USAGE BASED MATERIALS BY SIMULATING LAYERED IMPERFECTIONS

A Thesis

by

MELISSA RENEE PARSONS

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Ann McNamara
Committee Members,	Ergun Akleman
	Rodney Hill
Head of Department,	Tim McLaughlin

December 2017

Major Subject: Visualization

ABSTRACT

Maintaining the ability to make quick iterations is very important to any artist in Computer Graphics, which is not always easy for simulating realistic materials based on how they are used. This thesis will examine imperfections in materials and the way different imperfections interact with each other based on how they are used. A new system will be created to save artist time by simulating how imperfections are layered and positioned.

DEDICATION

To my family, friends, and professors that always supported me.

CONTRIBUTORS AND FUNDING SOURCES

This work was supervised by a thesis committee consisting of Professor Ann McNamara and Ergun Akleman of the Department of Visualization and Professor Rodney Hill of the Department of Architecture.

There are no outside funding contributions to acknowledge related to the research and compilation of this document.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Shading in Computer Graphics (CG) refers to the process of adding color, light interactivity, and the overall appearance of an object. Figure 1.1 illustrates how shaders affect the appearance of an object. Even though the flat (a) and plastic shader (b) have the same color attribute (red), the plastic shaded object is more believable because the shader simulates how plastic would react to light in the real world. Very few surfaces in the real world are without imperfections, even brand-new objects have some surface irregularities, seen in figure 1.1(c). CG objects can run the risk of not being believable when surfaces are too perfect. But when artists add imperfections such as scratches or dirt to an object "people can imagine them being part of their real world" (1, p.1). The believability of CG objects depends on the shading process, and the responsibility for simulating real-world materials lies with the shading artist.

When a shading artist is given a scene to work on, their first responsibility is to evaluate the context of that scene. The context of a scene is important because it informs the artist where to apply imperfections. For example, a parking garage stairwell. The stairs are painted cement, each step is structurally the same but have different imperfections based on foot traffic.

(a) (b) (c)

Figure 1.1: Examples of different shaders and a bouncy ball. Examples (a) and (b) is on the same model with the same lighting setup. (a) Flat shader with no light interactivity other than shadowing. (b) LMPlastic shader (Renderman shader) with light interactivity. (c) Reprinted from Körnerbrötchen (Own work) [CC BY-SA 2.0 de (https://creativecommons.org/licenses/by-sa/2.0/de/deed.en)], via Wikimedia Commons. New bouncy ball with a rough surface. The surface is new but not as smooth as the plastic in image (b).

The shading artist is informed that the parking garage is an older structure and does not have an elevator. Even without an elevator the garage is busy, and each level is used daily. With this story context, textures should suggest that each flight of stairs would show heavy foot traffic. The paint on the steps would be worn off and dust would collect in corners. Creating the textures to show these usage effects is important, unfortunately it is also time consuming. Added to this is the possibility that the story concept can change. The director makes an update, the parking garage now has an elevator and isn't busy enough to fill each level daily. With these changes, the stairwell would have lower foot traffic as the levels increase. Those simple changes would mean all the shading from the first scenario no longer match the story! All shaders would need to be redone to match new context. Since the texturing is done by hand, the amount of time spent on texturing this scene would now exceed budget in terms of time and money.

This thesis proposes a method to employ a simple texture map (painted image) to dictate the placement of imperfections. This will save time, effort, and money by allowing automatic edit placement of imperfections within a single texture map. We call this map the usage map. This map describes the areas of a scene in terms of usage. The goal is to save artist time by simulating how imperfections are layered and positioned. To evaluate the proposed method, a simple scene will be shaded then edited for context changes. Images will be generated to illustrate the efficiency of the proposed method.

2. BACKGROUND AND LITERATURE REVIEW

2.1. Imperfections in Materials

Imperfections like scratches, rust, and moss form due to different factors. These factors can be complex and irregular. Their formation patterns however are recognizable and change the overall look of materials. Such changes and imperfections are generally defined as weathering. To simulate these imperfections, it is necessary to understand how they form in real-world materials and weather.

(a) (b) (c)

Figure 2.1: Examples of weathering. (a) Reprinted from Acabashi (Own work) [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)], via Wikimedia Commons. Example of biological weathering. (b) Reprinted from Eric Magnan (Own work) [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)], via Wikimedia Commons. Example of chemical weathering. (c) Reprinted from darwin Bell from San Francisco, USA (paint job Uploaded by Fæ) [CC BY 2.0 (http://creativecommons.org/licenses/by/2.0)], via Wikimedia Commons. Example of mechanical weathering.

Dorsey [2] breaks the weathering of materials into three general categories:

- Biological
- Chemical
- Mechanical

Biological weathering is the process of living organisms growing and forming patterns on surfaces, for example mold or moss, seen in figure 2.1a. Biological growth can lead to mechanical and chemical weathering as well. While plant roots put physical stress on another surface the stress is exerted by the biological growth of the plant. Chemical weathering can be a secondary reaction from the production of "organic acids which help to dissolve minerals" and further weakening the surface (3). Lifecycle changes such as the changing of leave colors are also included as biological weathering. The weakening of materials when living organisms grow on surfaces, for example mold or moss. It can also cause chemical weathering from the production of organic acids which dissolve minerals.

Chemical weathering alters the physical characteristics of the original material, by either forming or destroying minerals, seen in figure 2.1b. This process weakens and causes materials to be more susceptive to weathering through chemical reactions, such as oxidation, hydrolysis, and carbonation. An example is corrosion which "forms a complicated system of reacting layers, consisting of the metal, corrosion products, surface electrolyte, and atmosphere (2, p. 195)." Corrosion happens when rust forms on iron when it is oxidized. The weakening of materials through chemical reactions, such as oxidation, hydrolysis, and carbonation. These processes either form or destroy the minerals, which alters the materials' mineral composition. An example would be rust eroding away metal.

Mechanical or physical weathering weaken materials from physical forces, which break the material down into smaller fragments, seen in figure 2.1c. These physical forces include temperature fluctuation and abrasion. Temperature fluctuations can cause surfaces to either expand or contract creating a physical stress to the surface, which leads to cracks or breaks in the surface [3]. Abrasion from constant use or just from exposure to natural elements slowly breaks down the surface. Common results of temperature fluctuation and abrasions are peeling or cracking paint, scratches, and dents. Some physical forces cause the moving and depositing of other materials, such as dirt or dust. The flow of rainwater "may clean some areas by washing dirt away, while staining other areas by depositing dirt and other substances (2, p.202)." Foot traffic can also deposit dirt or remove dust from a surface. The weakening of materials from physical forces, which break the material down into smaller fragments. These physical forces include temperature fluctuation, moving and depositing of other material, such as dirt or dust, and abrasion.

This thesis will focus on mechanical weathering and its effects on interior dielectrics (substance that is a poor conductor of electricity). That limit is set to reduce the amount of real world external factors that affect weathering, as it is beyond the scope of this thesis. The three forms of imperfections that commonly affect interior dielectrics are dust, dirt, and scratches. These are all examples of mechanical weathering, examples can be seen in Figure 2.2.

(a) (b) (c)

Figure 2.2: Examples of mechanical weathering on interior surfaces. (a) Dust build up on a flat surface that rarely gets used. (b) Dirt build up on a stairstep with frequent use. (c) Scratches on a wood floor with high traffic.

2.1.1. Scratches

Scratches are imperfections that modify the quality of a surface by increasing their facet distribution value, such as cutting small grooves into the surface. This imperfection could range from micro grooves like brushed aluminum or deep grooves like a scratch removing paint on a car. For surfaces that have more than one material layered on top of each other, like cars; scratches could remove the top material to reveal the composite material underneath. The formation of scratches is highly dependent on the shape and use of an object.

For example, your kitchen table would have heavy amount of scratches and form predominantly where people sit the most. Areas with little accessibility of the table would have hardly any scratches. This accessibility theory of placing imperfections was first introduced by Hsu and Wong's tendency distribution model [4]. The tendency model is an adaptation of previous work on simulating dust accumulation [5]. The tendency model can be used to distribute many kinds of imperfections but the model does not account for usage or multiple imperfections at one time. The distribution of imperfections is determined by two core geometric factors, surface exposure and accessibility, and surface curvature. The core factors are then adapted to generate specific imperfections, such as dust, patina, and peeling. The peeling example is like the effect of scratches removing paint off a composite surface. Scratches and peeling factor in surface curvature higher than surface exposure. "This is because the protrusive nature increases the chance of being attacked by external forces (4, p.145)." Geometry dependent texture generation is still commonly used by artists, and in texturing software like Substance Designer and Painter.

Other previous work on scratches focus on rendering a highly realistic model of scratches. The most realistic method was developed by [6] [7]. His method is based on existing BRDFs (bidirectional reflectance distribution functions) combined with texture mapping to shift how a shader handles reflectance in a scratched or unaffected area of the material. Bosch introduces another physically based model improving the previous model "by allowing an accurate modelling of scratches' geometries (8, p.369)." Geometries qualities inside the scratches are simulated by taking scratch formation process into account.

2.1.2. Dust

Dust consists of small amounts of pollen, pet/human hair, skin cells, and many other small particles. Dust accumulates in every environment “the amount of dust accumulation occurs in two phases (5, p.18).” The first phase establishes the amount of dust that would accumulate if there weren’t any external factors, also called “normal” amount. The normal amount is based on the inclination and stickiness of the surface. The second phase is influenced by the external factors including surface exposure and the effect of objects around the surface that scrap off some settled dust.

Substance Designer dust mask generator has some similarities to Hsu and Wong 1995 method for simulating dust accumulation. Surface accessibility is again a major factor in simulating where dust would collect the most and “has less chance of being removed by wind or cleared by scraping (4, p.147).” Substance Designer uses baked texture maps to define the surface exposure and the inclination of the surface. Both methods of simulating dust are effective but do not include usage of the surface as a factor for dust placement. They also do not have an intuitive way of combining other imperfects with dust accumulation.

Guo [9] presents another model for dust accumulation, which improves simulating and rendering dust. A physically based reflectance model for rendering dust is added to Hsu and Wong’s model. Because the placement of dust was not changed, this approach will be used.

2.1.3. Dirt

Dirt and dust form in a similar manner but not always for the same reasons. Dust and dirt form from a lack of use, but dirt has many other ways of forming. One area of focus simulates dirt staining from flowing water. Dorsey [10] introduced a phenomenological system for simulating those distinct patterns of dirt staining. This system uses a particle system, where each particle represents a “drop” of water. The flow of the particles is determined by parameters linked to natural forces like gravity, friction, wind, and roughness. Then the rate of absorption or sedimentation of dirt deposits is determined by a set of differential equations. This method was later improved to “overcome the problem of using either texture synthesis or procedural models alone (11, p.20:3).” Bosch achieved more natural variations by using details from reference images utilized in a phenomenological simulation.

Dirt can also transfer to surfaces around the original or source surface in a very organic manner. A method for simulating this occurrence involves tracing aging particles called γ-tons [12]. Tracing the particles as they hit surfaces determines the distribution of imperfections. This method produces weathering specific to the scene and weathering sources. While there are more recent papers simulating weathering, Chen’s method is the only one with the ability to simulate multiple weathering processes and their interactions.

Even with these highly realistic simulation approaches, surface accessibility is still “useful for modeling processes such as dirtying, cleaning, aging, or polishing (2, p.221).” Substance Designer’s dirt mask generator uses an approach like their dust mask generator. The only difference is the amount of influence the inclination of the surface has on dirt build up.

2.2. Material Creation Methods

A common material generation method is to layer basic materials to create a complex one, using painted masks to blend between the different material attributes. This method has been adopted by many studios and software packages. Naughty Dog, creators of Uncharted 4: A Thief's End, even paired up with Allegorithmic, software designer, to develop a method that best fits the studios needs [13]. The software that came out of Allegorithmic and Naughty Dog's symbiotic relationship was Substance Designer 3 and 4. Instead of using their typical method of manually painting textures, Naughty Dog successfully used an early material layering method. Material layering makes the process much less complicated. While other methods have a lot of complex interactions and connection dependencies which make it harder for iterations.

(a) (b) (c)

Figure 2.3: Example of layered materials in Substance Designer. (a) Wood material. (b) Paint material. (c) Complex material made from layering wood and paint together.

Material layering owes a lot of its success to material libraries and a nondestructive texturing workflow [14]. For example, the complex material seen in figure 2.3c is quickly created from the combination of the wood (figure 2.3a) and paint (figure 2.3b) materials. The look is controlled by a single mask that can either be manually painted in Substance Painter or procedurally generated in Substance Designer. The wood and paint materials came from a collection of modular tileable base materials, called a material library. And the mask is nondestructive because it allows the user to make changes to the materials without overwriting the original material attributes. The material flexibility and nondestructive workflow of Substance is applicable to an efficient imperfection layering method.

3. METHODOLOGY

3.1. Asset Creation

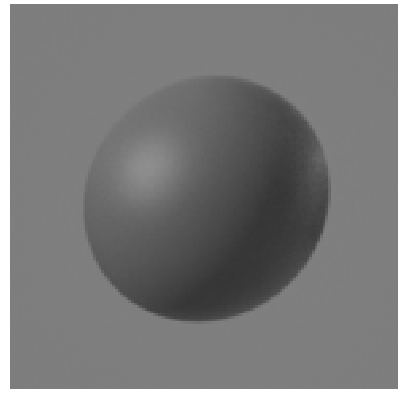
Various CG objects, materials, and light rigs are needed to create and evaluate the tool, all of which are referred to as assets. The CG objects, commonly called models, are created inside Maya. Then the models are exported to Substance Designer and Painter for texturing and material creation. Lastly the textures are brought back into Maya for final rendering. Each asset is lit with the same set up, an environment sphere with a corresponding High Dynamic Range image (HDRI).

Environment lights are a popular rendering technique for physical-based rendering, known as Image-Based Lighting. The HDRI is a texture processed from one or more pictures of a real environment. Then the HDRI is projected onto an infinitely distant sphere to simulate light from that real-world setup [15]. It provides fast, easy setup, highly detailed lighting, shadowing, reflections and refractions; all of which are important when evaluating a material [16]. The light rig ensures that all the materials are going to behave consistently and react correctly in another lighting scenario. The HDRI light rig in Maya is combined with an area light to also evaluate the materials reaction under a CG light.

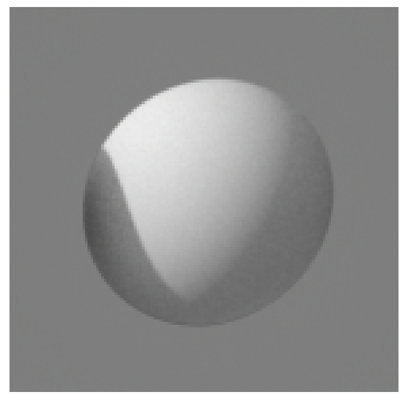
In each of my renders there are four spheres in either the left or right corner, called the diagnostic spheres. Each sphere has a specific job to diagnose the light rig and any problems the user could run into.

The four diagnostic spheres are as followed:

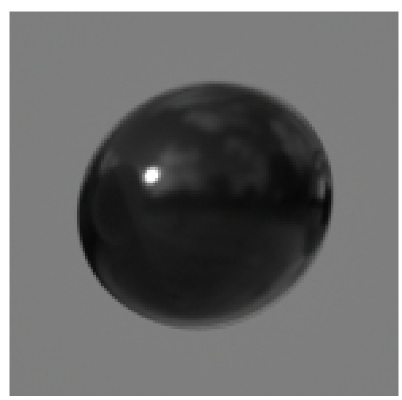
- Matte grey — A material with about 50% roughness and value in diffuse color. It shows how rough dielectrics should react in the light rig. This sphere and the white sphere show the quality of light.



- Diffuse white — A diffuse material with not BRDF calculations. This sphere helps set the exposure of the light rig. It also acts as a guide for other materials, the diffuse attribute should never be brighter than this sphere.



- Shiny black plastic — A glossy dielectric material with black in the diffuse attribute. This shows how dielectrics specular should react, no specular hit should be brighter than this sphere. This sphere also shows the size of the light source.



- Chrome A new chrome material, showing a perfect reflection of the HDRI, any other light positions, and how bright the lights are. This sphere acts as a guide for any metal material, reflections should not be brighter than this sphere.

3.1.1. Example Scenarios and Variety of Usage Maps

The first assets I created were tests objects then other real-world objects to best evaluate the tool's effectiveness during development. The cube shaped test object was modeled similarly to stairs, seen in figure 3.1. The test object required the following characteristics:

- Flat faces- flat objects are boring in both the real world and in CG scenes. It is especially challenging to make a flat object realistic and interesting in CG. Normally a flat object would not be ideal for testing shaders but it is important when testing the imperfection tool.
- Steps as a repeating form- Generally repeating forms would have repeating textures to save time but the goal is to break away from that tradition. So, in this test object the stairs steps are used to test the tool's ability to vary detail in repetitive forms.

- Rim around the top, similar to molding, giving a small number of crevices. When simulating dirt or dust it is important to have areas where it could easily collect. The rim around the top of the test object is modeled to simulate that in the most basic way possible.

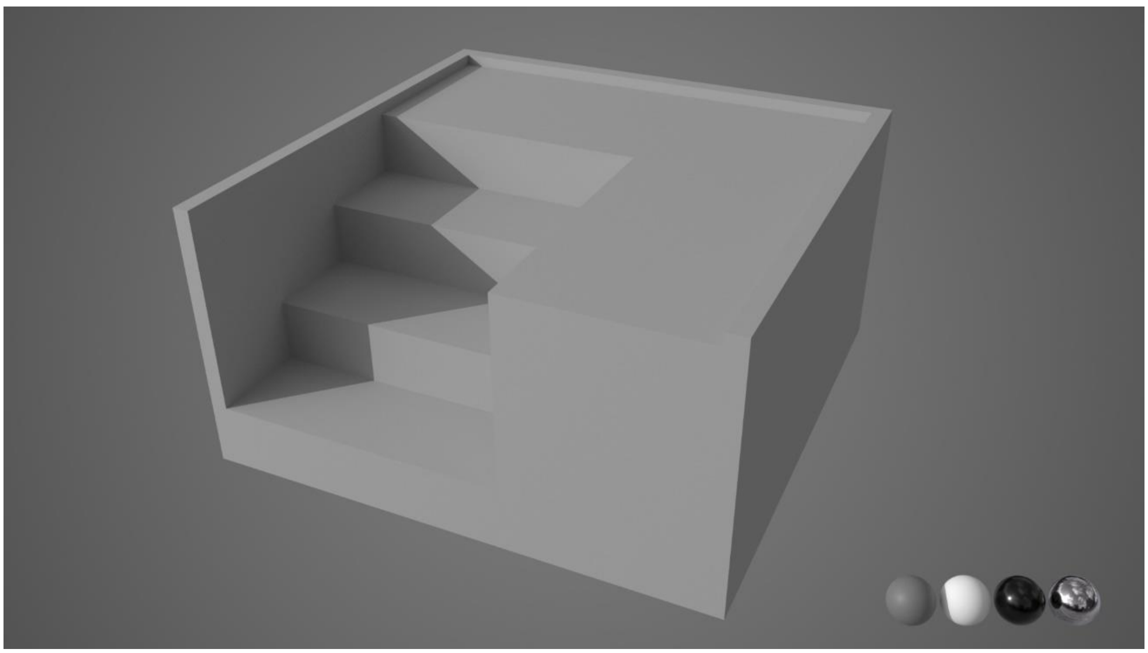


Figure 3.1: Test Cube, rendered in Maya with a diffuse grey material.

The second test object is a sphere modeled with a lot of grooves, crevices, and edges, seen in figure 3.2. The sphere is almost an opposite of the test cube. It is a free model from Substance software, similar shapes are commonly used to develop shaders. Its spherical form is ideal when developing shaders since they give the artist a great sense of reflections and specular response [17].

The sphere was used to evaluate the other end of the spectrum, more detailed geometry with rounded surfaces. These details were used to develop the layered shader aspect of the tool, ensuring the realism of the shaders and their interactions.

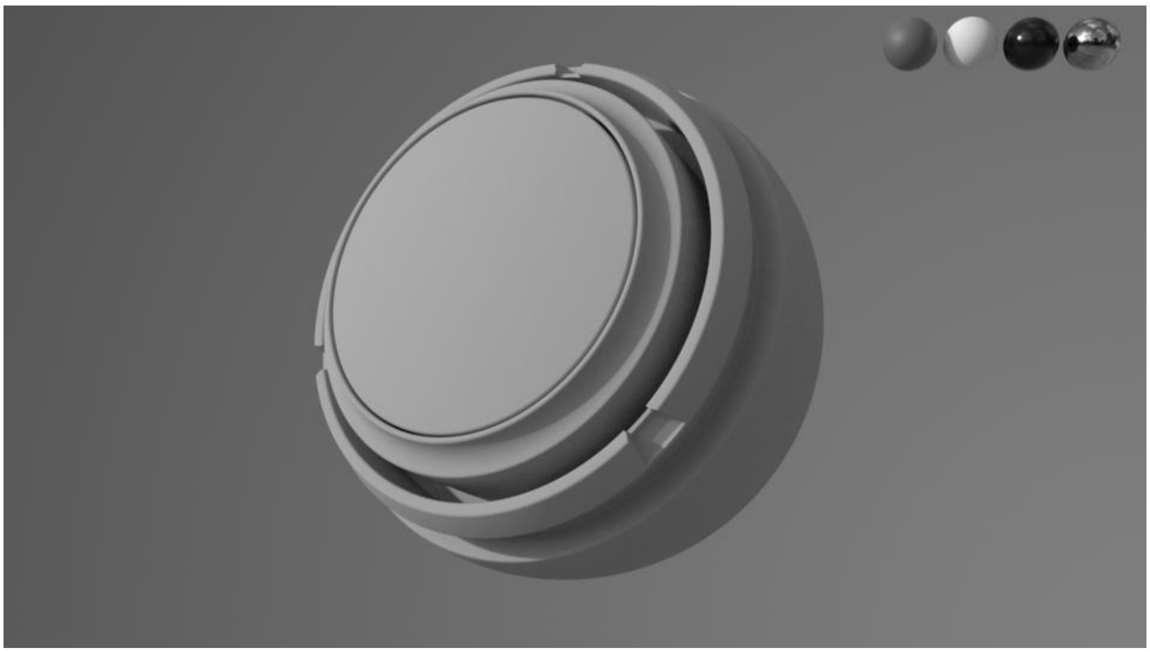


Figure 3.2: Test Sphere, rendered in Maya with a diffuse grey material.

The next 2 objects were used to evaluate the final application of the tool and how it holds up in real world examples. The stair case would have a variety of different used based on its foot traffic as described in the introduction, seen figure 3.3. The second real world example is a kitchen cabinet, seen figure 3.4. It also has repetitive forms that could look different based on how it is used. The cabinet has different type use, the doors that get used the most say a lot about what is inside that cabinet.

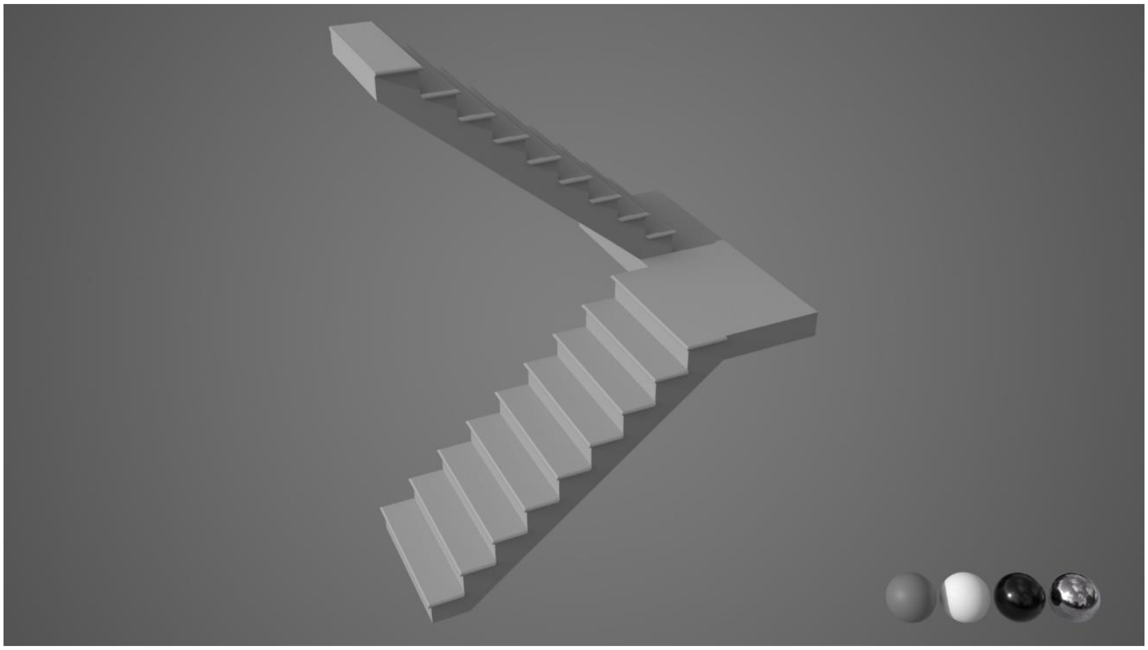


Figure 3.3: Stair case, rendered in Maya with a diffuse grey material.

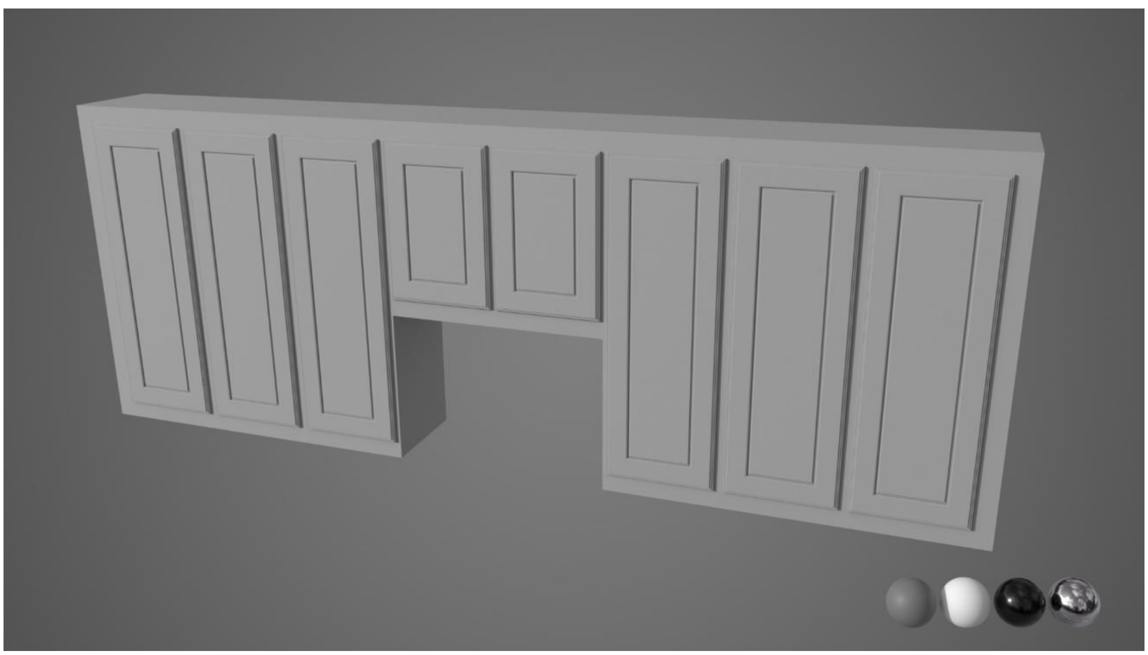
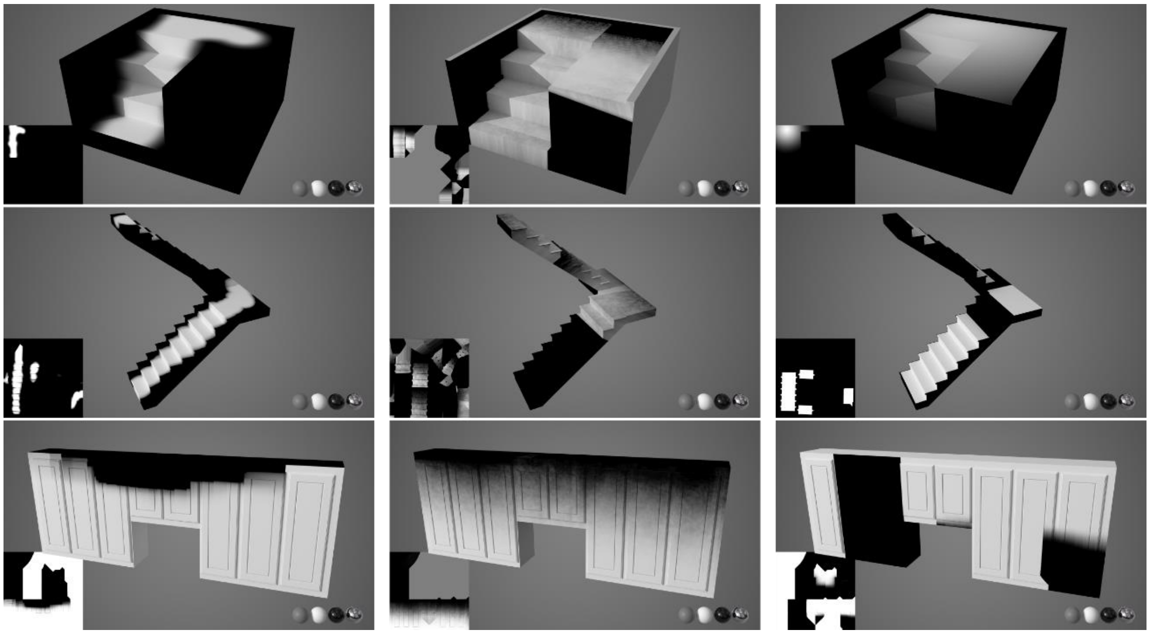


Figure 3.4: Cabinets, rendered in Maya with a diffuse grey material.



(a) (b) (c)

Figure 3.5: Usage maps on each object. (a) Each object with typical usage maps. (b) Each object with texturized usage maps. (c) Each object with random usage maps.

A variety of usage maps were created for each object. The usage maps vary by detailed to understand how refined they need to be for the tool to work. The first usage map (see figure 3.5a) were painted as a typical usage for that object. The second set of maps (see figure 3.5b) are also typical usage but a different paint brush was used in Painter to add natural patterning around the white fall off. This is done to see if the user could get better results by painting more detail in their usage map. The last set of usage maps (see figure 3.5c) were painted to test how little detail was needing to achieve an interesting look. This set was not done with typical usage but more of random usage.

3.1.2. Base Materials

Materials were model with Substance Designer procedurally, then recreated in Maya and rendered with Renderman [18]. The texture maps generated in Designer were used in Maya. Below are the side by side comparisons on a sphere.

Substance Designer uses physically based rendering (PBR) as their method of shading, which "provides a more accurate representation of how light interacts with surfaces (19, p.3)." PBR is used by most CG rendering software, including Renderman. This similarity makes it easy to transfer assets between Substance and Maya. The only difference comes from the way Substance handles their shader creation and workflow. The two main workflows are Metal Roughness and Specular Glossiness. Both workflows produce similar results but they each have their pros and cons. Metal Roughness provides a simple way of masking between a metal and a non-metal. With this workflow only 4 texture maps are needed, base color, normal, roughness, and metallic. The metallic map is the simplified way of differentiating between two different materials, metals and non-metals. Even though this workflow doesn't allow a lot of control for index of refraction (IOR) or Fresnel reflectance at 0 degrees (F0), it is still the most preferred workflow [19]. When the textures are used for final rendering inside Maya and Renderman, control over IOR and F0 is reintroduced.

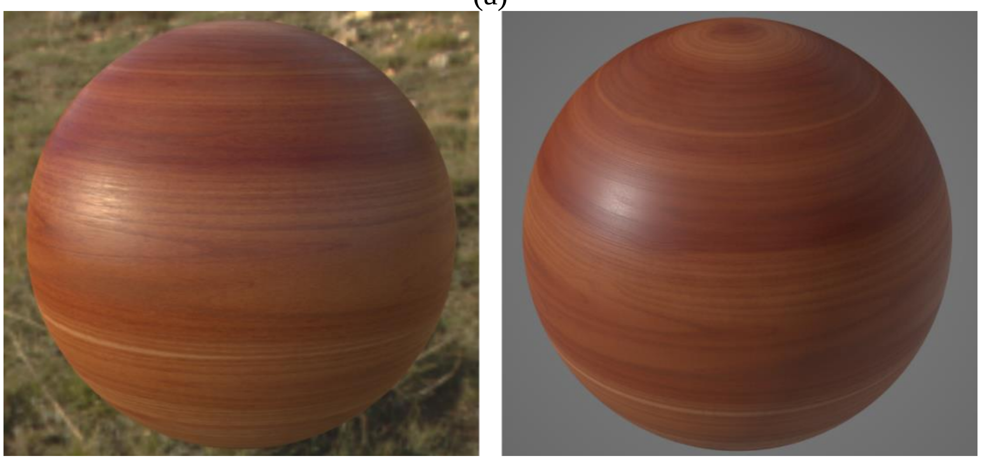
Once the textures are complete in Substance Painter they are then exported to be used inside Maya. Since there is not yet a simple way of exporting the shader from Substance to Maya, the shaders are recreated to match the original. During the material creation process, in both Substance and Maya, the parameters for the materials are achieved through the following consideration through texture maps [20]:

- Base color, diffuse, or albedo can be derived from researched data or artistically created to represent the reflected color for dielectric. Substance uses this colored data for the reflectance values as well.
- Specular values such as IOR and F0 are derived from previously researched data that represent the reflectance properties of a material.
- Roughness is completely an artistically controlled parameter, since there is no right or wrong. It represents how smooth or rough a material could look.
- Normal or bump maps simulate smaller surface details such as the small pits in the bouncy ball in figure 1.1. This parameter is also artist driven.

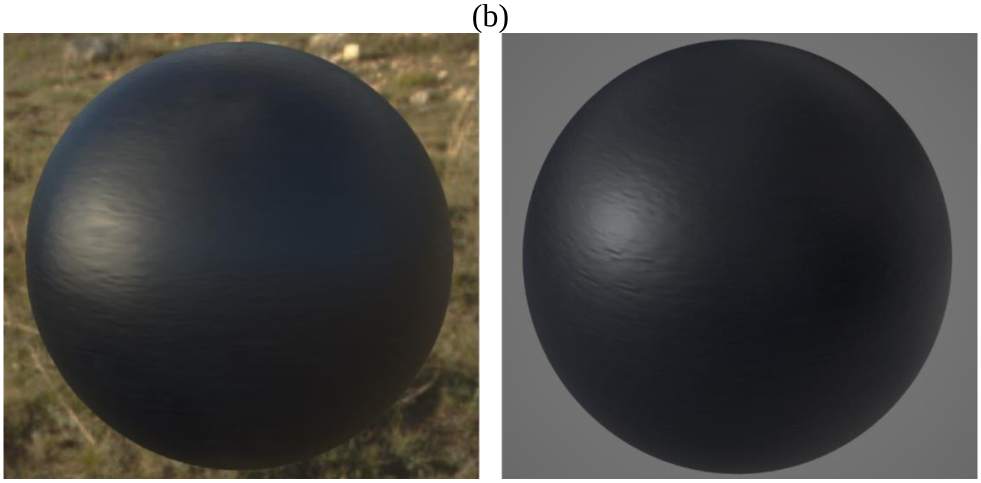
Three base materials were created to test the tool and give it even more context. Painted surfaces were the focus of this thesis. Paint, dirt, cement, and wood were created to be layered later as a complex material. Textures from Substance were exported and used with a similar shader inside Maya.

Figure 3.6a is the side by side comparison of cement base material, created in both Substance and Maya. The cement textures were adapted from a Substance Designer base shader. Inside Substance I added a tileable attribute, so the material could be scaled and used on most CG objects. The normal map was also updated to add more detail.

(a)



(b)



(c)

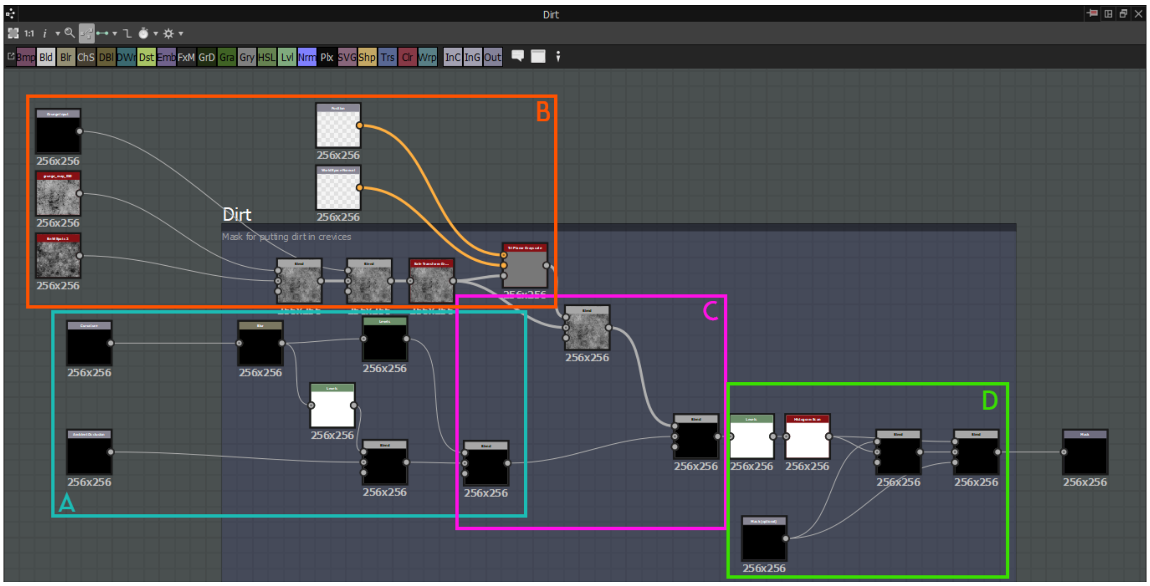
Figure 3.6: Base materials. Left image is the Substance Designer result, right is the Maya Renderman result. (a) Cement material. (b) Wood material. (c) Paint material.

Figure 3.6b is the side by side comparison of wood base material, created in both Substance and Maya. The wood textures were adapted from a substance designer base shader. Inside Substance I added a tileable attribute, so the material could be scaled and used on most CG objects. The height and ambient occlusion maps were disregarded for the shaders overall look but are later used in the tool.

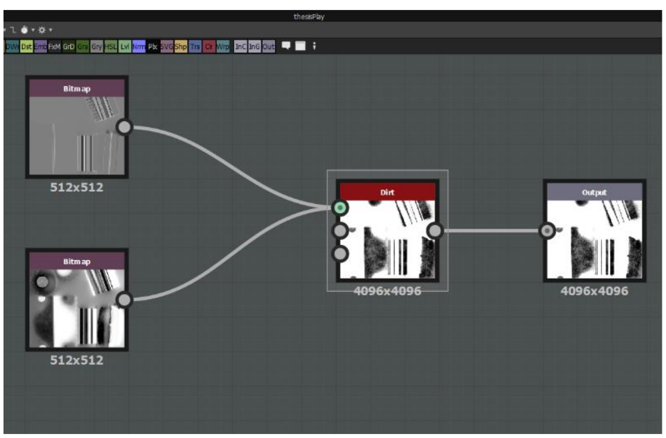
Paint material was created procedurally inside Substance Designer, seen in Figure 3.6c. Also has a tileable attribute. When shown by itself, the normal map is modeled after a basic sheet rock surface. Inside tool this normal map is discarded for the underlining composite material, since paint typically takes the surface quality of what it is covering.

3.2. Imperfection Masks

3.2.1. Dirt Filter



(a)



(b) (c)

Figure 3.7: Substance Designer mask generator called Dirt. (a) Node graph that contained inside the Dirt node. (b) Node graph users would see when using the Dirt node. (c) Mask output from the node graph in b.

First reviewed Substance Designer's Dirt mask generator which uses surface dependency method. Surface accessibility is commonly calculated through ambient occlusion. The Dirt node outputs a mask to layer another material on top of the original material.

Figure 3.7a shows the node graph that is embedded in the Dirt node. The filter requires at least 2 baked textures, curvature and ambient occlusion (Figure 3.7b). The nodes inside box A use the objects ambient occlusion(AO) baked texture to isolate areas of the object that are more accessible, such as cracks and other crevices. This section also subtracts the curvature map from the AO map, because dirt does not usually collect on the edges. The user can control the amount of edge masking affects the overall dirt mask.

To create a natural dirt, break up, a dirt map (painted or procedural) is multiplied on top of the adjusted AO mask. This process is done by the nodes inside box C. Here the user is given control over grunge or dirt amount.

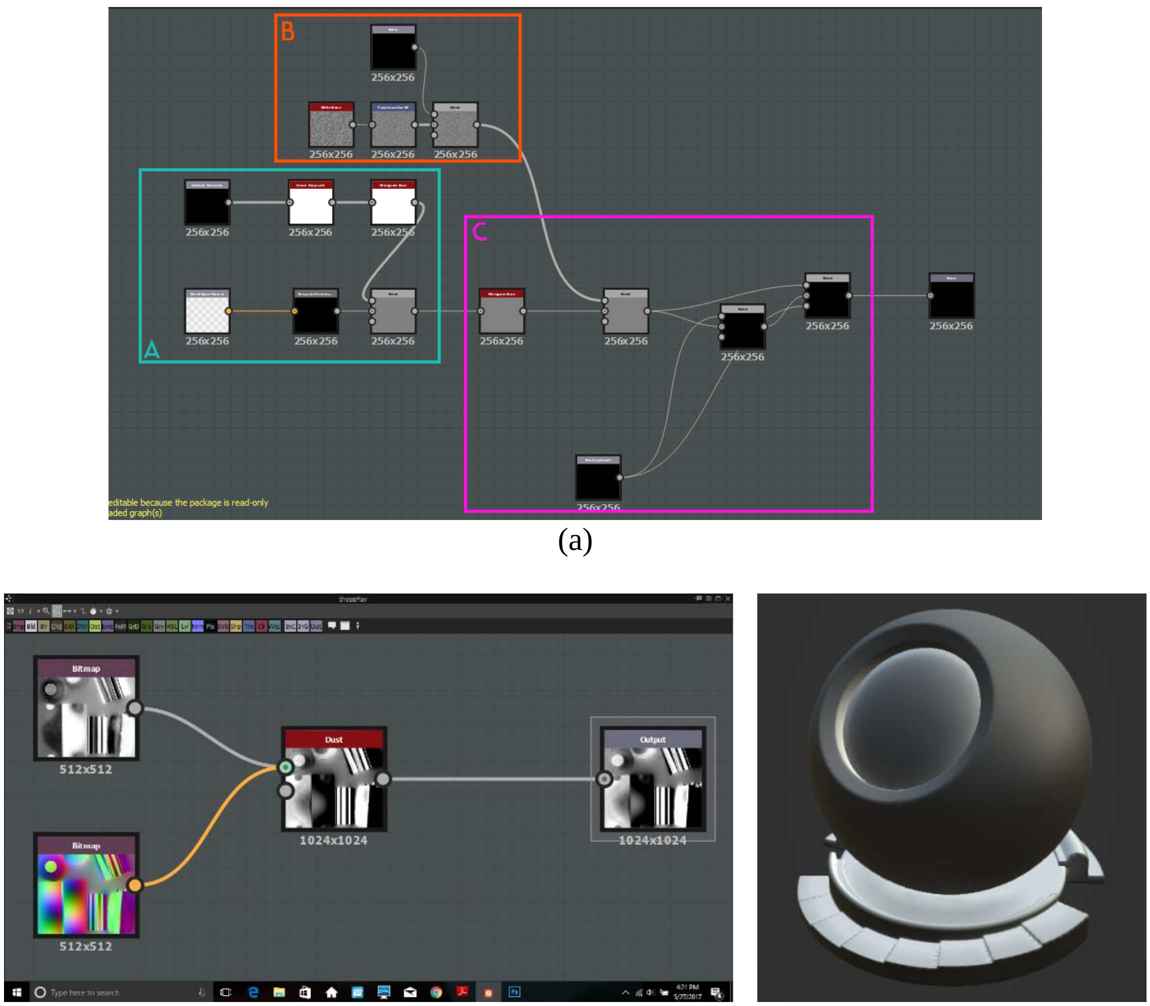
Boxes B and D are nodes meant to give the user more controls. Box B allows the user to choose between the provided procedural dirt map or a custom dirt map. While box D allows the user to control the dirt level, the mask's overall contrast, and potentially use a mask to remove dirt effects from certain areas. The result is seen on a test object in Figure 3.7c, default settings are used.

This node works great but is limited by the following things:

- Locked into that dirt pattern, patterns can look noticeably repeated.
- Doesn't handle surfaces face normals accurately.

- Doesn’t consider the slickness of the material it is covering.
- The mask between other imperfections would need to be carefully painted. Grey values in the mask might look strange and only create faded looking dirt.

3.2.2. Dust Filter



(a)

(b) (c)

Figure 3.8: Substance Designer mask generator called Dust. (a) Node graph that contained inside the Dust node. (b) Node graph users would see when using the Dust node. (c) Mask output from the node graph in b.

Substance Designer Dust mask generator has some similarities to Hsu and Wong 1995 method for simulating dust accumulation [5]. The Dust node outputs a mask to layer a dust effect on top of the original material.

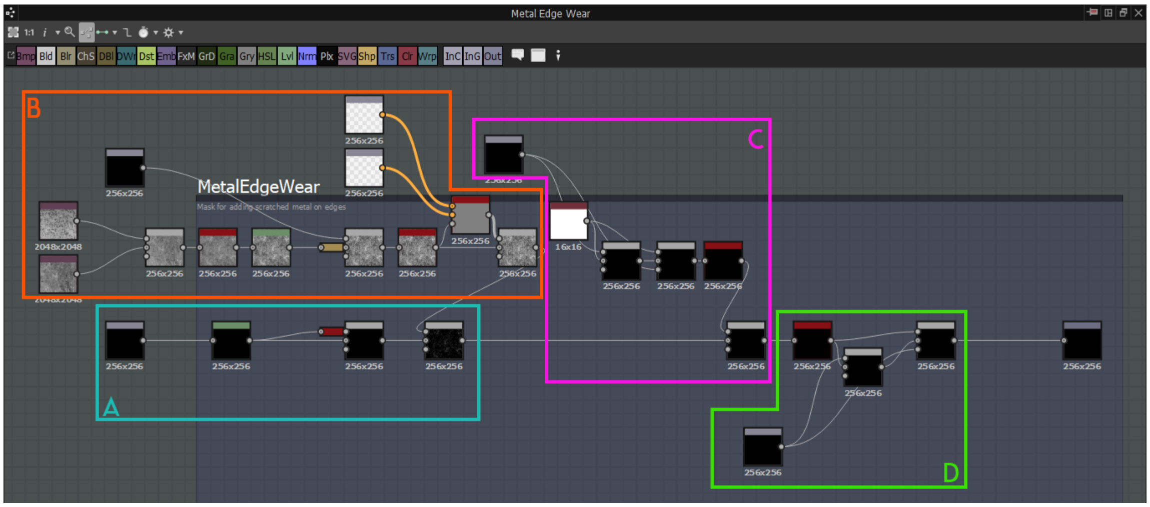
Surface accessibility is again a major factor in simulating where dust would collect the most and "has less chance of being removed by wind or cleared by scraping. (5, p.19)" Box A does this with ambient occlusion and world space normal baked textures. Ambient occlusion isolates where the dust would accumulate and stick the most. While world space normal focuses on inclination of the surface. Surfaces with a steeper inclination would not hold dust very well, due to gravity and wind.

Boxes B and C are nodes meant to give the user more controls. The nodes in box B allows the user to have noise in the dust and control the overall effects of the noise. While box C allows the user to control the dust level, the mask's overall contrast, and potentially use a mask to remove dust effects from certain areas. The result is seen on a test object in Figure 3.8c, default settings are used.

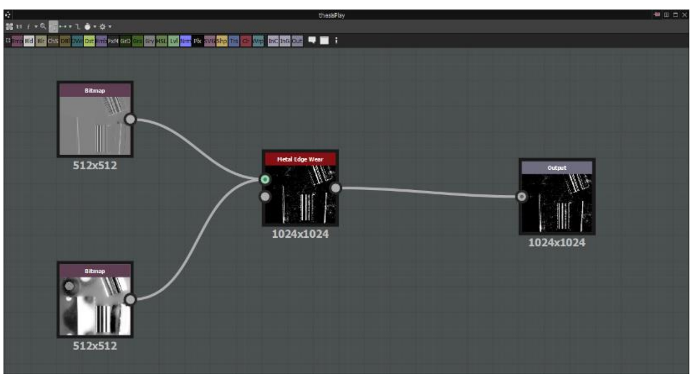
This node works great but is limited by the following things:

- Dust is applied evenly everywhere.
- Doesn't handle surfaces face normals accurately.
- Doesn't consider the slickness of the material it is covering
- The mask between other imperfections would need to be carefully painted. Grey values in the mask might look strange and only create faded looking dust. Is not close to the scraping effects done in Hsu and Wong method [5].

3.2.3. Edge Wear Filter



(a)



(b) (c)

Figure 3.9: Substance Designer mask generator called Metal Edge Wear. (a) Node graph that contained inside the Metal Edge Wear node. (b) Node graph users would see when using the Metal Edge Wear node. (c) Mask output from the node graph in b.

While there are many mask generators that focus weathering on edges, Metal Edge Wear creates the best edge mask. It is commonly used to mask where metal would be exposed on a painted metal surface. But could also be used to mask where impacts, scratch, and peeling. All of which usually occur on edges or protruding areas of an object, then spread out from there. “This is because the protrusive nature increases the chance of being attacked by external forces (4, p.145).” The nodes in box A isolates edges and softens the edge mask fall off by using a curvature baked texture.

Then to create a natural break up in the mask a grunge texture map is multiplied on top of the edge mask. A grunge texture is provided for the user or a custom texture could be used. This is all achieved in box B.

Ambient occlusion is also factored in this mask generator, shown in box C. It is used to remove areas of the object that are not as exposed, like crevices, do not receive a lot of impact weathering. The user has control over how much effect ambient occlusion has on the mask.

The last section, box D, are nodes meant to give the user more control. While box C allows the user to control the wear level, the mask’s overall contrast, and potentially use a mask to remove wear effects from certain areas. The result is seen on a test object in Figure 3.9c, default settings are used.

This node works great but is limited by the following things:

- The grunge texture doesn’t allow for irregular pattern.
- Doesn’t handle surfaces with face normal 90 degrees very well.
- The wear level does not fully simulate the natural spread of wear on objects.
- The mask between other imperfections would need to be carefully painted. Grey values in the mask might look strange and only create faded looking wear.

3.2.4. Connection of Imperfections

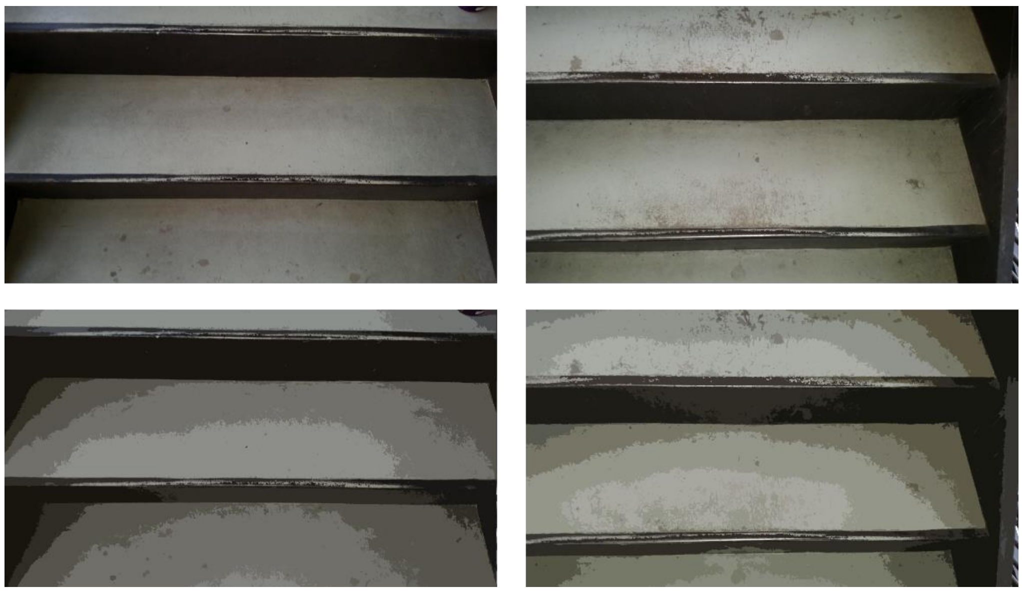
Reference images were taken at a parking garage as described in the introduction. The stairwell is indoors and made of painted cement. The garage has an elevator but the first 2 or 3 flights of stairs look very much used. The amount of imperfections varied based on the floor, Figure 3.10 and 3.15 are examples of what that might look like.

(a)

(b)

Figure 3.10: Reference images from parking garage. (a) The original. (b) Photoshop processed image.

To break apart the different layers of imperfections, I ran the reference image through a Photoshop filter called Cutout. The Cutout filter added high-contrast to the image, making it much easier to see the fall off for each imperfection. Figure 3.10b is the Cutout filter version of 3.14a.



(a) (b)

Figure 3.11: Reference images from parking garage. The top image is the original and bottom is the Photoshop processed image.

The scratches or worn off paint stay tight to the high amount of use (Figure 3.11b). The underlying material greatly effects the amount and shape of the paint peeling off. The pattern is very organic and changes shape as it gets to the edges, getting smaller and more separated.

The dirt or dust build up can be broken up into 3 levels, based on its amount and color (Figure 3.11a). Dirt and dust don't show up on top of each other. It is almost like there is one or the other. Dirt effects the paint differently as it falls off. The original clean paint is never fully seen, because the first amount of dirt has stained the area. The next level, going toward being less and less used, is a little color and is barely visible. At that level the dirt is mostly sitting on top the paint, hasn't stain the paint yet and could be easily removed. The least amount of use dirt area is much darker and thicker. It is almost just sitting on top. Those levels could either be places with dirt or with dust.

From this info I knew that the scratched away paint had to be the first part of the layering process. First model in that effect then that mask would inform the dirt falloff. I also noticed that the imperfections effected the surface differently based on the surfaces face normal. Such as the walls won't be able to hold the dirt in place because of its steep incline, but the walls of the steps did see a little bit of paint peeling off and a little bit of dirt collecting in the corners.

3.3. Filter Framework

These are the things I need to have in the filter for it to be successful.

1. Simple parameters. The parameters need to be easy to understand what they effect. And can't have complex combinations between parameters to achieve the desired look. Only have the parameter that they need.
2. Easy usage map setup, that can be painted in Substance Painter. The workflow should be simple and easy to understand as well.
3. Min and max sliders. A min and max slider would help the artist get the amount of imperfections desired. The imperfections will be modeled at the high and low extreme so the artist has plenty of room to choose from.
4. Some material options. At this point only 2 materials will be available, painted wood or painted cement. More options could be created in future work.

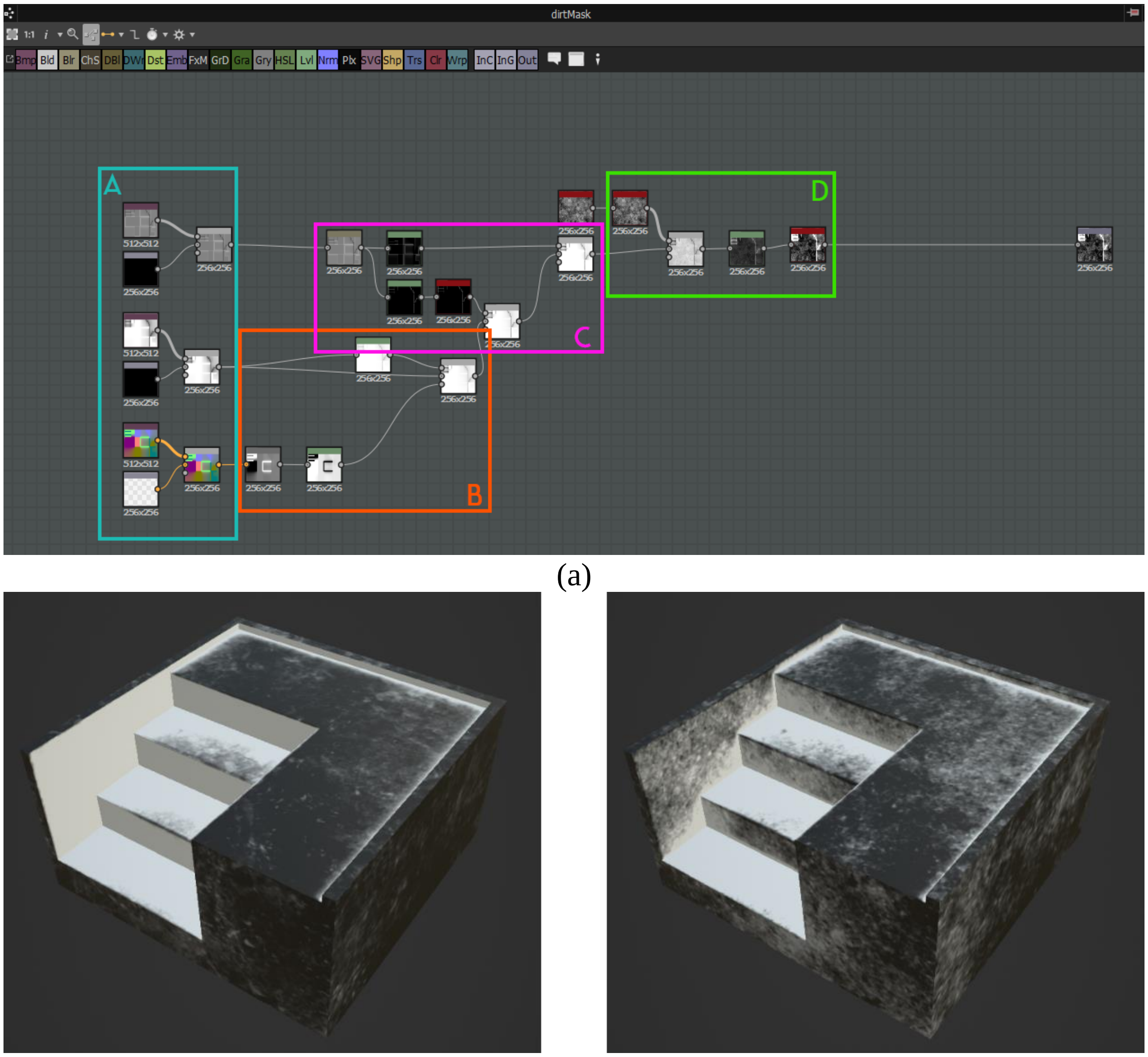
4. IMPLEMENTATION

The filter creation is in two parts. This is from the Substance texture creation process. It is setup in a way that Designer works alongside Painter. Filters are improved and nested to keep the workflow clean and simple.

4.1. Filter Creation in Substance Designer

The filter for applying dirt and scratches needed to be corrected before setting up the final tool. Figure 4.1 shows the node graph of my custom dirt filter. The filter requires 3 baked texture, curvature, ambient occlusion, and world space normals. The nodes inside box A are all the baked textures inputs that will be supplied by the artist in Substance Painter.

The nodes inside box B correct the issue with surface normals pointing in the z axis. With the use of a world space normal baked texture the problem faces can be isolated with the blue channel of the texture map, since the blue channel represents the z axis. This was a problem for the original dirt filter because it broke reality by allowing the dirt to collect on a face with that steep of an incline. My solution was to create a mask of these faces and brighten the ambient occlusion texture in those areas. Because AO is the driving factor in where the dirt is places only that map needs to be adjusted.



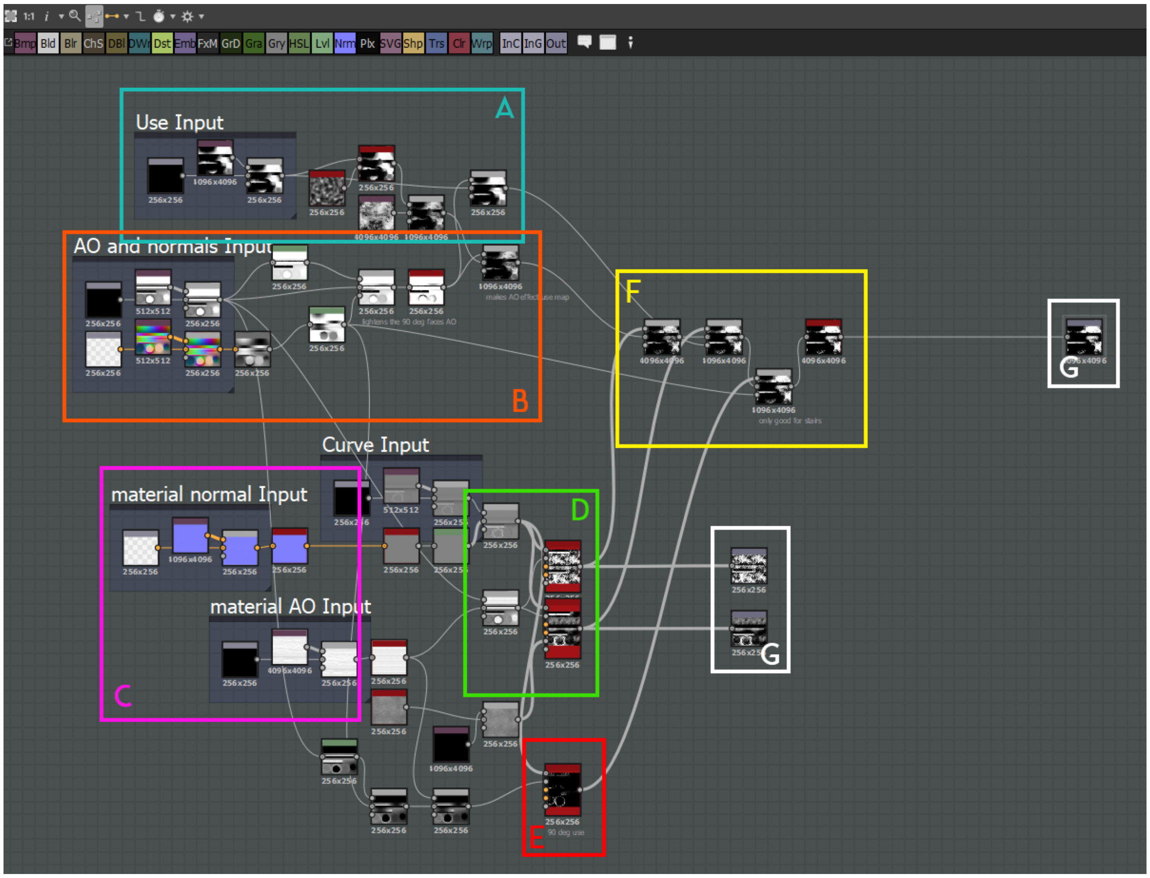
(a)

(b) (c)

Figure 4.1 Custom dirt filter node graph and output. (a) Node graph. (b) Output from the original dirt filter. (c) Output from the custom dirt filter.

The nodes inside box C are unchanged from the original dirt filter. It combines the adjusted AO with the curvature to make the base of the dirt mask. Then in box D a new grunge map is multiplied on the base dirt mask for natural break up in values. The updated grunge map is less noticeable when repeated. A simplified number of parameters are also applied in box D nodes. Only the grunge scale is fully exposed to the artist in the final form of the tool. The ones exposed here are only to be used in the main section of the tool.

As you can see in the result images the mask output from the custom dirt filter is much more realistic and natural than the original. Both images have the same default values in the parameters.



(a)

(b) (c) (d)

Figure 4.2 Custom scratch filter node graph and output. (a) Node graph. (b) High usage output. (c) Medium usage output. (d) Final output custom scratch filter.

Next the scratch filter was improved and applied the effects of the usage map. Figure 4.2 shows the node graph of my custom scratch filter and the 3 outputs from that filter. Box A is the usage map input and a few other nodes for breaking up the shapes. A slope blur and grunge map are applied to the use map to break up the shape and give the map a more natural shape. Two different variations are multiplied with the AO and use later to mask the high and low usage.

Box B corrects the issue with surface normals pointing in the z axis, same as the correction for the dirt mask filter. Scratches do still appear in this area but have a slightly different look to them. The AO is still adjusted for the regular application within the filter, fading the effect of scratches in that area. And the world space normal map is used in a parameter specific to stairs.

Box C takes inputs from the material underneath the paint to add detail and realism to the way the paint scratches away. Here the normal map input and AO input from selected material is added or multiplied to the objects curvature and AO baked texture.

Box D takes the results of box C then feeds it into the metal edge wear node. Two different nodes are used because depending on the level of use the mask output would change. The wear level, wear contrast, grunge pattern, and grunge scale differ. For the high use, the standard grunge pattern is used and is scaled lager than the low use. The contrast is also higher to maintain sharp edges in the patterns. For the low use, a custom grunge map is used, the scale, wear level, and contrast are lower. This is done to have a natural fall off for the scratches that follow the grain of the selected under material.

Box E nodes are a special parameter just for stairs. A switch in the parameters allows the user to turn this off and on. The switch is needed for when faces point in the z axis, for example the kitchen cabinets. It has a different AO input so it only effects the desired areas and similar parameter settings to the low use metal edge wear filter. The pattern is different from the low use filter.

Box F is where the low and high use filters are masked by the usage map. For the high use the adjusted usage map, one output from box A, is blend with an overlay operation which is a combination of multiply and screen. Values below 0.5 of the bottom layer are multiplied. While values higher above 0.5 of the bottom layer are screened with the top layer. Next the low use is added to the high use with a blend node with the second usage map output from box A. If the stairs switch is on then the stairs use filter is added in the desired areas. Finally, a histogram scan node is applied to bump up the contrast on the resulting mask. Contrast in this mask tends to look incorrect.
Because this isn’t the final filter for the tool more than one output is necessary. Three outputs are needed, seen in box G. The high and low use filter results are used in the main filter to help the dirt application.

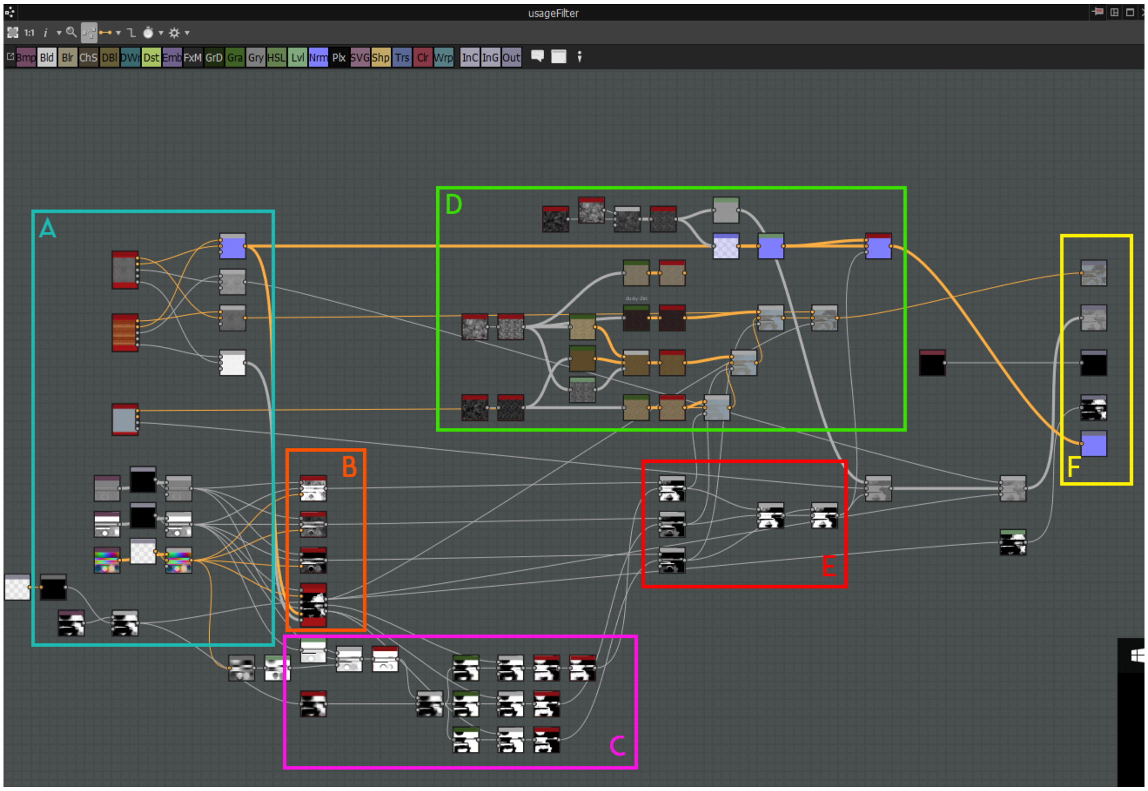


Figure 4.3 Main node graph for the tool.

The final step in the tool creation is shown in the node graph in figure 4.3. Box A has the four inputs needed from Substance Painter, curvature, AO, world space normals, and usage map. It is also where the base materials reside, they are hard coded inside the tool not an input from Substance Painter.

Box B has three custom dirt filter nodes, described above, and the custom scratch filter. The 4 inputs from Substance Painter are connected. The usage map was already applied to the custom scratch filter, but the custom dirt filter still requires some attention. Like the scratch filter different levels of use will have different parameter in the mask filter. From the high use to the low use the level, contrast, and amount decreases. The dirt pattern stays the same.

Box C is where the usage map is adjusted to isolate the 3 different levels of use for the dirt filter. Values in the usage map are reassigned to high, medium, or low usage. Values of the usage map between 0 – 0.46 are considered high usage. Values of the usage map between 0 – 0.24 are considered medium usage. And values of the usage map between 0 – 0.08 are considered low usage. To break up the 3 masks the high and low results from the scratch filter are blended with an overlay operation.

Box E takes the 3 masks then multiplies them with the results of the dirt filter. These results are then used to mask the dirt material on top the paint material.

Box D is where all the materials are layered together. The dirt is layered in 3 different colors and amount. The reference seemed to show this shift in color and amount based on how long it had been sitting on top of the paint material. Once the dirt is applied the exposed under surface is layered on top. This material is selected by the user inside Substance Painter.

Box F has the resulting layered material that is feed back into Substance Painter. The artist has some parameters to customize their overall look:

- Material and grunge scale, in one switch
- Switch for 2 materials
- Stairs switch
- Min and max usage

4.2. Filter Use in Substance Painter

Once the tool is exported out of Substance Designer it is ready to be used inside Substance Painter. Inside Painter the artist can paint the usage map, apply the tool, edit parameters, and paint more details on top of the tools final results. The tool is very simple to use and allows for easy iterations, as this is the goal of the tool.

Figure 4.4 is the interface of the tool inside Substance Designer. The steps for applying a custom tool inside Painter are the same as any other filter application after the artist imports the tool in their file [21]. After the tool is imported into the file the artist only needs to bake the 3 texture maps the tool requires. Then apply the tool to the top most layer, in the layers tab seen in box C of figure 4.4. The tool and baked texture map can be seen in box B of figure 4.4. The tool effects every layer underneath the tool. Once the tool is applied the artist will see the results in 3D viewer, box A. The artists can see the results on a material or by looking at each texture separately, for example figure 4.4 is showing the result in the diffuse/base color channel. In the properties tab, box D, the artist would edit parameters and connect baked texture maps.

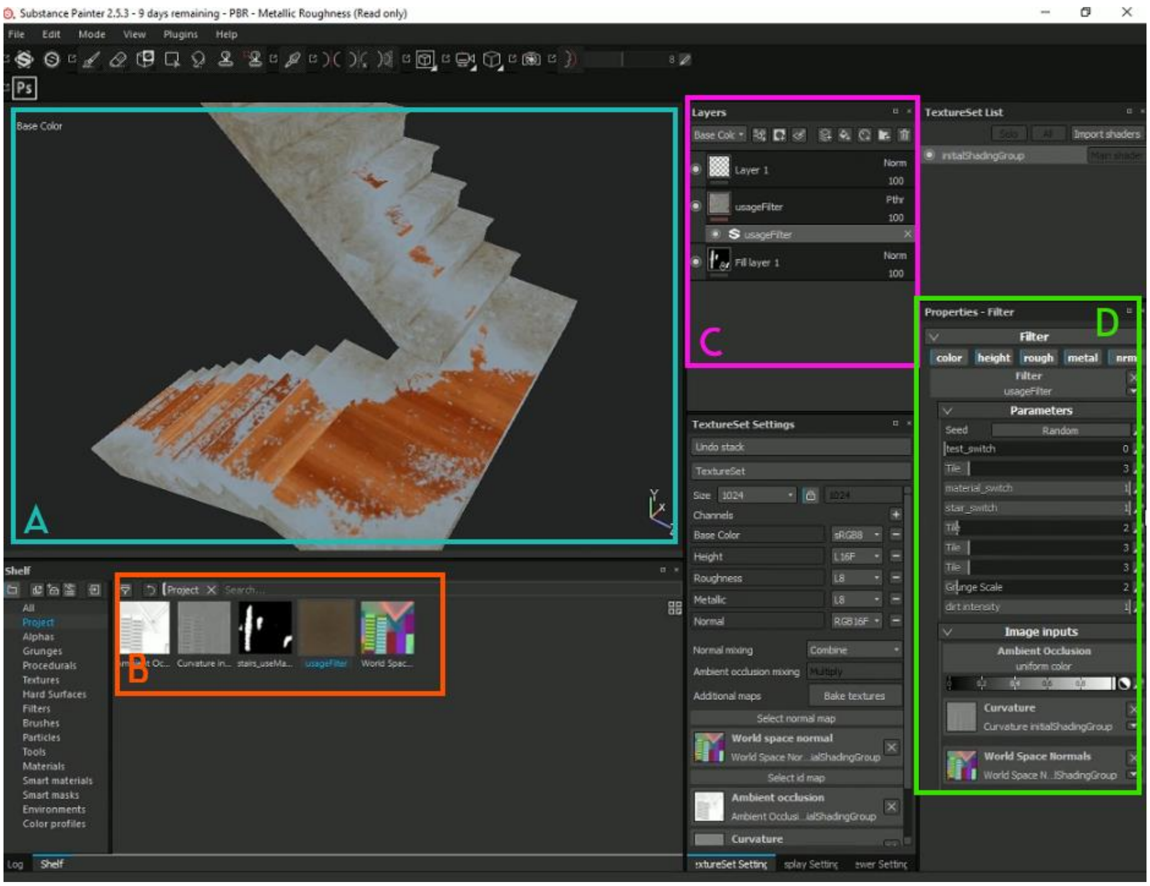
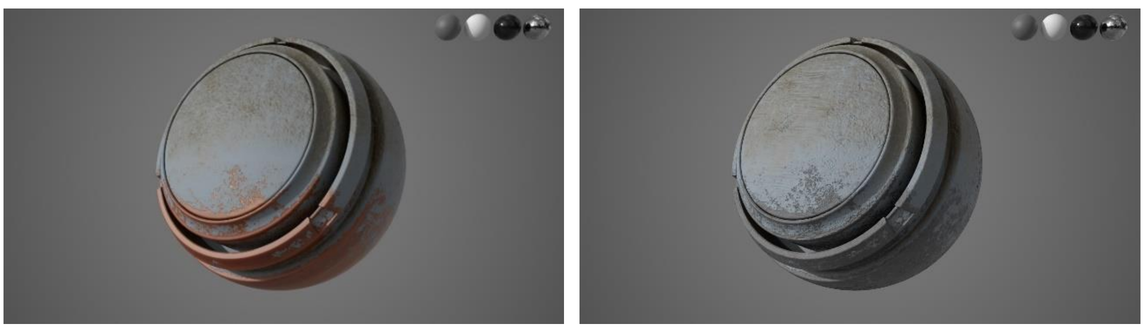


Figure 4.4: Tools interface inside Substance Painter.

The artists can update the usage map even after the tool has been applied if it done below the layer the usage tool is applied to. Any changes made to the tool's parameter or usage map can be see live, so the artists won't have to wait for results. These materials and textures are meant to be a very detailed base for the final look, so if the user wants to they can paint more details on top of the tools results. These qualities in the tool and Substance Painter are the main reason the tool achieves its goal in saving artists time.

4.3. Results

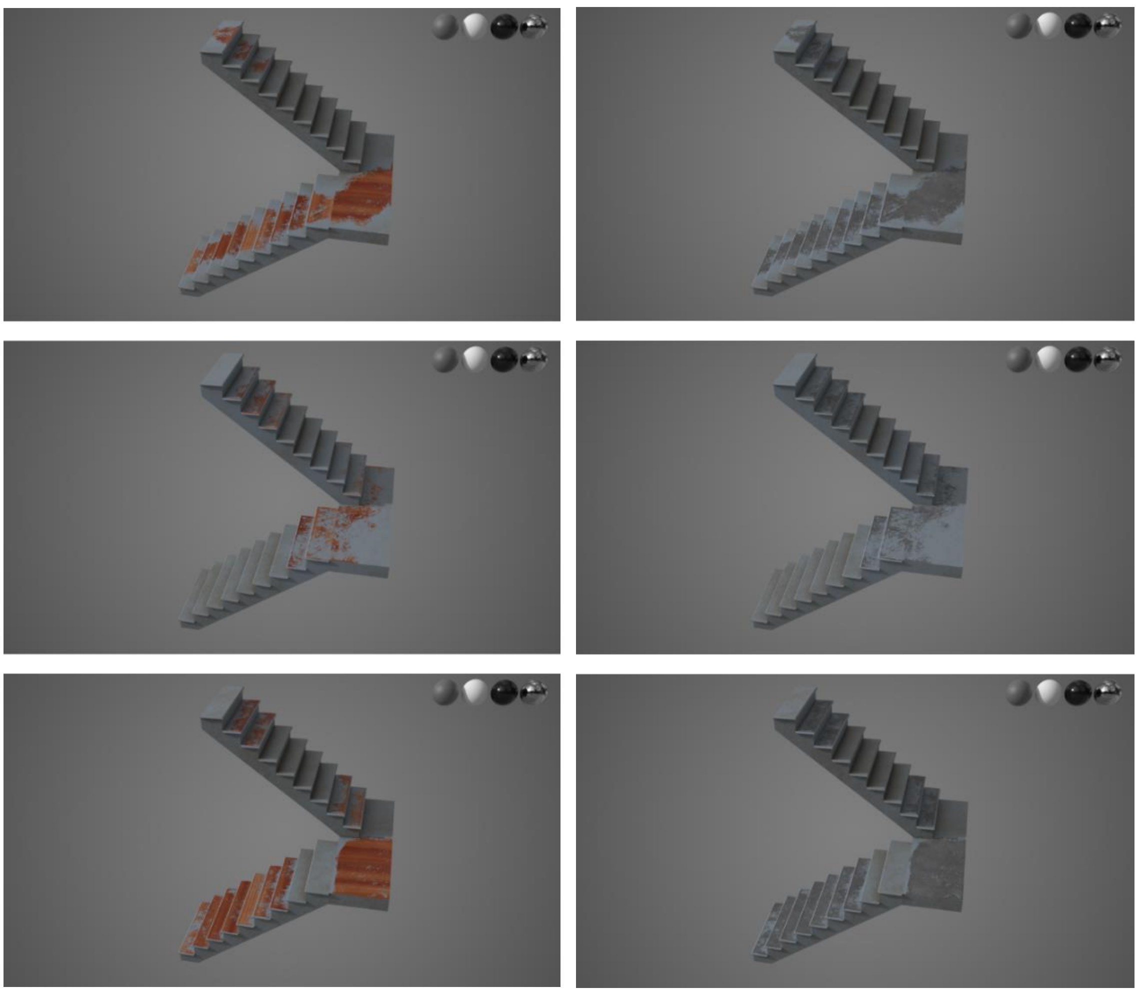
After some adjustments inside Substance Painter, texture maps were again exported to be applied in Maya and rendered with Renderman. The same approach was used when applying the exported textures in Maya as the base materials. The first result was rendered on the test sphere, seen in figure 4.5. This was done to again evaluate the shaders, textures, and layering of imperfections. While the test sphere doesn't have much usage context, it does show the materials and layers well.



(a) (b)

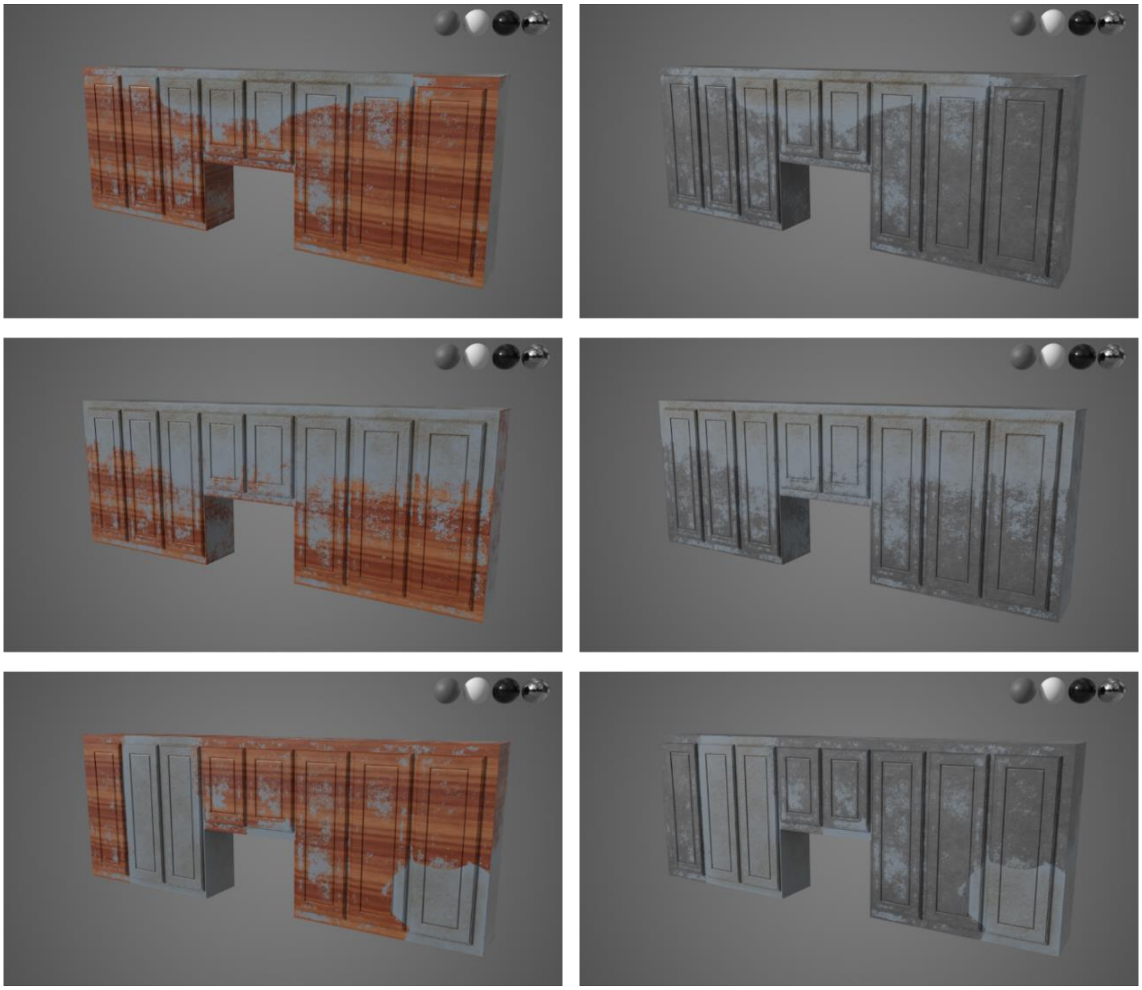
Figure 4.5: The wood and cement results on the test sphere. (a) Wood base material result. (b) Cement base material result.

In figure 4.6 each usage map and material are represented on the staircase model. The best results can be seen on the cement staircase, the right column. That material is more successful because the base material repeats with less obvious pattern repetition. While the wood material loses its believability the more it is repeated. The wood also has an issue with the orientation of the wood grain. A solution to this issue could be found in future work. Even with these issues the different usage maps give a very different feel to the look of the stairs. The 3rd usage map was painted in a random manner, so the result doesn't fully give a believable story but again has a unique break up in imperfections.



(a) (b)

Figure 4.6: The wood and cement results on the staircase object. Rendered with each usage map. (a) Wood base material result. (b) Cement base material result.



(a) (b)

Figure 4.7: The wood and cement results on the cabinet object. Rendered with each usage map. (a) Wood base material result. (b) Cement base material result.

Unfortunately, the cabinets were not as successful look. Figure 4.7 shows each usage map and material for the cabinet model. Because of the previously described wood issues the look of wood on the cabinets distracts you from the success of the imperfection layering. The texturized usage map result, middle row, is the only one that doesn’t take attention away from the tools actual goal. A cement cabinet isn’t fully plausible, but its example shows the imperfection layering without distracting textures.

5. CONCLUSIONS AND FUTURE WORK

A tool for layering imperfection based on how an object is used was created using Substance Designer and Painter. The tool utilizes a texture map, called usage map, to dictate the placement of imperfections. The usage map describes the areas of a scene in terms of usage. The tool saves artists time, effort, and money by allowing automatic edit placement of imperfections within a single texture map.

The thesis goal of saving artist time by simulating how imperfections are layered and positioned was achieved. On average an artist might spend 2-3 days shading a prop asset, or longer if there are any changes. But with my tool that task could be finished in a day or less, depending on how detailed it needs to be. But for the tool to be completely helpful on a big project it would need refinement. The issues with the texture patterns and so few materials would only make it useful in some cases. This tool has opened a lot of other possibilities for future work.

To add versatility to the tool, more materials would need to be added to the library. The expanded library could even include materials outside the dielectrics. Ideally the artist would be able to create their own base materials in Substance Painter. Then Substance Designer would take those inputs to properly place imperfects based on how those materials react to certain imperfections.

More variation in imperfection patterns would also help the finally results of the tool. Unfortunately, Substance Designer does not yet allow the user to change a pattern based on the values in another texture. There are 2 nodes that do part of what is necessary, but they are still not able to work together. This functionality is like a remap node inside Maya. Once this is possible, variation in the imperfections patterns could be done with ease.

More detail and precision could be added to the way the usage map is painted with a simulation. Like Dorsey's flow simulation driving the pattern of dirt, a foot traffic simulation could be used to describe a more exact usage map [10].

Other applications for the tool could also lead to interesting future work. Wang [22] and Bellini [23] present methods for texture generation with time being a factor. Time could also be an added factor to this tool with an animated level of usage. It could be useful in showing the limits of real-world objects and allow artists to create an animation of how this object would age over time. As of now Substance Designer and Painter are not capable of exporting textures in a sequence. Other CG software might be capable of adding time to the current model.

REFERENCES

[1] Schwärzler, M. (2007). Rendering Imperfections: Dust, Scratches, Aging,.. *Institute of Computer Graphics and Algorithms, Vienna University of Technology, Tech.Rep.TR-186-2-07, 9*

[2] Dorsey, J., Rushmeier, H., & Sillion, F. (2010). *Digital modeling of material appearance* Morgan Kaufmann.

[3] Mamo, M., Kettler, T. & McCallister, D. (2017). Soil Genesis and Development, Lesson 2 - Processes of Weathering. Retrieved from http://passel.unl.edu/pages/informationmodule.php?idinformationmodule=1124303183&topicorder=5&maxto=7&minto=1

[4] Wong, T., Ng, W., & Heng, P. (1997). A geometry dependent texture generation framework for simulating surface imperfections. *Rendering Techniques' 97* (pp. 139-150) Springer.

[5] Hsu, S., & Wong, T. (1995). Simulating dust accumulation. *IEEE Computer Graphics and Applications, 15*(1), 18-22.

[6] Mérillou, S., Dischler, J., & Ghazanfarpour, D. (2001). Surface scratches: measuring, modeling and rendering. *The Visual Computer, 17*(1), 30-45.

[7] Mérillou, S., & Ghazanfarpour, D. (2008). A survey of aging and weathering phenomena in computer graphics. *Computers & Graphics, 32*(2), 159-174.

[8] Bosch, C., Pueyo, X., Mérillou, S., & Ghazanfarpour, D. (2004). A Physically-Based Model for Rendering Realistic Scratches. Paper presented at the *Computer Graphics Forum, , 23*(3) 361-370.

[9] Guo, J., & Pan, J. (2014). Real-time simulating and rendering of layered dust. *The Visual Computer*, *30*(6-8), 797-807.

[10] Dorsey, J., Pedersen, H. K., & Hanrahan, P. (2005). Flow and changes in appearance. Paper presented at the ACM SIGGRAPH 2005 Courses, 3.

[11] Bosch, C., Laffont, P., Rushmeier, H., Dorsey, J., & Drettakis, G. (2011). Image-guided weathering: A new approach applied to flow phenomena. ACM Transactions on Graphics, 30(3), -.

[12] Chen, Y., Xia, L., Wong, T., Tong, X., Bao, H., Guo, B., & Shum, H. (2005). Visual simulation of weathering by γ-ton tracing. *ACM Transactions on Graphics (TOG), 24*(3), 1127-1133.

[13] Nelva, G. (2016, 03/16). Amazing Uncharted 4 Pictures Show Environments and More; Naughty Dog Talks Texture Design at GDC. Retrieved from http://www.dualshockers.com/amazing-uncharted-4-pictures-show-environments-and-more-naughty-dog-talks-texture-design-at-gdc/

[14] Allegorithmic (Producer), & . (2016). *Substance Designer Getting Started: 02 - Procedural Material Creation Workflows.* [Video/DVD] https://www.youtube.com/watch?v=ClbKIHtUT5Q:

[15] Autodesk. (2014). Retrieved from http://help.autodesk.com/view/MAYAUL/2015/ENU/?url=http://help.autodesk.com/cloudhelp/2015/ENU/Maya-Tech-Docs/Nodes/mentalrayIblShape.html

[16] 3Delight. (2016). Image Based Lighting. Retrieved from https://3delight.atlassian.net/wiki/spaces/3DFM/pages/16252931/Image+Based+Lighting

[17] Walker, Megan (2015). Realistic Aging of Materials in Computer Graphics. Master's thesis, Texas A & M University.

[18] Allegorithmic.Substance Designer User Guide. Retrieved from https://docs.allegorithmic.com/documentation/display/SD5/Substance+Designer+User+Guide

[19] Wirrmann, N., Noguer, J., & Bagard, A.The Comprehensive PBR Guide by Allegorithmic-vol. 2. *Light and Matter: Practical Guidelines for Creating PBR Textures,*

[20] Cyrille, D., & Wirrmann, N.The Comprehensive PBR Guide by Allegorithmic-vol. 1. *Light and Matter: The Theory of Physically-Based Rendering and Shading,*

[21] Allegorithmic.Substance Painter. Retrieved from https://docs.allegorithmic.com/documentation/display/SPDOC/Substance+Painter

[22] Wang, J., Tong, X., Lin, S., Pan, M., Wang, C., Bao, H., . . . Shum, H. (2006). Appearance manifolds for modeling time-variant appearance of materials. *ACM Transactions on Graphics (TOG), 25*(3), 754-761.

[23] Bellini, R., Kleiman, Y., & Cohen-Or, D. (2016). Time-varying weathering in texture space. *ACM Transactions on Graphics (TOG), 35*(4), 141.