A Study on Rendering Techniques to Visually Represent Sparkles

Frank Fan and Gladimir V. G. Baranoski
Natural Phenomena Simulation Group, David R. Cheriton School of Computer Science,
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Technical Report CS-2024-02

Abstract

Advances in rendering have allowed researchers to produce physically-based and realistic images of surfaces with a sparkly appearance. In this report, we provide a comprehensive survey that examines the phenomenon of sparkles and the different methods used to render them. We then highlight the current limitations of existing works and outline future research opportunities.

I. Introduction

Many materials have a sparkling (or glinty) appearance elicited by bright spots on a dark surround when illuminated by a directional light source such as the sun. As these spots, commonly referred to as "sparkles" (or "glints"), tend to float above the surface they originate from, they convey an ethereal and dazzling sensation. They can be found in a wide variety of man-made and natural materials such as metallic car paints, craft glitter, steel, snow, dew and precious gems (*e.g.*, diamonds). To date, the simulation of their optical effects has not been extensively investigated in computer graphics in comparison with other topics associated with realistic image synthesis.

This report aims to examine how these phenomena can be simulated and rendered using computer graphics techniques. Section II briefly defines the terms "sparkle" and "glint", and it explains how these phenomena occur in a natural environment. Section III presents a literature survey about the state of the art of the simulation and rendering. Section IV concludes the report and provides an overview of current limitations and future investigation avenues. Additional information about related light transport concepts is provided in the accompanying appendix.

II. Phenomena Overview

A. Terminology

A sparkle, as defined by ASTM (American Society for Testing and Materials) [1], is "the aspect of the appearance of a material that seems to emit or reveal tiny bright points of light that are strikingly brighter than their immediate surround and are made more apparent when a minimum of one of the contributors (observer, specimen, light source) is moved." We note that this terminology standard also defines a glint to be the same as a sparkle.

The differentiation between a sparkle and a glint is not clearly established in the existing literature. Some works, such as Wang *et al.* [2], have used the two terms interchangeably (along with glitter, glisten, gleam, and brilliance). Cann [3], on the other hand, reserved the term glint for "broad-beam reflections from curved surfaces" such as icicles or water ripples, and classified sparkles to be the narrow-beam reflections. Most articles, however, choose to exclusively use one of either terms. Yan *et al.* [4], for example, described a glint as the reflection from light sources that subtend a small solid angle. This definition was similar to Cann's [3] for sparkle. Furthermore, the use of either term has not been limited to where they occur or what surface they occur on. Both sparkle [5][6][7] and glint [8][4] terms have been used to describe the phenomenon occurring in paints and snow [8][9][10].

Many publications [5][11][8] referenced two of McCamy's articles [12][13] as works that give complete definitions for these phenomena. McCamy [12] defined the scattering of light from paint flakes to be "brilliance or sparkle." In a later article [13], he defined a glint to be the tiny specular reflection of light that can appear or disappear if "the geometric relationship of the illuminator, object, or observer changes during observation." This definition of glint is similar to that of the definition of sparkle given by ASTM [7].

The lack of existing definitions and clear differentiation for these terms has been explicitly stressed by researchers working in related fields. Kirchner *et al.* [11], for example, noted that it still remains unclear whether these terms refer to the same phenomenon. Furthermore, different articles often contradict each other when attempting to define these terms. Kirchner *et al.* [11] also found that some articles defined a sparkle to be the static reflection of light, while other articles defined it to be the reflection of light turning on and off. Therefore, they proposed definitions for new terms they created to avoid confusion. More precisely, they used McCamy's [13] definition of "glint" in their article to define their term of *Glint Impression* to be "the overall impression of several or many tiny light-spots (*i.e.*, glints) that are strikingly brighter than their surrounding."

In this report, we will focus on sparkles (as defined by ASTM [1]), implicitly including glints by analogy.

B. Physical Background

A sparkle occurs on a material's surface when an elementary area is under direct illumination of a point light source [7]. This tiny area on the surface has isolated microscopic mirrors, and it scatters the light into a single specific direction [7]. A sparkle is typically observed by one of the eyes of an observer (see Fig. 1) [3]. The human's depth perception mechanism is not able to fuse the sparkle images originating from each eye. Thus, it cannot determine the precise location of the sparkle [3]. This, in turn, produces the floating nature (see Fig. 2) of the sparkle, which can be neutralized by closing the unstimulated eye [3]. From a far enough distance, even multiple sparkles may fuse together due to the human eye's finite resolution [5].

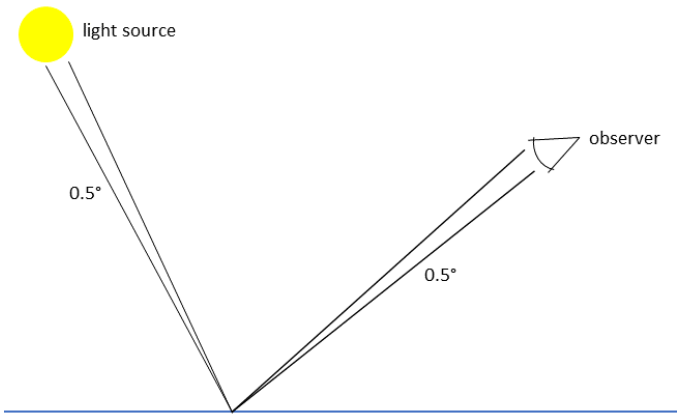

Fig. 1: Sketch depicting a tiny area on a surface reflecting light with beamwidth of 0.5° (based on the diagram from Cann [3]).

III. State of the Art

Naively rendering microstructures and their complex specular reflections produces unwanted noise, and it is inefficient due to each pixel footprint's intricate normal distribution. Achieving a converged result of a highly specular surface and a small light source would therefore require a very large amount of samples.

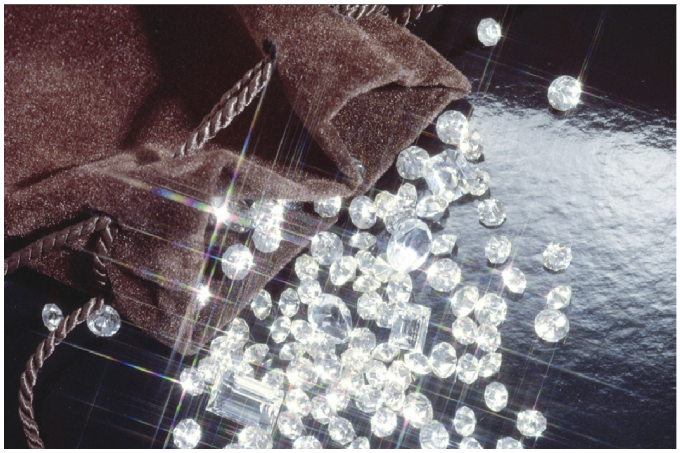

Fig. 2: Example of sparkles on diamonds (redrawn from [3]). The floating nature of the sparkle is observed due to the inability of the human's depth perception mechanism to fuse the two images seen by each eye.

To solve this problem, the microfacet BRDF (bidirectional reflectance distribution function) [14] is generally used to represent these microstructures and to model surfaces with tiny mirror flakes. The microfacet model is related to the works by Cook and Torrance [15] and Walter *et al.* [16]. Within this context, a particular normal distribution function (NDF) is employed to uniformly distribute a finite number of flakes in a pixel footprint (also known as a patch) representing a unit texture space that approximates the region of surface visible through a single pixel. This NDF determines the position and orientation of these microfacets, and it is used to calculate how many of them reflect light from incident direction ω_o to outgoing direction ω_i (see Fig. 3).

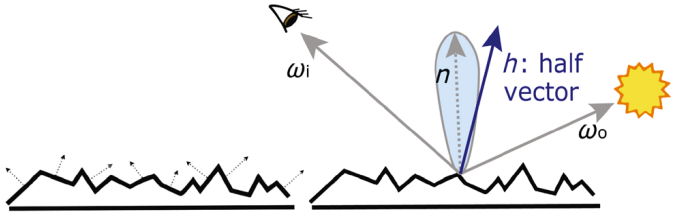

Fig. 3: Diagram illustrating the microfacet theory (redrawn from Zhu *et al.* [17]). **Left:** Each microfacet's orientation and normal are determined by the NDF. **Right:** n is the normal of the microfacet, h is the half vector, sometimes referred to as ω_h defined by incident ω_i and outgoing ω_o directions used to compute the BRDF.

The microfacet BRDF is defined as [15][16]:

$$f_r(\mathbf{x}, \omega_i, \omega_o) = \frac{F(\omega_i \cdot \omega_h)\, D(\mathbf{x}, \omega_h)\, G(\omega_i, \omega_o, \omega_h)}{4\,(\omega_i \cdot \mathbf{n_x})\,(\omega_o \cdot \mathbf{n_x})}, \tag{1}$$

where $\mathbf{x}$ denotes the position, F is the Fresnel reflection coefficient, D is the microfacet distribution, and G is the shadowing-masking term. The Fresnel reflection coefficient is a function of incident angle and wavelength, and it is used to obtain the angular variation of the intensity to describe how light is reflected from the microfacet [15]. The attenuation factor G accounts for the shadowing and masking of one facet by another [15].

Rendering techniques that employ microfacet models focus on the evaluation of a surface patch $\mathcal{P}$. Many methods used the patch as input to calculate the discrete patch-specific NDF ($\mathcal{P}$-NDF) (see Fig. 4). Researchers have explored different approaches to increase the correctness to cost ratio of $\mathcal{P}$-NDF evaluations.

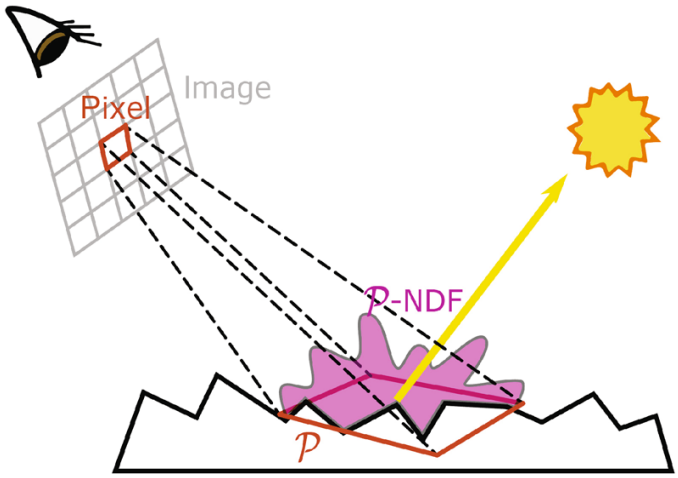

Fig. 4: Diagram of a single patch $\mathcal{P}$ corresponding to a pixel (redrawn from Zhu *et al.* [17]). The patch $\mathcal{P}$ contains many microfacets from which a $\mathcal{P}$-NDF can be calculated.

In fact, the main issues that need to be tackled in the rendering of sparkles are computational cost, physical correctness and the scarcity of supported surface material types. Existing works to alleviate these issues can be organized into two categories: offline rendering and real-time rendering. The offline rendering methods are generally physically-based and produce believable depictions of the specular phenomenon. They can be further subcategorized depending on whether they explicitly or implicitly represent material microstructures. In the remainder of this section, we concisely review representative works belonging to these categories and take a brief look at the detection and representation of sparkles on snow.

A. Offline Rendering

Explicit Representation

Yan *et al.* [4] introduced a framework for explicitly representing a microsurface with a normal map that enabled realistic renderings of sparkly (glinty) appearances. They introduced an algorithm to evaluate the NDF of a surface patch. This was then used to compute the appearance of the microstructure geometry. An advantage of using a normal map is that reflectance values of specific microstructures, such as scratches, bumps or brushes, can be correctly computed. The storage required to use this method, however, is not suitable for real-time rendering. As stated by Zeltner *et al.* [18], such a memory requirement tended to be upwards to tens of gigabytes when running their experiments. Zhu *et al.* [17] stated that the size of the normal map employed in the generation of the images depicted in Fig. 5 was $200K \times 200K$. Similarly, Wang *et al.* [2] claimed that, to avoid obvious repetition, a $10K \times 10K$ normal map was required to render one square centimeter. Therefore, the work of Yan *et al.* [4] seems to be more suitable for offline rendering applications. Yan *et al.* [19] later accelerated computation times using a position-normal distribution method. Additionally, in a subsequent work [20], they extended their method [4] even further to handle wave optics effects.

Several other methods have been explored to reduce storage requirements or improve computation speeds. Chermain *et al.* [22] introduced a method that makes use of microfacet-based normal mapping and multiple-scattering BRDFs together. They aimed to use the multiple-scattering BRDF to address the issue of dark renderings (see Fig. 6) due to singular light bounces and back-facing surfacing normals. Chermain *et al.* [22] also addressed the issue of dark artifacts generated from inefficient normal mapping sampling procedures and high sample weights. To avoid these black fringes, they used a microfacet-based BRDF that fakes normal perturbations. Incidentally, their patch-BRDF can be used in conjunction with multiple importance sampling [23] in Monte Carlo forward path tracing frameworks.

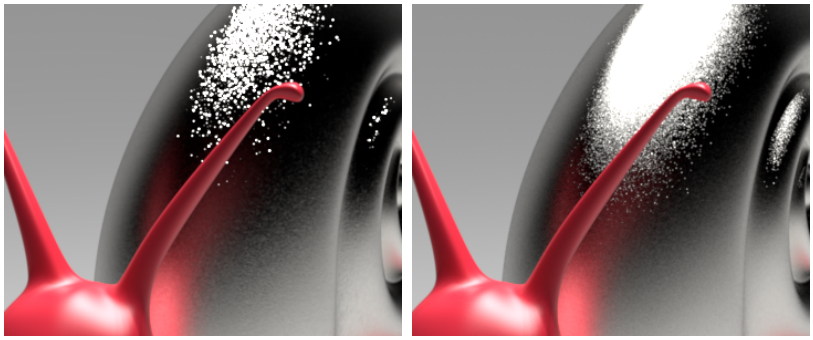

Fig. 5: Comparison of two renderings from the same normal map (redrawn from Yan *et al.* [4]). **Left:** Image rendered via naive sampling. Total render time: 2 hours. **Right:** Image rendered using the method proposed by Yan *et al.* [4]. Total render time: 17 min. Both images were rendered within the Mitsuba framework [21]. The two rendering time calculations were provided by Yan *et al.* [4]. According to Yan *et al.* [4], naive sampling fails to efficiently hit the microfacets with uniform pixel sampling.

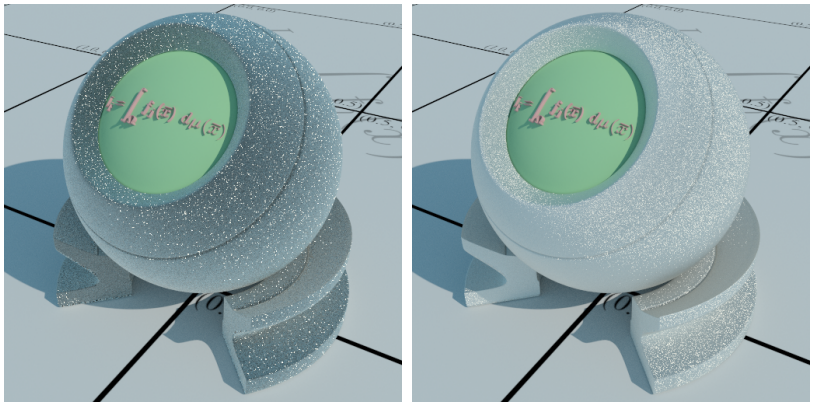

Fig. 6: Comparison of renderings of a glittery orb (redrawn from Chermain *et al.* [22]). **Left:** Image rendered using the method proposed by Yan *et al.* [19]. **Right:** Image rendered using the method proposed by Chermain *et al.* [22] showcasing the removal of the artifacts presented in the image on the left.

Wang *et al.* [24] presented a method that implicitly generated the normal map with range query capability to lower storage costs while also being able to use the method of Yan *et al.* [4]. They generated normal maps by blending patches from input examples, and they were able to render sparkles (glints) from a non-repeating microstructure. Their method, however, still yielded high algorithmic complexity. Deng *et al.* [25] presented a precomputation-based prefiltering approach to achieve constant performance and storage costs. They utilized angular point and range queries to compress NDF images and subsequently render sparkles (glints) in constant time.

Implicit Representation

To avoid high storage costs, Jakob *et al.* [8] designed a procedural model that stochastically generates these microfacets without the need to store or compute any textures. This low storage requirement is possible because they generate the facets and count the number of correctly oriented ones using a stochastic process during traversal. Although they reduced storage requirements, the method traverses through a 4D hierarchy — two dimensions of surface position and two dimensions of normal direction — to find the amount of light reflected in the footprint of each pixel. This results in a high computation cost and therefore, makes the method only suitable for offline rendering. Another limitation to their method is the inability to represent a variety of microstructure features such as scratches or brushes due to the random generation of these flakes. Fig. 7 showcases images generated by Jakob *et al.* [8],

which were rendered using the Mitsuba framework [21]. Their supplementary video also showed how the sparkles (glints) change and fluctuate depending on the view position.

Fig. 7: Images of scenes featuring Jakob *et al.*'s [8] discrete microfacet BRDF (redrawn from Jakob *et al.* [8]). **Left:** red high heels with glittery finish. **Right:** Christmas ornaments with varying model parameters such as particle count, surface roughness and anisotropy.

Atanasov and Koylazov [9] extended the method proposed by Jakob *et al.* [8] by using a different stochastic algorithm to improve computation time. Fig. 8 showcases images presented in their work. However, we note that they did not provide comparisons with images rendered using the method proposed by Jakob *et al.* [8].

Fig. 8: Images of scenes depicting stochastic microfacets (redrawn from Atanasov and Koylazov [9]). **Left (A):** A metal plate with surface consisting of flakes. **Center (B):** A car with glittery paint coat. **Right (C):** Sparkling snow scene.

Wang *et al.* [2] developed an extension of the method proposed by Jakob *et al.* [8]. It reduced computational costs by approximating the 4D hierarchy traversal via two 2D traversals: a spatial domain search and an angular domain search. Additionally, they used prefiltering techniques to increase the performance of the employed global illumination algorithms.

Differentiable regularization

Fan *et al.* [26] (see Fig. 9) brought forward an entirely different solution. They observed that scenes with increased surface roughness or light source size were easier to render and, hence, these scenes were sampled first. They then extrapolated these "easy" samples to obtain the result of the desired light source size and material roughness using differentiable path tracing [27]. Fan *et al.* [26] claimed that their method was the first to use differentiable regularization. Indeed, to the best of our knowledge, their work is the only one in the current literature to do so.

B. Real-Time Rendering

Real-time rendering methods have difficulties to directly use algorithms employed offline methods due to a graphics processing unit (GPU)'s video memory limit and its architectural difference from a central

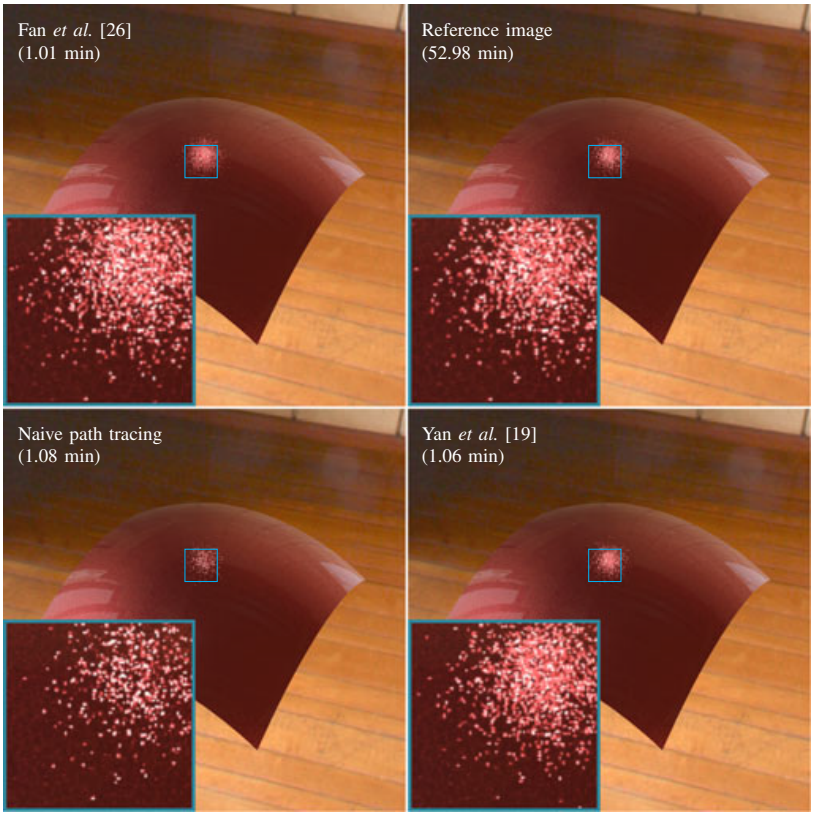

Fig. 9: Comparison of images rendered using different methods (redrawn from Fan *et al.* [26]). **Top left:** Image rendered using the method proposed by Fan *et al.* [26]. **Top right:** Reference image. **Bottom left:** Image rendered via naive path tracing. **Bottom right:** Image rendered using the method proposed by Yan *et al.* [19]. Fan *et al.* [26] claimed that their method produced higher quality results (closer to the reference image) with lower memory cost and render time compared to the method used by Yan *et al.* [19] and naive path tracing.

processing unit (CPU) [17]. Explicit material representations require a large amount of storage, which current GPU architectures cannot support. For this reason, existing works utilize implicitly represented materials to render sparkles (glints).

Zirr and Kaplanyan [28] extended the discrete microfacet method proposed by Jakob *et al.* [8] by introducing a biscale NDF. This avoids explicitly counting each correctly orientated facet. Instead, it uses a statistical average via a binomial law. However, Chermain *et al.* [22] noted that this method is unsuitable for photorealistic rendering (see Fig. 10). Also, Deliot *et al.* [10] stated that it still remains expensive for video game use.

Wang *et al.* [29] proposed another real-time method as an extension of the method proposed by Jakob *et al.* [8]. According to Wang *et al.* [29], while their method was slightly slower than Zirr and Kaplanyan's method [28], it produced results closer to the reference of Jakob *et al.* [8] (see Fig. 11).

Chermain *et al.* [30] presented an approach that featured mipmap structures to calculate the BRDF. A mipmap structure, as defined by Butterfield *et al.* [31], is "a pyramidal structure used in mapping two-dimensional textures." Their approach also used dictionaries of precomputed 1D marginal distributions to account for the normals of the flakes in the pixel footprint. Chermain *et al.* [30] noted that their results converged to Cook and Torrance's [15] microfacet BRDF, which assumes an infinite number of microfacets. Furthermore, their rendering times ranged from 50% to 156% of those using Zirr and Kaplanyan's method [28].

Deliot *et al.* [10] improved on Zirr and Kaplanyan [28]'s method by reducing the runtime of counting the number of facets while still maintaining correctness and accuracy (see Fig. 12).

Fig. 10: Images redrawn from Zirr and Kaplanyan [28]. **Left:** Dress scene rendered with varying microdetail size. **Right:** Sand scene with sunny light conditions

Fig. 11: Images of the same dress scene from Fig. 10 (redrawn from Wang *et al.* [29]). **Main left plate:** Image rendered using the method proposed by Wang *et al.* [29]. **Main center plate:** Image rendered using the method proposed by Jakob *et al.* [8]. **Main right plate:** Image rendered using the method proposed by Zirr and Kaplanyan [28]. **Small top plate:** Close-up of image rendered using the method proposed by Wang *et al.* [29]. **Small middle plate:** Close-up of image rendered using the method proposed by Jakob *et al.* [8]. **Small bottom plate:** Close-up of image rendered using the method proposed by Zirr and Kaplanyan [28]. Wang *et al.* [29] claimed that their method was closer to the reference of the image (rendered using the method proposed by Jakob *et al.* [8]) than Zirr and Kaplanyan's method [28] while slighty slower.

To the best of our knowledge and according to Zhu *et al.* [17], there still does not exist a real-time method that can render sparkly (glinty) appearances while utilizing high-frequency normal maps. This can be attributed to the relatively low video memory capabilities of current GPU architectures.

C. Case Study: Snow Sparkles

Nguyen *et al.* [32] performed a statistical analysis of sparkle in snow images. Although they did not simulate the sparkles with computer graphics methods, their article presented a dataset of digital, *in-situ* obtained snow images, and utilized the method developed by Ferrero *et al.* [33] to analyze them. They concluded that the method that is often used for painted or metallic surfaces can also be used to detect and estimate sparkles on snow surfaces.

Jakob *et al.* [8] briefly mentioned the sparkling effect that occurs in snow along with other natural phenomena such as crystals, rock and frost. Atanasov and Koylazov [34] produced some images with snow (see Fig. 8 and Fig. 13) as a demonstration of their method.

Wang *et al.* [35] aimed to create a flexible real-time sparkle effect suitable for video games and used a snow scene to test their model (see Fig. 14). They used point jittering in 3D grids to model the sparkles. Subsequently, Wang *et al.* [29] also rendered a snow scene (see Fig. 15) to compare their method with those methods presented in previous works [8], [2].

Fig. 12: Image of Christmas scene rendering (redrawn from Deliot *et al.* [10])

Fig. 13: Renderings of two snowy scenes (redrawn from Atanasov and Koylazov [34]).

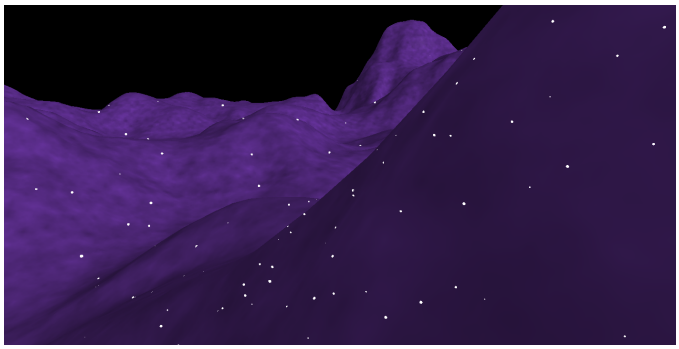

Fig. 14: Real-time rendering of a snowy scene (redrawn from Wang *et al.* [35]).

IV. Conclusion and Research Perspectives

In this section, we provide a brief summary of our observations, followed by an overview of current limitations and directions for future research in this field.

A. Summary

We remark that there is a lack of clear definitions of the terms “sparkle” and “glint” in the literature. While some works made a point to distinguish a sparkle from a glint, most works either used the two terms interchangeably or used one exclusively. For clarity, in this report, we focused on sparkles as defined by ASTM [1].

In order to render sparkles, naive rendering techniques, such as general path tracing [36], will not suffice due to the intricate normal distribution of these microsurfaces. Many samples would be required to achieve a converged and noiseless rendering. To solve this issue, the microfacet BRDF is often used to represent the microstructures and to model these surfaces with tiny mirror flakes. Furthermore, naive

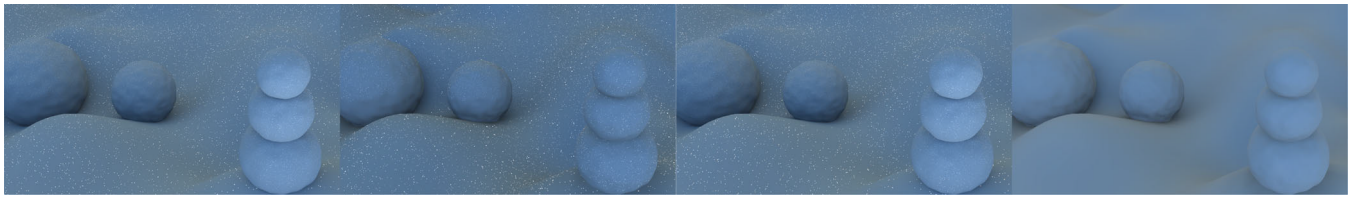

Fig. 15: Comparison of images of a snow scene (redrawn from Wang *et al.* [29]). From left to right: Image rendered with the method proposed by Wang *et al.* [29], total render time: 26.4 ms; Image rendered with the method proposed by Wang *et al.* [2], total render time: 15.5 min; Image rendered with the method proposed by Jakob *et al.* [8], total render time: 20.6 min; Image rendered with the method proposed by Wang *et al.* [29] without sparkles (glints), total render time: 17.1 ms.

rendering brings up the issues of computational cost, physical correctness and the scarcity of supported surface material types.

Existing works to alleviate these issues can be categorized into either offline rendering of real-time rendering. Offline rendering methods can further be subdivided into explicit and implicit representation. Explicit representation usually take the form of normal maps pioneered by Yan *et al.* [4]. The strength of explicit representation is the versatility in representing many different microstructures such as scratches and bumps. However, one common issue with explicit representation is the large amount of storage required to store the microstructure information. Methods that employ implicit representations avoid this storage issue by counting microfacets without actually generating them. However, the different types of microgeometry enabled by the explicit representation approach cannot be explicitly rendered using the implicit representation approach as the microsurface's information is not stored anywhere.

B. Current Limitations

The lack of quantitative evaluation approaches based on direct comparisons with actual depictions (*e.g.*, photos) of the real phenomenon hinders future research advances in this field. Oftentimes, the physical correctness and the quality of the images obtained using a given method are assessed by comparing them to images obtained using either naive techniques or previous methods, such as those proposed by Jakob *et al.* [8] or Yan *et al.* [4][19][20], which may have their own shortcomings. It is worth mentioning that a number of more recent works [18][24][25][26][29] in this area generated their representative images using the Mitsuba framework [21], which was also employed by Jakob *et al.* [8] and Yan *et al.* [4][19][20].

Such evaluation approaches may be sufficient when the goal is to generate believable images while keeping computations costs low. However, if the goal is to generate predictive [37] renderings of the target phenomenon, then the sparkle (glint) simulations' fidelity [38] should be also assessed through direct comparisons with the "real thing".

Lastly, as previously mentioned in this report, both implicit and explicit representation methods have their specific issues. Implicit representations deal with flexibility issues with respect to surface material types. Explicit representations deal with large storage requirements and high computational times. Due to these issues, their integration into real-time frameworks remains difficult.

C. Outlook

Sparkles (glints) still cannot be considered a solved problem in realistic image synthesis, particularly with respect to the predictive rendering of natural materials. Although there have been a few renderings depicting snowy scenes [9][10][29], most existing works tend to focus on metallic paints and man-made surfaces. Cann [3] noted that sparkles are ubiquitous in outdoor settings. According to Nguyen *et al.* [32], the mechanisms responsible for the occurrence of sparkles in snow are the same as those responsible for their occurrence in metallic paints. Assuming that this is really the case, then the rendering of natural scenes depicting sparkles could be conducted using the same algorithms employed for metallic paints. However, such an assumption remains to be fully verified.

Another possibility for future research involves the use of physically-inspired methods as opposed to physically-based. Deliot *et al.* [10] touched on this as they developed a real-time method that, according to them, can produce results comparable to results provided by the methods proposed by Zirr and Kaplanyan [28] and Chermain *et al.* [30] while being 1.5 to 5 times faster. Deliot *et al.* [10] explicitly noted that their method was not physically-based and, thus, did not face the limitations of Chermain *et al.*'s method [30] such as micro-roughness constraints and precomputed NDF dictionary restrictions. Despite their improved performance, Deliot *et al.* [10] still claimed that their shader was computationally costly compared to smooth physically-based renderers. Accordingly, it should be regarded as a stepping stone in integrating sparkling appearances into real-time applications such as video games, virtual reality simulators, animation studios and movie or TV visual effects.

REFERENCES

[1] ASTM, "Standard terminology of appearance," Tech. Rep. E284-17, ASTM (American Society for Testing and Materials), International, West Conshohocken, PA, USA, 2017.

[2] B. Wang, L. Wang, and N. Holzschuch, "Fast global illumination with discrete stochastic microfacets using a filterable model," *Computer Graphics Forum*, vol. 37, no. 7, pp. 55–64, 2018.

[3] A. Cann, "All about sparkles," *Optics and Photonics News*, vol. 24, no. 1, pp. 42–45, 2013.

[4] L-Q. Yan, M. Haan, W. Jakob, J. Lawrence, S. Marschner, and R. Ramamoorthi, "Rendering glints on high-resolution normal-mapped specular surfaces," *ACM Transactions on Computer Systems*, vol. 33, no. 4, pp. 1–9, 2014.

[5] S. Ershov, A. Khodulev, and K. Kolchin, "Simulation of sparkles in metallic paints," in *Proceedings of the International Conference Graphicon*, 1999.

[6] R. urikovi and W. Martens, "Simulation of sparkling and depth effect in paints," in *Proceedings of the 19th Spring Conference on Computer Graphics*. 2003, pp. 193–198, ACM.

[7] A. Ferrero, E. Perales, N. Basic, M. Pastuschek, G. Porrovecchio, A. Schirmacher, J. L. Velázquez, J. Campos, F. M. Martínez-Verdú, M. Šmid, P. Linduska, T. Dauser, and P. Blattner, "Preliminary measurement scales for sparkle and graininess," *Opt. Express*, vol. 29, no. 5, pp. 7589–7600, Mar 2021.

[8] W. Jakob, M. Haan, L-Q. Yan, J. Lawrence, R. Ramamoorthi, and S. Marschner, "Discrete stochastic microfacet models," *ACM Transactions on Computer Systems*, vol. 33, no. 4, pp. 1–10, 2014.

[9] A. Atanasov and V. Koylazov, "A practical stochastic algorithm for rendering mirror-like flakes," in *ACM SIGGRAPH 2016 Talks*. 2016, pp. 1–2, ACM.

[10] T. Deliot and L. Belcour, "Real-time rendering of glinty appearances using distributed binomial laws on anisotropic grids," *Computer Graphics Forum*, 2023.

[11] E. Kirchner, G-J. van den Kieboom, L. Njo, R. Supr, and R. Gottenbos, "Observation of visual texture of metallic and pearlescent materials," *Color Research and Application*, vol. 32, no. 4, pp. 256–266, 2007.

[12] C. S. McCamy, "Observation and measurement of the appearance of metallic materials. Part I. macro appearance," *Color Research and Application*, vol. 21, no. 4, pp. 292–304, 1996.

[13] C. S. McCamy, "Observation and measurement of the appearance of metallic materials. Part II. micro appearance," *Color Research and Application*, vol. 23, no. 6, pp. 362–373, 1998.

[14] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis, *Geometrical Considerations and Nomenclature for Reflectance*, p. 94145, Jones and Bartlett Publishers, Inc., 1977.

[15] R. Cook and K. Torrance, "A reflectance model for computer graphics," *ACM Transactions on Graphics*, vol. 1, no. 1, pp. 7–24, 1982.

[16] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance, "Microfacet models for refraction through rough surfaces," in *Proceedings of the 18th Eurographics Conference on Rendering Techniques*. 2007, EGSR'07, p. 195206, Eurographics Association.

[17] J. Zhu, S. Zhao, Y. Xu, X. Meng, L. Wang, and L-Q. Yan, "Recent advances in glinty appearance rendering," *Computational Visual Media (Beijing)*, vol. 8, no. 4, pp. 535–552, 2022.

[18] T. Zeltner, I. Georgiev, and W. Jakob, "Specular manifold sampling for rendering high-frequency caustics and glints," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 149:1–149:15, 2020.

[19] L-Q. Yan, M. Haan, S. Marschner, and R. Ramamoorthi, "Position-normal distributions for efficient rendering of specular microstructure," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–9, 2016.

[20] L-Q. Yan, M. Haan, B. Walter, S. Marschner, and R. Ramamoorthi, "Rendering specular microgeometry with wave optics," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–10, 2018.

[21] W. Jakob, "Mitsuba renderer," 2010, http://www.mitsuba-renderer.org.

[22] X. Chermain, F. Claux, and S. Mrillou, "Glint rendering based on a multiplescattering patch brdf," *Computer Graphics Forum*, vol. 38, no. 4, pp. 27–37, 2019.

[23] E. Veach and L. J. Guibas, "Optimally combining sampling techniques for monte carlo rendering," in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. 1995, SIGGRAPH '95, p. 419428, Association for Computing Machinery.

[24] B. Wang, M. Haan, N. Holzschuch, and L-Q. Yan, "Example-based microstructure rendering with constant storage," *ACM Transactions on Graphics*, vol. 39, no. 5, pp. 1–12, 2020.

[25] H. Deng, Y. Liu, B. Wang, J. Yang, L. Ma, N. Holzschuch, and L-Q. Yan, "Constant-cost spatio-angular prefiltering of glinty appearance using tensor decomposition," *ACM Transactions on Graphics*, vol. 41, no. 2, pp. 1–17, 2022.

[26] J. Fan, B. Wang, W. Wu, M. Hasan, J. Yang, and L-Q. Yan, "Efficient specular glints rendering with differentiable regularization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 6, pp. 1–1, 2023.
[27] C. Zhang, B. Miller, K. Yan, I. Gkioulekas, and S. Zhao, "Path-space differentiable rendering," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 143:1–143:19, 2020.
[28] T. Zirr and A. Kaplanyan, "Real-time rendering of procedural multiscale materials," in *Proceedings - 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, I3D 2016*. 2016, pp. 139–148, ACM.
[29] B. Wang, H. Deng, and N. Holzschuch, "Real-Time Glints Rendering with Prefiltered Discrete Stochastic Microfacets," *Computer Graphics Forum*, vol. 39, no. 6, pp. 144–154, Sept. 2020.
[30] X. Chermain, B. Sauvage, J.M. Dischler, and C. Dachsbacher, "Procedural physically based brdf for realtime rendering of glints," *Computer Graphics Forum*, vol. 39, no. 7, pp. 243–253, 2020.
[31] A. Butterfield, A. Kerr, and G. E. Ngondi, "Mipmap," 2016.
[32] M. Nguyen, J-B. Thomas, and I. Farup, "Statistical analysis of sparkle in snow images," *Journal of Imaging Science and Technology*, vol. 66, no. 5, pp. 050404–1–050404–1, 2022.
[33] A. Ferrero and S. Bayn, "The measurement of sparkle," *Metrologia*, vol. 52, no. 2, pp. 317–323, 2015.
[34] A. Atanasov and V. Koylazov, "A practical stochastic algorithm for rendering mirror-like flakes supplementary document," in *ACM SIGGRAPH 2016 Talks*. 2016, pp. 1–2, ACM.
[35] B. Wang and H. Bowles, "A robust and flexible real-time sparkle effect," in *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations*. 2016, EGSR '16, p. 4954, Eurographics Association.
[36] J. T. Kajiya, "The rendering equation," in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*. 1986, SIGGRAPH '86, p. 143150, Association for Computing Machinery.
[37] D. Greenberg, K. Torrance, P. Shirley, J. Arvo, E. Lafortune, J. Ferwerda, B. Walter, B. Trumbore, S. Pattanaik, and S.-C. Foo, "A framework for realistic image synthesis," in *International Conference on Computer Graphics and Interactive Techniques: Proceedings of the 24th Annual Conference on Computer graphics and Interactive Techniques*. 1997, pp. 477–494, ACM Press/Addison-Wesley Publishing Co.
[38] D.C. Gross, "Report from the fidelity implementation study group," in *Simulation Interoperability Workshop, Simulation Interoperability and Standards Organization*, Orlando, FL, USA, 1999, Paper 99S-SIW-167.
[39] M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods*, Wiley-VCH, Weinheim, 2nd rev. and enl. ed. edition, 2008.
[40] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, Methuen's Monographs on Applied Probability and Statistics. Methuen, 1965, c1964.
[41] E. Heitz, J. Hanika, E. d'Eon, and C. Dachsbacher, "Multiple-scattering microfacet bsdfs with the smith model," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–14, 2016.
[42] E. Heitz, "Understanding the masking-shadowing function in microfacet-based brdfs," *Journal of Computer Graphics Techniques (JCGT)*, vol. 3, no. 2, pp. 48–107, 2014.

V. Appendix

In this appendix, we provide a brief review of related light transport concepts applied in the rendering and simulation of sparkles. More specifically, we will examine how patch-NDFs are evaluated and provide a concise description of multiple importance sampling and multiple scattering BRDFs.

A. Evaluating $\mathcal{P}$-NDFs

The integral for evaluating a $\mathcal{P}$-NDF can be written as [4]:

$$D_{\mathcal{P}}(\mathbf{z}) = \int_{-\infty}^{\infty} G_p(\mathbf{u}) G_r(n(\mathbf{u}) - \mathbf{z})\, d\mathbf{u}, \tag{2}$$

where $\mathbf{z}$ is a given normal defined by *(s, t)* on the projected hemisphere to be queried, $\mathbf{u}$ is the texture parameter *(u, v)*, $n(\mathbf{u})$ is the normal map function returning a normal in (-1,1) such that the full normal vector is $(n(\mathbf{u}), \sqrt{1 - n(\mathbf{u})^2})$, G_p is a Gaussian reconstruction kernel projected by the patch $\mathcal{P}$, and G_r is the intrinsic roughness Gaussian. Discrete piecewise elements are used to analytically compute the $\mathcal{P}$-NDF [17]. Yan *et al.* [19] precomputed an approximation $\mathcal{N}(\mathbf{u}, \mathbf{z})$ to the factor $G_r(n(\mathbf{u}) - \mathbf{z})$ as a sum of 2D Gaussians $\mathbf{u}$ and $\mathbf{z}$. With scaling coefficients c_i, means $\mathbf{x}_i$, and covariance matrices $\mathbf{\Sigma}_i^{-1}$:

$$\mathcal{N}(\mathbf{u}, \mathbf{z}) = G_r(n(\mathbf{u}) - \mathbf{z}) \approx \sum_{i=1}^{m} G_i(\mathbf{u}, \mathbf{z}), \tag{3}$$

where

$$G_i(\mathbf{u}, \mathbf{z}) = c_i \exp(-\frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^T \mathbf{\Sigma}_i^{-1} (\mathbf{x} - \mathbf{x}_i)) \tag{4}$$

and $\mathbf{x} = (\mathbf{u}, \mathbf{z})^T$.

They then proposed a $\mathcal{P}$-NDF integral for explicit representation of microfacets written as [19]

$$D_{\mathcal{P}}(\mathbf{z}) = \int_{-\infty}^{\infty} G_p(\mathbf{u})\mathcal{N}(\mathbf{u}, \mathbf{z})\, d\mathbf{u} \approx \sum_{i=1}^{m} \int_{-\infty}^{\infty} G_{\mathcal{P}}(\mathbf{u}) G_i(\mathbf{u}, \mathbf{z})\, d\mathbf{u} \tag{5}$$

where *m* is the number of elements. It is worth noting that Eq. 5 is suitable for evaluations of $\mathcal{P}$-NDFs in methods that explicitly represent microstructures.

B. Multiple Importance Sampling

To the best of our knowledge, Wang *et al.*'s work [29] presents the only real-time use of importance sampling [39] in their rendering of sparkles to date. Their precomputations include using BRDF importance sampling to obtain the outgoing directions. Other real-time methods mentioned in this report do not make use of importance sampling. There is no mention of it in the works of Zirr *et al.* [28] and Wang *et al.* [2]. Both Chermain *et al.* [30] and Deliot *et al.* [10] mentioned the absence of importance sampling when sampling the limitations of their respective works. The former stated that it was a potential area for future work and improvements.

Yan *et al.* [19] stated that their method could model materials in a standard BRDF sampling framework with multiple importance sampling (MIS) [40], a technique introduced by Veach and Guibas [23] that combines any number of importance sampling distributions to evaluate the same integral in order to reduce variance. The idea of MIS is to use several different techniques, each with a different probability density function and sampling process, to sample different features of the integrand [23].

With respect to how MIS is used in the rendering of sparkles, the technique proposed by Yan *et al.* [19] involved sampling a Gaussian element proportional to its contribution to the pixel footprint and then picking a normal from that element. Fig. 16 shows the comparison between BRDF and light sampling on an illuminated scratched surface as well as the combined image of both. Wang *et al.* [24] and Deng *et al.* [25] both use a similar technique in their methods as well. Jakob *et al.* [8], however, defined a smooth density function to describe the distribution of flakes and importance sampled that instead of the $\mathcal{P}$-NDF. While this reduced computational costs, it was not capable of sampling "wide and heavy-tailed microfacet distributions" according to Atanasov and Koylazov [9].

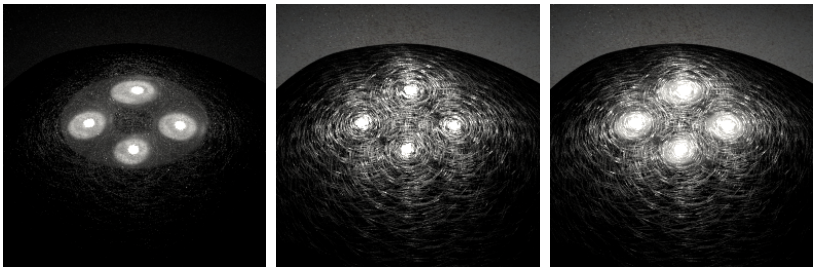

Fig. 16: Comparison of images of a scratched surface rendered by different methods (redrawn from Yan *et al.* [19]). **Left:** Image rendered with BRDF sampling. **Center:** Image rendered with light sampling. **Right:** Image rendered combining both techniques. The BRDF sampling was able to capture reflections from the light in flat areas, but could not depict the scratches that light sampling was able to render. The combined image provided benefits of both methods.

C. Multiple Scattering BRDF

Heitz *et al.* [41] noticed that many works in this area only consider single scattering on the microsurface and ignored any light bounces that would scatter multiple times between microfacets. According to

Heitz *et al.* [41] this omission affects energy conservation calculations and colour saturation in certain areas. Hence, they introduced a microfacet bidirectional scattering distribution function (BSDF) that modeled multiple scattering. Chermain *et al.* [22] noted that Heitz *et al.*'s method [41] involved high computational costs. They also claimed to introduce the first multiple scattering sparkle (glint) model based on using normal maps. They evaluated their results through comparisons with synthetic images and white furnace tests [42] whose results were composed to those obtained considering classic normal mapped surfaces. These tests are employed to verify that a BRDF is energy preserving by involving a 100% reflective object that becomes indistinguishable from the environment when uniformly lit.