

# 영상처리를 이용한 토마토 생육량 측정 시스템

김강섭\*, 홍영기\*, 김현중\*, 김국환\*, 김경철\*, 이명훈<sup>o</sup>

## Tomato Growth Rate Measurement System Using Image Processing

Kangsub Kim\*, Youngki Hong\*, Hyunjong Kim\*, Gookhwan Kim\*,  
Kyoungchul Kim\*, Meonghun Lee<sup>o</sup>

### 요약

농업에서 작물에 대한 품질을 높이는 작업은 중요하다. 스마트팜은 정보통신기술을 이용해 농작물의 생육 환경을 최적으로 유지하는 농업 시스템으로, 스마트팜을 이용하여 생산량 증가와 품질의 상승을 통해 시간, 노동력 등을 줄일 수 있다. 본 논문에서는 영상처리 방법을 사용하여 토마토의 생육량을 측정하고, 이를 이용해 작물이 올바르게 자라고 있는지 판단을 자동으로 하는 시스템을 구축한다. 토마토 생육량 측정 시스템의 단계로는 HSV 색상 모델을 이용해 색상 범위를 지정하고, 이 범위를 사용하여 블러링과 잡음 제거를 통해 해당 객체를 검출하고, 검출된 객체들을 통해 생장 길이를 계산한다. 또한 생장 길이를 계산하는데 중요한 역할을 하는 참조를 설정하여 여러 실험을 통해 최적의 참조를 찾아낸다. 앞으로 다양한 환경과 작물에서 취득한 데이터를 기반으로 측정 시스템을 더욱 발전해 나간다면 보다 농업 환경에 강건한 측정 시스템이 될 것이다.

**키워드** : 영상처리, 토마토, 생육량, 참조모델

**Key Words** : Image Processing, Tomato, Growth Rate, Reference Model

### ABSTRACT

It is important to improve the quality of agriculturally produced crops. Smart Farm is an agricultural system that optimally maintains the growth environment of crops using information and communication technology (ICT). Using smart farms, time and labor can be reduced by increasing production and increasing quality. In this paper, we use a Image processing technique to measure the growth rate of tomato, and use this to build a system that automatically determines whether crops are growing correctly. The step of Tomato Growth Measurement System are using HSV color model to specify a color range, detecting the object through blurring and noise reduction, and the growth length is calculated from the detected objects. In addition, a number of experiments are conducted to find the optimum reference, which plays an important role in calculating the growth length. In the future, if the measurement system is further developed based on data acquired in various environments and crops, it will be a more robust measurement system in agricultural environments.

※ 본 연구는 농촌진흥청 연구사업(과제번호: PJ013891012020) 지원으로 이루어진 것임.

• First Author : National Institute of Agricultural Sciences, rs1704@korea.kr, 연구원, 정회원

◦ Corresponding Author : National Institute of Agricultural Sciences, leemh5544@korea.kr, 연구사, 정회원

\* National Institute of Agricultural Sciences, sanm70@korea.kr, 연구관, hyunjongnt@korea.kr, 연구관, 정회원; meceng93@korea.kr, 연구사; kkcmmole@korea.kr, 연구사

논문번호 : 202006-120-0-SE, Received June 2, 2020; Revised June 23, 2020; Accepted June 26, 2020

## I. 서론

최근 스마트폰의 발전에 따라 영상 데이터가 손쉽게 취득된다. 이에 따라 영상 데이터의 양이 폭발적으로 증가하고, 이를 기반으로 정보 통신 기술과 영상 분야의 많은 방법들이 융합되어 발전해왔다. 영상을 활용한 최신 정보통신기술(ICT)과 농업기술이 융합된 스마트팜(Smart Farm)은 농가의 환경을 자동으로 관리 및 제어하는 지능화된 농장 시스템이다. 스마트팜의 연구가 최근 활발히 이루어지고 있으며<sup>1,2)</sup>, 그 중 영상을 활용한 생육량 측정 및 분석은 작물의 품질과 연관되기 때문에 많은 관심이 집중되고 있다.

생육량 측정은 측정된 작물의 생장 길이를 통해 작물이 올바르게 자라고 있는지 판단 할 수 있으며, 이후 작물에 대한 상황을 판단할 지표가 된다. 기존의 생육량 측정 시스템은 사람이 직접 하나씩 측정하는 작업이 진행되었다. 이에 영상을 이용한 생육량 측정 시스템이 개발되었고 그림 1과 같이 지난 주 생장점(n-1), 이번 주 생장점(n), 꽃 핀 화방의 줄기에 스티커를 붙인 뒤 인공지능 기술<sup>3)</sup>을 이용해 이를 찾아내는 방법이 개발되었다. 하지만 이는 스티커를 100% 찾아낼 수 없어서 생장점 사이의 거리를 매번 측정하기 힘들다. 또한 찾아낸 스티커 사이의 거리를 계산하기 위해 참조(Reference)를 함께 영상에 담아야 한다. 하지만 그림 1의 참조는 너무 커서 불편하다는 문제점이 발생한다. 결국, 기존의 생육량 측정 시스템은 정확도와 편의성이 현저히 떨어진다.

본 논문에서는 토마토 영상을 사용하여 지난 주(n-1) 생장점에서 이번 주(n) 생장점까지 사이의 거리를 계산한다. 이를 위해서 토마토를 재배하는 온실에서 생장점 위치에 빨간색 스티커를 부착하고, 정확도와 편의성이 보장되는 참조를 만들어 영상을 취득한다. 영상 처리에 많이 사용되는 블러링(Blurring)을 통해 잡음(Noise)을 제거하고 정확한 객체를 검출해서 측정 시스템의 편의성과 정확도를 높이는 방법을 제안한다.



그림 1. 기존 생육량 측정 시스템  
Fig. 1. Existing growth rate measurement system

## II. 관련 연구

본 장에서는 색상 모델들에 대해서 알아보고, 윤곽선을 검출하기 위한 영상처리 방법들과 객체를 검출하는 방법들에 대해 알아본다.

### 2.1 색상 모델

색상 모델에는 Gray 모델, RGB 모델, HSV 모델 등이 있다. 우선 Gray 모델은 색 정보를 사용하지 않고, 검정색을 표현하는 0부터 흰색을 표현하는 255까지 총 256단계의 밝기로 영상 픽셀 값을 표현한다. 다음으로 RGB 모델은 가장 기본적인 색상 모델로서 색을 빨간(Red), 초록(Green), 파란(Blue)의 성분으로 이루어져있다. RGB 모델에서 검은색은 모든 성분의 값이 0으로 나타내며 흰색은 모든 값이 255로 나타낸다. 각 성분의 값은 각각 0에서 255까지 총 256 단계를 가지며, 이 모델을 사용한다면 총  $256^3 = 16,777,216$  가지의 색을 표현 할 수 있다.

HSV 모델<sup>4)</sup>은 색조, 채도, 명도의 성분으로 색이 표현된다. 우선 색조(Hue)는 색의 계열을 나타내며, 하나의 색상이 푸른색 계열인지 붉은색 계열인지를 표현한다. 채도(Saturation)는 색이 얼마나 선명한 색인지를 나타내며, 명도(Value)는 밝기를 나타낸다. 영상처리에서 HSV 모델을 사용 할 때, 각각 값들은 0에서 255까지로 표현된다. 색조의 값은 색의 종류를 나타내기 때문에 해당 값은 단순한 순차(Index)를 나타내며, 채도 값은 0이면 무채색 255면 가장 선명한 색을 나타내며, 명도 값은 작을수록 어둡고 클수록 밝은 색임을 나타낸다. HSV 색상 모델은 그림 2에서와 같이 원뿔(Conic)의 형태와 원기둥(Cylindric)의 형태가 있으며 주로 사용되는 형태는 원뿔모양이다.

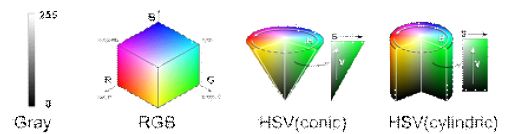


그림 2. 기본적인 색상모델  
Fig. 2. Basic Color Model

### 2.2 영상 처리

입력된 영상을 이용하여 원하는 결과를 추출하는 것은 쉬운 일이 아니다. 따라서 영상처리 방법으로 전처리를 하고 특징을 추출, 인식, 결정하는 단계가 필요하다<sup>5)</sup>. 본 절에서는 전처리에 필요한 여러 가지 영상 처리 방법들에 대해 알아본다.

2.2.1 영상 이진화(Binarization)

영상처리에서 이진화는 임계값(Threshold)을 기준으로 밝은 픽셀들은 모두 흰색으로 어두운 픽셀들은 검은색으로 바꾸는 것을 말한다. 이를 이용하여 스캔된 책의 영상에서 글자만 뚜렷하게 추출이 가능하고, 영상파일의 용량도 줄어드는 장점이 있다. 하지만 임계값 설정에 따라 좋은 결과가 나오거나 그렇지 못한 경우가 존재하기 때문에 임계값 설정은 중요하다. 임계값 설정에 대한 여러 가지 알고리즘 중에서 통계적인 방법을 이용한 Otsu 이진화 알고리즘<sup>6)</sup>이 가장 자연스러운 임계값 설정의 기준을 가진다. Otsu 이진화 방법은 임계값  $T$ 를 기준으로 영상 픽셀들을 두 클래스(Class)로 분류했을 때, 클래스 내 분산(Intra class variance)를 최소화하거나 또는 클래스 간 분산(Inter class variance)를 최대화하는  $T$ 를 찾는 방법이다. 입력 영상에서  $T$ 보다 어두운 픽셀들의 비율을  $\alpha$ , 밝기 평균을  $\mu_1$ , 분산을  $\sigma_1^2$ ,  $T$ 보다 밝은 픽셀들의 비율을  $\beta$ , 밝기 평균을  $\mu_2$ , 분산을  $\sigma_2^2$ 라고 했을 때 ( $\alpha + \beta = 1$ ), 클래스내 분산과 클래스 간 분산은 다음과 같다.

$$intra-classvariance = \alpha\sigma_1^2 + \beta\sigma_2^2 \quad (1)$$

$$inter-classvariance = \alpha\beta(\mu_1 - \mu_2)^2 \quad (2)$$

수식 1을 최소화 시키는 것은 수식 2를 최대화 시키는 것과 동일하며 둘 중 어느 기준을 사용해도 되지만 수식 2를 최대화 시키는 것이 효율적이다. 즉, Otsu 이진화 알고리즘은 임계값을 0부터 255까지 단계적으로 변화시키면서 수식 2가 최대가 되는 값을 찾아서 영상을 이진화하는 방법이다.

2.2.2 블러링(Blurring)

블러링(Blurring)은 스무딩(Smoothing)이라고도 불리며 빈번하게 사용되는 영상처리 방법이다. 블러링은 주로 영상에 있는 잡음(Noise)를 완화시키거나, 영상의 크기를 줄이는 경우에 사용된다. 블러링은 가중치를 가지는 필터를 이용해 입력 영상의 각 픽셀의 값을 계산하여 출력영상을 만들어 내는 방법이다. 필터의 가중치에 따라 다양한 블러링 방법이 존재하며, 대표적인 방법으로는 단순 블러링, 중간값 블러링, 가우시안 블러링 등이 있다.

단순 블러링은 평균 블러링이라고도 불리며  $n \times n$

영역의 크기를 가지고 해당 값들은 각각  $\frac{1}{n \times n}$ 로 이루어져있다. 출력영상의 픽셀은 입력 영상의 픽셀을 중심으로  $n \times n$  영역 값들의 평균으로 계산되며  $n$ 의 값이 커질수록 출력영상은 더욱 흐릿해진다. 다음으로 중간 값 블러링은 입력 영상의 픽셀을 중심으로  $n \times n$  영역안에 있는 값들 사이에서 가장 중앙에 있는 값이 출력영상의 픽셀 값이 되는 방법이다. 평균값을 사용하는 단순 블러링은 독립적으로 값이 크게 튀는 잡음을 가지는 입력 영상을 받았을 때 잡음제거가 원활히 이루어지지 않는다. 하지만 중간 값 블러링은 중간 값을 선택하기 때문에 크게 튀는 잡음을 무시하고, 원활한 출력영상을 만들 수 있다.

마지막으로 가우시안 블러링은 동작이 빠른 편은 아니지만 가장 유용한 필터이다. 가우시안 블러링은 입력 영상과 가우시안 필터를 계산하여 출력영상을 만들어 낸다. 가우시안 필터는 가우시안 분포를 영상 처리에 적용하여 정규분포, 확률분포에 의해 생성된 잡음을 제거하기 위한 필터이다. 가우시안 분포는 다음 식과 같다.

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3)$$

$\sigma$ 는 표준편차로서 파라미터로 작동한다. 표준편차의 값이 커질수록 함수가 더 넓게 퍼지는 모양이 되며 더 큰 블러링 효과를 가진다. 모든 픽셀에 똑같은 가중치를 부여했던 단순 블러링과 달리 가우시안 블러링은 중심에 있는 픽셀에 높은 가중치를 부여한다. 입력 영상에 윤곽선(Edge)이 있을 때, 단순 블러링은 윤곽선을 포함해서 전체적으로 블러링이 되는 반면에 가우시안 블러링은 윤곽선이 남아있는 상태에서 출력영상이 만들어진다. 따라서 가우시안 블러링은 윤곽선을 유지하면서 잡음을 제거하는데 사용된다.

2.2.3 모폴로지(Morphology)

모폴로지 필터링<sup>7)</sup>은 영상 분석과 처리에 사용되며 미리 특정한 형태를 띠는 필터를 만들고 이 필터를 영상에 씌워 새로운 영상을 얻어내는 방법이다. 이러한 모폴로지 연산은 사용 방법에 따라 여러 가지 형태의 결과물을 얻을 수 있다. 모폴로지 연산에는 침식(Erode)과 팽창(Dilate)가 있으며, 입력 영상의 픽셀을 기준으로 필터를 적용하여 필터 영역안의 값을 확인하고 출력영상을 만든다. 우선 침식은 필터 영역 내

픽셀 중 최소 픽셀 값을 현재 픽셀 값에 대입하여 출력력을 계산하고, 팽창은 필터 영역 내 픽셀 중 최대 픽셀 값을 대입하여 계산한다. 침식 연산을 통해 입력 영상에서 원하지 않는 작은 노이즈 들을 제거 할 수 있지만 잡음 성분이 제거되는 것뿐만 아니라 대상의 영역이 줄어들다는 문제가 있다. 팽창 연산은 입력 영상에서 대상이 가지는 작은 구멍들을 채우는 방법이다. 하지만 대상의 크기가 전체적으로 입력 영상보다 출력영상이 커진다는 문제와 잡음이 보다 커질 가능성이 존재하는 문제가 있다. 이러한 침식과 팽창이 가지는 문제를 해결하기 위해 침식과 팽창을 연이어 사용하는 열림(Opening)과 닫힘(Closing) 방법이 있다.

열림은 침식 연산은 우선적으로 수행한 후 팽창 연산을 연이어 수행하는 방법이다. 침식을 통해 잡음이 제거되고 팽창을 통해 크기가 줄어든 영역을 다시 키우는 방법이다. 다음으로 닫힘은 열림과는 반대로 팽창 연산을 먼저 사용하고 침식 연산을 연이어 수행한다. 이는 팽창을 통해 작은 구멍들을 채우고 침식을 통해 커진 영역을 다시 줄이는 방법이다. 열림과 닫힘, 두 필터는 객체 감지에 주로 사용된다. 열림 연산은 객체 외부의 잡음을 제거하는데 효율적이며, 닫힘 연산은 객체 내부의 잡음을 제거하는데 효율적이다.

#### 2.2.4 윤곽선 검출(Edge Detection)

윤곽선 검출은 경계선, 윤곽선을 찾아내는 것을 의미한다. 영상에서 윤곽선이란 영상의 밝기가 낮은 값에서 높은 값으로, 또는 높은 값에서 낮은 값으로 변하는 지점을 의미한다. 결국 윤곽선은 영상 안에 존재하는 객체의 경계를 말하는 것으로서, 모양, 방향성을 탐지할 수 있는 등의 여러 정보가 담겨있다. 결국 윤곽선 검출이란 윤곽선에 해당하는 화소를 찾는 과정이다. 윤곽선 검출은 크게 1차 미분 마스크 방법, 2차 미분 마스크 방법, 캐니 윤곽선 검출(Canny Edge Detection)<sup>[8]</sup>으로 구분할 수 있다.

먼저 1차 미분 마스크 방법은 화소의 변화율을 구하는 미분 연산을 이용해 윤곽선을 검출하는 방법이다. 미분 연산은 기울기를 계산하는 것이기 때문에 1차 미분 마스크 방법은 현재 화소에서 밝기의 기울기를 계산하는 것이다. 하지만 디지털 영상은 연속이 아닌 이산된 데이터의 나열이기 때문에 정확한 미분 연산을 수행할 수 없어 근사 값으로 계산하게 된다. 또한 영상은 2차원이기 때문에 가로 방향, 세로 방향 각각 미분을 구해야 한다. 그리고 최종적으로 영상에 나타낼 때는 각각 제곱을 더한 것에 제곱근을 취한다. 이러한 방법을 사용하는 마스크 방법으로는 로버츠

마스크(Roberts Mask), 프레윗 마스크(Prewitt Mask), 소벨 마스크(Sobel Mask) 방법들이 있다. 로버츠 마스크는 3\*3의 마스크에 대각선 방향에 -1과 1을 배치해 X성분과 Y성분의 윤곽선을 검출하는 방법이다. 대각선 원소를 제외한 나머지 원소의 값이 모두 0이기 때문에 다른 1차 미분 마스크에 비해서 속도가 빠른 장점이 있다. 하지만 한번만 차분을 계산하기 때문에 잡음에 매우 민감하고 윤곽선의 강도가 약한 단점이 있다. 프레윗 마스크는 로버츠 마스크와 달리 세 번의 차분을 합하기 때문에 윤곽선 강도가 강하며, 수직과 수평 윤곽선을 동등하게 찾는데 효과적이다. 이에 대각선 방향의 윤곽선보다는 수직, 수평 방향의 윤곽선을 잘 검출한다. 소벨 마스크는 프레윗 마스크의 중심화소의 차분을 2배 증폭시킨 방법이다. 중심화소의 차분을 증폭시킨 마스크이기 때문에, X성분과 Y성분뿐만 아니라 대각선 성분의 윤곽선도 잘 검출하는 특징이 있다.

1차 미분 마스크는 밝기가 급격하게 변화하는 영역 뿐 아니라 점진적으로 변화하는 부분까지 민감하게 윤곽선을 검출하기 때문에 너무 많은 윤곽선이 검출될 수 있다. 따라서 2차 미분 마스크 방법은 이를 보완하는 방법으로, 변화하는 영역의 중심에 있는 윤곽선만을 검출하며, 밝기가 점진적으로 변화되는 영역은 검출하지 않는다. 대표적인 방법으로는 라플라시안 윤곽선 검출(Laplacian Edge Detection) 방법<sup>[9]</sup>이 있다. 라플라시안 마스크는 중심화소를 4배 증폭시키고, 상하 좌우 화소는 중심화소의 반대 부호를 가지도록 설정한다. 4방향, 8방향 방법이 있으며 마스크의 모든 합이 0이 되도록 구성하면 된다. 마지막으로 캐니 윤곽선 검출 방법은 이전의 방법들보다 우월한 성능을 가진 윤곽선 검출 방법이다. 특히 윤곽선을 가장 잘 찾으면서, 잡음이 포함된 잘못된 윤곽선을 제거할 수 있는 방법이다. 캐니 윤곽선 검출 방법은 스무딩(Smoothing), 소벨 마스크(Sobel Mask), 비 최대치 억제(Non-maximum Suppression), 이력 임계값(Hysteresis Threshold)로 총 4단계로 진행된다. 우선 잡음을 제거하기 위해 스무딩 작업을 수행한다. 주로 가우시안 블러를 이용해 노이즈를 제거한 이미지를 사용한다. 두 번째 작업은 소벨 마스크를 이용해 회선을 수행한다. 이미지의 강도가 급격하게 변하는 부분이 윤곽선이기 때문에, X성분과 Y성분의 미분 값을 추측해 모든 윤곽선을 검출한다. 극 값에 따라 윤곽선이 아님에도 윤곽선으로 검출되는 영역은 세 번째 단계인 비 최대치 억제에서 이러한 영역들은 제거한다.



현재 선택된 픽셀의 윤곽선 강도가 가장 크다면 그 값을 보존하고, 아닌 경우에는 삭제(Zero)를 수행한다. 마지막 단계로는 임계값을 가지고 남아있는 윤곽선들을 최종적으로 판별하는 작업을 수행한다. 낮은 임계값(Low Threshold)과 높은 임계값(High Threshold)를 정의한다. 낮은 임계값보다 낮은 수치를 가지는 윤곽선은 삭제하고, 낮은 임계값과 높은 임계값 사이에 존재하는 윤곽선은 약한 윤곽선, 높은 임계값보다 높은 윤곽선은 강한 윤곽선으로 정의한다. 이때 약한 윤곽선은 강한 윤곽선과 연결되어 있을 경우에만 최종 윤곽선으로 판별된다.

$$H = \begin{cases} \frac{60(G - B)}{V - \min(R, G, B)} & \text{if } V = R \\ 120 + \frac{60(B - R)}{\min(R, G, B)} & \text{if } V = G \\ 240 + \frac{60(R - G)}{V - \min(R, G, B)} & \text{if } V = B \end{cases} \quad (6)$$

위 식에서 R, G, B는 0에서 1사이의 값을 가진다. 이때 기존의 RGB의 범위가 0에서 255 사이인 경우 255로 해당 값을 나눠준다. H의 값은 0에서 360까지의 값을 가지며, V와 S는 0에서 1사이의 범위를 가진다. 이때, H의 값이 0보다 작으면 360을 더하여 최종 H를 구한다. 하지만 색상이 단일 색만을 나타내는 것이 아니라 빛의 영향에 따라 같은 색상이라도 여러 형태로 표현 될 수 있기 때문에 다음 그림 3과 같이 비슷한 색을 추출한다. 추출된 색상은 각각 H, S, V의 범위를 가지며 해당 범위를 기반으로 색상을 추출하는데 사용된다. 이에 단일 색에 대해서 빛의 영향이 있더라도 정확한 색상 검출이 가능하게 된다.

### III. 본 론

본 장에서는 토마토 영상을 가지고 토마토의 주간 성장 길이를 찾아내는 방법에 대해 설명한다. 입력 영상에 대한 전처리부터 성장점 스티커와 참조 값을 검출하는 방법에 대해서 서술한다.

#### 3.1 전처리

지난주 성장점(n-1)에서 이번 주 성장점(n)까지 주간 성장 길이를 구하기 위해서는 영상에서 해당 지점을 찾아야한다. 성장점 위치를 찾는 방법에는 여러 가지 방법이 존재하나 본 연구에서는 색상 모델을 활용하여 성장점의 위치를 찾아낸다. 또한 위치를 찾아내는데 있어서 여러 잡음이 존재하기 때문에 블러 처리를 통해 잡음을 제거하여 원활한 성장점 검출을 한다.

##### 3.1.1 색상 추출

성장점에 부착된 스티커의 색상과 참조 값의 색상을 추출하기 위해 색상모델을 이용한다. 본 연구에서는 흔히 사용되는 RGB 색상 모델을 사용하는 것이 아니라, 인간이 직관적으로 표현할 수 있는 HSV 색상 모델을 사용한다. 일반 카메라로 취득된 RGB 영상을 다음 수식을 통해 HSV 색상 모델로 변환한다.

$$V = \max(R, G, B) \quad (4)$$

$$S = \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{if } V = 0 \end{cases} \quad (5)$$

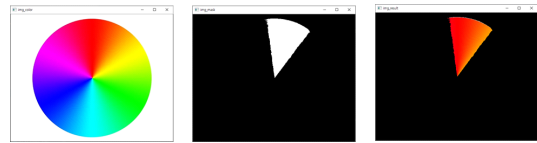


그림 3. HSV 검출. 왼쪽부터 원본, 마스크, 추출 영상  
Fig. 3. HSV Detection. From left to right original image, mask image, extracted image

##### 3.1.2 블러링 및 잡음 제거

HSV 모델을 이용해 색상의 범위를 추출하는 과정은 그림 3과 같이 매끄럽게 추출 될 수도 있다 하지만 그림 4과 같이 추출된 영상 주변에 잡음이 생길 수도 있으며 객체 검출에 어려움을 줄 수 있다. 잡음을 제거하는 여러 가지 영상처리 방법이 있으나, 본 연구에서는 많이 사용되는 가우시안 블러링, 중간값 블러링, 흑백 영상으로 변환 후 영상 이진화, 침식과 팽창을 순차적으로 적용하는 열림 연산 방법들을 통해 잡음을 제거한다. 그림 5에서는 가우시안 블러, 중간값 블러, 영상 이진화, OTSU 이진화, 열림 연산이 적용된 출력 영상이다. 왼쪽 위에서부터 차례로 원본영상, 가우시안 블러링을 적용한 영상, 중간값 블러링을 적용한 영상이며, 두 번째 줄은 왼쪽에서부터 영상 이진화, 가우시안 블러링을 적용 후 영상 이진화, 중간값 블러링을 적용 후 영상 이진화 한 결과이다. 가우시안 블러링보다 중간값 블러링이 좋은 결과를 나타내어 마지막 줄 영상에서는 중간값 블러링 적용 후 OTSU 이

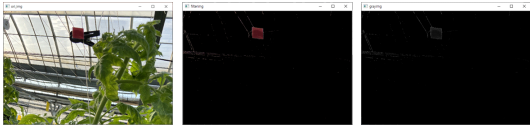


그림 4. 왼쪽부터 원본, 빨간색 추출, 흑백 변환 영상  
Fig. 4. Original, red extraction, gray conversion image

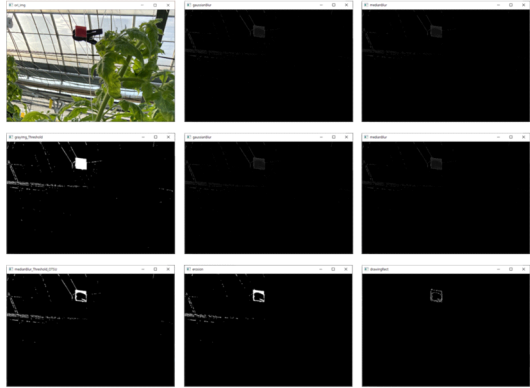


그림 5. 다양한 블러링, 영상 이진화를 통한 윤곽선 검출  
Fig. 5. Various blurring, contour detection through image binarization

진화를 적용하였고 그 다음 영상은 이전 영상을 가지고 열림 연산을 진행한 결과이며, 마지막 영상으로는 최종적으로 윤곽선을 추출한 영상이다.

### 3.2 검출(Detection)

색상 추출을 통해 윤곽선을 구한 후 해당 윤곽선의 객체를 검출해야 한다. 본 절에서는 색상 추출 후 성장점 객체 검출과 참조 객체 검출에 대해 설명한다.

#### 3.2.1 성장점

성장점은 영상으로 구분하기 어렵기 때문에 스티커를 부착하여 지난주와 이번주 성장점을 표시하고, 영상처리를 통해 스티커의 중심을 찾은 뒤, 성장점 간의 거리를 계산한다. 그림 4와 같이 HSV 색상 모델로 색상 범위를 지정하고 해당 색상에 따른 성장점을 추출한다. 다음으로 그림 5와 같이 블러링을 통해 잡음을



그림 6. 성장점 스티커 추출  
Fig. 6. Extraction of growth point stickers

제거하고 이진화와 열림 연산을 통해 최종 윤곽선을 얻어낸다. 최종적으로 얻어진 윤곽선을 기반으로 그림 6과 같이 각 꼭짓점을 구해 성장점의 위치를 검출한다. 그림 6은 왼쪽부터 입력영상, HSV 색상모델 기반 성장점 추출, 출력영상이다.

#### 3.2.2 참조

HSV 색상 모델로 색상의 범위를 지정하고 해당 색상에 따른 성장점 스티커를 추출해 그 사이의 거리를 계산한다. 하지만 영상의 해상도마다 1cm당 픽셀수가 다르기 때문에 참조를 기준으로 계산 할 수 있는 참조 값이 필요하다. 본 항에서는 다양한 참조 값의 종류와 참조를 검출하는 방법에 대해서 설명한다.

##### (1) 참조 값이란

실험에 사용되는 영상을 수집할 때 개인의 핸드폰을 사용해 취득한다. 이는 식물과의 거리, 높이에 따라 해상도가 달라지기 때문에 참조 할 수 있는 참조 값이 필요하다. 이에 본 연구에서는 다양한 참조를 사용하여 보다 정확한 참조 값을 찾아낸다.

##### (2) 참조의 종류

기존에 사용되던 참조는 그림 1과 같이 매우 크고 불편하다는 단점이 있었다. 이에 참조 값과 성장점 스티커를 합쳐서 그림 7과 같이 새로운 참조를 만들었다. 새로운 참조의 길이는 가로, 세로 각각 2.5cm이며 이를 이용해 빨간색 사각형 사이의 거리를 계산한다. 하지만 이와 같은 계산 결과는 그림 8에서 추출되는 사각형은 완전한 정사각형이 아니고, 회전이 이루어져 있기 때문에 정확한 참조가 될 수 없다.

이러한 문제점 때문에 어느 각도에서 찍어도 원래의 모습을 유지하는 참조가 필요하다. 이에 그림 9과 같이 어느 위치에서 사진을 찍어도 원이 찍히는 구를 새로운 참조로 선택했다. 그림 9의 구를 이용해 새로

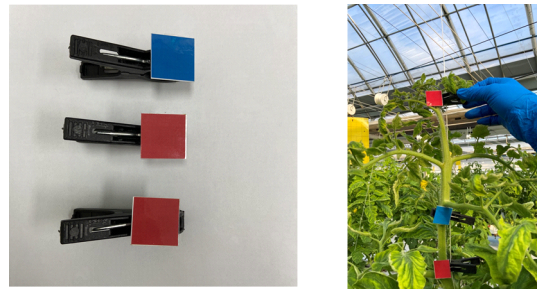


그림 7. 왼쪽부터 새로운 참조, 새로운 참조에 대한 영상  
Fig. 7. New reference, image from new reference

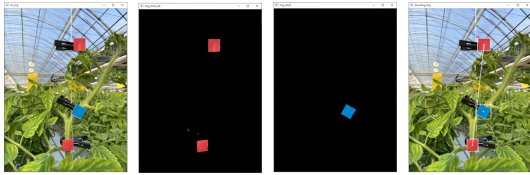


그림 8. 새로운 참조를 이용한 색상 추출  
Fig. 8. Color extraction using new reference

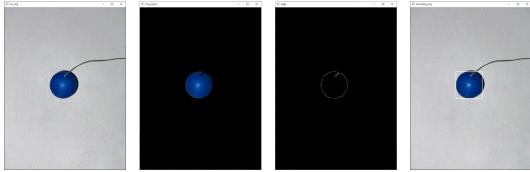


그림 9. 새로운 참조 (구)  
Fig. 9. New reference (sphere)

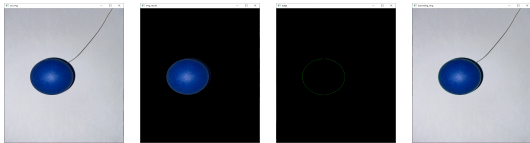


그림 10. 새로운 참조 (탁구공)  
Fig. 10. New reference (Table tennis ball)

운 영상을 취득하고 색 검출 및 참조의 길이를 계산한다. 구는 3D 프린터와 표면을 사포로 이용해 매끄럽게 만들지만, 이는 사람의 손으로 가공했기 때문에 정확한 원을 검출하기 어렵고 계산에 사용할 때 큰 오차가 발생한다. 따라서 시중에 판매하는 도색 된 탁구공을 이용하여 새로운 참조를 만들었다. 그림 10과 같이 표면이 일정하고 둥근 원을 추출해서 계산이 가능하다.

### (3) 참조의 검출

본 연구에서 정의된 참조의 종류에는 직사각형과 원의 형태가 있다. 우선 직사각형의 형태는 성장점에 사용된 스티커처럼 최종 윤곽선을 구한 뒤, 여러 영상 처리 작업을 통해 검출한다. 원의 형태는 회색 색상모델을 이용해 이진화를 진행하고 중간값 블러와 허프 원 변환(Hough Circle Transform)을 이용해 그림 9와 그림 10의 마지막 그림과 같이 가장 적합한 원의 테두리를 검출한다. 그 후 찾은 원을 기반으로 1cm당 픽셀 수를 계산하여 최종 성장점 사이의 길이를 구한다.

## IV. 실험

본 절에서는 3D 프린터로 만들어진 참조와 도색이

된 탁구공을 이용해 실험을 진행한다. 3D 프린터로 만들어진 구는 빨간색과 파란색으로 색상을 기반으로 비교 실험을 진행하고, 도색이 된 파란색 구를 기반으로 실험을 진행한다.

### 4.1 기본 구에 대한 실험 결과

3D 프린터로 만들어진 파란색구와 빨간색 구를 기반으로 여러 실험을 진행하고 그 결과를 나열한다.

#### 4.1.1 파란색 기본 구의 실험 결과

3D 프린터와 사포를 이용해 표면을 매끄럽게 만든 파란색 기본 구를 가지고 103장의 토마토 영상을 취득하여 실험을 진행했다. 103장의 데이터는 15개의 토마토 작물을 가지고 각각 6장에서 8장사이로 취득한 영상이다. 그림 11은 원본영상과 실험결과 중 일부이며, 빨간색 성장점 스티커와 파란색 기본 구를 찾아 바운딩 박스(Bounding Box)를 만들고 거리를 계산한다. 그림 12는 그림 11의 파란색 구를 확대한 영상이다. 검출된 원은 최대한 밀접하게 바운딩 박스를 만들었으나 외접한 원이 아니라 내접한 원을 찾으며, 이는 정확한 원이 아니기 때문에 성장점 길이를 계산할 때 오차가 발생한다.



그림 11. 새로운 참조 (구)를 이용한 실험 결과  
Fig. 11. Experimental results using the new reference (sphere)

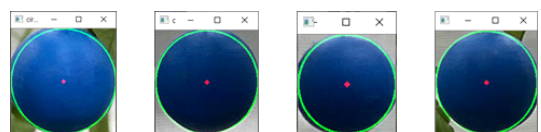


그림 12. 새로운 참조 (구)를 확대한 영상  
Fig. 12. New reference (sphere) enlarged image

표 1은 작물 1번부터 작물 15번까지의 실제 생장 길이와 빨간색 스티커 중심들 사이의 직선거리를 측정 한 값이다. 표 2는 작물별로 실제 생장 길이에 대한 실험 결과의 증감률을 나타낸다. 검출된 원은 직접 원을 검출하고 지름을 구해 생장 길이를 계산한 방법이

표 1. 작물별 직선거리와 실제거리 측정 값  
Table 1. Measurements of straight and actual distances by crop

작물번호	직선거리(mm)	실제거리(mm)
1	323	330
2	426	413
3	335	336
4	370	375
5	367	385
6	383	382
7	438	481
8	403	408
9	232	233
10	268	273
11	300	309
12	252	258
13	268	277
14	405	435
15	387	402

표 2. 원의 지름 계산 방법에 따른 증감률  
Table 2. Increase and decrease rates according to the method of calculating the diameter of the circle.

작물번호	검출된 원	수직거리	직선거리
1	5.90%	4.79%	4.69%
2	6.10%	15.31%	15.65%
3	2.06%	17.44%	17.36%
4	12.70%	4.58%	4.63%
5	4.92%	10.22%	9.67%
6	4.85%	5.48%	5.59%
7	17.79%	11.28%	11.28%
8	13.31%	11.85%	11.58%
9	22.30%	10.94%	10.75%
10	5.87%	7.05%	7.05%
11	15.83%	2.52%	2.31%
12	7.48%	5.90%	5.79%
13	17.70%	4.81%	4.81%
14	13.25%	5.01%	5.01%
15	13.58%	18.21%	18.21%
전체평균	10.91%	9.03%	8.96%

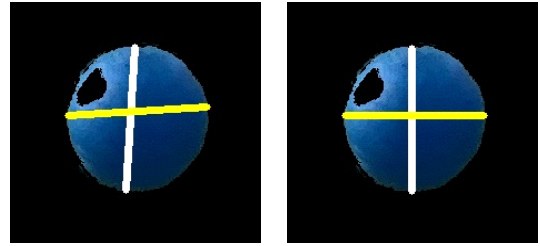


그림 13. 검출된 원에 대한 지름 계산 방법  
Fig. 13. Diameter calculation method for detected circle

며, 수직거리는 그림 13의 왼쪽과 같이 검출된 원에서 수직으로 가로 세로 길이를 구해 더 긴 부분을 지름으로 선택해 계산한 방법이다. 마지막으로 직선거리는 그림 13의 오른쪽과 같이 검출된 원을 가지고 X축에서 가장 긴 부분, Y축에서 가장 긴 부분을 찾아 그 중 더 긴축을 지름으로 계산한 방법이다. 해당 실험에서는 각각 10.91%, 9.03%, 8.96%로 직선거리가 가장 좋은 결과를 나타냈고, 실제 생장 길이에 대한 증감률은 9.13%, 7.30%, 8.93%로 수직거리가 가장 좋은 결과 값을 가진다.

4.1.2 색상에 따른 실험 결과

3D 프린터로 기본 구를 만들 때, 파란색 구 뿐만 아니라 빨간색 구도 제작을 하여 실험을 진행했다. 비교를 위해 15개의 토마토를 가지고 파란 색구와 빨간 색 구별로 영상을 40장씩 취득했다. 그림 14, 그림 15는 실험에 사용된 원본 영상과 실험 결과 영상이다.

표 3과 표 4는 이전 항 “파란색 기본 구의 실험 결과”에서 나오는 표와 같은 의미를 가진다. 표 4에서 1-1은 첫 번째 작물을 파란색 구로 측정 한 결과이며, 1-2는 첫 번째 작물을 빨간색 구로 측정 한 결과이다.



그림 14. 색상에 따른 실험에 대한 영상 (파란색 구)  
Fig. 14. Experimental image according to color (blue sphere)





그림 15. 색상에 따른 실험에 대한 영상 (빨간색 구)  
 Fig. 15. Experimental image according to color (red sphere)

다른 작물들도 같은 의미를 가진다. 12-2와 13-2는 취득한 데이터는 있으나 노을을 빨간색 원으로 인식해 오류가 난 데이터이다. 파란색 구 평균과 빨간색 구 평균을 비교하면 대체적으로 빨간색 구에 대한 평균이 더 좋다. 하지만 빨간색 구를 검출 못하는 오류와 검출된 원을 기반으로 하는 계산법을 제외한 수직거리와 직선거리의 평균은 큰 차이를 보이지 않으므로 파란색을 참조로 사용하는 것이 효율적이다.

표 3. 작물별 직선거리와 실제거리 측정 값  
 Table 3. Measurements of straight and actual distances by crop

작물번호	직선거리(mm)	실제거리(mm)
1	135	155
2	75	105
3	90	115
4	105	160
5	140	155
6	157	180
7	160	192
8	122	125
9	135	138
10	133	137
11	152	163
12	173	178
13	98	115
14	145	145
15	108	111

표 4. 원의 지름 계산 방법에 따른 증감률  
 Table 4. Increase and decrease rates according to the method of calculating the diameter of the circle

작물번호	검출된 원	수직거리	직선거리
1-1	10.83%	5.11%	4.99%
1-2	7.04%	5.36%	5.36%
10-1	5.02%	6.05%	6.05%
10-2	4.42%	3.44%	3.44%
11-1	5.04%	4.63%	4.63%
11-2	2.19%	2.36%	2.36%
12-1	23.04%	5.70%	5.70%
12-2	-	-	-
13-1	0.65%	1.42%	1.42%
13-2	-	-	-
14-1	14.38%	9.75%	9.75%
14-2	4.47%	1.29%	1.29%
15-1	11.50%	10.50%	10.50%
15-2	2.61%	1.25%	1.25%
2-1	6.81%	6.29%	6.29%
2-2	3.70%	5.95%	5.95%
3-1	9.09%	6.89%	5.65%
3-2	9.88%	8.89%	8.89%
4-1	15.41%	7.24%	6.77%
4-2	3.81%	3.28%	3.28%
5-1	6.44%	3.67%	4.04%
5-2	2.17%	2.71%	2.71%
6-1	7.64%	7.43%	7.43%
6-2	3.00%	3.39%	3.39%
7-1	6.39%	4.64%	4.53%
7-2	7.50%	2.95%	2.95%
8-1	4.31%	1.84%	1.84%
8-2	2.43%	3.88%	3.88%
9-1	4.32%	2.33%	2.33%
9-2	4.33%	0.68%	0.68%
파란 평균	8.72%	5.57%	5.46%
빨간 평균	4.43%	3.49%	3.49%

4.2 탁구공에 대한 실험 결과

이전의 3D 프린터로 가공된 구는 표면을 사람이 사포로 가공했기 때문에 매끄러운 원 검출이 어렵고 계산하는 방법에 따라 차이가 난다. 이에 본 실험에서는 40mm의 지름을 가진 파란색 탁구공을 참조로 사용했다. 총 105장의 영상을 가지고 실험을 진행하였으며, 그림 16은 실험에 사용된 원본 영상과 실험 결과 영상이다.

표 5과 표 6는 이전 항 “파란색 기본 구의 실험 결과”에서 나오는 표와 같은 의미를 가진다. 표 6에서 전체 평균의 값은 각각 8.77%, 6.09%, 6.00% 이며, 이는 기본 파란색 구를 참조로 사용한 실험보다 좋은 결과를 나타낸다. 또한 색상, 크기, 표면적으로 볼 때 가장 적합한 참조라 할 수 있다.

표 5. 작물별 직선거리와 실제거리 측정 값  
Table 5. Measured values of straight and actual distance by crop

작물번호	직선거리(mm)	실제거리(mm)
1	113	116
2	87	84
3	93	98
4	72	74
5	162	165
6	233	238
7	192	189
8	167	167
9	136	141
10	218	227
11	228	238
12	105	107
13	251	258
14	223	234
15	232	235

표 6. 원의 지름 계산 방법에 따른 증감률  
Table 6. Increase and decrease rates according to the method of calculating the diameter of the circle

작물번호	검출된 원	수직거리	직선거리
1	11.52%	8.63%	8.44%
2	11.86%	5.91%	5.83%
3	10.64%	6.18%	6.06%
4	11.41%	6.07%	5.91%
5	5.30%	5.33%	5.29%
6	4.23%	4.53%	4.40%
7	6.21%	3.49%	3.45%
8	17.42%	4.10%	3.99%
9	6.75%	5.11%	4.91%
10	8.92%	7.72%	7.69%
11	8.20%	7.63%	7.67%
12	6.95%	6.73%	6.66%
13	8.91%	6.71%	6.64%
14	4.17%	6.59%	6.42%
15	8.99%	6.65%	6.57%
전체평균	8.77%	6.09%	6.00%



(a) 원본영상과 실험결과 1  
(a) Original & result image 1



(b) 원본영상과 실험결과 2  
(b) Original & result image 2

그림 16. 탁구공에 대한 원본 영상과 실험 결과  
Fig. 16. Original image and experiment results for table tennis balls

## V. 결 론

본 논문에서는 토마토 작물에 대한 영상을 여러 영상처리 방법들을 사용해 작물의 주간 성장 길이를 계산한다. HSV 색상모델을 이용해 색상 범위를 추출하여 원하는 객체를 얻고, 블러링과 잡음 제거를 통해 최적의 윤곽선을 얻었다. 이러한 방법들을 통해 지난 주 성장점과 이번주 성장점 그리고 참조를 검출해 참조의 길이를 기반으로 최종 성장 길이를 계산한다. 여러 참조의 종류 중에 표면이 매끄러운 탁구공이 가장 적합했으며, 색상으로는 빨간색 구는 노을에 영향을 받아 오검출을 하는 반면에 파란색 구는 오검출이 없었고 결과 또한 크게 차이나지 않기 때문에 파란색 구가 더 효율적인 색상이다. 또한 원의 지름을 선택하는 부분은 검출된 원의 직선거리를 지름으로 계산하는 방법이 가장 정확했다. 하지만 영상의 환경에 따라 참조의 검출, 성장점 스티커의 검출 등 오류가 나타나는 것을 볼 수 있었다. 이는 다양한 각도, 환경을 가진 영상 취득 후, 더 많은 영상을 기반으로 실험을 통해 더 나은 방법을 찾을 계획이다.



References

- [1] M. Lee, J. Hwang, and H. Yoe, "Agricultural production system based on IoT," *2013 IEEE 16th Int. Conf. Computational Sci. and Eng.*, Sydney, NSW, pp. 833-837, 2013.
- [2] S. Kim, M. Lee, and C. Shin, "IoT-based strawberry disease prediction system for smart farming," *Sensors*, vol. 18, no. 11, 2018.
- [3] Y. Kang, "A study on the improvement for smart farm policy using AHP analysis," M.S. Thesis, Dept. of Food Industrial Management, Dongguk University Graduate Schools, 2020.
- [4] H.-S. Park, "Vehicle tracking system using HSV color space at nighttime," *J. KIIECT*, vol. 8, no. 4, pp. 270-274, 2015.
- [5] S. E. Umbaugh, "*Digital image processing and analysis : human and computer vision applications with CVPtools*," Boca Raton, FL: CRC Press. ISBN 978-1-4398-0205-2, 2010.
- [6] D. Liu and J. Yu, "Otsu method and K-means," *Ninth Int. Conf. Hybrid Intell. Syst. IEEE*, vol. 1, pp. 344-349, 2009.
- [7] 신선미, "Morphology를 기반으로 하는 윤곽선 검출 및 대역 선택적 필터를 이용한 선명도 향상 시스템의 구현," M.S. Thesis, Dept. of Electron. Eng., Donga University Graduate Schools, 2009.
- [8] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 8, pp. 679-714, 1986.
- [9] R. Kimmel and A. M. Bruckstein, "On regularized Laplacian zero crossings and other optimal edge integrators," *Int. J. Computer Vision*, vol. 53, no. 3, pp. 225-243, 2003.

김강섭 (Kangsub Kim)



2017년 2월: 전북대학교 컴퓨터공학부 졸업  
 2019년 2월: 전북대학교 전자정보공학부 석사  
 <관심분야> 컴퓨터공학, 영상처리, 자연어처리  
 [ORCID:0000-0003-2403-3811]

홍영기 (Youngki Hong)



2001년 3월: 일본 동경농업대학교 농업공학전공 석사  
 2004년 3월: 일본 동경농업대학교 생물환경조절학전공 박사  
 2006년 3월: 일본 동경농업대학교 종합연구소 박사후연구원  
 2007년 12월~현재: 농촌진흥청 국립농업과학원 농업연구사  
 <관심분야> 영상처리, 패턴인식, 기계학습  
 [ORCID:0000-0002-9772-9820]

김현종 (Hyunjong Kim)



1991년 2월: 서울대학교 축산학과 졸업  
 1993년 2월: 서울대학교 동물자원학과 석사  
 2000년 2월: 서울대학교 동물자원학과 박사  
 <관심분야> 시계열 데이터분석, 생체정보, 패턴인식  
 [ORCID:0000-0001-6217-9374]

**김 국 환 (Gookhwan Kim)**



2010년 2월 : 경희대학교 기계공학과 박사

2010년 3월 : 국립농업과학원 박사후연구원

현재 : 국립농업과학원 농업연구사

<관심분야> 농업 로봇, 지능형 농작업기, 자율항법

[ORCID:0000-0002-7278-3476]

**이 명 훈 (Meonghun Lee)**



2004년 2월 : 순천대학교 정보통신공학과 졸업

2006년 2월 : 순천대학교 정보통신공학과 석사

2011년 2월 : 순천대학교 정보통신공학과 박사

2017년 02월~현재 : 농촌진흥청 국립농업과학원 농업연구사

<관심분야> 인공지능, 표준, 무선통신, ICT융합

[ORCID:0000-0001-8863-7580]

**김 경 철 (Kyoungchul Kim)**



2009년 2월 : 전북대학교 정밀기계공학과 (공학석사)

2015년 8월 : 전북대학교 정밀기계공학과 (공학박사)

2016년 7월~2019년 1월 : 농업기술실용화재단 연구원

2019년 2월~현재 : 국립농업과학원 스마트팜개발과 연구사

<관심분야> 농업로봇, 농업드론, 스마트 팜

[ORCID:0000-0001-6699-881x]