Interest Point Detection

RODNEY DOCKTER

MAY 2018
Last Week

- Hough Transform
- Circle Detection
- Otsu Method
Outline

• Template Matching
• Interest Point Detection
• Corner Detection
• Robust Features
Applications

• Comparison between 2 or more images
  • Image alignment
  • Image stitching
  • 3D reconstruction
  • Object recognition (matching)
  • Indexing and database retrieval
  • Object tracking
  • Robot navigation
Template Matching

- Elements to be matched are image patches of fixed size

Task: find the best (most similar) patch in a second image

Known as: template Matching Method, with option to use correlation as a metric
Template Matching

• Correlation

\[ f * h = \sum_{k} \sum_{l} f(k,l)h(i + k, j + l) \]

\[ f = \text{Image} \]
\[ h = \text{Kernel} \]
Template Matching

\[ f \ast h = \sum_k \sum_l f(k, l)h(i + k, j + l) \quad \text{Cross correlation} \]

\[ f \ast f = \sum_k \sum_l f(k, l)f(i + k, j + l) \quad \text{Auto correlation} \]

- Use Cross Correlation to find the template in an image
- Maximum indicates high similarity
Template Matching

• What do we do for different scales of the patch?
  • Compute multiple Templates at different sizes
  • Match for different scales of the template

• What can we do to identify the shape of the pattern at different brightness levels
  • Subtract the mean of the template

• Example in 1D

$$\begin{array}{ccccc}
3 & 1 & 1 & 4 & 5 \\
\end{array} \ast \begin{array}{ccccc}
3 & 1 & 1 & 4 & 5 \\
\end{array} = 9 + 1 + 1 + 16 + 20 = 47$$
Template Matching

• Not all Patches are created equal

This would be a good patch for matching, since it is distinctive (there is only one patch in the second frame that looks similar).
Template Matching

• Not all Patches are created equal

This would be a bad patch for matching, since it is not very distinctive (there are many similar patches in the second frame).

We need the template to contain robust feature descriptors for matching
Template Matching

• Local features associated with a significant change of an image property or several properties simultaneously (e.g., intensity, color, texture).
Template Matching

• Why extract interest points?
  • Corresponding points (or features) between images enable the estimation of parameters describing geometric transforms between the images.
Interest Points

• What if we don’t know the correspondences?
• Need to compare *feature descriptors* of local patches surrounding interest points
Interest Points

- Properties of good features
  - **Local**: features are local, robust to occlusion and clutter (no prior segmentation!).
  - **Accurate**: precise localization.

- **Invariant/ Covariant**
  - **Robust**: noise, blur, compression, etc. do not have a big impact on the feature.
  - **Distinctive**: individual features can be matched to a large database of objects.
  - **Efficient**: close to real-time performance.

- **Repeatable**
Interest Points

• A function $f(\ )$ is **invariant** under some transformation $T(\ )$ if its value does not change when the transformation is applied to its argument:

  \[
  \text{if } f(x) = y \text{ then } f(T(x)) = y
  \]

• A function $f(\ )$ is **covariant** when it commutes with the transformation $T(\ )$:

  \[
  \text{if } f(x) = y \text{ then } f(T(x)) = T(y)
  \]
Interest Points

• Features should be detected despite geometric or photometric changes in the image.
• Given two transformed versions of the same image, features should be detected in corresponding locations.
Interest Points

• Example: Panorama Stitching

• How do we combine these two images?
Interest Points

• Example: Panorama Stitching

Step 1: Extract features
Step 2: Match features
Interest Points

- Example: Panorama Stitching

Step 1: Extract features
Step 2: Match features
Step 3: Align images
Interest Points

• Example: Panorama Stitching

Use features with gradients in at least two (significantly) different orientations: Patches? Corners?
Interest Points

• What features should we use?

(Auto Correlation)
Interest Points

• Corners vs. Edges
  • **Corners** are locations where variations of intensity function \( f(x,y) \) in both \( X \) and \( y \) are high
    • Both partial derivatives \( (f_x \text{ and } f_y) \) are large
  • **Edges** are locations where the variation of \( f(x,y) \) in certain directions are high, while variations in the orthogonal direction are low
    • When edge is oriented along \( Y \), \( f_x \) is large, \( f_y \) small
Corner Detection

- Corners in images
- Corner is sharp turn of contour
- Corners are used in shape analysis and motion analysis
- They are dominant in human perception of 2D shapes

- Two types of corner detection:
  - In digital curves: assumes extracted contours
  - In digital images: does not assume extracted contours

- We only consider corner detection in grayscale images
Corner Detection

• The Aperture Problem

• Motion vectors are **ambiguous** at edges
  • Locally, normal component can only be determined
  • Tangential component cannot be determined

• Motion vectors are **unambiguous** at corners
Corner Detection

- Corner Detection
Corner Detection

- Corner Detection - Basic idea
  - We should easily recognize the point by looking through a small window
  - Shifting a window in any direction should give a small change in response
Corner Detection

- Corner Detection - Basic idea

  - "Flat" region: No change in all directions
  - "Edge" region: No change along the edge direction
  - "Corner" region: Significant change in all directions
Corner Detection

- Corner detection using intensity:
  - Image gradient has two or more dominant directions near a corner.
  - Shifting a window in any direction should give a large change in intensity.

“Flat” region: No change in all directions

“Edge” region: No change along the edge direction

“Corner” region: significant change in all directions
Corner Detection

- Corner detection using edge detection?
  - Edge detectors are not stable at corners
  - Gradient is ambiguous at corner tip
  - Discontinuity of gradient direction near corner
Corner Detection

• L- junction
• Y-junction
• T-junction
• Arrow- junction
• X- junction
Corner Detection

• Main steps to corner detection
  • For each pixel in the input image, the corner operator is applied to obtain a “cornerness” measure for this pixel
  • Threshold “cornerness” map to eliminate weak corners
  • Apply non-maximal suppression to eliminate points whose “cornerness” measure is not larger than the cornerness values of all points within a certain distance
Corner Detection

• Main steps to corner detection
Corner Detection

• Moravec Detector (1977)
  • Measure intensity variation at \((x,y)\) by shifting a small window (3X3 or 5X5) by one pixel in each of the eight principle directions (horizontally, vertically, and four diagonals)
Corner Detection

• Moravec Detector (1977)
  • Calculate intensity variation by taking the sum of squares of intensity differences of corresponding pixels in these two windows

\[
S_W(\Delta x, \Delta y) = \sum_{x_i \in W} \sum_{y_i \in W} (f(X_i, Y_i) - f(X_i - \Delta x, y_i - \Delta y))^2
\]

\[
= (A_1 - B_1)^2 + (A_2 - B_2)^2 + \cdots + (A_9 - B_9)^2
\]

8 Directions
\[\Delta x, \Delta y \in \{-1, 0, 1\}\]
\[S_W(-1, -1), S_W(-1, 0), \ldots, S_W(1, 1)\]
Corner Detection

- Moravec Detector (1977)
- The “cornerness” of a pixel is the minimum intensity variation found over the eight shift directions:

\[
\text{cornerness} (x, y) = \min\{S_w(-1, -1), S_w(-1, 0), \ldots S_w(1,1)\}
\]

Note response to isolated points!

Cornerness Map (normalized)
Corner Detection

- Moravec Detector (1977)
  - Using Non-maximal suppression will yield the final corners. Zero out all pixels that are not the maximum along the direction of the gradient (look at 1 pixel on each side)

![Corner Detection Diagram]

True Corner

Non-maximum (set to zero)
Corner Detection

• Moravec Detector (1977)

• Does a reasonable job of finding the majority of true corners

• Edge points not in one of the eight principle directions will be assigned as relatively large cornerness value
Corner Detection

- Moravec Detector (1977)
  - The response is anisotropic (directionally sensitive) as the intensity variation is only calculated at a discrete set of shifts (i.e. not rotationally invariant)
Corner Detection

• Mathematics!
• Change in appearance of window $w(x, y)$ for the shift $[u, v]$:

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

$I(x, y)$

$w(x, y)$

$E(u, v)$
Corner Detection

- Mathematics!
- Change in appearance of window $w(x, y)$ for the shift $[u, v]$:

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

- Mathematics!
- Change in intensity for the shift \([u, v]\):

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

Window function
Shifted intensity
Intensity

Window function \(w(x, y) =

1 \text{ in window, } 0 \text{ outside}

\text{ or }

Gaussian
Harris Detector

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

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

- Improves the Moravec operator by avoiding the use of discrete directions and discrete shifts.
- Uses a Gaussian window instead of a square window.
Harris Detector

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

\[
I(x, y) \\
\text{v}
\]

\[
w(x, y)
\]
Harris Detector

• Change in intensity for the shift \([u, v]\):
Harris Detector

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

\[
I(x, y)
\]

\[
w(x, y)
\]
Harris Detector

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

\[ I(x, y) \]

\[ w(x, y) \]
Harris Detector

\[ S_W(\Delta x, \Delta y) = \sum_{x_i \in W} \sum_{y_i \in W} (f(X_i, Y_i) - f(X_i - \Delta x, y_i - \Delta y))^2 \]

- Using first-order Taylor approximation:

\[
f(x_i - \Delta x, y_i - \Delta y) \approx f(x_i, y_i) + \left[ \frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] \Delta\begin{bmatrix} x \\ y \end{bmatrix}
\]

\[
f(x) = f(a) + \frac{f'(a)}{1!}(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x - a)^n + O(x^{n+1})
\]
Harris Detector

\[ S_W(\Delta x, \Delta y) = \sum_{x_i \in W} \sum_{y_i \in W} (f(X_i, Y_i) - f(X_i - \Delta x, Y_i - \Delta y))^2 \]

\[ S_W(\Delta x, \Delta y) = \sum_{x_i \in W} \sum_{y_i \in W} \left( f(x_i, y_i) - f(x_i, y_i) - \left[ \frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] [\Delta x] [\Delta y] \right)^2 \]

\[ = \sum_{x_i \in W} \sum_{y_i \in W} \left( -\left[ \frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] \right)^2 \]

\[ = \sum_{x_i \in W} \sum_{y_i \in W} \left( \left[ \frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] \right)^2 \]

Since \( u^2 = u^T u \)
Harris Detector

\[ A_w(x, y) = \begin{bmatrix} \sum_{x_i \in W} \sum_{y_i \in W} \left( \frac{\partial f(x_i, y_i)}{\partial x} \right)^2 & \sum_{x_i \in W} \sum_{y_i \in W} \frac{\partial f(x_i, y_i)}{\partial x} \frac{\partial f(x_i, y_i)}{\partial y} \\ \sum_{x_i \in W} \sum_{y_i \in W} \frac{\partial f(x_i, y_i)}{\partial x} \frac{\partial f(x_i, y_i)}{\partial y} & \sum_{x_i \in W} \sum_{y_i \in W} \left( \frac{\partial f(x_i, y_i)}{\partial y} \right)^2 \end{bmatrix} \]

2X2 matrix (auto-correlation or 2nd order moment matrix)
Harris Detector

- General case – use window function (step):

\[
S_W(\Delta x, \Delta y) = \sum_{x_i, y_i} w(x_i, y_i) [f(x_i, y_i) - f(x_i - \Delta x, y_i - \Delta y)]^2
\]

Default window function \(w(x, y):\)

1 in window, 0 outside

Harris Matrix

\[
A_w(x, y) = \begin{bmatrix}
\sum_{x,y} w(x, y)f_x^2 & \sum_{x,y} w(x, y)f_xf_y \\
\sum_{x,y} w(x, y)f_xf_y & \sum_{x,y} w(x, y)f_y^2
\end{bmatrix} = \sum_{x,y} \begin{bmatrix} f_x^2 & f_xf_y \\ f_xf_y & f_y^2 \end{bmatrix}
\]

where

\[
f_x = \frac{\partial f(x, y)}{\partial x}, \quad f_y = \frac{\partial f(x, y)}{\partial y}
\]
Harris Detector

- Harris uses a Gaussian window \( w(x, y) = G(x, y, \sigma_1) \) where \( \sigma_1 \) is called the “integration” scale:

\[
A_w(x, y) = \begin{bmatrix}
\sum_{x,y} w(x, y)f_x^2 & \sum_{x,y} w(x, y)f_xf_y \\
\sum_{x,y} w(x, y)f_xf_y & \sum_{x,y} w(x, y)f_y^2
\end{bmatrix} = \sum_{x,y} \begin{bmatrix}
f_x^2 & f_xf_y \\
f_xf_y & f_y^2
\end{bmatrix}
\]
Harris Detector

Describes the gradient distribution (i.e. local structure) inside window

\[ A_w = \sum_{x,y} \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix} \]

Does not depend on \( \Delta x, \Delta y \)
Harris Detector

• Since the matrix is symmetric we have:

\[ A_w = Q^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} Q \]

• We can visualize \( A_w \) as an ellipse with axis length determined by the eigenvalues and orientation determined by \( R \)

Ellipse equation:

\[ \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} A_w \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \text{const} \]
Harris Detector

- Harris Matrix
- The eigenvectors of $A_w$ encode the direction of intensity changes
- The eigenvalues of $A_w$ encode the magnitude of intensity changes
Harris Detector

Distributions of gradients $f_x$ and $f_y$: 

The distribution of the $x$ and $y$ derivatives is very different for all three types of patches.
Harris Detector

The distribution of $x$ and $y$ derivatives can be characterized by the shape and size of the principal component ellipse.

Corner Response

Corner Response

Distributions of gradients $f_x$ and $f_y$: $R = 28.07$ for Corner, $R = 0.3328$ for Linear Edge.
Harris Detector

Original Image

The 15x15 pixel neighborhoods of some of the image points for which \( \lambda_2 > 20 \)

Histogram of \( \lambda_2 \) values across the image
Harris Detector

• Measure of corner response:

$$R = \text{det}(A_w) - k (\text{trace}(A_w))^2$$

where:

$$\text{det}(A_w) = \lambda_1 \lambda_2$$

$$\text{trace}(A_w) = \lambda_1 + \lambda_2$$

(k – empirical constant, k = 0.04-0.06)

(Shi-Tomasi variation: use min(\lambda_1, \lambda_2) instead of R)
Harris Detector

• Measure of corner response:

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

(k – empirical constant)

Which is equal to:

\[ R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \]
Harris Detector

- $R$ depends only on eigenvalues of $M$
- $R$ is large for a corner
- $R$ is negative with large magnitude for and edge
- $R$ has a small magnitude for a flat region
Harris Detector

Classification of pixels using the eigenvalues of $A_w$:

$\lambda_1$ and $\lambda_2$ are small; intensity is almost constant in all directions.

$\lambda_2 \gg \lambda_1$

“Edge”

$\lambda_1$ and $\lambda_2$ are large, $\lambda_1 \sim \lambda_2$; intensity changes in all directions.

“Corner”

“Flat” region

$\lambda_1 \gg \lambda_2$

“Edge”

extra
Harris Detector

Harris Corner Detector Algorithm:

1. Filter the image with a Gaussian
2. Estimate intensity gradient in two perpendicular directions for each pixel $\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}$. This is performed by 2 – 1D convolutions with the kernel approximating the derivative.
3. Calculate the local structure matrix $A_w$
4. Evaluate the response function $R(A_w)$
5. Choose the best candidate corners by selecting a threshold on the response function $R(A_w)$ and perform non–maximal suppression
Harris Detector

Harris Corner Detector Algorithm:

1. Filter the image with a Gaussian

2. Estimate intensity gradient in two perpendicular directions for each pixel \( \frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y} \). This is performed by 2 – 1D convolutions with the kernel approximating the derivative.

3. Calculate the local structure matrix \( A_w \)

4. Evaluate the response function \( R(A_w) \)

5. Choose the best candidate corners by selecting a threshold on the response function \( R(A_w) \) and perform non–maximal suppression

\[
\frac{\partial f(x_i, y_i)}{\partial x} = \frac{\partial}{\partial x} G(x, y, \sigma_D) * f(x_i, y_i) \quad \frac{\partial f(x_i, y_i)}{\partial y} = \frac{\partial}{\partial y} G(x, y, \sigma_D) * f(x_i, y_i)
\]

\( \sigma_D \) is called the “differentiation” scale
Harris Detector Steps

1. Compute the horizontal and vertical Gaussian derivatives

\[ f_x = \frac{\partial}{\partial x} G(x, y, \sigma_D) * f(x_i, y_i) \quad f_y = \frac{\partial}{\partial y} G(x, y, \sigma_D) * f(x_i, y_i) \]  
\( \sigma_D \) is the “differentiation” scale

2. Calculate the components of structure matrix \( A_w \)

\[ A_w = \sum_{x \in w, y \in w} w(x, y) \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix} \]

\[ f_x^2 = f_x f_x \]
\[ f_y^2 = f_y f_y \]
\[ f_x f_y = f_x f_y \]
Harris Detector Steps

3. Convolve each element with the windowing function (Gaussian)

4. Determine cornerness measure:

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

5. Find local maxima
Harris Detector

Derivative of Gaussian kernel review:

$$\frac{\partial}{\partial x} G(x, y, \sigma_D) =$$

\[
\begin{array}{cccccc}
0.003 & 0.013 & 0.022 & 0.013 & 0.003 \\
0.013 & 0.059 & 0.098 & 0.059 & 0.013 \\
0.022 & 0.098 & 0.162 & 0.098 & 0.022 \\
0.013 & 0.059 & 0.098 & 0.059 & 0.013 \\
0.003 & 0.013 & 0.022 & 0.013 & 0.003 \\
\end{array}
\]

Gaussian (5x5)

* [1 − 1] Derivative Approximation
## Harris Detector

Derivative of Gaussian kernel review:

\[
\frac{\partial}{\partial x} G(x, y, \sigma_D) =
\]

<table>
<thead>
<tr>
<th></th>
<th>0.003</th>
<th>0.013</th>
<th>0.022</th>
<th>0.013</th>
<th>0.003</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.013</td>
<td>0.098</td>
<td>0.162</td>
<td>0.098</td>
<td>0.022</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.059</td>
<td>0.098</td>
<td>0.098</td>
<td>0.013</td>
<td></td>
</tr>
<tr>
<td>0.013</td>
<td>0.059</td>
<td>0.098</td>
<td>0.098</td>
<td>0.013</td>
<td></td>
</tr>
<tr>
<td>0.003</td>
<td>0.013</td>
<td>0.022</td>
<td>0.013</td>
<td>0.003</td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{bmatrix}
1 \\
-1
\end{bmatrix}
\]

Gaussian (5x5)

*Derivative Approximation*
Harris Detector

Derivative of Gaussian kernel review:

\[ y \text{-direction} \quad x \text{-direction} \quad y \text{-direction} \]
Harris Detector

Example

\[\begin{align*}
det(M) - \lambda &\quad \text{trace}(M) = \\
M &= \begin{bmatrix}
g*I_x^2 & g*(I_xI_y) \\
g*(I_xI_y) & g*I_y^2
\end{bmatrix}
\end{align*}\]
Harris Detector

Example
Harris Detector

Compute corner response $R$: 
Harris Detector

Find point with large corner response R>threshold
Harris Detector

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

Map corners back to original image:
Harris Detector

• The Harris detector requires two scale parameters:

I. A differentiation scale $\sigma_D$ for smoothing prior to the computation of image derivatives

II. An integration scale $\sigma_I$ for defining the size of the Gaussian window (i.e. integrating derivative responses)

$$A_w(x, y) \rightarrow A_w(x, y, \sigma_I, \sigma_D)$$

• Typically, $\sigma_I = \gamma \sigma_D$ (i.e. a linear relationship)
Harris Detector

• Is the Harris detector invariant to geometric and photometric changes?

• Geometric:
  - Rotation:
  - Scale:
  - Affine:

• Photometric
  - Affine intensity change: $l(x, y) \rightarrow aI(x, y) + b$
Harris Detector

- Rotation Invariance?
- Ellipse Rotates but its shape (i.e. eigenvalues) remains the same

Corner response $R$ is invariant to image rotation
Harris Detector

• Rotation
Harris Detector

- Photometric Invariance?
- Affine intensity change
- Only derivatives are used → invariance to intensity shift $I(x, y) \rightarrow I(x, y) + b$
- Intensity scale: $I(x, y) \rightarrow aI(x, y)$

Corner response $R$ is partially invariant to affine intensity change
Harris Detector

• Scale Invariance?

• All points will be classified as edges (at limit)

Corner response R is not invariant to scaling (or affine transforms)
Harris Detector

- Disadvantages
- Sensitive to:
  - Scale change
  - Significant viewpoint change
  - Significant contrast change
Harris Detector

• How to handle scale changes?
  • \( A_W \) must be adapted to scale changes
  • If the scale change is known we can adapt the Harris detector to the scale change (i.e. adjust \( \sigma_I, \sigma_D \) proportional to the scale)
  • What if the scale change is unknown?
Harris Detector

- Multi-scale Harris Detector
- Detects interest points at varying scales

\[
R(A_w) = \det(A_x(x, y, \sigma_I, \sigma_D)) - k \cdot trace(A_w(x, y, \sigma_I, \sigma_D))^2
\]

Loop through multiple scales

\[\sigma_n = \alpha^n \sigma\]
\[\sigma_D = \sigma_n\]
\[\sigma_I = \gamma \sigma_D\]

Adjust parameters at each scale
Harris Detector

- Exhaustive search
- Multiscale-approach:
Harris Detector

- Exhaustive search
- Multiscale-approach:
Harris Detector

- Exhaustive search
- Multiscale-approach:
Harris Detector

- Exhaustive search
- Multiscale-approach:
Harris Detector

• How to handle scale changes?
• Not a good idea!
• There will be many points representing the same structure, complicating matching
• Note that point locations shift as scale increases

The size of the circle corresponds to the scale at which the point was detected
Harris Detector

- How to handle scale changes?
- How do we choose corresponding circles independently in each image?
Harris Detector

- How to handle scale changes?
- Alternate approach: Use scale selection to find the characteristic scale of each feature
- Characteristic scale depends on the feature’s spatial extent (i.e. local neighborhood of pixels)
Harris Detector

- Automatic scale selection
- Design a function $F(x, \sigma_n)$ which provides some local measure
- Select points at which $F(x, \sigma_n)$ is maximal over $\sigma_n$

max of $F(x, \sigma_n)$ corresponds to characteristic scale
Harris Detector

- Automatic scale selection
- Design a function on the region which is “scale invariant” (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 (patch width)
Harris Detector

- Automatic scale selection
- Common approach: take a local maximum of this function
- Observation: region size, for which maximum is achieved should be *invariant* to image scale
- Important: this scale invariant region size is found in each image *independently*
Harris Detector

- Automatic scale selection

Same operator responses if the patch contains the same image up to scale factor.
Harris Detector

Function responses for increasing scale (scale signature)

\[ f(I_{x \cdot \sigma}) \]

\[ f(I_{x' \cdot \sigma}) \]
Harris Detector

Function responses for increasing scale (scale signature)
Harris Detector

Function responses for increasing scale (scale signature)
Harris Detector

Function responses for increasing scale (scale signature)
Harris Detector

Function responses for increasing scale (scale signature)
Harris Detector

Function responses for increasing scale (scale signature)
Harris Detector

- Scale Selection

- Use the scale determined by detector to compute descriptor in a normalized frame
Harris Detector

- Scale Selection

- Characteristic scale is (mostly) independent of the image scale

- The ratio of the scale values corresponding to the max values is equal to scale factor between the images

Scale selection allows for finding spatial extend that is covariant with the image transformation.

Scale factor: 2.5
Harris Detector

• Automatic Scale Selection

• What local measure $F(x, \sigma_n)$ should we use?

• Should be rotation invariant

• Should have one stable peak
Review

• Patch/Template Matching

• Interest Points:
  • Corner Detection
  • Moravec Detector
  • Harris Detector

• Automatic scale detection