International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.8 (2016), pp.295-310
http://dx.doi.org/10.14257/ijsip.2016.9.8.26

Digital Contour Segmentation Based on Statistical Straightness

Liang Jia and Jiuzhen Liang*

School of Information Science & Engineering, Chang Zhou University, China
**jzliang@cczu.edu.cn*

Abstract

A novel algorithm for segmenting digital contours in edge images to digital straight segments (DSS) is proposed in this paper. While exploring contours, the algorithm maintains a dynamically-updated histogram of directions between every two adjacent contour pixels which are considered connected with respect to a generalized concept of connectedness. Directions of pixels are compared with the direction of the greatest value in histogram, and the ones of differences less than a given threshold are collected as the candidates of DSS. Finally, DSS is split and merged based on the analyses of the properties associated with the histogram such as balance and modality. Implementation of the proposed algorithm is visually introduced by using UML activity diagrams, and the experimental results of the artificial and real-life images are illustrated and discussed. Analyses of the experiments showed the validity of the segmentations and the bottleneck of the proposed algorithm.

Keywords: *Digital straight segment; Line segment; Feature extraction; Digital image processing*

1. Introduction

Digital straight segment is a fundamental geometrical descriptor for digital image processing. Due to its multifarious applications as fingerprint recognition [1], Shape classification [2], aerial stereo images analysis [3], geometric estimators [4], *etc.*, plentiful algorithms have been developed for tackling identifying DSS in digital images during past several decades. The essential issues of segmenting DSS from the given images mainly involve how to estimate directions and endpoints of DSS. Commonly, rectangular coordinate system is introduced for representing DSS analytically and directions are estimated by computing the slopes. The slope calculations are based on various means such as arithmetic approaches [5-7], convex hull [8-12] and dual-space [9-12]. According to DSS definitions, *i.e.*, whether DSS is assumed to be 4 or 8-connected, these algorithms may be coarsely classified to two categories: algorithms of tight connectedness [5-10] and loose connectedness [11-12].

Conventionally, DSL is defined as $\{(\mathrm{x},\mathrm{y}) \in \mathbb{Z}^2 | \mu \leq \mathrm{ax} - \mathrm{by} < \mu + |\mathrm{a}| + |\mathrm{b}|\}$ with the common assumption that $0 < a < b$ and $\gcd(\mathrm{a},\mathrm{b}) = 1$. DSS is a segment of DSL with respect to two endpoints. For a DSS, there are two leaning lines $\mathrm{ax} - \mathrm{by} = \mu$ and $\mathrm{ax} - \mathrm{by} = \mu + |\mathrm{a}| + |\mathrm{b}| - 1$which in combination determine the borders of DSS. Points of DSS on leaning lines are called leaning points. Weakly exterior points lie on lines $\mathrm{ax} - \mathrm{by} = \mu - 1$ and $\mathrm{ax} - \mathrm{by} = \mu + |\mathrm{a}| + |\mathrm{b}|$ and both lines are outside of DSS and next to its leaning lines.

For algorithms of tight connectedness, DSL is commonly assumed to be 8-connected [5-10] and DSS is assumed to be either 8-connected [7-10] or 4-connected [6]. One emblematic algorithm named DR95 [5] inspires many algorithm designs of tight

*Corresponding Author

ISSN: 2005-4254 IJSIP

connectedness including the recent work [6]. DR95 incrementally updates the slope of DSS based on arithmetical relationship between its leaning points and weakly exterior points found by traversing along a given 8-connected digital contour. Like DR95, algorithms proposed by Lachaud and Said [6], Ouattara *et. al.*, [7] and Debled-Rennesson *et. al.*, [9] also depend on arithmetic means but they focus on some special cases, *i.e.*, segmenting DSS of minimal parameters or characteristics with respect to the known endpoints of a digital straight line of known parameters. These additional conditions lead to elaborately-designed algorithms of high efficiency. Although, algorithms of tight connectedness are highly efficient, additional conditions and requirements like noiseless digital contour, their applications in practice are limited [8-12].

Oppositely, algorithms of loose connectedness take into account that noise, geometrical width or thickness, disconnectedness presented in segmentation. Commonly, connectedness is mainly reflected by thickness. Thickness of DSS in growing is dynamically measured by using convex hull and compared with a given threshold for controlling thickness [11-12]. Buzer [11] proposed a greedy algorithm which may lead to incorrect segmentations. This deficiency is remedied by Faure *et. al.*, [12] by using tangential cover which generates strip-like polygons based on segmented DSS. Because the tangential cover is originally designed for one-pixel-width contour, boundaries of thick contour have to be extracted and combined to a chain of boundary points. This extraction is error-prone and sometimes may be impracticable for real-life images.

This paper proposed an algorithm of loose connectedness to identify DSS statistically for given digital contours in edge images. The statistical approach is completely different from the conventional DSS segmentation algorithms as the ones discussed above. No additional information or preprocessing is required and no assumption of 4 or 8-connectedness is assumed. DSS growing is controlled by dynamical estimations of the direction and width by using statistically-designed measurement named statistical straightness and linear regression. The experiments yielded positive results for real-life images.

This paper is organized as follows. Section 2, introduces the schemas of classical algorithms and common concepts employed in the proposed algorithm. Section 3, mathematically describes the concept of statistical straightness and the relevant definitions for identifying DSS statistically. Section 4 visually illustrates the algorithm implementation by using UML activity diagrams [13], and experimental results are analyzed in Section 5. Section 6 draws the conclusions.

2. Related Works and Basic Concepts

This section introduces algorithms proposed by Lachaud and Said [6], Ouattara *et. al.*, [7], Debled-Rennesson *et. al.*, [8-9], Sivignon [10], Buzer [11] and Faure *et. al.*, [12] sequentially, and illustrates two common concepts about directions and their differences in digital images.

Said [6] and Ouattara *et. al.*, [7] both employed arithmetic approaches. Said [6] designed two algorithms named SmartDSS and ReverseSmartDSS based on the patterns calculated by using Berstel formulae. SmartDSS predicts weakly exterior points arithmetically and updates the slope by extending partial quotients of current DSS. According to the types of weakly exterior points, updating is determined by the number of patterns. ReverseSmartDSS additionally requires two upper leaning points closest to endpoints are given and starts by predicting an upper leaning point between two given points. It recursively estimates slopes associated with the greatest, smallest and reverse subpatterns until one of the estimated slope agrees with the deepest slope. The slopes estimated by two algorithms are represented by Stern–Brocot tree, a visual representation of partial quotients. The estimation order of slopes can be visualized as traveling between top and bottom of Stern–Brocot tree. Ouattara *et. al.*, [7] converted the estimation of minimal DSS to estimating coordinates of the two upper or lower leaning points based on the maximal or minimal remainders defined as $\mathcal{R}(\mathrm{x}) = (\mathrm{ax} - \mathrm{c}) \bmod \mathrm{b}$ for a point of abscissa x.

Debled-Rennesson *et. al.,* [8], Sivignon [10] and Buzer [11] all focused on the thick digital contours by using convex hull. Debled-Rennesson *et. al.,* [8] conceptualized the thickness by arithmetic property $(|\mathrm{a}| + |\mathrm{b}|)/\max(\mathrm{a}, \mathrm{b})$ called order and the segmentation is controlled by dynamically estimating the order of current DSS and the comparison with a pre-defined threshold. If the order exceeds threshold then the growing of current DSS stops and a new DSS starts. Although 8-connectedness is still required, the concept of dynamic thickness is first introduced. Sivignon [10] refined this algorithm through adopting convex hull to guarantee each segmented DSS is as long as possible and of minimized order with respect to the threshold. Buzer [11] developed an algorithm capable of segmenting disconnected DSS of thickness restricted by a threshold α. This algorithm is similar with the one designed by Debled-Rennesson *et. al.,* [9], *i.e.,* it also employs convex hull cooperated with the α-thickness defined as $\alpha \cdot \max(\mathrm{a}, \mathrm{b})$ to control the DSS thickness.

Debled-Rennesson *et. al.,* [9] and Faure *et. al.,* [12] both combined convex hull and dual space in their researches. Debled-Rennesson *et. al.,* [9] proposed two algorithms to estimate the minimal DSS. Both algorithms adapts the separating lines $\mathrm{ax} - \mathrm{by} + \rho = 0$ where $\mu \leq \rho < \mu + 1$ and $\rho \in \mathbb{R}$. The first computes the upper convex hull of points X in DSL and the lower convex hull of points $\mathrm{X} + (0,1)$. The minimal parameters are determined by convex hull edge of the maximal denominator associated with its slope. The second maps the separating lines $0 \leq \alpha\mathrm{x} - \mathrm{y} + \beta < 1$ to a dual space (α, β) and the minimal characteristics are represented by a point $\Lambda = (\mathrm{a}/\mathrm{b}, \mu/\mathrm{b})$ in Farey Fan, *i.e.*, a set of rays $\mathrm{R}(\mathrm{x}, \mathrm{y}) = \{(\mathrm{a}, \mathrm{b}) | \mathrm{b} = -\mathrm{ax} + \mathrm{y}\}$ where $0 \leq \mathrm{y} \leq \mathrm{x} \leq \mathrm{n}$, $0 \leq \alpha, \beta \leq 1$ and n is the length of DSS. In Farey Fan, the algorithm investigates the intersections between the highest ray through or below Λ, and the minimal characteristics are determined by two vertical lines of abscissae with the denominators smaller than n and closest to a/b. Faure *et. al.,* [12] employed tangential cover for mapping DSS to arcs defined in a dual space, and the arcs are adapted to generate polygons overlapping DSS. The inner and outer borders of thick digital contour are extracted and combined to a single chain based on both the border growing ratio and the visibility between points joined by chain. The chain is segmented to DSS's by using convex hull with respect to threshold α defined by Buzer [11]. Finally, DSS's are mapped through tangential cover and the polygons of the least vertices overlapping DSS are generated.

The basic and common concepts are introduced in the rest of this section. Let $\mathrm{I_F}$ denotes the foreground of an edge image. Digital contours are represented by connected pixels in $\mathrm{I_F}$. For any two pixels $\mathrm{P_{k1}}, \mathrm{P_{k2}} \in \mathrm{I_F}$, $\mathrm{k1}, \mathrm{k2} \in \mathbb{Z}^+$, $1 \leq \mathrm{k1}, \mathrm{k2} \leq |\mathrm{I_F}|$, $\mathrm{k1} \neq \mathrm{k2}$, vector $\overrightarrow{\mathrm{P_{k1}P_{k2}}}$ reflects not only the Euclidean distance between $\mathrm{P_{k1}}$ and $\mathrm{P_{k_2}}$ by its magnitude $|\overrightarrow{\mathrm{P_{k1}P_{k2}}}|$, but also an unique direction θ_{k2} whose definition is given below.

Definition 1. Direction $0 \leq \theta_{\mathrm{k2}} < 2\pi$ of vector $\overrightarrow{\mathrm{P_{k1}P_{k2}}}$ is the angle formed by anticlockwise rotating the positive x-axis (denoted by $\mathbb{O}$) originated at $\mathrm{P_{k1}}$ until overlapping $\overrightarrow{\mathrm{P_{k1}P_{k2}}}$. This process is formulated as $\theta_{\mathrm{k2}} = \mathrm{R_{ac}}(\mathbb{O}, \overrightarrow{\mathrm{P_{k1}P_{k2}}})$, *e.g.*, θ of $\overrightarrow{\mathrm{P_1P_2}}$ shown in Figure 1. Conversely, any given direction $0 \leq \theta < 2\pi$ is considered as the angle formed by continuously rotating $\mathbb{O}$ originated at arbitrary pixel $\mathrm{P_{k1}}$ until the radius scanned by $\mathbb{O}$ equals θ.

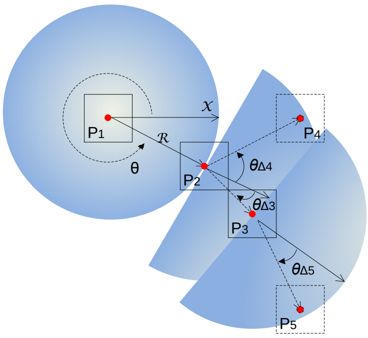


Figure 1. Proposed Algorithm Explores Foreground Pixels

There is no requirement about 4 or 8-connectedness for $\mathrm{P_{k1}}$ and $\mathrm{P_{k2}}$. The connectedness of $\mathrm{P_{k1}}$ and $\mathrm{P_{k2}}$ is determined by a given parameter $\mathcal{R}$, *i.e.*, $\mathrm{P_{k1}}$ and $\mathrm{P_{k2}}$ are connected iff $|\mathrm{P_{k1}} - \mathrm{P_{k2}}| \leq \mathcal{R}$. Hence, any pixel within the circle of radius $\mathcal{R}$ expanded at $\mathrm{P_{k1}}$ will be considered as connected to $\mathrm{P_{k1}}$. However, for a $\mathrm{P_{k1}}$comprised by a DSS, not every pixel $\mathrm{P_{k2}}$ connected with $\mathrm{P_{k1}}$ can become a member of the DSS. Their memberships are determined by differences between θ_{k2} and direction $\bar{\theta}$ of DSS which is statistically estimated and dynamically updated.

For two given direction θ_1 and θ_2, $0 \leq \theta_1, \theta_2 < 2\pi$, set their initial points of vectors coincide, then the difference $\theta_{\Delta 2}$ between θ_1 and θ_2 is the smaller angle formed by clockwise or anticlockwise rotating vector of θ_1 until overlapping vector of θ_2. Formally, let $\mathrm{R_c}(\theta_1, \theta_2)$, $\mathrm{R_{ac}}(\theta_1, \theta_2)$ respectively denotes the angles formed by clockwise and anticlockwise rotating vector of θ_1, and $\mathrm{sgn}(\mathrm{a} - \mathrm{b})$ is 1 when $\mathrm{a} \geq \mathrm{b}$ and -1 when $\mathrm{a} < b$, then definition of direction difference is given below.

Definition 2. For any two direction θ_{n1} and θ_{n2}, $0 \leq \theta_{n1}, \theta_{n2} < 2\pi$, $\mathrm{n1}, \mathrm{n2} \in \mathbb{Z}$, the direction difference between θ_{n1} and θ_{n2} denoted by $\theta_{\Delta n2} = \Delta(\theta_{n1}, \theta_{n2})$ is defined as followings.

$$\theta_{\Delta n_2} = \mathrm{sgn}\left(R_c(\theta_{n_1}, \theta_{n_2}) - R_{ac}(\theta_{n_1}, \theta_{n_2})\right) \min\left(R_c(\theta_{n_1}, \theta_{n_2}), R_{ac}(\theta_{n_1}, \theta_{n_2})\right) \qquad (1)$$

For a given direction θ_2 of $\overrightarrow{\mathrm{P_1P_2}}$ shown in Figure 1, difference $\theta_{\Delta 3}$ for $\overrightarrow{\mathrm{P_2P_3}}$ is the angle formed by rotating vector of θ_2 clockwise, and the difference $\theta_{\Delta 4}$ for $\overrightarrow{\mathrm{P_2P_4}}$ is angle formed by rotating vector of θ_2 anticlockwise. $\theta_{\Delta 3} < 0$ because $\mathrm{R_c}(\theta_2, \theta_3) < \mathrm{R_{ac}}(\theta_2, \theta_3)$ and $\theta_{\Delta 4} > 0$ comes from $\mathrm{R_c}(\theta_2, \theta_4) > \mathrm{R_{ac}}(\theta_2, \theta_4)$.

3. Statistical Straightness

This section mathematically illustrates the ideas of segmenting arbitrary digital contour to DSS with respect to the statistical straightness and the generalized connectedness described by the previous sections. It consists of three subsections. Section 3.1, introduces the statistical straightness and how to obtain DSS through segmentations based on it. Section 3.2, describes analyzing statistical straightness of a given DSS for enhancing its straightness. Section 3.3, supplements Section 3.2, for merging DSS.

3.1. DSS Based on Statistical Straightness

Suppose an arbitrary pixel $\mathrm{P_1} \in \mathrm{I_F}$ is given, a circle of radius $\mathcal{R}$ can be expanded at $\mathrm{P_1}$ and $\mathrm{S_1} = \left\{\mathrm{P} \in \mathrm{I_F} \setminus \{\mathrm{P_1}\} \middle| \left|\overrightarrow{\mathrm{P_1P}}\right| \leq \mathcal{R}\right\}$ represents foreground pixels found in circle. If $|\mathrm{S_1}| \neq 0$, then $\bar{\theta}_1 = \sum_{\mathrm{i1}=1}^{|\mathrm{S_1}|} \theta_{\mathrm{i1}} / |\mathrm{S_1}|$ where $\theta_{\mathrm{i1}} = \mathrm{R_{ac}}\left(\mathbb{O}, \overrightarrow{\mathrm{P_1P_{i1}}}\right)$, $\mathrm{P_{i1}} \in \mathrm{S_1}$ and $\mathrm{i1} \in \mathbb{Z}^+$

is the average direction of DSS consisting of accepted pixels $\left\{P \in S_1 \middle| \left|\Delta\left(\overline{\theta}_1, R_{ac}(\mathbb{O}, \overrightarrow{P_1P})\right)\right| \le \Theta\right\}$ and not-rejected pixels $\left\{P \in S_1 \middle| \left|\Delta\left(\overline{\theta}_1, R_{ac}(\mathbb{O},\overrightarrow{P_1P})\right)\right| > \theta\right\}$ with respect to a given direction difference threshold $\Theta < \pi/2$.

Afterwards, a semicircle is expanded at the first pixel of $\left\{p \in S_1 \middle| \min_p \left|\Delta\left(\overline{\theta}_1, R_{ac}(\mathbb{O}, \overrightarrow{P_1P})\right)\right|\right\}$ denoted by P_2 and its diameter is perpendicular to $\overline{\theta}_1$. Let $S_2 = \left\{P \in I_F \setminus \{S_1 \cup \{P_1, P_2\}\} \middle| |\overrightarrow{P_2P}| \le \mathcal{R}\right\}$ denote pixels found in semicircle. Suppose $|S_2| \ne 0$, let $f: \mathfrak{L} \to \mathbb{Z}$ denotes the map between interval $\mathfrak{L}(\theta_{i2}) = (\theta_{i2} - \sigma, \theta_{i2} + \sigma)$ where $\sigma \in \mathbb{R}$, $\theta_{i2} = R_{ac}(\mathbb{O},\overrightarrow{P_2P_{i2}})$, $P_{i2} \in S_2$, and number of vectors whose directions lying in $\mathfrak{L}$, then there is an inverse function $f^{-1}: \mathbb{Z} \to \mathfrak{L}$ maps $f(\mathfrak{L}(\theta_{i2}))$ back to θ_{i2} or the median of the ordered set comprising directions mapped to the same value of f. Then $\overline{\theta}_2 = f^{-1}(\max_{\theta_{i2}} f(\mathfrak{L}(\theta_{i2})))$, *i.e.*, $\overline{\theta}_2$ is the direction determined by $\mathfrak{L}$ of maximum value in a histogram of axes $\mathfrak{L}$ and f.

Generally, let $t \in \mathbb{Z}^+$ denotes the times of expanding circle or semicircle of radius $\mathcal{R} \ge 2$ at pixels $P_t \in I_F$ accepted by a DSS, then the set of foreground pixels detected by the tth circle expansion can be defined as below.

$$S_t = \begin{cases} \left\{P \in I_F \setminus \{P_1\} \middle| |\overrightarrow{P_1P}| \le \mathcal{R}\right\} & t = 1 \\ \left\{P \in I_F \setminus \{S_{t-1} \cup \{P_{t-1}\}\} \middle| |\overrightarrow{P_tP}| \le \mathcal{R} \cup \Delta\left(\overline{\theta}_{t-1}, R_{ac}(\mathbb{O},\overrightarrow{P_{t-1}P})\right) < \frac{\pi}{2}\right\} & t > 1 \end{cases} \tag{2}$$

Where $\overline{\theta}_{t-1}$ denotes the direction of DSS at the $(t-1)$th circle expansion and its definition is given by Definition 3 in the followings. If $DSS(t)$ denotes the set of points contained by a DSS at the tth circle expansion, w_t denotes the geometrical thickness or width of $DSS(t)$ and $\alpha \in \mathbb{R}$, then set of accepted pixels denoted by S_t^1 for S_t, is defined as below.

$$S_t^1 = \left\{P \in S_t \middle| \Delta\left(\overline{\theta}_{t-1}, R_{ac}(\mathbb{O},\overrightarrow{P_{t-1}P})\right) \le \Theta \cup w_t \le \alpha\mathcal{R}\right\} \tag{3}$$

Definition for set of not-rejected pixels S_t^0 is similar with S_t^1 except $\Delta\left(\overline{\theta}_{t-1}, R_{ac}(\mathbb{O}, \overrightarrow{P_{t-1}P})\right) \le \Theta$ is replaced by $\Delta\left(\overline{\theta}_{t-1}, R_{ac}(\mathbb{O},\overrightarrow{P_{t-1}P})\right) > \theta$. Let $P_1 \in I_F$ denotes an arbitrary given pixel, then pixel $P_t \in I_F$ which is chosen as the center for expanding circle or semicircle can be defined as following.

$$P_t \in \begin{cases} \{P_1\} & t = 1 \\ \left\{P \in S_t^1 \middle| \min_p \left|\Delta\left(\overline{\theta}_{t-1}, R_{ac}(\mathbb{O},\overrightarrow{P_{t-1}P})\right)\right|\right\} & t > 1, |S_t^1| \ne 0 \end{cases} \tag{4}$$

Definition 3. If Θ denotes the threshold of direction difference, $\sigma \in \mathbb{R}$ and $\mathfrak{A}_\theta = \left\{\theta_{i_j} = R_{ac}\left(\mathbb{O},\overrightarrow{P_jP_{i_j}}\right) \middle| P_{i_j} \in S_j^1, i, j \in \mathbb{Z}^+, 1 \le i \le |S_j^1|, 1 \le j \le t\right\}$ denotes the directions of all accepted pixels, statistical straightness or direction of DSS is defined as below.

$$\overline{\theta}_t = \begin{cases} \sum_{i=1}^{|S_1|} \theta_{i_1} / |S_1| & t = 1, |S_1| \ne 0 \\ f^{-1}\left(\max_{\theta_{i_j} \in \mathfrak{A}_\theta} f\left(\mathfrak{L}\left(\theta_{i_j}\right)\right)\right) & t > 1, |S_t^1| \ne 0 \end{cases} \tag{5}$$

Definition 4. If $|S_i^1| \ne 0$, $i \in \mathbb{Z}^+$, $1 \le i \le t$, then definition of generalized DSS without requirement of 4 or 8-connectedness at the tth circle expansion is given as following.

$$\mathrm{DSS}(t) = \cup_{\mathrm{i}=1}^{t}\left(\mathrm{S}_{\mathrm{i}}^{1} \cup \mathrm{S}_{\mathrm{i}}^{0}\right) \tag{6}$$

$\cup_{\mathrm{i}=1}^{t}\mathrm{S}_{\mathrm{i}}^{1}$ and $\cup_{\mathrm{i}=1}^{t}\mathrm{S}_{\mathrm{i}}^{0}$ respectively represent sets of accepted pixels and not-rejected pixels for $\mathrm{DSS}(t)$. If $\left|\mathrm{S}_{t+1}^{1}\right| = 0$, then $\mathrm{DSS} = \mathrm{DSS}(t)$ which means there will be no more circle expansion for segmentation of DSS and it will be analyzed for guaranteeing a strong statistic straightness.

A simple example may illustrate definitions mentioned above. Suppose $\Theta = 15°$, $\alpha = 2$ and $\mathrm{R} = 1.5$ pixel, for a given digital contour of eight pixels shown by squares of red centers in Figure 2, DSS is segmented by following steps.

For $t = 1$, P_1 in Figure 2, is initially given, then P_2 and Q are found in a full circle (not shown) expanded at P_1. At this moment, there are two possible directions: θ_2 of $\overrightarrow{\mathrm{P}_1\mathrm{P}_2}$ and θ_{Q} of $\overrightarrow{\mathrm{P}_1\mathrm{Q}}$. However, θ_2 and θ_{Q} are equally close to average direction, hence both P_2 and Q are accepted and chosen as centers for expanding next circle, meanwhile they are erased in I_{F}. DSS_1 denotes DSS containing three pixels.

For $t = 2$, two semicircles (not shown) are simultaneously expanded at P_2 and Q. In either semicircles, P_3 is found and $\overline{\theta}_{\mathrm{DSS1}}$ is estimated based on (5). $\theta_{\Delta 3}$ is found to be 0 with respect to $\overline{\theta}_{\mathrm{DSS1}}$ and $w_{\mathrm{DSS1}} < 2\mathcal{R}$, hence P_3 is accepted and chosen for the center of the next circle expansion. P_3 is erased.

For $t = 3$, P_4 is found in range, $\overline{\theta}_{\mathrm{DSS1}}$ remains same based on (5) and and $w_{\mathrm{DSS1}} < 2\mathcal{R}$. P_4 is not-rejected because $\theta_{\Delta 4} > \Theta$ and no pixel can be chosen as center for next expansion.

For $t = 4$, $|S_4^1| = 0$ because no semicircle can be expanded and segmentation of DSS_1 terminates. Finally, $DSS_1 = \{P_1, P_2, P_3, Q\} \cup \{P_4\}$. Since P_4 is not erased, P_4 is the initial pixel for DSS_2 shown in Fig.2. DSS_2 and DSS_3 are segmented in similar procedures.

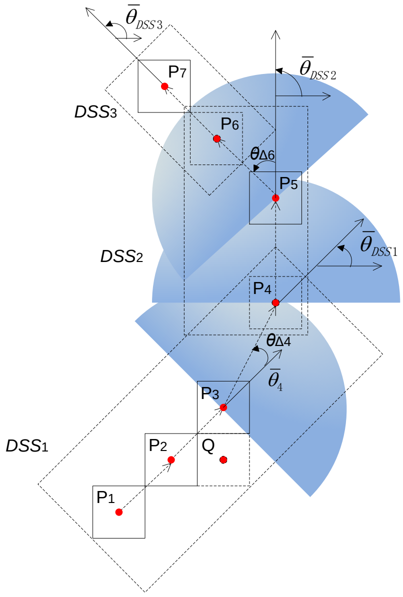

$\overline{\theta}_{DSS3}$
P7
DSS3
$\overline{\theta}_{DSS2}$
P6
$\theta_{\Delta 6}$
P5
$\overline{\theta}_{DSS1}$
DSS2
P4
$\theta_{\Delta 4}$
θ_4
P3
DSS1
P2
Q
P1

Figure 2. Not-Rejected Pixels

3.2. Analyzing Statistic Straightness of DSS

For any $\mathrm{DSS}(t)$ of $\left|\mathrm{S}_{t+1}^{1}\right| = 0$, its statistical straightness or direction should be analyzed because the segmentation based on Section 3.1 never takes into account the cases that the initial direction $\overline{\theta}_1$ may be an incorrect assumption of $\overline{\theta}_t$ and the direction distribution of $\mathrm{DSS}(t)$ may be multimodal.

Incorrect assumption of $\overline{\theta}_1$ leads to misclassifications of the accepted and not-rejected pixels, and thus resulting in unbalanced values of $\left|\bigcup_{i=1}^{t} S_i^1\right|$ and $\left|\bigcup_{i=1}^{t} S_i^0\right|$. The balance is simply estimated by the following criterion.

$$\frac{\left|\bigcup_{i=1}^{t} S_i^0\right|}{\left|\bigcup_{i=1}^{t} S_i^1\right|} > b \Leftrightarrow \mathrm{DSS}(t) \text{ is unbalanced} \tag{7}$$

Where $b \in \mathbb{R}^+$ is a given balance threshold. If (7) is met, then pixels in $\bigcup_{i=1}^{t} S_i^1$ and $\bigcup_{i=1}^{t} S_i^0$ are interchanged. Multimodal direction distribution leads to weak statistical straightness *i.e.*, the DSS is geometrically curvaceous. Multimodal distribution can be detected by investigating histogram h_t of axes $\mathfrak{L}\left(\theta_{i_j}\right)$ and f where $\mathfrak{L} = (\theta_{i_j} - \sigma, \theta_{i_j} + \sigma)$, $\theta_{i_j} \in \mathfrak{A}_\theta$, σ and $\mathfrak{A}_\theta$ are defined in Definition 3. For a given multimodal threshold $m \in \mathbb{R}^+$ and $\theta_{\max} = f^{-1}\left(\max_{\theta_{i_j} \in \mathfrak{A}_\theta} f\left(\mathfrak{L}\left(\theta_{i_j}\right)\right)\right)$ which is a fixed value on account of $\left|S_{t+1}^1\right| = 0$, h_t is considered multimodal if following condition is met.

$$|\mathfrak{I}| = \left|\left\{\mathfrak{L}\left(\theta_{i_j}\right) \in h_t \left| \frac{\max_{\theta_{i_j} \in \mathfrak{A}_\theta \setminus \{\theta_{\max}\}} f\left(\mathfrak{L}\left(\theta_{i_j}\right)\right)}{\max_{\theta_{i_j} \in \mathfrak{A}_\theta} f\left(\mathfrak{L}\left(\theta_{i_j}\right)\right)} > m \right.\right\}\right| > 1 \Leftrightarrow h_t \text{ is multimodal} \tag{8}$$

If (8) is met and $\Delta(\theta_{n1}, \theta_{n2}) > \Theta$ where θ_{n1} and θ_{n2} are consecutive elements in ordered set $\left\{\theta_{ij} \middle| \mathfrak{L}\left(\theta_{ij}\right) \in \mathfrak{I}\right\}$, then directions like θ_{n1} and θ_{n2} are collected in set $\mathfrak{S}_\theta$ and $\mathrm{DSS}(t)$ should be further segmented based on $\mathfrak{S}_\theta$. According to (4), there are $P_1, P_2, \ldots, P_t$ pixels as the centers of circles corresponding to resulting S_1^1, S_2^1, …, S_t^1. Suppose $t > 1$, $n \in \mathbb{Z}^+$ denotes a given parameter and $\arg\min_{\theta_\mathfrak{S}} \Delta\left(\overline{\theta}_j, \theta_\mathfrak{S}\right)$ represents direction value $\theta_\mathfrak{S} \in \mathfrak{S}_\theta$ minimizing $\Delta\left(\overline{\theta}_j, \theta_\mathfrak{S}\right)$, then $\mathrm{DSS}(t)$ is split at positions of pixels defined below.

$$\mathfrak{S}_p = \left\{P_j \left| \min_{\theta_\mathfrak{S} \in \mathfrak{S}_\theta}\left(\Delta\left(\overline{\theta}_j, \theta_\mathfrak{S}\right)\right)^{-1} \neq \min_{\theta_\mathfrak{S} \in \mathfrak{S}_\theta}\left(\Delta\left(\overline{\theta}_{j-n'}, \theta_\mathfrak{S}\right)\right)^{-1}, n' = 1,2,\ldots,n \right.\right\} \tag{9}$$

Where $n' = 1,2,\ldots,n$. Namely, $\theta_\mathfrak{S} \in \mathfrak{S}_\theta$ differing minimally from $\overline{\theta}_j$ is found for arbitrary P_j and compared with the last n consecutive directions in $\mathfrak{S}_\theta$. If all n comparisons fail, then P_j is envisaged as a split position for $\mathrm{DSS}(t)$. For all $P_j \in \mathfrak{S}_p$, $\mathrm{DSS}(t)$ is thus split to DSS_{n3}^s, $n3 = 1,2,\ldots,\left|\mathfrak{S}_p\right| + 1$ and the direction of each DSS_{n3}^s, is updated by using linear regression. Suppose $(x_{i'}, y_{i'}) \in \mathbb{Z}^2$ where $1 \le i' \le n = |\mathrm{DSS}_{n3}^s,|$ denotes coordinates of pixels contained by DSS_{n3}^s,, then its direction is estimated by the following formula.

$$\theta_{\mathrm{DSS}_{n_3}^s} = \tan^{-1}\left(\frac{\sum_{i'=1}^{n}(x_{i'} - \overline{x})(y_{i'} - \overline{y})}{\sum_{i'=1}^{n}(x_{i'} - \overline{x})^2}\right) \text{ where } \overline{x} = \frac{\sum_{i'=1}^{n} x_{i'}}{n}, \overline{y} = \frac{\sum_{i'=1}^{n} y_{i'}}{n} \tag{10}$$

Once $\theta_{\mathrm{DSS}_{n3}^s}$ is obtained, $w_{\mathrm{DSS}_{n3}^s}$ can be approximated by difference of maximal and minimal ordinates in the coordinate system whose origin is $(\overline{x}, \overline{y})$ and positive x-axis coincides with $\theta_{\mathrm{DSS}_{n3}^s}$. Pixels of maximal or minimal coordinates in rotated coordinate system are named marginal points and denoted by set $\mathfrak{E}_{\mathrm{DSS}_{n3}^s}$. Any pixel causing $w_{\mathrm{DSS}_{n3}^s} > \alpha\mathcal{R}$ will be removed from DSS_{n3}^s which guarantees DSS_{n3}^s is compatible with (6).

Figure 3 depicts a simple case of splitting a given DSS: DSS_1. Suppose $\Theta = 15°$, $m = 0.7$, $n = 2$ and a digital contour consisting of seven pixels in Figure 3 is segmented to DSS_1,then h_t of DSS_1 comprises two peaks: $\overline{\theta}_{\mathrm{DSS}_2} = 0$ of times 2 and $\overline{\theta}_{\mathrm{DSS}_3} = 45°$ of times 2, *i.e.*, $|\mathfrak{I}| = \left|\left\{\overline{\theta}_{\mathrm{DSS}_2}, \overline{\theta}_{\mathrm{DSS}_3}\right\}\right| = 1 > m$ based on (8) and h_t is multimodal. Because $\Delta\left(\overline{\theta}_{\mathrm{DSS}_2},\overline{\theta}_{\mathrm{DSS}_3}\right) > \mathit{\Theta}$, $\mathfrak{S}_\theta = \mathfrak{I}$. According to (9), $\mathfrak{S}_\mathrm{p} = \{\mathrm{P}_4\}$. Namely, follow the pixels defined by (4) for DSS_1, the direction of minimal difference to $\mathrm{R}_\mathrm{ac}\left(\mathbb{O},\overrightarrow{\mathrm{P}_1\mathrm{P}_2}\right)$ is $\overline{\theta}_{\mathrm{DSS}_2}$, for $\mathrm{R}_\mathrm{ac}\left(\mathbb{O},\overrightarrow{\mathrm{P}_2\mathrm{P}_3}\right)$ it also is $\overline{\theta}_{\mathrm{DSS}_2}$, for $\mathrm{R}_\mathrm{ac}\left(\mathbb{O},\overrightarrow{\mathrm{P}_3\mathrm{P}_4}\right)$ it changes to $\overline{\theta}_{\mathrm{DSS}_3}$. However, $\overline{\theta}_{\mathrm{DSS}_2}$ consecutively occurs 2 times which equals $\mathscr{n}$, this causes P_4 is labeled as a split position.

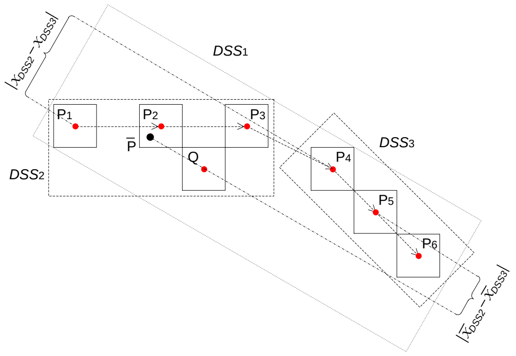


Figure 3. Split or Merge DSS

3.3. Merging DSS

Split discussed in Section 3.2, implies possibility of merging the split DSS with ones not from the same source. The merging is based on the differences between the directions of two compared DSS and their relative positions. For DSS_1, DSS_2, $\ldots\mathrm{DSS}_{|\mathfrak{D}|}$ in $\mathfrak{D}$ containing all DSS found in I_F, they are categorized to collections $\mathfrak{D}_\mathrm{M}$ defined below.

$$\left\{DSS_{k_1}, DSS_{k_2} \in \mathfrak{D} \,\middle|\, \left|\Delta\left(\overline{\theta}_{k_1},\overline{\theta}_{k_2}\right)\right| \leq \Theta, \left|\overline{x}_{k_1} - \overline{x}_{k_2}\right| \leq \alpha\mathcal{R}/2\right\} \tag{11}$$

Where $\mathrm{k1},\mathrm{k2} \in \mathbb{Z}^+$, $\mathrm{k1} \neq \mathrm{k2}$, $1 \leq \mathrm{k1},\mathrm{k2} \leq |\mathfrak{D}_\mathrm{M}|$, $\overline{\theta}_\mathrm{ki}$ and $\overline{\mathrm{x}}_\mathrm{ki}$ respectively denote direction and mean abscissa of DSS_ki, $\mathrm{i} = 1{,}2$ in a coordinate system of positive x-axis pointing to $\sum_{\mathrm{ki}=1}^{|\mathfrak{D}_\mathrm{M}|}\overline{\theta}_\mathrm{ki}/|\mathfrak{D}_\mathrm{M}|$. $\mathfrak{D}_\mathrm{M}$ is then further split to subgroups $\mathfrak{D}_{\mathrm{M}_\mathrm{k}}$, $\mathrm{k} = 1{,}2,\ldots$ defined as following for merging.

$$\left\{DSS_{k_3} \in \mathfrak{D}_M \,\middle|\, \left|x_{P_{n_4}} - x_{P_{n_5}}\right| \leq \alpha\mathcal{R}, P_{n_4} \in \mathfrak{E}_{DSS_{k_3}}, P_{n_5} \in \mathfrak{E}_{DSS_\mathrm{Max}}\right\} \tag{12}$$

Where $\mathrm{k3},\mathrm{n4},\mathrm{n5} \in \mathbb{Z}^+$, $1 \leq \mathrm{k3} \leq |\mathfrak{D}_\mathrm{M}|$, $1 \leq \mathrm{n4} \leq \left|\mathfrak{E}_{\mathrm{DSS}_\mathrm{k3}}\right|$, $1 \leq \mathrm{n5} \leq \left|\mathfrak{E}_{\mathrm{DSS}_\mathrm{Max}}\right|$, $\mathrm{DSS}_\mathrm{Max} = \arg\max_{\mathrm{DSS}_\mathrm{ki}}\left|\mathrm{S}^1_{\mathrm{DSS}_\mathrm{ki}}\right|$, $\mathrm{DSS}_\mathrm{ki} \in \mathfrak{D}_{\mathrm{M}_\mathrm{k}} \setminus \mathfrak{D}_{\mathrm{M}_{\mathrm{k}'}}$, $\mathrm{M}_\mathrm{k} \neq \mathrm{M}_{\mathrm{k}'}$ and $\mathrm{DSS}_\mathrm{Max}$ is the member of the maximal number of accepted points $\mathrm{S}^1_{\mathrm{DSS}_{\mathrm{k}_\mathrm{i}}}$ in $\mathfrak{D}_{\mathrm{M}_\mathrm{k}}$. Real values $\mathrm{x}_{\mathrm{P}_\mathrm{n4}}$ and $\mathrm{x}_{\mathrm{P}_\mathrm{n5}}$ respectively denote the abscissae of points P_n4 and P_n5.

Once $\mathfrak{D}_{\mathrm{M}_\mathrm{k}}$ is obtained, all members in each $\mathfrak{D}_{\mathrm{M}_\mathrm{k}}$ are removed from $\mathfrak{D}$ and their accepted points and not-rejected points are assigned to $\mathrm{DSS}_\mathrm{Max}$. The direction of $\mathrm{DSS}_\mathrm{Max}$ is finally updated by using linear regression of (10).

If DSS_2 and DSS_3 in Figure 3 are given for merging and suppose $\Theta = 45°$, $\alpha = 1.6$, $\mathcal{R} = \sqrt{5}$ pixel, then DSS_2 and DSS_3 will be merged as DSS_1. Because $\left|\Delta\left(\overline{\theta}_{\mathrm{DSS}_2},\overline{\theta}_{\mathrm{DSS}_3}\right)\right| = 45° \leq \Theta$ and $\left|\overline{\mathrm{x}}_{\mathrm{DSS}_2} - \overline{\mathrm{x}}_{\mathrm{DSS}_3}\right| \approx 0.75\sqrt{2} \leq \alpha\mathcal{R}/2$, *i.e.*, $\mathfrak{D}_\mathrm{M} = \{\mathrm{DSS}_2,\mathrm{DSS}_3\}$ based on (11), and $\mathrm{DSS}_\mathrm{Max} = \mathrm{DSS}_2$. In Figure 3, the mean of DSS_2 is a black point labeled by $\overline{\mathrm{P}}$ and P_5 is the mean of DSS_3. The largest difference between all marginal points of DSS_2 and DSS_3 is denoted by $\left|\mathrm{x}_{\mathrm{DSS}_2} - \mathrm{x}_{\mathrm{DSS}_3}\right|$ and shown in Figure 3. Since $\left|\mathrm{x}_{\mathrm{DSS}_2} - \mathrm{x}_{\mathrm{DSS}_3}\right| \approx 3/\sqrt{2} \leq \alpha\mathcal{R}$ holds, DSS_2 and DSS_3 are confirmed to be merged by (12). The resulting DSS is DSS_1 of direction computed based on (10).

4. Algorithm Implementation

This section introduces the implementation of ideas proposed in previous sections. The whole procedure consisting of two main subroutines: Subroutine 1: find DSS; Subroutine 2: merge DSS. The procedure is shown in Figure 4. Subroutine 1 reflects the segmentation and split discussed in Section 3.1, and 3.2, respectively. Subroutine 2, provides functionality of merging DSS described in Section 3.3.

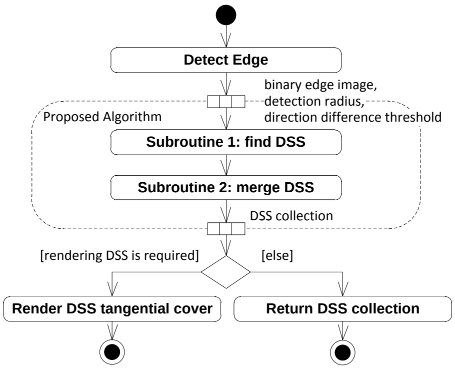


Figure 4. General Design of the Proposed Algorithm

4.1. General Schema of DES

Three classes are employed by the proposed algorithm, *i.e.*, DSSGroup, DSS and DSSPoint depicted by Figure 5. Names of these classes and their properties are self-introductory and they are discussed in the reverse order of their dependencies shown in Figure 5, in followings.

DSSPoint preserves the deviations of the current pixel and the previous pixel. The deviations determine the direction computed according to Definition 1. It also retains coordinates with respect to different coordinate systems for finding marginal points and estimating DSS direction by using (10). DSS Class holds points defined by (6). The marginal points are referenced by properties of type DSSPoint. DSSGroup maps to $\mathfrak{D}_{\mathrm{M}}$ of (11) and its property DSSForMerging reflects $\mathfrak{D}_{\mathrm{M}_\mathrm{k}}$ defined by (12). Its class diagram also depicts a constructor accepting two parameters DSS1 and DSS2. The rest properties of DSSGroup serve the computations required by (11).

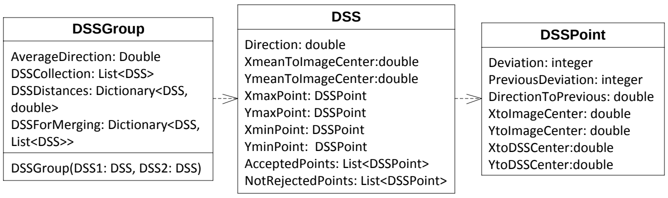

DSSGroup
AverageDirection: Double
DSSCollection: List<DSS>
DSSDistances: Dictionary<DSS, double>
DSSForMerging: Dictionary<DSS, List<DSS>>
DSSGroup(DSS1: DSS, DSS2: DSS)
DSS
Direction: double
XmeanToImageCenter:double
YmeanToImageCenter:double
XmaxPoint: DSSPoint
YmaxPoint: DSSPoint
XminPoint: DSSPoint
YminPoint: DSSPoint
AcceptedPoints: List<DSSPoint>
NotRejectedPoints: List<DSSPoint>
DSSPoint
Deviation: integer
PreviousDeviation: integer
DirectionToPrevious: double
XtoImageCenter: double
YtoImageCenter: double
XtoDSSCenter:double
YtoDSSCenter:double

Figure 5. Class Diagram for the Proposed Algorithm

Based on the classes depicted in Figure 5, the schema of the proposed algorithm is shown in form of UML activity diagram of Figure 6. Subroutine 1 and 2 respectively contains 5 and 3 subroutines. Subroutine 1.1 and 1.2 implement segmentation described by Section 3.1. Subroutine 1.3, to 1.5 split DSS based on criterions discussed in Section 3.2. Packing these subroutines into a single unit Subroutine 1 avoids the redundancy of two separate loops respectively for the segmentation and splitting. Subroutine 2 merges DSS according to the logic presented in Section 3.3.

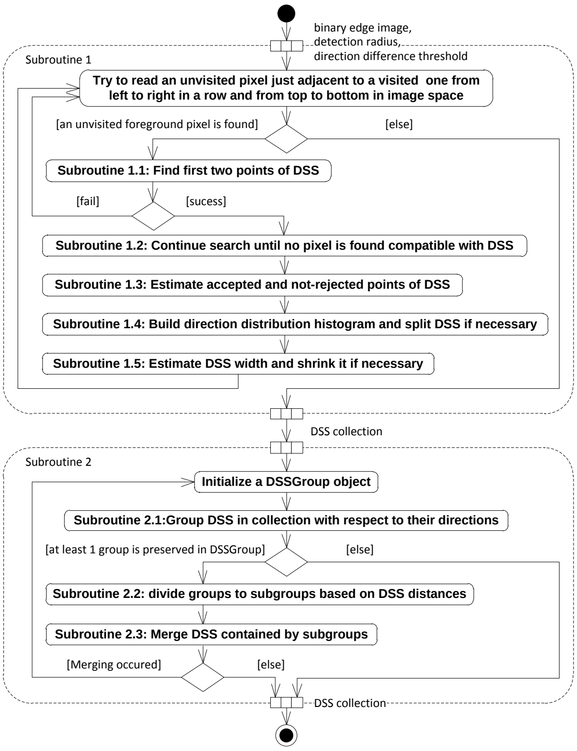

binary edge image,
detection radius,
direction difference threshold
Subroutine 1
Try to read an unvisited pixel just adjacent to a visited one from left to right in a row and from top to bottom in image space
[an unvisited foreground pixel is found]
[else]
Subroutine 1.1: Find first two points of DSS
[fail]
[sucess]
Subroutine 1.2: Continue search until no pixel is found compatible with DSS
Subroutine 1.3: Estimate accepted and not-rejected points of DSS
Subroutine 1.4: Build direction distribution histogram and split DSS if necessary
Subroutine 1.5: Estimate DSS width and shrink it if necessary
DSS collection
Subroutine 2
Initialize a DSSGroup object
Subroutine 2.1:Group DSS in collection with respect to their directions
[at least 1 group is preserved in DSSGroup]
[else]
Subroutine 2.2: divide groups to subgroups based on DSS distances
Subroutine 2.3: Merge DSS contained by subgroups
[Merging occured]
[else]
DSS collection

Figure 6. General Schema of the Proposed Algorithm

4.2. Algorithms for Identifying and Splitting DSS

This section sequentially introduces subroutines comprised by Subroutine 1 in order of the logic flow shown in Figure 6. The proposed algorithm begins with Subroutine 1.1 which assumes the input pixel is the first point of an unknown DSS and expands a full circle at the point which corresponds to P_1 of (4). All foreground pixels lying in circle are envisaged as potential points of DSS, *i.e.*, S_1 from (2), and their directions corresponding to input pixel are computed based on Definition 1. The average of the computed directions, *i.e.*, $\overline{\theta}_1$ of (5), is in turn compared with these directions and employed for categorizing pixels as accepted or not-rejected corresponding to S_1^1 and S_1^0 from (3). The point with the direction closest to the average is chosen as the center of the initial semicircle expansion in Subroutine 1.2, *i.e.*, P_2 of (4).

Subroutine 1.2 assumes α in (3) is set to 2, namely, the maximal value of $\mathcal{w}_t$ for any DSS is the diameter of semicircle. Subroutine 1.2 repeatedly expands semicircle whose diameter perpendicular with the DSS direction computed in previous search, *i.e.*, $\overline{\theta}_{t-1}$ from (5) at pixel P_t given by (4). It then collects pixels within the semicircle as S_t defined by (2) and statistically updates DSS direction as $\overline{\theta}_t$ based on S_{t-1}^1 of (3). For $\theta_{\mathrm{i_j}} \in \mathfrak{A}_\theta$, $\mathfrak{L}\left(\theta_{\mathrm{i_j}}\right) = (\theta_{\mathrm{i_j}} - \sigma, \theta_{\mathrm{i_j}} + \sigma)$ is obtained by using $\left\lfloor\theta_{\mathrm{i_j}}\right\rfloor$, *i.e.*, value of σ varies with $\theta_{\mathrm{i_j}}$. Collected pixels are finally categorized to S_t^1 and S_t^0 based on (3). This procedure repeats until no further pixel can be found within semicircle.

Subroutine 1.3 is the first step to split a given DSS. It simply computes the ratio defined by (7) with assumption that $\mathcal{b}$ is set to 1.5 for estimating whether the numbers of accepted and not-rejected pixels of a given DSS are unbalanced.

Subroutine 1.4 attempts to check whether $\mathcal{h}_t$ of a given DSS is multimodal and estimate split positions if so. The multimodal threshold $\mathcal{m}$ adapted in (8) is set to 0.7 and $\mathcal{n}$ employed in (9) is set to 4. When a DSS is given, its histogram of axes $\mathfrak{L}\left(\theta_{\mathrm{i_j}}\right)$,

$\theta_{i_j} \in \mathfrak{A}_\theta$ and f is built and ratio of maximal and submaximal values of f in $\mathcal{h}_t$ is computed. Value of $\mathfrak{L}\left(\theta_{i_j}\right)$ is obtained by using $\left\lfloor\theta_{i_j}\right\rfloor$ just as Subroutine 1.2 and the ratio is compared with $\mathcal{m}$. The values of θ_{i_j} meeting criterion (8) are recorded and removed from $\mathcal{h}_t$. The comparison of maximal and submaximal values of altered $\mathcal{h}_t$ continues until (8) is not met and all values of θ_{i_j} causing (8) met are preserved in $\mathfrak{J}$. If there is at least one pair of such values retained in $\mathfrak{J}$, *i.e.*, $|\mathfrak{J}| > 1$, then Subroutine 1.4 tries to find the split positions based on $\mathfrak{A}_\theta$ and (9). (9) is implemented by sequentially visiting pixels in their collective order and check whether the value in $\mathfrak{A}_\theta$ closest to direction of current pixel is different from the previous closest value in $\mathfrak{A}_\theta$. If the time of closest value changes $\mathcal{n} = 4$ times, then the current visited pixel is marked as a split position. Then DSS is split at these pixels and directions of all new DSS are updated by using (10). Subroutine 1.5 simply removes any pixels causing geometrical thickness of DSS exceeding diameter of semicircle.

4.3. Algorithms for Merging DSS

This section introduces Subroutine 2.1, to 2.3 comprised by Subroutine 2 serving for merging DSS. Subroutine 2.1 attempts to categorize DSS based on their direction similarities estimated by (11). Since the coordinate computation is relatively coasty, the distance estimation associated with (11) is integrated with Subroutine 2.2.

Subroutine 2.2, is critical for merging DSS. It finds DSS of maximal number of accepted pixels then estimates the distances involved in (11) between the found DSS and all other DSS in the group, then it estimates the distances of marginal points associated with (12) to categorize DSS to different merging groups.

Subroutine 2.3 completes merging phrase by simply assigning all points of DSS for merging to the DSS of maximal number of accepted points for each merging group. Then the collection of merged DSS is returned as the final result of the proposed algorithm.

5. Experimental Results

This section illustrates the experimental results of adopting the proposed algorithm for ideal image Ellipse and six real-life images respectively named Cameraman, House, Lena, Pepper, Puzzle and Tower. Visual segmentation results are shown in Figure 7, and Figure 8. Time consumed by eight subroutines of the proposed algorithm is shown through Figure 9, to Figure 13. Except radius $\mathcal{R}$ and difference threshold Θ given by user input, the rest parameters are empirically set, *i.e.*, α in (3) is set to 2, $\mathcal{b}$ of (7) is set to 1.5, $\mathcal{m}$ adopted in (8) is set to 0.7 and $\mathcal{n}$ employed in (9) is set to 4.

In Figure 7, contour of an ideal ellipse is segmented by using the proposed algorithm with respect to different values of $\mathcal{R}$ and Θ respectively reflected by the letter R and D of labels, *e.g.*, label R5D5 denotes $\mathcal{R} = 5$ and $\Theta = 5$. The rows of Figure 7, show visual results when one of $\mathcal{R}$ and Θ is fixed and the other increases. Apparently, the proposed algorithm is somehow more sensitive to $\mathcal{R}$ than Θ, *e.g.*, the differences among results listed in upper row are relatively small than lower row. This is also reflected by columns, especially the middle one.

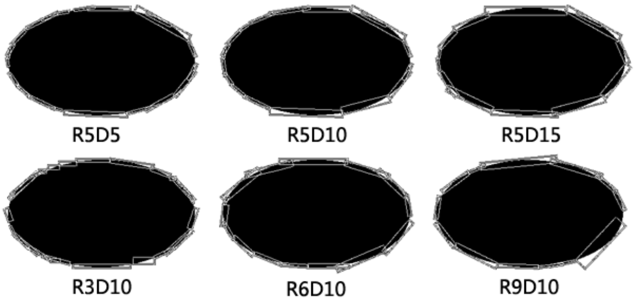


Figure 7. Segmentation of Ellipse Contour

In Figure 8, visual results of six real-life images of resolution 256-by-256 processed by the proposed algorithm are labeled by their names. Since their original images can be easily found in internet, they are not shown here. Because of the low resolution, value of $\mathcal{R}$ is limited to 3 which reduces the detection range of the proposed algorithm and number of sampled pixels. Although insufficient range guarantees the fine contour can be segmented reasonably, it leads to discontinuous segmentations of long edges as presented in bottoms of House and Tower, or top of Puzzle.

Discontinuous segmentations can be eliminated at cost of increasing resolutions. Figure 9 illustrates partial segmentation results of image House and Puzzle at resolution 1280-by-1280 and it also depicts Canny edge images, *i.e.*, inputs of the proposed algorithm. There are mainly two lines in edge image of House, the upper one is fully identified by the proposed algorithm and the lower one is partially recognized due to the noise. Edge image of Puzzle contains discontinuous edges and most of them are identified and connected by the proposed algorithm.

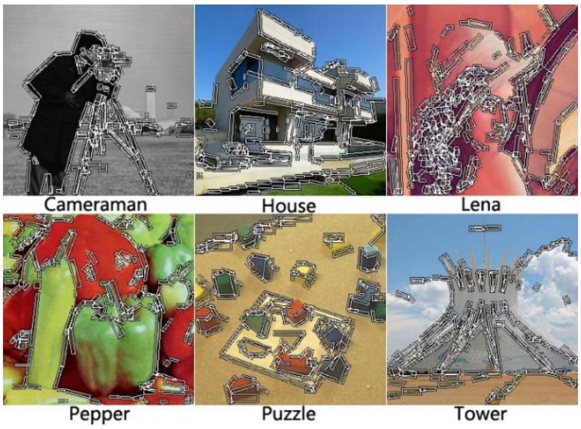
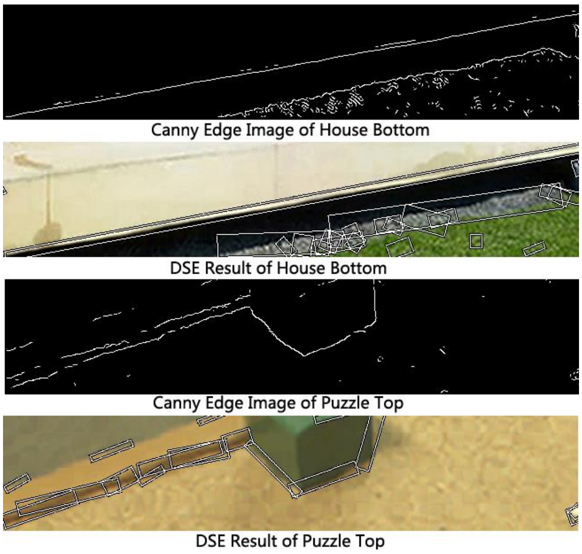


Figure 8. Segmentation Results of Real-Life Images

Figure 10, depicts the general time of processing six images whose resolutions vary within range 128-by-128 to 1280-by-1280 by steps of 32-by-32. The abscissa only shows one dimension of resolution. As shown in Figure 10, the most computationally expansive processing is adopted by Lena. Puzzle requires the least computational time. This is mainly because there are a large number of edges tangled in lower-left area of Lena while edges in Puzzle mostly are isolated.



Canny Edge Image of House Bottom

DSE Result of House Bottom

Canny Edge Image of Puzzle Top

DSE Result of Puzzle Top

Figure 9. Segmentation Results of Real-Life Images

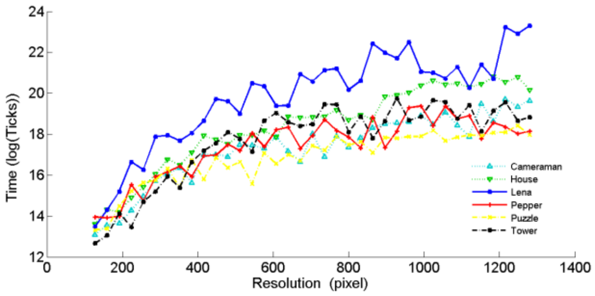


Figure 10. General Time of Processing Images

Figure 11, illustrates the time consumed for processing six images at resolutions 128-by-128, 640-by-640 and 1280-by1280 from left to right for each image. Each bar in Figure 11 consists of eight blocks distinguished by colors. From bottom to up, these blocks sequentially represent the time consumed by Subroutine 1.1 to Subroutine 2.3. Obviously, processing of Lena and Puzzle consume the most and the least time respectively. From the perspective of subroutines, Subroutine 2.1, *i.e.*, the orange block, makes the most contribution for increasing processing time, especially for cases of Lena and Tower. Oppositely, when Subroutine 2.1 remains stable just like in case of Puzzle, it results in a general low processing time.

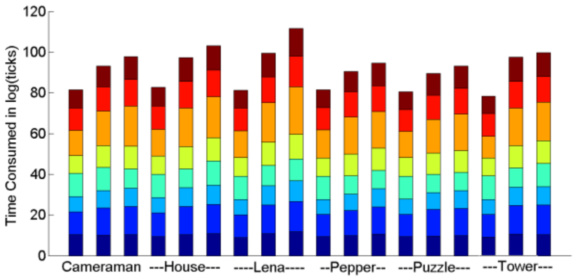


Figure 11. Time Consumed By Subroutines

Figure 12, to Figure 14, sequentially show the detailed distributions of the time consumed by subroutines for processing Lena, Tower and Puzzle at resolutions varying from 128-by-128 to 1280-by-1280. As indicated in Figure 11, most subroutines except

Subroutine 2.1 approximately consume similar time for different resolutions in Figure 12, to Figure 14. Subroutine 2.1, *i.e.*, the upper most polygonal line in all three figures, varies drastically in Figure 12, and the left half part of Figure 13. This maps to sever drops for three bars of Lena and the first two bars of Tower in Figure 11, and the shape of polygonal line of Lena in Figure 10, approximately resembles the one of Subroutine 2.1 shown in Figure 12. Mild changes of Subroutine 2.1 in Figure 14, lead to small drops among bars of Puzzle in Figure 11, and a low increasing rate of general time shown in Figure 10.

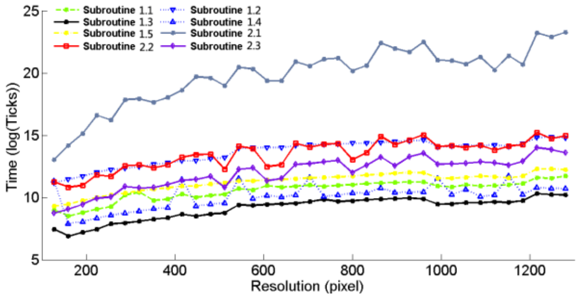


Figure 12. Time Consumed for Processing Image Lena

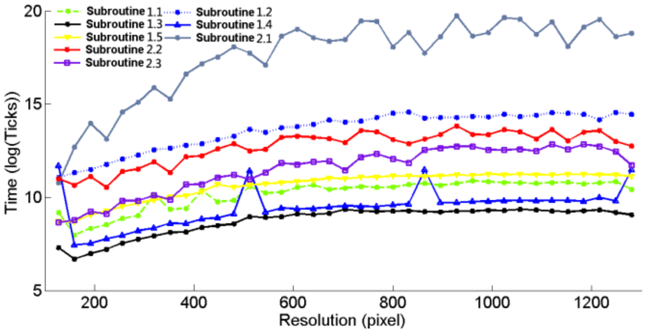


Figure 13. Time Consumed for Processing Image Tower

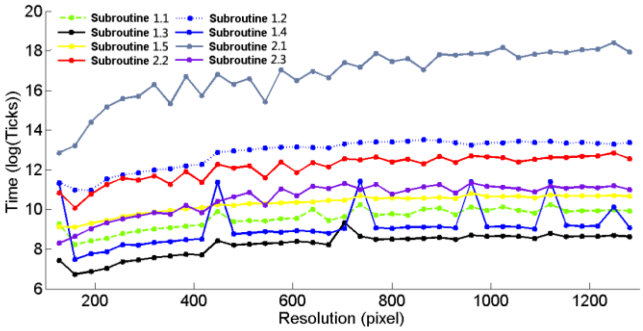


Figure 14. Time Consumed for Processing Image Puzzle

6. Conclusion

This paper proposed a DSS identifying algorithm based on a statistical strategy. The strategy depends on the dynamic estimating of the direction between two foreground pixels bound by the generalized connectedness, and it infers DSS direction through investigating the distribution of the estimated directions in the exploration. After the exploration, all found DSS are further analyzed based on the modality of histograms; splitting and merging are performed when the histograms are found multimodal. Conceptual implementation of the proposed algorithm is introduced graphically by using

UML activity diagrams. The experimental results of the artificial and real-life images are illustrated and discussed. The bottleneck of the proposed algorithm is found to be Subroutine 2.1 and the optimization of this subroutine will be focused in our future work.

References

[1] X. Jiang, X. You, Y. Yuan and M. G. Gong, "A method using long digital straight segments for fingerprint recognition", Neurocomputing, vol. 77, no. 1, **(2012)** Feb., pp. 28-35.

[2] J. Junior and A. R. Backes, "Shape classification using line segment statistics", Inform. Sci., vol. 305, **(2015)** Jun., pp. 349-356.

[3] A. O. Ok, J. D. Wegner, C. Heipke, F. Rottensteiner, U. Soergel and V. Toprak, "Matching of straight line segments from aerial stereo images of urban areas", Journal of Photogrammetry and Remote Sensing, vol. 74, **(2012)** Nov., pp. 133-152.

[4] B. Kerautret and J. O. Lachaud, "Meaningful scales detection along digital contours for unsupervised local noise estimation ", IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 12, **(2012)**, pp. 2379-2392.

[5] I. Debled-Rennesson, "A linear algorithm for segmentation of digital curves", Int. J. of Pattern Recognition and Artificial Intell., vol. 9, no. 4, **(1995)**, pp. 635-662,

[6] J.-O. Lachaud and M. Said, "Two efficient algorithms for computing the characteristics of a subsegment of a digital straight line", Discrete Appl. Math., vol. 161, no. 15, **(2013)** Oct., pp.2293-2315.

[7] J. S. D. Ouattara, E. C. Andres, G. L. S. R. Zrour and T. M. Y. Tapsoba, "Remainder approach for the computation of digital straight line subsegment characteristics", Discrete Appl. Math., vol. 183, **(2015)**, Mar., pp. 90-101.

[8] I. D. Rennesson, R. Je. Luc, J. R. vDegli, "Segmentation of discrete curves into fuzzy segments", Electron. Notes in Discrete Math., vol. 12, **(2003)** Mar., pp. 372-383.

[9] I. D. Rennesson, F. Feschet and J. R. Degli, "Optimal blurred segments decomposition of noisy shapes in linear time", Comput. & Graph., vol 30, no 1, **(2006)** Feb., pp. 30-36.

[10] I. Sivignon, "Fast recognition of a Digital Straight Line subsegment: Two algorithms of logarithmic time complexity", Discrete Appl. Math., vol. 183, **(2015)** Mar., pp. 130-146.

[11] L. Buzer, "A simple algorithm for digital line recognition in the general case", Pattern Recognition, vol. 40, no. 6, **(2007)** Jun., pp. 1675-1684.

[12] A. Faure, L. Buzer and F. Feschet, "Tangential cover for thick digital curves", Pattern Recognition, vol. 42, no. 10, **(2009)** Oct., pp. 2279-2287.

[13] J. Rumbaugh, I. Jacobson and G. Booch, Unified Modeling Language Reference Manual, 2nd ed. MA, USA: Addison-Wesley, **(2010)**.