

# Algorithms to estimation of size and shape tomato using Artificial Vision Techniques

Wolfgang D. Niño P., Eng.<sup>1</sup>, Fabián R. Jiménez L., MSc.<sup>2</sup>; Andrés F. Jiménez L., PhD<sup>3</sup>  
Pedagogical and Technological University of Colombia - Department of Electronic Engineering - I<sup>2</sup>E Research Group,  
Tunja, Colombia.<sup>1,2</sup> Email: {wolfgang.nino, fabian.jimenez02}@uptc.edu.co  
Universidad de los Llanos - Faculty of Sciences and Engineering, Villavicencio, Colombia.<sup>3</sup>  
Email: ajimenez@unillanos.edu.co

**Abstract**– This paper describes the development of estimation algorithms for the size and shape of tomato fruits, implemented on a portable electronic system used in greenhouses, with the purpose to classify and collect of tomatoes milano and chonto type.

The algorithms were implemented in a Raspberry-Pi embedded card through the use of free software libraries (Python, OpenCV, Pillow and Scikit-Image) and using a Pi-Camera for real-time processing, taking into account the parameters of classification defined in national (NTC1103-1) and international (USDA) standards.

Were applied classification algorithms based on Canny to estimate the size of the tomato fruit, as well as techniques based on estimation of the eccentricity and statistical moments descriptors for the measurement of the shape, which were evaluated with a performance superior to 90 %.

Classification techniques of tomato shape and size using artificial vision overcame subjective techniques of visual and tactile type classification carried out by farmers and specialized technicians.

**Keywords**– Fruit Classification System, Canny Algorithm, Statistical Moments Techniques, Eccentricity Algorithm, Artificial Vision, Process Automation.

## I. INTRODUCTION

In post-harvest handling of fruits and vegetables of climacteric type such as mango, tomato, lemon, banana, apple, papaya, citrus, etc., characteristics detection and evaluation methods are required for their classification, either size, morphology or coloration. Methodologies, norms and related indexes have been implemented to define the size and shape of the fruits, and to evaluate the quality and fruit application, which in many cases involve sensory type tests and are of invasive-destructive nature.

The implementation of detection and evaluation methods for tomato fruit characteristics such as artificial vision applied in the agro-food sector has been carried out in recent years in developing countries (Latin America, South Asia and Africa), which in turn are the largest producers of climatic and agricultural fruits.

Among the advantages of artificial vision systems are the quantitative physical features characterization of the fruits such as morphology, coloration, size and texture properties, in a non-invasive, objective, accurate way, and do not require human inspection that tends to be expensive, imprecise and subjective.

Digital Object Identifier (DOI):

<http://dx.doi.org/10.18687/LACCEI2019.1.1.5>

ISBN: 978-0-9993443-6-1 ISSN: 2414-6390

**17<sup>th</sup> LACCEI International Multi-Conference for Engineering, Education, and Technology:** “Industry, Innovation, And Infrastructure for Sustainable Cities and Communities”, 24-26 July 2019, Jamaica.

The artificial vision techniques are increasingly useful in the fruit industry especially for quality control, inspection and classification applications by color or by some type of defect [1].

The tomato quality control is defined by the Colombian Technical Norm NTC-1103-1 [2], which is unknowing for several food industries and major distributors. This standard establishes a set of quality requirements for the different fruit varieties, including parameters such as size, tolerance, color, presentation and labeling.

Many of the techniques used to analyze the caliber and morphology of tomato fruits are still based on the reliability of a visual and tactile analysis, which do not guarantee to carry out a quality process in optimal conditions and may present significant errors. The size, morphology and maturity classification are the most used for tomato fruit [3].

The tomato classification automation is implemented in a reliable way using Digital Image Processing techniques that allow to identify and analyze the different quality control evaluation parameters for tomato selection.

### A. Tomato Classification by Size or Caliber

The tomato size is determined by the maximum equatorial diameter. For all the varieties except the Cherry type, and the classification by size is established according to the Colombian technical standard NTC 1103-1. The tomato size can be classified as large, when its size is greater than 82 mm, medium with calibers between 57 and 81 mm, and small ones of less than 56 mm. The minimum size for round and oval tomatoes is 35 mm and for elongated tomatoes 30 mm [2].

### B. Tomato Classification by Morphology

There are different forms of tomato fruit in the market. Tomatoes differ according to their intention or final use, whether for fresh consumption, industrial processing or classification based on their external form. The tomato shape depends on the type, the growth treatment (fertilization) and the seeds chemical modification to obtain a specific shape or composition [4].

The tomato shape can be globular, round or flat, these characteristics largely determine the market and package type for marketing. For example, flat and medium-sized fruits are required for presentation in trays [5].

## II. DESIGN AND IMPLEMENTATION OF TOMATO CLASSIFIER PROTOTYPE

A portable electronic prototype was designed, which facilitates the task of classifying tomatoes under different lighting conditions, whether in an open field, greenhouses or other uncontrolled environments.

The prototype design to analyze the physical characteristics of the tomato was integrated by subsystems dedicated to particular functions that allow the image acquisition, its processing, analysis, description and classification according to the physical parameters established in terms of form and size, to then visualize and store them. The subsystems proposed to be implemented in the electronic system of the classifier are shown in Fig. 1. The following subsystems were considered for the design of the tomato classification equipment:

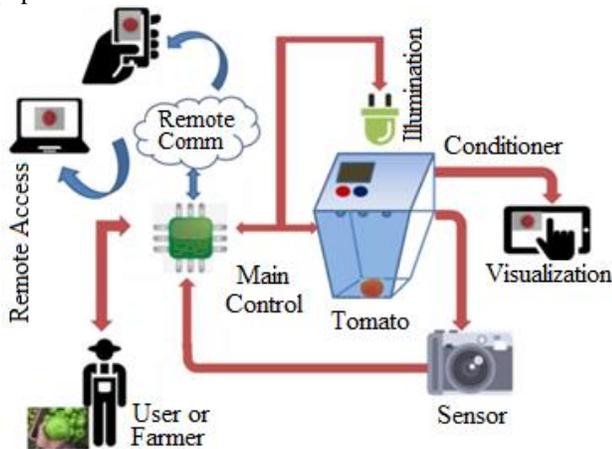


Fig. 1 General Scheme and General Scheme of the Portable Tomato Classifier Prototype.

**Sensor:** This device converts the fruit scene into a digital image to be processed and storage. In this development the optical sensor used was the Pi-camera module V.01 [6, 7], with a resolution of 5 Megapixels (2592 x1944), allowing to control parameters such as brightness, contrast and image resolution. The distance between the camera and the fruit is constant, and the scene background is white to facilitate the processing of the image.

**Illumination System:** this system conditions the capture environment of the scene, to obtain an image in the best conditions of luminosity, besides keeping regulated the light intensity and directionality to optimize the contrast and to define appropriately the object characteristics. Around the Pi-Camera, LED lighting is arranged, where the diodes were connected to the I/O port GPIO of the Raspberry® Pi. The system has a white compartment that works as diffuser material of natural, reducing the formation of shadows or luminous points on the fruit skin.

**Main Controller:** is the embedded system that controls the capture, storage, processing, communication and visualization of the classifier. Analyze the images and/or videos captured.

To do this functions, the Raspberry® Pi 2 model B card was selected [8].

**Analysis Software:** is the platform that hosting the processing algorithms for the analysis and classification of the acquired images. For this development, the OpenCV platform of Python™ [9] was used, due it's an open source tool, has specialized libraries for image processing, the image processing applications implemented are executed with more speed and keeps compatibility with the hardware platform.

**Visualization System:** reproduce in real time the graphic result of the classification in the prototype, and allows the user to interact with several control functions (image capture or classify them). A 3.2-inch touch screen was incorporated, designed to be connected into the GPIO pins of the Raspberry® Pi, in addition to being compatible with the control software.

**Communication System:** is the interface to communicate the equipment with other peripherals or subsystems, or access other protocols to transmit information. The Raspberry® Pi module has a connection with Ethernet cable that can be used to connect either directly to a router or a laptop to access the main functions of the system. The implemented interface consists of a Wi-Fi communication module, which allows to remotely control the system from an external device wirelessly (cellphone or computer).

### A. Tomato Classifier Prototype

The image acquisition of tomato fruits requires a conditioning for the adequate transmission of natural or artificial light in the scene. For this purpose, was adapted a plastic material compartment semi-transparent (See Fig. 2) with a white cover, to guarantee uniform lighting conditions, avoiding problems with the shadows and blurring most of the natural light in the environment. A diffused light transmission environment is emulated in the plastic compartment.

The camera device was adapted in the upper part of the compartment, along with the directional LED lighting. The distribution of the LEDs was arranged in a circular shape around the camera to capture the images. Fig. 2 shows the prototype implemented for the image acquisition and tomato classification. The equipment elements numbered Fig. 2 corresponds to:

1. Image acquisition module.
2. Container compartment, left side view.
3. Battery container compartment.
4. Connectors (USB Ports) and Ethernet Port connector.
5. Power on and power off buttons.
6. Opening for the lid of the image acquisition compartment.
7. Prototype front view, container compartment.
8. Image acquisition module front view.
9. Power switch-button (front view).
10. On/ Off potentiometer for independent illumination brightness control.
11. LCD Display Screen (3.2 inches).

- 12. Prototype rear view.
- 13. Battery container compartment rear view.
- 14. Container compartment rear view.

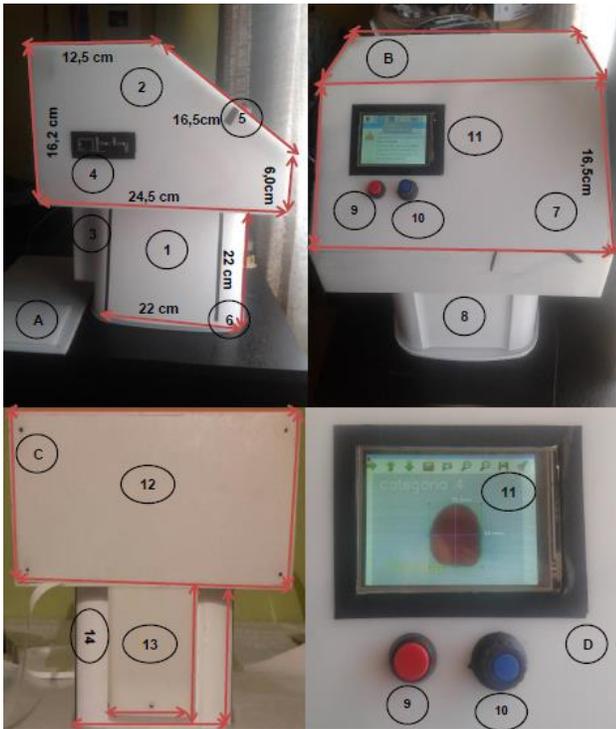


Fig. 2 A. Prototype right side view. B. Prototype front side view. C. Prototype rear side view. D. Display screen and control buttons

**B. Image Pre-Processing**

The stages for the pre-processing of the image require the application of digital image processing techniques, for the subsequent design of the automatic learning algorithm. The image processing system for the classifier was constituted according to the flow diagram shown in Fig. 3.

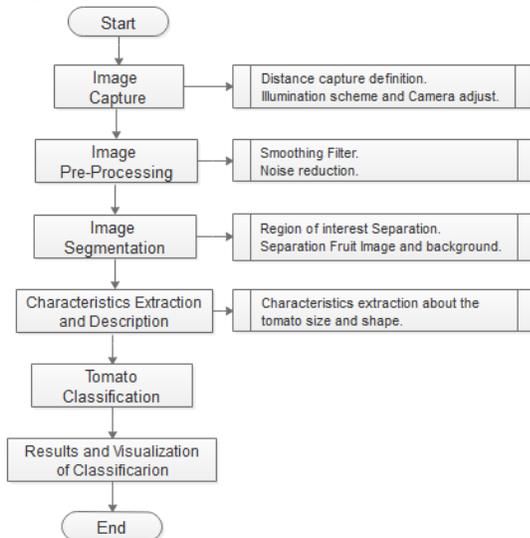


Fig. 3 Flow Diagram of the artificial vision algorithm for tomatoes classification.

**1) BGR format conversion to grayscale.**

Once the image was acquired in BGR format (Blue, Red and Green), it was converted to a gray scale format for facilitate the pre-processing of the image comprising the filtering processes and obtaining the characteristic histogram; in addition to reduce noise in the image. The conversion to gray scale was done according (1):

$$I_{esc-gris}(x, y) = \alpha I_{color}(x, y, r) + \beta I_{color}(x, y, g) + \gamma I_{color}(x, y, b) \quad (1)$$

Where the weighting coefficients ( $\alpha, \beta, \gamma$ ), are defined according to the perceptive response of the human eye, where  $\alpha = 0.2989, \beta = 0.5870$  and  $\gamma = 0.1140$ . Although a grayscale image contains less information than a color image, the most important parameters related to its characteristics and information are maintained, such as, for example: borders, regions, spots, junctions, etc. [10].

In Fig. 4, the acquisition of an image of a mature tomato is shown and its corresponding preprocessing in gray scale.

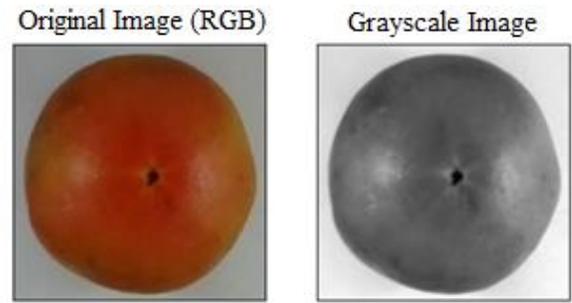


Fig. 4 Left: Original color image in RGB color space. Right: Grayscale image after conversion.

**2) Application of Low Pass Filter.**

Once the image is converted to a gray scale format, the quality of the acquired image is improved by applying a non-linear statistical filter of median. The median filter reduces the noise in the image with similar results to the averaging filter, however, its performance is better since more useful details are preserved in the image (for example, the edges) [11]. In Fig. 5, the results of the application of median filter are showed.



Fig. 5 Left: Original acquired image. Right: Final result after applying a median filter.

As shown in Fig. 5, when applying the median filter, the details and tomato characteristics are preserved, softening the

fruit interior. Additionally, preserve the edges in the image and reduce the impulsive noise.

### 3) Image Thresholding.

In this stage, the object pixels (fruit) are separated from the elements that belong to the scene background, in order to extract the region of interest (ROI) to be analyzed. The Otsu segmentation technique was applied (Nobuyuki Otsu, 1979) [12], that obtain an optimal threshold to binarize the image, from the characteristic bimodal histogram.

The Otsu method works best with bimodal distributions, although the performance may deteriorate when there is a significant imbalance between the number of pixels of the object and the background.

The Otsu algorithm finds a threshold value ( $t$ ) that minimizes the variance within each class given the relationships shown in (2), (3), (4) and (5).

$$(\sigma_w(t))^2 = q_1(t) * \sigma_1^2(t) + q_2(t) * \sigma_2^2(t) \quad (2)$$

$$q_1(t) = \sum_{i=1}^t P(i) \quad y \quad q_2(t) = \sum_{i=t+1}^s P(i) \quad (3)$$

$$\mu_1(t) = \sum_{i=1}^t \frac{i * P(i)}{q_1(t)} \quad y \quad \mu_2(t) = \sum_{i=t+1}^s \frac{i * P(i)}{q_2(t)} \quad (4)$$

$$\sigma_1^2(t) = \sum_{i=1}^t [1 - \mu_1(t)]^2 * \frac{P(i)}{q_1(t)} \quad (5)$$

$$\sigma_2^2(t) = \sum_{i=t+1}^s [1 - \mu_2(t)]^2 * \frac{P(i)}{q_2(t)}$$

The threshold value  $t$  is between the two peaks of intensity level such that the variations of both classes are minimal. The application of this method is observed in Fig. 6 by calculating the bimodal histogram on the grayscale image of Fig. 4.

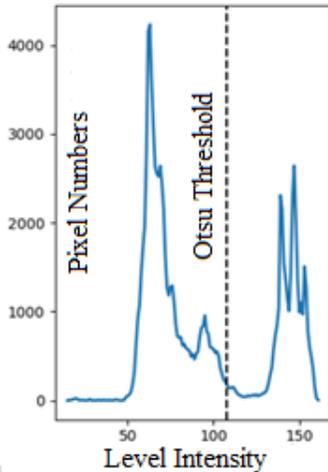


Fig. 6 Bimodal Histogram of the grayscale image. The dotted line intersects the histogram to find the threshold  $t$ .

To binarize the image, an appropriate threshold value  $t$  was chosen within the gray levels.

### 4) Image Segmentation.

Once the optimum threshold has been found, the image is converted to a binary format (binarization) and its separated into two main regions: the fruit (digital value 255), and the background (digital value 0), as observed in Fig. 7.

A new image is obtained, where the gray values dispersion is eliminated by segmenting the pixels belonging to the object respect to the other regions of the image.

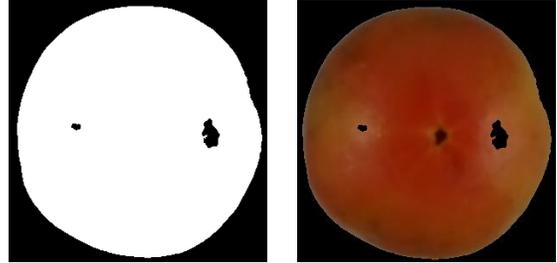


Fig. 7 Left: Image Binarized to separate the Region of Interest ROI. Right: Extraction of ROI by applying a binary mask.

With the binarized image in Fig. 7 (Left), an AND operation is performed with the original image and the background scene is eliminated. In addition, the ROI is extracted (tomato pixels). The logical AND operation is used due its commonly used to detect image differences or highlight target regions. The result of the extraction of the ROI from the image is shown in Fig. 7 (right).

The ROI was cut, discarding all the pixels of the background of the scene. For this procedure the contours information of the image was used. The points that form a closed region are detected, and the coordinates in  $(x, y)$  of said points are stored forming a cumulative list.

Once the points coordinates were stored, a function was implemented that swept each coordinate of the list and found the maximum and minimum values. Finally, its obtained a group of four points of coordinates that correspond to the upper limits left and right, and lower limits left and right of the segmented image. Once the coordinates of the border points are located, a cut is applied to obtain a new image.

## III. ALGORITHMS TO DETERMINE TOMATO SIZE AND SHAPE.

### A. Algorithm to estimate the Tomato Size (Caliber)

The algorithm that determines the tomato size was made with the purpose of measuring the scaled dimensions of the fruit in real time using functions of the OpenCV libraries and the Pi-Camera module. The algorithm was based on the detection of the contour of the tomato to be measured and the calculation of the minimum rectangle that encloses it [13]. The next requirements were raised to the measurement:

- The fruit should be placed on a uniform color background, preferably white to facilitate the image processing.
- The tomato must have enough contrast to be detected by the camera.
- The camera must be located at a fixed distance and angle from the object.

- Only one fruit (tomato) could be measured.

To calculate the contours, it was necessary to extract the fruit edges, with the implementation of a high-pass filters. The best results were obtained using the Canny operator, although the only disadvantage of this algorithm was the selection of an adequate threshold (critical in the subsequent processing stages). To obtain the tomato size, it was necessary to analyze the arrangement between the object and the camera (see Fig. 8).

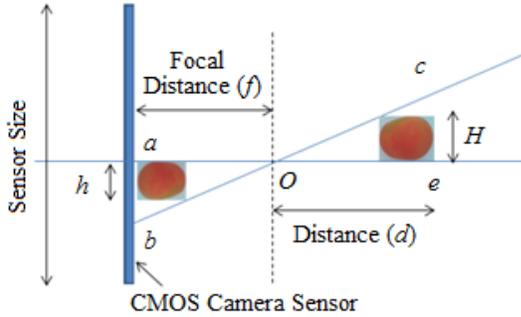


Fig. 8 General Layout of the camera and tomato.

The video captured by the camera was in BGR format, but for the frames processing frames it was necessary the gray scale images conversion, to later calculate the tomato borders and contours [14]. The object size was calculated by the ratio of the two triangles formed in Fig. 8:  $\Delta oab$  y  $\Delta oce$ , this relationship was defined by (6):

$$\frac{d}{H} = \frac{f}{h} \quad (6)$$

Where  $H$ , corresponds to the real height of the fruit, and  $h$  to the height of the fruit represented in number of pixels. From the expression shown in (6), the value of the actual size of the tomato can be obtained, but was necessary to convert the pixels to a real measure. This was achieved with the application of a factor of proportion given by (7):

$$factor_{prop} = \frac{S}{Camera\ Resolution} \quad (7)$$

The conversion factor is a relation between millimeters and pixels, and defines the value required to pass the size of the object expressed in pixels to the real size. Another way to obtain the conversion factor was to perform the test with a tomato in the capture plane and establishing the unitary conversion factor, when executing the procedure, the fruit width and length dimensions in pixels was obtained, using (8):

$$factor_{conv} = \frac{Height_{real}}{Height_{pixels}} \quad factor_{conv} = \frac{Weight_{real}}{Weight_{pixels}} \quad (8)$$

The expression (9) finds the size of the resulting object:

$$H = \frac{Factor_{conv} \cdot d \cdot h}{f} \quad (9)$$

The measurement of the tomatoes size was implemented in real time (Video-Streaming), together with the digital

processing of the frames captured by the Pi-Camera. Initially the normalization of the camera parameters was performed, such as brightness, video resolution and contrast of the frames. The real video capture was made using the OpenCV library, in BGR format and then converted in gray scale format. Next a median filter was applied to eliminate the noise in the scene.

The edge detection of the tomato fruit was implemented with the application of a high-pass filter using the Canny operator, which depends on the minimum and maximum thresholds established, to obtain the best contour of the fruit [15].

The Canny algorithm was chosen due is the most popular and efficient edge detection techniques.

The Canny algorithm depends on several parameters such as: the standard deviation of the Gaussian filter  $\sigma$ , the thresholds of the hysteresis process and the values of the low and high intensity level to obtain the pixels of the edge that define the tomato real contour [16].

The application of the Canny operator allows to obtain a sketch of the tomato contour, which is sensitive to the luminosity changes, and adapts the low and high thresholds. The subsequent image analysis requires obtaining a closed contour. The application of morphological operators allows to obtain a closed contour region. In Fig. 9 the result of the application of the Canny operator on the captured image and the detected edges is shown.

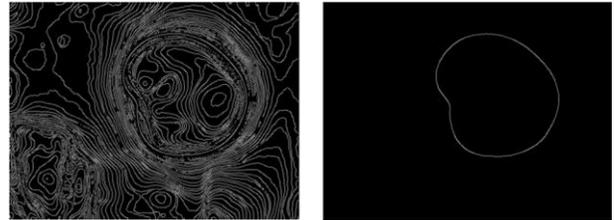


Fig. 9 Detection of tomato edges using Canny Operator. Left: Fruit Edge map obtained without hysteresis thresholds. Right: application of hysteresis thresholds and contour obtained.

For the fruit contour detection were used dilation and erosion operators applied on the edge map obtained. This procedure fills the gaps between the pixels of the tomato edges that were not correctly sealed. In Fig. 10 the application of the dilatation and erosion operators is shown together on the right image of Fig. 9.

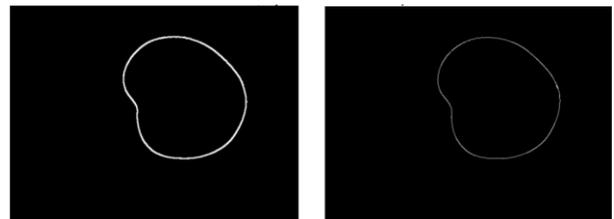


Fig. 10 Left: Application of morphological dilation operator on the fruit contour. Right: application of the erosion operator.

Once the image has been pre-processed, along with the tomato edge map, the contours of Fig. 10 (right) are found,

using the contour functionality of the OpenCV library [16]. Finally, a list of all image contours was obtained, where individually each one corresponds to a coordinate matrix  $(x, y)$  that defines the object's limit points.

The approach method was implemented to eliminate redundant points and compress the contour list, allowing to save memory at the time of executing the application. If the contour is not large enough, the region is discarded, assuming that it is introduced noise in the edge detection process.

If the contour region is large enough, a bounded rectangle is calculated on the image, which can be adjusted (rectangle rotation) to the detected contours. Then the coordinates were organized in the detector rectangle corners: top left, top right, bottom right and bottom left, as shown in Fig. 11.

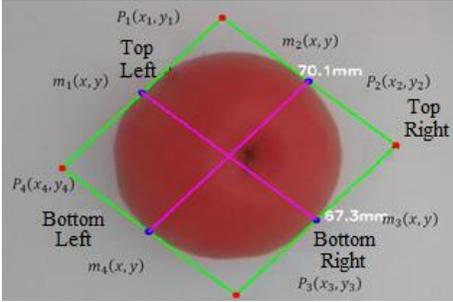


Fig. 11 Bounded Rectangle (Green Line) of fruit contour and coordinates in each vertex (Red Points).

With the vertices of the bounded rectangle and knowing its coordinates, the midpoints on each of the lines that make up the rectangle (Blue Dots) were calculated using (10) and (11).

$$m_1(x, y) = \left( \frac{x_1 + x_4}{2}, \frac{y_1 + y_4}{2} \right) \quad m_2(x, y) = \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right) \quad (10)$$

$$m_3(x, y) = \left( \frac{x_2 + x_3}{2}, \frac{y_2 + y_3}{2} \right) \quad m_4(x, y) = \left( \frac{x_3 + x_4}{2}, \frac{y_3 + y_4}{2} \right) \quad (11)$$

Where  $m_1(x, y)$  is the midpoint between the vertices of  $P_1$  and  $P_4$ ,  $m_2(x, y)$  the midpoint between the vertices  $P_1$  and  $P_2$ ,  $m_3(x, y)$  the midpoint between the vertices  $P_2$  and  $P_3$ , and  $m_4(x, y)$  the midpoint between the vertices  $P_1$  and  $P_4$ . The midpoints were connected with violet lines (See Fig. 11) and the Euclidean distance between the sets of midpoints was calculated, according (12) and (13):

$$d_A = d_{24} = \sqrt{((m_{4x} - m_{2x})^2 + (m_{4y} - m_{2y})^2)} \quad (12)$$

$$d_B = d_{31} = \sqrt{((m_{3x} - m_{1x})^2 + (m_{3y} - m_{1y})^2)} \quad (13)$$

Where the variable  $d_A$  defines the tomato height distance and  $d_B$  express the tomato width length both in pixels. Finally, the scale factor defined by (9) was calculated and multiplied with the corresponding values of  $d_A$  and  $d_B$  to obtain the actual height and width of the tomato. Fig. 12 illustrates the flowchart describing the algorithm for determining the tomato size.

## B. Algorithm to estimate the Tomato Shape

According to Arjenaki's research [17], the tomato shape can be identified by its curvature, where some shapes can be categorized as "rounded" and "oblong". To find the tomato shape index, its eccentricity  $\varepsilon$  is calculated. This measure establishes how the object deviates from being circular. The eccentricity of a circle being equal to zero, but for an ellipse its eccentricity is greater than zero and less than one.

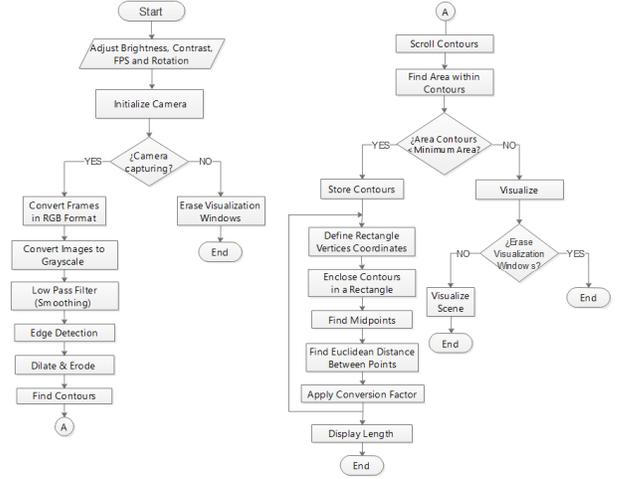


Fig. 12 Flow Diagram for the algorithm to determine the tomato size (major and minor axes).

The eccentricity needs four parameters to be measured: the gravity center coordinates, the width, the height and the area of the image. The eccentricity can be calculated according to (14):

$$\varepsilon = 2 \cdot \frac{\sqrt{\left(\frac{L_{maj}}{2}\right)^2 - \left(\frac{L_{min}}{2}\right)^2}}{L_{maj}} \quad (14)$$

Where  $L_{maj}$  is the major axis length and  $L_{min}$  is the minor axis length [10]. Another method with similar results is the study of the image statistical moments. The statistical moments definition of a function  $f(x,y)$  that represents an object is defined by (15):

$$m_{p,q} = \iint x^p \cdot y^q \cdot f(x,y) dx dy \quad (15)$$

The integration calculates the object area, being necessary that  $f(x, y)$  is a binary function of the image defined by (16):

$$f(x, y) = \begin{cases} 1 & \text{object, ROI} \\ 0 & \text{background} \end{cases} \quad (16)$$

The moments were defined as: zero order moment  $((p,q) = (0,0))$ , according to (17):

$$m_{0,0} = \iint 1 \cdot f(x,y) dx dy \quad (17)$$

The zero order moment describes the object area  $A$ , which is invariant to the image rotation or translation. The first order moments  $((p, q) = (1,0)$  or  $(0,1))$ , were defined by (18):

$$m_{1,0} = \iint x \cdot f(x,y) dx dy \quad m_{0,1} = \iint y \cdot f(x,y) dx dy \quad (18)$$

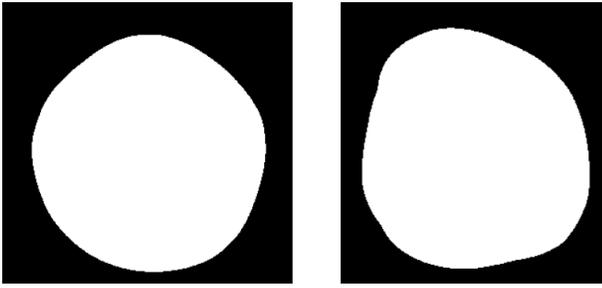


Fig. 13 Binary Format of a tomato preprocessed image, through calculation of first and second statistical moments.

The first order moments contain information about the object gravity center (centroid).

$$x_c = \frac{m_{1,0}}{m_{0,0}} \quad y_c = \frac{m_{0,1}}{m_{0,0}} \quad (19)$$

The second order moments  $((p, q) = (2,0) \text{ or } (0,2) \text{ or } (1,1))$ :

$$\begin{aligned} m_{2,0} &= \iint x^2 \cdot f(x, y) dx dy \\ m_{0,2} &= \iint y^2 \cdot f(x, y) dx dy \\ m_{1,1} &= \iint x \cdot y \cdot f(x, y) dx dy \end{aligned} \quad (20)$$

From the knowledge of the statistical moments, the central moments were obtained with respect to the object gravity center  $(x_c, y_c)$ , and can be calculated as:

$$\mu_{p,q} = \iint (x - x_c)^p \cdot (y - y_c)^q \cdot f(x, y) dx dy \quad (21)$$

And the following results were obtained:

$$\mu_{0,0} = m_{0,0} \quad \mu_{1,0} = \mu_{0,1} = 0 \quad (22)$$

The central moments can be calculated directly using the spatial statistical moments:

$$\mu_{p,q} = \frac{m_{p,q}}{m_{0,0}} - \left(\frac{m_{1,0}}{m_{0,0}}\right)^p \cdot \left(\frac{m_{0,1}}{m_{0,0}}\right)^q \quad (23)$$

Where the central moments of first and second order can be derived in the following way [10, 18]:

$$\begin{aligned} \mu_{1,0} &= \frac{m_{1,0}}{m_{0,0}} - \left(\frac{m_{1,0}}{m_{0,0}}\right)^1 \cdot 1 = 0 \\ \mu_{0,1} &= \frac{m_{0,1}}{m_{0,0}} - 1 \cdot \left(\frac{m_{0,1}}{m_{0,0}}\right)^1 = 0 \end{aligned} \quad (24)$$

$$\begin{aligned} \mu_{2,0} &= \frac{m_{2,0}}{m_{0,0}} - \left(\frac{m_{1,0}}{m_{0,0}}\right)^2 = \frac{m_{2,0}}{m_{0,0}} - x_c^2 \\ \mu_{0,2} &= \frac{m_{0,2}}{m_{0,0}} - \left(\frac{m_{0,1}}{m_{0,0}}\right)^2 = \frac{m_{0,2}}{m_{0,0}} - y_c^2 \end{aligned} \quad (25)$$

$$\mu_{1,1} = \frac{m_{1,1}}{m_{0,0}} - \left(\frac{m_{1,0}}{m_{0,0}}\right) \cdot \left(\frac{m_{0,1}}{m_{0,0}}\right) = \frac{m_{1,1}}{m_{0,0}} - x_c \cdot y_c \quad (26)$$

The eccentricity  $\varepsilon$  allows to establish how round or oblong is the tomato fruit and can be calculated using the second order central moments by the expression (27):

$$\varepsilon = \frac{(\mu_{2,0} - \mu_{0,2})^2 - 4(\mu_{1,1})^2}{(\mu_{2,0} + \mu_{0,2})^2} \quad (27)$$

#### IV. RESULTS OF SHAPE AND SIZE ESTIMATION OF TOMATO FRUITS

The tomatoes shape estimation was made through the calculation of the first and second order central moments over the binary images (27). This estimate was obtained through the thresholding process by using the approximate eccentricity measure. As a second option, the eccentricity was calculated according to the expression (14), depending only on the major and minor axis measurement of the fruit, from the algorithm that calculates the tomato diameters.

Using the image pre-processing techniques, was obtained the image binary mask, where the region of interest corresponded to the binary area with digital value of 1. For the calculation of first and second order spatial statistical moments, equations (18) to (20) were implemented in the algorithm, then the first and second order central moments were obtained.

Using a representative set of tomato fruits images, the calculation of statistical parameters is summarizing in Table I, for shape estimation.

TABLE I  
SHAPE ESTIMATION ALGORITHMS COMPARISON

#	Tomato Shape Estimation by Central Moments.				Tomato Shape Estimation by axes measurement and eccentricity.		
	$\mu_{2,0}$	$\mu_{0,2}$	$\mu_{1,1}$	$\varepsilon$	Eje Mayor (Píxeles)	Eje Menor (Píxeles)	$\varepsilon$
1	1877220	1260605	3633	0.191	1449.9504394	1429.5461	0.16955
2	1916459	1336805	7073	0.16	1486.7531738	1421.7946	0.30571
3	1482345	1046090	-183	0.172	1165.90454	1047.48962	0.48874
4	2972240	1081154	920	0.4656	1297.594	1284.01171	0.14583
5	2273428	958036	1448	0.408	1133.5697021	1044.93347	0.42052
6	2200291	1306865	380	0.2543	1286.2393798	1255.5911	0.22229
7	1687878	1403532	-103	0.0921	1473.5795898	1471.1855	0.05707
8	384077	83740	159	0.640	449.20135498	404.71487	0.48158
9	402880	114321	-1512	0.5696	507.78210449	441.9009	0.56603
10	403738	154790	53	0.4453	626.44622802	499.70718	0.75602
11	382336	227174	1350	0.2457	497.60211181	458.02209	0.42461
12	300063	140117	81	0.3626	521.55847168	503.77636	0.26803
13	262882	161622	-355	0.241	384.81103515	353.07049	0.43345
14	552853	151742	-1664	0.5787	513.69696044	435.61138	0.62501
15	463960	183619	1363	0.4244	462.50238037	439.73278	0.32594

According results obtained in Table I, two eccentricity values ( $\varepsilon$ ) were obtained, the first corresponds to the application of the second order central moments algorithm.

The moment  $\mu_{1,1}$  provides information about the image orientation (some indicators were negative), these do not affect the deduction of the eccentricity due the rotation invariance of the moments applying (27).

The second eccentricity value was computed from the major and minor axis measurements (in pixels), that were obtained from the size algorithm, applying (14).

The most accurate calculation corresponds to the algorithm that estimates the tomato diameters, with a maximal error of 10% when compared with the actual measurement. In some measurements the eccentricity calculated by the statistical moments method was similar (samples 1, 4, 5, 6, 9 and 14 of Table 1), but in other cases reached an error up to 40%. Due this situation, the statistical moments algorithm was not considered a reliable estimation.

The results of the size estimation algorithm using the axes measurements were collected to compare the obtained values obtained for the major and minor axis. The results of these measures were summarizing for a samples set of several sizes and shapes (See Table II and III).

TABLE II  
COMPARISON BETWEEN ACTUAL AXES MEASUREMENTS AND RESULTS OBTAINED BY THE SIZE ESTIMATION.

#	Algorithm Measures		Actual Measures		Error Percentages	
	Minor Axis (mm)	Major Axis (mm)	Minor Axis (mm)	Major Axis (mm)	Minor Axis Error (%)	Major Axis Error (%)
1	83,8	87,4	79	83	5,727	5,3012
2	73,5	80,3	74	77	0,680	4,2857
3	76,8	79	72	77	6,25	2,5974
4	72,4	79,7	68	75	6,077	6,2666
5	91,7	95	85	89	7,306	6,741
6	71,9	73	66	70	8,205	4,2857
7	64,9	66,1	62	63	4,468	4,9206
8	62,6	65,2	61	62	2,555	5,1612
9	67,3	69,1	61	70	9,361	1,285
10	59	71,7	59	66	0	8,636

For most of the measurements estimated with the axes size estimation algorithm given in Table II, the error did not exceed 10% when compared with the actual measurement. The maximum difference was 9 mm between the two measurements for the particular case in the sample 9. These errors were due to the algorithm error that was not accurate and to systematic errors in the tomato diameters measurement of the diameters, since they were made with a conventional instrument (rule with smaller scale given in millimeters). In the same way, the location and configuration of the camera lens properties, very possibly affected the measurement.

As shown in Table III, the tomatoes eccentricity was calculated, where the results estimated by the shape algorithm were very close to the actual measurements. With this

methodology, the fruit shape estimation was more accurate, compared to the calculations made with the second order central moments.

TABLE III  
COMPARISON BETWEEN ACTUAL ECCENTRICITY WITH THE ECCENTRICITY OBTAINED BY THE AXES SIZE ESTIMATION ALGORITHM.

#	Eccentricity (Algorithm)	Eccentricity (Actual)	Estimated Shape	Actual Shape
1	0,284047975	0,306696917	Rounded	Rounded
2	0,402732872	0,276412944	Rounded	Rounded
3	0,234351629	0,354476469	Rounded	Elongated
4	0,418088013	0,421847787	Elongated (Oval)	Elongated
5	0,261279613	0,296424853	Rounded	Rounded
6	0,172944866	0,333197251	Rounded	Rounded
7	0,189681428	0,177465712	Rounded	Rounded
8	0,279578973	0,178879621	Rounded	Rounded
9	0,226759448	0,49052242	Elongated (Oval)	Elongated
10	0,568225211	0,448187862	Elongated (Oval)	Elongated

These results were satisfactory and had a good approximation to actual measurements. The implemented algorithms can be implemented to measure other fruits of similar characteristics of size and shape.

## CONCLUSIONS

The tomato size and shape classification algorithms were implemented on the classifying prototype, using training databases of tomato images and real time captured images.

The performance evaluation of sizes and shapes estimation were based in the national standard NTC-1103, obtaining as a result accuracies equal to or greater than 90% compared with actual measurements.

The tomato caliber calculation and the fruit diameters estimation was executed in real time (video streaming) using the OpenCV artificial vision library, however, a negligible error of less than  $\pm 10$  mm (millimeters) was presented, when these results were compared with actual measurements. The radial and tangential distortions of the camera lens presented this error.

The tomato size detection algorithm was sensitive to the choice of the high and low thresholds for the Canny operator application. The appropriated selection of these thresholds guarantees the best edges detection in the hysteresis stage.

The tomato morphology estimation was more precise using the eccentricity calculation using the minor and major axes measurements, which correspond to the shape estimation algorithm.

The implementation of the portable tomato classification prototype was developed using artificial vision techniques implemented in a Raspberry Pi development platform, that can be replicated for fruits and vegetables study and analysis with tomato similar characteristics. These applications can be

adapted through the implemented algorithms reconfiguration depending on the conditions that are required.

## REFERENCES

- [1] M. P. Arakeri and Lakshmana, "Computer Vision Based Fruit Grading System for Quality Evaluation of Tomato in Agriculture industry," in *Procedia Computer Science*, vol. 79, pp. 426-433, 2016.
- [2] Tomate de mesa (Industrias Alimentarias) 1995-08-23, Norma Técnica Colombiana NTC 1103-1, Instituto Colombiano de Normas Técnicas y Certificación (ICONTEC). Segunda Actualización, 2001.
- [3] J. Jaramillo and L. Sánchez, *Tecnología para el Cultivo de Tomate bajo condiciones protegidas*, Bogotá, D.C.: Ed. Corporación Colombiana de Investigación Agropecuaria, CORPOICA. 2013, pp. 119-129.
- [4] J. Jaramillo, V. Rodríguez, M. Guzmán, M. Zapata and T. Rengifo, *Technical Manual: Good Agricultural Practices in the Production of tomato under protected conditions*, FAO, 2007.
- [5] H. Escobar and R. Lee, *Manual de producción de tomate bajo invernadero*, Bogotá D.C.: Ed. Universidad Jorge Tadeo Lozano, 2009, pp. 14 – 18.
- [6] Color CMOS QSXGA ¼" (5 megapixel) image sensor with Omni BSI™ Technology OV5647. Datasheet Preliminary Especification. Versión 1.0. Available: [https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/ov5647\\_full.pdf](https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/ov5647_full.pdf).
- [7] D. Jones, *Manual de manejo librería módulo Pi-Camera*. Available: <https://picamera.readthedocs.io/en/release-1.13/>.
- [8] Raspberry PI Foundation. Documentación en línea Raspberry-Pi y foros de discusión online. Available: <https://www.raspberrypi.org/>.
- [9] Artificial Vision Library OpenCV 3.0.0, Available: [https://docs.opencv.org/3.0beta/doc/py\\_tutorials/py\\_imgproc/py\\_table\\_of\\_contents\\_imgproc/py\\_table\\_of\\_contents\\_imgproc.html](https://docs.opencv.org/3.0beta/doc/py_tutorials/py_imgproc/py_table_of_contents_imgproc/py_table_of_contents_imgproc.html).
- [10] C. R. Gonzalez and R. E. Woods, *Digital Image Processing*, Fourth Edition, Prentice Hall, Cap. 6, Color Image Processing, 2016, pp. 299.
- [11] Q. Uvais and C. H. Chen, *Digital Image Processing: An Algorithmic Approach with MATLAB®*, Taylor & Francis Group. 6000 Broken Sound Parkway, New York, 2009, pp. 108.
- [12] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.
- [13] D. Mayorca, "Diseño y construcción de prototipo para la clasificación de lulo, *Solanum sp* mediante técnicas de visión artificial y automatización," Grade Thesis, Dept. Mechatronic Eng., Universidad Mariana, Pasto, Colombia, pp.431-446, 2016.
- [14] R. F. Teimourlou, A. Arefi and A. M. Motlagh, "A Machine Vision System for the Real-Time Harvesting of Ripe Tomato," in *Journal of Agricultural Machinery Science*, vol. 7, no. 2, pp. 159-164, 2011.
- [15] R. Klette, "Concise Computer Vision, An Introduction into Theory and Algorithms", Ed. Springer. Computer Science, Department University of Auckland, London, 2014.
- [16] S. Manohar, "Mastering Machine Learning with Python™ in Six Steps," Ed. Apress Media, Bangalore, Karnataka, India. 2017, pp. 53.
- [17] O. O. Arjenaki, P. A. Moghaddam and M. Motlagh, "Tomato Sorting Based on Maturity, Surface Defects and Shape by using of Machine Vision System," in *Turkish Journal of Agriculture and Forestry*, vol. 37, pp. 62 - 68, 2013.
- [18] J. Prateek, "Artificial Intelligence with Python™," Ed. Packt Publishing Ltd. Birmingham, B3 2PB, UK., 2017.



**Wolfgang D. Niño P.** Graduated in Electronic Engineering in 2018 from the Pedagogical and Technological University of Colombia. He is currently researcher in the I2E Research Group. Experience in research in the following fields: Precision Farming, Digital Image Processing and Embedded Systems.



**Fabián R. Jiménez L.** Graduated in Electronic Engineering in 2000 and Specialization in Industrial Automation in 2002 from the Pedagogical and Technological University of Colombia. Master's Degree in Engineering with an emphasis in Automation and Control from the Faculty of Engineering of the National University of Colombia in 2011.

He is currently Assistant Professor of the School of Electronic Engineering of the Pedagogical and Technological University of Colombia, belongs to the I2E Research Group. Experience in research in the following fields: Electronic Control, Digital Processing of Signals and Images, Industrial Instrumentation and Automation of Industrial Processes.



**Andrés F. Jiménez L.** Graduated in Electronic Engineering in 2003, Master of Science in the Faculty of Sciences of the National University of Colombia in 2011 and currently pursuing a Doctorate in Mechanical and Mechatronic Engineering from the National University of Colombia. Nowadays, it acts as Assistant Professor of the Faculty of Sciences and Engineering of the Universidad de los Llanos and is member of MACRYPT Research Group.

Research experience in the following fields: Precision Agriculture, Remote Sensing, Applied Mechanics, Automation and Control Systems.