Assessing the Performance of Corner Detectors for Point Feature Tracking Applications

[**]P. Tissainayagam
raj@tns.nec.com.au
Transmissions Systems Division
NEC (Australia) Pty. Ltd.
649-655 Springvale Road, Mulgrave, Victoria 3170, Australia

D. Suter
d.suter@eng.monash.edu.au
Dept. of Electrical and Computer Systems Engineering
Monash University
Clayton, Victoria 3800, Australia

Abstract

In this paper we assess the performance of corner feature detecting algorithms for feature tracking applications. We analyze four different types of corner extractors, which have been widely used for a variety of applications. They are the Kitchen-Rosenfeld, the Harris, the Kanade-Lucas-Tomasi, and the Smith corner detectors. We use corner stability and corner localization properties as measures to evaluate the quality of the features extracted by the 4 detectors. For effective assessment of the corner detectors, we employed image sequences with no motion (simply static image sequences), so that the appearance and disappearance of corners in each frame is purely due to image plane noise and illumination conditions. Such a setup is ideal to analyze the stability and localization properties of the corners. The corners extracted from the initial frame are then matched through the sequence using a corner matching strategy. We employed 2 different types of matchers, namely the GVM (Gradient Vector Matcher) and the Product Moment Coefficient Matcher (PMCM). Each of the corner detectors was tested with each of the matching algorithms to evaluate their performance in tracking (matching) the features. The experiments were carried out on a variety of image sequences.

Key words: Point features, Corner detector, Point feature tracking, Feature extraction, Corner matching.

1 Introduction

Low-level descriptors may be broadly classified into three main types: region-based, edge-based and point-based [21]. *Regions* (or “blobs”) [27, 21, 22] normally correspond to smooth surface patches. Tracking such regions is not always easy, since minor differences between frames (due to image noise or image motion) can lead to very different segmentation in consecutive image frames [21]. Despite recent progress (for example: Meyer and Bouthemy [16] tracked convex-hull approximations to region boundaries, Etoh and Shirai [10] used advanced statistical region descriptors), further theoretical and empirical work is needed before reliable region tracking becomes feasible.

Edges are loci of one-dimensional spatial change [21], located where the change in intensity is significant in one direction. They are generally detected by finding either maxima in the first image derivative [5], or zero-crossings in the Laplacian of the Gaussian of the image [11]. Their usefulness in motion algorithms is limited by

[**] Corresponding author.

the “*aperture problem*”, which arises from a locally linear expansion of the spatio-temporal image intensity function; without assumptions about the nature of the flow, only the component of flow *perpendicular* to the edge element can be found [8]. Unfortunately, these assumptions invariably lead to inaccuracies in the estimated flow, particularly at motion boundaries [21]. The use of higher order derivatives is unsatisfactory since differentiation accentuates noise. Moreover, until the advent of snakes [14, 2, 4], arbitrarily curving edges were difficult to describe and track, and simultaneous tracking of multiple open edge contours with automatic snake initialization still largely remains an open problem [21, 2]. An additional problem with tracking edges through image sequences is that edge segments tend to *split* and *merge,* which complicates the tracking process considerably.

Point features are distinctive image points corresponding to objective 3D scene elements that are in most instances accurately locatable and recur in successive images, which makes them explicitly trackable over time. The term “corners” is used to refer to point features that are loci of two-dimensional intensity change, i.e. *second-order features*. This includes points of occlusion (e.g. T, Y and X junctions), structural discontinuities (e.g. L junctions) and various curvature maxima (e.g. texture flecks or surface markings). Corners impose more constraint on the motion parameters than edges, therefore the *full* optic flow field is recoverable at corner locations [21]. Corners are also often more abundant than straight edges in the natural world making them ideal features to track in an indoor and outdoor environment. To find further details on various corner detectors, the reader is referred to [17, 21].

Despite the large amount of material reported in the literature in the area of low level feature tracking, very little has been published in terms of a performance analysis for many of these algorithms. Our primary contribution in this paper is to evaluate the suitability of corners extracted by 4 different corner detectors for point tracking purposes. For point (corner) feature tracking it is essential that corners extracted in each frame be well localized and temporally stable throughout the image sequence. To test these corner properties it is preferred to use static image sequences where object (or camera) motion will not be a concern. Using indoor and outdoor static image sequences with varied levels of illumination, we assess the quality of corners extracted by the *Kanade-Lucas-Tomasi (KLT)* [26], the *Harris* [13], the *Kitchen-Rosenfeld* [15] and the *Smith* [25] corner detectors in the presence of varied noise.

A direct comparison of the performance of corner detectors is difficult because establishing ground truth for ‘corner points’ is non-trivial, particularly for complex scenes (as studied in this paper). Even for human eyes declaring the best *N* corners from a complex scene is very difficult and the choice of selection can vary from person to person. Therefore, for each corner detector considered in this paper, we provide the allowance of selecting ‘their best *N* corners’. The quality of corners extracted by each detector is then assessed against the localization and stability properties (discussed later). The assessment based on these 2 measures will reveal the number of corners that are most resilient to image plane noise.

This paper is organized as follows. Section 2 briefly describes the four corner detectors that we analyze. Section 3 provides the performance measures that we employ for our analysis. Section 4 gives the inter-frame corner matching strategies employed. Section 5 derives the tests that are applied for empirical evaluation of the corner detectors. Section 6 gives the results, and section 7 provides a discussion on the evaluation outcome. Finally section 8 gives the conclusion.

2 Corner (Point) Features as Tracking Tokens

A number of algorithms for corner detection have been reported in recent years [7, 9, 10, 13, 15, 17, 19, 20, 21, 22, 23, 25, 26]. They can be divided into two groups. Algorithms in the first group involve extracting edges and then finding the points having maxima curvature or searching for points where edge segments intersect. The second, and largest group, consists of algorithms that search for corners directly from the grey-level image. In this paper we focus on the second group of feature detectors.

We decided to assess the performance of the *Kitchen-Rosenfeld* [15], the *Harris* [12, 13], the *KLT* [26] and the *Smith* [25] corner detectors when applied on real data. The choice of these 4 corner detectors was made because, the *Kitchen-Rosenfeld* method uses second and first order derivatives in calculating the cornerness value, while *Harris* method uses only first order derivatives. The *Smith* method uses a geometrical criteria in calculating the cornerness value (no derivations are required), while *KLT* detector uses information from an inter-frame point displacement technique to declare corners. Therefore an assessment in terms of the localization and stability properties for these 4 corner detection methods seemed useful when they are applied to a variety of image sequences. The following sub-sections briefly describe the corner detectors employed.

2.1 The Kitchen-Rosenfeld Corner Detector

Kitchen-Rosenfeld algorithm [15] is one of the earliest corner detectors reported in the literature, hence it has been used as a bench mark for future researchers developing corner detection algorithms. This algorithm calculates the ‘cornerness’ value C as the produce of the local gradient magnitude and the rate of change of gradient direction. The quantity C is given by,

$$C = \frac{\mathbf{I}_{xx}\mathbf{I}_y^2 + \mathbf{I}_{yy}\mathbf{I}_x^2 - 2\mathbf{I}_{xy}\mathbf{I}_x\mathbf{I}_y}{\mathbf{I}_x^2 + \mathbf{I}_y^2} \qquad (1)$$

where $\mathbf{I}$ is the grey-level value, and $\mathbf{I}_{xx}$ is the second derivative of **I**, etc. Points in an image are declared corners if the ‘cornerness value’ meets some threshold requirement (ie. the lower the value of C, better the corner). Many well-known corner extractors use this threshold (eg:[9, 19]). The primary reason for considering this detector is to use it as a ' bench mark' to evaluate the performances of the *Harris*, the *KLT*, and the *Smith* corner detectors.

2.2 The Harris Corner Detector

This algorithm, known as the *Harris* corner detector [13] is based on an underlying assumption that corners are associated with maxima of the local autocorrelation function. It is less sensitive to noise in the image than most other algorithms, because the computations are based entirely on first derivatives. The algorithm has proved popular due to its high reliability in finding L junctions and its good temporal stability [20], making it an attractive corner detector for tracking. It should be noted that because these algorithms rely on spatial derivatives, image smoothing is often required to improve their performance. While improving the detection reliability, it has been shown that smoothing may result in poor localization accuracy [18]. The *Harris* corner detector was used successfully to detect features for the DROID 3D vision project [12, 13].

The *Harris* corner detector also computes a *cornerness* value, *C*, for each pixel in an image. A pixel is declared a *corner* if the value of *C* is below a certain threshold. Where *C* is calculated as follows:

- Calculate the intensity *x*-gradient, $\mathbf{I}_x$, and the intensity *y*-gradients, $\mathbf{I}_y$ using 3x3 convolution masks.
- Calculate $\mathbf{I}_x^2, \mathbf{I}_y^2, \mathbf{I}_x\mathbf{I}_y$.
- Using a Gaussian smoothing kernel of standard deviation σ, calculate the sampled means $<\mathbf{I}_x^2>, <\mathbf{I}_y^2>, \text{and} <\mathbf{I}_x\mathbf{I}_y>$. See Fig. 1.
- Calculate the cornerness value of a pixel, *C* as follows:

$$C = \frac{<\mathbf{I}_x^2> + <\mathbf{I}_y^2>}{<\mathbf{I}_x^2><\mathbf{I}_y^2> - <\mathbf{I}_x\mathbf{I}_y>^2} \tag{2}$$

A good corner is defined as having a small value of *C*; the *best* corner thus having the lowest value of *C*. The number of surrounding pixels required to calculate *C* is determined by the size of the Gaussian smoothing kernel. A 3x3 pixel smoothing kernel gives a 5x5 pixel computation area, a 5x5 pixel smoothing kernel gives a 7x7 pixel computation area, *etc.* (Figure 1).

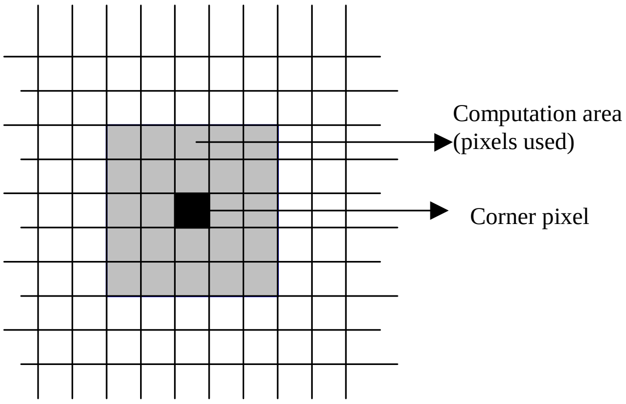


Figure 1: Pixel area used when calculating the Harris cornerness C, assuming a Gaussian smoothing kernel of 3x3 pixels.

2.3 The Smith (SUSAN) Corner Detector

Smith [25] developed a very simple corner detector that uses no spatial derivatives at all. The *Smith* corner detector does not require any smoothing and so there is no degradation in localization accuracy due to smoothing. This detector has been implemented as part of a scene segmentation algorithm ASSET (A Scene Segmenter Establishing Tracking) [24].

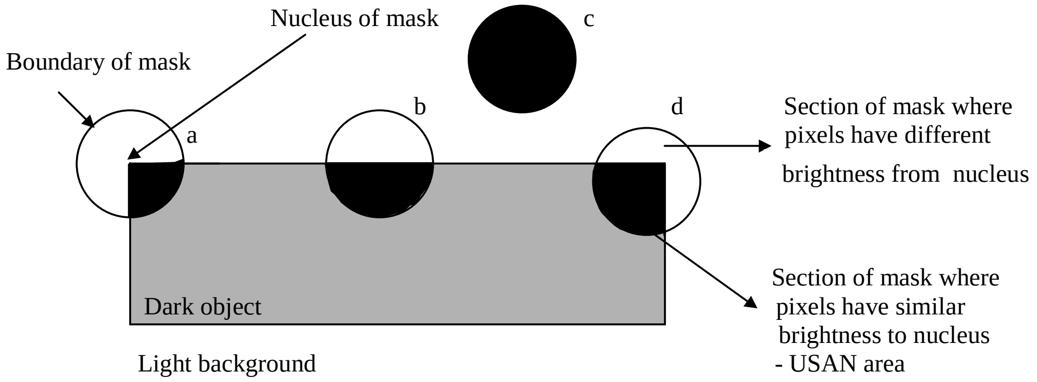

Nucleus of mask
c
Boundary of mask
b
d
a
Section of mask where pixels have different brightness from nucleus
Section of mask where pixels have similar brightness to nucleus - USAN area
Dark object
Light background

Figure 2: Four Smith corner finding masks at different positions in an image.

The *Smith* corner detector [25] is different from the other detectors in nature. Each pixel in an image is used as the center of a small circular mask. The greyscale values of all the pixels within this circular mask are compared with that of the center pixel (the *nucleus*). All pixels with *similar* brightness to that of the nucleus are assumed to be part of the same structure in the image.

Figure 2 shows the masks with pixels of *similar* brightness to the nucleus coloured black, and pixels with *different* brightness coloured white. Smith calls the black image area the *Univalue Segment Assimilating Nucleus* (USAN). He argues that the USAN corresponding to a corner (case (a) in Figure 3) has an USAN area of less than half the total mask area. It is clear from Figure 3 that a local minimum in USAN area will find the exact point of the corner.

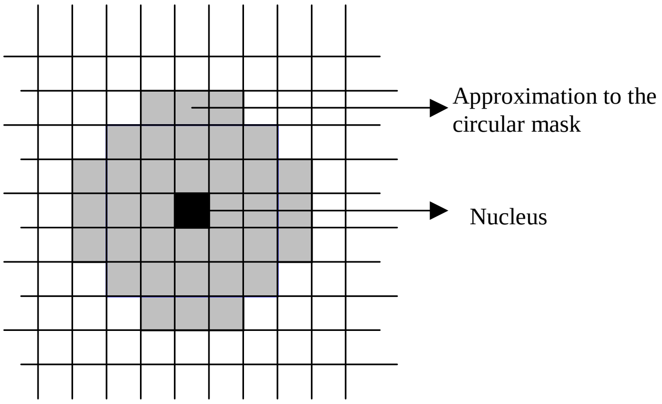


Figure 3: The Smith USAN (SUSAN) corner finding mask.

In practice, the circular mask is approximated using a 5 x 5 pixel square with 3 pixels added on to the center of each edge (Figure 3). The intensity of the nucleus is then compared with the intensity of every other pixel within the mask using the following comparison function:

$$c(r, r_0) = 100e^{-\left(\frac{\mathbf{I}(r)-\mathbf{I}(r_0)}{t}\right)^6} \quad (3)$$

where r_0 is the position of the nucleus, r is the position of any other point within the mask, **I***(r)* is the brightness of any pixel, and t is the so-called *brightness difference threshold*. Eq. (3) is chosen to allow a pixel's brightness to vary slightly without having too large an effect on *c,* even if it is near the threshold position. The sixth power is used to obtain the theoretical optimum, see [25] for details. This comparison is done for each pixel in the circular mask, and a running total, *n,* of the outputs, *c,* is made:

$$n = \sum_r c(r, r_0) \quad (4)$$

The total *n* is 100 times the USAN's area (the factor of 100 coming from Equation 3). The USAN area *n* is then thresholded to extract the corners. A pixel is declared a corner if its USAN area, *n,* is less than half the maximum possible USAN area (The maximum USAN area is given by the area of the circular mask times 100, which is 3700). The *geometric threshold, g,* therefore sits at 1850 [(25+12)*100/2]. Smith points out that the value of *g* affects the shape of the corners detected, and that reducing the value of *g* results in only the sharpest corners being detected [25]. The brightness difference threshold *t,* affects the quantity of corners detected by determining the allowed variation in brightness within the USAN.

Finally, an intermediate image is created from the value *n* calculated for each pixel in the image. If *n* is greater than the geometric threshold, *g*, then a zero is placed in the intermediate image, otherwise the value *(g-n(x,y))* is used. The intermediate image is then searched over a square 5 pixel by 5 pixel region for local maxima, and it is these local maxima pixels that are declared corners.

2.4 The Kanade-Lucas-Tomasi (KLT) Corner Detector

The KLT corner detector [26] operates by comparing a patch of image information in 2 consecutive frames of an image sequence (developed for the KLT tracking algorithm [26, 23]). It assumes that images taken at near time instants are usually strongly related to each other, because they refer to the same scene taken from only slightly different view points. This property can be explained by the following equation:

$$\mathbf{I}(x, y, t+\tau) = \mathbf{I}(x-\Delta x, y-\Delta y, t)$$

where **I** is the image intensity function having 3 parameters (space and time variables *x, y* & *t*). The inter-frame displacement $\mathbf{d} = (\Delta x, \Delta y)$ is the displacement of point **x** = (*x, y*) between time instants *t* and *(t+τ).* For the rest of this section, the notation *x, y, t* are dropped for convenience.

An important problem in finding the displacement **d** of a point from one frame to the next is that a single pixel cannot be reliably tracked, unless it has a very distinct character with respect to all of its neighbors. This is because of image plane noise, clutter etc. Because of these problems, KLT does not track a single pixel, but windows of pixels, and windows are looked for that contain sufficient texture. Using small window size is considered important because only small amount of change would have been accounted for within a small area. Any discrepancy between successive windows that cannot be explained by a translation is considered to be an error, and the displacement vector is chosen to minimize this residue error. This is expressed for a given window size *W* as:

$$\varepsilon = \int_W [\mathbf{I}(\mathbf{x}-\mathbf{d}) - \mathbf{J}(\mathbf{x})]^2 w d\mathbf{x}$$

where $\mathbf{J(x)} = \mathbf{I(x-d)} + n\mathbf{(x)}$, with noise *n*. Assuming the inter-frame displacement vector is small, using Taylor series (truncated to linear term) expansion, the following is possible: $\mathbf{I}(\mathbf{x}-\mathbf{d}) = \mathbf{I}(\mathbf{x}) - \mathbf{g.d}$. Now the residue equation reduces to,

$$\varepsilon = \int_W (h - \mathbf{g.d})^2 w d\mathbf{x},$$

where $h = \mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x})$. Differentiating the residue equation with respect to **d** and setting the result equal to zero provides the following easily solvable expression:

Gd = e,

where the 2x2 matrix $\mathbf{G} = \int_W \mathbf{gg}^T wdA$, and the 2 dimensional vector $\mathbf{e} = \int_W (\mathbf{I} - \mathbf{J})\mathbf{g}wdA$. With these expressions, **d** can be evaluated (see [26] for complete details). For a stable system, the 2x2 coefficient matrix **G** must be both above the image noise level and be well-conditioned. In turn, the noise requirement implies that both eigenvalues of **G** must be large, while the conditioning requirement means that they cannot differ by several orders of magnitude. If the two eigenvalues of **G** are λ_1 and λ_2, then a corner is accepted in a window if $\min(\lambda_1, \lambda_2) > \lambda$, where λ is a predefined threshold. The KLT corner detector and tracker (the process of corner detection and tracking are interrelated) complement each other and have been reported to perform well [26]. In our implementation of KLT, we independently extracted corners from each frame, thus eliminating any bias that might be introduced by the KLT tracker.

3 Performance Measures for Assessing the Quality of Corners for Tracking

A requirement for point feature tracking is that, having found corners in one frame, the same corners should be found and matched in successive frames, thereby constructing a time history of corners and allowing their motion to be analyzed. The ability to consistently find and match corners in this way relies on the corners being *temporally stable* and *well localized* [18]:

- **Good temporal stability** - corners should appear in every frame of a sequence (from the time they are first detected), and should not *flicker* (turn on and off) between frames.

- **Accurate localization** - the calculated image-plane position of a corner, given by the detector, should be as close to the *actual* position of the corner as possible.

Apart from the above two properties that are crucial for tracking features, a good corner detector should also be robust with respect to noise, and be efficient (computationally cheap to calculate) to run in real-time (or near real-time).

The *Kitchen-Rosenfeld*, The *KLT*, the *Harris* and the *Smith* corner detectors have been used for tracking applications in the past and have been reported to have good corner detection properties [17]. The *Harris* detector was originally developed as part of the DROID 3D vision algorithm [12] and was designed to be temporally stable. The *Smith* detector was used in the ASSET series projects [24]. ASSET used the 2D image-plane flow of corners to segment a scene into independently moving objects. The *Kitchen-Rosenfeld* detector is widely reported and has been used in many varied applications [19, 9, 6]. The *Lucas-Kanade-Tomasi* detector was employed for the KLT tracker [26] successfully. Cox *et al* have also used a variant of this detector for their MHT tracker [6]. All four corner detectors were reported to perform well as part of their respective motion algorithms.

In this paper, we use the corner localization and corner temporal stability properties to assess the quality of corner detectors for tracking applications (it is worth noting here that the internal parameters of each corner detector were adjusted to give the best possible result). A common ground to assess these corner detectors was essential. The best possible scenario was to use static image sequences. By definition, static scenes contain no moving objects and therefore no moving corners. If images could be captured with zero noise, all corners in a static scene would remain completely stationary in the image and would be *seen* in every frame. Unfortunately, this is not the case, and noise is always present in an image. A static scene is therefore an ideal way to assess the performance of corner detectors, because the motion induced by the movements of the camera are known to be zero and any failures of the detectors are due entirely to image-plane (sensor) noise.

The experiments were carried out on a variety of image sequences. First, an indoor image sequence of a toy dog is used with only artificial light interference. Secondly, an outdoor sequence of a building is considered (illuminated only with natural lighting). Then we considered an indoor lab sequence with plenty of identifiable corners (also contained direct light sources). Finally, we used a computer image sequence with lots of light reflections and curved objects. All four sequences (30 frames in length) were static (with no motion), so that the appearance and disappearance of corners in each frame was purely due to image plane noise and illumination conditions. Such a setup is ideal to analyze the stability and localization properties of the corners. Since the analysis was based on an automated tracking process, matching a corner in every frame required a suitable corner matcher (we avoided manual matching to emulate a true tracking scenario). In the following section we discuss two corner matchers, and later in the paper we evaluate their effectiveness (reliability) for corner matching.

4 Matching Corners

The feature tracking process is implemented by first extracting corner features from every frame of a static image sequence using a corner detector, and then finding a match between every corner in the initial frame and the subsequent frames. Therefore, it is important to employ a reliable feature matching strategy to correlate features between frame (no special tracking algorithms were required for this experiment).

Feature motion prediction is never completely accurate due to image noise, poor motion prediction or random motions of the camera. It is therefore very common to search for a matched feature in a *Region of Interest (ROI)*, around the predicted position of the feature in the image-plane [21]. The simplest method of corner matching is to declare the strongest corner within the ROI as being the same corner feature as the one in the previous frame using only corner positional information (commonly known as the nearest neighbor block matching technique). Although this is computationally very efficient, it is not very robust due to the presence of noise, and more significantly due to the presence of other (stronger) corners that may enter the *ROI*. A more reliable matching scheme is therefore required to prevent mismatches occurring. In the following sub-sections we discuss 2 matching schemes that have been successfully employed in many tracking applications. They are the Gradient Vector Matcher [12, 18] and the Product Moment Coefficient Matcher [21].

4.1 Gradient Vector Matcher (GVM)

The GVM was developed as part of the DROID project [12]. The DROID algorithm generated a match confidence by comparing the image-plane intensities and spatial gradients of the corner pixels to be matched. All corners with a low value of *C* (*i.e*. strong corners) are considered candidate corners for a match with the current corner. The philosophy behind this matcher is that as much of the information as possible already available via the feature extraction process should be used. Three image attributes are compared, these being $\mathbf{I}$ (grey-level intensity), $\mathbf{I}_x^{'}$ (*average* grey-level x-gradient), and $\mathbf{I}_y^{'}$ (*average* grey-level y-gradient). $\mathbf{I}_x^{'}$ and $\mathbf{I}_y^{'}$ are calculated by taking the square roots of $<\mathbf{I}_x^2>$ and $<\mathbf{I}_y^2>$ at each pixel, and by averaging over a 3 x 3 neighborhood.

Grey level intensities tend to vary from frame to frame, and so directly comparing $\mathbf{I}, \mathbf{I}_x$ and $\mathbf{I}_y$ of the candidate corner's pixels is not very robust. Most cameras have an automatic iris which regulates the amount of light falling on to the CCD, preventing it from becoming flooded. The implication of this is that bright objects in one part of an image will affect the grey-level values of objects in other regions of the same image. A vector is constructed from the three components, $\mathbf{I}$, $\mathbf{I}_x^{'}$ and $\mathbf{I}_y^{'}$ and compared to the equivalent vector of the current corner (see Fig. 4). A match confidence value *m*(**v,w**) may then be calculated by comparing the normalised magnitude of the difference vectors between each candidate corner vector (**w**) and the current corner's vector (**v**). As shown by Equation (5).

$$m(\mathbf{v},\mathbf{w}) = \frac{|\mathbf{v}-\mathbf{w}|}{\sqrt{|\mathbf{v}||\mathbf{w}|}} \tag{5}$$

Because linear changes in $\mathbf{I}$ result in linear changes in both $\mathbf{I}_x^{'}$ and $\mathbf{I}_y^{'}$, this method is invariant to linear changes in lighting conditions. The candidate corner which has the minimum value of *m*(**v,w**), as long as it is below a predefined threshold, is then declared the matched corner. This threshold therefore sets the *quality* of the match.

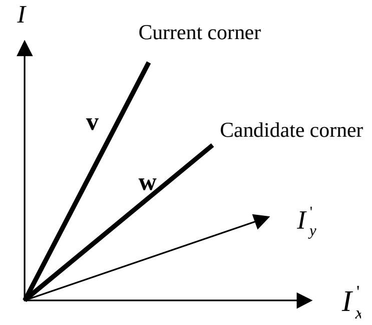


Figure 4: The Gradient Vector Matcher (GVM) - match vector

4.2 Product Moment Coefficient Matcher (PMCM)

Shapiro, *et al.* [21] used a template matching technique to find corner matches. The confidence measure used was the *product moment coefficient*, given by Equation (6).

$$cor = \frac{\sum_{i=1}^{n}(t_i - \bar{t})(p_i - \bar{p})}{\sqrt{\sum_{i=1}^{n}(t_i - \bar{t})^2 . \sum_{i=1}^{n}(p_i - \bar{p})^2}}, \qquad -1 \le cor \le 1, \tag{6}$$

Where t_i and p_i are the intensity values of the template and patch respectively (the template is the area of image about the corner to be matched, and the patch is the area of image about the candidate corner pixel in the subsequent image), $\bar{t}$ and $\bar{p}$ are their means. As with the gradient vector matcher, this measure is also invariant to lighting changes and therefore compares the *structure* of the patches, rather than their absolute intensities. Only positive values of *cor* are considered since a negative value would imply an intensity inversion. A perfect correlation is obtained when *cor* = 1. As with the GVM, a threshold is used to set the quality of the matches. Only matches having a value of *cor* above this threshold are therefore considered successful matches.

5 The Localization and Stability Tests of Corners

The temporal stability test/comparison was constructed so that a stable corner was defined as a corner that had been successfully tracked throughout the image sequence from the first frame until the current frame. The localization accuracy test differed from the temporal stability test, in that corners that had been successfully tracked for *d* frames (*d = 3* in the experiments reported in this paper) up to and including the current one were used to compare the positional accuracy of the corner detectors. The temporal stability (number of stable corners) and the localization accuracy (corner displacement-CD) measures were defined as follows:

The number of stable corners:

$$\text{No. of stable corners} = \sum_{t=1}^{F}\sum_{i=1}^{N} a(i,t), \tag{7}$$

where,

$$a(i,t) = \begin{cases} 1 \text{ if the } i\text{-th corner has been tracked for } t \text{ frames} \\ 0 \text{ otherwise} \end{cases}$$

where *N* equals the total number of corners and *F* the total number of frames in the image sequence.

The Corner Displacement-CD (pixels) is given by Eq. (8) for the *t*-th frame.

$$CD_t = \frac{\sum_{i=1}^{N} \{b(i,t) \times ([x_i(t) - x_i(t-1)]^2 + [y_i(t) - y_i(t-1)]^2)^{1/2}\}}{\sum_{i=1}^{N} b(i,t)} \tag{8}$$

where.

$$b(i,t) = \begin{cases} 1 \text{ if the } i\text{-th corner has been tracked for } d \text{ frames, and has appeared in frames } t\text{-}1 \text{ and } t \\ 0 \text{ otherwise} \end{cases}$$

The mean corner displacement – μ, variance of corner displacement - σ^2, the percentage of stable corners (γ) successfully tracked, and the number of corner matches in a frame are also useful indicators of the overall performance of each of the corner detectors. These measures are defined as given below.

The Mean Corner Displacement (MCD):

$$\mu(CD) = \frac{1}{(F-d)}\sum_{t=d}^{F} CD_t \tag{9}$$

The variance of corner displacement (σ^2):

$$\sigma^2(CD) = \frac{1}{(F-d)}\sum_{t=d}^{F} (CD_t - \mu CD)^2 \tag{10}$$

Note: The value *d* (taken to be 3 in this paper, but can be set to any value) appears because we declare a track valid only if it has been tracked for more than *d* frames.

The percentage of stable corners at the *t*-th frame:

$$\gamma_t = \left(\frac{\text{Number of stable corners in frame } t}{\text{Total no of corners in frame 1}} \right) \times 100 \tag{11}$$

The number of corner matches:

The number of corners found in each frame of a sequence that appeared in the initial frame (not necessarily stable corners). The mean matches - $\mu(mat)$, is simply the average of matches across the image sequence, and σ^2 *(mat*), is the variance for the corner matches.

These overall performance measures are calculated for each corner detector using both matching methods (GVM and PMCM) for the test image sequences considered.

6 Results

1. **Corner stability result**: The corner stability result reveals the number of stable corners identified throughout the sequence. In other words, the corners that appeared in every frame of the sequence are considered stable corners. The result is given as a percentage of the total number of corners extracted in the initial frame.
2. **First frame corner matches result**: This quantity indicates the number of corners found in the initial frame that appeared in the subsequent frames. These corners might not have appeared in every frame (appeared and disappeared throughout the sequence), but might have made their appearance in most number of frames. The mean of this quantity will give the average number of corners that were matched throughout the sequence. The first frame corner matches along with corner stability result gives a good indication of the corner detector's stability property.
3. **Corner displacement result**: The corner displacement result reveals the displacement of a corner in the *n*-th frame from its position in the initial frame (assuming that the corner considered appeared in the *n*-th frame). If the corner considered did not make an appearance in any one of the subsequent frames, then a value of $CD = 3$ pixels (displaced by 3 pixels) are assigned, indicating poor localization of that corner. The mean corner displacement value will indicate the average displacement of a corner from its initial position. An important aspect to notice for this test is the assumption that the positions of the corners in the first frame of a sequence are considered correct. Such an assumption may not be correct from a theoretical point of view, but from a practical sense, it is acceptable. This is because one would normally like to track object features from the initial frame that they view, therefore it makes sense to make the positions of the corners in the initial frame as the reference positions.

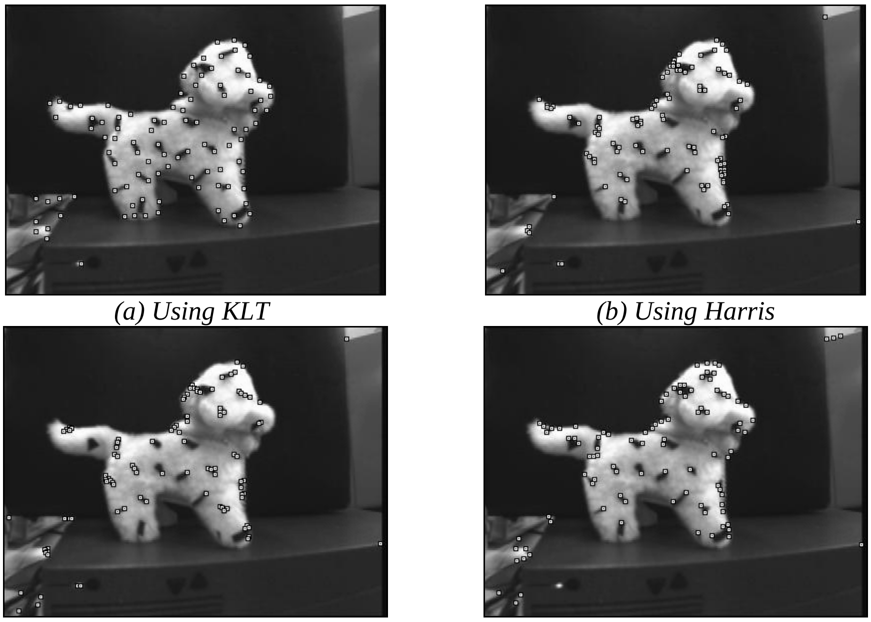
The following experiments are carried out using the above measures to assess the quality of corners extracted by the 4 corner detectors.

Experiment 1 - General performance: The above properties (1-3) are measured for each of the sequences considered.

Experiment 2 - Performance under noise: Uncorrelated Gaussian noise is added to each frame of a sequence at a specified level (experiments were carried out with noise variance ranging from 0 – 25). The 4 corner detectors are applied on the noisy sequence, and the 3 corner properties are observed at each noise level considered.

The complete results for all the experiments carried out are quantitatively and qualitatively displayed in Figures 5 - 14.

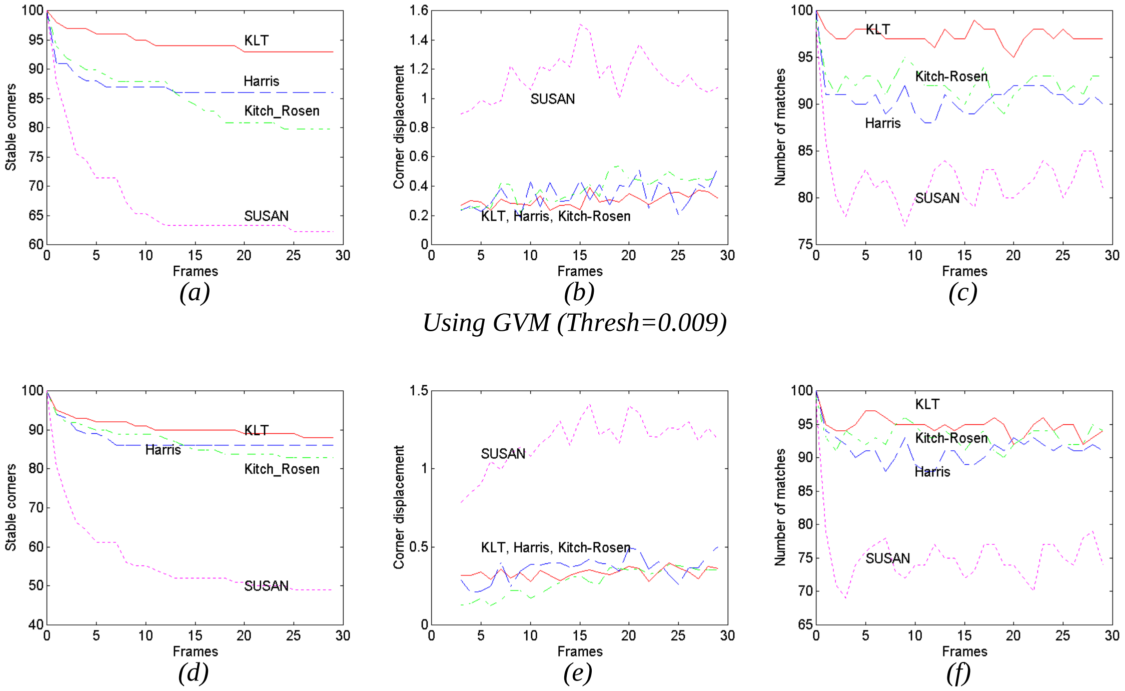
Dog Sequence Results



(a) Using KLT *(b) Using Harris*

(c) Using Kitchen-Rosenfeld *(d) Using SUSAN*

Figure 5: The best 100 corners extracted from the indoor static dog sequence. (a) KLT corner detector. (b) Harris corner detector. (c) Kitchen-Rosenfeld corner detector. (d) SUSAN corner detector.



(a) *(b)* *(c)*

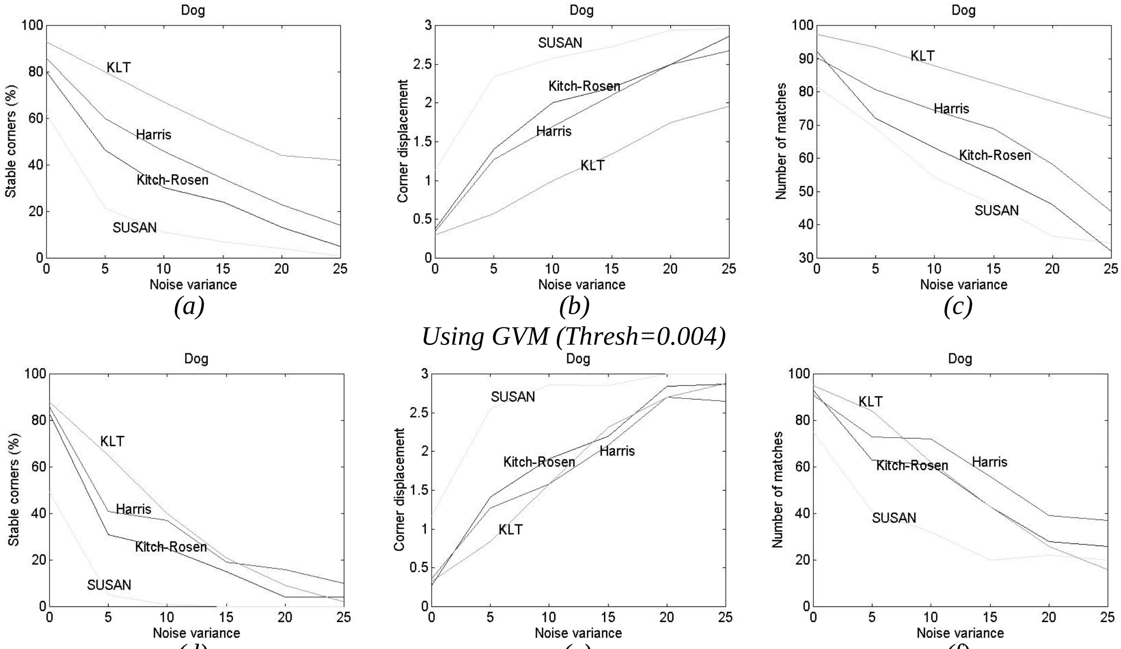
Using GVM (Thresh=0.009)

(d) *(e)* *(f)*

Using PMCM (Thresh = 0.7)

*Figure 6: Corner detector performance-test for the '**static dog**' sequence (the best 100 corners as seen by each detector are extracted from each frame). (a) Percentage stable corners (using GVM). (b) Corner displacement (GVM). (c) Number of first-frame matches (GVM). (d) Percentage stable corners (using PMCM). (e) Corner displacement (PMCM). (f) Number of first-frame matches (PMCM).*

Dog Sequence Results (with noise)



(a) *(b)* *(c)*

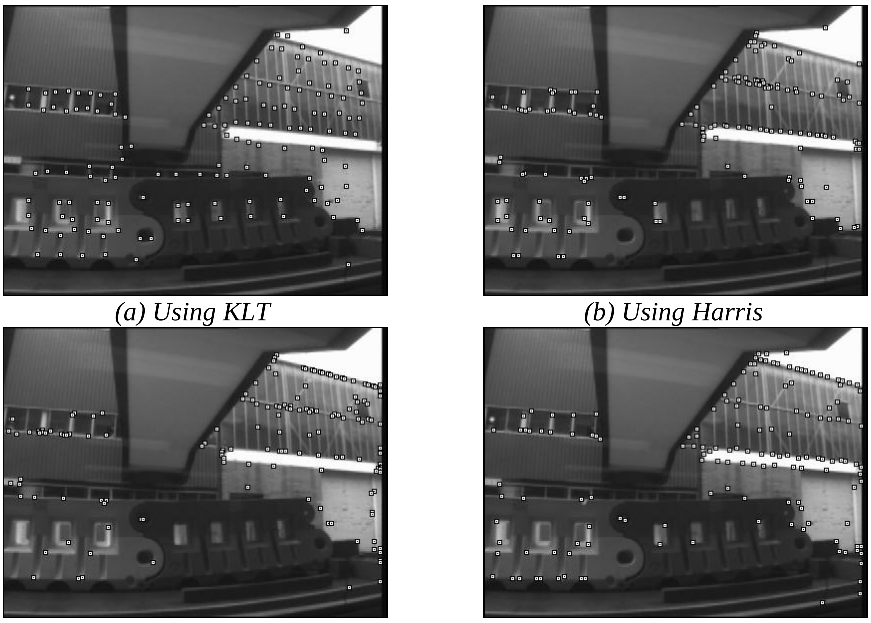
Using GVM (Thresh=0.004)

(d) *(e)* *(f)*

Using PMCM (Thresh = 0.8)

*Figure 7: Performance of the corner detectors when applied to the '**static dog'** sequence at varied noise levels (noise variance ranging from 0 – 25). (a) Percentage stable corners (using GVM). (b) Corner displacement (GVM). (c) Number of first-frame matches (GVM). (d) Percentage stable corners (using PMCM). (e) Corner displacement (PMCM). (f) Number of first-frame matches (PMCM).*

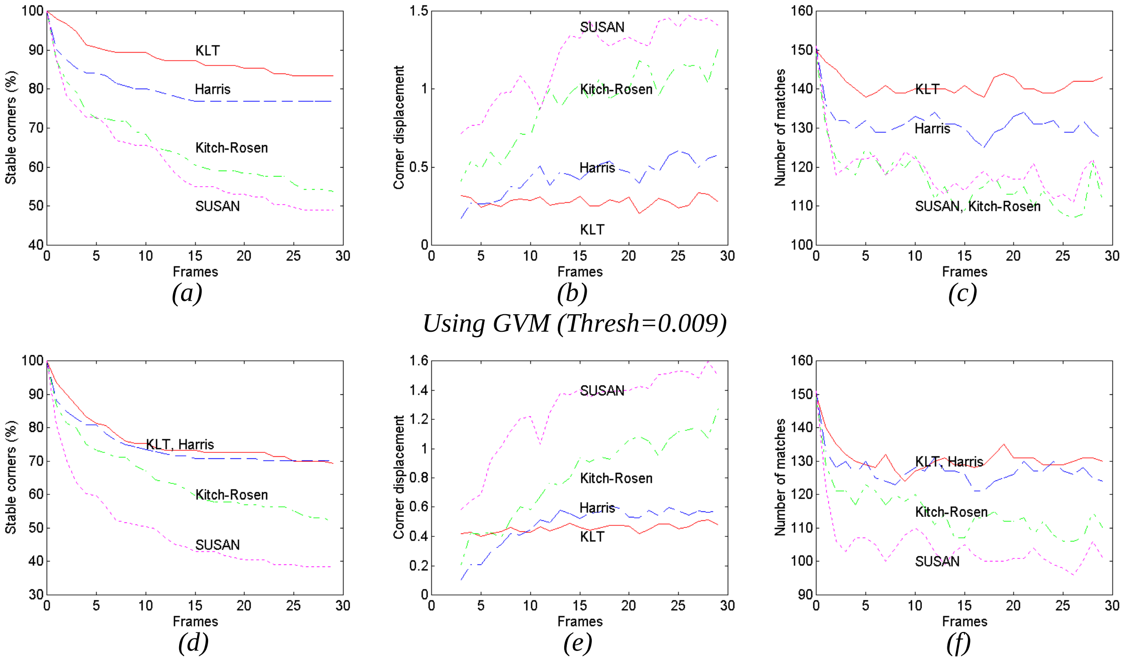
Building Sequence Results



(a) Using KLT *(b) Using Harris*

(c) Using Kitchen-Rosenfeld *(d) Using SUSAN*

Figure 8: The best 150 corners extracted from the outdoor static building sequence. (a) Using KLT corner detector. (b) Using Harris corner detector. (c) Using Kitchen-Rosenfeld corner detector. (d) Using SUSAN corner detector.



(a) *(b)* *(c)*

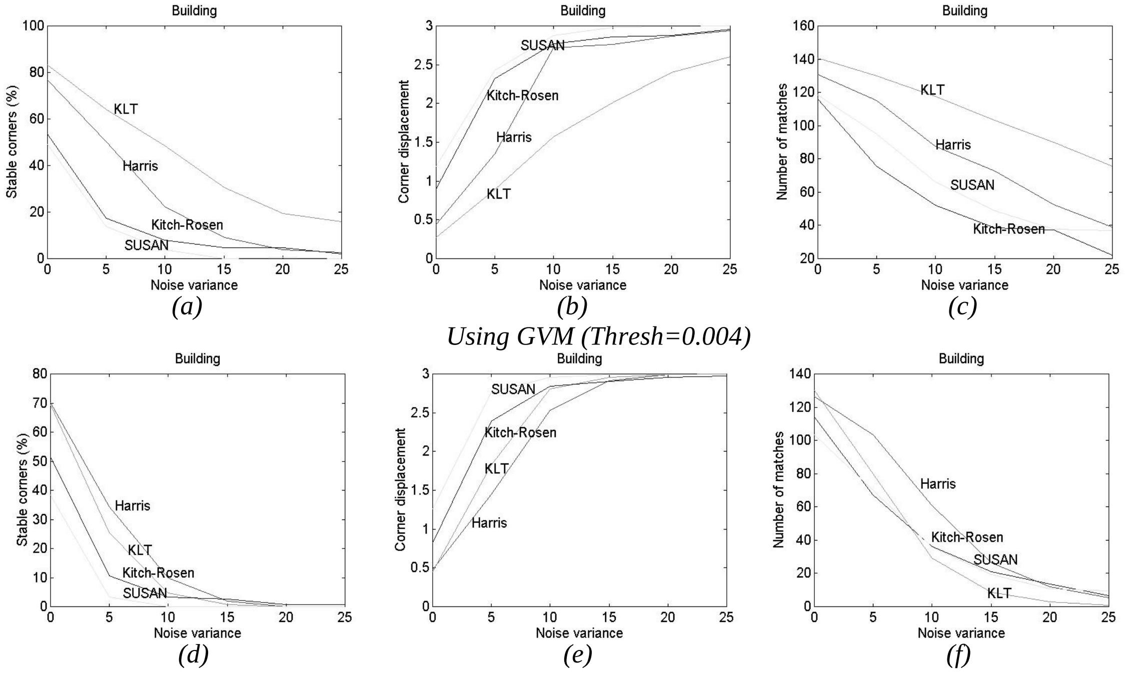
Using GVM (Thresh=0.009)

(d) *(e)* *(f)*

Using PMCM (Thresh = 0.7)

Figure 9: Performance-test for the ***'static building'*** *sequence (the best 150 corners as seen by each detector are extracted from each frame). (a) Percentage stable corners (using GVM). (b) Corner displacement (GVM). (c) Number of first-frame matches (GVM). (d) Percentage stable corners (using PMCM). (e) Corner displacement (PMCM). (f) Number of first-frame matches (PMCM).*

Building Sequence Results (with noise)



(a) *(b)* *(c)*

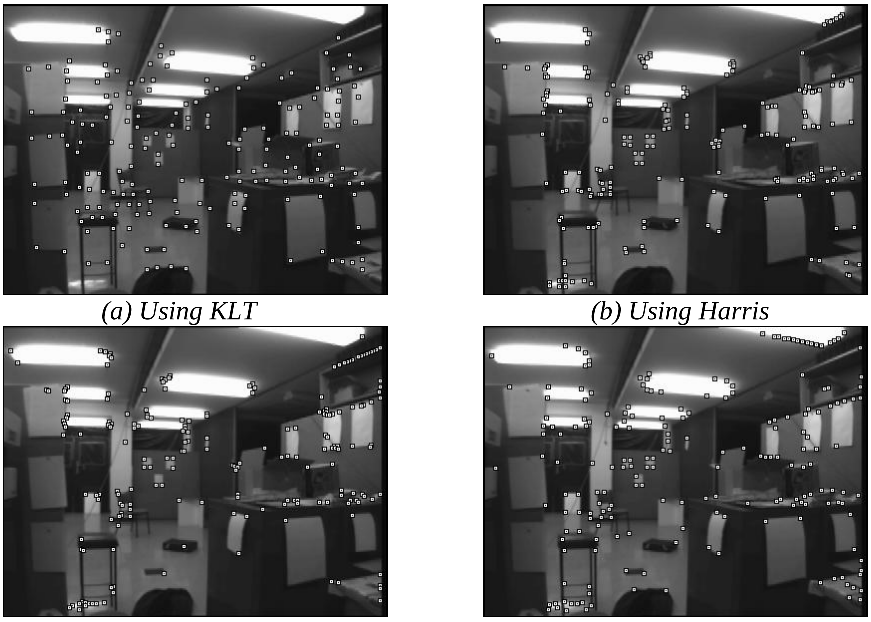
Using GVM (Thresh=0.004)

(d) *(e)* *(f)*

Using PMCM (Thresh = 0.8)

Figure 10: Performance for the ***static building*** *sequence at varied noise levels (noise variance ranging from 0 – 25). (a) Percentage stable corners using GVM. (b) Corner displacement using the GVM. (c) Number of first frame corner matches using GVM. (d) Percentage stable corners using PMCM matcher. (e) Corner displacement using the PMCM. (f) Number of first frame corner matches using PMCM.*

Lab Sequence Results



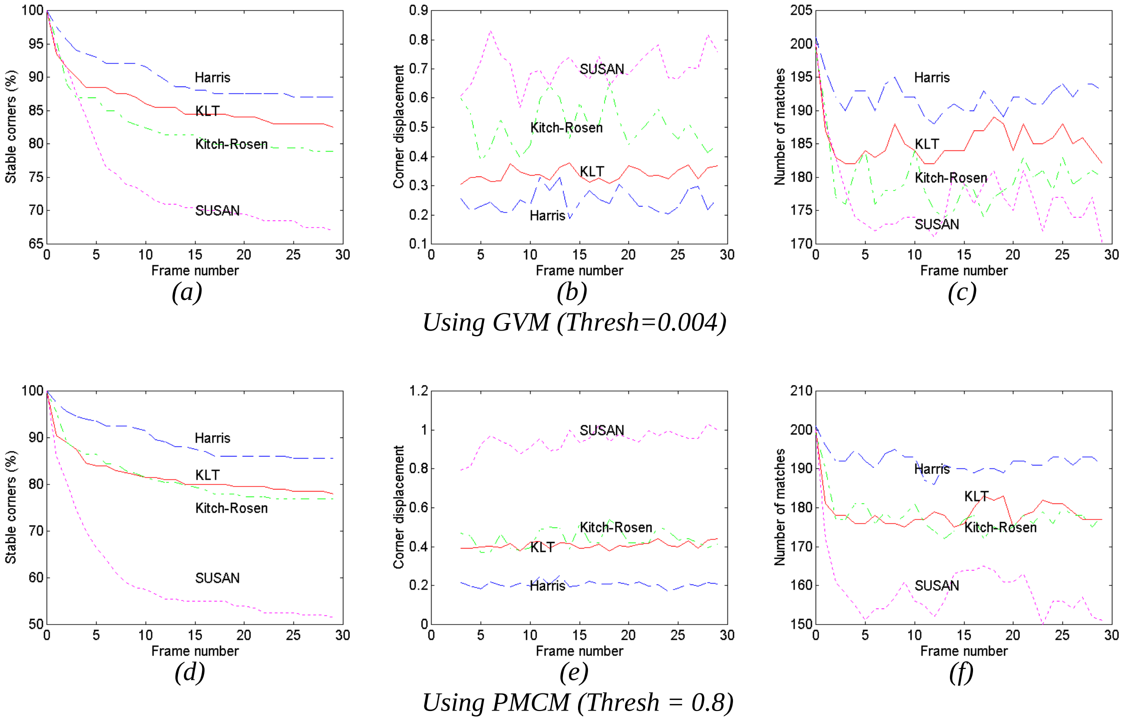
(a) Using KLT

(b) Using Harris

(c) Using Kitchen-Rosenfeld

(b) Using SUSAN

Figure 11: The best 200 corners extracted from the static lab sequence. (a) Using KLT corner detector. (b) Using Harris corner detector. (c) Using Kitchen-Rosenfeld corner detector. (d) Using SUSAN corner detector.



(a) *(b)* *(c)*

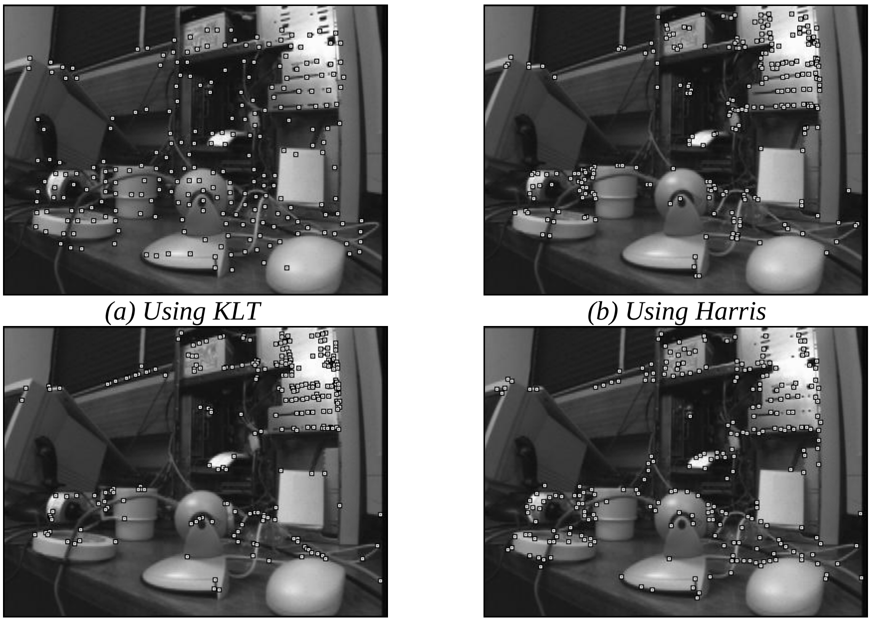
Using GVM (Thresh=0.004)

(d) *(e)* *(f)*

Using PMCM (Thresh = 0.8)

*Figure 12: Corner detector performance-test for the '**static lab**' sequence (the best 200 corners as seen by each detector are extracted from each frame). (a) Percentage stable corners (using GVM). (b) Corner displacement (GVM). (c) Number of first-frame matches (GVM). (d) Percentage stable corners (using PMCM). (e) Corner displacement (PMCM). (f) Number of first-frame matches (PMCM).*

Computer Sequence Results



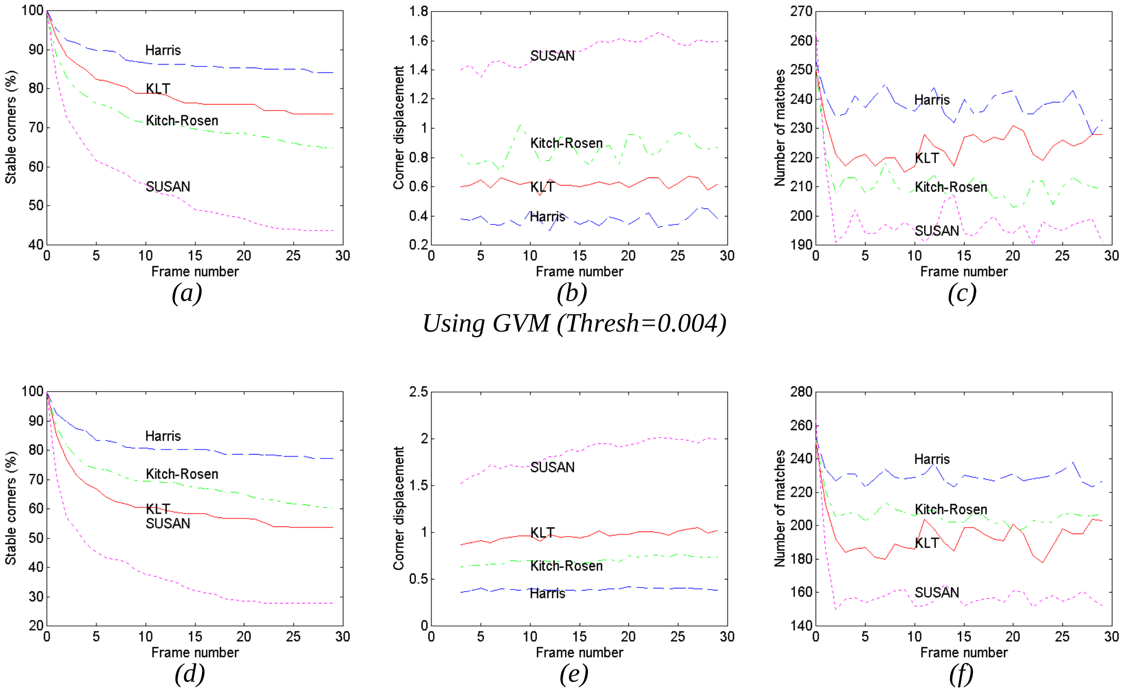
(a) Using KLT

(b) Using Harris

(c) Using Kitchen-Rosenfeld

(d) Using SUSAN

Figure 13: The best 250 corners extracted from the static computer sequence. (a) Using KLT corner detector. (b) Using Harris corner detector. (c) Using Kitchen-Rosenfeld corner detector. (d) Using SUSAN corner detector.



(a)

(b)

(c)

Using GVM (Thresh=0.004)

(d)

(e)

(f)

Using PMCM (Thresh = 0.8)

*Figure 14: Corner detector performance-test for the '**static computer**' sequence (the best 250 corners as seen by each detector are extracted from each frame). (a) Percentage stable corners (using GVM). (b) Corner displacement (GVM). (c) Number of first-frame matches (GVM). (d) Percentage stable corners (using PMCM). (e) Corner displacement (PMCM). (f) Number of first-frame matches (PMCM).*

7 Discussion

The following sub-sections give detail results on the empirical evaluations carried on the four image sequences considered.

7.1 Test Results for Indoor Dog Sequence

A 30 frame static dog sequence was considered. The 100 best corners as seen by each of the corner detectors were extracted. Fig. (5) shows the qualitative results obtained by applying the 4 corner detectors with the 100 corners superimposed on the first frame.

7.1.1 General Performance

From the results reported in Fig. 6, it can be seen that the KLT detector provided the largest number of stable corners using the GVM matcher with a 0.009 threshold (93% stable corners) and using the PMCM matcher with a 0.7 threshold (88% stable corners). The number of first-frame corner matches also indicates that KLT provided the best result. About 97% of matches are reported using GVM matcher and about 95% matches are reported employing the PMCM matcher. The mean corner displacement result indicates that the KLT provided about 0.3 pixel displacement using both matchers. Both matchers were also tested with more stringent threshold values (0.0007 for GVM and 0.9 for PMCM), which provided finer results, but the trend in performance for the 4 corner detectors were similar to that of Fig. 6. Simulations were also performed by considering two different patch sizes for the matchers (5x5 and 7x7), but no significant differences were observed.

The Harris corner detector also provided equally good results. Fig. 6 indicates that 86% stable corners are detected using GVM matcher (with 0.009 threshold) and about 87% stable corners are reported using the PMCM matcher (with 0.7 threshold). The first-frame corner matches also shows that about 90% matches are found using GVM and PMCM matchers for the same threshold values. The mean corner displacement result suggest that the Harris detector is as good as the KLT detector in providing around 0.3 pixel displacement for both matchers, which indicate good corner localization property. Tests carried out using tighter threshold values (GVM wit 0.0007 and PMCM with 0.9, not shown) reveals that Harris detector performed slightly better than the KLT detector. As before, the difference in patch sizes did not result in significant difference for each of the tests carried out.

Kitchen-Rosenfeld detector resulted with about 82% stable corners using GVM with a threshold of 0.009 and about 84% using PMCM with a 0.7 threshold. The number of first-frame matches and mean corner displacement are as good as for the Harris detector (Fig. 6). Again with tighter thresholds, the performance results are refined as one might expect.

The SUSAN detector is the least impressive of the 4 detectors. Only around 62% stable corners are reported using GVM (with 0.009 threshold) and about 50% using PMCM matcher (with 0.7 threshold). The mean first frame corner matches also indicate that on average only about 80% corners are matched throughout the sequence using GVM, and about 75% using PMCM matcher. The mean corner displacement result shows that about 1.2 pixel displacement is observed using GVM, and about 1.35 pixels using PMCM. This suggests poor localization of corners for the sequence considered. Tighter matching constraints resulted in 0% stable corners detected (for both matchers) and a poor mean corner displacement. A value of 3 pixels is assigned (indicating poor localization property) for corner displacement if there was no valid match of a corner is reported.

7.1.2 Performance under Noise

We observed the results of the 4 corner detection algorithms on the same sequence, after adding Gaussian noise to each frame (apart from frame 1) at varied noise levels (with noise variance ranging from 0 - 25) prior to applying the detectors. The quantitative results are reported in Fig. 7. As expected the quality of result decreases rapidly with added noise. The stable corners using KLT dropped from 90% (at $\sigma^2 = 0$) to about 40% (at $\sigma^2 = 25$) using GVM (at threshold 0.004), while Harris performance dropped from around 83% to 15%, Kitchen-Rosenfeld dropped from about 80% to 8%, and SUSAN dropped from 60% to 0 %. Added noise has more effect using the PMCM matcher (0.8 threshold), because it uses an image patch correlation technique to match corners in consecutive frames, thus the result can be somewhat inaccurate (See Fig. 7). The number of first frame corner matches also follows similar trend as the stable corners. The mean corner displacement also deteriorates rapidly with noise. KLT provided around 0.3 pixels at $\sigma^2 = 0$, which increased to nearly 2 pixels at $\sigma^2 =25$, while Harris result jumped from around 0.3 pixel to 2.5 pixels for the same range of noise variation. Kitchen-Rosenfeld detector reported similar corner displacement values to Harris method, while using SUSAN, the displacement increased from 1.2 pixels to nearly 3 pixels. Similar trends are also reported using the PMCM matcher (Fig. 7). The overall experiments suggest that KLT and Harris corner detectors still outperform the Kitchen-Rosenfeld and SUSAN detectors under noise.

7.2 Test Results for Outdoor Building Sequence

A 30 frame ‘static outdoor building’ sequence with only natural lighting was considered. The 150 best corners as seen by each of the corner detectors were extracted. Fig. 8 shows the qualitative results obtained by applying the 4 corner detectors.

7.2.1 General Performance

From results reported in Fig. 9, it is clear, that in general the number of percentage stable corners tracked by all 4 detectors are less than for the dog sequence. The KLT provided the best result using GVM matcher (with 0.009 threshold) with about 85% stable corners, followed by Harris detector with around 80%, Kitchen-Rosenfeld with 55%, and SUSAN with only 50% stable corners. Using PMCM matcher (with 0.7 threshold), KLT and Harris provided around 70% stable corners, followed by Kitchen Rosenfeld (50%) and SUSAN (40%) detectors. The average number of first frame corner matches using GVM is significantly higher for KLT (with around 140 corner matches), followed by Harris (with 130), then Kitchen-Rosenfeld and SUSAN with each around 117 matches. Using PMCM matcher, the KLT and Harris provide around equal number of match (130), followed by Kitchen-Rosenfeld (110) and SUSAN (100) detectors. The corner displacement test again reveals that KLT provides the best localization property with around 0.25 pixel displacement, which is followed by Harris (0.4 pixels), Kitchen-Rosenfeld (1 pixel), and SUSAN (1.5 pixels) detectors. Corner displacement test using PMCM (with 0.7 threshold) revealed that, KLT and Harris provide around 0.4 pixel displacement, followed by Kitchen-Rosenfeld (0.8 pixel), and SUSAN (1.4 pixels) detectors.

7.2.2 Performance under Noise

All 4 corner detectors were employed to extract corners from noisy frames of the building sequence (Gaussian noise is added at different variances ranging from 0 – 25). The results observed are given in Figure 10. KLT still provided the best result for the most number of stable corners using GVM (at 0.004 threshold), providing around 20% stable corners at noise variance σ^2 = 25, while the other 3 detectors provided only around 5% at σ^2 = 25. With PMCM (at 0.8 threshold), all four corner detectors resulted with 0% stable corners at σ^2 = 25. This is expected, as outdoor sequences already have image plane noise, and by adding extra noise, causes the PMCM correlation matcher to result in very low match coefficient. The number of first frame corner matches using GVM resulted with KLT having around 80 matches at noise variance at σ^2 = 25, followed by Harris (40 matches), Kitchen-Rosenfeld (40 matches), and SUSAN (with 0 matches). Using PMCM (with 0.8 threshold) provided around 10 matches for all 4 detectors at σ^2 = 25. The mean corner displacement for KLT at σ^2 = 25 is around 2.1 pixels, which is followed by the other 3 detectors providing around 3 pixels displacement. The same experiment using PMCM results with all 4 detectors providing 3 pixels displacement, indicating poor quality corner localization under considerable noise.

7.3 Test Results for the Static Lab Sequence

A 30 frame lab sequence with direct light interference was considered. The 200 best corners as seen by each of the detectors are extracted, and are qualitatively displayed in Fig. (11).

7.3.1 General Performance

The results reported in Figure 12 indicate that Harris detector provided the best result for this sequence. Harris provided nearly 90% stable corners while KLT provided about 82% stable corners. Kitchen-Rosenfeld and SUSAN detectors resulted with 80% and 67% stable corners respectively using the GVM matcher (with 0.004 threshold). The number of first-frame corner matches also indicates that Harris gives the highest number of matches than the other detectors (a mean of 192 for Harris, 185 for KLT, 177 for Kitchen-Rosenfeld and 173 for SUSAN) using the GVM matcher. The mean corner displacement result also shows that Harris provides the best result with around 0.25 pixel displacement, followed by KLT with 0.3 pixel, Kitchen-Rosenfeld with 0.5 pixel, and SUSAN with 0.7 pixel displacement using GVM matcher. The trends in observations are consistent when employing the PMCM matcher with a 0.8 threshold.

7.4 Test Results for the Static Computer Sequence

A 30 frame computer sequence with light reflections was considered. This sequence had plenty of curved objects, with ‘less easily definable’ corners, thus presenting a challenge for each of the corner detectors. The 250 best corners extracted using each corner detector is displayed in Fig. 13.

7.4.1 General Performance

Figure 14 shows that Harris detector provided the best result for the most number of stable corners (85%), followed by KLT (75%), which is followed by Kitchen-Rosenfeld (65%) and SUSAN (45%) detectors using the GVM matcher (0.004 threshold). The observations are similar for PMCM (0.8 threshold) except Kitchen-Rosenfeld performed better than KLT detector (see Fig. 17). The number of first-frame matches also shows that Harris with around 240 matches on average outperformed the KLT (220 matches), the Kitchen-Rosenfeld (210) and the SUSAN (195) detectors for both matchers. The mean corner displacement result also suggest that Harris provides the lowest displacement with 0.4 pixel, followed by KLT with 0.6 pixel, followed by Kitchen-Rosenfeld (0.9 pixel) and SUSAN (1.4 pixel) detectors for the GVM matcher. The results are consistent when using the PMCM matcher.

7.5 Overall Observation of Results

The overall observation of the results suggest that the KLT and Harris corner detectors are more suitable for tracking features in long sequences, while Kitchen-Rosenfeld and SUSAN are less reliable for long term corner tracking.

Further tests carried out (not reported in this paper due to space limitations) also suggest that for tracking large number of corner points, Harris provides slightly better result than KLT, while for tracking small number of corners, KLT provides better quality results. It is also observed that for sequences with varying light sources, Harris detector provides better quality results than KLT. The qualitative results also show (Figs. 5, 8, 11, 13) that KLT picks the best *N* corners from all parts of the image frame (which is highly desirable for multiple object tracking) while the other detectors tend to pick corners from objects where there is significant difference in contrast. Kitchen-Rosenfeld and SUSAN detectors also tend to pick several corners from edges (despite ‘edge suppression’ applied to the detectors), which is undesirable for point feature tracking applications.

The empirical evaluations also shed some insight into the matcher' s ability to correctly associate corners. The overall results suggest that for an indoor sequence, GVM or PMCM give equally good results, but for an outdoor sequence the GVM provides better quality result. This is expected, because the PMCM matcher compares a patch of image information surrounding the corner in two frames. With image plane noise (generally the case for outdoor images) one would expect a reduced correlation coefficient, which in turn leads to less reliable results. The two patch sizes examined did not make a significant difference, which indicates that a 5 x 5 image patch size is adequate for most applications. Setting matcher threshold is more of a design issue. It is important that threshold chosen should impose restraints for disallowing false corners being accepted as correct match.

These results pose the question; which is best, a matcher that produces a few good matches per frame ?, or a slightly less accurate matcher that produces more matches, but also produces a few bad matches ?. It is the opinion of the authors that a matcher that produces a high number of matches (also with high percentage of stable corners) is preferable, even if the data generated contained bad matches. A good tracking algorithm will be able to discard bad matches over a period of time.

From the results reported in this paper, it is reasonably clear that the KLT and Harris detectors are appropriate corner extractors to track point features in long image sequences, with GVM as the preferred matcher particularly for outdoor sequences, and PMCM for indoor sequences (easier to implement than GVM).

8 Conclusion

The results are interesting because unlike the *Harris* and *Kitchen-Rosenfeld* detector, the *Smith* detector uses no spatial derivatives. Spatial derivatives are normally associated with poor performance in the presence of noise since they magnify its effect, and hence it would be expected that the *Smith* detector would perform better than the *Harris* and the *Kitchen-Rosenfeld* detectors in the presence of noise (assuming that both detectors perform equally as well when there is no noise present). This is clearly not the case. A possible explanation for these results is that the *Harris* detector has a built-in smoothing function as part of its formulation. The products of the intensity gradients used to calculate the cornerness *C* are first Gaussian smoothed over a 3x3 pixel image patch. It is this smoothing that makes the *Harris* detector more robust to noise than the *Smith* detector even though it uses noise sensitive first derivatives (see also [18]).

Because of its poor performance in the presence of image-plane noise and hence its very temporally unstable corners, the *Smith* corner detector is not an appropriate feature detector for tracking in long sequences. It performed well in the ASSET project because there, segmentation was performed using two frame matching, which was good enough to produce good number of matches in consecutive frames.

The *Kitchen-Rosenfeld* method does not use a built-in smoothing function and also has second order derivatives, and as a result its' performance is poorer than the *Harris* method (*K-R* was used only as a bench-mark). The *KLT* detector on the other hand is aided by its tracking framework (the tracking process cannot be easily decoupled from the corner detection process). The *KLT* tracker and corner detectors work together to provide high quality corners as indicated by the results. The overall empirical results revealed that the *KLT* and *Harris* detectors provided the best quality corners (qualitatively and quantitatively). The corners extracted by the *Kitchen-Rosenfeld* and the *Smith (SUSAN)* detectors are less desirable for point feature tracking in long image sequences.

Further feature tracking examples with considerable motion in image sequences were also considered employing the KLT and Harris corner detectors giving good tracking performances [28, 30]. In cases where there was significant motion present (or multiple motions present) specialised tracking algorithms were employed to predict the likely positions of the features in subsequent frames (see [28, 29, 30]).

References

[1] A. M. Baumberg, "Learning deformable models for tracking human motion", PhD thesis, *School of Computer Studies, University of Leeds*, 1995

[2] A. Blake and M. Isard, Active contours, *Springer*, 1998

[3] A. Blake, M. Isard, and D. Reynard, "Learning to track the visual motion of contours", *Artificial Intelligence*, Vol.78, pp.101-134, 1995

[4] A. Blake and A. Yuille, Active Vision, *The MIT Press*, 1992

[5] J. Canny, “Computational Approach to Edge Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.8(6), pp.679-698, 1986

[6] I. J. Cox and S. L. Hingorani, “An Efficient Implementation of Reid’s Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp.138-150, Feb. 1996

[7] R. Deriche and G. Giraudon, “A Computational Approach for Corner and Vertex Detection”, *International Journal of Computer Vision*, Vol.10(2), pp.101-124, 1993

[8] L. Dreschler and H. H. Nagel, “Volumetric model and 3D trajectory of a moving car derived from monocular TV-frame sequence of a street scene”, *Computer Vision, Graphics and Image Processing*, Vol. 20, No.3, pp.199-228, 1982

[9] L. Dreschler and H. H. Nagel, “On the Selection of Critical Points and Local Curvature Extrema of Region Boundaries for Interframe Matching”, In *International Conference on Pattern Recognition*, pp.542-544, 1982

[10] M. Etoh and Y. Shirai, “Segmentation and 2D Motion Estimation by Region Fragments”, ICCV-93, May 1993, pp.192-199

[11] R. Gonzales and P. Wintz, "Digital Image Processing", *Addison-Wesley*, 1987

[12] C. G. Harris, "Tracking with rigid models", In *Blake, A. and Yuille, A., editors, Active Vision*, pp.59-74 MIT, 1992

[13] C. G. Harris and M. J. Stephens, “Combined Corner and Edge Detector”, In *Proceedings of the Fourth Alvey Vision Conference*, Manchester, pp.147-151, 1988

[14] M. Kass, A. Witkin and D. Terzopoulos, “Snakes: Active contour models”, *Int. Journal of Computer Vision*, pp.321-331, 1988

[15] L. Kitchen and A. Rosenfeld, “Gray level corner detection”, *Pattern Recognition Letters*, pp.95-102, 1982

[16] F. Meyer and P. Bouthamy. “Region-based tracking in an image sequence”, In G. Sandini (ed.), *Proc. ECCV*, pp.476-484, 1992

[17] F. Mokhtarian and R. Suomela, “Robust image corner detection through curvature scale space”, *IEEE Trans. on PAMI*, Vol.20(12), pp.1376-1381, 1998

[18] J. M. Roberts, "Attentive Visual Tracking and Trajectory Estimation for Dynamic Scene Segmentation", *Phd Thesis, Dept. of Elect. and Comp. Science*, University of Southampton, UK, 1994

[19] P. L. Rosin, “Augmenting corner descriptors”, *Graphical models and image processing*, Vol.58(3), pp.286-294, 1996

[20] J. A. Noble. “Finding Corners”, *Image and Vision Computing*, Vol.6(2), pp.121-128, 1988

[21] L. S. Shapiro, Affine analysis of image sequences, *Cambridge University Press*, New York, 1995

[22] L. S. Shapiro, H. Wang, and J. M. Brady, “A Matching and Tracking Strategy for Independently-Moving, Non-Rigid Objects”, In *3rd British Machine Vision Conference*, pp.306-315, 1992

[23] J. Shi and C. Tomasi, “Good features to track”, In *CVPR-94,* pp.593-600

[24] S. Smith, "Asset-2: Real-time motion segmentation and shape tracking", *In Proc. 5th Int. Conf. on Computer Vision*, pp.237-244, 1995

[25] S.M. Smith and J.M. Brady, "SUSAN - a new approach to low level image processing", *Int. Journal of Computer Vision*, Vol.23(1), pp.45-78, May 1997

[26] C. Tomasi and T. Kanade, “Detection and Tracking of Point Features”, Carnegie Mellon University, *Tech. Report CMU-CS-91-132,* April 1991

[27] S. Tsuji, M. Osada and M. Yachida, “Tracking and segmentation of moving objects in dynamic line images”, *IEEE Trans. Patten Anal. Machine Intell*., Vol. PAMI-2, No. 6, pp. 516-522, Nov. 1980

[28] P. Tissainayagam and D. Suter, “Visual Tracking with Automatic Motion Model Switching”, *International Journal of Pattern Recognition*, Vol.(34), pp.641-660, 2001.

[29] P. Tissainayagam and D. Suter, “Performance Prediction Analysis for Visual Tracking Algorithms”, *International Journal of Computer Vision and Image Understanding*, Vol.84(1):104-125, October 2001.

[30] P. Tissainayagam and D. Suter, “Visual tracking and motion determination using the IMM algorithm”, In *14th International Conference on Pattern Recognition (ICPR '98)*, volume 1, pages 289-291, 1998.