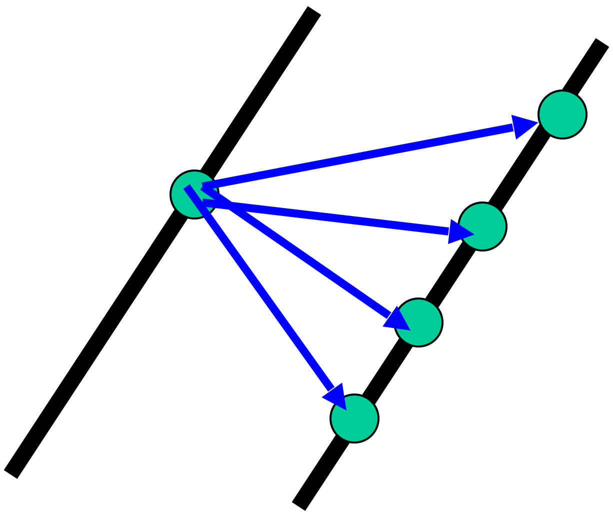
Harris Corner Detection

Mohammad Nayeem Teli

Corner detection

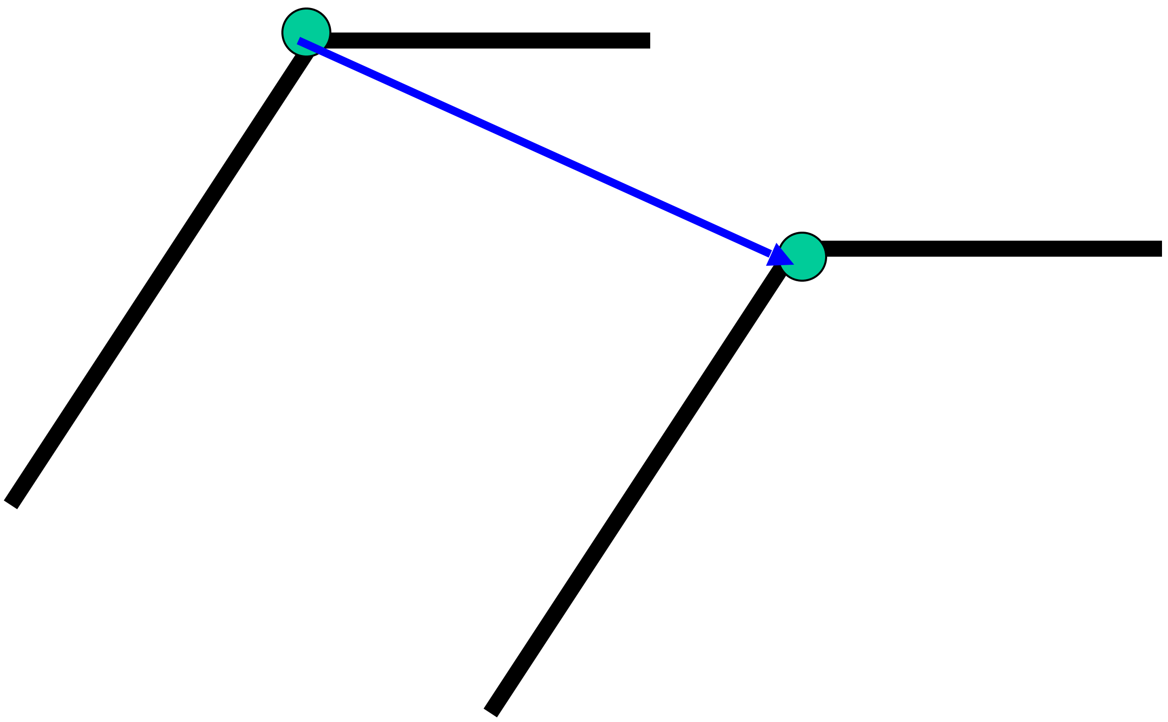
Corners contain more edges than lines.

A point on a line is hard to match.

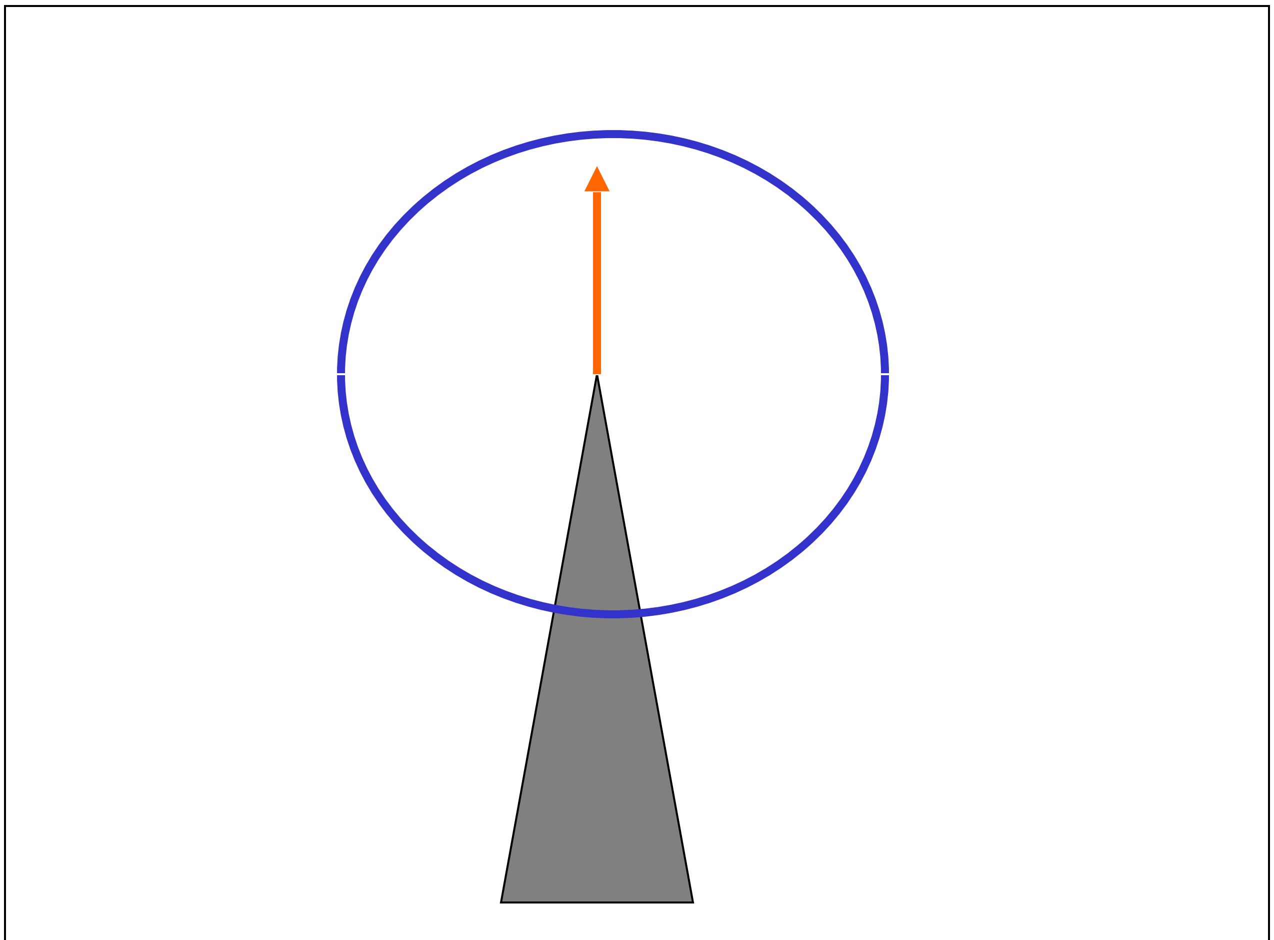


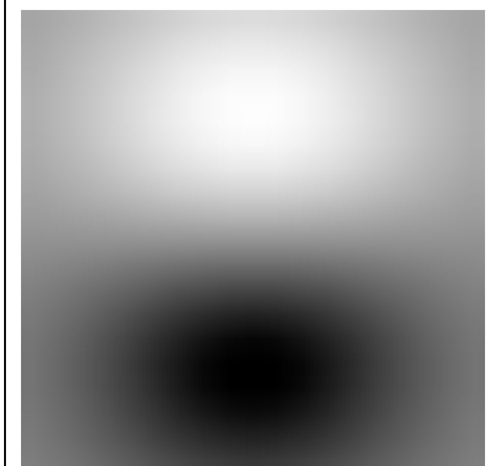
Corners contain more edges than lines.

A corner is easier



Edge Detectors Tend to Fail at Corners



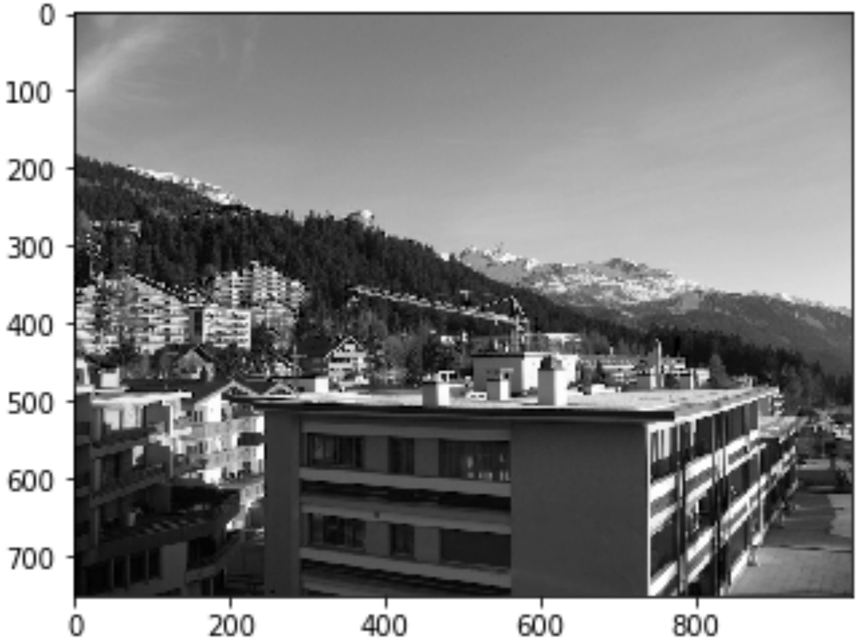


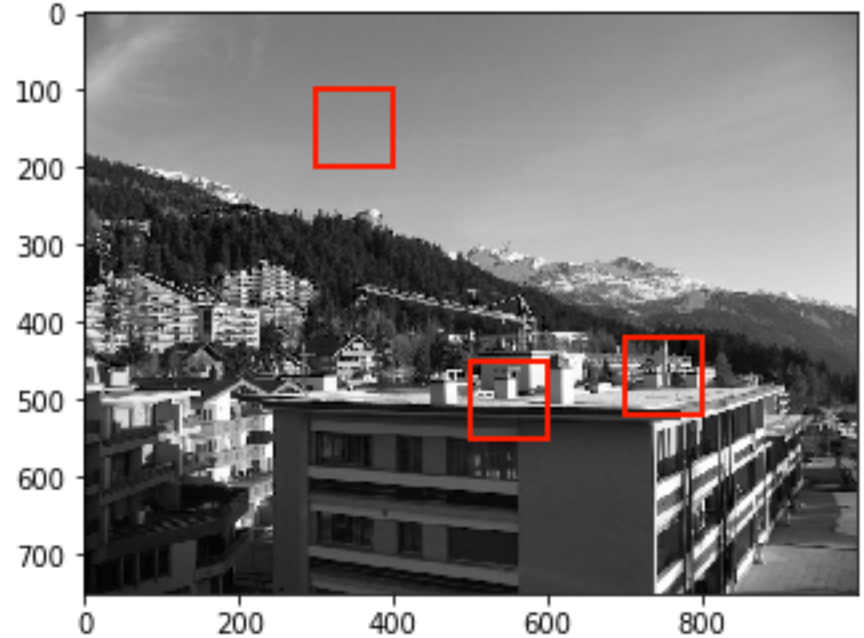
Finding Corners

Intuition:

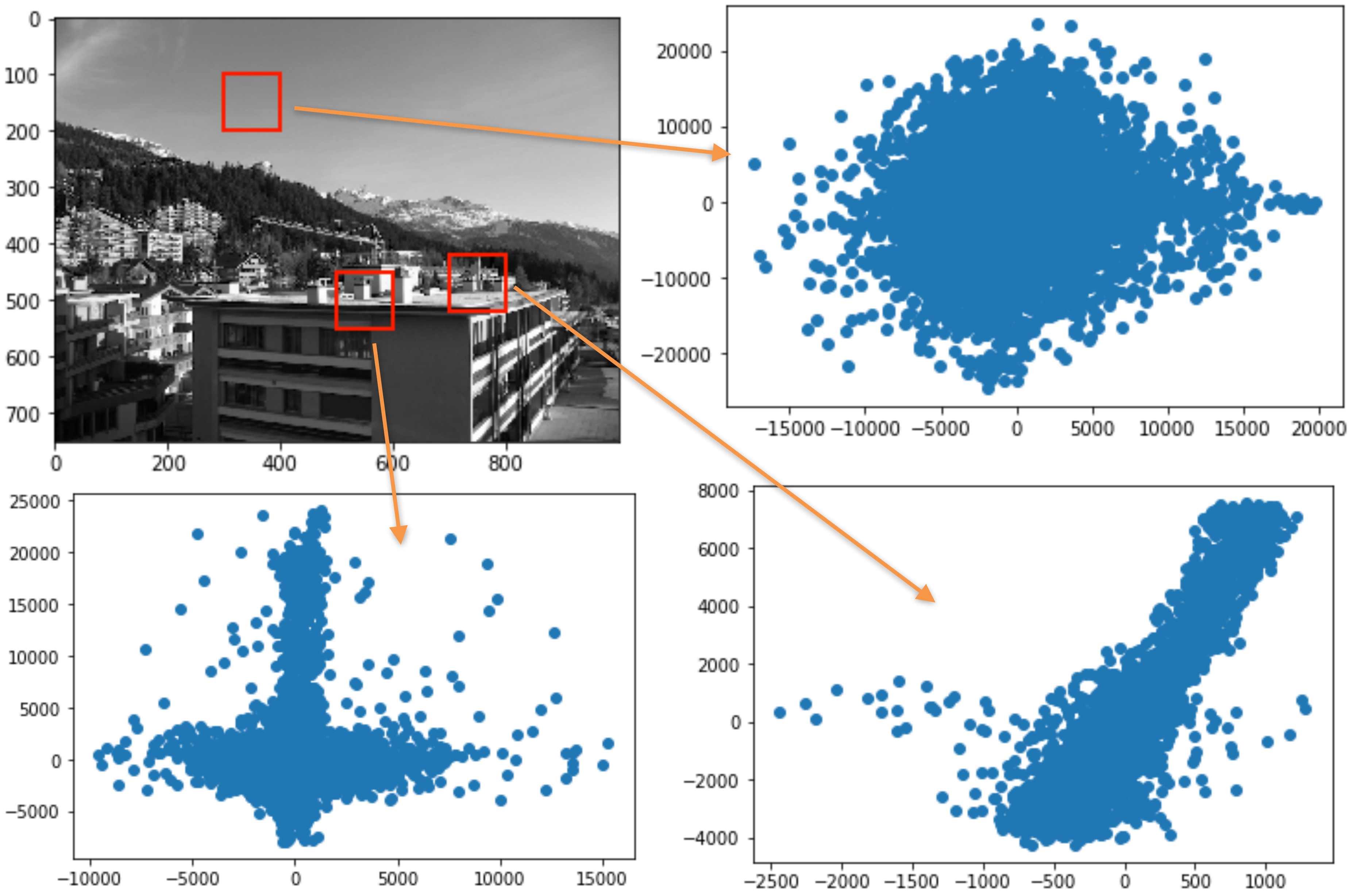
- Right at corner, gradient is ill defined.
- Near corner, gradient has two different values.

Background


0
100
200
300
400
500
600
700
0
200
400
600
800


0
100
200
300
400
500
600
700
0
200
400
600
800

Background



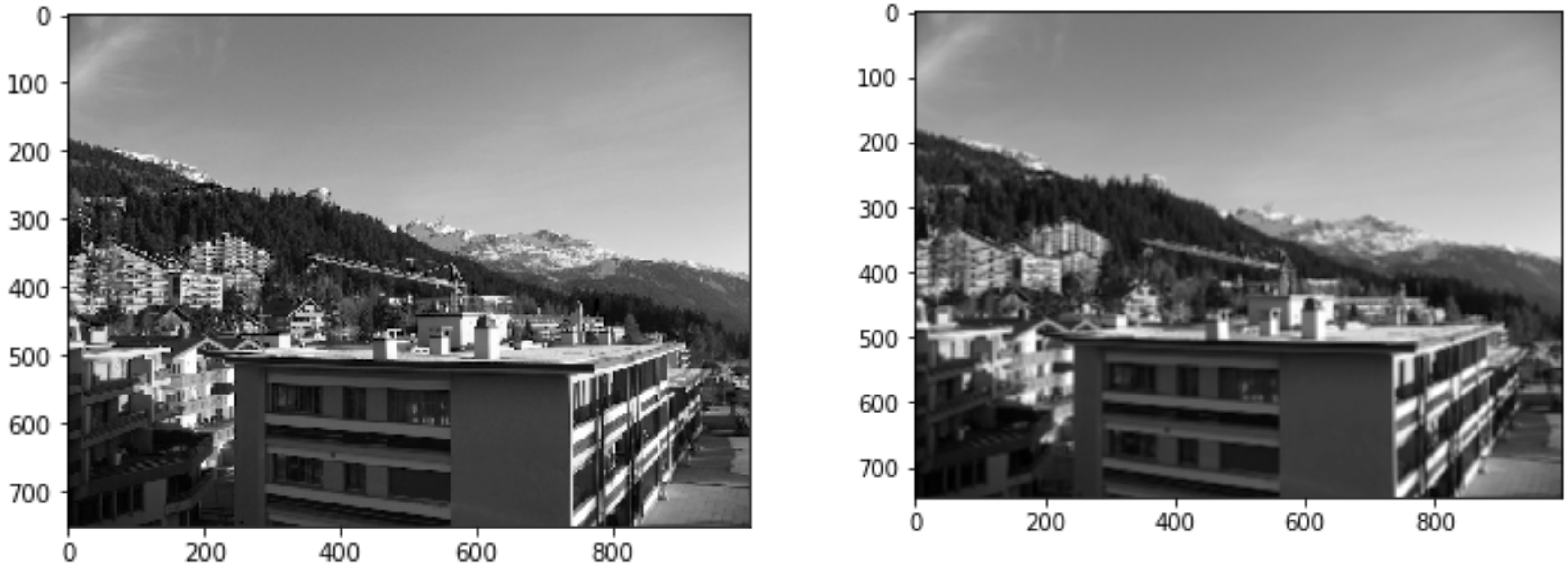
0
100
200
300
400
500
600
700
0
200
400
600
800
20000
10000
0
−10000
−20000
−15000 −10000 −5000 0 5000 10000 15000 20000
25000
20000
15000
10000
5000
0
−5000
−10000 −5000 0 5000 10000 15000
8000
6000
4000
2000
0
−2000
−4000
−2500 −2000 −1500 −1000 −500 0 500 1000

Sum of Square Differences (SSD)

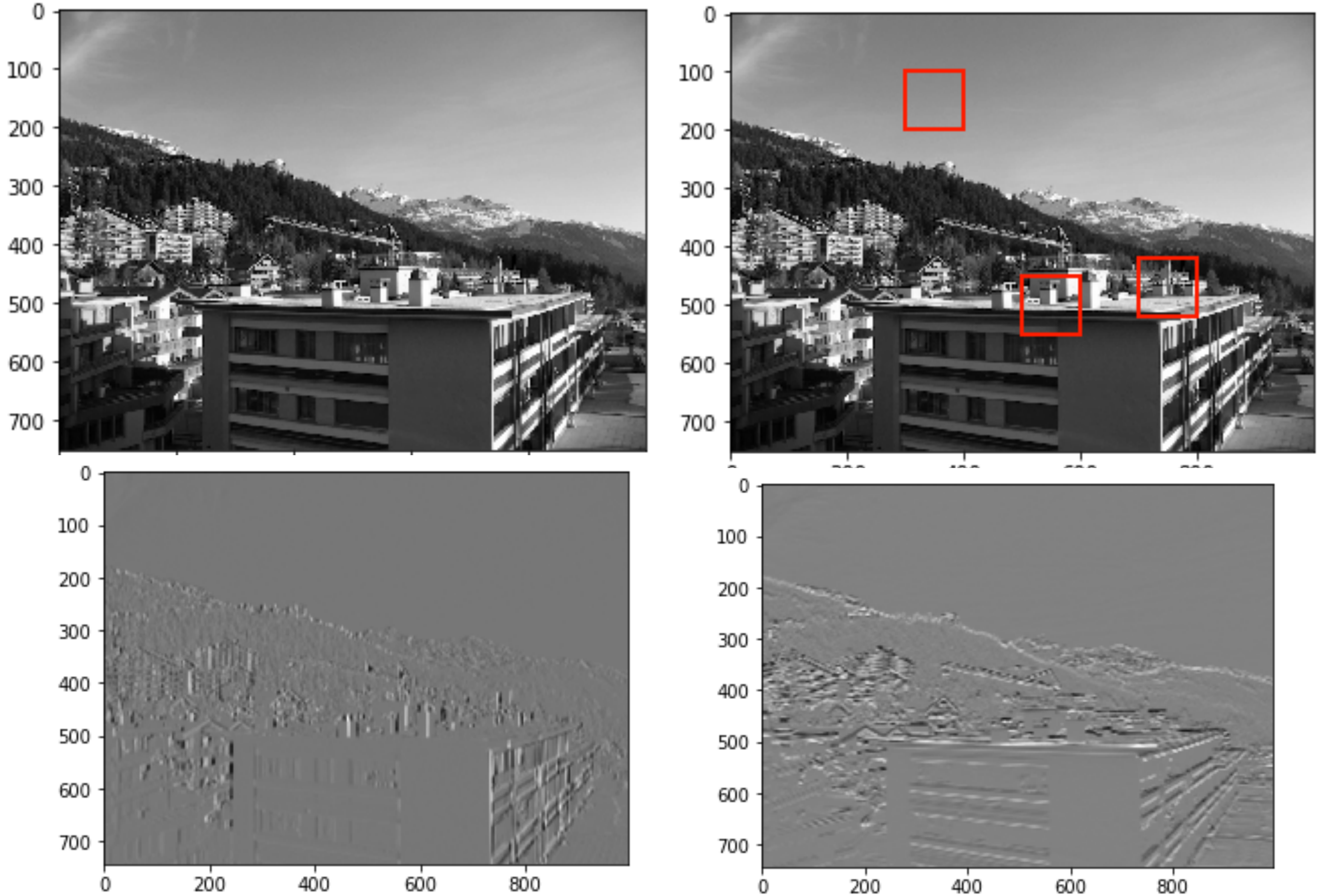
Intuition:

- Uses SSD to detect any fluctuation in the gradient of the image.
- Gradient should have significant change in two directions.

Smoothing


0
100
200
300
400
500
600
700
0
200
400
600
800
0
100
200
300
400
500
600
700
0
200
400
600
800

Gradient Images (I_x, I_y) - also subtract the mean



Finding Corners

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u,y+v) - I(x,y)]^2$$

E is the difference between the original and the moved window

u is the window's displacement in the x direction

v is the window's displacement in the y direction

$w(x,y)$ is the window at position (x,y). This acts like a mask.

I is the intensity of the image at a position (x,y)

$I(x+u,y+v)$ is the intensity of the moved window

Finding Corners

$$E(u, v) = \sum_{x,y} w(x, y)[I(x + u, y + v) - I(x, y)]^2$$

maximize E

$$\Longrightarrow \quad \text{maximize} \sum_{x,y} [I(x + u, y + v) - I(x, y)]^2$$

Taylor series expansion:

$$I(x + u, y + v) \approx I(x, y) + u\frac{\partial}{\partial x}I(x, y) + v\frac{\partial}{\partial y}I(x, y)$$

$$I(x + u, y + v) \approx I(x, y) + uI_x + vI_y$$

$$E(u, v) \approx \sum_{x,y} w(x, y)[I(x, y) + uI_x + vI_y - I(x, y)]^2$$

Finding Corners

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2$$

$$E(u,v) \approx \sum_{x,y} w(x,y)[I(x,y) + uI_x + vI_y - I(x,y)]^2$$

$$= \sum_{x,y} w(x,y)[uI_x + vI_y]^2$$

$$= \sum_{x,y} w(x,y)[u^2I_x^2 + 2uvI_xI_y + v^2I_y^2]$$

$$[u^2I_x^2 + 2uvI_xI_y + v^2I_y^2] = [u \quad v] \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

Finding Corners

$$[u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2] = [u \quad v] \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$E(u,v) \approx \sum_{x,y} w(x,y)[u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2]$$

$$E(u,v) \approx [u \quad v] \left(\sum w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

$$E(u,v) \approx [u \quad v]\, M \begin{bmatrix} u \\ v \end{bmatrix}$$

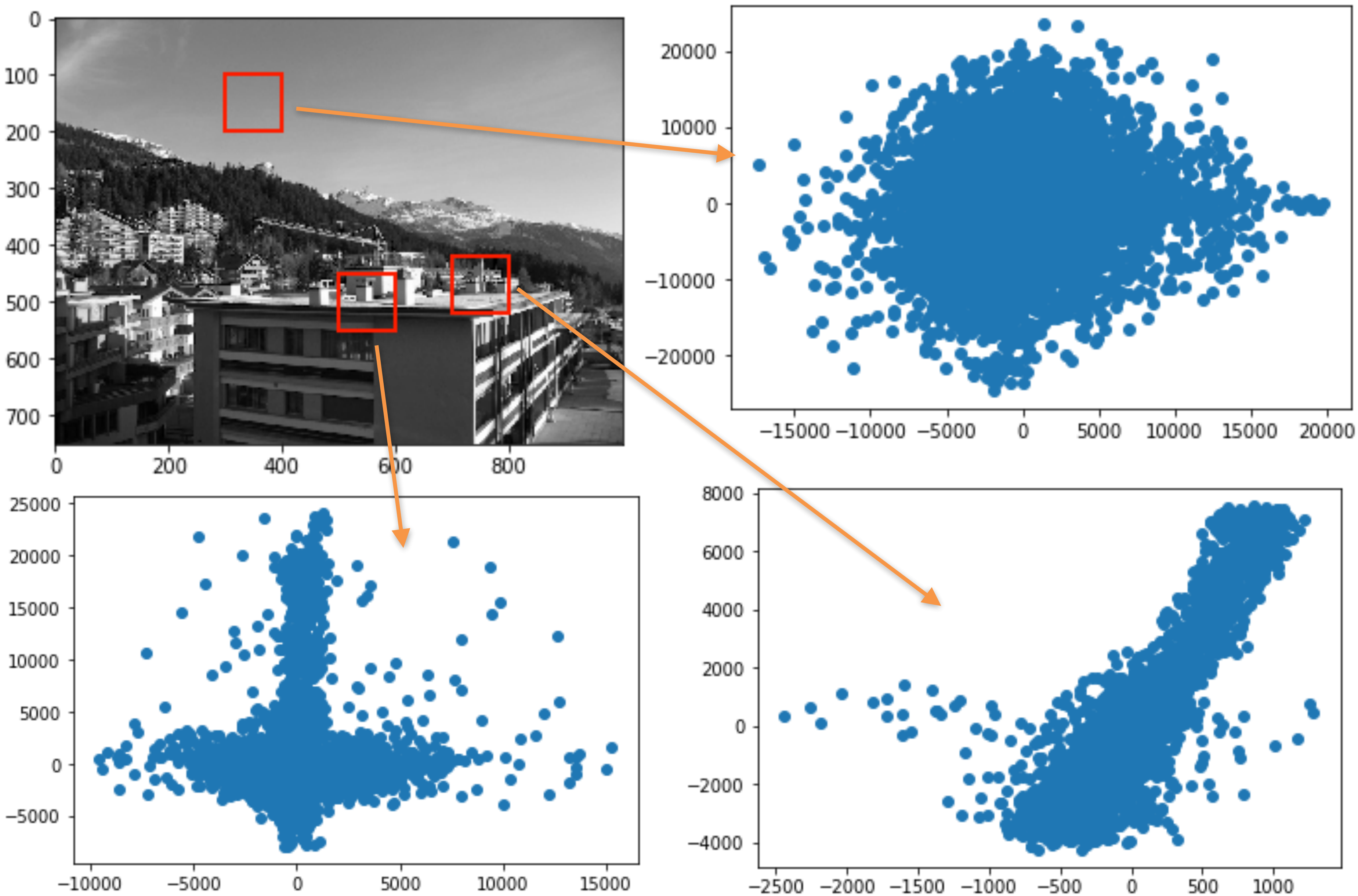
$$\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}$$

Gradient Covariance Matrix

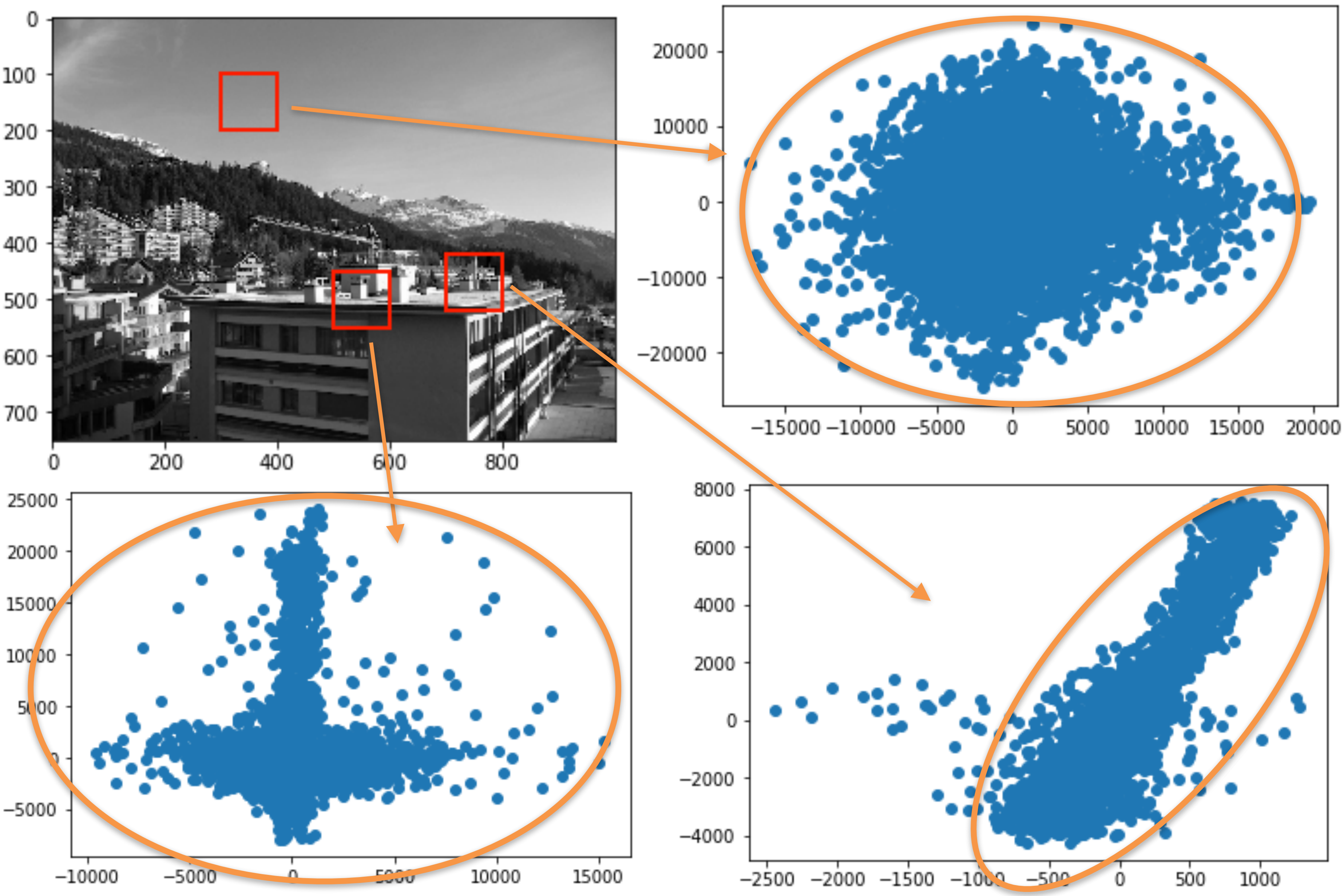
$$M = \sum w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}$$

windowing function - computing a weighted sum

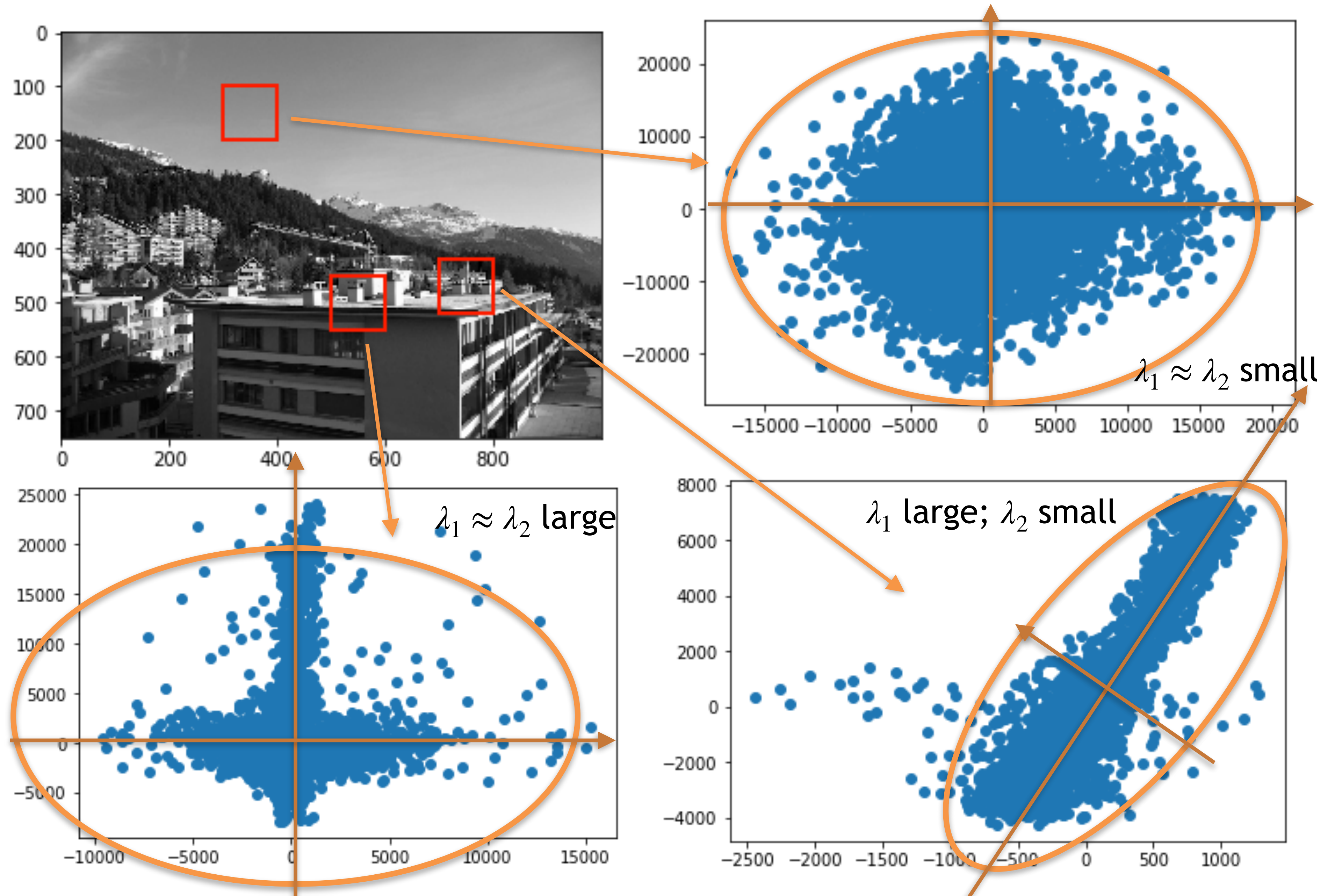
Gradient plots - I_x vs. I_y



Gradient plots - I_x vs. I_y



Gradient plots - I_x vs. I_y



Score for each window

$$E(u,v) \approx [u \quad v]\, M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \sum w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Eigen values of the matrix, M, can help determine the suitability of a window

Score, $R = det(M) - k(trace(M))^2$

$$det(M) = \lambda_1 \lambda_2$$

$$trace(M) = \lambda_1 + \lambda_2$$

k is an emppirically determined constant; $k = 0.04 - 0.06$

Corner detection

- If λ_1 and λ_2 are small, means we are in a flat region
- If $\lambda_1 >> \lambda_2$ significant change in one direction, it is an edge
- If $\lambda_1 \approx \lambda_2,$ and both are large, it is a corner

Score, $R = det(M) - k(trace(M))^2$

$$det(M) = \lambda_1 \lambda_2$$

$$trace(M) = \lambda_1 + \lambda_2$$

k is an emppirically determined constant; $k = 0.04 - 0.06$

Harris corner detector algorithm

-Compute magnitude of the gradient everywhere in x and y directions I_x, I_y

- Compute $I_x^2, I_y^2, I_x I_y$

- Convolve these three images with a Gaussian window, w. Find M for each pixel,

$$M = \sum w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

- Compute detector response, R at each pixel.

$$R = det(M) - k(trace(M))^2$$

- find local maxima above some threshold on R. Compute nonmax suppression.

Harris Corner Detector - Example