A Review on Methods for Automatic Counting of Objects in Digital Images

Article *in* IEEE Latin America Transactions · September 2012

DOI: 10.1109/TLA.2012.6362356

Jayme Garcia Arnal Barbedo
Brazilian Agricultural Research Corporation (EMBRAPA)

A Review on Methods for Automatic Counting of Objects in Digital Images

J. G. A. Barbedo

***Abstract*— A growing number of routine and research activities, in a wide variety of fields, have the counting of certain types of objects (cells, people, insects, etc.) as one of their main components. In most cases, such counting procedure is performed manually, in a process that is often lengthy and tedious. For that reason, several methods for automatically counting the objects of interest have been proposed in the last two decades. The vast majority of those methods rely on digital images containing the objects to provide an estimate as close as possible to the results manually obtained by human experts. The review is organized in a tutorial-like form, that is, instead of grouping the references according to a given criterion and then describing them, the paper describes some of the main tools and techniques used in this field of research, and then cites the references as sources for additional information and inspiration.**

***Keywords*— automatic object counting, digital images.**

I. Introduction

ONE of the main applications of digital image processing is the estimation of the number of certain types of objects in an image. The variety of object counting problems is considerable, each having their own characteristics and challenges. The difficulty of the problem depends on many factors: contrast between object and background, degree of object clustering, object texture and its variation, object size and its variation, complexity of the objects, among others.

Despite the differences between object counting problems, most of them have at least some characteristics in common. As a result, strategies developed to solve a certain problem may, potentially, incorporate solutions that would be useful in other cases. On the other hand, such a diversity means that the proposals are published in a wide variety of periodicals and conferences. Since it is unlikely that someone refers to publications that are not directly related to their area of research, many potential solutions may be ignored.

Normally, literature review papers present the references divided into categories which are usually selected according to their technical characteristics. This approach works well in most cases, but it was not adopted here for a number of reasons. There are some criteria that can be used to classify methods for automatic counting of objects, such as the type of segmentation or the type of pattern recognition adopted. The problem with those criteria is that they describe only part of the method, which is evidenced by the fact that most methods are, basically, composed by a chain of tools and techniques of same importance. As a result, any kind of classification will be somewhat limiting and artificial.

Because of that, a more context appropriate approach was adopted. As a result, the organization of the text is more tutorial-like, in which the tools and techniques are the focal point of the text, instead the references themselves, which are presented as additional sources of information and inspiration. It is important to emphasize that all references and all information that would be included if a more conventional approach had been adopted are still present. It is also important to notice that, due to length limitations, it would be unpractical to include all tools and techniques in existence, thus a selection process was applied in order to include only those more relevant in this context. In order to better organize the text, the techniques are divided into eight general categories.

II. Bibliographical Review

As commented before, the techniques presented in this section are divided into eight categories. Each category is briefly described, with more detail being dedicated to the techniques themselves.

A. Morphological Operations

Morphology is a large set of image processing operations based on form [1]. The morphological operations apply a structuring element to an input image, creating, in its output, an image with the same dimensions. In a morphological operation, the value of each pixel in the output image is based on the comparison between the corresponding pixel and its neighbors in the input image [1]. Morphological operations are usually applied to binary images, although it is possible to apply them to grayscale images as well.

The two most basic morphological operations are *dilation* and *erosion*. Other morphological operations derive from those two, such as *opening*, *closing* and *skeletonization*, as will be seen in the following.

1) Dilation: the morphological operation *dilation* expands the edges of the object in the image by adding pixels according to a structuring element (usually a square or a disk with a certain radius). This kind of operation is usually applied to eliminate holes in the interior of objects, and also to fuse elements that should not be separated.

Barber et al. [2] use dilation to guarantee that the pixels in the edges detected for each object are 4-connected (see Section 2.G.4). A similar objective motivates its use in the work by Poomkrakand and Neatpisarnvanit [3]. Cosío et al

J. G. A. Barbedo, Embrapa Agricultural Informatics (CNPTIA), Brazilian Agricultural Research Corporation (Embrapa). Campinas, São Paulo, Brazil, jayme.barbedo@embrapa.br.

[4] apply this operation to eliminate spurious points resulting from the application of a watershed algorithm (see Section 2.G.2). Hamghalam and Ayatollahi [5] use dilation to merge structures present in the interior of cells. Other uses for dilation can be found in [6,7,8].

2) Erosion: this operation has the opposite effect of dilation, that is, it shrinks the borders of objects by removing pixels according to a structuring element. This kind of operation is normally applied to separate objects that for some reason appear merged, as is the case in the works by Salis and Pereira [7] and Mauricio et al. [9], and/or to eliminate small spurious objects, as proposed by Stajnko et al. [10], Shen et al. [11], Poomcokrak and Neatpisarnvanit [3], and Yan and Shen [12].

3) Closing: the closing operation consists in a dilation followed by an erosion, using the same structuring element. Its dual operation, *opening*, is presented in the next subsection. Dilation, performed first, tends to close gaps and holes present in the object, while erosion shrinks the objects back to the original size, without allowing the holes to reappear. Therefore, this operation is closely related to dilation, but it is less destructive, as the original dimensions are held. This operation is used by Refai et al. [13], Lippold et al. [14], Goyal [15], Mello et al. [16], Ahn et al. [17], Bazi et al. [18], Skodras et al. [8], Toh et al. [19], Zhao and Li [20], and Pedrosa et al [21].

4) Opening: the opening operation is the dual operation of closing, consisting of an erosion followed by dilation, using the same structuring element. Erosion, applied first, tends to eliminate small objects and spurious protrusions, while dilation, applied in the sequence, grows the objects back to their original sizes. As a result, this operation produces approximately the same effects as erosion alone, however keeping the main morphological features of the object of interest. This operation is used by Bewes et al. [22], Wijethunga et al. [23], Bazi et al. [18], Zhao e Li [20], and Pedrosa et al. [21].

5) Skeletonization: this is the process of removing as many pixels as possible from an object without altering the general structure of the image [24], that is, after the pixels are removed, the object should still be recognizable. There are different skeletonization algorithms, however the resulting patterns do not vary much among them. A particular form of skeletonization is *thinning*, in which the resulting skeleton has a 1-pixel width.

Sossa and Guzmán [25] and Sossa et al. [26] skeletonize the objects in order to identify certain points, whose number and characteristics are used to estimate the number of objects. Embleton et al. [6] use skeletonization to make phytoplankton filaments more identifiable and separable. Gajendran and Rodriguez [27] use the thinning technique to skeletonize chromosomes. Lippold et al. [14] apply this technique to the problem of counting radon tracks.

6) Hole filling: although this is not usually classified as a morphological operation, hole filling acts over the morphological characteristics of the image by removing parasitic holes that may arise from other operations such as thresholding (see Section 2.G.1). In some cases, hole filling is performed by means of other morphological operations, like dilation and closing. Hole filling can also be obtained by filtering (see Section 2.B). This subsection presents techniques that do not apply any of those options.

One of the most direct ways to eliminate holes consists in the identification of background regions (normally black) totally surrounded by an object (normally white), with those regions being then incorporated to the object. This approach is adopted by Embleton et al. [6], D'Souza [28], Goyal [15], Poomcokrak e Neatpisarnvanit [3], Kothari et al. [29], Han et al. [30] and Zhang et al. [31]. Additional rules can be applied; for example, the hole may be filled only if it is smaller than a certain threshold, like proposed by Azuela et al. [32].

B. Filtering

Filtering is the process of convolving a kernel (usually a square matrix) with an image. This operation may serve one of three main objectives:

i) Smoothing: removal or reduction of noise, artifacts and spurious objects (low-pass filter).

ii) Sharpening: improves contrast between contiguous regions (high-pass filter).

iii) Directional emphasis: aims to highlight objects and structures with a certain spatial orientation (directional filtering).

Some of the most used filters are presented in the following.

1) Median Filter: the median filter aims to replace the value of the centermost pixel inside the filtering window by the value of the median of all pixels inside that window. The objective is to reduce noise, being particularly effective to remove spurious points and to reduce "salt-and-pepper" noise [33]. This filter has the desirable property of reducing noise and preserving edges at the same time. On the other hand, the lack of information in the edges of the image may cause those areas to remain noisy, and undesirable distortions may be introduced. Those problems can be minimized by applying certain rules to the problematic areas. This filter is arguably the most widely employed in digital image processing.

Embleton et al. [6] apply this filter to smooth the perimeter of the regions of interest. Gajendran and Rodriguez [27] use the median filter to remove salt-and-pepper noise and also small holes inside objects (chromosomes). Flores et al. [34] apply the median filter five times in order to enhance the contrast of the objects edges (larvae). Su et al. [35] compbine a median filter with pulse-coupled neural networks in order to remove noise from images containing blood cells. Besides those, many other works employ this kind of noise-reducing filter [8,28,30,31,36-41].

2) Mean filter: The mean filter is also used for image smoothing and noise removal, which is done by replacing the value of the center pixel by the mean of the pixels inside the window. In comparison with the median filter, this one deals better with images strongly affected by noise, and it is also computationally cheaper. On the other hand, it is less effective in preserving edges and under mild noise conditions.

Dahle et al. [36] use the mean filter for image smoothing, which is followed by a Laplacian filter for edge sharpening. Salis and Pereira [7] apply the mean filter three times in order to remove enough noise. Mean filter for smoothing is also employed by Marotz et al. [42], Shen et al. [11], Wijethunga et al. [23], Zhang et al [43] and Kothari et al. [29].

3) Gaussian filter: The Gaussian filter is similar to the mean filter. However, in this case, the degree of smoothing is controlled by the standard deviation of the Gaussian used as kernel - the smaller is the standard deviation, more weight is given to the center pixel, resulting in less smoothing.

Cosío et al. [4] use the Gaussian filter before a watershed algorithm in order to avoid excessive division. Bewes et al. [22], Kumagai et al. [44] and Zhou et al. [45] use this filter to reduce high frequency noise.

4) Laplacian filter: the Laplacian filter is often used for computing the second derivatives of an image. Such an information may be explore to detect object edges. The kernels of Laplacian filters usually have negative values arranged into a cross pattern.

Netten et al. [46] apply a non-linear Laplacian filter to separate touching objects in interphase cell nuclei. As commented before, Dahle et al. [36] use this filter to sharpen edges that were smoothed by a mean filter. Flores et al. [34] apply the Laplacian filter three times (together with a median filter applied five times) to highlight the edges of the objects of interest (larvae).

5) Laplacian of Gaussian filter: Due to their derivative nature, Laplacian filters are very sensitive to noise. In order to avoid distorted results, it is usual that a Gaussian filter be applied before the Laplacian, hence the name "Laplacian of Gaussian". The setup of this filter must take into account the compromise between noise reduction (Gaussian filter) and edge sharpening (Laplacian filter). This kind of filter is used by Cosío et al. [4].

6) Others: Adaptive filters, whose kernels are modified along several iterations until a stable result is achieved, are employed by Chen et al. [47] and by Deng et al. [48]. Disperati et al [49] use a local maxima filter to identify the points with maximum reflectance, which, in turn, reveal the positions of the objects (trees). Wu and Kuo [50] apply a filter for horizontal noise removal, preserving vertical features.

C. Contrast Enhancement

Techniques to enhance contrast aim to improve the distinction between the objects and the rest of the scene. This section presents three techniques.

1) Normalization: The normalization, also known as contrast extension, is the simplest way to enhance contrast, and was adopted by Refai et al. [13], Poomcokrak and Neatpisarnvanit [3], Skodras et al. [8], Shen et al. [41] and Pedrosa et al [21]. It modifies the pixel values in such a way the darkest pixel is made zero, and the brightest one assumes the full-scale value (255 for images quantized with 8 bits). There are many ways to perform the normalization; one of the simplest ones is given by the expression

$$\overline{P}(x,y) = \frac{P(x,y) - P_{min}}{P_{max} - P_{min}} \cdot P_{vm} \quad (1)$$

where x and y are the pixel coordinates, $P(x,y)$ and $\overline{P}(x,y)$ are, respectively, the original and current pixel values, P_{min} and P_{max} are, respectively, the minimum and maximum values of the pixels in the original image, and P_{vm} is the full-scale value.

The result of applying this technique is that dark pixels become even darker, and bright pixels become brighter, increasing the contrast. There are some problems with this approach. First, if there is significant lighting variation over the image, the difference between the darkest and the brightest pixels will be artificially large, reducing the effectiveness of equation 1. This problem may be partially overcome by using local normalization, solution adopted by Marotz et al. [42]. Another possible problem is that a few spurious pixels may reduce the procedure effectiveness. For example, very dark or very bright pixels may be present due to the salt-and-pepper effect, effectively rendering equation 1 useless. This problem may be partially solved by means of a smoothing filter.

2) Histogram equalization: this equalization is a sophisticated way of enhancing contrast by manipulating the image histogram into some desired shape. Contrary to normalization, histogram equalization can employ non-linear and non-monotonic functions to map the original pixel values into the desired ones.

Zhang et al. [43] use contrast limited adaptive histogram equalization (CLAHE) [52], which considers small regions instead of the entire image; neighbor regions are later combined by linear interpolation in order to remove artificially introduced edges. Wu and Kuo [50] histogram equalization to minimize the influence of lighting conditions on the object detection. Other methods that use this strategy were proposed by Yan and Shen [12] and Zhang et al. [31].

3) Principal component analysis (PCA): The PCA is a statistical technique originally proposed by Pearson [53] more than a century ago. It uses an orthogonal transformation to generate, from a set of possibly correlated variables, a set of non-correlated variables called *principal components*. Loukas et al. [54] applied this technique for contrast enhancement of certain cells amidst a wide variety of other kinds of cells and objects. In this context, PCA may also be seen as a segmentation technique, since it is capable of discriminating and enhance the objects of interest.

D. Morphological Characteristics

In many cases, the objects to be counted have distinct morphological characteristics that vary little from one element to another. In those cases, it is convenient to introduce some rules in order to discard detected objects that seem to not fit the expected morphological characteristics. This section presents some rules based on three of those morphological characteristics.

1) Area: It is expected that all kinds of objects have a size within certain limits. Thus, it is likely that an element whose area is outside those limits is of another type or, if the area is larger than the upper limit, it may be the result of multiple merged objects. Rules based on the area may instruct the algorithm to either discard the object, or to take another action to identify the identity of the element. This kind of situation is found in the work by Benali et al [55], Refai et al. [13], Dahle et al. [36], Lee et al. [56], Bewes et al. [22], Yan and Shen [12], Costa and Yang [57], Hamghalam and Ayatollahi [5], and Kothari et al. [29].

In some cases, area is used as a feature that feeds either a classifier or a set of rules that will identify and count the objects, as proposed by Mussio et al. [58] and Embleton et al. [6].

Some proposals explore a more direct approach: after a cluster of objects is detected, its area is estimated and divided by the typical area expected for individual objects, resulting in an estimate for the number of objects. This approach was adopted by Sokkarie and Osborne [59], Mello et al. [16], Ahn et al. [17], Ates and Gerek [39], Gusmão et al. [60], Toh et al. [19], and Nguyen et al. [61].

2) Shape: In many cases, it is expected that the objects have certain shapes. In biological images, those shapes are almost always circular or elliptical. This can be explored by conditions that the objects must meet in order to be considered, otherwise they are discarded. The values of circularity and ellipsity - how close a shape is from a circle or an ellipsis, respectively - can be submitted to a classifier that identifies and labels the objects to be counted.

Circularity is explored by Sokkarie and Osborne [59], Dahle et al. [36], Deng et al. [48], Goyal [15], Ates and Gerek [39], Costa and Yang [57], Ren et al. [62], and Schier and Kovar [63]. Ellipsity is used by Mussio et al. [58] and Lee et al. [56].

3) Concavities: The objects to be counted are usually convex, that is, a line linking any two points inside the object is always entirely contained by that object. In these cases, the presence of concavities is a strong indicator that the object is either spurious, or there are multiple touching objects forming clusters, which, in general, is a difficult problem to be solved. This approach was adopted by Azuela et al. [32], Ahn et al. [17] and Kothari et al. [29].

E. Transforms

The term *transform* refers to an alternative mathematical representation of an image, created from the traditional representation as a function of two spatial variables [1]. They are powerful tools, being largely used in a large variety of problems that go way beyond digital image processing.

Among the many transforms found in literature, seven were identified as being used in the context of object counting, as will be seen in the following.

1) Hough Transform: The Hough transform was proposed in 1962 [64], and its original idea was to combine an accumulator space with a voting procedure to detect lines and curves. Later, Ballard [65] proposed the generalized Hough transform, which is capable of detecting any arbitrary shape. A formal definition for the Hough transform can be found in Princen et al. [66].

According to Sonka et al. [67], the Hough transform has several desirable characteristics: it is capable of recognizing partial and lightly deformed shapes, being capable of recognizing partially occluded objects; can be used to measure the similarity between a model and a detected object; it is robust in the presence of additional structures in the image; it is relatively insensitive to noise; among others. Those characteristics make the Hough transform an attractive tool to be used in the context of object counting, in which case it should be preceded by an edge detector.

Barber et al. [2] use a compact version of the Hough transform to highlight the centers of circular objects. It was designed to emphasize circular shapes and to suppress straight lines, and was optimized to speed up the image processing.

Bewes et al. [22] implemented the complete version of the generalized Hough transform to detect round objects. According to the authors, the computational burden had prevented its use until that moment (2008) but, with more computational resources available, this option became viable, resulting in the algorithm called CHiTA.

Syed et al. [68] combined the Hough transform with a circular filter to create the so-called Halo transform. According to the authors, this transform is more effective in detecting superimposed circles.

Xiong et al. [69] use the *multi-scale circular Hough transform* (MSCHT) proposed by Smereka and Duleba [70] in order to deal with size-varying circular objects.

Zou et al. [71] also use a modified version of the Hough transform to detect circular objects. In their approach, a gradient field of the image intensity is computed. This field is converted into an accumulation matrix, whose peaks are detected by a Laplacian of Gaussian filter. Those peaks correspond to the positions of the objects to be counted.

2) Distance transform: distance transforms are usually applied to binary images. This kind of transform specifies the distance between each pixel in the image and the closest non-zero pixel. Distance transforms have a very important role in the comparison between binary images, especially in the case of the images resulting from the application of techniques for local feature detection, like edge and corner detection [72].

In the context of object counting, many segmentation decisions can be based on the knowledge of how "deep" a pixel is in an image, and that can also help watershed algorithms to generate the catchment basins. This combination of distance transform with the watershed algorithm (see Section 2.G.2) was used by Men et al. [73] and by Zhao et al. [20]. If the edges of the objects can be properly isolated, either by edge detection (see Section 2.F) or by thresholding (see Section 2.G.1), applying the distance transform will generate peaks at the centers of the objects, revealing their locations. This was the approach chosen by Mukherjee et al. [74] and Nguyen et al. [61].

3) Fourier transform (FT): The Fourier transform expands a signal as an infinite linear combination of complex exponentials, with variable magnitudes, frequencies and phases. As a result, it breaks the signal into its constituent frequencies, the so-called frequency spectrum. In practice, it is usual to employ the sampled version of the FT, the so-called Discret Fourier Transform (DFT). Both the input and output of the DFT are discrete, making it more appropriate for computational manipulations. Another convenient aspect of DFT is that it has an algorithm for fast computation, the Fast Fourier Transform (FFT).

The FT has been used in a wide range of applications, such as image restoration, enhancement and compression, but its use has been incipient in object counting. To the authors knowledge, the only counting method that explicitly uses the FT was proposed by Johnson et al. [75]. The authors use a variation of the short-space Fourier analysis to count treads in a painting. The horizontal and vertical weave densities can be used to verify the authenticity of paintings. The characteristics and peculiarities of this problem are quite different from other counting problems, which can explain why the FT was successfully applied in this case, while it is absent in other circumstances.

4) Wavelet transform (WT): As commented before, the Fourier transform provides information about the frequencies present in the signal, but does not give a clue about their location. The short-space FT solves this problem dividing the image into small regions and treating each one of them separately [67]. The problem with this solution is the well-known space-frequency resolution tradeoff: small regions will result in good spatial resolution, but poor spectral resolution, and vice-versa.

The wavelet transform approaches this problem differently, minimizing the tradeoff effects. There are two main differences between wavelet transform and other integral transforms, like FT. First, the basis functions, called wavelets, have the ability to offer, to a certain point, location information. Second, the analysis is performed in multiple scales (multiple resolutions), resulting in a more compact representation of the signals, especially in the case of localized phenomena, like the corners of the image. The location in the spatial domain, together with the frequency location of the wavelets, often result in a sparse representation of the image [67]. This favors the use of WT in applications such as image compression, noise filtering, and feature extraction.

Bernal et al. [76] use wavelet-based templates to identify cells in an image. Gao et al. [77] use a non-decimated complex wavelet transform (shift invariant) to extract texture information. Such an information feed a watershed algorithm, which in turn performs the segmentation. Tai and Cheng [78] combine the WT with a curvelet transform and a basis pursuit algorithm to count cell nuclei.

5) Curvelet transform (CT): the CT is a multi-scale directional transform that allows an almost optimal non-adaptive sparse representation of objects with edges [79]. Curvelets are based on multi-scale ridgelets combined with a spatial pass-band filtering to isolated the different scales [80]. The curvelet transform was developed as a solution for a problem that affects the wavelet transform: in order to properly reconstruct the borders of the image, many coefficients are needed [79]. Besides, according to Tai and Cheng [78], wavelets do not deal properly with curves. Because of that, they combined both transforms in order to explore their strengths.

6) Top-hat transform: there are two types of top-hat transform, the *white*, which is obtained by the difference between an image and its opening, and the *black*, which is obtained by the difference between the closed and original images. The white top-hat produces an image containing all objects in the original image that are both smaller than the structuring element used to open the image, and brighter than their surroundings. On the other hand, the black top-hat transform produces an image containing that are both smaller than the structuring element, and darker than their surroundings. Therefore, the top-hat transform is useful to detect small objects.

Netten et al [46] explore the characteristics of the white top-hat transform to highlight small fluorescent points. Salis and Pereira [7] use the same transform to correct non-uniform lighting in the image containing the objects to be counted.

7) Fast radial transform (FRT): The FRT is a simple and fast operator based on gradient, whose objective is to detect regions with high radial symmetry [81]. It was inspired by the results of the generalized symmetry transform [82] but, instead of considering the contribution of a local neighborhood to a central pixel, it determines the contribution of each pixel in the symmetry of the pixels surrounding them. Schier and Kovar [63] use this transform in substitution to the Hough transform, which, according to them, performs poorly when the objects to be counted are clustered.

F. Edge and Contour Detection

In the context of object counting, the detection of contours and edges has as main objective to find the limits between the objects of interest and their surroundings. The terms *edge* and *contour* often are used as synonyms but, although the final objective is usually the same, they are conceptually different. According to Gonzalez and Woods [83], the contour of a finite region results in a closed path, thus being a global concept. On the other hand, edges are composed by pixels with derivative values that exceed a certain threshold, that is, its existence is based in a measurement for the discontinuity of the grayscale, thus being a local concept. Gonzalez and Woods also establish that the edge points may be connected into edge segments, which in turn may be connected in such a way they correspond to contours, but this is not always the case.

In the case of object counting, the final objective is to identify contained elements, so, at some point, detected edges must be connected in order to correctly delineate the objects. For that reason, edge and contour detection techniques were grouped in the same section.

1) Binary image-based techniques: This is the most simple and direct method for detecting edges in an image. Before detection, the image is thresholded (see Section 2.G.1) and, in most cases, filtered to remove parasite objects and holes. Then, the edges are detected simply by detecting where pixels with different values touch. This is the approach adopted by

Azuela et al. [32], Ahn et al. [17], Kothari et al. [29] and Han et al. [30].

2) Sobel method: The Sobel edge detector is based on the Sobel operator, which is a discrete differentiation operator that computes an approximation for the gradient of the image's intensity function. The Sobel operator is relatively cheap computationally, since it perform the convolution of the image with a small integer-valued filter. The main disadvantages of this kind of detector is its inadequacy to approximate high frequency variations, and the high sensitivity to noise. Barber et al. [2] use two perpendicular Sobel operators to find the edges. Poomcokrak and Neatpisarnvanit [3] use the Sobel operator together with the Canny method to find the edges of red blood cells.

3) Canny method: Canny proposed an approach for edge detection that is optimal for step edges corrupted by white noise. The optimality of the detector is related to three criteria: the detection criterion, which expresses the fact that important edges should not be missed and that there should be no spurious detection; the location criterion, which states that the distance between the real and detected edge positions should be minimum; the unique solution criterion, which combines multiple solution into a single edge. This last criterion makes the method effective under noisy conditions [67]. This method is used by Gonzalez et al. [85] and Toh et al. [19].

4) Laplacian of Gaussian method: The edge detector based on the Laplacian of the Gaussian is an operator that combines a smoothing operation performed by a Gaussian filter, with a differentiation operation performed by a discrete Laplacian filter (see Laplacian of Gaussian filter in Section 2.B.5). The edges are located by identifying zero-crossings in the resulting output. Loukas et al. [54] adopted this approach because, according to them, the smoothing operation performed by the Gaussian filter is desirable under the highly noisy conditions found in histological material. This method is also used by Nasution and Suryaningtyas [86].

5) Prewitt method: The Prewitt operator is similar to Sobel's (Section 2.F.2), since both try to approximate the gradient values. The only difference is that they use slightly different kernels. This operator was used by Prasad and Badawy [38].

6) Active contours: active contours, also known as *snakes*, is a technique proposed by Kass et al. [87], used to delineate the contour of objects. The snakes energy minimizing splines, that is, they are active shapes that can be deformed and moved according to energy minimizing criteria. A snake can be seen as a rubber band that deforms to fit the contour of an object. In order to work correctly, the snakes need additional information provided by either the user, or another tool of image analysis. Initial snakes may have any shape, and should be located near the objects of interest. Theerapattanakul et al. [88] use this technique to count leukocyte.

7) Complex diffusion method: Miranda et al. [89] explored the characteristics of the complex diffusion filters proposed by Gilboa et al. [90] to develop a new strategy for edge detection applied to the problem of counting cashew trees. They observed that, by simply manipulating the complex diffusion coefficients, the imaginary part of the filter becomes a good approximation for the second derivative of the image, making it appropriate for edge detection.

G. Segmentation and Classification Techniques

This section presents techniques that aim to segment images in order to discriminate between objects of interest and the remainder of the scene. This section also includes classification techniques that, besides segmenting, also label regions, which is necessary when multiple types of objects need to be identified and counted. Segmentation and classification techniques are, in most cases, the last step before the counting, being part of virtually all algorithms for object counting. Because of that, this section is extensive, describing almost twenty different approaches for the problem.

1) Thresholding: Thresholding, also known as binarization, is a technique that quantizes pixel values of an image into two levels. The decision of which value will be assigned to each pixel is based on a threshold. The general idea is that pixels belonging to the objects receive a certain value (usually 1), and the remainder pixels receive the other (usually 0). In a few cases, the thresholding process may include more than two levels, normally for discriminating different types of objects. Thresholding with multiple levels was adopted by Goin et al. [91] and Cordiki et al. [92].

As can be seen, the selection of the threshold value is very important. It must be chosen in such a way at least some pixels of each object are identified as such, avoiding, as much as possible, that background pixels be identified as part of an object. If the objects and their neighborhood are homogeneous and highly contrasting, this task is simple. However, in real images, this is rarely the case. There are several methods to determine appropriate thresholds for each situation, as will be seen in the following.

One of the most common strategies to select the threshold is to search, empirically, for the best separation between objects and background, having as basis a set of typical images that represent the conditions to be tackled by the algorithm under development. This approach was adopted by Osowsky and Gamba [93], Benali et al. [55], Embleton et al. [6], Stajnko et al. [10], Salis and Pereira [7], Mauricio et al. [9] and Pedrosa et al. [21].

Another common way of determining the threshold was proposed by Otsu [94]. The algorithm assumes that the image to be thresholded has two classes of pixels (e.g., objects and background), and then calculates the class separation threshold that minimizes the intra-class variance and maximizes the inter-class variance. This approach was adopted by Mukherjee et al. [74], Sossa et al. [26], Gajendran e Rodriguez [27], Selinummi et al. [95], Deng et al. [48], Syed et al. [68], Zhang et al. [51], Ates and Gerek [39], Xiong et al. [69], Nguyen et al. [61] and Zhang et al. [31].

In the adaptive thresholding, the image is divided into regions and different thresholds are calculated and applied to each one of them. This approach is appropriate when it is expected that different parts of the image have different characteristics in terms of objects, background, or both. This strategy was adopted by Marotz et al. [42], Fang et al. [96], Refai et al. [13], Flores et al. [34], Mitev et al. [97] and Shen et al. [41].

Methods based on the histogram explore explore the image histogram characteristics to infer the best threshold. The most common strategy is the balanced histogram thresholding, which weights and removes bars of the histogram iteratively, until an equilibrium is achieved. This is the method used by Lucarini et al. [98]. Other histogram-based criteria may be used [5,77], and prior knowledge about the image characteristics may be explored [54]. Dahle et al. [36] use a histogram-based approach that requires user intervention.

Histeresis thresholding uses a histeresis loop to identify connected objects. Any pixel above a certain upper threshold is made white, and the surrounding pixels are then recursively considered. If their values are above the lower threshold, they are also made white. As a result, fewer spurious objects are detected [99]. This method was used by Gajendran and Rodriguez [27] and Skodras et al. [8].

The isodata approach calculates the threshold t solving the expression $t = 0{,}5 \cdot (p_1 + p_2)$, where p_1 is the average value of all pixels with value equal to or smaller than t, and p_2 is the average of all pixels with value greater than t. This strategy was used by Netten et al. [46].

Many other strategies for threshold selection can be found in the literature. They are based on techniques such as Discriminant Analysis (adopted by Kumagai et al. [44]), the Huang method (used by Mello et al. [16]), successive iterations [43], color similarity [51], modified Otsu [62] and double thresholding [100].

In some cases, the authors state that thresholding is applied, but do not provide any information about the type of selection adopted [4,11,12,17,19,25,32,37,40,57,62,87,101,102].

2) Watershed: According to Gonzalez and Woods [83], the concept of watershed is based on the visualization of the image in three dimensions, the two space coordinates plus the gray tones. In such interpretation, there are three types of points: (*i*) points pertaining to local minima; (*ii*) points from which a water drop would flow to a local minimum; (*iii*) points from which a water drop can flow to more than a local minimum with equal probability. The set of points satisfying condition (*ii*) for a certain local minimum is called *catchment basin* of that minimum. Points satisfying condition (*iii*) are called watershed lines.

According to this topographic point of view, the objects often present themselves as vales, crests, plateaus, etc., that is, they can appear as prominent elements in the tridimensional landscape. In this context, applying the watershed algorithm may yield good results.

Cosío et al. [4] apply a modified watershed algorithm to binary images filtered by a Laplacian of Gaussian filter to identify individual cells. Fang et al. [96] also apply a modified version of the algorithm, called *dynamic water immersion*, to identify tumor cells. In order to avoid oversegmentation, Deng et al. [48] proposed the so-called *modified labeled watershed algorithm* to extract follicular cysts. Gao et al. [77] combine the watershed algorithm with an adaptive threshold segmentation to identify leukocytes. Ates and Gerek [39] apply the watershed algorithm twice according to an appropriate test. Zhao et al. [20] present a modified watershed algorithm based on opening and closing operations and on the distance transform.

Other methods using watershed were proposed by Chen et al. [47], Nudol [37], Selinummi et al. [95], Chen and Zhang [103], Men et al. [73], Nasution and Suryaningtyas [86], Costa and Yang [57] and Skodras et al. [8].

3) Template matching: in the template matching, the objects, in order to be counted, must present a certain similarity with one or more references selected *a priori* to represent the entire set of objects of interest. Those references may be characterized by one or more features, like shape, texture, color, area, among others. This kind of approach is more appropriate when the objects are expected to be relatively homogeneous among them.

Forshaw and Wiles [104] calculate the correlation between the image and templates selected to identify volcanoes in images of Venus surface. Hinz [105] uses a hierarchical template matching, in which each level increases the model detail level, in order to detect different types of cars in aerial images. The problem tackled by Shimada et al. [106] is a little different, since they try to count cell nuclei using 3D images, obtained by stacking 2D images; the nuclei are detected by matching these with spherical voxels (the 3D equivalent to pixels), which work as templates. Round templates are used by Zhang et al. [43] to detect the extremities of stacked steel bars. In the approach adopted by Vibha et al. [40], the templates are automatically generated for each image, instead of being determined *a priori*; such an approach is used to detect trees in aerial images. Kothari et al [29] use elliptical templates to detect cells.

4) Connected component labeling: This is a simple and widely used method to extract blobs, and it is usually applied after image thresholding (see Section 2.G.1). The method identifies, groups and labels connected pixels. In 3D images, there are two connectivity criteria: the 4-pixel criterion, in which pixels are considered connected if they touch vertically or horizontally, and the 8-pixel criterion, in which diagonal adjacency is also considered. The result is a list of all isolated entities in the image, which ay correspond to the objects to be counted. Additional rules can be applied to refine the results, like area thresholds (see Section 2.D.1) to remove too small entities and to divide too big elements. The number of entities remaining is often taken as an estimate for the number of objects in the image.

This method was adopted by Mukherjee et al. [74], Lucarini et al. [98], Ishizu et al. [107], Mello et al. [16], Men et al. [73], Nasution and Suryaningtyas [86], Tai and Cheng [78], Yan and Chen [12], Ahn et al. [17] and Gusmão et al. [60].

5) Artificial neural network: Artificial neural networks (RNA) are mathematical or computational models that try to emulate biological neural networks. RNAs are composed by a number of interconnected nodes, called neurons. Each connection has a weight that is multiplied to the signal passing through it. Those weights are predefined by a training process. Each neuron has an activation function associated, which converts incoming signals into an output activation. The theory related to neural networks is extensive, with several books dedicated to this subject (for example, Haykin [108]).

Neural networks are applied to object counting when there are different types of objects to be counted separately, or when the objects of interest must be identified amidst several other objects that are not to be considered.

There are several types of neural networks with different topologies, strategies and uses. At least four of them have been used in the context of object counting, as will be seen in the following.

One of the most used neural networks is the *Multilayer Perceptron* (MLP). This is a feedforward artificial neural network model that maps input data into a suitable output. An MLP has a certain number of connected layers (usually three), each of them containing a certain number of neurons with a non-linear activation function associated. Training is performed using the backpropagation technique. Comunello et al [109] use MLP to classify segments in computer tomography images. Embleton et al. [6] apply this technique to discriminate between different groups of phytoplankton. MLPs are also used by Nasution and Suryaningtyas [86], Poomcokrak and Neatpisarnvanit [3] and Zhou et al. [45].

Cellular neural networks (CNN) are also used often in the context of object counting. The main difference between this approach and other types of neural networks is that, in this case, communication is allowed only between neighbor units. The formal definition of this kind of network, as defined by its creators [110], is extensive and will not be included here. CNNs are especially useful in digital image processing due to their flexibility and fast computation. Seiler [111] applies CNN to count small objects. Feng et al. [112] present a CNN optimized for counting nucleated cells. Fasih et al. [113] explore CNNs as basis for a very fast counting algorithm. Sevgen et al. [114] introduce a hardware implementation of a counting algorithm based on CNN.

Other less common neural networks were used by Schaich et al. [115] (probabilistic neural network) and by Su et al. [35] (pulse-coupled neural network).

6) Support vector machines (*SVM*): SVMs are classifiers that use an N-dimension hyperplane to optimally separate the data into two categories. This makes this tool appropriate for object counting, as images normally have two classes to be discriminated (objects and background). SVM models are strongly related to neural networks - an SVM using a sigmoidal kernel is equivalent to a two-layer perceptron neural network.

Chen and Chang [103] use an SVM to separate a certain variety of bacteria colonies from other lineages. Christophe and Inglada [116] use this technique to discriminate and classify several types of elements in remote sensing aerial images. Xiong et al. [69] use three two-class SVMs to classify different regions in blood images. Zhang et al. [31] employ this technique to count bacteria in food samples.

7) k-Means: this is a simple clustering method that groups the elements of an image into a certain number of classes. The idea is to define *k* clusters, each with a corresponding mean. Each element in the image is then associated to the cluster with the closest mean. Cointault and Gouton [117] use the k-means algorithm to separate wheat ears from the remainder of the scene. Gusmão et al. [60] use the channels *a** and *b** of the L*a*b* color space, together with the k-means algorithm, to group the pixels into three distinct classes: *Aedes Aegypti* eggs, egg traps, and intermediate regions. Kothary et al. [29] use this algorithm to generate a binary mask directly from the RGB representation of the image, where the objects to be highlighted are cellular nuclei; the initial points of the algorithm are selected by the user.

8) Color channel segmentation: color images are usually composed by three channels (for example, red, green and blue in the RGB representation). In some cases, the objects of interest may present a strong contrast with respect to its surroundings in one or two of those channels. This can be explored in order to adequately segment the image.

Schönholzer et al. [118] use the green channel of RGB images to separate bacteria amidst debris. Wijethunga et al. [23] use the *a** and *b** channels of the L*a*b* color space to identify and segment kiwi fruit. Zhang et al. [51] use a measure for color similarity in the HSV representation to aid the process of edge detection.

9) Others: as stated before, there are many different segmentation and classification techniques used in the context of object counting. Due to space restrictions, it is not possible to describe all of them. This section aims to cite and offer some references related to those techniques, including: fuzzy logic (Marotz et al. [42] and Montseny et al. [119]), genetic algorithms (Miranda et al. [89] and Zhou et al. [45]), mean shift (Lee et al. [56] and Christophe and Inglada [116]), Bayes classifier (Cosío et al. [4]), elastic energy model (Comunello et al. [109]), fractal dimensions (Li e Qi [120]), fuzzy clustering (Selinummi et al. [121]), Gabor filter (Sun et al. [122]), Gaussian process classifier (Bazi et al. [18]), hidden Markov models (Lu and Qin [123]), *k* nearest neighbors (Tek et al. [100]), Hu's invariant moments (Ahn et al. [17]), neurofuzzy (Ruz et al. [124]), recursive segmentation (Prasad and Badawy [38]), and Munsell soil diagram [125].

H. Other Techniques

This section presents the techniques that do not fit any of the adopted classes.

The *background subtraction* can be seen to a more direct alternative to filtering. The objective is to remove the background, in such a way only the objects of interest remain. There are many techniques to achieve this objective, the simplest one being capturing the same scene with the objects present and absent, under the exact same conditions, and then subtracting the first image from the second. There are more sophisticated approaches, but this simple one is usually preferred in the context of object counting. Osowsky and Gamba [93] and Zhang et al. [31] apply this technique to the problem of counting bacteria. Kumagai et al. [44] use background subtraction to count asbestos particles.

SIFT (Scale-Invariant Feature Transform) [126] and its variation, SURF (Speeded Up Robust Feature) [127], are image descriptors largely used in many areas of digital image processing. They extract a large number of features from an image, which can be used, for example, to characterize objects in images. Despite that, their use in object counting has been limited. To the author's knowledge, the only method to explicitly use this kind of descriptor was proposed by Lempitsky and Zisserman [128], in which the object counting problem is seen as an estimation of an image density whose integral over any region provides an estimate for the number of objects in that region.

Some methods are inherently algorithmic and do not use any established method. This is the case of the proposal by Vinod and Murase [129]. Other proposals that use unorthodox approaches are described in [130-132].

III. Conclusion

This paper presented a comprehensive survey of works dedicated to the subject of object counting in digital images. The survey was presented in a tutorial form, in which the main techniques and strategies are divided in eight groups. The text was written in such a way it may be useful both as a first contact with the subject, as a reference and idea repository for experts on the subject. The description of each technique was kept succinct for concision reasons, but the text cites several references that can be consulted for more detail.

The references cited in the text cover a wide range of applications. This is a consequence of the search depth of this work, being also a deliberate attempt to contemplate the largest number of potential readers. Table 1 presents all references cited along the text, together with their main application.

TABLE I
APPLICATIONS AND RESPECTIVE REFERENCES.

Aplication	References
Cell	[2],[3],[4],[5],[9],[13],[22],[28],[29],[35],[36],[38],[47],[54], [55],[58],[61],[68],[69],[76],[77],[86],[88],[91],[96],[106],[112],[119],[122]
Bacteria	[15],[31],[39],[41],[42],[43],[73],[74],[92],[93],[95],[103],[118]
Trees	[18],[40],[49],[59],[85],[89]
Insects	[11],[16],[60],[123]
Fruits	[10],[23]
Soil Features	[125]
Fungi Colonies	[63],[98]
Pollen	[57],[101]
Ears	[30],[117]
Chromosomes	[12],[27]
Wood Defects	[120],[124]
Fish	[19],[130]
Grains	[20]
Malaria Parasite	[70],[100]
Remote Sensing	[116]
Ovaries	[8],[48]
Fluorescent points	[46],[78],[121]
Geological Elements	[21],[104]
Asbestos	[44],[107]
Radiation Tracks	[14],[97],[131]
Steel Bars	[7],[51]
Cars	[105]
Sperm	[62]
Plankton	[6]
Larvae	[34]
Calcifications	[109]
Bread Defects	[17]
Urine Structures	[45]
Moles	[56]
Printed Circuits	[50]
Gamma Ray Sources	[115]
Precious Stones	[37]
Painting Treads	[75]
Generic Objects	[25],[26],[32],[102],[111],[113],[114],[128],[129],[132]

References

[1] Mathworks, *Image Processing Toolbox User's Guide*, 2012. Available: http://www.mathworks.com/help/pdf_doc/images/images_tb.pdf.

[2] P. R. Barber, B. Vojnovic, J. Kelly, C. R. Mayes, P. Boulton, M. Woodcock, and M. C. Joiner, "Automated counting of mammalian cell colonies," *Phys. in Medicine and Biology*, vol. 46, pp. 63–76, 2001.

[3] J. Poomcokrak and C. Neatpisarnvanit, "Red blood cells extraction and counting," in *Proc. Int. Symp. Biomedical Eng.*, 2008, pp. 199–203.

[4] F. A. Cosío, J. A. M. Flores, M. A. P. Castaneda, S. Solano, and P. Tato, "Automatic counting of immunocytochemically stained cells," in *Proc. Int. Conf. of the IEEE EMBS*, 2003, pp. 790–793, 2003.

[5] M. Hamghalam and A. Ayatollahi, "Automatic counting of leukocytes in Giemsa-stained images of peripheral blood smear," in *Proc. Int. Conf. Digital Image Process.*, 2009, pp. 13–16.

[6] K. V. Embleton, C. E. Gibson, and S. I. Heaney, "Automated counting of phytoplankton by pattern recognition: a comparison with a manual counting method," *Journal of Plankton Research*, vol. 25, pp. 669–681, 2003.

[7] T. T. Salis, G. A. S. Pereira, "Contagem automática de tarugos de aço por meio de visão computacional," Brasil, *Relatório Técnico*, 2006.

[8] A. Skodras, S. Giannarou, M. Fenwick, S. Franks, J. Stark, and K. Hardy, "Object recognition in the ovary: Quantification of oocytes from microscopic images," in *Proc. Int. Conf. Digital Signal Process.*, 2009.

[9] C. R. M. Mauricio, F. K. Schneider, and L. Santos, "Image-based red cell counting for wild animals blood," in *Proc. Int. Conf. IEEE Eng. in Medicine and Biology Society*, 2010, pp. 438–441.

[10] D. Stajnko, M. Lakota, and M. Hocevar, "Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging," *Computers and Electronics in Agriculture*, vol. 42, pp. 31–42, 2004.

[11] W. Z. Shen, C. L. Zhang, and Z. L. Chen, "Research on automatic counting soybean leaf aphids system based on computer vision technology," in *Proc. Int. Conf. Machine Learning and Cybernetics*, 2007, pp. 1635–1638.

[12] W. Yan and S. Shen, "An automatic counting algorithm for chromosomes," in *Proc. Int. Conf. Bioinformatics and Biomedical Engineering*, 2008, pp. 2492–2494.

[13] H. Refai, L. Li, T. K. Teague, and R. Naukam, "Automatic count of hepatocytes in microscopic images," in *Proc. Int. Conf. on Image Process.*, 2003, pp. 1101–1104, 2003.

[14] J. Lippold, P. R. Marpu, R. Gloaguen, and R. Jonckheere, "Automatic counting of fission tracks using object-based image analysis for dating applications," in *Proc. IEEE Int. Geoscience and Remote Sensing Symp.*, 2007, pp. 440–443.

[15] M. Goyal, "Machine vision based bacteria-colony counter," Master's thesis, Thapar University, India, 2008.

[16] C. A. B. Mello, W. P. Santos, M. A. B. Rodrigues, A. L. B. Candeias, and C. M. G. Gusmão, "Image segmentation of ovitraps for automatic counting of *Aedes aegypti* eggs," in *Proc. Annual Int. Conf. IEEE Engineering in Medicine and Biology Society*, 2008, pp. 3103–3106.

[17] I. Ahn, M. M. Zo, and C. Kim, "A robust bread defect detection and counting system," in *Proc. Int. Conf. Advanced Comm. Tech.*, 2009, pp. 1315–1320.

[18] Y. Bazi, H. Al-Sharari, and F. Melgani, "An automatic method for counting olive trees in very high spatial remote sensing images," in *Proc. IEEE Int. Geoscience and Remote Sensing Symp.*, 2009, pp. 125–128.

[19] Y. H. Toh, T. M. Ng, and B. K. Liew, "Automated fish counting using image processing," in *Proc. Int. Conf. Computational Intelligence and Software Engineering*, 2009.

[20] P. Zhao and Y. Li, "Grain counting method based on image processing," in *Proc. Int. Conf. Information Engineering and Computer Science*, 2009.

[21] M. M. Pedrosa, P. Pina, E. A. Silva, J. R. Nogueira, "Detecção de crateras de impacto na superfície marciana por meio de técnicas de Morfologia Matemática," in *Anais XV Simpósio Brasileiro de Sensoriamento Remoto*, 2011, pp. 7698–7704.

[22] J. M. Bewes, N. Suchowerska, and D. R. McKenzie, "Automated cell colony counting and analysis using the circular hough image transform algorithm (CHiTA)," *Physics in Medicine and Biology*, vol. 53, pp. 5991–6008, 2008.

[23] P. Wijethunga, S. Samarasinghe, D. Kulasiri, and I. Woodhead, "Digital image analysis based automated kiwifruit counting technique," in *Proc. Int. Conf. Image and Vision Computing*, 2008.

[24] D. Azar, "Hilditch's algorithm for skeletonization," 1997. [Online]. Available: http://cgm.cs.mcgill.ca/~godfried/teaching/projects97/azar/skeleton.html.

[25] H. Sossa and G. Guzmán, "New method to count objects into an image," in *Proc. Int. Conf. Pattern Recognition*, 2000, pp. 470–473.

[26] H. Sossa, G. Guzmán, O. Pogrebnyak, and F. Cuevas, "Object counting without conglomerate separation," in *Proc. Mexican Int. Conf. Computer Science*, 2003, pp. 216–220.

[27] V. Gajendran and J. J. Rodriguez, "Chromosome counting via digital image analysis," in *Proc. Int. Conf. Image Processing*, 2004, pp. 2929–2932.

[28] A. C. D'Souza, "Automated counting of cell bodies using Nissl stained cross-sectional images," Master's thesis, Texas A&M University, USA, 2007.

[29] S. Kothari, Q. Chaudry, and M. D. Wang, "Automated cell counting and cluster segmentation using concavity detection and ellipse fitting techniques," in *Proc. IEEE Int. Symp. Biomedical Imaging: From Nano to Macro*, 2009, pp. 795–798.

[30] Z. Z. Han, Y. Z. Li, J. P. Zhang, and Y. G. Zhao, "Counting ear rows in maize using image process method," in *Proc. Int. Conf. Information and Computing*, 2010, pp. 329–332.

[31] R. Zhang, S. Zhao, Z. Jin, N. Yang, and H. Kang, "Application of SVM in the food bacteria image recognition and count," in *Proc. Int. Cong. Image and Signal Process*., 2010, pp. 1819–1823.

[32] J. H. S. Azuela, G. G. Lugo, and R. S. Rangel, "Counting the number of blobs in an image," in *Proc. Int. Conf. Image Processing*, 2001, pp. 1086–1089.

[33] G. R. Arce, *Nonlinear Signal Processing: A Statistical Approach*. New Jersey: Wiley, 2005.

[34] A. Flores and J. L. P. Crisóstomo, "Peruvian scallop larvae counting system using image processing techniques," in *Proc. Int. Caribbean Conference on Devices, Circuits and Systems*, 2008.

[35] M. J. Su, Z. B. Wang, H. J. Zhang, and Y. D. Ma, "A new method for blood cell image segmentation and counting based on PCNN and autowave," in *Proc. Int. Symp. Commun., Control and Signal Process.*, 2008, pp. 6–9.

[36] J. Dahle, M. Kakar, H. B. Steen, and O. Kaalhus, "Automated counting of mammalian cell colonies by means of a flat bed scanner and image processing," *Cytometry A*, vol. 60, pp. 182–188, 2004.

[37] C. Nudol, "Automatic jewel counting using template matching," in Proc. *Int. Symp. Comm. and Inf. Tech*., 2004, pp. 654–658.

[38] B. Prasad and W. Badawy, "High throughput algorithm for leukemia cell population statistics on a hemocytometer," in *Proc. IEEE Biomedical Circuits and Systems Conf.*, 2007, pp. 27–30.

[39] H. Ates and O. N. Gerek, "An image-processing based automated bacteria colony counter," in *Proc. Int. Symp. Comp. Inf. Sciences*, 2009, pp. 18–23.

[40] L. Vibha, P. D. Shenoy, K. R. Venugopal, and L. M. Patnaik, "Robust technique for segmentation and counting of trees from remotely sensed data," in *Proc. IEEE Int. Advance Computing Conf.*, 2009, pp. 1437–1442.

[41] W.-Z. Shen, J. Zhao, Y.-C. Wu, and H. Zheng, "Experimental study for automatic colony counting system based on image processing," in *Proc. Int. Conf. Computer Application and System Modeling*, 2010, pp. 612–615.

[42] J. Marotz, C. Lübbert, and W. Eisenbeiß, "Effective object recognition for automated counting of colonies in Petri dishes (automated colony counting)," *Computer Methods and Programs in Biomedicine*, vol. 66, pp. 183–198, 2001.

[43] C. Zhang, W. B. Chen, W. L. Liu, and C. B. Chen, "An automated bacterial colony counting system," in *Proc. IEEE Int. Conf. Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2008, pp. 233–240.

[44] H. Kumagai, S. Morishita, K. Kuniaki, H. Asama, and T. Mishima, "Accuracy improvement of counting asbestos in particles using a noise redacted background subtraction," in *Proc. IEEE Int. Conf. Multisensor Fusion and Integration for Intelligent Systems*, 2008, pp. 74–79.

[45] X. Zhou, X. Xiao, and C. Ma, "A study of automatic recognition and counting system of urine-sediment visual components," in *Proc. Int. Conf. Biomedical Engineering and Informatics*, 2010, pp. 78–81.

[46] H. Netten, L. J. Vliet, H. Vrolijk, W. C. R. Sloosz, H. J. Tankez, and I. T. Youngy, "Fluorescent dot counting in interphase cell nuclei," *Bioimaging*, vol. 4, pp. 93–106, 1996.

[47] Y. Chen, K. Biddell, S. Aiying, P. Relue, and J. Johnson, "An automatic cell counting method for optical images," in *Proc. Joint BMES/EMBS Conf*., 1999, p. 819.

[48] Y. Deng, Y. Wang, and P. Chen, "Automated detection of polycystic ovary syndrome from ultrasound images," in *Proc. Annual Int. Conf. IEEE Engineering in Medicine and Biology Society*, 2008, pp. 4772–4775.

[49] A. A. Disperati, J. R. Santos, P. C. Oliveira Filho, and T. Neeff, "Aplicação da técnica "filtragem de locais máximas" em fotografia aérea digital para a contagem de copas em reflorestamento de *Pinus elliottii*," *Scientia Forestalis*, vol. 76, pp. 45–55, 2007.

[50] P. H. Wu and C. H. Kuo, "A counting algorithm and application of image-based printed circuit boards," *Tamkang Journal of Science and Engineering*, vol. 12, pp. 471–479, 2009.

[51] D. Zhang, Z. Xie, and C. Wang, "Bar section image enhancement and positioning method in on-line steel bar counting and automatic separating system," in *Proc. Congress on Image and Signal Processing*, 2008, pp. 319–323.

[52] Z. Karel, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*. Academic Press Professional, 1994.

[53] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.

[54] C. G. Loukas, G. D. Wilson, B. Vojnovic, and A. Linney, "An image analysis-based approach for automated counting of cancer cell nuclei in tissue sections," *Cytometry A*, vol. 55, pp. 30–42, 2003.

[55] A. Benali, I. Leefken, U. T. Eysel, and E. Weiler, "A computerized image analysis system for quantitative analysis of cells in histological brain sections," *Journal of Neuroscience Methods*, vol. 125, pp. 33–43, 2003.

[56] T. K. Lee, M. S. Atkins, M. A. King, S. Lau, and D. I. McLean, "Counting moles automatically from back images," *IEEE Transactions on Biomedical Engineering*, vol. 52, pp. 1966–1969, 2005.

[57] C. M. Costa and S. Yang, "Counting pollen grains using readily available, free image processing and analysis software," *Annals of Botany*, vol. 104, pp. 1005–1010, 2009.

[58] P. Mussio, M. Pietrogrande, P. Bottoni, M. Dell'Oca, E. Arosio, E. Sartirana, M. Finanzon, and N. Dioguardi, "Automatic cell count in digital images of liver tissue sections," in *Proc. IEEE Symp., Computer-Based Medical Systems*, 1991, pp. 153–160.

[59] A. Sokkarie and J. Osborne, "Object counting and sizing," in *Proc. IEEE Southeastcon*, 1994, pp. 380–382.

[60] G. Gusmão, S. C. S. Machado, and M. A. B. Rodrigues, "A new algorithm for segmenting and counting *Aedes aegypti* eggs in ovitraps," in *Proc. Annual Int. Conf. IEEE Engineering in Medicine and Biology Society*, 2009, pp. 6714–6717.

[61] N. T. Nguyen, A. D. Duong, and H. Q. Vu, "A new method for splitting clumped cells in red blood images," in *Proc. Int. Conf. Knowledge and Systems Eng.*, 2010, pp. 3–8.

[62] Y. Ren, P. Wen, S. Li, Y. Liang, and W. Huang, "An improved algorithm of rat sperm image segmentation and counting," in *Proc. Int. Conf. Intelligent Computing and Integrated Systems*, 2010, pp. 201–204.

[63] J. Schier and B. Kovar, "Automated counting of yeast colonies using the fast radial transform algorithm," in *Proc. Int. Conf. Bioinformatics Models, Methods and Algorithms*, 2011, pp. 22–27.

[64] P. V. C. Hough, "A method and means for recognizing complex patterns," Patent US 3,069,654, 1962.

[65] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Patter Recognition*, vol. 13, pp. 111–122, 1981.

[66] J. Princen, J. Illingworth, and J. Kittler, "A formal definition of the Hough transform: Properties and relationships," *Journal of Mathematical Imaging and Vision*, vol. 1, pp. 153–168, 1992.

[67] R. B. M. Sonka, V. Hlavac, *Image Processing: Analysis and Machine Vision*, 3rd ed. Thomson, 2008.

[68] T. Q. Syed, V. Vigneron, S. Lelandais, G. Barlovatz-Meimon, M. Malo, C. Charriere-Bertrand, and C. Montagne, "Detection and counting of in vivo cells to predict cell migratory potential," in *Proc. Workshop on Image Processing Theory, Tools & Applic.*, 2008.

[69] W. Xiong, S. H. Ong, C. Kang, J. H. Lim, J. Liu, D. Racoceanu, and K. Foong, "Cell clumping quantification and automatic area classification in peripheral blood smear images," in *Proc. Int. Conf. Biomedical Eng. Inform.*, 2009.

[70] M. Smereka and I. Duleba, "Circular object detection using a modified Hough transform," *International Journal of Applied Mathematics and Computer Science*, vol. 18, pp. 85–91, 2008.

[71] L. H. Zou, J. Chen, J. Zhang, and N. García, "Malaria cell counting diagnosis within large field of view," in *Proc. Int. Conf. Digital Image Computing: Techniques and Applications*, 2010, pp. 172–177.

[72] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," Cornell Computing and Information Science, Tech. Rep., 2004.

[73] H. Men, Y. Wu, X. Li, Z. Kou, and S. Yang, "Counting method of heterotrophic bacteria based on image processing," in *Proc. IEEE Conf. Cybernetics and Intelligent Systems*, 2008, pp. 1238–1241.

[74] D. P. Mukherjee, A. Pal, S. E. Sarma, and D. D. Majumder, "Bacterial colony counting using distance transform," *International Journal of Bio-Medical Computing*, vol. 38, pp. 131–140, 1995.

[75] D. H. Johnson, C. R. Johnson, A. G. Klein, W. A. Sethares, H. Lee, and E. Hendriks, "A thread counting algorithm for art forensics," in *Proc. IEEE Digital Signal Processing Workshop*, 2009, pp. 679–684.

[76] A. J. Bernal, S. E. Ferrando, and L. J. Bernal, "Cell recognition using wavelet templates," in *Proc. Canadian Conf. on Electrical and Computer Eng.*, 2008, pp. 1219–1222.

[77] W. Gao, Y. Tang, and X. Li, "Segmentation of microscopic images for counting leukocytes," in *Proc. Int. Conf. Bioinform. Biomedical Engin.*, 2008, pp. 2609–2612.

[78] Z. Tai and S. Cheng, "Automatic fluorescent dot counting in interphase cell nuclei using basis pursuit image analysis," in *Proc. IEEE Region 5 Conf.*, 2008.

[79] J. Ma and G. Plonka, "The curvelet transform: A review of recent applications," *IEEE Signal Processing Magazine*, vol. 27, pp. 118–133, 2010.

[80] E. J. Candès and D. L. Donoho, "Curvelets: a surprisingly effective nonadaptive representation of objects with edges," in *Curve and Surface Fitting: Saint-Malo*, 2000.

[81] G. Loy and A. Zelinsky, "Fast radial symmetry for detecting points of interest," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 959–973, 2003.

[82] D. Reisfeld, H. Wolfson, and Y. Yeshurun, "Context free attentional operators: the generalized symmetry transform," *International Journal of Computer Vision*, vol. 14, pp. 119–130, 1995.

[83] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Prentice Hall, 2007.

[84] J. F. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.

[85] J. González, C. Galindo, V. Arevalo, and G. Ambrosio, "Applying image analysis and probabilistic techniques for counting olive trees in high-resolution satellite images," *Lecture Notes on Artificial Intelligence*, vol. 4678, pp. 920–931, 2007.

[86] A. M. T. Nasution and E. K. Suryaningtyas, "Comparison of red blood cells counting using two algorithms: Connected component labeling and backprojection of artificial neural network," in *Proc. Photonics Global Conf.*, 2008.

[87] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1988.

[88] J. Theerapattanakul, J. Plodpai, and C. Pintavirooj, "An efficient method for segmentation step of automated white blood cell classifications," in *Proc. TENCON*, 2004, pp. 191–194.

[89] J. I. Miranda, J. Camargo Neto, V. H. Oliveira, and E. D. Assad, "GeoCaju – um software para contagem não supervisionada de pés de caju," in *Anais XIV Simpósio Brasileiro de Sensoriamento Remoto*, 2009, pp. 6967–6973.

[90] G. Gilboa, N. Soche, and Y. Y. Zeevi, "Image enhancement and denoising by complex diffusion processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1020–1036, 2004.

[91] J. E. Goin, D. R. Kline, and M. J. Hippe, "Classification models for the counting of cellular objects," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 20, pp. 283–291, 1990.

[92] G. Cordiki, R. Diaz-Uribe, J. L. Folch-Mallol, and J. Nieto-Sotelo, "COVASIAM: an image analysis method that allows detection of confluent microbial colonies and colonies of various sizes for automated counting," *Applied and Environmental Microbiology*, vol. 64, pp. 1400–1404, 1998.

[93] J. Osowsky and H. R. Gamba, "Sistema automático para contagem de colônias em placas de Petri," *Revista Brasileira de Engenharia Biomédica*, vol. 17, pp. 131–139, 2001.

[94] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, 1979.

[95] J. Selinummi, J. Seppälä, O. Yli-Harja, and J. Puhakka, "Software for quantification of labeled bacteria from digital microscope images by automated image analysis," *BioTechniques*, vol. 39, pp. 859–862, 2005.

[96] B. Fang, W. Hsu, and M. L. Lee, "On the accurate counting of tumor cells," *IEEE Trans. on Nanobioscience*, vol. 2, pp. 94–103, 2003.

[97] K. Mitev, Y. Madzhunkov, G. Gerganov, I. Dimitrova, S. Georgiev, and D. Pressyanov, "Automatic counting of electrochemically etched tracks in compact discs. Application to retrospective measurements of Rn-222," *IEEE Trans. Nuclear Science*, vol. 57, pp. 300–308, 2010.

[98] A. C. Lucarini, L. A. Silva, and R. A. C. Bianchi, "Um sistema para contagem semi-automática de microorganismos," Pesquisa e Tecnologia, vol. 26, pp. 36–40, 2004.

[99] M. Nixon and A. S. Aguado, *Feature Extraction & Image Processing for Computer Vision*, 2nd ed., Academic Press, 2008.

[100] F. B. Tek, A. G. Dempster, and I. Kale, "Parasite detection and identification for automated thin blood film malaria diagnosis," *Computer Vision and Image Understanding*, vol. 114, pp. 21–32, 2010.

[101] A. Bechar, S. Gan-Mor, Y. Vaknin, I. Shmulevich, B. Ronen, and D. Eisikowitch, "An image-analysis technique for accurate counting of pollen on stigmas," *New Phytologist*, vol. 137, pp. 639–643, 1997.

[102] C. Pornpanomchai, F. Stheitsthienchai, and S. Rattanachuen, "Object detection and counting system," in *Proc. Congress on Image and Signal Processing*, 2008, pp. 61–65.

[103] W. B. Chen and C. Zhang, "Bacteria colony enumeration and classification for clonogenic assay," in *Proc. IEEE Int. Symp. Multimedia*, 2008, pp. 487–488.

[104] M. R. B. Forshaw and C. R. Wiles, "Counting volcanoes on Venus: comparison between humans, computers and theory," in *Proc. Int. Conf. Image Processing and its Applications*, 1995, pp. 65–69.

[105] S. Hinz, "Detection and counting of cars in aerial images," in *Proc. Int. Conf. Image Processing*, 2003, pp. 997–1000.

[106] T. Shimada, K. Kato, A. Kamikouchi, and K. Ito, "Analysis of the distribution of the brain cells of the fruit fly by an automatic cell counting algorithm," *Physica A*, vol. 350, pp. 144–149, 2005.

[107] K. Ishizu, H. Takemura, K. Kawabata, H. Asama, T. Mishima, and H. Mizoguchi, "Image processing of particle detection for asbestos qualitative analysis support method – particle counting system based on classification of background area," in *Proc. Int. Conf. Control, Automation, Robotics and Vision*, 2008, pp. 868–873.

[108] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Prentice Hall, 2008.

[109] E. Comunello, A. Wangenheim, D. Krechel, and K. Faber, "CT scans with neurocysticercosis in epileptics patients: a computer-based method for detection and quantification," in *Proc. IEEE Symp. Computer-Based Medical Systems*, 1999, pp. 42–47.

[110] L. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Transactions on Circuits and Systems*, vol. 35, pp. 1257–1272, 1988.

[111] G. Seiler, "Small object counting with cellular neural networks," in *Proc. IEEE Int. Workshop on Cellular Neural Networks and their Applications*, 1990, pp. 114–123.

[112] Q. Feng, S. Yu, and H. Wang, "An new automatic nucleated cell counting method with improved cellular neural networks (ICNN)," in *Proc. Int. Workshop on Cellular Neural Networks and Their Applic*., 2006.

[113] A. Fasih, J. Chedjou, and K. Kyamakya, "Ultra fast object counting based-on cellular neural network," in *Proc. Int. Workshop on Nonlinear Dynamics and Synchronization*, 2008, pp. 181–183.

[114] S. Sevgen, F. Karabiber, E. Yucel, and S. Arik, "Implementation of a CNN based object counting algorithm on bi-i cellular vision system," in *Proc. Int. Conf. Electrical and Electronics Engineering*, 2009, pp. 394–397.

[115] P. C. Schaich, G. A. Clark, S. K. Sengupta, and K. P. Ziock, "Automatic image analysis for detecting and quantifying gamma-ray sources in coded-aperture images," *IEEE Trans. on Nuclear Science*, vol. 43, pp. 2419–2426, 1996.
[116] E. Christophe and J. Inglada, "Object counting in high resolution remote sensing images with OTB," in *Proc. IEEE Int. Geoscience and Remote Sensing Symp*., 2009, pp. 737–740.
[117] F. Cointault and P. Gouton, "Texture or color analysis in agronomic images for wheat ear counting," in *Proc. Int. IEEE Conf. Signal-Image Technologies and Internet-Based System*, 2007, pp. 696–701.
[118] F. Schönholzer, D. Hahn, B. Zarda, and J. Zeyer, "Automated image analysis and in situ hybridization as tools to study bacterial populations in food resources, gut and cast of Lumbricus terrestris L." *Journal of Microbiological Methods*, vol. 48, pp. 53–68, 2002.
[119] E. Montseny, P. Sobrevilla, and S. Romani, "A fuzzy approach to white blood cells segmentation in color bone marrow images," in *Proc. FUZZ-IEEE*, 2004, pp. 173–178.
[120] L. Li and D. Qi, "Detection of cracks in computer tomography images of logs based on fractal dimension," in *Proc. IEEE Int. Conf. Automation and Logistics*, 2007, pp. 2259–2264.
[121] J. Selinummi, J. R. Sarkanen, A. Niemistö, M. L. Linne, T. Ylikomi, O. Yli-Harja, and T. O. Jalonen, "Quantification of vesicles in differentiating human SH-SY5Y neuroblastoma cells by automated image analysis," *Neuroscience Letters*, vol. 396, pp. 102–1076, 2006.
[122] N. Sun, S. Xu, M. Cao, and J. Li, "Segmenting and counting of wallpasted cells based on Gabor filter," in *Proc. IEEE Engineering in Medicine and Biology*, 2005, pp. 3324–3327.
[123] Y. Lu and S. Qin, "Stored-grain insect image processing based on a Hidden Markov Model," in *Proc. Int. Conf. Electrical and Control Engineering*, 2010, pp. 1997–2000.
[124] G. A. Ruz, P. A. Estévez, and C. A. Perez, "A neurofuzzy color image segmentation method for wood surface defect detection," *Forest Products Journal*, vol. 55, pp. 52–58, 2005.
[125] T. K. O'Donnell, K. W. Goyne, R. J. Miles, C. Baffaut, S. H. Anderson, and K. A. Sudduth, "Identification and quantification of soil redoximorphic features by digital image processing," *Geoderma*, vol. 157, pp. 86–96, 2010.
[126] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. on Computer Vision*, 1999, pp. 1150–1157.
[127] H. Bay, A. Ess, and L. V. G. T. Tuytelaars, "SURF: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, 2008.
[128] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. Conf. of NIPS*, 2010.
[129] V. V. Vinod and H. Murase, "Counting multi-colored objects using active search," in *Proc. Int. Conf. Information, Communications and Signal Processing*, 1997, pp. 185–189.
[130] E. F. Morais, M. F. M. Campos, F. L. C. Pádua, and R. L. Carceroni, "Particle filter-based predictive tracking for robust fish counting," in *Proc. Braz. Symp. Computer Graphics and Image Process*., 2005, pp. 367–374.
[131] L. Tsankov, D. Pressyanov, K. Mitev, S. Georgiev, and I. Dimitrova, "Automatic counting of chemically etched tracks by means of a computer scanner," *Radiation Measurements*, vol. 39, pp. 557–559, 2005.
[132] S. Subramanian and L. V. Bhadrinarayana, "A memory efficient algorithm for real time object counting," in *Proc. IEEE Int. Advanced Computing Conf.*, 2009, pp. 245–248.