



HAL
open science

Automatic Generation of Geological Stories from a Single Sketch

Maxime Garcia, Marie-Paule Cani, Rémi Ronfard, Claude Gout, Christian Perrenoud

► **To cite this version:**

Maxime Garcia, Marie-Paule Cani, Rémi Ronfard, Claude Gout, Christian Perrenoud. Automatic Generation of Geological Stories from a Single Sketch. Expressive '18 - The Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering, Aug 2018, Victoria, Canada. pp.17-19, 10.1145/3229147.3229161 . hal-01817923

HAL Id: hal-01817923

<https://inria.hal.science/hal-01817923>

Submitted on 18 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Automatic Generation of Geological Stories from a Single Sketch

Maxime Garcia
Univ. Grenoble Alpes, CNRS, Inria,
Grenoble INP*, LJK
38000 Grenoble, France

Marie-Paule Cani
Ecole Polytechnique, CNRS, LIX
9128, Palaiseau, France

Rémi Ronfard
Univ. Grenoble Alpes, CNRS, Inria,
Grenoble INP*, LJK
38000 Grenoble, France

Claude Gout
Total
F-64000 Pau France

Christian Perrenoud
Museum national d'Histoire naturelle,
UMR 7194 CNRS
France

ABSTRACT

Describing the history of a terrain from a vertical geological cross-section is an important problem in geology, called *geological restoration*. Designing the sequential evolution of the geometry is usually done manually, involving many trials and errors. In this work, we recast this problem as a storyboarding problem, where the different stages in the restoration are automatically generated as storyboard panels and displayed as geological stories. Our system allows geologists to interactively explore multiple scenarios by selecting plausible geological event sequences and backward simulating them at interactive rate, causing the terrain layers to be progressively un-deposited, un-eroded, un-compacted, un-folded and un-faulted. Storyboard sketches are generated along the way. When a restoration is complete, the storyboard panels can be used for automatically generating a forward animation of the terrain history, enabling quick visualization and validation of hypotheses. As a proof-of-concept, we describe how our system was used by geologists to restore and animate cross-sections in real examples at various spatial and temporal scales and with different levels of complexity, including the Chartreuse region in the French Alps.

CCS CONCEPTS

•Computing methodologies → Planning for deterministic actions; Physical simulation; •Applied computing → Environmental sciences;

KEYWORDS

Sketch-based animation, storytelling, geology, simulation

ACM Reference format:

Maxime Garcia, Marie-Paule Cani, Rémi Ronfard, Claude Gout, and Christian Perrenoud. 2018. Automatic Generation of Geological Stories from a Single Sketch. In *Proceedings of the Joint Symposium on Computational Aesthetics, Sketch-Based Interfaces and Modeling, and Non-Photorealistic Animation and Rendering*, Victoria, BC, Canada, Aug. 17–19 (EXPRESSIVE 2018), 15 pages.

DOI: 10.475/123.4

*Institute of Engineering Univ. Grenoble Alpes

1 INTRODUCTION

Hand-drawn sketches are the most extensively used mode of communication among geologists, and are also an easy way to convey some geological knowledge to the public (eg. [Fossen 2016]). They bring a simple, yet powerful way of expressing geological settings, enabling experts to illustrate, share and test hypotheses. Manually drawn sketches can be used to represent 2D vertical cross-sections of the layers of rocks composing the earth crust, and eventually help reconstructing 3D models of the subsurface using an interpolation between a collection of such sketches, or to appropriate annotations combined with maps of the actual terrain (eg. [Amorim et al. 2014]).

Figure 1 shows a sketch of a cross-section of the Chartreuse mountain range, a part of the French Alps, provided by geologists highlighting the different parts of the geological structure .

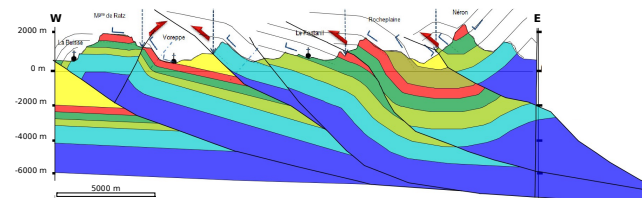


Figure 1: Our system generates geological stories from a single sketch such as this cross-section of the Chartreuse mountain. Layers (and maximum ages) are indicated with different colors: Yellow - Miocene (-23×10^6 years), Khaki - Late Cretaceous (-90×10^6 years), Red - Urgonian (-130×10^6 years), Green - Hauterivian (-133×10^6 years), Light green - Berriasian (-145×10^6 years), Light blue - Tithonian (-152×10^6 years), Blue - Oxfordian (-163×10^6 years).

Geometry and ages related colors of a single cross-section are however too limited for fully expressing and visualizing hypotheses on the past history of a terrain. In this work, we focus on the problem of **geological restoration** which consists in representing the geometry of the same 2D vertical slice of a terrain at different periods in time. Standard restoration requires manually creating a series of sketches, representing the hypotheses on the main steps of the terrain formation. More precisely, the process used by geologists consists in progressively undeforming the cross-section that represents their hypothesis on the current state of soil layers back to its initial state - when the existing geological layers

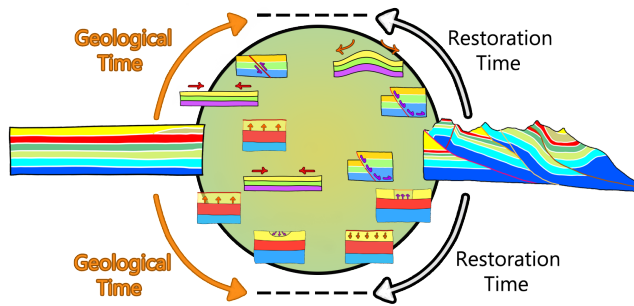


Figure 2: Cross section restoration process: From left to right, the original terrain from several million years ago undergoes events that will transform it to its current state. From right to left, the current terrain is restored and undergoes undo events that will transform it back to its original state. The bottom and top parts show different restoration paths.

were flat (deposition time of layers). The intermediate sketches are drawn by progressively deforming terrain layers to undo geological events such as erosion, sedimentation, folding and faulting (see Figure 7). The consistency of the result enables them to validate or to invalidate and improve the interpretation used to build the input cross-section sketch. Restoring a cross-section is a labor intensive task, which requires many trials and errors.

Geological restoration is a very challenging, ill-posed problem, due to the large number of unknown parameters and to the lack of ground truth data for validation. As a result, it is a relatively under-explored problem in academic research [Bulnes and McClay 1999; Dahlstrom 1969; Erickson et al. 2000; Groshong 1999; Yamada and McClay 2003]. It is nonetheless a topic with a high practical importance in applied geology for the exploration of hydrocarbons and rare materials. Several commercial tools have been proposed to assist geologists in geological restoration tasks [Exploration 2014; Halliburton 2009; Schlumberger 2011], using one geological cross-section and additional geological knowledge as input, depending on the technique used for partial restoration. They are either a support for mostly manual manipulation and editing, or provide off-line simulation mechanisms. In the last case, they model a single geological behavior (such as soil layers sliding over each other) and require knowledge of the physical properties of the layers, which may not be readily available. In a separate line of research, Lidal et al. have proposed a system allowing geologists to freely organize their sketches into storyboardlike graphical representations, called geological stories [Lster Lidal et al. 2012]. While their system can be used to illustrate the restoration process, the task remains labor-intensive since it requires geologists to draw every single step in the restoration process as a separate storyboard panel without a way to keep the geometrical consistency across storyboards.

The goal of our work is to provide an interactive tool enabling geologists to automatize most of their task during geological restoration. Our method can be used to generate one or several possible geological storyboard(s) from one single input cross section. The storyboard can then be used to generate a forward animation of the

terrain formation process, enabling geologists to quickly validate or refine their hypotheses.

Rather than providing a fully automatic method, which would require expressing the entire restoration process with precise rules, a seemingly impossible task, we instead propose an interactive tool enabling geologists to draw a cross-section, choose between geological events suggested by our system and undo each event in turn by running an approximate backward simulation of it. At each stage, the input cross-section is deformed and simplified until it becomes fully restored. More precisely, starting from a single vector graphics sketch and a small number of geological hypotheses such as the materials and ages of the different strata, our system automatically generates a list of geological events that need to be undone, partially orders them, and allows the user to iteratively choose the next event to undo at every stage of the restoration. Each of these undo event - including un-sedimenting, un-eroding, un-compacting, un-folding and un-faulting - is then simulated using a pseudo-physical backward simulation. At the end of the simulation, the new geological cross-section is stored as a possible step of the restoration storyboard. At each iterative step, the user can backtrack her choices and select other events to undo, exploring new restoration scenarios. This process ends when the user is satisfied with the scenario and there is no more event to undo. The resulting storyboard can be used to create a forward simulation explaining the formation of the input terrain, as we show on the complex example of the Chartreuse mountain range.

After reviewing related work (Section 2), we give a more detailed description of the cross section restoration process and an overview of our system (Section 3). We then describe how we extract and sort geological events from the input drawing (Section 4), how we iteratively update a story graph of geological events to be undone (Section 5) and how we generate keyframes of the geological storyboard with a backward pseudo-physical simulation (Section 6). We present experimental results in Section 7 and discuss limitations and directions for future work in Section 8.

2 RELATED WORK

2.1 Terrain and geological modeling

Terrain and geological modeling has been a topic of interest in Computer Graphics research for several decades [Natali et al. 2013]. While early work made use of fractals or noise models for modeling terrains [Musgravet et al. 1989; Olsen 2004], recent frameworks used geological knowledge and/or simulation of geological or hydrological laws to create compelling landscapes [Cordonnier et al. 2016; Gènevaux et al. 2013]. In parallel, sketch-based modeling techniques, either for terrain-surfaces [Tasse et al. 2014] or for full layer-cake models [Lopes et al. 2016; Ronan 2017] emerged as new trends for providing intuitive interfaces, making the modeling process more intuitive and efficient.

In contrast with these works, we are not tackling the problem of sketch-based terrain modeling, but of *sketch-based animation* of the evolution of a terrain over time, based on geological knowledge.

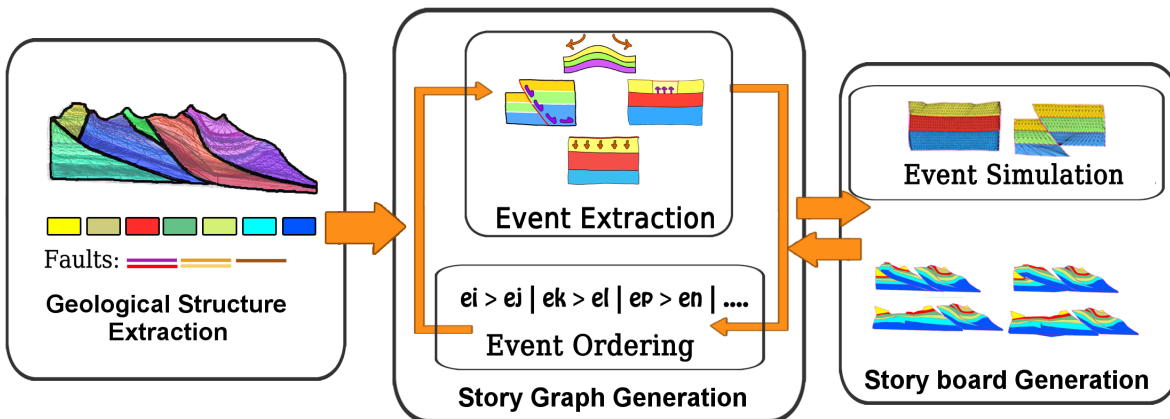


Figure 3: Process Overview (from left to right): (i) input sketch analysis for geological structure extraction; (ii) generation of branching geological story graph; (iii) Undo event simulation and storyboard sketch generation.

2.2 Geological restoration

Geological restoration has not been explored much in computer graphics research, but commercial tools exist to help geologists in this difficult task. These tools ease the consistent restoration of a specific terrain cross-section using either geometric constraints, physically-based simulation, or sketch-based interfaces.

Geometric based methods allow the user to directly manipulate the cross section of a terrain to undo geological events while maintaining consistency thanks to dedicated kinematic algorithms. Geologists can undo a large range of events - even complex ones such as isostasy and compaction. Move2D [Exploration 2014] uses this kind of technique to tackle the cross restoration problem and extend it to 2.5D using several slightly shifted cross sections to undo 3D events. However, this technique is not automatic and requires time and efforts from geologists. Typically, skilled geologists can spend up to one week to draw and restore a cross-section of the scale of the Chartreuse.

In contrast, physics-based methods run either forward or backwards mechanical simulations over a cross section using Coulomb theory and finite elements schemes in order to simulate a wide range of geological phenomena while preserving mechanical consistence. Forward simulation [Mary 2012] takes a hypothetic restoration of the current terrain as input and run a simulation over it to enable comparison with the current terrain, and validate or unvalidate the restoration. On the opposite side of the spectrum, Dynel2D [Schlumberger 2011] runs a backward simulation on the terrain's current cross-section by inverting the way forces were supposedly applied to the original terrain state. While these methods are automatic, they are compute-intensive, which prevents the user from interacting with the simulation. Moreover, they tackle a single geological event and require precise knowledge of the material and mechanical properties of the terrain.

Closer to computer graphics research, Lidal and al. [Lster Lidal et al. 2012] proposed a sketch based method for visualizing cross

section restoration, enabling to organize sketches into story trees. Starting from an initial sketch, geologists draw several hypotheses on different branches of the tree, the last sketch being displayed with semi-transparent opacity to help maintaining geometric consistence. This method reduces the time spent drawing cross-sections and gives geologists the opportunity to explore multiple scenarios more easily. The method remains labor-intensive since every step of every restoration hypothesis needs to be drawn by hand. It is nonetheless attractive to geologists for quickly comparing and communicating restoration scenarios.

2.3 Automatic story-boarding

Automatic story-boarding, ie. automatically generating storyboards or story lines from natural language or image sequences or videos is not a new idea. Previous work has described interactive methods for generating storyboards from existing movies [Goldman et al. 2006] using computer vision techniques. Others have attempted to automatically generate animation storyboards from movie scripts [Ye and Baldwin 2008] using natural language processing. The Darkshak system generates storyboards from a given story and a 3D environment [Jhala and Young 2015] and uses them to plan camera shot sequences for interactive storytelling environment. Other researchers have used a 3D environment and the relation between characters to generate a structured and paced story [Dominguez et al. 2015]. Our geological story graphs are most similar to the branching story graphs introduced by Riedl and Young [Riedl and Young 2006] who use classical partial order planning methods for dynamically piecing together storyboard-like sequences in response to user events in narrative games. Borrowing from their previous work, we dynamically generate geological story graphs in response to interactive event sequencing hypotheses from the geologist. In contrast with previous sketch-based restoration methods, the user does not need to draw storyboard panels one by one and manually maintain their consistency.

3 SYSTEM OVERVIEW

Figure 3 shows a representation of our storyboarding and animation process, which involves three main parts: (i) a sketch interpretation stage parses the input cross-section into meaningful geological structures, and recovers the list of geological events that needs to be undone for restoring the cross section (Section 4); (ii) a story graph generation stage offers choices of events to be undone to the geologist and updates the list of events until the sketch is fully restored (Section 5); (iii) a pseudo-physical simulation stage is used to undo geological events and build novel keyframes into the restoration storyboard (Section 6). Steps (ii) and (iii) are repeated until the cross-section is restored, as represented in Figure 3.

In this process, the geologist is allowed to backtrack from his previous choices and explore alternative restoration scenarios, following a branching story graph as defined by [Riedl and Young 2006]. As a result, one or more complete paths may be generated, each one corresponding to a different possible geological history. Using keyframes along each path, we finally recompute animations, this time in direct chronological order, in order to explain and illustrate all the possible histories (see the accompanying video for details and examples).

4 SKETCH INTERPRETATION

4.1 Structural depiction of subsurface geology

Geological cross sections are geometric descriptions of the distribution of rock material in a terrain, usually defined by their geological age interval. They are structured in a way that geologists are able to restore them within one or several annotated sketches. They divide the terrain into several geological elements delimited by lines and curves, in particular layers, blocks, faults and units. A layer represents a geological surface made of a single material identified by its age and represented by colored areas (one specific color for each layer). Faults represents breaks dividing the cross sections into blocks. Each portion of layer contained in a block is called a geological unit. Geologists distinguish three types of faults: normal, reverse or strike-slip [Fossen 2016]. Normal and reverse faults correspond to mostly vertically oriented breaks which make blocks slide relative to each other. Normal faults occur during extension while reverse faults happen during compression. Strike-slip faults, where the offset is predominantly horizontal, are a more difficult case that cannot be easily restored in a 2D context. We therefore focus on reverse and normal faults.

This structural choice fits the cross section restoration process because, at its birth, any terrain is composed of only one block containing one layer. Therefore, restoring a cross section consists in transforming, removing and merging its units over time, until only one layer contained into a single block remains.

These transformations are equivalent to undoing geological events that occurred during the history of the cross-section. In our system, we consider the five most frequent geological events which can be represented in the 2D case: sedimentation, compaction, erosion, faulting and folding. The effect of each event is presented in Figure 4. Our system includes mechanisms for automatically detecting these events as well as automatically undoing them.

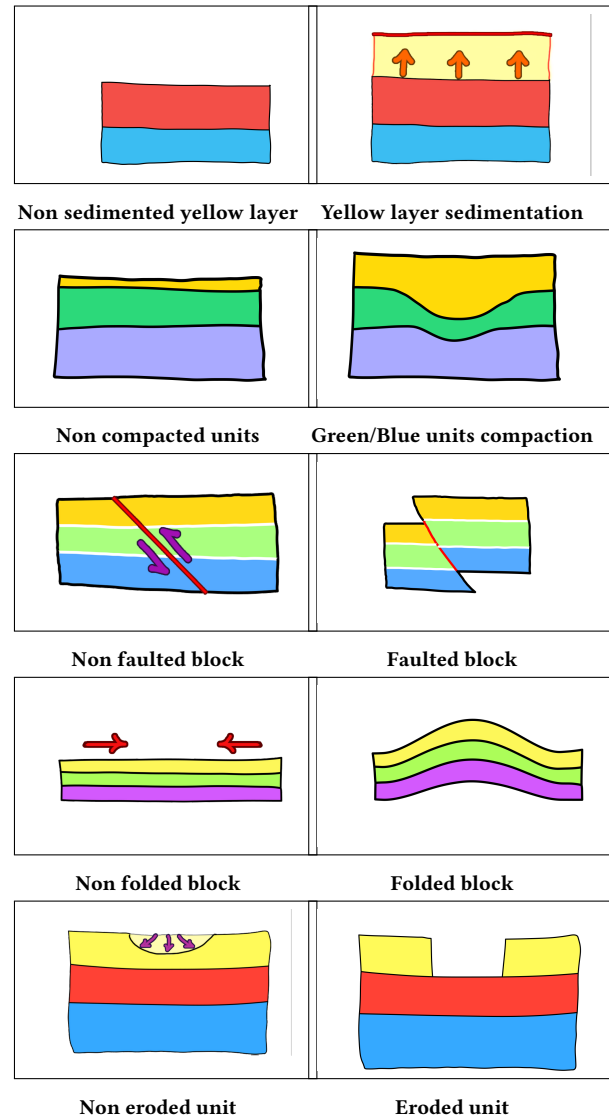


Figure 4: Geological events effects. During sedimentation, layers deposit on the ground horizontally; here the yellow layer deposits on the red ones. Compaction arises when sediments are loaded on underlying layers; the green and blue layers are being compacted while the yellow layer is sedimenting. Erosion removes units sediments on the contact surface with air or water. Folds occur during terrain compression where blocks start to deform and can reach critical points where they break, giving birth to faults which makes block sliding against each other.

4.2 Geological structure extraction

The first step of our multi-scenario restoration system is to extract a geological structure from a hand-drawn cross-section. More precisely, we extract the structure in terms of blocks, layers, faults and units.

To do so, the input sketch is drawn using the Vpaint vector graphics software¹ and stored as a Vector Graphics Complex (VGC) [Dalstein et al. 2014] which is adapted to our cross section case because it can represent arbitrarily complex topological relations between faces (blocks, units and layers), edges (faults and boundaries) and vertices. Additionally, we can store color-based annotations on vertices, edges and faces which contribute to the geological structure extraction. Units are represented as colored faces, where each unit of the same layer has the same color. Furthermore, we color in white shared edges between each unit of the same block enabling to identify blocks as a set of units with only white shared edges. Therefore, faults are colored in any color other than white and black, clearly dividing neighboring units into different blocks.

Finally, the user specifies the age of each layer, giving to our system temporal information about sedimentation events and consequently about events occurring between two sedimentations. By using all this information and by reading our VGC file starting from the youngest units, we can easily build a geological structure with blocks containing units and identifying where the faults are located. Figure 3 (left) shows the structure we extract from an input VGC of the Chartreuse cross-section inspired from Figure 1.

Using the previous extracted structure, we compute the state of all units, layers and blocks, which together summarize the state of the terrain in the cross-section. All layers in the cross-section are marked as "sedimented", and the top layer is marked as "non-covered" while all other layers are marked as "covered". Units adjacent to a fault are marked as either "left-faulted" or "right-faulted". Other units are marked as "non-faulted". Units with constant thickness are marked as "non-eroded" while units with non-constant thickness are marked as "eroded". Disconnected units from the same layer are also marked as "left-eroded" or "right-eroded" (in this case the block unit has been eroded to the point that it as been divided into several sub-units). Units with a torsion energy - as defined in Section 6 - above a threshold are marked as "folded" while units under the same threshold are marked as "non-folded".

5 STORY GRAPH GENERATION

In this Section we describe the interactive process of generating and updating a story graph of possible restoration steps, which the geologist can choose, view and validate.

We formulate restoration as a classical partial order planning problem, where the possible actions are the undo events inferred from the cross-section, the starting state is the present state of the terrain as depicted in the cross-section, and the goal state is the restored cross-section consisting of the oldest layer alone, in its original condition (non-folded, non-faulted, non-eroded and non-compacted).

A partial ordered plan (POP) is a quadruplet (A, P_o, L, B_v) where A is a set of actions that progressively lead to a predefined goal, P_o is a partial order of the action set, L is a set of causal links defining action consequences with respect to other actions preconditions and B_v is a set of variable bindings defining a set variable constraints. Previous works has proposed a variety of algorithms for automatically generating partial order plans and total order plans [Russell

and Norvig 2003]. In our case, we use a non deterministic approach where a user will successively execute actions whose preconditions are satisfied until she either reaches the expected goal, or reaches a threat (failure case). In the last case the user can backtrack from her previous choices and explore a new plan.

To solve this problem efficiently, we make the assumption that geological history is an invertible planning problem, in the sense that each geological event can be inverted with a corresponding undo event. We use the definition of invertible plans from Koehler and Hoffman [Koehler and Hoffmann 2000], who state that a planning problem is invertible iff $\forall s, a \exists a^{-1} result(result(s, a), a^{-1}) = s$. In other words, for all states and actions, there is inverse action that returns the world to its previous state.

In terms of planning operators, we represent direct geological events and inverse undo events using the classical STRIPS representation [Fikes and Nilsson 1971], which consists of a list of preconditions (describing necessary conditions on the state of the world for the event to take place) and a list of effects (describing how the state of the world changes as a result of the event). Effects are described with a list of facts that must be added to the state of the world and a list of facts that must be deleted from the state of the world. If a direct event has preconditions $pre(e)$ and effects $add(e)$ and $del(e)$ then the corresponding inverse event has preconditions $pre(e) + add(e) - del(e)$ and effects $add(e^{-1}) = del(e)$ and $del(e^{-1}) = add(e)$. This allows us to easily derive preconditions and effects for all undo events using available geological knowledge. See Figure 4 for details.

As a result, we can find a partial order plan for the restoration problem using progressive state-search (in the direction of restoration time), as in classical STRIPS planning [Fikes and Nilsson 1971] where we iteratively choose available undo events whose preconditions are satisfied, and update the state of the restoration accordingly. Computing a complete story graph of all available story plans in this fashion would be possible. But for our purpose, we rather allow the geologist to choose at each step the next undo event from a short list, enabling her to generate one path in the story graph at a time. In some cases, the geologist may choose to backtrack to a previous state and explore a different path. Our system will then generate a story tree with a (small) subset of the entire story graph. Note that the resulting story trees are similar to the ones that would be drawn by hand using previous methods [Ister Lidal et al. 2012], except that they run backwards in time. We show examples of story trees generated by our system in Figure 11.

5.1 Event generation

In our current implementation, we are able to simulate the inverse of 5 geological events : sedimentation, folding, faulting, compaction and erosion. Considering this set of events, we build the set of undoable phenomena that lead a given cross-section to its current state.

More precisely, we create un-sediment events for all layers, un-fold events for all folded units, un-fault events for all left-faulted and right-faulted blocks, un-erode events for eroded units and un-compact events for all compacted units (we consider all covered units as being compacted and let the user evaluate the plausibility of a compaction when un-sedimenting covering layers).

¹<https://www.vpaint.org/>

Our system first adds an un-sediment event to action set for each different layer contained in the cross section. Then, for each newly un-sediment event added, we add a partial order link from the considered event to each of the older un-sediment event which is deduced from the layer ages. The next step, consist in adding other events and complete the list of partial time and causal ordering constraints progressively. Faulting events are directly deduced from the VGC sketch where faults are highlighted and divide the slice into blocks annotated cross section which highlight faults which divide the slice into blocks. As faulting events split one module into two, we create a time ordering constraint from each faults event whose faults divide modules exactly into two parts to faults which divide them into more than two parts. This ordering can be interpreted as un-faulting faults that are adjacent to two blocks or contained inside one block before others.

Folded event are extracted by checking for each block if at least one of its unit is folded. If that is the case, our system adds a fold event in the POP as well as a time constraint with all sedimentation events that occurred after the youngest folded unit.

5.2 Event ordering

After creating the undo events, we create a partial temporal order between them based on layer ages, level of erosion and faults configuration. Indeed, younger layers obviously sedimented after older ones and erosion of units occurred after the sedimentation of the layer they are part of.

Furthermore, geologically speaking, faults occurs only between two blocks, one sliding against the other. This feature allows our system to order faults events where those which are dividing the slice in only two blocks will be undone before the others. Furthermore, we can establish a correlation between un-sediment events and un-fault ones by noticing that faults can appear to not fully divide a block into two, with higher units being laterally continuous (not divided into several parts); this situation implies that the fault event occurred before the sedimentation of the lowest united unit.

Using those time dependencies, we are able to generate, at a given sketch state, an ordered list of plausible event to undo where events with no time dependencies are placed to the front of the list while the others (less plausible at the current state of the sketch) are placed to the back. The ordering will be pushed further in the next section, when iteratively restoring the cross-section, by giving more hints to the user before choosing an event to undo.

Furthermore, as our system only treats events sequentially and not in parallel, we impose a time dependency between un-fault and un-erode events; before un-faulting a block which contains several units of the same layer, un-erode events should be applied before in order to trigger the un-fault event with a more stable structure. At this stage, our system proposes the same ordered list as in section 4 but is more adapted to our looping process as it is updated using a partially-ordered plan structure.

Because events are only partially ordered, there is usually a large number of choices offered to the geologist at any point in time. To make this choice easier, we rank the possible events according to their geological plausibility. Thus, our system sorts the most plausible events using heuristic evaluation functions, which we now describe.

In order to evaluate un-faults events we compute the side length of each corresponding units (same layer) of the two concerned blocks. The smaller the sum of the differences of each corresponding unit side length is the greater is the probability for the un-fault event to occur. We justify this function by the fact that most of the time, when a fault is closing, it is done almost instantly with few unit height change. Un-sedimentation event evaluation function for one layer returns a probability of 1 the slice contains only one unit (thus one layer) of the considered layer and 0 otherwise. This function is equivalent to testing if the layer has been broken or highly eroded. Additionally this function returns 0 if the upper bound of the layer is folded because sediments always deposit in a flat manner.

Un-erode event are evaluated using the amount of lacking height a unit needs to have a constant height over all of its surface and which correspond to the eight of its corresponding layer itself. If a unit has been divided into two or more parts, this function return 1. Finally un-folding events are measured using the sum of torsion energy accumulated by each unit of a block (see Section 6). The higher the energy, the higher the probability.

When an event is selected, we run a backward physical simulation of it, as described in Section 6. At the end of the simulation, our system stores the resulting sketch as a keyframe in the storyboard. At the same time, the undo event list is updated, removing events that cannot occur anymore because of terrain changes, and adding new ones by running a new sketch analysis similar to the one described in Section 4. Finally, the process loops to the next event choice.

At this point, the user is given the opportunity to cancel previous undo events and backtrack to an alternative event choice. This feature enables geologists to explore multiple restoration scenarios, building a story tree where each node represents a storyboard panel and each tree path represents a restoration storyboard. The iterative process ends when the user is satisfied with one or several restoration scenarios. Figure 11 shows examples of generated story trees containing one complete restoration scenario and several incomplete alternatives proposed at different steps.

6 BACKWARD PHYSICAL SIMULATION

At each step in the restoration process, we perform a backward simulation which takes the current state of the cross-section and updates it to take into account the effects of the corresponding event.

6.1 Mass-spring system

We consider the terrain as a deformed elastic system during each event, and compute its equilibrium state by simulating a mass spring system on each block of the cross-section. Mass-spring systems are more versatile and faster to solve than the more common approaches used in geological restoration, such as finite element methods (FEM) and boundary element methods (BEM). This allows to run each simulation at interactive rate and to present the results to the geologist in real-time.

We map mass-spring systems to all blocks using rectangular grids since the final shape of the slice must be rectangular or trapezoidal. We add torsion springs on all vertices, with the effect of

Direct operators	Erode(Unit)	Fold(Block)	Compact(Unit)	FaultAndSlide (Left,Right)	Sediment(Top,Bottom)
Preconditions	Sedimented(Unit), NonEroded(Unit), NonCovered(Unit)	Sedimented(Block), NonFolded(Block)	Sedimented(Unit), Covered(Unit), NonCompacted(Unit)	Sedimented(Left), Sedimented(Right), NonRightFaulted(Left), NonLeftFaulted(Right)	NonSedimented(Top), Sedimented(Bottom), NonCovered(Bottom)
Add	Eroded(Unit)	Folded(Block)	Compacted(Unit)	RightFaulted(Left), LeftFaulted(Right)	Sedimented(Top), NonCovered(Top), NonFolded(Top), NonEroded(Top), NonCompacted(Top), NonFaulted(Top), Covered(Bottom)
Delete	Noneroded(Unit)	NonFolded(Block)	NonCompacted(Unit)	NonRightFaulted(Left), NonLeftFaulted(Right)	NonCovered(Bottom)
Inverse operators	Un-Erode(Unit)	Un-Fold(Block)	Un-Compact(Unit)	Un-FaultAndSlide(Left,Right)	Un-Sediment(Top,Bottom)
Preconditions	Sedimented(Unit), NonCovered(Unit), Eroded(Unit)	Sedimented(Block), Folded(Block)	Sedimented(Unit), Covered(Unit), Compacted(Unit)	Sedimented(Left), Sedimented(Right), RightFaulted(Left), LeftFaulted(Right)	Sedimented(Top), Noncovered(Top), NonFolded(Top), NonEroded(Top), NonCompacted(Top), NonFaulted(Top), Covered(Bottom)
Add	NonEroded(Unit)	NonFolded(Block)	NonCompacted(Unit)	NonRightFaulted(Left), NonLeftFaulted(Right)	NonSedimented(Top), NonCovered(Bottom)
Delete	Eroded(Unit)	Folded(Block)	Compacted(Unit)	RightFaulted(Left), LeftFaulted(Right)	Sedimented(Top), Covered(Bottom)

Figure 5: Direct and inverse geological events, with their pre-conditions and effects.

un-folding the blocks by applying a torque (torsion force) around the z-axis (pointing in front of the screen) and making horizontal (respectively vertical) springs reach an horizontal (respectively vertical) direction.

We build the mapping of each geological unit to a mass-spring system in three steps: first, we put masses at the borders in a uniform way; we then compute Hermite curves between each pair mass of the two level boundaries and place masses uniformly on those curves; finally, we link neighboring masses with torsion springs. The resulting network is illustrated in Figure 6 .

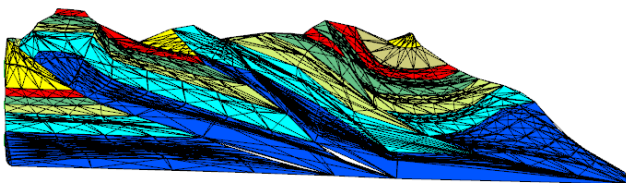


Figure 6: Chartreuse mass-spring mapping

Using this common framework, we are able to simulate the different geological events as follows.

6.1.1 *Un-Sediment*. Un-sedimenting a layer is simulated by decreasing gradually, depending on how long the material sedimented, the equilibrium length of vertical and shear springs until it reaches 0. Horizontal springs are preserved in order to keep the layer flat during the simulation.

6.1.2 *Un-Erode*. Un-eroding a unit correspond to gradually increasing the equilibrium length of vertical and shear springs of parts whose height is lower than the final height of the corresponding layer. This final height is computed when importing the initial sketch; it corresponds to the the maximum height of each corresponding layer’s units.

6.1.3 *Un-Fold*. Un-folding is a event which is applied to blocks and is easily simulated by activating the torsion force of the concerned block as by construction this force un-fold the block and give it its original trapezoidal shape.

6.1.4 *Un-Compact*. The Un-compact operator reverses the effect of the weight of the upper units on the lower units, resulting in thickness variations. During the Un-Compaction of one unit, we change the rest-length of the vertical springs to match the average thickness of the unit. In addition, we also change the rest length of the horizontal springs to match the *upper left corner-upper right corner* distance of the unit, which removes unwanted bulges and cavities in the unit.

6.1.5 *Un-Fault*. Un-fault is a two-step event composed on a sliding sub-event and a merging sub-event. Our system first simulates the sliding event by making one block slide relative to the other. To simulate this effect we run the physical loop described in Section 6, apply a compression or extension force (depends on the slice) on the sliding block and compute collision with others. When the corresponding units of the two blocks approximately face each other, we merge them. This merging occurs nearly at the same time for all corresponding units.

6.2 Numerical solution

In terms of physical equations, for a single mass spring system, we use an implicit Euler scheme modified using a method proposed by Desbrun et al. [Desbrun et al. 1999]. This is more stable than an explicit scheme and makes it possible to use a larger time step ($dt = 0.02$).

All particles in our mass-spring system have the same mass m and are described at step n with a position x_i^n , a velocity v_i^n , and an applied force F_i^n (containing all filtered, corrected and external forces).

At each time step we solve the second Newton's law:

$$v_i^{n+1} = v_i^n + F_i^{n+1} \frac{dt}{m} \quad (1)$$

$$x_i^{n+1} = x_i^n + v_i^{n+1} dt \quad (2)$$

with:

$$F_i^{n+1} = - \sum_{j \in Neighb(i)} k_{ij}(x_i - x_j) + k_{ij} l_{ij}^0 \frac{(x_i - x_j)}{\|(x_i - x_j)\|}$$

Those equations cannot be integrated as such because forces at step $n+1$ are unknown as we don't know the position of the masses at this step. Instead, we use a first order approximation to solve the problem at step $n+1$:

$$F_i^{n+1} = F_i^n + \frac{\partial F}{\partial x}(x_i^{n+1} - x_i^n)$$

Thus we need to compute $J = \frac{\partial F}{\partial x}$ which is the Jacobian of F . By using (1) (2) and (4) we obtain the following system:

$$\begin{aligned} (x_i^{n+1} - x_i^n) &= (v_i^n + (v_i^{n+1} - v_i^n)) dt \\ (v_i^{n+1} - v_i^n) &= (I - \frac{dt^2}{m} J)^{-1} (F_i^n + dt J v_i^n) \frac{dt}{m} \end{aligned}$$

We now have approximated our equations into a solvable system and we can notice that an additional force came from this approximation $dt J v_i^n$. It is an implicit viscosity that takes into account the movement of the neighboring particles. Consequently we have for each particle i a new force:

$$\tilde{F}_i = k dt \sum_{j \in Neighb(i)} (v_j - v_i)$$

Following [Desbrun et al. 1999], we approximate the Jacobian J by integrating the linear part of the elastic force:

$$F_{(i,j)} = -k_{ij}(x_i - x_j) + k_{ij} l_{ij}^0 \frac{(x_i - x_j)}{\|(x_i - x_j)\|}$$

Consequently, J represents the Jacobian of $F = -k_{ij}(x_i - x_j)$ where:

$$\begin{cases} J_{ij} &= k_{ij} \text{ if } i \neq j \\ J_{ii} &= - \sum_{j \neq i} k_{ij} \end{cases}$$

Integrating only the linear part implies that we will have some error at the end of the integration. However we can notice that the non linear part $k_{ij} l_{ij}^0 \frac{(x_i - x_j)}{\|(x_i - x_j)\|}$ has a constant magnitude during the simulation between two steps. Consequently, this force only

creates a rotational movement that we compensate with another force. Thus, the induced angular momentum δT is compensated with correcting forces:

$$\begin{aligned} F_i^{corrected} &= (x_G - x_i) \wedge \delta T dt \\ \delta T &= \sum_{i=1}^n (x_G - x_i) \wedge \sum_{i=1}^n F_{ji} W_{ij} \end{aligned}$$

In addition, we introduce a torsion force creating an angular moment towards a desired angle which gives our simulation a better control over units deformation:

$$F_{ij}^{torsion} = -k_{torsion} \Delta \theta \vec{n}$$

where $\Delta \theta$ is the angle between a unit vector depending on the spring type (Vertical, Horizontal, DownShear or UpShear) and the vector $x_i - x_j$. \vec{n} is the normal vector to $x_i - x_j$ pointing toward the unit vector. In addition, from this torsion force we can extract a deformation energy $E = \frac{1}{2} * (k_{torsion} \Delta \theta)^2$ which can give to the user of our system, hints about the success rate of the simulation where highly deformed units are not desired.

7 EXPERIMENTAL RESULTS

We applied our method to three examples of very different space and time scales: Caune de l'Arago, Tautavel Fields and Chartreuse mountains.

These examples were created and restored by two professional geologists, respectively working in geological research and in the oil industry, using our prototype system. These two experts users are co-authors of this paper. In each case, they were familiar enough with the geology of the tested terrain to attest the plausibility of the choices for the next undo event provided by our system and to have their own hypotheses to express on the most plausible history of the site. For each example we show a complete restoration storyboard (Figures 8, 9 and 10) as well as the corresponding story tree (Figure 11).

7.1 Caune de l'Arago

Caune de l'Arago is a cave located in the South of France yielding the oldest to date French human remains. We simplified the original 8 meters wide and 6.5 meters high cross section into nine layers, which illustrate the sedimentation and its evolution since 550 ka.

Using our interactive system, our expert user was able to easily build a consistent story of the site in ten steps, namely nine periods of sedimentation - six of them taking place in conjunction with compaction phases, plus one erosion episode.

Figure 8 shows the storyboard of the cave output by our system. The initial sketch was drawn in 30 minutes using the *Vpaint* software. Both the storyboard and a forward animation (see the attached video) were generated in only 11 minutes using our semi-automatic restoration process. The user only had to successively validate the undo events automatically proposed by the system and to specify his hypotheses on the de-compaction phases. In comparison, restoring the cross-section by drawing each storyboard sketch would have taken 5 hours on standard software spending 30 minutes per step sketch, without any means to generate a continuous, forwards animation of the history of the site. While he pointed out that the

top layer representing the current state of sedimentation should have been flat at all stages, the expert user was very satisfied with the use of our system, enabling him to easily express his knowledge on the history of the site and to visually explain it to collaborators and to the public in a few moments.

7.2 Tautavel Fields

In a second experiment with the same expert user, we tested our method on the Tautavel fields, the region where the Caune de l'Arago is located. This terrain, whose cross-section is 5.7 km wide and 1 km high, is composed of five main layers of sediments and is fractured by three faults.

This example is more complex than the Caune de l'Arago since the three independent faults lead to several possible choices of chronological order for the undo events. Using our system, the expert user was able to restore the cross-section in 9 steps. Drawing the cross-section with *Vpaint* took him 40 minutes. Exploring the different scenarios described in Figure 11 (middle) while taking into account existing geological knowledge to select the most plausible sequence of events, took another 40 minutes. In total, we created and restored the cross-section in 1 hour and 20 minutes, resulting into both the storyboard in Figure 9 and a full animation sequence. The user estimated the time needed to draw the entire storyboard by hand to approximately 14 hours and the time needed to restore the cross-section using the available digital modeling and simulation software to three days. Although our results may not be as accurate as with standard software, this second experiment validated the usefulness of our tool to quickly express and visually test geological hypotheses.

7.3 Chartreuse

Finally, we applied our method to a cross-section of the Chartreuse mountain range as shown in Figure 1, with the help of a second expert user, and experienced geologist working in the industry. Chartreuse is the largest scale and most complex example we tested with our prototype system. The selected cross-section represents the actual geometry of the subsurface structure, here widely outcropping in the "Cluse de l'Isère", downstream of Grenoble city, as a stack of rock layers, highly deformed by the alpine tectonic compression between Oligocene (30 Ma) and present time [Philippe et al. 1998]. This cross-section is 15 km wide and 5 km high and is composed of seven sediment layers which deposited from the Jurassic to the Neogene period.

For simplicity reasons, we divided the cross-section into the west and the east sides on the range, which can be restored independently. The west side is composed of six layers, three eroded units and two faults while the east side is composed seven layers, two eroded units and one fault. In total, the full Chartreuse example was restored using 17 undo events. Again, our system enabled the geologist expert to explore several consistent stories and to express his hypotheses. See Figure 11 and the accompanying video for an illustration of the full restoration process, where we assembled back the results for the two sides of the range.

Drawing the full Chartreuse cross-section using *Vpaint* took one hour. The story graph generation took another three hours. As a result, the storyboard and the associated animation sequence were

produced in a total of four hours. Being familiar with industrial restoration software, the user asserted that it would have taken one week to produce the same results from the initial sketch drawing using an existing restoration software. More precisely, drawing the initial sketch could take up to one day, gathering data necessary to run an accurate physical simulation would have taken 4 days while the restoration stage using such software would have taken 2 days of trials and errors. In addition, it would have taken approximately 16 hours to draw the storyboard frame by frame. The second expert user was therefore really happy with the use of our system, enabling him to quickly express hypotheses and to validate them visually. He noted that the generated animation could be useful for communication purposes as well, such as for teaching geology.

8 LIMITATIONS AND FUTURE WORK

The combination of interactive storytelling techniques with physical simulation provides an effective approach for quickly and systematically generating geological cross sections storyboards from a single input VGC sketch. As observed on different real cases, we were able to generate restoration scenarios at least seven times faster using our system than using traditional means. However our method comes with several limitations, which we would like to address as future work. Firstly, even though our backward simulation method is general and robust enough as a tool to interactively generate consistent storyboards, more constraints could be added to improve visual plausibility, such as imposing that sedimentation events always results in a horizontal top layers, which our current implementation with soft physically-based constraint cannot achieve. Moreover, our system does not guarantee to produce geologically consistent animations when played back in the opposite (forward) direction. Better results could be obtained by recomputing a new simulation in the opposite (forward) time direction, using the storyboard sketches as boundary conditions. Since this final step does not require user interaction, we could use more sophisticated methods to produce more convincing results. Alternatively, we could use the space-time interpolation method readily available in the *Vpaint* software to directly produce animations in the Vector Graphics Animation format [Dalstein et al. 2015].

As most often done in geological restoration, we assume that the geological history of the terrain is a sequence of invertible events, which is indeed a simplification of the irreversible processes at work in geodynamics. As a result, coping with simultaneous events and parallel geometric processes which are common ground in geology is difficult. Our current prototype system only allows for simultaneous de-sedimentation and de-compaction processes, showing that jointly modeling simultaneous event would be possible within our framework. Lastly, our system is currently limited to five geological event types. More complex geological events such as strike-slip faults should also be taken in consideration in future work. This is difficult because they typically require knowledge of the local 3D structure of the terrain. A possible extension of our work would use at least two offset cross sections of the same terrain to take into account horizontal 3D events. Alternatively, a top view could be used to reconstruct the 3D structure of the terrain as in [Amorim et al. 2014].

9 CONCLUSION

We have proposed a novel approach for automatically generating complete geological stories from a single sketch. Our work bridges the gap between traditional geological restoration methods and the emerging approaches for geological storytelling. Our interactive solution combines artificial intelligence planning techniques with physically-based simulation methods to enable the user to systematically, yet quickly explore the space of all possible geological stories compatible with a given input sketch, and express their hypotheses. Our early prototype was tested on three real examples of geological restoration problems at different time and space scales, and was found to be expressive and responsive enough to guide the geologist towards the depiction of a plausible geological history in each case.

ACKNOWLEDGMENTS

The work was supported by the European Research Council advanced grant EXPRESSIVE (ERC-2011-ADG 20110209) and a grant by TOTAL FRANCE. Special thanks to Marie-Christine CACAS-STENTZ, Tristan CORNU and Sebastien SCHNEIDER for their advice and encouragement, and Estelle CHARLEROY for her animation of the Chartreuse mountain in the accompanying video.

REFERENCES

- Ronan Amorim, Emilio Vital Brazil, Faramarz Samavati, and Mario Costa Sousa. 2014. 3D geological modeling using sketches and annotations from geologic maps. *Sketch-Based Interfaces and Modeling* (2014).
- M. Bulnes and K. McClay. 1999. Benefits and limitations of different 2d algorithms used in cross-section restoration of inverted extensional faults: application to physical experiments. *Tectonophysics* 312 (1999), 175–189.
- Guillaume Cordonnier, Jean Braun, Marie-Paule Cani, Bedrich Benes, Eric Galin, Adrien Peytavie, and Eric Guérin. 2016. Large Scale Terrain Generation from Tectonic Uplift and Fluvial Erosion. *Computer Graphics Forum* 35, 2 (May 2016), 165–175.
- C.D.A. Dahlstrom. 1969. Balanced cross sections. *Canadian Journal of Earth Sciences* 6 (1969), 743–757.
- Boris Dalstein, Remi Ronfard, and Michiel Van De Panne. 2014. Vector Graphics Complexes. *ACM transactions on Graphics, Proceedings of ACM SIGGRAPH* (2014).
- Boris Dalstein, Remi Ronfard, and Michiel Van De Panne. 2015. Vector Graphics Animation with Time-Varying Topology. *ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH* 34, 4 (2015).
- Mathieu Desbrun, Peter Schröcker, and Alan Barr. 1999. Interactive animation of structured deformable objects. *Graphic Interface* (1999).
- Carlos Antonio Dominguez, Yuya Ichimura, and Mubbasir Kapadia. 2015. Automated interactive narrative synthesis using dramatic theory. *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games* (2015), 103–112.
- S. Erickson, S. Hardy, and J. Suppe. 2000. Sequential restoration and unstraining of structural cross sections: Application to extensional terranes. *AAPG Bulletin* 84 (2000), 234–249.
- Midland Valley Exploration. 2014. 2DMove. <http://www.mve.com/software/2d-kinematic>. (2014).
- Richard E. Fikes and Nils J. Nilsson. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. In *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence (IJCAI'71)*. 608–620.
- Clauzon G. Vaudin J.-L. Berger G.-M. et Fonteilles M., Leblanc D. 1993. *Carte géologique de la France (1/50 000), feuille Rivesaltes (1090)*. BRGM éd., Orlans, France.
- Haakon Fossen. 2016. *Structural Geology, 2nd Ed.* Cambridge University Press.
- Jean-David Génevaux, Éric Galin, Eric Guérin, Adrien Peytavie, and Bedrich Benes. 2013. Terrain Generation Using Procedural Models Based on Hydrology. *ACM TOG (Siggraph 2013)* 32, 4 (July 2013), 143:1–143:13.
- Dan B Goldman, Brian Curless, David Salesin, and Steven M. Seitz. 2006. Schematic Storyboarding for Video Visualization and Editing. *ACM Trans. Graph.* 25, 3 (July 2006), 862–871.
- R. Groshong. 1999. Structural validation, restoration and prediction. In *3D Structural Geology: A Practical Guide to Surface and Subsurface Map Interpretation*. Springer, Berlin, Heidelberg, 305–372.
- Halliburton. 2009. Lithotect. <https://www.landmark.solutions/LithoTect-Advanced-Structural-Modeling>. (2009).
- Arnab Jhala and R. Michael Young. 2015. Cinematic Visual Discourse: Representation, Generation, and Evaluation. *IEEE Transactions on Computational Intelligence and AI in Games* (2015), 69–81.
- Jana Koehler and Jörg Hoffmann. 2000. On Reasonable and Forced Goal Orderings and Their Use in an Agenda-driven Planning Algorithm. *J. Artif. Int. Res.* 12, 1 (June 2000), 339–386.
- Daniel Simoes Lopes, Daniel Mendes, Mauricio Sousa, and Joaquim Jorge. 2016. Expedition illustration of layer-cake models on and above a tactile surface. *Computers and Geosciences* (2016).
- Endre Mølster Lidal, Mattia Natali, Daniel Patel, Helwig Hauser, and Ivan Viola. 2012. Geological storytelling. *Computers & graphics* 37, 445-459 (2012).
- Baptiste Mary. 2012. SLAMTec. <https://www.u-cergy.fr/fr/laboratoires/laboratoire-gec/equipement/logiciels.html>. (2012).
- F. Kenton Musgraves, Craig E. Kolb, and Robert S. Mace. 1989. The Synthesis and Rendering of Eroded Fractal Terrains. *Transactions on Graphics* (1989), 41–50.
- Mattia Natali, Endre M. Lidal, and Daniel Patel Julius Parulek, Ivan Viola. 2013. Modeling Terrains and Subsurface Geology. *NPROCEEDINGS, EuroGraphics 2013 State of the Art Reports (STARS)* (2013), 445–459.
- Jacob Olsen. 2004. Realtime Procedural Terrain Generation. *Department of Mathematics And Computer Science (IMADA) University of Southern Denmark* (2004).
- Y. Philippe, E. Deville, and A. Masclé. 1998. Thin-skinned inversion tectonics at oblique basin margins: example of the western Vercors and Chartreuse Subalpine massifs. *Geological Society Special Publication* (1998), 239–262.
- Mark Riedl and Michael Young. 2006. From Linear Story Generation to Branching Story Graphs. *IEEE Computer Graphics and Applications Special Issue on Interactive Narrative* (2006).
- Amorim Ronan. 2017. *3D Geological Modeling from Concept Sketches and Annotations*. University of Calgary.
- Stuart J. Russell and Peter Norvig. 2003. *Artificial Intelligence: A Modern Approach* (2 ed.). Pearson Education.
- Schlumberger. 2011. Dynel2D. <http://www.software.slb.com/-/media/software/documents/external/product%20sheets/dynel.2d.ps.pdf>. (2011).
- Flora Ponjou Tasse, Arnaud Emilien, Marie-Paule Cani, Stefanie Hahmann, and Neil Dodgson. 2014. Feature-based Terrain Editing From Complex Sketches. *Computers and Graphics* (2014), 101–115.
- Y. Yamada and K. McClay. 2003. Application of geometric models to invert listric fault systems in sandbox experiments: hanging wall deformation and section restoration. *Journal of Structural Geology* 25 (2003), 1551–1560.
- Patrick Ye and Timothy Baldwin. 2008. Towards Automatic Animated Storyboarding. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1 (AAAI'08)*. AAAI Press, 578–583. <http://dl.acm.org/citation.cfm?id=1619995.1620089>

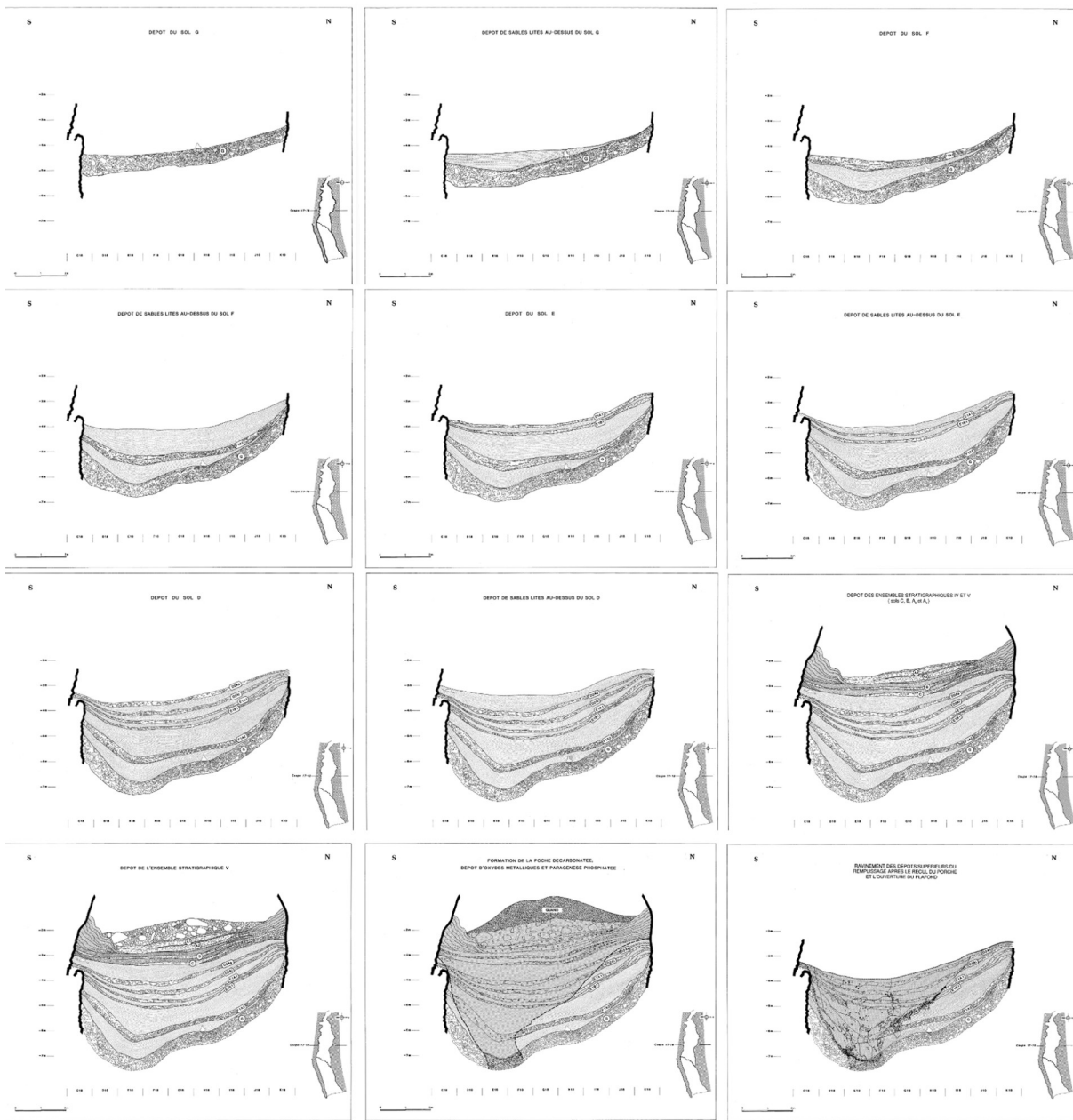


Figure 7: Manual cross section restoration process. Represented by a sketch sequence, geologists, Alain Fournier in this case, undo all geological event sequentially, retrieving all of terrain's past states. The sketches are then displayed in the reverse order, following the chronological order of the terrain formation. The example illustrated here represent a cross-section restoration process of the Caune de l'Arago described in Figure 8.

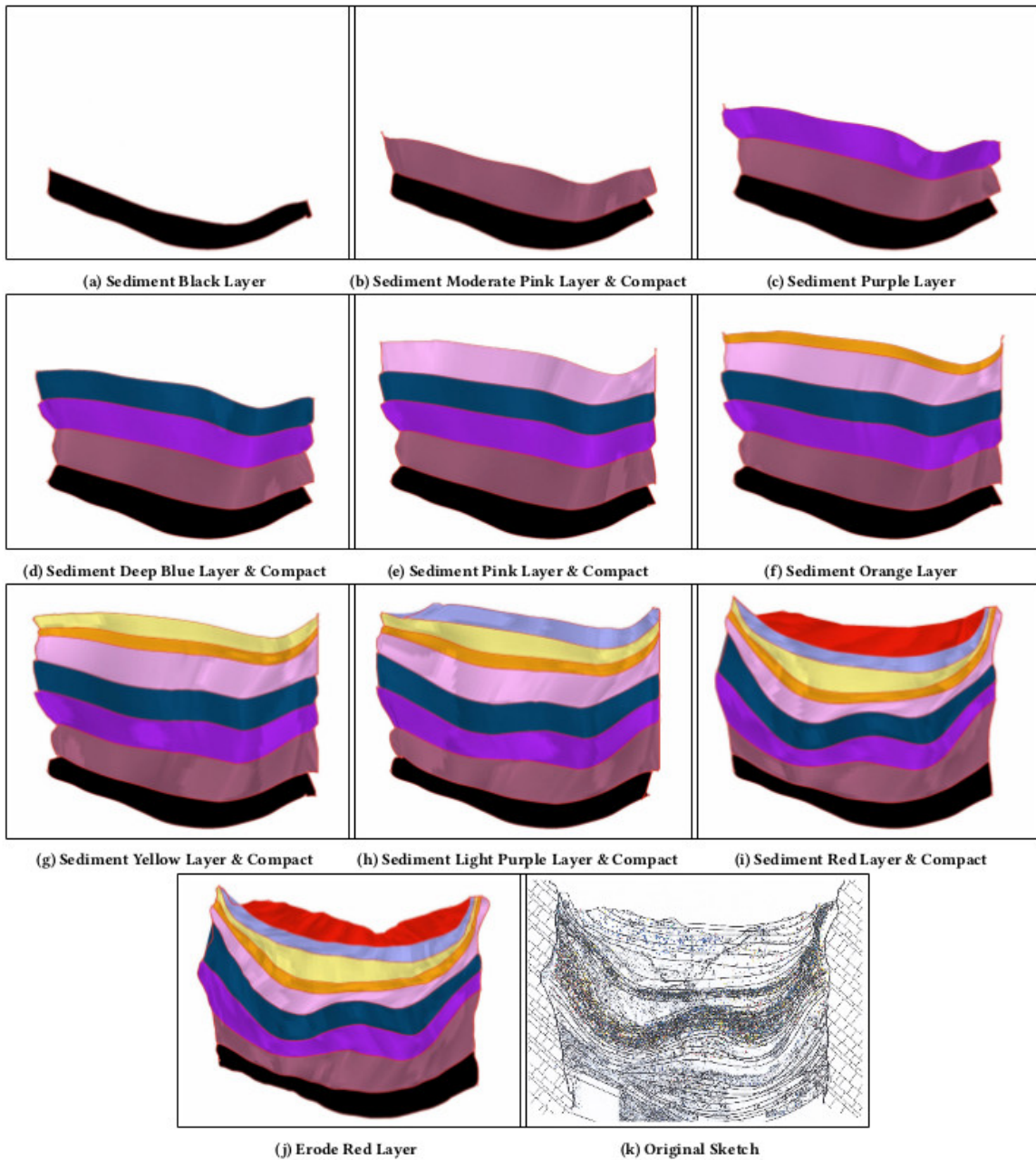


Figure 8: Caune de l'Arago restoration Storyboard. Caune de l'Arago is a cave located in the South of France, 40 km to the NE of the Pyrenees and 25 km west from the Mediterranean Sea, where 11 m of a 17 m thick Pleistocene infilling has been excavated since 1964, under the successive supervision of Henry de Lumley, Marie-Antoinette de Lumley, Anne-Marie Moigne and Christian Perrenoud. It yielded the oldest to date French human remains, together with hundreds of thousands of animal bone fragments and stone tools. This cross-section compiles lithostratigraphy, directly drawn from the field (sedimentary and postdepositionnal features, i.e. bedding, texture, color, cimentation, FeMn oxides,...) and informations obtained after the study of the components of the layers (sedimentology, biostratigraphy, ethnostratigraphy). This example was simplified into 9 layers that deposited and caused the compaction of the above layers.

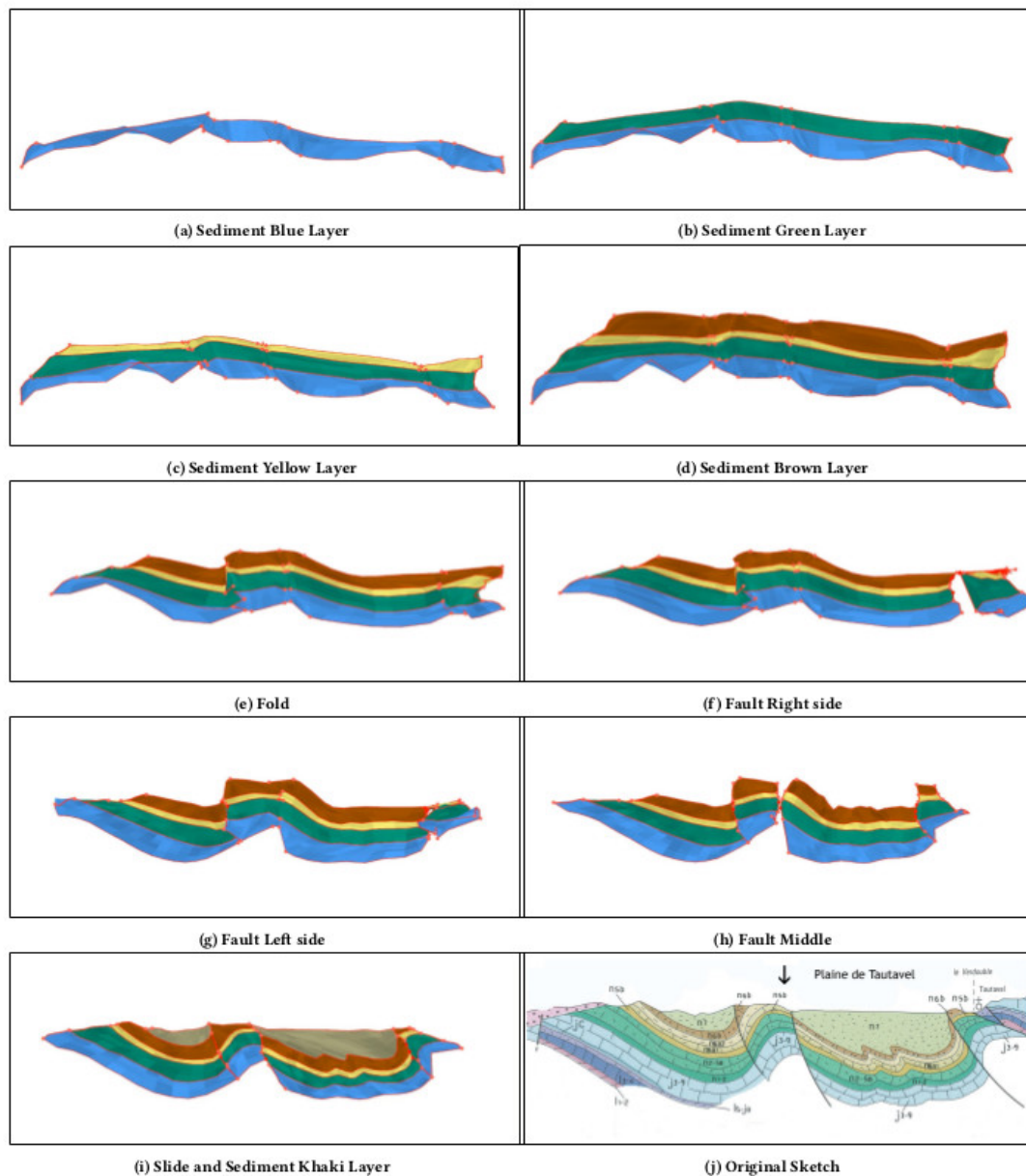


Figure 9: Tautavel Fields restoration Storyboard. Those fields have the same location than the Caune de l’Arago (below the arrow in (j) Original Sketch) but they extend to a 100 times wider scale. Being 5.7 km wide and 1 km high, this cross section is composed of 5 sediment layer as well as 3 Faults. In this generated storyboard, we present one restoration scenario where we un-sedimented the Khaki Layer, then closed successively all faults, un-folded the whole terrain and un-sedimented the remaining layers. Original Sketch modified from [Fonteilles M. 1993]

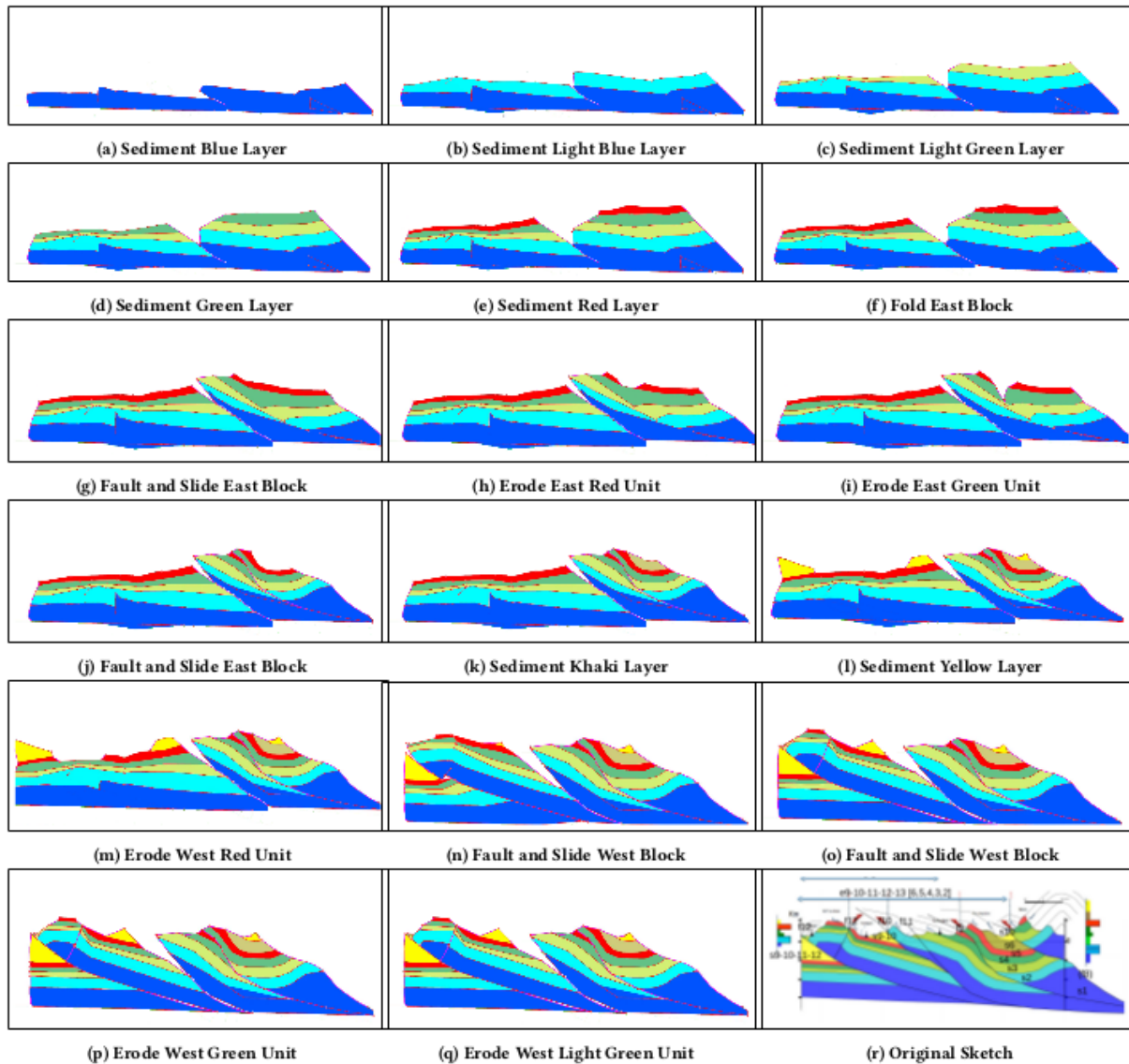


Figure 10: Chartreuse restoration Storyboard. Located in the French Alps, across the "Cluse de l'Isère", downstream to the city of Grenoble, this cross-section is 15 km wide and 5 km high and contains 7 sediment layers, 3 Faults and 5 Eroded units. In the chronological order, the first five layers (blue, light blue, light green, green and red) deposited. Then, after two small erosions, a faulting occurred on the east side. Two new layers (Khaki and Yellow) deposited before some local erosions on the west side. Finally, two faulting events occurred on the west side, making the Chartreuse fit its current state.

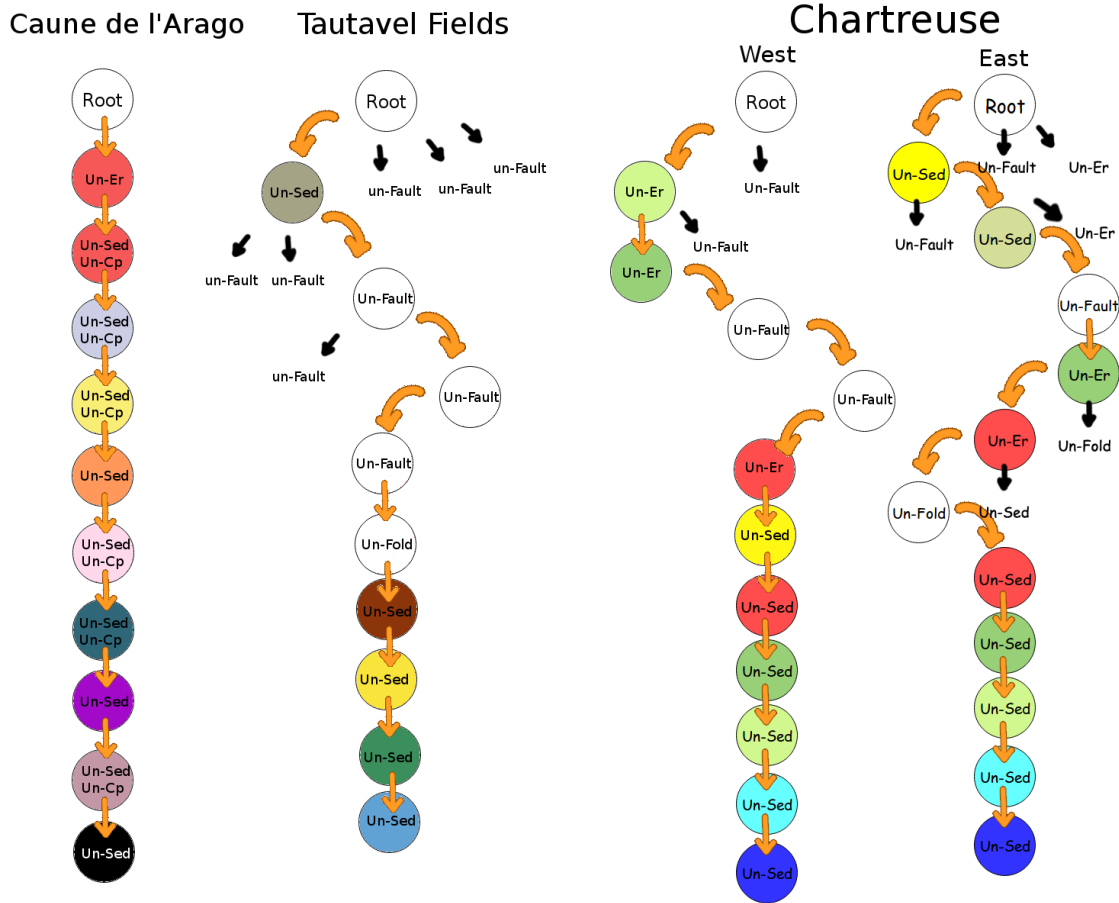


Figure 11: Story trees: (a) Caune de l'Arago; (b) Tautavel fields; (c) Chartreuse. The story tree describes the storyboard generation process where at each node, several undo events are proposed, letting the user choose one of them and continue the iterative process at the end of each event simulation. Un-Sed (respectively Un-Er and Un-Cp) represents an un-sedimentation (respectively un-erosion and un-Compaction) event of the layer corresponding to the circle filling color. Colored arrows correspond to the user's chosen events, while black ones correspond to alternative choices that were not chosen.