2001年度 第14回 信號處理合同學術大會 論文集 第14卷 1號

PDF417 이차원 바코드 디코딩 알고리즘의 구현

정 정구°, 한 희일
한국외국어대학교 공과대학 정보통신 공학과

Implementation of PDF417 2-dimensional Barcode Decoder

Joung-Koo Joung°, Hee-il Hahn
Department of Information and Communications Engineering, Hankuk University of Foreign Studies
jgchong@ice.hufs.ac.kr, hihahn@ice.hufs.ac.kr

요 약

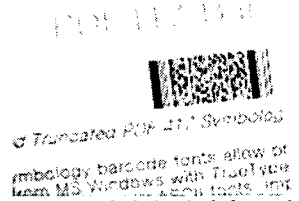
종래에 사용되어 왔던 1차원 바코드가 정보를 포함하고 있는 데이터베이스에 접근하는 데이터 키 역할을 주로 해온 것에 비해, 2차원 바코드는 다량의 데이터를 포함할 수 있고 고밀도의 데이터 표현이 가능하여, 호스트 컴퓨터의 데이터 베이스에 온라인 연결할 필요없이 확인하고자 하는 사람이나 대상물에 대한 정보를 얻을 수 있다. 본 논문에서는 가장 널리 사용되는 2차원 바코드 체계인 PDF417 을 중심으로, 디지털 카메라를 통하여 입력한 영상을 이진화하여 시작 심볼 또는 정지 심볼을 검색함으로써 2차원 바코드 영역을 추출한 다음, 추출된 영역으로부터 바코드의 행과 열의 수, 오류수정 정도 등의 헤더정보를 검출하고 이를 바탕으로 코드워드를 추출하는 알고리즘을 제안한다. 얻어진 코드워드는 데이터를 효율적으로 저장하기위해 정보가 숫자인지, ASCII코드인지, 혹은 바이트 정보인지에 따라 다른 방식으로 인코딩 되어 있는데, 그에 따른 디코딩 알고리즘을 제안한다

1. 서론

2차원 바코드는 바 (bar) 형태의 1차원 바코드가 수평방향 만의 정보를 갖고 있는 데 비해 수평, 수직 두 방향의 정보를 수록할 수 있으며 1차원 바코드의 1백배인 최대 2천자까지 문자를 수록할 수 있어 좁은 공간에 다양한 정보를 수록해야 하는 브라운관 공장이나 병원의 환자이력관리 프로그램 등에 사용되는 첨단 바코드 체계이다 [2]. 또한, 1차원 바코드의 경우 영문과 숫자만을 기록할 수 있는 데 비해 2차원 바코드는 영어, 일어, 한자, 숫자 등 다양한 문자를 기록할 수 있으며 1차원 바코드의 20~40배 이상의 밀도를 갖고 있어 전체 30% 이상의 데이터 손상 에도 복원이 가능하다 [1].

특히, 2차원 바코드에서 가장 널리 적용되고 있는 PDF417 코드는 라벨의 위치에 관계없이 360도 전방향에서 원격, 고속으로 바코드를 인식할 수 있으며, 데이터 컴팩션 모드에 따라 최대 ASCII 1850 캐릭터, 1108 바이트 또는 2710개의 숫자까지 데이터 표현이 가능하다 [5]. 따라서 생산관리, 품질관리 정보 등을 다양하게 표현하고 얻을 수 있어 복잡한 제조환경에서 정확한 생산관리를 구현할 수 있는 장점이 있다.

PDF417은 고밀도의 데이터 저장능력과 오류수정 기능이 포함된 다행의 가변 길이 심볼로지로서 선형 스캐너, 레이저 스캐너로 스캔할 수도 있지만, 본 논문에서는 CMOS 디지털 카메라를 이용하여 바코드 이미지를 입력하였다.



<그림 1> CMOS 디지털 카메라로 입력한 PDF417 바코드 이미지.

<그림1>은 CMOS 디지털 카메라로 입력한 PDF417 바코드 이미지를 보여주고 있다. PDF417 심볼은 최소 3개에서 90개 까지의 행으로 구성되어 있는데, 각 행은 <그림2>에 제시한 바와 같이, 시작 패턴, 종료 패턴, 좌우 행 지시자, 데이터 영역 등으로 나뉜다. 행의 수와 길이는 바코드가 인쇄될 때 선택할 수 있어서, PDF417 심볼의 크기 비율은 가변적이다. PDF417 심볼 캐릭터는 각각 4개의 바와 스페이스 (space) 의 배열로 표현된 17 모듈로 구성되어 있는데, 이로 인하여 PDF417이라는 이름을 갖게 되었다[5].

2차원 바코드시스템은 미국, 일본의 경우 자동차, 전자부품, 마이크로컴퓨터, 의료제약업계 및 정부자료통계 분야에서 도입되고 있으며, 미국 표준 기관인 ANSI를 중심으로 전자문서교환(EDI)메시지의 보완수

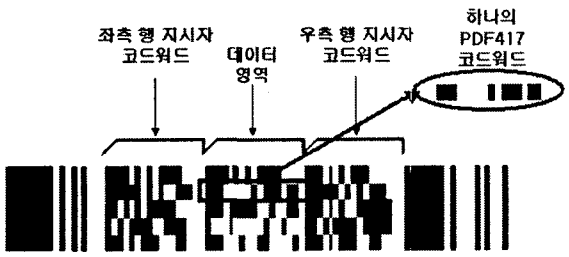
단으로 이를 채택키로 하는 한편, 세계 자동인식 산업협회에서 심볼로지의 표준사양 및 표준규격 제정을 추진하고 있다.

산업 전분야에 컴퓨터의 보급으로 컴퓨터를 이용하지 않고는 가격 대 성능의 경쟁력에서 이겨낼 수 없다. 이러한 측면에서 인건비를 절감하면서 정보의 분류와 검색이 용이하도록 하는 방법의 개발은 필수적이다. 기존에 사용되고 있는 1차원 바코드는 제한된 형식과 정보량을 가지고 있어서, 이러한 정보 형태로는 여러 응용 분야에 적용이 극히 제한적이다. 따라서 정보의 형태에 유연성을 부여하는 2차원 바코드의 개발은 매우 시급하다고 할 수 있다.

본 논문에서는 CMOS 디지털 카메라를 이용하여 스캔한 PDF417 바코드 이미지로부터 코드워드를 추출하는 알고리즘과 얻은 코드정보를 원래의 데이터로 복원하는 알고리즘을 제안한다.

2. PDF417 바코드의 구조

2.1 심볼 구조



<그림 2> PDF417 심볼 구조

<그림2>과 같이 PDF417 심볼은 여러 개의 열을 수직적으로 누적하여 배열하며 바코드의 주위에는 빈 여백(quiet zone)이 존재한다. 시작패턴과 종료패턴은 리더기가 바코드의 방향과 위치를 인식할 수 있도록 해준다. 시작 패턴은 바(bar)와 스페이스(space) 순서로 81111113 이고 종료 패턴은 711311121이다. 좌측, 우측 행 지시자와 데이터 영역을 구성하는 각 심볼 문자는 4개의 바와 4개의 스페이스의 조합으로 구성되며, 17모듈의 너비를 갖는다. 각 심볼 문자는 0-928까지의 값을 가지는데 이 심볼 문자의 값을 코드워드(codeword)라고 한다. 좌측, 우측 행지시자의 코드워드는 데이터를 저장하지 않으며, 단지 행번호, 행의 수, 데이터 영역의 열의 수, 오류수정 레벨의 정보를 가진다. 데이터 영역의 열의 수는 1-30까지 가능하며, 행의 수는 3-90까지 가능하다. 데이터 영역의 첫번째의 코드워드는 심볼 길이 지시자(symbol length descriptor)로서 데이터 코드워드의 총 개수를 나타내는데 그 값은 자신의 코드워드를 포함하나, 오류 검출용 코드워드는 제외한다. 한 심볼 당 코드워드의 개수는 928을 초과할 수 없다. 데이터 영역의 코드워드는 좌측에서 우측으로 위에서 아래로 심볼 길이 지시자, 데이터 코드워드, 패드(pad) 코드워드, 오류 검출용 코드워드 순서로 배열한다[5].

2.2 컴팩션 모드

컴팩션(compaction) 모드는 많은 양의 데이터를 최소의 코드워드로 만들기 위해 고안된 것으로, 텍스트 컴팩션(TC) 모드, 바이트 컴팩션(BC) 모드, 뉴메릭 컴팩션(NC) 모드가 있다. 각 모드간에는 모드랫치(latch) 와 모드쉬프트(shift) 지시자를 사용하여 이동할 수 있는데, 모드쉬프트는 순간적인 모드변환으로서 모드변환코드워드 다음의 한 코드워드에만 모드변환이 적용되고 원래의 모드로 돌아오는 것이고, 모드랫치는 영구적으로 모드를 변환한다.

텍스트 컴팩션(TC) 모드는 ASCII 문자들을 최소한의 코드워드로 표현하며, Alpha, Lower Case, Mixed, Punctuatioin의 4개의 부 모드로 구성된다. 각 부 모드마다 0에서 29사이의 값이 할당되어 있으며, 코드워드 하나에 두개의 ASCII 값이 저장된다.

바이트 컴팩션(BC) 모드는 다국어를 표현하거나, ASCII에 없는 문자를 저장하는데 사용될 수 있다. 또한 그림이나 특수한 파일을 바이트 자체로 저장할 수 있다. 각 바이트의 값은 GLI와 매핑되어 값을 표현한다. BC모드에서는 아래의 식과 같이 900진법을 256진법으로 변환함으로써 6바이트를 5개의 코드워드로 표현할 수 있다. b가 바이트이고 c가 코드워드라고 할 때, 진법 변환식은 다음과 같다 [1].

$$b1*256^5+b2*256^4+b3*256^3+b4*256^2+b5*256+b6 = c1*900^4+c2*900^3+c3*900^2+c4*900+c5$$

그러나 BC모드의 코드워드수가 5바이트 미만일 경우, 각 코드워드는 바이트 값과 일대일로 매핑된다.

뉴메릭 컴팩션(NC) 모드는 숫자가 연속적으로 13개 이상일 때 사용되며, 인코딩 알고리즘은 다음과 같다. 44개의 숫자로 구성된 그룹으로 나눈 후, 숫자들의 앞에 1을 삽입한다. 10진법을 900진법으로 변환 함으로써 각 45개의 숫자 그룹을 15코드워드로 인코딩 한다. 마지막 그룹의 숫자의 개수는 45보다 작을 수도 있다. 다음은 디코딩을 하는 예를 보여주고 있다.[1]

코드워드가 1, 624, 434, 632, 232, 200 일 때, 다음의 수식 $1 * 900^5 + 624 * 900^4 + 434 * 900^3 + 632 * 900^2 + 232 * 900 + 200$ 은 십진수로 1000213298174000이다. 여기에서 앞의 1을 빼주어 000213298174000 인 원래의 데이터를 복원할 수 있다. NC모드에서 하나의 코드워드는 거의 3개의 숫자를 표현할 수 있다.

2.3 오류 검출 및 수정

코드워드를 원래의 데이터로 복원하기 전에 코드워드에 오류가 있는지를 검사하고 수정하기 위해서 PDF417심볼의 데이터 영역의 가장 뒷부분에는 오류수정 코드워드를 추가한다. 0-8사이의 오류수정 레벨에 따라서 추가되는 오류수정 코드워드의 수도 달라진다. 레벨이 높을수록 데이터의 안정도는 높아지지만 데이터의 밀도는 낮아진다. 오류수정 코드워드는 잘 알려진 Reed Solomon 오류제어 코드 알고리즘으로 생성된다[5].

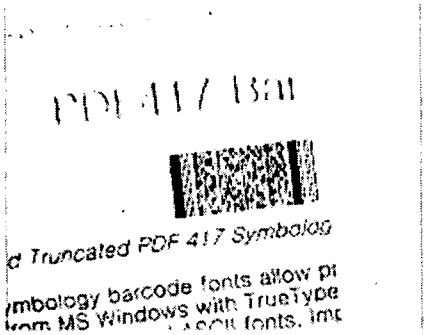
2.4 GLI(Global Label Identifier)

GLI는 인코딩된 코드워드를 원래의 데이터로 해석할 수 있도록 하기 위해 정의된 코드워드의 집합으로서 각 GLI에 따라서 코드워드의 해석이 달라진다. GLI값으로 811,800개가 가능하므로, 어느 나라의 언어라도 PDF417은 표현할 수 있다.

3. PDF417 바코드 코드워드 추출 알고리즘

바코드 이미지를 분석하고 해석하기 위해서는 우선, 바코드 이미지를 이진화하여야 한다. 이미지를 이진화하는 방식으로는 대상 이미지에서 구한 히스토그램으로부터 적절한 문턱값을 구하여 각 픽셀 값이 이 보다 크면 1로, 작으면 0으로 정하는 전역 이진화 방식 (global thresholding technique) 이 가장 널리 사용되고 있다. 그런데, 디지털 카메라를 이용하여 스캔한 바코드 이미지는 조명, 잡영, 텍스춰 등이 일정하지 않고, 또한 이미지와 카메라 간의 거리와 각도가 가변적이므로, 전역 이진화 방식은 바코드 이미지의 이진화에 적합하지 않다 [3]. 본 논문에서는 이들의 영향을 줄이기 위하여 다음과 같은 방법으로 바코드 이미지를 이진화하는 알고리즘을 제안한다.

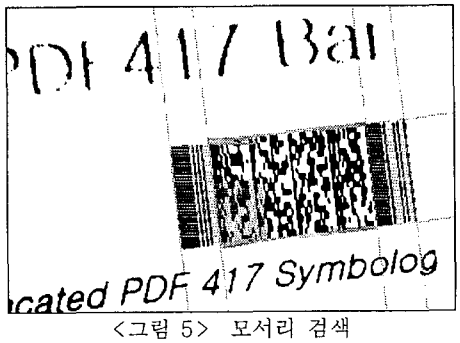
기본적으로 문턱값은 배경과 바코드 간의 경계와 바코드에서 바와 스페이스 간의 경계를 구분해야 하는데, 주어진 픽셀에서의 그레이던트 값은 그 픽셀이 경계의 밝은 면에 있는지 어두운 면에 있는지에 대한 정보를 제공해 준다 [4]. 이를 이용하여 본 논문에서는 각 행 마다 라플라시안 (Laplacian) 오퍼레이터를 이용하여 그레이던트 값을 구하고 이의 절대값이 주어진 값 보다 큰 픽셀에 대한 평균치를 구하여 이를 문턱치로 이용하였다. <그림4>는 제안한 방식으로 이진화한 결과를 보여주고 있다.



<그림 4> <그림 1>을 이진화한 이미지

PDF417 바코드는 특정한 시작과 종료 패턴을 정의하고 있다. 이러한 시작 패턴이나 종료 패턴을 찾으면 이미지 상에 바코드의 위치를 알 수 있다. 본 논문에서는 이진화된 바코드 이미지의 수평과 수직으로 스캔하면서 시작 패턴이나 종료 패턴을 찾고 이들의 시작점과 끝점을 저장한 다음, <그림 5>에 제시한 바와 같이 이 점들을 지나는 최적의 선의 방정식을 구한다.

카메라의 렌즈에 따라 입력 이미지에 상당한 왜곡이 발생하므로 전처리 과정으로 이미지 와핑 (warping) 을 수행하면 보다 높은 인식률을 얻을 수 있다. 이를 위해 네 개의 모서리를 찾아야 하는데, 이를 위해 구해진 선의 방정식에 수직인 방향으로 스캔하면서 바코드의 모서리를 예측한다. 예측된 모서리가 정확한 것인지를 확인하기 위해 모서리의 바로 위아래의 행 번호 정보를 추출하여 이를 검증한다. 만약 모서리를 찾지 못하였다면 행 지시자 정보를 분석하여 모서리의 위치를 추정한다.



<그림 5> 모서리 검색

네 개의 모서리를 찾아낸 후 <그림6>과 같이 와핑을 통해 데이터 영역만을 추출해 낸다.

<그림 6> 와핑을 통해 추출된 영역

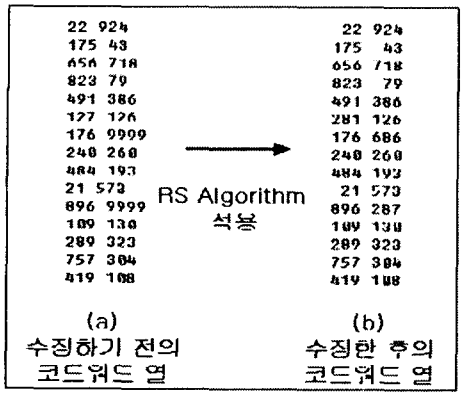
위와 같은 방법으로 바코드의 영역이 추출되면, 왼쪽에서 오른쪽으로, 위에서 아래로 스캔하면서 바-스페이스 패턴을 추출한다. 추출한 패턴이 옳은 것인지 확인하기 위해 "edge to similar edge" 측정 방법을 사용한다[2]. <그림7>은 스캔하여 얻은 패턴 정보를 보여준다.

```
81111113 41111414 31143122 41114141 61111232 711311121
81111113 31111163 22411124 12423131 11111444 711311121
81111113 51116111 24111511 31131332 11114243 711311121
81111113 21113513 23115131 12133511 31113323 711311121
81111113 41114114 12151142 11162151 41115122 711311121
81111113 21123242 31211126 12313214 21123153 711311121
81111113 61121132 22113314 22115231 11116232 711311121
81111113 21122172 14121341 12121541 61113113 711311121
81111113 11133134 11113136 11333213 21132332 711311121
81111113 21123413 12611232 41212412 00000000 711311121
81111113 11172113 00000000 21211164 61131211 711311121
81111113 11161116 31231322 31241123 11142521 711311121
81111113 41131412 13225112 21222115 00000000 711311121
81111113 31145111 12232241 21136112 11161241 711311121

|-------||-------||----------------||-------||--------|
    a       b              c             d        e
a, e: 시작, 종료패턴
b, d: 행 지시자
c   : 데이터 영역
```

<그림 7> 스캔하여 추출한 바-스페이스 패턴정보

<그림7>의 패턴정보를 미리 정의된 변환표와 비교하여 실제 인코딩된 코드워드를 얻을 수 있다. 그렇게 하여 얻어진 코드워드가 <그림8>의 (a)이며, 이것을 (b)와 같이 Reed-Solomon 오류수정 알고리즘을 사용하여 오류를 검출하고 복원한다.



<그림 8> (a) 그림 5. 의 바-스페이스 패턴으로부터 구한 코드워드. (b) Reed-Solomon 오류수정 알고리즘으로 오류를 검출한 후의 코드워드.

4. PDF-417바코드의 디코딩 알고리즘

<그림 10>에서 얻어진 코드워드는 다음과 같다.

22
924 175 43 656 718 823 79 491 386 281
126 176 686 240 260 484 193 21 573 896
287 109 130 289 323 757 304 419 108

이렇게 얻어진 코드워드로부터 원래 정보로 디코딩 하는 과정을 아래에 단계별로 설명하기로 한다.

단계1: 심볼 길이 지시자인 첫 코드워드는 22이므로, 실제 디코딩에 사용되는 코드워드의 개수는 21개이다.

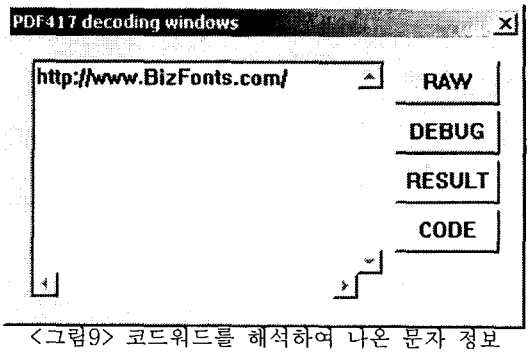
단계2: 2번째 코드워드가 924이므로 BC모드이며, 인코딩된 바이트의 개수가 6으로 나누어진다.

단계3: 3번째부터 7번째 코드워드까지 5개의 코드워드를 2.2.4절에서 설명한 900 to 256연산을 하면, 아래의 결과를 얻을 수 있다.

$$175*900^4+ 43*900^3+ 656*900^2+ 718*900+ 823 =$$
$$104*256^5+ 116*256^4+ 116*256^3+ 112*256^2+ 58*256+ 47$$

5코드워드의 열(175, 43, 656, 718, 823)은 6바이트의 열 (104, 116, 116, 112, 58, 47)로 변환 된다. 구한 6바이트와 현재의 GLI(디폴트 0)를 비교하여 문자를 얻어낸다. 비교해 보면, " http:/ " 를 얻을 수 있다.

단계4: 8번째 코드워드 값이 900보다 작으므로, 다음에 오는 5바이트도 동일한 모드로 지속되며, 22번째 코드워드까지 지속된다. 심볼 길이 지시자의 값이 22이므로 더 이상 해석할 코드워드가 없으므로 <그림11>과 같이 문자를 출력하고 디코딩을 종료한다. 이 예에서는 총 24문자를 구하였다.



<그림9> 코드워드를 해석하여 나온 문자 정보

5. 결 론

본 논문에서는 CMOS 디지털 카메라로 입력한 PDF417 바코드 이미지로부터 바코드 영역을 추출한 후에 바-스페이스 패턴을 추출하고 이를 코드워드로 변환하는 알고리즘을 제안하였다. 또한 코드워드로부터 원래의 데이터로 디코딩 하는 알고리즘을 제안하였다. 코드워드 정보 출출 알고리즘에 있어서 입력 이미지의 이진화 결과가 성능에 상당한 영향을 끼치고 있음을 실험을 통해 확인하였다. 따라서, 이진화 알고리즘을 개선하면 보다 안정적이고 향상된 결과를 얻을 수 있을 것으로 기대된다. 디코딩 알고리즘의 구현에 있어서는 바이트 컴팩션 모드와 뉴메릭 컴팩션 모드의 구현에 있어서 계산의 중간값의 크기가 C언어의 32비트 정수변수의 범위를 넘어서므로 진법변환을 위해서는 특별한 함수를 만들 필요가 있었다. 오류수정 레벨이 높아지면 계산 시간이 오래 걸리므로, 알고리즘을 개선하거나, 혹은 오류수정 알고리즘을 하드웨어로 구현하면 계산속도를 상당히 증가시킬 수 있을 것이다. 앞으로의 과제인 한글에 대한 디코딩은 AIM에서 GLI를 정의하는 작업이 끝나면 구현할 것이다.

참고 문헌

[1] Roger C. Palmer, The Bar Code Book,

[2] 오호근, 바코드 기술 및 응용, 성안당, 1997.

[3] Eugene joseph and theo Pavlidis, "Waveform Recognition with Application to Barcodes", Symbol Technologies Inc. 116 Wilbur Place, Bohemia, NY 11716., 1991.

[4] R. C. Gonzalez, P. Wintz, "Digital Image Processing", Addison Wesley, 1987.

[5] Uniform Symbology Specification PDF417, AIM USA.