

Automatic Cell Detection in Bright-Field Microscope Images Using SIFT, Random Forests, and Hierarchical Clustering

Firas Mualla¹, Simon Schöll^{1,2,3}, Björn Sommerfeldt⁴, Andreas Maier^{1,3}, and Joachim Hornegger, *Member, IEEE*^{1,3}

¹Pattern Recognition Lab, Friedrich-Alexander University Erlangen-Nuremberg

²ASTRUM IT GmbH, Erlangen

³SAOT Graduate School in Advanced Optical Technologies

⁴Institute of Bioprocess Engineering, Friedrich-Alexander University Erlangen-Nuremberg

Abstract—We present a novel machine learning-based system for unstained cell detection in bright-field microscope images. The system is fully automatic since it requires no manual parameter tuning. It is also highly invariant with respect to illumination conditions and to the size and orientation of cells. Images from two adherent cell lines and one suspension cell line were used in the evaluation for a total number of more than 3500 cells. Besides real images, simulated images were also used in the evaluation. The detection error was between approximately zero and 15.5% which is a significantly superior performance compared to baseline approaches.

1. INTRODUCTION

Though fluorescence microscopy is widely used, many applications require the use of unstained cell imaging. One of motivating factors is that the signal obtained by fluorescence microscopy cannot be guaranteed to cover the entire cell, and therefore it is not suited for cell boundary segmentation [2]. Such shape information is important when studying cellular processes like mitosis and apoptosis [35]. Furthermore, the fluorescent dyes are sometimes considered non-neutral or even toxic. For instance, cell-tracing dyes used in cell viability identification and cytotoxicity tests dramatically change the cell stiffness and the cell-to-probe adhesion [24]. Phase-contrast microscopy and bright-field microscopy can be used for unstained cell imaging. Compared to phase-contrast, bright-field imaging does not need a specialized hardware and it is thus cheaper and easier to implement [9].

Unstained cell recognition in bright-field images is a challenging problem [20], [21], [36], [38], [42]. Cells exhibit a great diversity in shape and size. In addition, a simple microscopy technique like bright-field does not always offer sufficient contrast between cells and background [8], [36]. In fact, adherent cells are almost invisible at focus [1], [2], [4]. The contrast can be improved by defocusing the microscope. An image is considered positively defocused when the objective approaches the object and negative otherwise [1].

In [4], a positively defocused image was segmented by applying the watershed algorithm on the distance transform of a thresholded image. An in-focus image was processed with the self-complementary top-hat followed by thresholding and watershed. A negatively defocused image was preprocessed with a Canny edge detector before analyzed by anisotropic contour completion [10]. These three results were then combined in order to select micro-injection points inside the cells.

In [3], a positively defocused image was subtracted from a negatively defocused one. The difference image was thresholded and the result was post-processed with an appropriately chosen size filter. Each connected component of the resulting mask was used to initialize a level-set evolution.

The aforementioned techniques are based on heuristics. In contrast, there is a family of methods for cell detection that are learning-based. In these approaches, a classifier labels each pixel as either a cell or a non-cell pixel. The output of the pixel-wise classifier delivers a confidence map. The maxima of the latter correspond to cell centers.

For training the classifier, features are extracted from a fixed-size patch sampled in the neighborhood of the pixel under investigation. Several papers share this common strategy despite the differences in the classifier model and the image modality. For example, [31] used principal component analysis (PCA) features extracted from 15 x 15 sized patches which are then analyzed by an artificial neural network. In [20], Fisher discriminant analysis was used instead of PCA. In [21], a support vector machine (SVM) replaced the neural network. In [41], a bag of local Bayesian classifiers was used, each of them trained on a cluster of the training data.

Most of these learning-based approaches require parameter tuning. These parameters are related to thresholding the confidence map, applying morphological operators on the thresholding result, and/or searching for the maxima of the confidence map. In all mentioned learning-based approaches, the optimal size of the patch should conform to the mean cell size which is not always known a priori. In addition, the square neighborhood used in these methods, does not fit non-circular cells.

We think that a *good* cell detection system should fulfill

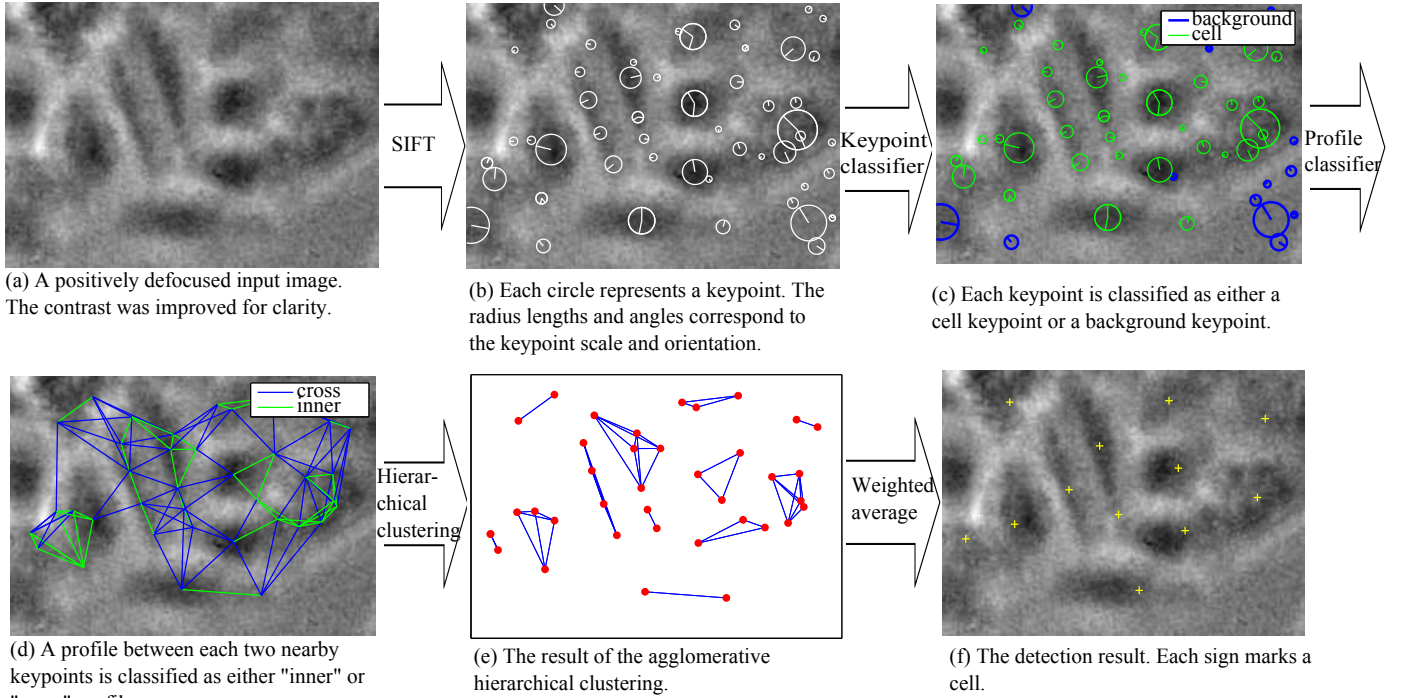


Fig. 1. A general overview of our automatic cell detection approach.

the following criteria: a) It should be invariant to cell size and orientation. b) It should be invariant to illumination conditions. c) It should be automatic, i.e. it does not need manual parameter tuning. d) If tracking is required after detection, then the detected cells should be equipped with trackable features. e) Lastly, the degradation of the detection rate because of the previous invariance requirements should be minimal. With these criteria in mind, we developed a learning-based system for automatic unstained cell detection in bright-field images.

We tested the system on three cell-lines: adherent CHO, adherent L929, and Sf21 in suspension. The adherent cells had very low contrast and were almost invisible at focus. In addition, a cell simulation software [17] was used to generate synthetic images with noise and illumination artifacts. All analyses on the real and the simulated images were done without any manual parameter tuning. The only difference between these experiments was in the training data.

A. System overview

A cell is expressed in bright-field images as one or more *blobs* in intensity. A blob is a maximum of the normalized Laplacian in scale space [19]. It is also called an interest point or a keypoint. SIFT [22] is a class of local image features which characterize the neighborhood of each keypoint in a scale- and orientation-independent way.

Our proposed cell detection algorithm is sketched in Figure 1. The starting point is the extraction of SIFT keypoints from a defocused image. These keypoints are classified into cell keypoints and background keypoints. For each cell keypoint, an intensity profile to each *nearby* cell keypoint is extracted and classified into either *inner* or *cross* profile. A

profile between two keypoints is called *inner* if these two keypoints belong to the same cell. Otherwise, it is called *cross* profile. The output of the profile classifier is probabilistic. The probability that a profile between two keypoints is an inner profile can be seen as a similarity measure between the keypoints. Based on this similarity measure, an agglomerative hierarchical clustering of the keypoints with a customized linkage method is applied. A weighted mean of the keypoint coordinates inside each cluster marks the detected cell center.

No manual tuning of any parameters is required during training or detection. The system learns its parameters automatically in a scale- and orientation-invariant manner.

The rest of the paper is organized as follows: Section 2 describes the features and the classifier model of the keypoint classifier. Section 3 discusses the different aspects of the profile learning. In Section 4, the hierarchical clustering step is explained. Section 5 provides details of the training phase. The detection quality measures are defined in Section 6. Section 7 contains the results. The paper ends with a discussion in Section 8 and a conclusion in Section 9.

2. KEYPOINT LEARNING

In order to determine, whether the cells in the training data tend to have a bright or a dark appearance, we perform a calibration step before the actual cell detection procedure. In the remainder of this paper, we call a set of keypoints *one-sided* if all the keypoints in the set have positive difference-of-Gaussians (DOG) values or all of them have negative DOG values. In order to automatically detect the keypoint type which fits the training data, the system uses the following measure:

$$\text{PositiveFit} = \frac{\sum_{i=1}^{N_C} s(\mathbf{p}_i) |DOG(\mathbf{p}_i)| H(DOG(\mathbf{p}_i))}{\sum_{i=1}^{N_C} s(\mathbf{p}_i) |DOG(\mathbf{p}_i)|} \quad (1)$$

where $\mathbf{p}_i, i = 1..N_C$ are the cell keypoints in the training data, N_C is their number, s is the scale, and H is the Heaviside step function. If PositiveFit evaluates to more than 0.5, the system ignores keypoints with negative DOG values during training and detection. Otherwise, it ignores keypoints with positive DOG values.

After that, the maximum $|DOG|$ in each training cell is computed and the first percentile of all the previous maxima is considered a SIFT threshold. During training and detection, SIFT keypoints which have lower $|DOG|$ than this threshold will be discarded.

In order to separate cell keypoints from non-cell keypoints, we used several sets of features from the literature and adjusted them. We utilized SIFT to make these features scale- and orientation-invariant. We also made them invariant to local shift of intensity. The importance of invariance to the local shift of intensity is discussed in Section 8.

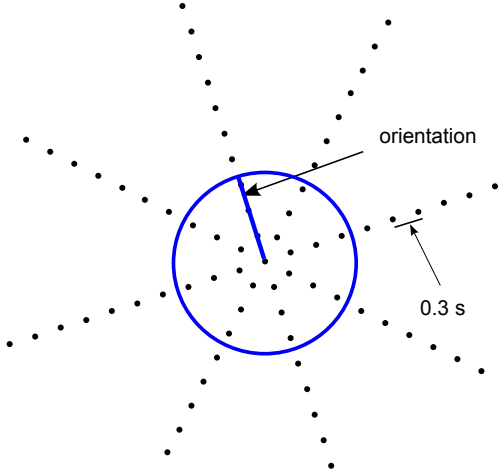


Fig. 2. Radial intensity stencil. The stencil is aligned with the keypoint orientation. The distance between two successive nodes is $0.3s$, where s is the keypoint scale.

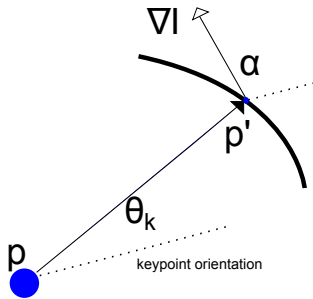


Fig. 3. Ray features adjusted to keypoints. This figure is an adapted version of the pixel-based ray features in [34].

More specifically, the following set of features was extracted for each keypoint:

- 1) Intensity stencil: We computed intensity stencils [12], [26] around each keypoint. As shown in Figure 2, we align the stencil with the keypoint orientation and measure the distance between the sampling points in units of scale instead of pixels. The result is a scale- and orientation-invariant stencil. In order to make the stencil invariant to the local shift of intensity, we subtract the mean intensity of the stencil from all stencil nodes. The intensity values at the stencil nodes after the aforementioned subtraction form the stencil feature set.
- 2) Ray features: In order to compute ray features [34] at a keypoint \mathbf{p} (cf. Figure 3) in an image \mathbf{I} , the closest edge point \mathbf{p}' along a direction θ_k is found. Ray features according to [34] are four sets of features. The first three are: the distance $R_d(\mathbf{p}, \theta_k)$ between \mathbf{p} and \mathbf{p}' , the gradient norm $R_n(\mathbf{p}, \theta_k)$ at \mathbf{p}' , and the gradient angle $\alpha = R_a(\mathbf{p}, \theta_k)$ at \mathbf{p}' . If eight values of the angle θ are used, i.e. $k = 1..8$, we obtain 24 features. The fourth set is the distance difference $R_{dd}(\mathbf{p}, \theta_k, \theta_{k'}) = |R_d(\mathbf{p}, \theta_k) - R_d(\mathbf{p}, \theta_{k'})|$. Therefore, for 8 angles, the fourth feature set is $8 \cdot 7/2 = 28$ features.

Ray features are well-designed but are sensitive to scale and orientation. In order to make them orientation-invariant, we define all angles, i.e. the eight θ_k angles and the gradient angle feature $R_a(\mathbf{p}, \theta_k)$, with respect to the keypoint orientation.

In order to make the ray features scale-invariant, we measure the distances $R_d(\mathbf{p}, \theta_k)$ and $R_{dd}(\mathbf{p}, \theta_k, \theta_{k'})$ in terms of scale. Furthermore, we compute the gradient using the following equation for its x component:

$$\frac{\partial \mathbf{I}(\mathbf{p}')}{\partial x} = \mathbf{I}(p'_x + \tau s(\mathbf{p}), p'_y) - \mathbf{I}(p'_x, p'_y) \quad (2)$$

where $s(\mathbf{p})$ is the scale of the keypoint at \mathbf{p} . τ is a constant that we set to 1. A similar equation is used for the y component. Before applying Eq. (2), the image is smoothed using a Gaussian kernel with a standard deviation equal to 1.

The edges are obtained using Canny edge detection [6]. The thresholds were set to the default values in the Matlab implementation of Canny.

- 3) Variance map: Based on the variance map [40], we created a keypoint-based scale-invariant version by taking a variable-size neighborhood with a size proportional to the keypoint scale. For each keypoint \mathbf{p} , we extract three variance map features $\text{VMap}(\mathbf{p}, 2)$, $\text{VMap}(\mathbf{p}, 4)$, and $\text{VMap}(\mathbf{p}, 6)$ which correspond to variance map in a square neighborhood of side length $2s(\mathbf{p})$, $4s(\mathbf{p})$, and $6s(\mathbf{p})$, respectively. The map is by construction invariant to the local shift of intensity.
- 4) SIFT descriptors: SIFT features were used according to the original publication [22], as they are inherently scale- and orientation-invariant and partially illumination-invariant.
- 5) Other features: The values of the DOG and the principal curvatures ratio (PCR) at each keypoint were also obtained from SIFT.

We chose random forest (RF) [5] as a background-cell classifier. It is an ensemble of classifiers, each of which is a decision tree trained on a bootstrap sample of the training data. During training, splits are based on a feature at each node. In contrast to conventional decision trees, only a random subset of the features is considered at each node of a RF tree. In order to be able to deal with imbalanced class labels, we chose a balanced random forest (BRF) [7]. Similar to [14], we set the number of trees to 500 and the number of the randomly selected features to $F_N/5$, where F_N is the number of features.

3. PROFILE LEARNING

The keypoints which were classified as background by the keypoint classifier are discarded. The remaining ones are cell keypoints. In order to resolve whether two cell keypoints belong to the same cell, we extract an intensity profile $f(v)$ between them, where $v = 1..N_F$ are the sampling points along the profile, and N_F is the number of the points. We then extract profile features and use them to classify the profile into either *inner* or *cross* profile.

The following features are extracted for each intensity profile $f(v)$:

- The standard deviation, the skewness, and the kurtosis of the profile.
- The standard deviation, the skewness, the kurtosis, the maximum, the minimum, and the mean of the first derivative df/dv and the second derivative d^2f/dv^2 of the profile.
- Two other features:

$$V_1 = \max(f) - \min(f) \quad (3)$$

$$V_2 = f(1) - 2\max(f) + f(N_F) \quad (4)$$

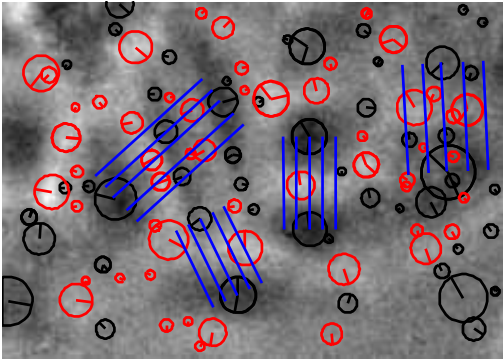


Fig. 4. Profile sets between different keypoints.

The profiles are sampled using a fixed number of points $N_F = 50$. The derivatives are Gaussian derivatives with $\sigma = 0.1N_F$.

As a classifier model, we use again BRF.

For the training of the profile classifier and during the detection phase, the algorithm extracts a profile between two keypoints only if they are *nearby*. In order to achieve this goal in a scale-invariant manner, the algorithm learns the maximum inner profile length from the training data.

Naturally, it is inappropriate to measure the profile length in pixels. We use the average scale $U(\mathbf{I})$ of all cell keypoints inside the image as a profile length unit for this image.

During training, the maximum inner profile length L_k in each training image \mathbf{I}_k is computed in terms of $U(\mathbf{I}_k)$. Then the maximum of all the L_k values is considered the maximum inner profile length in the training data, $L = \max(L_k), k = 1..N_t$, where N_t is the number of training images.

Given a cell keypoint \mathbf{p}_1 in an image \mathbf{I} , another cell keypoint \mathbf{p}_2 in \mathbf{I} is considered *nearby* if the Euclidean distance between the two keypoints, in units of $U(\mathbf{I})$, is smaller than ζL . ζ is a safety parameter that we set to 2.

The image area which is sampled by one intensity profile is actually very small. It is thus plausible to expect an improvement in the detection accuracy when the profile captures information from a wider image area.

Instead of extracting one profile between the two considered keypoints, one could extract a set of profiles, i.e. several parallel profiles as demonstrated in Figure 4. The geometry of the set can be described, for instance, in terms of the maximum scale of the two keypoints. In Figure 4, the profile set contains five parallel profiles, one profile every $0.75s_{ij}^{max}$. s_{ij}^{max} is the maximum scale of the two considered keypoints \mathbf{p}_i and \mathbf{p}_j .

Alternatively, smoothing is a well-known fast and simple means of information consolidation. We make smoothing *scale adaptive* by setting the standard deviation of the Gaussian kernel which is used to smooth an image \mathbf{I} to the mean scale of the one-sided keypoints of \mathbf{I} . We call this process scale adaptive smoothing (SAS). A third approach is to combine the previous two ones. In our algorithm, the SAS was used.

If the number of the inner profiles in the training data is too small, the algorithm extracts artificial inner profiles until the number of the inner profiles in the training data is at least N_{inner} . If the number of the cross profiles in the training data is too small, the algorithm generates artificial cross profiles until the number of the cross profiles in the training data is at least N_{cross} . They are generated by shifting the considered training image \mathbf{I} by $\beta U(\mathbf{I})$ and then overlaying the original and the shifted version. β was set to 6. Both N_{inner} and N_{cross} were set to 15 in our experiments.

4. HIERARCHICAL CLUSTERING

In this section, we address the problem of combining the results of the keypoint learning and the profile learning in order to detect cells. One could employ a graph-based approach: Assume a graph G with the cell keypoints as nodes. Two nodes are connected if the profile between them is an inner profile. The nodes of G are obtained from the keypoint classification and the edges are obtained from the profile classification. Intuitively, each connected component in G can be seen as a detected cell. This technique will be referred to later in this paper as the connected components (CC).

The utilization of the available information using CC is suboptimal. Alternatively, one can think of clustering the keypoints in an agglomerative manner starting from the most reliable ones: the agglomerative hierarchical clustering (AHC) of the keypoints. The AHC is characterized by a similarity

measure and a linkage method. Two cell keypoints \mathbf{p}_i and \mathbf{p}_j are similar if they belong to the same cell. Thus, it is plausible to define the similarity between \mathbf{p}_i and \mathbf{p}_j as the probability that the profile between them is an inner profile $P_{inner}(\mathbf{p}_i, \mathbf{p}_j)$. This is equivalent to defining $P_{cross}(\mathbf{p}_i, \mathbf{p}_j) = 1 - P_{inner}(\mathbf{p}_i, \mathbf{p}_j)$ as a dissimilarity measure between the two cell keypoints.

A. Linkage method

One plausible choice for the linkage method is the group average link. In order to incorporate more information, we use a customized group average linkage instead of the traditional one. According to this customized linkage, the dissimilarity between two clusters A and B is:

$$\Phi(A, B) = \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} \frac{\omega_{ij}}{\Omega_{AB}} \phi(\mathbf{p}_i, \mathbf{p}_j) \quad (5)$$

$$\phi(\mathbf{p}_i, \mathbf{p}_j) = P_{cross}(\mathbf{p}_i, \mathbf{p}_j) \quad (6)$$

$$\omega_{ij} = \frac{|DOG(\mathbf{p}_i)|s(\mathbf{p}_i)|DOG(\mathbf{p}_j)|s(\mathbf{p}_j)}{\|\mathbf{p}_i - \mathbf{p}_j\|^2} \quad (7)$$

$$\Omega_{AB} = \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} \omega_{ij} \quad (8)$$

where \mathbf{p}_i is a keypoint in the cluster A . \mathbf{p}_j is a keypoint in the cluster B . N_A and N_B are the cardinalities of A and B , respectively. ω_{ij} is the weight of the dissimilarity between \mathbf{p}_i and \mathbf{p}_j . This weight is proportional to the scale and the absolute DOG of the two points and inversely proportional to the squared Euclidean norm of the profile $\|\mathbf{p}_i - \mathbf{p}_j\|^2$. Indeed, ω_{ij} is scale-invariant because the scale and the distance terms in the numerator and the denominator have the same order. Ω_{AB} is the normalization factor of the weights.

The obtained dissimilarity from this customized linkage is always in the range $[0, 1]$. The final clusters are obtained by cutting the dendrogram at a cutoff equal to 0.5.

In the so-called combinatorial clustering strategies, the new dissimilarities can be computed from the old ones using the Lance-Williams dissimilarity update formula [15], [16]: If A and B are merged into one cluster AB , then the dissimilarity between any other cluster C and the cluster AB is given by:

$$\begin{aligned} \Pi(AB, C) &= \psi\Pi(A, C) + \gamma\Pi(B, C) + \\ &\nu\Pi(A, B) + \mu|\Pi(A, C) - \Pi(B, C)| \end{aligned} \quad (9)$$

where ψ , γ , ν , and μ are the coefficients of this linear model. The values of the coefficients are known for the standard linkage methods like single, complete, average, median, centroid, Ward, and McQuitty [30]. As we use a customized linkage, we need to find the corresponding coefficient values. It can be shown that the model coefficients for our customized linkage are:

$$\psi = \frac{\Omega_{AC}}{\Omega_{AC} + \Omega_{BC}} \quad (10)$$

$$\gamma = \frac{\Omega_{BC}}{\Omega_{AC} + \Omega_{BC}} \quad (11)$$

$$\nu = \mu = 0 \quad (12)$$

It is desirable to have monotonic clustering strategies as the reversals caused by non-monotonic strategies are inconvenient and hard to interpret [27], [29]. When μ is zero, the linear model $\Pi(AB, C)$ is monotonic under the condition $\psi + \gamma + \nu \geq 1$ [16]. Obviously, this is fulfilled by our customized linkage and the trees built by Φ in Eq. (5) are thus monotonic.

B. Finding the hit-point

Applying the hierarchical clustering will result in different clusters of keypoints. Each cluster represents one cell. In order to determine a single hit-point, we use the following equation:

$$\mathbf{p}_h = \frac{1}{\sum_{j=1}^{N_L} \lambda_j} \sum_{i=1}^{N_L} \lambda_i \mathbf{p}_i \quad (13)$$

where N_L is the number of keypoints inside the considered cluster. \mathbf{p}_i is the keypoint i in the cluster. λ_i is a reliability weight that we set to $\lambda_i = |DOG(\mathbf{p}_i)|$.

5. SYSTEM TRAINING

Equipped with the previously described methods, one is now able to train the proposed system from a given set of images and their cell segmentation (ground truth). First, the system is calibrated to detect bright or dark cells according to Eq. (1). Depending on the result of this step, the keypoints with positive or negative DOG values will be discarded. After that, a SIFT threshold is learned from the training data and then applied as described in Section 2. Using the remaining set of keypoints, the mean image scale is computed and SAS is applied. Next, keypoints are detected in all training images and their respective features are extracted. Based on the segmentation, the class of each keypoint can be determined. Using this information a BRF is trained on the keypoint features. These steps enable the system to automatically extract keypoints and to determine whether they belong to a cell or the background.

Based on the cell keypoints, the profile learning process can then be started. As a first step, we calibrate the maximum inner profile length L from the training images and their segmentation. Next, we train a BRF using the profile features. As classes, we use *inner profile* and *cross profile*. This yields a system that is automatically able to distinguish these two types of profiles.

For the hierarchical clustering, no training is required. The method can be applied directly on the output of the profile BRF.

6. EVALUATION MEASURES

We use the following measures to evaluate the detection quality: precision, recall, detection error, and centeredness error. Precision is strictly defined. Therefore, if several hit-points (cf. Eq. (13)) lie inside the mask of a cell, one of them is considered correct and the others are false positives.

Detection error is the arithmetic average of the precision loss and the recall loss:

$$\text{Detection error}(\mathbf{I}) = \frac{1}{2} \left(\frac{Z^U}{Z} + \frac{H^B + H^O}{H} \right) \quad (14)$$

where H is the total number of the hit-points. H^B is the number of the hit-points in the background. H^O is the number of the over-detected hit-points. For instance, if five hit-points were detected in one cell, then one of them is considered correct and the other four are considered over-detected hit-points. Z is the total number of cells in the image \mathbf{I} . Z^U is the number of the undetected cells, i.e. the cells which contain no hit-points.

Since a hit-point can lie anywhere inside the cell mask, another measure is needed in order to evaluate the centeredness of the hit-point inside the detected cell. For this purpose, we define a new measure:

$$\text{Centeredness error}(\mathbf{I}) = \frac{1}{Z^C} \sum_{i=1}^{Z^C} \frac{\|\mathbf{p}_h^i - \mathbf{p}_m^i\|}{\chi_i} \quad (15)$$

where Z^C is the number of the cells which were detected by only one hit point. Therefore, the over-detected cells are not considered by this measure. The numerator is the Euclidean distance between \mathbf{p}_h^i the hit-point inside the cell i and \mathbf{p}_m^i the center of mass of this cell. The denominator χ_i is the major axis length of the ellipse which represents the covariance matrix of the binary mask of the cell i . This normalization is important in order to make the centeredness error independent of the cell size.

7. RESULTS

A. Materials

We evaluated the system on both real and simulated images. Table 1 shows a summary of the cell lines used for the evaluation.

The first three rows of the table are real cell lines: CHO adherent cells (cf. Figure 5(a)), L929 adherent cells (cf. Figure 5(b)), and Sf21 suspension cells (cf. Figure 5(c)). By the term *adherent cell line*, we mean that almost all cells were adherent. Due to biological reasons, it was not possible to force all cells to adhere. Figure 5(b) exemplifies this case where some cells in the adherent cell line L929 are in suspension.

Two bioprocess engineering experts have manually labeled the cells in the three real cell lines using the LabelMe annotation framework [33]. The total number of the manually labeled cells is 3510. The cell culturing process of these cell lines is explained in Section 7-A1. While the image acquisition details are described in 7-A2.

The last two rows of Table 1 are our simulated images. SIMCEP [17] was used to simulate two cell lines. The first (cf. Figure 5(d)) is simulated with high SNR, while the second (cf. Figure 5(e)) is simulated under severe Gaussian noise conditions. The simulation details are explained in Section 7-A3.

1) *Cell culturing*: CHO-K1 epithelial-like cells and L929 murine fibroblast cells were pre-cultured and maintained in exponential growth phase in T-25 polystyrene culture flasks (Sarsted 8318.10) using DMEM/Ham's F-12 (1:1) (Invitrogen 21331-046) with 10% fetal calf serum (PAA A15-102) and 4 mM Glutamine (Sigma-Aldrich G7513-100ML) at 37°C and 7% CO₂ containing atmosphere. For image acquisition, cells were detached from the T-Flask using Accutase (Sigma-Aldrich A6964-100ML) on the day before and seeded out in

Cell line	Description	Images	Cells
CHO	Real CHO adherent cells	6	1431
L929	Real L929 adherent cells	5	1078
Sf21	Real Sf21 cells in suspension	5	1001
Simulated A	Simulated cells with SNR \approx 63	100	15000
Simulated B	Simulated cells with SNR \approx 0.07	100	15000

TABLE 1
THE CELL LINES USED IN THE EVALUATION

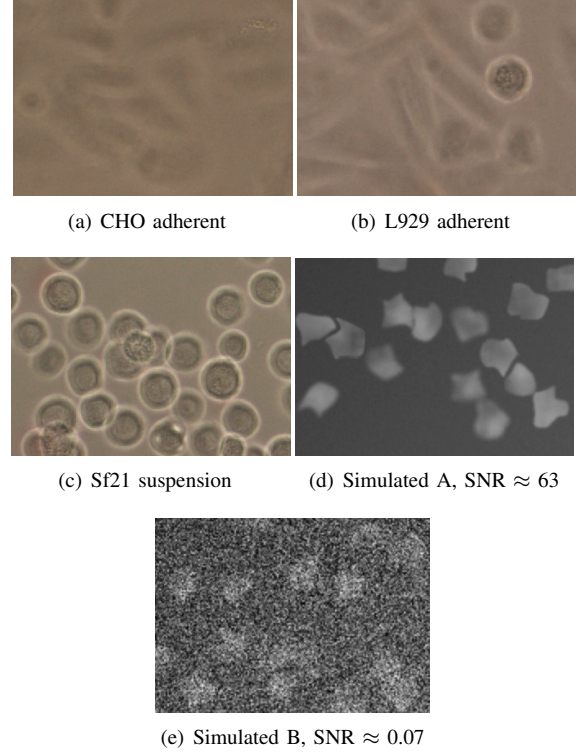


Fig. 5. Examples cut from the evaluation images. The defocus distance in (a), (b), and (c) is $+30\mu\text{m}$, $+30\mu\text{m}$, and $+15\mu\text{m}$, respectively.

24-well plate format using a working volume of 600 μl of the mentioned medium composition. Cells were allowed to attach and spread out in the well-plate for at least 18 hours but not more than 30 hours.

Sf21 insect cells were maintained in exponential growth phase in silicon solution treated shaker flasks in Ex-Cell 420 medium (Sigma-Aldrich: 24420C) at 27°C and normal air CO₂ level. On the day of investigation, an aliquot of this culture was transferred to a 24-well plate in a final volume of 600 μl and cells were allowed to sediment before image acquisition.

2) *Real image acquisition*: The images of the three real cell lines in Table 1 were manually taken with an inverted Nikon Eclipse TE2000U microscope using Nikon's USB camera. Cells were illuminated by a halogen light bulb for standard bright-field microscopy. The used microscope's objective has 20x magnification, 0.45 numerical aperture, and 7.4 mm working distance. The image resolution is 1280 x 960 pixels with 0.49 μm / pixel.

The most important acquisition parameter is probably the defocus distance. This distance was empirically set to $+30\mu\text{m}$ for the adherent cell lines and $+15\mu\text{m}$ for the suspension cell

line. For our approach, one focus level is sufficient. However, we compare our system with other approaches which need images at several focus levels. Therefore, in addition to the previous focus level, we acquired images at levels 0 and $-30\mu\text{m}$ for the adherent cell lines and at levels 0 and $-15\mu\text{m}$ for the suspension cell line.

3) *Image simulation*: As mentioned above, the software in [17] was used to simulate the two artificial cell lines in Table 1. It is important to point out that this software was designed for fluorescence microscopy. Nonetheless, we chose to use it for the following reasons: Firstly, to the best of our knowledge, there is no simulation software available for bright-field microscopy. Secondly, if the cytoplasm is excluded from the simulation, the cell nuclei resemble the negatively defocused bright-field images even though this resemblance is partial due to differences at the cell borders.

Cells were generated according to a shape model described in [17] with dynamic range equal to 0.3 of the allowed image bins number. Then an illumination field with scale 10 was added to each image. This scale is the ratio between the illumination energy (sum of squared values) and the ideal image energy. Afterwards, white Gaussian noise was added so that the signal to noise ratio is approximately 63 in the first cell line (cf. Figure 5(d)) and 0.07 in the second (cf. Figure 5(e)). The SNR is the ratio between the ideal image energy and the noise energy. The illumination energy does not contribute to the SNR.

In Section 7-B, all cell lines of Table 1 were used to evaluate the detection accuracy. The real cell lines were used in Section 7-C to assess the contribution of the different components of our algorithm to the overall detection accuracy. In Section 7-D, we compare our system with two other approaches. For this comparison, the CHO cell line was used first as is, and then perturbed with illumination and scale changes. In Section 7-E, we perturb the three real cell lines with orientation, scale, and illumination changes in order to assess the system's invariance to these factors. Detection time was evaluated in Section 7-F. Finally, the ability of the algorithm to learn to detect cells when the training data contains several cell lines was evaluated in Section 7-G.

B. Evaluation of the overall detection accuracy

We evaluated the detection accuracy of the proposed system on the five cell lines given in Table 1. For each cell line, only one image was used for training and the others were used for testing. For the real cell lines, this was repeated for each image. For the simulated cell lines, this was repeated five times. Table 2 shows the results of this cross validation. The detection quality measures are defined in Section 6

Table 2 shows that the error was close to zero for the high SNR simulated images. Even under severe Gaussian noise conditions with $\text{SNR} \approx 0.07$, the error was only 4 %. It is also clear from the table, that the system achieved higher detection rates with suspension cells compared to adherent cells. This is plausible, as the latter have considerably lower contrast than suspension cells.

C. Evaluation of the system components

We also evaluated the contribution of specific components of the system to the detection accuracy. Tables 3, 4, and 5 summarize this evaluation on the L929, CHO, and Sf21 respectively. In the first column, the outputs of the two random forests were combined using the connected components as described in Section 4. The same was done in the second column but the keypoints were thresholded according to Section 2. This thresholding is also used in columns 3 to 8. In the third column, the agglomerative hierarchical clustering with the group average linkage was used instead of the connected components. In the fourth one, our customized linkage method was used instead. This customized linkage is also used in columns 5 to 8. In the fifth column, Eq. (13) was used to find the hit-points instead of the simple arithmetic average of the coordinates of the keypoints inside each cluster. Eq. (13) is also used in columns 6 to 8. In the profile expansion columns, three strategies were tested: The first is using parallel profile sets. The second is using SAS. The third is combining both of them.

The estimates in tables 3, 4, and 5 are cross validation estimates, where one image per cell line is used for training and the remaining ones are used for testing.

An important result of these tables is that SAS achieves higher detection scores than the profile set. On the other hand, using SAS together with the profile set delivers a bit higher detection rate than using SAS alone. Nevertheless, we sacrifice this small improvement in the detection rate in favor of better detection times. All other experiments in this paper were done with SAS alone. Therefore, the SAS columns in tables 3, 4, and 5 correspond to the estimates in Table 2.

As shown in Figure 5, simulated cells tend to form negative DOG blobs, whereas real cells tend to form positive DOG blobs in the positively defocused images. The system was able to automatically detect the right type of blobs in each case using Eq. (1).

We did not conduct a thorough analysis of the feature importance. However, the random forest has an internal mechanism to rank its features: the mean classification rate increase [5]. A random image from each cell line was used to train the system and the feature with the highest rank was recorded. Table 8 shows the result. The last column displays the highest ranked feature when the previous five images are used together to train the system. According to this table, at least one keypoint feature from each of the five keypoint feature sets was ranked the best. This does not imply that all feature sets are necessary to obtain the detection rate reported in Table 2. However, it gives an indicator that all keypoint feature sets are informative. Regarding profile features, Table 8 shows that the mean value of the second derivative and V_2 (cf. Eq. (4)) are the two highest ranked features.

D. Comparison with other approaches

We compared our system with [4] and [3] (cf. Section 1). Table 6 summarizes the required input for each approach. In [3], there is a well-developed segmentation approach. It is, however, worth pointing out that only the detection part

	CHO	L929	Sf21	Simulated A	Simulated B
Precision	77.7±8.0	82.8±4.4	97.3±0.9	98.9 ±0.7	95.4 ±2.1
Recall	92.9±3.0	92.6±2.9	96.4±3.2	99.4 ± 0.7	96.5 ±1.6
Detection error	0.147±0.03	0.123±0.01	0.031±0.01	0.008 ±0.005	0.040 ±0.01
Centeredness error	0.477±0.13	0.377±0.07	0.164±0.02	0.173 ±0.10	0.222 ±0.11

TABLE 2
CELL DETECTION ACCURACY ON DIFFERENT CELL LINES.

	CC	SIFT Threshold	AHC average	AHC custom	Weighted avg.	Profile expansion		
						Profile sets	SAS	Both
Precision	67.1±4.2	76.8±4.2	72.6±5.1	76.0±4.5	76.9±4.4	78.9±5.1	82.8±4.4	83.4±4.0
Recall	66.2±5.9	73.2±3.4	91.0±4.5	90.2±4.4	91.0±4.6	91.7±4.0	92.6±2.9	92.6±3.0
Detection error	0.334±0.04	0.250±0.01	0.182±0.01	0.169±0.01	0.161±0.01	0.147±0.02	0.123±0.01	0.120±0.01
Centeredness error	0.528±0.08	0.381±0.09	0.484±0.11	0.406±0.08	0.382±0.07	0.382±0.07	0.377±0.07	0.369±0.07

TABLE 3
THE CONTRIBUTION OF THE DIFFERENT SYSTEM COMPONENTS TO THE DETECTION ACCURACY (L929)

	CC	SIFT Threshold	AHC average	AHC custom	Weighted avg.	Profile expansion		
						Profile sets	SAS	Both
Precision	70.5±5.3	76.8±5.9	68.5±10.7	73.4±8.8	74.0±8.6	76.4±8.4	77.7±8.0	78.8±7.8
Recall	58.7±6.4	65.5±4.7	91.4±3.9	90.1±4.4	90.7±4.8	91.9±4.3	92.9±3.0	93.1±3.2
Detection error	0.354±0.05	0.288±0.04	0.201±0.03	0.183±0.02	0.177±0.02	0.159±0.02	0.147±0.03	0.141±0.03
Centeredness error	0.472±0.20	0.432±0.07	0.652±0.24	0.503±0.14	0.501±0.14	0.455±0.10	0.477±0.13	0.446±0.11

TABLE 4
THE CONTRIBUTION OF THE DIFFERENT SYSTEM COMPONENTS TO THE DETECTION ACCURACY (CHO)

	CC	SIFT Threshold	AHC average	AHC custom	Weighted avg.	Profile expansion		
						Profile sets	SAS	Both
Precision	89.0±2.2	93.1±1.9	90.9±1.6	94.1±0.9	94.3±0.8	95.0±1.5	97.3±0.9	97.4±0.9
Recall	78.6±2.4	83.3±4.1	95.8±0.9	95.1±0.9	95.2±1.2	96.3±1.1	96.4±3.2	96.8±3.1
Detection error	0.162±0.02	0.118±0.03	0.066±0.01	0.054±0.00	0.053±0.01	0.043±0.01	0.031±0.01	0.029±0.01
Centeredness error	0.132±0.00	0.180±0.01	0.216±0.01	0.189±0.02	0.163±0.01	0.157±0.01	0.164±0.02	0.162±0.02

TABLE 5
THE CONTRIBUTION OF THE DIFFERENT SYSTEM COMPONENTS TO THE DETECTION ACCURACY (Sf21)

is used in the comparison. [4] utilizes three algorithms at three different focus levels and combines the results. In our evaluation, instead of combining the results of these three algorithms, we select the one which has the minimum error. This strategy gave better results on our images.

Due to the difficulty of the manual parameter tuning, this comparative evaluation was performed using only one cell line, CHO, and without cross validation. One image was randomly chosen and used to train our system. The same image was used for parameter tuning of [4] and [3]. The rest of the images were used for testing.

The software of [4] was obtained from its authors while we implemented the cell detection part of [3] ourselves. The optimal value of the single parameter of [3] was found by scanning the parameter domain and selecting the value which minimizes the detection error. On the other hand, we optimized the parameters of [4] manually. The result of the comparison is shown at the left hand side of Table 7.

In order to investigate how the three approaches perform under illumination and scale change, we did the following experiment: The comparison was applied on the CHO cell line, but after perturbing the images. An illumination field was applied on all CHO images. The same field was applied on all testing and training images. In addition, the testing images

were resampled using the following scales: 0.5, 0.75, 1, 1.25, and 1.5. The results are shown at the right hand side of Table 7.

	[3]	[4]	Our approach
Required number of images	2	3	1
Manually tuned parameters	1	> 9	0

TABLE 6
INPUT REQUIREMENTS FOR [3], [4], AND OUR APPROACH

E. Evaluation of illumination, orientation, and scale invariance

The simulation software in [17] can generate illumination artifacts on simulated images. In order to make the experiment more realistic, we applied the simulated illumination field on our real cell lines. Figure 6 shows how the detection error changes with the illumination scale, i.e. the ratio between the illumination energy and the image energy. Figure 7 shows the difference between an image at illumination scale zero and another one at illumination scale 100. The detection error change between these two extreme cases, as shown in Figure 6, was 8% in the worst case.

Only one image in each cell line was used for training. It is an image at illumination scale zero. The other images in

	CHO			CHO with illumination and scale change		
	Our approach	[3]	[4]	Our approach	[3]	[4]
Precision	88.1 ± 2.3	56.1 ± 11.1	80.9 ± 3.2	80.8 ± 3.7	81.0 ± 12.7	43.0 ± 16.2
Recall	87.6 ± 4.2	91.8 ± 3.5	61.3 ± 9.3	91.4 ± 2.5	36.5 ± 18.9	23.4 ± 10.6
Detection error	0.122 ± 0.024	0.260 ± 0.067	0.288 ± 0.042	0.138 ± 0.021	0.412 ± 0.060	0.668 ± 0.041
Centeredness error	0.373 ± 0.342	0.552 ± 0.396	0.495 ± 0.581	0.469 ± 0.276	0.665 ± 0.729	1.350 ± 2.03

TABLE 7
COMPARISON WITH OTHER APPROACHES

	CHO	L929	Sf21	Simulated A	Simulated B	All
Keypoint features	$R_n(\mathbf{p}, \pi)$	SIFT descriptor feature	SIFT descriptor feature	Stencil feature	VMap($\mathbf{p}, 6$)	DOG
Profile features	$\text{mean}(d^2 f / dv^2)$	V_2	V_2	$\text{mean}(d^2 f / dv^2)$	V_2	$\text{mean}(d^2 f / dv^2)$

TABLE 8
HIGHEST RANKED FEATURES ACCORDING TO THE RANDOM FOREST MEAN CLASSIFICATION RATE INCREASE.

the cell line were used for testing at each of the following illumination scales: 0, 20, 40, 60, 80, and 100.

Figure 8 depicts the system’s invariance to image scale. In this experiment, the system was trained using one image per cell line and tested on other up- or down-sampled images from the same cell line. The figure shows that the detection error change is in the range of 4% excluding a sudden increase in the detection error of Sf21 at scale 0.5. This indicates that scale invariance is limited from the bottom. The reason is that downsampling reduces the number of SIFT keypoints due to the structure degradation.

Finally, Figure 9 shows the degree of invariance with respect to cell orientation. Rotating the images in order to test the system invariance to the orientation change is not the best choice because of the diversity of cell orientations in each image. Therefore, we simulated cell images so that all cells inside the same image have the same orientation. The shape model in the simulation software [17] cannot generate elongated cells with dominant orientation. Therefore, for this experiment, we replaced its shape model with an elliptical one. The system was trained using one image at orientation zero and tested on five images at each of the following orientations: 0° , 30° , 60° , 90° , 120° , and 150° . Each image contained 150 simulated cells.

F. Evaluation of the detection time

In order to investigate the feasibility of the algorithm, we measured the detection time for all cell lines. The evaluation was done on a Dell laptop with 8 GB RAM and an Intel Core i7-2720QM processor with clock speed 2.20 GHz. The implementation details are as follows: The feature extraction was implemented in Matlab, the classification was done using the R package randomForest [18], the clustering was implemented in Java, and SIFT features were obtained from VIFeat [39]. All modules were put together in a single Matlab application.

The system was trained and tested as described in Section 7-B. The detection times are reported in the upper part of Table 9. The lower part of the table shows the results when the same experiment was applied on subsampled images (subsampling factor 0.5). As can be seen in the table, the detection time is approximately in the range [30, 46] seconds

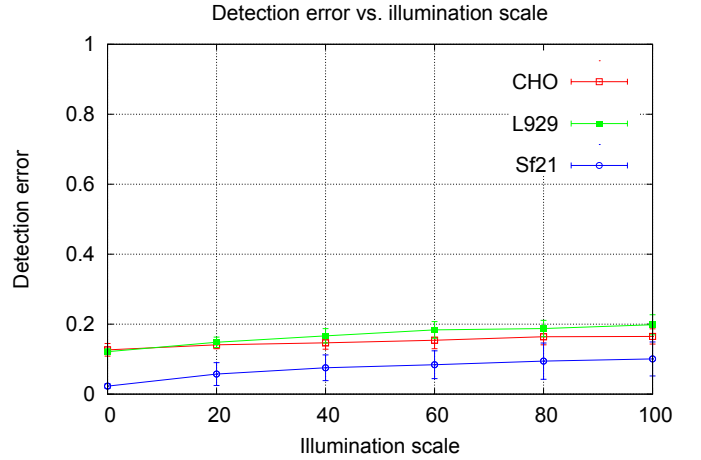


Fig. 6. Illumination invariance: An image at illumination scale 0 was used for training and the other images were used for testing at different illumination scales.

per image and drops to the range [5, 14] seconds per image after subsampling.

G. Evaluation of the generalization on multiple cell lines

In Section 7-B, the system was trained separately for each cell line. In fact, it is more challenging to learn to detect cells when images from different cell lines are used in the training. In this experiment, one image from each cell line was randomly chosen. The five chosen images were used to train the system. The rest of the images were used for testing. This process was repeated five times. As the images are of different dynamic ranges and/or modalities, each image was normalized to [0, 1]. The simulated images were also inverted in order to have one-sided cell keypoints in the training data. Table 10 shows the results. Comparing Table 10 to Table 2, one can see a relatively considerable increase of the detection error for Sf21 and Simulated B. Nevertheless, the maximum detection error is still 15.5%.

In order to investigate whether similar cells are more suited for joint training, we conducted two additional experiments. Table 11 shows the results of the same process described

	CHO	L929	Sf21	Simulated A	Simulated B
Detection time (seconds/image)	45.88 \pm 13.60	36.69 \pm 5.59	40.65 \pm 7.13	30.47 \pm 0.33	31.00 \pm 1.88
Detection error	0.147 \pm 0.03	0.123 \pm 0.01	0.031 \pm 0.01	0.008 \pm 0.005	0.040 \pm 0.01
Detection time on subsampled images (seconds/image)	13.75 \pm 2.11	12.68 \pm 0.93	8.97 \pm 1.07	7.91 \pm 0.34	4.93 \pm 0.26
Detection error on subsampled images	0.128 \pm 0.01	0.112 \pm 0.01	0.071 \pm 0.02	0.009 \pm 0.001	0.043 \pm 0.004

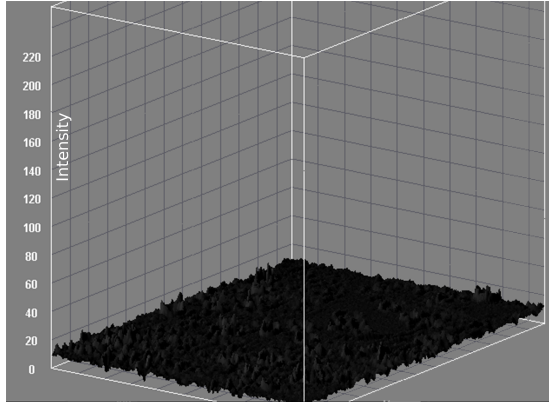
TABLE 9

EVALUATION OF THE DETECTION TIME. RESOLUTION OF CHO, L929, AND Sf21 IMAGES IS 1280 X 960 PIXELS. RESOLUTION OF SIMULATED A AND B IS 1200 X 1200 PIXELS. ALL RESOLUTIONS ARE GIVEN BEFORE SUBSAMPLING.

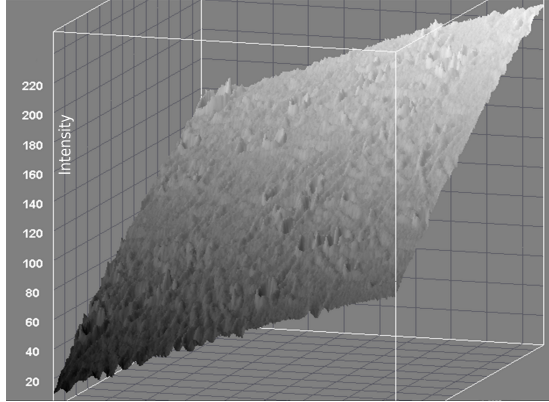
	CHO	L929	Sf21	Simulated A	Simulated B
Precision	77.7 \pm 7.2	79.6 \pm 8.0	73.2 \pm 2.4	98.9 \pm 0.1	81.8 \pm 5.7
Recall	92.2 \pm 4.3	89.4 \pm 6.6	99.7 \pm 0.1	98.3 \pm 0.4	97.6 \pm 0.3
Detection error	0.150 \pm 0.02	0.155 \pm 0.01	0.136 \pm 0.01	0.014 \pm 0.00	0.103 \pm 0.03
Centeredness error	0.498 \pm 0.08	0.438 \pm 0.12	0.179 \pm 0.04	0.177 \pm 0.00	0.256 \pm 0.01

TABLE 10

JOINT TRAINING: FIVE IMAGES WERE RANDOMLY CHOSEN, ONE FROM EACH CELL LINE. THEY WERE USED TO TRAIN THE SYSTEM AND THE REST WERE USED FOR TESTING. THIS PROCESS WAS REPEATED FIVE TIMES.



(a) CHO image at illumination scale = 0



(b) CHO image at illumination scale = 100

Fig. 7. Illumination invariance example: The upper image was used for training, and the lower for testing.

above, but training and testing were applied only on the adherent cell lines. The same applies for Table 12, but for the simulated cell lines. The detection error in both tables is very close to the detection error in Table 2.

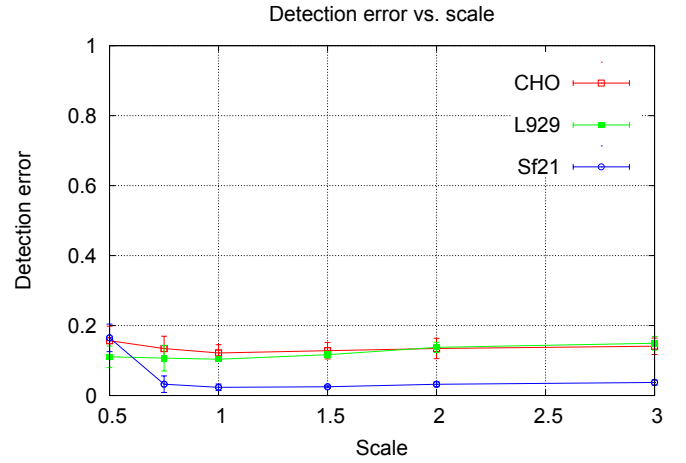


Fig. 8. Scale invariance: An image at scale 1 was used for training and the other images were used for testing at different scales.

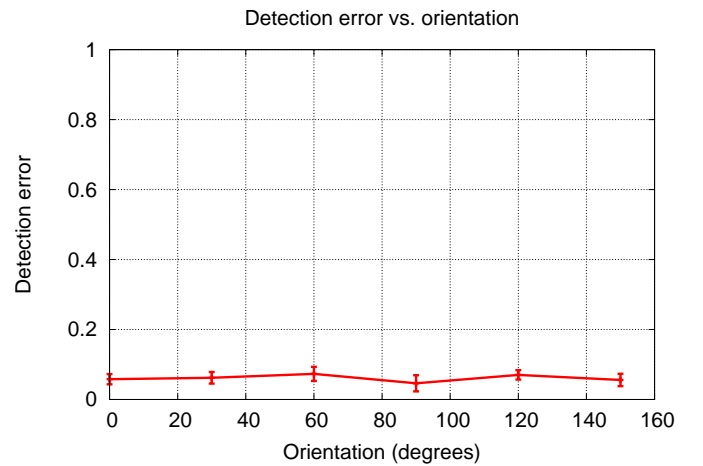


Fig. 9. Orientation invariance: An image at orientation 0 was used for training. Five images at each orientation were used for testing. Each image contains 150 cells simulated under an elliptical shape model.

	CHO	L929
Precision	76.8 ± 5.0	80.7 ± 5.1
Recall	94.3 ± 2.2	92.0 ± 3.6
Detection error	0.145 ± 0.01	0.137 ± 0.01
Centeredness error	0.497 ± 0.13	0.462 ± 0.06

TABLE 11

JOINT TRAINING: TWO IMAGES WERE RANDOMLY CHOSEN, ONE FROM CHO AND ANOTHER ONE FROM L929. THEY WERE USED TO TRAIN THE SYSTEM AND THE REST WERE USED FOR TESTING. THIS PROCESS WAS REPEATED FIVE TIMES.

	Simulated A	Simulated B
Precision	98.8 ± 0.2	93.3 ± 1.0
Recall	98.1 ± 0.4	97.0 ± 0.3
Detection error	0.015 ± 0.00	0.048 ± 0.00
Centeredness error	0.173 ± 0.00	0.229 ± 0.01

TABLE 12

JOINT TRAINING: TWO IMAGES WERE RANDOMLY CHOSEN, ONE FROM SIMULATED A AND ANOTHER ONE FROM SIMULATED B. THEY WERE USED TO TRAIN THE SYSTEM AND THE REST WERE USED FOR TESTING. THIS PROCESS WAS REPEATED FIVE TIMES.

8. DISCUSSION

Several approaches [2], [3], [4], [28] utilize, though in different ways, images at two or three focus levels to improve the contrast. This is appealing as it is possible to get considerably higher contrast from the intensity change with the defocus distance. The drawback is that at least two images are needed. As this is not always available, our system was developed to work with a single defocused image. In fact, using one image has another important advantage: It facilitates extending the algorithm in the future for more image modalities.

We found that the approach in [32] on phase-contrast microscopy has a partial conceptual similarity with our approach. This similarity lies in the use of two classifiers which have goals analogous to the goals of our keypoint and profile classifiers. However, our features were carefully designed and heavily tested for the rotation-, scale-, and illumination-invariance. Another difference is that the whole system is fully automatic and its internal details, e.g. learning the maximum inner profile length, were designed for automation and invariance. In addition, the use of hierarchical clustering in order to optimally aggregate the second classifier results was a novel contribution which proved to be effective, especially using our customized linkage method. Lastly, the system is adapted to low contrast bright-field microscopy. In fact, compared to the bright-field approaches in [4] and [3] which need both multiple images and manual parameter tuning, our approach delivered higher detection accuracy and higher invariance degree to illumination and scale changes.

Before using the classifiers, the system must be trained. For the training, a set of images with ground truth is required. Due to the invariance of the extracted features and the use of random forests, the system can learn from a relatively small amount of training data. Only one image per cell line was used for training in our experiments.

The ground truth is segmentation that may have been performed with any manual, semi-automatic, or automatic method. For the this study, we chose a manual segmentation as we did not want to introduce a bias into the system performance caused by suboptimal training data. The images should be defocused and the same defocus distance should be

used for training and detection. This defocus can be either in the positive or the negative direction. The defocus is important for at least two reasons: Firstly, as already mentioned, some adherent cells are totally invisible at focus. Secondly, the defocus smooths out the tiny details which degrade the system performance in terms of time and detection accuracy.

SIFT keypoints in bright-field images result from cell structures, but also from debris, noise in the background, and other image artifacts. It is possible to eliminate some irrelevant keypoints by imposing a high DOG threshold or a low PCR threshold [23]. However, some cell keypoints may have high principal curvatures ratio because of the elongation of the cells. Other cell keypoints may have low DOG value due to cell adherence or insufficient contrast of the considered image modality. Consequently, it is not always possible to separate cell keypoints from non-cell keypoints using these two features. Nevertheless, it is possible to reduce the number of irrelevant keypoints by computing the maximum $|DOG|$ in each training cell, then considering the minimum of these maxima as a threshold. Therefore, in our experiments, we set the DOG threshold to the previous value and the PCR threshold to infinity.

Cell keypoints have either positive or negative DOG values when they form valley-like or mountain-like structures, respectively. We have found in preliminary experiments (data not shown) that using one-sided keypoints leads to a better profile learning. For this reason, our system considers either the positive DOG keypoints or the negative, but not both.

In [12], the intensity is sampled using a stencil instead of a patch and used for neuron detection in electron microscopy. A similar idea was used in a completely different field [26], where a radial sampling pattern was employed to sample 3D vessels. Both methods used fixed-size stencils. In our case, the scale and orientation of the keypoint deliver additional information. This can then be used to make the stencil scale- and orientation-invariant. In order to make it invariant to the local shift of intensity, one can subtract the minimum or the mean intensity of the stencil from all stencil nodes. We chose the mean because it is less sensitive to outliers.

In [34], it was shown that ray features are better than Haar-

like features for cell recognition. Thus, we used ray features in our work. Our contribution to this feature set is that we made it scale- and orientation-invariant. In order to achieve scale-invariance for such features, computing the distances with respect to scale is insufficient. In fact, the gradient computation has to be additionally performed with respect to scale. If the gradient is computed using conventional kernels like Sobel or Prewitt, its norm $R_n(\mathbf{m}, \theta_k)$ will be scale-dependent. We use a gradient computation that handles this issue correctly.

In [40], it was shown that variance maps can distinguish cells from background. The variance map value at a pixel is simply the variance of intensities in a neighborhood centered at this pixel. The neighborhood size is fixed and based on the cell size [40]. We expanded this concept and made it scale-invariant.

The number of the keypoints inside each cell depends on the noise level, the cellular details level, the defocus distance, the cell shape, and the SIFT parameters (the PCR and the DOG value). The goal of the profile learning was to connect the keypoints which belong to the same cell together so that they are recognized as one cell.

We mentioned that the derivatives which are used in the profile features are Gaussian derivatives with $\sigma = 0.1N_F$. As the scale of the derivative, i.e. the previous sigma, is measured in units of points, not in pixels, the derivative signal is to a large extent scale-invariant. Note that, all profile features are also invariant to the local shift of intensity and that the mean, the maximum, and the minimum of the profile do not belong to the profile feature set because they are sensitive to this shift.

If cells are circular and noise-free, we may get almost one keypoint per cell. Therefore, in such case, we'll get very few or even zero inner profiles for training. If the cells are too far from each other, we may get very few or even zero cross profiles for training. This occurs because the algorithm extracts profiles between nearby keypoints only. Both cases occurred in some preliminary experiments on data which was not shown in this study. The algorithm handles these two cases which makes the training robust against odd situations in the training data.

A main disadvantage of using a set of profiles is the computational cost. Furthermore, it turned out that SAS leads to a higher detection rate as shown in the results section. This is, probably, due to the fact that SAS serves also as a preprocessing step. For instance, we noticed that the SAS improved the edges drastically. Consequently, the edge-based features like the ray features, gained higher discriminative power.

In our experiments, we achieved robustness to low frequency changes in the illumination by using features that are locally invariant to an offset in intensity. In a small region of the image, this local shift of intensity can be interpreted as a constant. We regard this as an important feature of our system, as many real world images suffer from inhomogeneous illumination. It is worth pointing out that images with illumination artifacts should be used without normalization, because the image measures which are usually used in normalization like mean, standard deviation, maximum, and minimum of the image intensity depend on the illumination information.

If the distance between the two keypoints of a profile is

large, then the illumination artifacts at this profile cannot be approximated by a constant. In other words, the invariance to the local-intensity shift is beneficial only if the profiles are extracted between *nearby* keypoints. Therefore, the system should somehow avoid extracting profiles between far keypoints. In order to achieve this goal, the algorithm learns the maximum inner profile length L in a scale-independent manner from the training data. This contributes to the partial illumination-invariance of the profile learning and reduces the detection and training times. In fact, learning L is particularly useful because this length is a cell characteristic. Whereas, for instance, learning the cross profile length does not make sense because it depends on the distribution of the cells in the cell culture.

We used RF as a classifier model for the keypoint and the profile learning. The RF does not need parameter tuning which was one of our design goals. It is also inherently a multi-class classifier. This makes extending the keypoint learning in further research for debris and agglomeration detection easier. The previous two points can be considered as advantages of the RF over some other state-of-the-art classifiers like the SVM. In fact, some empirical studies [13], [14] showed that the RF outperformed the SVM in terms of area under ROC (AUC) on imbalanced data, even though that was not always the case [25]. Moreover, the immunity of the RF against overfitting grants the system the ability to learn from small training data sizes. This last point makes it favorable over classifiers like the probabilistic boosting trees [37].

There is, in general, no guarantee that the cell and the background classes are balanced. The same applies for the two classes in the profile learning. The imbalance problem can be solved by using a balanced random forest BRF or a weighted random forest WRF [7]. Both have similar ROC curves. But, according to the same reference, the BRF is computationally more efficient and less vulnerable to noise.

The output of the two classifiers can be seen as a graph whose nodes are the cell keypoints and whose edges are the inner profiles. Therefore, the connected components of this graph can be regarded as the detected cells. However, our results show that higher detection rates can be achieved when the probabilistic output of the profile classifier is used as a similarity measure in an AHC step. Moreover, the detection rate was further improved by using our customized linkage method where more application-specific information was involved. In fact, it is plausible to expect that the AHC is more robust than the CC because it starts aggregating the most reliable cases. Furthermore, the decision about the less reliable cases does not depend on a single classification but on the average of several profile classifications.

We pointed out in the introduction that a good cell detection approach should facilitate tracking when the latter is required. The output of our system after the clustering stage is a set of keypoints where each one is assigned to a cell and equipped with a set of features. A subgroup of these features has been used in tracking applications. See, for example, the use of the SIFT descriptor for tracking in [11].

We evaluated the system with respect to runtime on our image database and on a subsampled version of it. We noticed

that, on our data, the detection time of the subsampled images was 3-4 times shorter than the detection time of the original images. This suggests that the detection time is proportional to the number of pixels in the considered image. Moreover, with the exception of Sf21, the detection error was not increased by subsampling. In fact, for the adherent cell lines, it was decreased. This is probably caused by the sampling-induced smoothing of the cellular details which could otherwise mislead the detection algorithm and degrade its precision. However, this observation cannot be generalized without taking the original image resolution into account. In our experiments, the real cell line images before subsampling had a resolution of 1280 x 960 pixels with objective's magnification equal to 20x.

The system was also tested for its generalization ability. The results show that it can learn to detect cells even when several cell lines of different visual appearance are used for training. However, the detection error was smaller when only similar cell lines are used for training. Therefore, we propose to use different classifiers for different cell lines.

9. CONCLUSION

We have presented a novel system for cell detection in bright-field microscope images. The system is fully automatic as no manual parameter tuning is needed in training or in detection.

For the evaluation, three real cell lines were used: CHO, L929, and Sf21. CHO and L929 were adherent cells while Sf21 were cells in suspension. Adherent cells had very low contrast and were almost invisible at focus. In addition, simulated cells with both high and very low SNR were used in the evaluation. In total, more than 3500 real cells and 30000 simulated cells were used.

The system was designed with special care for robustness to illumination artifacts and invariance to cell size and orientation. It was trained with specific image scale, orientation, and illumination conditions. It was then tested on images of different scale, orientation, and illumination conditions. Our results show that the system yields a high detection accuracy and high invariance scores in reasonable computation time.

All experiments on real and simulated images were done without any manual parameter tuning. Because of this generality, we are currently investigating the system's ability in detecting molecules in scanning tunneling microscope images.

ACKNOWLEDGMENT

We highly appreciate the efforts of our partners in the Institute of Bioprocess Engineering, Friedrich-Alexander-University, Erlangen-Nuremberg. These efforts include cell culturing, image acquisition, and labeling. The authors would also like to thank the Bavarian Research Foundation BFS for funding the project COSIR under contract number AZ-917-10 and our industrial partners for the fruitful collaboration. We gratefully acknowledge funding of the Erlangen Graduate School in Advanced Optical Technologies (SAOT) by the German Research Foundation (DFG) in the framework of the German excellence initiative. Special thanks for Dr. Gabriele Becattini for providing us with the software and the user manual of his cell detection method. Great thanks go to Dr. Elli Angelopoulou for proofreading the article.

REFERENCES

- [1] U. Agero, C. H. Monken, C. Ropert, R. T. Gazzinelli, and O. N. Mesquita, "Cell surface fluctuations studied with defocusing microscopy," *Physical Review E*, vol. 67, no. 5, p. 051904, May 2003.
- [2] R. Ali, M. Gooding, M. Christlieb, and M. Brady, "Phase-based segmentation of cells from brightfield microscopy," in *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, April 2007, pp. 57–60.
- [3] R. Ali, M. Gooding, T. Szilágyi, B. Vojnovic, M. Christlieb, and M. Brady, "Automatic segmentation of adherent biological cell boundaries and nuclei from brightfield microscopy images," *Machine Vision and Applications*, vol. 23, no. 4, pp. 607–621, 2012.
- [4] G. Becattini, L. Mattos, and D. Caldwell, "A novel framework for automated targeting of unstained living cells in bright field microscopy," in *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, April 2011, pp. 195–198.
- [5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [6] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, November 1986.
- [7] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," University of California, Berkeley, Tech. Rep., 2004.
- [8] C. Curl, T. Harris, P. Harris, B. Allman, C. Bellair, A. Stewart, and L. Delbridge, "Quantitative phase microscopy: a new tool for measurement of cell culture growth and confluency in situ," *Pflügers Archiv European Journal of Physiology*, vol. 448, no. 4, pp. 462–468, 2004.
- [9] K. Ghatak, *Techniques and Methods in Biology*. PHI Learning Pvt. Ltd., 2010.
- [10] D. Gil, P. Radeva, and F. Vilariño, "Anisotropic contour completion," in *Proceedings of the International Conference on Image Processing*, vol. 1, September 2003, pp. 869–872.
- [11] R. Jiang, D. Crookes, N. Luo, and M. Davidson, "Live-cell tracking using sift features in dic microscopic videos," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 9, pp. 2219–2228, September 2010.
- [12] E. Jurrus, A. Paiva, S. Watanabe, J. Anderson, B. Jones, R. Whitaker, E. Jorgensen, R. Marc, and T. Tasdizen, "Detection of neuron membranes in electron microscopy images using a serial neural network architecture," *Medical Image Analysis*, vol. 14, no. 6, pp. 770–783, December 2010.
- [13] M. Khalilia, S. Chakraborty, and M. Popescu, "Predicting disease risks from highly imbalanced data using random forest," *BMC Medical Informatics and Decision Making*, vol. 11, no. 1, p. 51, 2011.
- [14] T. Khoshgoftaar, M. Golawala, and J. Van Hulse, "An empirical study of learning from imbalanced data using random forest," in *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, no. 2, October 2007, pp. 310–317.
- [15] G. N. Lance and W. T. Williams, "A generalized sorting strategy for computer classifications," *Nature*, vol. 212, p. 218, October 1966.
- [16] —, "A general theory of classificatory sorting strategies: 1. hierarchical systems," *The Computer Journal*, vol. 9, no. 4, pp. 373–380, 1967.
- [17] A. Lehmussola, P. Ruusuvi, J. Selinummi, H. Huttunen, and O. Yli-Harja, "Computational framework for simulating fluorescence microscope images with cell populations," *IEEE Transactions on Medical Imaging*, vol. 26, no. 7, pp. 1010–1016, July 2007.
- [18] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002. [Online]. Available: <http://CRAN.R-project.org/doc/Rnews/>
- [19] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.
- [20] X. Long, W. Cleveland, and Y. Yao, "A new preprocessing approach for cell recognition," *IEEE Transactions on Information Technology in Biomedicine*, vol. 9, no. 3, pp. 407–412, 2005.
- [21] —, "Automatic detection of unstained viable cells in bright field images using a support vector machine with an improved training procedure," *Computers in Biology and Medicine*, vol. 36, no. 4, pp. 339–362, 2006.
- [22] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision*, no. 2, 1999, pp. 1150–1157.
- [23] —, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.
- [24] V. Lulevich, Y.-P. Shih, S. H. Lo, and G.-Y. Liu, "Cell tracing dyes significantly change single cell mechanics," *The journal of physical chemistry. B*, vol. 113, no. 18, pp. 6511–9, 2009.

- [25] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, no. 1-2, pp. 169–186, 2003.
- [26] S. Mittal, Y. Zheng, B. Georgescu, F. Vega-Higuera, S. Zhou, P. Meer, and D. Comaniciu, "Fast automatic detection of calcified coronary lesions in 3d cardiac ct images," in *Machine Learning in Medical Imaging*, ser. Lecture Notes in Computer Science. Springer, 2010, vol. 6357, pp. 1–9.
- [27] B. Morgan and A. Ray, "Non-uniqueness and inversions in cluster analysis," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 44, no. 1, pp. 117–134, 1995.
- [28] F. Mualla, S. Schödl, B. Sommerfeldt, and J. Hornegger, "Using the Monogenic Signal for Cell-Background Classification in Bright-Field Microscope Images," in *Proceedings des Workshops Bildverarbeitung für die Medizin 2013*, 2013, pp. 170–174.
- [29] F. Murtagh, "Multidimensional clustering algorithms," *Compstat Lectures, Vienna: Physika Verlag*, 1985, vol. 4, 1985.
- [30] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.
- [31] T. Nattkemper, H. Ritter, and W. Schubert, "Extracting patterns of lymphocyte fluorescence from digital microscope images," *Intelligent Data Analysis in Medicine and Pharmacology*, vol. 99, pp. 79–88, 1999.
- [32] J. Pan, T. Kanade, and M. Chen, "Learning to detect different types of cells under phase contrast microscopy," in *Microscopic Image Analysis with Applications in Biology*, September 2009.
- [33] B. Russell, A. Torralba, K. Murphy, and W. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 157–173, May 2008.
- [34] K. Smith, A. Carleton, and V. Lepetit, "Fast ray features for learning irregular shapes," in *Proceedings of the IEEE International Conference on Computer Vision*, October 2009, pp. 397–404.
- [35] D. J. Stephens and V. J. Allan, "Light microscopy techniques for live cell imaging," *Science (New York, N.Y.)*, vol. 300, no. 5616, pp. 82–86, 2003.
- [36] M. Tscherepanow, F. Zöllner, M. Hillebrand, and F. Kummert, "Automatic segmentation of unstained living cells in bright-field microscope images," in *Advances in Mass Data Analysis of Images and Signals in Medicine, Biotechnology, Chemistry and Food Industry*, ser. Lecture Notes in Computer Science. Springer, 2008, vol. 5108, pp. 158–172.
- [37] Z. Tu, "Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, October 2005, pp. 1589–1596.
- [38] W. van Opstal, C. Ranger, O. Lejeune, P. Forgez, H. Boudin, J. Bisconte, and W. Rostene, "Automated image analyzing system for the quantitative study of living cells in culture," *Microscopy Research and Technique*, vol. 28, no. 5, pp. 440–447, 1994.
- [39] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," <http://www.vlfeat.org/>, 2008.
- [40] K. Wu, D. Gauthier, and M. Levine, "Live cell image segmentation," *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 1, pp. 1–12, January 1995.
- [41] Z. Yin, R. Bise, M. Chen, and T. Kanade, "Cell segmentation in microscopy imagery using a bag of local bayesian classifiers," in *The IEEE International Symposium on Biomedical Imaging (ISBI) 2010*, April 2010.
- [42] A. Zaritsky, S. Natan, J. Horev, I. Hecht, L. Wolf, E. Ben-Jacob, and I. Tsarfaty, "Cell motility dynamics: A novel segmentation algorithm to quantify multi-cellular bright field microscopy images," *PLoS ONE*, vol. 6, no. 11, p. e27593, 11 2011.