KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Computer Science and Engineering

Second cycle, 30 credits

Realism of Procedural Lichens on Horizontal Stone Surfaces

Simulation and evaluation of lichens

ANDERS STEEN

Stockholm, Sweden, 2022

Realism of Procedural Lichens on Horizontal Stone Surfaces

Simulation and evaluation of lichens

ANDERS STEEN

Degree Programme in Computer Science and Engineering
Date: May 6, 2022

Supervisor: Christopher Peters
Examiner: Tino Weinkauf
School of Electrical Engineering and Computer Science
Host company: Resolution Games
Swedish title: Realism i procedurella lavar på horisontella stenytor
Swedish subtitle: Simulering och evaluering av lavar

Sammanfattning

Människor använder kreativitet och manuell färdighet för att skapa saker, vanligtvis i en arbetsintensiv process. Procedurell innehållsgenerering (eng. PCG) är ett tillvägagångssätt där datorer gör majoriteten av det manuella arbetet i skapandet av virtuella objekt, och en människa leder den kreativa processen. Ett problem med PCG är att de skapade objekten tenderar att ha ett för nytt och artificiellt utseende, vilket har lett till forskning kring att simulera åldrandeprocessen för virtuella objekt. Forskning har bland annat fokuserat på att simulera tillväxten av levande organismer såsom lavar på existerande virtuella objekt. Flera tidigare tekniker saknar en tydlig evaluering av resultatens realism. I denna studie implementeras en lavtillväxtsimulering som evalueras med avseende på upplevd realism. Simuleringens mål är specifikt att simulera lavar på horisontella stenytor representerade av en uppsättning texturer. Simuleringen är långsam men robust och flexibel, och kapabel till att generera ett brett omfång av resultat. 75 deltagare rekryterades bland universitetsstudenter och bland arbetare i datorspelsindustrin, och ombads jämföra realismen hos lavar genererade med olika parameterinställningar. De undersökta parametrarna är skalan på lavarna, kompaktheten på lavarna och antalet lavar (densitet). Deltagarna upplevde att ytor med hög och mellanhög densitet av småskaliga semikompakta lavar var mest realistiska, medan ytor med låg densitet av storskaliga kompakta eller mycket utspridda lavar upplevdes som minst realistiska. De lavar som upplevdes vara mest realistiska är också bland de beräkningsmässigt dyraste att generera. Detta resultat visar att det finns en avvägning mellan realism och simuleringstid i lavtillväxtsimuleringen. Simuleringen kan generera ett brett omfång av lavutseenden, men simuleringens snabbhet kan förbättras på flera sätt.

Realism of Procedural Lichens on Horizontal Stone Surfaces

Anders Steen
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology
astee@kth.se

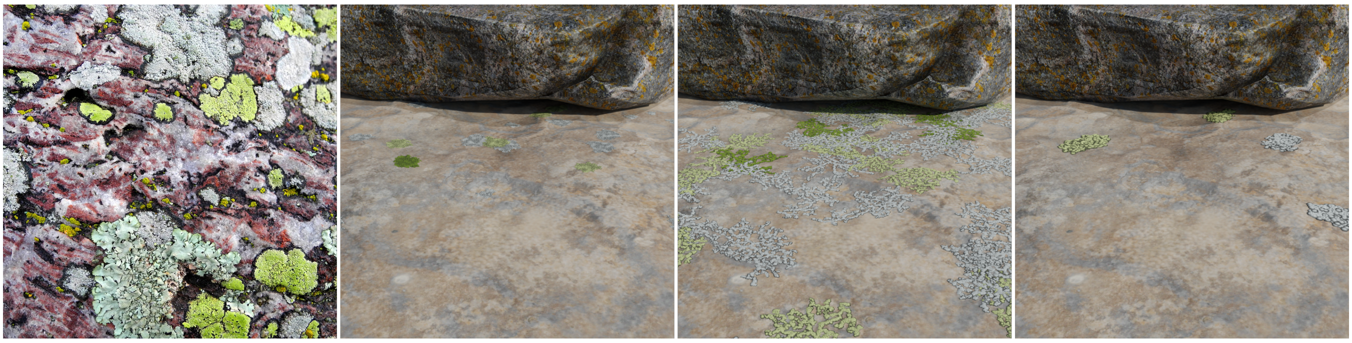

Figure 1: Based on the photograph[1] in the leftmost image, lichens were generated with varying scale, compactness and number of lichens (density), a sample of which are shown in the three rightmost images. The realism of the images was rated in a user study, and from left to right the images are ordered from the most realistic to the least realistic.

[1]https://www.maxpixel.net/Nature-Moss-Rocks-Rock-Texture-Stone-Lichen-1461368

ABSTRACT

Humans use creativity and manual dexterity to create things, usually in a labor-intensive process. Procedural content generation (PCG) is an approach where computers do most of the manual work of creating virtual objects, while a human leads the creative process. One problem with PCG is that created objects tend to look too perfect and artificial, leading to research into simulating the aging process of created virtual objects. Simulating the growth of living organisms such as lichens on existing virtual objects has been a focus of research. Most previous techniques lack a clear evaluation of the realism of the results. In this thesis, a lichen growth simulation is implemented and evaluated in terms of perceived realism. Specifically, the target of the simulation is lichen growth on horizontal stone surfaces represented using a set of textures. The lichen growth simulation is slow but robust and flexible, capable of generating a wide range of output textures. 75 participants were recruited among university students and video game industry professionals, asked to compare the realism of lichens generated with different parameter settings. The investigated parameters are the scale of the lichens, the compactness of the lichens, and the number of lichen clusters (density). Participants perceived surfaces with a dense or medium-dense cover of small scale, semi-compact lichen clusters as the most realistic, while surfaces with a sparse cover of large scale, very compact or very lacy lichen clusters were perceived as the least realistic. The lichens perceived as the most realistic are also among the most computationally expensive to produce. This outcome tells us that there is a trade-off between realism and simulation time in the lichen growth simulation. The simulation is able to generate a wide range of lichen appearances, though the speed of the simulation can be greatly improved in multiple ways.

KEYWORDS

procedural generation, lichen, perceived realism, simulation

1 INTRODUCTION

Modern computer generated imagery is often of very high quality, sometimes to the point of being indistinguishable from photographs. Considerable human creativity and labor is involved in creating such imagery, which limits the speed at which the imagery can be created. Humans need to craft 3D models, textures, animations and much more to create the computer-generated images.

Procedural content generation (PCG) aims to automate much of the manual labor by having computers perform the labor-intensive process, leaving humans to perform the creative process. PCG can be used to create essentially anything involved in computer generated imagery, which would otherwise be created by humans.

One problem that automated generation using PCG carries with it is that created artifacts tend to be too perfect. Reality is seldom completely smooth, symmetrical or clean. While it takes work to make real objects perfect in these regards, virtual objects usually start off perfect and it takes work to make them less perfect and more like reality. This work can be done by humans, and therefore it can be made a target of PCG. Multiple highly varied approaches exist that address this problem [Bandeira and Walter 2009; Chen et al. 2005; Dorsey et al. 1996; Rosenberger et al. 2009; Xue et al. 2011].

Biological growth on a surface can indicate its age, and simulate reality better. One such growing phenomenon is lichens, a symbiotic (or perhaps parasitic) relationship between fungi and plant organisms. Lichens can grow in places where other organisms struggle to survive, so lichens growing on bare rock or stone is very common [Ahmadjian 1973]. Previous research has studied simulation methods for biological growth in general and lichen growth in particular [Bachman 2019; Desbenoit et al. 2004].

Stone is a material that has been used by humans to create and build objects for a very long time. The commonality of the material, as well as the fact that lichens often grow on stone surfaces, makes it a good material to study the simulated growth of lichens on.

1.1 Research question

With all this in mind, this study intends to implement a lichen growth simulation on flat horizontal stone surfaces and to evaluate the perceived realism of the simulation. The overarching research question is: *How can lichen growth on horizontal stone surfaces be generated procedurally while maximizing perceived realism and performance, and guaranteeing expressivity?* With specific subquestions:

- RQ1: What limits are there to the expressivity of the procedural model?
- RQ2: How do the simulation parameters affect the perceived realism of the lichen growth?
- RQ3: How is the performance of the lichen growth simulation affected by the simulation parameters?

Expressivity refers to the width of the range of possible outputs a procedural model is able to generate. Perceived realism is a perceptual measure of realism that relies on the human perceptual system, rather than on objectively measured similarity with physical reality. Answering questions about the expressivity, perceived realism and performance of a procedural model should give reasonable insight into the advantages and drawbacks of using the model. Perceived realism is especially relevant to models like the one in this study, which attempts to replicate reality, and realism evaluation is seldom seen in procedural generation.

1.2 Report overview

This report is structured in the following manner. Section 2 covers previous research in lichen growth, perceived realism, and PCG model expressivity. Section 3 contains a detailed description of the lichen growth simulation algorithm, and section 4 explains how the algorithm is evaluated and also presents the results of the evaluation. The results are discussed and put in context in section 5, where future work is also proposed. The conclusion of the study is presented in section 6.

2 RELATED WORK

In this section, previous research related to the topic of this study is presented. The main areas are organic growth simulations, evaluation of perceived realism and methods of measuring expressivity of procedural models.

2.1 Organic growth

The simulation of organic or biological growth is an old problem with many approaches. Prusinkiewicz [Prusinkiewicz 2004] provides a survey of the field of modeling the growth of plants computationally. Many techniques stem from the concept of L-systems [Lindenmayer 1968], which is a well-known technique in PCG able to create branching recursive structures based on formal grammars. Newer techniques take various biological factors into account, limiting or accelerating growth depending on the environment and the structure of the plants [Prusinkiewicz 2004].

Plants are not the only type of organic growth, there are many other growth phenomena that can be simulated, even when limiting the growth to exist solely on the surface of an object. [Bachman 2019] outlines a set of methods for simulating organic growth on surfaces. These methods are able to simulate the growth of flowers, pine cones, leaves, veins, and lichens.

A technique called Diffusion Limited Aggregation (DLA) introduced in [Witten Jr and Sander 1981] and mentioned in [Bachman 2019], is highly suitable for simulating lichen growth. It employs particles moving by Brownian motion and colliding with other particles to form branching and spreading clusters of particles that resemble lichens. DLA is expanded to Open DLA (ODLA) by Desbenoit, Galin and Akkouche [Desbenoit et al. 2004] and used for a sophisticated lichen simulation that takes the surrounding environment into account. To create lichen geometry, the simulation uses cellular textures, as defined by [Fleischer et al. 1995]. Cellular textures is a technique that transforms a set of points, such as particles from an ODLA simulation, into an organic-looking texture on a surface.

A procedural approach to generating textures resembling lichens is proposed by [Cutler et al. 2002]. The approach is more generally meant to create solid 3D models, something rather unrelated to lichen growth, but an interesting side result is the generation of lichen-like surface textures. The generated textures are mostly localized discolorations and their realism does not hold up to close scrutiny, particularly when compared to the lichens of [Desbenoit et al. 2004].

The ODLA technique from [Desbenoit et al. 2004] is the most important one for this study, forming the basis of the lichen growth simulation. The texture generation proposed in 3.2.2 is inspired by the cellular textures from [Fleischer et al. 1995].

2.2 Perceived realism

Perceived realism in computer-generated images has been an object of research for a long time. An early study was conducted by [Meyer et al. 1986], where participants were asked to discriminate between a real scene and a computer-generated image of the same scene. [Rademacher et al. 2001] performed a similar study, comparing photographs with computer-generated images. These studies are examples of successful studies into perceived realism.

Subsequent studies of perceived realism are often similar to the early studies, employing a 2-alternative forced choice (2AFC) study design, where participants are shown stimuli and forced to choose one of two responses. [Chen et al. 2020; Giunchi et al. 2021; O'Connor et al. 2015; Ramanarayanan et al. 2007; Xue et al. 2012] all use a common study design, where participants answer 2AFC tasks asking them to discriminate the realism between two images. The images are similar, but differ in a single parameter, meaning the responses can be analyzed to conclude which parameter settings result in the greatest perceived realism. The user study design in the present study incorporates the insights into 2AFC design and subsequent data analysis from this related work.

There have also been efforts to create computational models for measuring perceived realism. In a model proposed by [Fan et al. 2017], data gathered from human participants about image visual realism is used to create a computational model for predicting the perceived realism of an image. [Rajasekaran et al. 2019] uses a similar method for creating the model, but is specialized on the perceived realism of images depicting computer-generated landscapes.

2.3 Expressivity

PCG models generate objects or phenomena, and arguably the most important aspect of a model is how expressive it is. Expressivity is a qualitative measure indicating the range of different objects or phenomena that can be created using the model. Fully evaluating a model's expressivity is difficult, since the output set of a model is often infinitely large. [Smith and Whitehead 2010] propose a measurement called expressive range, which is able to visualize the expressivity of a model given a very large number of sample outputs and a set of metrics defined on the output set. Using the generated visualizations, different PCG models with similar output sets can be compared more easily. The present study takes inspiration from the concept of expressive range in the design of the expressivity evaluation.

3 IMPLEMENTATION

In this section, the implementation of the procedural model is described in detail. The main parts of the implementation is the handling of input textures and the lichen growth simulation, which in turn is divided into growth simulation and texture generation.

3.1 Textures

Representing stone as a virtual object can be approached from many directions. The stone will consist of a 3D surface mesh, covered by textures, images containing information about the surface. In physically based rendering (PBR), which is a common approach to rendering, a combination of textures is used to instruct the computer how light interacts with the surface. Surfaces have properties like color, roughness and small-scale bump detail, represented by a color map texture, a roughness map texture and a heightmap texture respectively.

The color map texture is a regular RGB texture describing the color of the surface. The heightmap texture is a texture with values from black to white representing heights from lowest to highest. The roughness map texture is similar to the heightmap, but instead describes how rough the surface is, deciding the mix between specular and diffuse light scattering on the surface.

In this study, the 3D mesh is not relevant as the stone is always a horizontal stone plane. Therefore, the focus is on the three aforementioned textures. The algorithm needs these three textures as input. The textures should be of square shape, and of arbitrary but mutually equal size. The output of the algorithm described in this section will be the corresponding lichen-covered versions of the three textures.

3.2 Lichens

The lichen simulation is based on the implementation in [Desbenoit et al. 2004]. It consists of two distinct parts, first the growth simulation, then the texture generation from the simulation data. Only the growth simulation is heavily based on [Desbenoit et al. 2004], the texture generation is novel, proposed by this study.

3.2.1 Growth simulation. The method in [Desbenoit et al. 2004] operates on a triangle mesh, while the proposed method operates in an abstract lichen space. The lichen space is an infinite plane, which can be mapped to and from the input texture space. The input texture space is simply a plane constrained in size by the dimensions of the textures. Mapping from texture space to lichen space is a simple scaling operation, because the origins and axes of the spaces coincide. Mapping from lichen space to texture space is a scaling operation followed by a translation to map points that would fall outside the texture space bounds into their equivalent positions as if the texture space tiled to infinity. The scaling factor between the spaces is set by the user.

To begin the simulation, a user-defined number $p_{\text{startClusters}}$ of lichen seed particles are distributed uniformly at random in the input texture space. Each seed starts a new lichen cluster, which is a set of lichen particles with a given initial particle and a given species. The species specifies parameters governing the growth, as well as the color of the lichen, and is selected from a weighted set of species defined by the user. Species parameters are set by the user, and are denoted s_x, and general parameters set by the user are denoted p_x, where x is the name of the parameter. New clusters are then created in the same way in subsequent iterations with a given probability $p_{\text{newSeedProbability}}$ until the number of clusters reaches $p_{\text{maxTotalClusters}}$. In each following iteration of the simulation, for a subset of clusters determined by $p_{\text{clustersPerIteration}}$, a source particle is selected at random from the cluster. Then, a new particle is created in a random direction at an offset $2 \cdot s_{\text{particleRadius}} + s_{\epsilon}$ from the source particle.

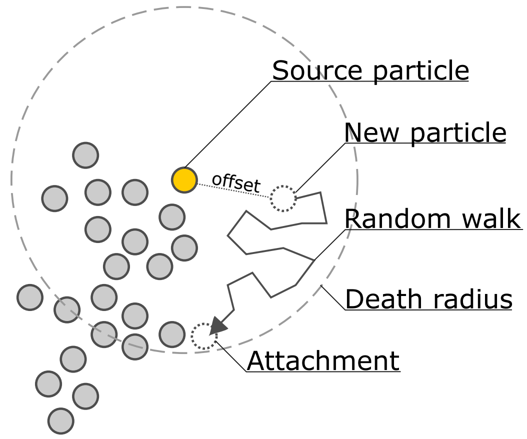

Figure 2: The lichen growth simulation procedure. A new particle spawns a distance away from a source particle, moves randomly and then attaches to the existing cluster.

The simulation for the particle continues until the particle is attached to the cluster, the particle dies or a maximum number of steps p_{maxPath} is reached. This is illustrated in figure 2. In each step, a particle dies if its distance to the source is greater than the death radius $s_{\text{deathRadius}}$. If the particle survives a step, it checks for collisions between itself and all other particles in the cluster. A collision happens if the distance between particles is less than $s_{\text{particleRadius}}$. Then, the probability that the particle will be aggregated to the cluster P is calculated according to equations 1, 2, and 3.

$$P = E \cdot A \tag{1}$$

$$A = s_\alpha + (1 - s_\alpha) \cdot e^{-s_\sigma \cdot (n - s_\tau)^2} \tag{2}$$

$$E = \min(I, L, W) \tag{3}$$

Where A is the theoretical aggregation probability, n is the number of particles at a distance less than s_ρ from the particle. E is the environmental aggregation probability, where I, L and W are functions for indirect lighting, direct lighting and water. The functions $I, L, W \colon \mathbb{R} \to \mathbb{R}$ are defined by curves indicating the lichen species' adaptability to the given condition. The particle's position in the input texture space determines where the adaptability functions are sampled. Direct light is a function $2h^2$ where h is the heightmap value at the particle's position. This value comes from the assumption that direct light will not reach into the deeper areas of the heightmap, and fall off with the square of the height. Indirect light is light that has been scattered by surfaces after leaving a light source, which should have a less steep falloff than direct light, and is therefore equivalent to h. Water amount is assumed to increase at lower heights, and is equal to $1 - h$.

The conjunction of A and E gives the probability that the particle will be aggregated. If the particle is aggregated, it is fixed in a position that is offset $2 \cdot s_{\text{particleRadius}}$ from the particle it collided with. If the particle is not aggregated, it is killed. Otherwise, if the particle never collided, it is moved $s_{\text{stepDistance}}$ in a random direction. Once the simulation is done, the result is a set of clusters, each containing a set of particles representing the distribution of the lichen cluster in lichen space.

3.2.2 Texture generation. The approach in [Desbenoit et al. 2004] uses cellular textures [Fleischer et al. 1995] to create the visual results from the lichen simulation. The output of the [Fleischer et al. 1995] approach is a collection of 3D geometry, something that is not possible to unify with the 2D texture output of the proposed approach. Instead, the output will combine color, height and roughness data to achieve visually plausible results without the need for geometry.

There are essentially three types of lichens. Two of the types are foliose and fruticose lichens, which are difficult to approximate without using 3D geometry due to their structure. [Desbenoit et al. 2004] uses one approach for those two types, and a different approach for the third type. The third type is crustose lichens, which grows directly on surfaces as an extra layer of material, a perfect candidate to approximate using textures. Observing crustose lichens from photographs such as the one in figures 1 and 5, they have a cellular structure with bright cells separated by darker crack-like formations. The structure is reminiscent of a Voronoi pattern, which is rather simple to construct graphically. A Voronoi texture of this type is created by a set of points on a plane, when each texel (texture pixel, the smallest element of a texture) is colored according to its Euclidean distance to the closest point.

Each cluster from the simulation defines a set of points on a plane, when the lichen particles are mapped to texture space. The points are used to generate a Voronoi pattern. First, the distance from a texel is calculated with texture tiling in mind, so that clusters that span the texture boundaries are not clipped but instead tiled. Then, after the distance is calculated, 2D Perlin noise is added to make the Voronoi pattern less uniform along its edges. The resulting distance is then mapped so that small distances stay small, large distances become small, and medium distances stay medium. The final resulting distance is used to directly determine the lichen height and roughness, then to determine the color by multiplying it with s_{color}. The centers and edges of the cells become dark.

The Voronoi pattern generation is handled by the graphics processing unit (GPU) in a fragment shader program. The fragment shader program is executed once per texel per cluster in ideal cases, but GPU memory restrictions lead to clusters being split up into multiple overlapping Voronoi patterns computed separately. This is an efficient way to calculate the Voronoi patterns, as using the central processing unit (CPU) would mean intractable computation times. When each cluster has been processed, the color, height and roughness results are applied to the textures. The color result is overlaid on the color texture, while the height and roughness results are scaled then added to the heightmap texture and roughness textures respectively. The result is high-roughness, colored lichens that are slightly raised from the surrounding surface, a reasonable approximation of crustose lichens.

4 EVALUATION

In this section, both the evaluation method and the evaluation results are presented. The major part of the method is how the user study is conducted, with smaller parts about the expressivity and performance evaluations.

4.1 Method

The main part of the evaluation is a user study aimed at determining how the simulation parameters affect the perceived realism of the lichens. Combining the information about perceived realism with the execution time of the simulation then yields interesting insight into how to minimize the execution time while sacrificing as little realism as possible.

4.1.1 Stimulus generation. To generate the stimuli for the user study, a set of parameters that affect the perceived realism while also having a clear impact on the execution time were selected. These parameters are the scale of the lichens, the compactness of each cluster and the total number of clusters (density), which are quantified according to table 1 where they are also given unique identifiers. Limiting the tested parameters to these three is based on the fact that these parameters have the clearest impact on realism and execution time, and including too many parameters would expand the scope of this study beyond what is reasonable.

The scale of the lichens affects the size of each particle, a larger scale means a large lichen coverage of the surface can be achieved with fewer iterations than would be needed with a smaller scale.

The chosen setting values for the scale parameter is motivated by the need to show the total range of appearances achievable through varying the parameter, as well as not getting exceedingly long execution times. A scale smaller than 1x is not necessary as the lichens lose detail in those scenarios, and would lead to very long execution times. A scale larger than 5x is not necessary to display the smallest details of the lichens, so that is the upper bound for the scale.

Compactness of the lichen clusters is determined by a combination of the parameters s_α, s_σ, s_τ, and s_ρ. A compact lichen cluster has a round shape while a non-compact (lacy) cluster has a spread out shape. Figure 3 shows how the relationship between theoretical aggregation probability A from equation 2 and the number of neighbors changes depending on the compactness setting. The performance implications of compactness are not intuitively clear, but there is an empirically obvious effect. As with the scale parameter settings, the compactness settings are motivated by the need to show diverse compactness appearances. Exploring the parameters s_α, s_σ, s_τ, and s_ρ shows that clusters do not become more compact than what the compact setting achieves, and the lacy setting sufficiently shows that a lacy appearance is achievable, without exaggerating the effect. The semi-compact setting shows a reasonable medium setting between compact and lacy.

The lichen density settings are mostly motivated by the upper bound of 100 clusters. Exceeding 100 clusters gives intractable execution times. The lower bound is chosen to show a minimum amount of lichens without removing them entirely. The medium density setting is the middle point between the upper bound and no lichens.

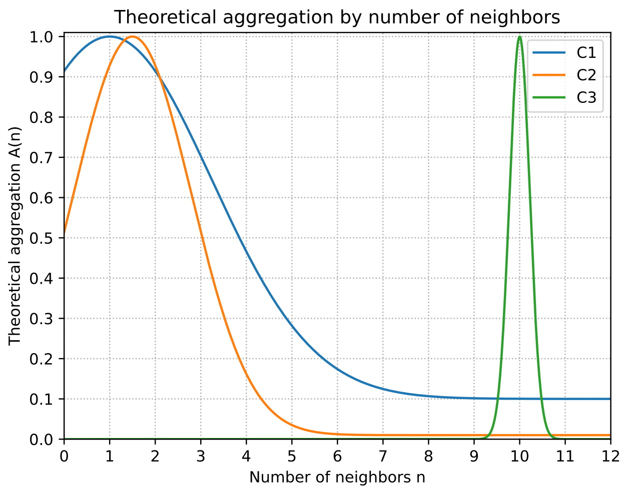

Figure 3: The relationship between the theoretical aggregation probability and the number of neighbors, depending on different compactness settings as in table 1, where C1 is the most compact and C3 is the least compact. A higher theoretical aggregation probability makes it more likely for a particle to attach to a lichen cluster.

Table 1: Quantification of the parameters used for stimulus generation.

ID	Name	Quantification
S1	small scale	1x
S2	medium scale	2x
S3	large scale	5x
C1	compact	$s_\alpha = 0.1$, $s_\sigma = 0.1$, $s_\tau = 1$, $s_\rho = 1$
C2	semi-compact	$s_\alpha = 0.01$, $s_\sigma = 0.3$, $s_\tau = 1.5$, $s_\rho = 2.25$
C3	lacy	$s_\alpha = 10^{-5}$, $s_\sigma = 10$, $s_\tau = 10$, $s_\rho = 5$
N1	low density	10 clusters
N2	medium density	50 clusters
N3	high density	100 clusters

Each stimulus is associated with a set of parameters combining scale, compactness and density. The full set of stimuli comprises all the combinations of the parameter settings, leading to a total of 27 stimuli. The simulation is implemented as an editor tool in Unity[2], and initially run once for each parameter setting, generating the color, heightmap and roughness textures for the surface. For running the simulation, an Intel® Core$^{\mathrm{TM}}$ i5-11300H 3.10GHz CPU and an NVIDIA GeForce RTX 3060 Laptop GPU are used. The input textures to the simulation depict a natural rock surface, and were exposed to a weathering simulation with hydraulic erosion greatly inspired by [Beyer 2015], and also [Cordonnier et al. 2017; Krištof et al. 2009], and discoloration inspired by [Dorsey et al. 2005; Merillou et al. 2011; Shahidi et al. 2005]. The rationale behind this is that lichens take a long time to grow, so a weathered underlying surface texture would provide a higher level of realism and not interfere with the realism of the lichens. The chosen textures depict a natural rock surface rather than a man-made stone tile surface, to best fit the mental model of where lichens grow that study participants are expected to have. The texture resolution was chosen as 2048×2048 pixels to be high enough for participants to not notice individual pixels, which could distract from the intended realism measurements.

To create a stimulus, the textures from one simulation have to be put into a context where they can be observed and perceived correctly. In a scene in a 3D modeling software (Blender[3] was used in this study), the textures are applied to a horizontal plane, and a model of a rock is placed on the plane to block the horizon as well as to give reference to the lichen size and lighting conditions. The scene is lit using environmental lighting captured from reality, and the camera is angled so that the lichens are in view. Figure 4 shows a set of stimuli that highlights the effects of the different parameters. The stimuli are cropped to better show the lichens.

4.1.2 Experiment design. The approach to evaluating the perceived realism of the simulation is to quantify how each parameter setting contributes to the perceived realism. This is achieved by having participants compare two stimuli which only differ in one parameter and answer which one is more realistic to them. This constitutes a two-alternative forced choice (2AFC) design [Cunningham and Wallraven 2011].

[2]https://unity.com/
[3]https://www.blender.org/

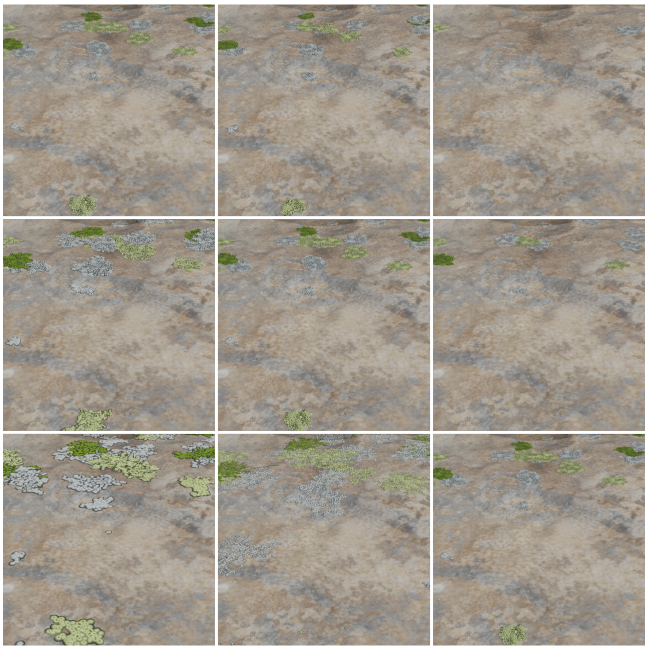

Figure 4: Examples of stimuli showing the difference between parameter settings, cropped to better display the lichens. From top to bottom, the left column shows scale (S1, S2, S3), the center column shows compactness (C1, C2, C3), and the right column shows number of clusters (N1, N2, N3).

Each trial consists of two stimuli being shown simultaneously on the left and right sides of a computer monitor along with a button for each image that participants click to indicate which one is more realistic. Once a button is clicked, a new trial begins, with a new pair of stimuli. The left and right ordering of the stimuli is randomized for each stimulus pair for each participant, and the order of the trials is randomized for each participant. The number of trials is the number of pairs of stimuli which only differ in one parameter, generating a total of 75 trials.

Before the experiment, participants consent to partaking, and are presented with written instructions. The instructions only explain the general appearance of a stimulus, that lichens will be visible in the stimuli, how a trial is completed and how many trials there are. One concern is that participants might not be familiar with the appearance of lichens, so participants are shown three reference photographs as seen in figures 1 and 5 of lichens that have a similar appearance to those in the stimuli. The lichens in the reference photographs represent only a subset of all real crustose lichen appearances, and participants with a strong familiarity with lichens might notice this. It is necessary to not include all types of appearances in order to manage the scope of the experiment, and the choice of which appearances that are included is not expected to have a significant impact on the realism results.

4.1.3 Participants. The experiment takes the form of an online survey on a custom platform to give full control over the procedure. There are implications of using an online survey, stemming from that it is impossible to get the amount of control that an in-person user study affords, but the ease of distribution and participation makes it perfect for this study.

Figure 5: Two of the three reference images shown to participants in the user study. The left image is from Pxhere[4], and the right image from Flickr[5].

[4]https://pxhere.com/en/photo/1369678
[5]https://www.flickr.com/photos/gemstone/2624337546/in/photostream/

Before the main user study, a small pilot user study was conducted, to verify that it was possible to measure any difference in perceived realism between parameter settings. Amazon Mechanical Turk (MTurk) was used to recruit 10 participants from anywhere in the world, of any age, and with any technical computer graphics expertise. MTurk was used to get results quickly, and to not exhaust other recruitment channels before the main user study was conducted. MTurk workers were paid to participate, which Redi and Povoa discovered might have adverse effects on the reliability of results, but promises to give a higher rate of task completion [Redi and Povoa 2014].

For the main user study, two groups of participants were recruited. The first group is graduate and undergraduate students at KTH Royal Institute of Technology and BTH Blekinge Institute of Technology studying computer science. This group consists of 49 mostly young adults mostly originating from Sweden with considerable technical knowledge. The second group consists of 26 employees from the video game company Resolution Games. This group is made up of adults with high technical knowledge and experience, also mostly from Sweden with only a handful of participants from other countries. These two demographic groups are quite similar, large enough to yield interesting results but small enough to not require a large amount of participants to generalize the results to that demographic group.

4.1.4 Expressivity. A second evaluation objective is to explore the expressivity of the procedural model, which means finding the limits of what the model is able to generate. The model can be thought of as a function, with a domain and a range, with the set of all parameter settings making up the domain and the set of all simulation results making up the range. Note that both the domain and range are infinite sets, so exploring the expressivity of the model exhaustively is not possible. Decisions have to be made about what parts of the domain to evaluate in order to reach the most varied results and therefore see a representative sample of the range of the model. This is done by varying parameters one at a time, seeing their effect, and deciding which values would produce varied results.

In this exploration, all parameters are manipulated, not just those selected and presented in table 1. A sample of the results can be found in section 4.2.3

4.2 Results

4.2.1 Realism results. The most interesting results are the participants' realism ratings of the stimuli. From the stimulus ratings, the perceived realism contribution of each parameter setting listed in table 1 can be extracted. The parameter settings are scored in the following manner for each participant: every parameter setting starts with a score of 0, indicating it was neither rated as realistic or unrealistic. In a trial, the stimulus pair differs in one parameter, meaning each one of the two stimuli represents one setting for that parameter. The parameter setting represented by the stimulus selected by the participant gets its score incremented, while the other parameter setting gets its score decremented. This scoring takes place for each trial, and the score for each parameter is then normalized into the range [-1, 1].

Given the length of the experiment, and its format of an online survey, there is a risk that participants become exhausted part of the way through the experiment. An exhausted participant might quit the survey, or start selecting e.g. only the left image for every trial. The random order of the trials is partially meant to ensure that all stimuli are rated somewhat fairly, since it is random which stimuli pairs appear near the end of the experiment, when participants run the risk of being exhausted. In addition to this, if a participant has a significantly skewed ratio of selecting left and right, her answers are disregarded. No participant had such a skewed ratio, however.

Realism score results can be seen in figure 6. The box plots are constructed from the collected realism scores of each participant from the main user study, and they are grouped by parameter setting category. The orange lines are the median scores for each parameter setting, and the red crosses mark outliers. From the box plots, S1 appears to contribute positively to perceived realism, and that S3 contributes negatively to perceived realism. While the compactness and density parameter settings do not present as clear a preference for the realism, C2 and N2 are slightly higher in realism contribution compared to the other settings in their categories.

A paired two-tailed t-test on the scores of each pair of parameter settings in each category (e.g. S1 and S2 or C1 and C3) reveals the statistically significant differences in perceived realism. The means are different from each other except when comparing C1 against C3 and N2 against N3 ($p < 0.05$). Figure 1 shows the stimuli which correspond to the most realistic, the least realistic and the medium realistic parameter settings, selected by the medians ignoring statistical significance.

4.2.2 Performance results. The performance of a simulation is measured using its execution time with given parameter settings. During stimulus generation, the execution time is measured for each stimulus. The simulation was performed a total of three times to get a more accurate execution time measurement. Then, for each parameter setting in table 1, the average execution time for a stimulus with that parameter setting was calculated. Those results can be seen visualized in figure 7.

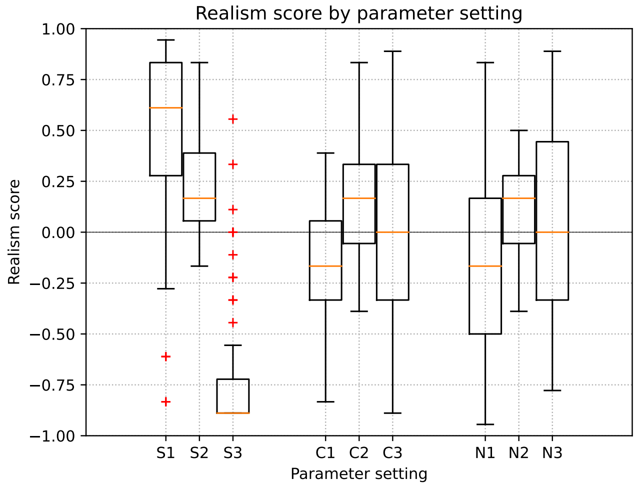

Figure 6: Realism scores based on the participants' ratings of the lichen stimuli, grouped by scale (S), compactness (C) and density (N) as defined in table 1. A higher score indicates a higher level of perceived realism.

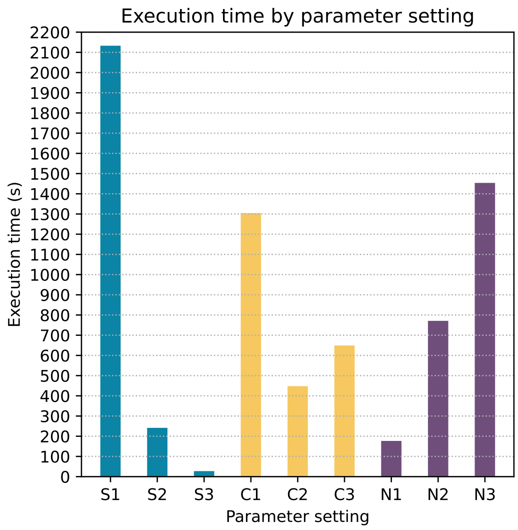

Figure 7: The mean execution time of the lichen growth simulation grouped by scale (S), compactness (C) and density (N) as defined in table 1, measured in seconds. A parameter setting with a longer execution time means that that parameter setting contributes to an overall slower execution.

As can be seen in figure 7, the execution times differ depending on the parameter settings. Expectedly, the scale parameter has a large effect on the execution time, with larger scale meaning shorter execution time. The execution time appears to be proportional to the density parameter, which is also expected. The compactness parameter, however, does not have an obvious relationship with the execution time. Highly compact lichens take the longest to simulate, and lacy lichens take shorter. The unexpected result is that semi-compact lichens are the fastest to simulate, despite not being at any extreme of the parameter settings.

4.2.3 Expressivity results. Varying the parameters as outlined in 4.1.4 results in the textures shown in figure 8. The results shown are a sample of the possible attainable results, illustrating the procedural model's ability to produce a wide range of textures.

Figure 8: Images resembling the stimuli in the user study, but with parameters selected in order to illustrate the expressivity of the procedural model.

The model is also able to generate intermediary results, such as the lichen heightmap before it is added to the main heightmap. This can act as a mask showing where lichens are growing in an image, as only the regions occupied by lichens are opaque. Figure 9 shows an example of how the lichen mask can be used to control the emission of an object, creating a glowing lichen effect.

5 DISCUSSION

In this section, the results and their implications and potential errors are discussed in order to answer the research questions. Expressivity, perceived realism and performance are handled separately, to then be combined to answer the overarching research question.

5.1 RQ1: Expressivity

Clearly, the procedural model has a wide expressivity in the context for which it was constructed. Figure 8 shows a small subset of the possible results, which despite its small size demonstrates that the model can produce quite varied results. Lichens can be given arbitrary color, shape, height, scale, and various other properties, giving users the ability to create any lichen appearance. Especially when using the intermediary results, the lichen appearance can be significantly modified at the discretion of the user's creativity.

Outside the intended context, the model has limited expressivity. If the generated textures are applied to an object that is not a horizontal plane, the result can be unexpected. The textures can be tiled to expand to essentially infinite size, and wrapped around an

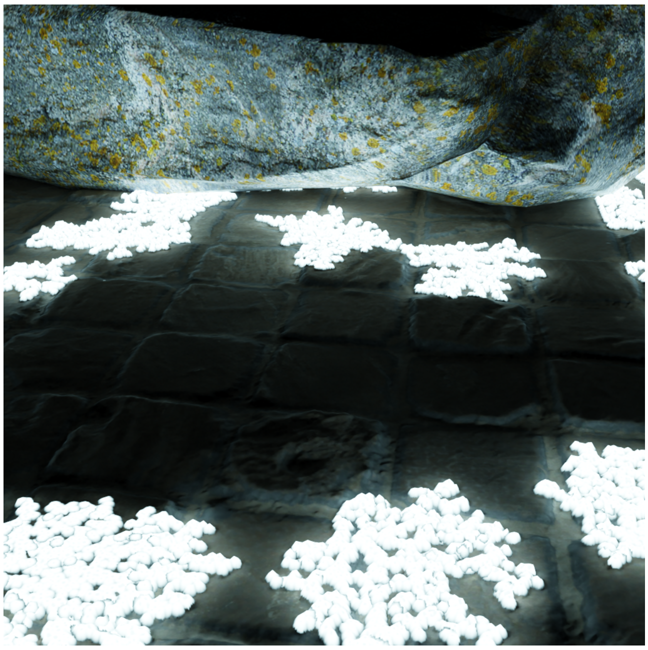

Figure 9: The result of using the lichen mask to apply an emissive effect with a light blue tint under low-light conditions. The original image, using the same simulation parameters, is the lower-right image of figure 8.

object according to user instructions. However, the lichens were simulated as if they existed on a flat plane with regard to environmental factors, meaning lichens will appear to grow in unrealistic regions when wrapped around a different object.

The model is unable to generate lichens of the foliose or fruticose types, severely limiting the model's ability to represent lichens in general. In fact, even the crustose lichens that can be generated will have limited realism if rendered in a context that does not take the heightmap into account. Without a heightmap, lichens will look less like a layer on top of a surface, and more like colorful patches of the surface itself. A possible remedy in contexts where heightmaps cannot be used is to generate a normal map from the heightmap and use that instead. Normal maps are textures that define how each point of a surface is oriented, rather than how the surface is displaced, and they are often better supported in real-time rendering.

Execution time is practically a limit to the expressivity of the model. Certain parameter settings which have a long execution time, e.g. a dense covering (N3) of compact (C1) and small-scale (S1) lichens, are theoretically possible but very difficult to generate. Despite this limitation, users can use the results in section 4.2.2 to first prototype simulations using settings with short execution time before committing to the intended simulation with a potentially long execution time.

Smith and Whitehead [Smith and Whitehead 2010] propose a method for evaluating the expressivity of procedural models called expressive range. The method relies on defining metrics that can be applied to the output of a procedural model and then visualizing using a 2D histogram plot how the output is distributed in terms of the metrics. This method requires a very large number of outputs to be generated, so unfortunately due to the intractable execution time of the present procedural model, the expressive range is impossible to evaluate.

5.2 RQ2: Realism

Most of the parameters have a noticeable and statistically significant effect on perceived realism, judging by the results in figure 6. The parameter with the most noticeable effect on realism is the scale, which is to be expected from visual inspection of the stimuli. Clearly, the smaller-scale lichens more closely resemble the reference images.

More interesting is the perceived realism of the compactness and density parameters. The semi-compact lichens appear to be the most similar to the reference images, and in terms of statistical significance, the only conclusion that can be drawn is that semi-compact lichens are the most realistic. Participants might have seen the compact lichens as far too man-made as they are very round, while the lacy lichens share very little resemblance with the reference images.

The acquired results depend heavily on the efficacy of the scoring procedure described in section 4.2.1. The scoring assumes that the scale, compactness and density settings contribute equally to the perceived realism. This might not be the case, in fact the results suggest that the scale parameter setting could overshadow the other parameters. For instance, a participant might have compared two stimuli both with large-scale lichens (S3) but a difference in compactness and not noticed the difference in compactness, leading her to pick one of the stimuli at random. If this is the case, the perceived realism results would be less reliable. Hopefully, separating the parameters and comparing all possible pairs of settings guarantees that participants perceived the stimuli as expected in most comparison instances.

Comparing pairs of stimuli differing in a single parameter is only one method of comparison. It is possible to compare pairs differing in two or even three parameters. Doing this might prove that parameters have a different impact on realism when combined, compared to their individual impacts. While those results would be interesting, the acquisition of those results would force the study design to be more complicated, and the number of trials would be greatly increased. The results that were acquired in this study are deemed to be sufficient to draw conclusions about perceived realism, as the parameters' combined impact would likely be weaker than the individual impacts.

Another source of error in the perceived realism results is the participant groups. The pilot study results are not consistent with the main study results, showing different mean scores for nearly every parameter setting. This might be due to the different demographics of the main study groups and the pilot study group. The pilot study participants were from the USA and India, with a larger variance in age and a generally smaller amount of experience with computer graphics. This differs significantly from the main study demographics. Also, as noted by [Redi and Povoa 2014], paid participants tend to give less reliable answers than volunteers. Therefore, it is questionable whether the pilot study results can be considered reliable. Due to the small sample size, none of the results from the pilot study are statistically significant either.

Comparing the two groups of participants in the main study strengthens the reliability of the results. The groups answered remarkably similar to each other. Comparing the means of the results for each parameter setting using an independent two-tailed t-test, the means are only different in the case of S3 and C2 ($p < 0.05$). Even then, the overall relationship between the parameter settings are identical between the two groups. These factors, as well as the two groups' similarity in age, geographic spread and computer graphics experience motivates the combination of the results from both groups to attain a larger sample size. The factors also mean that the results can be considered more reliable.

Generalizing the perceived realism results to the global population is not motivated by this study. Given the unreliability of the pilot study, those results can not be considered representative of the global population. The demographic specificity of the main study participants also makes it hard to motivate the generalizability of the results. However, thanks to the reliability of the main study results, the results should at least be generalizable to the sampled population. With confidence, it can be concluded that the results are generalizable to a western, relatively young group with significant experience with computer graphics.

5.3 RQ3: Performance

From the results in section 4.2.2, it is clear that each parameter setting has an effect on the execution time of the procedural model. The scale and density effects on performance are easily explained. Larger scale leads to fewer iterations needed, while a higher density means more particles and therefore more collision checks. These parameters have an intuitively understandable linear relationship with execution times.

Compactness does not appear to have a linear relationship with execution time. This is verified empirically; the theoretical explanation is less clear. Differences in compactness affect the theoretical aggregation probability, as compactness depends on the values of the parameters in equation 2. Figure 3 helps in understanding the theory. For C1, the aggregation probability leads to a quick accumulation of particles. C2 is more selective about when aggregation is possible leading to a slower growth in particle numbers. C3 is extremely selective, but because s_ρ is large, it is able to look for neighbors over a long distance, leading to many particles that stretch out in a lacy pattern. According to this, it is expected that C1 and C3 lead to longer execution times, with a minimum execution time between them, where aggregation probability is low.

5.4 Realism and performance optimization

Having answered the sub-questions RQ1, RQ2, and RQ3, it is now possible to answer the overarching research question: *How can lichen growth on horizontal stone surfaces be generated procedurally while maximizing perceived realism and performance, and guaranteeing expressivity?*

The most critical part of the question concerns how to maximize realism and minimize execution time. The maximum realism is achieved with a medium density, semi-compact, small scale covering of lichens. A stimulus using these parameter settings is seen in the leftmost image of figure 1. Unfortunately, scale simultaneously has the largest effect on perceived realism and on execution time. This is somewhat offset by the most realistic compactness setting also being the cheapest performance-wise and the most realistic density possibly being only the second-most expensive. However, there is still clearly a tradeoff between perceived realism and performance. Lichens of a large scale are conducive to good performance, but detrimental to perceived realism. The tradeoff between realism and performance is therefore mostly decided by the scale parameter.

In the scenario posed in this study, larger lichen scale means that each particle will appear larger, which is not necessarily the case in all scenarios. If the input texture is small, but meant to be tiled across a surface many times, the scale of the lichens has to be large in order for the lichens to appear appropriately sized. This way, the execution time is limited while still producing an appearance that would be perceived as realistic. A problem that arises with tiling the lichen texture is that although the lichens can cross over texture boundaries, repeating the tiling over a large surface might lead to an unrealistically uniform visual result. There are established solutions to this problem, such as modifying the texture to tile better, [Stam 1997] proposes a solution specifically for planes.

The main research question does not have one clear answer. The simulation technique used in this study is effective in simulating lichen growth with a high level of expressivity, but carries with it a fundamental conflict between perceived realism and performance.

5.5 Future work

There are multiple ways to extend the procedural model and evaluation of this study to achieve additional significant results.

This study evaluates the realism of the procedural model by comparing the results of simulations with different parameter settings. While this is able to answer questions about how the model can be used optimally, it cannot answer the question whether the lichens are realistic overall. Comparing the model's results with matching photographs of reality in a user study could answer such questions, and would be an interesting addition to this study.

One major limitation of this study and the procedural model in general is its long execution time. In order to achieve the most realistic results the execution time grows to be quite long. If the algorithm could be improved so that it can produce the same or visually equivalent results in much shorter time spans, that would open possibilities for additional evaluation and findings. Being able to produce tens of thousands of output textures in a reasonable timeframe would make it possible to apply the expressive range method [Smith and Whitehead 2010] in the evaluation of the procedural model. One suggestion for shortening the model execution time includes parallelizing the growth simulation so that each cluster is simulated in parallel. Using a different execution environment, i.e. implementing the model using C++ instead of C# in Unity might also improve performance somewhat, without removing any features.

Apart from evaluation and performance improvements, it would also be interesting to expand the simulation in multiple ways. Generalizing the simulation to be able to simulate lichen growth on an arbitrary triangle mesh, similar to the original paper [Desbenoit et al. 2004] is one way of expanding the simulation. Improving the simulation realism by making the lichens die and decompose after growing to a certain age, as well as changing color with age would be a further extension. Coupling the lichen growth with the appearance of the underlying surface by introducing discolorations caused by the lichens as in [Chen et al. 2000] is another way of extending the model. Performing a similar evaluation to the one in this study after such extensions would be interesting.

6 CONCLUSION

The procedural model presented and evaluated in this study is able to simulate the growth of lichens on horizontal stone surfaces. The model is expressive to such an extent that essentially any crustose lichen appearance, limited to the realism of the model, is possible to generate. Human participants perceive generated lichens that are small in scale, semi-compact and cover the surface with a medium number of clusters (density) as the most realistic lichens. Execution times of the lichen simulations are long, and higher realism is coupled with longer execution time in most scenarios. In particular, it is the scale of the lichen that has the largest effect on realism and execution time, forming the basis of the tradeoff between realism and performance.

ACKNOWLEDGMENTS

This thesis is done in association with Resolution Games [6]. I want to thank my supervisor at KTH Christopher Peters, my examiner Tino Weinkauf, supervisor at Resolution Games Henrique Furtado, as well as other colleagues, friends, and family. The success of this thesis is not in small part thanks to you all.

REFERENCES

Vernon Ahmadjian. 1973. *The lichens.* Academic Press, New York.

David Bachman. 2019. Procedural organic modeling. In *ACM SIGGRAPH 2019 Educators Forum.* 1–59.

Djalma Bandeira and Marcelo Walter. 2009. Synthesis and transfer of time-variant material appearance on images. In *2009 XXII Brazilian Symposium on Computer Graphics and Image Processing.* IEEE, 32–39.

Hans Theobald Beyer. 2015. *Implementation of a method for hydraulic erosion.* Bachelor's thesis. Technische Universität München.

Chaona Chen, Oliver GB Garrod, Philippe G Schyns, and Rachael E Jack. 2020. Dynamic Face Movement Texture Enhances the Perceived Realism of Facial Expressions of Emotion. In *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents.* 1–3.

Jie Chen, Hans-Peter Blume, and Lothar Beyer. 2000. Weathering of rocks induced by lichen colonization—a review. *Catena* 39, 2 (2000), 121–146.

Yanyun Chen, Lin Xia, Tien-Tsin Wong, Xin Tong, Hujun Bao, Baining Guo, and Heung-Yeung Shum. 2005. Visual simulation of weathering by γ-ton tracing. In *ACM SIGGRAPH 2005 Papers.* 1127–1133.

Guillaume Cordonnier, Eric Galin, James Gain, Bedrich Benes, Eric Guérin, Adrien Peytavie, and Marie-Paule Cani. 2017. Authoring landscapes by combining ecosystem and terrain erosion simulation. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–12.

Douglas W Cunningham and Christian Wallraven. 2011. *Experimental design: From user studies to psychophysics.* CRC Press.

Barbara Cutler, Julie Dorsey, Leonard McMillan, Matthias Müller, and Robert Jagnow. 2002. A procedural approach to authoring solid models. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 302–311.

Brett Desbenoit, Eric Galin, and Samir Akkouche. 2004. Simulating and modeling lichen growth. In *Computer Graphics Forum*, Vol. 23. Wiley Online Library, 341–350.

Julie Dorsey, Alan Edelman, Henrik Wann Jensen, Justin Legakis, and Hans Køhling Pedersen. 2005. Modeling and rendering of weathered stone. In *ACM SIGGRAPH 2005 Courses.* 4–es.

[6] https://www.resolutiongames.com

Julie Dorsey, Hans Køhling Pedersen, and Pat Hanrahan. 1996. Flow and changes in appearance. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 411–420.

Shaojing Fan, Tian-Tsong Ng, Bryan Lee Koenig, Jonathan Samuel Herberg, Ming Jiang, Zhiqi Shen, and Qi Zhao. 2017. Image visual realism: From human perception to machine computation. *IEEE transactions on pattern analysis and machine intelligence* 40, 9 (2017), 2180–2193.

Kurt W Fleischer, David H Laidlaw, Bena L Currin, and Alan H Barr. 1995. Cellular texture generation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 239–248.

Daniele Giunchi, Riccardo Bovo, Panayiotis Charalambous, Fotis Liarokapis, Alastair Shipman, Stuart James, Anthony Steed, and Thomas Heinis. 2021. Perceived Realism of Pedestrian Crowds Trajectories in VR. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*. 1–5.

Peter Krištof, Bedrich Beneš, Jaroslav Křivánek, and Ondrej Št'ava. 2009. Hydraulic erosion using smoothed particle hydrodynamics. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 219–228.

Aristid Lindenmayer. 1968. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of theoretical biology* 18, 3 (1968), 280–299.

Nicolas Merillou, Stephane Merillou, Eric Galin, and Djamchid Ghazanfarpour. 2011. Simulating how salt decay ages buildings. *IEEE Computer Graphics and Applications* 32, 2 (2011), 44–54.

Gary W Meyer, Holly E Rushmeier, Michael F Cohen, Donald P Greenberg, and Kenneth E Torrance. 1986. An experimental evaluation of computer graphics imagery. *ACM Transactions on Graphics (TOG)* 5, 1 (1986), 30–50.

Stuart O'Connor, Fotis Liarokapis, and Chrisina Jayne. 2015. Perceived realism of crowd behaviour with social forces. In *2015 19th International Conference on Information Visualisation*. IEEE, 494–499.

Przemyslaw Prusinkiewicz. 2004. Modeling plant growth and development. *Current opinion in plant biology* 7, 1 (2004), 79–83.

Paul Rademacher, Jed Lengyel, Edward Cutrell, and Turner Whitted. 2001. Measuring the perception of visual realism in images. In *Eurographics Workshop on Rendering Techniques*. Springer, 235–247.

Suren Deepak Rajasekaran, Hao Kang, Martin Čadík, Eric Galin, Eric Guérin, Adrien Peytavie, Pavel Slavík, and Bedrich Benes. 2019. PTRM: Perceived Terrain Realism Metric. *ACM Transactions on Applied Perceptions (TAP)* (2019).

Ganesh Ramanarayanan, James Ferwerda, Bruce Walter, and Kavita Bala. 2007. Visual equivalence: towards a new standard for image fidelity. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 76–es.

Judith Redi and Isabel Povoa. 2014. Crowdsourcing for rating image aesthetic appeal: Better a paid or a volunteer crowd?. In *Proceedings of the 2014 international acm workshop on crowdsourcing for multimedia*. 25–30.

Amir Rosenberger, Daniel Cohen-Or, and Dani Lischinski. 2009. Layered shape synthesis: automatic generation of control maps for non-stationary textures. *ACM Transactions on Graphics (TOG)* 28, 5 (2009), 1–9.

Salman Shahidi, Stéphane Mérillou, and Djamchid Ghazanfarpour. 2005. Phenomenological Simulation of Efflorescence in Brick Constructions.. In *NPH*. 17–23.

Gillian Smith and Jim Whitehead. 2010. Analyzing the expressive range of a level generator. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. 1–7.

Jos Stam. 1997. *Aperiodic texture mapping.*

Thomas A Witten Jr and Leonard M Sander. 1981. Diffusion-limited aggregation, a kinetic critical phenomenon. *Physical review letters* 47, 19 (1981), 1400.

Su Xue, Aseem Agarwala, Julie Dorsey, and Holly Rushmeier. 2012. Understanding and improving the realism of image composites. *ACM Transactions on graphics (TOG)* 31, 4 (2012), 1–10.

Su Xue, Julie Dorsey, and Holly Rushmeier. 2011. Stone weathering in a photograph. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 1189–1196.

TRITA-EECS-EX- 2022:106