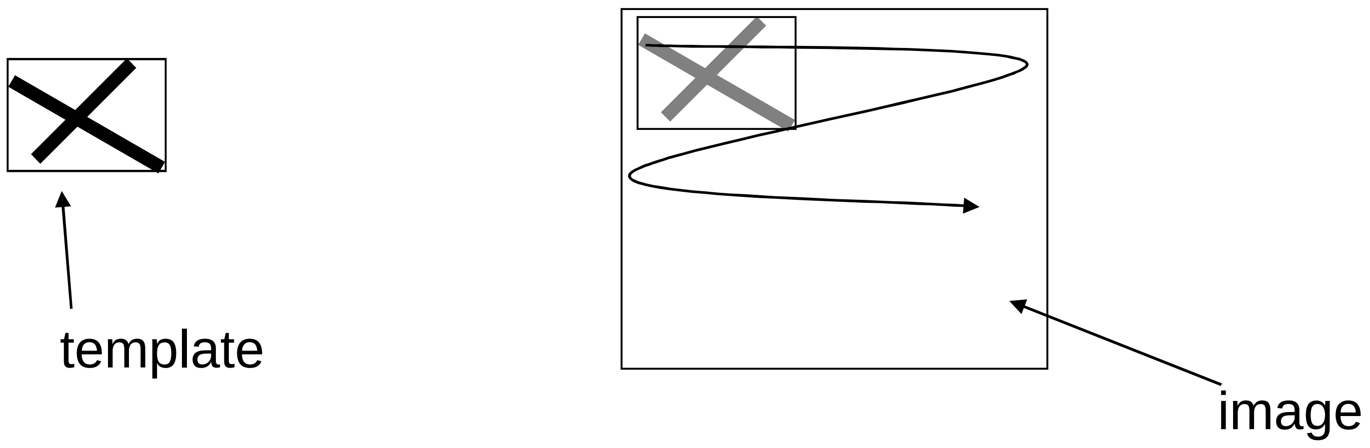
Interest Points

Introduction

- A way to detect some previously defined object within a given image is to perform *template matching*
- A *template* is a sub-image representing the “ideal” pattern that is sought in the image
- Involves the translation of the *template* to every possible position of the image, and the evaluation of a the level of *match* between the template and the image in that position
- *Global* versus *local* template



- For each position of the template in the image we evaluate a *similarity measure* between the template and the image
- Possible measures are *summation difference* or *cross-correlation*
- These measures are not only used for template matching but also to estimate the level of similarity between two signals

1. Euclidean distance

- Estimate the similarity between *g* and *t* in *m,n*:

$$E(m,n) = \sqrt{\sum_i \sum_j \left[g(i,j) - t(i-m, j-n) \right]^2}$$

The sum is computed for all points for which (i-*m,j-n)* is a valid coordinate of *t*, in other words for all points in the template image

- To find the *best match* we look for the smallest value in *E(m,n)*

- From an intuitive point of view we are computing the distance between two *HxW* dimensional points, where *H,W* is the size of the template
- Since we are not interested in the exact value of the distance (we search for the minimum in *E*) we can avoid computing the square root (Sum of Squared Differences):

$$SSD(m,n) = E^2(m,n) = \sum_i \sum_j \left[g(i,j) - t(i-m, j-n) \right]^2$$

- Or:

$$S(m,n) = \sum_i \sum_j \left| g(i,j) - t(i-m, j-n) \right|$$

$$E^2(m,n) = \sum_i \sum_j \left[g(i,j)^2 + t(i-m, j-n)^2 - 2g(i,j)t(i-m, j-n) \right]$$

assume this constant this is constant

2. Cross-correlation

$$R(m,n) = \sum_i \sum_j \left[g(i,j)t(i-m, j-n) \right]$$

t and *g* are similar when *R(i,j)* is **large**

Problems with cross-correlation:

-False matches if the image energy $\sum\sum g(i,j)^2$ varies with the position

-Values of R depends on the size of the template

-Not invariant to illumination change

Consider the correlation with a constant pattern, of gray value *v*

a	b	c
d	e	f
g	h	i

v	v	v
v	v	v
v	v	v

$R=v*(a+b+c+\ldots+i)$

Now consider the correlation with a constant image twice as bright

a	b	c
d	e	f
g	h	i

2v	2v	2v
2v	2v	2v
2v	2v	2v

R=2*v*(a+b+c+…+i)>v*(a+b+c+…i)

We get higher correlation, with the same template…

- Solution: normalize the intensity
  - subtract the mean of both signals
  - divide by std. deviation

$$\hat{g} = \frac{g - \overline{g}}{\sqrt{\sum\left(g - \overline{g}\right)^2}}, \qquad \hat{t} = \frac{t - \overline{t}}{\sqrt{\sum\left(t - \overline{t}\right)^2}}$$

The *normalized cross-correlation* is defined as:

$$NCC(m,n) = \frac{\sum_{i,j}\left[g(i,j) - \overline{g}_{m,n}\right]\left[t(i-m, j-n) - \overline{t}\right]}{\sqrt{\sum_{i,j}\left[g - \overline{g}_{m,n}\right]^2 \sum_{i,j}\left[t(i-m, j-n) - \overline{t}\right]^2}} \in [-1,1]$$

$\overline{t}$ mean value of t, $\overline{g}_{m,n}$ mean value of g in the region under t

Problems with template matching

- The template represents the object as we expect to find it in the image
- The object can indeed be scaled or rotated
- This technique requires a separate template for each scale and orientation
- Template matching become thus too expensive, especially for large templates
- Sensitive to:
  - noise
  - occlusions

Local template matching

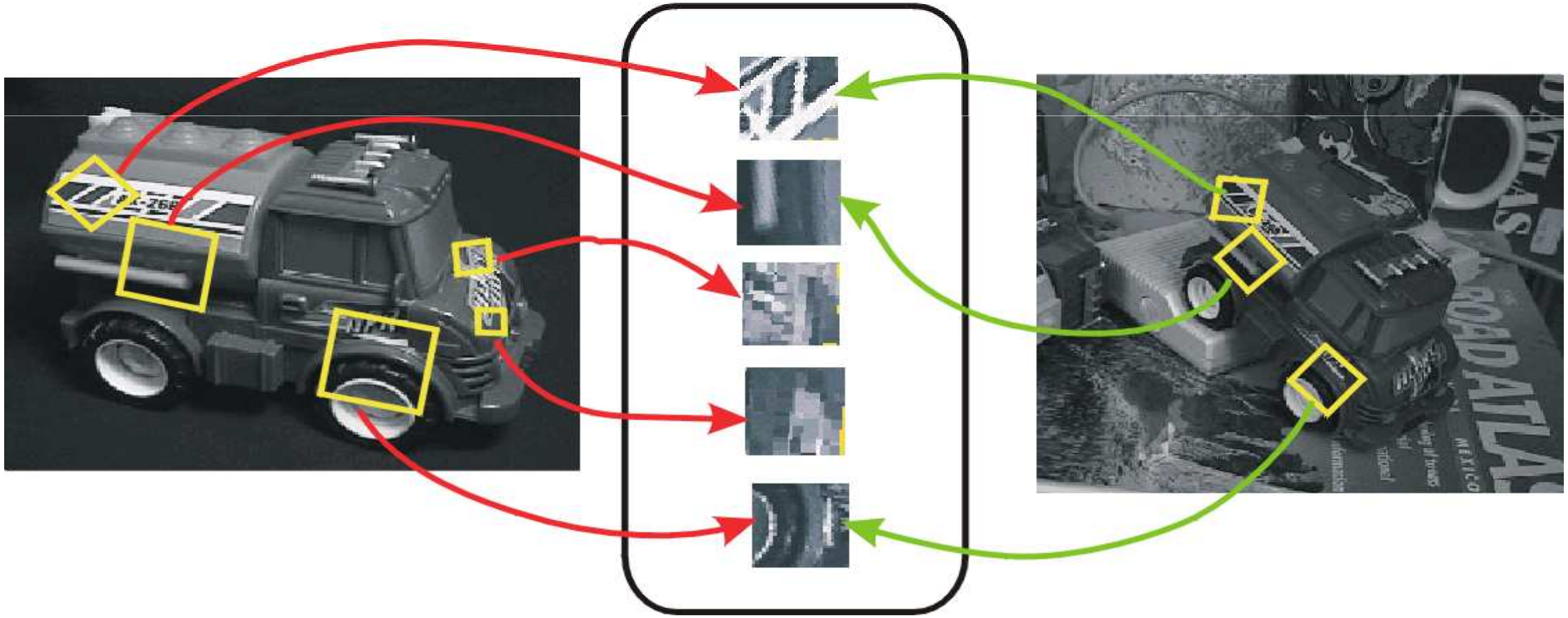
- A possible solution is to reduce the size of the templates, and detect *salient* features in the image that characterize the object we are interested in
- Extract a set of local features that are invariant to translation, rotation and scale…
- Perform matching only on these local features
- We then analyze the spatial relationships between those features

See for example:

Corner detector (Harris and Stephens,1988)

SIFT (Lowe, 1999)

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



Applications

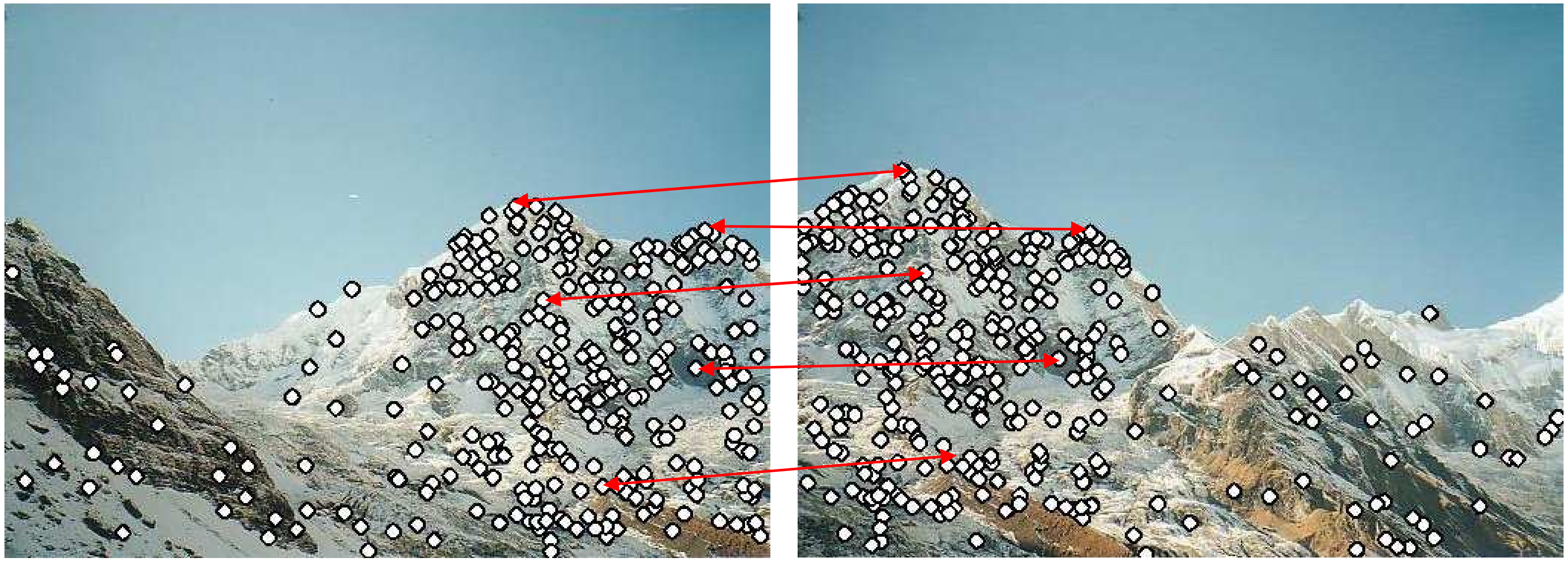
- Object recognition
- 3D reconstruction
- Motion detection
- Panorama reconstruction

How do we build a panorama?

- We need to match (align) images

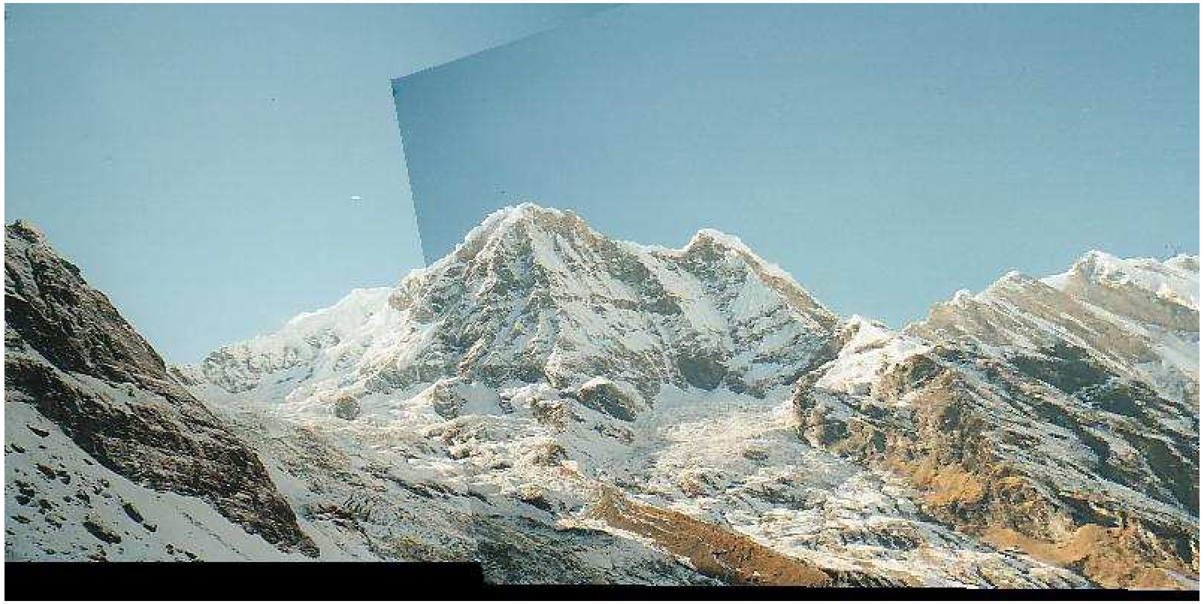
Matching with Features

- Detect feature points in both images
- Find corresponding pairs



Matching with Features

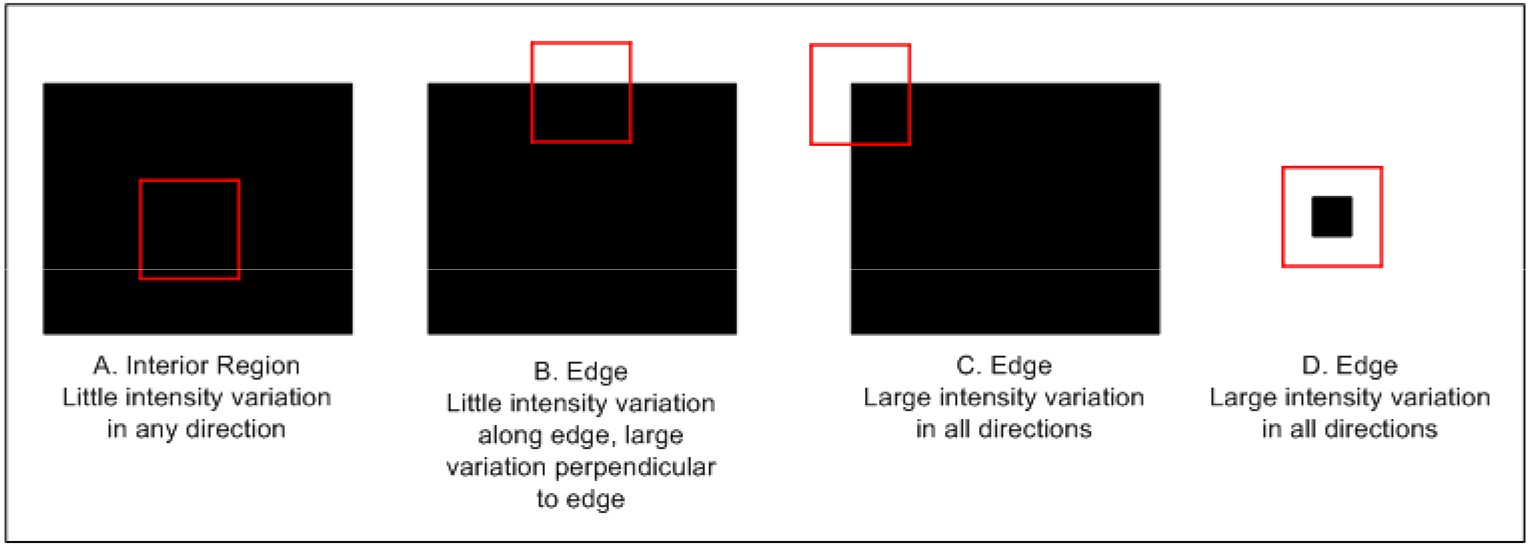
- Detect feature points in both images
- Find corresponding pairs
- Use these pairs to align images



Moravec interest operator

- This operator was developed back in 1977 for navigation
- Defines “points of interest” regions in the image that are good candidates for matching in consecutive image frames
- It is considered a “corner detection” since it defines interest points as points in which there is large intensity variation in every direction

- Place a small square window (3x3, 5x5, 7x7…) centered at a point P
- Shift this window by one pixel in each of eight principle directions (four diagonals, horizontal and vertical)
- Take the sum of squares of intensity differences of corresponding pixels in these two windows
- Define the intensity variation V as the *minimum* intensity variation over the eight principle directions


A. Interior Region
Little intensity variation
in any direction
B. Edge
Little intensity variation
along edge, large
variation perpendicular
to edge
C. Edge
Large intensity variation
in all directions
D. Edge
Large intensity variation
in all directions

Algorithm

for each x,y in the image

$$V_{u,v}(x,y) = \sum_{a,b} \left(I(x+u+a, y+v+b) - I(x+a, y+b) \right)^2$$

(u,v) are the considered shifts: $(1,0),(1,1),(0,1),(-1,1),(-1,0),(-1,-1),(0,-1),(1,-1)$

$$C(x,y) = \min(V_{u,v}(x,y))$$

threshold, set C(x,y) to zero of C(x,y)<T

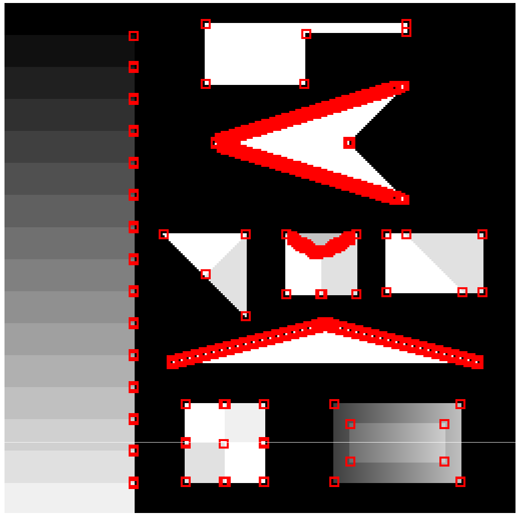
non-maximal suppression to find local maxima

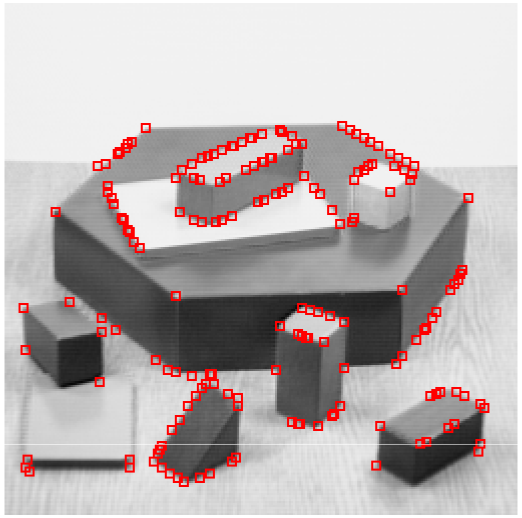
Non maximal suppression

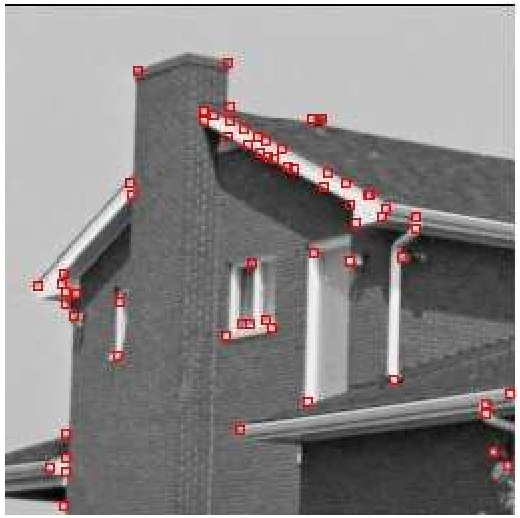
$$I(x,y)=\begin{cases}1 & if \quad V(x,y)\geq V(p,q), \forall (p,q)\in N(x,y)\\ 0 & otherwise\end{cases}$$

Source: http://www.cim.mcgill.ca/~dparks/CornerDetector/index.htm

Results







- Noise sensitivity
- Small imperfection in edges will be picked up as corners
- Edges not oriented along one of the height directions will also be candidate corners

Harris and Stephens

- Similarly to Moravec, consider now a continuous function:

$$E_h(\mathbf{x}) = \sum_{d \in W} \left(I(\mathbf{x}+\mathbf{d}) - I(\mathbf{x}+\mathbf{d}+\mathbf{h}) \right)^2$$

$$\mathbf{x} = (x, y) \in R^2$$

$$\mathbf{d}, \mathbf{h} \in R^2$$

- Perform local Taylor expansion:

$$I(\mathbf{x}+\mathbf{h}) \approx I(\mathbf{x}) + \nabla I(\mathbf{x})^T \mathbf{h} \Rightarrow I(\mathbf{x}+\mathbf{h}) - I(\mathbf{x}) = \nabla I(\mathbf{x})^T \mathbf{h}$$

$$\nabla I(\mathbf{x}) = \left[\frac{\partial I(\mathbf{x})}{\partial x}, \frac{\partial I(\mathbf{x})}{\partial y} \right]^T$$

$$E_h(\mathbf{x}) = \sum_{d \in W} \left(\nabla I(\mathbf{x}+\mathbf{d})^T \mathbf{h} \right)^2 = \sum_{d \in W} \mathbf{h}^T \nabla I(\mathbf{x}+\mathbf{d}) \nabla I(\mathbf{x}+\mathbf{d})^T \mathbf{h} =$$

$$= \sum_{d \in W} \mathbf{h}^T \cdot \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \cdot \mathbf{h} =$$

$$= \mathbf{h}^T \begin{bmatrix} \sum_W I_x^2 & \sum_W I_x I_y \\ \sum_W I_x I_y & \sum_W I_y^2 \end{bmatrix} \mathbf{h}$$

$$I_x = \frac{\partial I(\mathbf{x}+\mathbf{d})}{\partial x}$$

$$I_y = \frac{\partial I(\mathbf{x}+\mathbf{d})}{\partial y}$$

It is possible to introduce a weighting window w() for each point. For example a Gaussian:

$$w(\mathbf{d}) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-\|\mathbf{d}\|^2}{2\sigma^2}}$$

…and compute the filtered variation

$$\hat{E}_h(\mathbf{x}) = h^T \hat{C} h$$

$$\hat{C} = \begin{bmatrix} \sum I_x^2 w(\mathbf{d}) & \sum I_x I_y w(\mathbf{d}) \\ \ldots & \ldots \end{bmatrix}$$

the eigenvalues of C will be proportional to the principle curvatures of the image surface and form a rotationally invariant description of C

or, from linear algebra, if h is unit vector:

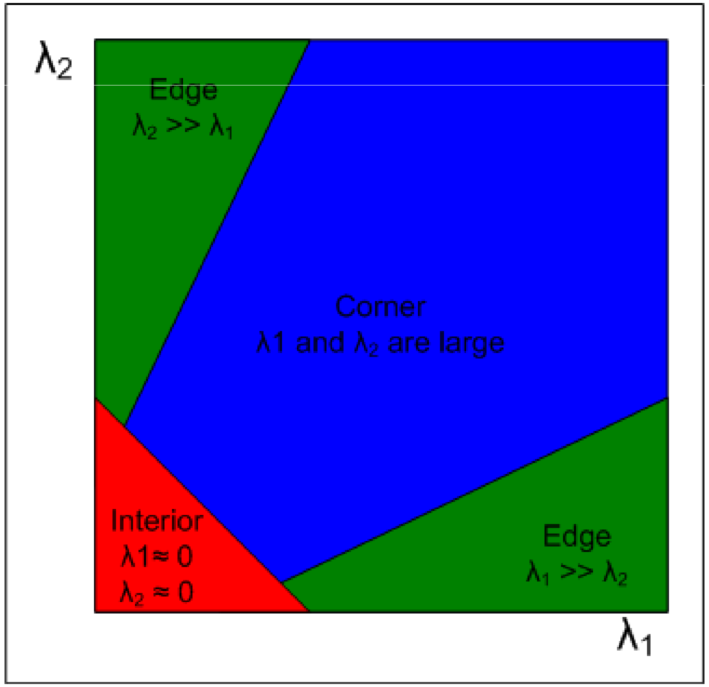
$$\lambda_1 < E_h\left(\mathbf{x}\right) < \lambda_2$$

Three cases

$$\lambda_1 \approx \lambda_2 \approx 0 \quad \Rightarrow \quad \text{no structure}$$

$$\lambda_1 \gg \lambda_2 \approx 0 \text{ or } \lambda_2 \gg \lambda_1 \approx 0 \quad \Rightarrow \text{edge}$$

$$\lambda_1 \gg 0 \text{ and } \lambda_2 \gg 0 \quad \Rightarrow \text{corner}$$



define the following index R

$\hat{C} = \hat{C}^t \Rightarrow$ trace and determinant are preserved

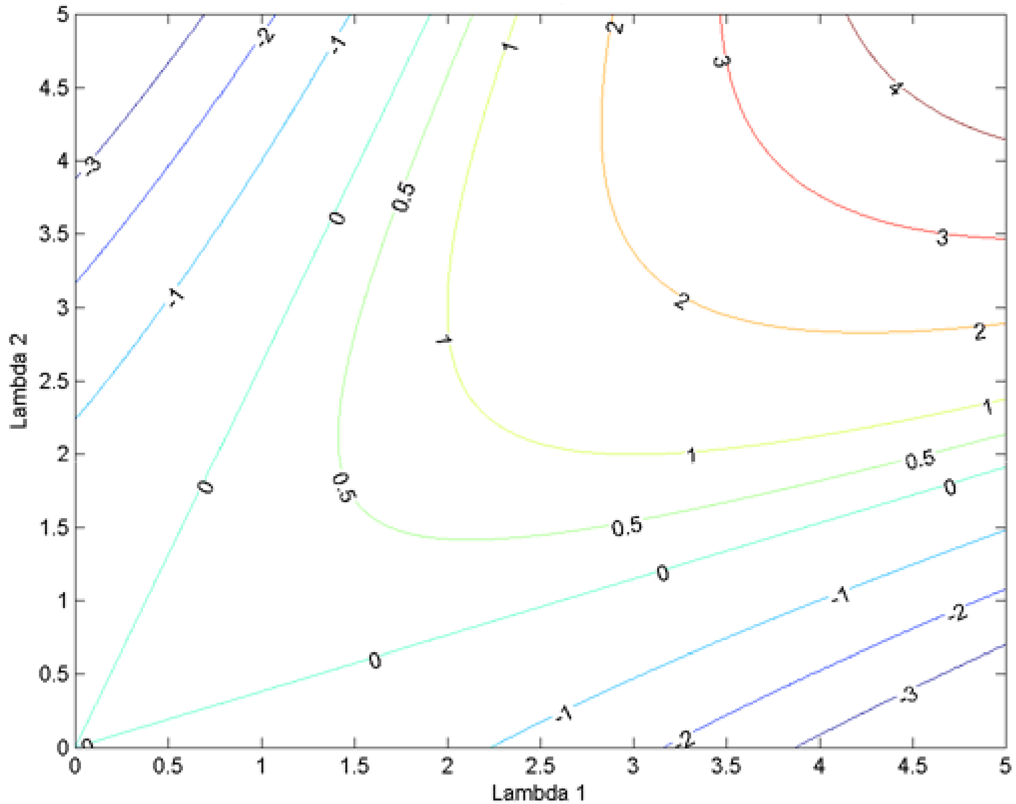
$$R = \det(\hat{C}) - k\,tr^2\left(\hat{C}\right)$$

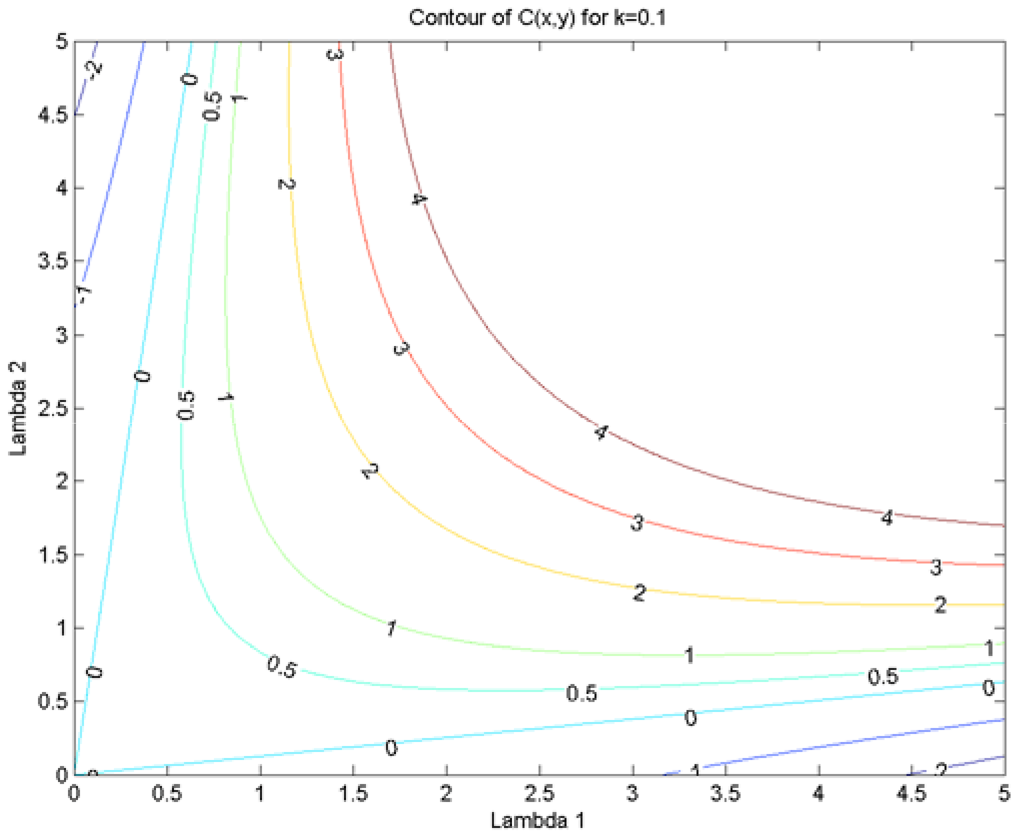
$$tr(\hat{C}) = \lambda_1 + \lambda_2 = \hat{I}_x^2 + \hat{I}_y^2$$

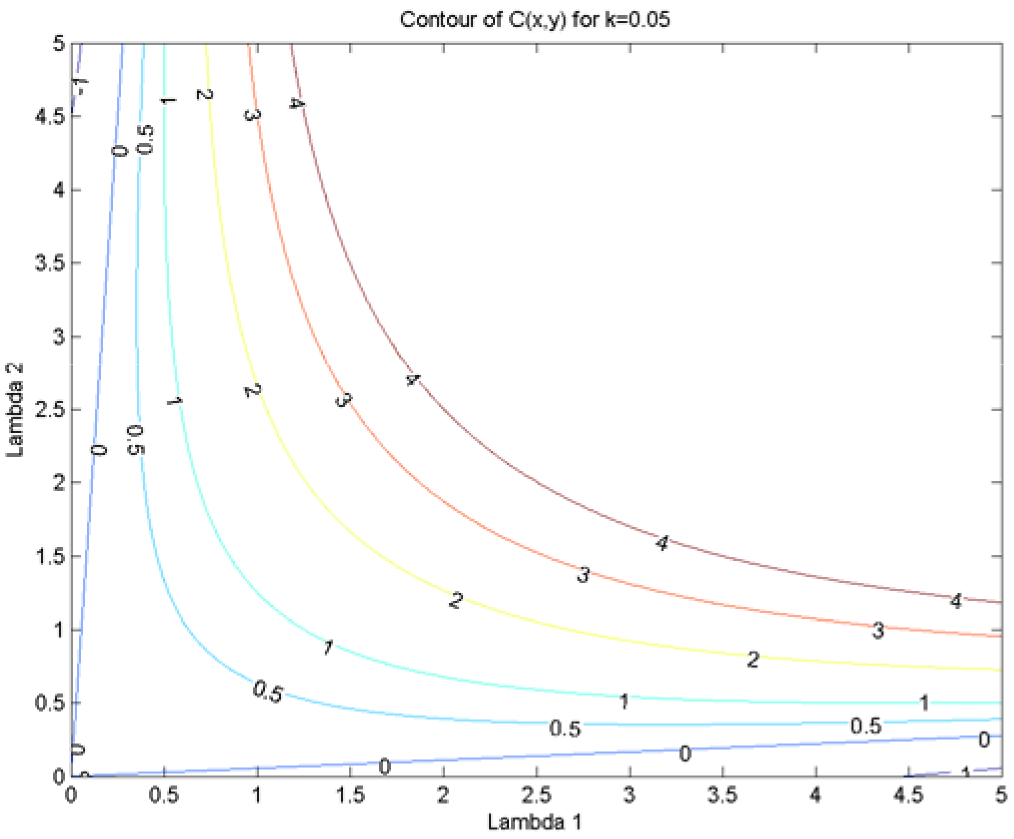
$$\det(\hat{C}) = \lambda_1 \lambda_2 = \hat{I}_x^2 \hat{I}_y^2 - \left(\widehat{I_x I_y}\right)^2$$

usually, k=0.04

R(x,y) for k=0.2

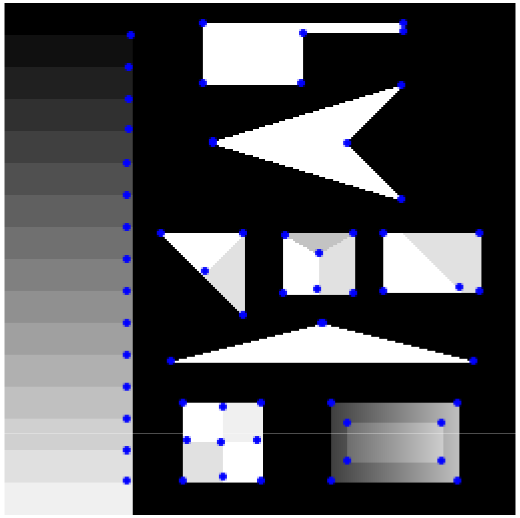


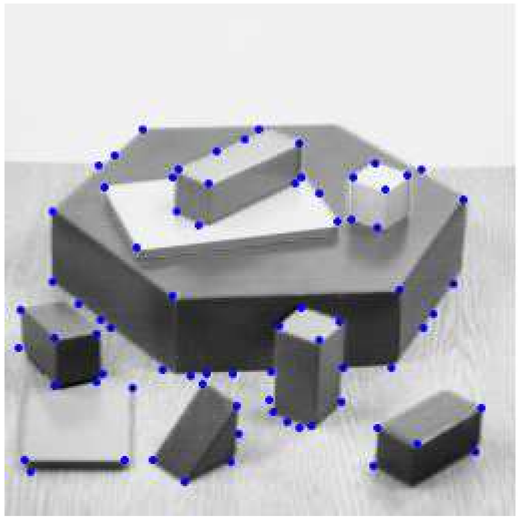

Contour of C(x,y) for k=0.1
Lambda 2
Lambda 1
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
-2
-1
0
0.5
1
2
3
4


Contour of C(x,y) for k=0.05
Lambda 1
Lambda 2
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
-1
0
0.5
1
2
3
4

Source: http://www.cim.mcgill.ca/~dparks/CornerDetector/index.htm

Results





Problems…

- Moravec and Harris have problems with rotations…