# Image processing and classification algorithm for yeast cell morphology in a microfluidic chip

Bo Yang Yu
Caglar Elbuken
Carolyn L. Ren
Jan P. Huissoon

SPIE

# Image processing and classification algorithm for yeast cell morphology in a microfluidic chip

**Bo Yang Yu, Caglar Elbuken, Carolyn L. Ren, and Jan P. Huissoon**
University of Waterloo, Department of Mechanical and Mechatronics Engineering, Waterloo, Ontario, N2L 3G1, Canada

**Abstract.** The study of yeast cell morphology requires consistent identification of cell cycle phases based on cell bud size. A computer-based image processing algorithm is designed to automatically classify microscopic images of yeast cells in a microfluidic channel environment. The images were enhanced to reduce background noise, and a robust segmentation algorithm is developed to extract geometrical features including compactness, axis ratio, and bud size. The features are then used for classification, and the accuracy of various machine-learning classifiers is compared. The linear support vector machine, distance-based classification, and $k$-nearest-neighbor algorithm were the classifiers used in this experiment. The performance of the system under various illumination and focusing conditions were also tested. The results suggest it is possible to automatically classify yeast cells based on their morphological characteristics with noisy and low-contrast images. © *2011 Society of Photo-Optical Instrumentation Engineers (SPIE)*. [DOI: 10.1117/1.3589100]

## 1 Introduction

A complete life cycle of eukaryotic cells is normally divided into four phases: first gap phase ($G_1$ phase), synthesis phase ($S$ phase), mitosis phase ($M$ phase), and second gap phase ($G_2$ phase). The process of DNA replication, which is regulated by several control mechanisms, happens in the $S$ phase. Following the $S$ phase, replicated chromosomes separate during the $M$ phase and segregate into two nuclei that will eventually be endowed to each newborn daughter cell at cell division. The $G_1$ phase and $G_2$ phase separate cell birth from the $S$ phase, and the $S$ phase from the $M$ phase, respectively.

Understanding cell cycle regulation is of vital importance to the understanding of cancer development.[1] The budding yeast cells, Saccharomyces cerevisiae, are frequently used as a model species in the study of cell cycles, because the basic elements of the yeast cell's structure are extensively homologous to those in higher plant and animal cells,[2] and the progression of the yeast cell cycle is easily monitored via changes in cell morphology.[3,4] As shown in Fig. 1, cells in the $G_1$ phase are characterized by a simple ellipsoidal shape. When cells enter the $S$ phase, a readily visible bud emerges and, as the bud size grows larger, the cell enters the $M$ phase. The ability to accurately identify yeast cells in different division phases, especially cells in the $S$ phase, is critical in the modeling of cell cycles.[1,5] Currently, the classification of the cell cycle is currently done manually, which is often subjective, inconsistent, and time-consuming.[6] In addition, there is no effective method for collecting and isolating cells in the phase of interest. The development of an automated device that can identify and isolate cells in a particular cell cycle phase is thus crucial to the systematic study of cell cycle modeling.

Several image-based yeast cell morphology identification devices/algorithms have been previously described. Koschwanez et al.[7] reported a fiber-optic bundle imaging device capable of identifying a budding and nonbudding yeast cell. The device does not require a microscope but cannot extract any information on bud size, thus the classifier cannot distinguish between cells in the $S$- and $M$-phase. Ohtani et al. developed an image processing program (called CalMorph) that can extract quantitative information on cell morphology such as cell size, roundness, and bud neck position, etc.; however, this is limited to fluorescent stained microscopic images.[8] There are also several microfluidic devices for yeast cell imaging mentioned in the literature. Lee et al. reported a microfluidic chip with a passive filtering-and-trapping mechanism that is capable of fixing cells in the same focal plane for image without any moving components on the chip.[9] Ohnuki et al. developed an active cell trapping mechanism that uses flexible partical desportin mass spectrometry membranes to hold cells stationary in the same focal plane.[6] However, both designs lack the ability to isolate or manipulate single yeast cells.

Although the above mentioned studies have applied image analysis to the morphological analysis of yeast cells, attempts to combine pattern recognition and image processing for the classification of yeast cells cycle phases are scarce. Supervised pattern recognition methods have been successfully applied to the imaging detection of bacterial colony,[10,11] lesions,[12] and cancers.[13,14] The classification of yeast cell cycle phases is also an excellent application for supervised machine learning, as training data can be easily obtained from the vast databases already developed for yeast cell morphology. In this paper, an image-based machine learning algorithm is reported which is

---

Address all correspondence to: Carolyn L. Ren, University of Waterloo Department of Mechanical and Mechatronics Engineering, Waterloo, Ontario, N2L 3G1, Canada. Tel: 1-519-888-4567×33030; Fax: 1-519-885-5862; E-mail: c3ren@uwaterloo.ca.
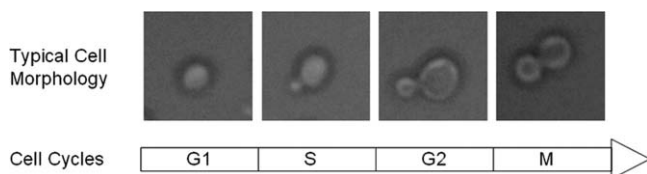
**Fig. 1** Yeast cell morphology through cell cycle progression.

not dependent on fluorescent staining. The algorithm analyzes nonfluorescent microscopic images of yeast cells, extracts morphological features from the cells, and classifies the cells into the most-fitting cell cycle phases based on the information it was trained on previously. This algorithm is intended for implementation in a microfluidic chip involving cell studies; therefore, it must account for all the constraints and special circumstances in a microfluidic channel environment, such as high and uneven background noise and blurring due to rotation and drifting.

The details of the design and experimentation of the algorithm is explained in Secs. 2–5. Section 2 describes the methods for cell harvesting and data collection. Section 3 introduces an image enhancement and segmentation algorithm that initially eliminates background noise and improves the contrast of the microscopic cell images, then applies threshold and extract geometrical information from the images. In Sec. 4, the appropriate features are selected and three machine-learning classifiers are chosen. In Sec. 5, the performance of the image analysis algorithm and the classifiers under different conditions are studied.

## 2 Materials and Preparation methods

### 2.1 Yeast Culture

The W303 strain of yeast cells were used for this study. The following procedure was used for the cell culturing: 1. Frozen permanents are streaked out on media plates and incubated overnight at 37 °C until the colonies are visible. 2. A single colony is picked from the plate using a sterile pipette tip and used to inoculate a 10 ml culture of liquid media. 3. Incubate 10 ml culture overnight until saturated. 4. Inoculate saturated culture into fresh liquid media. Incubate overnight, and then dilute in the YPD media to a concentration of approximate $10^7$ cells/ml. (The YPD media contains 1% Bacto Yeast extract, 2% Bacto Peptone, and 2% glucose) This cell solution is constantly agitated using a magnetic stirrer during the experiments to keep the cells from clumping.

### 2.2 Stationary Cell Images

A drop of the cell culture media was placed on a glass slide and observed using an Olympus BX51 microscope with a 50×/0.5 objective that has a depth of field of 1 $\mu$m. Images of the cells were captured using a CCD camera interfaced with ImagePro software as 8-bit TIFF and later converted to type double in MATLAB. The exposure time was set to 90 min for regular cell images. One image with no-cells in the viewing area was taken to serve as the background image. 100×100 pixel image clips containing only one cell (referred to as cell-clip) were cropped from the raw images to form a data set, and for each cell-clip, the same-size image clip was cropped at the same location from the background image. This clip (background-clip) serves as an estimation of the true background of the cell-clip. Each of these

cell-clips were labeled according to the relative size of its bud as class 1 (phase $G_1$: no bud), class 2 (phase S: small bud), and class 3 (phases $G_2$/M: large bud). In total, 240 stationary cell-clips were collected with 70 samples each in class 1 and class 2, and 100 samples in class 3.

## 3 Image Processing Algorithms

The main goal of the image processing algorithm is to extract statistically relevant features from the cell images in order to classify cells in different division phases. The first step is to isolate the cell areas from the background, also known as image segmentation. Image segmentation of cells is an active research area of biomedical image processing, with active contour methods being considered the first choice of cell image segmentation.[15] Both the parametric form (i.e., snake mode) contour approaches,[16,17] as well as the nonparametric approach (i.e., level set),[18–20] have been successfully demonstrated in cell detection and tracking studies. Compared to the contour-based, the traditional threshold methods for segmentation tend to be more prone to noise, but are conceptually simpler and often very effective,[21] also suggested by previous yeast cell morphological studies.[7,8,22] Since the algorithm is intended for a controlled environment (microfluidic chip) where background noise can be determined and eliminated, the proposed image processing algorithm will use a threshold approach along with an image enhancement step.

### 3.1 Image Enhancement Sub-Algorithm

This sub-algorithm uses the prior knowledge of the background noise in the microscope field of view to eliminate noise and improve the contrast of cell images, to help aid in the image segmentation. The approach taken by this algorithm is to examine each individual pixel of the image clip and then map the intensity of the pixel toward the mean intensity value if the pixel intensity is within the noise range of the background, or map it away from the mean intensity if the pixel intensity is outside the noise range of the background.

First, a clip is cropped from a blank microscope field of view, and is averaged using a 20×20 mask to produce a background-mean matrix: $\mu_{bg}$, and the variance of each 20×20 region is also computed to form a variance matrix: $\sigma_{bg}^2$ of the background. Then, for the intensity of each pixel in the cell clip, its generalized euclidean distance (GED, also known as Mahalanobis distance) to the background mean is computed:[17]

$$\text{GED}_{i,j} = \frac{p_{i,j} - \mu_{bg-i,j}}{\sigma_{bg-i,j}}, \quad \forall i, j, \quad (1)$$

where $p_{i,j}$ is the gray level of the pixel at index $(i, j)$. Next, a mapping function is applied to each pixel's GED to map it to a new value GED' using Eq. (2):

$$\begin{cases} \text{GED}'_{i,j} = \left(\dfrac{\text{GED}_{i,j}}{n}\right)^3, & |\text{GED}_{i,j}| < n \\ \text{GED}'_{i,j} = A \times \text{GED}_{i,j}, & |\text{GED}_{i,j}| \geq n \end{cases} \quad \forall i, j, \quad (2)$$

where $n$ is a threshold value and $A$ is an amplification factor. This mapping function maps all pixels with GED less than $n$ closer to zero, while it amplifies the pixels with GED greater than $n$ further away from $n$. The value of $n$ can significantly alter the image segmentation process described in Sec. 3.2: smaller
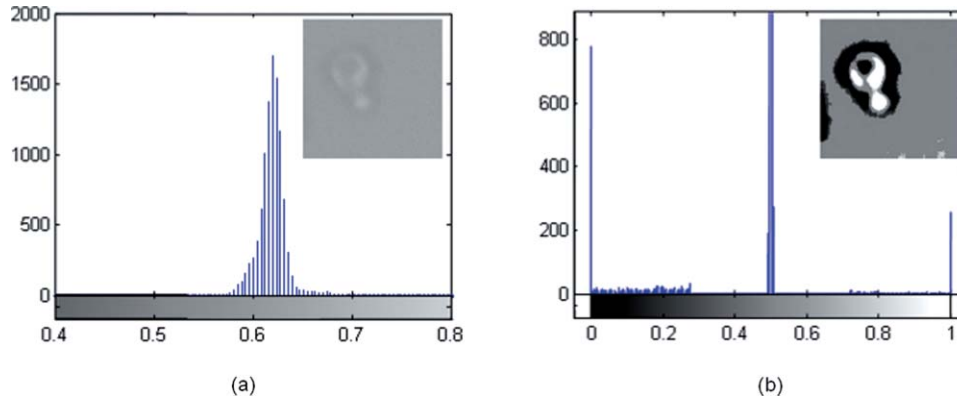
**Fig. 2** (a) Cell clip and histogram prior to enhancement. (b) Enhanced cell clip and histogram.

values of $n$ result in a thicker cell boundary but with more noise, while larger values of $n$ create thinner cell boundaries and less noise. The impact of $A$ on the segmentation process is not as significant. The value of $A$ only controls the distance between the clusters, a sufficiently large value of $A$ will result in a large enough separation between the clusters. The preferred values for the amplification factor $A$ and threshold $n$ were both determined heuristically to be 20 and 0.8, respectively.

To convert GED′ back to pixel values, the reversion of Eq. (1) is used, while setting the new mean intensity to 0.5 and hard limiting the intensity range to between 0 and 1. Fig. 2(b) shows a cell-clip after applying the enhancement algorithm and its histogram. It is obvious that the enhanced clips have lower noise, and the contrast is significantly improved.

This method and homomorphic filters achieve similar results because both are able to increase contrast and normalize the brightness of an image.[23] However, homomorphic filters achieve illumination correction by assuming the inhomogeneities as low frequency components,[24] while the report method can completely remove the background information, which include both the nonuniform illumination and random high frequency inhomogeneities. This is especially important, since for the microfluidic cell sorting application of this algorithm, there will likely be unwanted objects in the microscope field of view, including channel walls, debris from fabrication, and dust particles. These objects are stationary and could be completely removed by the algorithm. One drawback with the enhancement algorithm is that the mapping depends solely on the pixel intensity, and certain orientations of the cell could result in some of the cell bound-

ary/interior pixels having intensity values within the noise range. This will result in visible gaps in the cell boundary/interior after enhancement (Fig. 3). A more robust enhancement algorithm would take into account the spatial distribution of the pixels as well as pixel intensities.

### 3.2 Shape Identification Sub-Algorithm

The enhanced images now have clearly defined intensity differences between the background, cell boundary, and cell interior. By observing the enhanced cell-clips, it was noted that the interior edge of the dark cell boundary represents the shape of the cell very well, therefore a simple threshold method can be used to obtain this boundary shape. However, for some cell images, such an edge may not exist if the cell boundary does not close completely, for example the clip in Fig. 3. A heuristic method is determined to connect and close an open boundary by obtaining an estimate of the boundary and then add it back to the original image. The steps of the shape identification algorithm are shown in Fig. 3.

By enhancing the image before segmentation, the proposed algorithm overcomes the traditional poor performance of threshold based segmentation methods under noisy environments. Another major limitation of this segmentation method—also a common limitation of threshold methods—is the inability to handle multiple cells sticking to each other.[16] However, for the specific application of cell sorting, the cells must be isolated from each other. The segmentation algorithm has the benefits of the threshold-based methods in terms of speed and simplicity. At the same time, for the specific application of microfluidic cell sorting, the general limitations of threshold methods does not limit the algorithm's performances.
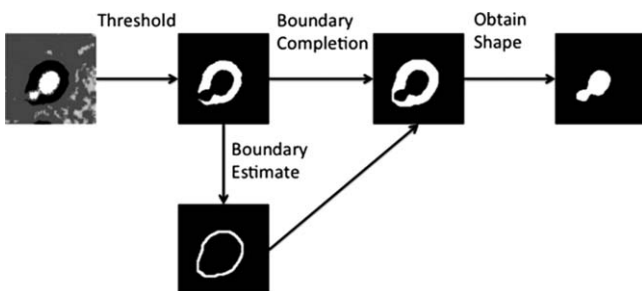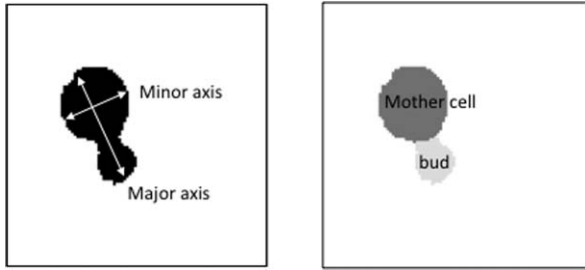
### 3.3 Feature Extraction Sub-Algorithm

With the cell shapes successfully extracted from the original images, it is now possible to extract specific features from the shapes to classify between cells in different cell division phases. Manual classification of yeast cells is done by first looking for a visible bud ($G_1$ or other) and then examines the size of the bud ($S$, $G_2$, and $M$). The automated classifier will also follow this two-step guideline. The major difference between budding and nonbudding cells is their shape: nonbudding cells look like compacted circles while budding cells have a peanut-like shape. Thus, features must be extracted from the cell images that can reflect the shape and bud size of the cells.



**Fig. 3** Shape identification steps. Direct threshold could produce a cell boundary region with gaps. An estimate of the cell edge can be obtained with several iterations of pixel dilation and erosion. This boundary estimate can then be added back onto the cell boundary region to fill any gaps in order to obtain the final cell shape.

**Fig. 4** Description of features. Left: Major and minor axis. Right: Mother cell and bud.

The compactness is a geometric feature that is measured by the ratio of a shape's area to its perimeter squared.[25] A nonbudding cell should have a lower compactness measurement while a budding cell will have a higher compactness measurement.

The axis ratio (shown in Fig. 4) is the ratio of the shape's major axis to its minor axis. A nonbudding cell, looking more like a circle, will have an axis ratio closer to 1, while cells with larger buds will have larger axis ratios. The quantities needed for calculating these two features can all be obtained using MATLAB's binary region analysis tool.

To extract the bud size feature from the cell shapes, the bud must be isolated from the parent cell. An existing method is look for convex cell boundaries that represent the location of the bud neck.[8] This method is very efficient but its accuracy can be unreliable. The proposed bud isolation algorithm assumes that the parent cell is circular and tries to locate it using circle detection techniques. First, the area of the cell shape $A_c$ is determined (in pixels), and then its equivalent radius is calculated as $\sqrt{A_c/\pi}$ rounded up to an integer (pixel) value. Then, a circular mask of the same radius is created and slid across all the pixels that belong to the cell shape in order to find the location where the mask best matches the cell shape. All the pixels that are covered by the circular mask are then removed and the remaining pixels represent the bud. Then, the bud-to-area ratio feature can be obtained by dividing the size of the bud in pixels to the size of the entire cell.

**Table 1** Inter- and Intraclass distances.

| | Axis ratio | Compactness | Bud to area ratio |
|---|---|---|---|
| Interclass distance Sb | 0.0112 | 0.0212 | 0.0479 |
| Intraclass distance Sw | 0.0138 | 0.0125 | 0.0289 |
| Ratio of Sb to Sw (signal-to-noise ratio) | 0.814 | 1.699 | 1.6558 |

Of the three features mentioned above, the axis ratio and compactness can be obtained with the shortest time, while the bud ratio is the most complicated feature to obtain and requires more computational time.

## 4 Feature Spaces and Classifiers

All of the 240 stationary cell-clips were enhanced and segmented using the image process algorithm described in Sec. 3.2. The three features–axis ratio, compactness, and bud-to-area ratio–were extracted, and the sample means and variances for each class in each feature space were computed. Two values that represent class distances were calculated for each of the 1D feature spaces: the inter-class distance $S_b$ (between-scatter) that describes the scattering of the class-dependent sample means around the overall average, and the intraclass distance $S_w$ (within-scatter) that describes the average scattering within classes[19] (the formulas are not introduced due to their complexity, they can be found in the referenced textbook). These numbers are tabulated in Table 1. The 2D distributions in the compactness-axis ratio space and in the bud ratio-compactness space along with the histogram distributions of each feature spaces are plotted in Fig. 5.
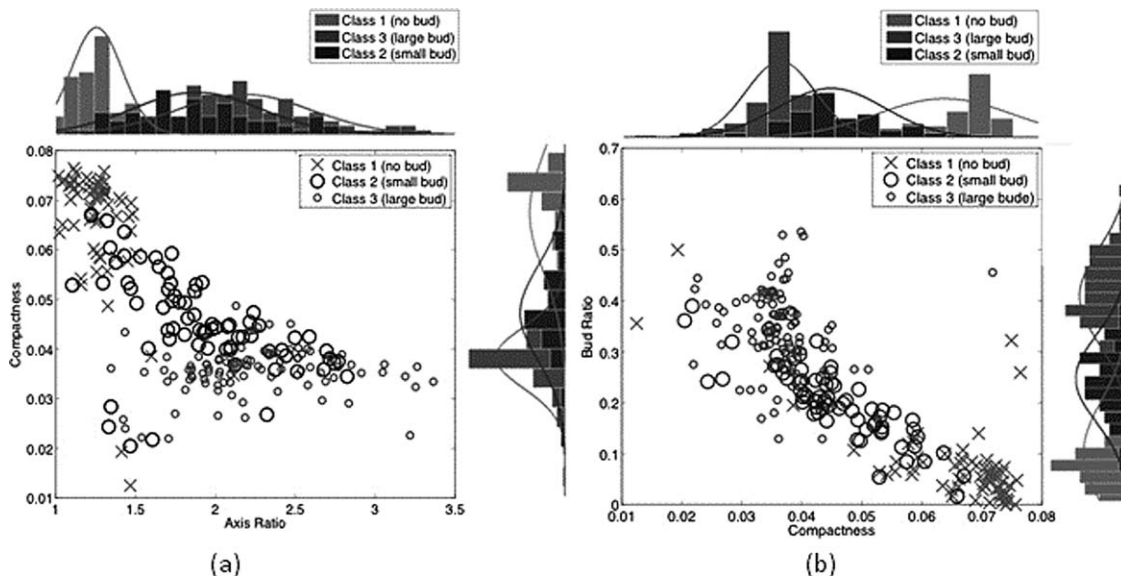


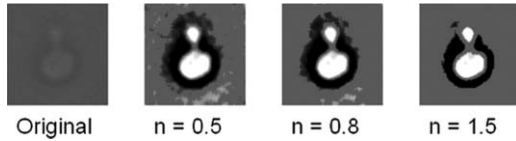**Fig. 5** Class distributions in different features spaces.

**Fig. 6** Enhancement result for different threshold value *n*.



**Fig. 7** Examples of cell clips. (a) original images. (b) segmented images. (c) bud separation result.

The following observations can be made from these figures:

1. The classes are roughly Gaussian in shape, but are not linearly separable in any feature spaces.

2. Class 1 (no bud) has the most compact class shape in any feature space as suggested by the histograms, while classes 2 and 3 tend to be more scattered with more outliers.

3. The feature "bud ratio" shows the best separation between classes in this feature space (higher between-scatter value in Table 1). The classes are most compact in feature space "compactness" (low within-scatter value in Table 1) among three classes. Both features have similar signal-to-noise ratios.

4. In all feature spaces, some data points from class 2 and 3 are consistently found in the region of class 1. These data points correspond to cells that lost their buds during the image segmentation process, because occasionally, the orientation of the cells make the bud appear extra white in color and do not show a dark boundary around the bud.

5. Some cells in class 1 were identified with large bud ratio and/or axis ratio, because these cells have elongated ellipsoidal shapes, possibly due to the fact that stirring of the cell solution stretched the cells.

6. The axis ratio and compactness features both show good separation of class 1 from the rest of the classes, but not good separation between class 2 and 3, especially for axis ratio.

7. The bud ratio feature would be the only feature that is capable for classifing between classes 2 and 3 with reasonable accuracy.

Three different feature sets are proposed. Set #1 uses only single features (1-Dimensional feature spaces) for classification: it uses compactness to classify between class 1 cells from the rest of the classes, and bud size to classify between class 2 cells and class 3 cells. Set #2 uses two features (2D feature spaces) simultaneously: axis ratio and compactness together for class 1, and compactness and bud size together to classify between classes 2 and 3. In Set #3, all three features are used for classification between the three classes (3D feature space).

Three classification methods are also proposed: the GED, or (Mahalanobis distance) classification, *k*'th nearest neighbors (kNN), and linear-kernel support vector machine (SVM). The GED classifier is a parametric classification method that is only accurate if the classes are Gaussian in shape.[25] The linear-kernel SVM tries to find a linear discriminant that best separates two classes; it is the most computationally efficient of the three but also requires that the classes be non-onvex for accurate classification. kNN is a nonparametric classifier that assigns a sample to
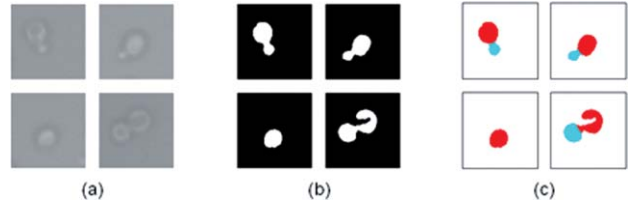
the class of its kth closest training data point.[18] It applies to any class shape, but is more computationally inefficient compared to the other two methods.

## 5 Performance

In this section, the performances of various components of the image based classification algorithm are evaluated, so the optimal parameters can be chosen and the limitations of the algorithm can be understood.

### 5.1 Image Enhancement

### 5.1.1 Threshold parameter

As mentioned in Sec. 3.1, the threshold value *n* in Eq. (2) has a critical role in the entire algorithm. The value of *n* represents the cut-off value between the background noise and cell features, and the effect of various values of *n* is shown in Fig. 6:

As *n* increases, more background noise is eliminated by the enhancement algorithm and the edge of the cell membrane becomes sharper. However, a high *n* value results in unclosed cell membrane boundaries, and subsequently the failure to extract features from the image since the segmentation algorithm only tries to find the inner edge of the cell membrane. A low *n* value will also result in incorrect features due to background noise.

An experiment is conducted where the image enhancement, segmentation, and feature extraction algorithms are applied to the entire set of cell clips with different values of *n*. Two criteria were used to evaluate the performance of these algorithms: the number of clips (out of 240) that failed to return a cell shape and the ratio of interclass distance (Sb) to intraclass distance (Sw)

**Table 2** Comparison of image segmentation performance for different *n* values.

| n | Fail to extract feature | Interclass distance (Sb) to intraclass distance (Sw) ratio | | | Average classification accuracy |
|---|---|---|---|---|---|
| | | Feature 1 | Feature 2 | Feature 3 | |
| 0.5 | 0 | 0.384 | 0.383 | 0.0017 | 49% |
| 0.8 | 1 | 0.814 | 1.699 | 1.656 | 82% |
| 1 | 5 | 0.669 | 1.556 | 0.631 | 81% |
| 1.5 | 23 | 0.360 | 0.886 | 0.483 | 49% |

**Table 3** Confusion matrices: average classifier performances classifier results.

| | | GED | | | SVM | | | kNN ($k = 3$) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| | 1 | 87% | 7% | 7% | 89% | 5% | 5% | 79% | 11% | 10% |
| Actual Class | 2 | 14% | 68% | 18% | 13% | 73% | 14% | 16% | 67% | 17% |
| | 3 | 2% | 13% | 85% | 1% | 20% | 79% | 2% | 21% | 77% |

of each feature space. The resulting performance is tabulated in Table 2. For n>1, the number of failed clips increase dramatically and thus is not recommended. $n = 0.8$ has the highest inter- to intraclass ratio, meaning that the classes are densely compact and have good separation; thus, $n = 0.8$ was chosen as the preferred threshold value.

### 5.1.2 Segmentation and bud isolation

Figure 7 shows the result of the segmentation and bud isolation on several cell-clips. In general, the segmented shapes closely represent the actual cell shape, although the bud sizes are slightly enlarged due to the segmentation algorithm. The segmentation algorithm is not completely fail-proof: an image with a very large gap (4th clip in Fig. 7) in the cell boundary cannot be closed completely, thus resulting in the algorithm returning an empty shape. The bud isolation algorithm overestimates the size of the parent cell, but since the area of a circle is proportional to the square of the radius, the overestimation is fairly consistent even with different bud sizes. This overestimation is introduced purposely to attempt to negate the enlargement of the bud during the segmentation process, and to make sure that small defects on the boundary are not identified as buds.

### 5.1.3 Computational complexity

The entire algorithm is coded in MATLAB using as many built-in image analysis functions as possible. On average, for each $100 \times 100$ cell clip, the image enhancement algorithm needs a computational time of 10 min per clip, the image segmentation algorithm needs 90 min per clip, and the feature extraction algorithm needs 50 min per clip, of which 40 min is dedicated to

the bud isolation process. Thus, a total of 150 min is needed to perform every image analysis and feature extract procedures.

### 5.2 Optimal Feature Space and Classifier

For each classification method, 120 cells, 40 from each class, were randomly chosen to form a training set and the classifier is obtained using the training set. Then, the classifier is used to classify the other 120 cells, the results of which are compared with their actual class labels below. For cross-validation, each classifier is tested 10 times in each set of feature spaces by selecting 10 random training and testing sets from the original data set. Table 3 shows the confusion matrices of the three classifiers averaged over different feature spaces, and the average classification accuracies of each classifier for each feature space are shown in Table 4.

The performances for different feature sets show that set 2 results in the most accurate classification. Set 1 is inaccurate because it is 1D in nature, making it difficult to separate overlapping classes. Set 3 results in similar accuracy to set 2.

The performances of different classifiers are very similar, with kNN having slightly lower accuracy compared to the other methods. All three methods can classify samples in class 1 with near 90% accuracy; this was expected, since good separation was observed between class 1 and class 2/3 in most feature spaces.

The Mahalanobis distance method missclassifies many of the class 2 samples. This is mainly due to the inaccuracy in the covariance estimation of the classifier, since class 1 and class 3 have more outliers compared to class 2. The support vector machine classifier is less prone to outliers compared to the Mahalanobis distance method. It can classify class 2 samples more accurately. However, it is still inaccurate when classifying between class 2 and 3. This is expected because these two classes have major overlaps.

**Table 4** Summary of classifier accuracies.

| | Average accuracy of different classifiers | | |
|---|---|---|---|
| | GED | SVM | kNN($k = 3$) |
| Feature set 1 (1D features) | 79% | 78% | 73% |
| Feature set 2 (2D features) | 81% | 82% | 77% |
| Feature set 3 (3D features) | 80% | 81% | 73% |

**Table 5** Computational speeds of different classifiers.

| | Classification time (per sample) (ms) |
|---|---|
| Mahalanobis | 2 |
| SVM | 0.7 |
| kNN ($k = 3$) | 20 |
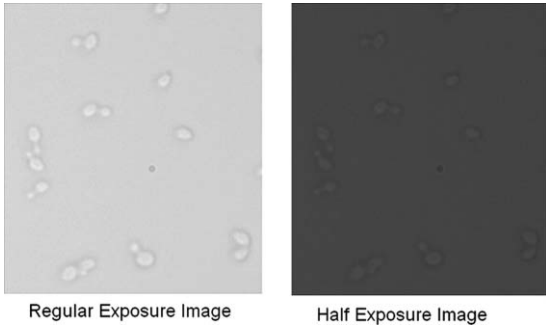
Fig. 8 Cell images under different exposure setting (images brightened to shown dark content).

kNN was expected to be the most accurate classifier overall, since kNN does not assume any class/boundary shape. However, the results do not show any improvement in accuracy, in fact the kNN results are slightly worse than the Mahalanobis distance method and SVM. The likely reason for the poor performance is the lack of data points. If more data points were available, the actual shape of the class would be shown more clearly.

Table 5 shows the computational speeds of each classifier algorithm, not including the image analysis. Although kNN requires a much longer time (20 ms) to make the classification decision compared to SVM and Mahalanobis distance, it will not be the bottleneck of the entire image analysis system, since the image enhancement and segmentation require a much longer computational time.

Based on these results, it can be concluded that feature Set #2, which uses the axis ratio and compactness feature space, and compactness and bud size feature space sequentially, is the optimal feature set for this application. Regarding the most optimal classifier, any one of them could be chosen since they all demonstrate very similar performances. The author suggests using the support vector machine since it has the fastest speed, and if abundant training data is available, use kth nearest neighbor since it does not make assumptions on class or boundary shapes.

## 5.3 System Performance Under Different Conditions

### 5.3.1 Effect of intensity

In addition to the original set of cell clips, another set of clips were taken with half the exposure setting as the original, as shown in Fig. 8.

The image enhancement and segmentation algorithms were performed on these clips with the same parameters, and then classified using the kNN classifier obtained using the original clips as a training set. A comparison of the image enhancement results for full exposure and half exposure settings is shown in
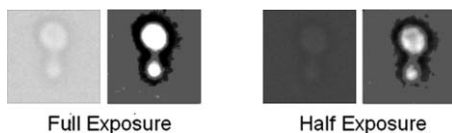
**Table 6** Performance of algorithm under half exposure setting

| Fail to extract feature | Inter-to-intra class ratio | | | Average kNN accuracy |
|---|---|---|---|---|
| | Feature 1 | Feature 2 | Feature 3 | |
| 0 | 0.610 | 1.521 | 1.134 | 67% |

Fig. 9, and the performances of the algorithms for half exposure setting are shown in Table 6.

It was noticed that the enhanced image under half exposure conditions has a lower contrast compared to the full exposure image, since the unenhanced image already has a lower contrast due to insufficient exposure.

At the low exposure setting the inter to intraclass ratios are slightly lower to those with regular exposure settings, due to lower contrast and high noise ratio. Despite this, the image enhancement and segmentation algorithms were able to extract meaningful features from the cell clips. The kNN classification accuracy is lower than for the properly exposed images (67% accuracy compared to 77% in Table 4). These observations show that the higher signal-to-noise ratio of low exposure setting has a negative impact on the performance of the system.

### 5.3.2 Effect of focusing

Due to the ellipsoidal shape and refractive index difference between the cell content and surrounding liquid media, yeast cells act similar to convex lenses and have significantly different appearances when observed in-focus and out-of-focus.[26] The original cell images are taken with the microscope lens focused near the foci of the cells (about 10 $\mu$m from the actual cell position), resulting in the bright spot in the middle of the cell.[26] This focus setting is referred to as standard focus, and has many optical advantages. It results in bigger cell sizes, a more circular cell shape, and higher contrast between the cells and the background. To test the performance of the imaging algorithms under different focusing conditions, two more sets of cell clips were taken. One set is taken at 5 $\mu$m further away from the cell (overfocus), and the second set is taken with the microscope focused on the cell (underfocus, 10 $\mu$m from the original focal plane) resulting in the images shown in Fig. 10.

The overfocused cells generally look larger than the original, especially the buds. The cell membranes are also blurry. The overfocused condition is not an ideal environment for both manual and automated classification, as the bud information is



Fig. 9 Image enhancement under different exposure.
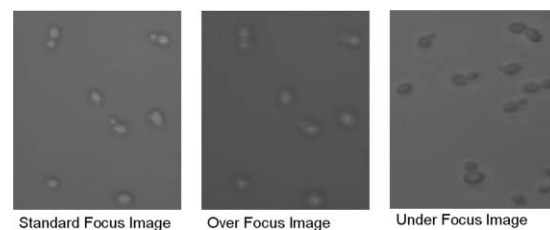


Fig. 10 Appearance of cells under different focal setting.
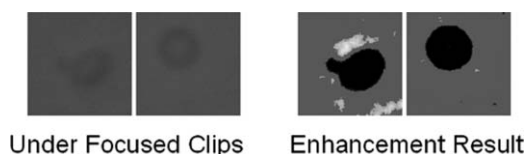
**Fig. 11** Enhancement result for under focused images.

difficult to infer from a blurry image. Some preliminary testing using the algorithm has also shown difficulty to obtain bud size. In general, it is not recommended to use the current algorithm to identify overfocused images.

The underfocused cells generally look darker, and do not show a bright interior spot. This appearance is very different to the standard focused cells. The image enhancement algorithm can still improve the contrast of the cell clips, shown in Fig. 11, but the image segmentation algorithm, which tries to find an interior boundary of the darker regions, fails completely. However, the enhanced images can easily be segmented using a different algorithm. Then the feature extraction and classification algorithms can still be used for underfocused cells.

## 6 Conclusions

In this paper a complete image processing and classification algorithm for yeast cell morphology is presented. The algorithm contains four sequential processes, first an image enhancement process that removes background noise and improves image quality, then a segmentation algorithm that converts the images to a binary matrix that contain the cell shape, followed by feature extraction methods, and a self-learning classifier that distinguishes between nonbudding, small bud, and large bud cells. This algorithm is more consistent compared to manual classification, and is fully automatic. The operator only needs to provide it with labeled training data. During this study the training data were labeled manually by a single individual, however, less-subjective labeling methods can be used for better training.

It was found that the image enhancement algorithm is capable of removing the effect of uneven illumination and sensor noise. The most accurate and efficient feature space for classification between budding and nonbudding cells is the axis ratio and compactness feature space, while the compactness and bud ratio feature space is capable of classifying between cells with small and large buds. The class shapes in these feature spaces are not Gaussian due to the number of outliers, and none of the classes are linearly separable; the three classification methods tested: kNN, SVM, and Mahalanobis distance all showed similar accuracy, in which kNN being the most versatile classifier, and SVM has the fastest processing time.

The algorithm was also tested under different illumination and focusing settings, and it was found that under low exposure settings the higher noise results in lower accuracy. The system can tolerate a slight variation in focusing, but for underfocused images a different segmentation algorithm needs to be implemented for accurate classification.

### References

1. B. P. Ingalls, B. P. Duncker, and B. J. McConkey, "Systems level modeling of the cell cycle using budding yeast," *Cancer Information* **3**, 357–370 (2007).
2. K. C. Chen, L. Calzone, A. Csikasz-Nagy, F. R. Cross, B. Novak, and J. J. Tyson, "Integrative analysis of cell cycle control in budding yeast," *Mol. Biol. Cell* **15**(8), 3841–3862 (2004).
3. F. V. D. Heijden and (Firm) Knovel, "Classification, parameter estimation, and state estimation an engineering approach using MATLAB," Chap 16, p. 423 Wiley, New York (2004).
4. I. Herskowitz, "Life-cycle of the budding yeast Saccharomyces-Cerevisiae," *Microbiol. Rev.* **52**(4), 536–553 (1988).
5. J. M. Sidorova and L. L. Breeden, "Precocious G1/S transitions and genomic instability: the origin connection," *Mutat. Research-Fundamental and Molecular Mechanisms of Mutagenesis* **532**(1–2), 5–19 (2003).
6. S. Ohnuki, S. Nogami, and Y. Ohya, "A microfluidic device to acquire high-magnification microphotographs of yeast cells," *Cell Div* **4**, 5 (2009).
7. J. Koschwanez, M. Holl, B. Marquardt, J. Dragavon, L. Burgess, and D. Meldrum, "Identification of budding yeast using a fiber-optic imaging bundle," *Rev. Sci. Instrum.* **75**(5), 1363–1365 (2004).
8. M. Ohtani, A. Saka, F. Sano, Y. Ohya, and S. Morishita, "Development of image processing program for yeast cell morphology," *J. Bioinf. Comput. Biol.* **1**(4), 695–709 (2004).
9. P. J. Lee, N. C. Helman, W. A. Lim, and P. J. Hung, "A microfluidic system for dynamic yeast cell imaging," *Biotechniques* **44**(1), 91–95 (2008).
10. B. Bayraktar, P. P. Banada, E. D. Hirleman, A. K. Bhunia, J. P. Robinson, and B. Rajwa, "Feature extraction from light-scatter patterns of Listeria colonies for identification and classification," *J. Biomed. Opt.* **11**(3), 034006 (2006).
11. B. Javidi, I. Moon, and S. Yeom, "Three-dimensional identification of biological microorganism using integral imaging," *Opt. Express* **14**(25), 12096–12108 (2006).
12. S. B. Gokturk, C. Tomasi, B. Acar, C. F. Beaulieu, D. S. Paik, R. B. Jeffrey Jr., J. Yee, and S. Napel, "A statistical 3-D pattern processing method for computer-aided detection of polyps in CT colonography," *IEEE Trans. Med. Imaging* **20**(12), 1251–1260 (2001).
13. S. Srivastava, J. J. Rodríguez, A. R. Rouse, M. A. Brewer, and A. F. Gmitro, "Computer-aided identification of ovarian cancer in confocal microendoscope images," *J. Biomed. Opt.* **13**(2), 024021 (2008).
14. A. Bazzani, A. Bevilacqua, D. Bollini, R. Brancaccio, R. Campanini, N. Lanconelli, A. Riccardi, and D. Romani, "An SVM classifier to separate false signals from microcalcifications in digital mammograms," *Phys. Med. Biol.* **46**(6), 1651–1663 (2001).
15. O. Dzyubachyk, W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering, "Advanced level-set-based cell tracking in time-lapse fluorescence microscopy," *IEEE Trans. Med. Imaging* **29**(6), 1331–1331 (2010).
16. C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, and J. C. Olivo-Marin, "Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing," *IEEE Trans. Med. Imaging* **21**(10), 1212–1221 (2002).
17. X. Wang, W. He, and D. Metaxas, "Cell segmentation and tracking using texture-adaptive snakes," *4th IEEE Int. Symp. on Biomedical Imaging: Macro to Nano (ISBI 2007)*, Washington, DC, Vol. 1(3), pp. 101–104 (2007).
18. D. Padfield, J. Rittscher, N. Thomas, and B. Roysam, "Spatio-temporal cell cycle phase analysis using level sets and fast marching methods," *Med. Image Anal.* **13**(1), 143–155 (2009).
19. A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillen-Aghion, J. C. Olivo-Marin, and C. Zimmer, "Segmenting and tracking fluorescent cells in

dynamic 3-D microscopy with coupled active surfaces," *IEEE Trans. Image Process.* **14**(9), 1396–1410 (2005).

20. D. P. Mukherjee, N. Ray, and S. T. Acton, "Level set analysis for leukocyte detection and tracking," *IEEE Trans. Image Process.* **13**(4), 562–572 (2004).

21. D. L. Pham, C. Y. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Annu. Rev. Biomed. Eng.* **2**, 315–337 (2000).

22. A. Niemistö, M. Nykter, T. Aho, H. Jalovaara, K. Marjanen, M. Ahdesmäki, P. Ruusuvuori, M. Tiainen, M.-L. Linne, and O. Yli-Harja, "Computational methods for estimation of cell cycle phase distributions of yeast cells," *EURASIP J. Bioinform. Syst. Biol.* **2007**, 46150 (2007).

23. B. Belaroussi, J. Milles, S. Carme, Y. M. Zhu, and H. Benoit-Cattin, "Intensity nonuniformity correction in MRI: Existing methods and their validation," *Med. Image Anal.* **10**(2), 234–246 (2006).

24. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed., p. 954, Pearson/Prentice Hall, Upper Saddle River, NJ (2008).

25. M. Graña and R. J. Duro, "Computational intelligence for remote sensing," in *Studies in Computational Intelligence*, p. 133, Springer, Berlin (2008).

26. C. Bittner, G. Wehnert, and T. Scheper, "In situ microscopy for on-line determination of biomass," *Biotechnol. Bioeng.* **60**(1) 24–35 (1998).