CEE598 - Visual Data Sensing and Analytics for Civil Infrastructure Eng. & Mgmt.

Session 10 - Linear Filters
From Spatial Domain to Frequency Domain

Mani Golparvar-Fard
Department of Civil and Environmental Engineering
Department of Computer Science
3129D, Newmark Civil Engineering Lab
e-mail: mgolpar@illinois.edu
NVIDIA and Kespry Demo AI Prototype Drone
Class Projects

- Design collaborative, auditable, digital workflows
  - Reduce project risk and improve project quality

https://www.aconex.com/blogs/2014/03/construction-workflow-management.html
Outline

What we have learned so far:

• Review of lighting
• Reflection and absorption
• What is image filtering and how do we do it?
• Color models (if time allows)

This Session:

• Fourier transform and frequency domain
  ▫ Frequency view of filtering
  ▫ Sampling

Reading: [FP] Chapters 7,8
Review: image filtering

\[ g[\cdot, \cdot] = \frac{1}{9} \]

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
90 & 90 & 90 \\
90 & 90 & 90 \\
90 & 90 & 90 \\
90 & 90 & 90 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[
h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l]
\]
Review: image filtering

\[ f[\cdot, \cdot] \]

\[ h[\cdot, \cdot] \]

\[ h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l] \]
Review: image filtering

\[ f[.,.] \quad h[.,.] \]

\[
h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l]
\]

Credit: S. Seitz
Review: image filtering

\[ f[\cdots] \quad h[\cdots] \]

\[ h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l] \]
Review: image filtering

\[ f[\cdot,\cdot] \quad \text{and} \quad h[\cdot,\cdot] \]

\[ h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l] \]

Credit: S. Seitz
Review: image filtering

\[ f[m,n] = \sum_{k,l} g[k,l] \cdot f[m+k,n+l] \]

Credit: S. Seitz
Review: image filtering in spatial domain

Sobel filter:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>

This image shows a Sobel filter applied to a portrait of Albert Einstein, highlighting edges and details.
Difficulty in Spatial Domain

- Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?
Thinking in terms of frequency

Why does a lower resolution image still make sense to us? What do we lose?
Fourier analysis in images

- An important image processing tool to decompose an image into its sine and cosine components.
- The output represents the image which is in spatial domain in the Fourier or frequency domain.
- In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image.

Intensity Image

Fourier Image

More: http://www.cs.unm.edu/~brayer/vision/fourier.html
Log Magnitude

Strong Vertical Frequency (Sharp Horizontal Edge)

Diagonal Frequencies

Strong Horz. Frequency (Sharp Vert. Edge)

Low Frequencies

Log Magnitude
Fourier Transform

- Fourier transform stores the magnitude and phase at each frequency
  - Magnitude encodes how much signal there is at a particular frequency
  - Phase encodes spatial information (indirectly)
  - For mathematical convenience, this is often notated in terms of real and complex numbers

Amplitude: \[ A = \pm \sqrt{R(\omega)^2 + I(\omega)^2} \]
Phase: \[ \phi = \tan^{-1} \frac{I(\omega)}{R(\omega)} \]

Euler's formula: \[ e^{inx} = \cos(nx) + i \sin(nx) \]
Discrete Fourier Transform (DFT)

- For a square image of size $N \times N$, the two-dimensional DFT is given by:

$$H(\omega) = \mathcal{F}\{h(x)\} = A e^{j\phi}$$

$$F(k, l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) e^{-j2\pi\left(\frac{k_i}{N} + \frac{l_j}{N}\right)}$$

where $f(a, b)$ is the image in the spatial domain and the exponential term is the basis function corresponding to each point $F(k, l)$ in the Fourier space.

The equation can be interpreted as:
the value of each point $F(k, l)$ is obtained by multiplying the spatial image with the corresponding base function and summing the result.
The Convolution Theorem

- The Fourier transform of the convolution of two functions is the product of their Fourier transforms
  \[ F[g \ast h] = F[g]F[h] \]

- The inverse Fourier transform of the product of two Fourier transforms is the convolution of the two inverse Fourier transforms

- **Convolution** in spatial domain is equivalent to **multiplication** in frequency domain!
  \[ F^{-1}[gh] = F^{-1}[g] \ast F^{-1}[h] \]
Properties of Fourier Transforms

- Linearity
  \[ \mathcal{F}[ax(t) + by(t)] = a\mathcal{F}[x(t)] + b\mathcal{F}[y(t)] \]

- Fourier transform of a real signal is symmetric about the origin

- The energy of the signal is the same as the energy of its Fourier transform

See Szeliski Book (3.4)
Filtering in spatial domain

\[
\begin{bmatrix}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1 \\
\end{bmatrix}
\]

Intensity image

\[\ast\]

Resulting image
Filtering in frequency domain

1. **Intensity Image**: Input image.
2. **FFT**: Fourier Transform applied, converting to frequency domain.
3. **Filtering in Frequency Domain**: Applying filters in the frequency domain.
4. **Inverse FFT**: Reversing the process to convert back to the spatial domain.

The diagram illustrates the process of filtering data in the frequency domain, which is commonly used in signal processing and image analysis.
FFT in Matlab

- Filtering with fft

```matlab
im = ... % "im" should be a gray-scale floating point image
[imh, imw] = size(im);
fftsize = 1024; % should be order of 2 (for speed) and include padding
im_fft = fft2(im, fftsize, fftsize); % 1) fft im with padding
hs = 50; % filter half-size
fil = fspecial('gaussian', hs*2+1, 10);
fil_fft = fft2(fil, fftsize, fftsize); % 2) fft fil, pad to same size as image
im_fil_fft = im_fft .* fil_fft; % 3) multiply fft images
im_fil = ifft2(im_fil_fft); % 4) inverse fft2
im_fil = im_fil(1+hs:size(im,1)+hs, 1+hs:size(im, 2)+hs); % 5) remove padding
```

- Displaying with fft

```matlab
figure(1), imagesc(log(abs(fftshift(im_fft)))), axis image, colormap jet
```
Filtering

Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?

Gaussian

Box filter
Box Filter
Why does a lower resolution image still make sense to us? What do we lose?
Subsampling by a factor of 2

Throw away every other row and column to create a 1/2 size image
Aliasing problem

- 1D example (sinewave):
Aliasing problem

- 1D example (sinewave):
Aliasing problem

- Sub-sampling may be dangerous….
- Characteristic errors may appear:
  - “Wagon wheels rolling the wrong way in movies”
  - “Checkerboards disintegrate in ray tracing”
  - “Striped shirts look funny on color television”
Aliasing in video

Imagine a spoked wheel moving to the right (rotating clockwise). Mark wheel with dot so we can see what’s happening.

If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):

Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)
Aliasing in graphics

Disintegrating textures

Source: A. Efros
Sampling and aliasing

256x256  128x128  64x64  32x32  16x16
When sampling a signal at discrete intervals, the sampling frequency must be $\geq 2 \times f_{\text{max}}$.

- $f_{\text{max}} = \text{max frequency of the input signal}$
- This will allow to reconstruct the original perfectly from the sampled version.

Nyquist-Shannon Sampling Theorem
Anti-aliasing

Solutions:

- Sample more often

- Get rid of all frequencies that are greater than half the new sampling frequency
  - Will lose information
  - But it’s better than aliasing
  - Apply a smoothing filter
Algorithm for downsampling by factor of 2

1. Start with image(h, w)
2. Apply low-pass filter
   \[
   \text{im\_blur} = \text{imfilter}(\text{image}, \text{fspecial('gaussian', 7, 1)})
   \]
3. Sample every other pixel
   \[
   \text{im\_small} = \text{im\_blur}(1:2:end, 1:2:end);
   \]
Anti-aliasing

Forsyth and Ponce 2002
Subsampling without pre-filtering

1/2

1/4 (2x zoom)

1/8 (4x zoom)
Subsampling with Gaussian pre-filtering

Gaussian 1/2

G 1/4

G 1/8
Why does a lower resolution image still make sense to us? What do we lose?
Clues from Human Perception

- Early processing in humans filters for various orientations and scales of frequency
- Perceptual cues in the mid-high frequencies dominate perception
- When we see an image from far away, we are effectively subsampling it

Early Visual Processing: Multi-scale edge and blob filters
Most operation-level progress is non-physical

- Sequence of construction activities are not formalized
- Low visibility of as-built elements
Method Overview

Key components:
- D4AR model (image-based point cloud + 4D BIM)
- Reasoning about occlusions
- Material classification method
- Inference on the State of progress
1. Back-Projection of BIM elements

1. BIM and point cloud registered manually
   Similarity Transformation (7DOF) in m3 points

2. Projective transformation 3D-2D
   \[ p_{\text{cam}}^k = (R v^k + T) / p_{\text{cam}}^k \cdot z \]
   Transform \( v^k \) into camera coordinate
   Perspective division

3. Reason about occlusions
   Sort those faces by their distance to the camera location

4. Choose one face for assessment
   Max 2D back-projected area of \( E^i \)
2. Sample from Back-Projections

Back-projection of BIM onto each camera

Visible areas of each element
3. Infer Progress Based on Material

Material Classification Method

Result of Material Recognition

Frequency of Appearance (%)

Concrete Wood Steel Soil Misc.

Element ID #112
Viewpoint #0

Element ID #112
Viewpoint #1

Element ID #112
Viewpoint #3

Element ID #112
Viewpoint #5
Overview of Material Classification Algorithm

K-Means Cluster and Form Texton Dictionary

Joint Histograms

Categorize and Label Material

One-vs.-all Classification:

Multiple non-linear χ² kernel SVM Classifiers

Andrey Dimitrov, Mani Golparvar-Fard, Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections, Advanced Engineering Informatics, 28 (1), 2014, 37-49.
Image Filters

Color Filters
Hue, saturation, & value (brightness)

Texture Filters
36 oriented filters at 6 orientations, 3 scales, and 2 filters & 8 derivative filters and 4 low-pass Gaussian filters.
Experimental Setup

- **Construction Material Library**

- 22 Material Types
- 3,000+ 30x30 pixel images from 7 existing/under-construction sites
  - Varying degrees of viewpoint, scale, and illumination changes
- Approximate physical length of 12", 24" and 48"
- Three to five view orientations
- 9-15 poses make up each material sub-category

Dataset is public at: [http://raamac.cee.illinois.edu/materialclassification](http://raamac.cee.illinois.edu/materialclassification)
Accuracy of Material Classification

- Confusion matrix

![Confusion matrix diagram](image)
Experimental Setup (2) and Results

- Inferring State of Progress

<table>
<thead>
<tr>
<th></th>
<th>BIM</th>
<th># of IFC elements</th>
<th># of Images</th>
<th>condition</th>
<th>in-place</th>
<th>missing</th>
<th>Non-visible</th>
</tr>
</thead>
<tbody>
<tr>
<td>RH1</td>
<td>LOD300</td>
<td>134</td>
<td>160</td>
<td>Sunny</td>
<td>125</td>
<td>9</td>
<td>63</td>
</tr>
<tr>
<td>RH2</td>
<td>LOD300</td>
<td>134</td>
<td>112</td>
<td>Sunny</td>
<td>123</td>
<td>11</td>
<td>34</td>
</tr>
<tr>
<td>SD</td>
<td>LOD200</td>
<td>117</td>
<td>288</td>
<td>Cloudy</td>
<td>114</td>
<td>3</td>
<td>24</td>
</tr>
</tbody>
</table>

e.g., State of Progress for three elements in RH1

We can successfully detect all visible concrete elements
Things to Remember

- Sometimes it makes sense to think of images and filtering in the frequency domain
  - Fourier analysis

- Can be faster to filter using FFT for large images (\(N \log N\) vs. \(N^2\) for auto-correlation)

- Images are mostly smooth
  - Basis for compression

- Remember to low-pass before sampling
Practice question

1. Match the spatial domain image to the Fourier magnitude image
CEE598 - Visual Data Sensing and Analytics for Civil Infrastructure Eng. & Mgmt.

Session 10b - Image Detectors, Part I

Mani Golparvar-Fard
Department of Civil and Environmental Engineering
Department of Computer Science
3129D, Newmark Civil Engineering Lab
e-mail: mgolpar@illinois.edu

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Outline

Image Detectors, Part I

• Edge feature detectors
• Corner feature detectors

Reading: [FP] Chapters 8,9

Some slides in this lecture are courtesy to Prof. S. Savarese, prof F. Li, prof S. Lazebnik, and various other lecturers
Goal

- Identify interesting regions from the images (edges, corners, blobs...)

Descriptors

e.g. SIFT

Matching / Indexing / Recognition
Linear filtering

- Convolution:

\[(f * g)[m, n] = \sum_{k,l} f[k, l] g[m - k, n - l]\]

- Smoothing
- Differentiation
Smoothing with a Gaussian

- Weight contributions of neighboring pixels by nearness

\[
G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}
\]

- Constant factor at front makes volume sum to 1 (can be ignored, as we should normalize weights to sum to 1 in any case).

Slide credit: Christopher Rasmussen
Smoothing with a Gaussian
Edge Detection
What causes an edge?

Identifies sudden changes in an image

- Depth discontinuity
- Surface orientation discontinuity
- Reflectance discontinuity (i.e., change in surface material properties)
- Illumination discontinuity (e.g., shadow)
Edge Detection

- Criteria for **optimal edge detection** (Canny 86):
  - **Good detection accuracy:**
    - minimize the probability of false positives (detecting spurious edges caused by noise),
    - false negatives (missing real edges)
  - **Good localization:**
    - edges must be detected as close as possible to the true edges.
  - **Single response constraint:**
    - minimize the number of local maxima around the true edge
    - (i.e. detector must return single point for each true edge point)
Edge Detection

• Examples:

True edge

Poor robustness to noise

Poor localization

Too many responses
Designing an edge detector

- **Edge:** a location with high gradient (thus, use derivatives)

- Need two derivatives, in x and y direction.

- Need smoothing to reduce noise prior to taking derivative
\[
\frac{d}{dx} (f \ast g)
\]

“derivative of Gaussian” filter

Source: S. Seitz
Canny Edge Detection

- Most widely used edge detector in computer vision.

- First derivative of the Gaussian closely approximates the operator that optimizes the product of signal-to-noise ratio and localization.
Canny Edge Detection

Steps:

1. Gaussian smoothing
2. & Derivative = Derivative of Gaussian
3. Find magnitude and orientation of gradient
4. Extract edge points: ‘Non-maximum suppression’
5. Linking and thresholding ‘Hysteresis’:
   - Matlab: `edge(I, 'canny')`
Canny Edge Detector - First 2 Steps

- **Smoothing**

\[ I' = g(x, y) * I \quad g(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2 + y^2}{2\sigma^2}} \]

- **Derivative**

\[ S = \nabla (g * I) = (\nabla g) * I = \begin{bmatrix} g_x \\ g_y \end{bmatrix} * I = \begin{bmatrix} g_x * I \\ g_y * I \end{bmatrix} \]

\[ \nabla g = \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \]
Canny Edge Detector
Derivative of Gaussian

\[ g(x, y) \]

\[ g_x(x, y) \]

\[ g_y(x, y) \]
Canny Edge Detector - First 2 Steps

\[ S = \nabla(g \ast I) = (\nabla g) \ast I \]

\[ S = \begin{bmatrix} S_x & S_y \end{bmatrix} \]

= gradient vector
Increased smoothing:

- Eliminates noise edges.
- Makes edges smoother and thicker.
- Removes fine detail.
Canny Edge Detector- Third Step

- magnitude and direction of

\[ S = \begin{bmatrix} S_x & S_y \end{bmatrix} \]

magnitude = \[ \sqrt{(S_x^2 + S_y^2)} \]

direction = \[ \theta = \tan^{-1} \frac{S_y}{S_x} \]
Canny Edge Detector - Fourth Step

Non maximum suppression
Canny Edge Detector - Fourth Step

1. Initialize:
   - Slice gradient magnitude along the gradient direction
   - Mark the point along the slide where the magnitude is max

2. Propagate chain from current point:
   - Predict next points using the normal to the gradient at that point
   - Find which point is a local max magnitude in gradient direction
   - Retain in magnitude > T
Example: Non-maximum depression

- Original image
- Gradient magnitude
- Non-maxima suppressed

courtesy of G. Loy
Canny Edge Detector - Step 5: Thresholding

- Set a threshold $T$ to suppress gradients with magnitude $< T$
high threshold
(strong edges)

low threshold
(weak edges)
Canny Edge Detector
Step 5: Hysteresis Thresholding

- **Hysteresis**: A lag or momentum factor

  - Idea: Maintain two thresholds $k_{\text{high}}$ and $k_{\text{low}}$
    - Use $k_{\text{high}}$ to find strong edges to start edge chain
    - Use $k_{\text{low}}$ to find weak edges along the edge chain

  - Typical ratio of thresholds is roughly
    \[ k_{\text{high}} / k_{\text{low}} = 2 \]
hysteresis threshold
Effect of $\sigma$ (Gaussian kernel spread/size)

- The choice of $\sigma$ depends on desired behavior
  - large $\sigma$ detects large scale edges
  - small $\sigma$ detects fine features

Source: S. Seitz
Demo

http://www.cs.washington.edu/research/imagedatabase/demo/edge/
Other edge detectors:

- Sobel
- Canny-Deriche
- Differential
Extract useful building blocks: Corners
Extract useful building blocks: blobs
Characteristics

- **Repeatability**
  - The same feature can be found in several images despite geometric and photometric transformations

- **Saliency**
  - Each feature is found at an “interesting” region of the image

- **Locality**
  - A feature occupies a “relatively small” area of the image;
Repeatability

Illumination invariance

Scale invariance

Pose invariance
  • Rotation
  • Affine
• Saliency

• Locality
Harris corner detector

Harris Detector: Basic Idea

Explore intensity changes within a window as the window changes location

“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

Shifted intensity

Intensity

Proportional to the gradient

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

1 in window, 0 outside

Gaussian
For small shifts \( [u,v] \) we have a *bilinear* approximation:

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

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

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

\[ M = \begin{bmatrix} \sum W I_x^2 & \sum W I_x I_y \\ \sum W I_x I_y & \sum W I_y^2 \end{bmatrix} \]

Sum over a small region around the hypothetical corner (we can omit “w”)

Gradient with respect to x, times gradient with respect to y

\[ (g_x * I)(g_y * I) \]

Matrix is symmetric

Slide credit: David Jacobs

Second-moment matrix

\[ M = \begin{bmatrix} \sum W I_x^2 & \sum W I_x I_y \\ \sum W I_x I_y & \sum W I_y^2 \end{bmatrix} \]

Sum over a small region around the hypothetical corner (we can omit “w”)

Gradient with respect to x, times gradient with respect to y

\[ (g_x * I)(g_y * I) \]

Matrix is symmetric

Slide credit: David Jacobs
Second-moment matrix

First, consider case where dominant gradient directions aligned with x or y

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$
Second-moment matrix

First, consider case where dominant gradient directions aligned with x or y

\[
M = \begin{bmatrix}
\sum I_x^2 & \sum I_x I_y \\
\sum I_x I_y & \sum I_y^2
\end{bmatrix}
\]

Structure tensor

analyzing the eigenvalues of A

\[
\begin{bmatrix}
\lambda_1 & 0 \\
0 & \lambda_2
\end{bmatrix}
\]

If either \(\lambda\) is close to 0, then this is an edge
Second-moment matrix

First, consider case where dominant gradient directions aligned with x or y

\[ M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \]

If both \( \lambda \)s are close to 0, then this is a flat region
Second-moment matrix

For generic window alignments, the eigenvalue decomposition of $M$ returns similar information:

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = U^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U$$

Lambda 1, 2 are the eigenvalues of $M$
For generic window alignments, the eigenvalue decomposition of $M$ returns similar information:

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = U^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U$$

If either $\lambda$ is close to 0, then this is an edge.

Non-zero eigenvector of $M$ gives direction of the edge.
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 $\lambda_2$ are large, $\lambda_1 \sim \lambda_2$; $E$ increases in all directions.
- $\lambda_1 >> \lambda_2$.
Harris Detector: Mathematics

Measure of corner response:

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

\[
\det M = \lambda_1 \lambda_2
\]

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

\( (k \text{ – empirical constant, } k = 0.04-0.06) \)
Harris Detector: Algorithm

- Filter image with Gaussian to reduce noise
- Compute magnitude of the x and y gradients at each pixel
- Construct M in a window around each pixel (Harris uses a Gaussian window)
- Compute $\lambda$s of M
- Compute $R = \det M - k \left( \text{trace } M \right)^2$
- If $R > T$ a corner is detected; retain point of local maxima
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: Workflow
Harris Detector: Workflow

Compute corner response $R$
Harris Detector: Workflow

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

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

- Rotation invariance

Corner response $R$ is invariant to image rotation

$$C = U^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U \quad \Rightarrow \quad R = R(\lambda_1, \lambda_1) \text{ doesn’t change!}$$
Harris Detector: Some Properties

- But: non-invariant to *image scale*!

All points will be classified as *edges*

Corner!
Harris Detector: Some Properties

- Partial invariance to *affine intensity* change
  
  • invariance to intensity shift $I \rightarrow I + b$  (*why?)
    
    (only derivatives are used)
  
  • Intensity scale: $I \rightarrow aI$
Next lecture:

- Descriptors
- Detectors part 2