# Procedural Modeling Facilities for Hierarchical Object Generation

Chantelle Morkel*
Department of Computer Science
Rhodes University
Grahamstown, South Africa

Shaun Bangay†
Department of Computer Science
Rhodes University
Grahamstown, South Africa

Figure 1: Application of procedural modeling tools

## Abstract

We modify a selection of interactive modeling tools for use in a procedural modeling environment. These tools are selection, extrusion, subdivision and curve shaping. We create human models to demonstrate that these tools are appropriate for use on hierarchical objects. Our tools support the main benefits of procedural modeling, which are: the use of parameterisation to control and vary a model, varying levels of detail, increased model complexity, base shape independence and database amplification. We demonstrate scripts which provide each of these benefits.

**CR Categories:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling - Curve, surface, solid and object representations

**Keywords:** procedural modeling, modeling tools, human modeling

## 1 Introduction

### 1.1 Problem Statement

Model generation is an essential aspect of creating computer animations, virtual environments and computer games. Such geometry can be created manually, which requires considerable human intervention. Procedural shaders are widely used to automate the generation of surface texture and detail, which would also otherwise require extensive manual effort. Procedural modeling promises similar benefits, such as producing a wide variety of similar models by varying parameters, producing models with any level of detail, and efficiently specifying complex models.

Existing tools used in 3D modeling applications rely heavily on constant user intervention. These tools cannot be directly reused in a procedural context, although it is desirable that existing model generation strategies be converted to the procedural paradigm. Equivalent tools are required that allow the reuse of these modeling techniques. These tools must be appropriate for use in a non-interactive environment, where the user input is limited to script-writing and parameter setting.

In this paper, we identify and adapt a minimal set of interactive 3D modeling tools to work in a procedural setting. These tools are able to work with models at arbitrary levels of detail. They work consistently regardless of the shape that is being manipulated, and allow flexible code reuse.

To demonstrate these tools we apply them to a common class of models, those composed of a hierarchical structure. This includes humans, most animals and plants, and many artificial structures. Specifically, in this paper, we are using the human figure as a test-case for the concepts defined as part of procedural modeling. Therefore, while the human model should be recognisable, it need not be flawless at this stage. The concepts described are also applicable to other hierarchical objects.

---

*e-mail: theriaca@gmail.com
†e-mail:s.bangay@ru.ac.za

AFRIGRAPH 2006, Cape Town, South Africa, 25–27 January 2006.

## 1.2 Requirements of Procedural Modeling

Procedural modeling is the process of creating 3D models through the use of parameterised procedures or scripts. Human interaction is limited to the initial script-writing process, with the tweaking of parameters forming the last vestige of interaction. There are several advantages associated with the use of procedural modeling techniques. All of which must be retained by any modeling tools adapted for procedural modeling.

**Parameterised Procedures:** A limitation of manual modeling is that the end result is a single model. If further similar objects are required, additional effort is necessary. Procedural modeling overcomes this pitfall by parameterising the procedures. Variation is introduced into a model by changing parameter values, and generating a new model. This offers an almost limitless number of unique variations for any procedurally generated model.

**Model Complexity:** The complexity of a model is limited to the capabilities of the human modeler. It may be infeasible to interactively create highly intricate models due to time, skill and storage constraints. Procedural modeling overcomes this by: [Marshall et al. 1980]

1. Evaluating procedures on demand, which allows control over the level of detail produced in the model.
2. Through the use of procedures, models possess an almost infinite level of detail. Refinement operators are used to create multi-resolution models [Velho et al. 2001].

**Base-Shape Independence:** An advantage of creating procedures is the ability to re-use them. A procedure can be applied in a variety of contexts. For example a procedure used to create hair could be used on a human head model, but equally well on an orangutan, dog or shaggy rug.

**Database Amplification:** Procedural modeling promotes database amplification; the creation of large amounts of information from a small input set. In a script-based scenario, database amplification is the generation of a complete model from a relatively small amount of code [Apodaca and Gritz 2000]. This is largely attributable to the use of looping structures commonly found within procedures.

The benefits of procedural modeling are dependent on the abilities of the modeling operators, or tools. A wide variety of interactive modeling tools exist in 3D modeling applications. From these, we must select an appropriate subset: those that can be adapted to use in a procedural modeling scenario. In this paper, we describe tools suitable for hierarchical object modeling.

We have selected hierarchical structures as a class of objects commonly used in graphical applications. Many animation systems rely on the use of a hierarchical skeleton for introducing dynamics. During the human modeling process, a human is constructed from a parent object with gradual refinement down to the extremities. For example the human torso could be considered the parent component for the arms, legs and head. The next level down, the thigh could be considered the parent for the knee and shin, which in turn is the parent for the foot.

## 1.3 Overview

The remainder of the paper is set out as follows. Section 2 discusses work related to procedural modeling, human modeling and modeling tools. Section 3 looks at the design of the procedural modeling tools. Section 4 discusses particular implementation details. Section 5 shows the application of the procedural modeling paradigm to a human model. Section 6 presents the results achieved using procedural modeling algorithms. Finally, Section 7 provides the conclusions drawn from the paper.

# 2 Related Work

We provide a survey on previous approaches to procedural modeling, considering degree of interactivity, operations supported and object representations. Existing interactive human modeling strategies are described and suggest a suitable strategy that can be adapted to procedural modeling.

## 2.1 Procedural Modeling

Procedural modeling techniques can be divided into three categories: those that create objects procedurally without user interaction, those explicitly based on scripting with a procedural modeling language, and those that require user interaction and provide visual feedback.

### 2.1.1 Non-interactive Procedural Modeling Techniques

Lindenmayer systems (L-systems) are parallel rewriting grammars, often used to model branching structures, in areas such as plant generation [Prusinkiewicz and Lindenmayer 1990]. The grammar used determines the nature of the model produced. The feather generation system of Chen et al. [2002] parameterizes the textures, curves and barbs of the feathers which are then generated through the use of an L-System.

Fractals describe geometry that is self-similar over different scales. Fractals can be used procedurally to control the level-of-detail. Fractional Brownian motion is commonly used for the generation of fractal terrains [Musgrave et al. 1989], and to generate displacement shaders in Renderman [Ebert et al. 2002].

Particle systems use procedural control over the dynamics of large numbers of particles to approximate objects that do not have distinct boundaries, and are fuzzy in nature [Reeves 1983]. Examples include fire, water and clouds. Particles can be used to model surfaces subject to dynamic constraints, such as cloth [Choi and Ko 2002; Oshita and Makinouchi 2001]. Oriented particle systems have been used in an object modeling application [Szeliski and Tonnesen 1992], to provide surfaces that are malleable, allowing manipulation using other objects as tools. The surface adapts to additional detail by adding extra particles as required.

Procedural modeling can be used to allow the combination or modification of existing geometric elements [Marshall et al. 1980]. Detail can be added using displacement maps, or procedures based on noise and turbulence functions [Perlin 1985]. Detail can also be added by defining the desired shape in terms of a function [Velho et al. 2001]. The detail is produced by taking the difference between these functions on different scale levels, which is then blended over all mesh resolutions. Deformations can be used procedurally to progressively mould models [Lewis and Jones 2004]. Hart [2002] instances geometry, contained within a procedure, using lazy evaluation to control level of detail.

### 2.1.2 Procedural Modeling Languages

Several authors [Cartwright et al. 2005; Cutler et al. 2002; Green and Sun 1988; May et al. 1996; Reeves et al. 1990] have taken non-interactive procedural modeling to another level by creating procedural modeling (and shading) languages. The procedural modeling languages are generally based on existing languages. This allows the procedural modeling language to inherit the functionality and constructs of the underlying language. Each of the procedural modeling languages presents the programmer with different tools and data structures.

The Renderman Shading Language is a procedural shading language that embodies several of the desirable aspects of a procedural modeling language. Some procedural modeling facilities, such as displacement shaders, are available. The shading language provides domain specific functions and operators which facilitate commonly used shading operations [Upstill 1993].

The operators provided by procedural modeling languages include union, intersection and subtraction [Cartwright et al. 2005; Cutler et al. 2002], the synthesis of shapes on existing geometry [Perlin and Hoffert 1989; Velho et al. 2001], weathering of geometry [Cutler et al. 2002; Musgrave et al. 1989], cutting operations [Szeliski and Tonnesen 1992] and extrusion [Lewis and Jones 2004; May et al. 1996; Parish and Muller 2001; Reeves et al. 1990].

Model representations include both surface and volumetric models. Surface models can be explicit sets of polygons, implicit surfaces [Cartwright et al. 2005], patches, quadric surfaces and spheres [May et al. 1996; Reeves et al. 1990]. Volumetric representations include tetrahedral meshes and signed distance fields [Cutler et al. 2002] and particles [Reeves 1983; Szeliski and Tonnesen 1992].

### 2.1.3 Interactive Modeling Techniques

Interactive procedural modeling techniques often have a modeling package interface, which allows a user to select the objects to be modified, and the changes to perform. Many 3D modeling applications provide a scripting or plug-in interface that allows some degree of procedural geometry generation [Autodesk, Inc 1999; Bayne et al. 2004; Brooks et al. 2001; NewTek 2001; Roosendaal and Selleri 2004]. In some cases the manual modeling process can be recorded as the sequence of modeling commands, and can be reused in a script [Lewis and Jones 2004]. Upon completion of the model, the user is able to modify any step of the process and have the resulting model regenerated.

An advantage of these systems is the ability to visualise a model and select changes to be made directly on the model. They provide a visual programming facility for procedural modeling.

Modeling packages also support interactive non-procedural modeling, which results in output that is difficult to re-use. Numerous modeling tools are found in various modeling packages [Autodesk, Inc 1999; Bayne et al. 2004; Brooks et al. 2001; NewTek 2001; Roosendaal and Selleri 2004].

Vertex positions can be manipulated using transformations such as rotation, translation and scaling. New faces, edges and vertices are created by using subdivision. This is useful for adding additional detail, and providing extra vertices for displacement and manipulation. A knife tool inserts additional faces with greater control over placement of the new vertices. Surface detail and smoothing can be improved using subdivision surfaces. Models can be deformed to simulate movement, or to reshape an object. Deformation tools include armatures, lattices and warps. Extrusion tools allow portions of the object to be "pulled out". The process is useful for extending and shaping a model, and adding additional faces and vertices. Various forms of extrusion are supported, such as repeated extrusions along a curve.

Various model representations are supported, such as polygon meshes, spline surfaces, subdivision surfaces, implicit surfaces and particle systems.

## 2.2 Human Modeling Approaches

A number of strategies exist for creating human models using interactive 3D modeling applications:

**Box Modeling:** The modeling process commences with a polygonal cube. Additional faces and vertices are created through subdivision (or use of a knife tool). Alternatively, extrusion is used to create additional faces and vertices. These faces and vertices are then displaced to match a human form [Ratner 2003]. Additional shaping is performed by applying a beveling tool.

**Patch Modeling:** This modeling technique uses spline surface patches (such as NURBS patches) to create human models [Ratner 2003]. The surface of the human is modeled as a series of patches which are shaped and then stitched together.

**Curve Modeling:** A background image which details the major curves of the human body is used as a template for shaping the model [Saastamoinen 1999]. The body is usually modeled from two points of view, a front view and a side view.

**Deformation Modeling:** The *MakeHuman* plug-in for Blender morphs previously modeled human characters to create new models [MakeHuman Team 2005]. The human models are based on accurate anatomical references and medical writings. The plug-in caters for the addition of parameters to create new characters. These parameters include control over body mass, gender and type of physique of the model.

We believe that the approach using box and curve modeling described by Saastamoinen [1999] is well suited to adaption to a non-interactive modeling environment. This modeling approach can be used on a wide range of hierarchical models, in addition to human models. The interactive tools used for this approach serves as the basis for the design and development of corresponding non-interactive modeling tools that are described in this paper.

The steps used to create a human model are:

1. Starting with an image template and a single cube, apply repeated extrusion until the image is completely covered with a coarse grid. This consists of the process:

```
while image not covered
    select interior faces
    extrude to increase coverage
```

2. Displace the control points on the object boundary to line up with the template.

3. Additional control points may be added to improve the fit of the model, or to smooth the geometry.

The operations used, which have to be adapted to a procedural modeling context, are selection, extrusion, curve shaping, and increasing detail through the addition of control points.

# 3 Procedural modeling tools

Our goal is to provide facilities (tools and object representations) for procedural modeling, that allow us to enjoy the benefits listed in section 1.2. Most interactive modeling packages support a wide range of tools and object representations, as described in Section 2.1.3. For the purposes of our prototype, we restrict ourselves to a minimal set capable of supporting the human modeling process described by Saastamoinen [1999] (Section 2.2).

The operations involved in this process include selection, extrusion and shaping to curves. Each of these operations must be capable of being used in a procedural context, specifically:

- Procedural modeling is effectively write once, use anywhere. The same models/procedures must be reusable, regardless of context. For example, a procedure to create a wooden plank can be used on different parts on the same model, but also used of different models with different parameters (such as length). The procedure needs to be independent of local geometry of the base shape, so it can be invoked regardless of the starting conditions.
- Little to no human interaction is required to create object models, once the procedure has been written. Steps such as selection cannot rely on human intervention.
- Procedural modeling offers the possibility of dynamically adapting the amount of detail in the model. The tools should provide this in an elegant fashion, without requiring extensive expertise on the part of the coder.

The tools should be applicable regardless of the underlying object representation. For our prototype we use a polygon mesh to represent the surface of the object. However many of the concepts described are independent of this representation.

One of the greatest advantages of using traditional, manual modeling approaches is the user interactivity afforded by graphics packages. Human judgement can be used to select relevant vertices when reshaping portions of an object. This becomes a disadvantage as the complexity of the model increases, and interactive access to the relevant portions of the model become more restricted, both because of visual and computational complexity.

We have selected a set of basic tools required to create human models, as described in section 2.2. The tools to be implemented are: selection, extrusion, curve shaping tool, and subdivision.

Non-interactive procedural modeling paradigms need to employ alternative methods of performing these basic functions.

## 3.1 Selection

Existing scripting languages allow selection based on numerical indices into lists of faces or vertices [Bayne et al. 2004; Brooks et al. 2001; NewTek 2001]. This is inappropriate when the selected region is identified by *geometric* properties, such as position in space. Numerical indices are also neither robust under changes in geometry, nor are they conducive to clarity in scripting. The modeler requires knowledge of how the model representation works, and which faces are located in each part of the model.

Representing the selected region as a set of indices shares similar pitfalls. Changes to model structure that remove or reorganize primitives invalidate these pointers.

We describe a selection through the use of two data structures:

- A set of labels associated with each primitive.
- A corresponding list containing all the primitives associated with a single label.

Both structures are updated during each operation that manipulates selections, which includes the other tools described in the following sections.

Selection in a procedural modeling context must be performed without human intervention. The facility allows the programmer to identify and specify the region of interest at the time of writing the script. The selection needs to be sufficiently robust to withstand variations in the underlying geometry that may result from the effect of changes in parameters used in previous modeling steps.

Existing interactive modeling approaches rely entirely on human intervention for selection.

Our strategy uses regions of space to define selections. While a variety of strategies could be used to define regions of space, we find that a strategy using bounding volumes works well for hierarchical objects.

Regions are demarcated through the use of bounding volumes. A selection is made by locating the faces that fall within a particular area defined relative to the dimensions of the bounding volume.

Such a selection works well if the geometry concerned fits the bounding volume. In the case of hierarchical objects, such as the human models described in section 5, each parent component is roughly cubical before any of the child limbs are extruded. A region selected relative to a bounding box then encompasses the relevant set of primitives in a manner unaffected by fine geometric detail, resulting in a robust selection mechanism.

Selections can be created in a hierarchical fashion, by allowing a selection within a selection. This representation supports the creation of hierarchical models, as components of a model can be individually referenced, and extended by another procedure.

Two categories of selection are supported. Anonymous selections have the scope of local variables, and are used for intermediate modeling operations. Named selections are associated with the model, are globally accessible, and are used to identify important geometric components. Named selections are preserved during modeling operations.

The hierarchical relationship amongst selections is explicitly maintained, and accessible by the modeling procedure. This hierarchical structure is intended to facilitate later operations on the model, such as the placement of a skeleton for animation of the model.

## 3.2 Extrusion

The extrusion tool is used to reshape a model. Selected regions are displaced and additional geometry is created to fill in the gaps.

Newly created geometry inherits the labels from the original selected region, and can optionally be tagged with its own unique label.

## 3.3 Curve Shaping

Saastamoinen [1999] shapes the outline of a human model by aligning the model boundary with an image template containing a projection of the desired model. A non-interactive version of the same process provides a useful tool for modeling humans and other objects.

The curves depicting boundaries can either be created explicitly (manually) based on an image template, or preferably in a procedural fashion, with control points being influenced by parameters passed to the procedure.

A corresponding set of primitives on the boundary of the object is selected, and individual control points are displaced to match the curve.

The quality of the resulting object can be improved by increasing the number of control points (for example through the use of subdivision) which ensures a more accurate fit to the curve. The curve shaping process, however, is unaffected. Thus curve shaping is independent of the underlying level of detail.

### 3.4 Subdivision

The $\sqrt{2}$ subdivision scheme of Li et al. [2004] is used to facilitate the generation of additional faces, and the smoothing of models. There are several benefits to using this subdivision scheme which include:

- The number of polygons at each step increases by a factor of two, unlike most other schemes which usually increase the polygon count by three or four. Providing finer grained control over level of detail.
- The approach works for meshes containing polygons with variable numbers of sides.
- The $\sqrt{2}$ subdivision scheme supports adaptive subdivision, allowing refinement to apply only to specific portions of the model.

Adaptive subdivision is used to control the level of detail within a model, as well as the refinement of larger polygons to get an even spread of regularly sized polygons over the entirety of the mesh.

Each subdivided face inherits the labels from the original face.

## 4 Implementation

This section describes the implementation aspects of each of these tools.

### 4.1 Model Representation

The model representation chosen is a polygonal mesh which is a popular format that affords inter-operability with other modeling applications. We use a variation of the winged-edge structure [Joy et al. 2002], with explicit representation of faces, vertices and edges. The connectivity information afforded by maintaining these lists allows the mesh structure to be traversed efficiently and with ease.

In addition selection information is stored as labels associated with each face. Vertex labels could also be used, but have not been required to date. Thus each face keeps track of the selections with which it is associated, just as the selections keep track of their respective faces.

Operations that manipulate the vertices, faces and edges of the mesh representation are responsible for updating these lists, and ensuring consistency within the structure.

Bounding volumes are generated from selections when required. We use axis-aligned bounding boxes which fit well with the cubical extrusion process used in human modeling.

### 4.2 Tools

**Selection**

The selection operation takes as input a minimum and maximum value for each dimension. Additionally, a parent selection and name for the current selection are parameters to the selection method.

The parent selection is used to determine which faces are considered to be candidates for the current selection. The bounding volume of the parent selection is used to determine the region of object space from which faces will be selected.

The coordinates passed to the tool represent ratios along the corresponding sides of the bounding box, and are values within the range $[0,1]$. A face is selected if all its vertices fall within the axis-aligned parallelepiped defined by these scale independent parameters.

Other geometric conditions for selection are also possible, but have not yet been needed.

**Extrusion**

The extrusion tool takes as input a selection consisting of a set of faces, and a vector specifying the magnitude and direction of the extrusion. A corresponding new vertex needs to be created for each vertex in the supplied set. The position of the new vertex is the position of the old vertex with the specified displacement applied to it.

A number of new faces are added to the model: one new face per boundary edge, and an additional face to cap the extrusion. This preserves mesh integrity and ensures that manifold meshes remain so, which is required by the $\sqrt{2}$ subdivision scheme.

**Curve Shaping Tool**

Bezier splines or B-Splines are used to create the curves to which the model is to be shaped. The curve shaping tool takes as input a selection, a curve, a projection vector and a direction in which to displace control points. The selected region of the object is displaced to fall onto the surface formed from the curve extended along the projection vector. We do this by projecting both control points and curve onto a 2D plane perpendicular to the projection vector.

The new position of each model control point falls at the intersection of the projected curve, and ray from the projected control point in the direction of the displacement vector. Since we allow any class of spline curve to be used, intersection is performed by approximating the curve with a pair of straight line segments, and testing for ray intersection on each of these. The closest curve section is then recursively subdivided, and the process repeated.

**Subdivision**

The $\sqrt{2}$ subdivision scheme inserts a center point into each face, which is then used to create new quadrilateral faces. Edges are used to form the new faces by connecting each of their end points to the face points of adjacent faces. The resulting mesh only contains quadrilaterals, although triangles can be introduced into the mesh on boundaries.

(a) (b) (c)

Figure 2: Original donkey (a), 1 level of subdivision (b) and 3 levels of subdivision (c)

Figure 2 illustrates the application of subdivision to a model of a donkey.

Adaptive subdivision is also applicable to selections on the model. The subdivision process can be viewed as adaptive subdivision applied to the entire model.

In addition to providing extra detail, subdivision can perform smoothing on a model. Smoothing is the reduction of jaggedness in the appearance of a model. Smoothing rounds off the edges of a model, which in some instances may be undesirable as a uniformly divided surface is required. A smoothness parameter is used to specify how much smoothing should occur during subdivision.

## 5 Human Modeling

Section 2.2 outlines the modeling strategy employed by Saastamoinen [1999] to create human models in modeling packages. Algorithm 1 lists the steps and modeling tool calls used to procedurally model the human form.

The basic human shape is created hierarchically, embodied as a series of calls to procedures which construct the major components of the body. While parameter values have been omitted from the algorithm presented, these can and are used to control proportions of the model.

This approach mirrors the strategy described by Saastamoinen [1999] in that it first constructs the basic structure of the model before applying detail in the form of the curve shaping operations.

The parent node in the hierarchy, the torso, is created independently of the children (arms, legs and head). Thus changes to the geometry of the parent can be made without affecting the procedures for the children.

The advantage of the procedural modeling approach can be seen by the lack of a mirroring step, as required in the original strategy. Building both sides symmetrically requires little extra coding effort, and so complications such as stitching the two halves together after mirroring can be avoided.

The hierarchical nature of the modeling process is represented in the model. This information is available as part of the model, and can be made available to processes that may require this information, such as matching the model to a skeleton for animation purposes.

Algorithm 2 provides one possible torso modeling strategy. It creates a rectangular parallelepiped consisting of many small adjacent cubes occupying a volume with the given dimensions. Extrusion is used to provide the vertices required for later shaping. The size of the basic cube determines the level-of-detail of the model. If the cube size decreases many more are required to fill the same volume, and more vertices are available during later shaping steps. Other torso modeling strategies could also be used without affecting the working of the rest of the model (section 6.4).

**Algorithm 1** A Human Modeling Algorithm

```
HumanModel ()
  createTorso (0.288, 0.398, 0.611, 6.0)
  headSel←torso.select(17.5%,82.5%,99%,100%,0%,100%)
  head←createHead (headSel)

  leftArmSel←torso.select(99%,100%,0%,100%,0%,100%)
  rightArmSel←torso.select(0%,1%,0%,100%,0%,100%)
  leftArm←createArm (left, leftArmSel)
  rightArm←createArm (right, rightArmSel)

  leftLegSel←torso.select(50%,100%,0%,1%,0%,100%)
  rightLegSel←torso.select(0%,50%,0%,1%,0%,100%)
  leftLeg←createLeg (left, leftLegSel)
  rightLeg←createLeg (right, rightLegSel)

  torsoCurves (torso)
  headCurves (head)
  armCurves (left, leftArm)
  armCurves (right, rightArm)
  legCurves (left, leftLeg)
  legCurves (right, rightLeg)
```

**Algorithm 2** A Torso Creation Algorithm

```
createTorso (Depth, Width, Height, unit)
  Create a cube of unit size

  repeat Depth/unit times
    select front faces of the torso cube
    extrude faces along the forward axis

  repeat Width/unit times
    select right faces of the torso cube
    extrude faces along the right axis

  repeat Height/unit times
    select bottom faces of the torso cube
    extrude faces along the downward axis
```

Curve shaping is used to refine the shape of the basic body components. The process for the torso is outlined in Algorithm 3. The curves are created dynamically, and are proportioned to match the geometry provided. Control points are displaced to match the desired outline. The accuracy with which the model fits the curve depends entirely on the level of detail with which it is initially created. Subdivision could also be used between geometry creation and curve shaping to increase the level of detail of the final model.

To make the foot modeling procedure robust, we have devised an algorithm that creates additional detail as it is required. Algorithm 4 shows the algorithm used to select and create a foot and toes.

The number of toes for each foot is specified as a parameter, as is the threshold level. The threshold value controls how detailed a selection is. For example if the threshold value is set to 10, a selection is valid if it has at least 10 faces. If any selection contains less than the specified number of faces, the region from which the toes are

**Algorithm 3** Shaping the Torso Model to Curves

```
torsoCurves (torsoSelection)
  bv←torsoSelection.BoundingVolume ()
  minMax←bv.minX + bv.maxX

  //Assign Control Points
  CP1←(bv.maxX, bv.maxY, 0.0)
  CP2←(bv.maxX-(0.15*minMax), bv.maxY/2, 0.0)
  CP3←(bv.maxX-(0.1*minMax), bv.minY, 0.0)

  rightCurve←BezierSpline (CP1, CP2, CP3)

  //Re-assign Control Points for the front,
  //back and left curves
  leftCurve←BezierSpline (CP1, CP2, CP3)
  frontCurve←BezierSpline (CP1, CP2, CP3)
  backCurve←BezierSpline (CP1, CP2, CP3)

  shapeToCurve (torsoSelection, rightSideCurve)
  shapeToCurve (torsoSelection, leftSideCurve)
  shapeToCurve (torsoSelection, frontSideCurve)
  shapeToCurve (torsoSelection, backSideCurve)
```

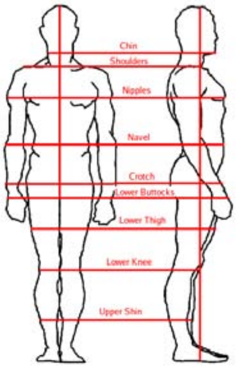

Figure 3: The human figure (by Loomis)

selected is adaptively subdivided. This process is repeated until all of the toes are valid (in terms of the threshold value) selections.

# 6 Results

Several experiments have been conducted to test the versatility of the procedural modeling tools which have been implemented. The primary experiment is the construction of a human model in a non-interactive context. The secondary experiments are designed to illustrate the use of parameterisation to introduce variation into a model, the re-usability of procedures in different models and the application of variable levels of detail in a model.

The models have been rendered in Blender. The textures have been added manually to the models.

## 6.1 Experiment 1 - The Human Model

The proportions for the human model are taken from sketches by Loomis [1943]. These define a set of curves that are used to shape the human body.

**Algorithm 4** A Foot Creation Algorithm

```
FootModel (numToes, threshold, initialSelection)
  toeSize←1.0/numToes
  allSelectionsValid←false
  while NOT allSelectionsValid
    anonSelect←initialSelection.select
                (0%,100%,0%,40%,75%,100%)
    xMin←0.0
    xMax←toeSize
    valid←true
    for i from 1 to numToes
      toe$_i$←anonSelect.select
            (xMin,xMax,0%,100%,0%,100%)
      if toe$_i$.numFaces() is less than threshold
        valid←false
      xMin←xMax
      xMax←xMax + toeSize
    if NOT valid
      anonSelect.adaptivelySubdivide ()
    allSelectionsValid←valid

  for i from 1 to numToes
    extrude toe$_i$
```

Figure 3 shows the front and side views of the sketches used to determine the curves of the human model. Each component of the model, ie the torso, arms, legs, neck and head, are dealt with individually. Figure 4 illustrates the stages of the human modeling process. Initially, a cubical torso is created, after which arms, legs, neck and head are extruded. The shaping is the final step in the modeling process.

## 6.2 Experiment 2 - The use of Parameterisation

Procedural modeling is reliant on the use of parameters as the conduit for user interaction. The use of parameters allows the introduction of variation between models. To illustrate this concept, we have taken the same human model created in experiment 1, and through the use of parameters changed the appearance of the model. Parameters controlling the curves around the abdomen and buttocks are adjusted to get different effects, the results of which can be seen in Figure 5.

To further illustrate the flexibility of parameterisation, we have created a table and chair model. The models themselves were trivial to implement, with most of the features, such as leg length, table length, and chair back height, being parameterised. Figure 1 shows an arrangement of chairs, tables and a luggage model. Each of the chairs and tables was created with these models, and variation is introduced by changing parameter values.

## 6.3 Experiment 3 - Base Shape Independence

The procedural components are not dependent on the initial starting shape which makes them base shape independent.

The human model in section 6.1 uses a cubical torso model as the basis for the human. To illustrate base shape independence, we apply the same procedures used to create limbs on the initial human model to an egg-shaped torso. The egg-shaped torso is a parallelepiped that has been subdivided to create a curved surface. Figure 6 shows that the same steps are taken to model the egg-shaped

Figure 4: Human model - front and side views

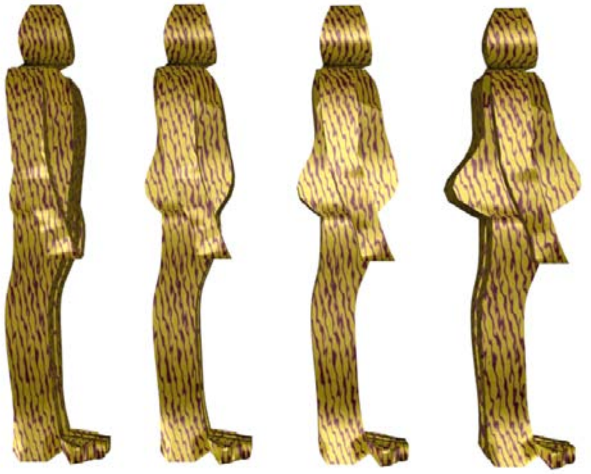

Figure 5: Human model with different parameter values

human as the ordinary human. The two exceptions are that no head is added to the egg-model, as the character should resemble "Humpty-Dumpty", and no torso shaping has been performed in order to preserve the egg shape.

### 6.4 Experiment 4 - Re-usability

A benefit of procedural modeling is the re-use of models and model components. To test the viability of this attribute we re-use components of previous models to create a model similar to Terry Pratchett's character "The Luggage" [Pratchett 1985].

To create the luggage, we created a hollow trunk model through the repeated use of selection and extrusion. The lid is modeled using the curve shaping tool. The legs of the luggage model are grown from selections at the base of the trunk. The leg model is the same one used to model the legs of both the human model and the egg-shaped human model. Figures 1 and 7 show the resulting luggage. The number of legs can be varied.

Figure 7: The Luggage model

### 6.5 Experiment 5 - Variable levels of detail

Our tools are robust enough to handle different levels of detail. The more detail available, the finer the resulting model. This is especially true when applying curves to a model. The more detail there is to work with, the better the fit to the curve. Figure 8 illustrates this capability by applying curve shaping to torso models containing varying numbers of vertices.

### 6.6 Summary

Experiments 1 through 5 look at the various procedural modeling benefits afforded through the use of our procedural modeling tools. The experiments are designed to test the claims of variation through parameterisation, re-usability and variable levels of detail.

The initial human model is the prototype to test the capability of our modeling tools to support the creation of a hierarchical model. This experiment also illustrates that the manual modeling approach created by Saastamoinen [1999] can be ported to a non-interactive modeling environment.

The tools operate independently of the complexity of the model.

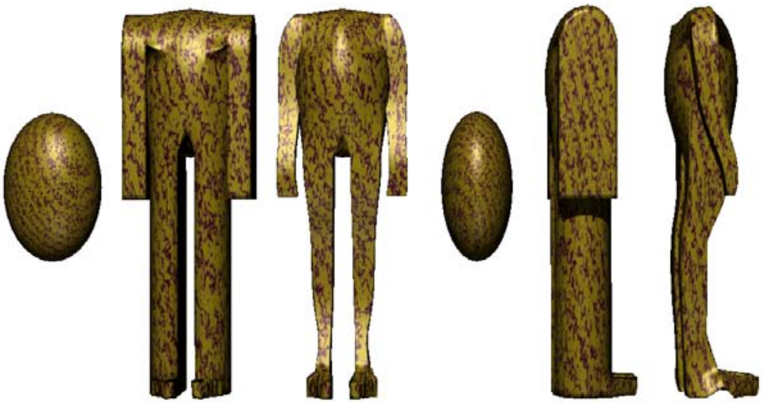

Figure 6: Human model with egg-shaped torso - front and side views

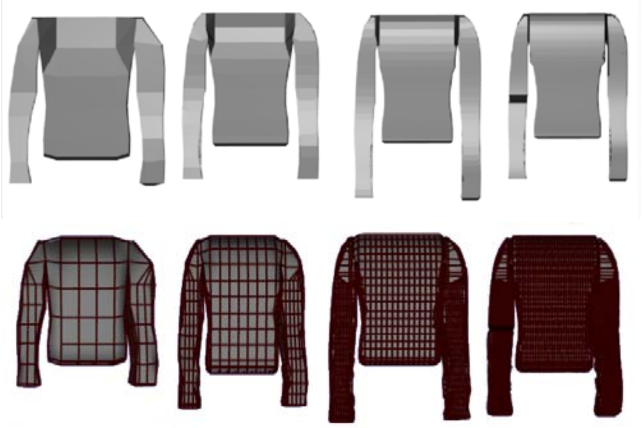

Figure 8: Curve shaping applied to varying levels of detail

## 7 Conclusion

We have created a minimal set of tools that are suitable for non-interactive procedural modeling. We demonstrate the applicability of these tools by adapting an interactive modeling process for human figures so that it can be performed entirely using only the tools that we have created.

Hierarchical models, specifically human models, can be generated in a non-interactive modeling environment. Moreover, the tools that have been implemented are well-suited to the support of hierarchical model creation.

We show that the modifications to these tools allow us to enjoy the benefits of procedural modeling, specifically:

1. Variation is introduced into a model by changing parameter values. We have created numerous different human models using a single procedure.
2. Detail can be added to a model when required by subdividing, which provides extra vertices. Adaptive subdivision allows this detail to be confined to regions where it is required.
3. The tools are independent of the base geometry provided, and prove resilient under changes to the amount of detail, and shape of the starting object. Procedures have been shown to be reusable for different models.
4. Individual procedures can be reinvoked many times to create models of the required complexity.

### 7.1 Contributions

We have created a strategy for selecting vertices that requires no direct human interaction. Our extrusion, subdivision and curve fitting tools all use and preserve these selections.

We have defined a set of tools that allow modeling of hierarchical objects procedurally. Additionally, these tools are capable of working at arbitrary levels of detail, and are base-shape independent.

### 7.2 Future Work

Work is in progress to integrate these tools into a complete procedural modeling environment. Further developments include extending these tools to other categories of model, and providing a procedural modeling language to simply and efficiently specify modeling operations.

## References

APODACA, A. A., AND GRITZ, L. 2000. *Advanced Renderman: Creating CGI for Motion Pictures*. Academic Press, San Diego, California.

AUTODESK, INC. 1999. *3D Studio Max Release 3, Reference Volume II.* Autodesk, Inc.

BAYNE, J., FULLER, G., GOULET, E., KRUGER, E., LANGEVIN, L., RAHAL, J., AND VINCELLI, G., 2004. Softimage|XSI version 4.0 tutorials 1.

BROOKS, S., FURGUSON, S.-B., FORD, L., MACRI, C., PARK, S., RAMEY, D., AND ROSE, L. 2001. *Mel Version 4*. Alias | Wavefront.

CARTWRIGHT, R., ADZHIEV, V., PASKO, A. A., GOTO, Y., AND KUNII, T. L. 2005. Web-based shape modeling with hyperfun. *IEEE Computer Graphics and Applications 25*, 60–69.

CHEN, Y., XU, Y., GUO, B., AND SHUM, H.-Y. 2002. Modeling and rendering of realistic feathers. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM Press, 630–636.

CHOI, K.-J., AND KO, H.-S. 2002. Stable but responsive cloth. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM Press, San Antonio, Texas, 604–611.

CUTLER, B., DORSEY, J., MCMILLAN, L., MULLER, M., AND JAGNOW, R. 2002. A procedural approach to authoring solid models. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM Press, 302–311.

EBERT, D. S., MUSGRAVE, F. K., PEACHY, D., PERLIN, K., AND WORLEY, S. 2002. *Texturing and Modeling*, third ed. Morgan Kaufmann.

GREEN, M., AND SUN, H. 1988. A language and system for procedural modeling and motion. *IEEE Computer Graphics and Applications 8*, 6 (November), 52–64.

JOY, K. I., LEGAKIS, J., AND MACCRACKEN, R. 2002. Hierarchical approximation and geometric methods for scientific visualization. In *Data Structures for Multiresolution Representation of Unstructured Meshes*, G. Farin, H. Hagen, and B. Hamann, Eds. Springer-Verlag, Heidelberg, Germany.

LEWIS, T., AND JONES, M. 2004. A system for the non-linear modelling of deformable procedural shapes. *Journal of WSCG (Winter School of Computer Graphics) 12*, 1-3 (February).

LI, G., MA, W., AND BAO, H. 2004. $\sqrt{2}$ subdivision for quadrilateral meshes. *The Visual Computer 20*, 2 (May), 180–198.

LOOMIS, A. 1943. *Figure Drawing, For All It's Worth*. Viking Press, New York.

MAKEHUMAN TEAM, 2005. Makehuman project. [Last Accessed: 29 September 2005].

MARSHALL, R., WILSON, R., AND CARLSON, W. 1980. Procedure models for generating three-dimensional terrain. In *SIGGRAPH 1980: Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 154–162.

MAY, S. F., CARLSON, W. E., PHILLIPS, F., AND SCHEEPERS, F. 1996. Al: A language for procedural modeling and animation. Tech. rep., Ohio State University and CSIR.

MUSGRAVE, F. K., KOLB, C. E., AND MACE, R. S. 1989. The synthesis and rendering of eroded fractal terrains. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, ACM Press, 41–50.

NEWTEK. 2001. *Lightwave 3D 7 Reference Guide*. NewTek, San Antonio, Texas.

OSHITA, M., AND MAKINOUCHI, A. 2001. Real-time cloth simulation with sparse particles and curved faces. In *Proceedings of Computer Animation 2001*, 220–227.

PARISH, Y. I. H., AND MULLER, P. 2001. Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM Press, 301–308.

PERLIN, K., AND HOFFERT, E. M. 1989. Hypertexture. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, ACM Press, 253–262.

PERLIN, K. 1985. An image synthesizer. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, ACM Press, 287–296.

PRATCHETT, T. 1985. *The Colour of Magic*. Corgi Adult.

PRUSINKIEWICZ, P., AND LINDENMAYER, A. 1990. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York.

RATNER, P. 2003. *3-D Human Modeling and Animation*, second ed. Wiley.

REEVES, W. T., OSTBY, E. F., AND LEFFLER, S. J. 1990. The Menv modelling and animation environment. *Journal of Visualization and Computer Animation 1*, 1 (August), 33–40.

REEVES, W. T. 1983. Particle systems - a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics 2*, 2, 91–108.

ROOSENDAAL, T., AND SELLERI, S., Eds. 2004. *The Official Blender 2.3 Guide : free 3D creation suite for modeling, animation and rendering*. No Starch Press, USA.

SAASTAMOINEN, O.-P., 1999. Modeling a human body. [Last Accessed: 25 July 2005].

SZELISKI, R., AND TONNESEN, D. 1992. Surface modeling with oriented particle systems. In *SIGGRAPH 1992: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 185–194.

UPSTILL, S. 1993. *The Renderman Companion: A Programmer's Guide to Realistic Computer Graphics*. Addison-Wesley, Boston.

VELHO, L., PERLIN, K., YING, L., AND BIERMANN, H. 2001. Procedural shape synthesis on subdivision surfaces. In *Proceedings of the 14th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*.