

光線追蹤的相干性聚集：硬體光線追蹤的優勢

作者：Rys Sommefeldt



儘管在理論上實現現代 GPU 的方法是無限的，但真正有效的方法是確實的瞭解問題，並著手將方案變為現實。製造現代高性能半導體器件，以及試圖加速當前可編程光柵化技術所面臨的問題，顯示出 GPU 硬體產業發展的未來趨勢。

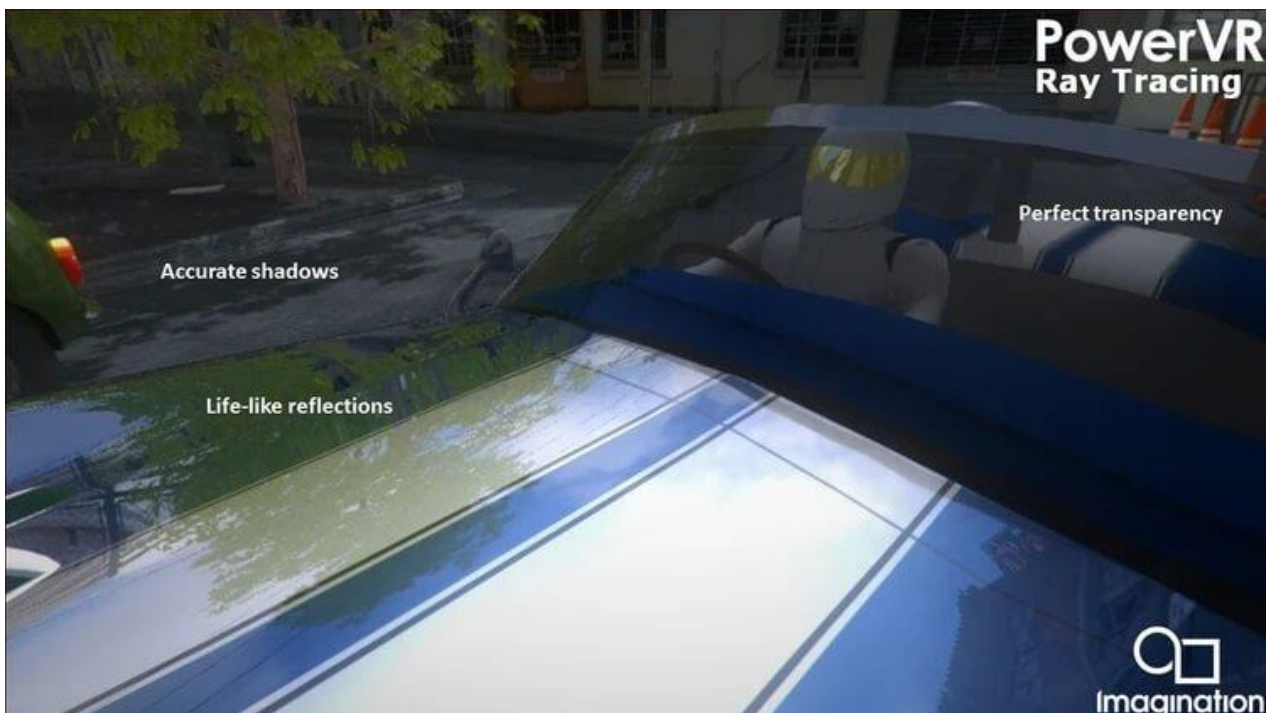
例如在現代 GPU 中 SIMD 處理和固定功能紋理單元是必不可少的，以至於不使用它們來設計的 GPU 方案，幾乎肯定意味著在研究之外不具有商業上的可行性和實用性。即使是過去 20 年來，任何一個 GPU 最瘋狂的願景，也沒有捨棄這些核心原則。安息吧，Larrabee（Intel 的 GPU 晶片）。

過去 15 年來[即時光線追蹤加速](#)一直被默認為是 GPU 設計中最令人煩心的問題，關於光線追蹤應該如何在 GPU 上實現的主要規格是微軟推出的 DXR，它要求的執行模型卻不能真正融入到 GPU 的工作模式，這無疑給任何需要支援它的 GPU 設計者帶來一些嚴重的潛在問題。如果即時光線追蹤是過去十年時間沒有考慮過的事情，那麼這個問題會更加明顯，而 Imagination 一直在關注這個問題。

光線追蹤面臨的關鍵挑戰

如果你依照 DXR 規格，思考以何種方式能有效提升 GPU 的運算，那麼你很快發現一些任何解決方案都會面臨到的問題：

首先，你需要一種方法來生成和處理一組包含幾何體的數據結構，進而能以更有效的方式根據幾何體來跟蹤光線。其次，當追蹤光線時，GPU 要測試光線是否與之相交，要提供一些用戶可定義的編程介面。第三，被追蹤的光線可以發出新的光線！ DXR 規格還需要考慮其他問題，但是從整體來看，這三個因素是最重要的。



PowerVR 光線追蹤混合渲染效果

為有效生成和使用加速數據結構來表現幾何體，代表 GPU 可能要用一個全新的執程序對光線做交互測試，然後我們需要用全新的介面函數處理這些新的數據結構，測試是否相交，之後在程式人員的控制下，根據相交測試的結果實現一些功能。GPU 是並行的設計，所以同時處理一堆光線意味著什麼？這樣做是否發現了新挑戰，而這些挑戰與傳統的幾何和圖元並行處理所帶來的挑戰卻大不相同？

上一個問題的答案是非常肯定的，的確這些差異對如何將光線追蹤映射到現有的 GPU 執行的模型中有著深遠的影響。這些 GPU 存在運算資源和記憶體的不平衡，導致資料存取成為一種寶貴的資源，而浪費這些資源是導致效率和性能低下的最主要原因之一。

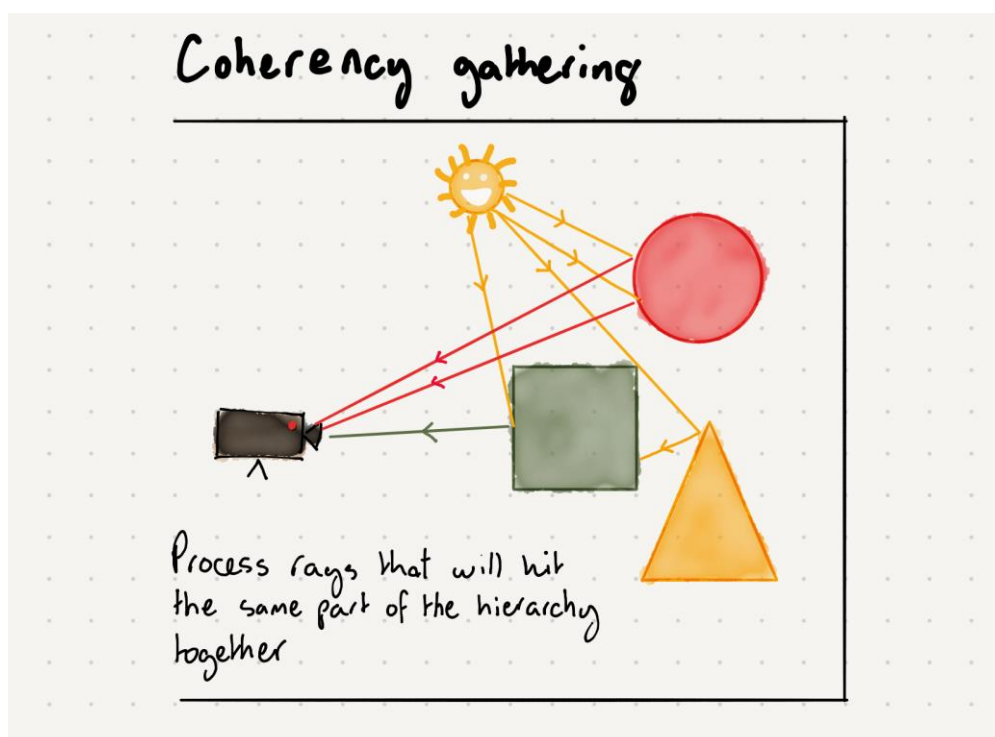
哦不！我們做了什麼？

GPU 被設計成可以充分以任何形式充分使用或與 DRAM 連結，不管是記憶體的空間開發或時間局部性，都能作為實現這一目的的方法。值得慶幸的是，最常見和最現代化的光柵化渲染有一個很好的特性，即在著色期間（尤其是圖元著色，通常是任何給定幀的主要工作負載）三角形和圖元頂點有可能與它們的近鄰共用相關數據。因此，您訪問一組圖元所需的任何緩存數據，很可能下一個相鄰的組將需要使用您已經從 DRAM 中提取並緩存的部分或全部內存數據。對於當今大多數柵格化渲染工作負載而言，這都是正確的，因此我們都可以鬆一口氣，並圍繞該屬性設計 GPU 架構。

當我們使用光線追蹤，這些就都失效了。光線追蹤使所有空間局部性消失。下面讓我們來分析其中的原因。

物體表面問題

最簡單的思考方式就是觀察四周，在你坐下來閱讀這篇文章時，注意光線在你所處環境中的作用。由於光線追蹤模擬了光線從所有光源傳播時的屬性，因此它必須處理光線照射在場景中任何表面時發生的情況。也許我們只關心光線照射哪些物體，也許物體的表面以均勻的方向散射光線，但它也可能完全是隨機的。也許表面吸收了所有的光，因此不會有次級光線的傳播。也許表面有一種材質屬性，使它能夠部分吸收幾乎所有照射來的光線，然後隨機散射剩餘不能吸收的少量光線。



只有第一種場景可以映射到 GPU 的利用記憶體存取局部性的工作模式，即使如此也只有當所有並行處理的光線都照射到同一類型的三角形時才可以。

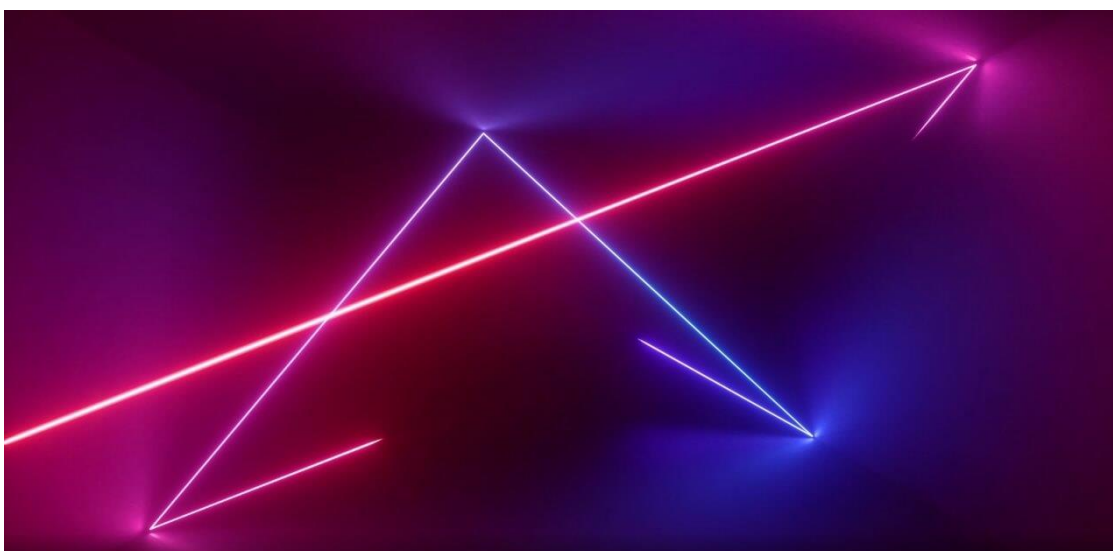
正是這種明顯分歧的可能性，導致了這些問題，如果並行處理的任何光線相互之間可能會有不同的作用，包括撞擊不同的加速數據結構或發出新的光線，那麼 GPU 能高效工作的基本前提就會被破壞，而且這通常比在傳統的幾何圖形，或像素處理中遇到的發散現象更具有破壞性。

相干性聚集

PowerVR 對光線追蹤硬體加速的實現所做的是，硬體光線追蹤和排序，它與當今業內任何其他硬體光線追蹤加速相比都是獨一無二的，這對軟體方面來說是完全透明的，確保硬體上並行追蹤的發射光線具有潛在的相似性。我們稱之為相干性聚集。

硬體維護了一個數據結構，用於層次化的記憶體發出的正在被硬體處理的光線，並能夠根據它們的方向按它們在加速結構中前進的位置進行選擇和分組。這意味著當它們被處理時更可能共用記憶體中被存取的加速數據結構中的數據，且額外的優勢是能夠最大化隨後要並行處理的光線-幾何體相交計算的數量。

透過分析由硬體調度的光線我們可以確保以 GPU 友好的方式對它們進行分組，從而更高效的進行後續處理，這些是該系統成功的關鍵，有助於避免打破 GPU 產業為高效的光柵化渲染而精心設計的運行模式，這就避免了光線追蹤硬體對特殊類型記憶體的需求，因此提供了與 GPU 的其他部分更容易集成的方案。



相干性聚集機制本身相當的複雜，因為它需要快速的追蹤，排序和調度所有的被提交到硬體中處理的光線，而不會反壓前級用於發射光線的調度系統，也不會造成後級以排序好的光線，和加速數據結構為輸入的硬體的空間。

如果沒有硬體系統來幫助 GPU 處理光線排序，那麼就需要依賴應用程式或遊戲開發人員，以某種方式在主機上處理光線的相干性問題，或者在 GPU 上加入一個中間的運算環節來處理光線排序—前提是這種方式被硬體所支援，以上假設的方式中沒有一個能在即時的硬體平台上提升效率和性能，然而 Imagination 是市場上唯一擁有這種硬體光線追蹤系統的 GPU IP 供應商。

跟著光潮走

我們之所以成為業內唯一針對硬體光線追蹤提供解決方案的供應商，是因為我們已經致力於解決這個問題很長時間了，與產業中其他正在緩慢的進展相比，[光線追蹤已經成為當今圖形技術廣泛採用 API 之一了](#)。

我們的相干性聚集特性與目前產業的光線追蹤相互相容（如果光線恰好發射出新的光線，堆棧將會被釋放，也可能發射出新的光線等等），在每個階段進行相干性聚集處理，並確保我們盡可能的實現硬體光線追蹤的強大性能。

在現代的硬體光線追蹤系統中最重要的是，測量光線束、峰值並行測試率或空光線發射和未命中率，這些是描述光線追蹤硬體性能的簡單方式，但是也並不是非常的有用，畢竟開發人員並不只關心高峰值並行測試率或缺失率。

我們的目標是在整個加速系統中使用全面的光線追蹤，這樣開發人員就可以用光線束預算要實現哪些有用的功能。我們的相干性聚集系統與我們提供的解決方案，共同達成了這個目標，[與業內的其他方案相比都是獨一無二的](#)。

--

不要錯過 Imagination 的消息，您可以在 [Twitter](#)，[Facebook](#) 和 [LinkedIn](#) 上關注我們。

原文鏈接：<https://www.imgtec.com/blog/coherency-gathering-in-ray-tracing-the-benefits-of-hardware-ray-tracking/>

聲明：本文為原創文章，轉載需註明作者、出處及原文鏈接，否則，本網站將保留追究其法律責任的權利。