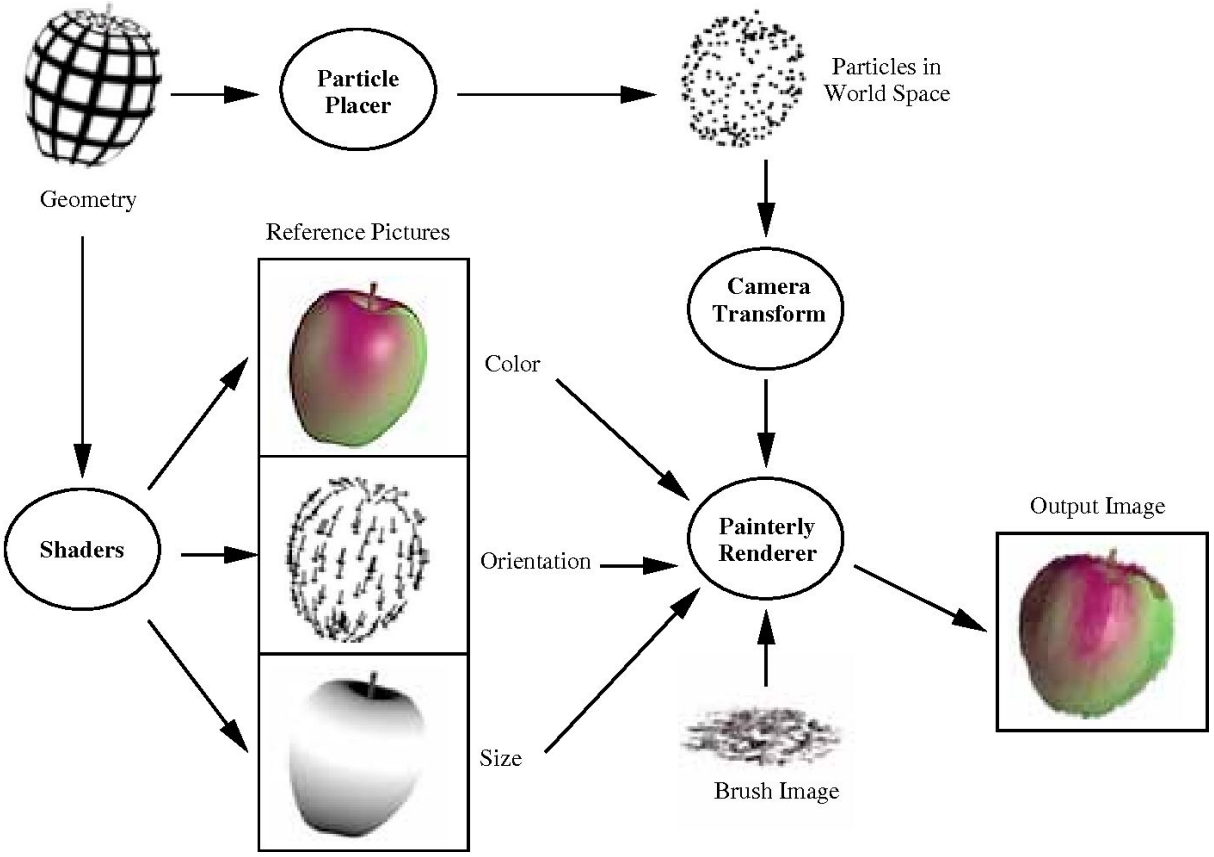
Programmable brushes

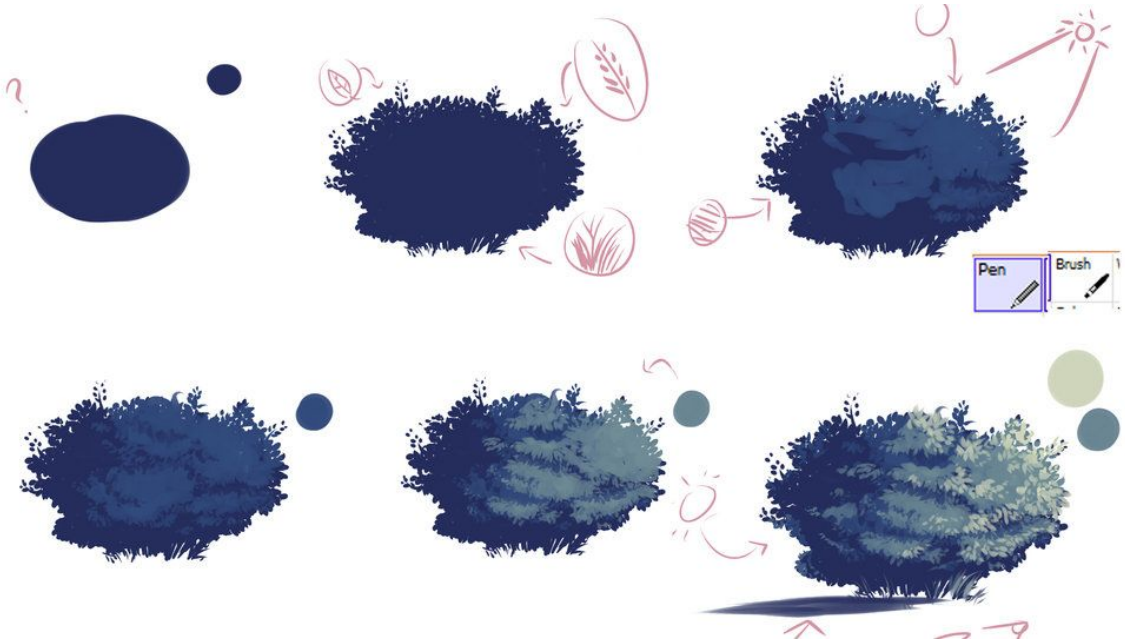
Maverick team at Inria.
Contacts: Joelle.Thollot@inria.fr, Romain.Vergne@inria.fr

Stroke based rendering is a well known approach to produce painterly images (see [1] for a survey). The general idea is to distribute some brush strokes on the input image with attributes computed according to the input and some user parameters. When dealing with 3D scenes, strokes can be distributed either in screen-space or in object space. See for example [2] for a typical painterly rendering pipeline:



In this project we would like to extend this body of work by giving to the user more control and flexibility on the brush strokes behavior so that he could design complex styles. For that we propose to take advantage of the compositing stage of the rendering pipeline where the user has access to various G-buffers (depth map, normal map, shading, etc…). Ideally we would like the user to be able to design a stroke behavior by linking its attributes, position, shape, etc. to any information stored in G-buffers. It raises several complex questions in terms of interface and computation. Based on the student interest several sub-problems can be addressed. Here are some of them:

Knowing the strokes path, how to design its attributes?



Here we suppose that the stroke path is given, either drawn by hand or generated, and we would like to find a meaningful stroke model and an interface allowing the user to design complex stroke appearances according to the input data. The stroke appearance can include its color, texture, width, falloff, randomness, etc…
As in [3], we think that a programmable approach would be interesting as a First step to identify what kind of operators, attributes and scene properties are needed to easily produce complex behaviors.
Inspiration can also be taken in node-based programmable approaches as used in the blackink http://blackink.bleank.com/ software.

How to compute stroke paths?

Complementary to the previous question, we would like to study how to control the strokes distribution and shape. Most previous approaches use either points distributions on top of which a simple shape is rendered, or specific algorithms to optimize a given property.

Here we would like to propose a generic approach that let the user design his own constraints that strokes should follow. For example he should be able to design strokes that orient according to the silhouettes of the input object while maintaining a target density.

A complete system

Finally, if we imagine a full system based on previous questions, new questions arise. Ideally a lot of strokes should be manipulated causing computation time problems. Strokes with various behaviors should be composited which poses some problems in terms of strokes ordering, transparency computation, layering, etc.

Bibliography

[1] Stroke Based Painterly Rendering - David Vanderhaeghe, John Collomosse - https://hal.archives-ouvertes.fr/hal-01342483/document
[2] Painterly Rendering for Animation - Barbara J. Meier - SIGGRAPH 1996
[3] Hugo Loi, Thomas Hurtut, Romain Vergne, Joëlle Thollot. Programmable 2D Arrangements for Element Texture Design. *ACM Transactions on Graphics*, Association for Computing Machinery, 2017, 36 (3), pp.Article No. 27