

支援通用存取之使用者意向聚焦與資源感知語意轉形

黃俊雄

國立東華大學資訊工程學系
jhuang@mail.ndhu.edu.tw

吳秀陽

國立東華大學資訊工程學系
showyang@csie.ndhu.edu.tw

摘要

伴隨著網際網路的普及與行動裝置的大量的使用，連帶的提升了使用者對於資訊存取的需求，故網路服務的提供必須隨著多元化的網路型式、使用者裝置及使用者喜好而進行改變。也因此有了轉形系統(transcoding system)的出現，其目的是根據文件種類和使用者裝置的特性，調整輸出內容給使用者。但目前被廣泛使用的轉形系統，其設計理念，往往偏向單純的媒體轉形或是語意轉形，並未考慮到使用者在擷取資訊時的意向所在，以及對使用者裝置上的可用資源做最有效的運用。於是我們設計了一個以通用存取(universal access)為目標的轉形系統架構，結合對使用者意向(user focus)偵測與使用者裝置資源感知(resource aware)的特性，能夠在產生轉形計劃時，進行符合使用者意向的語意轉形，同時有效率的使用裝置上的資源。這二項特性來自於我們所提出以本體論(ontology)為基礎的使用者意向樹(user focus tree)建立、意向權重值(focus weight)計算、資源消耗指數(resource consumption index)、資訊保留指數(information preservation index)、資源感知語意轉形演算法(resource aware semantic transcoding algorithm)等技術的結合而達成。實驗結果顯示，我們所提出的方法可以相當正確而迅速的找出使用者瀏覽意向，並據以在儘可能配合使用者意向及提高資訊保留比的情形之下，妥善的運用使用者裝置上的可用資源。

關鍵詞：語意轉形、通用存取、使用者意向偵測、資源感知、本體論。

1. 前言

隨著行動裝置與無線網路的普及，如何能夠在不同的裝備與不同的網路環境中，迅速、正確的提供資訊給使用者，成為一個重要的課題。其間雖已有不少的研究被提出，但大部份的研究都是針對檔案內容單一物件進行調適或是只對影像檔進行轉形(transcoding)，並未考慮到不同的裝置所能使用消耗的資源各不相同，各個文件內容中不同的物件彼此也具備著觀念上的結構關係，使用者在查看資訊時通常具有只對部份資訊特別有興趣等等的特性。我們提出一個架構與方法，可以針對不同的使用者與裝置，依據裝置可使用的資源進行最佳化的

內容轉形，再結合透過使用者意向聚焦，尋找使用者較有興趣的內容，在轉形時，給予較好的裝置資源分配，達到資源感知語意轉形的目的。我們以本體論為基礎，提出使用者意向樹(user focus tree)結構來追蹤記錄使用者的瀏覽意向與興趣關聯，再對樹中的節點導出意向權重值(focus weight)計算公式，將使用者意向予以量化。為了有效率的使用裝置資源，我們提出資源消耗指數(resource consumption index)的概念，將原始資訊中的各個物件可能消耗的資源值予以量化，並導出計算資源消耗值的公式。接著我們結合使用者意向權重值及資源消耗指數到資料轉形運算中，設計一個資源感知語意轉形演算法(resource aware semantic transcoding algorithm)，根據這兩項主要參數，來進行轉形計劃的產生，達到最高的資訊保留指數(information preservation index)與最大的資源利用率。為了評估系統的效果，我們設計實作了了一個離型系統與系列實驗來分析演算法及計算公式的準確性和效能。實驗結果顯示，我們所提出的方法可以相當正確而迅速的找出使用者瀏覽意向，並據以在儘可能配合使用者意向及提高資訊保留比的情形之下，妥善的運用使用者裝置上的可用資源。

本文架構如下，第二節介紹所用到的技術，最新的研究和相關系統，以及他們的做法。第三節詳細說明我們系統的架構與理論。第四節說明及展示系統實作成果，各項技術的效能評估與分析。第五節對整個研究做結論，並探討未來的立即目標及進一步研究方向。

2. 相關技術與研究

通用存取(universal access)的研究是讓資訊擷取擺脫時間、空間、網路環境、使用者裝置等等的限制，達到無時不刻、無所不在、無網路不通、無裝置不可的目的[7][14]。其中最重要的技術就是所謂的轉形(transcoding)[3][6][8][12]，目的是將多媒體資訊加以轉換，以適應各種不同的網路環境和使用者裝置，特別是無線網路和行動裝置[1][2]，近年來更進一步發展到串流媒體(streaming media)環境[10]。舉例而言，[11]提出了以內容為基礎的圖形檔轉形架構，將圖形檔依格式和用途分類，並設計各種轉形函數作為轉形運算之基礎。[5]則是提出 Intermediary-Based 的轉形架構，透過 WBI(Web Intermediaries)，開發者可以很方便的在 HTTP 協定

刪除: 1

表 1 RCI 對應表

Resource Type And Media Type	RCI Value
Text	n
GIF image	2n
JPG image	5n
MPEG Video	10n
Audio	n
MP3	10n

3.3 裝置與使用者特性描述(Device and User Profiles)

為了能有效率的決定轉形運算，瞭解使用者裝置以及使用者的喜好資料是不可或缺的，我們定義了裝置側寫(device profile)及使用者側寫(user profile)的參數值供系統判斷使用。在裝置側寫上，我們借用 W3C 制定中的 CC/PP 標準[15]，用來瞭解與記錄使用者裝置特性，主要有二大分類，一為裝置的硬體特性，一為瀏覽器或 user agent 的特性，欄位明細如表 2 所示。所取用欄位的因素取決於此項特性，在計算 RCI 與轉形運算是否為必要資料而定。例如螢幕解析度有助於圖形轉形時的比例設定，播放影像能力則決定此裝置能否接收影像形態的資料。

表 2 裝置側寫欄位表

HardwarePlatform	與硬體相關特性
ScreenSize	螢幕解析度
Model	裝置型號
ScreenSizeChar	單一字元所佔大小
BitsPerPixel	每點最大位元數
ColorCapable	顯示彩色能力
ImageCapable	顯示圖形能力
SoundOutputCapable	輸出聲音能力
VideoCapable	播放影像能力
BrowserUA	軟體 user agent 特性
BrowserName	User agent 名稱
CcppAccept	可接受的媒體 MIME 型態
CcppAccept-Encoding	可接受的編碼
FramesCapable	是否支援 Frame 能力
TablesCapable	是否支援 Table 能力

使用者側寫則是用來記錄使用者的各項喜好特性，一部份資料開放給使用者自行設定，其他則由系統主動收集記錄。表 3 為我們所使用的欄位對照表，其中去除了與使用者個人資料及隱私相關的欄位，只保留了在系統運算中必要的欄位。例如喜好的 ontology 提供給系統在建構 Focus Tree 時可以預先將喜好的某類 ontology 加入 Focus Tree。而不使用的媒體型態，則是在產生轉形計劃時，能夠在前置處理時，先行進行轉形。本文所用到的使用者側寫資料，仍是以系統主動收集學習者為主。

表 3 使用者側寫欄位表

Favorite_ontology	喜好的 Ontology
Disable_media type	不使用的媒體型態
Resource_usage	資源的使用比率
Information_usage	資訊的保留比率

3.4 使用者意向聚焦樹(User Focus Tree)

為分析使用者瀏覽意向，我們設計了以本體論為基礎，逐步建構使用者意向聚焦樹 (User Focus Tree)的方法，來計算出使用者的興趣所在。聚焦樹初始時只含 root 節點，隨著使用者瀏覽，我們在 domain ontology tree 中找出直接符合與間接相關之觀念節點加入聚焦樹中。圖 2 顯示一個 domain ontology tree 以及使用者在瀏覽 e 觀念節點之後所建立的使用者意向聚焦樹。

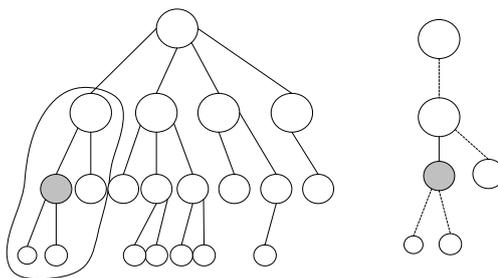


圖 2 瀏覽 e 後的 Domain Ontology Tree 與 User Focus Tree

聚焦樹中每個節點，都要記錄 3 個變數值，以計算權重，分別是: P 為主要節點權重累計值; R 是相關節點權重累計值; F 則是節點的 focus weight 計算值。另外定義了一個 scaling factor 數值 k 做為調整興趣差距之用。整個建構由下列五個步驟而成:

1. 新增使用者所選節點 e，以實線連結，代表此節點為主要節點，並累計 e 節點的主要權重值 P，如圖 3 中步驟 1。
2. 新增 e 節點的母節點 a，以虛線連結，代表相關節點，並累計 $k/(2*(a.childs + 1))$ 到 a 節點的 R 參數，如圖 3 中步驟 2，在概念上父節點所擁有的子節點越多，代表主要節點的兄弟節點就越多，連帶的主要節點與父節點的關係越弱。
3. 新增 e 節點的兄弟節點 f，以虛線連結，代表相關節點，並累計 $k/(2*(e.childs + 1 + a.childs))$ 到 f 節點的 R 參數，如圖 3 中步驟 3。主要概念來自於考慮主要節點與兄弟節點之間的關係值，以親屬關係而言，子節點關係應強過兄弟節點關係，故計算式同時使用子節點數與兄弟節點數來做為計算值。
4. 新增 e 節點的子節點 m 及 w，以虛線連結，代表相關節點，並累計 $k/(2*(e.childs + 1))$ 到 m 及 w 節點的 R 參數，如圖 3 中步驟 4。計算式中

刪除: 1

m w

同樣的使用到子節點數量來計算子節點與父節點 e 的相關值。

- 當主要節點與相關節點都完成處理後，便對 focus tree 裏的每一節點，重新計算 F (focus weight) 值，計算公式為 $F(n) = n.P + n.R/\sqrt{(n.R^2 + 1)}$ ，成為該節點的權重值。

在進行節點新增時，如果節點已存在，則不再新增，如為主要節點，只將連結改為實線標記，如為相關節點，則連結線狀態不做改變，節點權重值，則依前列方法累計。再將節點依 focus weight 進行排名動作，如此一來便能夠得到使用者目前的意向所在為那些節點，圖 4 為使用者在選擇了 e, a, h, 及 j 節點後的 focus tree，圖 5 為最終的 focus weight 計算值。經過整理排序之後檢視節點順序與權重比例，我們發現整個建構方法確實能合理的反應節點相對重要性與相互關係，進而得知使用者的興趣偏好。

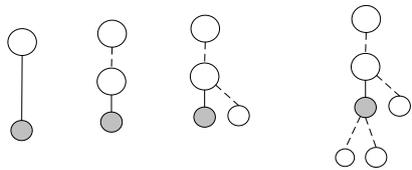


圖 3 User focus tree 建構分解步驟圖

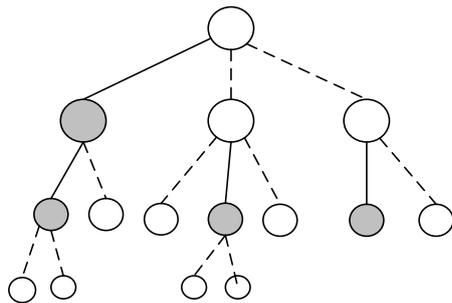


圖 4 a、e、h、j 選擇節點加入後的 User focus Tree

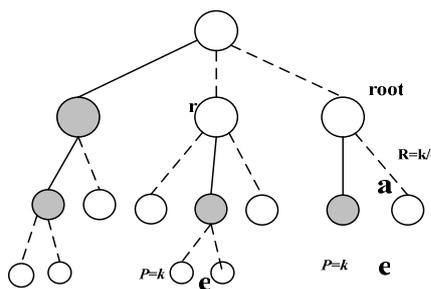


圖 5 a、e、h、j 選擇節點加入後的權重值

3.5 轉形運算(Transcoding Operations)

為了搭配使用者意向權重值計算，必須要能夠有系統的界定轉形運算的特性，才能決定如何進行轉形。因此我們設計了如表 4 之轉形對照表，每一個轉形運算，接受輸入的來源內容將之轉形為輸出目的內容，而目的與來源內容的檔案大小比值即為資訊保留比 IPR，為轉形計畫制定的重要依據。

轉形運算的設計與決定原則，在於不同型態之間的轉形運算，應儘可能提供在目標型態中，含蓋全部的媒體型態，如此有利系統支援較多的裝備。其次則為相同型態的轉形運算數量要夠多，且轉形後的資訊保留比 IPR 應儘可能的分佈均勻，以利系統在資源分配及最佳化時得到較好的效果。

表 4 轉形運算對照表 (N: original size, L: Text Length)

原始型態	Transcoding Operation	目標型態	IPR
Image	解析度降 1/2	Images	1/4
Image	解析度降 1/4	Images	1/16
Image	解析度降 1/8	Images	1/64
Image	解析度降 1/16	Images	1/256
Image	解析度降 1/32	Images	1/1024
Image	32bits 彩色轉黑白	Images	1/1024
Image	32bits 彩色轉 4 bits 灰階	Images	1/8
Image	32bits 彩色轉為 8 bits	Images	1/4
Image	32bits 彩色轉為 16 bits	Images	1/
Image	32bits 彩色轉為 24 bits	Images	3/4
Image	image link url	link	L/N
Image	image annotation	Text	L/N
Image	image file name	Text	L/N
Image	annotation	Text	L/N
Text	Realignment	Text	L/N
Text	start frame	images	L/N
Video	video annotation	Text _{root}	L/N
Video	video link url	link	L/N
Video	Annotation	Text	$R=k/(2*(a.children+1))$
Audio	audio link url	link _a	L/N

3.6 資源感知語意轉形演算法(Resource f Aware Semantic Transcoding Algorithm)

在 3.5 節中，我們已經設計了轉形運算，使用者意向聚焦樹的建立、以及權重值的計算之後，就可以進行語意轉形。圖 6 展示了語意轉形演算法的步驟：

1. Add selected node
2. Add Parent node
3. Add sibling node
4. Add child nodes

刪除: 1

root

R=k/8
P=0

R=k/6
P=0

R=k/6
P=k

a

b

c

計畫產生的動作。我們設計了一個資源感知語意轉形演算法，其整體步驟如圖 6 所示，可分成五個 phase 進行運作。

1. Pre Processing (前置處理): 根據 user profile 及 device profile 對要處理的物件，檢查原始資料形態是否為使用者或裝置可接受的型態，如為兩者都無法接受，則在可接受的資料型態中找出轉形後最高資訊保留比者做為新的物件資料型態，如無法找到轉形後可接受的型態則將此物件予以移除不輸出。
2. Transcoding Limits Setting (轉形上限設定): 根據物件權值，設定各物件的轉形資源使用上限，此為結合使用者意向聚焦與資源感知語意轉形的核心所在。從 Focus Tree 中獲得物件的權值，全部物件的 Weight 加總後再逐一計算各個物件的權值百分比，即為此該物件可以使用裝置資源百分比的上限。如此一來，使用者最感興趣的主題將獲得最高比例的資源分配。
3. Objects Filtering (物件過濾): 根據各個物件的資源使用上限，過濾掉超過上限的物件。
4. Generating Initial Transcoding Plan (產生初始轉形計劃): 從過濾後所得的物件及資源上限和可用的轉形運算產生初步的轉形計劃。產生方法是針對每一個物件，在所有可用之轉形運算中，選擇轉形之後不超過資源上限且資訊保留比最高的轉形運算。
5. Resource and Transcoding Plan Optimization (資源使用及轉形計劃最佳化): 根據裝置資源的使用，將初始轉形計劃進一步做最佳化處理，以達最有效率的資源使用。主要方法是迴圈式的檢視目前剩餘可用裝置資源，然後依據物件權重順序，檢查是否存在更好的轉形運算可以充分利用剩餘資源，一直到資源完全分配，或是找不到更好的轉形運算，才結束最佳化動作。

演算法的完整運作流程圖詳見附錄 A。

我們所提出的演算法有幾個重要特色：

1. 充分照顧到使用者瀏覽意向，較有興趣的主題也會在初始計畫的產生和最佳化過程中獲得較多的資源分配，因此選擇資訊保留比較高的轉形運算，得出更接近原始資訊的轉形結果。
2. 動態依照裝置的不同，盡可能運用裝置資源，做最佳比例的分配。讓使用者可以獲得裝置能力所允許的最佳內容。
3. 因為轉形運算的選擇充分考慮到資訊保留比，依據最終轉形計畫所得出之內容，也是符合資源分配比例的所有可能轉形計畫中，資訊保留比最高的轉形計畫。
4. 使用者所不感興趣或是裝置能力無法處理的物件，會在最早可能階段即予以排除，大幅降低後續處理複雜度，提升轉形效率。
5. 演算法的普及性強，能夠與任何提供型態或語意轉形運算的系統合作，將之提升為具備使用者意向聚焦和資源感知能力的轉形系統。

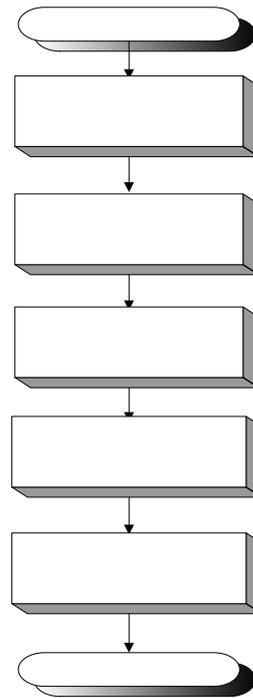


圖 6 資源感知語意轉形演算法整體步驟

4. 系統實作與效能分析

我們以 Delphi IDE 做為系統設計工具，考量到實作完整系統的複雜度，我們採用部分實作部份模擬的方式，獲得轉形計畫後，再輔以手動轉形工具程式，對物件進行轉形，獲得最終結果。系統運作於作業系統 Windows 2000，在圖形的轉形採用了 ImageMagick 6.02 Tools and Library。在本節裡，我們將呈現實作成果，對轉形系統進行評估與測試，並對測試結果做分析與比較。

4.1 資源感知語意轉形效果評估

我們以國立東華大學校史多媒體簡介網頁（內含文字、圖片、影片等五個物件）為測試對象，並以一個虛擬的典型學生建立聚焦樹，然後以資源感知語意轉形演算法，針對 Desktop、PDA、Mobile Phone 等三種裝置來進行實驗，分別產生轉形計畫，實際套用轉形運算之後，檢視轉形效果。由於 Desktop 資源充足，故未使用任何轉形。而 PDA 裝置無法呈現巨大的 video 檔案，演算法選擇予以轉形為 Image 型式，而 Mobile Phone 由於可用資源更小，故起始轉形計畫將二個 Video 物件都轉為 Annotation，只保留一個 Image 物件，但資訊保留比先給予較小的 1/32 值轉形。在資源最佳化階段中，Mobile Phone 由於尚有部份 RCI 值可使用，故逐步的調整 Transcoding Operator，最終改為較高的

資訊保留比的 1/2 值轉形。圖 7、圖 8、圖 9 分別是在 Desktop、PDA、Mobile Phone 上的呈現結果，可以明顯的看出系統確實能夠正確的針對不同裝置進行適當的轉形處理，達成語意轉形及通用存取的目的。



圖 7 網頁內容於 Desktop IE 之轉形結果呈現



圖 8 網頁內容於 PDA 之轉形結果呈現



圖 9 網頁內容於 Mobile Phone 之轉形結果呈現

4.2 使用者意向聚焦效果評估

為了量測系統對使用者意向的追蹤符合度及效率，我們先產生五個假設的使用者興趣項目資料及對應的興趣度權值，而後逐一的對此五個項目模擬使用者點選動作。隨著每次的點選，系統的 Focus Tree 逐次的增長，再將每次點選後之 Focus Tree 與假設的權值進行符合度比較。另外又為了評估系統

能否快速的追蹤到使用者的意向改變，在五個項目點選完後重新再產生五個項目，以觀察系統的追蹤效率，所得實驗結果如圖 10 所示。我們發現，演算法可以確實有效的追蹤到使用者的真正意向，每次的點選都能逐次提升符合度，並在最後一次點按後達到 87% 的符合度，以及在使用者意向改變後快速的再度追蹤到使用者意向。

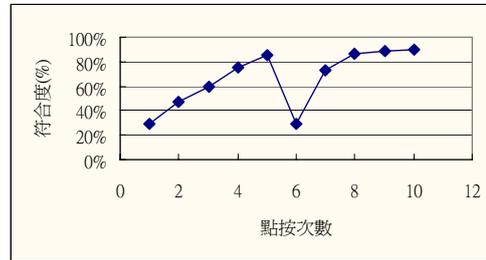


圖 10 使用者意向追蹤符合度與效率

4.3 語意轉形效能評估

轉形運算的執行效率對成功的轉形系統來說，是相當重要，一般轉形運算在處理較大的資料量時所需時也會線性成長。為了測試我們的系統在原始媒體資源 (SRCI) 超過裝置資源 (DRCI) 時轉形的速度能否維持在合理的範圍內，我們先測量記錄各個轉形運算所需的時間，再存入資料庫中計算，使用實驗資料為 10 個多媒體物件和 30 個轉形運算於 Desktop 裝置上，所得實驗結果如圖 11 所示。在比值 1 與 2 會需要較多的時間，但當比值超過 3 倍後則維持在一個時間範圍內振盪，並不會隨著原始媒體的複雜度增加。分析實驗結果，可以了解當比值在 1-3 之間時，因轉形的資訊保留比較高，故在處理上會花費較多時間，但當超過 3 之後，因可能已有些物件即使在轉形後裝置也無法負荷，故被移除或轉為 annotation 的機會也升高，需要轉形的物件數量反而少，也導致轉形所需的總體時間下降，最後維持在一個時間範圍內。

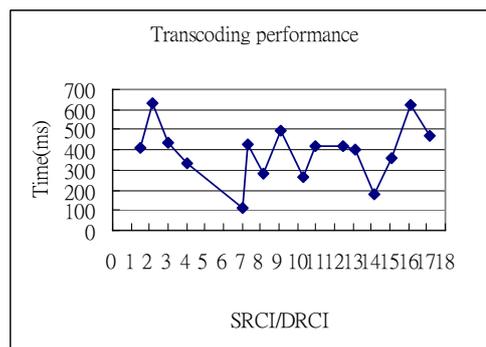


圖 11 轉形平均執行速度圖

4.4 語意轉形後的資源使用率評估

系統在產生轉形計劃後，對使用者裝置上的資源能否確實的有效運用是相當重要的。為了觀察系統對裝置資源的使用效率，我們針對不同的裝置進行資源使用率的測試實驗，實驗結果如圖 12 所示。顯示我們的演算法所產生出來的轉形計劃確實可以有效率的使用裝置上的資源，但必須提供足夠數量的轉形運算，且轉形運算的資訊保留比的數值分佈足夠均勻。

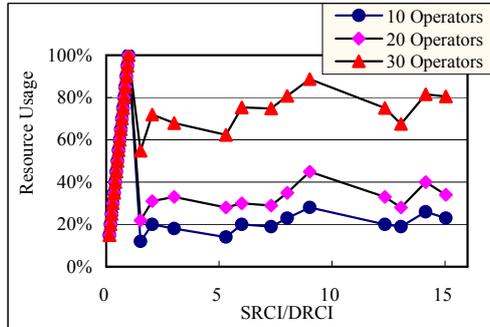


圖 12 轉形資源平均使用率比較圖

5. 結論與未來展望

我們結合了使用者意向聚焦與資源感知運算應用於語意轉形上，再配合資源消耗指數與資訊保留指數概念，有效的產生轉形計畫，既能符合使用者瀏覽意向，又能夠妥善運用裝置資源，足以成功應用於通用存取之上，實驗結果顯示我們的方法可以很快的抓住使用者興趣，同時在轉形運算足夠的時候，能夠成功地產生高資訊保留比的轉形計畫。往後將進一步朝向自動化輸出內容的配置轉換 (Content Rendering)，以及自動化轉形運算產生兩方面積極進行，結合語意註解、調適性呈現、及個人化技術，徹底解決語意轉形與通用存取問題。

致謝

本研究部分由國科會計畫 92-2213-E-259-010 以及 93-2213-E-259-012 支持，謹此致謝。

參考文獻

[1] Harini Bharadvaj, Anupam Joshi, Sansanee Auephanwiriyaikul, "An Active Transcoding Proxy to support Mobile Web Access", *17th IEEE Symposium on Reliable Distributed Systems*, 1998, pp 20-23.

[2] Surendar Chandra, Carla Schlatter Ellis, Amin Vahdat, "Multimedia Web Services for Mobile

Clients Using Quality Aware Transcoding", *2nd ACM International Workshop on Wireless Mobile Multimedia*, August 1999.

[3] R Cucchiara, C Grana, A Prati, "Semantic Transcoding for Live Video Server," *ACM Multimedia*, 2002.

[4] Anita W. Huang, Neel Sundaresan, "A Semantic Transcoding System to Adapt Web Services for Users with Disabilities", *4th International ACM Conference on Assistive Technologies*, November 2000.

[5] S. C. Ihde, P. P. Maglio, J. Meyer, and R. Barrett, "Intermediary-Based Transcoding Framework" *IBM Systems Journal*, 40(1): 179-192, 2001.

[6] B Knutsson, H Lu, J Mogul, B Hopkins, "Architecture and performance of server-directed transcoding," *ACM Transactions on Internet Technology* 3(4), November 2003.

[7] R Mohan, JR Smith, CS Li, "Adapting Multimedia Internet Content for Universal Access," *IEEE Transactions on Multimedia* 1(1):104-114, March 1999.

[8] Katashi Nagao, Yoshinari Shirai, Kevin Squire, "Semantic Annotation and Transcoding: Making Web Content More Accessible," *IEEE Multimedia* 8(2): 69-81, April-June 2001.

[9] Thomas Phan, George Zorpas, Rajive Bagrodia, "An Extensible and Scalable Content Adaptation Pipeline Architecture to Support Heterogeneous Clients", *22nd International Conference on Distributed Computing Systems*, 2002, pp 507 - 516.

[10] B Shen, SJ Lee, S Basu, "Caching Strategies in Transcoding-Enabled Proxy Systems for Streaming Media Distribution Networks," *IEEE Transactions on Multimedia* 6(2): 375-386, April 2004.

[11] John R. Smith, Rakesh Mohan, Chung-Sheng Li, "Content-based Transcoding of Images in the Internet" *ICIP'98. International Conference on Image Processing*, Oct. 1998.

[12] John R. Smith, Rakesh Mohan, Chung-Sheng Li, "Transcoding Internet Content for Heterogeneous Client Device", *ISCAS'98: IEEE International Symposium on Circuits and Systems*, 1998, Vol 3, pp 599-602.

[13] C. Stephanidis, A. Paramythis, D. Akoumianakis, M. Sfyarakis, "Self-Adapting Web-based Systems: Towards Universal Accessibility," In C. Stephanidis & A. Waern (Eds.), *Proceedings of the 4th ERCIM Workshop "User Interfaces for All"*, Stockholm, Sweden, 1998, pp. 19-21.

[14] Constantine Stephanidis: "Adaptive Techniques for Universal Access," *User Modeling and User-Adapted Interaction* 11(1-2):159-179, 2001.

[15] W3C CC/PP Information Page. (<http://www.w3.org/Mobile/CCPP/>)

附錄 A 資源感知語意轉形演算法流程

