

coding을 한다.

이 보드는 데이터의 고속 Input/Output에 있어서 병목현상(bottleneck)을 없애기 위해 64 비트까지의 넓은 대역폭과 32MHz의 전송속도를 가진 PCI 버스를 이용한 PCI Interface를 채용했고 AISC는 I/O timing과 보드 내부의 status를 측정하기에 적합한 ALTERA tool을 사용하였으며 Register들의 종류와 기능은 표 2.1과 같다.

표 2.1 Linescan Board Register List

name	bit	offset	R/W	설명
HEndRegH	5	0x0 (0)	R/W	capture 할 pixel수 정의
HEndRegL	8	0x1 (1)	R/W	
VEndRegH	2	0x2 (2)	R/W	capture 할 line수 정의
VEndRegM	8	0x3 (3)	R/W	
VEndRegL	8	0x4 (4)	R/W	
HSkipReg	8	0x5 (5)	R/W	skip(dummy) pixel수
ControlReg	8	0x6 (6)	R/W	capture 동작 control
Edge Target Address 3	8	0x7 (7)	R/W	edge좌표 data
Edge Target Address 2	8	0x8 (8)	R/W	
Edge Target Address 1	8	0x9 (9)	R/W	
Edge Target Address 0	8	0xA (10)	R/W	
SkipLineReg 1	8	0xB (11)	R/W	Line skip counter byte
SkipLineReg 0	8	0xC (12)	R/W	
High Period Reg	8	0xD (13)	R/W	sync 주기 counter의 byte
Low Period Reg	8	0xE (14)	R/W	
FIFO Status	8	0xF (15)	R/W	FIFO의 상태 확인, reset
Target Load Flag	8	0x10 (16)	R/W	target load 요구/확인

2.1.3 조명장치

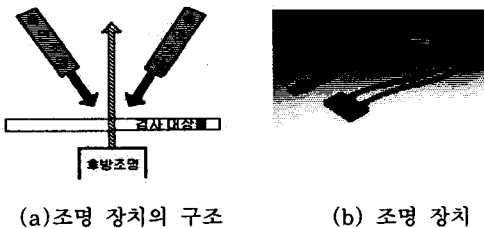
본 시스템에서 사용하는 카메라는 CCD가 선형으로 배열된 Linescan 카메라이고, 검사대상물들은 대부분 폭이 넓은 특징이 있으므로 이를 고려하여 두 종류의 조명을 적용하였다.

1. 측면조명(Side light)

입의 각도로 최대의 대비가 나타나도록 구성한다. 광원은 할로겐램프를 사용하고 광섬유를 통하여 검사물의 표면에 조명하는 방식으로 입의 각도에 따른 검사물의 특성을 살펴보기에 적합하다.

2. 후방조명(Back light)

광원은 형광등을 사용하여 검사대상물에 구멍이 있거나 투명, 반투명인 물체의 표면에 패턴을 검사할 때와 물체의 외곽을 강조하고 때로는 약하게 하기 위해 사용한다. 이러한 조명을 그림 2.3과 같이 조합하였다.



(a) 조명 장치의 구조

(b) 조명 장치



(c) 조명장치 외부함

그림 2.3 시스템에 적용된 조명장치

2.1.4 구동 장치

Linescan 검사는 검사물의 속도와 밀접한 관계가 있기 때문에 구동 장치의 속도는 검사 사양에 중대한 영향을 끼친다. 본 시스템의 구동 장치는 크게 모터와 Inverter, 기구장치로 구성되는데 그림 2.4와 같다.

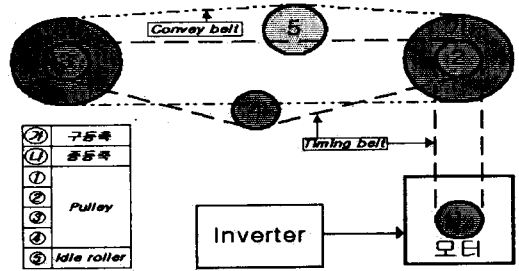


그림 2.4 기구 장치의 구조

헬리컬 가이드 모터와 Microprocessor 제어방식을 사용하여 PWM 방법으로 모터를 제어할 수 있는 frequency inverter를 사용하였다.

2.2 소프트웨어

입력신호에 잡음이 포함된 경우 이를 제거하거나 약화시키는 전 처리를 고속 검사 시스템에서 사용하려면 하드웨어에서 고속 실시간으로 처리하는 것이 바람직하므로 전 처리를 위한 알고리즘을 구현한다.

1. 줄 길이 부호화(Run-length coding)

줄 길이 부호화는 균일한 영역의 부호화에 대한 효율적인 방법으로 영상이나 비트 평면에서 나타나는 자료의 반복성을 이용하여 자료를 압축하는 방법이다. 기본적인 개념은 각 행을 왼쪽에서 오른쪽으로 살펴가면서 만나는 gray값의 연속적인 모임을 이 값에 대응하는 길이로 각각 부호화하고, gray 값을 결정하는 방법을 만든다.

2. 다단계 이치화(Multi-level Threshold)

현재 시스템에서 사용하는 카메라는 PRNU가 10%이다. 즉, 최대 화소값이 255이므로 ± 25.5 gray level의 차이는 잡음 성분이 포함된 차이므로 무시해도 결과에는 큰 영향을 주지 않는다. 실제로 고속처리를 위해서는 줄 길이 부호화 단계에서 영상을 많이 압축할수록 영상처리 단계에서 계산하는 데이터량이 줄어들어서 고속처리가 가능해진다. 그러므로 최소한 25level 차이와 압축률을 고려하여 gray영상을 32로 나눈 8단계 양자화를 하여 부호화를 하는데 MSB(most significant bit) 3bit만을 사용하여 양자화를 하고, 나머지 5bit는 사용하지 않는다. 특히, 홀(hole) 검사와 같이 흑백이 명확히 구분되는 경우에는 이치화만을 사용한다.

3. 검사 알고리즘

ASIC 처리시간은 소프트웨어와 비교할 수 없을 정도로 빠르며 데이터 압축률은 8단계 양자화의 경우 256 level에 비해 약 10배의 압축률을 보이며 이치화 영상의 경우 약 100배의 압축효과를 볼 수 있다. 일반적인 2차원 영상인 경우 검사 시 필요한 화소만큼의 위치를 이동하여 검사하기 때문에 처리시간은 길지만 처리 방법은 간단하다. 그러나 줄 길이 부호화 된 영상은 가로축으로 입의 위치로 이동하기 위해서는 줄의 시작위치부터 탐색을 해야하기 때문에 처리 시간은 짧지만 방법은 그만큼 복잡해진다. 여기서는 본 시스템에서 제안된 알고리즘에 대해서 서술한다.

(1) 고속적응 줄 매칭(Fast Adaptive line Matching)

패턴 매칭의 중요한 점은 검사 패턴과 마스터 패턴의 위치 정렬(position alignment)이다. 이를 위해서 패턴에 기준 마크(fiducial mark)를 표시하여 정렬 정보를 제공하게 된다. Linescan의 경우 검사물이 일정하게 흘러가기 때문에 각도 정렬(angle alignment)은 고려하지 않아도 된다.

기준 마크 정보를 기준으로 마스터와 검사 패턴의 차이를 차(差)run-length로 구하는 방법을 설명하기로 한다.

가. 마스터와 현재 run-length의 edge 좌표의 위치가 가까운 것을 차(差)run-length의 좌표에 저장한다.

나. 차(差)run-length의 gray에 마스터와 현재 run-length의 gray 차(差)를 저장한다.

다. 위치를 비교할 경우 좌우로 이동한 거리를 반드시 고려한다.

이러한 방법은 2차원 영상인 경우 모든 좌표에 대해서 비교를 하지만 압축된 run-length인 경우 edge 위치에 서만 비교를 하기 때문에 계산시간이 많이 줄어든다. 앞에서 8단계 gray run-length인 경우 약 10배의 데이터 압축효과가 있다고 했는데 이럴 경우 비교 시간은 1/10로 줄어든다. 마찬가지로 이치화한 run-length의 경우는 1/100로 줄어든다.

(2) Labling

본 시스템은 run-length code를 이용하여 Labling을 하기 때문에 기존의 방법들과는 차이가 있으며 새롭게 고안된 방법이다. 이를 간략히 나타내면 다음과 같다.

가. Lable의 구조

- (1)number : Lable의 전체개수를 나타낸다.
- (2)line[*] : line의 위치를 표시한다.
- (3)start[*] : 시작점의 위치를 표시한다.
- (4)end[*] : 끝점의 위치를 표시한다.
- (5)finish[*] : label의 상태를 표시한다.
 - (가) 0 : label data가 없음을 나타낸다.
 - (나) 1 : label이 진행중임을 나타낸다.
 - (다) 2 : label이 끝났음을 나타낸다.

나. Run-length로부터 Lable을 만드는 방법

(1)변수의 초기화 : number와 finish를 0으로 한다.

(2)Run-length data 유효성 검사

(가) 길이에 대한 유효성 검사

최소 길이와 최대 길이를 정하여 이 범위 안에 들어오는 것만을 Lable에 포함시킨다.

(나) gray 값에 대한 유효성 검사

차(差) run-length인 경우 gray값이 2이상인 경우 무조건 Lable에 포함시킨다. 홀 검사와 같이 이치화 run-length인 경우 임의의 threshold값 이하(흰색 바탕에 홀이 흰색인 경우)로 나누어서 처리한다.

(3) 새로운 Lable인 경우

Finish와 number를 1로 하고, line, start, end에 현재의 run-length 데이터를 저장한다.

(4) 추가된 Lable인 경우

Number를 증가시키고, line, start, end에 현재 run-length 데이터를 저장한다. 이전 Lable들 중 마지막 line이 현재 run-length와 교차하는지 확인한다. 다음은 이전 label과 교차성을 판단하는 기준이다.

(가) 이전 label의 start가 현재 run-length의 start와 end사이에 있는 경우

(나) 이전 label의 end가 현재 run-length의 start와 end사이에 있는 경우

(다) 이전 label이 현재 run-length를 포함하는 경우

(라) 현재 run-length가 이전 label을 포함하는 경우

이를 그림으로 나타내면 그림 2.5와 같다.

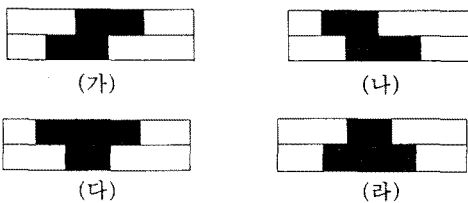


그림 2.5 Label과 run-length의 차

(5) 현재 run-length에 대한 결과 처리 종료

한 줄이 끝나면 모든 Lable에 대하여 마지막 line이, 현재의 line이 아니면 새로 추가된 데이터가 없는 경우이므로 finish를 2로 하여 Lable을 닫는다.

다. Labling

앞에서 설명한 레이블링 기법을 그림 2.6의 임의의 영상을 이용하여 설명하면 다음과 같다.

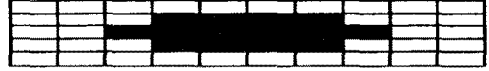


그림 2.6 Labling을 위한 임의의 영상

표 2.2 그림2.5에 대한 run-length code

Edge	EOL		
gray	7		
Edge	2	6	EOL
gray	7	0	7
Edge	1	7	EOL
gray	7	0	7
Edge	2	6	EOL
gray	7	0	7
Edge	EOL		
gray	7		

(1) 변수 초기화

number와 finish를 0으로 한다.

표 2.3 초기화된 Lable

Label	1	number	0	finish	0	
	1	2	3	4	5	...
Line						...
Start						...
End						...

(2) 새로운 Lable만들기

number와 finish를 1로 한다. line은 2이고, start는 3, end는 7이다.

표 2.4 새로 생성된 Lable

Label	1	number	0	finish	0	
	1	2	3	4	5	...
Line						...
Start						...
End						...

(3) 추가된 Lable붙이기

number를 2로 한다. line은 3이고, start는 2, end는 8이다.

number를 3으로 한다. line은 4이고, start는 3, end는 7이다.

표 2.5 추가된 Lable

Label	1	number	3	finish	1	
	1	2	3	4	5	...
Line	2	3	4			...
Start	3	2	3			...
End	7	8	7			...

(4) lable 종료하기

finish 2는 종료를 나타내며 이후에 다른 line을 붙일 수 없다.

표 2.6 종료된 Lable

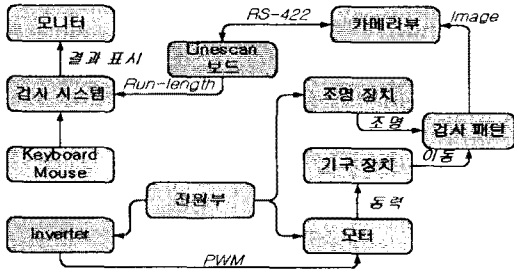
Label	1	number	3	finish	2	
	1	2	3	4	5	...
Line	2	3	4			...
Start	3	2	3			...
End	7	8	7			...

2.3 적용 결과

본 절에서는 2.1의 하드웨어와 2.2의 소프트웨어를 합쳐서 하나의 시스템을 구성하고 이를 인쇄된 web과 홀 검사에 적용하여 시스템의 성능을 평가한다.

2.3.1 시스템 구성

1. 전체적인 구성



시스템은 검사패턴을 구동하는 구동부분과 검사와 관련된 검사 부분으로 나눌 수 있다. 이를 간략하게 그림 2.7에 나타내었다.

그림 2.7 시스템의 전체적인 구성

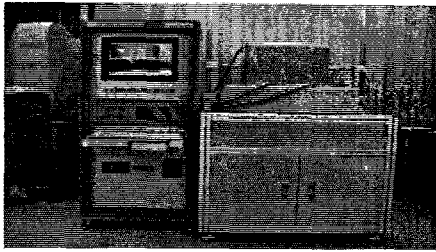


그림 2.8 시스템의 외관

가. 구동부분

구동부분은 Inverter, 모터, 기구장치로 구성되며 기구장치에서는 구동축과 종동축이 검사 패턴을 일정한 속도로 이동시킨다.

나. 검사부분

검사부분은 조명 장치, 카메라와 렌즈, Linescan 보드, 검사 시스템(PC)으로 구성된다. 카메라는 획득한 영상을 RS-422형태로 Linescan 보드에 전송하고 전송된 data는 보드내부에서 형태론적 전 처리와 run-length 과정을 거쳐 PC의 메모리에 run-length code를 저장하고 영상획득이 끝났음을 프로그램에게 알린다. 프로그램은 다시 영상획득 명령을 내리고 현재 메모리에 저장된 code를 읽는다. 프로그램 내부에서 code를 처리하여 그 결과를 모니터에 표시하는 것으로 한 cycle을 종료한다.

2.3.2 검사 결과

1. 검사 사양

- 가. 전체 길이 : 1800(mm)
- 나. 전체 폭 : 500(mm)
- 다. 패턴 폭 : 400(mm)
- 라. 패턴 길이 : 300(mm)
- 마. 기준 거리 : 400(mm)
- 바. 초당 이동 거리 : 785(mm/sec)

2. 검사 방법

검사는 앞에서 제안된 영상의 전 처리 알고리즘과 검사 알고리즘을 통해서 실행되며 전체적인 검사방법은 아래의 그림 2.9의 순서도와 같다.



그림 2.9 검사 알고리즘 순서도

3. 검사 예제

(1) Web 검사

검사 패턴의 실제 모양(그림 2.10)과 결합이 포함된 영상(그림 2.11)과의 차를 구해서 검사 알고리즘으로 처리하여 그림 2.12와 같은 결과를 얻었다.



(가) 원(原) 영상

(나) 원(原) 영상

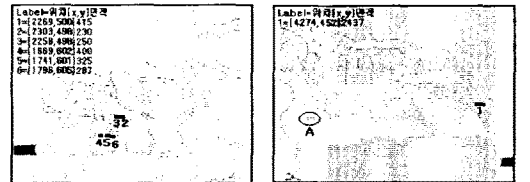
그림 2.10 실제 패턴의 Frame과 fiducial mark



(가) 결합 영상

(나) 결합 영상

그림 2.11 결합이 포함된 영상



(가) 차(差) 영상

(나) 차(差) 영상

그림 2.12 결합을 검출한 영상

결과를 살펴보면 왼쪽 영상은 두 개의 결합이 분리되어 Labling 되었었는데 이것은 후 처리(Merging)를 통해서 하나로 묶어주면 된다. 차 영상에서 A로 표시된 부분의 결합은 gray값의 차가 적기 때문에 다른 배경의 차들과 구분이 되지 않았으나 카메라의 PRNU 10% 오차를 고려하면 gray level 25이하는 무시할 수 있다.

전체적인 검사 시간은 영상 1 frame(800Line)을 획득하는데 약 0.4초가 걸린다. 시스템에서 사용된 PC는 Intel Pentium- 800Mhz이며 이를 표준으로 계산한

각 항목의 검사시간은 표 2.7과 같다.

표 2.7 인쇄된 Web의 검사 시간

항 목	시 간 [mm/sec]
Capture 속도	400
Run-length 읽기	50±10
Fiducial Mark 위치	20
차 run-length	100±20
Labling	100±20
Merging	20
결과 표시	30
총 검사시간	320±50

(2) 홀(Hole) 검사

검사에는 미국 Sheldahl社의 Novaclad(천공 동 박지)를 사용하였으며, Labling과 Merging을 통하여 분리된 홀들은 면적별로 분리하고, 홀의 면적이 기준치보다 작거나 모양이 이상한 것들은 불량으로 판단하여 별도로 불량 홀이나 결함으로 표시했다.

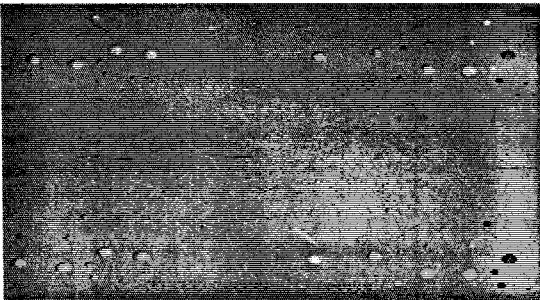
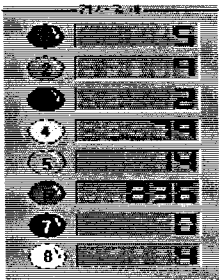
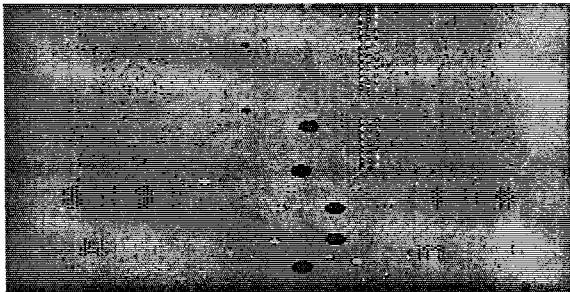


그림 2.13 Sheldahl社의 Novaclad(천공 동 박지)



	매우 큰 hole
	아주 큰 hole
	큰 hole
	중간 hole
	작은 hole
	아주 작은 hole
	불량 hole
	결함

그림 2.14 홀 검사 최종 결과

전체적인 검사 시간은 영상 1 frame(1800Line)을 획득하는데 약 0.6초가 걸린다. 시스템에서 사용된 PC는 Intel Pentium-III 800Mhz이며 이를 표준으로 계산한

각 항목의 검사시간은 표 2.8과 같다

표 2.8 Novaclad(천공 동 박지)의 검사 시간

항 목	시 간 [mm/sec]
Capture 속도	600
Run-length 읽기	100±10
Labling	150±30
Merging	50
결과 표시	50
총 검사시간	350±40

Run-length 읽기, Labling은 영상의 복잡도에 따라 크기가 변하므로 약 20%의 오차가 발생 할 수 있다. 전체 검사 시간은 310~390 [mm/sec]로 검사하기에 충분한 여유가 있다.

3. 결 론

본 논문에서는 Linescan 카메라를 사용하여 고속으로 이동하는 web을 검사하는 시스템에 대해 연구했다. Linescan 보드는 처리속도를 고려하여 PCI용으로 설계, 제작하였으며 ASIC칩을 사용하여 기본적인 기능과 데이터 압축 기능을 추가 및 변경할 수 있도록 하였고, noise제거를 위한 전처리와 빠른 데이터 처리를 위한 run-length coding을 구현하였다. 최종 데이터는 8단계 양자화된 run-length code이고 원래 영상에 비해 4배의 데이터 압축효과와 10배의 처리 효과를 얻었고, 이로 인해 검사 시간도 4배정도 절약되었다.

검사 알고리즘으로 압축된 데이터를 처리하기 위해서 Labling과 Merging에 대한 새로운 방법을 제안하였는데 일반적으로 사용되는 Labling에 비해 성능은 약간 떨어지지만 데이터의 개수가 적으므로 검사속도의 향상과 저장 시간 단축을 이루었다. 현재의 알고리즘은 전처리 과정과 영상압축이 ASIC으로 되어있기 때문에 검사 속도 면에서 기존의 화소 단위처리에 비해 빠른 감정을 지니고 있다. 또한 결함 검출력과 검사 속도를 높이기 위해서 조명장치와 카메라를 수정한다면 세계적으로 활용되고 있는 산업용 검사 장비와 비교하여 검출 능력에 손색이 없는 검사장비가 될 것이며, 영상처리 ASIC기술 및 Windows NT 응용 소프트웨어는 국내 비점속식 자동화 라인 구축과 검사 계측장비 산업발전에 기여할 수 있을 것이다.

(참 고 문 헌)

- [1] 장동식, 이만희, 부창완, Linescan Camera를 이용한 검사시스템에서의 새로운 영상처리 알고리즘, 제어.자동화시스템공학 논문지, 제3권, 제4호, pp406-414, 1997
- [2] R. M. Haralick, and L. G. Shaporp, Computer and Robot Vision, Vol 1, Addison-Wesley, pp329-333, 1990
- [3] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, New York, 1982
- [4] 김한균, 윤정민, 오주환, 최태영, 수리 형태론을 이용한 texture 영상의 방향성 결함 검출, 전자공학회 논문지, 제 33권 B편, 제4호, pp771-777, 1996
- [5] Giardina, C.R. and Dougherty, E.R. Morphological Morphological Method in Image and Signal Processing, Prentice-Hall, Englewood Cliffs, N.J.,1998
- [6] R. Jain, R. Kasturi, and B. G. Schunch, Machine Vision, McGraw-Hill, 1995