

Integrating Shape and Pattern in Mammalian Models

Marcelo Walter*
Unisinos – Brazil

Alain Fournier
Imager – UBC – Canada

Daniel Menevaux†
Laboratoire SIC – France

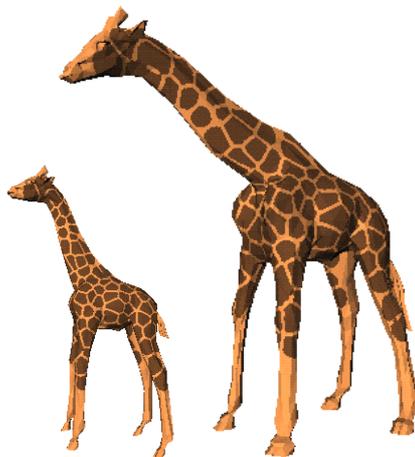


Figure 1: A giraffe and its newborn generated with techniques described in the paper.

Abstract

The giraffe and its patches, the leopard and its spots, the tiger and its stripes are spectacular examples of the integration of a pattern and a body shape. We present an approach that integrates a biologically-plausible pattern generation model, which can effectively deliver a variety of patterns characteristic of mammalian coats, and a body growth and animation system that uses experimental growth data to produce individual bodies and their associated patterns automatically. We use the example of the giraffe to illustrate how our approach takes us from a canonical embryo to a full adult giraffe in a continuous way, with results that are not only realistic looking, but also objectively validated. The flexibility of the approach is demonstrated by examples of big cat patterns, including an interpolation between patterns. The approach also allows a considerable amount of user control to fine-tune the results and to animate the resulting body with the pattern.

CR Categories: I.3.7 [Three-Dimensional Graphics and Realism]: Texture Synthesis I.3.5 [Computational Geometry and Object Modeling]: Object Representations

Keywords: Natural Phenomena, Texture Synthesis, Clonal Mosaic Patterns, Integration, Animal Models, Animal Patterns, Growth

* marcelow@exatas.unisinos.br

† daniel@sic.sp2mi.univ-poitiers.fr

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM SIGGRAPH 2001, 12-17 August 2001, Los Angeles, CA, USA
© 2001 ACM 1-58113-374-X/01/08...\$5.00

1 Introduction

The task of generating a computer graphics rendering of an object is traditionally broken down into two parts: the modeling of the shape or geometric attributes such as height, width, *etc.* and the modeling of the visual attributes, or how the object is going to look. A final *integration* step connects the two, that is, a visual attribute has to be defined for every point on the surface of the object.

The separation of these two modeling tasks makes the whole process highly flexible and powerful, and from a conceptual point of view easier to handle. While generally good for many classes of objects, this two-step process is nevertheless prone to problems when the geometry of the object is complex and therefore the mapping of visual characteristics to every point of the surface is non-trivial. Another potential drawback of this approach is that it implicitly assumes that there is no interplay between the two processes which define the shape and the visual attributes of an object. For some objects, however, the visual aspect is the result of an interaction between the two processes and this interaction plays a significant role on the final result. The typical examples are patterned animals such as giraffes and leopards. The pattern visible on the fur of an adult animal is the result of a much earlier process which took place while the animal was in the womb. It is important, in these cases, to model not only the individual processes themselves, but also the interplay between the embryo growing and the pattern formation process.

We present a method whereby the visual attributes are defined directly on the surface of the object and, more importantly, where we take into account the dynamic change of shape undergone by the object because of growth or other reasons. We show how a mammalian coat pattern can be generated by a biologically-plausible model simulated on the surface of a changing geometry, where the interplay between the pattern formation model and the geometry plays an important role in the final result.

2 Related Work

2.1 Mammalian Coat Patterns

An advantage of using biologically-plausible models in computer graphics is their potential to deliver more realistic simulations which can usually be translated into more realistic-looking results, particularly as procedural texture methods. In a biological context the images generated can be used as a powerful argument either against or in favour of the validity of the model [27].

The basic reaction-diffusion (RD) systems studied in biology [30, 20] can generate a set of interesting but visually limited patterns (stripes, spots, etc). Turk [32] developed the suggestion made earlier by Bard [1] about cascade RD processes, where a RD system is simulated having as a starting point a previous RD simulation. This has enabled the generation, for instance, of the pattern of large and small spots found on cheetahs. Variations on the way two or more RD processes interact can lead to many different patterns. The reticulated pattern found on giraffes is explained by the simulation of two different RD systems (spots formation and stripe formation) together into one. Turk also introduced the idea of simulating the RD process on the surface of the object being textured, an important contribution which avoids many of the problems of texture mapping. However, his method did not use information about the geometry of the model to drive the pattern mechanism.

Witkin and Kass [37] extended the range of possible RD patterns. Their main contribution was to extend the basic idea of RD by incorporating anisotropy into a RD system, a suggestion also made by Bard [1]. In their work, anisotropy is introduced by assigning different diffusion rates in the RD system as a function of direction in a local frame of reference. The control of different diffusion rates for different parts of the surface is achieved through *diffusion maps* defined by the user. In some sense, the use of diffusion maps indirectly transfers to the user the task of pattern definition since the user has to provide an array with values for the anisotropic diffusion rates. It is fair to say that while both papers show a giraffe-like pattern among their examples, neither patterns are very convincing (while the giraffe pattern is very striking, it is quite amazing how far from the true pattern even skilled artists allow themselves to go).

2.2 Integration of Pattern and Shape

Most of the work on the integration of the shape and the visual aspects of an object has been done in the context of texture mapping and focussed on how to deal with the problems intrinsic to the technique, such as texture placement and texture distortion. More than twenty years after texture mapping was first described by Catmull [5], the graphics community is still addressing such problems. This is clearly a strong motivation for researching alternative methods for pattern placement and much progress has been made.

In [16], for example, an interactive system for texture placement is proposed. The idea is to define the mapping indirectly via user manipulation of the texture while it is being placed on the surface of the object. Objects and camera are constrained to remain fixed during the whole texturing process, which can be a great limitation in some cases. Also, it is not very clear how the technique would work when the same texture is shared by many surfaces.

An alternative approach for texture placement is to paint a texture directly onto the surface of the object [14]. The objects are modeled as a collection of many very small polygons from which a parameterization of the surface can be derived. The mapping function is indirectly established by the user “painting” on the surface. There is no distortion to be corrected since there is no *a priori* texture map to be distorted. The drawbacks are that *i*) the final result is still highly dependent on the artistic abilities of the user and therefore achieving a visually elaborate texture can be difficult and *ii*) as pre-

sented, the approach does not handle a pre-existing texture, which could be useful to correct texture distortions by visual inspection.

A solution to minimize texture distortions when mapping pre-computed textures was presented in [17]. They define a metric for the distortion and try to minimize its value globally. This approach is limited since for very complex objects this minimization step might not be feasible. One alternative is to use a mapping function which is local and not necessarily continuous. In other words, the object is split into *charts* and a collection of charts is called an *atlas*. The creation of charts takes into account surface curvature and the user interacts visually to achieve the best atlas for a particular object. The idea behind charts is to represent a non-developable surface as a set of developable surfaces. A developable surface is a surface that can be deformed to planar shape without changing length measurements in it [6]. In a similar fashion, the work presented in [3] “cuts” a given 3D parametric surface into regions which can be flattened out without warping. The minimization of distortion is achieved through a compromise between cuts and distortions. A more recent approach with the same flavour was presented in [24], where texture patches are pasted onto an arbitrary surface mesh, exploring the fact that small regions of the surface are locally mappable onto a 2D plane.

When the user has control over how the texture is being generated some more effective ways to avoid distortions are possible. The method by Witkin and Kass [37] uses models described parametrically as a collection of patches and synthesizes textures with RD systems. The problem of texture distortion caused by the mapping from the texture parameter space to the surface space was solved in an integrated manner. The texture synthesis incorporates a correction factor for the distortion, that is, the diffusion rates present in the RD system were controlled to account for the geometric distortions present on the surface. This correction, however, only works for surfaces that can be described by a single parametric function, usually not the case for complex surfaces. Seamless periodic textures were created using cyclic boundary conditions, i.e., points that shared a common boundary in different patches had the same boundary conditions. In the RD context this means that the chemicals involved have the same concentrations at the boundaries of the parametric patches.

There has been little work addressing integration as a task *per se*, outside the context of texture mapping. Two remarkable exceptions are the work by Turk [32] and by Fowler, Meinhardt and Prusinkiewicz [8]. Both papers present variations on the fundamental idea of computing the pattern on the surface of an object as a “growth-in-place” procedure.

Turk’s work, as mentioned in the previous section, used RD textures. Instead of mapping the generated pattern onto a polyhedral or parametric model, his approach simulates the RD system on the surface of model, without the intermediate mapping from texture space to object space. Basically, the surface of the model is divided into cells and the RD system is simulated directly on the mesh formed by these cells. The cells for the simulation are the regions of a planar variation of a Voronoi diagram [22] computed from a polyhedral description of the model. The approach does not have the usual problems of texture discontinuity and distortion since there is just one mesh over which the RD system is simulated. Nevertheless, this solution does not make any use of the geometric information about the shape of the object being textured and therefore the texture on the surface can appear strangely regular and artificial. Moreover, to achieve a more natural-looking result, Turk introduced adjustments so that the texture could grow faster in some areas and slower in others. This was achieved with the user defining initiator-cells and different diffusion rates for different parts of the surface.

Fowler *et al.* [8] similarly approached the modeling of seashells by discretizing the growing edge of a parametric model of a shell

into polylines. Each segment of the polyline is treated as a cell for the one-dimensional RD simulation. The geometric and visual attributes of the shells are easier to integrate since both shape and texture can be unequivocally expressed as a function of time. The integration is straightforward since time can be used as an integration factor. Their exceptional results suggest that the use of an integrated approach is, in some cases, not only useful but imperative.

Another closely related work is by Fleischer *et al* [7] on cellular texture generation. In many ways their work is more general than ours, allowing for full programmable and context-sensitive behaviour of the cells creating the textures. As usual in modelling, however, too much power can be a bad thing as there is no obvious relationship between the equations governing the cells and the results. In other words it is possible to generate an extraordinary range of models, but hard to see what they are models for. Our overall goal is to match for mammalian bodies and coat patterns the level of achievement reached by Fowler *et al*. There will be many critical differences in the methods used. First, we do not believe that mammalian coat patterns are well modelled by RD methods, and we use a different model, called *Clonal Mosaic*, summarized below. Second, the process obviously has to take place at least in two dimensions corresponding to the skin surface. Finally, since in mammals the pattern is established in the fetal stage, and undergoes changes due to body growth both before and after birth, we have to integrate pattern formation and growth in an effective way.

3 Basic Tools and Data

3.1 How and When is the Pattern Established

The coloured pattern in mammalian coats is produced by the various colours of the hair constituting the *pelage*, as the collection of hair is called [29]. The skin of mammals is made of two layers, a superficial layer called the *epidermis*, and an inner layer called the *dermis*. The epidermis is composed of flattened cells, and hair is produced by an *invagination* of the epidermis called a *follicle*. The hair is formed by division of cells in the *bulb* at the base of the follicle. Pigments giving the hair colour are incorporated into the hair by *melanocytes*, basal cells of the epidermis specialized in the production of *melanin* [26]. Melanins are polymers synthesized from *thyroxine* (an amino acid) and have colour ranging from pale yellow to black, through buff, red-brown and brown. Basically, the colour of the hair is determined by the amount and the nature of the produced melanin. This same melanin contributes to the colouration and protection of the skin in humans as well. The melanocytes have migrated during embryonic development from the *neural crest* to their final position in the epidermis as part of a complex of cells called an *epidermal melanocyte unit*.

Direct observation of mammals shows that their characteristic coat pattern is already established at birth, and after this is modified only due to differential growth of the body. For instance, the spots on an adult giraffe can easily be recognized from the spots of the same individual at birth. Note that sometimes the spots can fade or disappear due to a change in “colour map”. For instance, lions have spots at birth, which quickly fade and are not visible in the adult. Therefore, we can distinguish two phases in the creation of the pattern. The first phase happens in the fetal stage, where both growth and pattern formation take place. The second phase happens both before and after birth. In this phase the pattern formation process is over and only differential growth affects the pattern. Of course the pattern itself is only visible after the pelage has grown, but we can assume that the distribution of the melanocytes responsible for the colour has already taken place.

We have estimated the bounds for the starting time of the first phase and second phase as follows [33]. The start of the pattern formation process is most likely as soon as the melanocytes have

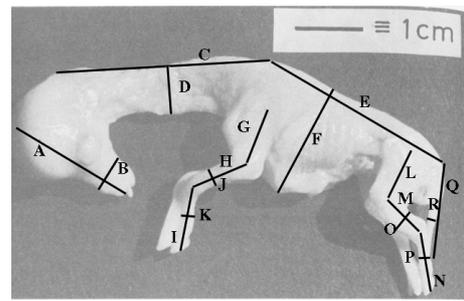


Figure 2: A 35-45 days giraffe embryo (457 days gestation) from [21].

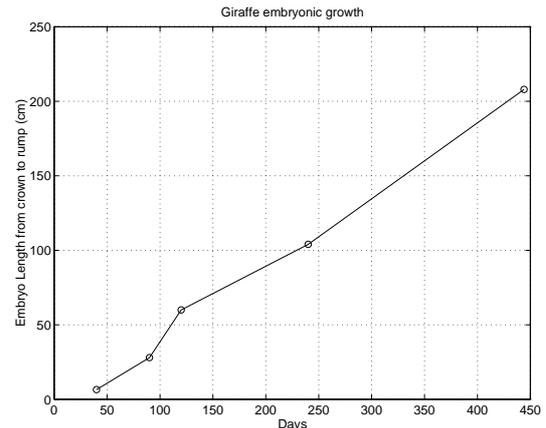


Figure 3: Fetal Length for Giraffes.

finished their migration from the neural crest, about 35 days for the giraffe (out of 457 days for the total gestation time). At this time the fetus already has a recognizable shape, as can be seen in Figure 2, which shows a giraffe embryo with an estimated age of 35 to 45 days.

The upper bound for the end of phase one is the end of gestation, but there are reasons to believe that it might be sooner. Murray [21], from considerations based on a RD model, deduced that the total duration of the pattern formation phase is quite narrow, but that does not necessarily apply to other models, such as clonal mosaic. In our simulations we have generally assumed a time of about 200 days for the giraffe (about half of the gestation time). A very useful fact is that during this whole time we can safely assume a linear growth. Figure 3 shows the plot obtained for the length (from crown to rump, in centimeters) of the giraffe fetus from measurements published in [23] and [2]. We would like to note that it actually is a distinct advantage to “go back” to the fetus, since we have only to use a *canonical* fetus, which will then be grown to the dimensions of a given individual by the process described in Section 3.3.

3.2 The Clonal Mosaic Model

The model is explained and illustrated in a paper by Walter, Fournier and Reimers [35] and we will only stress the main points here. The Clonal Mosaic theory for mammalian coat pattern formation proposes that the typical yellow-black striped and spotted patterns occurring in several species of mammals, reflect a spatial arrangement – a mosaic – of epithelial cells which derive from a single progenitor, i.e., they are clones. Hence we use the name Clonal Mosaic (CM). Different hair colors result from different types of underlying cells. The model takes into account important and re-

cent biological experimental data such as the migration of and interactions among cells [11, 28], particularly epithelial cells [13].

The model is appealing for procedural texture synthesis in computer graphics since it can provide a large number of 2D patterns with a relatively small number of parameters. These patterns can be used inside a traditional texture mapping framework. Another strong appeal of the CM model from a computer graphics point of view is its reasonably straightforward extension for simulation over arbitrary surfaces. Thus, it is possible to generate patterns directly on the object's surface, without the mapping step. In an integrated framework using the CM model as the texture generator, the geometry of the object can then play an important role in the patterns generated, increasing the realism of animals synthesized this way.

3.2.1 Implementation

The patterns we will use are generated by a simulation of the CM process. For all patterns of interest we use a maximum of three types of cells named foreground, middleground and background. The type of a cell defines its behaviour in the system. Table 1 summarizes the main parameters for cells and their meanings.

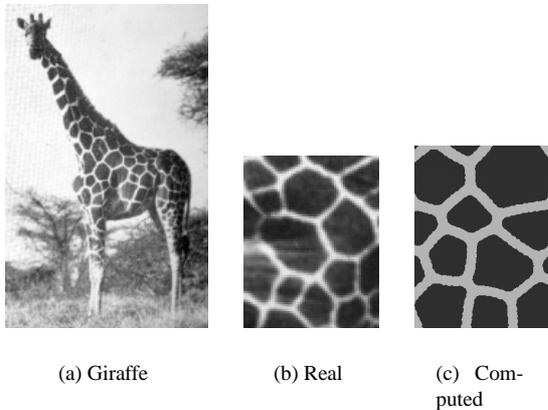


Figure 4: Pattern on giraffe.

Attribute	Meaning
Color	RGB
Division Rate	Mean time between divisions Controls the absolute and relative numbers of cells of a given type
Initial Probability	Probability to be of a given type
Mutation Probability	Probability to switch to other type
Adhesion	Drag between types Controls the tendency of cells to stay together

Table 1: Attributes of a cell.

Cells are modeled as points for computing purposes. Points are usually the first choice to represent a biological structure such as a cell [12]. To turn points into a tessellation of the surface, we compute their Voronoi polygons. The Voronoi polygon of a point in a given domain is the region of the domain that contains all the points closer to that particular point than any other [25]. An initialization step randomly distributes among background cells a number of foreground cells. The same result can be accomplished by having background cells switch to foreground cells in a controlled manner. Cells then divide at rates that are characteristic of their type (usually the foreground cells divide faster than the background cells). As new cells appear they try to maintain a constant area (in fact a

volume, but we assume that the thickness of the layer is constant) by repulsing their immediate neighbours. The repulsive force in the model is a linear function of the distance to the neighbours, reaching zero at a distance equal to the *repulsive radius of influence* of cells. Relaxation steps allow time for the cells to repel each other.

For each cell and each relaxation step the resultant force causes a displacement in the direction of the force and proportional to both the force intensity and to the adhesion. The adhesion factor is characteristic of the pair of cells interacting. The new position of the cell is then computed from the displacement. This is done in parallel for all the cells. In order to model anisotropic displacement, the actual cell displacement is the weighted sum of the previously computed displacement and its projection in a given anisotropy vector. For the sake of efficiency the relaxation step is not applied at every cell division event, but on a user controlled schedule defined before the simulation starts and kept constant for the whole simulation.

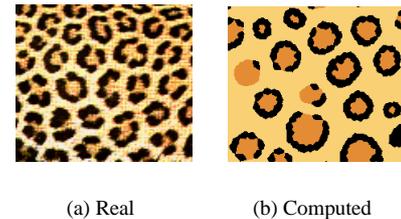


Figure 5: Rosettes.

This model and its simulation can produce a large range of patterns closely resembling mammalian patterns, such as the spots of the giraffe (shown in Figure 4), the stripes of the tiger (with anisotropy), the rosettes of the leopard (shown in Figure 5) and of the jaguar with three types of cells and controlled mutation, i.e., when a given cell splits it can be of a different type than its parent. A fundamental *motif* of these patterns is a Voronoi diagram structure, seen in its purest form in the reticulated giraffe (*Giraffa camelopardalis reticulata*). All these results were computed in a square domain. The similarity of the results and the real patterns was verified with statistics on size distribution and closeness to an ideal Voronoi diagram (see [33] and [35]).

3.3 Controlling Shapes and Pattern Growing Parameters

With only a very small loss in generality, we assume that the patterns will be integrated with polygonal models of the animal bodies. Since the body has to be grown over a span of time which covers gestation up to adulthood, we need a flexible system to apply and control the growth, but at the same time we need to be able to deal with very sparse data. We will use the techniques described by Walter and Fournier in [33, 34] and briefly summarize them here.

In that work they presented a technique to transfer growth data to polygonal models of animals. For each section of the body that will grow and generally be transformed independently, a cylindrical coordinate system is attached to it. The cylinder is positioned so that it encloses that part of the body it controls (this is not strictly necessary, but convenient for the user).

These cylinders are predefined by the user, and are designed so that they segment the object into *coarse groups* of geometry that represent the object in a higher structured level. For each cylinder it is also defined a pair of *features* which indirectly control the cylinder's scale. Each feature is a length measurement between two specific points on the body. Once this structure of cylinders and features is in place, the user can modify any body part or the whole by acting on the features. In the case of growth simulation the fea-

tures are normally chosen so that they correspond to the available or obtainable growth data.

The cylinders are part of a classical modelling hierarchy. Figure 6 shows an example of the 18 cylinders defined for a giraffe's model. The blue sections of the body are the triangles which belong to more than one cylinder. For clarity reasons the features are not shown in this figure.

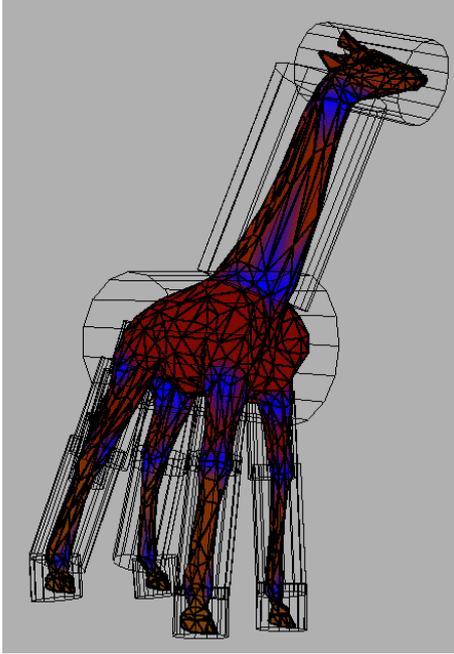


Figure 6: Giraffe body and 18 control cylinders.

The main difference between a normal transformation hierarchy and the one used for controlled transformation such as growth is that since the growth data available is generally expressed in absolute terms, only translation and rotation are inherited from the ancestors in the growing hierarchy, but the scaling is local and in absolute terms. Each cylinder A has its own canonical coordinate system, and an associated matrix to transform to the world coordinate system:

$$M_{W \leftarrow A} = [T_{W \leftarrow A} R_{W \leftarrow A} G_{W \leftarrow A}]$$

where $T_{W \leftarrow A}$ is a translation matrix, $R_{W \leftarrow A}$ is a rotation matrix and $G_{W \leftarrow A}$ is a scaling matrix, the growth matrix. The growth matrix is given by:

$$G_A = \begin{bmatrix} L_A & 0 & 0 & 0 \\ 0 & R_A & 0 & 0 \\ 0 & 0 & R_A & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where R_A and L_A are the world radius and length of the cylinder, which can be easily derived from such measurements as the length and girth of features of the real animal or its pictures (see for instance the black lines labelled A through R used for the giraffe fetus shown in Figure 2).

The operation to transform a point P_B in feature B (associated with a cylinder B) to a point P_A in the coordinate system of its parent feature A (associated with cylinder A) is:

$$P_A = [T_{A \leftarrow B} G_A^{-1} R_{A \leftarrow B} G_B] P_B$$

where:

$$T_{A \leftarrow B} = [M_{A \leftarrow W} T_{W \leftarrow B} T_{A \leftarrow W} M_{W \leftarrow A}]$$

One then applies these transformations to convert to the coordinate system of the ancestor of A , until the root is reached, at which point we have the point in world coordinate system. The growth process then consists in applying these transformations to each needed vertex for each time at which we have measurements for the feature. The growth data can be interpolated between time using any suitable interpolating formula.

To guarantee continuity as the shape changes, and to achieve a degree of smoothness in the resulting surface, the cylinders have to overlap, and therefore one has to decide how to weight the influence of the cylinders on a vertex which belongs to more than one. The solution proposed uses a weight inversely proportional to the distance from the vertex to the axis of the cylinder.

It is important to note that applying the technique has two practical consequences: any measurement from a real animal once applied to the polygonal model has the effect of changing the proportions of the original model (which could be an idealized or otherwise fictitious animal) to the proportions of the real measured animal. That means that for a single model the technique can create individual bodies with their own measurements [36]. The second point is that the same hierarchy can be used for animation, since it allows for independent rotations and translations to be applied to the various local coordinate systems enclosing the body.

4 Integration of Patterns and Shape

4.1 Simulation of the CM Model on an Arbitrary Surface

This section describes how the CM model presented in Section 3.2 is integrated with the shape control presented in the previous section. This will allow the synthesis of CM patterns directly on a shape-changing geometry, such as the body of the animal growing and illustrated in Figure 12. Figure 7 shows a schematic representation of the whole process of pattern formation development in connection with the body growing.

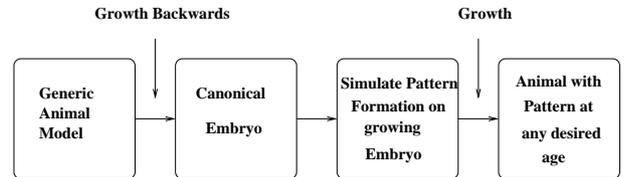


Figure 7: Pipeline of the system.

We describe below the steps of this process in turn.

- **Deriving Cell Splitting Rates from Growth Information**
Our patterns can be viewed as many small cells multiplying to form a specific arrangement – the growth of a tissue. Shape can be indirectly expressed through growth-curves and the final shape of a given natural object is a result of cells dividing at different rates. It is reasonable therefore to use growth as an integration factor driving the patterns and the changes in shape. In practice we need to be able to compute how fast the initial set of cells should split in order to keep up with the increase in area of the object.

For the following description we will assume, without loss of generality, that we only have 3 types of cells called F , B , and M . The total number of cells is $N_t = \sum N_i$ where $i = F, B, M$. At time $t = 0$ we have the individual area for a single cell as $a = \frac{A_0}{N_0}$ where A_0 is the initial area of

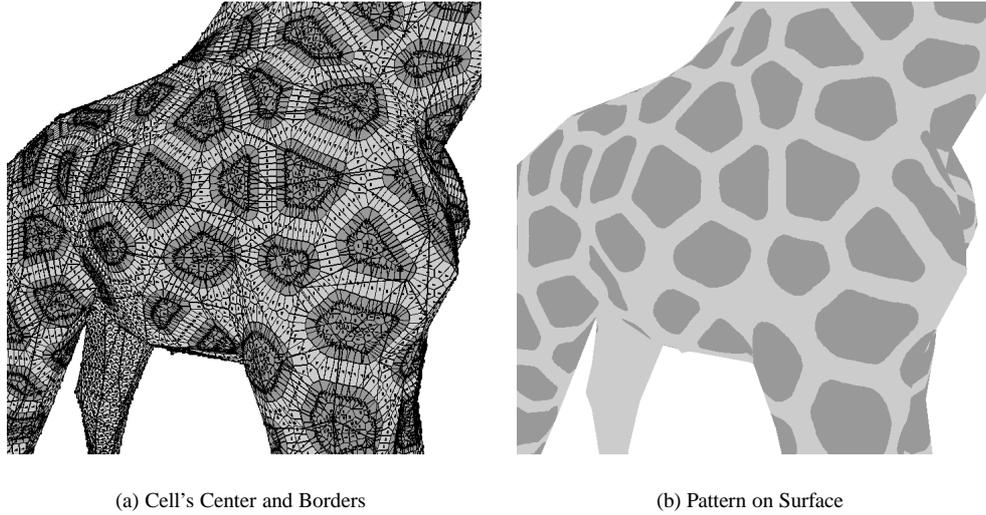


Figure 8: Voronoi diagram of cells generated directly on the polygonal mesh.

the surface of the model, usually the area of the model for the time when we estimate that the pattern formation process starts, and N_0 is the arbitrary initial number of cells. The goal is to keep the area of a single cell constant as the model grows. We also have to establish the relative rates of splitting between the different types of cells. We will call these rates g_F, g_B, g_M , with $\sum g_i = 1$. The net increase in the number of cells is proportional to the increase in the area of the model, that is $\Delta N = \frac{\Delta A}{a}$. A factor $k = \frac{\Delta N}{\sum N_i g_i}$ is computed to express the net increase in the number of cells among the different types and the net increase in the number of cells for each type is $\Delta N_i = k N_i g_i$. The instantaneous rate of splitting is $s_i = \frac{\Delta N_i}{N_i \Delta t}$ and the rate r_i at which we need the cells to split is the reciprocal of s , that is, $r_i = \frac{\Delta t N_i}{\Delta N_i}$.

- Triangulation of the Model**
 The animal models we are using are represented by triangular meshes. In the context of the CM model the use of triangles guarantees that the tessellation provided by the Voronoi polygons, which expresses the pattern, matches exactly the original triangular mesh, since triangles are planar. The more complex mesh formed by the Voronoi polygons exactly replaces the original triangle mesh, i.e., the pattern *is* the surface. This has a distinct advantage to be explored in animation tasks, since the pattern will just follow whatever animation information is applied to the model.
- Distributing Random Points on the Surface of a Polyhedral Model**
 The initial distribution of random points representing the cells on the surface of the polyhedral model representing the animal is implemented with an algorithm presented by Turk [31]. This algorithm guarantees an uniform distribution of points on the surface based on the relative areas of the triangles describing the surface.
- Relaxation of Points on the Surface of the Model**
 The relaxation process used to maintain cell size has to deal with the cells defined on the surface of the model. We have to be able to compute distances on the surface of the model between cells that are neighbours. We use approximated distances as explained below. With an increase in cost we could use more sophisticated solutions for computing the shortest

path between cells, such as the one presented by Lee *et al.* in [15]. Ideally, all faces which have cells within the area

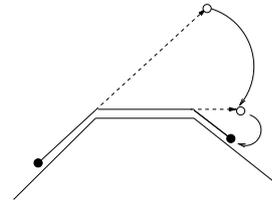


Figure 9: Mapping cells from face to face.

defined by the repulsive radius should be considered. This would imply an arbitrary, possibly large, number of neighboring faces. To avoid this cost we have restricted the search for neighbours among the faces which share either an edge (called *primary* neighbours) or a vertex (called *secondary* neighbours) with the face in question. That is equivalent to stating that the shortest height of a triangle is longer than the effective radius of repulsion. In most practical cases this limitation does not significantly affect the results since the ratio between the number of cells versus number of faces guarantees that all neighboring cells are living in either primary or secondary faces. Ideally, we would have to guarantee that all cells in tertiary faces (not primary neither secondary neighbours) are at a distance greater than the radius of repulsion. If there is a minimum angle for the triangles and a minimum edge length then one can show that there is a minimum distance between any point of the face in question and any tertiary face.

The distances are computed on the plane of the face where the cell lives. Therefore, we need a way of mapping all neighboring cells to this plane. For each pair of faces P and N that share an edge we precompute the two rotation matrices $M_{P \rightarrow N}$ and $M_{N \rightarrow P}$ which bring a point defined on the plane of P to the plane of N and vice-versa. For primary neighbours the mapping is trivial using the precomputed matrices. For secondary neighbours we use a sequence of rotations around the edges which define a path between the face in question and the secondary neighbour. In order to update the cell's position due to the relaxation forces, cells travel freely and can eventually move to another face. When a cell changes

face we find which edge the cell crossed and using the pre-computed rotation matrices we bring the cell's position onto the plane of the new face. This process is repeated until the cell rests on some face, as illustrated in Figure 9. The amount of traveling a given cell undergoes is arbitrary and for some geometries it is possible that the cell returns to its original face (for example, on a thin giraffe leg).

- **Computing the Voronoi Diagram on a Surface**

Okabe [22] defines a *polyhedral Voronoi diagram* as a Voronoi diagram where the sites are defined on the surface of a polyhedron and the distances are measured on this surface. Computing the exact polyhedral Voronoi diagram can be quite computationally expensive since for any two arbitrary sites on the surface there are many possible paths and finding the shortest one is a difficult problem [9]. For this reason, some solutions to this problem make use of approximations to the real Voronoi. The solution proposed by Mount [19], for instance, computes a polyhedral Voronoi diagram where the sites are used to create new polygons and a path between two sites always goes through edges of the polyhedron.

We use an approximation for the actual Voronoi diagram where the final diagram for the whole polyhedron will be the combination of the individual Voronoi diagrams computed for each cell in the triangles comprising the polyhedron. There is some redundancy involved, since the elements of the Voronoi diagram between cells on different triangles will be computed twice (or more for triangles sharing only vertices), but this is more than offset by the fact that triangles and their neighbours are in effect used as *buckets* to limit the search for neighbours.



Figure 10: Giraffe Cube.

Turk used a similar idea in his RD work [32] but in his case he did not need to explicitly maintain the diagram since it was only used indirectly to establish diffusion amounts between cells. In our case, the pattern is defined by the Voronoi diagram and therefore we need to build and maintain the Voronoi cells as a whole. The deviation from an exact computation of a Voronoi diagram is not critical in our case for two reasons. First, the pattern is defined through a large number of cells per triangle, which means that the approximated solution is correct for all cells but possibly the ones that are closer to the edges of the face. Second, the pattern is defined by the overall combination of many cells, which possibly spread over many faces. It is possible that the cells with Voronoi errors are “inside” a given pattern element and therefore are not individually visible.

In order to compute the Voronoi diagram for all cells resting on a single face we map all neighboring cells to the plane of the face in question using the approach explained above. With all cells on the same plane, we compute the Voronoi diagram on this plane. The Voronoi polygons are then clipped against the edges that define the face. Figure 8 illustrates this process. In (a) we show both cell's centers and the Voronoi borders and in (b) we show the end result with the overall pattern. There are two special simpler cases that we should mention but in practice are often negligible. When a face has zero or one cell only, it means that the Voronoi polygon for this face is the face itself.

4.2 Pattern Generation without Growth

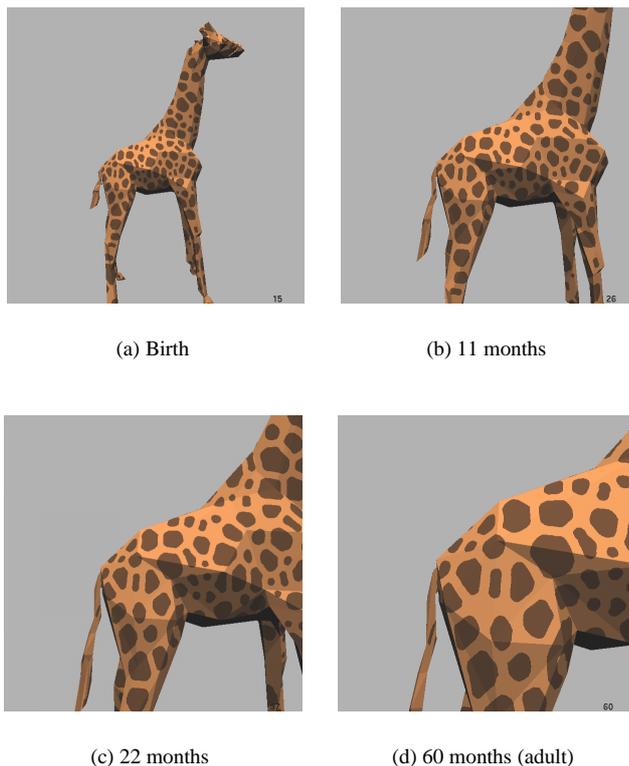


Figure 11: Giraffe Growth.

Although the CM model was designed to work in conjunction with a shape changing geometry, we can also use it as a pattern generation mechanism on a static body model. The pattern formation process in this case is driven by pre-defined parameters, that is, splitting rates of cells are not computed from growth information. Figure 10 shows the result of simulating a giraffe-like pattern on the surface of a cube. The spots are distributed over the whole shape without visible discontinuities and are very similar to the pattern on the real giraffe or the two-dimensional simulation (see Figure 4).

Once a pattern is computed on a static model, we can still apply the growth transformation in order to obtain models at different ages with the same pattern. This is illustrated in Figure 1 where we show an adult and a newborn giraffe with the same pattern. In order to adapt the pattern computed for an adult body to a newborn body we apply the same growth transformation applied to the adult model to all cells and re-compute the Voronoi expressing the pattern. As a result the pattern can be output as uniformly coloured convex poly-

gons (the cells of the Voronoi diagram). This is the approach used in the images in the paper. Alternatively, to save space, if there is a large number of cells per triangle, the original triangles of the mesh can be output with their colour pattern as a local texture.

The real measurements were determined from photographs of real giraffes and averaged from a few individuals. Figure 11 shows yet another result illustrating the growth after the pattern is already established. The giraffe body is grown from the newborn proportions to the adult (about 5 years old). It is important to note that if the pattern is close to a Voronoi diagram, and if some parts of the body grow anisotropically, such as the neck, whose length grows by a factor of 4 while its diameter grows by a factor of 3, then the final pattern cannot be a true Voronoi diagram. This can be observed on Figure 1 (even though it is not obvious without measurement), and more importantly has been verified experimentally.

4.3 Pattern Generation with Growth

In this section we show the results of simulating the CM model with splitting rates being computed from the growth information as discussed on item 1 of Section 4.1. The whole point of integrating the pattern formation and growth is that the two are tightly linked in the fetal stage and hopefully this approach will create more accurate models of patterned animals.

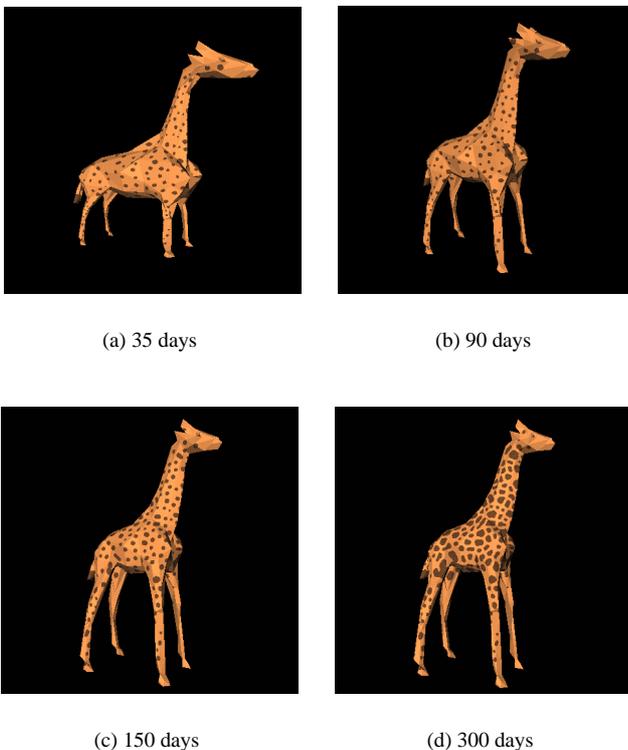


Figure 12: Pattern Development.

Figure 12 shows four phases in the development of the giraffe pattern on the fetus at 35 days (estimated start of pattern development), 90 days, 150 days and 300 days. It should be compared to the pattern of the giraffe at birth. Notice that the shape gradually approximates the proportions of the newborn while the spots get bigger. Figure 13 shows the same fetus at 35 days, but put in a position similar to the real one shown in Figure 2, for easier comparison (we do not suggest that the pattern is actually that visible on the embryo, just that it is there). From the fetus to the adult, the

body size grows by a linear factor of 100, from 3cm to 3m.

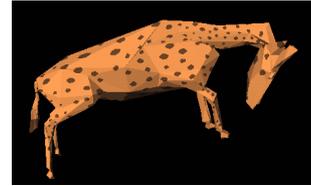


Figure 13: Fetus and cells at 35 days in fetal position.

5 Control of Parameters

From our simulations it is clear that differences in growth rates alone do not explain the differences in spot sizes, for instance between the main body and the leg. Other factors, such as the anisotropy direction for the tiger stripes and the specific markings on the face, show clearly that there are phenomena external to the CM system that affect the parameters. To deal with this, and to also be able to introduce arbitrary effects (such as writing our names with the spots of the cheetah), we introduced a way to control externally the parameters of the CM model. Since the cylinders are already in place to control the shape and growth, they are also convenient to provide support for textures which can control any of the parameters used in the CM system. Each cylinder can point to many ordinary image files, with a range to map the texel value to CM parameters. It is of course convenient that the cylinders have a natural

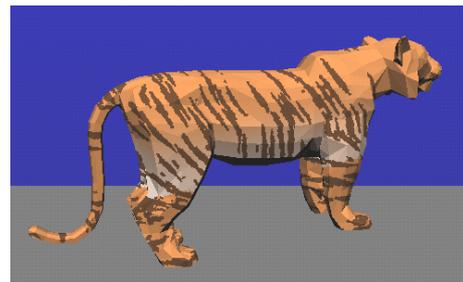


Figure 14: Tiger anisotropy and colour controlled by textures.

parameterization, being developable surfaces. For each point of the polygons, its cylindrical coordinates are determined from the intersection of a line between the point and the location on the cylinder axis with the same x (coordinate along the cylinder axis) and the cylinder. This is mapped to the $(0, 1)$ range and used to index into the texture file. This is very similar to the technique of *two-pass texture mapping* described by Bier and Sloan [4].

Figure 14 shows a striped tiger (actually elongated spots), where the orientation of stripes and colour of body parts was controlled by textures on the cylinders. Another use of this is illustrated on Figure 1 where we prevented spots from appearing on the head and lower parts of the legs of the giraffe.

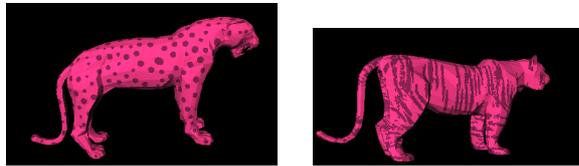
6 Conclusions and Future Work

We presented an approach that integrates a biologically-plausible pattern generation model with a body growth and animation system that can use experimental growth data to produce individual bodies and their associated patterns automatically.

We used mainly the example of the giraffe, because the reticulated giraffe presents a pattern that can be objectively validated, and also because of the tremendous change in overall size and proportions between the fetus and the adult. We showed as well that the

Parameters	ρ	wr	$time$	wd	mitosis F	mitosis B	α FF	α BB	anisotropy
Giraffe (fig. 1)	18	2.4	80	0.067	10	150	0.9	0.6	0.0
Pink panther (fig. 15)	18	2.6	10	0.066	10	120	0.9	0.2	0.0
Pink tiger (fig. 15)	18	1.1	80	0.01	10	50	0.7	0.2	0.5
Interpolation (fig. 16)	18	1.85	45	0.038	10	85	0.8	0.2	0.25

Table 2: Parameters for some of the computed patterns.



(a) Spots

(b) Stripes

Figure 15: Pink panther and pink tiger.

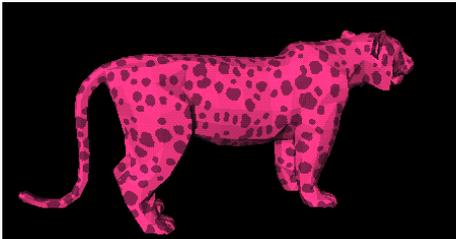


Figure 16: Interpolation between pink panther and pink tiger.

system can “do” most of the big cats, even though more detailed work is still needed in this respect. The method also allows a considerable amount of user control through cylinders which control the growth, the animation and the parameters through textures.

There are many avenues left for extensions of this work. Since the parameters for different patterns are part of the same set, it is very easy to interpolate between different patterns by interpolating between parameters. Of course, there is no guarantee that the pattern will look to us as a legitimate intermediary between the two extremes. As a simple example, consider the pink panther and the pink tiger shown in Figure 15. The interpolated result is shown in Figure 16. For these kind of results the parameters are interpolated beforehand and a CM simulation is triggered with the new parameter values. It is clear in this case that the interpolated pattern does not yet exhibit much anisotropy, but the number of spots has visibly increased. In this example we kept the same polygonal mesh for both bodies (the tiger), even though it is possible to morph the bodies as we morph the pattern.

Another interesting immediate possibility is to try to synthesize a giraffe pattern directly as a Voronoi diagram, overpassing a CM simulation and considering the fact that the overall giraffe pattern is similar to a Voronoi diagram. On another front, we want to use the texture control to generate realistic face markings (again, the CM model will not be directly responsible for it). Our long term goal is to extend the morphing capabilities of the system by using *generalized cylinders* instead of ordinary cylinders to better control the growth and the shape. This will also allow us to define the shape parameters from real animals and pictures thereof in a more powerful way. Naturally, in mammalian coats the pattern is expressed as fur, and we plan to use our system to assign colour to simulated fur, as the final look very much depends on the *filtering* role of hair. The fur can be distributed on our polygonal models in a manner similar to the one described in work by Van Gelder and Wilhelms [10]. In addition, we have a method to control the behaviour of the entire

pelage while allowing individual motion to single hairs. Finally, a more thorough exploration of the parametric space (maybe *a la* Design Galleries [18]) for the CM model will certainly be rewarding.

Table 2 shows the CM parameters used for some figures. There ρ is the number of relaxation steps per simulated day, wr is a weight applied to the ideal radius within which cells can cause a displacement, $time$ is the total simulated time (days), wd is the strength at which a cell repulses its neighbours, mitosis are in average days between divisions, α is the adhesion (all cross-adhesions were set to 0.0), and the anisotropy is the length of the anisotropic vector (to be compared to wd). In the tiger images, the displacement is 80% in the anisotropy direction.

Acknowledgments

Michael McAllister provided Voronoi code for our patterns, while many people in Imager and the anonymous reviewers provided helpful suggestions. Manuel Menezes de Oliveira Neto provided careful proofreading. The first author gratefully acknowledges the financial support of CNPq – Brazil and FUNDEPE (Unisinos). We gratefully acknowledge the support of the Canadian National Science and Engineering Research Council through Research Grants.

To Alain Fournier for his guidance, support, knowledge, and inspiration.

References

- [1] J. B. L. Bard. A model for generating aspects of zebra and other mammalian coat patterns. *Journal of Theoretical Biology*, 93(2):363–385, November 1981.
- [2] E. Beddard. Description of the external characters of an unborn fetus of a giraffe. *Journal of Zoology*, pages 626–631, 1906.
- [3] C. Bennis et al. Piecewise surface flattening for non-distorted texture mapping. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH 91 Conference Proceedings)*, volume 25, pages 237–246. Addison-Wesley, July 1991. ISBN 0-201-56291-X.
- [4] E. A. Bier and K. R. Sloan, Jr. Two part texture mappings. *IEEE Computer Graphics and Applications*, 6(9):40–53, September 1986.
- [5] E. E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. Ph.D. thesis, University of Utah, December 1974.
- [6] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, 1990.
- [7] K. Fleischer, D. Laidlaw, B. Currin, and A. Barr. Cellular texture generation. In Robert Cook, editor, *Computer Graphics (SIGGRAPH 95 Conference Proceedings)*, pages 239–248. Addison-Wesley, August 1995. ISBN 0-201-84776-0.

- [8] D. R. Fowler, H. Meinhardt, and P. Prusinkiewicz. Modeling seashells. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH 92 Conference Proceedings)*, pages 379–387. Addison-Wesley, July 1992. ISBN 0-201-51585-7.
- [9] W. R. Franklin, V. Akman, and C. Verrilli. Voronoi diagrams with barriers and on polyhedra for minimal path planning. *The Visual Computer*, 1:133–150, 1985.
- [10] Allen Van Gelder and Jane Wilhelms. An interactive fur modeling technique. In *Proceedings of Graphics Interface 1997*, pages 181–188, May 1997.
- [11] N.S. Goel and G. Rogers. Computer simulation of engulfment and other movements of embryonic tissues. *Journal of Theoretical Biology*, 71:103–140, 1978.
- [12] R. Gordon. *Computational Embryology of the Vertebrate Nervous System*, pages 23–70. Elsevier Biomedical Press, 1983.
- [13] R. Gordon and A. G. Jacobson. The shaping of tissues in embryos. *Scientific American*, pages 106–113, June 1978.
- [14] P. Hanrahan and P. Haeberli. Direct WYSIWYG painting and texturing on 3D shapes. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH 90 Conference Proceedings)*, volume 24, pages 215–223. Addison-Wesley, August 1991. ISBN 0-201-50933-4.
- [15] A. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution mesh morphing. In Alyn Rockwood, editor, *Computer Graphics (SIGGRAPH 1999 Conference Proceedings)*, pages 343–350. Addison-Wesley, August 1999. ISBN 0-201-48560-5.
- [16] P. Litwinowicz and G. Miller. Efficient techniques for interactive texture placement. In Andrew Glassner, editor, *Computer Graphics (SIGGRAPH 94 Conference Proceedings)*, pages 119–122. Addison-Wesley, July 1994. ISBN 0-89791-667-0.
- [17] J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH 93 Conference Proceedings)*, volume 27, pages 27–34. Addison-Wesley, August 1993. ISBN 0-201-58889-7.
- [18] J. Marks et al. Design galleries: A general approach to setting parameters for computer graphics and animation. In Turner Whitted, editor, *Computer Graphics (SIGGRAPH 97 Conference Proceedings)*, volume 25, pages 389–400. Addison-Wesley, August 1997. ISBN 1-89791-896-7.
- [19] D. Mount. On finding shortest paths on convex polyhedra. Technical Report 1495, Department of Computer Science, University of Maryland, 1985.
- [20] J. D. Murray. On pattern formation mechanisms for lepidopteran wing patterns and mammalian coat markings. *Philosophical Transactions of the Royal Society of London B*, 295(1078):473–496, October 1981.
- [21] J. D. Murray. A pre-pattern formation mechanism for animal coat markings. *Journal of Theoretical Biology*, 88:161–199, 1981.
- [22] A. Okabe, B. N. Boots, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley & Sons, 1992.
- [23] R. Owen. Notes on the birth of the giraffe at the zoological society gardens. *Transactions of the Zoological Society of London*, 3:21–28, 1849.
- [24] E. Praun, A. Finkelstein, and H. Hoppe. Lapped textures. In Kurt Akeley, editor, *Computer Graphics (SIGGRAPH 2000 Conference Proceedings)*, pages 465–470. Addison-Wesley, July 2000. ISBN 1-58113-208-5.
- [25] F. Preparata and M. Shamos. *Computational Geometry - An Introduction*. Springer-Verlag, 1985.
- [26] G. Prota. *Melanins and Melanogenesis*. Academic Press, 1992.
- [27] P. Prusinkiewicz. Modeling and visualization of biological structures. In *Proceedings of Graphics Interface 1993*, pages 128–137, May 1993.
- [28] G. Rogers and N.S. Goel. Computer simulation of cellular movements: Cell-sorting, cellular migration through a mass of cells and contact inhibition. *Journal of Theoretical Biology*, 71:141–166, 1978.
- [29] A. G. Searle. *Comparative Genetics of Coat Colour in Mammals*. Logos Press Limited, 1968.
- [30] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B*, 237:37–72, 1952.
- [31] G. Turk. Generating random points in triangles. In Andrew Glassner, editor, *Graphics Gems I*, pages 24–28. Academic Press, 1990.
- [32] G. Turk. Generating textures on arbitrary surfaces using reaction-diffusion. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH 91 Conference Proceedings)*, volume 25, pages 289–298. Addison-Wesley, July 1991. ISBN 0-201-56291-X.
- [33] M. Walter. *Integration of Complex Shapes and Natural Patterns*. Ph.D. thesis, University of British Columbia, December 1998.
- [34] M. Walter and A. Fournier. Growing and animating polygonal models of animals. *Computer Graphics Forum (Eurographics'97)*, 16(3):C-151–C-158, September 1997.
- [35] M. Walter, A. Fournier, and M. Reimers. Clonal mosaic model for the synthesis of mammalian coat patterns. In *Proceedings of Graphics Interface 1998*, pages 82–91, June 1998.
- [36] M. Walter and C.G. Franco. Fast customization of geometric models. In *SIBGRAPI 2000*, pages 82–91. IEEE Press, Oct 2000.
- [37] A. Witkin and M. Kass. Reaction-diffusion textures. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH 91 Conference Proceedings)*, volume 25, pages 299–308. Addison-Wesley, July 1991. ISBN 0-201-56291-X.