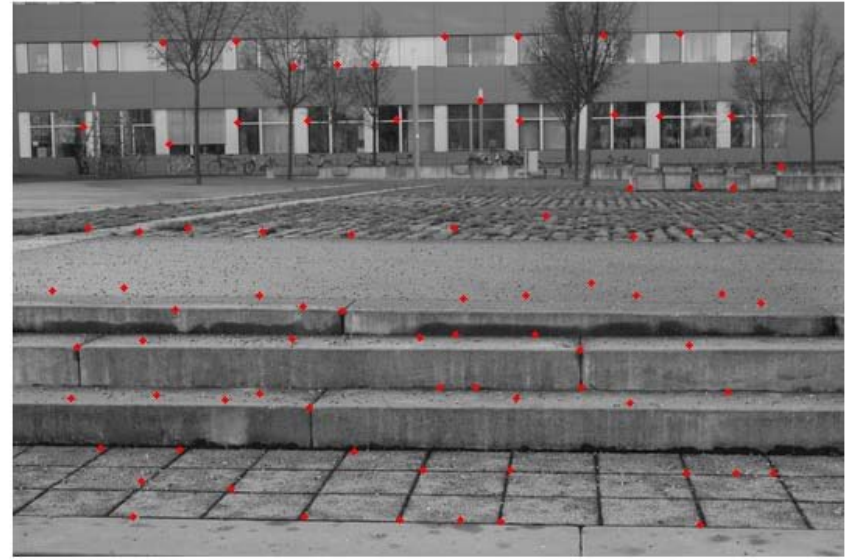
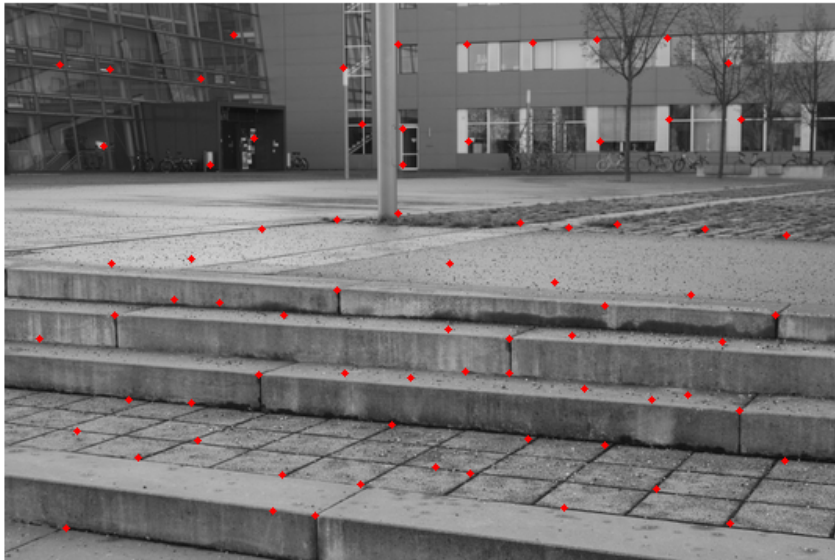


# Image Stitching

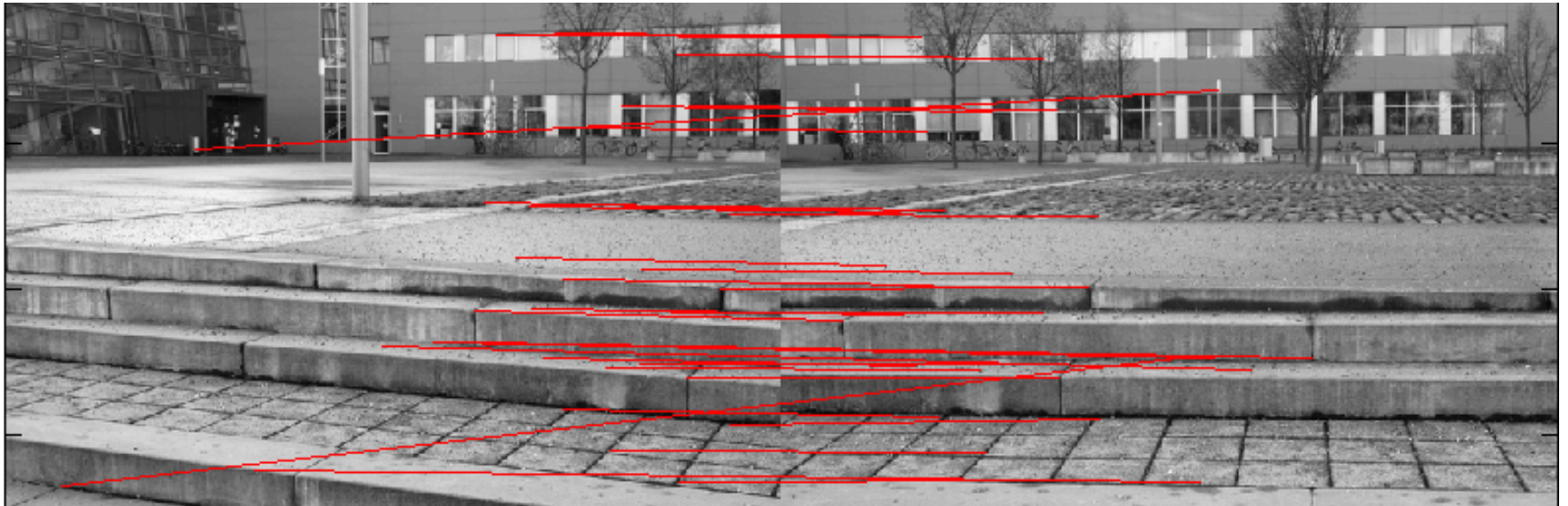
## *Automatic Feature Detection and Correspondence Estimation*

Pierre Georgel and Darko Zikic

# Feature Extraction: Harris Corners



# Matching of the Corresponding Features



## Harris Corner extraction 0/3

- To decrease noise, we process the original input image  $I'$ , by filtering it with a Gaussian kernel with standard deviation  $\sigma$ , and we call the result  $I$ 
  - Matlab commands: `fspecial` and `imfilter` (or `filter2` or `conv2`)
  - When setting up a filter mask, keep in mind that the size of the patch should be at least odd and at least  $1+2*3*\text{ceil}(\text{sigma})$  to properly approximate the Gaussian (Why? See [http://en.wikipedia.org/wiki/Empirical\\_rule](http://en.wikipedia.org/wiki/Empirical_rule) )
- The further processing is done on  $I$  !!!
- For the image  $I$ , we define the derivatives as  $I_x$  and  $I_y$
- The derivatives are computed by central differences along the respective dimensions, that is

$$I_x(x, y) = \frac{\partial I(x, y)}{\partial x} = \frac{I(x+1, y) - I(x-1, y)}{2}$$

$$I_y(x, y) = \frac{\partial I(x, y)}{\partial y} = \frac{I(x, y+1) - I(x, y-1)}{2}$$

## Harris Corner extraction 1/3

- Now we can define the Cornerness Function (aka 2D Structure Tensor)  
compare also:

[http://en.wikipedia.org/wiki/Corner\\_detection#The\\_Harris\\_26\\_Stephens\\_26\\_Plessey\\_corner\\_detection\\_algorithm](http://en.wikipedia.org/wiki/Corner_detection#The_Harris_26_Stephens_26_Plessey_corner_detection_algorithm)

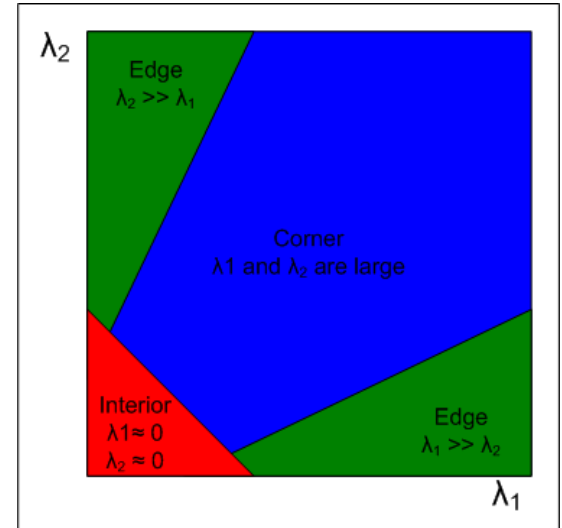
$$\mathbf{C}(x,y) = \sum_{u=-r}^r \sum_{v=-r}^r w(u,v) \begin{bmatrix} I_x(x+u,y+v)^2 & I_x(x+u,y+v)I_y(x+u,y+v) \\ I_y(x+u,y+v)I_x(x+u,y+v) & I_y(x+u,y+v)^2 \end{bmatrix}$$

- $r$  is a radius of a patch which is used for averaging the matrices
- $w$  is the averaging kernel, which can be for example uniform or a Gaussian

## Harris Corner extraction 2/3

- Lets take a look at the eigenvalues  $\lambda_1, \lambda_2$  of  $\mathbf{C}$

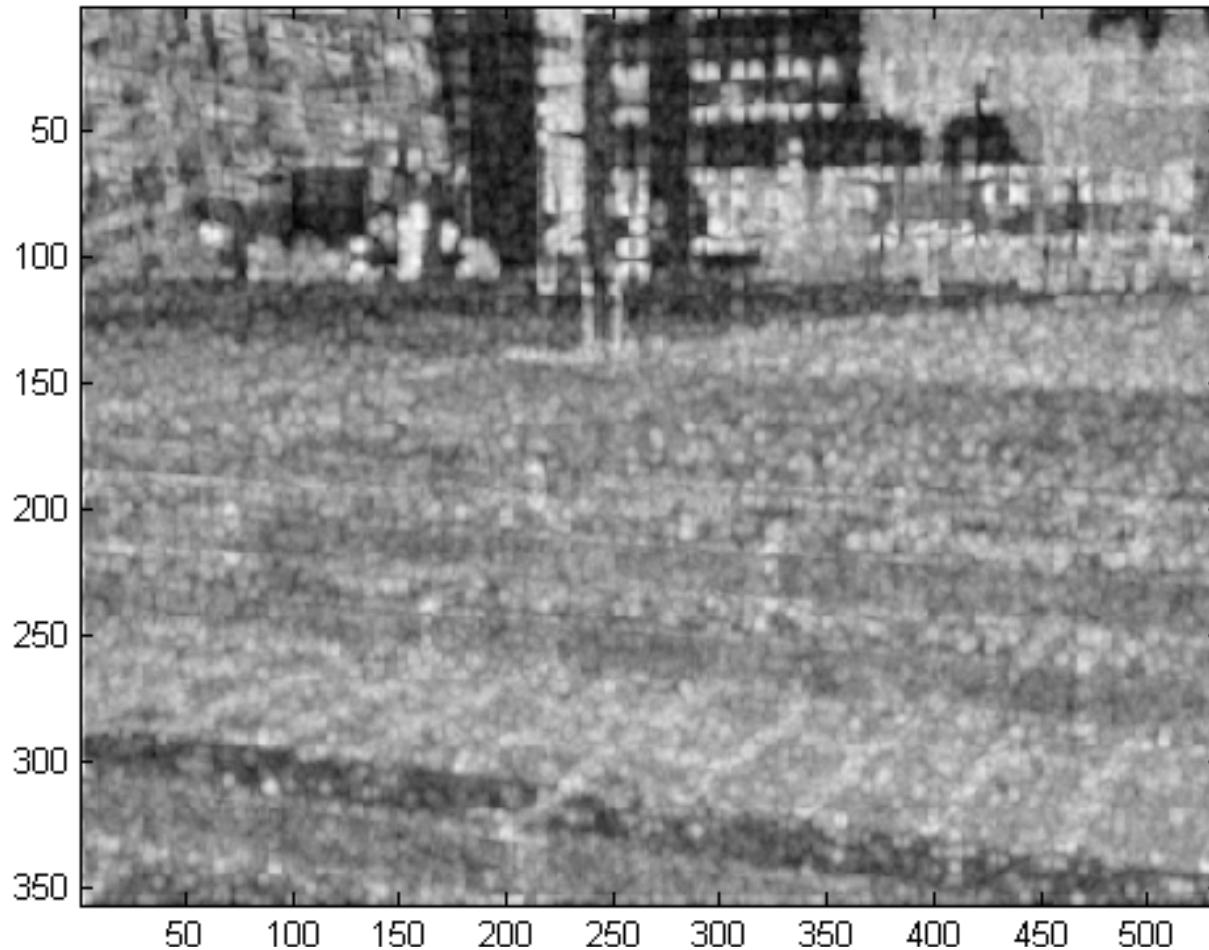
$$\mathbf{C} = \mathbf{U}^T \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \mathbf{U}$$



- Since we don't want to compute the eigenvalues we define

$$\mathbf{D} = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \frac{\det(\mathbf{C})}{\text{trace}(\mathbf{C})}$$

## Harris Corner extraction 3/3



In order to select only points which are real corners,  
**NON MAXIMUM SUPPRESSION** has to be performed.

## Correlation Based Matching of the Corner Features (Correspondence Estimation)

- Let  $p_j$  be a patch of size  $s \times s$ , around the a feature  $f_j$ .
- We can measure the similarity of two patches  $p_j$  and  $p_k$  by computing the correlation between the two patches by

$$\text{COR}(p_j, p_k) = \frac{\sum_{x,y \in [1,s]} (p_j(x,y) - \bar{p}_j) (p_k(x,y) - \bar{p}_k)}{\sqrt{\sum_{x,y \in [1,s]} (p_j(x,y) - \bar{p}_j)^2 \cdot \sum_{x,y \in [1,s]} (p_k(x,y) - \bar{p}_k)^2}}$$

- Here, the mean values of the patches are denoted by  $\bar{p}_j, \bar{p}_k$
- COR gives values between from  $[-1,+1]$ , where higher values indicate larger similarities
- Corresponding features can be found by computing the COR on patches between all possible feature pairs, and taking the maximum values
- To get only correct correspondences, only those above a certain **threshold** should be taken into account
- Learn more about correlation at: <http://en.wikipedia.org/wiki/Correlation>

# Assignment

- Implement:
  1. Function to compute a set of good Harris corners including non maximum suppression:

```
[feature_pts] =  
detect_harris(I, sigma, threshold, width_of_non_max_sup_window)
```

    - output **feature\_pts** is a nx2 matrix containing the X/Y coordinates of the n computed feature points
  2. Compute correspondences from features by correlation:

```
[p1, p2] =  
match_by_correlation(feature_pts_1, feature_pts_2, I1, I2, patch_  
size)
```

    - Output **p1, p2** are two nx2 matrices containing in each row the coordinates of the corresponding features in images I1 and I2.
- Integrate the above implementations in the framework developed in Ass. 4 to obtain a simple fully automatic stitching
  - Experiment with the parameters to obtain good (and possibly fast results)