perception

low level operations

computation of optical flow

texture analysis

edge detection

low level operations are good candidates for implementation in parallel hardware

optical flow is the apparent motion in an image

mid/high level operations

object/face detection/recognition can be considered a "high-level" operation

segmenting an image into regions can be considered a "mid-level" operation

edges are straight lines or curves in the image across which there's significant change in image brightness

we detect edges to produce a more abstract representation of an input image

edges correspond to the locations in image where the brightness undergoes a sharp change

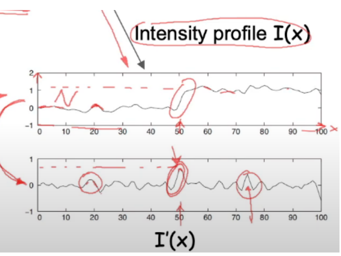
Theorem: For any functions f

(g*f)′ is equal to the convolu

- Instead of smoothing the
  - Convolve the image with t

Principle: Build a profile of image brightness **I(x)** along a dimension, and differentiate the image to look for places where the magnitude of the derivative **I′(x)** is large.

- There is a peak at around x = 50
- There is also another peak at x = 75 (noise)



aro
po
in

- Consid
  - Ste
    cer
    (x_0 -
  - Ste
    eac
    (x, y
  - Ste

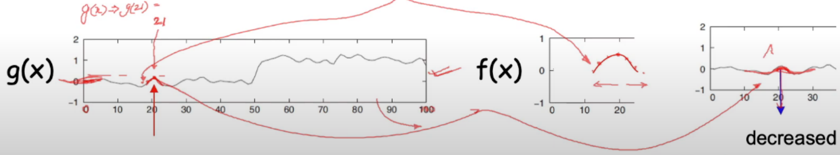
Convolution for edge detection/smoothing - 1D/2D

- A function **h** is the convolution of two functions **f** and **g** (denoted by **f * g**) if we have

$$h(x) = (f * g)(x) = \sum_{u=-\infty}^{+\infty} f(u)\, g(x-u)$$

small

- Usually **g(x)** is the image matrix and **f(x)** is the Gaussian filter. Instead of running **u** from **-inf** to **+inf** we usually run it from -3 to +3.



2D:

$$h(x,y) = (f * g)(x,y) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} f(u,v)\, g(x-u, y-v)$$

Theorem: For any functions **f** and **g**, the derivative of the convolution, **(g*f)′** is equal to the convolution with the derivative **g * (f′)**.

- Instead of smoothing the image and then differentiating
  - Convolve the image with the derivative of the smoothing function

Operation

Identity

optical flow

optical flow is the apparent motion in an image

optical flow encodes useful information about the scene structure

it describes the direction and speed of motion of features in the image

to calculate optical flow

(x0,y0) is the pixel in the current frame(at time =t)

and **g**, the derivative of the convolution,

ution with the derivative **g * (f′)**.

e image and then differentiating

the derivative of the smoothing function

Operation
Identity

image patches
und corresponding
oints have similar
ntensity patterns

der a block of pixels centered at pixel p, (x_0, y_0), at time t

p 1: We compare this block of pixels with many pixel blocks ntered at various candidate pixels $(x_0 + D_x, y_0 + D_y)$ at time $t + D_t$

$+ D_x, y_0 + D_y)$ is the center of a candidate block at time D_t

$Dx = 5$
$Dy = 5$

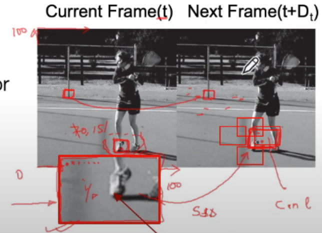
p 2: Calculate Sum of Squared Differences (SSD) of intensities for ch of the many blocks

$$\mathrm{SSD}(D_x, D_y) = \sum_{(x,y)} (I(x,y,t) - I(x + D_x, y + D_y, t + D_t))^2$$

) ranges over pixels in the block centered at (x_0, y_0)

p 3: Find the (D_x, D_y) that minimizes SSD -10 $+10$

Current Frame(t) Next Frame(t+D_t)



Center of the block pixel p, (x_0, y_0)

segmentation of images

segmentation is the process of breaking an mage into regions of similar pixels

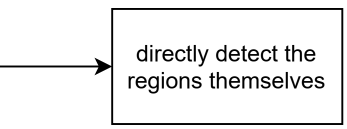
theres two approaches to segmentation

detect the boundaries of the regions and then compute regions

segmentation is based on low level, local attributes such as brightness and color cannot be expected to deliver the final correct boundaries of all the objects in the scene

to reliably find object boundaries we need high-level knowledge of the likely kinds of objects in the scene

representing this knowledge is a topic of active research


directly detect the regions themselves