Grease Pencil: Integrating Animated Freehand Drawings into 3D Production Environments

Joshua Leung*
Blender Foundation

Daniel M. Lara†
Pepe School Land

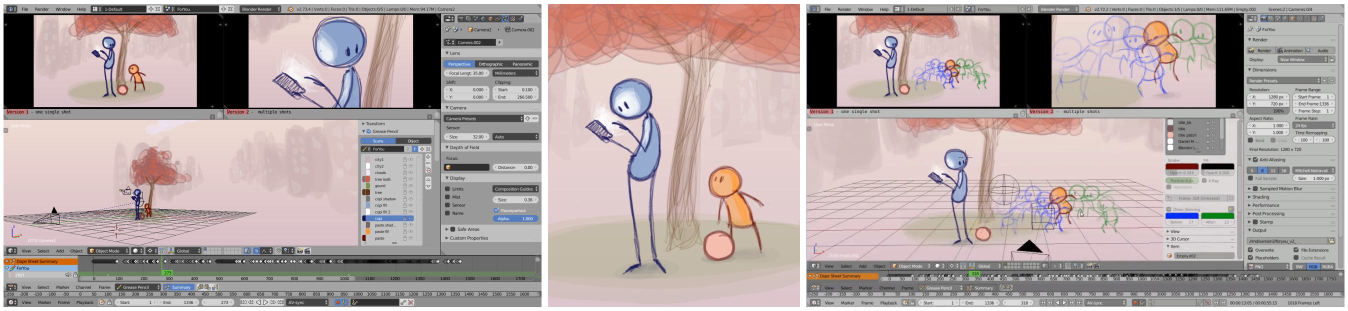

Figure 1: *Screenshots from the short film "For You" by Daniel M. Lara, for testing the stroke editing functionality provided by the Grease Pencil toolset. a) The scene can be rendered from different cameras/perspectives, b) A frame from the finished film, c) The onion skinning functionality*

Abstract

Freehand drawing is one of the most flexible and efficient ways of expressing creative ideas. However, it can often also be tedious and technically challenging to animate complex dimensional environments or dynamic choreographies. We present a case study of how freehand drawing tools can be integrated into an end-to-end 3D content creation platform to reap the benefits from both worlds. Creative opportunities and challenges in achieving this type of integration are discussed. We also present examples from short films demonstrating the potential of how these techniques can be deployed in production environments.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: animation, drawing, interface, 2D, 3D, storyboarding, storytelling, freehand, NPR, sketch, stroke-based illustration

1 Introduction

Over the past few years, there has been a cross-disciplinary movement to explore and develop new visual styles for computer generated imagery. This stems in part from a sense of boredom with the big budget hyper-photorealistic "CGI Cartoon Look" epitomised by the output of studios such as Pixar, Dreamworks Animation, and Disney to name a few. Another key reason is that photorealistic rendering technology has advanced to the point where digitally created assets are routinely indistinguishable from their real life counterparts. With the previous "holy grail" of rendering attained, there has been burgeoning interest in the untapped potential of techniques such as Non-Photorealistic Rendering and mixed-media productions. Non-Photorealistic Rendering (NPR) is a broad umbrella-term for many different visual styles including those which attempt to mimic traditional art styles (such as toon/anime [Harvill 2007], impressionistic oil paintings [Meier 1996], pen/pencil hatching [Winkenbach and Salesin 1994], and traditional Chinese calligraphy/paintings [Sun et al. 2012]). Mixed-media productions have also become increasingly popular, such as the increasing use of 3D printing and VFX techniques for stop motion animation at Laika, or performing stop-motion animation using 3D software (as used by Animal Logic for *The Lego Movie* (2014)).

An underexplored area with potential for interesting developments is the combination of hand drawn 2D animation and 3D content creation software (3D DCC). Freehand drawing is recognised as one of the most flexible and efficient ways of expressing creative ideas, with a long and storied history as a production technique for feature length film productions. Complex ideas and emotions can be conveyed using just a few simple lines – or line, singular – a feat which may sometimes take many weeks of modelling, texturing, shader design, lighting, and high resolution rendering to begin to approximate.

However, hand drawn 2D animation has several notable limitations. Most notably, it is not well suited to situations involving complex spatial relationships or transformations. Examples of scenarios where it often becomes technically challenging (sometimes prohibitively so) include swooping and/or rotating camera moves, complex forms being deformed or viewed from different angles, and dynamic scenes where the spatial relationships between objects is constantly changing (e.g. characters caught in a raging river). While these are all quite difficult for humans to achieve unaided, they become much easier to accomplish in a 3D DCC package. This led us to ask the question: What if "2D animation" could be done using 3D software to take advantage of the best of both worlds?

This paper presents a case study on how we integrated the ability to draw and animate freeform 2D strokes in an end-to-end production suite, Blender 3D, the challenges faced, and a discussion of creative possibilities that such 2D-3D hybrid integrations present for

*e-mail:aligorith@gmail.com
†email:web@pepeland.com

artists. The "Grease Pencil" system presented here allows animators to draw freeform strokes in a 3D viewport (see Figure 1), with the ability to edit these strokes afterwards (i.e. to change their location, orientation, or scale in 3D space, or to reshape these strokes using many of the same editing tools available for editing traditional 3D geometry). As these strokes are created in 3D space, it is possible to integrate these alongside/among assets (i.e. geometry, particles, and/or physics simulations) created using more conventional 3D CGI techniques. There is also the ability to create and edit animations in a flipbook style, using a standard dopesheet interface for viewing, retiming, and duplicating these keyframes.

There are several benefits to our approach over traditional 2D animation workflows and prior systems in the literature. These are:

1. **It provides artists with an efficient non-destructive workflow** – Artists do not need to worry about scanning, naming, storing, and importing images every time they make changes to the animation [Roosendaal and Mendiola 2015].

2. **It allows artists to leverage 3D techniques to assist their work in technically tricky shots** – Specifically, the ability to move and position the camera in 3D space to perform complex camera moves, and the ability to manipulate and position the sketches in 3D space as needed to achieve different effects are particularly valuable.

3. **Artists can precisely tweak and reshape their strokes afterwards** – In most other systems, artists need to redraw any strokes which do not have the desired shapes. Our tool allows artists to perform fine manipulations on a point by point basis, for more control over the final result.

4. **It is natively integrated into a fully-featured, end-to-end 3D content production environment** – Artists can make use of conventional 3D assets and rendering tools. They can also make use tools such as motion tracking and masking to integrate sketches into real world environments, compositing tools for applying standard post-production effects, and can also non-destructively sequence up different shots from different cameras with little effort.

This paper is divided into several sections. First, we review the design space for tools involving both hand drawn/2D elements and 3D elements, and discuss how our system fits into this design space (Section 2). Second, we discuss the general architecture of the Grease Pencil tool and the capabilities of the first and second iterations of our Grease Pencil implementation – the first implemented basic annotation and review functionality (Section 3), and the second added editing features and other features needed to make Grease Pencil more viable for producing hand drawn animation in a 3D environment (Section 4).

2 Methods for Integrating Freehand Strokes into 3D Environments

Elements created using 2D and 3D techniques can be integrated together in several different ways. The design space for these techniques can be defined in terms of the following two criteria:

1. **The manner in which the 3D environment is used** – That is, does the 3D environment or 3D objects actually feature in the final output? Alternatively, is the shot rendered using the 3D environment?

2. **The manner in which the 2D strokes are integrated into the scene** – That is, are the 2D strokes applied as standalone entities, or are they attached to 3D geometry?

2.1 3D as Proxy Geometry for Animating in 2D

Due to technical limitations, the use of 3D CGI in early attempts was mainly limited to providing reference geometry in the form of wireframe proxies (e.g. boxes, spheres, and/or cones) that artists could use then draw, paint, and animate over in 2D to create the final output[Disney 1983]. Notable early examples includes Glen Keane and John Lasseter's work on the "Where the Wild Things Are" short [Amidi 2011], the ballroom scene from Beauty and the Beast [Disney 1991], and Dreamworks Animation's work on Prince of Egypt and Spirit: Stallion of the Cimarron [Cooper 2002].

2.2 3D as Canvas for Hand Drawn Strokes

More recently, there has been another class of attempts aimed at combining the flexibility and "human touch" of hand drawn art, with the power and dimensionality offered by 3D software. Strokes in this category are deformed and projected so that they are attached to the geometry in 3D space. The *Deep Canvas* system was used on Disney's *Tarzan* (1999) to texture and dress the intricate jungle environments using 3D paint strokes, making it possible to choreograph complex swooping camera moves (which were prohibitively expensive/challenging otherwise), while still benefiting from the expertise of experienced set painters.

Another notable attempt was the *Meander* system used on Disney's *Paperman* short [Whited et al. 2012]. This was used to allow animation artists to hand draw line strokes on top of traditionally animated 3D models, in order to capture the charm of 2D animation while maintaining many of the production advantages of 3D CGI production pipelines. Temporal cohesion algorithms were developed to deform and interpolate between strokes drawn by artists on keyframes.

2.3 Hand Drawn Strokes Integrated in 3D Environment

It is not necessary to "attach" hand drawn strokes to any traditional 3D objects. Instead, hand drawn strokes can "free floating" geometry elements in their own right, allowing them to exist *alongside* conventionally modelled elements in a 3D environment. Schmid et al.'s *Overcoat* [Schmid et al. 2011] allows artists to create strokes in 3D space as volumetric entities; these strokes are rendered by accumulating brush samples in the affected regions. Bourguignon et al. [Bourguignon et al. 2001] developed an approach where strokes are created as Bezier surfaces instead. Bae et al.'s *ILoveSketch* [Bae et al. 2008] allows artists to create models as NURBS curves in a gestural interface optimised for pen-based interaction.

2.4 Where Does Grease Pencil Fit?

The Grease Pencil system can technically be used in all of the ways described here (i.e. strokes can screen aligned, can be projected onto geometry, or drawn free standing). Our work provides several key contributions and insights over prior systems. First, we provide some practical examples of how hand drawn strokes can be used in a 3D environment alongside props and sets that have been created using standard 3D modelling techniques. Second, instead of using Bourguignon et al.'s [Bourguignon et al. 2001] positioning widgets or the workplane approaches used by Bae et al. [Bae et al. 2008] and Dorsey et al. [Dorsey et al. 2007], we use the "3D Cursor" concept built into the host environment to provide a tangible reference point for controlling how points are projected from screen-space to 3D-space. Finally, we introduce the concept of providing tools to edit these strokes in 3D space as if they were meshes. Thus, strokes in our system are not restricted to being 2D "cards" that are either unable to be edited or can only be edited "in plane".

3 Grease Pencil 1.0 – Annotation Tool

Grease Pencil was originally intended as an in-viewport planning and review tool for use within the different editor contexts in Blender. For example, users could use Grease Pencil to draw freehand notes or corrections directly on their models, renders, and shader/compositing networks; such annotations help users document their work and aid collaboration in teams (e.g. giving art direction to remote workers). Animators could use the basic flipbook-style animation capabilities to quickly thumbnail key poses directly in the viewport to aid in planning out their shots. As Grease Pencil was not aimed at providing output-quality final renders, many unconventional design decisions were made to prioritise speed and ease of drawing over line quality.

Strokes are implemented as simple polylines (i.e. a chain of straight line segments joining each point to its immediate neighbours). Polylines are used instead of splines in an attempt to minimise the unwanted distortions that splines introduce (e.g. shrinkage in corners and the sharp turns found in handwriting) and to minimise cursor/refresh latency when users are drawing. Jitter is reduced by using user-tweakable Euclidean and Manhattan distance thresholds, and by performing an optional smoothing pass (5-point weighted average) on the screen-space coordinates before converting to 3D-coordinates when the user stops drawing. Each stroke point is stored as a three-dimensional vertex (the z-coordinate is unused/ignored in 2D views). Stylus pressure values are used to modulate the thickness of the stroke at each point. Strokes are rendered using simple OpenGL lines; the quality of these lines depends on the graphics drivers present on the user's machine, but is sufficient for most of the original use cases.

Unlike in many other 2D drawing tools, stroke appearance properties (i.e. colour, opacity, thickness) are set on a per-layer basis instead of per-stroke. The advantage of this approach is that it makes it easier for users to quickly change the appearance of a set of strokes; this is great for the original use cases where each layer was intended to correspond to a different set of notes. However, the consequence of this is that additional layers are needed for each additional colour/shade needed. In complex scenes with many characters/props and/or different colours used, artists reported having trouble managing all the layers required. Allowing each layer to use an indexed color palette (as found in TVPaint) may help make the number of layers required more manageable.

Since its introduction, Grease Pencil has become an indispensable part of many artists' workflows, sometimes in ways that we did not anticipate. For instance, Grease Pencil is often used by add-on authors for developing tools requiring sketch based input, as the extensive Python API access to stroke data means that developers do not need to build their own stroke input subsystems. Popular examples of modelling tools relying on Grease Pencil include BSurfaces, GStretch, and Grease Scatter.

4 Grease Pencil 2.0 – Storytelling Tools

A promising application of Grease Pencil is for creating animatics or "moving storyboards" (see Figure 2 for an example). We worked in close collaboration with artists experimenting with the existing tools to develop features which streamline the animation experience. Features introduced include the stroke editing tools (such as selection, transformation, and duplication tools), improvements to the onion-skinning support (by supporting nicer visualisation of before/after states), and by implementing support for more stroke styles (notably filled strokes, and "volumetric" effects).

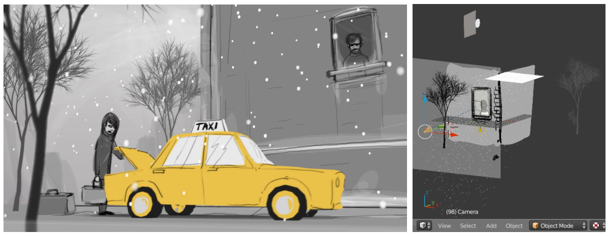

Figure 2: *An example from an animatic for the short film, "Bye" by Daniel M. Lara, where 3D models (trees) and particle effects (snow) are used alongside hand drawn elements (everything else). Multiple shots were animated and sequenced in a single file.*

4.1 Stroke Editing Tools

The ability to edit strokes after drawing them was the most important change in the improved toolset. Specifically, we adapted familiar selection and editing tools to let users select the individual stroke points and manipulate/transform/copy these as if they were the vertices of a mesh. This functionality was requested by storyboard artists as a time-saving tool when animating, considering that most of the time, characters do not change substantially from frame to frame. Thus, the ability to reuse and repurpose existing sketches (or part thereof) is very useful. For example, the jumping movement shown in Figure 1c (see the ghosted frames) was animated by distorting the basic character, using "proportional editing" tools similar to those shown in Figure 3. Users also quickly became fond of the ability to quickly nudge stray points or wonky curves into shape without redrawing the entire stroke.

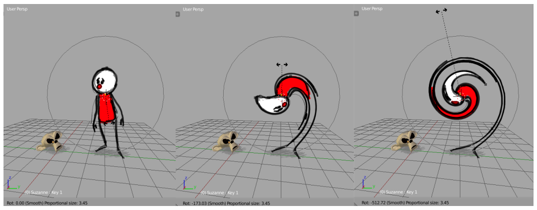

Figure 3: *Soft brush rotate tool can be used to deform strokes for subtle effects (as in Figure 1), or extreme psychedelic effects (as shown here).*

A notable challenge encountered while implementing these tools was that point densities are higher for strokes than for meshes. This meant that care is needed to prevent having the shape of the strokes obscured by all the vertex indicators. The current solution is to only make selected points obvious; unselected points on a stroke with some selected points are also drawn, but much smaller.

4.2 Fills and Volumetric Strokes

Filled strokes (Figure 4a) are implemented using `GL_POLYGON`. This implementation works for non-planar strokes, but can only handle convex shapes as concave regions get erroneously shaded. Despite these limitations, artist feedback was generally positive, noting that this implementation worked well for filling in character shapes and that by playing with the fill opacity, it is possible to achieve two-tone fill effects using a single layer.

Volumetric strokes (Figure 4b,c) are implemented by drawing screen aligned discs on each stroke point only. The effect is most effective using opacity < 0.8, and closely spaced stroke points with regular spacing between them.

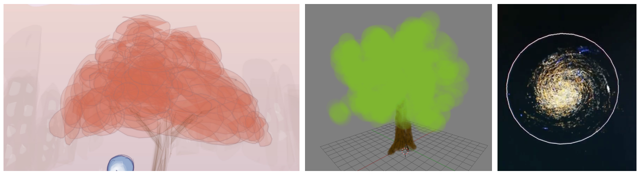

Figure 4: *Examples of filled strokes and volumetric strokes. a) Filled strokes, b) Foliage rendered using volumetric strokes, c) Cosmic effect using volumetric strokes and rotation softbrush*

5 User Feedback and Future Work

Overall, artist feedback of these tools has been largely positive. In particular, artists really enjoyed the "versatility to use the tool for several tasks: notes, corrections, sketching ideas, 2D animations, storyboarding, etc." and the ability to combine the "spontaneity of working on ideas in 2D, with the powerful benefits of enhancing them in a 3D environment using camera movements, fx, 3D objects, and the ability to edit the drawings like meshes".

Artists also found that compared to a traditional 2D workflow for producing story reels, the Grease Pencil only workflow allowed them to create in half a day what would have taken at least twice as long [Roosendaal and Mendiola 2015], noting that "this is particularly true on sequences that requires a lot of camera movements that otherwise are so difficult to achieve with the traditional workflow".

There are many exciting directions for future developments of the Grease Pencil toolset to address the main limitations artists currently face.

- **Rendering** – One of the improvements most frequently requested by artists is to improve the line rendering quality (e.g. allowing smoother and thicker lines), and to provide more artistic styling options (e.g. different brush types). Of particular interest is investigating how strokes can be automatically integrated with the Freestyle [Grabli et al. 2004] offline NPR renderer, for creating art directable and user-programmable 3D line effects. Strokes can already be manually converted to curve geometry and rendered, but this is time consuming (i.e. it must be done frame by frame).
- **Rigging** – Another much requested feature is the ability to attach strokes to 3D objects (invisible helpers or geometry), making the strokes follow the transforms (e.g. keyframed, from parenting/constraints, or simulated) of those objects.
- **Instancing** – It would also be interesting to use instancing or particle system tools to populate the scene with many copies of the same hand-drawn objects (e.g. for generating hand-drawn forests).

Acknowledgements

We would like to thank the Blender community for embracing the Grease Pencil tools and providing valuable feedback. Particular thanks go to Matias Mendiola for the storyboarding tests, and Matt Ebb, Campbell Barton, and Ton Roosendaal for help and support. The first author is funded by a UC Doctoral Scholarship.

References

AMIDI, A., 2011. Early CG Experiments by John Lasseter and Glen Keane.

BAE, S.-H., BALAKRISHNAN, R., AND SINGH, K. 2008. ILoveSketch: As-natural-as-possible sketching system for creating 3D curve models. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, ACM, New York, NY, USA, UIST '08, 151–160.

BOURGUIGNON, D., CANI, M.-P., AND DRETTAKIS, G. 2001. Drawing for illustration and annotation in 3D. *Computer Graphics Forum 20*, 3, 114–122.

COOPER, D. 2002. 2D/3D Hybrid character animation on "Spirit". In *ACM SIGGRAPH 2002 Conference Abstracts and Applications*, ACM, New York, NY, USA, SIGGRAPH '02, 133–133.

DISNEY. 1983. Experimenting with Computer Generated Graphics. In *Disney Newsreel*, vol. 12. June 10.

DISNEY, 1991. The Making of Beauty and the Beast. Documentary.

DORSEY, J., XU, S., SMEDRESMAN, G., RUSHMEIER, H., AND MCMILLAN, L. 2007. The Mental Canvas: A tool for conceptual architectural design and analysis. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, Washington, DC, USA, PG '07, 201–210.

GRABLI, S., TURQUIN, E., DURAND, F., AND SILLION, F. 2004. Programmable style for NPR line drawing. In *Rendering Techniques 2004 (Eurographics Symposium on Rendering)*, ACM Press.

HARVILL, A. 2007. Effective toon-style rendering control using scalar fields. In *ACM SIGGRAPH 2007 Sketches*, ACM, New York, NY, USA, SIGGRAPH '07.

MEIER, B. J. 1996. Painterly rendering for animation. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '96, 477–484.

ROOSENDAAL, T., AND MENDIOLA, M., 2015. A new storyboard workflow. URL: `https://gooseberry.blender.org/blender-2-73-a-new-storyboard-workflow/`, January.

SCHMID, J., SENN, M. S., GROSS, M., AND SUMNER, R. W. 2011. Overcoat: An implicit canvas for 3D painting. In *ACM SIGGRAPH 2011 Papers*, ACM, New York, NY, USA, SIGGRAPH '11, 28:1–28:10.

SUN, M., WANG, Z., ZHAO, L., AND REN, G. 2012. GPU based 3D chinese painting animation. In *Software Engineering and Knowledge Engineering: Theory and Practice*, W. Zhang, Ed., vol. 162 of *Advances in Intelligent and Soft Computing*. Springer Berlin Heidelberg, 763–771.

WHITED, B., DANIELS, E., KASCHALK, M., OSBORNE, P., AND ODERMATT, K. 2012. Computer-assisted animation of line and paint in Disney's Paperman. In *ACM SIGGRAPH 2012 Talks*, ACM, New York, NY, USA, SIGGRAPH '12, 19:1–19:1.

WINKENBACH, G., AND SALESIN, D. H. 1994. Computer-generated pen-and-ink illustration. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '94, 91–100.