Final Project

Interest Points and Image Matching

Jill Perreault & Céline Bensoussan

Harris Detector

Mathematics

The Harris Corner Detector is based on the local auto-correlation function of a signal. The local auto-correlation function measures the local changes of the signal with patches shifted by a small amount in different directions *(Koutsourakis)*.

Given a shift (x, y) and a point the auto-correlation function is defined as:

$$c(x, y) = \sum_W [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \qquad (1)$$

where $I(x_i, y_i)$ denotes the image function and (x_i, y_i) are the points in the window W centered on (x, y). A window or a Gaussian can be used *(Derpanis)*.

If the shift is small, the shifted image can be approximated by a Taylor expansion truncated to the first order terms *(Koutsourakis)*.

$$I(x_i + \Delta x, y_i + \Delta y) \approx \left[I(x_i, y_i) + \left[(x_i, y_i) I_y(x_i, y_i) \right] \right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \qquad (2)$$

where $I_x(x_i, y_i)$ and $I_y(x_i, y_i)$ denote the partial derivatives in x and y respectively.

Substituting Eq. (2) in Eq. (1) gives the following:

$$\begin{aligned} c(x, y) &= [\Delta x \quad \Delta y] \begin{bmatrix} \sum_W \left(I_x(x_i, y_i) \right)^2 & \sum_W \left(I_x(x_i, y_i) I_y(x_i, y_i) \right) \\ \sum_W \left(I_x(x_i, y_i) I_y(x_i, y_i) \right) & \sum_W \left(I_y(x_i, y_i) \right)^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= [\Delta x \quad \Delta y] c(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \end{aligned}$$

with $c(x, y)$ the auto-correlation matrix, or Harris matrix, which captures the intensity structure of the local neighborhood.

Let λ_1 and λ_2 be the eigenvalues of $c(x, y)$, then there are 3 cases to consider:

1. Both eigenvalues are high: interest point (corner)
2. One eigenvalue is high: contour (edge)
3. Both eigenvalues are low: uniform region (constant intensity) *(Derpanis)*.

The Harris corner detector gives a mathematical approach for determining which case holds.

Harris noted that the exact computation of the eigenvalues is computationally expensive and instead suggested:

$$H_{Corner\ Response} = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(c) - k \cdot trace(c)^2$$

where k is a constant determined empirical. Values of k in the range 0.04 - 0.15 can be found in literature *(Wikipedia)*. $k = 0.04$ is usually used *(Koutsourakis)*.

Thus we can consider that:

- H is large for a corner
- H is negative with large magnitude for an edge
- $|H|$ is small for a uniform region

A point is considered as a corner or interest point if $H \geq T$ where $T > 0$ is a certain fixed threshold predefined by the user *(Koutsourakis)*.

Algorithm

1. Compute the partial derivatives x and y of the image: I_x and I_y
2. For each pixel compute the Harris matrix.
3. Create the corner map $c(x, y)$, by computing the Harris Corner Response for each pixel.
4. Threshold the corner map by setting all pixels below T to 0.
5. Perform non-maximal suppression to find local maxima.

Image Matching: SIFT and SSD

The SIFT (Scale-Invariant Feature Transform) is an image matching algorithm first developed by David Lowe in 1999. It consists of calculating what are called "SIFT descriptors". Those descriptors contain information about the neighborhood of an interest point in the image, therefore, when comparing to similar points on different images, they would match.

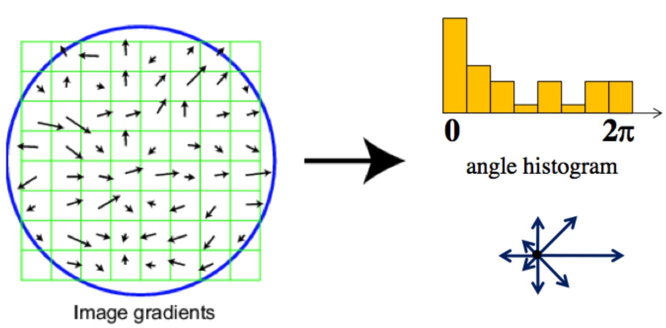
It can be done in 3 different ways:

- Compare all the points in the first image with all the points in the second image (very computationally expensive, $\theta(n^2)$).
- Compare the interest points of image 1 that are found using the Harris Corner Detector with all the points in image 2 (still $\theta(n^2)$).
- Compare the interest points of image 1 and the interest points of image 2 (this is what our algorithm is doing)

The SSD (Sum of squared differences) is used to compare the descriptors of two different images: the smaller the SSD, the more similar the images are.

Algorithm

1. Calculate the direction angle of the gradient (orientation) around each interest point in image 1 and image 2.
2. Divide each neighborhood in 4x4 windows and build an angle histogram for each of these windows in the neighborhood. These histograms are called the descriptors.



3. Calculate the SSD of each pair of points using the descriptors.

The Bensoussan & Perreault Implementation

Harris Detector

We chose to compute the partial derivatives of the image instead of the derivatives of the Gaussian. To do so, we used the function derivative7 by Peter Kovesi. His function "computes 1st and 2nd derivatives of an image using the 7-tap coefficients given by Farid and Simoncelli." Peter states "the results are significantly more accurate than Matlab's gradient function on edges that are at angles other than vertical or horizontal. This in turn improves gradient orientation estimation enormously." This function and its license can be found in the Borrowed Functions directory of our submission.

Before computing its derivatives, the image is padded with its reflection in order to avoid detecting its corners as interest points.

To avoid setting a parameter k, Nobel's corner response is used:

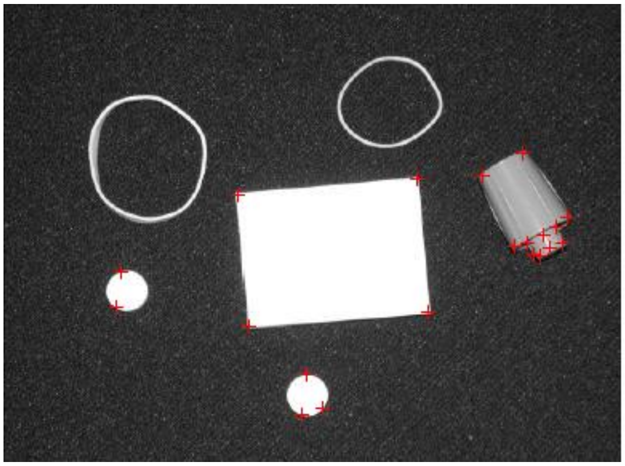
$$N_{Corner\ Response} = 2\ \frac{\det(c)}{trace(c) + \epsilon}$$

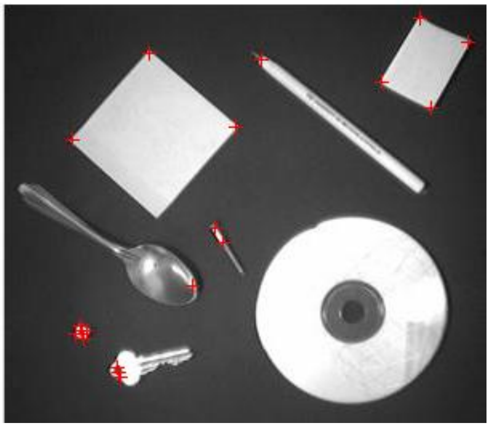
Nobel's corner measure "amounts to the harmonic mean of the eigenvalues" *(Wikipedia)*.

The algorithm originally states to threshold the corner map prior to performing non-maximal suppression. Our function needs to find n corners, thus non-maximal suppression is performed prior to thresholding. The threshold, in our case, is equal to the nth best corner response.

Examples of Harris Corner Detection:

n = 20





Céline Bensoussan
Jill Perreault

Image Matching

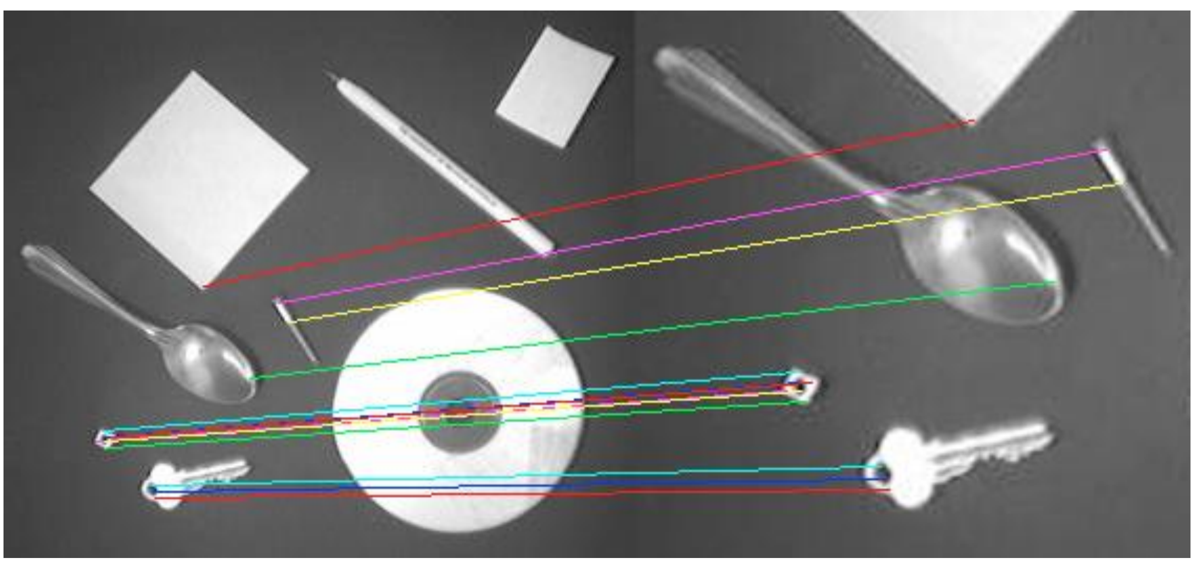
We implemented the SIFT and the SSD from scratch.

The image matching is only done from the interest points of first image to the interest points of the second image.

For each interest point, the neighborhood taken in consideration is a 16x16 window. This window is then divided in 4x4 cells; therefore there are 16 cells per window. We calculate the orientation of the gradient for each pixel and build one descriptor for each 4x4 cell (16 descriptors per window).
To calculate the orientation of the gradient, we used the function implemented for the second and third assignment.

The way it is done:

- We find the best match (point with smallest SSD) of a point p_a in image 1 to a point p_b in image 2.
- We then check if the best match for p_b is p_a
- If yes, we match them
- If not, we find the second best match of p_a, which would be p_c
- We then check if the best match for p_c is p_a. If yes, we match them and if not, we assume p_a has no match in image 1.



Software Tutorial

The main program is the final_p.m file. For final_p.m to run, you must set your path to include all of our project's directories.

The only part that the user needs to change is in the initialization part.

- im1 is the first image to analyze.
- im2 is the second image to analyze.
- sigma used for the Gaussian filter and for the gradient orientation.
- n is the number of interest points that need to be found in im1 and im2
- disp should be set to 1 if you want to display the n corner points overlaid on the original images (red crossed on image).

The functions related to Harris Corner Detection are:

- harrisCornerDetector.m
- derivative7.m (Borrowed Functions)
- gaussFilter.m

The functions related to Image Matching are:

- imageMatching.m
- gradientOrientation.m
- getDescriptors.m
- SSD.m
- getCorrelation.m
- match_plot.m (Borrowed Functions)

Works Cited

Derpanis, Konstantinos G.. “The Harris Corner Detector”. October 27, 2004. https://docs.google.com/viewer?a=v&q=cache:J9wLDSL1CQAJ:windage.googlecode.com/svn/trunk/Mindmap/Tracking/Papers/%255B2004%255D%2520The%2520Harris%2520Corner%2520Detector.pdf+&hl=en&gl=ca&pid=bl&srcid=ADGEEShO7h1TepbGst6S-j6xSaYrIQ3D2Vos4E1oWV_w7Vpk5IDTMmWV2Ijchp04OPujse1tJ1_adUSHYF9RNLTsBTF4yPXt_tBbbFwwvS1hbms7ZW4bBDx7mXa956I191DcnW4hmVFQ&sig=AHIEtbT8z9czQR8ptLMEw4qyO_nk_Y4e5w .

Koutsourakis, Simon, and Wang. “TP5: Interest Points and Object Recognition”. 2007. http://vision.mas.ecp.fr/Personnel/simon/files/Documents/Teaching/VisionLabs/lab5.pdf

Wikipedia. Corner Detection. http://en.wikipedia.org/wiki/Corner_detection#The_Harris_.26_Stephens_.2F_Plessey_.2F_Shi-Tomasi_corner_detection_algorithm

CSE Washington University. “Feature of Image Matching” http://www.cs.washington.edu/education/courses/cse455/09wi/Lects/lect6.pdf