Computer Vision – Lecture 16

Camera Calibration & 3D Reconstruction

02.07.2019

Bastian Leibe
Visual Computing Institute
RWTH Aachen University
http://www.vision.rwth-aachen.de/

leibe@vision.rwth-aachen.de
Course Outline

• Image Processing Basics
• Segmentation & Grouping
• Object Recognition
• Local Features & Matching
• Deep Learning
• 3D Reconstruction
  ➢ Epipolar Geometry and Stereo Basics
  ➢ Camera calibration & Uncalibrated Reconstruction
  ➢ Structure-from-Motion
• Motion and Tracking
Recap: What Is Stereo Vision?

- Generic problem formulation: given several images of the same object or scene, compute a representation of its 3D shape.
Recap: Depth with Stereo – Basic Idea

• Basic Principle: **Triangulation**
  - Gives reconstruction as intersection of two rays
  - Requires
    - Camera pose (calibration)
    - Point correspondence

Slide credit: Steve Seitz
Recap: Epipolar Geometry

- Geometry of two views allows us to constrain where the corresponding pixel for some image point in the first view must occur in the second view.

- Epipolar constraint:
  - Correspondence for point $p$ in $\Pi$ must lie on the epipolar line $l'$ in $\Pi'$ (and vice versa).
  - Reduces correspondence problem to 1D search along conjugate epipolar lines.
Recap: Stereo Geometry With Calibrated Cameras

- Camera-centered coordinate systems are related by known rotation $R$ and translation $T$:

$$X' = RX + T$$
Recap: Essential Matrix

\[ X' \cdot (T \times RX) = 0 \]
\[ X' \cdot (T_x \ RX) = 0 \]

Let \( E = T_x R \)

\[ X'^T EX = 0 \]

- This holds for the rays \( p \) and \( p' \) that are parallel to the camera-centered position vectors \( X \) and \( X' \), so we have:

\[ p'^T E p = 0 \]

- \( E \) is called the **Essential matrix**, which relates corresponding image points [Longuet-Higgins 1981]
Recap: Essential Matrix and Epipolar Lines

Epipolar constraint: if we observe point \( p \) in one image, then its position \( p' \) in the second image must satisfy this equation.

\[
p'^T Ep = 0
\]

\( l' = Ep \) is the coordinate vector representing the epipolar line for point \( p \)

(i.e., the line is given by: \( l'^T x = 0 \))

\( l = E^T p' \) is the coordinate vector representing the epipolar line for point \( p' \)

Slide credit: Kristen Grauman
Recap: Stereo Image Rectification

- In practice, it is convenient if image scanlines are the epipolar lines.

- Algorithm
  - Reproject image planes onto a common plane parallel to the line between optical centers
  - Pixel motion is horizontal after this transformation
  - Two homographies (3x3 transforms), one for each input image reprojection
Recap: Dense Correspondence Search

• For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Examine all pixels on the epipolar line and pick the best match (e.g. SSD, correlation)
  - Triangulate the matches to get depth information

• This is easiest when epipolar lines are scanlines
  ⇒ Rectify images first

adapted from Svetlana Lazebnik, Li Zhang
Alternative: Sparse Correspondence Search

- Idea:
  - Restrict search to sparse set of detected features
  - Rather than pixel values (or lists of pixel values) use feature descriptor and an associated feature distance
  - Still narrow search further by epipolar geometry

What would make good features?

Slide credit: Kristen Grauman
Dense vs. Sparse

• Sparse
  - Efficiency
  - Can have more reliable feature matches, less sensitive to illumination than raw pixels
  - But…
    – Have to know enough to pick good features
    – Sparse information

• Dense
  - Simple process
  - More depth estimates, can be useful for surface reconstruction
  - But…
    – Breaks down in textureless regions anyway
    – Raw pixel distances can be brittle
    – Not good with very different viewpoints

Slide credit: Kristen Grauman
Difficulties in Similarity Constraint

Untextured surfaces

Occlusions

Slide credit: Kristen Grauman
Summary: Stereo Reconstruction

• Main Steps
  - Calibrate cameras
  - Rectify images
  - Compute disparity
  - Estimate depth

• So far, we have only considered calibrated cameras…

• Today
  - Uncalibrated cameras
  - Camera parameters
  - Revisiting epipolar geometry
  - Robust fitting
Recap: A General Point

- Equations of the form
  \[
  Ax = 0
  \]

- How do we solve them? (always!)
  - Apply SVD

\[
A = U D V^T = U \begin{bmatrix}
  d_{11} & \cdots \\
  \vdots & \ddots \\
  d_{NN} & \cdots
\end{bmatrix} V^T
\]

- Singular values of \( A \) = square roots of the eigenvalues of \( A^T A \).
- The solution of \( Ax = 0 \) is the **nullspace** vector of \( A \).
- This corresponds to the **smallest singular vector** of \( A \).
Topics of This Lecture

• **Camera Calibration**
  - Camera parameters
  - Calibration procedure

• **Revisiting Epipolar Geometry**
  - Triangulation
  - Calibrated case: Essential matrix
  - Uncalibrated case: Fundamental matrix
  - Weak calibration
  - Epipolar Transfer

• **Active Stereo**
  - Laser scanning
  - Kinect sensor
Recall: Pinhole Camera Model

\[(X, Y, Z) \mapsto (f \frac{X}{Z}, f \frac{Y}{Z})\]

\[
\begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix}
\mapsto
\begin{pmatrix}
\frac{fX}{Z} \\
\frac{fY}{Z}
\end{pmatrix} =
\begin{bmatrix}
f & 0 \\
f & 0 \\
0 & 1
\end{bmatrix}
\begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix}
\]

\[x = PX\]

Slide credit: Svetlana Lazebnik
Pinhole Camera Model

\[
\begin{pmatrix}
fx \\
fy \\
Z
\end{pmatrix} = 
\begin{bmatrix}
f & & \\
& f & & \\
& & 1 & & 0
\end{bmatrix}
\begin{pmatrix}
X \\
Y \\
Z
\end{pmatrix}
\]

\[
x = PX \\
P = \text{diag}(f, f, 1)[I | 0]
\]
Camera Coordinate System

- **Principal axis**: Line from the camera center perpendicular to the image plane
- **Normalized (camera) coordinate system**: Camera center is at the origin and the principal axis is the z-axis
- **Principal point** (p): Point where principal axis intersects the image plane (origin of normalized coordinate system)

Slide credit: Svetlana Lazebnik
Image from Hartley & Zisserman
Principal Point Offset

- Camera coordinate system: origin at the principal point
- Image coordinate system: origin is in the corner

Principal point: $\left( p_x, p_y \right)$
Principal Point Offset

Principal point: \((p_x, p_y)\)

\((X, Y, Z) \leftrightarrow (fX/Z + p_x, fY/Z + p_y)\)

\[
\begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix} \leftrightarrow \begin{pmatrix}
fX + Zp_x \\
fY + Zp_x \\
Z
\end{pmatrix} = \begin{bmatrix}
f & p_x & 0 \\
p_x & f & p_y & 0 \\
1 & 0 & 1 & 0
\end{bmatrix} \begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix}
\]

Slide credit: Svetlana Lazebnik

Image from Hartley & Zisserman
Principal Point Offset

Principal point: \((p_x, p_y)\)

\[
\begin{pmatrix}
  fX + Zp_x \\
  fY + Zp_x \\
  Z
\end{pmatrix} =
\begin{bmatrix}
  f & p_x \\
  f & p_y \\
  1 & 1 & 1 & 0
\end{bmatrix}
\begin{pmatrix}
  X \\
  Y \\
  Z \\
  1
\end{pmatrix}
\]

\[
K = \begin{bmatrix}
  f & p_x \\
  f & p_y \\
  1 & 1 & 1 & 1 & 0
\end{bmatrix}
\]

Calibration matrix

\[
P = K[I | 0]
\]

Slide credit: Svetlana Lazebnik

Image from Hartley & Zisserman
Pixel Coordinates: Non-Square Pixels

$m_x$ pixels per meter in horizontal direction, $m_y$ pixels per meter in vertical direction

$$K = \begin{bmatrix} m_x & m_y & 1 \\ f & p_x & \alpha_x \\ f & p_y & \alpha_y \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

Pixel size: $\frac{1}{m_x} \times \frac{1}{m_y}$

Slide credit: Svetlana Lazebnik
Camera Rotation and Translation

In general, the camera coordinate frame will be related to the world coordinate frame by a rotation and a translation.

\[ \tilde{X}_{cam} = R(\tilde{X} - \tilde{C}) \]

- **\( \tilde{X}_{cam} \)**: coords. of point in camera frame
- **\( \tilde{X} \)**: coords. of a point in world frame (nonhomogeneous)
- **\( \tilde{C} \)**: coords. of camera center in world frame

Image from Hartley & Zisserman

Slide credit: Svetlana Lazebnik
Camera Rotation and Translation

In non-homogeneous coordinates:

\[ \tilde{X}_{\text{cam}} = R (\tilde{X} - \tilde{C}) \]

\[
X_{\text{cam}} = \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \tilde{X} \\ 1 \end{bmatrix} = \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix}X
\]

\[
x = K [I | 0] X_{\text{cam}} = K [R | -RC] X \quad P = K [R | t], \quad t = -RC
\]

Note: \( C \) is the null space of the camera projection matrix \( PC = 0 \)
Summary: Camera Parameters

- **Intrinsic parameters**
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

\[
K = \begin{bmatrix}
m_x & m_y & f & S & p_x \\
1 & & 1 & & 1
\end{bmatrix}
= \begin{bmatrix}
\alpha_x & S & x_0 \\
\alpha_y & y_0 & 1
\end{bmatrix}
\]
Summary: Camera Parameters

- **Intrinsic parameters**
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew* (*non-rectangular pixels*)
  - *Radial distortion*

- **Extrinsic parameters**
  - Rotation $\mathbf{R}$
  - Translation $\mathbf{t}$
    (both relative to world coordinate system)

- **Camera projection matrix**
  \[
  \mathbf{P} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} = \begin{bmatrix}
  P_{11} & P_{12} & P_{13} & P_{14} \\
  P_{21} & P_{22} & P_{23} & P_{24} \\
  P_{31} & P_{32} & P_{33} & P_{34}
  \end{bmatrix}
  \]

*How many degrees of freedom does $\mathbf{P}$ have?*

Slide adapted from Svetlana Lazebnik
Camera Parameters: Degrees of Freedom

- **Intrinsic parameters**
  - Principal point coordinates: 2 DoF
  - Focal length: 1 DoF
  - Pixel magnification factors: 1 DoF
  - Skew (non-rectangular pixels): 1 DoF
  - Radial distortion

- **Extrinsic parameters**
  - Rotation $R$: 3 DoF
  - Translation $t$: 3 DoF
    (both relative to world coordinate system)

- **Camera projection matrix**
  - General pinhole camera: 9 DoF
  - CCD Camera with square pixels: 10 DoF
  - General camera: 11 DoF

$$K = \begin{bmatrix}
\phi_x & s & p_x \xi_0 \\
\phi_y & p_y \eta_0 & 1
\end{bmatrix}$$

$$P = K[R | t]$$
Calibrating a Camera

- Compute intrinsic and extrinsic parameters using observed camera data.

Main idea

- Place “calibration object” with known geometry in the scene
- Get correspondences
- Solve for mapping from scene to image: estimate $P = P_{\text{int}} P_{\text{ext}}$
Camera Calibration

- Given n points with known 3D coordinates $X_i$ and known image projections $x_i$, estimate the camera parameters.
Camera Calibration: Obtaining the Points

• For best results, it is important that the calibration points are measured with *subpixel accuracy*.
• How this can be done depends on the exact pattern.

• Algorithm for checkerboard pattern

  1. Perform Canny edge detection.
  2. Fit straight lines to detected linked edges.
  3. Intersect lines to obtain corners.

    > If sufficient care is taken, the points can then be obtained with localization accuracy < 1/10 pixel.

• Rule of thumb

  > Number of constraints should exceed number of unknowns by a factor of five.

  ⇒ For 11 parameters of P, at least 28 points should be used.
Camera Calibration: DLT Algorithm

\[ \lambda x_i = PX_i \]

\[
\begin{bmatrix}
    x_i \\
    y_i \\
    1
\end{bmatrix} = 
\begin{bmatrix}
    P_{11} & P_{12} & P_{13} & P_{14} \\
    P_{21} & P_{22} & P_{23} & P_{24} \\
    P_{31} & P_{32} & P_{33} & P_{34}
\end{bmatrix}
\begin{bmatrix}
    X_{i,1} \\
    X_{i,2} \\
    X_{i,3} \\
    1
\end{bmatrix} = 
\begin{bmatrix}
    P_1^T \\
    P_2^T \\
    P_3^T
\end{bmatrix} X_i
\]

\[ x_i \times PX_i = 0 \]

\[
\begin{bmatrix}
    0 & -X^T \\
    X^T & 0 & y_iX^T \\
    -y_iX^T & x_iX^T & 0
\end{bmatrix} 
\begin{bmatrix}
    P_1 \\
    P_2 \\
    P_3
\end{bmatrix} = 0
\]

Only two linearly independent equations
Camera Calibration: DLT Algorithm

\[
\begin{bmatrix}
0^T & X_1^T & -y_1X_1^T \\
X_1^T & 0^T & -x_1X_1^T \\
\vdots & \vdots & \vdots \\
0^T & X_n^T & -y_nX_n^T \\
X_n^T & 0^T & -x_nX_n^T
\end{bmatrix}
\begin{bmatrix}
P_1 \\
P_2 \\
P_3
\end{bmatrix} = 0 \quad \text{Ap} = 0
\]

Solve using… SVD!

• Notes
  - P has 11 degrees of freedom (12 parameters, but scale is arbitrary).
  - One 2D/3D correspondence gives us two linearly independent equations.
  - Homogeneous least squares (similar to homography est.)
  - \( \Rightarrow \) 5 \( \frac{1}{2} \) correspondences needed for a minimal solution.
Camera Calibration: DLT Algorithm

\[
\begin{bmatrix}
0^T & X_1^T & -y_1X_1^T \\
X_1^T & 0^T & -x_1X_1^T \\
\vdots & \vdots & \vdots \\
0^T & X_n^T & -y_nX_n^T \\
X_n^T & 0^T & -x_nX_n^T
\end{bmatrix}
\begin{pmatrix}
P_1 \\
P_2 \\
P_3
\end{pmatrix} = 0 \quad Ap = 0
\]

• Notes
  - For coplanar points that satisfy \( \Pi^T X = 0 \),
    we will get degenerate solutions \((\Pi,0,0), (0,\Pi,0),\) or \((0,0,\Pi)\).
  - \( \Rightarrow \) We need calibration points in more than one plane!
Camera Calibration

- Once we’ve recovered the numerical form of the camera matrix, we still have to figure out the intrinsic and extrinsic parameters.
- This is a matrix decomposition problem, not an estimation problem (see F&P sec. 3.2, 3.3)
Camera Calibration: Some Practical Tips

• For numerical reasons, it is important to carry out some data normalization.
  - Translate the image points $x_i$ to the (image) origin and scale them such that their RMS distance to the origin is $\sqrt{2}$.
  - Translate the 3D points $X_i$ to the (world) origin and scale them such that their RMS distance to the origin is $\sqrt{3}$.
  - (This is valid for compact point distributions on calibration objects).

• The DLT algorithm presented here is easy to implement, but there are some more accurate algorithms available (see H&Z sec. 7.2).

• For practical applications, it is also often needed to correct for radial distortion. Algorithms for this can be found in H&Z sec. 7.4, or F&P sec. 3.3.
Topics of This Lecture

• Camera Calibration
  ➢ Camera parameters
  ➢ Calibration procedure

• Revisiting Epipolar Geometry
  ➢ Triangulation
  ➢ Calibrated case: Essential matrix
  ➢ Uncalibrated case: Fundamental matrix
  ➢ Weak calibration
  ➢ Epipolar Transfer

• Active Stereo
  ➢ Laser scanning
  ➢ Kinect sensor
Two-View Geometry

• **Scene geometry (structure):**
  - Given corresponding points in two or more images, where is the pre-image of these points in 3D?

• **Correspondence (stereo matching):**
  - Given a point in just one image, how does it constrain the position of the corresponding point $x'$ in another image?

• **Camera geometry (motion):**
  - Given a set of corresponding points in two images, what are the cameras for the two views?
Revisiting Triangulation

- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point.

Slide credit: Svetlana Lazebnik
Revisiting Triangulation

- We want to intersect the two visual rays corresponding to $x_1$ and $x_2$, but because of noise and numerical errors, they will never meet exactly. How can this be done?
Triangulation: 1) Geometric Approach

• Find shortest segment connecting the two viewing rays and let \( X \) be the midpoint of that segment.
Triangulation: 2) Linear Algebraic Approach

\[ \lambda_1 x_1 = P_1 X \quad x_1 \times P_1 X = 0 \quad [x_1 \times] P_1 X = 0 \]

\[ \lambda_2 x_2 = P_2 X \quad x_2 \times P_2 X = 0 \quad [x_2 \times] P_2 X = 0 \]

Cross product as matrix multiplication:

\[
\mathbf{a} \times \mathbf{b} = \begin{bmatrix}
0 & -a_z & a_y \\
-a_z & 0 & -a_x \\
-a_y & a_x & 0 \\
\end{bmatrix}
\begin{bmatrix}
b_x \\
b_y \\
b_z \\
\end{bmatrix} = [\mathbf{a} \times] \mathbf{b}
\]
Triangulation: 2) Linear Algebraic Approach

\[ \lambda_1 x_1 = P_1 X \quad x_1 \times P_1 X = 0 \quad [x_{1x}]P_1X = 0 \]

\[ \lambda_2 x_2 = P_2 X \quad x_2 \times P_2 X = 0 \quad [x_{2x}]P_2X = 0 \]

Two independent equations each in terms of three unknown entries of \( X \)

\[ \Rightarrow \text{Stack them and solve using SVD!} \]

- This approach is often preferable to the geometric approach, since it nicely generalizes to multiple cameras.

Slide credit: Svetlana Lazebnik
Triangulation: 3) Nonlinear Approach

- Find $X$ that minimizes

$$d^2(x_1, P_1 X) + d^2(x_2, P_2 X)$$
Triangulation: 3) Nonlinear Approach

- Find $X$ that minimizes
  \[ d^2(x_1, P_1X) + d^2(x_2, P_2X) \]

- This approach is the most accurate, but unlike the other two methods, it doesn’t have a closed-form solution.

- Iterative algorithm
  - Initialize with linear estimate.
  - Optimize with Gauss-Newton or Levenberg-Marquardt (see F&P sec. 3.1.2 or H&Z Appendix 6).
Revisiting Epipolar Geometry

• Let’s look again at the epipolar constraint
  ➢ For the calibrated case (but in homogenous coordinates)
  ➢ For the uncalibrated case
Epipolar Geometry: Calibrated Case

Camera matrix: \([I|0]\)

\[ X = (u, v, w, 1)^T \]

\[ x = (u, v, w)^T \]

Camera matrix: \([R^T \; -R^T t]\)

Vector \(x'\) in second coord. system has coordinates \(Rx'\) in the first one.

The vectors \(x, t,\) and \(Rx'\) are coplanar

Slide credit: Svetlana Lazebnik
Epipolar Geometry: Calibrated Case

\[ x \cdot [t \times (Rx')] = 0 \quad \iff \quad x^T E x' = 0 \quad \text{with} \quad E = [t \times] R \]

Essential Matrix
(Longuet-Higgins, 1981)
Epipolar Geometry: Calibrated Case

- $x \cdot [t \times (Rx')] = 0 \quad \Rightarrow \quad x^T Ex' = 0 \quad \text{with} \quad E = [t_x]R$

- $E x'$ is the epipolar line associated with $x'$ ($l = E x'$)
- $E^T x$ is the epipolar line associated with $x$ ($l' = E^T x$)
- $E e' = 0$ and $E^T e = 0$ \textit{Why?}
- $E$ is singular (rank two) \textit{Why?}
- $E$ has five degrees of freedom (up to scale)

Slide credit: Svetlana Lazebnik
Epipolar Geometry: Uncalibrated Case

- The calibration matrices $K$ and $K'$ of the two cameras are unknown.
- We can write the epipolar constraint in terms of unknown normalized coordinates:

\[ \hat{x}^T E \hat{x}' = 0 \quad x = K \hat{x}, \quad x' = K' \hat{x}' \]
Epipolar Geometry: Uncalibrated Case

\[ \hat{x}^T E \hat{x}' = 0 \quad \Rightarrow \quad x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1} \]

\[
\begin{align*}
    x &= K \hat{x} \\
    x' &= K' \hat{x}'
\end{align*}
\]

Fundamental Matrix
(Faugeras and Luong, 1992)
Epipolar Geometry: Uncalibrated Case

\[ \hat{x}^T E \hat{x}' = 0 \Rightarrow x^T F x' = 0 \quad \text{with} \quad F = K'^{-T} E K^{-1} \]

- \( F x' \) is the epipolar line associated with \( x' \) (\( l = F x' \))
- \( F^T x \) is the epipolar line associated with \( x \) (\( l' = F^T x \))
- \( F e' = 0 \) and \( F^T e = 0 \)
- \( F \) is singular (rank two)
- \( F \) has seven degrees of freedom

Slide credit: Svetlana Lazebnik
Estimating the Fundamental Matrix

• The Fundamental matrix defines the epipolar geometry between two uncalibrated cameras.
• How can we estimate F from an image pair?
  ➢ We need correspondences…
The Eight-Point Algorithm

\[ x = (u, v, 1)^T, \quad x' = (u', v', 1)^T \]

\[
\begin{pmatrix}
F_{11} & F_{12} & F_{13} \\
F_{21} & F_{22} & F_{23} \\
F_{31} & F_{32} & F_{33}
\end{pmatrix}
\begin{pmatrix}
u' \\
v'
\end{pmatrix}
= 0
\]

\[
\begin{bmatrix}
F_{11} \\
F_{12} \\
F_{13} \\
F_{21} \\
F_{22} \\
F_{23} \\
F_{31} \\
F_{32} \\
F_{33}
\end{bmatrix}
= 0
\]

- Taking 8 correspondences:

\[
\begin{bmatrix}
u_1 \quad u_1 & u_1' & v_1' & v_1 & u_1 & v_1 & 1 \\
u_2 \quad u_2 & u_2' & v_2' & v_2 & u_2 & v_2 & 1 \\
u_3 \quad u_3 & u_3' & v_3' & v_3 & u_3 & v_3 & 1 \\
u_4 \quad u_4 & u_4' & v_4' & v_4 & u_4 & v_4 & 1 \\
u_5 \quad u_5 & u_5' & v_5' & v_5 & u_5 & v_5 & 1 \\
u_6 \quad u_6 & u_6' & v_6' & v_6 & u_6 & v_6 & 1 \\
u_7 \quad u_7 & u_7' & v_7' & v_7 & u_7 & v_7 & 1 \\
u_8 \quad u_8 & u_8' & v_8' & v_8 & u_8 & v_8 & 1
\end{bmatrix}
\begin{bmatrix}
F_{11} \\
F_{12} \\
F_{13} \\
F_{21} \\
F_{22} \\
F_{23} \\
F_{31} \\
F_{32} \\
F_{33}
\end{bmatrix}
= 0
\]

Solve using… SVD!

This minimizes:

\[
\sum_{i=1}^{N} (x_i^T F x'_i)^2
\]
Excursion: Properties of SVD

- **Frobenius norm**
  - Generalization of the Euclidean norm to matrices
  \[
  \|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\sum_{i=1}^{\min(m,n)} \sigma_i^2}
  \]

- **Partial reconstruction property of SVD**
  - Let \(\sigma_i i=1,\ldots,N\) be the singular values of \(A\).
  - Let \(A_p = U_p D_p V_p^T\) be the reconstruction of \(A\) when we set \(\sigma_{p+1},\ldots,\sigma_N\) to zero.
  - Then \(A_p = U_p D_p V_p^T\) is the best rank-p approximation of \(A\) in the sense of the Frobenius norm (i.e. the best least-squares approximation).
The Eight-Point Algorithm

- Problem with noisy data
  - The solution will usually not fulfill the constraint that $F$ only has rank 2.
  - \textit{There will be no epipoles through which all epipolar lines pass!}

- Enforce the rank-2 constraint using SVD

\[
F = UDV^T = U \begin{bmatrix}
d_{11} \\
d_{22} \\
d_{33}
\end{bmatrix} \begin{bmatrix}
v_{11} & \cdots & v_{13}
\vdots & \ddots & \vdots \\
v_{31} & \cdots & v_{33}
\end{bmatrix}^T
\]

- As we have just seen, this provides the best least-squares approximation to the rank-2 solution.

Set $d_{33}$ to zero and reconstruct $F$
Problem with the Eight-Point Algorithm

• In practice, this often looks as follows:

\[
\begin{bmatrix}
 u'_1 u_1 & u'_1 v_1 & u'_1 & u'_1 v'_1 & v'_1 & u_1 & v_1 & 1 \\
 u'_2 u_2 & u'_2 v_2 & u'_2 & u'_2 v'_2 & v'_2 & u_2 & v_2 & 1 \\
 u'_3 u_3 & u'_3 v_3 & u'_3 & u'_3 v'_3 & v'_3 & u_3 & v_3 & 1 \\
 u'_4 u_4 & u'_4 v_4 & u'_4 & u'_4 v'_4 & v'_4 & u_4 & v_4 & 1 \\
 u'_5 u_5 & u'_5 v_5 & u'_5 & u'_5 v'_5 & v'_5 & u_5 & v_5 & 1 \\
 u'_6 u_6 & u'_6 v_6 & u'_6 & u'_6 v'_6 & v'_6 & u_6 & v_6 & 1 \\
 u'_7 u_7 & u'_7 v_7 & u'_7 & u'_7 v'_7 & v'_7 & u_7 & v_7 & 1 \\
 u'_8 u_8 & u'_8 v_8 & u'_8 & u'_8 v'_8 & v'_8 & u_8 & v_8 & 1 \\
\end{bmatrix} \begin{bmatrix}
 F_{11} \\
 F_{12} \\
 F_{13} \\
 F_{21} \\
 F_{22} \\
 F_{23} \\
 F_{31} \\
 F_{32} \\
 F_{33} \\
\end{bmatrix} = \begin{bmatrix}
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
\end{bmatrix}
\]
Problem with the Eight-Point Algorithm

- In practice, this often looks as follows:

\[
\begin{bmatrix}
250906.36 & 183269.57 & 921.81 & 200931.10 & 146766.13 & 738.21 & 272.19 & 198.81 \\
2692.28 & 131633.03 & 176.27 & 6196.73 & 302975.59 & 405.71 & 15.27 & 746.79 \\
416374.23 & 871684.30 & 935.47 & 408110.89 & 654384.92 & 916.90 & 445.10 & 931.81 \\
191183.60 & 171759.40 & 410.27 & 416435.62 & 374125.90 & 893.65 & 465.99 & 418.65 \\
48988.86 & 30401.76 & 57.89 & 298604.57 & 185309.58 & 352.87 & 846.22 & 525.15 \\
116407.01 & 2727.75 & 138.89 & 169941.27 & 3982.21 & 202.77 & 838.12 & 19.64 \\
135384.58 & 75411.13 & 198.72 & 411350.03 & 229127.78 & 603.79 & 681.28 & 379.48
\end{bmatrix}
\begin{bmatrix}
F_{11} \\
F_{12} \\
F_{13} \\
F_{21} \\
F_{22} \\
F_{23} \\
F_{31} \\
F_{32} \\
F_{33}
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]

⇒ Poor numerical conditioning
⇒ Can be fixed by rescaling the data
The Normalized Eight-Point Algorithm

1. Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels.

2. Use the eight-point algorithm to compute $F$ from the normalized points.

3. Enforce the rank-2 constraint using SVD.

\[
F = U D V^T = U \begin{bmatrix}
d_{11} & & \\
& d_{22} & \\
& & d_{33}
\end{bmatrix} \begin{bmatrix}
v_{11} & \cdots & v_{13}
\end{bmatrix}^T
\]

Set $d_{33}$ to zero and reconstruct $F$

4. Transform fundamental matrix back to original units: if $T$ and $T'$ are the normalizing transformations in the two images, than the fundamental matrix in original coordinates is $T^T F T'$.
The Eight-Point Algorithm

• Meaning of error $\sum_{i=1}^{N} (x_i^T F x'_i)^2$:

Sum of Euclidean distances between points $x_i$ and epipolar lines $Fx'_i$ (or points $x'_i$ and epipolar lines $F^T x_i$), multiplied by a scale factor

• Nonlinear approach for refining the solution: minimize

$$\sum_{i=1}^{N} \left[ d^2 (x_i, F x'_i) + d^2 (x'_i, F^T x_i) \right]$$

- Similar to nonlinear minimization approach for triangulation.
- Iterative approach (Gauss-Newton, Levenberg-Marquardt,…)
Comparison of Estimation Algorithms

<table>
<thead>
<tr>
<th></th>
<th>8-point</th>
<th>Normalized 8-point</th>
<th>Nonlinear least squares</th>
</tr>
</thead>
<tbody>
<tr>
<td>Av. Dist. 1</td>
<td>2.33 pixels</td>
<td>0.92 pixel</td>
<td>0.86 pixel</td>
</tr>
<tr>
<td>Av. Dist. 2</td>
<td>2.18 pixels</td>
<td>0.85 pixel</td>
<td>0.80 pixel</td>
</tr>
</tbody>
</table>

Slide credit: Svetlana Lazebnik
3D Reconstruction with Weak Calibration

• Want to estimate world geometry without requiring calibrated cameras.

• Many applications:
  - Archival videos
  - Photos from multiple unrelated users
  - Dynamic camera system

• Main idea:
  - Estimate epipolar geometry from a (redundant) set of point correspondences between two uncalibrated cameras.
Stereo Pipeline with Weak Calibration

• So, where to start with uncalibrated cameras?
  - Need to find fundamental matrix $F$ and the correspondences (pairs of points $(u',v') \leftrightarrow (u,v)$).

• Procedure
  1. Find interest points in both images
  2. Compute correspondences
  3. Compute epipolar geometry
  4. Refine

Slide credit: Kristen Grauman

Example from Andrew Zisserman
Stereo Pipeline with Weak Calibration

1. Find interest points (e.g. Harris corners)
Stereo Pipeline with Weak Calibration

2. Match points using only proximity

Slide credit: Kristen Grauman

Example from Andrew Zisserman
Putative Matches based on Correlation Search

- Many wrong matches (10-50%), but enough to compute $F$
RANSAC for Robust Estimation of F

- Select random sample of correspondences
- Compute F using them
  - This determines epipolar constraint
- Evaluate amount of support – number of inliers within threshold distance of epipolar line
- Iterate until a solution with sufficient support has been found (or for max #iterations)
- Choose F with most support (#inliers)
Putative Matches based on Correlation Search

- Many wrong matches (10-50%), but enough to compute F

Example from Andrew Zisserman
Pruned Matches

- Correspondences consistent with epipolar geometry
Resulting Epipolar Geometry

Example from Andrew Zisserman
Epipolar Transfer

- Assume the epipolar geometry is known
- Given projections of the same point in two images, how can we compute the projection of that point in a third image?

Slide credit: Svetlana Lazebnik
Extension: Epipolar Transfer

- Assume the epipolar geometry is known
- Given projections of the same point in two images, how can we compute the projection of that point in a third image?

\[ l_{31} = F_{13}^T x_1 \]
\[ l_{32} = F_{23}^T x_2 \]

When does epipolar transfer fail?

Slide credit: Svetlana Lazebnik
Topics of This Lecture

• Camera Calibration
  ➢ Camera parameters
  ➢ Calibration procedure

• Revisiting Epipolar Geometry
  ➢ Triangulation
  ➢ Calibrated case: Essential matrix
  ➢ Uncalibrated case: Fundamental matrix
  ➢ Weak calibration
  ➢ Epipolar Transfer

• Active Stereo
  ➢ Laser scanning
  ➢ Kinect sensor
Microsoft Kinect – How Does It Work?

- Built-in IR projector
- IR camera for depth
- Regular camera for color
Recall: Optical Triangulation

Image plane

Camera center

3D Scene point

$\mathbf{x}_1$

$\mathbf{x}$?
Recall: Optical Triangulation

• Principle: 3D point given by intersection of two rays.
  - Crucial information: point correspondence
  - Most expensive and error-prone step in the pipeline…
Active Stereo with Structured Light

• Idea: Replace one camera by a projector.
  - Project “structured” light patterns onto the object
  - Simplifies the correspondence problem
What the Kinect Sees…
3D Reconstruction with the Kinect
Laser Scanning

- **Optical triangulation**
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/
Laser Scanned Models

*The Digital Michelangelo Project*, Levoy et al.

Slide credit: Steve Seitz
Laser Scanned Models

The Digital Michelangelo Project, Levoy et al.
Laser Scanned Models

*The Digital Michelangelo Project*, Levoy et al.
Laser Scanned Models

The Digital Michelangelo Project, Levoy et al.

Slide credit: Steve Seitz
Laser Scanned Models

*The Digital Michelangelo Project*, Levoy et al.

Slide credit: Steve Seitz

B. Leibe
Poor Man’s Scanner

The idea

Desk
Lamp

Stick or pencil

Camera

Desk
Slightly More Elaborate (But Still Cheap)

Software freely available from Robotics Institute TU Braunschweig
http://www.david-laserscanner.com/

B. Leibe
References and Further Reading

• Background information on camera models and calibration algorithms can be found in Chapters 6 and 7 of

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

• Also recommended: Chapter 9 of the same book on Epipolar geometry and the Fundamental Matrix and Chapter 11.1-11.6 on automatic computation of F.