복잡한 다단 문서 영상으로부터 구조화된 하이퍼문서의 자동 생성

이 지연, 강 희중, 이 성환

고려대학교 컴퓨터학과/인공시각연구센터

E-mail: {jylee, hjkang, swlee}@image.korea.ac.kr

Automatic Generation of Structured Hyperdocuments from Multi-Column Document Images

Ji-Yeon Lee, Hee-Joong Kang and Seong-Whan Lee

Dept. of Computer Science and Engineering/Center for Artificial Vision Research, Korea University

요 약

본 논문에서는 다양한 객체를 포함한 다단 문서 영상을 원본 문서와 거의 유사한 형태의 HTML 문서로 변환할 수 있는 방법을 제안한다. 또한 논문이나 메뉴얼, 책의 한 단원 등 여러 장의 입력 문서의 경우, 문서의 논리적인 구조 분석을 수행하고 장이나 절 등의 섹션 제목들을 계층화하여 다단 문서의 변환과 동시에 구조화된 목차 페이지도 함께 자동 생성하는 방법을 제안한다.

제안된 다단 문서 변환 알고리즘을 잡지, 신문, 광고지, 메뉴얼 등 비정형화된 문서에 적용한 결과, 원본 문서의 형태와 구조에 큰 변형없이 유사하게 변환되었고, 논리적인 구조 분석 및 섹션 제목들의 계층화 작업 또한 정확히 수행되어 구조화된 목차 페이지의 자동 생성이 가능하였다.

1. 서론

오늘날 인터넷 사용이 급증하고 그 수요가 폭발적으로 증가하면서 보다 접근이 간편하고 전송이 용이한 웹 기반의 HTML 문서가 중요한 형태의 새로운 문서 종류로 자리 잡게 되었다. 그러므로, 종이 문서를 단순히 전자 문서로 변환하는 기술 뿐만 아니라, 웹 상의 구조화된 하이퍼문서로 변환하는 기술이 절실히 요구된다.

HTML, SGML, XML과 같은 구조화된 전자 문서로 종이 문서를 변환하기 위한 연구로 Worring과 Smeulders[1]는 그림과 텍스트 객체만으로 구성된 메뉴얼을 대상으로 변환하는 시스템을 구현하였다. 이는 그림 객체를 뽑아 내고, 그 안에 텍스트와 본문 중의 일치하는 텍스트를 하이퍼링크로 구성해 주는 방식이다. 그 외에 테이블 안에서 형성될 수 있는 라인들의 교차점 유형을 모두 정의하고, 변환시 이 교차점 유형에 따라 정의된 태그와 Colspan, Rowspan 값을 계산하는 테이블 변환 알고리즘[2][3]을 제안한 연구가 있었다. Faure[4]는 대문자와 숫자, 그리고 몇 개의 키워드를 가진 텍스트 객체들을 대상으로 섹션 제목 후보들을 뽑아내고 이를 함께 고려함으로써 목차를 생성하였다. 그러나, 이러한 조건을 모두 만족하면서도 제목이 아닌 경우나 잘못 추출된 데이타에 대한 확인 및 오류 수정 단계가 없다는 문제점을 가진다.

본 논문에서는 테이블 셀들의 정보만으로도 더욱 간단하고 빠르게 변환할 수 있는 알고리즘을 제안하고, 이를 다단 문서 변환에도 그대로 적용할 수 있도록 하였다. 또한 서로 연관된 여러 장의 문서를 변환시, 각각의 문서를 변환함과 동시에 섹션 제목들을 레이블링하고 이들을 계층화한 후, 확인 및 오류 수정 단계를 거침으로써 정확한 제목 리스트를 생성한다.

2. 다단 문서의 변환

2.1 Table-to-HTML 알고리즘을 이용한 변환

이 장에서는 테이블 객체를 변환하는 알고리즘을 다단 문서 변환에도 그대로 적용하기 위한 방법을 그림 1과 같은 순서로 제안한다.

우선 HTML 문법에서 지원하는 <Table> 태그를 이용하기 위해서는 테이블을 만들 수 있는 형태로 객체들을 변경해야 한다. 기하학적인 구조 분석을 통해 분류된 객체들을 수평, 수직으로 투영하여 비어있는 공간을 수평과 수직으로 나누고, 상하에 이웃한 객체가 텍스트일 때는 이를 병합하며, 서로 다른 객체일 때는 이웃한 단의 가장 가까운 객체와의 중간 지점을 나누는 방식을 사용하는데, 나눠지지도 않고 테이블을 만들 수도 없는 구조로 분류된 객체들을 알맞게 나누기 위한 방법이 그림 2와 같다. 이렇게 테이블을 만들 수 있는 형태로 객체들을 병합한 후, 병합된 객체들을 셀로 간주하여 가상 테이블을 만들고 셀들이 생성되는 순서로 객체들을 정렬한다. 아래의 두 가지 조건을 만족하는 순서로 재정렬하면 테이블을 변환하는 알고리즘을 적용하기 이전 단계가 모두 끝나게 된다.

(1) $|y_i^{TL} - y_j^{TL}| < Th$:O_i와 O_j 객체는 동일한 행에 존재
$|y_i^{TL} - y_j^{TL}| > Th$:O_i와 O_j 객체는 다른 행에 존재

(2) $x_i^{TL} < x_j^{TL}$:O_i가 O_j 객체보다 우선

$y_i^{TL} < y_j^{TL}$:O_i가 O_j 객체보다 우선

O_i는 i번째 객체를 말하고, x_i^{TL}는 i번째 객체의 TopLeft x 좌표를, y_i^{TL}는 i번째 객체의 TopLeft y 좌표를 말한다.

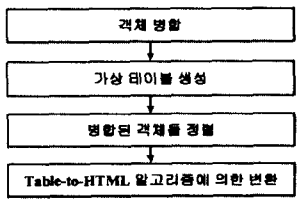


그림 1. <Table> 태그를 이용한 변환 단계

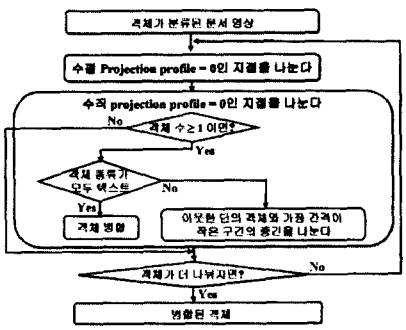
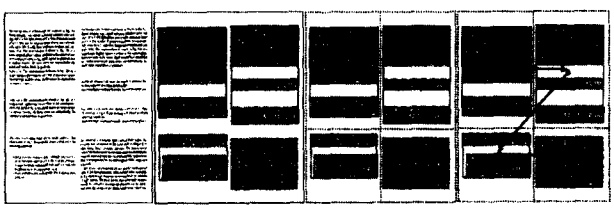


그림 2. 객체들 병합 단계



(a) 원본 문서 (b) 객체 병합 (c) 가상라인 생성 (d) 순서 정렬

그림 3. 객체 병합의 예

변환 알고리즘을 적용하기 전 단계까지를 수행한 결과가 그림 3와 같고, 다음은 마지막 단계에 필요한 테이블 변환(Table-to-HTML) 알고리즘을 보여준다. $Cell[i][j]_T$와 $Cell[i][j]_B$, $Cell[i][j]_L$, $Cell[i][j]_R$들은 각각 테이블 i번째 행의 j번째 열에 위치하는 셀의 Top, Bottom, Left, Right 라인을 말한다.

---

if $Cell[i][j]_T$ and $Cell[i][j]_L = 1$,
  (1) $Cell[i][j]_R$값 검사
    if $Cell[i][j]_R = 1$, then Colspan = 1
    else { j와 Colspan값 1씩 증가,
      $Cell[i][j]_R = 1$이 나올 때까지 검사 }
  (2) $Cell[i][j]_B$값 검사
    if $Cell[i][j]_B = 1$, then Rowspan = 1
    else { i와 Rowspan값 1씩 증가,
      $Cell[i][j]_B = 1$이 나올 때까지 검사 }
else 다음 셀로 이동

2.2 객체의 형태와 위치 정보를 이용한 변환

이 장에서는 다단 문서를 표현하기 위해 원본 문서의 크기를 사용자 스크린 크기에 알맞게 변경하여 문서 객체들의 실제 좌표에 해당되는 스크린의 좌표들을 지정해 줌으로써 다단 문서를 변환하는 방법이다. 여기서는 각 객체들이 상하좌우의 다른 객체들과 겹치거나 실제 이상으로 멀리 떨어지는 문제를 방지하기 위해 각 객체마다 가질 수 있는 속성들을 정확히 뽑아 내고 적용시키는 것이 중요하다. $C = D * \{S_{length}/D_{length}\}$. 이 식이 문서 전체의 크기 변경 뿐만 아니라 각 객체 및 글자 크기의 변경 등에도 일반적으로 사용되는 변환식이다. C는 변환될 객체의 가로 또는 세로 길이를 말하고, D는 원본 문서 객체들의 가로 또는 세로 길이를 말한다. $\{S_{length}/D_{length}\}$이 변환 비율을 나타낸 것으로써 S_{length}는 스크린의 가로 또는 세로 길이이고, D_{length}는 원본 문서의 가로 또는 세로 길이를 나타낸 것이다.

이렇게 크기를 변경한 다음, 글자 크기와 줄 간격, 들여쓰기, 정렬 상태 등 객체들이 가질 수 있는 속성들을 정확히 추출해야 한다. 다음은 <Layer> 태그를 사용할 때 가장 중요한 속성들을 추출하는 방법이다. WS는 텍스트 라인들 사이의 여백들을 말하고, O_H^X는 텍스트 객체의 높이를, B_{i-1}^{ex}는 i-1번째 텍스트 객체의 한 라인 블록의 오른쪽 x 좌표를 말한다. 그리고 C_w는 한 문자의 width를, B_N은 블록 라인의 갯수를 말한다.

- 줄 간격과 글자 높이(B_H): $(B_H * B_N) + WS = O_H^X$
- 들여쓰기가 들어가는 새로운 라인: $B_{i-1}^{ex} + C_w < B_i^{ex}$

3. 구조화된 하이퍼문서의 자동 생성

본 논문에서는 서로 관련된 여러 장의 문서를 변환시, 이들의 섹션 제목들을 뽑아 내고 계층화시켜 구조화된 목차 페이지를 자동 생성한다. 그러기 위해서는 문서 객체들에 대한 논리적인 구조 분석이 선행되어야 한다. 문서 영상의 구조 분석 결과물인 텍스트 영역에 대한 논리적 구조 분석을 수행하여 페이지 번호와 캡션, 섹션 제목들을 추출하고 레이블링한 후, 그 중 섹션 제목 후보들만을 뽑아낸다. 여기에 할당된 숫자들을 정렬하여 계층화하고 이를 동일 페이지에 속하는지 그렇지 않은지를 검사함으로써 첫 번째 확인 작업을 거치고, 정렬된 제목의 단계별로 각 객체 정보를 비교하는 두번째 확인 작업을 거쳐 섹션 제목들을 최종적으로 계층화한다.

3.1 객체 레이블링

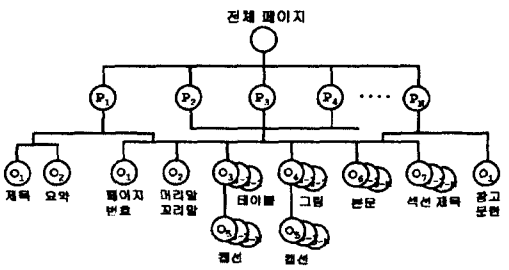


그림 4. 논문의 논리적 계층 구조 정의

본 논문에서는 일반적인 논문들의 논리적 계층 구조를 그림 4처럼 정의하였다. 계층화된 목차 페이지를 생성하기 위해서 페이지 번호, 캡션, 섹션 제목만을 다음과 같이 레이블링한다.

- 페이지 번호: 문서의 맨 위나 맨 아래의 텍스트 객체 중 라인수가 1이고 숫자로만 구성된 객체
- 캡션: 그림이나 테이블의 바로 위아래로 인접한 텍스트 객체이면서 fig, Figure, 그림, 테이블, Table 등의 키워드로 시작하는 객체
- 섹션 제목 후보: 라인 수가 임계치 이하인 텍스트 객체

섹션 제목의 후보로 라인 수가 적은 텍스트 객체를 모두 뽑아내어 앞에서부터 10자리만을 [숫자+기호] 패턴으로 변경하고, 몇 가지 조건을 비교하여 제목들을 계층화한다. 이 조건들은 다음 장에서 설명된다.

3.2 섹션 제목에 할당된 숫자들 정렬

그림 5는 논문의 임의의 한 페이지에서 섹션 제목 후보들로 나온 객체들을 [숫자+기호] 패턴으로 10자리만을 표현한 것이다. 아래에 번호를 붙인 순으로 정렬하여 그림 6처럼 단계별로 트리를 구성하는 것이 계층화의 첫번째 단계이다.

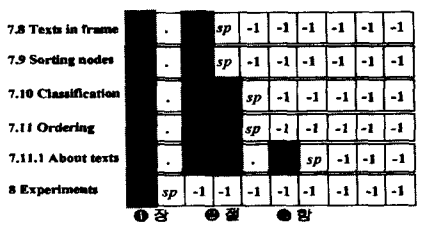


그림 5. 섹션 제목 후보들을 뽑은 예

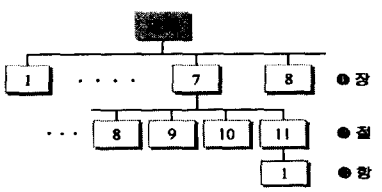


그림 6. 섹션 제목 단계별로 구성된 트리

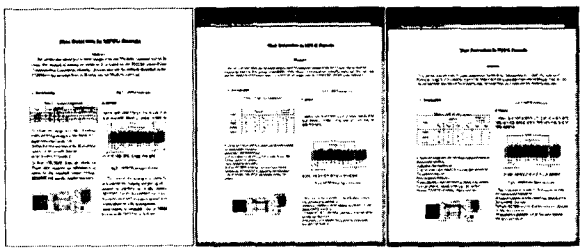
계층화의 두번째 단계는 그림 6에서 각 단계의 객체 정보들을 서로 비교하여 동일하지 않은 객체를 위나 아래 단계의 다른 객체 정보와 비교하여 단계를 조정함으로써 섹션 제목들을 최종적으로 계층화한다.

4. 실험 결과 분석 및 토의

본 논문에서는 잡지나 광고지, 신문 등과 같은 비정형화된 복잡한 다단 문서를 대상으로 제안된 변환 알고리즘을 적용해 보았다. 변환된 정도를 정량적으로 평가할 수 있는 기준이 모호하나, 원본 문서의 형태와 구조가 표현상 유사하게 변환됨을 확인할 수 있었다.

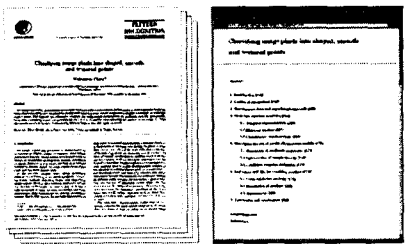
그림 7은 <Table> 태그와 <Layer> 태그를 이용해 복잡한 다단 문서를 변환한 HTML 문서 영상을 보여준다. (b)와 (c) 모두 약간의 차이는 있으나, 원본 문서와 유사하게 변환되었음을 알 수 있다.

그림 8은 임의의 논문 한 편을 변환하면서 자동으로 생성된 목차 페이지를 보여준다. 입력 문서의 각 페이지들은 그림 7의 (b), (c)와 같이 모두 변환된다.



(a) 원본 문서 (b) <Table> 태그 이용 (c) <Layer> 태그 이용

그림 7. 다단 문서의 변환 예



(a) 원본 문서 (b) 구조화된 목차 페이지

그림 8. 구조화된 하이퍼문서 생성의 예

<Table> 태그와 <Layer> 태그를 사용했을 때의 차이점은 다음과 같다. <Table> 태그를 사용하는 경우, 단과 단 사이에 들어가는 객체들을 표현할 수 없고, 그림이나 도표의 캡션이 객체 병합 과정에서 분리되어 논리적으로 깊은 연관 관계가 있는데도 불구하고 너무 멀게 위치하는 경우가 발생할 수 있다는 단점을 가지나, 객체들이 서로 겹치는 문제는 일어나지 않는다. 반면에 <Layer> 태그는 <Table> 태그와는 반대로 텍스트 객체의 줄 간격이나 글자 크기 등을 잘못 계산하면 객체들이 서로 겹치거나 원본 문서의 간격보다 터무니 없이 멀리 떨어지는 경우가 생길 수 있으나, 단과 단 사이에 들어가는 객체나 매우 복잡하다고 여겨지는 다양한 문서의 변환이 가능하다는 장점을 가진다. 어느 하나의 태그만으로도 모든 경우의 문서를 제대로 변환하지 못하므로 다단 문서의 변환이 어려우나, 다양한 태그의 활용과 전처리 작업으로 더욱 자연스럽고 정확하게 복잡한 다단 문서의 변환이 이루어져야 할 것이다.

참고 문헌

[1] M. Worring and A. W. M. Smeulders, "Content based internet access to paper documents," *IJDAR*, Vol. 1 No. 4, pp. 209-220, 1999.

[2] T. Tanaka and S. Tsuruoka, "Table Form Document Understanding Using Node Classification Method and HTML Document Generation," *Proc. of DAS'98*, Nagano, pp..157-158, 1998.

[3] 이 지연, 이 성환, "도표를 포함하는 다단 문서 영상의 HTML 자동 변환 알고리즘," 한국정보과학회 봄 학술발표회 논문집, 목포, pp. 600-602, 1999년 4월.

[4] C. Faure, "Preattentive Reading and Selective Attention for Document Image Analysis," *Proc. of 5th ICDAR*, Bangalore, pp. 577-580, 1999.