Computer Animated Stomach Growth

Roman Ďurikovič, Kazufumi Kaneda, Hideo Yamashita
Electric Machinery Laboratory, Faculty of Engineering
Hiroshima University, Japan

Abstract

In the approach described herein, both empirical and formal techniques are used to construct a framework for simulating organ growth. The algebraic L-systems able to handle the discrete and continuous events are formally introduced. The topology of the organ is represented by a tree-like skeleton mathematically expressed by the algebraic L-systems having constructs dependent on time and smooth growth functions. The skeleton also undergoes the global geometric transformations such as rotation, bending and twisting. The shape of the organ is a smooth surface enveloping spheres (or meta-balls) centered at points of the skeleton with the radii calculated according to a growth function.

The method in particular captures the development of human embryo stomach starting from third week of development consistent with static observations of the real embryo. Computer animation and visual interpretation make it possible to understand the modeled processes accurately and lead to realistic exploration of human stomach at various stages of development.

Key Words: Biological structures, three-dimensional modeling, L-system, organ development

1 Introduction

Computer simulation of biological growth of natural objects such as human organs is a challenging task for two reasons. One reason is the complexity of known changes in shape during development often determined by interactions with neighbor organs or by the genetic information coded in the organism. For example, the genetic information often determines the physical properties of each cell its direction and speed of growth. The other reason is the large amount of geometrical information that must be managed. Different approaches have been introduced to model natural growth. Among the empirical methods, the most popular is direct modeling of the growing form using observed data measured from a real object [2, 5]. Formal techniques, L-systems for example, involve direct modeling of the biological processes underlying organ growth. The L-system formalism, as a general framework for plant modeling, was proposed by Lindenmayer [7]. He co-authored the book [11] that extensively describes the L-system with a few extensions that allow for context-sensitivity, random variations and branch cutting.

Several attempts to extend the L-system to handle the interaction between plants and their environment including neighboring plants have been made [6, 10]. A recently-proposed method considers the plant and its environment as two separate models with information flow between them in both directions [8].

There has been extensive research on the development of mathematical models to describe the growth of plants, trees, fungi or molds [11, 13], but no computer graphics simulations of human or animal organs have been published.

Trees or plants monotonically increase the length and thickness of branches although entire branches can be cut off the tree. By contrast, the growth process of organs is not understood in detail. A number of processes are known to contribute to organ growth, but the details of these processes remain to be discovered. At early stages of development, many significant changes occur, making it very difficult to simulate organ growth. Generally, organs have no tree structure and their growth is not a monotonic process. For example, in stomach and gut development, different combinations of rotation, bending and twisting can be observed. These difficulties are unique to the modeling of organ growth; they are not encountered in the modeling of tree or plant growth.

In the approach described herein, both empirical and formal techniques are used to construct a framework for simulating organ growth. The topology of the organ is represented by a tree-like skeleton that grows according to formal L-systems as in plant development (the formal technique). Consequently, the shape of the organ is a smooth surface enveloping spheres centered at nodes or branch-points of the tree or skeleton with a radius calculated according to a growth function (the

empirical technique).

Next section starts with a brief introduction to the open L-systems and then the continuous algebraic L-system depending on the local time is proposed. Continuous behavior of the system is described by the explicit functions and by the ordinary differential equations. In Section 3 we propose continuous growing functions defined as a combination of local and global transformations such as bending and twisting while preserving the arc length. Biological process of stomach growth as observed by biologists is described in Section 4. Section 5 defines the mathematical model of stomach growth based on three L-system tables. First table determines the growth and the topology of the stomach skeleton. Second table calculates the arc length for each point along the skeleton. The last table simulates the complicated bending process of the stomach. Finally, the animation snapshots are shown.

2 Open L-systems

An L-system [1] is a parallel rewriting system based on an *alphabet*, an *axiom* and a set of rewriting rules or *productions*. In each step, rewriting rules or productions replace all letters (modules) in the predecessor string by successor modules. A production has the format:

$$id : lc < pred > rc : cond \rightarrow succ : prob,$$

where id is the production label, $lc, pred$, and rc are the left context, the predecessor, and the right context respectively, $cond$ is the condition, $succ$ is the successor, and $prob$ is the probability of production application (in stochastic L-systems). If the system contains the left and right contexts it is called a *context sensitive* L-system. The rewriting process is initiated by an axiom, for example

$$\omega : A(1)B(3).$$

The production that $A(x)$ should be changed to the module $B(x+1)$ with supporting branch $A(x)$ and condition $x < 4$ is represented as

$$p1 : A(x) : x < 4 \rightarrow B(x+1)[A(x)].$$

Consequently, the first derivation step may have the form:

$$A(1)B(3) \Rightarrow B(2)[A(1)]B(3).$$

Modules in the derived string can be related to structural elements in the growing form. For visualization

[1] The recommended review of open L-systems with examples of applications is [11].

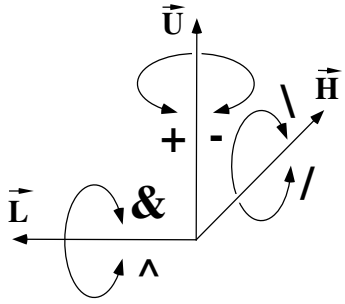


Figure 1: Turtle orientation and rotation in 3D.

purposes, each module of the resulting string acts as a command to a drawing device by *turtle graphics* [1]. The *turtle state* is characterized by a position $\vec{P}$ and its local coordinate system $\vec{H}, \vec{U}$ and $\vec{L}$, indicating the turtle's heading, the up direction, and the direction to the left (see Fig. 1).

Typically used command modules are F to draw a line in the current direction, f to move forward without drawing a line, [to initiate a branch,] to end branch process and modules $+, -, \&, \wedge, /$ and $\backslash$ rotate the turtle around one of the vectors $\vec{H}, \vec{U}$ and $\vec{L}$, as shown in Fig. 1. The length of the line and the rotation angle are specified as module parameters.

2.1 Continuous Algebraic L-systems

Algebraic L-systems are extended to the dL-systems [9] by introducing continuous global time control over the productions, stochastic rules to capture small variations, and the explicit functions of time to describe continuous aspects of model behavior in addition to differential equations. The continuous time flow is introduced instead 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 modules grows (parameters $\mathbf{w}$ are updated) in a continuous way. At the moment when $\mathbf{w}$ reaches the boundary of $\mathcal{D}_A$ while $\tau \in \mathcal{T}_A$ the production rule is applied as in L-systems (discrete event). The local time τ of the module is continually updated if $\tau \notin \mathcal{T}_A$.

The continuous behavior of $A(\mathbf{w}, \tau)$ in the context of $A_l(\mathbf{w}_l, \tau_l) < A(\mathbf{w}, \tau) > A_r(\mathbf{w}_r, \tau_r)$, $\mathbf{w} = (w_1, \ldots, w_s, w_{s+1}, \ldots, w_n)$ is described by a system of

explicit functions

$$w_i = F_A(\mathbf{w}_l, \mathbf{w}_r),\ i = 1, \ldots, s$$

that determine parameters $w_1, \ldots, w_s$ and ordinary differential equations determining the rate of change dw_i/dt of parameters $w_{s+1}, \ldots, w_n$

$$\begin{aligned} \frac{dw_i}{dt} &= f_A(\mathbf{w}_l, \mathbf{w}, \mathbf{w}_r, t),\ i = s+1, \ldots, n \\ \frac{d\tau}{dt} &= 1. \end{aligned}$$

We assume that $\mathcal{D}_A$ is a simple connected and open set (a set with no self intersecting boundary $\mathcal{C}_A$ and with no holes). In order to ensure the existence and uniqueness of the solution, the function $f_A(\mathbf{w_l}, \mathbf{w}, \mathbf{w_r}, t)$ is assumed to have a bounded first derivation according to $\mathbf{w}$.

The production rule comes into effect when one of 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, the time when boundary is reached. Production at time t_β is then written:

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

The index k is used to emphasize that different productions can be associated with individual points or areas of the boundary $\mathcal{C}_A$. m_k is the number of newly created modules associated with production p_{A_k}. 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 $h_{A_{k,j}}$:

$$\mathbf{w}_{k,j} = \lim_{t \rightarrow t_\beta^-} h_{A_{k,j}}(\mathbf{w}_l(t), \mathbf{w}(t), \mathbf{w}_r(t)).$$

The new vector $\mathbf{w}_{k,j}$ must belong to the domain $\mathcal{D}_{B_{k,j}}$. 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 on module A is determined randomly.

The derivation process is the calculation of the sequence of strings $\mu(0) = \mu_0, \mu(\Delta t) = \mu_1, \ldots, \mu(n\Delta t) = \mu_n$ representing the stages of the growing 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 $[t', t + \Delta t)$.

3 Growth Functions

Continuous processes such as elongation of individual cells, gradual increase of branching angles, and smooth local or global transformation (bending or twisting) of individual structures over time can easily be described by the growth functions. Growth functions are included into algebraic L-systems as explicit functions or the differential equations.

Initial growth is often slow, then it accelerate and near the maximum it again slows down and eventually terminates. A popular example of the growth function is an S-shaped function monotonically increasing from x_{min} to x_{max} with zero growth rates at both ends of an interval T. For example, the cubic growth function (see Fig. 2) has the form:

$$x(t) = -2\frac{\Delta x}{T^3}t^3 + 3\frac{\Delta x}{T^2}t^2 + x_{min} \equiv g_{\Delta x,T}(t),$$

where $\Delta x = x_{max} - x_{min}$ and $t \in [0, T]$. It can be expressed by the equivalent differential equation

$$\frac{dx}{dt} = G_{\Delta x,T}(t) = \begin{cases} 6\frac{\Delta x}{T^2}(1 - \frac{t}{T})t & \text{for } t \in [0, T] \\ 0 & \text{for } t \in (T, +\infty) \end{cases}$$

with the initial condition $x_0 = x_{min}$.

It is often known when the developmental processes start and end in the case of normal development of an organ. Although the organ has no sense of time, its growth follows the time intervals for different humans quite accurately. This makes the use of growth function with the explicit dependence on time reasonable for biological modeling of the organ growth.

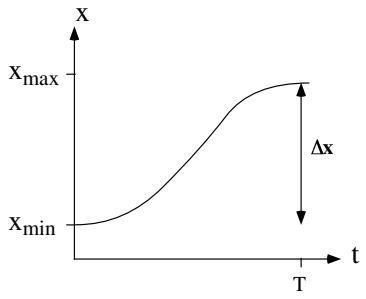


Figure 2: A cubic growth function $G_{\Delta x,T}$.

3.1 Global and Local Transformations

The special type of growth functions are transformations such as twisting, bending and tapering. The local transformations are easily combined creating complex shapes while preserving (or modifying) volume, surface or arc-length. The main transforming equation [3] 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

$$y(s) = \int_{s_0}^{s} \mathbf{M}(s)\mathbf{x}'(s)ds, \quad (1)$$

where $\mathbf{M}(s)$ is the tangent transformation matrix and $\mathbf{x}'$ is a tangent vector of space curve. For the global twisting and bending the transformation matrix $\mathbf{M}$ is the Jacobian matrix of respective transformation [4].

Any local bending transformation preserves the arc-length when $\mathbf{M}$ is a rotation matrix. Let us consider the local bending along the Z-axis with the radius of curvature $1/k$. The center 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:

$$\begin{aligned} \theta &= k(\hat{s} - s_c), \\ C_\theta &= \cos(\theta), \\ S_\theta &= \sin(\theta), \end{aligned}$$

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

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

Multiple bending or twisting transformations with different parameters can be hierarchically combined together by multiplication of their tangent transformation matrices. Fortunately, the integral in Eq. 1 do not need to be calculated because our interest is on the rate of angle change between successive line segments in L-system. Tangent vectors of the original and transformed curve can be used for this purpose. The angle growth function transforming model locally depends on the space parameter $s_0 \leq s \leq s_1$ of original curve, the time $t \in [T_a, T_b]$, tangent vector $\mathbf{x}'(s)$ to the original curve, and the transformation matrix $\mathbf{M}$ specified by the time varying parameters:

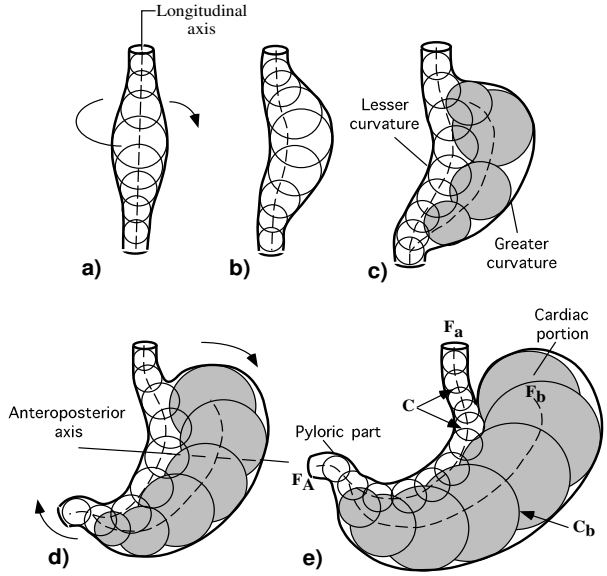


Figure 3: Modeling the surface of an organ. a), b) Stomach at early stages of development. Rotation along the longitudinal axis accures from 4th to 10th week of development. c), d), e) The stomach shape became concave while rotating around the anteroposterior axis at later stages of development.

$$G_M(\mathbf{x}', \mathbf{y}', s, t) = sign((\mathbf{x}'(s) \times \mathbf{y}'(s))_y) \arccos\left(\frac{\mathbf{x}'(s) \cdot \mathbf{y}'(s,t)}{|\mathbf{x}'(s)||\mathbf{y}'(s,t)|}\right)$$

where $\mathbf{y}'(s,t) = \mathbf{M}_{s_c, s_{min}, s_{max}, k, t}(s)\mathbf{x}'(s)$ is a tangent vector of transformed curve and $\cdot$ is a dot product. Final angle has the sign equal to the sign of Y coordinate of a vector product $\mathbf{x}' \times \mathbf{y}'$. Transformation parameters s_c, s_{min}, s_{max} and k vary in time according to the cubic growth function $G_{\Delta x, T}$ to generate the smooth transformation (deformation) of structure.

4 A Biological Study of Stomach Growth

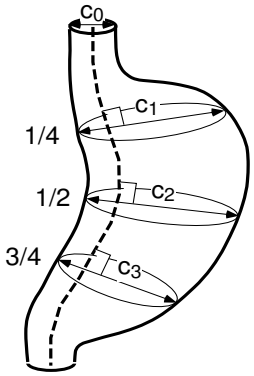
Positional changes of the stomach are easily explained by assuming that the stomach rotates around a *longitudinal* and an *anteroposterior* axis (see Fig. 3a).

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

Table 1: Measurements of the human embryo stomach.

day	length [mm]		diameter [mm]			
	embryo	stomach	c_0	c_1	c_2	c_3
21	3	0	0.084	0.084	0.084	0.08
28	4.3	0.84	0.084	0.115	0.210	0.10
35	9.6	1.37	0.137	0.274	0.342	0.20
70	50	8.89	0.889	3.374	3.552	2.84
84	71	12.62	1.262	5.409	5.709	4.20
140	160	33.33	3.333	14.67	13.33	9.99
252	500*	61.5	4.92	25.75	23.39	18.4

* Total hight including the legs.



such that the *pyloric part* moves to the right and upward and the *cardiac portion* moves to the left and slightly downward (see Fig. 3d). Several statistical measurements have been made for use in growth simulation. Table 1 shows the mean stomach thickness measured as a radius of a circle perpendicular to the main stomach axis. Measuraments were taken at places c_0, c_1, c_2 and c_3 [12] relative to the length of stomach axis.

5 Modeling of Stomach Growth

New difficulties arising from the simulation of organ growth are addressed and partially solved by the proposed methods described in this section. The topology of an organ is represented by a tree-like skeleton with spheres centered at nodes or branches of the skeleton and connected by cones. Organs having a concave shape but no holes can be approximated by multiple branching of the skeleton without cycles. Using meta-balls instead of spheres along the skeleton gives a smooth implicit surface. An open L-system is used to simulate the growth of the skeleton and associated spheres (their radius).

For example, in early stages of growth, the stomach has a simple tubular shape represented accurately by a longitudinal axis without branching (see Figs. 3a and b). The concave shape, that appears smoothly at later stages of development, can be simulated by allowing a branch to grow from the main axis (see Figs. 3c, d, and e).

5.1 Skeleton Growth Table

Every symbol in the algebraic L-systems depends on its local time (age). Therefore, different productions related to a symbol 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 bellow controls the expansion of all components and gradually increases branching angles over time.

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 $D_A(kx,\tau)C(r_0,\tau)F_A((1-k)x,\tau)$ (a)
  if $x = x_{th}$ & $\tau = \tau_b$ & $i = A$
    produce $D_A(kx,\tau)C(r_0,\tau)[+(\alpha_0)F_b(x_0,\tau)]$ (b)
      $F_A((1-k)x,\tau)$
  if $x = x_{th}$ & $\tau \in [\tau_s, \tau_e]$ & $i = a$
    produce $F_a((1-k)x,\tau)C(r_0,\tau)D_a(kx,\tau)$ (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** $\frac{dx}{dt} = G_{\Delta x, T_1}(t)$ (e)
$C_i(r,\tau)$:
  if $i = a, A$
    solve $\frac{dr(s,t)}{dt} = G_{\Delta r(s), T_2(s)}(t)$ (f)
  if $i = b$
    solve $\frac{dr(s,t)}{dt} = G_{\Delta r(s), T_3(s)}(t)$ (g)
$\pm(\alpha)$: **solve** $\frac{d\alpha}{dt} = G_{\Delta\alpha, T_4}(t)$ (h)
}

Table 2: L-system table controlling the growth of stomach skeleton.

First part of Table 2 supports the reading of initial axiom from file "axiom.tmp". 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 $-(180)$ i.e. reverse the turtle. Apical segments F_a, F_A, and F_b located at both ends of main axis and at the endpoint of branch cause elongation of the stomach (see Fig. 3e). They are most active during development of the primitive stomach from a single cylinder.

The largest part of Tab. 2 describes the behavior of 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 a spherical node C_i and subdivides into internode D_i of length kx and a shorter apex with length $(1-k)x$ given by productions a, c, and d in Tab. 2. Apex F_A that is growing downward produces the additional branching apex at time $\tau = \tau_b$ (see Tab. 2b). See Fig. 4 for geometrical explanation.

Cubic growth functions describe the increasing length of internodes D_i and the magnitude of branching angle α (see Tab. 2e and h). Sperical node C_i is graphically represented as a sphere. It has two parameters r and τ which indicate its current radius and local time. Radius of a spherical node increases according to growth function depending on time and the position of the node calculated from arc-length s (see Tab. 2f and g).

Dependence of the growing function on position is quite natural in biology because many organs grow with different rates depending on location. This kind of growing function can be used to capture the complicated growth processes.

5.2 Arc-length Calculation Table

All modules belonging to the main skeleton axis depends on the arc-length parameter s having value equal to distance between apex F_a and the actual position of the module, and on the total length S of main axis. Similarly, modules belonging to the branch contain the arc-length parameter measuring the length from the origin of the branch, and parameter of the total length of branch. For simplicity, the arc length parameter s and the total length parameter S have been omitted from the above skeleton table shown in Tab. 2.

Each module on the main skeleton axis has its arc-length parameter calculated from sending the signal starting at apex F_a and collecting the length of internode segments. When signal reaches the bottom apex F_A it sends back the signal containing the total length of main axis. Similar process is implemented for the branch. The algebraic L-system sending signal up and down described in [11] can be easily modified to handle the arc-length. The arc-length calculation table does not update the local time τ of symbols and symbols do not grow, therefore rewriting process stops after the finite number of derivation steps when the signal comes back.

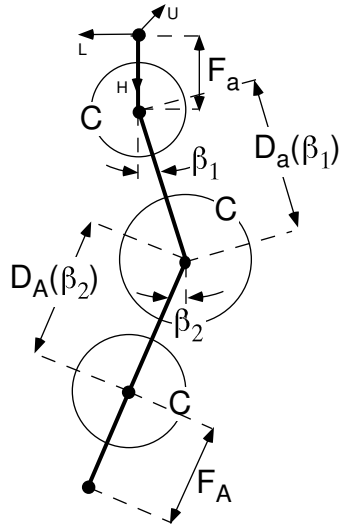


Figure 4: Geometrical representation of modules in the string. The bending angle β for each internode D is shown. Initial curve for the bending transformation is a straight line $\mathbf{x}(s)$, the result of transformation is curve $\mathbf{y}(s)$.

5.3 Skeleton Bending Table

Bending of the main skeleton axis and its branch can be measured for few static stages of development and then a smooth interpolation can be used to get the global continuous transformation of skeleton over the time. Total bending transformation can consist of multiple local bending transformations changing their parameters over the time. Let us focus on bending of main axis, the bending of associated branch can be performed at the same way. To further simplify the model we consider the bending along the Z-axis, i.e., the skeleton is a planar curve. Straight vertical line $\mathbf{x} = (0, 0, -s)$ is continually bendended into required shapes. The result of this transformation is a bending angle, in other words, the angle between the straight vertical line and the bended segment (see Fig. 4). Table 3 is an algebraic L-system that determines the bending angle of each internode segment D_i taking into account section 3.1.

Axiom of Tab. 3 is actually an output string of skeleton growth table processed by arc-length calculation table loaded from file "axiom.tmp". After given number of iteration steps the output string is stored again into the file "axiom.tmp". The local time τ for all symbols in the string remains unchanged after processing the table. Rewriting iteration process will stop after $N + 4$ steps, where N is number of local transformations.

table Bending {
initial string: load from file axiom.tmp
output string: save to file axiom.tmp
 # define $\mathbf{x}' = (0,0,-1)$
$D_i(x,\tau,\beta,s)$:
produce
$Y(\mathbf{x}',0)$
$B_1(s_c^1, s_{min}^1, s_{max}^1, k^1, \tau, s, \mathbf{M}^1, 0)$
$B_2(s_c^2, s_{min}^2, s_{max}^2, k^2, \tau, s, \mathbf{M}^2, 0)$
$\vdots$
$B_N(s_c^N, s_{min}^N, s_{max}^N, k^N, \tau, s, \mathbf{M}^N, 0)$
$D_i^{end}(x,\tau,\beta,s)$ (a)
$B_i(s_c, s_{min}, s_{max}, k, \tau, s, \mathbf{M}, type)$:
if $\tau \in [\tau_a^i, \tau_b^i]$ & $type = 0$
calculate $s_c = g_{\Delta s_c, \tau_b^i - \tau_a^i}(\tau - \tau_a^i)$,
$s_{min} = g_{\Delta s_{min}, \tau_b^i - \tau_a^i}(\tau - \tau_a^i)$,
$s_{max} = g_{\Delta s_{max}, \tau_b^i - \tau_a^i}(\tau - \tau_a^i)$,
$k = g_{\Delta k, \tau_b^i - \tau_a^i}(\tau - \tau_a^i)$,
$\mathbf{M} = \mathbf{M}_{s_c, s_{min}, s_{max}, k, \tau}(s)$, (b)
type =1
$Y(\mathbf{y}',n) < B_i(s_0, s_{min}, s_{max}, k, \tau, s, \mathbf{M}, type)$:
if $type = 1$
produce $Y(\mathbf{M}\mathbf{y}', n)$ (c)
$Y(\mathbf{y}',n) < D_i^{end}(x,\tau,\beta,s)$:
produce $D_i(x, \tau, G_M(\mathbf{x}', \mathbf{y}', s, t), s)$ (d)
$Y(\mathbf{y}',n)$: **if** $n = 0$ **produce** $Y(\mathbf{y}', 1)$
$Y(\mathbf{y}',n)$: **if** $n = 1$ **produce** ϵ (e)
}

Table 3: L-system table controlling the bending of stomach skeleton.

Although, parameters β and s were not written in Tab. 2, the internode segments D_a, D_A, D_b depend on segment length x, local time τ, bending angle β and arc-length (position) s.

First, the internodes D_i subdivides into internode Y with initial tangent direction $\mathbf{x}' = (0,0,-1)$, the N local bending transformations B_j, $j = 1, \ldots N$, and the terminal segment D_i^{end} storing the original parameters of segment D_i (see production in Tab. 3a). In order to know the end of the series of local bending transformations the terminal segment D_i^{end} was introduced. Modules Y, B_j and D_i^{end} have no graphical meaning and are deleted from the string when transformation is completed.

Local bending transformation B_j has all bending parameters and a flag *type* indicating that the module is processed. After subdivision of all segments the bending parameters and tangent transformation matrices $\mathbf{M}$ are updated by the explicit cubic growth function of time τ (see Tab. 3b).

Internode Y has two parameters $\mathbf{x}'$ and n which indicate the original tangent vector (vertical vector in our case) and counter. Internode Y is then sent from left to right as an signal while the used transformation matrices, i.e., used modules B_j are deleted. During movement of internode Y, the original tangent vector is sequentially multiplied by all tangent transformation matrices M^j and the result is stored again in internode Y (see Tab. 3c).

Finally, when internode Y comes just before the terminal segment D_i^{end} (after N multiplications) the bending angle is determined from the growth function G_M (see Tab. 3d). The original segment D_i with updated bending angle β is created while all temporally used symbols Y and D_i^{end} are erased (see Tab. 3e).

5.4 System of Tables

Every process describing the part of development is represented as an algebraic L-system table with prescribed number of iterations specified as an argument. The tables describing the vertical rotation and twisting of stomach have been omitted for simplicity. The simplified model of stomach growth repeatedly processes the following series of tables in each time step Δt:

Table Skeleton(1), Table Arclength(∞), Table Bending(N+4).

Resulting string of each iteration step can be represented graphically and shading methods can be used for visualization. An example of the stomach growth simulation based on L-system along with bending and rotation growth functions of the skeleton is shown in Fig. 5. The entire stomach model in Fig. 5b was scaled to fit the frame, note the variation in the size of texture. Although meta-balls were used to get the smooth appearance, creating the surface from swept-sphere rather than a collection of implicit surfaces would probably yield a smoother stomach [14].

6 Conclusions

The proposed growth model of the human embryo stomach produces convincing simulated images. Particular simulations also show considerable likeness to real human stomachs at different stages of development. In this paper, we introduced a method for the modeling and simulation of human or animal organs obeying the real biological movements known to embryologists.

We designed the L-system and the growth functions of the stomach model. The proposed framework is also

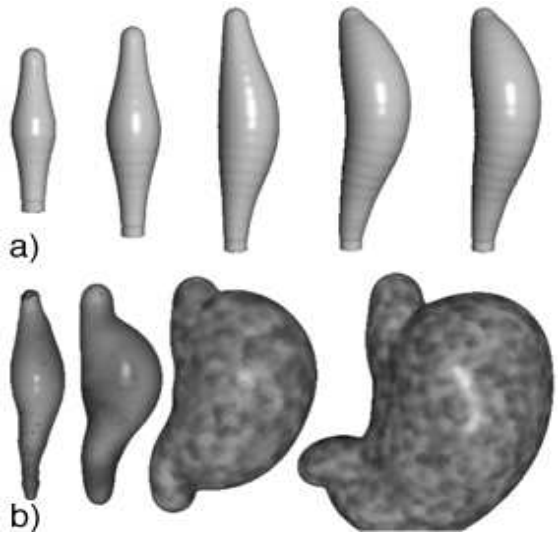


Figure 5: Three-dimensional model of the stomach growth was modeled with proposed L-system while meta-balls where used to get the smooth appearance. a) Creation process of the stomach at 4th week of development. b) Later stages of the model (up to 8th week) based on bending growth functions used in our L-system.

designed for simulation of the growth of other organs. Our current research concerns development of a model of the growth of human gut with all its complicated movements.

7 Acknowledgment

Authors wish to thank 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 Japanese Ministry of Education, Sport and Culture (MONBUSHO).

References

[1] H. Abelson and A. A. diSessa. *Turtle geometry.* MIT Press, Cambridge, MA, 1982.

[2] M. Aono and T. L. Kunii. Botanical tree image generation. *IEEE Computer Graphics and Applications*, 4(5):10–34, 1984.

[3] Alan H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, January 1981.

[4] Alan H. Barr. Global and local deformations of solid primitives. In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 21–30, held in Minneapolis, Minnesota; 23–27, July 1984.

[5] J. Bloomenthal. Modeling the mighty maple. *ACM Computer Graphics*, 19(3):305–311, 1985.

[6] C. M. Liddell and D. Hansen. Visualizing complex biological interactions in the soil ecosystem. *The Journal of Visualization and Computer Animation*, 4:3–12, 1993.

[7] A. Lindenmayer. Mathematical models for cellular interaction in development, parts i and ii. *Journal of Theoretical Biology*, 18:280–315, 1968.

[8] R. Měch and P. Prusinkiewicz. Visual models of plants interacting with their environment. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 397–410, held in New Orleans, LA, August 1996.

[9] P. Prusinkiewicz, M. S. Hammel, and E. Mjolsness. Animation of plant development. In *Proceedings of SIGGRAPH '93 (Anaheim, California, August 1-6, 1993) in Computer Graphics Proceedings, Annual Conference Series*, pages 351–360, New York, August 1993. ACM SIGGRAPH.

[10] P. Prusinkiewicz, M. James, and R. Měch. Synthetic topiary. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, pages 351–358, held in Orlando, Florida, July 1994.

[11] P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants.* Springer-Verlag, New York, 1990.

[12] T. W. Sadler. *Langman's medical embryology, 7th edition.* Williams & Wilkins, New York, 1995.

[13] A. Tunbridge and H. Jones. An l-system approach to the modelling of fungal growth. *The Journal of Visualization and Computer Animation*, 6:91–107, 1995.

[14] J. J. Wan Wijk. Ray tracing objects defined by sweeping a sphere. In *Proceedings of EUROGRAPHICS*, pages 73–82, held in Copenhagen, Denmark, September 1984.