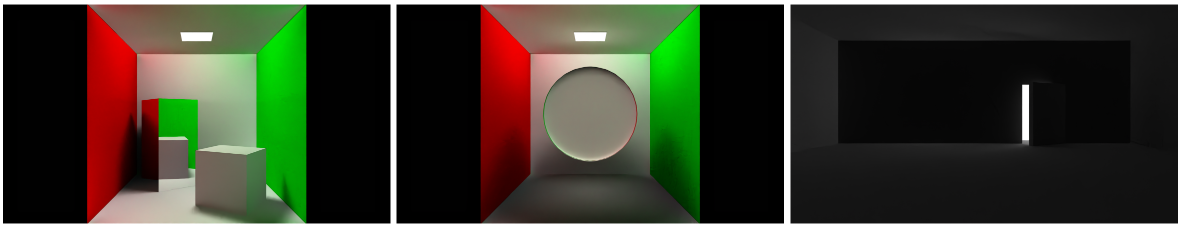
ISSN 2464-4617 (print)
ISSN 2464-4625 (online)
Computer Science Research Notes - CSRN 3401
http://www.wscg.eu
WSCG 2024 Proceedings

Combining bidirectional path tracing, DDGI, and ReSTIR to improve real-time rendering quality

Viktor Kretzschmar
Hochschule Darmstadt
h_da
Schöfferstraße 3
64295, Darmstadt,
Germany
viktor.kretzschmar
@h-da.de

Prof. Dr. Benjamin Meyer
Hochschule Darmstadt
h_da
Schöfferstraße 3
64295, Darmstadt,
Germany
benjamin.meyer
@h-da.de

Prof. Dr. Elke Hergenröther
Hochschule Darmstadt
h_da
Schöfferstraße 3
64295, Darmstadt,
Germany
elke.hergenroether
@h-da.de



Three example images generated by the algorithm described in this paper. Each image shows soft shadows, indirect illumnination and refraction.

Abstract

This paper introduces a new algorithm for real-time rendering that builds upon bidirectional path tracing, reservoir-based spatio-temporal importance resampling, and dynamic diffuse global illumination. The combination of these algorithms produces an image with reduced noise compared to real-time run algorithms like path tracing, while retaining details like caustics. The resulting darkening, which is discussed in greater detail in this paper, is also reduced due to the usage of importance resampling of points on light-emitting surfaces. While the standard algorithms, such as bidirectional path tracing, cannot be run in real-time with satisfactory quality, a set of novel approaches have emerged to fill this gap. These algorithms are capable of running in real-time, although they do suffer from certain limitations. This paper describes the combination of bidirectional path tracing, DDGI and ReSTIR. This rectifies the drawbacks of missing indirect reflection, darkening and missing caustics of these algorithms. Ultimately, all the results of these algorithms are compared by verifying real-time rendering time and comparing quality to reference images. The quality is evaluated by using comparions for darkening and the similarity of the real-time rendered result to an offline path-traced result. The results of this paper demonstrate that the algorithm presented improves upon previous algorithms in terms of quality, while still maintaining real-time rendering constraints.

Keywords

real-time rendering, global illumination, ray tracing

1 INTRODUCTION

Over the past decade, ray tracing has moved from pre-rendered media to real-time applications. While hardware advances have made real-time ray tracing possible on consumer-grade hardware, real-time path tracing remains out of reach. However, path tracing offers several significant advantages over traditional ray tracing.

Path tracing produces more realistic images by incorporating indirect lighting and refraction without sampling the light sources themselves. As scenes get larger, the probability of selecting a visible light decreases, leading to darkness without any adjustment to brightness. Path tracing, on the other hand, is unaffected by darkening in larger scenes and maintains consistent illumination.

Ideally, there would be an algorithm that supports direct illumination, indirect illumination and refraction, while still being able to run in real-time (i.e. $1/30$ seconds of render time per frame). This paper compares several approaches and presents a combined algorithm that supports both indirect illumination and refraction, providing a solution for darkening larger scenes while improving realism.

This paper uses terms such as ray tracing, path tracing and bidirectional path tracing. Both path tracing and bidirectional path tracing are implemented as described in [PJH16] using uniform direction sampling. Ray tracing is used as a short term for the algorithm presented in [Whi80].

2 RELATED WORK

This paper discusses three main techniques: Firstly, the bidirectional path tracer. Second, the improvements to ray tracing by Talbot *et al.* [Tal05] with reservoir-based spatio-temporal importance resampling (ReSTIR) by Bitterli *et al.* [BWP+20]. And thirdly, improvements in global illumination by McGuire *et al.* [MMNL17] improved by Majercik *et al.* [MGNM19, MMK+21, MMK+22] with Dynamic Diffuse Global Illumination (DDGI). All these algorithms are the basis of the algorithm discussed in this paper and are compared with each other and the result of this paper.

ReSTIR is a relatively new algorithm developed in 2020. It is a method based on importance sampling that uses reservoirs to speed up the sampling process. It is based on the idea that importance sampling becomes more accurate the closer the light distribution is to the distribution from which samples are taken [Vea98]. The accuracy of the light distribution match can be improved by reusing information across space and time, also known as spatio-temporal reuse. This has been applied to bidirectional path tracing [GKDS12, Kel97, LW98, VG95, VG97], to path guidance [DK18, Jen95, LW95, MGN17, VKŠ+14], and to resampling strategies [HKD14, KSKAC02, Tal05, VG97, KMA+15]. While resampling is a good strategy, it causes significant overhead when implemented without an optimisation approach. For this reason, reservoir-based resampling has been used, [CHA82]. Based on this, ReSTIR was formulated by Bitterli *et al.* [BWP+20].

DDGI is an algorithm that improves global illumination by adapting ray tracing with irradiance caching. Irradiance caching is a method of improving illumination by caching illumination data. It was introduced by Ward *et al.* [WRC88], and has been further developed by [A+86, Hec90]. These implementations are statically rendered, i.e. they do not react to changing or moving geometry in the scene. In contrast to static lighting is dynamic lighting, which incorporates changes in the scene. Partially dynamic approaches have been implemented [GS12, RZD14, SL17, SSS+20, SNRS12, SJJ12, VPG14]. DDGI by Majercik *et al.* [MGNM19, MMK+21, MMK+22] further develops these techniques based on McGuire *et al.* [MMNL17] to introduce fully dynamic rendering.

This paper builds on the Vulkan API to access the render time acceleration of graphics cards. The render-time acceleration is achieved through acceleration structures built by the graphics drivers themselves, which limits the use of improved techniques such as Multi-Level Hashed Grids [CPJ10], which can only be easily implemented by the vendor. Also, this work does not use any denoising or machine learning in the actual rendering process, such as CNN-enhanced Poisson Interpolation [ELM23], which could be used to further improve the resulting images.

3 DARKENING PROBLEM

Contrary to expectation, some rendering algorithms produce darker scenes when more light sources are added. This darkening effect particularly occurs in larger scenes where light sources are scattered amongst obstructive geometry. Algorithms that select a light source are prone to darkening, as they often encounter interrupted light paths in bigger scenes, casting shadows on objects. Larger scenes with more light sources increase the likelihood of sampling a light that doesn't contribute to the scene's brightness. This darkening effect is addressed by ReSTIR [BWP+20], though not explicitly defined as such in their research. Notably, algorithms like path tracing are unaffected by darkening since they do not directly sample light.

Basic ray tracing is impacted by darkening. It traces rays from the camera, hitting geometry and connecting to light-emitting surfaces to determine brightness. While efficient for direct lighting, sampling light-emitting surfaces becomes problematic in larger scenes, as the probability of connecting to the relevant light sources decreases (see Figure 1).

Reservoir-based spatio-temporal importance resampling (ReSTIR) [BWP+20] addresses the darkening issue in real-time rendering time. By prioritizing points on light-emitting surfaces based on hit probability, it ensures that non-contributing points are disregarded while important ones are more likely to be selected (see Figure 2). It also uses a step in which it disregards non connecting paths. This results in a uniformly lit scene.

To evaluate the darkening produced by each algorithm, two scenes are compared: one with a single Cornell box and another with five non-intersecting Cornell boxes. By positioning the camera to capture the same angle in both scenes, a pixel-by-pixel brightness comparison yields a percentage value indicating the degree of darkening. Values range from 20% (no darkening countermeasures) to 100% (completely darkening-free). Re-

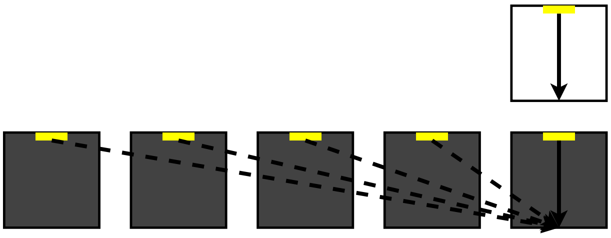


Figure 1: This illustration visualizes the darkening effect. In the first row, a single box with a light source is lit normally since all light points are unobstructed. In the second row, with five boxes, selecting a light from a different box in four out of five cases results in an 80% darker scene. Algorithms like ray tracing and bidirectional path tracing are impacted since they require selecting a scene light source.

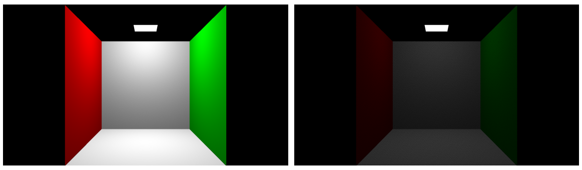


Figure 2: Comparison between two seemingly identical Cornell boxes. Both are rendered using ray tracing, but the right one is in a scene with five other Cornell boxes, resulting in a darker image.

sults and their implications are thoroughly analyzed in the Results section.

4 REFRACTION PROBLEM

Refraction is the bending of light when transported throu transparent materials. Depending on the refractive index of the material, the direction of the light ray is changed. For this to work, the entry and exit of a light ray must be processed. Just trying to connect two points in space will not work, because the object with the transparent material will be interpreted as an obstruction. Simple ray tracing only traces a ray from the camera to a point, and from that point to a point on a light-emitting surface. A transparent object will therefore cast a shadow. This is visualised, along with an example of its absence, in figure 3. This requires the light rays to be focused on a specific point. Rendering algorithms such as path tracing have a highter amount of paths traced from one pixel to a light source because of refraction, resulting in caustics. Both methods are computationally expensive, leading to a variety of algorithms that trade caustics for performance.

In ray tracing, transparent materials and caustics cannot be simulated by either ReSTIR, DDGI or a combination of the two. ReSTIR only improves the sampling of points on light emitting surfaces; it does not change how rays are actually traced. As a result, hitting transparent geometry will still cast a shadow. Similarly, DDGI only adds an estimate for global illumina-

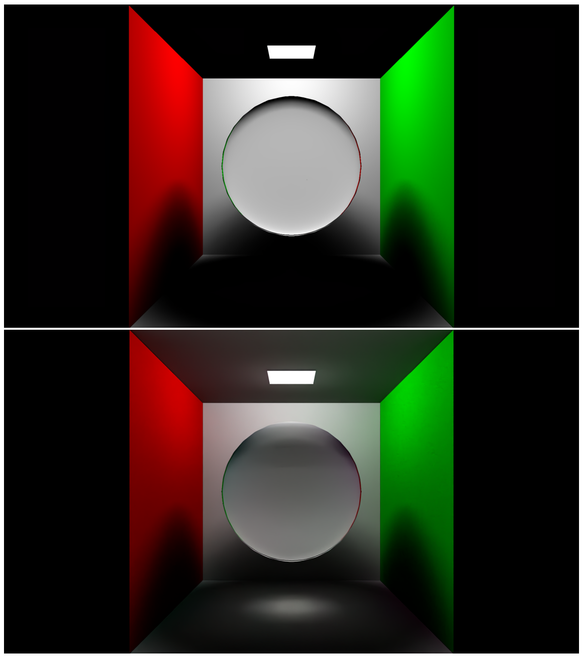


Figure 3: Both images contain a glass sphere that should refract the light coming from the lamp above. However, only the lower image has a visible focal point of light, while the upper image is completely dark below the sphere. The glass sphere ideally bends the light coming from the lamp and focuses it on the ground, creating caustics. The upper image is rendered using ray tracing, which does not simulate refraction, resulting in the dark shadow below the sphere. The lower image is rendered using bidirectional path tracing (without the constraints of real-time) and shows the light being focused by simulated refraction, creating a bright spot below the sphere.

tion. It also does not change the way rays are traced, so transparent materials will still cast shadows. But these shadows are less dark, thanks to the global illumination estimate. There are still no caustics at the focal points.

Both path tracing and bidirectional path tracing support transparency and refraction, and thus simulate caustics. Their main problem is that they do not run in real-time for good quality, even on dedicated hardware. The ideal would be to combine the speed of ReSTIR and DDGI with the quality of (bidirectional) path tracing. This would ideally preserve their speed while providing results such as caustics.

5 COMBINING DIFFRENT ALGORITHMS

As discussed earlier, ReSTIR, DDGI and bidirectional path tracing all have specific advantages. ReSTIR improves the sampling of points on light emitting surfaces, resulting in improved light levels in larger scenes. DDGI provides a way to simulate global illumination

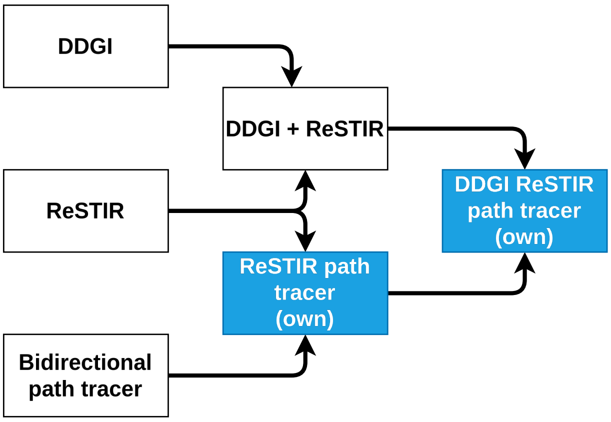


Figure 4: These are all the algorithms discussed in this paper and how they relate to each other. Two algorithms that point to a third show that they are combined into that algorithm. DDGI is from [MGNM19], ReSTIR is from [BWP^{+}20] and the combination of the two is from [MMK^{+}22]. Parts marked "own" with a blue background are new ideas from this paper.

without the performance penalty of path tracing. Bidirectional path tracing itself is a high quality, reliable algorithm that balances accuracy with speed. There are several ways to combine these algorithms. This paper is mainly concerned with the combination of all three algorithms, but also discusses their predictors. The relationship between the different algorithms can be seen in the figure 4.

ReSTIR improves ray tracing by preventing darkening (see section 3). This is done by introducing a sampling bias into the sampling of points on light emitting surfaces, giving each possible starting point for a ray a weight. DDGI adds the ability to estimate indirect lighting by building on top of ray tracing. It collects different light information by simply tracing rays, as is done when generating the actual image. Comparing the two, it can be seen that ReSTIR changes the start of the ray tracing operation, while DDGI changes the end. This allows them to be combined, which has already been done by Majercik et al. [MMK^{+}22] As a result, DDGI, which previously suffered from darkening, now has a way to mitigate this.

Bidirectional path tracing is very similar to ray tracing. Both sample a point on a light-emitting surface, and both trace initial rays from the camera. As a result, both suffer from similar problems. Bidirectional path tracing, like ray tracing, has a problem with darkening. However, this is the result of using the same sampling algorithm for both algorithms. This makes it possible to use ReSTIR sampling in bidirectional path tracing, which reduces darkening. However, this adds noise to the image due to the still high processing cost, but inherits the ability to simulate transparency.

Finally, DDGI's Global Illumination Estimation can be added. DDGI's Global Illumination Estimation calculates an estimate based on ray tracing information previously collected by probes. These probes can then be used to calculate a specific background light level at any point in space. This is used in DDGI at the point where the traced ray hits the geometry, and the estimate is collected at the hit point. The estimate and the result of the ray tracing operation are added together to produce the final result. Because it is based on ray tracing, it inherits its limitations when it comes to transparency. Interestingly, ray tracing in DDGI can be replaced by bidirectional path tracing because it does not modify the ray itself. The only exception is the addition of the estimate.

Bidirectional path tracing starts from both a point on a light-emitting surface and a vision ray. Because the ray of sight is always the same for the same pixel, it can be reused, reducing the amount of computation required. The results of this ray are averaged. Coincidentally, the first hit of the vision ray is also the point at which DDGI injects its estimate. As a result, the estimate can simply be added to the result of the bidirectional path tracer at the first hit point of the vision ray. Adding the estimation adds more colour data to the overall image, smoothing out noise while preserving transparency and refraction. Adding ReSTIR also gives it the ability to mitigate darkening through improved sampling of points on light-emitting surfaces.

This combination of all three algorithms - bidirectional path tracing as the base, ReSTIR as an improved sampling of points on light emitting surfaces, and DDGI as an additional estimation of global illumination - is the central algorithm of this paper. The combination can be expressed as $L = L_{DDGI} + L_{BPT}(P_{ReSTIR})$. L describes the illumination, where L_{DDGI} describes the illumination given by the DDGI estimates. P_{ReSTIR} are the points on light emitting surfaces selected by ReSTIR and used by L_{BPT} as points from which to trace light rays. L_{BPT} stands for Bidirectional path tracing and results in a light value. Both light values are then added together for the final result, the same way DDGI handles the combination of direct illumination by ReSTIR and its own illumination estimation. This can lead to certain radiance contributens being accounted for multiple times, reasulting in a brighter image. The combination of all three algorithms still is an improvement in real-time rendering, as it is comparable to regular ray tracing in terms of performance, while providing caustics with reduced darkening and noise compared to pure bidirectional path tracing. A better breakdown of the results can be found in the next section.

6 RESULTS

The results are compared in four categories. Firstly, rendering times are discussed. These are the baseline to put all algorithms on an equal footing in the comparison by configuring them to be at least 30 FPS fast. Next, we discuss the darkening of the different algorithms. It shows which algorithms suffer more from darkening than others. The actual quality is then compared by comparing each image with a reference image from the same camera position and angle. The calculation compares each pixel on how different it is from the offline rendered reference. This results in values that can be compared across algorithms. Finally, some of the missing details are shown by providing images of certain details of the scene.

Rendering time is a critical test case for real-time rendering. Speeds of 30 or even 60 FPS are desirable. Therefore there is only a time window of 33.3 to 16.6 milliseconds for each frame. Figure 5 shows the rendering times. It shows lines for each algorithm and its rendering time per scene. Marker lines are also inserted to show the 30 and 60 FPS markers.

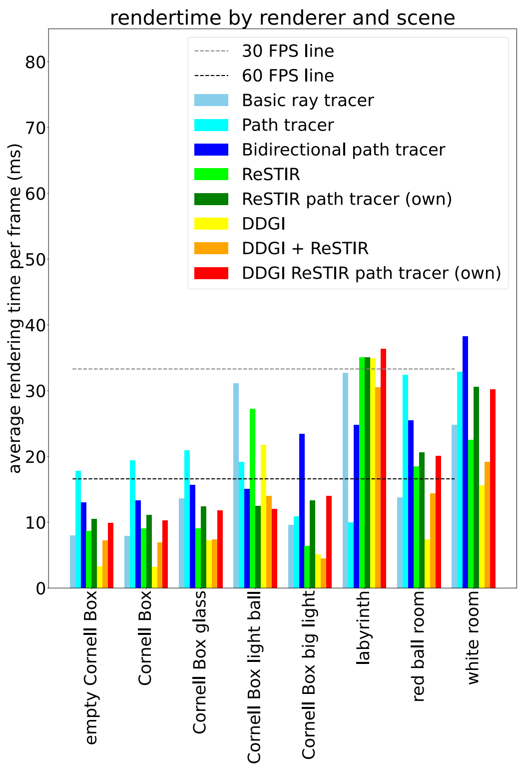


Figure 5: All algorithms and scenes and their respective render times. The two dotted lines also show the render time limits for 30 and 60 FPS.

Rendering algorithms have a number of parameters that represent a trade-off between performance and quality. Real-time rendering is constrained in terms of performance by the 30 FPS minimum. Consequently, the parameters of the algorithms have been chosen so that they do not fall below this minimum. This makes the algorithms comparable, as no algorithm can take advantage of excessive rendering times. Each algorithm has to meet the requirement to run in real-time. It is interesting to note that the most demanding scenes are the Cornell box with the light sphere, the maze and the white room. However, it is worth noting that some algorithms are better suited to certain scenes than others.

Darkening is the next problem to be discussed. As noted in section 3, scenes with more obstructed light-emitting surfaces are darker for some algorithms. The one and five Cornell box test is used to visualise this. This test is an original work of this paper. It tests darkening by comparing an image rendered in a scene with a single Cornell box with an image generated in a scene with five Cornell boxes. A correct result is an identical image or images with the same brightness when rendered with the same renderer. Both images are compared based on the average brightness preserved from the single to the five Cornell boxes as a relative value, as shown in figure 2. The brightness is calculated pixel by pixel by dividing the pixel brightness of the five Cornell box scene image by the pixel brightness of the single Cornell box scene image. All resulting values are averaged to give a value between $[0,1]$. Pixels where the brightness of the single Cornell box scene image is zero are ignored due to division by zero. A value of 100% means identical brightness, while a value of 20% would mean a reduction in brightness of 80%. 100% is ideal, i.e. no darkening. The calculated results are shown in the figure 6.

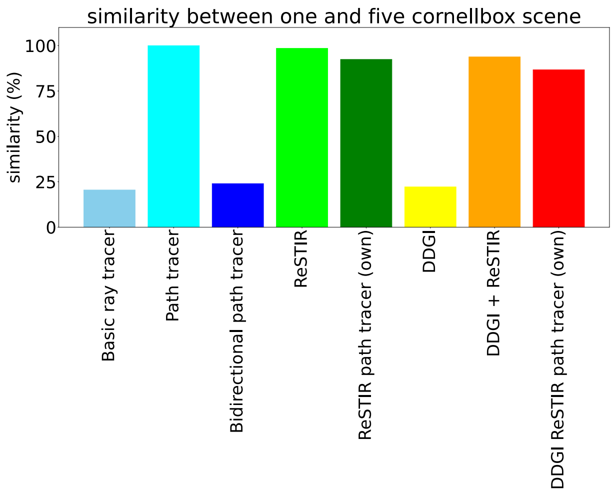


Figure 6: Brightness is maintained when comparing the rendering result between a single and five Cornell boxes. 100% is optimal, while in this case 20% is the worst case.

Both ray tracing and bidirectional path tracing have a darkening problem, whereas path tracing does not. This is because both ray tracing and bidirectional path tracing use a step where a light is sampled, whereas path tracing does not. If the scene contains multiple light-emitting surfaces, it becomes more likely that a sampled light will be obstructed, affecting the hit point of the vision ray, a problem that cannot be solved by increasing the number of samples taken because the probability, and hence the distribution of the resulting hits, does not change. path tracing, on the other hand, works perfectly because it does not have to worry about sampling points on light-emitting surfaces. It follows the ray until it hits a light-emitting surface or a specified maximum depth at which the ray stops. As a result, it is not influenced by other light-emitting surfaces in the scene. The results are identical because the rays hit the same geometry.

Of the other algorithms, only pure DDGI has a darkening problem. ReSTIR is a ray tracing algorithm with resolved darkening, so it is expected to have almost no darkening. As a result, all algorithms that use ReSTIR also have very low levels of darkening. It is important to note that as more algorithms are combined, the darkening increases slightly. The DDGI ReSTIR path tracer does not have as extreme a darkening as pure ray tracing, but it is still noticeable at around 10% in reduced image brightness. This concludes the darkening comparisons.

The quality is compared with reference images. A bidirectional path tracer without real-time rendering constraints is used to generate the reference images. The comparison between the result and the reference is made using the mean squre error method. The higher the value, the worse the resulting overall difference.

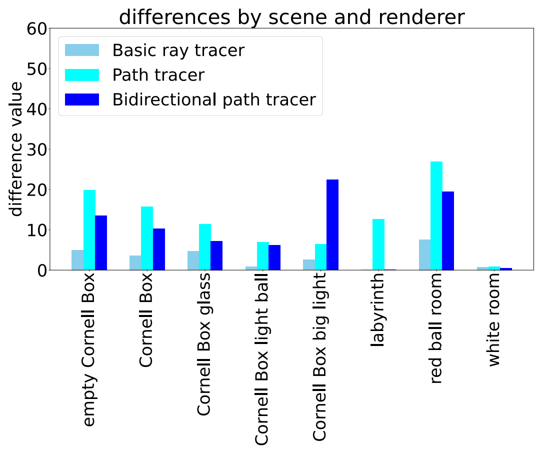


Figure 7: This figure compares the simple ray tracer, the regular path tracer and the bidirectional path tracer with mean square error. It can be seen that the ray tracer is the best of the three renderers in all cases. And the path tracer is worse than the bidirectional path tracer in almost all cases, which is to be expected.

The first comparison is made between the ray tracer, simple and bidirectional path tracers in the figure 7, which is done to show an example where the bidirectional path tracer has a higher quality than the simple path tracer in most cases. The improvement of the bidirectional path tracer is expected because the algorithm is faster than the regular path tracer, which means it can give a better result in real-time, which is the case in every scene except the Cornell Box big light. The immense size of the light-emitting surface favours simple path tracing. On the other hand, while the improvement in all the other scenes is quite significant, the difference in the Labyrinth scene is as good as the raytracer's results. The ray tracer, which is faster than both path tracers, can achieve better results than both because of the speed improvement. The simplification of the traced path results in a speed improvement because the path contains only one bounce. These three compared algorithms are just to show that the comparison method works. It shows the expected results by showing that the bidirectional path tracer has a higher quality than the regular path tracer.

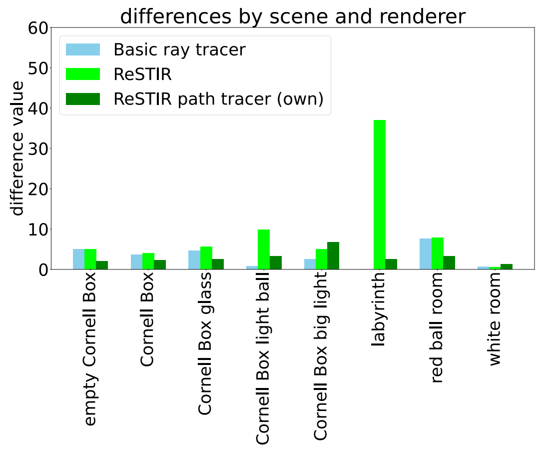


Figure 8: Comparison of the basic raytracer with ReSTIR and the bidirectional path tracer combined with ReSTIR, called the ReSTIR path tracer. It can be seen that the raytracer is better than the other two algorithms, while the other two are better or worse than each other depending on the scene.

The ray tracer, ReSTIR and the ReSTIR path tracer are compared with the chart in figure 8. The simple ray tracer beats the other algorithms in terms of quality, sometimes by a small margin, as in the case of the empty Cornell Box scene, or by a large margin, as in the case of the Cornell Box scene with large lights. The other two algorithms alternate in quality. As always, the Maze scene is the standout, with ReSTIR looking the worst by a wide margin. To sum up, the basic raytracer has the highest quality of the three algorithms when only looking at this statistic. If you take into account the darkening seen in the previous section, you can make an argument for the ReSTIR raytracer. It has almost no darkening, like standard ReSTIR, but has better quality in half the cases, but never really worse compared to ReSTIR. The basic raytracer has a darkening problem, which makes it generally worse. But more algorithms can be compared.

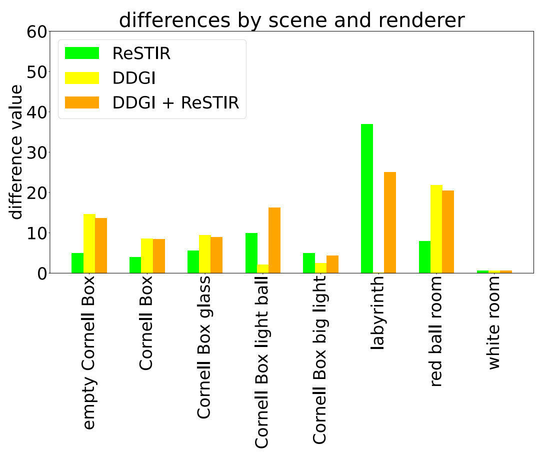


Figure 9: The comparative results for the ReSTIR and DDGI renderers and their combination. It is interesting to note that some of these algorithms alternate in quality depending on the scene. The quality changes with the scenes, but the overall quality is similar.

Next is a comparison between ReSTIR, DDGI and ReSTIR with DDGI, as shown in the figure 9. It depends very much on the scene to say which algorithm is the best and which is the worst. For example, the ReSTIR algorithm is the best for the empty Cornell box scene, but the worst for the maze scene. DDGI is the worst for the empty Cornell Box scene, but the best for the maze, by an impressively wide margin. You would expect ReSTIR and DDGI together to be in the middle, but in half of the scenes it is the worst algorithm. Sometimes this is a small margin, as in the Cornell Box scene, but other times it is a significant margin, as in the Cornell Box big light. The result is a heterogeneous mix in which no algorithm is truly the best.

In the final comparison, DDGI, DDGI with ReSTIR and the bidirectional path tracer with DDGI and ReSTIR are discussed. The results are shown in Figure 10. DDGI and DDGI with ReSTIR are either very similar, with DDGI being slightly worse in some cases, or DDGI is massively better. While this is interesting, the critical part is the DDGI ReSTIR path tracer, which is the best algorithm in half of the scenes, and the worst by a small margin in two scenes.

Figure 11 is included to show the full comparison between all algorithms. It shows all the algorithms side by side, with the final algorithm of this paper, the DDGI ReSTIR raytracer, being one of the better algorithms, being the best in two cases and otherwise having a low error value. The ray tracer and DDGI are both better in some cases, but have a darkening problem which makes them generally worse. In conclusion, the results show

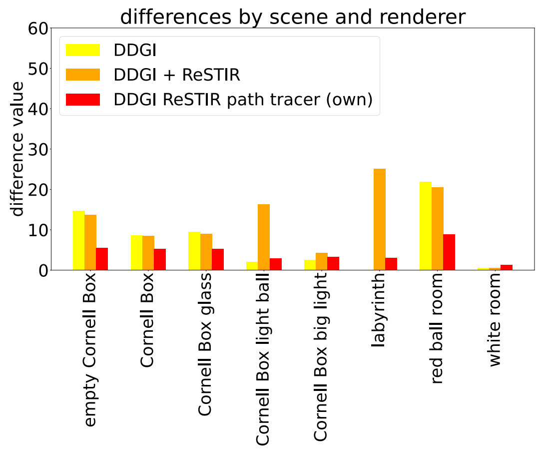


Figure 10: All three different DDGI-based algorithms were compared. The DDGI ReSTIR path tracer is an original algorithm that combines DDGI, ReSTIR and bidirectional path tracing. It is one of the best algorithms in this comparison for most scenes.

that the combination of the bidirectional raytracer with DDGI and ReSTIR gives the best overall quality.

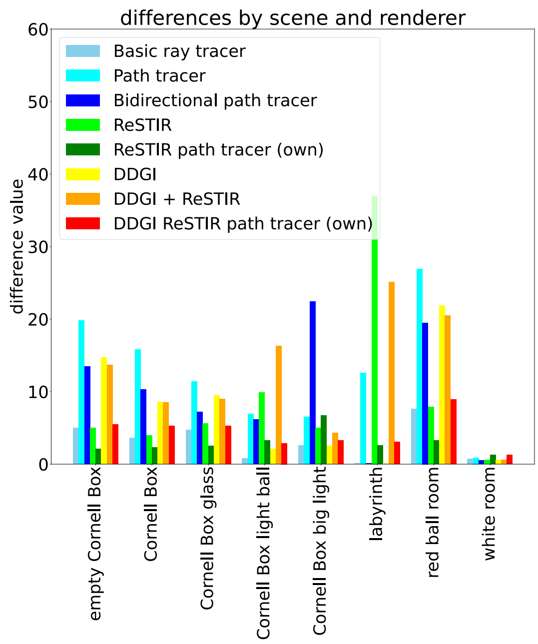


Figure 11: All quality comparison results are shown in the figure.

Some details are not quantified in this paper, but should be shown for completeness, mainly concerning caustics, for which the scene Cornell Box glass is used. The light is concentrated at the bottom, resulting in a bright spot. A visual comparison between different algorithms can be seen in figure 12. The images were denoised with the Intel ODIN denoiser, as were the other images in this paper. The visual comparison focuses on the Cornell Box Glass scene and shows both the indirect illumination and the caustics for comparison.

The reference image is bright but casts dark shadows. Especially the ceiling and under the glass sphere are very dark. There are also caustics on the floor, resulting in a bright spot. These caustics are not visible in the next two renderers, the raytracer and the DDGI with ReSTIR. Both render dark shadows. But because of its focus on global illumination, DDGI with ReSTIR has a non-black ceiling. It is slightly brighter than the reference and has similar color values. Finally, the final result of this paper, the DDGI ReSTIR path tracer, has caustics. It has a slight but noticeable bright spot on the floor. The ceiling is also illuminated, but brighter than in the other images. As a result of the visual inspection, the solution of this paper still needs improvement, but it produces good results. It shows potential in the area of caustics and global illumination, but is noisy in its raw form. This noise is present on both the ceiling and the floor.

7 CONCLUSION

In conclusion, this paper presents a novel algorithm usable in real-time rendering that simulates both indirect illumination and refraction. This algorithm is a combination of three algorithms: Bidirectional path tracing, DDGI and ReSTIR. All of these algorithms have their advantages and disadvantages, and thus their combination rectifies these to some extent. They have been compared in both quality and speed to see their advantages and disadvantages. But this paper also showed that the new algorithm of this paper is the best overall algorithm when compared in quality and details like caustics.

While the results of this new algorithm improve on its predecessors, it also inherits their limitations. The use of bidirectional path tracing in real-time rendering has always resulted in noisy results. This algorithm is no exception, even if the noise is reduced by the other algorithms. While ReSTIR solves the darkening problem, there is still some darkening when combined with other algorithms, not much, but still measurable. DDGI is an approximation algorithm, and as such does not give correct results, but better results than the other algorithms discussed in this paper, which also affects the final algorithm.

These algorithms mostly rely on bidirectional path tracing to do the heavy lifting of the algorithm. As a result, the rendering result is noisy. While DDGI compensates for this to some extent, highlights such as caustics are particularly noisy, leaving room for optimization.

Neural denoising and neural sampling Neural denoising and neural sampling were developed after the advent of deep learning. Both have been applied to real-time rendering applications [HMS+20]. These denoising and sampling techniques could improve the results by removing noise or improving ReSTIR based sampling.

Caching of paths The algorithm described in this paper is based on bidirectional path tracing. As such, multiple paths are constructed through the scene. These paths could be reused in later frames to improve rendering quality, which has been done based on ReSTIR with regular path tracing [OLK+21]. The application of ReSTIR-based reuse in world space [Boi21] could also be applied to this algorithm for further improvement.

8 REFERENCES

[A+86] James Arvo et al. Backward ray tracing. *Developments in ray tracing*, 12:259–263, 1986.

[Boi21] Guillaume Boissé. World-space spatiotemporal reservoir reuse for ray-traced global illumination. In *SIGGRAPH Asia 2021 Technical Communications*, SA '21 Technical Communications, New York, NY, USA, 2021. Association for Computing Machinery.

[BWP+20] Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph.*, 39(4), July 2020.

[CHA82] M. T. CHAO. A general purpose unequal probability sampling plan. *Biometrika*, 69(3):653–656, 12 1982.

[CPJ10] Vasco Costa, Joao Pereira, and Joaquim Jorge. Multi-level hashed grids for ray tracing. *WSCG*, 2010.

[DK18] Ken Dahm and Alexander Keller. Learning light transport the reinforced way. In Art B. Owen and Peter W. Glynn, editors, *Monte Carlo and Quasi-Monte Carlo Methods*, pages 181–195, Cham, 2018. Springer International Publishing.

[ELM23] Blaž Erzar, Žiga Lesar, and Matija Marolt. Fast incremental image reconstruction with cnn-enhanced poisson interpolation. *Václav Skala-UNION Agency*, 2023.

[GKDS12] Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. Light transport simulation with vertex connection and merging. *ACM Trans. Graph.*, 31(6), nov 2012.

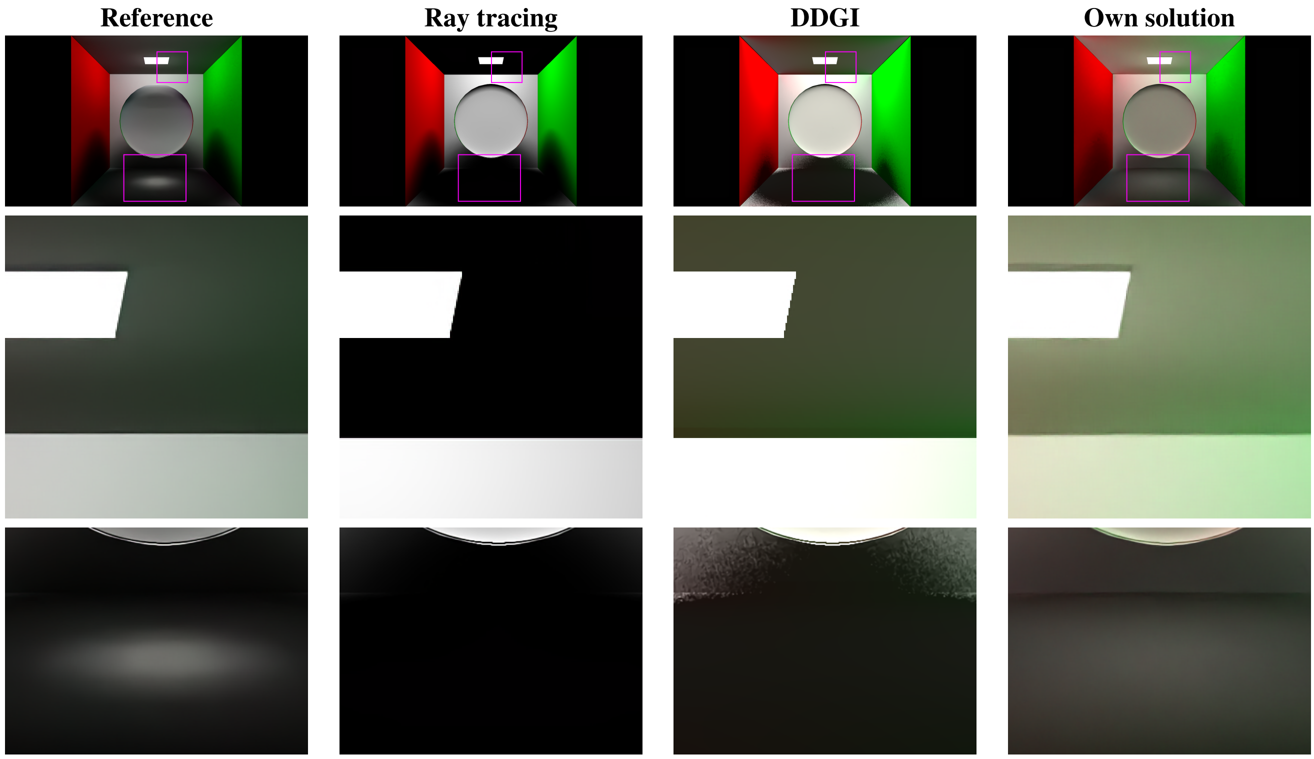


Figure 12: The results of rendering the Cornell Box Glass scene in different renderers for visual comparison. From left to right: Bidirectional path tracing offline reference, the raytracer, DDGI with ReSTIR, and the final result. The magenta boxes are enlarged in the bottom two images below the top images. These images show (the lack of) indirect illumination in the middle image and caustics in the bottom image.

[GS12] NS Mickael Gilabert and Nikolay Stefanov. Deferred radiance transfer volumes, global illumination in far cry 3. In *Game Developers Conference*, volume 2, 2012.

[Hec90] Paul S Heckbert. Adaptive radiosity textures for bidirectional ray tracing. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 145–154, 1990.

[HKD14] Toshiya Hachisuka, Anton S. Kaplanyan, and Carsten Dachsbacher. Multiplexed metropolis light transport. *ACM Trans. Graph.*, 33(4), jul 2014.

[HMS+20] J. Hasselgren, J. Munkberg, M. Salvi, A. Patney, and A. Lefohn. Neural temporal adaptive sampling and denoising. *Computer Graphics Forum*, 39(2):147–155, 2020.

[Jen95] Henrik Wann Jensen. Importance driven path tracing using the photon map. In Patrick M. Hanrahan and Werner Purgathofer, editors, *Rendering Techniques '95*, pages 326–335, Vienna, 1995. Springer Vienna.

[Kel97] A Keller. Instant radiosity, siggraph 97 proceedings, 1997.

[KMA+15] Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. Gradient-domain path tracing. *ACM Transactions on Graphics (TOG)*, 34(4):1–13, 2015.

[KSKAC02] Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. A simple and robust mutation strategy for the metropolis light transport algorithm. In *Computer Graphics Forum*, volume 21, pages 531–540. Wiley Online Library, 2002.

[LW95] Eric P Lafortune and Yves D Willems. A 5d tree to reduce the variance of monte carlo ray tracing. In *Rendering Techniques' 95: Proceedings of the Eurographics Workshop in Dublin, Ireland, June 12–14, 1995 6*, pages 11–20. Springer, 1995.

[LW98] Eric Lafortune and Yves Willems. Bidirectional path tracing. *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics'*, 93, 01 1998.

[MGN17] Thomas Müller, Markus Gross, and Jan Novák. Practical path guiding for efficient light-transport simulation. In *Computer Graphics Forum*, volume 36, pages 91–100. Wiley Online Library, 2017.

[MGNM19] Zander Majercik, Jean-Philippe Guertin, Derek Nowrouzezahrai, and Morgan McGuire. Dynamic diffuse global illumination with ray-traced irradiance fields. *Journal of Computer Graphics Techniques (JCGT)*, 8(2):1–30, June 2019.

[MMK+21] Zander Majercik, Thomas Müller, Alex Keller, Derek Nowrouzezahrai, and Morgan McGuire. Dynamic diffuse global illumination resampling. *Computer Graphics Forum*, page 13, Dec 2021.

[MMK+22] Zander Majercik, Thomas Müller, Alexander Keller, Derek Nowrouzezahrai, and Morgan McGuire. Dynamic diffuse global illumination resampling. *Computer Graphics Forum*, 41(1):158–171, 2022.

[MMNL17] Morgan McGuire, Michael Mara, Derek Nowrouzezahrai, and David Luebke. Real-time global illumination using precomputed light field probes. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, page 11, February 2017. I3D 2017.

[OLK+21] Y. Ouyang, S. Liu, M. Kettunen, M. Pharr, and J. Pantaleoni. Restir gi: Path resampling for real-time path tracing. *Computer Graphics Forum*, 40(8):17–29, 2021.

[PJH16] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation 3rd Edition*. Elsevier, November 2016.

[RZD14] Hauke Rehfeld, Tobias Zirr, and Carsten Dachsbacher. Clustered pre-convolved radiance caching. In *EGPGV EuroVis*, pages 25–32, 2014.

[SJJ12] Jorge Schwarzhaupt, Henrik Wann Jensen, and Wojciech Jarosz. Practical hessian-based error control for irradiance caching. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.

[SL17] Ari Silvennoinen and Jaakko Lehtinen. Real-time global illumination by precomputed local reconstruction from sparse radiance probes. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017.

[SNRS12] Daniel Scherzer, Chuong H Nguyen, Tobias Ritschel, and Hans-Peter Seidel. Pre-convolved radiance caching. In *Computer Graphics Forum*, volume 31, pages 1391–1397. Wiley Online Library, 2012.

[SSS+20] Dario Seyb, Peter-Pike Sloan, Ari Silvennoinen, Michał Iwanicki, and Wojciech Jarosz. The design and evolution of the uberbake light baking system. *ACM Transactions on Graphics (TOG)*, 39(4):150–1, 2020.

[Tal05] Justin F Talbot. *Importance resampling for global illumination*. Brigham Young University, 2005.

[Vea98] Eric Veach. *Robust Monte Carlo methods for light transport simulation*. Stanford University, 1998.

[VG95] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*, pages 145–167. Springer, 1995.

[VG97] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 65–76, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[VKŠ+14] Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics (TOG)*, 33(4):1–11, 2014.

[VPG14] K Vardis, G Papaioannou, and A Gkaravelis. Real-time radiance caching using chrominance compression. *Journal of Computer Graphics Techniques Vol*, 3(4), 2014.

[Whi80] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, jun 1980.

[WRC88] Gregory J Ward, Francis M Rubinstein, and Robert D Clear. A ray tracing solution for diffuse interreflection. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 85–92, 1988.