Rapid Visualization of Geological Concepts

Mattia Natali*, Ivan Viola*†, Daniel Patel†*
*University of Bergen, Norway
†Christian Michelsen Research, Bergen, Norway
E-mail: mattia.natali@uib.no , ivan.viola@uib.no, daniel@cmr.no

***Abstract*—We describe a sketch-based system for constructing an illustrative visualization of the subsurface. An intuitive and rapid modelling tool is defined, which takes as input user's strokes and creates a 3D layer-cake model of the earth. Our tool enables users to quickly express and communicate their ideas directly using a 3D model. For sketching, we have created geometric operators that capture the domain specific modelling requirements. We have devised sketching operators for expressing folding and faulting processes. This makes it possible to produce a large span of scenarios. Moreover, for communicating layer properties such as rock type and grain size, our system allows for associating user defined texture to each layer which can be deformed with a few sketch strokes.**

***Keywords*-illustrative visualization; sketch-based technique; layer-cake model; stratigraphy; structural geology.**

I. INTRODUCTION

Many tools have recently been developed to permit better flow from human visual thoughts to digital representations. One approach in this respect is to use sketch-based methods. They give more freedom than *Computer Aided Design (CAD)* systems in designing a general shape. Another advantage of sketch-based methods is that, during planning, giving only a draft of a model is faster than providing all the shape details. In turn, this increases convergence towards the final structure. Detailed models are often made with tools that can be unintuitive for non-experts. The relatively new techniques of free-form sketching have found many applications: for example in toy generation (*Plushie* [1]), generic object deformation (*Teddy* [2], [3], [4], [5], [6], [7]) and terrain modelling [8], [9], [10].

It is advantageous when 3D digital models can be created without processes which require many hours of work. Procedural methods allow the generation of detailed models by only defining few parameters. But these types of tools lead to an approximation of the model that we want to create due to limited control of the outcome. Procedural methods do not give complete freedom in the construction. They give general global control, but no specific local shape control. A possible alternative is to use sketch-based techniques, replacing the use of parameters with free-form user strokes.

Our goal is to quickly create the 3D illustrative models found in geological text books. Figure 1 shows an example made with our method. The produced model will be a way for geologists to express and explain processes involved in the subsurface. The tool could also accelerate the creation of geological illustrations in text books.

With our tool, the user first draws the boundaries of geological layers on a 2D vertical plane. Then, by sketching two of the most important geological processes, folding and faulting, we are able to illustrate a large amount of situations and behaviours occurring inside the crust of the earth. To further give a context to the model, textures are assigned to each layer and deformed according to the shape of the layer. The user can override the default texture deformation using sketching if it is not representative. The user-defined texture deformation can be used for expressing the processes of erosion and tilting. Figure 2 shows how to define a fold, a fault and a guided texture.

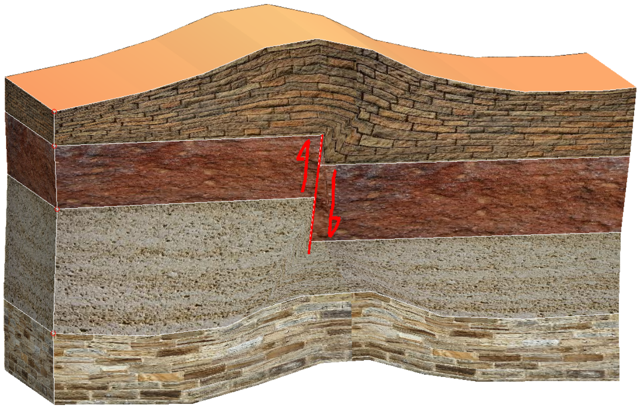


Fig. 1. Layer-cake illustration obtained with our approach.

Our method focuses on simplicity and speed as opposed to time consuming detail-editing of complex models. That is, we develop a method for producing fast qualitative results, since users lack such a tool in the geosciences.

The next section lists motivations for our research. In Section III, we give a short overview on related work on sketch-based modelling, for general modelling and for specific modelling in natural sciences. Section IV describes the type of representation we obtain and states the goal of the tool and what can be designed with it. Furthermore it gives a detailed description of the method. We demonstrate our technique in Section V and give conclusions in Section VI. Finally, we propose possible future work in Section VII.

II. MOTIVATION

Geologists need tools for generating earth surface and subsurface renderings in a rapid way. We introduce a method that is specifically adapted to create such geological models. During the development of our work, we target modelling

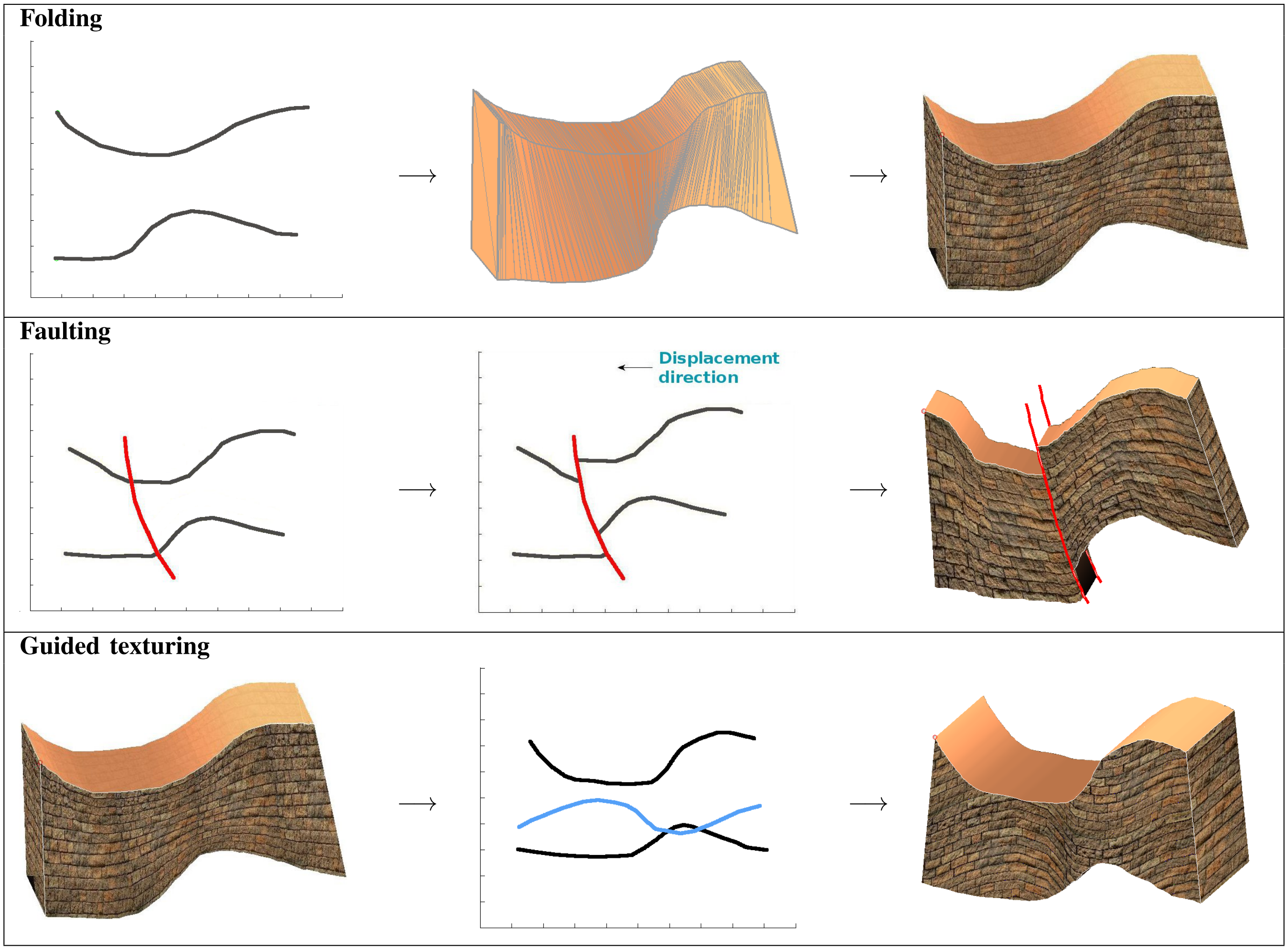


Fig. 2. Global overview: (top) folding definition with just two strokes that generate the model, which is triangulated and textured on the side; (middle) faulting process after having specified the (red) fault and the direction of forces acting on it; (bottom) guided texturing process, from the default adapting texture to the user-defined (with a blue stroke) behaviour of the texture.

problems encountered in geology, thus our method is not meant as a general modelling tool. For the latter purpose, many attempts have already been done. We achieve our scope by introducing geologically relevant sketching techniques, thereby getting an illustrative visualization of subsurface stratigraphy (material layering). The result is a qualitative representation giving a user the possibility to communicate how the earth has behaved or will behave, and of processes that take place. These illustrations make part of a geological illustrator's work, either in subsurface exploration companies, for showing expectations and results, or in text books writing, for visual explanation by expressive examples.

The work in this paper springs from requirements made by geologists, specified during meetings we had with them. They were important in the choice of the geological attributes we have introduced in the sketching toolbox. Several other geological attributes could be introduced in our layer-cake representation as geometric operators that would just need simple sketch-based input from the user. Amongst them we find sketching erosion, channels, salt domes that induce neighbour layers deformation, delta-shape to describe landscape changes due to river flowing, dikes or igneous intrusions. However, in this paper, we have focused on two fundamental phenomena taking place in the earth. They can originate everywhere, are important for interpreting earth movements and can be obtained with a few expressive operators. They are known as the process of *folding* and the process of *faulting*. A fold is obtained when elastic layers of rock are compressed. It is defined as a permanent deformation of an originally flat layer (usually produced by a sedimentation process), that has been bent by forces acting in the crust of the earth. The deformation generating a fold may have different origins: tectonic and convection stress; hydrostatic pressure; pore pressure; high temperature range; salt-, igneous- or sand-intrusion. Folds can have different sizes and occur either isolated or inside a set of deformed layers. Faulting, on the other hand, originates when forces that act on a specific layer are so strong that they overcome the rock's elasticity and yield a fracture.

The novelty of our paper lies in the combination of sketch-based modelling of stratigraphy together with the illustrative representation of geological features. In such a way, models are easy to create and simple to understand such as illustrations in text books, as shown in Figure 3.

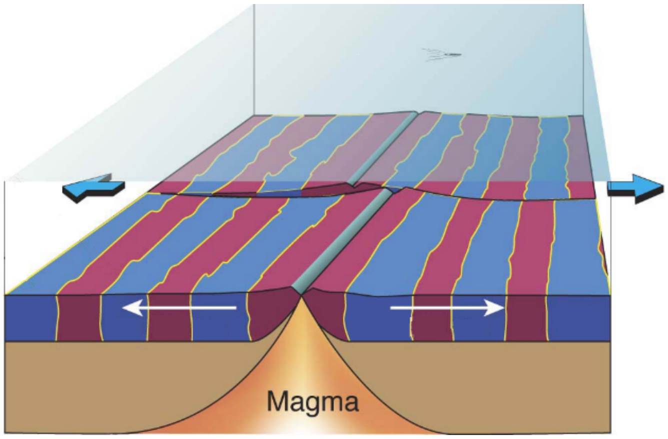


Fig. 3. Example of an illustrative image (courtesy of Haakon Fossen [11]) that can be found in a geological text book.

III. RELATED WORK

As far as we know, no previous works discuss sketching subsurface geological models. Rather, there are some articles dealing with texturing stratigraphic models, like, for example, by Patel et al. [12], by Takayama et al. [13] and by Wang et al. [14]. All of them use volumetric textures for visualizing layered stratigraphy, but none of them focus on sketch-based, fast definition of the models and their appearance.

Recently, many surveys on sketch-based techniques have been made [15], [16], [17]. Some papers, related to *Sketch Based Interfaces and Modeling (SBIM)* have gained much attention such as *Plushie* [1]. *Plushie* introduces a method enabling 3D free-form sketching to create a rounded and smooth object that fits well as a toy prototyping application. Its pipeline consists of specifying two dimensional input strokes which define the silhouette of the object from different points of view. Afterwards, they inflate the silhouettes to a 3D shape and allow the user to add other features directly on the 3D shape. Another method, presented by Brazil et al. [18], takes as input a set of strokes and constructs an implicit surface, interpolating the positions and normals of the samples. They use *Hermite Radial Basis Function (HRBF)* interpolation. The computationally expensive calculation of the implicit surface reduces interactivity when a large number of strokes occurs.

SketchUp [19], is a mature sketching tool, primarily made for architecture. We compared our method with it [20] and we observed that subdividing a solid by drawing folds in a 2D plane is quite similar to our approach. However, *SketchUp* takes more time to reach a final geological model. Moreover, it is not straightforward to generate displacements along partial intersections for modelling blind thrust faults (i.e. when the the top layer is not faulted, such as in Figure 4 and Figure 5). In addition, a single fault can only be generated as a simple straight line that cuts the model [21]. While with our method, a fault is defined on the geological cross-section as a generic sketch together with a definition of the direction of stresses converging to or diverging from the fracture. In addition, free form deformations of textures is not possible in *SketchUp*. We allow such a guided texture deformation on surfaces (for solid texturing, Zhang et al. [22] already proposed their solution in 2010).

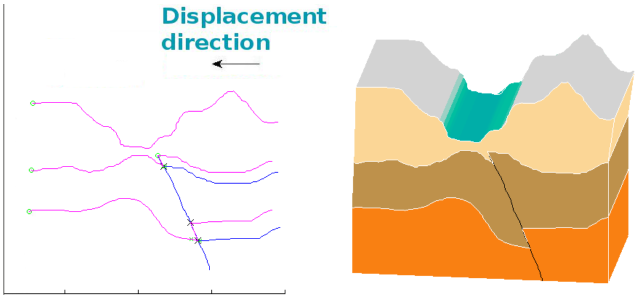


Fig. 4. On the left, three layer boundaries and a fault with its force have been defined. After extrusion, a river has been drawn to obtain the final model to the right.

There is much focus on modelling terrains (one example is given by Peytavie et al.'s paper [23]), but few has been made exploiting the intuitivity of the sketch-based approach. One of the latest works which focuses on creating and deforming landscapes with *SBIM* techniques has been published in 2009 by Gain et al. [8]. They describe a procedural terrain generation tool named *Terrain Sketching*, with the support for sketching mountains and valleys. Their approach overcomes some limitations of previous methods on sketch-based terrain modelling: for instance, Cohen et al. [24] only allow straight areas of influence and boundaries; Watanabe and Igarashi also employ a straight shadow line and they do not give the user the possibility to change the proposed shape. Zhou et al. [25] allow landforms to have a more freely definable shape by using a height-map sketching technique as guidance for a patch-based texture synthesis of a terrain. As opposed to the method by Gain et al. [8], they provide low and indirect control over the height and boundary of the resulting landform, when they choose the type of model as the input example. To reach an intuitive 3D sketch, curves are projected on an existing surface, as opposed to defining strokes from different view-points. None of these methods target the generation of subsurface illustrations as we do.

Following the classification given by Joshi [26], our technique falls in a three dimensional shape modelling category named *curve-based modelling*. Amongst the three further subdivisions of the category, we fit in *extruding 2D shapes* (the other two are: *inflating 2D shapes* and *drawing 3D curves*).

IV. METHODOLOGY

Our aim is a rapid sketching tool for creating illustrative visualizations of structural geology, as seen in geology text books and as used in exploration companies to describe subsurface situations. Structural geology is the study of the

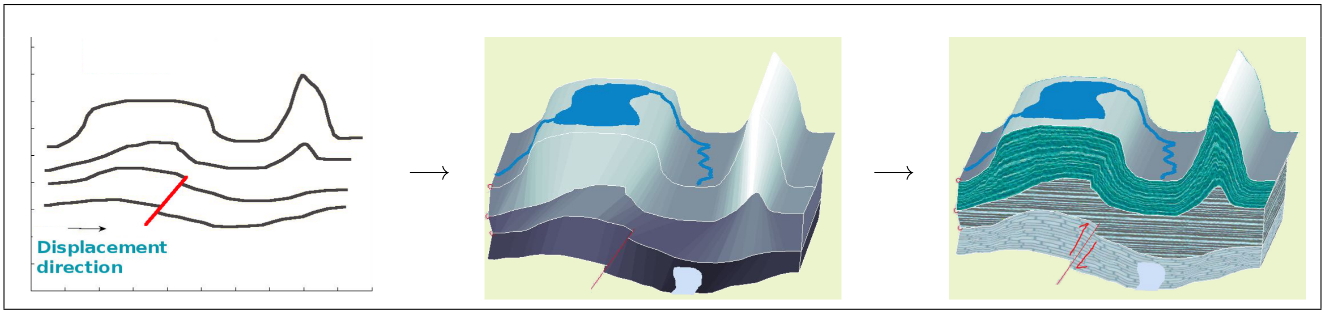


Fig. 5. Complete example of our method, from user's strokes to 3D illustration. Left image shows acquired sketches: black curves define folds and the red curve defines a fault. The model with geometry and projected drawings of a lake and a salt dome intrusion in the bottom layer is shown in the middle image. The final textured model is on the right, with guided texturing adopted for the middle layer.

three-dimensional distribution of rock units with respect to their deformational histories.

Models in structural geology basically consist of stacked layers in a so called layer-cake configuration. Therefore, we construct our model layer by layer. Each layer is represented by its boundary surface, defined by a curve on a 2D cross section. The user simply draws the top and the bottom boundaries of the layer (see for instance the upper part of Figure 2). It is possible to add as many layers as desired. At any time, strokes can be selected and redrawn. Boundary strokes can be folded by selecting one or several of them and defining a deformational stroke. The deformational stroke will vertically offset the selected layer boundaries (see red stroke in Figure 6). After the folded layers are defined, the user can fault them by sketching a fault curve and defining a force (see Figure 7). Each layer can be textured with individual patterns that represent materials such as shale, silt, sand or salt. The textures are automatically deformed according to the folded shape of the layers and are discontinuous over faults. However, the texture deformation can be overridden by the user to represent a particular orientation, deformation or erosional history of a layer. To reach the final visualization in a quick fashion, the user's strokes are acquired on a vertical planar slice (technically called a *"geological section"*) and then extruded into three-dimensional space. After extrusion, the user can draw on the top and side surfaces of the 3D model to add details.

A. Folding

The starting point for creating a sketch, is a blank window representing a geological cross-section. Here, every stroke defines a boundary of a layer. Each stroke is stored as a sequence of line segments. When defining a fold, any kind of stroke that does not self-intersect is allowed.

The user can select one or several boundaries and apply folding (see Figure 6). Folding is achieved by defining a new stroke that deforms the selected boundaries; similar to the approach by Bujans [27]. This new stroke does not have to be of the same length as the boundaries. In the length interval of the folding stroke, the y values of the selected boundaries are displaced according to the y values of the stroke.

Technically, user's strokes are acquired as piece-wise linear approximations of the samples along the stroke. Each planar region enclosed by two consecutive boundaries is triangulated. For this purpose we use a constrained Delaunay triangulation (employing CGAL [28]) to deal with non-convex polygons, where the polygon is defined by the two boundaries of the layer and constraints are given by the edges of the polygon. This constitutes the initial mesh. The mesh will need to be refined when we will later use a conformal map to generate the texture for the layer.

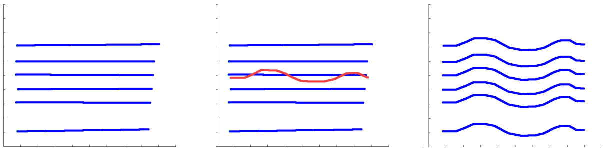


Fig. 6. Fast global folding of horizontal layers by defining a deformational stroke in red that deforms all layer boundaries equally.

If we do not need to add faults in the model, we can extrude the drawn layers and get the first approximation of the structure we have in mind to represent. The fold geometry, represented in an xy plane, is extruded in the z direction. The curves represent surface interfaces of different geological bodies and serve as an initial texture parameterization discussed in Section IV-C. After that, the layer is ready to be visualized. Default colours are automatically applied to each layer when the model is ready.

As an alternative to sketching abstract ideas, the user can trace out folds and fault structures on a background image, trying to represent the real situation observed on the study field.

B. Faulting

To create a fault, the user sketches a curve on the geological section and specifies the direction of forces acting on the fault (referred as "*displacement direction*" in the figures), on either side of the fault. As in nature, if the stress is fault convergent, i.e. it pushes the sides divided by the fault together, the overlying block of the fault (the *hanging wall*, see Figure 7), moves over (*reverse fault*, as in Figure 4) the underlying block of the fault (the *foot wall*, see Figure 7). On the other hand, if the stress is fault divergent, i.e. it pushes the sides apart, the *hanging wall* shifts down (*normal fault*, as in Figure 7 and the example in Figure 8) along the curve representing the fault (see Figure 7). Geometrically, the fault divides each intersected layer in two new layers. Two types of faults are feasible: the so called *thrust fault* (see Figure 7), where the fracture also intersects the top surface, and the *blind thrust fault* (see Figure 4), where the fracture does not reach the top surface.

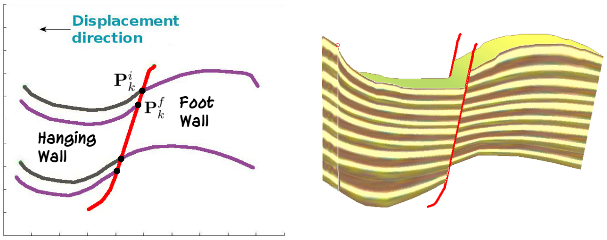


Fig. 7. Example of a textured layer which has undergone folding and faulting.

The system derives from the displacement direction whether the force is pointing towards the fault or away from it. From the trace of the fault, we detect which boundaries are involved and their corresponding intersections. The displacement of layers generated by the fault is proportional to the length of the sketched vector. Depending on the inclination of the fault, the direction of the vector and which side it acts on, we move the point of intersection of each boundary with the fault by shifting it up or down along the fault. Subsequently, we apply the same translation to the entire boundary. Boundaries on the side of the foot wall remain in their position, while boundaries on the side of the hanging wall move of a vector $\mathbf{s}_k$. The displacement vector $\mathbf{s}_k$, related to the k-th boundary, is defined by $\mathbf{s}_k := \mathbf{P}^f_k - \mathbf{P}^i_k$, where $\mathbf{P}^i_k$ is the point of intersection of the k-th boundary with the fault before faulting, whilst $\mathbf{P}^f_k$ is the final position of the point $\mathbf{P}^i_k$ after being moved along the fault trace. The point moves along the fault until the length of its path is equal to the required displacement.

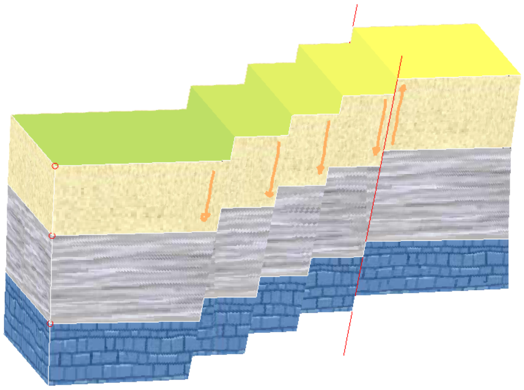


Fig. 8. An example of multiple faults.

When applying a fault on a layer, two cases may occur: either the layer remains connected or the displacement due to the fault completely separates the entire layer in two blocks. In the latter case, we must change the global structure of the model by splitting the layer, originally defined by two boundaries, into four boundaries and keep track of which boundaries belong to the same layer block. Consequently, we have to separate the surface that was representing the unfaulted layer.

C. Texturing

To let each layer represent a type of rock material, we allow the user to place a texture on its visible sides. We let the texture follow the shape of the sketched layer for giving an idea on how the material compresses or deforms under the action of the physical forces (see Figure 9 for our texture application and deformation). The deformation is initially defined by the boundaries of each layer (Figure 9, top), however it can be overridden by the user according to a new sketch (Figure 9, bottom).

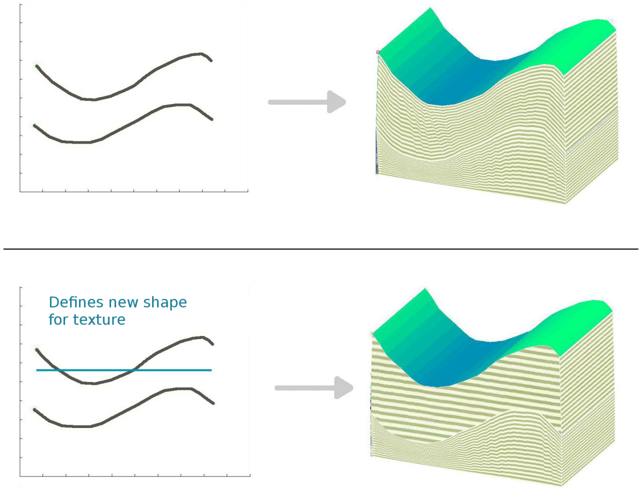
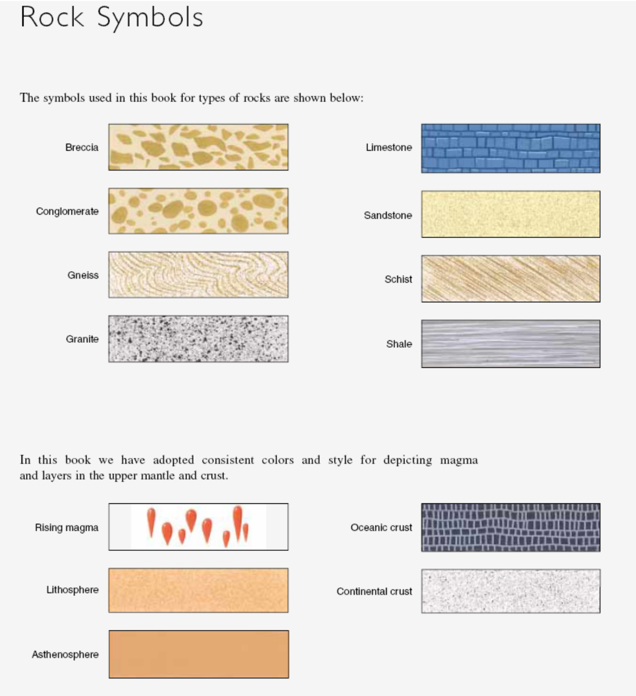


Fig. 9. Layer texturing: default texture shown in the top image, while a new sketch (bottom-left image) defines the modified shape of the texture in the upper layer (bottom-right image).

The textures used in geology are symbolic and communicate different rock types. Figure 10 shows a legend of different rock symbols. When shearing such textures, their symbols might become unrecognizable. To maintain the repeating patterns in textures as recognisable as possible after deformation, we perform an angle preserving (*conformal*) parameterization. A conformal map lets us achieve a more recognizable, robust and aesthetically pleasing result.

Conformal maps preserve angles when mapping from model space to texture space. A conformal map allows the texture to faithfully follow the behaviour of the boundaries and minimizes distortion that may occur during texturing. Prior to the conformal mapping we refine the mesh using one-to-four triangles up-sampling and then we employ a conformal parametrization.

As described by Floater and Hormann [30] and by Mullen et al. [31], to get a conformal map $\mathbf{u} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ (see Figure 11),



Rock Symbols

The symbols used in this book for types of rocks are shown below:

Breccia

Limestone

Conglomerate

Sandstone

Gneiss

Schist

Granite

Shale

In this book we have adopted consistent colors and style for depicting magma and layers in the upper mantle and crust.

Rising magma

Oceanic crust

Lithosphere

Continental crust

Asthenosphere

Fig. 10. Example from a geological book [29] of a legend of different rock symbols, courtesy of Graham R. Thompson.

we have to minimize the *Dirichlet energy*

$$E_D(\mathbf{u}) = \frac{1}{2}\int_{\chi} |\nabla \mathbf{u}|^2 dA,$$

where χ is a differential surface patch and $\nabla \mathbf{u}$ indicates the gradient of the function $\mathbf{u}$. Let $\mathcal{A}(\mathbf{u})$ be the area of the image of $\mathbf{u}$. Since we know that $E_D(\mathbf{u}) \geq \mathcal{A}(\mathbf{u})$ [32], we get a conformal map by imposing the energy $E_C(\mathbf{u}) := E_D(\mathbf{u}) - \mathcal{A}(\mathbf{u})$ to be zero. To calculate this in a discrete environment, which in our case is a triangular mesh, it is necessary to discretize the map $\mathbf{u}$. This has been solved using the *Discrete Conformal Map (DCM)* [33] or the *Least Squares Conformal Map (LSCM)* [34]. For *LSCM*, the aim is to have the gradient of the u coordinate and the gradient of the v coordinate orthogonal and with the same norm. But it is known [31] that *LSCM* is equivalent to minimizing $E_C(\mathbf{u})$ and we use it in our implementation. *LSCM* lets us achieve good behaviours of the texture following the shape of the two boundaries (Figure 1).

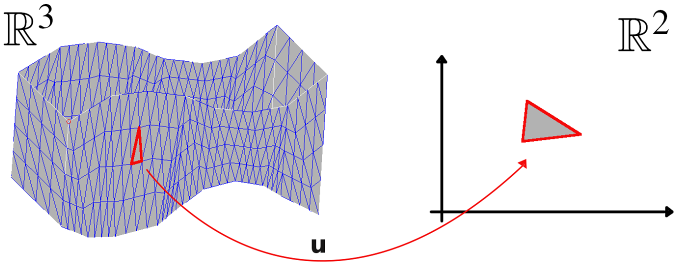


Fig. 11. Conformal map $\mathbf{u}$.

We give the conformal map algorithm these constraints:

- a strip which represents a layer in the model is a closed surface and needs to be opened with a proper cut, for instance along a vertical edge of one of the sides;
- *pinned* vertices [34] are chosen to be the vertices on the boundary of the cut strip. They are mapped to the boundary of the texture accordingly to the piece-wise linear distance along the boundary, from an arbitrary fixed vertex. They do not change during the *LSCM*, but they are used for computing internal vertices;
- internal vertices change their position during *LSCM*, but we maintain their connectivity which we have previously obtained applying one-to-four triangles up-sampling.

Then the conformal map is calculated which satisfies these constraints and guarantees an aesthetic and angle-preserving texture in between.

In presence of faults, the texture image has to be split according to their number and position within the layer. Afterwards *LSCM* is applied to each separate part independently.

Fig. 12. New texture, obtained on the basis of a user's stroke (split in $\mathbf{s}_1$ and $\mathbf{s}_2$). $\mathbf{b}_1$ and $\mathbf{b}_2$ are the top and bottom boundary of the layer and they initially deform the texture, while $\mathbf{s}_1$ and $\mathbf{s}_2$ are copies of the user's stroke that is acquired to guide the behaviour of the texture.

The user has the option to override the default texture constraints, initially taken from the sketched boundaries, as done by Zhang et al. [22] for solid texturing. This is useful for describing different types of erosional situations. For instance, a layer that once was horizontal might have lost material due to a glacier sliding on top of it. This results in the top boundary becoming concave. In this case we want the texture to keep its horizontal shape to communicate the depositional history, and not be affected by the top boundary. Another example is when the user wants to represent a situation where a fold arises from a river erosion as opposed to ground compression. In these cases, the user selects the layer and changes the texturing with a simple sketch (Figure 9 shows how this effect can be reached in a few steps). The new input stroke $\mathbf{s}_1$ is duplicated and set from its copy $\mathbf{s}_2$ on a distance which let the whole layer fit in between. Therefore, if the layer is identified by its two boundary curves $\mathbf{b}_1$ and $\mathbf{b}_2$, then

$$\min\{\mathbf{s}_1\}_y \geq \max\{\mathbf{b}_1\}_y \quad \text{and} \quad \min\{\mathbf{b}_2\}_y \geq \max\{\mathbf{s}_2\}_y.$$

Afterwards, the texture deforms according to the new pair of strokes ($\mathbf{s}_1$, $\mathbf{s}_2$) and the image for the layer is extracted from the inside (see Figure 12). Alternatively, the user can specify two boundaries instead of one. This additionally allows to express compression, as shown in Figure 13.

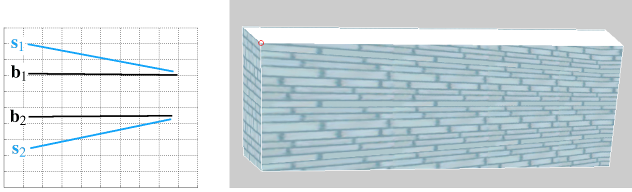


Fig. 13. Changing the default texturing, that would follow the layer boundaries $\mathbf{b}_1$ and $\mathbf{b}_2$, with two new sketches, $\mathbf{s}_1$ and $\mathbf{s}_2$, to convey layer compression.

D. Projected Drawing

Drawings such as those shown in Figure 5 and arrows in Figure 8 are obtained by sketching curves or patches directly on the 3D model. A ray that starts from the viewpoint intersects triangles while the sketch is drawn and the curve is directly projected on the model. The user can choose colours and thickness of the curves. Projected drawing makes it possible to define, for instance, rivers, lakes, magma intrusions, but also explanatory arrows, handmade notes or labels for features of the model.

We perform automatic colouring of the terrain based on its height values.

V. RESULTS

As mentioned earlier, we aim to obtain, from scratch, a layer-cake visualization that is in correspondence to the illustrations used in the text books for explaining geological events in the crust of the earth.

All the models that appear in the figures of this paper have been constructed with very few curves, nevertheless these few input strokes let us represent and express many different situations. In short time (less than a second on an *Intel Xeon E5620 CPU*), the system generates the geometric structure of the layer-cake model. To compute the conformal parameterization, one or two seconds more are needed. The amount of time depends on the number of samples which have been used to acquire and store the boundary curves defining all the layers.

The implementation has been done in Matlab. The code is not optimized and we do not make use of GPU acceleration. For this reason, we believe that the computational time can be further reduced. In any case, with our implementation, a user that wants to represent a model can easily do it in less than a minute, starting from an empty canvas to the final 3D illustrative visualization. This is less than the time a person would employ to draw the same concept on a physical paper with a pencil or on a PC with a painting software (as shown in Table I, which compares required times using our method and a more classical approach). Furthermore, in both cases, one ends up with a 2D picture instead of a 3D model. These are some of the goals that a sketch-based technique aims to achieve. Table I shows an approximate comparison of times that are necessary to create illustrations such as in some of the figures in our paper. The second column contains the required time for a user to sketch curves and choose textures. The third column lists the processing time including time for performing the conformal map. The last column shows the time that would be necessary to create a similar model of the corresponding figure, according to an estimate given by the same geologist and illustrator who produced Figure 3.

TABLE I
APPROXIMATE COMPARISON OF TIMES.

Time	User interaction	Processing	Illustrator's estimation
Figure 1	∼ 20 sec.	∼ 2 min.	∼ full day
Figure 8	∼ 20 sec.	∼ 1 min.	∼ 1-2 hours
Figure 14	∼ 20 sec.	∼ 1 min.	∼ 1 hour
Figure 15	∼ 20 sec.	∼ 1 min.	< 1 hour

Our method allows for creating either illustrative-style (Figure 8, 14) or photo-realistic style (Figure 1) images depending on what class of textures is used. The former is best suited for discussing and brainstorming of scenarios during subsurface exploration and for creating illustrations for pedagogical and presentational use. The latter can be used for describing different outcrop constellations to geology students and for relating to observed field data.

Regarding the use of the default texture application (adapting to the limit boundaries) with respect to the texture that is deformed by a user's sketch, we have examples of the first eventuality in figures 1, 7 and 14. We observe examples of the second case in figures 2 (bottom-right image), 5 (right image, middle layer) and 9, where the layer on which the texture has been re-adapted, following the new shape defined with the input stroke, is easily recognizable. Figure 15 shows how to give to the illustrative model the appearance of being submerged (left image), by simply adding transparency to surfaces of the top layer, and how to better convey a context by adding a few projected drawings on layers (right image).

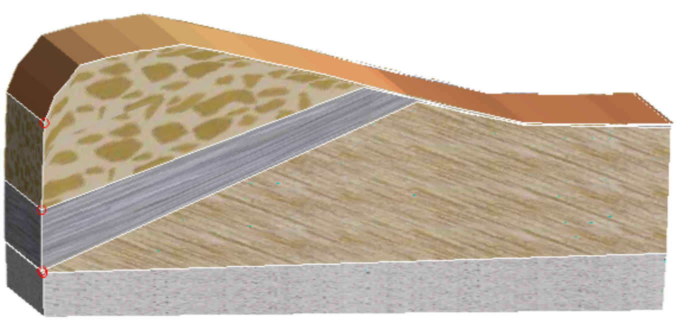


Fig. 14. Example with illustrative textures.

VI. CONCLUSIONS

We have developed a tool for sketch-based modelling, which is specific for the construction of stratigraphic structures of the subsurface with few strokes. The whole process leads to an illustrative 3D visualization (layer-cake representation). We have avoided procedural modelling in favour of getting more control over the generation of the model. All has been done following guidance from professional geologists and illustrators. They found our tool helpful for them in many ways: it solves needs described in Section II (e.g. simple, rapid, illustrative, interactive) and can be a supplement to their current approach to generate illustrations. This is because of interactivity of 3D models and handle-ability of textures on layers.

Such a tool can be a new way for users, in particular geoscientists, who want to share visual thoughts, that originates as abstract ideas, and communicate them through a digital representation.

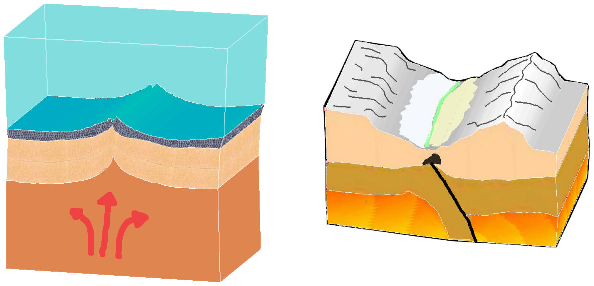


Fig. 15. Example of the oceanic crust with red arrows suggesting movements of the mantle (left) and model with projected drawings that enhance context (right).

VII. FUTURE WORK

An interesting way to face the same problem of generating 3D illustrative stratigraphic representations is to interpret the process going backwards in time. That is, concerning faults, instead of defining the crack and then the displacement of the layers, we start sketching an already existing fault process (perhaps coming from an observation on the field) and we reassemble/put in correspondence blocks of the same rock that belong to the same original layer.

ACKNOWLEDGMENT

The authors would like to thank all reviewers that gave comments and suggestions for improvements, Haakon Fossen for authorization to Figure 3, valuable feedback and expertise and Graham R. Thompson for authorization to Figure 10. This work is funded by the Petromaks program of the Norwegian Research Council through the Geoillustrator project (#200512).

REFERENCES

[1] Y. Mori and T. Igarashi, “Plushie: an interactive design system for plush toys,” in *Proceedings of SIGGRAPH ’07*, 2007, pp. 1–8.

[2] T. Igarashi, S. Matsuoka, and H. Tanaka, “Teddy: a sketching interface for 3D freeform design,” in *Proceedings of SIGGRAPH ’99*, 1999, pp. 409–416.

[3] L. Eggli, C. Hsu, B. D. Brüderlin, and G. Elber, “Inferring 3D models from freehand sketches and constraints,” *Computer-Aided Design*, vol. 29, no. 2, pp. 101 – 112, 1997.

[4] B. R. de Araújo, R. A. Redol, J. Armando, and P. Jorge, “Blobmaker: Free-form modelling with variational implicit surfaces,” in *In Proc. of the 12th Portuguese Computer Graphics Meeting*, 2003, pp. 17–26.

[5] J. J. Cherlin, F. Samavati, M. C. Sousa, and J. A. Jorge, “Sketch-based modeling with few strokes,” in *Proceedings of SCCG ’05*, 2005, pp. 137–145.

[6] L. Olsen and F. F. Samavati, “Image-assisted modeling from sketches,” in *Proceedings of Graphics Interface 2010*, 2010, pp. 225–232.

[7] L. Olsen, F. Samavati, and J. Jorge, “Naturasketch: Modeling from images and natural sketches,” *IEEE Computer Graphics and Applications*, vol. 31, no. 6, pp. 24–34, Nov. 2011.

[8] J. Gain, P. Marais, and W. Straßer, “Terrain sketching,” in *Proceedings of I3D ’09*, 2009, pp. 31–38.

[9] A. Bernhardt, A. Maximo, L. Velho, H. Hnaidi, and M.-P. Cani, “Real-time terrain modeling using CPU-GPU coupled computation,” in *SIBGRAPI ’11*, August 2011, pp. 1–1.

[10] N. Watanabe and T. Igarashi, “A sketching interface for terrain modeling,” in *SIGGRAPH ’04 Posters*, 2004, pp. 73–73.

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

[12] D. Patel, C. Giertsen, J. Thurmond, and M. E. Gröller, “Illustrative rendering of seismic data,” in *Proceedings of Vision Modeling and Visualization ’07*, Nov. 2007, pp. 13–22.

[13] K. Takayama, M. Okabe, T. Ijiri, and T. Igarashi, “Lapped solid textures: filling a model with anisotropic textures,” *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)*, vol. 27, no. 3, pp. 1–9, 2008.

[14] L. Wang, Y. Yu, K. Zhou, and B. Guo, “Multiscale vector volumes,” *ACM Trans. Graph.*, vol. 30, no. 6, pp. 167:1–167:8, Dec. 2011.

[15] L. M. V. Cruz and L. Velho, “A sketch on sketch-based interfaces and modeling,” *SIBGRAPI ’10 Tutorials*, pp. 22–33, 2010.

[16] M. T. Cook and A. Agah, “A survey of sketch-based 3-d modeling techniques,” *Interact. Comput.*, vol. 21, pp. 201–211, July 2009.

[17] L. Olsen, F. Samavati, M. Sousa, and J. Jorge, “Sketch-based modeling: A survey,” *Computers & Graphics*, vol. 33, pp. 85–103, 2009.

[18] E. Vital Brazil, I. Macedo, M. Costa Sousa, L. H. de Figueiredo, and L. Velho, “Sketching variational Hermite-RBF implicits,” in *Proceedings of SBIM ’10*, 2010, pp. 1–8.

[19] *SketchUp*, http://sketchup.google.com/.

[20] I. Viola, “Using *SketchUp* for creating a layer-cake model,” 2011, http://dl.dropbox.com/u/44934338/layercake_screencapture.wmv.

[21] I. Viola, “Using *SketchUp* for editing a layer-cake model and producing a fault,” 2011, http://dl.dropbox.com/u/44934338/fault_screencapture.wmv.

[22] G.-X. Zhang, S.-P. Du, Y.-K. Lai, T. Ni, and S.-M. Hu, “Sketch guided solid texturing,” *Graphical Models*, vol. 73, no. 3, pp. 59–73, 2011.

[23] A. Peytavie, E. Galin, S. Merillou, and J. Grosjean, “Arches: a Framework for Modeling Complex Terrains,” *Computer Graphics Forum (Proceedings of Eurographics)*, vol. 28, no. 2, pp. 457–467, 2009.

[24] J. M. Cohen, J. F. Hughes, and R. C. Zeleznik, “Harold: a world made of drawings,” in *Proceedings of NPAR ’00*, 2000, pp. 83–90.

[25] H. Zhou, J. Sun, G. Turk, and J. M. Rehg, “Terrain synthesis from digital elevation models,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 834–848, July/August 2007.

[26] P. Joshi, “Curve-based shape modeling - a tutorial,” *IEEE Comput. Graph. Appl.*, vol. 31, pp. 18–23, Nov. 2011.

[27] R. A. Bujans, “A thesis on sketch-based techniques for mesh deformation and editing,” 2006.

[28] *Computational Geometry Algorithms Library*, http://www.cgal.org.

[29] G. Thompson and J. Turk, *Introduction to physical geology*, ser. Saunders Golden Sunburst Series. Saunders College Pub., 1998.

[30] M. S. Floater and K. Hormann, “Surface parameterization: a tutorial and survey,” in *Advances in Multiresolution for Geometric Modelling*, 2005, pp. 157–186.

[31] P. Mullen, Y. Tong, P. Alliez, and M. Desbrun, “Spectral conformal parameterization.” *Comput. Graph. Forum*, vol. 27, no. 5, pp. 1487–1494, 2008.

[32] U. Pinkall and K. Polthier, “Computing discrete minimal surfaces and their conjugates,” *Experimental Mathematics*, vol. 2, pp. 15–36, 1993.

[33] M. Desbrun, M. Meyer, and P. Alliez, “Intrinsic parameterizations of surface meshes.” *Comput. Graph. Forum*, vol. 21, no. 3, pp. 209–218, 2002.

[34] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, “Least squares conformal maps for automatic texture atlas generation,” in *Proceedings of SIGGRAPH ’02*, 2002, pp. 362–371.