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 have been done to solve this problem of *temporal coherence* [19, 4, 1]. This problem is three sections:

1.2.1 Flatness

The impression of drawing on a flat surface gives the *flatness*. The stylization has a good *flatness* is the image rendered has a good 2D appearence. In order to keep this effect the size and the distribution of the marks of your stylization has to be independant to 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 semi-transparent layer of marks, this is called *shower door* effect [14].

1.2.3 Temporal continuity

Work in progress

1.2.4 Procedural textures

Work in progress

— 2 —
Previous Work

Image stylization has been around for years. Researchers first start to stylize images [12, 7, 18, 20, 10, 13, 12, 9] in order to have non-photorealistic images. Then they tried to stylize video[11, 12, 9, 2] some of them use the advantage to have the motion flow to improve the *temporal coherence*. In our approach, we want to stylize 3D object. The advantages of it is that we have more information (like the position of each vertices, 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

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 simple to implement, it can be implemented for GPU and it needs low computation. Cel-shading, toon art mapping , gooch shading and others[5] are texture based rendering in object space[16, 8, 4, 3, 6] which are used to stylize scene. As said by Bénard et *al.* [4] 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[8, 4] tries to use mipmaps (combining multiple scales of textures) to improve *flatness*. Bénard et *al.*[3] 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 problem of *temporal continuity*. In our we will use this technique of fractalization of a procedural noise.

As in real painting, some techniques to stylize is to draw elements often they are strokes and sometimes they are dots and with the convolution of dots it creates lines. Overcoat[17] choose to draw strokes on a 3D model, this is an interactive software to help artist. It has 3

tools, the hair tool that permit to draw starting in same direction of the normal at any of point of the surface, the feather tool that works the same but with the tangent of the surface and the level set tool that permit to draw at a certain distance of the object but keeping the curvature of the surface. This technique has a good *temporal continuity* and on the results they present it seems to have a good *motion coherence* and a good *flatness* but these results were do manually by artist.

The problem of the *texture mapping* in object space is that it gives a bad *flatness* due to the distance of the camera of each vertex.

2.2 Image Space

We can also obtain stylized images by manipulating only its pixels. In MNPR[15] a framework for real-time expressive non-photorealistic rendering they use procedural noise to modify the density of pigments on their images to make an effect of rendering on a real paper sheet. Bousseau et *al.*[2] use bidirectional texture advection in order to make a watercolor style.

Bibliography

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Santiago E. Montesdeoca, Hock Soon Seah, Amir Semmo, Pierre Bénard, Romain Vergne, Joëlle Thollot, and Davide Benvenuti. MNPR: a framework for real-time expressive non-photorealistic rendering of 3d computer graphics. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering - Expressive '18*, pages 1–11, Victoria, British Columbia, Canada, 2018. ACM Press.

[16] 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.

[17] Johannes Schmid, Martin Sebastian Senn, Markus Gross, and Robert W. Sumner. OverCoat: an implicit canvas for 3d painting. In *ACM SIGGRAPH 2011 papers on - SIGGRAPH '11*, page 1, Vancouver, British Columbia, Canada, 2011. ACM Press.

[18] 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.

[19] 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.

[20] 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.