An Algebraic Model for fast Corner Detection

Andrew Willis*, Yunfeng Sui
University of North Carolina at Charlotte
Charlotte, NC 28223 USA
arwillis@uncc.edu

Abstract

This paper revisits the classical problem of detecting interest points, popularly known as "corners," in 2D images by proposing a technique based on fitting algebraic shape models to contours in the edge image. Our method for corner detection is targeted for use on structural images, i.e., images that contain man-made structures for which corner detection algorithms are known to perform well. Further, our detector seeks to find image regions that contain two distinct linear contours that intersect. We define the intersection point as the corner, and, in contrast to previous approaches such as the Harris detector, we consider the spatial coherence of the edge points, i.e., the fact that the edge points must lie close to one of the two intersecting lines, an important aspect to stable corner detection. Comparisons between results for the proposed method and that for several popular feature detectors are provided using input images exhibiting a number of standard image variations, including blurring, affine transformation, scaling, rotation, and illumination variation. A modified version of the repeatability rate is proposed for evaluating the stability of the detector under these variations which requires a 1-to-1 mapping between matched features. Using this performance metric, our method is found to perform well in contrast to several current methods for corner detection. Discussion is provided that motivates our method of evaluation and provides an explanation for the observed performance of our algorithm in contrast to other algorithms. Our approach is distinct from other contour-based methods since we need only compute the edge image, from which we explicitly solve for the unknown linear contours and their intersections that provide image corner location estimates. The key benefits to this approach are: (1) performance (in space and time); since no image pyramid (space) and no edge-linking (time) is required and (2) compactness; the estimated model includes the corner location, and direction of the incoming contours in space, i.e., a complete model of the local corner geometry.

*This work was supported in part by the National Science Foundation under IIS-0808718. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

1. Introduction

Detection of corners, also known as interest points, in images has been a long-standing problem in computer vision having applications in a wide variety of pattern recognition and computer vision tasks such as feature detection, 3D reconstruction from a collection of affine views, object recognition, and tracking. Many approaches have been proposed to this problem with early work on the subject dating back more than 20 years [1]. This paper proposes a new approach to this important problem and compares the performance of our approach against a wide variety of generic feature point detectors. The approach is simple to understand and implement and provides a new method for stable edge detection which has linear computational complexity and requires spatial coherence between edge points. Existing approaches such as edge linking have inherent bias (the order of that edges are linked) and can be complicated, especially at edge-junctions. Our model handles the edge image data elegantly and simply via a solution to a simple linear fitting problem.

1.1. Previous work

The most popular approach for corner detection is the Harris corner detector originally proposed in [2] for which there now exist many variants. The Harris corner detector computes the eigenvalues of the matrix $\mathbf{A}$ (see equation (1)) which can be viewed as the scatter matrix of the image gradient computed over a small region of the image.

$$\mathbf{A} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \quad (1)$$

Corners are detected by thresholding some function of the two positive eigenvalues (see [3] for one possible variant) where positive corner detection results lie at (x, y) locations where both eigenvalues of $\mathbf{A}$ are large. There is also a collection of works that compute contours from the edge image and then estimate the curvature of the contour [4, 5, 6] that detects corners from the estimated curvature of edge contours. Horaud [7] groups edge contour into lines and then looks for local intersections of these fit lines.

A second class of feature detection algorithms, which we refer to as region-based methods, look for "blobs," i.e., simple closed regions in the image that have some distinctive

characteristic and are often referred to as "blob detectors". Early work on these methods include [8] which proposed a scale-space approach to detecting blobs by computing a scale-space generated by convolving the image with a Gaussian kernel with increasing variance and subsequently detecting scale-space maxima of the Laplacian of Gaussian (LoG) operator. A number of researchers have adopted this approach or the closely related Difference of Gaussian (DoG) proposed in [9] that provides similar results at a reduced computational cost. Scale invariant versions of corner detectors include [8, 10] and [11] which provide increasingly sophisticated models for corner detection that are affine-invariant and robust to illumination changes. Further, in two separate papers Schaffalitzky [12] and Mikolajczyk [11] discuss scale-space extensions to the (LoG) based on the Hessian matrix $\mathbf{H}$ (see equation (2)) for which performance functionals may be defined for scale-space blob detection.

$$\mathbf{H} = \begin{bmatrix} \sum I_{xx} & \sum I_{xy} \\ \sum I_{xy} & \sum I_{yy} \end{bmatrix} \quad (2)$$

The second order derivatives gives strong responses on blobs and ridges.

Other important feature detectors include the very popular Scale Invariant Feature Transform (SIFT) proposed by Lowe [13] which includes many of the concepts above in a novel framework and has been shown to perform well under a wide variety of conditions it also includes methods designed for real-time applications that have low computation cost such as Features from Accelerated Segment Test (FAST) [14], Smallest Uni-value Segment Assimilating Nucleus (SUSAN) [15], and a detector similar to SUSAN proposed in [16].

Currently, the literature does not provide any explicit methods for computing edge chains within images which can be especially challenging at edge junctions. Methods such as edge-linking [17] have several shortcomings which include: (1) an inherent bias due to the sequence in which edges are "chained" together, (2) no allowance for cross-shaped, i.e., "+", junctions, and (3) iterative searches are needed to trace out and fills gaps in image edge curves. In addition, determining an appropriate value for filling-in edge gaps automatically is not clear and can significantly change detection results. In contrast, our polynomial fit respects the global organization of edge pixels within the window and has linear computational complexity with respect to the number of pixels within the window.

Our method works on points extracted from the image and locally fits a shape model to these contours that consists of a pair of intersecting lines. Our approach is distinct from other contour-based methods since we need only compute the edge image which is used to explicitly solve for the unknown linear contours that intersect to form corners in the image. The key benefits to this approach are:

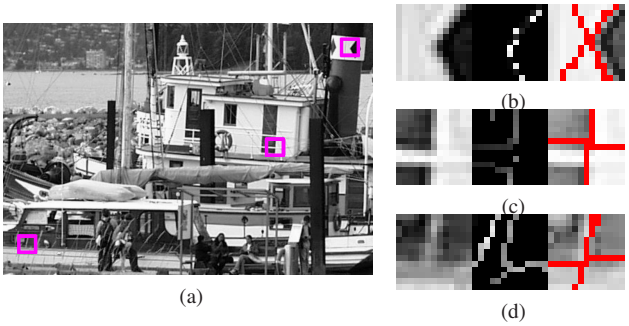

Figure 1: (a) shows a structural image of a boat, (b,c,d) each show three 13x13pixel windows taken from (a). Shown from left to right are (i) the window image data, (ii) the edge magnitude image, and (iii) the lines of the hyperbolic asymptotes that jointly model the local corner structure and whose intersection is taken as the estimated corner location.

(1) performance (in space and time); since no image pyramid (space) and no edge-linking (time) is required and (2) compactness; the estimated model includes the corner location, and direction of the incoming contours in space, i.e., a *complete* model of the local corner geometry. In contrast, methods based on the image gradient such as [2] or image surface curvature do not enforce that the gradient or curvature within the image region be spatially coherent, i.e., that locations having large first and second order derivatives lie along a continuous curve. Our model enforces this constraint which we consider to contain significant information.

2. Methodology

Our approach consists of five steps:

1. Compute the edge image (see §2.1).
2. For each (x, y) point in the edge image, compute the shape model, i.e., the coefficients of the hyperbolic curve fit a local region about (x, y) that approximates the local shape of the image contours (see §2.2).
3. Extract estimates of the corner location and the lines that best approximate the image contours (see §2.3).
4. Compute a vector of features which are functions of the edge image data and extracted curve coefficients (see §2.4).
5. Apply a threshold on the feature vectors to identify the set of salient features in the image (see §2.5).

2.1. Computing the edge image

Let $I(x, y)$ denote the recorded image. Compute the edge image, $E(x, y, \sigma)$, by computing $\mathbf{L}(x, y, \sigma) = \nabla \left(I(x, y) \star G(x, y, \sigma) \right)$ where $G(x, y, \sigma)$ is a Gaussian filter having (x, y) dimensions $W \times W$, zero mean and stan-

dard deviation σ and ∇ denotes the gradient operator implemented by convolving the image with a central difference filters: $h_x(x,y) = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$ and $h_y(x,y) = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}^t$. Edge points within the edge image image, $E(x,y,\sigma) = \|\mathbf{L}(x,y,\sigma)\|^2$, are thinned as in the Canny edge detector [18], i.e., by locally applying non-maximum suppression in the direction of the computed image gradient. Finally, a set of candidate corner positions, $\widehat{E}(x,y)$, is generated by discarding all edge points having magnitude less than the average edge magnitude value over the entire image, i.e., $\widehat{E}(x,y) = \left\{ E(x,y) | E(x,y) > \overline{E(x,y)} \right\}$ where $\overline{E(x,y)} = \frac{1}{N} \sum_x \sum_y E(x,y)$ and N is the total number of image pixels (omitting pixels at the image edge). Results generated in this paper use $W = 13$ and $\sigma = 1.4$.

2.2. Fitting hyperbolic curves to the edge image

Using another $W \times W$ window we then visit each non-zero edge pixel to apply our local corner detector. Using our knowledge of the arc-length of potential edge patterns, we quickly discard short and noisy image contours by discarding those corner candidates that include less than $W + 1$ edge points within their window. A large computational benefit is gained by also discarding all $W \times W$ windows that do not include an edge point at their center.

For the remaining edge points, we fit a hyperbola to the set of (x,y) locations within the window having non-zero gradient magnitudes. Let $\mathcal{D} = \{ (x_1,y_1), (x_2,y_2), \ldots (x_{N_w},y_{N_w}) \}$ denote the (x,y) locations of N_w edge points lying within each candidate corner's window. Our fitting method is a variant of that originally proposed by Bookstein [19] which now has many variants, some of which are discussed in [20] and [21]. This approach estimates the coefficients of a quadratic polynomial defined implicitly as shown in (3) where the coefficient vector $\boldsymbol{\alpha} = \begin{bmatrix} a & b & c & d & e & f \end{bmatrix}^t$.

$$f(x,y,\boldsymbol{\alpha}) = ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (3)$$

The 2D curve fit to the (x,y) data is taken as the zero-set of the implicit function, i.e., all (x,y) locations where $f(\mathbf{p},\boldsymbol{\alpha}) = 0$. As in [19], we minimize the squared algebraic distance between the implicit function and the data which is an approximation of the Euclidean distance (for details see discussion in [21]).

The fitting approaches in [19] and [20] focus on fitting ellipses to data which is accomplished by forcing the quadratic discriminant $J(\boldsymbol{\alpha})$ to be a positive number, i.e., $J(\boldsymbol{\alpha}) = 4ac - b^2 = 1 = 4 \begin{vmatrix} a & \frac{b}{2} \\ \frac{b}{2} & c \end{vmatrix}$. Since the zero-set of $f(\mathbf{p},\boldsymbol{\alpha})$ remains unchanged when the coefficients are scaled by a constant, i.e., $f(\mathbf{p},\boldsymbol{\alpha}) = 0 = f(\mathbf{p},k\boldsymbol{\alpha})$ for all scalars k, the particular value we constrain $J(\boldsymbol{\alpha})$ to have is unimportant.

In general, we wish to fit a pair of intersecting lines to the data. Yet, implementation of such a fitting method requires enforcing non-linear (cubic) constraints on the values of the polynomial coefficients. Instead, our method fits a hyperbolic curve which requires a quadratic constraint on the polynomial coefficients, a problem that can be explicitly solved. However, to fit hyperbolic curves we must change the sign of the quadratic discriminant constraint: $J(\boldsymbol{\alpha}) = 4ac - b^2 = -1$.

Algebraic curve fitting is accomplished by defining a monomial matrix, $\mathbf{M}$, where the i^{th} row of the matrix, $\mathbf{m}_i$, consists of the quadratic monomials computed from one of the edge positions in $\mathcal{D}$ weighted by the value of the edge magnitude at that image position (see equations (4) and (5)).

$$\mathbf{m}_i = \left\| \widehat{E}(x_i,y_i) \right\|^2 \begin{bmatrix} x_i^2 & x_i y_i & y_i^2 & x_i & y_i & 1 \end{bmatrix} \quad (4)$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_1^t & \mathbf{m}_2^t & \mathbf{m}_3^t & \ldots & \mathbf{m}_{N_w}^t \end{bmatrix}^t \quad (5)$$

The design matrix, $\mathbf{C}$, is used to constrain the coefficients such that the fit is a hyperbolic curve. Given only this constraint, it is theoretically possible that the algebraic fit could be two intersecting lines; yet, this solution is very unlikely to occur due to noise, and, as one will see as we proceed, these instances do not adversely effect the proposed algorithm.

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

The constrained fitting problem is then solved by introducing a Lagrange multiplier as shown in (7).

$$\left(\mathbf{M}^t\mathbf{M} - \lambda \mathbf{C} \right) \boldsymbol{\alpha} = 0 \quad (7)$$

The solution to (7) is taken as the eigenvector associated with the smallest eigenvalue of $\left(\mathbf{M}^t\mathbf{M}\right)^{-1}\mathbf{C}$ which we denote as $\boldsymbol{\alpha}$, the coefficients of the hyperbolic curve that minimizes the squared algebraic distance. For integer edge coordinates, degenerate edge structures such as horizontal and vertical lines and perfect cross junction can generate numerical instability in the fit. This is avoided by adding a minute (1E-4) amount of random noise to integer edge locations.

2.3. Computing corner location and local shape

The designed corner detector seeks to find image regions where two linear contours intersect. Our shape model for these linear contours are the asymptotes of a hyperbolic algebraic curve fit to the edge image data. The coefficients

of these lines are an explicit function of the hyperbolic coefficients that can be computed directly from the coefficients. This may be accomplished by bringing the fit polynomial into standard position, i.e., Euclidean transforming the equation (3) such that $f(x^{'}, y^{'}, \boldsymbol{\alpha}^{'}) = a^{'}x^{'2} + c^{'}y^{'2} + d^{'}x^{'} + e^{'}y^{'} + f^{'} = 0$ such that $a' > 0$ (this implies $c^{'} < 0$). The Euclidean rotation may be found by diagonalizing the matrix of quadratic coefficients, i.e., rotating by $-\theta$ as defined in equation (8), and the Euclidean translation may be found by completing the square (see [22] for details). The asymptotic lines $\mathbf{v_1}, \mathbf{v}_2$ are represented in parametric form as $\mathbf{l}_i = \beta \mathbf{v}_i + \mathbf{c}$ where $\mathbf{c}$ is the (x, y) position of the corner location and $\mathbf{v}_i$ is a 2D unit vector in the direction of the asymptotic line ($i = 1, 2$). Note that the estimated corner position, $\mathbf{s}$, is taken as the position in the image where the asymptotic lines intersect. Explicit equations for $\mathbf{s}$, $\mathbf{v}_1$, and $\mathbf{v}_2$ are provided below in terms of the coefficients of the fit hyperbolic function.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

where

$$\cot(2\theta) = \frac{c - a}{b} \tag{8}$$

the corner location $\mathbf{s}$ is:

$$\mathbf{s} = (-\frac{d'}{2a'}, -\frac{e'}{2c'}) \tag{9}$$

where

$$\begin{aligned} a' &= a\cos^2\theta - b\sin\theta\cos\theta + c\sin^2\theta, & d' &= d\cos\theta - e\sin\theta \\ c' &= a\sin^2\theta + b\sin\theta\cos\theta + c\cos^2\theta, & e' &= d\sin\theta + e\cos\theta \\ & & f' &= f \end{aligned}$$

and the directions of the asymptotes is:

$$\mathbf{v}_{1,2} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \sqrt{a'} \\ \pm\sqrt{-c'} \end{bmatrix} \tag{10}$$

Since these parameters are explicit functions of the hyperbolic parameters, estimating the shape model from the coefficients requires very little computation. Using the computed asymptotes, the Euclidean distance is computed for each point that corresponds to the distance between that point and the closest point on the shape model. This value can be obtained quickly using the equations of the lines $\mathbf{l}_1$ and $\mathbf{l}_2$ mentioned above. As an optional step, we can re-estimate both the corner location and asymptote directions by discarding outliers, i.e., discard those (x, y) edge points that lie far from the fit shape model, for the results presented in this paper we discard edge pixels whose Euclidean distance is larger than 1 pixel. Figure (1) shows results that arise when fitting within small windows of the image, and Figure (2) shows a global result for the same image.

Figure 2: An example of corners detected using our algorithm

2.4. Compute corner features

Interest points are detected based on a feature vector having four components $\boldsymbol{\Phi} = \begin{bmatrix} \epsilon & \lambda & \Delta & \psi \end{bmatrix}^t$. A brief explanation of each component follows: (i) ϵ, the average Euclidean distance between the edge points and the closest line from the shape model, (ii) λ, the proportion of inliers associated with each linear model, (iii) Δ, an algebraic shape parameter (see equation (11)), and (iv) ψ, is the angle between the two line models. All of these features may be computed quickly given from the hyperbolic coefficients and the pair of lines from the shape model.

A value proportional to the angle between the lines of the shape model is computed as $\psi = \tan^{-1}\left(\frac{\mathbf{v}_2}{\mathbf{v}_1}\right)$. Both ϵ and λ are easily computed by matching the edge points from the curve fitting with the lines $\mathbf{l}_1$ and $\mathbf{l}_2$ of the shape model. Specifically, for each candidate corner, the edge points are divided into two sets based on their proximity to the two lines present in the shape model. Let set L_1 be the set of points associated with line $\mathbf{l}_1$ and L_2 be the set of points associated with line $\mathbf{l}_2$. Then $\epsilon = \Sigma_{i \in L_1} d(\mathbf{l}_1, \mathbf{p}_i) + \Sigma_{i \in L_2} d(\mathbf{l}_2, \mathbf{p}_i)$ where $\mathbf{p}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^t$ is an edge point in the vicinity of the shape model and $\lambda = \frac{card(L_1)}{card(L_1)+card(L_2)}$ where $card(S)$ denotes the cardinality, i.e., the number of elements in the set S. Finally, Δ is computed from the coefficients of the hyperbolic algebraic curve as indicated in equation (11).

$$\Delta(\boldsymbol{\alpha}) = \begin{vmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{f}{2} \\ \frac{d}{2} & \frac{f}{2} & g \end{vmatrix} \tag{11}$$

2.5. Identify the set of salient image features

Recognition of corners in this 4-dimensional feature space, for expediency, is performed by simple threshold-

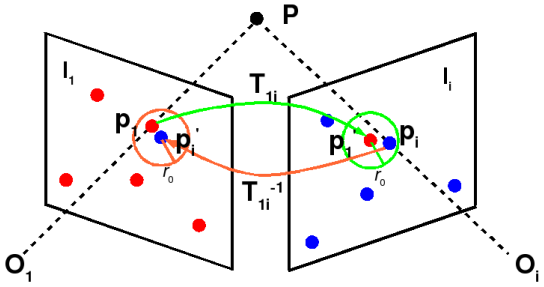

Figure 3: The criterion required for a feature correspondence between image O_1 (on left) and image O_i (on right) as defined in [25] is shown in green. Our proposed evaluation requires symmetry and uniqueness when matching features between images (in red and green), i.e., if $\mathbf{T}\mathbf{p}_i \rightarrow \mathbf{p}_j$ then $\mathbf{T}^{-1}\mathbf{p}_j \rightarrow \mathbf{p}_i$.

ing on the parameter vectors of the candidate corners. Our thresholds for corner recognition is specified as a collection of real-intervals that bound each of the vector components of $\boldsymbol{\phi}$ as follows: $\epsilon < 0.5$, $0.3 < \lambda < 0.7$, $-10^{-6} < \Delta < 10^{-6}$, $0.2rad < \psi < 1.3rad$. For our experiments, we found that recognition of structure by simple thresholding is sufficient and provides has the benefit of low computational cost. However other applications might apply more sophisticated clustering and classification methods to identify salient feature which we expect would improve these results.

Results from algebraic geometry state if $\Delta(\boldsymbol{\alpha}) < 0$ and $J(\boldsymbol{\alpha}) < 0$, then the quadratic curve must be a real-valued hyperbola [23] and the quadratic curve is a pair of real-valued intersecting lines iff $\Delta(\boldsymbol{\alpha}) = 0$. Hence, small values of $\Delta(\boldsymbol{\alpha})$ and ϵ correspond to curves that are "close" to a pair of intersecting lines in terms of their algebraic coefficients and fit the image edge pattern well. Note that, as an eigenvector, our coefficient vector is unit length, i.e., $\|\boldsymbol{\alpha}\| = 1$, which makes the determinant of the coefficients,$\Delta(\boldsymbol{\alpha})$, small (typically on the order of 10^{-5} or smaller). The resulting set of candidate corners is then reduced by applying fast non-minimum suppression as suggested in [24] on the corners in terms of their $\Delta(\boldsymbol{\alpha})$ values in a $W \times W$ window. Results for the algorithm are shown in Figure (2).

3. Evaluation Method

As others have in the past, we wish to compare our feature detection algorithm against some popular existing approaches. When examining the various approaches for evaluating feature detection algorithms, we considered the extensive work on the topic by Mikolajczyk and Schmid [26, 25, 27, 28]. However, recent work in their comparative analysis [29, 25, 27] has concentrated on similarity metrics for blob detection. Since our method returns interest points from images, we initially adopted the *repeatability rate, R,*

(a) (b) (c) (d)

Figure 4: (a-d) show examples from a standard set of test images from [25]. These images were used to evaluate our detector and compare it against several other approaches (see Figure ()).

metric suggested in [26] which measures the consistency of interest point detections under of different varieties of image variations, e.g., illumination, affine projection, etc. In [26], the repeatability rate for a pair of images (I_1, I_2) having (n_1, n_2) interest points each and a known relative affine transformation $\mathbf{T}$ is provided in equation (12)

$$R(I_1, I_2, \mathbf{T}) = \frac{\#\, correspondences}{\min(n_1, n_2)} \qquad (12)$$

where two interest points, $\mathbf{s}_1 = [x_1, y_1]^t \in I_1$ and $\mathbf{s}_2 = [x_2, y_2]^t \in I_2$ are said to correspond iff $dist(\mathbf{T}\mathbf{s}_1, \mathbf{s}_2) < r_0$ where r_0 is a pre-defined threshold. In other words, the affine-transformed position of the interest point in I_1 is within a radius r_0 of *any* corner in image I_2. Given the number of shape constraints enforced with our detection scheme, the method tends to provide a small number of detected features. However, these features satisfy a number of significantly distinct characteristics; hence we have found that the computation of these features is repeatable for our test images. The authors of [26] acknowledge that the measure (12) tends to favor methods that produce large numbers of corners such as the Harris detector which can (and often does) generate many thousands of detected interest points. In light of our highly constrained interest point detector, we made a small modification to the repeatability rate by adding a second requirement: *Two interest points,* $\mathbf{s}_1 = [x_1, y_1]^t \in I_1$ *and* $\mathbf{s}_2 = [x_2, y_2]^t \in I_2$ *are said to correspond iff* $\min_{\mathbf{s}_i \in I_2} dist(\mathbf{T}\mathbf{s}_1, \mathbf{s}_i) = \mathbf{s}_2$ *and* $dist(\mathbf{T}\mathbf{s}_1, \mathbf{s}_2) < r_0$ *and* $\min_{\mathbf{s}_i \in I_1} dist(\mathbf{s}_i, \mathbf{T}^{-1}\mathbf{s}_2) = \mathbf{s}_1$ *and* $dist(\mathbf{s}_1, \mathbf{T}^{-1}\mathbf{s}_2) < r_0$. *This criterion requires that corresponding feature points be nearest neighbors to each other under forward and inverse transformation and that their point-to-point distance be less than* r_0 *(see Figure (3)). Algorithms evaluated in this paper were performed with a value* $r_0 = 5$.

The computational speed of our algorithm was compared against two other methods: edge linking [17] and the Harris

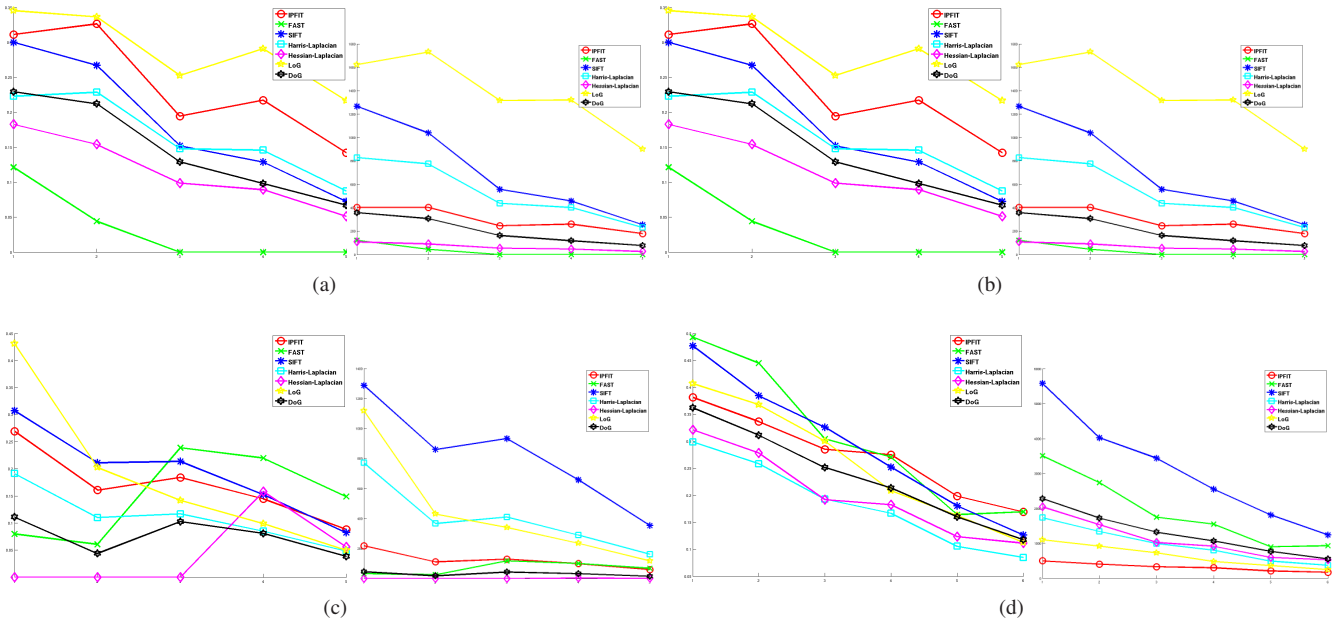

Figure 5: (a-d) show pairs of graphs. For each pair, the left graph shows the repeatability rate between images that have increasingly different content over a set of 6 images (see Figure (4) for details) and the right shows the number of interest points detected in each image. Results for seven different methods are shown, our method (IPFIT) is shown in red (see §3 for details).

corner detector [2]. The results of our tests are provided in the table below that shows the average run time for each of these methods for the image data sets shown in Figure 4(a-left), (b-right), (c-left) and (d-left).

Method	Harris	Edge-Linking	Hyperbola Fitting
Avg. Run Time (sec)	3.70	3.98	2.68

These times were recorded for implementations of these three algorithms in Matlab using code provided by the Matlab central file repository for Harris detector, code provided by authors of [17]for edge-linking and our implementation of corner detection. Tests were run on a desktop computer with an Intel Core2 Quad (Q9950) CPU and 8GB of system memory available.

4. Discussion

Implementations for the Harris-Laplacian, Hessian-Laplacian, DoG, LoG were obtained from [30] and implementation of the SIFT and FAST algorithms were provided by code available at the authors websites [13, 14]. One can see that, while our method does not always outperform others, it consistently places among the top two methods. We feel that such results show promise for the application of this feature detector for images containing man-made structures. Notable high-points in the performance of this algorithm are the results in Figure (5 a,b) obtained for the data sets shown in Figure (4 a,b) which exhibit a large amount of structure and many linear contours which are incrementally blurred (a) and have decreasing amounts of illumination (b). Lower performance is observed in Figure (5 c,d) obtained for the data sets shown in Figure (4 c,d) which exhibit changes in orientation and scale respectively. For data set (c), one can expect lower performance since this is a natural scene that exhibits contours that are rarely linear. Results for data set (d) are somewhat surprising given that our method does not currently include any modifications to cope with scale variation. A partial explanation of this unexpectedly high performance is that the overlapping region between the different scales is small and contains few corners from our method Figure (5 d-right) which artificially inflates the number of matches. Also, many structural edges have scale larger than our 13x13 window in both images which accounts for many correct matches.

5. Conclusion

A new method for corner extraction from images has been proposed that models corners as the intersection of two linear contours within a local region of the image. The method fits hyperbolic implicit polynomial curves to patterns of edge points within small local regions of the image. The asymptotes of the hyperbolic curve are used as a shape model for the linear contours in the image. Their

equations and that of their intersection point, i.e., the corner, may be computed explicitly from the coefficients of the fit hyperbolic curve. Four features are extracted from the shape model that include the angle between the lines, their goodness-of-fit to the edge points, the distribution of edge points along each line and a similarity measure that expresses the distance between the fit hyperbolic curve and a quadratic curve defined to be the intersection of two real lines. Fast classification of significant image features is accomplished using a set of defined thresholds in the feature space. The proposed method competes with many of the most popular region-based interest point detectors with no affine- or scale-invariance built-in. Given the simplicity of implementation, we feel this is an interesting result that suggests edge-base detectors such as this are perhaps preferable for structured scenes where regions are typically not radially symmetric and have non-homogeneous intensities, e.g., windows of a building. In terms of computational speed our method was found to be faster than two other edge-based corner detectors: edge linking and the classic Harris corner detector. Programming implementation variations for other methods in this paper prevented the compilation of equitable performance evaluations for all of the methods discussed in this paper. We propose a modified version of the repeatability rate for which our detector performs well, especially for structural scenes, where relatively stable and accurate matches have been observed. Compelling future modifications to this algorithm include incorporating a region with the detected feature points and the inclusion of scale invariance using LoG or DoG concepts as others have done for the Harris detector.

References

[1] H. Moravec, "Towards automatic visual obstacle avoidance," in *5th International Joint Conference on Artificial Intelligence*, p. 584, 1977. 1

[2] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, 1988. 1, 2, 6

[3] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conf. Computer Vision ans Pattern Recognition*, pp. 593–600, 1994. 1

[4] H. Asada and M. Brady, "The curvature primal sketch," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 2–14, 1986. 1

[5] H. Wang and M. Brady, "Real-time corner detection algorithm for motion estimation," *Image and Vision Computing*, vol. 13, no. 9, pp. 695–703, 1995. 1

[6] G. Medioni and Y. Yasumoto, "Corner detection and curve representation using cubic b-splines," *Computer Vision, Graphics and Image Processing*, vol. 39, no. 1, pp. 267–278, 1987. 1

[7] R. Horaud, T. Skordas, and F. Veillon, "Finding geometric and relational structures in an image," in *1st European Conference on Computer Vision*, pp. 373–384, 1990. 1

[8] T. Lindeberg and J. Garding, "Shape-adapted smoothing in estimation of 3-d shape cues from affine deformation of local 2-d brightness structure," *Image and Visual Computing*, vol. 15, no. 6, pp. 415–434, 1997. 2

[9] A. Baumberg, "Reliable feature matching across widely separated views," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 774–781, 2000. 2

[10] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 77–116, 1998. 2

[11] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *European Conference on Computer Vision*, pp. 128–142, Springer, 2002. 2

[12] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets," in *7th European Conference on Computer Vision*, pp. 414–431, 2002. 2

[13] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004. 2, 6

[14] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, pp. 430–443, 2006. 2, 6

[15] S. M. Smith and J. M. Brady, "Susan - a new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997. 2

[16] M. Trajkovic and M. Hedley, "Fast corner detection," *Image and Vision Computing*, vol. 16, no. 2, pp. 75–87, 1998. 2

[17] X. He and N. Yung, "Curvature scale space corner detector with adaptive threshold and dynamic region of support," in *17th International Conference on Pattern Recognition*, pp. 791–794, 2004. 2, 5, 6

[18] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–714, 1986. 3

[19] F. L. Bookstein, "Fitting conic sections to scattered data," *Computer Graphics and Image Processing*, vol. 9, pp. 56–71, 1979. 3

[20] W. Gander, G. H. Golub, and R. Strebel, "Least-square fitting of circles and ellipses," *BIT*, vol. 43, no. 4, pp. 558–578, 1994. 3

[21] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D. J. Kriegman, "Parameterized families of polynomials for bounded algebraic curve and surface fitting," *IEEE. Trans. Pat. Anal. and Mach. Intel.*, vol. 16, no. 3, pp. 287–303, 1994. 3

[22] E. W. Weisstein, *"Hyperbola." From MathWorld–A Wolfram Web Resource*. 2008. 4

[23] W. H. Beyer, *CRC Standard Mathematical Tables*. CRC Press., 28th ed., 1987. 5

[24] A. Neubeck and L. V. Gool, "Efficient non-maximum suppression," in *18th International Conference on Pattern Recognition*, pp. 850–855, 2006. 5

[25] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005. 5

[26] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 151–172, 2000. 5

[27] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. VanGool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1, pp. 43–72, 2006. 5

[28] T. Tuytelaars and K. Mikolajczyk, "All detectors - survey," *In CVG*, vol. 3, no. 1, pp. 1–110, 2008. 5

[29] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *8th International Conference on Computer Vision*, pp. 525–531, 2001. 5

[30] G. Dorko and C. Schmid, "Selection of scale invariant neighborhoods for object class recognition," 2003. 6