

OpenMP를 이용한 차량 헤드라이트 얼라인먼트 보정 및 분류 방법

(Vehicle Headlight Alignment Calibration and Classification Using
OpenMP)

문창배¹⁾, 김근홍²⁾, 김병만³⁾, 오득환^{4)*}

(Moon Chang-Bae, Kim Kun-Hong, Kim Byeong-Man, and Oh Dukhwan)

요약 본 논문에서는 CPU기반 병렬처리인 OpenMP를 적용하여 헤드라이트의 차종을 판별하기 위한 처리속도를 개선하고, 생산부품의 얼라인먼트를 보정 후 특징을 추출하여 차종을 판단하는 방법을 제안하였다. 제안방법의 세부성능을 분석하기 위해 그레이영상을 이용한 방법 및 라인 검출을 이용한 방법과의 판별성능과 처리속도를 비교·분석하였다. 분석결과, 판별성능의 경우 제안방법과 라인 검출을 이용한 방법이 좋은 성능을 보였지만 처리속도 측면에서 라인검출 방법보다 제안 방법이 더 좋은 성능을 보였다. 처리속도의 경우 그레이 영상을 이용한 방법이 가장 좋은 성능을 보였으나 판별성능 측면에서 제안방법이 그레이 영상을 이용한 방법보다 좋다고 할 수 있다.

핵심주제어 : OpenMP, 영상처리, 우도, 병렬처리, 차량 헤드라이트, 얼라인먼트

Abstract In This Paper, the Classification Speed of Vehicle Headlight Modules is Improved by a CPU-based Parallel Processing Using OpenMP. Also, a Classification Method of Headlight Modules which Extracts their Features after Revising their Alignment is Proposed. To Analyze the Performance of the Proposed Method, the Discrimination Accuracy and the Processing Speed were Compared with the Method Using Gray Image and the Method Using Line Detection. As the Results of the Analysis, in the Discrimination Accuracy, the Proposed Method and the Line Detection Method Showed good Performance, but the Proposed Method Showed Better Performance than the Line Detection Method by the Processing Speed. Also, the Gray-based Method was the Best in Processing Speed, but the Proposed Method is Better than the Gray-based Method in the Discrimination Accuracy.

Key Words : OpenMP, Image Processing, Likelihood, Parallel Processing, Vehicle Headlight, Alignment

* Corresponding Author : dhoh@kumoh.ac.kr

† 이 논문은 2016년 금오공과대학교 연구비 지원에 의해 연구되었음.

Manuscript received Feb 27, 2017 / revised Apr. 17, 2017 /
accepted Apr. 25, 2017

1) 금오공과대학교 ICT융합특성화연구센터, 제1저자
2) 금오공과대학교 컴퓨터IT학과, 프론틱 코퍼레이션, 제2저자
3) 금오공과대학교 컴퓨터소프트웨어공학과, 제3저자
4) 금오공과대학교 컴퓨터소프트웨어공학과, 교신저자

1. 서론

산업이 발전함에 따라 생산시스템 역시 발전하는 추세인데, 생산시스템의 핵심 기술에는 머신비전(Machine Vision)기술, 산업용 로봇제어기술, 산업용 PC 기술, PLC(Programmable Logic Controller)기술 그리고 센서(Sensor)기술 등으로 분류할 수 있다. 그중 머신비전 기술의 경우 생산시스템의 생산 비용을 절감하거나 생산성이 향상시킬 수 있는 중요한 기술이라 할 수 있다[1].

완제품에 불량 부품을 사용하면 부품의 불량이나 완제품의 불량으로 판정됨으로 부품조립과정은 완제품 불량률을 낮추는 중요한 요인이라 할 수 있다. 부품조립과정은 입력된 부품을 작업자가 판단하여 해당 부품의 부속부품들을 조립하는 과정으로, 능숙한 작업자가 제조할 경우 불량 발생률은 줄어들지만 작업자의 능력 및 작업환경 등에 따라 불량률에 영향을 줄 수 있다. 따라서, 조립불량으로 인한 완제품 불량률을 줄이기 위해 자동화는 매우 중요한 요인이라고 할 수 있다[2].

머신비전 기술은 부품생산 전/후를 기준으로 두 가지로 분류할 수 있다. 그 첫 번째가 부품생산 후 불량 유무를 판별하는 기술이고, 두 번째가 부품생산 전 작업자의 작업효율을 향상시키기 위한 목적으로 생산부품의 부속부품들을 분류하는 기술이다. 두 기술에서 필수적으로 사용하는 기술 중 한 가지는 얼라인먼트 보정기술이라 할 수 있고, 이 기술은 보편적으로 생산품에 각인된 마크 또는 생산품에 존재하는 유니크한 패턴을 이용하여 생산품의 각도와 위치를 파악하는 기술이지만 차량 부품과 같이 생산부품의 구조적 특성으로 유니크 패턴과 마크의 각인이 불가능한 경우도 발생한다.

차량 부품 분류 방법에서 중요하게 여기는 것은 정확도 및 처리속도라 할 수 있다. 정확도는 입력된 부품의 얼라인먼트를 보정한 후 특징을 추출하여 차종판별 시 그 정확도를 의미하는 것이고, 처리속도는 차종판별까지의 소요시간이다. 이러한 맥락으로 본 논문에서는 부품의 얼라인먼트를 보정 후 부품의 사이즈를 측정하여 차종을 분류하는 방법을 제안하고, CPU기반 병렬처리를 적용하여 차종판별 처리속도를 개선하고자 한다.

기존 얼라인먼트 보정기술에는 라인검출 방법 [3], 적분 히스토그램(Integral Histogram) 방법 [4, 5] 그리고 프로젝션(Projection)기반 적분히스토그램 방법[1] 등이 존재한다. 라인검출 방법은 부품의 특성인 라인을 검출하여 부품의 얼라인먼트를 보정하는 방법으로 영상의 부품에 직선이 존재하는 경우 부품의 라인을 이용하여 빠르게 얼라인먼트를 보정할 수 있지만, 차량 헤드라이트의 경우 곡면으로 구성되어 얼라인먼트 오차가 발생할 수 있고, 처리 시간비용이 높아질 수 있다. 적분 히스토그램 방법은 적분 히스토그램기반 템플릿매칭을 적용하여 원과 십자 모양 등의 마크를 검출하는 방법으로 부품에 마크가 없다면 좋은 판별성능은 기대할 수 없다. 적분 히스토그램 방법을 개선한 방법으로 프로젝션(Projection)기반 적분히스토그램 방법이 존재하지만 차량 헤드라이트 부품의 특성상 유니크한 패턴 사용이 불가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 적용하는 기술의 전체 시스템 구조를 설명하고, 3장에서는 본 논문에서 제안한 차량 부품 분류 및 얼라인먼트 보정 방법, 4장에서는 본 논문에서 제안한 방법의 정확도와 처리속도에 대한 실험결과, 마지막으로 5장에서는 본 논문을 결론짓도록 한다.

2. 시스템 구성 및 실험

전통적 차량용 헤드라이트 제조 작업 환경은 Fig. 1 a와 같은 구조로 하나의 컨베이어시스템에 여러 명의 작업자가 2개 이상의 헤드라이트의 부품을 조립하는 환경이고, 작업자가 헤드라이트의 차종을 판단하여 조립할 부품을 선택한다. 그러나 일부 부품의 경우 Fig. 1 b, c와 같이 헤드라이트의 구조가 유사하고, 이로 인하여 작업자의 작업 오류가 발생할 수 있다.

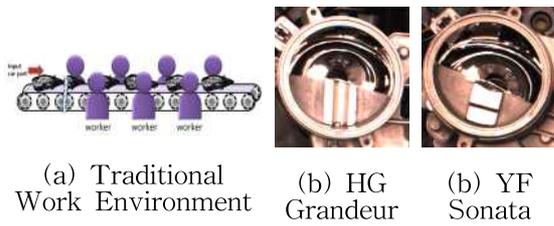


Fig. 1 Traditional Work Environment and Headlight Images

본 논문에서 제안한 방법은 이종부품 분류기의 검사기에 적용되고(Fig. 2 참고), 최초 차량용 헤드라이트가 이종부품 분류기에 입력되면 차량용 헤드라이트의 특징을 추출하여 해당 차종을 판단하여 헤드라이트를 분류한다.

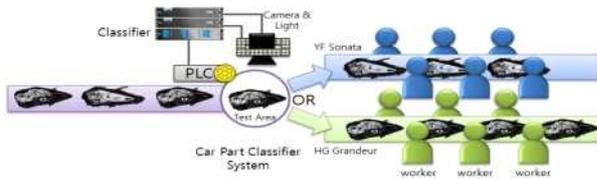


Fig. 2 Vehicle Parts Classification System Structure

이종부품 분류기 시스템은 검사기 모듈, PLC 모듈, 검사조명 모듈 그리고 카메라 모듈로 구성되고, 각 모듈의 기능은 다음과 같다.

- 검사조명 모듈 : 광량을 조절하는 모듈로 차량용 헤드라이트 부품이 이종부품 분류기의 검사영역에 입력되면 촬영을 위한 광량을 제공, 검사기 모듈에서 본 모듈을 제어
- 카메라 모듈 : 차량용 헤드라이트 부품이 이종부품 분류기의 검사영역에 입력되면 부품을 촬영하여 검사기 모듈로 영상을 송신
- 검사기 모듈 : 차종을 판별하기 위한 차량용 헤드라이트 영상을 카메라 모듈로부터 수신, 수신한 영상을 이용하여 차종을 판별, 판별 후 PLC로 차종에 대한 정보를 송신
- PLC 모듈 : 검사기 모듈과 검사조명에서 영상 수신 및 영상촬영을 위한 정보를 송신하고, 검사기 모듈에서 차종에 대한 정보를 수신 후 모터를 제어하여 차량용 헤드라이트를 분류

3. OpenMP를 이용한 차량 헤드라이트 얼라인먼트 보정 및 차종 분류 방법

본 논문에서 제안한 분류기 모듈의 큰 구조는 총 3단계로 구성된다. 그 첫 번째 단계는 얼라인먼트 보정용 장축 좌표검색이고, 두 번째 단계는 영상 이동 및 회전이며 마지막이 차종 판별 단계이다. 얼라인먼트 보정용 장축 좌표검색단계에서는 차량용 헤드라이트(객체)의 장축 좌표를 검색하고, 회전 및 이동좌표계산 & 이동 및 회전 단계에서는 장축좌표를 이용하여 회전 각도와 영상 이동 좌표를 계산하여 객체를 영상의 중심으로 이동 및 회전 시키는 단계이다. 마지막으로 차종 판별단계는 객체의 특징들을 추출하여 차종을 판별하는 단계이다.

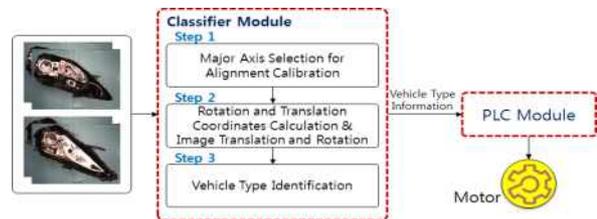


Fig. 3 Module Structure of Classifier

3.1 얼라인먼트 보정용 장축 좌표검색 단계 (단계 1)

얼라인먼트 보정용 장축 좌표검색 단계의 프로세싱과정은 Fig. 4와 같이 총 3단계로 구성된다. 그 첫 번째 과정은 파티클 필터(Particle Filter) [6, 9]에서 사용하는 우도(Likelihood)기반 색상모델 변환이고, 두 번째 과정은 객체 획득 및 객체 홀 제거 그리고 마지막이 객체 장축 좌표 검색으로 최종적으로 영상 이동 좌표와 객체의 회전 각



Fig. 4 Selection Process of Major Axis for Alignment Calibration

도를 계산을 위한 장축 좌표를 검색한다. Fig. 4의 샘플링 좌표는 우도를 적용하기 위한 오퍼레이터의 입력좌표이고, 본 논문에서는 배경을 선택한다.

검사 조명을 이용하여 차량용 헤드라이트를 촬영하는 경우 헤드라이트의 구조로 인한 음영이 존재하고(Fig. 5 a 참고), 그레이영상[8] 기반에서 이진화를 진행 할 경우 히스토그램에서 색상의 분포가 한쪽으로 편중되는 증상(Fig. 5 d 참고)이 발생한다. 이로 인하여 임계값으로 설정할 구간이 협소(Fig. 5 d ① 참고)하다. 또한 빛의 영향력을 최소화할 목적으로 HSV모델을 사용하는 경우 3개의 임계값을 설정할 필요가 있다.

본 논문에서는 빛의 영향력 최소화를 목적으로 위해 1.1 단계에서 RGB모델을 HSV모델로 변경 후 우도와 정규화를 적용함으로 3개의 임계값 대신에 하나의 임계값만을 사용할 수 있고, 그레이영상 기반 임계값 설정구간 보다 본 논문의 방법이 넓은 설정구간을 사용할 수 있다.

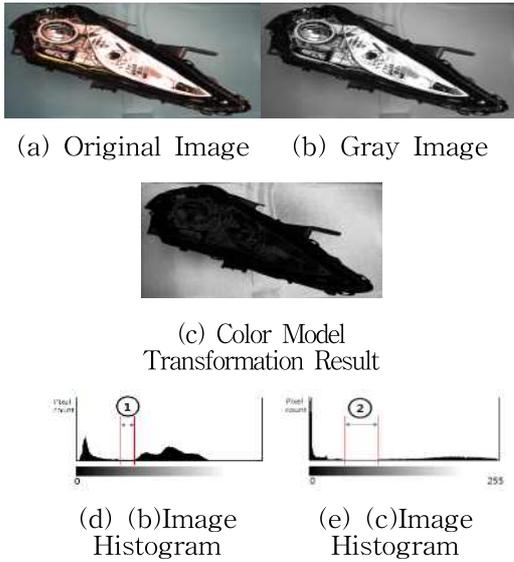


Fig. 5 Comparison of Likelihood-Base Color Model Image with Gray Color Model Image

3.1.1 단계 1.1 : 우도(Likelihood) 기반 색상 모델 변환

본 단계에서 적용한 우도는 식 (1)과 같고, 샘플링 좌표의 HSV값과 입력 영상의 HSV값을 이용하여 우도 값을 계산한다. 우도 값 계산 후 식 (2)를 적용하여 정규화를 적용한다. 본 단계의 결과는 Fig. 5 c와 같고, 임계값 설정구간은 Fig. 5 e의 ②와 같다. 또한 연산시간 단축을 목적으로 OpenMP를 적용한다.

플링 좌표의 HSV값과 입력 영상의 HSV값을 이용하여 우도 값을 계산한다. 우도 값 계산 후 식 (2)를 적용하여 정규화를 적용한다. 본 단계의 결과는 Fig. 5 c와 같고, 임계값 설정구간은 Fig. 5 e의 ②와 같다. 또한 연산시간 단축을 목적으로 OpenMP를 적용한다.

$$L(d) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{d^2}{2\sigma^2}\right), \quad (1)$$

$$d = \sqrt{(h_s - h_i)^2 + (s_s - s_i)^2 + (v_s - v_i)^2},$$

$$L(y, x) = \frac{L(y, x) - MIN}{MAX - MIN}, \quad (2)$$

$$MIN = \min [L(1, 1) \dots L(h, w)],$$

$$MAX = \max [L(1, 1) \dots L(h, w)].$$

여기서, $L(d)$ 는 픽셀의 우도결과 값을 의미하고, h_s 는 샘플링 좌표(s)의 휴(h)값, h_i 는 이미지 좌표(i)의 휴(h)값을, $L(y, x)$ 는 영상 좌표 (y, x) 에서의 정규화 값을, (h, w) 는 입력영상의 높이(h)와 폭(w)을 의미한다.

OpenMP 적용방법은 Fig. 6과 같이 두 개의 함수를 사용한다. 첫 번째 함수는 calc_likelihood로 샘플링 값과 이미지 픽셀 값의 우도를 계산한

```

Function calc_Likelihood
Input: 샘플링 값 h_s, s_s, v_s, 입력 이미지 값 h_i, s_i, v_i
Output: 식 (1)의 Likelihood 결과 L(d) 값
Method
begin
    식 (1):
end

Function doHSVLikelihood
Input: RGB기반 영상
Output: Likelihood기반 영상
Method
begin
    RGBtoHSV():
    refh ← H[샘플링좌표][샘플링좌표]: refs ← S[샘플링좌표][샘플링좌표]: refv ← V[샘플링좌표]
    [샘플링좌표]: max_likelihood ← 0.0: min_likelihood ← 10.0

    #pragma omp parallel for shared(max_likelihood, min_likelihood) private(i, j)
    for j ← 0 until h do
        #pragma omp parallel for shared(max_likelihood, min_likelihood) private(i)
        for i ← 0 until w do
            likelihood[i][j] ← calc_likelihood(refh, refs, refv, H[i][i], S[i][i], V[i][i])
            if likelihood[i][j] > max_likelihood then
                max_likelihood ← likelihood[i][j]
            end
            if likelihood[i][j] < min_likelihood then
                min_likelihood ← likelihood[i][j]
            end
        end
    end

    #pragma omp parallel for shared(max_likelihood, min_likelihood) private(i, j)
    for j ← 0 until h do
        #pragma omp parallel for shared(max_likelihood, min_likelihood) private(i)
        for i ← 0 until w do
            식 (2):
        end
    end
end
    
```

Fig. 6 OpenMP Code

다. 두 번째 함수의 경우 doHSVLikelihood로 전 체이미지에 대한 색상변환과 정규화 과정을 진행 하고, calc_likelihood함수와 정규화에 대하여 병 렬처리를 적용한다(7번째 라인 #pragma omp parallel 참고). doHSVLikelihood함수에서 RGBtoHSV함 수의 경우 RGB색상체계를 HSV색상체계로 변환 하는 함수이고, RGBtoHSV함수역시 doHSVLikelihood 함수와 유사하게 OpenMP를 적용한다.

3.1.2 단계 1.2 : 객체 획득 및 객체 홀 제거

객체 획득을 목적으로 이진화 적용 후 Fig. 7 a와 같이 배경에 존재하는 노이즈를 제거하기 위 해 플러드 필(Flood Fill)[7]을 적용하여 Fig. 7 b 와 같이 객체를 획득한다. 플러드 필을 적용하기 위해서는 객체의 좌표가 필요하고, 객체의 좌표 는 오퍼레이터가 입력한다. 플러드 필을 이용하여 획득한 객체의 경우 객체 내부에 미세한 홀이 존재하여 객체의 특징 추출에 영향을 줄 수 있 다. 이를 방지하기 위해 본 과정에서는 배경을 획득하는 방법으로 객체의 홀을 제거하고, 처리 속도개선을 목적으로 이진화와 배경획득 방법에서 OpenMP를 적용한다.

객체에 존재하는 홀을 제거는 Fig. 7 d와 같은 과정으로 처리과정은 다음과 같고, 객체의 홀을 제거한 결과는 Fig. 7 c와 같다.

- ① : 플러드 필을 이용한 객체 획득
- ② : 이미지 테두리의 값을 0과 1이 아닌 임의 값으로 변경
- ③, ④ : 상단과 하단 병렬처리로 진행, 현재 픽셀 값이 1이고 이전 픽셀(h-1 또는 h+1)의 값이 테두리 값인 경우 현재 픽셀 값을 테두리 값으로 변경, 현재 픽셀 값이 0인 경우 픽셀 값은 변경 없음(Fig. 6과 유사한 방법으로 OpenMP를 적용)
- ⑤ : 현재 픽셀의 값이 0또는 1인 경우 픽셀 값을 1로 변경
- ⑥ : 최종결과 획득

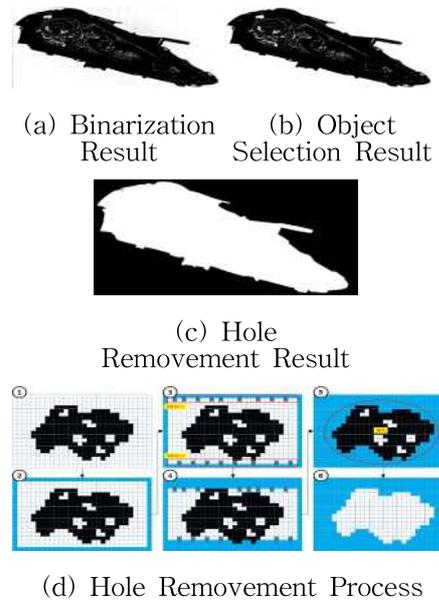


Fig. 7 Hole Removal Process and Result

3.1.3 단계 1.3 : 객체 장축 좌표 검색

객체의 장축좌표를 검색은 Fig. 8 a와 같이 객체의 상단좌표와 하단의 좌표 그리고 좌측좌표와 우측좌표를 이용하여 객체의 중심좌표를 계산 후 Fig. 8 b와 같이 객체의 중심으로부터 객체의 좌우측 픽셀 값이 1인 좌표까지의 거리를 계산하고, 계산한 좌우측 좌표까지의 거리 중 거리 값이 가장 큰 좌표를 구하는 방법으로 객체의 좌우측 장축좌표를 구한다. 식 (3, 4)는 이를 수식화한 것이다.

$$index^L = \operatorname{argmax} [L_1, L_2, \dots, L_{n-1}, L_n], \quad (3)$$

$$L_n = \sqrt{(x_c - x_n)^2 + (y_c - y_n)^2},$$

$$index^R = \operatorname{argmax} [R_1, R_2, \dots, R_{m-1}, R_m], \quad (4)$$

$$R_m = \sqrt{(x_c - x_m)^2 + (y_c - y_m)^2}.$$

여기서, $index^L$ 과 $index^R$ 는 객체 좌(L)/우측(R) 좌표를 의미하고, x_c 와 y_c 는 객체의 중심(c)의 좌표(x, y), x_n, y_n 와 x_m, y_m 는 객체 좌측(n) 좌표와 우측(m) 좌표 중 픽셀값이 1인 좌표를 의미한다.

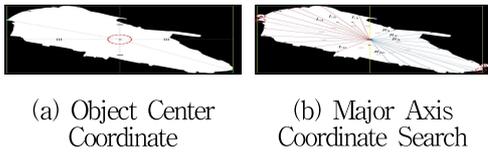


Fig. 8 Object Center Coordinate and Major Axis Coordinate Search

3.2 영상 이동 및 회전 단계 (단계 2)

영상 이동 및 회전 방법은 Fig. 9과 같이 총 3 단계로 진행되고, 그 첫 번째 단계는 회전각도 및 이동 좌표 계산이고, 두 번째 단계는 영상 이동변환 그리고 마지막이 영상 회전변환 단계이다. 본 단계에서는 처리속도 개선을 목적으로 영상 이동변환 단계와 영상 회전변환 단계에서 OpenMP를 이용한 병렬처리를 적용한다.

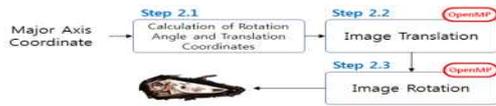


Fig. 9 Image Translation and Rotation Step Process

3.2.1 단계 2.1 : 회전각도 및 이동좌표 계산

본 단계에서 이동좌표는 장축의 중심으로부터 영상중심 좌표까지 이동할 거리를 의미하고, 회전각도는 장축이 회전할 각도를 의미한다. 이동좌표계산은 Fig. 10 a와 같이 좌측 장축 좌표와 우측 장축 좌표의 중심 좌표를 구하고, 이 중심좌표와 영상 중심좌표의 차를 구함으로써 이동 좌표를 계산한다. 회전각도는 Fig. 10 b와 같이 좌우측 장축 좌표((x_1, y_1), (x_2, y_2))를 이용하여 영상이 회전할 각도(θ)를 식 $\theta = atan((y_2 - y_1)/(x_2 - x_1))$ 를 이용하여 회전각도를 계산한다.

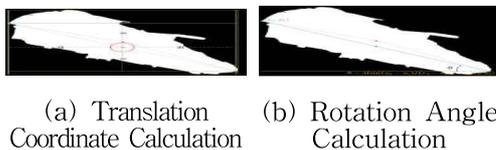
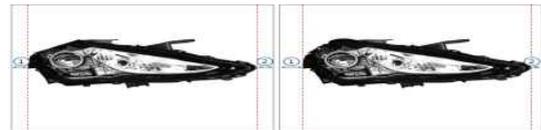


Fig. 10 Calculation of Translation Coordinate and Rotation Angle

3.2.2 단계 2.2, 2.3 : 영상 이동변환 및 영상 회전변환

2.1단계 결과의 응용인 장축 길이를 이용하여 헤드라이트의 차종을 판별하는 경우 본 논문에서 고려한 차종은 분류가 가능하다. 하지만 하드웨어에서 헤드라이트의 입력이 들어지는 경우와 새로운 차종의 헤드라이트가 입력되는 경우 이를 구분하기 위한 기준이 필요할 수 있다. 즉, 본 논문에서는 입력영상의 면적대비 등의 특징을 사용하기 위해 이동변환 및 회전변환을 적용한다.

본 단계에서는 회전각도 및 이동좌표 계산 단계에서 획득한 회전 각도와 이동좌표를 이용하여 영상 이동 및 회전을 적용한다. 영상 회전변환을 우선 적용하는 경우 영상 회전으로 인하여 2.1단계에서 계산한 이동좌표가 영향을 받지만 영상 이동변환을 우선 적용하는 경우 영상 이동으로 인한 2.1단계에서 계산한 회전각도에는 영향이 없으므로 영상 이동변환을 우선 적용한다. 즉, 영상 이동변환 후 회전변환은 Fig. 11 a ①, ②와같이 좌우측 간격은 동일하지만, 영상 회전변환 후 이동변환은 Fig. 11 b ①, ②와같이 좌우측 간격은 서로 다르므로 Fig. 11 c와 같이 이동변환 적용 후 Fig. 11 d와 같이 회전변환을 적용한다.



(a) Translation/Rotation Result (b) Translation/Rotation Result



(c) Translation Result (d) Rotation Result

Fig. 11 Image Translation and Rotation

3.3 차종 판별 단계 (단계 3)

차종을 판별을 목적으로 2.2단계의 결과를 이용하여 헤드라이트의 폭, 높이, 면적, 회전변환 영상 면적 대비 객체 면적, 회전변환 이전 영상

면적 대비 객체 면적 등의 특징을 추출한다. 본 논문에서는 두 개의 차종인 HG그랜저와 YF소나타만을 고려하였고, 이 경우에는 헤드라이트 폭만을 사용하여 차종을 분류할 수 있다. 하지만 본 논문에서 고려한 차종이 아닌 이외의 차종을 분류하는 경우 다른 특징들을 이용하여 분류할 수도 있기 때문에 높이, 면적 등의 다른 특징들도 추출하였다. 각 특징은 다음과 같다.

- 헤드라이트의 폭 : 2단계에서 획득한 영상에서 헤드라이트의 폭, Fig. 12 a의 ①을 의미함
- 헤드라이트의 높이 : 2단계에서 획득한 영상에서 헤드라이트의 높이, Fig. 12 a의 ②를 의미함
- 헤드라이트의 면적 : 2단계에서 획득한 영상에서 헤드라이트의 면적, Fig. 12 a의 $h^r \times w^r$ 를 의미함
- 회전변환 영상 면적 대비 객체 면적 비율: 회전변환을 적용한 영상에서 객체 면적비율, Fig. 12 b ①영역의 픽셀 사이즈/ $(h^r \times w^r)$ 를 의미함
- 회전변환 이전 영상 면적 대비 객체 면적 비율: 회전변환 적용 이전 영상에서 객체 면적비율, Fig. 12 c ①영역의 픽셀 사이즈/ $(h^o \times w^o)$ 를 의미함

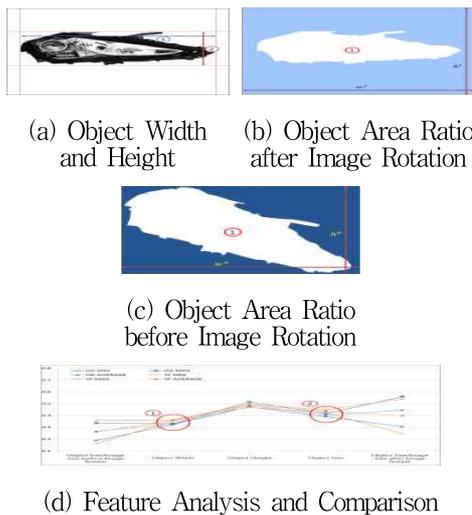


Fig. 12 Feature Extraction and Analysis

HG그랜저와 YF소나타를 분류하기 위해 각 특

징을 분석한 결과, Fig. 12 d와 같이 회전변환 이전 영상 면적대비 객체 면적, 객체 높이 그리고 회전변환 영상 면적대비 객체 면적의 경우 두 차종 분류 기준인 임계값을 적용하기가 힘들지만 객체 폭, 객체 면적의 경우 ①, ②와 같이 분류가 가능한 것을 알 수 있고, 특히 객체 폭의 경우 객체 면적보다 평균 및 MIN·MAX 값의 편차가 큰 것을 알 수 있다. 이 분석을 기반으로 본 논문에서는 두 개의 차종을 분류 한다.

4. 실험 및 결과

실험을 목적으로 본 논문에서는 Fig. 13과 같이 차량용 헤드라이트 촬영 환경을 구축하였다. 촬영 시 사용한 조명은 LED 조명으로 Fig. 13 a와 같이 중앙에 카메라를 장착 할 수 있는 판 조명을 사용하였고, Fig. 13 b 상단에 위치한 카메라와 렌즈를 사용하여 부품을 촬영하였다. 카메라 제원과 렌즈 제원은 Table 1과 같고, Intel(R) Core(TM) I7-4790 3.6GHz CPU, 8GB RAM 그리고 Win7 64bit 환경에서 영상처리 성능을 측정하였다. 영상촬영 시 2개 차종 각 1개의 시료를 사용하였고, 차량용 헤드라이트를 다시 마운트 시키는 방법으로 서로 다른 200장의 이미지를 확보하였다.

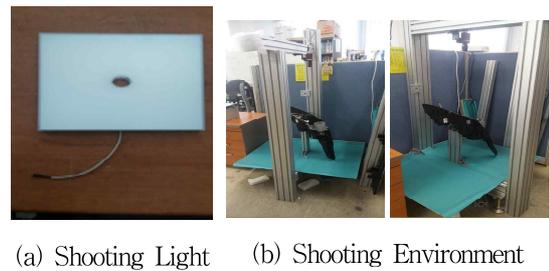


Fig. 13 Headlight Shooting Environment

Table 1 Camera and Lens Specifications

Camera	spec.	Lens	spec.
Resolution	1624 (H) × 1234 (V)	Focal length	8mm
Sensor	Sony ICX274	Image size	2/3"
Sensor Type	C C D Progressive	Iris range (F-stop)	F1.4 ~ Close
Sensor Size	Type 1/1.8	Focusing range	0.12 ~
Cell Size	4.4um × 4.4um	Contron	Iris (Manual), Focus (Manual)
Lens mount	C-Mount	Shooting range at M.O.D	117.3 (H) X 88 (V)
Frame rate	30.0 fps	Resolution	1201p/mm (Center), 1001p/mm (Corner)
ADC	12 Bit	Distortion	-0.6%
Image buffer	32 MByte	Filter thread (mm)	M27 X P0.5

본 논문의 제안방법을 이용하여 YF소나타와 HG그랜저를 분류한 경우 100% 판별율을 보였고, 세부적인 성능 비교를 목적으로 1.1단계와 1.3단계의 알고리즘 성능을 다른 방법과 비교하였다. 세부적인 성능 비교 대상과 방법은 제안방법의 우도기반 색상모델 변환을 이용한 객체검출 방법과 그레이영상을 이용한 객체검출 방법의 판별성능 및 처리속도를 비교하였고, 제안방법의 객체 장축 좌표를 이용한 회전각도 및 이동좌표 계산 방법과 라인[8] 검출을 이용한 회전각도 및 이동좌표 계산 방법의 판별성능 및 처리속도를 비교하였다. 라인검출을 이용한 방법은 Hough 변환을 이용한 방법은 소벨 연산과 Hough 변환을 이용하여 라인을 검출하고, 검출된 라인을 이용하는 방법이다. 또한 얼라인먼트 보정성능의 경우 미세한 각도를 측정하기 위한 기구를 별도로 구축하여 보정성능을 검증하여야 하지만 본 논문의 최종 목적은 헤드라이트의 차종분류이기 때문에 차종판별로 대처하였다.

제안방법과 그레이영상을 이용한 방법 그리고 라인 검출을 이용한 방법 판별 성능을 비교한 결과는 Table 2와 같이 제안방법과 라인 검출을 이용한 방법의 경우 판별율이 100%이고, 그레이영상을 이용한 방법의 경우 판별율이 99%이다.

그레이영상을 이용한 방법의 기타 1%가 발생하는 원인은 단계 1.2의 플러드 필에서 사용자 좌표 입력의 오류이다.

Table 2 Proposed Method vs Gray-Based Method vs Line Detection Method

	Proposed Method	Gray-Based Method	Line Detection Method
True Discrimination Rate	100 % (200/200)	99 % (198/200)	100 % (200/200)
False Discrimination Rate	- (0/200)	- (0/200)	- (0/200)
Etc	- (0/200)	1 % (2/200)	- (0/200)

플러드 필에서 오퍼레이터 좌표입력의 오류가 발생한 원인을 분석한 결과, 제안한 방법의 경우 Fig. 14와 같이 그레이영상을 이용한 방법보다 차량용 헤드라이트 내부의 홀이 작은 것을 알 수 있고, 차량 내부의 광 반사판이 존재함으로 그레이영상을 이용한 방법을 적용한 객체에 홀이 많은 것을 알 수 있다. 또한 그레이영상을 이용한 방법의 경우 홀 내부에도 하나의 객체로 분류할 수 있는 불완전한 요인들(Fig. 14 b의 검정색 영역)이 존재하고, 오퍼레이터가 이 영역을 좌표로 입력할 경우 입력좌표로 인한 오판별율이 높아질 수 있다. 즉, 플러드 필에서 사용자 좌표 입력의 오류를 최소화 시키는 방법에서 그레이영상을 이용한 방법보다 제안방법이 좋은 결과를 도출할 수 있는 가능성이 높다.

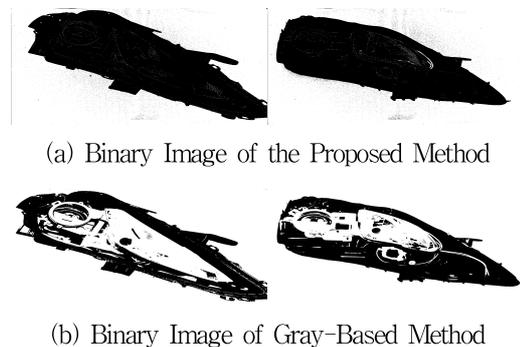


Fig. 14 Analysis and Comparison of Proposed Method and Gray-Based Method

제안방법과 그레이영상을 이용한 방법 그리고 라인 검출을 이용한 방법 처리속도를 비교한 결과, Fig. 15와 같이 제안방법 평균 0.331s, 그레이영상을 이용한 방법 0.263s, 라인 검출을 이용한 방법 3.050s로 그레이영상을 이용한 방법이 가장 좋은 성능을 보이는 반면 라인 검출을 이용한 방법이 가장 안 좋은 성능을 보인다. 하지만 그레이영상을 이용한 방법의 경우 Fig. 14에서 설명한 것과 같이 플러드 필에서 사용자 좌표 입력 오류를 고려한다면 제안방법이 사용자 측면에서는 더 좋다고 할 수 있고, PLC모듈과 모터구동 시간 등의 전체적인 시스템 관점에서 약 0.07s의 경우 미세한 차이로 할 수 있다.

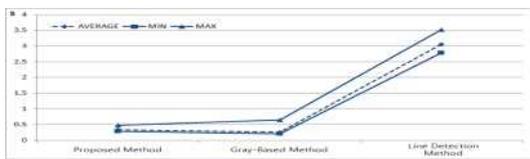


Fig. 15 Proposed Method vs Gray-Based Method vs Line Detection Method

제안한 방법에서 OpenMP를 적용한 처리속도와 비적용한 처리속도를 비교결과는 Fig. 16과 같이 OpenMP를 적용한 결과가 평균 약 51.18% (0.248s) 더 좋은 결과를 보이는 것을 알 수 있고, 제안방법이 OpenMP를 적용하지 않더라도 라인 검출을 이용한 방법보다 좋은 성능을 보임을 알 수 있다.

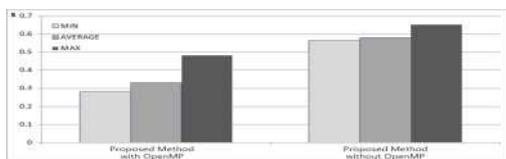


Fig. 16 Proposed Method with OpenMP vs without OpenMP

5. 결론

본 논문에서는 자동차 헤드라이트를 조립하는 과정 중에 이중 부속부품 사용으로 인한 불량을

자동으로 검사하기 위해 헤드라이트 부품의 차종을 판별하고 얼라인먼트를 보정하는 기술을 제안하였다. 차량 부품 분류 및 얼라인먼트 보정 방법에서 중요한 요인은 정확도 및 처리속도라 할 수 있는데, 이러한 맥락에서 본 논문에서는 병렬처리를 적용하여 처리속도를 개선하고, 생산품의 정확한 위치와 각도를 이용하여 차종의 사이즈를 측정 후 차량의 부품을 분류하는 방법을 제안하였다.

세부적인 성능 비교를 목적으로 제안방법과 그레이영상을 이용한 방법 그리고 라인 검출을 이용한 방법의 판별성능과 처리속도를 비교하였다. 판별성능은 제안방법과 라인 검출을 이용한 방법이 가장 좋은 성능을 보였지만 라인검출 방법보다 제안 방법이 더 좋은 성능을 보였고, 처리속도는 그레이영상을 이용한 방법이 가장 좋은 성능을 보였으나 사용자 좌표 입력 오류를 고려한다면 제안방법이 사용자 측면에서는 더 좋다고 할 수 있다. 또한 처리속도 향상을 목적으로 CPU기반 병렬처리방법인 OpenMP를 적용하였다.

본 논문에서는 CPU기반 병렬처리방법인 OpenMP를 적용하여 두 개의 차종에 대해서만 차량용 헤드라이트를 분류 방법을 제안하였다. 향후, 더 많은 헤드라이트를 추가하여 실험할 필요가 있고, 정확한 성능을 검증하기 위해 공정과정 시스템에 적용할 필요성이 있다.

References

[1] Moon C.B., Kim H.S., Kim H.Y., Lee D.W., Kim T.-H., Chung H., Kim B. M., "A Fast Way for Alignment Marker Detection and Position Calibration", Korea Information Processing Society, Vol.5, No.1 pp.35~42, 2016. 6

[2] Moon C.B., Ahn Y.H., Lee H.-Y., Kim B. M and Oh D.-W., "Implementation of Automatic Detection System for CCFL's Defects based on Combined Lighting", Journal of the Korea Society Industrial Information System, Vol.15, No.2, June, 2010.

[3] Jung U.-K., Moon C. B., Lee H.-Y., Kim B. M., Yang H. S., "Preprocessing Algorithm for Detecting CCFL Defects", 2019 Conference on Korean Institute of Information Scientists and Engineers, Vol.36, No.2, pp359-362, 2009.

[4] Kwon S. J., Park C and Lee S. M., "Kinematics and Control of a Visual Alignment System for Flat Panel Displays," Journal of Institute of Control, Robotics and Systems, Vol.14, No.4, Apr., pp.369-375, 2008.

[5] Park C. S. and Kwon S. J., "An Efficient Vision Algorithm for Fast and Fine Mask-panel alignment," Proc. of 2006 SICEICASE International Joint Conference (SICE-ICCAS 2006), pp.1461-1465, Oct., 2006.

[6] <http://opencv.jp/sample/estimators.html>

[7] Shane Torbert, Shane, Applied Computer Science (Second Edition), Springer, p.158, 2016

[8] Hwang Sun-gyu, Image Processing Programming By Visual C++, Hanbit Publishing Network, 2007

[9] <https://www.slideshare.net/kohta/particle-filter-tracking-in-python>.



문 창 배 (Moon Chang Bae)

- 정회원
- 2007년 : 금오공과대학교 소프트웨어공학과 공학사
- 2010년 : 금오공과대학교 소프트웨어공학과 공학석사
- 2013년 : 금오공과대학교 소프트웨어공학과 공학박사
- 2013년 ~ 2014년 : 영풍전자(주) 선임연구원
- 2014년 ~ 현재 : 금오공과대학교 ICT융합특성화센터 연구교수
- 관심분야 : 인공지능, 영상처리, 지식검색, 감성공학



김 근 홍 (Kim Kun Hong)

- 정회원
- 1998년 : 금오공과대학교 컴퓨터공학과 공학사
- 1997년 ~ 2000년 : 에이스 시스템 SE팀
- 2000년 ~ 2002년 : CLS 자동차 사업부
- 2002년 ~ 현재 : 프로텍 코퍼레이션 수석연구원
- 2015년 ~ 현재 : 금오공과대학교 컴퓨터IT학과 석사과정
- 관심분야 : 인공지능, 공장자동화, 영상처리



김 병 만 (Kim Byeong Man)

- 정회원
- 1987년 : 서울대학교 컴퓨터공학과 학사
- 1989년 : 한국과학기술원 전산학과 석사
- 1992년 : 한국과학기술원 전산학과 박사
- 1992년 ~ 현재 : 국립금오공과대학교 교수
- 1998년 ~ 1999년 : 미국 UC, Irvine 대학 방문교수
- 2005년 ~ 2006년 : 미국 콜로라도 주립대학 대학 방문교수
- 관심분야 : 인공지능, 정보검색, 정보보안



오 득 환 (Oh Dukhwan)

- 정회원
- 1982년 : 경북대학교 전자공학과 (전산 모듈) 학사
- 1985년 : 한국과학기술원 전산학과 석사
- 1993년 : 한국과학기술원 전산학과 박사
- 1989년 ~ 현재 : 국립금오공과대학교 교수
- 관심분야 : 정보시스템 성과, e-Learning, Neural Network