Contour expansion algorithm preprocessed by Hough transform and circular shortest path for ovoid objects

Tleis, M.; Verbeek, F.J.; Jennane, R.

Contour Expansion Algorithm Preprocessed by Hough Transform and Circular Shortest Path for Ovoid Objects

Mohamed Tleis[1] and Fons J. Verbeek[1]
[1] Section Imaging and BioInformatics, LIACS, Leiden University, Leiden, The Netherlands
e-mail: m.tleis@liacs.leidenuniv.nl, f.j.verbeek@liacs.leidenuniv.nl

Abstract— **Circular and ovoidal objects such as cells and supramolecular complexes are very common in bio-imaging. In order to do measurements on such objects, the accurate detection of these objects from microscope images is essential. Therefore our objective is finding accurate and reliable methods to extract the contour delineating the object. In earlier work, we have shown to be successful in achieving this objective through dynamic programming, where we used Hough transform and a minimal path algorithm to estimate the contour location. However, due to the inherently fuzzy nature of edges and as microscope imaging can be very delicate, a refinement of the initial estimate is sometimes required. In these cases the minimal path is no longer the ultimate valid representation of the contour. This paper describes an expansion algorithm in the polar image that is created for each object; with this new algorithm we achieve the necessary refinement of contours. From previous results we can assess the improvement to existing segmentation methods, including our own approach. The exact contour contributes in the accomplishment of precise measurements of the objects so that machine learning techniques can better recognize the subtle patterns within the data. As the method is geared to ovoid like objects, it can be easily generalized to other image of other modalities. Here we apply the method in the domain of yeast cell biology.**

Keywords— **Contour Extracting, Contour Expansion, Yeast Cells, Ovoid-shaped objects, Hough Transform, Circular Shortest Path.**

I. Introduction

In bio-imaging, circular and ovoidal objects are very common, i.e. cells and supramolecular complexes. Such objects need to be detected in microscope images for analysis. We develop measurement frameworks for cell and microbiology and in order to test algorithms for the detection of ovoid-like shapes, test images are used that originate from the applications we develop for; here we focus on yeast cells. Detection of yeast cells enables to analyse their phenotypical information such as location and levels of gene expression. For measurement, the accurate detection of these cells from microscope images is essential. Our objective is to find reliable methods for the extraction of an accurate contour delineating the object. We have shown to be successful in achieving this objective through dynamic programming (DP) [1], where we used Hough transform and a minimal path algorithm to estimate the contour location [2].

Since microscope imaging can be very delicate and edges have an inherently fuzzy nature, a refinement of the initial estimate is sometimes required. For example, the objects to be segmented have contours that are described by more than one pixel. This is typically true for the yeast image dataset that we show as a case study in this paper. This problem is observed when high numerical aperture (NA) lenses are used that results in high resolution image sizes larger than 512x512 pixels. Further analysis of the results demonstrates that circular minimal path method has a bias toward the inner part of the object as a result of the fact that some inner pixel values are lower than the edge pixel values. We state that, in these cases, the minimal path is no longer the ultimate valid representation of the contour. In this paper we use a yeast dataset for which this is typically the case.

As a possible solution to this problem, we developed an expansion algorithm that directly evaluates the polar image P_i used to extract the minimal path. We refer to P_i as a polar image, which is a polar transformation of each object O_i in the original image. With our additional heuristic, we achieve the necessary refinement of contours. This contributes to the accomplishment of precise measurements of the objects so that machine learning techniques can better recognize the subtle patterns within the data.

In the next section, we review our initial approach to segment the objects using Hough Transform and minimal path algorithms. Subsequently, we explain our new contour expansion algorithm and its three control parameters: *resistance*, *limit* and *convergence*. Thereafter, we discuss the results we have obtained with this new approach by comparing it to the original method and two known yeast segmentation packages, *Cell Serpent* [3] and *CellStat* [4]. Finally, the last section is dedicated to our conclusions.

II. Hough Transform and Circular Minimal Path

Our initial segmentation entails a novel approach to detect ovoid objects through a combination of Hough transform and a minimal path algorithm. Here we shortly review the Hough transform followed by circular shortest path as a minimal path algorithm (*HCSP*). The circular Hough transform is used to detect circular arcs and subsequently the circles that these arcs belongs to. The minimal path algorithm extracts the contour of the actual circular or ovoid-like objects. This contour does not necessarily form a geometrical oval or a circle. First we review our implementation of the Hough transform and the proposed threshold (cf. eq. 1 and eq. 2). Next, we discuss the application of circular shortest path algorithm.

A. Hough Transform

Since many objects in images of molecular biology and especially the yeast cells taken here as a case study, are

978-1-4799-8637-8/15/$31.00 ©2015 IEEE

characterized by having a near circular shape [5], detecting circular arcs in such images would provide a good support in detecting the cells. The Hough transform is a well-known approach in the detection of circular arcs [6]. It is used in a 3-D parameter space with cube-like cells and an accumulator of the form $A(x, y, r)$, where x and y are coordinates of circle centers and r are the radii of these circles. Hough transform is used to obtain the center points as well as radii of all the circles in a binary image obtained through threshold and skeletonization of the intensity image. After filling the Hough accumulator, it is used to locate structures that form parts of a geometrical circle. The detected circular arcs belonging to that circle corresponds to the detected parts of the object (cell) contour. The center point of this circle corresponds to a point near the middle of the object. Hough values in the accumulator array are related to the number of pixels located in the circumference of a certain circle with a specified center c and radius r, where a maximum of $2\pi r$ pixels can belong to that circumference. A threshold value is set to allow detecting of structures that form a certain percentage of the circumference. This threshold starts at $2\pi r$ and decrements for smaller values of the radius. The threshold value is calculated according to eq. 1 and eq. 2.

$$T = 2\pi r - \{2\pi r \times \alpha + p\} \tag{1}$$

$$p = \beta \times (r_{max} - r_{min}) - r_{index} \tag{2}$$

where r is the radius of the circle, r_{max} and r_{min} are the maximum and minimum specified radius values, r_{index} is an index to track the current radius value. α and β are control values. Parameter p in eq.2 ensures that the threshold value is relatively high for smaller circles. This allows detecting some deformed shapes at a smaller radius $(r - 1)$ if not detected at radius r.

Starting from the largest radius and decrementing until the minimal radius, we loop through Hough values in every cell of the cube accumulator starting from the maximum Hough value until the specified threshold given in eq. 1. The parameter α is a weighting factor in the range $[0, 1]$ and is used to determine the percentage of pixels that must exist as edge pixels in order to consider them as being part of a closed circle; e.g. at $\alpha = 0.5$, at least 50% of the pixels must be edge pixels. The β parameter acts similar to α, except that it increases in value at the examination of circles with lower radius values, allowing more deformation at lower radii.

B. Circular Shortest Path

Once all the center of objects, i.e. cells, are estimated as circle centers, a polar image P_i relative to each circle center is created from the original intensity image [4]. The height of image P_i is determined based on the scale information of the images, and a priori generic knowledge on the objects, i.e. scale $\times$ maximum radius. The width of P_i is dependent of the length of the contour. In order to have a sufficient region to evaluate, we use a resampling length equal to twice the height of P_i. Next, minimal path algorithm finds the circular shortest path from the first to the last angular coordinate in image P_i.

The circular shortest path algorithm uses the concept of ordinary shortest path obtained with dynamic programming [7] but with the constraint that the first and last pixel are at the same level to assure a closed contour. The shortest path algorithm is applied just once starting at the first angular coordinate of each row in image P_i and only visiting the pixels that would have a high probability of belonging to the path. The global minimum of these shortest paths is guaranteed to be the global circular shortest path [8]. Our algorithm does not require visiting all the pixels while checking for shortest paths. It actually visits the pixels $(\frac{u^2 v}{2})$ instead of $u^2 v$ times stated in the standard circular shortest path algorithm [9], where u and v are the width and height of image P_i respectively. More details about the Hough transform implementation and the complete circular shortest path algorithm can be found in [2].

III. Contour Expansion

In some cases, the contour of the ovoid-shaped object is relatively thick in that it has a width of more than one pixel. This causes that minimal path algorithm not being able to locate the exact outer contour; and the extracted path seems to be biased towards the inner part of the object. This typically is the case for our yeast image datasets acquired under specific microscope settings. In these situations, the minimal path extraction is no longer an accurate contour extraction method. Therefore, we developed an additional algorithm that refines the initial detected contour.

In the following subsections, we provide a background and subsequently the control parameters are described. In the last subsection we discuss the contour expansion (*CE*) algorithm.

A. Background

Our goal is to obtain the exact contours of ovoid objects; here applied to yeast cells. The initial contour detected by *HCSP* is expanded out toward the background surrounding the object until it meets certain criteria. The expansion is realized using dynamic programming in the polar image P_i. Starting from the initial contour detected by *HCSP* [2], the contour is expanded in such a way that $r' \geq r$, where r' is the radial coordinate of the new contour pixel in image P_i and r is the initial radial coordinate. Every contour pixel considers a number of factors that decides whether it should be expanded to the next level or if it is a final contour point. The three parameters that control the contour expansion are the *resistance*, *limit* and *convergence*; these parameters are discussed hereafter.

B. Control parameters

The *resistance* parameter is the key player in this algorithm, its value is in the range $[0, 100]$, where 0 means that expansion is blocked, while 100 means the pixels in P_i would not block the expansion. At a value of 50 we threshold the image P_i at

the mean of its histogram and consequently the background pixels above that threshold value will block the expansion of the contour. At *Resistance* values below and above 50, we would threshold P_i according to eq. 3.

$$t = h_{min} + (h_\mu - h_{min}) \times \frac{r}{50} \quad \forall\, r \in [1, 50] \qquad (3)$$

$$t = h_\mu + (h_{max} - h_\mu) \times \frac{(r - 50)}{50} \quad \forall\, r \in (50, 100] \qquad (4)$$

where h_μ is the mean of the histogram of P_i, h_{min} is the minimum value of P_i histogram, and h_{max} is its maximum histogram value, r is the value of *resistance*, and t is the returned threshold value. The pseudocode of the threshold process is presented in **Box 1**.

Input : A polar image P_i of size $w \times l$
An integer $resistance \in [0, 100]$
Output: A binary image B_i of size $w \times l$

if $resistance \leq 50$ **then**
  $t \leftarrow h_{min} + (h_\mu - h_{min}) \times \frac{\text{resistance}}{50}$
end
else
  $t \leftarrow h_\mu + (h_{max} - h_\mu) \times \frac{(\text{resistance} - 50)}{50}$
end
$B_i \leftarrow$ threshold P_i at t.
return B_i

Box 1: Polar image threshold based on $resistance$

The *limit* control parameter indicates the maximally allowed radial coordinate for a new contour point. Specifically, it is the number of radial positions after the largest radial coordinate in the initial contour. After that level, the contour expansion is blocked.

The *convergence* parameter is the maximum number of iterations before the expansion is terminated. Subsequently, the contour at the last iteration is considered as a final contour. In general, the contour is saturated and reaches its final status after a few iterations. This will be shown in an example in the following sub-section.

C. The Expansion Algorithm

The flowchart in Fig. 1 explains how the expansion algorithm works and how the pixels are processed in the polar image P_i. The pseudocode of this algorithm is referred to in **Box 2**. The algorithm iterates until the *convergence* criteria is met, or until, in an iteration, no point is expanded anymore (cf. line 2). In each iteration, the *expanded* boolean variable is set to false (cf. line 3), and all the contour points are processed (cf. line 4). If any point is expanded at that iteration, then *expanded* variable is set to true, and the algorithm keeps running as long as the *convergence* criteria is not reached.

The processing procedure of the contour points at each iteration is shown in a separate block in the flowchart (*Process Contour Pixels*), and its pseudocode is presented separately as **Box 3**. This procedure evaluates each point in the contour and considers whether to expand it to the next radial coordinate or not. As long as the current contour point is not marked as a final contour point (cf. line 2), it is marked as final if the *limit* parameter is reached (cf. line 4); otherwise, the previous and the next neighbour contour points are checked if any of them is at a radial position lower than that of the current point (cf. line 7), in which case the point is skipped in this iteration. If none of these neighbour contour points is at a previous radial coordinate, these neighbour points are checked whether they are both marked as final points, forcing the current point to be final contour point as well (cf. line 8). If both neighbours are not marked as final points, the current point is marked as final if the point at the next radial position is blocked by the *resistance* parameter (cf. line 11). When the *resistance* control parameter does not block the expansion, the point is expanded to the next radial position unless one of the neighbours is marked as final and the current point has higher radial coordinate than that neighbour (cf. line 13), then the current point is marked as a final contour point (cf. line 14). Other than that, the point is expanded to the next radial position (cf. line 16), and normally this expanded contour point will be marked as a final contour point in the next iteration or when the *convergence* criteria is met before the next iteration.

Input : A binary image B_i of size $w \times l$.
Integer array CsP of initial contour, of size w.
Boolean array $isFinal$ of size w.
Boolean variable $expanded$.
Integer $limit \in [0, l]$.
Integer $covergence \geq 0$.
Output: Integer array $finalContour$ of size w.

initialize $isFinal$ and $expanded$ to $false$. $i \leftarrow 0$.
while *(*$expanded \neq false$ && $i < convergence$*)* **do**
  $i \leftarrow i + 1$
  $expanded \leftarrow false$
  $CsP \leftarrow$ `ProcessContour`(CsP) (**Procedure in Box 3**)
end
`/* Current contour is final */`
for *each element* p *in* CsP **do**
  $finalContour[p] \leftarrow CsP[p]$.
endfor
return $finalContour$

Box 2: Expand Contour Algorithm

Figure 2 illustrates an example of the initial estimate and the final expanded contour in a pathological cost image function. The image function shown in Fig. 2a is a sample of a polar image of ten angular and ten radial coordinates. The first and last angular coordinates are actually for the same point since this is a circular path. The *black* squares represent pixels with a value of 0, the *grey* squares represent pixels with a value

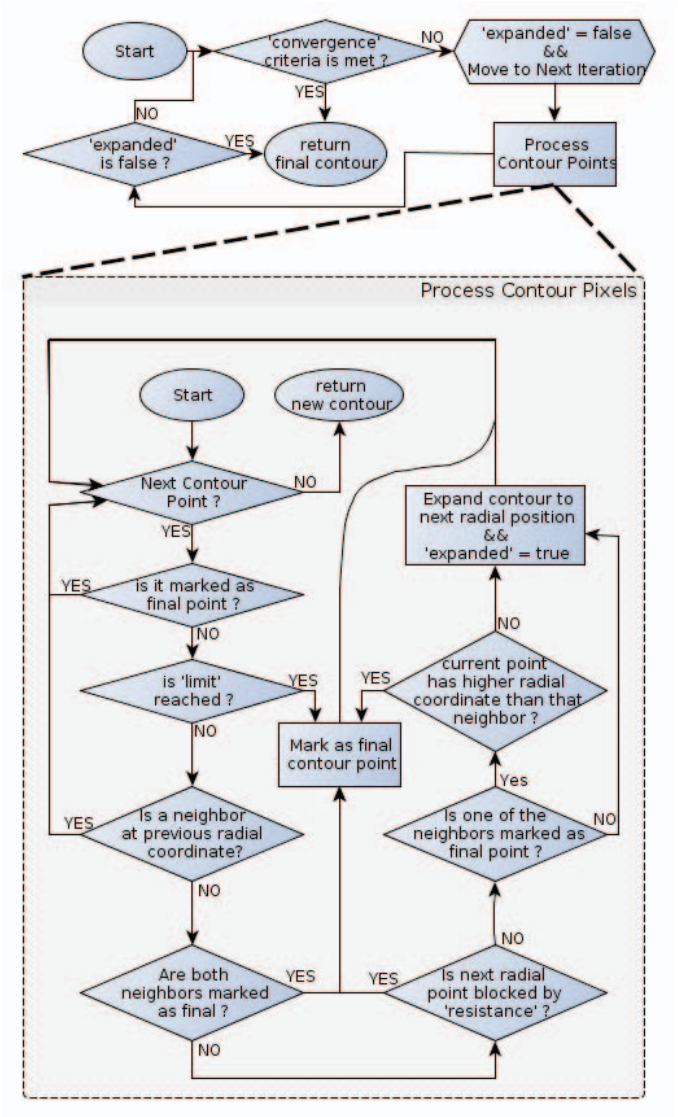

Fig. 1: Flow chart explaining the expansion algorithm

```
/* Process contour pixels                                    */
1  for each point c in CsP with radial coordinate r do
2      if isFinal[c] ← false then
3          if r ← limit then
4              isFinal[c] ← true
5          else
6              if (CsP[c − 1] ≠ (r − 1)) and
               (CsP[c + 1] ≠ (r − 1)) then
7                  if CsP[c − 1] ← true and
                   CsP[c + 1] ← true then
8                      isFinal[c] ← true
9                  else
10                     if (B_i[c, r + 1] ← Background
                       pixel then
11                         isFinal[c] ← true
12                     else
13                         if (isFinal[c − 1] ← true and
                           c has higher r than c − 1) or
                           isFinal[c + 1] ← true and c
                           has higher r than c + 1 then
14                             isFinal[c] ← true
15                         else
16                             Expand c to next radial
                               position (r + 1).
17                             expanded ← true
18                         end
19                     end
20                 end
21             end
22         end
23     end
24 end
```

Box 3: Process Contour Points Procedure

of 1, and the *white* squares represent pixels with a value of 11. In Fig. 2b, the initial contour points extracted by applying *HCSP* are displayed as *green* spots. To illustrate how the *CE* algorithm works, we apply contour expansion on this image with the parameters set as follows: *resistance = 0.5*, *limit = 1*, and *convergence = 5*. After the first iteration, the contour evolves to take the position illustrated in Fig. 2c. The points that were not considered for expansion in this iteration are left as *green* spots. Those expanded are displayed as *blue* spots and the *red* spot is the contour point marked as final. The point at the eighth angular coordinate is marked as final because the point at the next radial position was blocked by the *resistance* parameter. Figure 2d shows the contour points after the second iteration. Note here that the point at the seventh angular coordinate was marked as final because both its connected neighbours are marked final as well. Figure 2e shows the path after the third iteration. In the fourth iteration, the first four points and consequently the last contour point are marked as final points by the *limit* parameter (cf. Fig. 2f). In addition, the points at the fifth and ninth angular coordinates were marked final because their connected neighbours are marked final contour points as well. Since no contour point was expanded in this iteration, the contour expansion is terminated at this point, even before the *convergence* criteria is met.

In Fig. 3 an initial estimate is shown and the expanded contour for two yeast cells from a sample image acquired by a Zeiss LSM5 confocal microscope [10]. The *yellow* contours represent the initial estimate extracted by the Hough transform and circular minimal path algorithm, while the *red* contours represent the final expanded version of the contours acquired ensuing the application of our new contour expansion algorithm. Figure 3(a) shows the contours on the bright-field channel used for initial estimation and expansion of the contours, while Fig. 3(b) shows the same contours superimposed on the fluorescent counterpart of the cells acquired as a second image channel by the same microscope. The latter channel is used to obtain the meaningful measurements and pattern recognition

regarding protein levels and gene expression in the yeast cells. It is clear from this image, how much of the green signal would not be included in the cell measurement if contour expansion is not applied. This is even more quintessential when measuring proteins that are bound to the membrane of the cell.

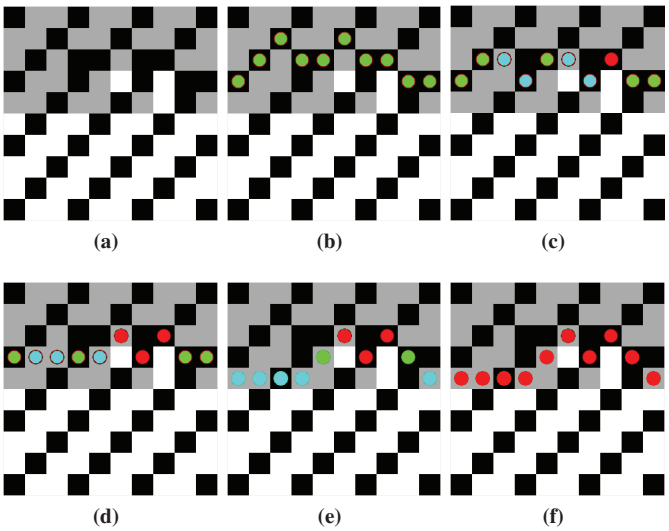

Fig. 2: Expanding the Initial estimate of the contour. *Black*, *grey* and *white* squares represent pixels with the values of 0, 1 and 11 respectively.

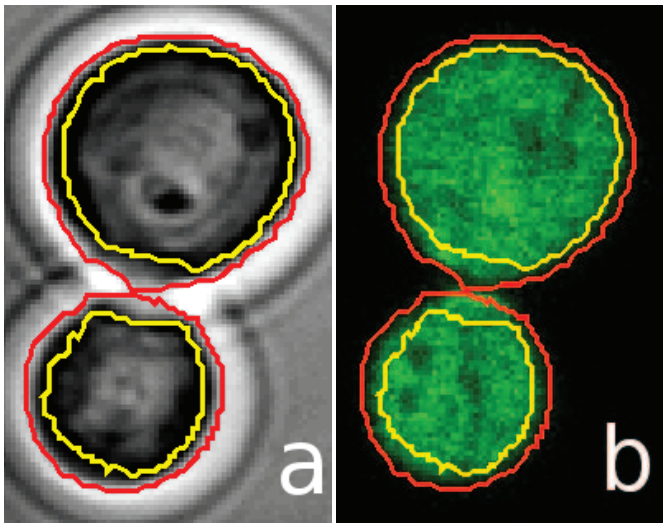

Fig. 3: The expanded contour is depicted in red, while the yellow contour is the initial estimate extracted by the Hough transform and Circular Shortest Path algorithm (*HCSP*). (a)- Bright-Field view of yeast cells. (b)- contours superimposed on a fluorescent channel of the same yeast cells.

IV. Results

In order to validate the new *HCSP-CE* method, its performance is compared to the previous *HCSP*, *CellSerpent* and *CellStat*. In this comparison we considered well-known metrics; i.e. the Pratt Figure of Merit (FoM) and F1-measure. Moreover, the tuning of parameters for the different segmentation methods and inspection of the segmented results were performed using *Paramorama* [11], a prototype developed to optimize parameters based on parameter sampling and interactive visual exploration. *Paramorama* is implemented as a plug-in for the *CellProfiler* biomedical image analysis framework [12].

The Pratt Figure of Merit (FoM) was considered because it has been reported as the best criterion to detect the differences in segmentation results [13]. Pratt's figure of merit provides a quantitative comparison of the results of the different contour detection methods by measuring the deviation of the output contours from a ground-truth. The ground-truth dataset were separately delineated using our dedicated structural annotation software (TDR) with a digitizer tablet (WACOM, Cintiq LCD-tablet) [14]. The Pratt measure is computed according to eq. 5.

$$P = \frac{1}{max(I_A, I_I)} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha.d^2(i)} \tag{5}$$

where I_A being the number of detected edge points, I_I the edge points in the ideal ground-truth image, α a scaling factor or an empirical calibration constant set to the optimal value established by Pratt, $\alpha = 1/9$ [15]. $d(i)$ is the distance between the detected edge point and the nearest edge point in the ideal ground-truth [16]. Pratt's figure of merit is an indicator of the quality of edge, and reflects the overall behaviour of the distances between the edges, being a relative measure, which varies in the range [0,1], where 1 represents the optimal value, i.e., the detected edges coincide with the ground truth [17].

Figure 4 shows the number of detected cells that gained a higher Pratt score using *HCSP* method followed by contour expansion (*CE*) algorithm (*HSCP-CE*). These cells were obtained from a dataset of 312 *Saccharomyces cerevisiae* yeast cells distributed over 14 bright-field images acquired by an embedded technique in a Zeiss LSM 5 Exciter-AxioImager M1 confocal microscope. Compared to *CellSerpent*, the *HCSP-CE* method has 163 cells with better Pratt scores, referred to as wins, while *CellSerpent* has only 138 wins. In addition, comparison with *CellStat* leads to 284 wins of cells with higher Pratt scores in the *HCSP-CE* method, with only 16 wins for *CellStat*. This is expected as the method implemented in *CellStat* looks for the minimal path as the final contour of yeast cells in a similar way to that of *HCSP*, which scored 18 wins. On the other hand, *CellSerpent* performs better than *CellStat* because its method is based on Active Contour models for cell detection.

In Table 1 a comparison between the four used segmentation methods on the dataset of 312 cells is listed. In the comparison Pratt score is used as well as the F1-measure. The F1-measure, also known as F1-score, is a measure of accuracy used in many domains including the comparison of segmentation algorithms. It considers the precision and recall in its computation [18]. The precision is the number of correctly detected cells divided by the number of detected objects by the segmentation method.

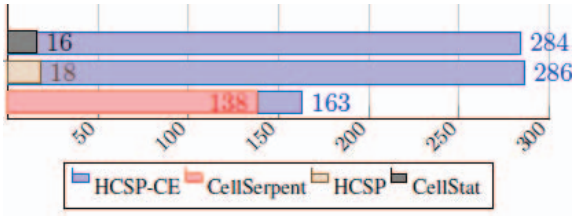

Fig. 4: Number of cells with higher Pratt score (wins).

The recall is the number of correctly detected cells divided by the number of actual cells. We considered a cell correctly detected if the centroid of its detected binary mask also exists as a foreground pixel in the ground-truth binary mask. The F1-score is computed according to eq. 6.

$$F1 = 2.\frac{precision.recall}{precision + recall} \tag{6}$$

The *HCSP-CE* segmentation method has an F1-score of 0.93, and an average Pratt score of 0.58, followed by *CellSerpent* with an F1 measure of 0.90 and average Pratt of 0.55. *CellStat* follows with an F1-measure of 0.67 and average Pratt of 0.15. *HCSP* scores similar F1-measure to that of *HCSP-CE* as they both use the same underlying detection method, however the Pratt score of *HCSP* is 0.19, which is noticeably worse than *HCSP-CE* and close to that of *CellStat* as they both look for minimal paths for contour extraction.

	HCSP-CE	**CellSerpent**	**HCSP**	**CellStat**
Detected	**322**	359	**322**	231
Correct	296	**302**	296	181
Precision	**0.92**	0.84	**0.92**	0.78
Recall	0.95	**0.98**	0.95	0.58
F1	**0.93**	0.90	**0.93**	0.67
Pratt	**0.58**	0.55	0.19	0.15

Table 1: F1-measure and average Pratt Score of different segmentation methods on a yeast dataset

V. Conclusion

In this work, a novel contour expansion algorithm is proposed to refine the contour of ovoid-shaped objects such as yeast cells in bright-field microscopy images. The core of the method is the processing of pixels in the polar image ensuing the application of Hough transform and circular minimal path algorithm. In addition, the results of this new method were compared to similar software packages. It achieves an F1 score of 0.93 and an average Pratt score of 0.58. These results are higher compared to the counterpart software tested in this work, i.e, *CellSerpent*, *CellStat* and *HCSP*. Moreover, the algorithm is implemented as part of a complete pipeline platform for yeast analysis [19].

Since this algorithm provides a more accurate contour estimate, in this manner consequently precise measurements of the objects can be accomplished so that machine learning techniques can better recognize the subtle patterns within the data. As the method is geared to ovoid like object, it can be easily generalized to other images of other modalities.

Acknowledgment

We would like to thank Paul van Heusden for providing the yeast cell image dataset, and the developers of *CellStat* at the Fraunhofer-Chalmers Centre (http://www.fcc.chalmers.se/sys) and *CellSerpent* at the University of Graz (http://microscopy.uni-graz.at) for providing their software. This research is partially supported by the Erasmus Mundus, *JOSYLEEN* program, the *Landelijke Stichting voor Blinden en Slechtziendenand* (LSBS) and the *Raymond Sackler* organization.

References

[1] J. Gerbrands, E. Backer, and W. Van Der Hoeven, "Quantitative evaluation of edge detection by dynamic programming," in *Pattern Recognition in Practice II*. Amsterdam, The Netherlands, 1986, pp. 91–99.

[2] M. Tleis and F. J. Verbeek, "Extracting contours of oval-shaped objects by hough transform and minimal path algorithms," in *Sixth International Conference on Digital Image Processing*. International Society for Optics and Photonics, 2014, p. 915903.

[3] K. Bredies and H. Wolinski, "An active-contour based algorithm for the automated segmentation of dense yeast populations on transmission microscopy images," *Computing and Visualization in Science*, vol. 14, no. 7, pp. 341–352, 2011.

[4] M. Kvarnstrom, K. Logg, A. Diez, K. Bodvard, and M. Kall, "Image analysis algorithms for cell contour recognition in budding yeast," *Opt Express*, vol. 16, no. 17, pp. 12–943, 2008.

[5] H. Feldmann, *Yeast. Molecular and Cell Biology*. Willey-BlackWell, 2010.

[6] R. Gonzalez and R. Woods, *Digital Image Processing*, 3rd ed. Pearson Education, 2008.

[7] M. Buckley and J. Yang, "Regularised shortest-path extraction," *Pattern Recognition Letters*, vol. 18, no. 7, pp. 621 – 629, 1997.

[8] B. Appleton and C. Sun, "Circular shortest paths by branch and bound," *Pattern Recognition*, vol. 36, no. 11, pp. 2513 – 2520, 2003.

[9] C. Sun and S. Pallottino, "Circular shortest path in images," *Pattern Recognition*, vol. 36, no. 3, pp. 709–719, 2003.

[10] Z. Inc., "Zeiss 5 exciter laser scanning microscope manual," 2012.

[11] A. J. Pretorius, M.-A. Bray, A. E. Carpenter, and R. A. Ruddle, "Visualization of parameter space for image analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2402–2411, 2011.

[12] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat *et al.*, "Cellprofiler: image analysis software for identifying and quantifying cell phenotypes," *Genome biology*, vol. 7, no. 10, p. R100, 2006.

[13] S. Chabrier, H. Laurent, C. Rosenberger, and Y. J. Zhang, "Supervised evaluation of synthetic and real contour segmentation results." *14th European Signal Processing Conference (EUSIPCO)*, 2006.

[14] F. J. Verbeek and P. J. Boon, "High-resolution 3d reconstruction from serial sections: microscope instrumentation, software design, and its implementations," in *International Symposium on Biomedical Optics*. International Society for Optics and Photonics, 2002, pp. 65–76.

[15] I. E. Abdou and W. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 753–763, 1979.

[16] F. J. Theis and A. Meyer-Bse, "Biomedical signal analysis: Methods and applications," *MIT Press*, 2010.

[17] I. Boaventure and A. Gonzaga, "Method to evaluate the performance of edge detector," in *Proceedings of the The Brazilian Symposium on Computer Graphics and Image Processing*, 2009.

[18] X. Hou, A. Yuille, and C. Koch, "Boundary detection benchmarking: Beyond f-measures," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, pp. 2123–2130.

[19] M. Tleis, G. Anemaet, P. van Heusden, and F. Verbeek, "Image analysis platform for yeast biologists," in *Advances in Biomedical Engineering (ICABME), 2nd International Conference*. IEEE, 2013, pp. 105–108.