

DEB: Definite error bounded tangent estimator for digital curves

Dilip K. Prasad, *Member, IEEE*, Maylor K.H. Leung, *Member, IEEE*, Chai Quek, *Senior Member, IEEE*, and Michael S. Brown, *Member, IEEE*

Abstract—We propose a simple and fast method for tangent estimation of digital curves. This geometric-based method uses a small local region for tangent estimation and has a definite upper bound error for continuous as well as digital conics, i.e. circles, ellipses, parabolas, and hyperbolas. Explicit expressions of the upper bounds for continuous and digitized curves are derived which can also be applied to non-conic curves. Our approach is benchmarked against 72 contemporary tangent estimation methods and demonstrates good performance for both conic, non-conic and noisy curves. In addition, we demonstrate good multigrid and isotropic performance and low computational complexity of $O(1)$ and better performance than most methods in terms of maximum and average errors in tangent computation for a large variety of digital curves.

Index Terms—Digitization, digital curves, tangent estimation.

I. INTRODUCTION AND BACKGROUND

WHILE the progress in image processing and computer vision has been fast and successfully applied in complex applications like face detection, object detection, etc., some fundamental problems experienced in image processing are often neglected despite their significant influence in these high end applications [1]–[7]. One important example is the tangent estimation of digital curves. Tangent estimation is important in many applications like shape and perimeter estimations, concavity analysis, segmentation, etc. Despite the significant influence due to tangent estimation, most researchers tend to use heuristics and application specifically tailored algorithms for tangent estimation. Also, they typically use complex optimization or curve fitting based algorithms that are computationally intensive, parameter controlled, sensitive to the digitization error, noise, and distortion.

Estimating tangents in the continuous space is easy if the curves are governed by analytical equations. This problem become significantly difficult in the digitized/quantized pixel space of images, as the analytical equations may not take any continuous solution. The chosen solution is almost always an approximate integer solution nearest to the actual solution

of the analytic equations. Digitization introduces a non-linear corruption in the continuous curve which cannot be analyzed using equations [6]–[9] and introduces the following conceptual challenges towards tangent estimation:

1. The tangent is typically defined on a point, though it is a property of the continuous curve to which the point belongs. Thus, it has local as well as global properties of the curve. Due to digitization, both these properties are affected and the nature or extent of the effect cannot be quantified or analyzed using simple mathematical tools. At best, some estimates of maximum error or localized precision may be developed.

2. Usually, while estimating the tangents, prior information about the nature of the curve is unavailable. Further, appropriate size of the local region around a point is also unknown. Hence, the choice of these parameters is mainly heuristically guided and non robust.

One of the methods to find the tangents is to use continuous function (typically second order) to approximate the curvature of the digital curve in a local region around the point of interest [4], [10], [11]. Then the derivative of the continuous function is used to determine the tangent. Such approaches are restrictive in the choice of the nature of continuous function and the definition, shape, and dimension of the local region, etc. Further, there are applications where tangents need to be computed to fit a shape (for example ellipse) on the digital curve [1], [5], [12], [13]. In such cases, it is difficult to rely on a method that first fits a shape in the local region to estimate the tangent, and then uses the tangent to fit a shape to the whole curve.

In order to overcome the problem of choosing the continuous function, researchers sometimes use a Gaussian filter to smoothen the digital curve and obtain a smooth continuous curve. This smoothened curve is then used for estimating the tangents [14]. This is similar to applying a one-dimensional spatial Gaussian filter. A similar approach is taken in [15], where one-dimensional spatial median filtering is used. Another method is to consider a family of continuous curves of various types. The complete digital curve is approximated by one of the continuous curves in the family using a global optimization technique. Then the tangents are computed on the curve chosen by optimization [16]. A different approach is to approximate the digital curves using line segments. Two main variations in this approach are in vogue. The first variation is based on the theory of maximal segments [8]. At the point of interest, the maximal line segments passing through it are found and weighted convex combination of their slopes is used to find the orientation of the tangent. This method is

Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org

D. K. Prasad and M. S. Brown are with the School of Computing, National University of Singapore, Singapore 117417 email: dilipprasad@gmail.com, brown@comp.nus.edu.sg, Tel: (+65)65167097, Fax: (+65)67794580.

M. K. H. Leung is with the Faculty of Infor. & Comm. Tech., Universiti Tunku Abdul Rahman (Kampar), Malaysia, email: asmkleung@gmail.com

C. Quek is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798, email: ashcquek@ntu.edu.sg

Manuscript received May 04, 2013; revised March 25, 2014

parameter-free, has asymptotic convergence, and incorporates convexity property. Though the theory of maximal segments is well-developed and fool-proof, the assumption that their weighted combination (and the value of the weights) is a true representation of the curvature is based on heuristics, rather than analytical foundation. Despite that, in our opinion, this is the first parameter independent tangent estimation method (though involving heuristic choice of a function) that provides good properties in tangent estimation.

The second variation is to approximate the digital curve using small line segments such that the maximum deviation of any point on the digital curve with one of the fitted line segments is small; for example, below a threshold value of a few pixels [14], [17]. This procedure divides the curve into small sub-curves each corresponding to a fitted line segment. Then the slope of the tangent at the midpoint of each sub-curve is considered to be the same as the slope of the corresponding line segment. The main restriction with this method is that the tangents are available only at some points of the digital curves, viz., the mid points of the digital sub-curves. This method is similar to the concept of maximal segments, especially if the threshold of the maximum deviation is less than or equal to 1.414 pixel.

This paper proposes a tangent estimation method that is considerably simpler than the above methods and has a firm analytical foundation. Further, to estimate the slope of the tangent, only two points at a certain distance from the point of interest should be identified. This is inspired by the earlier work on the tangent estimation of elliptical curves [9].

We prove that in a continuous conic, the slope computed by our method closely matches the slope of the actual tangent. The proof is presented and numerical examples are shown. For digital curves, this work establishes the numerical error bound for the proposed tangent estimation method, which is also derived using rigorous mathematical analysis. The method is tested rigorously for performance criteria like multigrid analysis, isotropicity, and mean and maximum errors in tangent computation over all the angles of various curves. We also demonstrate the performance of the proposed method for non-conic curves and noisy curves.

Since the proposed method has definite upper bounds for continuous curve and digital curve, the proposed method is called the Definite Error Bounded (DEB) tangent estimation method. DEB can be applied to any digital or continuous curve which may be noisy or noise-free, though an explicit analytical error bound may be difficult to derive. An algorithm for DEB and its computational complexity is provided.

The outline of the paper is as follows. Section II presents the proposed tangent estimation method, its pseudocode, and its computational complexity. Section III presents the error bound of a general digital conic. In this section, section III-A presents the error bound for a continuous conic, section III-B presents the error bound due to digitization, and section III-C discussed the choice of the control parameter. Section IV presents several numerical examples to illustrate the continuous, digital, and the total error bounds in comparison to the actual errors in tangent estimation using the proposed method. Section V presents detailed comparison with other tangent estimation methods for

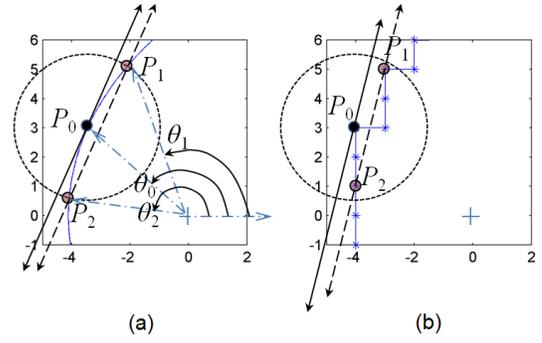


Fig. 1. Illustration of the concept for a smooth curve (a) and a digitized curve (b). (a) P_0 is the point on the continuous curve at which the tangent has to be computed. P_1 and P_2 are the points where the circle of radius R intersects the curve. (b) Digital curve corresponding to the continuous curve is shown. P_0 is the pixel at which tangent has to be computed. P_1 and P_2 are the pixels closest to the points where the circle of radius R intersects the digital curve.

several conic and non-conic curves. An application is also considered in section V-F. Section VI concludes the paper. Appendices A – C present the detailed derivations related to the content of section III-A.

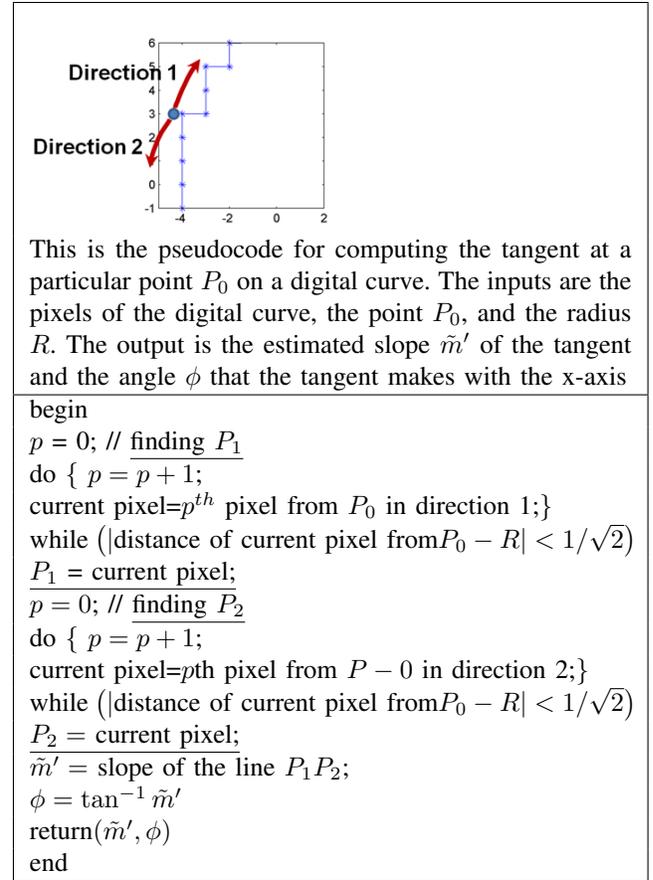


Fig. 2. Pseudocode of the proposed method.

II. PROPOSED TANGENT ESTIMATION METHOD

Consider a smooth curve shown in Fig. 1(a). Fig. 1(b) is the digital analogue of Fig. 1(a), which shall be used later, and is

introduced here for ease of comparison with the non-digitized case. For this section, Fig. 1(a) suffices in introducing the concept and should be referred in the context of the following discussion. Suppose we are interested in finding the tangent at the point $P_0(x_0, y_0)$, see Fig. 1(a). In reality, since we do not know the curve to which P_0 belongs, we cannot analytically compute the tangent. We propose to use a small circle of radius R centered at P_0 :

$$(x - x_0)^2 + (y - y_0)^2 = R^2. \quad (1)$$

The circle intersects the curve at points P_1 and P_2 , see Fig. 1(a). There are three steps for finding the tangent at P_0 :

1. Find the slope of the line P_1P_2 (denoted by \tilde{m})
2. Find a line with slope \tilde{m} passing through the point P_0 . The idea is demonstrated in Fig. 1. The slope \tilde{m} of the line P_1P_2 is given by:

$$\tilde{m} = (y_2 - y_1)/(x_2 - x_1). \quad (2)$$

3. Find the intercept $c : c = y_0 - \tilde{m}x_0$. The equation of the line is then given by:

$$y = \tilde{m}x + c. \quad (3)$$

The pseudocode for the proposed tangent estimator is presented in Fig. 2.

Computational complexity: It is evident that the above pseudocode requires a maximum of $2 \times \text{ceil}(R)$ executions of the do-while loops, where $\text{ceil}(R)$ is the smallest integer larger than or equal to R . Considering the additional two steps of computing the slope and the polar angle, the computational complexity of the algorithms is $2 \times \text{ceil}(R) + 2$ computations. Since R is a constant decided using eq. (21) (appearing later) for an application, the time required for computing the tangent is a constant (time taken for $2 \times \text{ceil}(R) + 2$ computations). So the time complexity of the proposed tangent estimator is $O(1)$. As evident, there are no shape fitting or optimization steps (as needed in most contemporary estimation methods). Thus, the computation complexity of the proposed algorithm is many magnitudes smaller than the other tangent estimation methods. In our knowledge, among other methods, implicit parabolic fitting [10], [11] has the lowest computational complexity, which is of the order $O(Q)$, where Q is the parameters that determines the local region in the vicinity of the point of interest.

III. THE ERROR BOUND FOR A GENERAL DIGITAL CONIC

This section presents the error bound of the proposed tangent estimator. First, the error bound of the proposed method for continuous conics is presented in section III-A. For convenience, this error is called the analytical error. Second, the effect of digitization is considered in section III-B. The error bound due to digitization is referred to as the digital error bound. The error bounds are used to choose the value of the control parameter R of the proposed method in section III-C.

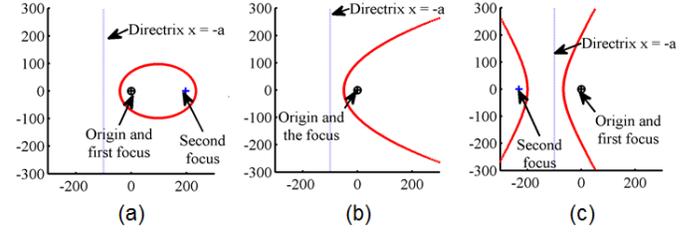


Fig. 3. Illustration of the conics, the directrix, and the foci corresponding to eq. (4) and $a = 100$. (a) an ellipse. (b) a parabola. (c) a hyperbola.

A. Error bound for a continuous conic

Let us consider a general conic equation [18]:

$$x^2 + y^2 = e^2(x + a)^2. \quad (4)$$

The above equation describes a conic of eccentricity e with one focus at the origin, the second focus (if any) along the x -axis, and the directrix given by the equation $x = -a$, where a is the distance between the first focus and the directrix (generally called the focal parameter). Fig. 3 shows samples of conics represented by the above equation. For clarity the foci and the directrix are also shown. The above equation can be greatly simplified by using the polar coordinates,

$$x = r \cos \theta, y = r \sin \theta, \quad (5)$$

as follows:

$$r(1 - e \cos \theta) = ae. \quad (6)$$

The equation of the slope of the tangent at P_0 can be computed analytically using eqs. (4) and (6) as follows:

$$m_0 = \left. \frac{dy}{dx} \right|_{P_0} = e \csc \theta_0 - \cot \theta_0. \quad (7)$$

The points $P_0(r_0, \theta_0)$, $P_1(r_1, \theta_1)$, and $P_2(r_2, \theta_2)$ lie on the conic defined by eq. (6), while the points P_1 and P_2 are also on the circle defined in eq. (1). For convenience, we substitute $\theta_i = \theta_0 + \Delta\theta_i$, $i = 1$ to 2 , where $\Delta\theta_i$, $i = 1$ to 2 are the two solutions of the simultaneous eqs. (6) and (1). Eqs. (6) and (1) are solved simultaneously for $\Delta\theta_i$, $i = 1$ to 2 in order to find the points P_1 and P_2 (details in Appendix A) and the solutions for $\Delta\theta_i$, $i = 1$ to 2 are:

$$\begin{aligned} \Delta\theta_1 &= D(dD - 1) \sum_{n=0}^{\infty} (dD)^{2n}, \\ \Delta\theta_2 &= D(dD + 1) \sum_{n=0}^{\infty} (dD)^{2n}, \end{aligned} \quad (8)$$

where

$$D = \frac{(1 - e \cos \theta_0)^2 (R/ae)}{\sqrt{(e \sin \theta_0)^2 + (1 - e \cos \theta_0)^2}}, \quad (9)$$

$$d = \frac{e \sin \theta_0}{(1 - e \cos \theta_0)}. \quad (10)$$

Further, the slope \tilde{m} of the estimated tangent given by eq. (2) can be approximated as (see details in Appendix B):

$$\tilde{m} = m_0 - 0.5e d D^3 \csc \theta_0 + O(D^4). \quad (11)$$

We make a note that:

$$D_{\max} = \max(D) = (1 + e^{-1}) (R/a). \quad (12)$$

It is shown in Appendix C that $\max(D)$ occurs at $\theta_0 = \pi$. Thus using eq. (11), \tilde{m} converges to m_0 subject to the condition that $D_{\max} \ll 1$. In the eq. (11), we need to pay additional attention to two special cases: $\theta_0 \in \{0, \pi\}$, where $\csc \theta_0$ is singular. However, noting that $d \csc \theta_0$ is not singular, there is no extra singularity other than the singularity of the actual slope m_0 . The angular error in the computation of the slope is given by (see eq. (33) in Appendix B):

$$\partial\phi \approx \left| \frac{0.5e d D^3 \csc \theta_0}{1 + m_0^2} \right|. \quad (13)$$

Specifically, for circle, i.e. $e = 0$, we get $\partial\phi = 0$. Further the error in the computation of the tangent is bounded by $\left| \frac{0.5e d D^3 \csc \theta_0}{1 + m_0^2} \right|$ and can be considered of order $O(D^3)$.

B. Error bound due to digitization

Due to digitization present in images, a general point $P(x, y)$ is approximated by a pixel $P'(x', y')$ as follows:

$$x' = \text{round}(x); \quad y' = \text{round}(y), \quad (14)$$

where $\text{round}(x)$ denotes the rounding of the value of real number x to its nearest integer. $P'(x' \in \mathbf{Z}, y' \in \mathbf{Z})$ satisfy $x' = x + \Delta x, y' = y + \Delta y$, and $-0.5 \leq \Delta x, \Delta y \leq 0.5$.

Let the slope of numeric tangent computed by pixels $P'_1(x'_1, y'_1)$ and $P'_2(x'_2, y'_2)$ (corresponding to P_1 and P_2 , as shown in Fig. 1) be denoted by \tilde{m}' . We shall call the numeric tangent computed with pixels as the digital tangent. Then \tilde{m}' can be solved as follows:

$$\tilde{m}' = \frac{(y'_2 - y'_1)}{(x'_2 - x'_1)} = \frac{\left(\tilde{m} + \frac{\Delta y_2 - \Delta y_1}{x_2 - x_1} \right)}{\left(1 + \frac{\Delta x_2 - \Delta x_1}{x_2 - x_1} \right)} \quad (15)$$

The angular difference between the numeric tangent and the digital tangent is used as the estimate of the error. This angular difference was derived in [19] and is given by:

$$\partial\tilde{\phi}_{\max} = \max \left(\frac{v}{s^3} |s^2 - s\vartheta + \vartheta^2| \right), \quad (16)$$

where $v = \left| \sin \tilde{\phi} \pm \cos \tilde{\phi} \right|$, $\vartheta = \left(\pm \cos \tilde{\phi} \pm \sin \tilde{\phi} \right)$, and

$$s = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad (17)$$

$$\tilde{\phi} = \tan^{-1}(\tilde{m}). \quad (18)$$

The result in eq. (16) proves that the error in the computation of the tangent converges even in the presence of digitization. The error in the computation of the slopes of a general continuous line and its corresponding digital line eq. (16) is shown in Fig. 4. For this, we use various values of s and plot $\partial\tilde{\phi}_{\max}$ for various values of $\tilde{\phi}$ in Fig. 4. It can be seen clearly that small values of s result in significant error while larger values of s significantly reduce the error.

The total error in the computation of the tangent is given as:

$$\partial\phi_{\max}^{\text{tot}} = \partial\phi + \partial\tilde{\phi}_{\max}. \quad (19)$$

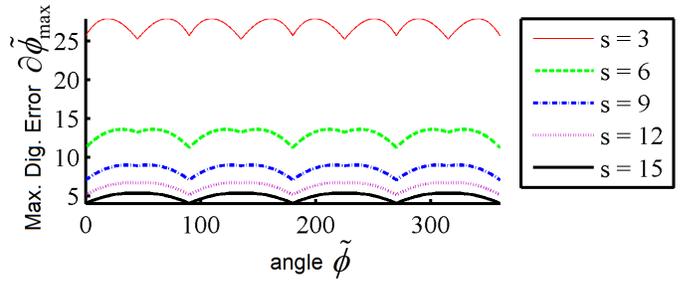


Fig. 4. Error $\partial\tilde{\phi}_{\max}$ (in degrees) for various values of s . It is evident that the error is lesser for larger values of s .

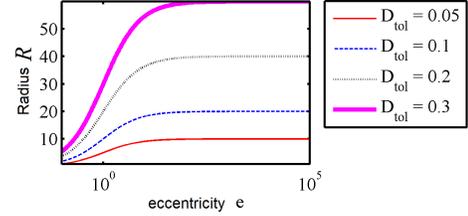


Fig. 5. Radii computed using eq. (20) for different values of eccentricity e and D_{tol} where $a = 100$.

C. Getting a realistic value of R

For the validity of the above analysis, it is required that $D_{\max} \ll 1$. Accordingly, we can choose R using:

$$R = \frac{D_{tol} a e}{1 + e}. \quad (20)$$

where D_{tol} is chosen to be very small $D_{tol} \ll 1$. The parameter R for a given eccentricity and selected values of D_{tol} is determined using eq. (20) and is plotted in Fig. 5 for $a = 100$, $e \in \{10^{-1}, 2 \times 10^{-1}, 3 \times 10^{-1}, \dots, 10^5\}$ (corresponding to 10000 conics). The maximum value of R (corresponding to $D_{tol} = 0.2$) is 60. However, for small ellipses with low eccentricity, the value of R is a few pixels.

Since the values of a and e are not available apriori in most practical scenarios, we can choose R as follows:

$$R \leq D_{tol} \rho_{\min}, \quad (21)$$

where ρ_{\min} is the radius of the smallest circle for which we intend to use the tangent estimator. It is important to consider the total error bound and the effect of R in the absence and presence of digitization. In the absence of digitization, the error bound is given by eq. (13). Upon substitution of eq. (20) in eq. (13), we see that the error bound $\partial\phi$ is proportional to R^3 . This implies that the smaller the value of R , the lesser is $\partial\phi$. In the case of digitization, the error bound is given by eq. (16) is a decreasing function of s , which in turn is related to R . The value of s is larger for larger values of R . This is illustrated using the family of conics considered above, for which the values of the minimum values of s are plotted in Fig. 6. It is seen that higher values of R result in higher values of s and consequently lower values of error due to digitization given by (16).

In general, the error due to digitization $\partial\tilde{\phi}_{\max}$ decreases with increasing value of R , while the analytical error

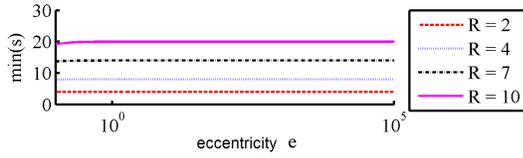


Fig. 6. Plot of $\min(s)$ for various values of R . High R implies high values for s .

$\max(\partial\phi; \forall\theta_0)$ increases with R . Thus, in the case of digitization, it is preferable to use as large values of R as possible, while satisfying eq. (21). In the event of this conflicting influence of the parameter R , eq. (21) serves as an important rule of thumb and we can choose the values closer to the upper limit given by eq. (21). Our observation is that $D_{tot} \leq 0.5$ is sufficient for the analytical error $\max(\partial\phi; \forall\theta_0)$ to be reasonably small.

For the example of the family of conics considered above, the length of the semi-minor axis of the smallest ellipse is 20.1. Assuming that the smallest circle in the family has radius $\rho_{\min} = 20$ pixels, a suitable value of R for $D_{tot} = 0.5$ is computed using eq. (21) as $R = 10$.

Now we consider the value of R in terms of the multigrid parameter h . In multigrid analysis, the parameter h determines the grid step size of an image. In other words, h^{-1} is the total number of pixels in the image. In our analysis, by default we have considered $h = 1$ owing to the digitization model given by eq. (14). However, for a general case, the suitable value of R can be given by modifying eq. (21) as follows:

$$R \leq D_{tot} \rho_{\min} h, \quad (22)$$

where, ρ_{\min} is the radius of the smallest circle in pixels. Further, using eqs. (13) and (22), the proposed tangent estimator is multigrid convergent of the order $O(h^{-3})$. **Guide for selection of the value of R :** In most images, it is reasonable to consider that the smallest circle may be of radius $\rho_{\min} = 5$ or 6 pixels, which implies that $R = 2.5$ or 3 may be used for estimating tangent. Nevertheless, if the estimated value of ρ_{\min} is higher, it is recommended to choose the largest possible value of R satisfying eq. (21).

IV. NUMERICAL EXAMPLES FOR ILLUSTRATING THE ERROR BOUND

In this section, the analytical error bound, digital error bound, and the total error bound are studied for various families of conics and they are compared against the actual error in tangent estimation. It is shown that indeed the total error bound is the upper bound for a wide range of conics.

A. Family of conics

We consider a family of conics given by $a = 200$, $e \in \{10^{-1}, 2 \times 10^{-1}, 3 \times 10^{-1}, \dots, 10^5\}$ (i.e. 10000 conics of different eccentricities), encompassing ellipses of very low eccentricity to hyperbolae of very high eccentricity. This family was also used for generating Fig. 5. The effect of the value of R on the analytical error bound ($\partial\phi$) can be seen in Fig. 7. The values of $\max(\partial\phi; \forall\theta_0)$ using four fixed values

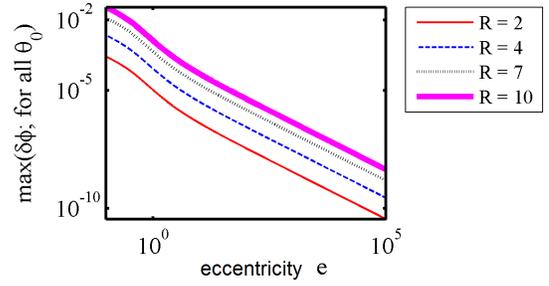


Fig. 7. Analytical error bounds for conics - $\max(\partial\phi; \forall\theta_0)$ for various values of R .

of R are plotted in Fig. 7. Evidently, for a given value of R , the conics with lower eccentricity demonstrate maximum value of error $\partial\phi$. This validates the applicability of eqn. (21) for most practical purposes. Further, using the recommended value $D_{tot} = 0.5$, the suitable value of R is computed using eq. (21) as $R = 10$. It is seen in Fig. 7 that the maximum analytical error for the value of $R = 10$ is 0.035° .

Now, the digital error bound for the above considered family of conics is studied. For a given value of R , the two points P_1 and P_2 and their corresponding digital pixels P_1' and P_2' are computed. These are used to compute \tilde{m} and \tilde{m}' , using eqs. (2) and (15) respectively. Subsequently, the actual error $\max(\partial\tilde{\phi})_{\forall\theta_0}$ due to digitization is computed for a family of conics and compared against $\partial\tilde{\phi}_{\max}$. For the family of conics considered above, the results are plotted in Fig. 8. Fig. 8(a) plots the digital error bound $\partial\tilde{\phi}_{\max}$ and the actual error due to digitization. It is noted that the actual error due to digitization is always less than $\partial\tilde{\phi}_{\max}$. Thus, it is verified that $\partial\tilde{\phi}_{\max}$ is indeed the upper bound of the error due to digitization.

Finally, the total error in tangent estimation by the proposed method is considered. The actual total error is computed as $\partial\phi^{\text{tot}} = |\tan^{-1}m_0 - \tan^{-1}\tilde{m}'|$ and is used to find the value of $\max(\partial\phi^{\text{tot}})$. The values of $\max(\partial\phi^{\text{tot}})$ for $R = 10$ are plotted in Fig. 8(b). The total error bound $\partial\phi_{\max}^{\text{tot}}$ computed using eq. (19), the analytical error bound $\max(\partial\phi; \forall\theta_0)$. The digital error bound $\partial\tilde{\phi}_{\max}$ are also plotted in Fig. 8(b). We note that $\partial\tilde{\phi}_{\max}$ is very close to $\partial\phi_{\max}^{\text{tot}}$, due to which the plot of $\partial\tilde{\phi}_{\max}$ is hardly visible in Fig. 8(b). This means that the error due to digitization is the main contributor. Further, the actual maximum error in the computation $\max(\partial\phi^{\text{tot}})$ of the ellipses is always less than $\partial\phi_{\max}^{\text{tot}}$.

B. Family of parabolae

Now, we consider a family of parabolas (i.e. $e = 1$) with $a \in [30, 500]$. We consider fixed values of R and compute analytical error bound for each value of θ_0 using eq. (13). The computed maximum angular errors are plotted in Fig. 9. The maximum error in tangent estimation $\max(\partial\phi; \forall\theta_0)$, corresponding to $R = 10$, is 0.3913° .

The actual error $\max(\partial\tilde{\phi})_{\forall\theta_0}$ due to digitization is computed for a family of conics and compared against $\partial\tilde{\phi}_{\max}$ in Fig. 10(a). It is noted that the actual error due to digitization is always less than $\partial\tilde{\phi}_{\max}$. The actual total error in the computation of tangent $\max(\partial\phi^{\text{tot}})$ is also smaller than the total error bound $\partial\phi_{\max}^{\text{tot}}$, as shown in Fig. 10(b). We just make

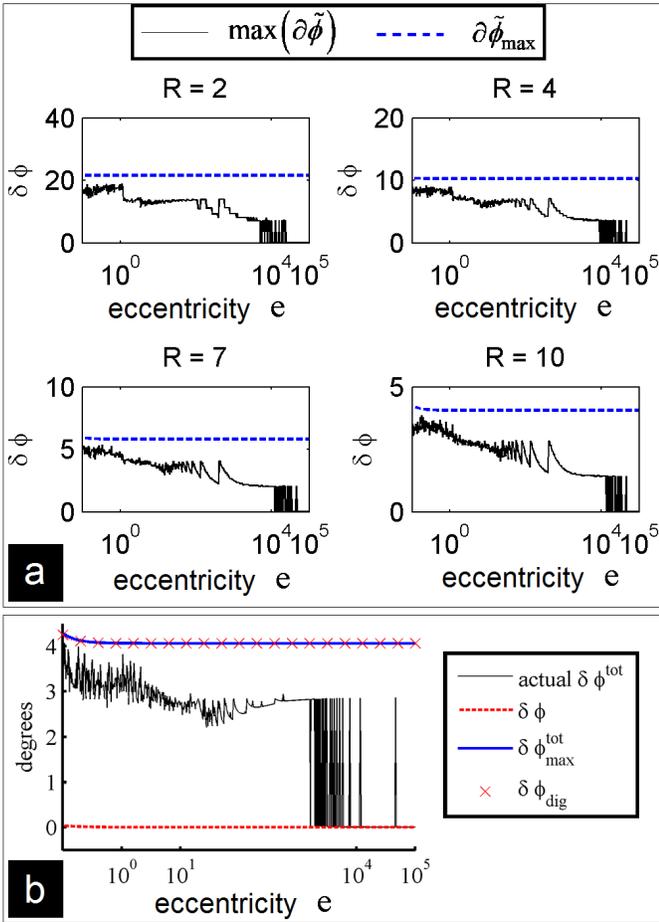


Fig. 8. Error in the computation of the tangents due to digitization for the conic family described in section IV. (a) Plots of $\partial\tilde{\phi}_{\max}$ and $\max(\partial\tilde{\phi})$ (in degrees) for various values of R . Inferences similar to (b). In addition, since $\min(s)$ does not vary much, $\partial\tilde{\phi}_{\max}$ also does not vary much with the eccentricity. (b) Plot of actual error, analytical error bound, digital error bound, and total bound for $R = 10$.

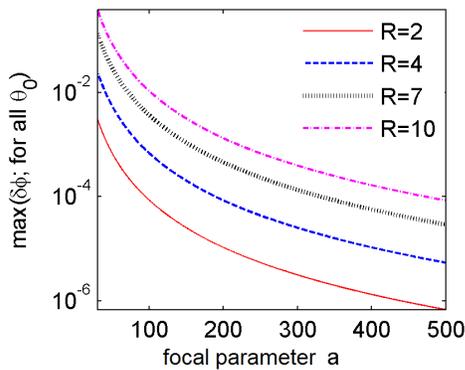


Fig. 9. Analytical error bounds for family of parabolae ($\max(\partial\phi; \forall\theta_0)$ for various values of R).

a note of the fact that for parabola with small values of a , the analytical error bound $\max(\partial\phi; \forall\theta_0)$ is non-negligible.

C. Family of circles

It was discussed in section III-A that the analytical error is $\partial\phi = 0$ for circles. However, the error due to digitization

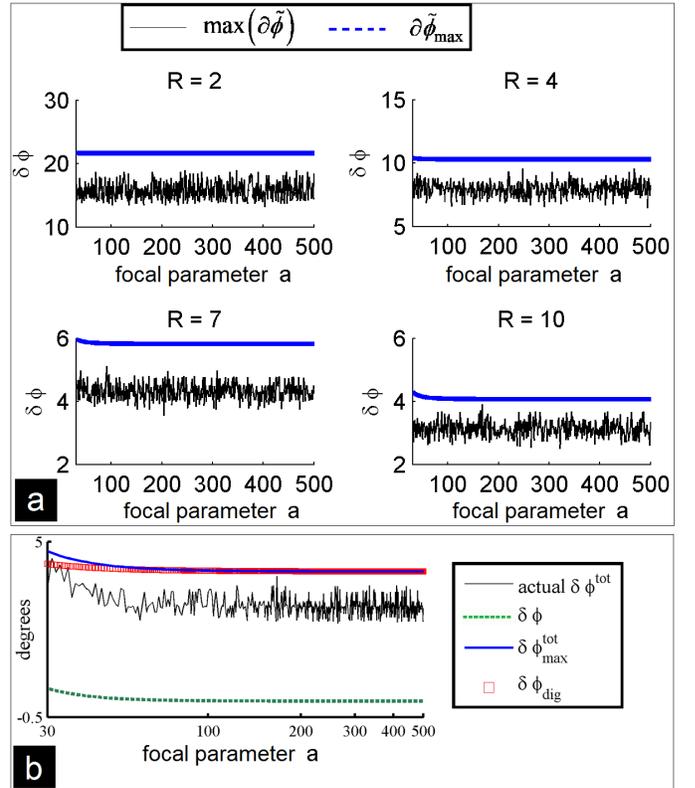


Fig. 10. Error in the computation of the tangents due to digitization for the family of parabolae described in section IV. (a) Plots of $\partial\tilde{\phi}_{\max}$ and $\max(\partial\tilde{\phi})$ (in degrees) for various values of R . Inferences similar to (b). In addition, since $\min(s)$ does not vary much, $\partial\tilde{\phi}_{\max}$ also does not vary much with the eccentricity. (b) Plot of actual error, analytical error bound, digital error bound, and total bound for $R = 10$.

$\partial\tilde{\phi}_{\max}$ is non-zero for circles. Thus, in this section, we consider a family of circles. The family contains circles with radii $\rho \in 10^z; z \in \{1.3, 1.31, 1.32, \dots, 5\}$, corresponding to 371 circles among which the smallest circle is of radius 19.95 and the largest circle of radius 10^5 . Thus, this family contains very small circles as well as very large circles. The results for this family are plotted in Fig. 11. It is noted again that the actual errors $\max(\partial\tilde{\phi})_{\forall\theta_0}$ are always less than the error bound $\partial\tilde{\phi}_{\max}$. Also, the error does not change much with the size of ellipses. This is because $\min(s)$, which is the main contributor in $\partial\tilde{\phi}_{\max}$, is strongly related to the value of R , and not ρ (until ρ is very small).

V. COMPARISON WITH OTHER ALGORITHMS

A. Algorithms used for comparison

The summary of the existing tangent estimators was provided in [8]. Based on the study performed in [8], we compare the performance of the proposed Definite Error Bounded (DEB) tangent estimator with the following tangent estimators (the codes of all of which have been developed by the authors in Matlab 2010):

- 1) **Linear regression order 1 to order 5 (LR1-LR5)** : This involves fitting an equation of order N on the coordinates of $2Q + 1$ pixels in the neighbourhood of the point of interest.

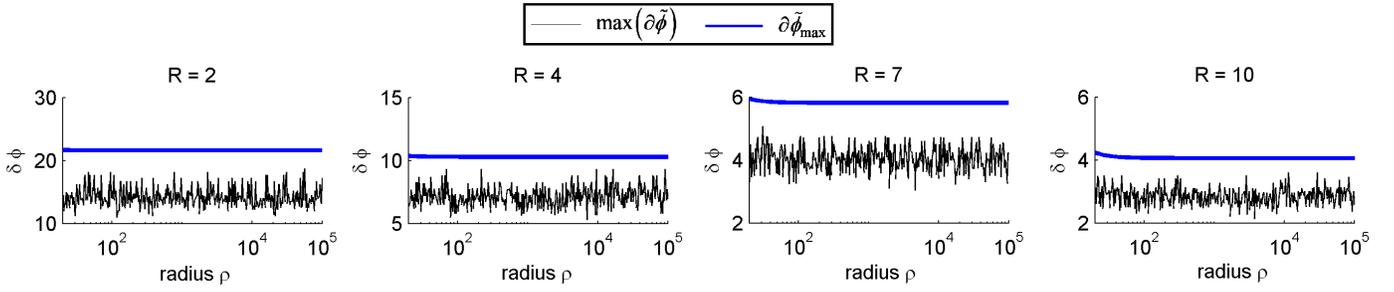


Fig. 11. Error in the computation of the tangents due to digitization for the family of circles (plots of $\partial\tilde{\phi}_{max}$ and $\max(\partial\tilde{\phi})$ (in degrees) for various values of R).

- 2) **Explicit parabola fitting (EPF)**: This involves fitting a parabolic equation on the coordinates of $2Q + 1$ pixels in the neighbourhood of the point of interest.
- 3) **Implicit parabola fitting (IPF)** [10], [11]: This is very similar to EPF with one difference that the coordinates are translated to a new coordinate system such that the point of interest is the new origin. Then analytical solutions of the parabolic equation can be directly computed.
- 4) **Independent coordinate IPF (ICIPF)** [10]: In this method, the coordinates of $2Q + 1$ pixels in the neighbourhood of the point of interest are represented as two independent parabolic functions of a fictitious parameter, say l , where the parabolic equations are determined using IPF. The derivatives for both x and y coordinates are computed with respect to the parameter l , which are subsequently used for computing the tangent.
- 5) **Gaussian derivative (GD)** [4], [14]: In this method, three variables, viz. (1) x coordinates of $2Q + 1$ pixels in the neighbourhood of the point of interest, (2) y coordinates of $2Q + 1$ pixels in the neighbourhood of the point of interest, and (3) the pixel number $q = -Q$ to Q , are considered. Two functions in the space of q are defined, which are the convolution of the $x(q)$ and $y(q)$ variables with the Gaussian derivative function of q . The tangent is then defined as the ratio of the convolved y function to the convolved x function.
- 6) **Median method by Matas (Matas)** [15]: In this method, the angles of the slopes of the lines connecting $2Q$ pixels in the neighbourhood of the point of interest (excluding the point of interest) to the point of interest are computed. The median value of these angles is used as the estimated tangents angle.
- 7) **λ MSG and λ MST** [20], [21]: A parameter independent method based on maximal segments was proposed in [20], [21]. In this a pencil of maximal segments is found for the point of interest. Then, a weighted sum of the slopes of the segments in this pencil is taken as the estimate of the tangent, where the weights are computed using a chosen function. For convenience, we call the method λ MSG if Gaussian curve is used to determine the weights and λ MST if triangular curve is used to determine the weights. We noticed that λ MST generates huge errors in the computation of tangents for certain situations because it forces the weights of the segments

at the extreme ends of the pencil to zero. Thus, we used a modified tangent function, in which the tangent function is elevated by 0.4, such that the floor of the tangent function is at 0.4.

- 8) **Hybrid methods** [8]: Methods 1-6 are dependent upon the parameter Q . In order to make them parameter free, the theory of maximal segments was used and six hybrid ways of determining the parameter Q adaptively and independently for each point of interest were proposed in [8]. These 6 hybrid ways are referred to as 10, 11, 12, 2, 3, 4 [8]. The details are avoided for the sake of brevity. The method used for computing the tangents is used as prefix. For example, EPF(01) implies that EPF was used at the core of tangent estimation and the hybrid way 01 was used for determining the parameter Q .

Summarizing the methods used for comparison, we mention that 10 parameter dependent methods (LR1-LR5, EPF, IPF, ICIPF, GD, Matas), 60 hybrid methods (6 hybrid ways for each of the 10 parameter dependent methods), and 2 parameter independent methods (λ MSG and λ MST) were used for comparison against the proposed method (DEB). Since it is difficult to represent the comparison of all the 72 methods together in a figure, we selectively present the most representative methods in all the figures, beginning from Fig. 12. By saying most representative methods, we mean a few methods that give the best performance. Also, if a hybrid method [8] gives better performance than the original method and other hybrid methods, it is preferred over all other variants of the same method. For example, if EPF(3) performs better than EPF, EPF(10,11,2,4,5), it is chosen among all of them as the representative of EPF.

B. Setup for comparison

For comparison, only closed curves are used. The curves are in a square digital image space of 300×300 pixels. For generating a digital curves, the curve is first generated using the continuous function of the curve where the center of the curve is randomly chosen within two pixel region of the center of the image space. Then, this continuous curve is digitized. For each geometry considered, 100 such random digital curves are generated.

For each of the 100 curves for a geometry, the data of $\partial\phi$ is computed for every pixel on the closed curve, such that almost all angles (with very small angular difference between

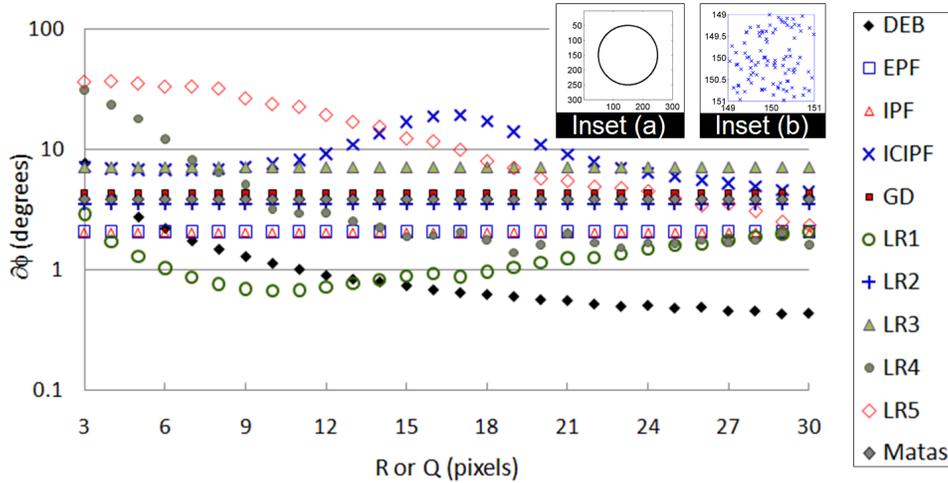


Fig. 12. The experiment with circles of radius 100 and randomly chosen centers (section V-C). Average error in estimation of tangents for 100 digital circles for various parameter dependent methods. The average error for the complete family of circles is plotted as a function of the control parameter (R for the proposed method and Q for methods 1-6 of section V-A). Inset (a): The envelope of the circles used in section V-C. Inset (b): The box shows the region from which the centers are randomly chosen. The cross marks (x) show the 100 randomly selected centers.

them) in the range $[0, 360]$ are considered. Since there are 100 curves (very slightly different from each other due to randomly chosen centers within two pixel region), one-to-one correlation between the pixels and angles is not present. So, for each curve, $\partial\phi$ is interpolated over the range $[0, 360]$ with uniform interval of 0.5 degrees. Thus, for each geometry, we get one value of $\partial\phi$ for angles $\theta = \{0, 0.5, 1, \dots, 360\}$ degrees. Then, as relevant, average and maximum values are computed over $\theta = \{0, 0.5, 1, \dots, 360\}$.

Three experiments are performed. In the first experiment, circular geometry of radius 100 is considered. In the second experiment, ellipses of various eccentricities are considered. In the third experiment, non conic shapes with inflexion points are considered. The details of each experiment and the results are discussed in subsequent sub-sections.

C. Experiment with circular geometry

In this experiment, we generate 100 circles of radius 100, and the coordinates of the centers are randomly chosen from the range $[149, 151]$, where the point $(150, 150)$ is the center of the digital image space of size 300×300 pixels. The actual envelope is shown in the Fig. 12(a) and the 100 randomly chosen centers are shown in Fig. 12(b). In Fig. 12(c), the average value of the error in tangent estimation $\partial\phi$ is presented as a function of the control parameter for DEB and the parameter dependent methods (1-6) of the list in section 5.1.

First, we discuss the performance of the proposed method (DEB). We see that as the value of R increases, the average error monotonically reduces. Since the analytical error in tangent estimation is zero for the circles, the digitization error is the only contributor of the error. As discussed in section 3.3, the value of s increases with the values of R , and as a consequence, the error in tangent estimation decreases. This explains the monotonic decrease in the $\partial\phi$ as the value of R increases.

Now, we compare the performance of the proposed method (DEB) against other parameter dependent methods. It is clearly

evident that for $R \geq 14$, DEB performs better than any other parameter dependent method. For smaller values of R , LR1 performs better than DEB for R and $Q < 14$. This can be explained using the fact that the circles have large radius in comparison to the edge segments considered using $2Q + 1$ pixels in the neighbourhood of any pixel of interest. Thus, the first order linear regression, also performs well in this case. Further, we notice that for R and $Q \leq 6$, EPF and IPF also perform better than DEB¹.

Fig. 13 summarizes the average error and maximum error of 48 various algorithms. For all the parameter dependent methods, we have chosen that value of R or Q for which the average error plotted in Fig. 12(c) is minimum². The results clearly demonstrate that DEB has lowest value of average and maximum errors. In terms of average error, GD(3) is the closest competitor, though it performs poorly for maximum error. In terms of maximum error, EPF(3) and IPF(3) and GD(2) are the closest competitors.

D. Experiment with elliptic geometry

In this experiment, we generate 10000 ellipses, which are divided into $g = 1$ to 100 groups, each group g containing $n = 1$ to 100 digital ellipses. The length of the semi-major axis of all the ellipses is fixed, i.e. $A = 100$ (note that this is different from the focal parameter a). The eccentricity of the ellipses within one group is fixed, i.e. $e_{g,n} = 0.01(g - 1); \forall n$. However, within one group, the 100 ellipses have 100 different

¹This can be explained as follows. We first make note of the fact that the error of EPF and IPF is almost invariant of the value of Q . This is because in EPF and IPF, though least squares fitting of parabola may result in large errors as the value of Q increases, the fit has the least residue in the close proximity of the point of interest and thus, the computed tangent has a reasonably small value of error.

²We have omitted the hybrid versions of LR2-LR5. LR3-LR5 are skipped because of their non-relevance for this curve, since the circle is a quadratic curve. While LR2, EPF, and IPF, all require quadratic curve fitting, EPF and IPF clearly perform better than LR2 and are thus sufficient in representing second order curve fitting. Thus, we omit LR2.

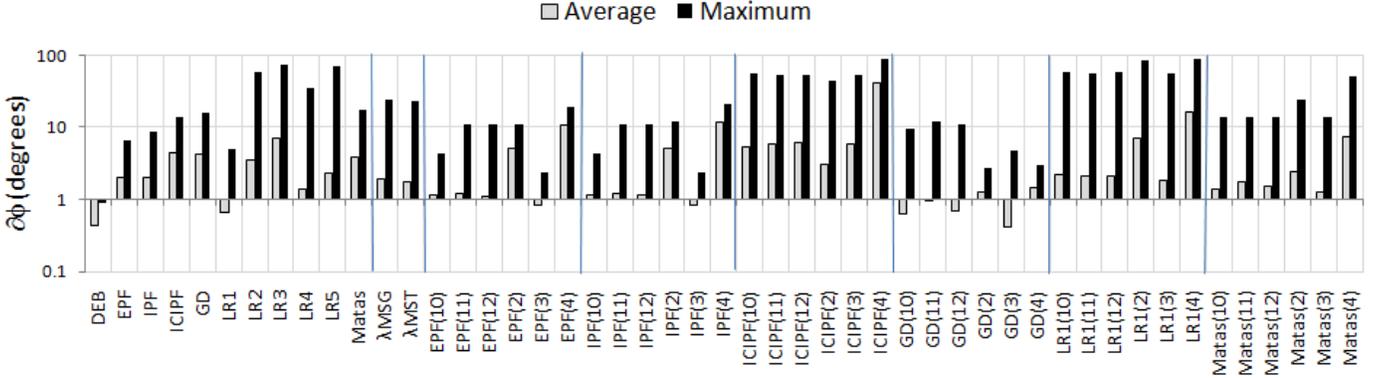


Fig. 13. Summary of results for the experiment tangent estimation for digital circles. Here, the average and maximum error for most algorithms and their hybrids are provided (a total of 48 methods). Hybrids of LR2-LR5 are avoided for brevity.

randomly chosen centers, where the centers are randomly chosen from the range $[149, 151]$ and the point $(150, 150)$ is the center of the digital image space of size 300×300 pixels.

The envelope of the digital ellipses for all the groups is illustrated in the inset of Fig. 14. As the eccentricity increases, the effect of digitization is expected to be severe for the left and right portions of the curves, as major curvature changes occur over a small digital portion of the image. On the other hand, for the bottom and top portions, the curvature hardly changes over a long portion of the curve for ellipses with high eccentricity and the error in computation of the tangent is expected to be low. See the most elliptic ellipse in the inset of Fig. 14, corresponding to $e = 0.99$ in context of the preceding discussion. Thus, for elliptic curves, the average error (averaged over all the angles) in the computation of the tangents is not a good measure of the quality of tangent computation. So, we use the maximum error over all the angles as the most representative parameter of the error in tangent computation. We compute the maximum error for each ellipse in a group and then average it over the group. Thus, we get a maximum error for one value of eccentricity.

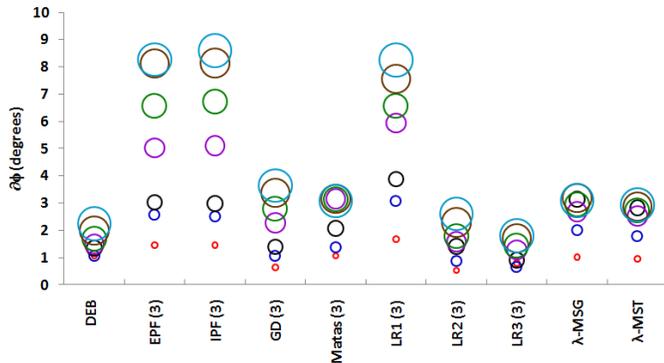


Fig. 16. Average values of errors for various algorithms. The size of the circles indicate the value of n . Smallest circle corresponds to $n = 1$, while the largest circle corresponds to $n = 7$ for each algorithm.

The maximum errors in tangent computation using various algorithms are plotted against the eccentricity of the ellipses in Fig. 14. First we note the proposed method (DEB, $R = 10$) performs better than all the methods for almost all the

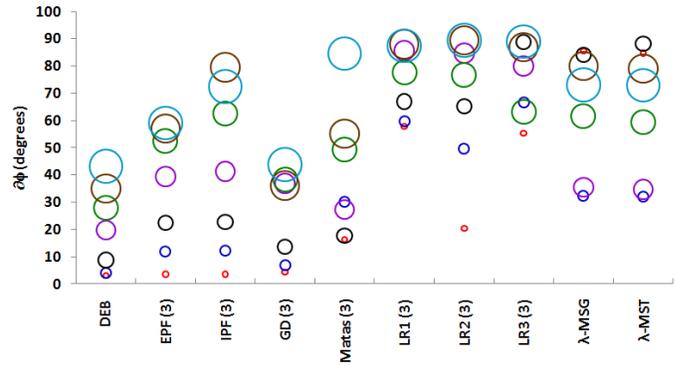


Fig. 17. Maximum values of errors for various algorithms. The size of the circles indicate the value of n . Smallest circle corresponds to $n = 1$, while the largest circle corresponds to $n = 7$ for each algorithm.

values of eccentricity. For low values of eccentricity, in general EPF(3) and IPF(3) are the next best competitors. However, for high eccentricities, GD(3) is also a good competitor. We also note that λ MSG and λ MST give reasonable error in tangent computation for low eccentricities $e < 0.3$. On the other hand, their errors become very high for high eccentricity ellipses, bordering close to 90 degrees for many values of eccentricities. In our opinion, this is because in the current forms as reported in [8], both λ MSG and λ MST are incapable handling cases with such eccentricity. We expect that a more sophisticated design of the shape function used in λ MSG and λ MST, this effect can be mitigated to a certain extent. However, this is beyond the scope of the present work and may feature in our future work.

E. Experiment with non-conic shapes containing inflexion points

In this experiment, we consider a family of non-conic curves given by:

$$r = R_{out} (1 - b \sin(n\theta)) / (1 + b), \quad (23)$$

where $x = r \cos \theta + x_0$, $y = r \sin \theta + y_0$, and R_{out} is the radius of the smallest circle (centered at origin) encompassing the entire geometry and the value of b determines the largest inner

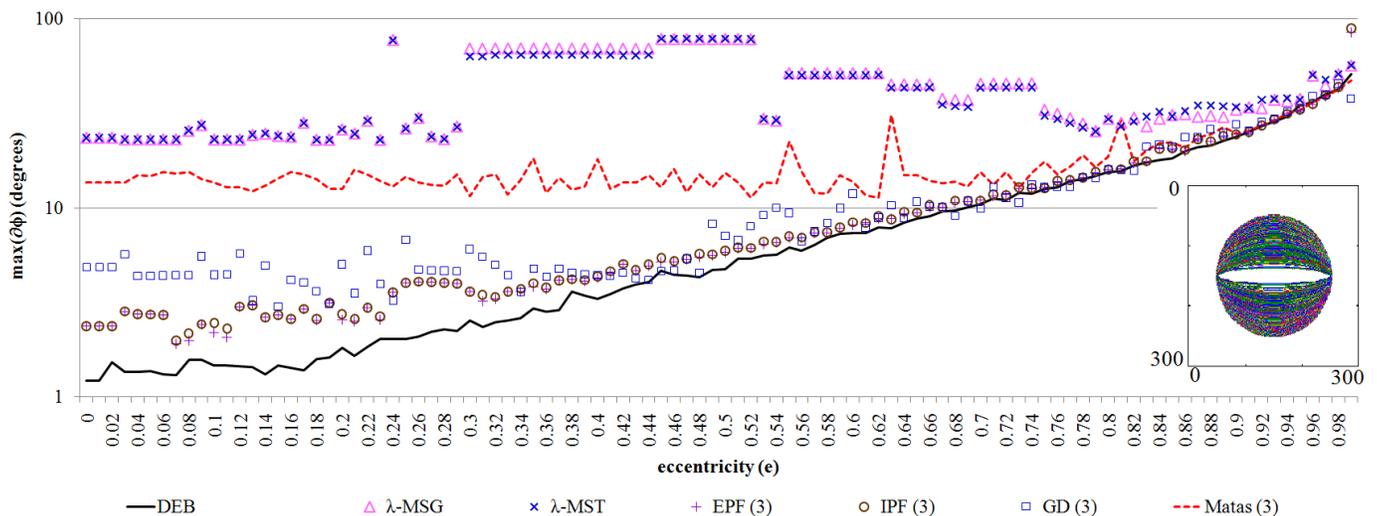


Fig. 14. Maximum error in tangent computation vs. the eccentricity for various algorithms for the experiment in section 5.4. The envelope of the ellipses is shown in the inset.

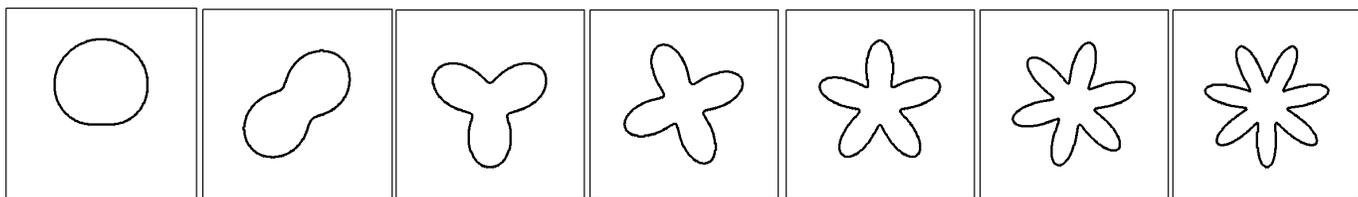


Fig. 15. Geometry given by eq. (23) for the values $n = 1$ to 7.

circle (centered at the origin) that touches the inflexion points, $R_{in}/R_{out} = (1 - b)/(1 + b)$. Also, the value of n determines the actual shape and also represents the number of inflexion points. In our experiment, we use $R_{out} = 100$ and $b = 0.5$. The example curves for $n = 1$ to 7 are shown in Fig. 15. We generate hundred such digital curves for each value of n , $n = 1$ to 7 with the center coordinates x_0 and y_0 chosen randomly from the range [149, 151].

For various algorithms, we compare the mean and maximum in Fig. 16 and Fig. 17, respectively. We have used $R = 10$ for the proposed method, DEB. In terms of the mean error, LR3(3) gives the best performance. This is because the complicated nature of curve can be better represented locally using a higher order curve than the second order curve³ Nevertheless, the performance of DEB comes really close to the performance of LR3(3). On the other hand, DEB clearly outperforms all the algorithms in terms of the maximum error, demonstrating consistent good performance for non-conic curves as well.

Further, we compare the performance of the algorithms for a noisy curve. For the non-conic given by $n = 4$, the curve is corrupted by adding Gaussian noise to the coordinates of the curves such that the envelope of the noisy curve is 6 pixels wide (see Fig. 18). The mean and maximum errors $\partial\phi$ for various methods are shown on the right side of Fig. 18. It is seen that DEB has the best performance. This is because

³On the other hand, curves of order 4, 5 or higher are too complicated for local fitting and the value of often results into underfitting of the high order curves in the local regions. For other families of curves, this effect may vary.

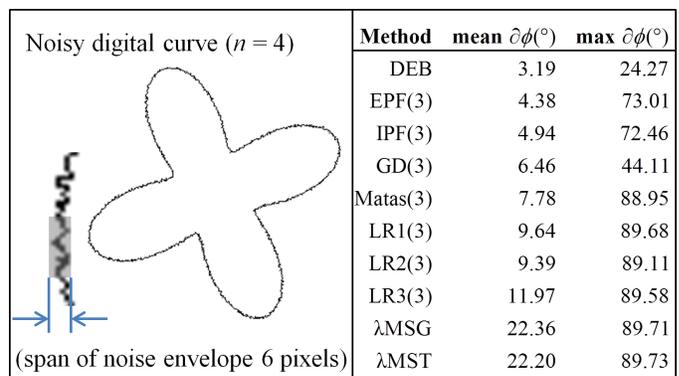


Fig. 18. Noisy non-conic curve and errors in tangent estimation using various methods

other methods consider all pixels with noise in the local region while DEB uses only two pixels closest to the radius R .

F. Application of tangent estimation in detecting elliptic shapes

We also consider a practical computer vision application that relies heavily on the accuracy of tangent estimation in digital curves. The application pertains to detecting elliptic shapes in images [13] as a part of generating shape features for object detection [22]. We consider a simple image with well defined elliptic shapes (see Fig. 19, top row). The digital edge curves detected using Canny edge detector are also shown

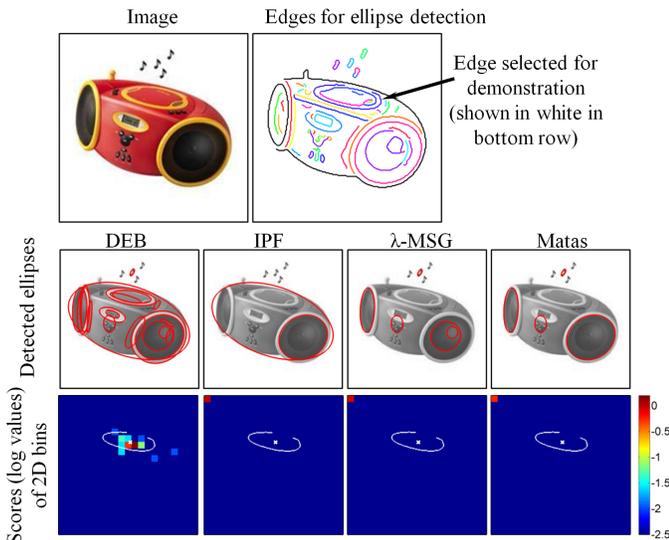


Fig. 19. An example of practical application that relies heavily on tangent estimation. Top row: example image and its digital curves. Middle row: ellipses detected by using different tangent estimators in [13]. Bottom row: scores of 2-D bins in the image region computed using [5], [13] for the white digital curve (higher score indicates higher chance of finding the ellipse's center). Actual center is shown using white 'x' marker.

in Fig. 19, top row. In [13], the method proposed in [5] which uses tangents to estimate the center of ellipses is employed as an intermediate step. In the algorithm of [13], we kept everything the same except the tangent estimation algorithm. Further, the algorithm is stopped at generating the elliptic hypotheses only and saliency based hypotheses filtering has not been employed so that all the generated ellipses can be seen. The results using different tangent estimators are shown in the second row of Fig. 19. It is seen that using DEB gives the best performance. This is because the accuracy of Yuen's method [5] is significantly dependent upon the accuracy of tangent estimation [1]. This is illustrated in the bottom row of Fig. 19 as well. It shows the scores of 2-D bins in the image region computed using [5], [13] for the white digital curve where higher score indicates higher chance of finding the center of the ellipse corresponding to the white curve in that bin. It is seen that using DEB leads to generating high score for the correct bin while using other methods leads to a very low score for the correct bin.

VI. CONCLUSION

A simple tangent estimator for conic digital curves is proposed. Explicit error bounds have been analytically derived for all kinds of conics. The upper bounds have been derived for both continuous and digitized conic curves. Extensive numerical experiments in section 4 and section 5 confirm the error bounds as being the upper bounds. In addition, since the performance and upper bounds of the algorithm are verified and shown to be small for very small as well as very large geometries (based on the value of focal parameter a), the results inherently also demonstrate the multigrid performance of the proposed algorithm (DEB). Owing to the simplicity of the proposed method, the algorithm is computationally inexpensive. The performance of DEB is compared against

72 contemporary methods of tangent estimation for various geometries, which include a large circle, ellipses of various eccentricities, and a non-conic curves with inflexion points. Using the example of circle as the standard, we show that the performance of DEB is superior to all the methods. Since 100 digital circles and all possible angles are considered, the results also demonstrate the isotropic property as discussed in [8]. DEB also gives the lowest maximum error in tangent computation for ellipses of almost all values of eccentricities. Finally, DEB demonstrates reasonable performance and the lowest value of maximum error in tangent computation for non-conic shapes with inflexion points as well. In conclusion, the proposed tangent estimator has been extensively studied, in terms of error bounds, multigrid performance, isotropicity, and statistical averages and maximum values of error in tangent computation for conic (continuous as well as digitized) curves. In addition, its utility as a generic tangent estimator is also clearly demonstrated, though the mathematical definite error bound may be difficult to compute for generic curves.

ACKNOWLEDGEMENT

The experiments in section V were inspired from [8] and we acknowledge the insightful feedback of Dr. J.O. Lachaud. We highlight that the codes for all the methods used in section 5 were developed by the authors of the paper in Matlab 2010. The Matlab source code of our method is available at <https://sites.google.com/site/diliprasad/source-codes>.

APPENDICES

A. Solution $\Delta\theta_i, i = 1$ to 2 for simultaneous eqs. (6), (1)

For the model of conics given by Eqs. (4)-(6) and the equation of the slope of the tangent given by eq. (7) computed at a point $P_0(r_0, \theta_0)$, it is required to calculate the points $P_1(r_1, \theta_1)$, and $P_2(r_2, \theta_2)$ on the conic as well as the circle given by eq. (1). For convenience, we substitute $\theta_i = \theta_0 + \Delta\theta_i, i = 1$ to 2, where $\Delta\theta_i, i = 1$ to 2 are the two solutions of the simultaneous equations (6) and (1). Accordingly, we solve equations (6) and (1) to find the points P_1 and P_2 as follows (while truncating the terms higher than the second order of $\Delta\theta$): For simplicity of expressions, we assign the following:

$$k = 1 - e \cos \theta; k_0 = 1 - e \cos \theta_0. \quad (24)$$

1. Substituting eqs. (5) and (6) in eq. (1), we get $k^{-2} + k_0^{-2} - 2 \cos \Delta\theta (k k_0)^{-1} = (R/ae)^2$.
2. Using Taylor series expansion [18] for $\cos \Delta\theta$: $\cos \Delta\theta = 1 - (\Delta\theta)^2/2 + O((\Delta\theta)^4)$, we get: $(k^{-1} - k_0^{-1})^2 + (\Delta\theta)^2 (k k_0)^{-1} = (R/ae)^2$.
3. Simplifying the above, we get $e^2 (\cos \theta - \cos \theta_0)^2 (k k_0)^{-2} + (\Delta\theta)^2 (k k_0)^{-1} = (R/ae)^2$.
4. Using Taylor series expansion for $\cos \theta$ in the numerator of the first term (left hand side):

$$\cos \theta = \cos \theta_0 - \sin \theta_0 (\Delta\theta) - \cos \theta_0 (\Delta\theta)^2/2 + O((\Delta\theta)^3), \quad (25)$$

$$\text{we write } (\Delta\theta)^2 \frac{(e^2 (\sin \theta_0 + \cos \theta_0 (\Delta\theta/2))^2 + k k_0)}{(k k_0)^2} = \left(\frac{R}{ae}\right)^2$$

5. Truncating the higher order terms of $\Delta\theta$: eqn. (2):

$$(\Delta\theta)^2(k_0 k_0)^{-2}(e^2 \sin^2 \theta_0 + k_0 k_0) = (R/ae)^2$$

$$\text{OR } (\Delta\theta)^2(e^2 \sin^2 \theta_0 + k_0 k_0) = (R/ae)^2(k_0 k_0)^2.$$

6. Using Taylor series expansion for $\cos \theta$ given in eqn. (25), we write:

$$\begin{aligned} (\Delta\theta)^2 \{e^2 \sin^2 \theta_0 + k_0(k_0 + e \sin \theta_0(\Delta\theta))\} &= \\ (R/ae)^2 \{k_0(k_0 + e \sin \theta_0(\Delta\theta))\}^2 & \end{aligned}$$

7. Substituting $B = e \sin \theta_0, C = R/ae$, we get $(\Delta\theta)^2(B^2 + k_0^2 + k_0 B \Delta\theta) = (k_0 C)^2(k_0 + B \Delta\theta)^2$.

8. Truncating the higher order terms of $\Delta\theta$: $(\Delta\theta)^2(B^2 + k_0^2) = (k_0 C)^2(k_0 + B \Delta\theta)^2$.

9. Simplifying the above equation and solving it: $\Delta\theta = k_0^2 C \left(-k_0 B C \pm \sqrt{B^2 + k_0^2}\right)^{-1}$.

10. Re-substituting for B and C :

$$\Delta\theta = \pm \frac{k_0^2(R/ae)}{\sqrt{(e \sin \theta_0)^2 + k_0^2}} \left(1 \mp \frac{(e \sin \theta_0)k_0(R/ae)}{\sqrt{(e \sin \theta_0)^2 + k_0^2}}\right)^{-1}. \quad (26)$$

11. It can be proven that $\left|(\sin \theta_0)k_0/\sqrt{(e \sin \theta_0)^2 + k_0^2}\right| < 1$. Thus, if $(R/a) < 1$, infinite geometric series expansion can be applied to get a converging series for $\Delta\theta$:

$$\begin{aligned} \Delta\theta &= \pm \frac{k_0^2(R/ae)}{\sqrt{(e \sin \theta_0)^2 + k_0^2}} \sum_{n=0}^{\infty} \left(\pm \frac{(e \sin \theta_0)k_0(R/ae)}{\sqrt{(e \sin \theta_0)^2 + k_0^2}}\right)^n, \\ &= \pm D \left(1 \pm dD + (dD)^2 \pm (dD)^3 \pm \dots\right). \quad (27) \end{aligned}$$

where D and d are given by eqs. (9) and (10) respectively. For convenience, we refer to the negative and positive solutions as $\Delta\theta_1$ and $\Delta\theta_2$ respectively, which is in accordance with Fig. 1(a). Accordingly, the angles $\Delta\theta_1$ and $\Delta\theta_2$ can be written as:

$$\begin{aligned} \Delta\theta_1 &= D(dD - 1) \sum_{n=0}^{\infty} (dD)^{2n}, \\ \Delta\theta_2 &= D(dD + 1) \sum_{n=0}^{\infty} (dD)^{2n}. \quad (28) \end{aligned}$$

We consider two special test cases to verify the validity of eq. (28).

Case 1: Circle $e = 0$: For this case, we know that the focus becomes the center of the circle and the focal parameter $a \rightarrow \infty$, the structure is rotationally symmetric, and the radius of the circle $\rho = ae$. Additionally, $\Delta\theta_1 = -\Delta\theta_2, \forall \theta_0$. Using eq. (27), $\Delta\theta \approx \pm(R/\rho)$. Hence, this case is verified.

Case 2: Symmetry along the x -axis, $\theta_0 = 0$ and $\theta_0 = \pi$: Since the considered conic equation (4) is symmetric along the x -axis, $\Delta\theta_1 = -\Delta\theta_2, \theta_0 = \{0, \pi\}$. Using eq. (28), $\Delta\theta|_{\theta_0=0} \approx \pm(1-e)(R/ae), \Delta\theta|_{\theta_0=\pi} \approx \pm(1+e)(R/ae)$. Hence, this case is also verified.

B. Computation of the slope of the tangent

Here, the derivation in Appendix A is used to derive the relationship between the estimated slope \tilde{m} and the analytical slope of the tangent m_0 given by eq. (7). Continuing from

$$\begin{aligned} \tilde{m} &= \frac{r_2 \sin \theta_2 - r_1 \sin \theta_1}{r_2 \cos \theta_2 - r_1 \cos \theta_1}, \\ &= \frac{\sin \theta_2 - \sin \theta_1 - e \sin(\theta_2 - \theta_1)}{\cos \theta_2 - \cos \theta_1}, \\ &= \frac{(e - \cos \theta_0)\alpha + 2 \sin \theta_0 \beta - 2e\gamma}{\alpha \sin \theta_0 + 2\beta \cos \theta_0}. \quad (29) \end{aligned}$$

$$\begin{aligned} \alpha &= (\sin \Delta\theta_1 - \sin \Delta\theta_2), \\ \beta &= (\sin^2(\Delta\theta_1/2) - \sin^2(\Delta\theta_2/2)), \\ \gamma &= (\sin \Delta\theta_1 \sin^2(\Delta\theta_2/2) - \sin \Delta\theta_2 \sin^2(\Delta\theta_1/2)). \quad (30) \end{aligned}$$

Now imposing the condition that $D_{max} \ll 1$ (see Appendix C), such that $\Delta\theta_i, i = 1$ to 2 are very small, $\Delta\theta_i \rightarrow 0$, eq. (29) can be simplified using eqs. (7) and (8) as follows:

$$\begin{aligned} \lim_{\Delta\theta_i \rightarrow 0} \tilde{m} &= \frac{(e - \cos \theta_0) + 0.5 \sin \theta_0 (\Delta\theta_1 + \Delta\theta_2) + 0.5e \Delta\theta_1 \Delta\theta_2}{\sin \theta_0 + 0.5 \cos \theta_0 (\Delta\theta_1 + \Delta\theta_2)} \\ &= \frac{1}{\left(1 + \cot \theta_0 D \sum_{n=1}^{\infty} (dD)^n\right)} \left\{ m_0 + D \sum_{n=1}^{\infty} (dD)^n + \right. \\ &\quad \left. 0.5e \csc \theta_0 D^2 \left((dD)^2 - 1\right) \sum_{n'=0}^{\infty} \sum_{n=0}^{\infty} (dD)^{n+n'} \right\}. \quad (31) \end{aligned}$$

Here, we have used $\lim_{t \rightarrow 0} \sin t = t$. It can be shown that if $D_{max} \ll 1$, then $|\cot \theta_0 dD^2| \ll 1$. Thus, by applying infinite geometric series expansion [18] and retaining terms up to $O(D^3)$, we get:

$$\tilde{m} \approx m_0 - 0.5e d D^3 \csc \theta_0. \quad (32)$$

Thus, \tilde{m} converges to m_0 , subject to the condition that $D_{max} \ll 1$. In the above expression, we need to pay additional attention to two special cases: $\theta_0 \in \{0, \pi\}$, where $\csc \theta_0$ is singular. However, noting that $d \csc \theta_0$ is not singular, there is no extra singularity other than the singularity of the actual slope m_0 . The angular error in the computation of the slope is given by:

$$\begin{aligned} \partial\phi &= \left| \tan^{-1}(m_0) - \tan^{-1}(\tilde{m}) \right| = \left| \tan^{-1} \left(\frac{m_0 - \tilde{m}}{1 + m_0 \tilde{m}} \right) \right| \\ &\approx \tan^{-1} \left| \frac{0.5e d D^3 \csc \theta_0}{1 + m_0^2} \right| \approx \left| \frac{0.5e d D^3 \csc \theta_0}{1 + m_0^2} \right|. \quad (33) \end{aligned}$$

Specifically, for circle, i.e. $e = 0$, we get $\partial\phi = 0$. Further the error in the computation of the tangent is bounded by $\left| \frac{0.5e d D^3 \csc \theta_0}{1 + m_0^2} \right|$ and can be considered of order $O(D^3)$.

In all the above analysis, we considered a conic with focus at the origin and the directrix at $x = -a$. Since an arbitrarily placed conic can be represented using the general equation for conic eq. (4) by applying suitable rotation and translation, the analysis is applicable to all the possible conics.

C. Maximum value of D

Here the maximum value of D is derived. Let

$$A = (1 - e \cos \theta_0)^2 (R/ae),$$

$$B = \left((e \sin \theta_0)^2 + (1 - e \cos \theta_0)^2 \right)^{0.5}.$$

TABLE I
THREE CASES IN INVESTIGATION OF $\max(D)$

Case 1: $A = 0$ and $(1 - e \cos \theta_0)B^3 \neq 0$.	This is not possible, since for $A = 0$, we need $(1 - e \cos \theta_0) = 0$, which violates the condition $(1 - e \cos \theta_0)B^3 \neq 0$.
Case 2: $\sin \theta_0 = 0$ and $(1 - e \cos \theta_0)B^3 \neq 0$	This is possible for $\theta_0 = 0$ or π . For $\theta_0 = 0$, we get $D = (1 - e)R/(ae)$ For $\theta_0 = \pi$: we get $D = (1 + e)R/(ae)$
Case 3: $(2B^2 - (1 - e \cos \theta_0)) = 0$ and $(1 - e \cos \theta_0)B^3 \neq 0$	Investigated in Appendix VI-C.

Thus using eq. (9), $D = A/B$. For finding the maximum value of D , we first impose $\partial D/\partial \theta_0 = 0$. The expression of $\partial D/\partial \theta_0$ is computed as:

$$\frac{\partial D}{\partial \theta} = \frac{1}{B} \left(\frac{\partial A}{\partial \theta} - \frac{A}{B} \frac{\partial B}{\partial \theta} \right), \quad (34)$$

where $\frac{\partial A}{\partial \theta} = 2(1 - e \cos \theta_0)(e \sin \theta_0) \left(\frac{R}{ae}\right) = 2A \frac{(e \sin \theta_0)}{(1 - e \cos \theta_0)}$ and $\frac{\partial B}{\partial \theta} = \left(\frac{1}{B}\right)(e \sin \theta_0)$. Substituting the above in eq. (34), we get $\frac{\partial D}{\partial \theta} = \frac{A(e \sin \theta_0)(2B^2 - (1 - e \cos \theta_0))}{(1 - e \cos \theta_0)B^3}$. Thus, $\partial D/\partial \theta_0 = 0$ can be satisfied in three different cases shown in Table I.

For investigating case 3, we consider the condition $(2B^2 - (1 - e \cos \theta_0)) = 0$ and substitute the expression of B in order to get $\theta_0 = \pm \cos^{-1}((2e^2 + 1)/3e)$. Using this expression in D in eq. (9), we get $D = 2(1 - e^2)(R/ae)(5 - e^2 - 4e^4)^{-1/2}$. Evidently, in case 2 as well as case 3, $\theta_0 = \pi$ is the point of maxima for D .

REFERENCES

- [1] D. K. Prasad and M. K. H. Leung, "Error analysis of geometric ellipse detection methods due to quantization," in *Fourth Pacific-Rim Symposium on Image and Video Technology (PSIVT 2010)*, 2010, pp. 58–63.
- [2] I. M. Anderson and J. C. Bezdek, "Curvature and tangential deflection of discrete arcs: a theory based on the commutator of scatter matrix pairs and its application to vertex detection in planar shape data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 27–40, 1984.
- [3] D. Coeurjolly and R. Klette, "A comparative evaluation of length estimators of digital curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 252–258, 2004.
- [4] M. Worring and A. W. M. Smeulders, "Digital curvature estimation," *Computer Vision and Image Understanding*, vol. 58, no. 3, pp. 366–382, 1993.
- [5] H. K. Yuen, J. Illingworth, and J. Kittler, "Detecting partially occluded ellipses using the hough transform," *Image and Vision Computing*, vol. 7, no. 1, pp. 31–37, 1989.
- [6] A. Faure, L. Buzer, and F. Feschet, "Tangential cover for thick digital curves," *Pattern Recognition*, vol. 42, no. 10, pp. 2279–2287, 2009.
- [7] D. M. Tsai and M. F. Chen, "Curve fitting approach for tangent angle and curvature measurements," *Pattern Recognition*, vol. 27, no. 5, pp. 699–711, 1994.
- [8] F. De Vieilleville and J. O. Lachaud, "Comparison and improvement of tangent estimators on digital curves," *Pattern Recognition*, vol. 42, no. 8, pp. 1693–1707, 2009.
- [9] D. K. Prasad, R. K. Gupta, and M. K. H. Leung, *An Error Bounded Tangent Estimator for Digitized Elliptic Curves*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6607, pp. 272–283.
- [10] T. Lewiner, J. D. Gomes Jr, H. Lopes, and M. Craizer, "Curvature and torsion estimators based on parametric curve fitting," *Computers and Graphics*, vol. 29, no. 5, pp. 641–655, 2005.

- [11] F. Cazals and M. Pouget, "Estimating differential quantities using polynomial fitting of osculating jets," *Computer Aided Geometric Design*, vol. 22, no. 2, pp. 121–146, 2005.
- [12] R. A. McLaughlin and M. D. Alder, "The hough transform versus the upwrite," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 396–400, 1998.
- [13] D. K. Prasad, M. K. H. Leung, and S. Y. Cho, "Edge curvature and convexity based ellipse detection method," *Pattern Recognition*, vol. 45, no. 9, pp. 3204–3221, 2012.
- [14] F. Mokhtarian and A. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 1, pp. 34–43, 1986.
- [15] J. Matas, Z. Shao, and J. Kittler, "Estimation of curvature and tangent direction by median filtered differencing," *Lecture Notes in Computer Science*, vol. 974, pp. 83–88, 1995.
- [16] B. Keratret and J. O. Lachaud, *Robust estimation of curvature along digital contours with global optimization*. Berlin, Heidelberg: Springer Verlag, 2008, vol. 4992, pp. 334–345.
- [17] E. Kim, M. Haseyama, and H. Kitajima, "Fast and robust ellipse extraction from complicated images," in *Proceedings of the International Conference on Information Technology and Applications*, 2002, pp. 357–362.
- [18] E. W. Weisstein, *CRC concise encyclopedia of mathematics*. Florida: CRC Press, 2005, vol. 2.
- [19] D. K. Prasad, C. Quek, M. K. H. Leung, and S. Y. Cho, "A parameter independent line fitting method," in *Asian Conference on Pattern Recognition (ACPR)*, 2011, pp. 441–445.
- [20] J. O. Lachaud, A. Vialard, and F. De Vieilleville, "Analysis and comparative evaluation of discrete tangent estimators," ser. 12th International Conference on Discrete Geometry for Computer Imagery, DGCI 2005, E. Andres, G. Damiand, and P. Lienhardt, Eds., vol. 3429, 2005, pp. 240–251.
- [21] J. O. Lachaud, A. Vialard, and F. de Vieilleville, "Fast, accurate and convergent tangent estimation on digital contours," *Image and Vision Computing*, vol. 25, no. 10, pp. 1572–1587, 2007.
- [22] D. K. Prasad, "Survey of the problem of object detection in real images," *International Journal of Image Processing (IJIP)*, vol. 6, no. 6, p. 441, 2012.



Dilip K. Prasad received PhD and B.Tech degree in Computer Science and Engineering from Nanyang Technological University, Singapore and Indian School of Mines, India in 2012 and 2003 respectively. Currently, he is a research fellow at National University of Singapore, Singapore. He has published over 40 internationally peer-reviewed research articles. His current research interests include image processing, pattern recognition and computer vision.



Maylor K. H. Leung received a BSc degree in physics from the National Taiwan University in 1979, and BSc, MSc and PhD degrees in computer science from the University of Saskatchewan, Canada, in 1983, 1985 and 1992, respectively. He is a Professor with the FICT, Universiti Tunku Abdul Rahman (Kampar), Malaysia. His research interest is in the area of Computer Vision, Pattern Recognition and Image Processing.



Chai Quek is currently an associate professor in the School of Computer Engineering, Nanyang Technological University, Singapore. He received his BSc and PhD degrees from Heriot-Watt University (Edinburgh). His research interests include Neurocognitive informatics and Computational Finance. He has published over 200 international conference and journal papers.



Michael S. Brown obtained his BSc and PhD in Computer Science from the University of Kentucky in 1995 and 2001. He is currently an associate professor in the School of Computing at the National University of Singapore. He has served as an area chair for CVPR'09,11,13, ICCV'11,13, ECCV'12 and is an associate editor for IEEE TPAMI, IJCV and CGF. His research interests include computer vision, image processing and computer graphics.