Welcome to Computer Vision

Computer Vision

Dr. Muhammad Tahir

DEPARTMENT OF COMPUTER SCIENCE,

FAST-NUCES, Peshawar

Course Details

LECTURES: Monday & Wednesday

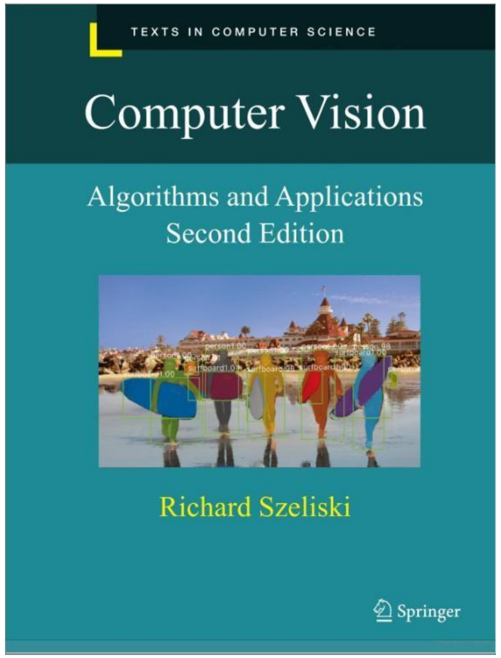
TIMINGS:

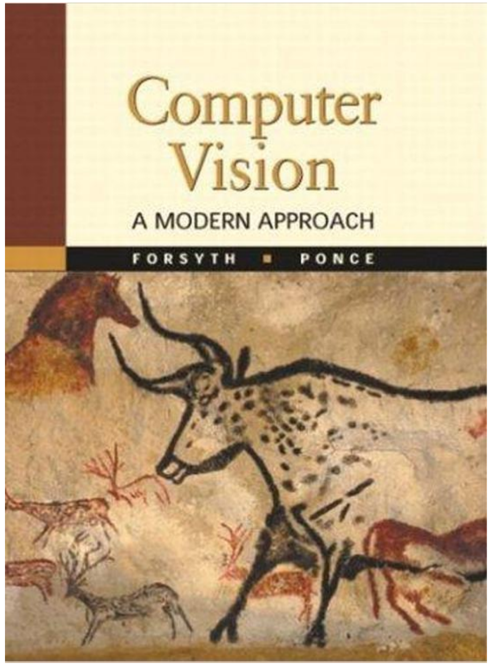
9:30 am – 11:00 am

MY OFFICE:

OFFICE HOURS:

EMAIL: m.tahir@nu.edu.pk





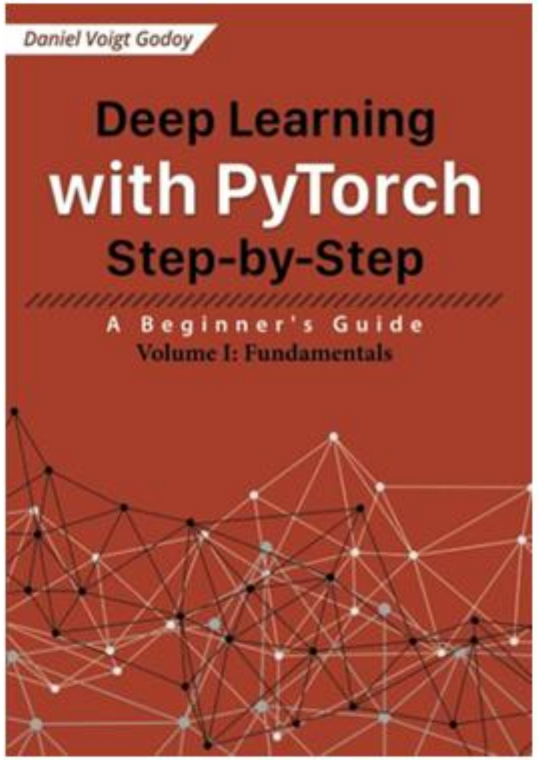
References

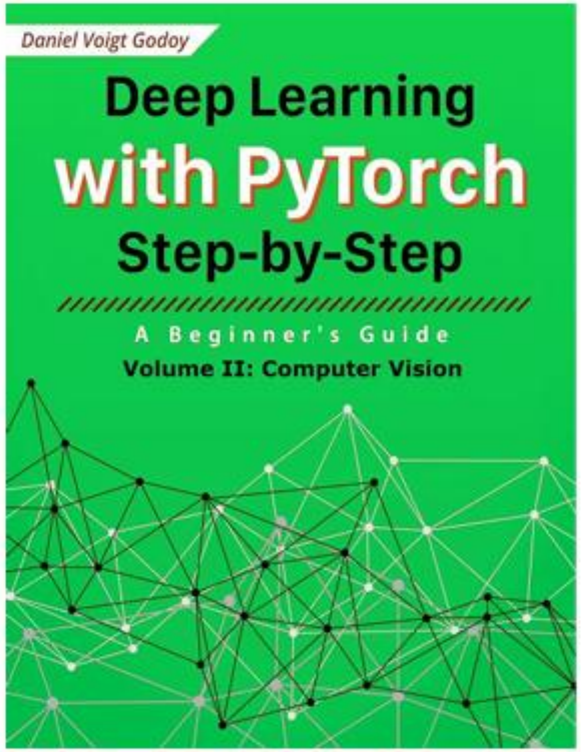
The material in these slides are based on:

1. Rick Szeliski's book: Computer Vision: Algorithms and Applications
2. Forsythe and Ponce: Computer Vision: A Modern Approach

Recommended Books

Deep Learning with PyTorch Step-by-Step by Daniel Voigt Godoy





Course Learning Outcomes

No	CLO (Tentative)	Domain	Taxonomy Level	PLO
1	Understanding basics of Computer Vision: algorithms, tools, and techniques	Cognitive	2	
2	Develop solutions for image/video understanding and recognition	Cognitive	3	
3	Design solutions to solve practical Computer Vision problems	Cognitive	3	

Outline

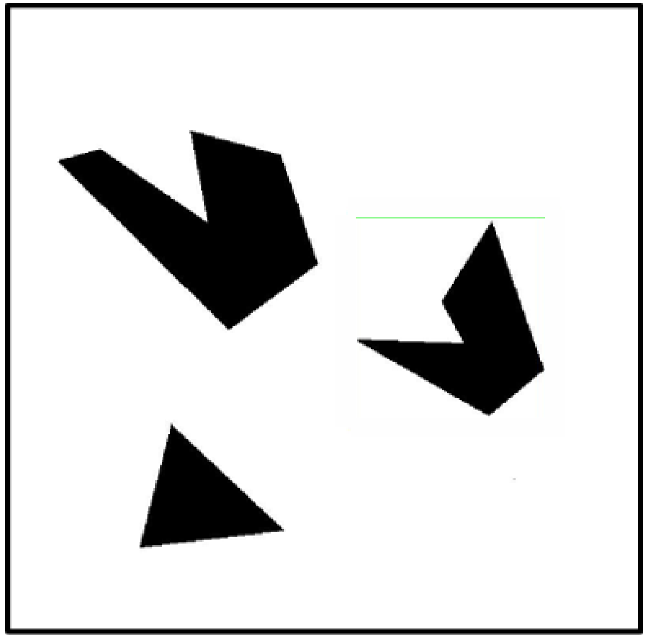
Feature Matching

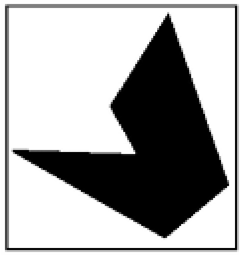
Filters for Feature Detection

- we used filters to reduce noise or enhance contours
- However, filters can also be used to detect “features”
- Goal: reduce amount of data to process in later stages, discard redundancy to preserve only what is useful (leads to lower bandwidth and memory storage)
  - Edge detection (we have seen this already; edges can enable line or shape detection)
  - Template matching
  - Keypoint detection

Filters for Template Matching

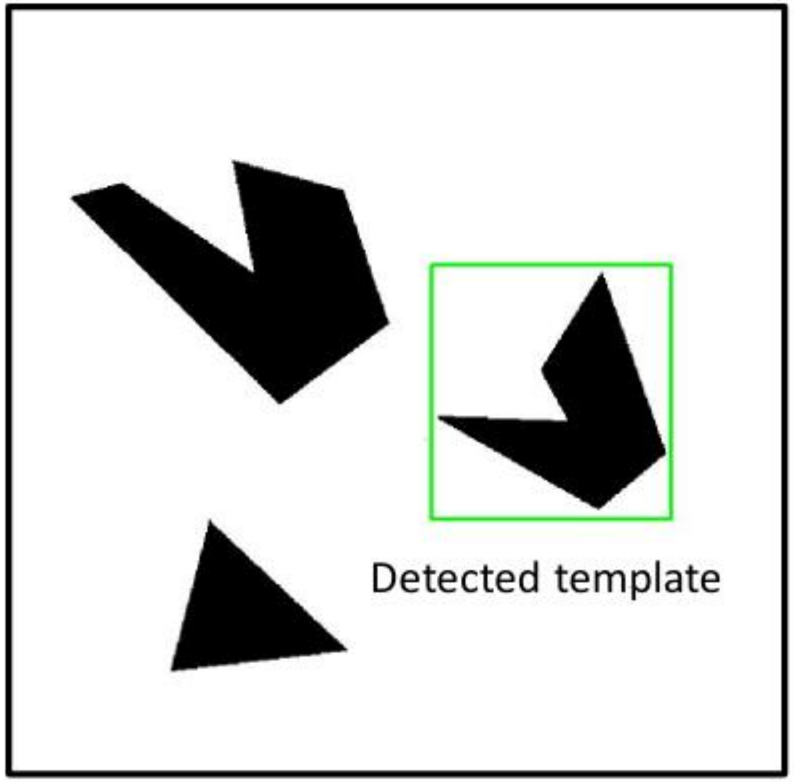
- Find locations in an image that are similar to a template
- If we look at filters as **templates**, we can use correlation (like convolution but without flipping the filter) to detect these locations

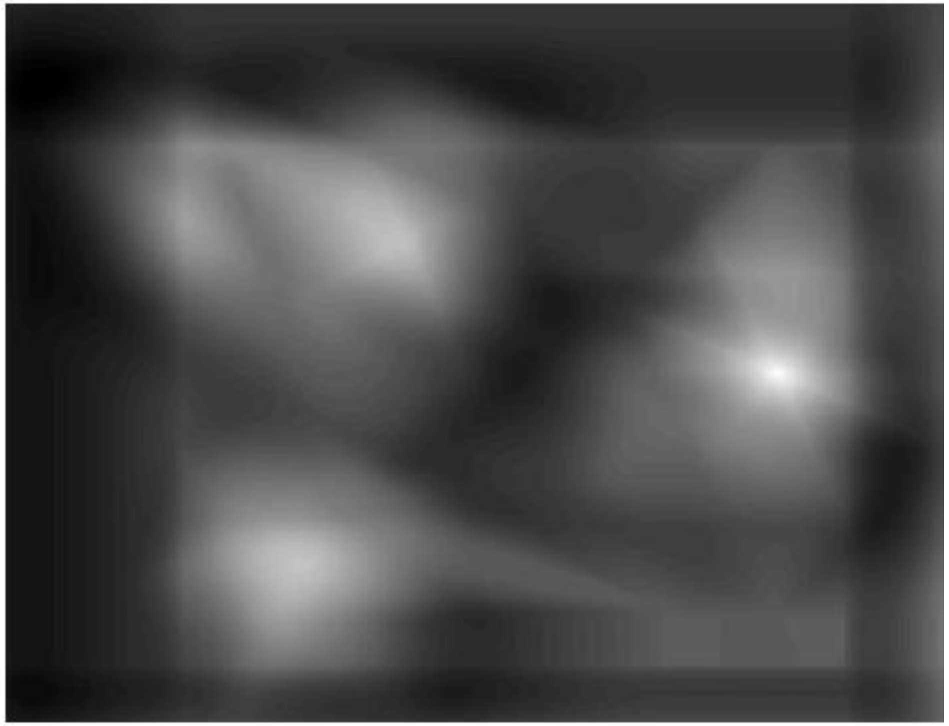




Template

Filters for Template Matching

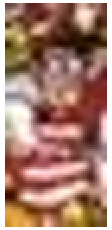




Correlation map

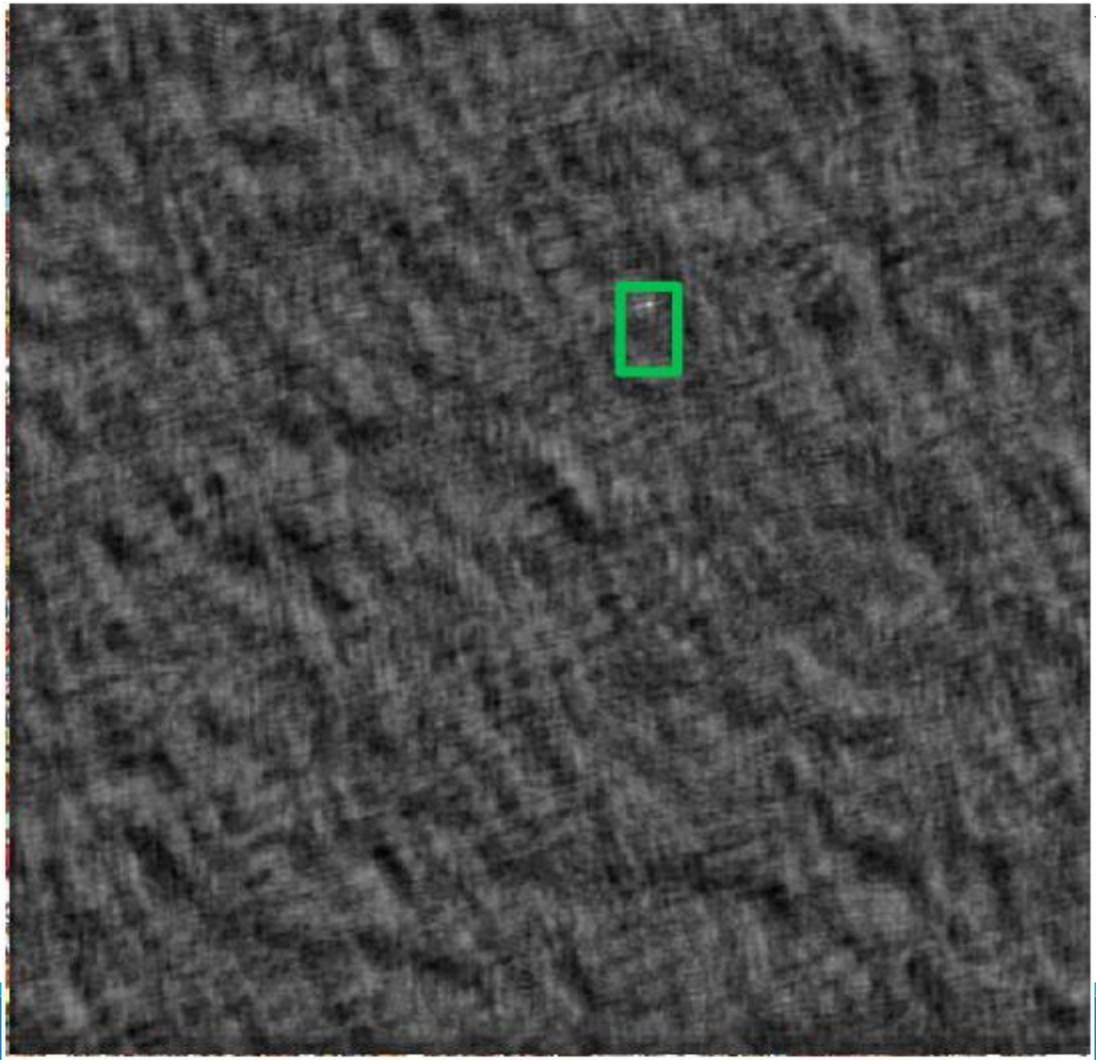
Where's Waldo?

Scene

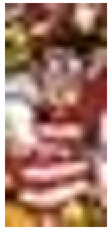


Template

Where’s Waldo?



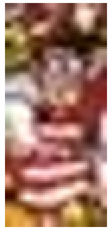
Scene



Template

Where's Waldo?

Scene



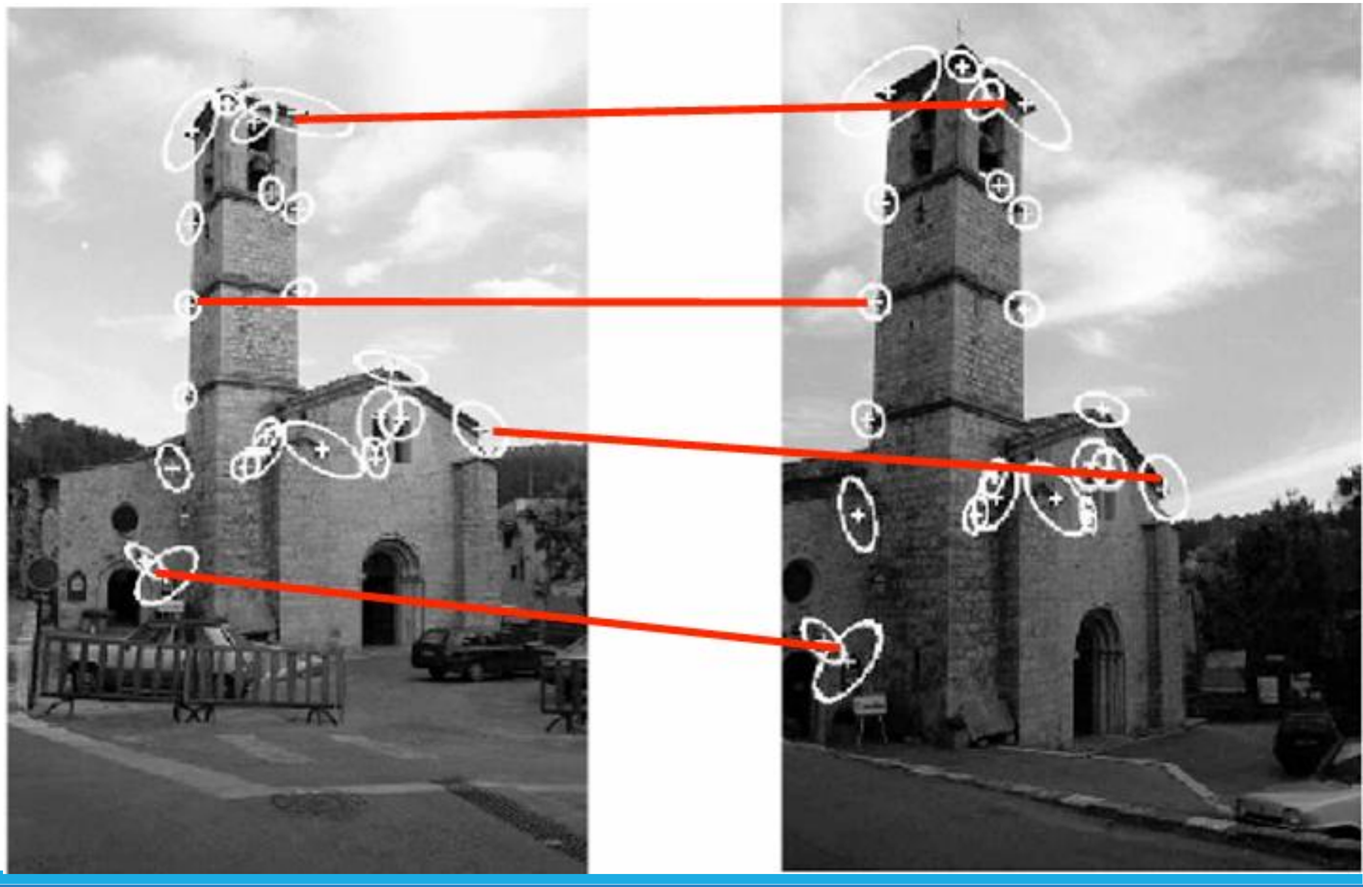
Template

Summary of filters

- Smoothing filter:
  - has positive values
  - sums to 1 → preserve brightness of constant regions
  - removes “high-frequency” components: “low-pass” filter
- Derivative filter:
  - has opposite signs used to get high response in regions of high contrast
  - sums to 0 → no response in constant regions
  - highlights “high-frequency” components: “high-pass” filter
- Filters as templates
  - Highest response for regions that “look similar to the filter”

Matching Problem

Vision tasks such as stereo and motion estimation require finding corresponding features across two or more views.

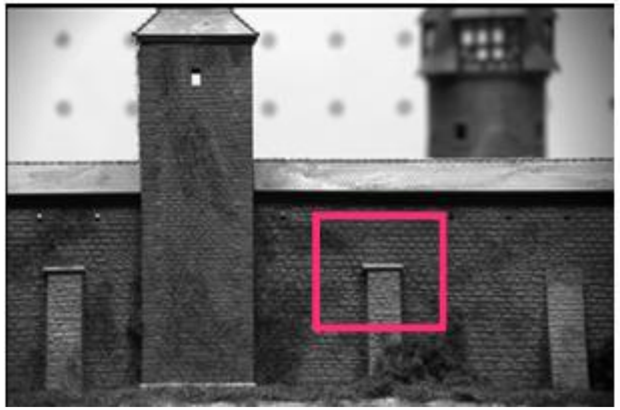


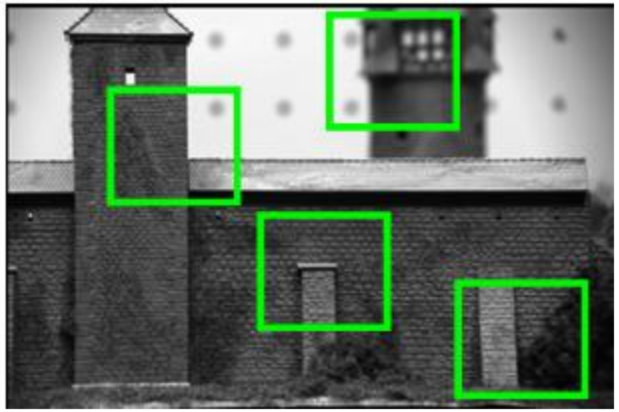
Patch Matching

Elements to be matched are image patches of fixed size

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

Not all Patches are Created Equal!

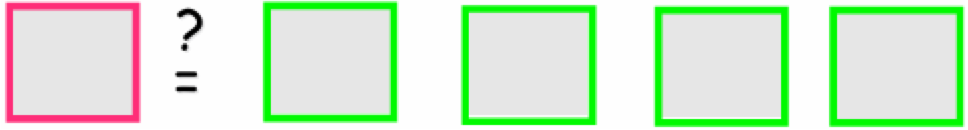




Intuition: this would be a good patch for matching, since it is very distinctive (there is only one patch in the second frame that looks similar).

Not all Patches are Created Equal!

Intuition: this would be a BAD patch for matching, since it is not very distinctive (there are many similar patches in the second frame)



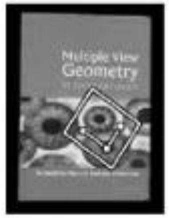
Template Matching

- What if the template is not identical to the object we want to detect?
- Template Matching will only work if scale, orientation, illumination, and, in general, the appearance of the template and the object to detect are very similar. What about the pixels in template background (object-background problem)?

Scene

Template

Scene

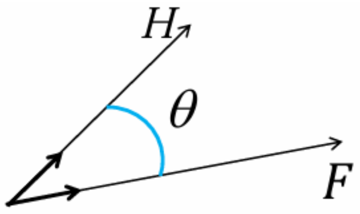


Template

Correlation as Scalar Product

We consider images H and F as vectors and express the correlation between them as

$$\langle H, F \rangle = \|H\| \|F\| \cos\theta$$



- If we use Normalized Cross Correlation (NCC) (highest complexity), we consider the unit vectors of H and F, hence, we measure their similarity based on the angle.
- For identical vectors one gets NCC = 1: this is why one can use NCC as a similarity measure. Note that NCC is invariant to linear intensity changes! It holds

$$\cos\theta = \frac{\langle H, F \rangle}{\|H\| \|F\|} = \frac{\sum_{u=-k}^{k} \sum_{v=-k}^{k} H(u,v) F(u,v)}{\sqrt{\sum_{u=-k}^{k} \sum_{v=-k}^{k} H(u,v)^2} \sqrt{\sum_{u=-k}^{k} \sum_{v=-k}^{k} F(u,v)^2}}$$

Other Similarity measures

- **Sum of Absolute Differences** (**SAD**) (used in optical mice)

$$SAD = \sum_{u=-k}^{k} \sum_{v=-k}^{k} \left| H(u,v) - F(u,v) \right|$$

- **Sum of Squared Differences** (**SSD**)

$$SSD = \sum_{u=-k}^{k} \sum_{v=-k}^{k} \left(H(u,v) - F(u,v) \right)^2$$

- **Normalized Cross Correlation** (**NCC**): takes values between -1 and +1 (+1 = identical)

$$NCC = \frac{\sum_{u=-k}^{k} \sum_{v=-k}^{k} H(u,v) F(u,v)}{\sqrt{\sum_{u=-k}^{k} \sum_{v=-k}^{k} H(u,v)^2} \sqrt{\sum_{u=-k}^{k} \sum_{v=-k}^{k} F(u,v)^2}}$$

Zero-mean SAD, SSD, NCC

To account for the difference in mean of the two images (typically caused by illumination changes), we subtract the mean value of each image:

- **Zero-mean Sum of Absolute Differences (ZSAD)** (used in optical mice)

$$ZSAD = \sum_{u=-k}^{k} \sum_{v=-k}^{k} \left| \left(H(u,v) - \mu_H \right) - \left(F(u,v) - \mu_F \right) \right|$$

- **Zero-mean Sum of Squared Differences (ZSSD)**

$$ZSSD = \sum_{u=-k}^{k} \sum_{v=-k}^{k} \left(\left(H(u,v) - \mu_H \right) - \left(F(u,v) - \mu_F \right) \right)^2$$

Are these invariant to affine illumination changes?

$$\mu_H = \frac{\sum_{u=-k}^{k} \sum_{v=-k}^{k} H(u,v)}{(2N+1)^2}$$

- **Zero-mean Normalized Cross Correlation (ZNCC)**

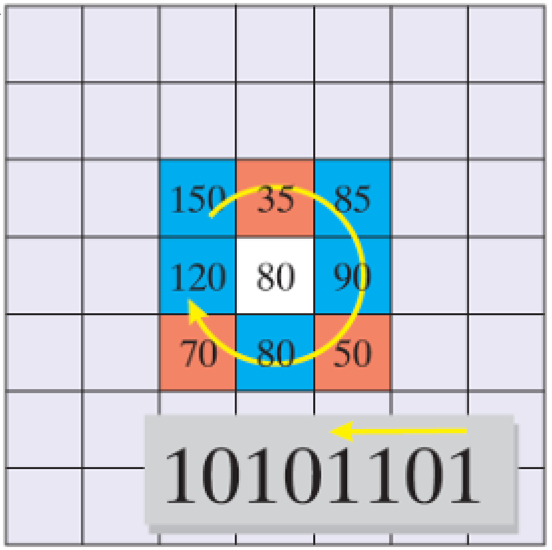
$$\mu_F = \frac{\sum_{u=-k}^{k} \sum_{v=-k}^{k} F(u,v)}{(2N+1)^2}$$

$$ZNCC = \frac{\sum_{u=-k}^{k} \sum_{v=-k}^{k} \left(H(u,v) - \mu_H \right) \left(F(u,v) - \mu_F \right)}{\sqrt{\sum_{u=-k}^{k} \sum_{v=-k}^{k} \left(H(u,v) - \mu_H \right)^2} \sqrt{\sum_{u=-k}^{k} \sum_{v=-k}^{k} \left(F(u,v) - \mu_F \right)^2}}$$

ZNCC is invariant to affine intensity changes
$I'(x,y) = \alpha I(x,y) + \beta$

Census Transform

- Maps an image patch to a bit string:
  - if a pixel is greater than the centre pixel its corresponding bit is set to 1, else to 0
  - For a $w \times w$ window the string will be $w^2 - \mathbf{1}$ bits long
- The two bit strings are compared using the Hamming distance, which is the number of bits that are different.
- This can be computed by counting the number of 1s in the Exclusive-OR (XOR) of the two bit strings
- **Advantages**
  - More robust to object-background problem
  - No square roots or divisions are required, thus very efficient to implement, especially on FPGA
  - Intensities are considered relative to the centre pixel of the patch making it invariant to monotonic intensity changes



Thank you