

우편물 자동처리를 위한 4-state 바코드 Reed Solomon 인코딩 시스템

박문성, 송재관, 황재각, 남윤석
한국전자통신연구원 우정자동화팀
전화 : 042-860-65815753 / 핸드폰 : 016-419-3848

Reed Solomon Encoding System of 4-state Bar Code for Automatic Processing in Mail Items

Moon-Sung Park, Jae-Gwan Song, Jae-Gak Hwang, Yun-Seok Nam
Postal Automation Technology Development Team, ETRI
E-mail : mspark@etri.re.kr

Abstract

Recently many efforts on the development of automatic processing system for delivery sequence sorting have been performed in ETRI, which requires the use of postal 4-state bar code system to encode delivery points. The 4-state bar code called postal 4-state bar code for high speed processing that has been specifically designed for information processing of logistics and automatic processing of the mail items. This paper describes a method of Reed-Solomon encoding for creating error correction codeword of 4-state bar code.

I. 서론

현재, 우편물 자동구분 처리 시스템은 3 out of 5 바코드를 적용하여 OCR(Optical Character Recognition)로 인식(인식율 85%)된 우편번호를 기계 바코드(총 길이 70mm)로 인쇄하고, 이 결과를 LSM(Letter Sorting Machine)의 바코드 판독기(판독율 95%)에 의하여 우편물 상에 인쇄된 기계 바코드를 판독하여 구분하고 있다. 또한, OCR의 판독율이 낮아 발생하는 오류우편물의 최소화를 위하여 우편고객에게 기계 바코드와 동일한 규격으로 우편봉투에 인쇄하도록 하고, 이 우편물은 OCR에 투입하지 않고 LSM에 직접 투입하여 처리됨으로써, 자동 구분처리 업무를 보다 효과적으로 수행할 수 있게 되었다[1,2,3]. 그러나, 우편물 자동구분의 범위를 확장하고 우편고객에게 다양한 부가서비스를 제공하기 위해서는 바코드 정보수득의 밀도가 높고, 고속 인쇄 및 판독이 용이한 4-state 바코드의 도입이 요구되고 있다. 이에 따라, 4-state 바코드에 우편번호, 우편물 배달순서코드를 수록하고, 나머지 공간에는 고객의 정보를 수록하여 다양한 부가서비스를 제공하기 위한 노력이 요구되고 있다. 또한, 다량 우편물을 취급하는 우편 고객을 대상으로 시행되고 있는 고객

바코드를 지원할 수 있도록 LSM에 흑백 바코드 판독 기능을 추가되어야 하며, 우편고객이 사전에 인쇄하기 위한 4-state 바코드 인코딩 시스템 개발이 요구되고 있다. 본 논문에서는 우편물 자동구분용 바코드 인쇄모듈과 고객 바코드로 활용될 예정인 4-state 바코드에 정보와 함께 수록될 Reed Solomon 오류정정 코드워드를 생성하기 위한 연산과정을 단순화시키기 위한 방법을 제시하고, Decoding 과정에도 적용하여 인코딩 결과를 바탕으로 검증시험을 수행한 것이다.

제 2장에서는 4-state 바코드 정보체계 및 바코드 인쇄 규격을 보이고, 제 3장에서는 4-state 바코드에 정보를 수록하고, 오류정정 코드워드를 생성하기 위한 시스템을 설계한 내용이다. 그리고, 오류정정 코드워드를 생성함에 있어 16 bit Exclusive OR 연산방법을 적용한 내용이며, 제 4장에서는 시험용 Reed Solomon Decoding시스템에 제시된 연산방법을 적용하여 판독시험한 결과를 보인 것이다. 제 5장에서는 연구결과의 활용 방안 및 추후 연구사항을 다루었다.

II. 우편용 4-state 바코드

우편용 4-state 바코드는 영국 Royal Mail에서 처음 개발하였으며, 캐나다, 일본, 호주 등의 국가에서도 이 4-state 바코드를 자국의 실정에 맞게 수정하여 적용하고 있다[4,5,6]. 4-state 바코드는 영국 Royal Mail 및 캐나다 CPC(Canada Post Corporation)의 4 State 바코드를 기반으로 국내 환경에 적합한 한글정보를 수록할 수 있도록 구성한 것을 특징으로 한다. 영국 방식은 4-state 바코드를 상하로 구분하여 0, 1, 2, 4의 바 가중치를 적용하고, 6x6 매트릭스 참조테이블에 의하여 36개의 문자(숫자, 영문 대문자)를 사용하고, parity bit는 상위 및 하위의 모두 더하여 6으로 나는 나머지의 값을 적용하고 있다.

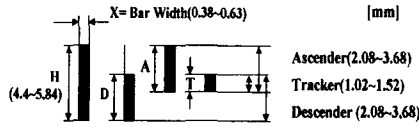


그림 1 우편용 4-state 바코드 인쇄규격
Fig. 1 Printing Specification

캐나다와 국내에서는 각 바에 0,1,2,3인 가중치 값을 부여하고, 바의 위치 값과 4의 계수 값들을 기반으로 바코드 심볼로지 값을 정의되도록 하는 방법을 적용하였으며, 이에 대응하는 수식은 다음과 같다.

$$B_n B_{n-1} \dots B_1 = V_n \times 4^{n-1} + V_{n-1} \times 4^{n-2} + \dots + V_1 \times 4^0$$

이 수식에서 B는 바를 표시하며, V는 가중치, n은 위치 값을 말한다. 수식에 바코드문자의 값이 ADTA 값일 경우 다음과 같다.

$$ADTA = 1 \times 4^0 + 2 \times 4^1 + 0 \times 4^2 + 1 \times 4^3 = 1 + 8 + 0 + 64 = 73$$

이와 같은 수식의 의하여 숫자는 0 ~ 9(4-state 2 바), 영문자(4-state 3바) 그리고 한글 완성형 문자(KS 5601, 2,350개)를 적용하기 위하여 49x48의 매트릭스 참조테이블을 정의하고 있다. 또한, 한글 1문자를 표현하기 위하여 행과 열의 값을 바코드 심볼로지 표현하는 방법이 정의된 상태(4-state 6바)이다. 이 바코드에는 우선적으로 우편물 자동구분을 위한 우편번호와 배달순서코드를 적용할 예정이다.

III. 우편용 4-state 바코드 인코딩 시스템 설계 및 구현

앞 장에서 보인바와 같이 다양한 정보를 수록할 수 있고, 우편물 자동구분 처리는 시간당 35,000통 이상을 바코드로 인쇄하고 판독하여 자동으로 구분해야 한다. 우편물 자동처리 촉진을 위하여 다량우편이용자에게 인쇄기준 및 인코딩 모듈을 제공하여 사용하도록 할 예정이다. 다량 우편이용자가 사용할 경우에는 많은 우편물에 바코드를 인쇄하여야 하므로 인쇄품질이 저하되거나, 인쇄시스템의 결함 등으로 인한 문제점들을 내포하고 있다. 또한, 고객 바코드를 이용하여 다양한 부가서비스를 제공하기 위해서는 바코드의 심볼로지가 부분적으로 훼손되어 있어도 보정이 되어야 한다. 즉, 4-state 바코드의 경우에는 다량의 우편물 정보를 고속으로 인쇄 또는 판독해야 하므로 바코드에 수록된 정보의 안정성을 확보하기 위한 방법을 적용하여야 한다. 이와 같은 요구사항을 고려하여 높은 판독율을 달성하기 위한 노력의 일환으로 바코드 심볼로지의 훼손에 따른 오류정정 기능은 필수적이다.

이에 따라, 본 절에서는 4-state 바코드에 수록된 수록된 정보의 안정성 확보를 위하여 데이터 통신을 위하여 주로 사용되는 Reed Solomon 알고리즘을 적용하고자 한다. 또한, 오류정정 코드워드를 생성함에 있어 보다 적은 단계의 연산에 의하여 오류정정 코드워드를 생성하는 방법을 제시하고자 한다. 기존의 Reed Solomon 알고리즘에 의한 오류정정 코드워드 생성하려면, $GF(2^N)$ 에 대응하는 범위를 결정하여야 한다. 4-state 바코드의 경우에 4^N 방법이 적용되어 있으므로 쉽게 적용할 수 있다. 4-state 바의 값이 0에서 3의 값을 가지므로 몇 개의 바 단위로 그룹핑하는가에 의하여 결정될 수 있다. 그리고, 4-state 2,3,4... 바 단위로 그룹화할 경우에 2^N 의 값 즉, 16, 64, 256... 등으로 설정될 수 있다. 그러나,

4-state 바코드는 최대 바의 개수가 78개 바이고, 인쇄길이의 제한조건이 있으므로 적절한 단위를 결정하여야 한다. 수록되는 정보의 양 및 길이도 고려되어야 한다.

본 논문에서는 4-state 4 바 이상으로 할 경우에는 오류정정 코드워드의 길이가 수록되는 정보의 길이보다 많이 점유 될 수 있으므로 4-state 3바 단위로 그룹화하였다. 이에 따라 $GF(2^6)$ 에 의한 기존 다항식을 적용하여 오류정정 코드워드를 생성할 수 있도록 하였다. 이러한 방법을 적용함에 있어 오류정정 레벨을 1로 설정하면, 4개의 오류정정 코드워드가 생성된다. 오류정정 범위는 4개의 심볼로지(12개 바)에 대한 소거 오류 및 5개 바 미만의 대치오류를 정정할 수 있게 된다. 즉, 4-state 바코드는 최대로 적용할 수 있는 바의 개수는 78개(1인치에 24개 바)이고, 오류정정 코드워드를 위하여 4개의 심볼로지(12개 바)를 적용하면, 66개 바에 정보만을 수록할 수 있게 되므로 숫자만 사용할 경우에 약 33자리의 숫자정보를 적용할 수 있다. 이와 같이 정의된 내용을 바탕으로 오류정정 코드워드를 생성하기 위하여 소프트웨어 모듈로 구현하는 방법을 채택하였는데 이는 다량 우편이용자도 손쉽게 활용할 수 있도록 하기 위해서이다.

오류정정 코드워드를 생성하는 기존의 방법 중에서 2차원 바코드인 PDF-417에 적용을 살펴보면 다음과 같다[8].

오류정정 레벨 1 을 위한 생성 다항식 차수계산

s	=	1	오류 정정 레벨 1
k	=	$2^s + 1 = 4$	오류 정정 문자의 수
$a_k(x)$	=	$(x - 3)(x - 3^2)(x - 3^4)(x - 3^8)$	
	=	$59049 - 29160x + 3510x^2 - 120x^4 + x^8$	
α_0	=	$59049 \text{ mod } 64 = 41$	
α_1	=	$-29160 \text{ mod } 64 = 24$	
α_2	=	$3510 \text{ mod } 64 = 54$	
α_3	=	$-120 \text{ mod } 64 = 8$	

1. 데이터 코드워드 순서로 식별 $d_{n-1}, d_{n-2} \dots d_0$
2. 오류정정 코드워드의 초기화 E_0, \dots, E_{k-1} 의 값 = 0
3. 각 코드워드에 대하여 $d_i = d_{n-1} \dots d_0$:

```

BEGIN
    t1 = (d_i + E_{k-1}) mod 64
    오류정정 코드워드에 대한 E_j = E_{k-1} ... E_1 계산
    BEGIN
        t2 = (t1 x alpha_j) mod 64
        t3 = 64 - t2
        E_j = (E_{j-1} + t3) mod 64
    END
    t2 = (t1 x alpha_0) mod 64
    t3 = 64 - t2
    E_0 = t3 mod 64
END
    
```

4. 각 오류정정 코드워드에 대하여 $E_j = E_0 \dots E_{k-1}$, 의 값을 계산

```

BEGIN
    If E_j > 0
        E_j = 64 - E_j
    END
    
```

그림 2 기존의 오류정정 코드워드 생성

Fig. 2 Legacy Generation Method of Error Correction Codeword

(그림 2)와 같은 방법에 의하여 오류정정 코드워드를 생성할 경우에는 초기 다항식부터 덧셈, 곱셈, 나눗셈 등에 의한 나머지 값의 계산 등 복잡한 연산 방법을 적용하여야 한

다. 이러한 연산과정으로 인하여 오류정정 코드워드를 계산하기 위한 시간이 많이 소요되는 문제점을 내포하고 있다. 또한, 바코드 심볼로지의 판독과정에서도 상기와 같은 연산 방법을 적용할 경우 판독을 위한 계산시간이 많이 소요된다. 이러한 문제점을 해결하기 위한 방법으로서 기초 다항식을 생성하는 과정부터 16 bit 연산방법을 적용하여 오류정정 코드워드의 생성을 위한 계산 시간의 최소화와 바코드 심볼로지의 디코딩과정에서도 이 방법을 적용하여 판독성능을 향상시키고자 한다. 이에 따라, Galois Field $GF(2^6)$, $f(a) = 1 + a + a^6 = 0$ 에 의하여 지수 값에 따른 바이너리 값으로 표현하면, 0부터 $\dots, a^{62}, 1 + a^5, (100001), a^{63} = 1$ 까지의 값이 생성된다.

$0(000000), 1(100000), a(010000), a^2(001000), a^3(000100), a^5(000001), a^6, 1 + a(110000), a^7, a + a^2(011000), \dots$

기존의 오류정정 코드워드를 생성하기 위한 일반 다항식에 α 지수의 값에 의하여 연산하는 방법을 다음과 같은 수식을 이용하여 산출하였다.

$$\begin{aligned}
 g(x) &= (x + \alpha)(x + \alpha^2)(x + \alpha^4)(x + \alpha^8) \\
 \alpha^2 + \alpha &= \alpha^9 \\
 \alpha^4 + \alpha^8 &= \alpha^5 \\
 &= (x^2 + (\alpha^2 + \alpha)x + \alpha^3)(x^2 + (\alpha^4 + \alpha^8)x + \alpha^7) \\
 &= (x^2 + \alpha^9x + \alpha^3)(x^2 + \alpha^5x + \alpha^7) \\
 &= x^4 + \alpha^9x^3 + \alpha^5x^2 + \alpha^3x + \alpha^7 \\
 &\quad + \alpha^9x^3 + \alpha^6x^2 + \alpha^12x \\
 &\quad + \alpha^7x^2 + \alpha^4x + \alpha^{10} \\
 &\quad \dots \\
 &= x^4 + \alpha^9x^3 + \alpha^4x^2 + \alpha^{14}x + \alpha^{10}
 \end{aligned}$$

연산의 결과로서 α 지수의 계수 값인 1(제외), 19, 41, 24, 10을 사용 원시 다항식의 계수 값을 기본으로 하여 데이터들에 의한 오류정정 코드워드 값을 생성할 수 있다. 이 값을 4개의 오류정정 코드워드를 생성하기 위한 기준 값으로 사용하기 위한 것이다. 이 값들을 이용하여 입력된 데이터를 encoding 할 수 있으며, 다음과 같은 절차에 의하여 수행될 수 있도록 하고자 한다. 이와 같은 방법에 의하여 α 에 대한 계수 값들을 이용하여 연산할 수 있도록 하였다. 이 결과를 이용하여 오류정정 코드워드를 생성하기 위해서는 다음과 같은 절차를 적용한다. 상기 과정에서 지수의 계수값과 일치되는 바이너리 값들을 호출하기 위한 참조 테이블1을 생성하고, 바이너리 값의 순으로 배열한 참조 테이블2를 생성하였다. 이는 연산 과정에서 지수의 계수의 값에 의하여 바이너리 값을 호출하여 16bit Exclusive OR 연산하기 위한 방법과 연산결과를 다시 지수의 계수 값으로 치환하기 위하여 사용하였다.

(그림 3)은 다음과 같은 절차에 의하여 오류정정 코드워드를 생성하게 된다.

- 1) 3개 바들에 의한 심볼로지를 값인 데이터에 해당하는 16진수와 α 의 계수 값의 16진수 값(참조테이블 1)을 Exclusive OR 연산 수행결과를 획득한다. 만일, 결과 값이 0x22 이라 가정하면, 이에 대한 10진수인 값 34가 획득된다.

- 2) 이 값은 bit 연산이 종료된 값을 다시 지수 함수의 계수 값을 산출하기 위하여 참조 테이블2를 이용하여 16 bit 값에 대응하는 α 의 계수 값을 획득한다.

예를 들어, 16진수 Exclusive OR 연산결과가 0x22이면 이 값은 10진수 34 이고, 참조테이블2에서 34번째 값을 취하면 "24"가 된다. 즉, α^{24} 가 획득된다.

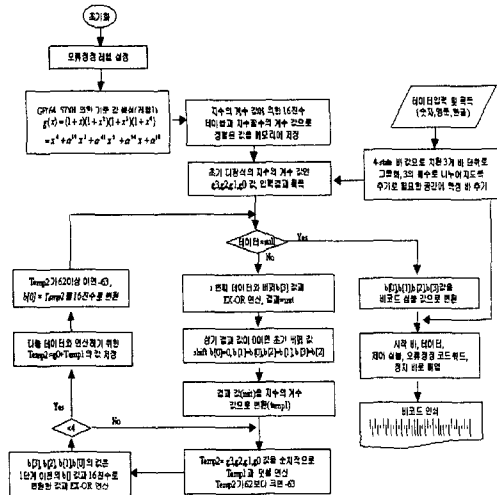


그림 3 오류정정 코드워드 생성 방법
Fig. 3 Generation Method of Error Correction Codeword
이와 같은 방법에 의하여 오류정정 코드워드를 생성방법을 적용할 경우에 나눗셈과 곱셈연산에 의한 계산시간을 최소화 할 수 있다. 또한, 지수의 계수 값들에 의한 연산으로 대체할 경우에 10진수 데이터를 16진수 bit로 변환하여 Exclusive OR 연산으로 대체하여 사용하면 됨을 알 수 있다.

IV. Reed Solomon Decoding 방법에 적용 및 시험

시험용 Reed Solomon decoding 시스템은 바의 값 중에서 시작 바 및 정지 바를 제외하고, 수신된 데이터에 의하여 decoding 하는 절차 중에서 syndrome 값을 구하는 절차에 16 bit Exclusive OR 연산방법에 의하여 수행되도록 구성하였다. Syndrome 값을 생성하는 절차는 다음과 같다.

- 1) 수신된 데이터의 총 길이 D_i 를 산출하고, 3개 바 단위로 그룹핑하여 이들의 값들을 구하여 buffer에 저장한다.
- 2) 수신된 데이터는 $[D_0, D_1, D_2, \dots, D_M]$ 로 정의된다.
- 3) 이 값들에 의하여 syndrome 값을 구하기 위하여

$$\begin{aligned}
 D_j / 3 \\
 Temp[j] &= \sum_{j=0}^2 \left[\sum_{i=0}^2 b_i 4^i - 1 \right] \\
 \therefore \text{만일 } b_i &= 4 \text{ 이면, } Temp[j] = 64, D_i = \text{수신된 데이터의 총길이}
 \end{aligned}$$

$GF(0)$ 부터 $GF(63)$ 까지 연산하는 과정에서 $D_j/3$, 즉 디코딩 대상 값들의 수만큼 반복적으로 다음과 같은 방법으로 수행한다. 이 syndrome 값은 수신된 데이터의 오류를 비교하기 위한 기본 데이터로서 적용되어야 한다.

$$\begin{aligned}
 \text{Syndrome}[s] &= \sum_{i=0}^{GF_{NUM}-1} \text{Syndrome}_{TEMP}[s], \quad |GF_{NUM} = 64 \\
 \text{Syndrome}_{TEMP}[i] &= \sum_{s=0}^{D_i-1} \{GF_{SUM}(GF_{MUL}(GF_{SUM}(Temp_{2(1)}, D_i), s), D_0)\} \\
 &\quad \because Temp_{2(1)}|_{i=0} = 0 \\
 \text{o 덧셈 연산} \\
 GF_{SUM}(Temp_2, D_i) &= GF_{EXP}(Temp_{2(16)} \text{ Ex-OR } D_{(16)}(0,62)) \\
 \text{만일 } Temp_2 = 0, D_i = 64 \text{ 이거나} \\
 Temp_2 = 64, D_i = 0 \text{ 이면, } GF_{SUM}(Temp_2, D_i) &= 0 \\
 \text{만일 } Temp_2 = 0, Temp_2 = 64 \text{ 이면, } GF_{SUM}(Temp_2, D_i) &= D_i \\
 \text{만일 } D_i = 0, D_i = 64 \text{ 이면, } GF_{SUM}(Temp_2, D_i) &= Temp_2 \\
 \text{o 곱셈 연산} \\
 GF_{MUL}(Temp_1, s) &= \text{result} = (Temp_1 - 1) + (s - 1) + 1 \\
 \text{만일 } \text{result} \geq GF_{NUM} \text{ 이면, } GF_{MUL}(Temp_1, s) &= \text{result} - GF_{NUM} + 1 \\
 \text{만일 } Temp_1 = 0 \text{ 이면, } GF_{MUL}(Temp_1, s) &= 0 \\
 \text{만일 } s = 0 \text{ 이면, } GF_{MUL}(Temp_1, s) &= 0
 \end{aligned}$$

이와 같은 방법에 의하여 64개의 syndrome 값이 생성할 수 있다. $GF_{EXP}(Temp_{2(16)} \text{ Ex-OR } D_{(16)}(0,62))$ 의 연산은 encoding 과정에 기술된 16 bit들로 변환한 후, 이 값들을 Exclusive OR 연산을 수행한 것이며, 이 결과 값이 16 진수이므로 GF 함수에 대응하는 지수의 계수 값을 산출하는 방법을 적용한 것이다.

이와 같은 방법에 의하여 Syndrome 값을 산출하는 과정에서 나눗셈에 의한 나머지 값을 산출하고, 곱셈하는 방법보다 연산과정이 단순하게 된다. 이 syndrome 값을 근간으로 하여 Berlekamp-Messey 알고리즘을 이용하여 오류위치 다항식의 값을 구하고, Recursive Extension 및 Inverse Transform 방법을 적용하여 오류정정 및 판독시험을 하였다.

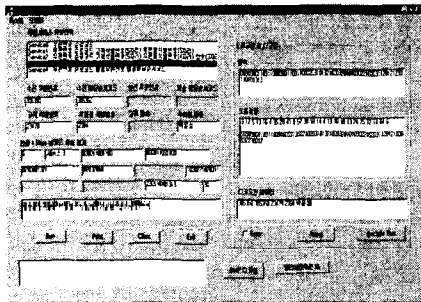


그림 4. 4-state 바코드 인쇄 및 판독시험 시스템
Fig.4 4-state Bar Code Printing and Reading Prototype System

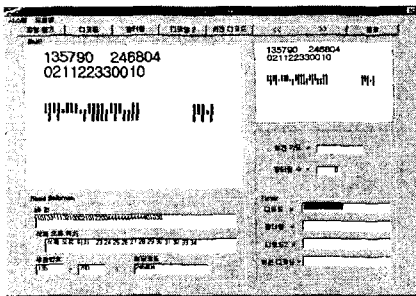


그림 5. 4-state 바코드 비트맵 이미지 판독시험 시스템
Fig.5 4-state Bar Code Bitmap Image Reading Prototype System

Reed Solomon Decoding 모듈에 본 논문에서 제시한 방법을 적용하여 시험 시스템을 개발하여 시험한 결과 오류검출 및 정정 범위를 그대로 유지함과 동시에 바코드 심볼로지 값에 대한 오류검출 및 오류정정을 위해 수행되는 연산 시간도 기존의 나눗셈, 곱셈 등의 연산수행 시간만큼 소요되었던 시간만큼 축소될 것으로 기대된다. (그림 5)는 4-state 바코드 심볼로지 중에서 12개의 바가 소거된 시험용 비트맵 이미지를 이용하여 판독 시험한 결과로 디코딩을 위하여 소요되는 시간은 약 47 msec가 소요되었으며, Reed Solomon Decoding 과정에서 소요된 시간은 1.6 msec이내에 수행됨을 확인하였다. 이 결과를 바탕으로 이송중인 실제 우편봉투의 이미지를 획득하는 기능을 추가하여 우편물 자동구분처리용 바코드 판독기에 적용될 수 있을 것으로 기대된다.

V. 결론

본 논문에서는 4-state 바코드의 신뢰성 및 판독성능 개선을 목적으로 오류정정 코드워드 생성방법이 보다 효과적으로 계산하는 방법을 제시한 것이다. 이 4-state 바코드 인코딩 시스템은 우편물을 자동 구분 처리하는 범위를 확장시키기 위한 목적으로 개발된 것이다. 이 연구결과는 우편물의 자동구분을 위하여 우편주소 인식결과에 대한 우편번호 및 배달순서코드를 인쇄하기 위하여 적용될 예정이다. 그리고, 우편물을 자동구분 처리 촉진을 위하여 다량 우편 이용자에게도 제공될 예정이다. 추후 연구사항으로는 본 논문에서 제시한 인코딩 방법에 의하여 우편봉투에 인쇄된 바코드를 고속으로 판독하기 위한 기술 개발 및 우편물 자동구분 처리뿐만 아니라 다양한 부가서비스를 생성하여 고품질의 우편서비스를 제공하기 위한 연구가 요구된다. 그리고, 기존의 자동구분 처리시스템은 초당 10 ~ 13통이 처리되어야 하므로 바코드 이미지 전처리 방법을 효과적으로 수행할 수 있도록 하기 위한 연구가 필요하다.

참고문헌

- [1] 박문성, "우편물 자동처리 촉진을 위한 한글 4상 4 바코드에 관한 연구", 한국정보처리학회 제2회 시스템통합연구회, pp.92-96. 1998.7.
- [2] 박문성, "우편용 한글 4 State 바코드", 제4회 우정workshop, pp.139-152. 1998. 9.
- [3] 박문성, 송재관, 우동진 "우편물 자동처리 촉진을 위한 3 of 5 고객 바코드 검증 시스템", 한국정보과학회 학술발표논문집 제 25권 2호, pp.496-498. 1998.11.
- [4] Donald Clydale, "Canada Post Corporation's Point of Call Identifier", Advanced Technology Conference, Vol 2, pp.771-782, December 1992.
- [5] J. Strohmeyer, J. Nice, "Carrier Sequence Bar Code Sorter", Advanced Technology Conference, Vol 2, pp.1061-1074, December 1992.
- [6] Royal Mail, "Mailsort Customer Barcoding Technical Specification", OXFORD, pp.1-pp.11. April 1995.
- [7] Rolland D. Culp, "High Density (2D) Bar Codes and EDI Enhance U.S. Mail Manifesting", pp.643-657. Advanced Technology Conference, Vol. 2, December 1992.
- [8] Shu Lin, Daniel J. "Error Control Coding: Fundamentals and Applications", Prentice-Hall Series, 1983.