

Chapter 9

Computer Graphics Procedural Modeling of Soil Structure

Hansoo Kim, Minerva J. Dorantes, Darrell G. Schulze
and Bedrich Benes

Abstract Soil scientists in the USA have created a large national database of written soil profile descriptions that follow a well-defined set of rules for describing soil morphological properties. Interpreting these soil descriptions is a skill that requires considerable practice and experience. While writing a soil description is straightforward, recreating a visual representation of a soil profile from a written description is very difficult. So far, there is no generalized approach for translating written or tabular soil descriptions into visual representations. We propose a novel procedural modeling approach inspired by procedural models commonly used in the field of computer graphics. Our framework takes tabular soil morphological data (i.e., soil profile descriptions) as textual input and translates it into visual features based on parametric models. These models can be used to generate two-dimensional soil profiles or to generate three-dimensional interactive models that allow rotation, scaling, and other forms of visual explorations. The procedural modeling technique enables the user to generate the soil profile visual representation with only a small amount of data. The images do not need to be stored because they are generated as needed.

Keywords Tabular soil morphological data · Procedural modeling · Computer graphics

H. Kim · B. Benes
Department of Computer Graphics Technology, Purdue University,
West Lafayette, IN 47907, USA

M.J. Dorantes · D.G. Schulze (✉)
Agronomy Department, Purdue University, West Lafayette, IN 47907, USA
e-mail: dschulze@purdue.edu

9.1 Introduction

Soil scientists have developed detailed terminology for describing the morphological properties of soil profiles. In the USA, there are large national databases of soil profile descriptions that follow a well-defined set of rules (Schoeneberger et al. 2012). With training and practice, writing a soil profile description is a fairly straightforward process, with the result that different trained soil scientists will write similar, but not necessarily identical, descriptions of the same soil profile. Interpreting written descriptions, however, is a skill that requires considerable practice and experience. Experienced soil scientists can draw on their experience to mentally visualize soil profiles based on written descriptions, but to students and the vast majority of more casual users of soils information, soil profile descriptions, whether in tabular or narrative form, can be inscrutable.

Soil profile descriptions, however, contain large amounts of information that can be used to reconstruct an image of the soil originally described. Some soil properties, such as horizon depth and dominant color, are easy to represent in simple, schematic profile diagrams. The Soil Web Apps produced by the California Soil Resource Lab (<http://casoilresource.lawr.ucdavis.edu>) take this approach. The resulting profiles, however, do not really look like soil profiles, although they do provide information and context to trained soil scientists. There is currently no generalized approach for translating written soil profile descriptions into visual representations that one would recognize as soil profiles, primarily because soil structure is so difficult to represent.

The field of computer graphics has developed very efficient and effective methods for data visualization and representation. *Procedural methods* have recently found a prominent place among traditional techniques that take existing data and display them in different forms (Ebert et al. 2002). Procedural methods have been used in areas ranging from modeling of cities (Parish and Müller 2001), plants (Prusinkiewicz et al. 1990), to entire virtual worlds (Smelik et al. 2014). The basic idea of procedural methods is to represent an image, which in soil science is the image of a soil profile, as a computer program with specific input parameters. Then, when the image is needed, the code is executed and the image is generated. An obvious advantage of procedural representations is the huge data compression, variability of the output since one code can generate various images, and versatility. Very recently, the *inverse problem* has become important in various fields. The inverse problem attempts to answer the question, ‘given existing data, for example, a written soil profile description, what is the code that can generate it?’ This is a very complex problem, and several solutions have been found so far for virtual plants (Stava et al. 2014), models of cities (Vanegas et al. 2012), and procedural art (Stava et al. 2010).

Here, we apply the inverse procedural modeling approach to the field of soil science. We propose a novel procedural modeling approach that takes tabular soil morphological data (i.e., soil profile descriptions) as textual input and then translates that data into visual features based on parametric models. The output of these models can then be displayed as 2-dimensional images of soil profiles, or as three-dimensional (volumetric) interactive models that can be rotated, scaled, and explored visually in other ways. The procedural modeling technique enables the user to generate a visual representation of a soil profile with only a small amount of data, and the images do not need to be stored because they are generated as needed.

9.2 System Overview

The overview of our system is depicted in Fig. 9.1. The system takes tabular soil data as input. The *visualizer* selects the appropriate procedural model according to the tabular soil data and applies the appropriate parameters. In this step, the procedural models produce a magnitude map, which is a three-dimensional array of points (x, y, z) where x and y give the location of a point in a right Cartesian coordinate system, and z is the height above the x - y plane. The magnitude map is equivalent to the digital elevation model (DEM) used in geographic information system (GIS) software. The *renderer* then generates the final image using user-defined rendering methods equivalent to hill shading in GIS. The renderer converts the cloud-like image produced by the procedural model into an image of the highlights and shadows of a solid surface illuminated by a light source. Details of each step are described below.

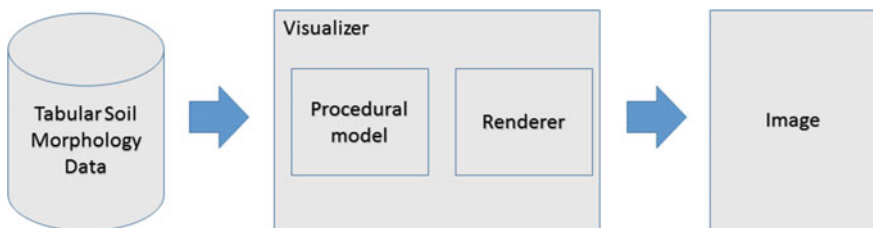


Fig. 9.1 System overview

9.3 Procedural Modeling of Tabular Soil Morphology Data

9.3.1 Data

Our goal is to use the Official Soil Series Descriptions (OSDs) available from the USDA Natural Resource Conservation Service (<http://www.nrcs.usda.gov/wps/portal/nrcs/site/soils/home/>) for input data. The OSDs consist of a national collection of more than 20,000 detailed soil series descriptions. Particularly, we focus on the soil structural descriptions in the OSDs. While properties such as color, depths, or boundary patterns are easily visualized by connecting them to quantitative parameters and values (e.g., converting the Munsell color designations to the RGB system), structural descriptions are difficult to quantify because they implicitly include many qualitative visual aspects. For instance, when a horizon description mentions ‘subangular blocky,’ we cannot clearly define its quantitative properties in a straightforward way, which is a crucial part of reconstructing a visual representation. Standardized soil structural terminology is commonly used for soil profile descriptions, so the task is to translate that terminology into images that look like the feature being described. For our initial work, we are focusing on ‘granular,’ ‘subangular blocky,’ ‘prismatic,’ and ‘platy’ soil structural keywords.

Soil structural keywords contain considerable qualitative information that is too subtle or vague to model directly by simple geometric techniques. For example, subangular blocky structure has properties of ‘cubic structural units’ and ‘rounded edges,’ but a model that simply filled space with cubes having rounded edges would not look like subangular blocky soil structure. ‘Rounded edges’ does not specify curvature, length, or direction, for example, and neither are global aspects of the ‘cubic structural units’ specified in terms of position, size, and transitions among units. In addition, these properties must have a degree of randomness in order to appear natural. To address the problem, we employ procedural noise functions (Fig. 9.2) and match their properties with structural keywords.

9.3.2 Procedural Modeling

As discussed above, there are several advantages in using procedural models. First, they generate seamless images because they are functions that can be evaluated at any point in a given domain (i.e., one-, two-, or three-dimensional space). Therefore, there are no ‘empty’ points in the output, and the entire output domain is controlled by the functions. This is especially beneficial when we map a reconstructed appearance onto a three-dimensional model because there are no undesirable discontinuities. Second, procedural noise functions can generate various

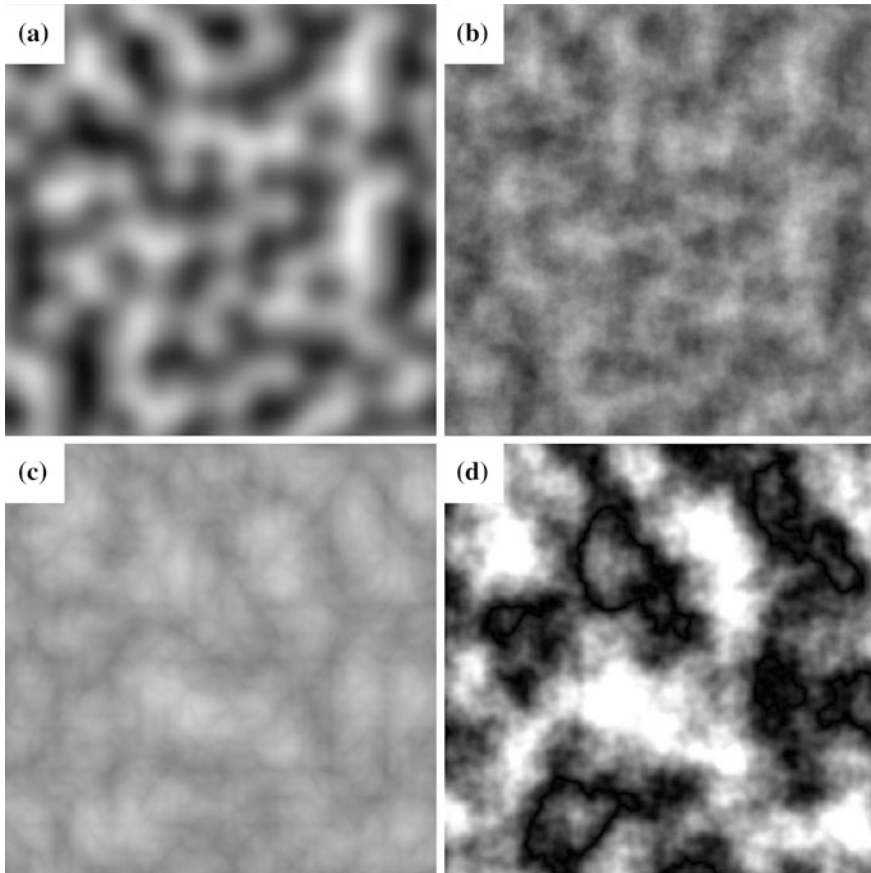


Fig. 9.2 The output from four different procedural models. **a** *Gradient Noise* has very high contrast and distinct structures. **b** *Fractal Synthesis* has a cloud-like appearance, medium contrast, and smooth but noticeable structures. **c** *Turbulence* has low contrast, smooth boundaries, and small rounded sub-patterns. **d** *Trigonometric Jittering* has distinct structure and high contrast

types of patterns by changing just a few input parameters. For example, trigonometric jittering, also known as ‘Perlin Marble,’ mainly creates distinguishable, but smoothly blended structures (Fig. 9.2d). On the other hand, fractal synthesis generates regularly distributed and smoother appearances (Fig. 9.2b). We are able to adjust the size, smoothness, and width–height ratio by using less than four input parameters. Third, procedural models are fast, and a 512×512 pixel, two-dimensional image is generated in tens of microseconds on today’s computers. This enables interactive visualization, which is a crucial aspect for images that will be used for teaching and learning.

Table 9.1 Parameters of the four procedural models

Parameter	Description
X seed	Unit evaluation interval in the horizontal direction (0.0–1.0). Greater values generate smoother horizontal appearances
Y seed	Unit evaluation interval in the vertical direction (0.0–1.0). Greater values generate smoother vertical appearances
Octave	Applies to all models except gradient noise (1–16). The value determines the amount of detail to be added. Octave value is actually the number of repetitions for adding high frequency values
Frequency	Only applies to trigonometric jittering (2.0–10.0). The value defines the ‘jitter’ value at each octave

We currently use four procedural models (Fig. 9.2). They all are the simplest types of procedural noise functions and are employed for several reasons. First, while we are able to obtain more complex and sophisticated appearances using more complicated procedural methods, they take a significantly longer time to generate an output. Second, they require only a few input parameters (Table 9.1) that can be correlated with soil structural terms that describe the size, grade, and shape of soil structural aggregates. The output of a procedural model can be adjusted using only a few parameters. For example, by changing the *X* seed and *Y* seed, we are able to achieve many different appearances (Fig. 9.3). Finally, we have good intuition about the structures in the generated output image, because they consist of only a few interpolation equations. Since we must conduct numerous trial and error tests to find input parameters that generate images that look like soil structures, simple procedural functions accelerate the process. The result of our user-assisted process is shown in Table 9.2.

9.3.3 Visualization

While procedural models generate desired structures with randomness, the resulting outputs look different from the soil structure one sees in a soil profile. We apply additional hill shading (Horn 1981) to the results generated by the procedural models. Hill shading is a grayscale, image-space shading technique that takes into account the light source’s position in order to model the pattern of highlights, shadows, and gray scale gradients that result from light shining on an opaque surface. Hill shading is frequently used for visualizing DEMs in GIS software.

The use of hill shading provides additional advantages. First, it converts excessively smooth boundaries into sharp edges by emphasizing high frequency details of procedural noise images that are hardly recognizable in the original output from the procedural models (Fig. 9.4). Although octave values greater than four do

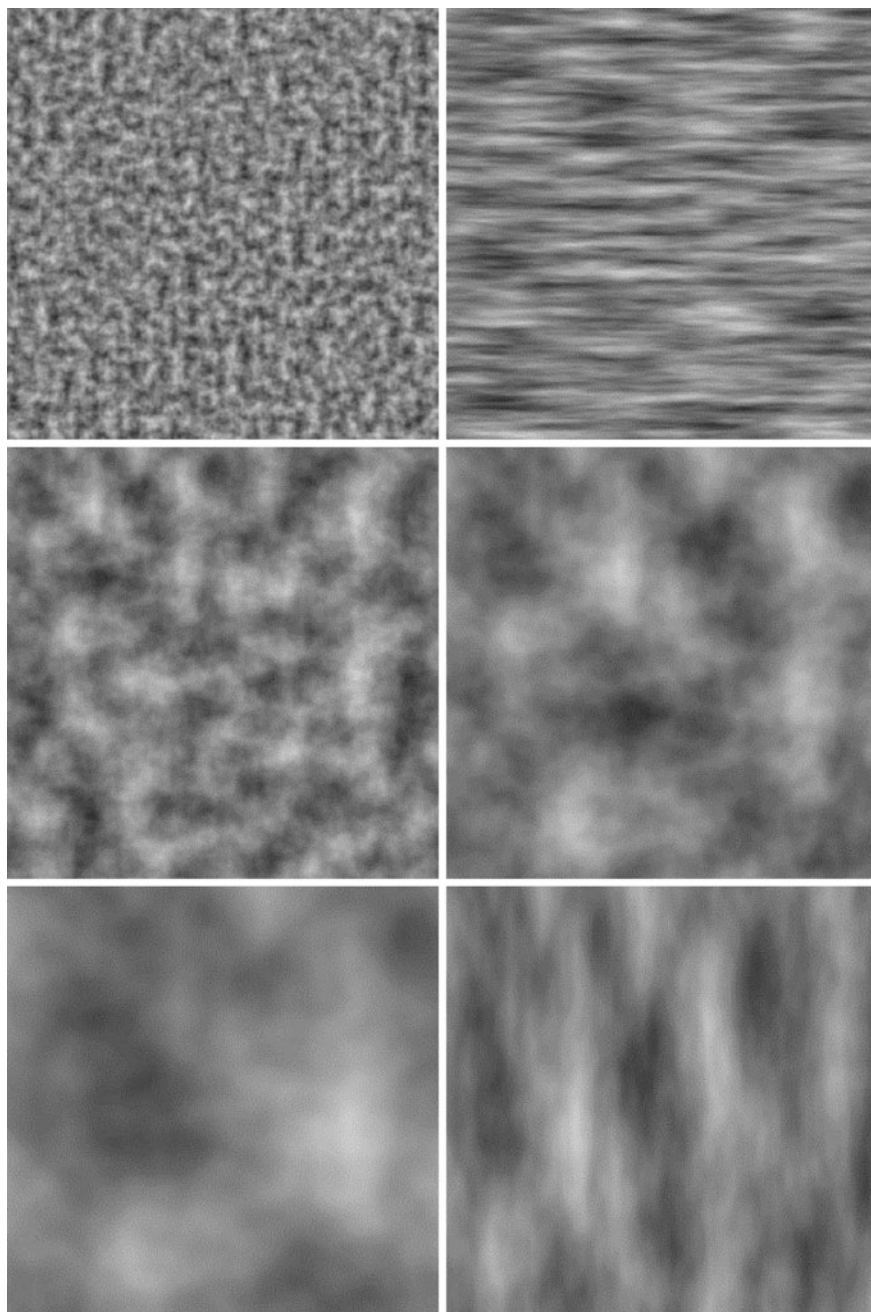


Fig. 9.3 Variations of fractal synthesis that result from different input parameters

Table 9.2 Relationships between structural keywords and procedural model parameters

Structure keyword	Features	Procedural model
Granular	<ul style="list-style-type: none"> • Small and rounded aggregates 	<ul style="list-style-type: none"> • Model: any model is acceptable • X and Y seed: smaller than 0.04
Subangular blocky	<ul style="list-style-type: none"> • Cubic (or similar) structural units • Rounded edges 	<ul style="list-style-type: none"> • Model: fractal synthesis or trigonometric jittering • X seed: 0.25–0.75 • Y seed: 0.25–0.75 • Octave: 4–16 • Frequency: 2.0~ (trigonometric jittering only)
Prismatic	<ul style="list-style-type: none"> • Cubic (or similar) structural units • Structural units are taller than they are wide 	<ul style="list-style-type: none"> • Model: fractal synthesis • X seed: 0.25–0.55 • Y seed: 0.3–0.6, but greater than the X seed • Octave: 4–16
Platy	<ul style="list-style-type: none"> • Structural units are wider than they are tall 	<ul style="list-style-type: none"> • Model: fractal synthesis or trigonometric jittering • X seed: greater than 0.5 • Y seed: smaller than 0.5 • Octave: 4–16

not add clearly noticeable details in the raw output of procedural noise images, hill shading makes these details visible so that we are able to obtain better appearance (Fig. 9.4).

9.4 Results and Discussion

Figure 9.5a, b shows an example of the recreation of a granular structure. We use small values for both X and Y seeds in order to achieve fine unit structures. Figure 9.5c, d shows recreation of platy soil structure using a greater X seed value than Y seed value. Figure 9.6a is a photograph of subangular blocky structure. Since subangular blocky structure has cube-like rounded shapes, we use trigonometric jittering as the procedural model (Fig. 9.6b). A high octave value adds further details. Figure 9.6c, d shows the regeneration of prismatic structure. Note that Y seed value is greater than X seed value to generate vertical structure. All results were generated at 1024 by 1024 pixel resolution and took <2 s to generate, but a 512 by 512 pixel image, which is sufficient for many purposes, can be generated in <1 s.

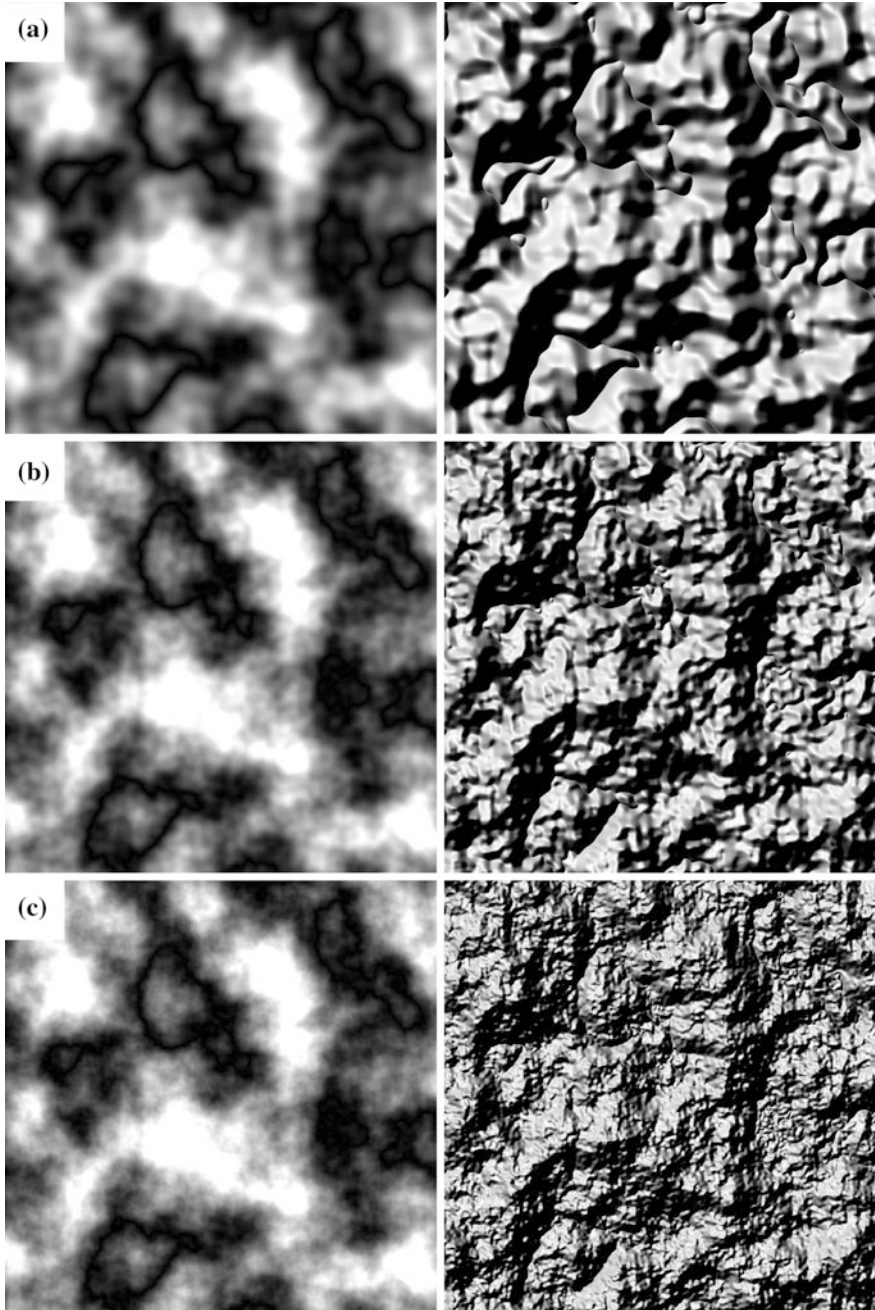


Fig. 9.4 Comparison between raw procedural noise (*left image*) and hill shading of the raw output (*right image*) for different octave values. **a** Octave value 4. **b** Octave value 5. **c** Octave value 6

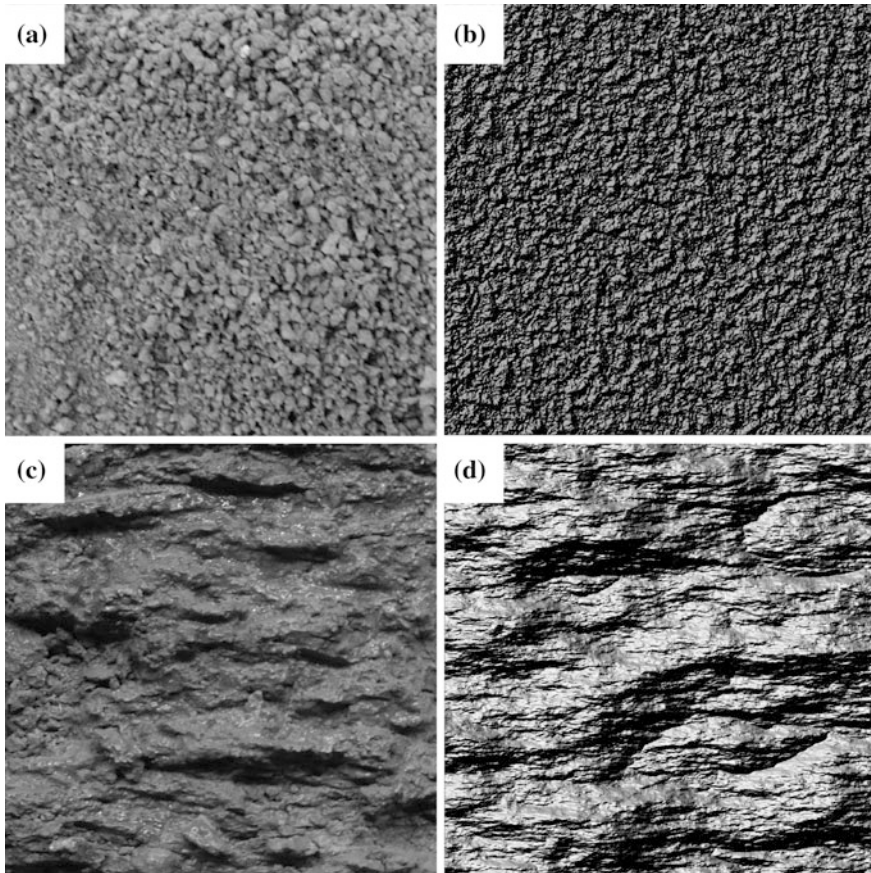


Fig. 9.5 **a** Photograph of granular soil structure. **b** Procedural model of granular structure using fractal synthesis (X seed = 0.037, Y seed = 0.037, octave = 6). **c** Photograph of platy soil structure. **d** Procedural model of platy soil structure using trigonometric jittering (X seed = 0.82, Y seed = 0.375, octave = 16, frequency = 2.17)

The procedural models described above have some drawbacks, however. First, due to the innate properties of the procedural approach, local feature control (i.e., controlling macrostructure) is very difficult. Adding specific macrostructure, for example a large prism, will require a redesign of the entire model so that pre-generated data will not be necessary. Second, the approach currently does not take into account blending between two different structures. Structure transitions in a soil profile are a subtle but important feature of soil profiles and we cannot yet depict it by using interpolations.

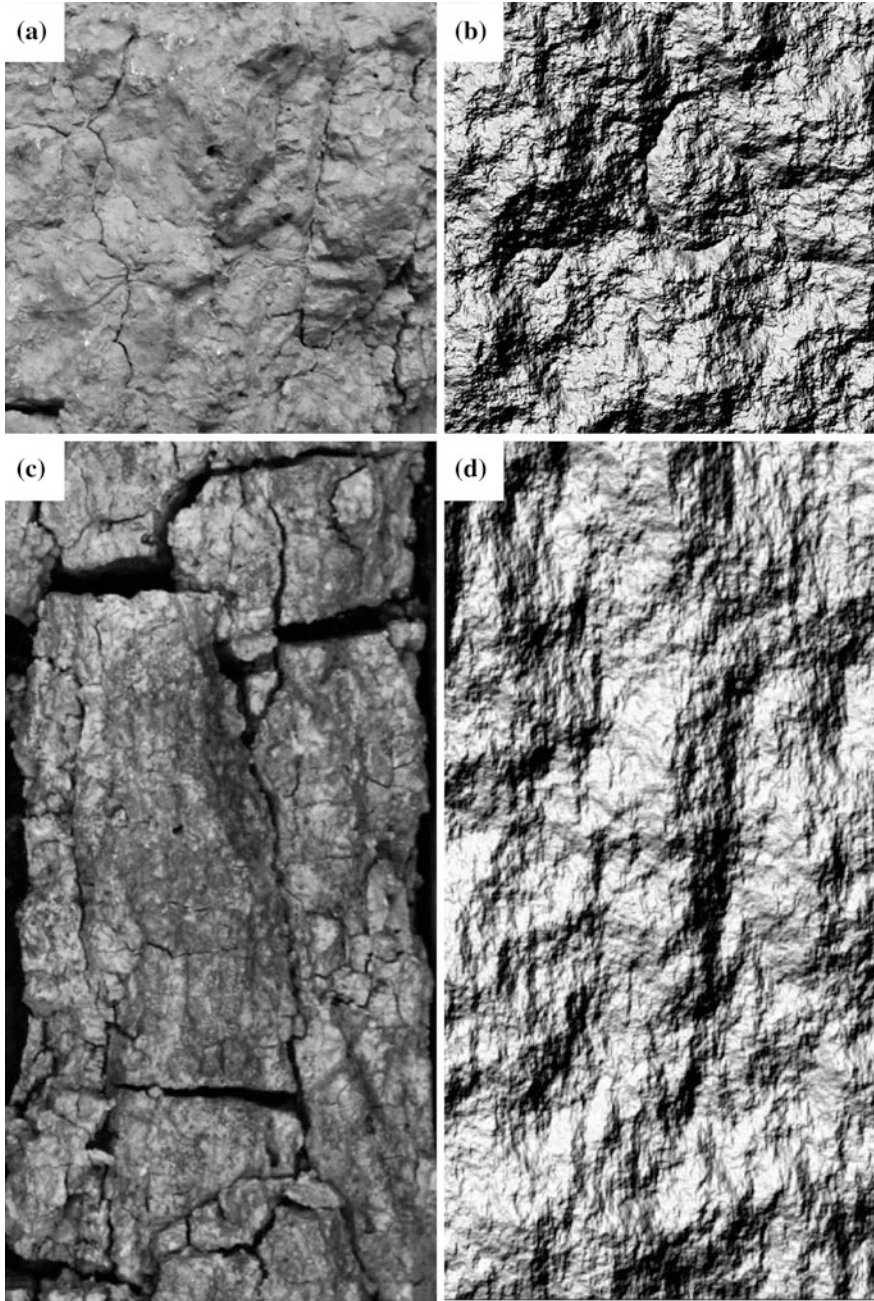


Fig. 9.6 **a** Photograph of subangular blocky soil structure. **b** Procedural model of subangular blocky soil structure using trigonometric jittering (X seed = 0.25, Y seed = 0.25, octave = 16, frequency = 2.17). **c** Photograph of prismatic soil structure. **d** Procedural model of prismatic soil structure using fractal synthesis (X seed = 0.25, Y seed = 0.375, octave = 10)

9.5 Conclusions and Future Work

We introduce a procedural model reconstruction framework for soil morphology data. Our framework takes tabular soil morphological data and converts it into visual features utilizing procedural models. The models generate appropriate procedural noise images that are then processed by a visualizer that applies hill shading to obtain images that are close approximations to actual soil structures. This approach will enable users to reconstruct soil structures with a minimum of data overhead.

There are several possible avenues for future work. As mentioned above, support for macrostructure would be useful. Also, good blending options to produce appropriate transitions from one type of soil structure to the next are still needed in order to simulate transitions between soil horizons.

So far, we have used user intuition to find appropriate procedural models and input parameters to generate images that look like different soil structures. In future work, we would like to investigate automatic inverse procedural approaches that would match an actual image of soil structure to an image generated by an appropriate procedural model. This may open the way to automatically quantifying soil structure without having to rely on the judgment of human observers.

Acknowledgements This material is based upon work supported by the Cooperative State Research, Education, and Extension Service, U.S. Department of Agriculture, under Award No. 2013-70003-20924. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the view of the U.S. Department of Agriculture.

References

- Ebert DS, Musgrave FK, Peachey D, Perlin K, Worley S (2002) *Texture & modeling: a procedural approach*. 3rd ed. Morgan Kaufmann
- Horn BKP (1981) Hill shading and the reflectance map. In: *Proceedings of the IEEE*, vol 69, pp 14–47
- Parish YI, Müller P (2001) Procedural modeling of cities. In: *Proceedings of the 28th annual conference on computer graphics and interactive techniques*. ACM, pp 301–308
- Prusinkiewicz ALP, Lindenmayer A, Hanan JS, Fracchia FD, Fowler D (1990) *The algorithmic beauty of plants*. Springer, New York
- Schoeneberger PJ, Wysocki DA, Benham EC, Soil Survey Staff (2012) *Field book for describing and sampling soils*, Version 3.0. Natural Resources Conservation Service, National Soil Survey Center, Lincoln, NE
- Smelik RM, Tuteneel T, Bidarra R, Benes B (2014) A survey on procedural modelling for virtual worlds. *Comput Graph Forum* 33(6):31–50
- Stava O, Pirk S, Kratt J, Chen B, Měch R, Deussen O, Benes B (2014) Inverse procedural modelling of trees. *Comput Graph Forum* 33(6):118–131
- Stava O, Benes B, Měch R, Aliaga DG, Krištof P (2010) Inverse procedural modeling by automatic generation of L-systems. *Comput Graph Forum* 29(2):665–674
- Vanegas CA, Garcia-Dorado I, Aliaga DG, Benes B, Waddell P (2012) Inverse design of urban procedural models. *ACM Trans Graph* 31(6):168