International Journal of Hybrid Information Technology
Vol.8, No.10 (2015), pp.217-226
http://dx.doi.org/10.14257/ijhit.2015.8.10.20

An Implementation-Friendly Algorithm for Rendering Tangential Frame of Digital Curves

Liang Jia[1] and Nigang Sun[2]

[1 2] *School of Information Science & Engineering, Chang Zhou University, China*
[1] *E-mail:* sanctifier.jia@aliyun.com [2] *E-mail: ngsun@cczu.edu.cn*

Abstract

Plentiful algorithms for identifying digital straight segment (DSS) are presented in literatures, but few mention how to indicate detected results in a visual-pleasure presentation. This paper proposed an algorithm for rendering tangential frames, i.e., edges of tangential covers overlapping detected DSS. The algorithm adapts strategies as linear regression and coordinate rotation to estimate points of frame. The implementation based on these strategies is surprisingly intuitive. The visual results of algorithm are shown and computational costs with respect to data of different dimensions are analyzed. The experimental results show proposed algorithm possesses a nearly linear time and its renderings are correct and visually-attractive.

Keywords: *Tangential Frame, Tangential cover, Digital straight segment, Digital image processing*

1. Introduction

As a cornerstone of analyzing digital shapes, DSS is employed to approximate curves to any desired accuracy in ideal case [1]. However, the rigorous definition of DSS requires connectivity and one-pixel width of digital curves which cannot be guaranteed in real-life images. To eliminate prerequisites of DSS definition, generalized versions of DSS are derived, *e.g.*, blurred segment [2] and α-thick [3]. These concepts well adapt situations like digital curve points scatter in a region. Since there are vast scenarios involving concepts derived based on DSS, *e.g.*, crack detection in materials [4], robot-navigation [5] and fingerprint recognition [6], numerous algorithms for identifying these versions of DSS are presented in literatures [7-9], but few described how to visually represent identified results in a delicate manner. Results are usually presented by one-pixel-width line segments of enlarged endpoints [10], colorful line segments [11] or solid blocks [12], each suffers its own limitations and these barely meet requirements of DSS indication in real-life images. Namely, the regions and directions of DSS indicated in foreground should be easily discernable for a human observer and background should not be blocked or largely changed by indications. This paper proposes an algorithm for rendering edges of tangential cover of given digital curves or segments. Proposed algorithm employs basic means as linear regression and coordinate system rotation to accomplish rendering. Consequently, it can be implemented easily and rapidly and indicated DSS meet criterions mentioned above.

2. Related Works

There are various indicative means to represent identified DSS in digital images. According to data sources, these means may be categorized to two classes, *i.e.*, indications limited to one-pixel width digital segment or not. Figure 1 depicts three typical indications. Left [10] and middle [11] ones in Figure 1 both are restricted to

ISSN: 1738-9968 IJHIT

one-pixel width, and right one [12] is suitable for data source of varying width. Although indications as right one in Figure 1 eliminate one-pixel width limitation, its drawback is obvious, *i.e.*, background pixels mingled with foreground are forcedly brushed by two monotonous foreground colors. This may not cause noticeable problems in ideal images as shown in right of Figure 1, but this can lead to observation difficulties when multiple DSS intersect at some point, especially when their lengths are small. Another obstacle caused by this drawback is information about background pixels is completely lost. Without alternatively checking original image, observer has no clue about details of regions overlapped by solid colors in resulting image. To overcome this drawback and inherit compatibility of varying-length DSS, definitions associated with indications like the one presented in [12] are investigated in this section.

Figure 1. Various DSS Indications

Actually, the narrow blocks depicted in Figure 1 right are formally named tangential cover in [3]. Tangential cover is a visual representation of generalized versions of DSS in some degrees, and its definition is essentially based on these versions. Hence, some definitions tightly bounded with tangential cover are introduced here. DSS definition varies for different modeling of digitalization. For a straight line in Euclidean space, if each pixel of its digital version has the nearest integral coordinates to real point, then DSS is defined as followings [1].

Definition 1. If digital points $p = (i_p, j_p)$ and $q = (i_q, j_q)$ where $i_p, i_q, j_p, j_q \in \mathbb{Z}$ and $i_p < i_q$, slope of pq lies between 0 and 1, then for each vertical line $\mathrm{x} = i \in \mathbb{Z}$ where $i_p \leq i \leq i_q$, there is a unique digital point in DSS whose coordinate is (i, j) corresponding to real point interested by $\mathrm{x} = i$ and line pq, and j is rounded off from y coordinate of real point.

$$\mathrm{DSS}(p,q) = \left\{(i,j) \in \mathbb{Z}^2 | i_p \leq i \leq i_q, j = round(y), \frac{y - j_q}{j_q - j_p} = \frac{i - i_q}{i_q - i_p}\right\}$$

Definition 1 requires slope of line joining any two pixels equals slope of real line. This implies connectivity and one pixel width of underlying digital segment. Blurred segment whose definition is given below resolves this limitation [2].

Definition 2. A set $\mathcal{D}(a, b, \mu, \omega)$ of points (x, y) satisfies inequalities $\mu \leq ax - by \leq \mu + \omega$ where $x, y, a, b, \mu, \omega \in \mathbb{Z}$, $b \neq 0$ and $\gcd(a, b) = 1$. Lines $ax - by = \mu$ and $ax - by = \mu + \omega$ are named leaning lines.

$$\mathcal{D}(a,b,\mu,\omega) = \{(x,y) \in \mathbb{Z}^2 | \mu \leq ax - by \leq \mu + \omega, a, b, \mu, \omega \in \mathbb{Z}, b \neq 0, \gcd(a,b) = 1\}$$

Obviously, Definition 1 is inappropriate for real-life images because segments in such images are always fragmental and wider than 1 pixel. Definition 2 defines a strip bounded by two leaning lines. The width of $\mathcal{D}(a, b, \mu, \omega)$ is indirectly reflected by parameter ω named arithmetical thickness. α-thick segment replaces ω by term $\alpha = sup\{|a|, |b|\}$ with respect to a digital line approximating ideal DSS and roughly centered between two leaning lines

3. Tangential Cover and Its Frame

Unlike definitions reviewed in section 2, parameters are given directly by procedure identifying digital segment, we employ linear regression to estimate ideal direction and coordinate rotation to find intercepts of leaning lines. Suppose a set of pixels has been segmented from digital image, namely, a set $\mathcal{D} = \{P_1, P_2, \dots P_n\}$ where P_i has coordinates $(x_i, y_i) \in \mathbb{Z}^2$ and $1 \le i \le n$, then its direction can be estimated by formula given below based on linear regression.

$$\theta = \tan^{-1}\left(\frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}\right) \quad \text{where} \quad \overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}, \overline{y} = \frac{\sum_{i=1}^{n} y_i}{n} \qquad (1)$$

Point of coordinates $(\overline{x}, \overline{y})$ is named *mean center* of $\mathcal{D}$. If we rotate coordinate system associated with $\mathcal{D}$ in a manner as rotating x-axis anticlockwise by θ and move origin to mean center, then points of maximal or minimal values of converted coordinates in $\mathcal{D}$ are named *marginal points* for $\mathcal{D}$. Let $x'_{lmax}, x'_{lmin}, y'_{wmax}, y'_{wmin}$ denote corresponding maximal or minimal coordinates, then lines $x = x'_{lmax}$, $x = x'_{lmin}$, $y = y'_{wmax}$ and $y = y'_{wmin}$ are *leaning lines* for $\mathcal{D}$ in converted coordinate system. Figure1 depicts a digital segment in two coordinate systems and indicates direction θ given by (1); apparently, distances between paralleled leaning lines are geometrical width or length of segment.

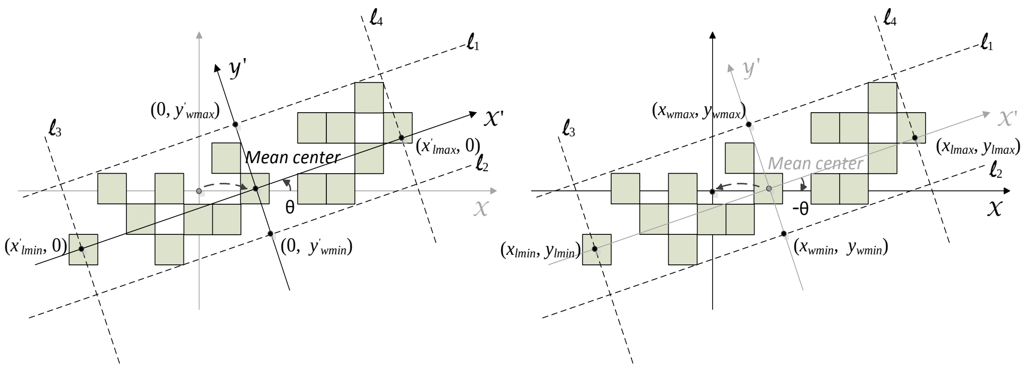

Figure 2. Leaning Lines and Marginal Points

Let (x', y') denotes the converted coordinates of (x, y) with respect to a system whose origin is (C_x, C_y) and direction of rotated positive x-axis directs differs from original positive x-axis by θ. Then, formulae for rotated and transferred coordinates are given as followings.

$$\begin{cases} x' = (x - C_x)\cos\theta + (y - C_y)\sin\theta \\ y' = -(x - C_x)\sin\theta + (y - C_y)\cos\theta \end{cases} \qquad (2)$$

The conversion can be cancelled by moving converted coordinate system back to former origin and rotating positive x-axis clockwise by θ. This requires converted coordinates of former origin and which are given by $\left(-C_x\cos\theta - C_y\sin\theta, C_x\sin\theta - C_y\cos\theta\right) = (O'_x, O'_y)$. If (C_x, C_y) and θ in (1) are substituted by (O'_x, O'_y) and $-\theta$, then we obtain the formulae to compute original coordinates of converted ones.

$$\begin{cases} x = (x' - O'_x)\cos\theta - (y' - O'_y)\sin\theta \\ y = (x' - O'_x)\sin\theta + (y' - O'_y)\cos\theta \end{cases} \qquad (3)$$

Hence, marginal points of coordinates $(x'_{lmax}, 0)$, $(x'_{lmin}, 0)$, $(0, y'_{wmax})$ and $(0, y'_{wmin})$ found in converted system can be mapped back to original system through (3). Let (x_{lmax}, y_{lmax}), (x_{lmin}, y_{lmin}), (x_{wmax}, y_{wmax}) and (x_{wmin}, y_{wmin}) denote corresponding original coordinates of marginal points, then

$$(x_{lmax}, y_{lmax}) = \left((x'_{lmax} - O'_x)\cos\theta + O'_y\sin\theta, (x'_{lmax} - O'_x)\sin\theta - O'_y\cos\theta\right) \text{ and}$$

$$(x_{wmax}, y_{wmax}) = \left(-O'_x \cos\theta - \left(y'_{wmax} - O'_y\right)\sin\theta, -O'_x \sin\theta + \cos\theta\left(y'_{wmax} - O'_y\right)\right).$$

The other two differ only by coordinate subscripts. In Figure1, marginal points are depicted by their coordinates with respect enabled coordinate system shown by solid lines. Hence, leaning lines $x = x'_{lmax}$, $x = x'_{lmin}$, $y = y'_{wmax}$ and $y = y'_{wmin}$ in original system are given by followings.

$$\begin{cases}(y_{lmax} - y_{lmin})x - (x_{lmax} - x_{lmin})y = (y_{lmax} - y_{lmin})x_{wmax} - (x_{lmax} - x_{lmin})y_{wmax} \\ (y_{lmax} - y_{lmin})x - (x_{lmax} - x_{lmin})y = (y_{lmax} - y_{lmin})x_{wmin} - (x_{lmax} - x_{lmin})y_{wmin}\end{cases} \quad (4)$$

And

$$\begin{cases}(y_{wmax} - y_{wmin})x - (x_{wmax} - x_{wmin})y = (y_{wmax} - y_{wmin})x_{lmax} - (x_{wmax} - x_{lmin})y_{lmax} \\ (y_{wmax} - y_{wmin})x - (x_{wmax} - x_{wmin})y = (y_{wmax} - y_{wmin})x_{lmin} - (x_{wmax} - x_{lmin})y_{lmin}\end{cases} \quad (5)$$

Lines given by (4) and (5) are shown in Figure 2 as dotted lines labeled by ℓ_1, ℓ_2, ℓ_3 and ℓ_4. Finally, assemble formulae presented in this section, definition of tangential cover is given below.

Definition 3 For a given set $\mathcal{D} = \{P_1, P_2, \dots P_n\}$ where P_i has coordinates $(x_i, y_i) \in \mathbb{Z}^2$ and $1 \le i \le n$, there are four marginal points $(x'_{lmax}, 0)$, $(x'_{lmin}, 0)$, $(0, y'_{wmax})$ and $(0, y'_{wmin})$ whose coordinates are estimated in a coordinate system obtained by transferring origin to mean center of $\mathcal{D}$ and rotating positive x-axis anticlockwise by θ given by (1). The counterparts of four points in original system can be computed by using (3) and denoted by (x_{lmax}, y_{lmax}), (x_{lmin}, y_{lmin}), (x_{wmax}, y_{wmax}) and (x_{wmin}, y_{wmin}). Suppose $0 < \theta < \pi/2$, then tangential cover $\mathcal{T}$ about $\mathcal{D}$ is given by following formula.

$$\mathcal{T}(\mathcal{D}) = \left\{(x, y) \in \mathbb{Z}^2 \,\middle|\, \begin{matrix} ax_{wmax} - by_{wmax} \le ax - by \le ax_{wmin} - by_{wmin} \\ a'x_{lmin} - b'y_{lmin} \le a'x - b'y \le a'x_{lmax} - b'y_{lmax}\end{matrix}\right\}$$

Where $a = y_{lmax} - y_{lmin}$, $b = x_{lmax} - x_{lmin}$, $a' = y_{wmax} - y_{wmin}$ and $b' = x_{wmax} - x_{wmin}$. Let ℓ_1, ℓ_2 denote leaning lines given by (4) and ℓ_3, ℓ_4 represent leaning lines given by (5). Let $\mathrm{dist}(P, \ell)$ denote Euclidean distance between a point P and a straight line ℓ. Then a more general definition is given below regardless of range of θ.

$$\mathcal{T}(\mathcal{D}) = \left\{P = (x, y) \in \mathbb{Z}^2 \,\middle|\, \begin{matrix} \mathrm{dist}(P, \ell_1) + \mathrm{dist}(P, \ell_2) \le |x'_{lmax}| + |x'_{lmin}| \\ \mathrm{dist}(P, \ell_3) + \mathrm{dist}(P, \ell_4) \le |y'_{wmax}| + |y'_{wmin}|\end{matrix}\right\}$$

Essentially, $\mathcal{T}(\mathcal{D})$ is a rectangle overlapping $\mathcal{D}$. Bounders of $\mathcal{T}(\mathcal{D})$ are leaning lines given by (4) and (5) intersected by marginal points defined by Definition 3. To avoid completely overlapping $\mathcal{D}$ and provide a visual feedback about $\mathcal{T}(\mathcal{D})$, we actually need to render frame of tangential cover and leave its inner area untouched. Let $d_i = \mathrm{dist}(P, \ell_i)$ where $1 \le i \le 4$, then definition of tangential frame $\mathcal{F}_{\mathcal{T}}$ is given below.

Definition 4.

$$\mathcal{F}_{\mathcal{T}}(\mathcal{D}) = \left\{(x, y) \in \mathbb{Z}^2 \,\middle|\, \begin{matrix} d_1 + d_2 = |x'_{lmax}| + |x'_{lmin}|, d_1 \cdot d_2 = 0 \\ d_3 + d_4 = |y'_{wmax}| + |y'_{wmin}|, d_3 \cdot d_4 = 0\end{matrix}\right\}$$

4. Algorithm Design

According to Definition 4, rendering tangential frame is equivalent to render leaning lines segmented by marginal points. Hence, points of collection should be represented in coordinate system whose center is mean center and positive x-axis directs to direction of collection. In this converted system, marginal points can be easily found. Because leaning lines in converted system also have formulae much simpler than (4) and (5) of original system, points of leaning line segments can also be easily computed. Finally, coordinates of these points can be converted back to original system based on (3) and deviations of pointers can be computed by using these coordinates. The resulting deviations are output for actual rendering program. This algorithm is visually represented by UML class and activity diagrams for describing its static structure and dynamic behaviors. The class diagram is shown in Figure 3.

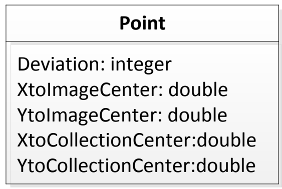

Figure 3. Class Diagram

Proposed algorithm mainly depends on Point class shown in Figure 3 to preserve computational results like coordinate values of different coordinate systems. The property named *Deviation* is employed to store the pointer deviation with respect to initial address of image data. The general activity diagram is shown in Figure 4. There are eight logical blocks represented by subroutines in the diagram and each encapsulates a main step of algorithm. Subroutines providing core functionalities will be described in details in following sections, the other will be briefly discussed.

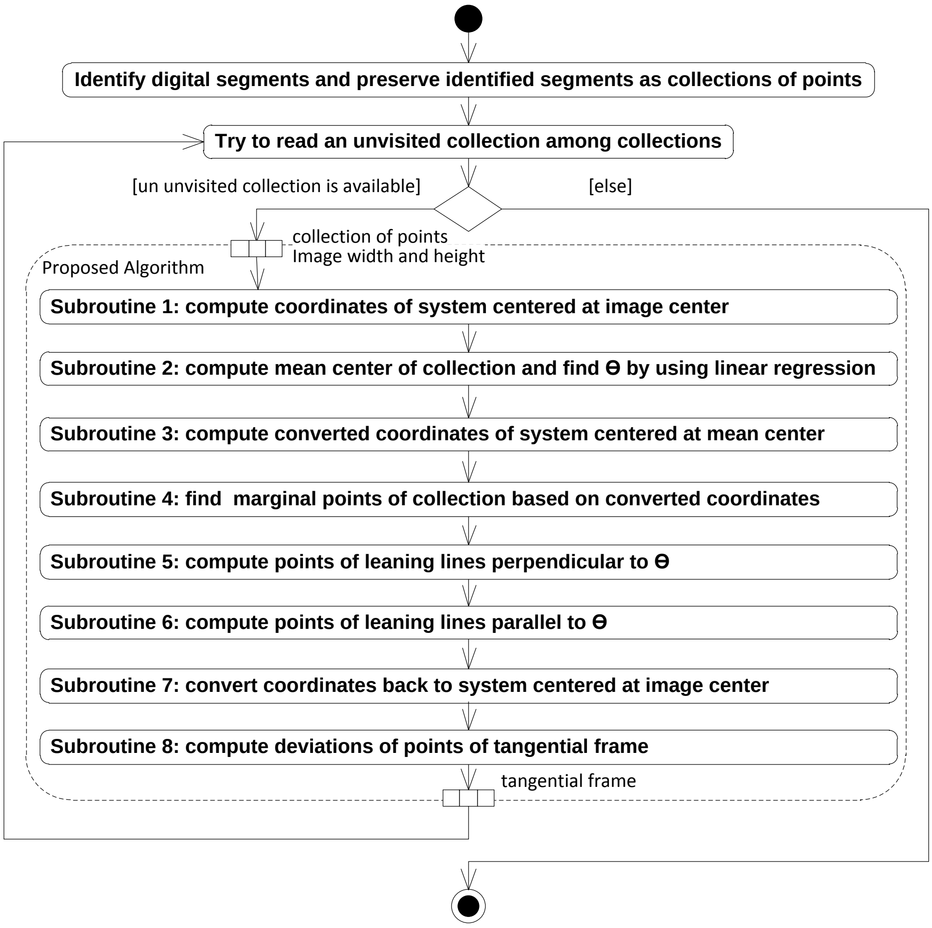

Figure 4. General Diagram

4.1. Estimate Direction of Collection by Using Linear Regression

To enable coordinate system transfer and rotation, coordinates of each point contained by collection given as input should be computed with respect of a chosen pixel in image. In our algorithm, image center is selected for origin of system and coordinates are computed based on following formulae.

$$\begin{cases} x = d - width \cdot \left\lfloor \frac{d}{width} \right\rfloor - \left\lfloor \frac{width}{2} \right\rfloor \\ y = \left\lfloor \frac{height}{2} \right\rfloor - \left\lfloor \frac{d}{width} \right\rfloor \end{cases}$$

Coordinate computation is performed in Subroutine 1. Details of Subroutine 2 are depicted in Figure 5. Subroutine 2 first computes mean center of collection, then computes terms $(x_i - \overline{x})$ and $(y_i - \overline{y})$ involved in (1) and accumulatively preserves resulting values of each computation in variables named *numerator* and *denominator*. Finally, the ratio of *numerator* and *denominator* is passed to inverse trigonometric function of tangent for estimating collection direction.

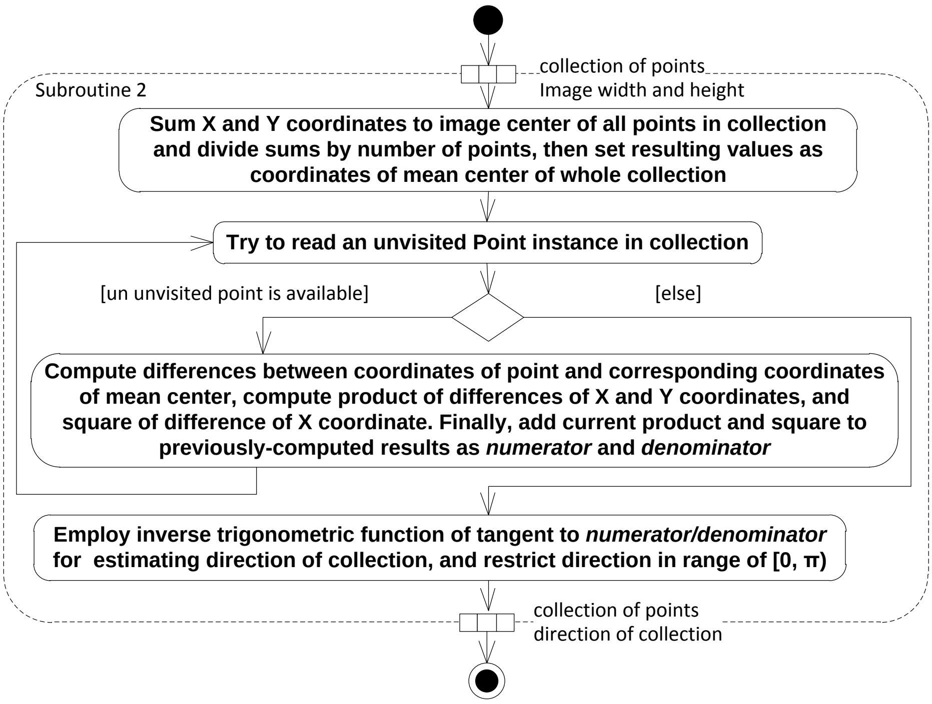

Figure 5. Subroutine 2

4.2. Render Tangential Frame in Converted Coordinate System

Once mean center and direction of collection are estimated by Subroutine 2, rotation and transfer of coordinate are performed by Subroutine 3 based on formulae given by (2). Resulting coordinates are preserved in properties XtoCollectionCenter and YtoCollectionCenter of each point in collection. Then Subroutine 4 finds maximal and minimal values of abscissae and ordinates among points in collection. Found coordinates are employed to generate marginal points shown in Figure 2 left.

Subroutine 5 and Subroutine 6 generate Point instances of structures shown in Figure 3, and these points form edges of tangential frame. Figure 6 and Figure 7 depict details of Subroutine 5 and 6 respectively. Computations involving in Subroutine 5 and 6 are based on formulae of leaning lines in converted coordinate system shown in Figure 2 left. Namely, two subroutines render tangential frame according to formulae of leaning lines, i.e., $x = x'_{lmax}$, $x = x'_{lmin}$, $y = y'_{wmax}$ and $y = y'_{wmin}$ in converted system. Consequently, implementations shown in Figure 6 and 6 are quite simple.

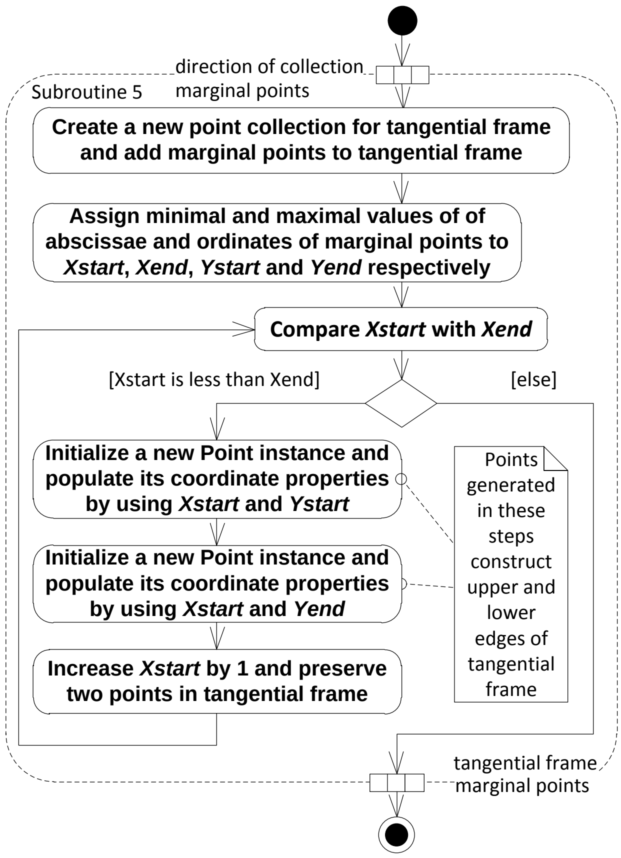

Figure 6. Subroutine 5

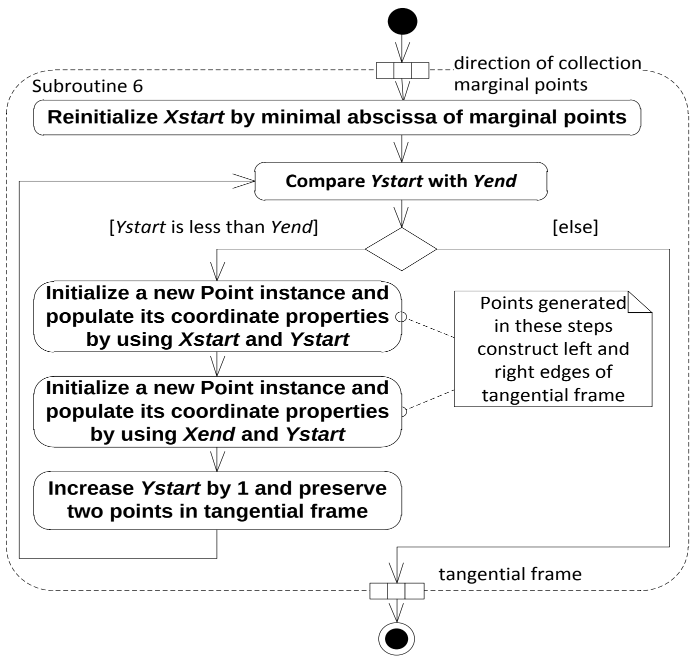

Figure 7. Subroutine 6

5. Experimental Results and Discussion

The algorithm is implemented by using C# in Visual Studio 2010. Digital segments, i.e., collections of points as input of proposed algorithm, are obtained by using a segmentation algorithm developed by authors. These collections are then input to proposed algorithm for rendering tangential frames. Four digital images and their resolution-varying versions are tested for proposed tangential frame rendering algorithm. Test results are represented by data of two categories, i.e., visual results of rendered tangential frames and time consumed by rendering. Section 5.1 introduces visual results and Section 5.2 describes analyses of consumed time.

5.1. Visual Results of Tangential Frame Rendering

Figure 8 to Figure 10 depict rendering results for four digital images of resolutions 128-by-128, 256-by-256 and 512-by-512. Tested images are named cameraman, house, lena and puzzle. Rendering results shown in Figure 8 to Figure 10 are listed in this order from left to right. In these applications, edges of tangential frames are rendered by three lines in colors of black, white, black instead of single lines. From results in these Figures, all tangential frames are correctly rendered within the regions of images

Figure 8. Tangential Frames Rendered for Images of Resolution 128×128

Figure 9. Tangential Frames Rendered for Images of Resolution 256×256

Figure 10. Tangential Frames Rendered for Images of Resolution 512×512

5.2. Render Tangential Frame in Converted Coordinate System

Figure 11 to Figure 13 depict time consumed by subroutines in different cases. Time consumed by each subroutine is represented by a colored stem and stems of all subroutines associated with a specific case are stacked in a bar in each Figure. The inputs of algorithm are same as images mentioned in Section 5.1.

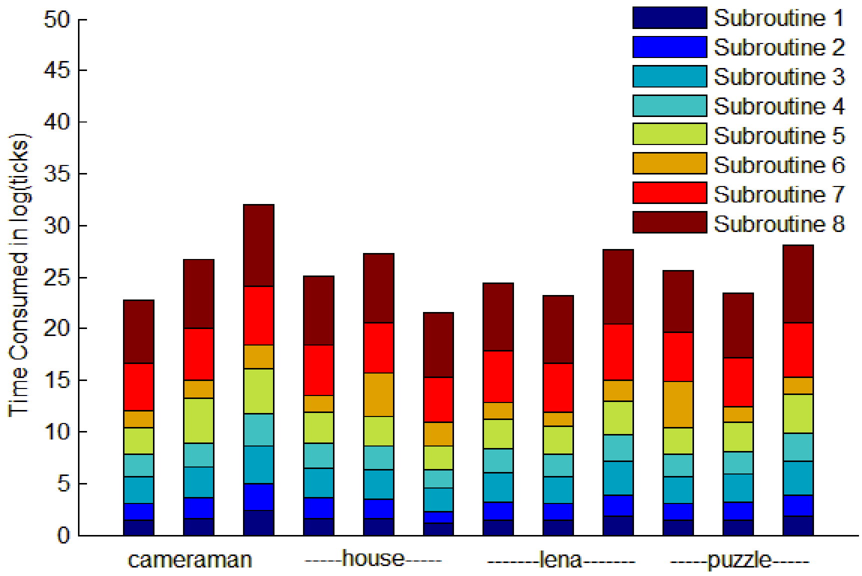

Figure 11. Averaged Time Consumed by Rendering Frames

Figure 11 shows averaged time of all collections with respect to images of different resolutions, *e.g.*, there are three bars with respect to image cameraman of resolutions 128-by-128, 256-by-256 and 512-by-512, and all bars are arranged in same order with respect to resolutions of images indicated by image names below the bars. From Figure 11, the most costly parts are Subroutine 7 and 8. Surprisingly, although implementations of Subroutine 5 and 6 are nearly same, their costs are quite different. This is mainly because Subroutine 5 consists of tangential frame creation. Although Subroutine 7 is inversion of Subroutine 3, their costs are different. Cost of Subroutine 7 is almost two times of Subroutine 3.

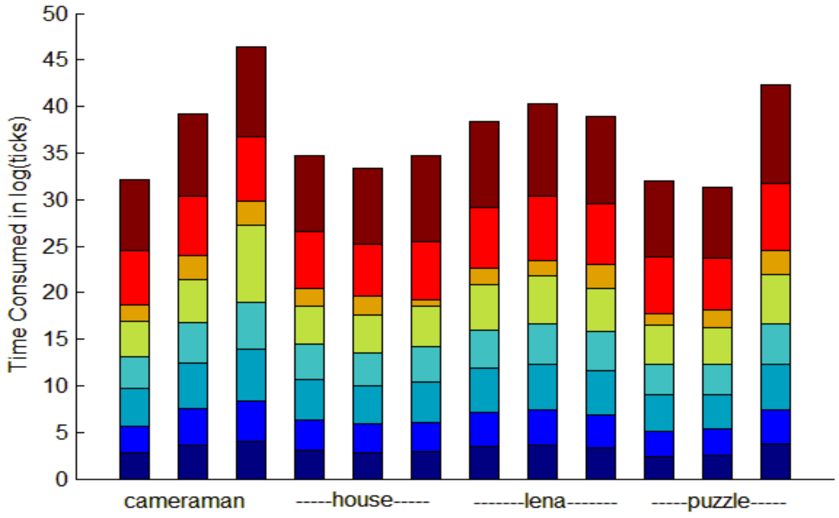

Figure 12. Time Consumed by Rendering the Largest Frames

Figure 12 and Figure 13 depict time of rendering largest and smallest frames. Figure 12 bascially resemble Figure 11, but Figure 13 is quite different from Figure 11, especially the 3th bar of image house and the 1st bar of image puzzle in Figure 13. These exceptional low costs should be ascribed to segement identification procedures. The identifications yield the exceptional small segments which finally lead to bars shown in Figure.

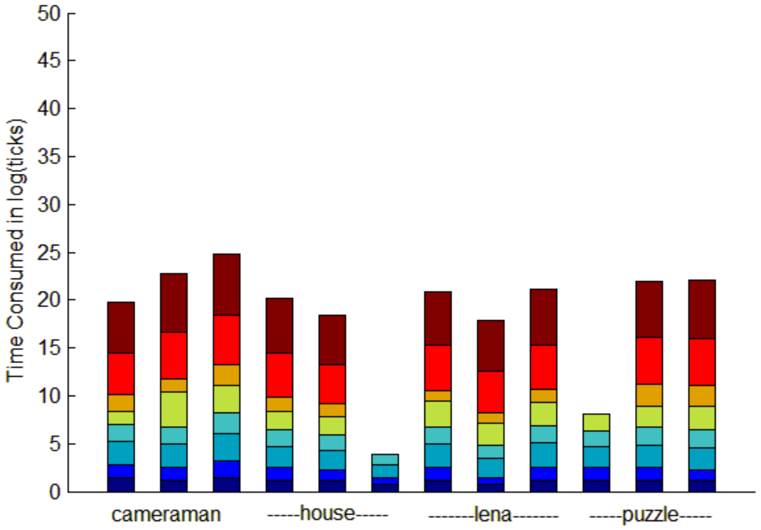

Figure 13. Time Consumed by Rendering the Smallest Frames

Figure 14 shows general time consumed by algorithm for images of different resolutions lying in range starting at 128-by-128 and ending at 512-by-512. Abscissa denotes one dimension of tested images and time is sampled every other 32 starting from 128. Ordinate denotes time consumed by algorithm with respect to different tested images. From Figure 14, the computational time of algorithm is nearly linear with some acceptable undulations occurred about 450. The only exception is case of image house which possesses a bell shape whose zenith is around 425. The decline at high resolution of image house may be caused by identification procedure which employs fixed parameters to generate point collections for rendering algorithm. The parameters accommodate low resolutions better than high ones, which can be inferred by inspecting rendering results (not shown).

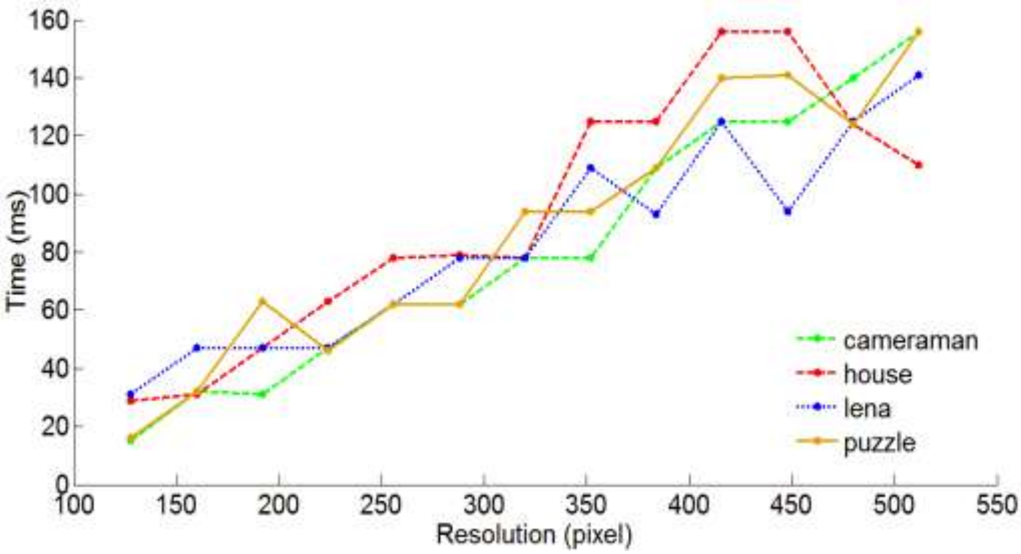

Figure 14. Time Consumed by Rendering Frames for Images of Different Resolutions

6. Visual Results of Tangential Frame Rendering

Wherever Times New Roman is specified, Times Roman, or Times may be used. If neither is available on your word processor, please use the font closest in appearance to Times New Roman that you have access to. Please avoid using bit-mapped fonts if possible. True-Type 1 fonts are preferred.

7. Main Text

Type your main text in 11-point Times New Roman, single-spaced with 13-point interline spacing. Do not use double-spacing. All paragraphs should be indented 1 pica (approximately 1/6- or 0.17-inch or 0.422 cm). Be sure your text is fully justified—that is, flush left and flush right. Please do not place any additional blank lines between paragraphs.

Figure and table captions should be 11-point Helvetica boldface (or a similar sans-serif font). Callouts should be 10-point Helvetica, non-boldface. Initially capitalize only the first word of each figure caption and table title. Figures and tables must be numbered separately. For example: “Figure 1. Database contexts”, “Table 1. Input data”. Figure captions are to be below the figures. Table titles are to be centered above the tables.

8. Conclusion

This paper provides an algorithm to render representation of given DSS for human observer. Tangential frame is defined for indicating given DSS in their containing images and related rendering algorithm is developed based on linear regression and coordinate rotation. The implementation is intuitive and can be easily completed. The experimental results exhibit visual renderings and diagrams of computational time. Visual results are apparently correct and computational costs are acceptable and nearly linear.

References

[1] J. Mukhopadhyay, P. Bhowmick, *et al.*, “Digital Geometry in Image Processing”, Boca Raton, FL: CRC Press, ch. 4, sec. 1, **(2013)**, pp. 130.
[2] I. Debled-Rennessona, R. Jean-Lucb and J. Rouyer-Deglic, "Segmentation of discrete curves into fuzzy segments", Electron. Notes in Discrete Math., vol. 12, **(2003)** March, pp. 372-383.
[3] A. Faurea, L. Buzerb and F. Fescheta, "Tangential cover for thick digital curves", Pattern Recognition, vol. 42, no. 10, **(2009)** October, pp. 2279-2287.
[4] S. Mahadevan and D. P. Casasent, "Detection of triple junction parameters in microscope images", Proc. SPIE, vol. 4387, **(2001)**, pp. 204-214.
[5] P. Kahn, L. Kitchen and E. M. Rieseman, "A fast line finder for vision-guided robot navigation", IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, no. 11, **(1990)**, pp. 1098–1102.
[6] X. Jiang, X. You, Y. Yuan and M. Gong, "A method using long digital straight segments for fingerprint recognition", Neurocomputing, vol. 77, no. 1, **(2012)** February, pp. 28-35.
[7] J. Lachaud and M. Said, “Two efficient algorithms for computing the characteristics of a subsegment of a digital straight line”, Discrete Applied Mathematics, vol. 161, no. 15, **(2013)** October, pp. 2293-2315.
[8] I. Sivignon, “Fast recognition of a Digital Straight Line subsegment: Two algorithms of logarithmic time complexity”, Discrete Applied Mathematics, in press.
[9] “Discrete Applied Mathematics”, In Press, Corrected Proof, Available online, **(2014)** June 23.
[10] J. S. D. Ouattara, E. A., G. Largeteau-Skapin, R. Zrour and T. M. Tapsoba, “Remainder approach for the computation of digital straight line subsegment characteristics”, Discrete Applied Mathematics, In press.
[11] P. Bhowmick and B. B. Bhattacharya, “Fast polygonal approximation of digital curves using relaxed straightness properties”, IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 9, **(2007)** September, pp. 1590-1602.
[12] L. Jia, N. Sun, “A line segment detection algorithm based on statistical analyses of quantified directions in digital image”, Comput. Modelling & New Technologies, vol. 18, no. 6, **(2014)**, pp. 79-88.
[13] L. Buzer, “A simple algorithm for digital line recognition in the general case”, Pattern Recognition, vol. 40, no. 6, **(2007)** June, pp. 1675-1684.

Authors

Liang Jia, He is a faculty member at Changzhou university of Jiangsu province in China. He received the MS degree in computer science from the Nanjing university of Science and Technology in 2009 and the BS degree in computer science from Beifang university of Nationalities in 2004. His research interests mainly include image processing, computer vision, service-oriented software development.

Nigang Sun, He received his Ph.D. degree in information security, University of Chinese Academy of Sciences, 2007. He joined East China University of Science and Technology in 2007 and Changzhou University in 2010. And now he is an associate professor in the department of the Computer Science and Technology. His research interests include formal methods, system modelling and analysis, information security.

