Improving the performance of image segmentation methods through background subtraction

Luca Frigau*
University of Cagliari, Italy - frigau@unica.it

Claudio Conversano
University of Cagliari, Italy - conversa@unica.it

Francesco Mola
University of Cagliari, Italy - mola@unica.it

Abstract

Many image segmentation algorithms have been proposed to partition an image into foreground regions of interest and background regions to be ignored. These algorithms use pixel intensities to partition the image, so it should be good practice to choose an appropriate background color as different as possible from foreground one. In the case of a unique digitizing operation the user can make the choice of background color by himself in order to obtain a good result in segmentation process, but in the case of several digitizing operations it would be useful to automate the whole process by removing any decision of the user about the choice of background color. Furthermore modern instruments allow capturing images with a high resolution characterized by a massive number of pixels, and pose speed problems to image segmentation algorithms based on a local thresholding approach. In this work an approach that adapt a widely used method for detecting moving objects from a video, called background subtraction, is introduced to the image segmentation framework characterized by the specific situation in which background of the image is changeable. Respect to the standard methods, it adds new information into segmentation process. A comparison between standard methods and the approach proposed has been presented, applying both a Global and a Local thresholding method. The background subtraction approach proposed allows to improve quality of segmentation output, to automatize the process when foreground color of images is not homogeneous, and to speed it up.

Keywords: binary segmentation; computational efficiency; Otsu thresholding; Sauvola thresholding.

1. Introduction

The concept of image is quite clear to everybody, because we deal with it everyday. In math, it can be modeled by a continuous function of two variables $f(x, y)$ where (x, y) are coordinates in a plane. If image is greyscale then $f(x, y) \to [0, 1]$ is a scalar function, whereas if image is expressed in color mode its range is three or four-dimensional. The RGB is a very common color mode image, in which the value of a particular color is expressed as a vector of three elements, i.e. $f(x, y) \to (R_i, G_i, B_i)$ where $(R_i, G_i, B_i) \in [0, 1]^3$ and R, G and B represent the intensity respectively of Red, Green and Blue color channels, whereas $i = (x, y)$ indicates pixel. In order to extract information contained on images for further statistical analyses, it is necessary to transform them into inputs for methods. This operation, called image processing, is very important because all results can be strongly influenced by data input accuracy. Image segmentation is one of the most important phases concerning to that operation. Its main goal is to divide an image into parts that are strongly associated to real objects or areas contained in the image. Binary image segmentation is a specific case of image segmentation field. It is applied when image consists of contrasted objects located on a uniform background and the aim is to separate foreground from background. Although object recognition is trivial (almost always) to human vision, it is still one of the most challenging problems in image processing, image understanding, and artificial intelligence (Chan & Shen, 2005).

It is useful to introduce a distinction between images where the background is unchangeable, such as a landscape taken by a camera, and those where it is changeable, such as a scanning. From the viewpoint of image processing, their main difference is that in the first case it is not possible to change any information conveyed by the image, whereas in the second one it can be modified to accomplish some specific goals. In

fact, in order to simplify segmentation process it could be useful to choose a background with a color that allows us to create a more massive contrast between background and foreground. In spite of that, a problem for automatizing the process can appear if foreground color switches between images.
In this work we developed a process that takes advantages of background changeability for automatizing and improving the quality of image segmentation.

2. Background subtraction

Normally, all information available for image segmentation process is hold in a single image, but often this information is not enough to carry out a good segmentation output. In fact, it is possible that objects convey in them different information, which makes difficult segmentation even if background can be chosen freely. In Figure 1 it is shown an example of how an object (in this case a seed) can convey non-homogeneous information since its color on the left part is bright, whereas that in the right part is much darker. In other words on the left part the pixel intensity values are higher than those on the right part. This arises complexity of the image segmentation process, since it separates background from foreground in function to a thresholding value of pixel intensities. Consequently if, as in the example, foreground objects present in them both low and high intensity values, and background middle ones, it will be difficult to have a thresholding value able to separate background from foreground. In order to solve that problem more information is needed. New

Figure 1: Example of different information (i.e. pixel intensities) conveyed by objects

information is got adding to segmentation process an extra image with the same foreground and different background compared to original one. In literature it is not common to carry out that process using two images that differentiate themselves just for background because, due to time and memory storage issues, the image to analyze is just one. Despite that, nowadays memory storage capacity is higher day by day, whereas the problem of time analysis is relative because it depends on situation. Since we want to take advantages of difference between the two backgrounds, it is useful to enlarge that difference choosing white and black as background colors of the two images. In order to use information added by the second image, it is possible to apply an approach a lot alike to background subtraction. Background subtraction is an approach widely used for detecting moving objects from a video. It consists in subtracting each image that arranges the video to its background image, i.e. an its image with no moving objects (Piccardi, 2004). In image subtraction absolute difference between pixel intensities of first image to those of second one is performed. Non-zero differences then represent moving objects. While in background subtraction what changes in images is foreground (i.e. objects), here it is the contrary: what changes is background. Therefore if subtraction is applied before segmentation process to the two images which differ just for the background, non-zero differences will represent background instead of foreground, and vice versa for zero ones. If an image is taken twice, usually it is very hard to have the pixel intensity values of the first image identical to the correspondent ones of the second image, because some little changes in lighting often occurs. As a result likely the absolute difference between foreground pixels of the two images will assume tiny non-zero values. However it is anyway possible to distinguish between background and foreground. In fact we know that background changes from white to black, i.e. from high (close to 1) to low (close to 0) intensity values, so that their absolute difference will provide values with high intensity. Instead foreground absolute difference will provide values close to zero. This situation is a perfect starting point of image segmentation process, because the difference between background and foreground pixel intensities are now more pronounced then that considering one image only.

3. Thresholding techniques

In literature several image segmentation techniques exist, and there is not a single method that can be considered as preferable for all images. Grey level thresholding is one of the most used techniques for image segmentation. Thresholding can be interpreted as the transformation of a grey level image f to a binary image o.

$$o(x,y) = \begin{cases} 0 & \text{for } f(x,y) < T \\ 1 & \text{for } f(x,y) \geq T \end{cases} \tag{1}$$

where T is the threshold value; $o(x,y) = 1$ is for foreground pixels and $o(x,y) = 0$ is for background pixels (Sonka et al., 2014). The main critical task of this method is to select a correct threshold, which is essential for a successful segmentation. In this technique it is possible to use global or local information, and as a consequence to distinguish between Global and Local thresholding. Global thresholding consists in finding a single threshold value for whole image. Concerning Local thresholding, it is characterized by calculating a threshold value $t(x,y)$ for each pixel using information of their neighbour pixels.

Badekas & Papamarkos (2005) designed a study for evaluating seven binarization algorithms, and found the Otsu's method (Otsu, 1975) and the Sauvola's method (Sauvola & Pietikäinen, 2000) as the two best among those. The Otsu's method calculates the global thresholding T analyzing the quality of grey level distribution between the two classes indicated by the letters F (foreground) and B (background). Let us consider grey levels as discrete, n_k as the number of pixels whose grey level is k and $N = \sum n_k$. We can express the probability density distribution with the form of histogram $p_k = n_k/N$. Assume

$$P_F = \sum_{k=0}^{T} p_k \quad \text{and} \quad P_B = \sum_{T+1}^{1} p_k \tag{2}$$

respectively as the probability that a pixel belongs to foreground and background class. Now it is possible to calculate the averages of the grey level of each class and of the whole image as

$$m_F = \sum_{k=0}^{T} k \cdot p_k, \quad m_B = \sum_{T+1}^{1} k \cdot p_k \quad \text{and} \quad m = P_F m_F + P_B m_B \tag{3}$$

Hence the variances are obtained as

$$\sigma_F^2 = \sum_{k=0}^{T} (k - m_F)^2 p_k \quad \text{and} \quad \sigma_B^2 = \sum_{T+1}^{1} (k - m_B)^2 p_k \tag{4}$$

and then the variances between and within in this way

$$\sigma_{between}^2 = P_F(m_F - m)^2 + P_B(m_B - m)^2 \quad \text{and} \quad \sigma_{within}^2 = P_F \sigma_F^2 + P_B \sigma_B^2 \tag{5}$$

Finally the value of T is obtained maximazing

$$\eta(T) = \frac{\sigma_{between}^2(T)}{\sigma_{within}^2(T)} \tag{6}$$

The Sauvola's method, instead, calculates $t(x,y)$ by using the mean $m(x,y)$ and standard deviation $s(x,y)$ of intensity values included in a $W \times W$ window centered in the pixel (x,y):

$$t(x,y) = m(x,y) \left[1 + \alpha \left(\frac{s(x,y)}{Q} - 1 \right) \right] \tag{7}$$

where Q is the maximum value of the standard deviation and α is a parameter which assumes positive values in the range $[0.2, 0.5]$ and controls the value of the threshold in the local window. Badekas & Papamarkos (2005) found that $\alpha = 0.34$ gives the best results in their study. The main shortcoming of Sauvola's algorithm is its high computational complexity. In fact computing $m(x, y)$ and $s(x, y)$ produces a computational complexity of $O(W^2N^2)$ for an $N \times N$ image and a $W \times W$ window. A solution has been proposed by Shafait et al. (2008) solving directly the computational problem. They proposed of computing $m(x, y)$ and $s(x, y)$ using integral images. An integral image of an input image f is defined as the image in which the intensity at a pixel position is equal to the sum of the intensities of all the pixels above and to the left of that position in the original image, inclusive the pixel itself. Therefore the intensity at position (x, y) can be written as

$$I(x, y) = \sum_{i=1}^{x} \sum_{j=1}^{y} f(i, j) \tag{8}$$

From that is possible to compute efficiently both $m(x, y)$ and $s(x, y)$:

$$m(x, y) = (I(x + w/2, y + w/2) + I(x - w/2, y - w/2) - I(x + w/2, y - w/2) - I(x - w/2, y + w/2))/w^2$$

$$s^2(x, y) = \frac{1}{w^2} \sum_{i=x-w/2}^{x+w/2} \sum_{j=y-w/2}^{y+w/2} f^2(i, j) - m^2(x, y) \tag{9}$$

Using this approach, computational complexity decreases from $O(W^2N^2)$ to $O(N^2)$.

4. Application

In this section we want to illustrate a real comparison between the background subtraction approach and the standard methods. Let us start defining the input of the segmentation process. The first thing we need is to define the objects (i.e. the foreground). In order to arise complexity of the image segmentation process, we have decided to consider six seeds characterized by different inner pixel intensities. They have been scanned twice using a black and a white background, obtaining the top-left and the top-middle images shown in Table 1. Then the absolute difference between the pixel intensities of these two images is performed. The result of this operation is a new 'artificial' image, where foreground pixels have values close to zero and background ones close to 1 (image in the top-right part of Table 1). In order to evaluate the background subtraction approach, we decided to compare its output to those obtained applying a standard image segmentation approach, i.e. using a single image as input. In other words, the segmentation process has been run three times: once using as input the image obtained applying the background subtraction operation, another time with the black background image and the last time with the white background image. So as to compare the differences in using Local and Global thresholding approaches, the three segmentation processes have been run applying both Sauvola's method and Otsu's method.

The results are shown in Table 1 highlighting in green the edges of foreground objects identified. What stands out is that the application of background subtraction operation provides us the best results. Both Otsu and Sauvola methods produce very good segmentations, but it is simple to note the best one is achieved through a local method. Because of the output obtained using the Sauvola's method on background subtraction image (image in bottom-right part of Table 1) can be clearly considered as the best one, it is interesting to measure how much the other outputs are unlike it. To do that we calculated how many pixels have been classified in the same way. The worst results are achieved through the white background. Only the 44.65% and 59.38% of image pixels have been classified, respectively for Otsu and Sauvola methods, as the best output. The inefficiency is due to similarity between intensity of seed color and background, and to the presence of shadows that obstructs a correct definition of the thresholds. A better result is achieved using the black background. Here the percentages of pixels classified as the best output go up to 95.58% and 68.44%, respectively for Otsu and Sauvola methods. The main reason is the stronger contrast between seed color and background. In this case we can note that a global threshold method works better local one, in contrast with that observed

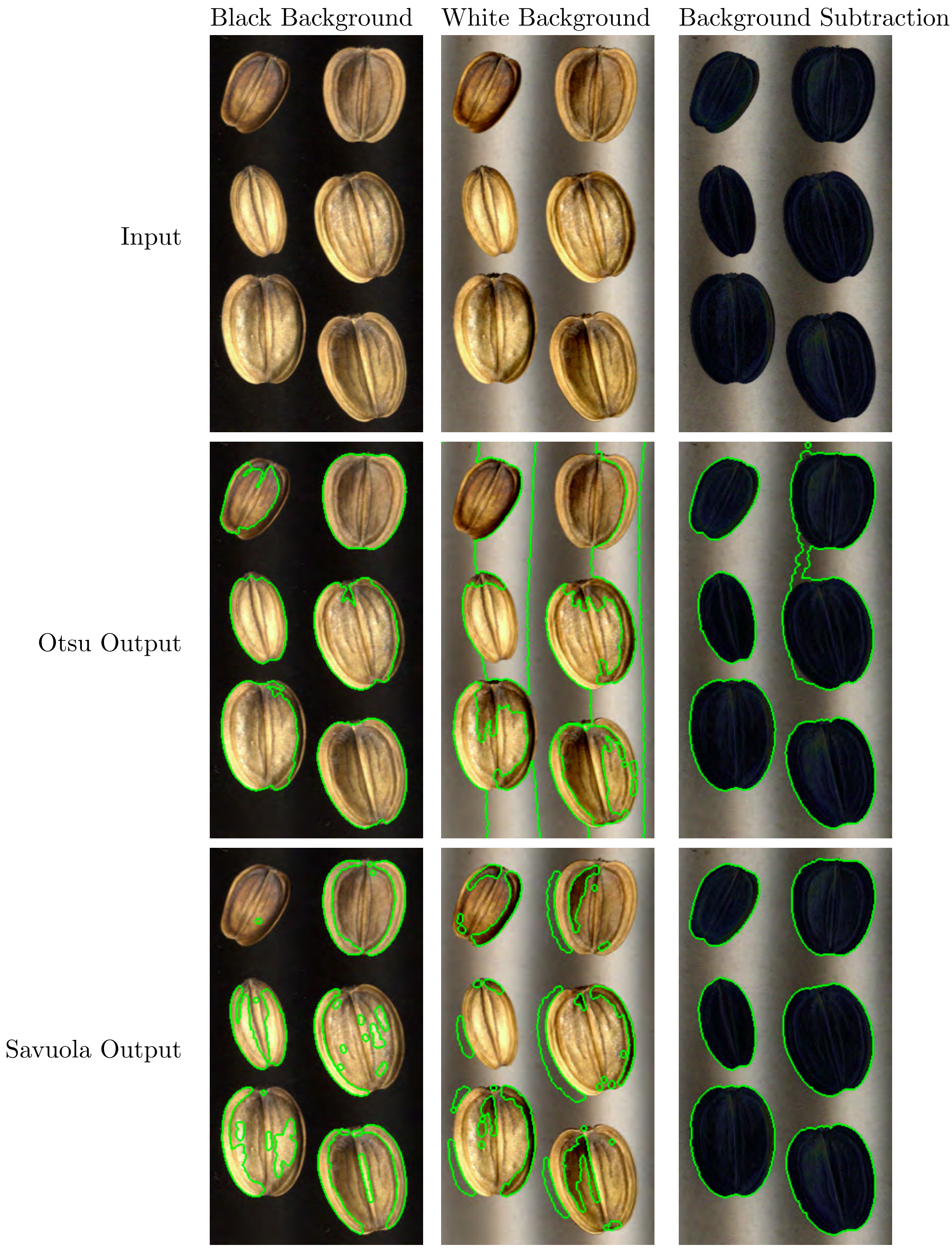

Table 1: The nine figures, arranged in a matrix 3×3, show the different combination achieved applying two different segmentation approaches to three different input images. In the first row the input images are placed, whereas in the last two the outputs achieved applying, respectively, the methods of Otsu and Sauvola. Instead, the columns concern the different input images, respectively to black background one, to white background one and to background subtraction image.

in the situation of white background. A simple reason is that global approaches tend to suffer a lot for the presence of shadows, inasmuch just one threshold value is calculated for all pixels.

5. Conclusions

The background subtraction approach provided, in the example presented above, the best results in terms of quality of segmentation for both Otsu's method and Sauvola's method. The main problems occurred with a single image as input have been the shadows and the non-homogeneous intensity values of foreground pixels. Both have been overcome by background subtraction approach.
Another important result achievable by this approach is the automizing of segmentation process. In fact the absolute difference between the pixel intensities of images with the same foreground, allows us to obtain a new 'artificial' image characterized always by tiny non-zero values in correspondence of foreground pixels, independently from original values of foreground pixels. As a result if we use background subtraction approach choosing as background colors of the two images white and black, we will expect to obtain a such good result as in the example presented above, independently from foreground pixel intensity values and their inner homogeneity.

References

Chan, T. & Shen, J. (2005). Image processing and analysis: variational, PDE, wavelet, and stochastic methods. Siam.

Piccardi, M. (2004). Background subtraction techniques: a review. In Systems, man and cybernetics, 2004 IEEE international conference.

Sonka, M., Hlavac, V., & Boyle, R. (2014). Image processing, analysis, and machine vision. Cengage Learning.

Badekas, E. & Papamarkos, N. (2005). Automatic evaluation of document binarization results. In Progress in pattern recognition, image analysis and applications. Springer.

Otsu, N. (1975). A threshold selection method from gray-level histograms. Automatica.

Sauvola, J. & Pietikäinen, M. (2000). Adaptive document image binarization. Pattern recognition.

Shafait, F., Keysers, D., & Breuel, T. M. (2008). Efficient implementation of local adaptive thresholding techniques using integral images. In Electronic Imaging 2008. International Society for Optics and Photonics.