



(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 I533147 B

(45)公告日：中華民國 105 (2016) 年 05 月 11 日

(21)申請案號：101107704

(22)申請日：中華民國 94 (2005) 年 09 月 02 日

(51)Int. Cl. : G06F17/00 (2006.01)

(30)優先權：2005/04/22 美國

11/111,983

(71)申請人：微軟技術授權有限責任公司 (美國) MICROSOFT TECHNOLOGY LICENSING, LLC
(US)

美國

(72)發明人：薛奇特葛列格 SCHECHTER, GREG (US)；沙克傑文 SAKS, JEVAN (US)；佛提爾契瑞斯 FORTIER, CHRIS (US)

(74)代理人：蔡坤財；李世章

(56)參考文獻：

TW 364089

US 2001/0028368A1

審查人員：彭智輝

申請專利範圍項數：20 項 圖式數：6 共 59 頁

(54)名稱

用於在一視窗管理上運用各現場視窗之縮圖的介面及系統

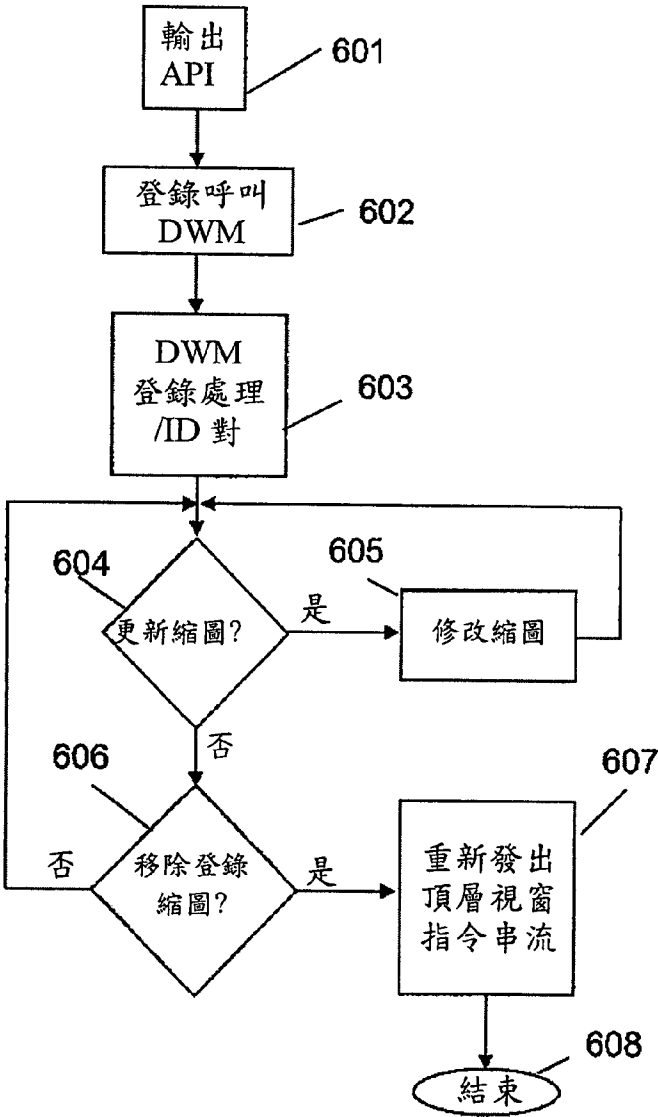
INTERFACE AND SYSTEM FOR MANIPULATING THUMBNAILS OF LIVE WINDOWS IN A WINDOW MANAGE

(57)摘要

本發明揭示一種用於在一顯示器上顯示一視窗的介面、系統或方法，其中一對應至一應用程式視窗的縮圖可根據該應用程式視窗內的修改來修改。該縮圖的修改也可即時達成。另外，本發明提供一種用於在一顯示器上顯示一視窗或一縮圖的介面、系統及方法，其中一桌面視窗管理器(DWM)登錄一來源視窗以及一目的地視窗，並且/或更新或修改該來源視窗伴隨之一縮圖。特效可套用至該縮圖。本發明另提供一種用於登錄並且/或修改一縮圖及其特性，或維護一縮圖登錄清單的方法。

The present invention provides an interface, system, or method for displaying a window on a display in which a thumbnail corresponding to an application window is capable of being modified based on modifications in the application window. The modifications of the thumbnail may be accomplished in real-time. Also, the present invention provides an interface, system, and method for displaying a window or a thumbnail on a display in which a Desktop Window Manager (DWM) registers a source window and a destination window and/or may update or modify a thumbnail associated with the source window. Special effects may be applied to the thumbnail. The present invention further provides a method for registering and/or modifying a thumbnail and its properties or maintaining a list of thumbnail registrations.

指定代表圖：



第 6 圖

發明專利說明書

(本說明書格式、順序，請勿任意更動，※記號部分請勿填寫；惟已有申請案號者請填寫)

※ 申請案號：101107704

※ 申請日期：94年9月2日

※ 原申請案號：94130191

※IPC 分類：

G06F17/60 (2006.01)

一、發明名稱：(中文/英文)

用於在一視窗管理上運用各現場視窗之縮圖的介面及系統

INTERFACE AND SYSTEM FOR MANIPULATING THUMBNAILS
OF LIVE WINDOWS IN A WINDOW MANAGE

二、中文發明摘要：

本發明揭示一種用於在一顯示器上顯示一視窗的介面、系統或方法，其中一對應至一應用程式視窗的縮圖可根據該應用程式視窗內的修改來修改。該縮圖的修改也可即時達成。另外，本發明提供一種用於在一顯示器上顯示一視窗或一縮圖的介面、系統及方法，其中一桌面視窗管理器(DWM)登錄一來源視窗以及一目的地視窗，並且/或更新或修改該來源視窗伴隨之一縮圖。特效可套用至該縮圖。本發明另提供一種用於登錄並且/或修改一縮圖及其特性，或維護一縮圖登錄清單的方法。

三、英文發明摘要：

The present invention provides an interface, system, or method for displaying a window on a display in which a thumbnail corresponding to an application window is capable of being modified based on modifications in the application window. The modifications of the thumbnail may be accomplished in real-time. Also, the present invention provides an interface, system, and method for

displaying a window or a thumbnail on a display in which a Desktop Window Manager (DWM) registers a source window and a destination window and/or may update or modify a thumbnail associated with the source window. Special effects may be applied to the thumbnail. The present invention further provides a method for registering and/or modifying a thumbnail and its properties or maintaining a list of thumbnail registrations.

四、指定代表圖：

(一)本案指定代表圖為：第（ 6 ）圖。

(二)本代表圖之元件符號簡單說明：

無

五、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

無

六、發明說明：

【發明所屬之技術領域】

本發明係關於視窗的縮圖，尤其係關於在一視窗管理器中操作縮圖的介面與系統。

【先前技術】

電腦系統顯示許多視窗讓使用者可輸入資料、運用資料或啟用程序。若同時執行一個以上的應用程式，許多視窗可能會顯示在電腦顯示器上，每個視窗都對應至一應用程式。任何一個應用程式也可有多個視窗。例如：若使用者在文書處理程式內輸入文字，同時也在試算表程式內作業，在顯示器上可能會開啟兩個視窗。視窗之一是文書處理應用程式視窗，第二視窗則為試算表程式視窗。若使用者額外在媒體播放程式應用程式上觀看視訊，則會有對應至媒體播放程式應用程式的額外視窗。隨著啟用的應用程式數量增加，顯示在電腦顯示器上的視窗數量也會增加。

同時使用多個應用程式的使用者通常會面臨顯示器上有許多視窗，造成桌面上雜亂不堪，導致被雜亂的顯示器混淆並且失望。例如：顯示器上有許多重疊視窗會在使用者要完成不同應用程式內的工作時，讓使用者每次都要浪費時間找尋所要的視窗。為了舒緩此問題，使用者可將視窗縮到最小或完全退出應用程式，以便降低雜亂(以及顯示器上視窗的數量)。不過，若視窗已經最小化，則使用者就不再可以立即存取對應的應用程式。若需要在視窗已經最小化的特定應用程式內採取動作，使用者必須先找出所要的視窗，並且在找出視窗後開啟所要的視窗。此過程相當耗時。類似地，若使用者完全退出應用程式來清除桌面的

混亂情況，則該應用程式就無法再度啟用。在此情況下，使用者必須重新啟動應用程式來完成所要的工作，這樣會浪費更多時間。另外，若應用程式執行一個進行中的功能，則在應用程式未啟用期間該功能會喪失。

通常，當應用程式視窗最小化，工作列按鈕會顯示在工作列上，指出該應用程式目前已啟用。雖然工作列按鈕可指示應用程式已啟用，不過工作列按鈕中通常只有應用程式的圖示或名稱。在無額外資訊之下，使用者必須開啟視窗才能檢視視窗內容。不過，開啟與關閉視窗僅是要檢查視窗內容(例如確定視窗的主體性)，這樣會浪費時間與精力，導致喪失效率與生產力。當啟用的應用程式數量以及視窗數量增加，這個問題就會加劇，因為使用者必須開啟與關閉視窗才能找到所要的視窗。

如此，業界內需要一種系統與方法，用於提供方便的計畫存取，來運用視窗或應用程式縮圖並使用一種在縮圖與縮圖內容顯示方面有彈性的方式來表示縮圖。另外，業界內需要一種編程介面及方法，用於實施該編程介面以有效管理與實施應用程式視窗及縮圖，並已彈性方式表示該視窗與縮圖。

【發明內容】

本發明的態樣提供一種用於在顯示器上顯示一視窗的介面、系統或方法，藉以處理上述一或多個議題。在此也提供對應至一應用程式視窗的動態縮圖，其中應用程式視窗中內容的修改會反映在對應的動態縮圖內。應用程式視窗內容的修改可另即時反映在對應的動態縮圖內。

【實施方式】

本說明書分章節以幫助讀者閱讀。這些章節包含：簡介、一般用途電腦環境、縮圖。

吾人注意到，在下列說明中揭示元件之間的許多連接。吾人注意到，除非有指定，否則這些連接一般為直接或間接，並且本說明書內並不受限於此。

簡介

A. 靜態縮圖

在一方式中，縮圖可伴隨於該應用程式。有了縮圖，可以顯示小版本的應用程式視窗，指出應用程式視窗的原始內容，讓使用者識別視窗的大約內容。不過，這些縮圖可能不提供更新的應用程式視窗內容。如此，若已經對應用程式視窗做過修改，縮圖會過時並且不能提供應用程式視窗實際內容的正確影像。如此，使用者難以在不用開啟應用程式視窗本身的情況下正確識別所要的應用程式視窗。

另外，縮圖缺乏顯示內容、放置、尺寸以及/或效果方面的彈性。應用程式視窗的縮圖提供應用程式視窗原始內容的靜態影像。不過，不僅內容保持靜態並且可能過時，而且縮圖的特性以及配置可能已預定並且無法根據使用者在任何已知時間上的需求而改變。

電腦作業系統運用負責管理使用者介面物件，像是縮圖的軟體層，並且提供使用者介面服務給軟體應用程

式。此軟體層可稱為「桌面視窗管理器(Desktop Window Manager, DWM)」。桌面視窗管理器(DWM)的呈現邏輯、輸入事件路由以及應用程式程式設計介面(API)會集合起來具體實現使用者介面政策，其輪流定義作業系統的整體使用者體驗。在呈現桌面時，DWM 通常藉由直接協調繪製在顯示畫面上，來處理畫面更新，像是視窗移動。因此，若關閉或移開重疊的視窗，則必須重新繪製新顯示的下層視窗或桌面。這項處理相當緩慢，並且需要浪費相當多時間與資源。另外，當直接在特定位置上繪製視窗並且在顯示器上如平常所完成的特定組態內，對應的視窗影像會位於該特定位置與組態上。修改視窗的位置或組態需要浪費大量資源來重新繪製影像。

B. 動態縮圖

在本發明的其他態樣內，提供一種介面、系統及方法，用於在顯示器上顯示縮圖，其中桌面視窗管理器(DWM)提供一登錄來源視窗以及目的地視窗的應用程式編程介面(API)。在此範例中，目的地視窗為其中可呈現縮圖的視窗。呈現在目的地視窗內的縮圖可表示至少部分對應的來源視窗。因此，藉由提供被繪製到至少部分目的地視窗內之至少部分來源視窗，縮圖可表示來源視窗與目的地視窗之間的關聯。DWM 可接收對應至來源視窗的視窗代碼(windows handle, HWND)以及對應至目的地視窗的 HWND。應用程式可另接收一代碼，其呈現對應至來源視窗的縮圖登錄。

在本發明的其他態樣內，提供一種介面、系統及方法，用於在顯示器上顯示縮圖，其中桌面視窗管理器(DWM)更新或修改對應的來源視窗伴隨之縮圖。在此範例中，DWM 可接收定義目的地視窗大小的資料、當成縮圖繪製到目的地視窗內的來源視窗區域、縮圖的不透明度、縮圖的可見度、縮圖的特效等等。

本發明的進一步態樣另提供一種動態縮圖，其包含輸出一應用程式編程介面來登錄一縮圖代碼、維護縮圖登錄清單、根據來源視窗繪製縮圖或目的地視窗，其中來源視窗可為應用程式視窗、縮圖的更新特性以及移除登錄動態縮圖。另，修改來源視窗會自動處理，以便在目的地視窗內產生對應的修改。

吾人注意到，在下列說明中揭示元件之間的許多連接。吾人注意到，除非有指定，否則這些連接一般為直接或間接，並且本說明書內並不受限於此。

本發明的態樣提供一種應用程式編程介面(API)、系統以及方法，用於控制顯示元件的顯示，像是視窗或其他視窗的表示，像是顯示器上的縮圖。在本發明的一個範例中，另外的應用程式視窗版本類似於應用程式的縮小版本，並且顯示在顯示器上。應用程式視窗的縮小版本可另顯示在顯示器上一位置內，這個位置不同於原始應用程式視窗。另外，原始應用程式視窗可最小化，如此就不會顯示原始應用程式視窗，而應用程式視窗縮小版本會顯示在顯示器上。另外，應用程式視窗的縮小版本

本身會最小化，例如若要在顯示器上有額外空間時。當應用程式視窗的另外版本可為說明的應用程式視窗縮小版本，吾人注意到，根據使用者的需求，應用程式視窗的另外版本可大於原始應用程式視窗或任何其他大小。

例如：應用程式視窗的另外版本可為應用程式視窗的縮圖。縮圖通常為小影像或較大圖形的表示，或較大影像的一部分，對應至應用程式視窗。縮圖也可表示來源視窗之間的關聯、提供縮圖的來源或內容，以及可為視窗的目的地視窗，其中縮圖提供來自來源視窗的內容。縮圖提供在傳統格式內一次顯示許多視窗，然而在顯示器上使用的空間卻最小的優點。另外，縮圖可讓選擇與排列影像與視窗更加容易。

在本發明的其他態樣，需要時應用程式視窗的縮圖或另外版本可顯示在顯示器上。例如：將游標停在工作列按鈕上會喚起顯示應用程式視窗縮圖。在一個範例中，縮圖為動態縮圖並且可隨原始應用程式視窗修改而修改。例如：對於應用程式視窗的變更會反映在對應至應用程式視窗的動態縮圖內。例如這種變更可即時反映在動態縮圖內。

在本發明的其他態樣內，縮圖或動態縮圖可安排在其他視窗或目的地視窗內。例如：來自不同應用程式多個動態縮圖可包含在目的地視窗，如此目的地視窗會顯示來自每個不同應用程式的縮圖。另外，顯示在目的地視窗內的任何動態縮圖可反映目前對應的應用程式視窗狀

態。另外，縮圖可顯示在其他縮圖內或縮圖可串接，如此第一縮圖可顯示在第二縮圖內，而第二縮圖可顯示在第三縮圖內等等。

仍舊在本發明的其他態樣內，提供一種準備畫面外視窗的系統與方法。然後可達成在畫面上合成視窗，如此節省電腦資源。另外，準備畫面外視窗並且在畫面上合成視窗可啟用本發明內說明的縮圖功能性。

一般用途計算環境

第 1A 圖說明其上實施本發明的合適計算系統環境 100 之範例。計算系統環境 100 只為合適計算環境的一個範例，並不用於對本發明的使用或功能性領域做任何限制。也不可將計算環境 100 解釋成具有任何從屬物，或與範例操作環境 100 內說明的任一組件或組件組合有關之需求。

本發明可用許多其他一般目的或特殊目的計算系統裝置環境或組態來操作。本發明熟知適用的計算系統、環境與/或組態範例包含，但不受限於個人電腦、伺服器電腦、手持式或膝上型裝置、多處理器系統、微處理器式系統、可程式設計消費者電子裝置、網路 PC、迷你電腦、主機電腦、包含上述任一系統或裝置的分散式計算環境等等。

請參閱第 1A 圖，實施本發明的說明系統包含形式上為電腦 110 的一般用途計算裝置。電腦 110 的組件可包含但不限於一個處理單元 120、一個系統記憶體 130 以

及可將包含系統記憶體之許多系統組件耦合到處理單元 120 之系統匯流排 121。系統匯流排 121 可以是許多種匯流排結構，包含記憶體匯流排或記憶體控制器、週邊匯流排以及使用許多匯流排架構任一之本機匯流排。藉由範例並且不做限制，這種架構包含工業標準架構 (ISA) 匯流排，微通道架構 (MCA) 匯流排、強化型 ISA (EISA) 匯流排、視訊電子標準協會 (VESA) 本機匯流排以及週邊組件互連 (PCI) 匯流排，也就是已知的 Mezzanine 匯流排。

電腦 110 一般包含許多電腦可讀取媒體，電腦可讀取媒體包含揮發與非揮發性媒體、可移除與不可移除的媒體。藉由範例，但不受限於此，電腦可讀取媒體可包括電腦儲存媒體以及通訊媒體，並且包含但不受限於 RAM、ROM、EEPROM、快閃記憶體或其他記憶體技術、CD-ROM、數位多功能光碟 (DVD) 或其他光學碟片儲存、磁帶、磁匣、磁片儲存或其他磁性儲存裝置，或其他任何可用於儲存所要資訊並且可由電腦 110 存取的媒體。通訊媒體一般為電腦可讀取指令、資料結構、程式模組或其他調變資料信號內的資料，像是載波或其他傳輸機制並且包含任何資訊傳遞媒體。「調變資料信號」一詞表示具有一或多個特徵集或以信號內編碼資訊這種方式來變更的信號。藉由範例並且不受限於此範例，通訊媒體包含像是有線網路或直接有線連線的有線媒體，以及像是聲音、RF、紅外線以及其他無線媒體的無線媒體。上述任何媒體的組合也應該包含在電腦可讀取媒體

的領域內。

系統記憶體 130 包含揮發與/或非揮發性記憶體形式的電腦儲存媒體，像是唯讀記憶體(ROM) 131 以及隨機存取記憶體(RAM) 132。基本輸入/輸出系統 133 (BIOS) 包含有助於在電腦 110 內元件之間(像是啟動時)傳輸資訊的基本常式，其一般儲存在 ROM 131 內。RAM 132 一般包含可立即存取或在處理單元 120 上操作呈現的資料與/或程式模組。藉由範例但不受限於此範例，第 1 圖說明作業系統 134、應用程式 135、其他程式模組 136 以及程式資料 137。

電腦 110 也可包含其他可移除/不可移除、揮發/非揮發性電腦儲存媒體。藉由範例，第 1 圖說明讀取或寫入不可移除、非揮發磁性媒體的硬碟機 140、讀取或寫入可移除、非揮發性磁碟 152 的磁碟機 151 以及讀取或寫入可移除、非揮發性光碟 156，像是 CD-ROM 或其他光學媒體的光碟機 155。其他可在範例作業環境內使用的可移除/不可移除、揮發/非揮發性電腦儲存媒體包含但不受限於：磁帶、快閃記憶卡、數位多功能光碟、數位影帶、固態 RAM、固態 ROM 等等。硬碟機 141 一般透過像是介面 140 這種不可移除記憶體介面連接至系統匯流排 121，並且磁碟機 151 與光碟機 155 一般透過像是介面 150 這種可移除記憶體介面連接至系統匯流排 121。

上面討論以及第 1A 圖內說明的磁碟機(光碟機)以及隨附之電腦儲存媒體提供了電腦可讀取指令、資料結

構、程式模組或其他電腦 110 資料的儲存。例如在第 1A 圖內說明硬碟機 141 用於儲存作業系統 144、應用程式 145、其他程式模組 146 以及程式資料 147。請注意，這些組件可與作業系統 134、應用程式 135、其他程式模組 136 以及程式資料 137 相同或不同。在此賦予作業系統 144、應用程式 145、其他程式模組 146 以及程式資料 147 不同的編號，以說明至少這些是不同的版本。使用者可透過輸入裝置，像是鍵盤 162 與指標裝置 161(通稱為滑鼠、軌跡球或觸控板)將命令與資訊輸入電腦 20 內。其它輸入裝置(未顯示)可包含麥克風、搖桿、遊戲控制器、衛星碟、掃描器等等。這些與其他輸入裝置透過與系統匯流排耦合的使用者輸入介面 160 連接到處理單元 120，但也可由其他介面與匯流排結構來連接，像是並列埠、遊戲埠或萬用序列匯流排(USB)。監視器 191 或他種顯示裝置也可透過介面，像是影像介面 190，連接到系統匯流排 121。除了監視器以外，電腦也可包含其他週邊輸出裝置，像是喇叭 197 與印表機 196，這些可透過輸出週邊介面 190 來連接。

電腦 110 可使用邏輯連接至一或多部遠端電腦，像是遠端電腦 180 來在網路環境內運作。遠端電腦 180 可為個人電腦、伺服器、路由器、網路 PC、點裝置或其他共用網路節點，並且通常包含上述有關電腦 110 的許多或全部元件，不過第 1A 圖內只顯示一個記憶體儲存裝置 181。第 1A 圖內說明的邏輯連接包含區域網路(LAN) 171

以及廣域網路(WAN) 173，但是也可包含其他網路。這種網路環境在辦公室、企業電腦網路、內部網路以及國際網路內相當稀鬆平常。

當用於 LAN 網路環境內，電腦 110 透過網路介面或配接器 170 連接到 LAN 171。當用於 WAN 網路環境內，電腦 110 一般包含數據機 172 或其他在 WAN 173 上建立通訊的構件，像是網際網路。內接或外接數據機 172 可透過使用者輸入介面 160 或其他適當機制連接至系統匯流排 121。在網路環境中，與電腦 110 相關的程式模組或部分程式模組可儲存在遠端記憶體儲存裝置內。藉由範例但不受限於此範例，第 1A 圖說明位於記憶體裝置 181 上的遠端應用程式 185。吾人可瞭解所顯示的網路連線為範例，所以可使用在電腦之間建立通訊鏈結的其他構件。

吾人可瞭解所顯示的網路連線為範例，所以可使用在電腦之間建立通訊鏈結的其他構件。假設有任何一種已知的通訊協定存在，像是 TCP/IP、乙太網路、FTP、HTTP 等等，並且系統可在主從式架構中運作，允許使用者從網路伺服器擷取網頁。任何一種傳統網路瀏覽器都可用於顯示與操縱網頁上的資料。

編程介面(或更簡單的「介面」)可看成任何機制、處理、通訊協定，可讓一或多程式碼區段與一或多其他程式碼區段提供的功能性通訊或存取。另外，編程介面可看成一系統組件的一或多機制、方法、函數呼叫、模組、

物件等等，其可通訊耦合至其他組件的一或多機制、方法、函數呼叫、模組等等。前面句子中的「程式碼區段」一詞用於包含一或多指令或程式碼列，並且包含例如程式碼模組、物件、副常式、函數等等，而不管套用的詞彙或程式碼區段是否個別套用、或是否提供程式碼區段當成來源碼、中間碼或物件碼、程式碼是否用於執行時間系統或處理，或其是否位於相同或不同機器上或分布通過多部機器，或程式碼區段所表示的功能性是否在整個軟體、整個硬體或硬體與軟體的組合內實施。

在想法上，編程介面一般可看成如第 1B 圖或第 1C 圖內所示。第 1B 圖說明介面 Interfacel 當成溝通第一與第二程式碼區段的管道。第 1C 圖說明一個介面，其包含介面物件 I1 和 I2 (可能是或不是部份第一第二程式碼區段)，其可讓系統的第一或第二程式碼區段透過媒體 M 通訊。從第 1C 圖看起來，吾人可考慮介面物件 I1 和 I2 當成相同系統的個別介面，並且也可考慮物件 I1 和 I2 加上媒體 M 構成介面。雖然第 1B 和 1C 圖顯示雙向流動以及每一側流動上的介面，特定實施只有在一個方向內有資訊流(或如下所述無資訊流)，或只有在一側上有介面物件。藉由範例，並且不受此限制，像是應用程式編程介面(API)、記錄點、方法、函數、副常式、遠端程序呼叫以及組件物件模型(COM)介面等詞都涵蓋在編程介面的定義內。

這種編程介面的態樣可包含藉此該第一程式碼區段將

資訊(在此廣為使用的「資訊」包含資料、命令、要求等等)傳輸至第二程式碼區段的方法、藉此該第二程式碼區段接收資訊的方法，以及資訊的結構、順序、語法、組織、法則、時機與內容。在此方面，下層傳輸媒體本身對於介面的操作、媒體為有線或無線或這兩者的組合以及該資訊是否由介面定義的方式傳輸可能不重要。在特定情況下，以傳統觀念來說資訊不會單向或雙向傳遞，而是資訊傳輸可透過其他機制(例如資訊放在與程式碼區段之間資訊流分開的緩衝區、檔案等等之內)或不存在，或是當一個程式碼區段簡單存取由第二程式碼區段執行的功能性。任何或所有這些態樣在已知情況下都很重要，例如根據程式碼區段是否為鬆弛耦合或緊密耦合組態內系統的一部分，如此此清單應該考慮說明並不受限制。

這個編程介面的概念為精通此技術的人士所熟知，並且從前面本發明的詳細說明中已經釐清。不過，還有其他方式實施編程介面，並且除非明確排除，否則全部包含在本說明書結尾上揭示的申請專利範圍內。這種其他方式似乎比從第 1B 圖和第 1C 圖簡化圖更精巧或複雜，儘管如此其執行類似功能來達成相同的結果。此時我們將簡單說明編程介面的某些替換實施。

分解

利用將通訊中斷到多個分散通訊中，可間接達成從一個程式碼區段到另一個的通訊。這用圖解方式說明於第

1D 圖和第 1E 圖內。如所示，某些介面可用可分的功能集合來說明。如此，第 1B 圖和第 1C 圖的介面功能性可分解來達成相同的結果，就像可用數學方式 $2 \times 2 \times 3 \times 2$ 產生 24 一樣。因此，如第 1D 圖內所示，介面 Interfacel 提供的功能可細分來將介面通訊轉換為多介面 InterfacelA、InterfacelB、InterfacelC 等等，而達成相同的結果。如第 1E 圖內所示，由介面 I1 提供的功能可細分為多介面 I1a、I1b、I1c 等等，而達成相同的結果。類似地，接收來自第一程式碼區段資訊的第二程式碼區段介面 I2 可分解為多個介面 I2a、I2b、I2c 等等。分解時，包含第一程式碼區段的介面數量並不需要與包含第二程式碼區段的介面數量吻合。在第 1D 圖和第 1E 圖的情況下，介面 Interfacel 和 I1 的功能精神分別與第 1B 圖和第 1C 圖的一樣。介面的分解也遵照連帶、相互以及其他數學特性，如此會難以認出分解。例如，操作順序可能不重要，因此介面帶有的功能可在到達介面之前由另外一些程式碼或介面執行，或由系統的個別組件執行。再者，精通編程技術的人士可瞭解，有許多製作不同函數呼叫而達成相同結果的方式。

重新定義

在某些情況下，可以忽略、新增或重新定義編程介面的特定態樣(例如參數)，而仍舊達成所要的結果。這說明於第 1F 圖和第 1G 圖內。例如：第 1B 圖的假設介面 Interfacel 包含一函數呼叫 Square(輸入，精度，輸出)，

包含三個參數：輸入、精度、輸出的呼叫，並且從第一程式碼區段發出至第二程式碼區段。若中間參數精度在已知案例中無關緊要，如第 1F 圖內所示，則可以忽略或甚至用無意義(在此情況下)參數取代。另外也可加入無關的額外參數。在這兩事件中，可達成平方的功能性，而在輸入用第二程式碼區段平方之後送回輸出。精度對於某些下游或計算系統其他部分而言是相當有意義的參數，不過一旦確認精度不需要用於計算平方，則可取代或忽略。例如：取代傳遞有效的精度值，可傳遞像是生日這種無意義值而不會結果有不利的影響。類似地，如第 1G 圖內所示，介面 I1 由介面 I1' 取代，重新定義為忽略或新增參數至介面。介面 I2 可類似重新定義為介面 I2'，重新定義來忽略非必要的參數或在別處已經處理過的參數。這裡的重點在於在某些情況下，一編程介面可包含像是參數這種領域，對於某些目的這些並不需要，所以其可忽略或重新定義，或在別處處理用於其他目的。

線內編碼

將兩個別程式碼模組的某些或所有功能性融合在一起也可行，如此其間的「介面」就會改變形式。例如：第 1B 圖和第 1C 圖的功能性可分別轉換成第 1H 圖和第 1I 圖的功能性。在第 1H 圖內，之前第 1B 圖的第一與第二程式碼區段會融入包含這兩者的模組內。在此案例中，程式碼區段仍舊可彼此通訊，但是介面會調整成更適合單一模組的形式。如此，例如不再需要正式 Call 和 Return

陳述式，但是根據介面 Interfacel 的類似處理或回應則仍就有效。類似地，如第 1I 圖內所示，來自第 1C 圖的部分(或全部)介面 I2 可線內寫入介面 I1 來形成介面 I1"。如所示，介面 I2 區分成 I2a 和 I2b，並且介面部分 I2a 已經用介面 I1 線內編碼形成介面 I1"。針對具體範例，考慮來自第 1C 圖的介面 I1 執行函數呼叫平方(輸入，輸出)，由介面 I2 接收，而在處理由第二程式碼區段用輸入傳遞值之後(用於計算輸入的平方)，用輸出傳送回平方的結果。在這種情況下，由第二程式碼區段(平方輸入)執行的處理可由第一程式碼區段執行，而不用呼叫該介面。

分離

利用將通訊中斷到多個分散通訊中，可間接達成從一個程式碼區段到另一個的通訊。這用圖解方式說明於第 1J 圖和第 1K 圖內。如第 1J 圖內所示，提供一或多段程式碼(分離介面，因為其從原始介面終將功能性以及/或介面功能分離)來轉換第一介面 Interfacel 上的通訊，確認其到達不同介面，在此案例中為介面 Interface2A、Interface2B 和 Interface2C。這可達成，例如有一已經安裝的應用程式基礎指定與根據 Interfacel 通訊協定與作業系統通訊，但是然後作業系統改變使用不同的介面，在此案例中為介面 Interface2A、Interface2B 和 Interface2C。這是因為第二程式碼區段所使用的原始介面已經改變，如此就不再與第一程式碼區段所使用的介

面相容，因此使用中間媒介讓新與舊介面相容。類似地，如第 1K 圖內所示，第三程式碼區段可導入分離介面 DI1，接收來自介面 I1 的通訊以及與分離介面 DI2 的通訊，將介面功能性傳輸至例如介面 I2a 和 I2b，重新設計運用 DI2，但是提供相同的功能結果。類似地，DI1 和 DI2 可一起運作，將第 1C 圖介面 I1 和 I2 的功能性轉譯成新的作業系統，提供相同或類似的功能結果。

重新撰寫

仍舊其他可能的改變為動態重新撰寫程式碼，用其他取代介面功能性但是達成相同整體結果。例如：有一種系統，其中提供呈現在中間語言(例如 Microsoft IL、Java ByteCode 等等)內的程式碼區段給執行環境內的 Just-in-Time (JIT) 編譯器或解析器(就像是由 .Net framework、Java Runtime 環境或其他類似 Runtime 類型環境所提供)。JIT 編譯器可撰寫成將通訊從第一程式碼區段動態轉換成第二程式碼區段，即是確認其可能為第二程式碼區塊(原始或不同的第二程式碼區塊)所需的不同介面。這說明於第 1L 圖和第 1M 圖內。如可從第 1L 圖所見，此方式類似於上述的分離案例。這可達成，例如有一已經安裝的應用程式基礎指定與根據 Interfacel 通訊協定與作業系統通訊，但是然後作業系統改變使用不同的介面。JIT 編譯器可用於確認從已安裝應用程式到作業系統新介面的進行之通訊。如第 1M 圖內所示，此動態重新撰寫介面的方法可套用至動態因素，或

改變介面。

吾人也應該注意，上述透過不同具體實施例的介面可達成相同或類似結果之案例也可用許多方式組合，串聯以及/或並聯，或用其他調停程式碼。如此，上面呈現的另一具體實施例並不互斥並且可以混合、匹配並且組合，來產生與第 1B 圖和第 1C 圖內所示一般案例相同或同等案例。吾人應該注意，對於大多數編程架構而言，還有此處未說明的其他類似方法可達成相同或類似介面功能性，但是儘管如此還是涵蓋在本發明的精神與領域內，即是吾人注意到，至少所表示並且可導致好處的部分功能性位於介面價值之下。

縮圖

第 4A 圖和第 4B 圖圖示其中提供應用程式視窗縮圖的一個本發明應用程式範例。在第 4A 圖的範例中，說明包含工作列按鈕的工作列 400。每個工作列按鈕對應至現用的應用程式。工作列按鈕 402 對應至目前在電腦上執行的「接龍」應用程式。類似地，工作列按鈕 403 對應至文書處理應用程式視窗、工作列按鈕 405 對應至媒體播放程式應用程式視窗並且工作列按鈕 406 對應至「小算盤」應用程式視窗。「接龍」縮圖 401 提供用於「接龍」應用程式視窗，並且「媒體播放程式」縮圖 404 提供用於「媒體播放程式」應用程式。對應的縮圖可以有許多喚起方式。在一非限制的範例中，利用將游標停在對應的工作列按鈕上就可喚起縮圖。例如：利用將游標

停在「接龍」工作列按鈕 402 上就可喚起「接龍」縮圖 401。

另外如第 4A 圖內所示，動態縮圖可提供對應至應用程式的更新或「現場」內容。例如：「媒體播程式」可正在播放影片。隨著影片在「媒體播程式」應用程式視窗(未顯示)內播放，影片也會在「媒體播程式」縮圖 404 內播放。若媒體播程式應用程式本身已經最小化，則媒體播程式縮圖 404 顯示媒體播程式應用程式在最小化之前的最後畫面。另外，即使媒體播程式應用程式視窗本身已經最小化，不過其繼續播放影片，則媒體播程式縮圖 404 可繼續播放影片。

第 4B 圖說明本發明的其他縮圖範例。說明工作列 410 其中包含對應至「接龍」應用程式視窗(未顯示)的「接龍」工作列按鈕 411、對應至文書處理應用程式視窗(未顯示)的文書處理工作列按鈕 412、對應至媒體播程式應用程式視窗(未顯示)的媒體播程式工作列按鈕 413 以及對應至小算盤應用程式視窗(未顯示)的小算盤工作列按鈕 414。在此範例中，提供「小算盤」縮圖 415。「小算盤」縮圖 415 顯示目前以及更新的「小算盤」應用程式內容。在本發明的一個具體實施例內，「小算盤」縮圖 415 也可更新應用程式視窗。例如：透過「小算盤」縮圖 415 可執行計算來達成算術結果。此結果可顯示在「小算盤」縮圖 415 以及「小算盤」應用程式視窗(未顯示)內。在此範例中，"=" 鍵選自於「小算盤」縮圖 415 內

並且在「小算盤」縮圖內顯示結果(即是"4")。即使計算透過「小算盤」縮圖 415 來執行，此結果也反映在「小算盤」應用程式視窗(未顯示)內。

第 5 圖說明其中縮圖顯示在 ALT-TAB 功能表內的其他本發明縮圖應用。如第 5 圖內所示，顯示器 500 顯示應用程式。在此情況下，在顯示器 500 上可看見部分「小算盤」應用程式視窗 502 和部分「接龍」應用程式視窗 501。每個應用程式在工作列中都具有對應的工作列按鈕。「小算盤」應用程式具有對應的「小算盤」工作列按鈕 503，並且「接龍」應用程式具有對應的「接龍」工作列按鈕 504。「媒體播放程式」應用程式也啟用(未顯示)，並且對應的「媒體播放程式」工作列按鈕 506 也呈現在工作列內。

在第 5 圖說明的範例內，提供顯示現用應用程式的動態縮圖之 ALT-TAB 功能表 505。在此範例內，每一動態縮圖都顯示對應應用程式視窗的「現場」(即是已更新的)內容。動態縮圖內容可即時更新。隨著應用程式視窗修改或更新，對應的動態縮圖也會據此更新。動態縮圖的更新也可即時達成。在此範例中，ALT-TAB 功能表 505 包含每一現用應用程式的動態縮圖，不管是否可在顯示器上看見對應的應用程式之應用程式視窗。「小算盤」縮圖 507、「媒體播放程式」縮圖 508 以及「接龍」縮圖 509 說明於 ALT-TAB 功能表 505 內，而已選取「接龍」縮圖 509。每一動態縮圖都提供「現場」內容。例如：即時在

「媒體播放程式」縮圖 508 內播放影片。

第 2A 圖和第 2B 圖圖解說明範例顯示器的組件。第 2A 圖內將桌面呈現為場景圖形或顯示器樹狀當成元件 200，並且包含桌面桌布 201 和兩應用程式：「小算盤」202 和「接龍」203。此外，桌面另包含一工作列 208。在此表示方法內，視窗訊框以及每個視窗的用戶區域都為場景圖形的所有部份。視窗移動變成改變顯示樹狀內的轉換值。此範例中的每個應用程式都分別包含一個視窗訊框 204、205 和視窗內容 206、207。工作列 208 包含視窗內容 209。在第 2A 圖內，顯示器上的元件可從左到右呈現。因此，桌面桌布 201 呈現在「小算盤」203 應用程式視窗之後，而「小算盤」則呈現在「接龍」202 應用程式視窗之後。

第 2B 圖說明顯示器，如第 2A 圖內所示。該顯示器包含背景的桌面桌布 220。「小算盤」應用程式視窗 222 顯示在桌面桌布之前，並且「接龍」應用程式視窗 221 顯示在「小算盤」應用程式視窗 222 和桌面桌布 220 之前。如第 2B 圖內所示，每一應用程式視窗都可由工作列 225 內的工作列按鈕表示。在此範例中，「小算盤」應用程式由工作列 225 內的工作列按鈕 223 表示，並且「接龍」應用程式由工作列 225 內的工作列按鈕 224 表示。每一「小算盤」應用程式視窗 222 和「接龍」應用程式視窗 221 都包含一視窗訊框以及視窗內容，如桌面桌布 220 上所顯示。

如第 2A 圖內說明的任何樹狀子節點都可用來當成其他樹狀節點的內容。如此，從樹狀內其他地方可看見部分樹狀的內容。在此有許多達成在樹狀內其他位置上彩線節點的方法。對於無限制範例來說，可使用 Visual Brush。「VisualBrush」為一種技術，用於參照場景圖形的現有部分，但是用圖形範圍的「目前」集合來呈現，像是位置、比例、旋轉、不透明、特效等等。這導致部分樹狀呈現在一個要採現在別處的地方上。例如：若想要「小算盤」應用程式視窗的動態縮圖，則呼叫適當的編程介面(例如應用程式編程介面或 API)提供縮圖主控 HWND 或視窗代碼。此範例圖示於第 3A 圖和第 3B 圖內。

第 3A 圖為說明本發明顯示器組件範例的場景圖形或顯示樹狀圖。第 3A 圖內將桌面呈現為元件或節點 300，並且包含桌面桌布 301 和兩應用程式：「小算盤」302 和「接龍」303。此範例中的每個應用程式都分別包含一個視窗訊框 305、306 和視窗內容 307、308。此範例內的桌面另包含一工作列 307。

在此範例中，由呼叫適當的 API 來提供「小算盤」應用程式視窗的縮圖。如第 3A 圖內所示，縮圖主控 HWND 304 視窗代碼由應用程式建立，並登錄應用程式的縮圖，像是「小算盤」應用程式。例如在第 3A 圖內，工作列 307 包含縮圖主控 HWND 304 以及視窗內容 310。縮圖主控 HWND 304 包含一「小算盤」縮圖 311，並且可將應用程式視窗的縮圖附加至縮圖主控。第 3A 圖內

的點狀箭頭說明縮圖主控 HWND 304 的「小算盤」縮圖 311 往回點，或從「小算盤」應用程式視窗呈現「小算盤」。結果，提供呈現出對應的應用程式視窗已更新內容（在此情況下為「小算盤」應用程式視窗），而不是僅是應用程式視窗內容的靜態檢視之「現場」或動態縮圖。

如第 2A 圖的範例內，第 3A 圖內所表現的顯示器上元件可從左到右呈現。因此，桌面桌布 301 呈現在「小算盤」302 應用程式視窗之後，而「小算盤」則呈現在「接龍」303 應用程式視窗以及工作列 208 之後。第 3B 圖圖示根據第 3A 圖的顯示器。如第 3B 圖所示，該顯示器包含背景內的桌面桌布 320。「小算盤」應用程式視窗 322 顯示在桌面桌布 320 之前，並且「接龍」應用程式視窗 321 顯示在「小算盤」應用程式視窗 322 和桌面桌布 320 之前。同時也顯示工作列 326。每個應用程式都伴隨於工作列 326 中對應的工作列按鈕。在此範例中，「小算盤」應用程式伴隨於工作列按鈕 324，並且「接龍」應用程式伴隨於工作列按鈕 325。此外，第 3B 圖說明所提供伴隨於「小算盤」應用程式的縮圖。「小算盤」應用程式與視窗 322 伴隨於工作列 326 內的工作列按鈕 324，並且「接龍」應用程式與視窗 321 伴隨於工作列按鈕 325。

另外如第 3B 圖內所示，「小算盤」應用程式視窗 322 的動態縮圖 323。縮圖 323 由往回點或呈現「小算盤」應用程式 302 的視窗訊框 305 與視窗內容 308 的縮圖主控 HWND 304 之「小算盤縮圖」節點 311 提供，如第 3A

圖內所說明。結果為「現場」的「小算盤」應用程式 322 之動態縮圖 323，即是縮圖隨對應用程式視窗本身進行的對應修改而修改。如第 3B 圖內所示，「小算盤」應用程式視窗 322 內計算出來的"243"值也反映在對應的縮圖 323 內。

在本發明的態樣中，透過登錄(register)、移除登錄(unregister)以及更新縮圖特性來達成動態縮圖。這種登錄、移除登錄以及更新縮圖特性或修改可透過桌面視窗管理器(DWM)內的應用程式介面(API)來達成。在此說明達成本發明縮圖的三個範例 API。

API 範例 1 - 縮圖的登錄

縮圖可從視窗產生，例如應用程式視窗。在此情況下，應用程式視窗為來源視窗，應用程式視窗提供縮圖內容的基礎。在一個範例中，縮圖為應用程式視窗的最小化版本，並且包含應用程式視窗的整個內容。另外，縮圖可只包含所選的來源視窗部分。

來自來源視窗的縮圖內容會呈現到第二區域內。此第二區域根據來源視窗指定容納縮圖內容的目的地視窗，此目的地視窗可形成縮圖本身的基礎。例如：應用程式視窗以外位置內的小視窗訊框、顯示器內的指定區域或蹦現式功能表可用來當成目的地視窗，來顯示根據應用程式視窗的縮圖。

當縮圖已經登錄，可執行建立來源視窗與目的地視窗之間關係的操作。視窗代碼(HWND)可指出用於指定目

的地視窗。指定的目的地視窗可當成縮圖呈現的目標，並且由呼叫或登錄縮圖的處理所擁有。若登錄縮圖的處理也擁有目的地視窗，則可避免由其他應用程式對該應用程式進行不想要的改變。類似地，來源視窗的 HWND 也可指定。

縮圖的快取位元圖大小也可指定。例如：最小的大小可指示系統已經收到來源視窗的最後已知位元圖之通知。如此，若來源視窗之後最小化並且在顯示器上看不見，則還是維持視窗的位元圖並且縮圖仍舊可見。大小可根據呼叫的應用程式需求而變。若相同來源視窗有多重登錄，處理會根據目前的需求決定適用哪種縮圖大小。例如：在變化中縮圖大小的多重登錄情況下，可使用最大的大小。

API 也可包含表示縮圖登錄的代碼。此代碼對於發出呼叫的處理而言是唯一性的，如此縮圖稍後之能從其所登錄的處理中移除登錄。

用於登錄縮圖的 API 範例如下：

```

Typedef HANDLE HTHUMBNAIL;
HRESULT
DwmRegisterThumbnail (
    HWND                dstWindow,
    HWND                srcWindow,
    SIZE                *lpMinimizedSize,
    OUT HTHUMBNAIL     * lphThumbnail

```

);

其中 `dstWindow` 表示目的地視窗、`srcWindow` 表示來源視窗、`lpMinimizedSize` 表示一旦 `srcWindow` 已經最小化，則其上系統會快取來源視窗(`srcWindow`)的最後已知位元圖之大小，並且 `lphThumbnail` 為一傳回值，其表示該唯一性代碼表示在呼叫中所做的登錄。

API 範例 2 - 縮圖特性的修改或設定

在縮圖登錄之後，來自來源視窗或其一部分的內容會放置成為目的地視窗內的縮圖。例如：呼叫時，更新的縮圖特性 API 可修改或更新目的地視窗內縮圖的呈現特性。例如：UpdateThumbnail Properties 可用於設定縮圖的特性，像是可見度、形狀、矩形等等。Update 縮圖特性 API 可指示要更新的縮圖代碼。對縮圖的更新包含縮圖的任何修改，包含但不受限於縮圖的位置、縮圖大小、縮圖所使用的來源視窗部份、縮圖所使用的目的地視窗部份、縮圖的不透明或透明度、縮圖的可見度、縮圖的特效(例如旋轉、扭曲、破碎等等)等等。

在其他範例中，縮圖可從其所登錄的處理中更新或修改。在一個範例中，重複呼叫縮圖代碼至目的地視窗，如此縮圖可再目的地視窗四週移動。在此情況下，縮圖可在目的地視窗內重新定位。例如：在 ALT-TAB 視窗內，利用重新呼叫縮圖至目的地視窗，改變縮圖的排列順序。

參數可用於縮圖特性的修改或設定。這裡有許多參數

用於修改或設定縮圖特性的範例，像是但不受限於旗標。透過使用參數，API 結構可更容易修改，讓 API 更有彈性在未來進行更新。參數可指定要使用的縮圖之特性，並且本身可由特定常數來定義。例如：當想要用少於整個來源視窗來用於縮圖，則參數可定義並對應至來源視窗參數，其定義要用來當成目的地視窗內縮圖的來源視窗區域或部分。在此範例中，只有由來源視窗參數所指定的來源視窗特定部份用於目的地視窗內的縮圖。在此範例中，用來在目的地視窗內當成縮圖的來源視窗部分會以顯示器的座標來指示。例如：若所要的部分為來源視窗內一矩形，API 可指定一參數來指出矩形與來源視窗的交叉座標，以獲得要當成縮圖的最終矩形。如範例所示，座標(0, 0)可設定為視窗區域的左上角。定義並對應至所要來源視窗參數的參數可設定，以指示只有特定來源視窗部分用來當成縮圖，指定的來源視窗部分會由提供的座標所定義。雖然有許多方式可指定來源視窗的座標以及所要部份，其中一個範例可包含使用由所提供矩形與交叉的實際來源視窗之交叉來定義部分來源視窗。

另外，要用於縮圖的部分目的地視窗可透過定義與對應至目的地視窗參數的參數來指定。在此情況下，不想要整個視窗用於縮圖並且只有部分目的地視窗要用於縮圖，則可由目的地視窗參數指定其中縮圖要呈現的目的地視窗區域。若有此情況，有一種方式只呈現由目的地

視窗參數定義的所要指定區域(即是當呈現發生在指定區域之外則裁切影像)。另外，一種決定部分目的地視窗中縮圖可繪製位置的方法為決定要使用的部分目的地視窗與目的地視窗本身的交叉。這會避免呈現到指定區域之外。另，定義與對應至目的地視窗參數的參數可用於指定該指定區域內來自該來源視窗用來當成縮圖的區域。在此範例中，若未定義目的地視窗參數，則整個指定的視窗可用來當成縮圖內容。

在 API 內可指定任何數量的其他縮圖特性。例如：可指定縮圖的不透明或透明度。在本發明此範例的一個態樣內，可指派參數來指定縮圖的不透明度(或透明度)，例如可指派 255 當成完全不透明，而 0 則為完全透明。如此，若設定定義與對應至不透明參數的參數，則適用於縮圖不透明度的指定指派數值。結果，縮圖可展現出使用者所要的不透明變化程度。

根據額外範例，縮圖可根據需求製作成可見或不可見。可提供定義與對應至能見度參數的參數。能見度參數可為布林值，如此當能見度參數為「真」，縮圖就可見，若能見度參數為「偽」，則看不見縮圖。本發明並不限於縮圖的任何特定修改，縮圖的任何修改都位於本發明範疇內。

用於更新或修改縮圖的 API 範例為 DwmUpdateThumbnailProperties，如下所示：

HRESULT

```

DwmUpdateThumbnail Properties (
    HTHUMBNAIL hThumbnail,
    DWM_THUMBNAIL_PROPERTIES *ptnProperties
);

#define DWM_TNP_RECTDESTINATION 0X00000001
#define DWM_TNP_RECTSOURCE 0X00000002
#define DWM_TNP_OPACITY 0X00000004
#define DWM_TNP_VISIBILITY 0X00000008

```

```

typedef struct_DWM_THUMBNAIL_PROPERTIES
{
    DWORD          dwFlags;
    RECT           rcDestination;
    RECT           rcSource;
    BYTE           opacity;
    BOOL           fVisible;
} DWM_THUMBNAIL_PROPERTIES;

```

其中 dwFlags 定義參數或旗標，其對應至指定要使用的後續特性之參數。在此範例中，rcSource 參數為指定要用來當成縮圖的來源視窗(srcWindow)區域之來源視窗參數、rcDestination 為指定縮圖要呈現的目的地視窗(dstWindow)之目的地視窗參數、opacity 為控制縮圖不透明度的不透明度參數(例如設定從 0 至 255)，並且

fVisible 為根據布林值指定縮圖不可見或可見的能見度參數。

API 範例 3 - 縮圖的移除登錄

當不在需要時，則移除來源與目的地視窗之間的關聯。例如：若視窗不再啟用，則利用移除登錄關聯可達成視窗與縮圖之間的關聯。這可釋放資源。再者，當退出登錄處理，則可另釋放縮圖。用於移除登錄縮圖的 API 範例為 DwmUnregisterThumbnail，說明如下：

HRESULT

```
DwmUnregisterThumbnail (
    HTHUMBNAIL hThumbnail
);
```

第 6 圖為說明用於提供本發明縮圖，包含「現場」或動態縮圖的處理範例流程圖。在步驟 601 內，由應用程式要求第一頂層視窗或來源視窗顯示一縮圖，其也可位於第二頂層視窗或目的地視窗的區段內，來提供 API。因此，第一頂層視窗或來源視窗可呈現成目的地視窗內的最上層物件。此實施的一個無限制範例為 ALT-TAB 視窗，其中在該 ALT-TAB 視窗(即是目的地視窗)內提供對應至來源視窗的縮圖。

在步驟 602 內，對 DWM 呼叫 Register 可提供縮圖代碼。在相同的來源視窗或相同的目的地視窗上也可有多個呼叫。在 ALT-TAB 範例內，相同來源視窗與相同目的地視窗上多個呼叫會導致 ALT-TAB 視窗內的縮圖重新

配置，可在選擇縮圖時提供較大以及/或具風格的應用程式縮圖呈現。在其他範例中，縮圖的 Z 順序可由後續登錄呼叫改變。若要縮圖的 z 順序相對於其他顯示元件而修改，可用其他縮圖登錄呼叫來登錄新的縮圖。在設定對應至舊縮圖的新縮圖之特性後，就可移除登錄舊縮圖。如此，在此範例中，縮圖的 Z 順序可由縮圖的多重登錄改變。

另外，對於相同目的地視窗可呼叫多個來源視窗。在此範例中，對一個目的地視窗呼叫多個來源視窗導致在目的地視窗內提供多個不同的縮圖。

仍舊在其他範例中，在其他動態縮圖內可顯示動態縮圖。例如：具有較高 z 順序的第一動態縮圖可放置在具有較低 z 順序的應用程式較大動態縮圖內。在此範例中，縮圖內嵌入其他縮圖內。另，特效可套用至縮圖或任何嵌入的縮圖。另外，任何 z 順序的縮圖可串接至任何 z 順序的其他縮圖。例如：至少一個縮圖(不管 z 順序)可放置在任何 z 順序的應用程式之較大動態縮圖內。類似地，較大動態縮圖可放置在任何 z 順序的其他動態縮圖內。如此，縮圖可串接而不管 z 順序。

在步驟 603 內，DWM 登錄處理 /ID 對給縮圖。DWM 可另維持縮圖登錄的全域清單，其中每個應用程式都與對應的縮圖代碼(HWND)配對。全域清單可根據使用者的需求用許多方式維持。例如：維持全域清單的合適方式為利用目的地視窗(dstWindow)將縮圖納入索引，如此

任何所要的目的地視窗都可有效存取。另外，任何特定目的地視窗內的縮圖都可另在邏輯順序內組織，像是在重疊順序內或根據 z 順序。

在其他範例中，DWM 可在視窗最小化之前「快照」視窗內容(即是儲存當時視窗的內容)。DWM 將視窗內容快照成例如特定大小的位元圖。大小可由重新登錄的 API 決定，並且是縮圖映射用的適當尺寸。在視窗最小化之後，視窗的內容已經保留下來，如此對應至應用程式視窗的縮圖仍舊可用。另外，已經最小化的視窗可暫時「停駐」在顯示器外，而不是真的最小化。在此方式中，應用程式可繼續傳送資訊與資料來更新縮圖。如此，若縮圖登錄發生在應用程式視窗「最小化」之後，則縮圖仍舊登錄並且更新就如同視窗未最小化一樣。另外，若縮圖登錄並不可行，則可由圖示或字幕列取代縮圖。

在步驟 604 內，決定並實施對於縮圖特性的更新。在修改或更新縮圖特性中，DWM 可據此修改全域清單。例如：DWM 應用程式可新增縮圖至其擁有的場景圖(例如第 2A 圖或第 3A 圖)，其中表示目標視窗的適當頂層視窗節點會根據額外繪製指示從來源視窗(即是 `sreWindow`)重新繪製。對於縮圖任何另改變或影響也可併入，例如縮放比例、定位、不透明度、不可見或特效。

在步驟 605 內，根據接收的指示來修改縮圖。例如：若 `fVisible` 參數為「真」，則 `UpdateThumbnail` API 可修改縮圖節點的能見度，或 `UpdateThumbnail` API 可增加

(或減少)縮圖的不透明度。在執行要對縮圖進行的所有修改之後，DWM 處理決定是否要對縮圖移除登錄。這項對於決定是否要移除登錄縮圖的處理可從 UpdateThumbnail 處理個別執行。在此情況下，可能由於 UnregisterThumbnail API 呼叫(未顯示)而發生移除登錄。在步驟 606 上，會判斷 DWM 處理是否接收到移除登錄的 API。若接收到移除登錄的 API，則處理與對應縮圖 ID 的配對就會移除(例如從維持在 DWM 處理內的縮圖登錄全域清單)。在步驟 607 上，因為縮圖已經移除登錄而重新發出頂層視窗指示串流。結果，縮圖就會移除。

在一個範例中，將游標停在工作列按鈕上會導致顯示對應應用程式的縮圖。當游標停留在工作列按鈕上，處理會建立主控縮圖的新頂層視窗代碼(HWND)。處理呼叫使用應用程式的視窗代碼(HWND)之重新登錄 API，當成縮圖內容的來源(即是 srcWindow)。主控縮圖的新頂層 HWND 為目的地視窗(dstWindow)。從來源視窗或其一部分的內容會放入目的地視窗來顯示的縮圖。縮圖可包含整個來源視窗內容或只包含上面揭示的預定或已選取部分。Update Thumbnail API 可修改或移除縮圖。例如：縮圖可相對於對應圖示而重新定位，或特效可套用至縮圖(例如透明度、不可見、扭曲等等)。當游標從工作列按鈕移開，主控縮圖的頂層 HWND 可移除登錄，如此就會隱藏縮圖。另外，縮圖可更新成不可見，如此縮

圖仍舊登錄並且後續仍舊可喚起。在另外的方法中，在建立每個視窗時可用工作列指派一建立的縮圖。在此範例中，例如工作列程式碼可用於運用縮圖能見度以及目的地矩形來放置縮圖。

應瞭解本發明態樣可有許多形式以及實施例。本文之具體實施例僅用於說明而非限制本發明，應瞭解在不悖離本發明範疇精神之下可進行改變。雖然已經顯示並說明本發明的圖解具體實施例，前揭說明意欲涵蓋其廣泛的修改、改變以及替換之範圍，並且在某些實施例中，可使用本發明的某些特色而無需對應使用其他特色。因此，應瞭解後附之申請專利範圍應做廣意之解讀，並且與本發明範疇一致。

【圖式簡單說明】

第 1A 圖說明用於實施本發明的系統範例，其包含形式上為電腦的一般用途計算裝置。

第 1B 圖至第 1M 圖顯示支援本發明一或多態樣的一般用途電腦環境。

第 2A 圖為說明顯示器組件範例的場景圖形或顯示樹狀圖。

第 2B 圖圖示根據第 2A 圖的樣本顯示器組件。

第 3A 圖為說明本發明顯示器組件範例的場景圖形或顯示樹狀圖。

第 3B 圖圖示根據第 3A 圖的顯示器。

第 4A 圖和第 4B 圖圖示其中提供應用程式視窗縮圖的一個本發明應用程式範例。

第 5 圖說明其中縮圖顯示在 ALT-TAB 功能表內的其他本發明縮圖應用程式。

第 6 圖為說明提供縮圖的處理範例流程圖。

【主要元件符號說明】

20 電腦	147 程式資料
100 計算環境	150 介面
110 電腦	151 磁碟機
120 處理單元	152 可移除磁碟
121 系統匯流排	155 光碟機
130 系統記憶體	156 可移除光碟
131 唯讀記憶體	160 使用者輸入介面
132 隨機存取記憶體	161 指標裝置
133 基本輸入/輸出系統	162 鍵盤
134 作業系統	170 配接器
135 應用程式	171 區域網路
136 程式模組	172 數據機
137 程式資料	173 廣域網路
140 硬碟機	180 遠端電腦
141 硬碟機	181 記憶體儲存裝置
144 作業系統	185 遠端應用程式

145 應用程式	190 像介面
146 程式模組	191 監視器
196 印表機	303 接龍
197 喇叭	304 縮圖宿主 HWND
200 元件	305 視窗邊框
201 桌面桌布	306 視窗邊框
202 小算盤	307 工作列
203 接龍	308 視窗內容
204 視窗邊框	309 視窗內容
205 視窗邊框	310 視窗內容
206 視窗內容	311 「小算盤」縮圖
207 視窗內容	320 桌面桌布
208 工作列	321 「接龍」應用程式視窗
209 視窗內容	322 「小算盤」應用程式視窗
220 桌面桌布	323 動態縮圖
221 「接龍」應用程式視窗	324 工作列按鈕
222 「小算盤」應用程式視窗	325 作列按鈕
223 工作列按鈕	326 工作列
224 工作列按鈕	400 工作列
225 工作列	401 「接龍」縮圖
300 元件	402 工作列按鈕

- 301 桌面桌布
- 302 小算盤
- 405 工作列按鈕
- 406 工作列按鈕
- 410 工作列
- 411 「接龍」工作列按鈕
- 412 文書處理工作列按鈕
- 413 媒體播放程式工作列按鈕
- 414 小算盤工作列按鈕
- 415 「小算盤」縮圖
- 500 螢幕
- 403 工作列按鈕
- 404 「媒體播放程式」縮圖
- 501 「接龍」應用程式視窗
- 502 「小算盤」應用程式視窗
- 503 「小算盤」工作列按鈕
- 504 「接龍」工作列按鈕
- 505 ALT-TAB 功能表
- 506 工作列按鈕
- 507 「小算盤」縮圖
- 508 「媒體播放程式」縮圖
- 509 「接龍」縮圖

七、申請專利範圍：

1. 一種儲存電腦可用指令之電腦儲存媒體，當該等電腦可用指令經一電腦裝置使用時，該等電腦可用指令導致該電腦裝置執行一方法，該方法包含以下步驟：

顯示步驟，顯示具有複數個工作列按鈕之一工作列，每一工作列按鈕對應至一啟用的應用程式視窗；

接收步驟，接收指示一使用者之輸入，該使用者定位一游標於該工作列中之一第一工作列按鈕上，該第一工作列按鈕對應於一第一啟用的應用程式視窗；

顯示步驟，在經接收該輸入後，即顯示一動態縮圖，該動態縮圖包含對應該第一啟用的應用程式視窗之內容；及

修改步驟，根據針對該第一啟用的應用程式視窗之該內容之修改，來動態修改該動態縮圖之該內容。

2. 如申請專利範圍第 1 項所述之電腦儲存媒體，其中隨著該第一啟用的應用程式視窗之該內容經修改，該動態縮圖之該內容經即時修改。
3. 如申請專利範圍第 1 項所述之電腦儲存媒體，其中該第一啟用的應用程式視窗包含一媒體播放視窗，其中該媒體播放視窗之該內容包含一經播放之視訊，且其中該動態縮圖之該內容對應經播放於該媒體播放視窗中之該視訊。
4. 如申請專利範圍第 3 項所述之電腦儲存媒體，其中該方法進一步包含以下步驟：

最小化步驟，最小化該媒體播放視窗，其中該媒體播放視窗具有一

關聯的媒體播放應用程式，該媒體播放應用程式繼續播放該視訊；以及

當該媒體播放視窗經最小化時，隨著該視訊繼續播放於該媒體播放應用程式中，會繼續顯示在該動態縮圖中之該視訊。

5. 如申請專利範圍第 1 項所述之電腦儲存媒體，其中該方法進一步包含以下步驟：

接收步驟，接收針對該動態縮圖之使用者輸入；及

修改步驟，根據針對該動態縮圖之該使用者輸入，修改該第一啟用的應用程式視窗。

6. 如申請專利範圍第 5 項所述之電腦儲存媒體，其中接收針對該動態縮圖之使用者輸入之步驟包含以下步驟：

接收步驟，接收針對該動態縮圖之該內容之使用者修改；以及

其中，修改該第一啟用的應用程式視窗之步驟包含以下步驟：

修改步驟，根據針對該動態縮圖之該內容之該使用者修改，來修改該第一啟用的應用程式視窗之該內容。

7. 如申請專利範圍第 5 項所述之電腦儲存媒體，其中該第一啟用的應用程式視窗包含一小算盤(calculator)應用程式視窗，且該動態縮圖之該內容包含一對應於該小算盤應用程式視窗之一小算盤，其中，接收針對該動態縮圖之使用者輸入之步驟包含以下步驟：

接收步驟，接收針對該動態縮圖之使用者輸入，該使用者輸入導致一計算結果經呈現於該動態縮圖中，且其中，

根據針對該動態縮圖之該使用者輸入而修改該第一啟用的應用

程式視窗之步驟包含以下步驟：

修改步驟，修改該第一啟用的應用程式視窗之該內容以顯示該計算結果。

8. 如申請專利範圍第 1 項所述之電腦儲存媒體，其中該方法進一步包含以下步驟：

接收步驟，接收用以最小化該動態縮圖之使用者輸入；及

最小化步驟，在經接收用以最小化該動態縮圖之該使用者輸入後，即最小化該動態縮圖。

9. 如申請專利範圍第 1 項所述之電腦儲存媒體，其中該動態縮圖係顯示於鄰近該第一工作列按鈕處。

10. 如申請專利範圍第 1 項所述之電腦儲存媒體，其中該方法進一步包含以下步驟：

接收步驟，接收指示該使用者之另一輸入，該使用者定位該游標於該工作列中之一第二工作列按鈕上，該第二工作列按鈕對應於一第二啟用的應用程式視窗；

顯示步驟，經接收該輸入後即顯示一第二動態縮圖，該第二動態縮圖包含對應該第二啟用的應用程式視窗內容之內容；及

修改步驟，根據針對該第二啟用的應用程式視窗內容之修改，動態修改該第二動態縮圖之該內容。

11. 如申請專利範圍第 10 項所述之電腦儲存媒體，其中該動態縮圖及該第二動態縮圖係同時地顯示。

12. 一種由一電腦裝置實施之方法，該電腦裝置包含一處理器與一或更多電

腦儲存媒體，該方法包含以下步驟：

顯示步驟，顯示對應於一啟用的應用程式視窗之一動態縮圖，該動態縮圖包含對應於該啟用的應用程式視窗之內容；

修改步驟，根據針對該啟用的應用程式視窗之該內容之修改，而修改該動態縮圖之該內容；

接收步驟，接收針對該動態縮圖之使用者輸入；及

修改步驟，根據針對該動態縮圖之該使用者輸入，修改該啟用的應用程式視窗。

13. 如申請專利範圍第 12 項所述之方法，其中接收針對該動態縮圖之使用者輸入之步驟包含以下步驟：

接收步驟，接收針對該動態縮圖之該內容之使用者修改；以及
其中，修改該啟用的應用程式視窗之步驟包含以下步驟：

修改步驟，根據針對該動態縮圖之該內容之該使用者修改，而修改該啟用的應用程式視窗之該內容。

14. 如申請專利範圍第 12 項所述之方法，其中該啟用的應用程式視窗包含一小算盤應用程式視窗，且該動態縮圖之該內容包含一對應於該小算盤應用程式視窗之一小算盤，其中，

接收針對該動態縮圖之使用者輸入之步驟包含以下步驟：

接收步驟，接收針對該動態縮圖之使用者輸入，該使用者輸入導致一計算結果經呈現於該動態縮圖中，以及其中，

根據針對該動態縮圖之該使用者輸入而修改該啟用的應用程式視窗之步驟包含以下步驟：

修改步驟，修改該啟用的應用程式視窗之該內容以顯示該計算結果。

15. 如申請專利範圍第 12 項所述之方法，其中該動態縮圖經顯示以回應接收指示一使用者之輸入，該使用者點擊一鍵盤上之一按鍵組合。
16. 如申請專利範圍第 15 項所述之方法，其中對應於複數個額外啟用的應用程式視窗之複數個額外動態縮圖亦經顯示，以回應接收指示該使用者之輸入，該使用者點擊該按鍵組合。
17. 如申請專利範圍第 12 項所述之方法，其中該動態縮圖經顯示以回應接收指示一使用者之輸入，該使用者定位一游標於一工作列中之一工作列按鈕上，該工作列按鈕對應於該啟用的應用程式視窗。
18. 一種儲存電腦可用指令之電腦儲存媒體，當該等電腦可用指令經一電腦裝置使用時，該等電腦可用指令導致該電腦裝置執行一方法，該方法包含以下步驟：

顯示步驟，顯示具有複數個工作列按鈕之一工作列，每一工作列按鈕對應至一啟用的應用程式視窗；

接收步驟，接收指示一使用者之輸入，該使用者點擊一鍵盤上之一按鍵組合；

顯示步驟，在經接收該輸入後即顯示複數個動態縮圖，每一動態縮圖包含對應一啟用的應用程式視窗內容之內容；及

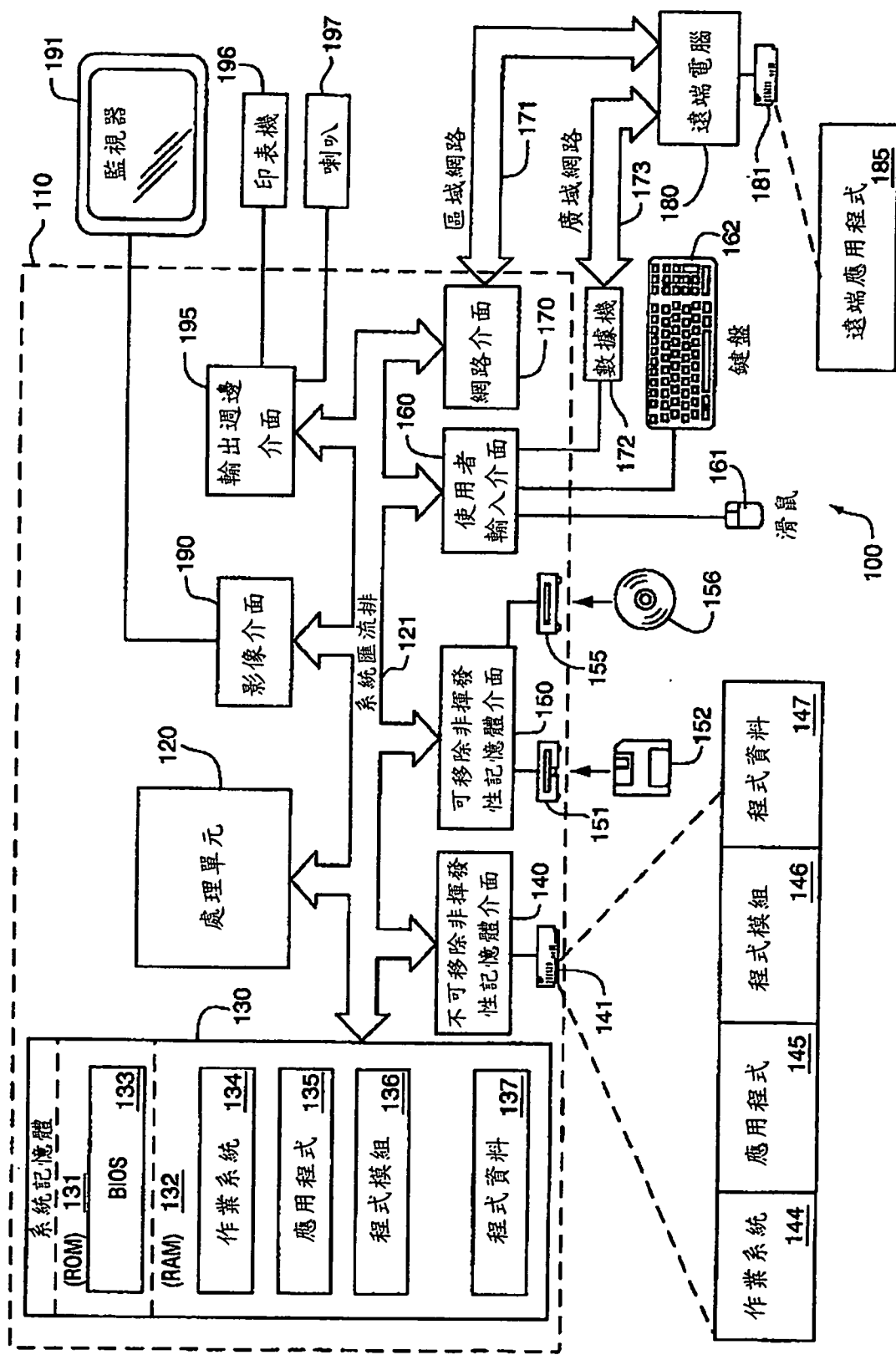
修改步驟，根據針對一對應啟用的應用程式視窗之該內容之修正，動態修改至少一動態縮圖之該內容。

19. 如申請專利範圍第 18 項所述之電腦儲存媒體，其中根據與每一動態縮圖相關聯之一初始 z 順序而初始化定位該等複數個動態縮圖，且其中該

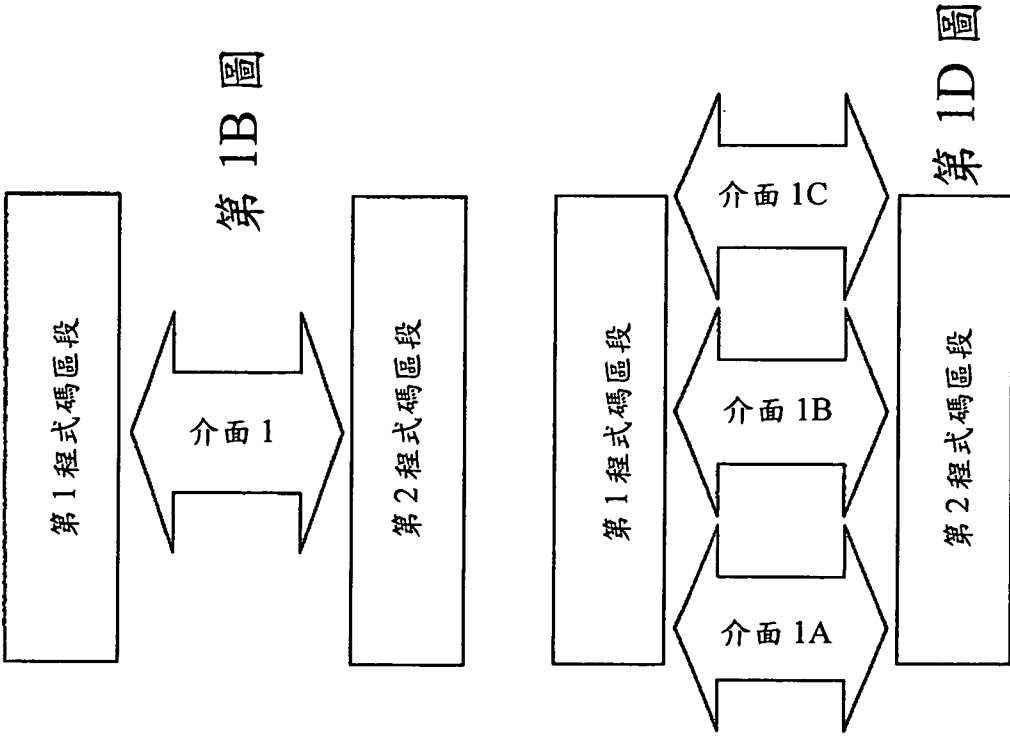
方法進一步包含以下步驟：

改變步驟，改變與該等複數個動態縮圖相關聯之該 z 順序。

20. 如申請專利範圍第 18 項所述之電腦儲存媒體，其中該等複數個縮圖為串接。

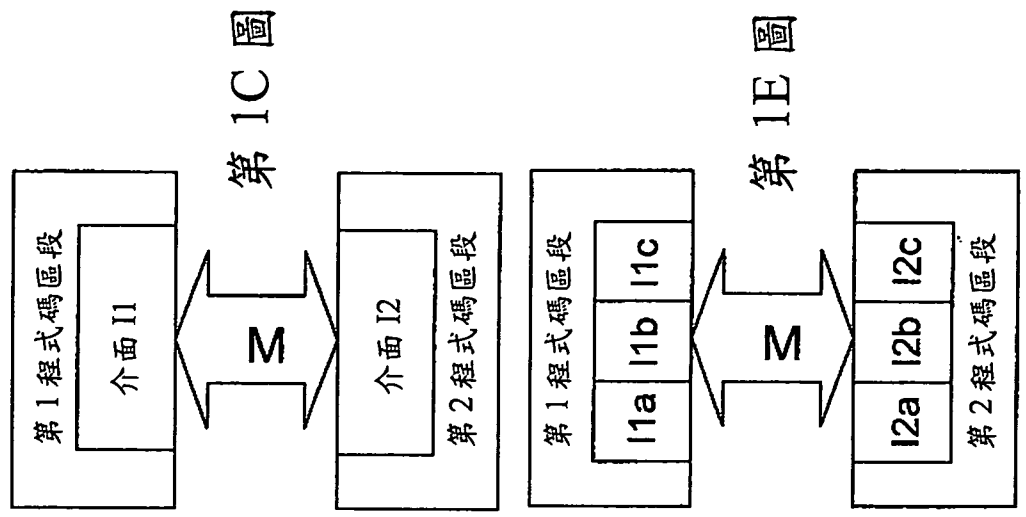


第 1A 圖



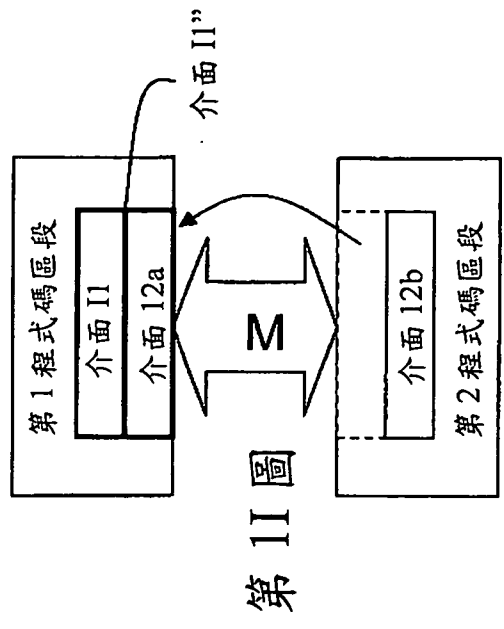
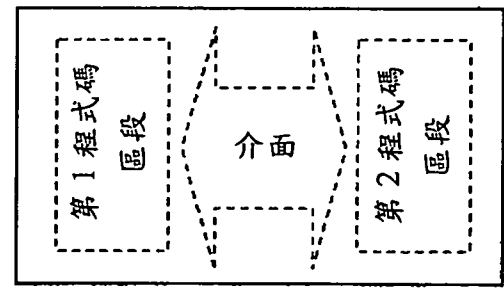
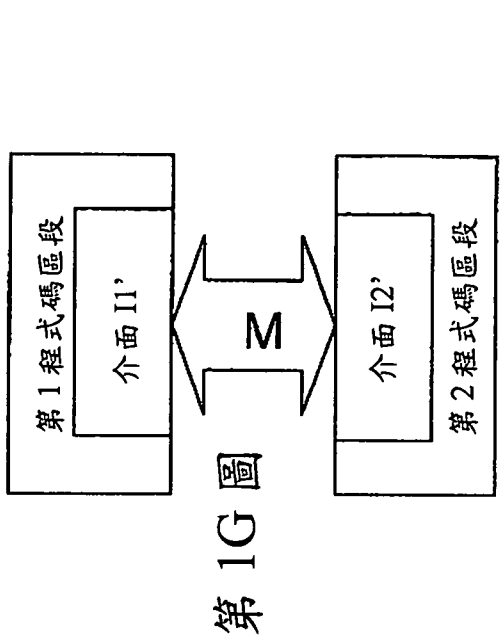
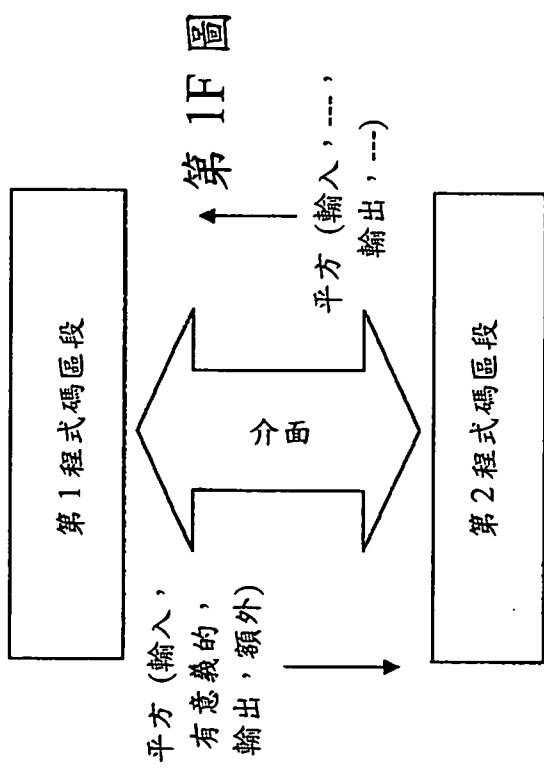
第 1B 圖

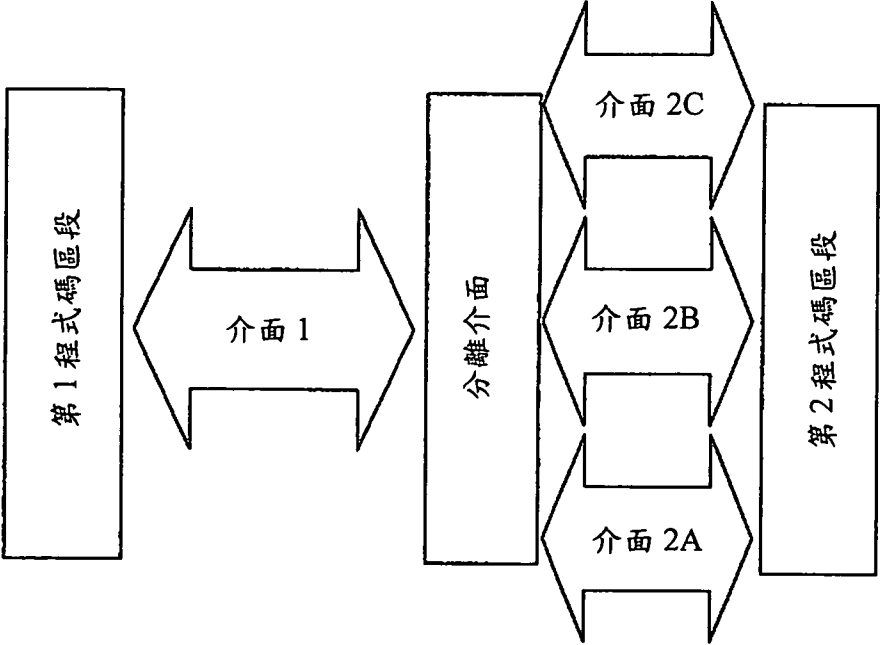
第 1D 圖



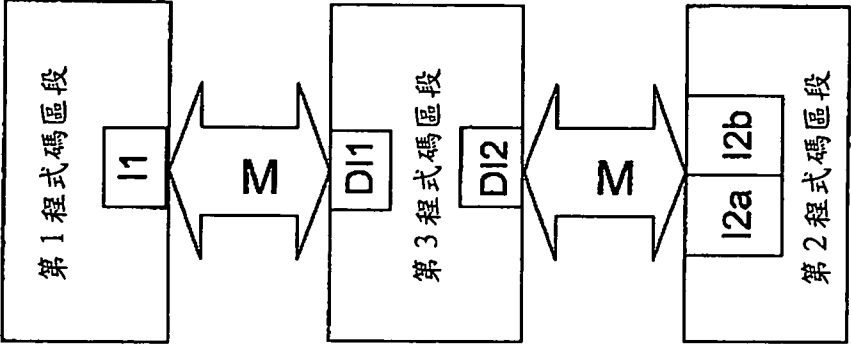
第 1C 圖

第 1E 圖

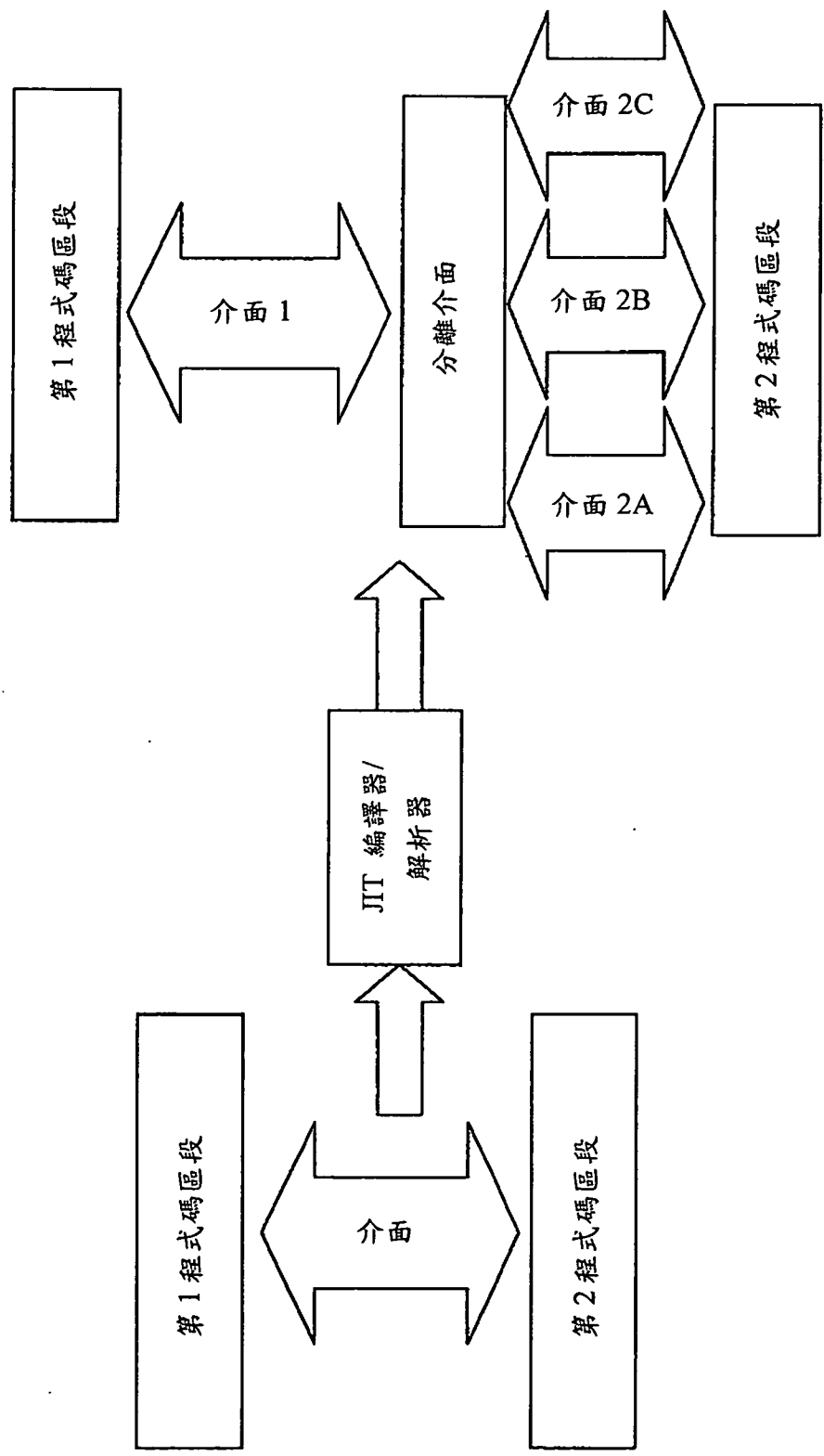




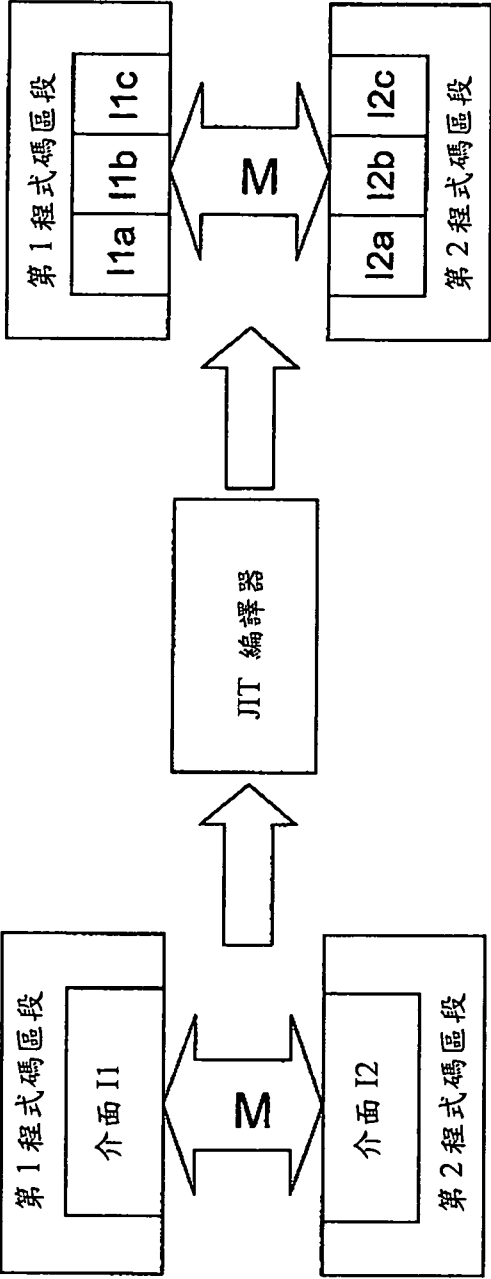
第 1J 圖



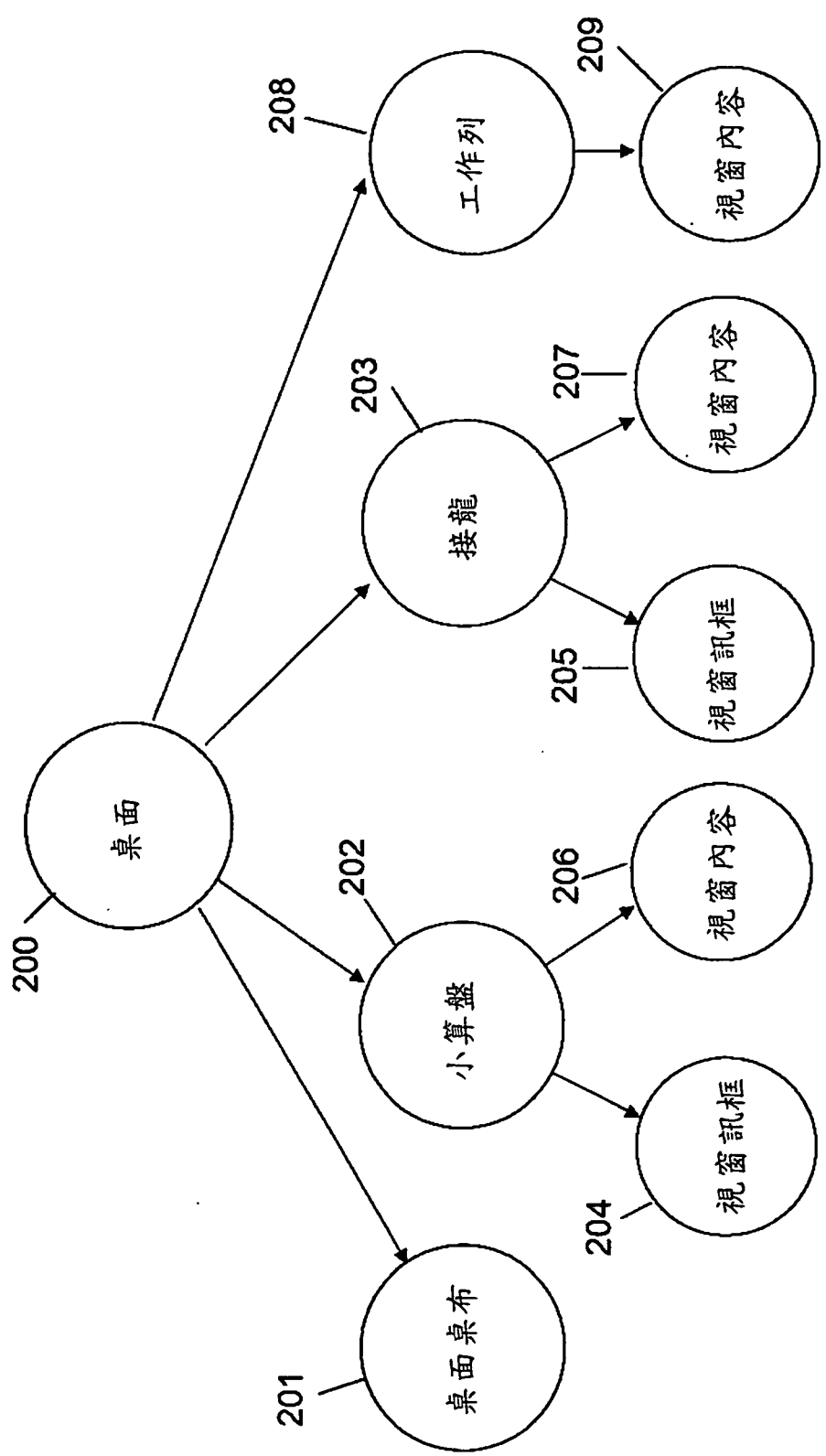
第 1K 圖



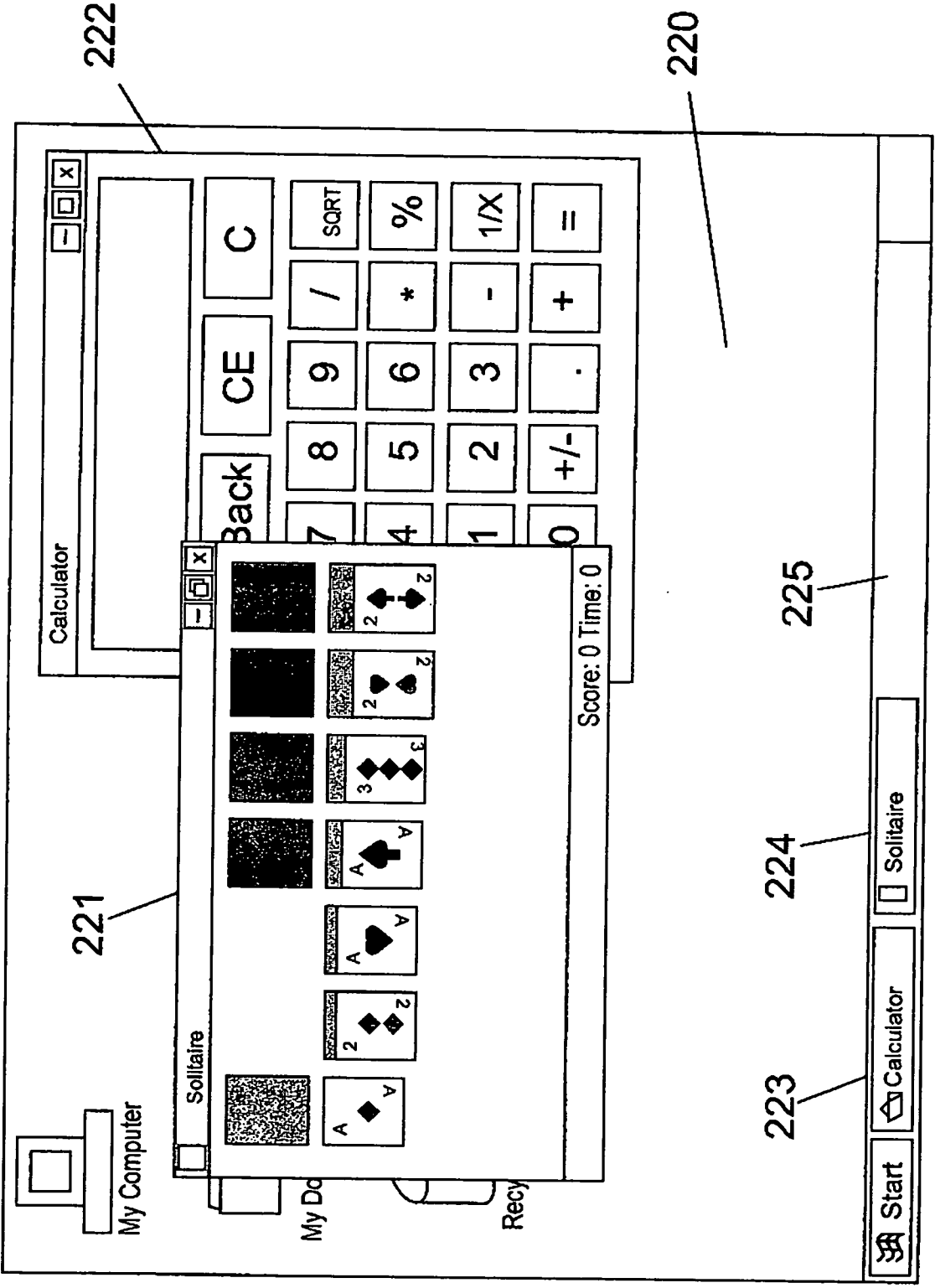
第 1L 圖



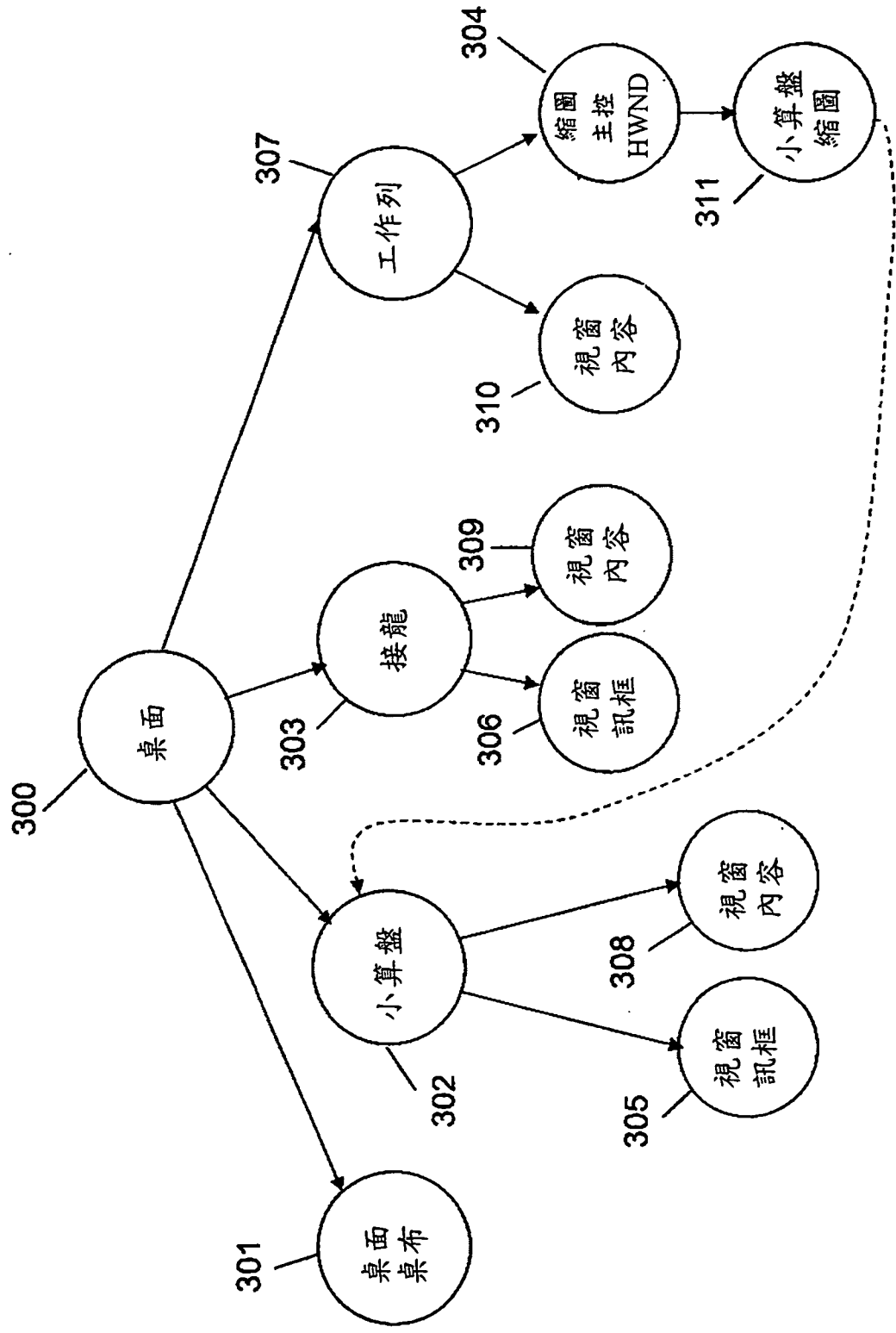
第1M圖



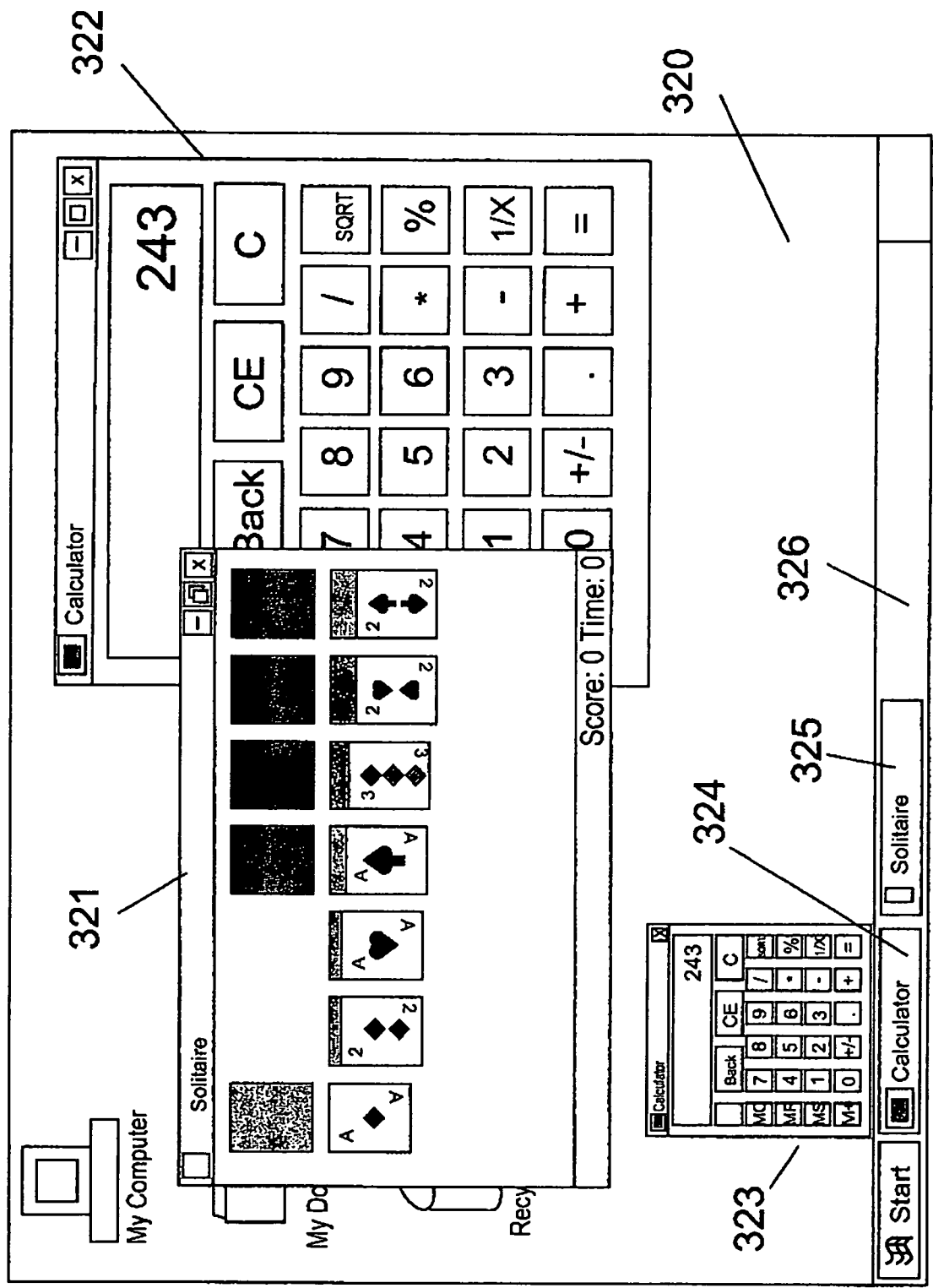
第 2A 圖



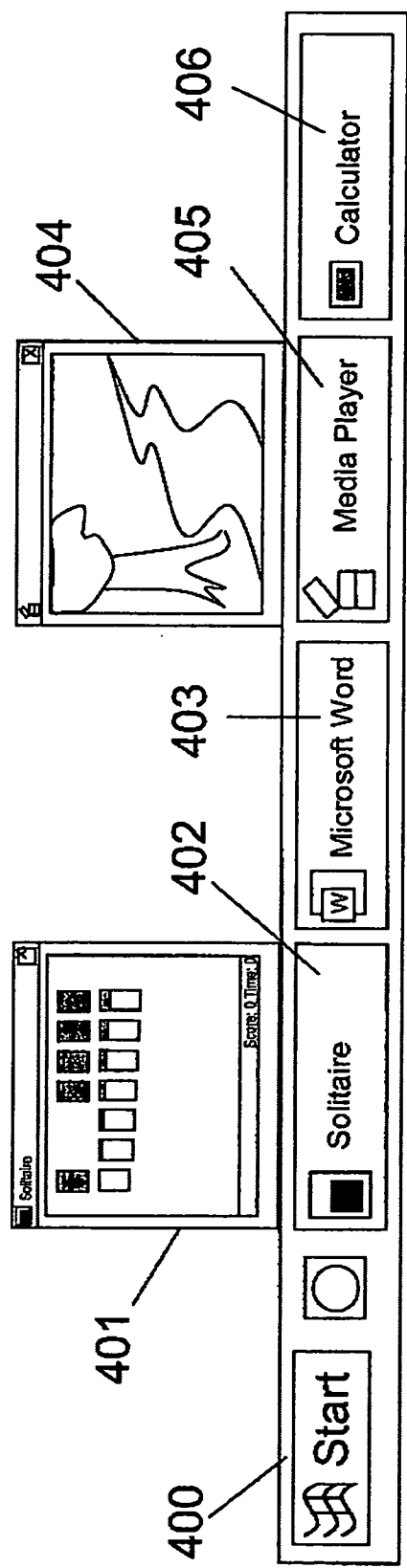
第 2B 圖



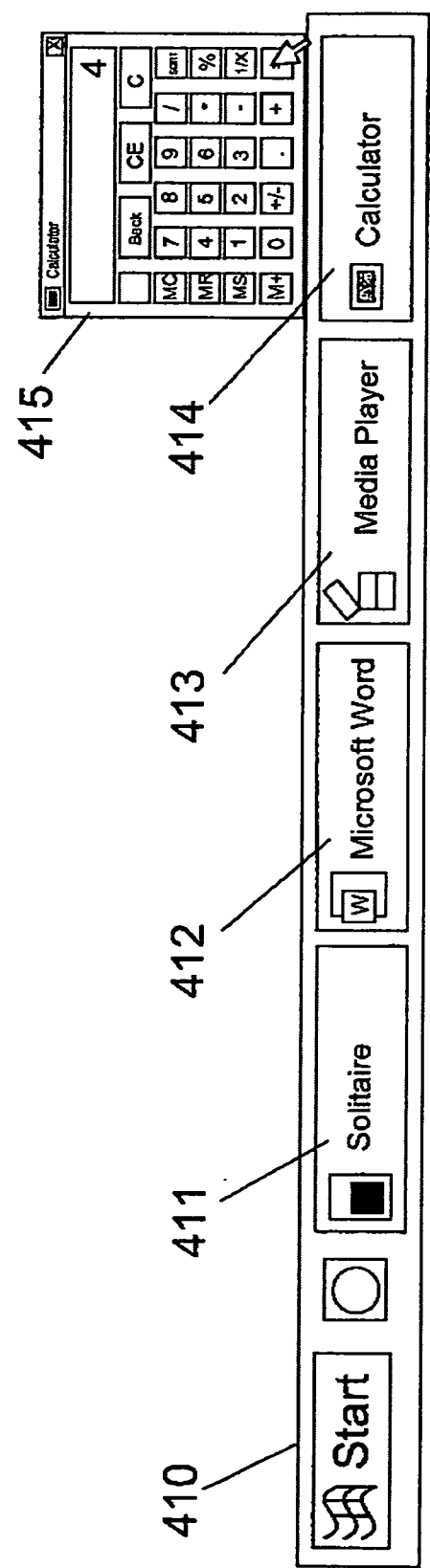
第 3A 圖



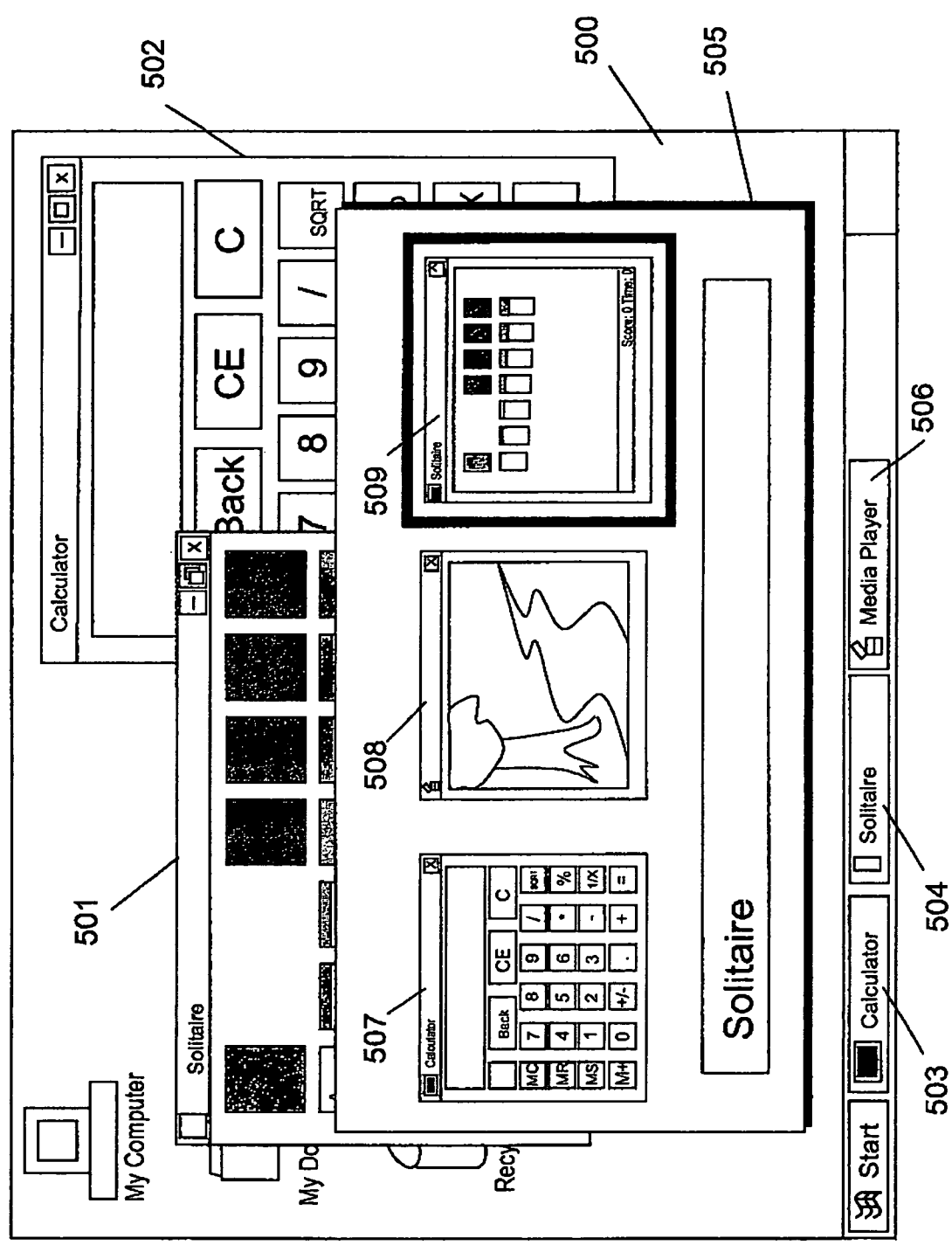
第 3B 圖



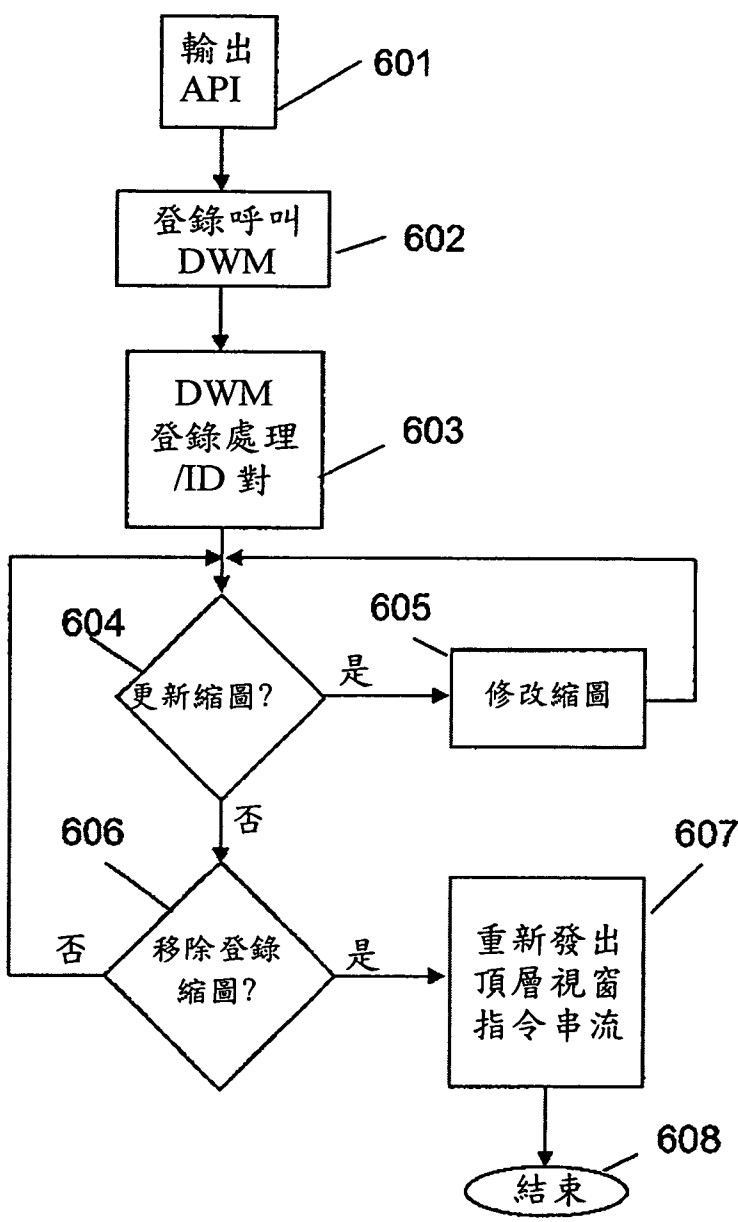
第 4A 圖



第 4B 圖



第 5 圖



第 6 圖