Terrain Sketching

James Gain[*]
University of Cape Town

Patrick Marais[†]
University of Cape Town

Wolfgang Straßer[‡]
WSI/GRIS, University of Tübingen

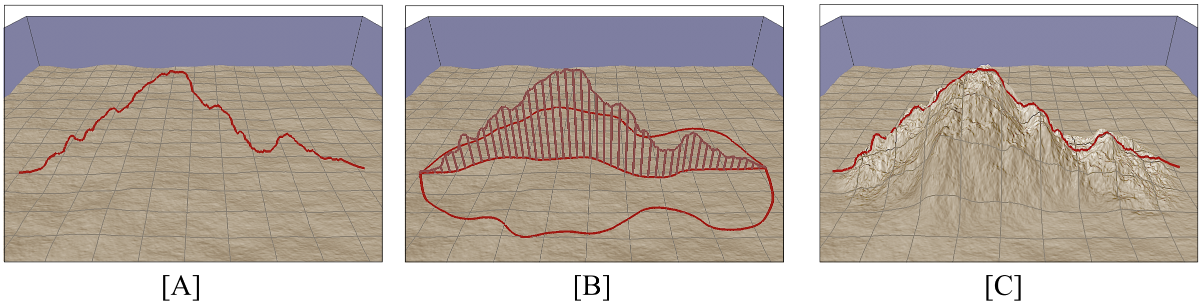

Figure 1: *Modeling landscapes by Terrain Sketching: [A] the user sketches a silhouette curve, and [B] optionally modifies the shadow and boundary curves. [C] A matching landscape is created by surface deformation with noise propagation.*

Abstract

Procedural methods for terrain synthesis are capable of creating realistic depictions of heightfield terrains with little user intervention. However, users often *do* wish to intervene in controlling the placement and shape of landforms, but without sacrificing realism. In this paper, we present a sketching interface to procedural terrain generation. This system enables users to draw the silhouette, spine and bounding curves of both extruding (hills and mountains) and embedding landforms (river courses and canyons).

Terrain is interactively generated to match the sketched constraints using multiresolution surface deformation. In addition, the wavelet noise characteristics of silhouette strokes are propagated to the surrounding terrain. With terrain sketching users can interactively create or modify landscapes incorporating varied and complex landforms.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric Algorithms

1 Introduction

Computer-based simulation of natural environments, of which terrains are a vital component, has found widespread application in computer games, visual effects, training and simulation. Such environments are typically both large and highly detailed and modeling them by hand is infeasible. Fortunately, procedural methods [Ebert et al. 2003] are capable of automating the process. They exhibit significant database amplification: in the most extreme case a single pseudo-random seed value can be used to procedurally generate an entire virtual world.

For the particular case of generating procedural terrain, there seem to be three broad strategies: noise functions, erosion simulation and texture synthesis. Noise functions, such as fractional Brownian motion, and particularly its extension to hybrid and ridged multifractals [Ebert et al. 2003], can be generated in real-time and capture the variable multi-scale nature of terrains. However, to introduce true drainage and weathering patterns requires a simulation of thermal and hydraulic erosion. Such simulations [Musgrave et al. 1989; Chiba et al. 1998] often represent severe simplifications, of necessity given limitations in computational resources and the complexity of the physical processes involved. More recently procedural techniques have begun to borrow realism from existing terrains by using texture synthesis [Brosz et al. 2006; Dachsbacher 2006; Zhou et al. 2007] of Digital Elevation Models (DEMs), such as those provided by the U.S. Geological Survey, with compelling results.

Paradoxically, the automation inherent in procedural methods is both a strength and weakness, in that reducing user involvement to manipulating a set of procedural parameters also means reducing their control. Amburn et al. [1986] provide an early counter to this trend by allowing users to script constraints for subdivision processes, but this does little to address usability issues. A means of improving both usability and control is to use image maps as input [Perlin and Velho 1995], exploiting the maturity and familiarity of paint programs. For instance, a typical interface to procedural terrain allows users to paint bumps at different frequencies, from a top-down perspective. An alternative is to develop an interface analogous to pen-and-paper sketching. Such sketch-based interfaces are often applied to geometric modelling [Zeleznik et al. 1996; Karpenko and Hughes 2006; Nealen et al. 2007] but they have also been used with success in the procedural modeling of trees [Okabe et al. 2005], flowers [Ijiri et al. 2005], clothing [Turquin et al. 2007] and, to a limited extent, terrains [Cohen et al. 2000; Watanabe 2004].

We expect that sketching of terrains is more natural to most users than either image painting or parameter manipulation, based, in part, on the long tradition of landscape sketching in both the arts and sciences [Hutchings 1960]. Even in the age of digital photography, field sketching is advocated in the geosciences as an aid

[*] e-mail: jgain@cs.uct.ac.za
[†] e-mail:patrick@cs.uct.ac.za
[‡] e-mail:wolfgang.Strasser@uni-tuebingen.de

to the detailed observation and abstraction of landscape [Green 1998]. Generally, an artist will sketch outlines of landforms and use crosshatching or pencil shading as a further indicator of slope. The outline strokes do not necessarily form strict silhouettes from the artist's precise viewpoint. Rather, they represent silhouettes from nearby (particularly lower) views [Whelan and Visvalingam 2003].

Motivated by the success of sketching interfaces in other areas of procedural modeling and the strength of landscape sketching as an established art-form we introduce Terrain Sketching. Fundamental to our interface is the drawing of feature curves, which represent the curvilinear, possibly branching, local extrema of a landscape, such as mountain ridges and river courses. We note that these features, as with paper-based sketching, do not always represent silhouettes, particularly in the case of indenting landforms.

When viewed from above, ridges and valleys rarely follow absolutely straight lines and so, unlike previous approaches [Cohen et al. 2000; Watanabe 2004; Zhou et al. 2007], we support truly general $2\frac{1}{2}$D feature curves. This has fundamental implications for both the design of the interface and its procedural realization. For instance, we decompose the process of creating a feature into sketching the silhouette (view-plane) and shadow (base-plane). This is beneficial, especially for indenting features, where it is easier for a user to first draw the shadow from above onto an existing landscape. A user is further able to control shape by sketching boundary curves where the landform joins with the surrounding terrain.

Based on these curves, a fast multiresolution surface deformation can be applied to the terrain, so that it conforms to user constraints while blending smoothly with the existing heightfield. As part of this process, the wavelet noise characteristics [Cook and DeRose 2005] of the feature curve are analysed and transfered to the resulting landform. This synthesis technique faithfully matches (rather than merely approximate) the user's strokes, which is another point of departure from previous work.

To be fair, previous methods place themselves as rapid prototyping tools, whereas Terrain Sketching is intended, in the spirit of FiberMesh [Nealen et al. 2007], for modeling fully-realized landscapes suitable for export to higher-end applications.

In keeping with most terrain synthesis systems, in both research and industry, and virtually all terrain interchange formats, we represent terrains as heightfields. This does mean, of course, that it is not possible to sketch fully three-dimensional terrain features such as overhanging cliffs, river cutaways and rock arches.

To summarize, our key contributions are: an interface that supports truly curvilinear, indenting and extruding landforms using both general $2\frac{1}{2}$D feature curves and attendant boundary curves, and a procedural method that faithfully matches these sketched user constraints while producing realistic terrains.

The remainder of this paper has the following structure. Section 2 provides a more detailed discussion of previous terrain sketching systems. The sketching interface is presented in section 3 and the underlying algorithm for surface deformation in section 4. The paper concludes with a discussion of the system's performance, usability and versatility (section 5) and a summary and recommendations for future work (section 6).

2 Related Work

Ours is not the first terrain sketching system, but it is the first to allow users to control silhouette, shadow and boundary curves.

An early example of Terrain Sketching appears in the *Harold* system [Cohen et al. 2000]. The fundamental question in sketching interfaces is how to project 2D strokes in screen space into 3D curves in world space. In Harold, a projection plane for silhouettes is created by projecting the endpoints onto existing terrain; an approach that we adopt. However, in other respects, Harold is limited to straight shadows and boundaries, and deals poorly with noisy strokes, which introduce striation lines.

Watanabe [2004] also employs a straight shadow line but infers a boundary for landforms using local minima and maxima in the user's sketch, effectively rotating a simplified version of the silhouette onto the base plane. Watanabe layers noise onto the terrain after, rather than during, surface deformation, which means that the user's strokes are never interpolated exactly. Their system does, however, support extremely rapid prototyping.

In contrast to these methods, Zhou et al. [2007] allow landforms to follow curvilinear paths using a top-down sketch map as guidance for a patch-based texture synthesis of terrain. Unfortunately, they provide no control over the height or boundary of the resulting landform except very indirectly through the type of DEM selected as an example. In this sense, their approach is closer to an image-map interface than sketching.

3 Interface

For a sketching interface to be useful for more than prototyping, it must support iterative refinement [Nealen et al. 2007], since users cannot be expected to finalize a complex curve with a single gesture. In Terrain Sketching, before a user commits a stroke it can be modified by oversketching, a process which parallels pencil-and-paper sketching. Also, wherever possible strokes are drawn onto existing surfaces, typically the terrain itself, which frees the user to change viewpoint during refinement.

Terrain Sketching has three distinct interface modes, each suited to creating different classes of landforms. Although modal interfaces are generally frowned upon, unless system status is signalled clearly [Norman 1990], we do take care to signal the current state using the color, weight and style attributes of strokes, as well as a mode icon.

3.1 Silhouette Mode

In silhouette mode (figure 2), a user begins by drawing and refining the silhouette of an intended terrain feature. This is committed with a single tap click and the system then generates attendant boundary and shadow curves, which are draped onto the existing terrain. By altering these, the user influences the footprint of the landform and the base-plane curve of the silhouette [Cohen et al. 1999]. Finally, terrain is created to fit these curves (as detailed in section 4).

In terms of implementation, the in-screen silhouette is projected onto a vertical plane passing through the stroke endpoints as raycast from the viewpoint through their on-screen position and onto the landscape. In order to ensure a stable center of projection, the viewpoint must remain fixed during sketching of a single silhouette, but this is the only case where such a restriction is required; other modes (and even the other phases of silhouette mode) allow repositioning of the viewpoint because, in these cases, strokes are projected directly onto existing surfaces.

Subsequently, the initial shadow is defined by intersecting the silhouette projection plane with the landscape. As the shadow is modified the silhouette is reprojected from the original viewpoint onto the vertical ruled surface passing through the shadow, enabling the silhouette to bend.

A plausible default boundary is more involved. We assume (as

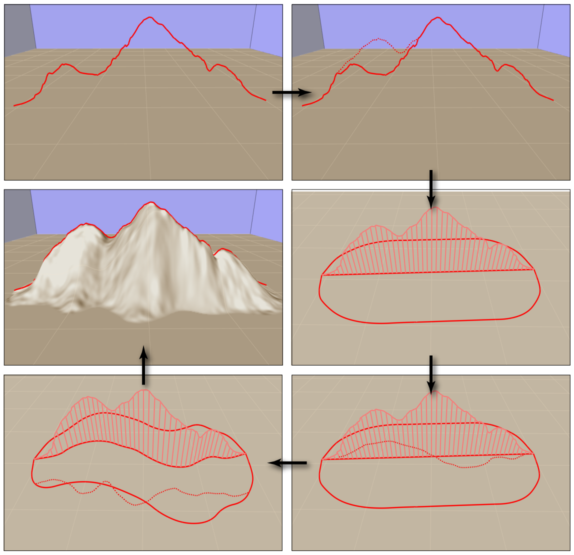

Figure 2: *Silhouette sketching mode: the user sketches a silhouette stroke and modifies it by oversketching. Next, default shadow and boundary curves are created, which the user can adjust, before the final landform is synthesised.*

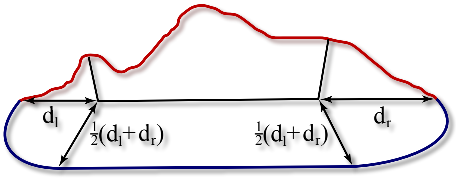

Figure 3: *Default boundary. the shoulder regions of the smoothed silhouette are segmented out and their base lengths d_l and d_r used to define a capsule-like boundary.*

shown in figure 3) that the length of the shoulder slopes of the silhouette, i.e., those sections at the end of the curve leading up to a local maxima/minima or at least a levelling-off signaled by high curvature, are representative of cross sections of the landform in other directions. The default boundary is given a capsule-like shape with rounded ends matched to these shoulders and a width that averages the shoulder lengths. To avoid problems with high frequency noise the silhouette is smoothed during segmentation of the shoulders. Thus, a silhouette leading gradually to a single maxima (e.g., a volcanic peak) will have a circular footprint, while a silhouette with short steep shoulders (e.g., a chain of mountains) will have an elongated capsule as footprint. This capsule, defined in the base plane, is then projected up onto the landscape where it can be edited.

There are various illegal conformations of the silhouette, shadow and boundary curves, which break projection and synthesis operations. For instance, the boundary should not cross over the shadow, nor should the silhouette or shadow fold back on or intersect themselves. Such contraventions are automatically detected and removed after the pen is lifted.

For modeling purposes the user is restricted to a square, albeit large, terrain. The silhouette can be pinned, not only to the terrain, but also to side-walls erected along its edges. This enables landforms that extend to the edge and continue into neighboring patches.

3.2 Aerial Mode

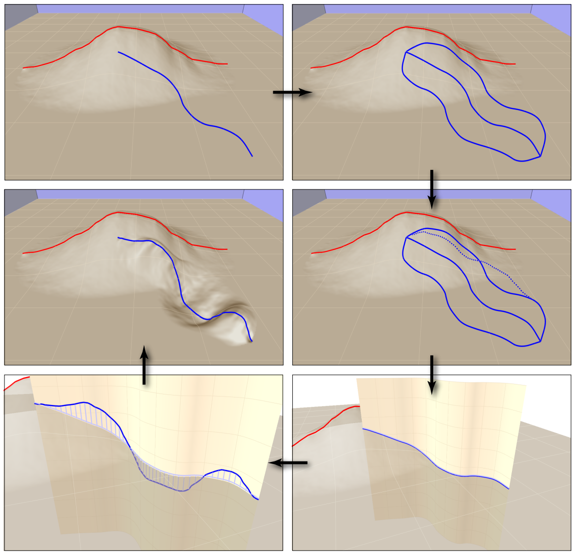

Figure 4: *Aerial sketching mode: the user sketches a shadow stroke onto existing terrain and associated boundary curves are initialized and optionally modified. Then the user sketches a silhouette stroke onto a vertical surface ruled through the shadow, before the final landform is synthesised.*

While silhouette mode is related to real-world landscape sketching, the correspondence is not exact. In particular, the spine of indenting landforms, such as canyons, are often partially, if not completely, occluded from a particular viewpoint and sketching them becomes perceptually awkward. Usually, it is the boundary curves of such features (e.g., the lips of a canyon) that appear in paper-based sketches. We briefly considered allowing users to sketch from the underside of terrains but this proved confusing and unnatural.

In aerial mode (figure 4), the problem of sketching indenting landforms is solved by reversing the order of sketching, starting with the shadow curves (usually from a top-down angle), followed by the boundaries and finally the silhouette, which is drawn onto a vertical surface ruled through the shadow. It is worth noting that all of these sketches are made onto existing surfaces.

3.3 Region Mode

Once the user commits to a final landform configuration in silhouette or aerial mode, a multiresolution deformation process (described in section 4) is used to warp the terrain so that it interpolates the boundary and silhouette strokes in a realistic way. As part of this process, the noise characteristics of the silhouette are analyzed and transferred to the landform. Thus, a jagged silhouette will generate a correspondingly rough, broken landform, while a smooth silhouette will create gentler terrain. However, finer control over terrain noise is often warranted. For instance, a user may require a landform with a ragged ridge-line to have smooth sides.

This is solved by the region mode (figure 5), where a user marks out a region of the terrain and then sketches a stroke with representative noise. Once committed (with a tap gesture) the demarcated region is overlaid with matching noise. The region is specified using a closed, possibly convoluted, stroke drawn onto the terrain. In order to avoid ambiguity in identifying the interior, self-intersections of this loop stroke are automatically removed. Subsequently, noise is

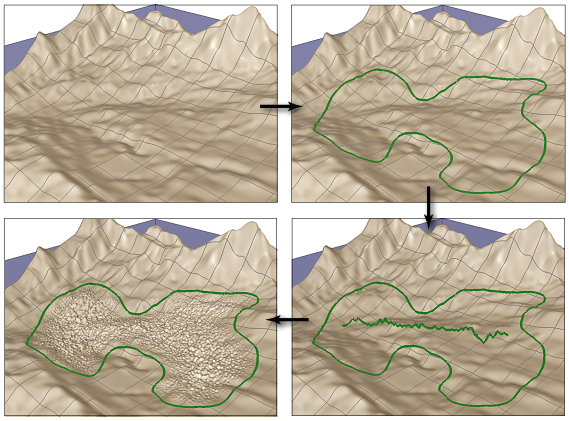

Figure 5: *Region sketching mode: the user sketches a closed region stroke onto existing terrain and then draws a representative noise stroke. Finally, matching noise is transferred to the region.*

specified using a single silhouetting stroke sketched anywhere on the terrain. This stroke is intended to be representative: its noise characteristics are extracted without it being interpolated.

This mode does not alter the underlying shape of the terrain but, rather, controls the surface roughness. As such, region mode is best applied after the aerial or silhouette modes have been used to define the overall structure of the landscape, where the fine-scale surface characteristics need adjusting.

3.4 Occlusion and Deletion

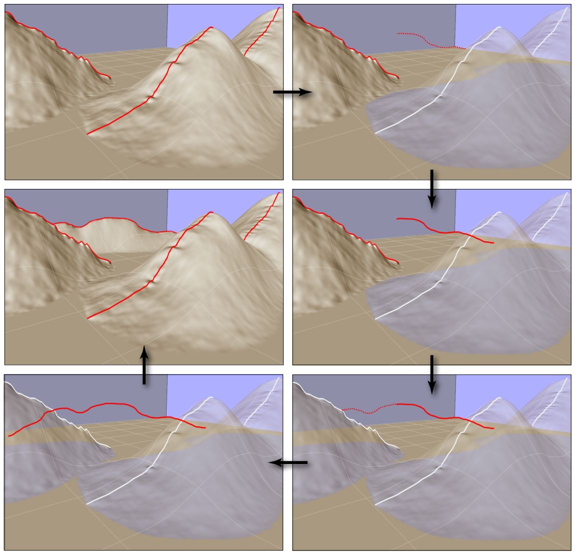

Figure 6: *Handling occlusion: if a stroke is paused near a terrain silhouette then the associated landform is rendered semi-transparent and the stroke can be continued behind. The same stroke can thread behind several landforms before being finalised.*

Fixing the initial viewpoint in silhouette mode is problematic when the user intends a silhouette to end behind an existing landform. Unfortunately, recent advances in completing smooth objects in the presence of occluded strokes [Karpenko and Hughes 2006] are of little help. The difficulty lies in inferring the user's intentions with respect to slope, noise and endpoint positioning. Instead, we avoid guesswork and allow the user to adjust the opacity of an existing landform (figure 6) by pausing a stroke near it's image silhouette in the current view. The stroke can then be continued across previously occluded terrain. It is even possible to remove multiple occluding landforms with this strategy.

We augment the image-based silhouette extraction of Raskar and Cohen [Raskar and Cohen 1999] to allow picking of landforms by colour indexing silhouettes. This requires uniquely associating triangles with particular landforms: easy enough when we have their boundaries available but more difficult when importing terrains from an external application. Techniques for automated landform segmentation do exist [Dragut and Blaschke 2006] but lie outside the scope of the current work.

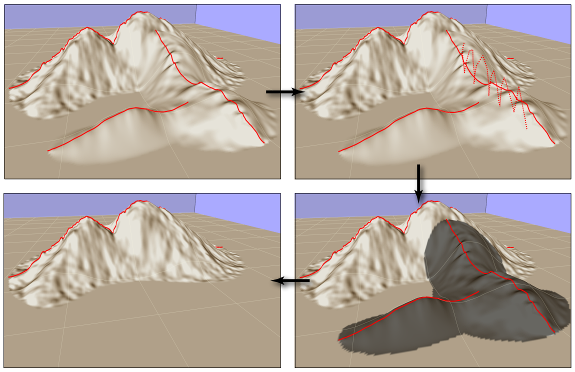

Figure 7: *Deleting landforms: a scratch gesture across a feature stroke selects for deletion, after which the selected landform and all its dependents are highlighted, before the deletion is committed with a tap gesture.*

Ideally, silhouette, shadow and boundary curves should persist indefinitely as handles for later editing [Nealen et al. 2007]. Unfortunately, altering a section of underlying terrain will invalidate all landforms whose endpoints are pinned there. In fact, landforms have an implicit dependency graph and only leaves of the graph can be freely altered, which is why we emphasize refinement during initial landform creation. On the other hand, deletion can be implemented (figure 7), as long as a user is willing to also remove a landform's children in the dependency graph. This graph is built explicitly by establishing a child-parent relationship wherever a new landform's footprint overlaps an existing landform. In this way, a coarse form of order-independent undo operation is supported.

4 Algorithms

In order to generate terrain, the user's sketched curves are submitted as constraints to a multiresolution deformation and noise propagation process. This deformation is required to interpolate the silhouette but not stray outside the boundary curve, while interactively generating results indicative of real terrain.

Our deformation approach relies on multiresolution techniques, in particular discrete wavelet transforms [Stollnitz et al. 1996]. This follows in a long tradition of multiresolution representations of terrain, for the purpose of fractal analysis [Mallat 1989] and efficient level-of-detail rendering [Hoppe 1998].

For wavelet decomposition we adopt the notation and techniques of Cook and DeRose [2005]. A regularly sampled signal (I), be it a 2D heightfield or 1D silhouette function, can be smoothed using downsampling to produce a lower frequency signal of half the resolution ($I^{\downarrow}$). The reverse operation is upsampling ($I^{\uparrow}$), which doubles the resolution but retains the same frequency characteristics. In this way, a signal can be smoothed ($I^{\downarrow\uparrow}$) and its details in a particular frequency band, with a filter width denoted by ϕ, extracted ($D = I - I^{\downarrow\uparrow}$). By repeated downsampling, upsampling, and differencing we derive detail signals at different scales but with constant resolution (indexed by $i = 0, \ldots, m$, where $i = m$ is the most detailed and $i = 0$ is the most smooth). Strictly speaking this makes our approach multi-scale rather than multiresolution.

A side benefit of this multi-scale hierarchy is that Exaggerated Shading [Rusinkiewicz et al. 2006], a nonphotorealistic rendering method that highlights shape and detail through the principles of cartographic relief, can be incorporated without the usual extensive pre-processing.

We design an efficient curve-based spatial deformation method, specifically targeted at terrain synthesis, which incorporates sketch-derived noise while interpolating the silhouette and smoothly blending with the existing terrain. Spatial deformation is far from the only option: linear variational surface deformation [Botsch and Sorkine 2008] is attractive because of its variational optimization principles and consequent fairing; and transfinite interpolation of a multiresolution hierarchy of subdivision surfaces [Schaefer et al. 2004] would allow true frequency-band limiting. But, ultimately, these alternatives are too expensive to be interactive across the multiple scales involved, particularly since the terrain and silhouette change at each smoothing scale.

Our method proceeds iteratively through smoothness levels $i = 0, \ldots, m$ of a terrain ($\mathcal{T}$). Within the landform boundary ($\mathcal{B}$), Wavelet Noise [Cook and DeRose 2005], whose variance is derived from an analysis of the silhouette curve ($\mathcal{H}$), is blended in. Different scales of silhouette detail are also applied, at each step shrinking the boundary in accordance with the filter width (ϕ_i), so that fine detail becomes more localised.

Parametrisation. Both noise propagation and terrain deformation for a given smoothness level i depend on a parametrisation step, in which terrain points (P) are attached to nearby points on the base-plane projection of the shadow ($\mathcal{S}'$) and boundary ($\mathcal{B}'_i$). Figure 9 shows the parametrisation and deformation process in detail.

Working in the base-plane, the closest point to P' on the line through the endpoints of the shadow provides a parameter t, which can be used to access a corresponding point on the shadow curve ($\mathcal{S}'(t)$) and boundary for level i, ($\mathcal{B}'_i(t) = (\phi_i/\phi_0)\mathcal{B}'_0(t)$), both parametrized on the base-line. (Note that the subscript i indicates a curve or variable specific to a particular smoothness level, and prime indicates a base-plane projection).

Both noise and deformation are weighted according to the relative position of the point P' between shadow and boundary:

$$W_i(P) = d\left(\frac{\|P' - \mathcal{S}'(t)\|}{\|\mathcal{B}'_i(t) - \mathcal{S}'(t)\|}\right), d(a) = (a^2 - 1)^2. \tag{1}$$

This C_1 blending function ensures full weighting on the shadow tapering to zero at the boundary. Note that, while the shadow curve is independent of the smoothness level, the boundary is scaled inwards with increasing i to localize detail.

The embedding of points indirectly on the base line rather than directly on the shadow curve is a departure from conventional curve-based deformations [Milliron et al. 2002]. However, our approach is at least an order of magnitude faster, since it avoids repeated closest point searches of the shadow curve and, furthermore, it is C_1, unlike other curve-based schemes which exhibit seams of C_0 discontinuity where the parametrization jumps along the medial axis of a bent curve. On the other hand, a base-line embedding orients detail orthogonally, which can cause shearing artefacts if the shadow curve angles away from the base line too severely, and disallows shadow curves that fold over relative to the base line. In these cases we must revert to a full curve search.

Noise Propagation. For level i, noise is applied as a smooth blend between terrain detail at P, $\Delta\mathcal{T}_i(P) = \mathcal{T}_i(P) - \mathcal{T}_{i-1}(P)$, and a wavelet noise layer, $\Delta\mathcal{N}_i(P, \sigma_i)$, to produce a new terrain offset, $\overline{\Delta\mathcal{T}_i}(P)$, as follows:

$$\overline{\Delta\mathcal{T}_i}(P) = (1 - W_0(P)) \cdot \Delta\mathcal{T}_i(P) + W_0(P) \cdot \Delta\mathcal{N}_i(P, \sigma_i). \tag{2}$$

The variance σ_i is derived from the silhouette $\mathcal{H}_i$. Also, observe that the weighting function (W_0) depends on the original, rather than scaled, boundary so that noise can be blended across the entire footprint.

We were initially concerned that the noise statistics of a sketched silhouette might not accurately reflect real terrain and so we undertook a small-scale calibration experiment. Each of 10 subjects were shown a variety of heightfield landscapes (flatlands, hills, mountains and canyons) and asked to sketch characteristic silhouettes with both pen and tablet, and mouse, using our interface. The presentation of terrains and use of input device were permuted between subjects to control for learning effects and the subjects had little or no experience with either pen and tablet or computerized sketching.

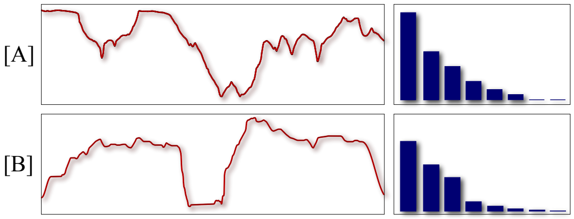

Figure 8: *Characteristic silhouettes and noise variance values: [A] real-world DEM terrain and [B] a user's sketch*

We then compared the noise statistics of the example terrains against the subjects' sketches. As expected, subjects had the most difficulty in reproducing very high frequency noise and, also as expected, the pen and tablet, since it is a drawing rather than pointing device, was more accurate in this respect even for inexperienced users. Generally, terrain exhibits a smooth exponential decay in variance (see figure 8[A]). While users are able to track this trend (see figure 8[B]) they cannot always do so accurately (the mean error of users' sketches as a percentage of terrain variance ranges from 5% to 50%). So, instead of using raw variance σ_i we fit an exponential decay function through the sketched noise and use this instead. In the case of mouse input we additionally discard the final few variance values from the fit.

Deformation. Once noise has been incorporated, the parametrised terrain is displaced according to the difference between the silhouette and its vertical projection onto the terrain. This deformation is contained inside a boundary that is scaled inwards to localize detail:

$$\overline{\overline{\Delta\mathcal{T}_i}}(P) = W_i(P) \cdot (\mathcal{H}_i(t) - \mathcal{T}_i(\mathcal{H}_i(t))). \tag{3}$$

The ends of the silhouette require special attention in order to guarantee C_1 continuity. Although we match scales (using ϕ), ultimately the silhouette and terrain are smoothed independently and

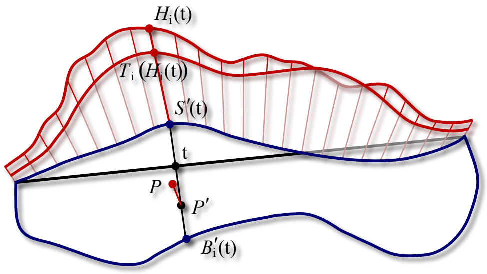

Figure 9: *Parametrisation and deformation at a particular smoothness (i): a terrain point P with base-plane projection P' is parametrised with respect to the base line as $t \in [0, 1]$. This provides points on the base projection of shadow ($\mathcal{S}'(t)$) and boundary ($\mathcal{B}'_i(t)$). Deformation further depends on the height difference of the silhouette ($\mathcal{H}_i(t)$) and its projection onto the terrain ($\mathcal{T}_i(\mathcal{H}_i(t))$).*

there is no guarantee that the endpoints of $\mathcal{H}_i$ will touch the terrain $\mathcal{T}_i$, smoothly or otherwise. To counter this we truncate the silhouette at either end by the filter width (ϕ_i), which narrows with increasing frequency, and fit a C_1 continuous blending section between terrain and silhouette. We also use the same filter width to blend in noise at the endpoints.

In summary, terrain is warped by proceeding through the multiresolution hierarchy and applying parametrisation (eqn. 1), noise blending (eqn. 2) and deformation (eqn. 3) to terrain vertices within the landform footprint.

The region mode demands a different strategy, since there is no notion of a shadow curve for parametrizing terrain points. A medial axis might be adapted to this purpose but, depending on the convolutions of the region loop $\mathcal{L}$, this could branch in a number of places. Rather, we parametrize points according to the distance to the closest point $\mathcal{L}(t)$ on the region loop and, for the upper multiresolution levels, blend between existing terrain and noise over the filter width.

The central challenge then becomes: how to efficiently generate a $2D$ distance field over the interior of the loop. First, we treat the loop as a polygon and use a scan-line polygon fill to determine its interior. Next, we place a dense sampling of m points along the loop and, inspired by Hausner [2001], use the z-buffer to determine the index of the closest loop sample. A collection of m uniquely-coloured cones is placed, each with its tip at a loop sample and its axis pointing away from the camera. This scene is rendered using an orthogonal projection, flat shading, and a screen resolution that matches the terrain resolution. Now, each pixel represents a terrain point and by z-buffer sorting its colour provides an index to the closest cone and corresponding loop sample. It is then trivial to derive an actual distance. Exploiting the GPU in this fashion improves speed by about a factor of 30 over the naïve approach of comparing every grid element against every loop sample, while retaining sub-pixel accuracy. This approach does not suffer from the C_0 discontinuities we were so careful to circumvent with aerial and silhouette modes, because the blending is based on a smooth distance field without any silhouette-dependent deformation component.

The outcome, for aerial and silhouette modes, is a surface deformation method that adjusts with scale, applies noise derived from the user's sketch and interpolates both the boundary and silhouette curves and, for the region mode, the result is a smooth concentric blending between existing terrain and derived noise.

5 Results

In this section we consider the following aspects of system performance: computational efficiency, ease of use, the range of achievable landforms, and the geomorphological realism of the resulting synthesis.

Efficiency. Our surface deformation scheme is highly accelerated and capable of deforming a 512×512 grid with 6 wavelet decomposition levels in under 2.3 seconds on a single CPU Intel 2.33 Ghz processor with 2 GB RAM. Region noise is even faster, requiring only 1.8 seconds for the same grid size and decomposition. However, users generally draw features that are much smaller than this. In complexity terms, deformation is linear in the number of grid elements undergoing deformation.

Usability. We conducted an informal usability study of the system by having an experienced user design a number of terrains, each within an alloted 30 minutes. The results appear in figure 12. Furthermore, the calibration experiment, while not intended to examine ease-of-use directly, does indicate that some users are surprisingly accurate in sketching the noise characteristics of real terrain, which we attribute in part to the precision allowed by iterative oversketching. Additional usability testing is certainly warranted but falls outside the scope of the current work.

Range. Figure 11 demonstrates, within a single landscape, the variety of landforms obtainable by Terrain Sketching, including steep slopes and indentations. Such effects are achieved by careful control over the interaction between silhouette, shadow and boundary. For instance, a cliff can be created by drawing a boundary curve that follows and lies close to the shadow. Another aspect of real terrain is its variation in noise characteristics [Musgrave et al. 1989]. Here again, Terrain Sketching provides control with the region mode.

Realism. We test realism by importing and modifying an existing DEM landscape (figure 10). The join between DEM and surface deformation is barely visible (figure 10[B]). However, modifications can be distinguished by careful inspection, since in real terrain, unlike wavelet noise, small-scale features such as erosion gullies are not anisotropic.

6 Conclusion

This paper introduced Terrain Sketching, a procedural modeling system that provides users with a high degree of control over the shape of landforms through a sketching interface. By means of surface deformation with noise propagation the system is capable of generating terrains that are large, heterogeneous and realistic. This has application in generating environments for computer games, visual effects and simulation.

There are several directions in which Terrain Sketching can fruitfully be extended. During surface deformation we transplant noise from the user's sketch to the terrain. Currently, this is implemented by fitting an exponential decay to the noise variance provided by the user. This ensures that the noise is more characteristic of real terrain but it is an approximation and we expect that a more nuanced mapping could be obtained through machine learning. Further geomorphological realism could be achieved by adapting recent research in terrain texture synthesis [Brosz et al. 2006; Zhou et al. 2007]. The difficulty here would be in constraining the synthesis to accurately track the silhouette and boundary curves.

Acknowledgements

This work is based upon research supported by the National Research Foundation of South Africa.

References

Amburn, P., Grant, E., and Whitted, T. 1986. Managing geometric complexity with enhanced procedural models. *SIGGRAPH Comput. Graph. 20*, 4, 189–195.

Botsch, M., and Sorkine, O. 2008. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics 14*, 1, 213–230.

Brosz, J., Samavati, F. F., and Sousa, M. C. 2006. Terrain synthesis by example. In *GRAPP 2006: Proceedings of the First International Conference on Computer Graphics Theory and Applications*, 122–133.

Chiba, N., Muraoka, K., and Fujita, K. 1998. An erosion model based on velocity fields for the visual simulation of mountain scenery. *The Journal of Visualization and Computer Animation 9*, 4 (October/December), 185–190.

Cohen, J. M., Markosian, L., Zeleznik, R. C., Hughes, J. F., and Barzel, R. 1999. An interface for sketching 3d curves. In *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, ACM, New York, NY, USA, 17–21.

Cohen, J. M., Hughes, J. F., and Zeleznik, R. C. 2000. Harold: a world made of drawings. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, ACM, New York, NY, USA, 83–90.

Cook, R. L., and DeRose, T. 2005. Wavelet noise. In *SIGGRAPH '05*, ACM, New York, NY, USA, 803–811.

Dachsbacher, C. 2006. *Interactive Terrain Rendering: Towards Realism with Procedural Models and Graphics Hardware*. PhD thesis, Universität Erlangen-Nürnberg.

Dragut, L., and Blaschke, T. 2006. Automated classification of landform elements using object-based image analysis. *Geomorphology 81*, 3–4 (November), 330–344.

Ebert, D. S., Musgrave, F. K., Peachey, D., Perlin, K., and Worley, S. 2003. *Texturing and Modeling: A Procedural Approach*, 3 ed. Morgan Kaufmann Publishers Inc.

Green, D. R. 1998. Mental mapping and field sketching in geography at the university of aberdeen. *Cartographic Journal 35*, 1, 83–85.

Hausner, A. 2001. Simulating decorative mosaics. In *SIGGRAPH '01*, ACM, New York, NY, USA, 573–580.

Hoppe, H. 1998. Smooth view-dependent level-of-detail control and its application to terrain rendering. In *IEEE Visualization 1998*, IEEE Computer Society, Los Alamitos, CA, USA, 35–42.

Hutchings, G. E. 1960. *Landscape Drawing*. London, Methuen.

Ijiri, T., Owada, S., Okabe, M., and Igarashi, T. 2005. Floral diagrams and inflorescences: interactive flower modeling using botanical structural constraints. In *SIGGRAPH '05*, ACM, New York, NY, USA, 720–726.

Karpenko, O. A., and Hughes, J. F. 2006. Smoothsketch: 3d free-form shapes from complex sketches. *ACM Trans. Graph. 25*, 3, 589–598.

Mallat, S. G. 1989. A theory for multiresolution signal decomposition: the waveletrepresentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11*, 7, 674–693.

Milliron, T., Jensen, R. J., Barzel, R., and Finkelstein, A. 2002. A framework for geometric warps and deformations. *ACM Trans. Graph. 21*, 1, 20–51.

Musgrave, F. K., Kolb, C. E., and Mace, R. S. 1989. The synthesis and rendering of eroded fractal terrains. In *SIGGRAPH '89*, ACM, New York, NY, USA, 41–50.

Nealen, A., Igarashi, T., Sorkine, O., and Alexa, M. 2007. Fibermesh: designing freeform surfaces with 3d curves. In *SIGGRAPH '07*, ACM, New York, NY, USA, 41.

Norman, D. 1990. *The design of everyday things*. Doubleday.

Okabe, M., Owada, S., and Igarash, T. 2005. Interactive design of botanical trees using freehand sketches and example-based editing. *Computer Graphics Forum 24*, 3, 487–496.

Perlin, K., and Velho, L. 1995. Live paint: painting with procedural multiscale textures. In *SIGGRAPH '95*, ACM, New York, NY, USA, 153–160.

Raskar, R., and Cohen, M. 1999. Image precision silhouette edges. In *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, ACM, New York, NY, USA, 135–140.

Rusinkiewicz, S., Burns, M., and DeCarlo, D. 2006. Exaggerated shading for depicting shape and detail. In *SIGGRAPH '06*, ACM Press, New York, NY, USA, 1199–1205.

Schaefer, S., Warren, J., and Zorin, D. 2004. Lofting curve networks using subdivision surfaces. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ACM, New York, NY, USA, 103–114.

Stollnitz, E. J., DeRose, T. D., and Salesin, D. H. 1996. *Wavelets for Computer Graphics: Theory and Applications*, 1 ed. Morgan Kaufmann Publishers Inc.

Turquin, E., Wither, J., Boissieux, L., Cani, M.-P., and Hughes, J. 2007. A sketch-based interface for clothing virtual characters. *IEEE Computer Graphics & Applications* (January/February).

Watanabe, N. 2004. *A Sketching Interface for Terrain Modeling*. Master's thesis, University of Tokyo.

Whelan, J. C., and Visvalingam, M. 2003. Formulated silhouettes for sketching terrain. In *TPCG '03: Proceedings of the Theory and Practice of Computer Graphics 2003*, IEEE Computer Society, Washington, DC, USA, 90.

Zeleznik, R. C., Herndon, K. P., and Hughes, J. F. 1996. Sketch: an interface for sketching 3d scenes. In *SIGGRAPH '96*, ACM, New York, NY, USA, 163–170.

Zhou, H., Sun, J., Turk, G., and Rehg, J. M. 2007. Terrain synthesis from digital elevation models. *IEEE Transactions on Visualization and Computer Graphics 13*, 4, 834–848.

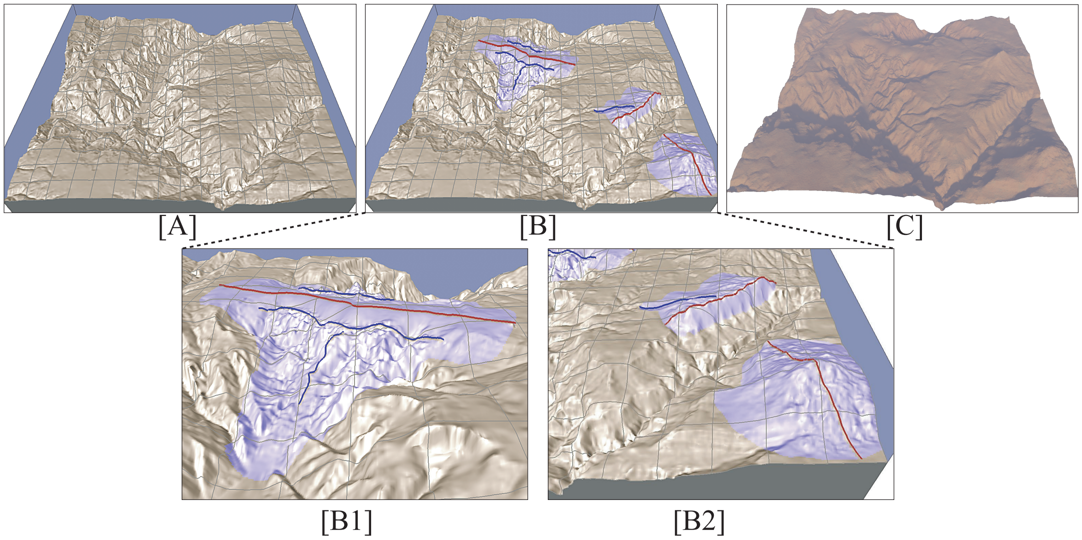

Figure 10: *Results - modifying existing terrain. A canyon [A] is modified by deformation [B] to create a bridge, mound and cliff extension. Details in close-up are provided [B1, B2], as well as a rendered version of the final terrain [C].*

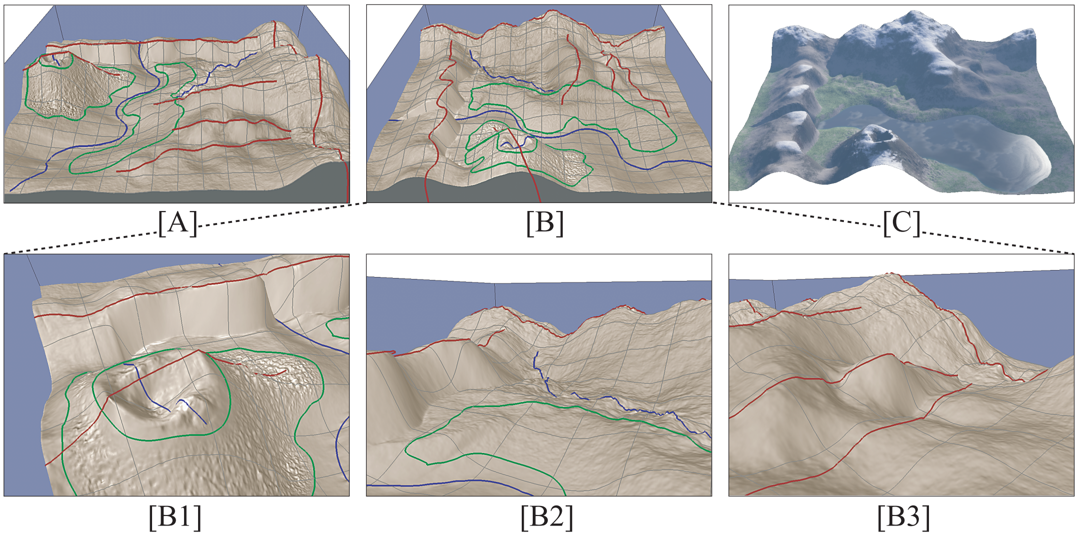

Figure 11: *Results - range of landforms. A variety of landscape features, including jagged peaks, extensive cliffs, a volcanic cone, rolling hills in different orientations, and a river canyon are sketched in a single terrain [A, B], with close-ups [B1-B3] and a final rendering [C].*

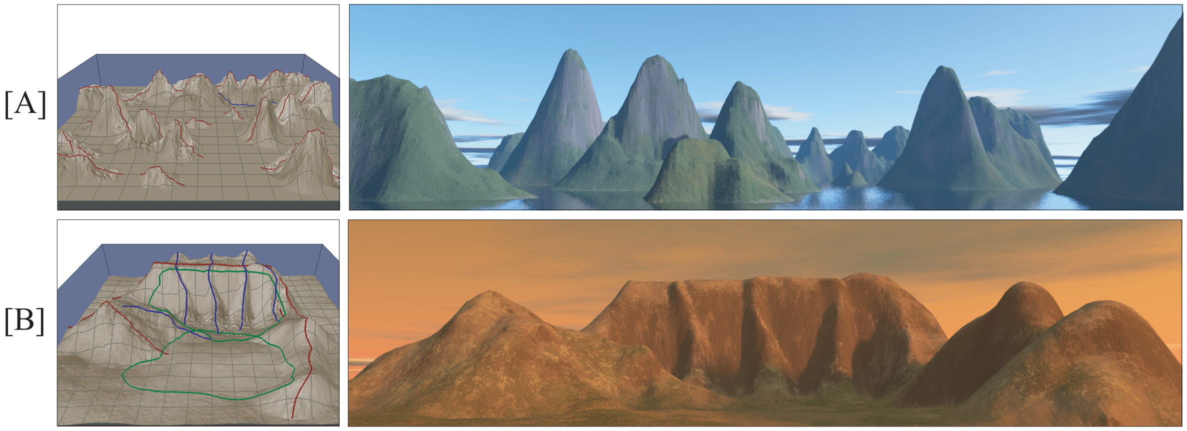

Figure 12: *Results - usability. Two sample terrains: [A] Tower Karst Islands, [B] A flat topped mountain, each modeled in under* 30 *minutes.*