Contents

1.1 Introduction

The biggest complaint about computer generated scenes are that they look too perfect and clean. Weathering modeling introduces blemishes such as dirt, rust, cracks and scratches to virtual scenery. This adds a whole now realism to the scene making it more acceptable to the audience.

This Thesis will concentrate more on the weathering of paint as to the build up of dirt and grime. The final effect created can be used as a post production effect on both computer generated and live action movies. In order to create this model it is first necessary to look at why paint and how paint weathers.

1.2 Causes of Paint Failure

Fig 1 Paint weathered surface

Excessive moisture on paint surfaces plays a big role in paint failure. Water gets soaked into the walls via joins or through existing fractures in the paint. This causes blisters (swelling) to form, the bigger the blister the more likely it will leave a scar. In worst case scenarios it will lead to the cracking and peeling of the paint layer. Blister size also depends on the thickness of the paint layer, the thicker the paint layer the smaller the blister. Until the layer is to rigid to form a blister and simply just cracks. This can happen in both indoor

and outdoor painted surfaces. The average human family of four can produce three gallons of water per day through respiration alone, not to mention all the home appliances like dishwashers and cookers contributing their own vapor to help damage paint surfaces. Exterior painted walls can also be damaged by having poor vapor insulation as vapor is able to diffuse through walls during cold weather and condenses in to liquid form. This can then directly attack the underside of exterior paint causing blistering. This phenomenon can be seen around laundry's, bathrooms and any other places of high humidity.

The only way to stop blistering occurring is to eliminate the moisture. This can be done by finding the cause (plumbing problems, leaky taps) and fixing them. Providing good ventilation to areas of high humidity (bathrooms, kitchens) will also help extend the life of paint surfaces

Intercoat peeling occurs when the bond between paint layers becomes too weak. This can be caused by the progression of time eroding the bond, moisture weakening the adhesive bond or the poor preparation of the old paint layer.

2.1 Previous work

2.2 Weathering

One of the first approach to weather modeling was explored by Becket and Badler in 1990 [1]. Their technique described a general system treating scratches, stains, splotches, and rust . Since then multiple methods have been developed including the weathering due to flow of liquids. Such methods are typically based on involved physical simulations that cost highly in the way of computation. However they are capable of producing results that are highly realistic. Other methods include the development of sophisticated

shaders to reproduce the effects of weathering. Merillou [2] defines a technique that states an anisotropic reflectance function (BRDF) for a given surface to simulate the look of scratches.

When dealing with paint weathering, the simulation of the formation and propogation of cracks is very important.. O'Brien and Hodgins [3] uses a subdivided tetrahedral mesh for cracking. This uses more complete simulation as it is physically based. The model developed emphasizes on the dynamics and the movement of the cracks produced, creating impressive realistic images, at a high computational cost.

Hirota [4] also uses a physically based simulation to do this. The model developed bases the propagation of cracks on the contraction of a surface. It uses a network of springs and a crack is birthed as and when a spring breaks. The cracks are constrained to this network of springs

Many other methods of weathering virtual objects consist of hand painted textures pasted on to an object. This is can be very time consuming and may require many artists to complete the job. In their paper "Context Aware Textures" Dorsey and Hanrahan introduce a technique to automatically simulate a number of aging effects. The problem with this approach is that a new model has to be developed for each specific effect. Also detailed knowledge of the actual weathering process is needed. Sometimes this information is not available.

Another method of creating weathered looking objects is to use a technique called appearance mani-Folds[5]. This modeling technique uses the time-variant surface appearance of a particular material from data captured at a single instant in time. It is a method that does not aim to be physically correct but aesthetically pleasing.

2.3 Shattering

The realistic shattering of glass, the breaking of walls creates a far more interesting scene for the audience to view. This technique has been in use for over a decade and the technology is only improving. With the production of more advanced hardware, this has led to the acceleration in the quality of graphics, including effects such as environments breaking apart.
In their 1999 Siggraph paper, O'Brien and Hodgins outline a method for calculating the procedure involved in simulating a brittle fracture. This is a physically accurate simulation that revolves around the concept that “fractures arise in materials due to internal stresses created as the material deforms.” While this technique produces great results, it's computation times are great. Using this method produces great results, but is extremely computationally expensive. Therefore, when it is simulated in computer graphics, the primary objective is usually optimization and aesthetics, so the effect of the external factors are faked as opposed to calculated.

The concept of post production effect in the movies is not a new one; however, since the introduction of digital effects in films such as Tron (1982), film-makers have been turning to digital post-production more often in order to realize their visions. One of these is the effect of crumbling of walls. There are times in a film's production where the actor/s would be put at risk if they were to be around breaking glass or crumbling walls. This is more commonly done nowadays in post-production.

Most industry standard animation packages such as Maya (Rigid Body Dynamics) and Houdini (DOP’s), they do not come with an inbuilt shattering/cracking tool. However there are certain pieces of software and plug-ins that specialize in the destruction of geometric objects. One of the market leaders is a piece of software caller 'Blastcode'. Created by Blastcode

Inc., the software operates as a plug-in for Maya. The product has testimonials from artists in companies such as Weta Digital.

For this Masters project a realistic simulation of paint weathering over time is to be produced. The Model produced will be not be physically based but will mimic physical observations by using forces from bellow the paint surface to create blisters on the paint surface until it cracks and starts peeling. The actual paint flakes are pre defined using Voronoi noise. The propagation of cracks should also follow a pre defined path. Controls for the colouration of paint as it weathers will also be added as will the flake height. The Tool created should be full capable of weathering the surfaces of objects. An animation will also be produced to demonstrate this.

3.1 Technical Background

The Paint weathering model will be created using Houdini. Houdini is a high end 3D animation package, created by Side Effects Software [7]. It has been used in Feature films such as X-Men 3, the Ant Bully and Happy Feet.

In order to pre define paint flakes, a method similar to the one used by Pixar in their 2006 film “Cars” could be used. Pixar Animation Studios required an innovative method of breaking up the geometry in one of their key scenes; where McQueen destroys a road surface. This method is outlined in their 2006 paper “The Wrecked Road in Cars –
or How to Damage Perfectly Good Geometry”. In it they describe an intuitive procedure that uses Voronoi noise [fig 02]. Voronoi noise creates cell-like patterns by scattering points using Poisson distribution

The procedure is essentially 4 steps:

1) - Use a curve for the path of the crack to emit a small quantity of particles.
2) - At every particle create a Voronoi tile in the rad geometry to represent the slabs of tarmac
3) - Fix a particle to every vertex of each slab produced.
4) - Use dynamic simulations with falloff to move the particles in a dynamic fashion

Pixar state that they did not use inter-slab collisions as there were in excess of 23,000 tiles. Instead, they manually cleaned up the larger
and more visible inter-penetrations.

As previously stated, the propagation for cracks is very important. In order to make this realistic, L-Systems can be used to define a path for cracks.

L-systems are a mathematical formalism proposed by the biologist Aristid Lindenmayer in 1968 as a foundation for an axiomatic theory of biological development. [8]. L systems are basically a set of rules used to model the growth of plant like structures. They achieve this well due to its recursive nature e.g. a tree can be defined as a branch which creates more branches

An L-system can be defined as

G = {V, S, ω , P} [1]

V states the number of replaceable variables used within the system. These variables are usually written as letters (a, b, c)
S states symbols that represent constants (locked variables) which cannot be changed
ω defines the start of the structure (axiom) and is made up of variables from V
P contains a set of rules that determine which variables are replaced (predecessor) and what they are replaced with (successor).

To make life easier Houdini has its own L-System generator at SOP level which can be easily integrated in to the network.

4.1 Solution

4.2 Shattering A Surface

In order create procedural weathered paint, it is first necessary to shatter the proposed surfaces first. Houdini has no inbuilt shattering or cracking tool, so it is necessary to build one from scratch. This could be done using curves, but the amount of flakes created will heavily depend on the number of curves used. A more suitable method is to generate flakes, is to use Voronoi noise. Using an online tutorial [9] which uses a similar approach to Pixar's solution used in cars (see above)and expanding on it.

The approach takes in input geometry (grid, sphere, OBJ) and adds Voronoi noise to it to produce the flakes. Each flake is then assigned a unique colour so they can split up and treated as individual flakes for further processing.

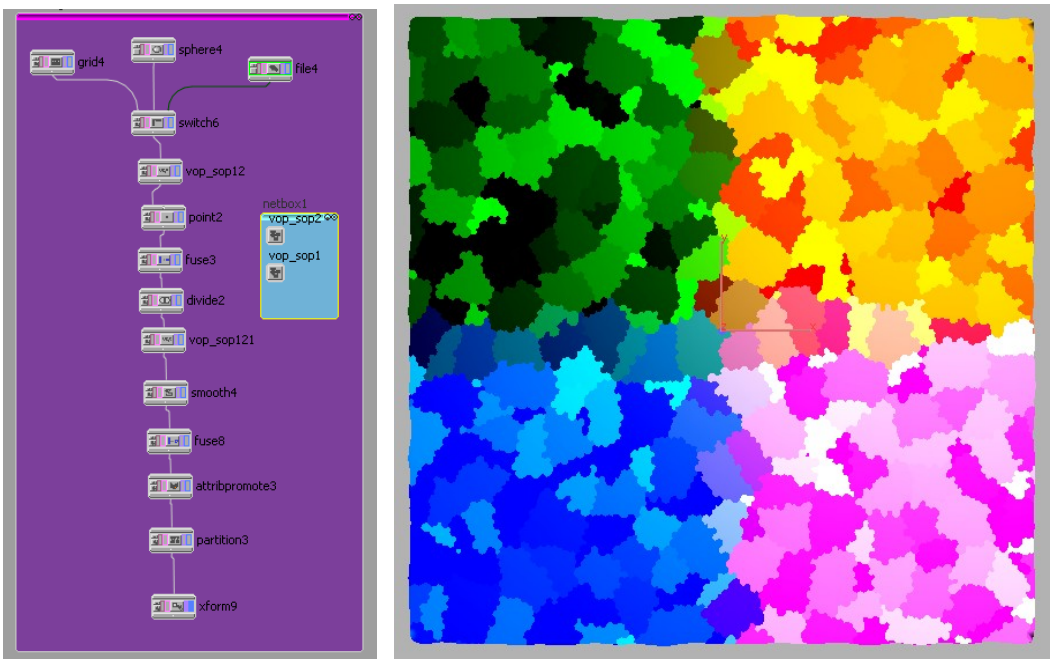

Fig 03 shows paint flakes created using Voronoi Noise.

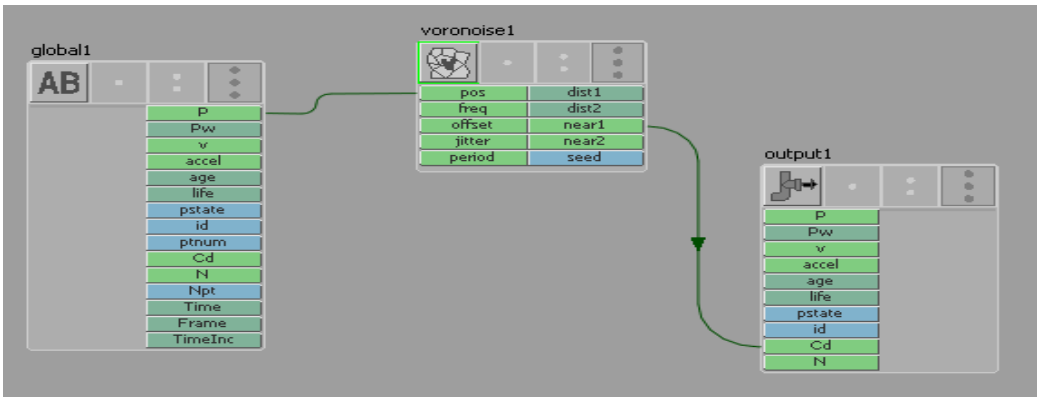

Fig 04 shows the VEX network created go give each flake a unique colour

4.3 Blistering

To create realistic looking blisters forming on our surface, an expanding force from below the painted surface must be created, mimicking real life. A closer

examination on the behavior between paint, the following observations were made:

1. Over time blisters tend to grow larger
2. When two blisters, they join and become one
3. Blisters and cracks tend to follow a path

Two of these behaviors can be simulated with in computer animation by using meatballs.

Metaballs are simply spherical force fields which are capable of fusing with other meatballs depending on their proximity and the density of their respective fields. The joining of two or more meatballs changes their shape(fig?) and increases the density of the force field.

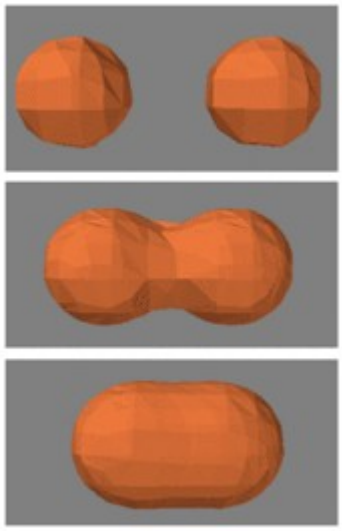

Fig 05 shows the interactions between two meatballs [10]
A metaball can then be attached to a simple particle system to regulate its birthing, while its field density is dependent on time. The scale of each metaball is varied using the stamp function. By doing this each flake will have a different height when undergoing weathering opposed to the same height

To actually deform the paint surface in Houdini, a force must be applied to the meatball model. Houdini has a few inbuilt force SOPS, one that closely matches the requirements need is the Magnet SOP.

The Magnet SOP lets you affect deformations of the input geometry with another object using a "magnetic field" of influence, this is usually defined by a metaball field. It is able to create and animate bumps and dents within objects.

The actual deformation comes from the translates of the Magnet SOP, and not from the metaball itself. The metaball defines the area of effect for the Translates of the Magnet SOP. The weight of the metaball determines the effectiveness of the Translates within the magnet SOP.

The power of the magnet is greatest at the centre of a metaball field and diminishes to nothing at the edge of the field. Which is useful when simulating this kind of blistering effect.

The weathering model uses the magnet SOP and copies it to the metaball using a Copy SOP. The Magnet SOP uses the density field of the Metaball as a Magnetic Deformation Field (MDF) to deform the cracked surface. The Metaballs are then launched onto the proposed painted surface with the moving particle system. As the MDF passes through the surface it causes the surface to bubble upwards emulating the blistering of paint.

To fulfill condition three, a path must be predefined for the metaball to be birthed along. Three solutions to this problem where developed. The first solution that was developed was to use a L-System as an emitter for the particles to propagate cracks.

To finish the blistering effect, a fuse SOP is appended to the end of the network. The “snap” parameter within the SOP is set to 0.225. This causes each flake to be glued to any flake that is touching it as long as the distance between them stays bellow 0.225. this creates a nice effect to the model as flakes stretch and warp before they pop off.

4.4 L-System based cracking

This requires using a source POP instead of a location POP within the POP network. As L-Systems are designed to simulate natural branching, it can be easily associated with cracks on the surface of an object. The L-System can be place anywhere underneath the painted surface to create the deformations. When used on a grid, the results are quite good.

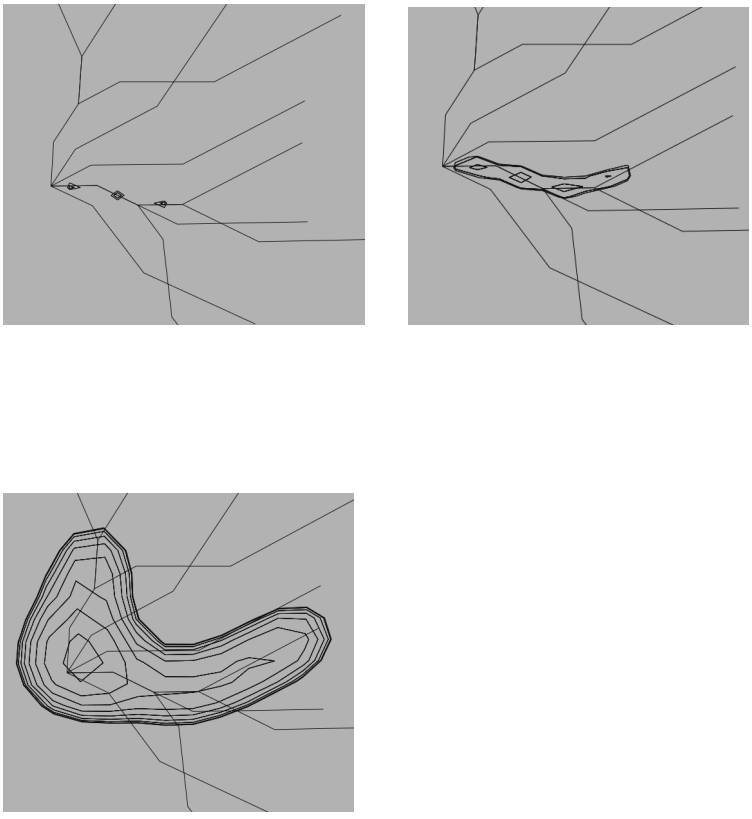

Fig 06 shows the progress of metaball birthing and growing along an L-System

Using such a technique on 3D objects is a different matter entirely. For this technique to work on say a sphere the L-System must have the same contours as geometry it is going to deform. At the moment the L-System being used is 2D. The solution to this problem includes the use of a Creep SOP.

The creep SOP allows you to deform and animate Source Input geometry (L-System/curve) along the surface of the Path Input geometry (paint surface).

This SOP changes each of the source input geometry points into a new space. The X and Y components of the source points will become the U and V positions on the surface. The Z component becomes a displacement along the surface's normal at the (U, V) position.

So, by using a scaled version of the geometry to be weathered, an L-System can be used in combination with this and a creep SOP to create a warped L-System that matches the contours of the geometry it will directly affect.

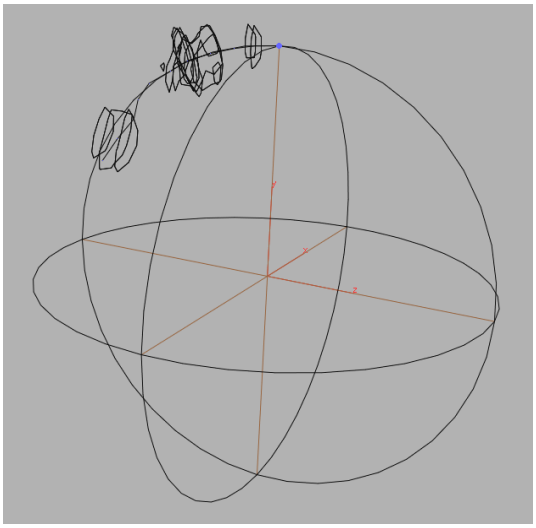

Fig 07 shows meatballs growing along a path specified by a creep SOP

A problem arises while using the creep SOP to weather more complex objects. The shattered object must be defined as a surface (e.g. NURBS

surface, primitive tube, etc.). To add to this. The creep SOP only allows a path (L-System) to be generated along a single surface, such as spheres, tori. So multi faced objects will only have a cracking occurring on a per surface basis. This is not a very efficient way of weathering objects.

During some experiments the Grid SOP, Skin SOP, Sweep SOP, Tube SOP, and Sphere SOP are able to produce suitable emitters for cracking.

4.5 Interactively Painting Cracks

In order to weather OBJ's and other multi faced surface a whole new method to deform the paint surface was created. This new approach allows the user to interactively paint cracks on to the surface, giving the user more artistic freedom as cracks can be any shape or size and do not need to be joined together.

This method takes advantage of the Paint Sop. This SOP Interactively paints color or other attributes on geometry using brush tools.

The points that have been painted are grouped together by their colour values.

```
If($CR == 1, 1, 0)
```

the above expression is used to test whether the colour value for a point is red. The points are then used to as emitters for the particle system.

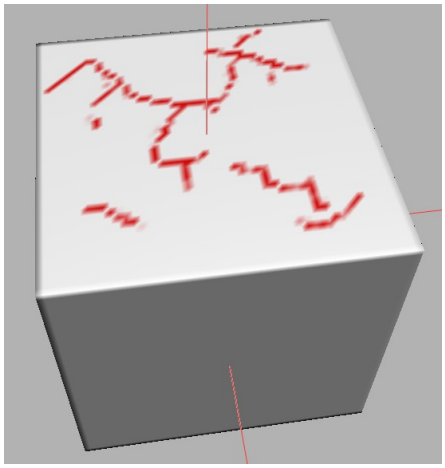

Fig 08 An example of painting cracks

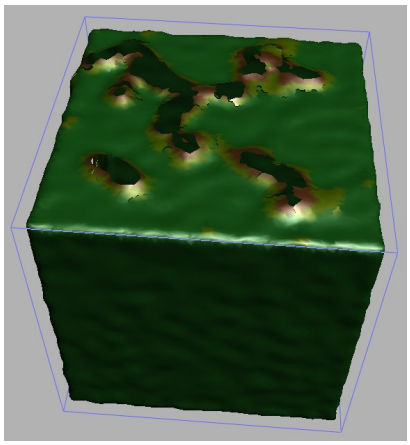

Fig 09 the results of Fig 08 at frame 360

Still this will only work if the emitter surface is inside the painted surface. Simply scaling a complex object down is not an option fig[]. The object must be made thiner so it fits snugly inside the shattered surface.

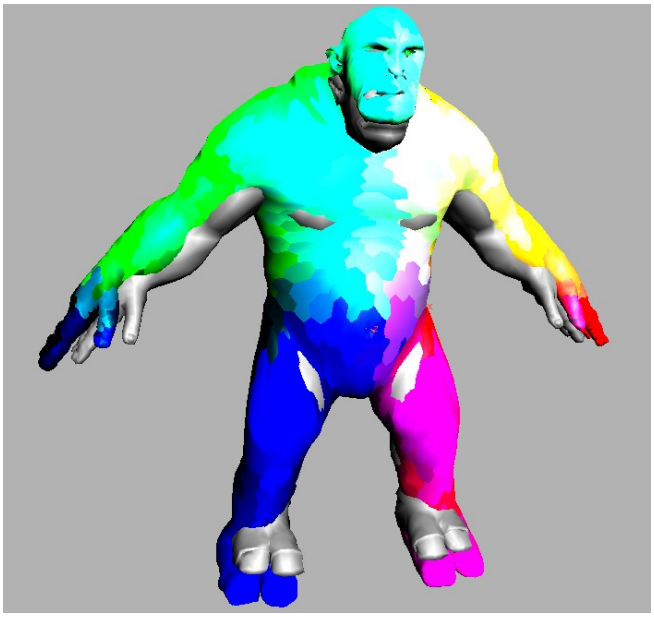

Fig 10 shows how scaled geometry causes problems when using it to weather an upper surface

Another solution to weathering complex objects is to paste pre weathered flat tiles and paste the on to the object using a Ray SOP.

The Ray SOP works by projecting rays from each point of the first input geometry in the direction of its normal, and then moves the point to any geometry the ray hits from the second input. It does this by moving the points of a Grid in the direction of the point normals. The first surface of the Collision Source (right input) will be where those points of the grid will rest. These points can rest on the other side of an object by enabling the “Intersect Farthest Surface” option in the SOP. This means that the points should continue to project to the farthest surface of the collision source

.The Ray SOP is used in the following fashion to weather more complex objects

1. A simulation created with the paint weathering tool, is called in to Houdini using a File SOP.
2. A Point SOP can then be linked to the File SOP. This is done so new normals can be added do the simulation when it is warped around the object.
3. . Next, the Point SOP can be joined to the Ray SOP and the object to be weathered can be appended to the right input of the Ray SOP.

This approach works quite well and can produce some nice results [s]. Any flat surface cracked can be stored within a library and can and called into the tool for weathering various objects. This approach to weathering makes the tool very useful and powerful as it reduces simulation time and streamlining the whole process as there may not be the need for a user to create a certain looking surface as it may already exist.

This SOP can also be used to drape clothes over surfaces, shrink-wrap one object with another, and other similar effects.

4.6 Cracking using textures

In order to give the user more control over the actual cracking of the paint surface a new method of using a texture to form the underlining basis for the cracks. This new method uses a COPnet to first load in an existing image and then uses a Point SOP to transfer the image on to the underlining geometry.

```
pic("../cop2net1/null1",$MAPU,$MAPV,D_CR)
```

the above expression transfers red colour values from a texture held in a COPNet to the points of an object. These points are then grouped separately according to their RGB values. This is done using an expression within the Group SOP. It is these points that are used to create the cracks, different colours are used to time the birthing of the cracks. The sensitivity to the RGB values are also tweakable within the tool, catering for many different types of images and the user demands.

```
if($CR >= chs("../Red_intensities/Red_Detection__High_Limit_") && $CB <=
chs("../Red_intensities/Red_Detection__High_Limit_") &&  $CG <=
chs("../Red_intensities/Red_Detection__High_Limit_") ,1,0)
```

The above is used to change the sensitivity of the group SOP grouping red values.

4.7 Giving it the weathered look

Fig 11 An example of a paint weathered surface

At the moment all the weathering tool does is bubble up the geometry. In order to turn this into a paint weathered surface, the delete Sop is used to cap the flakes at a certain height by using an expression for a flat surface or a bounding box for a 3D object. The SOP gives us an option on whether to delete points or primitives. Deleting by primitives gives the surface a polygonized look, while deleting by points gives a more realistic smoother weathered look to the surface.

The second thing that needs to be done to get the basic look of weathered paint is the colour change it undergoes as it weathers. Consider fig 11, notice how the color changes from its original light blue, to yellow, to a rustic colour along the edges. In order o simulate this in the paint weathering tool, the colour change of a flake is dependant on the height of that flake according to the surface. This is done by using the group SOP, to first group points together that have achieved a certain height and then assign a colour to those points using a point SOP. By colouring the points as opposed to the primitives, allows the transitions between colours to be a lot smoother and so gives it a more believable look.

5.1 Creating a usable tool

One of the aims for this project is to create a user friendly tool that is able to produce weathered looking paint surfaces.

These days, computers are used by almost everyone. Most users can be considered to be experts in their respective fields, but not computer experts (even animators). When creating a tool it is imperative to take these considerations into account but also not to create an interface that “talks down” to them. The users current knowledge should be enough to perform the

task successfully. The most successful computer applications have interfaces that feel natural and easy to use. This is so the user can concentrate on the task alone and not be bugged by interface design issues. It is these design issues that can be directly responsible for the success or failure of an application. A good example of a successful interface can be seen at www.Google.com. The success of this web page is in its simplicity in providing the user with an interface to a search engine. The page consists of a text box which is used to enter query and only two buttons. One returns the results and the other returns the most popular page for the query.

Also from the perspective of a company wishing to buy a new or upgrade an existing piece of software, are more likely to have their decision influenced by the user friendliness of its interface. This is primarily to help reduce the time required to train staff as training on computer systems can be quite expensive.

5.2 Houdini Digital Assets

To make things easier, Houdini has the ability to create custom interfaces for user made networks. These are called Houdini Digital Assets (HDA). HDA have now become the cornerstone of high-end effects pipelines around the world [11]. The interface is created by simply dragging and dropping panes from operators within the network onto the HDA user interface.

There are a few design guidelines to take into account when creating the paint weathering tool. These are basically to help to take into account the fact that the end user is human and not a machine. By using these guidelines it helps reduce human error and improves the time taken to complete a task using the tool. The guidelines are as follows:

- ✓ Vision
- ✓ Hearing

- ✓ Taste
- ✓ Smell
- ✓ Touch
- ✓ Long term
- ✓ Consistency
- ✓ Closure
- ✓ Relevance
- ✓ Supportiveness
- ✓ Flexibility

For a start, guidelines such as taste, smell and touch can be excluded as the have little or no relevance in the paint weathering tool.

Vision deals with the colour palette used and the layout of the interface. A bad colour palette (red and green) can cause the users eyes to grow tired, reducing efficiency, while the neatness of the interface can influence the aesthetic appeal of the interface. Color can also be used to send messages to the user. For example a “message box” in red can be a sign that an error has occurred or the user is doing something wrong.

Hearing deals with audible sounds to tell a user a piece of information, in most computer applications it is usual used to tell the user that a task has been completed or something has failed, as it attracts the users attention. Once the HDA is complete it will inherit all of the warnings and sounds Houdini uses.

Closure is the relief that you feel after you complete a task. As humans we like to know when a certain job has been completed so that we can move on to the next one. This can be done within the paint weathering tool by breaking down each step of the weathering process into different tabs, giving closure after each step.

Doing things this way also helps the interface stay neater and does not bombard the user with too much information at any on time. Human errors can also be reduced by only allowing the user to go to the next stage when the current stage is complete. This type of method is seen regularly on retail web sites such as Amazon and Play when trying to buy something.

Consistency is very important when designing any interface. It would be very annoying if a maya plug in was build with a Houdini style interface. The plug-in would more likely be rejected by the user as they now have to switch between two different ways of working within the same environment. An interface with good consistency should work with its environment and eventually become apart of it. That is what makes HDA's so great, in that it uses the panels and labels from operators in the network to create the asset. So everything from font to the colour and shape of the panes is the same as what Houdini uses. this makes the resulting HDA consistent within itself and Houdini.

A good interface should not require the user to input non relevant data that has no influence or that can damage the end result. It should give the user the minimum amount of data fields required to complete the task. This helps in not confusing the user with to many fields to complete and also helps streamline the process so the task is completed quicker.

The weathering tool should have supportiveness. Supportiveness provides the user with enough information to complete the task. This can be in the form of meaningful names and labels for data fields. For example most animators will not understand the term "metaball density field" so a more relevant term like "paint blister size" can be used instead. It is very important to consider the target audiences knowledge and how they portray things when designing a tool. Things other than labels to be considered are questions the user may want resolved like

- Where am I?
- How did I get here?
- What is happening
- Where can I go next?
- How do I get there
- What can I do?

Each stage of the tool has a short description of is done in that particular stage and the data fields have relevant names, The built tool should be quit natural and straight forward for the user to operate.

Flexibility requires an interface to be accessible to a range of users. Every user has different requirements and expectations for what the tool will be able to do for them. At the same time it must remain consistent for each user. This can include things like different languages The paint weathering tool does this by having an optional “advanced settings” panel on each stage. When enabled the user can change the core dynamics of the weathering to better suit what they are trying to achieve. All advanced settings have defaults so that the user will not be able to permanently damage the tool with silly settings.

6.1 Rendering

One of the aims for this masters project is to create a short clips that demonstrate the paint weathering model that has been created. The scene consists of a cube exported from Maya and a sphere primitive.

This was done by simulating the cracking within Houdini and then rendering it out with Renderman. This is quite easy to do, as Houdini has a ROP (Render Operator) that acts as an interface to Renderman. The scene was lit using two spot lights. As a nice touch a rust shader was added to the surface that is revealed when the paint weathers. To add more realism to the scene an ambient occlusion pass was rendered. In order to speed up matters the occlusion pass was rendered with Mantra by creating a ambient occlusion shader which was created at VOP level with a vex surface shader. The ambient occlusion pass softens the shadows

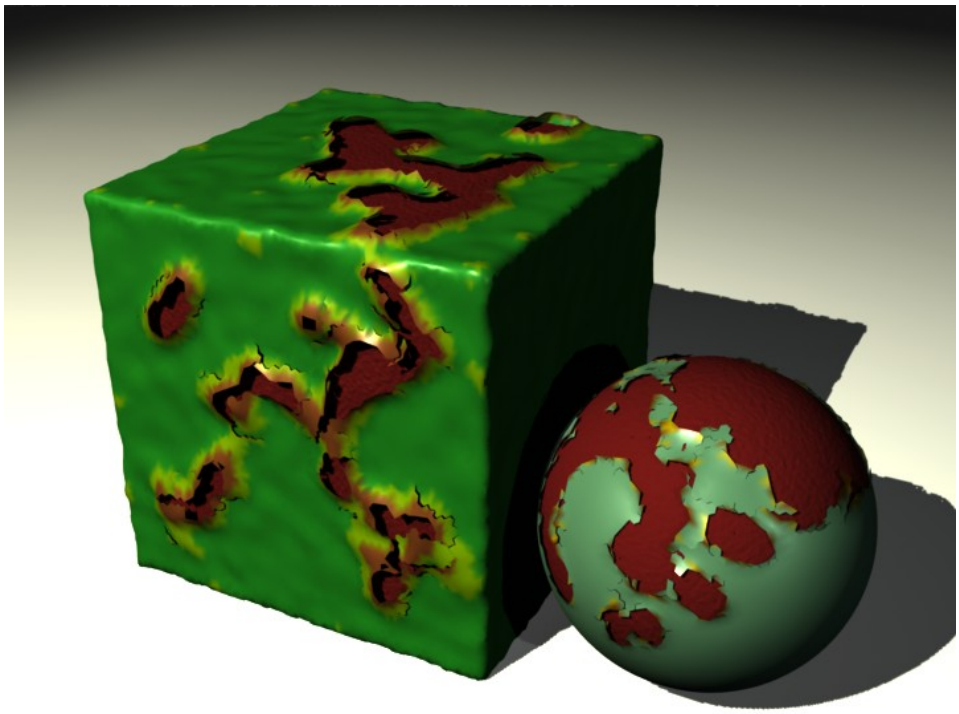

Fig 12 shows a weathered cube and sphere

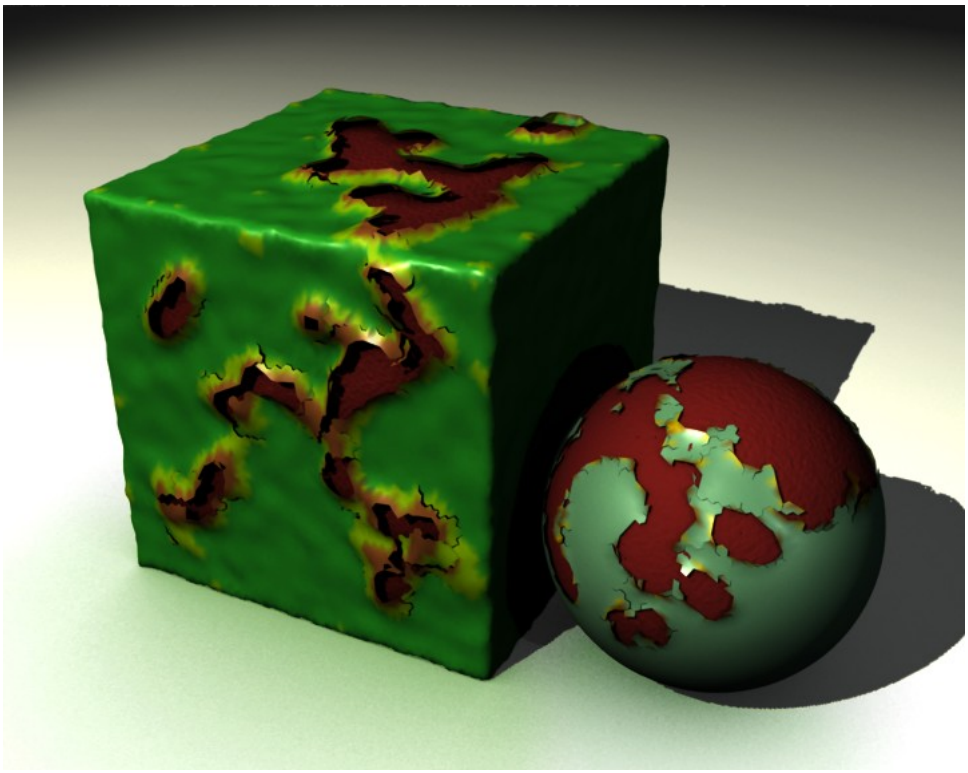

Fig 13 shows a weathered cube and sphere with ambient occlusion and colour bleed

The final two passes were then composited in Shake. To add further realism colour bleed was added to the occlusion pass by using rotoshapes and changing the gamma of it.

7.1 Conclusion

The weathering tool depends heavily on “painting cracks” using the paint SOP. Unfortunately the interactive brush that the Paint SOP uses cannot be

exported into an HDA. A few techniques were tested but all failed to providethe same results and functionality of the Paint SOP. One such technique was to use a sphere as a bounding object to collect points and store them in a group as it passed through them. The method was disregarded as it it was hard to move the sphere around with just the control handles. This would only lengthen the users job and frustrate them. Because of this the interface has been scaled down to only include grid objects cracking using L-Systems and textures. The network created can still be used to create all the effects proposed for the tool and axtra effort has been made to allow a user to effectively use it. These methods include grouping networks in netboxes colouring important SOPS and clearly labeling user controls.

The weathering tool is still fully capable of creating still images and realistic simulations of paint weathering on objects when in combination with the Ray SOP. The tool is user friendly attempts to abide to all of the design guidelines that are relevant. Certain guidelines could not be fulfilled as Houdini controls things like message pop-ups and audible sounds. The Paint weathering tool gives a user artistic ways to propagate cracks (textures) and realistic methods (L-Systems)

Interactively sticking weathered panels on to complex geometry works well only if the desired surface to be weathered is relatively smooth. If it is not, by the time the weathered panel is convincingly stuck to the geometry the Ray Sop has smoothed out all the blistering and flake effects. Still however it makes a nice smooth weathered surface. The way of getting around this problem is to scale down a weathered panel enough that the geometry it is being stuck to now seems smoother. The disadvantage of doing this is that a lot of panels will be needed to totally weather a large complex object like a naval vessel or tank.

Also, using Voronoi noise to shatter an object may not be the best approach to do it as the flakes produced are quite angular with lots of edges. Naturally flakes created by cracking tend to have smooth edges. This has been compensated for within the tool by having the cracks birthing from set paths (L-Systems) and having the flakes points deleted once they attain a certain height. This helps create a much smoother weathered look to the surface.

Overall, creating a tool that is able to continually re-use the simulations it creates in various different situation is a powerful way to work.

8.1 References

[1] W. Becket and N.I. Badler. Imperfection for realistic
image synthesis. Journal of Visualization and Computer Animation,
[2] A BRDF Postprocess to Integrate Porosity on Rendered Surfaces, 2000
[3] J.F. O'Brien and J.K. Hodgins. Graphical modeling and animation of brittle fracture, 1999
[4] K. Hirota, Y. Tanoue, and T. Kaneko. Generation of crack patterns with a physical mod, 1998
[5] Appearance Manifolds for Modeling Time-Variant Appearance of Materials, 2006
[8] http://en.wikipedia.org/wiki/L-system
[7] http://en.wikipedia.org/wiki/Houdini_(software)
[9] http://www.cmivfx.com/tutorials.asp
[10] http://en.wikipedia.org/wiki/Metaballs
[11] The Magic of Houdini, William Michael Cunningham