Ray tracing diffusion graphics using Optix

Author: Mika Zeilstra
Supervisors : Dr. Amal Parakatt
Prof. Elmar Eisseman

1. Diffusion Graphics

- Vector graphics means resolution independence
- Color spreads out from curves
- Diffusion allows for complex color gradients

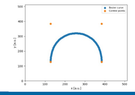

Figure 1 : Cubic Bézier curve with its 4 control points.

2. Research question

- Bowers, Leahey and Wang proposed that diffusion can be approximated as 2D global illumination problem [1]
- *Can dedicated ray tracing hardware be used to achieve real time rendering of diffusion curve graphics ?*

2. Optix

- Optix is an API / Framework for ray tracing using dedicated ray tracing cores on NVIDIA RTX cards
- Diffusion curves will be seen as lights
- We shoot rays in a circle for each pixel to estimate the light it receives, which will be its color

3. Features

- Color interpolation using control points as curve parameter
- Weight, and blur sigma as curve parameters
- Portals and partial transparency
- Endcaps
- Random ray direction and starting point within pixel
- Optix Denoiser

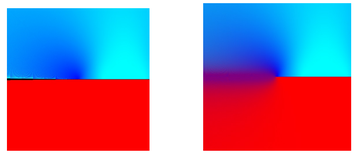

Figure 2 : Endcap example. Left without endcap, Right with endcap.

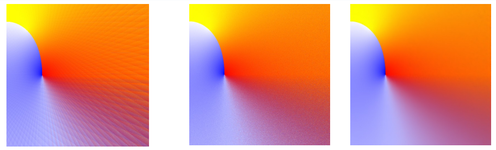

Figure 3 : Artifact removal using ray randomization. From left to right, original, Ray randomization, Denoised image.

4. Quality Comparison

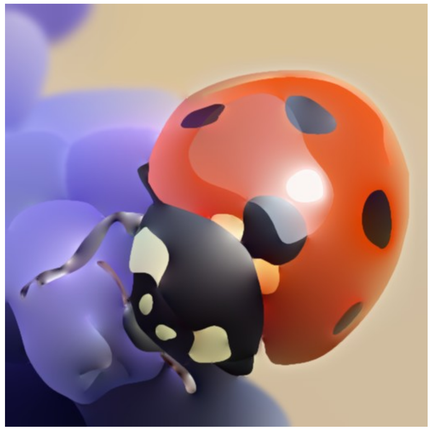

Figure 4a: Reference image created using the original diffusion curve renderer by Orzan et al. [2].

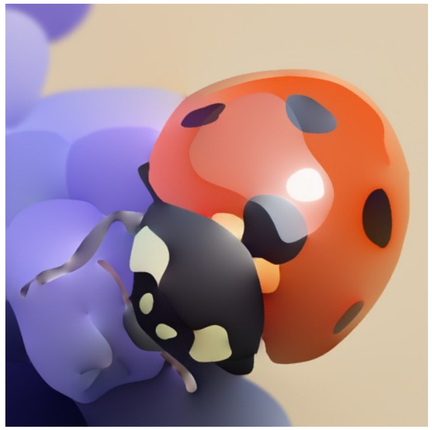

Figure 4b: Image rendered using 128 rays per pixel and all features enabled.

5. Results

- Average time until first frame 1-2 seconds
- Average time per frame for 128 rays 630.9 ms on a Quadro p1000, 98.7 ms on a RTX 2060.
- When accepting more quality loss ,the picture is still good at 8 rays per pixel

Figure 5 : Comparison between Quadro and RTX. Mind the 8x difference.

6. Conclusion

- Significant speedup ~8x for tested images.
- No significant loss of quality
- A lot of extra customization options for artists
- Future work : Implement algorithm by Prévost, Jarosz and Sorkine-Hornung [3]

References

[1] J. C. Bowers, J. Leahey, and R. Wang, "A ray tracing approach to diffusion curves," in Proceedings of the Twenty-Second Eurographics Conference on Rendering, ser. EGSR '11, Prague, CzechRepublic: Eurographics Association, 2011, pp. 1345–1352.doi:10.1111/j.1467-8659.2011.01994.x. [Online]. Available:https://doi.org/10.1111/j.1467-8659.2011.01994.x.

[2]. Orzan, A. Bousseau, H. Winnemöller, P. Barla, J. Thollot, and D. Salesin, "Diffusion curves:A vector representation for smooth-shaded images," inACM Transactions on Graphics (Pro-ceedings of SIGGRAPH 2008), vol. 27, 2008. [Online]. Available:http://maverick.inria.fr/Publications/2008/OBWBTS08.

[3] Romain Prévost, Wojciech Jarosz, and Olga Sorkine-Hornung. A vectorial framework for ray traced diffusion curves. Computer Graphics Forum, 34(1):253–264, 2015

TU Delft