Master of Science in Informatics at Grenoble
Master Informatique
Specialization Graphics, Vision and Robotics

Procedural Stylization

Isnel Maxime

June 2019

Research project performed at YOUR LAB

Under the supervision of:
Your Supervisor

Defended before a jury composed of:
Head of the jury
Jury member 1
Jury member 2

June 2019

Abstract

Your abstract goes here...

Acknowledgement

I would like to express my sincere gratitude to .. for his invaluable assistance and comments in reviewing this report... Good luck :)

Résumé

Your abstract in French goes here...

Contents

— 1 —

Introduction

1.1 Background

1.2 Problem Statement

The main problem of stylizing a 3D object in an animation is the *temporal coherence*. The effect given by the stylization has to be kept if the object is moving, rotating and scaling. Many research has been done to solve this problem of *temporal coherence* [29, 7, 3]. we separate this problem is three sections inspired by previous work[24, 11, 5]:

1.2.1 Flatness

The impression of drawing on a flat surface gives the *flatness*. The stylization has a good *flatness* if the image rendered has a good 2D appearance. In order to keep this effect, the size and the distribution of the marks of your stylization have to be independent of the distance between the stylized object and the camera.

1.2.2 Motion Coherence

Motion coherence is a correlation between the motion of marks and the motion of the 3D object. Bad *Motion coherence* will give the impression to see the scene through a semi-transparent layer of marks, this is called *shower door* effect [24], an example to illustrate what happens when there is a bad *Motion coherence* is the movie *Loving Vincent*[2]. The goal is to provide in 2D screen space a perceptual impression of motion as close as possible to the 3D displacement in object space.

1.2.3 Temporal continuity

Temporal continuity is the quality of minimizing changes from frame to frame to ensure fluid animations. In order to have good *temporal continuity*, the marks of the image have to fade slowly during the animation. Human perception is very sensitive to *temporal incoherence* according to some perceptual studies[30, 27].

1.2.4 Prcedural Noises

— 2 —
Previous Work

Image stylization has been around for years. Researchers first start to stylize images [22, 16, 28, 31, 19, 23, 22, 18] in order to have non-photorealistic images. Then they tried to stylize video[21, 22, 18, 4] some of them use the advantage to have the motion flow to improve the *temporal coherence*. In our approach, we want to stylize 3D objects. The advantages of it are that we have more information (like the position of each vertex, the normals, the distance from the camera, ...) about the scene than just an image or a video. In our approach, the goal is to make stylized rendering of 3D objects. There are two moments in a pipeline rendering when we can stylize an object, the first is when we manipulate the vertices and the color of each triangle it is the *object space*. The second is when we do the compositing with the textures that we have like shadow map, image filter, ... (manipulation of pixels of the screen) it is the *image space* and also called *screen space*

2.1 Object Space

Texture-based methods

One of the most used ways to colored object in 3D is the *texture mapping* [**?**]. It consists to add information to each vertex of the 3D object. These information many times are 2D coordinates that correspond to the position of a pixel in a 2D texture. This technique is very used in video games because it is easy to implement, it can be implemented for GPU and it needs low computation. Cel-shading, toon art mapping, gooch shading and others[8] are texture based rendering in object space[26, 17, 7, 6, 13] which are used to stylize scene. As said by Bénard et *al.* [7] textures naturally ensure *motion coherence* and *temporal continuity*. Indeed because each vertex has his color and so the color in moving with the object but gives a bad *flatness* because if the object gets bigger and bigger, pixelization will appear. In order to solve this problem, some[17, 7] tries to use mipmaps (combining multiple scales of textures) to improve *flatness*. Bénard et *al.*[6] use the same principle but with procedural textures. They create multiple noises with different frequency and combine them playing with transparency. Moreover, they overlap the noise to make an impression of infinite zoom effect (like in this example: ShaderToy). With this method patterns of the texture have an almost constant size regardless of the size of the object but it can create small problems of *temporal continuity*. In our method, we will use this technique of fractalization of a procedural noise.

Mark based methods

The natural way to stylize 3D objects is to as an artist apply paint strokes on the object.

These paint strokes can be represented with smalls images also called splats. Daniels[12] and Schmid[**?**] propose to project splats composed of stroke and stored them on the geometry of the model but this technique is expensive in term of storage. Some works [24, 1, 10](more in the state of the art [8]) use point distribution in order to make anchor points for splats. These point distributions are often computed in image space and then are projected on the model. Anchor these splats to the model improve the *motion coherence* because each splat will follow the motion of the 3D model. These splats are rendered in the image space as a 2D sprites so preserved the *flatness*. The problem is how to have the point distribution and how can we control it in order to have a uniform, not too sparse and not too dense distribution. Moreover, these point distribution does not provide control over the *temporal continuity*. In our method, we use procedural noise to anchor the splats.

2.2 Image space

Texture-based methods

Many methods to stylize in image space used texture based approaches. It consists to apply the texture to the entire image [8] but in the case of stylizing animated scenes, the problem is how do we deform the texture to minimize the apparition of sliding artefacts. We can distinguish two families of approaches to solve this problem. The first family of approaches use an approximation of the 3D camera motion with 2D transformations of the texture[11]. This gives a nice trade-off between *motion coherence* and *flatness* but it is limited to static scenes and a set of few camera motions. Moreover, sliding artefacts still occur with strong parallax so Fung et al.[14] and Breslav et al.[5] improve the approximation of the scene motion in order to reduce sliding artefacts.

The second family of approaches use non-rigid deformations to animate the texture[4]. These deformations are computed from the optical flow of a video. This is an extension of the methods used in vector field visualization by Neyret[25]. These deformations can distort the texture and alter the original pattern. The method of Bousseau et al.[4] is very effective with stochastic textures as the fractalization process but creates artefacts with structured patterns.

Mark based methods

A method very used to stylize in image space consists to draw strokes/splats at some place of the image[3, 29, 9, 31, 15]. The question of these mark based method is where do we place the marks in order to have a stylized rendering without losing the meaning of the scene. A first approach is to extract lines that are relevant like the silhouettes, etc. [29, 15, 20] and then stylize the image with this information, like keeping only the extracted lines and change the shape of each line or apply strokes along these lines as Vergne et al.[29] did try to have a good *temporal coherence*. The problem of these techniques is the popping marks due to a bad *temporal continuity*. A second approach is to segment the image in order to have the different parts of the scene[31, 21]. Thanks to this segmentation, they apply different strokes for each part of the image with the corresponding colors. The work of Lin et al.[21] is about videos so they use the optical flow of the videos in order to have a good *temporal coherence*. These mark based methods have a good impression of *flatness* thanks to the splatting in image space, this is something that we will use in our approach.

— 3 —
Realisation

3.1 Overview

3.2 Procedural noise and fractalization

3.3 Splatting

3.4 Stylization

— 4 —
Practical implementation

Bibliography

[1] *NPAR '00: Proceedings of the 1st International Symposium on Non-photorealistic Animation and Rendering*, New York, NY, USA, 2000. ACM. ACM Order No: 434000.

[2] Loving vincent, 2017.

[3] Alexandre Bléron, Romain Vergne, Thomas Hurtut, and Joëlle Thollot. Motion-coherent stylization with screen-space image filters. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering - Expressive '18*, pages 1–13, Victoria, British Columbia, Canada, 2018. ACM Press.

[4] Adrien Bousseau, Fabrice Neyret, Joëlle Thollot, and David Salesin. Video watercolorization using bidirectional texture advection. *ACM Transactions on Graphics*, 26(3):104, July 2007.

[5] Simon Breslav, Karol Szerszen, Lee Markosian, Pascal Barla, and Joëlle Thollot. Dynamic 2d Patterns for Shading 3d Scenes. page 6.

[6] P. Bénard, A. Lagae, P. Vangorp, S. Lefebvre, G. Drettakis, and J. Thollot. A Dynamic Noise Primitive for Coherent Stylization. *Computer Graphics Forum*, 29(4):1497–1506, August 2010.

[7] Pierre Bénard, Adrien Bousseau, and Joëlle Thollot. Dynamic solid textures for real-time coherent stylization. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games - I3D '09*, page 121, Boston, Massachusetts, 2009. ACM Press.

[8] Pierre Bénard, Adrien Bousseau, and Joëlle Thollot. State-of-the-Art Report on Temporal Coherence for Stylized Animations. *Computer Graphics Forum*, 30(8):2367–2386, December 2011.

[9] Pierre Bénard, Jingwan Lu, Forrester Cole, Adam Finkelstein, and Joëlle Thollot. Active Strokes: Coherent Line Stylization for Animated 3d Models. page 10.

[10] Ming-Te Chi and Tong-Yee Lee. Stylized and abstract painterly rendering system using a multiscale segmented sphere hierarchy. *IEEE transactions on visualization and computer graphics*, 12(1):61–72, February 2006.

[11] Matthieu Cunzi, Joelle Thollot, Jean-Dominique Gascuel, Sylvain Paris, and Gilles Debunne. Dynamic Canvas for Non-Photorealistic Walkthroughs. page 10.

[12] Eric Daniels. Deep canvas in disney's tarzan. In *ACM SIGGRAPH 99 Electronic Art and Animation Catalog*, SIGGRAPH '99, pages 124–, New York, NY, USA, 1999. ACM.

[13] Bert Freudenberg, Maic Masuch, and Thomas Strothotte. Walk-Through Illustrations: Frame-Coherent Pen-and-Ink Style in a Game Engine. *Computer Graphics Forum*, 20(3):184–192, 2001.

[14] Jennifer Fung and Oleg Veryovka. "Pen-and-ink textures for real-time rendering". page 6.

[15] Stéphane Grabli, Emmanuel Turquin, Frédo Durand, and François X. Sillion. Programmable rendering of line drawing from 3d scenes. *ACM Transactions on Graphics*, 29(2):1–20, March 2010.

[16] James Hays and Irfan Essa. Image and video based painterly animation. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering - NPAR '04*, page 113, Annecy, France, 2004. ACM Press.

[17] Allison W. Klein, Wilmot Li, Michael M. Kazhdan, Wagner T. Corrêa, Adam Finkelstein, and Thomas A. Funkhouser. Non-photorealistic virtual environments. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, pages 527–534, Not Known, 2000. ACM Press.

[18] J. E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg. State of the "Art": A Taxonomy of Artistic Stylization Techniques for Images and Video. *IEEE Transactions on Visualization and Computer Graphics*, 19(5):866–885, May 2013.

[19] Jan Eric Kyprianidis, Henry Kang, and Jürgen Döllner. Image and Video Abstraction by Anisotropic Kuwahara Filtering. *Computer Graphics Forum*, 28(7):1955–1963, 2009.

[20] Yunjin Lee, Lee Markosian, Seungyong Lee, and John F Hughes. Line drawings via abstracted shading. page 5.

[21] Liang Lin, Kun Zeng, Yizhou Wang, Ying-Qing Xu, and Song-Chun Zhu. Video Stylization: Painterly Rendering and Optimization with Content Extraction. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, page 13.

[22] Peter Litwinowicz. Processing images and video for an impressionist effect. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*, pages 407–414, Not Known, 1997. ACM Press.

[23] Jingwan Lu, Pedro V. Sander, and Adam Finkelstein. Interactive painterly stylization of images, videos and 3d animations. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games - I3D 10*, page 127, Washington, D.C., 2010. ACM Press.

[24] Barbara J. Meier. Painterly rendering for animation. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, pages 477–484, Not Known, 1996. ACM Press.

[25] Fabrice Neyret. iMAGIS-GRAVIR† / IMAG-INRIA. page 8.

[26] Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. Real-time hatching. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, page 581, Not Known, 2001. ACM Press.

[27] Michael Schwarz and Marc Stamminger. On predicting visual popping in dynamic scenes. In *Proceedings of the 6th Symposium on Applied Perception in Graphics and Visualization*, APGV '09, pages 93–100, New York, NY, USA, 2009. ACM.

[28] David Vanderhaeghe and John Collomosse. Stroke Based Painterly Rendering. In Paul Rosin and John Collomosse, editors, *Image and Video-Based Artistic Stylisation*, volume 42, pages 3–21. Springer London, London, 2013.

[29] Romain Vergne, David Vanderhaeghe, Jiazhou Chen, Pascal Barla, Xavier Granier, and Christophe Schlick. Implicit Brushes for Stylized Line-based Rendering. *Computer Graphics Forum*, 30(2):513–522, April 2011.

[30] Steven Yantis and John Jonides. Abrupt visual onsets and selective attention: Evidence from visual search. *Journal of experimental psychology. Human perception and performance*, 10:601–21, 11 1984.

[31] Kun Zeng, Mingtian Zhao, Caiming Xiong, and Song-Chun Zhu. From image parsing to painterly rendering. *ACM Transactions on Graphics*, 29(1):1–11, December 2009.