한국지능정보시스템학회논문지 제12권 제1호 2006년 3월(pp.33~41)

동영상에서 신발 밑창 모델 인식을 위한 인터레이스 제거 및 블록 코드 생성 기법

김철기
부산대학교 디자인학과
(ckkim@pusan.ac.kr)

본 논문에서는 공장 자동화 시스템의 한 예로, 컨베이어 벨트로 흘러 들어오는 생산품을 모델별로 자동 인식하기 위한 방법을 제안하고 있다. 일반적으로 NTSC 방식의 카메라를 사용할 경우 움직이는 물체는 카메라 고유의 잔상이 발생하게 된다. 잔상이 존재하는 영상을 이용하여 효율적인 처리가 불가능하므로 적당한 후처리 방법이 요구된다. 이를 위하여 제안하는 인터레이스 제거 기법을 통하여 잔상을 제거하고, 이진화를 통하여 대략적 물체 영역을 판별한 후 물체를 에워싸는 직사각형 영역을 구한다. 그 후 윤곽선 검출을 거쳐 직사각형 영역을 블록별로 세분화한 후 각 블록별 화소수를 계산하여 평균을 중심으로 재분류한 후 모델 코드를 생성하여 모델 분류를 하였다. 실험결과 본 논문에서 제안하는 방법의 경우 기존의 방법보다 높은 분류 성공률을 나타내었다.

논문접수일 : 2005년 6월 게재확정일 : 2006년 2월 교신저자 : 김철기

1. 서론

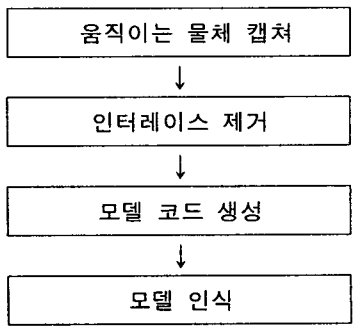
일반적으로 인식 시스템은 모집단의 데이터로부터 패턴이 가지는 특징을 추출하여 인식하는 시스템을 말한다. 기계의 경우 인간의 인식 시스템을 모사하여, 인식의 대상이 되는 형태와 특성을 분석하여 정량적으로 표현한 후 이를 분류하여 인식을 수행하게 된다(김강석 외, 2000). 이와 같은 경우, 인식 시스템의 대상을 패턴이라 하며 측정한 데이터로부터 각 패턴 간에 서로 구별되는 완전한 특징을 추출할 수 있다면 패턴 인식 및 분류가 가능하게 되는 것이다.

컴퓨터 비전 시스템이란 마치 인간의 시각 시스템과 같이 카메라에 기반한 H/W 시스템을 이용하여 2차원 위치 데이터를 입력 받아서 검사 및 제어를 수행하는 방법으로서 비접촉 방식에 해당하며 양질의 검사 결과를 얻을 수 있다는 장점이 존재한다. 따라서 본 논문에서는 이러한 비전 시스템의 장점을 이용하여 지금까지 수동으로 이루어졌던 신발의 밑창(outsole) 길이를 측정하여 제품의 모델에 따른 분류를 수행하는 방법을 제안하고자 한다.

인터레이스 제거 기법(de-interlacing)은 비월주사 방식(interlaced method)의 영상을 순차 주사 방식(progressive method)의 영상으로 변환하는 기법으로 비월 주사 방식에 기인한 여러 가지 시각적인 문제점들을 제거하기 위해 널리 사용되고 있다(G.D. Hann et al., 1998; C. J. Kuo et al., 1996). 현재 가장 널리 사용되고 있는 방법으로는 윤곽선 기반 라인 평균(Edge-based Line Average:

ELA) 방법으로 가장 상관성이 높은 샘플들의 방향으로 보간이 이루어지는 기법으로 간단한 형태의 구조에 비해 뛰어난 성능을 나타내어 많이 사용되는 방법이다(Tao Chen et al., 2000). 그러나 작은 화소값의 변화에 민감하고 잘못된 윤곽선 정보를 이용하는 등의 단점이 존재한다. 최근 들어 이러한 단점을 제거하기 위하여 움직임 보상에 의한 방법들이 제안되었다. 이러한 방법들은 움직임 궤적(trajectory)을 따라서 보간이 이루어진다(F.M. Wang et al., 1990; K.J. Sugiyama et al., 1999; 정유영 외, 2000). 그러나 이 방법 또한 지나치게 움직임 추정 기반이라는 데 단점이 존재한다.

본 논문에서는 이러한 방법들 중 윤곽선 기반 라인 평균에 기반한 방법의 단점을 개선하여, 전체 영역에 대한 보간을 적용함으로써 발생하는 오차를 줄이기 위하여 관심 영역(Region of Interest: ROI)의 설정 후 설정된 영역에 대한 인터레이스 제거 기법을 적용함으로써 성능을 향상시켰다.



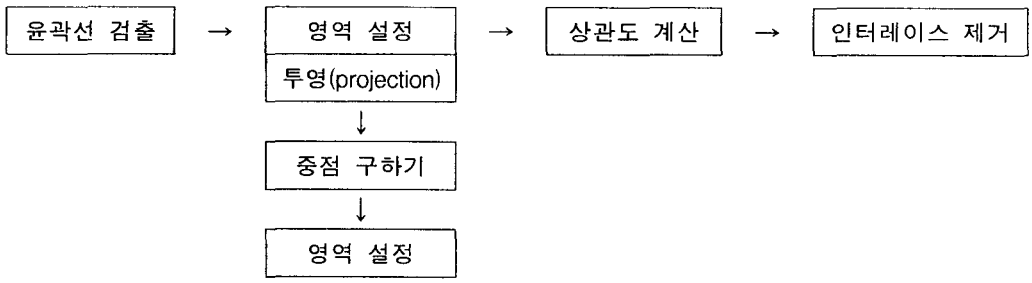
[그림 1] 대략적인 시스템 흐름도

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 인터레이스 제거 기법에 대하여 설명하고 3장에서는 인터레이스가 발생된 영상에 대하여 모델인식을 적용하는 방법을 설명하고 4장에서는 제안한 방법에 대한 실험 결과를 통해 성능 분석을 하며, 5장에서는 결론을 맺는다.

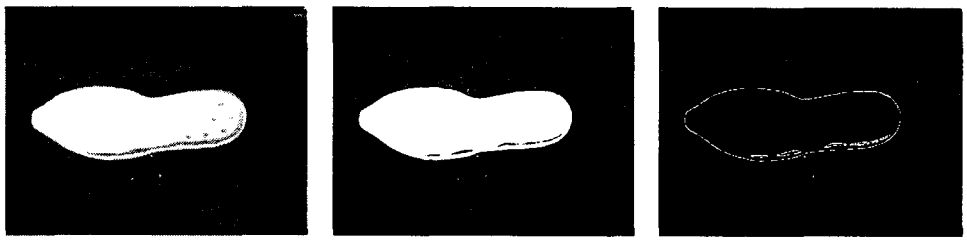
2. 인터레이스 제거

본 논문에서는 저가형 시스템의 구성을 위하여 NTSC 방식의 카메라를 이용하였다. 그러나 NTSC 방식 카메라의 경우 물체가 움직일 경우 고유의 잔상이 발생하는 단점이 존재한다. 이러한 잔상이 존재하는 영상으로는 정밀한 측정을 요구하는 분야에는 부적절한 면이 존재한다. 따라서 본 논문에서는 이러한 문제점을 개선하기 위하여 인터레이스 제거 기법을 제안하고 있다. 인터레이스 제거 효과는 비디오에서 캡쳐된 영상에서 인터레이스를 제거하여 부드럽게 표현하는 기능을 제공하므로 소프트웨어적 기법으로 하드웨어의 단점을 제거할 수 있다는 장점이 존재한다.

측정하고자 하는 물체가 컨베이어 벨트를 따라서 가로 방향(결이 생기는 방향)으로 흘러들어올 경우 카메라에서 측정하고자 하는 방향의 가로로 결이 발생하게 된다. 따라서 물체의 영역을 찾아 가로 방향에 해당하는 결을 제거하면 될 것이다. 본 논문에서 제안하는 인터레이스 제거 방법과 영역 설정을 위한 단계는 [그림 2]와 같다. 우선 반복 이진화 기법을 사용하여 이진화를 수행한 후, 윤곽선 검출을 위하여 라플라시안(Laplacian) 연산자를 적용한 후, 물체의 영역을 계산하였다. 이는 후에 상관도 계산에서 전체 영역에 대한 불필요한 계산을 줄여주는 효과와 물체 이외의 부분에 대한 불필요한 고려를 제거하므로 계산속도 및 정확도의 향상과 밀접한 관련이 있다.



[그림 2] 인터레이스 제거 방법



(a) 원영상 (b) 이진화 (c) 윤곽선 검출

[그림 3] 이진화 및 윤곽선 검출 결과

투영 단계의 경우 아래의 식 1, 2에 의해 이루어지게 된다.

$$x_{proj}(i) = \sum_{x=0}^{size\ of\ x} pixel(x,i) \quad (1)$$

$$y_{proj}(i) = \sum_{y=0}^{size\ of\ y} pixel(i,y) \quad (2)$$

투영 단계를 거친 후 물체의 중점은 아래의 식 3, 4에 의해 구해진다. $center_x$와 $center_y$는 각각 x와 y좌표에 대한 중점을 말한다.

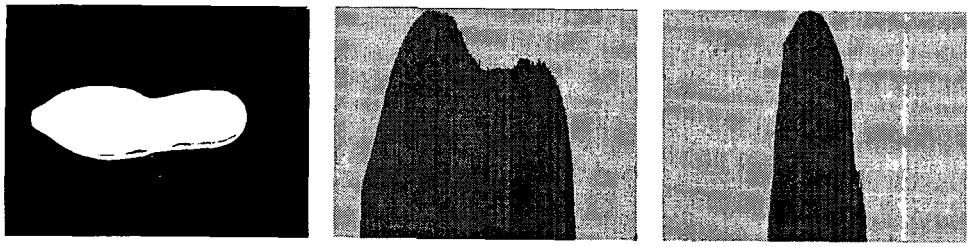
$$center_x = \frac{\sum_{x=0}^{size\ of\ x} x_{proj}(x)}{size\ of\ x} \quad (3)$$

$$center_y = \frac{\sum_{y=0}^{size\ of\ y} y_{proj}(y)}{size\ of\ y} \quad (4)$$

투영된 영상에 대하여 물체의 최종적인 영역 설정은 아래와 같이 투영된 영상의 x, y 각각의 좌표가 주어진 임계치보다 클 경우 픽셀 수를 하나씩 증가시키면서 이루어진다.

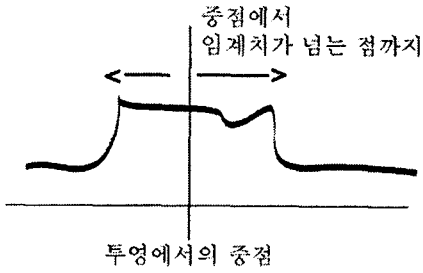
if ($x_{proj}(x) \geq threshold$) $x++$;
else break;
if ($x_{proj}(y) \geq threshold$) $y++$;
else break;

아래의 [그림 4]는 수직, 수평 투영 히스토그램을 보여주며, [그림 5]는 임계값(threshold)을 이용



(a) 원영상 (b) 수직 투영 히스토그램 (c) 수평 투영 히스토그램

[그림 4] 투영된 수직 · 수평 투영 히스토그램



[그림 5] 영역 설정

한 영역 설정을 위한 방법을 보여주고 있다.

영역 설정이 이루어진 후 인터레이스 제거를 위한 세 번째 단계로 상관도 계산 과정을 거치게 된다. 이 단계에서는 물체의 영역을 구한 후 홀수 라인과 짝수라인간의 자동 상관도 계수(auto correlation coefficients)를 구한 후 그 계수만큼 라인을 이동시키게 된다. 또한 본 과정에서는 홀수라인에서 얼마만큼의 offset을 주었을 때 짝수 라인과 일치하는가를 알아내야 한다. 이를 위한 최종 offset의 계산 방법은 식 8과 같다.

$$difference = \sum_{x=left}^{right} |pixel(x,y) - pixel(x+offset, y+1)| \quad (5)$$

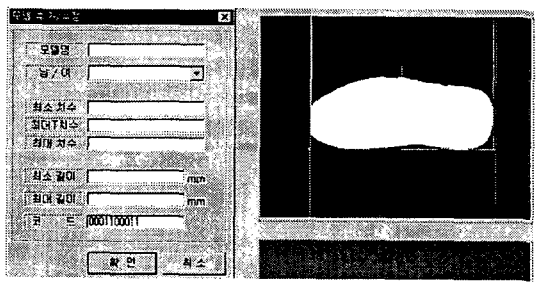
$$offset_{min} = \min\{ difference(|offset| < range) \} \quad (6)$$

$$offset_{sum} = \sum_{y=top}^{bottom} | offset_{min} | \quad (7)$$

$$offset_{final} = \frac{offset_{sum} \times 2}{bottom - top} \quad (8)$$

3. 모델 코드 생성

공장 현장에서 생산하는 제품에는 많은 수의 모델이 존재한다. 이 경우 여러 종류의 모델이 동시에 컨베이어 벨트를 통하여 측정 시스템으로 입력 될 경우, 각 모델마다 신발 측정에 대한 조건이 다르므로 해당 모델을 구별할 수 있어야 한다. 예를 들어 모델 별로 밑창(outsole)에 새겨진 구멍의 위치나 개수도 다를 것이며 모양의 다양성에 따른 문자 영역의 위치 또한 다를 것이다. 이러한 문제점을 해결하기 위하여 본 논문에서는 모델인식기법을 제안하고 있다. 즉 입력된 영상이 해당 모델임을 판별하는 방법으로 패턴인식 기법을 적용하



[그림 6] 모델 추가 작업 인터페이스

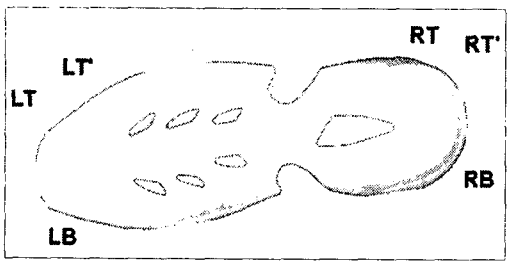
였다. 이를 통하여 각 영상의 특징을 추출 및 분류하여 모델 코드를 자동으로 생성해 낸다.

입력된 영상을 분류하는 방법에는 여러 가지가 존재하지만, 본 논문에서는 윤곽선 검출에 기반한 방법을 선택하였다. 전체 영상을 입력노드 또는 특징벡터로 사용하면 계산적 오버헤드가 크므로 학습 시간과 분류과정의 용이를 위해서 특징벡터의 수를 줄였다. 이를 위하여 블록 기반 코드열 생성 기법을 제안하고 있다.

먼저 물체의 영역을 설정하고, 일정 개수의 블록(본 논문의 경우 2*5의 입력 노드 적용)으로 나눈 다음 식 9에 의해 각 블록내의 화소수를 계산하여 입력 패턴을 생성한 후 모델인식을 수행한다. $block\ pixel_c$는 c번째 블록에 해당하는 화소수를 나타낸다.

$$block\,pixel_c = \sum_{y=y_c}^{block_y} \sum_{x=x_c}^{block_x} pixel(x,y) \tag{9}$$

또한 물체가 일정한 방향으로만 입력되는 것이 아니라 다양한 각도로 기울어진 형태로 입력되므로 기울어진 영상에 대한 보정을 통하여 모델 탐지가 이루어져야하겠다. 이를 위하여, 본 논문에서는 물체가 기울어졌을 때 평행한 직사각형으로 영역을 설정한 후 모델 입력 패턴을 생성시켰다.



[그림 7] 기울어진 영역 설정

[알고리즘 : 접선을 이용한 기울어진 영역 설정]

① 물체에 접하는 두 점을 기준으로 한 접선을 구한다. 그 접선을 따라 점이 이동하면서 수직의 직선이 물체에 접하는 두 점(LB, RB)를 구한다.

② 그 후 이 직선과 평행하게 이동시키면서 물체의 위쪽 가장자리와 일치하는 점 LT', RT'을 찾는다.

③ 다음의 단계를 통하여 LT, RT를 찾는다.

$$\begin{aligned} h &= LT'.y - LB.y \\ \theta &= \tan^{-1}\left(\frac{LB.x - RB.x}{LB.y - RB.y}\right) \\ w &= h \times \cos\theta \\ \Delta x &= w \times \sin\theta \\ \Delta y &= w \times \cos\theta \\ LT.x &= LB.x + \Delta x \\ LT.y &= LB.y - \Delta y \\ RT.x &= RB.x + \Delta x \\ RT.y &= RB.y - \Delta y \end{aligned} \tag{10}$$

④ 네 점(LT, RT, LB, RB)에 일차 회전 변환을 적용하여 평행 직사각형으로 변환한 후 각 픽셀을 계산하여 다시 일차변환을 적용 한다. (x_{source}와 y_{source}는 변환 전의 좌표를 나타내며, x_{dest}와 y_{dest}는 변환 후의 좌표를 나타낸다.)

$$x_{dest} = x_{source}\cos\theta - y_{source}\sin\theta \tag{11}$$

$$y_{dest} = x_{source}\sin\theta + y_{source}\cos\theta \tag{12}$$

⑤ 각 블록에서 윤곽선 검출 결과 픽셀수를 계산하여 코드를 생성한다.

$$average = \frac{\sum_{c=0}^{pattern\ size} block\ \pi xel_c}{pattern\ size} \quad (13)$$

$$\begin{cases} code_c = 1 & \text{if } (block\ \pi xel_c \geq average) \\ code_c = 0 & otherwise \end{cases} \quad (14)$$

⑥ 이미 학습된 코드와 비교하여 hamming distance를 측정하여 거리차가 최소인 것을 승자로 선택한다($node_{register}$와 $number_{pattern}$는 학습되어 있는 패턴에 대한 노드와 패턴의 수를 나타내며, $node_{input}$는 입력에 대해 할당된 노드를 나타낸다).

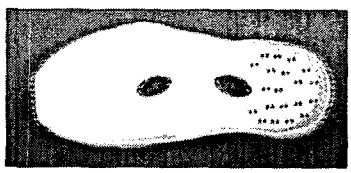
$$difference = \sum_{i=0}^{|node|} |node_{register}(number_{pattern}; i) - node_{input}(i)| \quad (15)$$

$$winner = \min\{difference\{number_{pattern} < \max_{model}\}\} \quad (16)$$

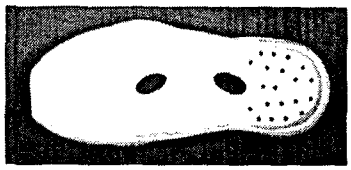
4. 실험 결과

본 논문에서 제안된 알고리즘의 성능을 분석하기 위하여 Intel Pentium-IV 2.3GHz CPU와 512MB RAM이 장착된 IBM 호환 PC상에서 Visual C++로 구현하였다. 실험에 사용된 데이터는 Samsung BW-2302 CCD 카메라를 사용하여 256 그레이 640×480 픽셀 크기를 가진 신발 밑창 영상 23개를 대상으로 실험하였다. [그림 8]은 본 논문에서 실험을 위하여 사용한 다양한 실험 영상의 샘플을 보여주고 있다.

본 논문의 실험은 크게 두 가지 결과 비교에 초점을 두고 이루어졌다. 첫째, 인터레이스 제거 기법을 적용하기 전, 후의 비교를 통하여 실제 영역 설정에 있어 발생하는 픽셀 오차를 구해보았으며, 둘째, 모델 분류를 위하여 제안한 기법을 적용하였을 경우 모델 분류 성공률을 계산하였다.

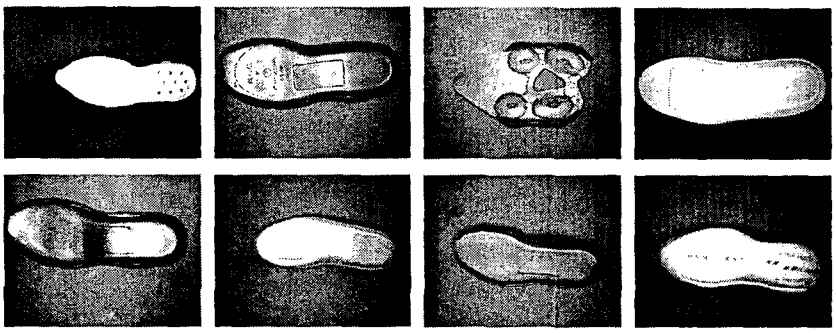


(a) 인터레이스 제거 전



(b) 인터레이스 제거 후

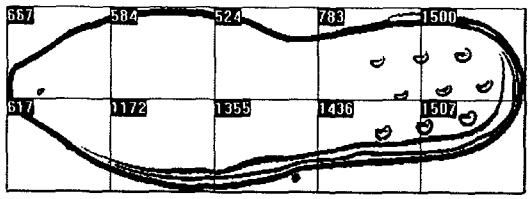
[그림 9] 인터레이스 제거 전후 비교



[그림 8] 실험 영상 샘플

본 논문에서 제안하는 관심영역에 기반한 인터레이스 제거 기법의 성능 테스트를 위하여 기존의 윤곽선 기반 라인 평균 방법과 단순한 관심영역의 설정의 경우를 비교 대상으로 실험하였다. <표 1>에서 보듯이 기존의 윤곽선 기반 라인 평균 방법의 경우 약 64% 정도의 인터레이스 제거 효과가 발생하였으며 픽셀 오차 또한 0.76 픽셀 정도로 나타났다. 그리고 단순한 관심영역의 설정에 기반한 방법의 경우도 윤곽선 기반 라인 평균 방법에 비해 85% 정도의 인터레이스 제거 효과를 보이며 다소 성능이 향상됨을 알 수 있으나, 본 논문에서 제안하고 있는 방법의 경우 96% 정도의 인터레이스 제거 효과를 보이며 평균 픽셀 오차 또한 0.04 픽셀로 윤곽선 기반 라인 평균 방법에 비해 약 1/20 정도의 절감 효과를 볼 수 있다.

인터레이스 제거된 영상에 대하여 모델코드를 인식하기 위하여 본 논문에서는 블록 기반 코드열 생성 기법을 제안하였다. 공장에서 새로운 모델이 생성될 때 마다 모델 데이터베이스에 새로운 모델을 입력하게 된다. 그러할 경우 컨베이어 벨트를 통하여 흘러들어오는 모델에 대하여 올바른 모델 분류를 하기 위해서는 정확한 모델 코드가 생성되어야 한다. 실험을 위하여 노란색, 파란색, 흰색, 검은색 밑창 23개의 영상에 대하여 실험을 하였다. 실험 결과 추출된 모델에 대한 평균 모델 분류 성공률은 표 2에서 나타난 바와 같이 밑창의 색깔에 큰 영향을 받고 있음을 알 수 있다. 특히, 검은색의 경우 배경 색과 유사한 색을 갖고 있는 특징에 기인하여 모델 분류에 영향을 받았음을 알 수 있었다.



[그림 10] 모델코드 생성 결과

<표 2> 모델 추출 실험 결과

	노란색 (8개)	파란색 (4개)	흰색 (7개)	검은색 (4개)
추출 개수	8개	4개	5개	2개

5. 결론

본 논문에서는 산업현장에서 저가형 시스템에 기반한 신발 밑창의 길이 측정을 위한 모델 인식 방법을 제안하였다. 실험의 전처리 단계로 윤곽선 검출과 상관도 검사에 기반하여 NTSC 카메라 고유의 인터레이스를 제거시키는 작업을 하였다. 또한, 모델인식을 수행 하여 사용자가 길이 측정을 위한 자동화된 모델선택을 가능하게 하였다. 그러

<표 1> 인터레이스 제거 실험 결과

	방법	인터레이스 제거율(%)	평균 오차 픽셀
1차 실험	영역설정하지 않았을 때	64%	0.76 pixel
2차 실험	영역설정 후	85%	0.15 pixel
3차 실험	기울기 보정 후 영역 설정 후	96%	0.04 pixel

나, 실험을 통하여 밑창 색깔에 따라 이진화 과정에서 최적의 임계치를 찾는데 성능의 차이가 발생함에 따라 인식률에 차이가 발생함을 알 수 있었다. 특히, 검은색 밑창의 경우 배경색과의 유사성에 의해 낮은 인식률을 보였다. 본 논문에서 제안한 알고리즘의 장점으로는 다양한 환경, 물체의 색깔, 밝기가 달라져도 공통적으로 이용할 수 있는 장점을 지닌다. 이는 향후 인간의 감각 및 감성적인 판단을 대신할 수 있는 각종 응용공학 분야에 많은 응용이 기대된다. 또한 공장 자동화에 따른 인력감소와 인건비 절감의 효과로 산업체의 가격 경쟁력을 향상시킬 수 있을 것으로 기대된다.

향후 연구과제로는 모델 색깔에 균일하게 적용될 수 있는 알고리즘의 개발과 모델 및 치수에 따라 변화하는 밑창에 양각된 문자 영역의 효율적 탐지 및 인식 기법에 관한 연구가 이루어져야겠다.

참고문헌

[1] 김강석, 김철기, 차의영, "동영상에서 모델인식을 이용한 문자 영역 추출," *멀티미디어합동학술대회논문집*(2000).

[2] G.D. Haan, and E.B. Bellers, "Deinterlacing - an Overview," *Proc. IEEE*, vol. 86, no.9 (1998), 1839-1857.

[3] C. J. Kuo, C. Liao, and C.C. Lin, "Adaptive Interpolation Technique for Scanning Rate Conversion," *IEEE Trans. Circuits Syst. Video Technol*, vol. 6, no. 3(1996), 317-321.

[4] F.M. Wang, and D. Anastassiou, and A.N. Netravali, "Time-recursive deinterlacing for IDTV and pyramid coding," *Signal Process. : Image Commun. 2*(1990), 365-374.

[5] K.J. Sugiyama, and H.Y. Nakamura, "A Method of De-interlacing with Motion Compensated Interpolation," *IEEE Trans. Consumer Electronics*, vol.45, no.3(1999), 611-616.

[6] 정유영, 고성제, "양방향 움직임을 사용한 효과적인 디인터레이싱 기법," *제13회 신호처리합동학술대회논문집*, 제13권, 1호(2000), 873-876.

[7] Tao Chen, Hong Ren Wu, Zheng Hua Yu, "An Efficient Edge Line Average Interpolation Algorithm for De-interlacing, In Visual Communications and Image Processing 2000", *Proceedings of SPIE*, vol.4067(2000), 1551-1558.

Abstract

De-interlacing and Block Code Generation For Outsole Model Recognition In Moving Picture

Kim, Cheol-Ki*

This paper presents a method that automatically recognizes products into model type, which it flows with the conveyor belt. The specific interlaced image are occurred by moving image when we use the NTSC based camera. It is impossible to process interlaced images, so a suitable post-processing is required. For the purpose of this processing, after it remove interlaced images using de-interlacing method, it leads rectangle region of object by thresholding. And then, after rectangle region is separated into several blocks through edge detection, we calculate pixel numbers per each block, re-classify using its average, and classify products into model type. Through experiments, we know that the proposed method represent high classification ratio.

Key words : De-interlacing, Block code generation, Model recognition, Factory automation

* Department of Design, Pusan National University