**INFO0013 Computer Vision**

Finding Correspondences Automatically

Justus H. Piater

---

**Where are we at?**

Given a pair of images and a list of point correspondences, we know how to recover cameras and scene structure so we can render an artificial object into it.

Now we’d like to generate the point correspondences automatically so we can **fully automate** the recovery of cameras and scene structure.

How might we go about this?

---

**Estimating Derivatives**

**Gradient Detection by Finite Differences**

Gradients have a **magnitude** and an **orientation**.

To estimate horizontal gradients:

- forward difference: kernel [1 -1]
- central difference: kernel [1 0 -1]
Gradient Estimation and Noise

Finite-difference filters respond strongly to \textit{noise}.

- Because they detect \textit{change}, and noise incurs change.
- The larger the noise, the stronger the response.

\textbf{Solution:} Make pixels look more like their neighbors (smooth).

The Response of a Linear Filter to Noise

Stationary, independent, additive, zero-mean, Gaussian noise:

\[
\sigma^2(\sum X) = \sum \sigma^2(X)
\]

\[
\sigma^2(kX) = k^2 \sigma^2(X)
\]

Filtering noise with a kernel \( K \):

\[
\sigma^2(K(X)) = \sigma^2(\sum_{i} K_i X_i) = \sum_i K_i^2 \sigma^2(X_i)
\]

Thus, the variance of the response of a 2D kernel \( K \) to noise of variance \( \sigma^2 \) is \( \sigma^2 \sum_{i,j} K_{ij}^2 \).

Effect of Noise on Gradient Estimation

[Figure from Forsyth/Ponce lecture notes]

Noise and Higher-Order Derivatives

This effect gets worse with higher-order derivatives:

- Higher-order derivatives are the derivatives of derivatives.
- The coefficients of higher-order derivatives increase in magnitude (see for yourselves!).
- The Fourier transform of \( \frac{\partial^2}{\partial x^2} f(x, y) \) is \( uF(u, v) \). Thus, \textit{differentiation emphasizes high frequencies}. 

Estimating Derivatives
Why Smoothing Helps

- Large derivatives incurred by noise tend to be local.
- Smoothing makes pixels look more than their neighbors, reducing local variance.
- Smoothing suppresses high frequencies (as seen).

Gaussian Derivatives

Since convolution is associative, we prefer to convolve with a derivative of a Gaussian kernel instead of first smoothing and then taking finite differences.

Gradient magnitude and orientation as usual:

\[ |\nabla| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2} \]
\[ \tan \theta_{\text{grad}} = \frac{\frac{\partial g}{\partial y}}{\frac{\partial g}{\partial x}} \]

Gradient Magnitude and Orientation

Detect The Gradient At Which Scale?

(Figure from Forsyth/Ponce lecture notes)
Interest Points and Local Appearance

The Harris Interest Point Detector (Harris and Stephens 1988)

Observation: Interesting image structure breaks local autocorrelation.

\[ M = G(\sigma)^\ast \nabla l \nabla l^\top = G(\sigma)^\ast \begin{bmatrix} l^2_x & l_x l_y \\ l_y l_x & l^2_y \end{bmatrix} \]

- quadratic terms of a Taylor expansion of the local autocorrelation function
- smooth, isotropic (Gaussian) window

Its eigenvalues are proportional to the principal curvatures of the local autocorrelation function:
- One large eigenvalue – edge
- Two large eigenvalues – corner

Looking for local maxima of the eigenvalues leads to one of the best known interest point detectors.

What points are good for matching?

They must be:
- precisely localizable
- distinctive in their appearance

The Harris Interest Point Detector (Harris and Stephens 1988)

Note

It is rotationally invariant. [Illustration from Schmid and Mohr 1997]
**Edges, Corners, Flat Regions**

![Graph showing edges, corners, and flat regions.](image)

(Figure by Sébastien Pérard)

**Implementing the Harris Corner Detector**

- \( I_x = \frac{\partial}{\partial x} G(\sigma) * I \), etc. The same \( \sigma \) can be used for smoothing the three images representing the elements of \( M \).
- The following identities allow us to avoid computing the eigenvalues \( \lambda_1 \) and \( \lambda_2 \) explicitly:
  
  \[
  \lambda_1 + \lambda_2 = \text{Tr}(M) = M_{11} + M_{22}
  \]
  \[
  \lambda_1 \lambda_2 = \text{Det}(M) = M_{11}M_{22} - M_{12}M_{21}
  \]

  We now measure the corner response by
  
  \[
  R = \text{Det}(M) - k \text{Tr}^2(M)
  \]
- Corners are all local maxima of \( R \) with \( R > 0 \) and \( \text{Tr} > t \).
- A typical choice is \( k = 0.04 \), and \( t \) is chosen to yield the desired number of interest points.

**Other Methods for Interest Point Detection**

**Blob Detection**: Look for local maxima of the Laplacian:

\[
 b = I_{xx} + I_{yy}
\]

where \( I_{xx} = \frac{\partial^2}{\partial x^2} G * I \).

**Other methods:**

- several variants based on the autocorrelation matrix
- Harris-Laplace, Hessian-Laplace (scale invariant)  
  - [Mikolajczyk and Schmid 2004]
- Affine-invariant region detectors  
  - [Mikolajczyk et al. 2005]

**Characterizing and Comparing Local Appearance**

Now that we have localized interest points, we would like to match them.

We might simply compare local pixel neighborhoods \( w \) and \( w' \) by their sum of squared differences

\[
 \text{SSD} = (w - w')^T (w - w') \]

or normalized cross correlation

\[
 \frac{d^T d'}{|d| |d'|}, \quad \text{where} \quad d = w - \bar{w}.
\]
Local Jet

Gray level neighborhood by Taylor expansion:

\[ l(x + \xi, y + \eta) = \sum_{n=0}^{\infty} \frac{1}{n!} \left( \frac{\partial^n l}{\partial \xi^n} \right)_{\xi=\eta=0} \]

\[ = \sum_{m=0}^{n} \sum_{k=0}^{m} \binom{m}{k} a_{x,y} \frac{\partial^m l}{\partial \xi^m} \frac{\partial^k l}{\partial \eta^k} \]

\[ a_{x,y} = \frac{\partial^m l}{\partial \xi^m \partial \eta^m} G * l(x, y) \]

where \( a_{x,y} \) has \( n \) subscripts \( x \) and \( m \) subscripts \( y \).

The local N-jet is represented by the coefficients \( a_{x,y} \) up to order \( N = n + m \).

Differential Invariants

The local jet is not invariant to rotation in the image plane.

Some differential invariants computed from the local jet:

\[ I_x^2 + I_y^2 \]

\[ I_x I_{xx} + 2I_{xy} I_{xy} + I_{yy} \]

\[ I_{xx} - 2I_{xy} I_{xy} + I_{yy} \]

\[ I_x + I_{yy} \]

\[ I_x^2 + 2I_{xy}^2 + I_y^2 \]

Now we have gained rotational invariance, but lost all information about orientation.

Steerable Filters [Freeman and Adelson 1991]

To obtain the response of a filter at orientation \( \theta \):

- Create a corresponding kernel and convolve it with the image.
- Better: Synthesize the response from responses to a finite set of basis filters.

A filter class is steerable if any given filter can be synthesized as a linear combination of basis filters.

First derivatives of Gaussians are steerable:

\[ G^0 = G^0 \cos \theta + G^0 \sin \theta \]

where \( G^0 = \frac{\partial}{\partial x} G \) and \( G^0 = \frac{\partial}{\partial y} G \).

Steering \( d \)th-order Derivatives of Gaussians

\( d + 1 \) basis orientations, equally spaced between 0 and \( \pi \):

\[ \theta_{k,d} = \frac{k \pi}{d + 1}, \quad k = 0, \ldots, d. \]

To synthesize a filter at orientation \( \theta \):

\[ G^\theta = \sum_{k=0}^{d} c_{k} G^\theta_{x} + \sum_{k=0}^{d} c_{k} G^\theta_{y} \]

\[ c_{k,1} = \cos (\theta - \theta_{k,1}) \]

\[ c_{k,2} = \frac{1}{2} (1 + \cos (2 \theta - \theta_{k,2})) \]

\[ c_{k,3} = \frac{1}{2} (\cos (\theta - \theta_{k,3}) + \cos (3 \theta - \theta_{k,3})) \]
Basis Filters For $G_1$ and $G_2$

Other Methods

Local Appearance:
- Primitives: color, gradients, ...
- Vectors of primitives
- Histograms of primitives
There are countless variants.

The highly popular Scale Invariant Feature Transform (SIFT) (Lowe 2004) addresses both scale-invariant interest point localization and rotation-invariant appearance characterization. There are some interesting variants (Mikolajczyk and Schmid 2005).

This is still a topic of active research.

RANSAC - RAndom SAmple Consensus

1. Repeat $k$ times:
   a. Draw a random sample of $n$ data points.
   b. Determine the model parameters using these points.
   c. For each data point outside the sample:
      - If its distance to the model is less than $d$, then label it as a “good” point.
   d. If there are at least $t$ “good” data points, then declare this model “good” and quit this loop.
2. Refit the model with the most “good” points using all of them, including the $n$ sample points.

[Fischler and Bolles 1981]
Issues

- Each sample comprises exactly $n$ data points for an $n$-parameter model.
- Measure of the distance of a data point to the model.
- Choice of the three parameters $k$, $d$, and $t$.

How Many Samples?

Given an $n$-parameter model and a fraction $w$ of inliers, we need $k$ samples.

**One Approach:**

\[ E[k] = P(\text{"good" in 1 draw}) + P(\text{"good" in 2 draws}) + \ldots \]

\[ = w^n + 2w^n(1 - w^n) + 3w^n(1 - w^n)^2 + \ldots \]

\[ = w^{-\gamma} \]

To increase confidence, add a few standard deviations $\sigma(k)$ to $k$:

\[ \sigma(k) = \frac{\sqrt{1 - w^n}}{w^n} \]

How Many Samples? (Continued)

**Another Approach:** From the probability $z$ of seeing only samples that contain at least one outlier, solve for $k$:

\[ z = (1 - w^n)^k \]

\[ k = \frac{\log z}{\log(1 - w^n)} \]

How Close Must a Point Be to Be “Good”?

RANSAC is quite insensitive to the choice of $d$. Choose by eyeballing, trial-and-error, ...
How Many “Good” Points for a “Good” Fit?
A reasonable rule of thumb is to require at least as many “good” points as we would expect from the proportion \( w \) of inliers present in the data:
\[
t = wN
\]
where \( N \) is the total number of data points.

Estimating the Proportion of Inliers

Estimate \( w \) by counting the number of “good” points:

1. Begin with \( w = 0 \).
2. After each RANSAC iteration:
   \[
   w = \frac{\text{number of “good” points}}{N}
   \]
   and recompute \( k \).

RANSAC for Automatic Computation of a Fundamental Matrix

Putative Correspondences: Use Harris to detect 7 interest points in each image; match (regionally) based on appearance.

Putative \( F \): Use the minimal algorithm. If there are 3 solutions, try them all.

Distance Measure: Better than the algebraic distance \( x^T F x \) is the Sampson distance
\[
\frac{(x^T F x)^2}{(F x_1^T)^2 + (F x_2^T)^2 + (F^T x_1^T)^2 + (F^T x_2^T)^2}
\]
(not discussed in class).

RANSAC for Automatic Computation of a Fundamental Matrix (Continued)

Finally: Re-estimate \( F \) from all inliers of RANSAC result by optimizing the same error function as used as the distance measure. We may also add new correspondences by searching in a small strip around the epipolar lines.
~500 Interest Points Per Image

[Figure from Hartley and Zisserman 2000]

Putative matches: 99 inliers, 89 outliers

[Figure from Hartley and Zisserman 2000]

All 188 putative matches, 157 revised matches after MLE

[Figure from Hartley and Zisserman 2000]

Conclusions
Conclusions

We now know how to compute a fundamental matrix fully automatically from a stereo pair of images:
1. Extract interest points from both images.
2. Determine their appearance vectors.
3. Compute a set of putative correspondences based on similarity of appearance (and proximity, ...).
4. Use RANSAC to compute $F$.

Reminder: Given $F$, how do we insert a graphical object into both images?

The next step: More than two images.

References


