Rapid Modeling of Geology

Morten Bendiksen*
Supervised by: Endre M. Lidal†, Ivan Viola‡

Institute of Informatics
University of Bergen

Abstract

Drawing three dimensional models of geological phenomena is today a process requiring training in specific programs and can be very time consuming. For illustration and communication of geological concepts, geologists therefore often limit themselves to drawing on paper or to two dimensional drawing applications.

We propose an approach for making rapid geologic illustrations in 3D. The novel idea for the approach consists of sketch based input on a cube in order to create a layered geological structure. Details can be added to the layers by sketching geological concepts such as rivers, mountains and valleys. Sedimentary deposits can be created through a procedural modeling approach. Awareness of the geologic domain enables a sparse amount of input strokes to be interpreted into geological structures. Results from user-studies show that the proposed approach can be used with success to model geological scenarios.

Keywords: Rapid modeling, Sketch based input and modeling, Geology

1 Introduction

Among geologists a common practice is to make sketched models by hand on either paper or computer. These sketches are used in both professional and educational settings, and facilitate communication and understanding. Geologic phenomena are four dimensional in nature since they occur over time in the three spatial dimensions. There are many techniques and standards for illustrating these phenomena in a two dimensional drawing. One can for example sketch three dimensional phenomena by using perspective drawing techniques, but the model is still confined to the 2D nature of the medium. These techniques and standards can also be limiting as they require significant time and training to master and understand. Before we started working on this project a problem was identified; there did not exist any tools aimed at helping geologists sketch 3D models for illustration purposes. On the computer it is already possible to make 3D models in traditional modeling approaches. However, existing tools are often complex, aimed at creating advanced and detailed models, and usually requires training to understand and use. It is from this background that the goal of this project was formed.

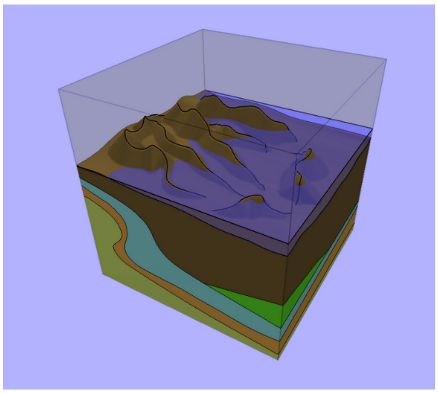

Figure 1: A typical sketch made with the proposed approach.

The goal is to enable the rapid creation of 3D models of geologic structures by creating an approach that lets geologists quickly specify input in an intuitive way that is easy to learn. The model will be used for illustrative purposes to facilitate communication between geologists by letting them create sketched models quicker, help lecturers explain concepts to students by creating models that can be changed interactively, and reduce the need for artistic skills and long training for students to master illustration techniques. The study of sedimentary layer structures and the processes that deform such layers are perhaps the fields of study that has resulted in the most knowledge about the history of the Earth. We concentrate most effort around the creation of rapid modeling techniques for geological layer structures. The aim is to create an approach for the creation of such structures.

2 Related work

The geologic understanding for this project was gained from reading the book “Geologi, stein, mineraler, fossiler og olje”, by Haakon Fossen [11]. For reference and technical background in the field of computer graphics the book “Real Time Rendering” [17] has been used. The book

*morten.bendiksen@gmail.com
†endre.lidal@gmail.com
‡ivan.viola@uib.no

"Curves and surfaces for CAGD: a practical guide" [10] was used to understand curve and surface theory.

Geology is a complex science with many subfields. Geomorphology is the study of landforms and the processes that shape the surface of the Earth. Sedimentology is about how particles are transported, where they are deposited and how they are compressed into rock. Structural geology is the study of how the rock layers and crust is deformed by various movements. Tectonics is closely related to Structural geology and describes the movement of Earth plates and how that causes the formation of mountain ranges and basins. There are many other fields, but these are the ones of most relevance to my research.

In geology, models are used for understanding and communicating about phenomena relating to the structure of the Earth and how it changes over time. A realistic geological scenario will follow certain constraints. Caumron et al. [7] give rules for modeling that define boundaries between layers. Traditional CAD systems have several problems when used to make geological models (Turner et al. [28], Kelk and Challen [14]). The gOcad tool described by Mallet [16], however, has been developed to make a CAD approach for geomodeling, by basing the modeling on a new interpolation method called "Discrete Smooth Interpolation". The geometry is defined by bridging together a set of nodes with a location is 3D space and with physical properties attached to these nodes. Petrel [25] is an example of a commercial program for geologic modeling that is in use. Most of such existing tools rely on an intensive work flow, and up to a year is spent on developing such models. Recently a need for rapid developments of geologic prospects have been identified.

Natali et al. explores different modeling techniques in their recent survey paper [18]. They show how geological modeling trends are approaching modeling methods that have been developed in computer graphics and give an in-depth description of selected methods that can be applied for geological modeling.

Approaches that aim at making geologic interpretation processes easier and quicker have emerged in recent years. Patel et al. describe techniques for rapid horizon extraction from seismic data in both 2D [22] and 3D [21]. Amorim et al. [1] have an interesting approach that allows sketching directly over the raw seismic reflection volume and its derived data to help build the structural model of the subsurface.

Procedural generation is often utilized to generate terrains. Before Olsen [20] fractal noise was mostly used to create terrain surfaces, because of computer limitations on simulating erosion processes. Olsen proposed a synthesized fractal terrain and applies an erosion algorithm on that. The representation is a 2D height-map. Hnaidi et al. [13] generate terrain that is constrained by a set of curves that characterize the features of the landscape. A method for eroding terrain is described by Benes et al. [3] where a concise voxel representation is created and then eroded by thermal weathering simulation. The representation allows for caves and hole structures. The same authors also propose a method for procedural modeling of terrain by hydraulic erosion [4]. Stava et al. [26] employ an interactive physics based hydraulic erosion.

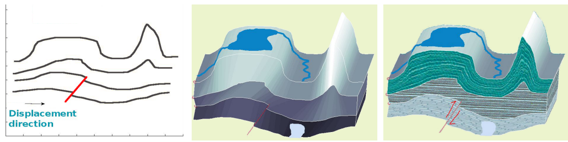

Figure 2: The proposed interface by Natali et al. [19].

Peytavie et al. [24] propose a way to model and render rock piles and stones which are found in most landscapes without any computationally demanding physically-based simulation. Peytavie et al. also have proposed a framework for representing complex terrains with such features as overhangs, arches and caves and including different materials such as sand and rocks [23].

Tasse et al. [27] propose a texture-based terrain synthesis framework controllable by a terrain sketching interface. They enhance the realism of the generated landscapes by using a novel patch merging method that reduces boundary artifacts caused by overlapping terrain patches.

Natali et al. [19] describe an approach where the user sketches the boundaries of geological layers. Then the user can sketch folding and faulting operations, and thus create many different scenarios. The input in this approach is restricted to making conceptually 2D sketches, although the visualization is in 3D. Projecting drawings on the 3D structure can however give more information and context to the 3D geometry. As far as we know, this is the only sketch based approach to modeling subsurface geological layers in 3D without measured data other than the one described in this paper. However, Lidal et al. [15] present Geological Storytelling, an approach for rapid and expressive geomodeling of a multitude of model variations in 2D over time.

Harold is an early example of a sketch based system that incorporates methods for sketching terrain, made by Cohen et al. [8]. In Harold, the user can sketch hills on the terrain by simple strokes that start and end on the terrain. The terrain is then warped to try and match the stroke. Watanabe et al. [29] made a further development of this, where the shape of the stroke also influences the width of hills that are generated, making for more natural looking hills. They also incorporated noise on top of the generated terrain to make the visualization more realistic. Gain et al. [12] later improved further on this by allowing the user to sketch the width of the hill and change the baseline along which this hill runs.

To achieve real-time terrain creation Bernhardt et al. [5] combine CPU and GPU processing in their sketch-based approach for generating and displaying complex and high-resolution terrains. The user can see the terrain changing as she is sketching. De Carpentier combines brushing and procedural terrain creation [9]. Applegate et al. [2] have a sketch based system for highway design. Their tool is guided by input sketches and a combination of prioritized constraints, including the curvature of roads, their inclination, and the volume of underlying terrain that is displaced. The rivers in my proposed solution are sketched in a similar way to this highway sketching method.

3 Methodology

We employ a sketch-based input that is projected onto a transparent cube. Layered geological structures are often sketched in a cube, and we therefore propose to mimic this technique for the sketching interface. The user can rotate around the cube and sketch on the four vertical faces of the cube. On the faces the user sketches the outlines of a surface (called a horizon) that will be the top boundary of one of the layer volumes. The horizon is then interpolated between the sketched outline. The top horizons of previously drawn layers become the bottom boundary of new layers. The user can thus create a stack of layers by adding the layers from bottom to top.

In order to change and model details on the layers, we propose methods for drawing further structure features such as mountains, rivers, valleys and deposits. The user can create ridges, rivers and valleys by sketching on the layers. Separate algorithms for each of the features will then modify the layer surface on which it was drawn. The features the user sketches are positioned on the 2D manifold of the surface it was sketched on, such that a change in the underlying layers representation can be made without having to redraw or manually reposition all the features that exist on that layer. Deposits are created by a procedure that distributes material from the point where the river meets the sea. The material is distributed by a volume preserving diffusion algorithm that considers the topology of the underlying layer surface to create a plausible flow of material from the river.

For changing input a simple oversketching procedure is used. This works by letting the user change parts of an already defined curve by drawing that part over. This new part is then inserted into the curve at the nearest points, and the new curve is smoothed.

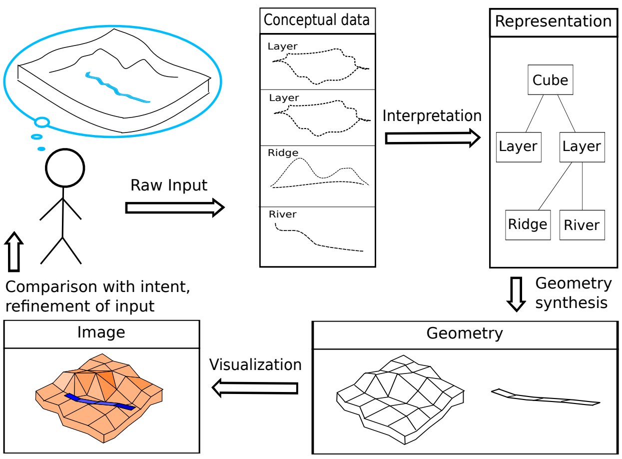

Figure 3: Conceptual overview.

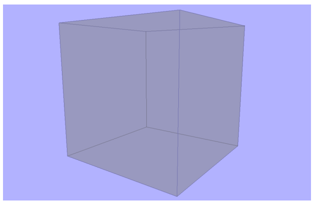

Figure 4: The initial state is the empty cube.

A conceptual overview of the approach is illustrated in Figure 3. The arrows represent processes, either in the computer or performed by the user. The rectangles represent a form of data. The user starts with an idea in her mind of a scenario to model. Through input using the mouse, she then indicates the different features of this scenario. The raw input date goes trough an initial interpretation resulting in the conceptual data. The program interprets the conceptual data, and for each feature recognized, creates a representation of it in the scene graph. The representation is then used by the geometry synthesis code, to create new geometry and alter the shape of existing geometry. This procedure is executed at interactive frame rates. Once the scene geometry is ready, it is used by the visualization code for creating an image that is given back to the user on the computer display. The user then compares what she sees with what she had in mind. She can then perform further refinement of the model by either changing some of what she already drew, or adding new modifications by drawing on the existing geometry.

The initial state for input is the empty cube (Figure 4). At this stage the input consists of the user rotating the camera around the cube and drawing on the cube to create layers. The users input is projected from the screen space coordinates onto the geometrical model (see Figure 5).There is a structure for each object in the scene that contains all the triangles that it consists of. Each of the vertices of the triangles are stored together with a two points that serve as the parameters that uniquely represent the point on the 2D manifold of the object.

When drawing on a screen you are limited to the resolution of the screen. This means that the input points that are gathered will also be limited to this resolution However, because the actual surface where you are interested in drawing exists in a point in space farther away and not on screen, moving from one pixel to the next, means you will move a much greater distance on that surface than on screen, creating jaggedness. The input is smoothed by regarding the n points of the input as the control points of a n-dimensional Bezier curve. The Bezier curve will approximate the control points, but will lie somewhere between them. Most of the points will lie on either side of the intended line, while a Bezier curve will lie somewhere

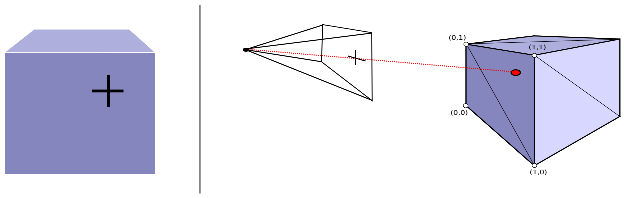

Figure 5: Illustration of intersection. Left: the view of the user with a black cross representing the mouse cursor. Right: The point of the mouse cursor is projected onto the objects by creating a ray from the camera through the cursor and checking for intersections with objects in the scene. The numbers indicate the parametric space values of the vertices of the surface.

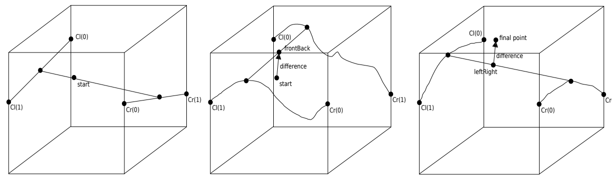

Figure 6: The calculation of a point in the layer grid. First, find the starting point by interpolating the four corners for the current position in the grid. Second, interpolate the front and back curves at the current position, and calculate the difference from the starting point. Third, interpolate the left and right curve at the current position, and finally add the previously calculated difference to this point, yielding the final point.

between.

The different features that can be drawn are represented in an internal representation before creating the structure that can be visualized. The representation is also visualized to the user, so she can make changes. Most of the features that can be drawn are built by using the curves the user drew in different combinations and using different interpretations. In many cases the curves are augmented by additional information, such as the height of ridges. With the deposits however, the representation does not include lines at all but the shape is rather defined by a procedural method.

All the features relate to each other in a child-parent relationship creating a tree structure. The cube is the top node in this tree. All layers are children of the cube. All the other features are then the children of a layer. This structure together with the parametric representation is useful to enable incremental refinement of features, meaning that any part of the whole structure can be modified at any time, without the user having to redraw every part that relates to that change.

Features have algorithms for creating geometry whenever the representation changes. These algorithms create triangles that are drawn on the screen by simple OpenGl functions. In order to achieve the transparency effect, features with geometry must be drawn in the correct order. Layers are drawn first, sketched curves second, rivers third, the sea fourth, and the cube last. This ensures that transparent objects are drawn last and from back to front.

The cubes geometry is generated based on width, depth and height. It is created by six surfaces, representing the faces of the cube. The user sketches input for layers on the front, back, left, and right hand surfaces. A suggestion algorithm will add lines on the other faces, so that minimal input is needed if the user is satisfied with the suggestion. If further changes are made another algorithm makes sure the four curves are always aligning at the corner points, by modifying any previously drawn lines to align with the new one. The cube will also maintain a hidden set of curves that represent the top of all previously drawn layers. This eases the creation of new layers.

A layer gets these four sketched curves as input, plus the hidden curves. The top horizon of the layer is created by a custom interpolation algorithm (see Figure 6). The algorithm starts by doing a bilinear interpolation of the four corner values at the point of consideration. Then a linear interpolation of the two points of the front and back curves currently being considered. The difference, $Diff$, between the points of these first two interpolations is then calculated. Another linear interpolation is done between the two points of the left and right curves. To this last point $Diff$ is added. The effect of this algorithm is that of a profile being dragged across the left and right curves while the profile is being interpolated between the front and back curves. The result is the same no matter if viewed as if dragging the front and back interpolated curves across the left and right curve or vice versa.

If a new layer overlaps with a previous one, only the volume above the preexisting layers is defined. This is done by drawing the surfaces with z-buffering enabled. For the side geometry of the layers, between the new set of curves and the hidden set of curves that represent the top of previous layers, polygons delimited by these curves are drawn only if they are above the previous layers.

Layers can also be changed by editing the curves that were drawn. This happens by oversketching the lines, either in their entirety or partially, thus modifying the shape of the curve. When the layer has changed, it will trigger a recalculation of its children's geometry and layers that were drawn on top of it. The Layers geometry will also need to update when child features are added or changed. Because the child feature is always defined by the 2D point in the manifold of the parent, it is easy to make changes incrementally to layers.

Rivers can be drawn on top of the surfaces by indicating its path with a curve. An algorithm then creates a river as shown in Figure 7. The initial curve is interpreted as the center of the riverbed. Each of the two sides is then computed by extending a new point in both directions from the center point along the rivers path. At the ends of the river a logarithmic function is used to create a smooth falloff towards zero, to make the two sides meet, in order to simplify visualization. The initial line is discarded. The two sides of the river become the representation of the river, and they are the lines that can now be further modified by

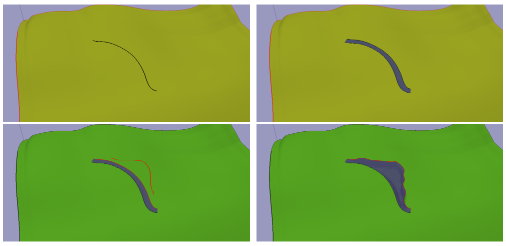

Figure 7: Sketching of rivers by indicating where it should run (top), and oversketching of the sides (bottom).

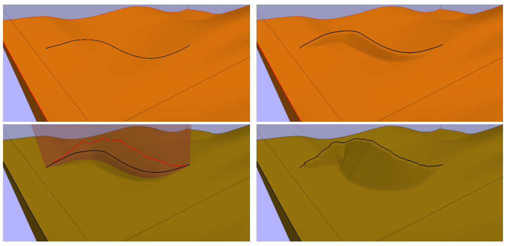

Figure 8: Sketching of a ridge by indicating where its base goes (top), and changing it by sketching the height (bottom).

the user by oversketching.

The oversketching is done on one side of the river at a time. The user also has the choice of replacing the entire side of the river if that will let her more easily make the changes she wants. When oversketching or changing the sides of the river, the layer geometry as it was before the river made any changes is used for the intersection tests. This is because it gets difficult to draw a new side of the river outline on the surface, if that side goes inside the river itself, as the terrain in that area is deformed by the river.

A valley functions almost identical to a river, only it does not create a geometry for any water and is initially wider and deeper than the river. It is, like the river, made by first drawing a line and then it can be changed by the same mechanism as the river. We will therefore not go into more details about the valleys.

Ridges are also drawn by a line on the layer surface. Once a line has been drawn and the user indicates that she wants a ridge, a generic shape of a ridge is created automatically as seen in Figure 8. The user then has the choice to change the height profile along the ridge's baseline. This is done by sketching on a temporary sketching surface that is constructed along the ridges baseline.

Input for the ridges is first drawn on a layer as a curve. A ridge is represented by this curve and a height associated with each point in the curve. The curve is the base line that the user drew on the layer where she wants the ridge to follow along. The heights are the height of the ridge at each point of the curve. Initially the height list is just a smooth function from side to side of the ridge, with a peak in the middle. The height can be changed if the user indicates so. This new height line is input on a temporary sketch wall constructed for this purpose. The input procedure is similar to other lines, but in the end it is not actually stored as regular curve. When the user is done inputing the height line, the height along the entire wall is stored in a list, one for each point on the base line.

The ridge object itself is visualized only by a contour along the top of the ridge. This is constructed by iterating along the points of the base line. For each point in the base line, the corresponding 3D point is found by looking up this point on the layer the ridge belongs to, that is its parent. Afterwards, the height of this point is simply increased based on the relevant height in the list. This yields a new list of points which can then be used to draw a line on screen. When the height of the ridge is being changed, the sketch wall is also shown. The sketch wall is transparent to let the user see other structures that lie behind, so that it is easier to judge how high to sketch.

The sea level can be enabled at any time and moved up or down as the user specifies. In addition to being used in scenes to illustrate the sea, the sea level is used as an input parameter for creating deposits. The sea level is implemented simply by creating a layer with straight outline curves. Each time any layer changes the sea level layer is recomputed. The input is made on the cube by dragging it up and down with the mouse until satisfied. It is visualized with a transparent blue color.

Sedimentary depositions can thus be modeled where a river meets the sea. The user indicates which river is to start depositing, and the rest is done procedurally by a simple simulation. The procedure continues until the user stops it. The user can indicate more than one deposit to be made for a single river. This will make the deposits build outward on top of each other in the direction of the river while also following the terrain.

In order to visualize the creation of a deposit as it builds over time, it needs an additional step to generate an intermediate representation of the deposit before generating the geometry. This step consists of simulating the flow of matter across the surface underneath. For the simulation a simple volume preserving diffusion algorithm is used, that is a modified version of the one by Boeschs [6]. An illustration of the approach by Boesch is given in 9.

The diffusion algorithm assumes a regular height grid, and all the underlying layers must be taken into account. The layers are represented as a irregular grid and thus a sampling must be performed for each cell in the grid. A cell represents the height of the terrain and the amount of deposit in one area of the grid. For each cell in the grid, a ray is cast directly down into the cube, doing intersection tests for each layer, updating the cells terrain height. The algorithm works by considering one cell at a time, and comparing the heights of the neighboring cells height from the previous iteration. Half of the difference, clamped by the available amount of water, will be added to the current cell. This is first done for each cell considering its neighbors along the x-axis. Then the process is repeated, but this time considering the cells neighbors along the y-axis.

In my version of the algorithm, even less than half of the difference is added according to how far from

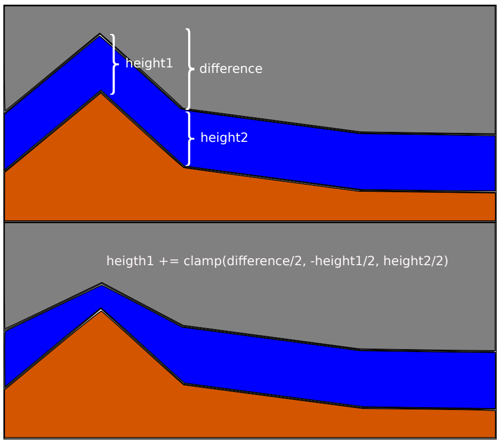

Figure 9: Recreation of figure from approach by Boesch [6].

the rivers mouth the cell is. This modification of the flow-rate is introduced to simulate the loss of energy of particles and eventually making them deposit. The distance is calculated by updating the shortest flow path from the mouth of the river to the current cell at each update, and storing it in a separate grid. This distance value is then used to modify the flow rate by changing the divisor so the final height modifying function becomes: $clamp(difference/(2 + distance^2), -height1/(2 + distance^2), height2/(2+distance^2))$higher than the current value. After the grid heights are found the simulation begins.

The simulation runs until the user is satisfied and stops it or if the deposit is a preexisting one, until the target deposit amount has been reached. The total amount of deposited matter is stored in the target variable when the user stops the simulation. Geometry is generated based on the height of the deposits and underlying terrain. When generating geometry special care needs to be given to the orientation of the triangles to give a uniform and smooth look to the visualization.

To decide where to add triangles, and in which orientation, for the space in between each of the grid cells, the four surrounding points are considered. If both the lower right and upper left cell has deposited material, then triangles will be created between these two cells and one triangle for the two other cells if they have material deposited to. Otherwise, if both the upper right and lower left cell has material deposited, then triangles are created using these two points and creating one triangle extending to each of the other two points as illustrated in Figure 10.

When creating a deposit, the layer object will check for previous deposits, and if such exists, the grid data will be reused for the next deposit. It is reused by copying the terrain height grid and then adding the height of deposits at the points of the grid. This gives speed improvement, and also enables deposits to stack on top of each other.

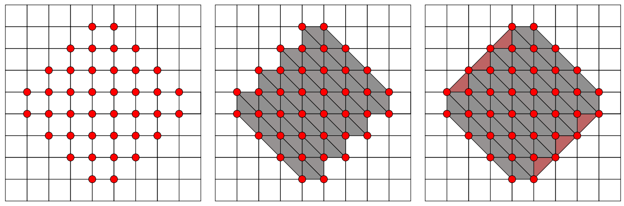

Figure 10: Triangle orientation. Left; the grid points that have matter deposits above the threshold. Middle; the first, nave approach where all triangles are oriented the same direction. Right; the more sophisticated approach, where the triangle orientation depends on which surrounding points have deposited matter. As seen on this illustration, the second approach gives a more uniform look on each side of the structure, while the first approach gives more jagged edges and non-uniform look.

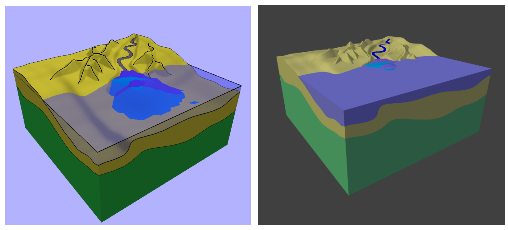

Figure 11: Imaginary terrain sketched, and a rendering made by ray-tracing the exported geometry.

4 Results

Figure 11 shows the same scene as reproduced by the author of the program along with a ray-traced image made in the program Blender. This sketch is made by drawing two layers and setting a color for them. Then mountain ridges are added. A valley is created between the ridges, and in it a rivers path is sketched. After the sea level has been indicated, deposits are created at the point where the river meets the sea.

Figure 12 shows the process of how a glacier erodes the landscape. The first sketch is made by drawing the outline of the rock with valley from the first illustration, then simply adding ridges and rivers. The second sketch is then made by editing the layer to create the valley that is carved by the glacier and deleting the rivers. Then the glacier is created by drawing the end lines on the front and back of the glacier as the contours of a new layer. On the other two sides, the slope of the glacier is indicated. The third sketch is then made by deleting the glacier layer, adding new rivers and valleys, setting the sea level, and creating a small deposit.

Figure 13 shows the process of how the oceanic part of a plate is submerged underneath a continent on another plate where they collide. The structures have to be build from bottom to top, or rather any layer that intersects another must be drawn last. The mantle must be drawn even though it does not appear in the sketch. It is shown in gray here for illustration, but it could be made invisible. Then the oceanic lithosphere is drawn, the oceanic crust follows.

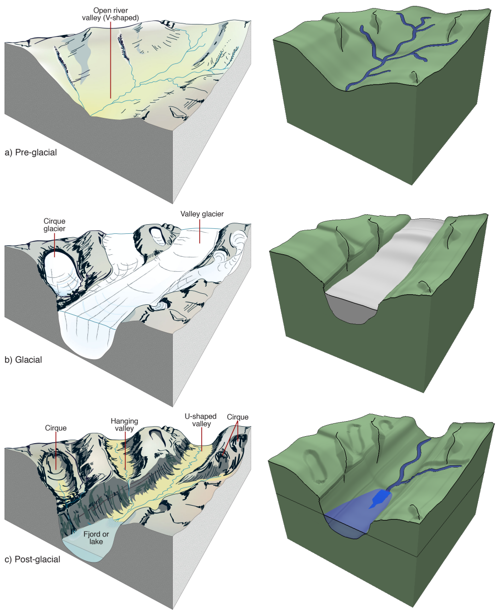

Figure 12: Glacier illustration and attempt at reproduction.

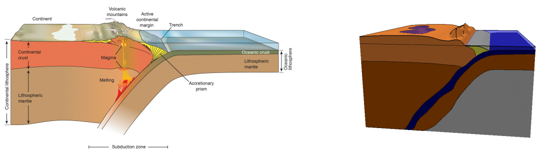

Figure 13: Illustration of subduction of oceanic lithosphere underneath continental lithosphere and attempt at reproduction.

Then the continental lithosphere and the continental crust is drawn such that the sketched curves intersect the oceanic crust at the subduction point. The last layer is the accretionary prism, which represents sediments scraped of from the subducting oceanic plate and gathered at the wedge between the plates. Finally, the sea level is indicated, ridges of mountains are added and an inland sea is created by drawing a river and widening it. The melting rock, rising magma and other volcanic features can not be modeled at this point.

4.1 User study

Evaluating the usability of a modeling approach is not easy. The study was conducted to gather feedback from four geology students. The study consisted of the users giving ratings of different aspects and features of the approach. The answers had to be given by indicating on a scale from 1 (worst) to 10 (best) according to how much the user liked or disliked a feature. Average scores for features and aspects of the approach can be seen in Table 1. In addition the subjects were given the opportunity to explain their choices and give comments for each aspect. A summary of what the subjects responded follows.

Aspect rated	Average
User experience	7
Ease of learning and using	6.5
Potential of approach	8.25
Ease of making changes	6.25
Concept of cube as starting point	9.5
Drawing of layers	8.5
Drawing of ridges	6.75
Drawing of rivers/valleys	7.5
Creation of deposits	7

Table 1: Avarage scores given in user study

The subjects found the tool useful for making illustrations and found the look pleasing. However, the menu items were reported to be confusing. The ease of use was praised as the most significant advantage of the approach. The approach was described to give the ability to play with the different ideas and thoughts around geological scenarios. One user said the approach gives the possibility to create simple illustrations quickly and easily compared to other methods. The approach was described as easy to understand, and in particular one user was impressed with the ability to make changes to the illustration after having made a basic version of the scene.

There was also a suggestion to make a list for selecting the different objects in the scene, as that could sometimes pose difficulties. One subject said that the tool can in its current form be useful for illustrating simple geological scenes, but that it would require more development for it to be useful for depicting more complex scenes. Suggestions for improvements was a feature for creating faults, the ability to alter the width of mountains and the depth of valleys.

One subject indicated a belief that if the features suggested could be implemented, the program could become useful for geology students. Another said that the way they make illustrations today is usually by hand, which is time consuming. This approach helps creating quick illustrations. One user remarked that it was easy to make changes to layers, and that the methods of changing all the terrain features and rivers was excellent. This user also said it was important to not complicate this too much, since the strength of the approach lies in its simplicity.

5 Conclusion & Further Work

A goal was stated in the introduction: to create an approach for rapid and easy sketching of geologic structures in 3D. The approach explained is mostly based on sketch input, but also incorporates a procedural method to explore the possibility of combination of these two rapid modeling metaphors.

Result screenshots and user study shows that such a tool is indeed highly interesting for creating geological sketches. The subjects of the study indicated their belief that the approach has potential. The input methods were described as easy to use, giving improvements in speed over sketching on paper, particularly as it allows incremental changes. Even in the state the implemented solution is in now, users expressed a possibility for applying the solution in educational settings, although further research has the potential to increase the number of possibilities in the approach.

Particularly fault structures are a feature that occurs in many geological illustrations and would be the natural next step we would like to research. Giving the possibility to extend the sketch into multiple cubes would allow bigger scenarios to be sketched. Also, improvements that give the user more control over ridge and river width, and similar would be a good improvement. When developing new features, focus should be given to what potential users are comfortable with and to preserving the ease of use that the approach already has, as this was indicated by users to be the strength of the approach. If implementing any new features it is therefore important to focus on the usability and ease of use.

We expect that when this approach or a similar alternative gains maturity, it could become a standard way to illustrate geological phenomena by students, teachers and researchers, based on the initial impressions from illustrated results and user study. Although further research and development is still needed to enable more geological scenarios to be illustrated, the user feedback indicated that the approach is intuitive to use and enables rapid illustration of certain geological scenarios. The goal of the work, to create an approach that can be used for making rapid 3D illustrations for geologic uses, has thus been reached.

References

[1] R. Amorim, E.V. Brazil, D. Patel, and M.C. Sousa. Sketch modeling of seismic horizons from uncertainty. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, pages 1–10. Eurographics Association, 2012.

[2] C.S. Applegate, S.D. Laycock, and A.M. Day. A sketch-based system for highway design. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, pages 55–62. ACM, 2011.

[3] B. Beneš and R. Forsbach. Layered data representation for visual simulation of terrain erosion. In *Computer Graphics, Spring Conference on, 2001.*, pages 80–86. IEEE, 2001.

[4] B. Beneš and R. Forsbach. Visual simulation of hydraulic erosion. *Journal of WSCG*, 10(1):79–86, 2002.

[5] A. Bernhardt, A. Maximo, L. Velho, H. Hnaidi, and M-P. Cani. Real-time terrain modeling using CPU-GPU coupled computation. In *Graphics, Patterns and Images (Sibgrapi), 2011 24th SIBGRAPI Conference on*, pages 64–71. IEEE, 2011.

[6] F. Boesch. WebGL GPU landscaping and erosion. In *http://codeflow.org/entries/2011/nov/10/webgl-gpu-landscaping-and-erosion/*, November 2011.

[7] G. Caumon, P. Collon-Drouaillet, C. Le Carlier de Veslud, S. Viseur, and J. Sausse. Surface-based 3d modeling of geological structures. *Mathematical Geosciences*, 41(8):927–945, 2009.

[8] J.M. Cohen, J.F. Hughes, and R.C. Zeleznik. Harold: A world made of drawings. In *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 83–90. ACM, 2000.

[9] G.J.P. de Carpentier and R. Bidarra. Interactive GPU-based procedural heightfield brushes. In *Proceedings of the 4th International Conference on Foundations of Digital Games*, pages 55–62. ACM, 2009.

[10] G. Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2001.

[11] H. Fossen. *Geologi, Stein, mineraler, fossiler og olje*. Fagbokforlaget, 2008.

[12] J. Gain, P. Marais, and W. Strasser. Terrain sketching. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, I3D '09, pages 31–38, New York, NY, USA, 2009. ACM.

[13] H. Hnaidi, E. Guérin, S. Akkouche, A. Peytavie, and E. Galin. Feature based terrain generation using diffusion equation. In *Computer Graphics Forum*, volume 29, pages 2179–2186. Wiley Online Library, 2010.

[14] B. Kelk and K. Challen. *Experiments with a CAD system for spatial modelling of geoscientific data*. 1992.

[15] E.M. Lidal, H. Hauser, and I. Viola. Geological storytelling: graphically exploring and communicating geological sketches. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, pages 11–20. Eurographics Association, 2012.

[16] J.L. Mallet. Gocad: A computer aided design program for geological applications. *Three-dimensional modeling with geoscientific information systems*, 354:123–142, 1992.

[17] T. Möller, E. Haines, and N. Hoffman. *Real-time rendering*. AK Peters Limited, 2008.

[18] M. Natali, Lidal E.M., I. Viola, and D. Patel. Modeling terrains and subsurface geology. In *Eurographics 2013*. Eurographics Association, 2013.

[19] M. Natali, I. Viola, and D. Patel. Rapid visualization of geological concepts.

[20] J. Olsen. Realtime procedural terrain generation. *Department of Mathematics And Computer Science (IMADA)*, 2004.

[21] D. Patel, S. Bruckner, I. Viola, and E.M. Gröller. Seismic volume visualization for horizon extraction. In *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, pages 73–80. IEEE, 2010.

[22] D. Patel, C. Giertsen, J. Thurmond, J. Gjelberg, and E.M. Gröller. The seismic analyzer: Interpreting and illustrating 2d seismic data. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1571–1578, 2008.

[23] A. Peytavie, E. Galin, J. Grosjean, and S. Merillou. Arches: a framework for modeling complex terrains. In *Computer Graphics Forum*, volume 28, pages 457–467. Wiley Online Library, 2009.

[24] A. Peytavie, E. Galin, J. Grosjean, and S. Merillou. Procedural generation of rock piles using aperiodic tiling. In *Computer Graphics Forum*, volume 28, pages 1801–1809. Wiley Online Library, 2009.

[25] Schlumberger Information Solutions (SIS). Petrel seismic interpretation software. In *http://www.slb.com/petrel.aspx*.

[26] O. Št'ava, B. Beneš, M. Brisbin, and J. Křivánek. Interactive terrain modeling using hydraulic erosion. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 201–210. Eurographics Association, 2008.

[27] FP Tasse, J Gain, and P Marais. Enhanced texture-based terrain synthesis on graphics hardware. In *Computer Graphics Forum*, volume 31, pages 1959–1972. Wiley Online Library, 2012.

[28] A.K. Turner. Challenges and trends for geological modelling and visualisation. *Bulletin of Engineering Geology and the Environment*, 65(2):109–127, 2006.

[29] N. Watanabe and T. Igarashi. A sketching interface for terrain modeling. In *ACM SIGGRAPH 2004 Posters*, SIGGRAPH '04, page 73, New York, NY, USA, 2004. ACM.