# 회전 및 거리에 무관한 바코드 영상인식 알고리즘에 관한 연구

김기순, 최종문, 김준식
호서대학교 전기정보통신공학부

# A Study on the Bar-Code Image Recognition Algorithm unrelated to Rotation / Distance

Kee-Soon Kim, Jong Moon Choi, Joon-Seek Kim
Dept. of Electrical and Information Telecommunication Engineering, Hoseo University

## 요 약

본 논문에서는 산업용으로 주로 사용되고 있는 바코드(code 93)를 비전시스템을 이용하여 자동으로 인식할 수 있는 알고리즘을 제안하였다. 제안된 알고리즘은 입력된 바코드 영상에 대해 회전에 관계없이 각도를 자동 추출하여, 모듈(module)을 구성하는 화소들을 추출한다. 각 모듈에 대해 적응적인 방법으로 바(bar)와 스페이스(space)를 구성하는 엘리먼트(element) 값을 구하고, 심벌 문자들의 엘리먼트 값을 9개의 그룹으로 나누어 바코드 값을 인식한다. 여러 종류의 바코드 영상을 대상으로 모의실험을 수행하여, 제안한 알고리즘의 성능을 검증하였다.

**Abstract** - In this paper, we proposed the automatical recognition algorithm of bar-code using a vision system, which can be used in the industrial application(code 93). The proposed algorithm extracts the pixels which consist of the bar-code modules unrelated to rotation, then that obtains the elements which consist of bars and spaces. After the obtained elements are divided by nine group, the value of bar-code is recognized. The performance of proposed algorithm is verified through the simulation. The proposed one has good performance.

## I. 서론

최근 바코드는 백화점이나 편의점, 슈퍼마켓 등 유통 분야에서 판매 관리와 자동 수 발주 관리 업무에 사용되고 있으며, ID카드에도 바코드를 부착하여 출입관리, 도서와 도면 대여 관리 등에 활용하고 있다[1]. 본 논문에서는 세계적으로 어느 정도 표준화가 이루어져 사용되고 있는 바코드 중에서 산업용, 특히, 자동차, 의료, 국방, 선박 등에서 사용되고 있는 바코드(code 93)를 자동 인식하는 알고리즘에 대해 연구하였다. 인식 방법은 바코드의 회전에 무관하게 바코드의 영역을 찾아 여러 라인으로 스캔한 값을 참조하기 때문에 오독률을 최소화할 수 있다. 줌 렌즈에 의한 확대가 가능하기 때문에 스캔거리에 무관하고, 바와 스페이스 넓이의 변화에 대한 영향을 최소화시킬 수 있으며, 바와 스페이스상의 백색(salt & pepper)잡음이나, 흐리게 인쇄된 바코드의 경우는 영상처리 알고리즘을 적용하여 정확한 값을 추출할 수 있다. 또한 공장 자동화 시스템과 같이 반복적인 작업에 본 논문에서 제안한 시스템을 적용하면 고가의 바코드 리더를 대체하는 효과를 얻을 수 있으며, 바코드 인식률도 높일 수 있고, 인식된 바코드 정보를 바로 데이터 베이스화하거나 응용할 수 있는 장점이 있다.

## II. 바코드(code 93)의 구조

바코드는 다양한 폭을 가진 바(bar, 검은 막대)와 스페이스(space, 흰 막대)의 배열 패턴으로, 정보를 표현하는 부호 또는 부호체계이다. 바코드 심벌의 구조상 최소단위는 모듈 또는 X 디멘젼이라고 부르며, 1개 또는 여러 개의 모듈이 모여서 바와 스페이스를 만드는데, 이들을 엘리먼트라고 한다[1]. 특히, code 93의 심벌로지(문자를 나타내는 규칙)는 모든 ASCII 데이터를 표현할 수 있는 연속형 심벌로지이다. 그림 1에는 code 93의 심벌 구조를 나타내고 있으며, 시작 문자(start), 데이터, 검증 문자(C), 검증 문자(K), 종료 문자(stop), 종료 바로 구성된다. 종료 문자로 사용될 경우에는 뒤에 종료 바(1X)가 추가되어 비트 패턴이 10X가 된다. 심벌 문자들은 9개의 모듈(9X)로 구성되며, 6개의 엘리먼트(3개의 바와 3개의 스페이스)로 구성된다.

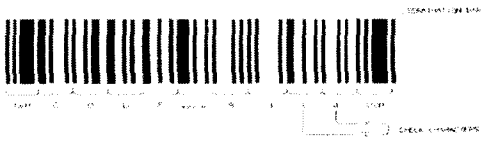


그림 1. Code 93의 심볼 구조

## III. 제안한 바코드 인식 알고리즘

1. 전체적인 구성

그림 2는 본 논문에서 제안한 알고리즘의 전체적인 흐름도를 나타내었다. 우선 바코드 영상을 입력한 후 이진화[2] 과정을 통해 이진영상으로 변환을 한다. 이 과정에서 발생하는 오류를 여러 가지의 형태의 마스크를 이용하여 보정하고, 바코드 영상의 외각라인을 검색하여 중심점의 좌표를 구한 후 중심점 좌표를 중심으로 31×31 블록을 적용하여 블록의 외각라인을 시계방향으로 검색하여 시작점(바)에 해당하는 좌표를 구한다. 시작점의 좌표로부터 chain code의 방향성분을 이용하여 바의 양단 좌표((x1, y1), (x2, y2))를 구한 후 두 좌표를 이용하여 기울기를 구한다. 기울기가 $45^\circ < \theta < 135^\circ$ 일 경우, 바의 양단 좌표를 지나가는 법선을 구해 법선의 y축 접점 두 개를, 기울기가 $0^\circ \le \theta \le 45^\circ$ 이거나 $135^\circ \le \theta < 180^\circ$ 인 경우, x축 접점 두 개를 이용하여, 두 접점을 11등분한 후 바코드를 라인 스캐닝할 좌표 10개를 구한다.

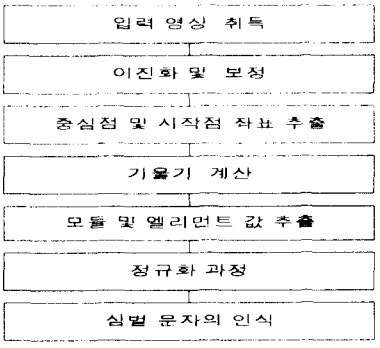
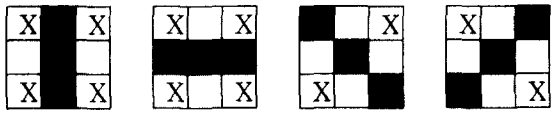



그림 2. 제안한 알고리즘의 전체 흐름도

이렇게 얻은 바코드에 관한 모듈 값 및 엘리먼트 값을 이용하여 바코드의 시작과 종료의 바가 정상적인지 확인 후, 비정상적일 경우, 앞 뒤 데이터를 반대로 입력한 후 정규화 과정을 거친다. 즉, 모듈 값의 배수에 관한 수로 정규화 값을 구하고, 정규화 값을 이용하여 심벌 문자를 인식한다.

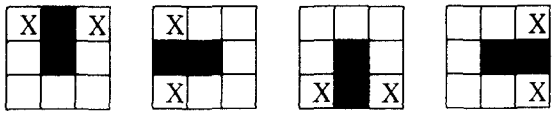
2. 이진화 및 보정

본 논문에서는 비트 패턴을 모듈로 변환 후 모듈의 X(평균값) 문자를 제거하는 정규화 과정을 통해 모듈의 배수만으로 심벌을 구분하는 알고리즘을 제한하였다. 입력된 바코드 영상은 256 level을 갖는 명암 영상으로, 제안된 알고리즘을 적용하는데 용이하도록 입력영상을 이진화하여 사용하였다. 바코드 영상은 이진화를 거치면서 또는 회전에 의해 바 성분 이외의 잡음이 발생한다. 따라서 정확한 좌표값을 얻기 위하여 방향성분에 영향을 줄 수 있는 잡음을 제거하고, 오목한 부분은 흑화소를 추가하는 과정을 거친다. 그림 3은 한 화소 두께의 바를 복원하는 여러 형태의 마스크를 도시했다. 그림 3(a), (b)에서는 수직/수평 방향으로 한 화소 두께로 존재하는 바를 복원하고, 그림 3(c), (d)는 대각방향으로 한 화소 두께로 존재하는 바를 복원한다. 여기서 X는 don't care이다. 복원방법은 마스크에서 don't care가 아닌 부분을 흑화소로 변환한다. 그림 4에서는 볼록한 부분을 제거하는 여러 형태의 마스크를 나타내었다. 그림 5에서는 대각으로 놓여있는 흑화소를 제거하는 마스크의 여러 형태를 도시했으며, 중심의 흑화소를 제거하여 복원한다. 그림 6은 오목한 부분을 복원하기 위한 마스크이다.



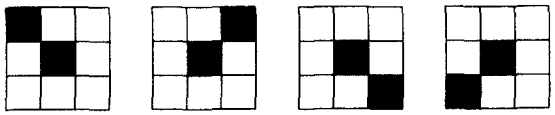
(a) 수직 (b) 수평 (c) 사선 (d) 사선

그림 3. 한 화소 두께의 바 복원 마스크



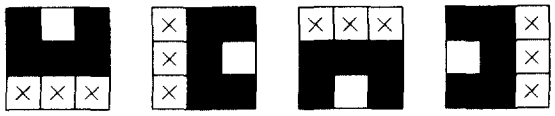
(a) (b) (c) (d)

그림 4. 볼록한 부분 제거 마스크



(a) (b) (c) (d)

그림 5. 독립적인 대각 성분 제거 마스크



(a) (b) (c) (d)

그림 6. 오목한 부분 복원 마스크

3. 중심점 및 시작점 좌표 추출

입력 영상에 대해 바코드와 문자영역을 포함하는 최소 사각형의 좌표를 구하게 되고, 최소 사각형의 좌표를 이용하여 중심좌표를 구한다. 이 중심좌표를 기준으로 31×31 블록에 해당하는 영상의 외각 부분만을 시계방향으로 검색하면서 화소값이 백화소에서 흑화소로 변하는 두 개의 화소좌표 ((start_x, start_y), (start_x2, start_ y2))를 구한다. 이 좌표는 바코드를 구성하는 바의 외각점이 되며, 이 좌표로부터 chain-code[3, 4]를 적용한다. 두 번째

좌표 (start_x2, start_y2)는 첫 번째 찾아진 바가 끊어진 경우를 대비해 구한 좌표이다. 여기서 31×31 블록을 사용한 이유는 바코드를 구성하고 있는 엘리먼트의 최대 넓이의 화소수는 대략 14화소 정도가 된다. 따라서 중심좌표(point_x, point_y)를 중심으로 31×31 블록으로 검색하면 반드시 두 개 이상의 바를 검색할 수 있다.

4. 기울기 계산

바의 양쪽 좌표 (start_x, start_y)를 시점으로 chain-code[4-6]를 구한다. chain-code는 바가 끊어진 경우 정확한 좌표를 구할 수 없다. 따라서 바의 평균적인 길이(460화소)를 이용하여 chain-code에서 찾아진 바의 길이와 평균길이의 오차가 ±5% 이상이면, 앞에서 구한 두 번째 중심좌표 (start_x2, start_y2)를 이용하여 chain-code를 구한다. 시점 (start_x, start_ y)에서 구해진 방향 성분값을 이용하여 바의 양 끝점의 좌표값을 찾아간다. 표 1은 정방향의 좌표값 변환, 표 2는 역방향의 좌표값 변환을 나타내고 있다.

표 1. 정방향 좌표변환

| 방향성분 | 좌표값 변환 |
|---|---|
| 0 | x-1, y |
| 1 | x-1, y+1 |
| 2 | x, y+1 |
| 3 | x+1, y+1 |
| 4 | x+1, y |
| 5 | x+1, y-1 |
| 6 | x, y-1 |
| 7 | x-1, y-1 |

표 2. 역방향 좌표변환

| 방향성분 | 좌표값 변환 |
|---|---|
| 0 | x+1, y |
| 1 | x+1, y-1 |
| 2 | x, y-1 |
| 3 | x-1, y-1 |
| 4 | x-1, y |
| 5 | x-1, y+1 |
| 6 | x, y+1 |
| 7 | x+1, y+1 |

바의 양 끝점의 좌표 (x1, y1), (x2, y2)은 정/역방향으로 진행하면서 방향성분이 125° 이상의 방향 바뀜이 발생하는 좌표까지 각 방향 성분 개수의 합에 의해 구한다. 두 좌표를 이용하여 바의 기울기 $\theta = \tan^{-1}((y_2 - y_1/(x_2 - x_1))$를 구하고, 이 기울기를 이용하여 바의 양 좌표를 지나는 법선의 방정식에 의해 스캐닝할 좌표를 구한다. 두 좌표의 차를 11등분하여 10라인으로 스캐닝을 실시하면, 바코드의 회전 보정없이 자동으로 기울어진 각도를 인식하여 바코드의 값을 검색할 수 있다. 10개의 바코드 라인 스캐닝 값을 저장하고 있는 각 배열에서 바를 구성하는 흑화소 개수와 스페이스를 구성하는 백화소 개수를 카운트한 후 저장한다. 이 과정은 첫 번째 흑화소에서 마지막 번째 흑화소까지 실행한다. 결과적으로 10개의 배열에는 바에 해당하는 흑화소의 카운트 값과 스페이스에 해당하는 백화소의 카운트 값을 저장한다.

5. 모듈 및 엘리먼트 값 추출

10개의 배열을 순서에 따라 각각의 평균값을 구하여 하나의 배열에 저장한다. 이때 각 배열 값이 평균값의 50%-150% 사이의 값을 갖지 않을 경우, 해당 배열 값을 평균값으로 대치한 후 평균을 다시 계산한다. 바코드를 이진화하는 과정에서 나타날 수 있는 이웃하는 두 바가 서로 붙어 넓어지는 경우와, 바의 넓이가 좁아지는 경우를 보정하기 위한 것이다. 심벌의 앞뒤가 바뀌었을 경우, 최대 넓이를 갖는 바는 시작 바로부터 다섯 번째에 위치하고, 마지막 바로부터 뒤에서 3번째에 존재하는 특징을 이용하여 찾을 수 있으며, 배열값을 수정한다. 본 논문에서 심벌 문자별로 인식하기 위해 배열에 저장 되어있는 바와 스페이스를 구성하는 엘리먼트(바와 스페이스를 구성하는 화소의 카운트 값) 값을 6개씩 나눈 후 정규화 과정을 거치게 된다. 우선 전체 평균값을 구한 후 각 배열의 값과 비교하여 평균값보다 작은 경우, 평균값으로 대치시킨다. 여기서 평균값은 엘리먼트(6개)에 의한 평균값이 아니라, 모듈(9개)에 의한 평균값이다. 여기서 모듈 값은 식 (1)에서와 같이 심벌 문자를 구성하는 모든 화소의 합과 심벌 문자를 구성하는 모듈의 개수의 비로 구할 수 있다.

$$\text{모듈 값}(X) = \frac{\text{심벌 문자를 구성하는 모든 화소의 합}}{\text{심벌 문자를 구성하는 모듈의 개수}} \quad \cdots (1)$$

$$\text{엘리먼트 값} = \begin{cases} \text{바를 구성하는 화소의 개수} \\ \quad = (1X,\ 2X,\ 3X) \\ \\ \text{스페이스를 구성하는 화소 개수} \\ \quad = (1X,\ 2X,\ 3X,\ 4X) \end{cases} \quad \cdots\cdots (2)$$

6. 정규화 과정

이웃하는 바와 스페이스의 폭은 서로 상대적으로 영향을 준다. 즉, 이웃하는 엘리먼트 값에 가장 영향을 많이 받게된다. 따라서 적응적인 방법을 적용하여 각 6개의 엘리먼트 값을 정규화한다. 즉, 식 (2)에서 보이는 모듈의 배수에서 수식적으로 $X$를 제거하는 것을 말하며, 본 논문에서는 정규화 값이라 부른다. 하지만, 실제 카메라로 취득한 바코드 영상의 엘리먼트 값에서 나오는 몫은 정수가 되지 않으므로 이 값들을 정수화시키는 과정으로 볼 수 있다. 식 (1)에서 모듈 값을 얻을 수 있으며, 6개 엘리먼트의 값을 모듈 값과 비교하여 모듈 값보다 작을 경우, 정규화 값은 1이 되며, 엘리먼트 값과 모듈 값 차의 절반을 이웃하는 엘리먼트 값이 모듈 값보다 클 경우 빼주고, 모듈 값보다 작을 경우 그대로 둔다. 다음으로 6개의 엘리먼트를 순서대로 2개씩 3개의 그룹으로 나누어 각 그룹의 합을 구하게 된다. 이 합을 모듈 값과 비교하여 근사적으로 몇 배수인지 확인을 한다. 2배수의 경우 정규화 값은 1과 1이고, 나머지 값들은 식 (3)에 의해 각 엘리먼트 값을 근사화시킨 후 정규화 시킨다. 식 (3)에서 $\text{element}_1$, $\text{element}_2$는 엘리먼트 값이다.

$$\frac{(\text{element}_1)}{(\text{element}_1 + \text{element}_2)} \times \text{모듈n 배수},$$

$$\frac{(\text{element}_2)}{(\text{element}_1 + \text{element}_2)} \times \text{모듈n 배수},\ n=3,\ 4 \cdots (3)$$

7. 심벌 문자의 인식

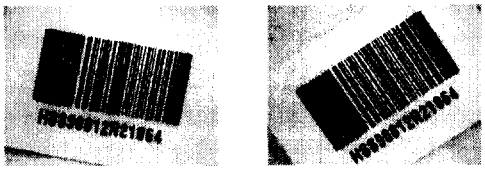
엘리먼트를 저장하고 있는 배열의 모든 값을 정규화 값으로 변환 후, 모든 심벌 문자의 정규화 값들 중에서

첫 번째, 두 번째 자리의 정규화 값을 이용하여 9개의 그룹((1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2))으로 나누고, 나머지 4자리를 가지고 고유값을 (hexa 값) 만들어 데이터베이스의 인덱스와 상응하는 심벌 문자로 인식한다. 구해진 심벌 문자들 중에서 뒤에서 2번째와 3번째 심벌 문자는 검증문자로서 인식된 심벌의 에러 유무를 확인한다. 따라서, 심벌 문자들을 뒤에서 4번째까지 가중치를 주어 구한 심벌 문자와 뒤에서 3번째 검증문자와 비교하고, 다시 앞에서부터 이 검증 문자까지 가중치를 주어 심벌 문자를 구하고 뒤에서 2번째 검증문자와 비교하여 에러의 유무를 확인하고, 시작/종료 문자와 마찬가지로 전송되지 않는다.

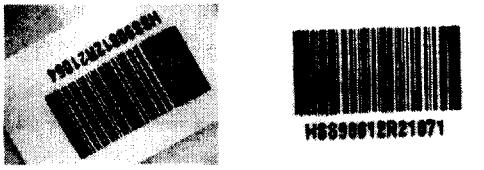
## Ⅳ. 모의 실험 및 결과

1. 실험 환경

본 논문에서 사용되는 영상은 실제 산업용으로 사용되고 있는 바코드를 삼성 NetScan 카메라 (SNC- 80/320)를 사용하여 거리를 달리하면서 영상의 크기는 700×700이고, 256 level을 갖는 명암 영상으로 취득하여 입력영상으로 사용하였다. 그림 7에는 여러 형태로 기울어진 입력 영상들이 나타나 있다. 여기서 기울기는 바코드 영상의 왼쪽에서 수직축으로 라인 스캐닝을 했을 때 가장 먼저 찾아지는 흑화소에서 수직선과 바코드간 시계 반대방향으로의 기울어진 정도를 말한다. 입력영상은 code 93 심벌로지를 갖는 10개의 바코드 영상을 임의로 회전시켜 각각을 20개씩 취득하여 총 200개의 영상을 사용하였으며, Microsoft사의 Visual C++ 6.0을 이용하여 알고리즘을 구현하고 모의 실험하였다.



(a) 입력 영상1 (173°) (b) 입력 영상2 (23°)

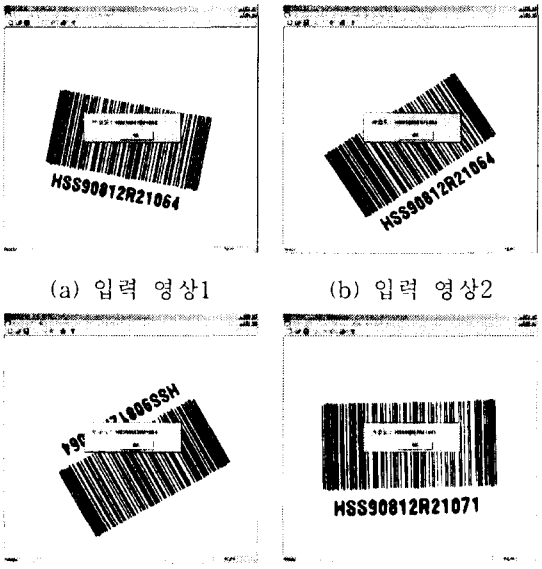


(c) 입력 영상3 (20°) (d) 입력 영상4 (2°)

그림 7. 입력 영상

2. 결과

다음 영상들은 본 연구에서 제안한 알고리즘을 구현하기 위해 Microsoft의 Visual C++ 6.0을 이용하여 구축한 소프트웨어에서 바코드를 인식한 결과를 나타내고 있다. 영상 중간에 있는 다이얼로그 박스에 인식된 심벌 문자들을 나타내고 있다. 결과 영상들을 보면 회전에 무관하고, 앞·뒤의 바뀜이 있는 그림 8(c)에서도 정확한 바코드 값을 인식하는 것을 볼 수 있다.




(a) 입력 영상1 (b) 입력 영상2

(c) 입력 영상3 (d) 입력 영상4

그림 8. 결과 영상

## Ⅴ. 결론

본 논문에서는 산업용(특히, 자동차, 의료, 국방, 상업용 등)으로 주로 사용되고 있는 바코드를 비전시스템을 이용하여 자동으로 인식할 수 있는 알고리즘을 제안하였다. 바코드 영상은 비전시스템에 의해 확대 취득하기 때문에 거리에 의한 제한이 없어지고, 입력된 바코드 영상에 대한 회전 보정없이, 각도를 자동 인식하여 10개의 라인으로 동시 스캐닝하기 때문에 정확한 인식이 이루어졌다. 또한 적응적인 방법으로 "바"와 "스페이스"를 구성하는 6개의 엘리먼트를 9개의 모듈 값 비율에 의해서 정규화시키므로 바와 스페이스 폭에 의한 오독을 최소화 할 수 있었다. 앞으로, 흐리게 인쇄된 바코드 인식에 관한 연구와 알고리즘 적용을 위한 하드웨어 구현이 필요하다.

## 참고 문헌

[1] 오호근, 최신 바코드 기술 및 응용, 성안당, 1997년.

[2] Howard E. Burdick, *Digital Imaging*, McGraw-Hill, 1997.

[3] E. Bribiesca, "A New Chain Code," *Pattern Recognition*, Vol. 32, No. 2, pp. 235-251, 1999.

[4] Hong-Chih Liu and M. D. Srinath, "Corner Detection from Chain-Code," *Pattern Recognition*, Vol. 23-A, No. 1, pp. 51-67, 1990.