlana Lazebnik
Affine Cameras

- A general affine camera combines the effects of an affine transformation of the 3D space, orthographic projection, and an affine transformation of the image:

\[
P = [3 \times 3 \text{ affine}] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} [4 \times 4 \text{ affine}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix}
\]

- Affine projection is a linear mapping + translation in inhomogeneous coordinates

\[
x = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = AX + b
\]

Projection of world origin

Slide credit: Svetlana Lazebnik
Affine Structure from Motion

• Given: \( m \) images of \( n \) fixed 3D points:
  \[ x_{ij} = A_i X_j + b_i, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \]

• Problem: use the \( mn \) correspondences \( x_{ij} \) to estimate \( m \) projection matrices \( A_i \) and translation vectors \( b_i \), and \( n \) points \( X_j \)

• The reconstruction is defined up to an arbitrary *affine* transformation \( Q \) (12 degrees of freedom):
  \[
  \begin{bmatrix}
  A & b \\
  0 & 1
  \end{bmatrix}
  \rightarrow
  \begin{bmatrix}
  A & b \\
  0 & 1
  \end{bmatrix}
  Q^{-1},
  \quad
  \begin{bmatrix}
  X \\
  1
  \end{bmatrix}
  \rightarrow
  Q
  \begin{bmatrix}
  X \\
  1
  \end{bmatrix}
  \]

• We have \( 2mn \) knowns and \( 8m + 3n \) unknowns (minus 12 dof for affine ambiguity).
  
  - Thus, we must have \( 2mn \geq 8m + 3n - 12 \).
  - For two views, we need four point correspondences.
Affine Structure from Motion

• Centering: subtract the centroid of the image points

\[
\hat{x}_{ij} = x_{ij} - \frac{1}{n} \sum_{k=1}^{n} x_{ik} = A_i X_j + b_i - \frac{1}{n} \sum_{k=1}^{n} (A_i X_k + b_i)
\]

\[
= A_i \left( X_j - \frac{1}{n} \sum_{k=1}^{n} X_k \right) = A_i \hat{X}_j
\]

• For simplicity, assume that the origin of the world coordinate system is at the centroid of the 3D points.
• After centering, each normalized point \( x_{ij} \) is related to the 3D point \( X_i \) by

\[
\hat{x}_{ij} = A_i X_j
\]
Affine Structure from Motion

- Let’s create a $2m \times n$ data (measurement) matrix:

$$D = \begin{bmatrix}
\hat{X}_{11} & \hat{X}_{12} & \cdots & \hat{X}_{1n} \\
\hat{X}_{21} & \hat{X}_{22} & \cdots & \hat{X}_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{X}_{m1} & \hat{X}_{m2} & \cdots & \hat{X}_{mn}
\end{bmatrix}$$

Cameras (2m)

Points (n)

---


Slide credit: Svetlana Lazebnik
Affine Structure from Motion

• Let’s create a $2m \times n$ data (measurement) matrix:

$$
D = \begin{bmatrix}
\hat{X}_{11} & \hat{X}_{12} & \ldots & \hat{X}_{1n} \\
\hat{X}_{21} & \hat{X}_{22} & \ldots & \hat{X}_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{X}_{m1} & \hat{X}_{m2} & \ldots & \hat{X}_{mn}
\end{bmatrix} = \begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_m
\end{bmatrix} \begin{bmatrix}
X_1 \\
X_2 \\
\vdots \\
X_n
\end{bmatrix}
$$

Points (3 × n)

Cameras (2m × 3)

• The measurement matrix $D = MS$ must have rank 3!

Factorizing the Measurement Matrix

\[ \text{Measurements} = \text{Motion} \times \text{Shape} \]

\[ \mathbf{D} = \mathbf{M} \mathbf{S} \]
Factorizing the Measurement Matrix

- Singular value decomposition of D:

\[ D = U W V^T \]
Factorizing the Measurement Matrix

- Singular value decomposition of $D$:

\[
\begin{align*}
D & = U W V_T \\
& = U_3 \begin{bmatrix} W_3 \\ I \end{bmatrix} \begin{bmatrix} V_3^T \\ I \end{bmatrix}
\end{align*}
\]

To reduce to rank 3, we just need to set all the singular values to 0 except for the first 3.
Factorizing the Measurement Matrix

• Obtaining a factorization from SVD:

\[ D = U_3 \times W_3 \times V_3^T \]
Factorizing the Measurement Matrix

- Obtaining a factorization from SVD:

\[
\begin{align*}
\mathbf{D} & = \mathbf{U}_3 \times 3 \mathbf{W}_3 \times \mathbf{V}_3^T \\
\mathbf{D} & = \mathbf{M} \times \mathbf{S}
\end{align*}
\]

Possible decomposition:
\[
\mathbf{M} = \mathbf{U}_3 \mathbf{W}_3^{1/2} \quad \mathbf{S} = \mathbf{W}_3^{1/2} \mathbf{V}_3^T
\]

This decomposition minimizes \(|\mathbf{D-MS}|^2|
Affine Ambiguity

- The decomposition is not unique. We get the same $D$ by using any $3 \times 3$ matrix $C$ and applying the transformations $M \rightarrow MC$, $S \rightarrow C^{-1}S$.
- That is because we have only an affine transformation and we have not enforced any Euclidean constraints (like forcing the image axes to be perpendicular, for example). We need a *Euclidean upgrade*.
Estimating the Euclidean Upgrade

• Orthographic assumption: image axes are perpendicular and scale is 1.

- This can be converted into a system of $3m$ equations:

  \[
  \begin{align*}
  \hat{a}_{i1} \cdot \hat{a}_{i2} &= 0 \\
  |\hat{a}_{i1}| &= 1 \\
  |\hat{a}_{i2}| &= 1 \\
  \end{align*}
  \]

  \[
  \begin{align*}
  a_{i1}^T C C^T a_{i2} &= 0 \\
  a_{i1}^T C C^T a_{i1} &= 1, \quad i = 1, \ldots, m \\
  a_{i2}^T C C^T a_{i2} &= 1 \\
  \end{align*}
  \]

  for the transformation matrix $C$ ⇒ goal: estimate $C$

Slide adapted from S. Lazebnik, M. Hebert
Estimating the Euclidean Upgrade

- System of $3m$ equations:
  \[
  \begin{cases}
  \hat{a}_{i1} \cdot \hat{a}_{i2} = 0 \\
  |\hat{a}_{i1}| = 1 \\
  |\hat{a}_{i2}| = 1
  \end{cases}
  \quad \Leftrightarrow \quad
  \begin{cases}
  a_{i1}^T C C^T a_{i2} = 0 \\
  a_{i1}^T C C^T a_{i1} = 1, \quad i = 1, \ldots, m \\
  a_{i2}^T C C^T a_{i2} = 1
  \end{cases}
  \]

- Let
  \[L = C C^T\]
  \[A_i = \begin{bmatrix} a_{i1}^T \\ a_{i2}^T \end{bmatrix}, \quad i = 1, \ldots, m\]

- Then this translates to $3m$ equations in $L$
  \[A_i L A_i^T = I, \quad i = 1, \ldots, m\]
  
  > Solve for $L$
  > Recover $C$ from $L$ by Cholesky decomposition: $L = C C^T$
  > Update $M$ and $S$: $M = MC$, $S = C^{-1}S$
Algorithm Summary

- Given: \( m \) images and \( n \) features \( x_{ij} \)
- For each image \( i \), center the feature coordinates.
- Construct a \( 2m \times n \) measurement matrix \( D \):
  - Column \( j \) contains the projection of point \( j \) in all views
  - Row \( i \) contains one coordinate of the projections of all the \( n \) points in image \( i \)
- Factorize \( D \):
  - Compute SVD: \( D = U W V^T \)
  - Create \( U_3 \) by taking the first 3 columns of \( U \)
  - Create \( V_3 \) by taking the first 3 columns of \( V \)
  - Create \( W_3 \) by taking the upper left \( 3 \times 3 \) block of \( W \)
- Create the motion and shape matrices:
  - \( M = U_3 W_3^{\frac{1}{2}} \) and \( S = W_3^{\frac{1}{2}} V_3^T \) (or \( M = U_3 \) and \( S = W_3 V_3^T \))
- Eliminate affine ambiguity

Slide credit: Martial Hebert
Reconstruction Results


Image Source: Tomasi & Kanade
Dealing with Missing Data

• So far, we have assumed that all points are visible in all views
• In reality, the measurement matrix typically looks something like this:
Dealing with Missing Data

• Possible solution: decompose matrix into dense sub-blocks, factorize each sub-block, and fuse the results
  - Finding dense maximal sub-blocks of the matrix is NP-complete (equivalent to finding maximal cliques in a graph)

• Incremental bilinear refinement

(1) Perform factorization on a dense sub-block

Dealing with Missing Data

- Possible solution: decompose matrix into dense sub-blocks, factorize each sub-block, and fuse the results
  - Finding dense maximal sub-blocks of the matrix is NP-complete (equivalent to finding maximal cliques in a graph)

- Incremental bilinear refinement

  1. Perform factorization on a dense sub-block
  2. Solve for a new 3D point visible by at least two known cameras (linear least squares)


Slide credit: Svetlana Lazebnik
Dealing with Missing Data

• Possible solution: decompose matrix into dense sub-blocks, factorize each sub-block, and fuse the results
  ➢ Finding dense maximal sub-blocks of the matrix is NP-complete (equivalent to finding maximal cliques in a graph)

• Incremental bilinear refinement

1. Perform factorization on a dense sub-block
2. Solve for a new 3D point visible by at least two known cameras (linear least squares)
3. Solve for a new camera that sees at least three known 3D points (linear least squares)


Slide credit: Svetlana Lazebnik
Comments: Affine SfM

• Affine SfM was historically developed first.
• It is valid under the assumption of *affine cameras*.
  - Which does not hold for real physical cameras...
  - ...but which is still tolerable if the scene points are far away from the camera.

• For good results with real cameras, we typically need projective SfM.
  - Harder problem, more ambiguity
  - Math is a bit more involved...
    (Here, only basic ideas. If you want to implement it, please look at the H&Z book for details).
Topics of This Lecture

• Structure from Motion (SfM)
  - Motivation
  - Ambiguity

• Affine SfM
  - Affine cameras
  - Affine factorization
  - Euclidean upgrade
  - Dealing with missing data

• Projective SfM
  - Two-camera case
  - Projective factorization
  - Bundle adjustment
  - Practical considerations

• Applications
Projective Structure from Motion

- Given: \( m \) images of \( n \) fixed 3D points
  \[
  x_{ij} = P_i X_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n
  \]
- Problem: estimate \( m \) projection matrices \( P_i \) and \( n \) 3D points \( X_j \) from the \( mn \) correspondences \( x_{ij} \)
Projective Structure from Motion

- Given: $m$ images of $n$ fixed 3D points
  - $z_{ij} x_{ij} = P_i X_j$, $i = 1, \ldots, m$, $j = 1, \ldots, n$

- Problem: estimate $m$ projection matrices $P_i$ and $n$ 3D points $X_j$ from the $mn$ correspondences $x_{ij}$

- With no calibration info, cameras and points can only be recovered up to a $4 \times 4$ projective transformation $Q$: $X \rightarrow QX$, $P \rightarrow PQ^{-1}$

- We can solve for structure and motion when $2mn \geq 11m + 3n - 15$

- For two cameras, at least 7 points are needed.
Projective SfM: Two-Camera Case

- Assume fundamental matrix $F$ between the two views
  - First camera matrix: $[I|0]Q^{-1}$
  - Second camera matrix: $[A|b]Q^{-1}$
- Let $\tilde{X} = QX$, then $zx = [I \mid 0]\tilde{X}$, $z'x' = [A|b]\tilde{X}$
- And
  $$z'x' = A[I \mid 0]\tilde{X} + b = zAx + b$$
  $$z'x' \times b = zAx \times b$$
  $$(z'x' \times b) \cdot x' = (zAx \times b) \cdot x'$$
  $$0 = (zAx \times b) \cdot x'$$
- So we have
  $$x'^T[b_{\times}]Ax = 0$$
  $$F = [b_{\times}]A \quad \text{b: epipole (}F^Tb = 0\text{)}, \quad A = -[b_{\times}]F$$
Projective SfM: Two-Camera Case

- This means that if we can compute the fundamental matrix between two cameras, we can directly estimate the two projection matrices from $F$.

- Once we have the projection matrices, we can compute the 3D position of any point $X$ by triangulation.

- How can we obtain both kinds of information at the same time?
Projective Factorization

\[
D = \begin{bmatrix}
    z_{11}x_{11} & z_{12}x_{12} & \cdots & z_{1n}x_{1n} \\
    z_{21}x_{21} & z_{22}x_{22} & \cdots & z_{2n}x_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    z_{m1}x_{m1} & z_{m2}x_{m2} & \cdots & z_{mn}x_{mn}
\end{bmatrix}
= \begin{bmatrix}
P_1 \\
P_2 \\
\vdots \\
P_m
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{bmatrix}
\]

Points (4 \times n)

Cameras (3m \times 4)

\[
D = MS \text{ has rank 4}
\]

- If we knew the depths \( z \), we could factorize \( D \) to estimate \( M \) and \( S \).
- If we knew \( M \) and \( S \), we could solve for \( z \).
- Solution: iterative approach (alternate between above two steps).
Sequential Structure from Motion

- Initialize motion from two images using fundamental matrix
- Initialize structure
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image - *calibration*

Slide credit: Svetlana Lazebnik
Sequential Structure from Motion

- Initialize motion from two images using fundamental matrix
- Initialize structure
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image - *calibration*
  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera - *triangulation*
Sequential Structure from Motion

- Initialize motion from two images using fundamental matrix
- Initialize structure
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image - *calibration*
  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera - *triangulation*
- Refine structure and motion: *bundle adjustment*

Slide credit: Svetlana Lazebnik
Bundle Adjustment

- Non-linear method for refining structure and motion
- Minimizing mean-square reprojection error

\[ E(P, X) = \sum_{i=1}^{m} \sum_{j=1}^{n} D(x_{ij}, P_i X_j)^2 \]
Bundle Adjustment

- Seeks the Maximum Likelihood (ML) solution assuming the measurement noise is Gaussian.
- It involves adjusting the bundle of rays between each camera center and the set of 3D points.
- Bundle adjustment should generally be used as the final step of any multi-view reconstruction algorithm.
  - Considerably improves the results.
  - Allows assignment of individual covariances to each measurement.
- However...
  - It needs a good initialization.
  - It can become an extremely large minimization problem.
- Very efficient algorithms available.
Projective Ambiguity

- If we don’t know anything about the camera or the scene, the best we can get with this is a reconstruction up to a projective ambiguity $Q$.
  - This can already be useful.
  - E.g. we can answer questions like “at what point does a line intersect a plane”?

- If we want to convert this to a “true” reconstruction, we need a *Euclidean upgrade*.
  - Need to put in additional knowledge about the camera (calibration) or about the scene (e.g. from markers).
  - Several methods available (see F&P Chapter 13.5 or H&Z Chapter 19)
Self-Calibration

- Self-calibration (auto-calibration) is the process of determining intrinsic camera parameters directly from uncalibrated images.
- For example, when the images are acquired by a single moving camera, we can use the constraint that the intrinsic parameter matrix remains fixed for all the images.
  - Compute initial projective reconstruction and find 3D projective transformation matrix \( Q \) such that all camera matrices are in the form \( P_i = K [R_i \mid t_i] \).
- Can use constraints on the form of the calibration matrix: square pixels, zero skew, fixed focal length, etc.
Practical Considerations (1)

1. Role of the baseline
   - Small baseline: large depth error
   - Large baseline: difficult search problem

• Solution
  - Track features between frames until baseline is sufficient.

Slide adapted from Steve Seitz
Practical Considerations (2)

2. There will still be many outliers
   - Incorrect feature matches
   - Moving objects

⇒ Apply RANSAC to get robust estimates based on the inlier points.

3. Estimation quality depends on the point configuration
   - Points that are close together in the image produce less stable solutions.

⇒ Subdivide image into a grid and try to extract about the same number of features per grid cell.
General Guidelines

• Use calibrated cameras wherever possible.
  ➢ It makes life so much easier, especially for SfM.

• SfM with 2 cameras is *far* more robust than with a single camera.
  ➢ Triangulate feature points in 3D using stereo.
  ➢ Perform 2D-3D matching to recover the motion.
  ➢ More robust to loss of scale (main problem of 1-camera SfM).

• Any constraint on the setup can be useful
  ➢ E.g. square pixels, zero skew, fixed focal length in each camera
  ➢ E.g. fixed baseline in stereo SfM setup
  ➢ E.g. constrained camera motion on a ground plane
  ➢ Making best use of those constraints may require adapting the algorithms (some known results are described in H&Z).
Structure-from-Motion: Limitations

- Very difficult to reliably estimate **metric** SfM unless
  - Large (x or y) motion  or
  - Large field-of-view and depth variation

- Camera calibration important for Euclidean reconstruction

- Need good feature tracker
Topics of This Lecture

• Structure from Motion (SfM)
  - Motivation
  - Ambiguity

• Affine SfM
  - Affine cameras
  - Affine factorization
  - Euclidean upgrade
  - Dealing with missing data

• Projective SfM
  - Two-camera case
  - Projective factorization
  - Bundle adjustment
  - Practical considerations

• Applications
Commercial Software Packages

- boujou
  (http://www.2d3.com/)
- PFTrack
  (http://www.thepixelfarm.co.uk/)
- MatchMover
  (http://www.realviz.com/)
- SynthEyes
  (http://www.ssontech.com/)
- Icarus
  (http://aig.cs.man.ac.uk/research/reveal/icarus/)
- Voodoo Camera Tracker
  (http://www.digilab.uni-hannover.de/)
Applications: Matchmoving

- Putting virtual objects into real-world videos

  **Original sequence**  **SfM results**  **Tracked features**  **Final video**

Videos from Stefan Hafeneger
Applications: Large-Scale SfM from Flickr


B. Leibe
References and Further Reading

- A (relatively short) treatment of affine and projective SfM and the basic ideas and algorithms can be found in Chapters 12 and 13 of


- More detailed information (if you really want to implement this) and better explanations can be found in Chapters 10, 18 (factorization) and 19 (self-calibration) of

  R. Hartley, A. Zisserman *Multiple View Geometry in Computer Vision* 2nd Ed., Cambridge Univ. Press, 2004