

A Fast Way for Alignment Marker Detection and Position Calibration

Moon Chang Bae[†] · Kim HyunSoo^{††} · Kim HyunYong^{†††} · Lee Dongwon^{††††}
Kim Tae-Hoon^{†††††} · Chung Hae^{††††††} · Kim Byeong Man^{†††††††}

ABSTRACT

The core of the machine vision that is frequently used at the pre/post-production stages is a marker alignment technology. In this paper, a method to detect the angle and position of a product at high speed by use of a unique pattern present in the marker stamped on the product, and calibrate them is proposed. In the proposed method, to determine the angle and position of a marker, the candidates of the marker are extracted by using a variation of the integral histogram, and then clustering is applied to reduce the candidates. The experimental results revealed about 5s 719ms improvement in processing time and better precision in detecting the rotation angle of a product.

Keywords : Alignment Marker, Template Matching, Integral Histogram, Candidate Clustering, Projection Integral Histogram

Alignment Marker 고속 인식 및 위치 보정 방법

문창배[†] · 김현수^{††} · 김현웅^{†††} · 이동원^{††††} · 김태훈^{†††††} · 정해^{††††††} · 김병만^{†††††††}

요 약

얼라인(Align) 보정은 제품 생산 전/후 빈번하게 사용되는 머신비전 기술 중 하나이다. 본 논문에서는 생산품에 각인된 마커(Marker) 또는 생산품에 존재하는 유니크한 패턴을 이용하여 생산품의 각도와 위치를 고속으로 판별하고 보정하는 방법을 제안하였다. 본 논문에서 사용한 방법은 템플릿매칭(Template Matching)의 속도를 개선한 적분 히스토그램(Integral Histogram)의 변형을 이용하여 후보들을 추출하고, 클러스터링을 적용하여 후보들을 축소하는 방법을 적용 후 마커의 각도와 위치를 판별하는 방법을 제안하였다. 실험결과, 클러스터링을 적용하기 전 보다 클러스터링을 적용 후 약 5s 719ms 개선된 것을 알 수 있었고, 각도 판별에서도 우수한 성능을 보임을 확인할 수 있었다.

키워드 : 얼라인먼트 마커, 템플릿 매칭, 구간 히스토그램, 후보 클러스터링, 프로젝션 적분 히스토그램

1. 서 론

산업이 발전함에 따라 생산시스템 역시 발전하는 추세이고, 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 비용

을 절감하거나 생산성이 향상시킬 수 있는 중요한 기술이라 할 수 있다.

머신비전 기술은 제품생산 전/후를 기준으로 두 가지 기술로 분류할 수 있다. 그 첫 번째가 생산 공정에서 사용하는 생산품의 위치를 보정하는 기술이고, 두 번째가 생산품의 불량률 판별 기술[1]이다. 두 기술의 공통 핵심기술은 얼라인(Align)을 보정하는 기술이라 할 수 있고, 이 기술은 생산품에 각인된 마크를 이용하거나 생산품에 존재하는 유니크한 패턴을 이용하여 생산품의 각도와 위치를 파악하고, 파악한 각도와 위치를 보정할 수 있도록 하드웨어에 메시지를 전송하는 기술이다. 하지만 생산품의 경우 공정 사정에 따라 반복된 패턴을 사용하는 경우도 발생할 수 있다.

얼라인 기술에서 중요한 요인은 정확도 및 처리속도라 할 수 있다. 정확도의 경우 입력된 생산품의 정확한 각도와 위치를 판별하는 것이고, 처리속도의 경우 정확한 각도와 위치를 판별하는데 소요되는 시간이다. 이러한 맥락으로 본

※ 본 논문은 중소기업청에서 지원하는 2014년도 산학협력 기술개발 사업(No. C0190640)의 연구수행으로 인한 결과물임을 밝힙니다.

† 준 회원 : 금오공과대학교 ICT-CRC 연구교수

†† 준 회원 : 디투이모션(주) 책임연구원

††† 비 회원 : 금오공과대학교 전자공학과 석사과정

†††† 비 회원 : (주)한화 구미사업장 선임연구원

††††† 비 회원 : (주)노바소프트 대표

†††††† 비 회원 : 금오공과대학교 전자공학부 교수

††††††† 종신회원 : 금오공과대학교 컴퓨터소프트웨어공학과 교수

Manuscript Received : June 5, 2015

First Revision : September 30, 2015

Second Revision : November 3, 2015

Accepted : November 9, 2015

* Corresponding Author : Kim Byeong Man(bmkim@kumoh.ac.kr)

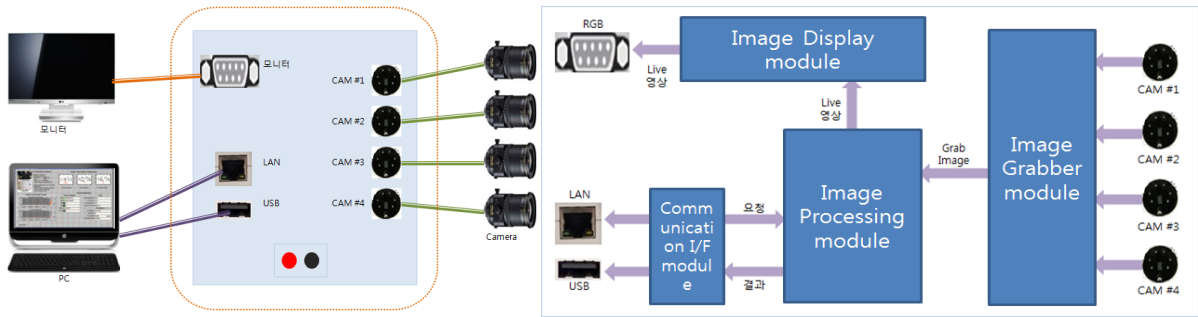


Fig. 1. Hardware System Structure

논문에서는 생산품의 마커 또는 유니크한 패턴을 고속으로 인식하고, Marker 또는 유니크한 패턴의 각도와 위치를 검출하는 방법과 반복된 패턴을 검출하는 방법을 제안한다. 기존 얼라인 기술에는 기하적 템플릿 매칭[5, 6]방법이 존재하고, 기하적 템플릿 매칭방법은 얼라인 마커가 원과 십자 모양 등의 비교적 단순한 마크를 이용해야 하지만 본 논문에서는 제품에 포함된 패턴을 이용하는 방법이다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 적용하는 기술의 하드웨어 시스템 구조를 설명하고, 3장에서는 본 논문에서 제안한 얼라인 마커 인식 및 위치(보정)값 산출 방법, 4장에서는 본 논문에서 제안한 방법의 정확도와 처리속도에 대한 실험결과, 마지막으로 5장에서는 본 논문을 결론짓도록 한다. 또한 본 논문에서는 기존 방법들과 성능 비교를 하였다.

2. 하드웨어 시스템 구조

본 논문에서 제안한 방법을 적용하기 위한 하드웨어는 Fig. 1과 같이 모니터 포트, USB포트, 네트워크 통신포트, 카메라 포트로 구성되고 하드웨어 시스템 구조의 각 기능은 다음과 같다.

- 이미지 디스플레이 모듈(Image Display module) : 캡처된 이미지를 모니터로 출력한다.
- 네트워크 통신포트 및 통신 I/F 모듈 : 네트워크 통신을 통하여 원격에 위치한 데이터베이스에 진행 중인 작업 내용의 결과를 저장한다.
- USB 포트 및 통신 I/F 모듈 : 비상시에 사용하는 포트로 네트워크 마비를 대체한다.
- 이미지 그래버(Image Grabber) 모듈 및 이미지 프로세싱(Image Processing) 모듈 : 이미지 데이터를 수신 및 생산품의 얼라인을 보정하기 위한 알고리즘을 처리하는 모듈이고, 본 논문에서는 이미지 프로세싱 모듈에서 사용할 알고리즘을 제안한다.

얼라인을 보정하기 위한 하드웨어는 Fig. 2와 같이 생산품이 입력되면 광학현미경 카메라를 장착한 FPGA가 각도와 위치를 보정하는 기능을 수행한다. 하지만 광학현미경 카메라를 장착한 FPGA를 제작할 경우 고 비용이 발생할

수 있고, 이의 문제점을 해결하기 위한 방법은 검증된 알고리즘을 사용하는 것이다. 이러한 맥락으로 본 논문에서는 광학현미경 카메라를 장착한 FPGA를 제작하기 이전 본 논문에서 제안한 방법을 검증하기 위한 목적으로 PC에서 제안한 방법의 성능을 검증하였다.

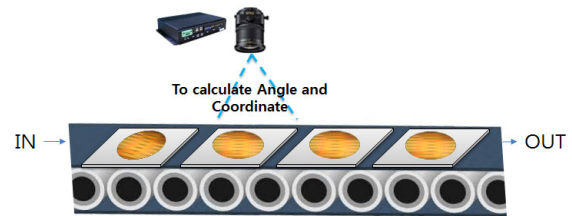


Fig. 2. Alignment System

3. 얼라인 마커 인식 및 위치(보정)값 산출 방법

본 논문에서 제안하는 방법은 Fig. 3과 같이 두 개의 처리과정을 수행한다. 그 첫 번째가 PC에서 마커위치 또는 유니크한 패턴위치를 설정하면 설정한 영역을 n번 회전하여 n개의 특징을 획득하는 과정이고, 두 번째가 광학현미경 카메라를 장착한 FPGA에서 n개의 특징을 이용하여 생산품의 각도 및 위치를 보정하는 과정이다. 본 논문에서는 유니크한 패턴 및 마커, 반복 패턴을 표준패턴으로 정의한다.

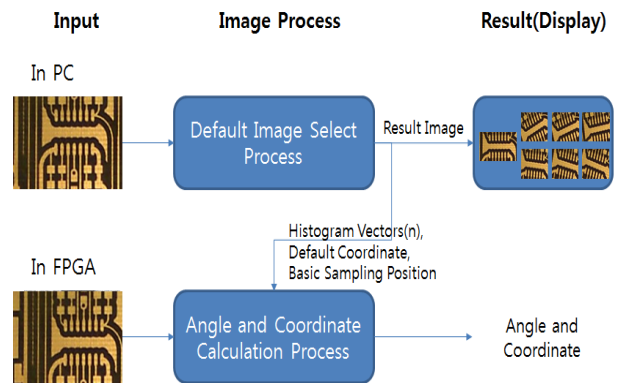


Fig. 3. Software Structure of Alignment System

3.1 표준 패턴 설정 방법

표준 이미지 설정방법은 Fig. 4와 같이 생산품의 위치와 각도가 보정된 이미지를 입력하고, 사용자가 표준 패턴의 위치를 설정하면 FPGA에서 사용할 n 개의 특징벡터가 생성된다. 여기서, n은 회전각도에 의해 결정되는데, 본 논문에서는 -5도 부터 5도까지 0.1단위로 총 101개의 특징 벡터들을 생성하고, 회전이 없는 이미지의 특징을 기본 특징(히스토그램)으로 정의한다.

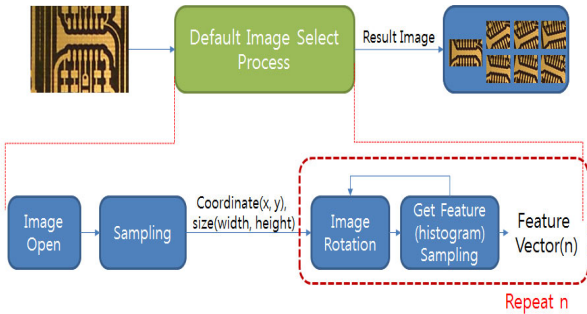


Fig. 4. Default Image Selection Process

표준 이미지 설정방법의 구성은 “이미지 오픈(Image Open)” 단계, “샘플링(Sampling)” 단계, “이미지 회전(Image Rotation)” 단계 그리고 “특징추출(Get Feature(histogram) Sampling)” 단계로 구분되고, 각 단계의 기능은 다음과 같다.

- 이미지 오픈 단계 : 생산품의 위치와 각도가 보정된 이미지를 읽어오는 단계로 사용자가 해당 이미지를 입력한다.
- 샘플링 단계 : User가 이미지에서 표준 위치 또는 마커의 위치를 선택하면 해당영역의 위치 좌표와 사이즈 그리고 이미지 배열을 다음단계로 전송한다.
- 이미지 회전 단계 : 이미지를 n번 반복 회전하는 단계로 앞에서 언급한 것과 같이 -5도 부터 5도까지 0.1단위로 총 101번 회전한다.
- 특징추출 단계 : 회전 후 표준패턴의 특징을 추출하는 단계로 본 논문에서 사용한 특징은 두 개의 프로젝션 히스토그램(Projection Histogram)을 사용한다. 그 첫 번째가 x축 프로젝션 히스토그램이고, 두 번째가 y축 프로젝션 히스토그램[2]이며, 두 개의 프로젝션 히스토그램을 한 쌍으로 사용한다.

3.2 패턴 인식 및 위치(보정)값 산출 방법

패턴을 인식하고 위치(보정) 값을 산출하는 방법은 Fig. 5와 같이 각도와 위치를 보정할 생산품의 이미지가 입력되면 해당 생산품에 포함된 표준 패턴의 각도와 위치를 계산한다.

패턴 인식 및 위치(보정)값 산출 방법의 구성은 “이미지 읽기(Image read)” 단계, “후보영역 선택(Select Candidate Area)” 단계, “후보 클러스터링(Candidate Clustering)” 단계, “각도 및 좌표판별(Measuring Angle and Coordinate)” 단계로 구분되고, 각 단계의 기능은 다음과 같다.

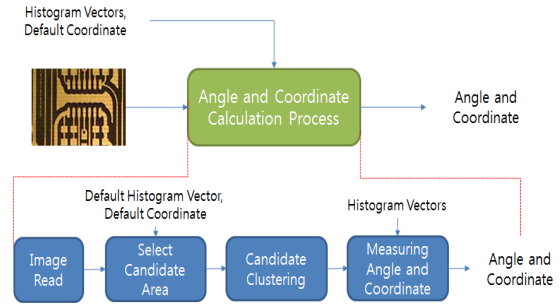


Fig. 5. Process of Marker Detection and Position Calibration

- 이미지 읽기 단계 : 이미지 그래버 모듈에서 이미지를 받아오는 단계이다.
- 후보영역 선택 단계 : 본 단계에서는 기본 특징을 이용하여 후보의 위치를 선택하는 단계로 본 논문에서는 기본적으로 템플릿매칭(Template Matching)[3] 방법의 속도를 개선한 적분히스토그램(Integral Histogram)[4] 방법을 적용하였다. 하지만, 적분히스토그램방법만을 적용하는 경우 회전각도 파악이 불가능하다. 이를 개선하기 위해 본 논문에서는 프로젝션(Projection) 기반 적분히스토그램방법을 사용하여 m개의 후보를 선택한다. 적분히스토그램[4] 방법과 본 논문에서 제안한 방법의 경우 기본 구조는 유사하지만 적분히스토그램방법의 경우 Fig. 6과 같이 일반 히스토그램 방법을 사용하고, 본 논문에서 제안한 방법의 경우 Fig. 7과 같이 프로젝션 히스토그램을 사용한다.

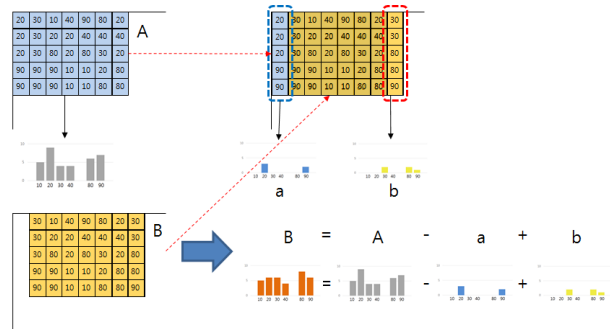


Fig. 6. Integral Histogram Method

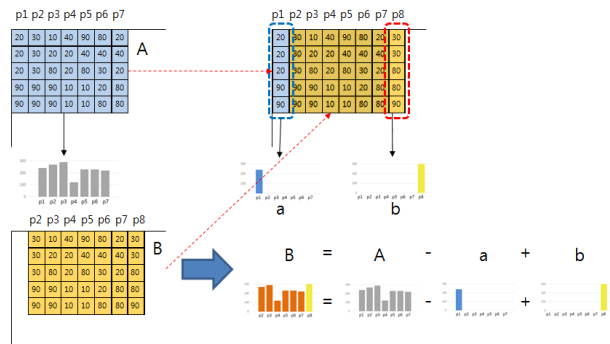


Fig. 7. Proposed Method

일반 히스토그램 경우 이미지의 픽셀값의 수를 카운팅하는 방식으로 전체 히스토그램을 구성한다. 회색 이미지인 경우, 대상 영역에 대해 256차원의 히스토그램 벡터를 얻을 수 있는데, 각 차원의 값은 (해당 차원 - 1)의 픽셀값을 갖는 픽셀의 수를 의미한다. 예로, Fig. 6의 A에서 대상영역의 크기를 5×7이라 할 때, 이 영역에 대한 픽셀값 10은 5개, 20은 9개, 30은 4개, 40은 4개, 80은 6개, 90은 7개이기 때문에 11차원의 값은 5, 21차원은 9, 31차원은 4, 41차원은 4, 81차원은 6, 91차원은 7이고 나머지는 0인 256차원의 벡터를 얻을 수 있다. [4]에서 제시한 적분히스토그램은 이러한 히스토그램을 전체 이미지에 대해 대상영역을 1픽셀씩 이동하면서 구할 경우의 시간을 단축하기 위한 방법이다. 이 방법의 기본 아이디어는 1픽셀 이동한 영역은 이전 영역의 1열을 제외한 나머지 영역은 동일하고 새로운 1열이 추가된 형태이기 때문에, 이 점을 고려하여 이전 영역의 히스토그램 벡터 Fig. 6의 A에서 제외된 1열의 히스토그램 벡터 Fig. 6의 a를 빼고 대신에 추가된 1열의 히스토그램 벡터 Fig. 6의 b를 더해주면 새로운 영역 Fig. 6의 B와 같은 히스토그램 벡터를 구할 수 있다는 것이다.

하지만, 일반 히스토그램인 경우 위치정보가 반영되어 있지 않아 회전각도를 파악할 수 없다. 따라서 본 논문에서 이를 해결하기 위해 일반 히스토그램 대신에 프로젝션 히스토그램을 사용하였다. 프로젝션 히스토그램은 행 또는 열 단위로 그 안에 있는 픽셀들의 픽셀값을 더하는 방식으로 행에 대한 히스토그램과 열에 대한 히스토그램을 계산한다. 따라서, 일반 히스토그램에서는 256차원의 히스토그램 벡터가 생성되지만, 프로젝션 히스토그램 방법에서는 행 또는 열 크기의 벡터가 생성된다. 본 논문에서는 행과 열 히스토그램 모두를 사용하였다. 예로, Fig. 6A에서 대상영역의 크기를 5×7이라 할 때, 행 히스토그램 벡터는 (290, 210, 280, 390, 450)이고 열 히스토그램 벡터는 (240, 270, 290, 100, 230, 250, 240) 이다. 프로젝션 히스토그램에 대해서도 전체 이미지에 대해 대상영역을 1픽셀씩 이동하면서 구할 경우의 시간을 [4]와 유사하게 단축시킬 수 있다. 이 방법의 기본 아이디어도 1픽셀 이동한 영역은 이전 영역의 1열을 제외한 나머지 영역은 동일하고 새로운 1열이 추가된다는 것에 바탕을 두고 있다. 이 점을 고려하여 이전 영역의 프로젝션 열 히스토그램 벡터 Fig. 7의 A를 1차원 좌측 이동하고 대신에 마지막 차원의 값으로 새로 추가된 열의 픽셀값 합으로 하면 새로운 영역 Fig. 7의 B의 열 히스토그램을 얻을 수 있다. 즉, (270, 290, 100, 230, 250, 240, 310)을 얻을 수 있다. 행 히스토그램 벡터는 이전 영역의 프로젝션 행 히스토그램 벡터에서 제거된 열 벡터를 빼고 대신에 추가된 열의 열 벡터 Fig. 7의 b를 더해주면 새로운 영역 Fig. 7의 B의 히스토그램 벡터를 구할 수 있다. 즉, 프로젝션 행 히스토그램 벡터로 (290, 210, 280, 390, 450) - (20, 20, 20, 90, 90) + (30, 30, 80, 80, 90) = (300, 220, 340, 380, 450)를 얻을 수 있다.

이 단계에서 m개의 후보를 선택하는 목적은 1개의 후보

를 사용할 경우 정확한 생산품의 각도와 위치 (Fig. 8 적색 박스의 좌측 상단 x축과 y축 좌표) 검출이 불가능하여 m개의 가장 유사도가 높은 후보를 사용하고, m은 4장에서 실험으로 선택 하였다.

- 후보 클러스터링 단계 : 이전 단계에서 선택한 m개의 후보를 모두 사용할 경우 Fig. 8의 적색 박스와 같이 유사도가 높은 후보들이 하나의 지점에 몰리는 증상이 발생하여 의미 없는 연산이 추가될 수 있고, 이는 영상 처리속도를 저하시키는 원인이 될 수 있다. 이를 개선하기 위한 목적으로 “후보영역 선택 단계”에서 획득한 후보들의 좌표를 이용하여 클러스터링을 진행하였는데, 본 논문에서는 시스템의 특성을 고려하여 Leader-Follower 클러스터링[7]을 응용한 방법을 사용하였다. Fig. 8의 우측 이미지는 보안상의 문제로 실제 마커에 모자이크처리를 하였다.

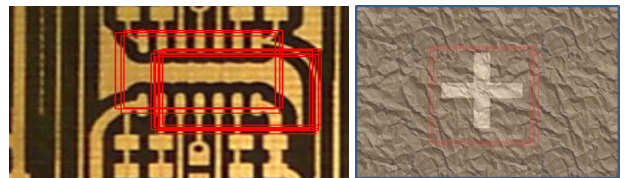


Fig. 8. Selection Result of Candidate Areas (red box: candidate)

본 논문에서 사용한 클러스터링은 Fig. 9와 같이 각 후보들의 x축 거리와 y축 거리를 계산 후 임계값보다 작으면 같은 클러스터로 포함시키고, 임계값 보다 크면 서로 다른 클러스터로 배정하는 방법을 사용한다.

하나의 클러스터의 경우 k개의 후보로 인하여 k개의 좌표가 존재한다. 본 논문에서는 k개의 좌표들 중에서 대표좌표를 선택하기 위해 유사도를 사용한다. 예로, 만약 어떤 클러스터가 두 개의 좌표로 구성되고, 첫 번째 후보의 좌표가 (10, 10) 유사도가 99%이고, 두 번째의 좌표가 (11, 11) 유사도가 95%인 경우, 첫 번째 후보의 유사도가 더 좋기 때문에 클러스터의 대표 좌표는 (10, 10)이 된다.

```

clusterNum = 1
first candidate cluster = clusterNum
loop from second candidate cluster(i'th candidate index) to the last candidate
{
    loop from j'th candidate index(j=0) to i'th candidate index
    {
        if (i'th candidate cluster != 0) continue;
        calculate x-axis distance from i'th to j'th
        calculate y-axis distance from i'th to j'th

        if (y-axis distance > y-axis threshold && x-axis distance > x-axis threshold)
        {
            i'th candidate cluster = j'th candidate cluster
            if (i'th candidate similarity > j'th candidate similarity) (delete j'th candidate)
            else (delete i'th candidate)
        }
    }
    if (i'th candidate cluster == 0) i'th candidate cluster = clusterNum++;
}
    
```

Fig. 9. Candidate Clustering

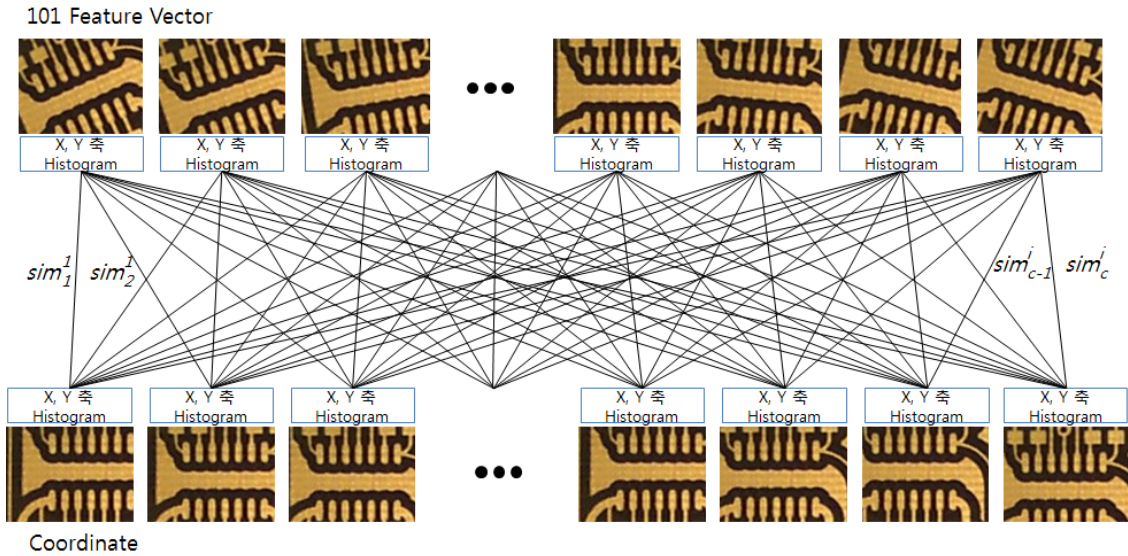


Fig. 10. Measuring Angle and Coordinate Method

- 각도 및 좌표판별 단계 : 최종적으로 표준 패턴의 각도와 좌표를 판별하는 단계로 “3.1 표준설정 방법”에서 획득한 101개의 특징벡터와 “후보 클러스터링 단계”에서 획득한 후보들을 이용하여 생산품의 회전각도와 위치를 판별한다. 즉, Fig. 10과 같이 특징벡터가 101개이고, “후보 클러스터링 단계”에서 획득한 후보의 수가 c 인 경우 총 $(101 \times c)$ 개의 유사도(sim)를 Equation (1)을 이용하여 계산한다.

Equation (1)에서 $histo_n^h$ 는 특징벡터의 히스토그램 중 n 번째 값을 의미하고, $histo_n^c$ 는 “후보 클러스터링 단계”에서 획득한 후보 c 의 히스토그램 중 n 번째 값을 의미한다.

$$sim_c^i = \sum_{n=0}^{len} |histo_n^i - histo_n^c| \quad (1)$$

where: i = 특징벡터인덱스,

len = 히스토그램길이, c = 후보인덱스

$(101 \times c)$ 개의 유사도(sim)를 계산 후 Equation (2)를 적용하여 생산품의 각도와 위치를 판별한다.

$$[r, p] = \maxarg(sim_1^1, \dots, sim_c^1, sim_1^2, \dots, sim_c^2, \dots, sim_1^{i-1}, \dots, sim_c^{i-1}, sim_1^i, \dots, sim_c^i) \quad (2)$$

where: r = 회전각도, p = 위치,

i = 특징벡터인덱스, c = 후보인덱스

4. 실험 및 성능평가

본 논문의 실험환경은 Intel i5 2.40GHz, 8G Ram에서 실험을 하였고, 두 가지 관점에서 성능을 평가하였다. 그 첫

번째가 처리속도이고, 두 번째가 정확도이다. 실험에 사용한 생산품의 이미지는 총 35종의 이미지를 사용하였다.

실험 및 검증에 사용할 이미지의 수가 부족한 관계로 본 논문에서는 기본 이미지를 이용하여 실험에 사용할 이미지를 생성하였다. 즉, “3.1의 표준 이미지를 설정”에 사용한 생산품 이미지를 이동변환 후 회전변환하여 “3장의 얼라인 마커 인식 및 위치(보정)값 산출” 알고리즘 검증에 사용하였다. 다시 말해서, 하나의 생산품 이미지를 대상으로 한 번의 이동변환과 0.1도 단위로 101번 회전하여 생성한 이미지를 검증에 사용하였고, 총 3,535(이미지 35장 \times 101번 회전)장의 생산품 이미지를 이용하여 제안한 “얼라인 마커 인식 및 위치(보정)값 산출” 알고리즘의 검증에 사용하였다.

본 제안 방법의 성능 개선은 두 가지 요소에 기인하는데, 하나는 클러스터링에 의한 유사후보 제거이고, 다른 하나는 프로젝트 히스토그램의 고속 계산이다. 먼저, 첫 번째 요인의 효과를 검증하기 위해 클러스터링 적용 전 정확도에 대한 실험을 실시하였고, 실험결과는 Table 1과 같이 후보의 수가 증가하면 에러의 수가 감소하지만 영상처리 시간은 증가한다는 것을 알 수 있다. 생산 현장에서 사용할 경우 에러가 발생하면 제품에 대한 불량으로 처리된다는 점을 고려하면 20개의 후보를 사용하는 것이 당연하지만 처리시간이 오래 걸려 생산성이 저하된다는 문제점이 발생한다. Table 1에서 처리 시간은 전체 실험대상의 평균 처리시간이다.

20개의 후보를 대상으로 클러스터링을 적용한 결과, 평균 처리시간은 1s 449ms로 클러스터링 적용 전 보다 약 5s 754ms 개선되었고, 에러율은 클러스터링 이전 20개의 후보를 사용한 결과와 동일하다. 즉, 20개의 후보를 이용하여 클러스터링을 적용하는 방법이 클러스터링 적용 전보다 에러율과 처리속도 면에서 좋은 성능을 보인다.

20개의 후보를 대상으로 클러스터링을 적용한 결과, 평균 처리시간은 1s 449ms로 클러스터링 적용 전 보다 약 5s 754ms 개선되었고, 에러율은 클러스터링 이전 20개의 후보를 사용한

결과와 동일하다. 즉, 20개의 후보를 이용하여 클러스터링을 적용하는 방법이 클러스터링 적용 전보다 예러율과 처리속도 면에서 좋은 성능을 보인다.

Table 1. Performance According to the Number of Candidates before Clustering (NofCs: Number of Candidates, NofESs: Number of Error Samples)

NofCs	Processing Time	Mean Error (degree)	Min Error (degree)	Max Error (degree)	NofESs
1	0s 742ms	0.30	0.1	2.4	18
2	1s 106ms	0.27	0.1	2.1	15
3	1s 431ms	0.25	0.1	1.6	12
5	2s 147ms	0.26	0.1	0.8	9
10	3s 843ms	0.30	0.1	0.8	3
20	7s 203ms	0.00	0.0	0.00	0

본 논문에서는 먼저 클러스터링 임계치를 정하기 위해 거리에 따른 실험을 진행하였다. 실험결과, Fig. 11과 같이 x, y축의 거리가 각 4 픽셀 이상인 경우 처리속도는 거의 동일하지만 x, y축의 거리가 각 2 픽셀인 경우 4 픽셀 이상인 경우 보다 처리속도가 늘어난 것을 알 수 있다. 이는 클러스터링의 거리를 정하는 거리 값이 후보의 수에 영향을 주기 때문이라 할 수 있고, 본 논문에서는 클러스터링의 거리 임계값을 4 이상의 값으로 설정하였다.

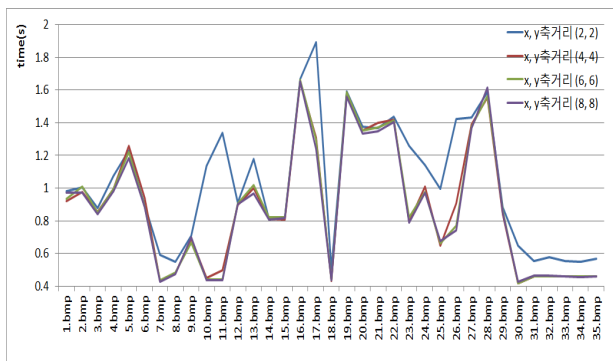


Fig. 11. Result of Clustering Threshold

두 번째 개선 요인, 즉, 프로젝션 히스토그램의 고속 계산의 효과를 보이기 위해 클러스터링을 적용한 일반 템플릿매칭과 제안한 방법의 처리속도 실험을 하였다. 실험결과를 Fig. 12와 같다. 일반 템플릿매칭의 경우 평균 4s 544ms의 처리속도를 보이고, 제안방법의 경우 평균 1s 449ms의 처리속도를 보였다. 약 3s 95ms 성능 향상을 보였다. 특히, 이미지 16, 17, 19, 20, 21, 22 그리고 28의 경우 평균 5s 233ms 성능이 향상된 것을 알 수 있는데, 이는 해당 이미지에 유니크한 패턴이 여러 개 존재하는 경우로, 이러한 경우 제안방법이 일반 템플릿매칭보다 더 좋은 성능을 보임을 알 수 있다.

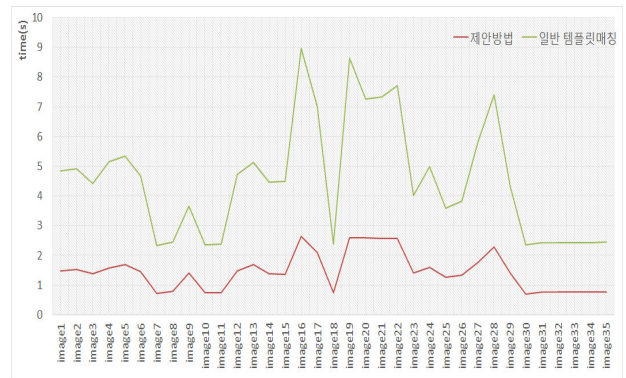


Fig. 12. The Proposed Method vs. Native Template Matching Method

[4]의 방법도 제안방법과 유사한 방법으로 히스토그램을 고속으로 계산한다. 그래서 속도 측면에서는 별 차이가 없다. 하지만 [4]의 방법은 일반 히스토그램을 사용하기 때문에 프로젝션 히스토그램을 사용하는 제안방법에 비해 회전 각도를 제대로 파악하기가 쉽지 않다. 이를 검증하기 위해 본 논문에서는 [4]의 방법과 제안방법의 정확도관련 비교실험을 하였다. 정확도관련 비교실험은 Fig. 13의 이미지를 사용하였다.



Fig. 13. Testing Image

회전각 정확도관련 실험결과, 제안방법의 경우 Fig. 14와 같이 실제 회전각도와 동일한 결과를 보이는 반면, [4]의 방법의 경우 실제 각도를 판별하지 못하는 것을 확인할 수 있었다.

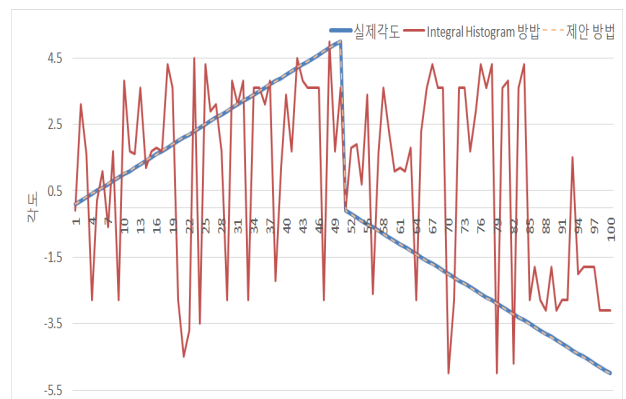


Fig. 14. Detection Accuracy of Rotation Angles

5. 결 론

본 논문에서는 생산품에 각인된 마커, 생산품의 유일한 패턴 그리고 반복된 패턴을 인식하여 생산품의 얼라인을 보정하는 방법을 제안하였다. 본 제안방법에서는 프로젝션 히스토그램을 적용한 적분히스토그램 방법과 클러스터링 기법을 사용하여 표준 패턴의 각도와 위치를 판별하였다.

본 제안방법의 성능 개선은 두 가지 요소에 기인하는데, 하나는 클러스터링에 의한 유사후보 제거이고, 다른 하나는 프로젝션 히스토그램의 고속 계산이다. 처리시간을 개선하기 위해 클러스터링을 적용한 경우, 약 5s 754ms 속도 개선을 얻을 수 있었고, 프로젝션 히스토그램의 고속 계산으로 일반 템플릿매칭보다 평균 3s 95ms 속도 개선을 얻을 수 있었다. 또한, 본 제안 방법은 프로젝션 히스토그램을 사용하기 때문에 일반 히스토그램을 사용하는 방법에 비해 회전 각도를 보다 잘 검출할 수 있었다.

본 논문에서 제안한 방법을 통해 나름 성능 개선을 할 수 있었다. 하지만 실제 현장에 적용하기 위해서는 추가의 속도 개선이 필요하다. 소프트웨어만으로는 현장에 필요한 속도를 얻기에는 한계가 있을 것으로 보인다. 향후, 본 논문에서 제안한 방법을 FPGA로 구현하여 추가의 속도 개선을 얻고자 한다.

References

- [1] C. B. Moon, Y. H. Ahn, H.-Y. Lee, B. M. Kim, and D.-W. Oh, "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.
- [2] S. Huwer, Heinrich Niemann, "2D-Object Tracking Based on Projection-Histograms," London, UK, 1998. Springer-Verlag.
- [3] R. Brunelli, "Template Matching Techniques in Computer Vision: Theory and Practice," Wiley, ISBN 978-0-470-51706-2, 2009
- [4] F. Porikli, "Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces," *Proc. of Int'l Conf. on CVPR*, pp.829-836, 2005.
- [5] S. J. Kwon, C. Park, and S. M. Lee, "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.
- [6] C. S. Park and S. J. Kwon, "An efficient vision algorithm for fast and fine mask-panel alignment," *Proc. of 2006 SICE-ICASE International Joint Conference (SICE-ICCAS 2006)*, pp.1461-1465, Oct., 2006.
- [7] Richard O. Duda et al., "Pattern Classification," Wiley-Interscience, pp.561-563, 2001.



문 창 배

e-mail : moonyeses@naver.com

2007년 금오공과대학교 소프트웨어공학과 (공학사)

2010년 금오공과대학교 소프트웨어공학과 (공학석사)

2013년 금오공과대학교 소프트웨어공학과 (공학박사)

2013년~2014년 영풍전자(주) 선임연구원

2014년~현 재 금오공과대학교 ICT-CRC 연구교수

관심분야: 인공지능, 영상처리, 정보검색, 감성공학



김 현 수

e-mail : hyunsoo@d2emotion.com

2008년 금오공과대학교 컴퓨터공학부 (공학사)

2010년 금오공과대학교 소프트웨어공학과 (공학석사)

2014년 금오공과대학교 소프트웨어공학과 (공학박사)

2014년~2015년 금오공과대학교 ICT-CRC 박사후과정

2015년~현 재 디투이모션(주) 책임연구원

관심분야: 인공지능, 소프트웨어공학, 디자인패턴



김 현 용

e-mail : kimw89@naver.com

2014년 금오공과대학교 전자공학부(학사)

2014년~현 재 금오공과대학교 전자공학과 석사과정

관심분야: 무선통신, 인공지능, IoT



이 동 원

e-mail : prolinuxer@naver.com

2004년 동명정보대학교 정보통신공학과 (공학사)

2008년 금오공과대학교 소프트웨어공학과 (공학석사)

2012년 금오공과대학교 소프트웨어공학과 (공학박사)

2013년~현 재 (주)한화 구미사업장 선임연구원

관심분야: 수중통신, 모델설계, 임베디드시스템설계



김 태 훈

e-mail : openzero@nate.com
2002년 울산대학교 전자계산학과(학사)
2012년 금오공과대학교 소프트웨어공학과
(공학석사)
2003년~현 재 (주)노바소프트 대표
관심분야: 소프트웨어공학, 디자인패턴,
공정정보관리



김 병 만

e-mail : bmkim@kumoh.ac.kr
1987년 서울대학교 컴퓨터공학과(학사)
1989년 한국과학기술원 전산학과(석사)
1992년 한국과학기술원 전산학과(박사)
1992년~현 재 금오공과대학교
컴퓨터소프트웨어공학과 교수
1998년~1999년 미국 UC, Irvine 대학 방문교수
2005년~2006년 미국 콜로라도 주립대학 대학 방문교수
관심분야: 인공지능, 정보검색, 정보보안



정 해

e-mail : hchung@kumoh.ac.kr
1987년 한양대학교 전자통신공학과
(공학사)
1991년 한국과학기술원 전기및전자공학과
(공학석사)
1996년 한국과학기술원 전기및전자공학과
(공학박사)

1993년~1998년 엘지전자 선임연구원
1998년~현 재 금오공과대학교 전자공학부 교수
2003년~2004년 University of Texas at Dallas 방문교수
관심분야: 가입자액세스망, 5G, 스마트 디바이스