

# Modeling Terrains and Subsurface Geology



**Daniel Patel, Mattia Natali, Endre M. Lidal, Julius Parulek, Emilio Vital Brazil, and Ivan Viola**

**Abstract** The process of creating terrain and landscape models is important in a variety of computer graphics and visualization applications, from films and computer games, via flight simulators and landscape planning, to scientific visualization and subsurface modelling. Interestingly, the modelling techniques used in this large range of application areas have started to merge in the last years. This chapter is a report where we present two taxonomies of different modelling methods. Firstly we present a data oriented taxonomy, where we divide modelling into three different scenarios: the *data-free*, the *sparse-data* and the *dense-data* scenario. Then we present a workflow oriented taxonomy, where we divide modelling into the separate stages necessary for creating a geological model. We start the report by showing that the new trends in geological modelling are approaching the modelling methods that have been developed in computer graphics. We then introduce the process of geological modelling followed by our two taxonomies with descriptions and comparisons of selected methods. Finally, we discuss the challenges and trends in geological modelling.

---

D. Patel (✉)

Rapid Geology AS and Western Norway University of Applied Sciences (HVL), Bergen, Norway  
e-mail: [daniel.patel@hvl.no](mailto:daniel.patel@hvl.no)

M. Natali

Norwegian Institute of Bioeconomy Research (NIBIO), Aas, Norway  
e-mail: [mattia.natali@nibio.no](mailto:mattia.natali@nibio.no)

E. M. Lidal

Ulriken Consulting, Bergen, Norway

J. Parulek

Equinor, Bergen, Norway  
e-mail: [jparu@equinor.com](mailto:jparu@equinor.com)

E. V. Brazil

IBM Research, Rio de Janeiro, Brazil  
e-mail: [evital@br.ibm.com](mailto:evital@br.ibm.com)

I. Viola

King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia  
e-mail: [ivan.viola@kaust.edu.sa](mailto:ivan.viola@kaust.edu.sa)

© Springer Nature Switzerland AG 2021

D. Patel (ed.), *Interactive Data Processing and 3D Visualization of the Solid Earth*,  
[https://doi.org/10.1007/978-3-030-90716-7\\_1](https://doi.org/10.1007/978-3-030-90716-7_1)

# 1 Introduction

Realistic appearance of natural sceneries has been a key topic in computer graphics for many years. The outcome of this research primarily targets the film and gaming industries. The modelling is often procedural and can be constructed with little user intervention, at interactive framerates. In most cases, only the top surface is the final product of the modelling process, even if subsurface features have been taken into account during the modelling.

Parallel to this development, the modelling of geological structures has been developed from the geological domain. This modelling process usually requires heavy user involvement and substantial domain knowledge. The model creation can often take up to one year of intensive work. The modelling process also includes data acquisition from the site which is to be modelled. The resulting model is usually a very complex 3D structure, consisting of a number of different subsurface structures.

The needs of the entertainment industry and the geoscientific domain are substantially different, although they represent similar natural phenomena. While the former one puts emphasis on interactive realistic visual appearance, the latter one focuses on the correctness from the geoscientific point of view. An analogy can be observed in digital capture of the flow phenomenon: computer graphics proposes visually pleasing methods of flow, while computational fluid dynamics proposes physically plausible methods.

In recent years, research in geosciences has identified the importance of rapid generation of geologic models at early stages in exploration for the purpose of expressing and communicating scenarios where a hydrocarbon trap is present. For rapid generation of models, the extensive development period of a typical geological model becomes a severe limitation. This raised the need for rapid modelling approaches that are common practice in computer graphics terrain modelling. This is one of the reasons why geoscientific modelling techniques are approaching traditional terrain modelling approaches.

One essential difference still remains: While the terrain synthesis for entertaining industries is carried out by artists for the purpose of content creation, the geoscientific models are created based on actual measurements and are done by geologists and other geoscientists based on a substantial level of background knowledge and expertise. Moreover, when modelling based on measurements, the input data are either densely covering a certain spatial area, for instance by means of large-scale acoustic surveys, or consist of sparse samples that are completed by extrapolating known values over the areas where no measurements are taken. There is a multitude of approaches to collect geological information, for instance from seismic surveys (2D, 3D, 4D), boreholes (1D), virtual outcrops from LIDAR scans (3D), or vertical outcrop analysis (2D) [1]. It is outside the scope of this report to provide in-depth information on the acquisition processes. However, the nature of these measurements strongly influences the modelling approach from the geoscientific perspective. The aim of this report is to assess different approaches for modelling geological structures, to compare methods from the computer graphics and geoscience domains and to suggest promising topics for future research agendas.

It is important to note that only a part of the research in geomodelling is of academic origin and publicly available. There is a substantial amount of geological modelling research available only in the form of ready-to-use tools in commercial software packages. This research is, unfortunately for this report, protected by commercial vendors and their algorithmic details are often not known to academia. Therefore we focus on describing those geomodelling methods that have been made publicly available. The aim is not to describe the functionality of commercial modelling packages or to attempt to reverse-engineer the methodology behind them.

We firstly provide, in Sect. 2, a light-weight background on essential properties of geo-bodies as compared to man-manufactured objects typically modelled by means of Computer-Aided Design (CAD). After becoming familiar with essential background knowledge, we discuss our proposed taxonomies and highlight principal differences in Sects. 3 and 4. High-level distinctions between a computer graphics approach and a domain science workflow are reviewed. The aim is to give the interested reader a notion of possible future integrative tendencies between these two fields by means of mutual adaptation of methodologies typical for one or the other domain. Section 5 consists of a comparison of selected techniques for modelling surfaces and for modelling solids. Finally, in Sect. 6 we suggest possible developments of computer graphics in geological applications.

## 2 Geological Elements

The study of structural geology divides the subsurface into geo-bodies [2] of different categories. Central objects are layers, horizons, faults, folds, channels, deltas, salt domes and igneous intrusions.

Much of the reviewed work shows how to represent geological feature such as horizons, folds, faults and deposition. Deposition occurs when eroded particles in nature are brought by wind, water or gravity to a different place, where they accumulate to form a new rock layer. The subsurface is composed of a set of layers with distinct material composition. The surface which delimits two adjacent layers is known as a *horizon*. Two fundamental geological phenomena involve modification of the original structure of horizons: 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 that has been bent by forces acting in the crust of the Earth. Faults originate when forces acting on layers are so strong that they overcome the rock's elasticity and yield a fracture. Horizons are thereby displaced and become discontinuous across a fault. Channels are tubular subsurface structures, they are either fluvial paths that directly originate under the ground or are created when what was once a river has been filled with sediments and then covered with other layers together with the surroundings. Geological models can be divided into two different categories, *layer-based models* and *complex terrain models* [3]. The layer-based models, built of multiple horizontal oriented surfaces, are typically created to model sedimentary geological environments during ground-water mapping,

or oil and gas exploration. In regions with complex geological structures or where the layering is not dominant, for instance when modelling igneous and metamorphic terrains, a more complex terrain model is needed which can be of a probabilistic nature. These complex terrains are modelled when exploring for metal and mineral resources. In this report, we focus on the layer-based models as they are spatially well-defined and share similarities with terrain modelling in computer graphics.

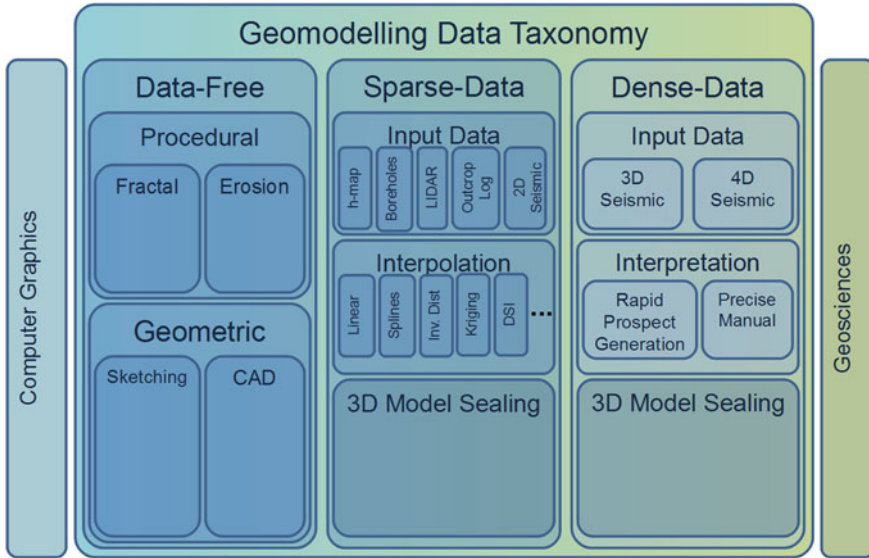
In two viewpoint articles, Turner [3, 4] provides a thorough introduction to the challenges of creating computer tools for modelling and visualization of geological models. The article formulates the essential domain needs and the capability to interactively model and visualize: geometry of rock and time-stratigraphic units; spatial and temporal relationships between geo-objects; variation in internal composition of geo-objects; displacement and distortions by tectonic forces; and the fluid flow through rock units.

Furthermore the following characteristics of the geo-bodies are highlighted: complex geometry and topology, scale dependency and hierarchical relationships, indistinct boundaries defined by complex spatial variations, and the intrinsic property heterogeneity and anisotropy of most subsurface features. These characteristics are, according to Turner, not possible to satisfy with traditional CAD-based modelling tools. Thus, dedicated geological modelling and visualization tools are necessary.

In geological modelling there are often scenarios that lack sufficient data, so in order to build a meaningful model, the creator must interpolate between the sparse sampled or derived data available. Traditional interpolation schemes for discrete signal reconstruction are not sufficient as the process needs to be guided by geological knowledge, often through many iterations, to produce a successful result. A plausible geological scenario has to follow certain geo-physical constraints. Caumon et al. [5] describe specific structural modelling rules for geological surfaces defining boundaries between different lithological layers. Geo-bodies exhibit spatial continuity, therefore abrupt geometric variations such as sudden change of normal orientation on the surface, and abrupt changes within a fault are not common. This implies that a structural model may be validated via reconstructing its depositional state. Caumon et al. also describe the typical process of creating a structural model. The modelling usually starts with fault modelling. The mesh can either be produced directly as triangle strips, based on the dip information or indirectly using a specific interpolation scheme. The second and most important step is to define the connectivity among fault surfaces. The last step is the horizon modelling.

Wellmann and Caumon [6] present a large work from the geoscientists point of view regarding geological modelling. They review geometric representations of subsurface structures with focus on geological realism with respect to observations, information, and knowledge. They also present methods to analyze, quantify, and communicate uncertainties in the models.

In Sect. 3 we will briefly categorize previous works in terms of the type of data that is being addressed (see Fig. 1) followed by an in-depth categorization of papers according to where they fit in the workflow of geomodelling (see Fig. 2) in Sect. 4.



**Fig. 1** Several methods are classified according to which domain they originate from (computer graphics or the geosciences) and what underlying data they use (data-free, sparse or dense data)

### 3 Geomodelling Data Taxonomy

By comparing the outcome of computer graphics terrain modelling with geoscientific modelling, we can roughly divide modelling into three distinct categories: the *data-free*, *sparse-data*, and *dense-data* scenario. The first category represents current and future trends in rapid modelling, where current methodology originates from the computer graphics research, while the latter two categories are developed from explicit needs in the geoscientific domain. This categorization forms one of the high-level taxonomies of the discussed modelling approaches. Figure 1 shows the taxonomy together with different modelling scenarios.

The *data-free* scenario has no ground truth information and therefore the geometric synthesis relies entirely on *procedural* [7, 8] and *geometric* modelling. The typical computer graphics research agenda proposes methodologies that alleviate the user from labor-intensive tasks by automating parts of the modelling. Procedural modelling offers the modeller specific, easy to handle input parameters which control the process of geometry generation. The geometry typically represents terrain surfaces. Procedural techniques in modelling have been facilitated mainly through *fractal* modelling [9, 10]. In the dynamic case, simple *erosion* models [11, 12] are utilized to create dynamic and realistic landscapes [13, 14].

The shortcoming of procedural modelling is usually the lack of direct control over the landscape development. The modeller has a rough idea of the landscape, but implicit parameter settings do not guarantee a match with the modeller’s idea

of an intended shape. Therefore a combination of explicit geometric modelling, to represent the modeller's expectations, with procedural modelling, to add realism, is a preferred strategy. On the other hand, geometric modelling can be a labor-intensive task. For rapid modelling scenarios, various forms of *sketching* metaphors [15–17] or modelling by example [18] provide fast ways to express the rough structure of a terrain or of a stratigraphic model [19, 20].

The *sparse-data* modelling scenario is the most frequently seen in the geoscience domain [21]. Very often it contains networks of boreholes [22–24], where the data needs to be interpolated between. Besides boreholes, there are often other acquisition types available, such as surface elevation models [25, 26], obtained through the process of remote sensing, typically acquired using satellite or aircraft-based sensor technologies. This heterogeneous pool of geoscientific data raises the challenge of data integration and data interpolation.

The main interpolating methods are the Kriging method, the Discrete Smooth Interpolation (DSI) method, the Natural Neighbor Interpolation method, Radial Basis functions, the Inverse Distance method, and spline methods. They will be discussed in Sect. 4.2.3.

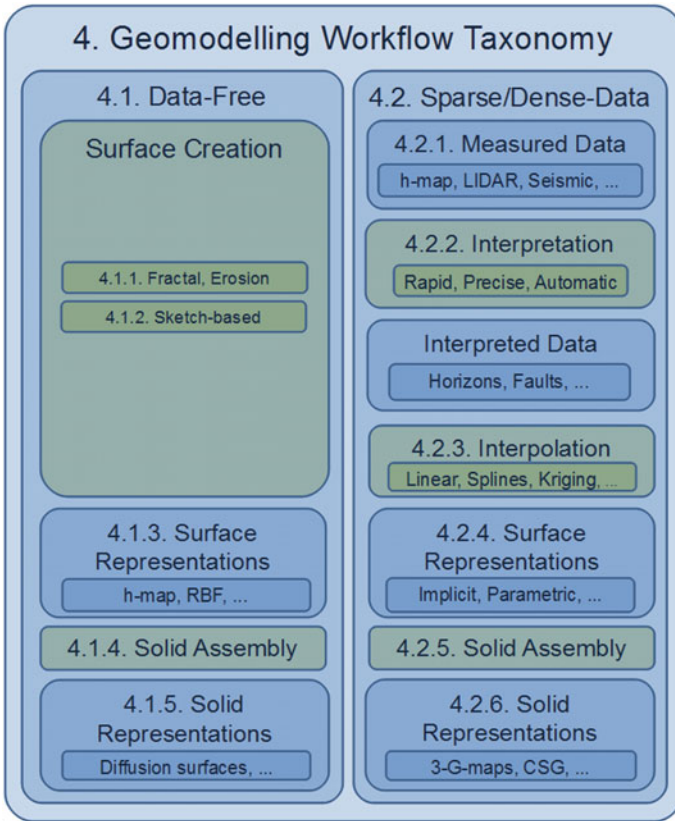
Turner [3] demonstrates how to build a typical geological model from a *sparse-data* scenario by firstly interpreting bore-hole logs to construct triangulated surfaces of horizons [27] and then create the geo-bodies [28, 29] in sealed, boundary-representations [30] of the volumes between these surfaces. The modelling of faults [31] is also very important and Turner describes the challenge of modelling the interface between the boundary representation and the fault to avoid an unwanted crossing or empty spaces between the fault and geo-bodies. Using a structured mesh representation of the boundary surface can result in discretization errors, while using an unstructured grid representation [32] adds computational complexity and results in slow model construction.

The *dense-data* scenario is typically based on a single- or multi-attribute volumetric seismic dataset. The first challenge is purely of computational character, i.e., how to interactively display huge amounts of volumetric data, addressed, e.g., in the work of Plate et al. [33]. Utilizing volume rendering concepts, these datasets can be displayed without prior extraction of geo-bodies. Extracting geological structures from this data is necessary for consecutive steps along the workflow, such as reservoir modelling. This process is known as geoscientific interpretation and is a very time-demanding task. Typically, the original seismic dataset, consisting of the amplitudes of reflected sound waves, can be used to extract a number of derived attributes. These attributes are not geo-bodies, but their distribution over the 3D domain indicates the presence of certain geological structures. The *SHIVR* interpretation system [34] can extract geo-bodies based on scatter plots, as shown by Andersen and van Wijngaarden [35]. Rapid prospect generation can certainly benefit from faster interpretation. Patel et al. proposed methods for rapid horizon extraction in two [36] and three dimensions [37]. Afterwards, once the interpretation is available, 3D visualization can assist in validating the correctness of the extracted horizons with respect to original or derived attribute data [38].

A natural next step after 3D structural modelling is the development of a time-varying structural model. Inverse methods are often utilized in geomodelling to restore hypothetical geological scenario by going backwards in time [39–41]. Such an approach aims at restoration of deposited sedimentary layers, for example through unfolding. Restored information about palaeogeography often gives good indication where to search for hydrocarbon reservoirs.

## 4 Geomodelling Workflow Taxonomy

In Fig. 1, methods are categorized according to which domain they arise from, which is also tightly correlated with the amount of measured geologic data they handle. However, for the rest of the paper we have found it more appropriate to describe methods in the order they would be applied in a workflow for creating a geological model. Such a taxonomy is shown in Fig. 2. We have defined two separate workflows,



**Fig. 2** Workflow taxonomy. Blue boxes represent data and green boxes represent action on the data to create richer data. The smaller boxes inside show examples of methods for that subtopic



one for creating models from no data, and one for creating models when data exists. The steps in each workflow are aligned with each other and we have specified in front of each category which subchapter they are described in. In the workflow where no measured data is required as input (data-free), some papers focus on surface creation, other general papers describe different mathematical surface representations. Several sketch-based papers describe different ways of fast sketching and assembling of solid objects and some focus on compact representations and fast rendering of complex solid objects. For the case where data is being used (the sparse/dense-data column), the workflow begins with measuring data, interpreting relevant structures, interpolating these into higher-order objects and representing these in some appropriate mathematical way. The structures are then assembled into solid geometry describing the subsurface. We have devoted a subsection to each of the topics described in the geomodelling workflow taxonomy.

## 4.1 *Data-Free*

This section describes works that do not rely on any measured or sampled input data, and where the models are created from scratch, driven by imagination or concept ideas and domain knowledge.

### 4.1.1 **Fractal and Erosion Surface Creation**

There are usually three approaches to generate synthetic terrains: fractal landscape modelling, physical erosion simulation and terrain synthesis from images or sample terrain patches. Before the work by Olsen [7], it was mostly possible to use simple fractal noise to obtain terrain surfaces, because computers were not fast enough to simulate erosion processes in real-time. Olsen proposes a synthesized fractal terrain and applies an erosion algorithm to this. His representation of terrains is a two-dimensional heightmap. To simulate erosion, he considers the terrain slope as one of the main parameters: a high slope results in more erosion, a low value produces less erosion. Starting from a noisy surface, called the *base surface*, erosion occurs to simulate weathering on a terrain. He applies two types of erosion algorithms to the base terrain: thermal and hydraulic. After testing the two erosion methods, he decides to combine the advantages of each, namely, the speed of the thermal erosion and the realism of the hydraulic erosion.

The ability to model and render piles of rocks without repetitive patterns is one of the achievements of the work by Peytavie et al. [8]. They focus on rocks and stones, which are found everywhere in landscapes. They provide realism to the scene, reveal characteristics of the environment and hint on its age. Before this paper, the canonical way of generating rocks was to produce a few models by artists, which were then instantiated in the scene. To create piles of rocks, collision detection techniques were applied with a high computational cost and low control. The authors propose



aperiodic tiling of stones to avoid repetitive patterns. Two steps define the method proposed in the paper: the first is a preprocessing step that generates a set of aperiodic tiles constrained to maintain contact between neighbours; subsequently, the rock piles are created by exploiting aperiodic tiles from the previous step. Voronoi cells are employed to control the shape of rocks. In the construction of the Voronoi cells, an anisotropic distance to avoid round-shaped stones is used. Finally, erosion is applied to produce the final model. Meshes of the rocks are represented by using standard implicit surface meshing techniques.

Musgrave et al. [42] describe the creation of fractal terrain models, avoiding global smoothness and symmetry; these two drawbacks arise from the employment of the first definition of fractional Brownian motion (fBm) as introduced by Mandelbrot [43]. Moreover, there is a second stage in which the surface undergoes an approximation of a physical erosion process. Terrain patches are represented as heightmaps and the erosion process is subdivided into a thermal and a hydraulic part.

Concerning modelling terrains with rivers, Sapozhnikov et al. state that at the time when their paper [44] was written it was impossible to simulate the process of natural river network formation without making a substantial approximation; i.e. a simpler model that does not make use of the physical laws, but nevertheless reproduces the main geometrical features of a real river network. They use a random walk method to generate a set of river networks of various sizes.

Stachniak et al. [45] point out that fractal methods have been used to create terrain models, but these techniques do not give much control to the user. They try to overcome this by imposing constraints to the original randomly created model, according to the user's wishes. The method requires two inputs: the initial fractal approximation of the terrain and a function that incorporates the constraints to be satisfied in order to achieve the final shape. As an example, they show how to adapt a fractal terrain to accommodate an S-shaped flat region, representing a road. The constraint function defines a measure that indicates how close a terrain is to the desired shape. The final solution is provided by a minimization of the difference from the current terrain to the desired one.

Another way of combining procedural modelling with user constraints is described by Doran and Parberry [46]. In their work, they procedurally generate terrain elevation heightmaps, taking into consideration input properties defined by the user. The model lets the user choose amongst five terrain tools: coastline, smoothing, beach, mountain and a river tool. Together, these tools can be used to generate various types of landscapes.

A terrain surface is created by fractal noise synthesis in Schneider et al.'s work [9]. They aim to solve the problem that was one of the biggest disadvantages in fractal terrain generation at the time, namely the setting of parameters. They reduce such an unintuitive process of setting parameters by presenting an interactive fractal landscape synthesizer.

Roudier et al. [47] propose a method for terrain evolution in landscape synthesis. Starting from an initial topographic surface, given by a heightmap, they subsequently apply an erosion process to obtain the final 3D model. The erosion consists of mechanical erosion, chemical dissolution and alluvial deposition.

Chiba et al. [48] propose a method that overcomes the limitation of previous techniques for generating realistic terrains through fractal-based algorithms, but lacks ease of handling, i.e. it is not possible to modify the surface on the basis of constraints drawn by the user. The topology of the landscape is created by a quasi-physically based method, that produces erosion by taking velocity fields of water flow into consideration. The whole process of erosion, transportation and deposition is derived on the basis of the velocity field. Dorsey et al. [49] focus on erosion applied to one stone or rock, represented by its volume, taking into consideration weathering effects on it.

In the work by Benes et al. [50], a method for eroding terrains is described. A concise version of a voxel representation is utilized, and thermal weathering is simulated to erode the initial model. This new way to represent terrains has the advantage of being able to represent caves and holes. When applying erosion, all the layers and ceilings of the caves are involved in the process.

In a subsequent paper [51], a technique for procedural modelling of terrains through hydraulic erosion is introduced. The purpose is to use a physically-based approach together with a high level of control. Their algorithm takes in consideration that water dissolves material and transports it to other areas where, after evaporation, it is deposited. Contrary to previous techniques which tend to oscillate during water transportation, they provide a tool for hydraulic erosion that is fast and stable. They overcome oscillation by relying more on physical constraints than was previously the case. The erosion process consists of four independent steps, where each step can run repeatedly and in any order. These four steps are: introduction of new water (simulation of rain); material capture by water (erosion); transportation of material; and deposition at a different location.

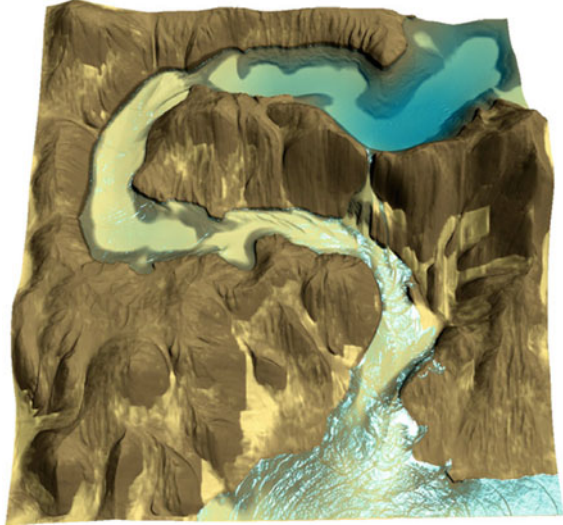
Another work by Benes et al. [52] applies a hydraulic erosion fully based on fluid mechanics and thus on the Navier-Stokes equations that describe the dynamics of their studied models. They use a 3D representation provided by a voxel grid and the erosion process leads to a model that can show a static scene or part of an animation illustrating the terrain morphology. At each iteration of the process of erosion, a solution to the Navier-Stokes equations is computed to determine a pressure and velocity field in the voxels.

Interactive physically-based erosion is employed by Stava et al. [53] (Fig. 3). This work is based on physics by making use of hydraulic erosion, and on interactivity, which allows the user to take an active part during the generation of the terrain. The technique is implemented on the GPU and, because of the limited GPU memory, the terrain is subdivided into tiles, which allow them to apply erosion to local areas. Each terrain-tile is represented as a heightmap.

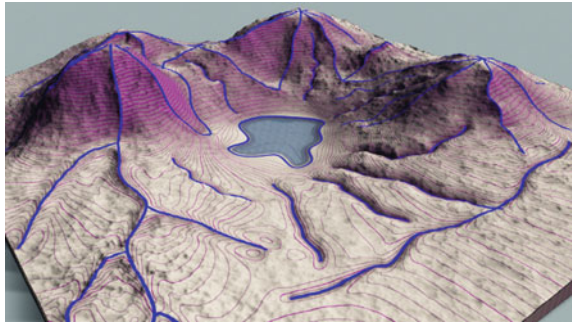
Kristof et al. [11] adopt 3D terrain modelling through hydraulic erosion obtained by fluid simulation using a Lagrangian approach. Smoothed Particle Hydrodynamics (SPH) [54, 55] is employed in this paper to solve dynamics that generate erosion. SPH requires low memory consumption, it acts locally, works for 3D features and is fast enough to work on large terrains.

For Hnaidi et al. [12], the terrain is generated from some initial parametrized curves which express features of the wished terrain (see Fig. 4). Each curve is enriched

**Fig. 3** The eroded terrain is obtained by simulating the movement of the water flow and transportation of rock particles [53]



**Fig. 4** Sketches, that are visible in the figure as blue strokes work as constraints during the method proposed by Hnaidi et al. [12]

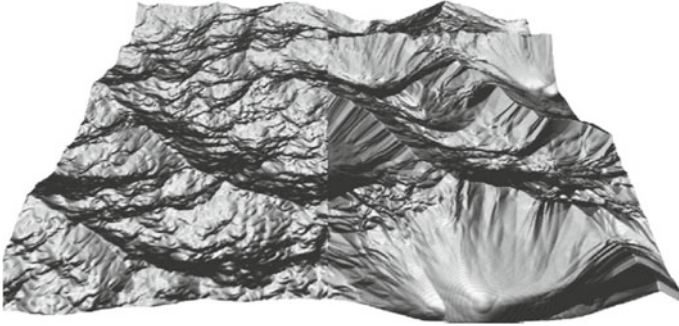


with different types of properties (such as elevation and slope angle) that become constraints during the modelling process.

Prusinkiewicz and Hammel [13] address the problem of generating fractal mountain landscapes, which also includes rivers. They do it by combining a midpoint-displacement method for the generation of mountains with a method to define river paths.

Hudak and Durikovic [14] tackle the problem of simulating terrain erosion over a long time period. They use a particle system and take into consideration that terrain particles can contain water. The Discrete Element Method (DEM) is used for the simulation of the soil material and Smoothed Particle Hydrodynamics (SPH) simulates water particles.

Instead of procedural and erosion synthesis, Brosz et al.’s paper [18] introduces a way to create realistic terrains from reference examples. This process is faster than starting the terrain generation from scratch. Two types of terrains are necessary to obtain the final one: a base terrain, used as a rough estimate of the result, and the



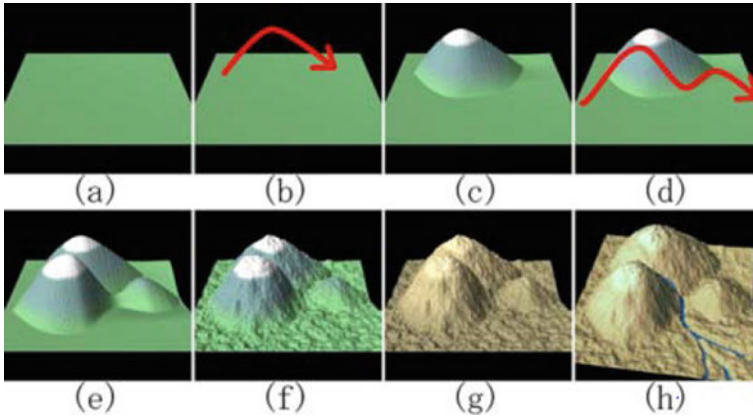
**Fig. 5** Two types of noise shown on the left and right side of the image applied by de Carpentier and Bidarra [56] to achieve a realistic terrain

target terrain that contains small-scale characteristics that the user wants to include in the reconstruction. Brosz et al. have two common ways to generate landscapes represented as a heightmap: using brush operations to bring some predefined information or action on the surface; alternatively, simulation and procedural synthesis can be applied to obtain a realistic terrain. One drawback of using simulation is that it can be slow, while in the case of procedural synthesis, expressability is reduced by a limited set of parameters. De Carpentier and Bidarra try to combine brushing and procedural synthesis in their work [56] (an example is shown in Fig. 5).

Cordonnier et al. [57] use a volumetric representation to create models of terrains and near-surface geology. The method uses layers and sublayers with different physical properties to simulate erosion (improving their former work [58]) and uplift movements of the earth’s crust. The authors embody these techniques in an interface that makes use of hand gestures to create large scale terrains ( $100 \times 100$  km) with a good level of realism. In another work, Cordonnier et al. [59], use a similar method to simulate erosion on multi-layer data but connecting it to combine various ecosystems.

A good source of further reading is Galin et al. [60] who perform an extensive review of the most recent works on terrain modeling from a computer graphics perspective. The authors classify the representation of the terrain into two broad groups, one using heightmaps, and one using volumetric representation. They also categorize over fifty papers according to factors such as:

- The variety of landforms the method can create.
- The realism of the landforms, where the authors used different ways to to measure realism.
- The range of scales of the landforms, which takes into account the precision and extensions—from meters to hundreds of kilometers.
- The authoring of the landforms, which evaluates how easy it is to create the terrain the designer has in mind.
- The efficiency of the algorithm, which takes into account time and space costs.



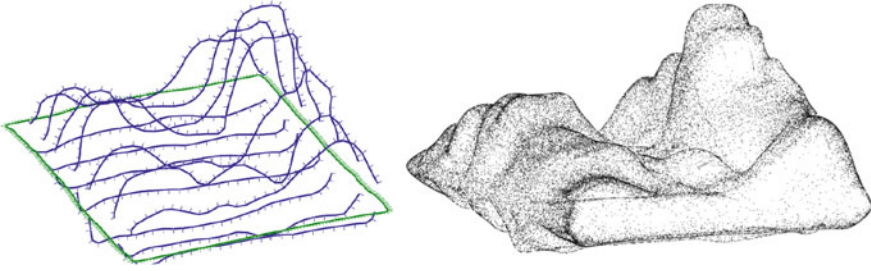
**Fig. 6** Watanabe and Igarashi's process for obtaining a landscape surface by sketching [62]

#### 4.1.2 Sketch-Based Surface Creation

Modelling a terrain surface with sketch-based tools rather than procedural techniques is a more controlled and intuitive process. Gain et al. present a paper [15] that describes procedural terrain generation with a sketching interface. Their approach aims to gather benefits and overcome some limitations of previous methods of sketch-based terrain modelling [61–63]. Watanabe and Igarashi [62] employ straight lines and, even though they yield a boundary for landforms using local minima and maxima of the user's sketch, they do not give the user the possibility to change the proposed shape (see Fig. 6). Furthermore, they apply noise onto the terrain after surface deformation, hence the obtained surface does not interpolate the user strokes exactly. Whereas landforms rarely follow straight lines, Zhou et al. [63] allow landforms to have more free-shape paths using a heightmap sketching technique as guidance for an example-based texture synthesis of terrain. In contrast to the method suggested by Gain et al. [15], they provide low and indirect control over the height and boundary of the resulting landform.

Extending the use of parametric curves to modeling terrains and their properties in a volumetric representation, Becher et al. [64] were able to create terrains with arbitrary vertical layouts, enabling terrain features such as arches, caves and overhanging cliffs. The application makes use of feature curves that define the local terrain properties and extrapolate these properties to the rest of the model area. By performing computations in compute shaders on the GPU, interactivity is achieved.

Guérin et al. [65] use a Conditional Generative Adversarial Network (cGAN) [66] to create terrains from sketches. A Generative Adversarial Network is a deep-learning method that maps images to images and has successfully been used in many applications [67]. In general, the cGAN method receives a pair of images as a training unit—the input image and the expected output image. To generate the terrains, the authors trained four cGANs, each one specialized in one task. The first one takes



**Fig. 7** Landscape example generated with a sketch-based approach by Vital Brazil et al. [17]. The model is represented with Hermite Radial Basis Functions. Sketch input is shown to the left, and result is shown with a stipple rendering style to the right

sketches representing rivers, valleys and ridges along with points with elevation information and creates the first terrain. Then the authors train a second network where the user can specify areas of constant elevation. To be able to complete missing information on the models, the authors train a third network with images of terrains. Finally, they create a network to simulate erosion processes. The first three cGANs were trained using real terrains images, while the last one uses a synthetic data set. After a large scale terrain model has been created, fine procedural details can be added by using the technique of *terrain amplification* [68].

Vital Brazil et al. [17] introduce a sketch-based technique to generate general 3D closed objects using implicit functions. They also show how to exploit their tool to obtain simple geological landscapes from few user strokes as shown in Fig. 7. Further sketch-based techniques that also define subsurface features in addition to the top terrain surface are presented in Sect. 4.1.4 Solid Assembly.

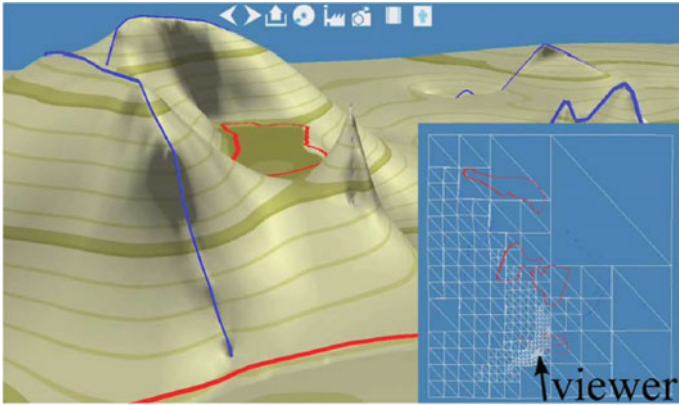
### 4.1.3 Surface Representations

Several of the fractal and erosional surface creation methods represent the surfaces as heightmaps. This is an easy-to-maintain datastructure which fits well with erosional calculations. The method by Vital Brazil et al. [17] can represent complex surfaces with overhangs or closed objects, using implicit functions defined as a sum of radial basis functions. Based on points with normals as input, a smooth implicit function, interpolating the points while being orthogonal to the normals, is created. Further details on this method is found in Sect. 4.2.3 Interpolation which also presents other methods that are closely related to surface representations.

Peytavie et al. [16] represent complex terrains with overhangs, arches and caves. They achieve this by combining a discrete volumetric representation, which stores different kinds of material, with an implicit representation for the modelling and reconstruction of the model.

Bernhardt et al. [69] present a sketch-based modelling tool to build complex and high-resolution terrains, as shown in Fig. 8. They achieve real-time terrain modelling





**Fig. 8** Surface modelling with the sketch-based tool proposed by Bernhardt et al. [69]

by exploiting both CPU and GPU calculations. To represent large terrains, they use an adaptive quadtree data structure which is tessellated on the GPU.

#### 4.1.4 Solid Assembly

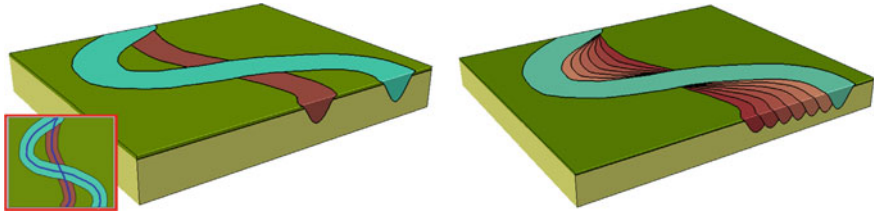
By solid assembly, we refer to the process of assembling boundary surfaces or basic solid building blocks into a complete solid object. This work process is supported by CAD based tools.

As opposed to solid assembly for the sparse/dense data scenario (Sect. 4.2.5), in the data-free scenario described in this section, sketch-based methods are predominantly used both for defining the building blocks and for assembling them into a final model. With sketch-based techniques, the process to shape geological structures becomes intuitive and fast.

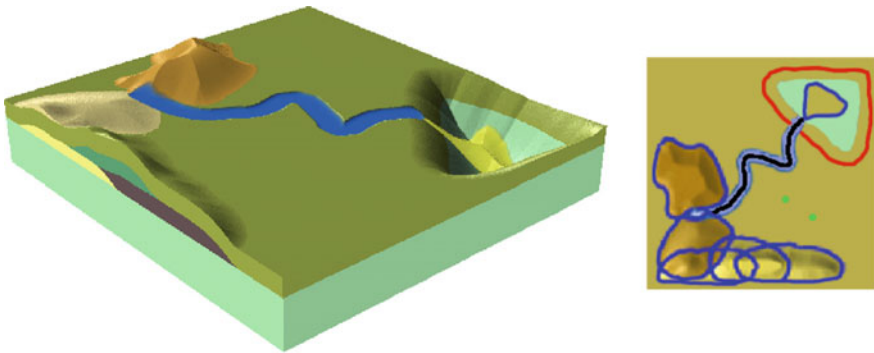
Natali et al. [19] present an example of sketch-based solid assembly of geological layer-cake models. Their approach allows a user to sketch geological layers and faults on a 2D cross section of the model which is then extruded to a 3D model. Texturing is applied on each layer and the texture can be deformed using conformal texture mapping to e.g. follow the shape of the layer. The texturing helps convey information that is not captured in the geometric model such as the material type and material orientation within each layer.

Some of the stratigraphic layers found on earth are formed by deposition and erosion caused by fluvial systems. In 2014, Natali et al. [70] explored a way to provide fluvial systems illustrations to geologists, by means of an interactive 3D modelling approach. The focus is on evolving depositional and erosional landscapes by modelling rivers, deltas, basins, lakes and mountains. For allowing fast sketching, a minimalistic user interface is applied where a user draws on the top surface of the model, either a point, an open curve or a closed curve. A point is interpreted as a





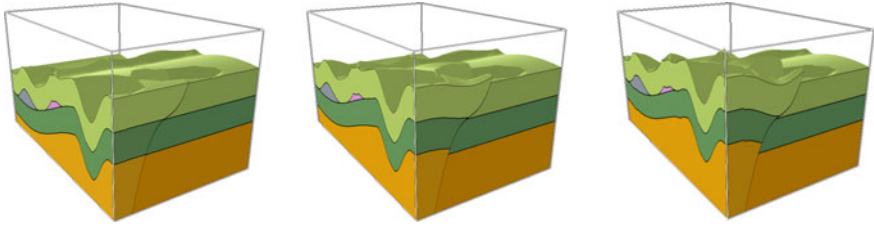
**Fig. 9** Left: first and last configurations of the river are sketched and imprinted. Right: imprint of additional five intermediate stages of the depositional history by interpolating the two sketched rivers [71]



**Fig. 10** Left: a sketch-based approach is used to create 3D surface and subsurface geological features. Right: map view of the sketched strokes used to define the model to the left. Black stroke defines a river, blue strokes define deposition, red stroke defines erosion, and green dots define constant deposition [70]

deposition of constant thickness on the whole model. Open curves are interpreted as rivers; the terrain is therefore eroded using a simplified mathematical shape that mimics a river cross section. The eroded river can be filled up again with deposited material. The depositional history of a meandering river can be represented on the 3D model by interpolating between two river curves (see Fig. 9). Closed curves can be associated to erosion for defining basins or lakes, or to deposition for creating deltas or mountains. The amount of deposition or erosion is defined as a function of the distance to the closed curve (see Fig. 10).

The internal representation of the 3D model associates each curve to a heightmap. The sequence of curves results in a stack of heightmaps which defines a solid model. Deposition translate to a heightmap with positive values. Erosion translates to a heightmap with negative values which affects the heightmap below in the stack. The final visualisation is achieved with a ray-casting technique (for each pixel, a ray is sent into the scene and intersections are found) on the heightmap stack. The paper includes a user study indicating that the approach is user friendly and that it covers most of the structures relevant for fluvial systems and depositions but is too limited for more general geology such as structural geology.



**Fig. 11** Geological models showing a sequence of sliding along a nonplanar fault [71]

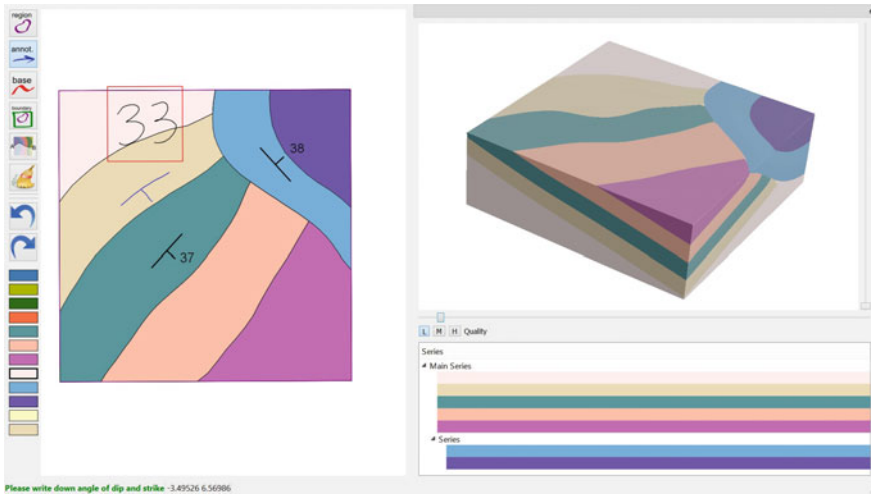
Further improvements and features are provided by Natali et al. [71] in a following article, where faulting and compaction are introduced in their models (see Fig. 11). The internal representation is founded on their previous work [70]. Faults can be directly drawn on the 3D model (one sketch on the side to define inclination and one on the top to shape the fault surface). From this information, each involved layer is detached in two by splitting the heightmap for each layer into two heightmaps. One of the advantages of this representation is that it can be easily parallelised on the GPU, which enables interactive sliding of the layers on either side of a nonplanar fault.

Amorim et al. [72] describe a sketch-based tool that produces a 3D conceptual model when the user draws and annotates a 2D geologic map. Their idea is to have the tool interpret the sketching symbols that geologists are trained in when drawing geological maps rather than requiring users to use the modelling metaphors provided by the programmer. Geologists register their observations on a 2D map which is directly translated to 3D. Amorim et al. proceed essentially in three main steps, as shown in Fig. 12: firstly they provide a 2D canvas where to draw geologic contacts (basically a separation of different rock layers) as seen from map view, and related annotations for depth and inclination. Then a graph-based representation is generated from the drawn geologic contacts which is used to detect unconformities and the rock layers sequence. Lastly, a Constructed Solid Geometry (CSG) conceptual model is built by combining surfaces that represents rock separation. The surfaces are implicit surface interpolations of the initial sketches with further gradient notion coming from the user's annotations.

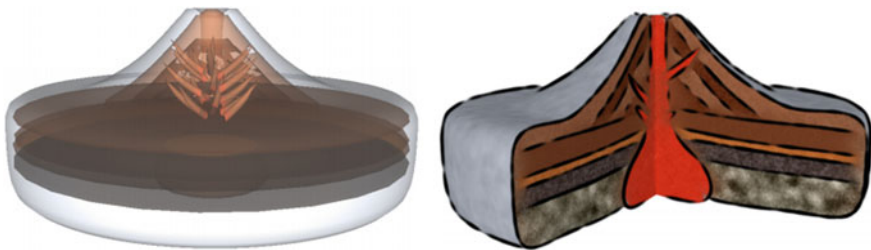
Lopes et al. [73] present a hardware and software solution where the user wears a VR headset and is able to shape a layered geological model using gestures in free air. The user defines with hand movements vertical-like surfaces which constitute the boundaries of the layer-cake model.

#### 4.1.5 Solid Representations

In this subchapter we consider solid representations. They differ from boundary representations in that they are not hollow, but have spatially varying properties inside. Takayama et al. [74] present diffusion surfaces as an extension of diffusion



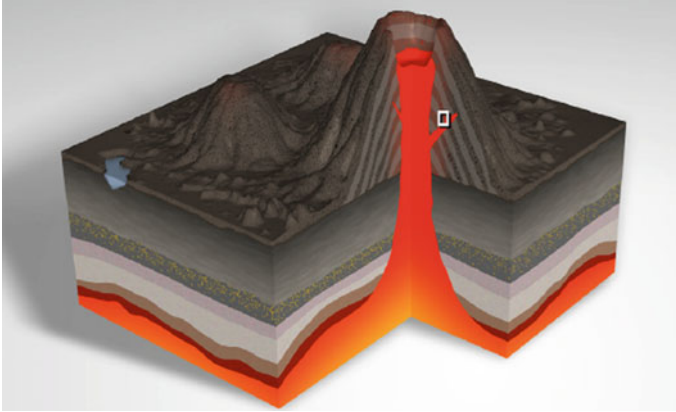
**Fig. 12** Left: the user draws a geologic map which includes geologic contacts (black curves) and dipping angles (T shaped symbols). Bottom right: the system finds a sequence of the rock layers from bottom to top that fits the sketch. Top right: a 3D geological model is generated from the user's sketch [72]



**Fig. 13** A volumetric representation of a geological scenario using diffusion surfaces [74]

curves [75] for defining solids. The representation consists of a set of coloured surfaces in 3D, describing the model's volumetric colour distribution. A smooth volumetric colour distribution that fills the model is obtained by diffusing colours from these surfaces. Colours are interpolated only locally at the user-defined cross-sections using a modified version of the positive mean value coordinates algorithm. A result of the work by Takayama et al. [74] is shown in Fig. 13.

In the work by Wang et al. [76], objects are represented as implicit functions using signed distance functions. Composite objects are created by combining implicit functions in a tree structure. This makes it possible to produce volumes made of many smaller inner components. This multi-structure framework lets them produce models irrespective of resolution (see Fig. 14 for their geological application example).



**Fig. 14** A volumetric representation of a geological scenario using an implicit representation [76]

## 4.2 *Sparse and Dense Data*

This section describes methods that use sparsely scattered geologically measured data such as wells, or dense data such as 3D seismic reflection volumes, for creating a subsurface model. In contrast to data-free modelling where the user produces a model based on a mental image or hypothesis, the modelling is now constrained by values in the data.

### 4.2.1 Measured Data

Subsurface data can be collected in several ways, at various effort and expense. Seismic 2D or 3D reflection data is collected by sending sound waves into the ground and analysing the echoes. When the sound waves enter a new material with a different impedance, a fraction of the energy is reflected. Therefore, various layer boundaries of different strength are visible in the seismic data as linear trends. Well logs are obtained by drilling into the ground and performing measurements and collecting material samples from the well. Outcrops are recorded by laser scans together with photography (LIDAR) to create a 3D point cloud of the side surface of geology [1]. This surface can be investigated and visible layer boundaries can be identified and outlined as curves along the surface. Heightmap data can be collected from state-run databases.

### 4.2.2 Interpretation

Many interpretation methods operate on dense data which we gave a description of in Sect. 3. Several commercial tools exist for interpreting 3D seismic data. One

example is the extensive software Petrel [77] where the user can set seed points and the system grows out a surface. The user can change the growing criteria or the seed points until a satisfactory surface is extracted. This can be time consuming. Kadlec et al. [78] present a system where the user interactively steers the growing parameters to guide the segmentation instead of waiting until the growing is finished before being able to investigate it. Fast extraction of horizon surfaces is the focus of Patel et al. [37]. Their paper introduces the concept of brute-force and therefore time-consuming preprocessing for extracting possible structure candidates in 3D seismic reflection volume. After preprocessing, however, the user can quickly construct horizon surfaces by selecting appropriate candidates from the preprocessed data. Compact storage of all surface candidates is achieved by using a single volumetric distance field representation that builds on the assumption that surfaces do not intersect each other. This representation also opens up for fast intersection testing for picking horizons and for high quality visualization of the surfaces. The system allows the user to choose among precomputed candidates, but editing existing surfaces is not possible. Editing is addressed by Parks [79]. He presents a method that allows to quickly modify a segmented geologic horizon and to cut it for modelling faults. Free-form modelling is achieved using boundary constraint modelling [80] (a method originating in the Computer Graphics community); this is simpler and more direct than spline modelling, which requires manipulation of many control points. Discontinuities arising from faults are created by cutting the mesh. Amorim et al. [81] allow for more advanced surface manipulation in their system. Surfaces with adaptive resolution can be altered and cut with several sketch-based metaphors. In addition, the sketching takes into account the underlying 3D seismic so that it can automatically detect strong reflection signals which may indicate horizons and automatically snap the sketched surface into position. Motta et al. [82] present a sketch-based approach to segment salt bodies in seismic data. This is useful as salt bodies are hard to extract automatically or even semi-automatically. Their work take an initial mesh and deform it using Laplacian methods for surface editing [83]. Liu et al. [84] present a sketch-based interpretation approach for 3D seismic data where the user can, in a 2D slice view, perform a fast partitioning of the slice for segmenting it and for illustrating it with textures of geological symbols.

The terrain and the subsurface on Earth is created through a series of geological events such as erosion, faulting and folding. Interpretation of the ground results in a model of the current subsurface state. The act of geological restoration is important in geology and tries to stepwise undo in reverse chronology the geological events that have taken place. In Lidal et al.'s paper [85], a tool for 2D creation, modelling and interpretation of geological scenarios in the subsurface is described. The tool allows for sketching up several alternative restoration timelines and organize them in a story tree. Garcia et al. [86] develop this concept further by incorporating a deformation simulation using a mass spring system which supports simple backward simulation directly on the 2D sketch. The backward simulation can undo the geological events of deposition, erosion, compaction, folding and faulting. Laurent et al. [87] apply the deformation method presented by Botsch and Kobbelt [80] to interactively deform, forward model and restore (backward simulate) 3D geological structures. The defor-

mations presented in their work are however more coarse and global than the ones presented in Garcia et al. [86].

### 4.2.3 Interpolation

Key interpolating methods for surfaces in geosciences are the Kriging method, the Discrete Smooth Interpolation (DSI) method [88–90], the Natural Neighbor Interpolation method [91], Radial Basis functions, the Inverse Distance method, and spline methods.

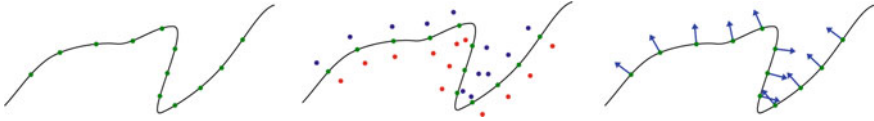
Kriging is a statistical approach to interpolation that incorporates domain knowledge and is uncertainty-explicit [92, 93]. Kriging, like exemplar-based synthesis, creates a surface that has similar properties to an example dataset. Kriging, originating from geostatistics, calculates, based on available samples, how the variation of heights between samples change as a function of the distance between the samples. In terrains, neighbouring points have more similar height values than points further away. By calculating a variogram, which has variance on the y-axis and distance on the x-axis, the variability is captured. This information is then used to interpolate values by finding height values, so that the interpolated point fits the characteristics of the variogram.

The Discrete Smooth Interpolation allows for integration of geo-physical constraints into the interpolation process. The interpolator takes as input a set of  $(x, y)$  positions, some with height values and others lacking. After interpolation, the lacking height values have been calculated. Discontinuities between positions can be defined so that certain points do not contribute during interpolation. Typically, for a horizon surface, discontinuities would be added over fault barriers. In addition, constraints such as having points being attracted towards other points, having points being limited to movement along predefined lines or on surfaces can also be defined. These constraints are useful for interpolating geologic surface data. However, the method might not be well suited for cases with very little or no observation data (as indicated by De Kemp and Sprague [28]), such as in the data-free scenario.

Natural Neighbor Interpolation is also based on a weighted average, but only of the immediate neighbours around the position to be interpolated. A Voronoi partition is created around all known points and the weight is related to the area of these partitions around the unknown point.

Inverse Distance Weighted Interpolation works on points with planar position  $(x, y)$  and height  $z$ . The height for an unknown  $(x, y)$  position is a weighted average of the known points. The weight assigned to each known point diminishes with the distance to the unknown point according to a power function with a user defined factor  $p$ . When  $p = 2$ , Euclidean distance is used.

Carr et al. [94] present a method to build 3D surfaces from point clouds using radial basis functions (RBF). The proposed technique creates an implicit surface  $S = \{(x, y, z) \mid F(x, y, z) = 0\}$ , where  $F$  is a RBF, and interpolates the given points; i.e.,  $F(p) = 0$  for  $p$  in  $\mathcal{P}$ , where  $\mathcal{P}$  is the given point cloud sampled from the surface. By the time of publication, the computational power restricted the number of points



**Fig. 15** Example of samples used by methods based on RBFs. From left to right: only (green) points on the surface; the auxiliary points proposed in [94] red inside and blue outside; and the points and normals used by the HRBF formulation in [96]

that were possible to fit using RBF's techniques. The authors showed that it was possible to adapt RBFs to fit a large number (up to  $3 \times 10^5$ ) of points using methods to accelerate the function evaluation. This work also introduced an approach to improve the surface description enabling the representation of complex topologies. The authors add auxiliary points with values representing the signed distance of each point to the surface, then fitting an RBF that interpolates these values, i.e.,  $F(x_i, y_i, z_i) = v_i$ , with  $v_i = 0$  for points on the surface, and  $v_i \neq 0$  for auxiliary points. The authors proposed to place the auxiliary points on the surface normals (see Fig. 15). This method has successfully been adopted in mining industry to create geobodies with respect to minerals [95].

Later, Macêdo et al. [96] improved the RBF formulation to interpolate the surface's points and normals – Hermite-RBF (HRBF). This approach eliminates the need to create auxiliary points and is more apt to represent complex geometries. This new formulation enables the creation of closed surfaces using sparse samples, which was used by Vital Brazil et al. [17] to create terrain models from sketches. In Fig. 15, we illustrate the auxiliary points proposed in [94] and the points and normals used in [96]. It is easy to see the possible issue when one uses auxiliary points to represent surfaces that have narrow parts; the distance assigned in the normal direction could not represent the real distance to the surface. On the other hand, the use of normals makes the surface representation much more robust to this problem.

While the HRBF interpolator by Macêdo et al. was developed in the computer graphics and modelling domain, the implicit function interpolator methods [97, 98] had been developed a few years earlier in the geosciences domain for interpolating geological interfaces. Both methods use implicit functions for creating a surface that interpolates data based on both surface points and surface normals.

Another method interpolating both points and normals is made by Wu et al. [99]. They introduced a numerical technique based on an augmented-Lagrangian method and give an example of creating terrains. The method allows for defining a terrain based on two types of data, either the height value at a specific  $(x, y)$  position, or the normal vector of the terrain surface at a specific  $(x, y)$  position. The input is similar to the method used in the Hermite-Birkhoff Radial Basis Functions [72], but the mathematics behind is different, and the final result is a heightmap instead a set of implicit functions (RBFs). This approach may be suitable for sketch-based techniques, although it does not support overhangs which methods based on Radial Basis Functions do.



For all methods using radial basis functions for interpolating surfaces, it can be noted that there exists three degrees of specialization. Methods only interpolating points [94] (with no extra information such as normal orientation) use regular radial basis functions. When each point also has a normal, the Hermite-RBF can be used [96]. Finally, when each point either has a normal, or a position, or both, the Hermite-Birkhoff RBF formulation can be used [72].

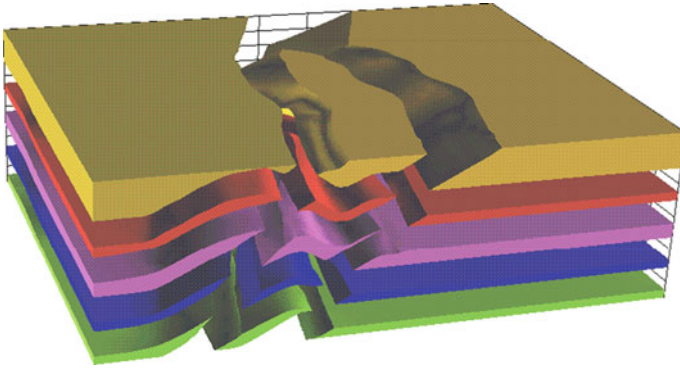
An interpolation and surface representation system for geology is discussed in the work by Floater et al. [100]. Scattered point measurements can come in many forms, uniformly scattered, scattered in clusters, along measurement lines or along iso-curves. Fitting a surface through the points requires interpolation. Different interpolation methods vary in quality depending on the distribution of the scatter data. Floater et al. offer interpolation in form of piecewise polynomials (splines) on triangulations, radial basis functions or least squares approximations.

Although more of a connectivity algorithm than an interpolation algorithm, Ming and Pan [24] present a method for constructing horizons from borehole data. Each borehole dataset consists of a sequence of regions. Each region has its start and end depth specified as well as its rock type. Figure 19a exemplifies this. One rock type might appear in several layers and also the rock type sequence might vary between boreholes. This results in several possible connectivity solutions. The challenge is to make a suitable matching of layers to create a solid layer for each rock type.

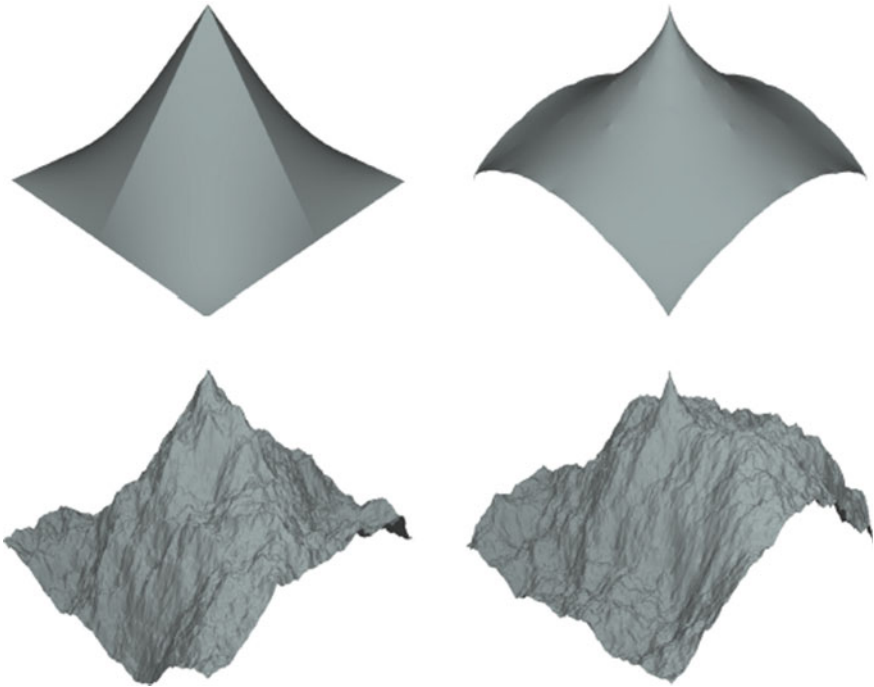
Faults define the discontinuity of horizons, however, when interpreting seismic data, fault and horizon surfaces will not be perfectly aligned and will either have gaps or overlaps between each other. Closing gaps by extending horizon surfaces slightly and then cutting them at fault intersections can yield topological errors. Euler et al. [101] propose to use a constrained interpolator (DSI) to control which horizons will be extrapolated and to which faults they will be extrapolated to. This creates a correctly sealed model (see Fig. 16).

The paper uses a test dataset called Overthrust model SEG-EAGE 1994 [102] consisting of four horizons and two faults that merge into one fault. The four horizons divide a cube into five layers. Each layer is divided into three parts by the faults resulting in fifteen distinct blocks.

Belhadj [10] models a terrain through a fractal-based algorithm. The aim of the author is to reconstruct Digital Elevation Map (DEM) models. The surface is reconstructed with constraints consisting of scattered points of elevation provided by a satellite or other sources of geological data acquisition. Furthermore, it is possible for the user to change the final shape of the terrain by intervening with sketches on the model. As the goal was to have an interactive model, the choice of the algorithm has been a fractal based approach instead of a physically based one. Specifically, part of the work is based on the so-called Midpoint Displacement Inverse process (MDI), shown in Fig. 17. MDI does not allow reconstruction constraints, therefore a new adapted version of this technique has been proposed, named Morphologically Constrained Midpoint Displacement (MCMD). To be able to include constraints in the interpolation computation, MCMD introduces changes in the order of computation of the midpoint displacement.



**Fig. 16** The work by Euler et al. [101] presents a solid model made from surfaces in the Standard overthrust SEG-EAGE 1994 model [102]



**Fig. 17** Example by Belhadj et al. [10] of constrained mid-point displacement. Five points define the constraints for the generated surface

#### 4.2.4 Surface Representations

In the work by Floater et al. [100], surfaces can be created from scattered data and are represented either as an explicit surface ( $f(x, y) = z$ ), parametric surface ( $f(i, j) = (x, y, z)$ ) or a triangulation. In addition they support offset surfaces defined by one parametric surface and a function that offsets the main surface along the surface normal. Parametric surfaces can be created from triangulated surfaces.

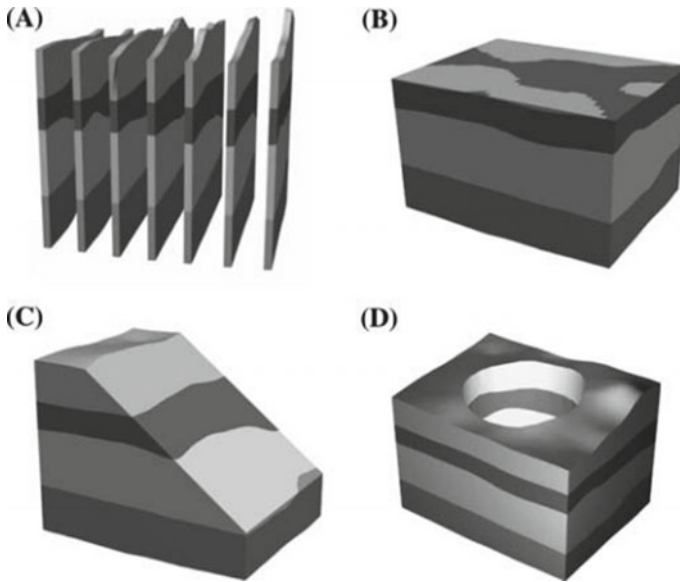
In the works by De Kemp and Sprague [28, 29], surface modelling using traditional Bezier curves and B-splines [103] is discussed. Bezier curves are used as approximate curves, while B-splines are employed when interpolative curves are better suited. All points can be interpolated or approximated. For non-interpolated points, attraction weights can be specified. However, controlling a large number of control points individually can be tedious and lead to meaningless localized distortions. To ameliorate this issue, the authors present the technique of having hierarchical control points of decreasing resolution so that the user can move control points in the hierarchy he/she wishes to displace the surface, to avoid manipulating an excessive number of control points at the lowest resolution level. They use the concept of structural ribbons for describing a curve with normals. It can be considered as a thin strip of the surface that can be fitted on available geological information such as outcrops or map traces.

In areas of importance in an interpretation with sparse interpolated data, the paper expresses the need for expert users to be able to override and alter the coarse approximation and easily update the model when new data arrives. An expert typically attempts to get an understanding of the processes that were operative in shaping the final geometry of a given structure while at the same time respecting the local observational data.

#### 4.2.5 Solid Assembly

Solid geometric representations of subsurface structure are important for analysis. A sealed model enables consistent inside/outside tests, providing well-defined regions and good visualizations. It is also the first step for producing physical simulations of liquid or gas flow inside the model at later stages.

Baojun et al. [27] suggest a workflow for creating a 3D geological model from borehole data using commercial tools and standards. They use ArcGIS [104] for creating interpolated surfaces from the sparse data. They use geological relevant interpolation such as Inverse Distance Weighted, Natural Neighbor, or Kriging interpolation. This approach results in a collection of heightmaps which are imported into 3D Studio Max and stacked into a layer cake model (see Fig. 18). Then Constructive Solid Geometry (CSG) [105] operators are used to create holes (by boolean subtraction) at places where data is missing in the well logs. The model is then saved as VRML [106] enabling widespread dissemination since it can be viewed in web browsers.



**Fig. 18** Geological model made with CSG operations in 3D Studio Max shown with different cut styles available in the program [27]

When representing subsurface volumes using geometric surfaces for each stratigraphic layer, Caumon et al. [30] describe two constraints that must be followed. The constraints are that only faults can have free borders, i.e. horizon borders must terminate into other surfaces, and that horizons can not cross each other. Following the rules results in a correct and sealed model. This requires that each volume is described by a boundary triangulation with no holes and with shared vertices on seams of intersecting surfaces. Maintaining these constraints when editing horizons and faults is discussed. Mass conservation and deformation constraints during editing is also discussed. In their later work [5], additional geometric rules are introduced. The surface orientation rule states that geological surfaces are always orientable (i.e. having no twists, no Möbius ribbon topology and no self-intersections). Due to the physical process of deposition, they suggest an optional constraint requiring that horizons must be unfoldable without deformation, i.e. that they are developable surfaces with zero Gaussian curvature everywhere. They state that using implicit surfaces instead of triangulated surfaces directly enforce several validity conditions as well as making model updates easier, however at the cost of larger memory consumption. They also discuss the importance of being aware of the varying degree of uncertainty in the different measured data modalities and, for instance, using triangulations of different coarseness according to the sparseness and uncertainty of the underlying observations. The paper presents general procedures and guidelines to effectively build a structural model made of faults and horizons from sparse data such as field observations. When creating a model, they start with fault modelling and

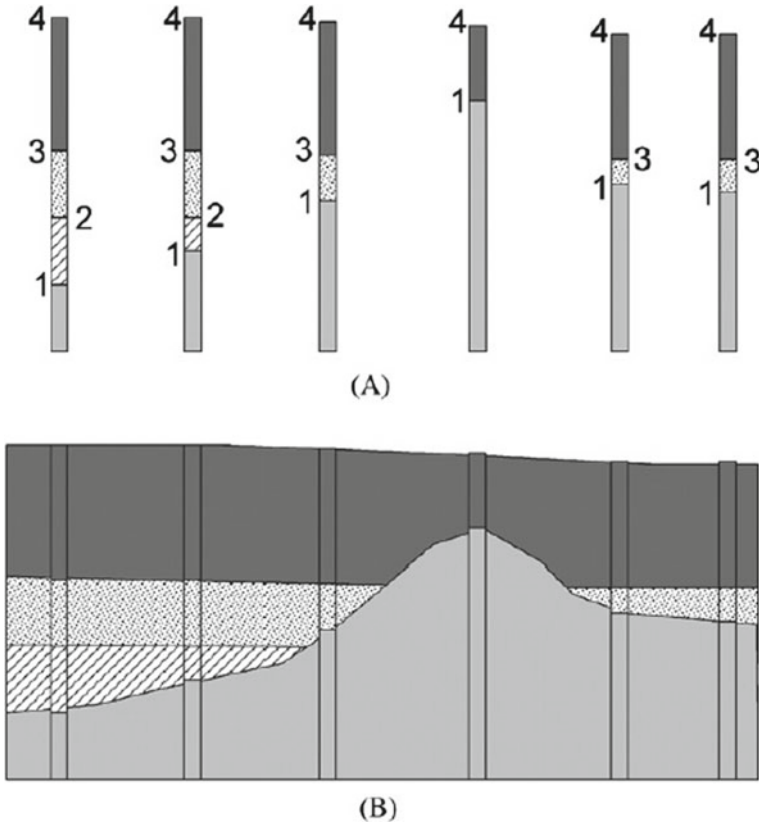


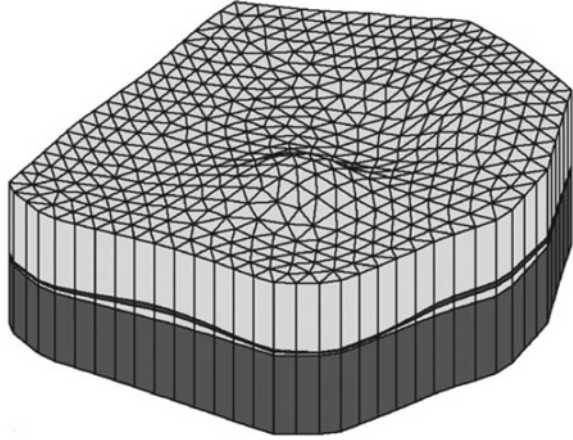
Fig. 19 Borehole data in **a** and resulting interpolation in **b** from the method by Lemon and Jones [22]

then define the connectivity among fault surfaces. Finally, horizons are introduced into the model. However, if the fault structure is very complex, they state that it is wiser to define the horizons first as if there were no faults and introduce the faults and their consequence on horizon geometry afterwards.

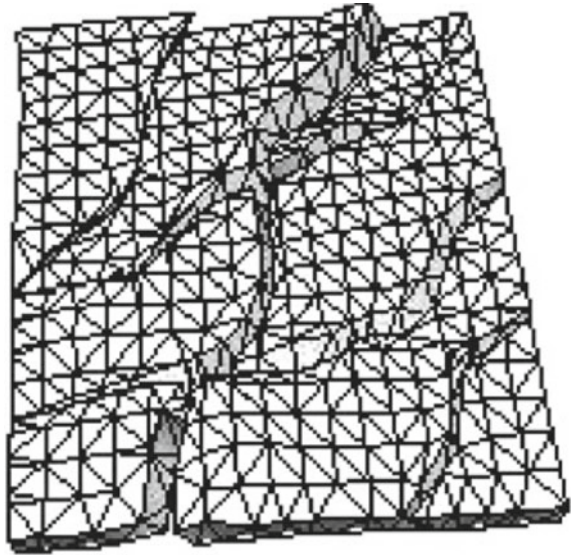
Lemon and Jones [22] present an approach for generating solid models from borehole data (see Fig. 19). The borehole data is interpolated into surfaces. For creating a closed model, they state and exemplify that CSG together with set operations can be problematic as the set operation trees grow quickly with increased model complexity. They simplify the model construction by representing horizons as triangulated surfaces while letting all horizon vertices have the same set of  $(x, y)$  positions and only varying the  $z$  positions (see Fig. 20). This simplifies intersection testing between horizons and makes it trivial to pairwise close horizons by triangulating around their outer borders.

Complexity increases when models must incorporate discontinuities in the layers due to the faults. Wu and Xu [31] describe the spatial interrelations between faults

**Fig. 20** Example of model created with method by Lemon and Jones [22]. The shared (x, y) vertex positions can be seen on the side surfaces



**Fig. 21** Mixed-mesh model employed by Wu [31]. A regular mesh is used in non-boundary continuous areas, whilst an irregular triangulated mesh is adopted elsewhere



and horizons using a graph with horizons and faults as nodes. The graph is used to find relevant intersections and bounding surfaces which are Delaunay triangulated to form closed bodies (as shown in Fig. 21). In a follow-up paper [107], two types of fault modelling techniques are compared (based on what they call *stratum recovery* and *interpolations in subareas*) and a unified modelling technique for layers and faults is presented to solve the problems of reverse faults (i.e. convergent sedimentation blocks), syn-sedimentary faults (when slumping of sedimentary material happens before it is lithified) and faults terminated inside the model (blind faults).

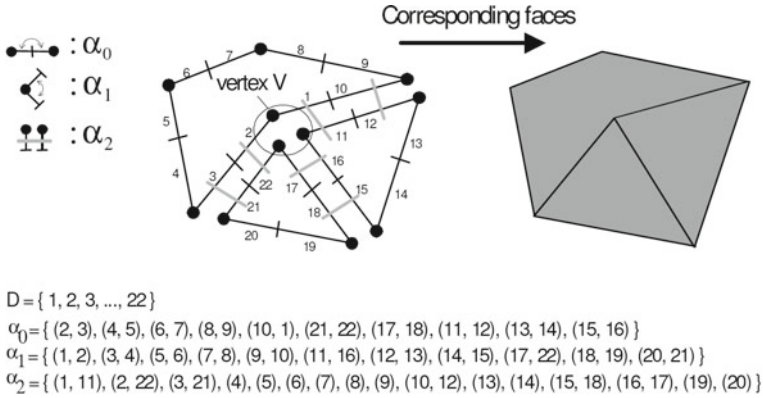


Fig. 22 Example of data structure explaining the 3-G-map definition [108]

### 4.2.6 Solid Representations

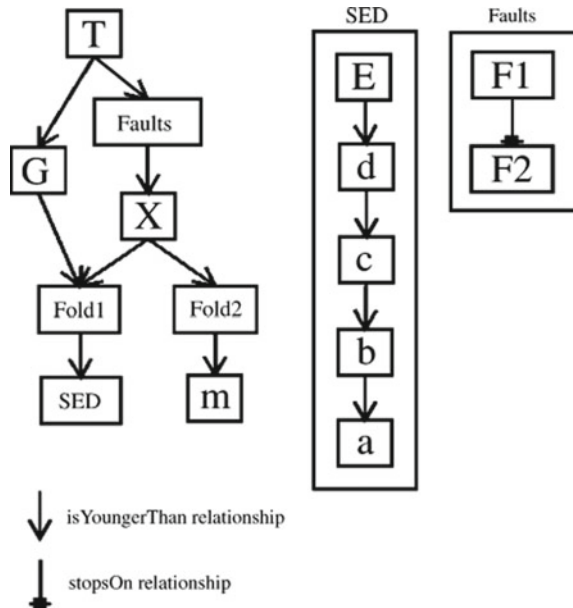
Solid modelling tools in CAD do not easily support subsurface features such as hanging edges and surface patches. Many papers describe data structures for representing the solid blocks that horizons and faults subdivide the subsurface into. Boundary representations are frequently used. Generalized maps, used for describing closed geological models, are introduced by Halbwachs and Hjelle [108].

A 3-Generalized map (3-G-map) [109] is a boundary representation appropriate for defining the topology of subsurface structures. A 3-G-map is defined as a set of darts  $D$  and three functions on them:  $\alpha_0, \alpha_1, \alpha_2$  and  $\alpha_3$  (see Fig. 22 for a 2D example). If one considers an edge as the line between two vertices, then a dart is a half-edge starting at a vertex and ending at the centre of the edge. The three functions map from darts to darts and sew half-edges into edges (by  $\alpha_0$ ), edges into polygons (by  $\alpha_1$ ), free polygons into connected polygons (by  $\alpha_2$ ), and defines neighbouring connected closed polygon volumes (by  $\alpha_3$ ). The 3-G-map is a simple yet powerful structure for defining the topology, in such a way that it is easy to traverse the space between connected or neighbouring vertices, surfaces and solids. Apel [110] presents a comparison of 3-G-maps with other boundary structures.

To create a 3-G-map, relations are defined on the faults and horizons, describing how a surface is terminated onto/cut by another surface. The construction process is divided in two, first geometries are created, then they are glued together through defining topology. The 3-G-map encapsulates the topology of the final model. For 3-G-maps, topology must be described very detailed. To relieve the user from this task, several abstractions have been suggested. By letting the user instead define the relation and cuts between horizons and faults in a graph or tree datastructure, the system can then generate a detailed topology description from this. In the work by Brandel et al. [111], the user specifies a graph of chronological order for when the surfaces have been physically created. In addition a graph describing the fault network using the relation “fault A stops on fault B”, is specified. This work is



**Fig. 23** Example of a graph that describes the relations between geological surfaces [112]



extended [112] to include meta-information in the nodes of the graphs for explicitly expressing the geological knowledge attached to each of the geological surfaces. Examples of such information is that a surface is onlap, erosional, older than or stops on another surface. An example of such a graph can be seen in Fig. 23.

Implicit surfaces (implicits) provide a suitable way to represent geological solids [98]. Essentially, such solids are described by implicit functions that can be expressed in different forms, e.g., distance based models, analytical functions, interpolation schemes like for instance RBFs, etc. Pasko et al. [113] generalized the above representations, which lead to an inequality,  $f \geq 0$ , also called *functional representation* of solids. Kartasheva et al. [114] introduced a robust framework to model complex heterogeneous solids, which was based on functional representation. The implicit solid definition is quite broad, and for instance, the terrain modelling using a heightmap can easily be represented by implicits [115].

Although not exactly a solid representation, we describe here methods for generating solid models having a discrete volumetric representation (i.e. consisting of voxels) based on descriptors such as parameters or 3D example datasets.

When a detailed model of subsurface geology is required, stochastic generation methods can be applied. A common usage is the filling of unknown volumetric content inside a closed boundary, such as the ore grade (grams per tonne) of a precious metal. Ensembles of plausible models are created stochastically for getting a better intuition of the probability distribution. This information can be used to decide where to extract the resource or where to take more samples to reduce uncertainty. The

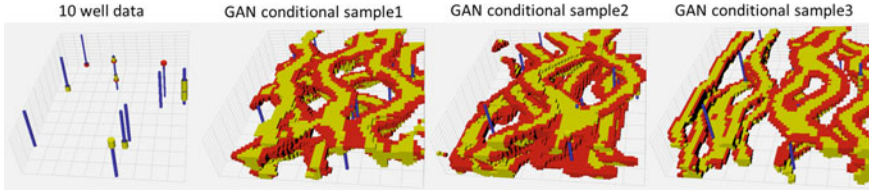
stochastic variability can be described by input parameters that match the (sparsely) measured sample data taken in the field.

Variograms (Kriging) as described in Sect. 4.2.3 for interpolating height data can also be used to create volumetric data. However, it produces smooth results and therefore cannot reproduce the heterogeneous distributions found in e.g. ore grade. Multiple-Point Statistics (MPS) is more powerful and is able to reproduce heterogeneities. As opposed to the variogram which uses two-point statistics as it is a function of two variables (the field values at two locations), MPS can closely reproduce spatial patterns from a training image (TI). The training image represents an explicit example of the heterogeneity that one wants to reproduce and can be physical field measurements from similar areas, or manually artificially made data that have the wanted variability. A recent overview of MPS has been made by Tahmasebi [116].

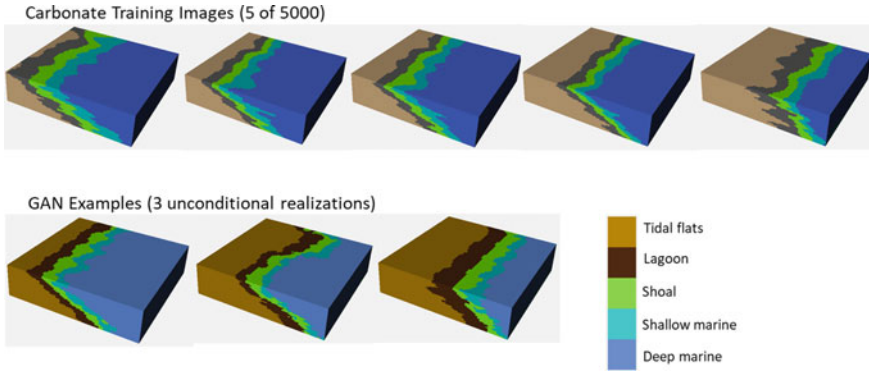
Well-defined geometric objects can also be generated stochastically. See Abdollahifard and Ahmadi [117] as an example. This is called object-based methods and is achieved by producing binary volumetric data instead of scalar-valued data. The binary volumetric data must follow certain topological constraints, such as having connected long strands in the case of creating channels. Stochastic methods have also been used for perturbing the geometry of an existing model. The motivation for perturbing the model is that the actual position of boundaries is only approximate due to e.g. limited resolution of measurements. An ensemble of models can be created to better represent the space of possible variations. Realistic simulations of, e.g., fluid flow (oil migration) can be performed on each model, and the results can be analyzed for identifying the sensitivities. The analysis can answer if, e.g., all variations create the same result, or if there are two main results (attractors) that the simulations evolve towards, and which parts of the model are most sensitive. When perturbing the model, it is important that the result also abides by known constraints. This is explored by Wellmann et al. [118] where they perturb horizons and faults of a model.

An interesting parallel development of MPS and texture synthesis in computer graphics has taken place and is described by Mariethoz and Lefebvre [119]. Both fields build on the same methods. In computer graphics, there is a need to generate realistic textures for applications such as video games and animated movies. For this, training images called exemplars are used as input.

Generative adversarial networks (GANs) are a relatively new technique that has been shown to create more realistic models than existing geostatistical modeling methods [120]. GANs have also been used to generate terrains, as discussed in Sect. 4.1.2. Using a GAN, a wide range of conceptual geological models honouring constraints such as well data can be generated. The GANs are trained using a library of models. GANs couple two competing deep convolutional neural networks: a generator, which creates new example models, and a discriminator, which differentiates between real and synthetic models. The networks are trained in an adversarial manner until the generator can create synthetic images that the discriminator cannot distinguish from “real” images. Two examples generated with the method by Zhang et al. [120] are shown in Figs. 24 and 25.



**Fig. 24** Conditional fluvial samples generated by GANs in 3D. Ten well data with the interpreted facies in the left-most display. Three categories are used (shale: blue; channel sand: yellow; levee: red). All three generated models (sample 1–3) generated by GANs honor the well data [120]



**Fig. 25** Training image examples (top) in 3D with 5 voxel categories (see legend bottom right) and unconditional realizations generated by GANs (bottom) [120]

## 5 Comparing Surface and Solid Representations for Geomodels

In this section, we compare and discuss the surface representation methods in Table 1 and solid representations in Table 2, in the context of how well suited they are for modelling geologic structures. The interesting features of such representations are: how close to a natural terrain the top surface is (*Terrain realism*, i.e., the ability to portray the properties of terrain such as randomness and the occurrence of all frequencies); modelling of faults (discontinuities); interpolation of input points (*gap-filling*); support for multi- $z$  values (*overhangs*). Ease of modelling (*control*), processing requirements, storage space requirements and the ability of simultaneous representation of high- and low-level details (*multiscale*) will also be discussed for each category. In Table 2, we compare the techniques with respect to their ability to model layers; support for tubular structures such as channels, caves or holes; ease of modelling; processing and storage requirements and multiscale support. These features are graded with plus for good support, minus for bad support, or 0 if neutral.

**Table 1** This table compares the abilities of surface representation methods in terms of modelling geological features

	Fractal & Noise-based	Erosional	Exemplar-based synthesis	Radial-basis function	Splines	Kriging	Discrete smooth interpolation
Terrain realism	+	+	++	-	-	+	0
Faults (discontinuities)	-	-	0	-	-	-	++
Gap-filling (interpolation)	+	-	+	++	+	++	+
Overhangs (multi-z values)	0	0	--	++	+	-	+
Control	-	-	0	+	+		
Processing requirements	-	-	-	-	+	-	--
Storage requirements	++	+	-	+	+	+	+
Multiscale	++	0	+	--	--	-	-

Methods to the left of the gray separator procedurally create surfaces, while methods to the right are interpolative

### 5.1 Surfaces

This subsection discusses the surface methods described in Table 1. The methods are split in two by a gray separating line. The three first methods produce surfaces procedurally, while the four last methods (four last columns) interpolate points into surfaces. Fractal techniques create a surface from input parameters; erosional methods create a surface from an input surface and simulation parameters; while exemplar-based methods create surfaces based on a collection of surface examples. Radial-basis functions and splines are defined by control points possibly set by a user. Kriging and DSI methods are completely automatic; therefore user control does not apply to them, and they are grayed out in Table 1. For comparing the capability to model faults, although any method can support this by splitting the surface into two, we strictly evaluate the methods in their mathematical formulation without allowing such a heuristic.

*Fractal and noise-based* methods (Sect. 4.1.1) are well suited for achieving a realistic appearance of the surfaces. In particular, fractals are ideal for expressing the self-similarity found in nature. In addition, noise can increase the random behaviour of real geological surfaces. Faults are difficult to represent with fractal or noise approaches, as they usually are represented by heightmaps that do not allow discontinuities. For the same reason, multi-z values can usually not be expressed with these techniques. Fractal and noise-based methods do not allow intuitive or local control of the surface,

but it is easy to vary few parameters to obtain a different result. There is no need to store data, fractal and noise behaviour is represented by compact analytical formulas. On the other hand, processing requirements can be high, depending on the complexity of the formula describing the surface shape. Multiscale behaviour is present in fractals by their definition.

*Erosion* (Sect. 4.1.1) is a process that affects terrain by simulating weathering. Therefore it is very well suited for modelling a natural appearance of the top layer. Erosion is modelled as a flow process and, therefore, does not handle discontinuities well. When used with Smoothed Particle Hydrodynamics (SPH), erosion needs to incorporate some data interpolation method to fill the gaps. Erosion processes can result in carvings, and thereby multiple  $z$  values. The erosion process is hard to control. Essentially a simulation with given parameters is initiated and the user can either accept the results or modify input parameters for a more satisfactory result. Storage requirements are low, while the resulting model can be of arbitrary size. On the other hand, erosion is a dynamic process that requires processing resources for the simulation.

*Exemplar-Based* (Sect. 4.1.1) techniques can, to a certain degree represent faults, but not real discontinuities since the methods (mostly) use heightmaps. It can, in theory, synthesize terrains with abrupt changes if the exemplars contain steep cliffs. The method was not initially designed for data interpolation. However, exemplar synthesis often works with having a filter expanding the border of the so-far-made-texture by filling in with parts of exemplars that have similar neighbourhoods. Therefore interpolation can be made by starting with a texture having the interpolation values set and letting the rest be synthesized. Multi- $z$  values are not supported for methods using a 2D heightmap. One could perform 3D texture synthesis, but this has not been explored for terrain generation. Classical exemplar-based methods offer no control at all, whereas more recent methods allow for a coarse input mesh [18] and can be guided by a user-sketched feature [63]. Brosz et al. [18] show how to use a base terrain and add details by texture synthesis. Storage requirements are quite high since many exemplars must be stored. Furthermore, creating the terrain is computationally expensive.

*Radial-Basis Functions (RBFs)* (Sect. 4.1.3) represent a variational interpolation technique that enables to fit/approximate an iso-surface to a given set of points and normals associated with these points. Here, the points can be given in arbitrary order, unlike splines which require a grid structure. An important feature, that might be seen as a drawback for geological models, is the  $C^n$  continuity of the resulting surface, which results in surfaces that are too smooth for geological structures. To produce highly realistic terrains one would need to specify a substantial number of points with varying normals to interpolate. On the other hand, the RBF method can easily fill the gaps in the surface model and model overhangs, which comes from the nature of the technique [121]. In practice, the specification of control points and normals can guide the appearance of the final surface. Moreover, modelling multiscale features is not directly supported due to the linear model composition. In order to visualize the final iso-surface, one needs to evaluate the function at the given point, which puts the computational burden on the surface generation step. Nevertheless, to

store the implicit function, one only needs to specify the function evaluation process based on the given points and normals.

*Splines* are defined as parametric surfaces that, similarly to other interpolation techniques, produce the surface from a set of control points and the tangent or normal vectors associated with the points. Note that splines require an ordered list of points, which makes the modelling procedure somewhat tedious. Similarly to other interpolation-based surfaces, spline surfaces are continuous by their nature, which makes it hard to create discontinuous faults or realistic terrains. In comparison to RBFs, splines require a greater effort to change a surface model to fill the gaps or to produce overhangs. On the other hand, the parametric form facilitates the computation and visualization of the resulting surface. Multiscale representations are natively, similarly to RBFs, not supported by the spline model definition.

*Kriging* (Sect. 4.2.3) produces good terrain realism because the interpolated values are correct in a statistical sense. In addition, Kriging is ideal for filling gaps in the input dataset since the method is tailored for interpolating terrains using statistics.

*Discrete Smooth Interpolation* (DSI) (Sect. 4.2.3) belongs to the family of interpolation techniques that compute the missing information (function values) on a given graph. As such, it provides a powerful framework for modelling specific features in geology. For instance, the information about discontinuities on a set of vertices can be specified by cutting out connected nodes or by adjusting their contributing weights [88]. Since the entire evaluation procedure that computes the unknown values at a graph node requires a minimization (iterative) algorithm, the processing complexity is very high compared to other interpolation techniques. However, DSI is efficient in iterative modelling when one needs to adjust an existing model. Essentially, to update the node values, only a few steps of the iterative minimization procedure are required. Since DSI evaluates values at nodes and not anywhere else, one stores only the graph nodes with their attributes and connectivity information. Due to this property, they do not support multiscale surface representations.

## 5.2 Solids

The output from the surface methods in Table 1 is input to the solid-creation methods in Table 2. Solids can then be faulted or carved after creation if the method supports this.

*Implicit Solids* (Sect. 4.2.6) do not offer any special classes of implicits aimed at geological models. Nevertheless, they offer a variety of techniques to represent such models. For instance, layers can be represented by a combination of implicit primitives or by the utilization of RBFs. Additionally, cavities can be realized by a subtraction operator applied to two or multiple compound objects [114]. The representation of implicits in multiscale models has also been successfully introduced [121]. Moreover, the interactive modelling capabilities become more and more prominent with the introduction of sketch-based interfaces [17, 122]. One of the major advantages of implicits, when representing even very complex objects, is their storage require-

**Table 2** This table shows and compares the abilities of solid representation methods in terms of modelling geological features

	Implicit solids	CSG	3-G-Maps	Voxel representation	Diffusion surfaces [TSNI10]	Vector volumes [WYZG11]
Layer support	+	+	++	+	○	+
Channels/cavities support	+	+	+	++	+	+
Ease of modelling	+	+	+	-	+	-
Processing requirements	--	○	○	+	+	○
Storage requirements	++	+	○	--	+	-
Multiscale	+	+	+	--	○	++

ments, which is simply represented by the function evaluation process. On the other hand, to visualize the final solids, one needs to convert the implicit models into a set of triangles or adopt a direct ray-casting method.

*Constructive Solid Geometry (CSG)* can be used to compose a layer-cake model with simple layers in terms of shape definition. It is also adaptable to multiscale solutions and channels/cavities representation (e.g., employing the logical set operator minus). CSG is defined by simple primitives and set operators, but the global shape is difficult to intuitively control when the model starts to become complex ([22]). If the primitives are basic geometrical objects, CSG does not require much memory, and their logical interactions are relatively quick.

*3-G-maps* are the representation of choice for several geological solid modelling approaches [108, 111]. This is a boundary representation where the boundaries are typically triangulations. Details at different scales are supported and depend on the detail level of the underlying geometry. During modelling, the triangulations and their topology must be synchronously updated. There are no particular challenges with respect to the processing or storage of 3-G-maps.

A *voxel representation* is essentially a regular 3D discretized volume representation with given values in each sample. It can store layer information by simply tagging each voxel with a bit pattern defining a certain segmentation mask that defines the layer. Due to its expressiveness, it can also easily handle faults. In both horizons and faults, the final representation might need to be further processed in order to avoid visual artefacts arising from the space discretization. It can also support complex shapes, like cavities and channels. Voxel representations have been used for expressing channels [16]. This data structure does not offer a natural modelling approach that would be simple to use for a modeller. Therefore, it is often combined with other modelling representations. A voxel representation is very space demanding, but it



does not require a computational stage for evaluation, as it explicitly stores values in memory.

*Diffusion Surfaces* (Sect. 4.1.5) apply to layered models, but the approximation introduced by Takayama et al. [74] in their tool restricts each layer to have a rotational symmetry. Diffusion surfaces lack ease of modelling when dealing with multiscale models because a user has to define every boundary surface that delimits a piece of the solid model. Cavities are representable with diffusion surfaces, in particular if they have symmetries in shape. In producing the volumetric colour distributions, it is not necessary to perform precomputations, as opposed to Poisson approaches. Storage requirements are low because colours are interpolated locally at cross-sectional locations.

*Vector Volumes* (Sect. 4.1.5) is a volumetric representation of objects represented as a tree of signed distance functions (SDF trees). Thus, vector volumes combine the benefits of voxel and implicit models. Since each SDF tree contains information about the interior and the exterior of an object in a hierarchical fashion, vector volumes provide a powerful way to represent solids at different levels of detail, although the storage requirements can become very high. Moreover, such a representation requires a tedious way to interactively update the model. Although a volumetric object markup language is described [76], we do not consider it a straightforward solution. A volumetric object is usually achieved instead by developing and updating the model via different representations, e.g., boundary representation, and then performing the conversion into vector volumes. Due to the unique voxel and SDF identification when performing a volume ray-casting, this representation becomes efficient in direct visualization. Nevertheless, it is still required to evaluate the implicit function at each node in addition.

## 6 Challenges and Trends in Geological Modelling

Geoscience technology on closed model representations and model updating has not progressed at the same speed as in computer graphics. Better knowledge transfer between these groups could be advantageous. Caumon et al. [5] state that beginners with 3D modelling too often lose their critical sense about their work, mostly due to a combined effect of well-defined graphics and non-optimal human-machine communication. It is also important that a structural model can be updated when new data becomes available, or perturbed to account for structural uncertainties. In other words, with current modelling technology, uncertainty is difficult to express, and models are hard to update.

Researched literature from this domain emphasizes a strong need for modelling technology for communication and further analysis of the Earth's subsurface. While several matured methods are now in use by the domains of geology and geosciences, all tools require considerable effort to build structural models. Current tools focus on precise modelling in favour of rapid modelling. We believe that rapid modelling

is the key for the ability of expertise exchange, especially in the early phases of the interpretation process.

As interesting research directions that would benefit from future attention of the graphics community, we point out two distinct ones: one research direction can be procedural geological modelling that takes advantage of sparsely defined acquired information about the subsurface. Ideally, an automated procedural method could be refined by the user through a series of sketches. Another research direction can be the consideration of temporal aspects in geology. Erosion has been investigated in this context, but geological processes are driven by many more phenomena than only surface erosion. Here, also the intended process can benefit from user input in the form of sketched information.

## References

1. J. Pringle, J. Howell, D. Hodgetts, A. Westerman, D. Hodgson, Virtual outcrop models of petroleum reservoir analogues: a review of the current state-of-the-art. *First Break* **24**, 33 (2006)
2. S. Houlding, *3D Geoscience Modeling: Computer Techniques for Geological Characterization*. Springer (1994)
3. A.K. Turner, Challenges and trends for geological modelling and visualisation. *Bull. Eng. Geol. Environ.* **65**(2), 109–127 (2006)
4. A. Turner, C. Gable, A review of geological modeling, in *Three-Dimensional Geologic Mapping for Groundwater Applications*. *Minnesota Geological Survey Open-file Report*, 2007, pp. 07–4
5. G. Caumon, P. Collon-Drouaillet, C. Le Carlier, S. de Veslud, Viseur, J. Sausse, Surface-based 3D modeling of geological structures. *Math. Geosci.* **41**(8), 927–945 (2009)
6. F. Wellmann, G. Caumon, Chapter one—3-d structural geological models: concepts, methods, and uncertainties, in *Advances in Geophysics*, vol. 59 (Elsevier, 2018), pp. 1–121
7. J. Olsen, Realtime procedural terrain generation, tech. rep., Department of Mathematics and Computer Science (IMADA) University of Southern Denmark, 2004
8. A. Peytavie, E. Galin, J. Grosjean, S. Merillou, Procedural generation of rock piles using aperiodic tiling. *Comput. Graph. Forum* **28**(7), 1801–1809 (2009)
9. J. Schneider, T. Boldte, R. Westermann, Real-time editing, synthesis, and rendering of infinite landscapes on GPUs, in *Vision, Modeling, and Visualization 2006: Proceedings*, 22–24 Nov 2006, Aachen, Germany, 2006, p. 145
10. F. Belhadj, Terrain modeling: a constrained fractal model, in *Proceedings of the 5th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, vol. 1, 2007, pp. 197–204
11. P. Krištof, B. Beneš, J. Krivánek, O. Št'ava, Hydraulic erosion using smoothed particle hydrodynamics. *Comput. Graph. Forum* **28**(2), 219–228 (2009)
12. H. Hnaidi, E. Guérin, S. Akkouche, A. Peytavie, E. Galin, Feature based terrain generation using diffusion equation. *Comput. Graph. Forum* **29**(7), 2179–2186 (2010)
13. P. Prusinkiewicz, M. Hammel, C. Tn, A fractal model of mountains with rivers, *Panorama*, 1993, pp. 174–180
14. M. Hudak, R. Durikovic, Terrain models for mass movement erosion, in *Proceedings of EG-UK TPCG Conference*, 2011
15. J. Gain, P. Marais, W. Straß er, Terrain sketching, in *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, 2009, pp. 31–38

16. A. Peytavie, E. Galin, J. Grosjean, S. Merillou, Arches: a framework for modeling complex terrains. *Comput. Graph. Forum* **28**(2), 457–467 (2009)
17. E. Vital Brazil, I. Macêdo, M. Costa Sousa, L. H. de Figueiredo, L. Velho, Sketching Variational Hermite-RBF Implicits, in *Proceedings of Sketch Based Interfaces and Modeling*, 2010, pp. 1–8
18. J. Brosz, F. Samavati, M. Sousa, Terrain synthesis by-example, *Advances in Computer Graphics and Computer Vision*, 2007, pp. 58–77
19. M. Natali, I. Viola, D. Patel, Rapid visualization of geological concepts, in *SIBGRAPI 2012 (XXV Conference on Graphics, Patterns and Images)*, Aug 2012
20. E. M. Lidal, M. Natali, D. Patel, H. Hauser, I. Viola, Geological storytelling, *Computers & Graphics*, 2013
21. R. Groshong, *3-D Structural Geology: A Practical Guide to Surface and Subsurface Map Interpretation* (Springer, 1999)
22. A. Lemon, N. Jones, Building solid models from boreholes and user-defined cross-sections. *Comput. Geosci.* **29**(5), 547–555 (2003)
23. O. Kaufmann, T. Martin, 3D geological modelling from boreholes, cross-sections and geological maps, application over former natural gas storages in coal mines. *Comput. Geosci.* **34**(3), 278–290 (2008)
24. J. Ming, M. Pan, An improved horizons method for 3D geological modeling from boreholes, *2009 International Conference on Environmental Science and Information Application Technology*, 2009, pp. 369–374
25. D. Dhont, P. Luxey, J. Chorowicz, 3-D modeling of geologic maps from surface data. *AAPG Bull.* **89**(11), 1465–1474 (2005)
26. Z. Zhang, 3D Terrain reconstruction based on contours, in *Computer*, 2005, pp. 3–8
27. W. Baojun, S. Bin, S. Zhen, A simple approach to 3D geological modelling and visualization. *Bull. Eng. Geol. Environ.* **68**(4), 559–565 (2009)
28. E. De Kemp, K. Sprague, Interpretive tools for 3-D structural geological modeling part I: Bezier-based curves, ribbons and grip frames. *GeoInformatica* **7**(1), 55–71 (2003)
29. K. B. Sprague, E.A. de Kemp, Interpretive tools for 3-D structural geological modelling part II: surface design from sparse spatial data. *GeoInformatica* **9**(1), 5–32 (2005)
30. G. Caumon, F. Lepage, C.H. Sword, J.-L. Mallet, Building and editing a sealed geological model. *Math. Geol.* **36**(4), 405–424 (2004)
31. Q. Wu, H. Xu, An approach to computer modeling and visualization of geological faults in 3D. *Comput. Geosci.* **29**(4), 503–509 (2003)
32. T. Frank, Geological information retrieval using tetrahedral meshes, *gocad.org*, 2006, pp. 12–15
33. J. Plate, M. Tirtasana, R. Carmona, B. Fröhlich, Octreemizer: a hierarchical approach for interactive roaming through very large volumes, in *Proceedings of the Symposium on Data Visualisation*, VISSYM '02, Eurographics Association, 2002, pp. 53–ff
34. E.M. Lidal, T. Langeland, C. Giertsen, J. Grimsgaard, R. Helland, A decade of increased oil recovery in virtual reality. *IEEE Comput. Graphics Appl.* **27**(6), 94–97 (2007)
35. C. Andersen, A.-J. van Wijngaarden, Interpretation of 4d avo inversion results using rock physics templates and virtual reality visualization, North Sea examples, *SEG Annual Meeting*, September 2007
36. D. Patel, C. Giertsen, J. Thurmond, J. Gjelberg, M.E. Gröller, The seismic analyzer: interpreting and illustrating 2D seismic data. *IEEE Trans. Vis. Comput. Graph.* **14**(6), 1571–8 (2008)
37. D. Patel, S. Bruckner, I. Viola, M.E. Gröller, Seismic volume visualization for horizon extraction. *IEEE Pacific Visual.* **2010**, 73–80 (2010)
38. D. Patel, C. Giertsen, J. Thurmond, M. Gröller, Illustrative rendering of seismic data, in *Proceedings of Vision Modeling and Visualization*, 2007, pp. 13–22
39. C. Wijns, Inverse modelling in geology by interactive evolutionary computation. *J. Struct. Geol.* **25**(10), 1615–1621 (2003)

40. A. Guillen, P. Calcagno, G. Courrioux, A. Joly, P. Ledru, Geological modelling from field data and geological knowledge Part II. Modelling validation using gravity and magnetic data inversion. *Phys. Earth Planetary Interiors* **171**(1–4), 158–169 (2008)
41. G. Caumon, Towards stochastic time-varying geological modeling. *Math. Geosci.* **42**(5), 1–25 (2010)
42. F. Musgrave, C. Kolb, R. Mace, The synthesis and rendering of eroded fractal terrains. *ACM SIGGRAPH Comput. Graph.* **23**(3), 41–50 (1989)
43. B.B. Mandelbrot, *The Fractal Geometry of Nature* (W. H. Freedman and Co., New York, 1982)
44. V. Sapozhnikov, V. Nikora, Simple computer model of a fractal river network with fractal individual watercourses. *J. Phys. A: Math. Gen.* **26**, L623 (1993)
45. S. Stachniak, W. Stuerzlinger, An algorithm for automated fractal terrain deformation, in *Computer Graphics and Artificial Intelligence*, 2005
46. J. Doran, I. Parberry, Controlled procedural terrain generation using software agents. *IEEE Trans. Comput. Intell. AI Games* **2**(2), 111–119 (2010)
47. P. Roudier, B. Peroche, M. Perrin, Landscapes synthesis achieved through erosion and deposition process simulation. *Comput. Graph. Forum* **12**(3), 375–383 (1993)
48. N. Chiba, K. Muraoka, An erosion model based on velocity fields for the visual simulation of mountain scenery. *J. Visual.* **194** (1998)
49. J. Dorsey, A. Edelman, H. Jensen, J. Legakis, H. Pedersen, Modeling and rendering of weathered stone, in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, 1999, pp. 225–234
50. B. Benes, R. Forsbach, Layered data representation for visual simulation of terrain erosion, in *Spring Conference on Computer Graphics (SCCG)*, 2001
51. B. Benes, R. Forsbach, Visual simulation of hydraulic erosion. *J. WSCG* (2002)
52. B. Benes, V. Tesinsky, J. Hornys, S.K. Bhatia, Hydraulic erosion. *Comput. Animat. Virtual Worlds* **17**(2), 99–108 (2006)
53. O. Stava, B. Benes, M. Brisbin, J. Krivanek, Interactive terrain modeling using hydraulic erosion, in *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2008, pp. 201–210
54. R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.* **181**, 375 (1977)
55. L. Lucy, A numerical approach to the testing of the fission hypothesis. *Astron. J.* **82**, 1013 (1977)
56. G.J.P. de Carpentier, R. Bidarra, Interactive gpu-based procedural heightfield brushes, in *Proceedings of the 4th International Conference on Foundations of Digital Games*, FDG '09, (New York, NY, USA), ACM, 2009, pp. 55–62
57. G. Cordonnier, M.-P. Cani, B. Benes, J. Braun, E. Galin, Sculpting mountains: interactive terrain modeling based on subsurface geology. *IEEE Trans. Visual Comput. Graphics* **24**, 1756–1769 (2018)
58. G. Cordonnier, J. Braun, M.-P. Cani, B. Benes, E. Galin, A. Peytavie, E. Guérin, Large scale terrain generation from tectonic uplift and fluvial erosion. *Comput. Graph. Forum* **35**, 165–175 (2016)
59. G. Cordonnier, E. Galin, J. Gain, B. Benes, E. Guérin, A. Peytavie, M.-P. Cani, Authoring landscapes by combining ecosystem and terrain erosion simulation. *ACM Trans. Graph.* **36**, 134 (2017). The paper was presented at Siggraph 2017
60. E. Galin, E. Guérin, A. Peytavie, G. Cordonnier, M.-P. Cani, B. Benes, J. Gain, A review of digital terrain modeling. *Comput. Graph. Forum* **38**(2) (2019)
61. J.M. Cohen, J.F. Hughes, R.C. Zeleznik, Harold: a world made of drawings, in *Proceedings of NPAR '00*, 2000, pp. 83–90
62. N. Watanabe, T. Igarashi, A sketching interface for terrain modeling, in *ACM SIGGRAPH 2004 Posters*, 2004, p. 73
63. H. Zhou, J. Sun, G. Turk, J.M. Rehg, Terrain synthesis from digital elevation models. *IEEE Trans. Visual Comput. Graphics* **13**(4), 834–48 (2007)

64. M. Becher, M. Krone, G. Reina, T. Ertl, Feature-based volumetric terrain generation and decoration. *IEEE Trans. Visual Comput. Graphics* **25**, 1283–1296 (2019)
65. E. Guérin, J. Digne, E. Galin, A. Peytavie, C. Wolf, B. Benes, B. Martinez, Interactive example-based terrain authoring with conditional generative adversarial networks. *ACM Trans. Graph.* **36** (2017)
66. P. Isola, J. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks. *CoRR abs/1611.07004* (2016)
67. T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, B. Catanzaro, High-resolution image synthesis and semantic manipulation with conditional gans, 2017
68. E. Guérin, J. Digne, E. Galin, A. Peytavie, Sparse representation of terrains for procedural modeling. *Comput. Graph. Forum* **35**, 177–187 (2016)
69. A. Bernhardt, A. Maximo, L. Velho, H. Hnaidi, M.-P. Cani, Real-time terrain modeling using CPU-GPU coupled computation, in *XXIV SIBGRAPI*, (Maceio, Brazil), Aug 2011
70. M. Natali, T.G. Klausen, D. Patel, Sketch-based modelling and visualization of geological deposition. *Comput. Geosci.* **67**, 40–48 (2014)
71. M. Natali, J. Parulek, D. Patel, Rapid modelling of interactive geological illustrations with faults and compaction, in *Proceedings of the 30th Spring Conference on Computer Graphics, SCCG '14*, ACM, 2014
72. R. Amorim, E.V. Brazil, F. Samavati, M.C. Sousa, 3d geological modeling using sketches and annotations from geologic maps, in *SBIM '14*, ACM, 2014, pp. 17–25
73. D. Lopes, D. Mendes, M. Sousa, J. Jorge, Expeditious illustration of layer-cake models on and above a tactile surface. *Comput. Geosci.* **90**, 02 (2016)
74. K. Takayama, O. Sorkine, A. Nealen, T. Igarashi, Volumetric modeling with diffusion surfaces, in *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, vol. 29, no. 6, 2010
75. A. Orzan, A. Bousseau, H. Winnemöller, P. Barla, J. Thollot, D. Salesin, Diffusion curves: a vector representation for smooth-shaded images. *ACM Trans. Graph.* **27** (2008)
76. L. Wang, Y. Yu, K. Zhou, B. Guo, Multiscale vector volumes. *ACM Trans. Graph.* **30**, 167:1–167:8 (2011)
77. SIS, Petrel seismic interpretation software, in *Schlumberger Information Solutions*. <https://www.software.slb.com/products/petrel/petrel-geology-and-modeling>, Accessed Mar 2021
78. B.J. Kadlec, H.M. Tufo, G.A. Dorn, Knowledge-assisted visualization and segmentation of geologic features. *IEEE Comput. Graphics Appl.* **30**, 30–39 (2010)
79. D. Parks, Freeform modeling of faulted surfaces in seismic images. *SEG Tech. Prog. Exp. Abst.* **28**(1), 2702–2706 (2009)
80. M. Botsch, L. Kobbelt, An intuitive framework for real-time freeform modeling, in *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, 2004, pp. 630–634
81. R. Amorim, E.V. Brazil, D. Patel, M.C. Sousa, Sketch modeling of seismic horizons from uncertainty, in *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, SBIM '12, 2012, pp. 1–10
82. S. Motta, A. Montenegro, M. Gattass, D. Roehl, A 3d sketch-based formulation to model salt bodies from seismic data. *Comput. Geosci.* **142** (2020)
83. A. Nealen, O. Sorkine, M. Alexa, D. Cohen-Or, A sketch-based interface for detail-preserving mesh editing, SIGGRAPH, (NY, USA), Association for Computing Machinery, New York, 2005
84. R. Liu, L. Shen, X. Chen, G. Ji, B. Zhao, C. Tan, M. Su, Sketch-based slice interpretative visualization for stratigraphic data. *J. Imag. Sci. Technol.* (2019)
85. E.M. Lidal, H. Hauser, I. Viola, Geological storytelling—graphically exploring and communicating geological sketches, in *Proceedings of Sketch-Based Interfaces and Modeling (SBIM 2012)*, 2012, pp. 11–20
86. M. Garcia, M.-P. Cani, R. Ronfard, C. Gout, C. Perrenoud, Automatic generation of geological stories from a single sketch, in *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering. Expressive '18*, 2018

87. G. Laurent, G. Caumon, M. Jessell, Interactive editing of 3d geological structures and tectonic history sketching via a rigid element method. *Comput. Geosci.* **74** (2015)
88. J.-L. Mallet, Discrete smooth interpolation. *ACM Trans. Graph.* **8**, 121–144 (1989)
89. J.-L. Mallet, Discrete smooth interpolation in geometric modelling. *Comput. Aided Des.* **24**(4), 178–191 (1992)
90. J. Mallet, Discrete modeling for natural objects. *Math. Geol.* **29**(2), 199–219 (1997)
91. R. Sibson, A brief description of natural neighbor interpolation, in *Interpreting multivariate Data*, 1981, pp. 21–26
92. J.-P. Chilès, P. Delfiner, *Geostatistics: Modeling Spatial Uncertainty* (Wiley, 1999)
93. T. Viard, G. Caumon, B. Lévy, Adjacent versus coincident representations of geospatial uncertainty: which promote better decisions? *Comput. Geosci.* (2010)
94. J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum, T.R. Evans, Reconstruction and representation of 3d objects with radial basis functions, in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, (New York, NY, USA), ACM, 2001, pp. 67–76
95. Leapfrog, Leapfrog radial basis function. <https://www.seequent.com/comparing-leapfrog-radial-basis-function-and-kriging/>, 2020. Accessed Aug 2021
96. I. Macêdo, J.A.P. Gois, L. Velho, Hermite radial basis functions implicits. *Comput. Graph. Forum* **30**(1), 27–42 (2011)
97. P. Calcagno, J. Chilès, G. Courrioux, A. Guillen, Geological modelling from field data and geological knowledge Part I. Modelling method coupling 3D potential-field interpolation and geological rules. *Phys. Earth Planet. Int.* **171**(1–4), 147–157 (2008)
98. P. McNerney, A. Goldberg, P. Calcagno, G. Courrioux, A. Guillen, R. Seikel, Improved 3D geology modelling using an implicit function interpolator and forward modelling of potential field data. *Proc. Explor.* **7**, 919–922 (2007)
99. B. Wu, T. Rahman, X.-C. Tai, Sparse-data based 3d surface reconstruction for cartoon and map, in *Imaging, Vision and Learning Based on Optimization and PDEs*, ed. by X.-C. Tai, E. Bae, M. Lysaker (Springer International Publishing, 2018)
100. M. Floater, Y. Halbwachs, O. Hjelle, M. Reimers, Omega: a cad-based approach to geological modelling.,” GOCAD ENSG, Conference Proceedings, June 1998
101. N. Euler, C.H. Sword, J.-C. Dulac, A new tool to seal a 3d earth model: a cut with constraints, in *Proceedings of 68th Annual Meeting*, Society of Exploration Geophysicists, 1998, pp. 710–713
102. F. Aminzadeh, N. Burkhard, N. Nicoletis, F. Rocca, W. K. SEG-EAGE 3D modeling project, in *The Leading Edge*, 1994
103. C. De Boor, *A Practical Guide to Splines*. No. v. 27 in Applied Mathematical Sciences (Springer, 1978)
104. ArcGIS, Arcgis mapping and analytics platform, *ArcGIS Software*. <https://www.esri.com/en-us/arcgis/about-arcgis/overview>, Accessed Mar 2021
105. F.P. Preparata, M.I. Shamos, *Computational Geometry: An Introduction* (Springer, New York, 1985)
106. Web3D, Vrm197 functional specification, *VRML Standard*. <https://www.web3d.org/documents/specifications/14772/V2.0/index.html>, Accessed Mar 2021
107. L. feng Zhu, Z. He, X. Pan, X. cai Wu, An approach to computer modeling of geological faults in 3d and an application. *J. China Univ. Min. Technol.* **16**(4), 461–465 (2006)
108. Y. Halbwachs, O. Hjelle, Generalized maps in geological modeling: object-oriented design of topological kernels, in *Advances in Software Tools for Scientific Computing* (Springer, Berlin, 2002)
109. P. Lienhardt, Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Comput. Aided Des.* **23**, 59–82 (1991)
110. M. Apel, *A 3d geoscience information system framework*. PhD thesis, TU Freiberg, pp 16–31, 2004
111. S. Brandel, S. Schneider, M. Perrin, N. Guiard, J.-F. Rainaud, P. Lienhardt, Y. Bertrand, Automatic building of structured geological models, in *Proceedings of the ninth ACM Symposium on Solid Modeling and Applications*, SM '04, 2004, pp. 59–69

112. M. Perrin, B. Zhu, J.-F. Rainaud, S. Schneider, Knowledge-driven applications for geological modeling. *J. Petrol. Sci. Eng.* **47**, 89–104 (2005)
113. A.A. Pasko, V. Adzhiev, A. Sourin, V.V. Savchenko, Function representation in geometric modeling: concepts, implementation and applications. *Vis. Comput.* **11**(8), 429–446 (1995)
114. E. Kartasheva, V. Adzhiev, P. Comninos, O. Fryazinov, A. Pasko, Heterogeneous objects modelling and applications, in *An Implicit Complexes Framework for Heterogeneous Objects Modelling* (Springer, Berlin, 2008), pp. 1–41
115. M.N. Gamito, S.C. Maddock, Topological correction of hypertextured implicit surfaces for ray casting. *Vis. Comput.* **24**, 397–409 (2008)
116. P. Tahmasebi, *Multiple Point Statistics: A Review*. 06 2018
117. M.J. Abdollahifard, S. Ahmadi, Reconstruction of binary geological images using analytical edge and object models. *Comput. Geosci.* **89**, 239–251 (2016)
118. F. Wellmann, M. Lindsay, J. Pohb, M. Jessell, Validating 3-d structural models with geological knowledge for improved uncertainty evaluations. *Energy Procedia* **59**, 374–381 (2014)
119. G. Mariethoz, S. Lefebvre, Bridges between multiple-point geostatistics and texture synthesis: review and guidelines for future research. *Comput. Geosci.* **66**, 66–80 (2014)
120. T.-F. Zhang, P. Tilke, E. Dupont, L.-C. Zhu, L. Liang, W. Bailey, Generating geologically realistic 3d reservoir facies models using deep learning of sedimentary architecture with generative adversarial networks. *Pet. Sci.* **16**(3) (2019)
121. R. Schmidt, B. Wyvill, M.C. Sousa, J.A. Jorge, Shapeshop: sketch-based solid modeling with blobtrees, in *ACM SIGGRAPH 2006 Courses, SIGGRAPH '06* (NY, USA), ACM, New York, 2006
122. O. Karpenko, J.F. Hughes, R. Raskar, Free-form sketching with variational implicit surfaces. *Comput. Graph. Forum* **21**(3), 585–594 (2002)