Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Vision and Image Understanding 102 (2006) 81–94
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

# Performance evaluation of corner detectors using consistency and accuracy measures

Farzin Mokhtarian *, Farahnaz Mohanna

*Centre for Vision, Speech and Signal Processing, School of Electronics and Physical Sciences, University of Surrey, Guildford, GU2 7XH, UK*

Received 19 January 2005; accepted 15 November 2005
Available online 28 December 2005

**Abstract**

This paper evaluates the performance of several popular corner detectors using two newly defined criteria. The majority of authors of published corner detectors have not used theoretical criteria to measure the consistency and accuracy of their algorithms. They usually only illustrate their results on a few test images and may compare the results visually to the results of other corner detectors. Some authors have proposed various criteria for performance evaluation of corner detection algorithms but those criteria have a number of shortcomings. We propose two new criteria to evaluate the performance of corner detectors. Our proposed criteria are *consistency* and *accuracy*. These criteria were measured using several test 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 the enhanced CSS corner detector performs better according to these criteria.


*Keywords:* Corner detection; CSS; Performance evaluation; Consistency; Accuracy

## 1. Introduction

Interest in corner detection is based on its use in matching, tracking, and motion estimation. A corner detector can be successfully used for these tasks if it has good consistency and accuracy. As there is no standard procedure to evaluate the performance of corner detectors, we have proposed two new criteria to evaluate the performance of corner detectors. These criteria are referred to as *consistency* and *accuracy*. We have also carried out a number of experiments to compare the consistency and accuracy of the enhanced CSS corner detector to four popular and frequently used corner detectors. Kitchen and Rosenfeld [28], Plessey [23], Susan [55], and curvature scale space (CSS)[1] corner detector [38] were chosen as our test corner detectors. Note that the CSS corner detector is a contour-based detector whereas the other detectors are neighborhood-based detectors. We believe ultimately it is irrelevant whether a specific corner detector is classified as one type or the other. When a corner detector is designed, the designer is free to make use of any technique or methodology which he/she considers advantageous. For the sake of performance evaluation, all corner detectors are subjected to the same evaluation criteria. What matters is how a corner detector performs according to the criteria, not which methods are employed in the design of that detector.

It should be pointed out that we are specifically interested in performance evaluation of corner detectors, and not the more general feature point detectors or interest point detectors. An advantage of corners is that they correspond to human intuition regarding visually distinguishable feature points. This property can be exploited to define ground truth which is essential in the measurement of accuracy. Essentially, a corner is an image point where 2D change can be observed in the image, or where the boundary of a distinguishable region in the image undergoes a sharp change in

* Corresponding author. Fax: +44 1483 686031.
*E-mail address:* F.Mokhtarian@surrey.ac.uk (F. Mokhtarian).
[1] The CSS-based shape descriptor has been selected for MPEG-7 standardization.

1077-3142/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2005.11.001

direction. On the other hand, feature points or interest points do not necessarily correspond to visually significant points, and therefore it may be very difficult to assess the performance of a feature point detector or an interest point detector.

The first set of experiments were concerned with measuring the consistency of test corner detectors. A number of corners were extracted from a test image using the test corner detectors. The test image was then subjected to a number of transformations including rotation, scaling, and affine transforms. The test corner detectors were then applied to the transformed images to recover the corresponding corners. The corners obtained from these experiments were used to compute values for the consistency criterion.

The second set of experiments were concerned with measuring the accuracy of test corner detectors. We have proposed a new approach for the creation of ground-truth which is crucial in the computation of accuracy.

The following is the organisation of the remainder of this paper. Section 2 presents an overview of a number of corner detection algorithms as well as previously proposed criteria for performance evaluation of corner detectors. Section 3 briefly describes the CSS method, followed by a review of the original CSS corner detector in Section 4. The shortcomings of the original CSS method are explained in Section 4.1. Section 5 briefly describes the extended CSS corner detector, ECSS. The theory underlying our criteria is explained in Section 6. In Section 7, the results of experiments to determine the consistency and accuracy of several popular corner detectors are illustrated. The conclusions are presented in Section 8.

## 2. Literature survey

This section presents a review of the existing literature on corner detection algorithms followed by a survey of previous criteria proposed for measuring performance evaluation of corner detectors.

### 2.1. Survey of corner detectors

A substantial number of corner detectors have been proposed by vision researchers. These methods can be divided into two main classes: contour based and intensity based. Contour based methods first recover image contours and then search for curvature maxima or inflection points along those contours. Intensity based methods estimate a measure which is intended to indicate the presence of a corner directly from the image greyvalues. In the following sections, we present corner detection methods for each of those categories.

#### 2.1.1. Contour based methods

Asada and Brady [5] extracted corner points for 2D objects from planar curves. Changes in the curvature functions of those contours are classified into several categories such as cranks, corners, endings, joins, and dents. Quddus and Fahmy [45] presented a wavelet-based scheme for detection of corners on 2D planar curves. Arrebola et al. [3] introduced different corner detectors based on local and circular [4] histogram of contour chain code. Zhang and Zhao [63] considered a parallel algorithm for detecting dominant points on multiple digital curves. Gallo et al. [22] detected, localized, and classified corners in digital closed curves based on correct estimation of support regions for each point. They computed multiscale curvature to detect and localize corners. As a further step, they classified corners into seven distinct types using a set of rules, which describe corners according to preset semantic patterns. Peng et al. [43] introduced a boundary-based corner detection method using wavelet transform for its ability for detecting sharp variations. Pikaz and Dinstein [44] proposed a method based on a decomposition of noisy digital curves into a minimal number of convex and concave segments. The detection of corners is then based on properties of pairs of sections determined in an adaptive way. Cooper et al. [15] presented a detector which tests the similarity of image patches along the contour direction to detect turns in the image contour. The CSS [38] and ECSS [36] corner detectors also belong to the category of contour based methods. These methods have been reviewed in Sections 4 and 5, respectively.

#### 2.1.2. Intensity based methods

Kitchen and Rosenfeld [28] computed a cornerness measure based on the change of gradient direction along an edge contour multiplied by the local gradient magnitude as follows:

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

The local maximum of this measure isolated corners using a non-maximum suppression method applied to the gradient magnitude before its multiplication with the curvature. This detector is sensitive to noise and shows poor localisation. Plessey [23] cornerness measure is

$$C_{\mathrm{P}}(x,y) = \frac{\langle I_x^2\rangle + \langle I_y^2\rangle}{\langle I_x^2\rangle\langle I_y^2\rangle - \langle I_xI_y\rangle^2}, \tag{2}$$

where $I_x$ and $I_y$ were found using the $(n\times n)$ first-difference approximations to the partial derivatives and calculated $I_x^2$, $I_y^2$, and $I_xI_y$, then applied Gaussian smoothing, and computed the sampled means $\langle I_x^2\rangle$, $\langle I_y^2\rangle$, and $\langle I_xI_y\rangle$ using the $(n\times n)$ neighbouring point samples. This algorithm does not show good localization in the case of large Gaussian convolutions. Furthermore, the application of constant-variable false corner response suppression causes it to be unstable. Baumberg [9] used Plessey corners at a set of scales and ordered those corners based on a scale normalized corner strength.

Smith and Brady [55] introduced the Susan algorithm as follows: consider an arbitrary image pixel and the

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, this area will 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. Chabat et al. [13] introduced an operator for detection of corners based on a single derivative scheme introduced in [62] by Yang et al. In [64], Zheng et al., proposed a gradient-direction corner detector that was developed from the Plessey corner detector. Moravec [39] 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. Beaudet [10] proposed a determinant operator which has significant values only near corners. Drescher and Nagel [18] used Beaudet's concepts in their detector. Lai and Wu [30] considered edge-corner detection for defective images. Tsai [59] proposed a method for boundary-based corner detection using neural networks. Fidrich and Thirion [21] extracted corners as the intersection of two iso-surfaces. To extract corners they used an algorithm, based on iso-surface techniques, which finds the corresponding singularities in scale space automatically. Ji and Haralick [26] presented a technique for corner detection with covariance propagation. Lee and Bien [31] applied fuzzy logic to corner detection. Chen and Rockett [14] utilized Bayesian labelling of corners using a grey-level corner image model in. Kohlmann [29] proposed corner detection in natural images based on the 2D Hilbert transform. Nobel [41] proposed a solution to finding T, X, and L junctions which have a local 2D structure. Wu and Rosenfeld [60] proposed a technique which examines the slope discontinuities of the $x$ and $y$ projections of an image to find the possible corner candidates. Paler et al. [42] proposed a technique based on features extracted from the local distribution of grey level values. Rangarajan et al. [47] proposed a detector which tries to find an analytical expression for an optimal function whose convolution with the windows of an image has significant values at corner points. Shilat et al. [52] worked on ridge corner detection and correspondence. Nassif et al. [40] considered corner location measurement. Sohn et al. [56] proposed a mean field annealing approach to corner detection. Davies [16] applied the generalised Hough transform to corner detection. Trajkovic and Hedley [58] described a corner detection algorithm based on the property of corners that the change of image intensity should be high in all directions. Consequently, the corner response function was computed as a minimum change of intensity over all possible directions. In [20], the slit rotational edge-feature detector (SRED) method was modified using weighted and interpolated SRED which is more independent on the edge directions. Bankman and Rogala [8] presented a non-linear corner detection algorithm that does not require prior image information or any threshold setting. In [19], Elias and Laganiere presented an approach to detect corners and determined their orientations through a data structure similar to pyramids but with circular levels. They referred to the data structure as a cone and the operation started at the top level where only one node exists. If the node is inhomogeneous, it will be split into two nodes forming the next lower level of the cone. Splitting continues until all nodes become homogeneous. At the base and according to a threshold, similar nodes are grouped together to shape the orientation of the corner. In [6] two oriented cross operators called Crosses as Oriented Pair were used to detect corners, which provided useful information to extract low-level features due to its characteristics, preference for edge with different direction, and simple direction determination. Quddus and Gabbouj [46] presented a technique for wavelet-based corner detection using singular value decompusition (SVD). In their method, SVD facilitated the selection of global natural scale in discrete wavelet transform and the natural scale was defined as the level associated with the most prominent eigenvalue. The eigenvector corresponding to the dominant eignvalue was considered the optimal scale. The corners were detected at the locations corresponding to modulus maxima. Zuniga and Haralick [66] proposed a corner detector based on the facet model. Ando [2,1] constructed a covariance matrix of the gradient vector and applied the canonical correlation analysis to it.

### 2.2. Previous criteria for performance evaluation

The majority of authors of published corner detectors have not used properly defined criteria for performance evaluation of their corner detectors. They have only demonstrated their detectors on a few images. Occasionally their results appear next to results from other corner detectors, but this is not always the case.

Some published results on corner detection include studies on the effects of noise and parameter variation on the proposed corner detectors. These parameters include Gaussian scale $\sigma$ [64,55,45], thresholds [58,57], signal-to-noise ratio [13], cross-correlation matching [58], cost function [61], and the width of the gray level transitions in original image [48].

Only a relatively small number of authors have proposed criteria for evaluation of edge or corner detectors. The existing methods can be grouped into six categories as following.

#### 2.2.1. Ground truth verification

Ground truth is generated by human judges, and can be used to determine the undetected features (false negatives) and the false positives. Bowyer et al. [11] used human generated ground truth to evaluate edge detectors. Their evaluation criterion is the number of false positives with respect to the number of unmatched edges which is measured for varying input parameters. Structured outdoor scenes such as airports and buildings were used in their experiments.

However, as far as we are aware, no work has been carried out on ground truth verification of corner detectors prior to this paper.

*2.2.2. Visual inspection*

Methods employing visual inspection rely directly on humans evaluating the results, so they can be very subjective. Heath et al. [24] evaluated edge detectors by making use of a visual rating score which indicates the perceived quality of the edges for identifying an object. Lopez et al. [34] defined a set of visual criteria to evaluate a number of ridge and valley detectors in the context of medical images.

*2.2.3. Localization accuracy*

This criterion measures whether a corner point is accurately located at a specific 2D location [54]. Evaluation often requires the knowledge of precise 3D properties, which limits the evaluation to simple scenes. In fact, localization accuracy is often measured by verifying that a set of 2D image points matches with the known set of corresponding 3D scene points. For example, Heyden and Rohr [25] extracted sets of points from images of polyhedral objects, and used projective invariants to compute a manifold of constraints on those points. Baker and Nayar [7] proposed a number of global criteria each of which corresponded to a specific very simple scene, and is measured using the extracted edgels. Kakarala and Hero [27] used statistical parameter estimation techniques to derive bounds on achievable accuracy in edge localization.

*2.2.4. Repeatability under various transformations*

In this approach, corner detectors are evaluated to determine how repeatable the results are under transformations (similarity or affine) of the image or the scene. For example, Trajkovic and Hedley [58] used a measure of

$$k = \frac{N_m}{N_c},$$

where $N_m$ and $N_c$ denoted the number of corner matches (between the original image and the transformed image), and the number of corners in the original image, respectively. A corner detector was considered better if $k$ is higher. Schmid et al. [51] applied the criterion of the ratio of total matches to the number of points extracted. An important shortcoming of both approaches is that if we have an algorithm which marked all of the pixels in one image as corners then $k$ would become 100%. In other words, algorithms with more false corners tend to have a larger number of matched corners.

*2.2.5. Theoretical analysis*

Methods using this approach study the behavior of feature detectors using theoretical feature models. However, such methods are limited since they are applicable only to very specific features. Examples follow: Deriche and Giraudon [17] studied the behavior of three corner detectors using an L-corner model. Their study allows them to estimate the localization bias. Rohr [49,50] also carried out a similar analysis for L-corners with aperture angles in the range [0°–180°].

*2.2.6. Evaluation based on specific tasks*

Edge detectors have occasionally been evaluated through specific tasks. The reasoning is that feature detection is not the end goal but only the input for further processing. Hence, the best performance measure is the quality of the input it prepares for the next stage. While this argument is correct to some extent, evaluations based on a specific task and a specific system are difficult to generalize and therefore of limited value. An example of this approach is that of Shin et al. [53] in which a number of edge detectors were compared using an object recognition algorithm. Test images show cars under different lighting conditions and against varying backgrounds.

## 3. The curvature scale space technique

The CSS technique is suitable for extraction of curvature features such as curvature extrema or zero-crossing points from an input contour at multiple scales. The curve $\Gamma$ is first parametrized by the arc length parameter $u$:

$$\Gamma(u) = (x(u), y(u)). \tag{3}$$

An *evolved version* $\Gamma_\sigma$ of $\Gamma$ is then computed. $\Gamma_\sigma$ is defined by [37]:

$$\Gamma_\sigma = (\mathcal{X}(u,\sigma), \mathcal{Y}(u,\sigma)), \tag{4}$$

where

$$\mathcal{X}(u,\sigma) = x(u) \otimes g(u,\sigma) \qquad \mathcal{Y}(u,\sigma) = y(u) \otimes g(u,\sigma).$$

Note that $\otimes$ is the convolution operator and $g(u,\sigma)$ denotes a Gaussian of width $\sigma$. The process of generating evolved versions of $\Gamma$ as $\sigma$ increases is referred to as the *evolution* of $\Gamma$. To find curvature zero-crossings or extrema from evolved versions of the input curve, one needs to compute curvature accurately and directly on an evolved version $\Gamma_\sigma$. Curvature $\kappa$ on $\Gamma_\sigma$ is given by [37]:

$$\kappa(u,\sigma) = \frac{\mathcal{X}_u(u,\sigma)\mathcal{Y}_{uu}(u,\sigma) - \mathcal{X}_{uu}(u,\sigma)\mathcal{Y}_u(u,\sigma)}{(\mathcal{X}_u(u,\sigma)^2 + \mathcal{Y}_u(u,\sigma)^2)^{1.5}}, \tag{5}$$

where the first and second derivatives of $\mathcal{X}$ and $\mathcal{Y}$ can be computed from the following:

$$\mathcal{X}_u(u,\sigma) = x(u) \otimes g_u(u,\sigma) \qquad \mathcal{X}_{uu}(u,\sigma) = x(u) \otimes g_{uu}(u,\sigma),$$

$$\mathcal{Y}_u(u,\sigma) = y(u) \otimes g_u(u,\sigma) \qquad \mathcal{Y}_{uu}(u,\sigma) = y(u) \otimes g_{uu}(u,\sigma).$$

## 4. Overview of the original CSS corner detector

The following is a summary of the algorithm used by Mokhtarian and Suomela [38] to detect corners in an image:

- Apply the Canny edge detector [12] to the gray level image and obtain a binary edge image.
- Extract image edge contours from the output of the Canny edge detector, fill the gaps and find T-junctions.
- Compute curvature at a high/coarse scale, $\sigma_{high}$, for each edge contour.
- Consider those local maxima as initial corners whose absolute curvature are above threshold $t$ and twice as much as one of the neighboring local minima.
- Track the corners to the lowest/finest scale to improve localisation.
- Compare the T-junctions to other corners and remove one of the two corners which are very close.

In this algorithm, the first step is to extract edges from the original image using the Canny edge detector. The edge detector may produce gaps in some edges. To achieve the best performance of the CSS detector, these gaps should be filled at this stage. When the edge-extraction method arrives at the endpoint of a contour, it performs two checks:

- If the endpoint is nearly connected to another endpoint, fill the gap and continue the extraction.
- If the endpoint is nearly connected to an edge contour, but not to another endpoint, mark this point as a T-junction corner.

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. The third step is to determine corner candidates on smoothed contours, which are normally the local maxima of absolute curvature. The fifth step is localization. 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, the level of smoothing is reduced gradually and the corners are tracked down to the original contour. The final step is intended to ensure that any T-junctions missed by the Canny detector are added to the final set without any duplications.

### 4.1. Shortcomings of the CSS corner detector

The second step of the CSS algorithm is to smooth edge contours by 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. We found 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 remedy to this shortcoming is to choose different scale of smoothing for contours with different lengths as described in the next section. The third stage of the CSS algorithm is 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. Furthermore, performance of the CSS detector depends on the selection of threshold value, $t$. The proper value of $t$ may change from one image to another. It is also subject to change for a particular image which transforms under rotation or scaling. The ECSS corner detector mainly tackles these problems. It is described in more detail in the following section.

## 5. Review of ECSS

ECSS is a new corner detection method which is an improvement of the original CSS detector. The outline of ECSS [36] 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 to smooth 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 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 new steps are described in detail in the remainder of this section.

### 5.1. Using different scales of CSS

Based on the number of points on each image edge contour, ECSS categorises all the image edge contours into three categories: *long*, *medium*, and *short*. Through many experiments, we defined contours which consist of less than 200 points as short, contours which consist of 200–600 points as medium and contours with more than 600 points as long. Note that these values did not change in the program for different test images. Note that the classification of contours may depend on the size of the image, so for example, a contour considered medium in an image may be considered long when that image is scaled up. However,

we found through many experiments that these definitions worked well, and produced high-quality corner detection output, even with scaling of input images. We believe the explanation is that as contours grow longer (through scaling or otherwise), more smoothing should be applied to them to control the number of corners recovered from them. Furthermore, we consider $\sigma_{high}$ as 4, 3, and 2 for long, medium, and short contours, respectively. These values have also been selected through many experiments. Evaluations carried out for the ECSS corner detector in Section 7 shows that these selections perform well for a large number of images. As a result, short contours are not smoothed too far which might remove their corners and long contours are sufficiently smoothed. In Fig. 1, one of the test images and its edge contours with two marked contours, short C1 and long C2 have been illustrated. The effect of selecting different scales in computation of absolute curvature for long and short contours can be seen in Fig. 2.

In Fig. 2A, the absolute curvature of C2 with $\sigma_{high} = 4$ has fewer false maxima due to noise in comparison to Fig. 2B. Fig. 2B shows the computation of absolute curvature of C2 with $\sigma_{high} = 2$. Also Fig. 2C illustrates the computation of absolute curvature of C1 with $\sigma_{high} = 4$. If we use the local maxima of this absolute curvature function for corner detection, only two corners are detected. Therefore, using high/coarse scale for smoothing short contours removes some local maxima of absolute curvature of these contours that correspond to real corners. Fig. 2D shows the absolute curvature of C1 with $\sigma_{high} = 2$. It can be seen that four corners can be recovered from this figure. Note that the issue of automatic selection of detection scales has also been addressed in [32] and [33].

### 5.2. Smoothing the absolute curvature of long contours

In this stage, after smoothing edge contours for computation of absolute curvature, some false maxima due to

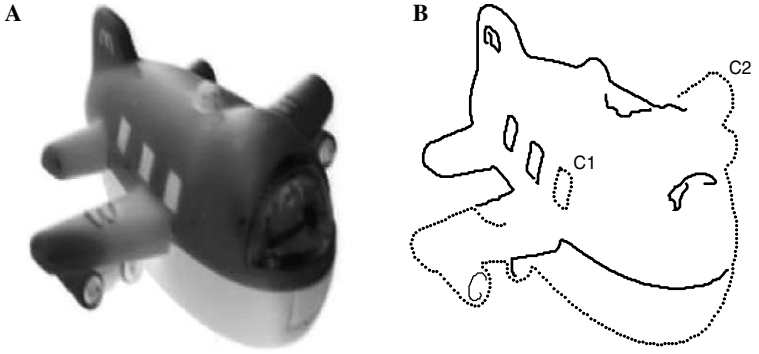


Fig. 1. Two marked contours in edge contours of test image, C1, short contour and C2, long contour. (A) Test image. (B) Edge contours of (A).

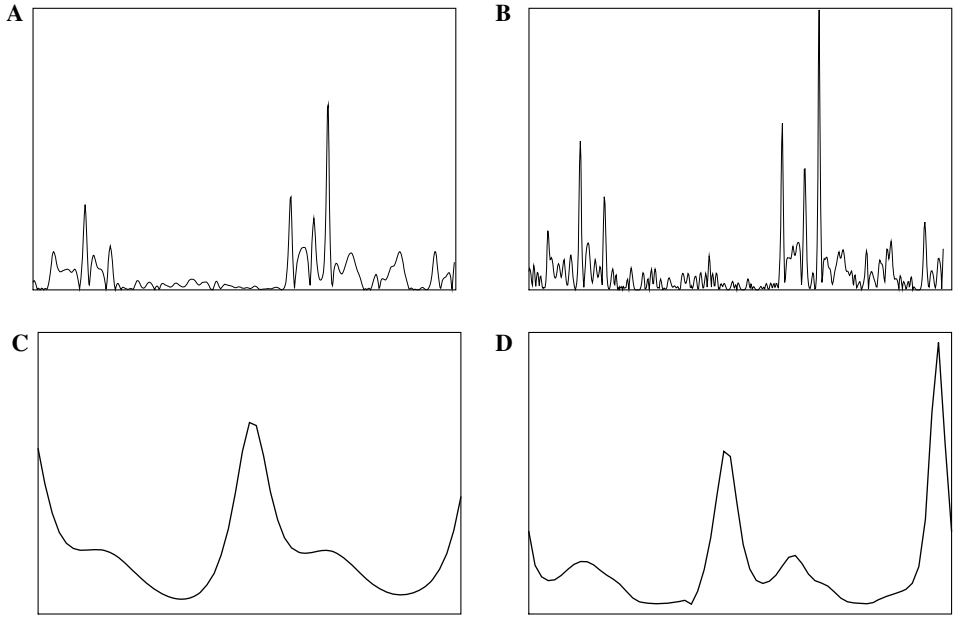


Fig. 2. Computation of absolute curvature. (A) C2, $\sigma_{high} = 4$. (B) C2, $\sigma_{high} = 2$. (C) C1, $\sigma_{high} = 4$. (D) C1, $\sigma_{high} = 2$.

noise can still be seen (see Fig. 2A). The simplest solution is to compute the absolute curvature of contour C2 using higher $\sigma$ such as 8. But as mentioned in Section 4, if a higher value of $\sigma$ is chosen not only false corners but also many real corners are removed as well. Therefore, the solution employed at this stage is to smooth the absolute curvature function of long contours using $\sigma = 4$. This has been illustrated in Fig. 3A. Comparing Fig. 3A to Fig. 2A, after smoothing, many false maxima of absolute curvature are removed. In this step, if the curvature function of short contours is smoothed, as seen in Fig. 3B, a number of real corners are lost. This can be seen in Fig. 3B where only two local maxima remain that indicate two corners on C1, whereas C1 is the window of airplane with four corners. The final criterion for removing false corners, after initialising local maxima points, is to compare the initial local maxima with two neighbouring local minima. The curvature of a corner should be more than twice as much as the curvature of one of the neighbouring local minima.

Applying this criterion, false corners such as 1, 2, and 3 are removed after comparison to the nearest local minima of absolute curvature in Fig. 3A. The positions of initial corners of Figs. 3A and 2D after applying this criterion have been illustrated in Figs. 4A and 4B respectively. The method finds four corners on C1 and no false corners on C2. C2 should have nine corners as seen in Fig. 4A, and C1 should have four corners as seen in Fig. 4B. Note that for short contours ECSS first computes absolute curvature with $\sigma_{high} = 2$, then applies the final criterion discussed above.

### 5.3. Tracking

After the initial corner points are located, tracking is applied to the detected corners. As the corners were detected at scale $\sigma_{high}$ the corner localisation might not be ideal. We compute curvature at a lower scale and examine the corner candidates in a small neighbourhood of the previous corners. Corner locations are updated, if needed, in this neighbourhood. Note that if initial corners on one contour are extracted at $\sigma_{high} = 4$, tracking for this contour can be accomplished at $\sigma = 3$, $\sigma = 2$, and $\sigma_{final} = 1$. If initial corners are extracted at $\sigma_{high} = 2$, tracking can be accomplished at $\sigma_{final} = 1$. The localisation of corners for contour C2 and C1 after tracking have been shown in Figs. 5A and B, respectively. This process results in very good localisation. The number of corners is determined at the initial $\sigma_{high}$ and tracking only changes the localisation, not the number of corners. Since the location of corners do not change substantially during CSS tracking, only a few other curvature values need be computed.

### 5.4. Unifying close corners

As described before, corners are detected using ECSS technique taking T-junctions into consideration. In some cases the two methods mark the same corner. The final part of ECSS is to examine T-junctions and the corners that result from tracking in Section 5.3. If they are very close to each other, the T-junction corners are removed. The ECSS method is robust with respect to noise, and performs

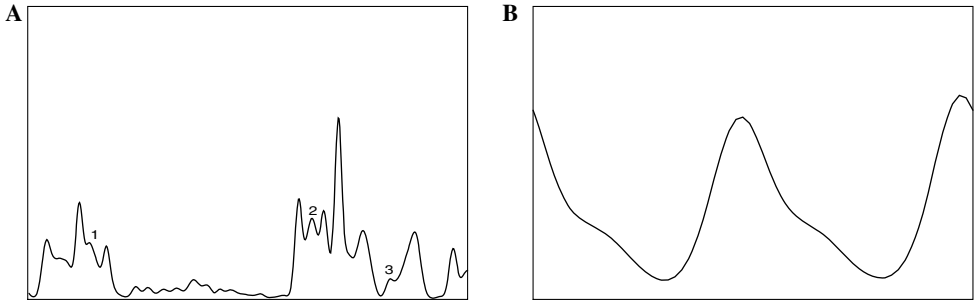


Fig. 3. Smoothing of absolute curvature function. (A) C2, $\sigma = 4$. (B) C1, $\sigma = 4$.

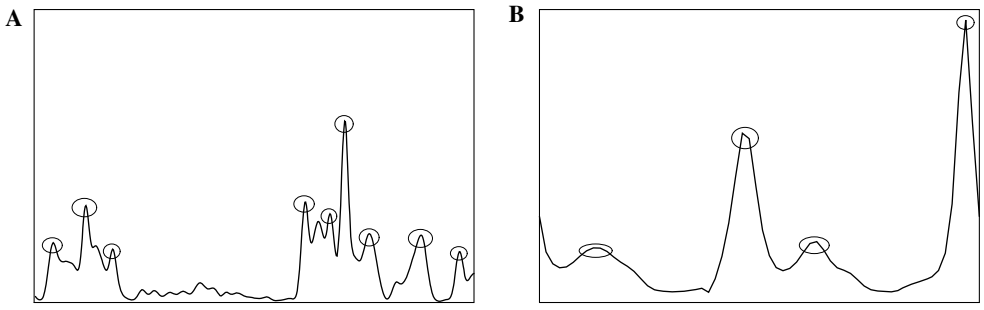


Fig. 4. Applying the criterion of comparing the initial local maxima with two neighbouring local minima. (A) On long contour C2. (B) On short contour C1.

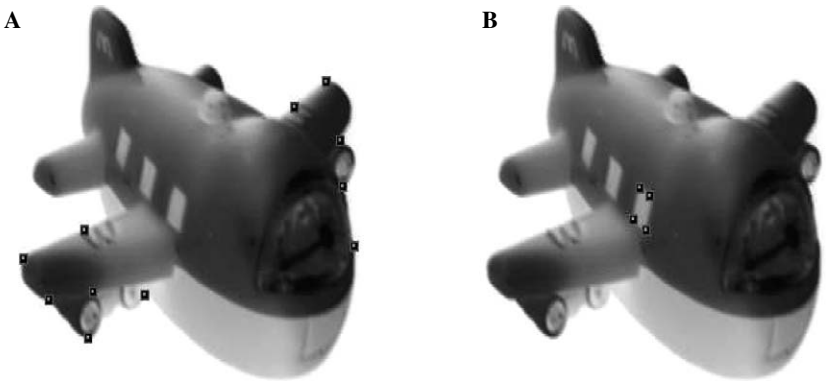


Fig. 5. Tracking corners through low scales for good localisation. (A) Real corners of C2. (B) Real corners of C1.

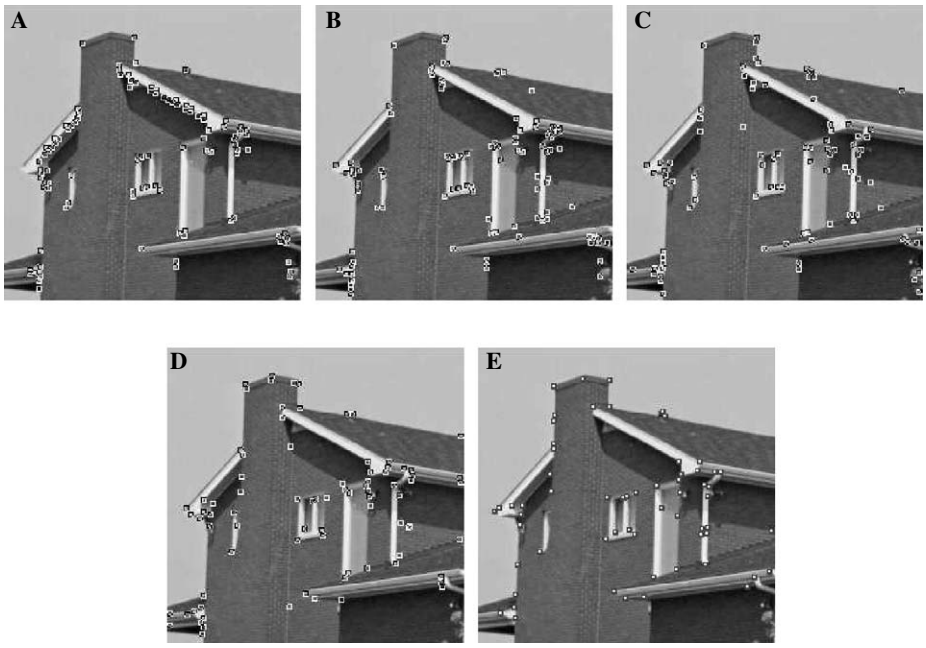


Fig. 6. Corner detection on the house image. (A) Plessy. (B) K and R. (C) Susan. (D) CSS. (E) ECSS.

better than tested corner detectors as shown in Section 7 (see Fig. 6). Note that this figure is not intended as the proof of better ECSS performance. Rather the argument in support of better ECSS performance is presented in Section 7.

## 6. New criteria for performance evaluation

In this section new criteria for measuring the consistency and accuracy of corner detectors are defined theoretically.

### 6.1. Consistency

Consistency means corner numbers should be invariant to the combination of noise, rotation, uniform or non-uniform scaling and affine transform. By noise we mean naturally occurring noise in images such as camera noise and discretization errors. No artificial noise was added to input images. We define the criterion of *consistency of corner numbers* as follows:

$$\mathrm{CCN} = 100 \times 1.1^{-|N_t - N_o|}, \tag{6}$$

where CCN stands for *consistency of corner numbers*. Since consistent corner detectors do not change the corner numbers from original image to transformed images then the value of CCN for stable corner detectors should be close to 100%. Any differences between the number of corners in the original image ($N_o$) and the number of corners in the transformed image ($N_t$), causes CCN to drop below 100% as $|N_t - N_o|$ grows larger. CCN is close to zero for corner detectors with many false corners. Note that we studied different formulae carefully, and chose an exponential form as the most suitable one since it always maps the CNN value to the [0–100%] range. Furthermore, a base of 1.1 was selected rather than a larger base to ensure that the CCN measure does not decay to zero too quickly.

Note that the CCN measure as defined considers only the number of detected corners without regard to whether they are correct or not. It should be pointed out that the correctness of corner detections is captured by the *accuracy*

measure so there is no point for the consistency measure to again focus on that. As an extreme example, if a corner detector found exactly the same number of corners in the transformed image but all of them at incorrect positions, the corner detector in question would do well in the *consistency* measure but not well in the *accuracy* measure.

### 6.2. Accuracy

Accuracy requires that corners should be detected as close as possible to their correct positions. In a given image, the corner positions and numbers can be different according to different individuals. 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 motivate this approach, we should point out that we are specifically interested in performance evaluation of corner detectors, and not the more general feature point detectors or interest point detectors. A suitable definition for a corner is an image point where two-dimensional change can be observed in the image. Using this definition, we believe it is quite reasonable to expect that the performance of a corner detector should correspond to human judgment.

Note that ground truth has been used extensively in computer vision and image processing. However, we believe it has never been used before in the context of corner detection. To create the ground truth, 10 persons who were familiar with the task of corner detection were chosen. This number is significant as it indicates that we did not rely on just one or a small number of people to determine the ground truth. In principle, it is possible to use even more human judges, but even with 10 judges, the experimental effort required was quite substantial. None of the judges were familiar with the algorithm used by the ECSS corner detector. We asked them individually to mark the corners of an image. To select the locations of the corners, the human judges viewed the image on a computer screen using a program that allowed them to read the coordinates of individual pixels in the image, and used a cursor to determine the pixel coordinates of each corner. The viewing conditions were essentially the same for all human judges. There is no reason to believe that the cursor affected the perception of the corners. The corners marked by at least 70% of individuals were selected as the ground truth for that image. There was some variation amongst the human judges, but not substantial variation. In general, the human judges showed strong consistency in determining the locations of corners, however, occasionally a corner was marked by some judges and not the others, so the purpose of the 70% figure was to ensure that a convincing majority of judges agreed on a corner before it was admitted as part of the ground truth. To determine whether points marked by two judges were the same corner, a neighborhood test was utilized to ensure that the points were quite close to each other if not identical. All judges looked at the image at the same scale, therefore there were no problems caused by scale changes. 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 10 individuals. We repeated the same procedure for other images. By comparing the corners detected using each of five corner detectors to the list of corners in the ground truth, the accuracy was computed as follows.

For a given image, let $N_o$ be the total number of corners detected by a specific corner detector (note that $N_o \neq 0$), let $N_g$ be the total number of corners in the ground truth, and let $N_a$ be the total number of matched corners when comparing the first set of corners (found using a corner detector) to the second set (ground truth) using a neighborhood test. The criterion of accuracy is defined as

$$\mathrm{ACU} = 100 \times \frac{\frac{N_a}{N_o} + \frac{N_a}{N_g}}{2}, \tag{7}$$

where ACU stands for *accuracy*. The value of ACU for accurate corner detectors should be close to 100%. However, the value of ACU for inaccurate corner detectors is close to zero. Note that if a corner detector finds more false corners which implies more matched corners, it does not follow that the ACU of this detector is high since in this case, if $N_a/N_g$ is near one, $N_a/N_o$ drops closer to zero. On the other hand, if a corner detector finds less corners which means less matched corners, $N_a/N_o$ goes to one and $N_a/N_g$ drops closer to zero. Therefore in both cases, the ACU of such detectors computed through Eq. (7) is less than 100%. Note that the case of $N_o = 0$ in Eq. (7) occurs if a test image has no corners or the test corner detectors can not detect any corners. These situations do not arise in practice as only images with many corners are used in experiments and corner detectors under consideration also find many corners in test images.

Some may argue that the more conventional terms of *false positives* and *false negatives* should be used to analyze accuracy. In fact, formula 7 captures both concepts at the same time. The concept of *false positives* is captured by the term $N_a/N_o$. When a corner detector is finding too many corners, $N_o$ will grow large and therefore $N_a/N_o$ will move towards zero. Furthermore, the concept of *false negatives* is captured by the term $N_a/N_g$. When a corner detector is not finding enough corners, $N_a$ will be much smaller than $N_g$, and therefore $N_a/N_g$ will move towards zero. Note that the relative cost of false positives and false negatives varies from application to application, but since our work was intended to be independent of specific applications, we used formula 7 to evaluate the accuracy of corner detectors. This formula in fact attaches equal cost to false positives and false negatives, and combines them in a single measure.

## 7. Results and discussion

We carried out many experiments using several images from our image collection which included a leaf image,

an airplane image, a fish image, a lab image and a building image. All the test images have been illustrated in Fig. 7. Note that while it would be ideal to capture images of transformed objects for corner detector testing, in practice it can be a very difficult and complicated process as it requires a special setup (for example, consider accurate rotation of the camera about its axis). Digital image transformations are a good approximation to that process which allow us to evaluate corner detectors more systematically and more comprehensively. Furthermore, all corner detectors are evaluated under the same testing conditions. A total of five experiments were carried out. The first four experiments are relevant to the computation of the *consistency* measure. The fifth experiment is relevant to the computation of the *accuracy* measure. The experiments were performed as following.

### 7.1. Experiment 1

The number of corners in original image were first extracted using the test corner detectors. The original image was then rotated with rotation angle chosen by uniform sampling of the interval $[-90°, +90°]$ excluding zero. Distance between consecutive samples was 10°. The number of corners in all rotated images were then extracted using the test detectors.

### 7.2. Experiment 2

A similar procedure was carried out for the original image and uniform scaling of this image with 10 scale factors chosen by uniform sampling of the interval [0.5, 1.5] excluding 1.0. Distance between consecutive samples was 0.1.

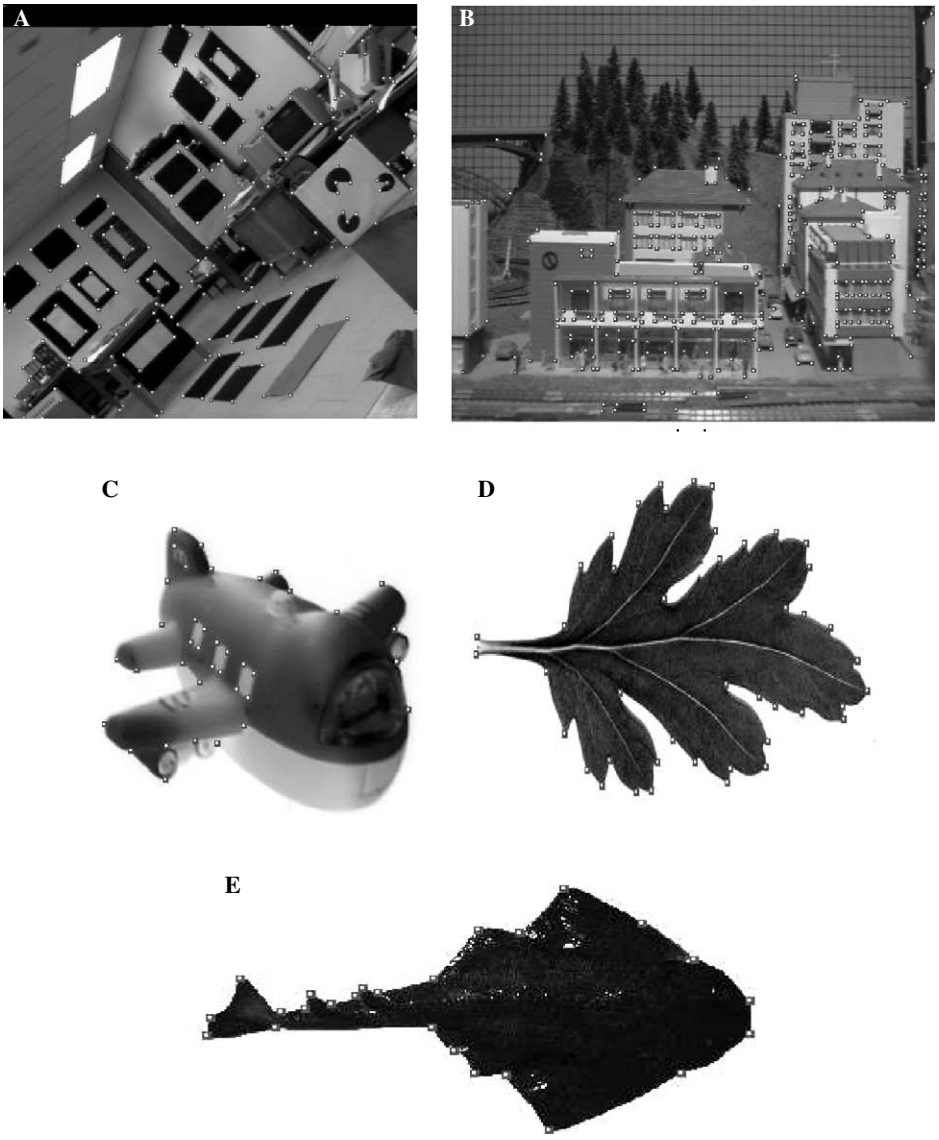


Fig. 7. Test images for computation of accuracy. The corner points in the ground truths have also been shown. (A) Lab. (B) Building. (C) Airplane. (D) Leaf. (E) Fish.

### 7.3. Experiment 3

A similar procedure was carried out for the original image and non-uniform scaling of that image using scale factors $x$-scale and $y$-scale. The $x$-scale and $y$-scale parameters were chosen by uniform sampling of the intervals [0.8, 1.0] and [0.5, 1.5], respectively. Distance between consecutive samples was 0.1.

### 7.4. Experiment 4

A similar procedure was carried out for the original image and affine transforms of that image. The affine transforms consisted of a rotation angle of $-10°$ and $+10°$ combined with non-uniform scaling with the $x$-scale and $y$-scale parameters chosen by uniform sampling of the intervals [0.8, 1.0] and [0.5, 1.5], respectively. Distance between consecutive samples was 0.1.

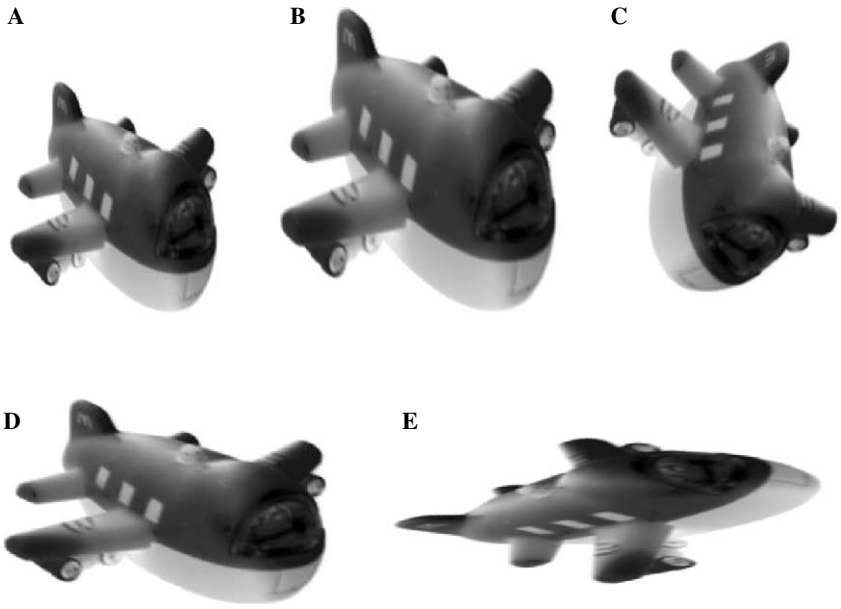


Fig. 8. Airplane image under similarity and affine transforms. In this figure $S$, $S_x$, $S_y$, and $\theta$ stand for the uniform scale factor, scale factors in $x$ and $y$ directions in non-uniform scaling, and the rotation parameter, respectively. (A) Original. (B) $S = 2$. (C) $\theta = -60°$. (D) $S_x = 1.5$, $S_y = 0.8$. (E) Affine transform: $\theta = +40°$, $S_x = 1.6$, $S_y = 0.6$.

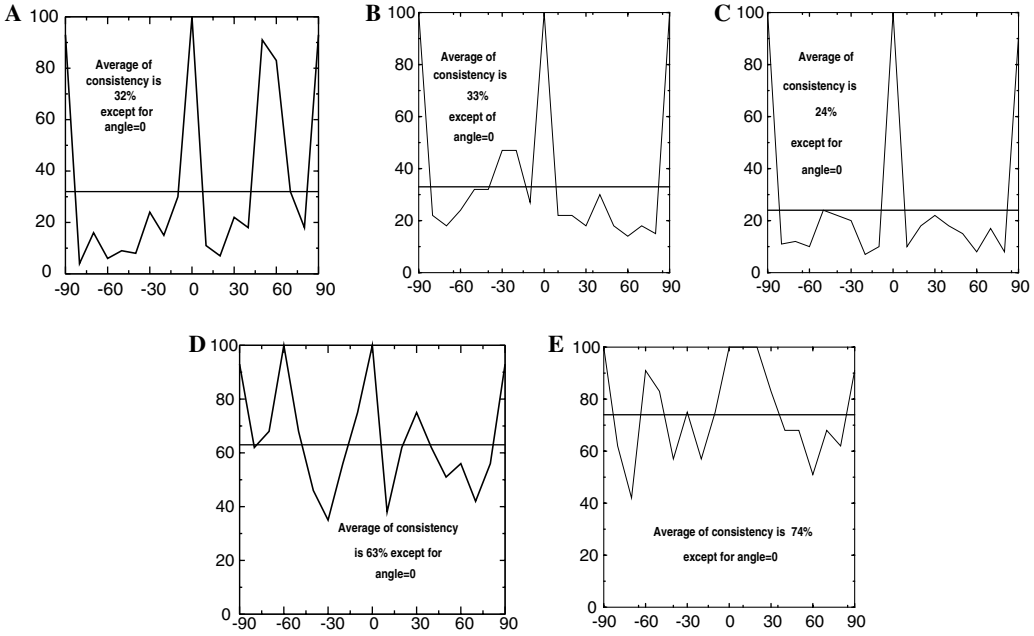


Fig. 9. Consistency of corner numbers for rotation. (A) Plessy. (B) K and R. (C) Susan. (D) CSS. (E) ECSS.

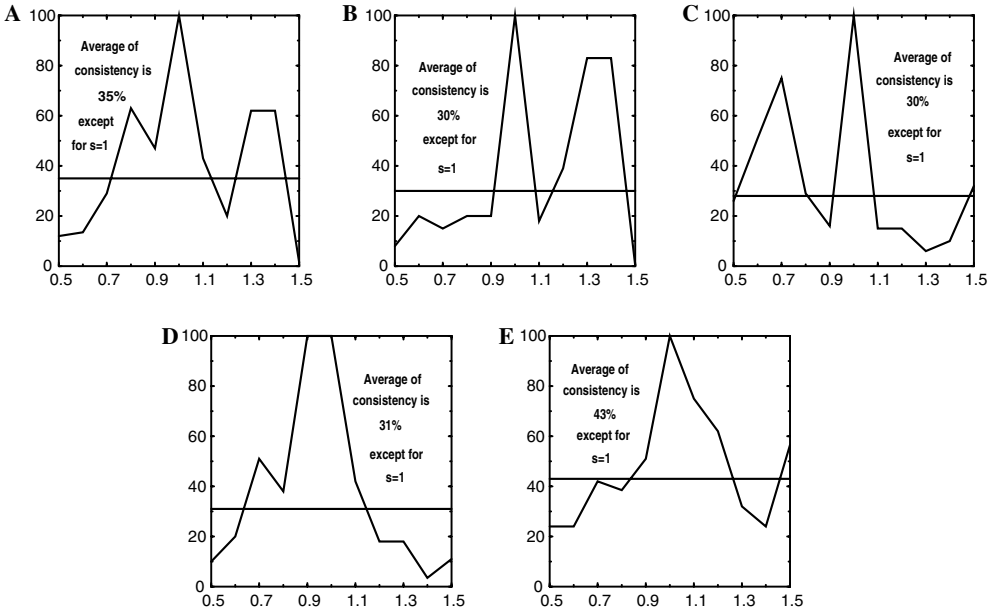


Fig. 10. Consistency of corner numbers for uniform scaling. (A) Plessy. (B) K and R. (C) Susan. (D) CSS. (E) ECSS.

Note that in Experiments 1–4, affine warps of the original test images are computed instead of using "real world examples." Note that use of real world examples would have resulted in a very complex hardware setup to carry out the experiments. For example, to achieve rotations of the image, we would have to put together a setup that would rotate the camera by a specific number of degrees each time so that a rotated image of the scene could be obtained. The hardware setup required to obtain an affine warped image of the same scene would be even more complex. We did not have access to the required hardware. As a result, application of mathematical transforms to the original images was considered an acceptable alternative. Furthermore, the testing conditions are exactly the same for all tested corner detectors, and therefore the conclusions are still valid.

### 7.5. Experiment 5

Ground truth corners were extracted from the original images using the procedure described in Section 6.2.

Examples of image transforms have been illustrated in Fig. 8. After performing the experiments on rotated, uniformly and non-uniformly scaled and affine transformed images, values for the CCN measure were computed. The results of these computations for rotation and uniform scaling have been illustrated in Figs. 9 and 10. The average values of the consistency measure for non-uniform and affine transforms have been shown in Table 1.

We also computed the mean and standard deviation of the accuracy measure for all the tested corner detectors as well as the operators. The results have been shown in Table 2. As this table shows, we found that the mean of accuracy for the operators was quite high and the standard deviation was quite small. This table also shows that the human operators performed much better than the best corner detector (ECSS) indicating that further research should be carried out by the research community on even better corner detectors.

It should be pointed out that all test images are real and contain noise. As a result, the transformed images also contain noise, and are by no means clean. However, no artificial noise was added to the test images. Furthermore, in Fig. 7, the corner points in the image ground truths also have been shown. Note that the size of each test image is always the same for all tested detectors, and while the size of different test images can vary, the conditions are the

Table 1
Mean and standard deviation of consistency for test corner detectors

| | Plessey | K and R | Susan | CSS | ECSS |
|---|---|---|---|---|---|
| *Consistency: mean and standard deviation* | | | | | |
| Rotation | 32% | 33% | 24% | 62% | 74% |
| Uniform scaling | 35% | 30% | 28% | 31% | 42% |
| Non-uniform scaling | 28% | 31% | 31% | 55% | 68% |
| Affine transform | 14% | 11% | 9% | 42% | 51% |
| Mean | 27.2 | 26.2 | 23.0 | 47.5 | 58.7 |
| Standard deviation | 8.0 | 8.9 | 8.4 | 11.9 | 12.8 |

Table 2
Mean and standard deviation of accuracy for test corner detectors

| | Plessey | K and R | Susan | CSS | ECSS | Operators |
|---|---|---|---|---|---|---|
| *Accuracy: mean and standard deviation* | | | | | | |
| Mean | 49.6 | 56.0 | 53.4 | 71.6 | 77.2 | 94.7 |
| Standard deviation | 10.4 | 9.0 | 10.9 | 10.9 | 10.7 | 3.2 |

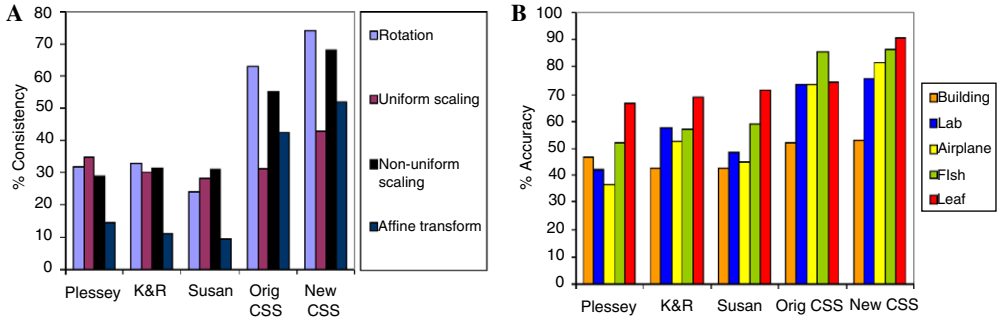


Fig. 11. Overall consistency and accuracy values for test corner detectors. (A) Consistency. (B) Accuracy.

same for each detector tested, and therefore the CCN values can be averaged across different images.

The overall consistency and accuracy values for test detectors have been illustrated in Fig. 11. These figures show that the ECSS detector has better accuracy and consistency than the other four corner detectors.

Regarding the computational requirements of the corner detectors tested, it should be pointed out that the Kitchen and Rosenfeld detector was the fastest of those detectors, but the rest of the detectors had quite similar speeds.

Note that none of the test corner detectors have been designed to be invariant to affine transformations. While it may be possible in principle to design invariant corner detectors ([35] and [65] are examples of invariant feature point detector and invariant descriptor detector, respectively), in practice such detectors depend on high-order derivatives and are therefore highly sensitive to noise in the image which significantly limits their practical value. In fact, the ECSS corner detector could be extended so that affine curvature rather than regular curvature is computed on edge contours. Maxima of affine curvature could then be put forward as invariant image corners. This idea was considered at an early stage of our work, but experiments showed that the resulting corners were not robust.

## 8. Concluding remarks

This paper evaluated the performance of the ECSS corner detector and compared it to the performances of a few popular corner detectors. This performance evaluation was carried out using two new criteria of consistency and accuracy. We also proposed a new approach for creation of ground truth used for computation of accuracy. The application of these criteria showed that the ECSS corner detector performed better in terms of consistency and accuracy.

## References

[1] S. Ando, Consistent gradient operators, IEEE Trans. Pattern Anal. Mach. Intell. 22 (3) (2000) 252–265.
[2] S. Ando, Image field categorization and edge/corner detection from gradient covariance, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2) (2000) 179–190.
[3] F. Arrebola, A. Bandera, P. Camacho, F. Sandoval, Corner detection by local histograms of contour chain code, Electron. Lett. 33 (21) (1997) 1769–1771.
[4] F. Arrebola, P. Camacho, A. Bandera, F. Sandoval, Corner detection and curve representation by circular histograms of contour chain code, Electron. Lett. 35 (13) (1999) 1065–1067.
[5] H. Asada, M. Brady, The curvature primal sketch, IEEE Trans. Pattern Anal. Mach. Intell. 8 (1) (1986) 2–14.
[6] S. Bae, I.S. Kweon, C.D. Yoo, Cop: a new corner detector, Pattern Recognit. Lett. 23 (ll) (2002) 1349–1360.
[7] S. Baker, S.K. Nayar, Global measures of coherence for edge detector evaluation, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Fort Collins, Colorado, 1999, pp. 373–379.
[8] I.N. Bankman, E.W. Rogala, Corner detection for identification of man-made objects in noisy aerial images, in: Proc. SPIE, vol. 4726, 2002, pp. 304–309.
[9] A. Baumberg, Reliable feature matching across widely separated views, in: Proc IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, 2000, pp. 774–781.
[10] P.R. Beaudet, Rotationally invariant image operators, in: Proc Int. Joint Conf. Pattern recognition, 1978, pp. 579–583.
[11] K.W. Bowyer, C. Kranenburg, S. Dougherty, Edge detector evaluation using empirical roc curves, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Fort Collins, Colorado, 1999, pp. 354–359.
[12] J.F. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.
[13] F. Chabat, G. Yang, D. Hansell, A corner orientation detector, Image Vis. Comput. 17 (1999) 761–769.
[14] W.C. Chen, P. Rockett, Bayesian labelling of corners using a grey-level corner image model, in: Proc. IEEE Internat. Conf. on Image Processing, vol I, 1997, pp. 687–690.
[15] J. Cooper, S. Venkatesh, L. Kitchen, Early jump-out corner detectors, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1991, pp. 688–689.
[16] E.R. Davies, Application of the generalized hough transform to corner detection, IEEE Proc. 135 (1988) 49–54.
[17] R. Deriche, G. Giraudon, A computational approach for corner and vertex detection, Int. J. Comput. Vis. 10 (2) (1993) 101–124.
[18] L. Dreschler, H.H. Nagel. Volumetric model and 3D trajectory of a moving car derived from monocular tv frame sequences of a street scene, in: IJCAI-81, 1981, pp. 692–697.
[19] R. Elias, R. Laganiere, Cones: a new approach towards corner detection, in: Proc Canadian Conf. on Electrical and Computer Engineering, vol. 2, 2002, pp. 912–916.
[20] Y. Etou, T. Sugiyama, K. Abe, T. Abe, Corner detection using slit rotational edge-feature detector, in: Proc ICIP, vol. 2, 2002, pp. 797–800.
[21] M. Fidrich, Stability of corner points in scale space: the effects of small non-rigid deformations, Comput. Vision Image Understand. 72 (1) (1998) 72–83.
[22] G. Gallo, A. Giuoco, L. Alessandro, Multi-scale corner detection and classification using local properties and semantic pattern, in: Proc. SPIE, vol. 4667, 2002, pp. 108–119.

[23] C.G. Harris, Determination of ego-motion from matched points, in: Proc. Alvey Vision Conf., Cambridge, UK, 1987.
[24] M.D. Heath, S. Sarkar, T. Sanocki, K.W. Bowyer, A robust visual method for assessing the relative performance of edge-detection algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 19 (12) (1997) 1338–1359.
[25] A. Heyden, K. Rohr, Evaluation of corner extraction schemes using invariance methods, in: Proc. Internat. Conf. on Pattern Recognition, vol. 1, Vienna, Austria, 1996, pp. 895–899.
[26] Q. Ji, R.M. Haralick, Corner detection with covariance propagation, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1997, pp. 362–367.
[27] R. Kakarala, A. Hero, On achievable accuracy in edge localization, IEEE Trans. Pattern Anal. Mach. Intell. 14 (7) (1992) 777–781.
[28] L. Kitchen, A. Rosenfeld, Gray level corner detection, Pattern Recognit. Lett. (1982) 95–102.
[29] K. Kohlmann, Corner detection in natural images based on the 2D Hilbert transform, Signal Process. 48 (3) (1996) 225–234.
[30] K.K. Lai, P.S.Y. Wu, Effective edge-corner detection method for defected images, in: Proc. Internat. Conf. on Signal Processing, vol. II, 1996, pp. 1151–1154.
[31] K.J. Lee, Z. Bien, Grey-level corner detector using fuzzy logic, Pattern Recognit. Lett. 17 (9) (1996) 939–950.
[32] T. Lindeberg, Junction detection with automatic selection of detection scales and localization scales, in: Proc. Internat. Conf. on Image Processing, vol. 1, 1994, pp. 924–928.
[33] T. Lindeberg, Feature detection with automatic scale selection, Int. J. Comput. Vis. 30 (2) (1998) 79–116.
[34] A.M. Lopez, F. Lumbreras, J. Serrat, J.J. Villanueva, Evaluation of methods for ridge and valley detection, IEEE Trans. Pattern Anal. Mach. Intell. 21 (4) (1999) 327–335.
[35] K. Mikolajczyk, C. Schmid, An affine invariant interest point detector, in: Proc Eur. Conf. on Computer Vision, 2002, pp. 128–142.
[36] F. Mokhtarian, N. Khalili, P. Yuen, Multi-scale free-form 3D object recognition using 3D models, Image Vis. Comput. 19 (2001) 271–281.
[37] F. Mokhtarian, A.K. Mackworth, A theory of multi-scale, curvature-based shape representation for planar curves, IEEE Trans. Pattern Anal. Mach. Intell. 14 (8) (1992) 789–805.
[38] F. Mokhtarian, R. Suomela, Robust image corner detection through curvature scale space, IEEE Trans. Pattern Anal. Mach. Intell. 20 (12) (1998) 1376–1381.
[39] H.P. Moravec, Towards automatic visual obstacle avoidance, in: Proc. Internat. Joint Conf. Artificial Intelligence, 1977, p. 584.
[40] S. Nassif, D. Capson, A. Vaz. Robust real-time corner location measurement, in: Proc IEEE Conf. on Instrumentation and Measurement Technology, 1997, pp. 106–111.
[41] A. Noble, Finding corners, Image Vis. Comput. 6 (1988) 121–128.
[42] K. Paler, J. Foglein, J. Illingworth, J. Kittler, Local ordered grey levels as an aid to corner detection, Pattern Recognit. 17 (5) (1984) 535–543.
[43] X. Peng, C. Zhou, M. Ding. Corner detection method based on wavelet transform. In: Proc SPIE, vol. 4550, 2001, pp. 319–323.
[44] A. Pikaz, I. Dinstein, Using simple decomposition for smoothing and feature point detection of noisy digital curves, IEEE Trans. Pattern Anal. Mach. Intell. 16 (8) (1994) 808–813.
[45] A. Quddus, M. Fahmy, Fast wavelet-based corner detection technique, Electron. Lett. 35 (4) (1999) 287–288.
[46] A. Quddus, M. Gabbouj, Wavelet-based corner detection technique using optimal scale, Pattern Recognit. Lett. 23 (2002) 215–220.
[47] K. Rangarajan, M. Shah, D.V. Brackle, Optimal corner detector, Comput. Vision. Graph. Image Process. 48 (1989) 230–245.
[48] K. Rohr, Recognizing corners by fitting parametric models, Int. J. Comput. Vis. 9 (3) (1992) 213–230.
[49] K. Rohr, Localization properties of direct corner detectors, J. Math. Imaging Vis. 4 (2) (1994) 139–150.
[50] K. Rohr, On the precision in estimating the location of edges and corners, J. Math. Imaging Vis. 7 (1) (1997) 7–22.
[51] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of interest point detectors, Int. J. Comput. Vis. 37 (2) (2000) 151–172.
[52] E. Shilat, M. Werman, Y. Gdalyahu, Ridge's corner detection and correspondence, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 976–981, 1997.
[53] M.C. Shin, D. Goldgof, K.W. Bowyer, Comparison of edge detectors using an object recognition task, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Fort Collins, Colorado, 1999, pp. 360–365.
[54] A. Singh, M. Shneier, Grey level corner detection: A generalization and a robust real time implementation, Comput. Vision Graph. Image Process. 51 (1990) 54–69.
[55] S.M. Smith, J.M. Brady, A new approach to low-level image processing, Int. J. Comput. Vis. 23 (1) (1997) 45–78.
[56] K. Sohn, J.H. Kim, W.E. Alexander, Mean field annealing approach to robust corner detection, IEEE Trans. Syst. Man Cybernetics. Part B: Cybernetics 28 (1) (1998) 82–90.
[57] J. Sporring, O. Olsen, M. Nielsen, J. Weickert, Smoothing images creates corners, Image Vis. Comput. 18 (2000) 261–266.
[58] M. Trajkovic, M. Hedley, Fast corner detection, Image Vis. Comput. 16 (2) (1998) 75–87.
[59] D.M. Tsai, Boundary based corner detection using neural networks, Pattern Reognit. 30 (1) (1997) 85–97.
[60] Z.O. Wu, A. Rosenfeld, Filtered projections as an aid to corner detection, Pattern Recognit. 16 (31) (1983).
[61] X. Xie, R. Sudhakar, H. Zhuang, Corner detection by a cost minimization approach, Pattern Recognit. 26 (8) (1993) 1235–1243.
[62] G.Z. Yang, P. Burger, D.N. Firmin, S.R. Underwood, Structure adaptive anisotropic image filtering, Image Vis. Comput. 14 (1996) 135–145.
[63] X. Zhang, D. Zhao, Parallel algorithm for detecting dominant points on multiple digital curves, Pattern Recognit. 30 (2) (1997) 239–244.
[64] Z. Zheng, H. Wang, E.K. Teoh, Analysis of gray level corner detection, Pattern Recognit. Lett. 20 (1999) 149–162.
[65] A. Zisserman, F. Schaffalitzky. Viewpoint invariant descriptors for image matching, in: Proc Workshop on MultiMedia Content Based Indexing and Retrieval, France, 2001.
[66] O.A. Zuniga, R.M. Haralick. Corner detection using the facet model, in: Proc. Conf. on Pattern Recognition and Image Processing, 1983, pp. 30–37.