I855201

(19)中華民國智慧財產局

(12)發明說明書公告本 **(11)證書號數：TW I855201 B**

(45)公告日： 中華民國 113 (2024) 年 09 月 11 日

(21)申請案號：109144270 (22)申請日： 中華民國 109 (2020) 年 12 月 15 日

(51)Int. Cl.： ***G06T1/20*** ***(2006.01)*** ***G06T7/20*** ***(2017.01)***

G06T5/20 ***(2006.01)***

(30)優先權：2020/03/15 美國 16/819,124

(71)申請人：美商英特爾股份有限公司 (美國) INTEL CORPORATION (US)
美國

(72)發明人：亞法拉 阿提拉 AFRA, ATTILA TAMAS (RO)；根瑟 約翰尼斯 GUENTHER, JOHANNES (DE)

(74)代理人：林志剛

(56)參考文獻：

TW 200949670A US 2017/0178387A1

網路文獻 Joyita Dutta, Richard M. Leahy, Quanzheng Li Non-Local Means Denoising of Dynamic PET Images 2013 www.plosone.org

審查人員：陳昱潭

申請專利範圍項數：19 項 圖式數：54 共 263 頁

(54)名稱

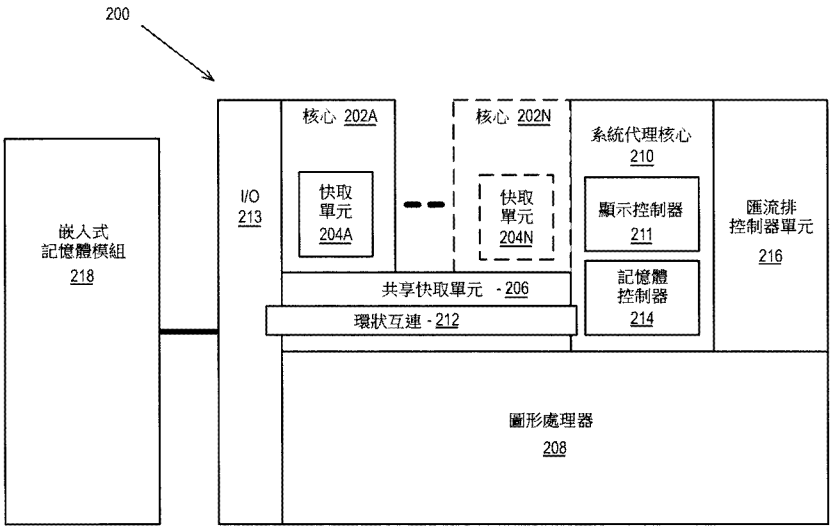
使用圖形處理器的運動估計電路執行非局部平均濾波之設備、方法及機器可讀取媒體

(57)摘要

用於使用圖形處理器的媒體處理器進行非局部平均濾波之設備及方法。例如，處理器的一個實施例包含：射線追蹤電路，其用以執行第一組的一或多個命令，以使射線遍歷通過邊界體積階層(BVH)，以識別與所述射線相交的 BVH 節點及/或基元；著色器執行電路，其用以回應於第二組的一或多個命令而執行一或多個著色器，以基於與所述射線相交的所述 BVH 節點及/或基元來繪製影像訊框序列；以及媒體處理器，其包含運動估計電路，所述運動估計電路用以執行第三組的一或多個命令，以基於跨所述影像訊框序列收集的平均像素值，執行非局部平均濾波以從所述影像訊框序列中移除雜訊。

Apparatus and method for non-local means filtering using a media processor of a graphics processor. For example, one embodiment of a processor comprises: ray tracing circuitry to execute a first set of one or more commands to traverse rays through a bounding volume hierarchy (BVH) to identify BVH nodes and/or primitives intersected by the ray; shader execution circuitry to execute one or more shaders responsive to a second set of one or more commands to render a sequence of image frames based on the BVH nodes and/or primitives intersected by the ray; and a media processor comprising motion estimation circuitry to execute a third set of one or more commands to perform non-local means filtering to remove noise from the sequence of image frames based on a mean pixel value collected across the sequence of image frames.

指定代表圖：



【圖 2A】

符號簡單說明：

200:處理器

202A-202N:處理器核心

204A-204N:內部快取單元

206:共享快取單元

208:整合圖形處理器

210:系統代理核心

211:顯示控制器

212:環狀互連

213:I/O 鏈路

214:整合記憶體控制器

216:匯流排控制器單元

218:高效能嵌入式記憶體模組

專利案號：109144270

113年03月01日 所提修正
民國 113 年 3 月 1 日修正

I855201

公告本

【發明摘要】

【中文發明名稱】

使用圖形處理器的運動估計電路執行非局部平均濾波之設備、方法及機器可讀取媒體

【英文發明名稱】

APPARATUS, METHOD AND MACHINE-READABLE MEDIUM FOR PERFORMING NON-LOCAL MEANS FILTERING USING MOTION ESTIMATION CIRCUITRY OF A GRAPHICS PROCESSOR

【中文】

用於使用圖形處理器的媒體處理器進行非局部平均濾波之設備及方法。例如，處理器的一個實施例包含：射線追蹤電路，其用以執行第一組的一或多個命令，以使射線遍歷通過邊界體積階層(BVH)，以識別與所述射線相交的BVH節點及/或基元；著色器執行電路，其用以回應於第二組的一或多個命令而執行一或多個著色器，以基於與所述射線相交的所述BVH節點及/或基元來繪製影像訊框序列；以及媒體處理器，其包含運動估計電路，所述運動估計電路用以執行第三組的一或多個命令，以基於跨所述影像訊框序列收集的平均像素值，執行非局部平均濾波以從所述影像訊框序列中移除雜訊。

【英文】

Apparatus and method for non-local means filtering using a media processor of a graphics processor. For example, one embodiment of a processor comprises: ray tracing circuitry to execute a first set of one or more commands to traverse rays through a bounding volume hierarchy (BVH) to identify BVH nodes and/or primitives intersected by the ray; shader execution circuitry to execute one or more shaders responsive to a second set of one or more commands to render a sequence of image frames based on the BVH nodes and/or primitives intersected by the ray; and a media processor comprising motion estimation circuitry to execute a third set of one or more commands to perform non-local means filtering to remove noise from the sequence of image frames based on a mean pixel value collected across the sequence of image frames.

【指定代表圖】第(2A)圖。
【代表圖之符號簡單說明】

200:處理器

202A-202N:處理器核心

204A-204N :內部快取單元

206:共享快取單元

208:整合圖形處理器

210:系統代理核心

211:顯示控制器

212:環狀互連

213:I/O鏈路

214:整合記憶體控制器

216:匯流排控制器單元

218:高效能嵌入式記憶體模組

【特徵化學式】無

【發明說明書】

【中文發明名稱】

使用圖形處理器的運動估計電路執行非局部平均濾波之設備、方法及機器可讀取媒體

【英文發明名稱】

APPARATUS, METHOD AND MACHINE-READABLE MEDIUM FOR PERFORMING NON-LOCAL MEANS FILTERING USING MOTION ESTIMATION CIRCUITRY OF A GRAPHICS PROCESSOR

【技術領域】

【0001】本發明總體上涉及圖形處理器領域。更具體地，本發明涉及一種用於執行穩定和短等待時間分類操作的設備及方法。

【先前技術】

【0002】射線追蹤是透過基於實體的繪製來模擬光傳輸的技術。直到幾年前，它還被廣泛用於電影繪製中，因此對於即時效能而言，它過於佔用資源。射線追蹤中的關鍵操作之一是處理稱為「射線遍歷」的射線-場景的相交的可見性查詢，其透過遍歷和相交定界體階層(BVH)中的節點來計算射線-場景的相交。

【圖式簡單說明】

【0003】可以從以下結合附圖的詳細描述中獲得對本發明的更好的理解，其中：

【0004】[圖 1]是具有包含一或多個處理器核心和圖形處理器的處理器的電腦系統的方塊圖；

【0005】[圖 2A-2D]顯示由本文描述的實施例提供的計算系統和圖形處理器；

【0006】[圖 3A-3C]顯示由本文描述的實施例提供的額外圖形處理器和計算加速器架構的方塊圖；

【0007】[圖 4]是用於圖形處理器的圖形處理引擎的方塊圖；

【0008】[圖 5A-5B]顯示根據本文描述的實施例包含在圖形處理器核心中採用的處理元件的陣列的執行緒執行邏輯 500；

【0009】[圖 6]顯示根據實施例的額外執行單元 600；

【0010】[圖 7]顯示圖形處理器執行單元指令格式；

【0011】[圖 8]是包含圖形管線、媒體管線、顯示引擎、執行緒執行邏輯和繪製輸出管線的圖形處理器的方塊圖；

【0012】[圖 9A]是顯示圖形處理器命令格式的方塊圖；

【0013】[圖 9B]是顯示圖形處理器命令序列的方塊圖；

【0014】[圖 10]顯示用於資料處理系統的範例性圖形軟體架構；

【0015】[圖 11A]顯示一個實施例的範例IP核心開發特徵；

【0016】[圖 11B-D]顯示用於不同實施例的各種封裝特徵；

【0017】[圖 12]顯示可以使用一或多個IP核心來製造的範例性系統單晶片積體電路；

【0018】[圖 13]顯示可以使用一或多個IP核心來製造的系統單晶片積體電路的範例性圖形處理器；

【0019】[圖 14]顯示包含圖 13的圖形處理器1310的一或多個MMU 1320A-1320B、快取1325A-1325B和電路互連1330A-1330B的圖形處理器1340；

【0020】[圖 15]顯示用於執行機器學習架構的初始訓練的架構；

【0021】[圖 16]顯示在運行時間期間如何對機器學習引擎進行連續訓練和更新；

【0022】[圖 17]顯示在運行時間期間如何對機器學習引擎進行連續訓練和更新；

【0023】[圖 18A-B]顯示如何在網路上共享機器學習資料；以及

【0024】[圖 19]顯示用於訓練機器學習引擎的方法；

【0025】[圖 20]顯示節點如何交換重影區域資料以執行分佈式降噪操作；

【0026】[圖 21]顯示跨複數個節點分佈的影像繪製和降噪操作的架構；

【0027】[圖 22]顯示用於分佈式繪製和降噪的架構的額外細節；

【0028】[圖 23]顯示用於執行分佈式繪製和降噪的方法；

【0029】[圖 24]顯示機器學習方法；

【0030】[圖 25]顯示複數個互連的通用圖形處理器；

【0031】[圖 26]顯示用於機器學習實現的一組卷積層和完全連接的層；

【0032】[圖 27]顯示卷積層的範例；

【0033】[圖 28]顯示機器學習實現中的一組互連節點的範例；

【0034】[圖 29]顯示類神經網路使用訓練資料集在其中學習的訓練框架；

【0035】[圖 30A]顯示模型平行性和資料平行性的範例；

【0036】[圖 30B]顯示系統單晶片(SoC)；

【0037】[圖 31]顯示包含射線追蹤核心和張量核心的處理架構；

【0038】[圖 32]顯示光束的範例；

【0039】[圖 33]顯示用於執行光束追蹤的設備；

【0040】[圖 34]顯示光束階層的範例；

【0041】[圖 35]顯示用於執行光束追蹤的方法；

【0042】[圖 36]顯示分佈式射線追蹤引擎的範例；

【0043】[圖 37-38]顯示在射線追蹤系統中執行的壓

縮；

【0044】[圖 39]顯示在射線追蹤架構上實現的方法；

【0045】[圖 40]顯示範例性混合射線追蹤設備；

【0046】[圖 41]顯示用於射線追蹤操作的堆疊；

【0047】[圖 42]顯示混合射線追蹤設備的額外細節；

【0048】[圖 43]顯示定界體階層；

【0049】[圖 44]顯示調用堆疊和遍歷狀態儲存；

【0050】[圖 45]顯示用於遍歷和相交的方法；

【0051】[圖 46A-B]顯示執行某些著色器如何需要多個分派週期；

【0052】[圖 47]顯示單一分派週期如何執行複數個著色器；

【0053】[圖 48]顯示單一分派週期如何執行複數個著色器；

【0054】[圖 49]顯示用於執行射線追蹤指令的架構；

【0055】[圖 50]顯示用於在執行緒內執行射線追蹤指令的方法；

【0056】[圖 51]顯示用於非同步射線追蹤的架構的一個實施例；

【0057】[圖 52]顯示射線遍歷引擎的一個實施例；

【0058】[圖 53]顯示用於執行非局部平均濾波的本發明的一個實施例；以及

【0059】[圖 54]顯示根據本發明的一個實施例的方法。

【發明內容】及【實施方式】

【0060】在下面的描述中，出於解釋的目的，闡述了許多具體細節，以便提供對以下描述的本發明的實施例的透徹理解。然而，對於本領域的技術人員將顯而易見的是，可以在沒有這些具體細節中的一些的情況下實踐本發明的實施例。在其它情況下，以方塊圖形式顯示眾所皆知的結構和裝置，以避免使本發明的實施例的基本原理難以理解。

範例性圖形處理器架構和資料類型

系統總覽

【0061】**圖1**是根據實施例的處理系統100的方塊圖。處理系統100可以用於單處理器桌上電腦系統、多處理器工作站系統或具有大量處理器102或處理器核心107的伺服器系統。在一實施例中，處理系統100是併入在系統單晶片(SoC)積體電路內的處理平台，以用於行動、手持或嵌入式裝置，如在具有連接到區域網路或廣域網路的有線或無線的物聯網(IoT)裝置內。

【0062】在一實施例中，處理系統100可以包含基於伺服器的遊戲平台；包含遊戲和媒體控制台的遊戲控制台；行動遊戲控制台、手持遊戲控制台或線上遊戲控制台、與其耦接或被整合在其內。在某些實施例中，處理系統100是行動電話、智慧型手機、平板計算裝置或諸如具

有低內部儲存容量的筆記型電腦之類的行動網際網路連接裝置的一部分。處理系統100還可以包含諸如智慧型手錶可穿戴裝置的可穿戴裝置；擴增實境(AR)或虛擬實境(VR)功能增強的智慧型眼鏡或服裝，以提供視覺、音訊或觸覺輸出，以補充現實世界中的視覺、音訊或觸覺體驗，或者提供文字、音訊、圖形、視訊、全息影像或視訊或觸覺反饋；其它擴增實境(AR)裝置；或其它虛擬實境(VR)裝置、與其耦接或被整合在其內。在一些實施例中，所述處理系統100包含電視或機上盒裝置，或者是其一部分。在一實施例中，處理系統100可以包含諸如公車、拖曳車、汽車、機車或電動機車、飛機或滑翔機(或其任何組合)之類的自動駕駛車輛、與其耦接或被整合在其內。自動駕駛車輛可以使用處理系統100來處理車輛周圍感知的環境。

【0063】在一些實施例中，一或多個處理器102各自包括一或多個處理器核心107，以處理指令，其在被運行時執行系統和使用者軟體的操作。在一些實施例中，一或多個處理器核心107中的至少一個配置成處理特定指令集109。在一些實施例中，指令集109可促進複雜指令集計算(CISC)、精簡指令集計算(RISC)或者經由超長指令字組(VLIW)的計算。一或多個處理器核心107可處理不同指令集109，其可包括促進其它指令集的仿真的指令。處理器核心107還可包括其它處理裝置，如數位訊號處理器(DSP)。

【0064】在一些實施例中，處理器102包括快取記憶

體104。取決於架構，處理器102能夠具有單一內部快取或者多級內部快取。在一些實施例中，快取記憶體在處理器102的各種部件之間共享。在一些實施例中，處理器102還使用外部快取(例如第3級(L3)快取或最終級快取(LLC))(未顯示)，其可使用已知快取一致性技術在處理器核心107之間共享。暫存器檔案106可附加地包含在處理器102中，並且可包括不同類型的暫存器，以用於儲存不同類型的資料(例如整數暫存器、浮點暫存器、狀態暫存器和指令指標暫存器)。一些暫存器可以是通用暫存器，而其它暫存器可以特定於處理器102的設計。

【0065】在一些實施例中，一或多個處理器102與一或多個介面匯流排110耦接，以便在處理器102與處理系統100中的其它部件之間傳送通訊訊號(如位址、資料或控制訊號)。在一實施例中，介面匯流排110可以是處理器匯流排，如直接媒體介面(DMI)匯流排的版本。然而，處理器匯流排並不局限於DMI匯流排，而是可包括一或多個周邊設備部件互連匯流排(例如PCI、PCI Express)、記憶體匯流排或者其它類型的介面匯流排。在一實施例中，(一或多個)處理器102包括整合記憶體控制器116和平臺控制器集線器130。記憶體控制器116促進記憶體裝置與處理系統100的其它部件之間的通訊，而平臺控制器集線器(PCH)130經由區域I/O匯流排來提供到I/O裝置的連接。

【0066】記憶體裝置120可以是動態隨機存取記憶體(DRAM)裝置、靜態隨機存取記憶體(SRAM)裝置、快閃記

憶體裝置、相變記憶體裝置或者具有用作執行緒記憶體的適當效能的一些其它記憶體裝置。在一實施例中，記憶體裝置120能夠作為處理系統100的系統記憶體進行操作，以便儲存資料122和指令121以供一或多個處理器102運行應用程式或執行緒時使用。記憶體控制器116還與可選外部圖形處理器118耦接，外部圖形處理器118可與處理器102中的一或多個圖形處理器108進行通訊，以執行圖形和媒體操作。在一些實施例中，圖形、媒體及/或計算操作可以由加速器112輔助，加速器112是可以配置成執行一組專門的圖形、媒體或計算操作的協同處理器。例如，在一實施例中，加速器112是用於最佳化機器學習或計算操作的矩陣乘法加速器。在一實施例中，加速器112是可用於與圖形處理器108協同執行射線追蹤操作的射線追蹤加速器。在一實施例中，外部加速器119可以代替加速器112或與加速器112一起使用。

【0067】在一些實施例中，顯示裝置111可以連接到(一或多個)處理器102。顯示裝置111可以是內部顯示裝置(如同行動電子裝置或膝上型裝置中一樣)或者經由顯示介面(例如DisplayPort等)所附接的外部顯示裝置中的一或多個。在一實施例中，顯示裝置111可以是頭戴式顯示器(HMD)，如供在虛擬實境(VR)應用或擴增實境(AR)應用中使用的立體顯示裝置。

【0068】在一些實施例中，平臺控制器集線器130使周邊設備能夠經由高速I/O匯流排來連接到記憶體裝置120

和處理器102。I/O周邊設備包括但不限於音訊控制器146、網路控制器134、韌體介面128、無線收發器126、觸控感測器125、資料儲存裝置124(例如，非揮發性記憶體、揮發性記憶體、硬碟驅動、快閃記憶體、NAND、3D NAND、3D XPoint等)。資料儲存裝置124能夠經由儲存介面(例如SATA)或者經由周邊匯流排(如周邊設備部件互連匯流排(例如PCI、PCI Express))進行連接。觸控感測器125可以包括觸控螢幕感測器、壓力感測器或指紋感測器。無線收發器126可以是Wi-Fi收發器、藍牙收發器或者行動網路收發器，如3G、4G、5G或長期演進(LTE)收發器。韌體介面128實現與系統韌體的通訊，並且可以是例如統一擴展韌體介面(UEFI)。網路控制器134能夠實現到有線網路的網路連接。在一些實施例中，高效能網路控制器(未顯示)與介面匯流排110耦接。在一實施例中，音訊控制器146是多通道高解析度音訊控制器。在一實施例中，處理系統100包括可選的舊有I/O控制器140，以用於將舊有(例如個人系統2(PS/2))裝置耦接到系統。平臺控制器集線器130還能夠連接到一或多個通用串列匯流排(USB)控制器142連接輸入裝置，如鍵盤和滑鼠143組合、相機144或者其它USB輸入裝置。

【0069】將會理解，所示處理系統100是範例性的而不是限制性的，也可使用不同地配置的其它類型的資料處理系統。例如，記憶體控制器116和平臺控制器集線器130的實例可整合到離散外部圖形處理器(如外部圖形處理器

118)中。在一實施例中，平臺控制器集線器130及/或記憶體控制器116可以在一或多個處理器102外部。例如，處理系統100可以包括外部記憶體控制器116和平臺控制器集線器130，其可配置為與(一或多個)處理器102進行通訊的系統晶片組內的記憶體控制器集線器和周邊控制器集線器。

【0070】例如，可以使用電路板(「底座」)，其上放置了諸如CPU、記憶體和其它部件之類的部件，意於提高散熱效能。在一些範例中，諸如處理器之類的處理部件位於底座的頂側，而諸如DIMM之類的近端記憶體位於底座的底側。由於此設計提供了增強的氣流，這些部件可以比一般系統以更高的頻率和功率等級操作，從而提高了效能。此外，底座被配置成與機架中的電源和資料通訊電纜盲目搭接，從而增強了其快速拆卸、升級、重新安裝及/或更換的能力。同樣地，位於底座上的各個部件(諸如處理器、加速器、記憶體和資料儲存驅動器)由於相互之間的間距增加而配置成易於升級。在說明性實施例中，部件額外地包含硬體證明特徵以證明其真實性。

【0071】資料中心可以利用支援多種其它網路架構(包含乙太網路和Omni-Path)的單一網路架構(「結構」)。底座可以透過光纖耦接到交換機，所述光纖相較於一般的雙絞線電纜(例如，類別5、類別5e、類別6等)提供了較高頻寬和較低延遲。由於高頻寬和低延遲的互連和網路架構，資料中心在使用中可能會池化資源(諸如記憶體、加速器(例如GPU、圖形加速器、FPGA、ASIC、類神經網路

及/或人工智慧加速器等等)，以及經過實體分解的資料儲存驅動器，並根據需要將其提供給計算資源(例如，處理器)，從而使計算資源能夠像區域資源一樣存取池中的資源。

【0072】電源供應或電源可以對處理系統100或本文描述的任何部件或系統提供電壓及/或電流。在一個範例中，電源供應包含AC到DC (交流到直流)適配器，以插入壁裝電源插座。這種AC電源可以是可再生能源(例如，太陽能)電源。在一範例中，電源包含DC電源，諸如外部AC至DC轉換器。在一範例中，電源或電源供應包含無線充電硬體，以經由與充電場的鄰近來充電。在一個範例中，電源可以包含內部電池、交流電源供應、基於運動的電源供應、太陽能電源供應或燃料電池電源。

【0073】**圖2A-2D**顯示了本文所述實施例提供的計算系統和圖形處理器。**圖2A-2D**中具有與本文中任何其它附圖的元件相同的元件符號(或名稱)的元件能夠根據與本文其它部分所述的方式相似的任何方式進行操作或者起作用，但不限於此。

【0074】**圖2A**是處理器200的實施例的方塊圖，處理器200具有一或多個處理器核心202A-202N、整合記憶體控制器214和整合圖形處理器208。處理器200可以包含高達並包含由虛線框表示的額外核心202N的額外核心。每個處理器核心202A-202N包含一或多個內部快取單元204A-204N。在一些實施例中，每個處理器核心還可以存取一或

多個共享快取單元206。內部快取單元204A-204N和共享快取單元206表示處理器200內的快取記憶體階層。快取記憶體階層可包括每個處理器核心內的至少一級指令和資料快取以及一或多級共享中間級快取，如第2級(L2)、第3級(L3)、第4級(L4)或其它級快取，其中外部記憶體之前的最高階快取排序為LLC。在一些實施例中，快取一致性邏輯保持各種快取單元206和204A-204N之間的一致性。

【0075】在一些實施例中，處理器200還可包括一或多個匯流排控制器單元216和系統代理核心210的集合。一或多個匯流排控制器單元216管理周邊匯流排的集合，如一或多個PCI或PCI express匯流排。系統代理核心210提供各種處理器部件的管理功能性。在一些實施例中，系統代理核心210包括一或多個整合記憶體控制器214，以管理對各種外部記憶體裝置(未顯示)的存取。

【0076】在一些實施例中，處理器核心202A-202N中的一或多個包括對同時多重執行緒的支援。在這種實施例中，系統代理核心210包括用於在多重執行緒處理期間協調和操作核心202A-202N的部件。系統代理核心210還可包括功率控制單元(PCU)，其包括調節處理器核心202A-202N和圖形處理器208的功率狀態的邏輯和部件。

【0077】在一些實施例中，處理器200還包括圖形處理器208，以運行圖形處理操作。在一些實施例中，圖形處理器208與共享快取單元206的集合以及包括一或多個整合記憶體控制器214的系統代理核心210耦接。在一些實施

例中，系統代理核心210還包括顯示控制器211，以便將圖形處理器輸出驅動到一或多個所耦接顯示器。在一些實施例中，顯示控制器211還可以是經由至少一個互連與圖形處理器所耦接的獨立模組，或者可整合在圖形處理器208內。

【0078】在一些實施例中，環狀互連單元212用來耦接處理器200的內部部件。但是可使用備選互連單元，如點對點互連、交換互連或者其它技術(包括本領域眾所周知的技術)。在一些實施例中，圖形處理器208經由I/O鏈路213與環狀互連212耦接。

【0079】範例性I/O鏈路213表示多種I/O互連的至少一個，包括封裝上I/O互連，其促進各種處理器部件與高效能嵌入式記憶體模組218(如eDRAM模組)之間的通訊。在一些實施例中，處理器核心202A-202N的每個和圖形處理器208可以將嵌入式記憶體模組218用作共享最終級快取。

【0080】在一些實施例中，處理器核心202A-202N是運行相同指令集架構的同質核心。在另一個實施例中，處理器核心202A-202N在指令集架構(ISA)態樣是異質的，其中處理器核心202A-202N的一或多個運行第一指令集，而其它核心的至少一個運行第一指令集的子集或者不同指令集。在一實施例中，處理器核心202A-202N在微架構態樣是異質的，其中具有相對較高功率消耗的一或多個核心與具有較低功率消耗的一或多個功率核心相耦接。在一實施

例中，處理器核心202A-202N在計算能力態樣是異質的。此外，處理器200能夠在一或多個晶片上或者作為具有所示部件以及其它部件的SoC積體電路來實現。

【0081】**圖2B**是根據本文描述的一些實施例的圖形處理器核心219的硬體邏輯的方塊圖。**圖2B**中具有與本文中任何其它附圖的元件相同的元件符號(或名稱)的元件能夠根據與本文其它部分所述的方式相似的任何方式進行操作或者起作用，但不限於此。圖形處理器核心219(有時稱為核心片)可以是模組化圖形處理器內的一或多個圖形核心。圖形處理器核心219是一個圖形核心片的範例，並且本文所述的圖形處理器可以基於目標功率和效能包絡包含多個圖形核心片。每個圖形處理器核心219可以包含與多個子核心221A-221F(也稱為子片)耦接的固定功能方塊230，其包含通用和固定功能邏輯的模組方塊。

【0082】在一些實施例中，固定功能方塊230包含幾何/固定功能管線231，其可以被圖形處理器核心219中的所有子核心共享，例如，在較低效能及/或較低功率的圖形處理器實現中。在各種實施例中，幾何/固定功能管線231包含3D固定功能管線(例如，如下述，如圖3和圖4所示的3D管線312)、視訊前端單元、執行緒產生器和執行緒分派器以及統一返回緩衝區管理器，所述管理器管理統一返回緩衝區(例如，如下述，圖4中的統一返回緩衝器418)。

【0083】在一實施例中，固定功能方塊230還包含圖

形SoC介面232、圖形微控制器233和媒體管線234。圖形SoC介面232提供系統單晶片積體電路中的圖形處理器核心219和其它處理器核心之間的介面。圖形微控制器233是可程式化的子處理器，其可配置成管理圖形處理器核心219的各種功能，包含執行緒分派、排程和搶占。媒體管線234(例如，圖3和圖4的媒體管線316)包含有助於對包含影像和視訊資料的多媒體資料進行解碼、編碼、預處理及/或後處理的邏輯。媒體管線234經由請求計算或取樣子核心221A-221F內的邏輯來實現媒體操作。

【0084】在一實施例中，SoC介面232致使圖形處理器核心219以與SoC內的通用應用處理器核心(例如，CPU)及/或其它部件進行通訊，包含諸如共享的最末級快取記憶體、系統RAM及/或嵌入在晶片或嵌入在封裝的DRAM的記憶體階層元件。SoC介面232還可以致使與SoC中的固定功能裝置(諸如相機成像管線)的通訊，並致使使用及/或實現可以在SoC中的圖形處理器核心219和CPU之間共享的全域記憶體原子。SoC介面232還可以實現用於圖形處理器核心219的電源管理控制，並啟用圖形核心219的時脈域和SoC中其它時脈域之間的介面。在一實施例中，SoC介面232致使從命令串流器和全域執行緒分派器接收命令緩衝器，所述命令串流器和全域執行緒分派器被配置成向圖形處理器內的一或多個圖形核心中之各者提供命令和指令。當要執行媒體操作時，可以將所述命令和指令分派到媒體管線234，或者當要執行圖形處理操作時，可以將所述命

令和指令分派到幾何和固定功能管線(例如，幾何和固定功能管線231、幾何與固定功能管線237)。

【0085】圖形微控制器233可以被配置成執行用於圖形處理器核心219的各種排程和管理任務。在一實施例中，圖形微控制器233可以在子核心221A-221F內部的執行單元(EU)陣列222A-222F、224A-224F中的各種圖形平行引擎上執行圖形及/或計算工作負載排程。在此排程模型中，在包含圖形處理器核心219的SoC的CPU核心上執行的主機軟體可以提交多個圖形處理器門鈴(doorbells)之一的工作負載，其調用適當的圖形引擎上的排程操作。排程操作包含確定接下來要運行的工作負載，將工作負載提交給命令串流器、搶占引擎上正在運行的現有工作負載、監視工作負載的進度以及在工作負載完成時通知主機軟體。在一實施例中，圖形微控制器233還可以促進圖形處理器核心219的低功率或閒置狀態，提供圖形處理器核心219具有獨立於系統上的作業系統及/或圖形驅動器軟體跨低功率狀態轉換在圖形處理器核心219內部保存和恢復暫存器的能力。

【0086】圖形處理器核心219可具有大於或小於所示的子核心221A-221F，最多*N*個模組化子核心。對於每組*N*個子核心，圖形處理器核心219也可以包含共享的功能邏輯235、共享及/或快取記憶體236、幾何/固定功能管線237以及額外的固定功能邏輯238，以加速各種圖形和計算處理操作。共享的功能邏輯235可以包含與圖4的共享功能邏

輯420相關的邏輯單元 (例如，取樣器、數學及/或執行緒間通訊邏輯)，其可以由圖形處理器核心219內部的每個*N*個子核心共享。共享及/或快取記憶體236可以是圖形處理器核心219中的所述組*N*個子核心221A-221F的最末級快取，並且還可以用作可由多個子核心存取的共享記憶體。幾何/固定功能管線237可以被包含來代替固定功能方塊230內的幾何/固定功能管線231並且可以包含相同或相似的邏輯單元。

【0087】在一實施例中，圖形處理器核心219包含額外的固定功能邏輯238，其可以包含供圖形處理器核心219使用的各種固定功能加速邏輯。在一實施例中，額外的固定功能邏輯238包含用於僅定位著色的額外的幾何管線。在僅定位著色中，存在兩個幾何管線、幾何/固定功能管線238、231內的完整幾何管線和剔除管線，其為額外的幾何管線，可以被包含在額外的固定功能邏輯238內。在一實施例中，剔除管線是完整幾何管線的修整版。完整管線和剔除管線可以執行同一應用程式的不同實例，每個實例具有單獨的上下文。僅定位著色可以隱藏丟棄三角形的長剔除運行，從而使著色在某些情況下可以更早完成。例如，在一實施例中，在額外的固定功能邏輯238內的剔除管線邏輯可以與主應用程式平行執行位置著色器，並且通常比完整管線更快地產生關鍵結果，因為剔除管線僅獲取和著色頂點的位置屬性，而無需對於訊框緩衝區執行像素的光柵化和繪製。剔除管線可以使用產生的臨界結果來計

算所有三角形的可見性資訊，而無需考慮那些三角形是否被剔除。完整的管線(在這種實例中，其可以稱為重播管線)可以使用可見性資訊來跳過剔除的三角形，以僅著色最終傳遞到柵格化階段的可見三角形。

【0088】在一實施例中，額外的固定功能邏輯238還可以包含機器學習加速邏輯，諸如固定功能矩陣乘法邏輯，以供實現包含針對機器學習訓練或推理的最佳化。

【0089】在每個圖形子核心221A-221F中包含一組執行資源，其可用於回應於圖形管線、媒體管線或著色器程式的請求來執行圖形、媒體和計算操作。圖形子核心221A-221F包含多個EU陣列222A-222F、224A-224F、執行緒分派和執行緒間通訊(TD/IC)邏輯223A-223F、3D(例如，紋理)取樣器225A-225F、媒體取樣器206A-206F、著色器處理器227A-227F和共享區域記憶體(SLM)228A-228F。EU陣列222A-222F、224A-224F每個都包含多個執行單元，其為通用圖形處理單元，能夠為圖形、媒體或計算操作(包含圖形、媒體或計算著色器程式)提供服務，以執行浮點和整數/定點邏輯運算。TD/IC邏輯223A-223F對子核心內的執行單元執行區域執行緒分配和執行緒控制操作，並促進在子核心的執行單元上執行的執行緒之間的通訊。3D取樣器225A-225F可以將與紋理或其它3D圖形相關的資料讀取到記憶體中。3D取樣器可以根據配置的取樣狀態和與給定紋理關聯的紋理格式不同地讀取紋理資料。媒體取樣器206A-206F可以基於與媒體資料相關的類型和格

式來執行類似的讀取操作。在一實施例中，每個圖形子核心221A-221F可以可替代地包含統一的3D和媒體取樣器。在每個子核心221A-221F中的執行單元上執行的執行緒可以利用在每個子核心內的共享區域記憶體228A-228F，以使在執行緒組中執行的執行緒能夠使用公用的晶載記憶體池執行。

【0090】**圖2C**顯示圖形處理單元(GPU)239，其包含佈置成多核心組240A-240N的專用圖形處理資源集。雖然只提供了單一多核心組240A的細節，應當理解，其它多核心組240B-240N可以配備相同或相似的圖形處理資源集。

【0091】如圖所示，多核心組240A可以包含一組圖形核心243、一組張量核心244和一組射線追蹤核心245。排程器/分派器241排程並分派圖形執行緒以在各種核心243、244、245上執行。當執行圖形執行緒時，一組暫存器檔案242儲存由核心243、244、245使用的運算元值。這些可以包含例如用於儲存整數值的整數暫存器、用於儲存浮點值的浮點暫存器、用於儲存緊縮資料元件(整數及/或浮點資料元件)的向量暫存器以及用於儲存張量/矩陣值的切片暫存器。在一實施例中，切片暫存器被實現為向量暫存器的組合集合。

【0092】一或多種組合的1級(L1)快取和共享記憶體單元247在每個多核心組240A內區域地儲存諸如紋理資料、頂點資料、像素資料、射線資料、定界體資料等的圖形資料。一或多個紋理單元247也可以用於執行紋理化操

作，諸如紋理映射和取樣。由多核心組240A-240N中的全部或部分子集共享的2級(L2)快取253儲存用於多個並發圖形執行緒的圖形資料及/或指令。如圖所示，L2快取253可以跨複數個多核心組240A-240N共享。一或多個記憶體控制器248將GPU 239耦接到記憶體249，其可以是系統記憶體(例如，DRAM)及/或專用圖形記憶體(例如，GDDR6記憶體)。

【0093】輸入/輸出(I/O)電路250將GPU 239耦接到一或多個I/O裝置252，諸如數位訊號處理器(DSP)、網路控制器或使用者輸入裝置。晶載互連可用於將I/O裝置252耦接到GPU 239和記憶體249。I/O電路250的一或多個I/O記憶體管理單元(IOMMU)251將I/O裝置252直接耦接到系統記憶體249。在一實施例中，IOMMU 251管理多組頁表，以將虛擬位址映射到系統記憶體249中的實體位址。在此實施例中，I/O裝置252、CPU 246和GPU 239可以共享相同的虛擬位址空間。

【0094】在一種實現中，IOMMU 251支援虛擬化。在這種情況下，可以管理用於將訪客/圖形虛擬位址映射到訪客/圖形實體位址的第一組頁表，以及用於將訪客/圖形實體位址映射到系統/主機實體位址的第二組頁表(例如，在系統記憶體249內)。第一和第二組頁表中的每一個的基底位址可以儲存在控制暫存器中，並在上下文切換器上換出(例如，以便向新的上下文提供對相關的頁表集的存取)。儘管未在圖2C中顯示，但核心243、244、245及/或

多核心組240A-240N中的每一個可以包含轉譯後備緩衝區(TLB)，以將訪客虛擬快取到訪客實體轉譯、將訪客實體快取到主機實體轉譯以及將訪客虛擬快取到主機實體轉譯。

【0095】在一實施例中，CPU 246、GPU 239和I/O裝置252被整合在單一半導體晶片及/或晶片封裝上。所示的記憶體249可以整合在同一晶片上，或者可以透過晶片外介面耦接到記憶體控制器248。在一種實現中，記憶體249包含與其它實體系統級記憶體共享相同虛擬位址空間的GDDR6記憶體，儘管本發明的基本原理不限於所述特定實現。

【0096】在一實施例中，張量核心244包含複數個專門設計用於執行矩陣運算的執行單元，矩陣運算是用於執行深度學習運算的基本計算運算。例如，同時矩陣乘法運算可用於類神經網路訓練和推理。張量核心244可以使用各種運算元精確度來執行矩陣處理，包含單精確度浮點數(例如32位元)、半精確度浮點數(例如16位元)、整數字組(16位元)、位元組(8位元)和半位元組(4位元)。在一實施例中，類神經網路實現提取每個繪製場景的特徵，潛在地從多個訊框中組合細節，以構建高品質的最終影像。

【0097】在深度學習實現中，可以排程平行矩陣乘法工作以在張量核心244上執行。類神經網路的訓練尤其需要大量的矩陣點積運算。為了處理N×N×N矩陣乘法的內積公式，張量核心244可以包含至少N個點積處理元件。在矩

陣乘法開始之前，將一個完整的矩陣載入到切片暫存器中，並在N個週期中每個週期載入第二矩陣的至少一行。每個週期有N個點積被處理。

【0098】取決於特定實現，可以用不同的精確度儲存矩陣元件，包含16位字組、8位元位元組(例如INT8)和4位元半位元組(例如INT4)。對於張量核心244可以指定不同的精確度模式，以確保將最有效的精確度用於不同的工作負載(例如，諸如可以將量化容忍到位元組和半位元組的推理工作負載)。

【0099】在一實施例中，射線追蹤核心245為即時射線追蹤和非即時射線追蹤實現加速了射線追蹤操作。特別是射線追蹤核心245包含射線遍歷/相交電路，以供使用定界體階層(BVH)來執行射線遍歷並識別射線與包圍在BVH容積內的基元之間的相交。射線追蹤核心245還可包含用於執行深度測試和剔除的電路(例如，使用Z緩衝器或類似佈置)。在一種實現中，射線追蹤核心245與本文所述的影像降噪技術一致地執行遍歷和相交操作，其中至少一部分可以在張量核心244上執行。例如，在一實施例中，張量核心244實現深度學習類神經網路，以對由射線追蹤核心245產生的訊框執行降噪。然而，CPU 246、圖形核心243及/或射線追蹤核心245還可以實現全部或部分的降噪及/或深度學習演算法。

【0100】此外，如上所述，可以採用分佈式降噪方法，其中GPU 239位於透過網路或高速互連與其它計算裝

置耦接的計算裝置中。在此實施例中，互連的計算裝置共享類神經網路學習/訓練資料，以提高整個系統學習針對不同類型的影像訊框及/或不同圖形應用執行降噪的速度。

【0101】在一實施例中，射線追蹤核心245處理所有BVH遍歷和射線-基元的相交，從而節省了圖形核心243不會因每條射線數千條指令而過載。在一實施例中，每個射線追蹤核心245包含用於執行定界框測試(例如，用於遍歷操作)的第一組專用電路和用於執行射線-三角形相交測試(例如，已被遍歷的射線相交)的第二組專用電路。因此，在一實施例中，多核心組240A可以簡單地發射射線探測器，並且射線追蹤核心245獨立地執行射線遍歷和相交並將命中資料(例如，命中、未命中、多次命中等)返回到執行緒上下文。其它核心243、244被釋放來執行其它圖形或者計算工作，同時射線追蹤核心245執行遍歷和相交操作。

【0102】在一實施例中，每個射線追蹤核心245包含執行BVH測試操作的遍歷單元和執行射線-基元相交測試的相交單元。相交單元產生「命中」、「未命中」或「多次命中」回應，並提供給適當的執行緒。在遍歷和相交操作期間，其它核心的執行資源(例如，圖形核心243和張量核心244)被釋放以執行其它形式的圖形工作。

【0103】在以下描述的一個特定實施例中，使用了一種混合光柵化/射線追蹤方法，其中工作在圖形核心243和

射線追蹤核心245之間分佈。

【0104】在一實施例中，射線追蹤核心245(及/或其它核心243、244)包含用於射線追蹤指令集的硬體支援，諸如Microsoft的DirectX射線追蹤(DXR)，其包含DispatchRays命令，以及射線產生、最接近命中、任何命中和未命中的著色器，其為每個物件啟用唯一的一組著色器和紋理分配。可能會被射線追蹤核心245、圖形核心243和張量核心244支援的另一個射線追蹤平台是Vulkan 1.1.85。然而，請注意，本發明的基本原理不限於任何特定的射線追蹤ISA。

【0105】通常，各種核心245、244、243可以支援的射線追蹤指令集，包含用於射線產生、最接近命中、任何命中、射線基元相交、每一基元和分層定界框結構、未命中、訪問和異常的指令/功能。更具體地，一個實施例包含執行以下功能的射線追蹤指令：

射線產生–可以為每個像素、樣本或其它使用者定義的工作分配執行射線產生指令。

最接近命中–可以執行最接近命中指令來定位射線與場景中基元的最接近交點。

任何命中–任何命中指令可識別射線與場景中基元之間的多個相交點，從而有可能識別新的最接近相交點。

相交–相交指令執行射線-基元相交測試並輸出結果。

每個基元定界框結構–此指令建立圍繞給定的基元或一組基元的定界框(例如，在建立新的BVH或其它加速度資料結構時)。

未命中–表示射線未命中場景中內的所有幾何，或場景中指定區域。

訪問–表示射線將遍歷的子容積。

異常–包含各種類型的異常處理程序(例如，針對各種錯誤條件調用的)。

【0106】**圖2D**是根據本文描述的實施例的可以被配置成圖形處理器及/或計算加速器的通用圖形處理單元(GPGPU)270的方塊圖。GPGPU 270可以經由一或多個系統及/或記憶體匯流排與主機處理器(例如，一或多個CPU 246)和記憶體271、272互連。在一實施例中，記憶體271是可以與一或多個CPU 246共享的系統記憶體，而記憶體272是專用於GPGPU 270的裝置記憶體。在一實施例中，GPGPU 270內的部件和裝置記憶體272可以被映射到一或多個CPU 246可存取的記憶體位址。透過記憶體控制器268可以有利於記憶體271和272的存取。在一實施例中，記憶體控制器268包含內部直接記憶體存取(DMA)控制器269，或者可以包含用以執行否則由DMA控制器執行的操作的邏輯。

【0107】GPGPU 270包含多個快取記憶體，包含L2快取253、L1快取254、指令快取255，以及共享記憶體256，其至少一部分也可以被劃分為快取記憶體。GPGPU 270還包含多個計算單元260A-260N。每個計算單元260A-260N包含一組向量暫存器261、純量暫存器262、向量邏輯單元263，以及純量邏輯單元264。計算單元260A-260N也可以

包含區域共享記憶體265和程式計數器266。計算單元260A-260N可與常數快取267耦接，其可用於儲存常數資料，所述常數資料是在GPGPU 270上執行的核心(kernel)或著色器程式的運行期間不會更改的資料。在一實施例中，常數快取267是純量資料快取，快取的資料可直接提取到純量暫存器262。

【0108】在操作期間，一或多個CPU 246可以將命令寫到GPGPU 270中已映射到可存取位址空間的暫存器或記憶體中。命令處理器257可以從暫存器或記憶體讀取命令，並確定將如何在GPGPU 270內處理這些命令。接著可以使用執行緒分派器258將執行緒分派給計算單元260A-260N以執行這些命令。每個計算單元260A-260N可以獨立於其它計算單元執行執行緒。此外，每個計算單元260A-260N可以獨立地配置用於條件計算，並且可以有條件地將計算結果輸出到記憶體。當所提交的命令完成時，命令處理器257可以中斷一或多個CPU 246。

【0109】**圖3A-3C**顯示由本文描述的實施例提供的額外的圖形處理器和計算加速器架構的方塊圖。具有與本文中任何其它附圖的元件相同的元件符號(或名稱)的圖3A-3C的元件可以用類似於本文中其它地方所描述的任何方式來操作或起作用，但不限於此。

【0110】圖3A是圖形處理器300的方塊圖，圖形處理器300可以是離散圖形處理單元，或者可以是與複數個處理核心整合的圖形處理器或其它半導體裝置(諸如但不限

於記憶體裝置或網路介面)整合的圖形處理器。在一些實施例中，圖形處理器經由到圖形處理器上的暫存器的記憶體映射I/O介面並且採用放入處理器記憶體的命令進行通訊。在一些實施例中，圖形處理器300包括存取記憶體的記憶體介面314。記憶體介面314可以是到區域記憶體、一或多個內部快取、一或多個共享外部快取及/或到系統記憶體的介面。

【0111】在一些實施例中，圖形處理器300還包括顯示控制器302，以便將顯示輸出資料驅動到顯示裝置318。顯示控制器302包括用於視訊或使用者介面元件的多層的顯示和組成的一或多個覆蓋平面的硬體。顯示裝置318可以是內部或外部顯示裝置。在一實施例中，顯示裝置318是頭戴式顯示裝置，如虛擬實境(VR)顯示裝置或擴增實境(AR)顯示裝置。在一些實施例中，圖形處理器300包括用於對媒體進行編碼、解碼或者向一或多個媒體編碼格式、從一或多個媒體編碼格式或在一或多個媒體編碼格式之間對媒體進行轉碼的視訊編解碼器引擎306，所述一或多個媒體編碼格式包括但不限於動態影像專家組(MPEG)格式(諸如MPEG-2)、高階視訊編碼(AVC)格式(諸如H.264/MPEG-4 AVC、H.265/HEVC、開放媒體聯盟(AOMedia)VP8、VP9)、以及電影&電視工程師協會(SMPTE)421M/VC-1和聯合影像專家組(JPEG)格式(諸如JPEG、以及動態JPEG(MJPEG)格式)。

【0112】在一些實施例中，圖形處理器300包括區塊

影像轉換(block image transfer；BLIT)引擎304以執行二維(2D)光柵器(rasterizer)操作，包括，例如，位元邊界區塊轉換。然而，在一實施例中，2D圖形操作係使用圖形處理引擎(GPE)310的一或多個部件來執行。在一些實施例中，GPE 310為用於執行圖形操作(包括三維(3D)圖形操作及媒體操作)的計算引擎。

【0113】在一些實施例中，GPE 310包括用於執行3D操作(諸如使用作用於3D基元形狀(例如，矩形、三角形等等)之處理功能來繪製三維影像與場景)的3D管線312。3D管線312包括可程式化及固定功能元件，其執行元件內之各種任務及/或產生執行緒至3D/媒體子系統315。雖然3D管線312可被使用以執行媒體操作，GPE 310的實施例亦包括媒體管線316，其係被特定使用以執行媒體操作，如視訊後處理(post-processing)及影像加強。

【0114】在一些實施例中，媒體管線316包括固定功能或可程式化邏輯單元以執行一或多個專門的媒體操作，如視訊解碼加速、視訊去交錯(de-interlacing)、及視訊編碼加速(取代、或代表視訊編解碼器引擎306)。在一些實施例中，媒體管線316額外地包括以產生用於在3D/媒體子系統315上執行的執行緒產生單元。產生的執行緒執行對於在包括於3D/媒體子系統315中的一或多個圖形執行單元上的媒體操作的計算。

【0115】在一些實施例中，3D/媒體子系統315包括用於執行藉由3D管線312與媒體管線316而產生的執行緒之邏

輯。在一實施例中，該等管線發送執行緒執行請求至3D/媒體子系統315，其包括用於仲裁(arbitrating)及配送各種請求至可用的執行緒執行資源的執行緒配送邏輯。執行資源包括圖形執行單元之陣列以處理3D與媒體執行緒。在一些實施例中，3D/媒體子系統315包括用於執行緒指令與資料之一或多個內部快取。在一些實施例中，子系統亦包括共用記憶體(包括暫存器與可定址記憶體)以共用在執行緒之間的資料並儲存輸出資料。

【0116】**圖3B**顯示根據本文所述實施例的具有平鋪架構的圖形處理器320。在一實施例中，圖形處理器320包含圖形處理引擎集群322，其具有在圖形引擎塊310A-310D內的圖3A的圖形處理引擎310的多個實例。每個圖形引擎塊310A-310D可以透過一組塊互連323A-323F來互連。每個圖形引擎塊310A-310D也可以透過記憶體互連325A-325D連接到記憶體模組或記憶體裝置326A-326D。記憶體裝置326A-326D可以使用任何圖形記憶體技術。例如，記憶體裝置326A-326D可以是圖形雙倍資料速率(GDDR)記憶體。在一實施例中，記憶體裝置326A-326D是高頻寬記憶體(HBM)模組，其可以與它們各自的圖形引擎塊310A-310D在晶粒上。在一實施例中，記憶體裝置326A-326D可以是堆疊在它們各自的圖形引擎塊310A-310D頂部的堆疊記憶體裝置。在一實施例中，每個圖形引擎塊310A-310D及其關聯記憶體326A-326D駐留在獨立小晶片上，其結合到基礎晶粒或基礎基板上，如圖11B-11D中進一步詳細描

述的。

【0117】圖形處理引擎集群322可以與晶片上或封裝上結構互連324連接。結構互連324可以實現圖形引擎塊310A-310D和諸如視訊編解碼器306和一或多個複製引擎304的部件之間的通訊。複製引擎304可用於將資料移出、移入記憶體裝置326A-326D和在記憶體裝置326A-326D與圖形處理器320外部的記憶體 (例如，系統記憶體)之間移動。結構互連324也可以用於將圖形引擎塊310A-310D互連。圖形處理器320可以選擇性地包含顯示控制器302以啟用與外部顯示裝置318的連接。圖形處理器也可以配置成圖形或計算加速器。在加速器配置中，顯示控制器302和顯示裝置318可以被省略。

【0118】圖形處理器320可以透過主機介面328連接到主機系統。主機介面328可以啟用圖形處理器320、系統記憶體及/或其它系統部件之間的通訊。主機介面328可以是例如PCI express匯流排或另一類型的主機系統介面。

【0119】**圖3C**顯示根據本文描述的實施例的計算加速器330。計算加速器330可以包含與圖3B的圖形處理器320相似的架構，並且被最佳化用於計算加速。計算引擎集群332可以包含一組計算引擎塊340A-340D，其包含針對平行或基於向量的通用計算操作進行了最佳化的執行邏輯。在某些實施例中，計算引擎塊340A-340D不包含固定功能圖形處理邏輯，儘管在一實施例中，一或多個計算引擎塊340A-340D可以包含用以執行媒體加速的邏輯。計算引擎

塊340A-340D可以透過記憶體互連325A-325D連接到記憶體326A-326D。記憶體326A-326D和記憶體互連325A-325D的技術可以與圖形處理器320相似，或者可以不同。圖形計算引擎塊340A-340D也可以透過一組塊互連323A-323F來互連，並且可以與結構互連324連接及/或透過結構互連324互連。在一實施例中，計算加速器330包含大型L3快取336，其可以被配置成裝置範圍的快取。計算加速器330還可以用與圖3B的圖形處理器320類似的方式透過主機介面328連接到主機處理器和記憶體。

圖形處理引擎

【0120】圖4是根據一些實施例的圖形處理器的圖形處理引擎410的方塊圖。在一實施例中，圖形處理引擎(GPE)410是圖3A所示GPE 310的一個版本，並且也可以表示圖3B的圖形引擎塊310A-310D。圖4中具有與本文中任何其它附圖的元件相同的元件符號(或名稱)的元件能夠根據與本文其它部分所述的方式相似的任何方式進行操作或者起作用，但不限於此。例如，顯示了圖3A的3D管線312和媒體管線316。媒體管線316在GPE 410的一些實施例中是可選的，並且可以沒有明確地包含在GPE 410中。例如並且在至少一個實施例中，獨立媒體及/或影像處理器係耦接到GPE 410。

【0121】在一些實施例中，GPE 410耦接到或者包括命令串流器403，其向3D管線312及/或媒體管線316提供命

令流。在一些實施例中，命令串流器403耦接到記憶體，其可以是系統記憶體或者內部快取記憶體和共享快取記憶體中的一或多個。在一些實施例中，命令串流器403從記憶體接收命令，並且將命令發送給3D管線312及/或媒體管線316。命令是從環形緩衝器所獲取的指示，環形緩衝器儲存3D管線312和媒體管線316的命令。在一實施例中，環形緩衝器還可以包括儲存成批多個命令的批次命令緩衝器。3D管線312的命令還可以包括對記憶體中儲存的資料的參照，諸如但不限於3D管線312的頂點和幾何資料及/或媒體管線316的影像資料和記憶體物件。3D管線312和媒體管線316透過經由相應管線中的邏輯執行操作或者透過向圖形核心陣列414分派一或多個執行執行緒來處理命令和資料。在一實施例中，圖形核心陣列414包括一或多個圖形核心(例如(一或多個)圖形核心415A、(一或多個)圖形核心415B)方塊，每個方塊包括一或多個圖形核心。每個圖形核心包括圖形執行資源集合，其包括執行圖形和計算操作的通用和圖形特定執行邏輯以及固定功能紋理處理及/或機器學習和人工智慧加速邏輯。

【0122】在各個實施例中，3D管線312包括固定功能和可程式化邏輯，以透過處理指令並且將執行執行緒分派給圖形核心陣列414來處理一或多個著色器程式，諸如頂點著色器、幾何著色器、像素著色器、片段著色器、計算著色器或者其它著色器程式。圖形核心陣列414提供統一執行資源方塊，以供在處理這些著色器程式中使用。圖形

核心陣列414的(一或多個)圖形核心415A-415B內的多用途執行邏輯(例如執行單元)包括對各種3D API著色器語言的支援，並且能夠運行與多個著色器關聯的多個同時執行執行緒。

【0123】在一些實施例中，圖形核心陣列414包括執行邏輯，以執行媒體功能，諸如視訊及/或影像處理。在一實施例中，執行單元包括通用邏輯，其可程式化以便除了圖形處理操作之外還執行平行通用計算操作。通用邏輯能夠與圖1的(一或多個)處理器核心107或者如圖2A中的核心202A-202N內的通用邏輯平行地執行處理操作或者結合。

【0124】由圖形核心陣列414上運行的執行緒所產生的輸出資料能夠將資料輸出到統一返回緩衝器(unified return buffer；URB)418中的記憶體。URB 418能夠儲存多個執行緒的資料。在一些實施例中，URB 418可用來在圖形核心陣列414上運行的不同執行緒之間發送資料。在一些實施例中，URB 418還可用於圖形核心陣列與共享功能邏輯420內的固定功能邏輯上的執行緒之間的同步。

【0125】在一些實施例中，圖形核心陣列414是可縮放的，使得所述陣列包括可變數量的圖形核心，其各自具有基於GPE 410的目標功率和效能等級的可變數量的執行單元。在一實施例中，執行資源是動態可縮放的，使得執行資源可根據需要來啟用或禁用。

【0126】圖形核心陣列414耦接到共享功能邏輯420，

其包括在圖形核心陣列中的圖形核心之間所共享的多個資源。共享功能邏輯420內的共享功能是硬體邏輯單元，其向圖形核心陣列414提供專用補充功能性。在各個實施例中，共享功能邏輯420包括但不限於取樣器421、數學邏輯422和執行緒間通訊(ITC)邏輯423。另外，一些實施例實現共享功能邏輯420內的一或多個快取425。

【0127】一種共享功能至少實現在對給定專用功能的需求不足以包含在圖形核心陣列414內的情況下。替代地，所述專用功能的單一例示被實現為共享功能邏輯420中的獨立實體並且在圖形核心陣列414內的執行資源之間共享。在圖形核心陣列414之間共享並且包含在圖形核心陣列414內的功能的準確集合跨實施例而改變。在一些實施例中，共享功能邏輯420內由圖形核心陣列414廣泛使用的特定共享功能可包含在圖形核心陣列414內的共享功能邏輯416內。在各個實施例中，圖形核心陣列414內的共享功能邏輯416可以包括共享功能邏輯420內的一些或全部邏輯。在一實施例中，共享功能邏輯420內的全部邏輯元件可在圖形核心陣列414的共享功能邏輯416內複製。在一實施例中，為了圖形核心陣列414內的共享功能邏輯416而未包含共享功能邏輯420。

執行單元

【0128】圖5A-5B顯示根據本文所述實施例包括圖形處理器核心中採用的處理元件陣列的執行緒執行邏輯

500。圖5A-5B中具有與本文中任何其它附圖的元件相同的元件符號(或名稱)的元件能夠根據與本文其它部分所述的方式相似的任何方式進行操作或者起作用，但不限於此。圖5A-5B顯示執行緒執行邏輯500的概述，其可以表示以圖2B的每個子核心221A-221F顯示的硬體邏輯。圖5A表示通用圖形處理器內的執行單元，而圖5B表示可以在計算加速器內使用的執行單元。

【0129】如圖5A所示，在一些實施例中，執行緒執行邏輯500包括著色器處理器502、執行緒分派器504、指令快取506、包括多個執行單元508A-508N的可縮放執行單元陣列、取樣器510、共享區域記憶體511、資料快取512和資料埠514。在一實施例中，可縮放執行單元陣列能夠透過基於工作負載的計算需求啟用或禁用一或多個執行單元(例如執行單元508A、508B、508C、508D至508N-1和508N中的任一個)來動態縮放。在一實施例中，所包含部件經由互連結構(其鏈接到部件的每個)來互連。在一些實施例中，執行緒執行邏輯500包括透過指令快取506、資料埠514、取樣器510和執行單元508A-508N中的一或多個到記憶體(諸如系統記憶體或快取記憶體)的一或多個連接。在一些實施例中，每個執行單元(例如508A)是獨立可程式化通用計算單元，其能夠運行多個同時硬體執行緒，同時對每個執行緒平行地處理多個資料元件。在各種實施例中，執行單元508A-508N的陣列可縮放成包括任何數量的單獨執行單元。

【0130】在一些實施例中，執行單元508A-508N主要用來運行著色器程式。著色器處理器502能夠處理各種著色器程式，並且經由執行緒分派器504來分派與著色器程式關聯的執行執行緒。在一實施例中，執行緒分派器包括用以仲裁來自圖形和媒體管線的執行緒發起請求並且在執行單元508A-508N中的一或多個執行單元上例示所請求執行緒的邏輯。例如，幾何管線能夠向執行緒執行邏輯分派頂點、鑲嵌或幾何著色器以供處理。在一些實施例中，執行緒分派器504還能夠處理來自運行著色器程式的運行時執行緒產生請求。

【0131】在一些實施例中，執行單元508A-508N支援指令集，其包括對許多標準3D圖形著色器指令的本機支援，使得來自圖形資料庫(例如Direct 3D和OpenGL)的著色器程式以最小轉化來運行。執行單元支援頂點和幾何處理(例如頂點程式、幾何程式、頂點著色器)、像素處理(例如像素著色器、片段著色器)和通用處理(例如計算和媒體著色器)。執行單元508A-508N的每個能夠進行多發佈單一指令多重資料(single instruction multiple data；SIMD)執行，並且多重執行緒操作在面對更高時間推遲記憶體存取中實現有效執行環境。每個執行單元中的每個硬體執行緒具有專用高頻寬暫存器檔案及關聯獨立執行緒狀態。執行是按時脈至管線的多重發出，能進行整數、單與雙精度浮點操作、SIMD分支能力、邏輯操作、超越操作(transcendental operation)、與其他雜項操作。在等待來自記憶體或共享

功能中的一個的資料時，執行單元508A-508N內的依賴邏輯使等待執行緒休眠，直到所請求的資料已返回。當等待執行緒正在休眠時，硬體資源可能會被專門用於處理其它執行緒。例如，在與頂點著色器操作相關聯的推遲期間，執行單元可以執行像素著色器、片段著色器或包括不同頂點著色器的另一類型的著色器程式的操作。各種實施例可以適用於透過使用單一指令多重執行緒(SIMT)來執行，作為對替代SIMD的使用或除了SIMD之外的使用。對SIMD核心或操作的參照也可以適用於SIMT或適用於SIMD與SIMT的結合。

【0132】執行單元508A-508N中的每個執行單元對資料元件的陣列進行操作。資料元件的數量是「執行大小」、或用於指令的通道數量。執行通道是用於資料元件存取、掩蔽和指令內的流程控制的執行的邏輯單元。通道的數量可以與針對特定圖形處理器的實體算術邏輯單元(ALU)或浮點單元(FPU)的數量無關。在一些實施例中，執行單元508A-508N支援整數和浮點資料類型。

【0133】執行單元指令集包括SIMD指令。各種資料元件可被儲存為緊縮資料類型於暫存器中且執行單元將基於元件之資料尺寸來處理各種元件。例如，當操作於256位元寬的向量時，256位元的向量被儲存於暫存器中且執行單元操作為四個獨立54位元之緊縮資料元件(Quad-Word (QW)尺寸資料元件)、八個獨立32位元之緊縮資料元件(Double Word (DW)尺寸資料元件)、十六個獨立16位元之

緊縮資料元件(Word (W)尺寸資料元件)、或三十二個獨立8位元資料元件(位元組(B)尺寸資料元件)之向量。然而，不同的向量寬度與暫存器尺寸是可能的。

【0134】在一實施例中，一或多個執行單元能夠結合為融合執行單元509A-509N，其具有融合EU共同的執行緒控制邏輯(507A-507N)。多個EU能夠融合為EU組。融合EU組中的每個EU能夠配置成運行獨立SIMD硬體執行緒。融合EU組中的EU的數量能夠根據實施例改變。另外，各種SIMD寬度能夠按EU執行，包括但不限於SIMD8、SIMD16和SIMD32。每個融合圖形執行單元509A-509N包括至少兩個執行單元。例如，融合執行單元509A包括第一EU 508A、第二EU 508B以及執行緒控制邏輯507A，其是第一EU 508A和第二EU 508B共同的。執行緒控制邏輯507A控制融合圖形執行單元509A上運行的執行緒，從而允許融合執行單元509A-509N內的每個EU使用共同指令指標暫存器運行。

【0135】一或多個內部指令快取(例如506)包含在執行緒執行邏輯500中，以快取執行單元的執行緒指令。在一些實施例中，包含一或多個資料快取(例如512)，以便在執行緒執行期間快取執行緒資料。在執行邏輯500上執行的執行緒還可以將明確地管理的資料儲存在共享區域記憶體511。在一些實施例中，包含取樣器510，以提供3D操作的紋理取樣和媒體操作的媒體取樣。在一些實施例中，取樣器510包括專用紋理或媒體取樣功能性，以便在向執

行單元提供取樣資料之前的取樣過程期間處理紋理或媒體資料。

【0136】在執行期間，圖形和媒體管線經由執行緒產生和分派邏輯向執行緒執行邏輯500發送執行緒發起請求。一旦一組幾何物件已經被處理並且光柵化為像素資料，調用著色器處理器502中的像素處理器邏輯(例如像素著色器邏輯、片段著色器邏輯等)，以進一步計算輸出資訊，並且使結果被寫到輸出表面(例如顏色緩衝器、深度緩衝器、模板(stencil)緩衝器等)。在一些實施例中，像素著色器或片段著色器計算各種頂點屬性的值，其跨光柵化物件來內插。在一些實施例中，著色器處理器502中的像素處理器邏輯則運行應用程式化介面(application programming interface；API)提供的像素或片段著色器程式。為了運行著色器程式，著色器處理器502經由執行緒分派器504向執行單元(例如508A)分派執行緒。在一些實施例中，著色器處理器502使用取樣器510中的紋理取樣邏輯來存取記憶體中儲存的紋理圖中的紋理資料。對紋理資料和輸入幾何資料的算術運算計算每個幾何片段的像素顏色資料，或者丟棄一或多個像素以免進一步處理。

【0137】在一些實施例中，資料埠514提供一種記憶體存取機制，以使執行緒執行邏輯500向記憶體輸出經處理資料以在圖形處理器輸出管線上進一步處理。在一些實施例中，資料埠514包括或者耦接到一或多個快取記憶體(例如資料快取512)，以經由資料埠來快取用於記憶體存

取的資料。

【0138】在一實施例中，執行邏輯500還可包含可以提供射線追蹤加速功能的射線追蹤器505。射線追蹤器505可以支援包含用於射線產生的指令/功能的射線追蹤指令集。射線追蹤指令集可以與圖2C中的射線追蹤核心245支援的射線追蹤指令集相似或不同。

【0139】圖5B顯示根據實施例的執行單元508的範例性內部細節。圖形執行單元508可以包括指令提取單元537、一般暫存器檔案陣列(GRF)524、架構暫存器檔案陣列(ARF)526、執行緒仲裁器522、發送單元530、分支單元532、SIMD浮點單元(FPU)534的集合，以及一個實施例中的專用整數SIMD ALU 535的集合。GRF 524和ARF 526包括與每個同時硬體執行緒(其在圖形執行單元508中可以是活動的)關聯的一般暫存器檔案和架構暫存器檔案的集合。在一實施例中，每執行緒架構狀態保持在ARF 526中，而執行緒執行期間所使用的資料儲存在GRF 524中。每個執行緒(包括每個執行緒的指令指標)的執行狀態能夠保存在ARF 526中的執行緒特定暫存器中。

【0140】在一實施例中，圖形執行單元508具有一種架構，其是同時多重執行緒(SMT)和細粒度交織多重執行緒(IMT)的組合。所述架構具有模組化配置，其能夠在設計時間基於同時執行緒的目標數量和每執行單元的暫存器的數量來微調，其中執行單元資源跨用來運行多個同時執行緒的邏輯來劃分。可以由圖形執行單元508執行的執行

緒邏輯的數量不限於硬體執行緒的數量，並且可以將多個邏輯執行緒分配給每個硬體執行緒。

【0141】一實施例中，圖形執行單元508能夠共同發出多個指令，其各自可以是不同指令。圖形執行單元執行緒508的執行緒仲裁器522能夠將指令分派給發送單元530、分支單元532或者(一或多個)SIMD FPU 534其中之一以供執行。每個執行執行緒能夠存取GRF 524內的128個通用暫存器，其中每個暫存器能夠儲存作為32位元資料元件的SIMD 8元件向量可存取的32個位元組。在一實施例中，每個執行單元執行緒有權存取GRF 524內的4 Kbyte，但是實施例並不局限於此，以及在其它實施例中可提供更多或更少暫存器資源。在一實施例中，圖形執行單元508被劃分為可以獨立執行計算操作的七個硬體執行緒，但是每執行單元的執行緒的數量也可根據實施例而變化。例如，在一實施例中，最多支援16個硬體執行緒。在七個執行緒可存取4 Kbyte的實施例中，GRF 524能夠儲存總共28 Kbyte。其中16個執行緒可以存取4 Kbyte，GRF 524可以總共儲存64 Kbyte。靈活的定址模式能夠允許暫存器被共同定址，以有效地構建更寬暫存器或者表示跨步矩形方塊(strided rectangular block)資料結構。

【0142】在一實施例中，記憶體操作、取樣器操作和其它較長時間推遲系統通訊經由「發送」指令(其由訊息傳遞發送單元530所運行)來分派。在一實施例中，分支指令被分派給專用分支單元532，以促進SIMD發散和最終收

斂。

【0143】在一實施例中，圖形執行單元508包括一或多個SIMD浮點單元(FPU)534，以執行浮點運算。在一實施例中，(一或多個)FPU 534還支援整數計算。在一實施例中，(一或多個)FPU 534能夠SIMD運行高達M個32位元浮點(或整數)運算，或者SIMD運行高達2M個16位元整數或16位元浮點運算。在一實施例中，(一或多個)FPU的至少一個提供擴展數學能力，以支援高產出量超越數學函數和雙精度54位元浮點。在一些實施例中，8位元整數SIMD ALU 535的集合也存在，並且可具體最佳化以執行與機器學習計算關聯的操作。

【0144】在一實施例中，圖形執行單元508的多個實例的陣列能夠在圖形子核心編組(例如子片)中例示。為了可縮放性，產品架構師能夠選擇每子核心編組的執行單元的準確數量。在一實施例中，執行單元508能夠跨多個執行通道來運行指令。在另一實施例中，圖形執行單元508上運行的每個執行緒在不同通道上運行。

【0145】**圖6**顯示根據實施例的額外執行單元600。執行單元600可以是用於例如如圖3C中所示的計算引擎塊340A-340D中的計算最佳化的執行單元，但不限於此。執行單元600的變體也可以用於如圖3B所示的圖形引擎塊310A-310D。在一實施例中，執行單元600包含執行緒控制單元601、執行緒狀態單元602、指令提取/預提取單元603和指令解碼單元604。執行單元600還包含暫存器檔案

606，其儲存可以分配給執行單元內的硬體執行緒的暫存器。執行單元600還包含發送單元607和分支單元608。在一實施例中，發送單元607和分支單元608可以與圖5B的圖形執行單元508的發送單元530和分支單元532類似地操作。

【0146】執行單元600還包含含有多個不同類型的功能單元的計算單元610。在一實施例中，計算單元610包含含有算術邏輯單元的陣列的ALU單元611。ALU單元611可以被配置成執行64位元、32位元和16位元整數和浮點運算。整數和浮點運算可以同時執行。計算單元610還可以包含脈動陣列612和數學單元613。脈動陣列612包含可用於以脈動方式執行向量或其它資料平行操作的資料處理單元的*W*寬和*D*深度網路。在一實施例中，脈動陣列612可以被配置成執行矩陣運算，諸如矩陣點積運算。在一實施例中，脈動陣列612支援16位元浮點運算，以及8位元和4位元整數運算。在一實施例中，脈動陣列612可以被配置成加速機器學習操作。在這種實施例中，脈動陣列612可以配置成支援bfloat 16位元浮點格式。在一實施例中，可以包含數學單元613，以比ALU單元611有效且低功率消耗的方式執行數學運算的特定子集。數學單元613可以包含數學邏輯的變體，其可以在由其它實施例提供的圖形處理引擎的共享功能邏輯中找到(例如，圖4的共享功能邏輯420的數學邏輯422)。在一實施例中，數學單元613可以被配置成執行32位元和64位元浮點運算。

【0147】執行緒控制單元601包含用於控制執行單元內的執行緒的執行的邏輯。執行緒控制單元601可以包含用以開始、停止和搶先執行執行單元600內的執行緒的執行緒仲裁邏輯。執行緒狀態單元602可以用於儲存分配在執行單元600上執行的執行緒的執行緒狀態。當執行緒被阻塞或閒置時，將執行緒狀態儲存在執行單元600內，使得執行緒的快速搶占。指令提取/預提取單元603可以從更高階的執行邏輯的指令快取(例如，如圖5A中的指令快取506)提取指令。指令提取/預提取單元603還可以基於對當前執行的執行緒的分析，發出對要載入到指令快取中的指令的預提取請求。指令解碼單元604可以用於解碼將由計算單元執行的指令。在一實施例中，指令解碼單元604可以用作輔助解碼器，以將複雜的指令解碼為組成的微運算。

【0148】執行單元600還包含暫存器檔案606，其可以被在執行單元600上執行的硬體執行緒使用。在暫存器檔案606中的暫存器可以在用於執行執行單元600的計算單元610內的多個同時執行緒的邏輯上劃分。可以由圖形執行單元600執行的邏輯執行緒數目不限於硬體執行緒的數目，並且可以將多個邏輯執行緒分配給每個硬體執行緒。暫存器檔案606的大小可以基於支援的硬體執行緒的數目而在各實施例中變化。在一實施例中，暫存器重命名可用於動態地將暫存器分配給硬體執行緒。

【0149】圖7是顯示根據一些實施例的圖形處理器指

令格式700的方塊圖。在一或多個實施例中，圖形處理器執行單元支援具有多種格式的指令的指令集。實線框顯示一般包含在執行單元指令中的部件，而虛線包括可選的或者僅包含在指令的子集中的部件。在一些實施例中，所述和所示的指令格式700是巨集指令，因為其為提供給執行單元的指令，與一旦處理指令則產生於指令解碼的微操作相反。

【0150】在一些實施例中，圖形處理器執行單元本機地支援根據128位元指令格式710的指令。基於所選指令、指令選項和運算元的數量，64位元壓縮指令格式730可用在一些指令。本機128位元指令格式710提供對全部指令選項的存取，而一些選項和操作限制在64位元格式730。根據64位元格式730可用的本機指令根據實施例改變。在一些實施例中，部分地使用索引欄位713中的索引值集合來壓縮指令。執行單元硬體基於索引值來參照一組壓縮表，並使用壓縮表輸出來重構採用128位元指令格式710的本機指令。可以使用其它大小和格式的指令。

【0151】對於每個格式，指令運算碼712定義執行單元要執行的操作。執行單元跨每個運算元的多個資料元件平行地運行每個指令。例如，回應加法指令，執行單元跨表示紋理元件或圖片元件的每個顏色通道來執行同時加法運算。預設地，執行單元跨運算元的所有資料通道來執行每個指令。在一些實施例中，指令控制欄位714實現對於諸如通道選擇(例如斷言)和資料通道順序(例如混合)之類

的某些執行選項的控制。對於採用128位元指令格式710的指令，執行大小欄位716限制將平行運行的資料通道的數量。在一些實施例中，執行大小欄位716不可用於64位元壓縮指令格式730。

【0152】一些執行單元指令具有總共三個運算元，其包括兩個來源運算元src0 720和src1 722以及一個目的地718。在一些實施例中，執行單元支援雙目的地指令，其中暗示目的地之一。資料操縱指令能夠具有第三來源運算元(例如SRC2 724)，其中指令運算碼712確定來源運算元的數量。指令的最後一個來源運算元可以是隨指令而傳遞的立即(例如硬編碼)值。

【0153】在一些實施例中，128位元指令格式710包括存取/位址模式欄位726，其指定例如是使用直接暫存器定址模式還是間接暫存器定址模式。當使用直接暫存器定址模式時，一或多個運算元的暫存器位址透過指令中的位元直接提供。

【0154】在一些實施例中，128位元指令格式710包括存取/位址模式欄位726，其指定指令的位址模式及/或存取模式。在一實施例中，存取模式用來定義指令的資料存取對齊。一些實施例支援包括16位元組對齊存取模式和1位元組對齊存取模式的存取模式，其中存取模式的位元組對齊確定指令運算元的存取對齊。例如，當處於第一模式時，指令可將位元組對齊定址用於來源和目的地運算元，而當處於第二模式時，指令可將16位元組對齊定址用於所

有來源和目的地運算元。

【0155】在一實施例中，存取/位址模式欄位726的位址模式部分確定指令是使用直接還是間接定址。當使用直接暫存器定址模式時，指令中的位元直接提供一或多個運算元的暫存器位址。當使用間接暫存器定址模式時，一或多個運算元的暫存器位址可基於指令中的位址暫存器值和位址立即欄位來計算。

【0156】在一些實施例中，指令基於運算碼712位元欄位來編組，以簡化運算碼解碼740。對於8位元運算碼，位元4、5和6允許執行單元確定運算碼的類型。所示的準確運算碼編組只是範例。在一些實施例中，移動和邏輯運算碼編組742包括資料移動和邏輯指令(例如移動(mov)、比較(cmp))。在一些實施例中，移動和邏輯編組742共享五個最高有效位元(MSB)，其中移動(mov)指令採取0000xxxxb的形式，以及邏輯指令採取0001xxxxb的形式。流程控制指令編組744(例如調用、跳轉(jmp))包括採取0010xxxxb(例如0x20)的形式的指令。混雜指令編組746包括指令的混合，其包括採取0011xxxxb(例如0x30)的形式的同步指令(例如等待、發送)。平行數學指令編組748包括採取0100xxxxb(例如0x40)的形式的逐個部件的算術指令(例如加法、乘法(mul))。平行數學編組748跨資料通道平行地執行算術運算。向量數學編組750包括採取0101xxxxb(例如0x50)的形式的算術指令(例如dp4)。向量數學編組對向量運算元執行算術、例如點積計算。在一實

施例中，顯示的運算碼解碼740可以用於確定執行單元的哪一部分將用於執行解碼的指令。例如，一些指令可以被指定為將由脈動陣列執行的脈動指令。諸如射線追蹤指令(未顯示)的其它指令可以被路由到執行邏輯的切片或分區內的射線追蹤核心或射線追蹤邏輯。

圖形管線

【0157】圖8是圖形處理器800的另一個實施例的方塊圖。圖8中具有與本文中任何其它附圖的元件相同的元件符號(或名稱)的元件能夠根據與本文其它部分所述相似方式的任何方式進行操作或者起作用，但不限於此。

【0158】在一些實施例中，圖形處理器800包括幾何管線820、媒體管線830、顯示引擎840、執行緒執行邏輯850和繪製輸出管線870。在一些實施例中，圖形處理器800是多核心處理系統中的圖形處理器，其包括一或多個通用處理核心。圖形處理器透過對一或多個控制暫存器(未顯示)的暫存器寫入或者透過經由環狀互連802向圖形處理器800發出的命令來控制。在一些實施例中，環狀互連802將圖形處理器800耦接到其它處理部件(例如其它圖形處理器或通用處理器)。來自環狀互連802的命令由命令串流器803來解釋，命令串流器803將指令提供給幾何管線820或媒體管線830的單獨部件。

【0159】在一些實施例中，命令串流器803指導頂點提取器805的操作，頂點提取器805從記憶體中讀取頂點資

料，並且運行命令串流器803所提供的頂點處理命令。在一些實施例中，頂點提取器805將頂點資料提供給頂點著色器807，其對每個頂點執行座標空間變換和照明操作。在一些實施例中，頂點提取器805和頂點著色器807透過經由執行緒分派器831向執行單元852A-852B分派執行執行緒來運行頂點處理指令。

【0160】在一些實施例中，執行單元852A-852B是向量處理器陣列，其具有用於執行圖形和媒體操作的指令集。在一些實施例中，執行單元852A-852B具有附接L1快取851，其對每個陣列是特定的或者在陣列之間共享。快取能夠配置為資料快取、指令快取或者單一快取，其劃分成在不同分區中包含資料和指令。

【0161】在一些實施例中，幾何管線820包括鑲嵌部件，以執行3D物件的硬體加速鑲嵌。在一些實施例中，可程式化外殼著色器811配置鑲嵌操作。可程式化域著色器817提供鑲嵌輸出的後端評估。鑲嵌器813在外殼著色器811的方向進行操作，並且包含專用邏輯，以便基於粗略幾何模型(其作為輸入來提供給幾何管線820)來產生詳細幾何物件集合。在一些實施例中，如果不使用鑲嵌，則能夠繞過鑲嵌部件(例如外殼著色器811、鑲嵌器813和域著色器817)。

【0162】在一些實施例中，完整幾何物件能夠由幾何著色器819經由分派給執行單元852A-852B的一或多個執行緒來處理，或者能夠直接進入剪輯器829。在一些實施例

中，幾何著色器對整個幾何物件而不是如圖形管線的先前級中一樣的頂點或頂點貼片進行操作。如果禁用鑲嵌，則幾何著色器819接收來自頂點著色器807的輸入。在一些實施例中，幾何著色器819是幾何著色器程式可程式化的，以便在禁用鑲嵌單元時執行幾何鑲嵌。

【0163】在光柵化之前，剪輯器829能夠處理頂點資料。剪輯器829可以是固定功能剪輯器或者具有剪輯和幾何著色器功能的可程式化剪輯器。在一些實施例中，繪製輸出管線870中的光柵器和深度測試部件873分派像素著色器，以便將幾何物件轉換為每像素表示。在一些實施例中，像素著色器邏輯包含在執行緒執行邏輯850中。在一些實施例中，應用能夠繞過光柵器和深度測試部件873，並且經由流輸出單元823來存取未光柵化頂點資料。

【0164】圖形處理器800具有互連匯流排、互連結構或者另外某種互連機制，其允許資料和訊息在處理器的主要部件之間傳遞。在一些實施例中，執行單元852A-852B及關聯邏輯單元(例如L1快取851、取樣器854、紋理快取858等)經由資料埠856來互連，以執行記憶體存取，並且與處理器的繪製輸出管線部件進行通訊。在一些實施例中，取樣器854、快取851和858以及執行單元852A-852B各自具有獨立記憶體存取通路。在一實施例中，紋理快取858還能夠配置為取樣器快取。

【0165】在一些實施例中，繪製輸出管線870包含光柵器和深度測試部件873，其將基於頂點的物件轉換為關

聯的基於像素的表示。在一些實施例中，光柵器邏輯包括加窗器/掩蔽器單元，以執行固定功能三角和線條光柵化。關聯繪製快取878和深度快取879在一些實施例中也是可用的。像素操作部件877對資料執行基於像素的操作，但是在一些實例中，與2D操作關聯的像素操作(例如採用混合的位塊影像傳輸)由2D引擎841來執行，或者由顯示控制器843使用覆蓋顯示平面在顯示時間來替代。在一些實施例中，共享L3快取875是全部圖形部件可用的，從而允許資料的共享，而無需使用主系統記憶體。

【0166】在一些實施例中，圖形處理器媒體管線830包括媒體引擎837和視訊前端834。在一些實施例中，視訊前端834從命令串流器803接收管線命令。在一些實施例中，媒體管線830包括獨立命令串流器。在一些實施例中，視訊前端834在將命令發送給媒體引擎837之前處理媒體命令。在一些實施例中，媒體引擎837包括執行緒產生功能性，以產生執行緒以供經由執行緒分派器831分派給執行緒執行邏輯850。

【0167】在一些實施例中，圖形處理器800包括顯示引擎840。在一些實施例中，顯示引擎840是處理器800外部的，並且經由環狀互連802或者另外某種互連匯流排或結構與圖形處理器耦接。在一些實施例中，顯示引擎840包括2D引擎841和顯示控制器843。在一些實施例中，顯示引擎840包含專用邏輯，其能夠與3D管線無關地操作。在一些實施例中，顯示控制器843與顯示裝置(未顯示)耦

接，顯示裝置可以是如同膝上型電腦中一樣的系統整合顯示裝置或者是經由顯示裝置連接器所附接的外部顯示裝置。

【0168】在一些實施例中，幾何管線820和媒體管線830可配置成基於多個圖形和媒體程式化介面來執行操作，而不是特定於任何一個應用程式化介面(API)。在一些實施例中，用於圖形處理器的驅動軟體將特定於特定圖形或媒體庫的API調用轉換成可以由圖形處理器處理的命令。在一些實施例中，為都來自Khronos Group的開放圖形庫(OpenGL)、開放計算語言(OpenCL)及/或Vulkan圖形和計算API提供支援。在一些實施例中，也可以為來自微軟公司的Direct3D庫提供支援。在一些實施例中，可以支援這些庫的組合。還可以為開源電腦視覺庫(OpenCV)提供支援。如果可以作出從未來的API的管線到圖形處理器的管線的映射，則具有兼容的3D管線的未來的API也將受到支援。

圖形管執行緒式化

【0169】圖9A是顯示根據一些實施例的圖形處理器命令格式900的方塊圖。圖9B是顯示根據實施例的圖形處理器命令序列910的方塊圖。圖9A中的實線框顯示一般包含在圖形命令中的部件，而虛線包括可選的或者僅包含在圖形命令的子集中的部件。圖9A的範例性圖形處理器命令格式900包括識別客戶端902的資料欄位、命令操作碼(運算

碼)904以及命令的資料906。子運算碼905和命令大小908也包含在一些命令中。

【0170】在一些實施例中，客戶端902指定處理命令資料的圖形裝置的客戶端單元。在一些實施例中，圖形處理器命令解析器檢查每個命令的客戶端欄位，以便調節命令的進一步處理，並且將命令資料路由到適當客戶端單元。在一些實施例中，圖形處理器客戶端單元包括記憶體介面單元、繪製單元、2D單元、3D單元和媒體單元。每個客戶端單元具有處理命令的對應處理管線。一旦命令由客戶端單元接收，則客戶端單元讀取運算碼904以及子運算碼905(若存在的話)，以確定要執行的操作。客戶端單元使用資料欄位906中的資訊來執行命令。對在一些命令，預計明確地命令大小908指定命令的大小。在一些實施例中，命令解析器基於命令運算碼自動確定命令的至少一部分的大小。在一些實施例中，命令經由雙字的倍數來對齊，可以使用其它命令格式。

【0171】圖9B的流程圖顯示範例性圖形處理器命令序列910。在一些實施例中，以圖形處理器的實施例為特色的資料處理系統的軟體或韌體使用所顯示的命令序列的版本來建立、執行和終止圖形操作集合。僅出於範例的目的顯示並描述了樣本命令序列，因為實施例不限於這些特定命令或者此命令序列。此外，所述命令可以被作為命令序列中的一批次命令來發佈，使得圖形處理器將至少部分同時地處理命令的序列。

【0172】在一些實施例中，圖形處理器命令序列910可開始於管線清除命令912，以便使任何活動圖形管線完成管線的當前未決命令。在一些實施例中，3D管線922和媒體管線924沒有平行地操作。執行管線清除，以便使活動圖形管線完成任何未決命令。回應管線清除，圖形處理器的命令解析器將停止命令處理，直到活動繪圖引擎完成未決操作並且使相關讀快取失效。可選地，繪製快取中標籤為'髒'的任何資料能夠清除到記憶體。在一些實施例中，管線清除命令912能夠用於管線同步或者在將圖形處理器投入低功率狀態之前使用。

【0173】在一些實施例中，當命令序列要求圖形處理器在管線之間明確地切換時，使用管線選擇命令913。在一些實施例中，在發出管線命令之前的執行上下文內僅要求一次管線選擇命令913，除非上下文是發出兩種管線的命令。在一些實施例中，就在管線經由管線選擇命令913進行切換之前，要求管線清除命令912。

【0174】在一些實施例中，管線控制命令914配置用於操作的圖形管線，並且用來對3D管線922和媒體管線924進行程式化。在一些實施例中，管線控制命令914配置活動管線的管線狀態。在一實施例中，管線控制命令914用於管線同步，並且用來在處理一批次命令之前從活動管線內的一或多個快取記憶體中清除資料。

【0175】在一些實施例中，返回緩衝器狀態命令916用來配置使相應管線寫入資料的返回緩衝器集合。一些管

線操作要求一或多個返回緩衝器(操作在處理期間將中間資料寫入其中)的分配、選擇或配置。在一些實施例中，圖形處理器還使用一或多個返回緩衝器來儲存輸出資料，並且執行相交執行緒通訊。在一些實施例中，返回緩衝器狀態916包括選擇用於管線操作集合的返回緩衝器的大小和數量。

【0176】命令序列中的其餘命令基於用於操作的活動管線而有所不同。基於管線確定920，命令序列適合開始於3D管線狀態930的3D管線922或者開始於媒體管線狀態940的媒體管線924。

【0177】配置3D管線狀態930的命令包括用於頂點緩衝器狀態、頂點元件狀態、恆定顏色狀態、深度緩衝器狀態以及在處理3D基元命令之前所配置的其它狀態變量的3D狀態設定命令。這些命令的值至少部分基於使用中的特定3D API來確定。在一些實施例中，3D管線狀態930命令還能夠在將不使用某些管線元件時有選擇地禁用或繞過那些元件。

【0178】在一些實施例中，3D基元932命令用來提交將要由3D管線所處理的3D基元。經由3D基元932命令來傳遞給圖形處理器的命令及關聯參數被轉發給圖形管線中的頂點獲取功能。頂點獲取功能使用3D基元932命令資料來產生頂點資料結構。頂點資料結構儲存在一或多個返回緩衝器中。在一些實施例中，3D基元932命令用來經由頂點著色器對3D基元執行頂點操作。為了處理頂點著色器，

3D管線922向圖形處理器執行單元分派著色器執行執行緒。

【0179】在一些實施例中，3D管線922經由運行934命令或事件來觸發。在一些實施例中，暫存器寫入觸發命令執行。在一些實施例中，執行經由命令序列中的「go」或「kick」命令來觸發。在一實施例中，使用管線同步命令來觸發命令執行，以便經過圖形管線來清除命令序列。3D管線將執行3D基元的幾何處理。一旦操作完成，所產生幾何物件經過光柵化，並且像素引擎對所產生像素著色。還可為那些操作包含控制像素著色和像素後端操作的附加命令。

【0180】在一些實施例中，圖形處理器命令序列910在執行媒體操作時沿用媒體管線924通路。一般來說，媒體管線924的程式化的特定使用和方式取決於待執行的媒體或計算操作。特定媒體解碼操作可在媒體解碼期間卸載到媒體管線。在一些實施例中，還能夠繞過媒體管線，並且媒體解碼能夠完全或部分使用一或多個通用處理核心所提供的資源來執行。在一實施例中，媒體管線還包括用於通用圖形處理器單元(GPGPU)操作的元件，其中圖形處理器用來使用計算著色器程式(其不是與圖形基元的繪製明確地相關的)來執行SIMD向量操作。

【0181】在一些實施例中，媒體管線924根據與3D管線922相似的方式來配置。配置媒體管線狀態940的命令集合在媒體物件命令942之前被分派或放入命令佇列中。在

一些實施例中，媒體管線狀態940的命令包括配置媒體管線元件(其將用來處理媒體物件)的資料。這包括配置媒體管線中的視訊解碼和視訊編碼邏輯的資料，例如編碼或解碼格式。在一些實施例中，媒體管線狀態940的命令還支援使用到包含一批狀態設定的“間接”狀態元件的一或多個指標。

【0182】在一些實施例中，媒體物件命令942提供到媒體物件的指標以供媒體管線處理。媒體物件包括記憶體緩衝器，其包含待處理視訊資料。在一些實施例中，所有媒體管線狀態在發出媒體物件命令942之前必須是有效的。一旦配置管線狀態並且將媒體物件命令942進入佇列，則媒體管線924經由運行命令944或者等效運行事件(例如暫存器寫入)來觸發。來自媒體管線924的輸出則可透過3D管線922或媒體管線924所提供的操作進行後處理。在一些實施例中，GPGPU操作根據與媒體操作相似的方式來配置和運行。

圖形軟體架構

【0183】圖10顯示根據一些實施例，用於資料處理系統1000的範例性圖形軟體架構。在一些實施例中，軟體架構包括3D圖形應用程式1010、作業系統1020和至少一個處理器1030。在一些實施例中，處理器1030包括圖形處理器1032和一或多個通用處理器核心1034。圖形應用程式1010和作業系統1020各自在資料處理系統的系統記憶體1050中

運行。

【0184】在一些實施例中，3D圖形應用程式1010包含一或多個著色器程式，其包括著色器指令1012。著色器語言指令可採用高階著色器語言，例如Direct3D的高階著色器語言(HLSL)、OpenGL著色器語言(GLSL)等等。應用還包括根據適合於通用處理器核心1034執行的機器語言的可執行指令1014。應用還包括透過頂點資料所定義的圖形物件1016。

【0185】在一些實施例中，作業系統1020是來自微軟公司的Microsoft® Windows®作業系統、專有類UNIX作業系統或者使用Linux核心的變體的開源類UNIX作業系統。作業系統1020能夠支援圖形API 1022，例如Direct3D API、OpenGL API或者Vulkan API。當Direct3DAPI被使用時，作業系統1020使用前端著色器編譯器1024來將HLSL的任何著色器指令1012編譯為低階著色器語言。編譯可以是適時(JIT)編譯，或者應用能夠執行著色器預編譯。在一些實施例中，在3D圖形應用程式1010的編譯期間將高階著色器編譯為低階著色器。在一些實施例中，著色器指令1012採用中間形式(例如Vulkan API所使用的標準可攜式中間表示(SPIR)的一個版本)來提供。

【0186】在一些實施例中，使用者模式圖形驅動器1026包含後端著色器編譯器1027，以便將著色器指令1012轉換為硬體特定表示。當OpenGL API被使用時，GLSL高階語言中的著色器指令1012被傳遞給使用者模式圖形驅動

器1026以供編譯。在一些實施例中，使用者模式圖形驅動器1026使用作業系統核心模式功能1028與核心模式圖形驅動器1029進行通訊。在一些實施例中，核心模式圖形驅動器1029與圖形處理器1032進行通訊，以分派命令和指令。

IP核心實現

【0187】至少一個實施例的一或多個態樣可透過機器可讀取媒體上儲存的代表程式碼來實現，其表示及/或定義積體電路(例如處理器)中的邏輯。例如，機器可讀取媒體可包括表示處理器中的各種邏輯的指令。在由機器讀取時，指令可使所述機器製作執行本文所述技術的邏輯。這類表示(稱作「IP核心」)是積體電路的邏輯的可再用單元，其可作為描述積體電路的結構的硬體模型來儲存在有形機器可讀取媒體上。硬體模型可提供給各種客戶或製造機構，其將硬體模型載入到製造積體電路的製作機器上。可製作積體電路，使得電路執行與本文所述實施例的任一個關聯描述的操作。

【0188】圖11A是顯示根據實施例，可用來製造執行操作的積體電路的IP核心開發系統1100的方塊圖。IP核心開發系統1100可用來產生模組化可再用設計，其能夠結合到較大設計中或者用來構成整個積體電路(例如SOC積體電路)。設計機構1130能夠採用高階程式化語言(例如C/C++)來產生IP核心設計的軟體模擬1110。軟體模擬1110能夠用來使用模擬模型1112來設計、測試和檢驗IP核心的

行為。模擬模型1112可包括功能、行為及/或時序模擬。暫存器傳輸級(RTL)設計1115接著能夠從模擬模型1112來建立或合成。RTL設計1115是積體電路的行為的抽象，其對硬體暫存器之間的數位訊號的流程進行模組化，其中包括使用模組化數位訊號所執行的關聯邏輯。除了RTL設計1115之外，還可建立、設計或合成在邏輯級或電晶體級的低階設計。因此，初始設計和模擬的具體細節可改變。

【0189】RTL設計1115或者等效體可由設計機構進一步合成到硬體模型1120(其可透過硬體描述語言(HDL)或者實體設計資料的另外某種表示中)。還可模擬或測試HDL，以檢驗IP核心設計。能夠儲存IP核心設計，以供使用非揮發性記憶體1140(例如硬碟、快閃記憶體或者任何非揮發性儲存媒體)傳遞給第三方製作機構1165。備選地，IP核心設計可透過有線連接1150或無線連接1160來傳送(例如經由網際網路)。製作機構1165接著製作至少部分基於IP核心設計的積體電路。所製作的積體電路能夠配置成執行根據本文所述至少一個實施例的操作。

【0190】圖11B顯示根據本文所述的一些實施例的積體電路封裝組件1170的截面側視圖。積體電路封裝組件1170顯示如本文所述的一或多個處理器或加速器裝置的實現。封裝組件1170包括連接到基板1180的硬體邏輯1172、1174的多個單元。邏輯1172、1174可至少部分採用可配置邏輯或固定功能性邏輯硬體來實現，並且可以包括(一或多個)處理器核心、(一或多個)圖形處理器或者本文所述的

其它加速器裝置中的任一者的一或多個部分。邏輯1172、1174的每個單元能夠在半導體晶粒內實現，並且經由互連結構1173與基板1180相耦接。互連結構1173可配置成在邏輯1172、1174與基板1180之間路由電子訊號，並且可以包括諸如但不限於凸塊或柱之類的互連。在一些實施例中，互連結構1173可配置成路由與邏輯1172、1174的操作關聯的電子訊號，諸如例如輸入/輸出(I/O)訊號及/或功率或接地訊號。在一些實施例中，基板1180是基於環氧樹脂的層壓基板。在其它實施例中，基板1180可包括其它適當類型的基板。封裝組件1170能夠經由封裝互連1183來連接到其它電子裝置。封裝互連1183可耦接到基板1180的表面，以便將電子訊號路由到其它電子裝置，例如主機板、另一晶片組或者多晶片模組。

【0191】在一些實施例中，邏輯1172、1174的單元與橋接器1182電耦接，橋接器1182配置成在邏輯1172、1174之間路由電子訊號。橋接器1182可以是密集互連結構，其提供電子訊號的路由。橋接器1182可包括由玻璃或適當半導體材料所組成的橋接器基板。電子佈線特徵能夠在橋接器基板上形成，以提供邏輯1172、1174之間的晶片對晶片連接。

【0192】雖然顯示邏輯1172、1174的兩個單元和橋接器1182，但是本文所述的實施例可包括一或多個晶粒上的更多或更少邏輯單元。一或多個晶粒可透過零或多個橋接器來連接，因為當邏輯包含在單一晶粒上時，可以不包含

橋接器1182。備選地，邏輯的多個晶粒或單元能夠透過一或多個橋接器來連接。另外，多個邏輯單元、晶粒和橋接器能夠根據其它可能配置(包括三維配置)共同連接。

【0193】**圖11C**顯示包含與基板1180連接的硬體邏輯小晶片的多個單元 (例如，基礎晶粒)的封裝組件1190。如本文所述的圖形處理單元、平行處理器及/或計算加速器可以由分別製造的各種矽小晶片組成。在此上下文中，小晶片是至少部分封裝的積體電路，其包含可以與其它小晶片組裝成較大封裝的邏輯的不同單元。可以將具有不同IP核心邏輯的各種小晶片組裝到單一裝置中。此外，可以使用主動中介層技術將小晶片整合到基礎晶粒或基礎小晶片中。本文描述的概念使得能夠在GPU內的不同形式IP之間進行互連和通訊。IP核心可以使用不同的製程技術進行製造，並在製造程序中進行組合，從而避免了將多個IP(尤其是在具有多種IP的大型SoC上)整合到同一製造程序的複雜性。致使多種製程技術的使用可以縮短上市時間，並提供一種經濟高效的方式來建立多個產品SKU。此外，分解後的IP更適合獨立控制電源，在給定工作負載下未使用的部件可以關閉電源、從而降低了總體功率消耗。

【0194】硬體邏輯小晶片可以包含專用硬體邏輯小晶片1172、邏輯或I/O小晶片1174及/或記憶體小晶片1175。硬體邏輯小晶片1172和邏輯或I/O小晶片1174可以至少部分地以可配置邏輯或固定功能邏輯硬體來實現，並且可以包含本文所述的處理器核心、圖形處理器、平行處理器或

其它加速器裝置中的任何一個的一或多個部分。記憶體小晶片1175可以是DRAM(例如，GDDR、HBM)記憶體或快取(SRAM)記憶體。

【0195】每個小晶片可以被製造為單獨的半導體晶粒並透過互連結構1173與基板1180耦接。互連結構1173可以被配置成在各種小晶片和基板1180內的邏輯之間路由電子訊號。互連結構1173可以包含互連，諸如但不限於凸塊或柱。在某些實施例中，互連結構1173可以被配置成路由電子訊號，諸如例如與邏輯、I/O和記憶體小晶片的操作相關的輸入/輸出(I/O)訊號及/或電源或接地訊號。

【0196】在一些實施例中，基板1180是基於環氧化物的層壓基板。在其它實施例中，基板1180可以包含其它合適類型的基板。封裝組件1190可以透過封裝互連1183連接到其它電子裝置。封裝互連1183可以耦接到基板1180的表面，以將電子訊號路由到其它電子裝置，諸如主機板、其它晶片組或多晶片模組。

【0197】在一些實施例中，邏輯或I/O小晶片1174和記憶體小晶片1175可以透過橋接器1187電耦接，橋接器1187配置成在邏輯或I/O小晶片1174和記憶體小晶片1175之間路由電子訊號。橋接器1187可以是提供電子訊號的路由的密集互連結構。橋接器1187可以包含由玻璃或合適的半導體材料組成的橋接器基板。可以在橋接器基板上形成電路由特徵，以提供邏輯或I/O小晶片1174和記憶體小晶片1175之間的晶片對晶片連接。橋接器1187也可以稱為矽

橋接器或互連橋接器。例如，在一些實施例中，橋接器1187是嵌入式多晶粒互連橋接器(EMIB)。在某些實施例中，橋接器1187可以簡單地是從一個小晶片到另一小晶片的直接連接。

【0198】基板1180可以包含用於I/O 1191、快取記憶體1192和其它硬體邏輯1193的硬體部件。結構1185可以嵌入到基板1180以使基板1180內的各種邏輯小晶片與邏輯1191、1193之間進行通訊。在一實施例中，可以將I/O 1191、結構1185、快取、橋接器和其它硬體邏輯1193整合到在基板1180的頂部上分層的基礎晶粒。

【0199】在各個實施例中，封裝組件1190可包含更少或更多數量的由結構1185或一或多個橋接器1187互連的部件和小晶片。封裝組件1190內的小晶片可以被3D或2.5D佈置。一般來說，橋接器結構1187可用於促進例如邏輯或I/O小晶片與記憶體小晶片之間的點對點互連。結構1185可用於將各種邏輯及/或I/O小晶片(例如，小晶片1172、1174、1191、1193)與其它邏輯及/或I/O小晶片互連。在一實施例中，基板內的快取記憶體1192可以用作用於封裝組件1190的全域快取、分佈式全域快取的一部分、或用作用於結構1185的專用快取。

【0200】**圖11D**顯示根據一個實施例包含可互換小晶片1195的封裝組件1194。可以將可互換小晶片1195組裝成一或多個基礎小晶片1196、1198的標準化插槽。基礎小晶片1196、1198可以經由橋接器互連1197被耦接，其可以類

似於本文所述的其它橋接器互連，並且可以是例如EMIB。記憶體小晶片也可以透過橋接器互連連接到邏輯或I/O小晶片。I/O和邏輯小晶片可以透過互連結構進行通訊。基礎小晶片可各自以用於邏輯或I/O或記憶體/快取之一的標準化格式支援一或多個插槽。

【0201】在一實施例中，SRAM和功率傳輸電路可以被製造為基礎小晶片1196、1198中的一或多個，其可以使用與堆疊在基礎小晶片的頂部上的可互換小晶片1195相關的不同製程技術來製造。例如，基礎小晶片1196、1198可以使用較大的製程技術來製造，而可互換小晶片可以使用較小的製程技術來製造。可互換小晶片1195中的一或多個可以是記憶體(例如，DRAM)小晶片。對於封裝組件1194可以基於針對使用封裝組件1194的產品的功率及/或效能來選擇不同的記憶體密度。此外，可以在組裝時根據產品的功率及/或效能來選擇具有不同數量類型的功能單元的邏輯小晶片。此外，可以將包含不同類型IP邏輯核心的小晶片插入可互換小晶片的插槽中，從而實現可以混合和匹配不同技術IP區塊的混合處理器設計。

晶片積體電路上的範例性系統

【0202】圖12-14顯示根據本文所述的各種實施例，可使用一或多個IP核心來製作的範例性積體電路及關聯圖形處理器。除了所示的之外，還可包含其它邏輯和電路，其中包括附加圖形處理器/核心、周邊介面控制器或者通

用處理器核心。

【0203】圖12是顯示根據實施例，可使用一或多個IP核心來製作的範例性系統單晶片積體電路1200的方塊圖。範例性積體電路1200包括一或多個應用處理器1205(例如CPU)、至少一個圖形處理器1210，並且還可包括影像處理器1215及/或視訊處理器1220，其任一個可以是來自相同或者多個不同設計機構的模組化IP核心。積體電路1200包括周邊或匯流排邏輯，包括USB控制器1225、UART控制器1230、SPI/SDIO控制器1235和I^2S/I^2C控制器1240。另外，積體電路可以包括耦接到高解析度多媒體介面(HDMI)控制器1250與行動產業處理器介面(mobile industry processor interface；MIPI)顯示介面1255中之一或多者的顯示裝置1245。儲存裝置可由快閃記憶體子系統1260(包括快閃記憶體和快閃記憶體控制器)來提供。記憶體介面可經由記憶體控制器1265來提供，以供存取SDRAM或SRAM記憶體裝置。一些積體電路還包括嵌入式安全引擎1270。

【0204】圖13-14是顯示根據本文所述實施例，供SoC內使用的範例性圖形處理器的方塊圖。圖13顯示根據實施例，可使用一或多個IP核心來製作的系統單晶片積體電路的範例性圖形處理器1310。圖13顯示根據實施例，可使用一或多個IP核心來製作的系統單晶片積體電路的附加範例性圖形處理器1340。圖13A的圖形處理器1310是低功率圖形處理器核心的範例。圖13B的圖形處理器1340是較高效

能圖形處理器核心的範例。圖形處理器1310、1340的每個可以是圖12的圖形處理器1210的變體。

【0205】如圖13所示，圖形處理器1310包括頂點處理器1305和一或多個片段處理器1315A-1315N(例如1315A、1315B、1315C、1315D至1315N-1和1315N)。圖形處理器1310能夠經由獨立邏輯來運行不同著色器程式，使得頂點處理器1305被最佳化以運行頂點著色器程式的操作，而一或多個片段處理器1315A-1315N運行片段或像素著色器程式的片段(例如像素)著色操作。頂點處理器1305執行3D圖形管線的頂點處理級，並且產生基元和頂點資料。(一或多個)片段處理器1315A-1315N使用頂點處理器1305所產生的基元和頂點資料來產生訊框緩衝器，其在顯示裝置上顯示。在一實施例中，(一或多個)片段處理器1315A-1315N被最佳化以運行如在OpenGL API中所提供的片段著色器程式，其可用來執行與如在Direct 3D API所提供的像素著色器程式相似的操作。

【0206】圖形處理器1310還包括一或多個記憶體管理單元(MMU)1320A-1320B、(一或多個)快取1325A-1325B和(一或多個)電路互連1330A-1330B。所述一或多個MMU1320A-1320B為圖形處理器1310，包括為頂點處理器1305及/或(一或多個)片段處理器1315A-1315N，提供虛擬到實體位址映射，所述虛擬到實體位址映射除了儲存在所述一或多個快取1325A-1325B中的頂點或影像/紋理資料之外還可以參照儲存在記憶體中的頂點或影像/紋理資料。在一實施

例中，所述一或多個MMU 1320A-1320B可以與系統內的其它MMU同步，所述其它MMU包括與圖12的所述一或多個應用處理器1205、影像處理器1215及/或視訊處理器1220相關聯的一或多個MMU，使得每個處理器1205-1220可以參與共享或統一的虛擬記憶體系統。根據實施例，所述一或多個電路互連1330A-1330B使得圖形處理器1310能夠經由SoC的內部匯流排或經由直接連接與SoC內的其它IP核心對接。

【0207】**如圖14**所示，圖形處理器1340包括圖13的圖形處理器1310的一或多個MMU 1320A-1320B、快取1325A-1325B和電路互連1330A-1330B。圖形處理器1340包括一或多個著色器核心1355A-1355N(例如1355A、1355B、1355C、1355D、1355E、1355F至1355N-1和1355N)，其提供統一著色器核心架構，其中單一核心或者類型或核心能夠運行所有類型的可程式化著色器程式碼、包括著色器程式碼，以實現頂點著色器、片段著色器及/或計算著色器。存在的著色器核心的準確數量能夠在實施例和實現之間改變。另外，圖形處理器1340包括核心間任務管理器1345，其充當向一或多個著色器核心1355A-1355N分派執行執行緒的執行緒分派器以及加速基於切片的繪製的平鋪操作的平鋪單元1358，其中場景的繪製操作在影像空間中細分，例如以利用場景中的區域空間一致性或者最佳化內部快取的使用。

利用機器學習的射線追蹤

【0208】如上所述，射線追蹤是一種圖形處理技術，其中透過基於實體的繪製來模擬光傳輸。射線追蹤中的關鍵操作之一是處理可見性查詢，所述查詢需要對定界體階層(BVH)中的節點進行遍歷和相交測試。

【0209】基於射線和路徑追蹤的技術透過追蹤穿過每個像素的射線和路徑，並使用隨機取樣來計算高級效果，諸如陰影、光澤度、間接照明等來計算影像。僅使用幾個樣本是快速的，但是會產生雜訊影像，而使用許多樣本產生高品質的影像，但成本高昂。

【0210】機器學習包含能夠逐步改善指定任務的效能或逐漸呈現更準確的預測或決定的任何電路、程式碼或其組合。一些機器學習引擎可以執行這些任務或呈現這些預測/決定，而無需明確地程式化以執行任務或呈現預測/決定。存在多種機器學習技術，包含(但不限於)監督學習和半監督學習、無監督學習和強化學習。

【0211】在過去的幾年中，即時使用的射線/路徑追蹤的突破性解決方案以「降噪」的形式出現-使用影像處理技術的程序從雜訊、低取樣數輸入產生高品質的、經濾波/降噪的影像。最有效的降噪技術取決於機器學習技術，在這種技術中，機器學習引擎會學習如果使用更多樣本計算出的雜訊影像可能看起來像什麼。在一個特定的實現中，機器學習由卷積類神經網路(CNN)執行；然而，本發明的基本原理不限於CNN實現。在這種實現中，利用低

樣本數輸入和真實性來產生訓練資料。訓練CNN可以從所討論像素周圍的雜訊像素輸入附近來預測會聚像素。

【0212】儘管不完美，但是這種基於AI的降噪技術已被證明具有驚人的效果。然而，需要注意的是，由於網路可能會預測錯誤的結果，因此需要良好的訓練資料。例如，如果動畫電影製片廠對過去的電影以陸地上的場景進行了降噪CNN訓練，接著嘗試使用訓練後的CNN對水上的新電影中的訊框進行降噪，則降噪操作將表現欠佳。

【0213】為了解決這個問題，可以在繪製的同時動態地收集學習資料，並且可以基於當前正在其上運行的資料來持續訓練諸如CNN之類的機器學習引擎，從而不斷地改善手頭任務的機器學習引擎。因此，訓練階段可能仍在運行時之前執行，但是會在運行時根據需要繼續調整機器學習權重。從而，透過將學習資料的產生限制在每訊框或每N訊框的影像的子區域，避免了計算訓練所需的參考資料的高成本。特別是，產生訊框的雜訊輸入，以利用當前網路對整個訊框進行降噪。此外，如下述，產生小區域的參考像素並將其用於連續訓練。

【0214】雖然本文描述了CNN實現，但是可以使用任何形式的機器學習引擎，包含但不限於執行監督學習 (例如，建立包含輸入和所需的輸出兩者的資料集的數學模型)、無監督學習(例如，評估某些類型的結構的輸入資料)及/或監督學習和無監督學習的組合的系統。

【0215】現有的降噪實現在訓練階段和運行時階段中

操作。在訓練階段，定義了一個網路拓撲，所述拓撲接收具有各種每個像素資料通道(例如像素顏色、深度、法線、法線偏差、基元ID和反照率)的NxN像素區域，並產生最終的像素顏色。一組「代表性」訓練資料是使用一訊框的低樣本數輸入值產生的，並參照以非常高的樣本數計算出的「所需」像素顏色。網路針對這些輸入進行訓練，從而為網路產生了一組「理想」權重。在這些實現中，參考資料用於訓練網路的權重，以使網路的輸出與所需結果最接近。

【0216】在運行時，載入給定的、預先計算的理想網路權重並初始化網路。對於每一訊框，產生低樣本數的降噪輸入影像(即與用於訓練的相同)。對於每個像素，像素輸入的給定鄰域將透過網路運行以預測「降噪」的像素顏色，從而產生降噪的訊框。

【0217】**圖15**顯示了初始訓練實現。機器學習引擎1500(例如，CNN)接收N×N像素的區域作為高樣本數影像資料1702，其具有各種按像素的資料通道，諸如像素顏色、深度、法線、法線偏差、原始ID和反照率並產生最終的像素顏色。使用一訊框的低樣本數輸入值1501產生代表訓練資料。網路針對這些輸入進行訓練，產生一組「理想」權重1505，機器學習引擎1500隨後使用權重1505在運行時對低樣本數影像進行降噪。

【0218】為了改進上述技術，增加了每個訊框或訊框子集(例如，每N個訊框，其中N = 2、3、4、10、25等)產

生新訓練資料的降噪階段。特別是，如圖16中所示，選擇每一訊框中的一或多個區域，這裡稱為「新參考區域」1602，其以高樣本數呈現到單獨的高樣本數緩衝器1604中。低樣本數緩衝器1603儲存低樣本數輸入訊框1601(包含與新參考區域1602相對應的低樣本區域1604)。

【0219】新參考區域1602的位置可以被隨機選擇。可替代地，可以用針對每個新訊框的預定方式來調整新參考區域1602的位置(例如，使用訊框之間的區域的預定義移動、限於訊框中心的指定區域等)。

【0220】不管如何選擇新的參考區域，機器學習引擎1600都使用它來連續地精煉和更新用於降噪的訓練後的權重1605。特別是，呈現來自每個新參考區域1602的參考像素顏色和來自對應的低樣本數區域1607的雜訊參考像素輸入。接著使用高樣本數參考區域1602和對應的低樣本數區域1607在機器學習引擎1600上進行補充訓練。與初始訓練相反，此訓練在運行時針對每個新參考區域1602連續進行，從而確保對機器學習引擎1600進行精確訓練。例如，每個像素的資料通道(例如，像素的顏色、深度、法線、法線偏差等)可被評估，其被機器學習引擎1600用來調整訓練後的權重1605。如在訓練情況下(圖**15**)，朝著一組理想權重1605來訓練機器學習引擎1600，以從低樣本數輸入訊框1601中移除雜訊以產生降噪訊框1620。然而，基於新型的低樣本數輸入訊框1601的新影像特性，持續更新訓練的權重1605。

【0221】可以在圖形處理器單元(GPU)或主機處理器上的背景程序中同時執行由機器學習引擎1600執行的重新訓練操作。可以被實現為驅動器元件及/或GPU硬體元件的繪製迴圈可以連續地產生新的訓練資料(例如，以新的參考區域1602的形式)，並將其放置在佇列中。在GPU或主機處理器上執行的背景訓練程序可以從此佇列中連續讀取新的訓練資料、對機器學習引擎1600進行重新訓練，並以適當的時間間隔以新的權重1605對其進行更新。

【0222】**圖17**顯示由主機CPU 1710實現背景訓練程序1700的一個這種實現的範例。特別是，背景訓練程序1700使用高樣本數的新參考區域1602和對應的低樣本區域1604以連續更新訓練後的權重1605，從而更新機器學習引擎1600。

【0223】如在多玩家線上遊戲的非限制性範例的圖18A中顯示的，不同的主機機器1820-1822單獨產生背景訓練程序1700A-C發送到伺服器1800(例如，諸如遊戲伺服器)的參考區域。接著，伺服器1800使用從主機1821-1822中之各者接收的新參考區域在機器學習引擎1810上執行訓練，如前述更新權重1805。將這些權重1805發送到儲存權重1605A-C的主機機器1820，從而更新每個單獨的機器學習引擎(未顯示)。因為可以在短時間段內向伺服器1800提供大量參考區域，所以可以針對使用者正在執行的任何給定應用程式(例如，線上遊戲)有效且精確地更新權重。

【0224】如在**圖18B**中顯示的，不同的主機機器可能

會產生新訓練的權重(例如，如前述的基於訓練/參考區域1602)，並與伺服器1800共享新訓練的權重 (例如，諸如遊戲伺服器)，或者使用同級間共享協定。伺服器上的機器學習管理部件1810使用從每個主機機器接收的新權重來產生一組組合權重1805。例如，組合權重1805可以是從新權重產生並如本文所述不斷更新的平均值。一旦產生，可以在每個主機機器1820-1821上發送並儲存組合權重1605A-C的副本，接著，它們可以使用本文所述的組合權重來執行降噪操作。

【0225】硬體製造商也可以使用半閉環更新機制。例如，參考網路可以作為由硬體製造商分發的驅動程式的一部分而包含在內。當驅動程式使用本文所述的技術產生新的訓練資料並將其連續提交回硬體製造商時，硬體製造商將使用此資訊繼續改進其機器學習實現，以用於下一次驅動程式更新。

【0226】在範例實現中(例如，在繪製場上的批量電影繪製中)，繪製器將新產生的訓練區域傳輸到專用伺服器或資料庫(在所述工作室的繪製場中)，所述專用伺服器或資料庫將來自多個繪製節點的資料隨時間聚合。在單獨的機器上進行單獨的處理會不斷改善工作室的專用降噪網路，並且新的繪製作業始終使用最新的經過訓練的網路。

【0227】在**圖19**中顯示機器學習方法。所述方法可以在本文描述的架構上實現，但是不限於任何特定的系統或圖形處理架構。

【0228】在1901處，作為初始訓練階段的一部分，針對複數個影像訊框產生低樣本數影像資料和高樣本數影像資料。在1902處，使用高/低樣本數影像資料來訓練機器學習降噪引擎。例如，可以根據訓練來更新與像素特徵相關的一組卷積類神經網路權重。然而，可以使用任何機器學習架構。

【0229】在1903處，在運行時，產生低樣本數影像訊框以及具有高樣本數的至少一個參考區域。在1904處，機器學習引擎及/或單獨的訓練邏輯(例如，背景訓練模組1700)使用高樣本數參考區域來持續改進機器學習引擎的訓練。例如，高樣本數參考區域可以與低樣本數影像的對應部分結合使用，以繼續教導機器學習引擎1904如何最有效地執行降噪。例如，在CNN實現中，這可能涉及更新與CNN相關的權重。

【0230】可以實現上述多個變體，諸如，配置到機器學習引擎的反饋迴路的方式，產生訓練資料的實體，將訓練資料反饋到訓練引擎的方式，以及如何將改進的網路提供給繪製引擎。此外，儘管上述範例使用單一參考區域來執行連續訓練，但是可以使用任何數量的參考區域。此外，如前述，參考區域可以具有不同的大小、可以在不同數量的影像訊框上使用，並且可以使用不同的技術(例如，隨機、根據預定模式等)定位在影像訊框內的不同位置中。

【0231】此外，儘管將卷積類神經網路(CNN)描述為

機器學習引擎1600的一個範例，但是本發明的基本原理可以使用能夠使用新的訓練資料持續改進其功能的任何形式的機器學習引擎來實現。作為範例而非限制，其它機器學習實現包含分組資料處理方法(group method of data handling；GMDH)、長短期記憶體、深層儲存計算(deep reservoir computing)、深層信念網路、張量深層堆疊網路和深層預測編碼網路，僅舉幾例。

高效的分佈式降噪的設備和方法

【0232】如上所述，降噪已成為具有平滑、無雜訊影像的即時射線追蹤的關鍵特徵。可以在多個裝置上的分佈式系統上完成繪製，但是到目前為止，現有的降噪框架都在單一機器上的單一實例上運行。如果正在跨多個裝置進行繪製，則它們可能沒有可存取以計算影像的降噪部分的所有繪製像素。

【0233】提出了一種既與人工智慧(AI)又基於非AI的降噪技術一起工作的分佈式降噪演算法。影像的區域或者已經透過分佈式繪製操作跨節點分佈，或者從單一訊框緩衝區拆分並分佈。在需要時，從相鄰節點收集計算足夠降噪所需的相鄰區域的重影區域，並將最終得到的切片合成為最終影像。

分佈式處理

【0234】**圖20**顯示執行繪製的多個節點**2021-2023**。

儘管為了簡化僅顯示三個節點，但是本發明的基本原理不限於任何特定數量的節點。實際上，單一節點可以用於實現本發明的某些實施例。

【0235】節點2021-2023每個都繪製影像的一部分，在此範例中產生區域2011-2013。儘管在**圖20**中顯示矩形區域2011-2013，但是可以使用任何形狀的區域，並且任何裝置可以處理任何數量的區域。節點執行足夠平滑的降噪操作所需的區域稱為重影區域2011-2013。換句話說，重影區域2001-2003代表以指定的品質程度執行降噪所需的全部資料。降低品質程度會減少重影區域的大小，因此所需的資料量會增加，而提高品質程度則會增加重影區域和所需的對應資料。

【0236】如果諸如節點2021之類的節點確實具有以指定品質程度對其區域2011進行降噪所需的重影區域2001的一部分的區域副本，則所述節點將從一或多個「相鄰」節點檢索所需資料，所述「相鄰」節點，諸如擁有重影區域2001的一部分的節點2022，如圖所示。類似地，如果節點2022確實具有以指定的品質程度對其區域2012進行降噪所需的重影區域2002的一部分的區域副本，則節點2022將從節點2021檢索所需的重影區域資料2032。所述檢索可以透過匯流排、互連、高速記憶體結構、網路(例如，高速乙太網路)、甚至可以是能夠在複數個核心之間分佈繪製工作的多核心晶片中的晶載互連(例如，用於以極高解析度或時變繪製大型影像)來進行。每個節點2021-2023可以在

圖形處理器內包含單獨的執行單元或執行單元的指定集合。

【0237】要發送的特定資料量取決於所使用的降噪技術。此外，來自重影區域的資料可以包含改善每個對應區域的降噪所需的任何資料。例如，重影區域資料可以包含影像顏色/波長、強度/alpha資料及/或法線。然而，本發明的基本原理不限於任何特定的重影區域資料集。

額外細節

【0238】對於較慢的網路或互連，可以使用現有的通用無損或有損壓縮來利用所述資料的壓縮。範例包含但不限於zlib、gzip和Lempel-Ziv-Markov鏈演算法(LZMA)。透過注意訊框之間的射線命中資訊中的增量可能非常稀疏，可以使用進一步的特定於內容的壓縮，並且當節點已經具有從先前訊框中收集到的增量時，僅需要發送有助於所述增量的樣本。可以選擇性地推送到收集那些樣本*i*的節點，或者節點*i*可以從其它節點請求樣本。無損壓縮用於某些類型的資料和程式碼，而有損資料用於其它類型的資料。

【0239】圖21顯示節點2021-2022之間的互動的額外細節。每個節點2021-2022包含用於繪製各個影像區域2011-2012和重影區域2001-2002的射線追蹤繪製電路2081-2082。降噪器2100-2111分別在區域2011-2012上執行降噪操作，每個節點2021-2022負責繪製和降噪。降噪器2021-

2022例如可以包含電路、軟體或其任意組合以分別產生降噪區域2121-2122。如所提及的，當產生降噪區域時，降噪器2021-2022可能需要取決於由不同節點擁有的重影區域內的資料(例如，降噪器2100可能需要來自節點2022所擁有的重影區域2002的資料)。

【0240】因此，降噪器2100-2111可以分別使用來自區域2011-2012和重影區域2001-2002的資料來產生降噪區域2121-2122，其中至少一部分可以從另一節點接收。如本文所述，區域資料管理器2101-2102可以管理來自重影區域2001-2002的資料傳輸。壓縮器/解壓縮器單元2131-2132可以分別執行在節點2021-2022之間交換的重影區域資料的壓縮和解壓縮。

【0241】例如，節點2021的區域資料管理器2101可以根據來自節點2022的請求，將資料從重影區域2001發送到壓縮器/解壓縮器2131，壓縮器/解壓縮器2131將資料壓縮以產生壓縮資料2106，並將其傳輸到節點2022，從而減少互連、網路、匯流排或其它資料通訊鏈路上的頻寬。接著，節點2022的壓縮器/解壓縮器2132對壓縮資料2106進行解壓縮，並且降噪器2111使用解壓縮後的重影資料來產生比僅來自區域2012的資料可能具有品質更高的降噪區域2012。區域資料管理器2102可以將重影區域2001中解壓縮後的資料儲存在快取、記憶體、暫存器檔案或其它儲存中，以在產生降噪區域2122時提供給降噪器2111。可以執行一組相似的操作，以將來自重影區域2002的資料提供給

節點2021上的降噪器2100，降噪器2100將資料與來自區域2011的資料結合使用以產生品質更高的降噪區域2121。

抓取資料或繪製

【0242】如果諸如節點2021-2022的裝置之間的連接較慢(也就是說，低於臨界值等待時間及/或臨界值頻寬)，則在繪製重影區域可能區域性地比從其它裝置請求結果更快。這可以在運行時透過追蹤網路交易速度以及針對重影區域大小的線性推斷繪製時間來確定。在繪製整個重影區域更快的情況下，多個裝置可能最終繪製影像的相同部分。可以基於基本區域的變化和確定的模糊程度來調整重影區域的繪製部分的解析度。

負載平衡

【0243】靜態及/或動態負載平衡方案可以用於在各個節點2021-2023之間分配處理負載。對於動態負載平衡，由降噪濾波器確定的變異數可能需要更多的時間進行降噪，但會驅動用於繪製場景的特定區域的樣本數量，而變異數低且影像的模糊區域需要更少的樣本。可以基於來自先前訊框的資料來動態地調整分配給特定節點的特定區域，或者可以在它們繪製時在裝置之間動態通訊，以使所有裝置具有相同的工作量。

【0244】圖22顯示在各個節點2021-2022上運行的監視器2201-2202如何收集效能度量資料，包含但不限於透

過網路介面2211-2212傳輸資料所消耗的時間、將(具有和不具有重影區域資料的)一個區域降噪時所消耗的時間，以及繪製每個區域/重影區域所消耗的時間。監視器2201-2202將這些效能指標回報給管理器或負載平衡器節點2201，管理器或負載平衡器節點2201分析資料以識別每個節點2021-2022上的當前工作負載，並可能確定處理各種降噪區域2121-2122的更有效模式。接著，管理器節點2201根據檢測到的負載將用於新區域的新工作負載分配給節點2021-2022。例如，管理器節點2201可以將更多的工作發送到那些負載不大的節點，及/或從那些過載的節點重新分配工作。此外，負載平衡器節點2201可以發送重新配置命令，以調整由每個節點執行繪製及/或降噪的特定方式(上面描述了其一些範例)。

確定重影區域

【0245】可以基於由降噪器2100-2111實現的降噪演算法來確定重影區域2001-2002的大小和形狀。接著可以基於檢測到的被降噪的樣本變異數來動態修改它們各自的大小。用於AI降噪本身的學習演算法可以用於確定適當的區域大小，或者在諸如雙邊模糊的其它情況下，預定的濾波器寬度將確定重影區域2001-2002的大小。在使用學習演算法的範例性實現中，機器學習引擎可以在管理器節點2201上執行及/或機器學習的部分可以在單一節點2021-2023的每一個上執行(例如，參見圖18A-B和上方的相關文

字)。

收集最終影像

【0246】可以透過收集來自節點2021-2023中的每個節點的繪製和降噪區域來產生最終影像，而不需要重影區域或法線。在**圖22中**，例如，將降噪區域2121-2122發送到管理器節點2201的區域處理器2280，其將區域組合以產生最終降噪的影像2290，接著將其顯示在顯示器2290上。區域處理器2280可以使用各種2D合成技術組合區域。儘管顯示為單獨的元件，但是區域處理器2280和降噪影像2290可以與顯示器2290整合在一起。各種節點2021-2022可以使用直接發送技術來發送降噪區域2121-2122，並且可能使用區域資料的各種有損或無損壓縮。

【0247】隨著遊戲進入雲端中，AI降噪仍然是一項昂貴的操作。因此，可能需要在多個節點2021-2022之間分佈降噪處理，以實現傳統遊戲或需要更高訊框速率的虛擬實境(VR)的即時訊框速率。電影製片廠也經常在大型繪製場中進行繪製，可將其用於更快的降噪。

【0248】**圖23**中顯示用於執行分佈式繪製和降噪的範例性方法。所述方法可以在上述系統架構的上下文實現，但不限於任何特定的系統架構。

【0249】在2301處，將圖形工作分派到複數個節點，這些節點執行射線追蹤操作以繪製影像訊框的區域。每個節點可能已經具有執行記憶體中的操作所需的資料。例

如，兩個或更多節點可以共享公共記憶體，或者節點的區域記憶體可能已經儲存了來自先前射線追蹤操作的資料。替代地，或額外地，可以將某些資料發送到每個節點。

【0250】在2302處，確定指定降噪程度(也就是說，在可接受的效能程度)所需的「重影區域」。重影區域包含執行指定降噪程度所需的任何資料，包含一或多個其它節點擁有的資料。

【0251】在2303處，在節點之間交換與重影區域(或其部分)有關的資料。在2304處，每個節點在其各自的區域上執行降噪(例如，使用交換的資料)，並且在2305處，結果被組合以產生最終的降噪影像訊框。

【0252】諸如**圖22**所示的管理器節點或主節點可以將工作分派到節點，接著將節點執行的工作組合以產生最終影像訊框。可以使用基於同級間體的架構，其中節點是同級間體，它們交換資料以繪製和降噪最終的影像訊框。

【0253】本文描述的節點(例如，節點2021-2023)可以是經由高速網路互連的圖形處理計算系統。替代地，節點可以是耦接到高速記憶體結構的單獨的處理元件。所有節點可以共享公共虛擬記憶體空間及/或公共實體記憶體。替代地，節點可以是CPU和GPU的組合。例如，上述管理器節點2201可以是CPU及/或在CPU上執行的軟體，並且節點2021-2022可以是GPU及/或在GPU上執行的軟體。可以使用各種不同類型的節點，同時仍然符合本發明的基本原理。

範例類神經網路實現

【0254】有許多類型的類神經網路；一種簡單類型的類神經網路為前饋網路。前饋網路可以被實現為其中節點被分層佈置的非循環圖。通常，前饋網路拓撲包括由至少一個隱藏層分開的輸入層和輸出層。隱藏層將輸入層接收到的輸入變換為可用於在輸出層中產生輸出的表示。網路節點經由邊緣完全連接到相鄰層中的節點，但每層中的節點之間沒有邊緣。在前饋網路之輸入層的節點處接收到的資料經由啟用函數被傳播(即，「前饋」)到輸出層的節點，啟用函數基於分別與連接層的每個邊緣相關聯的係數(「權重」)來計算網路中的每個連續層的節點的狀態。根據正在執行的演算法所表示的具體模型，類神經網路演算法的輸出可以採取各種形式。

【0255】在可以使用機器學習演算法來模擬特定問題之前，使用訓練資料組來訓練演算法。訓練類神經網路涉及選擇網路拓撲，使用表示由網路建模的問題之一組訓練資料，並且調整權重直到網路模型以訓練資料組的所有實例的最小誤差執行。例如，在用於類神經網路的監督學習訓練過程期間，回應於表示訓練資料組中的實例的輸入由網路產生的輸出與實例之標籤為「正確」的輸出進行比較、計算表示輸出和標籤的輸出之間差異的誤差訊號、以及調整與連接相關聯的權重，以在誤差訊號向後傳播通過網路層時最小化誤差。當從訓練的資料組之實例產生的每

一個輸出的誤差最小化時，網路被認為是「訓練的」。

【0256】機器學習演算法的準確度會受到用於訓練演算法的資料組之品質的顯著影響。訓練處理可能是計算密集型的，並且可能在習知通用處理器上需要大量的時間。因此，平行處理硬體被用於訓練許多類型的機器學習演算法。這對於優化類神經網路的訓練特別有用，因為在調整類神經網路中的係數中執行的計算本身適合於平行實現。具體來說，許多機器學習演算法和軟體應用已經適應於利用通用圖形處理裝置內的平行處理硬體。

【0257】圖24為機器學習軟體堆疊2400的概括圖。機器學習應用2402可被組態以使用訓練的資料組來訓練類神經網路或組態以使用訓練的深度類神經網路來實施機器智能。機器學習應用2402可包括用於在部署之前可以用於訓練類神經網路的類神經網路及/或專用軟體的訓練和推理功能。機器學習應用2402可以實施任何類型的機器智能，包括但不限於影像識別、映射和定位、自主導航、語音合成、醫學影像或語言轉譯。

【0258】機器學習應用2402的硬體加速可以經由機器學習框架2404啟用。機器學習框架2404可以在本文所述的硬體上實現，例如包含本文所述的處理器和部件的處理系統100。針對圖24描述的具有與本文中任何其它附圖的元件相同或相似名稱的元件描述了與其它附圖中相同的元件，可以用與之相似的方式操作或起作用，可以包含相同的部件，並且可以是鏈接到其它實體(如本文其它地方的

實體)，但不限於此。機器學習框架2404可以提供機器學習基元庫。機器學習基元通常是由機器學習演算法執行的基本操作。如果沒有機器學習框架2404，機器學習演算法的開發者將需要創建和優化與機器學習演算法相關的主要計算邏輯，然後在開發新的平行處理器時重新優化計算邏輯。相反，機器學習應用可以被組態為使用由機器學習框架2404提供的基元執行必要的計算。範例性基元包括張量卷積、激活函數和池化，其為訓練卷積類神經網路(CNN)時所執行的計算操作。機器學習框架2404還可以提供基元來實現由許多機器學習演算法(諸如矩陣和向量操作)執行的基本線性代數子程式。

【0259】機器學習框架2404可以處理從機器學習應用2402接收的輸入資料並且產生之計算框架2406的適當輸入。計算框架2406可提取被提供給GPGPU驅動器2408的基本指令，以致使機器學習框架2404利用經由GPGPU硬體2410之硬體加速而不需要機器學習框架2404對GPGPU硬體2410之架構深入了解。另外，計算框架2406可跨多種類型及世代的GPGPU硬體2410來致使機器學習框架2404之硬體加速。

【0260】

GPGPU機器學習加速

【0261】**圖25**顯示多GPU計算系統2500，它可以是處理系統100的變體。因此，與本文中的處理系統100結合的任何特徵的揭露也揭露了與多GPU計算系統2500的對應結

合，但不限於此。具有與本文中任何其它附圖的元件相同或相似名稱的圖25的元件描述了與其它附圖中相同的元件，可以用與之相似的方式操作或運行，可以包含相同的部件，並且可以鏈接如本文其它各處所描述的實體，但不限於此。多GPU計算系統2500可以包含經由主機介面開關2504耦接到多個GPGPU 2506A-D的處理器2502。主機介面開關2504可以是例如將處理器2502耦接到PCI express匯流排上的PCI express開關裝置，處理器2502可以透過PCI express匯流排與GPGPU 2506A-D的集合通訊。多個GPGPU 2506A-D中之各者可以是上述GPGPU的實例。GPGPU 2506A-D可經由一組高速點對點GPU至GPU鏈路2516來互連。高速GPU至GPU鏈路可經由專用GPU鏈路來連接至GPGPU 2506A-D之各者。P2P GPU鏈路2516致使GPGPU 2506A-D的各者之間的直接通訊而不需要透過主介面匯流排(其連接至處理器2502)來通訊。利用導引至P2P GPU鏈路之GPU至GPU流量，主介面匯流排保持可用於系統記憶體存取或者與多GPU計算系統2500的其它實例通訊，例如，經由一或多個網路裝置。代替將GPGPU 2506A-D經由主介面開關2504而連接至處理器2502，處理器2502可包括對於P2P GPU鏈路2516之直接支持並可直接地連接至GPGPU 2506A-D。

【0262】

機器學習類神經網路實現

【0263】由本文中所述之計算架構可被組態成執行多

類型的平行處理，其特別適於用於機器學習之訓練及部署類神經網路。類神經網路可被一般化成為具有圖形關係之功能的網路。如本領域中眾所周知，有用於機器學習的多種類型的類神經網路實現。如先前所述的，類神經網路的一種範例性類型為前饋網路。

【0264】類神經網路之第二範例性類型為卷積類神經網路(CNN)。CNN為用以處理具有已知、光柵狀拓撲之資料(諸如，影像資料)的一種專門前饋類神經網路。因此，CNN常用於計算視覺和影像辨識應用，但其還可被用在其它類型的圖案辨識，諸如語音及語言處理。CNN輸入層中的節點被組織為一組「過濾器」(由視網膜中所發現的個別欄位所驅使的特徵檢測器)，且各組過濾器之輸出被傳播至該網路之連續層中的節點。CNN之計算包括將卷積數學計算應用於各過濾器以產生該過濾器之輸出。卷積為由兩個函數所執行用以產生兩個原始函數之一的修改版本之第三函數之特殊種類的數學計算。於卷積網路術語中，至卷積的第一函數可被稱為輸入，而第二函數可被稱為卷積內核。輸出可被稱為特徵映射。例如，至卷積層之輸入可以是定義輸入影像之各種顏色成分的多維陣列資料。卷積內核可以是參數的多維陣列，其中該些參數係由類神經網路的訓練程序所調適。

【0265】遞迴類神經網路(RNN)為前饋類神經網路之家族，其包括介於層之間的回饋連接。RNN係藉由共享跨類神經網路之不同部分的參數資料以致使連續資料之建

模。RNN之架構包括週期。該些週期代表變數的目前值對於未來時刻的其本身值的影響，因為來自RNN之輸出資料的至少一部分被使用為用以處理序列中之後續輸入的回饋。此特徵使得RNN特別有用於語言處理，因為語言資料可被組成的可變本機。

【0266】以下所述之圖形呈現範例的前饋、CNN與RNN網路，以及描述用以個別地訓練並部署那些類型的網路之各者的一般程序。將理解，這些描述是範例的而非限制的，且一般來說，所顯示之概念可被通常地應用於深度類神經網路及機器學習技術。

【0267】上述範例性類神經網路可被用來執行深度學習。深度學習為使用深度類神經網路之機器學習。用於深度學習之深度類神經網路為由多個隱藏層所組成的人工類神經網路，不同於僅包括單一隱藏層的淺類神經網路。較深的類神經網路對於訓練通常在計算上是較為密集的。然而，網路的額外隱藏層致使多步驟圖案辨識，其導致相對於淺機器學習技術的降低輸出錯誤。

【0268】用於深度學習之深度類神經網路通常包括用以執行特徵辨識之前端網路，其耦接至後端網路，該後端網路代表一種數學模型，其可根據提供至該模型之特徵表示來執行操作(例如，物件分類、語音辨識，等等)。深度學習致使機器學習被執行而不需要針對該模型執行手動調整特徵工程。取而代之地，深度類神經網路可根據輸入資料內的統計結構或相關性來學習特徵。經學習的特徵可被

提供至一種可將測得特徵映射至輸出的數學模型。由該網路所使用之數學模型一般針對待執行的特定工作而被特殊化，並且不同的模型將用以執行不同的任務。

【0269】一旦類神經網路被構成，學習模型可被應用至該網路以訓練該網路來執行特定工作。該學習模型描述如何調整該模型內的權重以減少該網路之輸出錯誤。錯誤的後向傳播為用以訓練類神經網路的常見方法。輸入向量被提交至該網路以供處理。該網路之輸出係使用損失函數來與所欲輸出進行比較，並針對輸出層中之神經元的各者來計算誤差值。誤差值被接著向後傳播，直到各神經元具有約略地表示其對於原始輸出的貢獻的相關誤差值。該網路可接著使用演算法(諸如隨機梯度下降演算法)自那些錯誤學習，以更新該類神經網路之權重。

【0270】圖26及27顯示範例性卷積類神經網路。圖26顯示CNN內的各個層。如圖26中所示，用以建模影像處理之範例CNN可接收描述輸入影像之紅、綠及藍(RGB)成分的輸入2602。輸入2602可由多個卷積層(例如，卷積層2604、卷積層2606)來處理。來自多個卷積層之輸出可選擇性地由一組完全連接層2608來處理。如先前針對前饋網路所述的，完全連接層中之神經元具有對於先前層中之所有啟動的完全連接。來自完全連接層2608之輸出可被用來產生來自該網路之輸出結果。完全連接層2608內的啟動可使用取代卷積的矩陣乘法來計算。並非所有CNN的實現皆利用完全連接層。例如，在一些實現中，卷積層2606可產

生輸出給該CNN。

【0271】卷積層被稀疏地連接，其不同於完全連接層2608中所見之傳統類神經網路組態。傳統類神經網路層被完全連接，以致使每一輸出單元係與每一輸入單元互動。然而，如圖所示，卷積層被稀疏地連接，因為欄位之卷積的輸出被輸入(取代該欄位中之各節點的個別狀態值)至後續層之節點。與該些卷積層相關的內核係執行卷積操作，其輸出被傳送至下一層。在卷積層內所執行的維度減少為致使CNN縮放以供處理大影像之一種形態。

【0272】圖27顯示CNN之卷積層內的範例性計算階段。CNN之卷積層2712的輸入可在卷積層2714的三個階段中被處理。該三個階段可包括卷積階段2716、檢測器階段2718和池化階段2720。卷積層2714可接著輸出資料至連續卷積層。該網路的最終卷積層可產生輸出特徵映射資料或提供輸入至完全連接層，例如，用以產生輸入的排序值給該CNN。

【0273】在卷積階段2716中，平行地執行數個卷積以產生一組線性啟動。卷積階段2716可包括仿射變換，其為可被指明為線性變換加上轉換之任何變換。仿射變換包括旋轉、轉換、縮放及這些變換之組合。卷積階段計算函數(例如，神經元)之輸出，其被連接至輸入中之特定區，其可被判定為與該神經元相關的區域區。神經元計算介於該些神經元的權重與該些神經元所連接至之該區域輸入中的該區之間的內積。來自卷積階段2716之輸出係定義一組線

性啟動，其由卷積層2714之連續階段來處理。

【0274】該些線性啟動可由檢測器階段2718來處理。在檢測器階段2718中，各線性啟動係由非線性啟動函數來處理。非線性啟動函數係增加整體網路之非線性性質而不影響卷積層之個別欄位。數種類型的非線性啟動函數可被使用。一種特別類型是整流線性單元(ReLU)，其使用定義為f(x)=max(0,x)之啟動函數，以致使啟動係以零為臨限值。

【0275】池化階段2720係使用以附近輸出之摘要統計來替換卷積層2706之輸出的池化函數。池化函數可被用來將轉換不變性引入類神經網路，以致使對於輸入之小轉換不會改變池化的輸出。對於區域轉換之不變性可能在輸入資料中之特徵的存在比該特徵之精確位置更為重要的情境中是有用的。多種類型的池化函數可被用於池化階段2720期間，包括最大池化、平均池化及12-範數(norm)池化。此外，一些CNN實現不包括池化階段。取而代之地，此類實現係以相對於先前卷積階段具有增加的跨步的額外卷積階段來代替。

【0276】來自卷積層2714之輸出可接著由下一層2722來處理。下一層2722可以是額外卷積層或者完全連接層2708之一。例如，圖27的第一卷積層2704可輸出至第二卷積層2706，而第二卷積層可輸出至完全連接層2708的第一層。

【0277】圖28顯示範例性遞迴類神經網路2800。在遞

迴類神經網路(RNN)中，網路的先前狀態係影響網路的目前狀態的輸出。RNN可使用多種功能來以多種方式被建立。RNN之使用通常圍繞著使用數學模型以根據輸入之先前序列來預測未來。例如，RNN可被用來執行統計語言建模來根據字詞之先前序列而預測將來的字詞。所示的RNN 2800可被描述為具有接收輸入向量的輸入層2802、用以實施遞迴函數的隱藏層2804、用以致使先前狀態之「記憶體」的回饋機制2805及用以輸出結果的輸出層2806。RNN 2800係根據時間步驟來操作。在既定時間步驟之RNN的狀態係經由回饋機制2805根據先前時間步驟而被影響。針對既定時間步驟，隱藏層2804之狀態係由先前狀態以及目前時間步驟之輸入來定義。在第一時間步驟之初始輸入(x1)可由隱藏層2804來處理。第二輸入(x2)可由隱藏層2804使用在該初始輸入(x1)之處理期間所判定的狀態資訊來處理。既定狀態可被計算為s_t=f(Ux_t+ Ws_(t-1))，其中U和W為參數矩陣。函數f通常為非線性，諸如雙曲線正切函數(Tanh)或整流函數f(x)=max(0,x)之變化。然而，用於隱藏層2804之特定數學函數可根據RNN 2800之特定實現細節而變化。

【0278】除了所述的基本CNN及RNN網路之外，對於那些網路的變化可被致使。一種範例RNN變化為長短期記憶體(LSTM)RNN。LSTM RNN能夠學習對於處理較長的語言序列可能是必要的之長期依存性。CNN之變化為一種卷積深信網路，其具有類似於CNN之結構且係以一種類似於

深信網路之方式來訓練。深信網路(DBN)為一種產生的類神經網路，其由隨機(任意)變數的多個層所組成。DBN可使用貪心非監督學習而被層接層地訓練。DBN之經學習的權重可接著藉由判定針對該類神經網路之最佳初始組的權重被用來提供預先訓練類神經網路。

【0279】圖29顯示深度類神經網路的訓練和部署。一旦既定網路已針對一工作而被構成，則類神經網路係使用訓練資料集2902而被訓練。各種訓練框架2904已被開發來致使訓練程序之硬體加速。例如，上面所述機器學習框架可被組態成訓練框架。訓練框架2904可連結入未訓練類神經網路2906並致使未訓練類神經網使用本文中所述之平行處理資源而被訓練，以產生經訓練類神經網2908。

【0280】為了開始訓練程序，初始權重可被隨機地選擇或者藉由使用深信網路來預先訓練。訓練週期接著以受監督或未監督中任一者的方式被執行。

【0281】受監督學習係一種學習方法，其中訓練被執行為中介操作，諸如當訓練資料集2902包括與針對輸入的所欲輸出配對的輸入時，或其中訓練資料集包括具有已知輸出之輸入且類神經網路之輸出被手動地分級。該網路處理該些輸入並將所得輸出與一組預期或所欲輸出進行比較。錯誤接著透過系統回傳。訓練框架2904可調整控制未訓練類神經網路2906之權重。訓練框架2904可提供工具來監視未訓練類神經網路2906如何適當地根據已知的輸入資料以朝向一適於產生正確答案之模型收斂。隨著該網路之

權重被調整以改良由類神經網路所產生的輸出，訓練程序係重複地發生。訓練程序可持續，直到類神經網路達到與經訓練類神經網2908相關之統計上理想的準確度。經訓練類神經網路2908可接著被部署以實現任何數目的機器學習操作。

【0282】未監督學習為一種網路嘗試使用未標示資料來訓練其本身的學習方法。因此，針對未監督學習，訓練資料集2902將包括輸入資料而沒有任何相關的輸出資料。未訓練類神經網路2906可學習未標示輸入內的學習群集，並可判定個別輸入如何與整體資料集相關。未監督訓練可被用來產生自行組織映射，其為一種能夠執行可用來減少資料之維度的操作之經訓練類神經網路2907的類型。未監督訓練還可被用來執行異常檢測，其容許從資料的正常型態偏離之輸入資料集中的資料點的識別。

【0283】受監督及未監督訓練之變化還可被利用。半監督學習為一種訓練資料集2902包括相同分佈之已標示及未標示資料的混合之技術。遞增學習為受監督學習之變化，其中輸入資料被持續地用來進一步訓練模型。遞增學習致使經訓練類神經網路2908調適於新資料2912而不會忘記初始訓練期間之網路內所灌輸的知識。

【0284】無論受監督或未監督，特別是針對深度類神經網路之訓練程序可能對於單一計算節點是太計算密集的。取代使用單一計算節點，計算節點之分散式網路可被用來加速訓練程序。

【0285】圖30A為顯示分散式學習的方塊圖。分散式學習為一種使用多個分散式計算節點(諸如上述節點)來執行類神經網路之受監督或未監督訓練的訓練模型。分散式計算節點可各包括一或多個主機處理器及一或多個通用處理節點，諸如高度平行通用圖形處理單元。如圖所示，分散式學習可執行模型平行性3002、資料平行性3004或模型和資料平行性之組合。

【0286】在模型平行性3002中，分散式系統中的不同計算節點可針對單一網路之不同部分來執行訓練計算。例如，類神經網路之各層可由分散式系統中的不同處理節點來訓練。模型平行性之優點包括縮放至特別大的模型之能力。將與類神經網路之不同層相關的計算分離致使了非常大的類神經網路(其中所有層之權重將不配適入單一計算節點之記憶體)之訓練。在一些實例中，模型平行性可特別有用於執行大型類神經網路之未監督訓練。

【0287】在資料平行性3004中，分散式網路的不同節點具有該模型的完整實例且各節點係接收資料之不同部分。來自不同節點之結果被接著結合。雖然資料平行性的不同方式是可能的，但資料平行性訓練皆需要一種結合結果並將各節點間之模型參數同步化的技術。結合資料的範例性方案包括參數平均及更新為基的資料平行性。參數平均係訓練該訓練資料之子集上的各節點，並將全域參數(例如，權重、偏移)設為來自各節點之參數的平均。參數平均係使用保存參數資料的中央參數伺服器。更新為基的

資料平行性係類似於參數平均，除了對於該模型的更新被轉移，而不是將參數從節點轉移至參數伺服器。此外，更新為基的資料平行性可用一種分散方式來執行，其中該些更新被壓縮並在節點之間轉移。

【0288】結合的模型及資料平行性3006可被實現在例如各計算節點包括多個GPU的分散式系統中。各節點可具有該模型的完全實例，其中各節點內的分離GPU被用來訓練該模型之不同部分。

【0289】相對於單一機器上之訓練，分散式訓練具有增加的負荷。然而，本文中所述之平行處理器及GPGPU可各實現多種技術來減少分散式訓練之負荷，包括用以致使高頻寬GPU至GPU資料轉移及加速遠端資料同步化的技術。

範例性機器學習應用

【0290】機器學習可被應用來解決多種技術問題，包括但不限於電腦視覺、自動駕駛和導航、語音辨識及語言處理。電腦視覺傳統上已成為針對機器學習應用的最活躍研究領域之一。電腦視覺之應用的範圍從再生人類視覺能力(諸如辨識人臉)至產生新種類的視覺能力。例如，電腦視覺應用可被組態成從視訊中可見的物體中所引發的振動來辨識音波。平行處理器加速機器學習係致使電腦視覺應用使用相較於先前可行者顯著地更大的訓練資料集來訓練，並致使推理系統使用低功率平行處理器來被部署。

【0291】平行處理器加速機器學習具有自動駕駛應用，包括巷道和道路標誌辨識、障礙閃避、導航及駕駛控制。加速機器學習技術可被用來根據定義對於特定訓練輸入之適當回應的資料集來訓練駕駛模型。本文中所述之平行處理器可致使用於自動駕駛解決方案的越來越複雜的類神經網路之快速訓練，並致使一種適於整合在自動車內的行動平台中的低功率推理處理器之部署。

【0292】平行處理器加速深度類神經網路已致使自動語音辨識(ASR)的機器學習方案。ASR包括產生計算既定輸入聲波序列下最可能的語言序列之函數。使用深度類神經網路之加速機器學習已致使先前用於ASR的隱藏馬可夫(Markov)模型(HMM)及高斯(Gaussian)混合模型(GMM)的替換。

【0293】平行處理器加速機器學習還可被用來加速自然語言處理。自動學習程序可利用統計推理演算法來產生能抵抗錯誤或不熟悉輸入的模型。範例性自然語言處理器應用包括人類語言之間的自動機器轉譯。

【0294】用於機器學習的平行處理平台可被劃分為訓練平台及部署平台。訓練平台通常是高度平行的，且包括用以加速多GPU單節點訓練及多節點多GPU訓練的最佳化。適於訓練的範例性平行處理器包括本文所述之高度平行通用圖形處理單元及/或多GPU計算系統。反之，已部署的機器學習平台通常包括較低功率的平行處理器，其適於諸如相機、自動機器人及自動車等產品的使用。

【0295】**圖30B**顯示適用於使用訓練模型執行推理的範例性推理系統單晶片(SOC)3100。具有與本文中任何其它附圖的元件相同或相似名稱的圖30B的元件描述了與其它附圖中相同的元件，可以用與之相似的方式操作或起作用，可以包含相同的元件，並且可以鏈接如本文其它各處的其它實體，但不限於此。SOC 3100可整合包括媒體處理器3102、視覺處理器3104、GPGPU 3106及多核心處理器3108之處理組件。SOC 3100可額外地包括晶載記憶體3105，其可致使可由各處理組件所存取之共享的晶載資料池。處理組件可針對低功率操作來最佳化，以致使對於包括自動車及自動機器人的多種機器學習平台之部署。例如，SOC 3100的一種實現可被使用為自動車的主控制系統的一部分。當SOC 3100被組態為用於自動車時，SOC被設計並組態成符合部署管轄權之相關的功能安全性標準。

【0296】在操作期間，媒體處理器3102及視覺處理器3104可配合加速電腦視覺操作來運作。媒體處理器3102可致使多個高解析度(例如，4K、8K)視訊串流的低推遲時間解碼。經解碼的視訊串流可被寫入至晶載記憶體3105中的緩衝器。視覺處理器3104可接著解析該經解碼視訊並對於該經解碼視訊之框執行初步處理操作，以準備使用經訓練的影像辨識模型來處理該些框。例如，視覺處理器3104可加速被用來對於高解析度視訊資料執行影像辨識之CNN的卷積操作，而後端模型計算係由GPGPU 3106來執行。

【0297】多核心處理器3108可包括用以協助由媒體處

理器3102和視覺處理器3104所執行的資料轉移及共享記憶體操作之排序和同步化的控制邏輯。多核心處理器3108還可作用為用以執行可利用GPGPU 3106之推理計算能力的軟體應用的應用處理器。例如，導航和駕駛邏輯之至少一部分可被實現在多核心處理器3108上所執行的軟體。此軟體可直接地發送計算工作負載至GPGPU 3106，或者該計算工作負載可被發送至多核心處理器3108，其可將那些操作之至少一部分卸載至GPGPU 3106。

【0298】GPGPU 3106可包括計算群集，諸如高度平行通用圖形處理單元DPLAB00內的計算群集DPLAB06A-DPLAB06H的低功率組態。GPGPU 3106內的計算群集可支持被明確地最佳化以執行推理計算於經訓練類神經網路上的指令。例如，GPGPU 3106可支持用以執行諸如8位元及4位元整數向量操作的低精確度計算的指令。

射線追蹤架構

【0299】在一種實現中，圖形處理器包含用於執行即時射線追蹤的電路及/或程式碼。可以在圖形處理器中包含一組專用的射線追蹤核心，以執行本文所述的各種射線追蹤操作，包含射線遍歷及/或射線相交操作。除了射線追蹤核心之外，還可以包含用於執行可程式化著色操作的多組圖形處理核心和用於對張量資料執行矩陣運算的多組張量核心。

【0300】**圖31**顯示一個這種圖形處理單元(GPU)3105

的範例性部分，其包含佈置成多核心組3100A-N的圖形處理資源的專用集合。圖形處理單元(GPU)3105可以是圖形處理器300、GPGPU 1340及/或本文描述的任何其它圖形處理器的變體。因此，用於圖形處理器的任何特徵的揭露還揭露了與GPU 3105的對應組合，但不限於此。此外，具有與本文中任何其它附圖的元件相同或相似名稱的圖31的元件描述與其它附圖中相同的元件，可以用與之相似的方式操作或起作用、可以包含相同的部件，並且可以鏈接到其它實體，如本文其它地方所述，但不限於此。儘管僅提供了單一多核心組3100A的詳細資訊，但應理解，其它多核心組3100B-N可以配備有相同或相似的圖形處理資源集。

【0301】如圖所示，多核心組3100A可包含一組圖形核心3130、一組張量核心3140和一組射線追蹤核心3150。排程器/分派器3110排程並分派圖形執行緒以在各種核心3130、3140、3150上執行。一組暫存器檔案3120儲存了在執行圖形執行緒時核心3130、3140、3150使用的運算元值。這些可以包含例如用於儲存整數值的整數暫存器、用於儲存浮點值的浮點暫存器、用於儲存緊縮資料元件(整數及/或浮點資料元件)的向量暫存器以及用於儲存張量/矩陣值的切片暫存器。切片暫存器可以被實現為向量暫存器的組合集合。

【0302】一或多個1階(L1)快取和紋理單元3160在每個多核心組3100A中區域儲存圖形資料，諸如紋理資料、頂點資料、像素資料、射線資料、定界體資料等。由多核

心組3100A-N中所有或子集共享的2階(L2)快取3180儲存圖形資料及/或多個並發圖形執行緒的指令。如圖所示，L2快取3180可以在複數個多核心組3100A-N之間共享。一或多個記憶體控制器3170將GPU 3105耦接到記憶體3198，所述記憶體3198可以是系統記憶體(例如，DRAM)及/或專用圖形記憶體(例如，GDDR6記憶體)。

【0303】輸入/輸出(IO)電路3195將GPU 3105耦接到一或多個IO裝置3195，諸如數位訊號處理器(DSP)、網路控制器或使用者輸入裝置。晶載互連可以用於將I/O裝置3190耦接到GPU 3105和記憶體3198。IO電路3195的一或多個IO記憶體管理單元(IOMMU)3170直接將IO裝置3190耦接到系統記憶體3198。IOMMU 3170可以管理多組頁表，以將虛擬位址映射到系統記憶體3198中的實體位址。此外，IO裝置3190、CPU 3199和GPU 3105可以共享相同的虛擬位址空間。

【0304】IOMMU 3170還可以支援虛擬化。在這種情況下，它可以管理用於將訪客/圖形虛擬位址映射到訪客/圖形實體位址的第一組頁表，以及用於將訪客/圖形實體位址映射到系統/主機實體位址的第二組頁表(例如，在系統記憶體3198中)。第一和第二組頁表的每一個的基底位址可以儲存在控制暫存器中，並在上下文切換器上換出(例如，以便向新的上下文提供對相關的頁表集的存取)。儘管未在圖31中顯示，但每個核心3130、3140、3150及/或多核心組3100A-N都可以包含轉譯後備緩衝區(TLB)，

以快取訪客虛擬到訪客實體轉譯、訪客實體到主機實體轉譯，和訪客虛擬到主機實體轉譯。

【0305】CPU 3199、GPU 3105和IO裝置3190可以整合在單一半導體晶片及/或晶片封裝上。所示的記憶體3198可以整合在同一晶片，或者可以透過晶片外介面耦接到記憶體控制器3170。在一種實現中，記憶體3198包含與其它實體系統級記憶體共享相同虛擬位址空間的GDDR6記憶體，儘管本發明的基本原理不限於此特定實現。

【0306】張量核心3140可以包含複數個執行單元，這些執行單元專門設計用於執行矩陣運算，矩陣運算是用於執行深度學習運算的基本計算運算。例如，同時的矩陣乘法運算可用於類神經網路訓練和推理。張量核心3140可以使用各種運算元精確度來執行矩陣處理，包含單精確度浮點數(例如，32位元)、半精確度浮點數(例如，16位元)、整數字組(16位元)、位元組(8位元) 和半位元組(4位元)。類神經網路的實現還可以提取每個繪製方案的特徵，潛在地組合來自多個訊框的細節，以構建高品質的最終影像。

【0307】在深度學習實現中，可以排程平行矩陣乘法工作以在張量核心3140上執行。類神經網路的訓練尤其需要大量矩陣點積運算。為了處理N×N×N矩陣乘法的內積公式，張量核心3140可以包含至少N個點積處理元件。在矩陣乘法開始之前，將一個完整的矩陣載入到切片暫存器中，並在N個週期的每個週期將第二矩陣的至少一列載入。每個週期有N個點積被處理。

【0308】取決於特定的實現，可以用不同的精確度儲存矩陣元件，包含16位元字組、8位元位元組(例如，INT8)和4位元半位元組(例如，INT4)。可以為張量核心3140指定不同的精確度模式，以確保將最有效的精確度用於不同的工作負載(例如，諸如可以將量化容忍到位元組和半位元組的推理工作負載)。

【0309】對於即時射線追蹤和非即時射線追蹤實現，射線追蹤核心3150可以用於加速射線追蹤操作。特別是，射線追蹤核心3150可以包含射線遍歷/相交電路，所述射線遍歷/相交電路用於使用定界體階層(BVH)來執行射線遍歷並識別射線與包圍在BVH容積內的基元之間的相交。射線追蹤核心3150還可包含用於執行深度測試和剔除(例如，使用Z緩衝器或類似佈置)的電路。在一個實現中，射線追蹤核心3150與本文描述的影像降噪技術一致地執行遍歷和相交操作，其至少一部分可以在張量核心3140上執行。例如，張量核心3140可以實現深層學習類神經網路以對由射線追蹤核心3150產生的訊框執行降噪。然而，CPU 3199、圖形核心3130及/或射線追蹤核心3150也可以實現全部或部分降噪及/或深層學習演算法。

【0310】此外，如上所述，可以採用分佈式降噪方法，其中GPU 3105在透過網路或高速互連耦接到其它計算裝置的計算裝置中。互連的計算裝置可以額外地共享類神經網路學習/訓練資料，以提高整個系統學習用於針對不同類型的影像訊框及/或不同圖形應用執行降噪的速度。

【0311】射線追蹤核心3150可以處理所有BVH遍歷和射線-基元交點，從而避免圖形核心3130過載每條射線數千條指令。每個射線追蹤核心3150可包含用於執行定界框測試(例如，用於遍歷操作)的第一組專用電路和用於執行射線-三角形相交測試(例如，已被遍歷的相交射線)的第二組專用電路。因此，多核心組3100A可以簡單地啟動射線探測器，並且射線追蹤核心3150獨立地執行射線遍歷和相交並且將命中資料(例如，命中、未命中、多次命中等)返回到執行緒上下文。當射線追蹤核心3150執行遍歷和相交操作時，其它核心3130、3140可以被釋放以執行其它圖形或計算工作。

【0312】每個射線追蹤核心3150可以包含用以執行BVH測試操作的遍歷單元和執行射線-基元相交測試的相交單元。接著，相交單元可以產生「命中」、「未命中」或「多次命中」回應，並將其提供給適當的執行緒。在遍歷和相交操作期間，可以釋放其它核心(例如，圖形核心3130和張量核心3140)的執行資源以執行其它形式的圖形工作。

【0313】還可以使用工作在圖形核心3130和射線追蹤核心3150之間分配的混合光柵化/射線追蹤方法。

【0314】射線追蹤核心3150(及/或其它核心3130、3140)可包含對射線追蹤指令集的硬體支援，諸如Microsoft的DirectX射線追蹤(DXR)，其包含DispatchRays命令以及射線產生、最接近命中、任何命中和未命中的著色器，其

可為每個物件分配唯一的一組著色器和紋理。可以由射線追蹤核心3150、圖形核心3130和張量核心3140支援的另一個射線追蹤平台是Vulkan 1.1.85。然而，請注意，本發明的基本原理不限於任何特定的射線追蹤ISA。

【0315】通常，各種核心3150、3140、3130可以支援射線追蹤指令集，所述射線追蹤指令集包含用於射線產生、最接近命中、任何命中、射線-基元相交、按基元和階層定界框構建、未命中、訪問和異常的指令/功能。更具體地說，可以包含射線追蹤指令以執行以下功能：

【0316】射線產生–可以為每個像素、樣本或其它使用者定義的工作分配執行射線產生指令。

【0317】最接近命中–可以執行最接近命中指令以定位射線與場景中基元的最接近交點。

【0318】任何命中–任何命中指令可識別射線與場景中的基元之間的多個相交點，從而有可能識別新的最接近的相交點。

【0319】相交–相交指令執行射線-基元相交測試並輸出結果。

【0320】按基元定界框的構建–此指令圍繞給定的基元或基元組構建定界框(例如，在構建新的BVH或其它加速度資料結構時)。

【0321】未命中–表示射線未命中場景或場景指定區域中的所有幾何。

【0322】訪問–指示射線將遍歷的子容積。

【0323】異常–包含各種類型的異常處理程序(例如，針對各種錯誤條件而調用)。

【0324】媒體處理電路3197包含固定功能電路，其用於將媒體編碼、解碼和轉碼為一或多種媒體編碼格式，或在一或多種媒體編碼格式之間進行編碼、解碼和轉碼，包含但不限於運動影像專家組(MPEG)格式，諸如MPEG-2、進階視訊編碼(AVC)格式，諸如H.264/MPEG-4 AVC，以及美國電影電視工程師協會(SMPTE) 421M/VC-1，以及聯合攝影專家組(JPEG)格式，諸如JPEG和運動JPEG(MJPEG)格式。為了處理這些格式，媒體處理電路的一個實施例包含用於執行縮放/量化、訊框內預測、訊框間預測、運動補償和運動估計的邏輯。

【0325】顯示處理器3193可以包含2D顯示引擎和顯示控制器。顯示處理器3193還可以包含能夠獨立於GPU 3105的其它電路進行操作的專門邏輯。顯示控制器包含用於耦接至顯示裝置(未顯示)的介面，所述顯示裝置可以是系統整合的顯示裝置，例如在筆記型電腦中，或經由顯示裝置連接器連接的外部顯示裝置。

分層光束追蹤

【0326】定界體階層結構(Bounding volume hierarchies；BVH)通常用於提高對圖形基元和其它圖形物件執行操作的效率。BVH是基於一組幾何物件構建的分層樹形結構。樹形結構的頂部是根節點，所述根節點將給定場景中的所

有幾何物件包圍起來。各個幾何物件被包裹在形成樹的葉子節點的定界體中。接著將這些節點分組為小集合，並包含在較大的定界體內。接著，這些也以遞歸的方式被分組並包含在其它較大的定界體內，最終致使在樹的頂部具有根節點表示的具有單一定界體的樹形結構。定界體階層用於有效地支援一組幾何物件上的各種操作，諸如碰撞檢測、基元剔除以及射線追蹤中使用的射線遍歷/相交操作。

【0327】在射線追蹤架構中，射線透過BVH遍歷以確定射線-基元交點。例如，如果射線不穿過BVH的根節點，則所述射線不會與BVH包圍的任何基元相交，並且就此基元的集合而言，不需要對射線進行進一步的處理。如果射線穿過BVH的第一個子節點，但不穿過第二個子節點，則無需針對第二個子節點包含的任何基元測試射線。如此，BVH提供了一種有效的機制來測試射線-基元相交。

【0328】可以針對BVH而不是單獨的射線測試成組的連續射線(稱為「光束」)。**圖32**顯示由四個不同的射線概述的範例性光束3201。與由四個射線限定的貼片3200相交的任何射線都被認為在同一光束內。儘管**圖32**中的光束3201由射線的矩形佈置限定，但是可以用各種其它方式限定光束，同時仍然符合本發明的基本原理(例如，圓形、橢圓形等)。

【0329】**圖33**顯示GPU 3320的射線追蹤引擎3310如何

實現本文所述的光束追蹤技術。特別是，射線產生電路3304產生要對其執行遍歷和相交操作的複數條射線。然而，不是對單一射線執行遍歷和相交操作，而是使用由光束階層構建電路3305產生的光束3307的階層來執行遍歷和相交操作。光束階層類似於定界體階層(BVH)。例如，**圖34**提供了可以被細分為複數個不同部件的主光束3400的範例。特別是，主光束3400可以被劃分為象限3401-3404，並且每個象限本身可以被劃分為子象限，諸如象限3404內的子象限A-D。可以用多種方式將主光束細分。例如，主光束可以分為兩半(而不是象限)，並且每一半可以分為兩半，依此類推。無論如何進行細分，都以與BVH類似的方式產生分層結構，例如，具有表示主光束3400的根節點，子節點的第一級分別由象限3401-3404表示，子節點的第二級分別由子象限A-D表示，依此類推。

【0330】一旦構建了光束階層3307，遍歷/相交電路3306就可以使用光束階層3307和BVH 3308執行遍歷/相交操作。特別是，可以針對BVH和不與BVH的任何部分相交的光束的剔除部分來測試光束。使用**圖34**中所示的資料，例如，如果與子區域3402和3403相關的子光束不與BVH或BVH的特定分支相交，則可以相對於BVH或所述分支剔除它們。可以透過執行深度優先搜索或其它搜索演算法來針對BVH測試其餘部分3401、3404。

【0331】在**圖35**中顯示用於射線追蹤的方法。所述方法可以在上述圖形處理架構的上下文中實現，但不限於任

何特定架構。

【0332】在3500處，構建包含複數個射線的主光束，並且在3501處，對所述光束進行細分，並產生分層資料結構以建立光束階層。操作3500-3501可以作為單一整合操作來執行，所述單一整合操作從複數個射線構建光束階層。在3502處，光束階層與BVH一起使用以剔除射線(來自光束階層)及/或來自BVH的節點/基元。在3503處，為其餘的射線和基元確定射線基元相交。

在分佈式射線追蹤系統中的有損和無損封包壓縮

【0333】射線追蹤操作可以分佈在透過網路耦接在一起的複數個計算節點上。例如，**圖36**顯示包含複數個射線追蹤節點3610-3613的射線追蹤集群3600，其平行執行射線追蹤操作，從而有可能在節點之一上組合結果。在所示的架構中，射線追蹤節點3610-3613經由閘道器通訊地耦接到客戶側射線追蹤應用程式3630。

【0334】分佈式架構的困難之一是必須在每個射線追蹤節點3610-3613之間傳輸的大量緊縮資料。無損壓縮技術和有損壓縮技術都可以用於減少在射線追蹤節點3610-3613之間傳輸的資料。

【0335】為了實現無損壓縮，不是發送填充有某些類型的操作的結果的封包，而是發送允許接收節點以重建結果的資料或命令。例如，隨機取樣的區域光和環境光遮蔽(AO)操作不一定需要方向。因此，發送節點可以簡單地發

送隨機種子，接著接收節點將其用於執行隨機取樣。例如，如果場景分佈在節點3610-3612上(為了在點p1-p3取樣光1)則僅需要將光ID和原點發送到節點3610-3612。接著，每個節點可以獨立地隨機取樣光。隨機種子可以由接收節點產生。類似地，對於主射線命中點，可以在節點3610-3612上計算環境光遮蔽(AO)和軟陰影取樣，而無需等待連續訊框的原始點。此外，如果已知一組射線將到達同一點光源，則可以向接收節點發送指示所述光源的指令，以將其施加於所述組射線。作為另一範例，如果有N個環境光遮蔽射線透射過單一點，則可以發送命令以從此點產生N個樣本。

【0336】各種額外的技術可以施加於有損壓縮。例如，可以採用量化因子來量化與BVH、基元和射線相關的所有座標值。此外，用於資料(諸如BVH節點和基元)的32位元浮點值可以轉換為8位元整數值。在範例性實現中，射線封包的邊界以全精確度儲存，但是各個射線點P1-P3作為索引偏移被透射到邊界。類似地，可以產生使用8位元整數值作為區域座標的複數個區域座標系。這些區域座標系中之各者的原點的位置可以使用全精確度(例如32位元浮點)值來編碼，從而有效地連接全域座標系和區域座標系。

【0337】以下是無損壓縮的範例。射線追蹤程式內部使用的射線資料格式的範例如下：

```
struct Ray
{
 uint32 pixId;
 uint32 materialID;
 uint32 instanceID;
 uint64 primitiveID;
 uint32 geometryID;
 uint32 lightID;
 float origin[3];
 float direction[3];
 float t0;
 float t;
 float time;
 float normal[3];  //用於幾何相交
 float u;
 float v;

 float wavelength;
 float phase;  //Interferometry
 float refractedOffset;  //Schlieren-esque
 float amplitude;
 float weight;
};
```

【0338】代替發送所產生的每個節點的原始資料，可以透過對值進行分組並在可能的情況下使用適用的元資料建立隱式射線來壓縮所述資料。

【0339】

捆綁和分組射線資料

旗標可用於通用資料或具有修飾符的遮罩。

```
struct RayPacket
{
 uint32 size;
 uint32 flags;
 list<Ray> rays;
}
```

例如：

```
RayPacket.rays = ray_1 to ray_256
```

原點都是共享的

【0340】除了在所有射線中僅儲存單一原點之外，所有射線資料都被緊縮。為RAYPACKET_COMMON_ORIGIN設置了RayPacket.flags。接收後將RayPacket解緊縮時，將從單一原點值填充原點。

原點僅在某些射線中共享

【0341】除了共享原點的射線之外，所有射線資料都被緊縮。對於每組唯一的共享原點，都有一個運算子緊縮，所述運算子識別操作(共享原點)、儲存原點以及射線共享資訊的遮罩。可以對節點之間的任何共享值(諸如材料ID、基元ID、原點、方向、法線等)執行此操作。

```
struct RayOperation
{
  uint8 operationID;
  void* value;
  uint64 mask;
}
```

發送隱式射線

【0342】通常，可以在接收端得到射線資料，而使用最少的元資訊來產生射線資料。一個非常常見的範例是產生多個輔助射線以對區域進行隨機取樣。代替發送方產生輔助射線，將其發送並對其進行操作的接收方，發送方可以發送一條命令，即需要使用任何相關資訊來產生射線，並在接收端產生射線。在需要首先由發送方產生射線以確定將其發送到哪個接收方的情況下，產生射線並可以發送隨機種子來重新產生完全相同的射線。

【0343】例如，為了對命中點進行取樣，用64條著色射線來對區域光源進行取樣，所有64條射線與來自同一計算N4的區域相交。建立具有共同來源和法線的RayPacket。如果希望接收方著色所產生的像素貢獻，則可以發送更多的資料，但是對於本範例，我們假設我們僅希望返回射線是否命中另一節點的資料。建立RayOperation來產生著色射線操作，並為其分配要取樣的lightID值和隨機數種子。當N4接收到射線封包時，它會透過將共享的原始資料填充到所有射線中並基於利用隨機數種子隨機取樣的lightID設

置方向來產生完全填充的射線資料，以產生與原始發送者產生的相同射線。當返回結果時，僅需要返回每條射線的二進制結果，可以透過對於射線的遮罩來處理。

【0344】在此範例中，發送原始的64條射線將必須使用104位元組* 64條射線= 6656位元組。如果返回的射線也以其原始格式發送，則此也將增加一倍，達到13312位元組。使用無損壓縮僅發送常見的射線源、法線以及帶有種子和ID的射線產生操作時，僅發送29位元組，而為相交的遮罩返回8位元組。這致使需要透過大約360：1的網路發送資料壓縮率。這不包含處理訊息本身的負荷，這需要以某種方式進行識別，但這取決於實現。可以執行其它操作，以重新計算來自基元射線的像素ID的射線原點和方向、基於射線封包中的範圍重新計算像素ID，以及用於重新計算值的許多其它可能的實現。可以對發送的任何單一或一組射線使用類似的操作，包含著色、反射、折射、環境光遮蔽、相交、容積相交、著色、路徑追蹤中的反彈反射等。

【0345】**圖37**顯示兩個射線追蹤節點3710-3711的額外細節，射線追蹤節點3710-3711執行射線追蹤封包的壓縮和解壓縮。特別是，當第一射線追蹤引擎3730準備好將資料傳輸到第二射線追蹤引擎3731時，射線壓縮電路3720如本文所述執行射線追蹤資料的有損及/或無損壓縮(例如，將32位元值轉換為8位元值、將原始資料替換為重建資料的指令，等等)。壓縮射線封包3701透過區域網路(例

如10Gb/s、100Gb/s乙太網路)從網路介面3725傳輸到網路介面3726。射線解壓縮電路接著在適當時將射線封包解壓縮。例如，它可以執行命令以重建射線追蹤資料(例如，使用隨機種子來執行用於照明操作的隨機取樣)。接著，射線追蹤引擎3731使用接收到的資料來執行射線追蹤操作。

【0346】在相反的方向上，射線壓縮電路3741將射線資料壓縮，網路介面3726透過網路傳輸壓縮後的射線資料(例如，使用本文所述的技術)，射線解壓縮電路3740在必要時對射線資料進行解壓縮，並且射線追蹤引擎3730在射線追蹤操作中使用所述資料。儘管在**圖37**中被顯示為單獨的單元，但是射線解壓縮電路3740-3741可以分別整合在射線追蹤引擎3730-3731內。例如，就壓縮射線資料包含重建射線資料的命令而言，這些命令可以由每個各自的射線追蹤引擎3730-3731執行。

【0347】如**圖38**所示的，射線壓縮電路3720可以包含有損壓縮電路3801，以執行本文所描述的有損壓縮技術(例如，將32位元浮點座標轉換為8位元整數座標)和無損壓縮電路3803，以執行無損壓縮技術(例如，發送命令和資料以允許射線重新壓縮電路3821重建資料)。射線解壓縮電路3721包含有損解壓縮電路3802和用於執行無損解壓縮的無損解壓縮電路3804。

【0348】**圖39**中顯示另一範例性方法。所述方法可以在射線追蹤架構或本文描述的其它架構上實現，但不限於

任何特定架構。

【0349】在3900處，接收射線資料，所述射線資料將從第一射線追蹤節點傳輸到第二射線追蹤節點。在3901處，有損壓縮電路對第一射線追蹤資料執行有損壓縮，並且在3902處，無損壓縮電路對第二射線追蹤資料執行無損壓縮。在3903處，將壓縮的射線追蹤資料發送到第二射線追蹤節點。在3904處，有損/無損解壓縮電路執行射線追蹤資料的有損/無損解壓縮，並且在3905處，第二射線追蹤節點執行對解壓縮的資料進行追蹤的射線追蹤操作。

具有硬體加速混合射線追蹤的圖形處理器

【0350】接下來，提出一種混合繪製管線，所述混合繪製管線在圖形核心3130上執行光柵化，並在射線追蹤核心3150、圖形核心3130及/或CPU 3199核心上執行射線追蹤操作。例如，代替主射線投射台，可以在圖形核心3130上執行光柵化和深度測試。射線追蹤核心3150接著可以產生用於射線反射、折射和陰影的輔助射線。此外，將選擇射線追蹤核心3150將執行射線追蹤操作的場景的某些區域(例如，基於諸如高反射率程度的材料屬性臨界值)，而場景的其它區域將在圖形核心3130進行柵格化繪製。此混合實現可用於即時射線追蹤應用，其中潛伏時間是一個關鍵問題。

【0351】下面描述的射線遍歷架構可以例如使用現有的單一指令多重資料(SIMD)及/或單一指令多重執行緒

(SIMT)圖形處理器執行可程式化著色和射線遍歷控制，同時使用專用硬體加速進行諸如BVH遍歷及/或相交的關鍵功能。透過在遍歷期間和著色之前的特定點重新組合產生的著色器，可以改善非相干路徑的SIMD佔用率。這是透過使用專用硬體在晶片上動態排序著色器實現的。透過將函數拆分為連續部分來管理遞歸，這些連續部分在執行之前返回並重新組合連續部分時執行，以提高SIMD的佔用率。

【0352】透過將遍歷功能分解為可以實現為固定功能硬體的內部遍歷和在GPU處理器上執行並透過使用者定義的遍歷著色器實現可程式化控制的外部遍歷，來實現射線遍歷/相交的可程式化控制。透過在內部和外部遍歷之間的過渡期間保守地截斷內部遍歷狀態，可以降低在硬體和軟體之間傳遞遍歷上下文的成本。

【0353】可以透過下表A中列出的不同著色器類型來表示射線追蹤的可程式化控制。每種類型可以有多個著色器。例如，每種材料可以具有不同的命中著色器。

著色器類型	功能性
主	發射主射線
命中	雙向反射分佈函數(BRDF)取樣，發射輔助射線
任何命中	計算alpha紋理幾何的透射率
未命中	計算來自光源的輻射
相交	相交自定義形狀
遍歷	實例選擇和轉換
可調用	通用功能

表 A

【0354】可以透過API函數來啟動遞歸射線追蹤，所述API函數命令圖形處理器啟動一組主著色器或相交電路，其可以產生用於主射線的射線場景相交。依次產生其它著色器，諸如遍歷、命中著色器或未命中著色器。產生子著色器的著色器也可以從所述子著色器接收返回值。可調用著色器是通用功能的，其可以直接由另一個著色器產生，也可以將值返回給調用著色器。

【0355】**圖40**顯示包含著色器執行電路4000和固定功能電路4010的圖形處理架構。通用執行硬體子系統包含複數個單一指令多重資料(SIMD)及/或單一指令多重執行緒(SIMT)核心/執行單元(EU)4001(也就是說，每個核心可以包含複數個執行單元)、一或多個取樣器4002以及1階(L1)快取4003或其它形式的區域記憶體。固定功能硬體子系統4010包含訊息單元4004、排程器4007、射線BVH遍歷/相交電路4005、排序電路4008和區域L1快取4006。

【0356】在操作中，主分派器4009向排程器4007分派一組主射線，排程器4007將工作排程到在SIMD/SIMT核心/EU 4001上執行的著色器。SIMD核心/EU 4001可以是射線追蹤核心3150及/或上述圖形核心3130。主著色器的執行產生了要執行的額外工作(例如，將由一或多個子著色器及/或固定功能硬體執行)。訊息單元4004將由SIMD核心/EU 4001產生的工作分配到排程器4007、存取所需要的閒置堆疊池、排序電路4008或射線BVH相交電路4005。如果

額外工作被發送到排程器4007，將其排程為在SIMD/SIMT核心/EU 4001上進行處理。在排程之前，排序電路4008可將射線排序為如本文所述的分組或二元子(例如，將具有相似特性的射線分組)。射線BVH相交電路4005使用BVH容積來執行射線的相交測試。例如，射線BVH相交電路4005可以將射線座標與BVH的每個級別進行比較以識別被射線相交的容積。

【0357】可以使用著色器記錄，使用者分配的結構來參照著色器，所述使用者分配的結構包含指向入口函數的指標，特定於供應商的元資料以及由SIMD核心/EU 4001執行的著色器的全域參數。著色器的每個執行實例與調用堆疊關聯，所述調用堆疊可用於儲存在父著色器和子著色器之間傳遞的參數。調用堆疊還可以儲存對調用返回時執行的繼續功能的參照。

【0358】**圖41**顯示一組範例分配的堆疊4101，其包含主著色器堆疊、命中著色器堆疊、遍歷著色器堆疊、延續函數堆疊和射線BVH相交堆疊(如所描述的，可以是由固定功能硬體4010執行)。新的著色器調用可以實現來自閒置堆疊池4102的新堆疊。可以將調用堆疊(例如，由一組指定堆疊組成的堆疊)快取在區域L1快取4003、4006中，以減少存取的推遲。

【0359】可能存在有限數量的調用堆疊，每個調用堆疊在記憶體的連續區域中分配有固定的最大大小「Sstack」。因此，可以從堆疊索引(SID)直接計算堆疊的

基底位址，因為基底位址= SID * Sstack。當將工作排程到SIMD核心/EU 4001時，可由排程器4007分配和釋放堆疊ID。

【0360】主分派器4009可以包含圖形處理器命令處理器，其回應於來自主機(例如，CPU)的分派命令來分派主著色器。如果排程器4007可以為每個SIMD通道分配堆疊ID，則排程器4007可以接收這些分派請求並在SIMD處理器執行緒上啟動主著色器。可以從在分派命令的開始處初始化的閒置堆疊池4102分配堆疊ID。

【0361】執行著色器可以透過向訊息傳遞單元4004發送產生訊息來產生子著色器。此命令包含與所述著色器相關的堆疊ID，並且還包含指向每個活動SIMD通道的子著色器記錄的指標。父著色器只能為活動通道發出此訊息一次。發送所有相關通道的產生訊息後，父著色器可能會終止。

【0362】在SIMD核心/EU 4001上執行的著色器還可以使用產生訊息以及為固定功能硬體保留的著色器記錄指標，產生諸如射線BVH相交之類的固定功能任務。如所提到的，訊息傳遞單元4004將產生的射線BVH相交工作發送到固定功能射線BVH相交電路4005，並且將可調用著色器直接發送給排序電路4008。排序電路可以透過著色器記錄指標對著色器進行分組以得出具有相似特徵的SIMD批次。因此，來自不同父著色器的堆疊ID可以由排序電路4008在同一批次中分組。排序電路4008將分組的批次發送

到排程器4007，排程器4007從圖形記憶體2511或最末級快取(LLC)4020存取著色器記錄，並在處理器執行緒上啟動著色器。

【0363】連續(continuation)可以被視為可調用著色器，並且也可以透過著色器記錄來參照。當產生子著色器並將值返回給父著色器時，可以將指向連續著色器記錄的指標推到調用堆疊4101上。當子著色器返回時，可以從調用堆疊4101彈出連續著色器記錄，並且可能會產生連續著色器。選擇性地，產生的連續可以透過類似於可調用著色器的排序單元進行，並在處理器執行緒上啟動。

【0364】如在**圖42**顯示的，排序電路4008由著色器記錄指標4201A、4201B、4201*n*將產生的任務分組以建立SIMD批次以供著色。可以從不同的分派和不同的輸入SIMD通道對已排序批次中的堆疊ID或上下文ID進行分組。分組電路4210可以使用包含複數個條目的內容可定址記憶體(CAM)結構4201來執行排序，其中每個條目都用標籤4201來識別。如上所述，標籤4201可以是對應的著色器記錄指標4201A、4201B、4201*n*。CAM結構4201可以儲存有限數量的標籤(例如32、64、128等)，每個標籤與對應於著色器記錄指標的不完整的SIMD批次相關。

【0365】對於傳入的產生命令，每個SIMD通道具有對應的堆疊ID(在每個CAM條目中顯示為16個上下文ID 0-15)和著色器記錄指標4201A-B、...*n*(用作標籤值)。分組電路4210可以將每個通道的著色器記錄指標與CAM結構4201

中的標籤4201進行比較，以找到匹配的批次。如果找到匹配的批次，則可以將堆疊ID/上下文ID添加到所述批次。否則，可能會建立帶有新的著色器記錄指標標籤的新條目，從而可能會淘汰具有不完整批次的舊條目。

【0366】執行著色器可以透過將取消分配的訊息發送到訊息單元來在調用堆疊為空時取消分配所述調用堆疊。取消分配訊息將中繼到排程器，所述排程器將活動SIMD通道的堆疊ID/上下文ID返回到閒置池。

【0367】提出了使用固定功能射線遍歷和軟體射線遍歷的組合的用於射線遍歷操作的混合方法。因此，在保持固定功能遍歷效率的同時，提供了軟體遍歷的靈活性。**圖43**顯示可用於混合遍歷的加速結構，其為具有單一頂層BVH 4300和數個底層BVH 4301和4302的兩級樹。右側顯示圖形元件，以指示內部遍歷路徑4303、外部遍歷路徑4304、遍歷節點4305、具有三角形的葉節點4306和具有自定義基元的葉節點4307。

【0368】頂級BVH 4300中具有三角形的葉節點4306可以參照三角形，自定義基元的相交著色器記錄或遍歷著色器記錄。底層BVH 4301-4302的具有三角形的葉節點4306只能參照三角形和自定義基元的相交著色器記錄。參照的類型在葉節點4306中進行編碼。內部遍歷4303是指每個BVH 4300-4302中的遍歷。內部遍歷操作包含射線BVH交點的計算，並且跨越BVH結構4300-4302的遍歷稱為外部遍歷。內部遍歷操作可以在固定功能的硬體中高效實

現，而外部遍歷操作可以使用可程式化著色器以可接受的效能執行。因此，可以使用固定功能電路4010來執行內部遍歷操作，並且可以使用包含用於執行可程式化著色器的SIMD/SIMT核心/EU 4001的著色器執行電路4000來執行外部遍歷操作。

【0369】注意，為簡單起見，有時在本文中將SIMD/SIMT核心/EU 4001簡稱為「核心」、「SIMD核心」、「EU」或「SIMD處理器」。類似地，射線BVH遍歷/相交電路4005有時簡稱為「遍歷單元」、「遍歷/相交單元」或「遍歷/相交電路」。當使用替代用語時，如本文所述，用於指定對應電路/邏輯的特定名稱不會改變電路/邏輯執行的基礎功能。

【0370】此外，儘管出於解釋目的在**圖40**中被顯示為單一部件，但是遍歷/相交單元4005可以包含不同的遍歷單元和單獨的相交單元，它們中之各者都可以如本文所述在電路及/或邏輯中實現。

【0371】當射線在內部遍歷期間與遍歷節點相交時，可以產生遍歷著色器。排序電路4008可以透過著色器記錄指標4201A-B、*n*將這些著色器分組，以建立由排程器4007啟動以在圖形SIMD核心/EU 4001上執行SIMD的SIMD批次。遍歷著色器可以用幾種方式修改遍歷，支援廣泛的應用。例如，遍歷著色器可以選擇較粗糙的細節程度(LOD)的BVH或對射線進行轉換以實現剛體轉換。接著，遍歷著色器可以產生所選BVH的內部遍歷。

【0372】內部遍歷透過遍歷BVH並且計算射線-盒和射線-三角形相交來計算射線BVH相交。透過向訊息傳遞電路4004發送訊息來將內部遍歷以與著色器相同的方式產生，訊息傳遞電路4004將對應的產生訊息中繼到計算射線BVH相交的射線BVH相交電路4005。

【0373】用於內部遍歷的堆疊可以區域儲存在固定功能電路4010中(例如，在L1快取4006內)。當射線與對應於遍歷著色器或相交著色器的葉節點相交時，內部遍歷可能會終止，而內部堆疊會被截斷。可以將截斷的堆疊以及指向射線和BVH的指標寫入調用著色器指定的位置的記憶體中，接著可以產生對應的遍歷著色器或相交著色器。如果射線在內部遍歷期間與任何三角形相交，則可以將對應的命中資訊作為這些著色器的輸入參數提供，如以下碼所示。這些產生的著色器可以由排序電路4008分組以建立SIMD批次以便執行。

```
struct HitInfo {
    float barycentrics[2];
    float tmax;
    bool  innerTravComplete;
    uint  primID;
    uint  geomID;
    ShaderRecord* leafShaderRecord;
}
```

【0374】截斷內部遍歷堆疊減少了將其溢出到記憶體

的成本。「*Restart Trail for Stackless BVH Traversal, High Performance Graphics (2010)*」的第107–111頁中描述的方法，用於將堆疊截斷為堆疊頂部的少量條目，42位元重啟追蹤和6位元深度值可以被施加。重啟追蹤指示已在BVH內部進行的分支，深度值指示與最後一個堆疊條目相對應的遍歷深度。這是足夠的資訊，可在稍後恢復內部遍歷。

【0375】當內部堆疊為空並且沒有更多的BVH節點要測試時，內部遍歷完成。在這種情況下，將產生一個外部堆疊處理程序，所述處理程序會彈出外部堆疊的頂部，如果外部堆疊不為空，則恢復遍歷。

【0376】外部遍歷可以執行主遍歷狀態機，並且可以在由著色器執行電路4000執行的程式碼中實現。可以在以下條件下產生內部遍歷查詢：(1)當新射線由命中著色器或主著色器產生時；(2)當遍歷著色器選擇BVH進行遍歷時；及(3)當外部堆疊處理程序恢復BVH的內部遍歷時。

【0377】如在**圖44**中所示，內部遍歷產生之前，空間被分配到調用堆疊4405上，以供固定功能電路4010來儲存截斷內部堆疊4410。對於調用堆疊的頂部和內部堆疊的偏移4403-4404被保持在遍歷狀態4400中，遍歷狀態4400也儲存在記憶體2511中。遍歷狀態4400還包含世界空間4401和物件空間4402中的射線以及最接近的相交基元的命中資訊。

【0378】遍歷著色器、相交著色器和外部堆疊處理程序都由射線BVH相交電路4005產生。遍歷著色器在為第二

級BVH啟動新的內部遍歷之前在調用堆疊4405上分配。外部堆疊處理程序是負責更新命中資訊並恢復所有未完成的內部遍歷任務的著色器。遍歷完成後，外部堆疊處理程序還負責產生命中或未命中著色器。當沒有待處理的內部遍歷查詢產生時，遍歷完成。遍歷完成並找到相交後，將產生命中著色器；否則，將產生未命中著色器。

【0379】雖然上述的混合遍歷方案使用兩級BVH階層，但也可以實現任意數量的BVH階層，並在外部遍歷實現中進行對應的更改。

【0380】此外，儘管以上描述了用於執行射線BVH相交的固定功能電路4010，但是其它系統部件也可以在固定功能電路中實現。例如，上述外部堆疊處理程序可以是內部(使用者不可見)著色器，其可以潛在地在固定功能BVH遍歷/相交電路4005中實現。此實現可用於減少分派的著色器階段的數量和在固定功能相交硬體4005和處理器之間的往返。

【0381】本文描述的範例使用使用者定義的功能實現可程式化的著色和射線遍歷控制，這些功能可以在現有和將來的GPU處理器上以更高的SIMD效率執行。射線遍歷的可程式化控制實現了幾個重要功能，諸如程序實例化、隨機細節程度選擇、自定義基元相交和惰性BVH更新。

【0382】還提供了可程式化的多重指令多重資料(MIMD)射線追蹤架構，其支援對命中和相交著色器的推測執行。特別是，所述架構關注於減少在上面關於**圖40**描

述的可程式化SIMD/SIMT核心/執行單元4001與混合射線追蹤架構中的固定功能MIMD遍歷/相交單元4005之間的排程和通訊負荷。下面介紹了命中和相交著色器的多種推測執行方案，這些方案可以從遍歷硬體在單一批次被分派，從而避免了多次遍歷和著色往返。可以使用實現這些技術的專用電路。

【0383】在需要從射線遍歷查詢執行多個命中或相交著色器的用例中，本發明的實施例是特別有益的，當沒有專用硬體支援來實現時，射線遍歷查詢將施加相當大的負荷。這些包含但不限於最近的k命中查詢(為k個最近的相交點啟動命中著色器)和多個可程式化的相交著色器。

【0384】這裡描述的技術可以被實現為對**圖40**中顯示的架構的擴展(並且關於**圖40-44**進行了描述)。特別是，本發明的當前實施例建立在所述具有增強以改善上述用例的效能的架構上。

【0385】混合射線追蹤遍歷架構的效能限制是從執行單元啟動遍歷查詢的負荷，以及從射線追蹤硬體調用可程式化著色器的負荷。在遍歷相同射線期間調用多個命中或相交著色器時，此負荷在可程式化核心4001與遍歷/相交單元4005之間產生「執行往返」。這也對需要提取來自各個著色器調用的SIMD/SIMT一致性的排序單元4008施加了額外的壓力。

【0386】射線追蹤的數個態樣需要可程式化控制，所述可程式化控制可以透過上面表A中列出的不同著色器類

型來表示(也就是說，主要、命中、任何命中、未命中、相交、遍歷和可調用)。每種類型可以有多個著色器。例如，每種材料可以具有不同的命中著色器。有些著色器類型在當前Microsoft®射線追蹤API中定義。

【0387】作為簡要回顧，遞歸射線追蹤是由API函數啟動的，所述API函數命令GPU啟動一組主著色器，這些著色器可以為主射線產生射線場景相交(在硬體及/或軟體中實現)。接著，這可以產生其它著色器，諸如遍歷、命中或未命中著色器。產生子著色器的著色器也可以從所述著色器接收返回值。可調用著色器是通用功能，可以直接由另一個著色器產生，也可以將值返回給調用著色器。

【0388】射線遍歷透過遍歷和相交定界體階層(BVH)中的節點來計算射線場景相交。最近的研究指出，使用更適合固定功能硬體的技術(諸如降低精確度的算術、BVH壓縮、每射線狀態機、專用相交管線和自定義快取)可以將計算射線場景相交的效率提高超過一個數量級。

【0389】**圖40**所示的架構包含這種系統，其中SIMD/SIMT核心/執行單元4001的陣列與固定功能射線追蹤/相交單元4005互動以執行可程式化射線追蹤。可程式化著色器映射到執行單元/核心4001上的SIMD/SIMT執行緒，其中SIMD/SIMT的利用率、執行和資料一致性對於最佳效能至關重要。射線查詢經常由於各種原因而破壞一致性，例如：

- 遍歷差異：在有利於非同步射線處理的射線中，

BVH遍歷的持續時間變化很大。

- 執行差異：從同一SIMD/SIMT執行緒的不同通道產生的射線可能致使不同的著色器調用。
- 資料存取差異：例如，命中不同表面的射線對不同的BVH節點和基元進行取樣，並且著色器存取不同的紋理。多種其它場景可能致使資料存取差異。

【0390】所述SIMD/SIMT核心/執行單元4001可以是本文所述的核心/執行單元的變體，其包含圖形核心415A-415B、著色器核心1355A-*N*、圖形核心3130、圖形執行單元608、執行單元852A -B或本文所述的任何其它核心/執行單元。SIMD/SIMT核心/執行單元4001也可以代替圖形核心415A-415B、著色器核心1355A-*N*、圖形核心3130、圖形執行單元608、執行單元852A -B或本文所述的任何其它核心/執行單元。因此，任何特徵的揭露與圖形核心415A-415B、著色器核心1355A-*N*、圖形核心3130、圖形執行單元608、執行單元852A -B或本文所述的任何其它核心/執行單元結合還揭露了與圖40的SIMD/SIMT核心/執行單元4001的對應組合，但不限於此。

【0391】固定功能射線追蹤/相交單元4005可以透過單獨地和亂序地處理每個射線來克服前兩個挑戰。然而，這會破壞SIMD/SIMT分組。因此，排序單元4008負責形成新的、一致的著色器調用的SIMD/SIMT分組，以將其再次分派給執行單元。

【0392】與直接在SIMD/SIMT處理器上的基於純軟體

的射線追蹤實現相比，很容易看到這種架構的好處。然而，在SIMD/SIMT核心/執行單元4001(有時在本文中簡稱為SIMD/SIMT處理器或核心/EU)與MIMD遍歷/相交單元4005之間存在與訊息傳遞相關的負荷。此外，排序單元4008可能無法從不連貫的著色器調用中提取完美的SIMD/SIMT利用率。

【0393】可以識別遍歷期間可能特別頻繁地調用著色器之用例。描述了用於混合MIMD射線追蹤處理器的增強，以顯著減少核心/EU 4001與遍歷/相交單元4005之間的通訊負荷。當找到k個最接近相交點並實現可程式化相交著色器時，這可能特別有益。然而，請注意，本文所述的技術不限於任何特定的處理方案。

【0394】下面提供了核心/EU 4001與固定功能遍歷/相交單元4005之間的射線追蹤上下文切換的高階成本的摘要。每當在單射線遍歷期間需要著色器調用時，大多數效能負荷是由這兩個上下文切換致使的。

【0395】發射射線的每個SIMD/SIMT通道向與遍歷的BVH相關的遍歷/相交單元4005產生產生訊息。資料(射線遍歷上下文)經由產生訊息和(快取的)記憶體被中繼到遍歷/相交單元4005。當遍歷/相交單元4005準備向產生訊息分配新的硬體執行緒時，它載入遍歷狀態並在BVH上執行遍歷。在BVH上的第一個遍歷步驟之前，還需要執行設置費用。

【0396】**圖45**顯示可程式化射線追蹤管線的操作流

程。包含遍歷4502和相交點4503的著色元件可以在固定功能電路中實現，而其餘元件可以由可程式化核心/執行單元實現。

【0397】主射線著色器4501在4502將工作發送到遍歷電路，遍歷電路透過BVH(或其它加速結構)遍歷當前射線。當到達葉節點時，遍歷電路在4503處調用相交電路，其在識別到射線-三角形相交時，在4504處調用任何命中著色器(其可將結果提供回所指示的遍歷電路)。

【0398】備選地，遍歷可以在到達葉節點之前終止，並且在4507調用最接近的命中著色器(如果記錄了命中)或在4506調用未命中著色器(在未命中的事件中)。

【0399】如4505所示，如果遍歷電路到達自定義基元葉節點，則可以調用相交著色器。自定義基元可以是任何非三角形基元，諸如多邊形或多面體(例如，四面體、立體像素、六面體、楔形、金字塔或其它「非結構化」容積)。相交著色器4505將射線和自定義基元之間的任何相交識別給實現任何命中處理的任何命中著色器4504。

【0400】當硬體遍歷4502達到可程式化階段時，遍歷/相交單元4005可以產生著色器分派訊息給相關的著色器4505-4507，所述著色器對應於用於執行著色器的執行單元的單一SIMD通道。由於分派以射線的任意順序發生，並且它們在所調用的程式中是發散的，因此排序單元4008可以累積多個分派調用以提取相干的SIMD批次。遍歷/相交單元4005可以將更新後的遍歷狀態和可選的著色器參數

寫入記憶體2511。

【0401】在k個最接近相交點問題中，對前k個相交點執行最接近的命中著色器4507。根據傳統方式，這將意味著在找到最接近的相交時結束射線遍歷、調用命中著色器，並從命中著色器中產生新射線以找到下一個最接近的相交(具有射線原點偏移，因此相同的相交不會再次發生)。顯而易見，這種實現需要為單一射線產生k射線。另一實現方式是使用插入排序操作，對任何命中(any-hit)著色器4504進行操作，對所有相交點調用並維護最接近相交點的全域列表。這種方法的主要問題是，任何命中著色器調用沒有上限。

【0402】如所提及的，可以在非三角形(自定義)基元上調用相交著色器4505。取決於相交測試的結果和遍歷狀態(待處理節點和基元相交)，在執行相交著色器4505之後，相同射線的遍歷可能會繼續。因此，找到最接近的命中點可能需要多次往返執行單元。

【0403】透過改變遍歷硬體和著色器排程模型，也可以將重點放在減少相交著色器4505和命中著色器4504、4507的SIMD-MIMD上下文切換。首先，射線遍歷電路4005透過累積多個潛在的調用並以更大的批次分派它們來推遲著色器調用。此外，某些變成不必要的調用可能會在此階段被剔除。再者，著色器排程器4007可以將來自相同遍歷上下文的多個著色器調用聚集到單一SIMD批次中，這致使單一射線產生訊息。在一個範例性實現中，遍歷硬

體4005中止遍歷執行緒並等待多個著色器調用的結果。此操作模式在此稱為「推測性」著色器執行，因為它允許分派多個著色器，使用順序調用時可能不會調用其中的一些。

【0404】**圖46A**顯示遍歷操作在子樹中遇到多個自定義基元4650的範例，而**圖46B**顯示如何使用三個交集分派週期C1-C3解決此問題。特別是，排程器4007可能需要三個週期來將工作提交給SIMD處理器4001，並且遍歷電路系統4005需要三個週期來將結果提供給排序單元4008。遍歷電路系統4005所需的遍歷狀態4601可以被儲存在諸如區域快取(例如，L1快取及/或L2快取)的記憶體中。

A.推遲射線追蹤著色器調用

【0405】還可以修改管理硬體遍歷狀態4601以允許在列表中累積多個潛在的相交或命中調用的方式。在遍歷期間的給定時間，可以使用列表中的每個條目來產生著色器調用。例如，可以在遍歷硬體4005上及/或在記憶體中的遍歷狀態4601中累積k個最接近的交點，並且如果遍歷完成，則可以為每個元件調用命中著色器。對於命中著色器，BVH中的子樹可能會積累多個潛在的交集。

【0406】對於最接近k的用例，所述方法的好處在於，代替在SIMD核心/EU 4001上進行k-1次往返和在k-1條新的射線產生訊息上進行往返之外，所有命中著色器都是在遍歷電路4005上的單一遍歷操作期間從同一遍歷執行緒

調用的。潛在實現的挑戰在於，保證命中著色器的執行順序並不容易(標準的「往返」方法保證最接近的交點的命中著色器最先執行等)。這可以透過命中著色器的同步或排序的放鬆來解決。

【0407】對於相交著色器用例，遍歷電路4005事先不知道給定的著色器是否將返回正相交測試。然而，可以推測性地執行多個相交著色器，並且如果至少有一個返回正面命中結果，則將其合併到全域最接近的命中之中。特定實現需要找到最佳數量的推遲交集測試以減少分派調用的數量，但要避免調用過多的冗餘相交著色器。

B.遍歷電路的匯總著色器調用

【0408】當從遍歷電路4005上產生的同一射線分派多個著色器時，可以在射線遍歷演算法的流程中建立分支。這對於相交著色器可能是會有問題的，因為BVH遍歷的其餘部分取決於所有分派的相交測試的結果。這意味著同步操作必須等待著色器調用的結果，這在非同步硬體上可能是一個挑戰。

【0409】合併著色器調用的結果的兩點可以是：SIMD處理器4001和遍歷電路4005。關於SIMD處理器4001，多個著色器可以使用標準程式化模型來同步和聚合其結果。一種相對簡單的方法是使用全域原子並在記憶體中的共享資料結構中聚合結果，可以在所述記憶體中儲存多個著色器的相交結果。接著，最後的著色器可以解析資

料結構並調用回遍歷電路4005以繼續遍歷。

【0410】還可以實現更有效的方法，所述方法將對多個著色器調用的執行限制到SIMD處理器4001上相同SIMD執行緒的通道。接著，使用SIMD/SIMT縮減操作來區域減少相交測試(而不是依靠全域原子)。所述實現可以取決於排序單元4008內的新電路，以使一小批次的著色器調用停留在同一SIMD批次中。

【0411】遍歷執行緒的執行可以進一步在遍歷電路4005上暫停。使用傳統執行模型，當在遍歷期間分派著色器時，遍歷執行緒被終止，並且射線遍歷狀態被保存到記憶體中以允許在執行單元4001處理著色器的同時執行其它射線產生命令。如果僅將遍歷執行緒暫停，則不需要儲存遍歷狀態，並且可以分別等待每個著色器結果。所述實現可以包含避免死鎖(deadlocks)並提供足夠的硬體利用率的電路。

【0412】**圖47-48**顯示推遲模型的範例，所述推遲模型在具有三個著色器4701的SIMD核心/執行單元4001上調用單一著色器調用。當保留時，將在同一SIMD/SIMT分組內評估所有相交測試。因此，也可以在可程式化核心/執行單元4001上計算最近的交點。

【0413】如所提及的，著色器聚合及/或推遲的全部或一部分可以由遍歷/相交電路4005及/或核心/EU排程器4007執行。**圖47**顯示在排程器4007內的著色器推遲/聚合器電路4706如何可以將與特定SIMD/SIMT執行緒/通道相

關的著色器的排程推遲到發生指定的觸發事件為止。在檢測到觸發事件時，排程器4007將單一SIMD/SIMT批次中的多個聚合著色器分派到核心/EU 4001。

【0414】**圖48**顯示遍歷/相交電路4005內的著色器推遲/聚合器電路4805如何可以將與特定SIMD執行緒/通道關聯的著色器的排程推遲到指定的觸發事件發生之前。一旦檢測到觸發事件，遍歷/相交電路4005就以單一SIMD/SIMT批次將聚合的著色器提交給排序單元4008。

【0415】然而，注意，著色器推遲和聚合技術可以在諸如排序單元4008的各種其它部件內實現，或者可以分佈在多個部件上。例如，遍歷/相交電路4005可以執行第一組著色器聚合操作，並且排程器4007可以執行第二組著色器聚合操作，以確保用於SIMD執行緒的著色器在核心/EU 4001上被有效地排程。

【0416】致使聚集的著色器被分派到核心/EU的「觸發事件」可以是諸如特定數量的累積著色器或與特定執行緒相關的最小等待時間的處理事件。替代地或額外地，觸發事件可以是時間事件，諸如從第一著色器的推遲起的某個持續時間或特定數量的處理器週期。排程器4007還可以評估諸如核心/EU 4001和遍歷/相交單元4005上的當前工作負載之類的其它變量，以確定何時分派著色器的SIMD/SIMT批次。

【0417】基於所使用的特定系統架構和應用的需求，可以使用以上方法的不同組合來實現本發明的不同實施

例。

射線追蹤指令

【0418】下述射線追蹤指令包含在支援CPU 3199及/或GPU 3105的指令集架構(ISA)中。如果由CPU執行，則單一指令多重資料(SIMD)指令可以利用向量/緊縮源和目的地暫存器來執行所描述的操作，並且可以由CPU核心解碼和執行。如果由GPU 3105執行，則指令可以由圖形核心3130執行。例如，上述執行單元(EU)4001中的任何一個可以執行指令。替代地或額外地，指令可以由射線追蹤核心3150及/或張量核心張量核心3140上的執行電路執行。

【0419】**圖49**顯示用於執行下面描述的射線追蹤指令的架構。所顯示的架構可整合在可包含在不同處理器架構中的上文所描述的核心3130、3140、3150中的一或多者上(例如，參見**圖31**和相關的文字)。

【0420】在操作中，指令提取單元4903從記憶體3198提取射線追蹤指令4900，並且解碼器4995對所述指令進行解碼。在一種實現中，解碼器4995對指令進行解碼以產生可執行操作(例如，微編碼核心中的微操作或微指令)。可替代地，射線追蹤指令4900中的一些或全部可以在不解碼的情況下被執行，從而不需要解碼器4904。

【0421】在任一實現中，排程器/分派器4905跨一組功能單元(FU)4910-4912排程和分派指令(或操作)。所說明的實現包含用於執行對儲存在向量暫存器4915中的多個緊

縮資料元件上同時操作的單一指令多重資料(SIMD)指令的向量FU 4910，以及用於對儲存在一或多個純量暫存器4916中的純量值進行操作的純量FU 4911。可選的射線追蹤FU 4912可以對向量暫存器4915中儲存的緊縮資料值及/或純量暫存器4916中儲存的純量值進行操作。在沒有專用FU 4912的實現中，向量FU 4910以及可能的純量FU 4911可執行下面描述的射線追蹤指令。

【0422】各種FU 4910-4912存取執行來自向量暫存器4915、純量暫存器4916及/或區域快取子系統4908(例如，L1快取)的射線追蹤指令4900所需的射線追蹤資料4902(例如，遍歷/相交資料)。FU 4910-4912還可以經由載入和儲存操作來執行對記憶體3198的存取，並且快取子系統4908可以獨立地操作以區域性地快取資料。

【0423】儘管射線追蹤指令可以用於增加射線遍歷/相交和BVH構建的效能，但是它們也可以適用於其它領域，諸如高效能計算(HPC)和通用GPU(GPGPU)的實現。

【0424】在以下描述中，用語雙字組有時縮寫為*dw*，而無符號位元組縮寫為*ub*。此外，下面參照的來源暫存器和目的地暫存器(例如src0、src1、dest等)可以參照向量暫存器4915，或者在某些情況下可以參照向量暫存器4915和純量暫存器4916的組合。通常，如果指令使用的來源或目的地值包含緊縮資料元件(例如，來源或目的地儲存N個資料元件的地方)，則使用向量暫存器4915。其它值可以使用純量暫存器4916或向量暫存器4915。

去量化

【0425】去量化指令的一個範例將先前量化的值「去量化」。例如，在射線追蹤實現中，可以對某些BVH子樹進行量化以減少儲存和頻寬需求。去量化指令可以採用去量化dest src0 src1 src2的形式，其中來源暫存器src0儲存N個無符號位元組、來源暫存器src1儲存1個無符號位元組、來源暫存器src2儲存1個浮點值，而目的地暫存器dest儲存N個浮點值。所有這些暫存器都可以是向量暫存器4915。替代地，src0和dest可以是向量暫存器4915，而src1和src2可以是純量暫存器4916。

【0426】以下程式碼序列定義了去量化指令的一種特定實現：

```
for (int i = 0; i < SIMD_WIDTH) {
    if (execMask[i]) {
        dst[i] = src2[i] + ldexp(convert_to_float(src0[i]),src1);
    }
}
```

在此範例中，ldexp將雙精確度浮點值乘以指定的2的整數冪次(即ldexp(x, exp) = x * 2^{exp})。在上面的程式碼中，如果將與當前SIMD資料元件(execMask [i])關聯的執行遮罩值設置為1，則src0中位置i的SIMD資料元件將轉換為浮點值並乘以src1中的值的整數冪次($2^{src1value}$)，並將此值添加到src2中的對應SIMD資料元件。

選擇性最小或最大

【0427】如位元遮罩中的位元所指示的，選擇性的最小或最大指令可以對每個通道執行最小或最大操作(也就是說，返回一組值中的最小或最大值)。所述位元遮罩可以利用向量暫存器4915、純量暫存器4916或單獨的一組遮罩暫存器(未顯示)。以下程式碼序列定義了最小/最大指令的一種特定實現：sel_min_max dest src0 src1 src2，其中src0儲存N個雙字組、src1儲存N個雙字組、src2儲存一個雙字組，而目的暫存器儲存N個雙字組。

【0428】以下程式碼序列定義了選擇性最小/最大指令的一種特定實現：

```
for (int i = 0; i < SIMD_WIDTH) {
    if (execMask[i]) {
    dst[i] = (1 << i) & src2 ? min(src0[i],src1[i]) :
    max(src0[i],src1[i]);
    }
}
```

在此範例中，(1 < < i)& src2的值(i左移1與src2聯集)用於選擇src0和src1中的第i個資料元件的最小值或src0和src1中的第i個資料元件的最大值。僅當與當前SIMD資料元件(execMask [i])關聯的執行遮罩值設置為1時，才對第i個資料元件執行所述操作。

混洗索引指令

【0429】混洗索引指令可以將任何一組輸入通道複製到輸出通道。對於32的SIMD寬度，可用較低的產出量執行所述指令。所述指令採用以下形式：shuffle_index dest src0src1<可選旗標>，其中src0儲存N個雙字組、src1儲存N個無符號位元組(即索引值)，而dest儲存N個雙字組。

【0430】以下程式碼序列定義了混洗索引指令的一種特定實現：

```
for (int i = 0; i < SIMD_WIDTH) {
  uint8_t srcLane = src1.index[i];

  if (execMask[i]) {
    bool invalidLane = srcLane < 0 || srcLane >= SIMD_WIDTH
|| !execMask[srcLaneMod];

    if (FLAG) {
      invalidLane |= flag[srcLaneMod];
    }

    if (invalidLane) {
      dst[i] = src0[i];
    }
    else {
      dst[i] = src0[srcLane];
    }
  }
}
```

【0431】在以上程式碼中，src1中的索引識別當前通道。如果執行遮罩中的第i個值設置為1，則將執行檢查以確保來源通道在0到SIMD寬度的範圍內。如果是，則設置旗標(srcLaneMod)，並將目的地的資料元件i設置為等於src0的資料元件i。如果通道在範圍內(即有效)，則將src1的索引值(srcLane0)用作src0的索引(dst[i] = src0[srcLane])。

立即混洗Up/Dn/XOR指令

【0432】立即混洗指令可以基於指令的立即數來混洗輸入資料元件/通道。立即數可以基於立即數的值指定將輸入通道移位1、2、4、8或16個位置。可選地，可以將額外的純量來源暫存器指定為填充值。當來源通道索引無效時，填充值(如果提供)將儲存到目的地中的資料元件位置。如果未提供填充值，則資料元件位置設置為全0。

【0433】旗標暫存器可以用作來源遮罩。如果來源通道的旗標位元設置為1，則可以將來源通道標記為無效，接著繼續執行指令。

【0434】以下是立即混洗指令的不同實現的範例：

```
shuffle_<up/dn/xor>_<1/2/4/8/16> dest src0 <optional src1>
<optional flag>
shuffle_<up/dn/xor>_<1/2/4/8/16> dest src0 <optional src1>
<optional flag>
```

在此實現中，src0儲存N個雙字組、src1儲存填充值的一個雙字組(如果存在)，而dest儲存包含結果的N個雙字組。

【0435】以下程式碼序列定義了立即混洗指令的一種特定實現：

```
for (int i = 0; i < SIMD_WIDTH) {
  int8_t srcLane;
  switch(SHUFFLE_TYPE) {
  case UP:
    srcLane = i - SHIFT;
  case DN:
    srcLane = i + SHIFT;
  case XOR:
    srcLane = i ^ SHIFT;
  }

 if (execMask[i]) {
   bool invalidLane = srcLane < 0 || srcLane >= SIMD_WIDTH
|| !execMask[srcLane];

   if (FLAG) {
     invalidLane |= flag[srcLane];
   }

   if (invalidLane) {
     if (SRC1)
       dst[i] = src1;
     else
       dst[i] = 0;
   }
```

```
    else {
      dst[i] = src0[srcLane];
    }
  }
}
```

【0436】在此，基於立即數的值，輸入資料元件/通道被移位1、2、4、8或16個位置。暫存器src1是額外的純量來源暫存器，其用作填充值，當來源通道索引無效時，所述填充值被儲存到目的地中的資料元件位置。如果未提供填充值且來源通道索引無效，則目的地中的資料元件位置將設置為0。旗標暫存器(FLAG)用作來源遮罩。如果將來源通道的旗標位元設置為1，則將來源通道標記為無效，並且指令如上所述進行。

間接混洗Up/Dn/XOR指令

【0437】間接混洗指令具有控制從來源通道到目的地通道的映射的來源運算元(src1)。間接混洗指令可以採用以下形式：

```
shuffle_<up/dn/xor> dest src0 src1 <optional flag>
```

其中src0儲存N個雙字組、src1儲存1個雙字組，而dest儲存N個雙字組。

【0438】以下程式碼序列定義了立即混洗指令的一種特定實現：

```
for (int i = 0; i < SIMD_WIDTH) {
 int8_t srcLane;
 switch(SHUFFLE_TYPE) {
 case UP:
  srcLane = i - src1;
 case DN:
  srcLane = i + src1;
 case XOR:
  srcLane = i ^ src1;
 }

 if (execMask[i]) {
  bool invalidLane = srcLane < 0 || srcLane >= SIMD_WIDTH
|| !execMask[srcLane];

  if (FLAG) {
   invalidLane |= flag[srcLane];
  }

  if (invalidLane) {
   dst[i] = 0;

  }
  else {
   dst[i] = src0[srcLane];
  }
 }
}
```

【0439】因此，間接混洗指令以與上述立即混洗指令相似的方式操作，但是來源通道到目的地通道的映射由來源暫存器src1而不是立即數控制。

相交通道最小/最大指令

【0440】對於浮點和整數資料類型，可以支援相交通道最小/最大指令。相交通道最小指令的形式可以是lane_min dest src0，相交通道最大指令的形式可以是lane_max dest src0，其中src0儲存N個雙字組，而dest儲存1個雙字組。

【0441】作為範例，以下程式碼序列定義了相交通道最小值的一種特定實現：

```
dst = src[0];

for (int i = 1; i < SIMD_WIDTH) {
 if (execMask[i]) {
  dst = min(dst, src[i]);
 }
}
```

在此範例中，將來源暫存器的資料元件位置i中的雙字組值與目的地暫存器中的資料元件進行比較，並將這兩個值中的最小值複製到目的地暫存器中。相交通道最大指令以實質上相同的方式操作，唯一的區別是選擇了位置i和目的地值中的資料元件的最大值。

相交通道最小/最大索引指令

【0442】相交通道最小索引指令可以採取lane_min_index dest src0的形式，而相交通道最大索引指令可以採取lane_max_index dest src0的形式，其中src0儲存N個雙字組，而dest儲存1個雙字組。

【0443】作為範例，以下程式碼序列定義了相交通道最小索引指令的一種特定實現：

```
dst_index = 0;
tmp = src[0]

for (int i = 1; i < SIMD_WIDTH) {

 if (src[i] < tmp && execMask[i])
 {
   tmp = src[i];
   dst_index = i;
 }
}
```

在此範例中，目的地索引從0遞增到SIMD寬度，跨越目的地暫存器。如果設置了執行遮罩位元，則將來源暫存器中位置i的資料元件複製到臨時儲存位置(tmp)，將目的地索引設置為資料元件位置i。

相交通道排序網路指令

【0444】跨通道排序網路指令可以使用N寬(穩定)的

排序網路以升序(sortnet_min)或降序(sortnet_max)對所有N個輸入元件進行排序。指令的最小/最大版本可以分別採用sortnet_min dest src0和sortnet_max dest src0的形式。在一種實現中，src0和dest儲存N個雙字組。最小/最大排序是對src0的N個雙字組執行的，升序元件(用於min)或降序元件(用於max)以其各自的排序順序儲存在dest中。定義所述指令的程式碼序列的一個範例是：

dst = apply_N_wide_sorting_network_min/max(src0)。

跨通道排序網路索引指令

【0445】跨通道排序網路索引指令可以使用N寬(穩定)的排序網路對所有N個輸入元件進行排序，但是以升序(sortnet_min)或降序(sortnet_max)返回置換索引。指令的最小/最大版本可以採用sortnet_min_index dest src0和sortnet_max_index dest src0的形式，其中src0和dest分別儲存N個雙字組。定義指令的程式碼序列的一個範例是dst = apply_N_wide_sorting_network_min/max_index(src0)。

【0446】在**圖50**中顯示用於執行以上任何指令的方法。所述方法可以在上述特定處理器架構上實現，但是不限於任何特定處理器或系統架構。

【0447】在5001處，在處理器核心上執行主圖形執行緒的指令。這可以包含例如上述任何核心(例如，圖形核心3130)。當在5002處確定在主圖形執行緒內達到射線追蹤工作時，將射線追蹤指令卸載到射線追蹤執行電路，所

述射線追蹤執行電路可以是諸如以上關於**圖49**所描述的功能單元(FU)的形式或可以在關於**圖31**所描述的專用射線追蹤核心3150中。

【0448】在5003處，從記憶體中獲取解碼的射線追蹤指令，並且在5005，將指令解碼成可執行的操作(例如，在需要解碼器的實施例中)。在5004處，對射線追蹤指令進行排程和分派，以由射線追蹤電路執行。在5005處，射線追蹤電路執行射線追蹤指令。例如，指令可以被分派並且在上述FU(例如，向量FU 4910、射線追蹤FU 4912等)及/或圖形核心3130或射線追蹤核心3150上執行。

【0449】當針對射線追蹤指令的執行完成時，結果被儲存在5006處(例如，被儲存回到記憶體3198)，並且在5007處通知主要圖形執行緒。在5008處，在主執行緒的上下文中處理射線追蹤結果(例如，從記憶體讀取並整合到圖形繪製結果中)。

【0450】在實施例中，用語「引擎」或「模組」或「邏輯」可以是指(或其部分、或包括)特殊應用積體電路(ASIC)、電子電路、處理器(共享的、專用的或群組的)及/或執行一或多個軟體或韌體程式的記憶體(共享的、專用的或群組的)、組合邏輯電路及/或提供所描述功能的其它合適部件。在實施例中，引擎、模組或邏輯可以用韌體、硬體、軟體或韌體、硬體和軟體的任何組合來實現。

非同步射線追蹤的設備和方法

【0451】本發明的實施例包含用以執行射線追蹤的固定功能加速電路和通用處理電路的組合。例如，與定界體階層(BVH)的射線遍歷和相交測試有關的某些操作可以由固定功能加速電路執行，而複數個執行電路執行各種形式的射線追蹤著色器(例如，任何命中著色器、相交著色器、未命中著色器等)。一個實施例包含雙高頻寬儲存庫，其包含用於儲存射線的複數個條目和用於儲存BVH節點的對應雙堆疊。在此實施例中，遍歷電路在雙射線庫和堆疊之間交替以在每個時脈週期處理射線。此外，一個實施例包含優先順序選擇電路/邏輯，其分辨內部節點、非內部節點和基元，並且使用此資訊來智慧地排定處理所述BVH節點和由BVH節點定界的基元的優先順序。

【0452】一個特定的實施例減少了遍歷操作期間使用短堆疊來儲存有限數量的BVH節點的遍歷所需的高速記憶體。此實施例包含堆疊管理電路/邏輯，以有效地將條目推入短堆疊和並從短堆疊中彈出，以確保所需的BVH節點可用。此外，透過對追蹤資料結構執行更新來追蹤遍歷操作。當遍歷電路/邏輯暫停時，可以查詢追蹤資料結構，以在BVH中離開的同一位置開始遍歷操作。並且執行在資料結構追蹤中維護的追蹤資料，以便遍歷電路/邏輯可以重啟。

【0453】**圖51**顯示一個實施例，其包含：著色器執行電路4000，其用於執行著色器程式碼並處理關聯的射線追蹤資料4902(例如，BVH節點資料和射線資料)；射線追蹤

加速電路5110，其用於執行遍歷和相交操作；以及記憶體3198，其用於儲存由RT加速電路5110和著色器執行電路4000處理的程式碼和相關資料。

【0454】在一實施例中，著色器執行電路4000包含複數個核心/執行單元4001，其執行著色器程式碼，以執行各種形式的資料-平行操作。例如，在一實施例中，核心/執行單元4001可以跨多個通道執行單一指令，其中所述指令的每個實例對儲存在不同通道中的資料進行操作。例如，在SIMT實現中，指令的每個範例都與不同的執行緒關聯。在執行期間，L1快取儲存某些射線追蹤資料以進行有效存取(例如，最近或經常存取的資料)。

【0455】一組主要射線可以被分派到排程器4007，其將工作排程到由核心/EU 4001執行的著色器。核心/EU 4001可以是射線追蹤核心3150、圖形核心3130、CPU核心3199或能夠執行著色器程式碼的其它類型電路。一或多個主要射線著色器5101處理主要射線並且產生將由射線追蹤加速電路5110及/或核心/EU 4001執行的額外工作(例如，將由一或多個子著色器執行)。由主要射線著色器5101產生的新工作或由核心/EU 4001執行的其它著色器可以被分配給排序電路4008，其如本文描述將射線排序成組或二元子(例如，將射線分組成具有類似的特性)。接著，排程器4007將新工作排程在核心/EU 4001上。

【0456】可被執行的其它著色器包含任何命中著色器4514和最接近命中著色器4507，其如上述處理命中結果

(例如，對於給定的射線分別識別任何命中或最接近命中)。未命中著色器4506處理射線未命中(例如，在射線不與節點/基元相交的地方)。如所提到的，各種著色器可以使用著色器記錄被參照，所述著色器記錄可以包含一或多個指標、特定於供應商的元資料和全域參數。在一實施例中，著色器記錄由著色器記錄識別符(SRI)識別。在一實施例中，著色器的每個執行實例與調用堆疊5203相關，所述調用堆疊5203儲存在父著色器和子著色器之間傳遞的參數。調用堆疊5121還可以儲存對在調用返回時執行的繼續功能的參照。

【0457】射線遍歷電路5102使每個射線遍歷穿過BVH的節點，向下工作BVH的階層(例如，穿過父節點、子節點和葉節點)，以識別射線所遍歷的節點/基元。射線BVH相交電路5103執行射線的相交測試，確定基元上的命中點，並回應於命中而產生結果。遍歷電路5102和相交電路5103可以從一或多個調用堆疊5121檢索工作。在射線追蹤加速電路5110中，調用堆疊5121和相關的射線追蹤資料4902可以儲存在區域射線追蹤快取(RTC)5107或其它區域儲存裝置內，以供遍歷電路5102和相交電路5103有效存取。下面描述的一個特定實施例包含高頻寬射線庫(例如，參見圖52A)。

【0458】射線追蹤加速電路5110可以是本文所述的各種遍歷/相交電路的變體，其包含射線BVH遍歷/相交電路4005、遍歷電路4502和相交電路4503和射線追蹤核心

3150。所述射線追蹤加速電路5110也可以代替射線BVH遍歷/相交電路4005、遍歷電路4502和相交電路4503和射線追蹤核心3150或任何其它用於處理BVH堆疊及/或執行遍歷/相交的電路/邏輯使用。因此，任何特徵的揭露組合本文中描述的射線BVH遍歷/相交電路4005、遍歷電路4502和相交電路4503和射線追蹤核心3150也揭露了與射線追蹤加速電路5110的對應組合，但不限於此。

【0459】參考**圖52**，射線遍歷電路5102的一個實施例分別包含第一射線儲存庫5201和第二射線儲存庫5202，其中每個庫包含從記憶體載入的複數個用於儲存對應的複數個入射射線5206的條目。對應的第一和第二堆疊5203和5204分別包含從記憶體讀取並且區域地儲存以供處理的選定BVH節點資料。如本文所述，在一實施例中，堆疊5203-5204是「短」堆疊，其包含用於儲存BVH節點資料的有限數量的條目(例如，在一實施例中為六個條目)。同時從射線庫5201-5202單獨顯示，堆疊5203-5204也可以維持在對應射線庫5201-5202內。替代地，堆疊5203-5204可以被儲存在單獨的區域記憶體或快取中。

【0460】遍歷處理電路5210的一個實施例在選擇下一個射線和要處理的節點時，在兩個庫5201-5202和堆疊5203-5204之間交替(例如，以乒乓方式)。例如，遍歷處理電路5210可以在每個時脈週期從交替的射線庫/堆疊選擇新的射線/BVH節點，從而確保高效率的操作。然而，應當注意，此特定佈置對於遵守本發明的基本原理不是必需

的。

【0461】在一實施例中，射線分配器5205基於一組庫分配計數器5220的當前相對值來平衡入射射線5206分別進入第一和第二記憶體庫5201-5202的條目。在一實施例中，庫分配計數器5220維護第一和第二記憶體庫5201-5202中之各者中未遍歷的射線的數量的計數。例如，當射線分配器5205向第一庫5201添加新射線時，第一庫分配計數器可以遞增，並且當從第一庫5201處理了射線時，第一庫分配計數器可以遞減。類似地，當射線分配器5205向第二庫5201添加新射線時，第二庫分配計數器可以遞增，並且當從第二庫5201處理了射線時，第二庫分配計數器可以遞減。

【0462】在一實施例中，射線分配器5205將當前射線分配給與較小的計數器值相關的庫。如果兩個計數器相等，則射線分配器5205可以選擇任一庫或者可以選擇與上一次計數器相等時所選擇的庫不同的庫。在一實施例中，每條射線被儲存在庫5201-5202之一的一個條目中，並且每個庫包含32個條目，以供儲存多達32個射線。然而，本發明的基本原理不限於這些細節。

用於非局部平均濾波的設備和方法

【0463】「非局部平均」是用於執行降噪的影像處理技術。與「局部」平均濾波器透過確定目標像素周圍的像素區塊的平均值來平滑影像的方法不同，非局部平均濾波

器會確定影像中所有像素的平均值，並按這些像素與目標像素的相似程度加權，與局部平均演算法相比，可提高濾波後清晰度，並減少影像細節的損失。

【0464】在現有的實現中，非局部平均計算是在軟體中執行的。因此，這些實現不適合即時繪製環境。

【0465】本發明的一個實施例使用媒體處理器管線中的現有運動估計硬體區塊來執行非局部平均濾波。因為現有的運動估計電路被重新使用，所以僅需要少量的額外電路。使用這種硬體，與目前的軟體實現相比，本發明的實施例可以更加有效率地進行非局部平均濾波。

【0466】此外，因為運動估計電路固有地配置成評估跨越影像訊框的運動，所以，本發明的一個實施例使用此特徵來執行訊框間非局部平均濾波(而不是將操作限制在當前訊框中)。因此，在一種實現中，可以在影像訊框內及/或跨影像訊框間使用非局部平均濾波來執行有效且準確的時間降噪。

【0467】本發明的一個實施例包含新的介面，其包含硬體及/或軟體元件(例如，程式化介面)，以使媒體處理器的運動估計方塊為其它處理器元件和應用程式可存取。作為範例而非限制，這些可以包含3D圖形管線內的深度估計操作、視訊影像的去交錯，以及用於虛擬實境實現的視圖內插。

【0468】本文描述的許多處理器架構包含具有用於編碼和解碼視訊內容(例如，諸如H.264視訊)的專用電路的

媒體處理器(有時稱為「媒體引擎」或「媒體管線」)。這些媒體處理器包含媒體管線234(圖2B)、媒體管線316(圖3A、圖4)、媒體引擎837(圖8)和媒體處理引擎3197(圖31)。每個這種媒體處理器包含執行用於視訊處理的運動估計的電路。在一個實施例中，代替實現用於降噪的機器學習方法，本發明的一個實施例使用媒體處理器內的運動估計硬體來執行非局部平均濾波。

【0469】**圖53**顯示了本發明的一個實施例，其中媒體處理引擎3197的運動估計電路5320在繪製的影像訊框5330上執行非局部平均濾波以產生降噪訊框5340。特別是，在一個實施例中，媒體API 5302(也就是說，能夠由媒體處理引擎3197執行的命令/指令的集合)補充有非局部平均命令5305，以實現本文描述的技術。非局部平均命令5305存取運動估計電路5320的各種架構特徵，包含由運動估計電路5320提供的額外功率，以執行跨多個影像訊框5330的非局部平均濾波，而不是如當前的實現般僅在當前訊框內進行濾波。例如，當執行視訊編碼時，運動估計電路5320的主要功能之一是識別跨連續訊框的像素區塊(例如，巨集區塊、切片等)的運動。因此，不是透過確定單一影像訊框中所有像素的平均(基於與目標像素的相似度進行加權)對目標像素執行非局部平均濾波，而是使用運動估計電路5320確定分佈在多個影像訊框中的所有像素的平均。

【0470】可以在像素或像素區塊(例如，巨集區塊)的程度上執行非局部平均濾波。像素區塊可以包含由運動估

計電路5320所支援的任何排列的像素，包含在每個維度上具有相同數量的像素(16×16、8×8或4×4)區塊以及在每個維度上具有不同數量的像素(例如，16×8、16×4、8×4等)的區塊。此外，可以用不對稱排列的像素區塊來編碼影像訊框，使得影像訊框的中心區域以較高的精確度編碼(例如，使用4×4像素區塊)，而影像訊框的外圍以較低的精確度編碼(例如，使用16×16像素區塊)。

【0471】在一個實施例中，無論所使用的編碼類型，非局部平均命令5305控制運動估計電路5320，以將實際像素資料與影像訊框5330中的「雜訊」區別，以產生降噪影像訊框5340。具體而言，在一個實現中，使用運動估計電路5320，能在影像訊框內及/或跨影像訊框，使用非局部平均濾波，進行有效且精確的時間降噪。

【0472】在操作中，命令串流器5310將影像繪製命令串流到射線追蹤電路5110(也就是說，用於遍歷和相交)和著色器執行電路4000(也就是說，用於著色器的執行)，以使用本文所述的各種技術產生影像訊框5330。隨後或同時，命令串流器5310將非局部平均命令5305串流到媒體處理引擎3197，媒體處理引擎3197利用運動估計電路5320執行命令，以評估連續影像訊框5330以執行降噪並產生降噪影像訊框5340。

【0473】媒體API 5302的一個實施例配備有額外的命令，以使運動估計方塊5320對於其它應用程式為可存取。透過舉例的方式，而不是限制的方式，這些可以包含3D圖

形管線內的深度估計操作、視訊影像的去交錯，及/或視圖內插(例如，對於虛擬實境實現)。

【0474】在**圖54**中顯示了根據一個實施例的方法。所述方法可以在本文描述的架構的上下文中實現，但是不限於任何特定的處理器或系統架構。

【0475】在5401處，命令串流器將命令串流到射線追蹤電路和著色器執行電路。在5402處，射線追蹤電路和著色器執行電路執行命令以繪製影像訊框。在5403處，命令串流器將非局部平均(NLM)命令串流到媒體處理電路，而在5404處，使用運動估計電路執行命令，以透過評估訊框間像素區塊來執行降噪。

【0476】

範例

【0477】以下是本發明的不同實施例的範例實現。

【0478】範例1. 一種處理器，包含：射線追蹤電路，其用以執行第一組的一或多個命令，以使射線穿越通過邊界體積階層(BVH)穿越射線，以識別與所述射線相交的BVH節點及/或基元；著色器執行電路，其用以回應於第二組的一或多個命令而執行一或多個著色器，以基於與所述射線相交的所述BVH節點及/或基元來繪製影像訊框序列；以及媒體處理器，其包含運動估計電路，所述運動估計電路用以執行第三組的一或多個命令，以基於跨所述影像訊框序列收集的平均像素值執行非局部平均濾波以從所述影像訊框序列中移除雜訊。

【0479】範例2. 如請求項1的處理器，還包含：命令串流器，其用於將所述第一組命令串流到所述射線追蹤電路，將所述第二組命令串流到所述著色器執行電路，並且將所述第三組命令串流到所述媒體處理器。

【0480】範例3. 如範例2的處理器，其中所述第三組命令包含對與所述媒體處理器相關的應用程式化介面(API)的擴展。

【0481】範例4. 如範例1的處理器，其中對目標像素的所述非局部平均濾波包含確定與跨所述影像訊框序列的所有像素相關的平均值，所述平均值被以與所述目標像素的相似性加權。

【0482】範例5. 如範例1的處理器，其中所述著色器執行電路包含複數個執行單元(EU)，以執行複數個不同的著色器，以繪製所述影像訊框序列。

【0483】範例6. 如範例1的處理器，其中所述平均像素值係基於對跨所述影像訊框序列的像素區塊的評估來確定的。

【0484】範例7. 如範例6的處理器，其中所述像素區塊包含巨集區塊。

【0485】範例8. 如範例7的處理器，其中所述巨集區塊包含下列中的一或多者：16×16像素區塊、8×8像素區塊、4×4像素區塊、16×8像素區塊、8×4像素區塊、或16×4像素區塊。

【0486】範例9. 一種方法，包含：在射線追蹤電路

上執行第一組的一或多個命令，以使射線遍歷通過邊界體積階層(BVH)，以識別與所述射線相交的BVH節點及/或基元；回應於第二組的一或多個命令，執行一或多個著色器，以基於與所述射線相交的所述BVH節點及/或基元，來繪製影像訊框序列；以及在包含運動估計電路的媒體處理器上，執行第三組的一或多個命令，以基於跨所述影像訊框序列收集的平均像素值，執行非局部平均濾波，以從所述影像訊框序列中移除雜訊。

【0487】範例10. 如範例9的方法，所述方法還包含：將所述第一組命令串流到所述射線追蹤電路，將所述第二組命令串流到所述著色器，並且將所述第三組命令串流到所述媒體處理器。

【0488】範例11. 如範例10的方法，其中所述第三組命令包含對與所述媒體處理器相關的應用程式化介面(API)的擴展。

【0489】範例12. 如範例9的方法，其中目標像素的所述非局部平均濾波包含確定與跨所述影像序列的所有像素相關的平均值，所述平均值被以與所述目標像素的相似性加權。

【0490】範例13. 如範例9的方法，其中所述一或多個著色器將在複數個執行單元(EU)上執行，以繪製所述影像訊框序列。

【0491】範例14. 如範例9的方法，其中所述平均像素值係基於對跨所述影像訊框序列的像素區塊的評估來確

定的。

【0492】範例15. 如範例14的方法，其中所述像素區塊包含巨集區塊。

【0493】範例16. 如範例15的方法，其中所述巨集區塊包含下列中的一或多者：16×16像素區塊、8×8像素區塊、4×4像素區塊、16×8像素區塊、8×4像素區塊、或16×4像素區塊。

【0494】範例17. 一種機器可讀取媒體，其上儲存有程式碼，當所述程式碼由機器執行時，使所述機器執行以下操作：在射線追蹤電路上執行第一組的一或多個命令，以使射線遍歷通過邊界體積階層(BVH)，以識別與所述射線相交的BVH節點及/或基元；回應於第二組的一或多個命令執行一或多個著色器，以基於與所述射線相交的所述BVH節點及/或基元，來繪製影像訊框序列；以及在包含運動估計電路的媒體處理器上，執行第三組的一或多個命令，以基於跨所述影像訊框序列收集的平均像素值，執行非局部平均濾波，以從所述影像訊框序列中移除雜訊。

【0495】範例18. 如範例17的機器可讀取媒體，還包含：將所述第一組命令串流到所述射線追蹤電路，將所述第二組命令串流到所述著色器，並且將所述第三組命令串流到所述媒體處理器。

【0496】範例19. 如範例18的機器可讀取媒體，其中所述第三組命令包含對與所述媒體處理器相關的應用程式化介面(API)的擴展。

【0497】範例20. 如範例17的機器可讀取媒體，其中目標像素的所述非局部平均濾波包含確定與跨所述影像訊框序列的所有像素相關的平均值，所述平均值係被與所述目標像素的相似性加權。

【0498】範例21. 如範例17的機器可讀取媒體，其中所述一或多個著色器將在複數個執行單元(EU)上執行，以繪製所述影像訊框序列。

【0499】範例22. 如範例17的機器可讀取媒體，其中所述平均像素值係基於對跨所述影像訊框序列的像素區塊的評估來確定的。

【0500】範例23. 如範例22的機器可讀取媒體，其中所述像素區塊包含巨集區塊。

【0501】範例24. 如範例23的機器可讀取媒體，其中所述巨集區塊包含下列中的一或多者：16×16像素區塊、8×8像素區塊、4×4像素區塊、16×8像素區塊、8×4像素區塊、或16×4像素區塊。

【0502】本發明的實施例可以包含上面已經描述的各種步驟。這些步驟可以體現在機器可執行指令中，所述機器可執行指令可以用於使通用或專用處理器執行這些步驟。替代地，這些步驟可以由包含用於執行步驟的硬連線邏輯的特定硬體部件來執行，或者由程式化的電腦部件和自定義硬體部件的任意組合來執行。

【0503】如本文中所描述，指令可是指硬體的特定配置，諸如被配置成執行某些操作或具有預定功能性的特殊

應用積體電路(ASIC)或儲存在非暫態電腦可讀取媒體中體現的記憶體中的軟體指令。因此，可以使用在一或多個電子裝置(例如，終端站、網路元件等)上儲存和執行的碼和資料來實現附圖中所示的技術。這種電子裝置使用電腦機器可讀取媒體(諸如非暫態電腦機器可讀取儲存媒體(例如磁碟；光碟；隨機存取記憶體；唯讀記憶體；快閃記憶體裝置；相變記憶體)和暫態電腦機器可讀取通訊媒體(例如電、光、聲或其它形式的傳播訊號，諸如載波、紅外線訊號、數位訊號等)來儲存碼和資料並且(內部地及/或透過網路與其它電子裝置)進行通訊。

【0504】此外，此類電子裝置通常包含與一或多個其它部件耦接的一組一或多個處理器，諸如一或多個儲存裝置(非暫態機器可讀取儲存媒體)、使用者輸入/輸出裝置(例如鍵盤、觸控螢幕及/或顯示器)和網路連接。一組處理器和其它部件的耦接通常透過一或多個匯流排和橋接器(也稱為匯流排控制器)進行。儲存裝置和承載網路訊務的訊號分別表示一或多種機器可讀取儲存媒體和機器可讀取通訊媒體。因此，給定電子裝置的儲存裝置通常儲存用於在所述電子裝置的一或多個處理器的集合上執行的碼及/或資料。當然，可以使用軟體、韌體及/或硬體的不同組合來實現本發明的實施例的一或多個部分。在整個詳細描述中，出於解釋的目的，闡述了許多具體細節以便提供對本發明的透徹理解。然而，對本領域技術人員顯而易見的是，可以在沒有這些具體細節中的一些的情況下實踐本發

明。在某些情況下，沒有詳細描述眾所周知的結構和功能，以避免使本發明的申請標的不清楚。因此，本發明的範圍和精神應根據所附申請專利範圍來判斷。

【符號說明】

【0505】

100:處理系統

102:處理器

104:快取記憶體

106:暫存器檔案

107:處理器核心

108:圖形處理器

109:指令集

110:介面匯流排

111:顯示裝置

112:加速器

116:記憶體控制器

118:外部圖形處理器

119:外部加速器

120:記憶體裝置

121:指令

122:資料

124:資料儲存裝置

125:觸控感測器

126:無線收發器

128:韌體介面

130:平臺控制器集線器

134:網路控制器

140:I/O控制器

142:通用串列匯流排(USB)控制器

143:鍵盤和滑鼠

144:相機

146:音訊控制器

200:處理器

202A-202N:處理器核心

204A-204N :內部快取單元

206:共享快取單元

206A-206F:媒體取樣器

208:整合圖形處理器

210:系統代理核心

211:顯示控制器

212:環狀互連

213:I/O鏈路

214:整合記憶體控制器

216:匯流排控制器單元

218:高效能嵌入式記憶體模組

219:圖形處理器核心

221A-221F:子核心

222A-222F:執行單元(EU)陣列

223A-223F:執行緒分派和執行緒間通訊(TD/IC)邏輯

224A-224F:執行單元(EU)陣列

225A-225F:3D取樣器

227A-227F:著色器處理器

228A-228F:共享區域記憶體(SLM)

230:固定功能方塊

231:幾何/固定的功能管線

232:圖形SoC介面

233:圖形微控制器

234:媒體管線

235:共享的功能邏輯

236:共享及/或快取記憶體

237:幾何/固定功能管線

238:固定功能邏輯

239:圖形處理單元(GPU)

240A-240N:多核心組

241:排程器/分派器

242:暫存器檔案

243:圖形核心

244:張量核心

245:射線追蹤核心

246:CPU

247:共享記憶體單元

248:記憶體控制器

249:記憶體

250:輸入/輸出(I/O)電路

251:I/O記憶體管理單元(IOMMU)

252:I/O裝置

253:L2快取

254:L1快取

255:指令快取

256:共享記憶體

257:命令處理器

258:執行緒分派器

260A-260N:計算單元

261:向量暫存器

262:純量暫存器

263:向量邏輯單元

264:純量邏輯單元

265:區域共享記憶體

266:程式計數器

267:常數快取

268:記憶體控制器

269:內部直接記憶體存取(DMA)控制器

270:通用圖形處理單元(GPGPU)

271:記憶體

272:記憶體

300:圖形處理器

302:顯示控制器

304:區塊影像轉換(BLIT)引擎

306:視訊編解碼器引擎

310:圖形處理引擎(GPE)

312:3D管線

314:記憶體介面

315:3D/媒體子系統

316:媒體管線

318:顯示裝置

320:圖形處理器

322:圖形處理引擎集群

323A-323F:塊互連

324:結構互連

325A-325D:記憶體互連

326A-326D :記憶體裝置

310A-310D:圖形引擎塊

328:主機介面

330:計算加速器

332:計算引擎集群

336:L3快取

340A-340D:計算引擎塊

403:命令串流器

410:圖形處理引擎

414:圖形核心陣列

415A,415B:圖形核心

416:共享功能邏輯

418:統一返回緩衝器

420:共享功能邏輯

421:取樣器

422:數學邏輯

423:執行緒間通訊(ITC)邏輯

425:快取

500:執行緒執行邏輯

502:著色器處理器

504:執行緒分派器

505:射線追蹤器

506:指令快取

507A-507N:執行緒控制邏輯

508A-508N:執行單元

510:取樣器

511:共享區域記憶體

512:資料快取

514:資料埠

522:執行緒仲裁器

524:一般暫存器檔案陣列(GRF)

526:架構暫存器檔案陣列(ARF)

530:發送單元

532:分支單元

534:SIMD浮點單元(FPU)

535:專用整數SIMD ALU

537:指令提取單元

600:執行單元

601:執行緒控制單元

602:執行緒狀態單元

603:指令提取/預提取單元

604:指令解碼單元

606:暫存器檔案

607:發送單元

608:分支單元

610:計算單元

611:ALU單元

612:脈動陣列

613:數學單元

700:圖形處理器指令格式

710:128位元指令格式

712:指令運算碼

713:索引欄位

714:指令控制欄位

716:執行大小欄位

718:目的地

720:來源運算元

722:來源運算元

724:來源運算元

726:存取/位址模式欄位

730:64位元格式

740:運算碼解碼

742:移動和邏輯運算碼編組

744:流程控制指令編組

746:混雜指令編組

748:平行數學編組

750:向量數學編組

800:圖形處理器

802:環狀互連

803:命令串流器

805:頂點提取器

807:頂點著色器

811:可程式化外殼著色器

813:鑲嵌器

817:可程式化域著色器

819:幾何著色器

820:幾何管線

823:流輸出單元

829:剪輯器

830:媒體管線

831:執行緒分派器

834:視訊前端

837:媒體引擎

840:顯示引擎

841:2D引擎

843:顯示控制器

850:執行緒執行邏輯

851:L1快取

852A-852B:執行單元

854:取樣器

856:資料埠

858:紋理快取

870:繪製輸出管線

873:光柵器和深度測試部件

875:共享L3快取

877:像素操作部件

878:關聯繪製快取

879:深度快取

900:圖形處理器命令格式

902:客戶端

904:命令操作代碼(運算碼)

905:子運算碼

906:資料

908:命令大小

910:圖形處理器命令序列

912:管線清除命令

913:管線選擇命令

914:管線控制命令

916:返回緩衝器狀態命令

920:管線確定

922:3D管線

924:媒體管線

930:3D管線狀態

932:3D基元命令

934:運行

940:媒體管線狀態

942:媒體物件命令

944:運行命令

1000:資料處理系統

1010:3D圖形應用

1012:著色器指令

1014:可執行指令

1016:圖形物件

1020:作業系統

1022:圖形API

1024:前端著色器編譯器

1026:使用者模式圖形驅動器

1027:後端著色器編譯器

1028:作業系統核心模式功能

1029:核心模式圖形驅動器

1030:處理器

1032:圖形處理器

1034:通用處理器核心

1050:系統記憶體

1100:IP核心開發系統

1110:軟體模擬

1112:模擬模型

1115:暫存器傳輸級(RTL)設計

1120:硬體模型

1130:設計機構

1140:非揮發性記憶體

1150:有線連接

1160:無線連接

1165:製作機構

1170:封裝組件

1172:硬體邏輯

1173:互連結構

1174:硬體邏輯

1175:記憶體小晶片

1180:基板

1182:橋接器

1183:封裝互連

1190:封裝組件

1191:I/O

1192:快取記憶體

1193:硬體邏輯

1194:封裝組件

1195:可互換小晶片

1196:基礎小晶片

1197:橋接器互連

1198:基礎小晶片

1200:單晶片積體電路

1205:應用處理器

1210:圖形處理器

1215:影像處理器

1220:視訊處理器

1225:USB控制器

1230:UART控制器

1235:SPI/SDIO控制器

1240:I2S/I2C控制器

1245:顯示裝置

1250:高解析度多媒體介面(HDMI)控制器

1255:行動產業處理器介面(MIPI)顯示介面

1260:快閃記憶體子系統

1265:記憶體控制器

1270:嵌入式安全引擎

1305:頂點處理器

1310:圖形處理器

1315A-1315N:片段處理器

1320A-1320B:記憶體管理單元(MMU)

1325A-1325B:快取

1330A-1330B:電路互連

1340:圖形處理器

1345:核心間任務管理器

1355A-1355N:著色器核心

1358:平鋪單元

1500:機器學習引擎

1501:低樣本數輸入

1502:高樣本數影像資料

1505:權重

1600:機器學習引擎

1601:低樣本數輸入訊框

1602:新參考區域

1603:低樣本數緩衝器

1604:高樣本數緩衝器

1605:訓練後的權重

1607:低樣本數區域

1620:降噪訊框

1700:背景訓練程序

1700A-C:背景訓練程序

1710:主機CPU

1800:伺服器

1805:權重

1810:機器學習引擎

1820:主機機器

1821:主機機器

1822:主機機器

1901:操作

1902:操作

1903:操作

1904:操作

2001:重影區域

2002:重影區域

2003:重影區域

2011:區域

2012:區域

2013:區域

2021:節點

2022:節點

2023:節點

2081:射線追蹤繪製電路

2082:射線追蹤繪製電路

2100:降噪器

2101:區域資料管理器

2102:區域資料管理器

2111:降噪器

2121:降噪區域

2122:降噪區域

2131:壓縮器/解壓縮器單元

2132:壓縮器/解壓縮器單元

2201:監視器

2202:監視器

2211:網路介面

2212:網路介面

2280:區域處理器

2290:降噪影像

2301:操作

2302:操作

2303:操作

2304:操作

2305:操作

2400:機器學習軟體堆疊

2402:機器學習應用

2404:機器學習框架

2406:計算框架

2408:GPGPU驅動器

2410:GPGPU硬體

2500:多GPU計算系統

2502:處理器

2504:主機介面開關

2506A-D:GPGPU

2516:P2P GPU鏈路

2602:輸入

2604:卷積層

2606:卷積層

2608:完全連接層

2712:卷積層的輸入

2714:卷積層

2716:卷積階段

2718:檢測器階段

2720:池化階段

2722:下一層

2800:遞迴類神經網路

2802:輸入層

2804:隱藏層

2805:回饋機制

2806:輸出層

2902:訓練資料集

2904:訓練框架

2906:未訓練類神經網路

2907:經訓練類神經網

2908:經訓練類神經網

2912:新資料

3002:模型平行性

3004:資料平行性

3006:結合的模型及資料平行性

3100:推理系統單晶片(SOC)

3100A-N:多核心組

3102:媒體處理器

3104:視覺處理器

3105:晶載記憶體

3106:GPGPU

3108:多核心處理器

3110:排程器/分派器

3120:暫存器檔案

3130:核心

3140:核心

3150:核心

3160:1階(L1)快取和紋理單元

3170:IO記憶體管理單元(IOMMU)

3180:2階(L2)快取

3190:I/O裝置

3193:顯示處理器

3195:輸入/輸出(IO)電路

3197:媒體處理引擎

3198:系統記憶體

3199:CPU

3200:貼片

3201:光束

3304:射線產生電路

3305:光束階層構建電路

3306:遍歷/相交電路

3307:光束階層

3308:BVH

3320:GPU

3400:主光束

3401:象限

3402:象限

3403:象限

3404:象限

3500:操作

3501:操作

3502:操作

3503:操作

3600:射線追蹤集群

3610:射線追蹤節點

3611:射線追蹤節點

3612:射線追蹤節點

3613:射線追蹤節點

3630:客戶側射線追蹤應用程式

3701:壓縮射線封包

3710:射線追蹤節點

3711:射線追蹤節點

3720:射線壓縮電路

3721:射線解壓縮電路

3725:網路介面

3726:網路介面

3730:射線追蹤引擎

3731:射線追蹤引擎

3740:射線解壓縮電路

3741:射線解壓縮電路

3801:有損壓縮電路

3802:有損解壓縮電路

3803:無損壓縮電路

3804:無損解壓縮電路

3900:操作

3901:操作

3902:操作

3903:操作

3904:操作

3905:操作

4000:著色器執行電路

4001:單一指令多重執行緒(SIMT)核心/執行單元(EU)

4002:取樣器

4003:1階(L1)快取

4004:訊息單元

4005:射線BVH遍歷/相交電路

4006:區域L1快取

4007:排程器

4008:排序電路

4009:主分派器

4010:固定功能硬體

4020:最末級快取(LLC)

4101:分配堆疊

4102:閒置堆疊池

4201:內容可定址記憶體(CAM)結構

4201A, 4201B, 4201n :著色器記錄指標

4210:分組電路

4300:單一頂層BVH

4301:底層BVH

4302:底層BVH

4303:內部遍歷路徑

4304:外部遍歷路徑

4305:遍歷節點

4306:具有三角形的葉節點

4307:具有自定義基元的葉節點

4400:遍歷狀態

4401:世界空間

4402:物件空間

4403:偏移

4404:偏移

4405:調用堆疊

4410:截斷內部堆疊

4501:主射線著色器

4502:遍歷電路

4503:相交電路

4504:任何命中著色器

4505:相交著色器

4506:未命中著色器

4507:最接近的命中著色器

4601:遍歷狀態

4650:自定義基元

4701:著色器

4706:著色器推遲/聚合器電路

4805:著色器推遲/聚合器電路

4900:射線追蹤指令

4902:射線追蹤資料

4903:指令提取單元

4904:解碼器

4905:排程器/分派器

4908:區域快取子系統

4910:功能單元(FU)

4911:功能單元(FU)

4912:功能單元(FU)

4915:向量暫存器

4916:純量暫存器

4995:解碼器

5001:操作

5002:操作

5003:操作

5004:操作

5005:操作

5006:操作

5007:操作

5008:操作

5101:主要射線著色器

5102:遍歷電路

5103:射線BVH相交電路

5107:區域射線追蹤快取(RTC)

5110:射線追蹤加速電路

5121:調用堆疊

5201:射線儲存庫

5202:射線儲存庫

5203:堆疊

5204:堆疊

5205:射線分配器

5206:入射射線

5210:射線處理電路

5220:庫分配計數器

5248:遍歷/堆疊追蹤器

5249:追蹤陣列

5302:媒體API

5305:非局部平均命令

5310:命令串流器

5320:運動估計電路

5330:影像訊框

5340:降噪訊框

5401:操作

5402:操作

5403:操作

5404:操作

【發明申請專利範圍】

【請求項1】一種處理器，包含：

射線追蹤電路，其用以執行第一組的一或多個命令，以使射線遍歷通過邊界體積階層(BVH)，以識別與所述射線相交的BVH節點及/或基元；

著色器執行電路，其用以回應於第二組的一或多個命令而執行一或多個著色器，以基於與所述射線相交的所述BVH節點及/或基元來繪製影像訊框序列；

媒體取樣器，用以存取記憶體並針對儲存於該記憶體中的圖形資料執行紋理取樣操作；以及

媒體處理器，用以編碼與解碼該影像訊框序列，該媒體處理器包含運動估計電路，所述運動估計電路用以執行第三組的一或多個命令，以基於跨所述影像訊框序列收集的平均像素值，執行非局部平均濾波，以從所述影像訊框序列中移除雜訊，其中在當前影像訊框中的目標像素的該非局部平均濾波包含確定跨所述影像訊框序列的所有像素以及在各個個別訊框內的像素相關的平均值，其中跨所述影像訊框序列包含在該序列中的所述當前影像訊框之前與之後的影像訊框，所述平均值被以與所述目標像素的相似性加權，

其中該平均值是根據跨所述影像訊框序列的像素區塊的評估來確定，各個影像訊框係被以所述像素區塊的不對稱排列編碼，使得給定影像訊框的中心區域使用較小像素區塊，以較高的精確度編碼，而該給定影像訊框的外圍區

域使用較大像素區塊，以較低精確度編碼。

【請求項2】如請求項1的處理器，其中所述第三組命令包含對與所述媒體處理器相關的應用程式化介面(API)的擴展。

【請求項3】如請求項1的處理器，其中所述著色器執行電路包含複數個執行單元(EU)，以執行複數個不同的著色器以繪製所述影像訊框序列。

【請求項4】如請求項1的處理器，其中所述像素區塊包含巨集區塊。

【請求項5】如請求項4的處理器，其中所述巨集區塊包含下列中的一或多者：16×16像素區塊、8×8像素區塊、4×4像素區塊、16×8像素區塊、8×4像素區塊、或16×4像素區塊。

【請求項6】如請求項1的處理器，更包含第一介面，以使得該媒體處理器的該運動估計電路對於包含3D圖形管線內的深度評估操作、視訊影像的去交錯、及/或虛擬實境實現的視圖內插的其它應用處理器元件與應用為可存取。

【請求項7】如請求項1的處理器，更包含命令串流器，用以同時將所述第一組命令串流到所述射線追蹤電路、將所述第二組命令串流到所述著色器執行電路，及將所述第三組命令串流到所述媒體處理器的該運動估計電路。

【請求項8】一種執行非局部平均濾波的方法，包

含：

在射線追蹤電路上執行第一組的一或多個命令，以使射線遍歷通過邊界體積階層(BVH)，以識別與所述射線相交的BVH節點及/或基元；

回應於第二組的一或多個命令，執行一或多個著色器，以基於與所述射線相交的所述BVH節點及/或基元來繪製影像訊框序列；

以媒體取樣器，對儲存於記憶體中的圖形資料執行紋理取樣操作；

以媒體處理器，對該影像訊框序列執行編碼及/或解碼操作；以及

在該媒體處理器的運動估計電路上執行第三組的一或多個命令，以基於跨所述影像訊框序列收集的平均像素值，執行非局部平均濾波，以從所述影像訊框序列中移除雜訊，其中在當前影像訊框中的目標像素的所述非局部平均濾波包含確定跨所述影像訊框序列的所有像素以及在各個個別訊框內的像素相關的平均值，其中跨所述影像訊框序列包含在該序列中的所述當前影像訊框之前與之後的影像訊框，所述平均值被以與所述目標像素的相似性加權，

其中該平均值是根據跨所述影像訊框序列的像素區塊的評估來確定，各個影像訊框係被以所述像素區塊的不對稱排列編碼，使得給定影像訊框的中心區域使用較小像素區塊，以較高的精確度編碼，而該給定影像訊框的外圍區域使用較大像素區塊，以較低精確度編碼。

【請求項9】如請求項8的方法，其中所述第三組命令包含對與所述媒體處理器相關的應用程式化介面(API)的擴展。

【請求項10】如請求項8的方法，其中所述一或多個著色器將在複數個執行單元(EU)上執行，以繪製所述影像訊框序列。

【請求項11】如請求項8的方法，其中所述像素區塊包含巨集區塊。

【請求項12】如請求項11的方法，其中所述巨集區塊包含下列中的一或多者：16×16像素區塊、8×8像素區塊、4×4像素區塊、16×8像素區塊、8×4像素區塊、或16×4像素區塊。

【請求項13】如請求項8的方法，更包含透過命令串流器，同時將所述第一組命令串流到所述射線追蹤電路，將所述第二組命令串流到所述著色器，及將所述第三組命令串流到所述媒體處理器的該運動估計電路。

【請求項14】一種非暫態機器可讀取媒體，其上儲存有程式碼，當所述程式碼由機器執行時，使所述機器執行以下操作：

在射線追蹤電路上執行第一組的一或多個命令，以使射線遍歷通過邊界體積階層(BVH)，以識別與所述射線相交的BVH節點及/或基元；

回應於第二組的一或多個命令執行一或多個著色器，以基於與所述射線相交的所述BVH節點及/或基元來繪製

影像訊框序列；

以媒體取樣器，對儲存於記憶體中的圖形資料執行紋理取樣操作；

以媒體處理器，對該影像訊框序列執行編碼及/或解碼操作；以及

在所述媒體處理器的運動估計電路上執行第三組的一或多個命令，以基於跨所述影像訊框序列收集的平均像素值，執行非局部平均濾波，以從所述影像訊框序列中移除雜訊，其中在當前影像訊框中的目標像素的所述非局部平均濾波包含確定跨所述影像訊框序列的所有像素以及在各個個別訊框內的像素相關的平均值，其中跨所述影像訊框序列包含在該序列中的所述當前影像訊框之前與之後的影像訊框，所述平均值被以與所述目標像素的相似性加權，

其中該平均值是根據跨所述影像訊框序列的像素區塊的評估來確定，各個影像訊框係被以所述像素區塊的不對稱排列編碼，使得給定影像訊框的中心區域使用較小像素區塊，以較高的精確度編碼，而該給定影像訊框的外圍區域使用較大像素區塊，以較低精確度編碼。

【請求項15】如請求項14的非暫態機器可讀取媒體，其中所述第三組命令包含對與所述媒體處理器相關的應用程式化介面(API)的擴展。

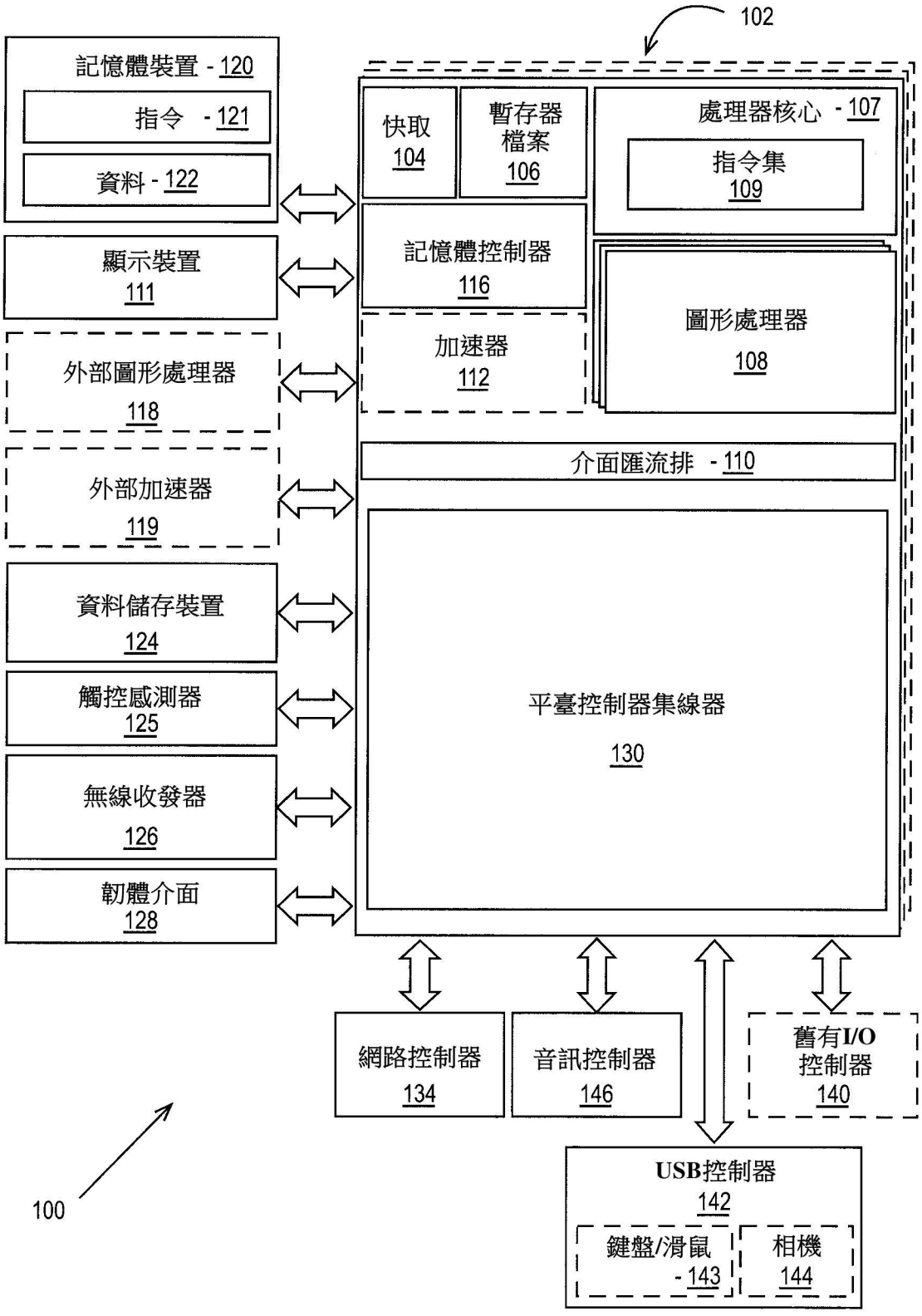
【請求項16】如請求項14的非暫態機器可讀取媒體，其中所述一或多個著色器將在複數個執行單元(EU)上執行，以繪製所述影像訊框序列。

【請求項17】如請求項14的非暫態機器可讀取媒體，其中所述像素區塊包含巨集區塊。

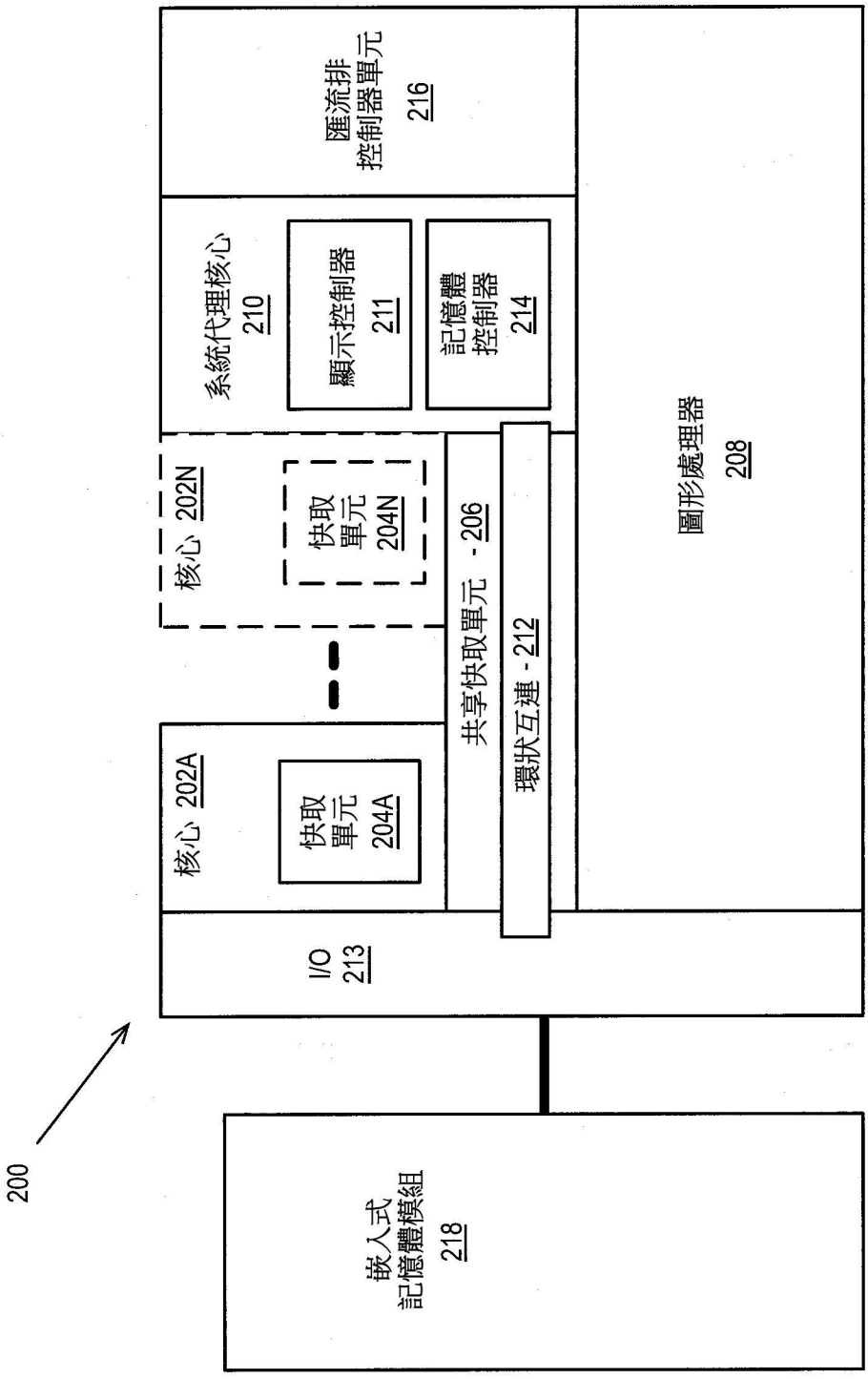
【請求項18】如請求項17的非暫態機器可讀取媒體，其中所述巨集區塊包含下列中的一或多者：16×16像素區塊、8×8像素區塊、4×4像素區塊、16×8像素區塊、8×4像素區塊、或16×4像素區塊。

【請求項19】如請求項14的非暫態機器可讀取媒體，其中所述操作更包含透過命令串流器，將所述第一組命令串流到所述射線追蹤電路，將所述第二組命令串流到所述著色器，及將所述第三組命令串流到所述媒體處理器的該運動估計電路。

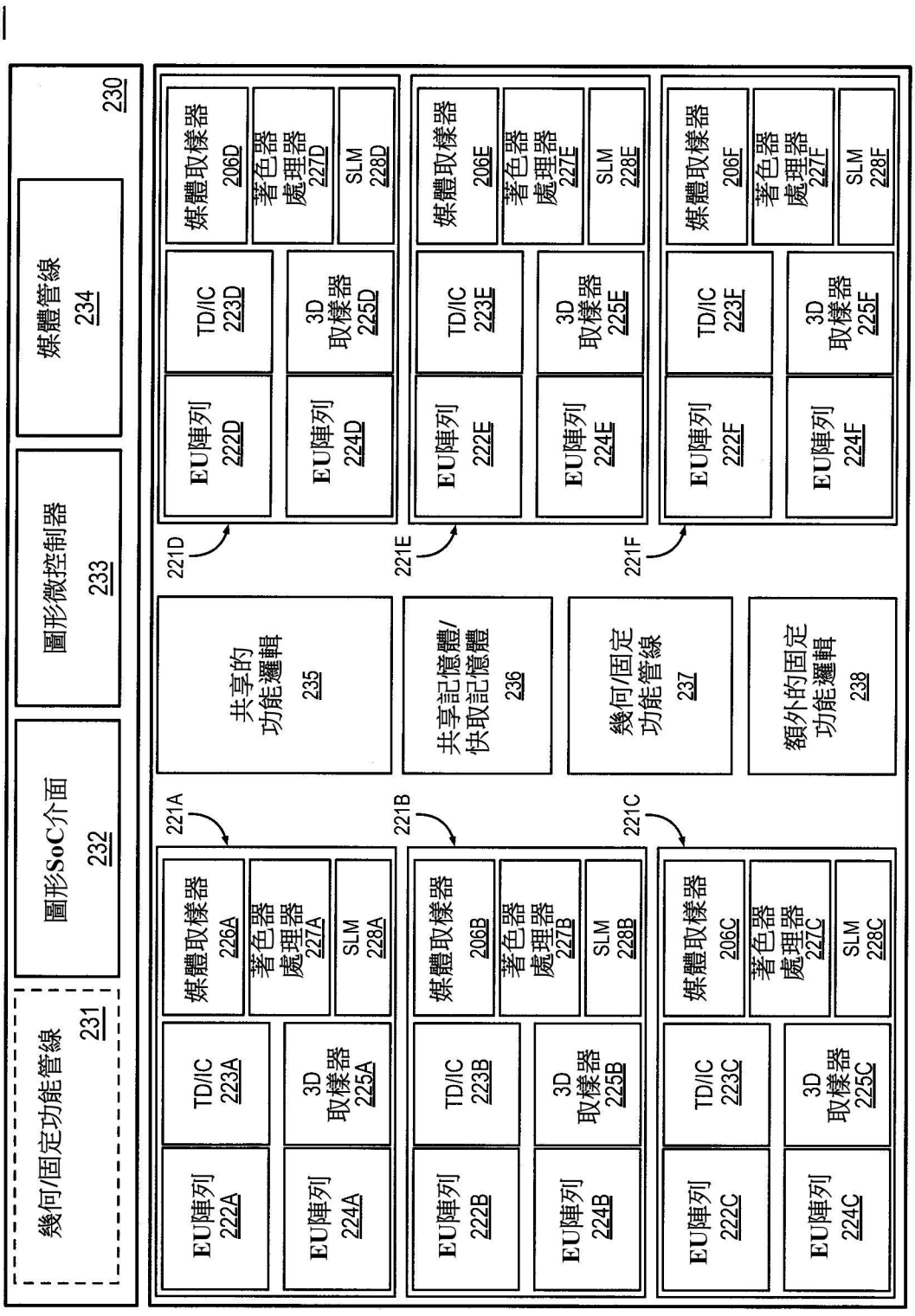
【發明圖式】


102
記憶體裝置 - 120
指令 - 121
資料 - 122
顯示裝置
111
外部圖形處理器
118
外部加速器
119
資料儲存裝置
124
觸控感測器
125
無線收發器
126
韌體介面
128
快取
104
暫存器
檔案
106
處理器核心 - 107
指令集
109
記憶體控制器
116
圖形處理器
108
加速器
112
介面匯流排 - 110
平臺控制器集線器
130
網路控制器
134
音訊控制器
146
舊有I/O
控制器
140
USB控制器
142
鍵盤/滑鼠
- 143
相機
144
100

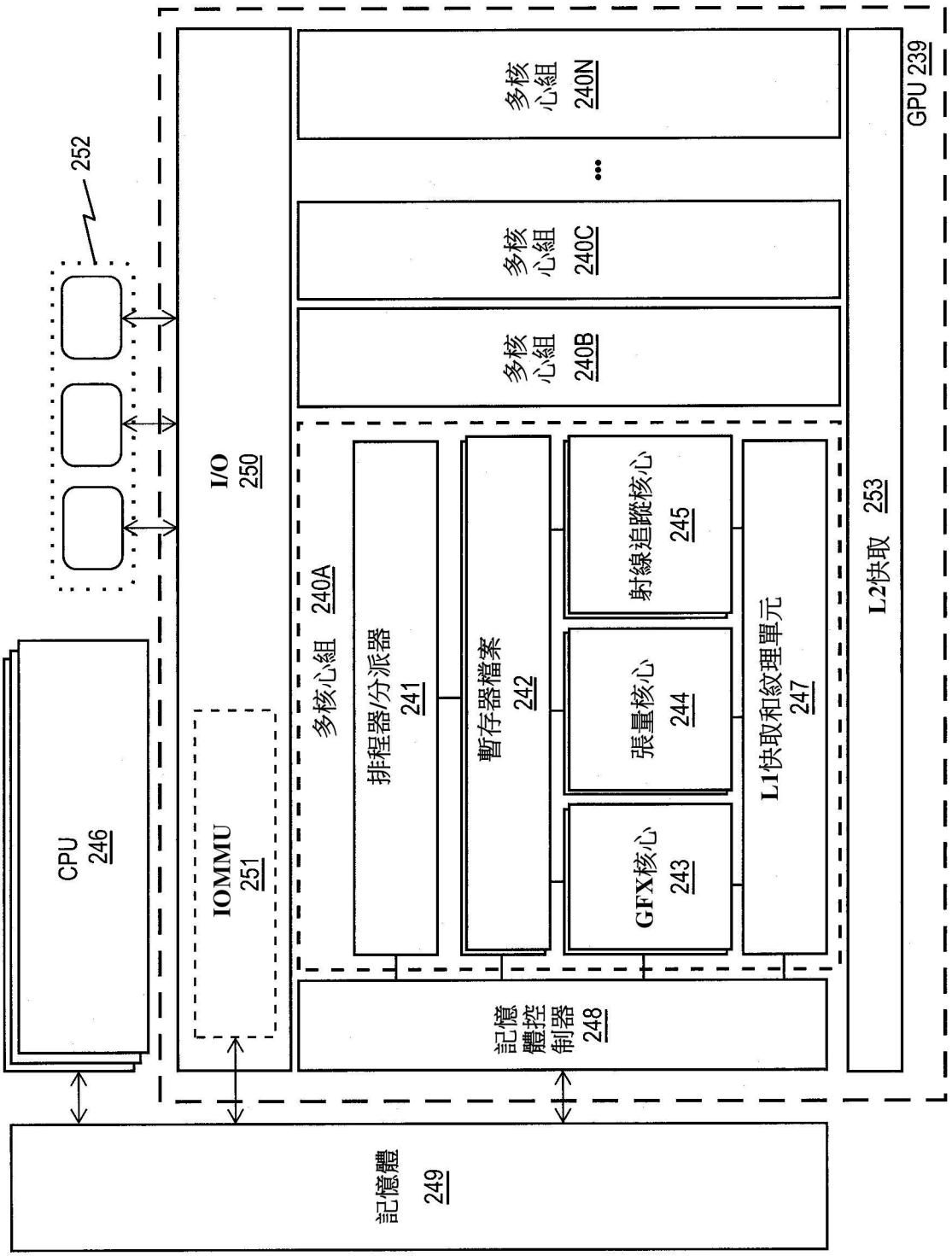
【圖 1】


200
嵌入式
記憶體模組
218
I/O
213
核心 202A
快取
單元
204A
核心 202N
快取
單元
204N
共享快取單元 - 206
環狀互連 - 212
系統代理核心
210
顯示控制器
211
記憶體
控制器
214
匯流排
控制器單元
216
圖形處理器
208

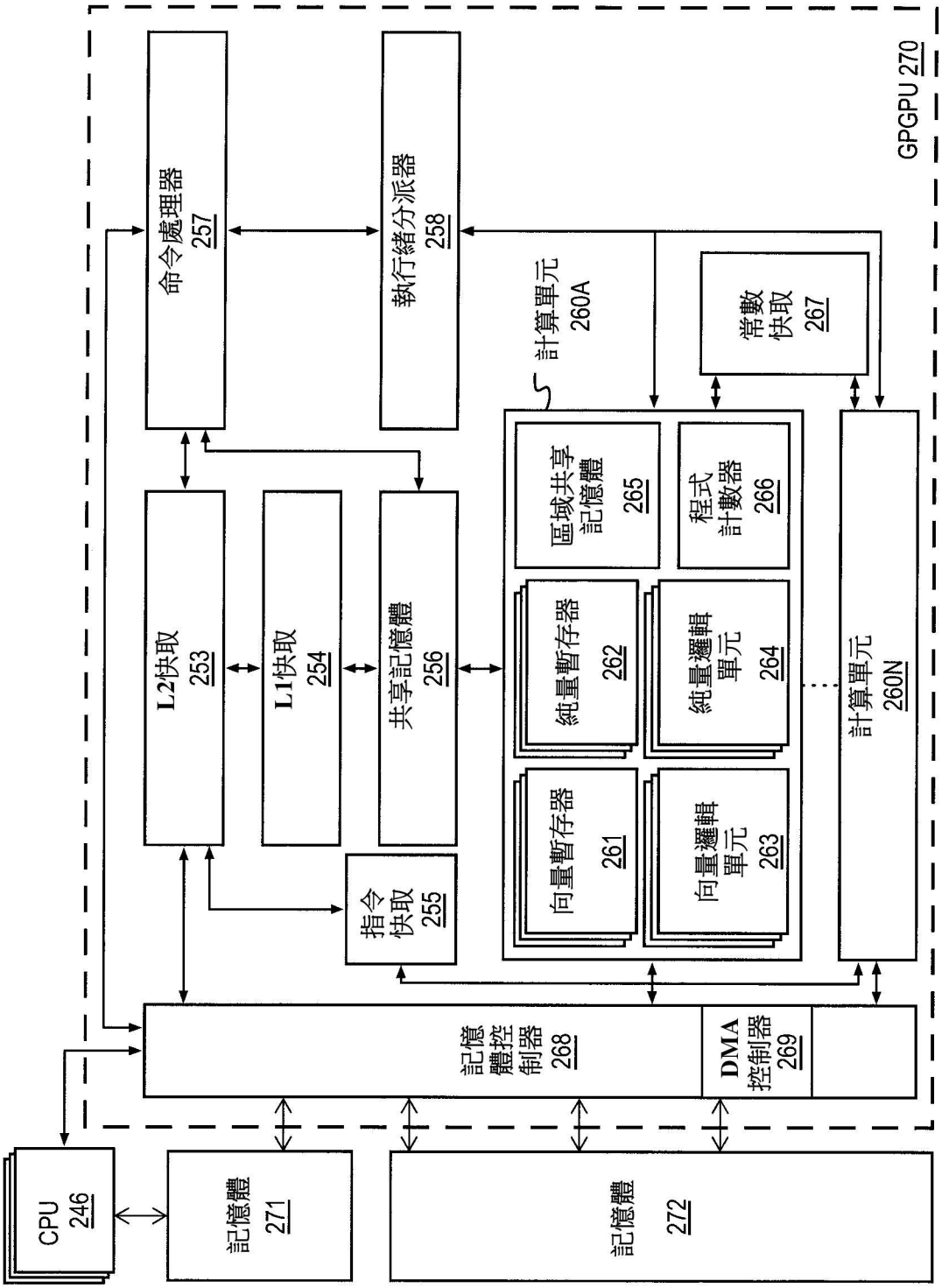
【圖 2A】


219
230
幾何/固定功能管線
231
圖形SoC介面
232
圖形微控制器
233
媒體管線
234
共享的
功能邏輯
235
共享記憶體/
快取記憶體
236
幾何/固定
功能管線
237
額外的固定
功能邏輯
238
221A
221B
221C
221D
221E
221F
EU陣列
222A
EU陣列
224A
TD/IC
223A
3D
取樣器
225A
媒體取樣器
226A
著色器
處理器
227A
SLM
228A
EU陣列
222B
EU陣列
224B
TD/IC
223B
3D
取樣器
225B
媒體取樣器
206B
著色器
處理器
227B
SLM
228B
EU陣列
222C
EU陣列
224C
TD/IC
223C
3D
取樣器
225C
媒體取樣器
206C
著色器
處理器
227C
SLM
228C
EU陣列
222D
EU陣列
224D
TD/IC
223D
3D
取樣器
225D
媒體取樣器
206D
著色器
處理器
227D
SLM
228D
EU陣列
222E
EU陣列
224E
TD/IC
223E
3D
取樣器
225E
媒體取樣器
206E
著色器
處理器
227E
SLM
228E
EU陣列
222F
EU陣列
224F
TD/IC
223F
3D
取樣器
225F
媒體取樣器
206F
著色器
處理器
227F
SLM
228F

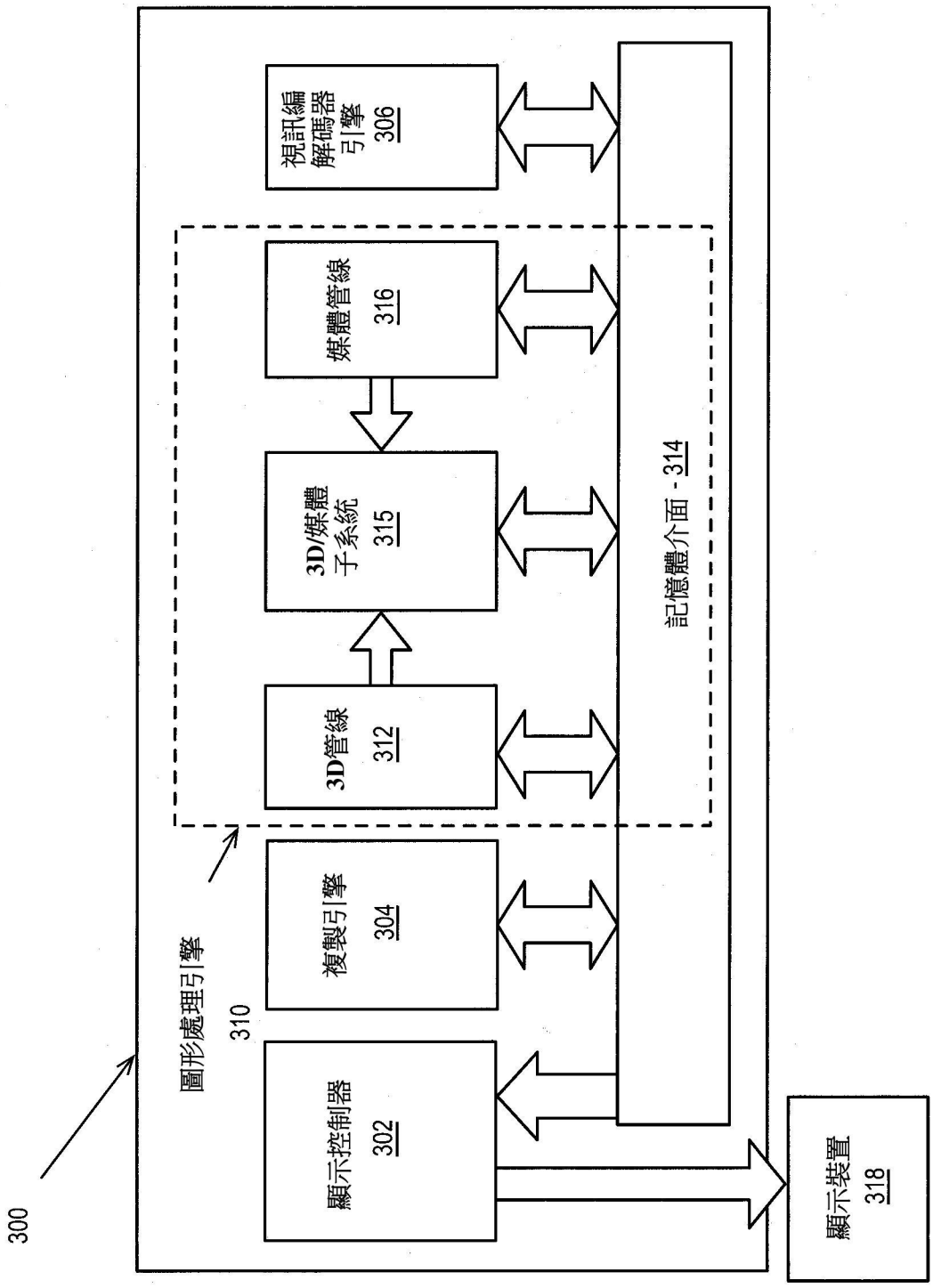
【圖 2B】


252
I/O
250
GPU 239
多核心組
240N
多核心組
240C
多核心組
240B
多核心組 240A
排程器/分派器
241
暫存器檔案
242
射線追蹤核心
245
張量核心
244
GFX核心
243
L1快取和紋理單元
247
L2快取 253
記憶體控制器
248
CPU
246
IOMMU
251
記憶體
249

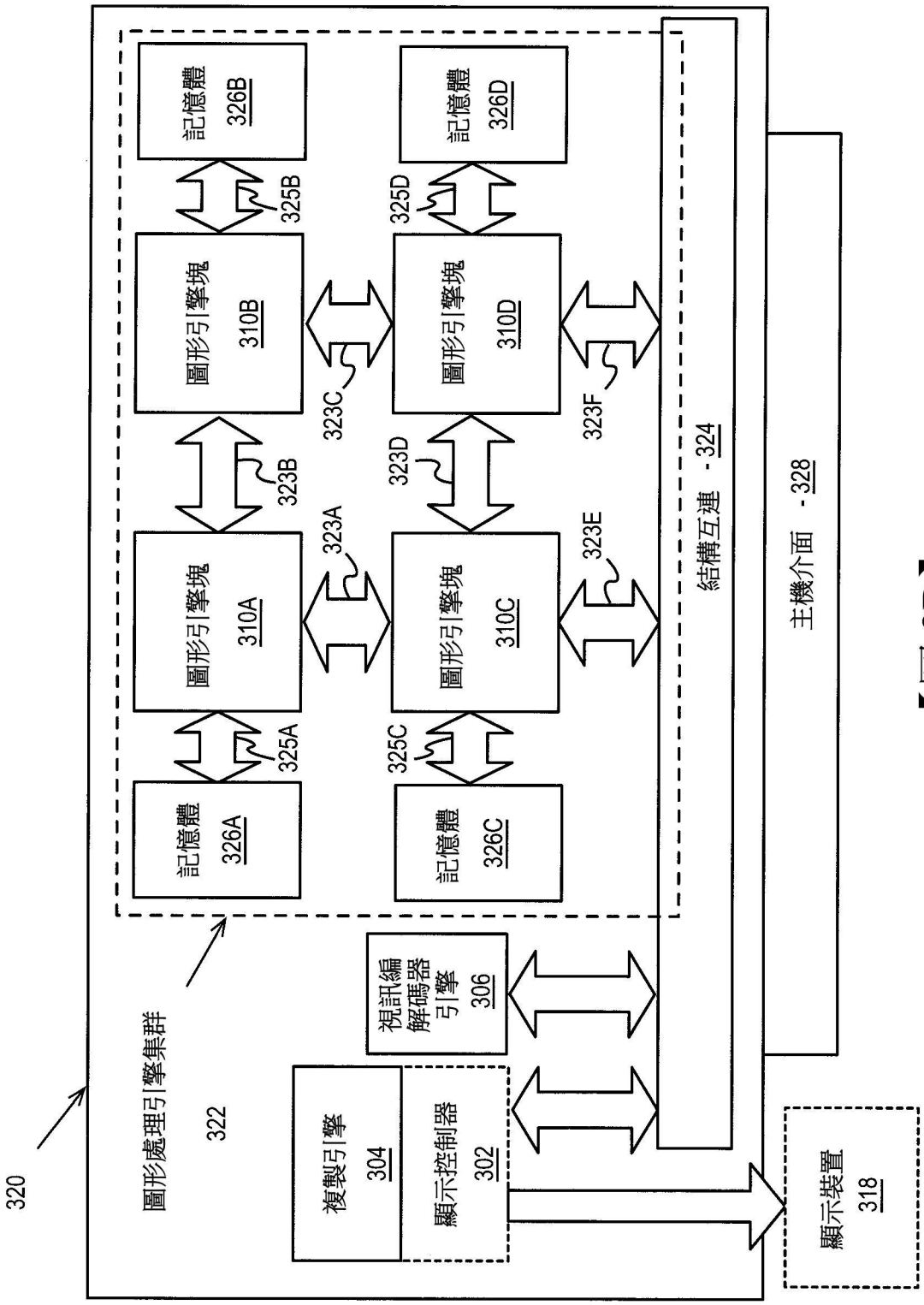
【圖 2C】


GPGPU 270
命令處理器 257
執行緒分派器 258
計算單元 260A
常數快取 267
區域共享記憶體 265
程式計數器 266
純量暫存器 262
純量邏輯單元 264
向量暫存器 261
向量邏輯單元 263
計算單元 260N
L2快取 253
L1快取 254
共享記憶體 256
指令快取 255
記憶體控制器 268
DMA控制器 269
CPU 246
記憶體 271
記憶體 272

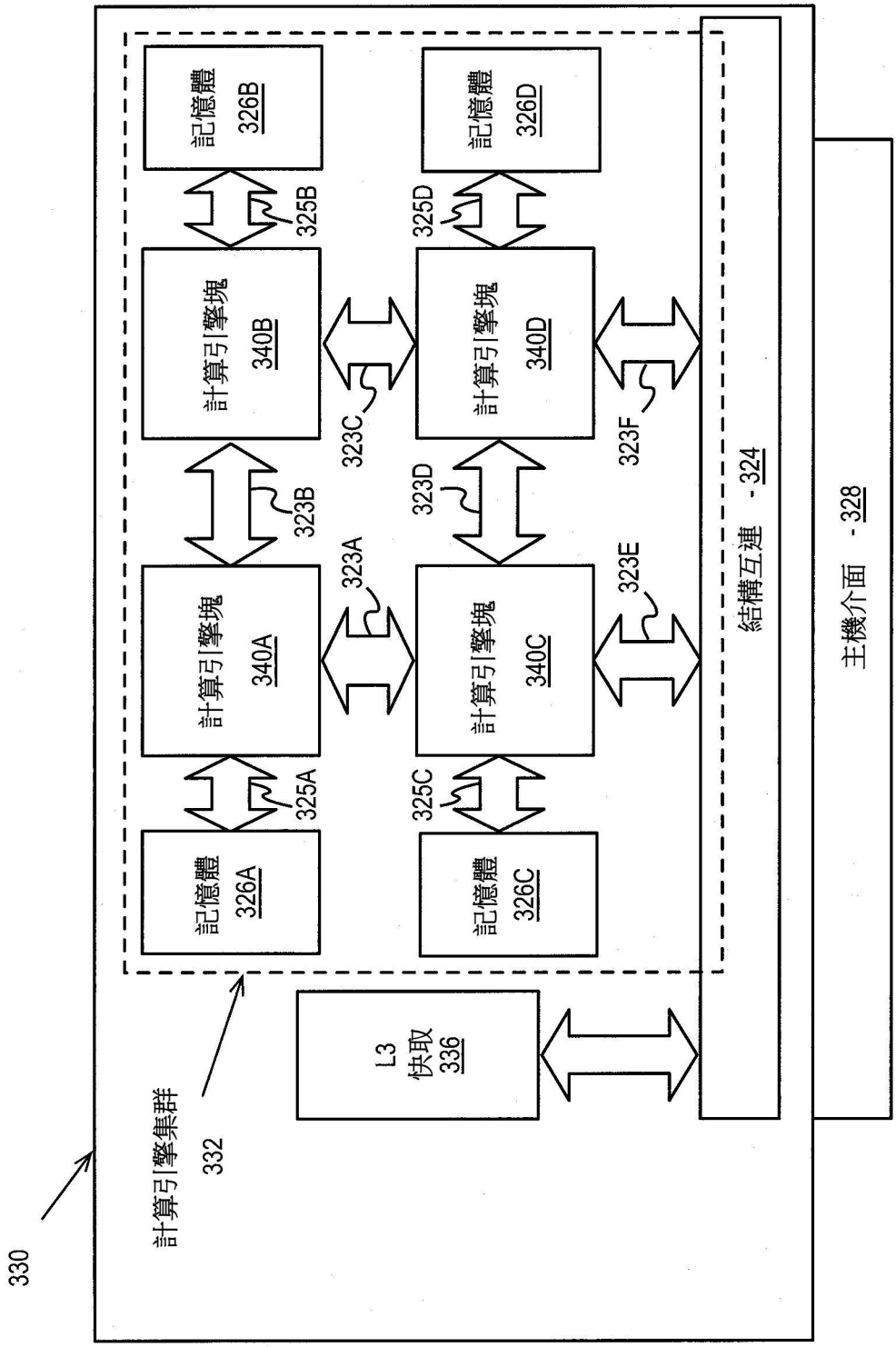
【圖 2D】


300
圖形處理引擎
310
顯示控制器
302
複製引擎
304
3D管線
312
3D/媒體
子系統
315
媒體管線
316
視訊編
解碼器
引擎
306
記憶體介面 - 314
顯示裝置
318

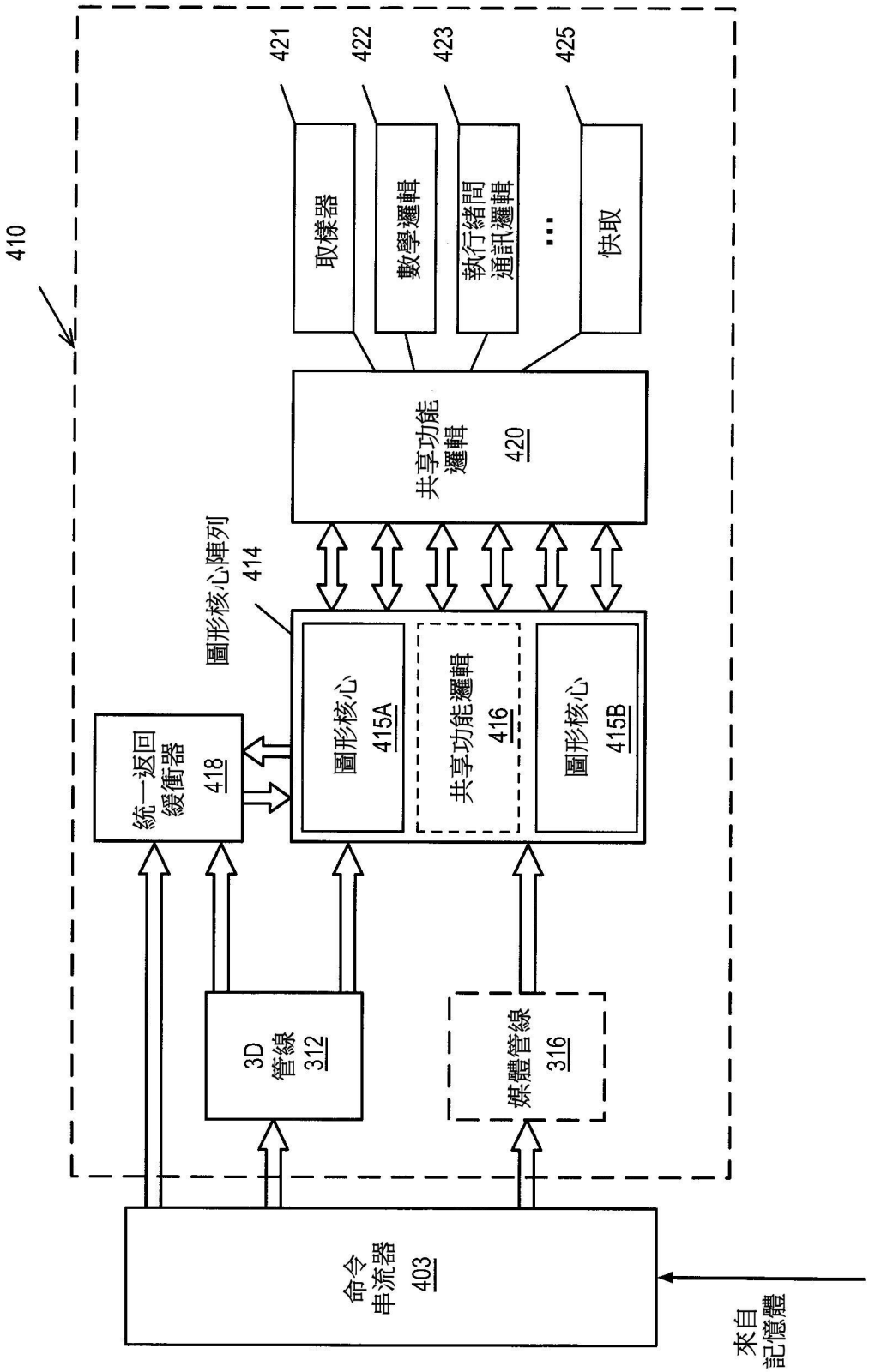
【圖 3A】


320
圖形處理引擎集群
322
記憶體 326A
記憶體 326B
記憶體 326C
記憶體 326D
圖形引擎塊 310A
圖形引擎塊 310B
圖形引擎塊 310C
圖形引擎塊 310D
325A
325B
325C
325D
323A
323B
323C
323D
323E
323F
結構互連 324
主機介面 328
視訊編解碼器引擎 306
複製引擎 304
顯示控制器 302
顯示裝置 318

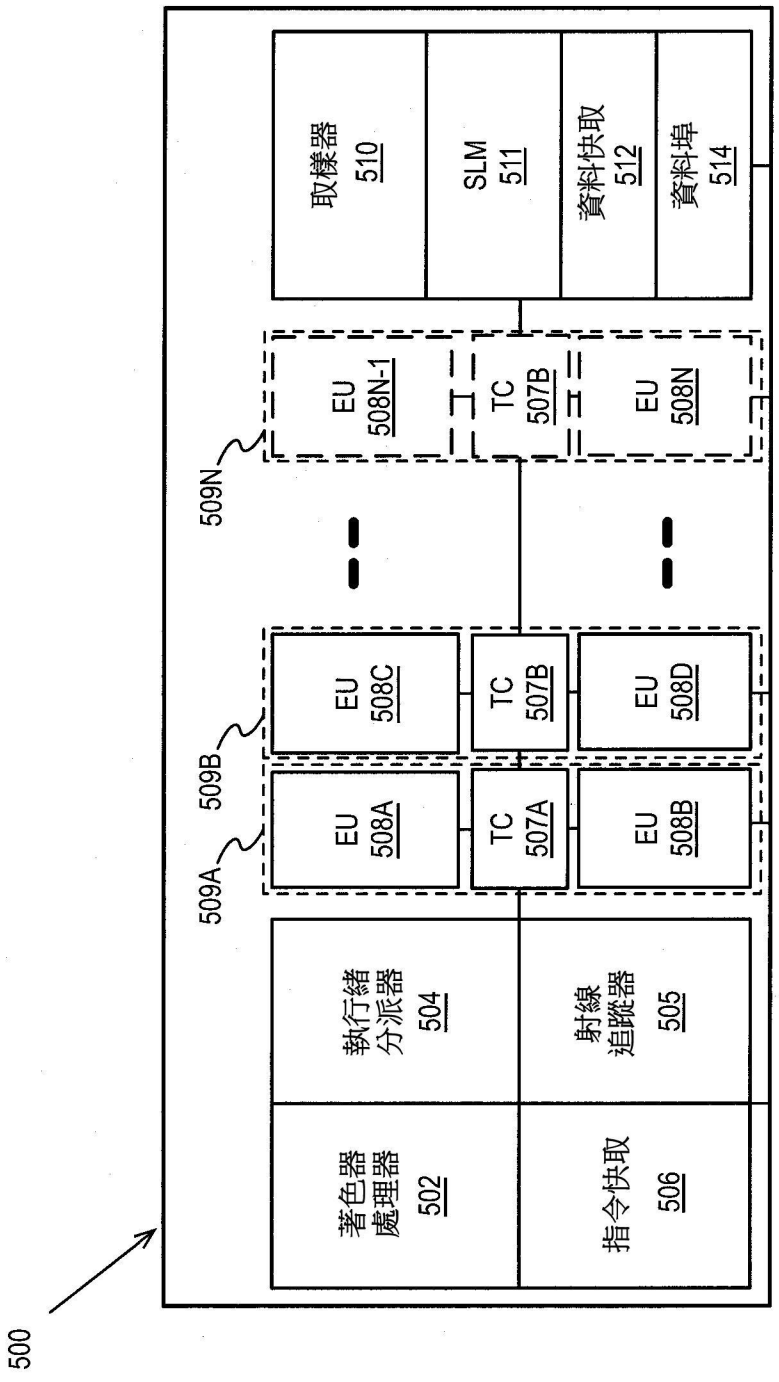
【圖 3B】


330
計算引擎集群
332
記憶體 326A
記憶體 326B
記憶體 326C
記憶體 326D
計算引擎塊 340A
計算引擎塊 340B
計算引擎塊 340C
計算引擎塊 340D
325A
325B
325C
325D
323A
323B
323C
323D
323E
323F
L3 快取 336
結構互連 -324
主機介面 -328

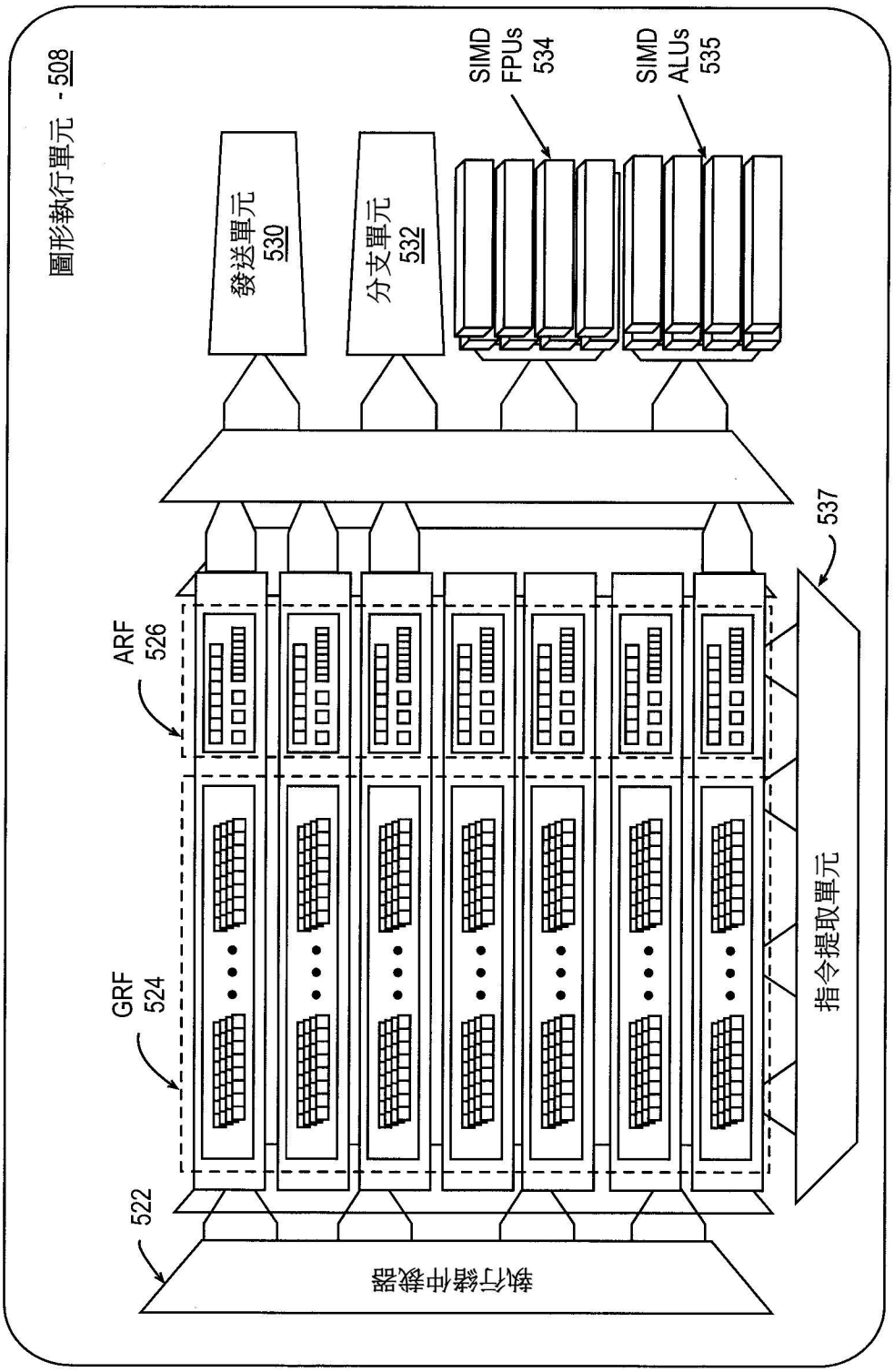
【圖 3C】


410
421
422
423
425
取樣器
數學邏輯
執行緒間通訊邏輯
...
快取
共享功能邏輯 420
圖形核心陣列 414
統一返回緩衝器 418
圖形核心 415A
共享功能邏輯 416
圖形核心 415B
3D 管線 312
媒體管線 316
命令串流器 403
來自記憶體

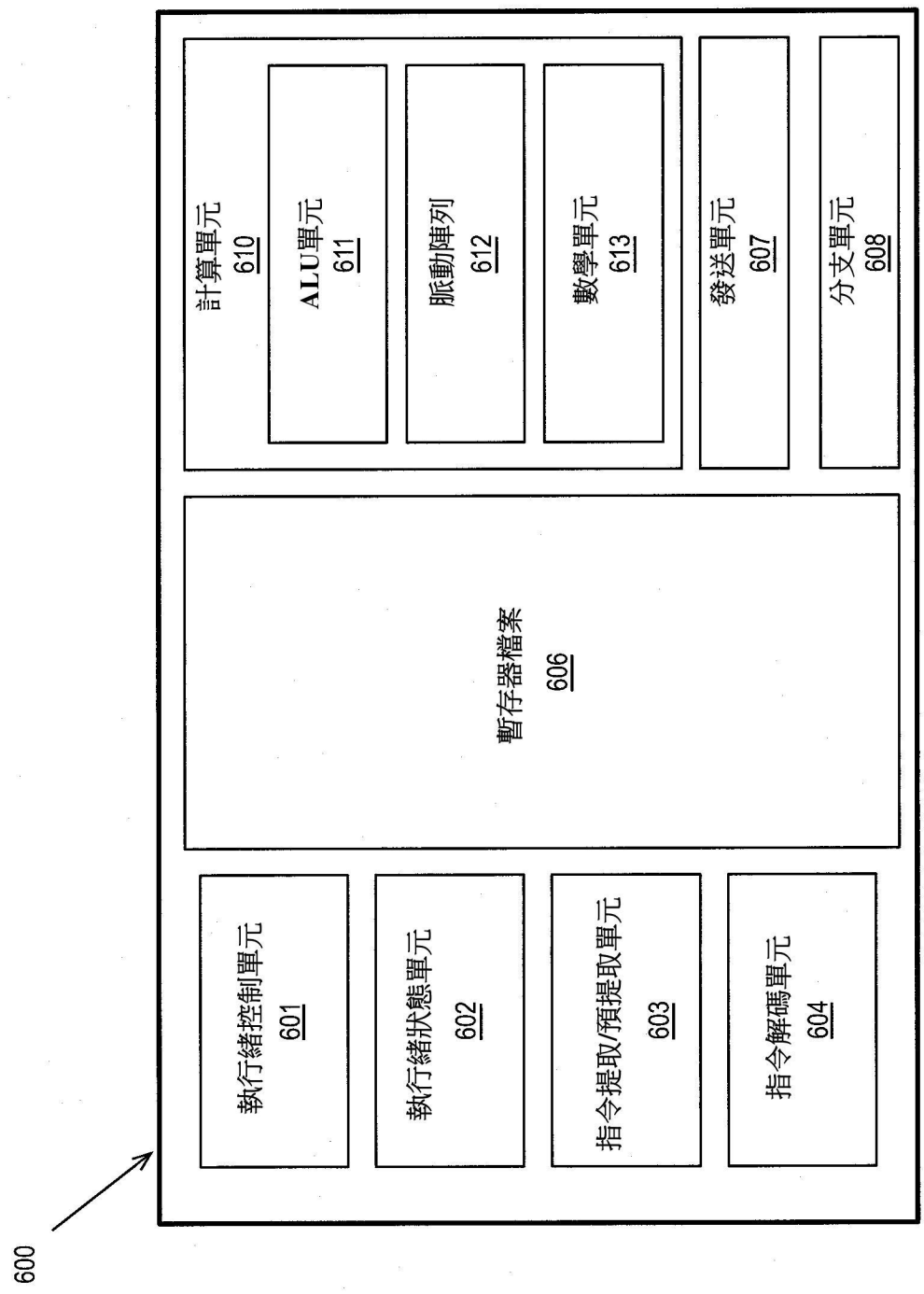
【圖 4】


500
著色器處理器 502
執行緒分派器 504
射線追蹤器 505
指令快取 506
509A
EU 508A
TC 507A
EU 508B
509B
EU 508C
TC 507B
EU 508D
509N
EU 508N-1
TC 507B
EU 508N
取樣器 510
SLM 511
資料快取 512
資料埠 514

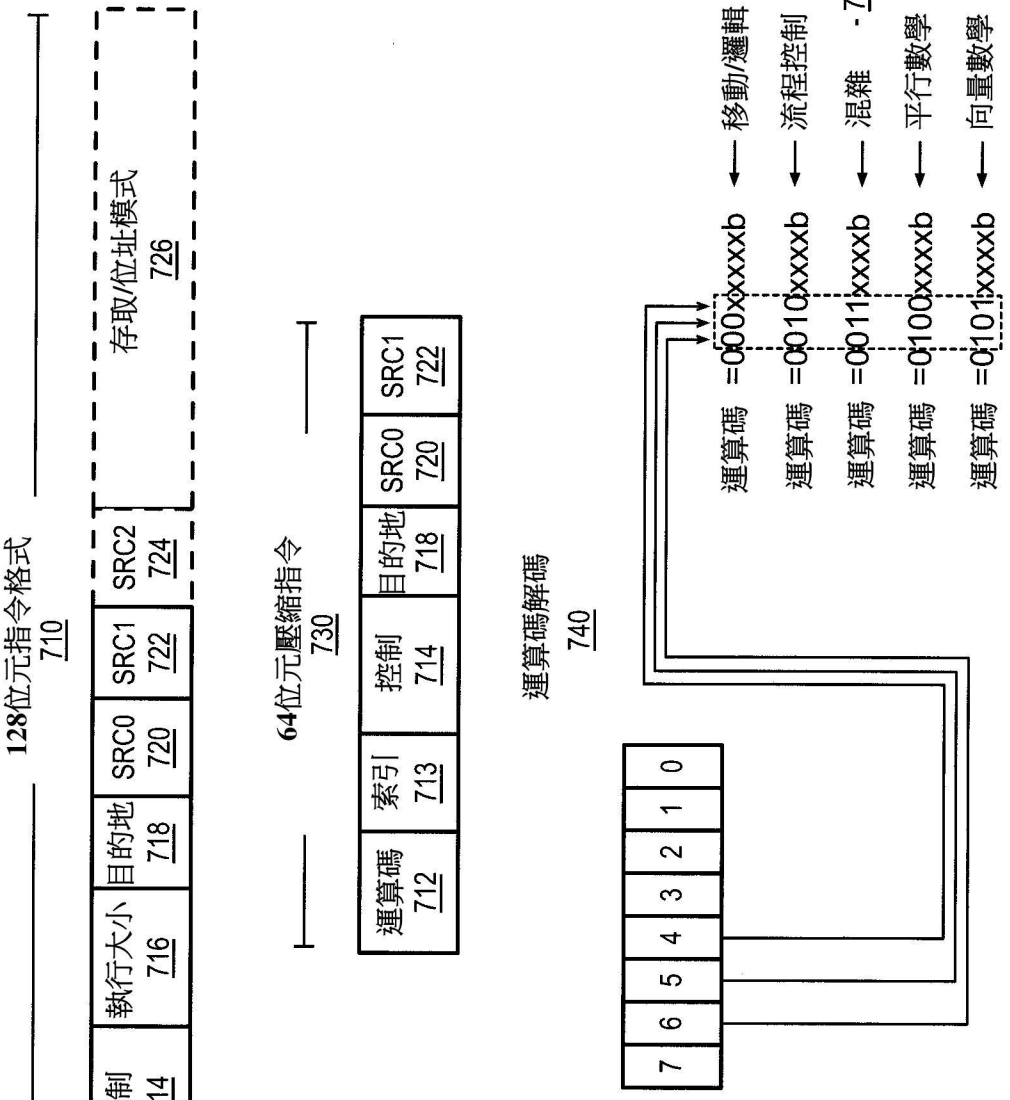
【圖 5A】


圖形執行單元 - 508
發送單元 530
分支單元 532
SIMD FPUs 534
SIMD ALUs 535
537
ARF 526
GRF 524
522
執行緒仲裁器
指令提取單元

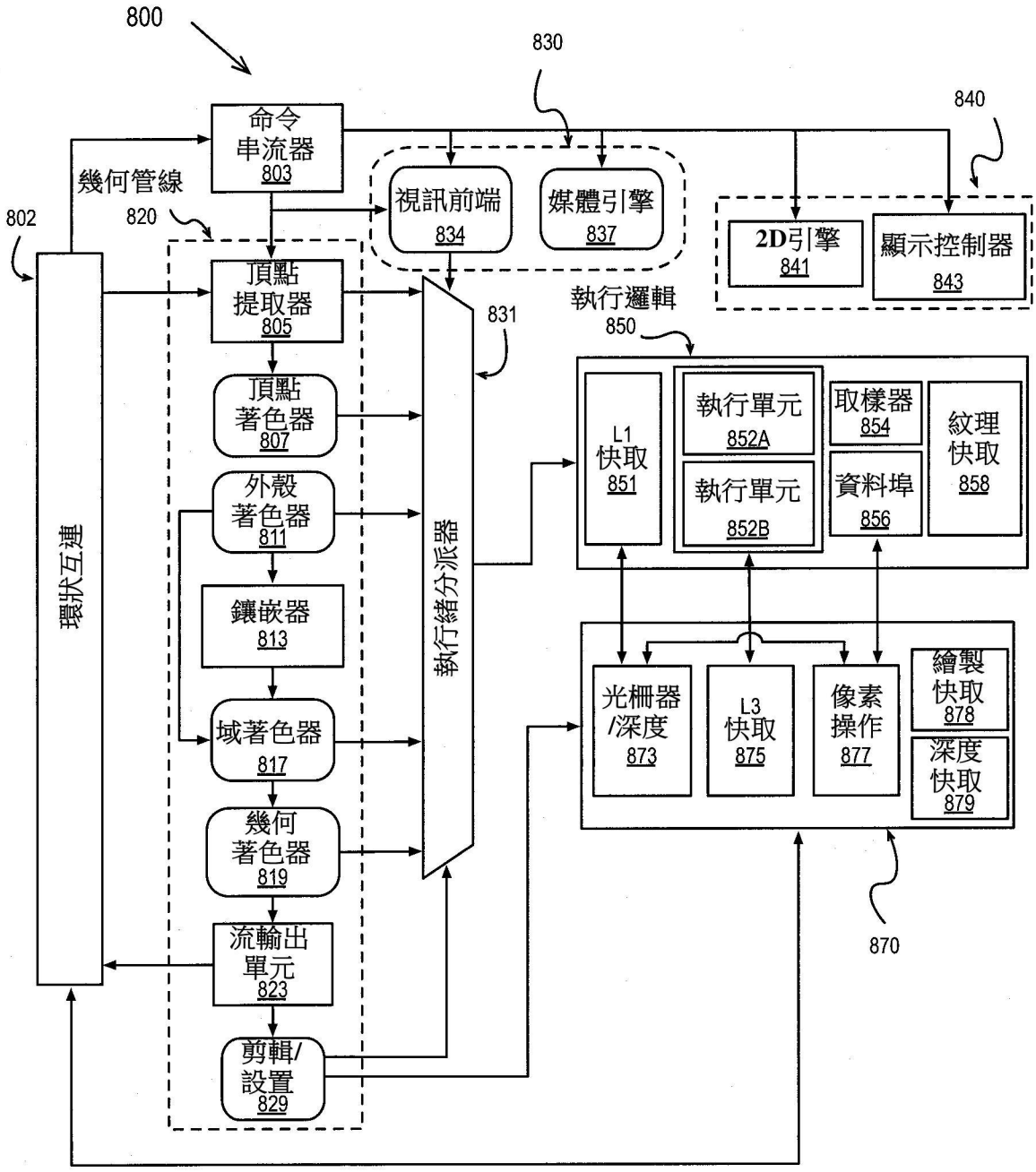
【圖 5B】


600
執行緒控制單元
601
執行緒狀態單元
602
指令提取/預提取單元
603
指令解碼單元
604
暫存器檔案
606
計算單元
610
ALU單元
611
脈動陣列
612
數學單元
613
發送單元
607
分支單元
608

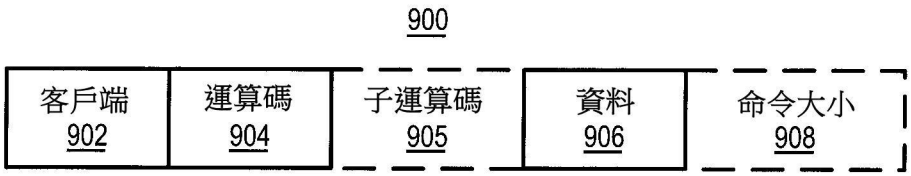
【圖 6】


700
128位元指令格式
710
運算碼 712
控制 714
執行大小 716
目的地 718
SRC0 720
SRC1 722
SRC2 724
存取/位址模式 726
64位元壓縮指令 730
運算碼 712
索引 713
控制 714
目的地 718
SRC0 720
SRC1 722
運算碼解碼 740
7 6 5 4 3 2 1 0
運算碼 =000xxxxxb
運算碼 =001xxxxxb
運算碼 =010xxxxxb
運算碼 =011xxxxxb
運算碼 =100xxxxxb
運算碼 =101xxxxxb
移動/邏輯 - 742
流程控制 - 744
混雜 - 746
平行數學 - 748
向量數學 - 750

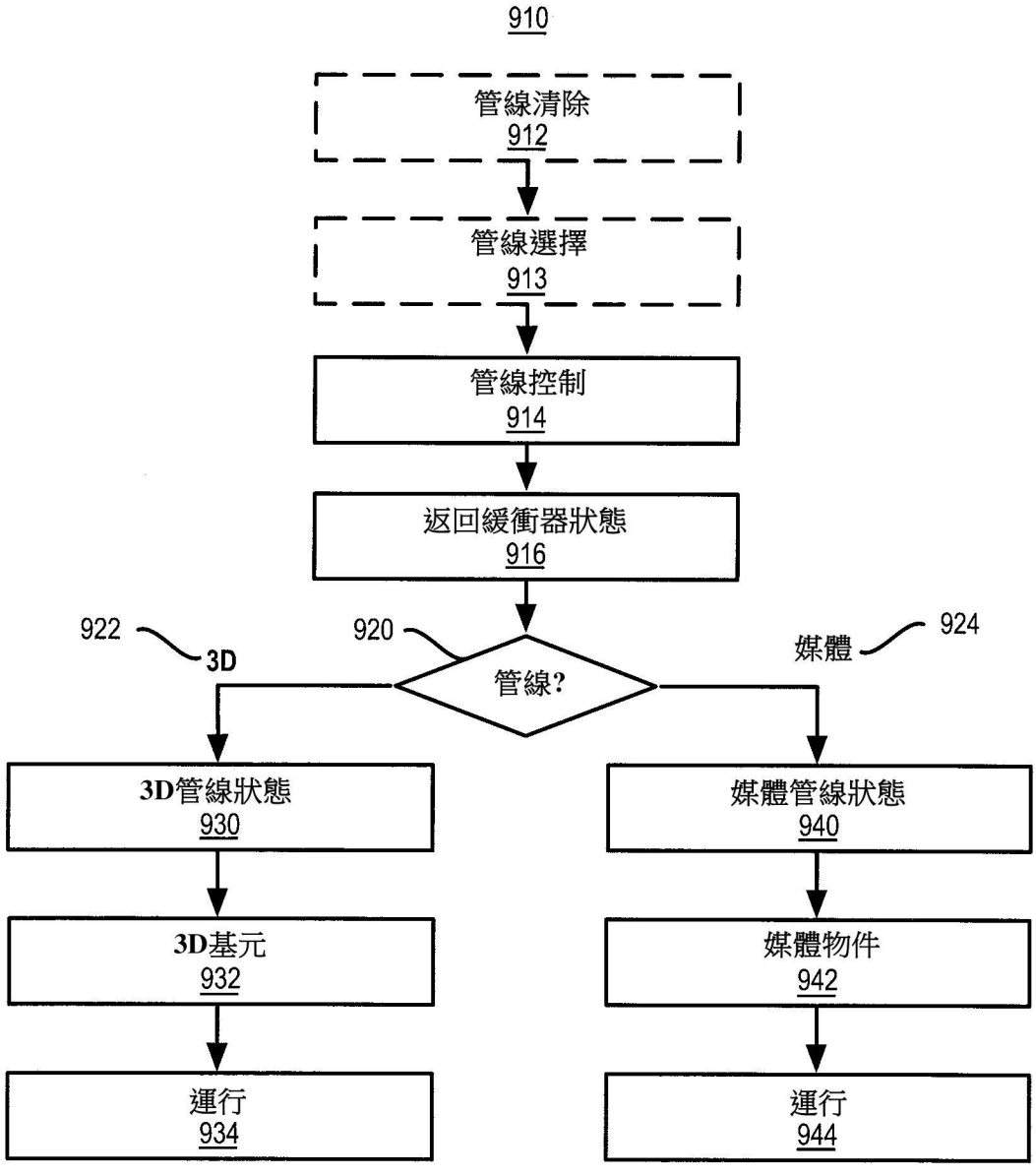
【圖 7】


800
830
840
命令
串流器
803
幾何管線
820
802
視訊前端
834
媒體引擎
837
2D引擎
841
顯示控制器
843
頂點
提取器
805
831
執行邏輯
850
頂點
著色器
807
外殼
著色器
811
鑲嵌器
813
域著色器
817
幾何
著色器
819
流輸出
單元
823
剪輯/
設置
829
環狀互連
執行緒分派器
L1
快取
851
執行單元
852A
執行單元
852B
取樣器
854
資料埠
856
紋理
快取
858
光柵器
/深度
873
L3
快取
875
像素
操作
877
繪製
快取
878
深度
快取
879
870

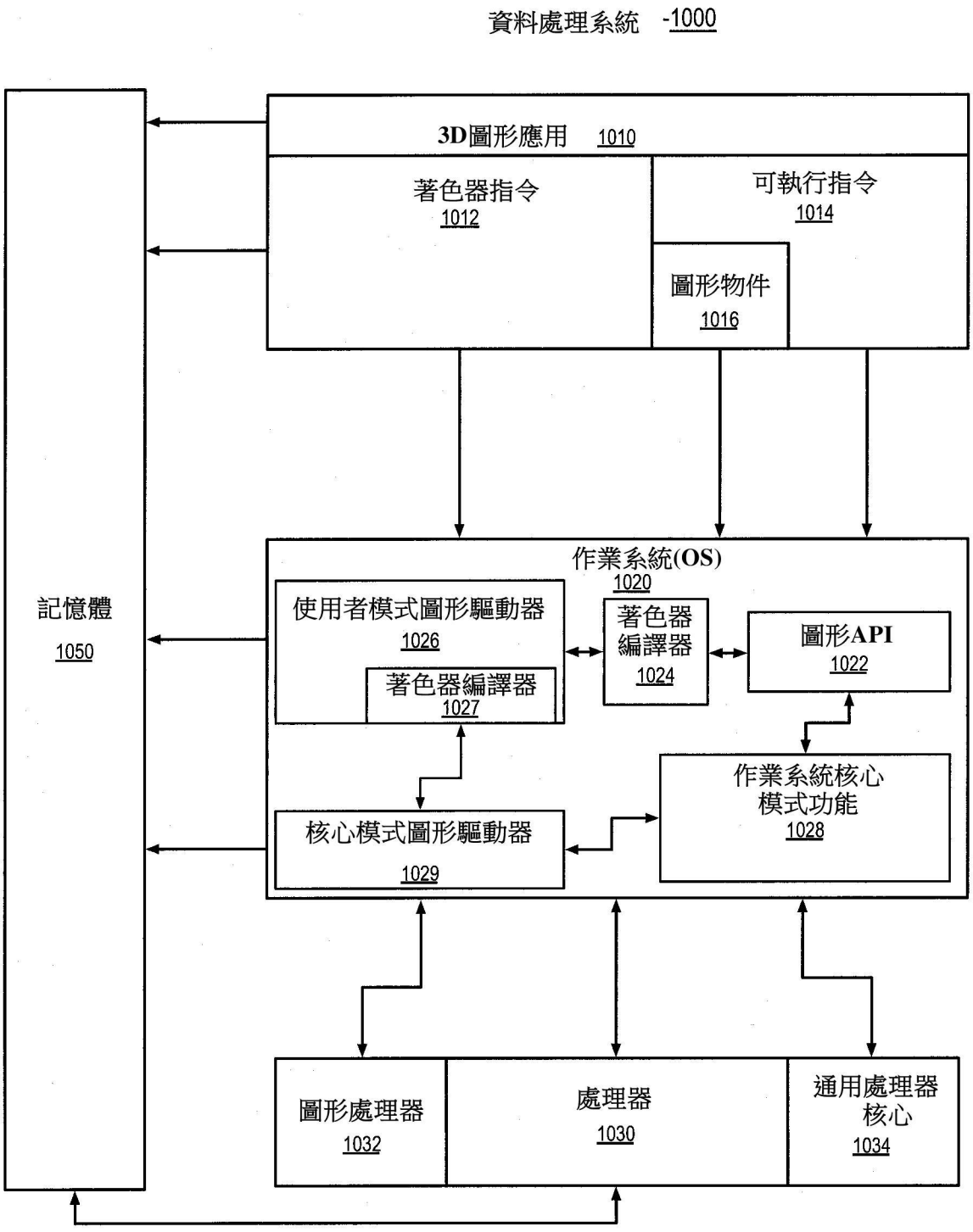
【圖 8】


900
客戶端
902
運算碼
904
子運算碼
905
資料
906
命令大小
908

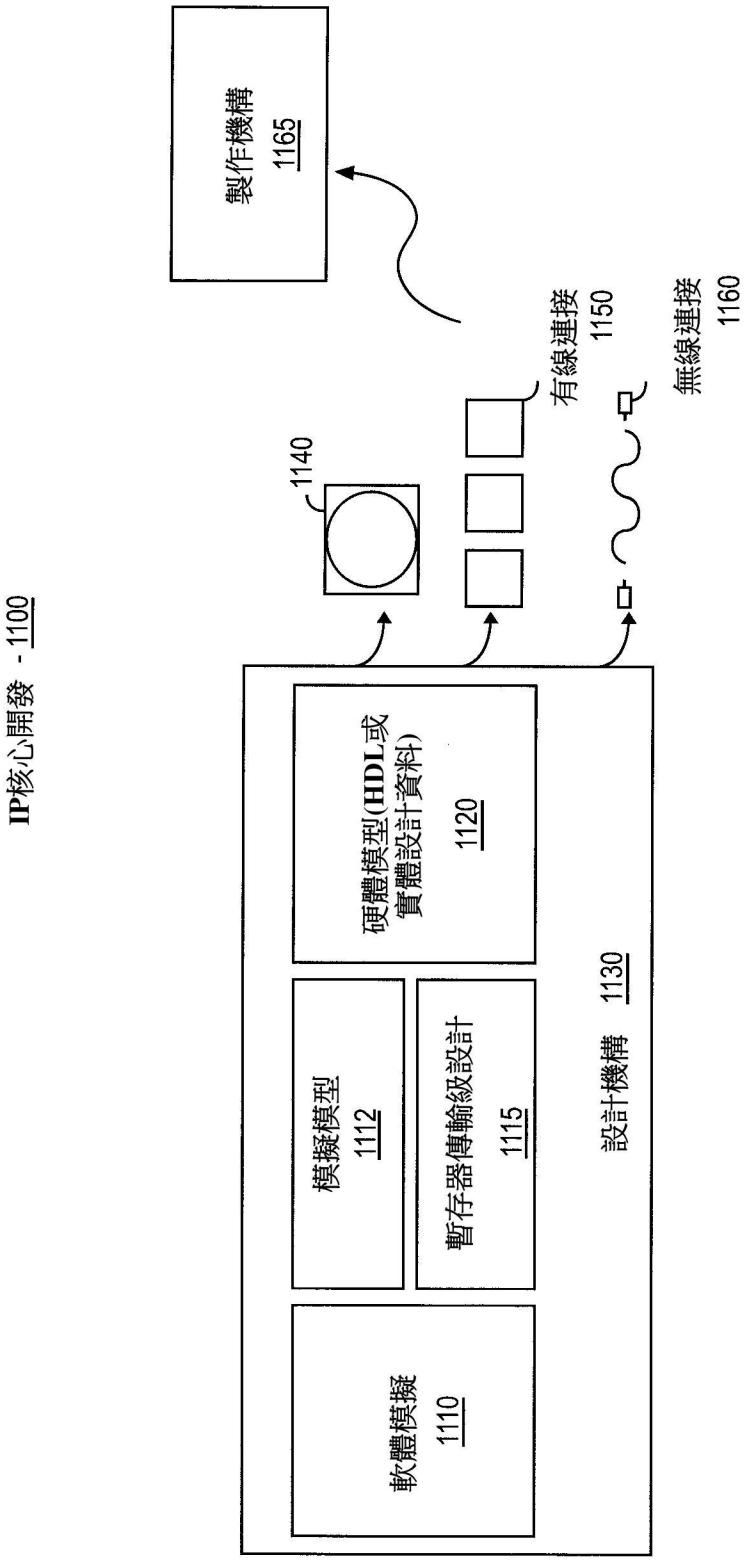
【圖 9A】


910
管線清除
912
管線選擇
913
管線控制
914
返回緩衝器狀態
916
922
3D
920
管線?
媒體
924
3D管線狀態
930
3D基元
932
運行
934
媒體管線狀態
940
媒體物件
942
運行
944

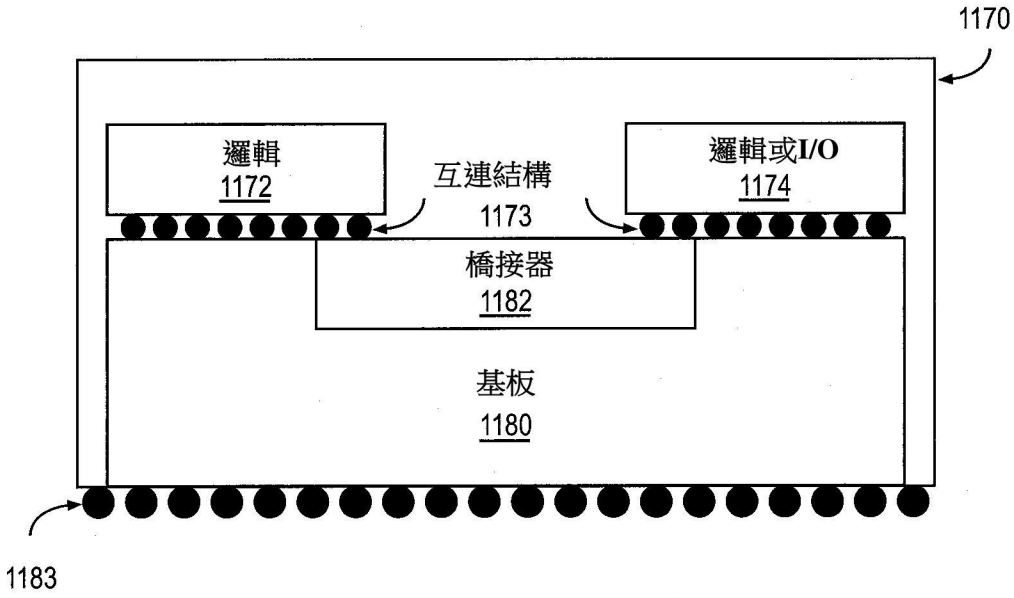
【圖 9B】


資料處理系統 -1000
3D圖形應用 1010
著色器指令 1012
可執行指令 1014
圖形物件 1016
記憶體 1050
作業系統(OS) 1020
使用者模式圖形驅動器 1026
著色器編譯器 1027
著色器編譯器 1024
圖形API 1022
作業系統核心模式功能 1028
核心模式圖形驅動器 1029
圖形處理器 1032
處理器 1030
通用處理器核心 1034

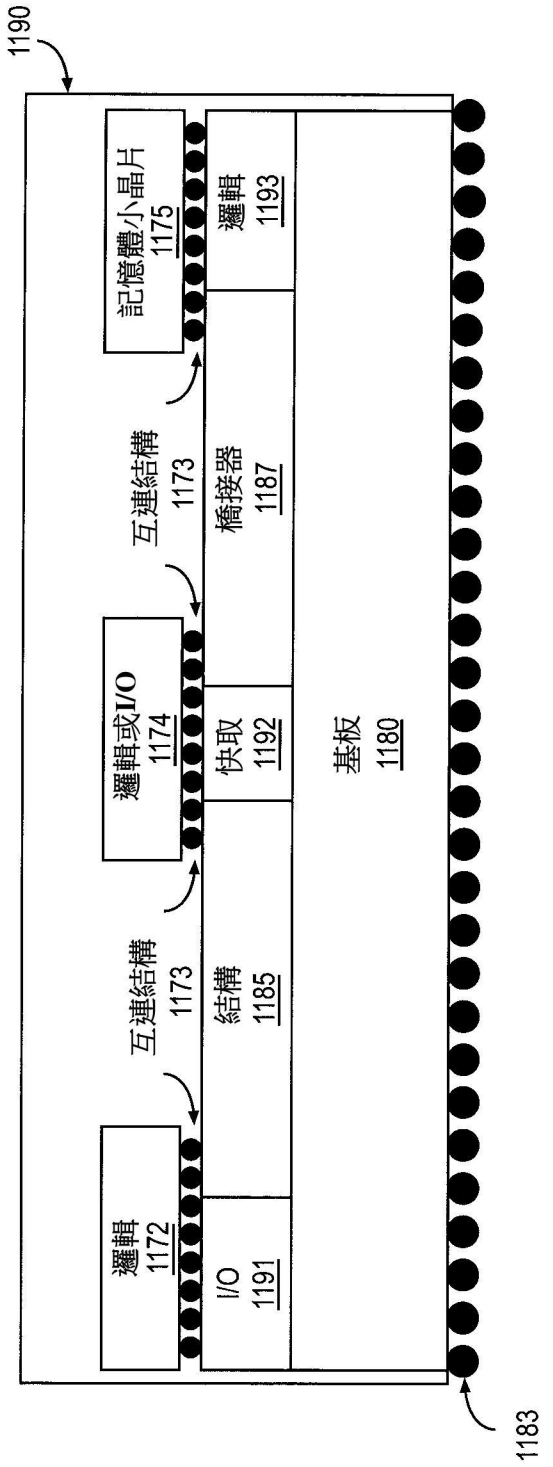
【圖 10】


IP核心開發 - 1100
製作機構
1165
1140
有線連接
1150
無線連接
1160
硬體模型(HDL或
實體設計資料)
1120
模擬模型
1112
暫存器傳輸級設計
1115
軟體模擬
1110
設計機構 1130

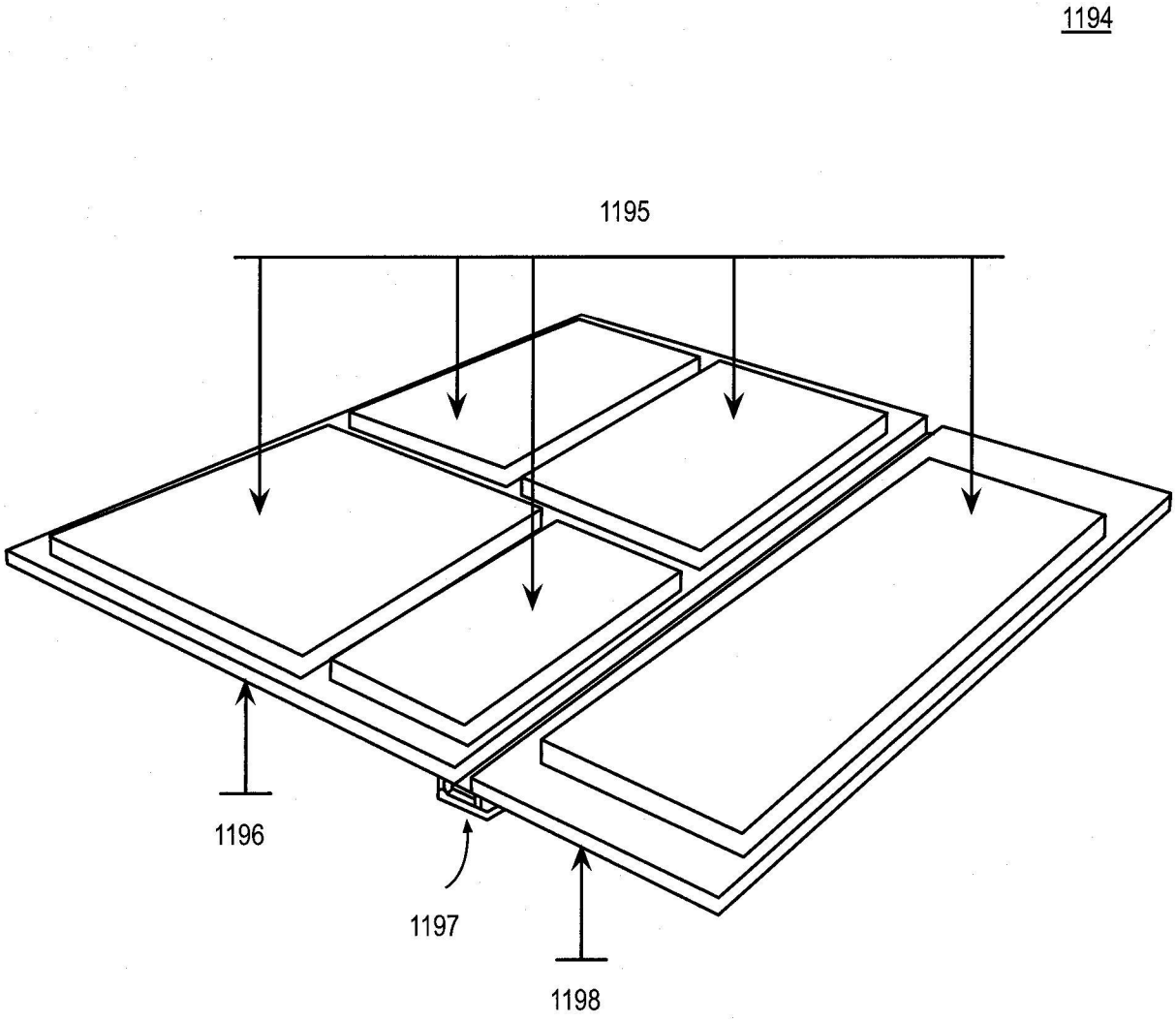
【圖 11A】



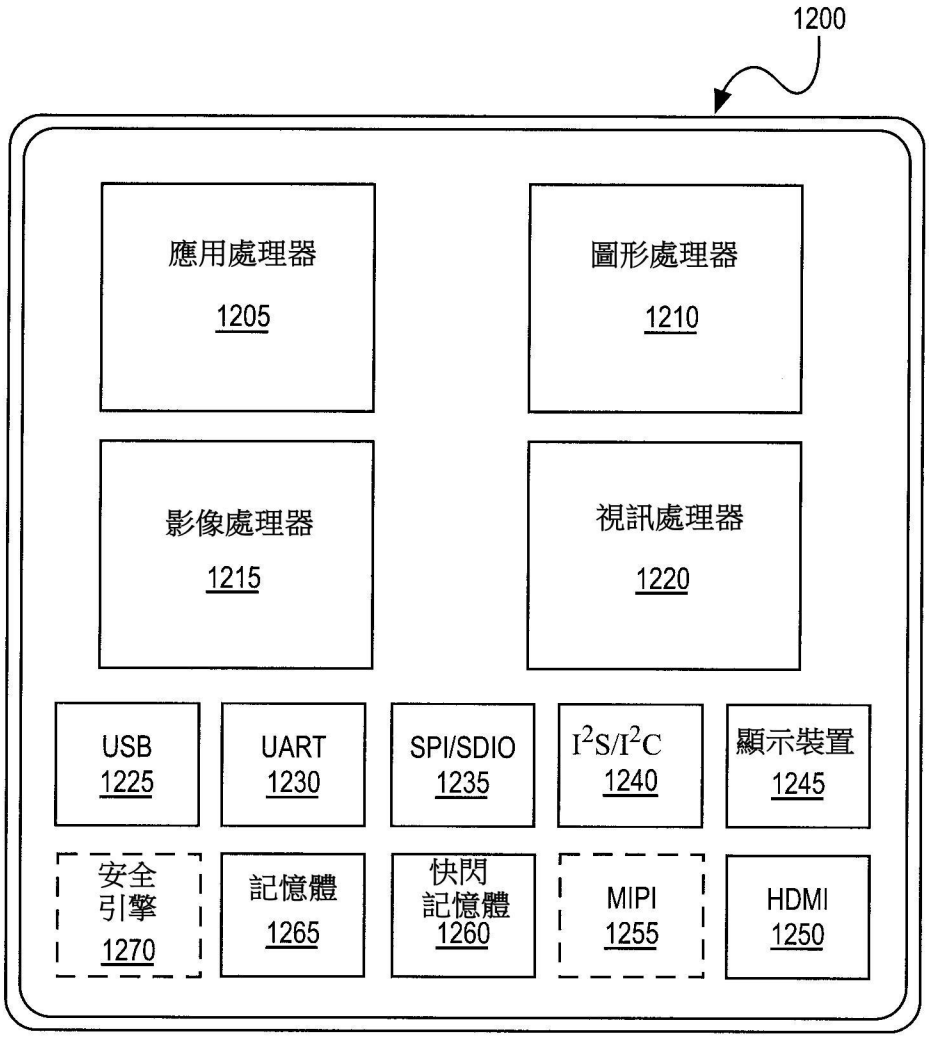
【圖 11B】


1190
記憶體小晶片
1175
邏輯
1193
互連結構
1173
橋接器
1187
邏輯或I/O
1174
快取
1192
基板
1180
互連結構
1173
結構
1185
邏輯
1172
I/O
1191
1183

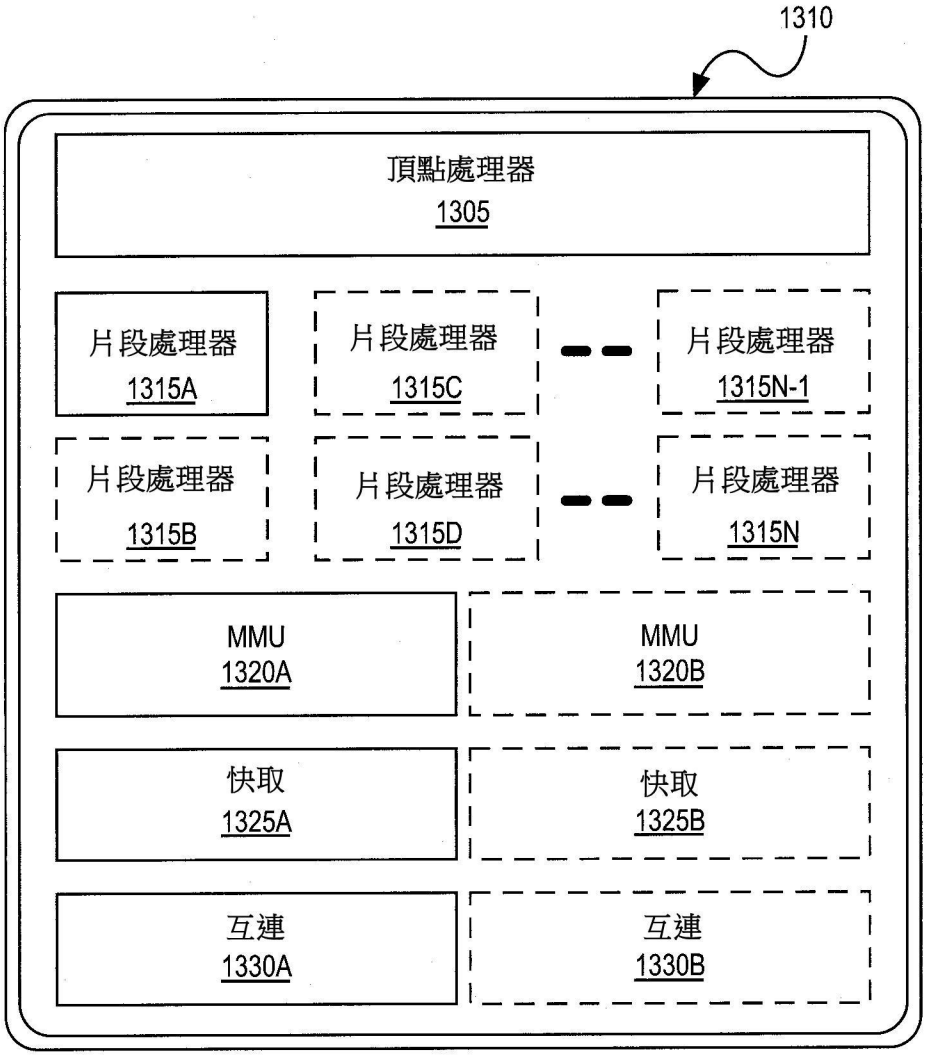
【圖 11C】



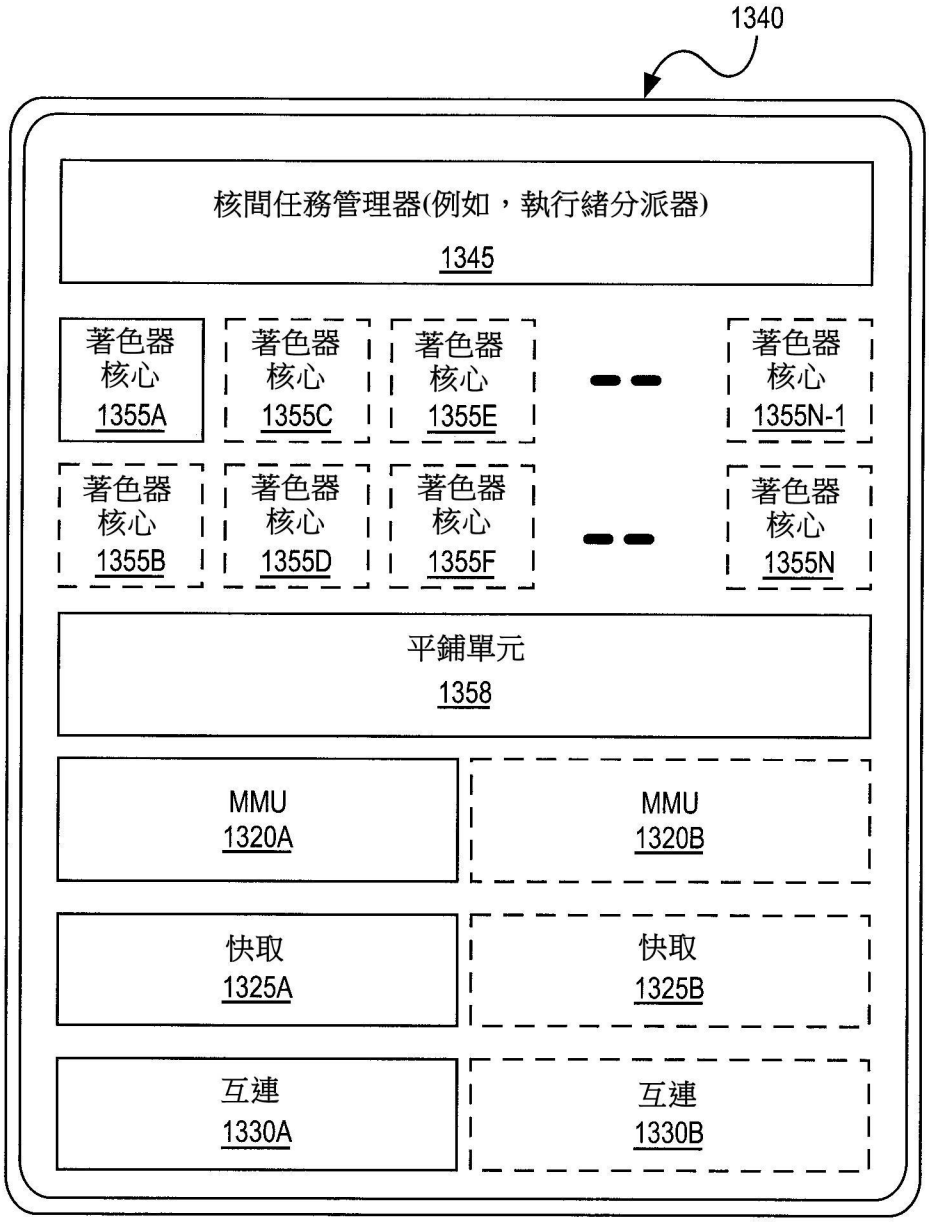
【圖 11D】


1200
應用處理器
1205
圖形處理器
1210
影像處理器
1215
視訊處理器
1220
USB
1225
UART
1230
SPI/SDIO
1235
I2S/I2C
1240
顯示裝置
1245
安全
引擎
1270
記憶體
1265
快閃
記憶體
1260
MIPI
1255
HDMI
1250

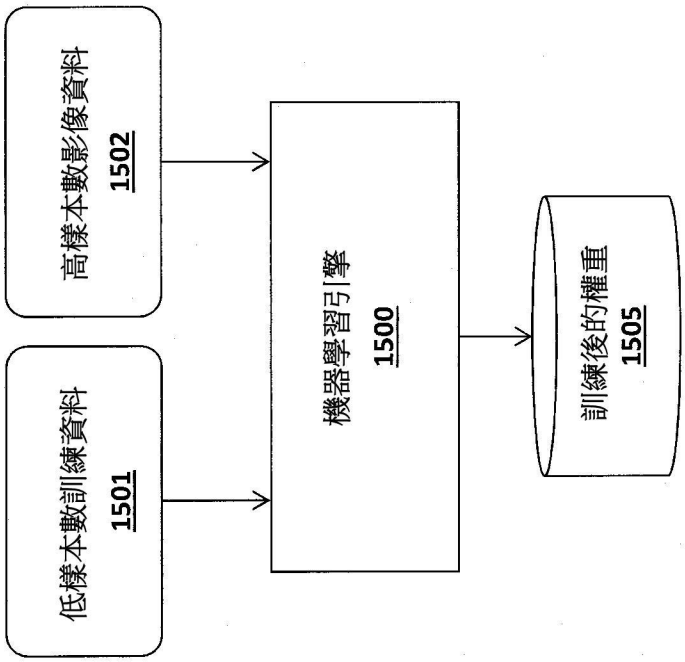
【圖 12】


1310
頂點處理器
1305
片段處理器
1315A
片段處理器
1315C
片段處理器
1315N-1
片段處理器
1315B
片段處理器
1315D
片段處理器
1315N
MMU
1320A
MMU
1320B
快取
1325A
快取
1325B
互連
1330A
互連
1330B

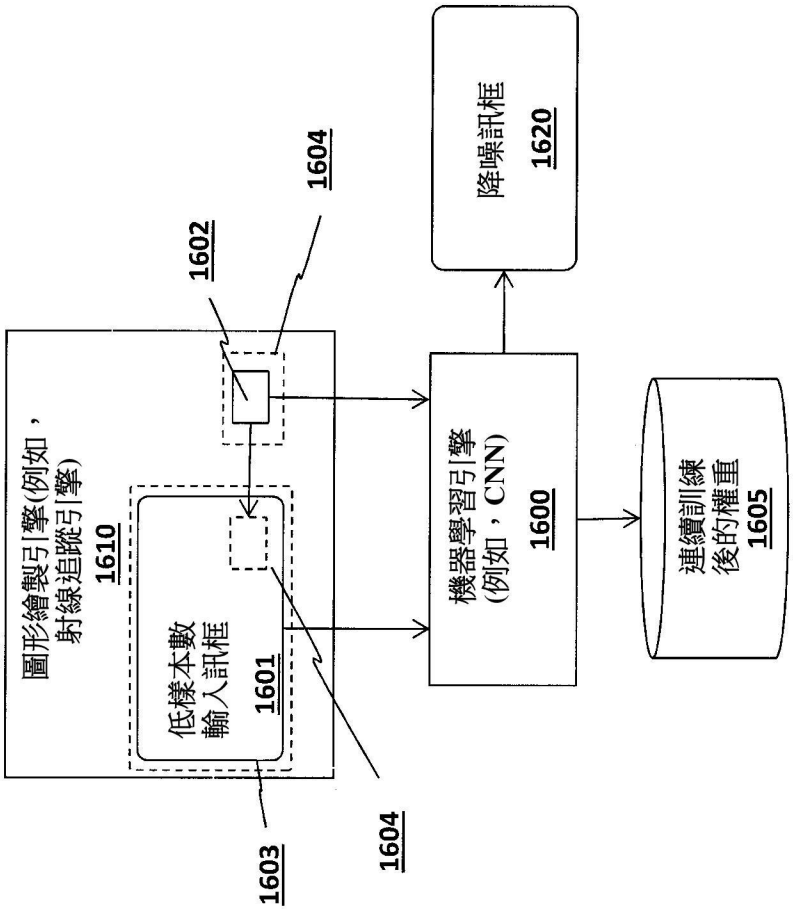
【圖 13】


1340
核間任務管理器(例如，執行緒分派器)
1345
著色器核心 1355A
著色器核心 1355C
著色器核心 1355E
著色器核心 1355N-1
著色器核心 1355B
著色器核心 1355D
著色器核心 1355F
著色器核心 1355N
平鋪單元
1358
MMU 1320A
MMU 1320B
快取 1325A
快取 1325B
互連 1330A
互連 1330B

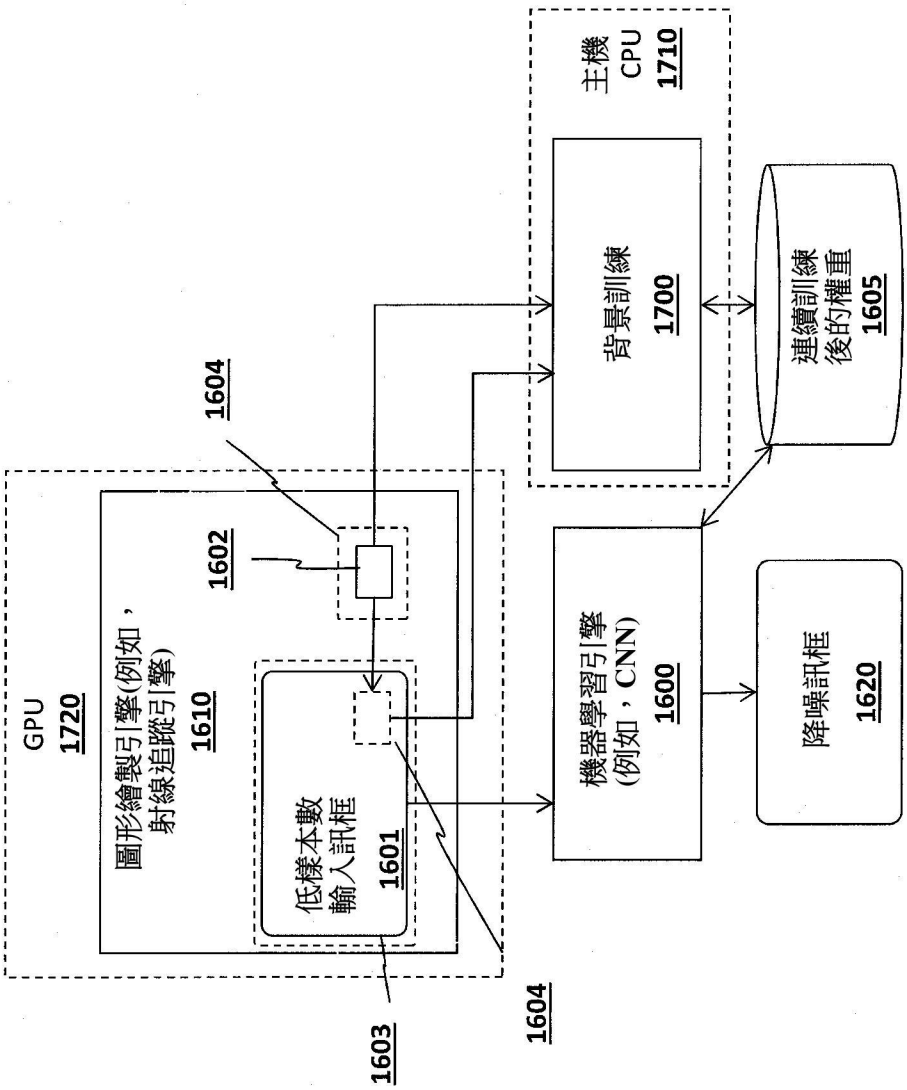
【圖 14】



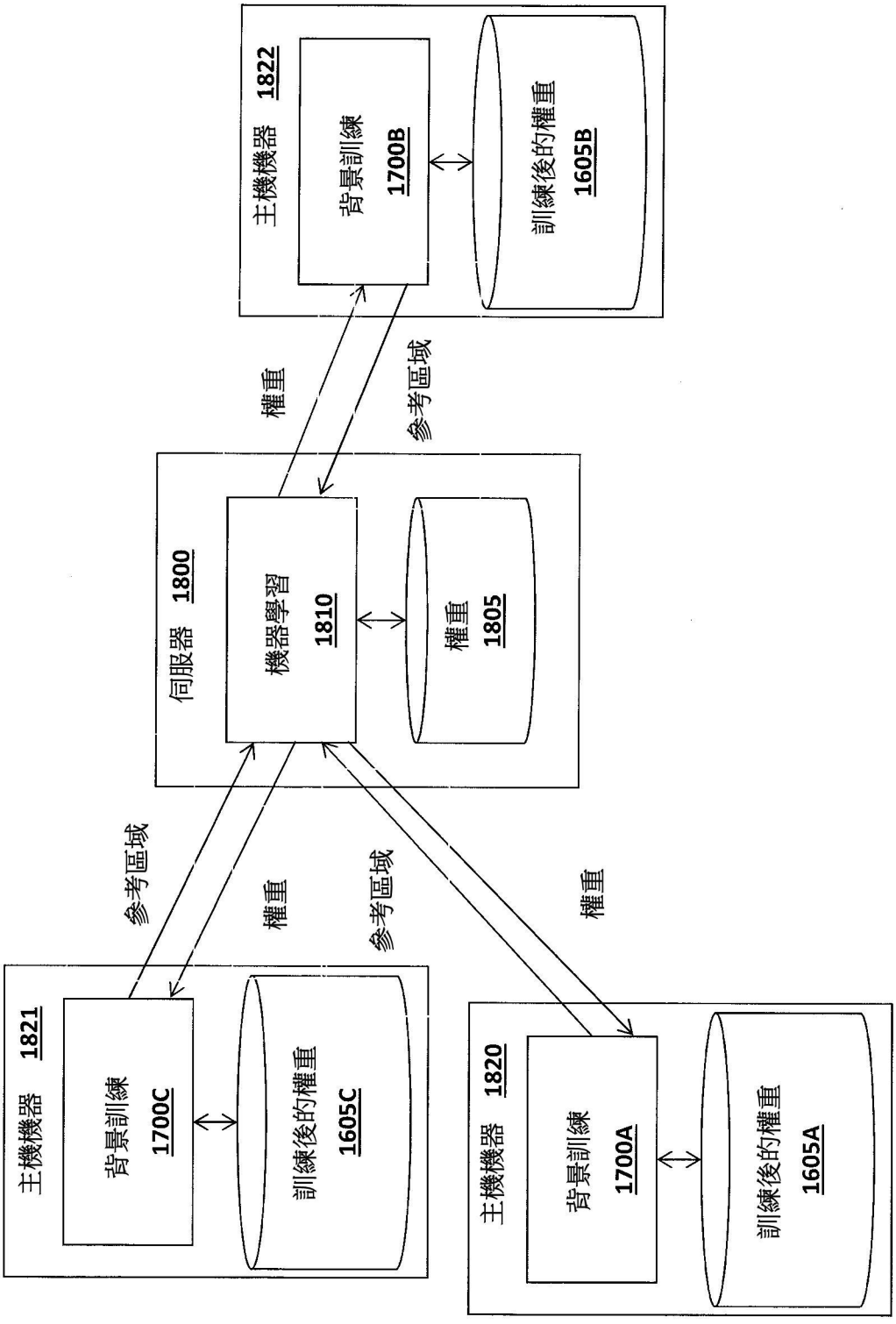
【圖 15】


圖形繪製引擎(例如，
射線追蹤引擎)
1610
低樣本數
輸入訊框
1601
1602
1604
1603
1604
機器學習引擎
(例如，CNN)
1600
降噪訊框
1620
連續訓練
後的權重
1605

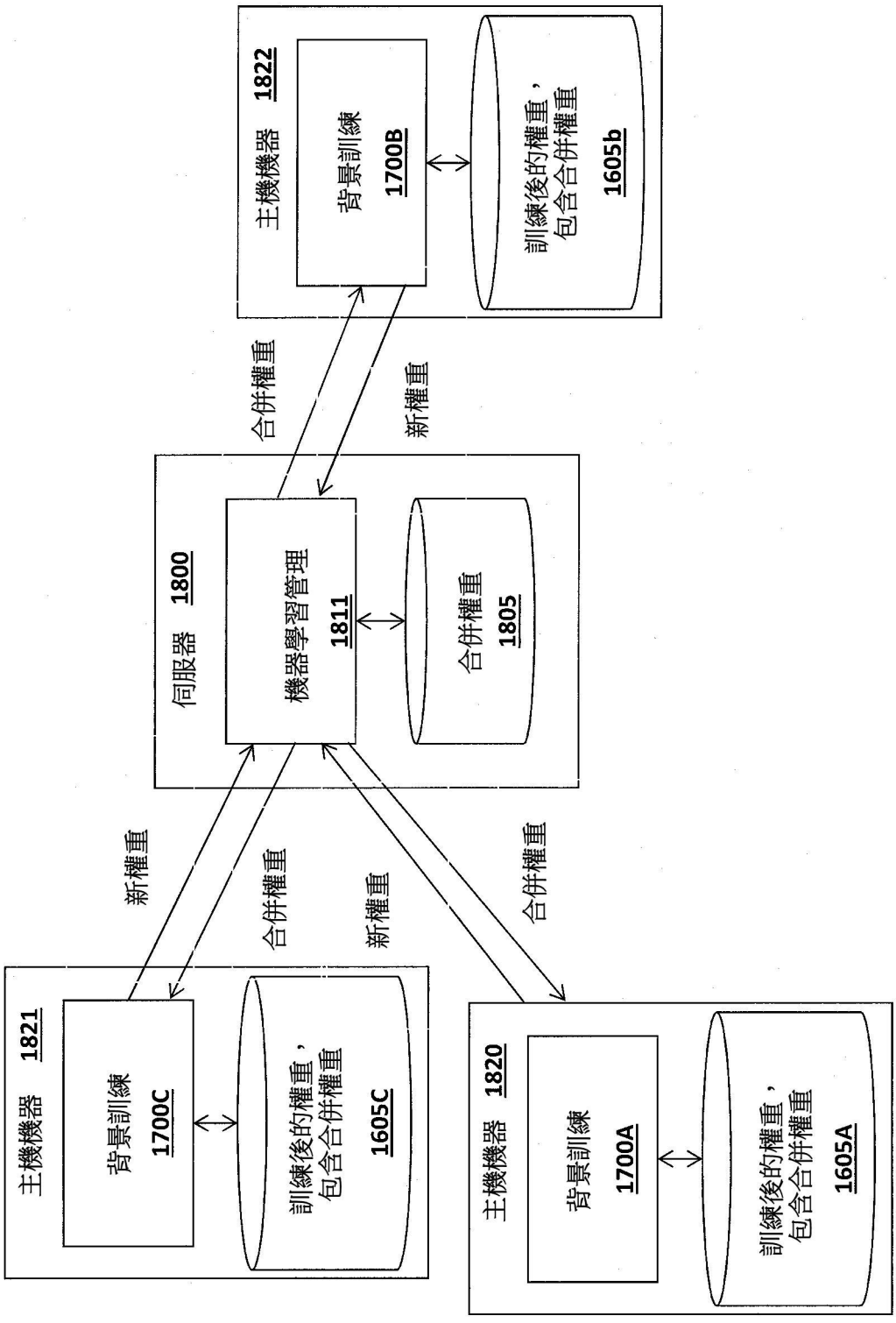
【圖 16】


GPU
1720
圖形繪製引擎(例如，
射線追蹤引擎)
1610
1602
1604
低樣本數
輸入訊框
1601
1603
1604
主機
CPU
1710
背景訓練
1700
連續訓練
後的權重
1605
機器學習引擎
(例如，CNN)
1600
降噪訊框
1620

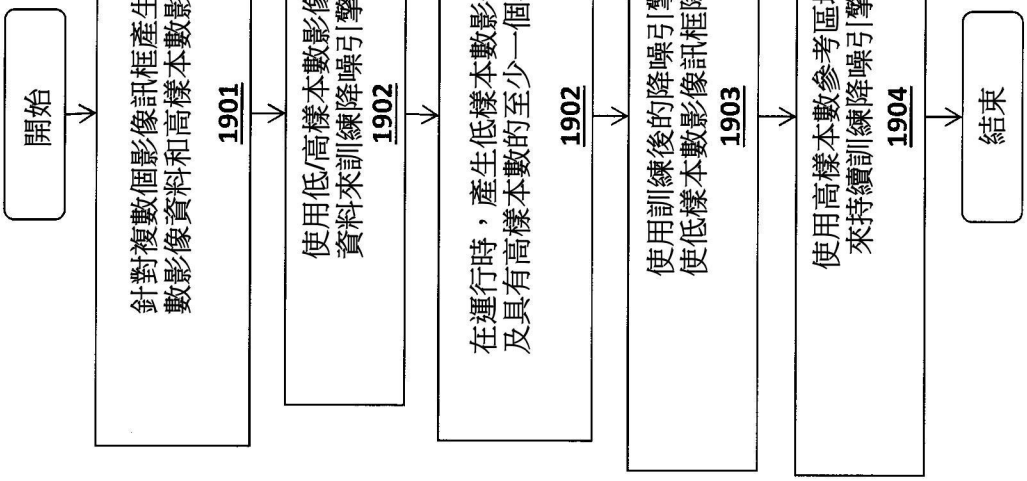
【圖 17】


主機機器 1822
背景訓練 1700B
訓練後的權重 1605B
權重
參考區域
伺服器 1800
機器學習 1810
權重 1805
參考區域
權重
參考區域
權重
主機機器 1821
背景訓練 1700C
訓練後的權重 1605C
主機機器 1820
背景訓練 1700A
訓練後的權重 1605A

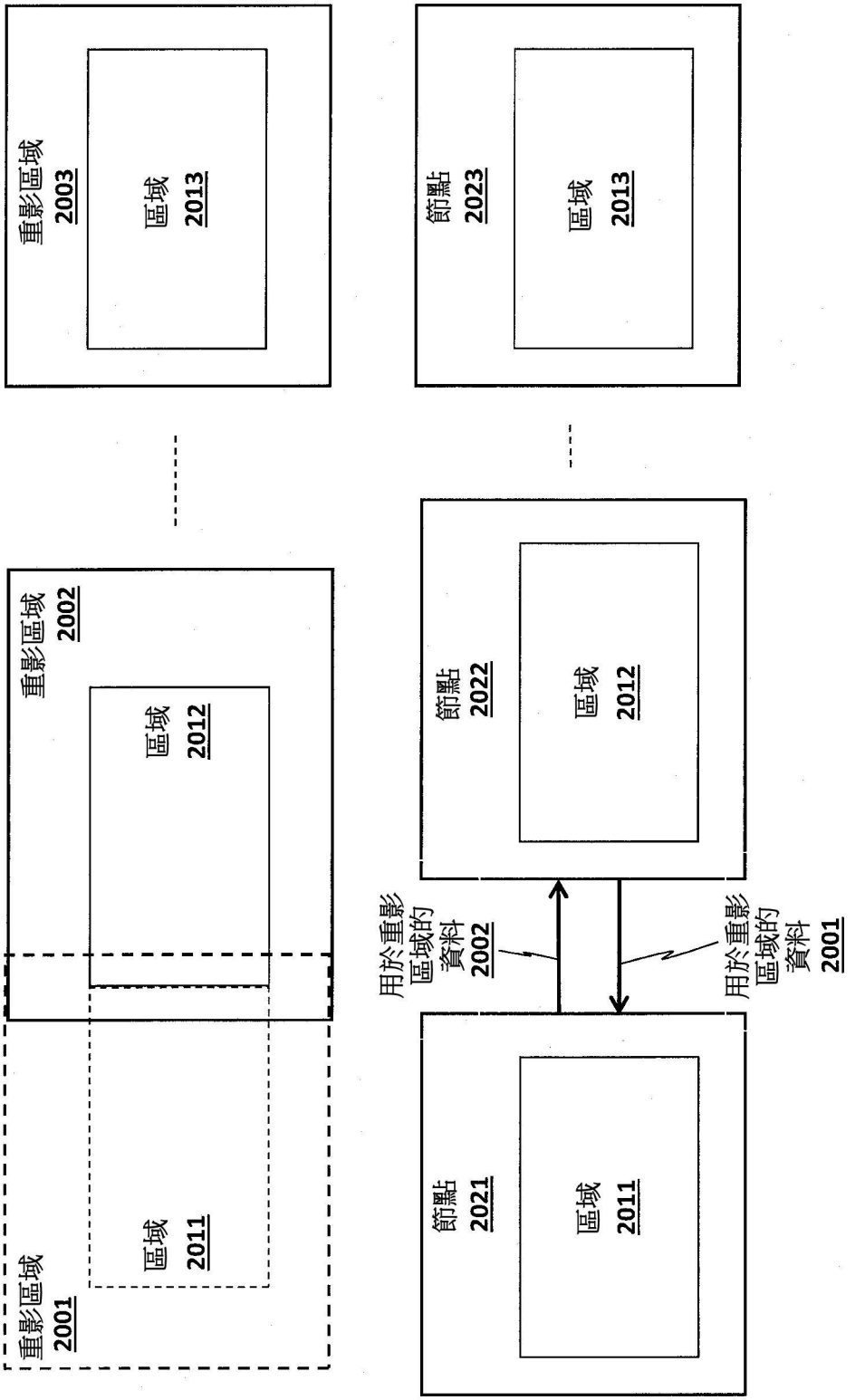
【圖 18A】


主機機器 1822
背景訓練 1700B
訓練後的權重，包含合併權重 1605b
合併權重
新權重
伺服器 1800
機器學習管理 1811
合併權重 1805
新權重
合併權重
新權重
合併權重
主機機器 1821
背景訓練 1700C
訓練後的權重，包含合併權重 1605C
主機機器 1820
背景訓練 1700A
訓練後的權重，包含合併權重 1605A

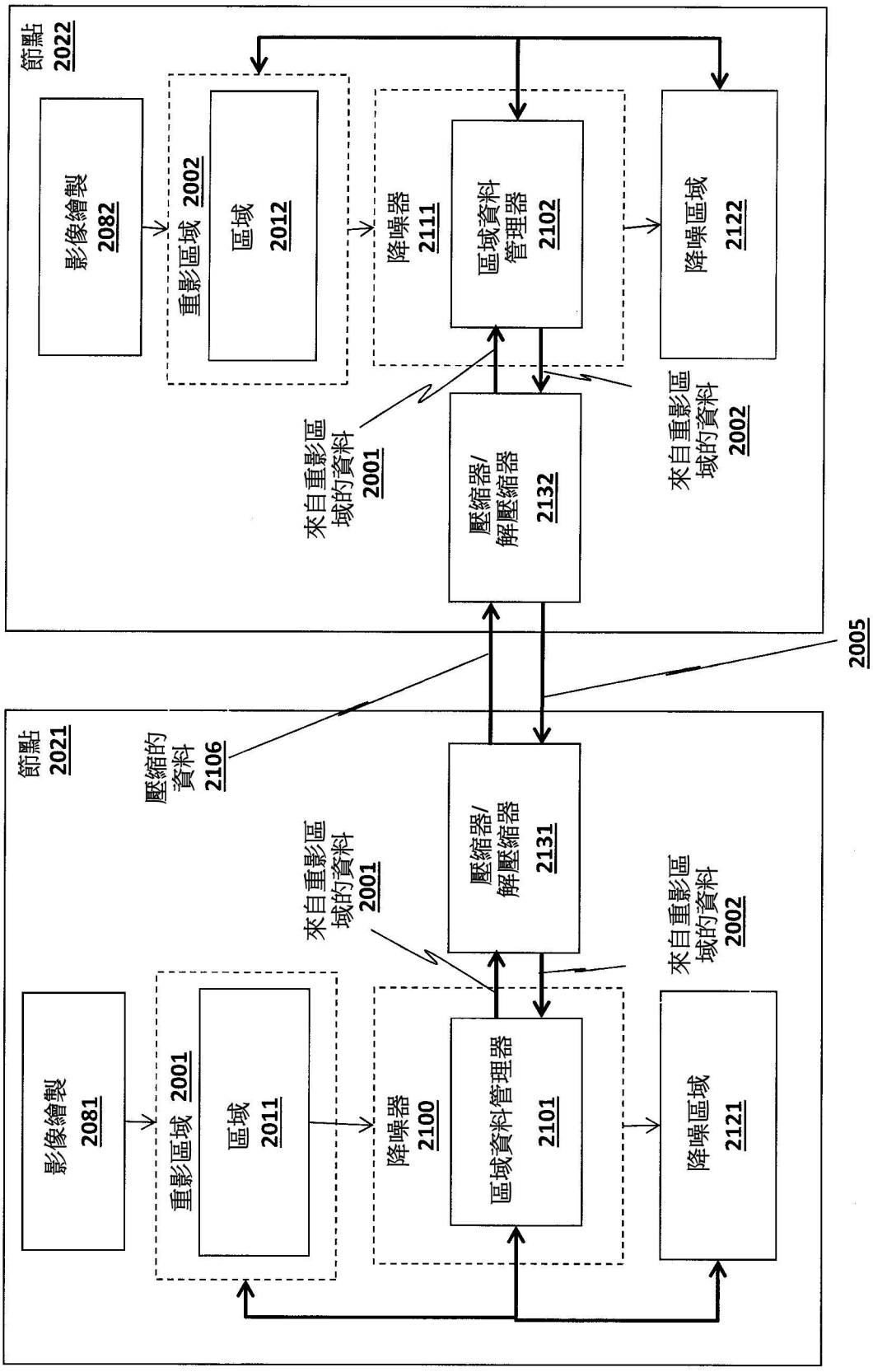
【圖 18B】


開始
針對複數個影像訊框產生低樣本數影像資料和高樣本數影像資料
1901
使用低/高樣本數影像資料來訓練降噪引擎
1902
在運行時，產生低樣本數影像訊框以及具有高樣本數的至少一個參考區域
1902
使用訓練後的降噪引擎來使低樣本數影像訊框降噪
1903
使用高樣本數參考區域來持續訓練降噪引擎
1904
結束

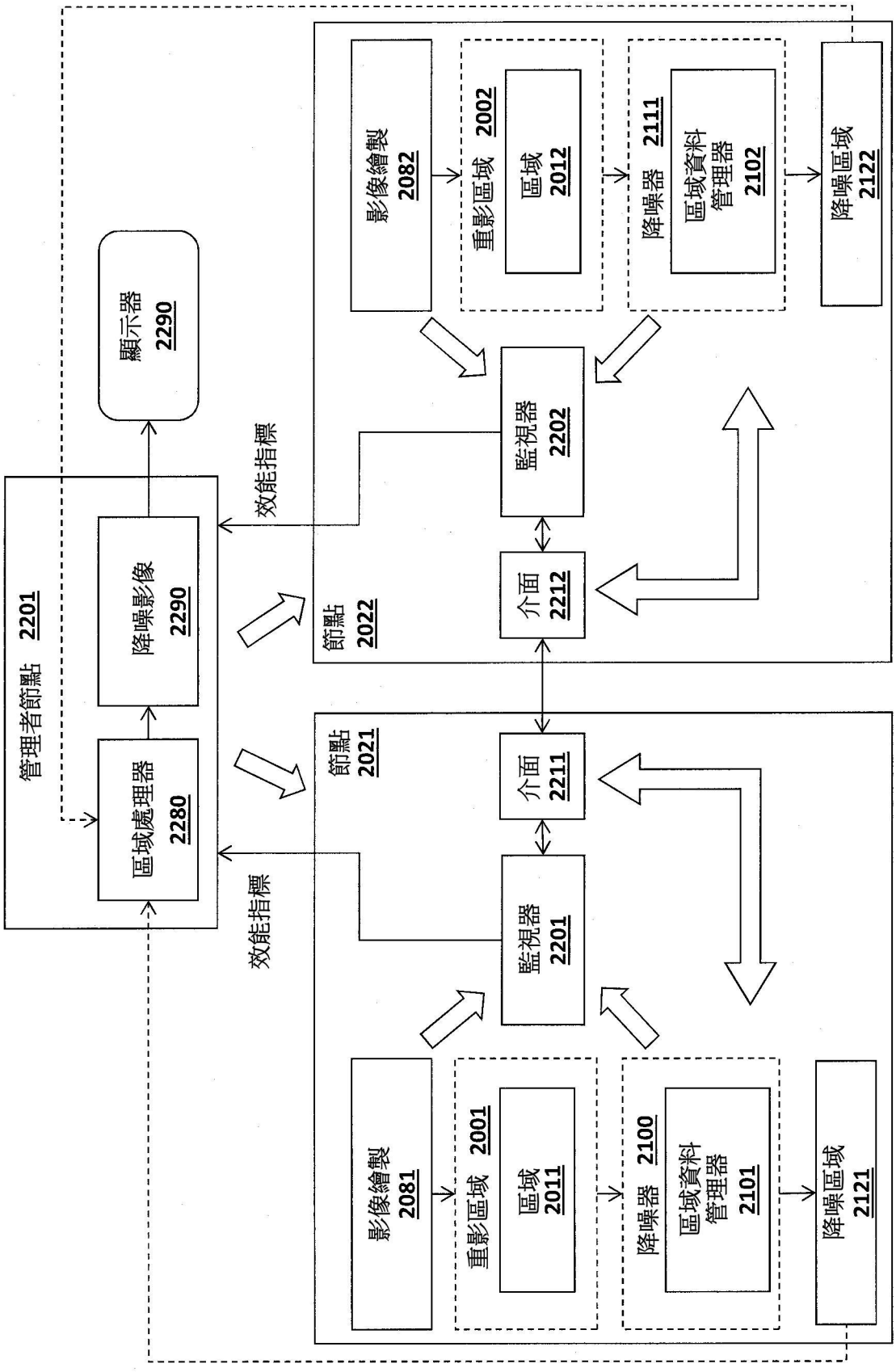
【圖 19】


重影區域
2003
區域
2013
節點
2023
區域
2013
重影區域
2002
區域
2012
節點
2022
區域
2012
用於重影區域的資料
2002
用於重影區域的資料
2001
重影區域
2001
區域
2011
節點
2021
區域
2011

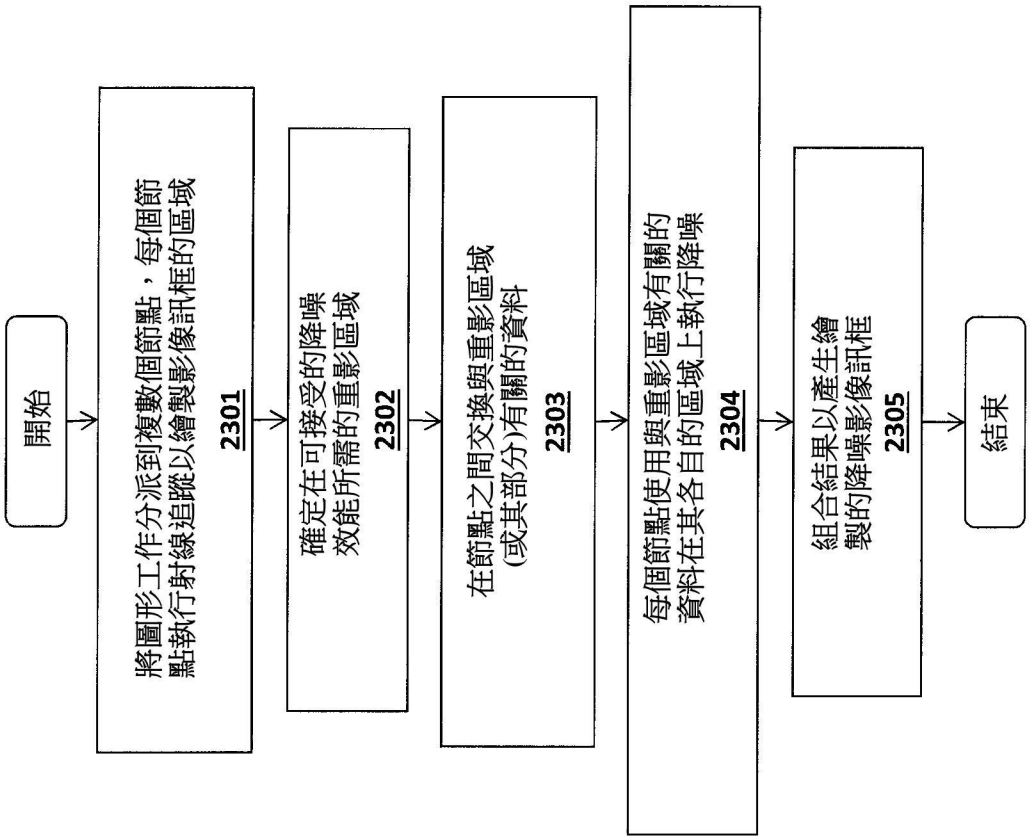
【圖 20】


節點 2022
影像繪製 2082
重影區域 2002
區域 2012
降噪器 2111
區域資料管理器 2102
降噪區域 2122
來自重影區域的資料 2001
壓縮器/解壓縮器 2132
來自重影區域的資料 2002
2005
節點 2021
壓縮的資料 2106
來自重影區域的資料 2001
壓縮器/解壓縮器 2131
來自重影區域的資料 2002
影像繪製 2081
重影區域 2001
區域 2011
降噪器 2100
區域資料管理器 2101
降噪區域 2121

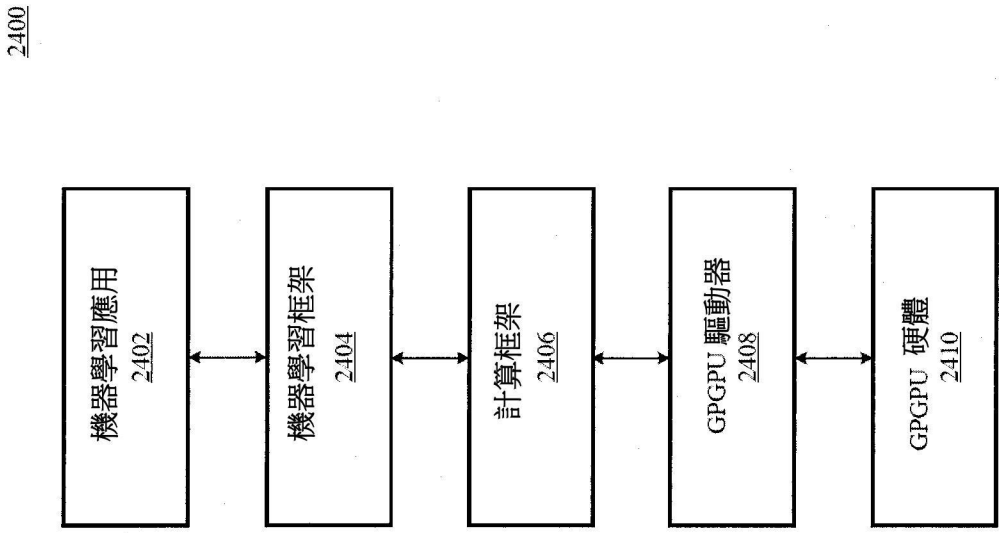
【圖 21】


管理者節點 2201
區域處理器 2280
降噪影像 2290
顯示器 2290
效能指標
效能指標
節點 2021
介面 2211
監視器 2201
影像繪製 2081
重影區域 2001
區域 2011
降噪器 2100
區域資料管理器 2101
降噪區域 2121
節點 2022
介面 2212
監視器 2202
影像繪製 2082
重影區域 2002
區域 2012
降噪器 2111
區域資料管理器 2102
降噪區域 2122

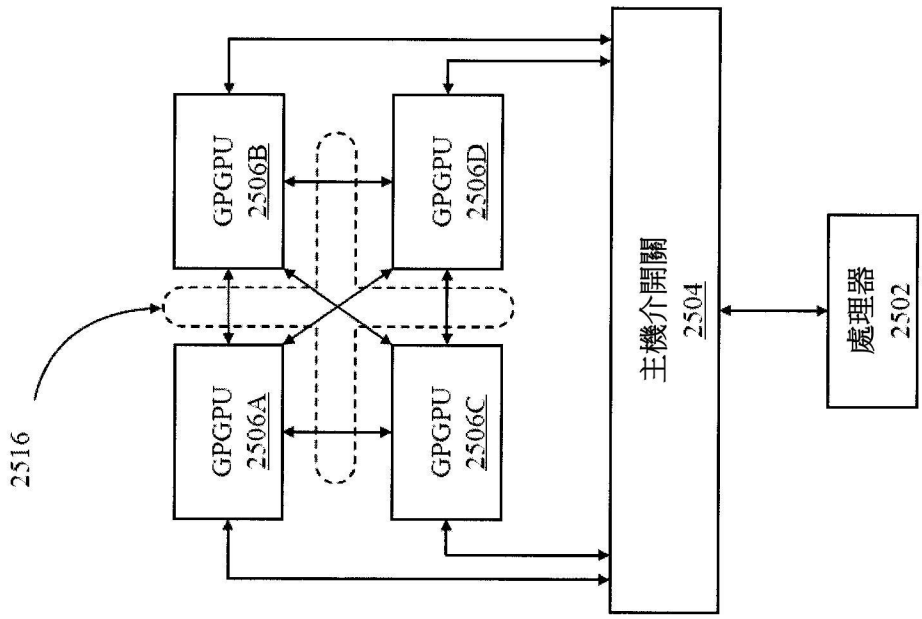
【圖 22】


開始
將圖形工作分派到複數個節點，每個節點執行射線追蹤以繪製影像訊框的區域
2301
確定在可接受的降噪效能所需的重影區域
2302
在節點之間交換與重影區域(或其部分)有關的資料
2303
每個節點使用與重影區域有關的資料在其各自的區域上執行降噪
2304
組合結果以產生繪製的降噪影像訊框
2305
結束

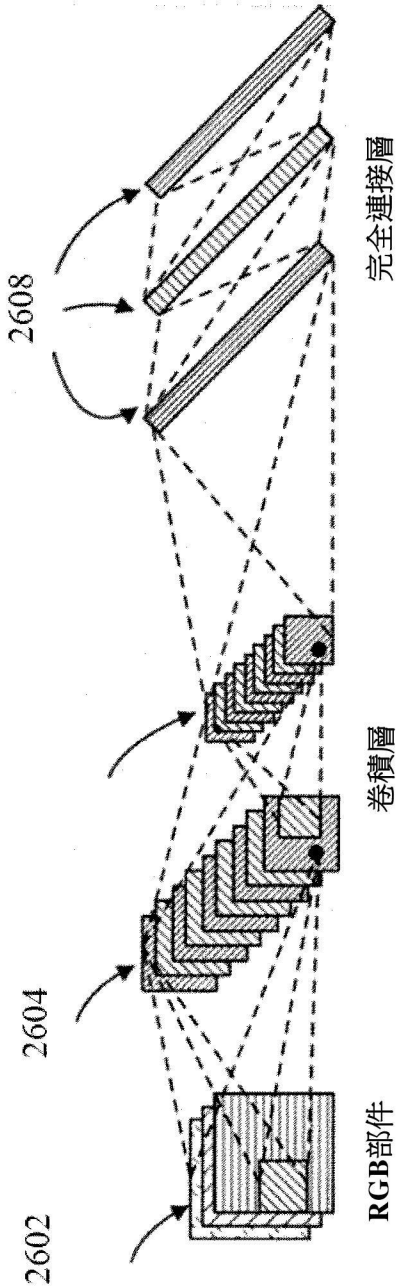
【圖 23】


2400
機器學習應用
2402
機器學習框架
2404
計算框架
2406
GPGPU 驅動器
2408
GPGPU 硬體
2410

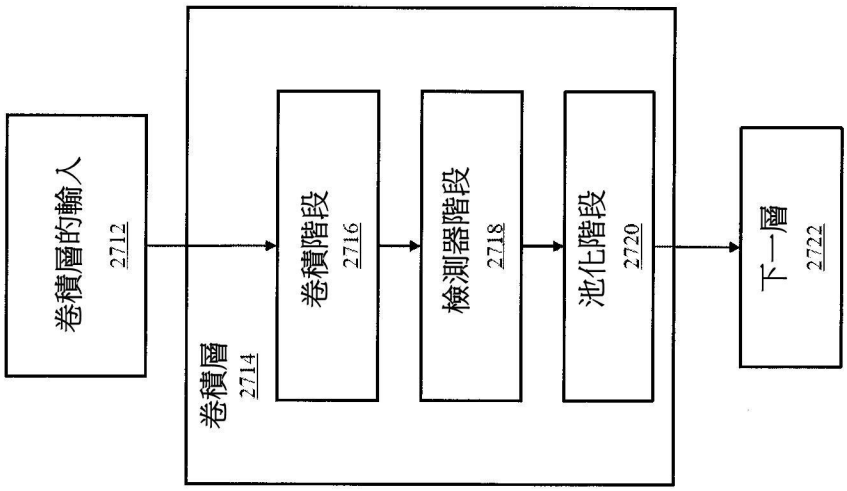
【圖 24】


2516
GPGPU 2506A
GPGPU 2506B
GPGPU 2506C
GPGPU 2506D
主機介開關 2504
處理器 2502

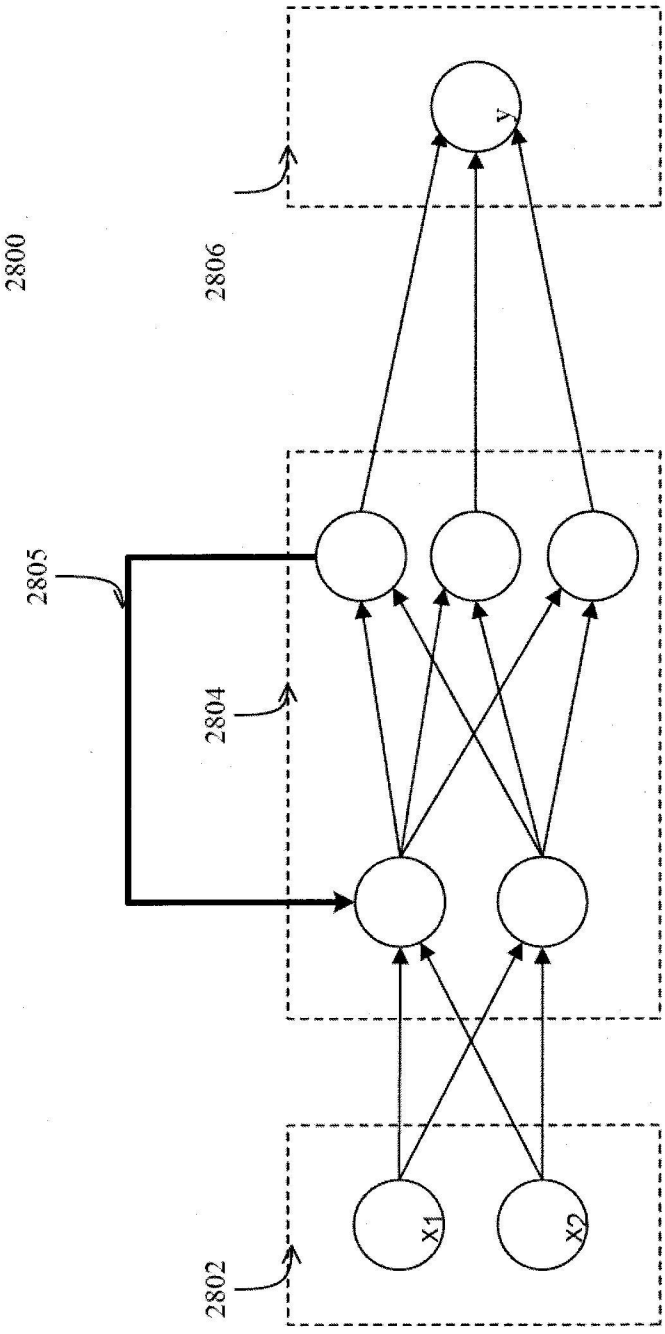
【圖 25】



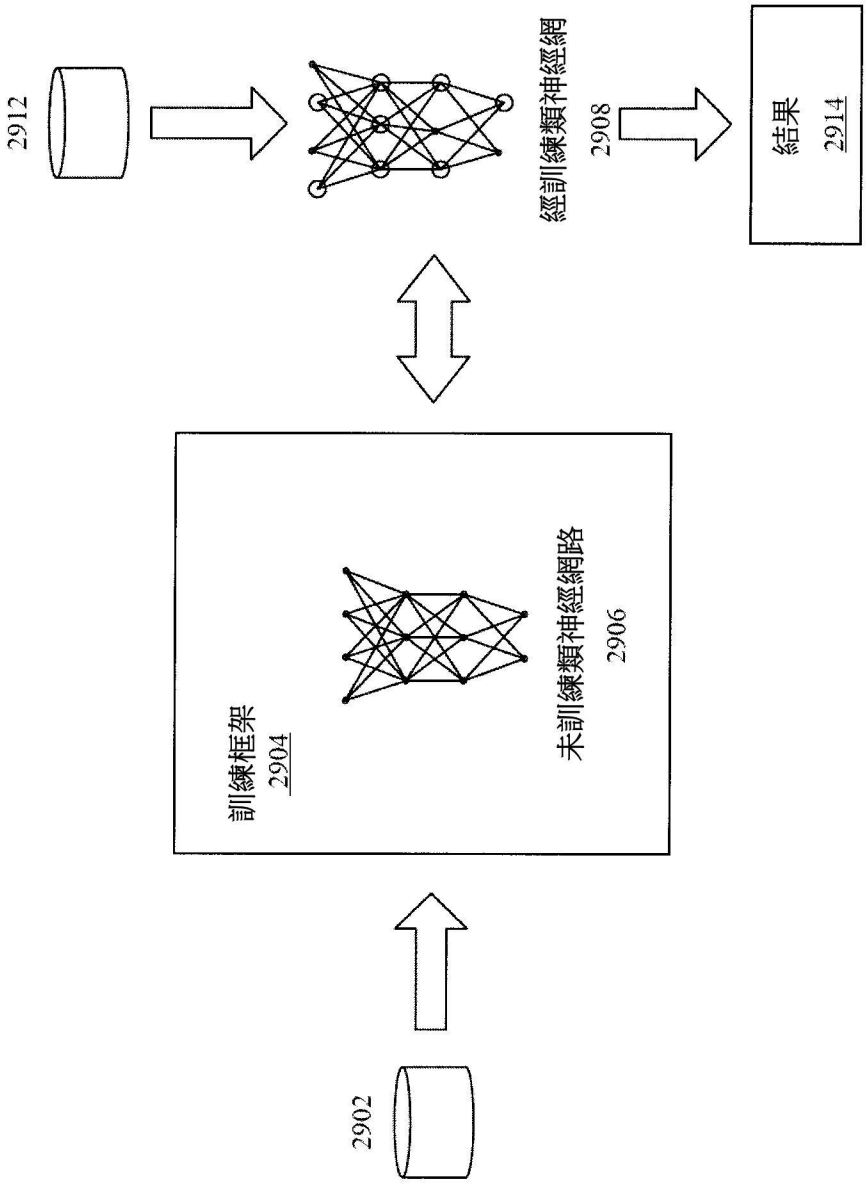
【圖 26】



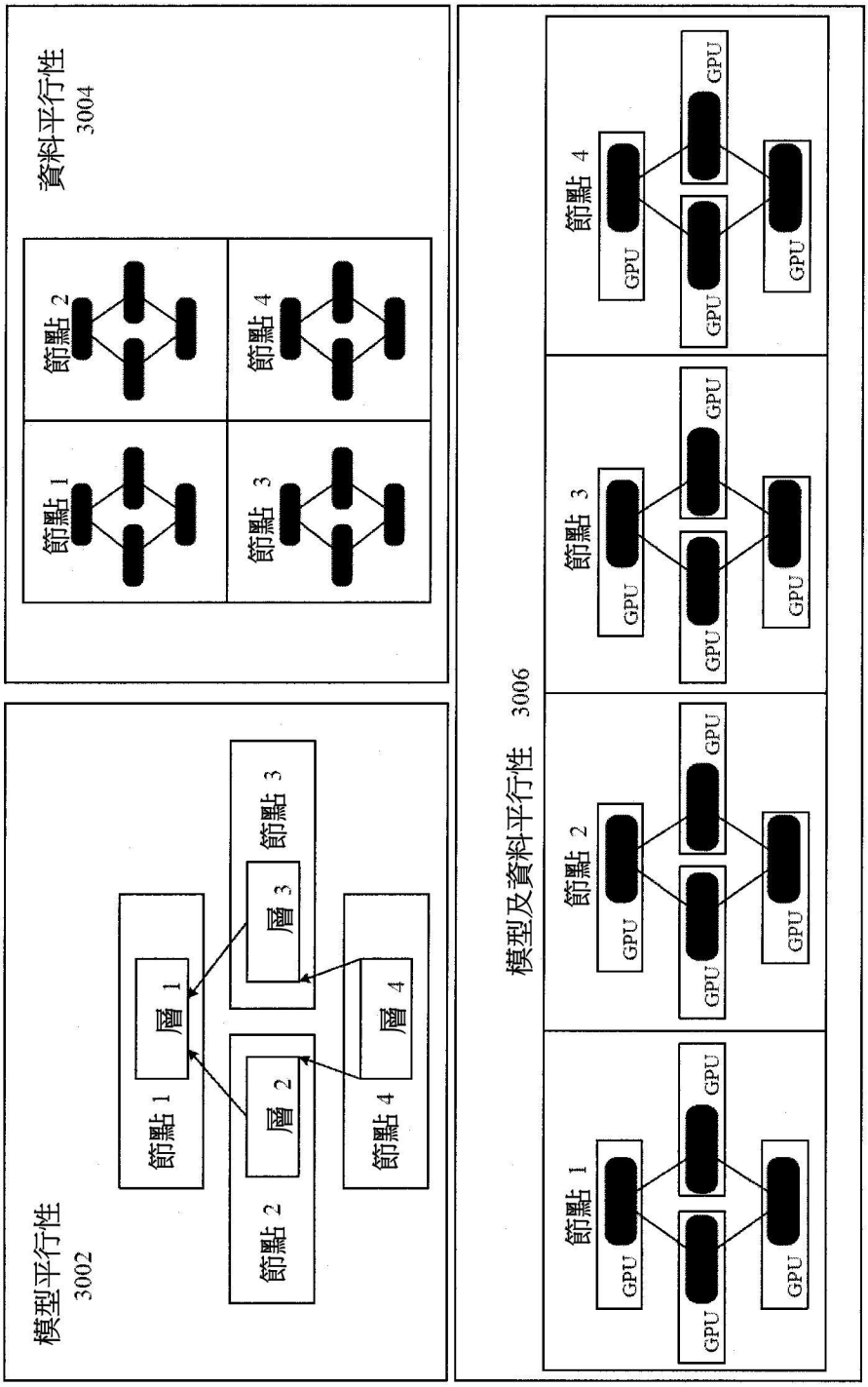
【圖 27】



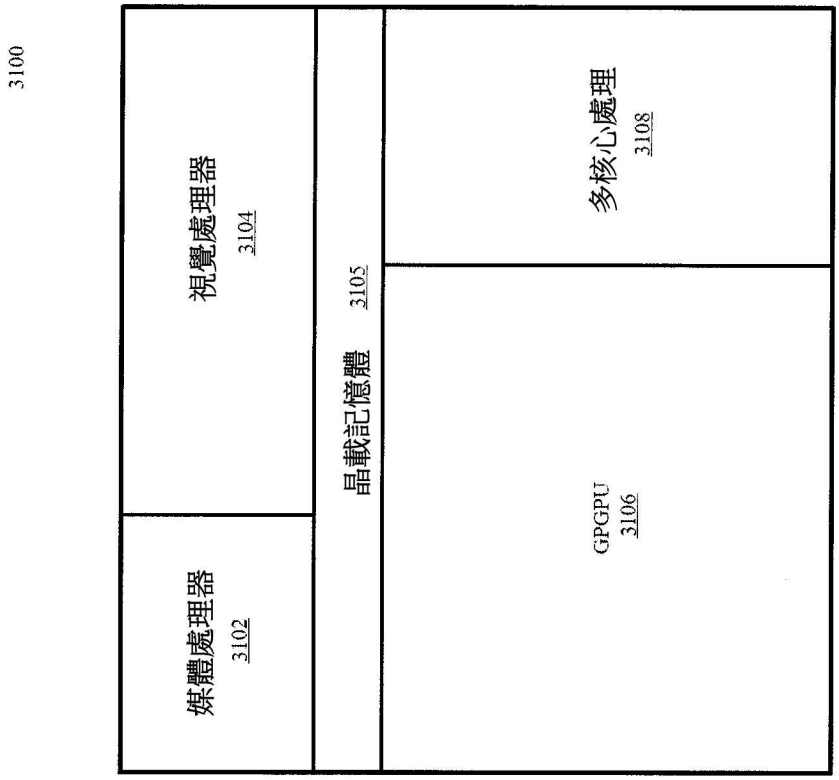
【圖 28】



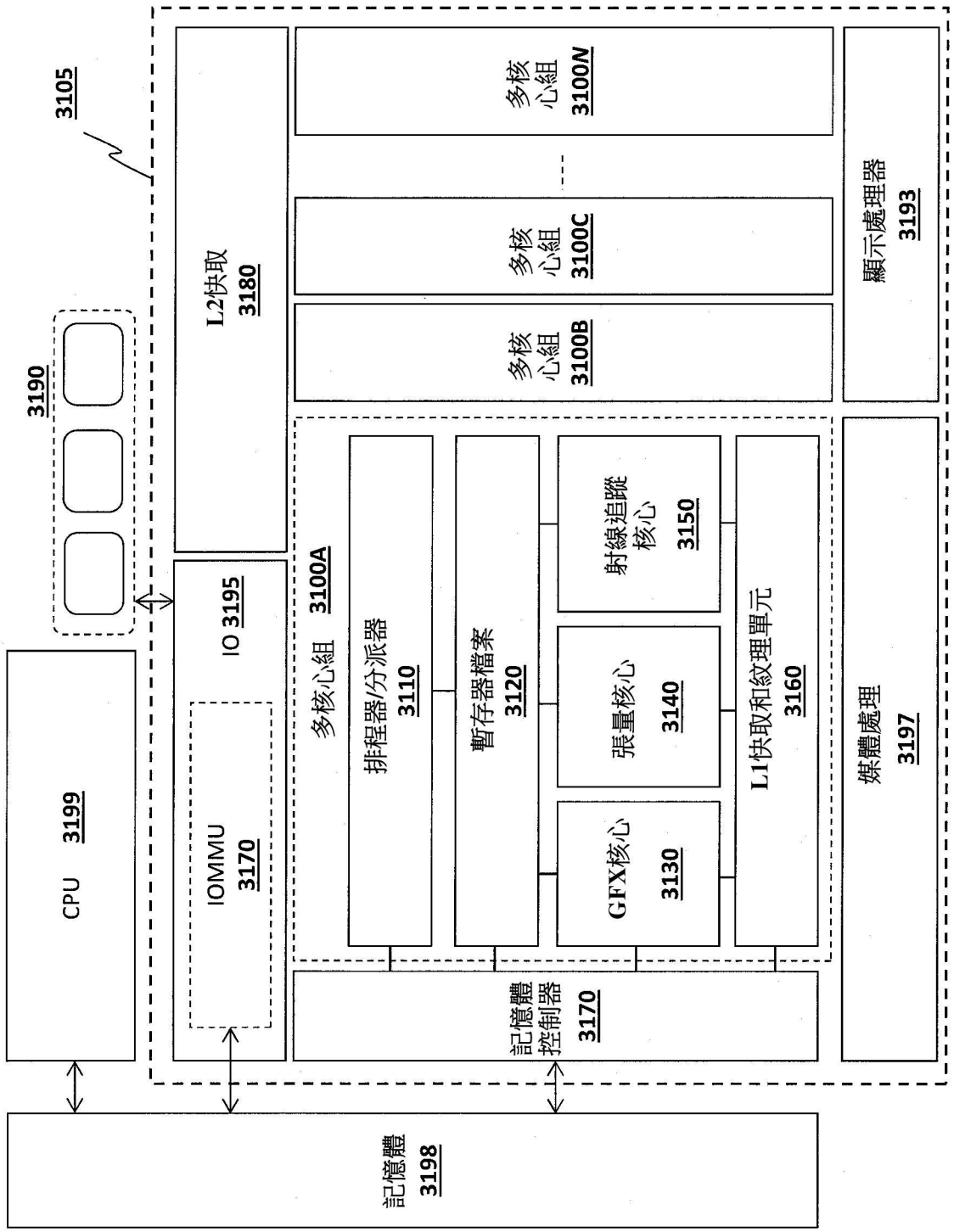
【圖 29】


模型平行性
3002
節點 1
層 1
節點 2
層 2
節點 3
層 3
節點 4
層 4
資料平行性
3004
節點 1
節點 2
節點 3
節點 4
模型及資料平行性 3006
節點 1
節點 2
節點 3
節點 4
GPU

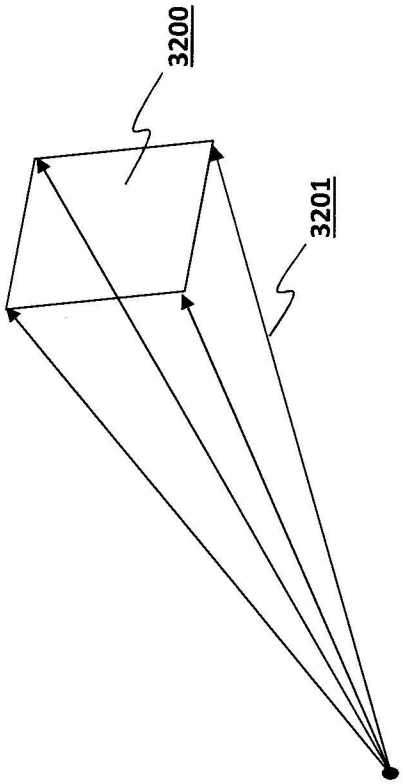
【圖 30A】



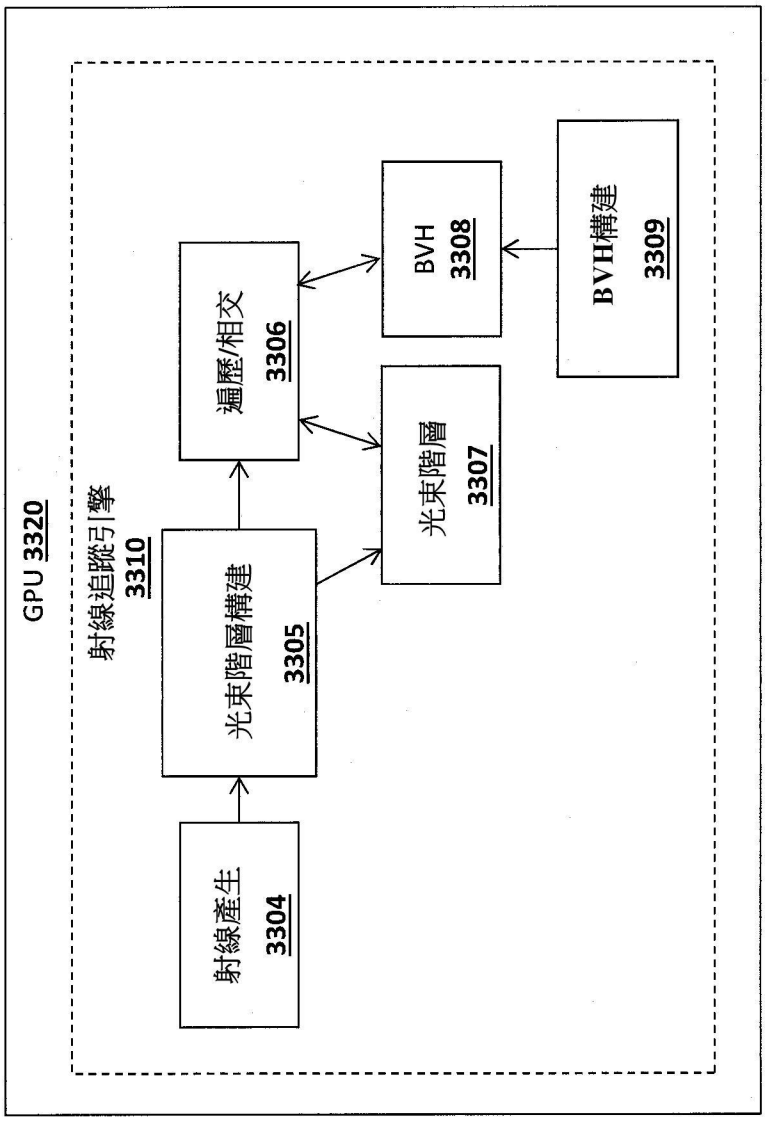
【圖 30B】


3105
3190
CPU 3199
IOMMU 3170
IO 3195
L2快取 3180
多核心組 3100A
排程器/分派器 3110
暫存器檔案 3120
GFX核心 3130
張量核心 3140
射線追蹤核心 3150
L1快取和紋理單元 3160
記憶體控制器 3170
多核心組 3100B
多核心組 3100C
多核心組 3100N
顯示處理器 3193
媒體處理 3197
記憶體 3198

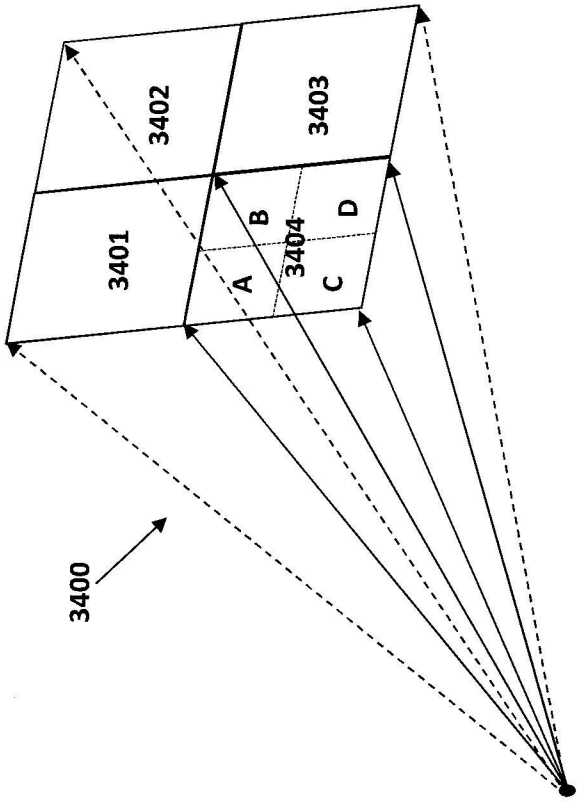
【圖 31】



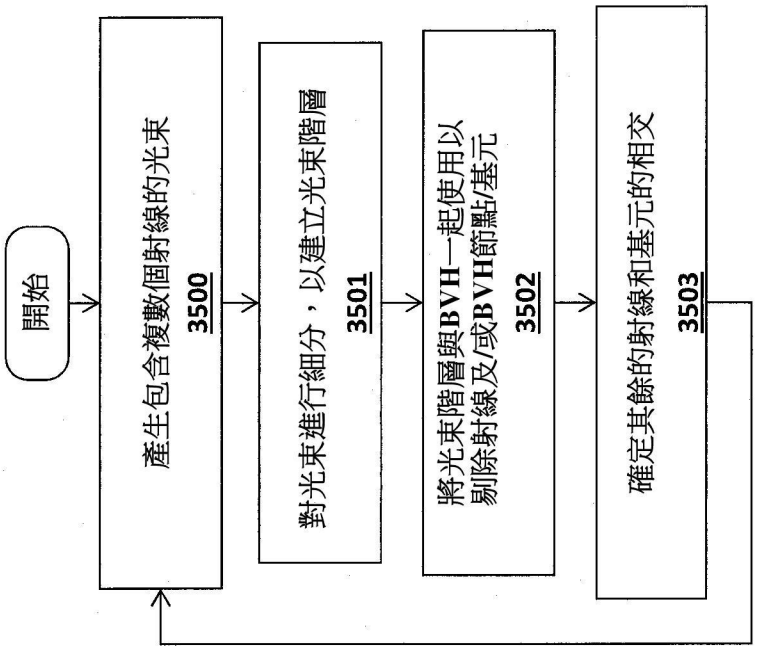
【圖 32】


GPU 3320
射線追蹤引擎 3310
射線產生 3304
光束階層構建 3305
遍歷/相交 3306
光束階層 3307
BVH 3308
BVH構建 3309

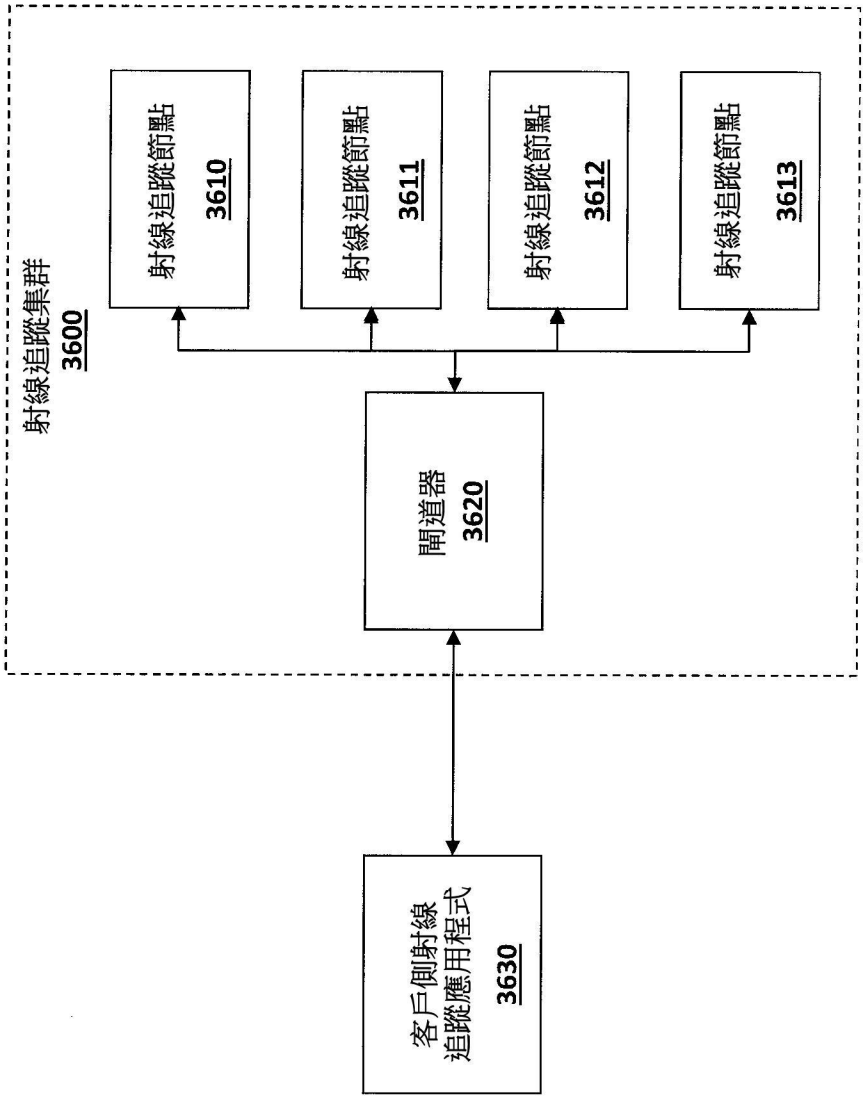
【圖 33】



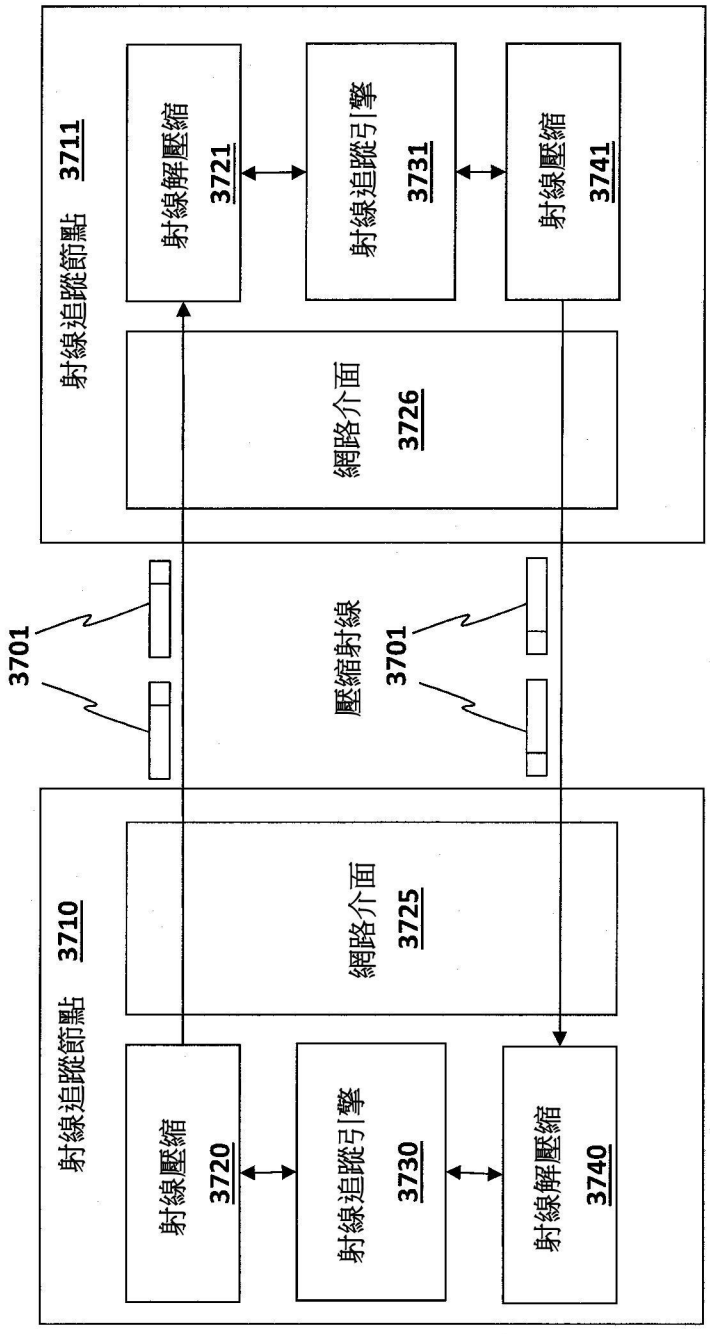
【圖 34】


開始
產生包含複數個射線的光束
3500
對光束進行細分，以建立光束階層
3501
將光束階層與BVH一起使用以
剔除射線及/或BVH節點/基元
3502
確定其餘的射線和基元的相交
3503

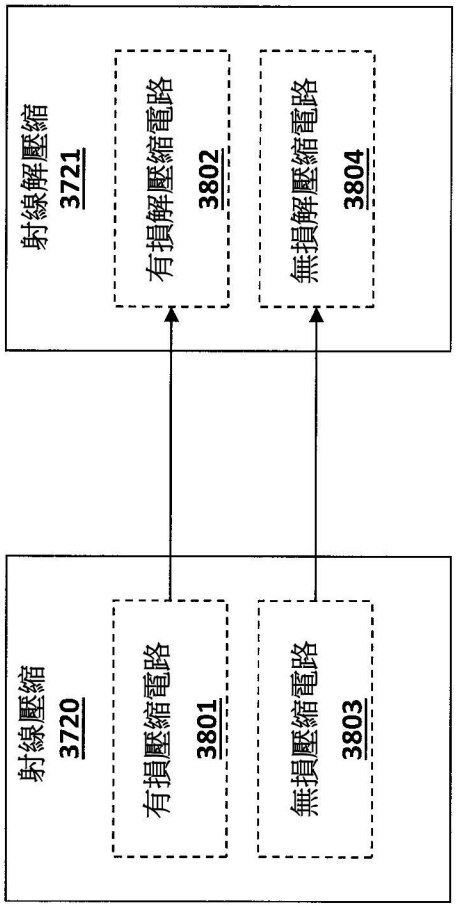
【圖 35】


射線追蹤集群
3600
射線追蹤節點
3610
射線追蹤節點
3611
射線追蹤節點
3612
射線追蹤節點
3613
閘道器
3620
客戶側射線
追蹤應用程式
3630

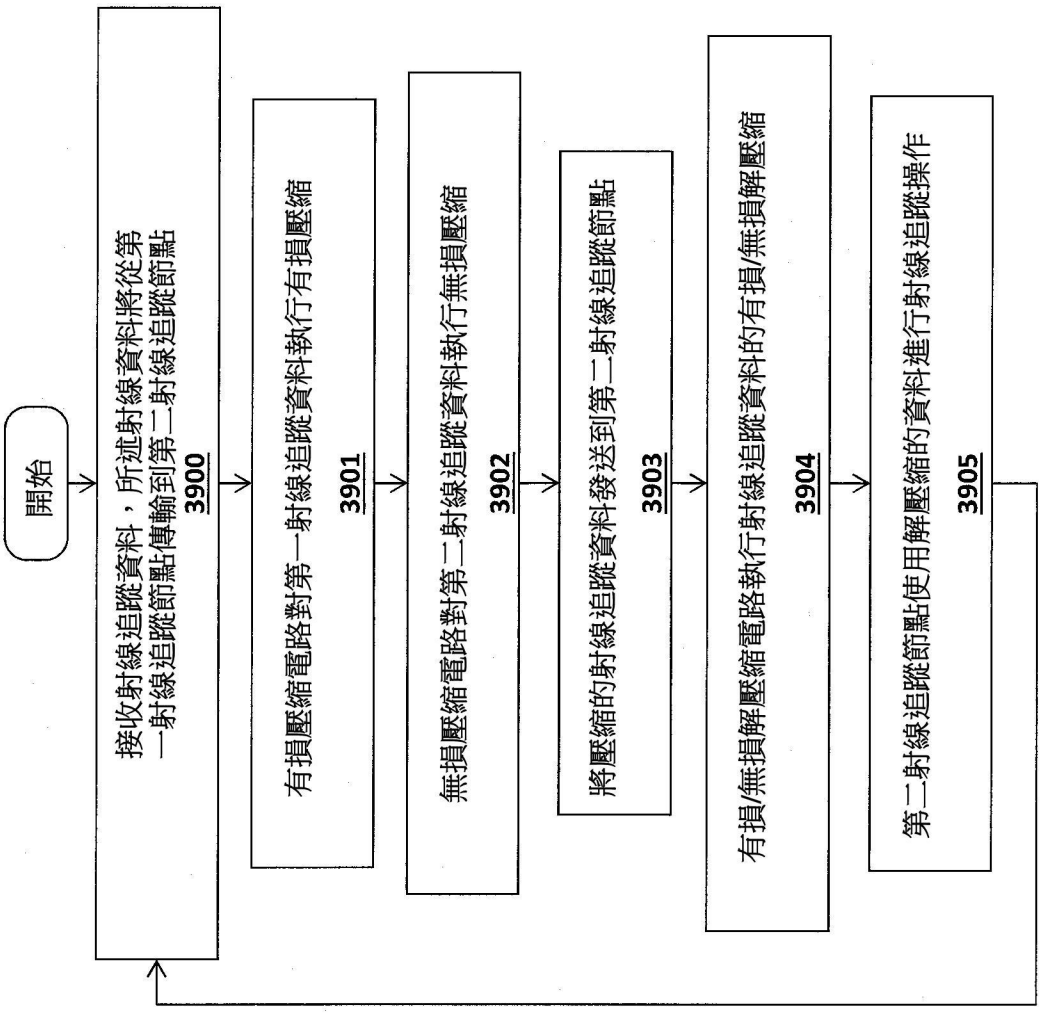
【圖 36】


射線追蹤節點 3711
射線解壓縮 3721
射線追蹤引擎 3731
射線壓縮 3741
網路介面 3726
3701
壓縮射線
3701
射線追蹤節點 3710
網路介面 3725
射線壓縮 3720
射線追蹤引擎 3730
射線解壓縮 3740

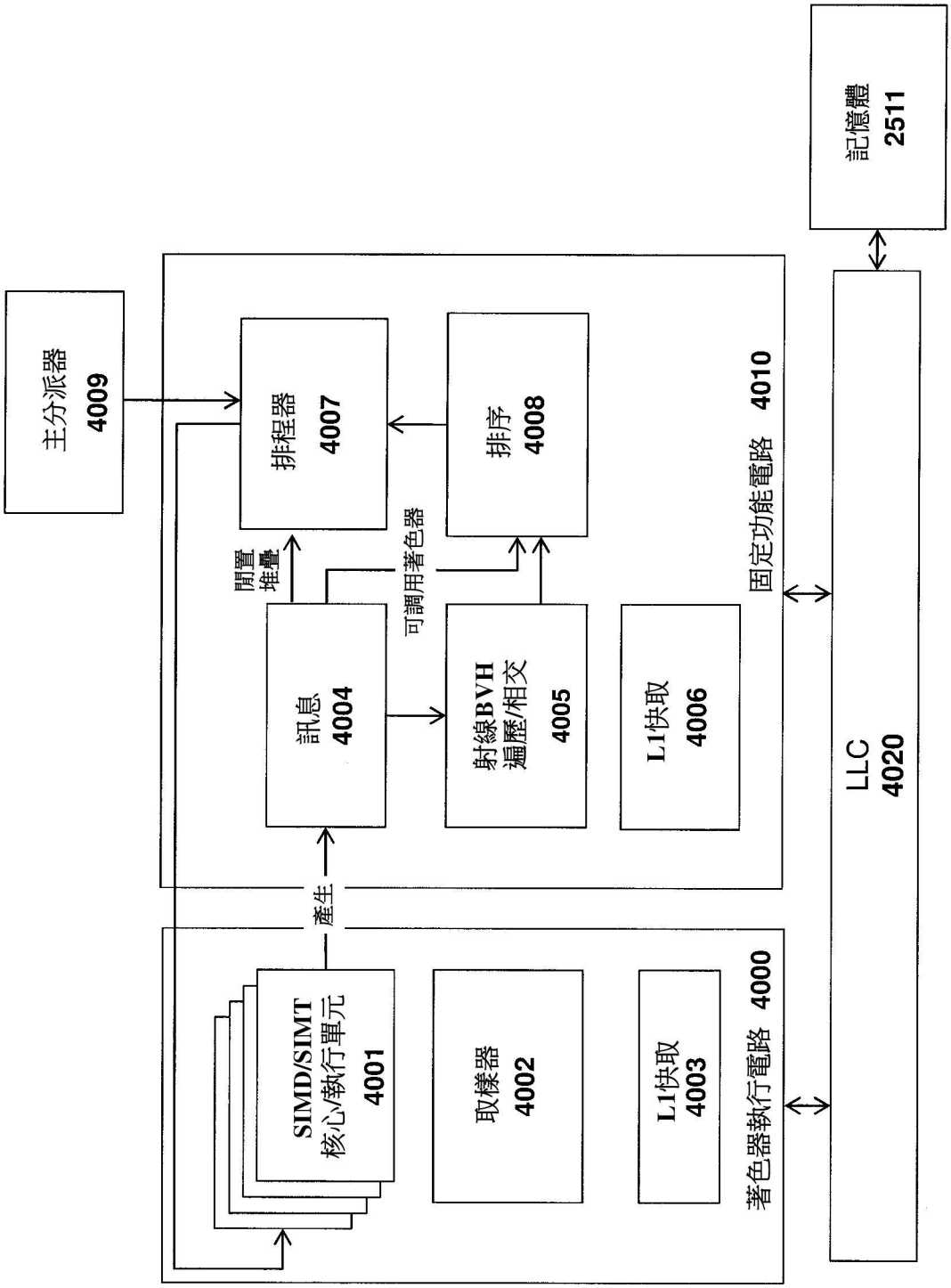
【圖 37】


射線壓縮
3720
有損壓縮電路
3801
無損壓縮電路
3803
射線解壓縮
3721
有損解壓縮電路
3802
無損解壓縮電路
3804

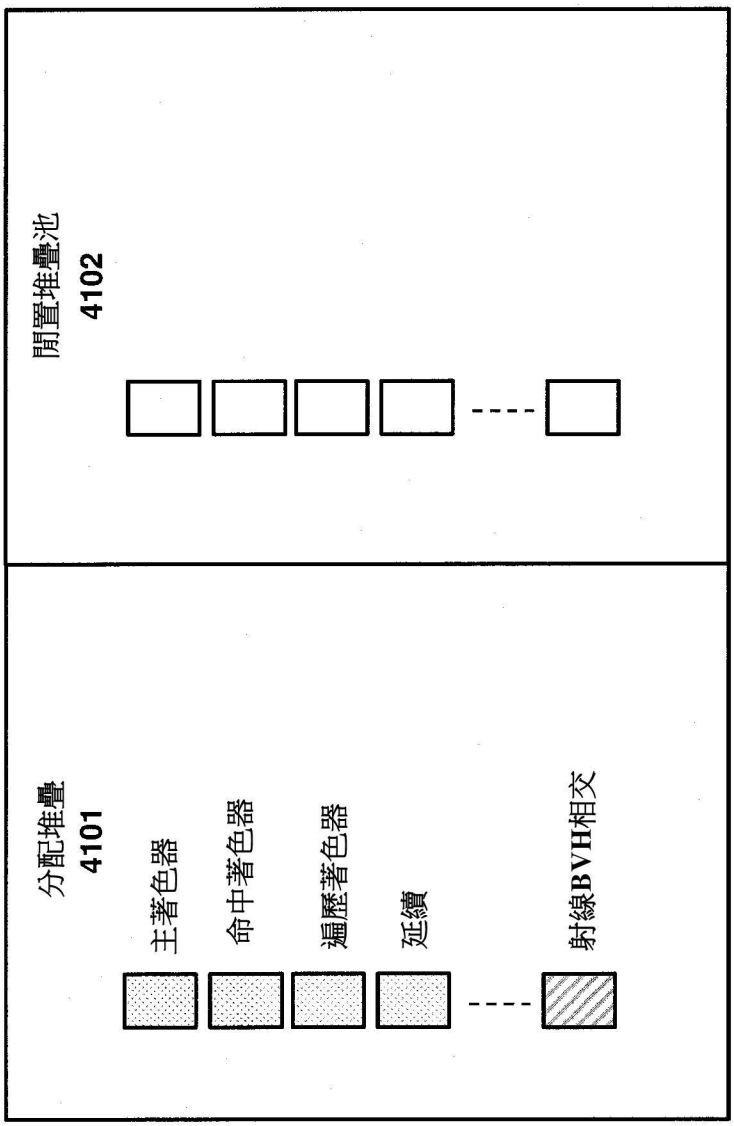
【圖 38】


開始
接收射線追蹤資料，所述射線資料將從第一射線追蹤節點傳輸到第二射線追蹤節點
3900
有損壓縮電路對第一射線追蹤資料執行有損壓縮
3901
無損壓縮電路對第二射線追蹤資料執行無損壓縮
3902
將壓縮的射線追蹤資料發送到第二射線追蹤節點
3903
有損/無損解壓縮電路執行射線追蹤資料的有損/無損解壓縮
3904
第二射線追蹤節點使用解壓縮的資料進行射線追蹤操作
3905

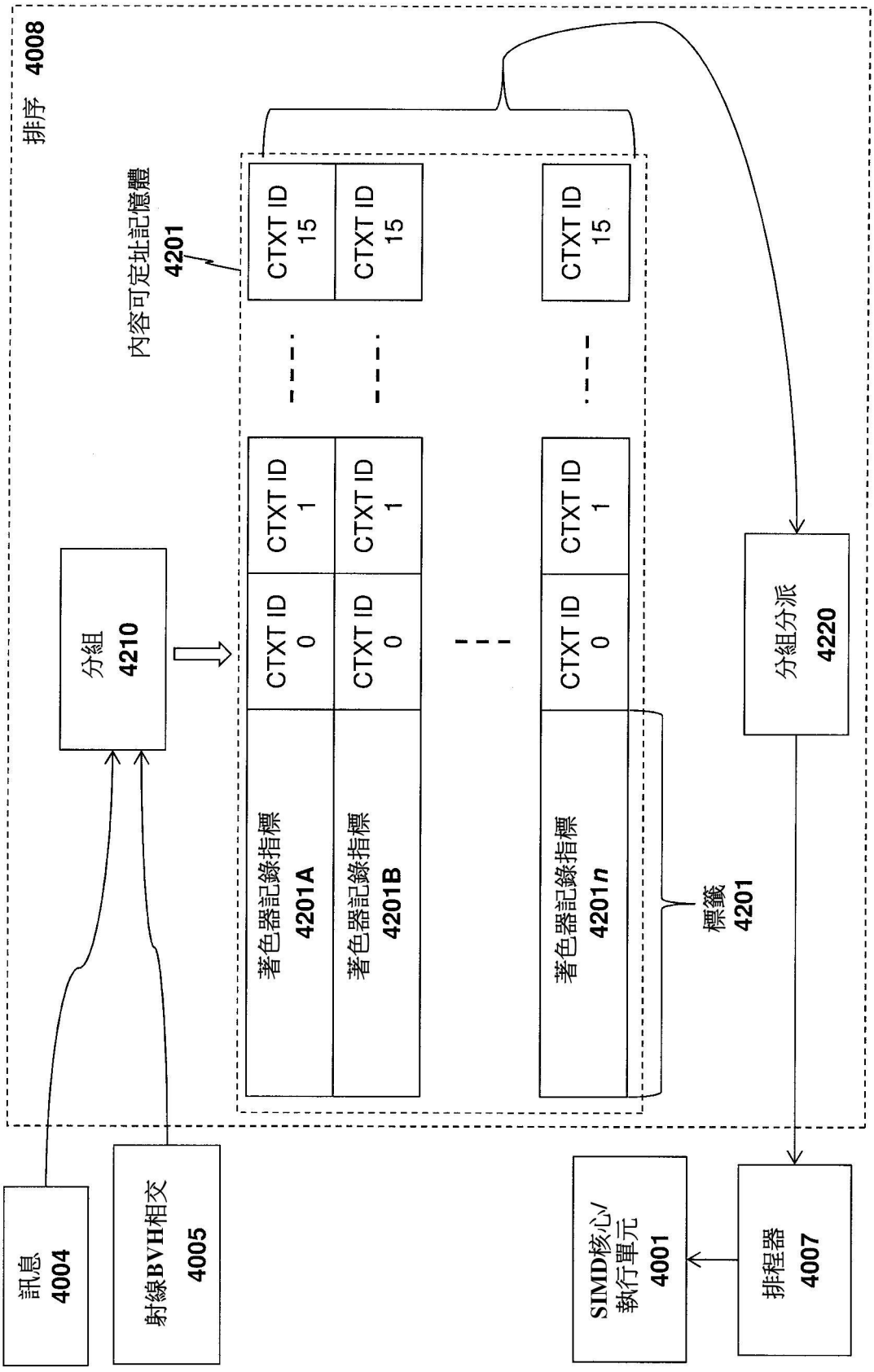
【圖 39】


主分派器
4009
排程器
4007
排序
4008
閉置
推疊
可調用著色器
訊息
4004
射線BVH
遍歷/相交
4005
L1快取
4006
固定功能電路 4010
記憶體
2511
LLC
4020
產生
SIMD/SIMT
核心執行單元
4001
取樣器
4002
L1快取
4003
著色器執行電路 4000

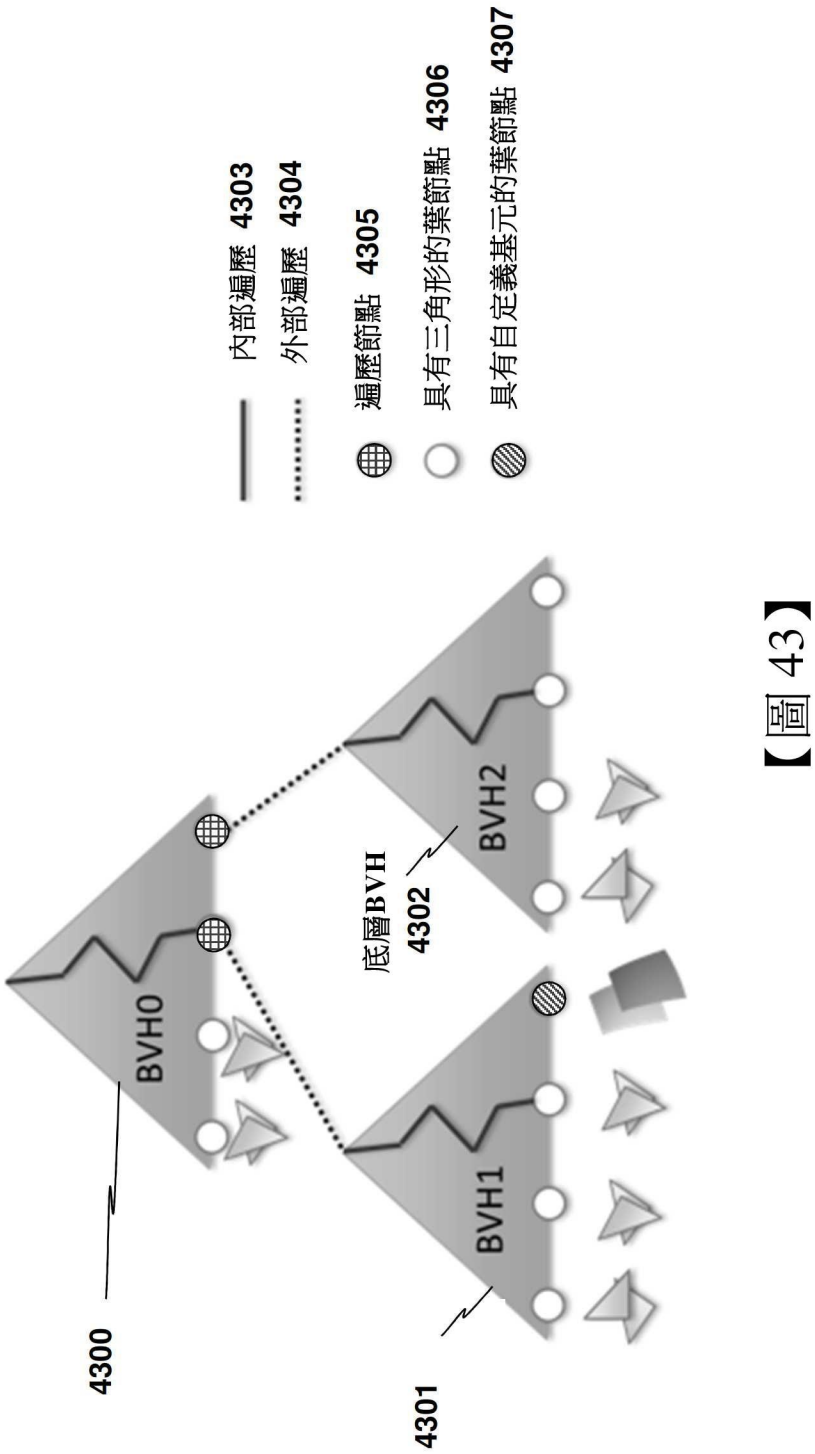
【圖 40】



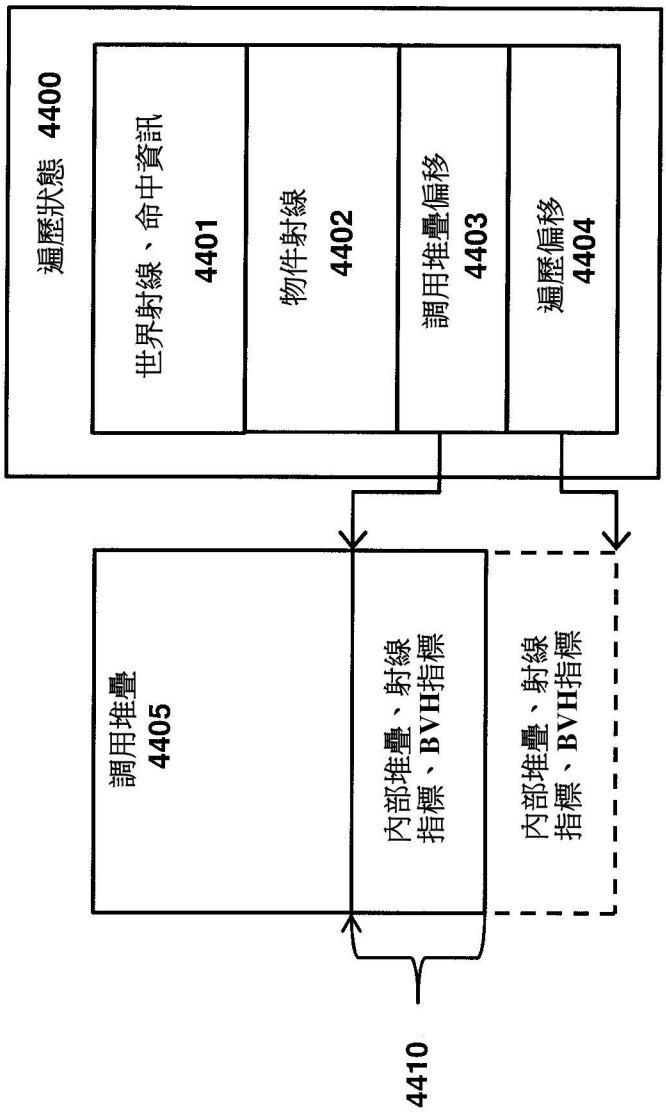
【圖 41】


排序 4008
內容可定址記憶體
4201
分組
4210
著色器記錄指標
4201A
著色器記錄指標
4201B
著色器記錄指標
4201n
CTXT ID
0
CTXT ID
1
CTXT ID
15
標籤
4201
分組分派
4220
訊息
4004
射線BVH相交
4005
SIMD核心/
執行單元
4001
排程器
4007

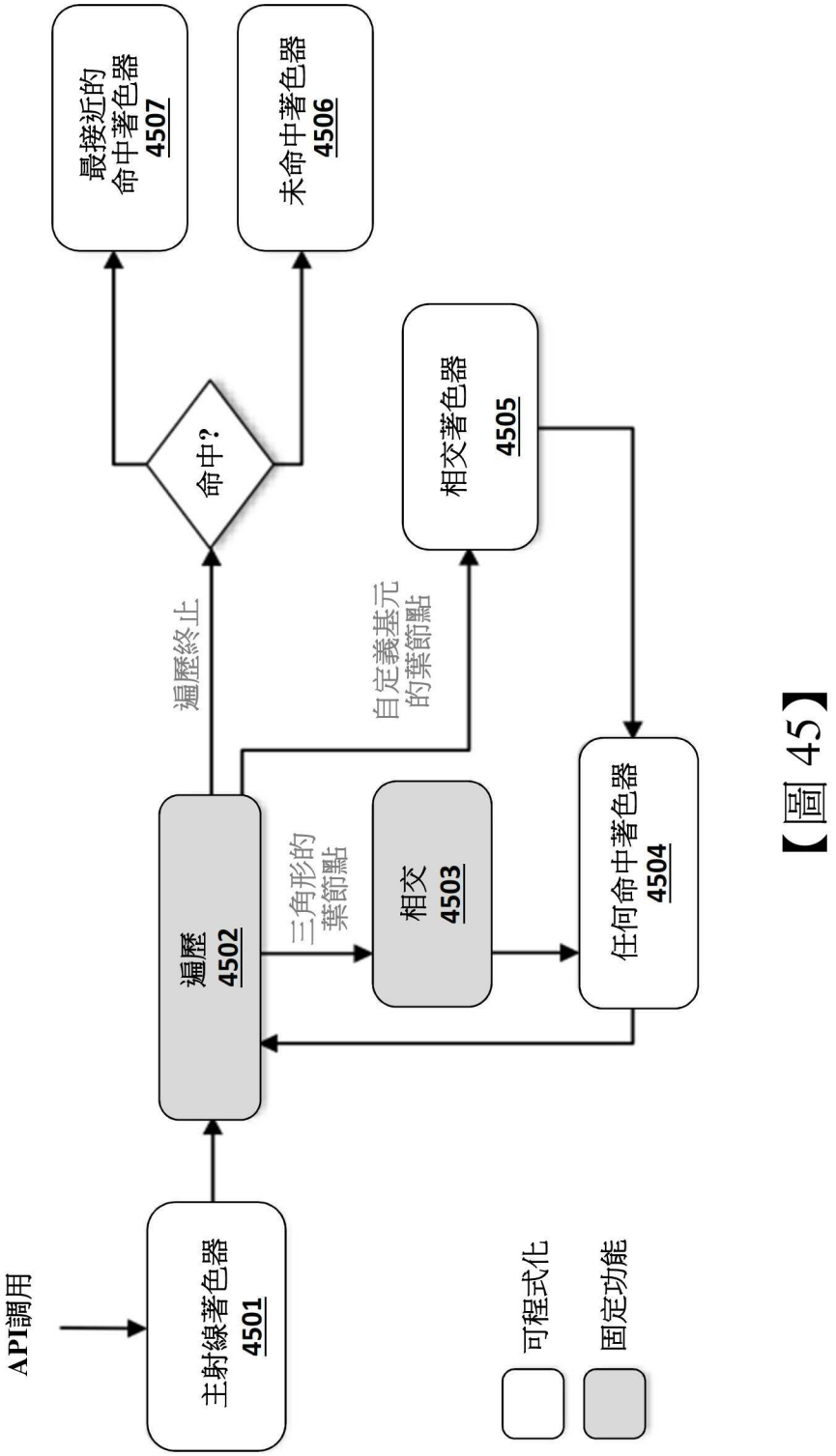
【圖 42】


內部遍歷 4303
外部遍歷 4304
遍歷節點 4305
具有三角形的葉節點 4306
具有自定義基元的葉節點 4307
4300
BVH0
底層BVH
4302
BVH2
BVH1
4301

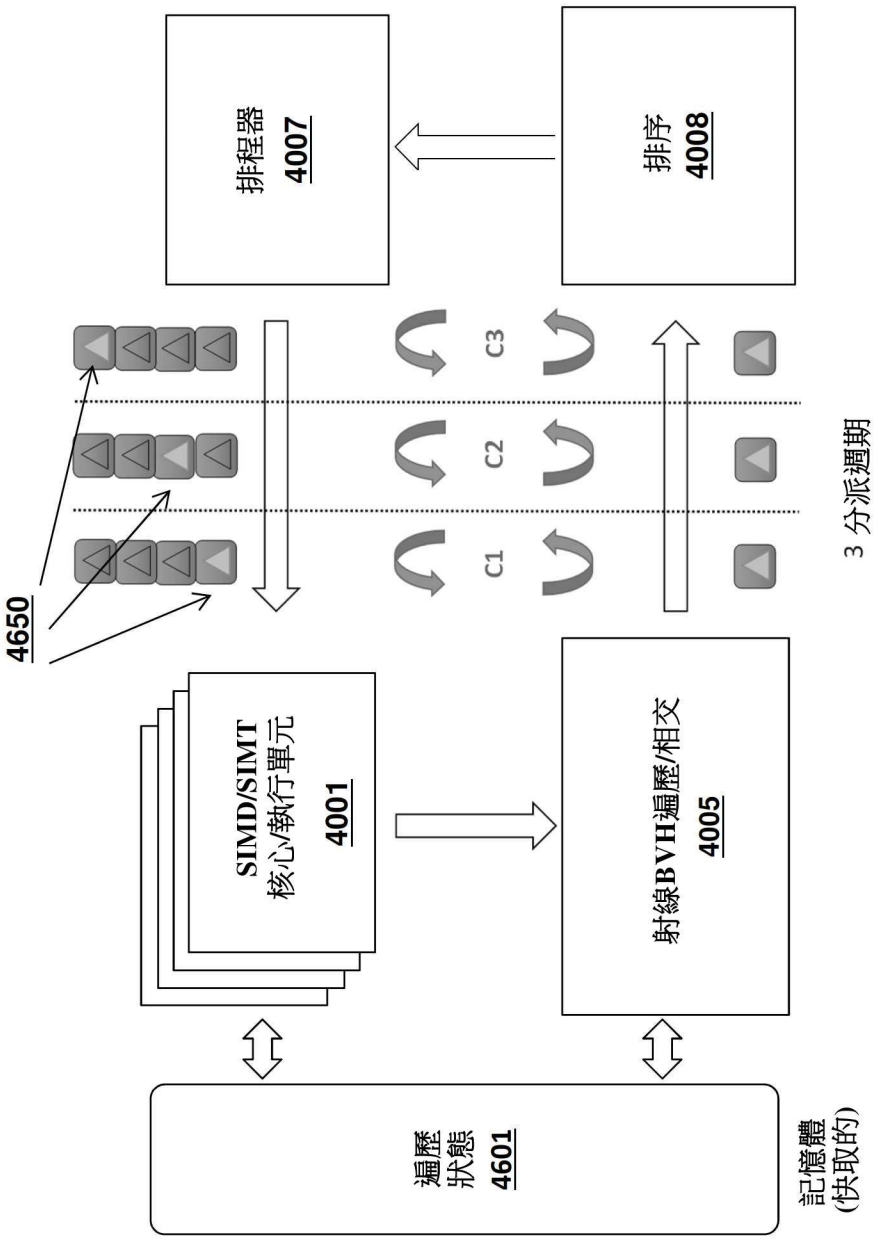
【圖 43】


遍歷狀態 4400
世界射線、命中資訊 4401
物件射線 4402
調用堆疊偏移 4403
遍歷偏移 4404
調用堆疊 4405
內部堆疊、射線指標、BVH指標
內部堆疊、射線指標、BVH指標
4410

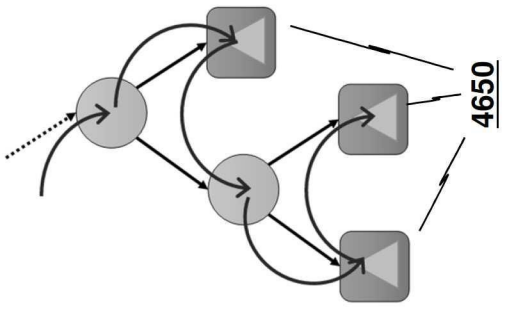
【圖 44】


API調用
主射線著色器
4501
遍歷
4502
三角形的
葉節點
相交
4503
任何命中著色器
4504
遍歷終止
命中?
最接近的
命中著色器
4507
未命中著色器
4506
自定義基元
的葉節點
相交著色器
4505
可程式化
固定功能

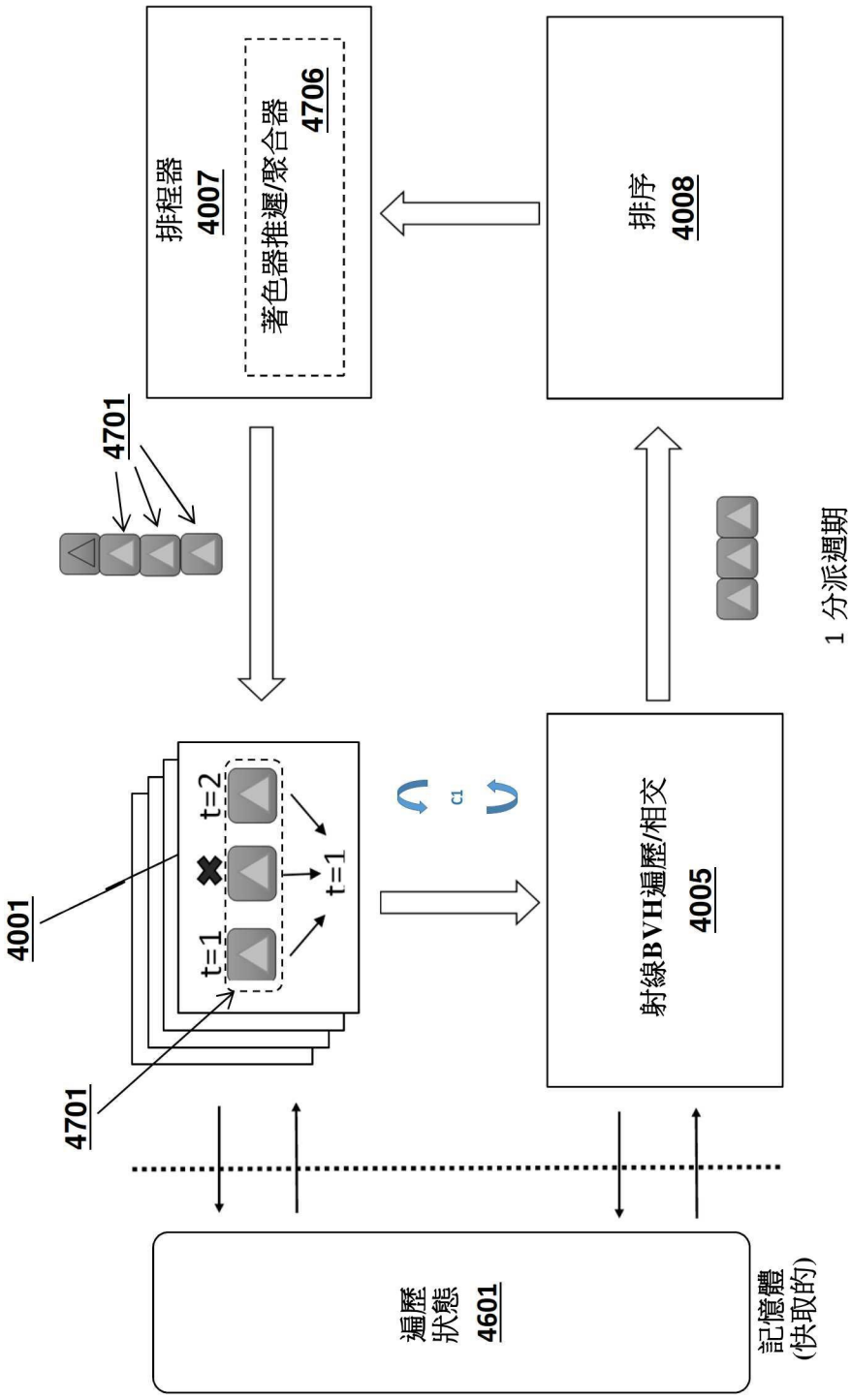
【圖 45】


排程器
4007
排序
4008
4650
C1
C2
C3
3 分派週期
SIMD/SIMT
核心/執行單元
4001
射線BVH遍歷/相交
4005
遍歷
狀態
4601
記憶體
(快取的)

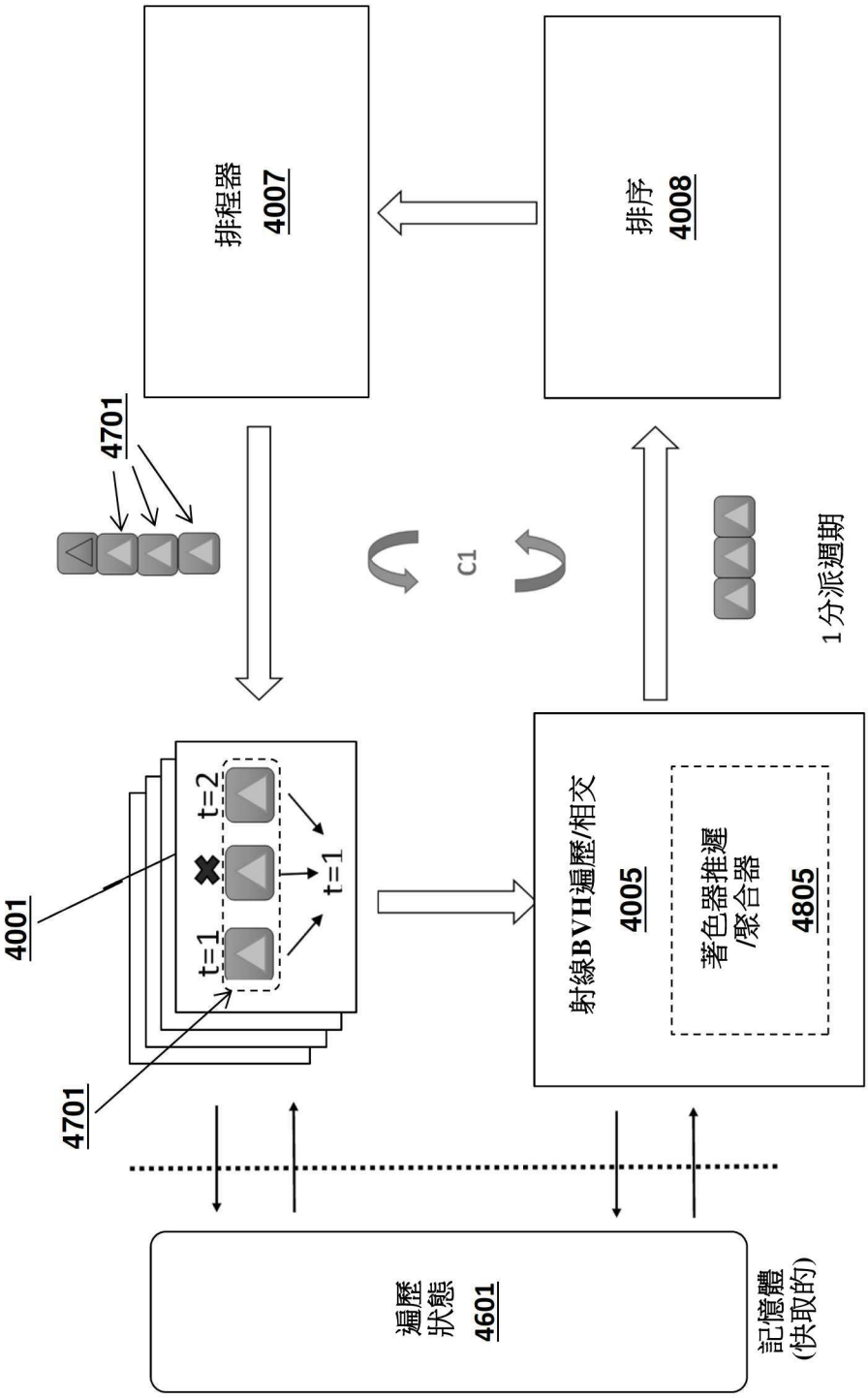
【圖 46B】


4650

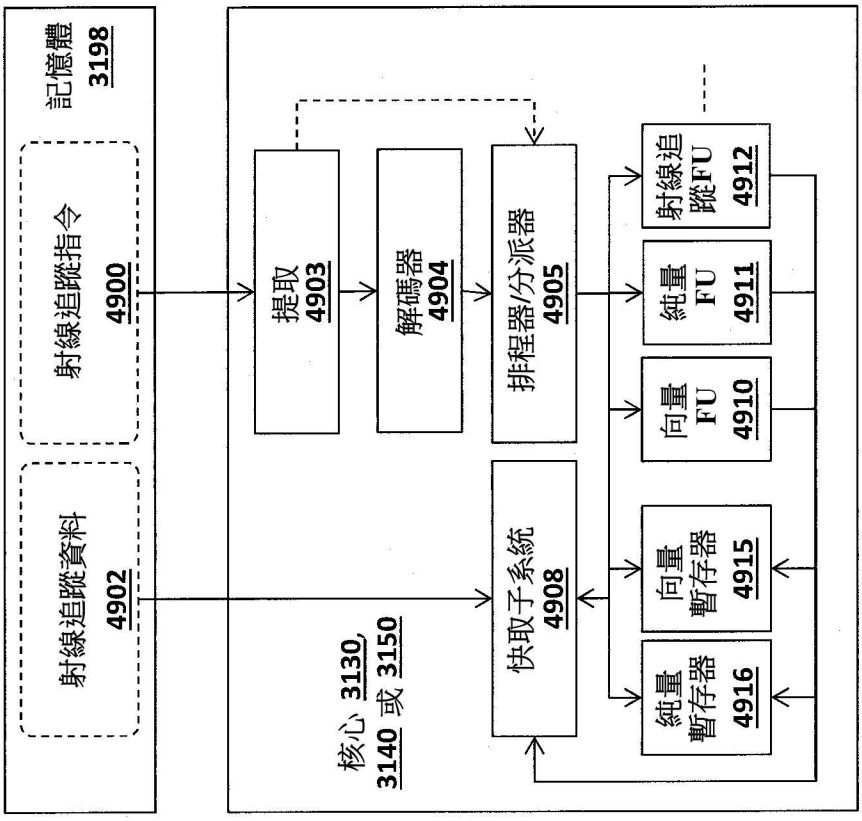
【圖 46A】


排程器
4007
著色器推遲/聚合器
4706
排序
4008
4701
1 分派週期
4001
t=1
t=2
t=1
α
射線BVH遍歷/相交
4005
4701
遍歷
狀態
4601
記憶體
(快取的)

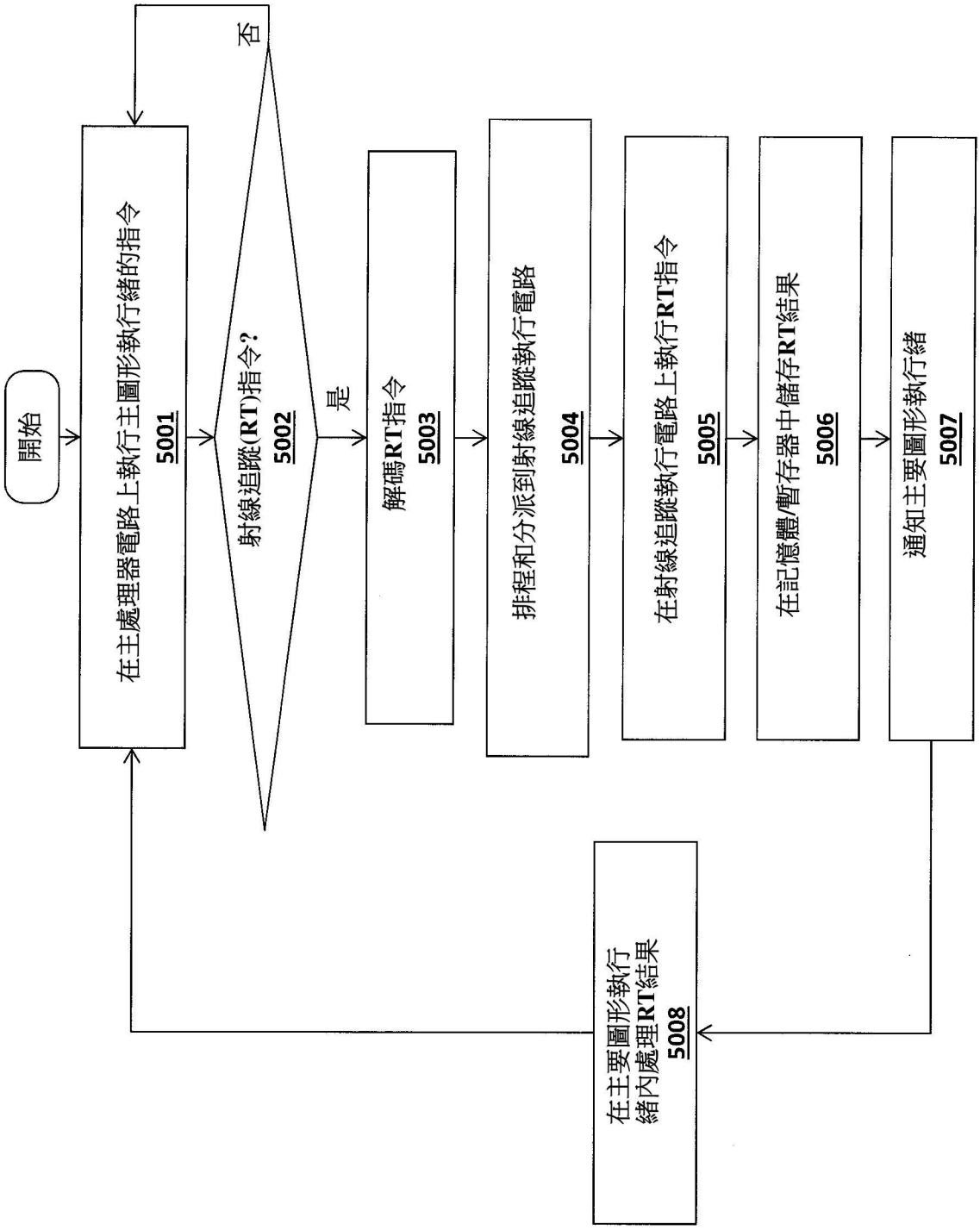
【圖 47】


排程器
4007
排序
4008
4701
C1
1分派週期
4001
t=1
t=2
t=1
射線BVH遍歷/相交
4005
著色器推遲
/聚合器
4805
4701
遍歷
狀態
4601
記憶體
(快取的)

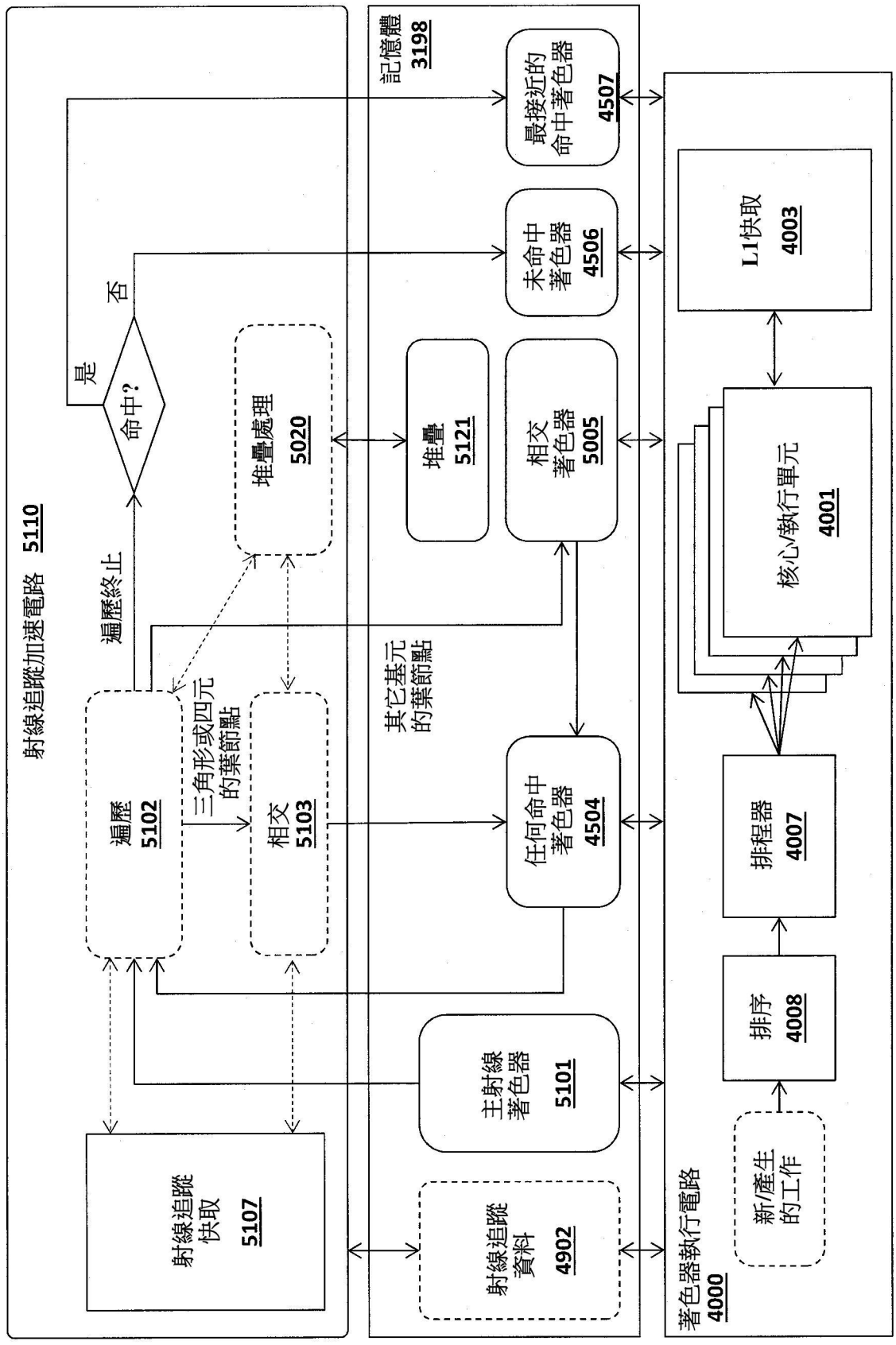
【圖 48】


記憶體 3198
射線追蹤指令 4900
射線追蹤資料 4902
核心 3130, 3140 或 3150
提取 4903
解碼器 4904
排程器/分派器 4905
快取子系統 4908
向量 FU 4910
純量 FU 4911
射線追蹤FU 4912
向量暫存器 4915
純量暫存器 4916

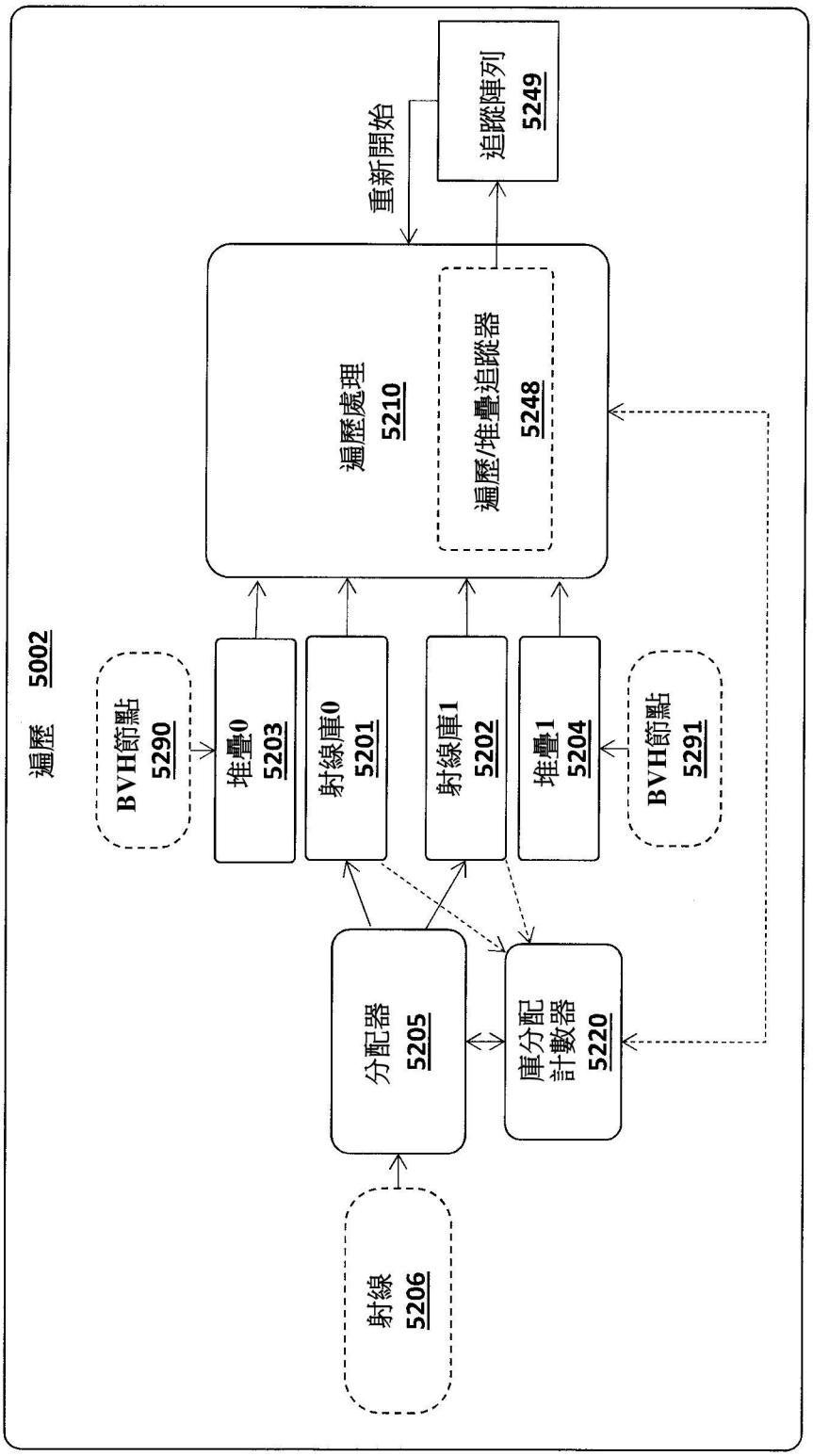
【圖 49】


開始
在主處理器電路上執行主圖形執行緒的指令
5001
射線追蹤(RT)指令?
5002
否
是
解碼RT指令
5003
排程和分派到射線追蹤執行電路
5004
在射線追蹤執行電路上執行RT指令
5005
在記憶體/暫存器中儲存RT結果
5006
通知主要圖形執行緒
5007
在主要圖形執行緒內處理RT結果
5008

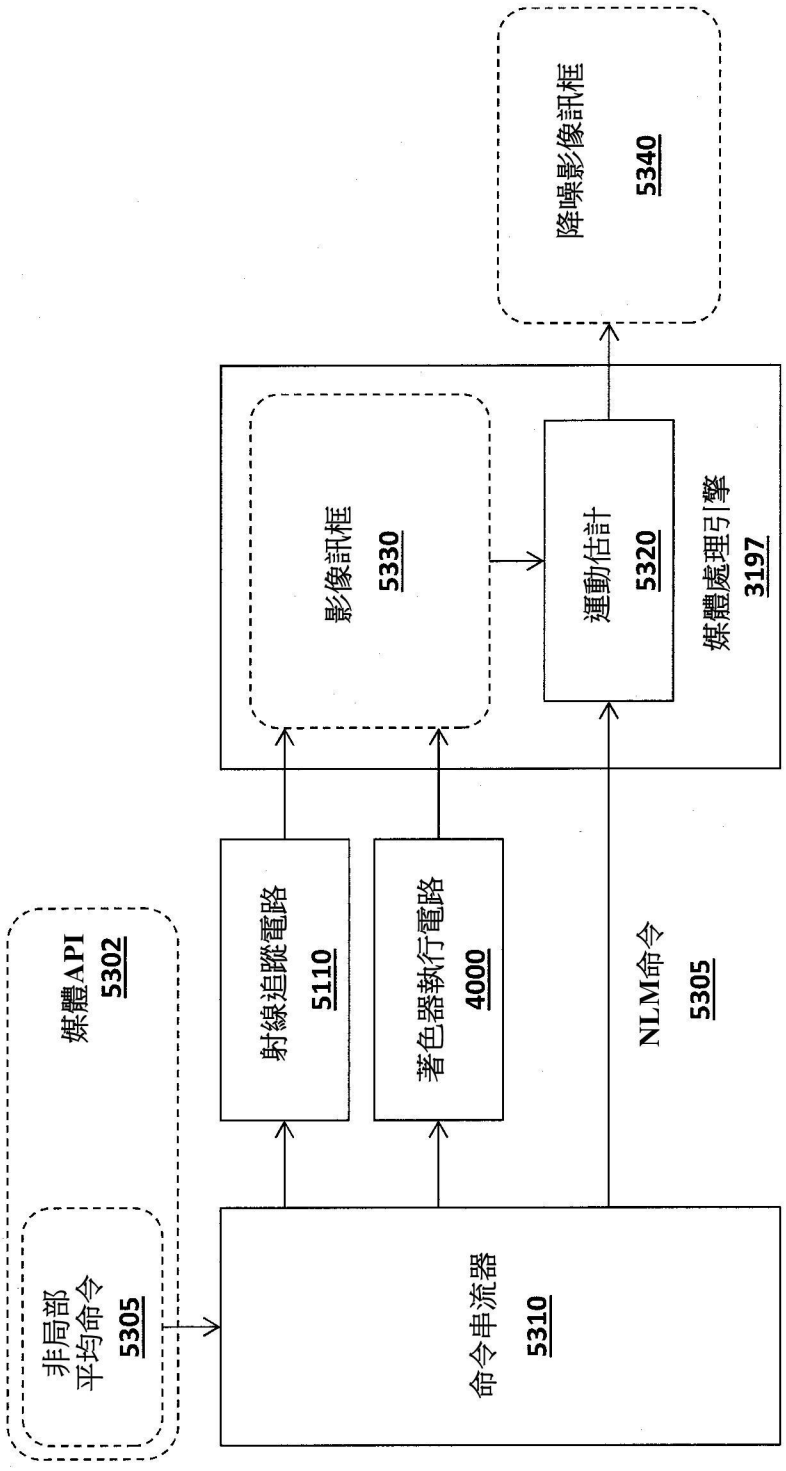
【圖 50】


射線追蹤加速電路 5110
遍歷 5102
三角形或四元的葉節點
相交 5103
遍歷終止
是
命中?
否
堆疊處理 5020
射線追蹤快取 5107
記憶體 3198
堆疊 5121
相交著色器 5005
最接近的命中著色器 4507
未命中著色器 4506
其它基元的葉節點
任何命中著色器 4504
主射線著色器 5101
射線追蹤資料 4902
著色器執行電路 4000
新/產生的工作
排序 4008
排程器 4007
核心/執行單元 4001
L1快取 4003

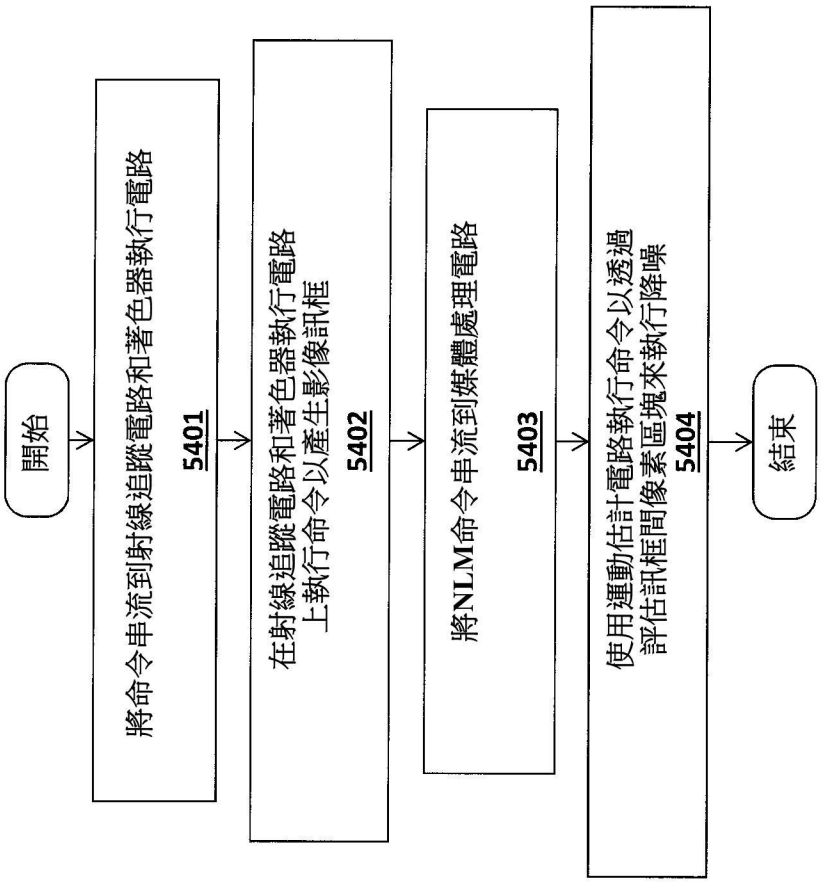
【圖 51】


遍歷 5002
BVH節點 5290
堆疊0 5203
射線庫0 5201
射線庫1 5202
堆疊1 5204
BVH節點 5291
遍歷處理 5210
遍歷/堆疊追蹤器 5248
追蹤陣列 5249
重新開始
分配器 5205
庫分配計數器 5220
射線 5206

【圖 52】


非局部
平均命令
5305
媒體API
5302
命令串流器
5310
射線追蹤電路
5110
著色器執行電路
4000
NLM命令
5305
媒體處理引擎
3197
影像訊框
5330
運動估計
5320
降噪影像訊框
5340

【圖 53】


開始
將命令串流到射線追蹤電路和著色器執行電路
5401
在射線追蹤電路和著色器執行電路
上執行命令以產生影像訊框
5402
將NLM命令串流到媒體處理電路
5403
使用運動估計電路執行命令以透過
評估訊框間像素區塊來執行降噪
5404
結束

【圖 54】