Real-time Pill Counting using Computer Vision

Le Quang Huy
Faculty of Electrical and Electronics Engineering
Ho Chi Minh City University of Technology
Ho Chi Minh City, Viet Nam
1611260@hcmut.edu.vn

Lam Minh Quan
Faculty of Electrical and Electronics Engineering
Ho Chi Minh City University of Technology
Ho Chi Minh City, Viet Nam
1612808@hcmut.edu.vn

Dang Thanh Tin
Faculty of Electrical and Electronics Engineering
Ho Chi Minh City University of Technology
Ho Chi Minh City, Viet Nam
dttin@hcmut.edu.vn

***Abstract* — Pill counting is one of indispensable steps of distributing medicine in medical facilities. Generally, pills are counted manually and this counting method takes a lot of time for pharmacists and patients. Nowadays, thanks to its high precision and convenience, computer vision application is very popular, many applications are studied and invented incessantly and they play an important role in a host of automatic systems in a variety of fields. These systems provide contactless control which eliminates the influence of human factors, reduces errors. As a result, the measurement will give faster results with higher accuracy. In this paper, we propose a real-time computer vision program in order to count the number of pills via video captured by a removable camera connected to the computer. Our program is based on Otsu's threshold method and the image segmentation of Watershed transformation which can count pills without considering other factors such as shape and size. We believe that this is a great tool to save time and increase accuracy for counting tablets, improving the productivity of pharmacists' works and saving patients' time.**

Keywords* — *counting pills, image segmentation, watershed transformation, processing area.

I. INTRODUCTION

The most common way for pharmacists to count pills in hospitals and pharmacies is using a tough depressor to divide the amount of tablets into small group of five or two and sweep it directly into the patient's bottle until reaching the required quantities. It is very popular because of its simplicity and savings.

However, the number of prescriptions is increasing day by day, each hospital receives thousands of medical visits every day, counting the number of pills in such a rudimental way takes too much time of pharmacists for other more important works such as instructing patients about the use of drugs, examining the effects of drugs on patients. To make matters worse, counting pills in such large quantities for hours together every day can lead to the possibility of errors, although the percentage is small but on the large scale as today, the damage is also worth paying attention to.

Fortunately, science is developing quickly and ceaselessly, a host of mechanical machineries as well as softwares supporting pill counting have been studied by researchers around the world to help pharmacists reduce and limit those errors. With these automatic systems, control and measurement can reach high productivity without errors. Our proposed application is based on algorithm of "Watershed transformation" and some image processing techniques. The complete program can count the number of pills of the same medicine independently of their size and shape through a removable camera in real-time.

This paper includes 5 sections. In addition to this section, we will analyze the method as well as the advantages and disadvantages of recent works in section 2. Section 3 will talk about our works. The experimental results will be section 4. Finally, things will be wrapped up in the conclusion in section 5.

II. RELATED WORKS

Firstly, there is a method for counting tablets which depends on the uniformity of pill weight, carried out by James Roy Bradley in the patent titled "Counting scale and method of counting involving determination of submultiples by means of a series of divisors" on the 4th July, 2010 [1].

Since the pill's ingredients are very well mixed, the weight of one pill can be used as a unit weight to calculate the total amount of pills. Some scales use bar code scanners to get information from the label pack. If not, the pharmacists are asked to pour a small quantity of pills directly into the actual vial that will be delivered to the patient, usually 10 pills, to compute the unit weight. After that, pharmacists will continue pouring until the required amount is reached [1] [2]. The general structure is shown in Fig. 1.

However, this method depends too much on the weight of the pill which has some errors and can cause deviations when the number of pills is large enough. Besides, the operation of the technician can also cause errors if they don't stop promptly when the number of pills is sufficient [2].

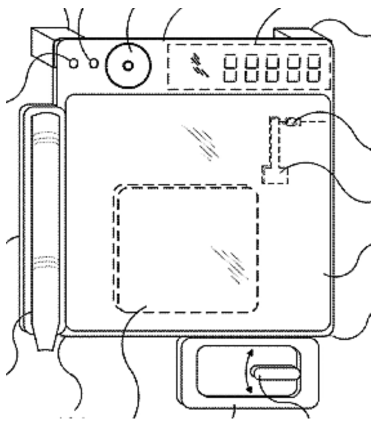


Fig. 1. General structure of a pill-counting scale [1].

On top of that, pills can be counted and dispensed by a system which has a myriad of microprocessors to identify the number of pills going through the system. There are a lot of studies about this method, one of the first ones is proclaimed in the patent "Method for counting and dispensing tables, capsules and pills" of Joseph H. Boyer and James P. Boyer, published on the 6th May, 1996 [3].

The pills are poured into the machine from a funnel and counted when passing through the sensor system installed on the way to the collection trough. The tablets will fall directly into the patient's vial at the end of the path. The collection chute usually has a diverting mechanism which activates when the correct count is achieved and sends the excess pills into a separate bin for return to the supply bottle [2] [3]. The general structure is shown in Fig. 2.

The method does not seem to be flawed, it even really works when the number of pills increases. However, the hardware structure is quite complicated and cumbersome, leading to high cost and causing difficulties in cleaning [2].

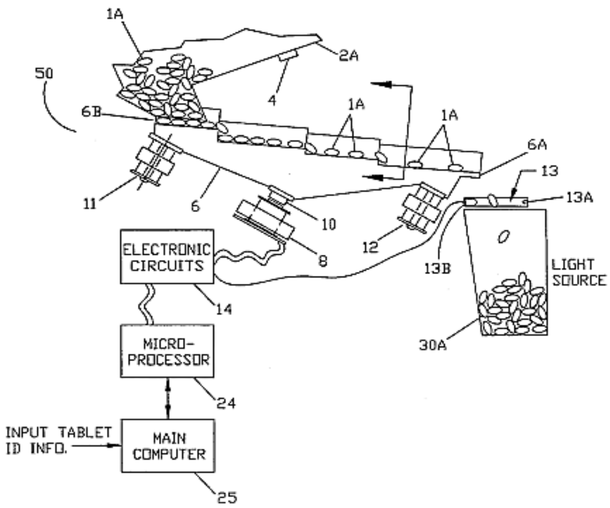


Fig. 2. General structure of pill counter using microprocessors [3].

And the latest method which applying image processing of the computer vision to calculate the number of pills is studied by Rodney N. Hamilton, Gardner, KS in the patent named “Pharmacy pill counting vision”, released on the 7th April, 2003 [4].

The tablets will be spread evenly on the tray. Their real-time images will be recorded by a removable camera and then transferred to the computer. The computer will use some image processing algorithms, segment tablets by positions and count the number of tablets. The tray can adjust the background color to increase the contrast between the tray and the pills to support image processing. The general structures are shown in Fig. 3 and Fig. 4.

However, the algorithm used to process images in this study is "Hough Transform" which can only handle and identify popular shapes which can easily be simulated by equations such as circle or ellipse [4].

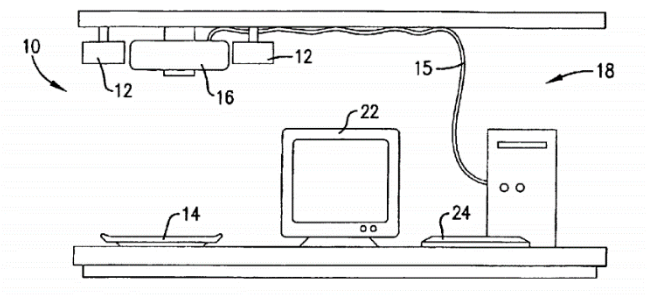


Fig. 3. A Semi-automated pill counting system [4].

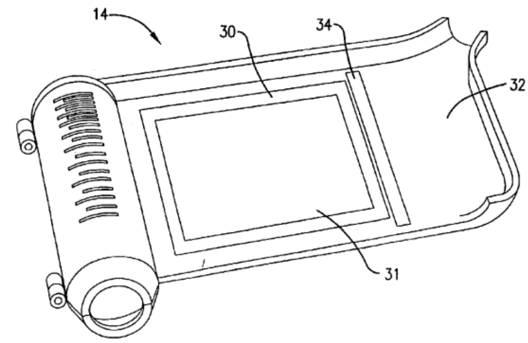


Fig. 4. A Modified pharmacist’s tray [4].

After consulting and analyzing of available studies, we believe the method as in [4] is the method that we should follow in that it ensures both elements of structural simplicity and high accuracy. In this paper, we propose a program to count the number of pills of the same medicine using a removable camera which will inherit the strengths as well as overcome the limitations as in [4].

III. OUR WORKS

In this section, we will present and analyze the method that we use to build the program. As seen on the diagram in Fig. 5 which includes 3 main steps: "Image pre-processing", "Watershed Transform" and "Processing area".

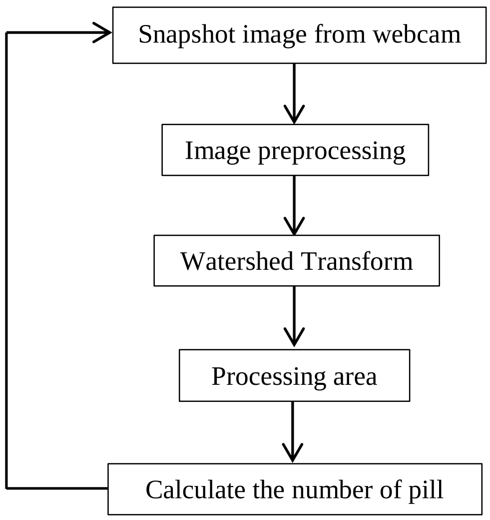


Fig. 5. Algorithm flow chart.

A. Image preprocessing

In this program, we will work with digital images. The captured image of the camera is stored as a multi-dimensional array of data mxnx3 which includes 3 matrices with 3 color bands that defines the red (R), green (G) and blue (B) color values stored at pixels’ location in the color plane [5]. In this form, we cannot apply image processing tools yet, but we must convert it into a binary image.

First of all, we will convert color image as in Fig. 7(a) into grey image as in Fig. 7(b). The primary color components used in the video and television industries incorporate a correction to compensate for the nonlinearity of the video monitors. According to ITU-R BT.601

standard, the average luminance of the reconstruction color space of a television monitor is computed [6]:

$$Y = 0.299R + 0.587G + 0.114B \qquad (1)$$

Now we will only work with 1 matrix (a x b). Things have become simpler but it isn’t the result we expect yet. From the grey image above, we will convert it to binary image with a threshold given by Otsu's threshold method.

Otsu's threshold method is an adaptive way for binarization algorithms in image processing. Firstly, the statistical data of an image is used to build a histogram to describe the distribution of grey scale in pixels as in Fig. 6.

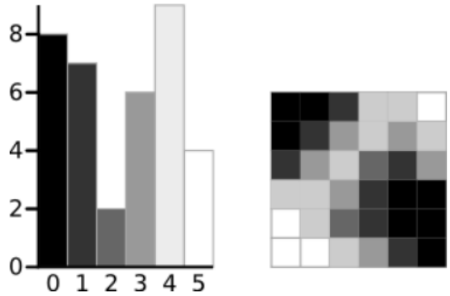
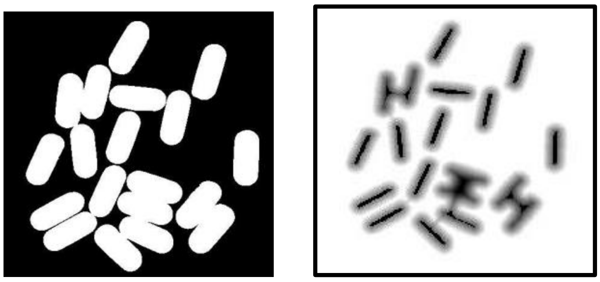


Fig. 6. A 6-level grey scale image and its histogram [12].

Subsequently, each grey scale level is used as a threshold value to distinguish the foreground and background. We will compare each grey intensity value according to the coordinates of the grey image matrix with the threshold value. If it is greater than the threshold, it will receive a value of 1 as the foreground; otherwise it will be 0 as the background [7]. The next step is to calculate the “Within-Class Variance” of every threshold value and find the threshold value where the “Within-Class Variance” of both foreground and background are at their minimum to make sure the homogeneity of each region. A faster approach is that we can calculate what is called the “Between Class Variance” and try to find the threshold value where the "Between Class Variance" is at its maximum to ensure the distinction between foreground and background [8].

Now we have a binary image as in Fig. 7(c), this is an important matrix for further processing steps.

(a) (b)



(c) (d)

Fig. 7. Results when apply image preprocessing algorithm to (a) RGB image which gives (b) a grey image, (c) a binary image and (d) a image that presents distances from pixels to the nearest nonzero-valued pixel.

B. *Watershed transform*

Watershed transformation is a computer analysis of objects in digital image which helps to define objects in a picture. Computer analysis of objects starts by finding which pixel belongs to the object. This is called image segmentation, the process of separating objects from the background as well as from each other [9].

From the binary image as in Fig. 7(c), we will convert it into “distance transformation of binary image” (Fig. 7(d)) by using another image processing called “Distance transform” to compute the distance from every pixel in the region of objects to the nearest zero-valued pixel. The farther the distance is, which means the closer the pixel is to the center of the object, the darker it will be and vice versa.

Any “distance transformation of binary image” can be considered as a terrain surface. Dark areas with homogeneous grey level are considered low areas (catchment basins) and bright areas are considered high areas (watershed lines). If we flood this surface from its minima and, if we prevent the merging of the waters coming from different sources, we can divide the image into two different sets: the catchment basins and the watershed lines which are the objects that we need to define and their boundaries [10].

This algorithm is a great tool to count pills that have multitudinous shapes. We process on similar regions that have approximately the same area without bothering their shape. It helps to overcome the limitations of "Hough Transform" [4] which counts pills by detecting objects described with its model which is similar to the shape simulated earlier, and so it can only handle analytically defined shapes such as circle or ellipse.

Nevertheless, there is a common circumstance in watershed segmentation which is called oversegmentation. Generally, there are some noise spots in “distance transformation" image because of the uneven distribution of light on the surface of the object due to smoothness or the structure of its surface. Besides, this phenomenon is immensely common at the asymmetrical object since the distance between the boundary pixels and the center pixels are quite different, leading to the dark area (low area) in “distance transformation" image not gathering in the middle of objects. As result, there are more than one local minimum in the object's region. Each local minimum, even if it is inconsiderable, forms a catchment basin and then a watershed region around them [11]. Subsequently, there are a host of watershed regions in a object's region as shown in Fig. 8.

One solution here is modifying the image to filter out tiny local minima or remove minima that are too shallow [9]. This is called "minima imposition" and the result is shown is Fig. 9.

Fig. 8. Over segmentation in Watershed transformation.

Fig. 9. Binary image after overcoming over segmentation.

As we can see, if we simply use "Watershed Transform", you cannot divide some objects that are too close to each other in that the pixel at the boundaries are too small to be define as a local minima after we use some filter to reduce over segmentation. In this paper, we propose a solution to deal with the above case. And we call it “Handling area”.

C. Handling area

After the Watershed step, sometimes, the objects still have a few parts that are not fully divided and still "stick together". So we need to do one more step so we can count exactly how many pills are displayed on the screen.

First of all, we will extract the area value of the regions identified as pills. And then, we selected a unit area value as standard. Because the pills are located in different locations around the camera with different angles, the regions will have different area values despite the fact that they are pills of the same medicine. Therefore, to minimize the error of division in the following part of the process, the average area value of regions that considered as a pill will be chosen as the unit area. After that, the area of each object will be divided by unit area, then rounded to the nearest integer. If the difference between the two values, before and after being rounded, is less than a nominal value, then receive that integer. Otherwise we will show a "Warning" text box to warn users to adjust the positions of pills as well as to check whether there are some pills of other medicine mixed in there. Finally, we add those integers together to get the number of pills.

IV. EXPERIMENTAL RESULTS

In order to test the accuracy of the software, we carried out an experiment with three different distances (from the camera to the tray below) which are 20 (cm), 25 (cm) and 30 (cm) respectively. Experimental results are presented in the Table I below.

A. Data

TABLE I. PERCENTAGE OF ERRORS

Reality	**Distance between camera and tray (cm)**					
	h = 20		***h = 25***		***h = 30***	
	Result	***Error (%)***	***Result***	***Error (%)***	***Result***	***Error (%)***
20	20	0.00	20	0.00	20	0.00
40	40	0.00	40	0.00	40	0.00
60	60	0.00	60	0.00	60	0.00
80	80	0.00	80	0.00	80	0.00
100	100	0.00	100	0.00	100	0.00
105	105	0.00	103	1.90	105	0.00
110	110	0.00	107	2.73	106	3.64
115	115	0.00	115	0.00	110	4.35
120	120	0.00	120	0.00	120	0.00
125	125	0.00	125	0.00	120	4.00
130	130	0.00	124	4.62	126	3.08

B. Review

As we can see, at all three distances, if the number of pills is smaller than 100 tablets, the software always gives the accurate results. From approximately 120 tablets or more, the system begins to have some errors which are between 1% and 5% at 25 centimeters and 30 centimeters. It is because that the frame size is fixed, so the larger the number of pills, the smaller the distance between them and maybe some pills get laid out of the frame, affecting identification. At the same time, the larger the distance between the camera and the tray, the smaller the boundary’s pixels between the pills and the easier it is to be ignored by the filter of "minima imposition".

However, the number of pills per count as well as the number of pills per prescription usually does not exceed 100 tablets, so this error does not affect too much. If there is a need to count over 100 tablets, we recommend that users adjust the camera so that the distance from it to the tray is 20 centimeters or less to ensure high accuracy. We also advise users to use tray which have color that is highly contrast to the color of pills to support image processing.

When compared with the current studies as in [4], our proposed approach gives fast results with high precision and high reliability. On top of that, our program can count independently of size and shape of pills, without knowing pill’s data first as in [4].

C. Improvement

Since our program is developed based on Matlab language platform which required many supporting libraries, it is quite heavy and its booting-time is relatively long at the moment, we can improve by modifying the commands, limiting the use of available commands with heavy libraries or developing our algorithm in other language platforms such as Python for programing and packaging the product to get the product as optimal as possible.

Moreover, we also can design a tray with a LCD surface which can change color flexibly to ensure the contrast with the pills' color.

The last thing for advanced development is that we can use camera with higher resolution to help image processing algorithm identify pixels at the boundaries more clearly.

V. Conclusion

In this day and age, computer vision applications are used in most of production lines. With these automatic systems, control and measurement can reach high productivity without errors. In this article, we present a real-time computer vision program in order to count the number of pills via video captured by a removable camera connected to the computer. The main algorithm which we used was "Watershed Algorithm". The images were taken in real time from a camera and processed with the computer vision's algorithms suggested in the study. Our complete program overcomed some limitations of previous researches and when compared with the current studies, it has its own advantages. We also found out several ways to improve our program which were already mentioned in the previous part. We believe that it will be a useful tool to support pharmacists' works, improve the overload status of medical treatment in large medical facilities.

References

[1] James Roy Bradley, "Counting scale and method of counting involving determination of submultiples by means of a series of divisors", U. S. Patent 8 530 763, Jun. 4, 2010.

[2] "Pill Counting Technique", Torbal Company, Apr. 4, 2011. [Online]. Available: http://www.pillcounting.com/latest-articles/pill-counting-techniques?fbclid=IwAR24jO_f8zmoaIQcY3h1nILX9uSY9yKb3nb-00JpqEpSsHVl7kGqaVoZNew. [Accessed: May 17, 2019].

[3] Joseph H. Boyer and James P. Boyer, "Method for counting and dispensing tablets, capsules, and pills", U. S. Patent 5 671 262, May 6, 1996.

[4] Rodney N. Hamilton, "Pharmacy pill counting vision system", U. S. Patent 6 738 723, Apr. 7, 2003.

[5] "RGB image representation", Geeks for Geeks.[Online]. Available: https://www.geeksforgeeks.org/matlab-rgb-image-representation/. [Accessed: May 17, 2019].

[6] Jonas Gomes, Luiz Velho, "Color Systems", in *Image Processing for Computer Graphics*, 1st Ed. New York, USA: Springer Science & Business Media, 1997, pp.94.

[7] "Converting a Greyscale Image to Binary Image using Threholding", Geeks for Geeks. [Online]. Available: https://www.geeksforgeeks.org/matlab-converting-a-greyscale-image-to-binary-image-using-thresholding/. [Accessed: May 17, 2019].

[8] Andrew Greensted, "Otsu Thresholding", Jun. 17, 2010. [Online]. Available: http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html. [Accessed: May 17, 2019].

[9] Steve Eddins, "The Watershed Transform: Strategies for image segmentation", Mathworks, 2002. [Online]. Available: https://www.mathworks.com/company/newsletters/articles/the-watershed-transform-strategies-for-image-segmentation.html. [Accessed: May 17, 2019].

[10] Serge Beucher, "Image segmentation and mathematical morphology", Centre de Morphologie Mathématique, May. 18, 2010. [Online]. Available: http://www.cmm.mines-paristech.fr/~beucher/wtshed.html?fbclid=IwAR3vxGvCYy-wEU87Acv4qY5UwIRVBRZ1o7qln0MHaVv-HQ_mnoOfg-6sX-8. [Accessed: May 17, 2019].

[11] Steve Eddins, "Watershed Transform Question From Tech Support", Nov. 19, 2013. [Online]. Available: https://blogs.mathworks.com/steve/2013/11/19/watershed-transform-question-from-tech-support/. [Accessed: May 17, 2019].