Epipolar Geometry, Feature Detection, and Feature Matching

Multi-View Geometry

- Different views of a scene are not unrelated
- Relationships exist between two, three and more cameras

- Question: Given an image point in one image, how does this restrict the position of the corresponding image point in another image?


Three Questions

1. **Correspondence geometry:** Given an image point \( x \) in the first view, how does this constrain the position of the corresponding point \( x' \) in the second image?

2. **Camera geometry (motion):** Given a set of corresponding image points \( \{x_i \leftrightarrow x'_i\} \), \( i = 1, \ldots, n \), what are the cameras \( P \) and \( P' \) for the two views?

3. **Scene geometry (structure):** Given corresponding image points \( x_i \leftrightarrow x'_i \), and cameras \( P, P' \), what is the position of (their pre-image) \( X \) in the 3D world?

Applications: Stereopsis, 3D scene reconstruction, making panoramic images, structure from motion

Multi-View Stereo

[Fitzgibbon and Zisserman, 1998]
Modeling Camera Projection

- The coordinate system
  - Pinhole camera model as an approximation
  - Put the pinhole (aka optical center, center of projection) at the origin
  - Put the image plane (projection plane) \textit{in front} of the optical center

Camera Parameters

- A camera is described by several parameters
  - Translation $T$ of the optical center from the origin of world coords
  - Rotation $R$ of the image plane
  - focal length $f$, principle point $(x'c, y'c)$, pixel size $(s_x, s_y)$
  - blue parameters are called “extrinsics,” red are “intrinsics”

- Projection equation

$$\Pi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - The projection matrix models the cumulative effect of all parameters
  - Useful to decompose into a series of operations

Epipolar Geometry

- Co-Planarity Constraint: $C, C', x, x'$ and $X$ are co-planar

Epipolar Geometry

- What if only $C, C'$, and $x$ are known?
  - Answer: $x'$ constrained to lie on epipolar line $l'$
Epipolar Geometry

All points on $\pi$ project onto **epipolar lines** $l$ and $l'$

Epipolar Geometry

Family of planes $\pi$ and lines $l$ and $l'$ intersect in **epipoles** $e$ and $e'$

**Epipolar Geometry**

- **Correspondence geometry**: Given an image point $x$ in the first view, how does this constrain the position of the corresponding point $x'$ in the second image?
- **Epipolar geometry constrains search for** $x'$ **from 2D to 1D**

epipoles $e, e'$
- intersection of baseline with image plane
- projection of optical center in other image
- vanishing point of camera motion direction

an epipolar plane = plane containing baseline (1D family)
an epipolar line = intersection of epipolar plane with image (always come in corresponding pairs)
Computing the Epipolar Geometry

- Given a scene point, define it in terms of 2 vectors wrt left and right cameras
  
  \[ P' = RP + t \]
  \[ p = MP \text{ with } M = [I \mid 0] \]
  \[ p' = M'P' \text{ with } M' = [R \mid t] \]
- Co-planarity constraint: \( P, P' \) and \( t \) are co-planar, so their mixed product = 0
Calibrated Camera

\[ p'[t \times (RP)] = 0 \quad \text{with} \quad \begin{cases} p = (u, v, 1)^T \\ p' = (u', v', 1)^T \end{cases} \]

\[ E = \text{Essential matrix} \quad p'Ep = 0 \quad \text{with} \quad E = [t_x]R = SR \]

Uncalibrated Camera

\[ p' = M_{\text{int}}^{-1} p \quad \text{and} \quad p' = M_{\text{int}}^{-1} p' \]

\[ p'EP = 0 \quad \text{with} \quad F = M_{\text{int}}^{-T} EM_{\text{int}}^{-1} \]

\[ F = \text{Fundamental matrix} \]

Properties of Fundamental and Essential matrices

- Matrix is 3 x 3
- Transpose: If F is fundamental matrix of cameras \((P, P')\), then \(F^T\) is fundamental matrix of camera \((P', P)\)
- Epipolar lines: Think of \(p\) and \(p'\) as points in the projective plane. Then \(Fp\) is projective line in the right image. That is \(l = Fp\quad l = F'p'\)
- Epipoles: Since for any \(p\) the epipolar line \(l'_p = Fp\) contains the epipole \(e\), so \((e^T F)p = 0\) for all \(p\). Thus \(e^T F = 0\) and \(F e = 0\)

Essential matrix, E

- Encodes information on the extrinsic camera parameters only
- E is of rank 2, since S has rank 2 (and R has full rank)
- Has only 5 degrees of freedom: 3 for rotation, 2 for translation (\(t\) can only be recovered up to a scale factor, meaning only the direction of translation can be obtained)
Fundamental matrix, $F$

- $F$ is the unique $3 \times 3$ rank 2 matrix that satisfies $x^T F x = 0$ for all $x \rightarrow x'$
- Encodes information on the intrinsic and extrinsic camera parameters
- $F$ is of rank 2, since $S$ has rank 2 ($R$ and $M$ and $M'$ have full rank)
- Has 7 degrees of freedom (There are 9 elements, but scaling is not significant and det $F = 0$)

### Computing Fundamental Matrix from Point Correspondences

- The fundamental matrix is defined by the equation $x_i^T F x_i = 0$ for any pair of corresponding points $x_i$ and $x'_i$ in the 2 images
- The equation for a pair of points $(x, y, 1)$ and $(x', y', 1)$ is:
  
  $x' x f_{31} + x' y f_{32} + x' f_{33} + y' x f_{31} + y' y f_{32} + y' f_{33} = 0$

- For $n$ point matches:

  $\begin{bmatrix} x_i \, y_i \, 1 \\ x'_i \, y'_i \, 1 \end{bmatrix}^T \begin{bmatrix} x_i \, y_i \, 1 \\ x'_i \, y'_i \, 1 \end{bmatrix} = 0$

### Normalized 8-Point Algorithm [Hartley, 1995]

1. Normalization: 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: $q_i = T p_i$ and $q'_j = T p'_j$
2. Solve linear system to compute $F$ from conjugate pairs $q_i$ and $q'_j$
3. Enforce rank-2 constraint by finding closest singular $F'$ to $F$
4. Denormalization: Output $F = T^T F' T$
Projective Reconstruction Theorem

Assume we determine matching points $x_i$ and $x'_i$. Then we can compute a unique fundamental matrix $F$.

The camera matrices $M, M'$ cannot be recovered uniquely.

Thus the reconstruction $(X_i)$ is not unique.

There exists a projective transformation $H$ such that

$$X_{2,i} = H X_{1,i}, \quad M_2 = M_1 H^{-1}, \quad M'_2 = M'_1 H^{-1}$$

Projective Reconstruction Theorem (Consequences)

- We can compute a projective reconstruction of a scene from 2 views based on image correspondences alone.
- We don’t have to know anything about the calibration or poses of the cameras.
- The true reconstruction is within a projective transformation $H$ of the projective reconstruction: $X_{2i} = H X_{1i}$.
Stratified Reconstruction

- Begin with a projective reconstruction
- Refine it to an affine reconstruction
  - Parallel lines are parallel; ratios along parallel lines are correct
  - Reconstructed scene is then an affine transformation of the actual scene
- Then refine it to a metric reconstruction
  - Angles and ratios are correct
  - Reconstructed scene is then a scaled version of actual scene

3D Scene Reconstruction: Basic Stereo Algorithm

For each epipolar line
  - For each pixel in the left image
    - compare with every pixel on same epipolar line in right image
    - pick pixel with minimum match cost
Improvement: match *windows*

Finding correspondences is relatively easy when baseline is small

State of the Art in 3D Reconstruction
(Structure from Motion)


Wide Baseline Matching

- Camera networks usually have cameras that are far apart, making correspondence problem very difficult
- Feature-based approach: Detect feature points in both images

Matching with Features

- Detect feature points in both images
- Find corresponding pairs

Matching with Features

Problem 1:
- Detect the same point *independently* in both images

We need a repeatable detector

Problem 2:
- For each point correctly recognize the corresponding one

We need a reliable and distinctive descriptor
Properties of an Ideal Feature

- **Local**: features are local, so robust to occlusion and clutter (no prior segmentation)
- **Invariant** (or covariant) to many kinds of geometric and photometric transformations
- **Robust**: noise, blur, discretization, compression, etc. do not have a big impact on the feature
- **Distinctive**: individual features can be matched to a large database of objects
- **Quantity**: many features can be generated for even small objects
- **Accurate**: precise localization
- **Efficient**: close to real-time performance

Applications

- **Wide baseline matching without scene segmentation**

Applications

- **Recognition of specific objects**
  - Rothganger et al. ‘03
  - Lowe et al. ‘02
  - Ferrari et al. ‘04

Applications

- **Object class recognition**
  - Bag-of-Word models
  - Constellation (graph) models
Recognition of object classes

- Bag-of-visual-words image representation:

Recent Work on Feature Detectors

- Hessian
- Harris
- Lowe: SIFT (DoG)
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- Others

Harris “Corner”/Interest Point Detector


Basic Idea

- We should easily recognize the point by looking through a small window
- Shifting a window in any direction should give a large change in response
Harris Detector: Basic Idea

“flat” region: 
no change in 
all directions

“edge”: 
no change along 
the edge direction

“corner”: 
significant change 
in all directions

Harris Detector: Mathematics

Change of intensity for the shift \([u,v]\):

\[
E(u,v) = \sum_{x,y} w(x,y) \left[ I(x+u, y+v) - I(x, y) \right]^2
\]

Window function \(W(x,y) = \)

1 in window, 0 outside

Gaussian

Harris Detector: Mathematics

Expanding \(E(u,v)\) in a 2nd order Taylor series, we have, for small shifts, \([u,v]\), a bilinear approximation:

\[
E(u,v) \equiv \begin{bmatrix} u \\ v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}
\]

where \(M\) is a 2 \(\times\) 2 matrix computed from image derivatives:

\[
M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}
\]

Harris Detector: Mathematics

Intensity change in shifting window: eigenvalue analysis

\[
E(u,v) \equiv \begin{bmatrix} u \\ v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}
\]

\(\lambda_1, \lambda_2\) – eigenvalues of \(M\)

Ellipse \(E(u,v) = \text{const}\)
Selecting Good Features

- \( \lambda_1 \) and \( \lambda_2 \) both large

Selecting Good Features

- Large \( \lambda_1 \), small \( \lambda_2 \)

Selecting Good Features

- Small \( \lambda_1 \), small \( \lambda_2 \)

Harris Detector: Mathematics

Classification of image points using eigenvalues of \( M \):

- \( \lambda_1 \) and \( \lambda_2 \) are small; \( E \) is almost constant in all directions

- \( \lambda_1 \) and \( \lambda_2 \) both large, \( \lambda_1 \approx \lambda_2 \); \( E \) increases in all directions

- \( \lambda_1 \gg \lambda_2 \); “Corner”

- \( \lambda_2 \gg \lambda_1 \); “Edge”

- “Flat” region
Harris Detector: Mathematics

Measure of corner response:

\[ R = \det M - k \left( \text{trace } M \right)^2 \]

\[ \det M = \lambda_1 \lambda_2 \]
\[ \text{trace } M = \lambda_1 + \lambda_2 \]

\( k \) is an empirically-determined constant; e.g., \( k = 0.05 \)

Harris Detector: Mathematics

- \( R \) depends only on eigenvalues of \( M \)
- \( R \) is large for a corner
- \( R \) is negative with large magnitude for an edge
- \( |R| \) is small for a flat region

Harris Detector: Example

Algorithm:

- Find points with large corner response function \( R \) \( (R > \text{threshold}) \)
- Take the points of local maxima of \( R \) (for localization)
Harris Detector: Example

Compute corner response $R$

Find points with large corner response: $R > \text{threshold}$

Take only the points of local maxima of $R$
Harris Detector: Example

Interest points extracted with Harris (~ 500 points)

Harris Detector: Some Properties

- Rotation invariance
  
  Ellipse rotates but its shape (i.e., eigenvalues) remains the same

  *Corner response $R$ is invariant to image rotation*

Harris Detector Properties: Scale Changes

- But not invariant to *image scale*

  Fine scale: All points will be classified as edges
  
  Coarse scale: Corner
Harris Detector: Some Properties

- Quality of Harris detector for different scale changes

Repeatability rate:
\[
\frac{\text{# correct correspondences}}{\text{# possible correspondences}}
\]


Scale Invariant Detection

- Consider regions (e.g., circles) of different sizes around a point
- Regions of corresponding sizes will look the same in both images

Scale Invariant Detection

- Problem: How do we choose corresponding circles \textit{independently} in each image?

Solution:
- Design a function on the region (circle) that is “scale invariant,” i.e., the same for corresponding regions, even if they are at different scales.

Example: Average intensity. For corresponding regions (even of different sizes) it will be the same

- For a point in one image, we can consider it as a function of region size (circle radius)
Scale Invariant Detection

- Common approach: Take a local maximum of this function

Observation: Region size, for which the maximum is achieved, should be invariant to image scale

Important: This scale invariant region size is found in each image independently!

Automatic Scale Selection

Lindeberg et al., 1996

Automatic Scale Selection

Function responses for increasing scale

Scale trace (signature)
Automatic Scale Selection
Function responses for increasing scale
Scale trace (signature)

\[ f(x_1, x_2) \]

Automatic Scale Selection
Function responses for increasing scale
Scale trace (signature)

\[ f(x_1, x_2) \]

Automatic Scale Selection
Function responses for increasing scale
Scale trace (signature)

\[ f(x_1, x_2) \]

Automatic Scale Selection
Function responses for increasing scale
Scale trace (signature)

\[ f(x_1, x_2) \]
Automatic Scale Selection

Function responses for increasing scale
Scale trace (signature)
Automatic Scale Selection

Function responses for increasing scale
Scale trace (signature)

Normalize: rescale to fixed size
Scale Invariant Detection

- A “good” function for scale detection has one stable sharp peak
  ![Graph showing a good function for scale detection with stable sharp peaks.]

- For many images: a good function would be a one that responds to contrast (sharp local intensity change)

Lowe’s SIFT (DoG) Detector

- Difference-of-Gaussian (DoG) as approximation of the Laplacian-of-Gaussian (LoG)
  ![Diagram showing the relationship between DoG and LoG.]

Scale Invariant Detectors

- Harris-Laplacian
  Find local maxima of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale

- SIFT keypoints
  Find local extrema of:
  - Difference of Gaussians in space and scale

Lowe’s SIFT (DoG) Detector

- Difference-of-Gaussians as approximation of the Laplacian-of-Gaussian
  ![Diagram showing the relationship between DoG and LoG sampling with step σ = 2.]
Previously we considered:
Similitude transform (rotation + uniform scale)

Now we go on to:
Affine transform (rotation + scale + skew)

Initialization with Harris Laplace
Estimate shape based on second moment matrix
Use normalization / deskewing
Iterative algorithm
Affine Invariant Detection:
Mikolajczyk’s Harris-Affine Detector
1. Detect multi-scale Harris points
2. Automatically select the scales
3. Adapt affine shape based on second order moment matrix
4. Refine point location

Affine Invariant Detection:
Tuyltelaars’s Intensity-based Regions
1. Select intensity extrema
2. Consider intensity profile along rays from each extremum point
3. Select maximum of invariant function \( f(t) \) along each ray
4. Connect all local maxima
5. Compute geometric moments of orders up to 2 for this region
6. Fit an ellipse

\[
\frac{1}{\max(\int_0^d |f(t)| dt, d)}
\]

\[
f(t) = \frac{\text{abs}(I_0 - I)}{\text{abs}(I_0 - I) dt}
\]
Quantitative Comparisons of Feature Detectors

- Scale and affine invariant interest point detectors, K. Mikolajczyk and C. Schmid, *Int. J. Computer Vision* 60(1), 2004
- A survey on local invariant features, T. Tuytelaars and K. Mikolajczyk
- Evaluation on 3D objects (Moreels & Perona, ICCV, 2005)
- Evaluation on 3D objects (Fraundorfer & Bischof, ICCV, 2005)

Evaluation Criterion: Repeatability

- Repeatability rate: percentage of correctly corresponding points

\[
\text{repeatability} = \frac{\#\text{correspondences}}{\#\text{detected}} \times 100\%
\]

Repeatability

Feature Point Descriptors

- We know how to detect points
- Next question: How to match them?

Point descriptor should be:
1. Invariant
2. Distinctive
Descriptors Invariant to Rotation

- Find local orientation
  Dominant direction of gradient:
- Compute description relative to this orientation

SIFT Keypoint Feature Descriptor

- Descriptor overview:
  - Compute gradient orientation histograms on 4 x 4 neighborhoods, relative to the keypoint orientation using thresholded image gradients from Gaussian pyramid level at keypoint’s scale
  - Quantize orientations to 8 values
  - 2 x 2 array of histograms
  - SIFT feature vector of length 4 x 4 x 8 = 128 values for each keypoint
  - Normalize the descriptor to make it invariant to intensity change

SIFT – Scale Invariant Feature Transform

- Empirically found to show very good performance, invariant to image rotation, scale, intensity change, and to moderate affine transformations

1 D.Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” IJCV 2004
References on Feature Descriptors


---

**Feature Detection and Description Summary**

- Stable (repeatable) feature points can be detected regardless of image changes
  - Scale: search for correct scale as maximum of an appropriate function
  - Affine: approximate regions with ellipses
- Invariant and distinctive descriptors can be computed
  - Invariant moments
  - Normalizing with respect to scale and affine transformation
- Limited affine invariance for large viewpoint changes; no projective invariant methods
- Incorporate color, texture into descriptor