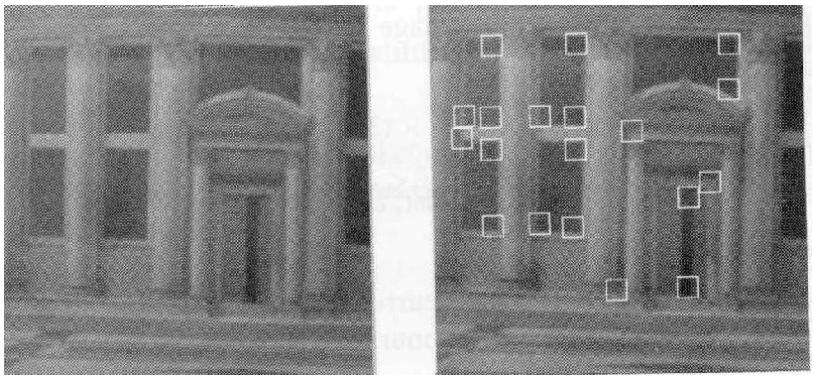
Point Detection

- We are often interested in detecting point features in an image.

- These features are usually defined as regions in the image where there is significant edge strength in two or more directions.



• A naive approach

- Apply a mask over the image.

- Apply thresholding:

if $|R| > T$, then possible discontinuity !

mask

-1	-1	-1
-1	8	-1
-1	-1	-1

*

original image

1	1	1	1	1
1	10	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

=

convolved image

-	-	-	-	-
-	72	-9	0	-
-	-9	-9	0	-
-	0	0	0	-
-	-	-	-	-

- Depending on the value of T we can get:

4 points ($0 < T \leq 9$)
1 point ($9 < T \leq 72$)
0 points ($T > 72$)

• **A better approach** (Trucco, pp. 82-85)

- Let us assume that f_x and f_y denote the partial derivatives of the image f.

- Let us assume a corner p and a neighborhood Q of p (e.g., 3x3, 5x5. etc.)

- The following matrix C is very important for deciding the presence of p (it characterizes the structure of the gray-levels):

$$C = \begin{bmatrix} \sum_Q f_x^2 & \sum_Q f_x f_y \\ \sum_Q f_x f_y & \sum_Q f_y^2 \end{bmatrix} = \sum_Q \begin{bmatrix} f_x \\ f_y \end{bmatrix} \begin{bmatrix} f_x & f_y \end{bmatrix}$$

- One way to examine the presence of a corner is to look the eigenvalues of C (they are both positive, why?):

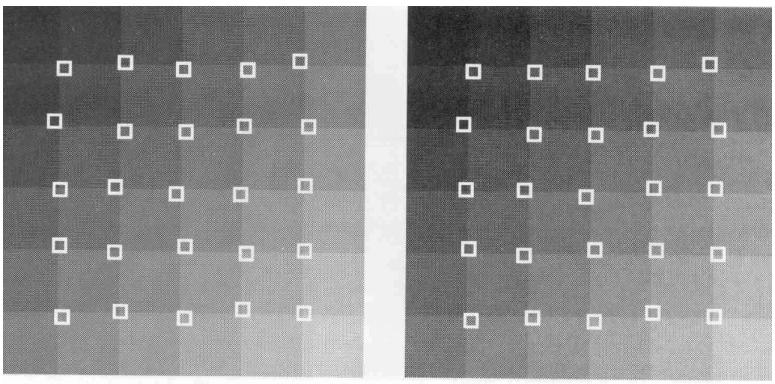
(1) *the eigenvectors encode edge directions*

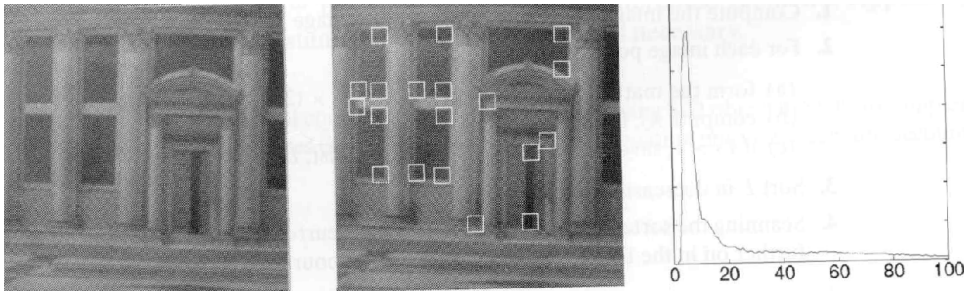
(2) *the eigenvalues encode edge strength*

(1) if the area is a region of constant intensity, both eigenvalues will be very small.

(2) if it contains an edge, there will be one large and one small eigenvalue (the eigenvector associated with the large eigenvalue will be parallel to the image gradient).

(3) if it contains edges at two or more orientations (i.e., a corner), there will be two large eigenvalues (the eigenvectors will be parallel to the image gradients).





- How should we compare the eigenvalues?

  * find locations where the smaller eigenvalue is greater than some threshold.

  * consider the ratio of the two eigenvalues.

Algorithm

Input: image f, threshold t for λ_2, size of Q

(1) Compute the gradient over the entire image f

(2) For each image point p:

  (2.1) form the matrix C over the neighborhood Q of p
  (2.2) compute λ_2, the smaller eigenvalue of C
  (2.3) if $\lambda_2 > t$, save the coordinates of p in a list L

(3) Sort the list in decreasing order of λ_2

(4) Scanning the sorted list top to bottom: delete all the points that appear in the list that are in the same neighborhood Q with p

• Problems

- How to choose t? (sometimes, the histogram of λ_2 values can help us pick a good value).

- How to choose the size of the neighborhood Q? (large neighborhoods do lead to good localization results while very small neighborhoods might not give good detection rates)