

GOVP1199702548

5

L2937
V.2

최종보고서

컴퓨터의 화상처리를 이용한 오이의 자동선별,
등급판정 및 포장시스템 개발

Development of automatic grading, sorting and
wrapping system for cucumbers
by using computer vision.

연구기관

전국대학교

성균관대학교

농림부

제 출 문

농림부 장관 귀하

본 보고서를 “컴퓨터의 화상처리를 이용한 오이의 자동선별, 등급판정 및 포장 시스템 개발”에 관한 연구과제의 최종보고서로 제출합니다.

1996 . 12 . .

주관연구기관명 : 건국대학교

총괄연구책임자 : 성시홍

협동연구기관명 : 성균관대학교

협동연구책임자 : 이대원

요 약 문

I. 제목

컴퓨터의 화상처리를 이용한 오이의 자동선별, 등급판정 및 포장 시스템 개발

II. 연구개발의 목적 및 중요성

오이는 모양 및 규격이 일정하지 않아 이들에 대한 선별, 등급판정 및 포장은 사람의 눈과 손에 의한 가장 단순하고, 반복적인 일관작업의 연속으로 작업자들이 쉽게 싫증을 느끼며, 비능률적이어서 오이의 고품위화, 고부가 가치화 및 저장성향상에 큰 걸림돌이 되고 있었다. 이러한 문제점 등을 해결하기 위하여 첨단 전자공학 기술을 이용한 자동 선별 및 포장 시스템 개발이 요구된다. 따라서 컴퓨터 화상처리를 이용한 오이의 자동선별, 등급판정 시스템을 개발 하였으며, 또한 오이를 날개로 포장할 수 있는 포장시스템도 개발하였다.

III. 연구개발 내용 및 범위

본 연구에서는 오이의 등급결정과 동시에 포장하고 분리하는 시작기를 개발하였다. 연구내용은, 설계제작한 조명장치를 가진 화상

처리 시스템에서 오이 패턴인식과 기하학적 특징에 의하여 오이등급을 선별하였으며, 등급이 결정된 오이는 컨베이어 벨트로 이송된 후 포장시스템에서 날개로 포장하였다. 오이 등급설정의 범위는 사용자가 임의로 할 수 있도록 하였으며, 오이 공급시간을 알려주는 신호장치를 부착하였다. 또한 포장시스템은 오이의 등급 정도에 따라 선별적으로 날개 포장이 가능하도록 하였다.

IV. 연구 개발결과 및 활용에 대한 건의

1차 년도에서는 오이 등급설정을 결정 할 수 있는 조명장치를 가진 화상처리 시스템을 개발하였으며, 2차년도에서 오이의 선별 및 포장 시스템을 개발하였다. 오이 등급을 결정하기 위하여 화상처리시스템을 개발하였으며, 화상처리 시스템에서는 긴 오이의 정확한 화상이미지를 얻기 위하여 45°로 기울어진 거울을 부착한 조명장치를 개발하였으며, 이 장치를 이용하여 오이의 등급설정을 하였다. 오이의 포장은 회전 틀러를 가진 포장시스템을 개발하여 포장하였다. 화상처리 시스템에서 등급이 결정된 오이가 컨베이어로 이송되기 전 랩(Wrap)을 일정한 크기로 절단한 후 이 랩용지로 오이를 포장하였다. 포장된 오이는 체인이 달린 컨베이어 벨트식 분리장치로 운반되어 오이 등급별로 분리상자 넣었다. 즉, 오이가 선별, 포장, 분리되는 일관시스템의 시작기를 개발하였으며, 이를 무역종합전시장(KOEX, '96. 11.)에서 전시하여 많은 사람들에게 작동되는 모습을 보여주었다. 또한 2건의 특허를 출원하였다 [1. 발명의 명칭: 오이 개별 자동 포장

기 (출원번호: 제 96-44462), 2. 발명의 명칭: 거울의 잔상을 이용한
화상처리 장치 및 방법 (출원번호: 제 96-68283)].

이 시스템은 오이의 등급설정과 포장작업 및 분리장치를 동시에
수행할 수 있어 많은 노력절감을 가져올 수 있다. 또한 . 개발된 시
스템을 좀더 보완 한다면 범용으로 이용이 가능하며, 여러 종류의
과채류의 등급, 선별과 포장의 개발에 활용될 수 있을 것이다. 하나
의 시스템이 여러 종류의 과채류에 활용한다면 더욱더 경제적인 것
이다.

SUMMARY

Currently, Cucumbers are graded and sorted by hand instead of using a machine system. And also they are directly wrapped by hand excepting being wrapped by a manual wrapping equipment. Its development involved the integration of a grading, sorting, and wrapping system with a vision system along with an IBM PC/DX 486 compatible. The grading system was designed and constructed with fluorescent lighting equipment, frame grabber, B/W CCD camera, VGA monitor to grade cucumbers. Software, written in C, combined the functions of image capture, image processing, and control into one program. Image processing algorithms were developed to extract both the geometric and the features of cucumbers. The wrapping system had one cucumber rolled with the limited wrapping paper at regular intervals. An automatic conveyor belt system was needed to move the cucumbers in and out of the working zone to sort and wrap cucumbers.

The integrating system proved to be a reliable system for grading cucumbers.

CONTENTS

Chapter 1. Introduction	11
1. Background	11
2. Necessity	13
Chapter 2. Automatic Grading Decision for Cucumber	15
1. Korean Cucumber Grade	15
2. Japanese Cucumber Grade	15
Chapter 3. Development of Computer Vision System	19
1. Introduction	19
2. Equipment and Material	23
3. Extraction of the Features for Cucumber	23
Chapter 4. Development of Wrapping System	35
1. Introduction	35
2. Equipment and Material	36
3. Operational Methods	36
3. Operational Range of Wrapping System	43
Chapter 5. Development of Dividing System	45

여 백

목 차

제 1 장 서 론	11
제 1 절 연구배경	11
제 2 절 연구의 필요성	13
제 2 장 오이등급 설정	15
제 1 절 한국 오이등급	15
제 2 절 일본 오이등급	15
제 3 장 화상처리 시스템	19
제 1 절 서언	19
제 2 절 장치 및 재료	19
제 3 절 오이 특징점 추출	23
제 4 장 포장 시스템	35
제 1 절 서언	35
제 2 절 장치와 재료	36
제 3 절 작동방법	36
제 4 절 포장장치 적용범위	43

제 5 장 분리장치	45
제 1 절 서언	45
제 2 절 장치와 재료	45
제 3 절 작동방법	48
제 6 장 재료 및 방법	51
제 1 절 실험재료	51
제 2 절 실험방법	51
제 7 장 결과 및 분석	53
제 1 절 오이 패턴인식 및 등급설정	53
제 2 절 포장장치 및 일관작업	61
제 8 장 결론 및 요약	64
참고문헌	67
부록	69

제 1 장 서 론

제 1 절 연구배경

WTO 출범과 더불어 우리나라의 농업이 생존할 수 있는 길은 지속적인 농산물에 대한 생산비 감소와 노동여건의 개선이며, 이를 위해 과거와 같은 노동집약적인 토지 생산성 향상이 아닌 기계화와 자동화등 자본 집약적인 노동 생산성 향상을 위한 농업기술의 도입과 기술개발에 있다. 또한 농산물의 품질향상과 부가가치의 향상을 위해 농산물의 수확후 가공 및 처리의 자동화등 전자공학 기술을 바탕으로 한 새로운 농업기술을 도입하여야 할 것이다.

이에 따라 정부에서도 농업의 기술경쟁력 강화및 취약한 국내 농업을 보호하기 위하여 농어촌 구조개선자금 ('85년 406억, '92년 1조234억, '93년 1조1865억)및 농특세(95년 부터 년1조 5천억 지원예정)를 신설하여 농업에 대한 지원을 강화하고 있다.

그러나 농어촌 시설 근대화 추진사업의 일환인 시설개체 및 자동화 자금이 국내로 유입되지 않고 국내 관련산업의 취약한 구조 및 우리나라실정에 적합한지의 여부도 검증되지 않은채 외국산 자동화 시설재의 무분별한 도입으로 피해 농가가 속출하고 있고 외국기업의 무차별 공략으로 막대한 외화(예 : 93년 양계시설 산업의 경우 3천억 규모의 국내 시장이 외국산에 50%정도 잠식되어 1,500억 유출)가 국외로 빠져 나가고 있는 실정이다.

한편 경제발전과 더불어 국민소득의 향상으로 국민들의 식생활 패턴은

신선한 과일 및 채소 등을 더욱 더 선호하고 있는 추세(채소:1인당 연간 소비량, '65년 45.5kg, '85년98.6kg, '92년 128.9kg)이다.

본 연구의 대상작물인 오이는 사과 다음가는 충주시 및 증원군의 특화품으로서 해마다 생산량이 증가하여 농가 소득에 일조를 담당하고 있는 작물 중의 하나이다.

그러나 오이는 모양 및 규격이 일정하지 않아 이들에 대한 선별 및 등급 판정은 사람의 눈과 손에 의한 가장 단순하고, 반복적인 일관작업의 연속으로 작업자들이 쉽게 싫증을 느끼며, 비능률적이어서 오이의 고품위화, 고부가 가치화에 큰 걸림돌이 되고 있다. 오이의 포장작업 역시 마찬가지로 대부분이 Box 단위 형태로 출하되어 판매되고 있는 실정으로 백화점이나 슈퍼마켓 등에서는 오이의 상품으로서의 생명인 신선도를 유지하고, 저장성을 향상시키기 위하여 다시 이를 Box에서 풀어 하나하나 랩으로 포장, 진열하여 판매하고 있는 형편이다. 따라서 이러한 노동집약적인 전근대적인 방법에서 탈피하여 노동생산성을 향상시키고, 오이의 국제경쟁력을 확보하기 위해서는 선별 및 등급, 포장 자동화를 통한 오이의 표준규격화, 고품위화, 고부가 가치화 및 저장성 향상이 반드시 필요하다.

이와 같은 점에 착안한 본 연구의 구체적 목적은 노동 생산성 향상 및 오이의 고품위화, 고부가 가치화 및 저장성 향상을 위하여 컴퓨터의 화상처리 시스템을 이용한 오이의 자동 선별, 등급 및 포장 시스템을 개발하여 산업화 하는데 있다.

제 2 절 연구의 필요성

1. 기술적 측면

- 오이의 자동선별 장치를 위한 선별 등급의 기술확보에 필요.
- 오이의 화상처리 시스템을 위한 등급 선별기술의 확보에 필요.
- 오이의 자동 등급판정 기술확보에 필요.
- 오이 랩핑 장치 개발에 필요.
- 오이 자동 이송장치 개발에 필요.
- 오이의 화상처리에 의한 패턴인식의 특징점 추출에 필요.
- 농산물 선별 분야에 대한 화상처리 기술 발달에 대한 기여.

2. 경제적 측면

- 자동화로 인한 노동 생산성 향상(5-10배).
- 화상처리에 의한 자동선별로 농산물의 규격화 및 고품위화 달성
- 오이 랩핑화에 의한 노동력 절감(5-10배).
- 오이의 저장성 향상(5-7배).
- 농산물의 규격화 및 고품위화로 인한 농가수익 증대 (30%-50%).
- 오이의 국제경쟁력 확보.

3. 사회적 측면

- 오이등급의 규격화로 인한 구매 및 판매의 합리화.
- 농민의 중노동으로부터 탈피로 인한 보다 나은 문화생활 영위 가능.

농산물 수입개방에 따른 농민의 불안 해소.

농업기술의 첨단화로 농업에 대한 국민들의 고정관념 해소

농가소득의 향상으로 이농현상 억제.

제 2 장 오이의 등급설정

제 1 절 한국 오이등급

한국의 오이등급 규격은 한 상자의 무게가 20Kg, 15Kg, 10Kg을 기준으로 하여 나누어진다. 그 중에서 대표적인 한 상자의 무게가 10Kg인 등급 규격은 표 2-1과 같다. 등급의 기준은 한 상자속의 평균 오이 하나 하나의 무게에 의하여 결정되고 있다. 단, 표 2-1에서 나타난 것과 같은 색깔, 결점과, 최소 오이 길이에 대하여 등급마다 최소한의 기준을 요구하고 있다.

제 2 절 일본 오이등급

일본의 오이등급 규격은 출하(표준) 규격 및 불량품(수출 불가능)구분으로 나누고 있다. 먼저 표준출하 규격은 품위기준, 대소기준, 무게기준, 포장기준 및 기타로 나누며, 순서대로 살펴보면 다음과 같다.

1. 표준출하 규격('93, 8. 16 공포)

가 품위기준

(1) 최저 기준

- 품종 고유의 형상, 색깔 색택을 갖고 전형적일 것.
- 적절히 생육된 것.
- 부패, 변질되지 않는 것.

- 병해 및 충해가 없는 것.

표 2-1 오이의 등급규격 (기준: 10 Kg)

구 분		특	상	중	하
갯수	단과형 계통	50 이내	60 이내	70 이내	80 이내
	장과형 계통	40 이내	45 이내	50 이내	60 이내
무게 (g)	단과형 계통	180 이상	160 이상	140 이상	120 이상
	장과형 계통	250 이상	220 이상	200 이상	180 이상
과일 크기의 균일정도		5 이내	10 이내	15 이내	20 이내
색 갈		품종 고유의 색깔을 갖춘 것			
결점과 (병충해과 상해과 외관불량과)		-	-	5% 이내	10% 이내
기타 참고사항		다른 품종, 부패, 변질된 것이 없어야 한다. 단과형계통: 과장 15cm 이상인 것 장과형계통: 과장 25cm 이상인 것			

(2) 등급 기준 (표2-2 참조)

표 2-2 오이의 등급기준

A	<ul style="list-style-type: none"> · 굽은 정도가 2Cm이내인것. · 양끝부분이너무 굵거나 가늘지 않는 것. · 상해가 없는 것.
B	<ul style="list-style-type: none"> · 굽은 정도가 4Cm이내인것. · 양끝부분이 굵거나 가늘기의 정도가 경미한 것. · 상해의 정도가 경미한 것.

나 대소기준 (표 2-3 참조)

대소구분은 1개의 길이 또는 중량에 의하되 1개의 중량에 대

해서는 상·하 각각 5%정도의 차이를 허용한다.

표 2-3 오이의 대소기준

등급	계급	1개의 길이	1개의 중량	1용기 중의 개수(참고)
A	L	22Cm이상25Cm미만	110g이상140g미만	42 - 47
	M	19Cm이상22Cm미만	80g이상110g미만	50 - 54
B	L	22Cm이상25Cm미만	110g이상140g미만	-
	M	19Cm이상22Cm미만	80g이상110g미만	-

다 최저기준을 충족시키는 것으로서 등급 구분 또는 대소기준에 적합하지 않은 것의 호칭은 보통이라 한다.

라 무게기준

1포장 단위의 무게는 시장 판매 시에는 5Kg으로 한다.

마 포장기준

포장용지는 골판지 상자로서 그 기준은 다음과 같다.

*컨테이너식:덮개가 없는 상자 몇단을 쌓아올려 제일 윗상자에 덮개를 씌운다.

- 불량품(대일수출 불가능품) 구분
- 현재 우리나라에서 일본으로 수출하고 있는 오이의 경우 일본 바이어의 요구에 따라 그 규격을 정하고는 있으나 대부분 아래에서 제시된 기준을 크게 벗어나지 않는다.

표 2-4 오이의 포장기준

	A식	컨테이너식
용기의 크기	길이(용기안쪽)420mm 폭 (용기안쪽)250mm 높이(용기안쪽)95mm	길이(용기안쪽)465mm 폭 (용기안쪽)370mm 높이(용기안쪽)70mm
재 질	재질은JIS Z1506외장용 양면 골판지상자 제2종 이상, 최대 내압 강도는 5kg, 용기는350kg 이상.	
포장방법	봉합은 폭 2mm이상의 STAPLE로 상부, 하부 각 2개소 이상으로 함. JIS1511포장용지 GUM테잎 제1종 이상 또는 동등 이상의 효력을 가진 자재 사용도 무방.	
표시사항 (외장)	품목명 · 품위구분 · 대소구분·중량 · 출하자명 또는 상표 · 산지.	

기타는 최저기준을 충족시키는 것으로 등급구분 또는 대소기준에 적합하지 않은 것의 호칭은 보통이라고 한다. 이 연구에서는 일본의 등급기준을 채택하여 정량화된 등급판정을 하였다. 단 일관작업에서는 사용자가 임의로 등급기준을 설정할 수 있도록 프로그램을 개발하였다.

제 3 장 화상처리 시스템

제 1 절 서언

오이는 모양 및 규격이 일정하지 않아 이들에 대한 선별 및 등급 판정은 사람의 눈과 손에 의한 가장 단순하고, 반복적인 일관 작업의 연속으로 작업자들이 쉽게 싫증을 느끼며, 비능률적이어서 오이의 고품위화, 고부가가치화에 큰 걸림돌이 되고 있다. 아직까지 오이에 대하여 자동 선별이 가능한 이미지 프로세싱에 대한 연구는 없었으며, 일관작업을 위하여 화상처리에 관한 연구를 이 과제에서 먼저 시도하였다. 특히, 과채류 중에서 오이가 가장 긴 과채류중의 하나이기 때문에 다른 과채류와는 시스템이 다른 것을 개발하였다.

제 2 절 장치 및 재료

그림 3-1과 같은 화상처리 시스템을 설계제작하였으며, 화상처리시스템에서 조명장치는 무엇보다도 중요하다. 조명장치가 잘못 설계되면 정확한 화상처리 알고리즘을 얻기란 불가능하다. 오이의 길이가 최대 45 cm 경우까지 측정하기 위해서는 카메라 초점부터 오이까지의 거리가 150 cm 정도 필요하였다. 이에 따라 처음에는 아주 큰 조명장치를 만들었으나 이럴 경우 전체 시스템의 크기가 너무 커져 실용화 하기에는 문제점이 있다고 판단되어 조명장치의 크기를 줄일 목적으로 거울을 달았다. 거울을 45° 방향으로 기울게 장치한 후 오이와 화상 이미지를 90°로

바꾼 후에 카메라에서 이 잔상 이미지를 얻었다(그림 3-2). 이와 같이 거울을 부착할 경우 조명장치의 길이를 1/2를 까지 줄일 수 있었다. 조명장치는 30 W의 형광등을 그림 3-3과 같이 카메라 둘레에 배치하였다.

프로파일(20mm x 20mm)로 시스템을 제작하였고, 그 안의 형광등을 양철로 고정하였으며, 또한 외부의 광을 차단하기 위하여 검은 캔트지를 부착하였다. 그 결과 조명장치는 외부의 영향을 어느 정도 배제할 수 있었다. 이 장치의 설치공간을 가능한 줄이기 위하여 가로로 세우지 않고 세로로 눕혀 설계하였다. 컨베이어 벨트로 이송되는 오이의 동적인 화상 이미지를 포착하기 위하여 화상처리시스템 아래쪽을 컨베이어 벨트 위 쪽으로 하였다.

설계제작된 조명장치의 크기는 세로, 가로 길이가 115.2cm, 34.4 cm이다. 이 조명장치의 가장 큰 특징은 거울을 이용하여 시스템의 크기를 줄여 설계하였다는 점으로 거울의 각도를 45° 방향으로 기울여 빛의 직진성을 최대한으로 이용하였다. 거울 사용은 하나로 하였는데 이는 2 개 이상 사용할 경우 외부의 다른 배경들이 카메라에서 이미지로 얻어지기 때문이다.

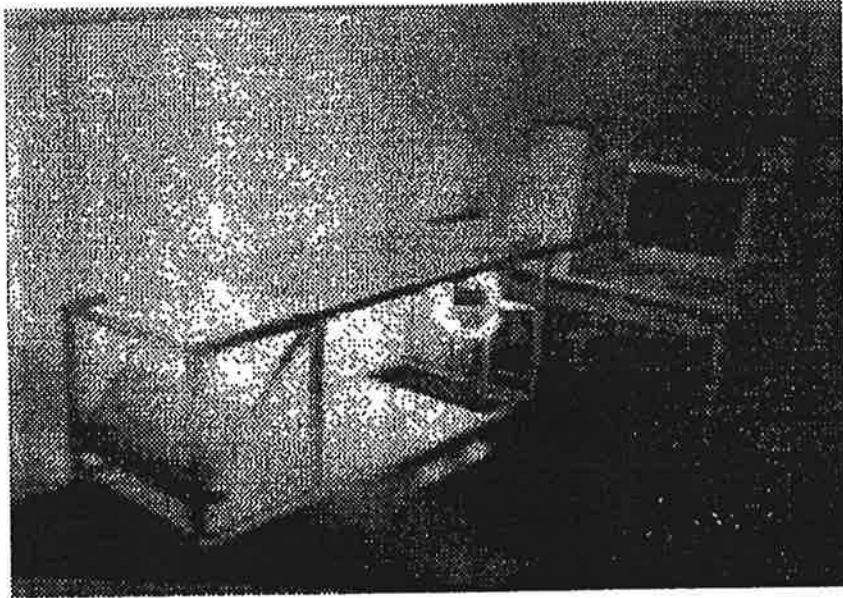


그림 3-1. 화상처리 시스템

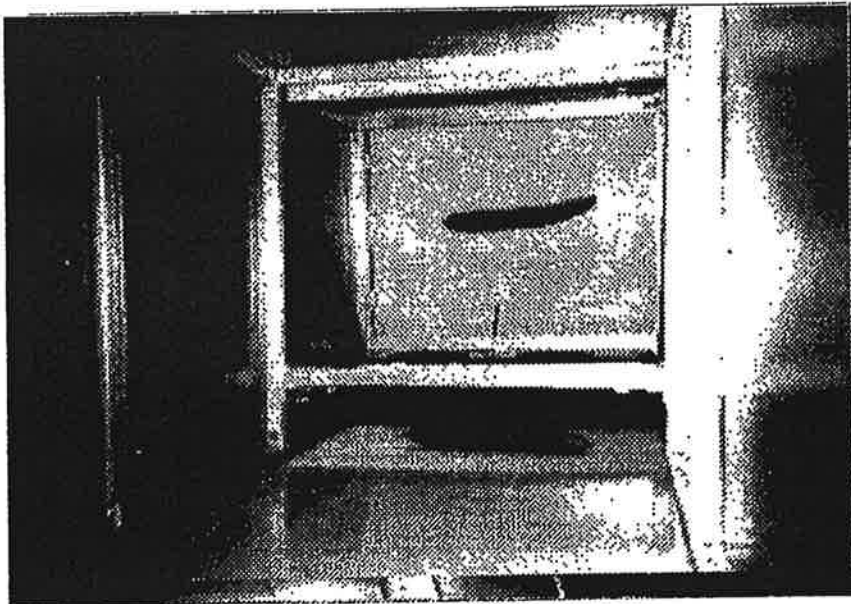


그림 3-2. 컨베이어 벨트 위에 있는 화상처리 시스템

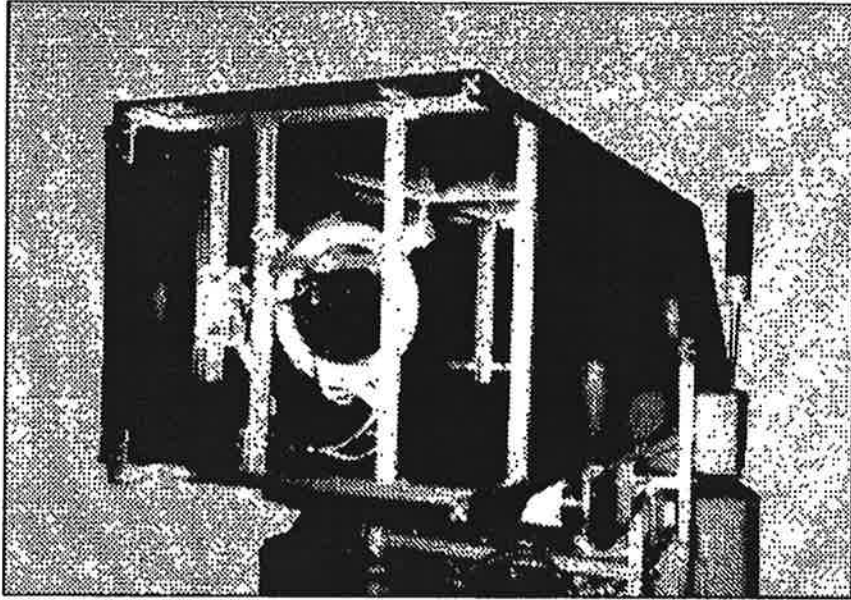


그림 3-3. 45° 기울어진 거울을 가진 조명장치

제 3 절 오이패턴 인식과 특징점 추출

1. 드레쉬홀드(Threshold)

일반적으로 대상물체는 다치의 그레이(Gray) 화상으로서 입력되는데 적용목적에 따라서 경계인식 및 패턴추출 등의 화상해석을 용이하게 하기 위하여, 또한 필요로 하는 연산시간의 최소화를 위하여 흑, 백의 2치 화상으로 변환 시키게 된다. 이를 화상의 2치화 또는 2진 처리라고 한다.

화상의 2치화는 입력화상을 $F=f(i,j)$ 출력화상을 $G=g(i,j)$ 로 할때 드레쉬홀드 값을 이용하여 다음과 같이 수행된다. 여기서 i 와 j 는 화상을 이루는 화소(Pixel) 좌표를 나타내며 f 와 g 는 상응하는 그레이 레벨(Gray level)을 뜻한다.

$$g(i,j) = 255 : f(i,j) \geq t$$
$$0 : f(i,j) < t$$

t 는 드레쉬홀드 값으로서 배경부분(흰색 : 255)과 추출할 대상체(흑색 : 0)를 분리하는 값이다. t 값의 올바른 선정은 선별을 위한 특징점의 추출에 매우 중요한 영향을 미친다.

이러한 t 값을 선정하는 방법에는 여러가지가 있다. 히스토그램의 모드(Mode)에 따른 분리도를 이용하는 방법, 통계적 기법을 이용하는 방법, 모멘트 보존법, 화상의 엔트로피(Entropy)를 이용하는 방법, 배경 윈도우를 확장하는 방법 등이 있다. 여기서는 윈도우 확장법과 히스토그램의

분리도를 이용하는 방법에 대해 간략히 설명하고 이들이 갖는 문제점을 해결할 수 있는 혼합형태에 의한 효과적 2치화 알고리즘을 기술한다.

대상체의 배경 영역을 설정하는 윈도우의 범위를 변환시켜 드레쉬홀드 값을 추출하여 화상을 2치화하는 알고리즘으로 다음과 같다.

1. 대상물을 포함하는 윈도우를 설정하고, 윈도우내의 화상 그레이 레벨 히스토그램 $A[i]$ ($0 \leq i \leq 255$)를 구한다. 여기서 $A[i]$ 는 i 의 그레이 레벨 갖는 화소의 수 또는 확률을 나타낸다.
2. 윈도우를 적당한 크기로 확장하여 다시 히스토그램 $B[i]$ 를 구한다.
3. 단지 배경 영역만이 확장 되었으므로 대상체의 히스토그램분포는 변화가 없다. 그러므로 $A[i]$ 와 $B[i]$ 를 비교하여 드레쉬홀드값 t 를 구한다. t 는 그레이 레벨 i 의 값을 0에서 순차적으로 증가시키면서 $A[i]$ 와 $B[i]$ 의 히스토그램 분포를 비교함으로써 추출한다.
4. 경계값 t 를 기준으로 하여 원화상을 2치화 한다.

2. 체인코드

자동으로 2치화된 2치 화상으로부터 대상물의 각 등급의 특징이 될만한 정보를 추출해 내기 위하여 Freeman의 체인 코딩을 사용 하였다. 우선 2치화된 화상 (배경은 흰색(Gray level 255),대상물은 검은색(Gray level 0)을 Raster Scan 하면서 처음으로 만나는 검은 화소(Pixel)를 시작점으로 하여 그 다음 연결점(검은화소)을 찾아 나간다. 그 다음 연결점은 8 방향중 한방향을 찾아서 연결이 되는데 그림 3-4는 8방향의 고유 벡터(Vector)번호를 나타내며 그림 3-5 는 탐색 방향에 따라서 조건에 맞는

연결점을 찾는 방법을 나타내었다. 이때 첫 시작점의 X, Y좌표는 기억 장소에 기억이 되며 항상 시작점과 비교하며 다음 연결점을 찾아 나가며 연결점에 해당하는 화소의 정보(X,Y좌표, 체인(Chain)번호, 체인방향)도 매번 기억 장소에 기억이 된다. 다음 연결점의 X,Y좌표와 시작점의 X,Y좌표가 일치하게 되면 체인 코딩은 끝나게 된다. 그림 3-6은 임의의 도형을 체인 코딩하는 간단한 예를 나타내었다.

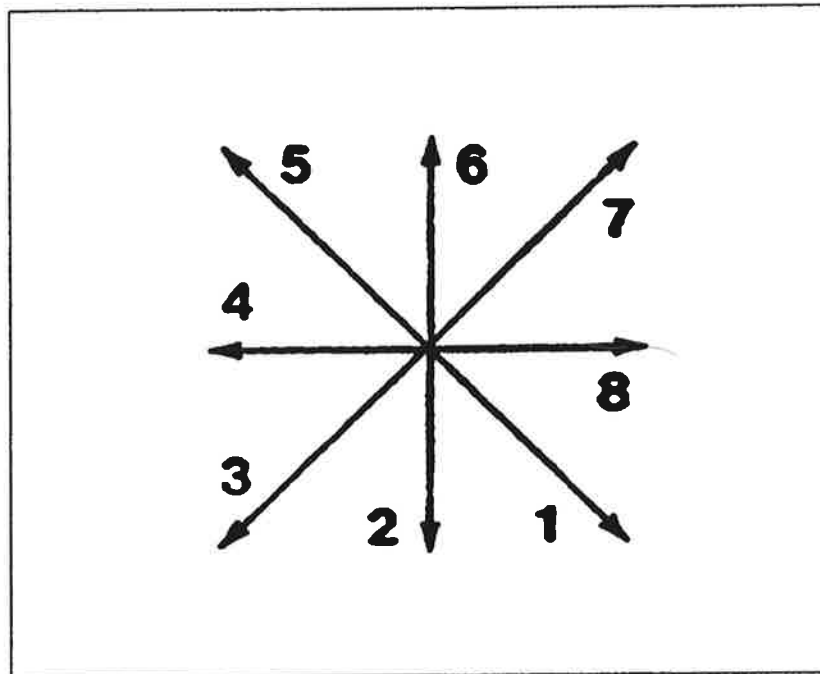


그림 3-4 8 가지 방향의 벡터

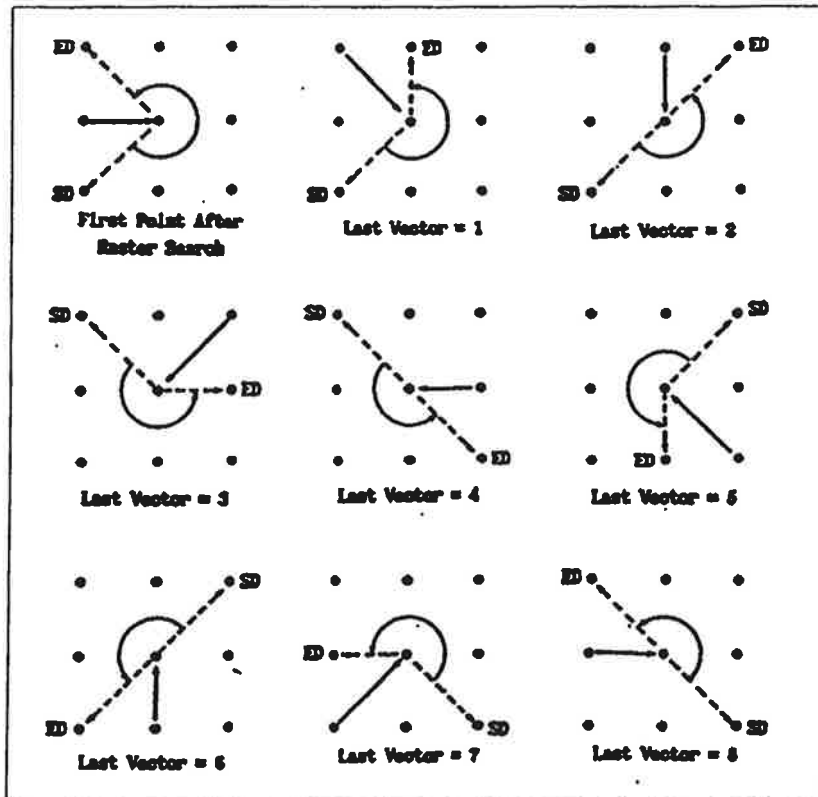


그림 3-5. 체인 코딩에 의한 방향탐지

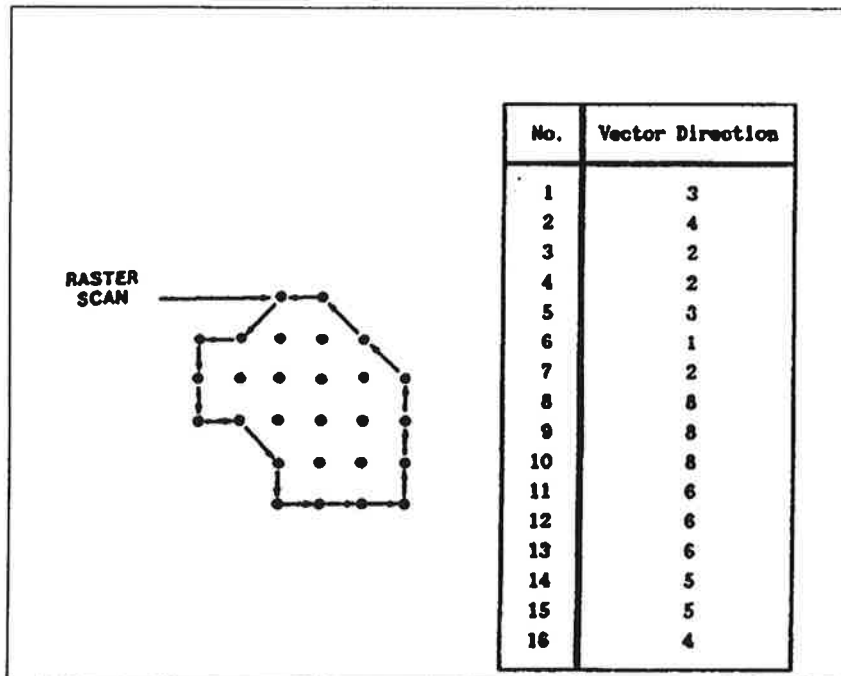


그림 3-6 체인 코딩의 예

체인 코딩을 끝낸 후 면적, 둘레의 길이, 최대/최소 반경비, 평균 반경, 도심점(Centroid), 원형도 및 복잡도 등의 정보가 얻어진다. 윤곽의 연결이 불완전한 화상의 체인 코딩은 체인 코딩을 수행 시키기 위한 전제 조건으로 대상물의 연결성을 들 수 있다. 대상물의 화상 윤곽선이 반드시 폐곡선을 이루고 있어야만 체인 코딩을 수행 시킬수 있다. 윤곽선을 이루는 점들을 찾아내는 방법이 윤곽선을 구성하는 점들 모두가 연결되어 있어야만 가능하고 대상물의 둘레, 면적, 도심점 등을 구하는 방법은 대상물 화상의 좌표와 체인 코딩에 의해 수행되기 때문이다. 그러나 대상물의 화상을 2치화 하는 과정에서 연결성이 결여되는 경우가 발생 할 수

있다. 그러나 앞에서 언급한 체인 코딩을 그대로 적용하는 경우 작은 화소 하나 만이라도 연결이 되어 있지 않으면 그 다음의 윤곽선 구성점을 찾지 못한다. 이러한 경우에 대비하여 3개 정도의 화소가 연결되어 있지 않더라도 그 다음 연결점을 찾아나갈 수 있도록 했다. 체인 코딩의 기본 알고리즘은 3 X 3 Mask를 이용하였다. 그러나 이 Mask를 7 X 7, 9 X 9로 점차 확장해 간다면 세개의 화소점이 떨어진 경우까지 연결이 가능 하다.

3. 특징점 추출

처음 만나는 화소에서부터 체인 코딩을 마칠 때 까지의 화소의 전체합 이 둘레의 갯수가 된다. 그런데 화소를 이루는 Grid는 그림 3-7과 같이 정사각형이 아니므로 X, Y 방향의 길이가 다르다. 8 방향 체인 코드에서 수평,수직 그리고 대각선 요소를 구분하여 스케일 값을 주어야만 실측 둘레를 얻게 된다.

짝수번의 벡터 성분중에서 수평화소에 해당하는 4, 8 방향의 총 갯수 를 HEVEN이라 하고 수직 화소에 해당하는 2, 6 방향의 총 갯수를 VEVEN이라 하자. 그리고 나머지 홀수번인 1, 3, 5, 7 방향은 대각선 요소에 해당하는 ODD라하고 수평, 수직 그리고 대각선 방향의 스케일 값을 SCX, SCY,SCD라 하면 실측둘레는 다음과 같다.

$$\text{둘레}=[(\text{HEVEN} \times \text{SCX})+(\text{VEVEN} \times \text{SCY})+(\text{ODD} \times \text{SCD})] \times \text{CONY}$$

여기서 $SCD = (SCX^2 + SCY^2)^{1/2}$

HEVEN : 수평방향 화소의 총 갯수

VEVEN : 수직방향 화소의 총갯수

ODD : 대각선방향 화소의 총갯수

CONY : 실측길이 변환 계수

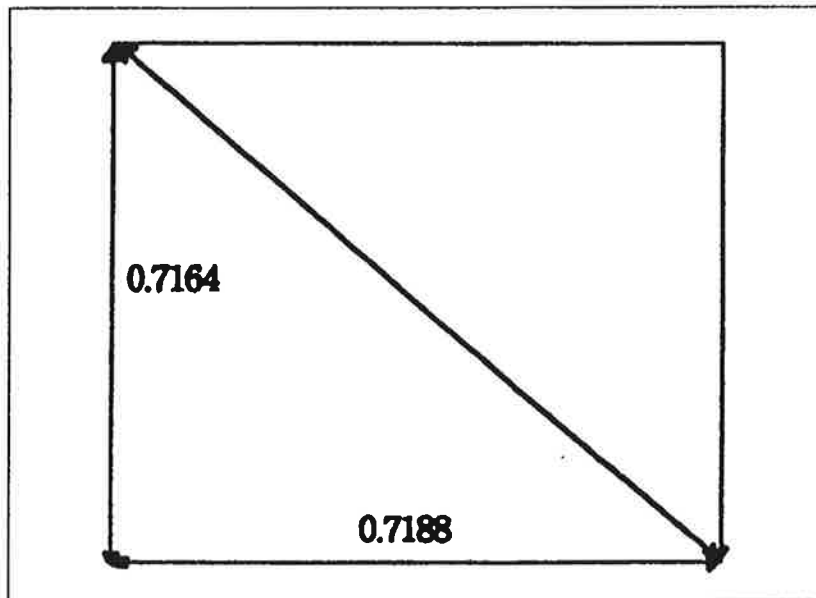


그림 3-7 화소 크기

추출된 윤곽에 의해 둘러 쌓인 물체의 면적은 8개의 Element Vector
와 "Y = 0 또는 X = 0"인 라인을 기준선으로 하여 이 기준선과 대상체
의 윤곽선에 둘러쌓인 면적의 합에 의해 측정된다.

그림 3-8에서의 8방향 부호 관계에 의해 1, 7, 8번 방향, 즉 Search Direction이 $X=0$ 라인에 대해 갖은 모멘트 방향이 반시계 방향 (C.C.W) 인 경우는 양(+)의 면적을 나타내고 3, 4, 5번 방향의 Search Direction은 시계방향(C.W)이므로 dma(-)의 면적을 나타낸다. 따라서 순면적(Net Area)은 각 방향이 가지고 있는 면적의 전체합으로서 나타내어 진다.

실측 화상의 면적을 구하기 위해 화소 캘리브레이션에 의해 구해진 가로, 세로 화소의 단위길이 SCX, SCY를 이용하여 각 방향 벡터의 면적은 그림 3-8을 참고로 하여 구하며, 표 3-1에 그 결과를 실었다.

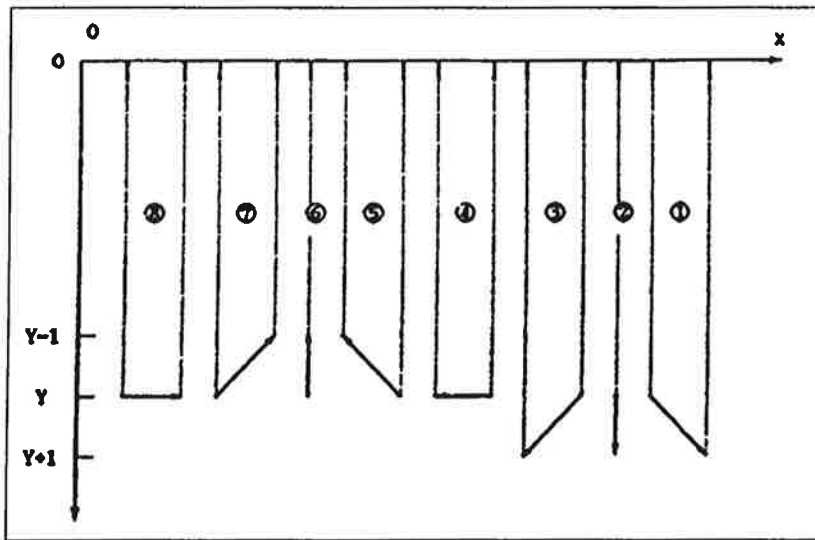


그림 3-8 각 방향의 면적

표 3-1 각 방향의 면적을 구하는 식.

VECTOR DIRECTION	AREA (X-SCALE : SCX, Y-SCALE:SCY)
8	SCX · SCY · y
7	SCX · (SCY · y-0.5)
6	0
5	-SCX · (SCY · y-0.5)
4	-SCX · SCY · y
3	-SCX · (SCY · y+0.5)
2	0
1	SCX · (SCY · y+0.5)

X=0, Y=0 인 라인에 대해서 둘러싸인 면적의 1차 모멘트는 도심점을 산출하기 위하여 쓰여진다. 1차 모멘트를 얻는 식은 다음과 같다.

$$M = \int_0^y y \cdot dA$$

1. X에대한 1차 모멘트

그림 3-9 에 유도 근거를 나타내었으며 표 3-2에 그 식들의 결과만을 실었다.

2. Y에 대한 1차 모멘트

그림 3-10 에 유도 근거를 나타내었으며 표 3-3에 그들의 결과값들을 실었다.

3. 실측 도심점 산출

체인을 모두 들고난 후의 면적의 전체합과 X와 Y에 대한 모멘트
 의 전체합을 가지고 화면상의 물체의 도심을 구할수 있다. 도심을
 구하는 식은 다음과 같다.

$$X \text{ Centroid} = \frac{\sum(My)}{\sum area} \quad , \quad Y \text{ Centroid} = \frac{\sum(Mx)}{\sum area}$$

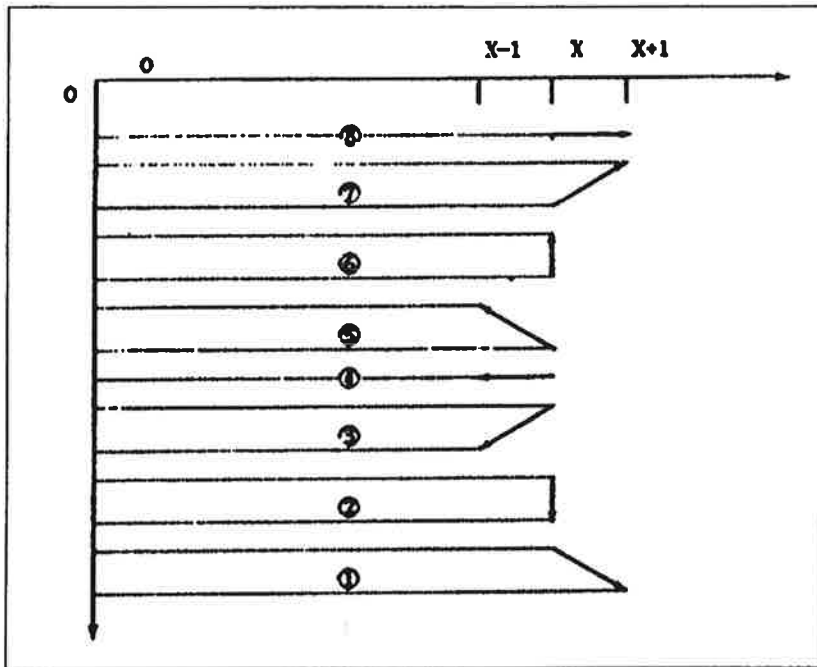


그림 3-9 X에 대한 1차 모멘트.

표 3-2 각 방향에 대한 모멘트를 구하는 식.

VECTOR DIRECTION	MOMENT OF X-DIRECTION
8	$SCX \cdot yn^2 / 2$
7	$SCX \cdot (yn^2 - 2yn + 1) + SCX/SCY \cdot (0.5yn - 1/3)$
6	0
5	$-SCX \cdot (yn^2 - 2yn + 1) - SCX/SCY \cdot (0.5yn + 1/3)$
4	$-SCX \cdot yn^2 / 2$
3	$-SCX \cdot yn^2 / 2 - SCX/SCY \cdot (3yn + 1) / 6$
2	0
1	$SCX \cdot yn^2 / 2 + SCX/SCY \cdot (3yn + 1) / 6$

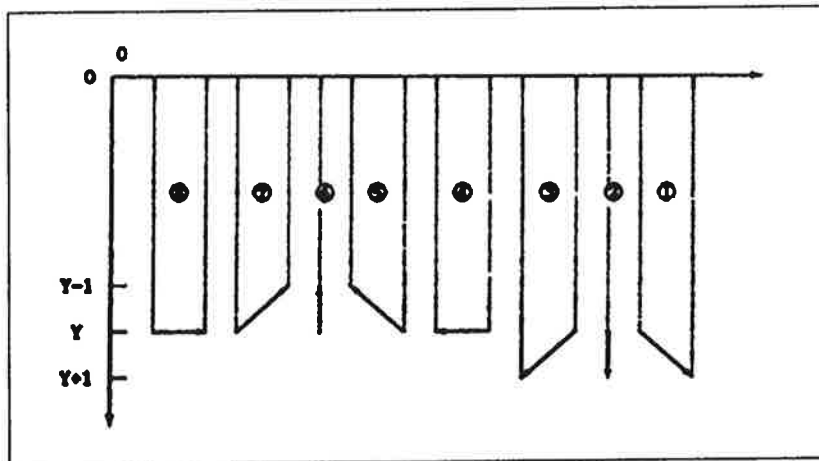


그림 3-10 Y에 대한 1차 모멘트.

표 3-3 각 방향에 대한 Y 모멘트 구하는 식

VECTOR DIRECTION	MOMENT OF Y-DIRECTION
8	0
7	$SCY \cdot xn^2/2 + SCY/SCX \cdot (0.5xn+1/6)$
6	$SCY \cdot xn^2/2$
5	$SCY \cdot (xn^2-2xn+1)/2 + SCY/SCX \cdot (0.5xn-1/3)$
4	0
3	$-SCY \cdot (xn^2 -2xn+1)/2 - SCY/SCX \cdot (0.5xn-1/3)$
2	$-SCY \cdot xn^2 / 2$
1	$-SCY \cdot xn^2 / 2 + SCY/SCX \cdot (0.5xn+1/6)$

제 4 장 포장 시스템

제 1 절 서언

오이 포장장치는 오이를 랩(Wrap)으로 낱개 포장하는 소형 자동포장 장치이다. 랩은 신선도를 오래 유지할 수 있도록 수분 유출은 방지하고 공기는 통하도록 하는 기능을 가지고 있을 뿐만 아니라, 가격도 저렴하고 오이 하나를 포장하는데 소요되는 랩의 양도 매우 작다. 따라서 오이를 산지에서 선별하여 쉽게 포장할 수 있는 본 장치를 사용하였을 경우, 기대되는 효과는 백화점 및 소매점에서 오이를 판매할 때 소요되는 시간으로 인하여 조기 부패되는 것을 방지하고 유통 시간을 1주일 정도 연장시킬 뿐만 아니라 운반 도중 예상되는 오이 표피의 손상을 방지하여 유통 중의 손실을 줄일 수 있다.

기존의 포장장치는 범용으로써 스티로폴로 정형화된 그릇을 만들어 이 안쪽에 각종 과일류를 넣고 그릇체 포장하게 되어 있어 스티로폴 그릇으로 인한 환경오염 및 포장비용 증가 등의 문제점이 있고, 또한 여러 개를 한꺼번에 포장함으로써 소비자가 원하는 수량을 선택하여 구입할 수 없다는 문제점이 있다. 하지만 본 연구에서 개발한 포장시스템은 손으로 일일이 포장하는 것과 동일한 효과를 발휘할 수 있는 정교한 자동 포장장치이다.

제 2 절 장치와 재료

오이 자동 포장장치는 랩을 오이 둘레에 싸기 위하여 크게 두 개의 장치로 구분되어 있다. 하나는 랩을 일정한 크기로 절단하는 장치이며, 다른 하나는 잘라진 랩으로 오이를 둘러싸는 장치이다. 포장용지 절단장치는 랩을 잡아걸어 주는 환봉, 감아 줄 랩, 랩 받침 로올러를 위한 실리콘 띠, 처음 랩을 감아 올리기 위한 길이 5mm 의 바늘 18개가 박힌 롤러, 랩 절단을 위한 가열판, 랩을 잡아 주기 위한 환봉 축, 환봉 축의 왕복 운동을 도와주는 모터로 구성되어 있다 (그림 4-1). 한 개의 오이를 감아 주는 장치는 장치 부착용 프레임, 고무 롤러를 돌려주는 감속 모터, 수평 왕복 운동을 위한 가이드, 핵심 장치 모듈을 회전시켜 주는 감속 모터, 랩이 겹쳐져 오이에 달라붙게 하기 위한 가이드 등으로 구성되어 있다.

제 3 절 작동방법

그림 4-2에 표시된 (12)는 시중에서 흔히 구할 수 있는 롤형태로 말린, 폭 45 Cm 의 랩 이고 (11)은 원기둥 형태의 환봉으로써 오이 한 개를 포장할 수 있는 만큼의 랩((14)지점에 걸쳐있는 랩을 끌어당기고 또한 포장 순서에서 일정한 인장력으로 잡아 당겨 주는 기능을 하는 장치이다. 그리고 (15)는 (11)이 오이를 랩 으로 감싸 안은 후 오이를 회전시켜 오이를 감싸안고 남은 여분의 랩을 오이에 밀착시켜 오이가 견고하게 랩핑 될 수 있도록 하는 기능을 한다.

그림 4-3은 본 포장장치의 작동 원리 와 순서를 알아보기 쉽도록 과정 별로 나열하여 그린 것이다. A 상태는 초기 상태이고 B 는(11)은 x1 방향 (18) 은 y1 방향으로 약간 올라간 상태를 보여주고 있다. C 는 회전 가능한 포장 기의 모듈이 연결된 감속 모터의 회전으로 CCW 방향으로 약 a° 만큼 회전한 모습이다.

D 는 (11) 은 x2 방향 (18) 은 y2 방향으로 1 Cm 정도 밀어진 상태를 보여준다 이는 (14)에 걸쳐있는 랩(Wrap)을 잡아당기기 위하여 (18)이 접촉하는 상황이다. 이때 (18)의 상부에 박힌 20개정도의 바늘이 랩을 잡아당기게 된다. E 는 모듈이 CW 방향으로 회전하면서 (18) 이 랩(17)을 잡아당겨 모듈로 끌어당기는 모습입니다. 이때 랩이 감겨져 있는 롤 (12) 는 회전하면서 풀리게 되다.

F 모듈이 E 지정보다 CW 방향으로 180° 더회전하게 되면 랩 커팅 (cutting)을 위하여 장치된 가열판(20)이 랩에 자동으로 닿게 되고 또한 한 개의 오이를 포장할 수 있는 너비의 랩(23)이 모듈의 (11)과 (18) 사이

에 걸쳐지게 된다. 이때 (18)에는 랩이 (18)에 박혀 있는 바늘에 의해 걸쳐지고 (11)에는 (22)부위에 환봉을 따라 붙여진 실리콘의 접착력으로 랩이 걸쳐지게 된다.

G 는 전기 히터(heater) 에 의해 가열된(100 °C) 열판 (20) 에 의하여 팔려 오던 랩이 커팅되는 모습이다. 이때 모듈은 계속해서 CW 방향으로 회전을 한다. H 는 모듈이 G 에서보다 CW 방향으로 90° 더 회전한 상태이다. I 는 (11)은 x3 방향으로 (18)은 y3 방향으로 2 Cm 정도 수평한 상태로 이는 (11)과(18) 사이에 걸쳐있는 랩이 아래로 약간 처지게 하여 오이가 x4 방향으로 투입될 때 안정성을 확보하기 위함이다. J 는 모듈에 오이(24) 가 올려진 상태이다. K 는 (18)은 x5 (11) 은 y5 방향으로 이동 랩이 오이를 감싸안으면서 고무 롤러 (15)가 오이를 y6 방향으로 압착한 상태이다. L 은 롤러 (15)가 오이 (24)를 압착한 상태에서 감속 모터(25) 가 x6 방향으로 회전 여기에 연결된 롤러 (15)가 역시 x6 방향으로 회전 하면서 랩에 감싸인 오이 (24)가 같이 회전하면서 (11) 과 (18) 에 걸쳐있던 랩이 오이 쪽으로 빨려 들고 이때 (18)과 (11) 도 랩이 오이 쪽으로 빨려가는 방향 즉 (18)은 x7 (11)은 x8 방향으로 랩이 다 빠져나갈 때까지 회전하게 됩니다. 이와 같이 롤러 (15)는 3회전한다. M 이후 (18)은 x9 방향으로 (11)은 y9 방향으로 이동하고 오이는 완전히 포장된 상태가 된다. N 은 모듈이 CCW 방향으로 180° 회전하여 오이가 x10 방향으로 떨어지기 바로 전 상태이다. O 는 Wrapping 된 오이가 밖으로 배출되는 상태입니다. 이후 다시 처음 상태인 A 상태로 돌아가 반복하게 된다.

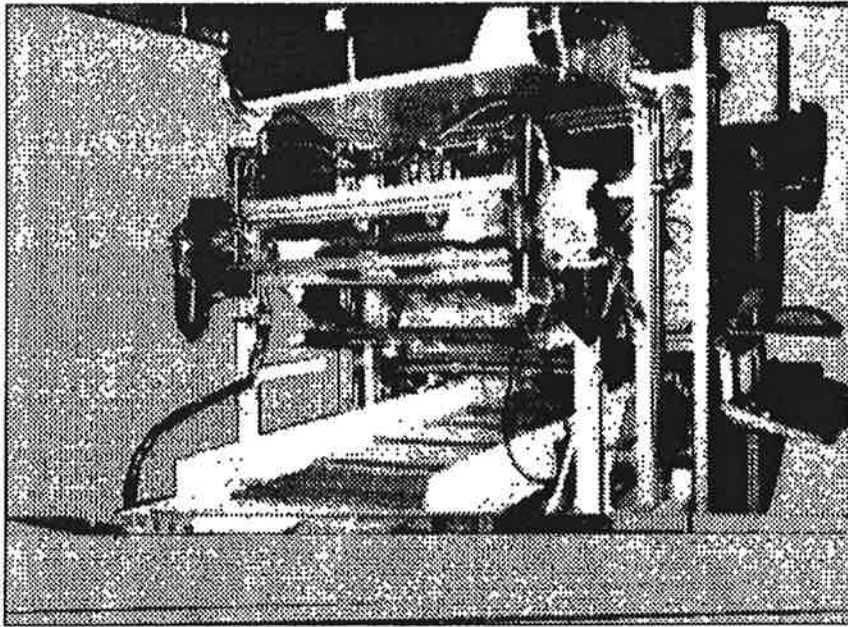
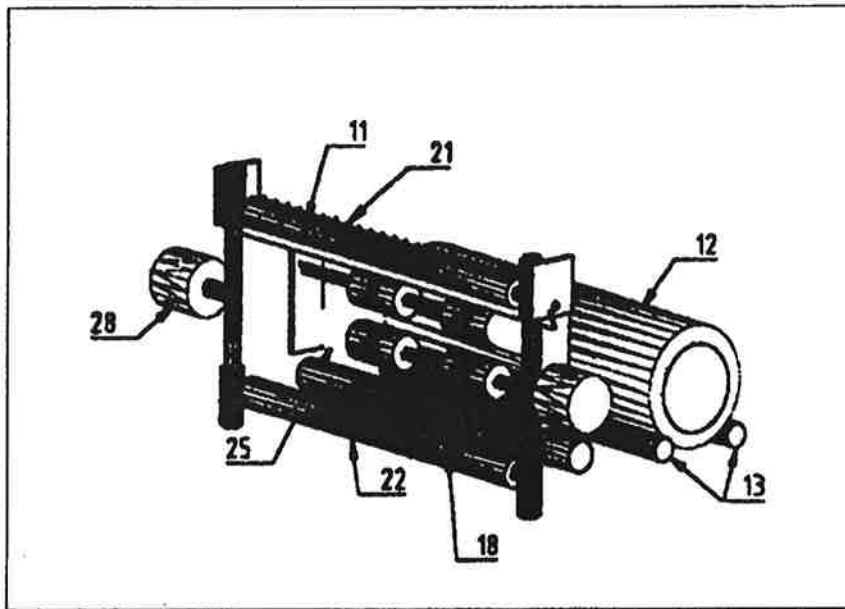
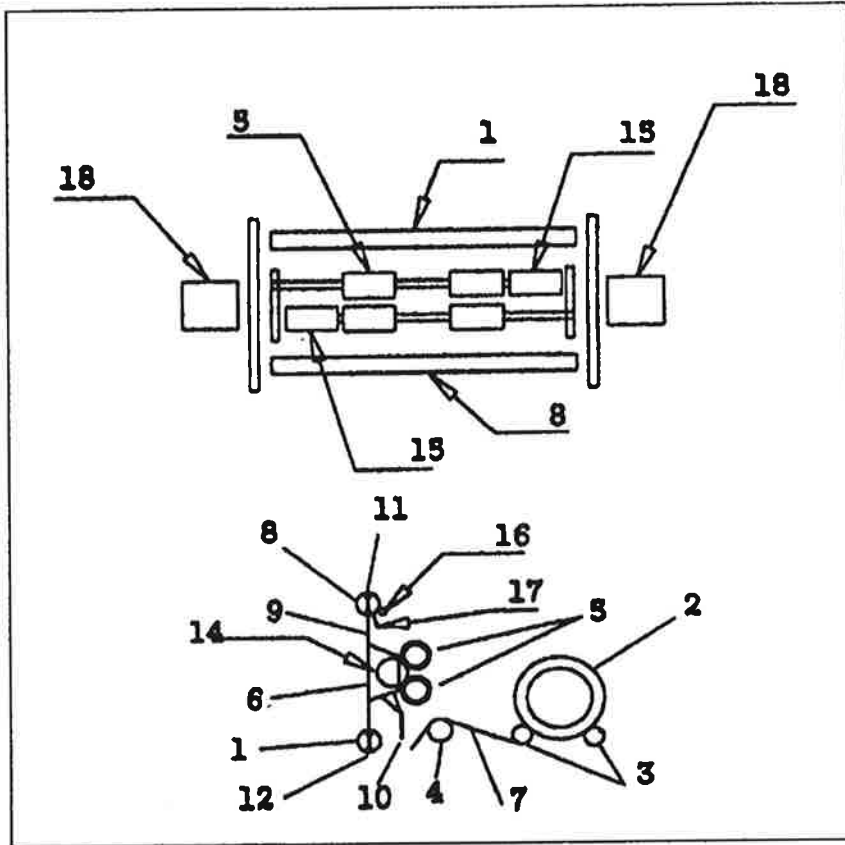


그림 4-1. 분리장치 컨베이어 위에 있는 포장장치



(A) 입체도



(B) 평면도와 측면도

- | | |
|-----------------------|-----------------------|
| 1 랩(wrap) 을 잡아걸어주는 환봉 | 11 처음 랩을 잡아 올리기 위한 길이 |
| 2 랩 | 5mm 의 바늘 18개가 박힌 것 |
| 3 14 랩 받침 로울러 | 12 환봉에 랩이 걸려 있게 하기 위한 |
| 실리콘 띠 | |
| 5 고무 로울러 | 13 오이 한 개 포장할 수 있을 량 |
| 의 랩 | |
| 6 장치 부착용 프레임 | 14 오이 |
| 7 랩 한접 | 15 고무 롤러 (15)를 돌려주는 감 |
| 속 모터 | 16 오이 회전을 원활하게 하여주 |
| 8 랩을 잡아주기위한 환봉 | 17 랩이 겹쳐져 오이에 달라붙게 |
| 는 축 | |
| 9 수평 왕복 운동을 위한 가이드 | 18 핵심 장치 모듈을 회전시켜 주 |
| 하기 위한 가이드 | |
| 10 랩 절단을 위한 가열판 | |
| 는 감속 모터 | |

그림 4-2. 포장장치의 대략도

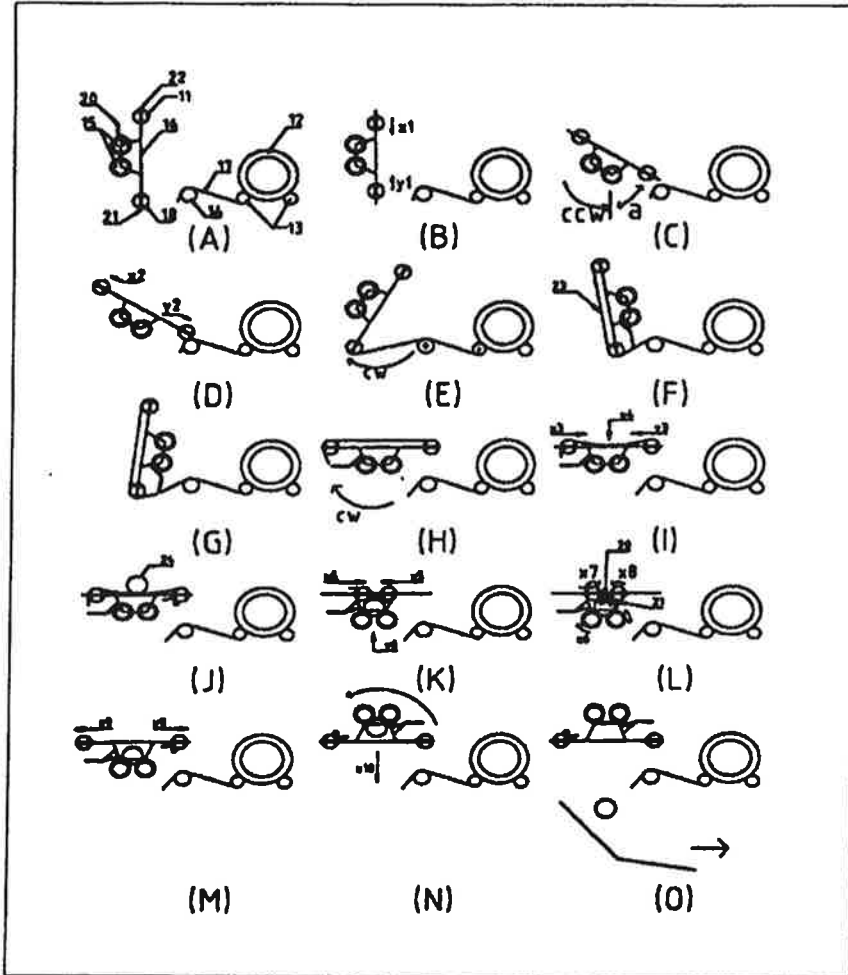


그림 4-3. 오이 포장장치의 작동 원리도

제 4 절 포장장치 적용범위

첫째는 랩이 걸칠 수 있는 칩(21)이 박혀 있는 환봉 (18)을 이용 랩 (Wrap)을 잡아당겨 오이 한 개 포장할 수 있을 정도의 길이로 열판(20)을 이용 커팅 하고 (11)과 (18) 이 오이를 향해 수평 이동하여 랩(wrap)으로 오이를 감싸안고 그 반대편에서 오이에 밀착시킨 고무 롤리(15)를 회전시켜 오이를 Wrapping 할 수 있게 한 장치이며, 둘째는 환봉(18)이 (14)축에 걸친 랩을 잡아당기기 위하여 모듈((11) (16) (18) (21) (15) (20) 으로구성된) 이 회전하는 구조로된 포장 장치이며, 셋째는 모듈이 회전하면서 고정된 열판에 의해 랩(wrap)이 자동으로 커팅되도록 한 구조를 한 포장 장치이다.

오이는 수분 함유량이 많아 쉽게 시들고 손상이 가기 쉽다. 따라서 랩으로 포장을 할 경우 오이 표피 손상을 방지하고 수분 유출을 막아 신선도를 오래 유지할 수 있다. 일부 백화점에서 판매하는 오이가 랩으로 포장되어 있기는 하나 일일이 손으로 오이를 랩으로 싸거나 신축성 있는 길다란 봉지에 집어넣는 형태이다. 따라서 손이 많이 가고 능률도 낮다. 하지만 본 발명품은 랩을 일정한 간격으로 자른 후 잘린 랩으로 오이를 감싸 팽팽히 밀착시켜 감아 주는 작업을 컴퓨터 프로그램으로 제어 동작하도록 되어 있다. 따라서 단순한 작업을 자동화 하여 작업 속도는 물론 소요 인력도 절감할 수 있다.

본 장치의 구성은 랩을 잡아 당겨 일정한 간격으로 자를 수 있게 길다란 환봉을 회전 가능한 모듈에 부착시켜 회전시키는 형태이다. 이렇게 구성함으로써 일정 너비의 랩을 간단한 장치로 쉽게 자를수 있고 또한 연속해

서 다음 공정인 오이를 받아 랩으로 감싼 다음 회전 공정으로 들어가 물러가 오이를 회전시켜 랩이 오이 주위로 말려지게 된다. 즉 일련의 공정이 소형의 한 장치 내에서 이루어진다.

제 5 장 분리장치

제 1 절 서언

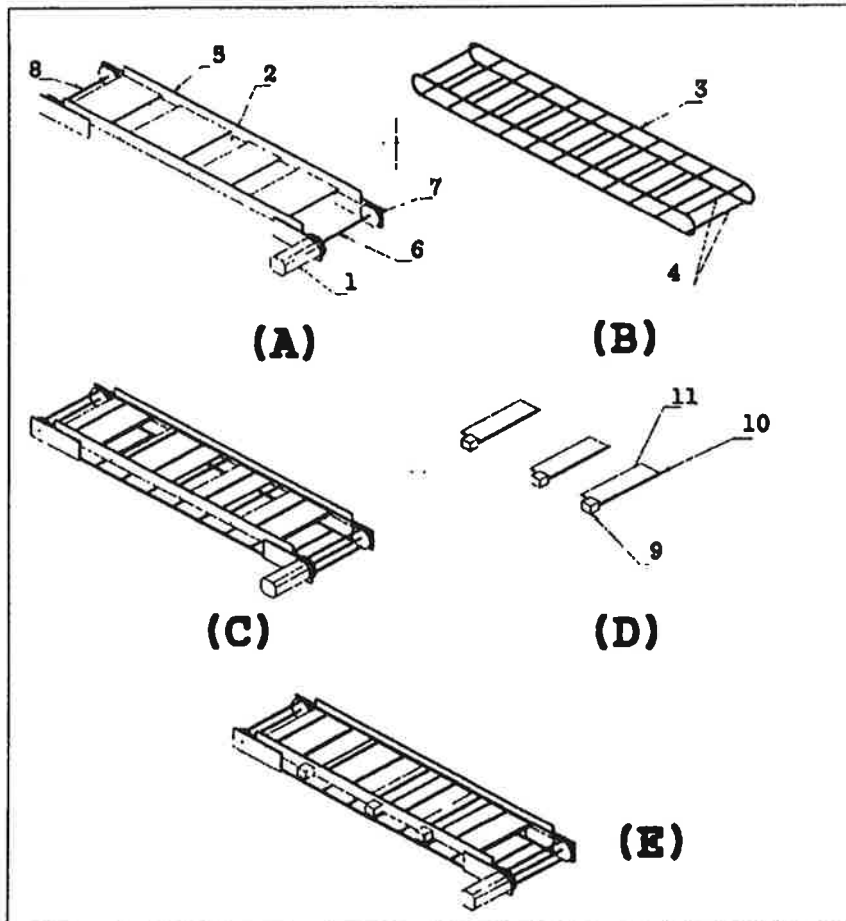
화상처리로 오이의 등급을 판정한 후 이 오이를 등급이 결정된 상자 속으로 이동하는 것이 필요하다. 등급이 결정된 오이를 분리상자에 넣기 위한 분리장치를 설계제작하였다. 분리장치의 시스템은 오이의 이송을 위하여 일정한 간격으로 폴리우레탄 호스로 구성된 컨베이어 벨트를 그림 5-1과 같이 배치하였다. 이송된 오이를 요구되는 상자에 넣기 위하여 개폐장치를 제작하였으며, 모든 제어는 컴퓨터에서 이루어진다. 이 연구에서는 오이의 등급에 따라 4개의 상자를 배치하였다. 물론 더 많은 등급이 요구되면 쉽게 제작변경이 가능하도록 설계하였다.

제 2 절 장치와 재료

그림 5-1과 같은 오이분리장치는 오이를 밀어서 이송시키는 체인형태의 콘베이어와 이 콘베이어를 구동시켜주는 모터, 그리고 등급에 맞추어 원하는 위치에 배출시키기 위한 문과 그 문을 구동시키는 솔레노이드로 구성되어 있다. 이 분리장치의 재료는 먼저 오이를 밀어서 이동시키기 위한 기구로 환봉이 가로로 연결된 체인형태의 회전 기구와, 다른 하나는 그 체인을 회전시키는 구동축과 풀리, 그리고 반대편에 설치된 중동 풀리와 중동축으로 구성되어 있다.

그 외의 재료로써 ABS 수지판으로 구성된 본체와 원하는 위치에 오

이를 낙하시키기 위한 문을 구동시키기 위한 솔레노이드, 그리고 솔레노이드를 구동시키기 위한 SSR(SOLID STATE RELAY)로 이루어져 있다. 컴퓨터에서 보내는 신호를 받아 SSR를 구동시켜주는 I/O 보드가 이 장치에 인터페이스되어 있다.



- | | | |
|------------|----------------------|---------------|
| 1. 감속모터 | 5. ABS 수지로 구성된 골격프레임 | 9. 문을 열기위한 모터 |
| 2. 오이배출구멍 | 6. 구동축 | 10. 문에 고정된축 |
| 3. 나일론호수 | 7. 풀리 | 11. 문 |
| 4. 폴리우레탄 붐 | 8. 중동축 | |

그림 5-1. 분리장치의 구성도

제 3 절 분리장치 작동방법

오이를 등급에 따라 상자로 분리하기 위한 적당한 장치로 구조를 설계 제작하였고 연속적으로 투입되는 오이에 대하여서도 분리가 가능하도록 하였다. 본 장치의 구성은 그림 5-1과 같으며, (A) 와 (B) 가 결합하여 (C) 형태가 되며 (C)와 (D) 가 결합하여 (E) 의 형태가 된다. 여기서 (A) 는 전동기와 중심 프레임을 표시한 것이고 (B)는 (A)에 걸쳐서 회전하는 체인으로서 (3)은 나이론 호수로 (4)는 폴리우레탄으로 구성되어 유연하게 휘어질수 있다. 먼저 (1)은 오이를 가이드(5) 위로 이송시키는 동력원인 감속모터이다. 이 모타가 체인 형식으로 구성된 (B)를 풀리(7)를 회전시킴으로써 끌어당기게 되어있다. 축(6)은 풀리(7)과 (8)이 같은 속도로 돌수 있도록 고정되어 있으며 이는 전동기(1)에 고정되어 회전하게 되어 있다. (11)은 (2) 와같이 뚫여있는 구멍에 설치될 문으로써 오이가 상판을 지나갈 때 감속모터(9)가 회전함으로써 문 (11)을 축 (10) 를 회전축으로 CCW 방향으로 회전시켜 문(11)이 아래로 열리도록 하는 구조로 되어 있다. 이렇게 함으로써 오이가 본 컨베이어 아래에 등급에 따라 놓여지는 상자로 떨어지는 것이다. 그림 5-2는 분리장치의 평면도와 정면도이며, 이 분리장치의 크기의 기본 치수가 나타나 있다. 그림 5-2는 오이를 등급에 따라 상자에 떨어뜨리기 위해 문이 열린상태의 그림이다. 본 장치의 배출구는 총 4개으로써 4가지의 등급대로 배출할수 있으며 이 후 확장이 필요할 경우에는 체인(B)와 프레임 (5)를 변경하고 문 (D) 를 늘려주면 얼마든지 등급간격을 늘릴수 있다.

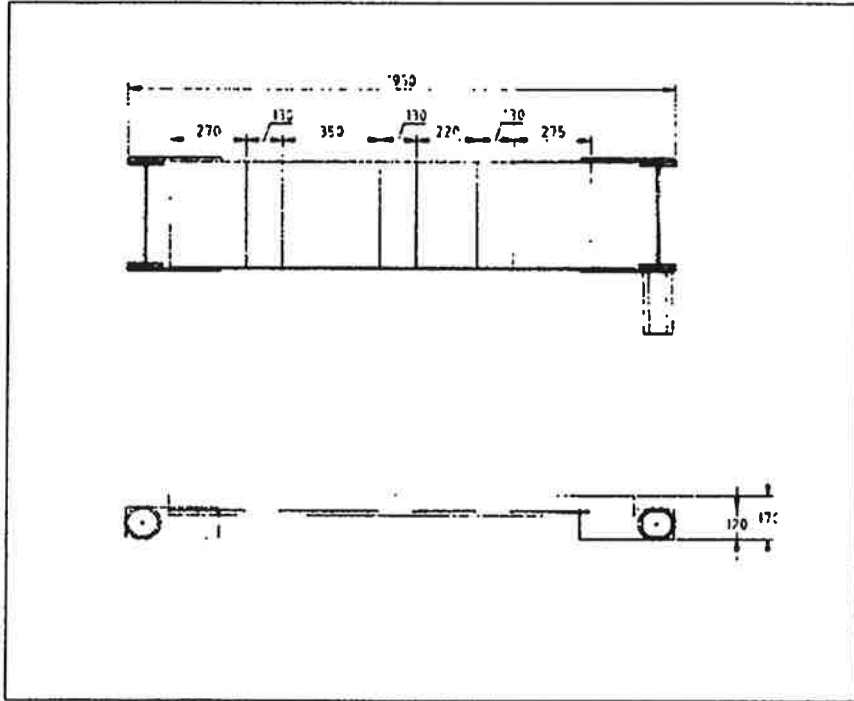


그림 5-2. 분리장치의 평면도와 정면도

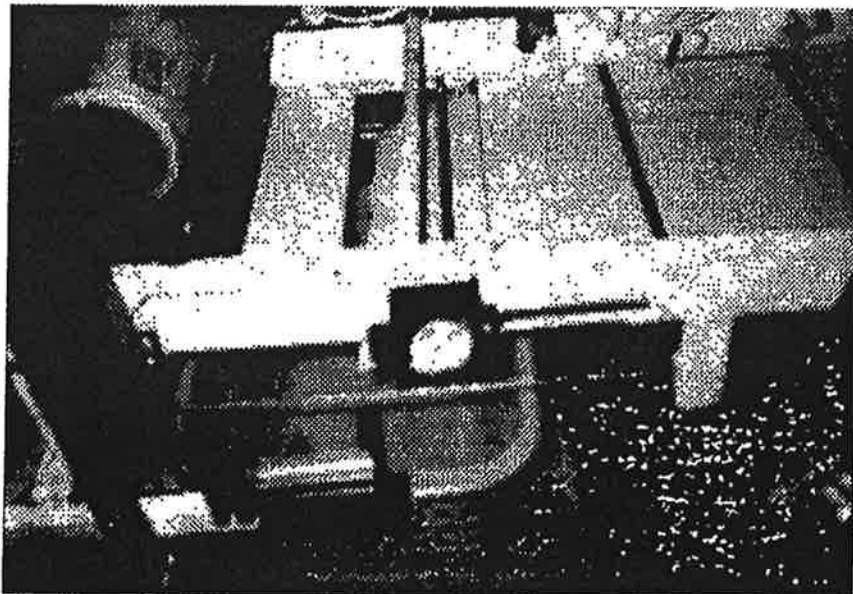


그림 5-3. 개폐장치가 열린 분리장치

제 6 장 재료 및 방법

제 1 절 실험재료

3개의 오이품종 즉 장과형 계통인 백다다기와 단과형 계통인 조선오이와 청오이를 시중에서 구입하였으며, 이를 이용하여 실험결과를 얻었다. 그림 6-1에 나타난 오이의 품종명은 청오이, 조선오이, 백다다기오이의 순서대로 나타내었다. 임의로 각 품종마다 굵어진 각도의 정도에 따라 품종당 3개의 오이를 사용하였다.

포장기의 성능 과 일관작업의 원활성을 측정하기 위하여 단과형 청오이를 무작위 채취한 샘플 100 개를 이용 반복 실험을 실시하였다.

제 2 절 실험방법

먼저 화상처리를 이용하여 등급설정을 구현할 수 있는 지 여부를 실험하였으며 여기서 오이의 특징점 추출로 모멘트, 도심, 복잡도, 원형도의 데이터를 얻었고, 등급을 결정하기 위하여 면적, 둘레, 오이의 장축길이, 굵어진 각도를 이용하여 오이의 등급을 결정하였다.

포장작업과 일관작업의 작업정도를 규명하기 위하여 실험을 실시하였다. 포장작업은 형태가 다른 여러 종류의 오이를 개별로 화상처리를 이용한 등급설정과 동시에 포장의 능률정도를 실험하였다. 또한 포장가능한 오이만을 선별하여 연속적으로 포장기에 투입시켜 10분동안 몇 개가 포

장가능한지를 반복실험을 통해서 알아보았다. 또한 포장실험과 동시에 등급별로 분리장치의 작동실험도 실시하였다.

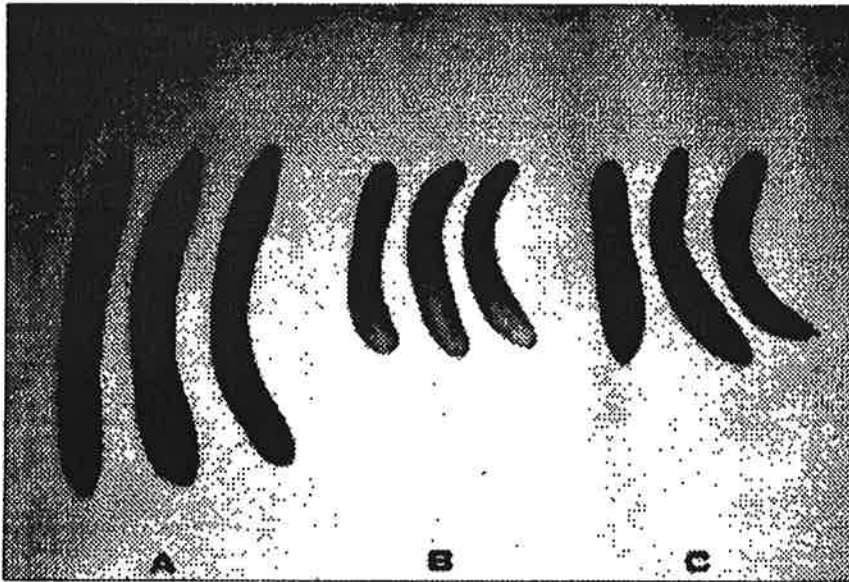


그림 6-1 등급 설정용 시료로 사용된 오이

제 7 장 결과 및 분석

제 1 절 오이 패턴인식 및 등급 결정

한국에서 사용되고 있는 등급 판정은 표 7-1에서와 같이 크기, 무게 그리고 크기가 단과형 15 cm 이상일 것, 장과형 25 cm 이상일 것을 요구하고 있으며, 또한 다른 품종, 부패, 변질된 것이 없어야 한다는 등급 기준을 정하고 있다. 이는 기준 설정에 있어서 정량화되어 있지 않아 이에 따라 등급 판정을 하기에는 애매모호 하였다. 즉, 등급 판정이 좋다고 하여 반드시 좋은 품질의 오이라고 할 수 없다. 왜냐하면 굵어진 각도와 균일한 크기가 더 주요한 요소이기 때문이다. 따라서 체계적이고 정량화된 일본의 등급기준을 이 연구의 오이 등급의 기준으로 결정하였다.

등급 결정을 하기 위하여 패턴인식 알고리즘의 프로그램을 개발하였다. 그림 7-1에서는 이 과정을 설명하여 주는 흐름도이다. 그림 7-2에서 보는 바와 같이 먼저 드레쉬홀드 값을 결정한 후 특징점을 추출하여 면적, 오이의 (장축)길이, 크기 굵어진 정도 등을 결정한 후 오이의 등급을 결정하는 알고리즘을 개발하였다.

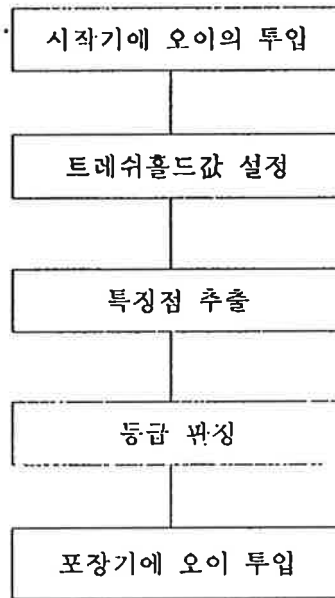


그림 7-1 오이 등급판정 흐름도

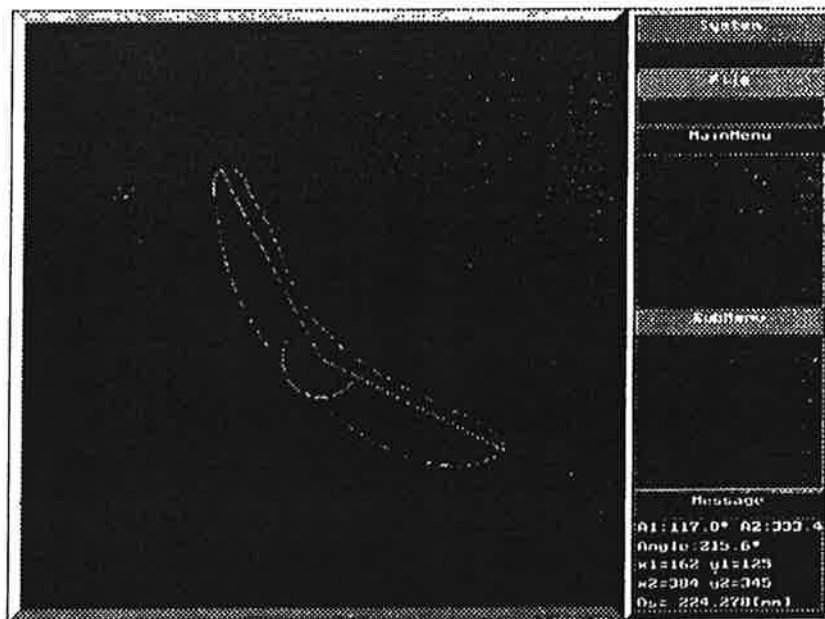


그림 7-2 체인 코딩에 의한 오이 인식

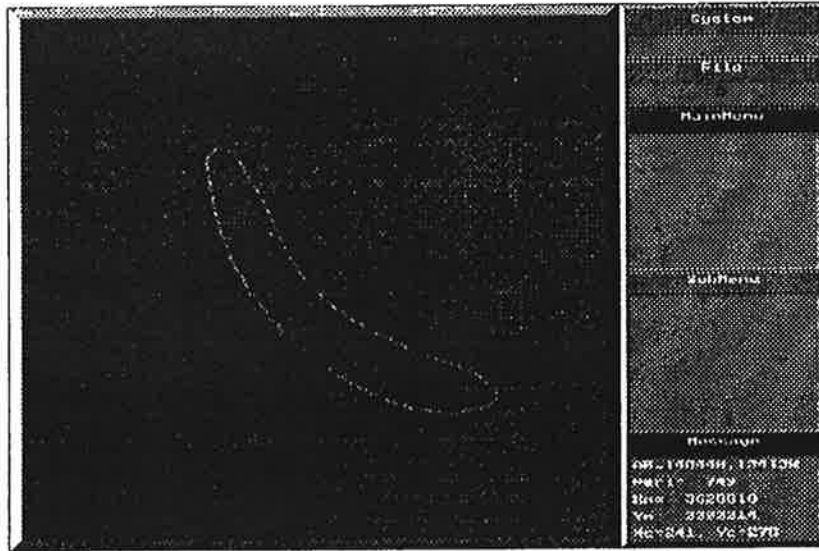


그림 7-3 오이의 도심을 이용한 굵은 각도

오이의 등급을 판정하기 위하여 먼저 하나의 화소 값의 크기를 결정하였으며, 이를 사용하여 드레쉬홀드 값을 구하기 위하여 히스토그램의 알고리즘을 개발하였다. 그후 Chain Code를 사용하여 오이의 특징점을 추출하였다. 체인코드를 통하여 모멘트, 도심, 복잡도, 원형도를 얻었으며, 또한 등급을 결정하기 위하여 면적, 둘레, 오이의 장축 길이, 굽어진 각도를 이용하여 오이의 등급을 결정하였다.

오이의 등급을 결정하기 위하여 이용한 각 알고리즘을 순서대로 설명하고자 한다. 먼저 화소의 크기를 알기 위하여 60 x 60 cm² 모눈종이를 이용하여 10 cm 간격으로 화소의 수가 몇개 들어 가는지를 5회 반복하여 측정하였다. 표 7-1에서 보는 바와 같이 가로(X)축의 하나의 화소치

의 실제 길이는 0.719 mm이었고, 세로(Y)축의 길이는 0.716이었다. 표준편차와 분산값이 아주 작게 나타는데 이는 화소 하나의 오차가 거의 없다는 것을 의미한다.

표 7-1 오이의 화상처리를 위한 한 화소길이

	평균값	표준편차(STD)	분산(VAR)
X 축	0.719	0.012	1.4×10^{-4}
Y 축	0.714	0.007	4.9×10^{-5}

드레쉬홀드 값을 구하기 위하여 히스토그램 알고리즘을 개발하였으며, 이 히스토그램의 그림은 그림 7-3에 나타나 있으며, 이 그림은 메뉴판에 나타난 것이며, 이 히스토그램의 그림에서 드레쉬홀드 값은 화소값이 50에서 150사이 이다. 높게 나타난 것이 오이이며 낮게 나타난 값이 배경을 나타낸다. 높은 화소치와 낮은 화소치의 경계값을 '드레쉬홀드 값' 이라고 한다. 이 값을 알기 위하여 히스토그램을 이용할 수 있지만, 실제로는 이 값을 구하기 위해서 프로그램의 알고리즘을 통해서 구하였다. 오이의 색깔의 변화가 심한 경우에는 부분적인 윈도우를 만든후 히스토그램을 이용하여 드레쉬홀드 값을 구하였다.

특징점을 추출하기 위하여 체인코드를 이용하여 모멘트, 도심, 복잡도, 원형도를 얻었으며, 등급을 결정하기 위하여 면적, 둘레(그림 7-2), 오이의 장축 길이, 굽어진 각도(그림 7-3)을 이용하여 오이의 등급을 결정하

었다. 사용자가 쉽게 이 프로그램을 이용할 수 있도록 메뉴판을 만들었다. 메뉴판은 System, File, Mainmenu, Submenu, Message 로 구분된다. 여기서 메뉴판은 데이터 정리와 사용자가 편리하게 사용하기 위하여 개발하였다.

기본적인 특징점인 모멘트, 도심, 복잡도, 원형도 추출을 구하였으며, 여기서 3가지 품종을 이용하였으며, 품종마다 3개의 오이(그림 6-1 참조)를 이용한 결과 값이다. 등급을 결정하기 위하여 면적, 둘레, 오이의 장축 길이, 굽어진 각도를 이용하여 오이의 등급을 결정한 것을 표 7-3에 나타내었다. 여기서 굽은 정도는 다음과 같은 식으로 나타낼 수 있다.

$$L = \frac{Ds/2}{\tan(\text{ngle}/2)} (mm)$$

여기서 L : 굽은정도

Ds : 장축의 길이

ngle : 각도

표 7-2 오이의 특징점 추출 (모멘트, 도심, 복잡도, 원형도)

품종		모멘트		도심		복잡도 K	원형도 R
		X _m	Y _m	X _c	Y _c		
청오이	1	3323643	2914474	228.2	274.4	39.2382	0.3204
	2	3710076	3713659	253	268	38.673	0.3248
	3	3205676	3482733	244	236.4	30.1102	0.4172
조선오이	1	3534198	3492160	240.2	242.6	38.5646	0.3258
	2	2710439	3432009	288.2	218	43.7092	0.2876
	3	3280323	3658911	272.4	243	39.7676	0.316
백다다기	1	55623333	5982902	276.8	246	55.9112	0.225
	2	5656812	6490284	273.8	233	49.3706	0.2548
	3	5058072	5471582	260.2	237	52.8056	0.2382

표 7-3 등급을 결정하기 위한 특징적인 요소

품 종		면 적	굽은정도	각 도	장축의길이	등급판정
		A	L	Angle	Ds	J
청오이	1	6571.959	38.66	135.58	189.0622	B M
	2	7540.822	39.63	140.4	219.753	B M
	3	7340.675	14.65	163.8	204.7054	A M
조선오이	1	7464.1	22.31	157.5	223.4664	B M
	2	6116.043	21.84	155.6	201.3946	B M
	3	69911.133	25.19	153.68	214.823	B M
백다다기	1	11122.21	34.67	156.02	325.358	B M
	2	12183.55	27.47	161.56	336.921	B M
	3	10803.63	11.58	172.5	342.598	A M

청오이 1과 2는 오이의 길이는 A 등급이지만 3.87cm, 3.96cm로 2cm보다 크고 4cm보다 작기 때문에 B등급으로 간주하였다. 그외, 조선오이 1,2,3과 백다다기 1,2는 굽은 정도가 2cm보다 크기때문에 B등급이다. 일 본의 등급 기준은 굽은 정도가 오이 등급결정에 큰영향을 미치는 것으로 나타났다.

제 2 절 포장장치 및 일관작업

시작기의 투입구에 오이를 놓으면 화상처리 시스템으로 오이가 지나가면서 특징점을 추출한 후 오이의 등급을 판정한다. 그 후 랩핑장치로 흘러가면 오이를 랩으로 포장한다. 포장된 오이는 등급별로 출구로 방출된다. 이 시스템의 전체 구조는 그림 7-4와 같다. 포장장치 및 일관작업의 성능 분석의 결과는 표 7-4에 나타나 있다. 등급이 AL, AM, BM 인 경우에는 포장작업 실험을 하였을 때 100% 완전 포장이 되었지만 BM은 91.9%로 100% 포장을 할 수 없었으며, 등의 X는 거의 포장이 불가능하였다. 실험한 자세한 자료를 부록에 실었다. 등의 오이를 제외한 포장 가능한 오이를 이용하여 10분간 포장되는 개수를 조사하였는데 포장된 개수는 표 7-5에 나타나 있으며, 평균 포장개수는 83.3개로 나타났다. 제작된 시스템은 직접 무게를 측정할 수 있는 장치를 설치하지 않았다. 그래서 무게를 화상처리의 오이 면적을 이용하여 측정하였다. 그 결과는 그림 7-5에 나타나 있다. 여기서 $y = 0.0313x - 70.244$ 의 회귀방정식을 얻었으며, y 는 무게(단위 g)을 x 는 면적(단위 mm^2)을 나타낸다.

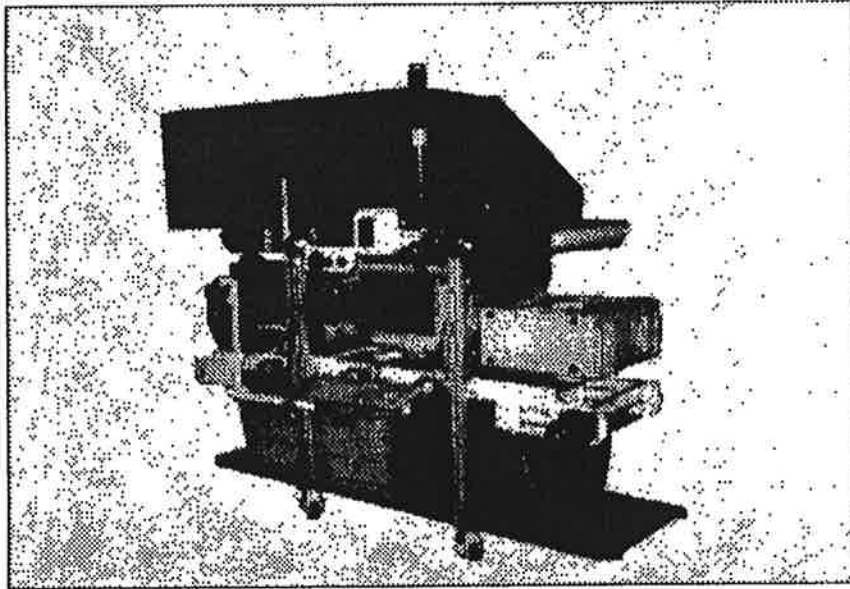


그림 7-4 전체시스템의 구조

표 7-4 일관작업의 성능 분석

작업 등급	총 개수	포장된 개수	포 장율 (%)	분리작업 개수	분리 성능 (%)	일관작업 개수	일관작업 성능 (%)
AL	24	24	100	24	100	24	100
AM	8	8	100	8	100	8	100
BL	37	34	91.9	37	100	34	91.9
BM	2	2	100	2	100	2	100
X	29	1	3.44	29	100	29	100

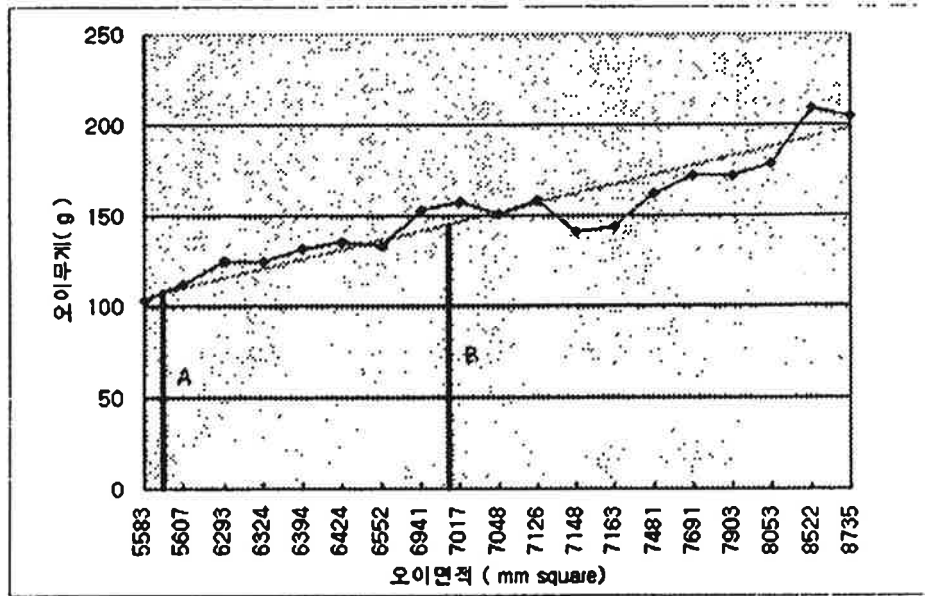


그림 7-5 화상처리의 오이 면적과 무게의 상관관계

표 7-5 10분간 포장가능한 오이의 개수

실험횟수	1회	2회	3회	4회	5회	평균
포장된 오이의 갯수	85	82	83	83	86	83.8

제 8 장 결론 및 요약

전체 시스템은 화상처리장치, 포장장치, 분리장치 등으로 구성되어 있다. 먼저 오이를 투입하면 화상처리 장치에서 오이의 등급을 선별한다. 선별된 오이는 화상처리장치의 빙벨트 컨베이어로 이송되어 포장장치에 도달하게 된다. 포장장치에서는 선별된 오이의 등급이 동의인 것을 제외하고는 모두 포장한다. 포장된 오이는 분리장치의 흰봉 컨베이어 벨트를 통하여 등급별로 등급상자에 이송된다. 이 전체 시스템의 크기는 가로 200 cm 세로 70 cm 높이 173 cm 이다.

1. 등급 선별시스템

등급 선별시스템은 카메라, 화상처리 프레임 그래버 (Frame grabber), 컴퓨터, VGA 모니터, 조명장치로 구성되어 있다. 이 시스템에서 조명장치는 무엇보다도 중요하다. 조명장치가 잘못 설계되면 정확한 화상처리 알고리즘을 얻기란 거의 불가능하다. 오이의 길이가 45 cm일 경우까지 측정하기 위해서는 카메라 초점부터 오이까지의 거리가 150 cm 정도까지 필요하였다. 이를 위하여 처음에는 아주 큰 조명장치를 만들었으나 이럴 경우 전체 시스템의 크기가 너무 커져 실용화하기에는 문제점이 있다고 판단되어 조명장치의 크기를 줄일 목적으로 거울을 달았다. 거울을 45° 방향으로 기울게 장치한 후 오이와 이미지를 90°로 바꾼 후에 카메라에서 이 이미지를 얻었다. 조명장치는 30 W의 형광등을 카메라 둘레에 배치하였다. 오이를 이미지 프로세싱하는 동안의 조명장치

는 형광등 불빛을 바깥에서 볼 수 없도록 검정 쉼트지로 둘러쌌다. 프로파일로 조명장치를 제작하였고, 그 안의 형광등을 양철로 고정하였으며, 또한 외부의 광을 차단하기 위하여 임시로 검정 쉼트지를 부착하였다. 그 결과 조명장치는 외부 조명의 영향을 어느 정도 배제할 수 있었다. 또한, 이 장치의 설치 공간을 가능한 줄이기 위하여 가로로 세우지 않고 세로로 놓혀 설계·제작하였다.

이 등급 선별시스템에서 오이의 등급선별에 따른 오이 분리작업은 실험 결과 성능에 따르면 문제점은 없었지만, 한국에서는 오이등급이 세분화되어 있지 않아 좀더 세분화된 표준화가 필요한 실정이었다.

2 포장작업 및 일관작업 시스템

포장장치는 오이의 크기나 굵기와는 관계없이 휘어진 안쪽 각도가 145°도 보다 작으면 모두 포장가능한 것으로 나타나고 있다. 또한 오이 1개의 포장시간이 7초 내외 이었으며, 일관작업의 성능은 7초 내외로 보는 것이 적합하다. 그 이유는 화상처리로 등급을 판단하는 시간은 컨베이어 속도와 관계없이 1초 이내이기 때문이다. 이렇게 산출된 오이선별능력은 상용화 하기에는 너무 적은 처리량이라고 예상되기 때문에 포장기를 1대가 아닌 각 등급별 배출구에 각 각 개별적으로 장착한다면 성능이 3배 즉 2초에 1개 처리 정도로 올라갈 것으로 예상하고 있다. 또한 한가지 방안으로는 포장하지않고 출하하는 경우에는 고속선별이 가능하여 초당 1개의 처리가 가능하다.

결과적으로 1일 4시간 선별 처리능력은 $1 \times 4 \times 60 \times 60 \times 1 = 14400$ (개) 이고 1일 4시간 1개포장기로 선별 및 포장 처리시간은 $1 \times 4 \times 60 \times 60 \times (1/7)$

= 2057 (개) 이다 또한 1일 4시간 3개 포장기를 장착한다고 가정할 경우 $1 \times 4 \times 60 \times 60 \times (1/2) = 7200$ (개) 처리 가능하다.

운반장치에서는 개폐장치와 운송장치가 잘 작동하여 오이 이송 및 분리를 잘 수행할 수 있었다. 그러나 오이가 포장장치에서 분리장치로 떨어지는 높이가 높아 오이가 상할 우려가 있어 운반장치 출구에 가이드를 부착하고, 탄력성이 있는 재질로 바꾸면 좀더 오이의 손상을 줄일 수 있을 것이다.

전체적인 시스템의 작업은 잘 수행되었지만 작업시간을 많이 요구하였기 때문에 시스템의 일관작업의 흐름도 혹은 작업속도를 고려한 연구개발이 더 필요한 실정이다. 또한 오이의 투입은 사용자의 수작업으로 이루어져야 하므로 전작업이 자동화 될 수 있도록 자동투입장치의 개발이 필요하다고 생각된다.

참고문헌

1. Brown, G. K. and J. A. Throop, 1986, Apple sorting with machine vision, Trans. of ASAE, 17(1), pp17-19
2. Gonzalez, R. C. and P. Wintz, 1987. Digital image processing. 2nd edition, Addition Wesley, pp414-417.
3. Groover, M. P., W. Mitchell, N. N. Roger and Odrey N. G. 1986, Industrial robots: Technology, programming and applications. McGraw Hill.
4. Fu, K. S., R. C. Gonzalez and C. S. G. Lee. 1987. Robotics: control, sensing, vision, and intelligence. McGraw-Hill.
5. 황헌, 이충호, 1994, 버섯 전후면과 꼭지부 상태의 자동 인식, 한국농업기계학회, 19(2):124-137
6. 金耕兵其. 1989. キュウリの曲かり果に関する諸問題[1] 曲かりか發生する栽培條件. 日本農業および園藝 64-1: 47-52
7. 조한근, 송현갑, 1994, 컴퓨터 시각에 의한 잎담배의 외형 및 색 특징 추출, 한국농업기계학회, 19(4):380-396.
8. Kitchen P. W. and A. Pugh. 1981. Processing of binary images. Robot vision
9. Lee, Dae-Weon. 1990. A robotic and vision system for locating and transferring container grown tobacco seedlings. Ph. D. Thesis, Department of biological and agricultural

engineering, North Carolina state university, Rleigh, NC.

10. Marc, L., T. Zimmerman, C. Baur, P. Gugerli and T. Pun, 1994, Potato Operation:automatic detection of potato diseases, Proceeding of Optics in Agricultural, Forestry, and Biological Processing, Vol.2345, pp2-7.
11. Miller B. K., M. J. Delwiche. 1989. A color vision system for peach grading. Trans. of ASAE. 32(4):1484-1490.
12. 노상하, 류관희, 김일웅, 1990, 영상처리 시스템을 이용한 과일
의 기하학적 특성 측정, 한국농업기계학회, 15(1):23-32
13. 노상하, 이종환, 이승훈, 1992, 흑백영상처리장치를 이용한 과일
선별기 개발에 관한 연구(I)-크기 및 선택판정, 한국농업기
계학회, 17(4):354-363
14. 농촌진흥청. 1989. 원예작물 생산과 연구의 국내의 동향. 농촌
진흥청 원예실험장
15. 농산물유통공사. 1995. 수출농수산물 품목별 무역정보
16. 농수산부, 농협중앙회. 1995. 농수산물 표준출하 규격집
17. Rehkugler G. E., A. J. Throop. 1986. Apple sorting with
machine vision. Trans. of ASAE. 29(5):1338-1397.
18. Sarkar, N., and R. R. Wolfe, 1985, "Feature extracion
techniques for sorting tomatoes by computer vision",
Transactions of the ASAE 28(3): pp 970-979.

부록 (Appendix)

- A. 프로그램
- B. 포장장치 및 입관작업의 실험자료

A. 프로그램

```

/*****
/*
/* 농산물이미지 프로세싱 프로그램, 주 헤더 파일  MENU.MAN.H  */
/*
/* 프로젝트: oei.ide, 컴파일러: 볼랜드 C++ 4.5 for windows (ide)  */
*****/

/* C 표준 라이브러리 헤더 */

#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#include <dir.h>
#include <alloc.h>
#include <conio.h>
#include <time.h>
#include <math.h>
#include <limits.h>
#include <bios.h>
#include <process.h>

/* 한라 프로 2.4 라이브러리 헤더 */
#include "ascii.h"
#include "extkey.h"
#include "hanbgi.h"

#include "hanlib.h"
#include "hanmou.h"
#include "hanmeq.h"

/* 알림판에 최대 표시되는 한 행당 문자열 */
#define MSG_MAX      19

```

```

#define HALT_ERROR      -1
#define TRUE            1
#define FALSE           0

#define SUCCESS         1
#define FAIL            0

#define ON              1
#define OFF             0

#define MAX_SCAN_PAGE  4

typedef short int BOOL;
typedef short int SWITCH;

/* 키보드 버퍼 초기화 */
void ClearKeybuf(void);

/* 마우스 커서가 위치한 곳을 알려 주는 상태 함수 */
void MouseAxisView(int mousePlane);

/*****
/*          Menu Modules functions          */
*****/

/* 프로그램 전처리 */
void Start(void);

/* 메뉴 관리자 */
void MainMenuMan(void);

/* 농작물 선택 */
void Crops(int, char*);

/* 화상정보 확대 기능 */

```

```

void ZoomView(int, char*);

/* 갖가지 필터 기능 */
void FilterView(int, char*);

/* 신별된 농작물 출하 허가 및 출하기준 수정 */
void OutCrops(int, char*);

/* 저장된 정보를 탐색하거나 정렬한다. */
void Sorting(int, char*);

/* 파일 관리자 */
void Files(int, char*);

/* 프린터 관리자 */
void PrintMan(int, char*);

/* 도우미 */
void Help(int, char*);

/* 종료 함수 */
void System(int, char*);

/*****
/* 화면 실시간 (조깅음) */
/* 버퍼 포인터 4개 (IBM 호환) */
*****/
extern byte far* CameraBuffer[MAX_SCAN_PAGE+1]

/* 버퍼의 정화상 정보의 갱신 여부 */
extern int bufUpdate;

/* 도움말을 위한 버퍼 */
extern byte far *HelpBuf[2];

/* 간단한 애니메이션을 위한 버퍼 */
extern byte far *faceBuf;

```

```

extern byte far *gearBuf;

extern char msgBuf[4][MSG_MAX+1];

/*****/
/* 프로그램이 위험할 때 사용 (TEST 용) */
/*****/

extern BOOL odxDrvFlag;

extern BOOL exitFlag;

/*****/
/* machine.h */
/*****/

#include <io.h>
#include <dos.h>
#include <time.h>

#define FATAL_ERROR -1
#define MACHINE_WAIT 1
#define MACHINE_READY 0

#define MACHINE_PORT 0x204
#define MACHINE_ERROR_PORT 0x304

#define ADDR_INIT_SWITCH 0x40

#define ADDR_FATAL_ERROR_LIGHT 0x02
#define ADDR_READY_LIGHT 0x04
#define ADDR_WAIT_LIGHT 0x08

```



```

#define ADDR_CLASS1_GATE      0x40
#define ADDR_CLASS2_GATE      0x80
#define ADDR_CLASS3_GATE      0x60
#define ADDR_OUTCLASS_GATE    0x20

#define ADDR_ALLOW_WRAP_OFF   0x01
#define ADDR_ALLOW_WRAP_ON    0x00

/*-----*/

#define MAX_LIFE_TIME          20
#define LIFE_TIME              25 //sec
#define MIN_LIFE_TIME          20

#define READY_ACCESS_TIME     1 //sec
#define NON_OEL_TIME          5 //sec
#define ZERO_TO_WRAP_TIME     10 //sec

/* lpt1    lpt2 */
#define porta    0x304 /* 0x3bc */
#define portb    0x304 /* 0x3be */
#define portc    0x302 /* 0x3bd */
#define port_con 0x307
#define mot1c    0x02
#define mot1ccw  0x01

#define mot2c    0x04
#define mot2ccw  0x08

#define mot3c    0x10

```

```

/*****/
/*      메뉴관리자, menu_man.c <- menu_man.h      */
/*****/

#include "main_man.h"
#include "menu_man.h"
#include "gui_man.h"

/*****/
/*      Module      */
/*****/
static byte substring[9][5][17] =
    { { "농산물 선택", "들어내기", "건본 부르기", "건본 지장", "완심 대
상" },
      { "8 X 8", "숫자로", "그래프로", "깨끗이", "히스토그램" },
      { "16 칼라", "2 칼라", "경제값 중심", "라플라시안", "엠펙싱" },
      { "출하 허가", "기준 보기", "기준 수정", "기준 비교", "우선 순위"
    },
    { "면적 순서", "둘레 순서", "휘어짐 순서", "생성 순서", "혼합 정
렬" },
    { "세포 저장", "보고서 보기", "보고서 수정", "보고서 비교", "새
보고서" },
    { "세포 출력", "부분Rptout", "미정", "미정", "미정" },
    { "도우미 사용법", "찾아보기", "미정", "미정", "Spider 란..." },
    { "끝내기", "속도조절", "깨끗이", "색깔 바꾸기", "다시 그리기" }
    };

void MainMenuMan(void)
{
    /*      initial value      */
    AREA Area = {
        SYSTEM, CROPS, SYSTEM, CROPS,
        SUB5, SUB1, SUB5, SUB1,
        MAINMENU_PLANE
    };
}

```

```

mouevnque_t *mouEvent;

int keySelected;

Area.old = MainGuidel.inc(&Area.old, &Area.pre);

mou_setwindowpxy(SUB_SX, SUB_SY, SUB_EX, SUB_EY)
mou_moveto(0,0);
mou_show();

do {
    mouEvent = meq_getevn();
    keySelected = getkey();

    if (CTRL_BREAK==keySelected) exitFlag=TRUE;

    switch (Area.menuControlPlane) {
        case MAINMENU_PLANE:
            if (NULL==keySelected) {
                Area = MouseSelected(mouEvent, &Area);
                MouseAxisView(MAINMENU_PLANE);
            }
            else {
                Area = KeySelected(&keySelected, &Area);
            }
            break;

        case SUBMENU_PLANE:
            if (NULL==keySelected) {
                Area =
                    SubMouseSelected(mouEvent, &Area);
                MouseAxisView(SUBMENU_PLANE);
            }
            else {
                Area =
                    SubKeySelected(&keySelected, &Area);
            }
    }
}

```

```

        break;
    }
} while(TRUE!=exitFlag);

meq_clrallv();
}

void ToggleChange(int menu_control_plane)
{
    hsolidbarpxy(488, 401, 631, 472, MAGENTA); /* Message Area
Painting */
    SubMenuBox();

    switch (menu_control_plane) {
        case MAINMENU_PLANE:
            str_box(487, 93, "주 차림판", WHITE, BLACK, 20);
            str_box(487, 236, "부 차림판", WHITE, LIGHTGRAY, 20);
            break;
        case SUBMENU_PLANE:
            str_box(487,93,"주 차림판", WHITE, LIGHTGRAY, 20);
            str_box(487,236,"부 차림판", WHITE, BLACK, 20);
            break;
    }
    ScrollButtonDisplay(490, 95, 0);
    ScrollButtonDisplay(490, 238, 0);
}

int MainGuideLine(const int* old_area, const int* pre_area)
{
    int i;
    int row[PRE+1];
    int col[PRE+1];

    int table[PRE+1];

    row[OLD] = (*old_area - 1) % 3;

```

```

col[OLD] = (*old_area -1) / 3;
row[PRE] = (*pre_area -1) % 3;
col[PRE] = (*pre_area -1) / 3;

table[OLD] = *old_area;
table[PRE] = *pre_area;

/* 가이드 라인 위치가 변화되었을 때 작동함 */
if (table[OLD] != table[PRE]) {

    for(i=0; i<PRE+1; i++) {
        if (OLD==i) {
            setcolor(MAGENTA);
        }
        else {
            setcolor(BLACK);
        }
        rectangle(491+row[i]*38,      118+col[i]*38,      528+row[i]*38
155+col[i]*38);
        rectangle(490+row[i]*38,      117+col[i]*38,      529+row[i]*38
156+col[i]*38);
    }

    SubMenuDisplay(*pre_area, OFF);
}
return table[PRE];
}

int SubGuideLine(const int *old_area, const int *pre_area)
{
    int i;
    int col[PRE+1];

    int table[PRE+1];

    col[OLD] = *old_area -1;
    col[PRE] = *pre_area -1;

    table[OLD] = *old_area;

```

```

table[PRE] = *pre_area;

/* 가이드 라인 위치가 바뀌었을 때만 갱신함 */
if (table[OLD] != table[PRE]) {

    for (i=0; i<PRE+1; i++) {
        if (OLD==i) {
            setcolor(MAGENTA);
        }
        else {
            setcolor(BLACK);
        }
        rectangle(491, 262+col[i]*22, 628, 283+col[i]*22);
        rectangle(490, 261+col[i]*22, 629, 284+col[i]*22);
    }
}
return table[PRE];
}

```

```

AREA MainSelectedLine(AREA* ptr_area)
{
    int i;
    int row[PRE+1];
    int col[PRE+1];

    row[OLD] = (ptr_area->oldSelected -1) % 3;
    col[OLD] = (ptr_area->oldSelected -1) / 3;

    for(i=0; i<2; i++) {
        hvline(492+i+row[OLD]*38, 119+i+col[OLD]*38, 36-i*2, WHITE);
        hhline(492+i+row[OLD]*38, 119+i+col[OLD]*38, 36-i*2, WHITE);
        hvline(527-i+row[OLD]*38, 119+i+col[OLD]*38, 36-i*2,
DARKGRAY);
        hhline(492+i+row[OLD]*38, 154-i+col[OLD]*38, 36-i*2,
DARKGRAY);
    }
}

```

```

row[PRE] = (ptr_area->preSelected -1) % 3;
col[PRE] = (ptr_area->preSelected -1) / 3;

for(i=0; i<2; i++) {
    hvline(492+i+row[PRE]*38,      119+i+col[PRE]*38,      36-i*2
DARKGRAY);
    hhline(492+i+row[PRE]*38,      119+i+col[PRE]*38,      36-i*2
DARKGRAY);
    hvline(527-i+row[PRE]*38,      119+i+col[PRE]*38,      36-i*2
WHITE);
    hhline(492+i+row[PRE]*38,      154-i+col[PRE]*38,      36-i*2
WHITE);
}

ptr_area->menuControlPlane = SUBMENU_PLANE;
// ToggleChange(SUBMENU_PLANE);

ptr_area->oldSub = SUB5;
ptr_area->preSub = SUB1;

AniRect(492+16+row[PRE]*38, 119+16+col[PRE]*38, 488, 258, 143, 119
ON);
ToggleChange(SUBMENU_PLANE);

SubMenuDisplay(ptr_area->preSelected, ON);

// AniRect(492+16+row[PRE]*38, 119+16+col[PRE]*38, 487, 257, 136
120, OFF);

ptr_area->oldSelected = ptr_area->preSelected;
ptr_area->oldSub      =      SubGuideLine(&ptr_area->oldSub
&ptr_area->preSub);

return *ptr_area;
}

```

```

AREA SubSelectedLine(AREA* ptr_area)
{
    int i;
    int col[PRE+1];

    col[OLD] = ptr_area->oldSubSelected -1;

    for(i=0; i<2; i++) {
        hvline(492+i,    263+i+col[OLD]*22,    20-i*2, WHITE);
        hhline(492+i,    263+i+col[OLD]*22,    136-i*2, WHITE);

        hvline(627-i,    263+i+col[OLD]*22,    20-i*2, DARKGRAY);
        hhline(492+i,    282-i+col[OLD]*22,    136-i*2, DARKGRAY);
    }

    col[PRE] = ptr_area->preSubSelected -1;

    for(i=0; i<2; i++) {
        hvline(492+i,    263+i+col[PRE]*22,    20-i*2, DARKGRAY);
        hhline(492+i,    263+i+col[PRE]*22,    136-i*2, DARKGRAY);

        hvline(627-i,    263+i+col[PRE]*22,    20-i*2, WHITE);
        hhline(492+i,    282-i+col[PRE]*22,    136-i*2, WHITE);
    }

    /*
        Run extern Module
    */
    /*
        RunMenuModule(ptr_area->oldSelected, ptr_area->preSubSelected)
    */

    for(i=0; i<2; i++) {
        hvline(492+i,    263+i+col[PRE]*22,    20-i*2, WHITE);
        hhline(492+i,    263+i+col[PRE]*22,    136-i*2, WHITE);

        hvline(627-i,    263+i+col[PRE]*22,    20-i*2, DARKGRAY);
        hhline(492+i,    282-i+col[PRE]*22,    136-i*2, DARKGRAY);
    }
    // printf("\n");
}

```



```

for (i=0; i<PRE+1; i++) {
    if (OLD==i) {
        setcolor(MAGENTA);
    }
    else {
        setcolor(BLACK);
    }
    rectangle(491, 262+col[i]*22, 628, 283+col[i]*22);
    rectangle(490, 261+col[i]*22, 629, 284+col[i]*22);
}

ptr_area->oldSubSelected = ptr_area->preSubSelected;
return *ptr_area;
}

```

```

AREA KeySelected(const int* key, AREA* kptr_area)
{
    int area = NONE;
    int row, col;

    row = (kptr_area->old - 1) % 3 + 1;
    col = (kptr_area->old - 1) / 3 + 1;

    switch (*key) {

        case UPARROW:
            if (1!=col) col--;
            area = row + 3*(col-1);
            kptr_area->old = MainGuideLine(&kptr_area->old, &area)
            break;

        case LEFTARROW:
            if (1!=row) row--;
            area = row + 3*(col-1);
            kptr_area->old = MainGuideLine(&kptr_area->old, &area)
            break;
    }
}

```

```

case RIGHTARROW:
    if (3!=row) row++;
    area = row + 3*(col-1);
    kptr_area->old = MainGuideLine(&kptr_area->old, &area)
    break;

case DOWNARROW:
    if (3!=col) col++;
    area = row + 3*(col-1);
    kptr_area->old = MainGuideLine(&kptr_area->old, &area)
    break;

case ENTER:
    area = row + 3*(col-1);
    kptr_area->preSelected = area;
    *kptr_area = MainSelectedLine(kptr_area);
    break;

case ESC: case ALT_X:
    area = SYSTEM;
    kptr_area->old = MainGuideLine(&kptr_area->old, &area);
    kptr_area->preSelected = area;
    *kptr_area = MainSelectedLine(kptr_area);
    break;

case F1:
    area = HELPS;
    kptr_area->old = MainGuideLine(&kptr_area->old, &area);
    kptr_area->preSelected = area;
    *kptr_area = MainSelectedLine(kptr_area);
    break;
}

return *kptr_area;
}

```

```

AREA SubKeySelected(const int* key, AREA* kptr_area)
{
    int i;

    int col = kptr_area->oldSub;

    int main_row, main_col;

    switch (*key) {

        case UPARROW: case LEFTARROW:
            if (SUB1!=col) {
                col--;
            }
            else {
                col=SUB5;
            }
            kptr_area->oldSub = SubGuideLine(&kptr_area->oldSub, &col)
            break;

        case DOWNARROW: case RIGHTARROW:
            if (SUB5!=col) {
                col++;
            }
            else {
                col=SUB1;
            }
            kptr_area->oldSub = SubGuideLine(&kptr_area->oldSub, &col)
            break;

        case ENTER:
            kptr_area->preSubSelected = col;
            *kptr_area = SubSelectedLine(kptr_area);
            break;

        case ESC:
            kptr_area->menuControlPlane = MAINMENU_PLANE;
//      ToggleChange(MAINMENU_PLANE);
    }
}

```

```

        main_row = (kptr_area->preSelected -1) % 3;
        main_col = (kptr_area->preSelected -1) / 3;
        AniRect(492+16+main_row*38, 119+16+main_col*38, 488, 258, 143
119, OFF);
        ToggleChange(MAINMENU_PLANE);

        for(i=0; i<2; i++) {
            hvline(492+i+main_row*38,      119+i+main_col*38,      36-i*2
WHITE);
            hhline(492+i+main_row*38,      119+i+main_col*38,      36-i*2
WHITE);
            hvline(527-i+main_row*38,      119+i+main_col*38,      36-i*2
DARKGRAY);
            hhline(492+i+main_row*38,      154-i+main_col*38,      36-i*2
DARKGRAY);
        }
        break;
    }

    return *kptr_area;
}

```

```

AREA MouseSelected(const mouevnque_t *mou_event,
                    AREA* mptr_area)
{
    int area=NONE;

    if (MAINMENU_AREA1) {
        area=CROPS;
    }
    else if (MAINMENU_AREA2) {
        area=ZOOM;
    }
    else if (MAINMENU_AREA3) {
        area=FILTER;
    }
}

```

```

else if (MAINMENU_AREA4) {
    area=OUTCROPS;
}
else if (MAINMENU_AREA5) {
    area=SORTING;
}
else if (MAINMENU_AREA6) {
    area=FILEMAN;
}
else if (MAINMENU_AREA7) {
    area=PRINTMAN;
}
else if (MAINMENU_AREA8) {
    area=HELPS;
}
else if (MAINMENU_AREA9) {
    area=SYSTEM;
}
else {
    area=NONE;
}

if (NONE!=area) {
    mptr_area->old = MainGuideLine(&mptr_area->old, &area)

    /* 더블 클릭 버전으로 바꾸기 (어려움) */
    if (meq_isdownevn(mou_event)) {
        meq_clrallvn();
        mptr_area->preSelected = area;
        *mptr_area = MainSelectedLine(mptr_area);
    }
}

return *mptr_area;
}

```

AREA SubMouseSelected(const mouevnque_t* mou_event,

```

                                AREA* mptr_area)
{
    int i;

    int main_row, main_col;

    int area=NONE;

    if (SUBMENU_AREA1) {
        area=SUB1;
    }
    else if(SUBMENU_AREA2) {
        area=SUB2;
    }
    else if(SUBMENU_AREA3) {
        area=SUB3;
    }
    else if(SUBMENU_AREA4) {
        area=SUB4;
    }
    else if(SUBMENU_AREA5) {
        area=SUB5;
    }
    else {
        area=NONE;
    }

    /* Case -> Cancel button */
    if (meq_isrdownevn(mou_event)) {
        area=NONE;

        mptr_area->menuControlPlane = MAINMENU_PLANE;
//      ToggleChange(MAINMENU_PLANE);

        main_row = (mptr_area->preSelected -1) % 3;
        main_col = (mptr_area->preSelected -1) / 3;
        AniRect(492+16+main_row*38, 119+16+main_col*38, 488, 258, 143
119, OFF);
        ToggleChange(MAINMENU_PLANE);

```

```

        for(i=0; i<2; i++) {
            hvline(492+i+main_row*38,      119+i+main_col*38,      36-i*2.
WHITE);
            hhline(492+i+main_row*38,      119+i+main_col*38,      36-i*2
WHITE);
            hvline(527-i+main_row*38,      119+i+main_col*38,      36-i*2
DARKGRAY);
            hhline(492+i+main_row*38,      154-i+main_col*38,      36-i*2
DARKGRAY);
        }
        return *mptr_area;
    }

    if (NONE!=area) {
        mptr_area->oldSub = SubGuideLine(&mptr_area->oldSub, &area);
        if (meq_isdownevn(mou_event)) {
            mptr_area->preSubSelected = area;
            *mptr_area = SubSelectedLine(mptr_area);
        }
    }
    return *mptr_area;
}

/* 함수간 통신 루틴 만들것 */
void RunMenuModule(int mainarea, int subarea)
{
    switch(mainarea) {
        case CROPS:
            Crops(subarea, substring[mainarea-1][subarea-1]);
            break;
        case ZOOM:
            ZoomView(subarea, substring[mainarea-1][subarea-1]);
            break;
        case FILTER:
            FilterView(subarea, substring[mainarea-1][subarea-1]);
            break;
        case OUTCROPS:
            OutCrops(subarea, substring[mainarea-1][subarea-1]);
    }
}

```

```

        break;
    case SORTING:
        Sorting(subarea, substring[mainarea-1][subarea-1]);
        break;
    case FILEMAN:
        Files(subarea, substring[mainarea-1][subarea-1]);
        break;
    case PRINTMAN:
        PrintMan(subarea, substring[mainarea-1][subarea-1]);
        break;
    case HELPS:
        Help(subarea, substring[mainarea-1][subarea-1]);
        break;
    case SYSTEM:
        System(subarea, substring[mainarea-1][subarea-1]);
        if (SUB3==subarea||SUB5==subarea) SubMenuDisplay(SYSTEM
ON);
        break;
    }
}

void SubMenuDisplay(int mainselected, BOOL run) /* 0 ----- 4 */
{
    int i, j;
    int view_color;
    switch (run) {
        case ON:
            view_color = BLACK;
            break;
        case OFF:
            view_color = DARKGRAY;
            break;
    }

    str_box(487, 257, "", 0, MAGENTA,120); /* Clear */

    /* ReDrawing SubMenuBox */
    for(i=0; i<2; i++) {
        for(j=0; j<5; j++) {

```



```

        hvline(492+i,    263+i+j*22,    20-i*2, WHITE);
        hhline(492+i,    263+i+j*22,    136-i*2, WHITE);

        hvline(627-i,    263+i+j*22,    20-i*2, DARKGRAY);
        hhline(492+i,    282-i+j*22,    136-i*2, DARKGRAY);
    )
}

hsetcolor(BLACK);
hsetbkcolor(LIGHTGRAY);
hsettextjustify(CENTER_TEXT);
registerhanfont(HAN10MD1);

hsetoutputmode(OVERLAP);

for(i=0; i<5; i++) {
    hsolidbarpxy(494, 265+i*22, 625, 280+i*22, LIGHTGRAY);
    hsetcolor(WHITE);
    hputspxy(560+1, 265+1+i*22, substring[mainselected-1][i]);
    hsetcolor(view_color);
    hputspxy(560, 265+i*22, substring[mainselected-1][i]);
}
hsetoutputmode(OVERWRITE);

registerhanfont(HAN412GS1);
}

```

```

/*****
/*      농산물 이미지 프로세싱 프로그램 , 초기화파일  Start.c      */
/*****

```

```

#include "main_man.h"
#include "menu_man.h"
#include "gui_man.h"

```

```

#include "odx_man.h"

/*****
/*                               */
/*****

void Start(void)
{
    struct palettetype pal;
    int color, maxcolor;

    inithanlib(BGIMODE, HANDETECT, HAN4I2GS1, DEFENGFONT)
    if (hgraphresult() < NOERROR)
        puterr("This program does not run in MDA or CGA!", 1);

    /* ODX 드라이버 초기화 */
    odxDrvFlag = Initial_odx();

    /*****
    /*       초기화면을 한꺼번에 띄움       */
    /*****

    maxcolor = getmaxcolor();
    /* grab a copy of the palette */
    getpalette(&pal);
    /* black out the colors one by one */
    for (color=1; color<=maxcolor; color++)
    {
        setpalette(color, BLACK);
    }

    RePaint();
    MessageView("시설자동화연구실",
                "95.8. - 96.",
                "개발자: Bool Whee",
                "후 원: 도깨비");
    meq_init(MOU_ARROW);
    meq_clrallv();
    Logo();

```

```

ViewBmp("2002.bmp",492, 257, 135, 120);
faceBuf = farmalloc(10000);

/* restore the palette colors */
setallpalette(&pal);
delay(1000);
Clear();

getimage(487, 257, 633, 377, faceBuf);

/* 화상 저장 비퍼를 할당 한다. ---> 매크로 처리(정확히 계산) *

CameraBuffer[0] = farmalloc(PAGE1_SIZE+1);
CameraBuffer[1] = farmalloc(PAGE2_SIZE+1);
CameraBuffer[2] = farmalloc(PAGE3_SIZE+1);
CameraBuffer[3] = farmalloc(PAGE4_SIZE+1);

if ( NULL==CameraBuffer[0]
    || NULL==CameraBuffer[1]
    || NULL==CameraBuffer[2]
    || NULL==CameraBuffer[3] ) {
    printf("Not enough memory\r\n");
    System(SUB1);
}

MessageView("영상처리 준비완료",
            "시스템 초기화를",
            "성공시켰습니다.",
            "선택하세요!");
}

void MouseAxisView(int mousePlane)
{
    int ij;

    int prestatearea=0;

```

```

mouevnque_t *mouOkCancelEvent;
int keybuf;

hsetcolor(BLACK);
hsetbkcolor(LIGHTBLUE);
hsettextjustify(CENTER_TEXT);

prestatearea++;
switch (mousePlane) {
  case MAINMENU_PLANE: case SUBMENU_PLANE:
    i=mou_getapx(); j=mou_getapy();
    if ( i>=492 && i<=527 && j>=119 && j<=154 ) {
      prestatearea=1;
    }
    else if ( i>=530 && i<=565 && j>=119 && j<=154 ) {
      prestatearea=2;
    }
    else if ( i>=568 && i<=603 && j>=119 && j<=154 ) {
      prestatearea=3;
    }
    else if ( i>=492 && i<=527 && j>=157 && j<=192 ) {
      prestatearea=4;
    }
    else if ( i>=530 && i<=565 && j>=157 && j<=192 ) {
      prestatearea=5;
    }
    else if ( i>=568 && i<=603 && j>=157 && j<=192 ) {
      prestatearea=6;
    }
    else if ( i>=492 && i<=527 && j>=195 && j<=230 ) {
      prestatearea=7;
    }
    else if ( i>=530 && i<=565 && j>=195 && j<=230 ) {
      prestatearea=8;
    }
    else if ( i>=568 && i<=603 && j>=195 && j<=230 ) {
      prestatearea=9;
    }
    else {

```

```

        prestatearea=0;
    }
    break;

case OKCANCEL_PLANE:
    meq_clralllevn();
    mou_setwindowpxy(MAIN_AREA_S,
                    MAIN_AREA_S,
                    MAIN_AREA_E,
                    MAIN_AREA_E);
    mou_moveto(230, 230);
    setcolor(LIGHTGRAY);

    //cursor_on();
    //  dialog("*.","Load Dialog Box");
    //cursor_off();

    for(;;) {
        mouOkCancelEvent=meq_getevn();
        keybuf = getkey();

        if (OK_BUTTON_AREA==(ENTER==keybuf)) {
            rectangle(174, 84, 205, 99);
            mou_setwindowpxy(SUB_SX, SUB_SY, SUB_EX, SUB_EY)
            mou_moveto(0,0);
            meq_clralllevn();
            break;
        }
        else if (CANCEL_BUTTON_AREA==(ESC==keybuf)) {
            rectangle(415, 84, 446, 99);
            mou_setwindowpxy(SUB_SX, SUB_SY, SUB_EX, SUB_EY)
            mou_moveto(0,0);
            meq_clralllevn();
            break;
        }
    }
    break;
}

```

```

    return;
}

/*****
/* 농산물 이미지 프로세싱 프로그램 , 안전 종료 파일 system.c */
*****/

#include "main_man.h"
#include "menu_man.h"
#include "gui_man.h"
#include "oidx_man.h"

/*****
/*                               Module                               */
*****/

void System(int sub)
{
    switch (sub) {
        case 1:
            farfree(CameraBuffer[0]);
            farfree(CameraBuffer[1]);
            farfree(CameraBuffer[2]);
            farfree(CameraBuffer[3]);

            farfree(faceBuf);
            if(NULL!=HelpBuf[0]||NULL!=HelpBuf[1]) {
                farfree(HelpBuf[0]);
                farfree(HelpBuf[1]);
            }
            fcloseall();
            meq_clrallv();
            meq_close();
            exit(0);
            break;

        case 2:
            GetGearPIC(gearBuf);
    }
}

```

```

        MoveGearPIC(gearBuf, NULL, NULL);
        FreeGearPIC(gearBuf);
        break;

    case 3:
        hsolidbarpxy(0,0,639,479,RED); /* Visable CameraiImage */
        getch();
        RePaint();
        break;

    case 4:
        hsetwindowpalette(0);
        break;

    case 5:
        odxDrvFlag = Initial_odx();
        RePaint();
        break;
    )
)

```

```

/*****
/* 농산물 이미지 프로세싱 프로그램, 출하 파일   Outcrops.c   */
*****/

```

```

#include "main_man.h"
#include "menu_man.h"
#include "gui_man.h"
#include "operator.h"
#include "odx_man.h"
#include "machine.h"

```

```

#define MAX_CHAR_NUM 6

```

```

#define CELL_COL_EA 6

```

```

#define CELL_ROW_EA 7
#define RANK_COL_EA 6
#define RANK_ROW_EA 7

#define WSIZE 54
#define HSIZE 20
#define GAP_X 4
#define GAP_Y 4

#define MORE 1
#define LESS -1
#define NONE 0

```

```
typedef short int SIGN;
```

```
extern CELL Cell[DIVIDE_EA+1][DIVIDE_EA+1];
```

```

/*****                               MACHINE          GLOBAL
*****/
int CropsClass = 0;

const char record_name[8][MAX_CHAR_NUM+1] = {
    " 등급 ", " 길이 ", " 두께 ", " 휨각 ", " 고르기", " 면적 ", " 무게 ", "
거칠기"
};
const char record_subname[8][MAX_CHAR_NUM+1] = {
    " ", "*E^2mm", "*E^3mm", "E3 Deg", "E3 [R]", "(mm)^2", "
g ", " ? ? "
};
const char unit_name[7][MAX_CHAR_NUM+1] = {
    " mm", " mm", " Deg", " [R]", "(mm)^2", " g ",
" ? ? "
};

const char class_name[7][MAX_CHAR_NUM+1] = {
    " 1등급", " 2등급", " 3등급", " 등외 ", ".....", ".....", " 현재 "
};

```



```

int StoreClass[3] = { -1, -1, -1 };

unsigned long convert_int_object[7] = {0, 0, 0, 0, 0, 0, 0};
/* SIGN convert_rank_object[7]
   = {NONE, NONE, NONE, NONE, NONE, NONE, NONE};
*/

const char init_record_value[MAX_CHAR_NUM+1] = "100000";
const char rank_str[3][MAX_CHAR_NUM+1] = { "이하 ", "부시
" 이상 " };
const SIGN init_record_rank = NONE;

char record_value[6][7][MAX_CHAR_NUM+1];
SIGN record_rank[6][7] = {
    { 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0 }
};

byte Port1 = ADDR_READY_LIGHT;
byte Port2 = 0;

/*****
/*          、 Module          */
*****/

BOOL SelectClassView(int crops_number, int option);

BOOL SelectClassRankView(int crops_number);

BOOL select_cell(int* x, int* y);

```

```

BOOL edit_cell(char* data, int cell_x, int cell_y);
BOOL select_rank(int* x, int* y);
BOOL edit_rank(SIGN* data, int cell_x, int cell_y);

BOOL load_cell(char* data, int cell_x, int cell_y);
BOOL save_cell(const char* data, int cell_x, int cell_y);

/***** Cell *****/
BOOL NewClass(const char* init);
BOOL LoadClass(char
data[CELL_COL_EA][CELL_ROW_EA][MAX_CHAR_NUM+1]);
BOOL EditClass(char
data[CELL_COL_EA][CELL_ROW_EA][MAX_CHAR_NUM+1]);
BOOL SaveClass(const char
data[CELL_COL_EA][CELL_ROW_EA][MAX_CHAR_NUM+1]);

/***** Rank *****/
BOOL NewClassRank(const SIGN init);
BOOL LoadClassRank(SIGN rank[RANK_COL_EA][RANK_ROW_EA]);
BOOL EditClassRank(SIGN rank[RANK_COL_EA][RANK_ROW_EA]);
BOOL SaveClassRank(const SIGN
rank[RANK_COL_EA][RANK_ROW_EA]);

BOOL real_cell_data_view(const unsigned long dec, int lx, int ly);
BOOL cell_data_view(const unsigned long dec, int lx, int ly);
BOOL cell_data_convert_dec(const char* origin, unsigned long* dec);
BOOL cell_data_convert_str(unsigned long* dec, char* newstr);

BOOL rank_data_view(const SIGN rank_value, int lx, int ly);

/***** Pre Object Process *****/
BOOL ConvertIntegerObject(OBJECT *obj, unsigned long preObj[7]);
BOOL PreObjectCellView(const unsigned long preObj[7]);
BOOL PreObjectRankView(const SIGN preObj[7]);
BOOL RealPreObjectRankView(const char
str[7][MAX_CHAR_NUM+1]);

```

```

BOOL RelationObject(unsigned long preObj[7],
                    const char
data[CELL_COL_EA][CELL_ROW_EA][MAX_CHAR_NUM+1],
                    const SIGN
rank[RANK_COL_EA][RANK_ROW_EA],
                    int* crops_class);
void print_class(const int crops_class);

/*****/
/*          Machine Life Cycle List          */
/*****/

BOOL Wrapping(int oei_class);
BOOL InitWrapping(void);

BOOL Process1(void);
BOOL Process2(void);
BOOL Process3(void);
BOOL Process4(void);
BOOL Process5(void);

BOOL LifeCycle(void);

void OutCrops(int sub, char* subname)
{
    MessageView(subname, "", "", "");

    MainDialogBox(subname, VIEW_READ);
    SubMenuBox();

    SelectClassView(1, sub);

/* switch (sub) {
    case 1:
        break;
    case 2:
        break;
}

```

```

        case 3:
            break;
        case 4:
            break;
        case 5:
            break;
    }
*/
getch();
Clear();
SubMenuDisplay(4, ON);
}

BOOL SelectClassView(int crops_number, int option)
{
    int i, j;

    BOOL status;

    if (SUB2 == option || SUB3 == option || SUB4 == option || SUB5
== option) {
        for (i=0; i<8; i++) {
            freedom_box(record_name[i],
                D_SX+GAP_X+i*(WSIZE), D_CLIENT, WSIZE, HSIZE,
NORMAL);
        }
        for (i=0; i<8; i++) {
            freedom_box(record_subname[i],
                D_SX+GAP_X+i*(WSIZE), D_CLIENT+HSIZE*1, WSIZE,
HSIZE, REVERSE);
        }
        for (i=2; i<8; i++) {
            freedom_box(class_name[i-2],
                D_SX+GAP_X, D_CLIENT+HSIZE*i, WSIZE, HSIZE, NORMAL);
        }
    }

    status = load_cell(record_value[0][0], 0, 0);
    if (FAIL == status) {

```

```

        NewClass(init_record_value);
    }
    LoadClass(record_value);

    status = LoadClassRank(record_rank);
    if (SUCCESS != status) {
        NewClassRank(init_record_rank);
        LoadClassRank(record_rank);
    }

    if (SUB2 == option || SUB3 == option || SUB4 == option || SUB5
== option) {
        for (i=1; i<8; i++) {
            for (j=0; j<6; j++) {
                freedom_box(record_value[j][i-1],
                    D_SX+GAP_X+i*(WSIZE), D_CLIENT+HSIZE*(j+2),
                    WSIZE, HSIZE, REVERSE);
            }
        }
    }

    freedom_box(class_name[6],
        D_SX+GAP_X, D_CLIENT+HSIZE*8+GAP_Y, WSIZE, HSIZE
NORMAL);
    for (i=1; i<8; i++) {
        freedom_box(record_name[i],
            D_SX+GAP_X+i*(WSIZE), D_CLIENT+HSIZE*8+GAP_Y, WSIZE
HSIZE, NORMAL);
    }

    for (i=0; i<8; i++) {
        freedom_box("",
            D_SX+GAP_X+i*(WSIZE), D_CLIENT+HSIZE*9+GAP_Y,
            WSIZE, HSIZE, REVERSE);
        freedom_box("",
            D_SX+GAP_X+i*(WSIZE), D_CLIENT+HSIZE*10+GAP_Y,
            WSIZE, HSIZE, REVERSE);
    }

```

```

    if (SUB2 == option || SUB3 == option || SUB4 == option || SUB5
== option ) {
        for (i=11; i<17; i++) {
            freedom_box(class_name[i-11],
                D_SX+GAP_X, D_CLIENT+HSIZE*i+GAP_Y*2, WSIZE, HSIZE,
NORMAL);
        }

        for (i=1; i<8; i++) {
            for (j=11; j<17; j++) {
                freedom_box(rank_str[record_rank[j-11][i-1]+1],
                    D_SX+GAP_X+i*(WSIZE), D_CLIENT+HSIZE*j+GAP_Y*2,
                    WSIZE, HSIZE, REVERSE);
            }
        }
    }

    if (SUB1 == option) {
        Lifecycle();
    }

    if (SUB3 == option) {
        EditClass(record_value);
        SaveClass(record_value);
        EditClassRank(record_rank);
        SaveClassRank(record_rank);
    }
    else if (SUB5 == option) {
        EditClassRank(record_rank);
        SaveClassRank(record_rank);
    }
    return SUCCESS;
}

/*****
/*----- Cell -----*/

BOOL NewClass(const char* init)

```

```

{
    FILE *fp;
    int i, j;

    fp = fopen("Class.dat", "w+");
    for (j=0; j<6; j++) {
        for (i=0; i<6; i++) {
            fprintf(fp, "%s,", init);
        }
        fprintf(fp, "%s\n", init);
    }
    fclose(fp);
    return SUCCESS;
}

```

```

BOOL                                LoadClass(char
data[CELL_COL_EA][CELL_ROW_EA][MAX_CHAR_NUM+1])
{
    int i, j;

    for (j=0; j<6; j++) {
        for (i=0; i<7; i++) {
            load_cell(data[j][i], i, j);
        }
    }
    return SUCCESS;
}

```

```

BOOL                                EditClass(char
data[CELL_COL_EA][CELL_ROW_EA][MAX_CHAR_NUM+1])
{
    int old_g_x;
    int old_g_y;
    int g_x;
    int g_y;

    int old_data_x = 0;
    int old_data_y = 0;
    int data_x = 0;

```

```

int data_y = 0;

int key;
BOOL end_while = FALSE;

old_g_x = D_SX+GAP_X+ (old_data_x+1)*(WSIZE);
old_g_y = D_CLIENT+(old_data_y+1)*HSIZE*2;
g_x = D_SX+GAP_X+ (data_x+1)*(WSIZE);
g_y = D_CLIENT+(data_y+1)*HSIZE*2;
freedom_box(data[data_y][data_x],
            g_x, g_y, WSIZE, HSIZE, NORMAL)

do {
    key = getkey();

    if (NULL!=key) {

        switch (key) {
        case LEFTARROW:
            if (0!=data_x) {
                data_x--;
            }
            break;
        case RIGHTARROW:
            if (6!=data_x) {
                data_x++;
            }
            break;
        case UPARROW:
            if (0!=data_y) {
                data_y--;
            }
            break;
        case DOWNARROW:
            if (5!=data_y) {
                data_y++;
            }
            break;
        case ENTER:

```



```

        break;
    case ESC:
        end_while = TRUE;
    }

    if ( (key!=ESC && key!=ENTER)
        && data_x!=old_data_x || data_y!=old_data_y ) {
        g_x = D_SX+GAP_X+ (data_x+1)*(WSIZE);
        g_y = D_CJ+HSIZE+(data_y+1)*HSIZE;
        freedom_box(data[old_data_y][old_data_x],
                    old_g_x, old_g_y, WSIZE, HSIZE, REVERSE);
        freedom_box(data[data_y][data_x],
                    g_x, g_y, WSIZE, HSIZE, NORMAL);
        old_data_x = data_x;
        old_data_y = data_y;
        old_g_x = g_x;
        old_g_y = g_y;
    }
    else if (ENTER==key) {
        edit_cell(data[data_y][data_x], g_x, g_y);
        freedom_box(data[data_y][data_x],
                    g_x, g_y, WSIZE, HSIZE, NORMAL);
    }
}
} while (TRUE!=end_while);

return SUCCESS;
}

```

```

BOOL SaveClass(const char
data[CELL_COL_EA][CELL_ROW_EA][MAX_CHAR_NUM+1])
{
    int i, j;

    for (j=0; j<6; j++) {
        for (i=0; i<7; i++) {
            save_cell(data[j][i], i, j);
        }
    }
}

```

```

    return SUCCESS;
}

/*****
/*----- Rank -----*/

BOOL NewClassRank(const SIGN init)
{
    FILE *fp;
    int i, j;
    fp = fopen("Rank.dat", "w+");
    for (j=0; j<6; j++) {
        for (i=0; i<6; i++) {
            fprintf(fp, "%2d", init);
        }
        fprintf(fp, "%2d\n", init);
    }
    fclose(fp);
    return SUCCESS;
}

BOOL LoadClassRank(SIGN rank[RANK_COL_EA][RANK_ROW_EA])
{
    int i, j;
    FILE *fp;
    char ch;

    fp = fopen("Rank.dat", "r");
    if (NULL!=fp) {
        for (j=0; j<6; j++) {
            for (i=0; i<7; i++) {
                // fseek(fp, (long)(i*2 + j*(2*7+j)), SEEK_SET);
                // fread(&rank[j][i], 1, 1, fp);
                fscanf(fp, "%2d%c", &rank[j][i], &ch);
            }
        }
        fclose(fp);
        return SUCCESS;
    }
}

```

```

else {
    fclose(fp);
    return FAIL;
}
}

BOOL EditClassRank(SIGN rank[RANK_COL_EA][RANK_ROW_EA])
{
    int old_g_x;
    int old_g_y;
    int g_x;
    int g_y;

    int old_data_x = 0;
    int old_data_y = 0;
    int data_x = 0;
    int data_y = 0;

    int key;
    BOOL end_while = FALSE;

    const char rank_str[3][MAX_CHAR_NUM+1] = { " 이하 ", " 무시 "
" 이상 " };

    old_g_x = D_SX+GAP_X+ (old_data_x+1)*(WSIZE);
    old_g_y = D_CLIENT+(old_data_y+1)*HSIZE*11+GAP_Y*2;

    g_x = D_SX+GAP_X+ (data_x+1)*(WSIZE);
    g_y = D_CLIENT+(data_y+1)*HSIZE*11+GAP_Y*2;

    freedom_box(rank_str[rank[data_y][data_x]+1],
                g_x, g_y, WSIZE, HSIZE, NORMAL);

    do {
        key = getkey();

        if (NULL!=key) {

```

```

        switch (key) {
        case LEFTARROW:
            if (0!=data_x) {
                data_x--;
            }
            break;
        case RIGHTARROW:
            if (6!=data_x) {
                data_x++;
            }
            break;
        case UPARROW:
            if (0!=data_y) {
                data_y--;
            }
            break;
        case DOWNARROW:
            if (5!=data_y) {
                data_y++;
            }
            break;
        case ENTER:
            break;
        case ESC:
            end_while = TRUE;
        }

if ( (key!=ESC && key!=ENTER)
    && data_x!=old_data_x || data_y!=old_data_y) {
    g_x = D_SX+GAP_X+ (data_x+1)*(WSIZE);
    g_y = D_CLIENT+HSIZE*10+(data_y+1)*HSIZE+GAP_Y*2;
    freedom_box(rank_str[rank[old_data_y][old_data_x]+1],
                old_g_x, old_g_y, WSIZE, HSIZE, REVERSE)
    freedom_box(rank_str[rank[data_y][data_x]+1],
                g_x, g_y, WSIZE, HSIZE, NORMAL);
    old_data_x = data_x;
    old_data_y = data_y;
    old_g_x = g_x;
    old_g_y = g_y;
}

```

```

    )
    else if (ENTER==key) {
        edit_rank(&rank[data_y][data_x], g_x, g_y);
        freedom_box(rank_str[rank[data_y][data_x]+1],
                    g_x, g_y, WSIZE, HSIZE, NORMAL);
    }
}
} while (TRUE!=end_while);

return SUCCESS;

}

BOOL          SaveClassRank(const          SIGN
rank[RANK_COL_EA][RANK_ROW_EA])
{
    FILE *fp;
    int i, j;
    fp = fopen("Rank.dat", "r+");
    for (j=0; j<6; j++) {
        for (i=0; i<6; i++) {
            fprintf(fp, "%2d,", rank[j][i]);
        }
        fprintf(fp, "%2d\n", rank[j][6]);
    }
    fclose(fp);
    return SUCCESS;
}

BOOL ConvertIntegerObject(OBJECT *obj, unsigned long preObj[7])
{
    if (0!=obj->NUM) {

        preObj[0] = (unsigned long)(obj->LongLength * 100);
        preObj[1] = (unsigned long)(obj->ShortLength * 1000);
        preObj[2] = (unsigned long)(fabs(obj->Angle) * 1000);
    }
}

```

```

        preObj[3] = (unsigned long)(obj->r_thick * 1000);
        preObj[4] = (unsigned long)(obj->AreaH);
        preObj[5] = (unsigned long)(obj->Weight);
        preObj[6] = 0;

        return SUCCESS;
    }
    return FAIL;
}

BOOL PreObjectCellView(const unsigned long preObj[7])
{
    // int i;
    /* for (i=1; i<8; i++) {
        cell_data_view(preObj[i-1], D_SX+GAP_X+i*(WSIZE)
            D_CLIENT+HSIZE*9+GAP_Y);
    } */
    // for (i=1; i<8; i++) {

        real_cell_data_view((unsigned long)(preObj[0]/100),

D_SX+GAP_X+1*(WSIZE),

D_CLIENT+HSIZE*9+GAP_Y);
        real_cell_data_view((unsigned long)(preObj[1]/1000),

D_SX+GAP_X+2*(WSIZE),

D_CLIENT+HSIZE*9+GAP_Y);
        real_cell_data_view((unsigned long)(preObj[2]/1000),

D_SX+GAP_X+3*(WSIZE),

D_CLIENT+HSIZE*9+GAP_Y);
        real_cell_data_view((unsigned long)(preObj[3]/1000),

D_SX+GAP_X+4*(WSIZE),

```

```

D_CLIENT+HSIZE*9+GAP_Y);
    real_cell_data_view(preObj[4], D_SX+GAP_X+5*(WSIZE),

D_CLIENT+HSIZE*9+GAP_Y);
    real_cell_data_view(preObj[5], D_SX+GAP_X+6*(WSIZE),

D_CLIENT+HSIZE*9+GAP_Y);
// }
    return SUCCESS;
}

BOOL PreObjectRankView(const SIGN preObj[7])
{
    int i;
    for (i=1; i<8; i++) {
        rank_data_view(preObj[i-1], D_SX+GAP_X+i*(WSIZE),
            D_CLIENT+HSIZE*10+GAP_Y);
    }
    return SUCCESS;
}

BOOL RealPreObjectRankView(const char str[7][MAX_CHAR_NUM+1])
{
    int i;
    for (i=1; i<8; i++) {
        freedom_box(str[i-1], D_SX+GAP_X+i*(WSIZE),
            D_CLIENT+HSIZE*10+GAP_Y, WSIZE, HSIZE,
NORMAL);
    }
    return SUCCESS;
}

BOOL RelationObject(unsigned long preObj[CELL_ROW_EA],
                    const char
data[CELL_COL_EA][CELL_ROW_EA][MAX_CHAR_NUM+1],
                    const SIGN
rank[RANK_COL_EA][RANK_ROW_EA],
                    int* crops_class)

```

```

{
    int i, j;
    int pass_count = 0;
    unsigned long tableObj[CELL_ROW_EA];

    for (j=0; j<CELL_COL_EA-2; j++) {
        pass_count = 0;
        for (i=0; i<CELL_ROW_EA; i++) {
            cell_data_convert_dec(data[j][i], &tableObj[i])
// printf("%ld ", tableObj[i]);
        }
        for (i=0; i<RANK_ROW_EA; i++) {
            switch (rank[j][i]) {
            case MORE:
                if (preObj[i] > tableObj[i]) {
                    pass_count++;
                }
                break;
            case LESS:
                if (preObj[i] < tableObj[i]) {
                    pass_count++;
                }
                break;
            case NONE:
                pass_count++;
                break;
            default:
                pass_count++;
                break;
            }
        }
        if (CELL_ROW_EA==pass_count) {
            *crops_class = (j+1);
            return SUCCESS;
        }
    }
    *crops_class = 4;
    return FAIL;
}

```



```

BOOL cell_data_view(const unsigned long dec, int lx, int ly)
{
    char str[MAX_CHAR_NUM+1];
    if (dec <= 999999L) {
        sprintf(str, "%6ld", dec);
        hsetcolor(DARKGRAY); hsetbkcolor(LIGHTGRAY);
        hputspxy(lx+3, ly+2, str);
        return SUCCESS;
    }
    return FAIL;
}

```

```

BOOL real_cell_data_view(const unsigned long dec, int lx, int ly)
{
    char str[MAX_CHAR_NUM+1];
    if (dec <= 999999L) {
        sprintf(str, "%6ld", dec);
        hsetcolor(DARKGRAY); hsetbkcolor(LIGHTGRAY);
        hputspxy(lx+3, ly+2, str);
        return SUCCESS;
    }
    return FAIL;
}

```

```

BOOL cell_data_convert_dec(const char* origin, unsigned long* dec)
{
    int i=0;

    *dec = 0L;
    while ( '\0'!=origin[i] ) {
        if (('0'<= origin[i] && '9'>= origin[i])
            || (0x20 == origin[i]) ) {
            if ( 0x20 == origin[i] ) {
            }
            else if (MAX_CHAR_NUM < i) {
                GetFileMsg(msgBuf[0], MSG_CRACK_DATA);
                MessageView(msgBuf[0], "OR", "Data Overflow", "Error");
            }
        }
    }
}

```

```

        return FAIL;
    }
    else {
        *dec = (*dec)*10L + (unsigned long)(origin[i] - '0');
    }
    i++;
}
}
return SUCCESS;
}

BOOL rank_data_view(const SIGN rank_value, int lx, int ly)
{
    const char rank_str[3][MAX_CHAR_NUM+1] = { " 이하 ", " 무시",
" 이상 " };

    hsetcolor(DARKGRAY); hsetbkcolor(LIGHTGRAY);
    hputspxy(lx+3, ly+2, rank_str[rank_value+1]);
    return SUCCESS;
}

BOOL cell_data_convert_str(unsigned long* dec, char* newstr)
{
    sprintf(newstr, "%6ld", *dec);
    return SUCCESS;
}

BOOL edit_cell(char* data, int cell_x, int cell_y)
{
    unsigned long old_data = 0L;
    unsigned long new_data = 0L;
    int key;
    BOOL status = FALSE;
    BOOL end_while = FALSE;

    int i;

    status = cell_data_convert_dec(data, &old_data);

```

```

if (FAIL==status) {
    for (i=0; i<MAX_CHAR_NUM-1; i++) {
        data[i] = 0x20;
    }
    data[MAX_CHAR_NUM-1] = '1';
    data[MAX_CHAR_NUM] = '\0';

    status = cell_data_convert_dec(data, &old_data)
    MessageView("Reinit", "As Zero", "", "");
}

```

```
cell_data_view(old_data, cell_x, cell_y);
```

```
new_data = old_data;
```

```

do {
    key = getkey();
    if (NULL!=key && 1000000L > new_data) {
        switch (key) {
        case RIGHTARROW:
            new_data++;
            if (1000000L <= new_data) {
                new_data = 999999L;
            }
            break;
        case LEFTARROW:
            new_data--;
            if (1000000L <= new_data) {
                new_data = new_data + 100L;
            }
            break;
        case UPARROW:
            new_data += 100L;
            if (1000000L <= new_data) {
                new_data = 999999L;
            }
        }
    }
}

```

```

        break;
    case DOWNARROW:
        new_data -= 100L;
        if (1000000L <= new_data) {
            new_data = new_data + 100L;
        }
        break;
    case PGUPKEY:
        new_data += 1000L;
        if (1000000L <= new_data) {
            new_data = 999999L;
        }
        break;
    case PGDNKEY:
        new_data -= 1000L;
        if (1000000L <= new_data) {
            new_data = new_data + 100L;
        }
        break;
    case ENTER:
        status = cell_data_convert_str(&new_data, data)
        end_while = TRUE;
        break;
    case ESC:
        status = cell_data_convert_str(&old_data, data);
        status = FALSE;
        end_while = TRUE;
        break;
    }
    cell_data_view(new_data, cell_x, cell_y);
}
else if (new_data > 2000000L) {
    new_data = 0L;
}
} while (TRUE != end_while);

return status;
}

```

```

BOOL edit_rank(SIGN* data, int cell_x, int cell_y)
{
    BOOL old_data = *data;
    BOOL new_data = 0;
    int key;
    BOOL status = FALSE;
    BOOL end_while = FALSE;

    rank_data_view(old_data, cell_x, cell_y);
    new_data = old_data;

    do {
        key = getkey();
        if (NULL!=key) {
            switch (key) {
                case RIGHTARROW:
                    new_data = MORE;
                    break;
                case LEFTARROW:
                    new_data = LESS;
                    break;
                case UPARROW:
                case DOWNARROW:
                    new_data = NONE;
                    break;
                case ENTER:
                    *data = new_data;
                    end_while = TRUE;
                    status = TRUE;
                    break;
                case ESC:
                    status = FALSE;
                    end_while = TRUE;
                    break;
            }
            rank_data_view(new_data, cell_x, cell_y)
        }
    } while (TRUE != end_while);
}

```

```
    return status;
}
```

```
BOOL load_cell(char* data, int cell_x, int cell_y)
{
    FILE *fp;
    fp = fopen("Class.dat","r");
    if (NULL!=fp) {
        fseek(fp,
              (long)( cell_x*(MAX_CHAR_NUM+1)
                    + cell_y*(MAX_CHAR_NUM+1)*7 + cell_y ), SEEK_SET);
        fread(data, MAX_CHAR_NUM, 1, fp);
        data[MAX_CHAR_NUM] = '\0';
        fclose(fp);
        return SUCCESS;
    }
    else {
        fclose(fp);
        return FAIL;
    }
}
```

```
BOOL save_cell(const char* data, int cell_x, int cell_y)
{
    FILE *fp;
    fp = fopen("Class.dat","r+");
    if (NULL!=fp) {
        fseek(fp,
              (long)( cell_x*(MAX_CHAR_NUM+1)
                    + cell_y*(MAX_CHAR_NUM+1)*7 + cell_y ), SEEK_SET);
        fwrite(data, MAX_CHAR_NUM, 1, fp);
        fclose(fp);
        return SUCCESS;
    }
    else {
        fclose(fp);
        return FAIL;
    }
}
```

```

    }
}

/*****
/*          Machine Life Cycle List          */
*****/

BOOL Process1(void)
{
    BOOL status = FALSE;

    Port1 = ~ADDR_WAIT_LIGHT&Port1;//0xfe&Port1;
    Port1 = ADDR_READY_LIGHT|Port1;
    outportb(MACHINE_PORT,Port1);

    while (FAIL==status) {
        CopyCameraBuffer(odxDrvFlag);
        status = InterActionCell(&Cell[0][0].p[0]);
        if (FAIL==status) {
            MessageView("오이가", "없네요!", "", "");
        }
        if (ESC==getkey()) {
            status = FAIL;
            return FAIL;
        }
    }

    Port1 = ~ADDR_WAIT_LIGHT&Port1;//0xfe&Port1;
    Port1 = ADDR_READY_LIGHT|Port1;
    Port1 = ~ADDR_READY_LIGHT&Port1;
    Port1 = ADDR_WAIT_LIGHT|Port1;
    outportb(MACHINE_PORT,Port1);

    if (status==SUCCESS &&
SUCCESS==ConvertIntegerObject(&OeiObject, convert_int_object)) {
        PreObjectCellView(convert_int_object);

        RelationObject(convert_int_object,
            record_value,

```

```

        record_rank, &CropsClass);
// PreObjectRankView(record_rank[CropsClass]);
RealPreObjectRankView(unit_name);

        freedom_box(class_name[CropsClass-1], D_SX+GAP_X,
                    D_CLIENT+HSIZE*9+GAP_Y, WSIZE, HSIZE
REVERSE);
        freedom_box(class_name[CropsClass-1], D_SX+GAP_X,
                    D_CLIENT+HSIZE*10+GAP_Y, WSIZE, HSIZE
REVERSE);

        StoreClass[0] = CropsClass;
// printf("[%d %d] ", StoreClass[0], StoreClass[1]);
Process2();
status = Wrapping(StoreClass[1]);
if (FAIL == status) {
    return FAIL;
}
// Wrapping(-1);
StoreClass[1] = StoreClass[0];
return status;
}

return FAIL;
}

BOOL Process2(void) {
// Port1 = 0;
switch (StoreClass[1]) {
    case -1:
        break;
    case 1:
        Port1 = 0x0f&Port1;
        Port1 = ADDR_CLASS1_GATE|Port1;
// outportb(MACHINE_PORT, 0x40);
        break;
    case 2:
// Port1 = ADDR_ALLOW_WRAP_ON|Port1;
        Port1 = 0x0f&Port1;

```



```

        Port1 = ADDR_CLASS2_GATE|Port1;
//      outportb(MACHINE_PORT, 0x40);
        break;
    case 3:
        Port1 = 0x0f&Port1;
//      Port1 = ADDR_CLASS3_GATE|Port1;
        break;
    default:
        Port1 = 0x0f&Port1;
//      Port1 = ADDR_ALLOW_WRAP_OFF|Port1;
        Port1 = ADDR_OUTCLASS_GATE|Port1;
//      outportb(MACHINE_PORT, 0x20);
        break;
    }
    outportb(MACHINE_PORT,Port1);
    return SUCCESS;
}

BOOL Process3(void)
{
    /*          while(          ADDR_INIT_SWITCH          !
(inportb(MACHINE_ERROR_PORT)) & ADDR_INIT_SWITCH) ) {
        delay(READY_ACCESS_TIME*1000);
    }
    */
    return SUCCESS;
}

BOOL Process4(void) {
//  delay(LIFE_TIME*1000);
/*  int i;
    for (i=0; i<20; i++) {
        CopyCameraBuffer(odxDrvFlag);
    }
    */
    return SUCCESS;
}

BOOL Process5(void) {

```

```

// Port1 = ADDR_READY_LIGHT;
// Port2 = 0;
// outportb(MACHINE_PORT,Port1);
// outportb(MACHINE_ERROR_PORT,Port2);
return SUCCESS;
}

BOOL LifeCycle(void) {
    BOOL status[5] = { OFF, OFF, OFF, OFF, OFF }
    int i;

// ViewBmp("skku600.bmp", 10, 320, 280, 128);

    Port1 = ADDR_WAIT_LIGHT;
    outportb(MACHINE_PORT,Port1);
    if (FAIL==InitWrapping()) {
        outportb(MACHINE_ERROR_PORT,
ADDR_FATAL_ERROR_LIGHT);
        return FAIL;
    }

    NextFlag = 0;
    Port1 = ADDR_READY_LIGHT;
    Port2 = 0;

    StoreClass[0] = -1;
    StoreClass[1] = -1;
    CropsClass = -1;

    do {
//      Clear();
//      Port1 = ADDR_READY_LIGHT;
//      Port2 = 0;

        status[0] = Process1();
        if (FAIL==status[0]) {
            return FAIL;
        }
//      status[1] = Process2();

```

```

        status[2] = Process3();
        status[3] = Process4();
        status[4] = Process5();

        for (i=0; i<5; i++) {
            status[i] = OFF;
        }
        NextFlag = 1;
    } while(ESC!=getkey());
    return SUCCESS;
}

BOOL InitWrapping(void)
{
    int inb = 0;
    clock_t start, d_time, end;
    const float bnd_time = 15.*CLK_TCK;

    start = clock();
    /* Init Machine */
    outportb(porta,mot2ccw);    /* close the rap bar */
    delay(320);
    outportb(porta,0);

    outportb(porta,mot1ccw);    /* rotate the module at rap start
point */
    inb=inportb(portb);

    while (0==(0x02&inb)) {
        inb=inportb(portb);
        end = clock();
        d_time = (end - start);
        if (bnd_time < d_time) {
            return FAIL;
        }
    }
    delay(10);
}

```

```

        outportb(porta,mot1c);
        delay(20);
        outportb(porta,0x00);
        delay(500);
        outportb(porta,mot2c);      /* open rap bar a little for catch
the rap */

        while(0==(0x01&inb)) {
            inb=inportb(portb);
            end = clock();
            d_time = (end - start);
            if (bnd_time < d_time) {
                return FAIL;
            }
        }
        delay(300);
        outportb(porta,0x00);

        outportb(porta,mot1c);      /* rotate the module with rap */
        inb=inportb(portb);

        outportb(porta,0x00);

        outportb(porta,mot1c);      /* rotate the module to cutting the
rap */
        inb=inportb(portb);

        while(0==(0x10&inb)) {
            inb=inportb(portb);
            end = clock();
            d_time = (end - start);
            if (bnd_time < d_time) {
                return FAIL;
            }
        }
        outportb(porta,mot1ccw); /* break the module */
        delay(50);

        outportb(porta,0);

```

```

        outportb(porta,mot2ccw);
        delay(200);
        outportb(porta,0);
        /*
            catch the time to get the oi
        */
    return SUCCESS;
}

BOOL Wrapping(int oei_class)
{
    int inb=0;

    clock_t start, d_time, end;
    const float bnd_time = 15.*CLK_TCK;

    switch (oei_class) {
        case 4:
            outportb(porta,mot1ccw);
            delay(800);
            outportb(porta,mot1c); /* rotate the module to receive
position for out oi */

            inb=inportb(portb);
            while(0==(0x10&inb)) {
                inb=inportb(portb);
                end = clock();
                d_time = (end - start);
                if (bnd_time < d_time) {
                    return FAIL;
                }
            }
            outportb(porta,mot1ccw); /* break the module */
            delay(60);
    }
}

```

```

outportb(porta,0);

delay(10000);
break;

/* Non process */
case -1:
    delay(10000);
    break;

case 1: case 2: case 3:
    start = clock();
    /*
        catch the time to get the oi
    */
    outportb(porta,mot2ccw); /* close the rap bar with oi */

    inb=inportb(portb);
    while(0==(0x20&inb)) {
        inb=inportb(portb);
        end = clock();
        d_time = (end - start);
        if (bnd_time < d_time) {
            return FAIL;
        }
    }
    outportb(porta,mot2c);
    delay(30);
    outportb(porta,0);
    outportb(porta,mot1ccw);
    delay(300);
    outportb(porta,0);
    outportb(porta,mot3c);
    delay(1600);

```

```

        outportb(porta,0);

        outportb(porta,mot1c); /* rotate the module to receive
position for out oi */

        inb=inportb(portb);
        while(0==(0x10&inb)) {
            inb=inportb(portb);
            end = clock();
            d_time = (end - start);
            if (bnd_time < d_time) {
                return FAIL;
            }
        }
        outportb(porta,mot1ccw); /* break the module */
        delay(60);

        outportb(porta,0);

        outportb(porta,mot2c); /* open rap bar a little for catch
the rap */

        while(0==(0x01&inb)) {
            inb=inportb(portb);
            end = clock();
            d_time = (end - start);
            if (bnd_time < d_time) {
                return FAIL;
            }
        }
        outportb(porta,0x00);
        break;
    )

    if (1==oei_class || 2==oei_class || 3==oei_class) {
        if (FAIL==InitWrapping()) {
            outportb(MACHINE_ERROR_PORT,
ADDR_FATAL_ERROR_LIGHT);
            return FAIL;
        }
    }

```

```
    }  
  }  
  
  return SUCCESS;  
}
```


B. 포장장치 및 일관작업의 실험자료

표 2 포장작업 및 일관작업 성능 (1회/10회)

등급	오이면적 (mm ²)	오이의 직선길이 (mm)	오이의 곡선길이 (mm)	오이의 휨각도 (°)	굽은 정도 길이 L (mm)	두께 최대 값 (mm)	두께 최소 값 (mm)	포장 가능 여부 (OX)	분리 장치 작동 성공 여부 (OX)	일관 작업 성공 여부 (OX)
X	6633	212.11	227.17	141.48	37.06	46.81	28.07	X	O	O
X	6624	211.28	224.98	143.36	34.98	65.78	29.00	X	O	O
X	6637	213.05	230.34	140.54	38.20	47.43	30.00	X	O	O
X	6686	211.91	227.25	141.00	37.52	48.70	30.00	X	O	O
X	6701	212.07	227.31	142.76	35.73	56.30	30.00	X	O	O
X	8617	255.01	271.45	142.76	42.96	46.17	32.24	X	O	O
X	8635	255.23	272.20	142.65	43.13	43.96	34.00	X	O	O
X	8453	254.15	269.51	143.84	41.49	53.14	30.06	X	O	O
X	8520	255.38	271.86	143.47	42.14	44.92	33.00	X	O	O
X	8359	253.04	267.35	141.45	44.24	58.87	30.06	X	O	O

표 2 포장작업 및 일관작업 성능 (2회/10회)

등급	오이면적 (mm ²)	오이의 직선길이 (mm)	오이의 곡선길이 (mm)	오이의 휨각도 (°)	굽은 정도 길이 L (mm)	두께 최대 값 (mm)	두께 최소 값 (mm)	포장 가능 여부 (OX)	분리 장치 작동 성공 여부 (OX)	일관 작업 성공 여부 (OX)
X	9135	255.43	270.07	140.60	45.73	51.24	32.00	X	O	O
X	9061	256.85	272.25	142.18	43.99	46.57	34.00	X	O	O
X	9051	256.34	271.10	141.53	44.72	55.36	34.23	X	O	O
X	9063	256.87	271.64	143.33	42.56	49.33	34.13	X	O	O
X	9063	256.03	270.61	141.79	44.34	47.80	34.00	X	O	O
AL	7307	244.4	247.45	161.55	19.85	34.00	29.27	O	O	O
BI	7444	245.34	248.17	161.34	20.15	35.00	30.00	O	O	O
AL	7391	245.35	248.33	162.40	18.99	35.51	32.24	O	O	O
AL	7360	244.52	247.23	161.68	19.71	36.22	30.00	O	O	O
AL	7333	243.52	246.31	163.11	18.07	36.05	27.01	O	O	O

표 3 포장작업 및 일관작업 성능 (3회/10회)

등급	오이면적 (mm ²)	오이의 직선길이 (mm)	오이의 곡선길이 (mm)	오이의 휨각도 (°)	굽은 정도 길이 L (mm)	두께 최대 값 (mm)	두께 최소 값 (mm)	포장 가능 여부 (OX)	분리 장치 작동 성공 여부 (OX)	일관 작업 성공 여부 (OX)
BI.	7576	248.29	256.25	153.13	29.65	35.69	31.57	O	O	O
BI.	7506	248.69	255.73	154.03	28.67	37.36	31.06	O	O	O
BI.	7448	247.13	252.28	156.51	25.69	42.72	30.00	O	O	O
BI.	7614	248.88	255.55	155.69	26.80	35.90	33.00	O	O	O
X	7595	249.09	256.74	154.83	27.80	42.05	31.06	O	O	O
X	8956	241.03	259.29	140.38	43.41	56.30	33.00	X	O	O
X	8972	241.03	259.20	140.38	43.41	54.70	34.13	X	O	O
X	8796	241.01	259.68	139.64	44.28	81.27	30.06	X	O	O
X	8883	240.67	263.49	137.02	47.38	54.40	34.05	X	O	O
X	8854	239.67	262.37	137.11	47.07	52.20	34.13	X	O	O

표 4 포장작업 및 일관작업 성능 (4회/10회)

등급	오이면적 (mm ²)	오이의 직선길이 (mm)	오이의 곡선길이 (mm)	오이의 휨각도 (°)	굽은 정도 길이 L (mm)	두께 최대 값 (mm)	두께 최소 값 (mm)	포장 가능 여부 (OX)	분리 장치 작동 성공 여부 (OX)	일관 작업 성공 여부 (OX)
AL	6358	223.27	225.45	164.61	15.08	36.67	27.07	O	O	O
AL	6042	224.08	226.68	164.54	15.21	34.52	26.30	O	O	O
AL	6052	224.08	226.47	165.61	14.14	33.54	26.07	O	O	O
AL	6210	223.36	225.81	163.61	16.08	44.59	26.01	O	O	O
AL	6190	224.26	226.29	165.26	14.50	33.06	26.68	O	O	O
AL	7873	258.93	261.89	164.47	17.65	36.00	28.16	O	O	O
AL	7822	258.32	261.50	163.25	19.02	37.85	27.01	O	O	O
AL	7769	258.00	261.04	163.33	18.90	35.22	28.01	O	O	O
AL	7763	258.00	261.16	163.33	18.90	35.05	28.01	O	O	O
AL	7901	257.59	260.17	164.89	17.08	47.12	31.57	O	O	O

표 5 포장작업 및 일관작업 성능 (5회/10회)

등급	오이면적 (mm ²)	오이의 직선길이 (mm)	오이의 곡선길이 (mm)	오이의 꺾임각도 (°)	굽은 정도 길이 L (mm)	두께 최대 값 (mm)	두께 최소 값 (mm)	포장 가능 여부 (OX)	분리 장치 작동 성공 여부 (OX)	일관 작업 성공 여부 (OX)
AI.	7236	241.20	243.03	167.06	13.68	37.10	30.00	O	O	O
AI.	7352	239.87	241.39	169.24	11.29	39.05	34.05	O	O	O
AL	7124	241.40	243.61	166.91	13.84	35.35	29.00	O	O	O
AL	7279	242.01	244.07	168.50	12.18	36.87	32.24	O	O	O
AI.	7250	237.00	239.02	164.75	15.86	53.48	33.54	O	O	O
BL	8430	252.62	268.22	145.05	39.76	53.48	30.06	X	O	O
BL	8542	251.50	266.81	145.11	39.52	55.36	31.06	X	O	O
BL	8386	252.72	268.00	145.17	39.64	49.04	33.24	X	O	O
X	8314	253.12	268.48	144.84	40.10	43.01	32.00	X	O	O
X	8200	252.83	268.72	144.83	40.06	53.48	29.27	X	O	O

표 4 포장작업 및 일관작업 성능 (6회/10회)

등급	오이면적 (mm ²)	오이의 직선길이 (mm)	오이의 곡선길이 (mm)	오이의 휨각도 (°)	굽은 정도 길이 L (mm)	두께 최대 값 (mm)	두께 최소 값 (mm)	포장 가능 여부 (OX)	분리 장치 작동 성공 여부 (OX)	일관 작업 성공 여부 (OX)
AM	5915	207.50	211.72	158.51	19.69	41.10	29.42	O	O	O
BM	5824	209.16	213.99	157.50	20.80	39.84	27.00	O	O	O
BM	5887	210.72	215.85	157.32	21.13	46.48	29.00	O	O	O
AM	5650	207.28	211.93	158.24	19.92	35.17	26.30	O	O	O
AM	5564	207.58	211.57	158.24	19.94	34.48	25.31	O	O	O
BL	7218	248.39	257.44	151.17	31.92	38.60	26.00	O	O	O
BL	7263	248.24	256.37	151.06	32.03	53.14	28.07	O	O	O
BL	7300	248.20	256.10	155.04	27.46	36.40	28.16	O	O	O
BL	7240	247.95	255.42	153.60	29.07	46.81	27.16	O	O	O
BL	7469	247.42	255.44	154.92	27.51	46.48	31.00	O	O	O

표 7 포장작업 및 일관작업 성능 (7회/10회)

등급	오이면적 (mm ²)	오이의 직선길이 (mm)	오이의 곡선길이 (mm)	오이의 휨각도 (°)	굽은 정도 길이 L (mm)	두께 최대 값 (mm)	두께 최소 값 (mm)	포장 가능 여부 (OX)	분리 장치 작동 성공 여부 (OX)	일관 작업 성공 여부 (OX)
AI.	6692	234.26	236.44	165.03	15.39	38.83	28.07	O	O	O
AI.	6653	235.13	237.48	165.78	14.66	34.13	26.07	O	O	O
AI.	6673	234.79	237.00	166.47	13.93	36.87	28.07	O	O	O
AI.	6674	235.25	237.51	166.09	14.35	35.05	26.07	O	O	O
AI.	6670	235.21	237.35	165.68	14.77	36.13	26.00	O	O	O
BI.	7927	255.78	267.85	148.97	35.50	45.88	30.00	O	O	O
BI.	8066	256.42	269.52	148.36	36.33	41.23	31.00	O	O	O
BI.	7966	254.60	266.71	145.53	39.49	59.13	29.42	O	O	O
BI.	8043	256.03	269.54	147.78	36.97	40.26	32.06	O	O	O
BI.	7848	256.13	268.41	146.73	38.26	48.38	26.07	O	O	O

표 8 포장작업 및 일관작업 성능 (8회/10회)

등급	오이면적 (mm ²)	오이의 직선길이 (mm)	오이의 곡선길이 (mm)	오이의 휨각도 (°)	굽은 정도 길이 L (mm)	두께 최대 값 (mm)	두께 최소 값 (mm)	포장 가능 여부 (OX)	분리 장치 자동 성공 여부 (OX)	일관 작업 성공 여부 (OX)
BL	7277	250.04	262.08	147.54	36.39	34.92	27.00	O	O	O
BL	6927	248.92	260.21	146.76	37.15	42.05	26.07	O	O	O
BL	6883	249.64	261.64	146.17	37.96	37.94	26.00	O	O	O
BL	7019	249.32	260.17	148.33	35.36	39.84	26.07	O	O	O
BL	6797	249.16	260.22	148.65	34.96	35.17	26.01	O	O	O
BL	6301	229.10	240.72	147.02	33.91	36.13	25.07	O	O	O
BL	6309	259.10	240.85	147.02	38.35	35.73	27.29	O	O	O
BL	6226	227.98	238.31	147.75	32.96	38.89	25.07	O	O	O
BL	6204	228.35	238.74	147.32	33.47	40.80	25.00	O	O	O
BL	6282	228.32	238.84	147.37	33.41	52.20	26.07	O	O	O

표 9 포장작업 및 일관작업 성능 (9회/10회)

종 급	오이면 적 (mm ²)	오이의 직선길 이 (mm)	오이의 곡선길 이 (mm)	오이의 휘각 도 (°)	굽은 정도 길이 L (mm)	두께 최대 값 (mm)	두께 최소 값 (mm)	포장 가능 여부 (OX)	분리 장치 작동 성공 여부 (OX)	일관 작업 성공 여부 (OX)
Bl.	6701	240.01	244.39	158.33	22.97	38.89	25.01	O	O	O
Bl.	6833	238.20	245.80	154.89	26.52	40.16	30.26	O	O	O
Bl.	7137	246.05	255.21	153.36	29.13	50.92	31.00	O	O	O
Bl.	6572	239.16	243.96	158.83	22.35	34.92	26.00	O	O	O
Bl.	6749	241.50	247.53	156.75	24.84	33.24	30.00	O	O	O
AM	5426	213.41	217.10	160.88	17.97	37.64	22.00	O	O	O
AM	5445	212.94	215.02	161.77	17.08	32.00	26.30	O	O	O
AM	5557	212.49	214.90	161.01	17.76	34.53	30.00	O	O	O
AM	5539	210.76	213.77	161.91	16.78	32.28	29.27	O	O	O
AM	5609	210.64	214.07	160.73	17.88	39.21	29.00	O	O	O

표 10 포장사업 및 인관작업 성능 (10회/10회)

등급	오이면적 (mm ²)	오이의 직선길이 (mm)	오이의 곡선길이 (mm)	오이의 휨각도 (°)	굽은 정도 길이 L (mm)	두께 최대값 (mm)	두께 최소값 (mm)	포장 가능 여부 (OX)	분리 장치 작동 성공 여부 (OX)	인관 작업 성공 여부 (OX)
Bl.	7327	251.00	258.25	154.30	28.63	35.90	28.01	O	O	O
Bl.	7556	250.69	257.18	156.37	26.22	50.92	33.31	O	O	O
Bl.	7040	250.45	257.75	154.90	27.88	330.37	29.06	O	O	O
X	7026	255.00	270.51	141.67	44.32	37.64	24.08	X	O	O
Bl.	7084	249.81	255.86	154.89	27.82	44.59	26.01	O	O	O
X	9259	256.11	269.93	145.05	40.31	55.97	37.05	X	O	O
X	9437	256.32	270.27	142.13	42.08	48.08	35.00	X	O	O
X	9146	256.63	270.70	142.01	44.17	46.84	30.00	X	O	O
X	9188	257.03	271.76	141.66	44.68	47.12	35.00	X	O	O
X	9129	256.03	270.74	140.56	45.89	49.33	33.06	X	O	O

본문 작성 요령

1. 본문의 순서는 장, 절, 1, 가, (1), (가), ①, ㉠등으로 하고,

- 장은 3호 고딕체
- 절은 4호 명조체
- 항은 5호 명조체
- 본문은 5호 명조체로 한다.

단, 본문의 내용중 중요부문은 고딕체로 사용할 수 있다.

2. 장은 원칙적으로 페이지를 바꾸어 시작한다.

3. 본문은 5호 횡으로 작성한다.

4. 1행은 35자, 1단은 25행으로 한다.

5. 페이지 번호는 하단 중앙끝에 5호 활자로 한다.

6. 각 주(註)는 해당 페이지 하단에 6호 활자로 표기하여 본문과 구분토록 한다.

7. 페이지수는 편집순서 2의 제출문부터 시작한다.

단, 삽입물이 있을 때는 그 삽입물의 크기에 불문하고 1면을 한 페이지 하여 일련번호를 붙인다.

8. 한글, 한문, 영문을 혼용한다.

9. 뒷면지에는 주의문을 넣는다.

[부 표]

인 쇄 내 용

주 의

1. 이 보고서는 농림수산부에서 시행한 농림수산특정연구사업의 연구보고서입니다.
2. 이 보고서 내용을 발표할 때에는 반드시 농림수산부에서 시행한 농림수산특정연구사업의 연구결과임을 밝혀야 합니다.
3. 국가과학기술 기밀유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 아니됩니다.

I. 인쇄규격

1. 크 기 : 4×6배판(가로 188mm×세로 275mm)
2. 제 본 : 좌철
3. 용 지 : ○ 표지 200g/m. 양면 아트지
○ 내용 80Kg/m. 모조지
4. 인쇄방식
 - 1) 표 지 : 바탕 백색, 활자 흑색
 - 2) 내 용 : 흑색 지정활자로 인쇄한다.

II. 편집순서

1. 표 지
2. 제 출 문
3. 요 약 문
4. Summary
5. Contents
6. 목 차
7. 본 문
8. 뒷 면 지

III. 참고사항

- 공판인쇄시에는 이 요령에 준한다.