

Simple Tool for Semi-automated Evaluation of Yeast Colony Images

Jan Schier and Bohumil Kovář

Institute of Information Theory and Automation of the ASCR
Pod vodárenskou věží 4, CZ-182 08 Prague 8, Czech Republic
{schier,kovar}@utia.cas.cz

Abstract. A tool for semi-automated evaluation of images of yeast colonies is presented in the paper. The input data for the tool is a batch of Petri dish images, possibly in nested directories, the output is written to a text file in a simple CSV (comma-separated values) format. The tool is designed to evaluate, for each dish image, the relative coverage of the dish and the number of colonies contained in the dish. It is intended for use in a research laboratory, where a general-purpose imaging setup is used to take the images, which are then processed off-line.

In the paper, the characteristic features of the yeast colony images are described and the processing flow and the user interface of the tool are reviewed. The performance of the counting is evaluated using a test set of 245 images with typical relative coverage of less than 10%.

The tool is available for download from our web page.

1 Introduction

Yeast, namely *Saccharomyces cerevisiae*, is often used as a *model organism* in biological research. This is mainly due to its many favourable properties, such as the short generation time, easy genetical manipulation, as well as thanks to its established use in industry.

When performing experiments with *S. Cerevisiae* yeast, part of the process is growing the yeast colonies in Petri dishes, and 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. Examples of the manual counters are the Colony counter SC6 by Stuart¹ or the EW-14211-02 Hand-Held Electronic Colony Counter from Cole-Parmer². 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 the coverage of the dish or the radii of the colonies.

There are also a number of automated colony counters available – it is of interest that such system has been described as early as in 1974 [1]. An automated counter usually consists of a single-purpose chamber with an illumination and camera system, where the dishes are placed either manually one-by-one or by an automated plate-handler,

¹ www.stuart-equipment.com

² www.coleparmer.com

and of a computer running an image-processing program that performs thresholding, segmentation, and statistical evaluation of the image.

An automated counter is definitely the system-of-choice for a commercial lab. Our system, however, has been developed for a the cooperating university research lab. The requirements were not only that it performs adequately for typical image variations, but also – quite importantly – to reuse the imaging equipment already available in the lab (to achieve a low-cost solution), and to provide an off-line batch processing of a set of images taken beforehand in the darkroom. Additionally, the tool should operate with minimum possible manual intervention of operator, while allowing 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 total coverage of the dish.

Finally, the open-source framework ImageJ [2] should be noted in the context of existing tools. It is a feature-rich tool for image processing, with focus in medical and biological applications. In the field of our interest, it provides many algorithms for image thresholding and segmentation. In our view, however, it is targeted more generally and to prepare a specialized application which shields the user of the complexity of the underlying system would require an effort similar to development of our solution. We have decided to use MATLAB with the Image processing toolbox, because of our previous experience with that environment.

In the paper, we describe the processing flow of our tool and present the performance data that we have achieved. The paper is structured in the following way: in Section 2, the characteristics of typical images are reviewed. In Section 3, the components of the system are discussed. Section 5 describes the experiments and performance results. Finally, Section 6 concludes the paper.

2 Characteristics of Typical Images

As has been stated, the requirement on our system was that it should be able to process batches of images taken with a simple general-purpose imaging system; the laboratory is using a camera/copy stand with lightning units by Kaiser Fototechnik³ (see Fig. 1). The dishes are manually placed roughly into the center of the image, on a mat black background. No dish holder or stopper is used.

Examples of typical images of yeast colonies are shown in Figure 2. The images are characterized by the following properties:

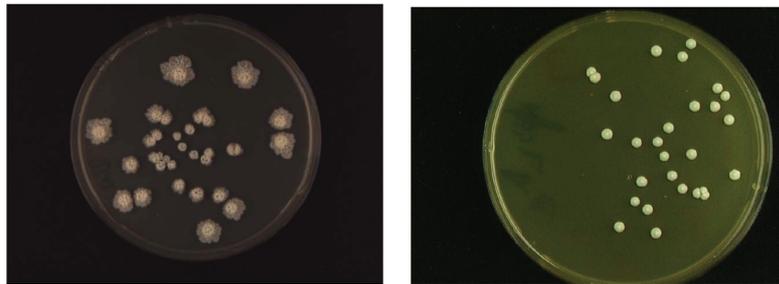
- dark background to increase image contrast
- illumination by two linear light sources along the short edges (this is not important for processing, though)
- variations of dish position – cf. Figure 2a and 2b
- varying diameter of the dish image and varying intensity of the dish background (both depend on the adjustment on the imaging setup in the particular session)

The colonies selves are characterized by differing types of morphologies (smooth or fluffy), colony size dispersion (even within single dish) and by colonies often touching

³ www.kaiser-fototechnik.de



Fig. 1. General-purpose imaging setup used with the proposed colony counter



(a) Fluffy and touching colonies, diverse radii, dish centered.

(b) Smooth colonies, dish touching edge of image.

Fig. 2. Examples of typical images of yeast colonies

each other. Let us note that the morphology and size of colonies depends both on their age and on the growth conditions.

3 System Description

Counting of colonies typically consists of several operations, which follow from the image characteristics[3,4]. This operations include image thresholding, dish localization, and the counting self. In this paper, we divide the processing into two basic parts – *image preprocessing* and *colony counting*.

3.1 Image Preprocessing

The goal of this step is to determine the exact position of the dish in the image, to determine the area of the dish, and to extract the image of colonies and measure its surface, so that the relative coverage can be determined. The image is checked for any faults in this step, such as missing parts (see Fig. 3a) or corrupted background (Fig. 3b). Let us note that while the missing part makes it impossible to determine properly the total area of the dish, the corrupted background can be eliminated to some extent.

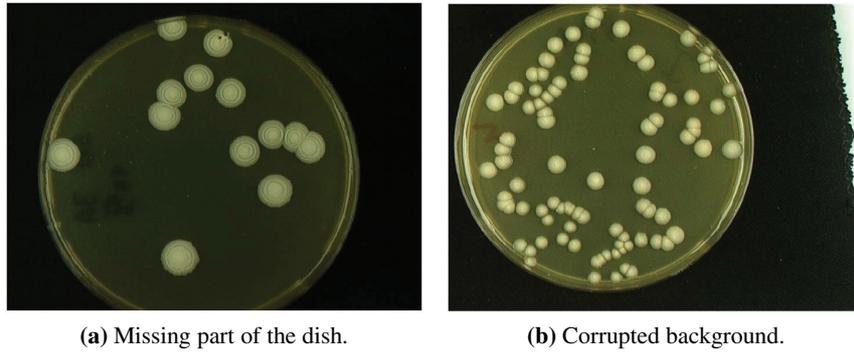


Fig. 3. Examples of faulty images

The processing flow of the preprocessing is outlined in Fig. 4 on the following page. It starts with operations related to dish localization: the image is first binarized. A threshold that is used in this process is computed using samples from the image corners. Let these samples be S_1, \dots, S_4 (see Fig. 5 on the next page), with the mean values $\bar{S}_1, \dots, \bar{S}_4$ and the standard deviations $\sigma_1, \dots, \sigma_2$. The background threshold θ_b is derived from the sample with minimum mean value:

$$i = \arg \min_i (\bar{S}_i) , \quad \theta_b = \bar{S}_i + 3\sigma_i + \text{offset}$$

Note: offset is used to ensure that the threshold will be set above the peaks of the background sample. In our implementation, we use offset equal to 5.

Next step is to project the binary image along the horizontal and vertical axis. The projections P_x and P_y of the binary image I_b are given as the pixel sum in the given direction:

$$P_x = \sum_j I_{b,j} ,$$

$$P_y = \sum_i I_{b,i} ,$$

where i is the row index and j is the column index.

The dish is, in these projections, manifested as a broad 'lobe', or peak (see Fig. 6 on page 115). The partly uncovered background – if present – would show up as another peak (cf. the graphs in the bottom row of Fig. 6 on page 115), however, the

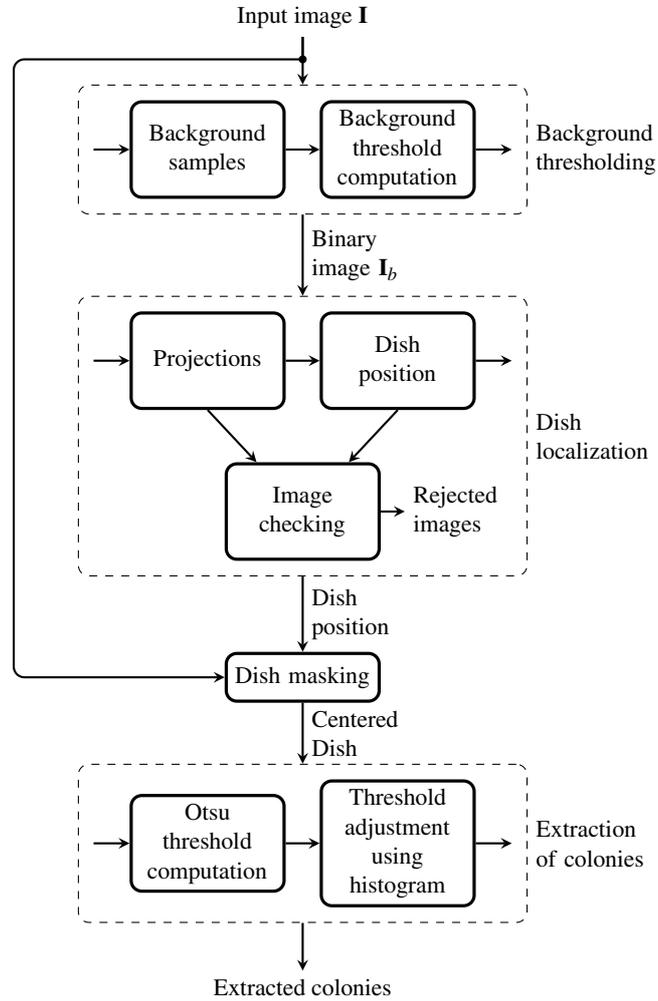


Fig. 4. Processing flow of dish preprocessing



Fig. 5. Position of image background samples

lobe corresponding with the dish is the broadest one. Additional check for uniformity of estimated diameters in horizontal and vertical direction is applied to eliminate heavily corrupted images.

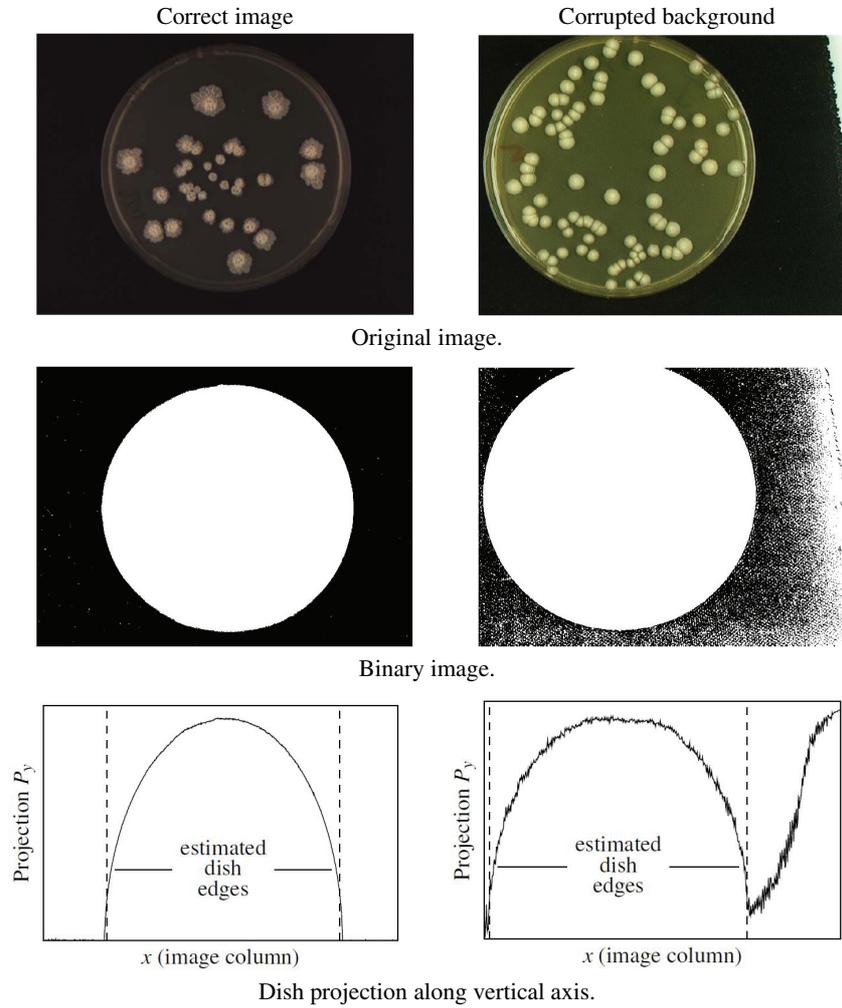


Fig. 6. Computation of dish projection P_y – example

Finally, the dish is approximated by a circle with the center and diameter derived from the edges determined in the dish localization step. The result of the preprocessing is a centered image of the dish, with eliminated background. An illustrative example is given in Fig. 7 on the following page.

The last step of the image preprocessing is the *extraction of colonies*, which is used to assess the total covered area, in order to determine the relative coverage of the dish. A new threshold is used in this step, located between the brightness levels of the dish background and of the colonies, as close as possible to the background.

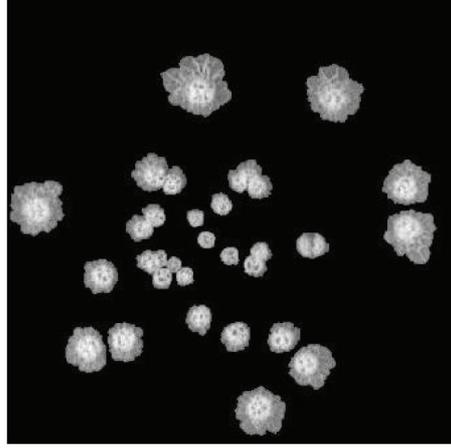


Fig. 7. Dish image after preprocessing

The most straightforward solution is to use the well-established Otsu thresholding [5]. However, the threshold based purely on this method may be set too high, if the colony edge is fading gradually. To avoid this, we have introduced a *histogram-based correction* to the method (see Fig. 8 on the next page and Fig. 9 on page 118).

The principle of this correction is following: first, by limiting the histogram by the Otsu threshold θ_O , we cut-off the part corresponding with (brighter) colonies. The peak of the remaining part (Fig. 9) corresponds with the dish background. The threshold should be placed just above the right edge of this peak.

More formally, let $H = [h_1, \dots, h_O]$ be the histogram of the pixels below the Otsu threshold θ_O . Let H_{MA} be histogram filtered by a moving average filter:

$$H_{MA} = [h_{MA,1}, \dots, h_{MA,O}], \quad h_{MA,i} = \left(\sum_{j=1}^N h_{i-j+1} \right) / N, \quad i-j+1 < 0 : h_{i-j+1} = 0.$$

Let $\Delta_{H_{MA}}$ be the first difference of the filtered histogram. Then, the dish threshold with correction θ_d is given as

$$\theta_d = \max(i), \quad \Delta_{H_{MA}}(i) < -s_d,$$

where s_d is a user-selected setpoint. In our implementation, we use $N = 4$ and $s_d = 100$, with good thresholding performance.

3.2 Colony Counting

In this step, the centered image of the colonies, resulting from preprocessing, is used to find the individual colonies.

The simplest approach to counting is to correlate the contour of the colonies with a circular pattern, assuming that the maxima of correlation are located at the centers of colonies. This has been the first method we have tested. During experiments, we

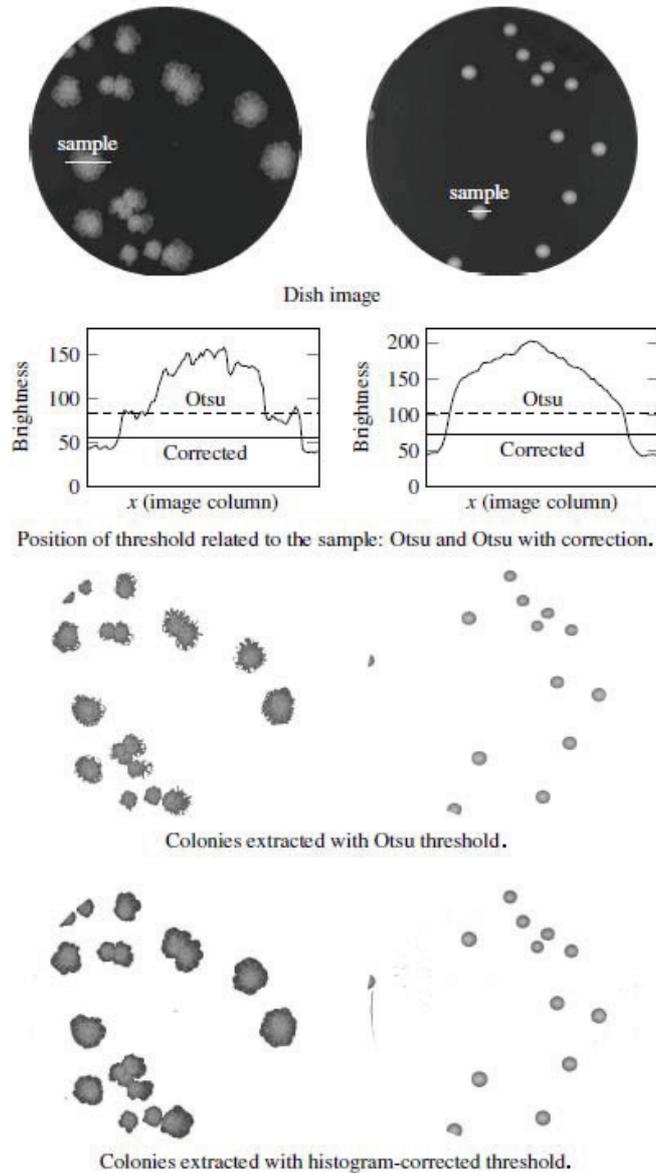


Fig. 8. Dish thresholding - pure Otsu method and a histogram-based correction

have experienced poor performance in the case of big clusters (the error rate of our implementation was reaching up to 30%) and sensitivity to mismatch between the radius of the circular pattern diameter and the radius of the colony (and, hence, sensitivity to dispersion of colony radii).

An interesting approach to the counting of bacteria colonies has been described in [3]: to find colonies, a number of shape and structure criteria are used on the

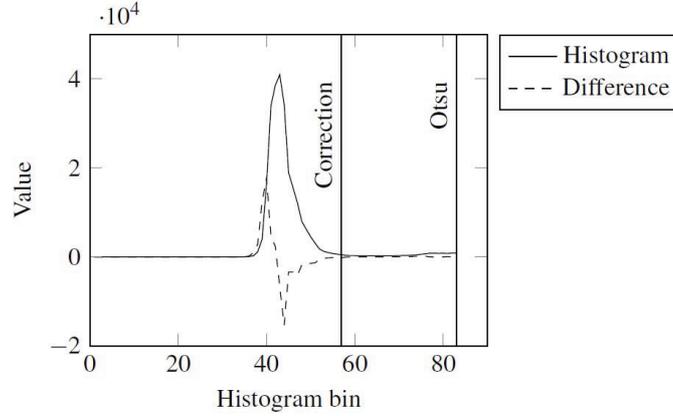


Fig. 9. Position of Otsu threshold and of threshold with correction relative to histogram

image pixels, including e.g. mean object radius, roundness of an object, compactness and asymmetry, radial monotony of fall-off in intensity, etc. These criteria are combined into shape and structure quality parameters and evaluated using fuzzy logic. The method works best under the assumptions of well-defined circular shape and monotone intensity fall-off of the colonies, which are well satisfied for bacteria (the case treated in the work of Marotz), but not necessarily in our case of the yeast colonies (for examples of possible morphologies see e.g. [6,7]).

A popular method to detect circular objects is the circular Hough transform (for definition see e.g. [8]). Based on our experiments, if multiple yeast colonies are overlapping or touching, it tends to detect (incorrectly) adjacent colonies as a single object. Also, following the results presented in [9], the output of the Hough transform may be rather noisy.

Another possibility is to use the methods based on *radial symmetry* – the radial symmetry transform has been introduced in [10].

Fast Radial Transform. In our solution, we use the Fast radial transform – a modified version of the radial symmetry transform, first presented in [9]. It is a transform that maps the original image to the transformed image according to its contribution to radial symmetry of the gradients at distance $n \in N$ (N is the set of radii) away from each point. For full details of the method see the original description.

First, image gradient $g_{i,j}$ at each point (i, j) is calculated. Then, the *positively-* and *negatively affected pixels* are determined. An affected pixel is defined as the point either *in-* or *counter* the direction of the gradient vector $g_{i,j}$, at a distance n pixels away from the point at coordinates (i, j) . The coordinates of affected pixels are given by

$$p_{+}(i, j) = (i, j) + \text{round} \left(\frac{g_{i,j}}{\|g_{i,j}\|} n \right) ,$$

$$p_-(i, j) = (i, j) - \text{round}\left(\frac{g_{i,j}}{\|g_{i,j}\|} n\right) .$$

The gradient matrix \mathbf{g} , together with the coordinates of affected pixels, is used to determine the *orientation* and *magnitude* projection images \mathbf{O}_n and \mathbf{M}_n for the given radius n :

$$\begin{aligned} O_n(p_+(i, j)) &= O_n(p_+(i, j)) + 1 , \\ O_n(p_-(i, j)) &= O_n(p_-(i, j)) - 1 , \\ M_n(p_+(i, j)) &= M_n(p_+(i, j)) + \|g_{i,j}\| , \\ M_n(p_-(i, j)) &= M_n(p_-(i, j)) - \|g_{i,j}\| . \end{aligned}$$

The radial symmetry at radius n is defined by convolution of matrices

$$\mathbf{S}_n = \mathbf{F}_n * \mathbf{A}_n ,$$

where the elements

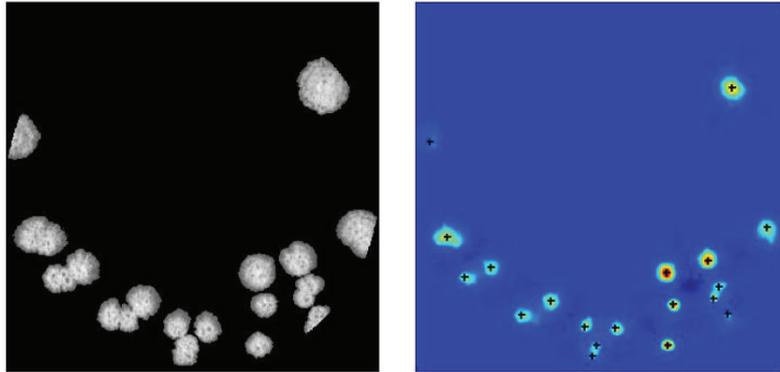
$$F_n(i, j) = \frac{M_n(i, j)}{k_n} \left(\frac{|\tilde{O}_n(i, j)|}{k_n} \right)^\alpha ,$$

and

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

Here, \mathbf{A}_n is a two-dimensional Gaussian, α is the so-called radial strictness parameter and k_n is a scaling factor used to normalize \mathbf{M}_n and \mathbf{O}_n (for settings used in our implementation see 5.1). Projection images \mathbf{M}_n and \mathbf{O}_n are initially set to zero.

An example of a symmetry map for a dish is given in Figure 10. Note that in our implementation, the symmetry map is only computed for the foreground pixels.



(a) Preprocessed image of colonies.

(b) Symmetry map S with the centers of colonies.

Fig. 10. Illustration of function of the radial symmetry algorithm

Colony Radius Estimation. Let us recall that there are often colonies with different radii contained in a single dish (see Sec. 2). Hence, prior to the radial transform computation, the set of radii N of the colonies in the dish is to be estimated, using the following procedure:

- the equivalent diameter d and eccentricity ε is computed for each object in the image. The diameter is derived from area A (number of pixels) of an object. The eccentricity equals to the eccentricity of an ellipse with the same second moments as the object. It ranges from $\varepsilon = 0$ for a circle to $\varepsilon = 1$ for a line segment.
- Nearly circular objects (with eccentricity $\varepsilon < \theta_\varepsilon$, where θ_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 v equidistant subintervals. The set of radii N for the fast radial transform is then given by:

$$\Delta = d_{\max} - d_{\min}$$

$$N = \{d_{\min} + [0, \dots, v] \cdot \Delta/v\}/2$$

Estimation of Colony Centers. The centers of the colonies are represented by local maxima of the symmetry matrix S , which are detected using the `nonmaxsuppts()` function described in [11]. The output of this function are the coordinates of the centers.

4 Counting Tool

The processing flow, as described in Section 3, has been implemented in a counting tool (Fig. 11 on the next page). It performs fully automatic thresholding of the image, localization of the dish and counting of colonies. The tool provides an environment for selection of the directory tree with images to process, of the file to store the counting results, and of the counting mode: in the semi-automatic mode, a simple point-and-click editor can be used for manual correction of the system output, in the fully-automatic mode, the output of the system is directly stored into the result file. This tool was used also to obtain the reference results used in this paper: using manual correction, we obtained the ground-truth counts of colonies in Petri dishes, which were used to evaluate the counting error rates.

5 Experiments and Results

To evaluate the system recognition performance, it has been tested using a set of 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 Fig. 12a. Figure 12b 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. In the set, the minimum and maximum counts were 8 and 95 colonies per dish, when 80% of dishes contained less than 50 colonies. The minimum and maximum coverage were 0.43% and 25.3%, respectively, and 80% of dishes had coverage up to 8.26%.



Fig. 11. Counting application

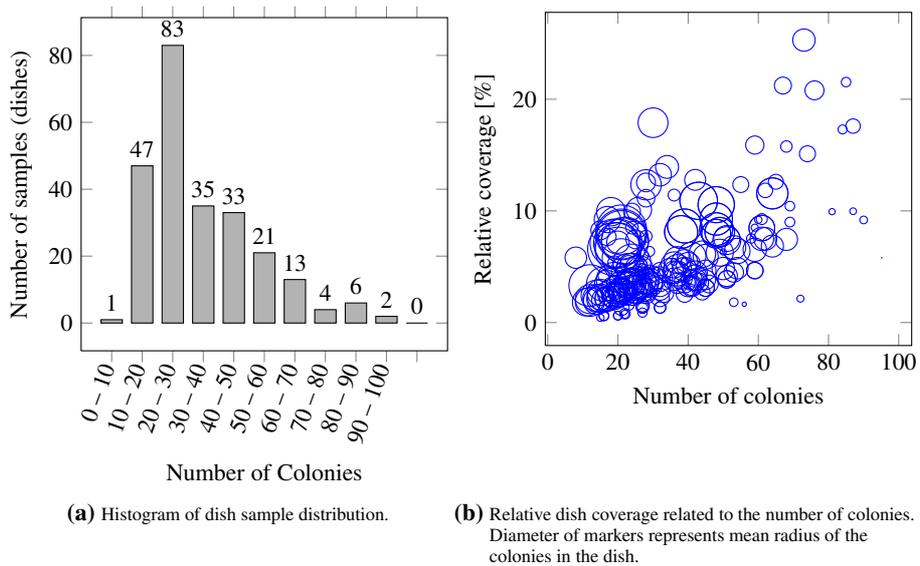


Fig. 12. Distribution of the test set

5.1 Algorithm Settings

The following settings were used in our experiments:

- Parameters of the fast radial transform:

$$k_n = 6, \alpha = 4$$

Let us note that setting $\alpha = 2$ (see Sec. 5.1) could be better choice – the response of the radial transform on fluffy colonies with less circular shape would be higher

and the computational demands of the method would be somewhat lower. However, all experimental data presented later have been obtained for $\alpha = 4$. We have performed a test with $\alpha = 2$ on a randomly selected sample of 15 images. The counting performance has been slightly improved, however, the system was more sensitive to the background noise along the dish edge (giving numerous false positives in some cases). The problem of elimination of the reflections on the dish edge has been treated e.g. in [4]; we perform only simple reduction of the dish radius, which should cover the dish lip region.

- Parameters of the non-maxima suppression `nonmaxsuppts()` [11]:

$$\text{thresh} = 4, \quad \text{radius} = 0.8 \cdot \min(N),$$

where N is the set of radii for the fast radial transform.

- Construction of N : eccentricity threshold θ_e (see Section 3.2) is initially set to $\theta_e = 0.25$. The number of equidistant intervals v is set to $v = 4$. At least five colonies of given eccentricity must be in the image, else the eccentricity is increased by step of 0.1 (we start from stricter requirement on circularity of colonies, to eliminate clusters. If there are not enough colonies of the given circularity, we loosen the criterion and accept those with less circular shape).

Let us note that the distribution of colony diameters should also be considered, however, this is not implemented in the tool at the moment.

5.2 Typical Detection Errors

Figure 13 illustrates typical detection errors of fast radial transform. A colony may be missed if it touches other colony or multiple colonies, so that they form a cluster. This detection error is almost absent if the colonies touch only at one point, creating a chain-like structure. More colonies touching each other 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.

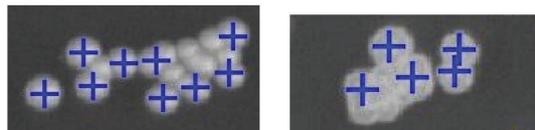


Fig. 13. Example of typical detection error of fast radial transform

5.3 Counting Performance

The counting errors for various number of colonies are summarized in Tab. 1 on the facing page. 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 (Fig. 13). Second, the number of samples with this high density of coverage was relatively low, thus increasing the evaluation error (Fig. 12a on the previous page).

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

# 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

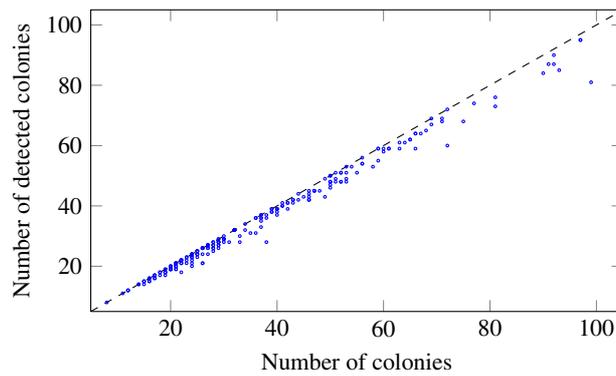


Fig. 14. Algorithm recognition performance related to the number of colonies

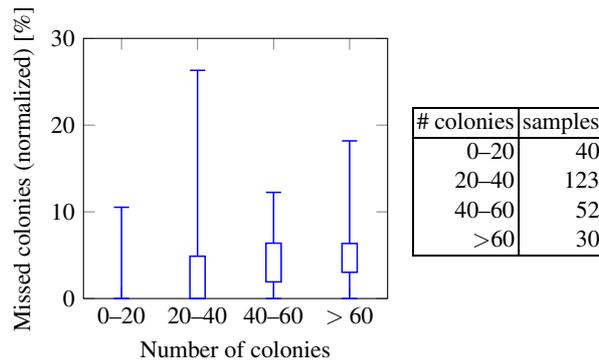


Fig. 15. Relative number of missed colonies dependent on the number of colonies

Figure 14 on the preceding page shows the recognition performance related to the number of colonies in the dish. The dishes, where some colonies have been missed fall below the dashed line.

Another view on the algorithm performance is provided by Fig. 15, which presents a box-whisker plot of relative counting error (missed colonies) in several groups of the colony counts. The numbers of samples per group are given in the attached table.

6 Conclusions

In the paper, we have presented the tool for evaluation of yeast colony images, targeted towards batch processing of images prepared with a general-purpose imaging system. The processing flow of the system has been described. The performance of the tool has been tested on a set of 245 images with different degrees of coverage. The distribution of the data in the test set and the performance of the system have been discussed. The average counting error is below 4%. It is difficult to compare the performance with the commercial solutions, since the performance data of these systems are not available.

There are still performance challenges, especially in resolution of clustered fluffy colonies. The resolution of clustered colonies is rather common problem, reported also in other works [4]. The performance, however, has been acceptable for our target laboratory user.

The application has been implemented in a Matlab. This tool has been successfully deployed in the cooperating biology research laboratory and is available for download⁴.

Acknowledgements. This research has been supported from the TA01010931 "System for support of the FISH method evaluation" project of the Technology Agency of 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, who have introduced us into the problem and provided us with the sample images from their experiments.

References

1. Goss, W.A., Michaud, R.N., McGrath, M.B.: Evaluation of an automated colony counter. *Appl. Microbiol.* 27, 264–267 (1974)
2. Burger, W., Burge, M.J.: *Digital Image Processing: An Algorithmic Introduction using Java*. Springer, Heidelberg (2007)
3. Marotz, J., Lübert, C., Eisenbeiss, W.: Effective object recognition for automated counting of colonies in petri dishes (automated colony counting). *Computer Methods and Programs in Biomedicine* 66, 183–198 (2001)
4. Chen, W.B., Zhang, C.: An automated bacterial colony counting and classification system. *Information Systems Frontiers* 11, 349–368 (2009)
5. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9, 62–66 (1979)

⁴ <http://zs.utia.cz/index.php?ids=results&id=yeastcolcount&lang=eng>

6. Granek, J.A., Magwene, P.M.: Environmental and genetic determinants of colony morphology in yeast. *PLoS Genetics* 6, e1000823 (2010)
7. Št'ovíček, V., Váchová, L., Kuthan, M., Palková, Z.: General factors important for the formation of structured biofilm-like yeast colonies. *Fungal Genetics and Biology: FG & B* 47, 1012–1022 (2010)
8. Ballard, D.H., Brown, C.M.: *Computer Vision*. Online Book (2003)
9. Loy, G., Zelinsky, A.: Fast radial symmetry for detecting points of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8, 959–973 (2003)
10. Reissfeld, D., Wolfson, H., Yeshurun, Y.: Context-free attentional operators: The generalized symmetry transform. *International Journal of Computer Vision* 14, 119–130 (1995)
11. Kovesi, P.D.: *MATLAB and Octave functions for computer vision and image processing*. School of Computer Science & Software Engineering, The University of Western Australia (2005), <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>