Proceedings of the IASTED International Conference
Signal Processing, Pattern Recognition, and Applications (SPPRA 2011)
February 16 - 18, 2011 Innsbruck, Austria

A SEMI-AUTOMATIC SOFTWARE TOOL FOR BATCH PROCESSING OF YEAST COLONY IMAGES

Jan Schier and Bohumil Kovář
Institute of Information Theory and Automation of the ASCR
Dept. of Signal Processing
Pod vodárenskou věží 4
Prague, Czech Republic
email: {schier,kovar}@utia.cas.cz

ABSTRACT
A software application for batch counting of yeast colonies in images of Petri dishes, based on the fast radial transform by Loy and Zelinsky, is introduced in the paper. The motivation to develop our own tool, instead of using a commercial solution, was mainly the requirement to reuse the imaging equipment already existing in the cooperating genetics laboratory, as well as the possibility to adapt the application to the needs of the laboratory. The basic aim is to process image sets prepared beforehand in the darkroom and to automate the process of counting as much as possible. The tool operates either in a fully automatic mode, or provides an interactive editor for correction of the initial machine counting result.

The underlying processing flow and algorithms are reviewed in the paper and the performance of the method is evaluated, using a test set of 245 images. The application is available for download or can be requested from the authors of the paper, in both cases free of charge.

KEY WORDS
yeast colonies, Petri dish, image segmentation, fast radial transform.

1 Introduction

Saccharomyces cerevisiae yeast is often used as a model organism in molecular and cell biology. This is mainly due to its favourable properties, such as the short generation time and easy genetic manipulation. When performing experiments using the *S. Cerevisiae* yeast, the process often includes – for colonies growing in Petri dish – evaluating the growth parameters, such as the coverage of the dish and the number of colonies. Traditionally, the colonies have been counted manually, using either manual colony counter or a counting raster. However, the hand counting is considered rather laborious and time-consuming process, error prone due to the fatigue and eye-strain of the technician. Also, it does not provide any figures on additional parameters such as the coverage of the dish or the radii of the colonies.

Today, there is a number of automated colony counters available on the market (it is of interest that an automated counter has been described as early as in 1974 [1]). Such system usually consists of an illuminated chamber with a camera system, where the dishes are placed either manually or by an automated plate-handler, and of a computer running an image-processing program that performs evaluation of the image. Such device is definitely the system-of-choice for a commercial lab performing an established set of analyses.

The motivation for our work was the need of the cooperating lab to reuse the existing general-purpose imaging system. With this in mind, we focused purely on the image processing tasks, that is, we did not do any changes to the imaging setup. Our aim was to optimize the functionality and performance of the software tool to the setup used in the laboratory, to achieve an adequate performance for typical image variations, as described in the next section, with minimum possible manual intervention of operator (this includes operations such as determination of the dish position, threshold settings, etc.). In the same time, it should allow for manual correction of the counting errors (multiple detections of a single colony or missed colonies). Finally, it should provide not only the count of colonies, but also evaluate the relative coverage of the dish.

In this paper, we focus on the system description, together with the underlying principles and algorithms and present the performance results that we have achieved. The paper is structured in the following way: in the next section, the characteristics of typical images are reviewed and the system is described. Then, the methods for detecting round or symmetrical objects are surveyed in brief and the fast radial transform algorithm is described. Finally, the properties of the test data set are described and the achieved results are discussed.

2 Image characteristics

An input for the tool is a set of images of Petri dishes, acquired by digital camera. Each snapshot contains an image of the dish with yeast colonies (Figure 1). Basic characteristics such as relative dish coverage and the number of colonies in each image have to be determined for further evaluation of the experiment.

The images are acquired in a dark room using simple imaging system which consists of a digital camera and two

DOI: 10.2316/P.2011.721-053

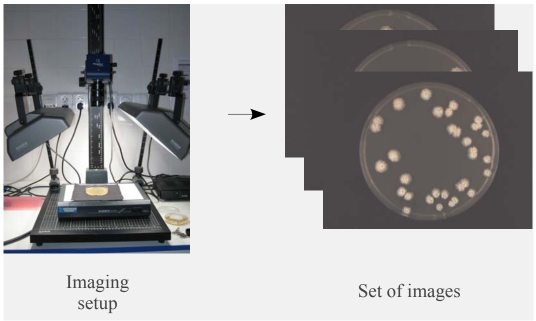


Figure 1: Image capture setup

light sources mounted on a stand. The dishes are manually placed roughly into the center of the image, on a mat black background. No dish holder or stopper is used.

The images are mainly characterized by the following properties:

+ typical resolution of an image is 1024×768 pixels,
+ dark background is used to increase image contrast,
+ illumination by two linear light sources along the short edges,
+ colonies are roughly round-shaped,
– variations of dish position,
– varying diameter of the dish in image,
– varying intensity of the dish background.
– colonies often touch each other and form clusters.
– the morphology of the colonies differs over different images (the colonies can be both smooth and fluffy).
– differing size of colonies, even in a single image (depends on growth conditions and age)

For illustration of possible morphology differences, see Figure 2.

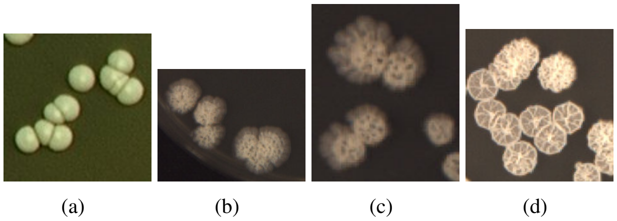


Figure 2: Examples of different morphologies of yeast colonies

3 System description

3.1 General requirements

The proposed system does not include the image acquisition part – it consists of a software application that processes an already acquired set of images.

As already stated in the introduction, it was required that the need for intervention of human operator was minized, so that the system would be able to operate in the batch mode, with an acceptable error rate. In the same time, it should allow for manual correction of the counting errors, if needed. Such correction further improves the accuracy of output, while the operator load is still much decreased compared to the counting all colonies in the dish.

3.2 Processing flow

Figure 3 shows the general processing flow: for each image of the set, the background is first eliminated. The exact location of the dish in the image is determined and segmentation is performed for the image area inside the dish. The output of this segmentation represents yeast colonies, or (in the case of touching colonies) clusters of colonies.

The goal of the next step is to resolve individual colonies in the clusters. This is currently performed by recognition of round objects in the image: a map of possible centers is computed, and the centers are determined as local maxima of this map. Finally, the statistics (coverage and number of colonies) for each dish are computed.

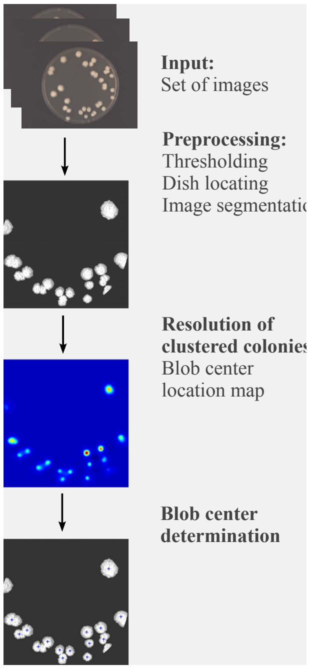


Figure 3: Image processing flow

In the following sections, each step will be described briefly.

3.3 Image preprocessing

The main objective of image preprocessing is to suppress the image background and to identify the exact location of the dish in the image. Correctness of the dish position is also checked for, so that significantly misplaced samples would not affect the results (the inner part of the dish must be completely contained in the image). Then, second threshold is used to separate the colonies from the dish background.

Let us introduce, for further description, the *dish background* as the background *inside* the dish and the *image background* as the background *outside* the dish.

3.3.1 Image background thresholding and dish location

Image background thresholding is used to reduce the background noise level in the image. To determine the threshold level, four square samples are taken in the corners of the image (see Figure 4) and for each sample, mean level and standard deviation is computed. The sample with minimum mean level is used as the reference, to eliminate eventual background distortions (such as that contained in the included example image). The threshold is set to the mean level of this sample plus three times its deviation.

The image is binarized using this threshold, and projections to horizontal and vertical axis are computed (Figure 5) in order to determine the dish location. The location is determined as the "broadest" peak of the projection: each projection is binarized at 20% of maximum level (green line in Figure 5) and the longest interval between the state change of the binary signal is considered. The resulting estimate of the dish diameter in both horizontal and vertical direction projection is compared with the image size to eliminate images containing severe background disruptions.

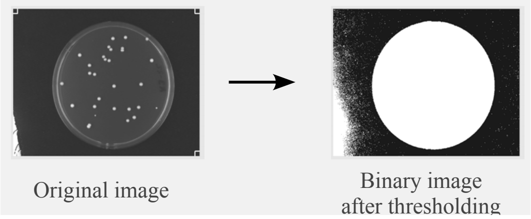


Figure 4: Background thresholding and image binarization

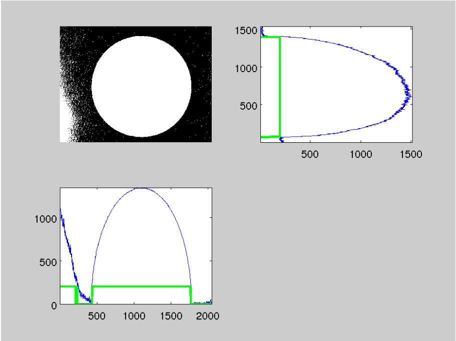


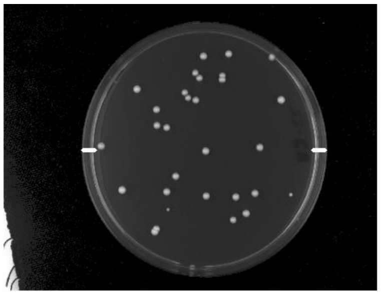
Figure 5: Projections of binary image

For further processing, only the circular region in the place of the located dish is used. Hence, any noise located in the image background, outside the dish area, does not affect the processing any more.

3.3.2 Dish segmentation

To perform dish segmentation, thresholding with global threshold is used. The image background has been eliminated in the process outlined in Section 3.3.1. Since the dish background is somewhat brighter (the dish has higher reflectivity than the image background), new threshold has to be used to eliminate it. The setting of this threshold is based on the following property of the images under our consideration: the dish background is darker than the dish rim and the colonies (and the rim self is usually darker than the colonies). Hence, the threshold level should be set somewhere between the dish background and the bright spots of the dish rim.

To determine the dish background threshold, image samples taken in the position of dish rim are used. These samples are taken along the image row passing through the dish center (see Figure 6, the samples are marked with bold white lines).



Position of samples marked with white lines

Figure 6: Position of rim samples for dish segmentation

The length of the samples (that is, the length of the rim projection) is given by the dimensions of the Petri dish and by the parameters of the imaging system (the focal length of the lens). As long as these parameters are constant, it depends only on the image size. The current version of the tool is designed for 90mm dishes, the rim width is set as fixed fraction of the image width.

The calculation of the threshold value is illustrated in Figure 7 on the next page, which shows the brightness along the central row of the dish. The bold parts of the graph represent the dish rim, which is also denoted by the vertical dashed lines. The horizontal dashed line denotes the minimum brightness of the dish background along this row, and the solid line shows the resulting position of the image background.

Let us denote the dish background vector (still considering the central dish image row, as shown in Figure 7) as $\mathbf{d}$ and both parts of the rim as $\mathbf{r}_1$ and $\mathbf{r}_2$. Let $d_{\mathtt{min}} = \min(\mathbf{d})$

be the minimum value of the dish background. We determine the dish threshold T_d as

$$T_d = \mathtt{mean}([\mathbf{r}_1; \mathbf{r}_2] - d_{\mathtt{min}}) + d_{\mathtt{min}}$$

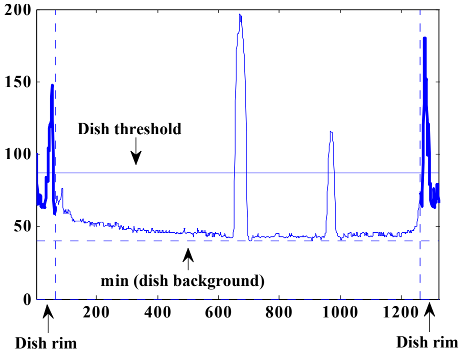


Figure 7: Dish threshold setting

3.4 Recognition of round object

A well-established method for recognition of circular objects is the Hough transform. Algorithm based on this transformation has been proposed for monitoring of yeast cell size distribution in a fermentor [2]. However, judging from the results of our experiments, and following the results presented in [3], the output of the Hough transformation may be rather noisy for the real-life images.

An interesting, rather sophisticated approach for counting bacteria colonies has been described in [4]: to find colonies, a number of shape and structure descriptors are used on the image pixels, including e.g. mean object radius, roundness of an object, average uniformity of radii in different directions, etc. These criteria are then combined into shape and structure quality parameters and evaluated using fuzzy logic. This solution, however, is based on rather uniform morphology of bacteria colonies, which is not satisfied in the case of experiments with yeast colonies.

In the recent years, methods based on radial symmetry have become very popular. The radial symmetry transform has been introduced in the seminal work of Reisfeld et all. [5]. Fast radial transform (a modified version of the radial symmetry transform with improved computational complexity) has been presented by Loy and Zelinsky [3]. Since we have used this method in our tool, we shall review it in more detail in the next section.

3.5 Fast Radial Transform

Fast Radial Transform is a robust method for the detection of symmetrical objects. This transform maps the original image to the transformed image according to its contribution to radial symmetry of the gradients a distance $n \in N$ (N is the set of radii) away from each point.

First, image gradient $\mathbf{g_{i,j}}$ at each point (i, j) is calculated using Sobel edge detector. Then, the *positively-* and *negatively-affected pixels* are determined. The *affected pixel* is defined as the point that the gradient vector $g_{i,j}$ is pointing to (for the positively-affected ones) or from (for the negatively-affected ones), a distance n away from the point at coordinates (i, j). The coordinates of affected pixels are given by

$$\begin{aligned} \mathbf{p}_+(i,j) &= (i,j) + \text{round}\left(\frac{\mathbf{g}_{i,j}}{\|\mathbf{g}_{i,j}\|} n\right) \\ \mathbf{p}_-(i,j) &= (i,j) - \text{round}\left(\frac{\mathbf{g}_{i,j}}{\|\mathbf{g}_{i,j}\|} n\right) \end{aligned}$$

The gradient matrix $\mathbf{g}$, together with the coordinates of affected pixels, are used to determine the *orientation* and *magnitude* projection images O_n and M_n for the given radius n:

$$\begin{aligned} O_n(\mathbf{p}_+(i,j)) &= O_n(\mathbf{p}_+(i,j)) + 1, \\ O_n(\mathbf{p}_-(i,j)) &= O_n(\mathbf{p}_-(i,j)) - 1, \end{aligned}$$

$$\begin{aligned} M_n(\mathbf{p}_+(i,j)) &= M_n(\mathbf{p}_+(i,j)) + \|\mathbf{g}_{i,j}\|, \\ M_n(\mathbf{p}_-(i,j)) &= M_n(\mathbf{p}_-(i,j)) - \|\mathbf{g}_{i,j}\|. \end{aligned}$$

The radial symmetry at radius n is defined by convolution

$$S_n = F_n * A_n,$$

where

$$F_n(i,j) = \frac{M_n(i,j)}{k_n}\left(\frac{|\tilde{O}_n(i,j)|}{k_n}\right)^{\alpha},$$

and

$$\tilde{O}_n(i,j) = \begin{cases} O_n(i,j) & if O_n(i,j) < k_n \\ k_n & \text{otherwise.} \end{cases}$$

A_n is a two-dimensional Gaussian, α is the radial strictness parameter and k_n is a scaling factor used to normalize M_n and O_n. Projection images M_n and O_n are initially set to zero.

To estimate the colony centers, the transform must be complemented by two additional procedures: by estimation of the set of radii N to be tested, and by a procedure to find local maxima (corresponding to the centers) in the symmetry map S (map S is a weighted sum of all symmetry maps S_n). To estimate the range of radii of the objects contained in the image, the following steps are used:

- the equivalent diameter d and eccentricity ϵ of all objects in the image is computed. The diameter is computed from area A (number of pixels) of an object. The eccentricity is evaluated as the eccentricity of an ellipse with the same second-moments as the object. It equals to $\epsilon = 0$ for circle and to $\epsilon = 1$ for line segment.

- Circular objects (with eccentricity $\epsilon < T_\epsilon$, T_ϵ is the eccentricity threshold, are selected.

- Min and max diameter $d_{\min}$ and $d_{\max}$ of the objects in the set of circular objects are determined. The interval between them is divided to ν equidistant subintervals. The set of radii N for the fast radial transform is then given by:

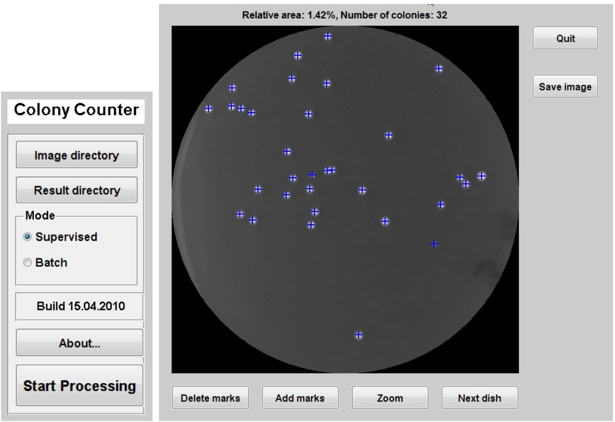
$$\begin{aligned} \Delta &= d_{\max} - d_{\min} \\ N &= \{d_{\min} + [0, \ldots, \nu] \cdot \Delta/\nu\}/2 \end{aligned}$$

The centers of the colonies are represented by local maxima of the symmetry matrix S. To find these maxima, the `nonmaxsuppts()` function by Kovesi [6] is used. It performs grey scale dilation of the input image and finds the points in the dilated image that match the original and that are greater than the threshold. Finally, it returns the row and column coordinates of these points.

3.6 ColonyCounter tool

The tool for semi-automatic counting of yeast colonies has been developed in Matlab 2008b with use of the Image Processing Toolbox v6.2.

The initial window of the tool is shown in Figure 8(a). It can operate in two modes: the Batch mode is for fully automatic operation (but without possibility to correct detection errors), the Supervised mode opens the graphical editor after each image has been processed (Figure 8(b)). The operator has possibility to manually correct any detection errors. The Zoom function is also available, for inspection of the parts of image in which the yeast colonies produce dense clusters. The results of counting are saved to a CSV file for further processing.



Mode
● Supervised
○ Batch

Relative area: 1.42%, Number of colonies: 32

(a) (b)

Figure 8: Colony Counter

4 Experiments and Results

4.1 Structure of test images

The system recognition performance was tested using 245 images, containing colonies with different morphology and relative coverage of the dish. The distribution of the test set in the terms of frequencies of the number of colonies in the dish is shown in Figure 9.

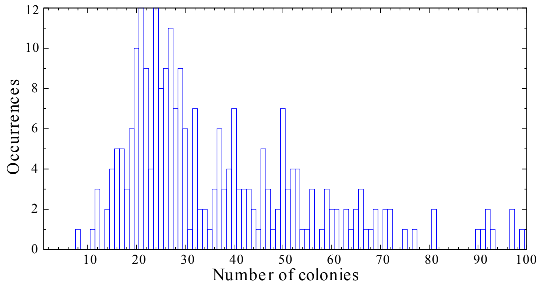


Figure 9: Petri dishes with the same number of yeast colonies

Figure 10 shows the distribution of relative coverage of the dish and mean diameter of colonies (marker size) in comparison with the number of colonies in the dish.

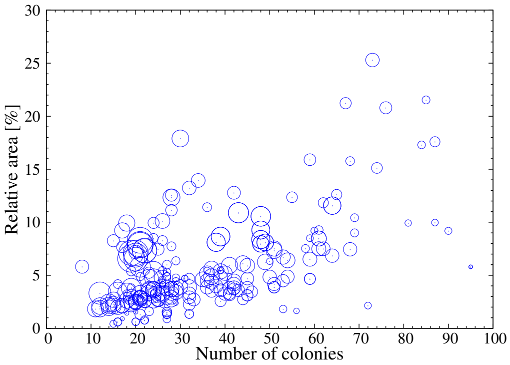


Figure 10: Distribution of the test set – relative dish coverage in comparison with the number of colonies. Diameter of markers represents mean radius of the colonies.

4.2 Algorithm settings

The following settings were used in our experiments:

- Parameters of the fast radial transform: $k_n = 6, \alpha = 4$
- Parameters of non-maxima suppression: $\text{threshold} = 4, \text{radius} = 0.8 \cdot \min(N)$, where N is the set of radii for the fast radial transform.
- Construction of N: eccentricity threshold θ_ϵ is initially set to $\theta_\epsilon = 0.25$. The number of equidistant intervals ν is set to $\nu = 4$. At least five colonies of given eccentricity must be in the image, else the eccentricity is increased by 0.1.

4.3 Detection errors

Figure 11 illustrates typical detection errors of fast radial transform. A colony may be missed if it touches other colonies so that they form a cluster. This detection error is almost absent if the colonies touch only at one point and thus creates a structure similar to a chain. If more colonies are touching each other, they form a structure in which their shape is distorted and internal colonies cease to be circular. The circular outer border of colony located in such cluster could be too short for the proposed method to work properly.

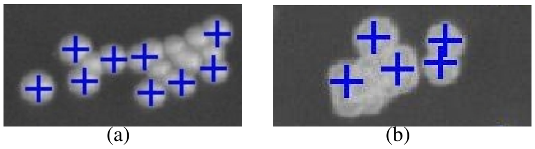

(a) (b)

Figure 11: Example of typical detection error of fast radial transform.

The counting errors for various number of colonies are summarized in the Table 1. The relatively high counting error for the dishes with the colony counts greater than 60 is given by two factors. First, with the increasing number of colonies increases also the probability that there will be colonies touching each others (Figure 11a). Second, the number of samples with this high density of coverage was relatively low, thus increasing the evaluation error (Figure 9).

♯ colonies	samples	missed [%]	false [%]
0 – 17	21	1.47	0
18 – 20	19	2.99	0
21 – 23	25	3.31	0
24 – 26	29	4.00	0
27 – 29	27	3.28	0
30 – 40	35	4.17	0
41 – 49	30	4.00	0
50 – 60	29	3.87	2.00
> 60	30	5.36	3.22

Table 1: Dependence of the counting errors on the number of colonies in the dish. The system average counting error is under 4%

Figure 12 shows the recognition performance regarding to the number of colonies in Petri dishes. A typical Petri dish used in our experiments contains from 10 to 40 yeast colonies. The figure also show the recognition performance of the fast radial transform for the detection of circular objects. The samples, where some colonies have not been detected, fall below the diagonal line.

Another view on the algorithm performance is provided by Figure 13, which contains a box-whisker plot of relative counting error (missed colonies) in several groups of the colony counts. The numbers of samples in each group are given in the table, included with this plot.

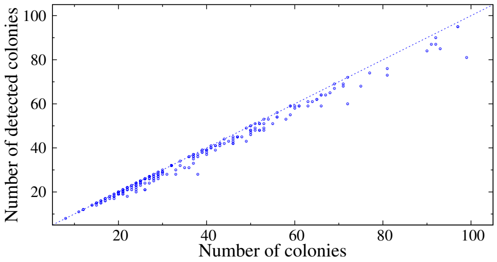
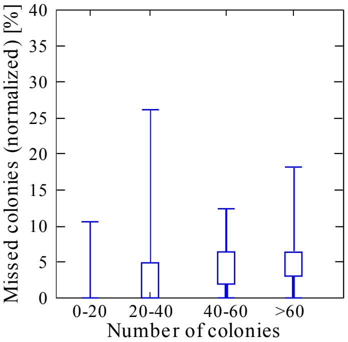


Figure 12: Algorithm recognition performance regarding to number of colonies.



# colonies	samples
0–20	40
20–40	123
40–60	52
>60	30

Figure 13: Relative number of missed colonies dependent on the number of colonies

Illustrative counting results for typical images used in these tests are shown in Figure 14.

5 Conclusion

In the paper, we have introduced a system for semi-automatic counting of yeast colonies in Petri dishes. It has been shown that the method presented here is reasonably accurate and that it is able to deal with clustered colonies.

The algorithm has been tested on a set of 245 images with different degrees of coverage and yeast morphology. The composition of the data in the test set and the experimental results were discussed. It should be noted that the average counting error is below 4%. It is difficult to compare the performance of the proposed tool with other solutions (commercial or academical), since the performance data or comparative benchmarks are not available, to our knowledge.

The algorithm has been implemented in a Matlab-based tool which provides environment for automatic or semi-automatic processing of batches of images. This tool has been successfully deployed in the cooperating biology lab and is available for download at `http://zs.utia.cz/index.php?ids=results&id=yeastcolcount&lang=eng`.

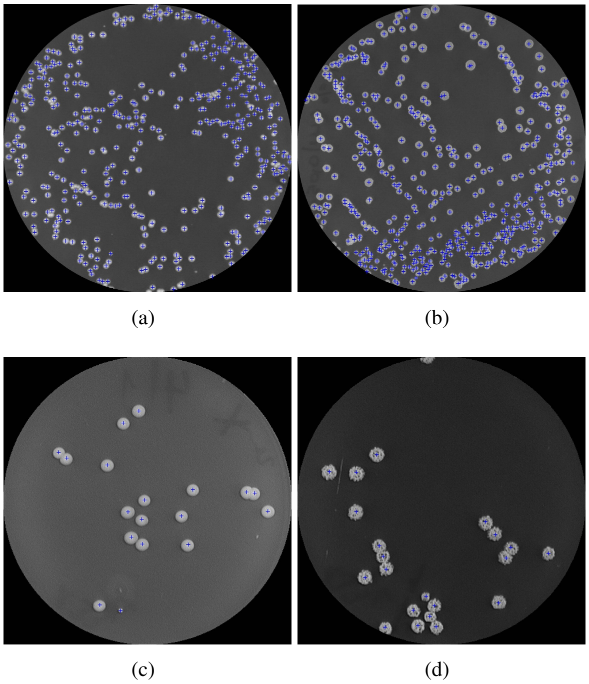


Figure 14: Examples of algorithm detection and counting results. Figures (a) - (b) illustrate dishes with high coverage, (c) - (d) dishes with low coverage.

5.1 Future work

One of the reviewers has pointed us towards the use of Boolean models to extract features of the image in the case of overlapping objects. We would like to thank him for the idea at this point. While there was not enough time to include the method in the paper, given the character of the colony images, it could be interesting to evaluate it in future.

Acknowledgements

This research has been supported from the 1M0567 “Centre for Applied Cybernetics” project of the Ministry of Education, Youth and Sports, the Czech Republic. We would like to thank the staff of the Yeast Colony Group, Department of Genetics and Microbiology, Faculty of Sciences, Charles University, Prague, who have introduced us into the problem and provided with the test images from their experiments.

References

[1] W. A. Goss, R. N. Michaud, and M. B. McGrath. Evaluation of an automated colony counter. *Appl Microbiol*, 27:264–267, Jan 1974.

[2] Yoshiyuki Yamashita, Masayoshi Kuwashima, Toshiyuki Nonaka, and Mutsumi Suzuki. On-line measurement of cell size distribution and concentration of yeast by image processing. *JOURNAL OF CHEMICAL ENGINEERING OF JAPAN*, 26(6):615–619, 1993.

[3] Gareth Loy and Alexander Zelinsky. Fast radial symmetry for detecting points of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(25):959–973, August 2003.

[4] J. Marotz, C. Lübert, and W. Eisenbeiss. Effective object recognition for automated counting of colonies in petri dishes (automated colony counting). *Computer Methods and Programs in Biomedicine*, 66(2–3):183–198, September 2001.

[5] Daniel Reisfeld, Haim Wolfson, and Yehezkel Yeshurun. Context-free attentional operators: The generalized symmetry transform. *International Journal of Computer Vision*, 14(2):119–130, March 1995.

[6] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. School of Computer Science & Software Engineering, The University of Western Australia, 2005. Available from: http://www.csse.uwa.edu.au/∼pk/research/ matlabfns/.