Performance Evaluation of Corner Detection Algorithms under Similarity and Affine Transforms

Farahnaz Mohanna and Farzin Mokhtarian
School of Electronic and Electrical Engineering
University of Surrey, Guildford, GU2 7XH, UK

Abstract

This paper evaluates the performance of our corner detector and four popular corner detectors under similarity and affine transforms. The majority of authors of published corner detectors have not used theoretical criteria to measure the stability and accuracy of their algorithms. They usually only illustrate their results on different test images and compare them to the results of other test corner detectors. A few of them use only one criterion. This criterion is the number of matched corners between original and transformed images, divided by the number of corners in the original image. This criterion is flawed since it favours algorithms which find more false corners in input images.
We propose two new criteria to evaluate the performance of corner detectors. Our proposed criteria are *consistency of corner numbers* and *accuracy*. These criteria were measured using many images and experiments such as rotation, uniform scaling, non-uniform scaling and affine transforms. To measure accuracy, we created ground truth based on majority human judgement. The results show that our corner detector performs better under similarity and affine transforms.
Keywords–corner detection, curvature scale space, consistency, accuracy, similarity and affine transforms.

1 Introduction

Our interest in corner detection comes from its use in matching, tracking and motion estimation. A corner detector can be successfully used for these tasks if it has good stability and accuracy. As there is no standard procedure to measure stability and accuracy of corner detectors, we performed a number of experiments to compare the stability and accuracy of our corner detector to four popular corner detectors. Kitchen and Rosenfeld [6], Plessey [5], Susan [14] and Curvature Scale Space (CSS[1]) corner detector [8] were chosen as our test corner detectors.
In the first experiment, we considered a test image as our *original image*. Then numbers and positions of corners were extracted from the original image using

[1] The CSS-based shape descriptor has been selected for MPEG-7 standardization

BMVC 2001 doi:10.5244/C.15.37

our corner detector and four other test detectors. Next, original image was rotated, then number and position of corners in all rotated images were extracted using tested detectors. Finally by comparing corners in original image and corresponding corners in transformed images, the number of matched corners in those transformed images were computed. In the second experiment, we did the same for original image and uniform scaling of this image with ten scale factors. We repeated the same in the third and fourth experiments with non-uniform scaling and affine transform of original image. Final test was performed for computation of accuracy. We propose a new approach for creation of ground-truth in this case. After performing the experiments, definition of appropriate criteria for comparing the results was very important. Appropriate criteria means criteria that show exactly which algorithm is the best for matching and tracking tasks taking into consideration all conditions especially extreme ones. Our criteria have been explained in section 4. These criteria were applied to our corner detector and four other test corner detectors. The results of stability and accuracy of these corner detectors have been illustrated. Overall, the results of consistency in numbers and accuracy show that our corner detector has the best accuracy and stability among the five corner detectors.

The following is the organisation of the remainder of this paper. Section 2 presents an overview of our corner detector and some other corner detectors and describes previous criteria for measuring stability of corner detectors. Section 3 briefly describes our corner detector, New-CSS. The theory underlying our criteria is explained in section 4. In section 5, the results of experiments to determine stability and accuracy of our corner detector in comparison to four other corner detectors are illustrated. The conclusions are presented in section 6.

2 Literature survey

Considerable research has been carried out on corner detection in recent years. The brief review of our corner detector and some other popular corner detectors are presented in this section, followed by the survey of previous criteria for measuring stability of corner detectors.

2.1 Survey of corner detectors

Kitchen and Rosenfeld computed a cornerness measure based on the change of gradient direction along an edge contour multiplied by the local gradient magnitude as follows:

$$C_{KR}(x,y) = \frac{I_{xx}{I_y}^2 - 2I_xI_yI_{xy} + I_{yy}{I_x}^2}{{I_x}^2 + {I_y}^2} \tag{1}$$

The local maximum of this measure isolated corners using a non-maximum suppression applied on gradient magnitude before its multiplication with the curvature. This detector is sensitive to noise with poor localisation and unstable. Plessey cornerness measure is:

$$C_P(x,y) = \frac{< {I_x}^2 > + < {I_y}^2 >}{< {I_x}^2 >< {I_y}^2 > - < I_xI_y >^2} \tag{2}$$

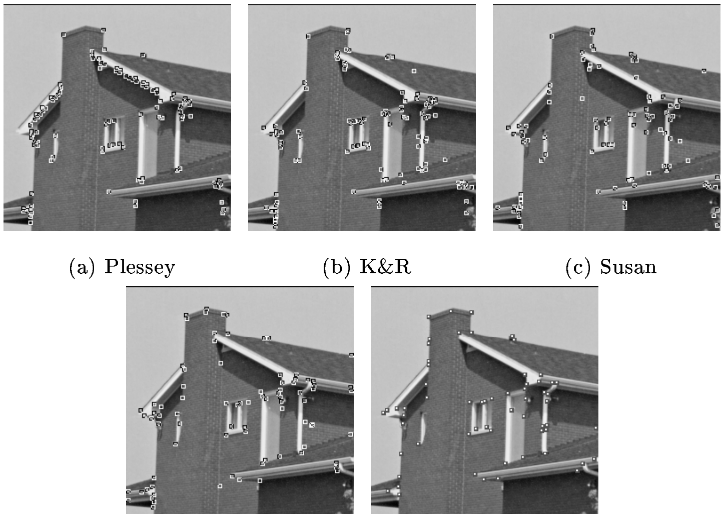
Where he found I_x and I_y using the $(n \times n)$ first-difference approximations to the partial derivatives and calculated ${I_x}^2$, ${I_y}^2$, and $I_x I_y$. Then used a Gaussian smoothing, computed the sampled means$< {I_x}^2 >$,$< {I_y}^2 >$and$< I_x I_y >$using the$(n \times n)$ neighbouring point samples. Sampled mean here is a weighted average of neighbouring values. This algorithm in the case of large Gaussian convolution has not a good localisation. Even more the application of constant-variable false corner response suppression causes it to be unstable. Smith and Brady [14] introduced the Susan algorithm as follows:

Consider an arbitrary pixel in the image and corresponding circular mask around it(the centre pixel shall be called the 'nucleus'). Provided image is a compact region within the mask whose pixels have similar brightness to the nucleus and this area whose be called USAN, an acronym standing for "Univalue Segment Assimilating Nucleus". To find corners they computed the area and the centre of gravity of the USAN, and developed a corner detector based on these parameters. CSS corner detector detected image corners based on the curvature scale space representation. Initial corners were defined as points where image edges have their maxima of absolute curvature. Finally in this algorithm, initial corners were tracked through multiple lower scales. Quddus and Fahmy [11] presented a wavelet-based scheme for detection of corners on 2D planar curves. Arrebola et al. introduced different corner detectors based on local [1] and circular [2] histogram of contour chain code . Chabat et al. [4] introduced an operator for detection of corners based on a single derivative scheme that had already been introduced in [19] by Yang et al. Moravec [9] observed that the difference between the adjacent pixels of an edge or a uniform part of the image is small but at the corner the difference is significantly high in all directions. The idea was later used by Harris [5] to develop Plessey algorithm. Beaudet [3] proposed a determinant operator which has significant values only near corners. Other corner detectors have been proposed in [16, 10, 17].

2.2 Previous criteria of stability

The Majority of published corner detectors have not used properly defined criteria for measuring the stability and accuracy of their corner detectors. They have only demonstrated their results on different images in comparison to other test corner detectors. Some published results on corner detection include studies on the effects of noise and parameter variation on results of their corner detectors. These parameters include Gaussian scale σ {[20], [14], [13], [11]}, σ white noise [20], thresholds {[16], [15]}, signal to noise ratio[4], cross-correlation matching [16], cost function [18] and the width of the gray level transitions in original image [12] but no definition of stability and its results. A few of them have used only one criterion to measure the stability of their corner detectors as follows:

Trajkovic and Hedley [16] used a measure of $k = \frac{N_m}{N_c}$, where N_m and N_c denoted number of strong matches and number of corners in the original image respectively. In terms of stability, a corner detector was better if k is higher. Schmid and Mohr [13], applied the criterion of the ratio of total matches to the number of points extracted. This ratio varies depending on the image as well as on the type of transformation between the images. The problem of both criteria is that if we have an algorithm which marked all of the pixels in one image as corners then k



(a) Plessey (b) K&R (c) Susan

(d) Orig-CSS (e) New-CSS

Figure 1: House image

would become 100%. In other words algorithms with more false corners tend to have a larger number of matched corners. Therefore this criterion is flawed for measuring the stability of corner detectors. Our criteria are Consistency of corner numbers and accuracy. Only with consideration of these criteria together, we can judge correctly on the best corner detectors for tracking and matching tasks.

3 Review of New-CSS

New-CSS is a new corner detection method which is an improvement of an earlier corner detector based on Curvature Scale Space representation. The outline of New-CSS corner detector is as following:

- Extract edges from the original image.
- Extract image edge contours, filling the gaps and finding T-junctions.
- Use different scales of the CSS for contours with different lengths.
- Compute the absolute curvature on the smoothed contours.
- Smooth the absolute curvature function for long contours.
- Detect initial local maxima of the absolute curvature for short contours.
- Detect initial local maxima of the smoothed absolute curvature functions for long contours.
- Consider those local maxima as initial corners that are more than twice as much as one of the neighbouring local minima.

- Track the corners down to the lowest scale for each contour to improve localisation.
- Compare the T-junction corners to the corners found using the curvature procedure to unify close corners.

The first step is to extract edges from the original image using Canny edge detector. Edge detector may produce gaps in some edges. To achieve the best performance of our detector these gaps should be filled at this stage.
The next step is to smooth edge contours by a Gaussian function and compute curvature at each point of the smooth contours. The width of the Gaussian function indicates the scale of the smoothing and must be large enough to remove noise and small enough to retain the real corners. Interestingly, we discovered that this scale should not be the same for all edge contours of the image. While for long contours, a large scale may be suitable, short contours need smaller scale of smoothing. The next stages are to determine corner candidates on smoothed contours, which are normally the local maxima of absolute curvature. However, we noticed that for long contours, the absolute curvature function must be smoothed prior to initial corner selection. The final step is localisation. As a result of smoothing, the edge contours shrink. The locations of corners on the shrunk contours differ from the locations of actual corners on the original contour. For each smoothed contour, we reduce the level of smoothing gradually and track the corners down to the original contour. New-CSS method is robust with respect to noise, and performs better than tested corner detectors(see Fig. 1 as an example). For more details about each stage of New-CSS corner detector described above, refer to [7].

4 Theory

In this section our criteria for measuring the stability and accuracy of corner detectors are defined theoretically. In the following, let N_o be the number of corners in original image(note that $N_o \neq 0$), N_m number of matched corners in each of transformed images when compared to original image corners and N_t number of corners in each of the transformed images.

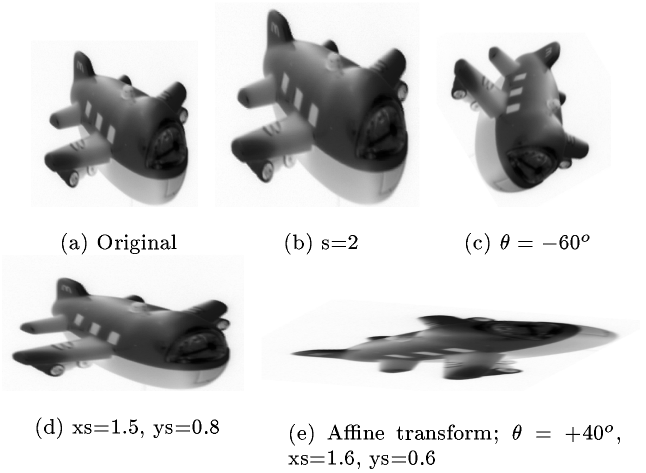
4.1 Consistency

Consistency means corner locations and numbers should be insensitive to the combination of noise, rotation, uniform or non uniform scaling and affine transform. More importantly, corner locations and numbers should not move when multiple images are acquired of the same scene. Previous criterion of consistency has been defined as follows:

$$Consistency = \frac{N_m}{N_o} \tag{3}$$

By this definition, algorithms which find more false corners in input images are favoured since they have higher number of matched corners. Therefore we replace this criterion by two new criteria, *consistency of corner numbers* and *accuracy.* We define the criterion of *consistency of corner numbers* as follows:

$$CCN = 100 \times 1.1^{-|N_t - N_o|} \tag{4}$$



(a) Original

(b) s=2

(c) $\theta = -60^o$

(d) xs=1.5, ys=0.8

(e) Affine transform; $\theta = +40^o$, xs=1.6, ys=0.6

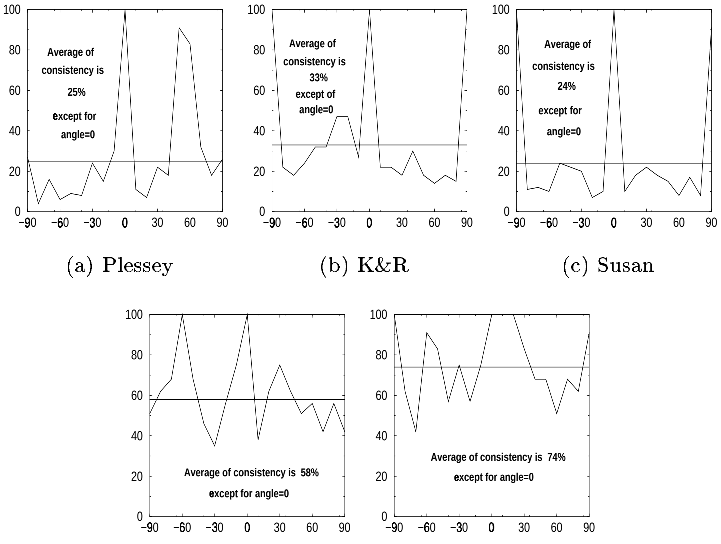
Figure 2: Airplane image under similarity and affine transforms. In this figure s, xs, ys and θ stand for uniform scaling, x-scale and y-scale in non-uniform scaling and rotation parameters respectively.

where CCN stands for "consistency of corner numbers". Since stable corner detectors do not change the corner numbers from original image to transformed images then in terms of consistency, the value of CCN for stable corner detectors should be close to 100%. This criterion for corner detectors with more false corners becomes closer to zero.

4.2 Accuracy

Accuracy means corners should be detected as close as possible to the correct position. In one image, the corner positions and numbers can be different according to different people. Also as there is no standard procedure to measure accuracy of corner detectors we adopted a new approach for creating ground-truth. This approach is based on majority human judgement. To create ground-truth, ten persons who were familiar with the task of corner detection were chosen. None of them were familiar with the algorithm used by our corner detector. We asked them individually to mark the corners of an image. The corners marked by at least 70% of individuals were selected as the ground-truth for that image. The position of a corner in ground-truth was defined as the average of the positions of this corner in individual images marked by those ten persons. We repeated the same for other images. Then by comparing the detected corners using each of five corner detectors to the list of corners in ground-truth, the accuracy was computed as follows:

Let N_o be the number of corners in original image(note that $N_o \neq 0$), N_a the number of matched corners in original image when compared to ground-truth corners



(a) Plessey (b) K&R (c) Susan

(d) Orig-CSS (e) New-CSS

Figure 3: Consistency of corner numbers for rotation

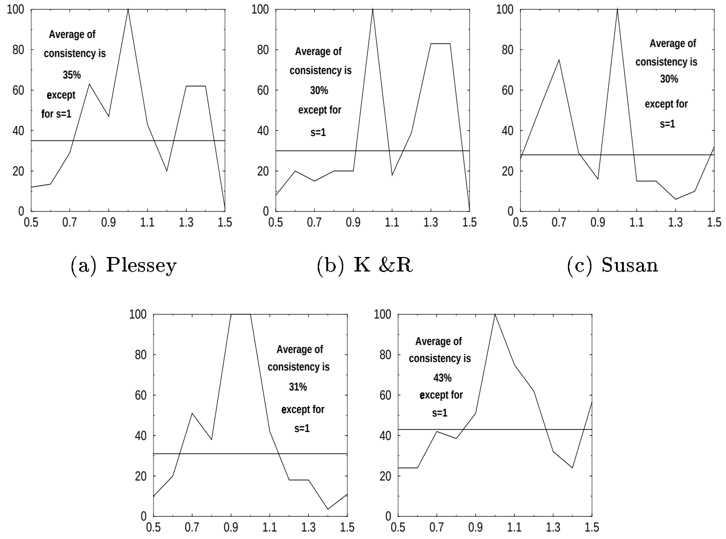
and N_g the number of corners in the ground-truth. The criterion of accuracy is

$$ACU = 100 \times \frac{\frac{N_a}{N_o} + \frac{N_a}{N_g}}{2} \tag{5}$$

where ACU stands for "accuracy". In terms of accuracy, the value of ACU for accurate corner detectors should be close to 100%. Then ACU for corner detectors with lower accuracy is closer to zero. The case of N_o=0 in Eq. 3 and Eq. 5 occurs if test images have no corners or tested corner detectors can not detect any corners. These situations do not arise in practise as only images with many corners are used in experiments and corner detectors under consideration also find many corners in test images.

5 Results and Discussion

We considered the results of our experiments on many images. Examples of image transforms have been illustrated in Fig.2. These experiments were performed as follows:
Experiment 1: In the first experiment, the number and position of corners in original image were extracted using the tested corner detectors. Next, original image was rotated with rotation angle in range of -90^o to $+90^o$. Then the number and position of corners in all rotated images were extracted using the tested detectors.
Experiment 2: In the second experiment, we did the same for original image and uniform scaling of this image with ten scale factors from 0.5 to 1.5.
Experiment 3: We repeated the same in the third experiment with non-uniform



(a) Plessey (b) K &R (c) Susan

(d) Orig-CSS (e) New-CSS

Figure 4: Consistency of corner numbers for uniform scaling

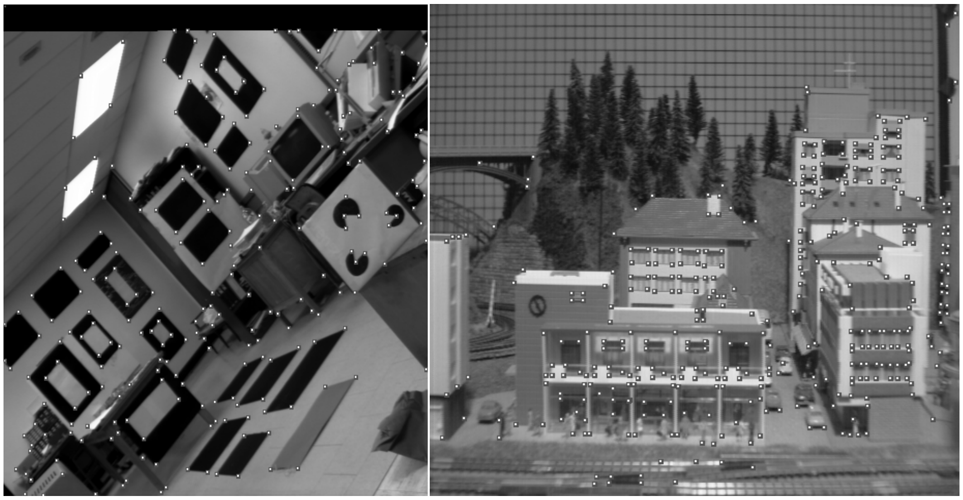
scaling. The variation of x-scale and y-scale in this transform were from 0.8 to 1.0 and 0.5 to 1.5 respectively.

Experiment 4: Affine transform was our fourth experiment that applied a rotation angle of -10^o and $+10^o$ combined with x-scale from 0.8 to 1.0 and y-scale from 0.5 to 1.5 .

After performing our experiments on rotated, uniformly and non-uniformly scaled and affine transformed images, we computed CCN. The results of these computations for rotation and uniform scaling have been illustrated in Fig.3 and Fig.4. The average of consistency in non-uniform and affine transform have been shown in Table 1. Final test was performed for computation of accuracy. We computed accuracy using our database which included leaves, airplanes and fish images and two motion sequences of lab and building. Due to lack of space, just two images of this database have been illustrated in Fig.5. Furthermore, in this figure the corner points of their ground truth also have been shown. The comparison of consistency and accuracy in similarity and affine transforms for tested detectors have been illustrated in Fig.6. Over all, the results of these comparisons show that our corner detector has the better accuracy and stability among these five corner detectors.

Average of CCN for					
	Plessey	K & R	Susan	Orig-CSS	New-CSS
in **non-uniform scaling**	28%	31%	31%	55%	68%
in **affine transform**	14%	11%	9%	42%	51%

Table 1: Average of consistency of corner numbers for tested corner detectors



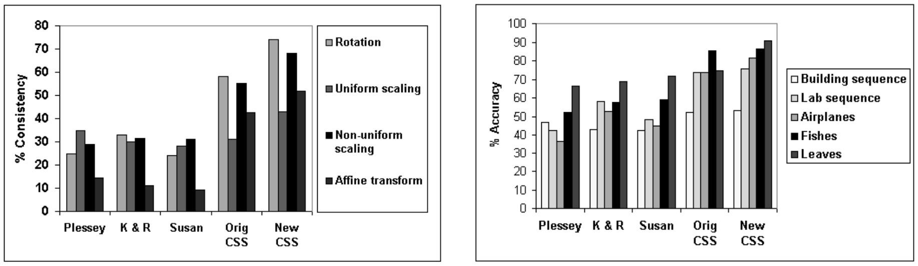
(a) Lab

(b) Building

Figure 5: Examples of test images for computation of accuracy

6 Conclusion

This paper evaluated performance of our corner detector and compared it to performances of several popular corner detectors. First our criteria, consistency of corner numbers and accuracy were defined theoretically. Then they were computed under similarity and affine transforms using many images for these corner detectors. We proposed a new approach for creation of ground-truth used for computation of accuracy. Application of this procedure resulted in the correct number of matched corners due to comparison to ground-truth. As a result, no false corners are taken into account when matching corners between original image and ground-truth. New definition of consistency of corner numbers prevented algorithms which find more false corners in input images from achieving a high score. Overall, the application of these criteria showed that our corner detector produced good results with respect to similarity and affine transforms.



(a) Consistency of corner numbers

(b) Accuracy

Figure 6: Comparison of consistency and accuracy in similarity and affine transforms for tested corner detectors

References

[1] F. Arrebola, A. Bandera, P. Camacho, and F. Sandoval. Corner detection by local histograms of contour chain code. *Electronics Letters*, 33(21):1769–1771, 1997.

[2] F. Arrebola, P. Camacho, A. Bandera, and F. Sandoval. Corner detection and curve representation by circular histograms of contour chain code. *Electronics letters*, 35(13):1065–1067, 1999.

[3] P. Beaudet. Rotationally invariant image operators. In *Int. Joint Conf. Pattern recognition*, pages 579–583, 1978.

[4] F. Chabat, G. Yang, and D. Hansell. A corner orientation detector. *Image and Vision Computing*, 17:761–769, 1999.

[5] C. Harris. Determination of ego-motion from matched points. In *Proc. Alvey Vision Conf.*, Cambridge, UK, 1987.

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

[7] F. Mokhtarian and F. Mohanna. Enhancing the curvature scale space corner detector. In *Proc 12th Scandinavian Conference on Image Analysis*, Bergen, Norway, 2001.

[8] F. Mokhtarian and R. Suomela. Robust image corner detection through curvature scale space. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 20(12):1376–1381, 1998.

[9] H. Moravec. Towards automatic visual obstacle avoidance. In *Proc Int. Joint Conf. Artificial Intelligence*, page 584, 1977.

[10] A. Noble. Finding corners. *Image and Vision Computing*, 6:121–128, 1988.

[11] A. Quddus and M. Fahmy. Fast wavelet-based corner detection technique. *Electronics Letters*, 35(4):287–288, 1999.

[12] K. Rohr. Recognizing corners by fitting parametric models. *Computer Vision*, 9(3):213–230, 1992.

[13] C. Schmid and R. Mohr. Matching by local invariants. Technical report, The French National Institute for research in computr science and control, 1995.

[14] S. Smith and J. Brady. A new approach to low-level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[15] J. Sporring, O. Olsen, M. Nielsen, and J. weickert. Smoothing images creates corners. *Image and Vision Computing*, 18:261–266, 2000.

[16] M. Trajkovic and M. Hedley. Fast corner detection. *Image and Vision Computing*, 16(2):75–87, 1998.

[17] H. Wang and M. Brady. A practical solution to corner detection. In *Proc International Conference on Image Processing*, volume 1, 1994.

[18] X. Xie, R. Sudhakar, and H. Zhuang. Corner detection by a cost minimization approach. *Pattern Recognition*, 26(8):1235–1243, 1993.

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

[20] Z. Zheng, H. wang, and E. Teoh. Analysis of gray level corner detection. *pattern recognition Letters*, 20:149–162, 1999.