

Core Trees: 大型階層式資訊結構之視覺化技術

Core Trees: An Approach for Visualizing Large Hierarchical Information Structures

楊正仁 高君豪

元智大學資訊工程學系

桃園縣中壢市 中華民國 R.O.C.

(csczyang cs.yzu.edu.tw; s871786@mail86.yzu.edu.tw)

http://syslab.cse.yzu.edu.tw

摘要

面對急速成長的資訊量，如何有效率的管理及存取大量的資訊，對現代的資訊管理領域而言是一個相當重要的問題。資訊視覺化(information visualization)的技術正是其中一個重要的研究課題。透過視覺化呈現，使用者能夠具體的看到各項資訊之間的關聯性，因此能更有效率地管理這些大量的資訊。以往已有許多研究討論這方面的議題。在大部份的研究中，雖然能達到將大型階層式資訊結構視覺化呈現的目的，然而幾乎無法避免視覺混亂(visual clutter)的問題。本篇論文中提出了一個新的資訊視覺化的技術，稱之 core trees。它有三項主要的優點：(1)經由平均作圖演算法，大型的階層式架構資訊可以清楚地呈現在2D的平面上，visual clutter的問題可以大為減輕。(2)由於 core trees 視覺化的演算法中避免複雜的座標轉換，使得視覺化的過程更有效率，適合加以應用在實際的軟體系統上。(3)利用新的固定焦點技術，使得使用者能夠更專注於有興趣的相關資訊。在本篇論文中，並舉例來說明如何 core tree 將網頁之間的關係具體呈現出來。從這個例子中將可看出 core tree 以上的優點。由於現實世界之資訊中仍存有許多大型階層式的關係，相信此技術將可進一步應用在許多領域上。

關鍵詞: Hierarchical Information Structures, Core Trees, Information Visualization, Visual Clutter Problem, Focus Changing.

1. 緒論

由於資訊技術急速地發展，現今每個人每天所需讀取或存取的資料量以遠遠地超過跟十年前所需讀取或存取的資料量。而每日新產生的資訊量，更遠超過以往。就以電子期刊的數量來看，在1991年為110種，而1997年時已經達到1465種[10]。如此一來，如何在浩瀚的資訊網路中搜尋出重要的資料，便成為現代資訊管理領域的一個重要課題。隨著 World Wide Web 的發展，許多著名的搜尋引擎，如 Yahoo、Alta Vista 和 Lycos 紛紛建立以幫助人們在千萬個網站中找尋所需的資訊。然而雖然在搜尋技術上有所進展，但在資訊結果呈現的視覺效果上卻進展有限。因為大部份的搜尋結果僅是在一個平面的網頁條列出各個結果，卻不再對每項結果顯示出更進一步相關的資訊。然而實際上使用者往往更需要的卻是在螢幕上顯示出每項資訊的階層相關性，使得可以更深入瞭解並找出所需要的資訊以及相關的資訊，以便能有效率的管理及存取。

為了將資訊物間之間的關係清楚的呈現，因此資訊

視覺化(information visualization)的技術成為資訊管理領域中的一個重要研究方向。而在眾多的資訊關係類型中，階層式資訊結構尤其佔有重要的份量。這是因為在階層式資訊結構不僅常常出現在實際生活例子中，而且當在階層架構中瀏覽時，階層頂點提供了瀏覽的初始焦點並展開完整的 top-down 關係。目前在此方向中最需要解決的，是如何視覺化大型階層式資訊結構。這是因為在大型資訊結構中，常存在著上千個以上的物件，因此需要透過視覺化呈現，使得使用者能夠更容易地在眾多的物件中，找尋所需的資訊，具體的看到各項資訊之間的關聯性，並且可以輕易地瀏覽整體資訊結構，而能更有效率地管理這些大量的資訊，減輕花在掌握整體架構上的負擔。

目前已有許多針對大型階層式資訊視覺化方法相關研究[2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13]，這些研究可以分為兩類：二維空間視覺化技術(2D visualization techniques) [2, 3, 4, 5, 12] 和三維空間視覺化技術(3D visualization techniques) [1, 2, 3, 4, 6, 7, 8, 9, 11, 13]。雖然3D 視覺化技術的視覺效果較符合人類視覺上的需求，但是因為3D空間必須投影到2D的平面才能顯現在螢幕

上，而在許多之前的研究中[2, 3, 5, 6, 8]，投影轉換時卻並未考慮到視覺混亂(visual clutter)的問題，使得在投影之後，某些物件或者重疊，或者相距很近以至於產生視覺上的混淆，甚至當改變遠近以得到清晰的視覺效果時，某些物件可能已經超出螢幕之外而不可見。這些 visual clutter 的問題使得三維空間視覺化技術仍離理想有一段距離。雖然虛擬實境(virtual reality)的技術可以解決部份 3D visual clutter 的問題，但是目前虛擬實境的設備仍然是相當昂貴，無法普遍推廣應用。

本篇論文中以二維空間視覺化技術為基礎提出了一個新的資訊視覺化的技術，稱之為 *core trees*。它有三項主要的優點：

1. 大型的階層式架構資訊可以清楚地呈現在平面上，visual clutter 的問題可以大為減輕
2. 由於 *core tree* 視覺化的演算法中避免複雜的座標轉換，使得視覺化的過程更有效率，適合應用在實際的軟體系統上。
3. 利用固定焦點 (fixed focus) 的技術，使得使用者在轉換焦點後，仍能清楚掌握相關資訊架構，因而能夠更專注於有興趣的相關資訊，不會被其他無關的資訊干擾。

因此透過 *core trees* 的方式呈現，使用者將能夠清楚的瞭解資訊間的階層關係。而由於焦點物件將一直保持在中心位置，使用者將更能夠掌握整體資訊結構。

論文其他的部份包括：第二節是相關研究工作的介紹。並對一些大型階層式資訊結構視覺化方法的優缺點作一探討比較。第三節中則介紹 *core trees* 視覺化演算法。第四節是一個網頁關係視覺化的範例。將透過此範例來說明 *core trees* 的運作以及它的優點。最後第五節則是未來發展方向的討論和結論。

2. 相關研究

資訊視覺化在人機介面互動 (computer-human interaction) 領域中是一個重要研究課題。Xerox 公司所提出的 *cone trees* [11] 是 3D 視覺化技術中最著名的一種。在 *cone tree* 中，每一項資訊以物件的方式表示，所有的資訊物件依照階層關係被放置在由許多圓錐所構成的三維空間裡。每一個圓錐的頂點是圓錐底部節點的父節點，所有的子節點則放置在圓錐底的圓盤的圓周上。透過圓錐所構成的三維空間，每個資訊物件的階層關係都可以標示出來。如果由頂點鳥瞰此空間，並可得到一個 2D *cone tree* 的投影。在 WebQuery [1] 中，就曾利用 2D *cone trees* 來顯示經過搜尋引擎找尋相關資訊過後的結果。

雖然 *cone tree* 可以完整地表現出資訊物件間的階層關係，但是無法避免 visual clutter 的問題[6]。此外根據 Lamping 等人[6]指出，*cone tree* 在做改變焦點時需要 3D 動畫來旋轉整個樹狀結構，成本很高。此外對於 2D *cone tree*，visual clutter 的問題仍然無法避免。透過觀察，資訊物件密集在樹狀階層外圍的部分，因此第二層以下的物件很容易就混淆在一起。

另外在 WebQuery [1] 中並利用 bulleye 的視覺化方法來呈現網頁查詢結果。這是一種 2D 的視覺化方法。在 2D 的平面上，以平面中心為起點，每個具有相同關係程度的資訊物件則放置在距離中心等距的同心圓上。因此不同半徑的同心圓也就表示不同的關係程度。

碎形樹 (fractal tree) 是由 Koike 和 Yoshihara [4, 5] 所提出。這是一種對 *cone tree* 的改良。碎形樹利用了 fractal 演算法，使得顯示出來的節點數大量增加，但是 fractal view 卻較為固定。然而 *cone tree* 所遭遇的 visual clutter 的問題仍然存在。此外在焦點改變後，*tree* 的某些部分則可能會因為超出螢幕的範圍而看不見，反而讓使用者不易掌握資訊之相關性。

Jeong 和 Pang [3] 所提出的 reconfigurable disc tree (RDT) 是另一種 *cone tree* 的改良。RDT 設計的重點便是在於減輕視覺混亂的問題。主要就是在 RDT 中加上了許多可調整的參數，因此可以適當的調整 RDT 的形狀，使得空間利用具有很大的彈性。因此根據 RDT 的演算法，重疊的節點數量將大為減少，使資訊可以較清楚地呈現。然而 RDT 只解決一部份問題。對於分佈在圓錐底部外層的節點而言，密度仍然十分的高，因此並未解決 *cone tree* 根本的問題。

Lamping 等人 [6] 所發展的 hyperbolic browse 是一種 3D 的視覺化方法，它是將階層式的資訊放置在曲面 (hyperbolic plane) 上，並且將此曲面投影到一個 2D 的平面上。如此一來便可以顯示出以指數成長的階層結構。而在 2D 圓形投影中，由於有魚眼式景觀 (fish-eye view) 的效果，使用者可以清楚地看到被放大的中心部分。另外瀏覽器也提供了改變焦點的功能，將使用者希望進一步瞭解的資訊，移到螢幕中心的位置。然而由於 hyperbolic browse 中並未考慮到圖形的均衡，在移動焦點之後，整個圖形可能會歪斜於一邊，反而使視覺化的圖形不易辨認。

H3 是由 Munzner 等人所提出，也是一種 3D 視覺化方法[8, 9]。而 H3 和 Lamping 等人方法的差異則是在 H3 的放置的方法為將階層式的資訊結構置於一個 3D hyperbolic space 當中而不是一個 hyperbolic plane 上。經過投影以 2D 的方式呈現。雖然 H3 充分利用到空間的擺設，但是當 hyperbolic 座標系統投射到 Euclidean 座標的時候仍然無法避免 visual clutter 的問題。

相較於以上的視覺化方法，我們所提出的 *core trees* 方法跟 hyperbolic browser 以及 WebQuery 中的 bulleye 方法較為相似。但和 hyperbolic browser 不同處為處理使用者改變焦點時的整個視覺化過程。在 hyperbolic browser 的方法中，當焦點被改變時階層結構圖形並不會重新調整，整個畫面很容易會歪斜到某一邊，但是在 *core trees* 的方法中，改變焦點後則會對階層結構圖形重新調整，使階層仍保留為平均放射狀的圖形。因此使用者仍然可以清楚的掌握資訊物件間的關係。和 bulleye 不同的地方在於 bulleye 並未提供聚焦或者改變焦距的功能，使用者必須自行再下達另一次的查詢，而有可能會與上一次查詢出來的圖形不同。因此使用者反而有可能看不到前一次查詢中所感興趣的資訊。另外 *cone tree* 的放置法在結構外層節點的密度很大，但是越靠近中心點的密度就變得很小，也就是說大部分節點都被放置在外圍的地

方，使得內層有許多空間沒有被利用到，所以容易造成 visual clutter 的問題。相較之下，core tree 在空間的利用上就比較平均，在平面的每個區域的密度大致上都差不多，所以不易造成 visual clutter 的狀況。

3. Core Trees

在資訊視覺化上，使用者往往希望擁有較佳的視覺效果，能夠非常清楚地瀏覽大型的資訊結構，並且在所呈現的畫面中得到更多的資訊，而不希望發生 visual clutter 的問題。為了能夠達到上述的效果，我們便提出了一個新的資訊視覺化的技術—Core trees。在設計 Core trees 時，我們主要是基於以下兩個原則：一為提供高效率的視覺化演算法，另一個則為清楚地呈現資訊結構，即使是焦點改變之後，所呈現的資訊結構亦容易分辨，不易產生 visual clutter 的問題。

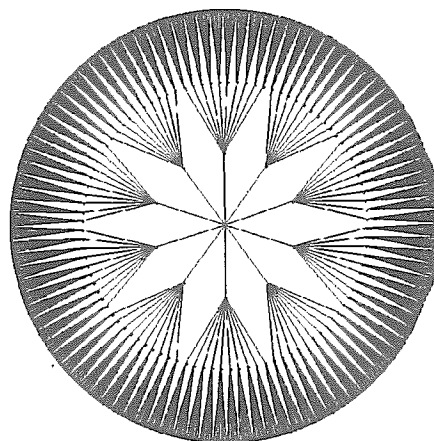
因此在視覺呈現時，我們目前並不考慮 3D 空間做圖，這是因為在螢光幕上呈像時，必須進行不同座標之投影，如此將減低視覺畫作圖的效率。同時由於我們提供改變焦點的方法中，每一點的位置均需重新計算，若是加上投影計算將嚴重影響做圖效能。

為了避免在作圖時得到嚴重的 visual clutter 視覺效果，在演算法中都将盡量避免所做的視覺圖形中有節點的重疊，線段的交錯，以及過於密集的節點等情況。我們透過將階層架構展開的方式，以得到最佳的平面性(planarity)，也就是線段與線段之間並沒有相交。並利用固定同心圓的方式，避免資訊物件節點過於集中某一區域。目前為了避免節點與節點之間重疊，我們將會計算同心圓之半徑需求，但是為了作圖效率上的考量，此半徑在視覺化過程中將保持不便，雖然如此將無可避免一些重疊得情況，然而由於之前的相關設計，卻已使得視覺化圖形 visual clutter 的效果減輕許多。

Core trees 的方法和 WebQuery 中 bulleye 方法的樣子類似，兩者都是將資訊放置在許多同心圓的圓周上，然而最大不同除了前面所述之外，bulleye 的方法僅將有相同等級的節點放在相同半徑的同心圓上，且圓心的位置放置的也不是階層結構的頂點。另外 core tree 的 2D 呈現方法與 Lamping 等人所提出的 Hyperbolic browser 的 2D 呈現方法，雖然在起始時十分類似，但是在改變焦點後，整體的視覺效果就有很大的差別。在 Hyperbolic browser 中會產生傾斜的視覺化階層架構，而在 core trees 中則避免了此種狀況的產生。以下將進一步介紹 core tree 的視覺化演算法則。

3.1 階層式資訊結構視覺化

首先我們以圖形說明 core tree 的視覺化效果。圖一所展現的是一個階層結構的深度為 4 而每個節點的分支度為 10 的完整樹狀架構(complete tree structure)。在此圖中總共呈現了 1111 個節點，每一個階層皆放置在不同的同心圓上。雖然在圖一中最外層圓周上的節點是非常靠近，但是利用這種同心圓的方法卻可以清楚的標示在內部的節點。使用者可以藉著追蹤這些節點的標籤清楚的掌握資訊物件間階層架構的關係。



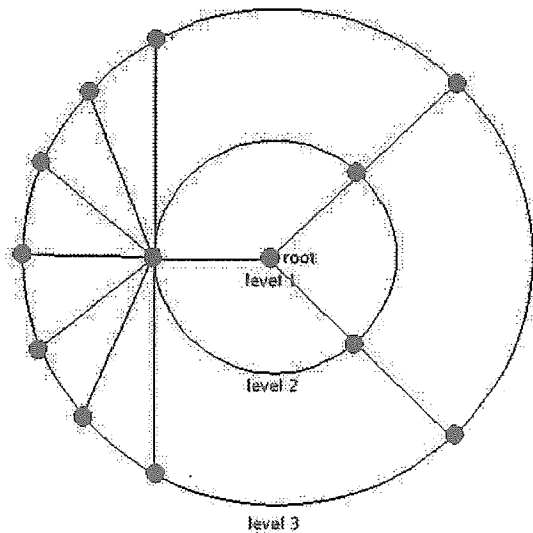
圖一：深度為 4 而每個節點的分支度為 10 的 core tree，其中共顯示出 1111 個節點。

在 core trees 方法中另外加入了聚焦(focusing)以及改變焦點(focus changing)的功能。所謂聚焦也就是使用者可以利用滑鼠點選任何一個可見的節點，而此節點將移動到圓心而成為新焦點所在。它的相關節點也會因著新焦點而根據其關係移動到其適當的位置。在此同時，整個 core tree 仍然會保持均衡的節點分布佈局，因此不論改變多少次的焦點，改變之後，所呈現的階層結構仍然保持為一個均衡放射狀的 core tree。如此使用者仍然能夠清楚的見到附近相關的資訊物件。

然而 2D 螢光幕顯示空間有限，實際上無法顯示所有資訊階層結構。為了避免 visual clutter 的問題，樹狀結構只能顯示部分的層數。因此在 core tree 中將會依照使用者的需求，顯示最相關的數層資訊物件，遠端的部分則暫時隱藏起來。然而在改變焦點時，這些暫時被隱藏的節點則會隨相關程度再次呈現出來。透過此一隱藏的步驟，使用者將不會為螢光幕上過多的節點數目所混淆。

3.2 視覺化法則

此處將更進一步介紹建構 core tree 的相關細節。以下每一個資訊物件將以「節點」代表以簡化我們的說明。假設現在使用者只希望觀察到三個階層內的相關資訊，下面的視覺化法則將依此假設說明其過程。在初始時，root 節點放置在圓心的位置，其階層數為 1(level 1)。然後將其子節點放在以 root 節點為圓心， r_1 為半徑的圓周上，也就是所謂的第三層子節點(level 3)。第二層子節點下的子節點則放在以 root 節點為圓心， r_2 為半徑的圓周上， $r_2 > r_1$ ，這些也就是第三層子節點(level 3)。為了讓同一層的節點不會發生互相覆蓋的狀況，所以各階層圓周的半徑會隨著各階層節點總數來做適當的調整。為了讓演算法簡單且有效率，系統的內定值則是將第 n 層同心圓的半徑和第 $n+1$ 層半徑的比例設為 $n:n+1$ 。若同一圓上的節點產生節點重疊結果，則將此圓半徑加以擴大但以不破壞平面性為原則。

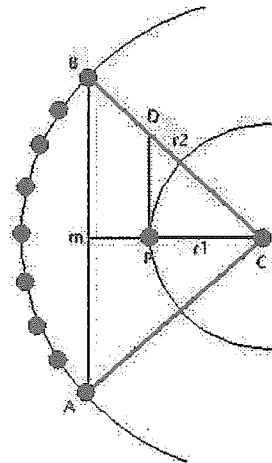


圖二：當其中一個階層為 2 的節點的子節點數大於所有子節點總數時的情形。

為了達到視覺效果上的美觀，若某第一層子節點下沒有任何的子節點，我們仍然在此節點下虛擬一個影子節點(shadow node)，雖不顯示出來，但佔有一個空間，可用來清楚區分出階層結構中的分支。

放置不同階層的節點時，先從外圍開始放起並試著平均分佈在圓周上。若是一個分佈平均的階層架構，也就是每一節點的分支數目大約相等，則視覺化後的圖形將為一個均衡的 core tree。但有時某一節點的分支數目可能過多，若平均分佈其子節點，則很可能造成分支與分支之間重疊而破壞圖形的平面要求，產生 visual clutter 的問題。此時為了視覺效果上的美觀，在 core tree 的設計中，將同時限制此節點所能展該的最大範圍為與該節點所在的圓相切的範圍。因此在 core tree 中會計算出一個最大可展開的圓周範圍。這些子節點將會經過計算放置在一個半切圓周之範圍內，如果平均分佈子節點在整個圓周上會使分佈超過其最大展開圓周，則這些子節點將以此展開圓周為範圍重新平均分佈，而其他屬於不同父節點的同層節點則平均分佈在剩餘的圓周上。例如圖二中，有三個第二層節點，其中一個具有七個第三層節點，另外兩個則均只有一個節點。此時若第三層的九個節點平均分佈在圓周上，則反而將使圖形變得複雜。而這七個第三層子節點若放置在以其父節點為切點的半切圓周上，則其圖形將可較清晰呈現各分支的關係。

Level 2 節點的放置方法如下。當外層節點位置決定後，便能向內層推展。為了使得系統的演算法則簡單且有效率，所以利用中垂線，由外層的圓周決定內層父節點所在的位置。如圖三中，第三層的節點已經都分佈好，在此利用一個擁有相同父節點的第三層子節點集合作為一個範例，首先利用第一個子節點 A 和最後一個子節點 B，計算出 \overline{AB} 線段的中點 M 的位置，如此一來就可以利用 \overline{CM} 線段和 r_1 的長度比，求出 P 點即其父節點的座標。以此類推，計算出所有第三層節點的父節點的位置。而圓心的位置則放置 root。



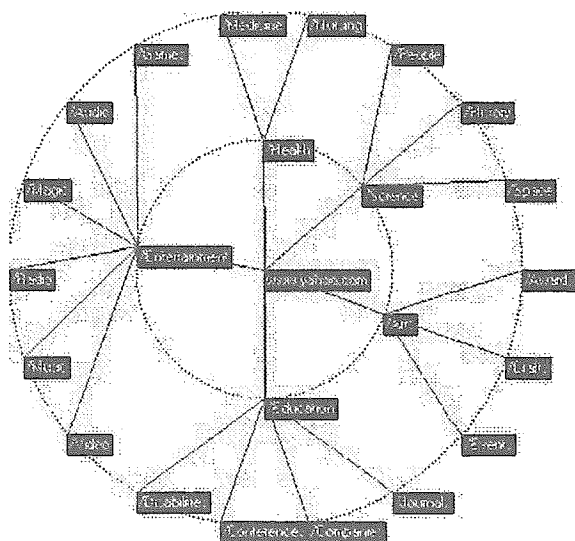
圖三：利用線段 \overline{CM} 和 r_1 的比例關係，用以求出 A 到 B 共九點的父節點即 P 點的位置。

當轉換焦點時，所有的節點會重新計算其座標位置。為了讓圖形保持一個平滑移動的視覺效果，當使用者選定其一個節點，會先紀錄起來目前點選的節點和其父節點原先的夾角，在轉換焦點時，會根據此夾角先放置其父節點的位置，然後再決定其父節點其他子節點的位置。

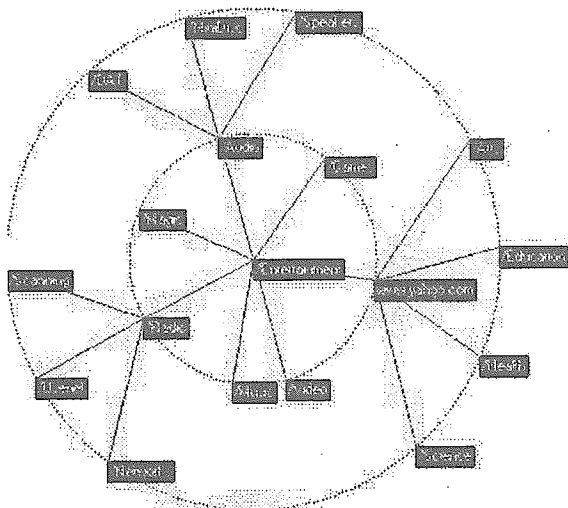
座標。接下來決定最外層節點的位置，利用前述之中垂線的方法計算出其父節點的座標。在圓心的位置放置點選的節點。

4. 視覺化範例

針對所提出的 core trees 視覺化方法，我們發展了一個離型軟體來視覺化大型的階層式資訊結構。圖四中所看到的就是一個將 yahoo 網站各網頁間的階層架構視覺化的例子。在此例子中，為了能夠清楚的解釋 core trees 的操作，只有部份 yahoo 網站的鏈結呈現出來。



圖四：利用 core tree 將 yahoo 網站上的網頁的鏈結關係表示出來。在此例中，只顯示出三層的階層架構。



圖五：改變焦點過後所呈現的視覺化效果。

在此例子中，一個節點即表示為一個網頁。在每個節點上，標籤表示其網頁的位置，每個節點可用不同的顏色代表不同的意義，例如在此例子中僅將整個階層最上端的節點的標籤顏色改成和其餘節點不同的紅色，因為如此可以幫助使用者在經過許多次瀏覽之後，可以很容易的區分目前焦點網頁與首頁之間的關係，甚至當首頁仍在顯示範圍中時，使用者能夠很容易地就能直接回到該網站的首頁。

同時在瀏覽過程中，使用者可能有需要區分上層資訊結構與下層資訊結構，為了要能清楚的追溯上層的資訊結構，另外當改變焦點過後，目前焦點網頁的父鏈結網頁的標籤也將以不同顏色（例如藍色）來區別。如此一來使用者便能夠更清楚的瀏覽整個階層式的架構。

圖五中所顯示的就是經過改變焦點後的效果。在此圖中，是用滑鼠點選了“/Entertainment”，而我們可以看到原本被暫時隱藏起來的子孫節點，在改變焦點後被顯示出來了。不過相對同時也有一些節點被隱藏起來。而整個 core tree 仍然保持一個均衡的視覺化呈現。此圖中也故意的去掉一些第三層的節點，以呈現影子節點的效果。利用此視覺化原理，使用者可以很清楚地觀察這種大型的階層式架構的資訊結構。

5. 未來工作展望與結論

本篇論文中提出一個 core trees 資訊視覺化的方法，利用數個同心圓的方式來呈現一個大型階層式資訊架構之間的關係。利用此一方法，使用者能夠清楚地觀察到這些資訊的階層式結構而不會產生 visual clutter 的問題。使用者可以更專注於有興趣的相關資訊而不會被其他無關資訊所干擾。除了本篇論文所舉的範例之外，core tree 還可以應用在資料庫、軟體工程、數位圖書館和檔案系統管理各個領域之中。只要有階層關係的資訊物件架構都可以使用 core tree 的方法來表示。

然而在發展 core tree 技術的過程中，未來仍有許多待完成得工作需要進一步研究。首先是 2D 視覺化技術仍然有其瓶頸。資訊物件顯示的數目便不如 3D 的視

覺化方法來的多。單單只是用 2D 空間的來呈現視覺化關係，仍然可以發現在整個螢幕中有許多沒有被利用到的空間。因此為了在顯示的節點數上能夠有所突破且更能夠有效的利用螢幕空間，3D 的資訊視覺化方法便成為未來需要更進一步的研究目標。

再者，在安排資訊節點時，目前仍然沒有一個恰當的視覺效果評估標準。目前大部份的視覺化相關研究都偏重於量測使用者搜尋速度上的改善，卻未考慮圖形視覺效果上的改善。目前我們在此方面雖然有初步的研究，但是仍然不進完善。因此這也是未來我們將進行的重點之一。

同時未來我們會在 core trees 改變焦點時加入動畫的效果，使得在改變焦點時，整個結構轉換的過程看起來更為平順。為了避免動畫效果會嚴重影響視覺化的效能，動畫效果與系統效能之間必須要有一個平衡點，如何在之間取捨，也是未來研究的目標。

總括來說，在呈現大型階層式資訊結構上 core trees 提供有效率且清楚的視覺化效果。它主要有三項的優點：

1. 大型的階層式架構資訊可以清楚 2D 呈現的平面上，visual clutter 的問題可以大為減輕
2. 由於 core tree 視覺化的演算法中避免複雜的座標轉換，使得視覺化的過程更有效率，適合加以應用在實際的軟體系統上。
3. 利用固定焦點 (fixed focus) 的技術，使得使用者在轉換焦點後，仍能清楚掌握相關資訊架構，因而能夠更專注於有興趣的相關資訊，不會被其他無關的資訊干擾。

由於資訊量急遽暴增，為了有效率地管理與存取大量的資訊，資訊視覺化必然是亟待解決的問題之一。而由目前的研究經驗中，相信 core trees 的研究對未來發展更高階的資訊視覺化技術必然有相當大的助益。

參考文獻

- [1] Jeromy Carrière and Rick Kazman, "WebQuery: Searching and Visualizing the Web through Connectivity," In 6th International World Wide Web Conference, pages 701-711, Santa Clara, CA, April 1997.
- [2] Jeromy Carrière and Rick Kazman, "Research Report: Interacting with Huge Hierarchies: Beyond Cone Trees," In Proceedings of the 1995 IEEE Symposium on Information Visualization, pages 77-81, Atlanta, GA, October 1995. IEEE.
- [3] Chang-Sung Jeong and Alex Pang, "Reconfigurable Disc Trees for Visualizing Large Hierarchical Information Space," In Proceedings of the 1998 IEEE Symposium on Information Visualization (InfoVis '98), pages 19-25, North Carolina, October 1998. IEEE.

- [4] Hideki Koike, "Fractal Views: A Fractal-Based Method for Controlling Information Display," *ACM Transactions on Information Systems*, 100(100):1-19, 1999.
- [5] Hideki Koike and Hiroataka Yoshihara, "Fractal Approaches for Visualizing Huge Hierarchies," In *Proceedings of the 1993 IEEE/CS Symposium on Visual Languages (VL '93)*, pages 55-60, Los Alamitos CA, 1993. IEEE.
- [6] John Lamping, Ramana Rao, and Peter Prolli, "A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies," In *Proceedings of the ACM SIGCHI'95 Conference on Human Factors in Computing Systems*, pages 401-408, Denver, CO, May 1995. ACM.
- [7] Jock D. Mackinlay, George G. Robertson, and Stuart K. Card, "The Perspective Wall: Detail and Context Smoothly Integrated," In *Proceedings of the ACM SIGCHI'91 Conference on Human Factors in Computing Systems*, pages 173 -179, New Orleans, LA, April 1991. ACM.
- [8] Tamara Munzner, "H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space," In *Proceedings of the 1997 IEEE Symposium on Information Visualization*, pages 2-10, Phoenix, AZ, October 1997. IEEE.
- [9] Tamara Munzner and Paul Burchard, "Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space," In *Proceedings of the First Annual Symposium on the VRML Modeling Language*, special issue of *Computer Graphics*, ACM SIGGRAPH, pages 33-38, San Diego, California, December 1995. ACM.
- [10] Taemin Kim Park, "The Maze of Electronic Journals in Digital Libraries," In *IT and Global Digital Library Development (Proceedings of the 11st International Conference on New Information Technology, NIT'99)*, Ching-Chih Chen ed., pages 313-322, MicroUse Information, Co., 1999.
- [11] George G. Robertson, Jock D. Mackinlay, and Stuart K. Card, "Cone Trees: Animated 3D Visualizations of Hierarchical Information," In *Proceedings of the ACM SIGCHI'91 Conference on Human Factors in Computing Systems*, pages 189 -194, New Orleans, LA, April 1991. ACM.
- [12] Manojit Sarkar and Marc H. Brown, "Graphical Fisheye Views," *Communications of the ACM*, 37(12):73-84, December 1994.
- [13] Peter Young, "Three Dimensional Information Visualisation," Technical Report 12/96, Centre for Software Maintenance, Department of Computer Science, November 1996.