A Corner Orientation Detector

F. Chabat*, G.Z. Yang, P. Burger, D.M. Hansell
Department of Imaging. Royal Brompton Hospital. London SW3 6NP
Department of Computing. Imperial College. London SW7 2AZ

Abstract

This paper introduces a corner detector that also estimates the corner's orientation. It can deal with junctions as well as corners. Corner points are detected as intensity patterns that are anisotropic along several directions. Pixels belonging to the corner's arms are then analysed, and a histogram search provides a measure of their dominant orientations. Based on a single derivative scheme proposed by Yang et al [3], the approach is proved to be insensitive to noise. Applications to synthetic and real-life images are shown.

Introduction

Corners and junctions are essential 2D features of an image. The certain identification of these points provides important information in numerous Computer Vision applications such as stereo-vision, motion detection and scene analysis. They indicate the presence and position of objects, narrowing down the search problem and making high-level interpretation of images easier. To this end, it is also important to evaluate the orientation of the arms, *i.e.*, the edges which intersect as corners or junctions.

Most corner detectors are based on second order derivative schemes. Kitchen and Rosenfeld [6]compute the rate of change of gradient direction along an edge, multiplied by the gradient magnitude, as an estimation of the strength of a corner by using the following expression:

$$c(\mathbf{x}) = \frac{I_{xx}I_y^2 - 2I_{xy}I_xI_y + I_{yy}I_x^2}{I_x^2 + I_y^2} \tag{1}$$

where $\mathbf{x}$ = (x, y), I($\mathbf{x}$) is the original image, $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, $I_{xx} = \frac{\partial^2 I}{\partial x^2}$, $I_{xy} = \frac{\partial^2 I}{\partial x \partial y}$ and $I_{yy} = \frac{\partial^2 I}{\partial y^2}$. Dreschler and Nagel [5], Zuniga and Haralick [7] adopted nearly equivalent approaches, all of them being sensitive to noise, because of the fact that derivatives amplify noise. It has been suggested that median corner detectors - that rely on the difference between an image and its median-filtered version - give better results than that of the Kitchen-Rosenfeld technique [1]. Other methods for corner detection include template matching (but a large

*Corresponding author. Email: fc3@doc.ic.ac.uk

set of templates is often necessary, making the search practically infeasible), generalised Hough transform (adapted to curved or chipped corners but with sensitivity and accuracy problems requiring a fine tuning) [2] and dissimilarity corner detectors (a fast and robust method that does not require image derivation, and therefore cannot provide edge information) [4].

Traditional corner detection does not tackle the problem of corner orientation. A common solution is to find the mean gradient orientation over a small neighbourhood of each estimated corner. This provides only the averaged orientation of the arms of the corner, with an accuracy less than 20 degrees [1]. These techniques are not well adapted to handling junctions.

In this paper we introduce a corner detector based on the analysis of local anisotropism [3] and identify corners as points of strong gradient intensity without a single dominant orientation. We then calculate the orientation of the arms of the corner or junction by measuring the likelihood of surrounding pixels being part of the corner structure.

Measure of Anisotropism

For the purpose of defining an adaptative filter, Yang et al [3] proposed a technique for defining and computing a measure of anisotropism at each point within an image. In that paper, the issue of corner identification has been addressed. Here we show how it can be extended to corner orientation detection.

For a strongly orientated intensity pattern along one direction, the power spectrum clusters along a line through the origin in the Fourier domain. By determining this line, and how closely it approximates the Fourier transform of the image, the orientation θ and the strength g of the anisotropism of the pattern can be derived. It has been demonstrated that the computation of g and θ does not require the actual computation of the Fourier transform. Indeed, the analytical expressions obtained are:

$$\theta(\mathbf{x}) = \frac{1}{2} tan^{-1} \frac{\iint_\Omega 2 I_x I_y dx dy}{\iint_\Omega \left(I_x^2 - I_y^2\right) dx dy} + \frac{\pi}{2} \tag{2}$$

$$g\left(\mathbf{x}\right) = \frac{\left(\iint_\Omega \left(I_x^2 - I_y^2\right) dx dy\right)^2 + \left(\iint_\Omega 2 I_x I_y dx dy\right)^2}{\left(\iint_\Omega \left(I_x^2 + I_y^2\right) dx dy\right)^2} \tag{3}$$

with the same notations as those used in Equation (1). In Equations (2) and (3) Ω is a small neighbourhood of $\mathbf{x}$. This scheme uses single derivatives only, which are integrated, and thus it reduces the effect of noise.

The value of $g(\mathbf{x})$ is 1 for a strongly orientated pattern along one direction, and is 0 for isotropic regions. For real images with a low signal-to-noise ratio, this method proves to be more robust than the mere estimation of the gradient direction.

In order to apply the above technique to corner recognition, we introduce two assumptions about the characteristic features of corners:

- like all edge points, corners and junctions have a strong intensity gradient
- unlike straight edges they do not have a single dominant orientation

These properties lead to an analytical expression of *cornerness*:

$$c(\mathbf{x}) = \psi\left(g(\mathbf{x})\right) \|\nabla I(\mathbf{x})\| \tag{4}$$

with $\psi(t)$ being a monotonic decreasing function from 1 to 0 within [0:1]. In our implementation, we chose:

$$\psi(t) = (1-t)^m \tag{5}$$

with $m = 2$ typically.

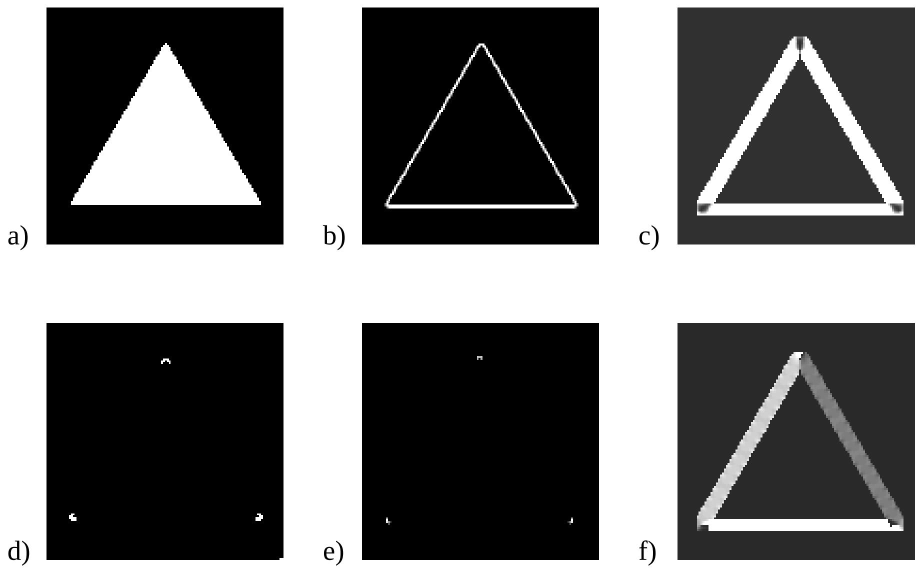


Figure 1: a) Original image $I(\mathbf{x})$. b) Gradient magnitude $\|\nabla I(\mathbf{x})\|$. c) Anisotropism $g(\mathbf{x})$. Straight edges are strongly anisotropic along one direction whereas corner points are not. d) Cornerness $c(\mathbf{x})$. e) Cornerness $c(\mathbf{x})$ with different window settings: precise corner pixels can be identified. f) Direction of anisotropism $\theta(\mathbf{x})$ (on a gray-scale from 0 - black to π - white). It gives the orientation of the edges, but cannot be used within the immediate neighbourhood of corners.

Figure 1 shows an example of the computation of the fields required for estimating cornerness. (a) is the original synthetic image. (b) shows the gradient magnitude. (c) shows the anisotropism which is high along the edges and low at corner points. (d) and (e) show cornerness with different window settings. The orientation $\theta(\mathbf{x})$ which is required at a later stage of the algorithm, is also given, shown as (f).

Extracting the actual corner points is achieved by analysing the histogram of the cornerness image. A small proportion of pixels (ϵ) with sufficiently high values can be regarded as belonging to corners. Each cluster of such points is labelled as one corner, at the position of its point of highest cornerness. Experiments show that some simple edge points may be misclassified as corners this way, especially if ϵ is chosen too high and for noisy images. This, however, has little detrimental effect on the algorithm, since at a later stage these pixels can be easily identified and discarded.

Corner Orientation

The process of evaluating a corner's orientation in our approach mainly involves finding the orientation of the surrounding edges that form the corner. Extracting such information is therefore essentially a local process. We define a function $s_c(\mathbf{x})$ that signifies the likelihood of a pixel being part of the edges associated with a particular corner C. The value of $s_c(\mathbf{x})$ is computed at each point M_i in a small neighbourhood around what has been labelled as a corner point C. The following assumptions about the configuration within corners are used:

- the pixels of the arms of a corner are edge points strongly orientated along one direction
- this direction should be compatible with the direction of $\mathbf{v} = \overrightarrow{CM_i}$

To satisfy the first assumption, we introduce the product $g(\mathbf{x})\|\nabla I(\mathbf{x})\|$ as a good edge detector. The mere gradient would be inconsistently strong at noise spikes, and the mere anisotropism would be misleadingly high on strongly orientated patterns such as a gradient intensity ramp. Therefore, the combination of two identifies edges properly. We will regard the direction of anisotropism $\theta(\mathbf{x})$ to be the direction of the edge points found.

The second assumption is necessary to deal with the problem of corner adjacency. Several edges may be found in the neighbourhood of a corner, they might not all belong to the same structure. Indeed, they could be associated with another corner nearby. If the direction of an edge pixel does not point toward the corner point C, then its value of $s_c(\mathbf{x})$ should be very low. To enforce this, we have introduced the angle α.

$$\alpha = (\widehat{\mathbf{u}, \mathbf{v}}) \tag{6}$$

with $\mathbf{u}$ being the direction of anisotropsim at M_i, derived from θ in Equation (2), and $\mathbf{v} = \overrightarrow{CM_i}$. As shown on figure 2, α is a measure of the difference between the direction of an edge at a pixel M_i and the direction of (CM_i).

Computing the cosine of α provides a good discriminant to establish which neighbouring corner an edge point belongs to. Figure 3 illustrates this property:

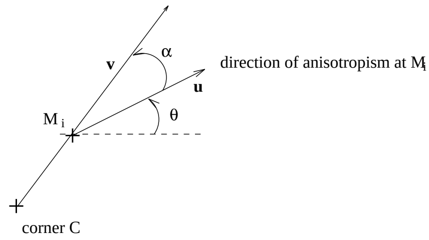


Figure 2: Definition of α. C is a corner pixel, M_i is a point located within its neighbourhood. α is the angular difference between the direction θ of anisotropism at M_i and the direction of (CM_i)

an edge point M is close to two different corner points. When studying the neighbourhood of one of these corners, M_i is discarded, because the direction of the edge it belongs to is not compatible with the direction of (CM_i).

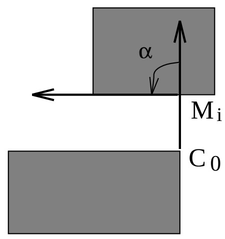


Figure 3: Adjacent corners: when studying corner C_0, M_i is discarded, for at $M_i : cos\,\alpha = 0$

Based on the remarks made above, we subsequently obtain an analytical expression of $s_c(\mathbf{x})$:

$$s_c(\mathbf{x}) = g(\mathbf{x})\|\nabla I(\mathbf{x})\|cos^n\alpha \tag{7}$$

with n being a parameter controlling how strictly direction compatibility should be enforced. In our application, we used the value $n = 3$.

To assist further understanding of Equation (7), Figure 4 gives a schematic summary of the main fields computed. (a) represents the original image of a corner (b) shows anisotropism g($\mathbf{x}$), (c) shows cornerness c($\mathbf{x}$) and (d) shows the value of $s_c(\mathbf{x})$ in a small window around the corner point. Only the pixels belonging to relevant edges are retained.

For each corner point C, the value of $s_c(\mathbf{x})$ is computed within a small neighbouring window Γ and a histogram H is constructed.

$$\forall\theta \in [0; 2\pi[: H(\theta) = \sum_{M\in\Gamma \mid arg(\overrightarrow{CM})=\theta} s_c(\mathbf{x}) \tag{8}$$

The analysis of the histogram gives the dominant directions of the edges associated with the corner. While the calculations seem formidable, the subset of corner

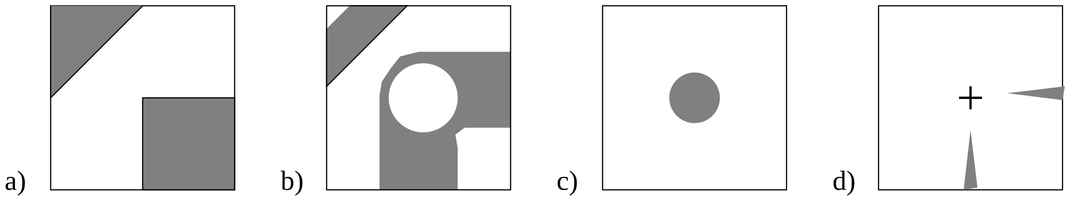


Figure 4: a) Example image $\mathrm{I}(\mathbf{x})$ of a corner. An un-related edge is located within its neighbourhood. b) Anisotropism $\mathrm{g}(\mathbf{x})$: the edges are strongly orientated along one direction whereas the corner point is not. c) cornerness $c(\mathbf{x}) = (1 - g(\mathbf{x}))^m \nabla I(\mathbf{x})$. d) $s_c(\mathbf{x})$: identification of the side points associated with the corner. The edge that is not associated with the corner is discarded.

points in an image is small enough to allow for the efficient implementation of our method without incurring excessive computational cost. One may notice that pixels at the centre of the corner are not taken into account. Indeed, they cannot provide reliable orientation information, because of noise and quantisation errors. Since their value of anisotropism is close to zero, the value of $s_c(\mathbf{x})$ will be small. This automatically removes them from the histogram. Figure 5 illustrates this in the case of a junction: the value of $s_c(\mathbf{x})$ is very small at the junction point whereas the three edges strongly influence the histogram H.

Figure 5: a) Image of a junction. b) Value of $s_c(\mathbf{x})$ in a neighbourhood Γ. No orientation information is extracted from the junction point. The arms are clearly identified.

The orientations of the arms of the corners are estimated by taking the local maxima of the histogram. Each maximum is found in the neighbourhood of a peak corresponding to one arm of the corner. A technique has been designed to determine the number of peaks, *i.e.*, the number of edges defining the corner: two for a simple corner, three or more for a junction. It is worth noting that care must be taken when defining the size of Γ. Making it too small may not provide enough information for an accurate estimation of the orientation. Making it too big would not be computationally expensive but it would include irrelevant information in the search process. Indeed, the corners we identify here are extrapolated but not interpolated like with a Hough transform approach.

Results

The algorithm was tested on synthetic and real-life images. The synthetic images were designed so as to test the detector's ability to cope with different angles, corner adjacency, gray-level distribution and noise level [8]. Figure 6a shows an original synthetic image. Figure 6b shows its cornerness, and to which extent corners can be identified for areas of low contrast. Figure 7a shows the product $g(\mathbf{x})\|\nabla I(\mathbf{x})\|$ as an edge detector. Figure7b shows the orientations of the corners found. The synthetic image on figure 8a was designed to estimate a measure of the accuracy of the detector. The corners found on figure 8b were, on average, less than half a pixel away from their theoretical location, and their orientation was, on average, within a 1.5 degree error range. The detector's ability to cope with noise is shown on figure 8c where a gaussian noise was added to the image before processing (signal-to-noise ratio SNR = 20 dB). It shows that some corners and junctions were missed, but the accuracy of the results remains good: the corners' location is with a .5 pixel error range on average, and their orientation is within a 2.1 degrees error range.

To demonstrate the capacity of the algorithm to deal with real-life images, we have applied it to a picture of a part of the Royal Festival Hall, London, as shown on figure 9a. The corners found are shown on figures 9b and 10a. The potential of the method for edge grouping is illustrated on figure 10b: a simple search algorithm joins corners with compatible directions to obtain grouped contours of the objects.

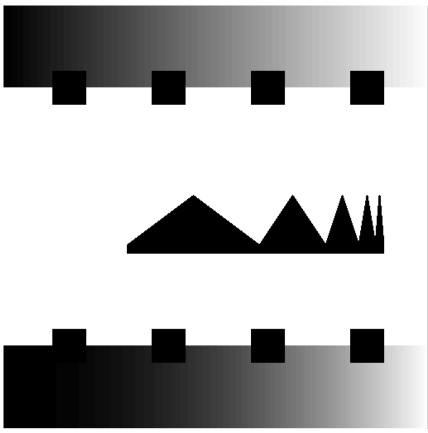
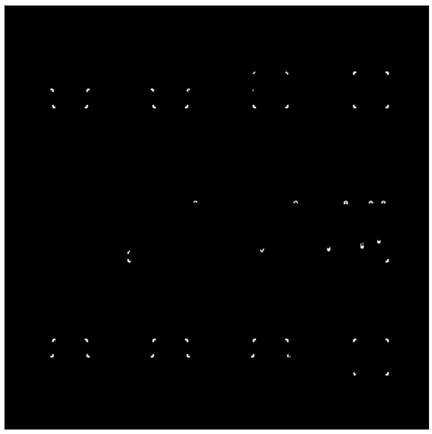
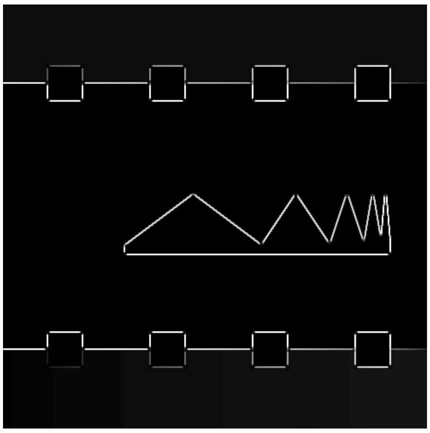
a)  b) 

Figure 6: a) Original synthetic image. The gradient ramp is linear on top of the image, and quadratic at the bottom of the image. b) Cornerness $c(\mathbf{x})$.

a) 

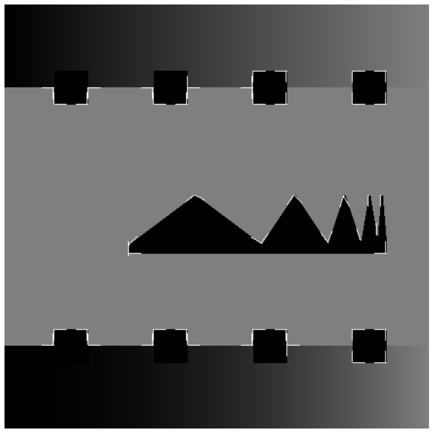
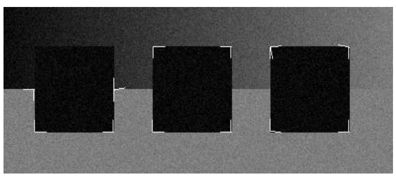
b) 

Figure 7: a) Edge detector $g(\mathbf{x})\|\nabla I(\mathbf{x})\|$ b) Orientations of the corners found. When the angle is narrow one direction only is given. Note that junctions are also identified.

a)

b)



c)

Figure 8: a) Original synthetic image. b) Orientation of the corners detected. The positions found are on average less than half a pixel away from the theoretical location, and the orientation error is on average less than 1.5 degree. c) Orientation of the corners found on the same image with added gaussian noise (signal-to-noise ratio SNR = 20 dB). Some corners are missed, but the accuracy of the measures found remains good (location error less than .5 pixel; orientation error less than 2.1 degrees)

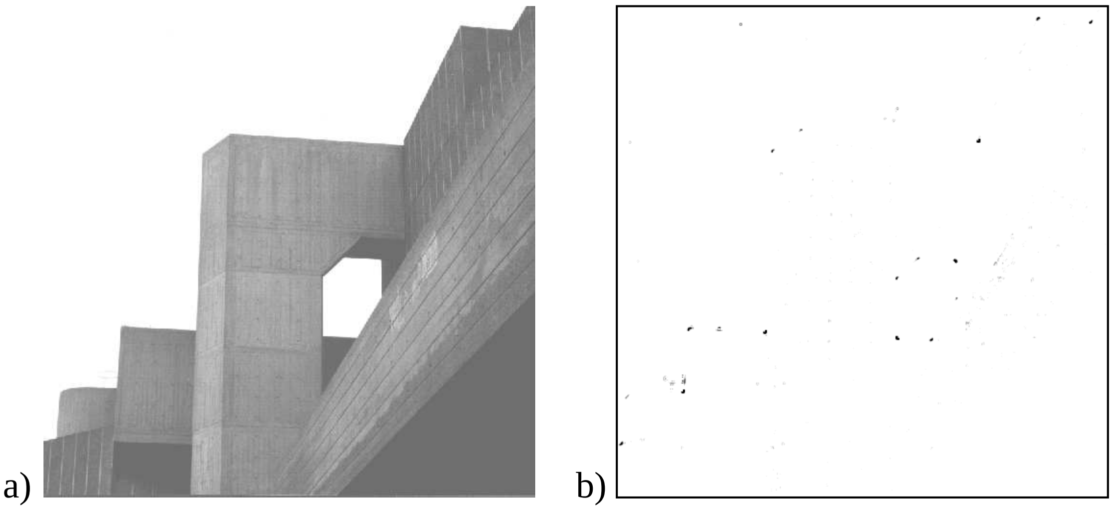


Figure 9: a) Original image of a part of the Royal Festival Hall, London. b) Cornerness $c(\mathbf{x})$ (high values in dark).

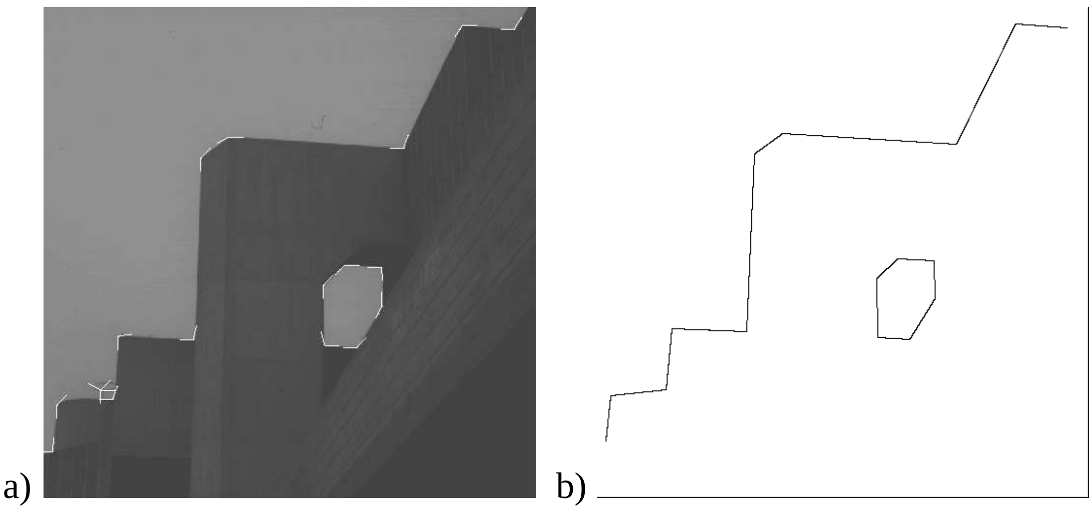


Figure 10: a) Orientations of the detected corners. On a curved corner, the tangential direction is given. b) Joining corners with compatible directions. The knowledge of the orientations of the corners makes grouping edges easy.

Conclusion

In this paper, we have presented a novel method for the detection of corners. Unlike most corner detectors, it also estimates the orientation of the structures associated with each corner. The same method works well with junctions. The information extracted from an image are essential 2D features. Knowing the orientation of corners and junctions as well as their position should facilitate the higher-level processes of image understanding. Indeed, this information provides important clues for the effective interpreting and grouping of low-level image features.

References

[1] Davies E.R. *Machine Vision : Theory, Algorithms, Practicalities.* Academic Press, 1997.

[2] Davies E.R. Application of the generalised Hough transform to corner detection. *IEE Proceedings*, 135:49–54, January 1988.

[3] Yang G.Z., Burger P., Firmin D.N., and Underwood S.R. Structure adaptative anisotropic image filtering. *Image and Vision Computing*, 14:135–145, 1996.

[4] Cooper J., Venkatesh S., and Kitchen L. The dissimilarity corner detector. *IEEE Proceedings on Image Processing*, pages 1377–1382, 1991.

[5] Dreschler L. and Nagel H.H. On the selection of critical points and local curvature extrema of region boundaries for interframe matching. *International Conference on Pattern Recognition*, pages 542–544, 1982.

[6] Kitchen L. and Rosenfeld A. Gray-level corner detection. *Pattern Recognition Letters*, pages 95–102, December 1982.

[7] Zuniga O.A. and Haralick R.M. Corner detection using the facet model. *Proc. Conf. on Patterb Recognition Image Processing*, pages 30–37, 1983.

[8] Rajan P.K and Davidson J.M. Evaluation of corner detection algorithms. *IEEE Proceedings on Image Processing*, pages 29–33, 1989.