

A Discrete Modelling of Soil Fragments Transport by Runoff

Gilles Valette^{1,2}, Stéphanie Prévost¹, Laurent Lucas¹, and Joël Léonard²

¹ CReSTIC/SIC/MADS, EA3804 University of Reims Champagne-Ardenne, Reims, France

laurent.lucas@univ-reims.fr

² INRA UR1158 Agronomie Laon-Reims-Mons, France

Abstract. We aim to model and visualize the evolution of the surface structure of a cultivated soil surface during rainfall. In this paper, we briefly present our model, based on an Extended Cellular Automaton, and the different simulated processes. Among these processes, we focus on runoff which is of high relevance as it drives the evolution of the soil surface structure by transporting and depositing the detached fragments of soil and thus inducing an evolution in the granulometry of the surface material. We propose a simple algorithm to model, in a discrete way, runoff and also the transport and deposition of soil fragments according to their size. In that way we are able to derive information about the evolution of soil surface granulometry. A validation of the runoff model is proposed, based on the comparison of the results obtained with results from a numerical solution of the Saint Venant's equations. Although no validation was attempted for transport, simulations yielded visually promising results.

1 Introduction

The objective of our project, called SoDA (Soil Degradation Assessment), is to develop and validate a dynamic simulation of the evolution of the surface structure of a cultivated soil surface during a rainfall at the meter scale, keeping in mind a constant care for visualization. Major aspects of this evolution are the formation of soil crusts and the development of cracks [1, 2, 3]: presence of a crust due to rainfall can mechanically inhibit seedling emergence whereas presence of cracks can allow seedlings to break through the soil surface. Therefore, such a model, able to predict soil structure under different initial soil conditions and climatic scenarios, would be for example a useful tool to select adequate tillage and sowing practices.

Crusts form during rainfall because of the detachment of soil particles by raindrops and their subsequent redistribution, either by projection or transport in the flow [4]. The size of the fragments produced by the desaggregation processes varies with the initial size distribution of particles, the hydric state of the soil, the intensity of the stresses exerted by raindrops. The size of soil fragments determines if runoff will be able to transport them or not. It is thus very important to have and to keep some information about particle size, as some types of crusts —like sedimentary crusts— are precisely defined by the occurrence of

sorting during sedimentation. This is one originality of our simulator to permit this tracking of the soil granulometry.

A primary version of our simulator has been presented in a previous article [5]. The aim of the present paper is to present our new model of runoff and transport of soil fragments, which is based on Cellular Automata rules which are simple interpretations of equations from hydraulics. The previous model used a single equilibrating law between two cells, thus without any physical principle, and was not parallelizable. We have validated this new model by comparing our results to the ones obtained using a numerical solution to the Saint Venant's equations [6]. Finally, we show that our model permits the simulation of the temporal and spatial dynamics of soil fragments sorting to produce some visually convincing results.

2 Related Work

We present in this section a brief review of models of transport of solid material by runoff found in the field of Computer Imagery and Soil Science followed by a few comments.

Musgrave et al. [7] proposed a new method for creating mountain fractal terrains with the use of height fields. They used a report algorithm to model runoff and an *ad hoc* model of hydraulic erosion. Beneš and Forsbach [8, 9] introduced a new data structure for visual simulation of 3D terrains. They divided the hydraulic erosion process into four independent steps that can be applied independently. The authors modeled the runoff with a report algorithm on a 8-neighbourhood, they used an empirical saturation level to calculate the amount of sediments in water, and they considered evaporation but not infiltration. Chiba et al. [10] presented a simple “quasi physically based” method for simulating the topography of eroded mountains based on velocity fields of water flow. In this method, the velocity fields of water flowing down the face of a mountain are calculated by simulation of the motion of “water particles”, and these velocity fields are used for simulating erosion and sedimentation. Neidhold et al. [11] presented a real-time method that combined a non-expensive fluid simulation based on a Newtonian physics approach on a two dimensional grid storing acceleration, velocity and mass. They provided an erosion algorithm. Their method does not take into account infiltration, and it uses specific constants, whose value has to be arbitrarily given by the user. Beneš et al. [12] introduced a full 3D hydraulic erosion, by coupling Navier-Stokes equations with cohesive or cohesionless material transportation and solving them on a 3D grid. Recently, shallow water fluid models were used by Beneš [13] and Mei et al. [14] in order to provide real-time hydraulic erosion based on 2D regular grids.

In the field of Soil Science, prediction of soil erosion generally means prediction of soil loss rather than analysis of the evolution of the soil surface and its relief. Current models for soil erosion by water are WEPP (Water Erosion Prediction Project) [15] and EUROSEM (European Soil Erosion Model) [16]. Favis-Mortlock et al. [17] developed the “RillGrow”. In this model emphasis is put on the evolution of the soil surface rather than on sediment exportation.

D'Ambrosio et al. [18, 19] and Avolio et al. [20] suggested a Cellular Automata (CA) model for soil erosion by water. This model involves a larger number of states, including altitude, water depth, total head, vegetation density, infiltration, erosion, sediment transport and deposition. A similar CA approach, based on the model of “*precipitons*” introduced by Chase [21], is used by Luo et al. [22] for the WILSIM (Web-based Interactive Landform Simulation Model) project. In order to simulate landscape erosion and deposition, Haff [23] uses a CA model called “*waterbot*”. Servat [24] suggested a description of flows in terms of heterogeneous agents that interact in a continuous space. This research has led to the development of the RIVAGE simulator of runoff and infiltration, with some additional trials to incorporate erosion processes.

Most of the models presented share the same principles, whatever the objectives, the scale or the processes considered. For example, they almost have in common the principle of flow of water or material according to the greatest gradient, or the transport capacity concept. On that point, our simulator does not significantly differ from other models. However, Computer Imagery models aim visually plausible results and not simulation, thus they often make simplifications about the processes considered and they cannot apply in our simulator. For example, these models assume that the material is dissolved by running water. At the scale of our simulator, runoff is not capable of detachment: the fragments which are mobilized must exist and have been created by the raindrops energy. Moreover, these models are often based on some user-defined parameters and quantities which are not directly linked to the properties of soil or water. On the contrary, our model uses formulas and parameters which are in adequation with the current knowledge of soil scientists. Finally, our model considers explicitly a 3D space and the processes of splash and infiltration, which are rarely considered. It also permits the precise tracking of the granulometry of detached and transported particles of the soil (as soon as it has been modified). This is its main originality and it is an important issue as it allows to describe a spatial (vertical and horizontal) and temporal evolution in the state of the soil, and to establish a relationship between this evolution and the processes at work.

3 Our Model

The simulator is based on an Extended Cellular Automaton [18, 20]. The patch of soil which the simulator has to manage is typically of 50cm x 50cm x 10cm, discretized into 2mm side cubic cells. Runoff can be considered as a centrimetric scale process and thus, we could use a larger resolution, i.e., 1cm instead of 2mm, but we need a millimetric resolution to treat with a good accuracy the changes in the altitude due to the transport of the smallest fragments. The cells contain variable proportions of solid, water and air. In order to be able to keep an information on the granulometry of the soil, the solid phase can be constituted of “continuous” matter (i.e., not fragmented) or of discrete particles which are distributed into 6 classes, from 0 to 2mm. The rainfall events are also discretized by raindrops whose size and number vary through time. The 3D-grid containing

the soil cells is completed with three 2D-grids: one to store the height of the terrain at the center of each surface cell, one for flow depth, one for the particles transported in the flow. The movement of water is due to runoff and infiltration. We don't take into account the effect of evaporation because the infiltration rate is always much greater than the evaporation rate, especially during a rainfall. In its main loop, our simulator considers successively the following processes:

- the raindrops generation, following a gamma distribution depending on the rain intensity;
- the soil detachment, which is proportional to the kinetic energy of each raindrop, and which is corrected with respect to the thickness of the surface water;
- the soil fragmentation, estimated from tests of structural stability;
- the projection by splash of the soil fragments, to a distance which is proportional to the momentum of each raindrop and which depends on the size of the fragments;
- infiltration, which is in the current version simply calculated with the physically based Green-Ampt method [25, 5], but will be replaced in a future work by a cellular automata infiltration model which will be coupled to the evolution of the soil surface structure;
- runoff, which is simulated using a report algorithm depending on the slope of the water surface. The altitude of the water surface is equal to the sum of flow depth and terrain height, and we use it to represent the hydraulic head, neglecting the kinetic component which is always very small in our study context.
- the transport and the deposition by runoff, with the hypothesis that runoff is only an agent of transport and does not play any role in the detachment of the fragments, which is true at the scale considered.

4 Runoff

Our modelling of the runoff process takes into account the flow depth and the altitude of the soil surface, the sum of these two quantities being used to define the total hydraulic head. For each iteration of our algorithm, for each cell, we have to determine how much water is transferred and where it is transferred. We have chosen to transfer water from one source cell to a unique target cell only, in the direction of the highest head gradient. This is a simplification of the reality, but as the gradient is changed when water has been transferred to a downstream cell, it remains possible that water on a cell which has only lower elevation neighbours be transferred all around during a few iterations. To compute the quantity of water which has to be transferred, we must consider four points: i) the discharge must be accurate; ii) the mass of water must be conserved; iii) the state of a cell established during the current iteration must not be necessary for establishing the state of another cell in the same iteration; iv) the algorithm has to converge to a stable state. The third point is necessary for two reasons: firstly because it is the correct behaviour in a CA model, secondly because this independence is required in order to get a parallelizable code.

4.1 Algorithm

In our runoff algorithm we treat all the surface cells from the lowest head cell to the highest head cell to calculate the quantity of water received by a cell and where it comes from. The pre-ordering of the cells allows to satisfy the constraint that a cell can transfer water only to the neighbouring cell which has the lowest hydraulic head. If we treated the cells receiving water in an arbitrary order, it would be possible that water be transmitted to a cell not being the lowest hydraulic head cell in the neighbourhood of the source cell (an example is shown Fig. 1). By using this pre-ordering and marking each cell which gives water, we insure that the current cell receives water only from unmarked neighbours. Instead of ordering the cells, the same result could be obtained by examining an extended neighbourhood of each cell, but the computational cost would be much higher. As we consider the 2D-space formed by the surface cells, the neighbourhood used can be the von Neumann neighbourhood (also called 4-neighbourhood) or the Moore neighbourhood (also called 8-neighbourhood) [19]. We have tried both these neighbourhoods and observed some differences (see Section 7). Notice that we use the standard CA behaviour, i.e., for one cell the transition function applies only to the adjacent cells of this cell. Thus, the size of the neighbourhood is limited. For the same reason, we do not have to consider a discrete distance computation, e.g., chamfer methods, because we only need the distance between two adjacent cells, which we consider as the Euclidean distance between their centers.

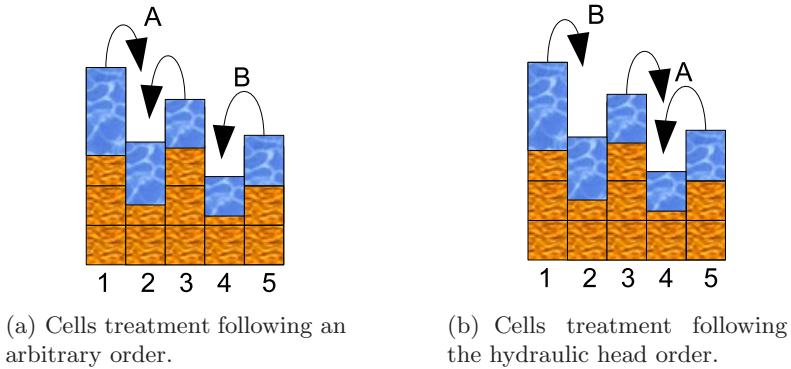


Fig. 1. Why the cells must be ordered: in (a) an arbitrary order from cell 1 to cell 5 is considered, cell 2 is treated before cell 4, thus cell 3 gives water to cell 2 which is not the lowest hydraulic head cell in its neighbourhood, in (b) cells are ordered by their hydraulic head, cell 4 is now treated before cell 2, thus cell 3 gives water to cell 4, which is correct

In our method, boundaries are easy to treat. We distinguish three cases. If they are considered as walls, they are simply ignored and there is nothing to do. If the terrain is seen as a torus, a cell on the boundary can receive water from

its opposite cells, so it suffices to add these cells in its neighbours. Finally, if boundaries are considered as holes, we treat them as “ghost” cells that are the lowest cells of the terrain. So we have to do a first pass in order to calculate the quantity of water these cells receive from their immediate neighbour, in the perpendicular direction relative to the boundary. This neighbour is marked, so it does not give any more water.

Finally, it is worth noticing that our algorithm is fully parallelizable, because the calculations are independant and exclusively established from the state obtained in the previous iteration. The ordering of the cells is not an issue for the majority of the cells, because it can be done locally, e.g., on the portion of the soil treated by a core. However, we have to treat differently the cells which are at the frontier of two cores by examining their extended neighbourhood, i.e., the neighbourhood of each (possible) emitting cell, in order to eliminate the cells which are adjacent to a cell with a lower hydraulic head.

4.2 Flow Discharge Calculation

The flow q from a source cell i to a target cell c is calculated using the Darcy-Weisbach equation [26]:

$$v = \sqrt{\frac{8 g S(c, i) W_i}{f}} , \quad (1)$$

combined to the definition of the flow:

$$q = v W_i dx , \quad (2)$$

where v is flow velocity, $S(c, i)$ is the friction slope, which we assume to be the slope of altitude plus water depth, W_i is the water depth, g is the acceleration due to gravity, dx the spatial step, and f is the Darcy-Weisbach friction factor. f depends on soil roughness and strongly determines flow velocity; it is an init parameter of our simulator.

4.3 Equilibrating Flow Calculation

One main issue in the report algorithm is that the quantity of water transferred to a cell must not be higher than the difference in hydraulic head between the source cells and the target cell, because the target cell would then become a source cell at the next iteration, and so on, causing instability. For this reason we have to calculate a maximum quantity of water that a cell can receive from each neighbour. As a cell can receive water from up to 8 cells (or 4, depending on the chosen neighbourhood), we have to find a way to calculate this maximum quantity. It is worth noticing that to permit this equilibrating flow calculation, we choose to consider a cell receiving water from its neighbours, and not a cell transferring water. This calculation uses the following definitions:

- W_i^t is the depth of water on the cell i at iteration t ;
- H_i is the height of the soil on the cell i , considered as constant;

- $\varphi_i^t = H_i + W_i^t$ is the hydraulic head of the cell i ;
- $\delta_i = W_i^t - W_i^{t+1}$ is the emitted water from cell i (positive quantity);
- $\delta_c = W_c^{t+1} - W_c^t$ is the received water in cell c (positive quantity);
- A is the set of not marked neighbouring cells of central cell c ;
- n is the cardinal of A , i.e., the number of cells which have to give water to c .

In order to avoid instability, the receiving cell c (i.e., the central cell of the neighbourhood) must always have a smaller hydraulic head than that of all the emitting cells i :

$$\forall i \in A \quad \varphi_c^t \leq \varphi_i^t . \tag{3}$$

Considering this condition at iteration $t+1$ and using the definitions of φ_i^{t+1} , φ_c^{t+1} , δ_i , δ_c it comes:

$$\forall i \in A \quad \varphi_c^t + \delta_c \leq \varphi_i^t - \delta_i . \tag{4}$$

The mass conservation implies that the quantity of water received should be equal to the quantity of water emitted:

$$\sum_{i \in A} \delta_i = \delta_c . \tag{5}$$

(4) and (5) give the condition for stability:

$$\forall i \in A \quad 2\delta_i + \sum_{j \in A - \{i\}} \delta_j \leq \varphi_i^t - \varphi_c^t . \tag{6}$$

If we consider the limit, i.e., the equivalence between these quantities, we get this matricial equation:

$$\begin{pmatrix} 2 & 1 & \dots & 1 \\ 1 & 2 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 2 \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_2 \\ \dots \\ \delta_n \end{pmatrix} = \begin{pmatrix} \varphi_1^t - \varphi_c^t \\ \varphi_2^t - \varphi_c^t \\ \dots \\ \varphi_n^t - \varphi_c^t \end{pmatrix} . \tag{7}$$

The first matrix can be easily inverted, thus we obtain the formulation of the solution:

$$\delta_i = \frac{n}{(n+1)}(\varphi_i^t - \varphi_c^t) - \sum_{j \in A - \{i\}} \frac{\varphi_j^t - \varphi_c^t}{(n+1)} . \tag{8}$$

In order to have a physically correct behaviour, we must insure that this value δ_i is always positive. Therefore, we eliminate from the neighbourhood A the cells for which this value is negative before recomputing all the values for this new neighbourhood. Finally, the transferred quantity of water is the minimum between δ_i and q , so the inequality (6) is always verified and the risk of instability is eliminated.

5 Mobilization and Deposition

Mobilization and deposition are based on a transport capacity approach. The transport capacity of the flow is defined by its maximal sediment concentration \mathcal{C} . The value of \mathcal{C} depends on flow conditions. The quantity of matter \mathcal{M} that a given flow can transport is then given by the product of the volume of water \mathcal{W} and the maximal sediment concentration \mathcal{C} . Transport capacity equations of the form $C \propto (\tau - \tau_c)^b$, where τ is the shear stress, τ_c is the critical shear stress, above which significant transport occurs and whose values can be found in the soil science literature, and b a parameter often larger than one, have been shown to be generally appropriate [27]. We thus have chosen to use this kind of formulation to estimate transport capacity for the class p :

$$\mathcal{C}(p) = a_p \left(\tau - \tau_c(p) \right)^{b_p}, \quad (9)$$

and we used a sediment transport database [27] to derive parameters $\tau_c(p)$, a_p and b_p of (9) for each of the size classes considered. The shear stress is calculated from $S(i, j)$, the total head slope between the source cell i and the target cell j , and the water depth W_i according to the following equation [26]:

$$\tau = \rho_w g W_i S(i, j). \quad (10)$$

The first step of our algorithm is the comparison between $\tau(i)$ and the threshold $\tau_c(p)$ to eliminate the classes of fragments for which no mobilization is possible: for these classes, we deposit all the fragments. From the other classes, we calculate the maximum sediment concentration \mathcal{C}_{max} using (9) and the mean weighted diameter. This value permits us to obtain the maximal mass of sediment \mathcal{M}_i we can have in the flow. If this mass is greater than the mass already present in the flow, we can take off some available fragments and put them in the flow. On the opposite, if this mass is less than the mass already present in the flow, deposition occurs, i.e., particles are added to the cell considered. To be able to mobilize the biggest fragments, we begin the loop starting from the coarsest particles, until we get the needed quantity. Finally, as the transformations are purely local, i.e., depending only on the state of the cell and made inside this cell, it is correct to apply them directly in a loop treating all cells in any order.

6 Transport

When a quantity of water moves because of runoff, it transports a proportional quantity of mobilized matter to the cell of destination. In this purpose, we calculate first the total mass of fragments mobilized in the source cell and deduce from this value the mass of fragments we have to move to the target cell. As we work with discrete number of fragments, in many cases it is impossible to move this exact quantity. Moreover, this quantity is often too small to move one of the biggest fragments. If this deficit was repeated too frequently, it would introduce

a bias in the transport of fragments. In order to avoid this, we have chosen to randomly begin this movement from the biggest class or from the smallest class, with an equal probability. In that way, we insure first to sometimes move some big fragments, and second to compensate this excess of transported mass in a following iteration.

7 Results

7.1 Validation of the Runoff Algorithm

For validation purpose, we simulated rainfall over a 30 cm x 2 cm terrain having a constant slope of 5%. The rain intensity was 30mm/h and the duration of the rainfall was 30s. Infiltration was inhibited, the soil being considered as impermeable. The friction factor f was set to a value of 1. In Fig. 2(a) a plot of cumulated runoff at the outlet of the terrain versus cumulated rainfall shows that water conservation is satisfied. Fig. 2(b) shows that the discharge at the outlet of the terrain reaches rainfall intensity in about 15s, keeps this value as long as rainfall continues, and then decreases to zero in about 50s after rainfall has stopped. We have compared these results to results obtained using a numerical solution of the classical Saint Venant's equations [6]. Both results are in close agreement. Simulations done with both a 4-neighbourhood and an 8-neighbourhood gave similar results but the flow appears more regular with the 4-neighbourhood.

7.2 Visual Result of Soil Fragments Transport

In order to test the transport algorithm, we used a terrain combining 3 different slopes: 10% over the first 20cm, 40% over 20cm, and 5% over 30cm. We limited

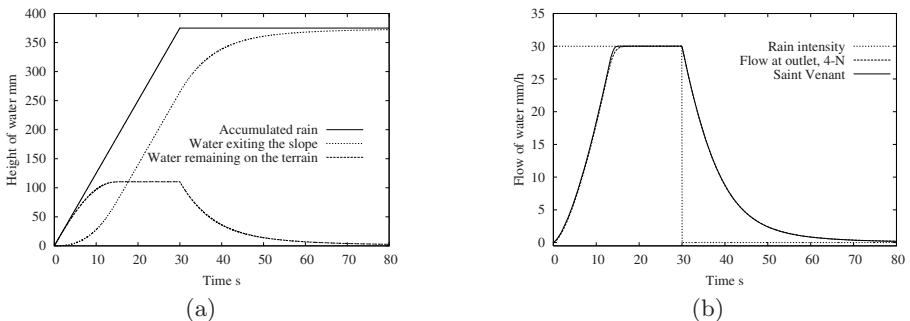


Fig. 2. (a) Verification of the conservation of the water during a simulation. (b) Comparison between the simulated discharge at the outlet of the terrain with a 4-neighbourhood and the curve given by a numerical solution of the Saint Venant's equations. The curve 8-neighbourhood is very close to the 4-neighbourhood but with only more oscillations on the flat part.

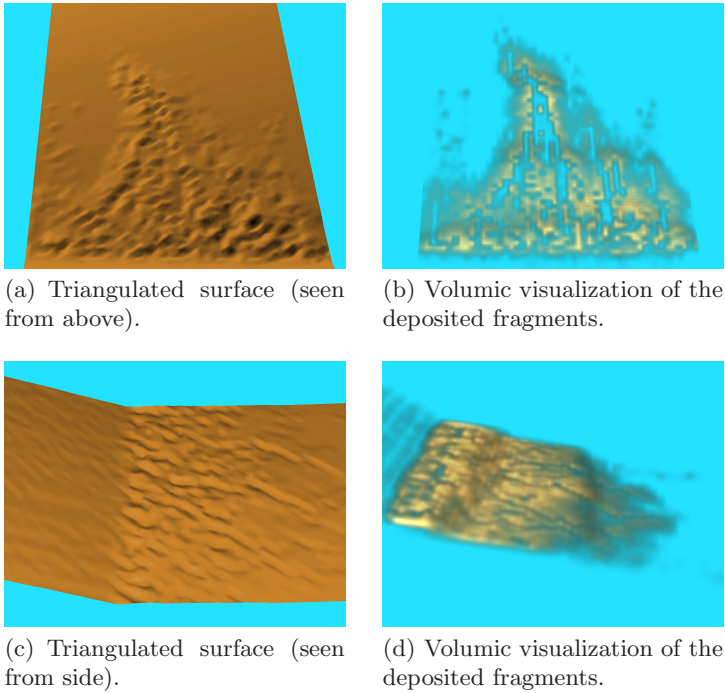


Fig. 3. Visual result of transport by runoff along a slope at its bottom: (a) and (b) with a 4-neighbourhood, (c) and (d) with an 8-neighbourhood

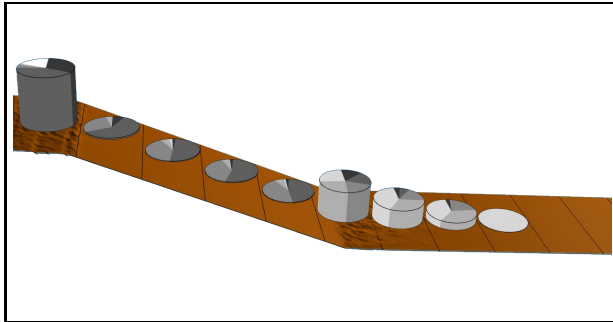


Fig. 4. Visualization of the granulometry: the height of a cylinder indicates the total mass of the fragments deposited. The levels of grey indicate the class of the fragments: white is the smallest size, black the biggest size.

rainfall to the first part of the terrain. Two simulations of 30 min duration, with a rainfall intensity of 30mm/h and without infiltration have been done: one with a 4-neighbourhood, the second with an 8-neighbourhood. Fig. 3 shows the results of these simulations, with both a triangulated surface, rendered by OpenGL with the Phong's illumination model, and a volumic visualization of the deposited

fragments, rendered with a direct volume rendering algorithm [28] which takes advantage of the natural bijection between the cells of our model and the voxels. Among a few differences, it is noticeable that the fragments went further and that less fragments were deposited on the slope with the 4-neighbourhood than with the 8-neighbourhood.

7.3 Sorting During Transport

Results from the simulation with the 4-neighbourhood (Fig. 4) show that soil fragments are numerous and of all sizes on the top of the terrain, where rainfall occurs and particles are created. However, soil fragments are almost absent on the second slope, because there is no rainfall here to produce particles, and existing ones are transported and not deposited by runoff because of the steep slope. At the bottom of the terrain, which has a lower slope, it can be observed that the biggest fragments were deposited here, and that their number and size are decreasing with the distance. This behaviour is very close to what can be currently observed in the nature and shows that the simulator adequately represents, at least qualitatively, the sorting of the soil material during transport.

8 Conclusion and Future Work

We develop a simulator of the evolution of soil surface structure of cultivated soils under rainfall, at the meter scale. This is an important issue as it has both theoretical and practical interest. In this paper, we have detailed the discrete model of runoff and transport of soil fragments we have implemented in this simulator. Our runoff model was shown to give results which are in close agreement with results from a numerical solution of the Saint Venant's equations. Results from the transport model have not been compared to experimental ones, but they were very close to what can be observed, which is an indication that the simulator adequately represents the sorting of the soil material during transport by runoff. This is an important issue for us as this sorting is an essential process of the development of sedimentary crusts, and the prediction of the dynamics —both temporal and spatial— of soil crusting is the main objective of the simulator. As a future work, we have to establish and implement a model of infiltration based on the same principles than the runoff model. We also want to document more precisely the consequences of the choice between the 4-neighbourhood and the 8-neighbourhood. Finally, a real experiment of rain simulation will be realized in order to calibrate the parameters of the simulator and to validate all the simulated processes.

Acknowledgment

This work is part of the SoDA project, supported by the regions of Champagne-Ardenne and Picardie (information and multimedia material available at *www.soda-project.com*).

References

1. Hamblin, A.P.: The Influence of Soil Structure on Water Movement, Crop Root Growth, and Water Uptake. *Advances in Agronomy* 38, 95–158 (1985)
2. Baranowski, R., Bakowski, B.: The Influence of Differentiated Soil Structure on Temperature Dynamics. *Roczniki Gleboznawcze* 28(1), 37–44 (1977)
3. Valette, G., Prevost, S., Lucas, L.: Modeling and visualization of cracks in a desiccating soil. In: 11th International Fall Workshop Vision, Modeling, and Visualization, Aachen, Germany, pp. 177–184. Aka Verlag, Berlin (2006)
4. Le Bissonnais, Y., Bruand, A.: Crust Micromorphology and Runoff Generation on Silty Soil Materials During Different Seasons. *Catena Supplement* (24), 1–16 (1993)
5. Valette, G., Prévost, S., Lucas, L., Léonard, J.: SoDA Project: a Simulation of Soil Surface Degradation by Rainfall. *Computers & Graphics* 30(4), 494–506 (2006)
6. Zhang, W., Cundy, T.W.: Modeling of two-dimensional overland flow. *Water Resources Research* 25, 2019–2035 (1989)
7. Musgrave, F.K., Kolb, C.E., Mace, R.S.: The Synthesis and Rendering of Eroded Fractal Terrains. In: *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 41–50. ACM Press, New York (1989)
8. Beneš, B., Forsbach, R.: Layered Data Representation for Visual Simulation of Terrain Erosion. *EEE Proceedings of SCCG* 25(4), 80–86 (2001)
9. Beneš, B., Forsbach, R.: Visual Simulation of Hydraulic Erosion. *Journal of Winter School of Computer Graphics* 10, 79–94 (2002)
10. Chiba, N., Muraoka, K., Fujita, K.: An Erosion Model Based on Velocity Fields for the Visual Simulation of Mountain Scenery. *The Journal of Visualization and Computer Animation* 9, 185–194 (1998)
11. Neidhold, B., Wacker, M., Deussen, O.: Interactive Physically Based Fluid and Erosion Simulation. In: Galin, E., Poulin, P. (eds.) *Eurographics Workshop on Natural Phenomena*, pp. 25–32 (Dublin) (2005)
12. Beneš, B., Těšínský, V., Hornyš, J., Bhatia, S.: Hydraulic erosion. *Computer Animation and Virtual Worlds* 17(2), 99–108 (2006)
13. Beneš, B.: Hydraulic Erosion by Shallow Water Simulation. In: *The 4th Workshop on Virtual Reality Interaction and Physical Simulation - Vriphys 2007* (to appear)
14. Mei, X., Decaudin, P., Hu, B.: Fast Hydraulic Erosion Simulation and Visualization on GPU. In: *Pacific Graphics 2007* (to appear)
15. Lane, L., Nearing, M.: USDA - Water Erosion Prediction Project: Hillslope Model. Technical Report 2, USDA-ARS National Soil Erosion Research Laboratory, NSERL, West Lafayette, Indiana, USA (1989)
16. Morgan, R., Quinton, J., Smith, R., Govers, G., Poesen, J., Auerswald, K., Chisci, G., Torri, D., Styczen, M.: The European Soil Erosion Model (EUROSEM): a Dynamic Approach for Predicting Sediment Transport from Fields and Small Catchments. *Earth Surface Processes and Landforms* 23(6), 527–544 (1998)
17. Favis-Mortlock, D., Boardman, J., Parsons, A., Lascelles, B.: Emergence and Erosion: a Model for Rill Initiation and Development. *Hydrological Processes* 14(11–12), 2173–2205 (2000)
18. D’Ambrosio, D., Gregorio, S.D., Gabriele, S., Gaudio, R.: A Cellular Automata Model for Soil Erosion by Water. *Physics and Chemistry of the Earth, Part B* 26(1), 33–40 (2001)
19. D’Ambrosio, D., Gregorio, S.D., Iovine, G.: Simulating Debris Flows Through a Hexagonal Cellular Automata Model: Sciddica S3-hex. *Natural Hazards and Earth System Sciences* 3, 545–559 (2003)

20. Avolio, M.V., Crisci, G.M., D'Ambrosio, D., Di Gregorio, S., Iovine, G., Rongo, R., Spataro, W.: An Extended Notion of Cellular Automata for Surface Flows Modelling. *WSEAS Transactions on Computers* 2, 1080–1085 (2003)
21. Chase, C.G.: Fluvial Landsculpting and the Fractal Dimension of Topography. *Geomorphology* 5, 39–57 (1992)
22. Luo, W., Duffin, K.L., Peronja, E., Stravers, J.A., Henry, G.M.: A Web-Based Interactive Landform Simulation Model (WILSIM). *Computers and Geosciences* 30, 215–220 (2004)
23. Haff, P.K.: Waterbots In: Landscape Erosion and Evolution Modeling. In: Harmon, R.S., Doe III, W.W. (eds.), Kluwer Academic/Plenum Publishers (2001)
24. Servat, D., Léonard, J., Perrier, E., Treuil, J.: The RIVAGE-Project: a New Approach for Simulating Runoff Dynamics. In: Feyen, J., Wiyo, K. (eds.) *Modelling of Transport Processes in Soils at Various Scales in Time and Space*, Wageningen Pers, Wageningen, The Netherlands, pp. 592–601 (1999)
25. Green, W.H., Ampt, G.A.: Studies on Soil Physics Part I: The Flow of Air and Water Soils. *Journal of Agricultural Science* 4(1), 1–24 (1911)
26. Chow, V., Maidment, D., Mays, L.: *Applied Hydrology*. In: *Water Resources and Environmental Engineering*, McGraw-Hill, New-York (1988)
27. Govers, G.: Evaluation of transporting capacity formulae for overland flow. In: Parsons, A., Abrahams, A. (eds.) *Overland flow. Hydraulics and erosion mechanics*, pp. 243–273. UCL Press, London (1992)
28. Benassarou, A., Bittar, E., John, N., Lucas, L.: MC Slicing for Volume Rendering Applications. In: Sunderam, V.S., van Albada, G.D., Sloat, P.M.A., Dongarra, J. (eds.) *ICCS 2005. LNCS, vol. 3515*, pp. 314–321. Springer, Heidelberg (2005)