Procedural Terrain: A Virtual Bulldozer

Stephen Ervin

Department of Landscape Architecture
Graduate School of Design
Harvard University
Cambridge, MA
USA

Caroline Westort

Spatial Data Handling Division
Geographisches Institut
Universität Zurich
Zurich
SWITZERLAND

We describe a system for procedural landform design, which uses the simple metaphor of a two- dimensional profile or 'blade', swept along a three-dimensional trajectory, or 'path', leaving behind a modeled surface in its wake. This 'path' and 'blade' system can mimic a common bulldozer - a simple rectangular blade moved along a path constrained to straight lines and large radius curves - as well as more fanciful landform designs- a blade of continually changing profile swept along an exponential spiral path, for instance. Our prototype 'bulldozer' (implemented in AutoLisp) operates in a field of procedurally defined landform 'primitives' to create a virtual surface, and uses a standard rectangular mesh for displaying the resultant landform.

Keywords: digital terrain model, procedural model, landscape design, landform

1 Introduction and background

Digital Terrain Models (DTM's) are an important class of surface models [1] -Ease the principal medium for much landscape architectural design, and the essential base upon which architects' models of buildings and cities (should!) sit. Modelling topography is a fundamental element in the palette available to landscape architects for environmental expression. Topography, vegetation and dynamic systems (such as water) together comprise the basic landscape design vocabulary [2] . Landforms may be used as design elements unto themselves, and also to organize and establish the base upon which other elements may be composed. Landscape architects think of topography in both functional quantitative and spatial qualitative terms, and through use of slope, elevation change, convex and concave regrading, both subtle and dramatic meaning can be invested into a landscape design project. Landscape designers use a variety of abstractions which help to organize or synthesize their design of landform, including 'signatures' of contour lines in plan; compositions of regular geometric solids and flat planes with break lines and transitions; and processes of land formation: erosion, deposition, scraping and piling, e.9The topographic condition of a site is therefore regarded by the profession as an essentially plastic one, potentially to be sculpted by a designer. Through 3-D modelling, rendering and visualization, designers are able to try out an assortment of design approaches on a topographic surface. Through manipulation of 3-D models or images design alternatives are seen, changed, and analyzed. For environmental designers, visualization works in close concert with manipulation and quantitative and qualitative analysis.

The process of landscape design has been characterized [4] as consisting of three phases: Phase One is the inventory and analysis phase where inventories of existing physical and cultural conditions of the site and its surroundings are recorded. Conflicts, constraints, and opportunities which will guide the design development are identified. Phase Two is design development which involves synthesis of observations and evaluations of Phase One into a design concept which generates one or more design solutions. Phase Three is design implementation, involving the translation of design intentions into construction drawings and specifications.

These tasks have been typically carried out on paper. Contour lines, or a grid of points, together with sectional cuts and the occasional perspective view are the chief means used to abstract topographic form onto a 2-dimensional surface. These representations are then evaluated and analyzed, changes are proposed, and the process repeated. This cycle conducted on paper results in a lag time between different generations of topographic changes and their representation.

The same rag time holds true for use of other analog models (clay, cardboard) during the above process. Employed in architecture and landscape design studios at every stage, during the inventory phase they are a means to record the before design' site circumstances. During the design development phase they are used at various degrees of detail, from very rough sketch models, to very detailed accurate ones to test out and present specific design scenarios. Choice of the modelling media varies according to the model's intended purpose and ease of generation and manipulation criteria.

Use of analog models in the landscape process is both an essential, and resourceintensive endeavor. A simple chipboard model often requires several hours to cut and assemble. Making changes to models in non-digital media often requires the designer to return to a contour representation on paper and through another cutting and assembly routine. Consequently, valuable time and energy are expended upon the model medium, and its opportunities and limitations, often at the expense of the landform design.

The objective of this research project is to place landscape designers closer to the target concern - spatial characteristics of landform - rather than the characteristics of the model medium. We seek digital tools for landscape architectural design practice which situate the topographic designer 'behind the (virtual) bulldozer', as conductor rather than as a constructor of model or drawing. This emphasis lets the designer do what she does best: spatially design topography. This requires both a modelling medium and sculpting tools which meet the following criteria:

(a) geometric control
(b) ease of handling
(c) quick response time
(d) quantitative accuracy

Such a toolset would 'tighten' the iterative design loop for landscape designers, i.e., bring the editing and analysis phases closer to the idea and representation cycle, thereby enhancing the design cycle with exploration and evaluation of alternatives. Modern programmable, user-extendible CAD systems offer the platform from which to build such tools.

1.1 Digital Landform Representations

Digital 3-D model manipulation strategies for topography have evolved chiefly in the CAD and GIS digital domains [5]; the former focusing primarily on the parameterization of objects, the latter on combinations of spatial attribute information with digital terrain models in several representations (TINS, DTMS, contours, e.g. see [6]). The simplest and most common of these is the regular tessellation, or mesh, displayed as a raster array. Large raster arrays are reasonable to use for representation of arbitrarily shaped natural features such as the earth's surface, but the manipulation tools available make them neither an efficient nor an expressive medium for design.

Geographic Information Systems have evolved with the ability to handle the large datasets characteristic of landscape, but have heretofore focused on the display and analysis of elevation data, rather than on active tools for manipulating landform. Manipulation strategies in many GIS's allow only either 'global or 'local' changes to a model: a local change is a change that happens to a single vertex, a global change is one that affects the entire topographic dataset, e.g., scale changes, which exaggerate the Z value. While some operations exist to limit the scope of activity to a mask of pixels or a specific polygon, these techniques are not especially useful for landform design. What is general missing is a specification of landform primitives - mound, swale, plane, e.g. -, which carry their own parametric definitions and constraints, and so enable regional changes, between local and global in their scope, in which slopes, radii and other dimensions may be specified and propagated.

Landform design using 2-D CAD systems relies heavily upon the representation of 3-D form as contour lines; an abstraction well suited to representation, but notoriously cumbersome for design. Manipulating contour lines effectively with CAD systems remains a daunting cha1lene, requiring spline curves and geometric constraints that have nowhere yet been satisfactorily packaged for - much less mastered by - landscape designers.

An obvious advantage of digital/virtual methods is the ability to directly manipulate a 3-dimensional representation. The following list, adapted from [7] summarizes some of the 3-dimensional topographic modelling tools available in modern CAD software packages:

Data structures:

(a) B-Rep - Boundary Representations - The surface of an object is 'polygonized' and a description stored as a list of vertices, lines joining the vertices, and list of faces.

(b) Primitives - Library of simple, generic, 3-dimensional models (cube, sphere, cylinder, cone, torus, wedge, plane and others). These primitives can be scaled, translated and rotated within the application, often both interactively (such as with a mouse), and by numerical input.

(c) CSG - Constructive Solid Geometry - An object is represented as a combination of simple primitives such as cube, sphere and cylinder. These basic solids are used as building blocks for more complex objects by means of a system which uses Boolean combinations (union, intersection, and difference) to describe the logical operations of adding two objects, subtracting one from another, or defining the overlap between two objects.

(d) Voxels - Volume modelling. Spatial occupancy enumeration divides 3dimensional space into cubic units called voxels, or 3-dimensional pixels. Operators:

(e) Swept Forms - a 2-dimensional (XY) section 'swept' along a third (Z) dimension.

(f) Extrusion - a template is swept in a direction orthogonal to the plane in which it lies. g. Surface of Revolution - a 2-dimensional template, closed or open, rotated about an axis.

(h) Skin - the ability to construct a 'skeleton' of a form and then wrap a surface skin around it to create an object. i. Patches - same as skin, except using boundaries as the skeleton.

Geometric modelling with primitives such as cones and cylinders is promising [8], but by itself is too limited for most landform design, which more often than not involves the design of a surface rather than of a solid. Directly manipulating 3-D curved patches remains too computationally intensive to be justified (or affordable) for landscape design. This paper presents a means to generate a '2.5 dimensional' [9] model of topography by adapting extrusion, sweep, Boolean operations and library primitive modelling techniques from CAD to arrive at a procedural toolset. Our present system is confined to the creation and editing of surfaces, or continuous fields (no tunnels or overhangs). This toolset operates on a simple (rectangular grid) dataset, and provides a variety of "scopes of action":

(a) local = a change to one vertex

(b) move = an incremental geometric change to the surface (more than one vertex)

(c) pass = a collection of moves (more than 1 move)

(d) global = a global change to the dataset (all vertices) (see Figure 1)

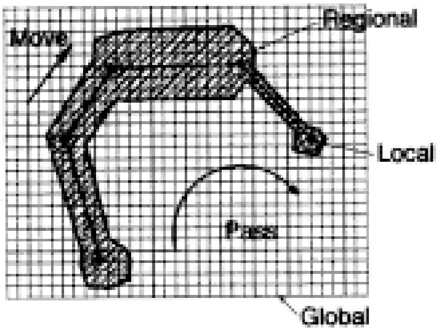

Figure 1. Conceptual Components of Landform Tool

Of these, the middle two (regional operators) are the most interesting and challenging.

2 The Virtual Bulldozer

The prototype landform design system presented here borrows the simple raster array data structure from GIS, and adapts the CAD modelling strategies skin, sweep, and Boolean combinations of Constructive Solid Geometry modelling - A set of integrated surface forming tools, surface primitives, and combinatorial operators enable the landform designer to incrementally manipulate local areas on a continuous digital topographic surface, in a sequence of 'moves' and 'passes'.

This 'virtual bulldozer' consists of three main parts:

(1) An "extrude" tool, made of a "path" and a 'blade"
(2) A library of parameterized primitives
(3) Boolean and algebraic combinations between primitives and extrusions.

2.1 Ext nide tool, or blade and path

A real bulldozer moves over terrain with a blade, or shovel, which displaces earth as it moves along the length of a specified path. In the virtual bulldozer, a 'blade' is defined as a user-definable number of points that define a 2-dimensional profile. A 'path') is defined as a user-definable number of 3D points, a 3D polyline. The blade is swept, or 'pushed', along the path such that it is centered upon and normal to the axis of the path although its center and angle with respect to the path can be specified). The result of moving the blade along each segment of the path, from one point to the next point, represents a 'move', while the collection of moves along the path represents a 'pass'. See Figure 2.

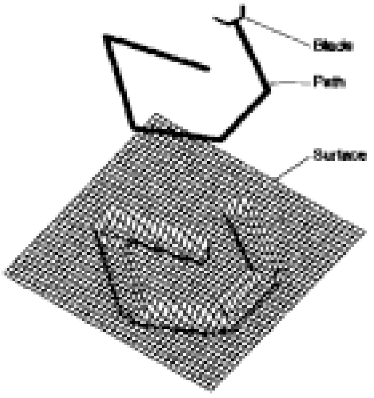

Figure 2 Illustrative Diagram of blade, Path, and Surface

A real bulldozer always cuts relative to the existing surface, as its treads are confined to traveling over the surface. For our virtual bulldozer, the points of a path may have no z-coordinate specified, in which case the tool will cut relative to the existing terrain, taking its base Z value from the terrain data. If a Z-value is specified in the path, the blade cuts along the absolute path given. A single path may have both specified and unspecified Z-values, allowing a mixed path. By making the Z-coordinate optional along the path, the virtual bulldozer has the ability to either accommodate or ignore existing topographic conditions of a site; to both mimic and go beyond the abilities of a real bulldozer.

The goal of this blade and path tool is to maximize geometric control over a digital topographic surface. In order for a digital terrain model to be useful to a digital landscape designer, a large universe of geometric forms must be describable in terms meaningful to the designer, and reflective of real earthworks considerations: "a long straight ramp at 7%, widening at the top", or "a spiral curve along the hillside, following the 20 meter contour, for example.

2.2 *Library of primitives*

A variety of landform designs can be made up of the intersection or union of simple geometric primitives - truncated cones, half cylinders, etc. A number of these primitives can be constructed using the extrude tool (half cylinder, e.g,) but for compactness of representation it is often preferable to enumerate a library of topographic primitives, such as mound, berm, swale, torus, box, etc., each with their geometric properties parameterized. For example, in our system the 'berm' primitive is defined with the following geometric parameters:

l = length
h1 = beginning height
r1 = beginning radius
h2 = end height
r2 = end radius
c = degree of curvature
c2 = degree of curvature above ground plane (see Figure 3)

Primitives are encoded as procedure calls which return a Z value given an X and Y - z = f (x y) - and so the basic library can be easily extended by the AutoLisp programmer-

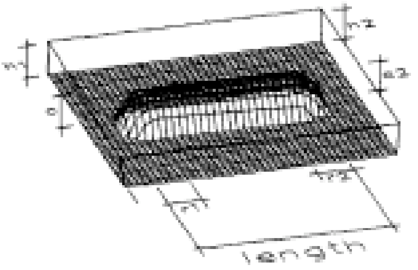

Figure 3. Diagram of Parameterized Berm Primitive

2.3 Combination operations

Primitives and extrusions can be combined into a single landform, using algebraic operators such as +, -, max, mm, average, etc. For example, in Figure 4 a mound and a box created by extrusion are added; in Figure 5 the box is subtracted from the mound (and the display is shown at a finer resolution.)

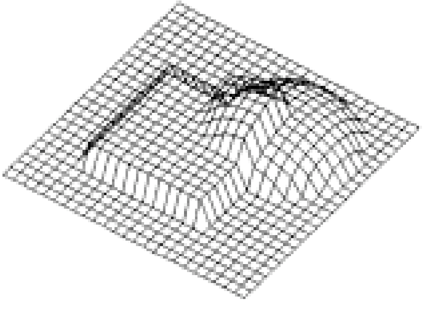

Figure 4. Mound and Box primitive ADDED

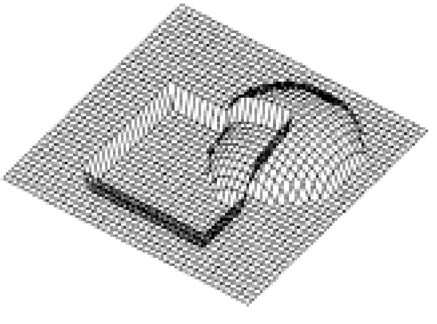

Figure 5. Mound and Box primitive SUBTRACTED.

2.4 Procedural Terrain Modelling

The virtual bulldozer code (in AutoLisp in its first manifestation) integrates the three above parts - lade and path, primitives, and combinatorial operators - in a single procedure. The principal logic of this procedural terrain generation is as follows:

Required Data: A terrain database: grid of Z values with the following scalar parameters:

N Columns

```
N Rows
X cellsize
Y cellsize
X Offset
Y Offset
A blade: a 2D polyline
A path: a 3D polyline: z= nil at any vertex signifies 'relative' cutting A List of
Primitives: a list of reserved words (box, mound, etc.) with required parameters A List of
combination Operators: one for each primitive in list
A window of operation: a rectangle from minx, miny to maxx, maxy
A resolution: scalar
Main Loop:
for each X from minx to maxx by step resolution
    for each Y from miny to maxy by step resolution
        ;; get base value
        Z0 := existing terrain @ [X Y]
        ;; compute primitives
        newZ = Z0
        for each Primitive in PrimList
        if (point XY) is in bounds of Primitive then
            compute newZ by function call
                e.g.: NewZ = mound (r, h, c, x, y, z)
            ;; get combination operator
            case    '+'
            newZ = newZ + Z0
            case    '-'
            newZ = newZ - Z0
            case 'max', 'min', 'average', etc. ...
        end ;; each primitive
        ;; call bulldozer
        for each line segment in path
        Rect0 = Rectangle defined by blade width extruded
            perpendicular to line segment
            if (point XY) within Rect0 then
            startZ= Z coord of starting point,
            endZ = Z coord of end point
            if startZ is nil then
            StartZ = lookup Z at startpt in base terrain
            if endZ is nil then
            endZ = lookup Z at endpt in base terrain
            compute distance along path line segment
            interpolate bladeZ between startZ & endZ
            compute blade distance perpendicular to path
            BullZ = interpolated blade Z value at XY
            (Repeat combination operator as above)
            FinalZ = newZ combined with bladeZ
            NewGrid[X Y] = FinalZ
    end ;; each line segment, each X, each Y
```

Note that in this implementation we favor simplicity, clarity and ease of implementation over efficiency, at the expense of speed of execution. We are now porting the code to C++, and making improvements in the efficiency of the main loop.)

The prototype implementation also provides a suite of graphic display, data storage and retrieval, and other useful functions. A particularly useful function is the ability to specify on the fly the desired resolution (grid cellsize) of the calculation and display. When relatively large or coarse, a sketch mode is implemented; for final displays or closer examination, further detail can be presented. These variable resolutions may be nested and combined in a single display.

The virtual bulldozer, through a set of basic but flexible and extendible operations (the blade and path metaphor), enables the landscape designer in the digital medium to both mimic and go beyond traditional analog landform design processes. For example, Figure 6. shows a fanciful design for a ziggurat'; a spiral path subtracted from the surface of a regular hemispherical mound. Figure 7 shows a replication of a well-known earthworks park [10] created in a digital environment using primitives and operators, and rendered with surface lighting and textures.

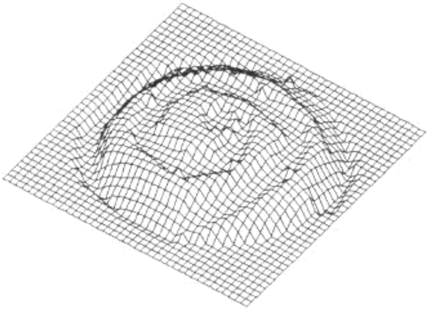

Figure 6. A complex three-dimensional form created by a spiral path on a mound primitive .

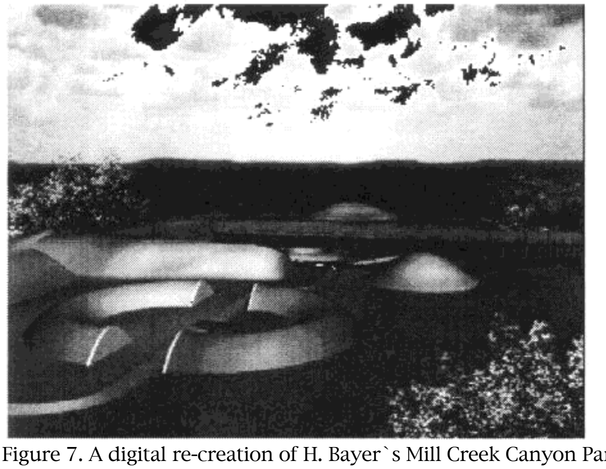

Figure 7. A digital re-creation of H. Bayer`s Mill Creek Canyon Park.

3 Conclusion

The goal of this project has been to more tightly couple the idea generation and visualization loop for landform design, while providing geometric control, ease of handling, quick response time, and quantitative accuracy. Building the virtual bulldozer within a well-established and robust CAD system has been a useful strategy; ease of handling, quick response time and quantitative accuracy arise from the underlying hardware and software platform. In addition, we have been spared from inventing display and rendering tools, and have been able to concentrate upon the provision of geometric control via the blade and path metaphor. We believe the model holds promise for virtual landform design, as well as for real-world earthworks operations using robot-controlled bulldozers.'

The prototype virtual bulldozer at present 'knows only about the instructions given by the designer in the form of blade(s), path(s), and primitives; but surely could 'know' more about geophysical constraints and respond to other site conditions (hydrology, vegetation) where appropriate. Modelling processes of land formation - whether natural or synthetic - is an appealing approach as it involves a dynamic rather than static conception of the medium, and potentially may account for physical and behavioral properties of the material being designed [11].

This approach to choreography of construction process is efficient and expressive as a medium for design - offering both computational advantages and drawbacks, - as compared to more static approaches representing landform as product, or object (data structure). Especially well suited to the inherently continuous, sweeping nature of landform, procedural dynamics are in general an important emerging approach in computer-aided design. The bulldozer is but one of a whole family of procedural agents we imagine populating a virtual design and construction world, promoting the human codesigner to a position more like a conductor than a craftsperson; an appropriate role, we think, for landform sculpting as well as other environmental design tasks.

4 Acknowledgment

Caroline Westorts research is supported by the US Army European Research Office, Contract No. N68171-94-9153.

5 References

1. Weibel, R. & M Heller, 1991. 'Digital Terrain Modelling', in Maguire, Goodchild & Rhind (eds) Geographical Information Systems: Principles & Applications. London: Longman Scientific.

2. Laurie, M, 1975. An Introduction to Landscape Architecture, New York: Elsevier

3. Motloch, J L. , 1991. Introduction to Landscape Design,. New York: Van Nostrand Reinhold.

4. Strom, S & K Nathan, 1988. Site Engineering for Landscape Architects; New York: Van Nostrand Reinhold.

5. Ervin, 5 1994, "Digital Terrain Models", Landscape Architecture Magazine, January 1994.

6. Laurini, R & D Thompson, 1992. Fundamentals of Spatial Information Systems. London, Academic Press

7. Mealing, 5 1994. Three Dimensional Modelling and Rendering on the Macintosh, Oxford, England.

8. Mallary, R & M Ferraro, 1977. "ECO-SITE: An application of Computer Aided Des n to the Composition of Landforms for Reclamation', Computer Graphics 11, Summer

9. Weibel, R. & M Heller, 1991. "Digital Terrain Modelling', in Maguire, Goodchild & Rhind (eds.) Geographical Information Systems: Principles & Applications. London: Longman Scientific.

10. Beardsley, J, 1984. Earthworks and beyond: contemporary art in the landscape, New York: Abbeville Press

11. Li, X, 1993. Doctoral Dissertation, Institute for Simulation and Training, University of Central Florida.