

Chapter 5

The Haunted Swamps of Heuristics: Uncertainty in Problem Solving

Artem Amirkhanov, Stefan Bruckner, Christoph Heinzl
and M. Eduard Gröller

Abstract In scientific visualization the key task of research is the provision of insight into a problem. Finding the solution to a problem may be seen as finding a path through some rugged terrain which contains mountains, chasms, swamps, and few flatlands. This path—an algorithm discovered by the researcher—helps users to easily move around this unknown area. If this way is a wide road paved with stones it will be used for a long time by many travelers. However, a narrow footpath leading through deep forests and deadly swamps will attract only a few adventure seekers. There are many different paths with different levels of comfort, length, and stability, which are uncertain during the research process. Finding a systematic way to deal with this uncertainty can greatly assist the search for a safe path which is in our case the development of a suitable visualization algorithm for a specific problem. In this work we will analyze the sources of uncertainty in heuristically solving visualization problems and will propose directions to handle these uncertainties.

5.1 Introduction

Solving visualization tasks is somewhat similar to finding a path through different terrain (see Fig. 5.1). This terrain consists of two areas with very different characteristics: steep mountains and dangerous swamps. The mountains represent high grounds of theory. Choosing a path through the mountains one is on stable ground, but the path may be steep, tedious, and inexperienced travelers might get lost or stuck at a dead

A. Amirkhanov (✉) · S. Bruckner · M.E. Gröller
Vienna University of Technology, Vienna, Austria
e-mail: artem@cg.tuwien.ac.at

S. Bruckner
e-mail: bruckner@cg.tuwien.ac.at

M.E. Gröller
e-mail: groeller@cg.tuwien.ac.at

C. Heinzl
Wels Campus, Upper Austrian University of Applied Sciences, Wels, Austria
e-mail: christoph.heinzl@fh-wels.at



Fig. 5.1 Task solving is finding a path from a problem to a solution

end. The resulting way can be a serpentine road or may even lead through a tunnel, if enough effort is invested in its construction. Another possibility is to go through the haunted swamps of heuristics. The path through the swamps seems to be rather easy and straight-forward, as it is flat walking, but in fact it is neither smooth nor safe. There might be unexpected turns and twists. There are few and badly marked paths through these swamps. In case of even slight deviations the pioneer can easily find himself at a dead end or even get sucked into the deadly waters. Only few researchers find viable, elegant paths through the swamps. If it works out it may result in a shorter way from problem to solution as compared to going on the high grounds of theory.

Using heuristics is often considered as a bad choice in the design of a visualization algorithm. Reviewers of visualization papers tend to dislike heuristics. They comment on heuristics like: *lots of parameter tweaking, only heuristics, yet another heuristic, too many heuristic choices, or ad hoc parameter specification*. Should we try to avoid heuristics and attempt to only find the theoretically well-grounded solutions?

5.2 Heuristics

Heuristic (or heuristics; Greek: “*Ευρισκω*”, meaning *to find* or *to discover*) refers to experience-based techniques for problem solving, learning, and discovering. As an adjective, heuristic pertains to the process of gaining knowledge or some desired result by intelligent guesswork rather than by following some pre-established rules, laws, or formulae. The underlying theory might not even be known. Humans often

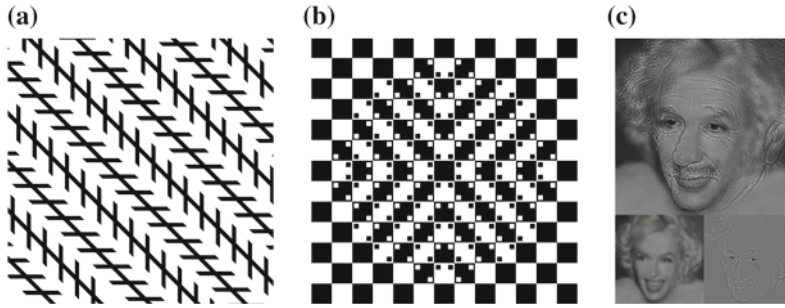


Fig. 5.2 Three well known optical illusions: Zöllner illusion (a), bulging checker board illusion (b), and blur and picture content illusion (c)

apply heuristic and approximate approaches if they have to solve complex problems. In many cases they do not have the complete information for a precise solution. Heuristics are about finding a good enough solution where an exhaustive search would be impractical. One of the most commonly used heuristics, which can initiate a problem solving process, is trial and error. Other common examples of heuristics are, e.g., drawing a figure for better problem understanding, working backward from an assumed solution, or examining a concrete example of an abstract problem. Previous experiences and known information result in such heuristic concepts as prejudices and stereotypes. By evolution some heuristic approaches are firmly anchored in perceptual and mental processes. Heuristic problem solving may work in many circumstances, but in some cases fails to deliver the correct solution. This can lead to cognitive biases in decision making or to imperfections in perception like optical illusions. Three of these optical illusions are shown in Fig. 5.2. Long black lines with horizontal and vertical marks in the Zöllner illusion [20] (Fig. 5.2a) are parallel to each other but do not seem to be. The apparent bulging of the checker board (Fig. 5.2b) is not real: the board is planar. The third example (Fig. 5.2c) shows how frequencies affect the perceived content of a picture. If the viewer examines the image from a short distance, Albert Einstein's face is seen, but, if looking from farther away, the face changes to that of Marilyn Monroe. These three examples are synthetic, but sometimes optical heuristics can fail in real life situations as well. For instance, the size of the moon seems to vary depending on its distance from the horizon.

Algorithm developers use heuristics for problem solving in many ways. Whenever the information available for a task is incomplete or exact solutions are too expensive, an algorithm or some algorithmic part is supplemented with heuristics. Often the heuristic portions of an algorithm are encoded in coefficients or parameters. The following section will elaborate more on the heuristic nature of parameters, issues connected to parameters and ways to explore the uncertainty of an algorithm by studying its parameters.

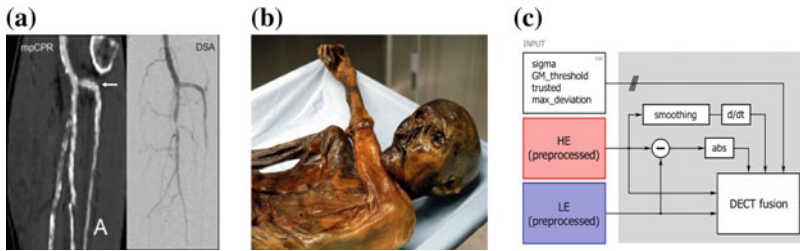


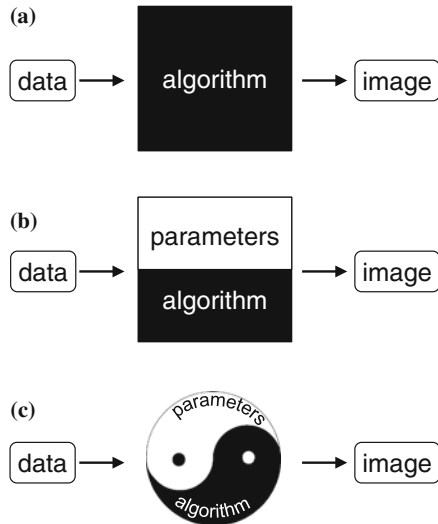
Fig. 5.3 Objects of desire in science. **a** Data (from Multipath CPR [15]). **b** Mummy [Ötzi the Iceman (© South Tyrol Museum of Archaeology—www.iceman.it)]. **c** Algorithm (courtesy of Heinzl [11])

5.3 Objects of Desire in Science

Every scientific discipline has its own object of desire, i.e., study focus. In some areas like geology or medicine these are the physical and medical phenomena and processes hidden inside the data. Data collection, classification, and analysis in order to get useful insights are in the center of the scientific activity (Fig. 5.3a). In other areas artifacts, fossils and mummies (Fig. 5.3b) are the investigated items. In yet other disciplines pieces of text or poems might be in the center of attention. In computer science algorithms (and data structures they work on) are the key entities researchers and developers are designing and investigating (Fig. 5.3c).

An algorithm is a set of instructions that operate on data which are given through constants and variables. And then there are parameters. Constants are, as the name says, constant during the execution of an algorithm. They are fixed and may be mathematical or physical quantities like the cosmological constant. Variables contain values that change during algorithm execution. So where do parameters fit into this picture? Parameters (again a Greek term) are auxiliary measures which are arbitrary but fixed. They are neither constants nor variables. If an algorithm simulates a specific model within a class of models that share the same characteristics, the parameter is fixed for this one model. Switching to another model in the class means varying the corresponding parameter. Parameters are somewhat dual in their nature. And they are just ‘auxiliary’. Computer scientists and also visualization researchers are very fond of the instruction part of their algorithms, which they dedicate a lot of time to. Constraints, boundary conditions, approximations, and calibrations are issues that often are encoded in parameters. Even more: if the algorithm still does not work properly, it can only be in the parameters or more of them are needed. They are our easy back-door out. Parameters in many cases do not get the necessary attention and are all too often supposed to be specified heuristically. And it is exactly here where heuristics get a bad reputation. Sometimes the inadequacy of an algorithm is covered up by a set of unintuitive parameters which the developer himself cannot control properly. So to make a virtue out of necessity, the parameters are declared to be user-defined and this is sold as additional flexibility. In reality often this puts an

Fig. 5.4 Algorithm and its parameters. **a** Algorithm without parameters. **b** Parameters on top of the algorithm. **c** Yin and Yang union of algorithm and its parameters



undue burden on the user and impacts the usability and applicability of an algorithm. Finding a path, i.e., solving a problem, requires an algorithm *and* parameters. Visualization research is often concerned with taking data and producing an image as visual result. This mapping is realized through an algorithm. But as said above an algorithm alone (like in Fig. 5.4a) is typically not sufficient by itself. Also an algorithm and its parameters do not live side-by-side or one is on top of the other (like in Fig. 5.4b). An algorithm and its parameters are closely intertwined in a yin yang union (like in Fig. 5.4c). Changing an algorithm has an immediate impact on the pertaining parameters, some of them may even vanish or new ones might come into existence. On the other hand, changing parameters may heavily impact the functioning of an algorithm. The results may even be similar to the results from a quite different algorithm (with other parameters of its own). In various disciplines parameter-space analyses are already well established to determine the robustness and stability of processes or procedures. In the area of visualization the investigation of parameters and the spaces they live in has gained increased interest only in recent years. Knowledge-assisted visualization or the visualization of variations and ensembles go into this direction. In the next section we will discuss parameter-space analysis in more detail.

5.4 Parameter-Space Analysis

New and improved imaging modalities, like dual energy computed tomography, allow measuring the same specimen with varying parameters. Increased computing performance (multi-core CPUs, GPUs) allows not only calculating one simulation run but hundreds or even thousands of runs with changing parameter settings. This necessitates investigating and visualizing large sets of simulations or data ensembles at the

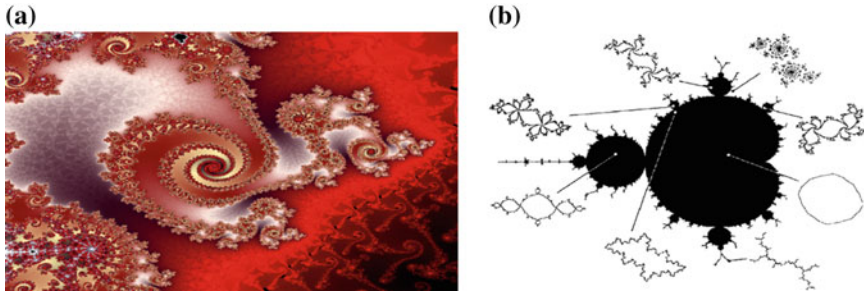


Fig. 5.5 Close-up of a Julia set (a) and the Mandelbrot set as parameter map of all Julia sets (b) (courtesy of Falconer [8] © John Wiley and Sons)

same time. Dynamical systems are an illustrative example where parameter-space analysis has already been applied for a long time. Julia sets (Fig. 5.5a) are the result of iterating a simple quadratic polynomial in the complex plane. Each polynomial is characterized by a parameter p . Different parameters lead to greatly varying results where the outcome may for example be a connected or disconnected Julia set. Doing an analysis of all possible parameters leads to a parameter space display where the beautiful and immensely intricate Mandelbrot set appears (Fig. 5.5b). The parameter p of the Julia sets turns into a variable in the parameter space where the Mandelbrot set resides. As a side note: the Mandelbrot set comprises all those values p whose corresponding Julia sets are connected. With dynamical systems a parameter-space analysis may be local or global. A local investigation looks at small perturbations of a parameter to identify for example stability properties which could be direction dependent. A global investigation looks at larger structures in parameter space, e.g., asymptotic behavior, basins of attraction, bifurcations or topological items like separatrices. In the visualization domain parameter-space analyses become feasible as well for data ensembles and parameterized simulation runs. Parameter-space investigations from other fields might act as guiding examples, though the peculiarities of our applications have to be taken into account. While for dynamical systems parameters often change continuously, in our applications parameters may be for example discontinuous, discrete, or categorical in nature. Certain regions of parameter space may be uninteresting or even meaningless because of the physical properties of the underlying phenomenon. With the holistic view on large ensembles or simulation runs interesting questions arise:

- What is the local stability of a (visualization) parameter setting?
- How do different parameters influence each other?
- What are permissible (visualization) parameter ranges?
- How can we automatically define parameter settings that optimize certain properties?
- How sensitive is the visualization outcome on parameter perturbations?
- How to efficiently sample the high dimensional parameter spaces?
- How to do reconstruction in these parameter spaces?

In the following recent examples of parameter-space exploration in the visualization domain are shortly discussed.

Ma [12] introduced a visualization system which presents information on how parameter changes affect the result image as an image graph based on data generated during an interactive exploration process. Berger et al. [4] study continuous parameter spaces in order to guide the user to regions of interest. Their system uses continuous 2D scatter plots and parallel coordinates for a continuous analysis of a discretely sampled parameter space. Not sampled areas of the parameter space are filled with predictions. The uncertainty of the predictions is taken into account and also visualized. The stability of the results with respect to the input parameters is visualized and explored.

In the work by Amir Khanov et al. [1] parameter space exploration is carried out in order to detect the optimal specimen placement on a rotary plate for industrial 3D X-ray computed tomography. The parameter space is represented by Euler angles defining the orientation of the specimen. The parameter settings providing the optimal scanning result were determined using a visual analysis tool. The stability of the result with respect to these parameters was additionally taken into account.

Analyzing how segmentation performs when parameters change and finding the optimal set of parameters is a tedious and time-consuming task. It is usually done manually by the developers of segmentation algorithms. Torsney-Weir et al. [16] presented a system to simplify this task by providing an interactive visualization framework. The *Tuner* tool samples the parameter space of a segmentation algorithm and runs computations off-line. A statistical model is then applied for the segmentation response. Hyper slices [19] of the parameter space and 2D scatter plots are used to visualize these data. Based on the prediction model, additional samples of parameter space may be specified in the regions of interest. The tool allows finding the optimal parameter values and estimating the segmentation algorithm's robustness with respect to its parameters.

FluidExplorer by Bruckner and Möller [7] is an example of goal-driven parameter exploration. They explore the parameters of physically-based simulations for the generation of visual effects such as smoke or explosions. First, the set of simulation runs with various parameter sets is run off-line. Then sampling and spatio-temporal clustering techniques are utilized to generate an overview of the achievable results. Temporal evolution of various simulation clusters is shown. The goal is to find the set of parameters resulting in a certain visual appearance. The metric is defined via user interaction when the user explores the simulation space.

The work of Waser et al. [17] uses *World Lines* to study complex physical simulations. In such time-dependent simulations parameters can change their values at arbitrary moments in time. Decision support is provided by the ability to explore alternative scenarios. A World Line is introduced as a visual combination of user events and their effects in order to present a possible future. The proposed setup enables users to interfere and add new information quickly to find the most appropriate simulation outcome. The usefulness of the technique is shown on a flooding scenario where a smoothed particle hydrodynamics simulation is used. Waser et al. further expand their framework in [18]. The authors take uncertainty of the

simulation parameters into account to provide the confidence in the simulation outcome. In the proposed solution, users can perform parameter studies through the World Lines interface to process the input uncertainties. In order to transport steering information to the underlying data-flow, a novel meta-flow (extension to a standard data-flow network) is used. The meta flow handles components of the simulation steering.

World lines are an example of how to handle uncertainty and parameter variations in a computational-steering environment. Now we move further down the visualization pipeline, take a look at the visualization-mapping stage and discuss how an integral view of parameter spaces may influence our view of ensembles of visualization algorithms.

5.5 Parameter Spaces and Visualization Algorithms

A-space [2] is a space where all visualization algorithms live. In A-space every algorithm with a specific parameter setting is represented by a unique point. Perturbing the parameters of an algorithm produces a point set (solution cloud) in A-space. The solution clouds of two quite different algorithms may overlap. This means that a visualization algorithm 1 with parameter 1 produces the same or very similar results as algorithm 2 with parameter 2.

The holistic view of visualization algorithms being embedded in a common space enables interesting investigations and may lead to novel visualization techniques. Sample questions are: What is the stability of an algorithm in A-space? Are there global structures in this space? Can there be smooth transitions between rather diverse algorithms? What would be sparse blendings between various algorithms? MIDA [6] is an interesting example where two well-established volume rendering techniques, i.e., direct volume rendering (DVR) and maximum intensity projection (MIP), are combined in a fine-grained fashion. A smooth transition between DVR, MIP and MIDA itself becomes possible and allows exploiting the strengths of DVR and MIP while avoiding their weaknesses. Another more coarse-grained combination of visualization algorithms would for example be two-level volume rendering [10].

5.6 Algorithms, Parameters, Heuristics—Quo Vadis?

Algorithms and their parameters are closely intertwined. They together constitute a path from the problem to the solution by mapping data to images. Even if parameters are ‘just auxiliary measures’ they definitely need our help. Heuristic parameter specification is a viable approach as long as some sort of sensitivity analysis is taken care of. This sensitivity analysis should not be only done in parameter and algorithm spaces it should also be extended to data and image spaces. Furthermore the sensitivity analysis should also be applied to interaction space, as we are often confronted with interactive visualization applications. An example in this respect is

the work of Gavrilescu et al. [9]. The increased complexity of data ensembles, large simulation runs and uncertainty in the data poses interesting visualization challenges. How shall we cope with the increased data and analysis complexity? Three of several possible directions include integrated views and interaction [3], comparative visualization [13] and fuzzy visualization [14]. With fuzzy visualization, techniques of information theory will play a bigger role in coping with large parameter spaces.

Currently problem solving in visualization is typically algorithm-centric and thus imperative by definition. With increased data complexity it will probably become more declarative and thus more data and image centric, as domain experts have always been data-centric. A data-centric approach means that the user does not specify how data is mapped to images but defines which features of the data he would like to see how in the result images. This is like specifying pre- and post-conditions but not the instructions to get from the first to the second. An optimization process should then automatically figure out which algorithms and parameter settings best fulfill the user defined declarations and constraints. Semantic layers [14] is a step in this direction.

Frameless rendering [5] is about efficiently rendering animation sequences where pixels are updated on a priority basis. At no point in time all pixels of the image are up-to-date, i.e., no frame is available though the animation sequence as a whole evolves. Analogously to this concept, we foresee algorithmless visualizations in the sense that not a single algorithm is explicitly specified by the user in a specific application. For different features of the data and for different parts of the image the most appropriate algorithm among a set of possible candidates might be automatically selected. Various combinations and integrations of visualization algorithms might be possible to best achieve the user goals and declarations. Each pixel or voxel might get its own algorithm on demand.

Interval arithmetic has long been used to cope with uncertainties due to rounding, measurement and computation errors. Handling ensemble data in an analogous manner may lead to densely visualizing intervals or even distributions. While there are already some approaches to locally investigate visualization parameter spaces, not much has been done in terms of a global or topological analysis. For quantitative results visualization algorithms will have to provide more stability and robustness analyses in the future. With the increased data complexity (massive-multiple, heterogeneous data) heuristic approaches and parameter space analyses will become even more important. This raises the need to visualize uncertain, fuzzy, and even contradictory information.

Very often heuristics are useful. But even if you do not (exactly) know what you are doing (this is what heuristics is about), you should make sure that it is safe what you are doing. Safety concerns robustness, stability, and sensitivity of an algorithm and its parameters. So heuristics are great, when handled with care. This way your paths through the haunted swamps will be safe ones. We for sure agree with a statement by Voltaire: “Doubt is not a pleasant condition, but certainty is absurd.”

Acknowledgments We would like to acknowledge the Bridge-Project SmartCT and the K-Project ZPT (<http://www.3dct.at>) of the Austrian Research Promotion Agency (FFG). We also acknowledge the ViMaL project supported by the Austrian Science Fund (FWF), grant no. P21695.

References

1. Amirkhanov, A., Heinzl, C., Reiter, M., Gröller, M.E.: Visual optimality and stability analysis of 3DCT scan positions. *IEEE Trans. Visual. Comput. Graph.* **16**(6), 1477–1486 (2010)
2. Balabanian, J.P., Gröller, M.E.: A. In: Hagen, H. (ed.) *Scientific Visualization: Interactions, Features, Metaphors*. Dagstuhl Follow-Ups, vol. 2, pp. 36–47. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2011)
3. Balabanian, J.P., Viola, I., Gröller, M.E.: Interactive illustrative visualization of hierarchical volume data. In: *Proceedings of Graphics Interface (best student paper award)* (2010)
4. Berger, W., Piringer, H., Filzmoser, P., Gröller, E.: Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Comput. Graph. Forum* **30**(3), 911–920 (2011)
5. Bishop, G., Fuchs, H., Mcmillan, L., Zagier, E.J.S.: Frameless rendering: double buffering considered harmful. In: *ACM Siggraph '94 Conference*, vol. 28, pp. 175–176 (1994)
6. Bruckner, S., Gröller, M.E.: Instant volume visualization using maximum intensity difference accumulation. *Comput. Graph. Forum* **28**(3), 775–782 (2009)
7. Bruckner, S., Möller, T.: Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Trans. Visual. Comput. Graph.* **16**(6), 1467–1475 (2010)
8. Falconer, K.: *Fractal Geometry: Mathematical Foundations and Applications*, 2 edn. Wiley, New York (2003)
9. Gavrilescu, M., Malik, M.M., Gröller, M.E.: Custom interface elements for improved parameter control in volume rendering. In: *14th International Conference on System Theory and Control*, pp. 219–224 (2010)
10. Hauser, H., Mroz, L., Bischl, G.I., Gröller, M.E.: Two-level volume rendering. *IEEE Trans. Visual. Comput. Graph.* **7**(3), 242–252 (2001)
11. Heinzl, C.: Analysis and visualization of industrial CT data. Ph.D. thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology (2009). URL:<http://www.cg.tuwien.ac.at/research/publications/2009/heinzl-2008-thesis/>
12. Ma, K.L.: Image graphs—A novel approach to visual data exploration. *Proc. IEEE Visual.* **1999**, 81–513 (1999)
13. Malik, M.M., Heinzl, C., Gröller, M.E.: Comparative visualization for parameter studies of dataset series. *IEEE Trans. Visual. Comput. Graph.* **16**(5), 829–840 (2010)
14. Rautek, P., Bruckner, S., Gröller, M.E.: Semantic layers for illustrative volume rendering. *IEEE Trans. Visual. Comput. Graph.* **13**(6), 1336–1343 (2007)
15. Roos, J., Fleischmann, D., Koechl, A., Rakshe, T., Straka, M., Napoli, A., Kanitsar, A., Sramek, M., Gröller, M.E.: Multipath curved planar reformation of the peripheral arterial tree in CT angiography. *J. Radiol.* **244**(1), 281–290 (2007)
16. Torsney-Weir, T., Saad, A., Möller, T., Weber, B., Hege, H.C., Verbavatz, J.M., Bergner, S.: Tuner: principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Trans. Visual. Comput. Graph.* **17**(12), 1892–1901 (2011)
17. Waser, J., Fuchs, R., Ribičić, H., Schindler, B., Blöschl, G., Gröller, M.E.: World lines. *IEEE Trans. Visual. Comput. Graph.* **16**(6), 1458–1467 (2010)
18. Waser, J., Ribičić, H., Fuchs, R., Hirsch, C., Schindler, B., Blöschl, G., Gröller, M.E.: Nodes on ropes: a comprehensive data and control flow for steering ensemble simulations. *IEEE Trans. Visual. Comput. Graph.* **17**(12), 1872–1881 (2011)
19. van Wijk, J.J., van Liere, R.: HyperSlice: visualization of scalar functions of many variables. In: *Proceedings of the 4th Conference on Visualization '93, VIS '93*, pp. 119–125. IEEE Computer Society (1993)
20. Zöllner, F.: Ueber eine neue Art von Pseudoskopie und ihre Beziehungen zu den von Plateau und Oepel beschriebenen Bewegungsphänomenen. *Annalen der Physik* **186**(7), 500–523 (1860)