구조화된 문서 생성을 위한 논리적인 구조 분석 기법: 구문론적인 접근 방식

이경호° 최윤철 조성배
연세대학교 컴퓨터과학과
{lkh, ycchoy}@rainbow.yonsei.ac.kr, shcho@csai.yonsei.ac.kr

Logical Structure Analysis for Structured Document Generation: A Syntactic Approach

Kyong-Ho Lee° Yoon-Chul Choy Sung-Bae Cho
Dept. of Computer Science, Yonsei University

요 약

본 논문에서는 다수의 페이지로 구성된 복잡한 구조의 문서로부터 SGML/XML에 기반한 전자 문서를 생성하기 위한 구문론적인 구조분석 방법을 제안한다. 제안된 방법은 구조분석의 정확성과 처리 속도의 향상을 위하여 텍스트 영역의 계층적인 트리를 파싱하여 논리적인 계층 구조를 추출한다. 또한 본 논문은 문서 유형의 논리적인 구조 정보와 기하적인 특성을 효과적으로 기술할 수 있는 문서 모델을 정의한다. 제안된 방법의 성능을 평가하기 위하여 IEEE Transactions on Pattern Analysis and Machine Intelligence로부터 스캐닝한 372개의 논문 영상으로 실험한 결과, 제안된 방법은 기존 연구와 달리 다수의 문서 영상으로 구성된 문서에 대하여 논리적인 구조분석을 효율적으로 지원하였다. 특히 제안된 방법은 논리적인 구조분석의 최종 결과로서 SGML/XML 문서를 생성하기 때문에 문서의 재사용성과 호환성을 높인다.

1. 서론

전자 문서의 활발한 보급에도 불구하고 종이 문서의 양도 급속도로 증가하고 있다. 이는 인간이 기본적으로 종이 형태의 문서를 선호한다는 것을 반영하는 것이다. 그러나 종이 문서는 전자 문서와 비교하여 저장, 검색, 갱신, 그리고 전송 등의 문서 처리의 다양한 면에 있어서 비효율적이다. 이에 종이 문서를 전자 문서로 자동 변환하는 방법의 개발은 매우 중요하다. 한편 SGML(standard generalized markup language) [1]과 XML (extensible markup language) [2]은 논리적인 계층 구조를 표현할 수 있으며 이 기종간의 호환이 가능하다는 장점 때문에 다양한 분야에서 전자 문서의 표준 포맷으로 자리잡았다. 따라서 문서 영상으로부터 논리적인 구성 요소를 추출하여 SGML/XML 문서를 생성하는 논리적인 구조분석 방법의 개발이 절실히 요구된다.

일반적으로 인간은 문서를 구성하는 텍스트 영역의 기하적인 특성으로부터 제목 또는 단락 등의 논리적인 구성 요소를 식별하고, 이를 병합하여 절 구조와 같은 복합적인 구성 요소를 식별함으로써 문서의 논리적인 계층 구조를 인식한다. 이와 같이 텍스트 영역의 기하적인 특성으로부터 직접적인 식별이 가능한 논리적인 요소를 주 구조(primary structure)라고 하며 이미 식별된 다수의 구성 요소들을 병합함으로써 추출 가능한 구성 요소를 부 구조(secondary structure)라고 한다 [3].

따라서 문서 영상으로부터 SGML/XML 에 기반한 전자 문서를 자동 생성하기 위해서는 주 구조는 물론이고 부 구조에 대한 논리적인 구조분석이 이루어져야 한다. 그러나 기존 연구의 대부분은 주 구조에 해당하는 구성 요소만을 추출하기 때문에 계층적인 구조 정보를 생성하지 못한다. 또한 주로 단일의 문서 영상을 처리 대상으로 하기 때문에 다수의 문서 영상으로 구성된 복잡한 구조의 문서를 지원하지 않는다 [4], [5].

한편 문서 영상의 논리적인 구조분석을 위해서는 문서 유형에 대한 지식을 표현한 문서 모델이 요구된다. 문서 영상으로부터 논리적인 계층 구조의 효과적인 추출을 위해서는 문서 유형의 기하적인 특성과 논리적인 계층 구조에 대한 다양한 정보를 표현할 수 있는 문서 모델이 요구된다.

따라서 본 논문에서는 다수의 문서 영상으로 구성된 복잡한 구조의 문서를 대상으로 논리적인 구조분석을 위한 구문론적인 방법을 제안한다. 일반적으로 문서 영상을 구성하는 텍스트 영역은 각각 제목 또는 단락 등에 해당하는 헤더(header) 또는 바디(body)로써의 기능을 한다. 특히 헤더와 바디는 해당 텍스트 영역의 기하적인 특성에 따라 다양한 종류로 구별된다. 본 논문에서는 헤더와 바디를 문서의 기능적인 구성 요소로 정의하고, 다양한 종류의 헤더와 바디로 구성된 계층적인 트리 구조를 기능 구조 트리(functional structure tree)라고 정의한다.

제안된 방법은 구조분석의 정확성과 처리 속도의 향상을 위하여 기존 연구와 달리 텍스트 라인이 아닌 기능 구조 트리에 제안된 문서 모델에 기반한 파싱 기법을 적용하여 각각의 노드에 레이블이 부여된 논리 구조 트리(logical structure tree)를 생성한다. 한편 본 논문에서는 문서 모델을 효율적으로 표현할 수 있는 언어인 DSDL (document structure description language)을 제안한다. DSDL 은 논리적인 계층 구조를 기술하기 위하여 문서 유형이 포함할 수 있는 주 구조의 기하적인 특성은 물론이고 부 구조가 포함할 수 있는 구성 요소의 종류, 순서 그리고 빈도수를 기술한다.

제안된 방법의 성능을 평가하기 위하여 IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)로부터 스캐닝한 372 개의 논문 영상으로 실험한 결과, 제안된 방법은 기존 연구와 달리 다수의 영상으로 구성된 문서에 대하여 논리적인 구조분석을 효율적으로 지원하였다. 특히 제안된 방법은 논리적인 구조분석의 최종 결과로서 SGML/XML 문서를 생성하기 때문에 문서의 재사용성과 호환성을 지원한다.

본 논문의 구성은 다음과 같다. 2 절에서는 문서 모델을 기술하기 위하여 정의된 DSDL 을 자세히 설명한다. 3 절에서는 제안된 구조분석 방법을 자세히 기술한다. 4 절에서는 실험 결과를 통하여 제안된 방법의 성능을 기존 연구와 비교 및 분석한다. 마지막으로 5 절에서는 결론 및 향후 연구 방향을 기술한다.

2. 문서 모델

DSDL 은 제안된 논리 구조 트리에서 중간 노드에 해당하는 부 구조 각각에 대하여 이에 포함될 수 있는 구성 요소의 종류, 순서, 그리고

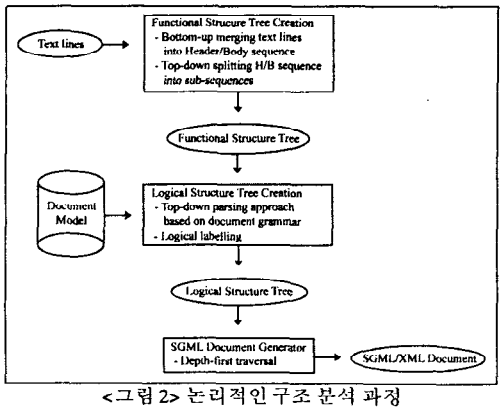
반복 횟수 등에 대한 정보를 정규 수식으로 표현한다. 본 논문에서는 이를 해당 부 구조의 내용 모델(content model)이라고 정의한다. 또한 주 구성 요소 각각에 대하여 이를 만족하는 텍스트 영역의 기하적인 조건으로서 단 유형, 텍스트 라인의 숫자와 높이, 텍스트 영역의 전후 여백, 검은 화소의 밀도 분포, 그리고 정렬 방식 등에 관한 특성을 기술한다. DSDL 은 <그림 1>과 같이 기호 "#"을 사용하여 부 구조의 내용 모델과 주 구조의 기하적인 특성 정보를 구별한다. 특히 본 논문에서는 DSDL 에 의하여 기술된 문서 모델을 문서 문법(document grammar)이라고 정의한다.

```
<!ELEMENT Document      (Title, Author, Affil, Abstract, Keyword, Sec-Body)>
<!ELEMENT Title         #( FUNTION_TYPE: HEADER
                           COLUMN_TYPE: SINGLE
                           MIN_LINE_HEIGHT: 31
                           MAX_LINE_HEIGHT: 42
                           MIN_LINE_NUMBER: 1
                           MAX_LINE_NUMBER: 3
                           MIN_SPACE_BEFORE: 105
                           MAX_SPACE_BEFORE: 120
                           MIN_SPACE_AFTER: 60
                           MAX_SPACE_AFTER: 70
                           MIN_BLACK_PIXEL_DENSITY: 0.307
                           MAX_BLACK_PIXEL_DENSITY: 0.525
                           JUSTIFY: CENTER)>
<!ELEMENT Author        #( ... )>
<!ELEMENT Affil         #( ... )>
<!ELEMENT Abstract      #( ... )>
<!ELEMENT Keyword       #( ... )>
<!ELEMENT Sec-Body      (Section*, Reference )>
<!ELEMENT Section       (Sec-Header, Paragraph*, Sub-Section*)>
<!ELEMENT Sub-Section   (Sub-Sec-Header, Paragraph*, Sub-Sub-Section*)>
<!ELEMENT Sub-Sub-Section (Sub-Sub-Sec-Header, Paragraph*)>
<!ELEMENT Reference     (Sec-Header, Ref-Item*)>
<!ELEMENT Sec-Header    #( ... )>
<!ELEMENT Sub-Sec-Header #( ... )>
<!ELEMENT Sub-Sub-Sec-Header #( ... )>
<!ELEMENT Paragraph     #( ... )>
<!ELEMENT Ref-Item      #( ... )>
```

<그림 1> 문서 모델의 예

3. 구문론적인 구조 분석 방법

제안된 논리적인 구조분석 방법은 <그림 2>와 같이 기능 구조 트리의 생성, 논리 구조 트리의 생성, 그리고 SGML 문서 생성의 세 단계로 구성된다.



<그림 2> 논리적인 구조 분석 과정

3.1 기능 구조 트리의 생성

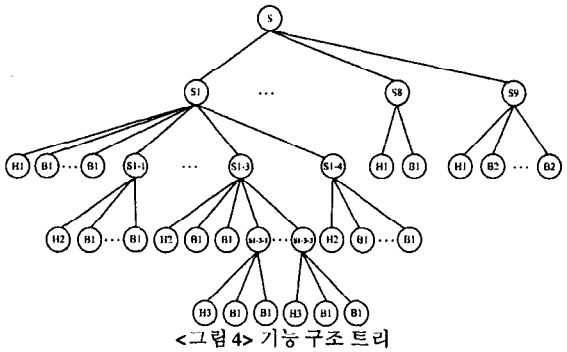
본 절에서는 텍스트 라인의 집합으로부터 기능 구조 트리를 생성하는 방법을 기술한다. 먼저 기하적인 특성이 유사한 인접한 텍스트 라인을 병합하여 헤더와 바디의 순차적인 집합을 생성한다. 이를 위하여 형태 심리학(gestalt psychology) [6]에서 사용하는 세 가지의 일반적인 원칙을 적용한다. 즉, 근접성(proximity)의 원칙에 따라 서로 다른 객체 사이의 줄 간격은 동일한 객체에 속하는 텍스트 라인 사이의 줄 간격 보다 크다. 또한 유사성(similarity)의 원칙에 따라 동일한 객체에 속하는 텍스트 라인의 기하적인 특성은 서로 유사하다. 마지막으로 연속성(contiguity)의 원칙을 적용하여 서로 다른 종류의 단 영역에 속하는 텍스트 라인은 구별된다. 예를 들어, 실험에 사용된 TPAMI 의 경우, 일단 영역은 논문 제목, 저자, 요약, 그리고 키워드를 포함하며 이단 영역은 계층적인 중첩 구조의 절과 참고 문헌을 포함한다.

추출된 헤더와 바디 각각은 이를 구성하는 텍스트 라인의 기하적인 특성에 따라 서로 다른 종류로 구별된다. 예를 들어, 문서의 제목과 절 제목은 서로 다른 종류의 헤더로 분류되며 요약과 단락 역시 서로 다른 종류의 바디로 구별된다. 예를 들어, TPAMI 의 경우, 논문 제목, 저자, 키워드, 그리고 다단계의 절 제목은 서로 다른 종류의 헤더로 구별되며 요약, 단락, 그리고 참고 문헌은 서로 다른 바디로 분류되었다.

제안된 방법은 헤더와 바디의 순차적인 집합으로부터 계층 구조의 기능 구조 트리를 생성하기 위하여 문서를 구성하는 구조의 반복적인 특성을 이용한다. 일반적으로 문서의 본문은 다수의 절을 포함하며 각각의 절은 계층적으로 중첩된 하위 레벨의 절로 구성된다. 또한 각각의 절 구조는 절 제목에 의하여 식별되며 하위 레벨의 절 제목은 이를 포함하는 상위 레벨의 절 제목보다 뒤에 위치한다.

S = { H1, B1, B1, B1, B1, B1, H2, B1, B1, B1, B1, H2, B1, B1, B1, H3, B1, B1, B1, H3,
B1, B1, B1, H2, B1, B1, H3, B1, B1, H3, B1, B1, B1, H3, B1, B1, H2, B1, B1, B1,
H1, B1, H2, B1, H3, B1, B1, B1, H3, B1, B1, B1, B1, H3, B1, B1, B1, H2, B1, B1,
H3, B1, B1, B1, H3, B1, B1, B1, H3, B1, B1, B1, B1, H3, B1, H2, B1, B1, H1, B1,
B1, H2, B1, B1, H3, B1, B1, H3, B1, B1, B1, H2, B1, B1, B1, B1, B1, H3, B1, B1,
B1, B1, H3, B1, B1, H3, B1, B1, B1, H3, B1, B1, B1, H2, B1, H1, B1, B1, B1, B1,
H1, B1, B1, B1, H2, B1, B1, B1, H2, B1, B1, B1, H2, B1, H2, B1, B1, B1, H3, B1,
H1, B1, B1, B1, H1, B1, B1, B1, H1, B1, H1, B2, B2, B2, B2 }

<그림 3> 헤더와 바디의 순차적인 집합



<그림 4> 기능 구조 트리

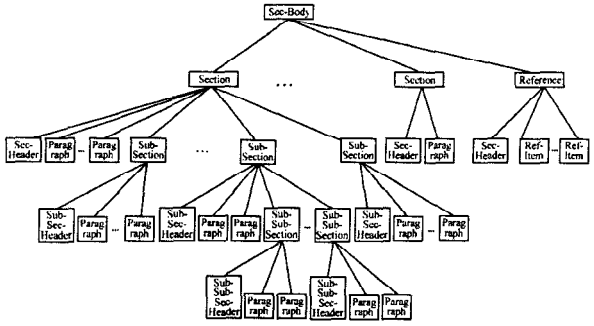
따라서 제안된 방법은 반복되는 헤더를 기준으로 순차적인 집합을 반복 분할하면서 기능 구조 트리를 하향식으로 생성한다. 예를 들어, TPAMI 에 속하는 논문의 본문 영역으로부터 추출된 헤더와 바디의 순차적인 집합과 이로부터 생성된 기능 구조 트리는 각각 <그림 3>과 <그림 4>와 같다. 제안된 방법은 가장 먼저 식별된 반복적인 헤더를 기준으로 순차적인 집합을 분할한다. 이와 같이 집합을 반복적으로 분할하는 것은 해당 집합을 노드로 하는 트리 구조를 하향식으로 확장해 가는 과정에 해당한다.

3.2 논리 구조 트리의 생성

제안된 방법은 기능 구조 트리에 문서 문법에 기반한 제안된 파싱 기법을 적용하여 논리 구조 트리를 생성한다. 제안된 방법은 기능 구조 트리를 하향식 깊이 우선 탐색하면서 기능 구조 트리를 구성하는 중간 노드의 계층 구조와 단말 노드의 기하적인 특성이 문서 모델의 해당 조건을 만족하는지의 여부를 검사한다. 제안된 파싱 기법은 뿌리 노드(root node)를 포함한 중간 노드에 대하여 해당 노드의 자식 노드가 단말 노드 또는 중간 노드인지의 여부에 따라 적절히 처리한다. 먼저 자식 노드가 단말 노드인 경우, 자식 노드의 기하적인 특성을 만족하는 엘리먼트의 이름을 레이블로 부여한다. 자식 노드가 중간 노드인 경우, 허용 가능한 엘리먼트의 이름을 부여한 후 해당 엘리먼트의 내용 모델이 적합한지의 여부를 검사한다.

만일 중간 노드의 파싱 과정에서 자식 노드를 만족하는 엘리먼트 선언이 존재하지 않을 경우, 제안된 방법은 중간 노드에 선택 가능한 또 다른 분법을 적용하기 위하여 백트래킹(backtracking)한다. 예

를 들어, <그림 4>에서 중간 노드 S 의 자식 노드 S9 에 <그림 1>의 문서 모델에서 엘리먼트 Sec-Body 의 문법을 적용한다고 하자. 이때 레이블 *Section* 과 *Reference* 의 적용이 가능하다. 따라서 제안된 방법은 먼저 레이블 *Section* 을 부여한 후, S9 의 자식 노드가 *Section* 의 내용 모델을 만족하는지의 여부를 검사한다. 그러나 해당 내용 모델은 두 번째 단말 자식 노드인 B2 의 기하적인 특성을 만족하는 엘리먼트를 포함하지 않는다. 따라서 제안된 방법은 부모 노드 S9 로 백트래킹하여 허용 가능한 또 다른 엘리먼트인 *Reference* 의 내용 모델을 적용한다. 예를 들어, <그림 1>의 문법을 기반으로 <그림 4>의 기능 구조 트리에 하향식 파싱 기법을 적용하여 생성된 논리 구조 트리는 <그림 5>와 같다.



<그림 5> 논리 구조 트리

3.3 SGML/XML 문서의 생성

본 논문은 논리적인 구조분석의 최종 결과로서 SGML/XML 문서를 생성한다. 일반적으로 사용자가 문서를 처음부터 끝까지 읽어가는 순서는 논리 구조 트리를 깊이 우선 탐색하는 과정으로 볼 수 있다. 제안된 방법은 논리 구조 트리를 구성하는 단말 노드와 중간 노드 각각에 대하여 서로 다른 방법을 적용하여 SGML/XML 문서를 생성한다. 먼저 논리 구조 트리를 깊이 우선 탐색하면서 중간 노드를 만나면 해당 레이블을 이름으로 갖는 엘리먼트의 시작 태그를 출력하며 해당 노드를 벗어날 때 끝 태그를 출력한다. 한편 단말 노드는 문서의 텍스트 영역과 직접적인 대응 관계를 갖는 주 구조에 해당한다. 따라서 단말 노드를 만나면 먼저 해당 레이블의 이름을 갖는 엘리먼트의 시작 태그를 출력하고, 해당 텍스트 영역의 문자 인식 결과와 끝 태그를 출력함으로써 엘리먼트를 생성한다.

4. 실험 결과 및 성능 분석

제안된 방법의 성능을 평가하기 위하여 1999 년 1 월부터 6 월 사이에 발행된 TPAMI 에 속하며 372 개의 논문 영상으로 구성된 정규 논문 26 편을 대상으로 실험하였다. 제안된 구조분석 시스템은 다수의 페이지로 구성된 각각의 논문에 속하는 텍스트 라인의 집합을 입력으로 받아들인다. 본 논문에서는 구조분석 방법의 정확성과 처리 속도의 두 가지 측면에서 제안된 방법의 성능을 분석한다.

제안된 구조분석 방법은 기능 구조 트리의 생성과 파싱의 두 단계로 구성되며 기능 구조 트리의 생성은 헤더와 바디의 식별 그리고 계층적인 구조의 생성의 두 단계로 구성된다. 따라서 본 논문에서는 구조분석의 정확성을 평가하기 위하여 헤더와 바디의 식별률, 계층 구조 생성의 정확성, 그리고 논리적인 객체의 식별률의 세 가지 평가 기준을 제안한다. 헤더와 바디의 식별률 면에 있어서, 제안된 방법은 시스템의 입력으로 텍스트 라인을 받아들이는 기존 연구 [7]보다 나은 결과를 보였다. 이는 제안된 방법이 기능적인 구성 요소의 식별을 위하여 다양한 종류의 기하적인 특성과 이웃한 페이지를 모두 고려하기 때문이다. 예를 들어, 단락은 인접한 페이지에 분할되어 위치하는 경우가 다수 존재하며 절 제목은 줄 간격은 물론이고 밀도 등의 다양한 종류의 기하적인 특성에 의하여 구별된다. 한편 제안된 방법은 헤더의 반복적인 특성을 이용하여 계층적인 구조 정보를 추출하기 때문에 헤더의 정확한 식별은 매우 중요하다. 제안된 시스템은 전체 실험 데이터에 대하여 파싱 오류 없이 레이블링을 수행하였다. 실험 결과, 논리적인 객체의 식별 오류는 모두 헤더와 바디의 식별 오류에 기인하였으며 정확히 식별된 헤더와 바디에 잘못된 레이블이 부여된 경우는 존재하지 않았다.

한편 제안된 파싱 기법은 처리 속도의 향상을 위하여 텍스트 라인이 아닌 기능적인 구성 요소인 헤더와 바디를 처리 대상으로 한다. 일반적으로 구문론적인 방법은 빈번한 백트래킹을 허용하기 때문에 처리 시간의 대부분이 파싱 과정에 소요된다. 본 논문에서는 전체 실험 영상을 대상으로 텍스트 라인과 기능적인 구성 요소의 빈도수를 조사하였다. 조사 결과, 텍스트 라인과 기능적인 구성 요소의 빈도수는 각각 24,123 과 3,043 으로서 1 : 0.126 의 비율을 나타났다. 따라서 제안된 구조분석 방법은 파싱의 기본 단위가 화소 또는 텍스트 라인인 방법보다 처리 속도가 보다 빠르다.

5. 결론 및 향후 연구 방향

SGML/XML 은 논리적인 구조 정보를 표현할 수 있으며 이 기종간의 호환이 가능하다는 장점 때문에 전자 문서의 표준 포맷으로 널리 사용되고 있다. 따라서 본 논문에서는 다수의 문서 영상으로 구성된 복잡한 구조의 문서로부터 SGML/XML 에 기반한 전자 문서를 생성하기 위한 구문론적인 구조분석 방법을 제안한다.

특히 제안된 구조분석 방법은 처리 속도의 향상을 위하여 계층적인 구조의 기능 구조 트리에 기반한다. 이를 위하여 형태 심리학의 세 가지 원칙을 적용하여 기능적인 구성 요소를 추출하고, 이로부터 헤더의 반복적인 특성을 적용하여 기능 구조 트리를 하향식으로 생성한다. 또한 구조분석을 정확성을 위하여 문서 모델을 효율적으로 표현할 수 있는 언어인 DSDL 을 제안한다. DSDL 은 논리적인 계층 구조를 기술하기 위하여 문서 유형이 포함할 수 있는 주 구조의 기하적인 특성은 물론이고 부 구조가 포함할 수 있는 구성 요소의 종류와 순서 그리고 빈도수 등에 대한 다양한 정보를 기술한다.

제안된 구조분석 방법은 기능 구조 트리를 하향식 깊이 우선 탐색하면서 DSDL 로 기술된 문법을 적용하여 논리 구조 트리를 생성한다. 제안된 방법의 성능을 분석하기 위하여 다양한 종류의 문서 영상에 실험한 결과, 기존 연구와 달리 다수의 문서 영상으로 구성된 문서에 대하여 논리적인 구조분석을 효율적으로 지원하였다. 특히 제안된 방법은 논리적인 구조분석의 최종 결과로서 SGML/XML 문서를 생성하기 때문에 문서의 재사용성과 호환성을 높인다.

참고 문헌

[1] International Organization for Standardization, Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML), *ISO/IEC 8879*, 1986.

[2] World Wide Web Consortium, Extensible Markup Language (XML) 1.0, http://www.x3c.org/TR/1998/REC-xml-19980210, 1998.

[3] K. M. Summers, "Toward a Taxonomy of Logical Document Structures," *Proc. Dartmouth Institute for Advanced Graduate Studies (DAGS'95)*, pp. 124 ~ 133, Boston, May 1995.

[4] A. Dengel, R. Bleisinger, R. Hoch, F. Fein, and F. H nes, "From Paper to Office Document Standard Representation," *IEEE Computer*, Vol. 25, No. 7, pp. 63 ~ 67, Jul. 1992.

[5] S. Tsujimoto and H. Asada, "Major Components of a Complete Text Reading System," *Proc. IEEE*, Vol. 80, No. 7, pp. 1133 ~ 1149, Jul. 1992.

[6] K. Koffka, *Principles of Gestalt Psychology*. Harcourt, Brace and World, New York, 1935.

[7] D. Rus and K. Summers, "Geometric Algorithms and Experiments for Automated Document Structuring," *Mathematical and Computer Modelling*, Vol. 26, No. 1, pp. 55 ~ 83, 1997.