COMPARATIVE ANALYSIS ON EXISTING BLOB COUNTING METHODS

Jade Kenneth S. Darunday
DCST
Visayas State University
Baybay City, Leyte
+639751375707
jadekennethdarunday@gmail.com

Kim S. Escabarte
DCST
Visayas State University
Baybay City, Leyte
+639275327177
kimescabarte@gmail.com

John Christopher A. Go
DCST
Visayas State University
Baybay City, Leyte
+639206666733
stephengabrielgo@gmail.com

Mikhail M. Lumanta
DCST
Visayas State University
Baybay City, Leyte
+639561835063
Lumanta.mikhail21299@gmail.com

ABSTRACT

This study reviews about counting of objects in an image using the selected image processing techniques also known as blob counting. Counting is one of the most common action performed by people. The objective of this study is to be able to gather information on how blob counting can make counting objects easier using some selected image processing techniques. These image processing techniques are: grayscale which is used in removing unnecessary color information aside from gray in an image, threshold which is used in separating the colors between black and white using its threshold value, dilation which is used to split the black objects from white objects in a threshold image and erosion which is used in removing noises from a threshold image.
There are also three methods of blob counting used in this study. These methods are Grassfire, Horizontal Layered Scanning and Blob Counting Using Template Matching.

KEYWORDS

Blob counting, C#, Morphological Operations, Median Filtering, Thresholding, Horizontal Layered Scanning, Template Matching

1. INTRODUCTION

An image is a visual representation of something, it may contain shapes, color that conveys its information. Counting involves approximating the number or getting the total units in a group. Counting is a basic skill learned from childhood, the common or basic counting known as the manual counting. It is done daily such as counting numbers in mathematics and counting money. Manual counting is suitable only in small amounts but how about counting large-scale quantities such as counting stars or rice grains, which is a huge inconvenience, impractical, and infeasible for us humans. It would take a long time to finish and will even be inaccurate and imprecise. Now, counting the impossible is possible with the help of technology (digital camera) and digital image processing. Technology's rapid improvement reach to its peak where computer had its own vision through some methods called pattern recognition in which developing this kind of application can be possible. This kind of counting application is essential in fields or studies such as biological sciences that involves blood cell counting and detection, where cell counting is very important and useful for medical diagnosis and biological research [4], and in agriculture [3] – grain counting. In this research, image processing techniques are applied to make counting blobs easier, faster, more accurate and convenient. The methods used are based on existing algorithms [1] [2].

2. RELATED WORKS

Blob counting is widely used in many fields such as agriculture and biology [3] [4]. Its function basically is to detect and count blobs. Examples are detecting and counting grains, and cell counting.

Commonly used image processing techniques used in blob counting include grayscale, threshold, and morphology. Grayscale is used since the color information of the image is unneeded. Threshold is applied to attain a binary image where the background and foreground is easily distinguishable. The blobs which are overlapping are separated through the use of morphological operations. These morphological operations are erosion, where noises are removed and the outline of any clustered pixels are thinned and; dilation, wherein the outline of any pixel clusters are thickened. Both go hand in hand to reduce image noises, improve object detection, enhance object contours and etc.

Gohil Asmitaba and Dhaval Pipalia presented an algorithm to detect and count small objects using image processing in MATLAB [1]. In their algorithm, an image is taken, turned into grayscale, then into a binary image using threshold. The overlapping objects are then separated through dilation. The detection and labeling follows after. Their algorithm accurately counts objects. Smaller objects will yield less accurate results. The bigger the objects, the more the accurate results the algorithm will produce.

Devandra Dattatray Bhosale proposed a method to use digital image processing for grain counting in MATLAB [2]. In the algorithm, the grains to be counted is captured using a digital camera. It is then converted to grayscale, median filtering is applied to remove noise, then transformed into a binary image by using threshold. The grains overlapping are identified and separated by erosion. The grains are counted and labeled at the end. This algorithm proved to be effective. Similar to the previous algorithm mentioned, its accuracy increases when the grains are bigger. Accuracy can be further increased by separating the grains first and taking the picture of the grains with a black background.

In both algorithms, blob detection accuracy is increased when the blobs are bigger, the background and foreground are easily distinguishable and the blobs are separated from each other.

Anne Kaspers published a paper that describes several blob detection methods [5]. It shows that these blob detection methods each have their strengths and weaknesses. The methods she used are Template Matching [7] where a template of a blob is moved over the search image and blobs are detected where the template matches a part of the image. This algorithm is very fast and high in accuracy according to Kaspers; Watershed Detection [8] which assumes an image to be grey value mountains and simulates the process of rain falling onto the mountains, running down the mountain range and accumulating in basins. According to Kaspers, this method is fast and high in accuracy; Spoke Filter [9] which is an early single scale blob detector. This method is fast and high in accuracy; Automatic scale detection [10], its principle: in the absence of other evidence, assume that a scale level, at which some combination of normalized derivatives assumes a local maximum over scales, reflects the size of the corresponding blob. Its accuracy is reasonable and its speed is quite slow; Sub-pixel precise blob detection [11] which defines blobs as a small, compact image primitive, defined as a rectangle with constant contrast. The algorithm works by extracting the possible center points of blobs, reconstructing the boundary around a point, and finally the extraction of blob attributes. According to Anne Kasper's paper, this algorithm is quite slow but is very high in accuracy; Effective maxima line detector [12], a method where connected curves of modulus maxima at different scales -called maxima lines - are effectively selected, to divide blobs from noise. This process is quite slow and has reasonable accuracy. And; Confidence detector [13] which is a complex multi-scale method to extract blobs from an image. It is reasonable in accuracy and is quite fast.

3. METHODOLOGY

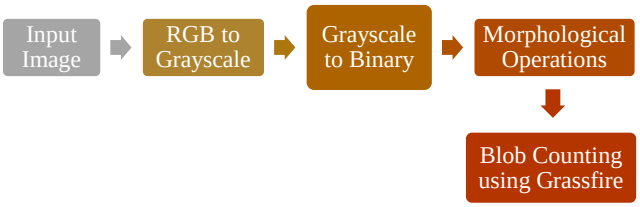

Figure 1. Block Diagram of Blob Counting using Grassfire Transform

3.1 Method 1.Grassfire Transform

Initially, an image of the objects to be counted are spread and placed on a dark background is taken. Next, the original image is transformed into grayscale. It is then converted into a binary image which easily distinguishes the background and foreground, black and white respectively. This is achieved by applying the image processing technique threshold to gray scaled image. After applying threshold, the noise around the objects is removed and separate the connected blobs. Doing this lessens the counting error. Lastly, the detection, counting and labeling is done. The counting algorithm used is the Grassfire Transform Algorithm [6]. In image processing, the grassfire transform is the computation of the distance from a pixel to the border of a region. It can be described as "setting fire" to the borders of an image region to yield descriptors such as the region's skeleton or medial axis. Harry Blum introduced the concept in 1967. The algorithm starts in the upper-left corner of the binary image. It then scans the entire image from left to right and from top to bottom. Method 1 is shown below.

Figure 2. Original Image

Figure 2. Grayscale Image

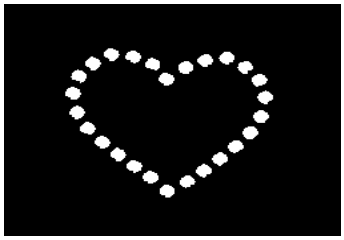

Figure 2. Threshold Image

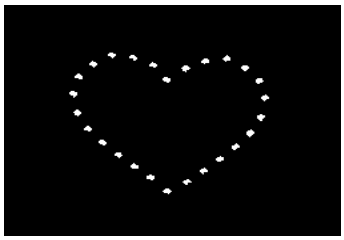

Figure 2. Dilated Image

Figure 2. Counted and Labeled Image

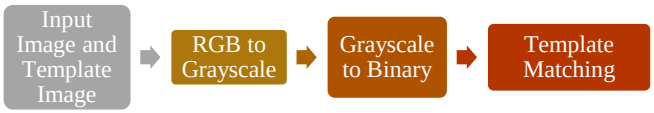

Figure 3. Block Diagram of Blob Counting using Template Matching

3.3 Method 2.Template Matching

In this method two images are inputted, the image to be searched and the template image. The template image is cropped from the original image. Both images are converted to grayscale then binary. Morphological operations are used when deemed necessary. The template traverses the image and finds matches. A threshold value for the likeness of both images is set to 75% to make detection easier. Method 2 is shown in the following page.

Figure 4. Original Image

Figure 4. Template Image

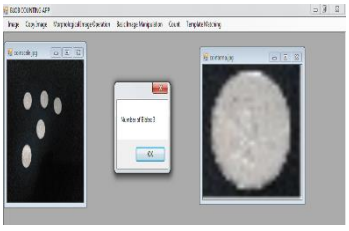

Figure 4. Processed Result

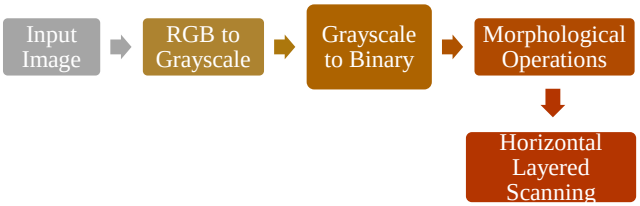

Figure 5. Block Diagram of Blob Counting using Horizontal Layered Scanning

3.4 Method 3. Horizontal Layered Scanning

Initially, an image of the objects to be counted are spread. The next step is applying the median filtering to the image. Median filtering is a nonlinear digital filtering technique, often used to remove noise from an image or signal. Such noise reduction is a typical pre-processing step to improve the results of later processing. Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise. Next, the original image is transformed into a binary image which easily distinguishes the background and foreground, black and white respectively. This is achieved by applying the Otsu-threshold to the original image. After applying Otsu-threshold, the noise around the objects is removed and separate the connected blobs. This is also achieved by applying the Morphological Operation Dilation and Erosion. Doing this lessens the counting error. Lastly, the detection, counting and labeling is done. For counting the Blobs, another algorithm used which is the Horizontal Layered Scanning. This algorithm is only applicable on whole blobs in an image. Example of this method is shown below.

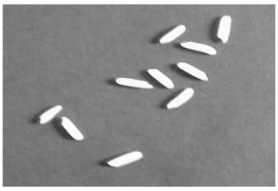

Figure 6. Original Image

Figure 6. Median Filtered

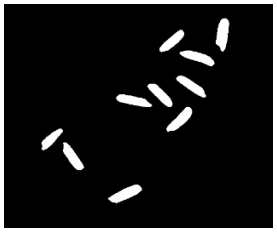

Figure 6. Threshold Image

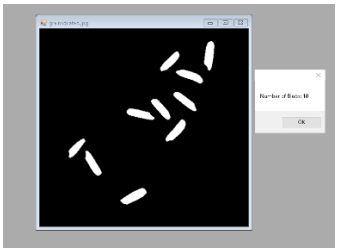

Figure 6. Processed Image

4. EXPERIMENTAL RESULTS

In this part, different images are used, as shown in Fig.7, 8, 9 and 10. These are used to compare the accuracy of each methods. The images in the figure shown below are processed and the results are compared. Method 1 and 3 are placed on left and right respectively. The template for Method 2 is also provided.

Figure 7.Pearls Shaped into a Heart

Figure 7. Template Image

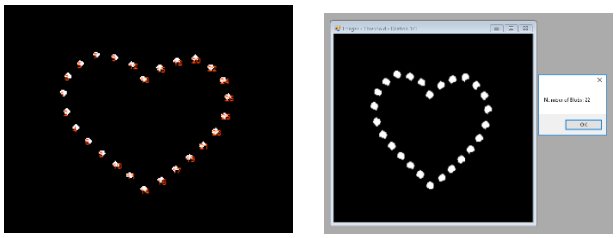

Figure 7. Processed Images

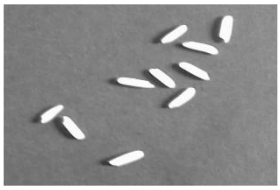

Figure 8. Rice Grains

Figure 8. Template Image

Figure 8. Processed Images

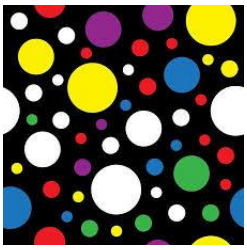

Figure 9.Colored Circles

Figure 9. Template Image

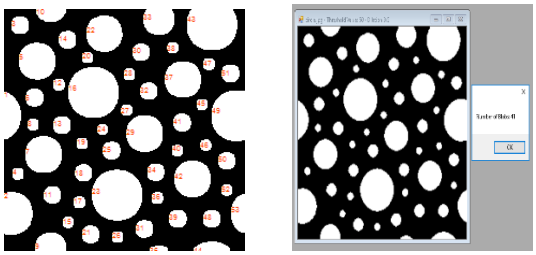

Figure 9. Processed Images

Figure 10. White Blood Cells

Figure 10. Template Image

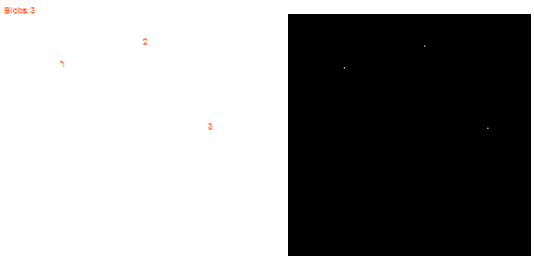

Figure 10. Processed Images

4.1 Counting Evaluation

The experimental result and percent error are given in Table 1. Here the different images in the fig. 5, 6, 7, and 8 are processed by the two methods. The Actual Value and the Output Value of the images are compared.

Table 1. Results

Figure	Actual Value	Method 1 Value	Method 2 Value	Method 3 Value
5	26	26	26	23
6	10	10	1	10
7	53	53	31	41
8	3	3	3	1

The accuracy of the two methods are tested using the percent error formula:

$$\text{Percent Error} = \left| \frac{Observed\ Value - Actual\ Value}{Actual\ Value} \right| x\ 100$$

Table 2.Percent Error

Figure	Method 1 % Error	Method 2 % Error	Method 3 % Error
5	0 %	0%	11.54%
6	0 %	90%	0%
7	0 %	41.51%	22.64%
8	0 %	0%	66.67%

5. CONCLUSION

Using the selected different methods in image processing is an effective method in implementing the counting of objects in an image. Although accuracy may not be entirely perfect, the three selected methods are still better in terms of efficiency and accuracy than manual blob counting. The three methods require the image to have separated objects and preferably no overlaps to be accurate. The foreground and background must be also easily distinguishable. The bigger the blobs, the better the accuracy. Grassfire transform proved to be better among the other methods.

6. REFERENCES

[1] G. Asmitaba and D. Pipalia, "Design an Algorithm to Detect and Count Small Size Object using Digital Image Processing," 2016.

[2] D. D. Bhosale, "Use of Digital Image Processing for Grain Counting," 2015.

[3] Grains Automatic Counting Method Based on Computer Version Wen-cheng Wang,Yu-mei Wang, Tao Ji. International Journal of Advancements in Computing Technology (IJACT) Volume4, Number13, July2012.

[4] Xiaomin Guo and Feihong Yu, "A Method of Automatic Cell Counting Based on Microscopic Image," 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, Vol. 1, Aug. 2013. pp. 293-296.

[5] Kasper, Anne, "Blob Detection", https://dspace.library.uu.nl/bitstream/handle/1874/204781/Blob%20detection%20Final%20version.pdf

[6] Blum, Harry. "A Transformation for extracting new descriptors of shape", 1967, "http://pageperso.lif.univ-mrs.fr/~edouard.thiel/rech/1967-blum.pdf", 6/8/2012

[7] M. Sonka, V. Hlavac, R. Boyle. Image Processing, Analysing, and Machine Vision. Thomson, 2008.

[8] S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. In International workshop on image processing, real-time edge and motion detection, 1979.

[9] L. G. Minor and J. Sklansky. Detection and segmentation of blobs in infrared images. IEEE Tranactions on Systems Man and Cyberneteics, SMC-11(3):194–201, March 1981.

[10] T. Lindeberg. Feature detection with automatic scale selection. International Journal of Computer Vision, 30(2):79-16, 1998.

[11] S. Hinz, Fast and Subpixel Precise Blob Detection and Attribution. In: Proceedings ICIP'05, Genua. pp. 457–460, 2005.

[12] C. Damerval and S. Meignen. Blob detection with wavelet maxima lines. IEEE Signal Processing Letters, 14(1):39–42, 2007.

[13] P.-E. Forssén and G. Granlund. Robust multiscale extraction of blob features. In Proceedings of the 13th Scandinavian Conference on Image Analysis, LNCS 2749, 11–18, 2003.