University of Idaho
Department of Computer Science
Coeur d'Alene

Lecture #8

Image Gradients and Topology

Garrett Wells

revised September 5, 2024

2024-09-05

Lecture #8

Table of Contents I

Table of Contents

Review

Image Topography

Image Gradient
- Sobel Kernel
- Scharr
- Laplacian Kernel
- Comparison

Table of Contents I

Table of Contents

Review

Image Topography

Image Gradient
Sobel Kernel
Scharr
Laplacian Kernel
Comparison

Review

Blurring and Smoothing Images

- Averaging
- Gaussian
- Median
- Bilateral
- Denoising

Review
Blurring and Smoothing Images
- Averaging
- Gaussian
- Median
- Bilateral
- Denoising

Topology & Topography

Topography

1. Detailed, precise description of a place or region.
2. Graphic representation of the surface features of a place or region on a map, indicating their relative positions and elevations.
3. A description or an analysis of a structured entity, showing the relations among its components.

Topology & Topography
Topography
1. Detailed, precise description of a place or region.
2. Graphic representation of the surface features of a place or region on a map, indicating their relative positions and elevations.
3. A description or an analysis of a structured entity, showing the relations among its components.

Topology & Topography

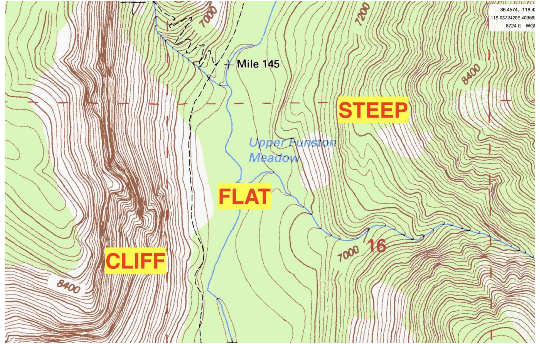


Figure: Typical Topographic Map

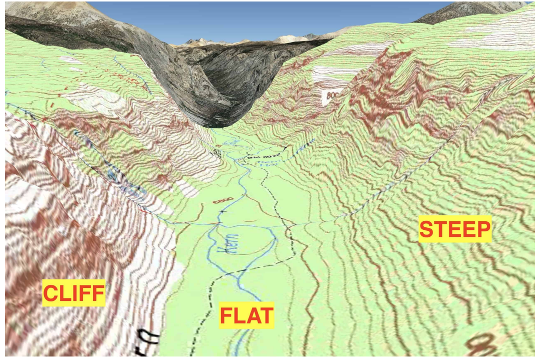
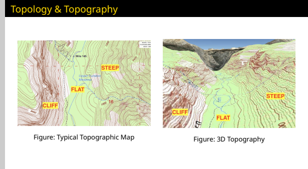


Figure: 3D Topography

2024-09-05

Lecture #8
└Review
└Topology & Topography



Topology

1. Topographic study of a given place, especially the history of a region as indicated by its topography.
2. The study of certain properties that do not change as geometric figures or spaces undergo continuous deformation. These properties include openness, nearness, connectedness, and continuity.

Topology & Topography
Topology vs. Topography
Topology
1. Topographic study of a given place, especially the history of a region as indicated by its topography.
2. The study of certain properties that do not change as geometric figures or spaces undergo continuous deformation. These properties include openness, nearness, connectedness, and continuity.

Image Topography

- Images as surfaces
- Pixel values correspond to the surface topology
- Typically look at *one* channel of image

Image Topography
- Images as surfaces
- Pixel values correspond to the surface topology
- Typically look at *one* channel of image

Image Gradient

Image Gradient

A derivative(rate of change) at a given pixel in the image. Measures the change in intensity in the x and y direction at each point in the image.

$$\triangledown f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1)$$

Image Gradient

Image Gradient

A derivative(rate of change) at a given pixel in the image. Measures the change in intensity in the x and y direction at each point in the image.

$$\triangledown f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1)$$

1. Image gradients are fundamental for edge detection since edges represent high gradient areas.
2. Gradient image calculation can reduce the impact of lighting or camera on edge detection since gradient is a measurement of change in *intensity* not color. [1]
3. Gradient is calculated as a vector whose components are the partial derivatives in x and y.

Image Gradient

Image Gradient

A derivative(rate of change) at a given pixel in the image. Measures the change in intensity in the x and y direction at each point in the image.

$$m = \sqrt{g_x^2 + g_y^2}$$
$$\theta = \arctan \frac{g_y}{g_x}$$

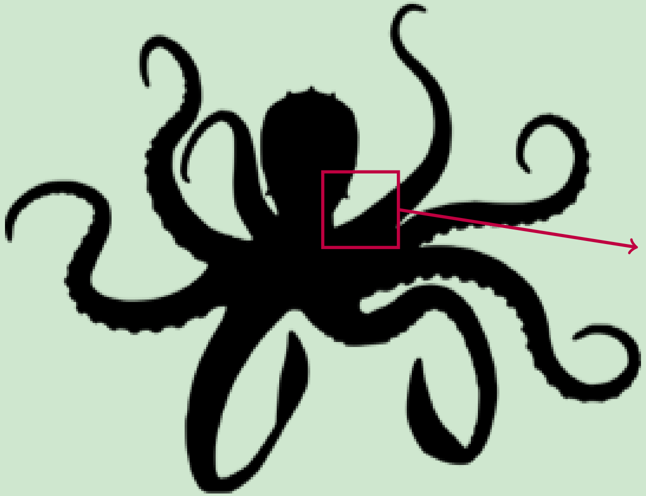
(1)

Image Gradient
Image Gradient
A derivative(rate of change) at a given pixel in the image. Measures the change in intensity in the x and y direction at each point in the image.
$$m = \sqrt{g_x^2 + g_y^2}$$
$$\theta = \arctan \frac{g_y}{g_x}$$
(1)

1. Magnitude, calculated as hypotenuse of two x & y component.

Image Gradient

Example



$$\begin{bmatrix} 50 & 200 & 200 \\ 50 & 200 & 200 \\ 200 & 50 & 50 \end{bmatrix}$$

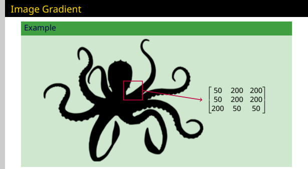
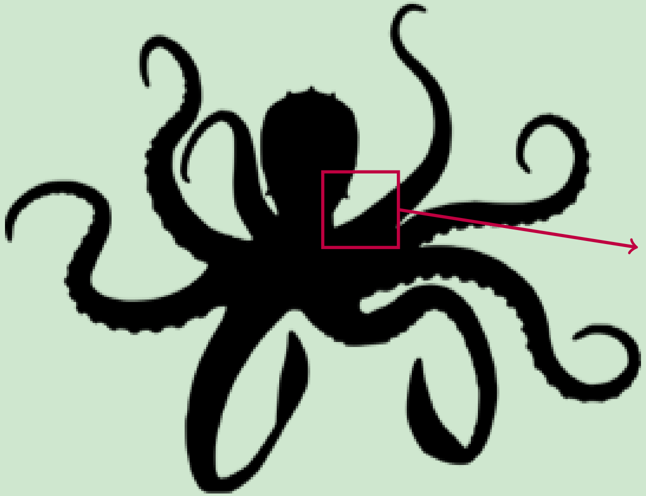


Image Gradient

Example



$$\begin{bmatrix} 50 & 200 & 200 \\ 50 & 200 & 200 \\ 200 & 50 & 50 \end{bmatrix}$$

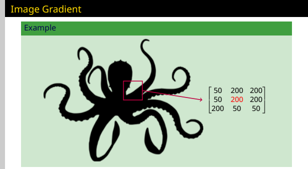
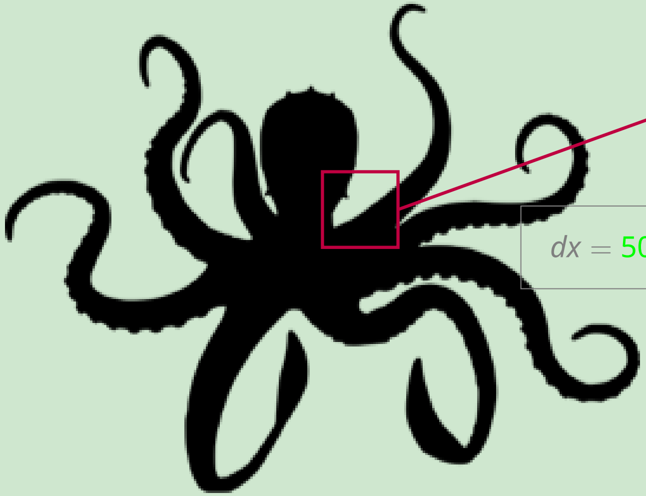


Image Gradient

dx

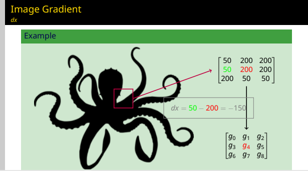
Example



$$\begin{bmatrix} 50 & 200 & 200 \\ \color{green}{50} & \color{red}{200} & 200 \\ 200 & 50 & 50 \end{bmatrix}$$

$$dx = \color{green}{50} - \color{red}{200} = -150$$

$$\begin{bmatrix} g_0 & g_1 & g_2 \\ g_3 & \color{red}{g_4} & g_5 \\ g_6 & g_7 & g_8 \end{bmatrix}$$

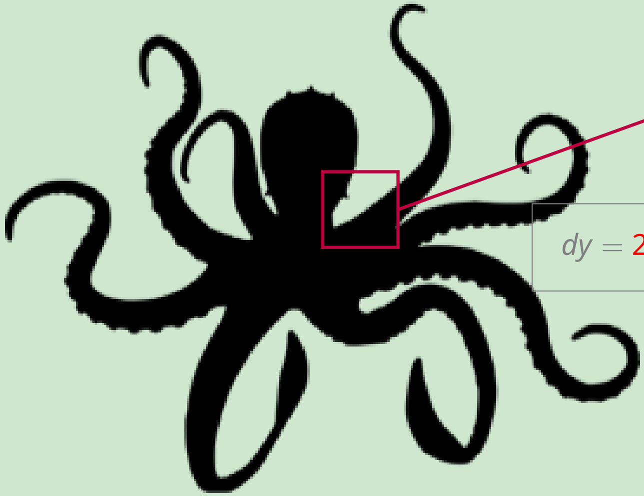


1. *dx* is calculated from left to right.

Image Gradient

dy

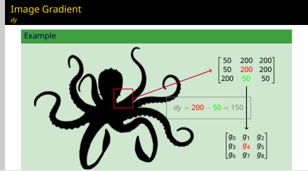
Example



$$\begin{bmatrix} 50 & 200 & 200 \\ 50 & 200 & 200 \\ 200 & 50 & 50 \end{bmatrix}$$

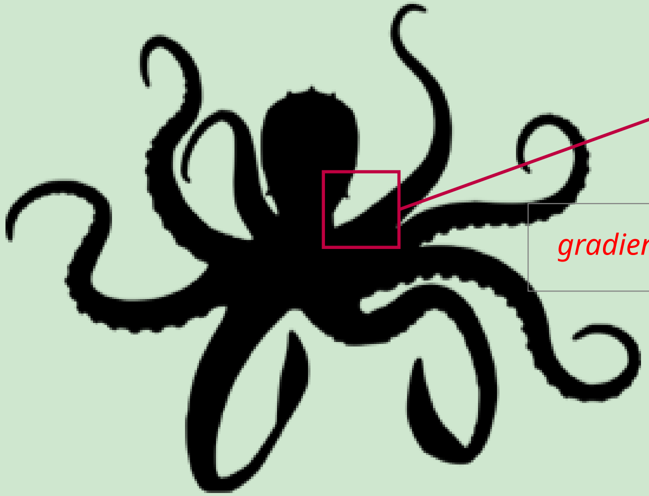
$$dy = 200 - 50 = 150$$

$$\begin{bmatrix} g_0 & g_1 & g_2 \\ g_3 & g_4 & g_5 \\ g_6 & g_7 & g_8 \end{bmatrix}$$



1. *dy* calculated from top to bottom.

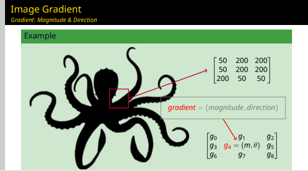
Example



$$\begin{bmatrix} 50 & 200 & 200 \\ 50 & 200 & 200 \\ 200 & 50 & 50 \end{bmatrix}$$

$$\textcolor{red}{gradient} = (magnitude, direction)$$

$$\begin{bmatrix} g_0 & g_1 & g_2 \\ g_3 & \textcolor{red}{g_4} = (m, \theta) & g_5 \\ g_6 & g_7 & g_8 \end{bmatrix}$$

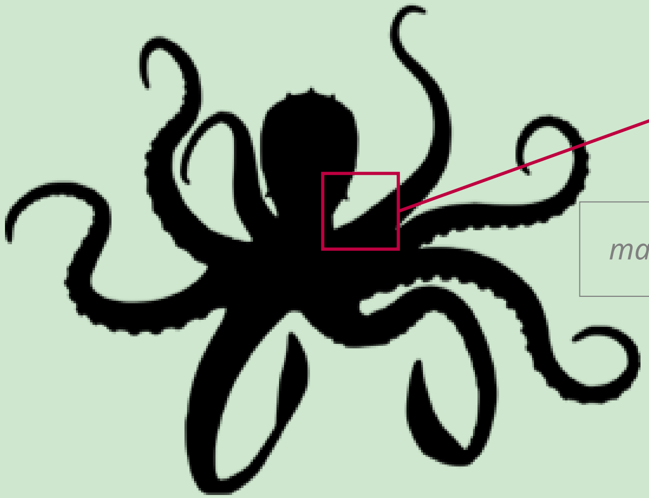


1. Adding the values of *dx* and *dy* doesn't anything particularly meaningful.
2. Better to represent as a *vector*, with both magnitude and direction.

Image Gradient

Gradient: Magnitude

Example



$$\begin{bmatrix} 50 & 200 & 200 \\ 50 & 200 & 200 \\ 200 & 50 & 50 \end{bmatrix}$$

$$magnitude = \sqrt{dx^2 + dy^2}$$

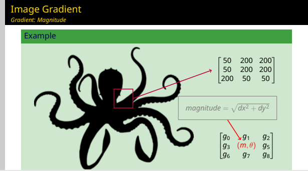
$$\begin{bmatrix} g_0 & g_1 & g_2 \\ g_3 & (m, \theta) & g_5 \\ g_6 & g_7 & g_8 \end{bmatrix}$$

2024-09-05

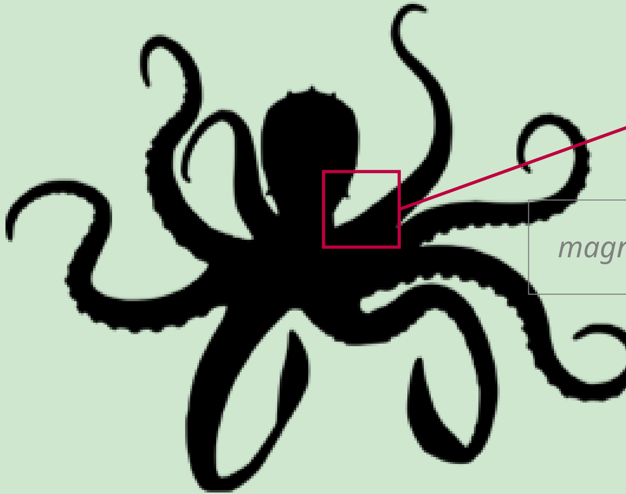
Lecture #8

└Image Gradient

└Image Gradient



Example



$$\begin{bmatrix} 50 & 200 & 200 \\ 50 & 200 & 200 \\ 200 & 50 & 50 \end{bmatrix}$$

$$\text{magnitude} = \sqrt{(150)^2 + (-150)^2}$$

$$\begin{bmatrix} g_0 & g_1 & g_2 \\ g_3 & ({\color{red}m} = 212.13, \theta = ?) & g_5 \\ g_6 & g_7 & g_8 \end{bmatrix}$$

2024-09-05

Lecture #8
└Image Gradient
└Image Gradient

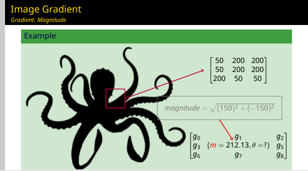
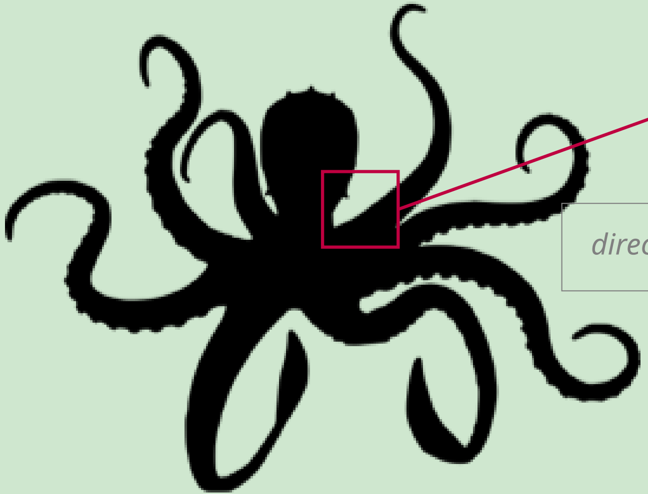


Image Gradient

Gradient: Direction

Example



$$\begin{bmatrix} 50 & 200 & 200 \\ 50 & 200 & 200 \\ 200 & 50 & 50 \end{bmatrix}$$

$$\mathit{direction}(\theta) = \arctan\left(dx/dy\right)$$

$$\begin{bmatrix} g_0 & g_1 & g_2 \\ g_3 & (212.13, \theta = ?) & g_5 \\ g_6 & g_7 & g_8 \end{bmatrix}$$

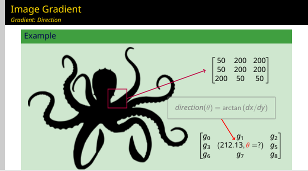
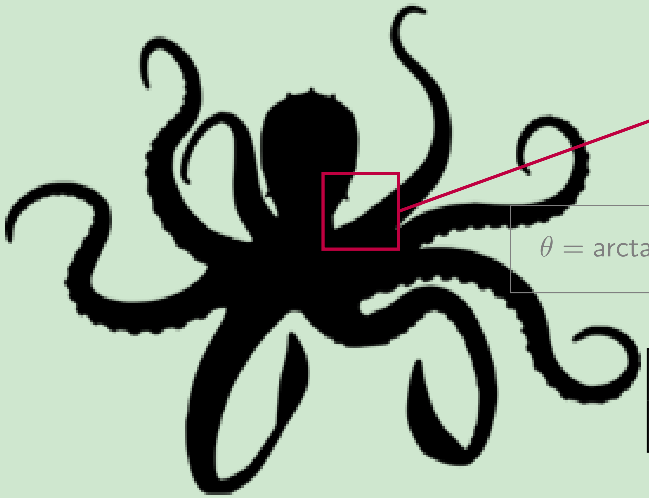


Image Gradient

Gradient: Direction

Example



$$\begin{bmatrix} 50 & 200 & 200 \\ 50 & 200 & 200 \\ 200 & 50 & 50 \end{bmatrix}$$

$$\theta = \arctan\left(-150/150\right) = \arctan\left(-1\right)$$

$$\begin{bmatrix} g_0 & g_1 & g_2 \\ g_3 & (212.13, \theta = -0.785) & g_5 \\ g_6 & g_7 & g_8 \end{bmatrix}$$

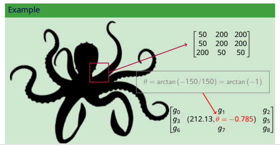


Image Gradient

- $g_n = (m, \theta)$
- Can use this to map changes in image
- Can calculate this with convolution

Image Gradient
- $g_n = (m, \theta)$
- Can use this to map changes in image
- Can calculate this with convolution

Sobel Kernel

Image Derivative

- Approximate derivative of image

- Different kernels for X/Y directions

$$dx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\mathrm{dy} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Figure: Sobel Kernels

Sobel Kernel
Image Derivative
Approximate derivative of image
Different kernels for X/Y directions
dx = [−1 0 1; −2 0 2; −1 0 1]
dy = [−1 −2 −1; 0 0 0; 1 2 1]
Figure: Sobel Kernels

1. Sobel operators combine Gaussian smoothing and differentiation

Sobel Kernel

Image Derivative

- Approximate derivative of image
  - at dimension 3, Sobel Kernel may produce inaccuracies
- Different kernels for X/Y directions

$$dx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\text{dy} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Figure: Sobel Kernels

Sobel Kernel
Image Derivative
Approximate derivative of image
at dimension 3, Sobel Kernel may produce inaccuracies
Different kernels for X/Y directions
Figure: Sobel Kernels

1. Sobel operators combine Gaussian smoothing and differentiation

Sobel Kernel

Image Derivative

- Approximate derivative of image
    - Scharr Kernel addresses this inaccuracy
- Different kernels for X/Y directions

$$dx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$dy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Figure: Sobel Kernels

Sobel Kernel
Image Derivative
Approximate derivative of image
Scharr Kernel addresses this inaccuracy
Different kernels for X/Y directions
Figure: Sobel Kernels

1. Sobel operators combine Gaussian smoothing and differentiation

Example

Sobel Kernel, dx

$$\begin{bmatrix} Pixel \\ Matrix \end{bmatrix} \begin{bmatrix} Sobel \\ dx \end{bmatrix} = \begin{bmatrix} dx \\ value \end{bmatrix} \tag{1}$$

Example
Sobel Kernel, dx

$$\begin{bmatrix} Pixel \\ Matrix \end{bmatrix} \begin{bmatrix} Sobel \\ dx \end{bmatrix} = \begin{bmatrix} dx \\ value \end{bmatrix} \tag{1}$$

Example

Sobel Kernel, dx

$$
\begin{bmatrix} 50 & 200 & 200 \\ 50 & 200 & 200 \\ 200 & 50 & 50 \end{bmatrix}
\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}
=
\begin{bmatrix} 50 \cdot -1 & 200 \cdot 0 & 200 \cdot 1 \\ 50 \cdot -2 & 200 \cdot 0 & 200 \cdot 2 \\ 200 \cdot -1 & 50 \cdot 0 & 50 \cdot 1 \end{bmatrix} \tag{1}
$$

2024-09-05

Lecture #8
└Image Gradient
　└Sobel Kernel
　　└Example

Example
Sobel Kernel, dx

Example

Sobel Kernel, dx

$$
\begin{aligned}
&= (50\cdot -1) + (200\cdot 0) + (200\cdot 1) \\
&\quad +(50\cdot -2) + (200\cdot 0) + (200\cdot 2) \\
&\quad +(200\cdot -1) + (50\cdot 0) + (50\cdot 1) \\
&= 300
\end{aligned} \tag{1}
$$

Example
Sobel Kernel, dx
$= (50\cdot -1) + (200\cdot 0) + (200\cdot 1) +(50\cdot -2) + (200\cdot 0) + (200\cdot 2) +(200\cdot -1) + (50\cdot 0) + (50\cdot 1) = 300$ (1)

Example

Sobel Kernel, dy

$$\begin{bmatrix} Pixel \\ Matrix \end{bmatrix} \begin{bmatrix} Sobel \\ dy \end{bmatrix} = \begin{bmatrix} dy \\ Value \end{bmatrix} \tag{2}$$

Example
Sobel Kernel, dy

$$\begin{bmatrix} Pixel \\ Matrix \end{bmatrix} \begin{bmatrix} Sobel \\ dy \end{bmatrix} = \begin{bmatrix} dy \\ Value \end{bmatrix} \tag{2}$$

Example

Sobel Kernel, dy

$$
\begin{bmatrix} 50 & 200 & 200 \\ 50 & 200 & 200 \\ 200 & 50 & 50 \end{bmatrix}
\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}
=
\begin{bmatrix} 50\cdot -1 & 200\cdot -2 & 200\cdot -1 \\ 50\cdot 0 & 200\cdot 0 & 200\cdot 0 \\ 200\cdot 1 & 50\cdot 2 & 50\cdot 1 \end{bmatrix}
\tag{2}
$$

Example
Sobel Kernel, dy

$$
\begin{bmatrix} 50 & 200 & 200 \\ 50 & 200 & 200 \\ 200 & 50 & 50 \end{bmatrix}
\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}
=
\begin{bmatrix} 50\cdot -1 & 200\cdot -2 & 200\cdot -1 \\ 50\cdot 0 & 200\cdot 0 & 200\cdot 0 \\ 200\cdot 1 & 50\cdot 2 & 50\cdot 1 \end{bmatrix}
\tag{2}
$$

Example

Sobel Kernel, dy

$$
\begin{aligned}
&= (50\cdot -1) + (200\cdot -2) + (200\cdot -1)\\
&\quad +(50\cdot 0) + (200\cdot 0) + (200\cdot 0)\\
&\quad +(200\cdot 1) + (50\cdot 2) + (50\cdot 1)\\
&= -300
\end{aligned} \tag{2}
$$

2024-09-05

Lecture #8
└Image Gradient
└Sobel Kernel
└Example

Example
Sobel Kernel, dy
$= (50\cdot -1) + (200\cdot -2) + (200\cdot -1) +(50\cdot 0) + (200\cdot 0) + (200\cdot 0) +(200\cdot 1) + (50\cdot 2) + (50\cdot 1) = -300$ (2)

Scharr Kernel

- Another derivative kernel
- **Goal:** achieve better accuracy at size 3 [2]

$$dx = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} \tag{3}$$

$$dy = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix} \tag{4}$$

Scharr Kernel

- Another derivative kernel
- **Goal:** achieve better accuracy at size 3 [2]

$$dx = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} \tag{3}$$

$$dy = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix} \tag{4}$$

Laplacian Kernel

- Approximates second order derivative of the image

Laplacian Kernel
- Approximates second order derivative of the image

1. In physics, first order derivative of position of an object corresponds to the *rate of change of position* a.k.a., *velocity*. Therefore, the second order derivative of position is *acceleration*. You may be able to apply this as an analogy for what the second derivative tells us about the image.

Laplacian Kernel

- Approximates second order derivative of the image
  - **First Order Derivative:** rate of change, min/max when zero

Laplacian Kernel
- Approximates second order derivative of the image
  - **First Order Derivative:** rate of change, min/max when zero

1. In physics, first order derivative of position of an object corresponds to the *rate of change of position* a.k.a., *velocity*. Therefore, the second order derivative of position is *acceleration*. You may be able to apply this as an analogy for what the second derivative tells us about the image.

Laplacian Kernel

- Approximates second order derivative of the image
  - **Second Order Derivative:** rate of change, of *the rate of change*

Laplacian Kernel

- Approximates second order derivative of the image
  - **Second Order Derivative:** rate of change, of *the rate of change*

1. In physics, first order derivative of position of an object corresponds to the *rate of change of position* a.k.a., *velocity*. Therefore, the second order derivative of position is *acceleration*. You may be able to apply this as an analogy for what the second derivative tells us about the image.

Laplacian Kernel

- Works in X *and* Y direction

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{5}$$

Laplacian Kernel
Works in X *and* Y direction
$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{5}$$

Derivative Kernel Comparison

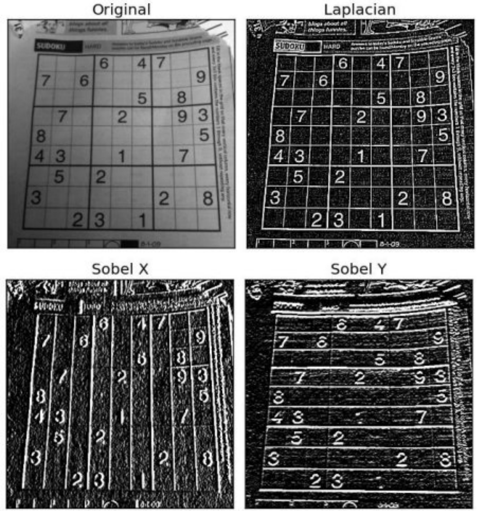
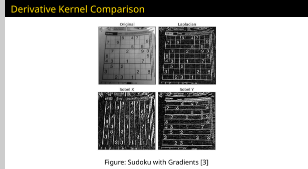


Figure: Sudoku with Gradients [3]


Derivative Kernel Comparison
Figure: Sudoku with Gradients [3]

Using Derivative Kernels

- What do we do with 3 color channels?

Using Derivative Kernels

- What do we do with 3 color channels?

Using Derivative Kernels

- ► What do we do with 3 color channels?
- ► Rather have gradient for one channel

Using Derivative Kernels
► What do we do with 3 color channels?
► Rather have gradient for one channel

Using Derivative Kernels

- What do we do with 3 color channels?
- Grayscale!

Using Derivative Kernels

- What do we do with 3 color channels?
- Grayscale!

Pre-processing

Convert to grayscale with $cv.cvtColor(img, cv.COLOR\_BGR2GRAY)$

Pre-processing

Convert to grayscale with $cv.cvtColor(img, cv.COLOR\_BGR2GRAY)$

Pre-processing

Convert to grayscale with $cv.cvtColor(img, cv.COLOR\_BGR2GRAY)$

- Now gradient looking at intensity in one channel

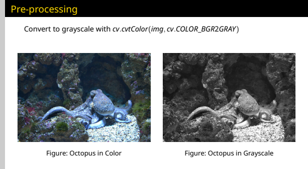
Pre-processing
Convert to grayscale with $cv.cvtColor(img, cv.COLOR\_BGR2GRAY)$
- Now gradient looking at intensity in one channel

Pre-processing

Convert to grayscale with $cv.cvtColor(img, cv.COLOR\_BGR2GRAY)$

Figure: Octopus in Color

Figure: Octopus in Grayscale



Pre-Processing

- Gradient can be “sensitive” or choppy
- How to make gradient smoother?
  1. Blurring/Smoothing
  2. Gaussian is popular

Pre-Processing

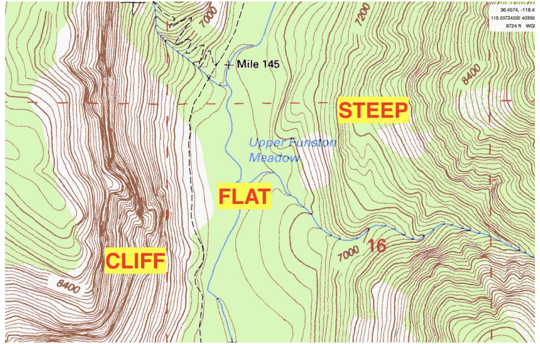
- Gradient can be “sensitive” or choppy
- How to make gradient smoother?
  1. Blurring/Smoothing
  2. Gaussian is popular

Code

```
cv.GaussianBlur()
cv.Sobel()
cv.Scharr()
cv.Laplacian()
```


Code
cv.GaussianBlur()
cv.Sobel()
cv.Scharr()
cv.Laplacian()

Isophotes and Ridges

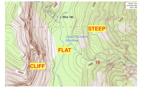


Isophote

A curve on a surface connecting points of equal brightness. [4]

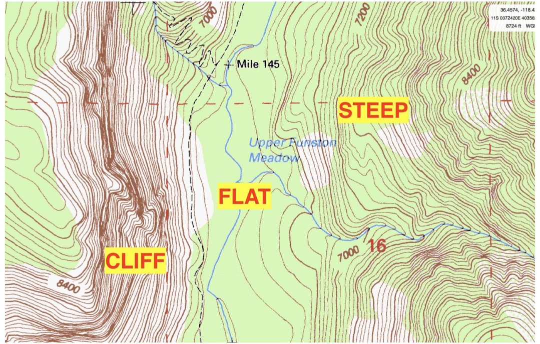
- Isophotes are like contour lines
  - brightness = elevation
  - used for optical evaluation of surface smoothness (CAD)

Isophotes and Ridges



Isophote
A curve on a surface connecting points of equal brightness. [4]
- Isophotes are like contour lines
  - brightness = elevation
  - used for optical evaluation of surface smoothness (CAD)

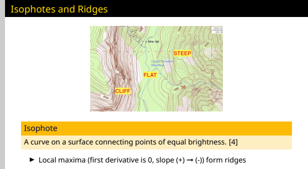
Isophotes and Ridges



Isophote

A curve on a surface connecting points of equal brightness. [4]

- Local maxima (first derivative is 0, slope (+) → (-)) form ridges


Isophotes and Ridges
Isophote
A curve on a surface connecting points of equal brightness. [4]
Local maxima (first derivative is 0, slope (+) → (-)) form ridges

Bibliography I

[1] *Image gradient*, in *Wikipedia*, Oct. 9, 2023. [Online]. Available: `https://en.wikipedia.org/w/index.php?title=Image_gradient&oldid=1179280667` (visited on 09/05/2024).

[2] "OpenCV: Sobel Derivatives," (), [Online]. Available: `https://docs.opencv.org/4.x/d2/d2c/tutorial_sobel_derivatives.html` (visited on 09/04/2024).

[3] "OpenCV: Image Gradients," (), [Online]. Available: `https://docs.opencv.org/4.x/d5/d0f/tutorial_py_gradients.html` (visited on 09/04/2024).

[4] *Isophote*, in *Wikipedia*, Nov. 18, 2023. [Online]. Available: `https://en.wikipedia.org/w/index.php?title=Isophote&oldid=1185683679` (visited on 09/02/2024).

Bibliography I

[1] *Image gradient*, in *Wikipedia*, Oct. 9, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Image_gradient&oldid=1179280667 (visited on 09/05/2024).

[2] "OpenCV: Sobel Derivatives," (), [Online]. Available: https://docs.opencv.org/4.x/d2/d2c/tutorial_sobel_derivatives.html (visited on 09/04/2024).

[3] "OpenCV: Image Gradients," (), [Online]. Available: https://docs.opencv.org/4.x/d5/d0f/tutorial_py_gradients.html (visited on 09/04/2024).

[4] *Isophote*, in *Wikipedia*, Nov. 18, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Isophote&oldid=1185683679 (visited on 09/02/2024).