

블록 텍스처를 이용한 소포 영상에서 바코드 ROI(Region Of Interest) 추출에 관한 연구

박문성 *, 최호석**, 김진석*, 김혜규*

* 한국전자통신연구원 우정기술연구센터

** 성균관대학교 전자공학과

A Study on the Barcode ROI Extraction Method using Block Texture in Parcel Image

Moon-Sung Park, Ho-Seok Choi**, Jin-Suk Kim, Hea-Kyu Kim

*Postal Technology Research Center, ETRI

**Department of Electronic Engineering, Kyunghee UNIV

mspark@etri.re.kr

요 약

본 논문에서는 블록 다중 텍스처 영상으로부터 바코드 영역을 추출하기 위한 한 방법을 제안한다. 일반적으로 택배 등의 물류 처리에서 사용되는 바코드는 직선 형태의 바로 구성되며, 물체의 윗면에 붙여진 바코드의 방향에 따라 바의 방향은 수직, 수평, 대각선의 방향으로 나타난다. 따라서, 제안된 방법에서는 다양한 텍스처의 특징 벡터를 사용하여 바코드의 특징을 검출한다. 또한 처리 시간의 단축을 위하여 전체 영상을 일정한 블록으로 분할한 후에 국부 특징 마스크를 사용하여 텍스처 특징 벡터를 산출하고, 우편물 영상에서 각각의 특징에 따른 분류를 통해 바코드 영역을 결정한다.

1. 서 론

일반적으로 바코드는 택배, 우편물, 다양한 상품등의 표면에 부착된 라벨로서 물건에 대한 다양한 정보를 저장하는 저장공간으로서의 역할을 하며, 이러한 바코드의 인식을 통해 자동화된 물류처리 및 관리 시스템 등에 활용되고 있다. 특히, 휴대용(hand held)형의 이동식 바코드 인식기뿐만 아니라 컨베이어 벨트 등의 이송 환경 위에서 사용할 수 있는 고정식 바코드 인식기의 경우에도 그 요구가 날로 증대하고 있는 실정이다. 특히, 이러한 고정식 바코드 인식기의 경우에는 바코드의 정보를 해독하기 이전에 바코드의 ROI(Region Of Interest) 추출하는 것이 선결과제이다.

정보 저장매체로서 바코드의 특성은 일정한 크기의 직선이 일정 영역 내부에 나열되어 있는 기하학적 특성을 가지고 있다. 직선을 검출하는 기법으로는 HT(Hough Transform)를 사용하는 기법[1]이 있다. 하지만 직선의 검출은 가능하나 바코드와 다른 캐릭터간의 구별이 힘든 단점이 있다. 또한 웨이블릿 특성을 사용한 기법[2]에서는 웨이블릿 변환 후 3개의 영역에서 각각 처리하므로 많은 처리시간이 소요된다. 본 연구에서는 이러한 바코드의 기본적인 특성에 근거하여 일정한 크기의 블록 내부의 바코드의 텍스처 에너지를 에너지 맵에 나타내고 각각의 텍스처 특징에 따른 분류를 통하여 후보 블록들을 검출 후에 미니 맵에서 레이블링과 확장을 통한 최종 바코드 영역을 산출한다. 본 논문에서 사용한 텍스처 에너지는[3]는 총 16개의 텍스처 특징으로 분류되며 그중 에서 에지 특징과 물결특징에서 바코드의 특징이 확실히 구분되는 특성을 사용하였다. 제 2장에서는 제안된 알고리즘의 절차를 간략히 기술하고, 단계적으로 수행된 결과를 바탕으로 검증하도록 한 것이다. 제 3장에서는 시험된 결과와 추후 연구사항에 대하여 다루었다.

2. 연구배경 및 알고리즘 제안

2.1 연구배경

본 절에서는 인터넷 기반의 전자상거래의 발전으로 인하여 다양한 물품에 대한 배달 서비스가 요구되고 있다. 특히, 소형통상 보다는 소포와 같이 부피가 존재하는 물품을 보다 효과적으

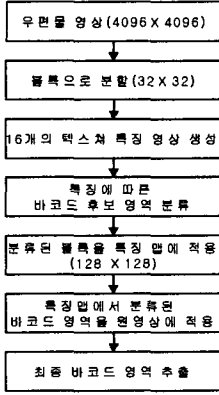
로 배달하기 위한 방법이 요구되고 있다. 우편물 취급과정에서 소포상에 표현되는 정보들이 다양하고, 구분 및 기록관리를 위하여 필요한 정보를 획득하기 위한 수단이 필요하다. 이에 따라, 소포를 구분 및 기록관리 대상 정보의 자동식별을 위한 방법에 적용될 수 있도록 바코드 영역을 추출하는 방법에 대해 다루었다. 그리고, 바코드 판독이 불가능할 경우에 우편주소 영역에 대한 추출이 필요하며 소포의 부피 또는 면적을 획득하기 위한 정보들 동시에 추출할 수 있어야 한다.

본 논문에서는 바코드 ROI 추출과 동시에 소포의 에지와 문자열이 존재하는 영역을 동시에 검출하기 위한 방법을 검토하기 위하여 선행 연구로써 블록 정보처리 기반 텍스처 영상 처리 방법을 통해 바코드 영역을 추출하고, 시험과정에 있어 문자열 또는 소포의 에지 영역검출이 가능한지 검토하고자 한다. 그리고, 바코드 ROI 추출 결과를 바탕으로 중심축 생성 방법을 제시하고자 하여 판독을 위한 기준 정보를 생성하도록 한 것이다.

2.2 알고리즘 제안

소포 이미지(4K x 4K)에 존재하는 다양한 정보들 중에서 바코드 ROI 추출을 위해 모든 영역을 순차적으로 검사하지 않고, 미세블록(32x32)들로 나누어 블록내에 존재하는 특징에 의하여 바코드, 문자열, 소포의 에지 등을 검출하는 방법을 고려하고자 한다. 이와 같은 조건을 만족되도록 각 블록 내에 존재하는 정보들 에지, 물결의 특징을 고려하여 적용하고자 한다. 이에 따라, 획득된 이미지들 32x32블록 단위로 분할하고, 다중 임계값은 각 블록들에 대한 최대, 최소값의 차 값을 구하여 사용한다. 또한, 소포이미지 중에서 컨베이어 벨트를 제거하기 위하여 최대와 최소값이 차이 값이 20 이하인 경우를 검사 대상영역에서 제외하는 방법을 적용하였다. 그리고, 16개의 텍스처 특징들에 의해 소포와 컨베이어 벨트의 경계 영역, 문자열, 바코드 영역을 탐색하기 위한 기준을 적용한 것이다. 이와 같은 텍스처 특징에 의하여 바코드 후보 영역을 분류하고, 분류된 블록들을 이용하여 특징 맵을 구성하고, 바코드 후보영역들의 라벨링 과정을 통

하여 최종 바코드 영역이 추출하는 과정으로 구성하였다. 이 결과물 바탕으로 중심축 라인을 생성하기 위하여 8개 모서리 점들을 기준과 블록들이 생성된 형태를 고려하여 바코드 판독이 가능한 중심축 라인이 생성되도록 하였다. (그림 1)은 바코드 영역 추출 기법의 전체 흐름도이다.

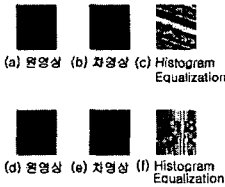


(그림 1) 전체 흐름도

2.3 블록 차영상 검출

원 영상을 32x 32 블록으로 분할 후 영상 주위의 작은 원도우의 이동에 의한 조명의 영향을 제거하기 위하여 각각의 픽셀로부터 블록의 평균을 빼면 픽셀의 평균 강도는 거의 0에 가까운 전처리 영상을 만들 수 있다. 다음의 수식(1)에 의해서 블록의 평균값을 산출하고 차 영상을 구할 수 있다.

$$S(x,y) = \frac{\sum_{x'=0}^{M-1} \sum_{y'=0}^{N-1} f(x',y')}{N \times M} \quad (1)$$



(그림 2) 평균값을 뺀 영상

(그림 2)의 (b)와 (e)는 원 영상(a)와 (d)의 차 영상이고 (c)와 (f)는 차 영상을 히스토그램 균일화한 영상이다. 본 절에서는 Law's의 텍스처 에너지 특징[3]을 사용하였고 다음의 4개의 벡터에서 총 16개의 특징 벡터가 된다.

- L5 (Level) = [1 4 6 4 1]
- E5 (Edge) = [-1 -2 0 2 1]
- S5 (Spot) = [-1 0 2 0 -1]
- R5 (Ripple) = [1 -4 6 -4 1]

여기서 L5 벡터는 국부 평균 무게 중심을 나타내고, E5 벡터는 에지들, S5 벡터는 점들, R5 벡터는 물결을 검출한다. 또한 블록 영상에 적용하기 위한 2D 컨볼루션 마스크는 벡터 쌍들의 외적을 계산해서 얻어진다. 예를 들어, 마스크 E5L5는 다음과 같이 E5와 L5의 외적으로써 산출된다.

$$\begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \times [1 \ 4 \ 6 \ 4 \ 1] = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Law's의 텍스처 에너지는 전처리 후에 16개의 5x5 각각의 마스크들은 전처리 영상에 적용하면 16개의 필터링된 영상들이 생성된다. 픽셀 [i, j]에서 k번째 마스크의 필터링 된 결과를 $F_k[i,j]$ 라 하면, 필터 k에서 텍스처 에너지 맵 E_k 는 수식(2)와 같이 정의된다.

$$E_k[r,c] = \sum_{j=c-16}^{c+16} \sum_{i=r-16}^{r+16} |F_k[i,j]| \quad (2)$$

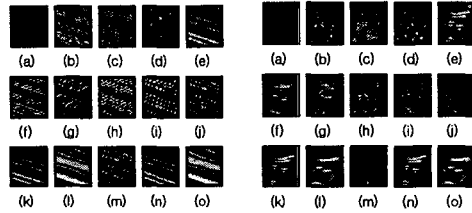
32x32 블록의 평균값으로 4096x4096 영상에서 각각의 텍스처 에너지 맵은 128x128 영상으로 생성된다.

일단 16개의 에너지 맵이 생성되면, 확실한 대칭 쌍은 그것의 평균으로 각각의 쌍을 대체하는 9개의 마지막 맵의 생성으로 조합된다. 예를 들어 E5L5 수평 에지를 측정하고, L5E5는 수직 에지를 측정한다. 이런 두 맵들의 평균은 전체 에지를 측정한다. 9개의 결과 에너지 맵은 다음의 <표 1>과 같다.

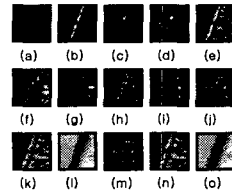
<표 1> 9개의 에너지 맵

L5E5/E5L5	L5S5/S5L5
L5R5/R5L5	E5E5
E5S5/S5E5	E5R5/R5E5
S5S5	S5R5/R5S5
R5R5	

(그림 3)과 (그림 4)에는 바코드, 글자, 물체의 블록 영상에서 16개의 에너지 맵을 보인 것이다.



(그림 3) 바코드 및 문자의 블록 영상



(그림 4) 물체의 블록 영상

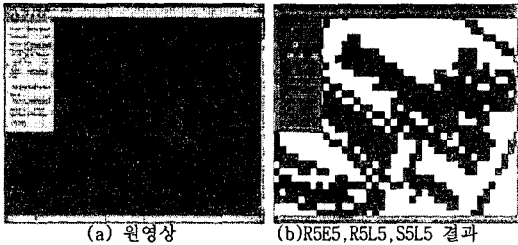
<표 2> 9개의 특징 벡터

	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
바코드	61.5	19.79	50.9	298.7	239.7	408	64.39	131.4	53.27
글자	56.61	18.95	52.72	220.2	130	160.5	33.54	42.21	26.79
물체	18.6	3.6	15.25	75.18	20.38	28.19	6.09	8.22	5.41

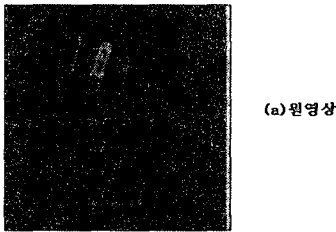
여기서 (a) 원영상, (b) E5E5, (c) S5S5, (d) R5R5, (e)E5L5, (f) S5L5, (g) R5L5, (h) S5E5, (i) R5E5, (j) R5S5, (k) L5E5, (l) L5S5, (m) L5R5, (n) E5S5, (o) S5R5를 각각 나타낸다. <표 2>는 바코드, 글자, 물체의 블록에 대한 9개의 특징 벡터에 의한 평균값을 각각 나타낸 것이다.

본 논문에서는 총 16개의 벡터 중에서 바코드 영역만을 추출하기 위하여 3가지 벡터만을 선택하여 에너지 맵을 생성하였다. 에너지 맵의 특성에 따라 R5L5는 반복성분을 강조하여 나타내므로 바코드의 반복적인 특징이 강하게 나타난다. 또한 R5E5는 에

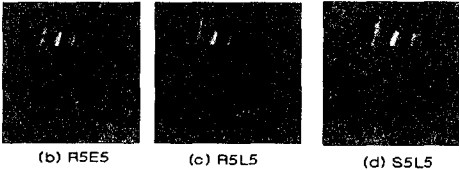
지의 반복을 나타내고, S5L5는 점 성분을 강조한다. (그림 5)에는 전체 영상과 3개의 성분을 나타내는 영상을 보인다. 본 논문에서는 위의 3개의 특징 벡터의 임계치를 사용하여 128×128 영상을 생성하였다. 생성된 영상에서 주변의 잡음을 제거하고 정확한 바코드 영역을 추출하기 위하여 레이블링 기법과 간단한 확장 알고리즘을 적용하였다.



(a) 원영상 (b)R5E5, R5L5, S5L5 결과



(a) 원영상



(b) R5E5 (c) R5L5 (d) S5L5

(그림 5) (a)원영상, (b)R5E5, (b)R5L5, (b)S5L5 결과

바코드 ROI를 보다 구체화하기 위해 바코드 표준크기를 기준으로 최대 2.9x15개(18개 정보가 수록된 경우)인 미세블록 수가 존재하므로 바코드의 최소크기인 3개(시작, 정지, 우편번호 3 자리 등) 이상의 문자가 수록된 경우를 가정하면 5개 미만인 경우에는 바코드가 존재하는 영역이 아니다. 이에 따라, 4개 이하로 connected된 경우에 배제되도록 수식(8) 및 y값의 변환만 발생하는 미세블록 들을 대상으로 4개 이상 connected 그룹의 레이블링하였다.

$$Count(k) = \sum_{x=0}^{x-1} \sum_{y=0}^{y-1} \left| \frac{T(x-1, y)}{Position} \frac{X}{L(x, y+1)} \right|_{Position > 0, X=1 \text{ or } 0}$$

$$Position = T(x-1, y), Position = T(x, y-1) \Rightarrow \forall Count(k)$$

$$Label_{Min} = 4 < \forall Count(k) < Label_{Max} = 50 \quad (3)$$

Count(k)는 현재의 검사 좌표를 기준으로 Position=L(x, y+1) 또는 Position=T(x-1, y)일 경우에 connect된 상태에 의해 판단한다. 그러나, 즉, Count(k)의 개수가 5이상이면, 60미만(정보 수록 양에 따라 조정 필요)일 경우에 바코드 영역일 가능성이 높아진다. 여러 개의 바코드가 존재할 경우를 위해 그룹의 개수를 설정한다.

Count(k)가 Position=L(x, y+1)=T(x-1, y)=0이고, 다음에 Label_{Min}=4 < \forall Count(k) < Label_{Max}=60 조건이 만족되면 그룹의 수를 증가시키도록 하였다.

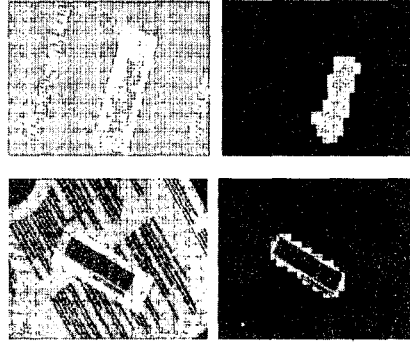
$$Compare_x[k] = |1 \ 1 \ 0 \ -1 \ -1 \ -1 \ 1 \ 1|$$

$$Compare_y[k] = |0 \ -1 \ -1 \ -1 \ 0 \ 1 \ 1 \ 1|$$

$$Count = \sum_{x=0}^{x-1} \sum_{y=0}^{y-1} buff(x + Compare_x[k], y + Compare_y[k]) > 0$$

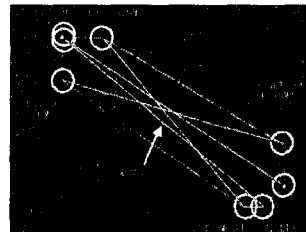
Count > 3이상이면, 추가 레이블링 (4)

(그림 6)에는 임계치를 적용한 영상과 레이블링, 확장을 한 영상을 각각 나타낸 것이다.



(그림 6) 바코드 영역 추출 결과

바코드 ROI 추출된 결과들 이용하여 미세블록들에 해당하는 원본 이미지를 획득하여 검토하고, 정보 해석을 위한 중심축 라인 생성방법을 적용하고자 한다. 미세블록에 의하여 바코드 ROI 영역이 생성된 것이므로 다각형의 형태로 구성된다. 즉, 직사각형의 형태로 추출되도록 하는 방법과 다각형의 외곽좌표인 Top left, Top right, Bottom left, Bottom right, Left top, Left bottom, Right top, Right bottom의 좌표를 기준으로 (그림 7)과 같이 8개의 외곽좌표를 마주보는 면 좌표 중에서 서로 반대가 되는 좌표를 연결하여 가장 긴축을 생성하는 방법을 적용하였다 [3].



(그림 7) 8개 외곽좌표에 의한 중심축 라인

(그림 7)과 같은 결과들 참고하여 가장 긴축을 구하기 위하여 8개의 외곽 좌표(x,y)를 기준으로 연결된 값의 길이는 두 좌표간의 기울기 각도에 의존된다. 기울기 각도θ의 변화량에 따른 값에 의해 길이를 구할 수 있도록 수식(5)로 정의하고 서로 연결된 Point₁(r₁, c₁), Point₂(r₂, c₂) 좌표를 기준으로 축의 길이 D를 산출하여 4개의 길이 중 가장 큰 값을 사용하기 하였다.

$$Q(\theta) = \begin{cases} \frac{1}{|\cos \theta|} : |\theta| < 45^\circ \\ \frac{1}{|\sin \theta|} : |\theta| > 45^\circ \end{cases}, D = \sqrt{(r_2 - r_1)^2 + (c_2 - c_1)^2} + Q(\theta) \quad (5)$$

외곽 좌표의 기울어진 방향에 따라, Top left = Left top, Top right = Right top, Bottom left = Left bottom, Bottom right = Right bottom의 좌표가 일치될 수 있으며, 이 경우에는 이들 값이 중심축 라인의 시작 및 정지 좌표로 적용하였다.

<표 3> 처리 시간 및 성능

종류	블록 처리시간	전체 처리시간
전체 특징 벡터	6ms	33000ms
3개의 특징 벡터	1.2ms	2300ms

시험우편물/ 재투입 수	ROI 검출 상태		바코드 디코딩	
	정확도	잘못 검출된 그룹	성공	실패
150통/3회	99.996%	0.006%	94.9%	5.1%

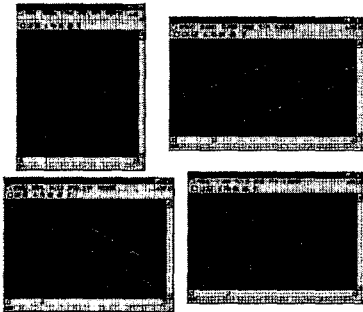
그리고, 8개의 축을 기준으로 중심축을 생성할 경우에 바코드 위치에 따른 기울기 정보를 포함하지 않고, 보다 빠른 시간내에 중심축을 생성하기 위하여 4개의 직선성분 중에서 가장 긴 값을 적용할 수 있는 것과 약간의 보정에 의하여 중심축 라인을 생성이 가능하다. 이에 따라, 바코드가 블록들 내에 어떠한 형태로 존재하는지에 따른 특징에 의하여 선택된 중심축에서 블록의 크기단위로 이동시켜 보정하는 방법을 적용하였다. 가장 긴축의 값에 대응하는 좌표값 기준으로 수식(6)을 적용하여 중심축을 부분적으로 보정할 수 있다.

$$Point_1(r_1, c_1) \leq Point_2(r_2, c_2) \text{ or } Point_1(r_1, c_1) \geq Point_2(r_2, c_2)$$

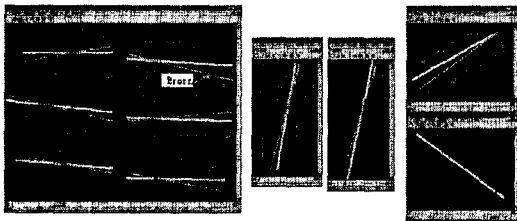
$$Point_1(c_1) = Point_2(c_2) \text{ or } Point_1(r_1) = Point_2(r_2) \text{ 이면}$$

$$Point_1(r_1 + c_1)/2, Point_1(r_2 + c_2)/2 \quad (6)$$

수식 (6)의 조건이 만족될 경우에는 (그림 8)의 (b)와 같게 되며, 나머지는 (a)와 같이 가장 긴축을 사용하면 된다.



(a) 가장 긴 라인 생성 방법



(b) 중심축 라인 보정
(그림 8) 중심 축 라인 생성 결과

이와 같은 경우를 제외한 나머지 부분에 대한 경우에는 보정 없이 적용하여도 됨을 확인할 수 있었다.

3. 실험 결과 및 결론

본 논문에서는 펜티III 450MHz에서 4096x 4096 영상에 대해서 실험하였다. 영상은 Metrox 라인스캔 카메라로 스캔된 영상이다. 다음의 <표 3>에는 블록의 처리시간과 전체 처리시간을 나타낸다. 전체적으로 상당한 처리 시간이 걸리는 것을 볼 수 있다. (그림 5)에서 (그림 8)까지는 바코드 ROI 추출한 결과 영상을 보인 것이다. 중심축 라인 상의 그레이 레벨 값을 순차적으로 획득하면 바코드 정보를 해석할 수 있다. 디코더 성능을 검토하기 위해 MIL에서 제공하는 Code 39, Code 128, Interleaved 2 of 5 등의 라이브러리를 사용하여 판독시험을 수행하였다.

위의 추출된 결과 영상에서 바코드의 중심에 중심점을 표시하였다. 표시된 중심점을 기준으로 디코더에서 바코드의 장축으로 스캔하여 바코드를 해석한다. 배경에 글자나 그림이 있는 영상에서도 상당히 우수한 결과를 나타내는 것을 볼 수 있다. 향후 텍스트 정보를 좀더 활용하면 글자 영역과 그림영역들도 분류가 가능하게 됨을 확인할 수 있었다.

그리고, 이 시험 시스템 성능 개선을 위한 연구가 요구된다. 바코드 영역 추출에 필요한 벡터만을 산출하고 또한 계산량을 줄이기 위해 일차적으로 컨베이어 벨트와 소포 이미지내의 배경영역을 제거하는 기법을 추가하면 상당한 시간적인 단축이 될 수 있다. 또한 각각의 특성에 맞는 벡터를 사용하여 글자 영역과 배경 등을 분류하는데도 사용할 수 있다. 또한, 블록 내의 특징치를 보다 적은 단계로 검사하여 문자열, 소포의 에지 정보 등에 관한 ROI 추출을 위한 연구가 요구된다.

참고문헌

- [1] Y. P. Zhou, C. L. Tan "Hough-based model for recognizing bar charts in document images," Proceedings of SPIE Document Recognition and Retrieval VIII, V.4307, 2001
- [2] 최호석, 정윤수, 박문성, 이용준, 박영태, "웨이블릿 도메인에서 바코드 영역 추출 기법," 한국정보과학회 2001 추계 학술발표회, Vol. 2, pp. 448-450, 2001
- [3] Shapiro and Stockman, "computer vision text book," 2000.
- [4] Gerhard X. Ritter, Joseph N. Wilson " Computer vision algorithm in Image Algebra", pp. 129-130.
- [5] J. R Parker "Algorithms for image processing and computer vision.", pp.250-274.