Declan Russell

Specialist project: Weathering of Metal Simulation Report

Introduction:

My aim for this project is to research and try and recreate the effect of weathering on metallic surfaces. This is something that is widely used in computer graphics to create old and worn looking models except normally this process is carried out by the texture artist. This seems somewhat ad hoc so I want to try and create an application that can do this process for you.

Initial Research:

To begin with I have gathered together some images of weathering to form an image of the effect that I wish to achieve

weathered-metal, http://www.ussteel.com/ , weathered-metal.jpg

Rust on iron, http://en.wikipedia.org/wiki/Rust , Rust on iron.jpg

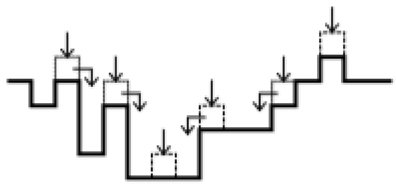
Copper Green Patina A Uncoated, http://www.searlesart.com/green-on-copper/2068/ , patina-A-uncoated-wp.jpg

Along with the obvious result of weathering rust, another phenomenon can occur in copper. This is known as a Patina. This acts in the same way as rust but forms a green coat onto the metal and does not corrode the metal. An example of where this occurs is the Statue of Liberty which get its notorious green colour from this phenomenon.

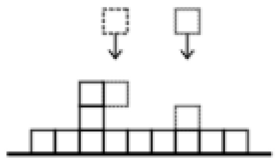
I looked into some research about the science behind rust and patinas to try and find out if there is any basic understanding of it and maybe gain some hints on how it may grow. The basic principle is that it is an oxidization of the metal in the presence of oxygen and water to create iron oxide or copper oxide in the case of patinas. Notably the water can come from just the moister in the air, this means that even if a water source is nearby the oxidisation process can still occur. Given sufficient time the oxidisation process will convert the entire material into its oxidised state.

After browsing the internet I managed to come across an ACM SIGGRAPH paper about Modelling and Rendering Metallic Patinas. This paper is paper about creating a phenomenological model to represent patinas for computer graphics based on their physical behaviour. This is not a real world representation of rust as the physics to represent this phenomenon does not exist yet. It describes the patina as a combination of multiple visible layers. From the new polished metal to a dull brown and finally reaching a reddish brown colour of the inactive oxide. To create this effect several different models had been set up which I have then experimented with implementation in c++ and openGL.

1. Steady thickening - Several points are sampled equally in the texture and given assigned an initial thickness. Intermediate points from the samples are interpolated. Finally we add some natural noise to make it look natural. Over time the sample points are increased and process repeats.
2. Random deposition – This is the model is the simplest growth model to utilize. A particle is dropped from a random position on to the texture. Where it falls onto the texture you will increase the thickness by a given value. There is also a variation in this model to allow the particle to diffuse a finite distance. E.g. lowest thickness nearest to point of collision



3. Ballistic deposition model - Again a particle is dropped from a random position above the texture and falls vertically. When it lands the thickness at this position in the texture is set to the larger of either the neighbouring point's thickness or the current point's thickness incremented by 1.



4. Directed Percolation Depinning – In this method our interface will grow in all dimensions. We also have a 2D lattice covered with a proportion of cells marked as blocked and unblocked. Unblocked points promote growth and blocked discourage it. This 2D lattice act to simulate the moisture on out metallic surface as we learned before that rust becomes much more prominent in oxygen and water rich environments.

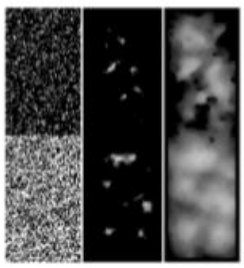


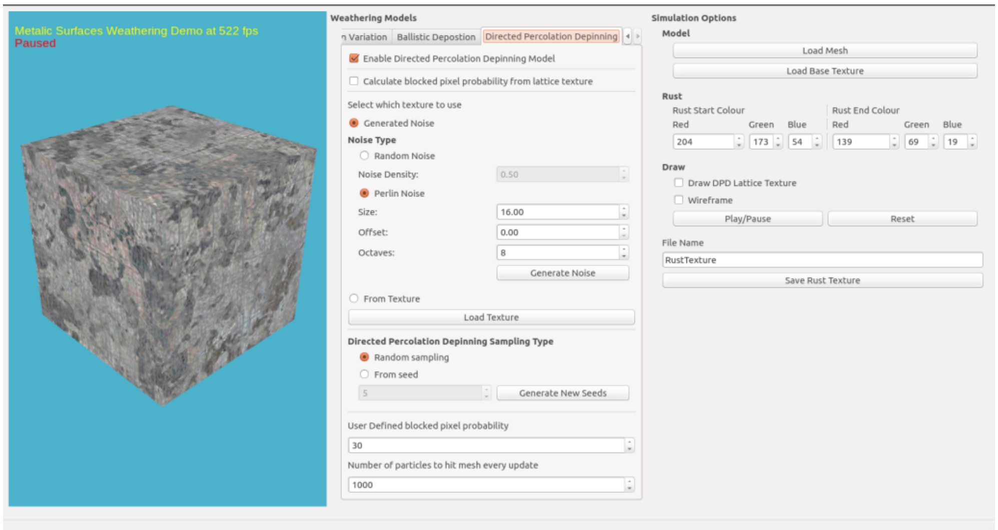
Figure 6: A lattice of blocked and unblocked cells (left), early stage of pattern formation (middle), late stage of pattern formation (right).

(Above) pictures sourced from Modeling and Rendering Metallic Patinas, Julie Dorsey Ph.D. – Massachusetts Institute of technology & Pat Hanrahan – Stanford University, ACM SIGGRAPH (1996)

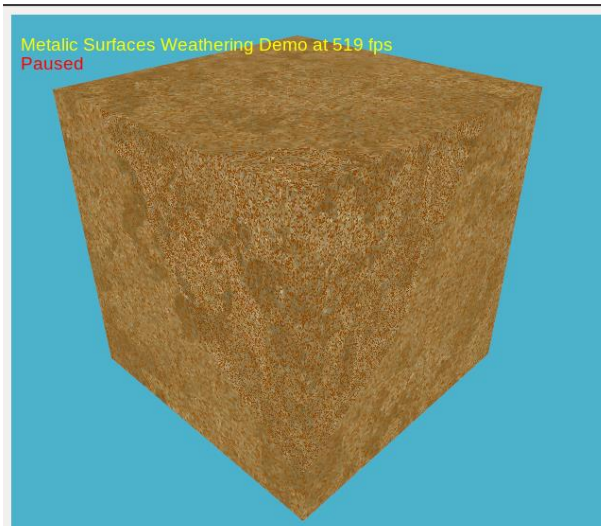
Declan Russell

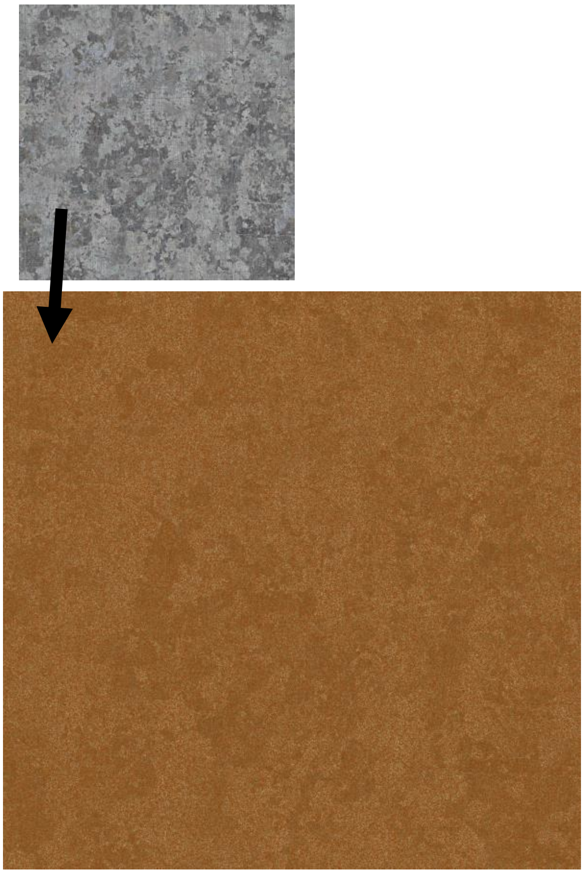
Test Models:

To represent the different layers of the rust on the surface I have just used linier interpolation between the base texture and a rust colour in the ratio relative to the height of the rust on the surface. For these examples I am just using a simple cube with an arbitrary metal texture.



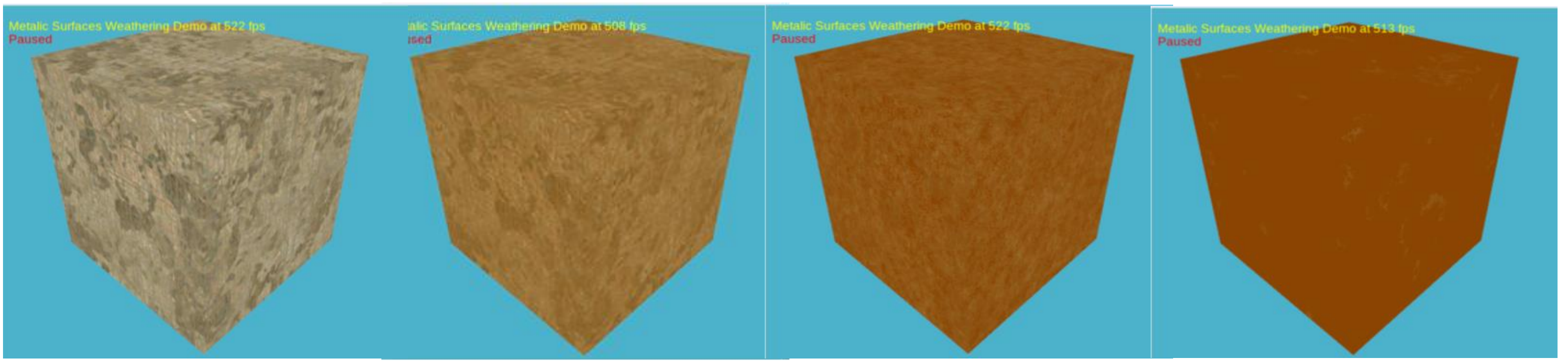
1. Random Deposition Model

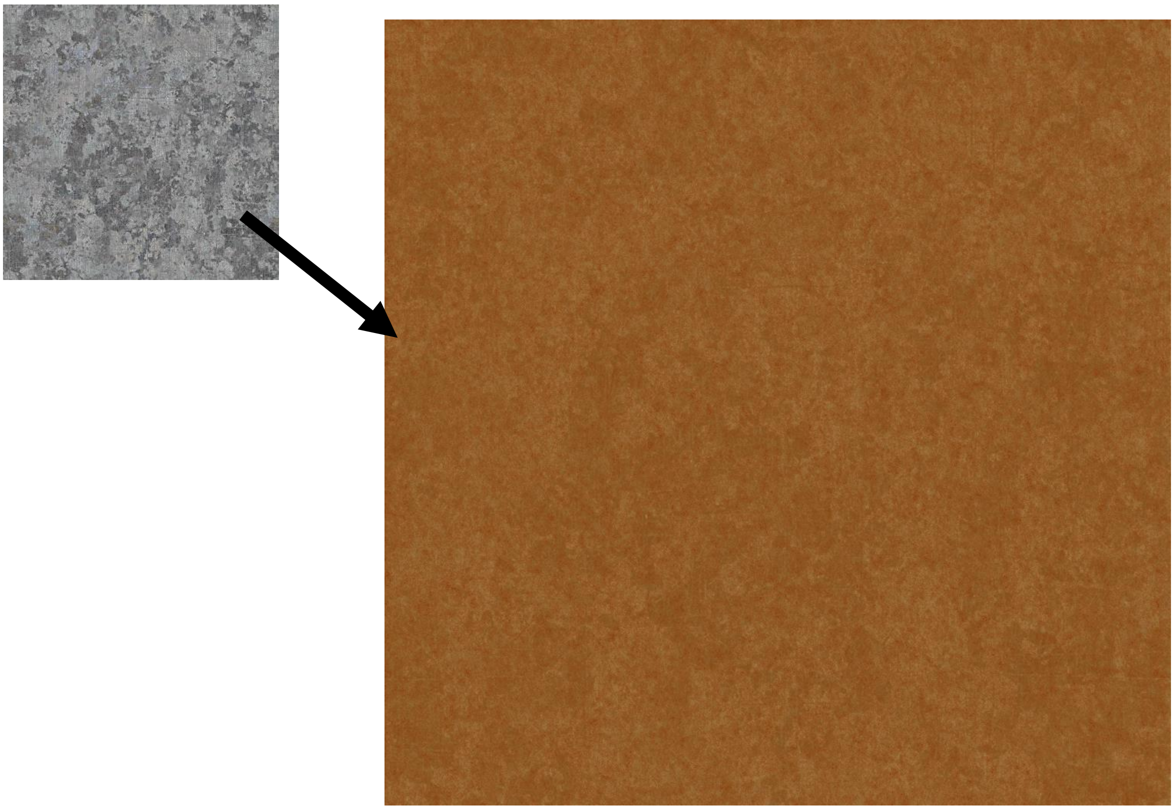




This model created a very rough finish which would grow linearly the texture. I felt that this effect result was far too grainy for actual rust.

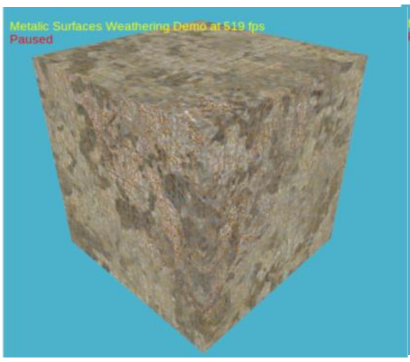
2. Random Deposition Model Variation

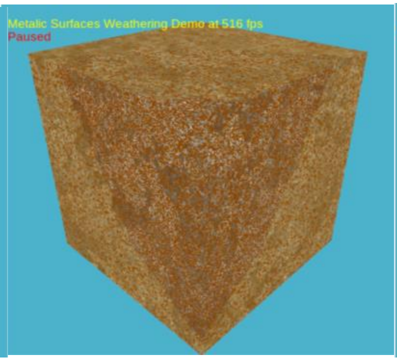


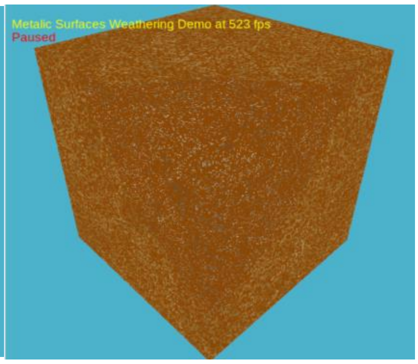


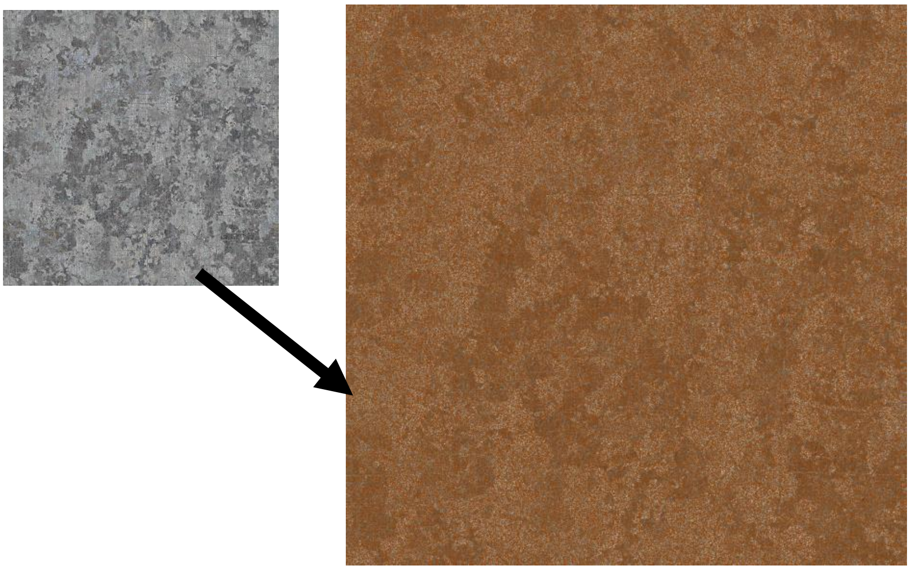
To get this model to work in real time I had to make some optimisation to the diffuse algorithm so that it would run real time. The algorithm I created would search a limited set of positions to diffuse to rather than all possible. Despite this however this model still created a much smoother finish than the original version.

3. Ballistic Deposition



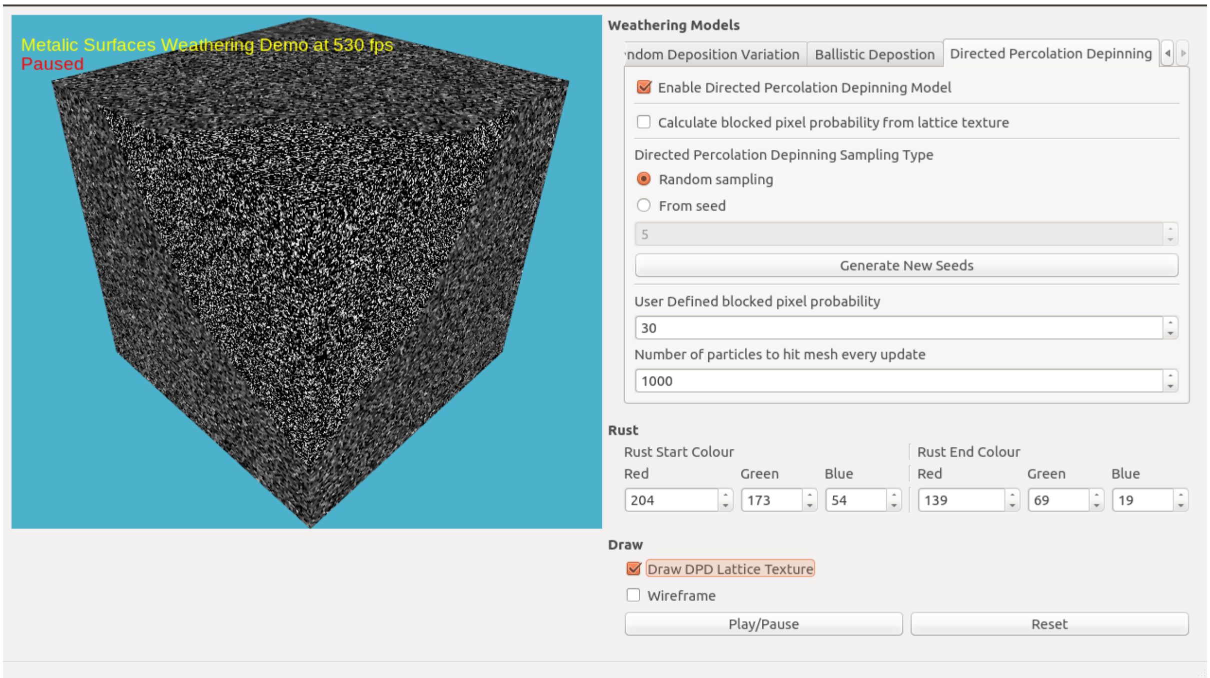




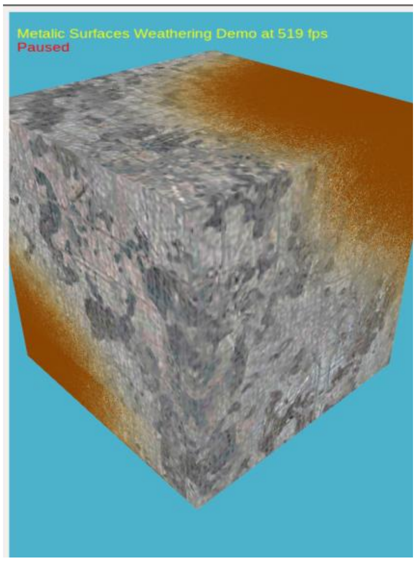


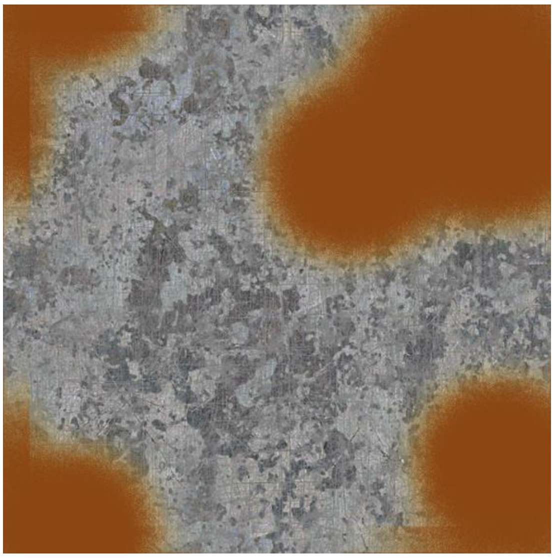
This model created a similar effect to the random deposition model previous this model produced a very grainy result to the texture.

4. Directed Percolation Depinning

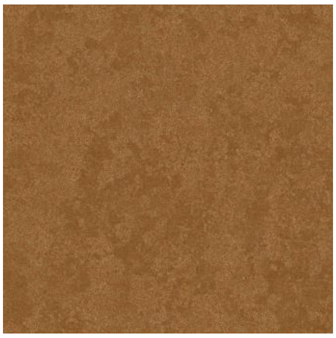


(Above) Initial Lattice



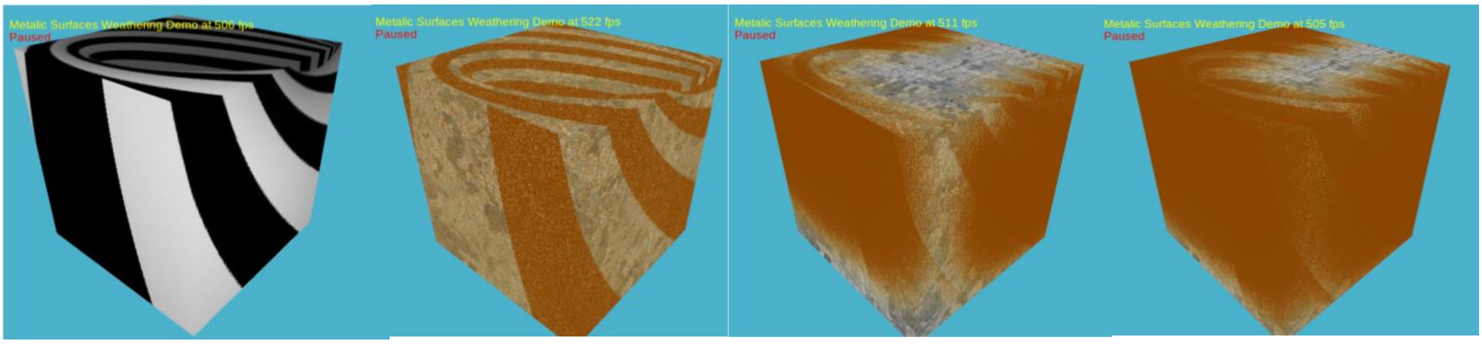


This, my favourite of the models I believe to have the greatest effect. The rust develops from initial seed locations and expands across the entire texture. The lattice texture for this example just being noise generated at random does a nice job of breaking up the rust around the edges but become more of a solid fill towards the center. I found that sampling at random created a much more varied result although did not have the growth of the seed approach.

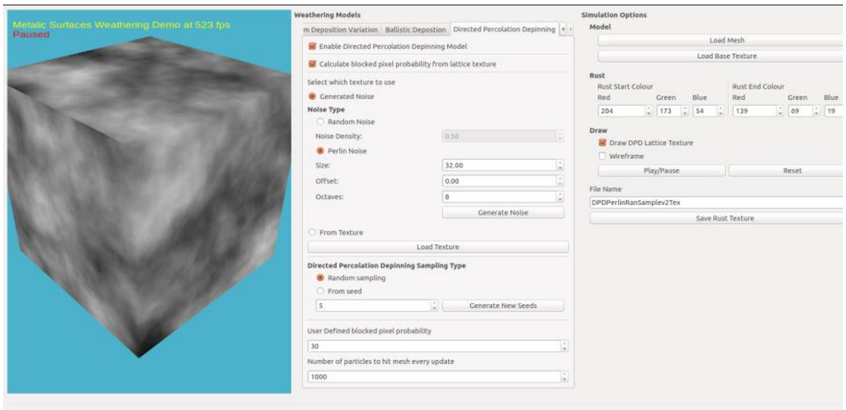


Declan Russell

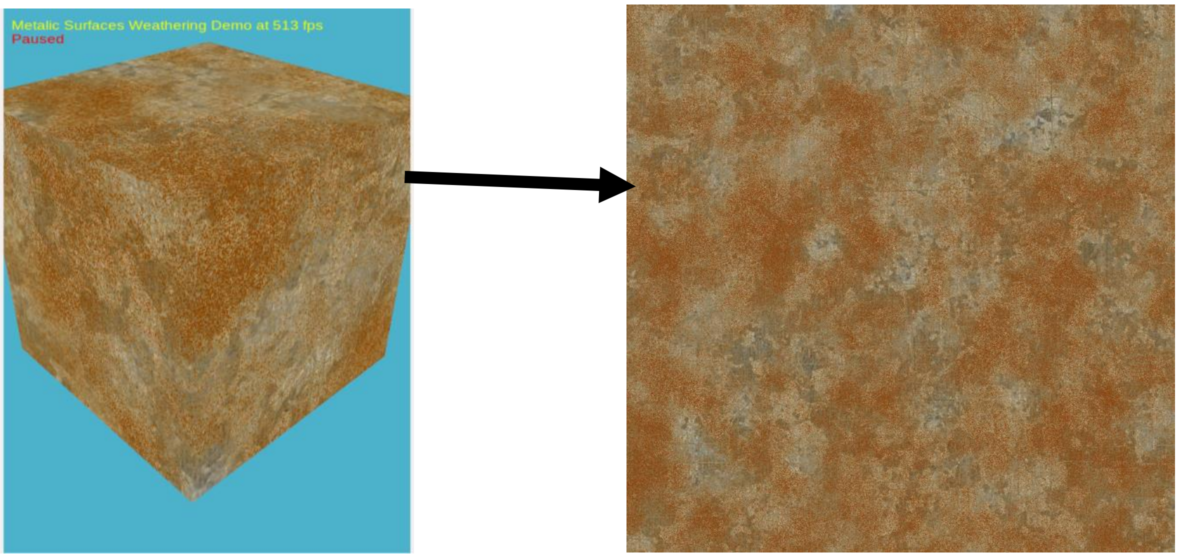
A useful effect I think of this model is that you could develop your own lattice texture to influence rust to form at a greater rate in locations of your choosing. An idea I have decided to expand upon demonstrated in the following few examples.



I decided to elaborate on this feature and try to develop a better lattice to use for the model. Perlin noise I found to be the most promising and implemented a simple algorithm to my application.



(Above) Perlin Lattice Generated



I feel that this method has created best effect so far in the project creating nice non-uniform patches around the texture that grow with time. The next step if I were to continue this research would be to look into different rendering methods to create a more realistic coloured rust as I feel that this texture still feels flat.

Specialist project: Weathering of Metal Simulation Bibliography

Wolff, D., 2011, OpenGL 4.0 Shading Cookbook, PACKT PUBLISHING – Chapter 4 Using Textures

This book is an introduction to shaders in OpenGL and common techniques used. This chapter was about the basics of using textures in your shader program. This covered loading your texture to the GPU and linking it with an Id in the shader program with uniforms. Furthermore It also described the process of linking your texture to the texture coordinates of your mesh and finally calculating the colour intensity of each fragment when you introduce lights into your scene. Something I found particularly useful in this chapter was the use of storing non-color data into textures for loading in information such as height maps or normal maps. I feel I could link this to my research as I could use it to store any weathering data across my mesh. I feel like this information was very conclusive and left me with a good understanding of the basic use of textures in shaders allowing me to set up basic textured scenes. This information is also very reliable being published work recommended by the lecturers at this University. It is also written by the author and industry professional David Wolff PhD the associate professor in the Computer Science and Computer Engineering Department at Pacific Lutheran University.

Wikipedia, 2012, Rust, Wikipedia. Available from http://en.wikipedia.org/wiki/Rust

This page was an article about the phenomenon 'Rust'. The article focused on the physics behind why and where rust happens and the chemistry behind the oxidization reaction behind the process. This served as a good basic understanding of some weathering that can happen upon metallic surfaces, why it happens and what environmental conditions are needed for the phenomenon to occur. It explains that a water and oxygen rich environment are ideal for rust to grow. As long as these conditions are met the rust will grow to eventually consume the entire metal. Although this article was quite clear about the chemistry behind the reaction it was fairly brief and dint actually describe any representation of how to simulate this effect which I will need if I am to try and imitate it into computer graphics. Another limitation is that there is no way of knowing how reliable this source is being from Wikipedia which anyone can submit to. On the other hand though it did provide some good references to other forms of weathering on metallic surfaces such as Patinas.

Wikipedia, 2014, Patina, Wikipedia. Available from http://en.wikipedia.org/wiki/Patina

This article is serves an brief description of the naturally occurring phenomenon of Patinas that form on surfaces such as stone, copper, bronze and other similar metals. This article focuses mainly on why the phenomenon occurs and gives examples of its appearance in naturally created patinas and some artificially applied patinas such as the statue of liberty which gets its green colour from the oxidization of its copper exterior. I found this article useful towards my research as it demonstrates that different types of weathering can effect a material depending on its physical properties which could be useful if I want to demonstrate multiple behaviors. For example this process does not erode away the object like rust would, sometimes Patinas as in fact artificially created to improve the structural integrity of a metallic object. This also gives me a broader horizon of weathering behaviors to research into. Again although this article was quite clear about its information it did not go into depth about the formation of this effect and has not given me any mathematical models to try and imitate this effect. Again the reliability of this article is questionable as sourced from Wikipedia which anyone can submit and edit.

Dorsey, J., and Hanrahan, P., 1996, Modeling and Rendering Metallic Patinas, ACM SIGGRAPH

A paper about creating a phenomenological model to represent patinas for computer graphics based on their physical behavior. J. Dorsey and P. Hanrahan have gained this information through a large variety of books and papers mainly about fractal concepts and surface growth and the sub surface scattering techniques in computer graphics. This is not a real world representation as the physics to represent this phenomenon does not exist yet. It describes the patina as a combination of multiple visible layers from the new polished metal to a dull brown and finally reaching a reddish brown colour of the inactive oxide. Dorsey and Hanrahan include the use of sub surface scattering techniques to imitate this effect in computer graphics. To create this effect several different models are described,

5. Steady Thickening
    a. Several points are sampled equally in the texture and given assigned an initial thickness. Intermediate points from the samples are interpolated.
6. Random Deposition
    a. This one of the simplest to implement simulates a particle that falls vertically onto the surface from a random position. Where ever this particle lands the height of the surface will increase by one increment.
7. Random Deposition Model Variation
    a. A variation of the previous model this model intimidates some diffuse upon the particle. Once the particle has fallen from a random position it will diffuse to a lower neighboring position. This is so that we create a more averaged out surface and a smoother end result.
8. Ballistic Deposition model
    a. Similar to the random deposition model this model will drop a particle from a rando point. The height of this position will be increased by the higher of either the current positions height incremented or to match a neighboring point’s height.
9. Directed Percolation Depinning
    a. The model uses an initially created lattice of blocked and unblocked points. Unblocked points incur rage the growth of our interface while opposing line the blocked points discourage growth.

This was really useful toward my project as it gave me several good models as a starting point to being experimenting with. Although the description about these algorithms was clear and fairly concise any algorithms behind these models were not discussed so I will have to research these from the papers referenced documentation. The main limitation of this paper is the fact that we can only create at most a phenomenological model of the weathering of metallic surfaces as the atmospheric corrosion of metals is not fully understood yet. This information is very reliable for two reasons. The first this paper is that it has be publish in the very prestigious ACM SIGGRAPH and the being it is from two highly qualified authors. Julie Dorsey Phd and professor of Computer Science at Yale University and Professor Pat Hanrahan Phd at Stanford University.

Barabasi, A.L., and Stanley, H.E., 1995, Fractal Concepts in Surface Growth, Cambridge University Press

This Book Dr Barabasi and Stanley explain how fractals can be used to describe the morphology of surface growth in nature. In this book Barabasi and Staley demonstrate this through numerous deposition models that use particles to gradually over time build up layers upon a subject surface that represent surface growth and surface roughness. These models include the models from the Modeling and Rendering Metallic Patinas paper. On the contrary to this though this book goes much further into detail about the algorithms behind these models and how to use them. I can use this information now to build upon the research I have done from the patinas paper and now convert the algorithms into a graphical representation of the phenomenon. Sadly these algorithms are not optimized for real time rendering and look like they will probably have a very slow computational time. This means that I will have to make some short cuts in the algorithms which could vary the end result from the intended outcome. Again this information is very trust worthy as it is written by two highly qualified authors Professor Albert-László Barabási of the University of Notre Dame and Professor Harry Eugene Stanley of Boston University.

Kim T., Sewall J., Sud A. and Lin M., 2005, A Fast Fractal Growth Algorithm, ACM SIGGRAPH

In this paper Kim, Sewall, Sud and Lin demonstrate a fast fractal growth algorithm for use in computer graphics. This was developed as fractal algorithms can be used to model a number of different natural phenomena such as snowflakes, lightning, fractures, coral, tree branches and river beds. This algorithm is very fast compared to other fractal growth algorithms and removes the need to solve a big matrix. This could be useful to my research as it is similar to the fractal patterns in which rust forms in. I could use this algorithm to try and mimic the growth of the rust in a non-linear way across my surface over time. Using this in my simulation could be counterproductive though as it looks like the fractal grows more in veins across a surface rather in the linier patch like behavior of rust and patinas. This source of information is very reliable as it has been sourced from the notorious ACM SIGGRAPH publications. It is also written authors who are all qualified with PhDs.

Amaral L. A. N., Barabasi A.L., Buldyrev S.V. Harrington S.T., Havlin S. 1995 Avalanches and the directed percolation depinning model: Experiments, Simulation and Theory.

In this paper research is presented about the simulating the deposition of avalanches. Again it used the directed percolation depinning model to achieve this. I read this paper as I have been using the directed percolation depinning model a lot in the current implementation of my application and would like to see if there are any better methods of this model. Unfortunately I found that this approach is far too in depth for my needs and understanding and was a hard read. On the positive side though if I ever do look to improve my simulation to this level in the future this paper is very reliable as all the authors are professors at their respective Universities.

Wikipedia, 2013, Perlin Noise, Wikipedia, Available from http://en.wikipedia.org/wiki/Perlin_noise

This article from Wikipedia describes the computer generated visual effect developed by Ken Perlin. It is a procedurally generated texture of gradient noise. This is regularly used in computer graphics to create effects such a smoke, fire and clouds due to its smooth and patchy appearance. This article is mainly just a brief description of the function and describes how the basic function can be developed. This is useful to my project as it can be used to develop the directed percolation depinning model I have implemented earlier. This model allows you to provide a lattice texture that influences the rust so that you can change the growth of the phenomenon to preferred locations. Rather than using a lattice texture of random values between 0 and 1 I could use Perlin noise to create a smoother lattice to encourage rust to grow in smooth patches. This is a much more natural looking effect. Limitations of this article is that the description of the algorithm is fairly vague so I will have to look elsewhere so I can learn how to implement it although it does give several references that I can use to try to find this information. Unfortunately it is not clear how reliable this information is as it is from Wikipedia which is publically available for alteration.

Green, S., 2005, Implementing Improved Perlin Noise, Addison Wesley

This is an article in Nvidia's GPU Gems 2 that implements a fast approximation to this procedural noise generation technique developed by Ken Perlin. This algorithm extends upon the original to create a higher quality looking texture with much smoother noise that has a smaller probability of repeating as often. This article provided much more information about how to compute the Perlin noise compared to the previous article giving very detailed implementation of the algorithms used. Unfortunately this technique is designed for generating the noise effect on the GPU which is not ideal for my project. To use this technique I would need to convert to function to host side which involves also implementing linier interpolation functions. Another issue is that the algorithm is written in Microsoft 3D Effects and CdFX runtimes which I am unfamiliar with so a lot of it could be difficult for me to translate. I believe this information to be very reliable as it has been published by professional graphics card manufacturers' nVidia in their GPU Gems 2 book.

Hugo Elias, Perlin Noise, Hugo Elias. Available from http://freespace.virgin.net/hugo.elias/models/m_perlin.htm

This was an in depth description of what Perlin noise is and how it is formed written by Hugo Elias. It explains that it is the sum of several different noise functions known as octaves. Each octave having a varying in amplitude. After creating your noise function you must smooth that values that are returned with using cubic interpolation between the values of the positions around that pixel. This page as very useful as it also came with a description of the algorithms used to create Perlin noise and how to use them. Unfortunately it is not possible to validate the reliability of the information provided in this article as it is from an unknown source Hugo Elias. The only way I will be able to test these theories is to put them into practice and see if they work.

Worley S. 1996, A cellular texture basis function, ACM SIGGRAPH

This is an article about the noise function also known as 'Worley Noise'. This is a function to compliment the Perlin Noise function that partitions the space into a random array of cells. This has been used to create an effect similar to tiled areas, organic crusty skin, crumpled paper, ice, rock, mountain ranges and craters. This paper is very clear and concise about the creation of this function. The basic idea is that you take random points in space and use the distance of the closest of one of these points as colour information for your current position. This could be useful to my research as I am currently using Perlin noise as a lattice texture and this function could be used to create a larger variety of effect to my current simulation. These effects could include breaking off the surface as it rusts or just trying to replicate the patchy fractural effect that rust grows in. This is a very trust worthy source as it is published by ACM SIGGRAPH. The algorithm is also very reliable to use as it is used in a variety of computer graphics in this day and age.