社群網路之聚焦式資訊視覺化系統
Focus-Dependent Social Network Visualization

劉俊良
Chun-Liang Liu
國立臺灣大學
National Taiwan University
being31@cmlab.csie.ntu.edu.tw

李庭嫣
Ting-Yen Li
國立臺灣大學
National Taiwan University
lydian@cmlab.csie.ntu.edu.tw

陳炳宇
Bing-Yu Chen
國立臺灣大學
National Taiwan University
robin@ntu.edu.tw

摘要

在本篇論文中，我們提出以站在使用者立場來觀察社群網路（Social Network）之論點，透過我們所提供的資訊視覺化（Information Visualization）呈現概念，試著提供更具辨識度的網路佈局（layout）視覺呈現、視覺聚焦（focus-dependent）工具、分群（clustering）工具、顏色輔助辨識等機制，來減緩一直以來大規模網路在視覺呈現上所存在著視覺混淆問題，提高透過資訊視覺化方式對於大規模社群網路乃至於一般網路呈現的理解能力。

關鍵字

資訊視覺化、社群網路、聚焦、網路佈局、分群方法

1. 簡介

近年來，資訊視覺化在資訊挖掘（Information Mining）上扮演著越來越重要的角色；而在資訊視覺化領域，個體（node）與個體之間的關連（relation）網路一直是受到高度興趣的目標資料；在小規模的關連網路裡面或許沒有多大的資訊價值，但是當關聯網路的規模隨著更多的節點和關聯性的加入而增加時，關聯網路的潛在價值便隨之提升；相對地，如何從大規模且複雜的關聯網路裡頭擷取出使用者有興趣的潛在價值資訊一直以來是該課題最大的挑戰。而在這裡我們把焦點放在如何運用資訊視覺化來解決此問題。

目前大部分的資訊視覺化方法是站在整體關聯網路的角度去做資訊呈現；但是，在很多時候使用者是先對關聯網路裡面的某個節點，或者是最有利害關係的節點來試著挖掘資料；例如在社群網路裡，使用者有極大的可能是希望能夠從自己或某特定人物節點的角度來觀看整個關聯網路，並進一步藉由資訊視覺化來發現各個節點之間的關係。

在這個靈感之下，我們提出一個社群網路之聚焦式資訊視覺化系統，應用三維空間來試著提升對於複雜關聯網路的視覺呈現效果，同時導入焦點式多層次網路佈局（focus-dependent multi-layer layout）、利用景深效果之關聯挖掘（relation mining by using depth of field）、聚焦式分群方法（focus-dependent clustering）、多群組著色方法（multi-group coloring）來增加關聯網路的視覺可辨識度。

2. 焦點式多層次網路佈局

三維空間固然增加了視覺化大規模複雜關聯網路的空間，但是由於目前的顯示器仍然只能顯示二維平面，如果缺乏良好的三維空間配置很有可能導致使用者在顯示器上看到的網路佈局是缺乏辨識度的，亦或是失去關聯網路的幾何分佈概念；如何妥善利用三維空間來放置更大規模聯絡網路，同時又不失去以提供更具辨識度的網路佈局為前提，是我們在網路佈局的主要目標。在之前我們已經提到過將以某個節點的立場去做整體網路佈局需求的安排，同時我們也需要對三維空間做一個妥善的佈局規劃，結合上述兩點，我們提出在三維空間裡頭的聚焦式多層次網路佈局方法。

首先假設，在社群網路中結點 R 代表著我們所將聚焦的節點，$D(R,N_i)$ 代表著節點 R 和節點 N_i 之間需要透過多少連結才能到達，如果 $D(R,N_i)=1$ 則代表節點 R 和節點 N_i 之間存在者直接的連結關係；若 $D(R,N_i)=d$，代表節點 R 和節點 N_i 至少要間接地透過 $(d-1)$ 個其他節點才能到達。

在以聚焦節點 R 作整體社群網路佈局的前提下，我們比較在意網路裡所有的節點到達聚焦節點 R 的距離，若距離較遠則代表該節點和聚焦節點 R 的關係較遠，反之亦然；同時我們也主張在三維空間裡面，網路配置必須要依循某個特定的視野方向（viewing direction），並且根據節點到聚焦節點 R 的最短距離做為決定該節點在於主要視野方向的深遠度。

若節點 N_i 和聚焦節點 R 的距離較近，則節點 N_i 在主要視野裡和聚焦節點 R 的深度上便相對較近；反之，若有一節點 N_j 在網絡連結上和聚焦節點 R 距離較遠的話，我們便將節點 N_j 擺置在較深的位置，以減少他在主要視野上的可見度，如此一來可以減少對關聯性較低的節點複雜度所帶來的視覺性混淆，但也保留所有節點的資訊以供可能地進一步查詢。

因此，在進入佈局演算法之前，必須要先使用廣度優先（breadth-first search, BFS）演算法，以決定所有節點和聚焦節點 R 之間的距離，並將之定義為該節點的深度，因此，節點 N_i 的深度值便可定義為 $N_i.depth = D(R,N_i)$。

綜合上述，在佈局演算法上，我們改進了[4]的做法，並且加入了各個節點 N_i 和聚焦節點 R 的連結性距離資訊 $D(R,N_i)$。以下便是針對這個演算法的詳細描述。

演算法 1: Calculate all Repulse Force in Graph *G*

for all nodes N_i *in graph* G
  $d = N_i.depth$
  for all other nodes N_j *in* $Layer_{(d-1)}$, $Layer_d$, $Layer_{(d+1)}$
    $diff = N_i.position - N_j.position$
    $norm = \text{Normalize}(diff)$
    $fr = \text{ForceRepulse}(norm, m\_K)$
    $N_i.dir += diff * fr$

在演算法 1 裡，*diff* 為一 3x1 的向量紀錄了兩節點的位置差異，*norm* 為 *diff* 正規化（normalize）後的 3x1 向量；$\text{ForceRepulse}(norm, m\_K) = m\_K^2 / norm$，其中 m_K 為節點之間的預設長度。

演算法 2: Calculate all Attract Force in Graph *G*

for all edges E_i *in graph* G
  $diff = E_i.NodeFrom.position - E_i.NodeTo.position$
  $norm = \text{Normalize}(diff)$
  $fa = \text{ForceAttractive}(norm, m\_K)$
  $E_i.NodeFrom.dir -= diff * fa$
  $E_i.NodeTo.dir += diff * fa$

在演算法 2 裡，$\text{ForceAttractive}(norm, m\_K) = norm^2 / m\_K$。

演算法 3: Calculate Predefined Depth Attract Force

For all nodes N_i *in graph* G
  $diff = N_i.position.z - \text{LayerDepth}(N_i.depth)$
  $norm = \text{Normalize}(diff)$
  $fa = \text{ForceAttractive}(norm, m\_K)$
  $N_i.dir.z -= diff * fa$

在演算法 3 裡，$\text{LayerDepth}(N_i.depth) = N_i.depth * \sqrt{BB_{Area}}$。其中 BB_{Area} 定義為目前整體關聯網路之包圍盒（bounding box）投射在主要視野方向上的二維面積。最後，對所有的節點 N_i，執行 $norm = \text{Normalize}(N_i.dir)$，以更新 $N_i.position += \min(norm, m\_T) * N_i.dir$。其中 m_T 記錄著當下整體網路的活躍性。

在正常的情況下，上述演算法的執行次數是根據網路內節點數量所決定，在每次執行結束後，便降低網路活躍性 $m\_T = \text{CoolDown}(m\_T, rate) = m\_T - (m\_T / rate)$，使得每次的網路佈局演算法執行會引導至一個安定的結果。其中 *rate* 為一安定常數。

值得一提的是，傳統的重力作用佈局（force directed layout）方法在計算節點 N_i 的總受力向量時，是考慮其他所有節點對於節點 N_i 的貢獻力，因此複雜度為 $O(n^2)$；但是在我們演算法中，因為已先行決定各個節點在主要視野的深度資訊，因此在計算某深度為 d 的 $Layer_d$ 下所有節點 N_i 的綜合受力時，只需考慮 $Layer_{(d-1)}$ 和 $Layer_{(d+1)}$ 的節點對於 N_i 的貢獻度，因此複雜度可降為 $O(n \log n)$。

此外，各個節點 N_i 的深度資訊 $D(R, N_i)$ 在我們演算法裡，利用吸引力趨使某個深度為 d 的節點 N_i 趨近特定的視野深度時，用吸引力的方式來決定節點的視野深度的好處在於：如果有一群深度一樣且高度相互連結的節點群，在高度連結吸引力的牽引下仍然有在視野深度上的緩衝以避免重疊（overlapping）。

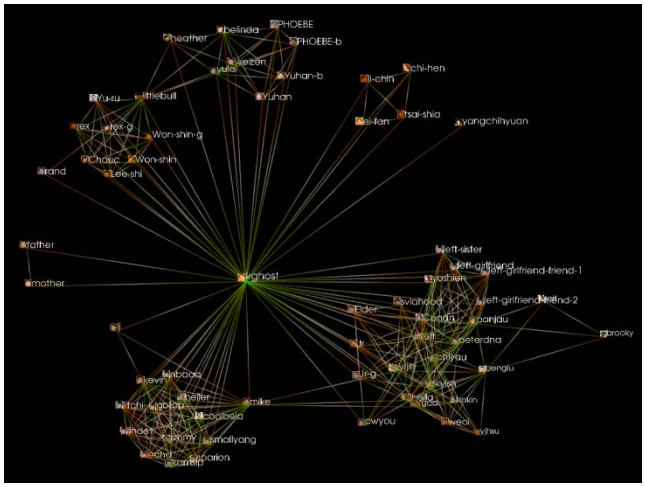

圖 1: 由主要視野方向觀看由聚焦式多層次網路佈局的結果

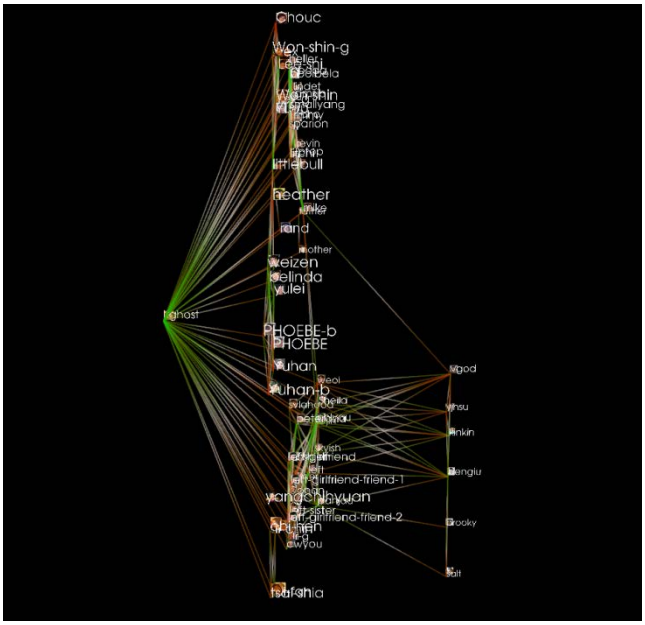

圖 2: 由側面觀看網路佈局在主要觀看視野的深度層次變化。

3. 利用景深效果之關聯挖掘

基於聚焦式多層次網路佈局的概念下，我們賦予了所有節點在主要視野方向上深度的意義，但是在主要視野上仍然存在著問題，不同深度的節點會因為投射至二維空間後而發

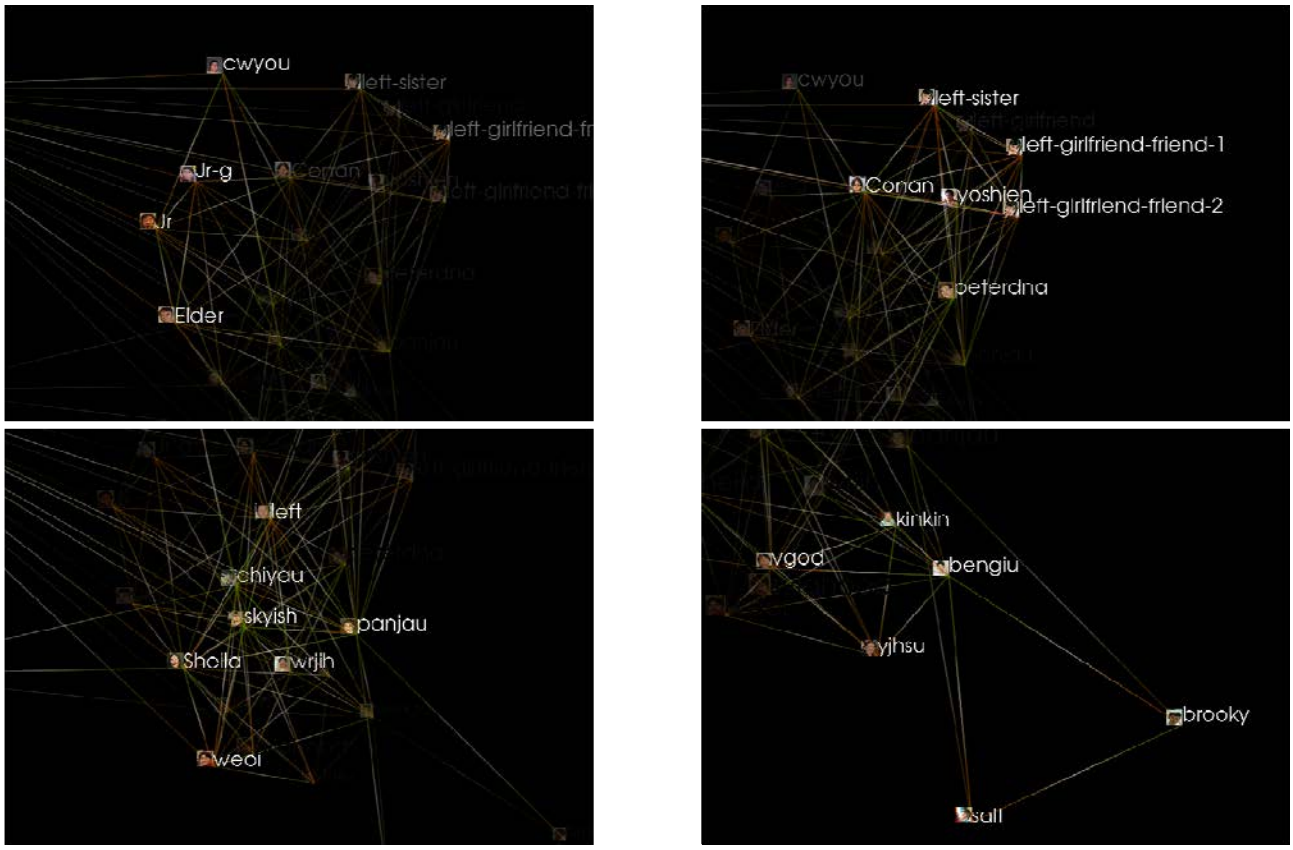

圖 4: 利用改變景深位置和範圍的一連串操作，使用者透過即時景深變化來揭露小範圍區域網路的內部連結狀況，以提升使用者對該網路的分布和連結的理解和記憶。

生重疊的問題，如圖 3所示。為求抒緩這種視覺混淆的現象，我們利用了攝影學的景深（depth-of-field）現象來解決此一問題。

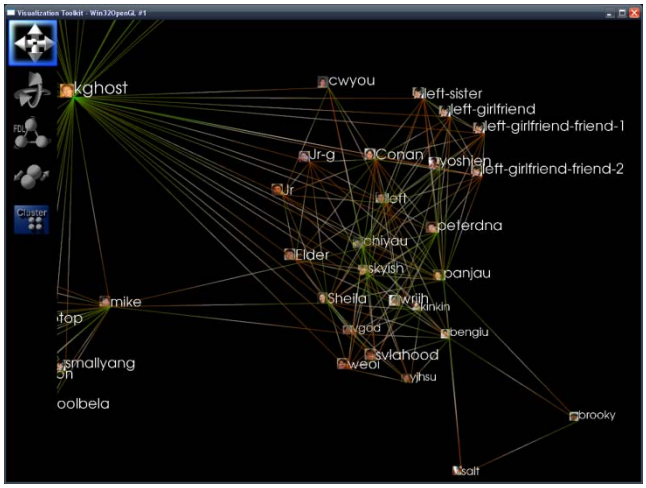

圖 3: 即使在深度上具有變化的網路佈局，也會因為投射在主要視野屏幕上，發生重疊混亂的現象。

攝影學的景深現象是指在以聚焦點（focus）為中心往前或往後的景深範圍裡頭的影像是較清楚的，超出景深範圍的物件會變得模糊（out-of-focus or blur），模糊程度和距離景深範圍平方成正比。利用景深的特性，我們提供使用者一個動態且互動式的機制，讓使用者可以動態的移動景深的位置與範圍，進一步突顯重點資訊和降低其他節點的視覺干擾。

在操作上，使用者可以藉由移動景深範圍在主要視野上深度位置，來突顯不同深度層面的網路；同時，使用者也可以藉由改變景深範圍，將景深範圍縮小，以獲得某個小深度範圍的清楚網路呈現，亦或增加景深範圍來獲得巨觀網路呈現；先前提到我們的網路佈局方式是透過吸引力/排斥力的方式來決定同一個 $Layer_d$ 內部與小區域高度連結複雜的區域網路配置，這使得複雜區域網路得以透過主要視野深度的變化，獲得配置上的舒緩；再配合上述兩者功能，使用者可以藉由即時景深變化來揭露小範圍區域網路的內部連結狀況，提升使用者對該網路的分布和連結的理解和記憶，如圖 4 所示。

值得一提的是，在同一層面網路下的各個節點，我們可以導入各個節點的屬性，賦予各節點在該層面（layer）下的主要視野深度變化的意義；舉例而言，在同樣和聚焦節點 R 的關連性深度為 d 的節點群所構成的層面網路裡，由使用者指定某個節點共有的屬性，根據該屬性的強弱或分類來決定各個節點在地區性主要視野深度的微幅變化。配合上述兩個景深操作，使用者視覺上所感受到的網路配置便能更具意義，在不失去原本視覺呈現資訊的成果上，提升在自訂篩選節點群的彈性。

4. 聚焦式分群方法

在社群網路裡，若資料來源足夠的話，在很多情況下會出現某些節點集合內部擁有高度密集的連結，這些節點集合在透過我們的焦點式多層次網路佈局顯示後，便會因為高度相互關聯所帶來的高相互吸引力，使得該節點集合在配置後會有明顯的聚集現象，並且會伴隨著內部連結高度錯綜複雜的視覺混淆現象，如圖 5所示。而這樣的視覺混淆現象一直是我們所要避免的。

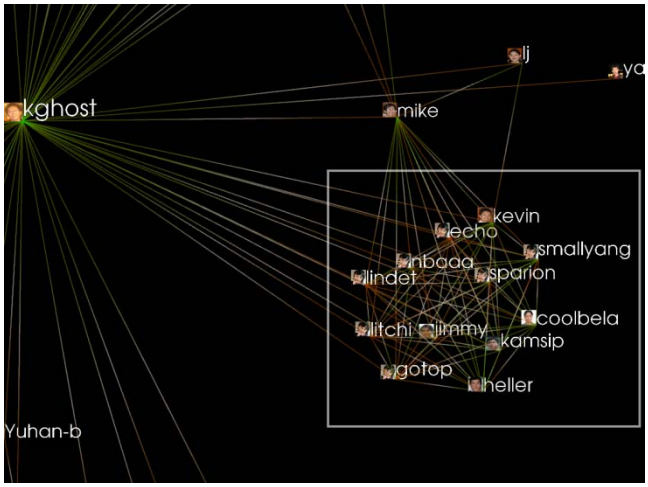

圖 5: 在很多情況下，網路佈局會出現某些節點集合內部擁有高度密集的連結；若整體網路出現很多這種結構，將會影響使用者對於該網路視覺的認知感受。

演算法 4: 節點分群演算法

對 $\mathbf{E}_{d-\text{reliable}}$ 的所有連結 E_i

Case1: 若 E_i.*NodeFrom* 節點和 E_i.*NodeTo* 節點都尚未屬於任何群組的話，便創立（create）一個新群組 $\mathbf{N}_{new}$，且 E_i.*NodeFrom* 節點和 E_i.*NodeTo* 節點均隸屬於 $\mathbf{N}_{new}$

Case2: 若 E_i.*NodeFrom* 節點或 E_i.*NodeTo* 節點之一已經屬於某群組 $\mathbf{N}_i$ 的話，則將尚未隸屬群組的節點納入該群組 $\mathbf{N}_i$

Case3: 若 E_i.*NodeFrom* 節點與 E_i.*NodeTo* 節點已經分別隸屬於不同群組 $\mathbf{N}_i$ 與 $\mathbf{N}_j$ 的話，則將 $\mathbf{N}_i$ 與 $\mathbf{N}_j$ 結合成為一個單一群組 $\mathbf{N}_i \leftarrow \mathbf{N}_i + \mathbf{N}_j$，並刪除另一群組 $\mathbf{N}_j$

因此，在本節之中，我們藉由地區性（local）分群演算法，試著從整體網路裡透過分群方法將上述所說高度相互關聯的地區性節點群做分群，並賦予該地區性節點群一個較單純的視覺配置，並且對原本交錯縱橫的連結做視覺匯流（alignment）；使用者在巨觀視野下可藉此減少地區性錯綜連結的視覺混淆，同時若使用者對地區性網絡微觀有興趣時，仍然可以藉由回復地區性節點關連細節來做細部挖掘。

首先，對於深度為 d 的子層面網路 $Layer_d$，找出其內部連結集合 $\mathbf{E}_d$；對所有隸屬於 $\mathbf{E}_d$ 集合的連結元素 E_i，E_i.*NodeFrom* 節點與 E_i.*NodeTo* 節點都屬於子層面網路 $Layer_d$。接著利用各個結元素 E_i.*length* 的長度作排序（sorting），以獲得以連結長度由小到大排序的連結序列 $\mathbf{E}_{d-\text{sorted}}$，之後藉由移除前 20%相對較長的連結元素後，便可獲得 $\mathbf{E}_{d-\text{reliable}}$，在經過接下來即將介紹的分群演算法之後，上述移除較長的連結元素的步驟，某種程度上是在對 $Layer_d$ 的內部網路做分割。

接下來，便利用這些可靠的連結集合 $\mathbf{E}_{d-\text{reliable}}$ 對子層面 $Layer_d$ 網路下的節點群做初步的分群。分群的方法如演算法 4 所示。

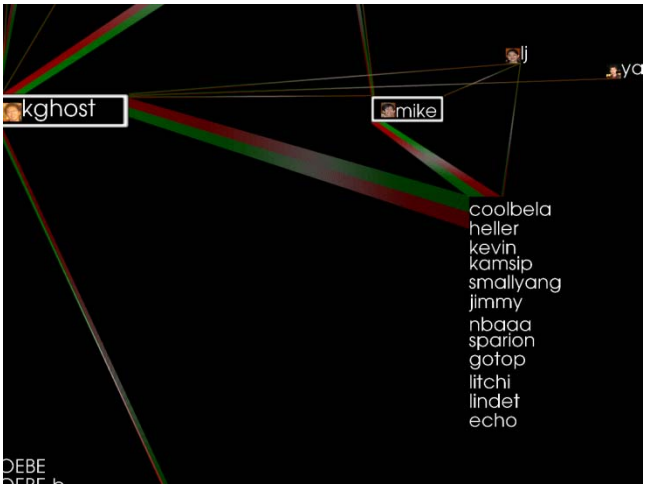

圖 6: 在演算法的分群之後，賦予各分群較單純的視覺配置，並且對原本交錯縱橫的連結做視覺匯流。

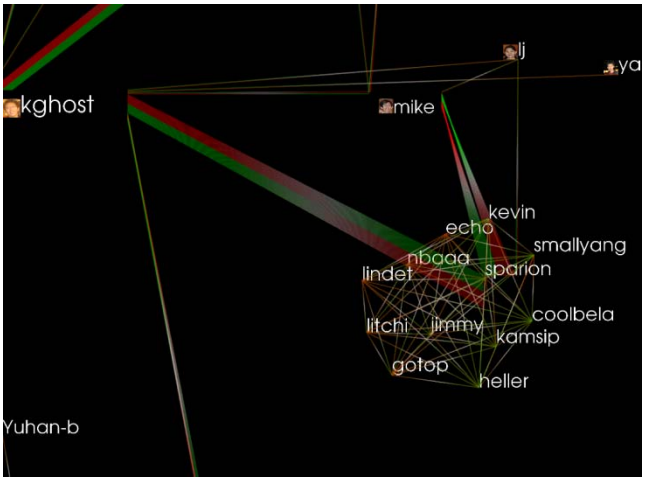

圖 7: 使用者仍然可以藉由回復地區性分群內部的關連性來做細部挖掘。

此外，在社群網路裡，若某個節點 $N_{\text{multi-grouped}}$ 同時和多個群組各別有超過一個以上連結現象的話，在經過我們的網路佈局方法後的結果可能會如圖 8所示。在圖 8中，Group A和

Group B會因為兩者之間關聯吸引力較少且排斥力較大而距離較遠，節點 $N_{\text{multi-grouped}}$ 和Group A與Group B的連結長度會因此較長而無法出現在上述的 $\mathbf{E}_{d-\text{reliable}}$ 裡，使得原本應該隸屬於多個群組的節點 $N_{\text{multi-grouped}}$ 因此而沒有被分到任何群組之中。因此，我們便再針對 $Layer_d$ 裡面尚未被分群的節點作是否為應該屬於多個群組（即是否為 $N_{\text{multi-grouped}}$ ）之辨識。

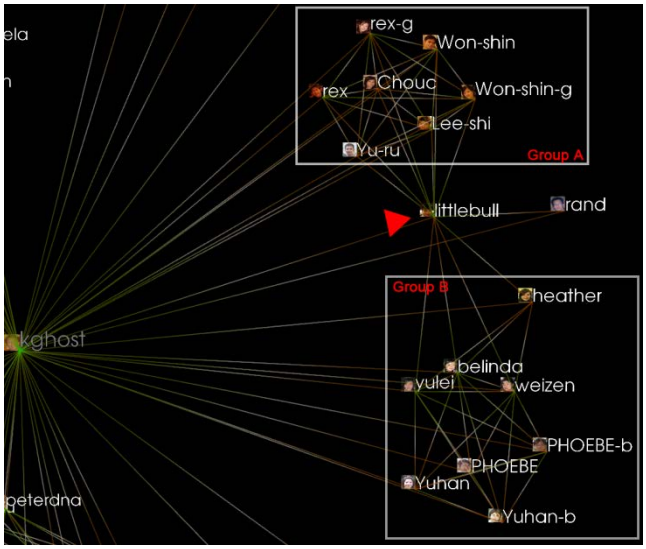

圖 8: 很明顯地看到圖中右上分群 Group A 和右下分群 Group B 個別對紅色箭頭的節點產生拉鋸效應。

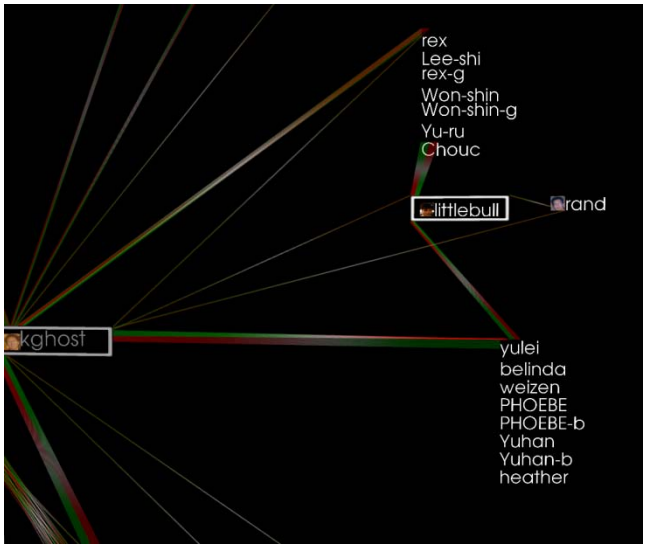

圖 9: 具有白色外框的即為隸屬多個分群的節點。

5. 多群組著色方法

分群一直是社群網路分析中很重要的一個工具，雖然在上一節中，我們已經對社群網路中的節點做一簡單的分群，但是，在現實生活中的社群網路乃至於人際關係並不能如此容易且單純的做一分群切割。因此，在本節之中，我們提出了一個簡單但可行的方式，使得使用者可以清楚地了解節點間的群組關係。

5.1 模糊分群

在社群網路中，我們發現大多數的節點很難單獨被歸類到某一個特殊的群組之下，大多數的情況下，一個節點往往可以分屬不同的群組。例如：一起做研究的朋友可能有些也會是一起出遊的朋友。因此，若僅是將一個節點強制性的規定只能屬於某一個特定的群組不但不合理，而且也往往會造成使用者對整個社群網路的誤解。

此外，在社群網路中，使用者對群組大小的概念是模糊（fuzzy）的，舉例而言：「同學」這個群組究竟指的是「高中同學」、「大學同學」、或者僅僅是「一起修過同一門課的同學」，這在不同的使用者都會有不同的定義，甚至同一個使用者在不同時間對「同學」的定義都會因為需求而有不同的變化。為了滿足使用者對群組的模糊概念，我們決定用替節點著色（coloring）的方式來取代目前大多數人使用的以群組為單位來上色的方法。

5.2 顏色擴散

我們先假定每個節點都有各自的基本顏色，而經由連結可以讓節點將自己的顏色擴散（spreading）並影響其他節點。假設有一節點 N_i，其基本色為 $N_i.color$，其相連的節點 N_j，其基本色為 $N_j.color$，若 N_i 與 N_j 間的關係定義為 $w(N_i,N_j)$，且值界於 0~1 之間，則我們可以定義出混色後的 N_i 點顏色為 $N_i'.color$，且

$$N_i'.color = N_i.color + \sum_j N_j.color \bullet w(N_i,N_j) \text{。} \qquad \textbf{(1)}$$

透過這種方式便可以讓有關係的節點有相近的顏色，且關係越接近的節點，顏色也會相對的極為相近。雖然指定顏色時並未考慮群組，但透過人類視覺上對相近色系的判斷，自然而然可以呈現出節點的分群。例如：一個黃色的點，我們可以很自然的觀察到它與淺綠色群組及橘色群組有關聯。而一個「大學同學」的深綠色系群組，則會與「同學」的綠色系群組同時在畫面上被呈現。

5.2.1 給定初始顏色

問題回到如何指定各節點的初始（initial）顏色。因為每個節點最後的顏色是混色（blending）的結果，所以在指定初始顏色時，要避免基本色與混色的結果混淆。例如：一個最後呈現為黃色的節點，可能是因為其初始色剛好是黃色，或是因為此節點與紅色系及綠色系的節點同時相關所混色而成。然而此兩種形成的原因在意義上卻是有極大的不同，因此，很可能會造成使用者誤解群組的關係。

為了避免上述的誤解產生，我們使用紅綠藍（Red, Green, Blue, RGB）三原色作為初始顏色來進行混色。因此，針對社群網路上的各節點，我們必須先在概念上盡可能的畫分出三大族群，並找出各族群中最具影響力的節點以分別指定紅、綠、藍三原色作為整個社群網路顏色擴散的起始點 R_i，$i \in \{R,G,B\}$。結合方程式 (1)後，可知節點 N_j 的顏色為 $N_j.color = \sum_{i\in\{R,G,B\}} R_i.color \bullet (1 - w(N_i,R_j))$。其中，節點 N_j 若有連結可以連到起始點 R_i，則 $w(N_i,R_j)$ 便定義為節點 N_j 到起始點 R_i 間之最短路徑（shortest path），且由於 $w(N_i,R_j)$ 的值必須界於 0~1 之間，因此，我們亦必須參考整體社群網

路的連結性，將這最短路徑予以正規化。若節點 N_j 與起始點 R_i 不相連，則 $w(N_i,R_j)=1$。因此，最後節點 N_j 的顏色就不會受到不相關的起始點 R_i 所影響。

將此多群組著色方法套用到子海（word clouds）關聯網路後，結果便如圖 10所示。從圖 10中可以明顯的看到除了初始給定的紅綠藍三原色的三群組外，在圖中的左上角並渲染（dithering）出偏於紫色的群組，以及左下角之偏向草綠色之群組。

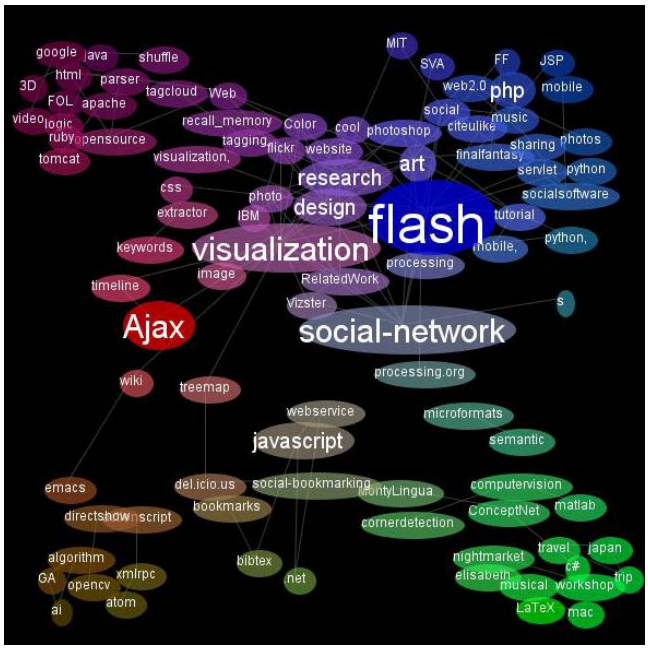

圖 10: 利用多群組著色法所產生的結果，藉此清楚了解節點的分群關係。

5.2.2 獨立節點

但是在不少的例子下，會發現扣除掉三大族群外，往往還會有一些獨立的節點或節點組。針對這些獨立的節點，我們另外指定了灰色，以標示出它們的獨立性。同時，為了避免與混色結果混淆，這些節點顏色的亮度還會被降低。因為在有些社群網路的研究中，我們必須避免這些獨立的群組影響主體的統計結果。

6. 結論

從結果看來，聚焦式的視覺呈現方式應用在社群網路的資訊視覺化的確是對於理解複雜的人際關係網路是有幫助的，最主要的原因在於觀察社群網路時，大部分使用者會先對裡面既有利害關係的個人節點有興趣；因此，我們便順應著這個既有需求，來對複雜的人際關係網路做資訊視覺呈現，在三維空間提供更多可能性的優勢下，應用焦點式多層次網路佈局演算法確實改善了原本在二維平面難以獲得較佳的網路佈局呈現的窘狀。搭配著景深效果工具來減少三維網路投影在二維顯示器上所產生的視覺混淆更近一步提供了多維度資訊呈現的可能性；藉由三原色的混色，使得社群網路中各節點間模糊的群組概念可以完整的被呈現出來。

未來希望能將我們目前的視覺化呈現概念，應用在其他的網路節點型態的資料上，並且持續改善大規模網路視覺化上的困難，例如交錯線段（cross edge）、變化觀看視野所產生的節點重疊現象、更有效地利用顯示空間等等，都是我們持續努力的方向；分群的部分在現有的演算法機制下，獨立節點（independent node）僅以灰色標示以便使用者分析時排除這些節點的干擾。然而現在也有一部分人會針對這些獨立的節點來分析，因此未來我們希望可以加強對獨立節點的呈現。

7. 致謝

感謝國立臺灣大學許永真教授及其領導之研究團隊提供本系統實驗所用之人際網路資料。本計畫由國科會資助，計畫編號為 95-2622-E-002-018 與 96-2622-E-002-002。

8. 參考文獻

[1] Eades, P. and Feng, Q.-W. 1997. Multilevel visualization of clustered graphs. *Proceedings of the Symposium on Graph Drawing 1996*, 101-112.

[2] Eades, P., Houle, M. E., and Webber, R. 1997. Finding the best viewpoints for three-dimensional graph drawings. *Proceedings of the Symposium on Graph Drawing 1997*. 87-98.

[3] Eades, P. and Sugiyama, K. 1990. How to draw a directed graph. *Journal of Information Proceedings*, 13, 4, 424-434.

[4] Fruchterman, T. M. J. and Reingold, E. M. 1991. Graph drawing by force-directed placement. *Software - Practice and Experience*. 21, 11, 1129-1164.

[5] Herman, I. and Marshall, M. S. 2000. GraphXML -- an XML based graph interchange format. Reports of the Centre for Mathematics and Computer Sciences.

[6] Juang, M. L. and Eades, P. 1998. A fully animated interactive system for clustering and navigating huge graphs. *Proceedings of the Symposium on Graph Drawing 1998*, 374-383

[7] Misue, K., Eades, P., Lai, W., and Sugiyama, K. 1995. Layout adjustment and the mental map. *Journal of Visual Languages and Computing*. 6, 183-210.

[8] Purchase, H. C., Cohen, R. F., and James, M. 1995. Validating graph drawing aesthetics, *Proceedings of the Symposium on Graph Drawing 1995*, 435-446.

[9] Rekimoto, J. and Green, M. 1993. The information cube: using transparency in 3D information visualization. *Proceedings of Workshop on Information Technologies & System 1993*. 125-132.

[10] Robertson, G. G., Card, S.K., and Mackinlay, J. D. 1993. Information visualization using 3D interactive animation. *Communication of the ACM*, 36, 4, 57-71.