



Cite this article: Aghamirmohammadali SMA, Bozorgmehry Boozarjomehry R, Abdekhodaie M. 2018 Modelling of retinal vasculature based on genetically tuned parametric L-system. *R. Soc. open sci.* **5**: 171639. <http://dx.doi.org/10.1098/rsos.171639>

Received: 27 October 2017

Accepted: 3 April 2018

Subject Category:

Engineering

Subject Areas:

computational biology/computer modelling and simulation/biomathematics

Keywords:

retinal vasculature, parametric L-system, L-system inverse problem, mathematical morphology

Author for correspondence:

Ramin Bozorgmehry Boozarjomehry
e-mail: rbozorgmehry@sharif.edu

Modelling of retinal vasculature based on genetically tuned parametric L-system

Seyed Mohammad Ali Aghamirmohammadali,
Ramin Bozorgmehry Boozarjomehry and
Mohammad Abdekhodaie

Chemical and Petroleum Engineering Department, Sharif University of Technology, Azadi Av., Tehran, Iran

SMAA, 0000-0002-9898-6554; RBB, 0000-0002-3270-9896

Structures of retinal blood vessels are of great importance in diagnosis and treatment of diseases that affect the eyes. Parametric Lindenmayer system (L-system) is one of the powerful rule-based methods that has a great capability for generating tree-like structures using simple rewriting rules. In this study, a novel framework, which can be used to model the retinal vasculature based on available images, has been proposed. This framework presents a solution to special instance of a general open problem, the L-system inverse problem, in which L-system rules should be obtained based on images representing a particular tree-like structure. In this study, genetic algorithm with a novel objective function based on feature matching and an L-system grammar comparison has been used along with nonlinear regression to solve the parametric L-system inverse problem. The resulting L-system growth rules have been employed to predict inaccessible vascular branches. Graphical and quantitative comparison between model results and target structures of different case studies reveals that the proposed framework can be used to generate the structure of retinal blood vessels accurately. Even in the cases lacking sufficient image data, it can provide acceptable predictions.

1. Introduction

Blood vessels play a critical role in diagnosis and treatment of many diseases such as cancer, hypertension, atherosclerosis and diabetes [1]. Over the past two decades, many works were devoted to study vasculogenesis [2], angiogenesis [3]

and blood vessel morphology [4]. These studies have often focused on physiology [5], modelling [6], and simulation of the morphology and behaviour of blood vessels [7].

Morphological analysis of blood vessels can be used to identify conditions and diseases [8]. Also, a reliable vascular structure model is necessary to investigate haemodynamic parameters and drug delivery conditions in the circulatory system [9]. Image data is a useful tool to extract blood vessel morphological properties such as length, width, tortuosity and branching pattern. Retinal fundus image is one of the most important data for diagnosis and treatment of diseases that change the morphological and functional characteristics of blood vessels [8]. This is due to the fact that retina is the only organ whose blood vessels can be visualized non-invasively and directly *in vivo*. Retinal blood vessel extraction and modelling has found applications in automatic early disease diagnosis and monitoring such as diabetic retinopathy, hypertensive retinopathy, and branch retinal vein occlusion [10], evaluation and optimization of eye nutrient, oxygen, and drug delivery [11,12], as well as examination of the factors affecting retinal vascularization and angiogenesis [13].

Previous works on retinal blood vessel segmentation and modelling can be mainly divided into three categories: image processing methods, mathematical modelling methods and rule-based methods.

Image processing is the most common tool for vessel extraction. Various techniques were developed for retinal fundus image analysis and segmentation [14]. Kirbas & Quek [15] have divided image processing methods for retinal blood vessel segmentation into five main categories: (i) pattern recognition techniques, (ii) model-based approaches, (iii) tracking-based approaches, (iv) artificial intelligence-based approaches, and (v) tube-like object detection approaches. Performance of each method can be evaluated and compared with indexes such as sensitivity, specificity and accuracy. The results of image processing methods are generally in good agreement with source images which typically have an accuracy of more than 0.8, a sensitivity of more than 0.6 and a selectivity of more than 0.9 [14]. However, image processing techniques are quite sensitive to image quality and resolution, and they are not able to estimate or reproduce low resolution branches. Furthermore, these techniques can only be used for identification of the vasculature. However, they are not capable of predicting the changes in the structure of vasculatures resulting from various diseases. In addition, time-variant modelling of angiogenesis or vasculogenesis needs other techniques such as mathematical modelling or rule-based modelling methods.

Extensive experimental works and mathematical modelling were conducted to investigate blood vessel formation, sprouting, and growing processes [16,17]. Differences between the models can be categorized in terms of problem scale, affecting factors, assumptions and model complexity. Various factors that include vascular endothelial growth factor, platelet-derived growth factor, fibronectin, tumour angiogenesis factor, matrix metalloproteinases and angiopoietin-1 are taken into the consideration to model retinal vasculature [18]. On the other hand, different modelling approaches such as continuum-based modelling, discrete modelling, stochastic modelling and deterministic modelling were also applied in this regard [19–21]. The mathematical modelling is based on the basic knowledge of vasculogenesis and angiogenesis physiology. The complex nature of blood vessel formation and growth, as well as limited knowledge of the interaction between effective factors, are the main barriers for mathematical modelling. It should also be mentioned that assumptions used in the various models along with unknown parameters result in an inconsistency between morphological properties of simulated vascular networks and image data.

The common rule-based modelling approaches are diffusion-limited aggregation (DLA) and Lindenmayer system (L-system). Since Witten & Sander [22] proposed DLA, some efforts were conducted to model biological structure, such as vasculature network [23,24]. DLA is based on Brownian motion and aggregation of particles to form tree-like fractal structure. Family *et al.* [25] showed that fractal dimension in the normal retina is approximately equal to that in the results from DLA modelling. However, structures derived from DLA modelling do not match the morphological information obtained based on the image data due to the stochastic nature of the DLA approach.

Originally, Lindenmayer [26] developed L-system to model cell division in multicellular organisms. Prusinkiewicz & Lindenmayer [27] discussed the evolution of simple L-system to parametric, stochastic, context-sensitive, and bracketed L-system, which can be used to simulate complex structures. Zamir [28] used parametric L-system to model arterial branching by applying a pre-defined rule to create tree-like fractal structures. He also introduced an algorithm based on bifurcation index [28] and Murray's law [29,30] to calculate morphological properties such as length, diameter and the angles in each branch. He showed that a constant bifurcation ratio results in a uniform fractal structure that is rarely observed in the physiological system. To alleviate this, he used random bifurcation ratio for

each parameter calculation to produce more realistic structures. Liu *et al.* [31] employed the constant bifurcation index and a random multiplier in calculations to have a more accurate pattern. Hichem & Malek [32] introduced an automatic computer-aided algorithm to determine branch crossovers and endpoints for diameter calculations. They also tried to test the prediction capability of parametric L-system with modelling of blood vessels that were smaller than image resolution. Owing to the complexity of unknown rules and parameters, none of those mentioned simulated structures was matched perfectly with image data.

Kókai *et al.* [33] presented grammatical retina description with evolutionary algorithms to model retinal vasculature structure automatically. They used genetic algorithm (GA) as an optimization tool to achieve grammatical representation of image data. However, the output of their algorithm is merely a final alphabetic string without any rewriting rule which is based on image resolution and cannot be used to predict inaccessible branches in the obtained structure.

In the present work, a novel method was proposed to model, simulate and analyse the retinal blood vessels. The proposed method is based on parametric L-system and GA for rule extraction and parameter estimation of the model which regenerates retinal blood vessels. In fact, such a problem is formulated as a special case of an open problem called parametric L-system inverse problem [34]. In the current work, an algorithm is proposed for the first time to solve this problem and find the parametric growth rules of a target structure. For this purpose, a novel hybrid fitness function was introduced based on which the modelling of retinal vasculature is formulated as a mixed integer nonlinear programming (MINLP) problem which was solved by GA to that purpose. Also using obtained L-system model to reconstruct inaccessible branches as prediction capability has been investigated.

The rest of the paper is organized as follows. Lindenmayer system is briefly described in §2, after which the application of L-system in the retinal vasculature modelling is explained in §3. Since there are lots of parameters involved in the growth rule of the parametric L-system, and they should all be estimated efficiently, the formulation of such a problem as an MINLP is elaborated in §4. The optimization technique which is a newly developed variant of GA is introduced in §5, which is then followed by §6 in which the proposed method is described. Section 7 contains the results obtained by the proposed framework for retinal vasculatures of two selected benchmarks.

2. L-system

The main concept of L-system is to use simple objects to generate complex objects by applying a set of rewriting rules. Each L-system contains an initiator string called axiom, and a set of rewriting rules.

Context-free and deterministic L-system, called DOL-system, can be shown as an ordered triplet $G = \langle V, \omega, P \rangle$. If V is the alphabet of the L-system, V^* and V^+ are the set of all words and non-empty words over V , respectively. $\omega \in V^+$ is a non-empty word called axiom and $P \subset V \times V^*$ is a finite set of rewriting or growth rules. A growth rule is written as 'predecessor \rightarrow successor'. The predecessor is iteratively replaced by the successor in the resulting string. In deterministic L-system, there is exactly one successor for each predecessor [27].

In order to take into account the impact of the external factors on the growth and evolution of the object structure, Lindenmayer [35] proposed parametric L-system. Parametric L-system is considered as an ordered quadruplet, $G = \langle V, \Sigma, \omega, P \rangle$, where Σ is the set of formal parameters [27]. Usually, parameters represent the length and angle dimensions which can be varied in each rewriting step through nonlinear or linear functions. Also, bracketed L-system was presented to model tree-like structures [26].

Turtle interpretation of string has been proposed to convert L-system strings to graphical structures [36]. A state of the turtle was considered as (x, y, α) , where (x, y) demonstrate the Cartesian coordinates of turtle position and the angle α represent turtle orientation. The step length and the step angle were considered to be d and δ , respectively [27]. L-system symbols which were used in this work for retinal vasculature modelling are shown in table 1.

According to the L-system model, the axiom is going through the changes described by the growth rules. At the end of each growth cycle, the obtained string acts similar to the axiom. This procedure continues until the stage at which none of the growth rules can induce any change at the end of cycle string, or it gets to the maximum desired growth cycle. The following example is provided to clarify the L-system rewriting and interpretation procedure.

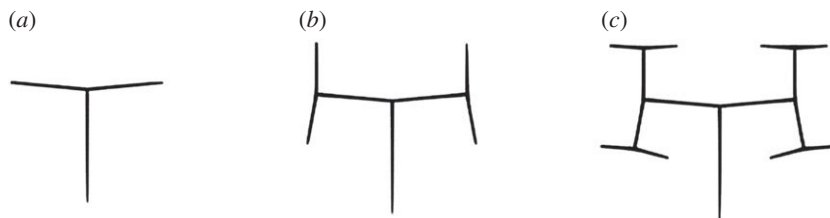


Figure 1. Interpretation of L-system defined in example 2.1: (a) growth cycle = 1, (b) growth cycle = 2, and (c) growth cycle = 3 [27].

Table 1. Turtle interpretation of symbols.

symbols	interpretation
$F(d), A(d)$ and $B(d)$	move forward and change the state to $(x + d \cos \alpha, y + d \sin \alpha, \alpha)$
$+(\delta)$	turn left and change the state to $(x, y, \alpha + \delta)$
$-(\delta)$	turn left and change the state to $(x, y, \alpha - \delta)$
[push the state on the stack
]	pop the state from the stack, move to (x, y, α)
\emptyset	null operator

Example 2.1. The following parametric L-system [27] results in the structures shown in figure 1:

$$\begin{aligned} \omega &: A(100, 90^\circ) \\ p_1 &: A(Z) \rightarrow F(Z) \left[+A \left(\frac{Z}{R} \right) \right] \left[-A \left(\frac{Z}{R} \right) \right] \} \\ R &= 1.5, \quad \delta = 85^\circ \end{aligned} \tag{2.1}$$

$$\begin{aligned} \text{growth cycle 0} & A(100) \\ \text{growth cycle 1} & F(100)[+A(66.6)][-A(66.6)] \\ \text{growth cycle 2} & F(100)[+F(66.6)[+A(44.4)][-A(44.4)]][-F(66.6)[+A(44.4)][-A(44.4)]] \\ \text{growth cycle 3} & F(100) \\ & \left[+F(66.6)[+F(44.4)[+A(29.6)][-A(29.6)]][-F(44.4)[+A(29.6)][-A(29.6)]] \right] \\ & \left[-F(66.6)[+F(44.4)[+A(29.6)][-A(29.6)]][-F(44.4)[+A(29.6)][-A(29.6)]] \right]. \end{aligned} \tag{2.2}$$

3. Application of L-system in retinal vasculature modelling

Parametric L-system can be used to model non-uniform tree-like structures such as retinal blood vessels. It could be shown that every tree-like structure can be represented as an L-system string. However, obtaining axiom and rewriting rules from a complex string is a complicated problem which is known as L-system inverse problem. The variant nature of string and parameters in L-system which can be changed during each rewriting growth cycle and also tracking these changes make this problem difficult to tackle.

Kókai *et al.* [33] converted a segment of preprocessed retina image shown in figure 2 to an L-system string.

The following string was given by Kókai *et al.* [33] to describe the structure of the top-right quarter of retina image shown in figure 2:

$$\begin{aligned} & F(37)[+(38)F(16)[+(27)F(11) + (14)F(12)+ \\ & (6)F(19)[+(47)F(19)][-(6)F(17)]][-(14)F(22) + (37)F(19)][-(6)F(12)+ \\ & (34)F(18) + (32)F(14)][-(43)F(24)]]][F(37)[-(7)F(8) + (43)F(24)][-(14)F(26)]]], \end{aligned} \tag{3.1}$$

in which $F(d)$ represents a line of length d and $\pm(\delta)$ denotes rotation by angle δ .

However, their work was only limited to finding a final L-system string while the growth rules required for producing this string were not provided. Growth rules can be applied to predict zones with low resolution, high noise or out of image scope. Also, segmentation of overlaid branches with different origins will be possibly more reliable.

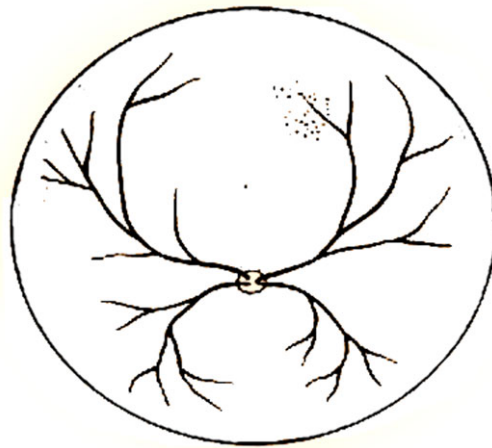


Figure 2. Preprocessed retina image presented by Kókai *et al.* [33].

Some efforts have been conducted to find rewriting rules from a non-biological structure. Koza [37] used genetic programming to achieve rewriting rules for non-parametric L-system using a pattern matching approach as the fitness function. Santos & Coelho [38] developed a method based on string growth analysis that can be used to convert a non-parametric string to its original L-system. Št'ava *et al.* [34] introduced a new method based on recognizing similar elements in images and assigning L-system alphabets to them. However, similar elements and uniform structures are rarely seen in biological structures. All of the mentioned methods are rooted in the concept of solving a non-parametric L-system inverse problem. Here, a new method is proposed to obtain parametric L-system from image data to simulate non-uniform structures such as retinal vasculature. This method was implemented in a C++ object-oriented platform [39] and used to solve two benchmarks to show its performance along with its flexibility.

4. Problem statement

The modelling of retinal vascularization can be stated in a more general and abstract manner as follows.

For a given tree-like target structure image (such as retinal blood vessel), the growth rules of an L-system with a set of general parameters that can provide a good approximate of the target structure are desired.

This problem can be formulated as the following optimization problem:

$$\left. \begin{aligned} &\min f(\text{Str}_t, \text{Str}_s) \\ &\text{subject to:} \\ &\text{Str}_s = G(\omega, p_{1\dots n}, M_{m,n}^b), \end{aligned} \right\} \tag{4.1}$$

where f calculates morphological differences between simulated structure (Str_s), resulting from L-system, and target structure (Str_t) obtained from fundus images. In fact, Str_s is obtained from the L-system supposed to model the tree-like structure of the retinal vasculature. Considering M_m as a sequence of L-system parametric symbols defined as below, equation (4.2) represents such a model ($G(\omega, p_{1\dots n}, M_{m,n}^b)$) in its general form:

$$\left. \begin{aligned} &\omega: T'_0(z_0, \theta_0) \\ &p_{1\dots n}: T'_{1,1\dots n}(Z, \Theta) \rightarrow M_{m,n}^0[M_{m,n}^1][M_{m,n}^2] \\ &M_{m,n}^b = (D^b(g^b(Z, \Theta))T^b(f^b(Z, \Theta)))_{1\dots m,1\dots n} \end{aligned} \right\} \tag{4.2}$$

where

$$\begin{aligned} T &\in \{F, A, B, \emptyset\}, \\ T' &\in \{F, A, B\}, \\ D &\in \{+, -, \emptyset\}, \\ g(Z, \Theta), f(Z, \Theta) &: \text{functions as the rules parameters} \\ z_0, \theta_0 &\in \mathbb{R}, \end{aligned}$$

and

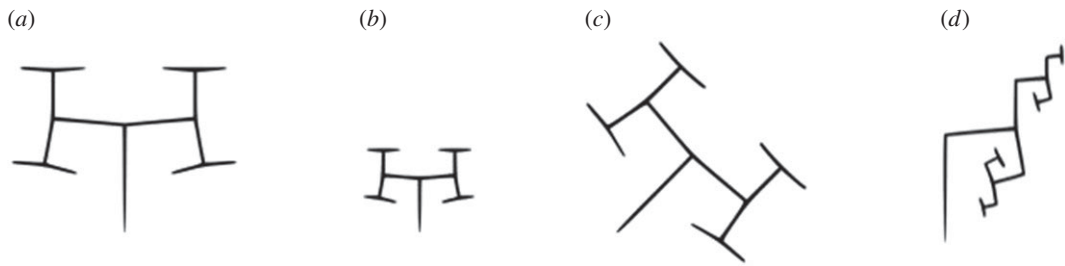


Figure 3. Failure of pixel-by-pixel comparison of (a) $A(100,90^\circ)$ with (b) $A(50,90^\circ)$ and (c) $A(100,45^\circ)$ having the same rules defined in example 2.1 and with (d) $A(100,90^\circ)$ which has a different rule.

in which ω is axiom, $p_{1\dots n}$ is rule set, and b , n , and m denote the number of blocks, rules, and alphabets, respectively. $g(Z, \Theta)$ and $f(Z, \Theta)$ are two general functions that specify the relationship between predecessor and successor parameters. To have a meaningful L-system, axiom and predecessor alphabet (T') could not be null (\emptyset). So, this is why T' was considered as the superset of T which shows successor alphabet.

The fitness function and decision variables are both further explained in §§4.1 and 4.2, respectively.

4.1. Fitness function

Due to the complexity of parametric L-system inverse problem, definition of an appropriate objective function representing the difference between the modelled structure and the reference retinal vascular image is very important. Here, challenges and weaknesses of various objective functions used in L-system inverse problem modelling are discussed and then a new hybrid objective function is proposed.

4.1.1. Pixel-by-pixel comparison

Kókai *et al.* [33] employed pixel-by-pixel comparison as the fitness function to convert image data to an L-system string. Through an illustrative example, it is shown that this fitness function is not useful to drive parametric L-system rewriting rules. Consider the L-system introduced in example 2.1 (figure 3a). As is shown in figure 3, changing axiom parameters while keeping the same rules, which leads to the same structure with different scale (figure 3b) and orientation (figure 3c), pixel-by-pixel comparison gives inappropriate fitness scores to another structure with a different set of rules provided in figure 3d.

Figure 3 shows that the main problem with the pixel-by-pixel comparison is that calculated scores do not provide an insight into morphological similarity between two structures.

4.1.2. Image feature matching

Feature matching is one of the most important methodologies to find an object in image data. It is based on detecting important points, called keypoints, in the object template and discovering information about surrounding area of those keypoints, called keypoint description. Descriptors, which contain keypoints description, are compared and matched with target images using some notation of similarity such as Euclidean or Hamming distance of descriptors. So, keypoint detection, description and matching are three main steps to detect object or structure in the image data [40].

Several keypoint detection and description techniques were developed during the past decade [41]. These methods include binary or real-valued descriptors. Many comparative studies were conducted on these methods [42]. Techniques involving binary descriptors require shorter computation times but provide less accurate results. In contrast, methods based on real-value descriptors like scale-invariant feature transform provide very accurate results but with much higher computation times for keypoint detection and description. In this work, the former techniques were selected owing to the faster speed of calculations. Accelerated Segment Test (FAST) [43] and Binary Robust Independent Elementary Features (BRIEF) [44] are well-known keypoint detection and description algorithms respectively. Oriented FAST and rotated BRIEF (ORB) is a keypoint detection and description approach that is a combination of modified FAST and BRIEF algorithms. ORB is rotation and scale invariant. Also, it is robust to data noise and affine transformations [45]. In this study, ORB was employed in both detection and description steps because of its appropriate computational speed and accuracy. Also, owing to its good speed and efficiency for small databases, brute force matcher algorithm was used to match keypoints in this work using Hamming distance calculation in OpenCV library in C++ [46].

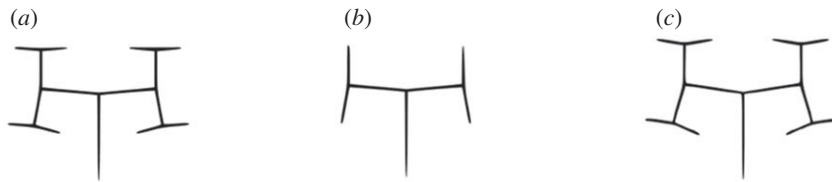


Figure 4. Failure of feature matching in L-systems with the same rules defined in example 2.1 and different parameters: (a) 3 growth cycles and $\delta = 85^\circ$, (b) 2 growth cycles and $\delta = 85^\circ$ and (c) 3 growth cycles and $\delta = 82^\circ$.

Hamming distance is based on executing XOR operations on binary codes. Brute force matcher can compare keypoint descriptors of two images with Hamming distance to correlate the closest ones. Good matches should have a lower distance than a ‘threshold’ which depends on algorithm and problem. Although feature matching can be useful to compare two images, it is not adequate by itself to solve L-system inverse problem.

Example 4.1. To illustrate one of the situations where feature matching fails, consider example 2.1 with different parameters shown in figure 4.

Owing to self-similarity of the structure defined in example 2.1, structure shown in figure 4b can be found three times in the one represented in figure 4a with different scales. Also, the only difference between structures shown in figure 4a,c is bifurcation angle that is negligible. Considering figure 4a as the target structure, figure 4c can be an acceptable solution, but figure 4b has a better similarity rank that is not good for optimization purposes.

4.1.3. L-system grammar matching

As previously noted, there are algorithms which have been proposed to obtain non-parametric L-system rules, from a string of non-parametric L-system symbols [38]. So, a potential fitness function for proposed L-system inverse problem could be obtained based on the comparison of converted image into an L-system string (target string) and the string generated by GA (final string).

Every graphical structure can be converted to a final string with different algorithms [33,47,48] whose graphical interpretation results in the given appropriate structure. However, none of these algorithms can solve L-system inverse problem to obtain parametric growth rules. Farooq *et al.* [47] showed that complicated tree-like structures could be converted to a final parametric string by a multilayered architecture of GA. They also examined the impact of different GA parameters, selection methods and the objective functions used as their calculated fitness function in GA on their algorithm efficiency.

Chuai-Aree *et al.* [48] developed a straightforward algorithm to achieve parametric L-system final string. This algorithm starts with reading the input image for two dimensions or sliced images for three dimensions. Anisotropic diffusion filtering is applied as the preprocessing step. Either region or volume growing technique is conducted by the set of given initial starting points. Thinning and skeletonization are the next steps respectively. At the end, L-system final string is generated and ready for further uses [48]. In this work, this straightforward algorithm was used to convert a two-dimensional target structure to a target string.

Although XOR comparison of target string against the string generated by GA can be helpful as the fitness function, it will fail in some situations in parametric L-systems. For example, consider the following two simple L-system strings:

$$\left. \begin{array}{l} a. F(100) \\ b. F(25) + (0)F(25) + (0)F(25) + (0)F(25). \end{array} \right\} \quad (4.3)$$

A simple line with a length of 100 is the graphical interpretation of both strings, but XOR comparison of alphabets of two strings leads to an inappropriate score. Accordingly, an inclusion of both graphical and grammatical criteria must be considered to define a proper fitness function.

4.1.4. Hybrid fitness function

In this work, a combination of feature, grammar and pixel-by-pixel matching was used to calculate the difference score between two structures. In order to have a more accurate comparison, each structure was split into its components. Nested brackets which represent branches in L-system were considered

Table 2. Mean Hamming and grammar distances between components shown in figure 5.

mean Hamming distance		grammar distance		pixel-by-pixel comparison	
$\hat{g}_H(C_a^1, C_b^1) = 0.33$	$\hat{g}_H(C_a^1, C_c^1) = 0.36$	$\hat{g}_G(C_a^1, C_b^1) = 0.56$	$\hat{g}_G(C_a^1, C_c^1) = 0$	$\hat{g}_P(C_a^1, C_b^1) = 0.23$	$\hat{g}_P(C_a^1, C_c^1) = 0.54$
$\hat{g}_H(C_a^2, C_b^2) = 0.25$	$\hat{g}_H(C_a^2, C_c^2) = 0.33$	$\hat{g}_G(C_a^2, C_b^2) = 0.64$	$\hat{g}_G(C_a^2, C_c^2) = 0$	$\hat{g}_P(C_a^2, C_b^2) = 0.28$	$\hat{g}_P(C_a^2, C_c^2) = 0.35$
$\hat{g}_H(C_a^3, C_b^3) = 0.22$	$\hat{g}_H(C_a^3, C_c^3) = 0.36$	$\hat{g}_G(C_a^3, C_b^3) = 0.89$	$\hat{g}_G(C_a^3, C_c^3) = 0$	$\hat{g}_P(C_a^3, C_b^3) = 0.4$	$\hat{g}_P(C_a^3, C_c^3) = 0.29$

as components, and ‘ η ’ is the number of branches. So the comparison is made between the target and simulated structures with the feature, grammar and pixel-by-pixel matching in local and global manners. Fitness function was defined as

$$f = \sum_{i=1}^{\eta} g_{H,i}(\text{str}_t, \text{str}_s) + \sum_{i=1}^{\eta} g_{G,i}(\text{str}_t, \text{str}_s) + \sum_{i=1}^{\eta} g_{P,i}(\text{str}_t, \text{str}_s), \quad (4.4)$$

where $g_H(\text{str}_t, \text{str}_s)$ calculates mean Hamming distance between matched keypoints of the target and simulated structures which were detected and described with ORB algorithm. $g_G(\text{str}_t, \text{str}_s)$ computes grammatical differences between target and simulated structures as was mentioned in the previous section. Pixel-by-pixel comparison is also conducted by $g_P(\text{str}_t, \text{str}_s)$. OpenCV C++ library [46] was used to implement XOR algorithm for pixel-by-pixel comparison and ORB algorithm for mean Hamming distance calculation between matched keypoints. Also, bitwise XOR operation for grammar comparison was implemented in our C++ platform [39].

Terms of the fitness function were normalized to lie in the range of [0,1]. The normalization was conducted as follows:

$$\left. \begin{aligned} \hat{g}_H(\text{str}_t, \text{str}_s) &= \frac{g_H(\text{str}_t, \text{str}_s)}{\tau}, \\ \hat{g}_G(\text{str}_t, \text{str}_s) &= \frac{g_G(\text{str}_t, \text{str}_s)}{\alpha}, \\ \hat{g}_P(\text{str}_t, \text{str}_s) &= \frac{g_P(\text{str}_t, \text{str}_s)}{\nu}, \end{aligned} \right\} \quad (4.5)$$

and

in which τ is Hamming distance threshold, α is the maximum length of L-system string allowed in the algorithm, and ν is the summation of non-white pixels in target and simulated images as the maximum possible pixel-by-pixel comparison difference between them. Fitness function was modified to

$$\left. \begin{aligned} f &= \sum_{i=1}^{\eta} \omega_1 \cdot \hat{g}_{H,i}(\text{str}_t, \text{str}_s) + \sum_{i=1}^{\eta} \omega_2 \cdot \hat{g}_{G,i}(\text{str}_t, \text{str}_s) + \sum_{i=1}^{\eta} \omega_3 \cdot \hat{g}_{P,i}(\text{str}_t, \text{str}_s) \\ \omega_1 + \omega_2 + \omega_3 &= 1, \end{aligned} \right\} \quad (4.6)$$

where ω_1 , ω_2 and ω_3 are weight factors that should be defined properly for each case. The weight factors can affect optimization convergence and run time. In this work, weight factors were obtained via a trial and error procedure, but their optimum values can be found through another optimization process. The following illustrative example is presented to show the calculation procedure.

Example 4.2. For calculation of the fitness function between three structures introduced in example 4.1, the first step is splitting them into their components. Owing to the symmetry of the structures, just three components out of seven were selected to calculate the fitness function (table 2).

The threshold (τ) was considered 40 as mentioned by Khan *et al.* [42]. α was assumed as the length of target string (figure 5a). Furthermore, ω_1 , ω_2 and ω_3 were all assumed to be 1/3.

Considering all components:

$$\left. \begin{aligned} f_{a,b} &= 3.17 \\ f_{a,c} &= 1.62, \end{aligned} \right\} \quad (4.7)$$

and

where $f_{a,b}$ and $f_{a,c}$ are differences between figure 5a,b, and figure 5a-c according to equation (4.6). Equation (4.7) shows that figure 5a and c have lower distance than figure 5a and b with this hybrid fitness function and proposed weight factors. In this work, if there is not a good match between two structures or there

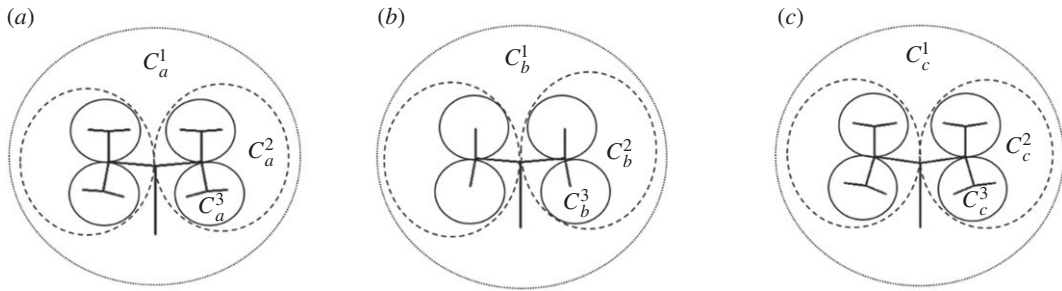


Figure 5. Components of L-systems defined in example 4.1: (a) 3 growth cycles and $\delta = 85^\circ$, (b) 2 growth cycles and $\delta = 85^\circ$ and (c) 3 growth cycles and $\delta = 82^\circ$.

Table 3. Decision variables.

decision variable	data structure
m, n	integer number
z_0, θ_0	real number
T, T', D	character
$g(Z, \Theta), f(Z, \Theta)$	parameter functions including characters and numbers

is no branch in a zone, mean Hamming distance was considered equal to the threshold. So, in example 4.2 mean Hamming distance between C_a^3 and C_b^3 is equal to 1 instead of 0.22 in our main algorithm. The reason is that it is not a good idea to match a single line to a linear structure.

4.2. Decision variables

As was mentioned in §4.1, most of the tree-like structures can be derived from equation (4.2). Due to the generality of equation (4.2), it was used to simulate retinal vasculature structure. All possible decision variables based on equation (4.2) are presented in table 3.

Here, we are also looking for $f(Z, \Theta)$ and $g(Z, \Theta)$ which are two unknown functions that show the relationship between length and angle of predecessor and successor, respectively. $f(Z, \Theta)$ and $g(Z, \Theta)$ are called parameter functions.

Due to different data structures, we have a MINLP problem. In this work, GA was employed to obtain the best L-system rules and parameters for a given structure.

5. Genetic algorithm

GAs are among the most powerful evolutionary algorithms that have been well addressed in the literature [49]. Defining an appropriate genetic representation, operators, and fitness function are the main steps to solve an optimization problem using GA. In the present work, an open source C++ framework called GALib [50] was employed to solve L-system inverse problem. The fitness function and its corresponding decision variables were described in the previous section.

As mentioned earlier, our optimization problem consists of various types of decision variables including character, integer and real data types. In GALib, GAAlleleSet is a container that can involve enumerated, bounded, or bounded with discretization sets of data types [50]. So, a composite chromosome or genome composed of GAAlleleSets of int, real and char data types was employed as the genetic representation (i.e. genotypes) of our decision variables (i.e. phenotypes). It should be noted that in order to avoid getting physically meaningless genotypes (i.e. those for which there are no physically meaningful decision variables), each AlleleSet has its own genetic operators in GALib and this is one of the great features of GALib that we have used. In this work, steady-state GA was used for optimization. In this algorithm, a temporary population of individuals was created and added to the previous population. Then, the worst individuals were removed based on roulette wheel

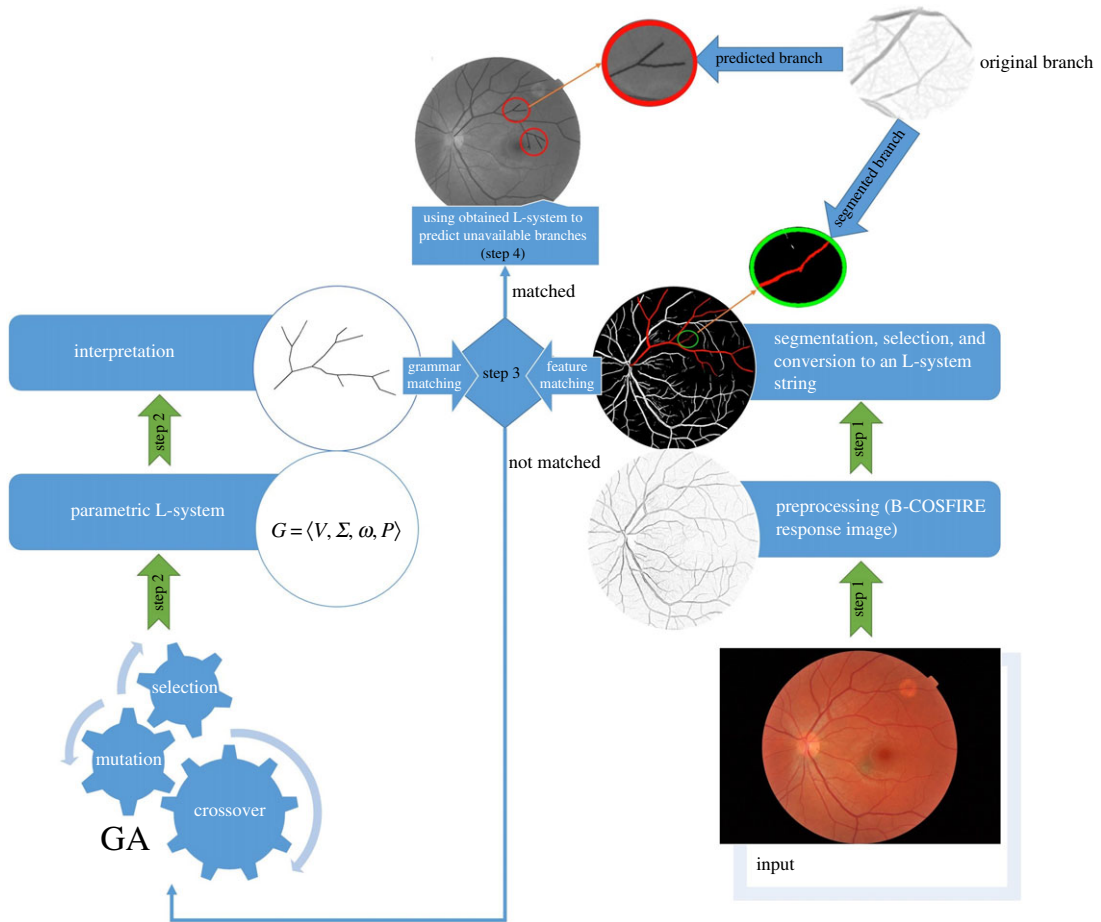


Figure 6. The framework flowchart.

selection scheme to achieve a population with the original size. UniformInitializer, FlipMutator and UniformCrossover defined in GALib [50] were used as the genetic operators.

6. Method

Figure 6 shows the flowchart of the procedure according to which various tentative L-system was used to obtain its corresponding retinal vascular structure and its fitness function. In the first step, target image should be preprocessed and converted to one L-system string. Automatic segmentation of blood vessel branches based on Combination of Shifted Filter Responses (COSFIRE), proposed by Azzopardi *et al.* [51], is used to extract blood vessels from original fundus image. Extracted branches obtained from segmentation are then converted to an L-system string using algorithm proposed by Chuai-Aree *et al.* [48]. The obtained L-system string is then compared against the one obtained by simulated structure resulting from GA in the second step, as mentioned in §4. Parameters in each growth cycle are calculated with simple linear functions. In the third step, the existence of general nonlinear functions for each parameter is investigated. Finally, unavailable branches would be predicted based on the obtained general nonlinear functions.

To illustrate the details of the proposed algorithm, it would be implemented step by step for the following case study.

Illustrative case study. In this case study, existence of simple rewriting rule for the structure presented by Kókai *et al.* [33] is investigated.

6.1. Step 1: Preprocessing and converting target structure to one L-system string

Owing to the lack of information about original image and inconsistency between given image scale and given L-system parameters, the image was preprocessed, and final L-system string was regenerated with

the algorithm presented by Chuai-Aree *et al.* [48].

$$\begin{aligned}
 &F(24)[+(35)F(16)[+(24)F(10) + (14)F(10)+ \\
 &(6)F(18)[+(47)F(25)][-(11)F(20)]][-(10)F(18) + (26)F(15)[-(6)F(10)+ \\
 &(34)F(16) + (32)F(12)][-(43)F(24)]]][F(32)[-(7)F(8) + (43)F(24)][-(14)F(26)]]]. \tag{6.1}
 \end{aligned}$$

6.2. Step 2: Solving L-system inverse problem

To solve L-system inverse problem for this structure, the problem was formulated in terms of equation (4.1). Equation (4.2) was simplified as follows:

$$\left. \begin{aligned}
 &\omega: A(z_0, \theta_0) \\
 &p_{1\dots n}: A(Z, \theta) \rightarrow M_1^0[M_1^1M_2^1][M_1^2M_2^2] \\
 &M_{m,n}^b = (D^b(g^b(Z, \theta))T^b(f^b(Z, \theta)))_{1\dots m,1\dots n}.
 \end{aligned} \right\} \tag{6.2}$$

Further, it was assumed that the axiom length and angle are equivalent to those for the main root of the target structure. In this case, the length of the main root and its angle relative to the axis X are 24 and 15, respectively. So, we have

$$\omega: A(24, 15). \tag{6.3}$$

On the other hand, it is assumed that $g^0(Z, \theta) = Z$. Based on described assumptions, $M_{m,n}^b$ are the only decision variables. The only remaining problem is to define $g(Z, \theta), f(Z, \theta)$ format.

In our algorithm, $g(Z, \theta), f(Z, \theta)$ was considered to be a simple linear function as $bZ + c\theta$. In every rewriting step, different constants would be considered by GA for these functions. Using this method, complicated functions can be approximated with linear ones at each growth cycle.

Following equations show one of the possible paths that could lead to the solution:

$$\left. \begin{aligned}
 &\text{Growth cycle 0} \\
 &A(24, 15).
 \end{aligned} \right\} \tag{6.4}$$

For the sake of brevity, the following rules resulting from optimization process have been suggested as candidate rules:

$$\begin{aligned}
 &A(Z, \theta) \rightarrow F(z)[+(g_1(Z, \theta))F(g_2(Z, \theta)) + (g_3(Z, \theta))A(g_4(Z, \theta))][-(g_5(Z, \theta))F(g_6(Z, \theta)) + \\
 &(g_7(Z, \theta))A'(g_8(Z, \theta))] \tag{6.5}
 \end{aligned}$$

and

$$\begin{aligned}
 &A'(Z, \theta) \rightarrow F(z)[+(g_1(Z, \theta))F(g_2(Z, \theta)) + (g_3(Z, \theta))F(g_4(Z, \theta))][-(g_5(Z, \theta))F(g_6(Z, \theta)) + \\
 &(g_7(Z, \theta))F(g_8(Z, \theta))]. \tag{6.6}
 \end{aligned}$$

The rewriting rules for both A and A' are the same, other than the fact that the one corresponding to A' is applied once because it does not contain any A or A' to be rewritten at the next growth cycle.

It was also assumed that length sizes are equal in each block, Hence

$$\left. \begin{aligned}
 &g_2(Z, \theta) = g_4(Z, \theta) \\
 &g_6(Z, \theta) = g_8(Z, \theta).
 \end{aligned} \right\} \tag{6.7}$$

and

As it was mentioned, linear functions would be replaced in the rule for each rewriting procedure. For example, GA result for the first growth cycle is

$$A(Z, \theta) \rightarrow F(z) \left[+ (2\theta)F \left(\frac{Z}{3} - 0.5 \right) + \left(\frac{\theta}{5} \right) A \left(\frac{Z}{3} - 0.5 \right) \right] \left[- \left(\frac{\theta}{3} \right) F \left(\frac{2Z}{3} \right) + \left(\frac{\theta}{3} \right) A' \left(\frac{2Z}{3} \right) \right]. \tag{6.8}$$

So,

$$\left. \begin{aligned}
 &\text{Growth cycle 1} \\
 &F(24)[+(30)F(7.5) + (3)A(7.5)][-(8)F(16) + (10)A'(16)].
 \end{aligned} \right\} \tag{6.9}$$

For the sake of brevity, only numerical parameters resulting from different linear functions will be reported afterward:

$$\left. \begin{aligned}
 &\text{Growth cycle 2} \\
 &F(24)[+(30)F(7.5) + (3)F(7.5)[+(35)F(20) + (10)A(20)][-(10)F(18)+ \\
 &(25)A'(18)]][-(8)F(16) + (10)F(16)[+(20)F(16) + (18)F(16)][-(22)F(12.5)+ \\
 &(5.5)F(12.5)]]
 \end{aligned} \right\} \tag{6.10}$$

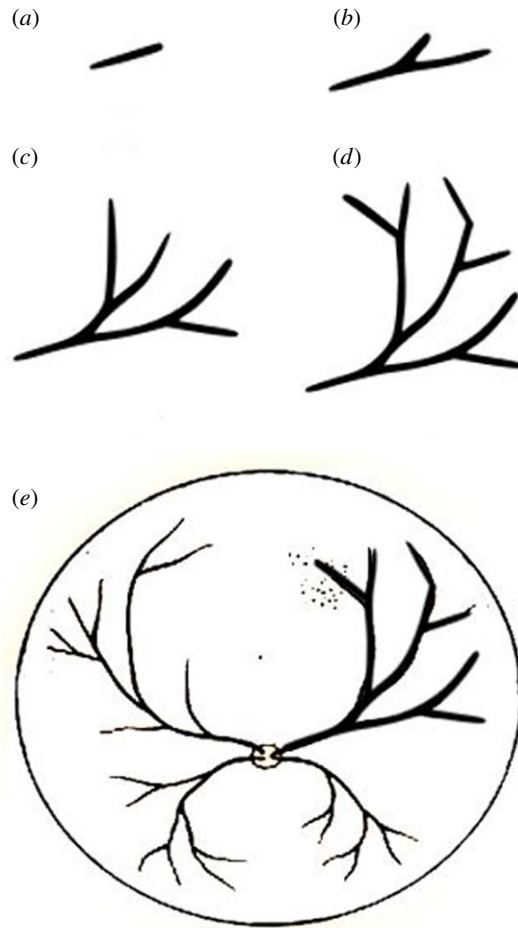


Figure 7. Graphical interpretation of (a) 0, (b) 1, (c) 2 and (d) 3 growth cycles of simulated structure and (e) graphical matching of simulated and target structures.

Table 4. Parameters of genetic algorithm.

population size	30
crossover rate	0.9
mutation rate	0.03
number of generation	500

and

$$\left. \begin{aligned}
 &\text{Growth cycle 3} \\
 &F(24)[+(30)F(7.5) + (3)F(7.5)][+(35)F(20) + (10)F(20)][+(50)F(12.5)+ \\
 &(4)A(12.5)][-(10)F(10) + (3)A'(10)]][-(10)F(18) + (25)F(18)][+(12)F(18)+ \\
 &(45)F(18)][-(45)F(10) + (0.01)F(10)]][-(8)F(16) + (10)F(16)][+(20)F(16)+ \\
 &(18)F(16)][-(22)F(12.5) + (5.5)F(12.5)].
 \end{aligned} \right\} \quad (6.11)$$

After three growth cycles, L-system final string will be compared against the one representing target structure with the hybrid fitness function. If the difference between simulated and target structures is acceptable, the optimization process is terminated. The values of GA parameters are given in table 4.

Final graphical string interpretation for each growth cycle and a comparison with the target image are shown in figure 7.

The result obtained by the proposed method and its corresponding reference one are compared in figure 7e. This figure shows that simulated and target structures are matched perfectly. To express the results quantitatively based on equation (4.4), the pixels that belong to the branch in the target image and that are simulated correctly are counted as true positives (TP), otherwise they are counted as false

Table 5. Quantitative results for illustrative case study.

accuracy	0.99
specificity	0.99
sensitivity	0.68
$\hat{g}_P(\text{str}_t, \text{str}_s)$	0.47
$\hat{g}_H(\text{str}_t, \text{str}_s)$	0.11
$\hat{g}_G(\text{str}_t, \text{str}_s)$	0.22

Table 6. Input–output data related to $g_4(Z, \Theta)$.

$Z_{\text{predecessor}}$		$\Theta_{\text{predecessor}}^a$		$Z_{\text{successor}}(L)$	
(Z)	(\bar{Z})	(degree)	(radian)	(Z)	(\bar{Z})
24	0.28	15	0.26	7.5	0.12
7.5	0.12	48	0.84	20	0.55
20	0.55	93	1.62	12.5	0.44
16	0.34	17	0.30	16	0.87
18	0.59	63	1.10	18	2.21

^aA line angle is defined as its angle relative to the axis X .

negatives (FN). The pixels that belong to the background and that are simulated as branch are counted as false positives (FP), otherwise they are counted as true negatives (TN). Accuracy, sensitivity and specificity are well-known metrics for pixel-by-pixel comparison [14] defined as follows:

$$\left. \begin{aligned} \text{Acc} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \\ \text{Se} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{Sp} &= \frac{\text{TN}}{\text{TN} + \text{FP}} \end{aligned} \right\} \quad (6.12)$$

and

The values of accuracy, sensitivity, specificity, $\hat{g}_H(\text{str}_t, \text{str}_s)$, $\hat{g}_G(\text{str}_t, \text{str}_s)$, and $\hat{g}_P(\text{str}_t, \text{str}_s)$, resulting from comparison of target and simulated branches, are provided in table 5.

In the next step, it is tried to find general functions for each g_1 to g_8 to reproduce the same results generated by linear functions in rewriting process. This is done by finding a correlation between the lengths and angles of the predecessor and successor.

6.3. Step 3: Searching for general parameter functions

In correlation step, angles were described in radians. Also, \bar{Z} was defined as a dimensionless variable:

$$\bar{Z} = \frac{Z}{L_m - L_0}, \quad (6.13)$$

where Z is the length of the line, L_m is the distance between initiation point and the farthest point in the structure and L_0 is the distance between initiation point and the centre of the line.

All inputs and outputs related to each of the functions $g_{1..8}$ in equation (6.5) were collected similar to what is shown in table 6 for $g_4(Z, \Theta)$. Nonlinear regression was used to find and fit general functions.

Correlation information for $g_4(\bar{Z}, \Theta)$ is shown in table 7.

Correlations for all other parameter functions are shown in table 8. As the result, the structure presented by Kókai *et al.* [33] can be reproduced with the following simple rule:

$$\left. \begin{aligned} \omega: & A(24, 15) \\ p_1: & A(Z, \Theta) \rightarrow F(z)[+(g_1(Z, \Theta))F(g_2(Z, \Theta)) + (g_3(Z, \Theta))A(g_4(Z, \Theta))] \\ & \quad [- (g_5(Z, \Theta))F(g_6(Z, \Theta)) + (g_7(Z, \Theta))A'(g_8(Z, \Theta))]. \end{aligned} \right\} \quad (6.14)$$

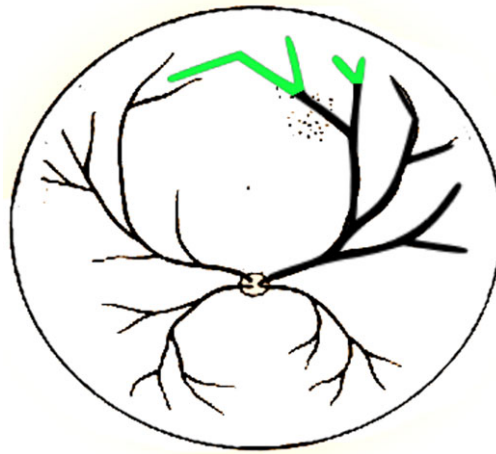


Figure 8. Predicted branches for illustrative case study using suggested general functions.

Table 7. Regression information for $g_4(Z, \Theta)$.

$g_4(\bar{Z}, \Theta): a + b\bar{Z} + c \ln(\Theta) + d \ln(\Theta)^2$		$R^2 = 0.99$	
variable	value	standard error	Prob(t)
a	-0.42	7×10^{-3}	0.010
b	4.93	1.7×10^{-2}	0.002
c	-2.61	1.1×10^{-2}	0.002
d	-2.41	9×10^{-3}	0.002

Table 8. Coefficients of functions presented in equation (6.13).

	$g(\bar{Z}, \Theta): a + b\bar{Z} + c \ln(\Theta) + d \ln(\Theta)^2$				R^2
	a	b	c	d	
$g_1(\bar{Z}, \Theta)$	0.94 ± 0.03	-1.41 ± 0.08	1.06 ± 0.05	0.79 ± 0.04	0.99
$g_3(\bar{Z}, \Theta)$	-0.20 ± 0.02	1.85 ± 0.06	-1.06 ± 0.03	-0.94 ± 0.03	0.99
$g_5(\bar{Z}, \Theta)$	-0.168 ± 0.001	1.771 ± 0.004	-0.883 ± 0.002	-0.763 ± 0.002	0.99
$g_7(\bar{Z}, \Theta)$	0.562 ± 0.005	-0.96 ± 0.01	0.056 ± 0.008	-0.044 ± 0.007	0.99
$g_8(\bar{Z}, \Theta)$	0.387 ± 0.002	1.100 ± 0.005	-0.504 ± 0.003	-0.554 ± 0.002	0.99

It should be noted that based on equation (6.7), $g_2(Z, \Theta) = g_4(Z, \Theta)$ and $g_6(Z, \Theta) = g_8(Z, \Theta)$.

6.4. Step 4: Predicting unavailable branches with suggested general functions

Applying suggested general functions, one more rewriting procedure was conducted on equation (6.14). Prediction results are shown in figure 8 with green lines.

7. Results and discussion

There are lots of Web-based fundus image galleries obtained through various ophthalmic imaging systems [52–54]. After going through the details of the proposed method for the development of retinal vascular structure with an illustrative case study, to show the accuracy and flexibility of the proposed method, it is used to develop the retina vascular structure based on practical images obtained from various imaging techniques.

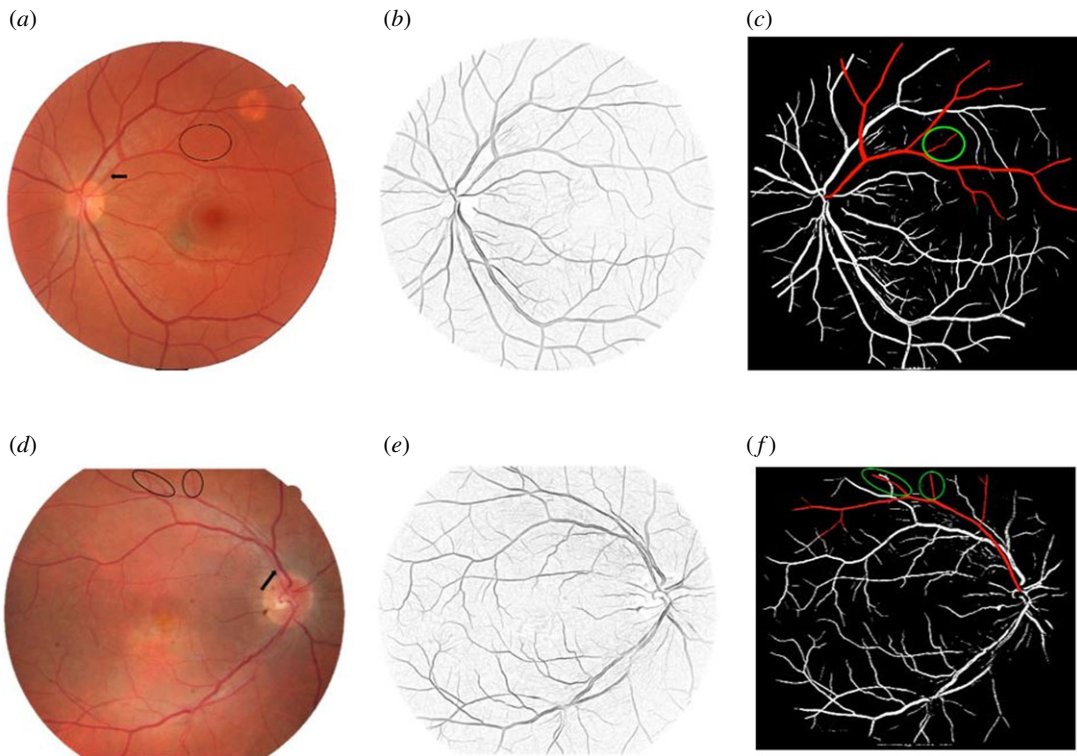


Figure 9. Thirty-year-old woman's left eye: (a) original fundus image, (b) B-COSFIRE filter response image, (c) segmented retinal blood vessels. Forty-nine-year-old man's right eye: (d) original fundus image [54], (e) B-COSFIRE filter response image, (f) segmented retinal blood vessels.

To investigate the similarity of growth rules of retinal vasculature in different eyes, 108 fundus images were analysed by the algorithm presented in this work. For instance, two fundus images along with their preprocessed and segmented images are shown in figure 9.

Branches shown by arrows in figure 9a,d were found to follow the same growth rule. Due to the lack of information about the circled branches, they would be predicted in the next section. For the sake of brevity, simplified branching rule will be presented in this section.

The proposed algorithm in the previous section was implemented, and the axioms and rewriting rules which are shown by equation (7.1) were obtained:

$$\left. \begin{aligned}
 \omega_{\text{for (a) in figure 8}} &: F(60, 44.5) + (0)A(60, 44.5) \\
 \omega_{\text{for (b) in figure 8}} &: F(80, 123) + (0)A(80, 123) \\
 p_1: & A(Z, \theta) \rightarrow F(z)[\pm(g_1(Z, \theta))F(f_1(Z, \theta)) \mp (g_2(Z, \theta))B(f_2(Z, \theta))] \\
 & \quad [\mp(g_3(Z, \theta))F(f_3(Z, \theta)) \pm (g_4(Z, \theta))A(f_4(Z, \theta))] \\
 p_2: & B(Z, \theta) \rightarrow F(z)[\pm(g_5(Z, \theta))F(f_5(Z, \theta)) \mp (g_6(Z, \theta))F(f_6(Z, \theta))] \\
 & \quad [\mp(g_7(Z, \theta))F(f_7(Z, \theta)) \pm (g_8(Z, \theta))F(f_8(Z, \theta))],
 \end{aligned} \right\} \quad (7.1)$$

where the first axiom and first rule correspond to the L-system representing figure 9a and the second ones reflects the L-system representing figure 9b. The following parametric strings are the final best matched strings resulting from the proposed algorithm after four L-system growth cycles:

$$\left. \begin{aligned}
 & \text{For figure 9a} \\
 & F(60) + (0)F(60)[+(58)F(60) - (23)F(65)[+(42.5)F(48) - (10)F(60)][-(22)F(65) + \\
 & (5)F(80)]][-(30)F(21) + (1)F(77)[+(27)F(80) - (-7)F(81)[+(24.5)F(37) - \\
 & (34)F(56)]][-(35)F(60) + (1)F(75)]][-(33)F(34) + (9)F(8)[+(X)F(X) - \\
 & (X)F(X)[+(X)F(X) - (X)F(X)][-(X)F(X) + (X)F(X)]][-(30)F(36) + \\
 & (30)F(52)[+(10)F(63) - (6.5)F(105)[+(26)F(45) - (1.5)F(40)][-(55)F(62) + \\
 & (40)F(80)]][-(55)F(57) + (37)F(6)[+(20)F(18) - (7) + B(30)][-(45)F(31.5) + \\
 & (-3)A(27)]]
 \end{aligned} \right\} \quad (7.2)$$

Table 9. Instances of general functions for two cases shown in figure 9a,b.

$f_1(\bar{Z}, \Theta) = a_1 b_1^{\Theta} \bar{Z}^{c_1}$					
	a_1	b_1	c_1	d_1	R_1^2
figure 9a	1.2 ± 0.6	0.3 ± 0.1	-1.0 ± 0.2	—	0.99
figure 9b	1.4 ± 0.2	0.05 ± 0.02	-1.9 ± 0.3	—	0.99
$g_2(\bar{Z}, \Theta) = a_2 + \frac{b_2}{\Theta} + \frac{c_2}{\ln(\Theta)^2} + \frac{d_2}{\ln(\bar{Z})}$					
	a_2	b_2	c_2	d_2	R_2^2
figure 9a	5.1 ± 0.6	-5 ± 1	-0.015 ± 0.007	-1.3 ± 0.4	0.92
figure 9b	4.90 ± 0.04	-4.9 ± 0.3	-0.030 ± 0.008	-1.9 ± 0.2	0.98

and

For figure 9b

$$\left. \begin{aligned}
 &F(80) + (0)F(80)[-(44.5)F(50) + (5)F(60)[-(7)F(28) + (17)F(30)] + (40)F(45) - \\
 &(12)F(45)] + (30)F(47) - (4.5)F(85)[-(X)F(X) + (X)F(X)[-(X)F(X) + \\
 &(X)F(X)] + (X)F(X) - (X)F(X)] + (21)F(40) - (3)F(35)[-(X)F(X) + (X)F(X)[-(X)F(X) + \\
 &(X)F(X)] + (X)F(X) - (X)F(X)] + (24)F(77.5) - (5.5)F(100)[-(9)F(55) + \\
 &(-5)F(20)[-(30)F(40) + (-15)F(60)] + (18)F(51) - (2.5)F(60)] + (83)F(34) - \\
 &(1)F(2)[-(55)F(11) + (0)B(10)] + (32)F(14) - (0)A(9)] + \dots
 \end{aligned} \right\} \quad (7.3)$$

The parameters shown by X correspond to the circled branches of figure 9 which should be predicted due to the lack of information.

Values of various parameters in equations (7.2) and (7.3) have all been correlated with a set of general functions representing g_i and f_i ($i = 1, 2, \dots, 8$), respectively. The obtained correlating functions for both cases (i.e. those represented by figure 9a,b) were compared to investigate the generalizability of the obtained correlating functions. To obtain general functions, Θ was considered as the angle relative to the axis Y described in radians. Also, two dimensionless variables were defined in the form of

$$\left. \begin{aligned}
 \bar{Z}_{\text{predecessor}} &= \frac{Z_{\text{predecessor}}}{L_0} \\
 \bar{Z}_{\text{successor}} &= \frac{Z_{\text{successor}}}{Z_{\text{predecessor}}}
 \end{aligned} \right\} \quad (7.4)$$

and

where $Z_{\text{predecessor}}$ is the length of predecessor, $Z_{\text{successor}}$ is the length of successor and L_0 is the distance between initiation point and the centre of predecessor.

The simulations and analysis of the existing cases in our database based on the mentioned procedure led to an astonishing result. Not only do the formats of all general functions describing the lengths ($f_{1..8}$) and angles ($g_{1..8}$) share the same formats for each case but also the formats and even the coefficients of general functions for couple of cases particularly those shown in figure 9a,b are the same. Table 9 presents this similarity for two typical functions (i.e. f_1 and g_2) for cases shown in figure 9a,b.

To show the correlation between $(\bar{Z}_{\text{predecessor}}, \Theta_{\text{predecessor}})$ and $(\bar{Z}_{\text{successor}}, \Theta_{\text{successor}})$ of all ten similar cases [52] along with figure 9a,b, coefficients of obtained general functions have been reported in table 10. Regression coefficients can be updated for each case such as those shown in table 9.

The general functions were used to predict branches shown in circled zones in figure 9. Also with the help of these functions, one more growth cycle was conducted on equations (7.2) and (7.3) to predict next generation branches that could not be recognized due to image resolution. Simulated and predicted branches are shown in figure 10.

In order to compare circled branches in figure 9b with predicted ones, wide-field fundus image of figure 9b [55] is provided in figure 10b. Black lines represent simulated structures. Green lines show predicted branches based on suggested general functions. Accuracy, sensitivity, specificity and fitness function values for simulated and predicted branches in both cases are given in table 11. As mentioned in §6, COSFIRE algorithm was used to extract blood vessels in the first step. Performance of the COSFIRE algorithm which was used in the first step of proposed algorithm affects the results of simulation performance with regard to accuracy, specificity and selectivity metrics. As shown in table 11, the

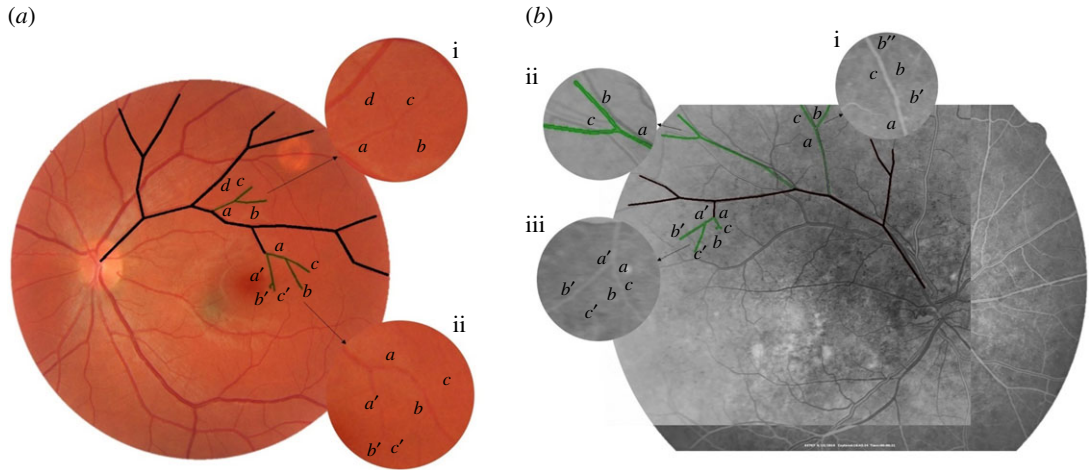


Figure 10. Comparison between simulated and predicted branches with target structures presented in (a) figure 9a and (b) figure 9d.

Table 10. Coefficients of functions presented in equation (6.14).

	a_i	b_i	c_i	d_i	R_f^2
$f_i(\bar{Z}, \Theta) = a_i b_i \Theta \bar{Z}^{c_i}$					
$f_1(\bar{Z}, \Theta)$	1.0 ± 0.1	0.24 ± 0.05	-1.1 ± 0.1	—	0.98
$f_2(\bar{Z}, \Theta)$	2.6 ± 0.8	0.09 ± 0.05	-1.4 ± 0.3	—	0.89
$f_3(\bar{Z}, \Theta)$	0.7 ± 0.2	0.9 ± 0.2	-0.4 ± 0.1	—	0.93
$f_4(\bar{Z}, \Theta)$	2.1 ± 0.6	0.3 ± 0.1	-0.8 ± 0.2	—	0.85
$f_5(\bar{Z}, \Theta)$	0.15 ± 0.05	0.8 ± 0.2	-0.9 ± 0.2	—	0.90
$f_6(\bar{Z}, \Theta)$	0.15 ± 0.03	0.7 ± 0.3	-1.0 ± 0.2	—	0.92
$f_7(\bar{Z}, \Theta)$	0.24 ± 0.07	0.7 ± 0.2	-0.8 ± 0.2	—	0.88
$f_8(\bar{Z}, \Theta)$	0.31 ± 0.09	0.9 ± 0.2	-0.7 ± 0.1	—	0.89
$g_i(\bar{Z}, \Theta) = a_i + \frac{b_i}{\Theta} + \frac{c_i}{\ln(\Theta)^2} + \frac{d_i}{\ln(\bar{Z})}$					
$g_1(\bar{Z}, \Theta)$	3.6 ± 0.5	-4.5 ± 1.1	-0.022 ± 0.008	-1.2 ± 0.2	0.98
$g_2(\bar{Z}, \Theta)$	4.9 ± 0.1	-5.2 ± 0.6	-0.03 ± 0.01	-2 ± 0.5	0.91
$g_3(\bar{Z}, \Theta)$	3.3 ± 0.7	-4 ± 1	-0.05 ± 0.02	-2.3 ± 0.3	0.89
$g_4(\bar{Z}, \Theta)$	5.1 ± 0.1	-6.1 ± 0.1	-0.03 ± 0.01	-2.4 ± 0.3	0.87
$g_5(\bar{Z}, \Theta)$	2.5 ± 0.1	-0.95 ± 0.03	-0.013 ± 0.001	-0.4 ± 0.1	0.96
$g_6(\bar{Z}, \Theta)$	3.1 ± 0.1	-0.75 ± 0.05	-0.011 ± 0.002	-0.15 ± 0.08	0.87
$g_7(\bar{Z}, \Theta)$	2.4 ± 0.1	-0.98 ± 0.01	-0.015 ± 0.001	-0.71 ± 0.01	0.98
$g_8(\bar{Z}, \Theta)$	3.1 ± 0.1	-0.77 ± 0.03	-0.013 ± 0.001	-0.09 ± 0.05	0.85

accuracy and specificity are not changed significantly compared to the segmentation step (COSFIRE results). Sensitivities of the simulated and predicted results are far less than segmentation result due to the limitations imposed by obtained growth rules. However, unlike image processing methods, the proposed algorithm has the capability to obtain the structure growth rules, if any exist.

It can be seen from figure 10 and table 11 that simulated and target branches are matched perfectly. Prediction results for each zone are discussed below.

Figure 10a

- (i) It seems branches a , b and c are well predicted by general functions. It is clear from the image that branch d is related to another vessel, but the most of automated segmentation algorithms

Table 11. Quantitative results for figure 10.

	figure 10a			figure 10b		
	COSFIRE [51]	simulated	predicted	COSFIRE [51]	simulated	predicted
accuracy	0.98	0.91	0.93	0.98	0.90	0.93
specificity	0.99	0.95	0.96	0.99	0.95	0.97
sensitivity	0.92	0.57	0.67	0.90	0.51	0.56
$\hat{g}_p(\text{str}_t, \text{str}_s)$	—	0.58	0.50	—	0.63	0.58
$\hat{g}_H(\text{str}_t, \text{str}_s)$	—	0.21	0.13	—	0.33	0.16
$\hat{g}_G(\text{str}_t, \text{str}_s)$	—	0.31	0.48	—	0.38	0.53

will fail to distinguish between abc and d . As is shown, predicted branch does not contain branch d , so our method can be useful for these situations.

- (ii) Next generation of L-system defined with equation (7.2) is presented in this zone to test the ability of our method for the prediction of smaller vessels that cannot be recognized due to the lack of resolution. As can be seen in figure 10a(ii), it seems that a, b, a', b' and c' are predicted appropriately. Predicted branch c does not have equivalent in target image due to the lack of resolution or functional error.

Figure 10b

- (i) It seems that branches a and c are predicted well, but b does not have any equivalent in that position. Branch b is probably biased from b' or b'' in the target image due to functional error in prediction.
- (ii) In this zone, branch a has a little deviation from the target at the beginning and end. For this reason, b and c are biased from the real branches. Despite the deviation, the prediction is acceptable in this case.
- (iii) For the next generation of L-system defined with equation (7.3), it seems that all a, b, c, a' and b' are predicted well and c' does not have equivalent in target image due to the lack of resolution or functional error.

It should be noted that in spite of the accuracy of the correlated functions at each growth cycle, they cannot be used for couple of consecutive growth cycles due to the error propagation and amplification. In other words, the obtained correlated functions should be used merely for prediction purposes.

8. Conclusion

In this work, a novel framework for the construction of retinal vasculatures has been proposed. The proposed method is based on obtaining a genetically tuned parametric L-system for a tree-like structure such as retinal vasculature. Retinal blood vessel morphology can be used to identify conditions of patients suffering from various diseases including eye disease in particular. On the other hand, L-system as a rule-based method can be useful to predict a segment of a structure that is not accessible by image processing algorithms due to the lack of information if the creation of that structure obeys the rules. Obtaining these rules from an available structure is called L-system inverse problem. The performance of the proposed method has been shown by its application for the development of retinal blood vessels of two practical case studies. It was shown that retinal blood vessels related to different people might follow the same rule which can be used to predict inaccessible parts of them. Both simulated and predicted structures matched perfectly with their corresponding target images. Prediction results improved the accuracy, sensitivity and specificity of simulated structures compared with target image. The proposed method is efficient to obtain an approximation of complicated system behaviour such as retinal vascular growth rules.

It should also be mentioned that, in some situations, deviations from target structure were seen due to the modelling error or the lack of information. Rule similarity for a group of retinal blood vessel structures related to different people can be considered as a new idea for classification of these

structures. On the other hand, it is shown that searching for more accurate parameter functions based on vasculogenesis mechanism is necessary to have a better prediction.

The algorithm proposed in this work can be used to obtain parametric rules in different disease stages which affect retinal blood vessels for identification of changes in morphology of blood vessels. A richer database including both normal and abnormal fundus images with similar growth rules should be applied for further study on the role of different diseases in deviating from normal condition. Having the parametric L-system model in hand, deviation from normal width, length, and tortuosity of retinal blood vessels due to a disease such as retinopathy of prematurity [56] or diabetic retinopathy [57] can be investigated numerically.

Three-dimensional reconstruction of retinal blood vessel structure with the help of L-system and morphological properties of eye is aimed as the subject for future studies and extension of this work.

Ethics. This research was carried out based on retinal fundus images which are available in Web-based databases. The authors declare that it was not required to complete any ethical assessment prior to conducting this research.

Data accessibility. Our C++ code is deposited at GitHub [39]: <https://github.com/Aaghamir/LSysInvProb>. Retinal fundus images which have similar growth rules to the proposed cases of this study are available in STARE database [52] with following numbers: im0118, im0124, im0178, im0180, im0205, im0208, im0227, im0245, im0297 and im0355: <http://cecas.clemson.edu/~ahoover/stare>.

Authors' contributions. S.M.A.A., R.B.B. and M.A. designed the study as PhD project. S.M.A.A. collected all image data for analysis. S.M.A.A. and R.B.B. proposed a framework to solve inverse problem. S.M.A.A. prepared C++ code for the framework. All authors interpreted the results, wrote the manuscript and gave final approval for publication.

Competing interests. We declare we have no competing interests.

Funding. We received no funding for this study.

Acknowledgements. S.M.A.A. acknowledges Seyed Mohammad Aghamirmohammadali for his great advice on developing the C++ code.

References

- Carmeliet P. 2005 Angiogenesis in life, disease and medicine. *Nature* **438**, 932–936. (doi:10.1038/nature04478)
- Ambrosi D, Bussolino F, Preziosi L. 2005 A review of vasculogenesis models. *J. Theor. Med.* **6**, 1–19. (doi:10.1080/1027366042000327098)
- Gariano RF, Gardner TW. 2004 Retinal angiogenesis in development and disease. *Nature* **438**, 960–966. (doi:10.1038/nature04482)
- Joshi V. 2012 *Analysis of retinal vessel networks using quantitative descriptors of vascular morphology*. Iowa city, IA: The University of Iowa. See <http://ir.uiowa.edu/etd/3321/>.
- Carmeliet P. 2000 Mechanisms of angiogenesis and arteriogenesis. *Nat. Med.* **6**, 389–395. (doi:10.1038/74651)
- Chaplain MAJ, McDougall SR, Anderson ARA. 2006 Mathematical modeling of tumor-induced angiogenesis. *Annu. Rev. Biomed. Eng.* **8**, 233–257. (doi:10.1146/annurev.bioeng.8.061505.095807)
- Gödde R, Kurz H. 2001 Structural and biophysical simulation of angiogenesis and vascular remodeling. *Dev. Dyn.* **220**, 387–401. (doi:10.1002/dvdy.1118)
- Winder RJ, Morrow PJ, McRitchie IN, Bailie JR, Hart PM. 2009 Algorithms for digital image processing in diabetic retinopathy. *Comput. Med. Imaging Graph.* **33**, 608–622.
- Bernabeu MO *et al.* 2014 Computer simulations reveal complex distribution of haemodynamic forces in a mouse retina model of angiogenesis. *J. R. Soc. Interface* **11**, 20140543. (doi:10.1098/rsif.2014.0543)
- Franklin SW, Rajan SE. 2014 Computerized screening of diabetic retinopathy employing blood vessel segmentation in retinal images. *Biocybern. Biomed. Eng.* **34**, 117–124. (doi:10.1016/J.BBE.2014.01.004)
- Jooybar E, Abdekhodaie MJ, Farhadi F, Cheng Y-L. 2014 Computational modeling of drug distribution in the posterior segment of the eye: effects of device variables and positions. *Math. Biosci.* **255**, 11–20. (doi:10.1016/J.MBS.2014.06.008)
- Causin P, Guidoboni G, Malgaroli F, Sacco R, Harris A. 2016 Blood flow mechanics and oxygen transport and delivery in the retinal microcirculation: multiscale mathematical modeling and numerical simulation. *Biomech. Model. Mechanobiol.* **15**, 525–542. (doi:10.1007/s10237-015-0708-7)
- Aubert M, Chaplain MAJ, McDougall SR, Devlin A, Mitchell CA. 2011 A continuum mathematical model of the developing murine retinal vasculature. *Bull. Math. Biol.* **73**, 2430–2451.
- Fraz MM, Remagnino P, Hoppe A, Uyyanonvara B, Rudnicka AR, Owen CG, Barman SA. 2012 Blood vessel segmentation methodologies in retinal images—a survey. *Comput. Methods Programs Biomed.* **108**, 407–433. (doi:10.1016/j.cmpb.2012.03.009)
- Kirbas C, Quek F. 2004 A review of vessel extraction techniques and algorithms. *ACM Comput. Surv.* **36**, 81–121. (doi:10.1145/1031120.1031121)
- Scianna M, Bell CG, Preziosi L. 2013 A review of mathematical models for the formation of vascular networks. *J. Theor. Biol.* **333**, 174–209. (doi:10.1016/j.jtbi.2013.04.037)
- Heck T, Vaeyens M, Van Oosterwyck H. 2015 Computational models of sprouting angiogenesis and cell migration: towards multiscale mechanochemical models of angiogenesis. *Math. Model. Nat. Phenom.* **10**, 1–46. (doi:10.1051/mmnp/201510106)
- Behzadian MA, Bartoli M, El-Remessy AB, Al-Shabraway M, Platt DH, Liou GI, Caldwell RW, Caldwell RB. 2008 Cellular and molecular mechanisms of retinal angiogenesis. In *Retinal and choroidal angiogenesis* (ed. J Penn), pp. 1–39. Dordrecht, The Netherlands: Springer. (doi:10.1007/978-1-4020-6780-8_1)
- Peirce SM. 2008 Computational and mathematical modeling of angiogenesis. *Microcirculation* **15**, 739–751. (doi:10.1080/10739680802220331)
- Watson MG, McDougall SR, Chaplain MAJ, Devlin AH, Mitchell CA. 2012 Dynamics of angiogenesis during murine retinal development: a coupled *in vivo* and *in silico* study. *J. R. Soc. Interface* **9**, 2351–2364. (doi:10.1098/rsif.2012.0067)
- Kleinstreuer N, Dix D, Rountree M, Baker N, Sipes N, Reif D, Spencer R, Knudsen T. 2013 A computational model predicting disruption of blood vessel development. *PLoS Comput. Biol.* **9**, e1002996. (doi:10.1371/journal.pcbi.1002996)
- Witten Jr TA, Sander LM. 1981 Diffusion-limited aggregation, a kinetic critical phenomenon. *Phys. Rev. Lett.* **47**, 1400. (doi:10.1103/PhysRevLett.47.1400)
- Masters BR. 2004 Fractal analysis of the vascular tree in the human retina. *Annu. Rev. Biomed. Eng.* **6**, 427–452. (doi:10.1146/annurev.bioeng.6.040803.140100)
- Fleury V, Schwartz L. 2000 Modélisation of 3-D microvasculature by interlaced diffusion limited aggregation. *Fractals* **8**, 255–259.
- Family F, Masters BR, Platt DE. 1989 Fractal pattern formation in human retinal vessels. *Phys. D*

- Nonlinear Phenom.* **38**, 98–103. (doi:10.1016/0167-2789(89)90178-4)
26. Lindenmayer A. 1968 Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *J. Theor. Biol.* **18**, 280–299. (doi:10.1016/0022-5193(68)90079-9)
 27. Prusinkiewicz P, Lindenmayer A. 2012 *The algorithmic beauty of plants*. Berlin, Germany: Springer Science & Business Media.
 28. Zamir M. 2001 Arterial branching within the confines of fractal L-system formalism. *J. Gen. Physiol.* **118**, 267–276. (doi:10.1085/jgp.118.3.267)
 29. Murray CD. 1926 The physiological principle of minimum work: I. The vascular system and the cost of blood volume. *Proc. Natl Acad. Sci.* **12**, 207–214.
 30. Murray CD. 1926 The physiological principle of minimum work applied to the angle of branching of arteries. *J. Gen. Physiol.* **9**, 835–841. (doi:10.1085/jgp.9.6.835)
 31. Liu X, Liu H, Hao A, Zhao Q. 2010 Simulation of blood vessels for surgery simulators. In *2010 Int. Conf. on Machine Vision and Human-Machine Interface, Kaifeng, China, 24–25 April 2010*, pp. 377–380. (doi:10.1109/MVHI.2010.182)
 32. Hichem G, Malek J. 2012 Generation of new blood vessels in the human retina with L-system fractal construction. In *6th Int. Conf. on Sciences of Electronics, Technologies of Information and Telecommunications, Sousse, Tunisia, 21–24 March 2012*, pp. 287–292. (doi:10.1109/SETIT.2012.6481930)
 33. Kókai G, Tóth Z, Ványi R. 1999 Modelling blood vessels of the eye with parametric L-systems using evolutionary algorithms. In *Artificial intelligence in medicine* (eds W Horn, Y Shahar, G Lindberg, S Andreassen, J Wyatt), pp. 433–442. Berlin, Germany: Springer. (doi:10.1007/3-540-48720-4_48)
 34. Štáva O, Beneš B, Měch R, Aliaga DG, Křištof P. 2010 Inverse procedural modeling by automatic generation of L-systems. *Comput. Graph. Forum* **29**, 665–674. (doi:10.1111/j.1467-8659.2009.01636.x)
 35. Lindenmayer A. 1974 Adding continuous components to L-systems. In *L systems* (eds G Rozenberg, A Salomaa). Lecture Notes in Computer Science, vol. 15, pp. 53–68. Berlin, Germany: Springer. (doi:10.1007/3-540-06867-8_3)
 36. Prusinkiewicz P. 1986 Graphical Applications of L-Systems. In *Proc. on Graphics Interface 1986—Vision Interface 1986*, pp. 247–253. Vancouver, Canada: Canadian Information Processing Society. (doi:10.20380/GI1986.44)
 37. Koza JR. 1993 Discovery of rewrite rules in Lindenmayer systems and state transition rules in cellular automata via genetic programming. In *Symp. on Pattern Formation, SPF93, Claremont, CA, USA*, pp. 1–19.
 38. Santos E, Coelho RC. 2009 Obtaining L-systems rules from strings. In *3rd Southern Conf. on Computational Modeling, Rio Grande, Brazil, 23–25 November 2009*, pp. 143–149. (doi:10.1109/MCSUL.2009.21)
 39. Aghamirmohammadali SMA, Boozarjomehry R, Abdekhodaie MJ. 2017 L-system inverse problem solver C++ + code. See <https://github.com/Aaghamir/LSysInvProb> (accessed on 27 October 2017).
 40. Brahmabhatt, Samarth. 2012 Practical OpenCV. Apress.
 41. Roshanbin N, Miller J. 2016 A comparative study of the performance of local feature-based pattern recognition algorithms. *Patt. Anal. Appl.* **20**, 1–12. (doi:10.1007/s10044-016-0554-y)
 42. Khan N, McCane B, Mills S. 2015 Better than SIFT? *Mach. Vis. Appl.* **26**, 819–836. (doi:10.1007/s00138-015-0689-7)
 43. Rosten E, Drummond T. 2006 Machine learning for high speed corner detection. In *Computer vision: ECCV 2006* (eds A Leonardis, H Bischof, A Pinz). Lecture Notes in Computer Science, vol. 3951, pp. 430–443. Berlin, Germany: Springer. (doi:10.1007/11744023_34)
 44. Calonder M, Lepetit V, Strecha C, Fua P. 2010 BRIEF: binary robust independent elementary features. In *Computer vision: ECCV 2010* (eds K Daniilidis, P Maragos, N Paragios). Lecture Notes in Computer Science, vol. 6314, pp. 778–792. Berlin, Germany: Springer.
 45. Rublee E, Rabaud V, Konolige K, Bradski G. 2011 ORB: an efficient alternative to SIFT or SURF. In *2011 Int. Conf. on Computer Vision, Barcelona, Spain, 6–13 November 2011*, pp. 2564–2571. (doi:10.1109/ICCV.2011.6126544)
 46. Bradski G. 2000 The OpenCV Library. *Dr Dobbs J. Softw. Tools* **25**, 120–125. (doi:10.1111/0023-8333.50.s1.10)
 47. Farooq H, Zakaria MN, Hassan MF, Sulaiman S. 2009 An approach to derive parametric L-system using genetic algorithm. In *Visual informatics: bridging research and practice* (eds H Badioze Zaman, P Robinson, M Petrou, P Olivier, H Schröder, TK Shih). Lecture Notes in Computer Science, vol. 5857, pp. 455–466. Berlin, Germany: Springer. (doi:10.1007/978-3-642-05036-7_43)
 48. Chuai-Aree S, Jäger W, Bock HG, Siripant S. 2008 Inverse problem of Lindenmayer systems on branching structures. In *Modeling, simulation and optimization of complex processes* (eds HG Bock, E Kostina, HX Phu, R Rannacher), pp. 163–185. Berlin, Germany: Springer. (doi:10.1007/978-3-540-79409-7_11)
 49. Goldberg DE. 1989 *Genetic algorithms in search optimization and machine learning*, p. 102. Boston, MA: Addison Wesley Publishing.
 50. Wall M. 1996 GALib: A C++ library of genetic algorithm components. *Mech. Eng. Dep. Massachusetts*. **87**, 54.
 51. Azzopardi G, Strisciuglio N, Vento M, Petkov N. 2015 Trainable COSFIRE filters for vessel delineation with application to retinal images. *Med. Image Anal.* **19**, 46–57. (doi:10.1016/j.media.2014.08.002)
 52. Goldbaum M, Hoover A. 1975 STARE dataset website. Clemson University. See <http://cecas.clemson.edu/~ahover/stare/> (accessed on 9 June 2017).
 53. Staal JJ, Abramoff MD, Niemeijer M, Viergever MA, van Ginneken B. 2004 Digital retinal image for vessel extraction (DRIVE) database. Utrecht, The Netherlands: Image Science Institute, University Medical Centre, Utrecht. See <http://www.isi.uu.nl/Research/Databases/DRIVE/> (accessed on 9 June 2017).
 54. RetinaGallery. 2010 See <http://retinagallery.com/displayimage.php?pid=1810> (accessed on 9 June 2017).
 55. RetinaGallery. 2010 See <http://retinagallery.com/displayimage.php?pid=1822> (accessed on 9 June 2017).
 56. Heneghan C, Flynn J, O’Keefe M, Cahill M. 2002 Characterization of changes in blood vessel width and tortuosity in retinopathy of prematurity using image analysis. *Med. Image Anal.* **6**, 407–429. (doi:10.1016/S1361-8415(02)00058-0)
 57. Patton N, Aslam TM, MacGillivray T, Deary IJ, Dhillon B, Eikelboom RH, Yegesan K, Constable IJ. 2006 Retinal image analysis: concepts, applications and potential. *Prog. Retin. Eye Res.* **25**, 99–127.