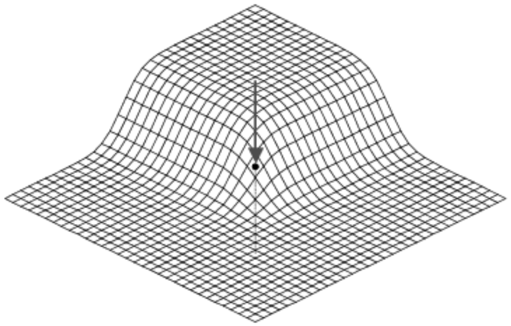
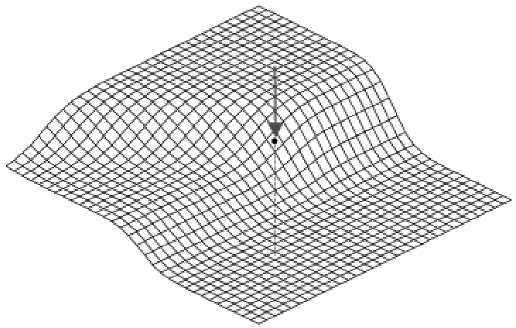
COMPSCI 773 S1C
Feature point extraction

– Harris (Plessey) detector
– Susan detector

What is a corner ?

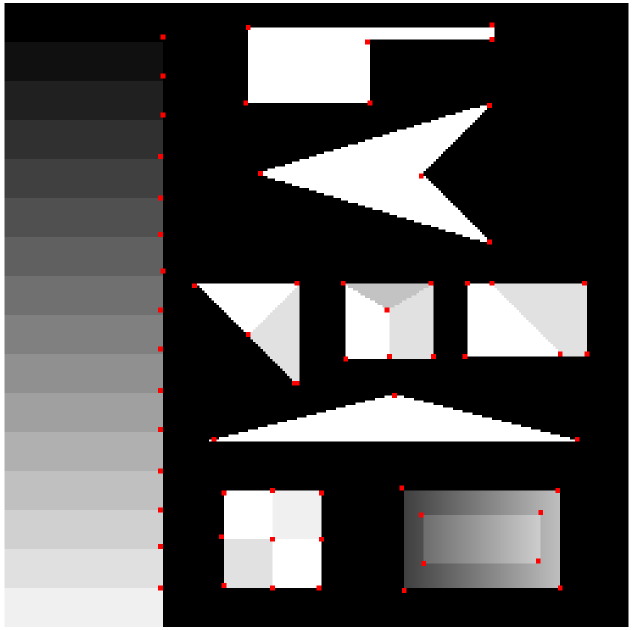
- The point at which the direction of the boundary of object changes abruptly
- Intersection point between two or more edge segments





In both images, the object is a continuous image area with a constant (or nearly constant) brightness or colour. The corner is at the intersection point between two (left image) or more edge segments (right image). Left image shows an example of a typical shape of the function of brightness in the neighbourhood of a corner. A more complicated brightness function is depicted in right image.

What is a corner ?



The two figures show an artificial and a real image, respectively, with the corners indicated in red.

Feature Point detection

The corner detectors should satisfy the following criteria:

- All (or most) the true feature points should be detected.
- No false feature points should be detected.
- Feature points should be well localized.
- Feature point detector should be robust with respect to noise.
- Feature point detector should be efficient.

2 families of corner detectors

1. Algorithms that work directly with the values of brightness of images (without segmenting the image in advance
   - Usually based on the study of derivatives (orientation, magnitude) of grey-level or color image
2. Algorithms that extract object boundaries first and analyze its shape afterwards
   - Boundaries often assumed to be extracted by edge-detectors
   - Usually based on the analysis of the curvature of boundaries

Group 2 seems to offer less reliability (use edge detectors for boundary extraction is not working well) and slower solutions.

The Harris-Plessey detector

Basic idea: Look at changes of Intensity in any direction on a small windows over any given point.

- Flat region: no change in all directions
- Edge: no change along the edge direction
- Corner: significant change in most directions

his translates into:

The following 2x2 symmetric matrix is considered at each image point (pixel) of the image.

$$M = \begin{bmatrix} \sum_w \left(\frac{\partial I}{\partial x}(x_k, y_k)\right)^2 & \left(\sum_w \frac{\partial I}{\partial x}(x_k, y_k)\right)\left(\sum_w \frac{\partial I}{\partial y}(x_k, y_k)\right) \\ \left(\sum_w \frac{\partial I}{\partial y}(x_k, y_k)\right)\left(\sum_w \frac{\partial I}{\partial x}(x_k, y_k)\right) & \sum_w \left(\frac{\partial I}{\partial y}(x_k, y_k)\right)^2 \end{bmatrix}$$

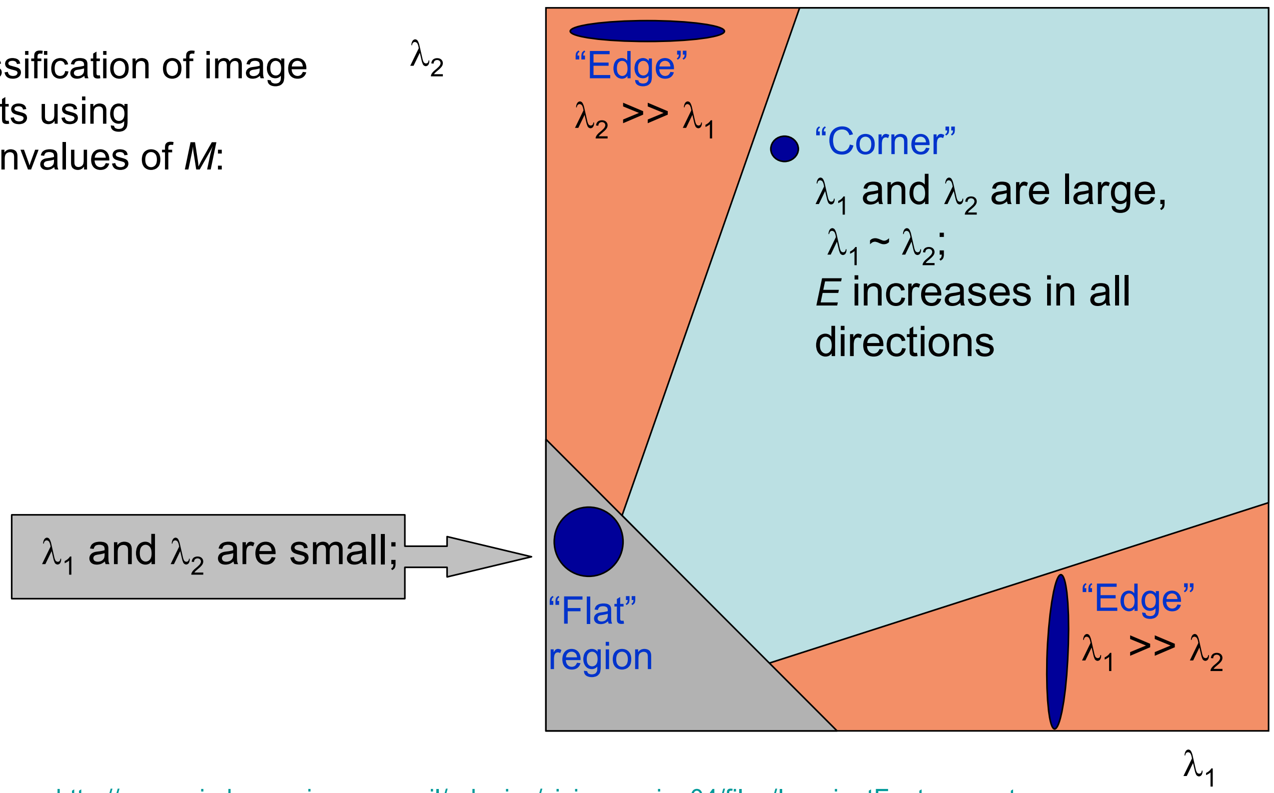
By evaluating the eigenvalues of the matrix M, we can detect the image feature by following rules:

1. If both eigenvalues are small, the intensity of the windowed image region is approximately constant (homogeneous region).
2. If one eigenvalues is high and the other is low, this indicates an edge.
3. If both eigenvalues are sufficiently large, the point is declared to be a corner.

→ A good (corner) point should have a *large intensity change* in *all directions*, i.e. *R* should be large positive

Harris Detector: Mathematics

Classification of image points using eigenvalues of *M*:



Modified from: http://www.wisdom.weizmann.ac.il/~deniss/vision_spring04/files/InvariantFeatures.ppt

The Harris-Plessey detector

1. In the Harris implementation, the corner is calculated as the ratio:

$$R_P = \frac{Trace(M)}{Det(M)}$$

- Thus, a point is marked as a corner if the value of Rp is less than the threshold and is the local minimum.
- Deemed unstable

The Harris-Plessey Detector: improvement

- Measure of corner response:

$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

$$\det M = \lambda_1 \lambda_2$$

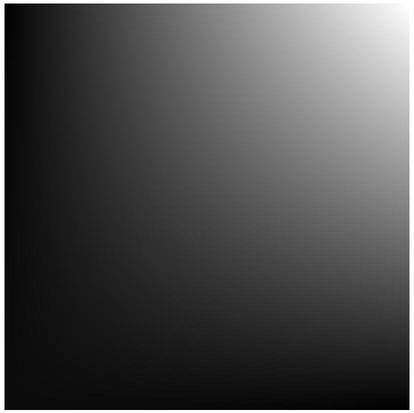
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

(k – empirical constant, k = 0.04-0.15)

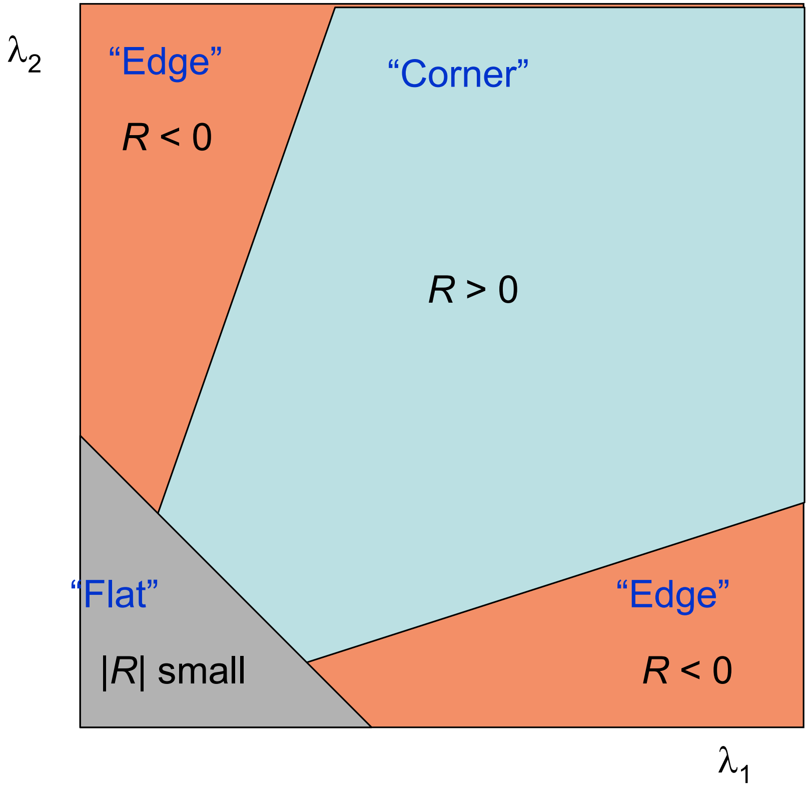
- The Algorithm:
  - Find points with large corner response function *R* (*R* > threshold)
  - Take the points of local maxima of *R*

Harris Detector: $R(\lambda_1,\lambda_2)$

- *R* depends only on eigenvalues of M
- *R* is large for a corner
- *R* is negative with large magnitude for an edge
- $|R|$ is small for a flat region



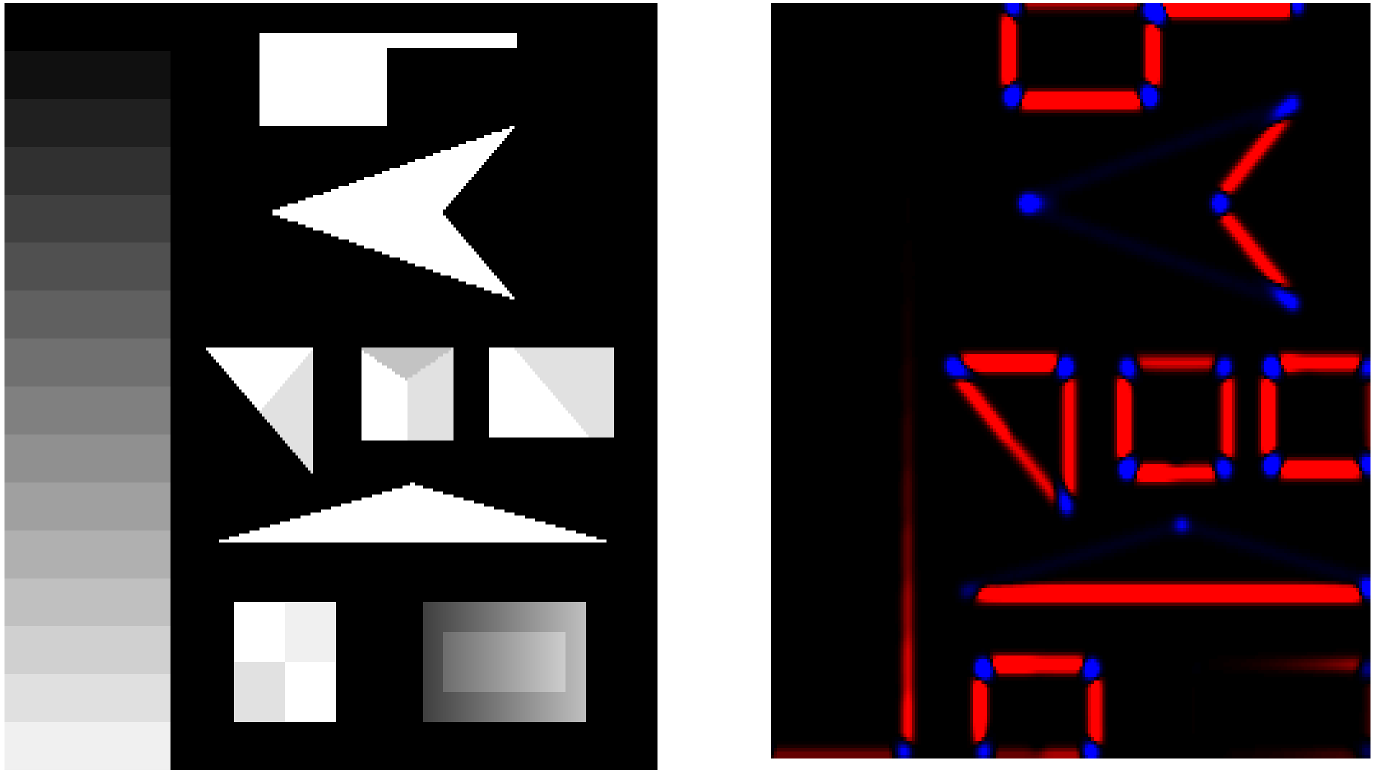
$$R=\lambda_1*\lambda_2-0.04(\lambda_1+\lambda_2)^2$$



Modified from: http://www.wisdom.weizmann.ac.il/~deniss/vision_spring04/files/InvariantFeatures.ppt

Example

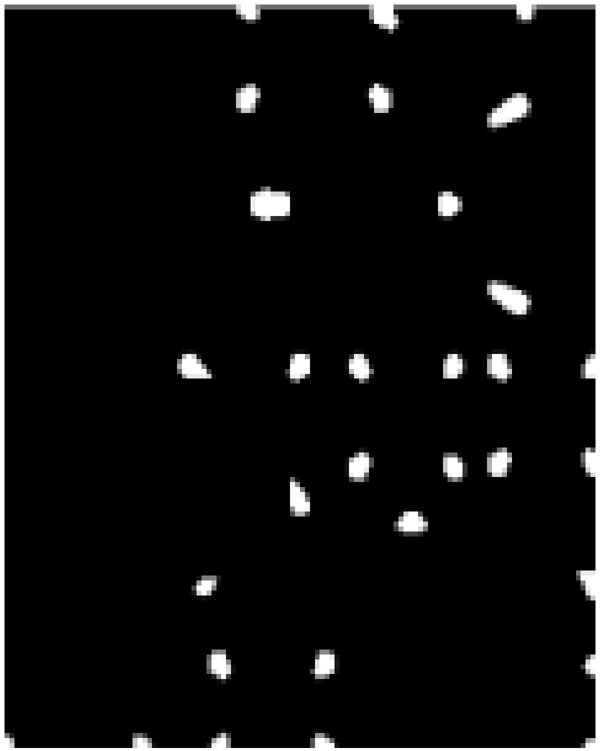
Harris Detector: Workflow



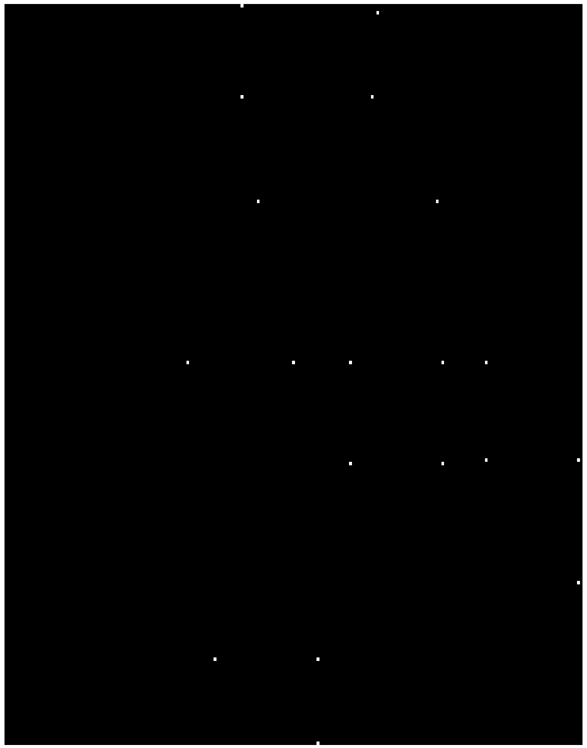
Coded R: *negative R in red; positive R in blue*

Harris Detector: Workflow

Find points with large corner response: R>threshold



Take only the points of local maxima of R



The Susan detector

The SUSAN detector differs from the others was published by Smith and Brady in 1997 [6] as it does not use any derivatives (no curvature or edge information requested).

The acronym SUSAN stands for Smallest "Univalue Segment Assimilating Nucleus".

1. The detector successively processes the points of the input image.
2. The point that is being examined is called nucleus.
3. The decision whether or not a point (nucleus) is a corner is based on examining a circular neighborhood centered around the nucleus.
4. The points from the neighborhood whose brightness is approximately the same as the brightness of the nucleus form the area referred to as USAN, standing for "Univalue Segment Assimilating Nucleus". In Figure 2, each mask from Figure 1 is depicted with its USAN shown in grey.

The shape of USAN conveys important information about the structure of the image in the region around the nucleus. For analysing the shape of USAN, its area and centroid are computed.

5. Compare the brightness difference between the nucleus and its neighbors (pixels within the same circular mask). To do so, for each pixel of the mask, estimate the difference:

where t is a brightness difference threshold; r_0 denote a nucleus and r a point in its neighborhood; c stands for the output of the comparison; and I(x) is the brightness of any pixel x.

$$c(r, r_0) = \begin{cases} 1 & \text{if } \left| I(r) - I(r_0) \right| \le t \\ 0 & \text{otherwise} \end{cases}$$

The Susan detector

6. The above comparison is done for each pixel r within the mask, and the USAN's area n (or count of pixels belonging to the USAN) can be defined by the equation:

$$n(r_0) = \sum_r c(r, r_0)$$

7. n is compared with a geometric threshold g, which is set to the half of the mask area (total pixels in the mask). For detecting the corners, the following function is introduced:

$$R(r_0) = \begin{cases} g - n(r_0) & if \quad n(r_0) < g \\ 0 & otherwise \end{cases}$$

8. The corners are detected at the points where the function $R(r_0)$ has its local maxima. This is a clear formulation of the SUSAN principle: the smallest USAN areas give the greatest values of $R(r_0)$ that are crucial for the detection.

The Susan detector (figures)

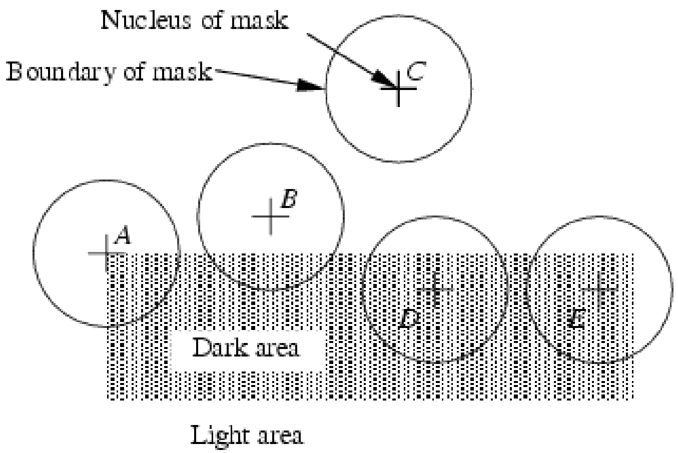


Figure 1: Five circular masks at different places on a simple image.

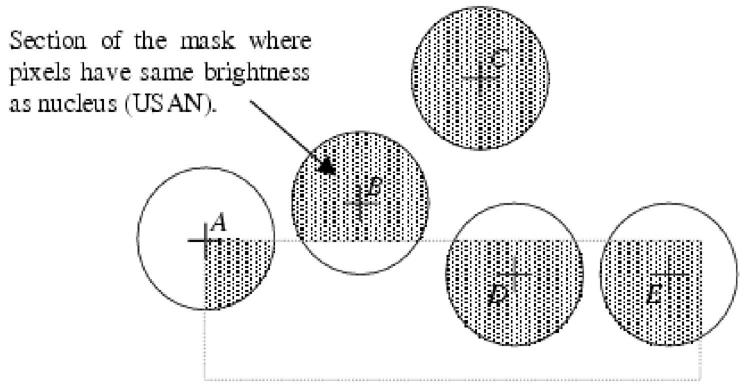
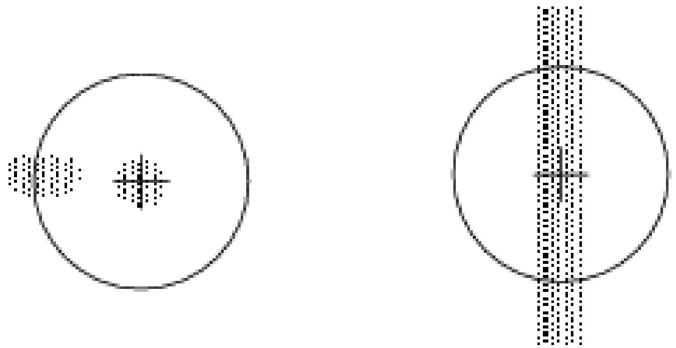


Figure 2: Five circular masks with similarity coloring; USANs are shown as the grey parts of the masks.

The Susan detector

Unfortunately, there exist some special cases in which the process described above fails. In the figure left below, The USAN is not continuous, but separated in two small regions. It’s obvious that the nucleus is not a corner, even though the function shows it is the local maxima. In another special case (Figure right below), the nucleus lies in a long thin area, which depicts the USAN is also very small. However, the value of R is high, which contradicts the fact that the point in question is not a corner.

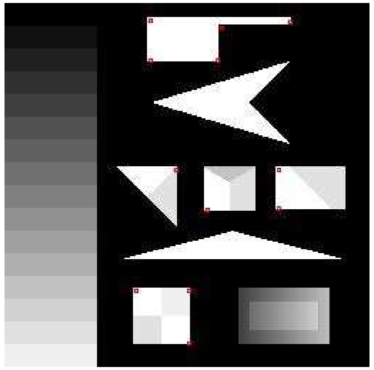


The Susan detector (improvement)

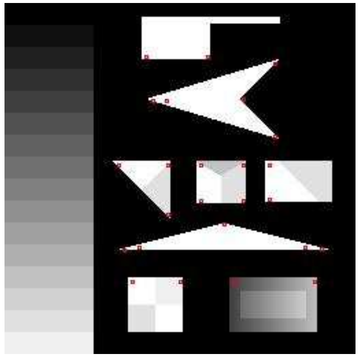
To overcome these problems, the authors propose to apply two rules:

1. Find the centre of gravity of the USAN and its distance from the nucleus. The point cannot become a corner if the distance is small.
2. The continuity of USAN is required. All the pixels within the mask, lying in the straight line pointing outwards from the nucleus in the direction of the centre of gravity must be part of the USAN for a corner to be detected.

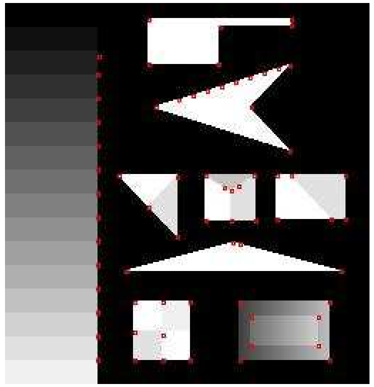
Results



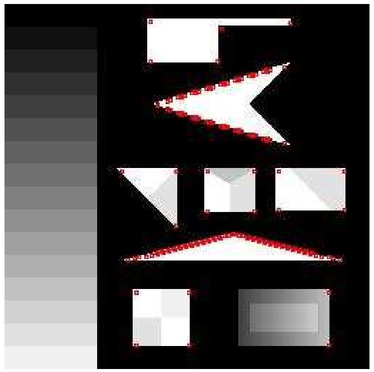
Harris, t=0.5



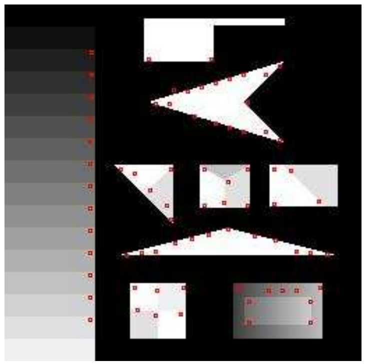
KLT, n=25
7*7 window



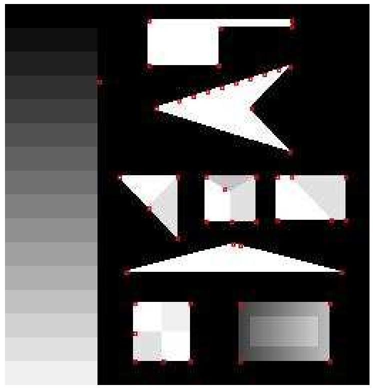
Susan, t=15



Harris, t=0.01



KLT, n=75
7*7 window



Susan, t=35

Susan Binary image example

1	1	1	1	0	0
1	1	1	0	0	0
1	1	0	0	0	0
1	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

A pixel is a local maxima if all pixels in its neighbourhood have smaller values than itself

R

	0	0	0	0	
	0	0	0	0	
	0	0	1	0	
	0	0	0	0	

n

	7	5	5	7	
	4	4	5	5	
	5	5	3	5	
	7	5	5	8	

Using a 3 by 3 mask as an approximation of a circular neighbourhood. This gives g=4 (or 4.5).
For binary images, t=0

Susan greyscale Image example

10	15	20	20	1	0
5	20	25	1	1	1
25	20	1	1	0	0
15	3	0	100	93	94
1	2	1	100	100	101
0	4	2	101	103	101

Using a 3 by 3 mask as an approximation of a circular neighbourhood. This gives g=4.
The choice of t is not fixed. Here the user decides. The use choses a t which best finds corners. Let's try t=5

A pixel is a local maxima if all pixels in its neighbourhood have smaller values than itself

R

	0	0	0	0	
	0	0	0	0	
	0	0	2	3	
	0	0	0	0	

All pixels with an intensity differing by less than 5 (t) are counted in the USAN

n

	5	5	5	7	
	4	4	5	5	
	5	5	2	1	
	7	5	4	6	

Susan Binary image example

1	1	1	1	0	0
1	1	1	0	0	0
1	1	0	0	0	0
1	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

A pixel is a local maxima if all pixels in its neighbourhood have smaller values than itself

R

	0	0	0	0	
	0	0	0	0	
	0	0	1	0	
	0	0	0	0	

n

	7	5	5	7	
	4	4	5	5	
	5	5	3	5	
	7	5	5	8	

Using a 3 by 3 mask as an approximation of a circular neighbourhood. This gives g=4 (or 4.5).
For binary images, t=0

Susan greyscale Image example

10	15	20	20	1	0
5	20	25	1	1	1
25	20	1	1	0	0
15	3	0	100	93	94
1	2	1	100	100	101
0	4	2	101	103	101

Using a 3 by 3 mask as an approximation of a circular neighbourhood. This gives g=4.
The choice of t is not fixed. Here the user decides. The use choses a t which best finds corners. Let's try t=5

A pixel is a local maxima if all pixels in its neighbourhood have smaller values than itself

R

	0	0	0	0	
	0	0	0	0	
	0	0	2	3	
	0	0	0	0	

All pixels with an intensity differing by less than 5 (t) are counted in the USAN

n

	5	5	5	7	
	4	4	5	5	
	5	5	2	1	
	7	5	4	6	

References

[1] C.G. Harris, *Determination of Ego-motion from Matched Points*, in 3rd Alvey Vision Conference, pages 189-192, 1987.

[2] Chris Harris and Mike Stephens, *A Combined Corner and Edge Detector,* Plessey Research Roke Manor, UK. 1988.

[3] J. Shi and C. Tomasi, *Good Features to Track*, IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600, 1994.

[4] B.D. Lucas and T. Kanade, *An Iterative Image Registration Technique with an Application to Stereo Vision*, International Joint Conference on Artificial Intelligence, pages 674-679, 1981.

[5] Stain Birchfield, *KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker*, and website: http://www.ces.clemson.edu/~stb/klt/.

[6] S.M. Smith and J.M. Brady, *SUSAN - A New Approach to Low Level Image Processing*, International Journal of Computer Vision, 23(1): 44-78, May 1997.

[7] F. Mokhtarian and R. Suomela, *Robust Image Corner Detection through Curvature Scale Space*, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 20, 1998.

[8] G. Scott and H. Longuet-Higgins, *An Algorithm for associating the features of two patterns*, in Proc. Royal Society London, v.B244, pages 21-26, 1991.

[9] E. Sojka, Reconstructing Three-dimensional Objects from the Images Produced by a Moving Camera, 21st International Workshop Advanced Simulation of System ASIS, Krnov, Czech Republic, 65-72, 1999.

[10] S. Pollard, J. Mayhew and J. Frisby, *PMF: A Stereo Correspondence Algorithm Using a Disparity Gradient Limit*, Perception, (14), 1985.

[11] S. Ullman, *the Interpretation of Visual Motion*, MIT Press, Cambridge MA, 1979.

[12] M. Pilu, Uncalibrated *Stereo Correspondence by Singular Value Decomposition*, HP laboratories Bristol, HPL-97-96, 1997.