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* [10, 4, 2]. 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 [8].

1.2.3 Temporal continuity

1.2.4 Procedural textures

— 2 —
Previous Work

Image stylization has been around for years. Algorithms were created to automatise this desire to stylize. Some techniques use line extraction algorithm to then use convolution of points to make hand drawing styles. Hertzmann with his *curve stroke* algorithm [9] succeed to create images that look like a traditional painting with paintbrushes. To do so he computes many control pint on the original image to further place strokes. But these create a problem when we wanted to stylize videos because it treats frames independantly and so it creates bad *motion continuity*. The movie *Loving Vincent*[1] can illustrate what can happen in this case of bad *motion continuity*.

Then some researches have be to propose a solution to this issue[7, 5, 3, 6]. The solution of Lin et *al.* [6] is to create a segmentation manually of each key frame and then for each part of this segmentation they compute the motion. With this motion they adapt the stroke based rendering of the next frames. To have a watercolor stylization on a video Bousseau et *al.* compute a texture advection to apply to the final image the wanted effect.

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 way to colored object in 3D is the *texture mapping* [**?**]. There are many mapping functions: flat mapping, cylindrical mapping, spherical mapping, cube mapping and the most common used the UV mapping. In UV mapping, for each vertex of the 3D object, there is a vector of texture coordinates (also called UV coordinates) that correspond to the position of a pixel in your texture (usually a 2D image). This pixel will gives the color to display for this vertex. This technique is very used in video games because it is simple to implement, it can be implement for GPU and it needs low computation. But the problem of this approach in stylization is because the mapping is done in object space we have this effect of depth on the scene and this impression that the object is in 3 dimensions. This gives a bad *flatness*. On the other hand, this way of stylizing gives a good *temporal continuity* [4] because every vertex has his color in the texture so when the object in moving the colors displayed move in the same way. They also use an another technique to improve the *temporal continuity* Bénard et *al.* [4]

use procedural fractal noises as textures that he adapts with the distance from the camera. This fractal noise can create an effect of inifinite zoom (like in this example: ShaderToy), it can be helpful in stylization to avoid problem of pixellisation when the scale of the rendered object is increase or decreased. These two approaches are used in our work in order to get a better *temporal continuity*.

2.2 Image Space

Bibliography

[1] Loving vincent, 2017.

[2] 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.

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

[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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.