다양한 유형의 서식문서 처리를 위한 효과적인 모형 기반 방법에 관한 연구*

변영철○ 이일병
연세대학교 컴퓨터과학과

Efficient Model-based Form Processing Methods for Various Kinds of Form Documents

요 약

본 논문에서는 여러가지 유형의 서식문서를 효과적으로 처리하기 위한 방법을 제안하고 모형 기반 서식 처리 시스템을 위한 프레임워크를 구현한다. 이를 위해 서식문서의 모형으로 등록되는 정보로서 네가지 유형의 서식문서에 관한 지식을 정의하고, 이를 기술하기 위한 서식 기술 언어를 정의한다. 먼저, 서식 등록 과정에서 서식에 관한 네가지 유형의 지식을 서식 모형으로 등록한다. 그리고 서식 처리 과정에서 시스템에 등록되어 있는 서식 모형을 이용하여 서식을 분류한 후 처리하고자 하는 항목을 인식하고 추출한다. 서식의 지식을 이용하여 서식 구조를 인식하고 부분적 하향식 비교 방법을 이용하여 서식을 분류함으로써 계산 시간을 줄일 수 있다. 실험결과 8개의 서식 모형이 등록되어 있을 경우에는 평균 서식 분류 시간은 0.74초였으며, 5개 혹은 6개의 항목을 추출하는데 걸리는 시간은 평균 0.45초였다. 본 방법은 서식 영상의 질이 좋지 않을 경우에도 잘 동작함은 물론 서식 모형만 추가함으로써 다른 서식 문서도 쉽게 처리할 수 있다.

1. 서 론

서식문서는 특정 목적을 위한 만들어진 특별한 형태의 문서이다. 오늘날 사용되는 서식문서에는 세금 계산서, 은행 입출금표, 신용카드 매출표 등 수많은 유형의 서식들이 존재하는데, 이러한 문서들은 대개의 경우 수작업으로 처리되고 있다. 수작업에 의한 서식문서의 처리는 매우 지루한 작업일 뿐만 아니라 비용이 많이드는 작업이기도 하다. 따라서 다양한 분야에서 컴퓨터를 이용함으로써 서식문서를 자동으로 처리할 수 있는 방법이 요구되고 있다. 컴퓨터를 이용하여 서식을 자동으로 처리하기 위해서는 일반적으로 서식 분석(form analysis) 단계와 서식 이해 단계(form understanding)가 필요하다. 서식분석 단계에서는 서식문서의 구조를 인식하고 해당 서식을 이해하는데 필요한 항목 영상들을 추출한다. 서식 이해 단계에서는 추출한 항목 영상을 인식함으로써 서식으로부터 정보를 추출한다[1]

서식 처리 시스템은 특정 서식만을 처리할 경우 해당 서식의 지식을 이용하여 비교적 쉽게 처리할 수 있다. 그러나 여러가지 서식문서를 대상으로 할 경우에는 해결해야 할 문제가 많다. 가령, 동일한 유형의 서식문서라 할지라도 작성자에 따라 서식의 구조가 조금씩 다를 수도 있으며, 필요에 따라 서식의 구조가 변할 수도 있기 때문이다. 또한, 서식문서 영상의 크기는 개별 문자의 크기에 비해 상당히 크기 때문에 서식 문서 처리시 많은 시간이 요구된다. 한편, 여러가지 유형의 서식문서 처리시 서식문서 처리 방법의 일반성만을 강조하다보면 서식문서를 효과적으로 처리하기가 어려우며, 보다 중요한 것은 효과적으로 타당한 시간동안에 서식을 처리할 수 있어야 한다.

본 논문에서는 서식에 관한 최소한의 정보를 서식 모형으로 정의한 후 이를 이용하여 다양한 유형의 서식문서를 효과적으로 처리할 수 있는 모형 기반 서식 처리 시스템에 관하여 설명한다. 다음 장에서는 다양한 서식문서를 처리하기 위한 서식 처리 시스템 프레임워크에 대해 설명한다. 3장에서는 네가지 유형의 서식 지식에 대해 설명하고 그러한 서식 지식을 기술하기 위한 서식 기술 언어를 설명한다. 4장에서는 서식 등록 도구를 이용한 서식 모형의 등록 방법에 대해 기술한다. 서식 구조 인식 및 분류 방법에 대해서는 5장에서 설명한다. 그리고 6장에서는 실험 결과에 대해 설명하며, 마지막으로 7장에서는 본 논문에 대해 결론을 맺는다.

2. 서식 처리 시스템 프레임워크

〈그림 1〉는 다양한 유형의 서식을 처리하기 위한 프레임워크이다. 이 프레임워크는 크게 서식 등록 단계와 서식 처리 과정으로 동작한다. 먼저 서식 등록 과정에서는 입력 서식에 대한 지식을 추출하여 서식의 모형으로 등록한다. 서식 모형을 구성하는 서식의 지식에는 크게 네가지 유형이 있다. 즉, 서식 구조에 관한 지식, 서식 특징에 관한 지식, 항목 유형 및 영역에 관한 지식, 그리고 항목 문맥에 관한 지식이 그것이다. 서식 처리 과정에서는 서식 모형 데이터베이스에

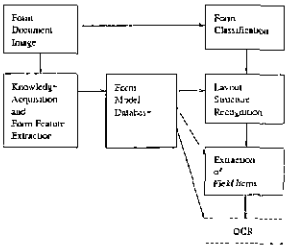

그림 1: 서식 처리를 위한 시스템 프레임워크

등록되어 있는 서식 구조에 대한 지식을 이용하여 서식의 구조와 특징을 추출한다. 그리고 추출된 서식 특징과 이미 서식 등록 단계에서 추출되어 모형에 등록되어 있는 서식 특징을 비교함으로써 입력 데이타 서식을 분류한다. 그 다음, 항목 유형 및 영역에 관한 지식을 이용함으로써 처리할 항목을 추출한다. 본 논문에서는 OCR 이전 단계까지의 내용을 다룬다.

3. 서식 지식 및 모형

3.1 서식문서의 특징

본 연구에서 다루고자 하는 서식문서는 선분 등과 같은 그래픽 요소가 중요한 역할을 담당하는 서식문서이다. 따라서 서식문서의 지형적, 논리적 구조를 효과적으로 추출하려면 그래픽 요소를 먼저 추

출하는 것이 바람직하다. 다루고자 하는 서식문서는 다음과 같은 특징을 갖는 문서이다. (1)선분은 서식문서의 이해에 있어서 중요한 시각적 단서를 제공한다. (2)채워진 항목의 위치는 선분 정보를 이용함으로써 유추할 수 있다. (3)항목간의 관계는 선분에 의해 정의된다

3.2 지식의 유형

서식문서에 대한 지식은 크게 서식 구조 지식과 항목 지식으로 나누어진다 서식 구조 지식은 다시 서식 구조에 관한 지식(FSN)과 서식 특징에 관한 지식(FFN)으로 나누어진다 항목 지식은 항목 유형 및 영역에 관한 지식(ITAN)과 항목 문맥에 관한 지식(ICN)으로 분류될 수 있다 서식을 이해하는데 있어서 모든 항목이 반드시 필요한 것은 아니기 때문에 ITAN 지식은 서식문서를 이해하는데 필요한 항목만을 정의한다. ICN 지식은 OCR 및 검증 단계에서 사용될 수 있는 지식으로서 항목의 구문적(syntactic), 의미적(semantic) 정보를 표현한다 따라서 본 연구에서는 FSN, FFN, ITAN, 그리고 ICN 지식을 이용하여 서식 모형을 정의한다.

3.3 서식 기술 언어와 서식 모델링

이 절에서는 앞서 설명한 서식 지식을 기술하기 위한 서식 기술 언어(FDL)에 대해 설명한다. FSN 지식은 서식 구조에 관한 지식으로서 다음과 같이 표현된다

$$\mathrm{FSN} = (L, (H|V), N, (L, T, R, B))^{+}$$

이때 L은 선분(line)을 의미하며, H와 V는 각각 선분의 유형인 수평(horizontal) 선분과 수직(vertical) 선분을 의미한다. (L, T, R, B)은 서식문서상의 영역을 표현한다 한편, $+$는 스크립트가 한번 이상 나타남을 의미한다 따라서 위의 FSN 표현은 서식문서 상에서 (L, T)과 (R, B)으로 결정되는 영역에 n개의 수평 혹은 수직 선분이 존재한다는 것과 하나 이상의 스크립트가 FSN를 구성할 수 있음을 의미한다. FSN은 이미 구해진 선분으로부터의 옵셋 정보를 이용하여 기술할 수 있다 가령, 다음의 스크립트는 (x_0, y_1)과 $(x_2, y_3 + 300)$으로 정의되는 영역에 수평선분이 8개가 존재함을 나타낸다

$$L\ H\ 8\ x0\ y1\ x2\ y3{+}300$$

이 경우 $x0$는 첫번째 수직 선분의 x좌표를 의미하고, $y3$는 네번째 수평선분의 y좌표를 의미한다. 이러한 논리정보를 이용함으로써 제한된 범위내에서 서식의 이동에 따른 문제를 극복할 수 있다

그 다음, 입력 데이터 서식을 분류하는데 사용되는 FFN 지식이 서식 등록 도구에 의해 자동으로 추출되어 등록된다 이를 위해, 수평선분과 수직선분이 각각 y좌표와 x좌표를 기준으로 정렬되어 모든 선분에 대해 이웃하는 두 선분간 좌표차가 계산된다. 모든 선분에 대해 구해진 값들은 서식 특징 벡터의 요소로 사용한다. 이는 서식 구조가 다르면 서식의 부류도 다르다는 가정에 근거한 것으로서 다음과 같다

$$\mathrm{FFN} = ((\alpha_1, \alpha_2, \ \ldots, \alpha_m)\ (\beta_1\ \beta_2, \ \ldots, \beta_n))$$
$$\alpha_i = |V_i - V_{i+1}|$$
$$\beta_j = |H_j - H_{j+1}|$$
$$1 \le i \le m-1$$
$$1 \le j \le n-1$$

이때 V_i와 H_j는 각각 i번째 수평선분의 y좌표값과 j번째 수직선분의 x 좌표값을 의미하며, m과 n은 각각 수평선분과 수직선분의 개수를 의미한다. 이러한 FFN은 서식 등록 도구에 의해 자동으로 추출되어 등록된다.

항목 유형 및 영역에 관한 지식인 ITAN은 다음과 같이 정의된다.

$$\mathrm{ITAN} = (IA, \sigma, (L, T, R, B))^{+}$$

이때, IA는 항목의 영역을 의미하며, $\sigma(\in \sum)$는 서식 처리 시스템에 의해 처리될 항목의 레이블을 의미한다. 가령, 다음의 스크립트는 TEL이라는 항목은 (x_1, y_1)과 (x_2, y_2)로 형성됨을 의미한다

$$IA\ TEL\ x1\ y1\ x2\ y2$$

이 스크립트는 서식 등록 도구에서 마우스 클릭 및 영역 선택에 의해 생성된다 마지막으로 항목 문맥에 관한 지식인 ICN은 다음과 같다

$$\mathrm{ICN} = (IC, \sigma, TYPE, ln, DIC)^{*}$$

이때 IC는 항목 문맥을 의미하며, $TYPE$은 항목의 유형을 표현한다 그리고 ln은 항목의 길 이를 나타내며, DIC은 항목이 나타나는 사전을 의미한다. $*$는 0번 이상 나타날 수 있음을 의미한다. 가령, 다음의 스크립트는 TEL 항목에 대한 문맥 정보를 정의한다.

$$IC\ TEL\ STR\ 8\ TABLE1$$

이 스크립트는 TEL이라는 항목은 8개의 문자로 구성되는 문자열이며 $TABLE1$에 나타남을 의미한다.

3.4 서식 모형

앞서 정의한 네가지 유형의 지식을 이용하여 서식 모형(FM)을 정의한다 즉, 서식 처리 시 스템이 다양한 유형의 서식을 효과적으로 처리할 수 있도록 FM을 정의한다 FM은 앞서 설명한 FSN, FFN, ITAN, 그리고 ICN을 이용하여 다음과 같이 표현할 수 있다.

$$\mathrm{FM} = (FSN, FFN, ITAN, ICN)$$

4. 서식 등록

4.1 모형 기반 방법

모형 기반 방법은 영상처리 기반 방법에 비해 응용 가능성이 많으며, 보다 융통성이 뛰어난 방법이다[3]. 따라서 모형 등록 단계에서 서식문서에 관한 지식을 추출하여 서식 모형으로 등록한 후 서식 처리 단계에서 이를 이용함으로써 다양한 서식문서를 효과적으로 처리할 수 있다. 즉, 서식 모형을 이용함으로써 서식 구조의 인식과 서식의 분류, 그리고 항목 인식 및 문자 인식도 효과적으로 수행할 수 있다 어느 정도까지의 지식을 사용할 것인지는 여전히 문제로 남지만, 서식문서에 대한 지식을 사용하지 않을 경우에는 극히 제한된 정보만 이용할 수 있으므로 문서를 효과적으로 처리하기가 어렵다 따라서 서식에 관한 최소한의 지식을 이용하면 처리하고자 하는 항목을 효과적으로 인식하고 추출할 수 있을 뿐만 아니라 문자 인식 및 검증 단계에서도 사용될 수 있다[4].

4.2 서식 등록 도구

서식 모형에 의해 표현되는 서식 정보는 원래의 문서에 관한 정보를 충분히 표현해야 한다. 또한 그러한 서식 모형은 쉽게 정의되어야 한다. 이를 위해 앞서 정의한 네가지 유형의 지식의 추출과 등록은 대화적 서식 정의 도구를 이용한다. 서식 정의 도구를 이용함으로써 서식 모형 정의시 발생할 수 있는 잠재적인 오류를 방지할 수 있을 뿐만 아니라 일반 사용자도 서식 모형을 쉽게 정의할 수 있다

먼저, 서식 정의 도구를 이용하여 서식 구조 분석시 사용될 지식인 FSN을 정의하여 등록한다 마우스를 이용하여 좌상단 좌표와 우하단 좌표를 선택하면 서식 등록 도구는 해당 영역내에서의 수평 선분과 수직 선분의 수를 구하여 사용자에게 FSN 스크립트를 보여준다 이때 수평 선분과 수직 선분의 수는 히스토그램을 이용하여 유추한다. FSN은 논리 정보로 기술됨으로써 제한된 범위내에서의 서식 이동에 따른 문제를 해결할 수 있도록 한다.

FSN 지식을 등록한 후, FFN 지식이 추출되어 등록된다. FSN과 FFN 지식을 등록한 후 ITAN 지식을 등록한다. ITAN 지식도 FSN 지식과 마찬가지로 대화적 사용자 인터페이스를 이용하여 등록할 수 있다. 만일 사용자가 마우스를 이용하여 특정 영역을 클릭하면 서식 등록 도구는 수평/수직 선분에 의해 둘러싸인 영역을 유추하여 스크립트를 보여준다. 항목 영역이 선분에 의해 결정되지 않을 경우

에는 직접 영역을 지정할 수도 있다. 이 경우 서식 등록 도구는 지정된 영역으로부터 가장 가까운 기존의 선분을 기준으로 한 논리 정보를 표현한 후 등록한다. 마지막으로, 특정 항목에 대한 구문적, 의미적 정보를 정의하여 등록함으로써 추출된 항목을 쉽게 해석할 수 있도록 한다.

5. 서식 처리

5.1 서식 구조 인식 및 서식 분류

우선 시스템은 FSN 지식을 이용하여 서식문서 구조를 구성하는 선분을 추출한다. 그리고 추출된 모든 수평, 수직 선분에 대해 각각 y와 x 좌표값에 따라 정렬하여 이웃하는 선분간의 거리를 구한 후 그 값을 원소로 하는 서식 특징 벡터를 구성한다. 이 특징 벡터와 모형으로 등록되어 있는 특징을 비교함으로써 서식 분류가 수행된다. 이 때 입력 데이터 서식은 시스템에 등록되어 있는 모형 서식중 가장 가까운 거리값을 갖는 서식으로 분류된다. 이는 다음의 d-차원 유클리드 거리 공식을 이용한다.

$$D = \left[\sum_{i=1}^{d}(I_i - M_i)^2\right]^{1/2}$$

이때 D는 입력 서식과 모형 서식의 거리이며, d는 특징 벡터의 차원이다. 그리고 I_i와 M_i는 각각 i번째 입력 및 모형 데이터 서식의 특징요소를 의미한다.

시스템에 등록되어 있는 서식 모형의 수가 증가할 수록 서식을 분류하는기 위한 시간도 증가한다. 따라서 계산 시간을 최소화하기 위하여 서식 처리 시스템은 부분적 하향식 비교 방법을 이용한다. 이 방법에서는 입력 데이터 서식과 모형 서식을 비교할 때 서식 전체를 비교하는 것이 아니라 서식의 일부분만을 비교한다. 입력 데이터 서식과 특징 부분이 일치하지 않는 서식들에 대해서는 더 이상 비교가 이루어지지 않기때문에 결과적으로 등록되는 서식 모형이 증가하더라도 입력 데이터 서식을 분류하는데 드는 시간을 상당히 줄일 수 있다.

5.2 필드 영상 추출

서식 구조 인식 및 서식 분류가 수행되면 ITAN 지식을 이용하여 처리하고자 하는 항목을 추출한다. ITAN 스크립트는 서식 구조를 구성하는 선분을 이용하여 논리 정보로 기술되며, 이를 이용으로써 시스템에서 처리하고자 하는 항목이 자동으로 분류되고 추출된다.

6. 실험 결과

제안한 방법의 성능을 테스트하기 위하여 모형 기반 시스템을 구현하였다. 그리고 국민은행, 조흥은행, 신한은행, 하나은행, 보람은행 등 시중 은행의 8가지 유형의 입출금표 서식을 이용하여 실험하였다. 펜티엄 233 MMX 상에서 C++ 프로그래밍 언어를 이용하여 시스템을 구현하였다. 서식의 수는 각 서식에 대해 각각 21개씩 전체 168개의 입출금표 서식을 실험에 사용하였다. 이 중 8개는 서식 모형을 등록하는데 사용하였으며, 160개는 테스트에 사용하였다. 모든 서식들은 200 dpi로 스캔하였으며, 서식 영상의 평균 크기는 1662 × 1124 픽셀이었다. 각각의 입출금표 서식에 대해, 앞서 설명한 바와 같이, 서식 등록 도구를 이용하여 네가지 유형의 서식 지식을 추출하여 등록한 후 다음의 실험을 수행하였다. (1)입출금표 유형에 따른 서식 인식률, (2)등록된 서식의 개수에 따른 서식 분류 시간, (3)항목의 개수에 따른 항목 추출 시간. 〈표 1〉은 서식 유형에 따른 서식 인식 결과를 보여준다.

8가지 유형의 서식에 대해 네가지 지식이 서식 모형으로 등록된 후 서식 분류가 수행된다. 즉, 입력 데이터 서식과 가장 유사한 서식 구조를 갖는 모형 서식을 찾는다. 실험결과 스캔시 서식문서가 잘못된 선분 정보를 갖고 있을 경우를 제외하고는 모두 바르게 인식되었다. 160개의 테스트 입출금표 서식문서에 대해 평균 99.4%의 인식

표 1: 서식 유형에 따른 서식 인식 결과

Type	No. of Rec.	No. of Rej.	Rec Rate(%)
T_0	20	0	100
T_1	20	0	100
T_2	20	0	100
T_3	20	0	100
T_4	19	1	95
T_5	20	0	100
T_6	20	0	100
T_7	20	0	100
Sum/Aver.	159	1	99.4

결과를 얻을 수 있었다. 한편, 하나의 모형 서식이 등록되어 있을 경우 서식을 인식하는데 걸리는 평균 시간은 0.13초였으며, 8개의 모형이 등록되어 있을 경우 걸리는 평균 시간은 0.74초였다.

표 2. 항목수에 따른 항목 추출 시간

Type	No. of items	Times(Sec)
T_0	5	0.43
T_1	5	0.42
T_2	6	0.52
T_3	5	0.37
T_4	6	0.51
T_5	6	0.53
T_6	5	0.38
T_7	5	0.41
Aver.	5.4	0.45

항목 추출시 걸리는 시간에 대한 성능을 테스트하기 위하여 모든 유형의 서식에 대해 계좌번호, 비밀번호, 금액(한글, 숫자), 성명, 전화번호 등의 5개 혹은 6개의 필드 항목을 정의하여 추출하였다. 〈표 2〉는 각 유형의 서식에 대해 항목수 및 항목 추출 시간을 보여준다. 올바로 인식된 모든 서식에 대해 100%의 항목 추출률을 보였으며 항목을 추출하는데 걸리는 시간은 평균 0.45초였다.

7. 결 론

본 논문에서는 다양한 서식문서를 처리하기 위한 모형 기반 방법에 대해 살펴보았다. 걸리는 시간을 줄일 수 있었다. 실험결과 8개의 서식 모형이 등록되어 있을 경우에는 서식 분류 시간은 평균 0.74초였으며, 5개 혹은 6개의 항목을 추출하는데 걸리는 시간은 평균 0.45초였다. 이는 응용가능성 측면으로 볼 때 매우 고무적인 결과로 보인다. 본 방법은 서식 영상의 질이 좋지 않을 경우에도 잘 동작함은 물론, 서식에 대한 지식을 이용함으로써 선분으로 구성되지 않은 항목들도 효과적으로 추출할 수 있었다. 그리고 서식 모형만 추가함으로써 다른 서식 문서도 쉽게 처리할 수 있다.

참고 문헌

[1] S. W. Lam, L. Javanbakht, S. N. Srihari, "Anatomy of a Form Reader," *ICDAR*, pp506-509 (1993).

[2] T. Watanabe, Q. Luo, N. Sugie, "Knowledge for Understanding Table-form Documents," *IEICE trans. on Inf. and Syst.* Vol. E77-D, No. 7, pp761-769 (1994).

[3] Y. Ishitani, "Model Matching Based on Association Graph for Form Image Understanding," *ICDAR*, pp287-292 (1995).

[4] T. Watanabe, T. Fukumura, "A Framework for Validating Recognized Results in Understanding Table-form Document Images," *ICDAR*, pp536-539 (1994).