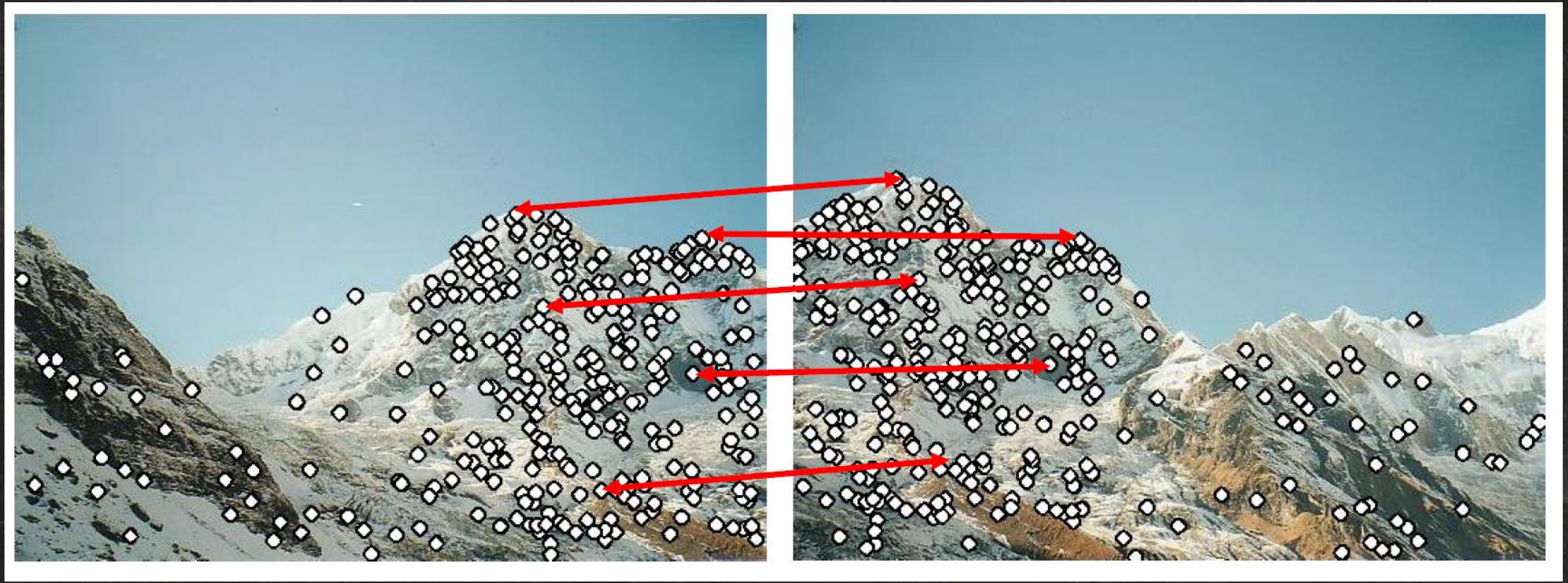
Lecture-05

Feature Extraction

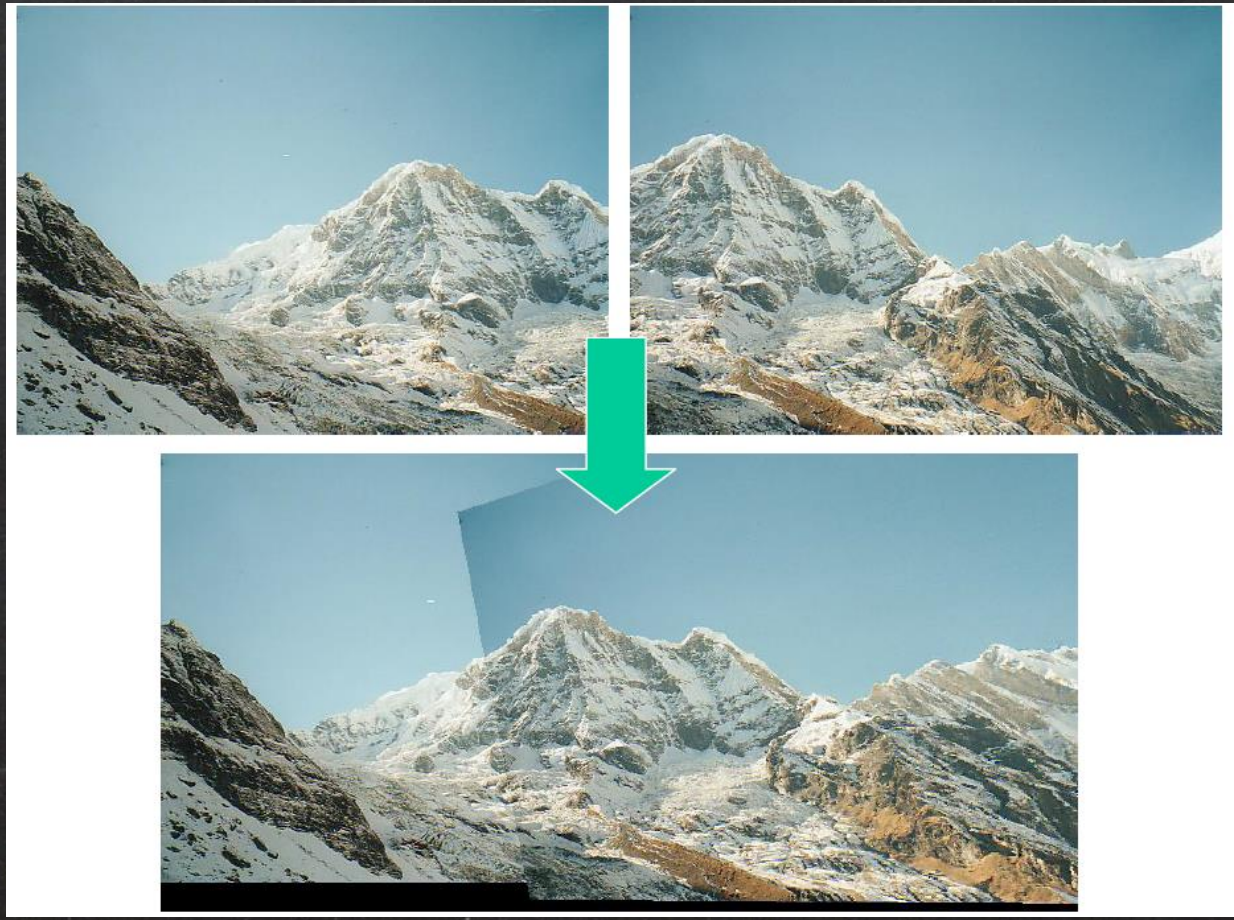
Introduction

- Feature detection and matching are an essential component of many computer vision applications.
- kinds of features
  - key-point features or interest points
  - Edges

key-point features



key-point features usage Example



panorama

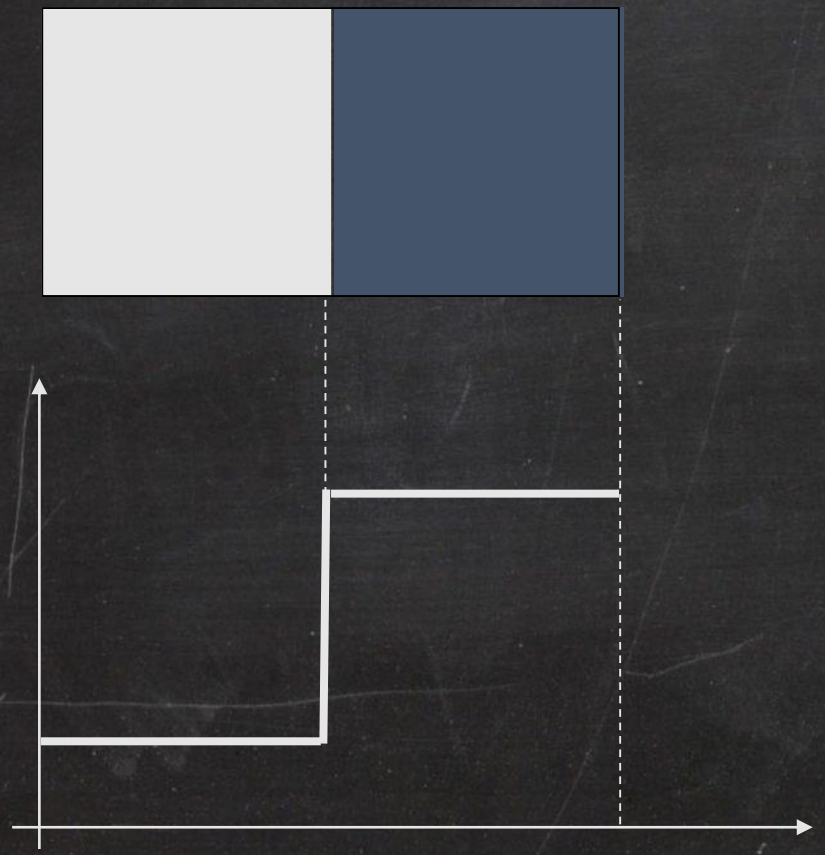
Edge Detection?

"The ability to measure gray-level transitions in a meaningful way."

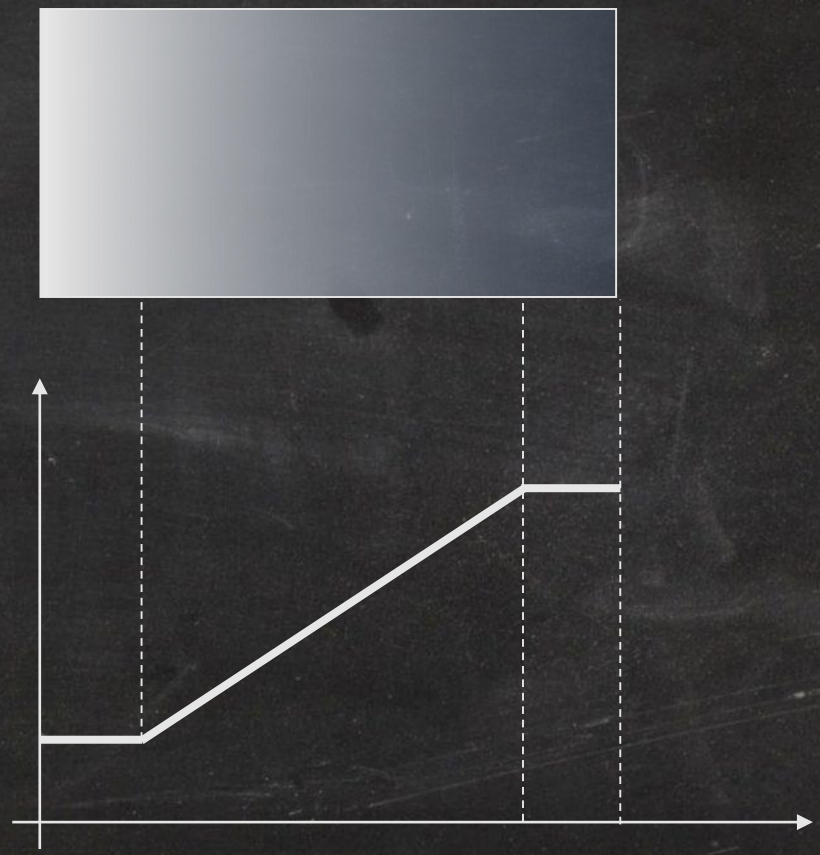
- (R.C. Gonzales & R. E. Woods – Digital Image Processing, 2nd Edition, Prentice-Hall, 2001)

Gray-Level Transition

Ideal

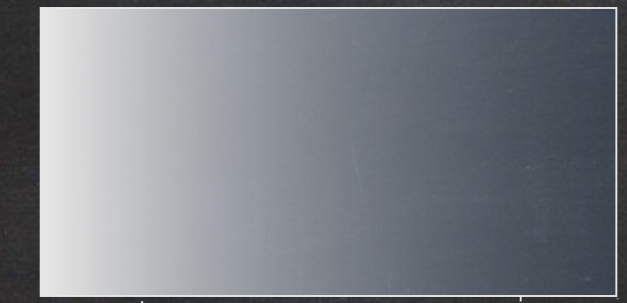


Ramp

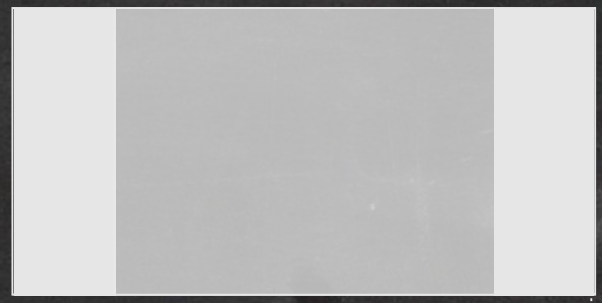


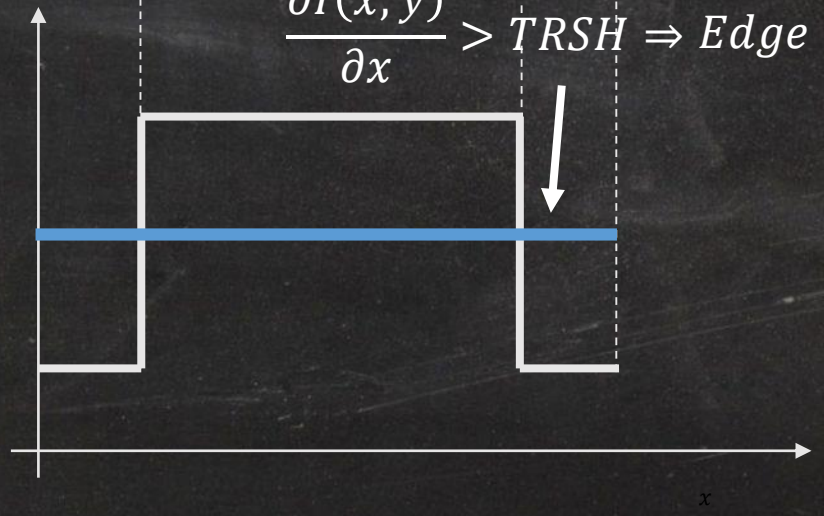
Detecting the Edge (1)

Original



First Derivative



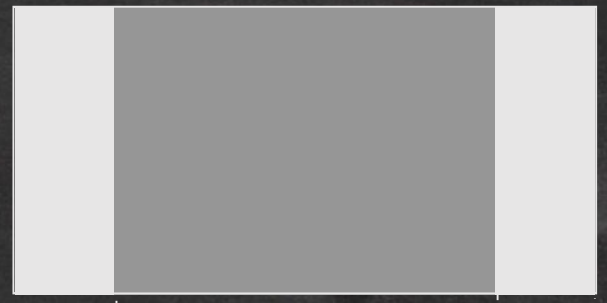


$$\frac{\partial I(x,y)}{\partial x} > TRSH \Rightarrow Edge \quad Detected$$

Detecting the Edge (2)

Original

First Derivative



$I(x,y)$

$\frac{\partial I(x,y)}{\partial x}$

TRSH

$\frac{\partial I(x,y)}{\partial x} < TRSH \Rightarrow Edge \quad Not \quad Detected$

Gradient Operators

- The gradient of the image I(x,y) at location (x,y), is the vector:

$$\overline{\nabla I} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{bmatrix}$$

- The magnitude of the gradient:

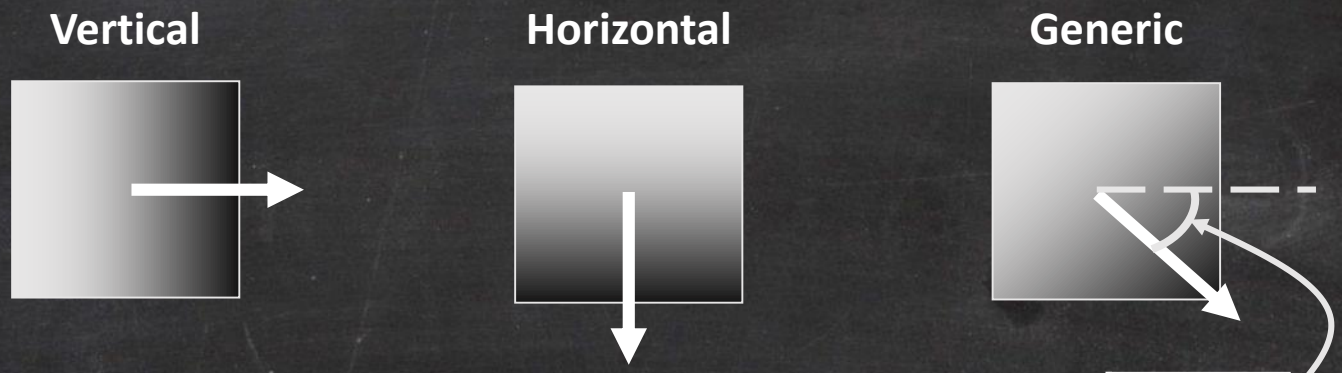
$$\nabla I = \left\| \overline{\nabla I} \right\| = \sqrt{\left[G_x^2 + G_y^2 \right]}$$

- The direction of the gradient vector:

$$\theta(x,y) = \tan^{-1}\left(\frac{G_x}{G_y} \right)$$

The Meaning of the Gradient

- It represents the direction of the strongest variation in intensity



Edge Strength:

$\|\overline{\nabla I}\| = G_x$

$\|\overline{\nabla I}\| = G_y$

$\|\overline{\nabla I}\| = \sqrt{[G_x^2 + G_y^2]}$

Edge Direction:

$\theta(x, y) = 0$

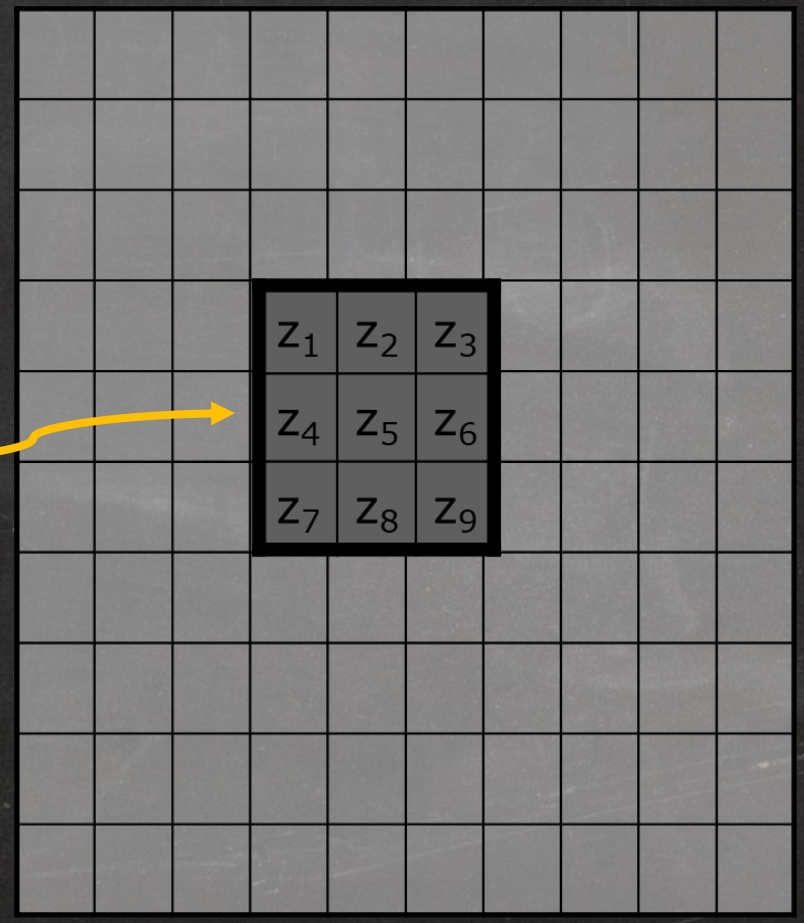
$\theta(x, y) = -\frac{\pi}{2}$

$\theta(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$

The direction of the edge at location (x,y) is perpendicular to the gradient vector at that point

Calculating the Gradient

- For each pixel the gradient is calculated, based on a 3x3 neighborhood around this pixel.



The Sobel Edge Detector

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

$$G_x \approx (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y \approx (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

The Prewitt Edge Detector

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

$$G_x \approx (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y \approx (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

The Roberts Edge Detector

0	0	0
0	-1	0
0	0	1

$$G_x \approx z_9 - z_5$$

0	0	0
0	0	-1
0	1	0

$$G_y \approx z_8 - z_6$$

The Roberts Edge Detector is in fact a 2x2 operator

The Canny Method

Two Possible Implementations:

1. The image is convolved with a Gaussian filter before gradient evaluation

$$h(r) = -e^{-\frac{r^2}{2\sigma^2}}$$

$$r = \sqrt{x^2 + y^2}$$

2. The image is convolved with the gradient of the Gaussian Filter.

The Edge Detection Algorithm

- The gradient is calculated (using any of the four methods described in the previous slides), for each pixel in the picture.
- If the absolute value exceeds a threshold, the pixel belongs to an edge.
- The Canny method uses two thresholds, and enables the detection of two edge types: strong and weak edge. If a pixel's magnitude in the gradient image, exceeds the high threshold, then the pixel corresponds to a strong edge. Any pixel connected to a strong edge and having a magnitude greater than the low threshold corresponds to a weak edge.