

## Original papers

## Real-time segmentation of strawberry flesh and calyx from images of singulated strawberries during postharvest processing



A. Durand-Petiteville\*, S. Vougioukas, D.C. Slaughter

University of California, Davis, Department of Biological and Agricultural Engineering, One Shields Avenue, Davis, CA 95616, USA

## ARTICLE INFO

## Article history:

Received 22 April 2017

Received in revised form 26 June 2017

Accepted 8 September 2017

Available online 18 September 2017

## Keywords:

Image processing

Postharvest processing

Strawberries

## ABSTRACT

This paper presents an image processing algorithm that automatically extracts the flesh and calyx areas from strawberry images. Images are captured by a camera included in a strawberry de-capping machine. Lighting is controlled and the background is known, conditions that are typical of postharvest processing. The goal is to extract as many flesh and calyx pixels as possible while rejecting any pixels belonging to the background. The proposed approach relies on image color segmentation in a two-dimensional color space, followed by a blob detection and selection stage. A set of 250 images is used to analyze the sensitivity of the algorithm with respect to user-defined parameters, and evaluate the performance of the approach. The algorithm appears to be easy to tune and allows accurate extraction of the areas of interest despite natural variation in strawberry shape and visual appearance. More than 98% of the flesh area was successfully extracted by the algorithm with less than 1% of the background pixels falsely included. Moreover, up to 79% of the calyx area could be extracted with less than 0.25% erroneous background pixels. Finally, the algorithm has been implemented using the C++ and Cuda languages and can be executed in real-time.

© 2017 Elsevier B.V. All rights reserved.

## 1. Introduction

Since 2000, the worldwide annual strawberry production has been constantly increasing. Recent data shows that between 2010 and 2014 production increased from 6.6 to 8.1 million tons (Food and Agriculture Organization, 2016). A significant portion of harvested strawberries is processed. For example, approximately 25% of all strawberries harvested in California (2.3 bn tons in 2014) are frozen for the processed market. Also, consumer demand for ready-to-eat, fresh-cut packaged fruits has been steadily increasing (James et al., 2011). Fresh strawberry processing tasks include capping (calyx removal), size sorting, grading, slicing, pick-and-placing and packing. Currently, these tasks (excluding size sorting) are performed manually. Automated, cost-effective execution of these tasks requires singulating, orienting and handling strawberries of varying sizes and shapes at high-throughput rates. Earlier attempts that utilized purely mechanical means such as vibration, rotation, pinching, suction to handle strawberries (e.g. Gerrans et al., 1963; Leban, 1976; Ledebuhr et al., 1978) proved impractical due to high failure rates in fruit positioning and selective handling. It seems then mandatory to

consider approaches including computer vision systems, similarly to various recent food processing projects (Du and Sun, 2004; Vibhute and Bodhe, 2012).

This paper addresses the flesh-calyx segmentation problem during strawberry processing at the postharvest stage. More specifically, a real-time vision system is presented that is part of a strawberry de-capping machine (see Fig. 1). The capping machine comprises several successive stages: feeding and singulation, orientation, vision system (calyx-flesh extraction), cutting (computation of cutting line and cutting action), and ejection. The images are captured under controlled lighting conditions as the fruits lie still on a moving roller conveyor (the rollers initially rotate to orient the fruits but stop before entering the imaging section). The vision system has to calculate two binary images representing the flesh and the calyx respectively from images containing singulated strawberries at random positions and orientations (see Fig. 2). The position of the calculated cutting line determines the amount of flesh removed and depends on the accuracy of the flesh and calyx extraction. Thus, to be as accurate as possible, the algorithm has to detect as many flesh and calyx pixels as possible (i.e., “the signal”) while minimizing the number of background pixels that are erroneously labeled as flesh or calyx (i.e., the “noise”). Moreover, the image processing algorithm has to be fast enough to run in real time and not limit the overall throughput of the de-capping machine. Indeed, the cost effectiveness of such a machine

\* Corresponding author.

E-mail addresses: [adurandp@ucdavis.edu](mailto:adurandp@ucdavis.edu) (A. Durand-Petiteville), [svougiouka@ucdavis.edu](mailto:svougiouka@ucdavis.edu) (S. Vougioukas), [dcslaughter@ucdavis.edu](mailto:dcslaughter@ucdavis.edu) (D.C. Slaughter).

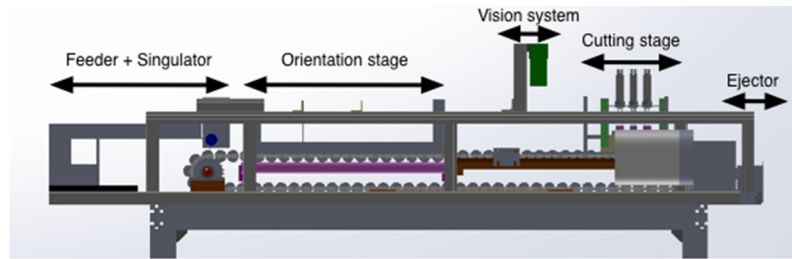


Fig. 1. Sketch of a de-capping machine.



(a) Original image (b) Binary flesh image (c) Binary calyx image

Fig. 2. Expected results.

is strongly related to its capacity to process a large number of berries per second.

Computer vision has already been used in strawberries harvesting and post harvest processes to provide feedback about berry orientation and position. In Nagata et al. (2000) the authors used a camera in order to compute strawberry orientation on a sorting device by extracting the flesh area. To do so, the image is segmented in gray color space based on a user-defined threshold. A similar approach is used in Saenz et al. (2013) to design a harvesting robot prototype. In the work presented in Hayashi et al. (2011), a system to handle strawberries using a suction device is presented. In this system, the flesh and calyx are obtained by segmenting the image in a color space made by a composition of the red green and blue layers. In these works, the areas of interest (flesh and/or calyx) are extracted using user-defined thresholds. In Feng et al. (2008), Han et al. (2012), and Qingchun et al. (2012) similar approaches using user-defined thresholds are used in the OHTA and HSI color spaces. The works presented in Liming and Yanchao (2010) and Yamamoto et al. (2012) propose to extract the flesh with a segmentation process that uses automatically computed thresholds (Otsu's method) either in the V color space of HSV, or in the G-R color space (green layer minus the red one). A similar approach using a variant of Otsu's method in the RGB color space is presented in Wei et al. (2014). Finally, the work presented in Ouyang et al. (2013) proposed a different and more advanced approach. In order to identify disease, a set of processes is applied to the image to extract the flesh: median filter denoising, segmentation using Otsu's method, mean shift clustering and morphological operations. The previously presented approaches focus solely on flesh extraction, except for the work presented by Liming and Yanchao (2010) and Han et al. (2012) which propose simple methods to extract also the calyx and the stems respectively. An image processing algorithm that extracts accurately and rapidly the flesh as well as the calyx from the images has not been reported.

The algorithm proposed in this paper relies also on a threshold based segmentation for both flesh and calyx but then extends the previous works with a blobs detection step to improve the accuracy of the extraction. This choice relies on two observations. First, more advanced segmentation techniques such as clustering, edge detection, classification or region growing are more relevant in challenging environments with uncontrolled lighting and background (e.g., fruit detection in orchard) but too time consuming for real-time postharvest processing operations where lighting and background are controlled. Second, the threshold based segmentation on its own does not provide highly accurate extraction of the flesh. Indeed, in the existing literature, the processes operate always at the pixel level, and therefore cannot consider higher-level features such as size, shape, position or orientation of a blob. This limits the ability to detect whether the pixels extracted via segmentation actually belong to the area of interest, or they are irrelevant.

Thus, the image processing algorithm proposed in this work first relies on a color-based segmentation using two color spaces. The first one R-G to detect the flesh, and the second one G-B for the whole fruit. Next, the obtained images are processed to detect existing blobs and compute some of their geometrical features. Finally, thresholds-based algorithms dedicated to strawberries are designed to keep or remove blobs. Thus, they maximize the accuracy of the extraction. This blob-based approach is novel because it relies on a higher level of abstraction than previous approaches and unlike prior work does not use morphological operations that typically distort the true fruit shape. Another contribution is the sensitivity analysis of the algorithms performance with respect to user-defined thresholds, which allows for easy tuning of the algorithm. Finally, although part of a specific system, the flesh-calyx extraction algorithm is generic, in that it operates under conditions (controlled lighting and background) that are typical of other strawberry postharvest processing operations.

The work is presented as follows. In the first section, the image acquisition equipment and methods used to acquire the images of the strawberries are presented. Next, the set of image processing operations used to extract the flesh and the calyx are described in detail. In the third section, a set of 250 images is used to evaluate the performance of the algorithm and its sensitivity to the thresholds. To do so, the percentage of extracted signal and noise, *i.e.*, the pixels respectively representing the fruit and the background, are used as performance metrics, and results are reported for several sets of features thresholds. Moreover, a comparison with results obtained using approaches from other works is presented. The last section summarizes the presented work and results, and discusses potential improvements.

## 2. Materials and methods

### 2.1. Image acquisition equipment

The images of the strawberries were acquired using a Basler avA1900-50gc camera with a 16 mm Fujinon lens. It provided a spatial resolution of  $1920 \times 1080$  pixels with 24-bit color images at a maximum frame rate of 51 frames per second. The camera communicated with a Dell Latitude E5440 laptop via a GigE interface. The laptop was equipped with 8 GB of RAM, an i7-4600 CPU running at 2.7 GHz and a NVIDIA GeForce GT 720 M 2 GB graphics card.

The pictures were taken in post-harvesting conditions, that is to say the fruits were placed on a conveyor which consists of rollers that could move at a programmable forward speed and rotate, if desired to view both sides of the fruit (see Fig. 3). In this work the rollers did not rotate. The space between two consecutive rollers is referred to as a 'pocket', and was size to contain only one strawberry. The camera was triggered by an infrared sensor whose signal would rise when the front edge of a roller was detected. As the conveyor traveled at a constant speed of 20 cm/s, each image always contained the same view of the conveyor. Moreover, due to the moving speed of the fruits, the exposure time was set at 2 ms in order to obtain non-blurry images. To provide consistent and adequate lighting, especially given the short exposure time, a set of four lights was installed above the pocket of interest. Thus, a clear and non-blurry image of the pockets was taken as the conveyor was moving. Finally, the acquired images were processed using the C++ and CUDA languages as well as the openCV library (Bradski, 2000). The specifics of the image processing implementation are beyond the scope of this paper. However, it should be noticed that the algorithms have been implemented to run in real time.

### 2.2. Image database

In this paper, 250 images<sup>1</sup> containing a single strawberry are used to illustrate and validate the proposed approach. The strawberries were harvested in August 2016 in the region of Salinas, California, USA. Then, they were stored for two days in a temperature-controlled chamber set at two degrees Celsius before being used for the images. Fig. 4 presents three typical samples of the fruit images.

### 2.3. Image processing algorithm

The aim of this work is to process the acquired images to identify the pixels corresponding to the flesh and calyx of a strawberry. The set and sequence of operations proposed in this work are sum-

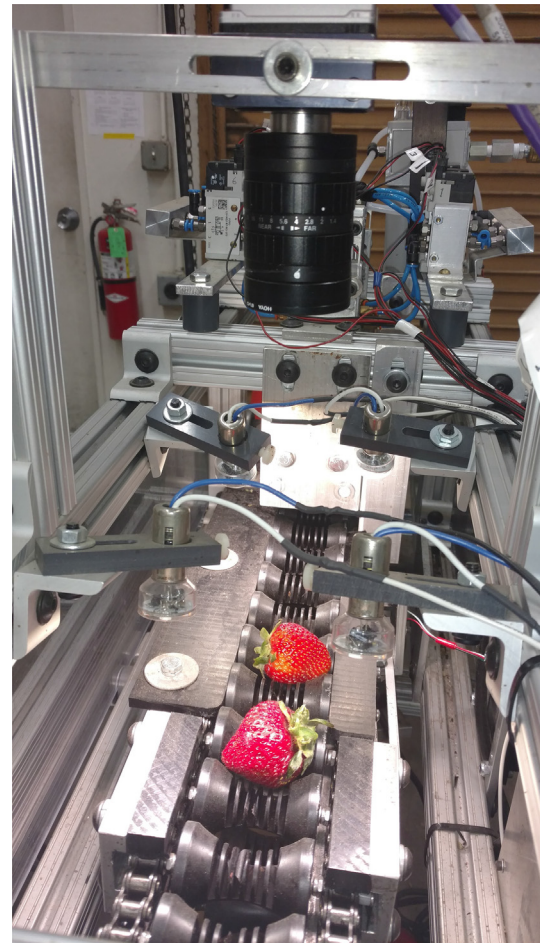


Fig. 3. View of the conveyor, camera and lights.

marized in the flowchart presented in Fig. 5. Each step of the image processing pipeline is presented next, in full detail.

#### 2.3.1. Step #1

The camera field of view is set to be wide enough to include several pockets in each acquired image  $I$ . In order to process only one strawberry per image, the picture area of interest is reduced to the area containing a single pocket. As the conveyor runs at a constant speed, the area of interest is always the same. It is denoted by  $AoI = [U_{min}, U_{max}, V_{min}, V_{max}]$  and is calculated during a calibration pre-processing step.

Then, for computational speed reasons, the raw image size is reduced by a ratio  $k_{Res}$ . This leads to an image  $I_{Res}$  whose dimensions are  $\left[ \frac{U_{max}-U_{min}}{k_{Res}}, \frac{V_{max}-V_{min}}{k_{Res}} \right]$  (Fig. 11(a)).

#### 2.3.2. Step #2

In order to extract the pixels corresponding to the flesh and the calyx, a novel approach is used, which extends the one presented in Liming and Yanchao (2010) and Hayashi et al. (2011). In these works, the authors use the color space  $R - G$ , which corresponds to the subtraction of the green layer from the red one. The approach proposed in this work is slightly different and uses two color spaces:  $R - G$  and  $G - B$ . The second one is obtained by subtracting the blue layer of the image from the green one. Thus, it is possible to obtain the gray level images  $I_{R-G}$  and  $I_{G-B}$  which respectively represent the first extraction step of the flesh (Fig. 11(b)) and of the whole fruit (Fig. 11(c)).

<sup>1</sup> The database is available at <https://doi.org/10.25338/B8WC70>.



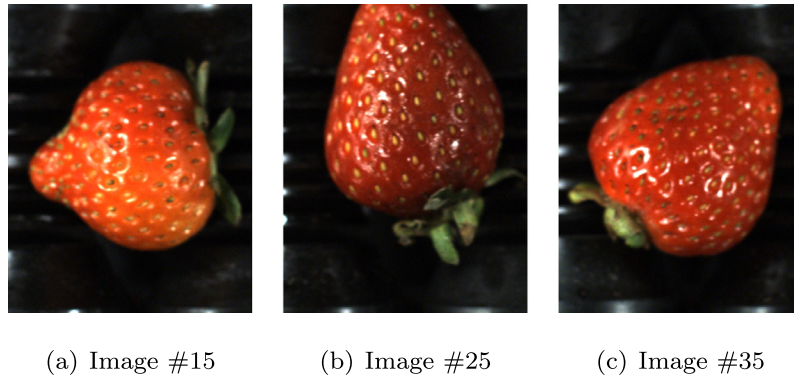


Fig. 4. Example of images.

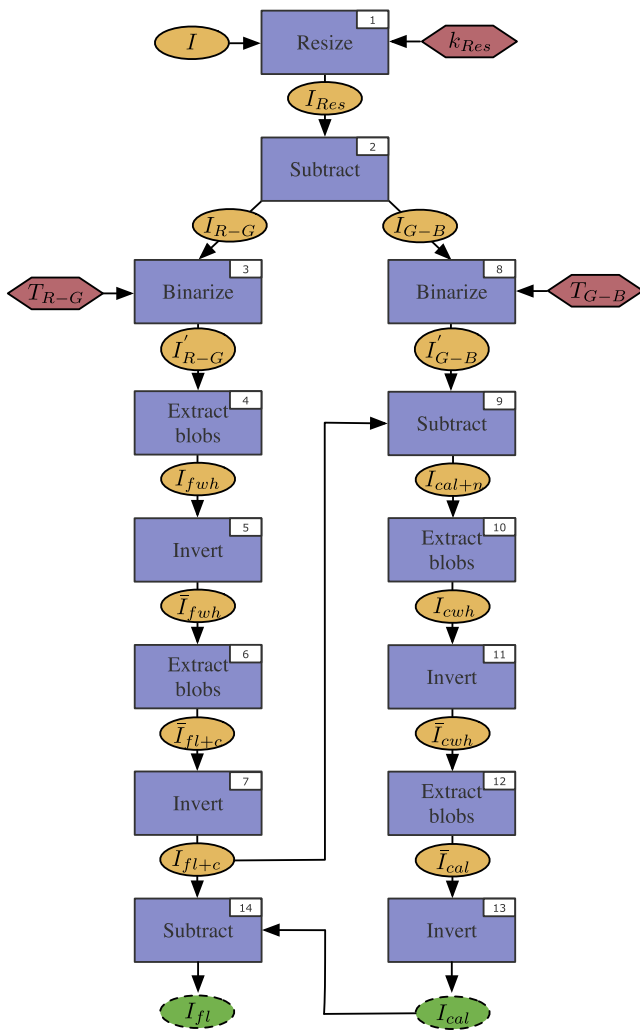
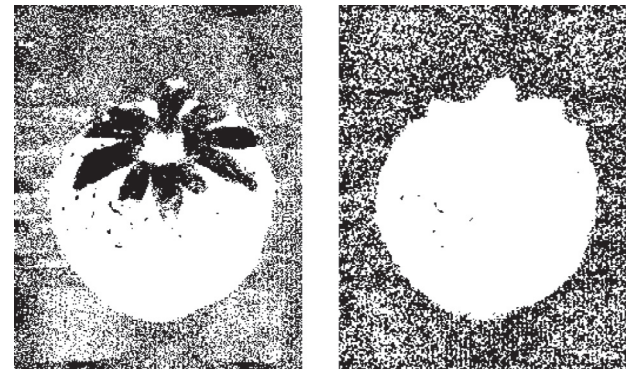


Fig. 5. Flowchart of the image processing. Ovals: Inputs/Outputs – Rectangles: Image processing – Hexagons: User defined parameters.

### 2.3.3. Steps #3 and #8

The obtained images  $I_{R-G}$  and  $I_{G-B}$  might look like a clean extraction of the pixels of interest but they actually contain a large amount of pixels representing the background. To highlight this problem, the images Fig. 6(a) and (b) have been computed by setting all pixels with non-zero values to 255. In order to remove most of the background pixels,  $I_{R-G}$  and  $I_{G-B}$  are binarized using



(a)  $I_{R-G}$  with pixels at 255 (b)  $I_{G-B}$  with pixels at 255

Fig. 6. Example of issues after subtraction.

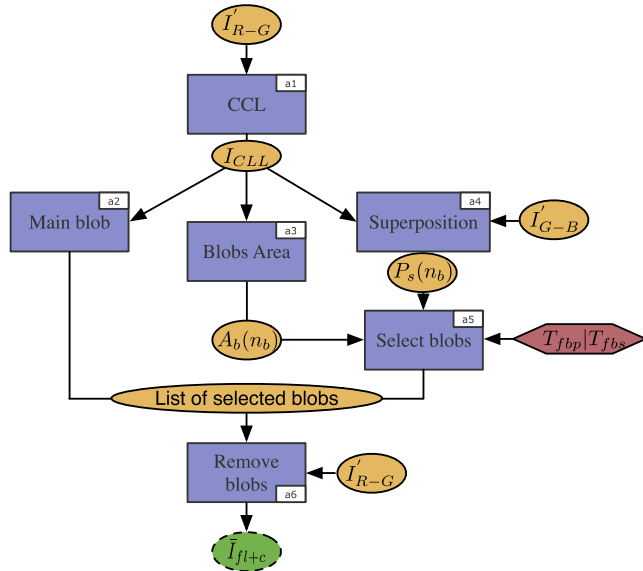
respective thresholds  $T_{R-G}$  and  $T_{G-B}$ . Thresholding leads to the binary images  $I'_{R-G}$  and  $I'_{G-B}$  (Fig. 11(d) and (e)).

### 2.3.4. Step #4

Selecting an efficient threshold to binarize one picture is not a difficult task. However, it becomes more challenging when the same threshold is used on a set of images. Indeed, the flesh and calyx colors are slightly different from one picture to another. For this reason, a simple static binarization process cannot accurately remove all the background pixels in a set of images, requiring extra image processing. In our proposed approach, image processing from this stage onward moves from the pixel level to the blob level. A blob is defined as a set of 8-connected pixels.<sup>2</sup> This approach has been selected for two reasons. First, it allows the use of higher level features, (e.g. size, distance), to select or remove the remaining pixels. Second, contrary to morphological operations, it does not alter the initial area and hence accurately preserves the shape.

Based on the previous analysis, the next step consists of computing the blobs present in the image and in identifying those belonging to the areas of flesh. The steps used to perform those tasks are summarized in the flowchart presented in Fig. 7. In order to create the blobs, the *Connected Component Labeling* (CCL) (Suzuki et al., 2003) algorithm is applied to the image  $I'_{R-G}$ . It leads to the image  $I_{CCL}$  whose connected pixels have a similar value (Fig. 11

<sup>2</sup> 8-connected pixels are neighbors to every pixel that touches one of their edges or corners.



**Fig. 7.** Flowchart of the flesh extraction Ovals: Inputs/Outputs – Rectangles: Image processing – Hexagons: User defined parameters.

(f)). Thus the pixels are assigned to blobs and it is now possible to work with the  $N$  computed blobs.

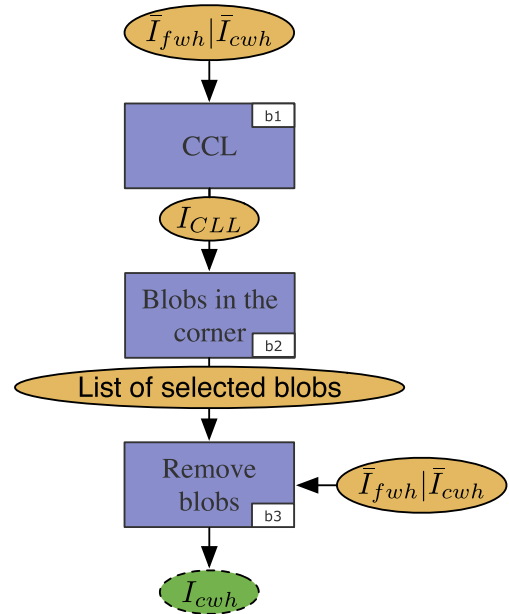
At this stage, it is necessary to identify the blobs belonging to the areas of flesh. To do so, the blob with the largest area is considered to be the main one representing the flesh. For example, in picture Fig. 11(f), blob number 27 has been selected as the main one. Most of the time this blob actually corresponds to the whole flesh area. Unfortunately, as it can be seen in picture Fig. 11(f), the flesh could be split up into several blobs because of the calyx leaves covering the flesh. In order to detect the small parts of the flesh, a feature  $P_s(n_b)$  is computed for each blob, with  $n_b$  taking values in the range  $[1, \dots, N]$ . It corresponds to the percentage of pixels in a blob that have a non-zero value in  $I'_{G-B}$ . Indeed, as  $I'_{G-B}$  supposedly contains the entire fruit and  $I'_{R-G}$  only the flesh, the pixels having a non-zero value in both  $I'_{G-B}$  and  $I'_{R-G}$  should belong to the flesh part. This assumption is based on the hypothesis that the background pixels in  $I'_{G-B}$  are not the same as the ones in  $I'_{R-G}$ . Finally, to decide if a blob corresponds to the flesh, two thresholds are used. The first one,  $T_{fbp}$ , represents the minimal percentage of pixels  $P_s(n_b)$  required for a blob in order to be considered as flesh. The second one,  $T_{fbs}$ , is the minimal size required for a blob to be assigned to flesh. For example, in picture Fig. 11(f), blobs number 5, 6, 11 and 23 satisfy both criteria and have been selected as being part of the flesh. Thus the final step of this part consists of removing in  $I'_{R-G}$  all the pixels belonging to blobs identified as background. The obtained image  $I_{fwh}$  (see Fig. 11(g)) represents the extracted flesh.

### 2.3.5. Step #5

$I_{fwh}$  represents the extracted flesh after removing the background pixels. However, some pixels belonging to the flesh area are still considered as background pixels. They could also be seen as the holes in the flesh. To overcome this problem, it is necessary to fill the holes in the flesh. One more time a blob-based approach is proposed and applied on  $\bar{I}_{fwh}$ . For this reason  $\bar{I}_{fwh}$  (see Fig. 11(h)), which is the inverse image of  $I_{fwh}$ , is computed.

### 2.3.6. Step #6

The flowchart used to fill the holes is presented in Fig. 8. It can be seen that, as before, in order to not consider pixels indepen-



**Fig. 8.** Flowchart to fill the holes Ovals: Inputs/Outputs – Rectangles: Image processing.

dently, the blobs in  $\bar{I}_{fwh}$  are computed using the CCL algorithm. This operation leads to a new image  $I_{CLL}$  (see Fig. 11(i)) with  $N$  blobs. Then the blobs containing the image corner pixels with a non-zero value are identified as belonging to the background. The other pixels blobs are removed from  $\bar{I}_{fwh}$  leading to the image  $\bar{I}_{fl+c}$  (see Fig. 11(j)).

### 2.3.7. Step #7

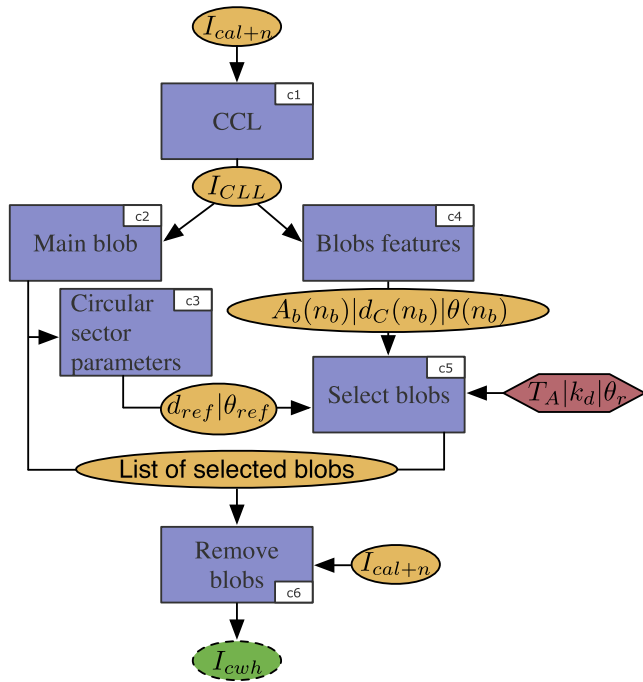
Now that the holes have been filled in  $\bar{I}_{fwh}$ , it is then necessary to invert  $\bar{I}_{fl+c}$  in order to obtain an image whose pixels of interest have a non-zero value. This operation provides the image  $I_{fl+c}$  (see Fig. 11(k)) which corresponds to the extracted flesh with the holes filled.

### 2.3.8. Step #9

The remaining steps of the image processing pipeline - including this one - focus on extracting the pixels corresponding to the calyx. As it was explained previously, the image  $I'_{G-B}$  corresponds to the entire fruit. In order to obtain only the calyx part of the image,  $I_{fl+c}$  is subtracted from  $I'_{G-B}$ . This leads to the image  $I_{cal+n}$  (see Fig. 12(a)) which contains the calyx and background pixels.

### 2.3.9. Step #10

In order to remove the background pixels, image processing is once again applied at the blob level. In the  $G - B$  color space, the range of blob size for background and calyx pixels overlap. Thus, a unique size threshold is not sufficient to separate the background pixels from the calyx. To overcome this problem, a heuristic position constraint is applied. Thus, in order to be considered as representing the calyx, a blob has to belong to a certain area, in this case a circular sector, and to have a certain size. The base of the circular sector is the flesh centroid. Moreover, the circular sector has to be oriented towards the calyx. To do so, the main calyx blob is identified and then used. Finally, once the circular sector is defined, it is used to identify the blobs representing the calyx. The flowchart in Fig. 9 presents the steps that accomplish the calyx blobs selection.



**Fig. 9.** Flowchart of the calyx extraction Ovals: Inputs/Outputs – Rectangles: Image processing – Hexagons: User defined parameters.

First, the CCL algorithm is applied to  $I_{cal+n}$  and leads to a new image  $I_{CCL}$  (see Fig. 12(b)) with  $N$  blobs. Next, in order to identify the main calyx blob, the area  $A_b(n_b)$  and centroid  $C_b(n_b)$  of each blob are computed, with  $n_b$  taking values in the range  $[1, \dots, N]$ . The main calyx blob corresponds to the one whose feature  $C(n_b)$  has the highest value.

$$C(n_b) = \frac{A_b(n_b)}{d_C(n_b)} \quad (1)$$

In (1),  $d_C(n_b)$  is the Euclidean distance between  $C_{fl}$  and  $C_b(n_b)$ , the centroid of the flesh and the  $n_b^{th}$  calyx blob respectively. The reasoning behind the feature defined by (1) is that the main blob has a large area and is close to the centroid of the flesh. The selection of the main blob is more difficult for the calyx than for the flesh. Indeed, in this color space, the area of a calyx blob could be similar to the background. This is the reason why the distance to the flesh centroid is taken into account when selecting the main blob.

After identifying the first blob, it is then required to define the circular sector of search. To achieve this, two values are computed: the Euclidean distance  $d_{ref}$  between the centroids of the flesh and the main blob, and  $\theta_{ref}$ , the orientation of the vector linking the centroids. Finally, the circular sector is defined by using the computed values as well as  $k_d$  and  $\theta_r$ , the length ratio and opening angle respectively (see Fig. 10). Thus, if the centroid of a blob is within the circular sector and if the blob is large enough, then the blob is identified as representing the calyx. In other words, the  $n_b^{th}$  blob is considered as calyx if:

$$d_C(n_b) < d_{ref} * k_d \quad (2)$$

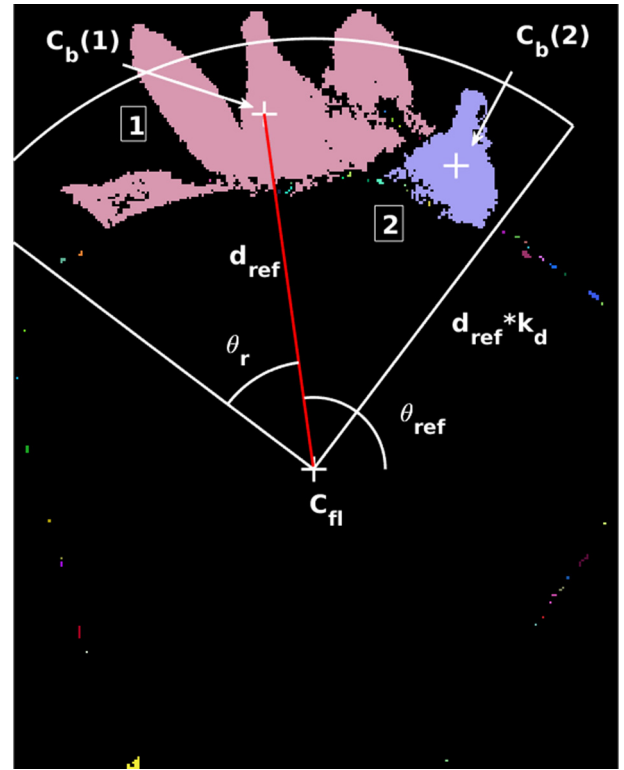
and

$$\theta_{ref} - \theta_r < \theta(n_b) < \theta_{ref} + \theta_r \quad (3)$$

and

$$A_b(n_b) > T_A \quad (4)$$

where  $\theta(n_b)$  is the orientation of the vector linking the flesh and the  $n_b^{th}$  blob centroids, and  $T_A$  is a threshold representing the



**Fig. 10.** Example of blob selection for the calyx.

minimum required area for a blob to be considered as calyx. In Fig. 10, the calyx blobs selection process is illustrated using a strawberry other than the one previously used as an example. In this case, the calyx is represented by blobs #1 and #2. Blob #1 is selected as the main one, based on criterion (1). Then  $d_{ref}$  and  $\theta_{ref}$  are computed and used to create the search circular sector. As it can be seen, blob #1 is the only one large enough, with its centroid inside the circular sector to be considered as calyx.

Using the calyx blobs selection leads to image  $I_{cwh}$  (see Fig. 12(c)). Similar to the flesh case, the extracted blobs may contain holes. In order to improve the calyx extraction, it is necessary to fill these holes.

### 2.3.10. Step #11

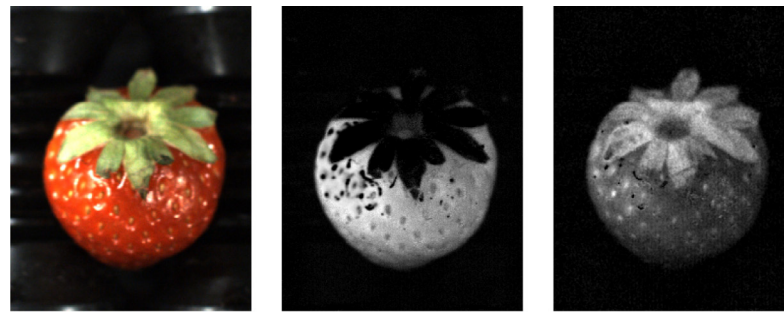
$I_{cwh}$  represents the extracted calyx, which may contain holes. To overcome this problem, a similar approach to the previous one is applied on  $\bar{I}_{cwh}$ . For this reason,  $\bar{I}_{cwh}$  (see Fig. 12(d)), which is the inverse image of  $I_{cwh}$ , is computed.

### 2.3.11. Step #12

As before, the steps used to fill the holes are summarized in the flowchart presented in Fig. 8. First, avoid relying simply on pixel-level processing, the blobs in  $\bar{I}_{cwh}$  are computed using the CCL algorithm. This operation leads to a new image  $I_{CCL}$  (see Fig. 12(e)) with  $N$  blobs. Then the blobs containing any corner pixels with non-zero values are identified as belonging to the background. The other pixels belonging to blobs are removed from  $\bar{I}_{cwh}$  leading to the image  $\bar{I}_{cal}$  (see Fig. 12(f)).

### 2.3.12. Step #13

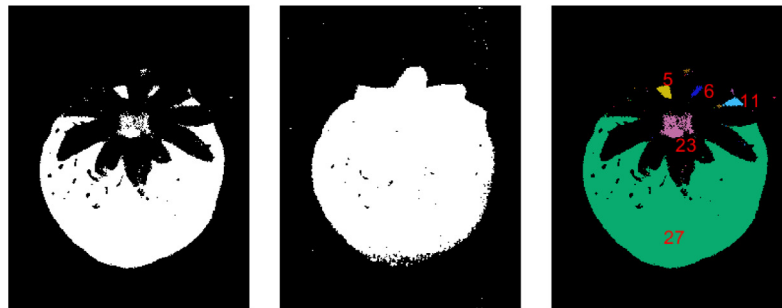
Now that the holes in  $\bar{I}_{cwh}$  have been filled, it is then necessary to invert  $\bar{I}_{cal}$ . This operation provides the image  $I_{cal}$  (see Fig. 12(g)), which corresponds to the extracted calyx with the holes filled.



(a) Resized image  $I_{Res}$   
with  $K_{Res} = 2$

(b)  $I_{R-G}$

(c)  $I_{G-B}$



(d)  $I'_{R-G}$  obtained with  $T_{R-G} = 10$

(e)  $I'_{G-B}$  obtained with  $T_{G-B} = 5$

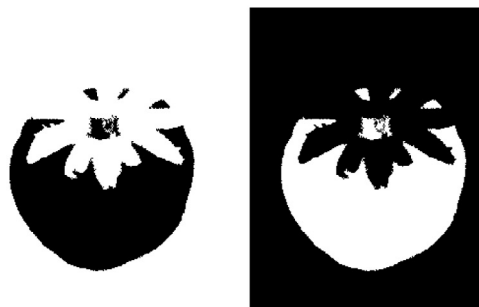
(f)  $I_{CLL}$  obtained for  $I'_{R-G}$



(g)  $I_{fwh}$

(h)  $\bar{I}_{fwh}$

(i)  $I_{CLL}$  obtained for  $\bar{I}_{fwh}$



(j)  $\bar{I}_{fl+c}$

(k)  $I_{fl+c}$

**Fig. 11.** Example of image processing - Part 1.

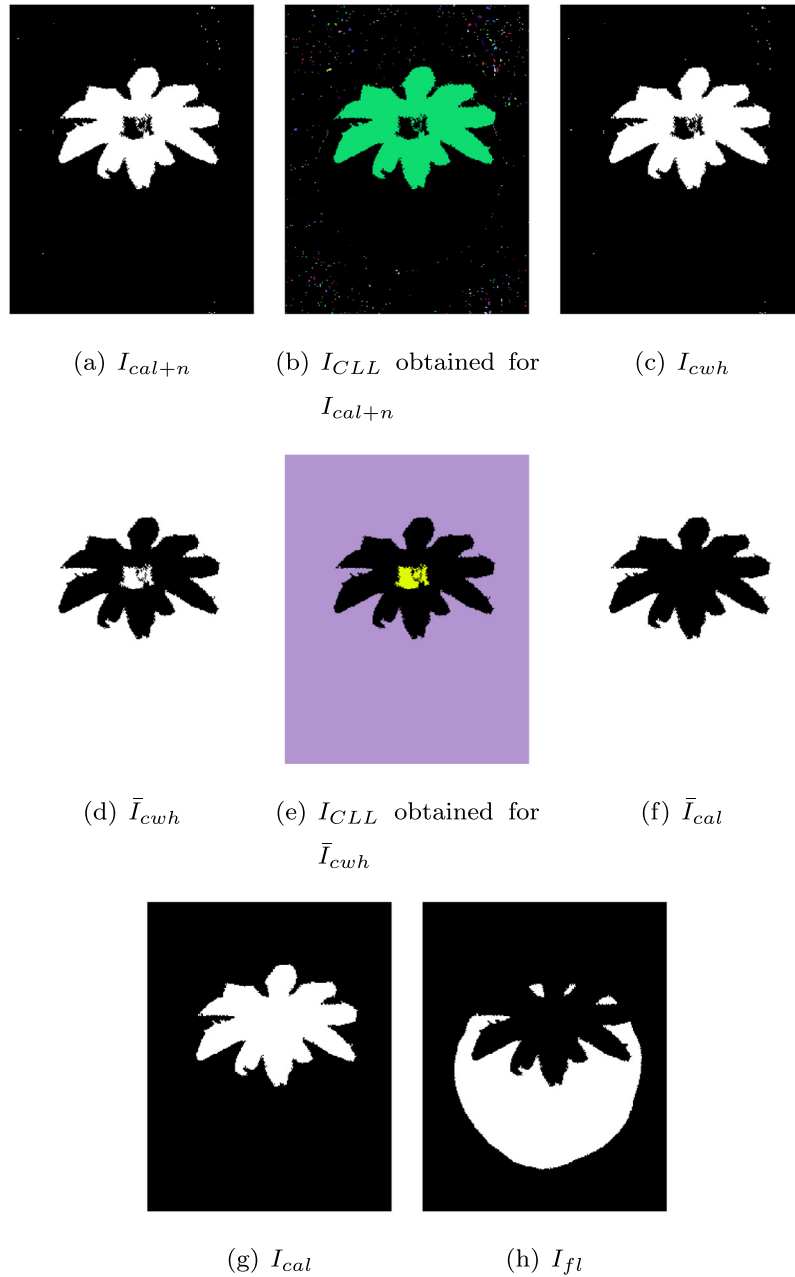


Fig. 12. Example of image processing - Part 2.

#### 2.3.13. Step #14

The final operation consists of removing from  $I_{fl}$  the parts belonging to the calyx. Indeed, as it can be seen in Fig. 11(k), the flesh blobs selection might have selected some calyx. This is due to the sensitivity of the process to the binarization thresholds. To do so,  $I_{cal}$  is subtracted from  $I_{fl}$ . This results in image  $I_{fl}$ , which should contain only pixels belonging to the strawberry's flesh.

#### 2.4. Metrics

In order to measure the performance of the image processing algorithm, a binary image  $I_1$  computed by the image processing algorithm is compared with a binary ground truth image  $I_2$ .<sup>3</sup> To do so, two metrics  $S(I_1, I_2)$  and  $N(I_1, I_2)$  are defined as:

$$S(I_1, I_2) = \frac{\sum [P_{u,v}(I_1) * P_{u,v}(I_2)]}{\sum P_{u,v}(I_2)} * 100 \quad (5)$$

$$N(I_1, I_2) = \frac{\sum [P_{u,v}(I_1) * \bar{P}_{u,v}(I_2)]}{\sum \bar{P}_{u,v}(I_2)} * 100 \quad (6)$$

In Eqs. (5) and (6),  $P_{u,v}(I_1)$  corresponds to the value (either 0 or 1) of the pixel at coordinates  $u, v$  in image  $I_1$ . Moreover  $\bar{P}_{u,v}(I_1) = 1 - P_{u,v}(I_1)$ . Thus  $S(I_1, I_2)$  computes the percentage of pixels in  $I_2$  belonging to the area of interest that have been extracted in  $I_1$ .  $S(I_1, I_2)$  can be considered as the percentage of flesh or calyx pixels extracted in  $I_1$ .  $N(I_1, I_2)$  gives the percentage of pixels that have been extracted in  $I_1$  but do not belong to the area of interest in  $I_2$ .  $N(I_1, I_2)$  can be considered as the percentage of background pixels extracted in  $I_1$ . In the case of a perfect extraction,

<sup>3</sup> The process to obtain the ground truth images is described in Section 3.1.



$S(I_1, I_2) = 100$  and  $N(I_1, I_2) = 0$ . Finally, in order to obtain a metric using more than one image to measure the extraction performance, we define:

$$\widehat{S}(I_1, I_2) = \frac{\sum_{k=0}^{k=M} S(I_1(k), I_2(k))}{M} \tag{7}$$

$$\widehat{N}(I_1, I_2) = \frac{\sum_{k=0}^{k=M} N(I_1(k), I_2(k))}{M} \tag{8}$$

where  $M$  is the number of used images and  $I_1(k)$  or  $I_2(k)$  are obtained using the  $k$ th image of the set of images. Thus  $\widehat{S}(I_1, I_2)$  and  $\widehat{N}(I_1, I_2)$  provide a mean value of the algorithm performance, evaluated over a set of  $M$  images.

### 3. Results and discussion

As it has been shown in the previous section, the proposed algorithm processes a strawberry image obtained from a camera installed on a de-capping machine. The images were obtained under normal conditions of use, *i.e.*, when the conveyor runs at 20 cm/s. The algorithm extracts the areas corresponding to the flesh and the calyx. It mostly relies on a color based segmentation and several blob selection steps. The image processing pipeline utilizes several parameters that must be provided by the user. This section investigates the extraction performance when using various sets of values for the parameters.

#### 3.1. Methodology

To measure the extraction performance, the database of 250 color images - each containing a strawberry (see an example in Fig. 13) - was used to build binary ground truth images. Each picture  $\tilde{I}(i_{db})$ , with  $i_{db}$  taking values in the range  $[1, 250]$ , was manually

processed with GIMP software to extract the areas  $\tilde{A}_f(i_{db})$  and  $\tilde{A}_c(i_{db})$ , which correspond to the ground truth for the flesh and the calyx respectively (see Fig. 13(b) and (c)). Next, the obtained color images were processed to compute two binary images  $\tilde{I}_{fb}(i_{db})$  and  $\tilde{I}_{cb}(i_{db})$  whose non-zero pixels correspond to the flesh and the calyx respectively (see Fig. 13(d) and (e)).

The set of 250 binary images  $\tilde{I}_{fb}(i_{db})$  and  $\tilde{I}_{cb}(i_{db})$ , and metrics (7) and (8) were then used to evaluate the extraction performance at different steps of the image processing pipeline. Thus, by measuring the percentage of extracted pixels belonging to the area of interest or to the background, it is possible to evaluate the impact of the user-defined parameters on the extraction at different steps. Table 1 summarizes the selected steps, the investigated parameters and the images used to compute the metrics.

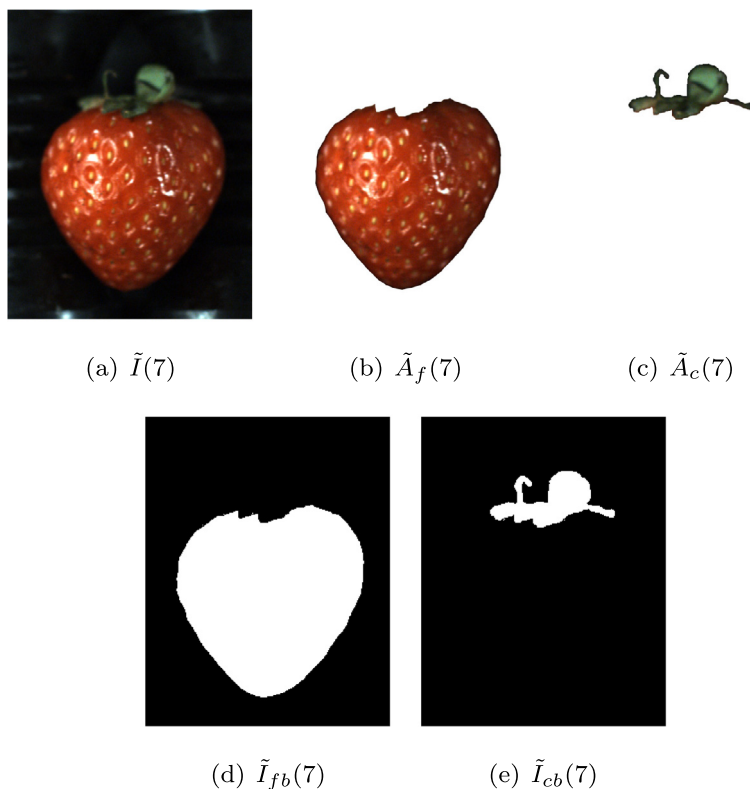
#### 3.2. Parameter sensitivity analysis and optimization

##### 3.2.1. Binarization

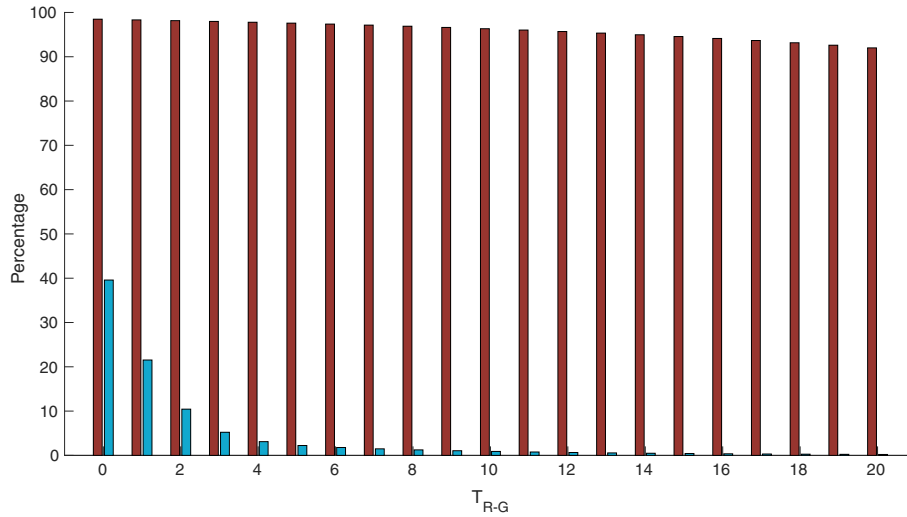
The image binarization steps of the algorithm are the first ones which require user-defined parameters (steps 3 and 8 in Fig. 5). Indeed, the binarization is performed using  $T_{R-G}$  and  $T_{G-B}$  as thresholds. In order to evaluate their influence on the extraction process, Figs. 14 and 15 present binarization results for different values for  $T_{R-G}$  and  $T_{G-B}$ .

**Table 1**  
List of steps investigated.

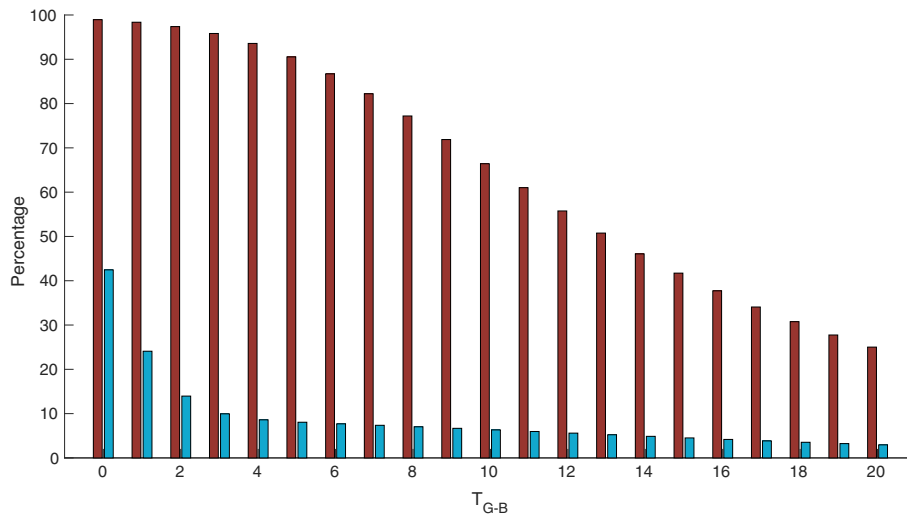
Steps	Parameters	$I_1$
[3][8] Binarization	$T_{R-G}, T_{G-B}$	$I'_{R-G}, I'_{G-B}$
[4] Selection of Flesh Blobs	$T_{fbp}, T_{fbs}$	$I_{fwh}, I_{fl+c}$
[9] Selection of Calyx Blobs	$T_A, k_d, \theta_r$	$I_{cwh}, I_{cal}$



**Fig. 13.** Example of ground truth database for one strawberry.



**Fig. 14.**  $\hat{S}(I'_{R-G}, \tilde{I}_{fb})$  (red) -  $\hat{N}(I'_{R-G}, \tilde{I}_{fb})$  (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 15.**  $\hat{S}(I'_{G-B}, \tilde{I}_{fb} + \tilde{I}_{cb})$  (red) -  $\hat{N}(I'_{G-B}, \tilde{I}_{fb} + \tilde{I}_{cb})$  (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

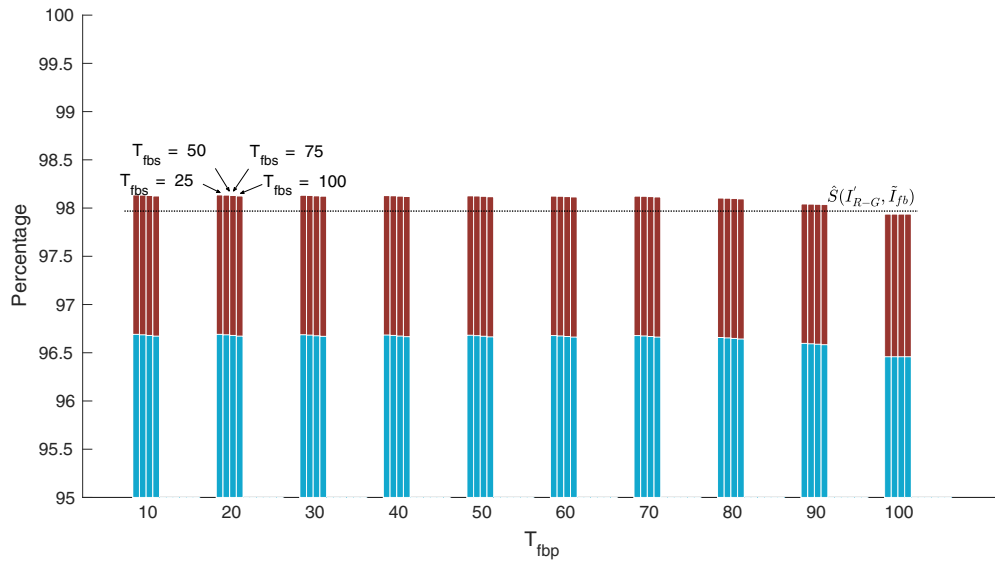
In Fig. 14, the percentage of flesh  $\hat{S}(I'_{R-G}, \tilde{I}_{fb})$  and the background pixels  $\hat{N}(I'_{R-G}, \tilde{I}_{fb})$  that have been extracted are given for threshold  $T_{R-G}$  taking values in the range  $[0, 20]$  with a step value of 1. This is used to analyze the influence of  $T_{R-G}$  when binarizing  $I_{R-G}$  in order to extract the area corresponding to the flesh. As it can be seen,  $\hat{S}(I'_{R-G}, \tilde{I}_{fb})$  barely changes when using different values for  $T_{R-G}$ . However,  $\hat{N}(I'_{R-G}, \tilde{I}_{fb})$  strongly decreases when  $T_{R-G}$  increases. This suggests that selecting a sufficiently high value of  $T_{R-G}$  in order to extract a large percentage of the area of interest can be accomplished without having too many background pixels.

Fig. 15 presents the results of the binarization of  $I_{G-B}$  for  $T_{G-B}$  taking values in the range  $[0, 20]$ . This process aims to extract the whole fruit and this is the reason why the reference area of interest is obtained by adding the non-zero pixels of  $\tilde{I}_{fb}$  and  $\tilde{I}_{cb}$ . As it can be seen in the figure, both the percentage of fruits pixels  $\hat{S}(I'_{G-B}, \tilde{I}_{fb} + \tilde{I}_{cb})$  and background pixels  $\hat{N}(I'_{G-B}, \tilde{I}_{fb} + \tilde{I}_{cb})$  decrease when  $T_{G-B}$  increases. Hence,  $T_{G-B}$  should be selected based on the final application needs. If it is required to extract as much fruit as possible, then  $T_{G-B}$  should be chosen between 1 and 3. If it is

necessary to extract a decent amount of fruit while having a low percentage of background pixels, then  $T_{G-B}$  should be chosen between 4 and 6. This step of the image processing algorithm requires the user to select a value for  $T_{G-B}$  based on their expectation for the  $\hat{S}(I'_{G-B}, \tilde{I}_{fb} + \tilde{I}_{cb}) / \hat{N}(I'_{G-B}, \tilde{I}_{fb} + \tilde{I}_{cb})$  ratio.

### 3.2.2. Selection of the flesh blobs

The second part of the analysis aims to highlight the sensitivity of the image processing to the parameters  $T_{fbp}$  and  $T_{fbs}$  when selecting the blobs of interest in  $I'_{R-G}$ . This process corresponds to step 4, which is also detailed in Figs. 5 and 7. For each set,  $T_{fbp}$  takes values in the range  $[10, 100]$  with a step value of 10 and  $T_{fbs}$  takes values in the range  $[25, 100]$  with a step value of 25. Thus, each figure presents results for 40 different combinations of  $T_{fbp}$  and  $T_{fbs}$ . Moreover, in the following figures, the results obtained after filling the holes in the selected blobs (steps 5, 6 and 7 in Fig. 5) are given. Even though those processes do not use user-defined parameters, these results assess the overall performances of the image processing algorithm.



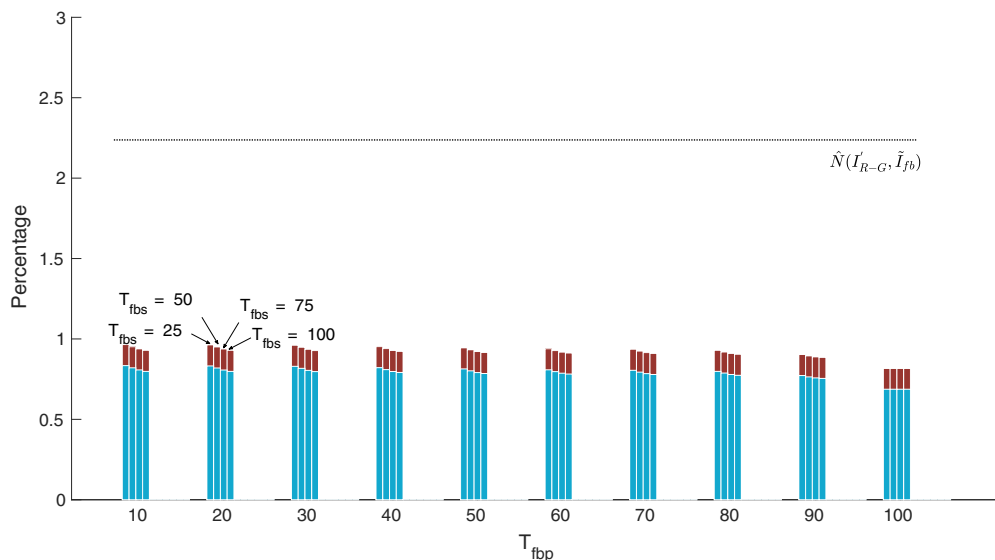
**Fig. 16.**  $\hat{S}(I_{fwh}, \tilde{I}_{fb})$  (blue) -  $\hat{S}(I_{fl+c}, \tilde{I}_{fb})$  (red) -  $T_{R-G} = 10$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The first set of results is given for  $I'_{R-G}$ , and is obtained with  $T_{R-G} = 10$ . In Fig. 16,  $\hat{S}(I_{fwh}, \tilde{I}_{fb})$ , step #4, and  $\hat{S}(I_{fl+c}, \tilde{I}_{fb})$ , step #6, are given for the different combinations of  $T_{fbp}$  and  $T_{fbs}$ . Moreover the value of  $\hat{S}(I'_{R-G}, \tilde{I}_{fb})$ , step #3, obtained with  $T_{R-G} = 10$  is plotted using a blacked dashed line. This permits visualizing the impact of the current steps on the previous results. As it can be seen in Fig. 16, the blobs selection process is barely sensitive to the variations of parameters  $T_{fbp}$  and  $T_{fbs}$ . Indeed  $\hat{S}(I_{fwh}, \tilde{I}_{fb})$  drops at almost the same level despite using different values for  $T_{fbp}$  or  $T_{fbs}$ . In the same way, when filling the holes inside the blobs,  $\hat{S}(I_{fl+c}, \tilde{I}_{fb})$  increases by almost the same amount for any combination of  $T_{fbp}$  or  $T_{fbs}$ .

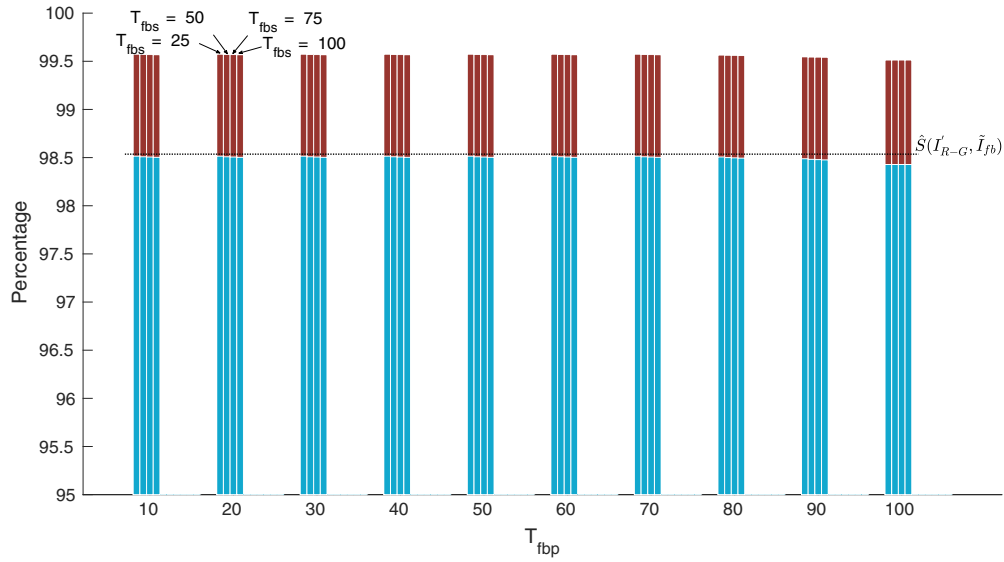
Fig. 17 displays  $\hat{N}(I_{fwh}, \tilde{I}_{fb})$  and  $\hat{N}(I_{fl+c}, \tilde{I}_{fb})$ , as well as  $\hat{N}(I'_{R-G}, \tilde{I}_{fb})$  obtained with  $T_{R-G} = 10$  plotted using a blacked dashed line. In this case, the same observation about the evolution of  $\hat{N}(I_{fwh}, \tilde{I}_{fb})$  and  $\hat{N}(I_{fl+c}, \tilde{I}_{fb})$  can be done. We can conclude that, in this test, the

selection of flesh blobs is rather insensitive to the parameters values. However, the advantages of performing a blobs selection followed by a holes filling process are highlighted. Indeed in most of the cases, the final value of  $\hat{S}(I_{fl+c}, \tilde{I}_{fb})$  is higher than the initial one  $\hat{S}(I'_{R-G}, \tilde{I}_{fb})$ , whereas the percentage of background pixels  $\hat{N}(I_{fl+c}, \tilde{I}_{fb})$  is significantly lower than the initial one  $\hat{N}(I'_{R-G}, \tilde{I}_{fb})$ . Adding those steps after binarization extracts more pixels belonging to the area of interest while reducing the amount of background pixels.

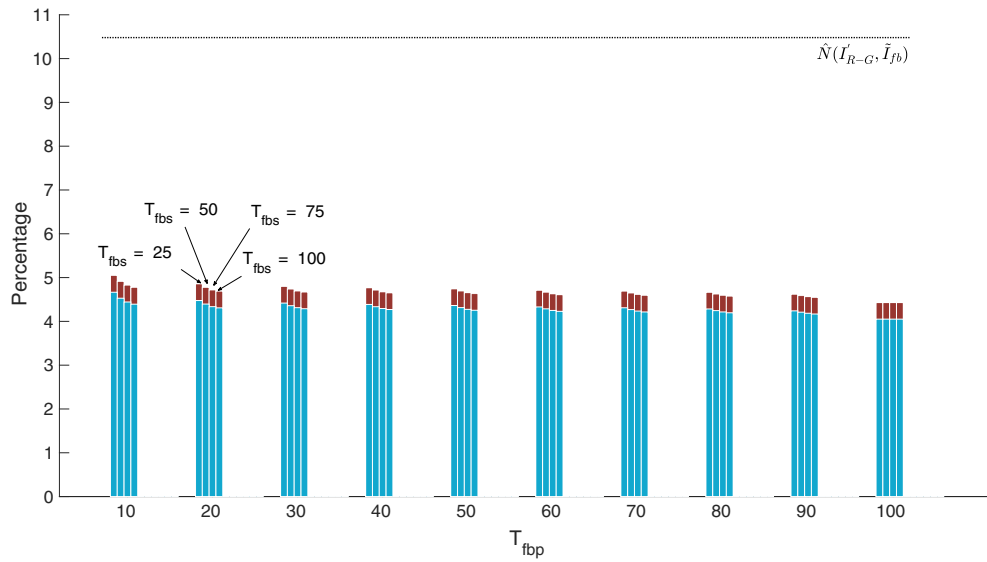
In order to check that the previous results do not depend on the value of  $T_{R-G}$ , a similar set of tests, with  $I'_{R-G}$  computed with  $T_{R-G} = 2$ , is performed. As it can be seen in Figs. 18 and 19, the flesh blobs selection and holes filling processes are rather sensitive to the variations of parameters  $T_{fbp}$  and  $T_{fbs}$ . Moreover those steps improve the quality of the flesh extraction by increasing  $\hat{S}(I_{fwh}, \tilde{I}_{fb})$  by 1% while decreasing  $\hat{S}(I_{fl+c}, \tilde{I}_{fb})$  by more than 5%.



**Fig. 17.**  $\hat{N}(I_{fwh}, \tilde{I}_{fb})$  (blue) -  $\hat{N}(I_{fl+c}, \tilde{I}_{fb})$  (red) -  $T_{R-G} = 10$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 18.**  $\hat{S}(I_{fbwh}, \tilde{I}_{fb})$  (blue) –  $\hat{S}(I_{fb+c}, \tilde{I}_{fb})$  (red) –  $T_{R-G} = 2$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 19.**  $\hat{N}(I_{fbwh}, \tilde{I}_{fb})$  (blue) –  $\hat{N}(I_{fb+c}, \tilde{I}_{fb})$  (red) –  $T_{R-G} = 2$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

### 3.2.3. Selection of the calyx blobs

In this third part, an analysis of the sensitivity of the calyx blobs selection with respect to the three parameters  $T_A$ ,  $k_d$  and  $\theta_r$  is provided. This corresponds to step 10, which is also detailed in Figs. 5 and 9. For each set,  $T_A$  takes values from the set  $[0, 30, 70]$ ,  $k_d$  from  $[1, 1.2, 1.4, 1.6, 1.8]$  and  $\theta_r$  from  $[30, 45, 60, 75, 90]$ . Thus each figure presents results for 75 different combinations of  $T_A$ ,  $k_d$  and  $\theta_r$ . Moreover in the following figures, the results obtained after filling the holes in the selected blobs (steps 11, 12 and 13 in Fig. 5) are given.

In Fig. 5 it can be seen that the calyx blobs selection uses the binary calyx image  $I_{cal+n}$  as input. As  $I_{cal+n}$  is obtained by subtracting  $I_{fb+c}$  to  $I'_{G-B}$ , it is mandatory to specify which parameters have been used to obtain  $I_{fb+c}$  and  $I'_{G-B}$ . For the following results  $I_{fb+c}$  is computed with  $T_{R-G} = 10$ ,  $T_{jsp} = 40$  and  $T_{fsb} = 50$ , whereas  $I'_{G-B}$  is obtained with  $T_{G-B} = 5$ .

In Figs. 20 and 21,  $\hat{S}(I_{cwh}, \tilde{I}_{cb})$ ,  $\hat{S}(I_{cal}, \tilde{I}_{cb})$ ,  $\hat{N}(I_{cwh}, \tilde{I}_{cb})$  and  $\hat{N}(I_{cal}, \tilde{I}_{cb})$  are given for the different combinations of  $T_A$ ,  $k_d$  and  $\theta_r$ . Moreover  $\hat{S}(I_{cal+n}, \tilde{I}_{cb})$  (Fig. 20) and  $\hat{N}(I_{cal+n}, \tilde{I}_{cb})$  (Fig. 21), obtained with  $T_{R-G} = 10$  and  $T_{G-B} = 5$ , are plotted using a blacked dashed line.

The first parameter to be addressed is  $T_A$ . As it can be seen in Figs. 20 and 21,  $\hat{S}(I_{cwh}, \tilde{I}_{cb})$ ,  $\hat{S}(I_{cal}, \tilde{I}_{cb})$ ,  $\hat{N}(I_{cwh}, \tilde{I}_{cb})$  and  $\hat{N}(I_{cal}, \tilde{I}_{cb})$  do not significantly change when different values of  $T_A$  are used. This can be explained by the fact that this parameter helps remove small blobs. Thus, its impact on the selected metrics is quite difficult to perceive. However, to minimize the amount of background pixels, it is advised to use at least 30 as a value for  $T_A$ . In Section 2.3.9, it was shown that the parameters  $k_d$  and  $\theta_r$  define a circular sector of search. Based on the results displayed in Figs. 20 and 21, it appears that the circular sector has to reach a certain size in order to guarantee that  $\hat{S}(I_{cal}, \tilde{I}_{cb})$  is larger than  $\hat{S}(I_{cal+n}, \tilde{I}_{cb})$ . Based



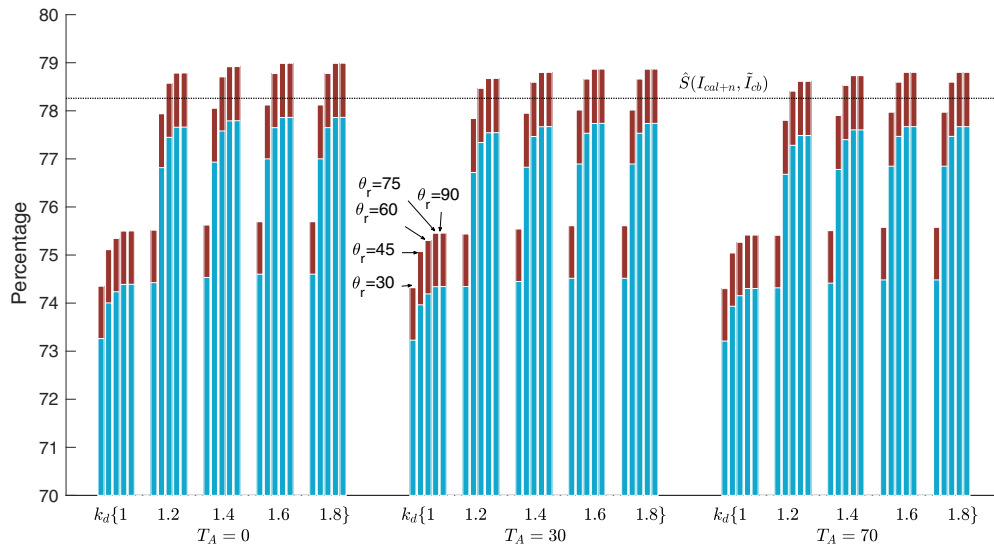
on the figures, the smallest circular sector providing acceptable results is obtained with  $k_d = 1.2$  and  $\theta_r = 60$  deg. Moreover, by analyzing  $\hat{S}(I_{cal}, \tilde{I}_{cb})$ , it can be seen that using a larger circular sector does not significantly increase the amount of background pixels.

As previously done for the flesh, in order to check that the previous analysis does not depend on the values of  $T_{R-G}$  and  $T_{G-B}$ , a similar set of tests, with  $T_{G-B} = T_{R-G} = 2$ , is performed. As it can be seen in Figs. 22 and 23, the same conclusions as before can be reached.  $T_A$  barely influences the results, whereas  $k_d = 1.2$  and  $\theta_r = 60$  deg are the minimal dimensions of the circular sector of search in order to obtain acceptable results. Moreover, increasing the size of the circular sector does not significantly modifies the results. However because the initial calyx extraction using  $T_{G-B} = T_{R-G} = 2$  is not as good as the previous one with  $T_{G-B} = 5$  and  $T_{R-G} = 10$ ,  $\hat{S}(I_{cal}, \tilde{I}_{cb})$  drops by 10% whereas the percentage of false-positive detection doubles. Even if using the blobs selection improves the accuracy of the extracted area, it is mandatory to first

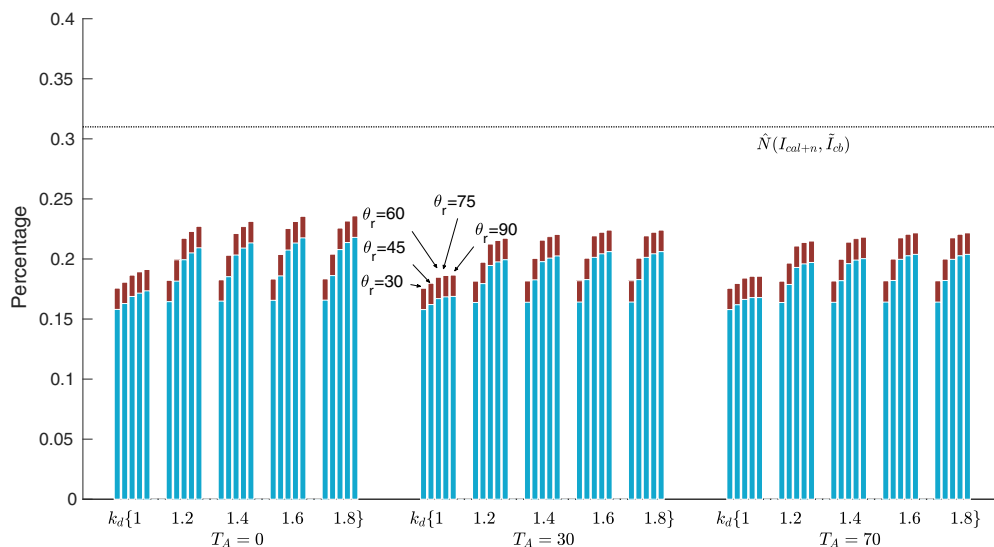
extract the areas of interest using the best thresholds  $T_{G-B}$  and  $T_{R-G}$  during the binarization step.

### 3.3. Examples of performance results and comparison with other approaches

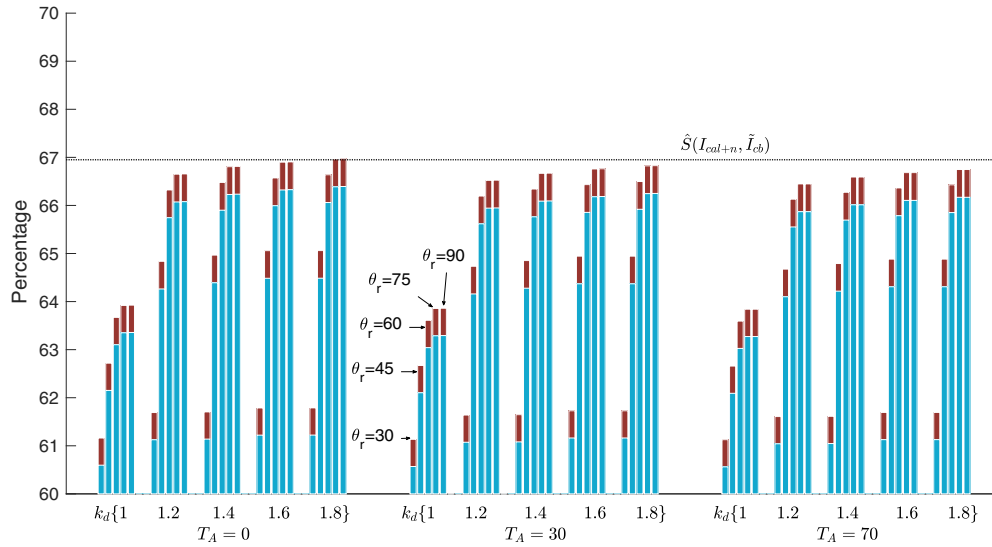
In this section, the algorithm has been tuned to minimize the number of background pixels while extracting as much flesh and calyx as possible. Based on the previous analysis, the following parameters have been used:  $T_{R-G} = 10$ ,  $T_{G-B} = 5$ ,  $T_{fbp} = 40$ ,  $T_{fbs} = 50$ ,  $T_A = 30$ ,  $k_d = 1.4$  and  $\theta_r = 75$  deg. As it can be seen in the pictures in Fig. 24, the flesh and calyx were correctly extracted from the initial images. Moreover, using Eqs. (7) and (8) as well as the 250 images, it has been computed that it allows extracting more than 98% of the flesh area with 0.4% of the background. Finally, up to 79% of the calyx area with 0.2% of the background pixels can be extracted. The calyx was not fully extracted because



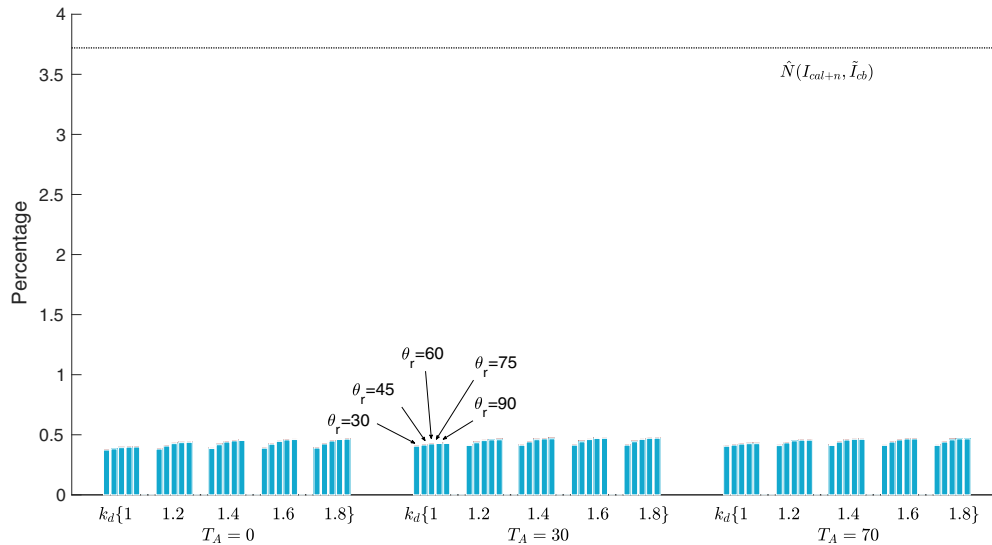
**Fig. 20.**  $\hat{S}(I_{cwh}, \tilde{I}_{cb})$  (blue) -  $\hat{S}(I_{cal}, \tilde{I}_{cb})$  (red) -  $T_{R-G} = 10$ ,  $T_{G-B} = 5$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 21.**  $\hat{N}(I_{cwh}, \tilde{I}_{cb})$  (blue) -  $\hat{N}(I_{cal}, \tilde{I}_{cb})$  (red) -  $T_{R-G} = 10$ ,  $T_{G-B} = 5$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 22.**  $\hat{S}(I_{cwh}, \tilde{I}_{cb})$  (blue) -  $\hat{S}(I_{cal}, \tilde{I}_{cb})$  (red) -  $T_{R-G} = 2, T_{G-B} = 2$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 23.**  $\hat{N}(I_{cwh}, \tilde{I}_{cb})$  (blue) -  $\hat{N}(I_{cal}, \tilde{I}_{cb})$  (red) -  $T_{R-G} = 2, T_{G-B} = 2$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the chosen value for  $T_{G-B} = 5$ . Indeed this choice minimizes the number of background pixels but does not maximize the calyx extraction. Concerning the remaining background pixels, displayed results show that they do not correspond to isolated blobs, and are not related in any way to the area of interest. They mostly represent pixels at the edge of the area of interest. The image processing algorithm provides promising results and correctly extracts the flesh and calyx despite different fruit orientations, sizes, colors, and shapes.

Then, the time to process the image was addressed. To do so four sets of threshold parameters were used:

- Case #1:  $T_{R-G} = 10, T_{G-B} = 5, T_{fbp} = 40, T_{fbs} = 50, T_A = 30, k_d = 1.4$  and  $\theta_r = 75$  deg.
- Case #2:  $T_{R-G} = 10, T_{G-B} = 5, T_{fbp} = 20, T_{fbs} = 30, T_A = 70, k_d = 1.2$  and  $\theta_r = 30$  deg.
- Case #3:  $T_{R-G} = 5, T_{G-B} = 4, T_{fbp} = 40, T_{fbs} = 50, T_A = 30, k_d = 1.4$  and  $\theta_r = 75$  deg.

- Case #4:  $T_{R-G} = 5, T_{G-B} = 3, T_{fbp} = 40, T_{fbs} = 50, T_A = 30, k_d = 1.4$  and  $\theta_r = 75$  deg.

The first case uses the set of parameters previously used to minimize the number of background pixels while extracting as much flesh and calyx as possible. The second case differs from the values used to select the flesh and calyx blobs. The last two ones differ from the thresholds values used to initially segment the image. Table 2 shows minimum, maximum, and mean processing times per frame, when the 250 images from the database are processed. It shows that the configuration of the first case does not require more than 20 ms to process the images. The results are similar when the parameters to select the blobs are changed (case # 2). The way blobs are selected does not significantly impact on the duration. However, the thresholds  $T_{R-G}$  and  $T_{G-B}$  have an impact on the processing time. Indeed, as it is seen with cases #3 and #4, when lower values are used, the obtained images contain more pixels. For this reason, the time to compute the blobs increases.

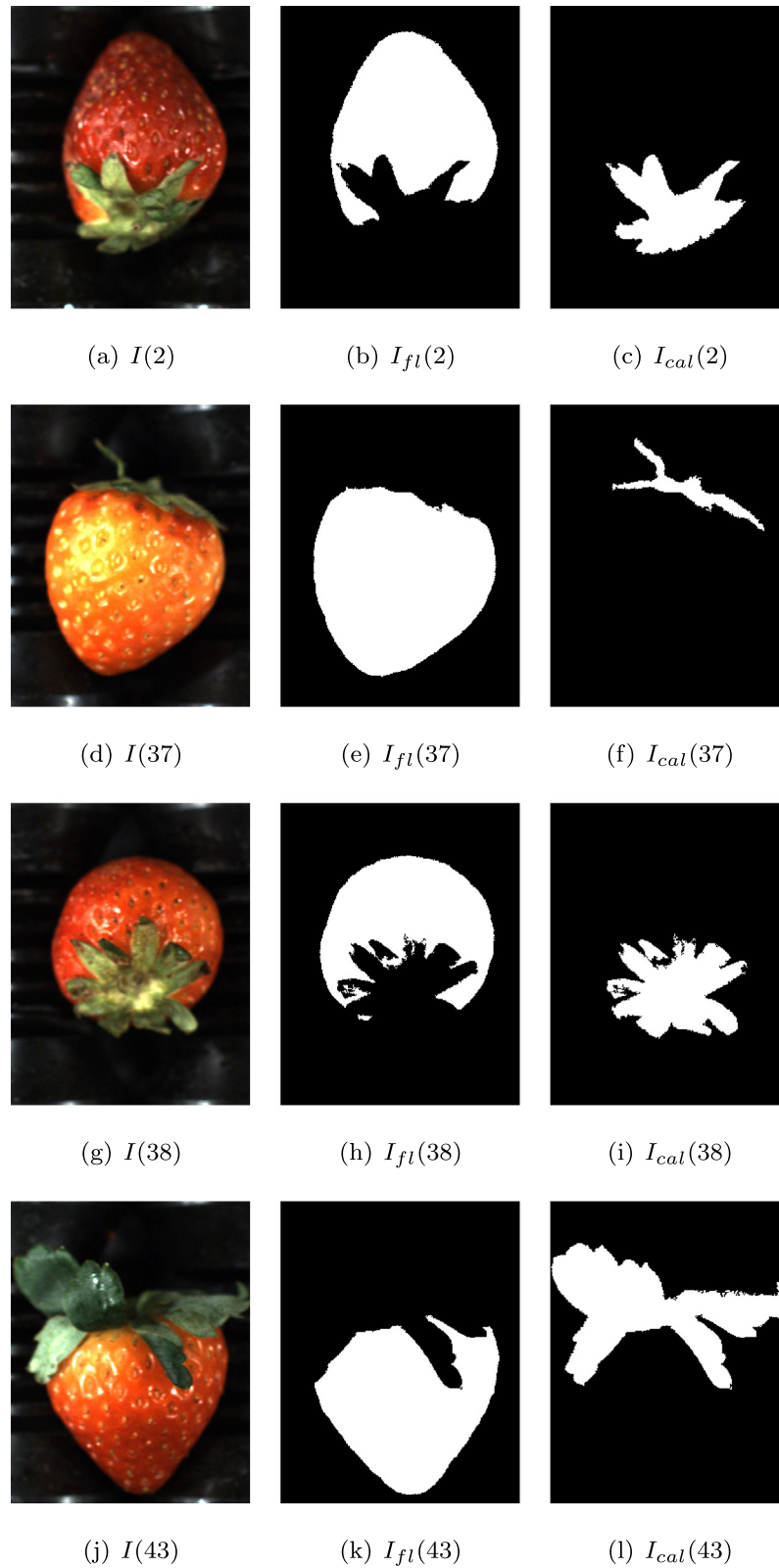


Fig. 24. Example of extraction results.

Then, appropriate values for  $T_{R-G}$  and  $T_{G-B}$  imply suitable performance in terms of extraction and also processing time.

Finally, the proposed approach was compared with the works related to post-harvest processing: Hayashi et al. (2011), Liming and Yanchao (2010), Yamamoto et al. (2012), and Ouyang et al.

(2013). To do so, the image processing pipeline of each work was implemented using Matlab. The results are presented in Table 3. Our proposed method has the best flesh extraction score and the second best flesh noise score. Concerning the calyx, our method gives significantly better results than the only existing calyx

**Table 2**  
Processing time (ms).

	Mean	Max	Min
Case #1	14.720	18.365	13.499
Case #2	15.113	17.177	13.849
Case #3	16.518	23.679	14.328
Case #4	20.199	31.311	15.553

**Table 3**  
Performances comparison (%).

	$\hat{S}(I_{fl}, \tilde{I}_{fb})$	$\hat{N}(I_{fl}, \tilde{I}_{fb})$	$\hat{S}(I_{cal}, \tilde{I}_{cb})$	$\hat{N}(I_{cal}, \tilde{I}_{cb})$
Our approach	98.1	0.4	79	0.2
Hayashi et al. (2011)	74.6	0.4	32.1	2.9
Liming and Yanchao (2010)	88	0.1	N/A	N/A
Yamamoto et al. (2012)	88.4	4.7	N/A	N/A
Ouyang et al. (2013)	94.7	0.7	N/A	N/A

extraction method proposed by Hayashi et al. (2011). The comparison results justify our decision to couple a threshold based segmentation with blob detection and selection steps to accurately extract both calyx and flesh.

#### 4. Conclusions

In this paper, an image processing algorithm was presented that extracts the calyx and flesh regions from images of strawberries under controlled lighting and background conditions that are typical during postharvest processing. The algorithm relies on a novel two dimensional color space segmentation approach followed by blobs detection and selection. The algorithm was described step by step and all user-defined parameters were presented. For experimental evaluation, 250 images were used to analyze the algorithm's sensitivity to user-defined parameters, and to evaluate its performance. As shown, the algorithm is easy to tune and performs very well despite variation in fruit orientation, color, shape, and size. When the image processing thresholds were tuned to minimize background pixels while extracting flesh and calyx performance was very promising, with 98% of flesh and 79% of calyx extracted while less than 1% of background pixels being extracted. These performance results are significantly better than the ones obtained using existing reported approaches.

#### References

- Bradski, G., 2000. openCV. Dr. Dobb's Journal of Software Tools.
- Du, Cheng-jin, Sun, Da-Wen, 2004. Recent developments in the applications of image processing techniques for food quality evaluation. *Trends Food Sci. Technol.* 15 (5), 230–249.
- Feng, Guo, Qixin, Cao, Masateru, Nagata, 2008. Fruit detachment and classification method for strawberry harvesting robot. *Int. J. Adv. Robot. Syst.* 5 (1), 4.
- Food and Agriculture Organization, 2016. Crops Data.
- Gerrans, Alfred W., Hartman, Kenneth R., 1963. Apparatus for Orienting and Halving Strawberries. US Patent 3092160 (June 4).
- Han, Kil-Su, Kim, Si-Chan, Lee, Young Bum, Kim, Sang Chul, Im, Dong Hyuk, Choi, Hong Ki, Hwang, Heon, 2012. Strawberry harvesting robot for bench-type cultivation. *J. Biosyst. Eng.* 37 (1), 65–74.
- Hayashi, Shigehiko, Takahashi, Kohei, Yamamoto, Satoshi, Saito, Sadafumi, Komeda, Takashi, 2011. Gentle handling of strawberries using a suction device. *Biosyst. Eng.* 109 (4), 348–356.
- James, J.B., Ngarmasak, T., & Rolle, R.S., 2011. Processing of Fresh-Cut Tropical Fruits and Vegetables: A Technical Guide, FAO Agricultural Service Bulletin, vol. 1, pp. 1–86.
- Leban, E., 1976. Apparatus for Decapping the Stems or Caps of Strawberries. US Patent 3952646 (Apr. 27).
- Ledebuhr, R.L., Hansen, C.M., Patterson, R.J., 1978. Fruit and Vegetable Decapping Machine. US Patent 4122766 (Oct. 31).
- Liming, Xu, Yanchao, Zhao, 2010. Automated strawberry grading system based on image processing. *Comp. Electron. Agricult.* 71, S32–S39.
- Nagata, Masateru, Bato, Pepito Menguito, Mitarai, Masafumi, Cao, Qixin, Kitahara, Takao, 2000. Study on sorting system for strawberry using machine vision (Part 1). *J. Japan. Soc. Agricult. Mach.* 62 (1), 100–110.
- Ouyang, Changqi, Li, Daoliang, Wang, Jianlun, Wang, Shuting, Han, Yu, 2013. The Research of the Strawberry Disease Identification Based on Image Processing and Pattern Recognition. Springer, Berlin, Heidelberg, pp. 69–77.
- Qingchun, F., Wengang, Z., Quan, Q., Kai, J., Rui, G., 2012. Study on strawberry robotic harvesting system. 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), vol. 1, pp. 320–324 (May).
- Saenz Edwin, Jimenez Mario, Ramirez Andres, 2013. Strawberries collecting robot prototype in greenhouse hydroponic systems. In: 2013 XVIII Symposium of IEEE Image, Signal Processing, and Artificial Vision (STSIVA), pp. 1–4.
- Suzuki, Kenji, Horiba, Isao, Sugie, Noboru, 2003. Linear-time connected-component labeling based on sequential local operations. *Comp. Vis. Image Understand.* 89 (1), 1–23.
- Vibhute, Anup, Bodhe, S.K., 2012. Applications of image processing in agriculture: a survey. *Int. J. Comp. Appl.* 52 (2).
- Wei, Xiangqin, Jia, Kun, Lan, Jinhui, Li, Yuwei, Zeng, Yiliang, Wang, Chunmei, 2014. Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. *Optik - Int. J. Light Electron Optics* 125 (19), 5684–5689.
- Yamamoto, S., Ochiai, Y., Saito, S., Hayashi, S., 2012. Study on an automatic packing system for strawberries. *Appl. Eng. Agricult.* 28 (4), 593–601.