Computational Science Technical Note **CSTN-196**

Interactive Simulation and Visualisation of Falling Sand Pictures on Tablet Computers

Bradley T. Pearce and K. A. Hawick

2013

Sand pictures are made from a mix of coloured sands and water or oil sandwiched between two sheets of glass are a common desktop amusement. However, they also provide a good example of mixing and layering in materials science. We construct a lattice-based simulation of a sand picture based around the Kawasaki spin-exchange model with empirical couplings between cells. A Monte Carlo stochastic dynamic scheme is used to update pairs of neighboring cells using a Boltzmann like energy controlled probability process. The sand cells then diffuse around, with a preference parameter for sand to adhere to other sand cells of the same or different types. This model can be perturbed with a preferred directional gravitational force that leads to nearly correct physical phase separation of the coloured sands. The model provides a visually realistic simulation that can be rendered in real time. We implement this using Android and Java on tablet computers with inbuilt gyroscopic sensors that allow the simulated system to adapt to real gravity in interactive time. We describe the model and the implementation and software architecture for this App and the associated performance tradeoffs. We discuss possible future performance improvements using graphical processing units and other tablet specific features.

Keywords: sand picture; App; tablet computer; gravity sensor; simulation

BiBTeX reference:

```
@INPROCEEDINGS{CSTN-196,
        author = {Bradley T. Pearce and K. A. Hawick},
        title = {Interactive Simulation and Visualisation of Falling Sand Pictures
                on Tablet Computers},
        booktitle = {Proc. 10th International Conference on Modeling, Simulation and Visualization
                Methods )MSV'13)},
        year = {2013},
        number = {CSTN-196},
        pages = {MSV2341},
        address = {Las Vegas, USA},
        month = {22-25 July},
        publisher = {WorldComp},
        institution = {Computer Science, Massey University, Auckland, New Zealand},
        keywords = {sand picture; App; tablet computer; gravity sensor; simulation},
        owner = {kahawick},
        timestamp = {2013.03.19}
}
```

Technical Report CSTN-196

Interactive Simulation and Visualisation of Falling Sand Pictures on Tablet Computers

B. Pearce and K.A. Hawick
Computer Science, Massey University, Albany, North Shore 102-904, Auckland, New Zealand
email: brad.pearce.nz@gmail.com, k.a.hawick@massey.ac.nz
Tel: +64 9 414 0800 Fax: +64 9 441 8181

February 2013

ABSTRACT

Sand pictures are made from a mix of coloured sands and water or oil sandwiched between two sheets of glass are a common desktop amusement. However, they also provide a good example of mixing and layering in materials science. We construct a lattice-based simulation of a sand picture based around the Kawasaki spin-exchange model with empirical couplings between cells. A Monte Carlo stochastic dynamic scheme is used to update pairs of neighboring cells using a Boltzmann like energy controlled probability process. The sand cells then diffuse around, with a preference parameter for sand to adhere to other sand cells of the same or different types. This model can be perturbed with a preferred directional gravitational force that leads to nearly correct physical phase separation of the coloured sands. The model provides a visually realistic simulation that can be rendered in real time. We implement this using Android and Java on tablet computers with inbuilt gyroscopic sensors that allow the simulated system to adapt to real gravity in interactive time. We describe the model and the implementation and software architecture for this App and the associated performance tradeoffs. We discuss possible future performance improvements using graphical processing units and other tablet specific features.

KEY WORDS
sand picture; App; tablet computer; gravity sensor; simulation.

1 Introduction

Sand pictures made from different grain sizes of sand suspended in oil or water and sandwiched between sheets of glass are common desktop toys. These complex fluid systems exhibit interesting layering and other turbulent patterns when they are inverted and the sand is allowed to fall in various ways.

Figure 1: A Sand picture (Photographed real system.)

Figure 1 shows a photo of a real sand picture system - typically of dimensions around 20 cm across and with two or three different sand-species in the mixture, these models are often mounted on a rotatable frame so the picture can be inverted to create new swirling patterns until a new equilibrium is reached.

The picture is close to two dimensional in that the enclosing glass plates usually sandwich a very thin layer of sand and fluid - often less than 1mm. Sand grain sizes will typically range from 0.1 to 1 mm in diameter, and so there is enough room for a few grains of sand to flow past one another but this is not the dominant effect observed. It is sand weighing down on sand and sand displacing fluid that drives the layering and pattern generation.

A three-dimensional model can be built too of course, but since the opaque sand precludes seeing inside the system there is little artistic value in adding a full depth third dimension. It is in fact interesting to restrict the system to nearly two dimensions and a gradual increase of depth thickness introduces slip, shearing and other physical effects that it is interesting to control.

Sand and granular systems offer an experimental model that can be observed in the laboratory, but there are industrial applications such as oil recovery from tar sands [9] where an understanding of the flow processes involved is important for achieving economically optimal extraction.

Other related applications where an understanding of sand flow is necessary include: dune modelling [2] and erosion studies [21]; use of sand in molding [18] for casting; sand conductivity problems [20]; sand slurry modelling [14]; and granular cohesion [11], fracturing [12] and layer shearing scenarios [1]. Sand avalanching has also provided a useful model platform for studying critical flow [5].

Models with downhill or gravitational granular flow effects [7] can be constructed in a number of ways. A molecular dynamical approach with many individual had disk or hard sphere particles that displace one another is one possibility. This approach is quite computationally intensive and to obtain a realistic number of individual sand particles, would require (x, y) directional floating-point positions and momenta values for each grain. A simpler approach is based on the notion of a lattice gas. In a lattice model, each cell is occupied by an integer value representing sand species one, sand species two, of fluid. A simple local update mechanism can be constructed to make the particles move appropriately.

A variation of the Kawasaki spin-exchange model [19] is a useful starting point. In this system each cell in the lattice can exchange positions [17, 28] with one of its neighbouring cells at each time-step. Probabilities that determine how likely particular sand species or fluid cells will exchange with one another can be incorporated in the form of a Boltzmann energy weighting factor and gravity can be applied by imposing a directional bias to these values. The total number of each species is conserved as only pair-wise swaps are executed.

The sand picture system is essentially a complex fluid - with a colloidal suspension of sand particles contained in water or oil. Modelling complex fluids is a challenging problem [3,22], particularly in situations with realistic geometric boundaries and barriers. One useful approach is the Invasion Percolation (IP) model [15] which has developed over many years [10, 29] and is a useful tool for experimenting with flow of immiscible fluids in model reservoir systems.

The IP model has a number of variations in its formulation and has been extensively researched in two and three dimensional configurations. It has been successfully used to model diverse applications ranging from drainage systems [23, 25, 27] to vascular network formation in tumours [4] as well as reservoir extraction and deposition processes where recent applications still find it useful [8].

We are interested in the somewhat simpler system of the sand picture in two dimensions. We believe it shares some dynamical growth properties in common with other systems and models that are also driven by an external force such as gravity. Although some research has been reported on the influence of gravity [6, 13, 24], it has not been thoroughly explored as a way of introducing a buoyancy parameter into the IP model.

Various heuristics and adjustments can be made to the model to make the sand falling behaviour look more realistic and in particular so that it forms layers [16] like a real sand picture. Typically a simple model would look “wrong” as there is no viscous drag component to make the fluid appear real, but this can be incorporated by adding a preference parameter for sand to adhere to other sand cells of the same or different types. This model can be perturbed with a preferred directional gravitational force that leads to nearly correct physical phase separation of the coloured sands. The model provides a visually realistic simulation that can be rendered in real time. We are implementing this using Android and Java on tablet computers with inbuilt gyroscopic sensors that allow the simulated system to adapt to real gravity in interactive time. We describe the model and the implementation and software architecture for this App and the associated performance tradeoffs. We discuss future performance improvements using graphical processing units and other tablet specific features.

Our article is structured as follows: In Section 2 we describe the rules for our sand picture simulation. We present selected results in Section 3 including some visual snapshots of the simulated system in Section 3.1 and also some computational performance analysis in Section 3.2. We discuss the implications of this sort of model in Section 4 and offer some conclusions and areas for further work in Section 5.

2 The Sand Picture Model

We have developed various simulation software for models like the Ising, Kawasaki, IP and sand picture system. In this work we are aiming to produce a graphical simulation model that runs effectively in real time so we can watch the complex fluidic behaviour and associated spatial patterns as they form.

The work shown here relates to the initial prototype Desktop PC implementation. Tablet computers with touch screen displays and built-in gyroscopic sensors allow both detailed user interaction and an automated sense of gravitational direction. We are experimenting with these devices [26] and are presently implementing an App version of the Sand picture simulation models which uses the Java and graphical support system available under the Android Operating system for mobile devices.

We base our sand model on a lattice of cells which contains a single variable that determines whether the cell is the suspending "water" or "oil" or contains a sand particle. We can further refine the model with multiple sand species or colours, which can be heavier or lighter.

We need to impose a dynamic scheme to determine how sand and the suspending liquid move around. A useful starting point for this is the Ising model on a lattice. In the Ising model, a heat-bath algorithm is used to emulate thermal effects on atoms in a magnetic material arranged in a crystalline lattice. The Ising system consists of a micro crystalline array of single bit magnetic moments or "spins" which interacts with its nearest neighbours. At each time step of the simulation each spin is considered in turn and the energy and thermal probability of it "flipping" – reversing its direction are considered. The probability of flipping is different, depending upon the applied temperature.

Ising spins align with their neighbours when the system is cold, but thermally randomize when it is hot. The interesting feature about the Ising system in 2 (or 3) dimensions is that there is a definite Curie temperature that can be measured. In real magnets the Curie temperature is the temperature above which the material stops being a magnet, or an alternative viewpoint is that materials like iron spontaneously become magnetic below their Curie temperature. This is known as a phase transition and is very difficult to explain simply without a model to demonstrate.

In the sand system however we need to fix the number of sand and suspending liquid cells. A useful related model is therefore the Kawasaki exchange model which is constructed in a similar manner to the Ising system. In this case however we preserve a fixed ratio of the two microscopic species since instead of flipping or changing species, in the Kawasaki system we only allow them to swap positions with one of their (randomly chosen) neighbours. In this respect the Kawasaki system models diffusion and phase separation or "unmixing" of the two species. The rate and manner of unmixing is like the separation of two atomic species in a binary alloy. This sort of dynamical behaviour is of great importance in real materials. Without some separated granules an alloy typically lacks strength and other physical properties but if too much separation occurs it can break apart and cause catastrophic failure in for example fuel rods in a reactor. In the sand system, an exchange or cell-switching dynamical scheme similar to the Kawasaki model can be used.

We have already experimented with a variation of the Kawasaki model in 2-D and 3-D with a gravitational bias imposed [16]. It successfully exhibits complex layers with multiple phases. In this present work, we experiment with tuning the microscopic rules to try to obtain a more realistic set of behaviours for the sand picture.

Figure 2 shows a 2-dimensional mesh surrounding individual cells A (red), B (blue) and C (green). The colours show the surrounding neighbourhood "halos" for these cells. We can use nearest-neighbour or Moore neighbourhoods with 4 (1,2,3,4) or 8 (1,2,3,4,5,6,7,8) neighbouring cells respectively. For the exchange or switching dynamics, we consider the energy consequences of particle A switching with particle B or with particle C. In the case of A and B they are both at the same height and therefore gravity plays no part - unless the system is at a angular tilt as shown in the screenshots in column 3. The case of A and C will have gravitational consequences and there should be a bias so that the heavier of A and C prefers to move downwards.

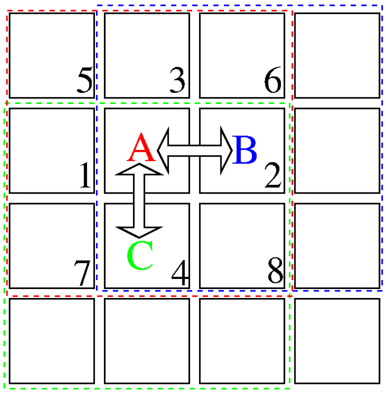

Figure 2: Mesh exchange dynamical schematic.

To make the model behave realistically the Boltzmann approach allows a heavier particle to move upwards - but only with a very low probability. This stochastic dynamics emulates diffusion amongst the particles and also compensates for the rigidity of the mesh.

The simulation algorithm therefore consists of the following stages:

- Initialize a mesh of for example 1024 by 768 cells, that can be mapped to individual pixels on a display.
- Populate each cell randomly with either water or heavy sand or light sand.
- Iterate time steps of the model where each step consists of:
  - Consider each cell in random order
  - Look at the cell "below" the chosen cell
  - Follow the microscopic rules given in column 4 to determine whether to exchange or switch the contents of the two cells
  - Repeat

This model is essentially a variant of the Kawasaki Spin exchange model, where we have taken the Boltzmann energy probabilistic rate equation usually used in a Metropolis update scheme and made it more sophisticated by adding additional terms and effects to take account of gravity and a pseudo-viscosity within the complete fluid suspension.

There are a number of parameters that need to be investigated experimentally to tune the macroscopic behaviour of the system. The relative strength of gravity compared to the normal random diffusion of particles is one. Another is the particular neighbour set or threshold to consider to introduce a realistic looking pseudo-viscosity.

We arrived at a rule set rule set for the evolution from one state to another by experimentation. Through testing of various rule-sets, the following rules gave the most realistic implementation

The simulations acts on the particle that that is defined be adding to the x and y coordinates of the current position. The values that are added to the x and y are defined in a look-up table. This is modified by the orientation of the system, on the Desktop PC version the state of both the gravity and the user defined rotation modify the look-up table. This means the rules can be defined arbitrarily, it also means that the rules do not have to be heavily modified for the port to mobile devices which use a combination of Android/Java.

The rules make the use of random number generation for tie-breaks and for some of the stochastic conditions, and are as follows:

If the sand is white (lightest sand)

- if there are seven or more white sand particles around the white sand particle then do nothing
- if the same white sand particle or the heavier black sand is "below" then half the time look and see if there is water particles (lightest) to the bottom left and right otherwise do nothing
- if both are water to the left and right then coin toss to see which one we swap with. Otherwise fall into the one that as the water.
- if no water present then do nothing

If the sand is black (heaviest sand)

- if there are five or more black sand particles around the particle then do nothing
- if the same black sand is below then try to fall to the left or right of the if there is a light sand or water is on either side
- if the particle below is water then switch the particles
- if the particle below is white sand
  - 1/3 of the time swap the heaver sand for the lighter sand
  - 1/3 of the time see if there is water below left and right if there is then push the white sand left or right and let the black sand drop down
  - the final 1/3 of the time if there is a lighter particle below to the left or right then fall into its place (swap)

These rules conserve the amount of each particle to that which was first initialized in the system. This means we do not lose any particles from the simulation. In a real life sand picture the mounting case that holds the sands and water is sealed so it is also impossible to lose or change the proportion of particles present. Real life systems often leak air bubbles into the system and an area for future work is to explore how air bubbles might be suitably simulated.

Other areas for further investigation concern the dimensionality of the system. A real sand picture is almost two-dimensional, with a very thin region of colloidal fluid between the glass plates. It is an open question as to how this thickness in the third dimension will affect the complex fluid flow. A simulated system may be able to probe this issue.

3 Results

We illustrate the simulated system with a series of screen shots, with associated commentary, before discussion performance achievements in terms of frames per second on various platforms.

3.1 Simulated Snapshots

This series of screen-shots shows the simulated sand system rendered with the Desktop PC version and showing the orientation of the system with respect to the gravitational direction, on the right hand side. In each picture, the system consists of a suspending liquid (water or oil) represented by a blue particle species; a light yellow coloured sand species and a heavier black coloured sand species.

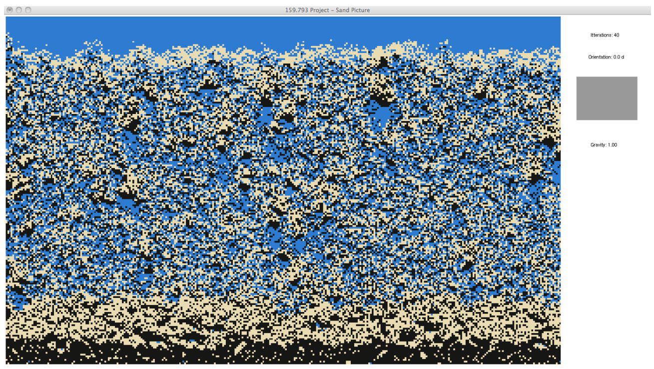

Figure 3: Sand starting to layer

Figure 3 shows the Desktop PC version of the sand simulation after 40 time steps. Following a random initialization, after only 40 time steps the black heavy particles are already starting to fall the fastest and are starting to accumulate at the bottom.

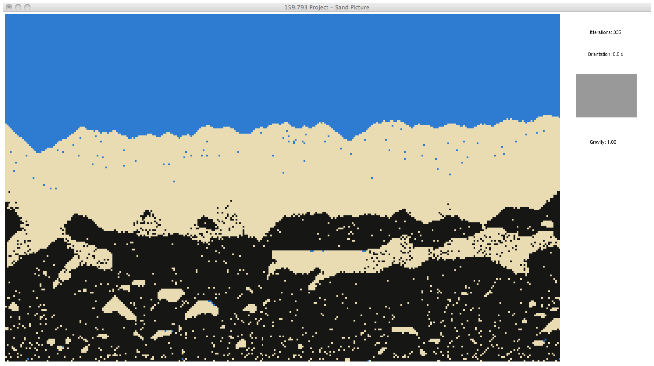

Figure 4: Layered sand simulation

Figure 4 shows a simulation at its later stages when the particles have fallen into layers with pockets of lighter material spread throughout. This shows a similar layer behaviour to the actual sand picture.

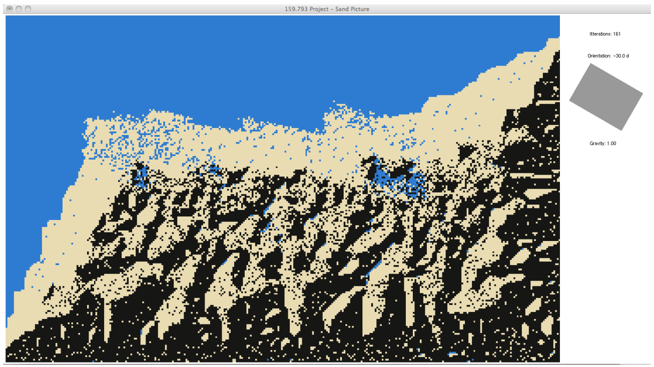

Figure 5: Layered sand with rotation changed

Figure 5 demonstrates the ability to rotate the sand picture and the layering effects it has on the sand simulation.

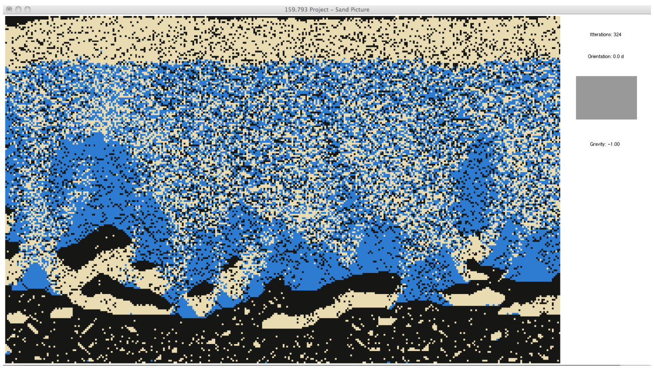

Figure 6: Relayering after change in gravity

Figure 6 shows the effect and re-layering after a gravity change in the sand simulation. As can be seen the lightest water particles are falling to the bottom and the heaviest black sand are moving to settle at the top of the screen.

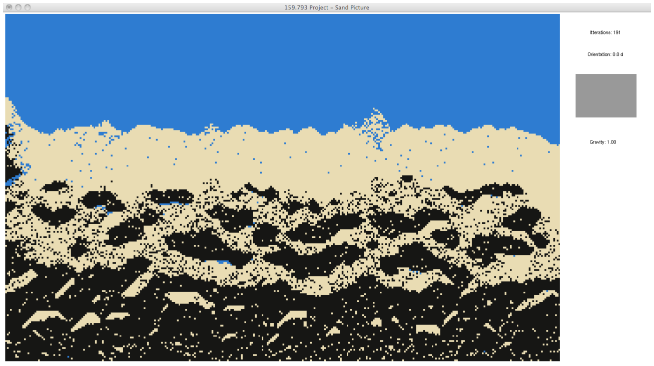

Figure 7: A near steady state of the simulation

Figure 7 shows a nearly steady-state of the sand simulation. This shows the effect that would happen in real life it the sand picture has been left for a few minutes for the particles to rearrange themselves.

3.2 Performance

It was determined empirically that a frame rate of at least four frames per second for the simulation to seem realistically similar to the flow seen on a real glass framed sand picture.

Two principle platforms were tested: a desktop PC running Ubuntu Linux 11.10 and with a Quad core 3.4GHz Intel processor with 8Gigabytes of memory and a high performance NVidia GTX 680 graphics card with its own graphics memory; and secondly a Google Nexus 7 tablet computer running Android 4.2.1 (Jellybean) and containing a quad core ARM processor running at 1.2GHz and with 1Gigabyte of available memory.

The desktop configuration obviously offers more choice of resolution and it was found that two interesting resolutions were firstly a 640×400 pixel resolution which yielded around 20 frames-per-second and used around 51 milliseconds to render each frame and approximately 45 milliseconds to execute the model computations. Secondly a full-screen resolution of 1280×800 pixel resolution yielded an adequate 5 frames-per-second and around 215 milliseconds to render with around 200 milliseconds used for computations.

The Nexus Android tablet implementation is obviously more interactive and "feels" more like a real sand picture. We obtained a frame rate of 5 frames per second when the resolution was lowered to 100×173 pixels. This gave around 100 milliseconds for rendering a frame but around 235 milliseconds for model computations.

The Nexus tablet is an attractive platform as it has accessible gravitational/gyroscopic sensors so it was possible to adapt the direction of simulated gravity to the orientation in which the tablet was held. This obviously carries an overhead however and the resolution had to be reduced to maintain a worthwhile frame rate.

4 Discussion

As with any visual simulation realism is a big factor. The sand simulation must look like and behave as the actual sand picture. I believe this has been achieved in both PC and Android implementations through the designed rule-set. The simulation does look visually like a real life sand picture with the sand layering appropriately due to the implemented rules and the weights of the different sand and water particles. The affect of pockets of white sand in black sand visually enhances the realism of the simulation. This shows clumping of life-like sand particles stuck in a state where they cannot move themselves free which could be due to being too tightly packed. This can be observed in a real sand picture and is a common phenomena. Complex rule-sets such as these can however be computationally expensive and this factor needed to be taken into account when considering interactivity and real time running. The main parameter of the system that affects realism is the resolution of the simulation. As sand particles are rather fine from about 0.0625 mm to 2 mm it is essential that the resolution is able to perform within or close to this range.

In both the PC and Android implementations the resolution is not extremely small and may not fall within this range, however it is fine enough to be a convincing simulation of sand. As the PC implementation has nearly twice the resolution as the Android implementation a visual difference can be seen, however, due to hardware performance limitations and the necessity for inter- activeness and real time running, a lower resolution was opted for as a tradeoff to achieve these other key factors. The resolution of the simulation is largely based on the speed and performance of the system it is implemented on. In this case the PC platform was far superior in performance and could perform.

As the real life sand picture can be rotated and flipped and is interactive, it was essential that the implemented simulations also had these features. The only way to measure this is to look at the simulations visually and see if can in fact be flipped and rotated. This unique interactivity gives both the sand picture and the implemented simulations a unique tactile feel. The implemented solutions can be manipulated this gives the affect of controlling the gravity in the simulations and determines how and in which direction the simulated sand particles fall in. Sensors were a key part in implementing this interactivity into the simulations. From simulated sensors in the PC version to the real gravity sensors in the Android version they both provided a fun and interactive interface that let the simulation be explored and played with. The raft of sensors available on the Android platform and through the range of devices made implementation of interactivity simple and easy to manage.

Finally for the implementations on both PC and Android the simulations needed to run in real time. This was needed for both the above realism and interactivity. Seeing the simulation happen in front of you was a key criteria and was crucial for interactivity. Throughout testing it became apparent at an early stage that at least 4 frames-per-second (FPS) needed to be rendered in order for interactivity and realism to really be realized. Therefore this became somewhat of a baseline for mainly the Android implementation as this was the lowest performing system. In the Android implementation an average of about 5 to 6 FPS is realized which could be better to enhance response time however realism required a certain resolution and therefore a compromise had to be met.

The PC implementation on a fast modern processor can perform in the range of 15 to 20 FPS and is visually faster. As can be seen from the above results and discussion all three factors of realism, interactivity and real time running all tie into each other. The common battle to balance the right mix of all three of these crucial aspects proved difficult throughout both development and testing of the simulation. The resulting systems on both PC and Android give realistic simulations of an actual sand picture.

5 Conclusion

The real life sand picture is an interesting desktop amusement that is interesting due to its interactive and visual aspects. The sand picture provided an excellent opportunity to simulate in a digital environment and take advantage of some of the latest technology available.

Simulation of the sand picture was not an easy task and clear objectives were set out to help achieve an appropriate solution. The objectives were realism, interactivity and real time running. Through the utilisation of various existing models and paradigms such as cellular automata, Boltzmann energy constants and Metropolis updates, a realistic looking sand picture was generated on both PC and mobile device. Interactivity was achieved on both PC and Android platforms through the use of both simulated and real gravitational sensors. This gave the sand picture the fun, interactive feel that causes the real life sand picture to be such a popular desktop assortment. Real time running was also achieved on both platforms however sacrifices had to be made on resolution which effected realism to achieve real time running. Performance was very much a factor on the Android platform and required a few tweaks to get the simulation to run as fast as possible.

In summary, a realistic algorithm was developed that produces realistic simulations of near physically sized sand grain on a desktop computer. It is likely that next generation tablet computers will have sufficiently powerful processors to enable similar realism. We There is also scope for a parallel partitioning of the problem amongst the multiple cores of the processor. We believe this will be worthwhile for 8-cored processing systems and higher.

References

[1] Alarcon, H., Geminard, J.C., Melo, F.: The effect of cohesion and shear modulus on the stability of a stretched granular layer. Phys. Rev. E 86, 061303–1–7 (2012)

[2] Araujo, A.D., Andrade, J.S., Maia, L.P., Herrmann, H.J.: Numerical simulation of particle flow in a sand trap. arXiv 0808.2649v1, Universidade Federal do Ceara, Ceara, Brazil (August 2008)

[3] Arratia, P.E.: Complex fluids at work. Physics 4(9), 1–3 (January 2011)

[4] Baish, J.W., Gazit, Y., Berk, D.A., Nozue, M., Baxter, L.T., Jain, R.K.: Role of tumor vascular architecture in nutrient and drug delivery: An invasion percolation-based network model. Microvascular Research 51, 327–346 (1996)

[5] Bak, P., Tang, C., Wiesenfeld, K.: Self-organized criticality: An explanation of 1/f noise. Phys. Rev. Lett. 59, 381–384 (1987)

[6] Birovljev, A., Furuberg, L., Feder, J., Jossang, T., Maloy, K.J., Aharony, A.: Gravity invasion percolation in two dimensions: Experiment and simulation. Phys. Rev. Lett. 67, 584–587 (1991)

[7] Borzsonyi, T., Ecke, R.E., McElwaine, J.N.: Patterns in flowing sand: Understanding the physics of granular flow. Phys. Rev. Lett. 103, 178302–1–4 (2009)

[8] Damron, M., Sapozhnikov, A.: Outlets of 2d invasion percolation and multiple-armed incipient infinite clusters. Probability Theory and Related Fields 150(1-2), 257–294 (2011)

[9] Doan, D.H., Delage, P., Nauroy, J.F., Tang, A.M., Youssef, S.: Microstructural characterization of a canadian oil sand. Can. Geotech J. 49, 1–9 (2012)

[10] Ebrahimi, F.: Invasion percolation: A computational algorithm for complex phenomena. Computing in Science and Engineering Mar/Apr, 84–93 (2010)

[11] Franklin, S.V.: Geometric cohesion in granular materials. Physics Today September, 70–71 (2012)

[12] Geminard, J.C., Champougny, L., Lidon, P., Melo, F.: Flexural fracturing of a cohesive granular layer. Phys. Rev. E 85, 012301–1–3 (2012)

[13] Glass, R.J., Conrad, S.H., Yarrington, L.: Gravity-destabilized nonwetting phase invasion in macroheterogeneousp orous media: Near-pore-scale macro modified invasion percolation simulation of experiments. Water Resources Research 37(5), 1197–1207 (May 2001)

[14] de Groot, M.B., Lindenberg, J., Mastbergen, D.R., den Ham, G.A.V.: Large scale sand liquefaction flow slide tests revisited. In: Proc. Eurofuge. pp. 1–22. Delft, The Netherlands (23-24 April 2012)

[15] Hawick, K.A.: Gravitational and barrier effects in d-dimensional invasion percolation reservoir models. In: Proc. Int. Conf. on Power and Energy Systems and Applications (PESA 2011). pp. 259–266. No. CSTN-134, IASTED, Pittsburgh, USA (7-9 November 2011)

[16] Hawick, K.: Visualising multi-phase lattice gas fluid layering simulations. In: Proc. International Conference on Modeling, Simulation and Visualization Methods (MSV'11). pp. 3–9. CSREA, Las Vegas, USA (18-21 July 2011)

[17] Hawick, K.A.: Domain Growth in Alloys. Ph.D. thesis, Edinburgh University (1991)

[18] Jakumeit, J., Jana, S., Waclawczyk, T., Mehdizadeh, A., Sadiki, A., Jouani, J.: Four-phase fully-coupled mold-
lling and solidi
cation simulation for gas porosity prediction in aluminum sand casting. IOP Conf. Series: Materials Science and Engineering 33, 012074 (2012)

[19] Kawasaki, K.: Diffusion constants near the critical point for time dependent Ising model I. Phys. Rev. 145(1), 224–230 (1966)

[20] Koch, K., Kemna, A., Irving, J., Holliger, K.: Impact of changes in grain size and pore space on the hydraulic conductivity and spectral induced polarization response of sand. Hydrology and Earth System Sciences 15, 1785–1794 (2011)

[21] Lammel, M., Rings, D., Kroy, K.: A two-species continuum model for aeolian sand transport. New Journal of Physics 14, 0930307–1–25 (2012)

[22] Martys, N., Cieplak, M., Robbins, M.O.: Critical phenomena in fluid invasion of porous media. Phys. Rev. Lett. 66(8), 1058–1061 (February 1991)

[23] Masek, J.G., Turcotte, D.L.: A diffusion-limited aggregation model for the evolution of drainage networks. Earth and Planetary Science Letters 119, 379–386 (1993)

[24] Meakin, P., Feder, J., Frette, V., Jossang, T.: Invasion percolation in a destabilizing gradient. Phys. Rev. A 46(6), 3357–3368 (September 1992)

[25] Prat, M.: Isothermal drying of non-hygroscopic capillary-porous materials as an invasion percolation process. Int. J. Multiphase Flow 21, 875–892 (1995)

[26] Preez, V.D., Pearce, B., Hawick, K.A., McMullen, T.H.: Software engineering a family of complex systems simulation model apps on android tablets. In: Proc. Int. Conf. on Software Engineering Research and Practice (SERP'12). pp. 215–221. SERP12-authors.pdf, CSREA, Las Vegas, USA (16-19 July 2012)

[27] Stark, C.P.: An invasion percolation model of drainage network evolution. Nature 352, 423–425 (1991)

[28] Wang, J.S., Binder, K., Lebowitz, J.L.: Computer simulation of driven diffusive systems with exchanges. J.Stat.Phys 56(5), 783–819 (1989)

[29] Wilkinson, D., Willemsen, J.F.: Invasion Percolation: a new form of percolation theory. J.Phys.A. 16, 3365–3376 (1983)