2D Aesthetics with a 3D Pipeline

Achieving a 2D Aesthetic with 3D Geometry

Faculty of Arts
Department of Game Design
Bachelor’s Thesis in Game Design, 15 hp
Author(s): Morgan Nilsson, Andreas Lundmark.
Supervisor(s): Ernest Adams
Examiner: Iwona Hrynczenko

May, 2017
Abstract

This thesis evaluates and tests different methods utilized to produce a 2D aesthetic within a 3D pipeline, as in, converting 3D geometry to an aesthetic that is similar to hand-drawn classic films such as Snow White And The Seven Dwarfs. This thesis explores methods to produce both exterior and interior lines that indicate shape and form of 3D models, the conclusion from the tested methods leaves with the statement that it is unlikely the human factor will be ever entirely replaced by automated solutions, and instead a mixed approach with shader relied solutions and involvement of texturing techniques which provides artistic controls where necessary, is deemed to be the most effective way of preserving the hand-drawn 2D aesthetic within a 3D-pipeline.

Keywords: Non-Photo Realistic Rendering, NPR, hand-drawn aesthetics, 3D animation, 2D aesthetic, stylized games, stylization.
Abstrakt

Denna avhandling undersöker och evaluerar olika metoder för att uppnå en hand-ritad estetik inom en 3D-pipeline, som i att få 3D modeller att nå en visuell stil som liknar en stil som återfinns i klassiska filmer såsom Snövit och De Sju Dvärgarna. Avhandlingen undersöker metoder som genererar både silhuetter samt inre linjer vars syfte är att indikera formen av 3D modeller, slutsatsen av de testade metoderna är att det är osannolikt att den mänskliga faktorn någonsin kommer bli helt utesluten i målet att uppnå den hand-ritade estetiken, och istället så är en blandning av metoder att generera hand-ritade linjer via shaders och texturerings tekniker där man behåller kontroll över aspekter av linjerna att föredra.

Nyckelord: Non-Photo Realistic Rendering, NPR, hand-ritade estetik, 3D animation, 2D estetik, stiliserade spel, stilisering.
# Table of Contents

1 Introduction .................................................................................................................. 1

2 Glossary of Terms ........................................................................................................ 2

3 Background ................................................................................................................... 3

3.1 Target aesthetic and key factors ................................................................................. 3

3.2 2D aesthetics in-depth ............................................................................................... 9

3.3 Benefits of a 2D aesthetic within 3D pipeline .......................................................... 12

3.4 Other factors important in achieving the hand-drawn 2D aesthetic ......................... 14

3.5 Shaders and rendering in 3D ..................................................................................... 16

4 Materials and Methods ............................................................................................... 18

4.1 Methods ..................................................................................................................... 18

4.1.1 Standard Outline Shader .................................................................................... 18

4.1.2 Camera Image Effects Shaders ............................................................................ 19

4.1.3 Texture Maps ......................................................................................................... 21

4.1.4 UV-Aligned Textures ......................................................................................... 22

4.2 Workflow and Evaluation ......................................................................................... 23

4.3 Tools .......................................................................................................................... 23

5 Results .......................................................................................................................... 25

5.1 Standard Outline Shader ......................................................................................... 25

5.2 SobelDepth Image Effect Operator .......................................................................... 26

5.3 RobertCross Image Effect Operator .......................................................................... 27

5.4 TriangleDepthNormal Image Effect Operator ......................................................... 28

5.5 Texture Maps ............................................................................................................ 30

5.6 UV-Aligned Method ................................................................................................. 31

5.7 Breakdown of methods in motion ............................................................................ 33
Table of Figures

Fig 1 (Walt Disney Productions. 1937) – Example of a desired 2D aesthetic ......................... 3

Fig 2 (Produced by artist known as Jax Pixeloo) – Realistic depiction of Homer Simpson ....... 4

Fig 3 (Fox. 2006) – The original look and aesthetic of Homer Simpson ............................. 4

Fig 4 (Mass Effect: Andromeda. 2017) - Typical image of unnatural facial animation during normal conversations between characters within the game Mass Effect: Andromeda .......................................................... 6

Fig 5 (Rusinkiewicz, DeCarlo & Finkelstein. 2005., under “4. Non-photorealistic Rendering: Intro and Motivation”, p. 15) – Example image of how a 2D aesthetic allows abstract storytelling ........................................................................................................................................ 7

Fig 6 (The Wolf Among Us. 2013) - A narrative driven 3D game depicting a comic style aesthetic ............................................................................................................................................... 8

Fig 7 (Produced by Kevin McShane, © Lobrau Productions, Inc.) - A sample of different comic styles by cartoonist and animator Kevin McShane in his Cartoon Kevin work .................................................. 9

Fig 8 - The Wolf Among Us (Top Left), World of Warcraft: Legion (Top Right), Borderlands (Bottom Left), Life Is Strange (Bottom Right) ...................................................................................... 10

Fig 9 (Walt Disney Productions. 1942) – Bambi, a Disney classic .......................................... 11

Fig 10 (Beane. 2012) A standard traditional 3D pipeline .......................................................... 12

Fig 11 (Williams. 2009) - Pose-to-pose animation where the main key poses are determined before any inbetweening is done ........................................................................................................ 14

Fig 12 (Williams. 2009) - Straight ahead animation, animating the entire animation from start to end .................................................................................................................................. 15

Fig 13 - Basic Edge/Line shader applied to differently shaped 3D models .............................. 18

Fig 14 - The Outline shader with a high property value in OutlineWidth ............................... 19

Fig 15 - Calculation for horizontal changes (Left), and vertical changes (Right) ................. 20

Fig 16 - Approximation calculation for the gradient at each point in the image .................. 20

Fig 17 – Convoluting the image with two kernels ..................................................................... 21

Fig 18 – Calculations to define the gradient among pixels similar to the SobelDepth operator ............................................................................................................................................. 21

Fig 19 - Texture maps with drawn strokes .............................................................................. 22
Fig 20 - The UV-Align method where UV shells of neighboring faces are aligned onto a pixel
perfect black square from both sides, outlined by the red circle ........................................ 22

Fig 21 - The Outline shader in a different camera angle ......................................................... 25

Fig 22 - Sobel Depth image effect shader applied on test geometry in Unity ......................... 26

Fig 23 - SobelDepth method failing to display lines with overlapping geometry .................... 26

Fig 24 - Normals Sensitivity set at 1.0 using the RobertCross operator for the image effects
shader ........................................................................................................................................ 27

Fig 25 - Normals Sensitivity set at 0 using the RobertCross operator for the image effects
shader ........................................................................................................................................ 27

Fig 26 - Normals Sensitivity set at 1.5 using the RobertCross operator for the image effects
shader ........................................................................................................................................ 28

Fig 27 - TriangleDepthNormals operator at 1.0 Normals Sensitivity ...................................... 28

Fig 28 - TriangleDepthNormals operator at 1.5 Normals Sensitivity ...................................... 29

Fig 29 - TriangleDepthNormals operator at 0 Normals Sensitivity ........................................ 29

Fig 30 - TriangleDepthNormals operator at 1.5 Normal Sensitivity and 25 Depth Sensitivity .... 29

Fig 31 - Quick example of texture method where strokes are directly drawn onto the surface
of the model .............................................................................................................................. 30

Fig 32 - The 3D character model using the texture method combined with the RobertCross
operator image effect shader ................................................................................................. 30

Fig 33 - Exemplary use of the texture map method to produce an aesthetically pleasing 2D
aesthetic ................................................................................................................................. 31

Fig 34 - Example image of the limited resolution strokes the texture method is suffering from
.................................................................................................................................................. 31

Fig 35 - The UV-Aligned method on display, the UV-Aligned method to the right with
"infinite resolution", and the regular low resolution appearance of a drawn stroke to the left. 32

Fig 36 - SobelDepth method stroke rendering in motion ......................................................... 33

Fig 37 - The SobelDepth operator failing at producing strokes when there is overlapping
gometry present ....................................................................................................................... 33

Fig 38 - The RobertCross operator producing unexpectedly decent results ......................... 34

Fig 39 - The RobertCross operator positives (green) and negatives (red) .............................. 34
1 Introduction

The aesthetics of 2D (for example, classical Disney movies such as *Snow White and The Seven Dwarfs*, *The Beauty and The Beast*, *The Lion King* and games such as *The Wolf Among Us* and *Life Is Strange*) have a sense of appeal and can convey emotions more powerful than photo-realistic equivalences can. The thesis explores different methods of achieving the desired 2D aesthetic through the use of different rendering and shading techniques typically seen in a 3D pipeline, to determine whether the results are within acceptable parameters, the output from the 3D pipeline is compared to typical 2D artwork in close proximity to the desired aesthetic. Where potential shortcomings from the 3D output will be discussed and analyzed to further determine what is required of the technology, what is the best method to preserve the benefits the aesthetic brings and if it is likely it ever will be attainable and where it stands today.
2 Glossary of Terms

**UV:** A common procedure to unfold 3D geometry similar to unfolding paper in order to project a two-dimensional image onto the surface.

**3D Geometry, Vertex and Faces:** 3D geometry consists of triangles which each have three vertices (vertex) that holds data such as position and vector in a scene. Faces is a visualization to make 3D geometry easier to work with in arbitrary shapes such as quads which contains 4 vertices or n-gons which can contain any number of vertices.

**Normals:** On a 3D models’ triangles, each vertex holds vector data that determines the surfaces orientation, which among other things in computer graphics, determines how light reacts to the surface.

**Polycount:** Polycount most often refers to the number of triangles present in a 3D model.

**Shader:** Is programmed instructions to how 3D geometry is displayed by computer hardware, how it should react to light, how reflective it is, if light bounces on to other surfaces, if it casts shadows and so forth.

**Backface culling:** Triangles whose normals is not oriented towards the camera is prevented from being rendered when backface culling is in effect.

**Inbweening:** In animation, there are key poses that are deemed the main poses of importance and inbetweening refers to filling the poses inbetween the key poses to have a complete animation.

**3D Data:** In this thesis, 3D data is a term meant for any data information file formats 3D applications such as Blender or Houdini can exchange with other software, which includes but is not limited to, volumetrics, 3d geometry, animations, lights and cameras.

**Rendering:** Similar to recording with a camera except within computer graphics it is generating images from data such as texture, lighting and shading information on 3D models and scenes.
3 Background

To understand what is required to achieve the desired 2D aesthetic the different elements and aspects of a typical 2D artwork must be broken down into manageable segments based on quality and function.

3.1 Target aesthetic and key factors

Figure 1 is an example of the desired 2D aesthetic especially in terms of the drawn lines describing the shape and form of what the viewer sees. It contains contour and suggestive lines, which will be explained in detail later. Other qualities evident in the image is different thickness and texture on the lines, these are qualities describing things such as material of the object and can include hardness, distance to the viewer and importance in the image, where some admittedly are abstract concepts that would seem very difficult to achieve via computer algorithms.

The 2D aesthetic is a desirable subject matter because of two major aspects, one being the fact that it has a certain kind of appeal that is difficult to achieve in any other medium, less detail sometimes is more as it enables the viewer to really focus on what is important. The other main aspect is the hand-drawn medium allows an audience to be subject more easily to a willing suspension of disbelief, thus resulting in an ability to show the audience abstract images, events and more that completely breaks boundaries that otherwise would make the audience question the whole scene, thus losing the audiences interest and engagement in the world they are experiencing at that given moment.
To clarify the different emotional impact a 2D aesthetic versus a 3D aesthetic may invoke among audiences, comparing the different images of the same character, as seen in figure 2 and figure 3, the character in a realistic 3D aesthetic (fig 2) is horrifying in comparison to the aesthetic that actually is used for the show (fig 3), it is a typical example of how a 2D aesthetic allow different aesthetics seem perfectly normal and acceptable to audiences, allowing the audience to immerse themselves in the story and emotional impacts the show may have, whilst the 3D aesthetic most likely would make the audience cringe or be sent screaming out

Fig 2 (Produced by artist known as Jax Pixeloo) – Realistic depiction of Homer Simpson

Fig 3 (Fox, 2006) – The original look and aesthetic of Homer Simpson
of the viewing hall in terror of what they have just seen. “Stylization is often also used to illustrate abstract concepts using very succinct imagery…” (Rusinkiewicz, DeCarlo & Finkelstein. 2005., under “4. Non-photorealistic Rendering: Intro and Motivation”, p. 15).

Different mediums allow for various kinds of storytelling and emotional impact; thus, it is important to have all the tools and technology available to help tell the story in the medium that may fit the requirements of that production, both regarding the story and the intended setting.

Both 2D and 3D aesthetics have their own respective pros and cons, an example is that in order to have an authentic photo-realistic 3D aesthetic is that the production value and costs is increasingly rising as the audiences expects more as the years progresses, this is particularly evident in real-time applications such as games where the expectations for animations have been a subject prone to heavy criticism if it does not reach the next-gen expectations audiences have, where next-gen tend to refer to current generation of consoles or pc hardware available to the consumers on the market. Naturally, with increased expectations, costs for game-developing studios and companies increases along with it. Capcom which is a game developing and publishing company claims the following:

Even though a full lineup of new home video game consoles will eventually be released, the industry is likely to be in a period of scant new product releases over the near future, awaiting the full-scale launch of the next-generation machines. In the meantime, development costs are projected to soar as advanced and multiple functions are added to hardware. Business alliances and consolidations may therefore occur in increasing numbers (Futter. 2013).

Capcom’s claim is further evident in other popular game titles, such as for example the 2015 title The Witcher 3 by CD Projekt RED, where Marek Ziemak, the producer of said title states that next-gen hardware allows for much more content, but in turn, also states it is difficult to deliver as the audience expects more content as well. Marek Ziemak said this regarding the next generation of games:

The world can be much bigger right now, much denser. We can squeeze much more content into the game, which makes me the development on one hand easier, because we don’t have to care about the amount of data we can have, but on the other hand it is more difficult because we have to deliver this more content. (Ohannessian. 2013).
The fact is that costs of developing realistic looking games is as Capcom stated, soaring as the expectations of audiences is increasing, a recent example of where the released game title did not reach the audiences expectations is *Mass Effect: Andromeda*. Figure 4 is an example of images that have been prominent in social media such as Facebook, Twitter and Reddit and it is an example of what the audience of the game *Mass Effect: Andromeda* is criticizing the big budget 3D AAA games’ animations for, the facial animation in particular as the game is near photo-real in almost all aspects for the setting the story is within, thus the lackluster according to the audience, animations, is breaking a lot of the audiences expectations and it is preventing the suspension of disbelief and it leads to difficulties for the audience to immerse themselves in the story and the world. An article from PC Gamer states:

...Mass Effect: Andromeda's facial animations have been an easy target for criticism. The worst examples, in which characters appear dead in the eyes or move their lips incongruously, have been packaged into short clips and run through the internet's many gauntlets of disparagement. (Wilde. 2017).

In the same article, it is also noted that a former BioWare animator, Jonathan Cooper, further states “The audience has grown more discerning, which makes our job more difficult but furthers animation quality (and animators) as a requirement,” (Wilde. 2017).

As a contrast, games that exhibit the 2D aesthetic tend to avoid the heavy critique in the visual department of the game, which most likely stems from the fact that 2D aesthetics are more indicative in nature, as opposed to photo-real graphics more direct aesthetic when it comes to telling how objects are constructed, the shape, form and material. In simpler terms, hand-drawn objects indicate material and other key factors through less detail, whereas photo-realistic graphics look exactly to how it would in the real world, the former allows the audience to interpret what they see from their own perspective whilst the latter is defined directly on the screen, little to no interpretation available.
Fig 5 (Rusinkiewicz, DeCarlo & Finkelstein. 2005., under “4. Non-photorealistic Rendering: Intro and Motivation”, p. 15) – Example image of how a 2D aesthetic allows abstract storytelling

According to Rusinkiewich at al:

…In this example the artist communicates an abstract idea to accompany an op-ed piece originally in the new york times concerning the role of higher education in our society. If, instead, we imagine replacing this with a comparable photoreal image the abstract concept would be lost amid the details. Rather we’d be distracted by the absurd image of a tiny man standing astride the mortar boards of two giant college graduates. (Rusinkiewicz, DeCarlo & Finkelstein. 2005., under “4. Non-photorealistic Rendering: Intro and Motivation”, p. 15).

Figure 5 is an example of how the 2D medium can allow a different level of immersion or storytelling despite being “imperfect” in contrast to the 3D medium, even with imperfections and flaws the 2D medium manages to maintain the audiences focus and immersion despite it not being realistic, the key is that it is consistent in its visual aesthetic and the focus for the audience lies elsewhere, the audience does not expect realistic characters and world thus the focus lies in the dialogue, narrative and other important aspects prominent in the different ways of telling stories to audiences, such as illustrations, games, movies or books.

The 2D medium can be closer compared to books as it allows an audience to themselves fill in the gaps in their minds, similar to how books do it, except the fact that books often do not even have pictures of the characters within that story, it is all descriptive through text whereas with the 2D medium it is indicated to a certain level depending on the aesthetic of that movie or game.
There are both games and movies experimenting with a blend between the two, utilizing a complete 3D pipeline and attempting to achieve a 2D aesthetic, which is the key focus of this thesis. A narrative driven game that heavily focuses on capturing the audiences’ emotional engagement, making them care about the characters and all the different choices and outcomes from those choices is Telltale’s *The Wolf Among Us*, but not limited to that title alone, Telltale’s trademark of games are games with a 2D aesthetic that focuses heavily on audience engagement and sympathy about characters and the game world. Because of their aesthetic, they can get away with imperfections in animations as that does not become the focus for the audience, it is the intricate dialogue and interesting characters, thus Telltale have not received such harsh critique for lack in animations like Bioware’s *Mass Effect: Andromeda* has. There are several other contributing factors as to why Telltale have not been as harshly criticized for animations, the 2D aesthetic however is a mitigating factor in their favor.
3.2 2D aesthetics in-depth

A 2D aesthetic can be a myriad of different art-styles as each artist tend to gravitate to their own way of drawing and painting, thus, whether a 2D aesthetic looks good or fits the requirements of storytelling for one drawing is highly subjective, there are also other factors to keep in mind when deciding on a specific style, some key factors to keep in mind is budget and allotted time available for any given project. Figure 7 is an example of several distinctive styles of drawing the same character and all exhibit an entirely different tone, mood and characteristics based on the drawing style alone.

NPR-rendering or non-photo-realistic rendering, which this paper refer to as 2D aesthetic, contain a large spectrum of different styles as opposed to photo-realistic which as the term implies attempts to mimic reality as close as possible, which with current technologies more often than not is indistinguishable from reality on the screen, at least when it comes to movies, and every year games come even closer to that particular goal of being indistinguishable from reality, which naturally increases production costs. The 2D aesthetic is liberated from attempting to mimic the real world, and thus can be indicative and exhibit less detail whilst granting the same visual information if not more to an audience purely through indicating key factors that represent whatever information the artist wants to portray, which can be anything from shape, form, material, hardness and so forth.
Since the 2D medium is not defined or restricted to anything, the artists have full control over how they wish to represent a setting, object or anything to the audience, assuming it is consistent overall. An example of different 2D aesthetics is the comic-style graphics displayed in the game title *The Wolf Among Us* or the unique hand-painted style seen in titles such as *World of Warcraft: Legion*, *Borderlands* and *Life is Strange*. The general mantra of games exhibiting the 2D aesthetic is that less is sometimes more, and that mantra is evident throughout the 2D medium overall.

The hand-painted genre of 2D aesthetics does not hide the fact from the audience that it is in fact 3D models which contain painted textures, but it grants the same benefit a true representation of the 2D medium would, it allows the artists to control what detail is important to showcase to the audience and at the same time can remain as indicative in nature, thus everything does not have to be perfect as it otherwise would have to be in a game that aims to be photo-realistic, as seen in the critique *Mass Effect: Andromeda* received about its animations. *Life is Strange*’s animation quality overall is not even in close proximity to the complexity in animations that *Mass Effect: Andromeda* portrays, yet it does not prevent the audience from losing focus and thus being unable to immerse themselves in the story. This same phenomenon can be seen in multiple games that exhibit the 2D aesthetic, they do not receive the same amount of critique when something is not perfect to photo-real as that is not their aim with that aesthetic. Aspects of photo-realism regarding graphics in games that are close to photo-realism are often referred to as uncanny valley as they are close in most aspects but something feels off to the audience (Mori. 2012), thus unsettling the audience and it therefor leads to the same misaligned focus and prevents immersion. An example of where the uncanny valley phenomenon occurred to an extreme for an audience is during the animations in *Mass Effect: Andromeda*. The key takeaway is the fact that the 2D aesthetic thanks to it being indicative rather than direct is able to control where the focus lies for the audience with
detail where it is important, and thanks to lesser detail overall imperfections are not as noticeable to the audience as well.

![Bambi and Thumper](image)

*Fig 9 (Walt Disney Productions, 1942) – Bambi, a Disney classic.*

There is a reason to why the 2D aesthetic in Disney’s classic movies has an elevated level of detail on background and static elements whereas the characters are simple in terms of detail and complexity. It has everything to do with budget and time constraints for creating the films, the 2D pipeline Disney used at the time of creating films such as *Snow White And The Seven Dwarfs* is that the most skilled artists created drafts, blocking out an animation with key poses and then passed it along to inbetweeners who then drew all the frames in-between the key poses. After the animation was complete with key poses and in-betweens it was then passed along to another artist who drew another pass over the drafts cleaning up the line-art, after that was done it was once again passed on to another artist who colored the animation. First, the budget, with a pipeline like this with many different people of varying skill and taste it would be nigh impossible to ensure that each animation in the film would be consistent with each other, secondly, which also ties in with the budget is the fact that to create highly detailed characters such as in figure 4 each frame for all the animations would take too much time even if all the artists were equal in both taste and skill (Johnston & Thomas, 1981).

Lines can convey many different things, including various combinations of lighting, material, surface markings, discontinuities, and shape. Part of what makes good artists good, in fact, is knowing how to select lines that simultaneously convey several of these cues. (Rusinkiewicz, DeCarlo & Finkelstein, 2005., under “5. Defining Lines on Surfaces”, p. 2).

Even though the final line-art on Disney’s movable characters and objects seems like simple drawn lines, they contain subtle visual information for the audience in ways not immediately apparent, this can be either through texture of the strokes, thickness, pointiness, if it is hard or soft, curved or angular and so forth. A quality line or stroke requires a lot of work and thought behind it before it can be deemed final. In figure 9 different drawn strokes display distinctive features and functions of a quality line. Lines in the drawings provide information to the
viewer if a surface is soft for example, which can be seen on the rabbit’s fur on the puffy cheeks, the lines that define the soft fur are loose and the line endings are fading carefully and deliberately, subtleties like these provide visual information that the viewers often do not even realize, they just inherently know and realize that the fur on the puffy cheeks indeed is soft to the touch. Just as line softness can describe properties of the surfaces, line thickness and width can visualize the material properties of any drawn object, for example Bambi’s rigid bony legs feature a thicker line to indicate that they are not as soft as his neck and where the back legs connect to his muscles on his softer body, organics tend to have soft wavy lines throughout whilst hard surfaces naturally tend to have harsh and abrupt lines to indicate that the surface indeed is hard to the touch, which can be seen by comparing the thick and hard lines seen on Bambi’s legs in contrast to the lines on the rabbit which is soft, thin and wavy all around.

3.3 Benefits of a 2D aesthetic within 3D pipeline

Figure 10 depicts a standard modern 3D pipeline, Since the 3D pipeline handles 3D data it can be easily iterated upon and sent back and forth in the pipeline without losing any significant amount of work. 3D data can be imported and exported from various different software and still remain intact, the 3D data can then serve multiple purposes whether it is for visual effects, compositing or lighting. The fact that it is data being passed around in turn then
means that the benefits of a 3D pipeline is agility, interchangability and iterability and enables workflows that are non-destructive.

Whereas a classic 2D pipeline used by Disney when they produced classics such as *Snow White and The Seven Dwarfs*, thanks to it being 2D (pencil drawings, coloring and so forth), each stage is hard final as 2D lacks data that can be modified without being redone from scratch, thus it must pass through each stage again if any changes are deemed necessary to be made. Essentially a 2D pipeline is similar to the 3D pipeline with the exception for the production stage seen in figure 10, the post-production stage is instead also the production stage except it is being refined continuously to the final output (Thomas & Johnston. 1981)

The immediate benefits and advantages a 3D pipeline grants to a studio is deemed as far too beneficial to ignore. It has been an ongoing effort in various industries to enable 3D models and data (such as water, fire and smoke, volumetrics and other visual effects) to be art directable into 2D aesthetics whilst maintaining the benefits of a 3D pipeline. Where art directable means that artists have control over certain aspects such as the finer qualities of the line-art produced by the shader, when and where they occur and so forth. Line or stroke rendering for 3D models have multiple areas of use, not restricted to entertainment industries such as movies and games, an example of other areas where it is beneficial to be able to output lines and strokes on 3D models is blueprints for mechanical design such as cars or machines, architectural pre-visualizations of buildings and more, which further is an incentive to achieve art directable techniques so it can fit into different specializations and needs in different industries.

Stylized lines play a role in many applications of non-photorealistic rendering (NPR) for 3D models. Lines can be used alone to depict shape, or in conjunction with polygons to emphasize features such as silhouettes, creases, and material boundaries. (Cole & Finkelstein. 2009).

The immediate benefits of a 3D pipeline is the fact that a traditional 3D pipeline is agile and iterable, which for large productions such as games or movies where hundreds of people and different studios may be involved on the same project is essential, and a traditional 2D pipeline utilized by Disney during the times when they produced classics such as *Bambi* or *Snow White And The Seven Dwarfs*, is reliant on key personnel maintaining the same artistic coherence for the entire duration of the production, which is difficult if not impossible when different studios with hundreds of people are collaborating from around the world.
3.4 Other factors important in achieving the hand-drawn 2D aesthetic

Achieving a still frame with a hand-drawn 2D aesthetic is already a great task to undertake, maintaining the aesthetic in motion however is another issue that may be difficult to overcome. There are some factors artists can take into consideration to combat the issue of the animations feeling computer generated as opposed to hand-drawn which may be the desired aesthetic of the animation.

Typically, when animating 3D models’ animators either animate pose-to-pose by blocking out the animation with key-poses or straight ahead from start to end in a linear fashion, this is also the case for hand-drawn animations. Both methods have their pros and cons but generally it is up to the animator’s preference (Williams. 2009). The relevance of how the animations feel, whether it feels hand-drawn or computer generated to the audience mostly stems from the frame rate of any given animation. An animation that is in 60 frames per second with interpolation in-between 10 key poses will seem computer generated as it will seem impossibly smooth for a hand-drawn animation, hand-drawn animation tends to be at a very low frame rate due to the large amount of work to animate that many frames per second, thus it is very important for animators to keep that in mind when they animate with a 2D aesthetic in mind within a 3D pipeline. To truly achieve a 2D aesthetic it is required of the animators to keep in mind how hand-drawn animations are done, if that is their key aesthetic to target, especially considering hand-drawn animations are imperfect in nature whereas 3D animations easily can be too perfect since they in nature by comparison to 2D is perfect in the sense of how computer generated models work.

Interpolation is basically automating the process of Disney’s inbetweening artists, it fills in frames in-between the key poses, since it is computed instead of hand-made it often requires cleanup by the 3D animators as otherwise strange and odd pops and glitches in the animation can occur, a typical example for a character animation can be that the limbs bend in ways that otherwise would not be humanly possible, which in turn would be quite offsetting for audiences.

![Figure 11](Williams. 2009) - Pose-to-pose animation where the main key poses are determined before any inbetweening is done

Pose-to-pose animation as seen in figure 11, where the main key poses are done, would then in a traditional Disney pipeline for hand-drawing animations be sent forward to an inbetweening artist, in a 3D pipeline it would be up to an animator to either do interpolation between the keyframes followed up with a clean up to prevent issues that could occur or manually animate as many frames as is required for the targeted frame rate for that animation.
In straight ahead animation however, as seen in figure 12, would be to manually animate the entire animation from start to end, which inherently has the disadvantage of not being as iterable as the key-poses animation method as the workflow is linear from start to end, whereas when animating with the pose-to-pose method the number of key poses are often few and the full animation can be previewed early on, thus animating with pose-to-pose is the preferred method for large studios as being agile and allowing quick changes is one of the advantages a 3D pipeline have over a traditional 2D pipeline, even when the final visual aspects are finalized, which is not the case for the 2D pipeline where everything would have to be sent back to the beginning of the linear workflow. (Williams. 2009)

Too truly achieve the feel of hand-drawn animations with the 2D aesthetic, 3D animated models should target the same frame-rate hand-drawn animations tend to play in, also, without interpolation as otherwise the animation may seem too artificial and smooth, which in turn would cause the feel of actual hand-drawn animations to be lost. Hand-drawn animation has a certain feel, and is shrouded in subtleties that are important to maintain that feeling, though for the 2D aesthetic it depends on the situation and medium it is being portrayed in to an audience. For example, certain 2D aesthetics such as the hand-painted style seen in Life Is Strange can get away with smooth high frame rate animations especially considering since it is a game other considerations such as playability and player control must be kept in mind. With movies however, assuming a hand-drawn 2D aesthetic is being targeted, the audience pays more attention to how everything moves and thus it is preferable if all aspects are kept close to hand-drawn animations.

Besides the frame rate, for movies, it is also important to mind how the camera (artificial camera in the case of fully computer generated imagery, also called CGI), is being placed and used in sceneries. Disney’s classic movies tend to have very limited camera movement, simply because it is very difficult to account for animated characters moving through a non-static scene. The reason being that with CGI everything is maintaining perfect perspective from the cameras point of view, in the hand-drawn medium however the artists have to attempt to draw proper perspective frame by frame which quite obviously is already difficult enough with a static or motionless camera, thus, the camera movement in the classic Disney movies tend to be mostly the camera panning to the side, where the perspective can be a little bit off without causing any distraught among the audiences as the error in perspective as the camera pans is negligible as long as the camera does not pan too far in the same area of the
current scene. This limitation as previously stated does not exist in the CGI medium thus it can be easy to neglect it, but, to truly achieve that feeling of a proper 2D aesthetic inspired by hand-drawn animations it is important to adhere to that limitation.

Games however, unless it is a true 2D game with only two axes to move between (X and Y), will not need to mind the camera limitation, but, if it is a 3D game with free movement between all three vectors (X, Y and Z) it is difficult to limit the camera as then ultimately the player would lose control over the camera which often will put the player off. Naturally it depends on the game as well, but for the sake of the argument let us assume it is a third person shooter, having limited control of the camera to bring forth a true hand-drawn feeling quite obviously would not be an enjoyable experience to play. This naturally causes a problem when attempting to portray that true feeling of a hand-drawn aesthetic, this in turn makes it especially important be mindful about what key factors in portraying the 2D aesthetic contribute to the experience of playing the game, partly, this dilemma may have been a contributing factor to the upbringing of the mixed aesthetic hand-painted games such as Life Is Strange or The Wolf Among Us portray.

3.5 Shaders and rendering in 3D

When attempting to produce any kind of aesthetic within a 3D pipeline, the 3D models need to have shaders applied to them. What the shaders essentially does it instructing the computer how it should calculate and display the 3D models. Which for example can be how the 3D model should behave when it is hit by light, the material it consists of, the reflectivity, color and so forth. A shader can be written to behave according to how it should work physically or by completely different rules, essentially, a shader determines the aesthetic of all 3D models in every aspect.

It is best explained in the, The Book of Shaders, by Patricio Gonzalez Vivo and Jen Lowe when explaining in simple terms what a shader is:

If you already have experience making drawings with computers, you know that in that process you draw a circle, then a rectangle, a line, some triangles until you compose the image you want. That process is very similar to writing a letter or a book by hand - it is a set of instructions that do one task after another.

Shaders are also a set of instructions, but the instructions are executed all at once for every single pixel on the screen. That means the code you write has to behave differently depending on the position of the pixel on the screen. Like a type press, your program will work as a function that receives a position and returns a color, and when it's compiled it will run extraordinarily fast. (Gonzales Vivo & Lowe. 2015).

Naturally, shaders in general is a very complex subject and attempting to create a shader that perfectly provides the hand-drawn 2D aesthetic is a difficult subject, albeit there have been advances in line rendering shaders and techniques over the years, it is still an issue that is difficult to tackle, the main reason it is so difficult to achieve is because there is no exact science behind where to lay down strokes and how they should be visualized, and often, it varies from artist to artist what is seen as the correct way of drawing the strokes. And since there is no exact science behind it, how can we expect a computer to determine where and
how to perfectly draw lines to achieve a certain targeted hand-drawn aesthetic. The only thing shaders have been reliably able to compute is the contour, which is the very edges of entire shapes, and occasionally ridges and valleys by calculating depth seen from the camera, which still leaves some desirable lines that details shape and form in subtler ways as often seen in Disney’s classic films.
4 Materials and Methods

The methods to gauge whether the results from the final shaders produced in this paper reaches the goal of achieving a 2D aesthetic is done with an analytical comparison to the target aesthetics, which in this particular case is the Disney hand-drawn style which is present in the classic films such as *Snow White And The Seven Dwarfs* and *The Lion King* but not limited to, as the paper also delves into different aesthetics for different purposes, such as the hand-painted aesthetic prominent in games such as *Life is Strange* and *The Wolf Among Us*.

4.1 Methods

This thesis tests and examines methods to achieve the 2D aesthetic primarily through the use of non-proprietary shaders and also techniques such as texture maps and a UV-Aligned method. The shaders are a standard outline shader and different camera image effect shaders, which are SobelDepth, RobertCrossDepthNormals and TriangleDepthNormals which are referred to as operators.

4.1.1 Standard Outline Shader

Shaders that are applied directly to models has been found to function in a comparable way, and the approach to generating outlines through a shader placed on a model is the same both in game engines and digital content creation apps such as Blender and Houdini. The outline shader used in this section is a modified version from a Unity community wiki page (Fernandez. Last Updated 2012).

Fig 13 - Basic Edge/Line shader applied to differently shaped 3D models

The shader copies vertex data and polygons/faces from the model and scales it outward in the normal direction of the face, so that the copied faces encapsulate the model. The shader then renders the copied scaled faces as a color defined in the shader.

This basic outline shader lacks a fundamental feature which is drawing of interior lines, as the nature of the shader is to copy and scale outward it is limited to only displaying contours/outlines of 3D models. The importance of interior lines is a factor which cannot be ignored in order to achieve a true hand-drawn 2D aesthetic which is based upon the classic Disney aesthetic from the films such as *Snow White And The Seven Dwarfs*, as interior lines are the most important strokes to be drawn as they portray the shapes and form of objects that
are more complex than simple cubes or cylinders, examples of scenarios where interior lines are of importance is when the arm bends on a character, otherwise the arm may seem broken or just not be distinguishable.

The only art directable properties this basic outline shader has is line Color, OutlineWidth and OutlineFill. Where the Color property is as the name implies the color for the lines/strokes drawn by the shader, it does not allow for a textured input thus it cannot be replaced by a texture that could mimic the look and appeal of a hand-drawn line drawn with a pencil. The OutlineFill property determines the spacing between the model and the copied faces, which in turn leads to line thickness, though it has the limitation that it is not dependent on camera distance and thus the thickness will remain the same thickness no matter how far or close to the 3D model the camera is placed. The spacing is determined by the degree of scaling from the normal direction of the original faces. The final property, OutlineWidth is similar to OutlineFill as it too controls spacing, but controls the spacing solely between the copied faces with no relation to the original faces.

![Image](image.png)

*Fig 14 - The Outline shader with a high property value in OutlineWidth*

When OutlineWidth is set to a high value the results will be as seen in figure 14, which demonstrates perfectly how the outline shader functions and solves the issue of rendering outlines for 3D models.

*Note: The OutlineWidth property is a modification added purely for demonstration purposes in this paper.*

### 4.1.2 Camera Image Effects Shaders

The following shaders are referred to as image effects within Unity, which are applied directly on the camera as opposed to the previous outline shader that is applied directly on the 3D models. The nature of the image effects shaders functionality is similar to the previously discussed outline shader as it still copies faces and scales them outwards according to vertex data available on the model, the main difference is that the image effects shader also takes surface normals and depth information from geometry into account when calculating where to
draw lines, the image effects shader uses operators which in this case refers to the math used for calculations. The operators decide where to draw a stroke if two pixels are far apart in depth and or if two pixels have differentiating normals, an example is, if a finger is in front of another, if the distances from the two neighboring pixels between the two fingers is far apart in depth from the cameras point of view, that is where the stroke will be drawn (Unity Technologies).

**SobelDepth Method**

The way to determine whether a stroke should be drawn or not at any given area in the standard SobelDepth operator is to look for changes in pixel intensity, if there is an intense area where there are sudden changes in the intensity among neighboring pixels it is likely that it is an area where an edge is occurring in the image. The way the Sobel operator looks for pixel intensity changes is by using derivatives, and thus, major pixel intensity changes are determined by high changes in the gradients for the image (OpenCV dev team. Last updated 2017).

SobelDepth Formulation:

The operator uses two 3x3 kernels, one is for horizontal changes and the other is for vertical changes. \( A \) is defined as the input image, \( G_x \) and \( G_y \) is two images which at each point contain the horizontal and vertical derivative approximations (OpenCV dev team. Last updated 2017).

\[
G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \ast A \quad \text{and} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \ast A
\]

*Fig 15 - Calculation for horizontal changes (Left), and vertical changes (Right)*

The \( x \) coordinates will increase in the horizontal direction to the right and \( y \) will increase in the vertical direction downwards on the image, at each point in the image. To get the approximation of the gradient the following equation can be used:

\[
G = \sqrt{G_x^2 + G_y^2}
\]

*Fig 16 - Approximation calculation for the gradient at each point in the image*

These calculations would be done for every pixel in the image and strokes would be drawn where a pixel’s gradient is at a higher value than its neighboring pixels. (OpenCV dev team. Last updated 2017).

This is how the SobelDepth operator works when converting regular images to seem as if they had been hand-drawn, the approach within Unity is similar with the exception that there is more information to calculate from, such as surface normals on the geometry, calculating from depth however is similar to the image conversion as the camera outputs depth as greyscale images itself.
The SobelDepth operator in Unity only have two changeable properties and those are Edge Exponent and Sample Distance, where the property Edge Exponent changes the depth threshold, as in, smaller values allow the shader to detect edges with smaller differences in depth. The Sampling Distance property is essentially the same property as OutlineFill in the standard outline shader which is applied directly on models, it determines the spacing or scale of copied faces, which in turn leads to thinner or thicker lines to be drawn on the object (Unity Technologies).

RobertCrossDepthNormals Method

The solution to detect edges is similar in most image effects based shaders whose intent is to draw strokes on 3D models. They look for changes in pixel intensity and if the gradient in the derivative is crossing a threshold, and thus the edges have been determined by code and the shader have an approximation of where to place strokes. The RobertCrossDepthNormals method is no different, the only difference is the math to compute where edges may be.

RobertCross Formulation:

With the RobertCross operator, the image first needs to be convoluted which is done with the following two kernels:

\[
\begin{bmatrix}
+1 & 0 \\
0 & -1
\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}
0 & +1 \\
-1 & 0
\end{bmatrix}.
\]

\( \nabla I(x, y) = G(x, y) = \sqrt{G_x^2 + G_y^2}. \)

The I(x,y) in figure 21 is referring to the input image and Gx(x,y) and Gy(x,y) is the merged/convolving kernels, the gradient is then defined (Fisher., Perkins., Walker., Wolfart. 2003.).

The RobertCross method have two properties over the SobelDepth method which are of interest. Namely the Depth Sensitivity property, and the Normals Sensitivity property. The Depth Sensitivity property as the name implies controls the shaders threshold to look for the difference between neighboring pixels in depth, similarly to the Depth Sensitivity property, the Normals Sensitivity property controls the threshold for the difference between neighboring pixels normals (Unity Technologies).

4.1.3 Texture Maps

The method is identical with ordinary workflows to apply texture and material definition on
3D geometry through the usage of texture maps. The key benefit of utilizing this method is that the results are completely and fully art *directable* by artists themselves, as they maintain control over what details are important and what can be omitted altogether.

![Texture maps with drawn strokes](image)

*Fig 19 - Texture maps with drawn strokes*

The texture method as previously stated utilizes texture maps that are manually drawn upon by an artist, such a texture map can look as the above image. For demonstrational purposes the maps are kept with as little detail as possible, though noteworthy is that it can contain any number of details and shading information the artists may choose to visualize on the character, examples of details are, self-shadowing with darker tones rather than a flat color, rosy cheeks, wear and tear on clothes and so forth.

### 4.1.4 UV-Aligned Textures

The method was utilized in the game *Guilty Gear Xrd* by the game development studio Arc System Works, and the method can basically produce strokes with infinite resolution, the reason it functions is because the UV’s of the model is aligned on two parts of the UV onto a pixel perfect section in the texture map, which in turn leads to no texture filtering that decreases the quality of the pixels the closer the camera gets to the geometry.

![The UV-Align method](image)

*Fig 20 - The UV-Align method where UV shells of neighboring faces are aligned onto a pixel perfect black square from both sides, outlined by the red circle*
As seen in figure 20, two halves of the geometry are aligned onto a pixel perfect black section of the texture map.

4.2 Workflow and Evaluation

The shader relied methods are first tested with a set of test geometry with differentiating normals to test the shaders in common geometry shapes. The outcome from the test geometry provides a gauge to how well the shaders perform when there is overlapping geometry and simple indentions and circular shapes where a drawn stroke is desirable to provide visual information to the shape and form when there is no other information besides the strokes, such as light and shadows.

The shader solutions and the texture map solution is then tested in practice on an animated character where three frames of the animation provide a gauge to how well the solutions work in a difficult in motion scenario.

The results from all the tests are then evaluated qualitatively per the authors of this thesis own ocular judgement, focusing on how it solved difficult areas for example, bends of arms and legs, and the quality of the displayed strokes and most importantly where artifacts such as skipped and inconsistent strokes become prominent.

The workflow used to conduct the tests is as following:

1. Create test geometry that has common shapes and forms of everyday objects that may typically be found in a scene, with a 3D digital content creation application.
2. Apply geometry shader on test geometry which is then rendered, in this particular case in Unitys real-time game renderer.
3. Apply camera image effect shaders on a virtual camera within Unity and render the results, fine-tune the parameters within the code of the shaders to produce desirable results on the 3D geometry.
4. Use an image editing application to hand-draw strokes on a 2D image to produce lines that depict shape and form on an existing UV map so that the drawn strokes are projected on a 3D character where desirable.
5. Use a 3D digital content creation application such as Blender to produce test geometry and create UV’s according to the UV-Aligned method on a pixel perfect area on a texture map.
6. Analyze and compare all results with a target 2D aesthetic in mind.

4.3 Tools

Materials used for the production and testing of the methods is listed below.

**Unity 5.6.0b10**
Unity is a game engine with customizable shading capabilities, as in, all shader code is open for inspection and modification. The image effects shader solutions tested in this thesis is part of a standard effects package included in Unity available for the public domain.

**Visual Studio 2017**
Visual Studio is a code editor which was used to create modifications to code used in this thesis as well as inspect and determine how the different operators function.
**Blender 2.78c**
Blender is a digital content creation suite and it was used to create basic models that would serve as test geometry to demonstrate the shaders generating lines on different shapes to determine the functionality and capabilities of those shaders in normal use-case scenarios.

**Mixamo**
Mixamo is an online software application service by Adobe that provides a large library of motion capture data available to be used on either existing 3D characters in the library or custom characters uploaded to the service.
5 Results

The results will be presented in a used method order, as in, the method used to produce the 2D aesthetic, followed with tests conducted with the solution methods in motion.

5.1 Standard Outline Shader

The outline shader may serve some purposes; unfortunately, at certain camera angles it produces artifacts that subjectively does not look aesthetically pleasing as seen in figure 21. The limited control and the way the shader functions it is prone to be unreliable on even simple objects as it has no way of controlling which copied faces to display other than normal direction of those faces, the reason the copied faces that is closest to the camera does not render is because of backface culling which is a method to prevent rendering faces that are not visible to the camera, for example, the back of the cube in figure 21, for the outline shader the normal direction of the copied faces is reversed so the opposite occurs.
5.2 SobelDepth Image Effect Operator

Using the SobelDepth method the results are similar to the standard method outlined in the outline shader, it only renders the contour of the 3D models and completely omits to render any interior lines. Though one thing to note is that the SobelDepth method have issues rendering lines where models overlaps, as seen in the image below.

The reason the SobelDepth method fails to generate a stroke when there is overlapping geometry is because the SobelDepth operator only takes the depth information from the camera to calculate and decide where to place strokes, and since the depth between the cubes have a similar gradient the shader then omits to place a stroke.
5.3 RobertCross Image Effect Operator

As can be seen in figure 24, the RobertCross operator tries to implement interior lines for the 3D geometry, though at the current setting it is unreliable and skips important lines depending on the shape of the geometry. Important to note is that each model is setup with different normals to demonstrate the difference in calculations depending on the normals of the geometry.

In figure 25, the Normals Sensitivity setting is set at 0, which removes any calculations that is related with the normals of the model, and it is close to the result of the SobelDepth operator apart from it being able to reliably calculate all the exterior contours on the test geometry even when there is overlap present.
At a setting of 1.5 on the threshold setting for the Normals Sensitivity property, the results are closer to the desired aesthetic, at least on the shapes present in figure 26. Important to note is that each model has differently set normals to demonstrate the importance of correctly setup normals using the RobertCrossDepthNormals method. From the right, the first model in comparison with the second model with different normals, it accurately calculates important strokes to define the shape, whereas on the second model the normals are what is usually referred to as set to flat shading at which point the RobertCross operator almost displays a wireframe of the models’ topology, which is even further evident on the fourth model, with a higher Normals Sensitivity setting the missing edges on the fourth model appear as well.

5.4 TriangleDepthNormal Image Effect Operator

Lastly, the TriangleDepthNormal operator which by name implies it is using a similar approach to the RobertCross operator.

Fig 26 - Normals Sensitivity set at 1.5 using the RobertCross operator for the image effects shader

Fig 27 - TriangleDepthNormals operator at 1.0 Normals Sensitivity
The TriangleDepthNormals operator provides identical results to the RobertCross operator at all settings with the slight difference of it producing thinner strokes at the same Sample Distance values. Although the results are visually identical, the difference lies in the code, TriangleDepthNormals use less samples than the RobertCross operator and thus for usage in games the TriangleDepthNormals operator may be a more viable solution for optimization purposes. In achieving the 2D aesthetic they can both be treated as equally viable as they use the same calculations for finding edges to determine where strokes should be drawn by the shader.
The Depth Sensitivity property which is present in both the TriangleDepthNormals and RobertCrossDepthNormals operators, when set at high values produces irregular lines for the test geometry as seen in figure 30, the Depth Sensitivity setting is a setting intended for more specific cases as the standard value at 1.0 produces the optimal results for the test geometry.

5.5 Texture Maps

![Fig 31 - Quick example of texture method where strokes are directly drawn onto the surface of the model](image)

The method seen in the game *The Wolf Among Us* utilizes the method seen in figure 31, albeit to a much higher quality and delivers aesthetically pleasing results whilst maintaining the feeling of a true 2D aesthetic.

![Fig 32 - The 3D character model using the texture method combined with the RobertCross operator image effect shader](image)
Combining the texture method with the RobertCrossDepthNormals or TriangleDepthNormals operator set to only render the contour of the character model, the end results are aesthetically pleasing to a potential audience, as seen in figure 32.

A notable example of this method in use is the game *The Wolf Among Us* by Telltale:

![Exemplary use of the texture map method to produce an aesthetically pleasing 2D aesthetic](image)

### 5.6 UV-Aligned Method

Whilst the texture map method produces visually pleasing results it exhibits one issue that in some cases are undesirable, and that is limited resolution of the strokes, which is evident on a close-up of the model as seen in the image below.

![Example image of the limited resolution strokes the texture method is suffering from](image)
As can be seen in figure 34, the strokes become blurry and fuzzy, typical results from limited resolution, which becomes more and more apparent the closer the camera gets to the geometry the texture maps are applied to. A potential solution is to increase the resolution of the texture maps but for games that would increase the demand on the computer hardware, resulting in a game that may or may not be difficult to run at an acceptable frame rate.

The results which can be seen in the image below showcases a distinctive difference in the resolution of the strokes, where the UV-Aligned method is to the right, and a regular painted stroke can be seen on the left. The only limiting factors of the UV-Aligned method is that it is geometry reliant, as in that the smoothness or shape of the stroke is dependent on the geometry polycount and the lines are essentially hard locked as distinct hard drawn lines, preventing the usage of soft lines to indicate soft material properties on for example fur or organics.

Figure 35 depicts the usage of the UV-Aligned method on identically shaped geometry and as can be seen the UV-Aligned method provides superior line quality over the texture maps method. The UV-Aligned method is however reliant on the number of polygons on the geometry in order to enable artists to shape the strokes freely, thus in order to achieve a really smooth stroke a very high polycount would have to be used. But, the quality of the strokes edges will always be crisp and clear in comparison to the texture maps method since the UV-Aligned method is aligned to pixel perfect sections of the texture map, any texture filtering or stepping to produce blurry and fuzzy lines is no longer present, no matter how far away or close to the camera the geometry is.
5.7 Breakdown of methods in motion

Producing a 2D aesthetic on still images is one thing, achieving it on characters in motion is an entirely different matter altogether.

In figure 36 the SobelDepth operator is used to produce the strokes/lines, as expected the image effects shader produces a result similar to the test geometry used previously. The SobelDepth operator manages to produce exterior outlines reliably except for edges where geometry overlaps. Which is outlined in the image below.

Fig 36 - SobelDepth method stroke rendering in motion

Fig 37 - The SobelDepth operator failing at producing strokes when there is overlapping geometry present
The RobertCross operator produces comparable results to the previous test geometry, though it also manages to produce desirable interior lines at certain key areas.

![Fig 38 - The RobertCross operator producing unexpectedly decent results](image1)

The RobertCross operator produced unexpectedly decent results even with interior lines, though it fails at certain parts of the model to draw a full line. Notable areas where it succeeded is creases at the legs, arms and finger definition at certain camera angles, which are outlined in green circles in the image below.

![Fig 39 - The RobertCross operator positives (green) and negatives (red)](image2)

Whilst the RobertCross operator produces aesthetically pleasing results, it also produces artifacts (which are outlined in red circles in figure 39) which are difficult to overcome even with different property settings. Also, opposed to the test geometry where the Depth Sensitivity property had limited positive effect whilst the Normal Sensitivity preferred higher values to produce the best results, the opposite was true for the 3D character in motion.
6 Analysis

Each method tested serves different purposes and each hold their own upsides and downsides. The standard outline shader which only extrudes faces and pushes them outward is the cheapest to compute but also provides the least quality when it comes to strokes, not to mention minimal control over the strokes themselves.

The methods that expands on that is SobelDepth, RobertCrossDepthNormals and TriangleDepthNormals, which are a step further to the desired results and in turn, the desired aesthetic, though they still only provide results that have limited art directable capabilities and at times have difficulties achieving a reliable line quality throughout on the models.

Whilst the TriangleDepthNormals operator was not tested in motion as it produced the exact same results as the RobertCrossDepthNormals operator, they both produced the best shader generated line aesthetic among the shader relied solutions tested in this thesis.

![Fig 40 (Walt Disney Productions. 1937) - Snow White in Snow White And The Seven Dwarfs](image)

If the above image is the target aesthetic for the shaders tested in this paper, take note of how the lines on Snow White is drawn, they are carefully placed and drawn to suggest the shape and form, most noticeable on the hands and the transition from the neck to the chin, if those lines were missing it would be difficult to tell where her head starts and the character would look unnatural. The lines are especially important in the Disney classics as the characters have no shading information, which means no shadows or light, everything is a flat color and the specific reason for that is because drawing all that shading information frame by frame would take far too much time and be too costly to produce.

Whilst the RobertCross operator produced the most aesthetically pleasing shader relied solution, it still suffers from limited control and is prone to produce unpredictable results, the results vary from geometry to geometry, as seen in the test geometry (fig 24-26) compared to
the character in motion (fig 38 and 39). The RobertCross operator managed to produce interior lines at important places which was one of the reasons this thesis was written to attempt to resolve and find alternate methods that may resolve the issue of producing interior lines through an existing non-proprietary shader solution.

One major difficulty with shader relied solutions to this day is to generate reliable lines at places of importance, and the primary difficulty is with faces which it is noticeable apparent in figure 38, even though it succeeds with producing a stroke for the shape of the nose, it lacks distinction at the neck and crevices in the face, as it is such small ridges and valleys it would require a dramatic increase in Depth Sensitivity, which in turn would produce artifacts all over on the model whilst also producing unreliable results elsewhere. The solution for the face would either be the texturing method or the UV-Aligned method as artistic control is maintained and deliberate lines can be placed by a skilled artist to provide the necessary visual information. The only drawback of those two methods is that lines remain indefinitely regardless of what the character is doing, thus it stands to reason that a combined approach between a shader relied solution and a texturing method may grant the necessary artistic control and still have the benefits of a 3D-pipeline intact whilst producing a 2D aesthetic that is pleasing to the eyes.
7 Conclusion

Comparing the results generated by the shader methods with the painted and UV-method the advantages of having complete art \textit{directable} strokes and still maintain the flexibility and iterability of the 3D-pipeline falls in favor for manually painted strokes still, it evidently is a proven method considering multiple games utilize those methods over a complete shader relied solution. In some cases the texture resolution would be a limiting factor for a 3D model who have drawn strokes in the texture maps, especially if the requirements for the model is to be able to look good at closeups, if closeups is a requirement the UV-aligned method is preferable although it requires more work, with the illusion of unlimited resolution the strokes maintain their function and remain art \textit{directable} with the exception of having textured strokes, they will essentially be hard-locked to a certain line aesthetic as soft lines and line endings is impossible to produce with the UV-aligned method.

According to the ocular judgement of the authors of this thesis, the results produced from the test geometry and the tests conducted on the 3D character. It is concluded that the most aesthetically pleasing method to achieve the 2D aesthetic is to combine a shader relied solution for exterior edges and then proceeding to utilize either painted texture maps for interior lines or if texture resolution is a limiting factor, the UV-aligned method though the requirements of the 2D aesthetic for a project utilizing the UV-aligned method, must not be to utilize soft lines to indicate soft material properties of either organics or soft fur.

Whilst the research for producing a 2D aesthetic through purely shader relied solutions are moving forward and improving as time moves on, it is undeniable that it will always be missing a crucial factor if the intention is to have an automatic solution to transform 3D geometry to 2D artwork, and that crucial factor is the human factor. There will always be subtleties in lines and different intentions with artwork, thus it is likely a computer-generated solution will never be able to replace the pen in the hand of a skilled artist, though for certain industries and pipelines it will absolutely serve as a tool to produce certain aesthetically pleasing art.
References


Games


Gearbox Software. 2009. *Borderlands*.


Movies & TV-series


Walt Disney Productions. 1937. *Snow White and The Seven Dwarfs*.

Walt Disney Productions. 1942. *Bambi*.

Code


*Note: Code has been modified for demonstrational purposes for this thesis, however shader functionality remains the same.*