



(19)中華民國智慧財產局

(12)發明說明書公開本

(11)公開編號：TW 202504312 A

(43)公開日：中華民國 114 (2025) 年 01 月 16 日

(21)申請案號：113134787

(22)申請日：中華民國 109 (2020) 年 03 月 03 日

(51)Int. Cl.：

*H04N19/176 (2014.01)**H04N19/186 (2014.01)**H04N9/68 (2023.01)*

(30)優先權：2019/03/04 美國

62/813,728

2019/03/12 美國

62/817,546

2020/02/28 美國

16/805,611

(71)申請人：香港商阿里巴巴集團服務有限公司 (香港地區) ALIBABA GROUP SERVICES LIMITED (HK)

香港

(72)發明人：葉琰 (CN)；陳杰 (CN)；廖如伶 (CN)

(74)代理人：林志剛

申請實體審查：有 申請專利範圍項數：10 項 圖式數：16 共 94 頁

(54)名稱

用於處理視訊內容的方法及系統

(57)摘要

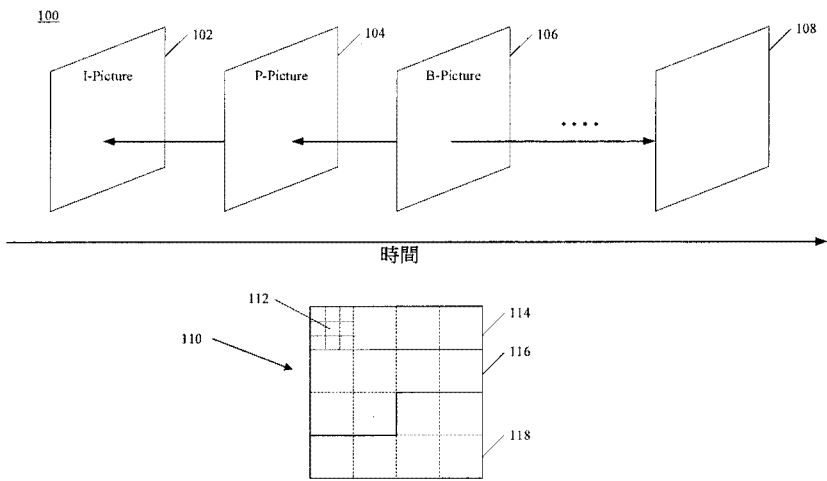
本揭露之實施例提供用於處理視訊內容之系統及方法。該方法包括：接收與畫面相關聯之色度區塊與亮度區塊；判定與該亮度區塊相關聯之亮度縮放資訊；基於該亮度縮放資訊判定色度縮放因素；以及使用該色度縮放因素處理該色度區塊。

Embodiments of the present disclosure provides systems and methods for processing video content. The methods include: receiving a chroma block and a luma block associated with a picture; determining luma scaling information associated with the luma block; determining a chroma scaling factor based on the luma scaling information; and processing the chroma block using the chroma scaling factor.

指定代表圖：

符號簡單說明：

- 100:視訊序列
- 102:畫面
- 104:畫面
- 106:畫面
- 108:畫面
- 110:結構
- 112:基本處理單元
- 114:區
- 116:區
- 118:區



【圖 1】

【發明摘要】

【中文發明名稱】

用於處理視訊內容的方法及系統

【英文發明名稱】

METHOD AND SYSTEM FOR PROCESSING VIDEO CONTENT

【中文】

本揭露之實施例提供用於處理視訊內容之系統及方法。該方法包括：接收與畫面相關聯之色度區塊與亮度區塊；判定與該亮度區塊相關聯之亮度縮放資訊；基於該亮度縮放資訊判定色度縮放因素；以及使用該色度縮放因素處理該色度區塊。

【英文】

Embodiments of the present disclosure provides systems and methods for processing video content. The methods include: receiving a chrome block and a luma block associated with a picture; determining luma scaling information associated with the luma block; determining a chroma scaling factor based on the luma scaling information; and processing the chroma block using the chroma scaling factor.

【指定代表圖】圖 1

【代表圖之符號簡單說明】

100:視訊序列

102:畫面

104:畫面

106:畫面

108:畫面

110:結構

112:基本處理單元

114:區

116:區

118:區

【特徵化學式】無

【發明說明書】

【中文發明名稱】

用於處理視訊內容的方法及系統

【英文發明名稱】

METHOD AND SYSTEM FOR PROCESSING VIDEO CONTENT

【技術領域】

【0001】本揭露一般相關於視訊處理，且更明確相關於用於執行具色度縮放的迴路內亮度映射之方法與系統。

相關申請案之交叉引用

【0002】本揭露主張2019年3月4日提交的美國臨時申請案No. 62/813,728和於2019年3月12日提交的美國臨時申請案No. 62/817,546之優先權權益，上述兩案均透過對其之引用而全部併入本文中。

【先前技術】

【0003】通常將視訊編碼系統用於壓縮數位視訊訊號，舉例而言用以降低消耗之儲存空間或用以降低與此等訊號相關聯之傳輸頻寬消耗。隨著高解析(HD)視訊(例如具有1920×1080像素之解析度之視訊)在各種視訊壓縮之應用(諸如線上視訊串流、視訊會議或視訊監控等)中趨向普及，有對研發出可以增加視訊資料之壓縮效率的視訊編碼

工具之持續需求。

【0004】舉例而言，視訊監控應用在許多應用情境(例如，保全，交通，或環境監控等)中被越來越多且廣泛地使用，以及監控裝置的數量和解析度持續快速成長。許多視訊監控應用情境傾向對使用者提供HD視訊，以擷取更多的資訊，這是因為HD視訊每訊框有較多可以擷取此等資訊之像素。然而，HD視訊位元流可能具有高位元率，這用於傳輸會需要高頻寬且用於儲存會需要大容量。舉例而言，針對現實生活傳輸而言，平均具有1920×1080解析度之監控視訊串流可能需要高達4Mbps之頻寬。另外，視訊監控通常是全天候連續監控，假若要儲存該視訊資料的話，這會對儲存系統產生很大的挑戰。對HD視訊之高頻寬與大儲存空間之需求因此變成在視訊監控中大規模佈署HD視訊之最大限制因素。

【發明內容】

【0005】本揭露之實施例提供一種用於處理視訊內容之方法。該方法可包括：接收與畫面相關聯之色度區塊與亮度區塊；判定與該亮度區塊相關聯之亮度縮放資訊；基於該亮度縮放資訊判定色度縮放因素；以及使用該色度縮放因素處理該色度區塊。

【0006】本揭露之實施例提供一種用於處理視訊內容之設備。該設備可包括：記憶體，其儲存一組指令；以及處理器，其耦接到該記憶體且經組態以執行該組指令以令

該設備執行：接收與畫面相關聯之色度區塊與亮度區塊；判定與該亮度區塊相關聯之亮度縮放資訊；基於該亮度縮放資訊判定色度縮放因素；以及使用該色度縮放因素處理該色度區塊。

【0007】 本揭露之實施例提供一種非暫態電腦可讀儲存媒體，其儲存一組可由裝置之一或多處理器執行之指令，以令該裝置執行用於處理視訊內容之方法。該方法包括：接收與畫面相關聯之色度區塊與亮度區塊；判定與該亮度區塊相關聯之亮度縮放資訊；基於該亮度縮放資訊判定色度縮放因素；以及使用該色度縮放因素處理該色度區塊。

【圖式簡單說明】

【0008】 本揭露之各種態樣與實施例被描繪於以下詳細說明及隨附圖式中。該等圖式中所示各種特徵並未依照比例繪製。

【0009】 **[圖 1]**描繪根據本揭露若干實施例的例示性視訊序列之結構。

【0010】 **[圖 2A]**描繪根據本揭露若干實施例的例示性編碼處理之示意圖。

【0011】 **[圖 2B]**描繪根據本揭露若干實施例的另一例示性編碼處理之示意圖。

【0012】 **[圖 3A]**描繪根據本揭露若干實施例的例示性解碼處理之示意圖。

【0013】[圖 3B]描繪根據本揭露若干實施例的另一例示性解碼處理之示意圖。

【0014】[圖 4]描繪根據本揭露若干實施例用於編碼或解碼視訊之例示性設備的方塊圖。

【0015】[圖 5]描繪根據本揭露若干實施例之具色度縮放的例示性亮度映射 (luma mapping with chroma scaling, LMCS)處理之示意圖。

【0016】[圖 6]描繪根據本揭露若干實施例的針對 LMCS分段線性模型之片(tile)群組層級的語法表。

【0017】[圖 7]描繪根據本揭露若干實施例的針對 LMCS分段線性模型之另一片群組層級的語法表。

【0018】[圖 8]是根據本揭露若干實施例的編碼樹單元語法結構之表。

【0019】[圖 9]是根據本揭露若干實施例的雙樹分割語法結構之表。

【0020】[圖 10]描繪根據本揭露實施例的簡化亮度預測區塊取平均值的實例。

【0021】[圖 11]是根據本揭露若干實施例的編碼樹單元語法結構之表。

【0022】[圖 12]是根據本揭露若干實施例的針對在片群組層級之 LMCS分段線性模型的調整後發訊之語法元素表。

【0023】[圖 13]是根據本揭露若干實施例用於處理視訊內容的方法之流程圖。

【0024】[圖 14]是根據本揭露若干實施例用於處理視訊內容的方法之流程圖。

【0025】[圖 15]是根據本揭露若干實施例用於處理視訊內容的另一方法之流程圖。

【0026】[圖 16]是根據本揭露若干實施例用於處理視訊內容的另一方法之流程圖。

【實施方式】

【0027】現在將詳細參照例示性實施例，其實例被描繪於所附圖式中。以下說明參考所附圖式，其中除非另外表示，否則不同圖式中的相同數字表示相同或相似的元件。在以下例示性實施例的描述中闡述的實施方式並不代表與本發明一致的所有實施方式。反而，其僅是與如所附請求項所載之本發明相關態樣一致之設備與方法之實例。除非另外註明，否則術語「或」涵蓋所有可能組合，除了不可行處。舉例而言，如果主張組件可以包括 A 或 B，則除非另有明確說明或不可行，否則組件可以包括 A 或 B，或 A 和 B。作為第二個實例，如果主張組件可以包括 A、B 或 C，則除非另有明確說明或不可行，否則組件可以包括 A 或 B 或 C，或 A 和 B，或 A 和 C，或 B 和 C，或 A 和 B 和 C。

【0028】視訊是一組靜態畫面(或「訊框」)，其經配置以按時間序列儲存視覺資訊。視訊擷取裝置(例如，攝影機)可用以擷取並按時間序列儲存此等畫面，以及視訊播放裝置(例如，電視機、電腦、智慧型手機、平板電

腦、視訊播放器、或任何具顯示功能之終端使用者端)可用以按時間序列顯示此等畫面。另外，在若干應用中，視訊擷取裝置可現實生活將擷取到視訊傳輸至視訊播放裝置(例如具螢幕之電腦)，諸如用於監視、會議或現實生活廣播。

【0029】 為了降低此等應用所需儲存空間與傳輸頻寬，在儲存與傳輸前可將視訊壓縮並在顯示前將其解壓縮。該壓縮與解壓縮可由被處理器(例如，通用電腦之處理器)執行的軟體或特殊硬體實作。用於壓縮之模組通常被稱為「編碼器」，以及用於解壓縮之模組通常被稱為「解碼器」。編碼器與解碼器可共同地被稱為「編解碼器」。編碼器與解碼器可被實作成各式各樣適當硬體、軟體、或其組合之任意者。舉例而言，編碼器與解碼器之硬體實作可包括電路，諸如一或多微處理器、數位訊號處理器(DSP)、應用特定積體電路(ASIC)、現場可程式化閘陣列(FPGA)、離散邏輯或其任意組合。編碼器與解碼器之軟體實作可包括程式碼、電腦可執行指令、韌體、或固定於電腦可讀媒體中的任何適當電腦實作演算法或處理。視訊壓縮和解壓縮可以透過各種演算法或標準來實現，例如MPEG-1，MPEG-2，MPEG-4，H.26x系列等。在若干應用中，編解碼器可從第一編碼標準將視訊解壓縮，並使用第二編碼標準將該解壓縮視訊進行再壓縮，在這種情況下編解碼器可被稱為「變換編碼器」。

【0030】 視訊編碼處理可識別並保留可用以重建畫面

之有用資訊，並忽略與該重建無關的資訊。若經忽略，該不重要資訊無法被完全重建，此種編碼處理可被稱為「有損(lossy)」。否則，其被稱作「無損(lossless)」。大部分編碼處理為有損，其是將所需儲存空間與傳輸頻寬降低之取捨。

【0031】被編碼之畫面(被稱為「當前畫面」)的有用資訊包括相對於參考畫面(例如，先前編碼與重建之畫面)之變化。這樣的變化可以包括像素的位置變化、亮度變化或顏色變化，其中最關注位置之變化。一群組像素之位置變化代表一物件可以反映該物件在參考畫面和當前畫面之間的運動。

【0032】沒有參照另一畫面即進行編碼之畫面(亦即，其是其自身參考畫面)被稱為「I-picture」。使用先前畫面作為參考畫面而進行編碼之畫面被稱為「P-picture」。同時使用先前畫面與未來畫面作為參考畫面而進行編碼(亦即，該參考是雙向)之畫面被稱為「B-picture」。

【0033】如前述，使用HD視訊之視訊監控面臨高頻寬與大儲存空間之挑戰。為解決此等挑戰，編碼視訊之位元率可被降低。在I-picture、P-picture、B-picture中，I-picture具有最高位元率。由於大多數監控視訊的背景幾乎都是靜態的，因此降低編碼視訊總位元率的一種方法可以是使用較少的I-picture進行視訊編碼。

【0034】然而，使用較少I-picture之改進可能是微不

足道的，因為 I-picture 典型上在編碼視訊中不常見。舉例而言，在典型視訊位元流中，I-picture、B-picture、P-picture 之比例可為 1:20:9，其中 I-picture 可佔少於 10% 之總位元率。換言之，在此實例中，即使移除全部的 I-picture，降低位元率也可不多於 10%。

【0035】本揭露提供用於視訊監控之基於特徵的視訊處理之方法、設備及系統。本文中「特徵」可指與畫面中視訊內容相關聯之內容特徵，與將畫面編碼或解碼的運動估計相關聯之運動特徵，或兩者。舉例而言，在視訊序列之一或多連續畫面中內容特徵可以是像素，該像素與畫面中至少一物件、場景、或環境事件相關。針對另一實例，運動特徵可包括與視訊編碼處理相關聯之資訊，將在下文描述其實例。

【0036】在本揭露中，當編碼視訊序列之畫面時，可使用特徵分類器以偵測與分類視訊序列之畫面的一或多特徵。不同類型之特徵可與不同優先度層級相關聯，該不同優先度層級進而與不同編碼位元率相關聯。不同優先度層級可與用於編碼之不同參數集相關聯，其可造成不同編碼品質層級。優先度層級越高，其相關參數集可產生的視訊品質越高。透過使用此基於特徵的視訊處理，可針對監控視訊大幅降低位元率而不會造成顯著資訊遺失。此外，本揭露之實施例可針對不同應用情境(例如，保全、交通、環境監控等)客製化優先度層級與參數集之對應關係，藉此大幅度改善視訊編碼品質並大幅度降低頻寬與儲存空間

之成本。

【0037】圖1描繪根據本揭露若干實施例的例示性視訊序列100之結構。視訊序列100可以是直播(live)視訊或已經擷取並歸檔之視訊。視訊100可以是現實生活視訊、電腦產生視訊(例如，電腦遊戲視訊)或其組合(例如，具增廣實境特效之現實生活視訊)。可從視訊擷取裝置(例如，攝影機)、包含先前擷取視訊之視訊歸檔檔案(video archive)(例如，儲存於儲存裝置中的視訊檔案)、或從視訊饋送介面(例如，視訊廣播收發器)輸入視訊序列100，以從視訊內容提供者接收視訊。

【0038】如圖1所示，視訊序列100可包括一系列之畫面，其沿著時間軸按時間配置，包括畫面102、104、106、及108。畫面102-106是連續的，且在畫面106與108間可有更多畫面。在圖1中，畫面102為I-picture，而其參考畫面是畫面102本身。畫面104是P-picture，其參考畫面是畫面102，如箭頭所指示。畫面106是B-picture，其參考畫面是畫面104與108，如箭頭所指示。在若干實施例中，一畫面(例如，畫面104)之參考畫面可能不是在該畫面立即之前或之後。舉例而言，畫面104之參考畫面可以是畫面102之前的畫面。應注意，畫面102-106之參考畫面僅為例示性，且本揭露並不限制參考畫面之實施例為圖1所示之實例。

【0039】典型上而言，由於此類工作的計算複雜度之緣故，視訊編解碼器不會一次將整個畫面編碼或解碼。反

而，其會將畫面分割成基本片段，並逐片段將畫面編碼或解碼。此種基本片段在本揭露中被稱為基本處理單元(basic processing unit, 「BPU」)。舉例而言，圖1中結構110示出視訊序列100之畫面(例如，畫面102-108中任意者)的實例結構。在結構110中，畫面被劃分成 4×4 個基本處理單元，以虛線示出其邊界。在若干實施例中，若干視訊編碼標準中(例如，MPEG family、H.261、H.263、或H.264/AVC)的該基本處理單元可以被稱為「巨集區塊(macroblock)」，或在若干其他視訊編碼標準中(例如，H.265/HEVC或H.266/VVC)被稱作「編碼樹單元(coding tree unit, CTU)」。在一畫面中，基本處理單元可具有可變尺寸，諸如 128×128 、 64×64 、 32×32 、 16×16 、 4×8 、 16×32 或具任何任意形狀與尺寸之像素。該基本處理單元之尺寸與形狀可以是根據編碼效率之平衡與基本處理單元中保留細節之層級而針對一畫面進行選擇。

【0040】該基本處理單元可以是邏輯單元，其可以包括儲存於電腦記憶體中(例如，在視訊訊框緩衝器中)一群組不同類型之視訊資料。舉例而言，有色畫面之基本處理單元可包括：亮度分量(Y)，其表示無彩亮度資訊；一或多色度分量(例如，Cb與Cr)，其表示顏色資訊；以及相關聯語法元素，其中該亮度與色度分量可具有與基本處理單元相同的尺寸。在若干視訊編碼標準(例如，H.265/HEVC或H.266/VVC)中，亮度與色度分量可被稱為「編碼樹區塊」(「CTB」)。對基本處理單元執行之任何操作都可以

對其亮度與色度分量中各者重複執行。

【0041】視訊編碼具有多階段之操作，例如將在圖2A-2B與3A-3B中詳細說明之實例。針對各階段，基本處理單元之尺寸仍可能是過大以致不適處理，且因此可將該單元進一步劃分成在本揭露中被稱作「基本處理子單元」之片段。在若干實施例中，若干視訊編碼標準中(例如，MPEG family、H.261、H.263、或H.264/AVC)，該基本處理子單元可以被稱為「區塊(block)」，或在若干其他視訊編碼標準中(例如，H.265/HEVC或H.266/VVC)被稱作「編碼單元(coding unit, CU)」。基本處理子單元可具有與基本處理單元相同或更小之尺寸。與基本處理單元相似，基本處理子單元亦可以是邏輯單元，其可以包括儲存於電腦記憶體中(例如，在視訊訊框緩衝器中)一群組不同類型之視訊資料(例如，Y、Cb、Cr、以及相關聯語法元素)。對基本處理子單元執行之任何操作都可以對其亮度與色度分量中各者重複執行。應注意，取決於處理需求，此種劃分可以被執行到進一步程度。亦應注意，不同階段可以使用不同方案來劃分該基本處理單元。

【0042】舉例而言，在模式決定階段(其實例將詳細描繪於圖2B)，編碼器可決定要對基本處理單元使用何種預測模式(例如，框內畫面預測或框間畫面預測)，其可能過大以致難以做出此決定。編碼器可將基本處理單元分割成多個基本處理子單元(例如，於H.265/HEVC或H.266/VVC中的CU)，並決定針對各個別基本處理子單元

之預測類型。

【0043】針對另一實例，在預測階段(其實例將詳細描繪於圖2A)，編碼器可在基本處理子單元(例如，CU)層級執行預測操作。然而，在若干情況下基本處理子單元仍可能過大以致難以處理。編碼器可進一步將基本處理子單元分割成較小片段(例如，在H.265/HEVC或H.266/VVC中被稱為「預測區塊」或「PB」)，在該層級上可執行預測操作。

【0044】針對另一實例，在變換階段(其實例將詳細描繪於圖2A)，編碼器可對殘差的基本處理子單元(例如，CU)執行變換操作。然而，在若干情況下基本處理子單元仍可能過大以致難以處理。編碼器可進一步將基本處理子單元分割成較小片段(例如，在H.265/HEVC或H.266/VVC中被稱為「變換區塊」或「TB」)，在該層級上可執行變換操作。應注意，相同基本處理子單元在預測階段與變換階段之劃分方案可不同。舉例而言，在H.265/HEVC或H.266/VVC中，相同CU之預測區塊與變換區塊可具有不同尺寸與數量。

【0045】在圖1之結構110中，基本處理單元112被進一步劃分成3×3個基本處理子單元，以點線示出其邊界。相同畫面之不同的基本處理單元可以不同方案被劃分成基本處理子單元。

【0046】在若干實作中，為了向視訊編碼與解碼提供並行處理之能力與誤差恢復力，畫面可被劃分成區以用於

處理，以使得針對該畫面之一區，該編碼或解碼處理可不取決於來自該畫面任何其他區之資訊。換言之，畫面各區可被獨立處理。藉此，編解碼器可並行處理畫面之不同區，並因此增加編碼效率。另外，當處理中一區之資料損壞或在網路傳輸中遺失時，編解碼器可將相同畫面之其他區正確編碼或解碼，而不會倚賴損壞或遺失之資料，藉此提供誤差恢復之能力。在若干視訊編碼標準中，畫面可被劃分成不同類型之區。舉例而言，H.265/HEVC與H.266/VVC提供兩種類型之區：「切片(slice)」與「片(tile)」。應註明，視訊序列100之不同畫面可具有不同的分割方案用於將一畫面劃分成區。

【0047】舉例而言，在圖1中，結構110被劃分成三個區114、116及118，其在結構110中被以實線示出其邊界。區114包括四個基本處理單元。區116與118各包括六個基本處理單元。應注意，圖1中結構110之基本處理單元、基本處理子單元、以及區僅為例示性，且本揭露並不限制於其實施例。

【0048】圖2A描繪根據本揭露若干實施例的例示性編碼處理200A之示意圖。根據處理200A，編碼器可將視訊序列202編碼成視訊位元流228。相似於圖1之視訊序列100，視訊序列202可包括按時間順序配置之一組畫面(稱作「原始畫面」)。相似於圖1之結構110，視訊序列202之各原始畫面可被編碼器劃分成基本處理單元、基本處理子單元、或區，以供處理用。在若干實施例中，編碼器可對

視訊序列 202 之各原始畫面執行在基本處理單元之層級上的處理 200A。舉例而言，編碼器可迭代執行處理 200A，其中編碼器可在處理 200A 之一迭代中將基本處理單元編碼。在若干實施例中，編碼器可對視訊序列 202 之各原始畫面之區(例如，區 114-118)並行執行處理 200A。

【0049】在圖 2A 中，編碼器可將視訊序列 202 之原始畫面的基本處理單元(稱作「原始 BPU」)饋送到預測階段 204，以產生預測資料 206 與預測 BPU 208。編碼器可從原始 BPU 減去預測 BPU 208 以產生殘差 BPU 210。編碼器可將殘差 BPU 210 饋送到變換階段 212 與量化階段 214，以產生量化變換係數 216。編碼器可將預測資料 206 與量化變換係數 216 饋送到二進制編碼階段 226，以產生視訊位元流 228。分量 202、204、206、208、210、212、214、216、226 及 228 可被稱為「轉送路徑」。在處理 200A 期間，編碼器在量化階段 214 後可將量化變換係數 216 饋送到反量化階段 218 與反變換階段 220，以產生重建殘差 BPU 222。編碼器可將重建殘差 BPU 222 添加到預測 BPU 208，以產生預測參考 224，該預測參考 224 在預測階段 204 中被使用以用於處理 200A 之下一迭代。處理 200A 之分量 218、220、222、及 224 可被稱為「重建路徑」。該重建路徑可用以確保編碼器與解碼器兩者皆使用相同的參考資料以用於預測。

【0050】編碼器可迭代執行處理 200A 以將原始畫面之各原始 BPU(在轉送路徑中)編碼，並且產生預測參考 224 以用於編碼原始畫面之下個原始 BPU(在重建路徑中)。在將

原始畫面之所有原始BPU編碼後，編碼器可繼續編碼視訊序列202中下個畫面。

【0051】參照處理200A，編碼器可接收由視訊擷取裝置(例如，攝影機)所產生的視訊序列202。本文所用術語「接收」可指稱接收、輸入、獲取、取得、獲得、讀取、存取或任意其他任何形式將資料輸入的動作。

【0052】在預測階段204，於當前迭代時，編碼器可接收原始BPU與預測參考224，並執行預測操作以產生預測資料206與預測BPU 208。可從處理200A之先前迭代的重建路徑產生預測參考224。預測階段204之目的在於透過從預測資料206與預測參考224中擷取可用以將原始BPU重建成預測BPU 208之預測資料206，來降低資訊冗餘。

【0053】理想上，預測BPU 208可與原始BPU相同。然而，由於不理想之預測與重建操作，預測BPU 208通常會與原始BPU有些微不同。為了記錄此等差異，在產生預測BPU 208後，編碼器可從原始BPU減去其以產生殘差BPU 210。舉例而言，編碼器可從原始BPU之對應像素的值減去預測BPU 208之像素的值(例如，灰階值或RGB值)。原始BPU與預測BPU 208之對應像素間的此等減法會導致殘差BPU 210之各像素可具有殘差值。與原始BPU相比之下，預測資料206與殘差BPU 210可具有較少的位元，但其可被用以重建原始BPU而不會有顯著品質劣化。因此，原始BPU被壓縮。

【0054】為了進一步壓縮殘差BPU 210，在變換階段

212，編碼器可透過將殘差BPU 210分解成一組二維「基礎圖案」，以降低殘差BPU 210之空域冗餘，各基礎圖案與「變換係數」相關聯。該基礎圖案可具有相同尺寸(例如，殘差BPU 210之尺寸)。各基礎圖案可表示殘差BPU 210之變異頻率(例如，亮度變異之頻率)分量。不能從任何其他基礎圖案的任何組合(例如，線性組合)重現此等基礎圖案中任一者。換言之，該分解可將殘差BPU 210之變異分解成頻率域。此種分解類似於函數的離散傅立葉變換，其中基礎圖案類似於離散傅立葉變換的基礎函數(例如，三角函數)，以及變換係數類似於與基處函數相關聯的係數。

【0055】不同變換演算法可使用不同的基礎圖案。各種變換演算法可用於變換階段212，諸如，離散餘弦變換、離散正弦變換等。變換階段212之變換為可逆的。亦即，編碼器可透過變換之反向操作(被稱為「反變換」)以復原殘差BPU 210。舉例而言，為了復原殘差BPU 210之像素，反變換可將基礎圖案的相應像素的值乘以個別的相關係數，然後將乘積相加以產生加權和。針對一視訊編碼標準，編碼器與解碼器兩者可使用相同的變換演算法(因此，相同的基礎圖案)。因此，編碼器可僅記錄變換係數，該解碼器可從其重建出殘差BPU 210，而無須從編碼器接收基礎圖案。與殘差BPU 210相比之下，變換係數可具有較少的位元，但其可被用以重建殘差BPU 210而不會有顯著品質劣化。因此，將殘差BPU 210進一步壓縮。

【0056】在量化階段214，編碼器可進一步將變換係數壓縮。在變換處理中，不同的基礎圖案可表示不同的變異頻率(例如，亮度變異頻率)。由於人眼通常較容易辨識出低頻率的變異，因此編碼器可忽略高頻率變異之資訊而不會造成解碼中的顯著品質劣化。舉例而言，在量化階段214，編碼器可透過將各變換係數除以整數值(被稱為「量化參數」)並將商四捨五入取最近整數值，來產生量化變換係數216。經過這樣的操作，可以將高頻率基礎圖案的一些變換係數變換成零，並且可以將低頻率基礎圖案的變換係數變換成較小的整數。編碼器可以忽略零值量化變換係數216，藉此進一步壓縮該變換係數。量化處理亦為可逆的，其中可以將量化變換係數216重建成在量化的反向操作(稱為「反量化」)中的變換係數。

【0057】因為在四捨五入運算中，編碼器會忽略此種除法的餘數，故量化階段214可為有損。典型上，在處理200A中量化階段214可貢獻最多的資訊損失。資訊損失越大，量化變換係數216需要越少的位元。為獲得不同層級之資訊損失，編碼器可使用不同值的量化參數或量化處理之任何參數。

【0058】在二進制編碼階段226，編碼器可使用二進制編碼技術將預測資料206及量化變換係數216編碼，該二進制編碼技術諸如熵編碼、可變長度編碼、算數編碼、霍夫曼編碼、情境自適應二進制算術編碼或任何其他無損或有損壓縮演算法。在若干實施例中，除了預測資料206與

量化變換係數216外，編碼器還可在二進制編碼階段226將其他資訊編碼，該其他資訊諸如在預測階段204使用的預測模式、預測操作的參數、在變換階段212之變換類型、量化處理的參數(例如，量化參數)、或編碼器控制參數(例如，位元率控制參數)等。編碼器可使用二進制編碼階段226之輸出資料以產生視訊位元流228。在若干實施例中，視訊位元流228可被進一步分封化(packetize)，用於網路傳輸。

【0059】參照處理200A之重建路徑，編碼器在反量化階段218可對量化變換係數216執行反量化，以產生重建變換係數。在反變換階段220，編碼器可基於重建的變換係數產生重建殘差BPU 222。編碼器可將重建殘差BPU 222添加到預測BPU 208，以產生預測參考224，該預測參考224在處理200A之下一迭代被使用。

【0060】應注意，處理200A之其他變異可用以將視訊序列202編碼。在若干實施例中，編碼器可以不同順序執行處理200A之階段。在若干實施例中，處理200A之一或多階段可被結合成單一階段。在若干實施例中，處理200A之單一階段可以被劃分成多個階段。舉例而言，變換階段212與量化階段214可被結合成單一階段。在若干實施例中，處理200A可包括附加階段，在若干實施例中，處理200A可省略圖2A中的一或多階段。

【0061】圖2B描繪根據本揭露若干實施例的另一例示性編碼處理200B之示意圖。處理200B可以是調整自處理

200A。舉例而言，處理200B可由編碼器使用以符合混合視訊編碼標準(例如，H.26x系列)。與處理200A相比，處理200B之轉送路徑額外地包括模式決定階段230並將預測階段204劃分成空域預測階段2042與時域預測階段2044。處理200B之重建路徑額外包括迴路過濾階段232與緩衝器234。

【0062】一般而言，預測技術可被分類成兩種類型：空域預測與時域預測。空域預測(例如，框內畫面預測或「框內預測(intra prediction)」)可使用來自相同畫面中一或多已經被編碼的相鄰BPU之像素來預測當前BPU。亦即，空域預測中的預測參考224可包括相鄰BPU。空域預測可降低畫面之固有空域冗餘。時域預測(例如，框間畫面預測或「框間預測(inter prediction)」)可使用來自一或多已經被編碼畫面之區來預測當前BPU。亦即，時域預測中的預測參考224可包括編碼畫面。時域預測可降低數畫面之固有時域冗餘。

【0063】參照處理200B，在轉送路徑中，編碼器在空域預測階段2042與時域預測階段2044執行預測操作。舉例而言，在空域預測階段2042，編碼器可執行框內預測。針對正被編碼之畫面的原始BPU，預測參考224可包括相同畫面中已經被編碼(於轉送路徑中)與重建(於重建路徑中)之一或多相鄰BPU。透過將該相鄰BPU外推，編碼器可產生預測BPU 208。該外推技術可舉例而言包括線性外推或內插、多項式外推或內插等等。在若干實施例中，編碼器

可執行像素層級之外推，諸如透過針對預測BPU 208之各像素外推對應像素之值。用於外推之相鄰BPU可位於相對於原始BPU之各種方向，諸如垂直方向(例如，在原始BPU上方)、水平方向(例如，在原始BPU左側)、對角線方向(例如，在原始BPU左下方、右下方、左上方或右上方)、或所用視訊編碼標準中定義之任何方向。針對框內預測，預測資料206可包括例如所用相鄰BPU之位置(例如，座標)、所用相鄰BPU之尺寸、外推之參數、或所用相鄰BPU相對於原始BPU之方向等等。

【0064】針對另一實例，在時域預測階段2044，編碼器可執行框間預測。針對當前畫面的原始BPU，預測參考224可包括已經被編碼(於轉送路徑中)與重建(於重建路徑中)之一或多畫面(被稱為「參考畫面」)。在若干實施例中，可逐BPU將參考畫面編碼及重建。舉例而言，編碼器可將重建殘差BPU 222添加到預測BPU 208，以產生重建的BPU。當產生相同畫面之所有重建BPU後，編碼器可生成重建畫面作為參考畫面。編碼器可執行「運動估計」之操作，以搜尋參考畫面範圍(被稱為「搜尋窗口」)中的匹配區。可基於當前畫面中原始BPU之位置，判定參考畫面中搜尋窗口的位置。舉例而言，搜尋窗口在參考畫面中可中心位於與當前畫面中原始BPU具相同座標之位置，並且可擴展一預定距離。當編碼器在搜尋窗口中識別出(例如，透過使用pel-recursive演算法或區塊匹配演算法等)與原始BPU相似之區時，編碼器可將此區判定成匹配區。匹配

區可具有與原始BPU不同的維度(例如，更小、相等、更大、或具不同形狀)。因為參考畫面與當前畫面是在時間軸上時間分離的(如圖1示出)，故可理解成隨時間推移該匹配區「移動」到原始BPU之位置。編碼器可將此種運動之方向與距離記錄成「運動向量」。當使用多個參考畫面(如圖1之畫面106)時，編碼器可搜尋匹配區並針對各參考畫面判定其相關聯運動向量。在若干實施例中，編碼器可對各別的匹配參考畫面之匹配區的像素值指定權重。

【0065】運動估計可用以識別各種類型之運動(諸如，平移、旋轉、或縮放等)。針對框間預測，預測資料206可例如包括匹配區之位置(例如，座標)、與匹配區相關聯之運動向量、參考畫面之數量、或與參考畫面相關聯之權重等等。

【0066】為了要產生預測BPU 208，編碼器可執行「運動補償」之操作。運動補償可用以基於預測資料206(例如，運動向量)與預測參考224重建預測BPU 208。舉例而言，編碼器可根據運動向量移動參考畫面之匹配區，其中編碼器可預測當前畫面之原始BPU。當使用多個參考畫面(如圖1之畫面106)時，編碼器可根據各別的運動向量與匹配區之平均像素值來移動參考畫面之匹配區。在若干實施例中，若編碼器已對各別的匹配參考畫面之匹配區的像素值指定權重，則編碼器可將所動匹配區之像素值的加權總和相加。

【0067】在若干實施例中，框間預測可以是單向或雙

向。單向框間預測可使用相對於當前畫面在相同時間方向中之一或多參考畫面。舉例而言，圖1中畫面104是單向框間預測畫面，其中該參考畫面(亦即，畫面102)在畫面104之前。雙向框間預測可使用相對於當前畫面在兩時間方向中之一或多參考畫面。舉例而言，圖1中畫面106是雙向框間預測畫面，其中該參考畫面(亦即，畫面104與108)在相對於畫面104之兩個時間方向中。

【0068】仍參照處理200B之轉送路徑，在空域預測2042與時域預測階段2044之後，編碼器於模式決定階段230時可選擇用於處理200B之當前迭代的預測模式(例如，框內預測或框間預測中之一)。舉例而言，編碼器可執行率失真最佳化技術，其中編碼器可取決於候選預測模式的位元率與候選預測模式下的重建參考畫面的畸變，可選擇出將成本函數之值最小化之預測模式。取決於所選預測模式，編碼器可產生對應預測BPU 208與預測資料206。

【0069】在處理200B之重建路徑中，若已經在轉送路徑中選定框內預測模式，則在產生預測參考224(例如，當前BPU，其已在當前畫面中被編碼與重建)之後，編碼器可將預測參考224直接饋送到空域預測階段2042以供稍後使用(例如，用於當前畫面之下個BPU的外推)。若已經在轉送路徑中選定框間預測模式，則在產生預測參考224(例如，當前畫面，其中所有BPU已經被編碼與重建)之後，編碼器可將預測參考224饋送到迴路過濾階段232，編碼器在該階段可對預測參考224施加迴路過濾以降低或消去由

框間預測所引入之畸變(例如，區塊化產物(blocking artifact))。在迴路過濾階段232，編碼器可施加各種迴路過濾技術，諸如去區塊、樣本適應性偏移、或適應性迴路過濾等。迴路過濾的參考畫面可以被儲存於緩衝器234中(或「解碼畫面緩衝器」)以供稍後使用(例如，可用作用於視訊序列202之未來畫面的框間預測參考畫面)。編碼器可將一或多參考畫面儲存於緩衝器234中，以在時域預測階段2044使用其。在若干實施例中，編碼器可在二進制編碼階段226將迴路過濾之參數(例如，迴路過濾強度)編碼，並連同編碼量化變換係數216、預測資料206以及其他資訊。

【0070】圖3A描繪根據本揭露若干實施例的例示性解碼處理300A之示意圖。處理300A可以是對應於圖2A中壓縮處理200A之解壓縮處理。在若干實施例中，處理300A可以與處理200A之重建路徑相似。根據處理300A，編碼器可將視訊位元流228解碼成視訊串流304。視訊串流304可以與視訊序列202極其相似。然而，由於壓縮與解壓縮處理(例如，圖2A-2B中量化階段214)中資訊損失之緣故，因此一般來說視訊串流304不會與視訊序列202相同。與圖2A-2B中處理200A與200B相似，解碼器可對視訊位元流228中編碼的各畫面執行在基本處理單元(BPU)層級之處理300A。舉例而言，解碼器可迭代執行處理300A，其中解碼器可在處理300A之一迭代中將基本處理單元解碼。在若干實施例中，解碼器可對視訊位元流228中編碼的各畫面

之區(例如，區114-118)並行執行處理300A。

【0071】在圖3A中，解碼器可將與編碼畫面的基本處理單元(被稱為「編碼BPU」)相關聯之視訊位元流228之一部分饋送到二進制解碼階段302。在二進制解碼階段302，解碼器可將該部分解碼成預測資料206與量化變換係數216。解碼器可將量化變換係數216饋送到反量化階段218與反變換階段220，以產生重建殘差BPU 222。解碼器可將預測資料206饋送到預測階段204，以產生預測BPU 208。解碼器可將重建殘差BPU 222添加到預測BPU 208，以產生預測參考224。在若干實施例中，預測參考224可以被儲存於緩衝器(例如，電腦記憶體中解碼畫面緩衝器)中。解碼器可將預測參考224饋送到預測階段204，用於在處理300A之下一個迭代執行預測操作。

【0072】解碼器可迭代執行處理300A以將編碼畫面之各編碼BPU解碼，並且產生預測參考224以用於將編碼畫面之下個編碼BPU進行編碼。在將編碼畫面之所有編碼BPU解碼後，解碼器可將畫面輸出到視訊串流304供顯示用，並繼續解碼視訊位元流228中的下一個編碼畫面。

【0073】在二進制解碼階段302，解碼器可執行編碼器所用二進制編碼技術之反向操作(該二進制編碼技術諸如熵編碼、可變長度編碼、算數編碼、霍夫曼編碼、情境自適應二進制算術編碼或任何其他無損壓縮演算法)。在若干實施例中，除了預測資料206與量化變換係數216外，解碼器還可在二進制解碼階段302將其他資訊解碼，該其

他資訊諸如預測模式、預測操作的參數、變換類型、量化處理的參數(例如，量化參數)、或編碼器控制參數(例如，位元率控制參數)等。在若干實施例中，若透過網路以封包傳輸視訊位元流228，則解碼器可在將該視訊位元流228饋送到二進制解碼階段302之前將其去封包化(depacketize)。

【0074】圖3B描繪根據本揭露若干實施例的另一例示性解碼處理300B之示意圖。處理300B可以是調整自處理300A。舉例而言，處理300B可由解碼器使用以符合混合視訊編碼標準(例如，H.26x系列)。與處理300A相比，處理300B額外地將預測階段204劃分成空域預測階段2042與時域預測階段2044，並且額外地包括迴路過濾階段232與緩衝器234。

【0075】在處理300B中，針對正被解碼之編碼畫面(稱作「當前畫面」)的編碼基本處理單元(稱作「當前BPU」)，由解碼器從二進制解碼階段302解碼出的預測資料206可包括各種類型之資料，其取決於編碼器是使用何種預測模式來將當前BPU編碼。舉例而言，若編碼器使用框內預測來將當前BPU編碼，則預測資料206可包括預測模式指示符(例如，旗標值)，該指示符指示框內預測、或框內預測操作之參數等。框內預測操作之參數可包括例如用作參考之一或多相鄰BPU之位置(例如，座標)、相鄰BPU之尺寸、外推之參數、或相鄰BPU相對於原始BPU之方向等等。作為另一實例，若編碼器使用框間預測來將當

前BPU編碼，則預測資料206可包括預測模式指示符(例如，旗標值)，該指示符指示框間預測、或框間預測操作之參數等。框間預測操作之參數可包括，例如，與當前BPU相關聯之參考畫面的數量、與參考畫面各別相關聯之權重、各別參考畫面中一或多匹配區的位置(例如，座標)、或與匹配區各別相關聯的一或多運動向量等等。

【0076】基於預測模式指示符，解碼器可決定是否在空域預測階段2042執行空域預測(例如，框內預測)，或在時域預測階段2044執行時域預測(例如，框間預測)。圖2B中已描述執行此等空域預測或時域預測之細節，因此並不會在下文重複其描述。在執行此等空域預測或時域預測之後，解碼器可產生預測BPU 208。解碼器可將預測BPU 208與重建殘差BPU 222相加，以產生預測參考224，如圖3A所示。

【0077】在處理300B中，解碼器可將預測參考224饋送到空域預測階段2042或時域預測階段2044，用於在處理300B之下一個迭代執行預測操作。舉例而言，若在空域預測階段2042使用框內預測將當前BPU解碼，則在產生預測參考224(例如，解碼的當前BPU)之後，解碼器可將預測參考224直接饋送到空域預測階段2042以供稍後使用(例如，用於當前畫面之下個BPU的外推)。若在時域預測階段2044使用框間預測將當前BPU解碼，則在產生預測參考224(例如，參考畫面，其中所有的BPU已經被解碼)之後，編碼器可將預測參考224饋送到迴路過濾階段232以降低或

消除畸變(例如，區塊化產物)。解碼器可以**圖 2B**所述方式施加迴路過濾來預測參考224。迴路過濾的參考畫面可以被儲存於緩衝器234中(例如，電腦記憶體中的解碼畫面緩衝器)以供稍後使用(例如，可用作用於視訊位元流228之未來編碼畫面的框間預測參考畫面)。解碼器可將一或多參考畫面儲存於緩衝器234中，以在時域預測階段2044使用其。在若干實施例中，當預測資料206之預測模式指示符指示框間預測被用以編碼當前BPU時，預測資料可進一步包括迴路過濾之參數(例如，迴路過濾強度)。

【0078】**圖 4**是根據本揭露若干實施例用於編碼或解碼視訊之例示性設備400的方塊圖。如**圖 4**所示，設備400可包括處理器402。當處理器402執行本文所述指令時，設備400可變成用於視訊編碼或解碼之專用機器。處理器402可以是能夠操控或處理資訊之任何類型的電路。舉例而言，處理器402可以包括任何數量的中央處理單元(或“CPU”)，圖形處理單元(或“GPU”)，神經處理單元(“NPU”)，微控制器單元(“MCU”)，光學處理器，可程式化邏輯控制器，微控制器，微處理器，數位訊號處理器，智財(IP)核心，可程式化邏輯陣列(PLA)，可程式化陣列邏輯(PAL)，通用陣列邏輯(GAL)，複雜可程式化邏輯裝置(CPLD)，現場可程式化閘陣列(FPGA)，系統單晶片(SoC)，或專用積體電路(ASIC)等之任意組合。在若干實施例中，處理器402亦可以是一組被分組成單一邏輯組件之處理器。舉例而言，如**圖 4**所示，處理器402可包括多個

處理器，包括處理器 402a、處理器 402b、以及處理器 402n。

【0079】設備 400 亦可包括記憶體 404，其經組態以儲存資料(例如，指令集、電腦碼、或中間資料等)。舉例而言，如圖 4 所示，儲存資料可包括程式指令(例如，用於實作處理 200A、200B、300A、或 300B 中階段的程式指令)與用於處理之資料(例如，視訊序列 202、視訊位元流 228、或視訊串流 304)。處理器 402 可存取程式指令與用於處理資料(例如經由匯流排 410)，並執行該程式指令以對該資料執行操作或運算以進行處理。記憶體 404 可包括高速率隨機存取儲存裝置或非揮發性儲存裝置。在若干實施例中，記憶體 404 可包括任何數量之隨機存取記憶體(RAM)、唯讀記憶體(ROM)、光碟片、磁碟、硬碟、固態硬碟、快閃驅動器、安全數位(SD)卡、記憶條、或緊湊型快閃記憶(CF)卡的任何組合。記憶體 404 亦可以是一組被分組成單一邏輯組件之記憶體(圖 4 中未示出)。

【0080】匯流排 410 可以是在設備 400 中的組件之間轉移資料之通訊裝置，該匯流排諸如內部匯流排(例如，CPU 記憶體匯流排)、或外部匯流排(例如，通用序列匯流排埠、週邊組件互連快速埠)等。

【0081】為了易於說明且不會造成不明確，在本揭露中，處理器 402 與其他資料處理電路被統稱為「資料處理電路」。資料處理電路可被完全實作成硬體、或軟體、硬體或韌體之組合。此外，資料處理電路可以是單一獨立模

組或可完全組合或部分組合成設備400之任何其他組件。

【0082】設備400可進一步包括網路介面406，以提供與網路(例如，網際網路、內聯網、區域網路、或行動通訊網路等)之有線或無線通訊。在若干實施例中，網路介面406可包括任何數量之網路介面控制器(NIC)、射頻(RF)模組、轉頻器(transponder)、收發器、數據機、路由器、閘道器、有線網路配接器、無線網路配接器、藍牙配接器、紅外線配接器、進場通訊(NFC)配接器、或蜂巢式網路晶片等之任意組合。

【0083】在若干實施例中，可選地設備400可進一步包括週邊介面408，以提供對一或多週邊裝置之連接。如圖4所示，週邊裝置可包括但不限於游標控制裝置(例如，滑鼠、觸控板或觸控螢幕)、鍵盤、顯示器(例如，陰極射線管顯示器、液晶顯示器、或發光二極體顯示器)、或視訊輸入裝置(例如，攝影機或耦接到視訊歸檔檔案之輸入介面)等等。

【0084】應注意，視訊編解碼器(例如，執行處理200A、200B、300A、或300B之編解碼器)可被實作成設備400中任何軟體或硬體模組之任意組合。舉例而言，處理200A、200B、300A、或300B之若干或所有階段可以被實作成設備400之一或多軟體模組，諸如可載入記憶體404中的程式指令。作另一實例，處理200A、200B、300A、或300B之若干或所有階段可以被實作成設備400之一或多硬體模組，諸如專用資料處理電路(例如，FPGA、ASIC、或

NPU等)。

【0085】圖5描繪根據本揭露若干實施例之具色度縮放的例示性亮度映射(luma mapping with chroma scaling, LMCS)處理500之示意圖。舉例而言，處理500可由解碼器使用以符合混合視訊編碼標準(例如，H.26x系列)。LMCS是被施加在圖2B之迴路過濾階段232之前的新處理區塊。LMCS亦可被稱為整形器(reshaper)。

【0086】LMCS處理500可包括基於適應性分段線性模型的亮度分量值之迴路內映射與色度分量的亮度相依之色度殘差縮放。

【0087】如圖5所示，基於適應性分段線性模型的亮度分量數值之迴路內映射可包括正向映射階段518與反射射階段508。色度分量的亮度相依之色度殘差縮放可包括色度縮放階段520。

【0088】映射前或反映射之後的樣本值可被稱為原始域中的樣本，且映射之後與反映射之前的樣本值可被稱為映射域中的樣本。當賦能LMCS時，處理500中的若干階段可以在映射域中執行以取代在原始域中執行。應瞭解，透過使用SPS旗標，可在序列層級將正向映射階段518與反映射階段508賦能/去能。

【0089】如圖5所示， Q^{-1} & T^{-1} 階段504、重建階段506、以及框內預測階段514可在映射域中執行。舉例而言， Q^{-1} & T^{-1} 階段504可包括反量化與反變換，重建階段506可包括亮度預測與亮度殘差之相加入，以及框內預測

階段 508 可包括亮度框內預測。

【0090】於原始(亦即，非映射)域中執行迴路過濾階段 510、運動補償階段 516 與 530、框內預測階段 528、重建階段 522、以及解碼畫面緩衝(DPB)階段 512 與 526。在若干實施例中，迴路過濾階段 510 可包括去區塊、適應性迴路過濾(ALF)、以及樣本適應性偏移(SAO)，重建階段 522 可包括色度預測與連同色度殘差之相加，以及 DPB 512 與 526 可將解碼畫面儲存作參考畫面。

【0091】在若干實施例中，可施加具分段線性模型之亮度映射。

【0092】亮度分量之迴路內映射可透過在動態範圍內重新分配碼字以提高壓縮效率，從而將輸入視訊之訊號統計數據加以調整。可透過正向映射函數「FwdMap」與對應反映射函數「InvMap」執行亮度映射。透過使用具 16 個相等塊(pieces)的分段線性模型將「FwdMap」函數發訊。「InvMap」函數不需要被發訊，而是從「FwdMap」函數衍生出。

【0093】於圖 6 之表 1 與圖 7 之表 2 示出分段線性模型之發訊。圖 6 之表 1 描繪片群組之標頭語法結構。如圖 6 所示，將整形器模型參數存在旗標發訊以指示當前片群組中是否存在亮度映射模型。若當前片群組中存在亮度映射模型，則透過使用圖 7 中表 2 所示語法元素，可將對應分段線性模型參數發訊於 tile_group_reshaper_model() 中。分段線性模型將一動態範圍之輸入訊號分割成 16 個相等塊。針對各

16個相等塊，使用分配給該塊的碼字之數量來表達該塊之線性映射參數。採用10位元的輸入當作實例。該16個塊各可具有預設分配給該塊的64個碼字。經發訊的碼字數量可用以計算縮放因素並據此對該塊調整該映射函數。**圖7**之表2亦界定最小索引「`reshaper_model_min_bin_idx`」與最大索引「`reshaper_model_max_bin_idx`」(包含上下界)，可對其發訊碼字之數量。若塊索引(`piece index`)小於`reshaper_model_min_bin_idx`或大於`reshaper_model_max_bin_idx`，則那個塊的碼字的數量沒有被發訊，並且被推斷為零(即，沒有分配碼字並且對該塊沒有施加映射/縮放)。

【0094】在將`tile_group_reshaper_model()`發訊後，於片群組的標頭層級，另一整形器賦能旗標「`tile_group_reshaper_enable_flag`」被發訊以指示如**圖8**所描繪之LMCS處理是否被施加於該當前片群組。若為了當前片群組賦能整形器且若該當前片群組不使用雙樹分割，則發訊進一步的色度縮放賦能旗標以指示是否針對當前片群組賦能色度縮放與否。雙樹分割(Dual tree partition)亦可被稱作色度分離樹。

【0095】分段線性模型可以是根據**圖7**之表2中發訊的語法元素而建構，如下述。「`FwdMap`」分段線性模型之各第*i*個塊，*i*=0、1、...、15是由兩輸入樞軸點`InputPivot[]`與兩個輸出(經映射)樞軸點`MappedPivot[]`定義。基於以下發訊語法計算出`InputPivot[]`與`MappedPivot[]`(沒有損失通用性，我們假設輸入視訊之位元深度是10位元

) :

- 1) $OrgCW = 64$
- 2) For $i = 0:16$, $InputPivot[i] = i * OrgCW$
- 3) For $i = reshaper_model_min_bin_idx : reshaper_model_max_bin_idx$, $SignaledCW[i] = OrgCW + (1 \rightarrow 2 * reshape_model_bin_delta_sign_CW[i]) * reshape_model_bin_delta_abs_CW[i]$;
- 4) For $i = 0:16$, $MappedPivot[i]$ 是如下方式計算:
 $MappedPivot[0] = 0$;
for($i = 0; i < 16; i++$)
 $MappedPivot[i + 1] = MappedPivot[i] + SignaledCW[i]$

【 0096 】 反 映 射 函 數 「 $InvMap$ 」 亦 可 透 過 $InputPivot[]$ 與 $MappedPivot[]$ 界 定 。 與 「 $FwdMap$ 」 不 同 ， 對 於 「 $InvMap$ 」 分 段 線 性 模 型 ， 可 透 過 $MappedPivot[]$ 界 定 各 塊 之 兩 個 輸 入 樞 軸 點 ， 且 可 透 過 $InputPivot[]$ 界 定 兩 個 輸 出 樞 軸 點 ， 其 與 $FwdMap$ 相 反 。 透 過 此 方 式 ， 「 $FwdMap$ 」 之 輸 入 被 分 割 成 相 等 塊 ， 但 「 $InvMap$ 」 之 輸 入 並 沒 有 被 保 證 會 被 分 割 成 相 等 塊 。

【 0097 】 如 圖 5 所 示 ， 針 對 框 間 編 碼 區 塊 ， 可 於 映 射 域 中 執 行 運 動 補 償 預 測 。 換 言 之 ， 在 運 動 補 償 預 測 516 之 後 ， Y_{pred} 是 基 於 DPB 中 參 考 訊 號 而 計 算 出 ， 可 施 加 「 $FwdMap$ 」 函 數 518 以 將 原 始 域 中 亮 度 預 測 區 塊 映 射 到 映 射 域 ， $Y'_{pred} = FwdMap(Y_{pred})$ 。 針 對 框 內 編 碼 區 塊 ， 並 沒 有 施 加 「 $FwdMap$ 」 函 數 ， 因 為 框 內 預 測 中 所 用 參 考 樣 本 已 經 在 映 射 域 中 。 在 重 建 區 塊 506 後 ， 可 計 算 出 Y_r 。 可 施 加 「 $InvMap$ 」 函 數 508 ， 以 將 映 射 域 中 重 建 亮 度 值 變 換 回 原 始 域 中 的 重 建 亮 度 值 ($\hat{Y}_i = InvMap(Y_r)$) 。 可 將 「 $InvMap$ 」 函 數 508 施 加 於 框 內 與 框 間 編 碼 亮 度 區 塊 。

【 0098 】 亮 度 映 射 處 理 (正 向 或 反 映 射) 可 透 過 使 用 查

找表(LUT)或使用現實生活計算而實作。若使用LUT，則可預先計算出並預先儲存表「FwdMapLUT[]」與「InvMapLUT[]」，以用於在片群組層級使用，且正向與反映射可分別被輕易實作成 $FwdMap(Y_{pred}) = FwdMapLUT[Y_{pred}]$ 與 $InvMap(Y_r) = InvMapLUT[Y_r]$ 。替代地，可使用現實生活計算。採取正向映射函數「FwdMap」作為實例。為了判定出亮度樣本屬於何塊，樣本數值可右移6位元(其對應於16個相等塊，假定是10位元之視訊)，以獲得該塊索引。接著，取回針對該塊之線性模型參數並現實生活施加以計算出映射亮度值。FwdMap函數可如下述評估出：

$$Y'_{pred} = FwdMap(Y_{pred}) = ((b2 - b1)/(a2 - a1)) * (Y_{pred} - a1) + b1$$

其中，“i”是塊索引，a1是InputPivot[i]，a2是InputPivot[i+1]，b1是MappedPivot[i]，b2是MappedPivot[i+1]。

【0099】「InvMap」函數可以類似的方式現實生活計算，除了在確定樣本值所屬的塊時，需要施加條件檢查而不是簡單的右移，因為映射域中的這些塊並不能保證尺寸相同。

【0100】在若干實施例中，可執行亮度相依色度殘差縮放。

【0101】色度殘差縮放經設計以補償亮度訊號與其對應色度訊號之間的交互作用。無論是否賦能色度殘差縮放，亦在片群組層級將其發訊。如圖6之表1所示，若亮度對映經賦能且若雙樹分割沒有被施加到當前片群組，則發訊附加旗標(例如，tile_group_reshaper_chroma_residual_

scale_flag)以指示亮度相依色度殘差縮放是否被賦能。當沒有使用亮度映射或當雙樹分割用於當前片群組中時，自動化去能亮度相依色度殘差縮放。此外，可針對其面積小於或等於4之色度區塊去能亮度相依色度殘差縮放。

【0102】色度殘差縮放取決於對應亮度預測區塊(針對框內與框間編碼區塊)之平均值。 $avgY'$ ，其是亮度預測區塊之平均值，可透過如下計算出：

$$avgY' = \frac{\sum_{i=0}^{width-1} \sum_{j=0}^{height-1} predSamples[i][j]}{width*height}.$$

【0103】從以下步驟計算出 $C_{ScaleInv}$ 之值：

- 1) 基於 $InvMap$ 函數找到分段線性模型之索引 Y_{Idx} ，其中 $avgY'$ 屬於該分段線性模型。
- 2) $C_{ScaleInv} = cScaleInv[Y_{Idx}]$ ，其中 $cScaleInv[]$ 是預先計算之16塊LUT。

【0104】在VTM4中的當前LMCS方法中，基於64條靜態LUT ChromaResidualScaleLut和SignaledCW [i]值，得出i範圍為0到15的預先計算的LUT $cScaleInv [i]$ ：

ChromaResidualScaleLut[64] = {16384, 16384, 16384, 16384, 16384, 16384, 16384, 8192, 8192, 8192, 8192, 5461, 5461, 5461, 5461, 4096, 4096, 4096, 4096, 3277, 3277, 3277, 3277, 2731, 2731, 2731, 2731, 2341, 2341, 2341, 2048, 2048, 2048, 1820, 1820, 1820, 1638, 1638, 1638, 1638, 1489, 1489, 1489, 1489, 1365, 1365, 1365, 1365, 1260, 1260, 1260, 1260, 1170, 1170, 1170, 1170, 1092, 1092, 1092, 1092, 1024, 1024, 1024, 1024};

shiftC = 11

– if (SignaledCW [i] == 0)

$cScaleInv [i] = (1 \ll shiftC)$

– Otherwise,

$cScaleInv [i] = ChromaResidualScaleLut[(SignaledCW [i] \gg 1)-1]$

【0105】靜態表 `ChromaResidualScaleLut[]` 包含 64 項目，以及 `SignaledCW[]` 是在 $[0, 128]$ 之範圍內(假設輸入是 10 位元)。因此，除以 2(例如，往右移 1)被用以建構色度縮放因素 `LUT cScaleInv []`。色度縮放因素 `LUT cScaleInv []` 可包含複數個色度縮放因素。`LUT cScaleInv []` 經建構於片群組層級。

【0106】若當前區塊是使用框內、CIIP 或框內區塊複製(IBC，亦已知為當前畫面參考或 CPR)模式進行編碼的，則 $avgY'$ 被計算成框內、CIIP、或 IBC 預測亮度值之平均值。另外， $avgY'$ 被計算成正向映射的框間預測亮度值(亦即，圖 5 之 Y'_{pred})之平均值。與基於樣本執行之亮度映射不同， $C_{ScaleInv}$ 是針對整個色度區塊之恆定值。透過使用 $C_{ScaleInv}$ ，於解碼器側處如下所示施加色度殘差縮放：

$$\text{解碼器側: } \hat{C}_{Res} = \hat{C}_{ResScale} * C_{ScaleInv}$$

【0107】其中， $\hat{C}_{ResScale}$ 是當前區塊之重建色度殘差。在編碼器側，如下執行正向色度殘差縮放(在變換與量化之前)：解碼器側： $C_{ResScale} = C_{Res} * C_{Scale} = C_{Res} / C_{ScaleInv}$ 。

【0108】在若干實施例中，可執行雙樹分割。

【0109】在 VVC draft 4 中，編碼樹方案支援亮度與色度具有分離區塊樹分割之能力。這亦被稱作雙樹分割。於圖 8 之表 3 與圖 9 之表 4 示出雙樹分割發訊。當「`qtbtt_dual_tree_intra_flag`」(其是 SPS 中發訊的序列層級控制旗標)被開啟時且當當前片群組是框內編碼時，可首先發訊針對亮度之區塊分割資訊，然後(彼此分離的)發訊針對色度之區

塊分割資訊。針對框間編碼片群組(P與B片群組)，不允許雙樹分割。當施加分離區塊樹模式時，亮度編碼樹區塊(CTB)被一編碼樹結構分割成CU，且色度CTB被另一編碼樹結構分割成色度CU，如以下圖9之表4所示。

【0110】當亮度與色度被允許具有不同種分割時，編碼工具可能會發生不同顏色分量間依賴性之問題。舉例而言，在LMCS之情況下，對應亮度區塊之平均值被用以確定要被施加到當前區塊之縮放因素。當使用雙樹時，這可能會造成整個CTU之潛時。舉例而言，若CTU之亮度區塊被垂直分割一次，且CTU之色度區塊被水平分割一次，則在CTU之第一色度區塊被解碼之前，CTU之兩個亮度區塊被解碼(以計算出平均值，其對於色度縮放因素之計算是必須的)。在VVC中，CTU可以大至 128×128 亮度樣本單位。這種大潛時對硬體解碼器管線設計有很大的問題。因此，VVC draft 4可禁止雙樹分割與亮度相依色度縮放之組合。當針對當前片群組賦能雙樹分割時，色度縮放被迫關閉(off)。應注意，在雙樹情況下LMCS之亮度映射部分仍是被允許的，因為其僅對亮度分量操作且不會有跨顏色分量依賴性之問題。編碼工具之另一實例倚賴顏色分量之間的依賴性以達到較佳編碼效率，其被稱作跨分量線性模型(CCLM)。

【0111】因此，片群組層級色度縮放因素LUT `cScaleInv[]`之導出並非易於擴展的。該導出處理目前取決於具64項目之恆定色度LUT `ChromaResidualScaleLut`。針

對具 16 塊之 10 位元視訊，必須施加除以 2 之附加步驟。當塊數量變化時，舉例而言使用 8 塊以取代 16 塊，則導出處理必須被改變以施加除以 4 之除法以取代除以 2 之除法。該附加步驟不僅造成精準性損失，亦為尷尬且不必要的。

【0112】另外，為了要計算出當前色度區塊之塊索引 Y_{Idx} ，其被使用以獲得色度縮放因素，可使用整個亮度區塊之平均值。這亦是不期望且很有可能是 unnecessary 的。考慮到最大 CTU 尺寸為 128×128 。在此情況下，基於 $16384(128 \times 128)$ 個亮度樣本計算出平均亮度值(此過程為繁複)。此外，若由編碼器選擇 128×128 之亮度區塊分割，區塊較可能包含同質內容。因此，區塊中亮度樣本之子集可能足夠以計算出亮度平均值。

【0113】在雙樹分割中，色度縮放可以是關閉以避免硬體解碼器之潛在管線問題。然而，若使用顯式發訊以指示要被施加色度縮放因素來取代使用對應亮度樣本以導出其，則可避免此依賴性。在框內編碼片群組中賦能色度縮放可進一步改善編碼效率。

【0114】分段線性參數之發訊可進一步改善。當前，針對 16 塊各者發訊 Δ 碼字值。可觀察到針對 16 塊通常只有使用有限數量之不同碼字。因此，可進一步降低發訊負擔。

【0115】本揭露之實施例提供一種用於透過移除色度縮放 LUT 來處理視訊內容之方法。

【0116】如前述，當使用其他分段線性模型時(例

如，8塊、4塊、及64塊等)，64項目之色度LUT之擴充可能是困難且富有問題的。因為為了達成相同的編碼效率，色度縮放因素可能被設定成與該對應塊之亮度縮放因素相同，因此這亦是不需要的。在本揭露若干實施例中，如下述可基於當前色度區塊之塊索引(Y_{idx})而判定色度縮放因素「chroma_scaling」。

- 若 $Y_{idx} > \text{resaper_model_max_bin_idx}$, $Y_{idx} < \text{resaper_model_min_bin_idx}$, 或若 $\text{SignaledCW}[Y_{idx}] = 0$, 則將 chroma_scaling 設定成預設值, $\text{chroma_scaling} = 1.0$ 。
- 否則, 將 chroma_scaling 設定成 $\text{SignaledCW}[Y_{idx}] / \text{OrgCW}$ 。

【0117】當 $\text{chroma_scaling} = 1.0$ 時，不施加縮放。

【0118】以上判定出色度縮放因素可能有分數精度。應瞭解，可施加定點近似以避免對硬體/軟體平台之依賴性。再者，於解碼器側，可執行反色度縮放。因此，可透過使用乘法後右移的定點算數來實現除法。可基於下述定點近似「CSCALE_FP_PREC」中位元數，判定出定點精準度中反色度縮放因素「inverse_chroma_scaling[]」。

$$\text{inverse_chroma_scaling}[Y_{idx}] = ((1 \ll (\text{luma_bit_depth} - \log_2(\text{TOTAL_NUMBER_PIECES}) + \text{CSCALE_FP_PREC})) + (\text{SignaledCW}[Y_{idx}] \gg 1)) / \text{SignaledCW}[Y_{idx}];$$

其中 luma_bit_depth 是亮度位元深度，TOTAL_NUMBER_PIECES 是分段線性模型中塊的總數量，其在 VVC draft 4 中被設定成 16。應瞭解，「inverse_chroma_scaling[]」之值可能僅需要對每片群組計算一次，且以上除法是整數除法運算。

【0119】可施加進一步量化以判定色度縮放與反縮放

因素。舉例而言，可針對「**SignaledCW**」之所有偶數($2 \times m$)值計算反色度縮放因素，且「**SignaledCW**」之所有奇數($2 \times m + 1$)值再使用相鄰偶數值之縮放因素的色度縮放因素。換言之，可使用以下者：

```
for(i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++)
{
    tempCW = SignaledCW[i] >> 1 << 1;

    inverse_chroma_scaling[i] = ((1 << (luma_bit_depth - log2(TOTAL_NUMBER_PIECES)
+ CSCALE_FP_PREC)) + (tempCW >> 1)) / tempCW;
}
```

【0120】色度縮放因素之量化可進一步被通用化。舉例而言，可針對「**SignaledCW**」的每第 n 個值計算反色度縮放因素「**inverse_chroma_scaling[]**」，所有其他相鄰值則共享該相同色度縮放因素。舉例而言，「 n 」可設定成4。因此，每四個相鄰碼字值可共享相同的反色度縮放因素值。在若干實施例中，「 n 」之值可以是2之冪，其允許使用移位來計算除法。將 $\log_2(n)$ 之值表示成 LOG2_n ，則以上方程式「 $\text{tempCW} = \text{SignaledCW}[i] \gg 1 \ll 1$ 」可被調整成：

$$\text{tempCW} = \text{SignaledCW}[i] \gg \text{LOG2_n} \ll \text{LOG2_n}。$$

【0121】在若干實施例中， LOG2_n 之值可以是分段線性模型中所用塊數量之函數。若使用較少的塊，則使用較大 LOG2_n 較有益。舉例而言，若 $\text{TOTAL_NUMBER_PIECES}$ 之值小於或等於16，則 LOG2_n 可以被設定成 $1 + (4 -$

$\log_2(\text{TOTAL_NUMBER_PIECES}))$ 。若 $\text{TOTAL_NUMBER_PIECES}$ 大於 16，則 LOG2_n 可以被設定成 0。

【0122】本揭露之實施例提供一種用於透過簡化亮度預測區塊之取平均步驟來處理視訊內容之方法。

【0123】如前述，為了要判定當前色度區塊之塊索引「 Y_{Idx} 」，可使用對應亮度區塊之平均值。然而，針對大的區塊尺寸，取平均處理可能涉及大量的亮度樣本數。最糟差情況為，取平均處理可能涉及 128×128 個亮度樣本。

【0124】本揭露之實施例提供一簡化取平均處理，以將最糟情況降低成僅使用 $N \times N$ 個亮度樣本 (N 為 2 之冪)。

【0125】在若干實施例中，若二維亮度區塊之兩個維度皆沒有小於或等於預設臨界值 M (換言之，兩維度之至少一者大於 M)，則施加「降取樣」以僅使用該維度中的 M 個位置。在沒有損失通用性前提下，採用水平維度作為實例。若寬度 (width) 大於 M ，則僅將位置 x 處之樣本使用於取平均中，其中 $x = i \times (\text{width} \gg \log_2(M))$, $i = 0, \dots, M-1$ 。

【0126】圖 10 描繪施加提議簡化以計算 16×8 個亮度區塊之平均值的實例。在本實例中，將 M 設定成 4，且僅使用該區塊中 16 個亮度樣本 (加陰影的樣本) 於取平均中。應瞭解預設臨界值 M 不受限於 4，且 M 可被設定成為 2 的冪之任意值。舉例而言，預設臨界值 M 可以是 1、2、4、及 8 等等。

【0127】在若干實施例中，亮度區塊之水平維度與垂直維度可具有不同預設臨界值 M 。換言之，取平均操作之

最糟情況為使用 $M1 \times M2$ 個樣本。

【0128】在若干實施例中，樣本數可受限於取平均處理中而不考慮維度。舉例而言，可使用 16 個樣本之最大值，其可採 1×16 、 16×1 、 2×8 、 8×2 、或 4×4 被分配於形式水平或垂直維度中，且適配當前區塊形狀之任何形式可被選定。舉例而言，若區塊狹長且為高，則可使用 2×8 樣本之矩陣；若區塊寬且為矮，則可使用 8×2 樣本之矩陣；以及若區塊為方形，則可使用 4×4 樣本之矩陣。

【0129】應瞭解到，當選定大區塊尺寸時，區塊內的內容趨向是較為同質。因此，雖然以上簡化可造成平均值與整個亮度區塊之真實平均值之間的差，此差可為小。

【0130】此外，在施加運動補償之前，解碼器側之運動向量精化 (DMVR) 需要解碼器執行運動搜尋以導出運動向量。因此，VVC 標準中 DMVR 模式可較為複雜，特別是針對解碼器之情況下。VVC 標準中雙向光學流動 (BDOF) 模式可進一步使情況複雜化，因為 BDOF 是額外的序列性處理，其需要再 DMVR 後被施加，以為了獲得亮度預測區塊。因為色度縮放需要對應亮度預測區塊之平均值，故 DMVR 與 BDOF 可在計算出平均值之前被施加。

【0131】為了解決潛時問題，在本揭露之若干實施例中，在 DMVR 與 BDOF 之前使用亮度預測區塊來計算出平均亮度值，且該平均亮度值被用以獲得色度縮放因素。這允許並行地對 DMVR 與 BDOF 處理施加色度縮放，且因此顯著降低潛時。

【0132】與本揭露一致，可考慮到潛時降低之變異。在若干實施例中，此潛時降低亦可與僅使用一部分亮度預測區塊來計算平均亮度值之前述簡化取平均處理結合。在若干實施例中，可在DMVR處理之後與BDOF處理之前使用亮度預測區塊來計算平均亮度值。接著，使用平均亮度值以獲得色度縮放因素。此種設計允許並行於BDOF處理施加色度縮放，同時維持判定色度縮放因素之準確度。由於DMVR處理可精化運動向量，因此在DMVR處理之後使用具精化運動向量之預測樣本會比在DMVR處理之前使用具運動向量之預測樣本更精準。

【0133】此外，在VVC標準中，CU語法結構「coding_unit()」包括語法元素「cu_cbf」，用來指示當前CU中是否有任何非零的殘差係數。在TU層級，TU語法結構「transform_unit()」包括語法元素「tu_cbf_cb」與「tu_cbf_cr」，用來指示當前TU中是否有任何非零的色度(Cb或Cr)殘差係數。習知技術中，在VVC draft 4中，若色度縮放經賦能於片群組層級，則總是呼叫對應亮度區塊之取平均處理。

【0134】本揭露之實施例進一步提供一種用於透過旁路亮度取平均處理來處理視訊內容之方法。與揭露實施例一致，由於色度縮放處理被施加到殘差色度係數，因此若沒有非零的色度係數則可旁路亮度取平均處理。這可基於以下條件予以判斷：

條件1：cu_cbf等於0

條件 2：tu_cbf_cr與tu_cbf_cb皆等於 0

【0135】如前述，「cu_cbf」可指示當前CU中是否有任何非零的殘差係數，以及「tu_cbf_cb」與「tu_cbf_cr」可指示當前TU中是否有任何非零的色度(Cb或Cr)殘差係數。當達成條件1或條件2中任一者時，可旁路亮度取平均處理。

【0136】在若干實施例中，只有使用預測區塊之 $N \times N$ 個樣本來導出平均值，這會簡化取平均處理。舉例而言，當 N 等於1時，只有使用預測區塊之左上樣本。然而，此使用預測區塊之簡化取平均處理仍需要產生預測區塊，藉此造成潛時。

【0137】在若干實施例中，可直接使用參考亮度樣本來產生色度縮放因素。這允許解碼器可以與亮度預測處理並行地導出縮放因素，從而降低潛時。下文將個別描述使用參考亮度樣本之框內預測與框間預測。

【0138】在例示性框內預測中，可將相同畫面中經解碼的相鄰樣本用作參考樣本，以產生預測區塊。此等參考樣本可包括例如當前區塊頂部之樣本，當前區塊左側之樣本，或當前區塊左上角處之樣本。可使用此等參考樣本之平均值以導出色度縮放因素。在若干實施例中，可使用此等參考樣本之部分的平均值。舉例而言，只有最靠近當前區塊左上角之 K 個參考樣本(例如， $K=3$)被取平均。

【0139】在例示性框間預測中，可使用來自時間參考畫面之參考樣本來產生預測區塊。此等參考樣本是透過參

考畫面索引與運動向量進行識別。若運動向量具有分數精度，可施加內插。用於判定參考樣本之平均值的參考樣本可包括內插之前與之後的參考樣本。內插之前的參考樣本可包括被裁到整數精度之運動向量。與本揭露實施例一致，所有此等參考樣本可用以計算該平均值。替代地，可以僅使用參考樣本(例如，對應於當前區塊左上角位置之參考樣本)之一部分來計算該平均值。

【0140】如圖5所示，可在整形域(reshaped domain)中執行框內預測(例如，框內預測514或528)，而在原始域中執行框間預測。因此，針對框間預測，可對預測區塊施加正向映射，且在正向映射之後的亮度預測區塊被用以計算該平均值。為降低潛時，使用正向映射之前的預測區塊計算該平均值。舉例而言，可使用正向映射之前的區塊、正向映射之後區塊之 $N \times N$ 部分、或正向映射之後區塊的左上樣本。

【0141】本揭露之實施例進一步提供一種用於對雙樹分割使用色度縮放處理視訊內容之方法。

【0142】因為對亮度之依賴性可能造成硬體設計複雜性，因此對賦能雙樹分割之框內編碼的片群組可關閉色度縮放。惟，這種限制可造成編碼效率損失。取代取對應亮度區塊之樣本數值的平均值來計算 $avgY'$ ，判定塊索引 Y_{idx} 以及獲得色度縮放因素 $inverse_chroma_scaling[Y_{idx}]$ ，可在位元流中顯式發訊色度縮放因素以避免在雙樹分割情況下對亮度之依賴性。

【0143】可在不同層級發訊色度縮放索引。舉例而言，可在編碼單元(CU)層級將色度縮放索引發訊，連帶發訊色度預測模式，如圖 11 表 5 所示。可使用語法元素「*lmcs_scaling_factor_idx*」來判斷對當前色度區塊之色度縮放因素。當不存在「*lmcs_scaling_factor_idx*」時，對當前色度區塊之色度縮放因素可被推得是等於 1.0(在浮點精度)或在定點精度中推得是相等($1 \ll \text{CSCALE_FP_PREC}$)。「*lmcs_chroma_scaling_idx*」之允許值的範圍是被判定在片群組層級，且因此將在文後討論。

【0144】取決於「*lmcs_chroma_scaling_idx*」之可能的值，發訊成本可能為高，對小區塊特別是如此。因此，在本揭露若干實施例中，圖 11 表 5 之發訊條件可額外地包括區塊尺寸條件。舉例而言，此語法元素「*lmcs_chroma_scaling_idx*」(以斜體字與畫陰影來強調)可僅在當前區塊包括多於給定數量之色度樣本時或在當前區塊具有比給定寬度 *W* 更大寬度或比給定高度 *H* 更大高度時被發訊。對較小區塊而言，若沒有發訊「*lmcs_chroma_scaling_idx*」，則可在解碼器側判定其色度縮放因素。在若干實施例中，色度縮放因素可以被設定成 1.0(於浮點精度中)。在若干實施例中，預設「*lmcs_chroma_scaling_idx*」值可添加於片群組標頭層級(參見圖 6 中表 1)。不具有發訊「*lmcs_chroma_scaling_idx*」之小區塊可使用此片群組層級之預設索引來導出對應之色度縮放因素。在若干實施例中，小區塊之色度縮放因素可自其具有顯式發訊縮放因素之相鄰

者(例如，上方或左側相鄰者)繼承。

【0145】除了在CU層級縮放此語法元素「`lmcs_chroma_scaling_idx`」以外，亦可在CTU層級將其發訊。然而，由於VVC中最大CTU尺寸是 128×128 ，因此在CTU層級進行相同縮放可能太粗糙。因此，在本揭露若干實施例中，可使用固定粒度將此語法元素「`lmcs_chroma_scaling_idx`」發訊。舉例而言，針對CTU中各 16×16 區域，將一個「`lmcs_chroma_scaling_idx`」發訊與施加到該 16×16 區域中所有的樣本。

【0146】對當前片群組之「`lmcs_chroma_scaling_idx`」的範圍取決於當前片群組中所允許的色度縮放因素值之數量。當前片群組中允許的色度縮放因素值之數量可以是基於前述64項目色度LUT來判定。替代地，當前片群組中允許的色度縮放因素值之數量可以是使用前述色度縮放因素計算來判定。

【0147】舉例而言，在「量化」方法中，`LOG2_n`之值可被設定成2(亦即，「`n`」被設定成4)，且於當前片群組之分段線性模型中各塊的碼字分配可被設定成下述者：
 $\{0, 65, 66, 64, 67, 62, 62, 64, 64, 64, 67, 64, 64, 62, 61, 0\}$ 。接著對該整個片群組只有2種可能的縮放因素值，因為從64到67之任何碼字值可具有相同的縮放因素值(分數精度中為1.0)，且從60到63之任何碼字值可具有相同縮放因素值(分數精度中為 $60/64=0.9375$)。對其沒有被分配到任何碼字之該兩個端塊(end piece)，色度縮放因素預設被

設定成 1.0。因此，在本實例中，一位元足以發訊對當前片群組中的區塊之「lmcs_chroma_scaling_idx」。

【0148】除了使用分段線性模型判定可能色度縮放因素數值之數量外，編碼器可在片群組標頭將一組色度縮放因素值發訊。接著在區塊層級，可透過使用該組色度縮放因素值以及針對區塊之「lmcs_chroma_scaling_idx」值判定對該區塊之色度縮放因素值。

【0149】CABAC編碼可被施加到碼「lmcs_chroma_scaling_idx」。區塊之CABAC上下文可取決於該區塊相鄰區塊之「lmcs_chroma_scaling_idx」。舉例而言，可使用左側區塊或上方區塊來形成CABAC上下文。在「lmcs_chroma_scaling_idx」之此語法元素的二元化方面而言，於VVC draft 4中對ref_idx_10與ref_idx_11語法元素所施加的相同截斷的Rice二元化可用以將「lmcs_chroma_scaling_idx」二元化。

【0150】將「chroma_scaling_idx」發訊的優點在於編碼器可根據率畸變成本來選擇出最佳「lmcs_chroma_scaling_idx」。使用率畸變最佳化來選擇「lmcs_chroma_scaling_idx」可改進編碼效率，其可幫助抵消發訊成本之增加。

【0151】本揭露之實施例進一步提供一種用於以LMCS分段線性模型之發訊來處理視訊內容之方法。

【0152】雖然LMCS方法使用有16塊之分段線性模型，一片群組中「SignaledCW[i]」之獨特值之數量趨向於

遠小於16。舉例而言，16塊中若干者可使用預設數量之碼字「OrgCW」，且16塊中若干者可具有彼此相同數量個碼字。因此，將LMCS分段線性模型發訊之替代方案可包括將一數量個獨特碼字「listUniqueCW[]」發訊，並傳送針對該各塊的索引以指示對當前塊之「listUniqueCW[]」的元素。

【0153】經調整後語法表被提供於圖12之表6。於圖12之表6中，新的或經調整後語法被以斜線與上陰影來強調。

【0154】所揭示發訊方法之語意為如下，其中對變化畫下底線：

reshaper_model_min_bin_idx指定整形器建構處理中要使用的最小筐(或塊)索引。**reshape_model_min_bin_idx**之值應在0到MaxBinIdx之範圍內，包括上下界。MaxBinIdx之值應等於15。

reshaper_model_delta_max_bin_idx指定最大允許筐(或塊)索引MaxBinIdx減去整形器建構處理中要使用的最大允許筐索引。**reshape_model_max_bin_idx**之值被設定成與MaxBinIdx-**reshape_model_delta_max_bin_idx**相等。

reshaper_model_bin_delta_abs_cw_prec_minus1加1指定用於表示語法**reshape_model_bin_delta_abs_CW[i]**之位元數量。

reshaper_model_bin_num_unique_cw_minus1加1指定碼字陣列listUniqueCW之大小。

reshaper_model_bin_delta_abs_CW[i] 指定對第 *i* 個筐之絕對 delta 碼字值。

reshaper_model_bin_delta_sign_CW_flag[i] 指定 **reshape_model_bin_delta_abs_CW[i]** 之正負號為如下：

- 若 **reshape_model_bin_delta_sign_CW_flag[i]** 等於 0，則對應變數 **RspDeltaCW[i]** 為正的值。
- 否則當 **reshape_model_bin_delta_sign_CW_flag[i]** 不等於 0 時，則對應變數 **RspDeltaCW[i]** 為負的值。

當 **reshape_model_bin_delta_sign_CW_flag[i]** 不存在時，其被推斷成等於 0。

變數 **RspDeltaCW[i]** 被導出成
$$\text{RspDeltaCW}[i] = (1 - 2 * \text{reshape_model_bin_delta_sign_CW}[i]) * \text{reshape_model_bin_delta_abs_CW}[i]$$

變數 **listUniqueCW[0]** 被設定成等於 **OrgCW**。變數 **listUniqueCW[i]** with $i = 1 \dots \text{reshaper_model_bin_num_unique_cw_minus1}$ ，包括上下界，導出如下：

- 變數 **OrgCW** 被設定成等於 $(1 << \text{BitDepth}_Y) / (\text{MaxBinIdx} + 1)$ 。
- $\text{listUniqueCW}[i] = \text{OrgCW} + \text{RspDeltaCW}[i - 1]$

reshaper_model_bin_cw_idx[i] 指定用於導出 **RspCW[i]** 的陣列 **listUniqueCW[]** 之索引。**reshaper_model_bin_cw_idx[i]** 之值應在 0 到 $(\text{reshaper_model_bin_num_unique_cw_minus1} + 1)$ 之範圍內，包括上下界。

RspCW[i] 被導出如下：

- 若 $\text{reshaper_model_min_bin_idx} \leq i \leq \text{reshaper_model_max_bin_idx}$
則 $\text{RspCW}[i] = \text{listUniqueCW}[\text{reshaper_model_bin_cw_idx}[i]]$ 。
- 否則 $\text{RspCW}[i] = 0$ 。

假若 BitDepth_Y 之值等於 10，則 $\text{RspCW}[i]$ 之值可以在 32 到 $2 * \text{OrgCW} - 1$ 之範圍內。

【0155】本揭露之實施例進一步提供一種用於在區塊層級利用條件式色度縮放來處理視訊內容之方法。

【0156】如圖 6 之表 1 所示，可透過在片群組層級發訊之「 $\text{tile_group_reshaper_chroma_residual_scale_flag}$ 」來判定是否施加色度縮放。

【0157】然而，在區塊層級判定是否施加色度縮放可以是有利的。舉例而言，在若干揭示實施例中，可發訊 CU 層級旗標以指示是否有對當前區塊施加色度縮放。CU 層級旗標之存在可取決於片群組層級旗標「 $\text{tile_group_reshaper_chroma_residual_scale_flag}$ 」。亦即，只有在色度縮放被允許於片群組層級時，可將 CU 層級旗標發訊。雖然編碼器是被允許可以基於色度縮放是否對當前區塊為有利來選擇是否使用色度縮放，其亦可能造成顯著發訊負擔。

【0158】與本揭露實施例一致，為了避免以上發訊負擔，對區塊施加色度縮放與否是取決於區塊之預測模式。舉例而言，當該區塊是經框間預測時，預測訊號趨向是良

好，特別是在其參考畫面靠近時間距離時。因此，由於預期殘差會極小，故可旁路色度縮放。舉例而言，在較高時間層級中畫面通常具有在時間距離相近之參考畫面。並且針對使用相鄰參考畫面之畫面中的區塊，色度縮放可以被去能。當前畫面與該區塊之參考畫面之間的畫面順序計數(POC)差異可用以判定此條件是否被達成。

【0159】在若干實施例中，可對所有的框間編碼區塊去能色度縮放。在若干實施例中，可對結合的框內/框間預測(CIIP)(其在VVC標準中被界定)模式去能色度縮放。

【0160】在VVC標準中，CU語法結構「coding_unit()」包括語法元素「cu_cbf」，用來指示當前CU中是否有任何非零的殘差係數。在TU層級，TU語法結構「transform_unit()」包括語法元素「tu_cbf_cb」與「tu_cbf_cr」，用來指示當前TU中是否有任何非零的色度(Cb或Cr)殘差係數。色度縮放處理可以取決於此等旗標。如前述，若沒有非零的殘差係數時，可呼叫對應亮度色度縮放處理之取平均。透過呼叫該取平均，色度縮放處理可以被旁路。

【0161】**圖13**描繪用於處理視訊內容的電腦實作方法1300之流程圖。在若干實施例中，可由編解碼器(例如，**圖2A-2B**中的編碼器或**圖3A-3B**中的解碼器)執行方法1300。舉例而言，編解碼器可被實作成用於編碼或變換編碼視訊序列的設備(如設備400)之一或多軟體或硬體組件。在若干實施例中，視訊序列可以是未壓縮視訊序列

(例如，視訊序列 202)或被解碼之壓縮視訊序列(例如，視訊串流 304)。在若干實施例中，視訊序列可以是由與設備處理器(例如，處理器 402)相關的監控裝置(例如，圖 4 中視訊輸入裝置)擷取的監控視訊序列。視訊序列可包括多畫面。設備可在畫面的層級執行方法 1300。舉例而言，在方法 1300 中，設備可一次處理一個畫面。針對另一種實例，在方法 1300 中，設備可一次處理複數個畫面。方法 1300 可包括以下步驟。

【0162】在步驟 1302，可接收到與畫面相關聯之色度區塊與亮度區塊。應瞭解，畫面可以與色度分量及亮度分量相關聯。因此，該畫面可以與包含色度樣本之色度區塊相關，且與包含亮度樣本之亮度區塊相關。

【0163】在步驟 1304，可判定與亮度區塊相關聯之亮度縮放資訊。在若干實施例中，亮度縮放資訊可以是畫面之資料串流中發訊的語法元素或是基於該畫面之資料串流中發訊的語法元素所導出的變數。舉例而言，亮度縮放資訊可包括以上方程式所述「`reshape_model_bin_delta_sign_CW[i]`及`reshape_model_bin_delta_abs_CW[i]`」及/或以上方程式所述「`SignaledCW[i]`」等等。在若干實施例中，亮度縮放資訊可包括基於該亮度區塊所判定之變數。舉例而言，可透過計算與該亮度區塊相鄰之亮度樣本(諸如在該亮度區塊頂部處一行中的亮度樣本，或在亮度區塊左側一行中的亮度樣本)的平均值來判定平均亮度值。

【0164】在步驟 1306，可基於亮度縮放資訊判定色度

縮放因素。

【0165】在若干實施例中，可基於亮度縮放資訊判定該亮度區塊之亮度縮放因素。舉例而言，根據以上方程式「 $\text{inverse_chroma_scaling}[i] = ((1 \ll (\text{luma_bit_depth} - \log_2(\text{TOTAL_NUMBER_PIECES}) + \text{CSCALE_FP_PREC})) + (\text{tempCW} \gg 1)) / \text{tempCW}$ 」，可基於亮度縮放資訊（例如，「tempCW」）判定亮度縮放因素。接著，可基於亮度縮放因素之值進一步判定色度縮放因素。舉例而言，色度縮放因素可以被設定成等於亮度縮放因素之值。應瞭解，在被設定成色度縮放因素之前，可對亮度縮放因素之值施加進一步計算。作另一實例，色度縮放因素可以被設定成等於「 $\text{SignaledCW}[Y_{Idx}] / \text{OrgCW}$ 」，其中可基於與亮度區塊相關聯的平均亮度值來判定當前色度區塊「 Y_{Idx} 」之塊索引。

【0166】在步驟1308，可使用色度縮放因素處理色度區塊。舉例而言，可使用色度縮放因素處理色度區塊之殘差，以產生該色度區塊之經縮放殘差。該色度區塊可以是Cb色度分量或Cr色度分量。

【0167】在若干實施例中，若滿足一條件則可處理該色度區塊。舉例而言，該條件可包括與沒有具有非零殘差之畫面相關聯的目標編碼單元；或包括與沒有具有非零色度殘差之畫面相關聯的目標變換單元。可基於該目標編碼單元之第一編碼區塊旗標之值來判定沒有具有非零殘差之目標編碼單元。且可基於針對該目標變換單元之第一分量的第二編碼區塊旗標之值與針對該目標變換單元之第二分

量的第三編碼區塊旗標之值，來判定沒有具有非零色度殘差之目標變換單元。舉例而言，第一分量可以是Cb分量，且第二分量可以是Cr分量。

【0168】應瞭解方法1300中之各步驟可做為獨立方法執行。舉例而言，步驟1308中所述用於判定色度縮放因素之方法可做為獨立方法執行。

【0169】圖14描繪用於處理視訊內容的電腦實作方法1400之流程圖。在若干實施例中，可由編解碼器(例如，圖2A-2B中的編碼器或圖3A-3B中的解碼器)執行方法1300。舉例而言，編解碼器可被實作成用於編碼或變換編碼視訊序列的設備(如設備400)之一或多軟體或硬體組件。在若干實施例中，視訊序列可以是未壓縮視訊序列(例如，視訊序列202)或被解碼之壓縮視訊序列(例如，視訊串流304)。在若干實施例中，視訊序列可以是由與設備處理器(例如，處理器402)相關的監控裝置(例如，圖4中視訊輸入裝置)擷取的監控視訊序列。視訊序列可包括多畫面。設備可在畫面的層級執行方法1400。舉例而言，在方法1400中，設備可一次處理一個畫面。針對另一種實例，在方法1400中，設備可一次處理複數個畫面。方法1400可包括以下步驟。

【0170】在步驟1402，可接收到與畫面相關聯之色度區塊與亮度區塊。應瞭解，畫面可以與色度分量及亮度分量相關聯。因此，該畫面可以與包含色度樣本之色度區塊相關，且與包含亮度樣本之亮度區塊相關。在若干實施例

中，亮度區塊可包括 $N \times M$ 個亮度樣本。 N 可以是亮度區塊之寬度且 M 可以是亮度區塊之高度。如前述，亮度區塊之亮度樣本可用以判定目標色度區塊之塊索引。因此，可接收到與一視訊序列之畫面相關聯的亮度區塊。應瞭解， N 與 M 可具有相同值。

【0171】在步驟 1404，為回應 N 與 M 之至少一者大於臨界值，因此可選定 $N \times M$ 個亮度樣本之子集。為了加速塊索引之判定，當達成特定條件時，亮度區塊可以被「降取樣」。換言之，亮度區塊中亮度樣本之子集可用以判定塊索引。在若干實施例中，該特定條件為 N 與 M 之至少一者大於臨界值。在若干實施例中，該臨界值可以基於 N 與 M 之至少一者。該臨界值可以是 2 之冪。舉例而言，臨界值可以是 4、8 及 16 等等。以 4 為實例，當 N 或 M 大於 4 時，可選定亮度樣本之子集。在圖 10 之實例中，亮度區塊之寬度與高度兩者皆大於為 4 之臨界值，且因此選擇 4×4 樣本之子集。應瞭解， 2×8 、 1×16 或等效物之子集亦可被選定以用於處理。

【0172】在步驟 1406，可判定 $N \times M$ 個亮度樣本之子集的平均值。

【0173】在若干實施例中，判定該平均值可進一步包括判定是否滿足第二條件；且回應於該第二條件已經被滿足之判定，判定 $N \times M$ 亮度樣本之子集的平均值。舉例而言，該第二條件可包括與沒有具有非零殘差係數之畫面相關聯的目標編碼單元；或包括沒有具有非零色度殘差係數

的目標編碼單元。

【0174】在步驟1408，可判定基於平均值之色度縮放因素。在若干實施例中，為了判定色度縮放因素，可基於該平均值判定色度區塊之塊索引，可判定色度區塊之塊索引是否滿足第一條件，以及接著回應於該色度區塊之塊索引滿足該第一條件之判定，色度縮放因素可以被設定成預設值。該預設值可指示沒有施加色度縮放。舉例而言，預設值可以是具分數精度之1.0。應瞭解，可對預設值施加定點近似。回應於該色度區塊之塊索引未滿足該第一條件之判定，可基於該平均值判定色度縮放因素。更明確的，色度縮放因素可以被設定成「 $\text{SignaledCW}[Y_{Idx}]/\text{OrgCW}$ 」，且可基於與對應亮度區塊相關聯的平均值來判定目標色度區塊「 Y_{Idx} 」之塊索引。

【0175】在若干實施例中，第一條件可包括比發訊碼字之最大索引還大的色度區塊之塊索引，或比發訊碼字之最小索引還小的色度區塊之塊索引。可判定發訊碼字之最大索引與最小索引為如下。

【0176】可透過使用分段線性模型(例如，LMCS)基於輸入訊號(例如，亮度樣本)來產生碼字。如前述，一動態範圍之輸入訊號可被分割成些許塊(例如，16塊)且輸入訊號各塊可用以產生碼字之筐做為輸出。因此，碼字之各筐可具有對應於該輸入訊號之塊的筐索引。在本實例中，筐索引之範圍可以是0-15。在若干實施例中，輸出(亦即碼字)之值是在最小值(例如，0)與最大值(例如，255)之

間，且具有在該最小值與該最大值之間的數值之複數個碼字可以被發訊。且可判定該經發訊的複數個碼字之筐索引。在該經發訊的複數個碼字之筐索引間，可進一步判定該發訊的複數個碼字之筐的最大筐索引與最小筐索引。

【0177】除了色度縮放因素以外，方法1400可進一步基於發訊的複數個碼字之筐來判定亮度縮放因素。可將度縮放因素用作反色度縮放因素。已經於上文中描述用於判定亮度縮放因素之方程式，且將在本文中省略對其之重複說明。在若干實施例中，複數個數量之經發訊的相鄰碼字共享該亮度縮放因素。舉例而言，二或四個經發訊的相鄰碼字可共享相同的亮度縮放因素，這可降低判定亮度縮放因素之負擔。

【0178】在步驟1410，可使用色度縮放因素處理色度區塊。如在前文參照圖5所述，複數個色度縮放因素可建構出在片群組層級之色度縮放因素LUT，並可在解碼器側被施加到目標區塊之重建色度殘差。相似的，亦可在編碼器側施加色度縮放因素。

【0179】應瞭解方法1400中之各步驟可做為獨立方法執行。舉例而言，步驟1308中所述用於判定色度縮放因素之方法可做為獨立方法執行。

【0180】圖15描繪用於處理視訊內容的電腦實作方法1500之流程圖。在若干實施例中，可由編解碼器(例如，圖2A-2B中的編碼器或圖3A-3B中的解碼器)執行方法1500。舉例而言，編解碼器可被實作成用於編碼或變換編

碼視訊序列的設備(如設備400)之一或多軟體或硬體組件。在若干實施例中，視訊序列可以是未壓縮視訊序列(例如，視訊序列202)或被解碼之壓縮視訊序列(例如，視訊串流304)。在若干實施例中，視訊序列可以是由與設備處理器(例如，處理器402)相關的監控裝置(例如，**圖4**中視訊輸入裝置)擷取的監控視訊序列。視訊序列可包括多畫面。設備可在畫面的層級執行方法1500。舉例而言，在方法1500中，設備可一次處理一個畫面。針對另一種實例，在方法1500中，設備可一次處理複數個畫面。方法1500可包括以下步驟。

【0181】 在步驟1502，可判定接收到視訊資料中是否存在色度縮放索引。

【0182】 在步驟1504，為了回應接收到視訊資料中沒有存在色度縮放索引之判定，可判定該接收到視訊資料沒有被施加色度縮放。

【0183】 在步驟1506，為了回應接收到視訊資料中有存在色度縮放索引之判定，可基於該色度縮放索引判定色度縮放因素。

【0184】 **圖16**描繪用於處理視訊內容的電腦實作方法1600之流程圖。在若干實施例中，可由編解碼器(例如，**圖2A-2B**中的編碼器或**圖3A-3B**中的解碼器)執行方法1600。舉例而言，編解碼器可被實作成用於編碼或變換編碼視訊序列的設備(如設備400)之一或多軟體或硬體組件。在若干實施例中，視訊序列可以是未壓縮視訊序列

(例如，視訊序列202)或被解碼之壓縮視訊序列(例如，視訊串流304)。在若干實施例中，視訊序列可以是由與設備處理器(例如，處理器402)相關的監控裝置(例如，圖4中視訊輸入裝置)擷取的監控視訊序列。視訊序列可包括多畫面。設備可在畫面的層級執行方法1600。舉例而言，在方法1600中，設備可一次處理一個畫面。針對另一種實例，在方法1600中，設備可一次處理複數個畫面。方法1600可包括以下步驟。

【0185】在步驟1602，可接收到針對一動態範圍之輸入視訊訊號的複數個獨特碼字。

【0186】在步驟1604，可接收到索引。

【0187】在步驟1606，可基於該索引選定該複數個獨特碼字中至少一者。

【0188】在步驟1608，可基於該選定至少一碼字判定色度縮放因素。

【0189】在若干實施例中，亦提供包括指令之非暫態電腦可讀儲存媒體，且該指令可由執行上述方法之裝置(諸如揭示編碼器與解碼器)執行。非暫態媒體之常見形態包括，例如，軟磁碟、軟碟、硬碟、固態硬碟機、磁帶或任何其他磁性資料儲存媒體、CD-ROM、任何其他光學資料儲存媒體，帶孔圖案的任何物理媒體、RAM、PROM、EPROM、FLASH-EPROM或任何其他快閃記憶體、NVRAM、快取記憶體、暫存器、任何其他記憶體晶片或磁盒式磁帶、及其網路版本等等。該裝置可包括一或多處

理器(CPU)、輸入/輸出介面、網路介面、及/或記憶體。

【0190】應瞭解上述實施例可由硬體、軟體(程式碼)、或硬體與軟體組合實作。若由軟體實作，則其可被儲存於上述電腦可讀媒體中。該軟體，當由處理器執行時，可執行所揭露之方法。於本揭露中描述之計算單元與其他功能性單元可被硬體、軟體、或硬體與軟體之組合實作。熟習該技藝者應能了解上述模組/單元之多者可被結合成一模組/單元，且上述模組/單元之各者可進一步被劃分成複數個子模組/子單元。

【0191】透過使用以下實例將進一步說明該等實施例：

- 1.一種用於處理視訊內容的電腦實作方法，其包含：
接收與畫面相關聯之色度區塊與亮度區塊；
判定與該亮度區塊相關聯之亮度縮放資訊；
基於該亮度縮放資訊判定色度縮放因素；以及
使用該色度縮放因素處理該色度區塊。
- 2.如實例1所述之方法，其中基於該亮度縮放資訊判定該色度縮放因素進一步包含：
基於該亮度縮放資訊判定該亮度區塊之亮度縮放因素；
基於該亮度縮放因素之值判定該色度縮放因素。
- 3.如實例2所述之方法，其中基於該亮度縮放因素之值判定該色度縮放因素進一步包含：
將該色度縮放因素設定成等於該亮度縮放因素之值。

4.如實例1-3中任一者所述之方法，其中使用該色度縮放因素處理該色度區塊進一步包含：

判定是否滿足第一條件；以及

執行下述一者：

回應於該第一條件被滿足之判定，使用該色度縮放因素處理該色度區塊；或

回應於該第一條件不被滿足之判定，旁路使用該色度縮放因素對該色度區塊進行之該處理。

5.如實例4所述之方法，其中該第一條件包含：

與沒有具有非零殘差之該畫面相關聯的目標編碼單元；或

與沒有具有非零色度殘差之該畫面相關聯的目標變換單元。

6.如實例5所述之方法，其中：

基於該目標編碼單元之第一編碼區塊旗標之值，判定沒有具有非零殘差之該目標編碼單元，以及

基於針對該目標變換單元之第一色度分量的第二編碼區塊旗標之值與針對該目標變換單元之第二色度分量的第三編碼區塊旗標之值，判定沒有具有非零色度殘差之該目標變換單元。

7.如實例6所述之方法，其中：

該第一編碼區塊旗標之值為0；以及

該第二編碼區塊旗標與該第三編碼區塊旗標之值為0。

8.如實例 1-7 中任一者所述之方法，其中使用該色度縮放因素處理該色度區塊包含：

使用該色度縮放因素處理該色度區塊之殘差。

9.一種用於處理視訊內容的設備，其包含：

記憶體，其儲存一組指令；以及

處理器，其耦接到該記憶體且經組態以執行該組指令以令該設備執行：

接收與畫面相關聯之色度區塊與亮度區塊；

判定與該亮度區塊相關聯之亮度縮放資訊；

基於該亮度縮放資訊判定色度縮放因素；以及

使用該色度縮放因素處理該色度區塊。

10.如實例 9 所述之設備，其中在基於該亮度縮放資訊判定該色度縮放因素步驟中，該處理器經組態以執行該組指令以令該設備進一步執行：

基於該亮度縮放資訊判定該亮度區塊之亮度縮放因素；

基於該亮度縮放因素之值判定該色度縮放因素。

11.如實例 10 所述之設備，其中在基於該亮度縮放因素之值判定該色度縮放因素步驟中，該處理器經組態以執行該組指令以令該設備進一步執行：

將該色度縮放因素設定成等於該亮度縮放因素之值。

12.如實例 9-11 中任一者所述之設備，其中在使用該色度縮放因素處理該色度區塊步驟中，該處理器經組態以執行該組指令以令該設備進一步執行：

判定是否滿足第一條件；以及

執行下述一者：

回應於該第一條件被滿足之判定，使用該色度縮放因素處理該色度區塊；或

回應於該第一條件不被滿足之判定，旁路使用該色度縮放因素對該色度區塊進行之該處理。

13.如實例12所述之設備，其中該第一條件包含：

與沒有具有非零殘差之該畫面相關聯的目標編碼單元；或

與沒有具有非零色度殘差之該畫面相關聯的目標變換單元。

14.如實例13所述之設備，其中：

基於該目標編碼單元之第一編碼區塊旗標之值，判定沒有具有非零殘差之該目標編碼單元，以及

基於針對該目標變換單元之第一色度分量的第二編碼區塊旗標之值與針對該目標變換單元之第二色度分量的第三編碼區塊旗標之值，判定沒有具有非零色度殘差之該目標變換單元。

15.如實例14所述之設備，其中：

該第一編碼區塊旗標之值為0；以及

該第二編碼區塊旗標與該第三編碼區塊旗標之值為0。

16.如實例9-15中任一者所述之設備，其中在使用該色度縮放因素處理該色度區塊步驟中，該處理器經組態以執

行該組指令以令該設備進一步執行：

使用該色度縮放因素處理該色度區塊之殘差。

17.一種非暫態電腦可讀儲存媒體，其儲存一組可由裝置之一或多處理器執行之指令，以令該裝置執行用於處理視訊內容之方法，該方法包含：

接收與畫面相關聯之色度區塊與亮度區塊；

判定與該亮度區塊相關聯之亮度縮放資訊；

基於該亮度縮放資訊判定色度縮放因素；以及

使用該色度縮放因素處理該色度區塊。

18.一種用於處理視訊內容的電腦實作方法，該方法包含：

接收與畫面相關聯之色度區塊與亮度區塊，該亮度區塊包含 $N \times M$ 個亮度樣本；

回應於該 N 與 M 之至少一者大於臨界值，選定該 $N \times M$ 個亮度樣本之子集；

判定該 $N \times M$ 個亮度樣本之子集的平均值；

基於該平均值判定色度縮放因素；以及

使用該色度縮放因素處理該色度區塊。

19.一種用於處理視訊內容的電腦實作方法，該方法包含：

判定接收到視訊資料中是否存在色度縮放索引；

回應於該接收到視訊資料中沒有存在該色度縮放索引之判定，判定該接收到視訊資料沒有被施加色度縮放；以及

回應於該接收到視訊資料中有存在該色度縮放索引之判定，基於該色度縮放索引判定色度縮放因素。

20.一種用於處理視訊內容的電腦實作方法，該方法包含：

接收針對一動態範圍之輸入視訊訊號的複數個獨特碼字；

接收索引；

基於該索引選定該複數個獨特碼字中至少一者；以及

基於該選定至少一碼字判定色度縮放因素。

【0192】除了透過使用電腦可讀程式碼來實現上述方法之外，還可以以邏輯閘、切換器、ASIC、可程式化邏輯控制器和嵌入式微控制器的形式來實現上述方法。因此，此種控制器可以被視為硬體組件，且包括於控制器中且被組態以實作各種功能的設備亦可被視為是該硬體組件中的結構。或者，被組態以實現各種功能的設備甚至可以被認為是被配置為實現硬體組件內部的方法和結構的軟體模組等兩者。

【0193】可以在由電腦執行的電腦可執行指令(例如程式模組)的一般情境下描述本揭露。一般而言，程式模組包括用於執行特定工作或實作特定抽象資料類型之常式、程式、物件、總成、資料結構、類、或等等。本揭露之實施例亦可被實作於分散式計算環境中。在分散式計算環境中，透過使用透過通訊網路連接之遠端處理裝置來執行工作。在分散式計算環境中，程式模組可位於本地與遠

端電腦儲存媒體中，包括儲存裝置。

【0194】應瞭解，本文中的相關術語諸如「第一」與「第二」僅用以將一實體或操作與另一實體或操作彼此差異化，且並不需要或暗示此等實體或操作之間的任何實際關聯性或順序。此外，詞語「包含」，「具有」，「包含」和「包括」以及其他類似形式在含義上是等效的，並且是開放式的，因此這些詞語中的任何一者之後所接的一或多個項目是並不意味著是此等項目之窮舉列表，也不意味著僅限於所列的項目。

【0195】在前述說明書中，已經參照可以隨實現方式而變化的許多具體細節描述了實施例。可以對所描述的實施例進行某些修改和調整。考慮到本文揭露的說明書和實施方式，其他實施例對於熟習該技術者而言是顯而易見的。因此，說明書與實例僅被視為是例示性的，且本揭露之實際範疇與精神將由以下申請專利範圍所示。還希望所附圖式中所示的步驟順序僅出於說明目的，並且不旨在限於任何特定的步驟順序。因此，熟習該技藝者將可體認到可以不同順序執行此等步驟，同時還算是在實作同個方法。

【符號說明】

【0196】

100:視訊序列

102:畫面

104:畫面
 106:畫面
 108:畫面
 110:結構
 112:基本處理單元
 114:區
 116:區
 118:區
 200A:編碼處理
 200B:編碼處理
 202:視訊序列
 204:預測階段
 2042:空域預測階段
 2044:時域預測階段
 206:預測資料
 208:預測BPU
 210:殘差BPU
 212:變換階段
 214:量化階段
 216:量化變換係數
 218:反量化階段
 220:反變換階段
 222:重建殘差BPU
 224:預測參考

226:二進制編碼階段

228:視訊位元流

230:模式決定階段

232:迴路過濾階段

234:緩衝器

300A:解碼處理

300B:解碼處理

302:二進制解碼階段

304:視訊串流

400:設備

402:處理器

402a~402n:處理器

404:記憶體

406:網路介面

408:週邊介面

410:匯流排

500:具色度縮放的亮度映射處理

502:視訊位元流

504: Q^{-1} & T^{-1} 階段

506:重建階段

508:反映射階段

510:迴路過濾階段

512:解碼畫面緩衝階段

514:框內預測階段

516:運動補償階段
 518:正向映射階段
 520:色度縮放階段
 522:重建階段
 524:迴路過濾階段
 526:解碼畫面緩衝階段
 528:框內預測階段
 530:運動補償階段
 1300:方法
 1302:步驟
 1304:步驟
 1306:步驟
 1308:步驟
 1400:方法
 1402:步驟
 1404:步驟
 1406:步驟
 1408:步驟
 1410:步驟
 1500:方法
 1502:步驟
 1504:步驟
 1506:步驟
 1600:方法

1602:步驟

1604:步驟

1606:步驟

1608:步驟

【發明申請專利範圍】

【請求項1】一種用於處理視訊內容的電腦實現的方法，其包含：

接收與畫面相關聯的色度區塊，所述色度區塊包括多個亮度樣本；

判定所述畫面中的多個亮度樣本的平均值；以及

基於所述平均值來判定色度縮放因素，

其中，所述色度縮放因素用於處理所述色度區塊的殘差，

其中，處理所述色度區塊的殘差包括：

判定所述色度區塊是否具有非零殘差；以及

回應於判定所述色度區塊具有非零殘差，將所述色度縮放因素應用於所述色度區塊的所述殘差。

【請求項2】如請求項1所述之方法，所述色度區塊是否具有非零殘差是基於與所述色度區塊相關聯的編碼區塊旗標之值來判定。

【請求項3】如請求項1所述之方法，所述方法更包括：

回應於與所述色度區塊相關聯的編碼區塊旗標之值為1，判定所述色度區塊具有非零殘差。

【請求項4】如請求項1所述之方法，其中，判定所述多個亮度樣本的所述平均值是透過判定滿足第一條件來觸發。

【請求項5】一種用於處理視訊內容的設備，所述設

備包括：

記憶體，其儲存有指令集；以及

處理器，所述處理器連接至所述記憶體並被配置為執行所述指令集，以使所述設備執行以下操作：

接收與畫面相關聯的色度區塊，所述色度區塊包括多個亮度樣本；

判定所述畫面中的多個亮度樣本的平均值；

基於所述平均值來判定色度縮放因素；以及

透過將所述色度縮放因素應用於所述色度區塊的殘差來處理所述色度區塊的所述殘差，

其中，所述色度區塊的所述殘差之所述處理包括：

判定所述色度區塊是否具有非零殘差；以及

回應於判定所述色度區塊具有非零殘差，將所述色度縮放因素應用於所述色度區塊的所述殘差。

【請求項6】如請求項5所述之設備，其中，

所述色度區塊是否具有非零殘差是基於與所述色度區塊相關聯的編碼區塊旗標之值判定。

【請求項7】如請求項5所述之設備，其中，所述處理器經組態以執行所述指令集，以使所述設備進一步執行以下操作：

回應於與所述色度區塊相關聯的編碼區塊旗標之值為1，判定所述色度區塊具有非零殘差。

【請求項8】一種非暫態電腦可讀儲存媒體，所述非暫態電腦可讀儲存媒體儲存有指令集，所述指令集能由設

備的一或更多處理器執行，以使所述設備執行用於處理視訊內容的方法，所述方法包括以下步驟：

接收與畫面相關聯的色度區塊，所述色度區塊包括多個亮度樣本；

判定所述畫面中的多個亮度樣本的平均值；

基於所述平均值來判定色度縮放因素；以及

透過將所述色度縮放因素應用於所述色度區塊的殘差來處理所述色度區塊的所述殘差，

其中，所述色度區塊的所述殘差之所述處理包括：

判定所述色度區塊是否具有非零殘差；以及

回應於判定所述色度區塊具有非零殘差，將所述色度縮放因素應用於所述色度區塊的所述殘差。

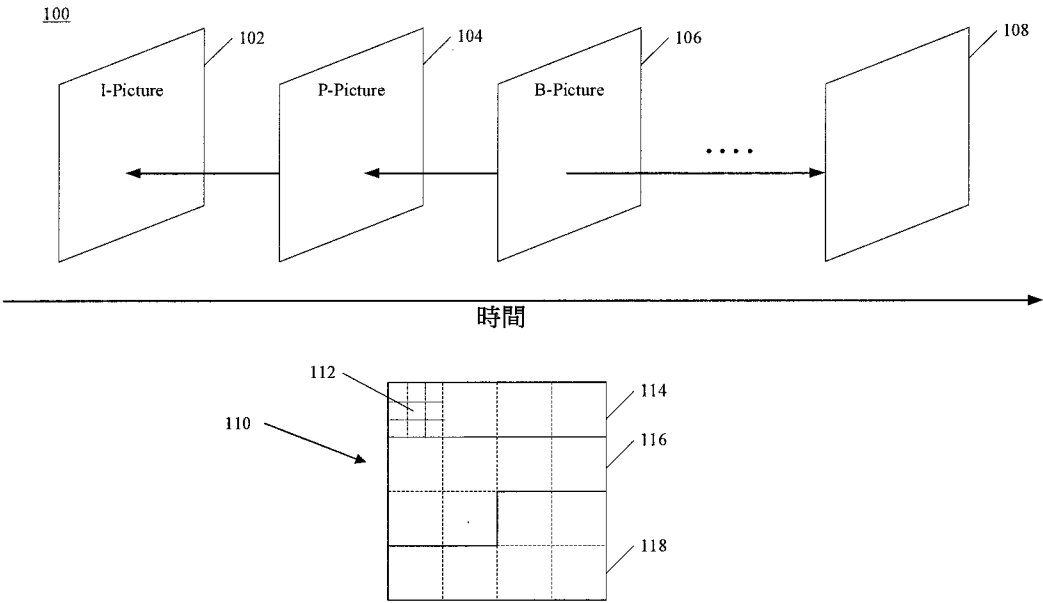
【請求項9】如請求項8所述之媒體，其中，

所述色度區塊是否具有非零殘差是基於與所述色度區塊相關聯的編碼區塊旗標之值判定。

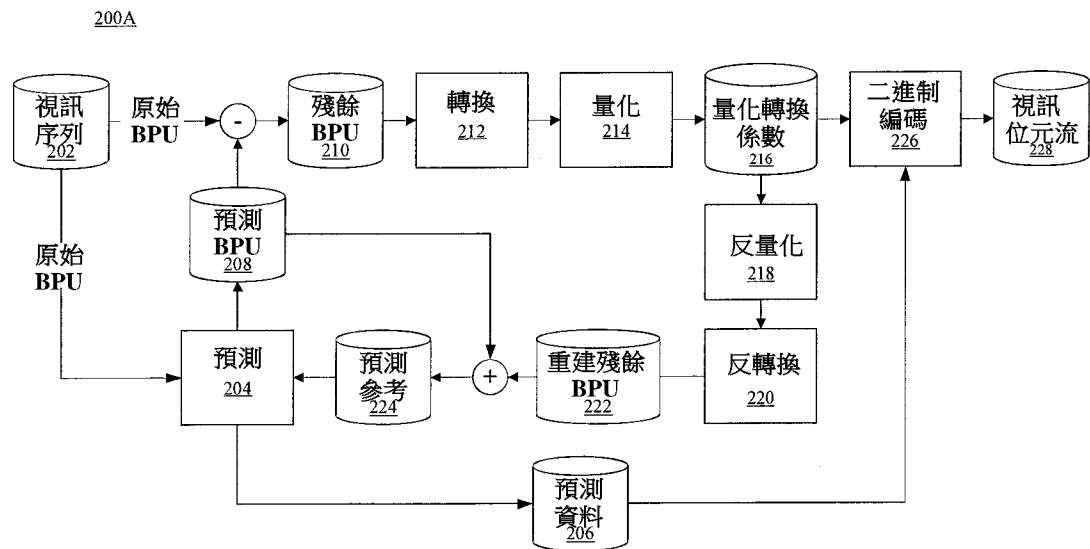
【請求項10】如請求項8所述之媒體，其中，所述方法進一步包括以下步驟：

回應於與所述色度區塊相關聯的編碼區塊旗標之值為1，判定所述色度區塊具有非零殘差。

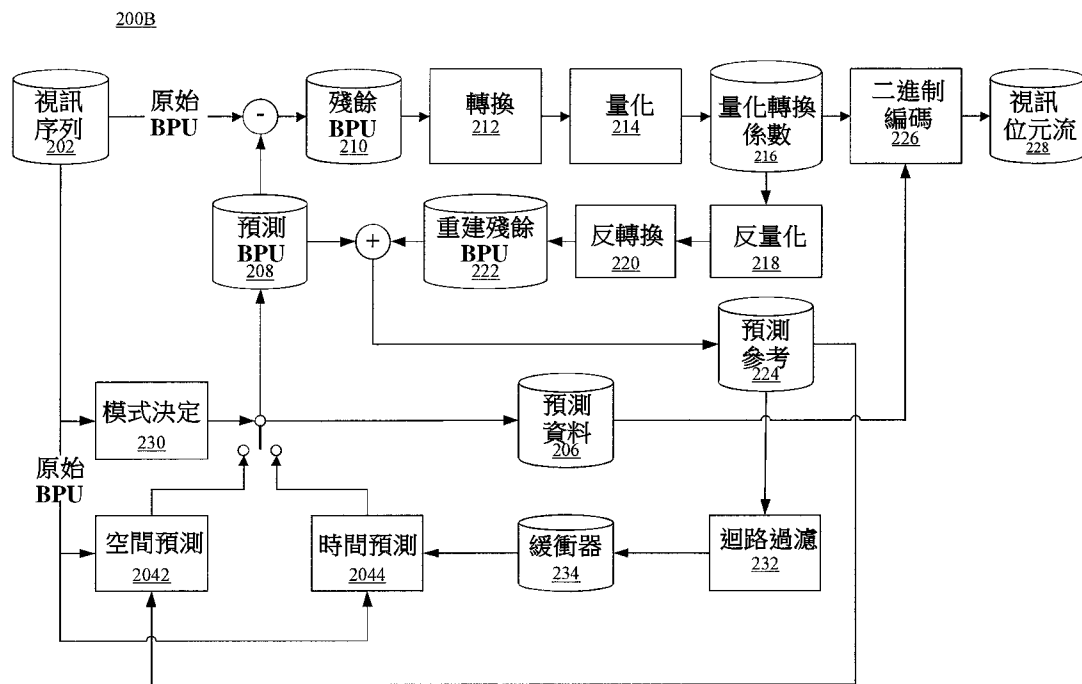
【發明圖式】



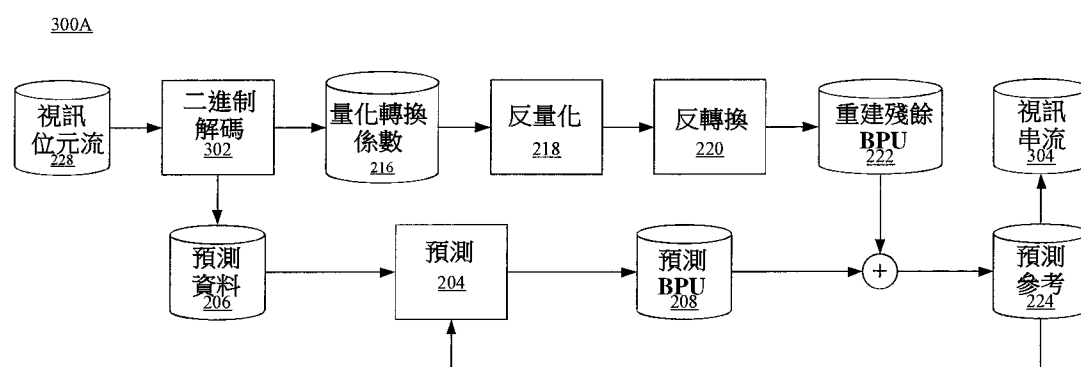
【圖 1】



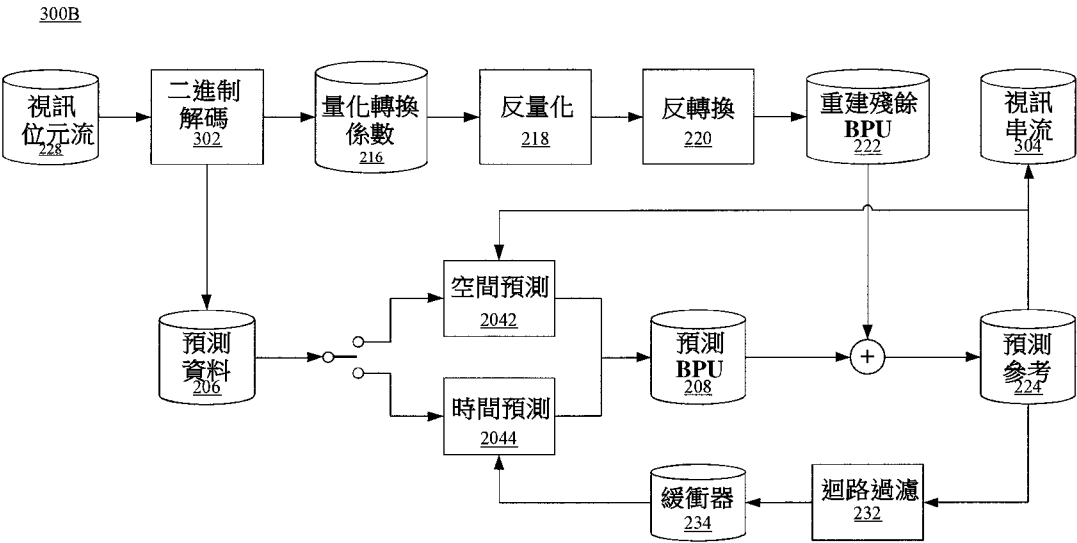
【圖 2A】



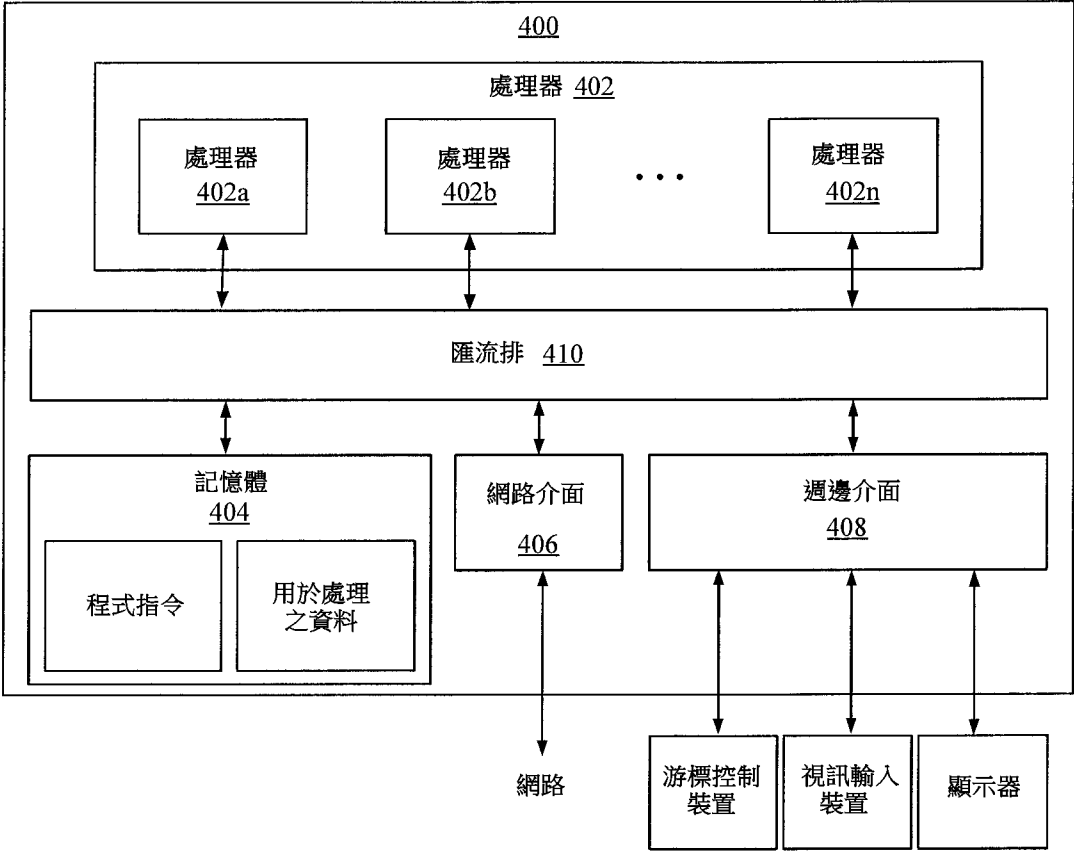
【圖 2B】



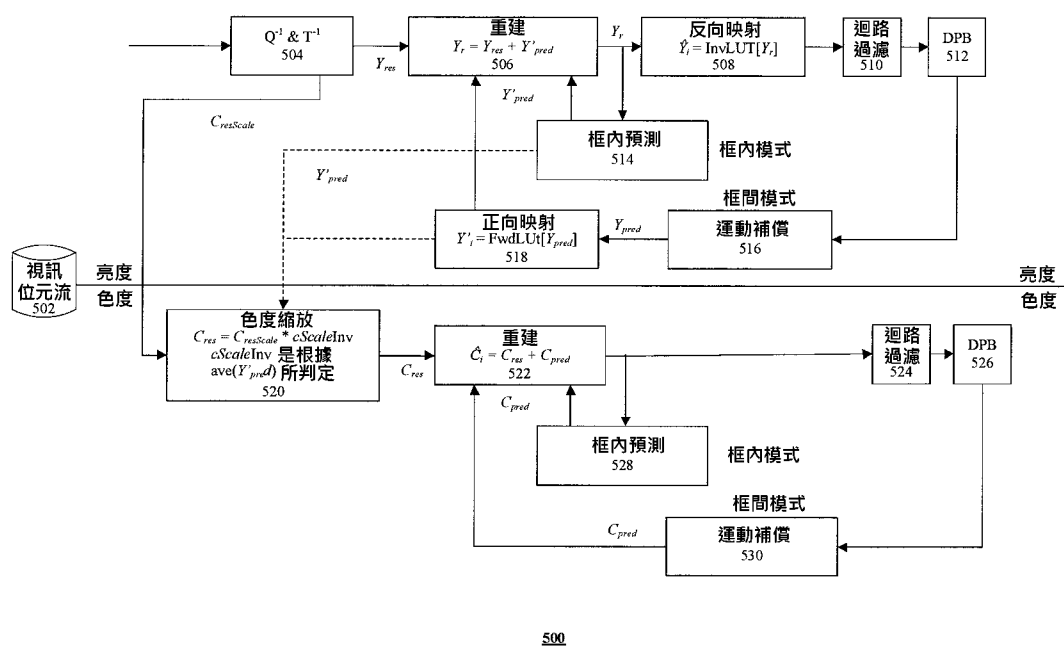
【圖 3A】



【圖 3B】



【圖 4】



【圖 5】

表 1

tile_group_header() {	描述符
...	
if (sps_reshaper_enabled_flag) {	
tile_group_reshaper_model_present_flag	u(1)
if (tile_group_reshaper_model_present_flag)	
tile_group_reshaper_model ()	
tile_group_reshaper_enable_flag	u(1)
if (tile_group_reshaper_enable_flag && !(qtbt_dual_tree_intra_flag && tile_group_type == I)))	
tile_group_reshaper_chroma_residual_scale_flag	u(1)
}	
byte_alignment()	
}	

【圖 6】

表 2

tile_group_resaper_model () {	描述符
reshaper_model_min_bin_idx	ue(v)
reshaper_model_delta_max_bin_idx	ue(v)
reshaper_model_bin_delta_abs_cw_prec_minus1	ue(v)
for (i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++) {	
reshape_model_bin_delta_abs_CW [i]	u(v)
if (reshaper_model_bin_delta_abs_CW[i] > 0)	
reshaper_model_bin_delta_sign_CW_flag[i]	u(1)
}	
}	

【圖 7】

表 3

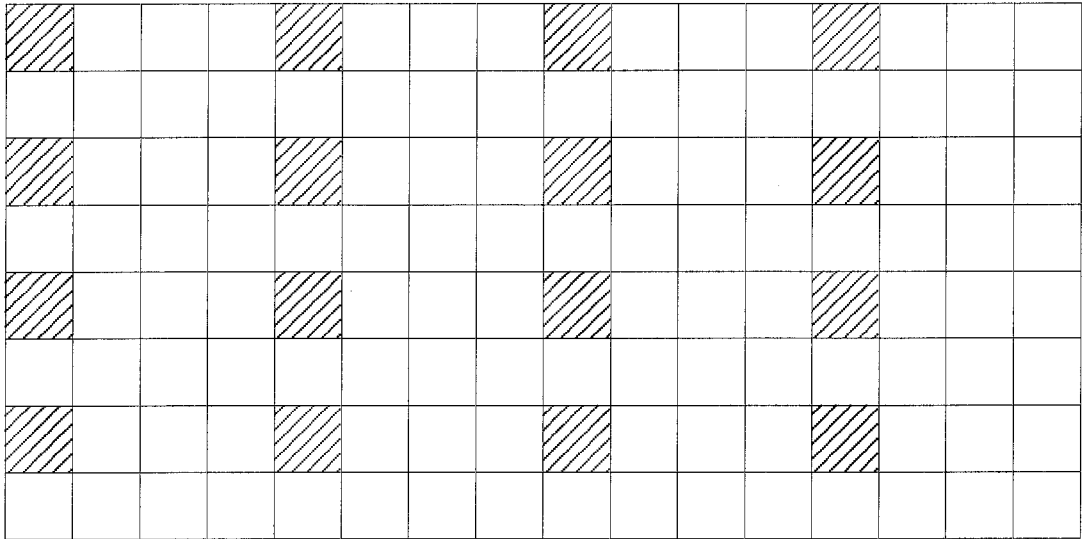
coding_tree_unit() {	描述符
xCtb = (CtbAddrInRs % PicWidthInCtbsY) << CtbLog2SizeY	
yCtb = (CtbAddrInRs / PicWidthInCtbsY) << CtbLog2SizeY	
if(tile_group_sao_luma_flag tile_group_sao_chroma_flag)	
sao(xCtb >> CtbLog2SizeY, yCtb >> CtbLog2SizeY)	
if(tile_group_alf_enabled_flag){	
alf_ctb_flag [0][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	ae(v)
if(alf_chroma_idc == 1 alf_chroma_idc == 3)	
alf_ctb_flag [1][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	ae(v)
if(alf_chroma_idc == 2 alf_chroma_idc == 3)	
alf_ctb_flag [2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	ae(v)
}	
if((tile_group_type == I CurrPicIsOnlyRef) && qtbtt_dual_tree_intra_flag)	
dual_tree_implicit_qt_split (xCtb, yCtb, CtbLog2SizeY, 0)	
else	
coding_quadtree(xCtb, yCtb, CtbLog2SizeY, 0, SINGLE_TREE)	
}	

【圖 8】

表 4

dual_tree_implicit_qt_split(x0, y0, log2CbSize, cqtDepth) {	描述符
if(log2CbSize > 6) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
dual_tree_implicit_qt_split(x0, y0, log2CbSize - 1, cqtDepth + 1)	
if(x1 < pic_width_in_luma_samples)	
dual_tree_implicit_qt_split(x1, y0, log2CbSize - 1, cqtDepth + 1)	
if(y1 < pic_height_in_luma_samples)	
dual_tree_implicit_qt_split(x0, y1, log2CbSize - 1, cqtDepth + 1)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
dual_tree_implicit_qt_split(x1, y1, log2CbSize - 1, cqtDepth + 1)	
} else {	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_LUMA)	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_CHROMA)	
}	
}	

【圖 9】



【圖 10】

表 5

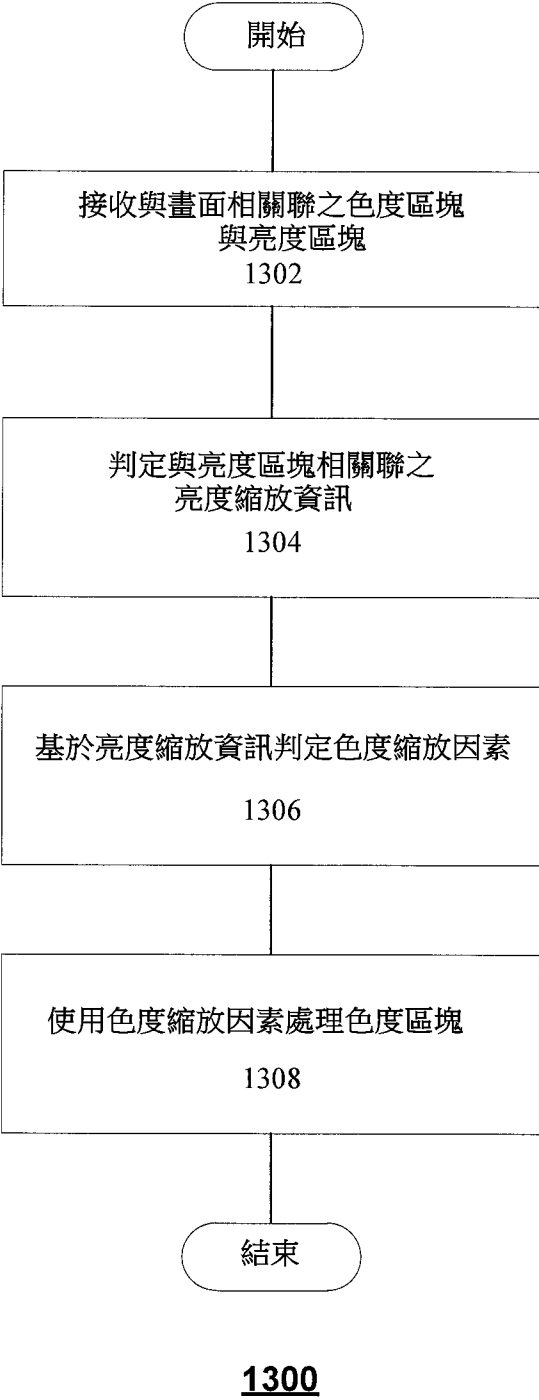
<code>coding_unit(x0, y0, cbWidth, cbHeight, treeType) {</code>	描述符
<code>...</code>	
<code>if(CuPredMode[x0][y0] == MODE_INTRA) {</code>	
<code>....</code>	
<code>if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)</code>	
<code>{</code>	
<code>intra_chroma_pred_mode[x0][y0]</code>	<code>ae(v)</code>
<code>if(tile_group_reshaper_chroma_residual_scale_flag)</code>	
<code>lms_chroma_scaling_idx[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>}</code>	
<code>....</code>	

【圖 11】

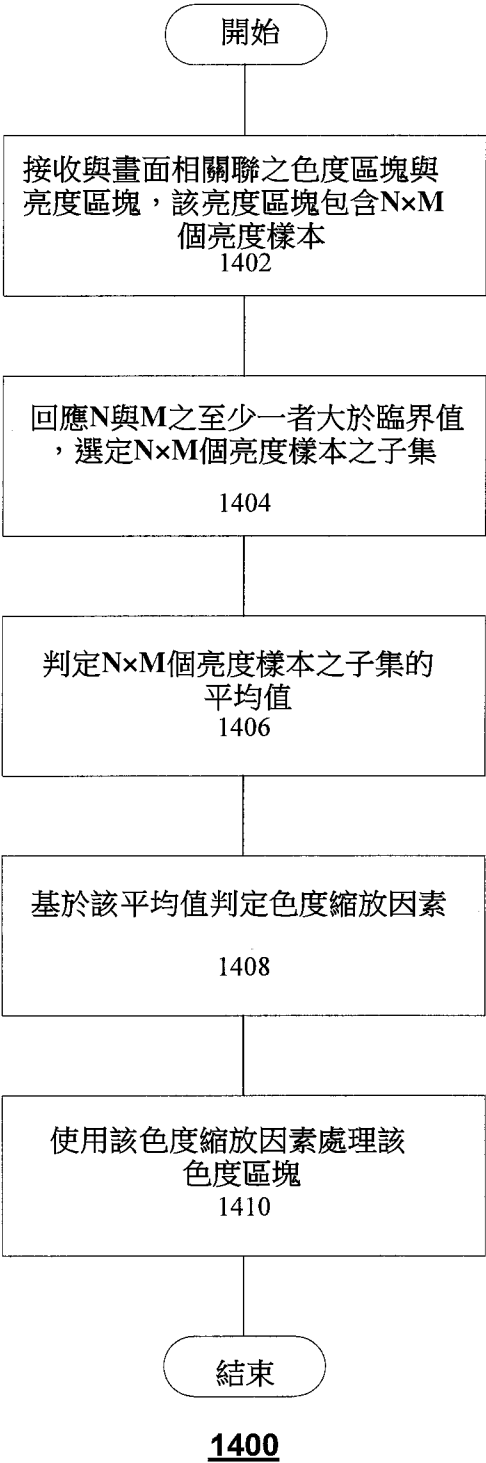
表 6

tile_group_reshaper_model () {	描述符
reshaper_model_min_bin_idx	ue(v)
reshaper_model_delta_max_bin_idx	ue(v)
reshaper_model_bin_delta_abs_cw_prec_minus1	ue(v)
reshaper_model_bin_num_unique_cw_minus1	ue(v)
for (i = 0; i <= reshaper_model_bin_num_unique_cw_minus1; i++) {	
reshape_model_bin_delta_abs_CW [i]	u(v)
if (reshaper_model_bin_delta_abs_CW[i] > 0)	
reshaper_model_bin_delta_sign_CW_flag[i]	u(1)
}	
for (i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++)	
reshaper_model_bin_cw_idx [i]	u(v)
}	

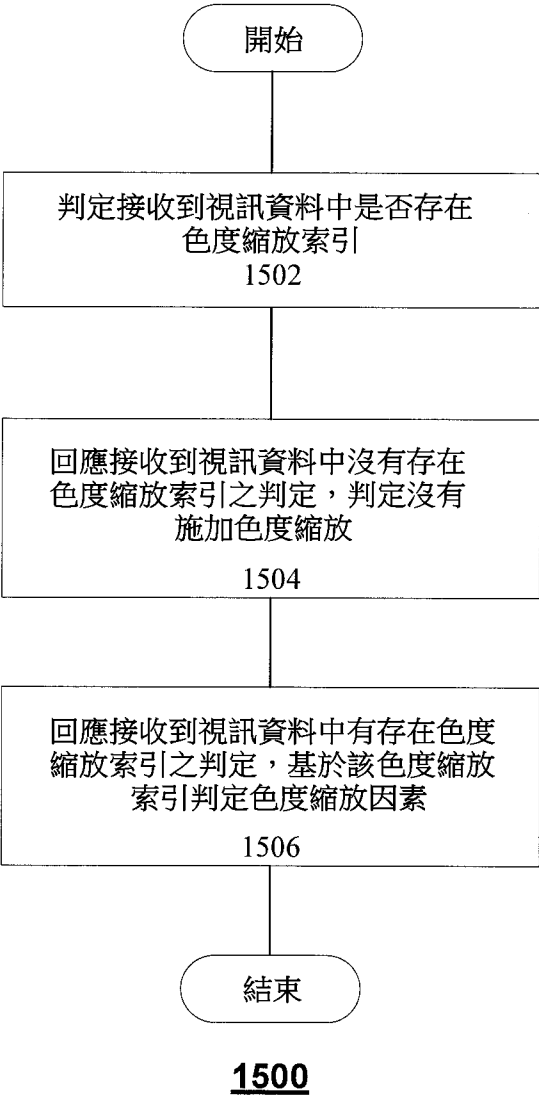
【圖 12】



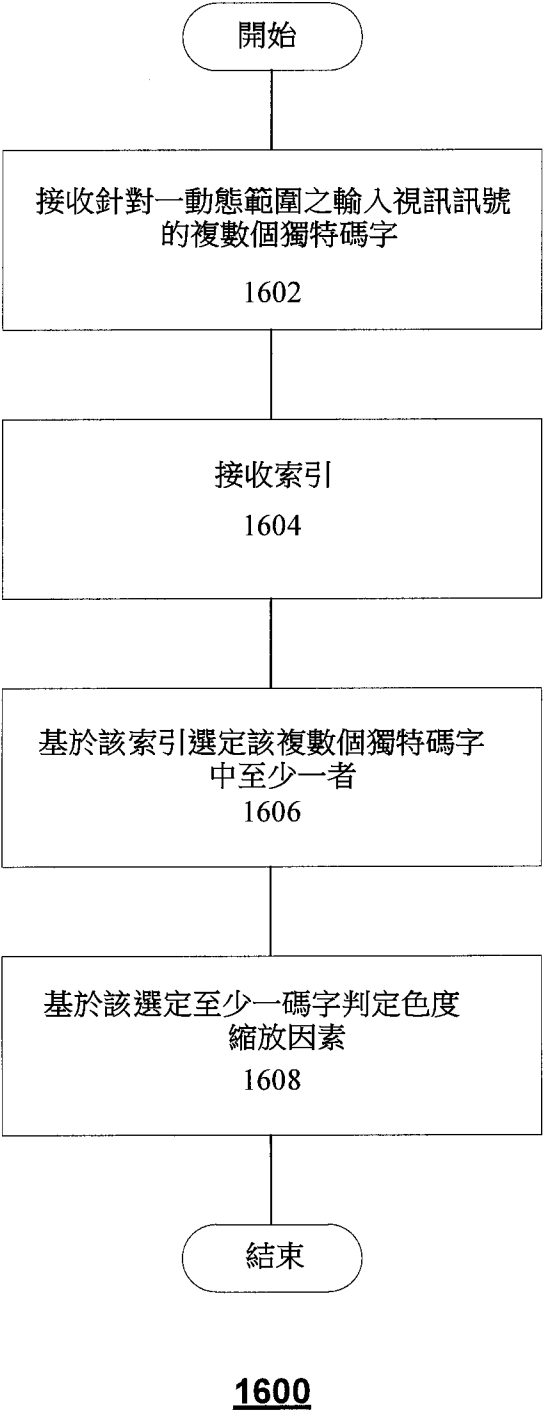
【圖 13】



【圖 14】



【圖 15】



【圖 16】