

Animation of Biological Organ Growth Based on L-systems

Roman Ďurikovič[†], Kazufumi Kaneda

Hideo Yamashita

Electric Machinery Laboratory, Faculty of Engineering, Hiroshima University,
1-4-1 Kagamiyama, Higashihiroshima, 739-8527 Japan.
email: {roman,kin,yama}@eml.hiroshina-u.ac.jp

Abstract

In contrast with the growth of plants and trees, human organs can undergo significant changes in shape through a variety of global transformations during the growth period, such as bending or twisting. In our approach, the topology of a human organ is represented by a skeleton in the form of a tree or cycled graph. The length of skeleton growth can be simulated by an algebraic L-system that also produces discrete events. The paper shows how to include global transformations into the formalism of L-systems to obtain a continuous process. The shape of the organ is approximated by a number of ellipsoidal clusters centred at points on the skeleton. The proposed growth model of the organ continually responds to the positional changes of surrounding organs, thereby changing the organ shape locally. In our study, the stomach of a human embryo is used for the demonstration of organ development, and the methodology employed is also applicable to the animation of animal organs and their development.

1. Introduction

The development of biological organs, such as the stomach or the intestines, takes place through a complicated process. The shape and structure of such organs change significantly over a short period of time. Embryologists therefore strive to visualise changes in organs over the passage of time. The visualisation of organ development is also important for education and training in medical school, as students must obtain knowledge related to the development of the human body.

Growth of biological organs is nonlinear and yet obeys certain rules. However, the entire growth process and mechanism have not yet been made clear, with many unknown factors remaining. Fortunately, the principal processes governing the development of

human organs and geometric structures at several stages have been investigated¹. The best way to visualise such development while making the most of achievements in the field of embryology is not key-frame animation; rather it is rule-based animation, such as in botanical growth models.

Several attempts have been made using computer graphics to visualise biological growth. Methods for modelling botanical trees were developed to generate natural tree images^{2, 3}. These methods directly model the growth process using statistical data, such as the angle between trunk and branch. An L-system formalism was proposed by Lindenmayer⁴, and the method has been used as a general framework for plant modelling. An L-system with several extensions was extensively described by Prusinkiewicz and Lindenmayer⁵, and the extensions allow for such factors as context-sensitivity, random variations, and branch cutting. An expansion of the L-system to handle the interaction between plants and their environment has also been developed^{6, 7}. A recently proposed method con-

[†] On leave from Department of Computer Graphics and Image Processing, Faculty of Mathematics and Physics, Comenius University, 820 13 Bratislava, SLOVAKIA.

siders the plant and its environment as two separate models with information flowing between them in both directions⁸. Another interesting extension of L-systems is a behavioural L-system capable of animating the autonomous actors by external tactile and behavioural forces⁹.

The L-system is an excellent method for modelling botanical growth, as demonstrated in many publications^{10, 8}. However, the method cannot be directly applied to human organ development. Trees or plants monotonically increase their length and thickness of branches. In contrast, the growth process of human organs is more complicated. In the early stages of human organ development, a number of significant changes occur, making it very difficult to simulate organ growth. Generally, organs have no tree structure and their growth is not a monotone process. For example, in stomach and intestine development, different combinations of rotation, bending and twisting take place. These difficulties are unique to the modelling of human organ growth; they are not encountered in the modelling of tree or plant growth. Although, attempts to bend the plant leaves and apices have been made in plant formation¹¹, they do not allow the bending transformation of the entire plant structure.

In this paper, we propose a method for modelling human organ growth with the aim of visualising organ development over the passage of time. Using a skeleton structure, we represent the topology of an organ. In addition, we expand the L-system to be able to handle cycles, as many organs have a complex topology, which can only be described by graphs with cycles. Besides topology, the depiction of the growth process requires the representation of geometry and interaction with surrounding objects. The geometrical structure of organs is modelled using ellipsoids centred at nodes established on the skeleton. We introduce several tables with which to store the database of statistical geometry of organs, such as size, growth speed, among others. The proposed method also takes into account the interaction between the organ and its surroundings.

In the next section, a newly developed biological model of stomach growth is described. In Section 3, after explaining algebraic L-systems, the system continuity and an expansion of the L-system with cycles is proposed. Growth functions and tables in which to store the statistical geometry of human organs are discussed in Sections 4 and 5, respectively. Interaction with the environment surrounding the organ is described in Section 6. To demonstrate the usefulness of the proposed method, an animation of the stomach of a human embryo was generated.

2. Biological model of stomach growth

The stomach appears as a fusiform dilation of the foregut in week four of development¹. Its appearance and position change greatly as a result of the different growth rates of the various regions of its wall, in addition to the position changes of surrounding organs. The liver is the nearest fast-growing organ with large volume. Positional changes are easily explained by assuming that the stomach rotates around a *longitudinal* and an *anteroposterior* axis.

The stomach rotates 90 degrees clockwise around its longitudinal axis. During this rotation, the original posterior wall of the stomach grows faster than the anterior portion, resulting in the formation of the *greater* and *lesser* curvatures. During further growth, the stomach rotates around an anteroposterior axis such that the *pyloric part* moves to the right and upward and the *cardiac portion* moves to the left and slightly downward (see Fig. 1).

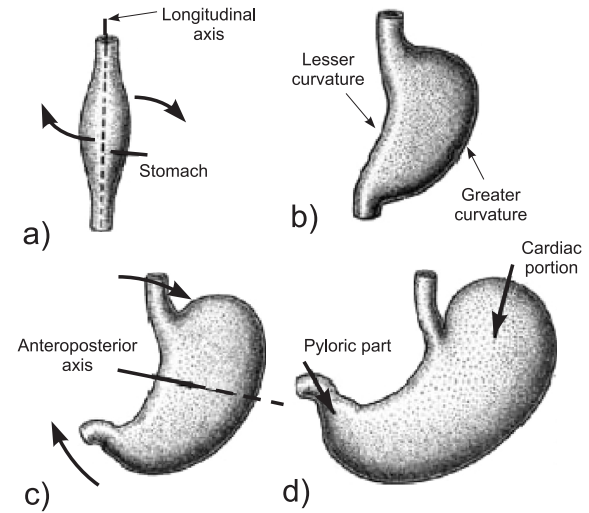


Figure 1: Schematic drawing to show stomach rotation. a), b) Rotation along the longitudinal axis at 4th and 10th week of development, respectively. c), d) Rotation of the stomach around the anteroposterior axis at 12th and 20th week of development, respectively.

2.1. Organ representation with ellipsoidal clusters of cells

Every organ is built up from a huge number of cells, each growing with different properties, such as speed and direction of growth. The objective is to reduce the number of cells without compromising to any great degree such organ properties as volume, surface area, topology, speed of grow in specific areas, and the like.

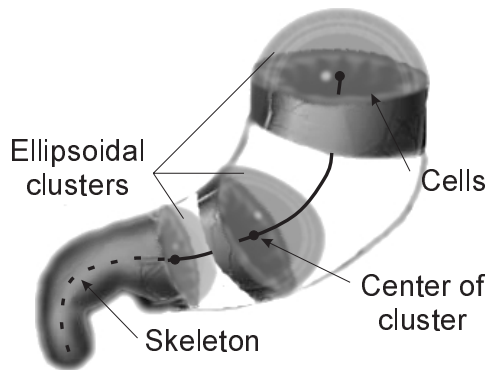


Figure 2: Process of stomach representation.

To approximate the shape of an organ while considering the speed and direction of cell growth at the same time, we group the entire set of cells into a number of ellipsoidal bunches (clusters), as indicated in Fig. 2. The cells included in a single cluster share similar growth properties within a certain deviation from the properties of the average cluster. Thus, the skeleton of the organ is defined by a chain of linear segments connecting the cluster centres. Organ growth can then be modelled by the growth of the skeleton, and variations in shape during the growth process can be captured by variations in cluster size. When an ellipsoidal cluster changed in size, it was understood that the organ cells grew in the directions emanating from the cluster centre. Similarly, when the skeleton segment underwent changes in length, it was understood that the cells included in two adjoining clusters grew in directions parallel to this segment. Nutrition flow and information exchange between cells along the organ is simplified by flow between clusters.

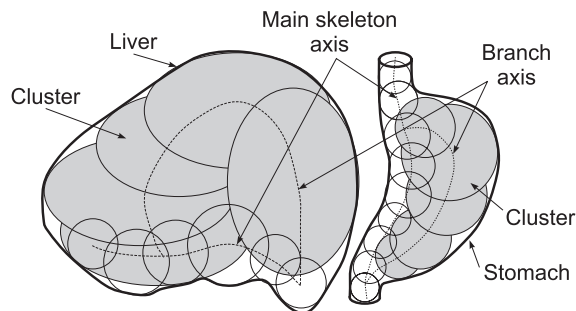


Figure 3: Object representation with cell clusters. Human embryo liver and stomach.

Published methods dealing with the spherical representation of 3-D objects¹² can be used to obtain

the initial skeleton and spherical clusters for the approximation of organ shape. The initial representation should be slightly modified to take into account similar growth properties within each cluster. For the sake of clarity, in Fig. 3, we demonstrate the ellipsoidal and spherical representation in two dimensions. For a 3-D representation, the term “circle” should be substituted with “sphere” and “2-D skeleton” with “3-D skeleton”. Figure 3 shows the skeleton and spherical clusters for a human embryo liver and stomach, respectively, that have grouped cells with similar growth speed.

2.2. Reconstruction of an ellipsoidal representation

The ellipsoidal representation is stable, that is, perturbation of several boundary points results in very similar ellipsoidal representations. On the other hand, the disadvantage of this representation is that the ellipse has very few degrees of freedom, only, limited types of shapes can be accurately represented by the union of a small number of ellipses. Nevertheless, the ellipsoidal representation can accurately capture the average growing properties of an organ. Further, such organ representation is useful for the biologically correct animation of growth processes successfully modelled by the theory of formal languages, as will be shown in this paper. Every skeleton segment and ellipsoidal cluster can be easily represented by a module using the terminology of L-systems⁴, while the growth can be controlled by module parameters.

As ellipses are used in the modelling process, the surface generated from the union of original ellipsoidal clusters is not smooth, a situation that unfortunately leads to an appearance that is not smooth. One effective way to smooth the surface without affecting the representation itself is to use scaled blended spheres¹³ (also known as *blobbies*) during the process of rendering the organ.

3. Continuous Algebraic L-systems

The proposed algebraic L-systems are extended to the dL-systems⁸ by introducing continuous global time control over the productions, stochastic rules for the capture of small variations, and explicit functions of time used to describe continuous aspects of model behaviour, in addition to differential equations.

In some cases it is convenient to describe continuous behaviour of the model using explicit functions of time rather than differential equations. For example, global shape transformations varying over time require a large and complicated system of differential equations, while only few explicit functions of time are

sufficient for description of the transformations. Continuous time flow is introduced, in place of discrete derivation steps. A module $A(\mathbf{w}, \tau)$ of a continuous algebraic L-system depends on parameters \mathbf{w} from the domain of values \mathcal{D}_A and local time $\tau \in [0, \infty)$. The interval $[\tau_a, \tau_b] \equiv \mathcal{T}_A$, $\tau_a \leq \tau_b < \infty$ determines the start and the end of module activity according to the local time τ . As long as the parameters \mathbf{w} belong to the interior of domain \mathcal{D}_A and $\tau \in \mathcal{T}_A$, the module grows (parameters \mathbf{w} are updated) in a continuous manner. At the moment \mathbf{w} reaches boundary \mathcal{D}_A while $\tau \in \mathcal{T}_A$, a production rule is applied as in L-systems⁴. The production rule replaces module $A(\mathbf{w}, \tau)$ by its descendants in a discrete event. The local time τ of the module is continually updated even if $\tau \notin \mathcal{T}_A$. The state of algebraic L-system at time t is represented as a sequence of modules

$$\mu = A_1, A_2, \dots, A_n.$$

The modules immediately preceding and following a given module are called the left context and the right context, respectively. Subscripts l, r are used to specify the left and right contexts.

The continuous behaviour of $A(\mathbf{w}, \tau)$ in the context of $A_l(\mathbf{w}_l, \tau_l) < A(\mathbf{w}, \tau) > A_r(\mathbf{w}_r, \tau_r)$, $\mathbf{w} = (\mathbf{w}_E, \mathbf{w}_D) = (w_1, \dots, w_s, w_{s+1}, \dots, w_n)$ is proposed to be described by a system of explicit functions

$$\mathbf{w}_E = F_A(\mathbf{w}_D, t) \tag{1}$$

that determine the vector of parameters $\mathbf{w} = (w_1, \dots, w_s)$, and ordinary differential equations determining the rate of change dw_i/dt of parameters w_{s+1}, \dots, w_n

$$\begin{aligned} \frac{d\mathbf{w}_D}{dt} &= f_A(\mathbf{w}_l, \mathbf{w}, \mathbf{w}_r, t) \\ \frac{d\tau}{dt} &= 1. \end{aligned}$$

We assume that \mathcal{D}_A is a simple connected and open set (a set with no self-intersecting boundary \mathcal{C}_A nor holes). To ensure the existence and uniqueness of the solution, the arbitrary function $f_A(\mathbf{w}_l, \mathbf{w}, \mathbf{w}_r, t)$ is assumed to have a bounded first derivation according to \mathbf{w} , and arbitrary function F_A is assumed to be continuous in all parameters.

The production rule comes into effect when one of the parameters \mathbf{w} reaches a point \mathcal{C}_{A_0} of the boundary \mathcal{C}_A while $\tau \in \mathcal{T}_A$. Let us note t_β as the time when the boundary is reached. Production at time t_β written

$$\begin{aligned} p_{A_k} : A_l(\mathbf{w}_l, \tau_l) < A(\mathbf{w}, \tau) > A_r(\mathbf{w}_r, \tau_r) \rightarrow \\ B_{k,1}(\mathbf{w}_{k,1}, \tau_{k,1}) B_{k,2}(\mathbf{w}_{k,2}, \tau_{k,2}) \dots \\ B_{k,m_k}(\mathbf{w}_{k,m_k}, \tau_{k,m_k}) : prob \end{aligned}$$

replaces module $A(\mathbf{w}, \tau)$ with descendant modules $B_{k,j}$. The index k is used to emphasise that different

productions can be associated with individual points or areas of the boundary \mathcal{C}_{A_k} . m_k is the number of newly created modules associated with production p_{A_k} . The last term in the production noted as *prob* is the probability of the production application. It is used when two or more productions are related to one module A with parameters satisfying the same boundary conditions p_{A_k} . In this case, the production to be applied to module A is determined randomly. An extension of the branching structures is straightforward using the definition of context as given by Prusinkiewicz and Lindenmayer⁵.

3.1. Continuity

Proposed here is the determination of the initial parameters for newly created modules. The hypothetical path of parameter vector \mathbf{w} in its domain \mathcal{D}_A is shown in Fig. 4a. The application of production on context can result in C^1 discontinuity of $\mathbf{w}(t)$. Moreover, if the explicit function of time F_A in Eq. 1 depends on left and right context parameters $\mathbf{w}_l, \mathbf{w}_r$, the production applied on context can result in the C^0 discontinuity of $\mathbf{w}(t)$. Since this is undesired behaviour, the parameters of function F_A are restricted to \mathbf{w}_D and t . The initial value of parameters associated with newly created module $B_{k,j}(\mathbf{w}_{k,j}, \tau_{k,j})$ is determined by a function $h_{A_{k,j}}$:

$$\begin{aligned} \mathbf{w}_{D_{k,j}} &= \lim_{t \rightarrow t_\beta^-} h_{A_{k,j}}(\mathbf{w}_l(t), \mathbf{w}(t), \mathbf{w}_r(t)) \\ \mathbf{w}_{E_{k,j}} &= F_A(\mathbf{w}_D, t_\beta). \end{aligned} \tag{2}$$

The new vector $\mathbf{w}_{k,j}$ must belong to the domain $\mathcal{D}_{B_{k,j}}$. As the general conditions Eqs. 2 are not sufficient for our application, stronger conditions will be used. First, we assume that all parameters $\mathbf{w}_E, \mathbf{w}_{E_{k,j}}$ have the same domain of values \mathcal{E} . Second, Eq. 1 is modified to guarantee continuity in time for parameter \mathbf{w}_E . Let us observe Fig. 4b showing the hypothetical trajectory of parameter \mathbf{w}_E along with two additional parameters $\mathbf{w}_{E_{k,1}}, \mathbf{w}_{E_{k,2}}$ assigned to modules $B_{k,1}(\mathbf{w}_{k,1}, \tau_{k,1})$ and $B_{k,2}(\mathbf{w}_{k,2}, \tau_{k,2})$ upon their creation at time t_β . There is an obvious discontinuity between parameters $\mathbf{w}_E, \mathbf{w}_{E_{k,1}}$ and $\mathbf{w}_{E_{k,2}}$ at time t_β , as the application of production determines that different explicit functions $F_{B_{k,1}}$ and $F_{B_{k,2}}$ are used to calculate parameters $\mathbf{w}_{E_{k,1}}, \mathbf{w}_{E_{k,2}}$. An S-shaped blending function is introduced to guarantee the time continuity of time-explicit parameters after application of the production rule (the discrete event). The blending function $b_{k,j}(\tau)$, monotonically increasing from 0 to 1 over the time interval $[\tau_{k,j}, T]$, is applied to Eq. 1 as follows

$$\begin{aligned} \mathbf{w}_{E_{k,j}} &= F_A(\mathbf{w}_D, t_\beta) + \\ &b_{k,j}(\tau)(F_{B_{k,j}}(\mathbf{w}_{D_{k,j}}, t_\beta) - F_A(\mathbf{w}_D, t_\beta)), \end{aligned} \tag{3}$$

where τ and $\tau_{k,j}$ denote the local time and initial local time of newly created module $B_{k,j}$, respectively. Explicate function F_A is a function from Eq. 1 related to the original module. Although the C^1 discontinuity can occur at boundary points of the blending function, the trajectory of parameters w_E and $w_{E_{k,j}}$ are continuous in time (see Fig. 4c). Higher order blending functions can solve the problem of C^1 discontinuity.

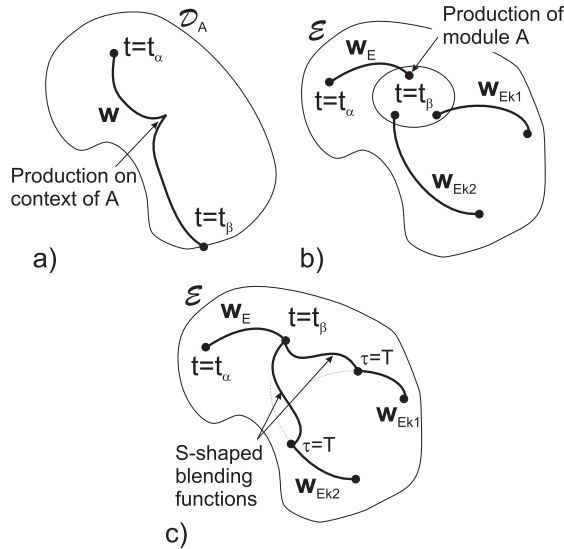


Figure 4: Hypothetical path of parameters in their domain. a) Parameter w_E in domain \mathcal{D}_A . b) Time discontinuity of parameters $w_E, w_{E_{k1}}, w_{E_{k2}} \in \mathcal{E}$ when the blending function is not used. c) C^0 continuity of $w_E, w_{E_{k1}}, w_{E_{k2}} \in \mathcal{E}$ using the blending function.

3.2. Algebraic L-systems with cycles

A class of planar graphs with cycles can be formally represented by *L-systems with markers*⁵. The markers proposed here are different from those introduced by Prusinkiewicz⁵, in which case they determined only the geometry of a system. In this paper, markers determine not only geometry but also neighbourhood relationships and information flow between modules. The *markers* specify the position of inserted points shared by the associated modules. Let us define a new star symbol ‘*’ to denote the marker. To define a single cycle, two markers should occur within the string. The markers are handled as pointers indicating the locations where other neighbourhood modules can be found during the derivation step. For example, let us consider the derived string $D[EF*]C * G$, where $[]$ defines a branch as usual⁵. The neighbourhood relationships are as follows: module D is a left context of modules E and C , module E is a left context of F , and

modules F and C are left contexts of G . See Fig. 5a for a possible graphical representation. The simplest way to define the multiple cycles is to use multiple markers defined by sequences of stars as ‘**’, ‘***’, ... For example, ‘***’ is a one symbol, that can be used to define the third cycle. The left and right contexts as understood from L-systems are therefore extended by multiple markers to an arbitrary neighbourhood in the cyclic L-system. Self loops such as the one in Fig. 5b can also be represented by this syntax using a symbol between two markers, as in the following $DC * E*$.

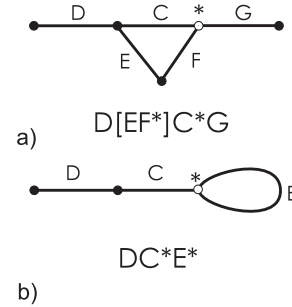


Figure 5: L-systems with markers used for representation of planar graphs with cycles.

3.3. Derivation process of algebraic L-systems

The derivation process is the calculation of the sequence of strings $\mu(0) = \mu_0, \mu(\Delta t) = \mu_1, \dots, \mu(n\Delta t) = \mu_n$ representing the stages of the growth model at the desired intervals Δt . Global time is advanced repeatedly by the step Δt . During each step, the explicit and differential equations are solved numerically. If the discrete event (production) should occur at time $t' \in [t, t + \Delta t)$, the interval is divided and the differential equations are integrated in the interval $[t, t')$. The parameters obtained are then used in the production application at time t' to determine the initial parameters for the explicit or differential equations associated with newly created modules. These equations are then integrated in interval $[t', t + \Delta t)$.

The derivation process applicable to the cycles within the string is an extension of the derivation process (string rewriting) discussed above. A *derivation step* in cyclic L-systems consists of two phases:

- Input string is scanned for *matching markers* to determine the neighbourhood relationship between modules.
- Each module in the string is replaced by successor modules using the corresponding production rule associated with the neighbourhood of the module.

4. Growth Functions

Continuous processes such as the elongation of skeleton segments, growth of cell clusters, and gradual increase in branching angles over time can easily be described by the growth functions. Growth functions are included into algebraic L-systems as explicit functions or differential equations. Growth is often slow initially, accelerating near the maximum stage, slowing again and eventually terminating. A popular example of the growth function⁸ is an S-shaped function $g_{\Delta x, T_0, T}(t)$, monotonically increasing from x_{min} to x_{max} with growth rates of zero at both ends of interval $[T_0, T]$. It is often known when the developmental processes begin and terminate in the case of normal development of organs. Although organs have no sense of time, their growth follows specific time intervals for different individuals quite accurately, making the use of growth functions with explicit dependence on time reasonable for the biological modelling of organ growth.

4.1. Global and local transformations

The special growth functions proposed herein represent such transformations as twisting, bending and tapering. The local transformations are easily combined to create complex shapes, while preserving (or modifying) volume, surface or arc length. The main transforming equation of any space curve $\mathbf{x} = (x_1(s), x_2(s), x_3(s))^T$, $s_0 \leq s \leq s_1$ preserving the arc length is given by

$$\mathbf{y}(s) = \int_{s_0}^s \mathbf{M}(s)\mathbf{x}'(s)ds, \quad (4)$$

where $\mathbf{M}(s)$ is the tangent transformation matrix, and \mathbf{x}' is a tangent vector of the space curve. Note that $|\mathbf{x}'| = 1$ for a curve is parameterized by arc length. For global twisting and bending, the transformation matrix \mathbf{M} is the Jacobian matrix of respective transformation¹⁴.

Any local bending transformation preserves the arc length when \mathbf{M} is a rotation matrix. For example, let us consider the local bending along the Z-axis with the radius of curvature $1/k$. The centre of the bend occurs at $s_c \in [s_0, s_1]$, and the range of the bend is controlled by $\Delta s_{min} \in [s_0, s_c]$ and $\Delta s_{max} \in [s_c, s_1]$. The tangent transformation matrix associated with bending is given by:

$$\mathbf{M} = \begin{pmatrix} C_\theta & -S_\theta & 0 \\ S_\theta & C_\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where the bending angle measured in radians is:

$$\theta = k(\hat{s} - s_c), \quad C_\theta = \cos(\theta), \quad S_\theta = \sin(\theta),$$

and where $\hat{s} - s_c$ is a clipping function

$$\hat{s} - s_c = \begin{cases} -\Delta s_{min} & s < s_c - \Delta s_{min} \\ s - s_c & s_c - \Delta s_{min} \leq s \leq s_c + \Delta s_{max} \\ \Delta s_{max} & s > s_c + \Delta s_{max}. \end{cases}$$

Multiple bending or twisting transformations with different parameters can be hierarchically combined together by using the multiplication of their tangent transformation matrices. Fortunately, the integral in Eq. 4 does not need to be calculated, as our interest lies in the rate of angle change between tangent vectors of the original curve and the transformed curve.

We define the *angle growth function* $G_y^M \in [-180, 180]$ transforming model locally

$$G_y^M(\mathbf{x}', \mathbf{y}', s, t) = \text{sign}((\mathbf{x}' \times \mathbf{y}')_y) \arccos(\mathbf{x}'(s) \cdot \mathbf{y}'(s, t)) \quad (5)$$

where $\mathbf{y}'(s, t) = \mathbf{M}(s)\mathbf{x}'(s)$ is a tangent vector of the transformed curve, and \cdot is a dot product. Function G_y^M corresponds to the rotation angle between vectors \mathbf{x} and \mathbf{y} around the Y principal axis at position s and in time t . The final rotation angle has a sign equal to the sign of the Y coordinate of a vector product $\mathbf{x}' \times \mathbf{y}'$. To generate a smooth transformation (deformation) of structure, the parameters s_c, s_{min}, s_{max} and k vary in time according to the S-shaped growth function. The rotation angles around the X and Z principal axes at position s and time t associated with transformation matrix \mathbf{M} are generated by the growth functions G_x^M and G_z^M , respectively. The angle growth functions G_x^M and G_z^M are derived by analogy using the above procedure.

5. Mathematical Modelling of Stomach Development

Many organs have a complex topology, which can only be described by graphs with cycles. Besides the topology, the growth process can require boundary conditions or information flow between the ends of branches. In particular, the stomach model (see Fig. 3) should satisfy the connectivity criterion between the main stomach axis and the branch in order to obtain the shape of the stomach. The branch was introduced into the topology with the aim of modelling the concave shape that results from different rates of growth in cell clusters along the main and branch stomach axes. Simple control of the angle between the main axis and the branch axis cannot guarantee shape connectivity at any point. To model the branch axis the L-system with cycles introduced above was used. An alternative approach is to compute one axis reflecting the centres of ellipsoidal clusters without a branch. Unfortunately, this approach can handle only the simple tubular shapes.

The modelling methodology proposed herein consists of two stages. First, the topology of the organ and its development is expressed using algebraic L-systems with cycles. At this stage the neighbourhood relationship between modules and the length of modules are established. This step captures the growth of cell clusters in size and along the organ skeleton. In the next stage, the geometry is modelled by filling up the space surrounded by cycles, and by applying the global transformations onto the skeleton.

The overall growth process naturally splits into a sequence of events described by “short” algebraic L-systems called *tables*. The output string of the table immediately preceding a given table is used as the input string to the given table. Each table has the prescribed number of iterations specified as an argument, see Tab. 1. An infinite number of iterations means that the table is repeatedly iterated until there are no changes in resulting strings. The simplified model of stomach growth repeatedly processes the following series of tables in each time step Δt :

- Skeleton growth table(1),
- Arc length calculation table(∞),
- Skeleton bending table(∞),
- Cluster growing table(1),
- Geometry representation table(1).

Table 1: Series of tables simulating stomach growth.

The resulting string from the table series in each time step can be represented graphically, and shading methods can be used for visualisation. In the following sections, the above tables are discussed in detail, while assuming a common notation for modules¹⁵:

- $+(\alpha)$, $-(\alpha)$ orientation change of the following module by $\pm\alpha$ degrees with respect to the preceding module,
- $[$, $]$ initialisation and termination of a branch, *i.e.* push and pop the module position and orientation from the stack, respectively.

5.1. Skeleton growth table

Each module in the algebraic L-systems depends on its local time (age). Therefore, different productions related to a module can be used at different time intervals, thus controlling the variety of processes that may take place in complex growth. The algebraic L-system model given in Fig. 6 controls the expansion of all components and gradually increases branching angles over time. A single marker ‘*’ was used to define the cycle between the branch and the main skeleton axis.

```

table Skeleton {
initial string: load from file axiom.tmp
output string: save to file axiom.tmp
 $F_i(x, \tau)$  :
  if  $x < x_{th}$  &  $\tau \in [\tau_s, \tau_e]$ 
    solve  $\frac{dx}{dt} = v, \frac{d\tau}{dt} = 1$ 
  if  $x = x_{th}$  &  $\tau \in [\tau_s, \tau_e]$  &  $\tau \neq \tau_b$  &  $i = A$ 
    produce  $D_A(kx, \tau)C(r(t), \tau)F_A((1-k)x, \tau)$  (a)
  if  $x = x_{th}$  &  $\tau = \tau_b$  &  $i = A$ 
    produce  $D_A(kx, \tau)C(r(t), \tau)[+(\alpha_0)F_b(x_0, \tau)*]$  (b)
     $F_A((1-k)x, \tau)$ 
  if  $x = x_{th}$  &  $\tau \in [\tau_s, \tau_e]$  &  $i = a$ 
    produce  $F_a((1-k)x, \tau)C(r_b(t), \tau)D_a(kx, \tau)$  (c)
  if  $x = x_{th}$  &  $\tau \in [\tau_s, \tau_e]$  &  $i = b$ 
    produce  $D_b(kx, \tau)C_b(r_0, \tau)F_b((1-k)x, \tau)$  (d)
  if  $\tau \notin [\tau_s, \tau_e]$ 
    solve  $\frac{dx}{dt} = 0, \frac{d\tau}{dt} = 1$ 
 $D_i(x, \tau)$  : solve  $\frac{dx}{dt} = g_{\Delta x, 0, T_1}(t)$  (e)
 $C_i(r, \tau)$  :
  if  $i = a, A$ 
    solve  $\frac{dr(s, t)}{dt} = g_{r_{max}, p}(r)$  (f)
  if  $i = b$ 
     $r(s, t) = G_b(s, t)$  (g)
 $\pm(\alpha)$  : solve  $\frac{d\alpha}{dt} = g_{\Delta\alpha, T_0, T_4}(t)$  (h)
}

```

Figure 6: L-system table controlling the growth of stomach skeleton. Function $g_{\Delta x, T_0, T}$ is the same as proposed by Prusinkiewicz et. al.⁸.

The first part of Fig. 6 supports the reading of initial axiom from file “axiom.tmp”. The output string derived from this table after the given number of iterations is saved to the same file “axiom.tmp”. Initially, the axiom stored in file “axiom.tmp” at time $t = 0$ has the form $|F_a(x_0, \tau_0)C(r_0, \tau_0)F_A(x_0, \tau_0)*$, where $|$ is equivalent to $-(180)$. Apical segments F_a , F_A , and F_b located at both ends of the main axis and at the endpoint of the branch cause elongation of the stomach (see Fig. 7). They are most active during development of the primitive stomach from a single cylinder. Two markers “*” create a single cycle in the stomach skeleton.

The largest part of Fig. 6 describes the behaviour of the apical segments. Every apex F_i has two parameters x and τ , which indicate its current length and local time. The combination of linear growth of apex segments with the cubic growth of internode segments D_i cause the first order continuity of the entire skeleton length. Upon reaching the threshold length x_{th} , the apex produces an ellipsoidal node C_i and subdivides into an internode D_i of length kx and a shorter apex with length $(1-k)x$, given by productions a, c, and d in Fig. 6. Apex F_A , which grows in a downward direction, produces an additional branching apex at time $\tau = \tau_b$ (see Fig. 6b). The S-shaped growth func-

tions describe the increasing length of internodes D_i and the magnitude of branching angle α (see Fig. 6e and h). Ellipsoidal node C_i represents a cell cluster with the shape of an ellipsoid. It has two parameters \mathbf{r} and τ , which indicate its current vector of radiuses and local time. The additional third vector of parameters θ can be added to manipulate the orientation of the ellipsoid with respect to the principal axes. The other possibility implemented in our approach¹⁶ is to use the module specifying orientation change with respect to the orientation of the previous module. The orientation change of the ellipsoidal modules was omitted from Fig. 6 for simplicity. The radiuses of the ellipsoidal node increase according to the growth function depending on global time and the position of the node calculated from arc length s (see Fig. 6f and g). Dependence of the growth function on position is quite natural in biology, as many organs grow at different rates depending on location.

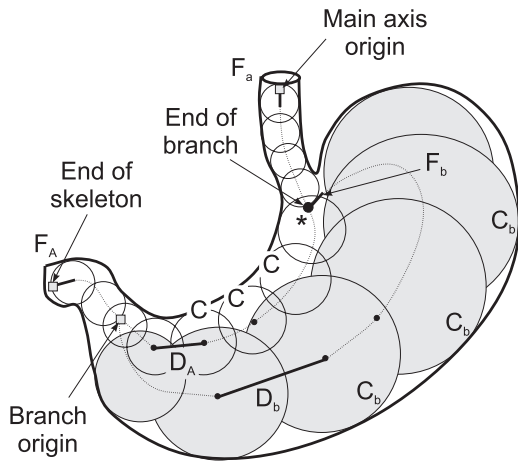


Figure 7: Geometrical representation of modules in the string.

5.2. Arc length calculation table

As mentioned above, few growth functions depend on the position of the actual module calculated from arc length parameter s . To have this information available at each time step, all modules in a system are made dependent on arc length parameter. This parameter is updated regularly, just after any module in a skeleton exhibits any growth. Therefore, all modules belonging to the main skeleton axis depend on the arc length parameter s equal to distance from the origin, and on the total length of main axis S . Similarly, modules belonging to the branch contain the arc length parameter measuring the length from the origin of the branch, and the parameter of the total branch length.

Each module on the main skeleton axis has its arc length parameter calculated from the sending of the signal starting at the skeleton origin (see Fig. 7) while collecting the length of internode segments along the string and storing the actual summation in parameter s . When the summation signal reaches the end of the skeleton, it sends back a signal containing the total length of the main axis and stores it as parameter S for each module passed. A similar process is implemented for the branch. The algebraic L-system for sending signals in upward and downward⁵ directions can be modified in a straightforward manner to handle the calculation of arc length. The arc length calculation table neither updates the global time t and local time τ of modules nor their growth. Therefore, the rewriting process ends after a finite number of derivation steps when the signals return.

5.3. Skeleton bending table

Total bending transformation can consist of multiple local bending transformations, whose parameters change over time. By focusing on transformation of the main axis, the transformation of the associated branch can be performed in the same way. Without loss of generality, suppose that at the very early stage of development the main axis of the stomach skeleton is a simple straight vertical line parametrized by arc length $\mathbf{x} = (0, 0, -s)$. The vertical line is then continually bent into required shapes using the transformation matrices that vary over time. The bending angle β is a new parameter introduced for this purpose into the internode module $D_i(x, \tau, \beta, s, S)$, although not written in Fig. 6. Since every module in the string depends on arc length s calculated from previous tables, the arc length at module D_i is also known. Assuming the given transformation matrix $M(s, t)$, nothing remains unknown for the evaluation of angle-growth function, Eq. 5, which can be further simplified to

$$G_y^M(\mathbf{y}', s, t) = -\text{sign}(\mathbf{y}'_y(s)) \arccos(-\mathbf{y}'_z(s)). \quad (6)$$

Bending angle β , in other words the angle between the vertical line \mathbf{x} and the actual internode segment D_i , is therefore given by an explicit function of time

$$\beta = G_y^M(\mathbf{y}', s, t).$$

The angle growth functions G_x^M and G_z^M can be simplified similarly. The continuity of the bending angle over time when the production rules are applied onto the internode segments is guaranteed by using the blending function, Eq. 3.

It is not a challenging task to implement the matrix productions of Eq. 6 using the L-system formalism as a separate table. The global and local times should remain unchanged for all modules in the derived string

after processing the skeleton bending table. Rewriting the iteration processes of this table will stop after application of a finite number of production rules.

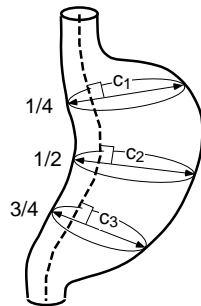
5.4. Cluster growth table

A separate table has been written to calculate the radiuses of all ellipsoidal modules C_b from Fig. 6g distributed along the stomach branch axis. The radius of an ellipsoidal cluster increases according to an explicit function of time and the position of ellipsoidal modules within a skeleton specified by arc-length s . Several statistical measurements have been made to specify the shape of explicit growth function $G_b(s, t)$. Table 2 shows the mean stomach thickness measured as a radius of a circle perpendicular to the main stomach axis. Measurements were taken at locations c_0, c_1, c_2 and c_3 relative to the length of stomach axis¹. Thereafter, the growth function $G_b(s, t)$ approximates the collected statistical measurements. Due to its complicated analytical form, it will not be stated here.

Table 2: Measurements of human embryo stomach.

day	length [mm]		diameter [mm]			
	embryo	stomach	c_0	c_1	c_2	c_3
21	3	0	0.08	0.08	0.08	0.08
28	4.3	0.84	0.08	0.12	0.21	0.10
35	9.6	1.37	0.14	0.27	0.34	0.20
70	50	8.89	0.89	3.37	3.55	2.84
84	71	12.6	1.26	5.41	5.71	4.20
140	160	33.3	3.33	14.7	13.3	9.99
252	500*	61.5	4.92	25.8	23.4	18.4

* Total height including the legs.



5.5. Geometry representation table

The aim of this table is to generate an actual 3-D object that can be visualised. Symbols in the derived string from the previous table can be related to structural elements in the growing form. Each structural element can have a geometric representation. The shape

of the organ in this paper is a smooth surface enveloping ellipsoids centred at nodes on the main axis and the branch skeleton axis, respectively. For simple visualisation purposes, each module of the resulting string acts as a command to a drawing device by *turtle graphics*¹⁵. The *turtle state* is characterised by position \vec{P} and its local coordinate system \vec{H}, \vec{U} and \vec{L} , indicating the turtle's heading direction, the upward direction, and the direction to the left. The local turtle coordinate system serves to represent the local orientation of each module. The brief introduction of the geometrical representation of modules from Fig. 6 is as follows

- *ellipsoidal module* C_i corresponds to a scaled blended sphere (blob¹³) fitting the sizes of the ellipsoid
- *internode segments* D_a, D_A with length equal to x located at the main skeleton axis corresponds to a meta-cylinder with length x defined by several blobbies in a line. The profile of the cylinder is scaled with respect to the radiuses of previous ellipsoidal modules.
- *internode segment* D_b located on the branch corresponds to a move with step x in the heading direction \vec{H}
- *bending angle* β corresponds to a rotation of internode an segment with respect to the upward direction \vec{U}
- *apex module* F_i is treated as an internode segment.

The implicit energy iso-surface defined by the collection of blobbies creates a smooth surface of the stomach without affecting the representation itself.

6. Taking into account positional changes of surrounding organs

In the previous section, a modelling methodology of organ growth determined by the different rates of growth in various regions of the organ was proposed. For the method to be in harmony with the field of embryology, the shape of the organ should accommodate the changes of surrounding organs through changes in position and size. Fast-growing organs of hard tissue overpower softer organs in their vicinity. For example, a fast growing liver affects the global bending of the stomach. This section extends the model of organ growth discussed above in the sense that the simulated organ continually responds to events occurring in the surrounding space, *the environment*. The model proposed herein is a closed-loop system involving the organ and its environment. After the organ reacts to the actual conditions taking place in its environment, it will reciprocally affect the environment. For example, when the simulated organ changes position, the tissue density and occupied space in the environment also change.

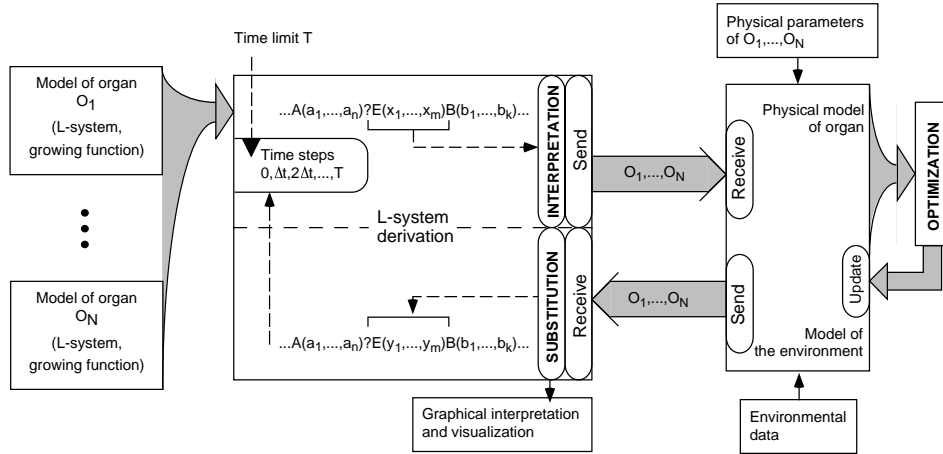


Figure 8: Block diagram of simulation growth of multiple organs sharing the same environment.

6.1. Information flow in the system

According to Fig. 8, the growth system controlling multiple organs can be understood as two independent blocks connected in a feedback loop. The two blocks represent the part for derivation of the algebraic L-system and the model of an environment including the physical model of an organ. This idea is similar to one proposed by Měch and Prusinkiewicz¹¹. However, the environment proposed in their research was limited to the modelling of tips of young plants and their changes, but changes in the entire plant in response to the environment could not be modelled. Here we propose the embedding of a physical model into the physical environment, thus allowing for local changes anywhere on the surface of a simulated organ.

Assume the model of an organ specified by the algebraic L-system, a string of symbols, equipped with growth functions derived from the statistical description of known changes in shape and position, proposed in Sec. 5. The proposed method also introduces the communication module¹¹, $?E(x_1, \dots, x_m)$ as a sensor perceiving environmental information. The communication module placed before each module in the resulting string from table series Tab. 1 carries complete information about the simulated organ. The sensors are placed before each module in the string in order to capture possible collisions with other organs or the environment occurring anywhere along the organ surface.

Definition of the organ is passed to the *derivation block* where the axiom of the model's L-system expands according to the rewriting rules. Derivation is performed sequentially in a number of small temporal steps Δt , in contrast to discrete integer steps used in plant development⁷.

In the *interpretation part*, the resulting string at each temporal step is scanned from left to right to determine the local coordinate system (state of the turtle) associated with each communication module. The state of the turtle and the information stored in the communication module is transferred serially in packages for each separate organ to the *environmental module*. Environmental data are defined by the user to allow easy modification of environmental parameters, such as space limitations of the environment or distribution of external forces.

Consequently, a *physical model* of the organ is created and placed into the bounded environment. Internal physical properties of the organ, such as the elasticity of the tissue, are used to simulate the competition for space between organs. Equilibrium among multiple organs and the environment is calculated by optimisation methods using finite differencing¹⁷. The internal forces of an organ, such as elasticity, stand in opposition to the external force from a pushing organ at places of collision. The result is an updated state of the surrounding environment and the optimal position of the organs, while at the same time preserving their volume.

Required parameter values are sent back to the derivation block to be substituted into communication modules. After all replies from the environment have been received, the resulting string may be interpreted and visualised. The next derivation step may be performed for a new time step, taking into account the newly stored parameters in the communication modules. Thus, another step of simulation is initiated.

6.2. Communication module

The communication module sends messages to the environment including the turtle position, orientation (vectors \vec{H} , \vec{L} , \vec{U}) and all the parameters of the immediately following module. For example, when the communication module is followed by an ellipsoidal module, all major radii are sent out. This allows the rebuilding of a simplified three-dimensional model in the environment. The communication module has the following format $E(x_1, \dots, x_m, \phi, m)$, where x_1, \dots, x_m are parameters of the next module, and the remaining parameters carry only received information. Messages returned to the organ model represent the direction of further growth given by a vector of rotation angles ϕ around the vectors \vec{H} , \vec{L} , \vec{U} and the magnitude of direction m .

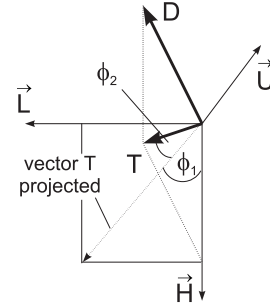


Figure 9: Definition of environmental output parameters ϕ and m .

6.3. Model of the environment

The environment comprises a three-dimensional discrete field that keeps track of organs, updating collision forces and transforming the organs according to the external forces applied. The optimisation method calculates the optimal position of organs, particularly their ellipsoidal modules, taking into account internal and external forces. The displacement vector \mathbf{D} defines a module's movement from its original position into an optimal one. The environment also determines further growth direction for each module. Assume a given module, the direction of future growth \mathbf{T} can be derived from a linear combination of the displacement vector \mathbf{D} and heading direction vector \vec{H} (see Fig. 9). Vector \mathbf{T} represented by the rotation angles $\phi = (\phi_1, \phi_2)$ with respect to \vec{L}, \vec{U} and by its magnitude $m = |\mathbf{T}|$ is sent back into the derivation block. Rotation angles ϕ_1 and ϕ_2 align vector \mathbf{T} with heading direction \vec{H} .

In the derivation block the turtle state of each module is updated by the rotation of heading direction \vec{H} about the angles ϕ_1 and ϕ_2 with respect to \vec{L}, \vec{U} . Therefore, the direction of internode segments is the same as the direction of further growth at that location.

7. Implementation and Results

One problem during implementation is selecting proper parameters of growth functions. The statistical measurements and a three-dimensional model of a real human embryo stomach over several stages of development were obtained from microscopic cross-sections and from Sadler's book¹ (see Tab. 2). First, the ellipsoidal representation of the organ for all available stages of development was established. Second, the development of an organ skeleton was modelled by elon-

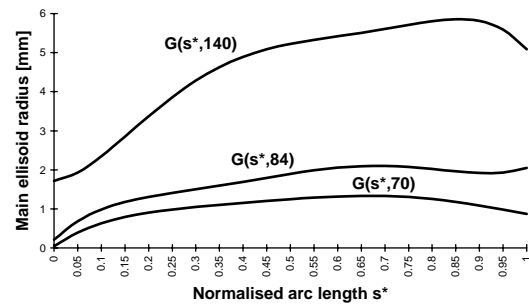


Figure 10: Function $G_b(s, t)$ shown for time 70, 84, and 140 days representing the size [mm] of the ellipsoid located at the skeleton branch. The arc length parameter was scaled to interval $[0, 1]$.

gating internode segments to accommodate the entire length of the stomach. Each of the inter-node segments grew 0.18mm per 154 days according to the S-shaped grow function $g_{0.18, 0, 154}(t)$, Fig. 6e. Such a small growth rate indicates a large increase of the length because of a large number of ellipsoidal clusters. Similarly, the function $g_{70, 28, 252}(t)$, monotonically increasing from 0 to 70 within the time interval measuring from 28 to 252 days, controlled the variation of the angle between the main and branch skeleton axis of the stomach, Fig. 6h. Assuming the unique length of the main skeleton axis, the global bending effect of a stomach was represented by four bending transformations varying over time applied on both the main and branch axes. For example, a 252-day-old stomach shown in Fig. 7 had the main axis transformed using the following parameters: the centre of the bends occurred at $s_c^1 = 0.85$, $s_c^2 = 0.67$, $s_c^3 = 0.26$, $s_c^4 = 0.23$; with the curvatures being $k^1 = 10$, $k^2 = -15$, $k^3 = -5$, $k^4 = 3.3$; and the range of the bends was $s_{\min}^1 = 0$, $s_{\max}^1 = 0.15$, $s_{\min}^2 = 0$,

$s_{\max}^2 = 0.07$, $s_{\min}^3 = 0.11$, $s_{\max}^3 = 0.25$, $s_{\min}^4 = 0.19$, $s_{\max}^4 = 0.06$. Lastly, the growth functions modelling the size of ellipsoidal clusters were determined. For example, the thickness of a stomach trunk – in other words the size of ellipsoids distributed along the main skeleton axis – used the *logistic* function¹⁸

$$gr_{\max,p}(r) = p \left(1 - \frac{r}{r_{\max}} \right) r,$$

where scaling factor $p = 0.031$ and maximal thickness $r_{\max} = 10.35$, (see Fig. 6f). The radius of an ellipsoidal cluster located at the stomach branch axis, however, increased according to explicit function of time $G_b(s, t)$, which approximates statistical measurements. For example, Fig. 10 demonstrates the growth function for the major radius of the ellipsoid cluster at time 70 days, 84 days, and 140 days, where the function values representing the major ellipsoid radius are measured in millimetres. Arc length parameter s in those graphs was normalised by the total length of the branch at the corresponding time.

Shown in Fig. 11 are several frames from a generated animation simulating stomach growth based on the proposed algebraic L-system using the above growth functions. The shape of the stomach model shown in this figure undergoes global bending transformation and deformations resulting from competition for space between the fast-growing liver and the stomach itself. Although meta-balls were used to obtain the smooth appearance, creating the surface from a swept-sphere rather than a collection of implicit surfaces would probably yield a smoother stomach¹⁹.

8. Summary

In this paper, we introduced a method for the modelling and simulation of human or animal organs obeying actual biological movements known to embryologists. We proposed a system in which multiple organs and the environment are separate processes, information from each of which is transmitted using communication modules. The physical environment and physical model allow for local changes anywhere on the surface of the simulated organ. The proposed algebraic L-systems with cycles introduced global transformations and explicit functions of time into the formalism of L-systems. By using blending functions, parameters having a common definite domain have guaranteed continuity in time, even after the application of discrete production rules. To store the statistical geometry of human organs, growth functions and L-system tables have been proposed.

The proposed methodology was demonstrated on a growth model of the human embryo stomach. The developed model produces convincing simulated images showing considerable likeness to an actual human

stomach at its different stages of development. Generated computer animation and visual interpretation make it possible to understand the modelled processes accurately and lead to realistic exploration of the human stomach at various stages of its development.

The proposed framework is also designed for use in the simulation of growth of other organs. Our future research is to concern development of a model of a human gut with all its complicated movements.

9. Acknowledgment

Authors wish to thank the embryologists Prof. Mineo Yasuda and Dr. Akinao G. Sato for their advice and for providing the sample data. The images were rendered by the Rayshade ray tracing program programmed by Craig Kolb. This research was sponsored by grants from the Japanese Society for the Promotion of Science, and the Japanese Ministry of Education, Sport and Culture.

References

1. T. W. Sadler, *Langman's medical embryology, 7th edition*. New York: Williams & Wilkins, (1995).
2. M. Aono and T. L. Kunii, "Botanical tree image generation", *IEEE Computer Graphics and Applications*, **4**(5), pp. 10–34 (1984).
3. J. Bloomenthal, "Modeling the mighty maple", *ACM Computer Graphics*, **19**(3), pp. 305–311 (1985).
4. A. Lindenmayer, "Mathematical models for cellular interaction in development, parts i and ii", *Journal of Theoretical Biology*, **18**, pp. 280–315 (1968).
5. P. Prusinkiewicz and A. Lindenmayer, *The algorithmic beauty of plants*. New York: Springer-Verlag, (1990).
6. C. M. Liddell and D. Hansen, "Visualizing complex biological interactions in the soil ecosystem", *The Journal of Visualization and Computer Animation*, **4**, pp. 3–12 (1993).
7. P. Prusinkiewicz, M. James, and R. Měch, "Synthetic topiary", in *Computer Graphics (SIGGRAPH '94 Proceedings)* (A. S. Glassner, ed.), pp. 351–358, ACM SIGGRAPH, (July 1994).
8. P. Prusinkiewicz, M. S. Hammel, and E. Mjølness, "Animation of plant development", in *Computer Graphics (SIGGRAPH '93 Proceedings)* (J. T. Kajiya, ed.), vol. 27, pp. 351–360, ACM SIGGRAPH, (Aug. 1993).

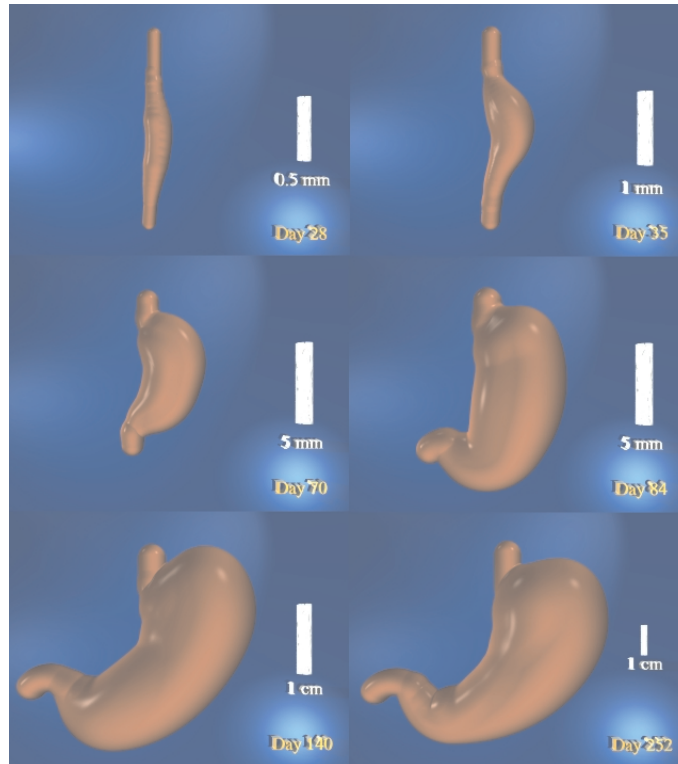


Figure 11: Development of a stomach simulated with proposed algebraic L-system taking into account positional changes of surrounding organs. Meta-balls were used to obtain a smooth appearance of the shape. The indicated stages represent 28, 35, 70, 84, 140, 252 days of an animated sequence.

9. H. Noser and D. Thalmann, "The animation of autonomous actors based on production rules", in *Proceedings Computer Animation'96, Geneva, Switzerland*, (Los Alamitos, California), pp. 47–57, IEEE Computer Society Press, (June 1996).
10. F. D. Fracchia, P. Prusinkiewicz, and M. J. M. de Boer, "Visualization of the development of multicellular structures", in *Proceedings of Graphics Interface '90*, pp. 267–277, (May 1990).
11. R. Mēch and P. Prusinkiewicz, "Visual models of plants interacting with their environment", in *Computer Graphics (SIGGRAPH '96 Proceedings)*, New Orleans, LA, pp. 397–410, ACM SIGGRAPH, (Aug. 1996).
12. V. Ranjan and A. Fournier, "Volume models for volumetric data", *IEEE Computer, special issue on volume visualization*, **27**(7), pp. 28–36 (1994).
13. J. F. Blinn, "A generalization of algebraic surface drawing", *ACM Transactions on Graphics*, **1**(3), pp. 235–256 (1982).
14. A. H. Barr, "Global and local deformations of solid primitives", in *Computer Graphics (SIGGRAPH '84 Proceedings)* (H. Christiansen, ed.), vol. 18, (held in Minneapolis, Minnesota; 23–27), pp. 21–30, (July 1984).
15. H. Abelson and A. A. diSessa, *Turtle geometry*. Cambridge, MA: MIT Press, (1982).
16. R. Āurikoviĉ, K. Kaneda, and H. Yamashita, "Visual modeling of stomach growth on the basis of L-systems", in *Proceedings of the Shape Modeling International*, (Aizu-Wakamatsu, Fukushima, Japan), pp. 121–128, (Mar. 1997).
17. A. M. Witkin, "An introduction to physically based modeling", in *Computer Graphics (SIGGRAPH'94 Tutorials)*, (Orlando, Florida), ACM SIGGRAPH, (1994).
18. L. Edelstein-Keshet, *Mathematical models in biology*. New York: Random House, (1988).
19. J. J. V. Wijk, "Ray tracing objects defined by sweeping a sphere", in *Proc. EUROGRAPHICS '84*, (held in Copenhagen, Denmark), pp. 73–82, (1984).