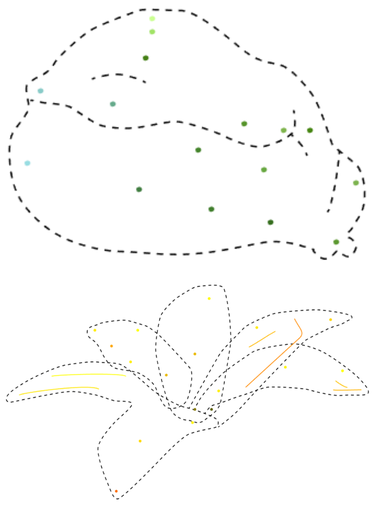
SIGGRAPH2016
Render the Possibilities

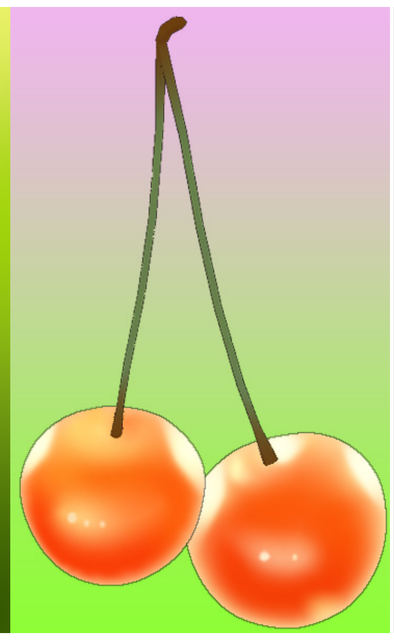
Ray-Traced Diffusion Points

Henrik Lieng

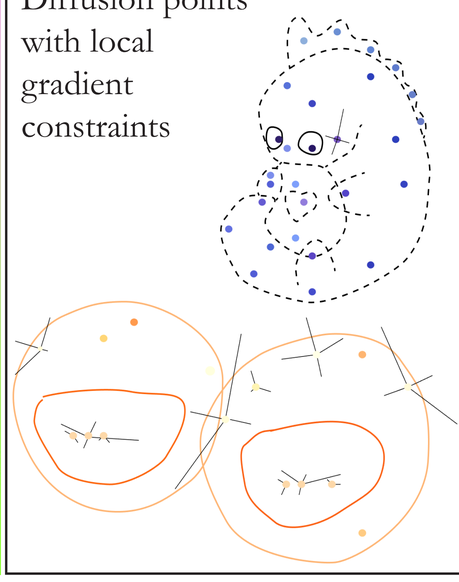
OSLO AND AKERSHUS
UNIVERSITY COLLEGE
OF APPLIED SCIENCES

Diffusion points and curves (selected input)





Diffusion points with local gradient constraints



Problem Statement

Diffusion curves [Orzan et al. 2008] (DCs) has risen as an attractive vector primitive for representing complex colour gradients. While DCs offer an intuitive approach to drawing vector graphics, it is a computationally-expensive primitive to render. A class of recent methods for evaluating DCs eliminates the need for the Laplacian diffusion process by instead employing ray tracing [Bowers et al. 2011, Prévost et al. 2015].

While good performance can be achieved via ray tracing [Prévost et al. 2015], ray-trace-based DCs do not support diffusion points and local colour gradient control, two aspects that are instrumental in diffusion-based DCs [Finch et al. 2011].

In this poster, I present a ray-trace-based DC framework with support for diffusion points and local gradient control.

Formulation

The vector primitive is defined by the following formulation. It defines the colour Z at an arbitrary point p in the image as:

$$Z(p) = \frac{1}{N(p)} \int_{\Omega} w\,(p, p^*)\, C\,(p^*)\, d\theta + \frac{1}{N(p)} \sum_{i=1}^{n} w\,(p, L_i)\, C\,(L_i)\, V\,(p, L_i)\,;$$

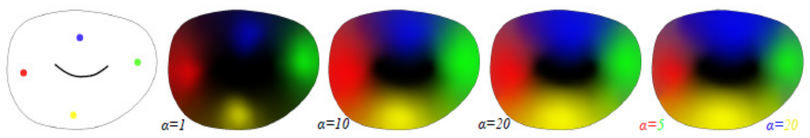
where:

- N is the normalisation term:

$$N(p) = \int_{\Omega} w\,(p, p^*)\, d\theta + \sum_{i=1}^{n} w\,(p, L_i)\, V\,(p, L_i)$$

- L_i represent the geometric coordinates and C the colour coordinates of the i^{th} diffusion point
- V determines the visibility between two points and is computed via ray tracing
- The weighting function w modulates colours according to the distance to the diffusion curve or point. It incorporates a weight α that is used to control the influence of diffusion points relative to diffusion curves:

$$w(p_1, p_2) = \frac{1}{1 + \alpha_{p_2} \|p_1 - p_2\|^2};$$



The integral, which evaluates the contribution from diffusion curves is computed according to the method by Prévost et al. [2015]; that is, via Monte Carlo ray tracing.

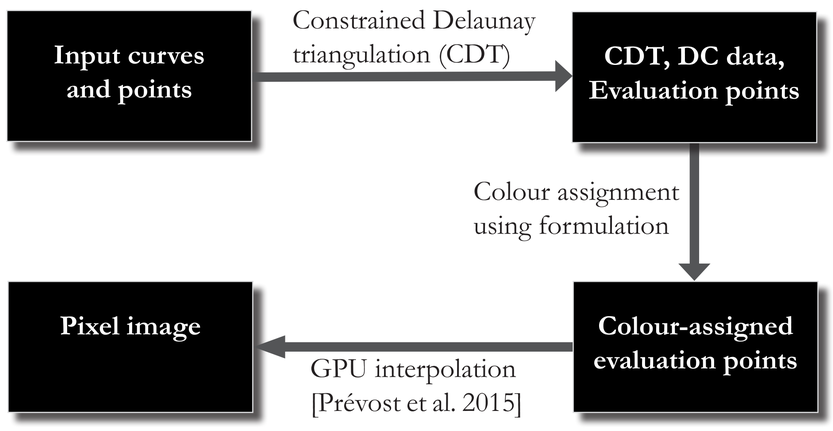
Intuition: the inclusion of points into the Monte Carlo ray tracing procedure it similar to how point lights are treated in the 3D version of path tracing: the point lights are explicitly traced because 'normal' ray are unlikely to hit the lights.

If the image, or layer, is composed of colours from diffusion points only, and not curves, the integral can be eliminated. In this case, the Monte Carlo ray-tracing procedure reduces to standard ray tracing. In my implementation, this decreased the rendering times by 1 order of magnitude (from about 50 ms to 5 ms).

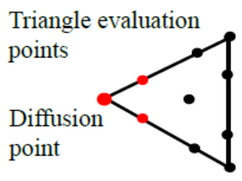
Further, if the domain enclosed by the curves in the image is convex, there is no need to invoke the ray tracing procedure to determine the visibility term. It is therefore suggested that the implementation evaluates the input data before the colour assignment procedure is invoked so that the integral and the visibility term are neglected if possible.

Rendering

The rendering of the primitive follows the following steps:



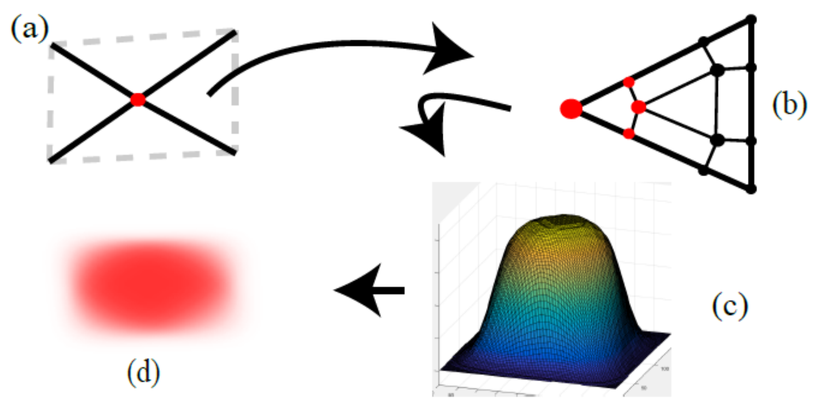
10 evaluation points are associated with each CDT triangle (image below). Colour assignment is performed on these evaluation points instead of individual pixels to increase performance.



Local control

Local gradient constraints can be associated with each diffusion point. These constraints are lines (at least two linearly independent vectors) defined on a plane.

These lines are used to create a local spline surface:



(a) The user can add lines to diffusion points, representing the local propagation of the colour of that point. These lines act as constraints to the CDT (the dashed lines show the added lines in the triangulation).

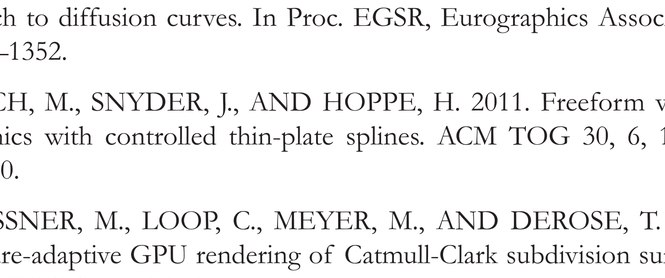
(b) The triangle mesh is subdivided to a denser mesh. Each triangle is subdivided to six quads surrounding a smaller triangle.

(c) The subdivided mesh is rendered as a Catmull-Clark subdivision surface and

(d) defines the local colour gradient at that point. This local colour gradient complements the global formulation using blending, like local diffusion curves by Prévost et al. [2015].

The subdivision surfaces are efficiently rendered on the GPU using Pixar's subdivision library [Niessner et al. 2012].

Discussion and future work



Because the formulation is fundamentally different to bi-Laplacian diffusion, the proposed method possesses different behaviour compared to that of previous methods. Due to the normalisation term, my method possesses the convex hull property in colour space. Thus, the resulting colour gradient does not stray past specified colours. By contrast, bi-Laplacian diffusion does not possess this (maximum principle) property and can therefore stray outside the convex hull of specified colours.

There might be evaluation points that are not visible from diffusion points and curves. These evaluation points will have un-defined colours. This unwanted behaviour does not occur with diffusion-based DCs. I would like to explore ways to modify the ray tracing procedure to deal with this scenario. For example, one could let rays bend to make them influence more than their visible region. After all, these rays are associated with colours, for colour mixing in vector graphics, and not light as in ray tracing for 3D environments. We are therefore not constrained to physical laws as we are with 'normal' ray tracing in 3D.

References

BOWERS, J. C., LEAHEY, J., AND WANG, R. 2011. A ray tracing approach to diffusion curves. In Proc. EGSR, Eurographics Association, 1345–1352.

FINCH, M., SNYDER, J., AND HOPPE, H. 2011. Freeform vector graphics with controlled thin-plate splines. ACM TOG 30, 6, 166:1–166:10.

NIESSNER, M., LOOP, C., MEYER, M., AND DEROSE, T. 2012. Feature-adaptive GPU rendering of Catmull-Clark subdivision surfaces. ACM TOG 31, 1, 6:1–6:11.

ORZAN, A., BOUSSEAU, A., WINNEMÖLLER, H., BARLA, P., THOLLOT, J., AND SALESIN, D. 2008. Diffusion curves: A vector representation for smooth-shaded images. ACM TOG 27,3 (Aug.), 92:1–92:8.

PRÉVOST, R., JAROSZ, W., AND SORKINE-HORNUNG, O. 2015. A vectorial framework for ray traced diffusion curves. CGF 34, 1, 253–264.