Automated image analysis- and soft computing-based detection of the invasive dinoflagellate Prorocentrum minimum (Pavillard) Schiller

A. Verikas a,b,⁎, A. Gelzinis a, M. Bacauskiene a, I. Olenina c,d, S. Olenin d, E. Vaiciukynas a

aDepartment of Electrical & Control Equipment, Kaunas University of Technology, Studentu 50, LT-51368, Kaunas, Lithuania
bIntelligent Systems Laboratory, Halmstad University, Box 823, S-30118 Halmstad, Sweden
cDepartment of Marine Research, Environmental Protection Agency, Taikos str., 26, LT-91144, Klaipeda, Lithuania
dCoastal Research and Planning Institute, Klaipeda University, H. Manto 84, LT-92294, Klaipeda, Lithuania

1. Introduction

Identification and counting of phytoplankton cells is being used for many purposes: monitoring of toxic algal blooms, assessment of water quality parameters, studies of long-term changes in aquatic ecosystems, etc. Much work in this area still remains in the form of conventional microscope analysis and is very time consuming and labor intensive. For example, derivation of quantitative concentration estimates requires recognizing and counting cells in tens or even hundreds of microscopic views. Robust automated image analysis-based system capable of recognizing different plankton species would be of great help and would enable analysis at much larger scales (Culverhouse et al., 2006).

While automated analysis of zooplankton images is a rather active field of research, developments in the field of automated analysis of phytoplankton images are rather limited. Work by Gorsky, Guilbert, and Valenta (1989) is one of pioneering attempts in this area. Using simple geometric features the authors were able to distinguish between three species of distinct size and shape. Sosik and Olson (2007) developed a system for automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. Embleton, Gibson, and Heaney (2003) trained a multilayer perceptron (MLP) to identify four species in lake water samples. Fourier descriptors, geometrical features, and features characterizing the grey level distribution in a region were used to select a set of suitable features for training the MLP. Cuiping, Chenhui, Huizhen, and Lin (2010) by using a variety of moment features in a support vector machine (SVM) with a radial basis function kernel were able to distinguish between 241 species of marine phytoplankton with 89% accuracy. Blaschko et al. (2005) achieved 50% to 70% classification accuracy in a task of phytoplankton categorization into 12 classes plus an “unknown” class. A large variety of features: shape features, moments, texture features, contour features (780 features in total) were used. Several classifiers, including decision trees, naive Bayes, ridge linear regression, k-NN, SVM, and bagged as well as boosted ensembles were explored. SVM was found to be the best classifier for the task. Culverhouse, Williams, Reguera, Herry, and Gonzalez-Gil (2003) studied the classification accuracy achieved by the neural network committee-based automated system DiCANN (Ellis, Simpson, Culverhouse, & Parisini, 1997) and argued that accuracy of about 72% achieved by the system in a six-class phytoplankton categorization task was similar.
to the accuracy achieved by the trained personnel. To capture more information for discrimination between five classes of phytoplankton, Rodenacker, Hense, Tting, and Gais (2006) applied fluorescence imaging in their image acquisition system. Sosik and Olson (2007) presented, perhaps, the most elaborated study regarding multiclass phytoplankton categorization using data obtained from Imaging FlowCytobot (Olson & Sosik, 2007). A combination of video and flow cytometric technologies is used in the Imaging FlowCytobot. Images of $1380 \times 1034$ pixels and resolution equal to $3.4$ pixels/$\mu m$ were used in the study. In total, 6600 visually identified and manually inspected images distributed across 22 categories were used in the study. Both training and test sets consisted of 3300 images. In total, 210 features characterizing geometry, shape, symmetry, texture, and invariant moments of objects were extracted and 131 features were selected and used in an SVM classifier for the categorization. Ten-fold cross validation on the training set was applied to select the hyper-parameters of the classifier. The overall accuracy of 88% was achieved on the test set. While the obtained accuracy is very encouraging, one very important task–image segmentation—is not addressed in the article.

The analysis shows that classification accuracy achieved when solving phytoplankton classification problems varies in a broad range depending on the task and the data. Due to the variety of tasks solved and data sets used, comparison of different approaches is a rather complicated matter. It is worth noting that one very important problem, namely object detection, is almost never addressed in the literature. Robust object detection, however, is a prerequisite for obtaining a robust system for automated analysis of plankton images, especially when rather simple imaging systems, generating images exemplified in Fig. 1 (there are overlapping objects and/or objects touching each other in the images), are used. Touching organisms bring difficulties in automated categorization and counting of objects. Irigoien et al. (2009) point out that a linear relation between the number of items and the automatic counting holds if the percentage of image area occupied by the items remains below 3%. Above this threshold, automatic counting underestimates abundance due increased percentage of organisms touching each other. In contrast to many previous techniques, a simple imaging system is used to obtain phytoplankton images in this study. A long-term goal of this work is an automated system for image analysis and soft computing-based detection, recognition and counting of objects representing different phytoplankton species. An automated recognition and counting system can essentially facilitate treatment of phytoplankton samples that may be useful in many practical applications (e.g. rapid ship ballast water analysis, harmful algae bloom monitoring, etc.). This article is limited, however, to analysis of one invasive species, a dinoflagellate Prorocentrum minimum (Pavillard) Schiller, which is known to cause harmful blooms in many estuarine and coastal environments (Azanzaa, Fukuyob, Yapa, & Takayama, 2005; Olenina et al., 2010; Tango et al., 2005). More specifically, this article is concerned with the very important object detection task in “over-crowded” images. A new technique, combining phase congruency-based detection of circular objects, stochastic optimization, and image segmentation was developed for solving the task.

2. Data

2.1. Sample preparation

Phytoplankton samples used for obtaining images were received from: (1) natural south-eastern Baltic Sea phytoplankton containing P. minimum cells, (2) cultured P. minimum; and, (3) natural phytoplankton mixed with cultured P. minimum. All samples were fixed with acetic Lugols solution (in proportion: 0.5 ml of solution for 100 ml of sample). Cells of P. minimum in natural phytoplankton varied in shape from triangular- to oval- and heart-shaped, while cultured cells were mostly oval. The length of P. minimum varied from 14 to 22 $\mu m$, while the width ranged from 12 to 18 $\mu m$.

2.2. Images

Images for the analysis were obtained from a colour camera of 1280 x 960 pixels attached to an inverted microscope with magnification of 400x. Images were recorded in the RGB colour space and then converted to the more homogenous Lab colour space (Wysekcki & Stiles, 1982) for further analysis. The background colour in the images may differ significantly depending on the solution applied in the sample preparation process. Figs. 1 and 2 exemplify the variation. In Fig. 2, three decision classes are identified, namely the invasive (an example of this class is enclosed in a rectangle), native (inclosed in a circle), and other (enclosed in a pentagon). Invasive (P. minimum) cells dominate the image. While the main task of this work is detection (extraction) of objects in phytoplankton images, in future work, algorithms capable of distinguishing P. minimum cells from objects coming from the other classes will be developed.
Presence of pixels in a central part of cells (both invasive and native) with \( B \) component values significantly lower than in the background is one characteristic feature of the images. Otherwise, information on colour does not bring very much help for the analysis. In total, 114 images have been collected for the analysis at several occasions.

3. Methods

We use original Lab images as well as preprocessed images, to solve the object detection task. A clear 2D shell is a characteristic feature of the invasive cells. To emphasize the feature, we applied preprocessing via the phase congruency-based enhancement of image edges (Kovesi, 2000).

3.1. Image preprocessing

The phase congruency idea is based on the assumption that features (edges, corners) are perceived in image points, where the signal Fourier components are maximally in phase. Kovesi (2000, 2003) proposed using the following phase congruency measure at a location \( x \) in the signal:

\[
PC(x) = \frac{\sum_n w(x) |A_n(x)| (\cos(\phi_n(x) - \overline{\phi}(x)) - \sin(\phi_n(x) - \overline{\phi}(x))) - T_n |}{\sum_n |A_n(x)| + \varepsilon}
\]

(1)

where \( A_n(x) \) and \( \phi_n(x) \) is an amplitude and a phase angle, respectively, of the \( n \)th Fourier component at the location \( x \), \( \overline{\phi}(x) \) is the amplitude weighted mean phase angle, \( w(x) \) is a parameter weighting frequency spread (phase congruency over a few frequencies is less significant than congruency over many frequencies), \( \varepsilon \) is a small constant preventing division by zero, \( T \) is a parameter meaning that only energy values exceeding \( T \) influence \( PC \) (\( T \) depends on the estimated noise level), and \( |y| = y \) if \( y > 0 \) and \( |y| = 0 \) otherwise.

Values of the \( PC \) measure belong to the interval \([0,1]\). If all the Fourier components are in phase \( PC \) approaches 1 and 0, if there is no coherence in phase. Since the measure is independent of the signal magnitude, the measure is invariant to variations in contrast and/or illumination. This is a very important property. The task of Fourier analysis applied here is to give local frequency information. As suggested by Kovesi (2000), we obtain this information by applying a bank of Gabor filters tuned to different spatial frequencies rather than via Fourier analysis. The appropriate \( T \) value is usually obtained from the responses of the filters applied to the image.

When working with 2D images, local energy is first computed in several orientations \( \theta \) using 2D Gabor filters (Kovesi, 2000) and 2D phase congruency \( PC_2(x) \) is then computed according to the following equation:

\[
PC_2(x) = \frac{\sum_i \sum_j W_{ij} |A_{ij}(x)| \Delta \Phi_{ij}(x) - T_{ij}}{\sum_i \sum_j |A_{ij}(x)| + \varepsilon}
\]

(2)

where the index \( i \) runs over orientations and

\[
\Delta \Phi(x) = \cos(\phi_n(x) - \overline{\phi}(x)) - | \sin(\phi_n(x) - \overline{\phi}(x)) |)
\]

We use a \( PC_2 \) image to extract objects in an original phytoplankton image. In addition to information on phase congruency available from a \( PC_2 \) image, we also utilize information of phase congruency variation with orientation. We obtain this information from an angle image \( \Phi \) (an angle of the principal axis about which the phase congruency moment is minimized Kovesi, 2003) and an image of the magnitude of the maximum moment \( M \) (the moment about an axis perpendicular to the principal axis Kovesi, 2003) of phase congruency. Values of \( \Phi \) and \( M \), computed in each image pixel, are given by Kovesi (2003):

\[
\Phi = \frac{1}{2} \tan^{-1} \left( \frac{b}{\sqrt{b^2 + (a - c)^2}} \right) \quad \text{and} \quad a - c \quad \text{for} \quad b \neq 0
\]

(4)

\[
M = \frac{1}{2} \left( c + a + \sqrt{b^2 + (a - c)^2} \right)
\]

(5)

where

\[
a = \sum_n |PC(\theta)\cos(\theta)|^2
\]

(6)

\[
b = 2 \sum_n |PC(\theta)\cos(\theta)||PC(\theta)\sin(\theta)|
\]

(7)

\[
c = \sum_n |PC(\theta)\sin(\theta)|^2
\]

(8)

where \( PC(\theta) \) is the phase congruency value computed at orientation \( \theta \) using Eq. (2).

3.2. Determining centre points of circular-shaped objects

\( P. \) minimum cells are approximately circular- or oval-shaped, see Figs. 2 and 1. To facilitate the cell detection step, a technique for determining centre points of circular objects was developed, based on \( M \) and \( \Phi \) images. One can consider image \( M \) as an image of edge clarity (certainty) values in each pixel of the original image, while image \( \Phi \) reflects edge direction in each image pixel. Fig. 3 presents an example of an original image and corresponding \( M \) and \( \Phi \) images.

Image \( C \) reflecting clarity (certainty) of a centre point in each image point is created first. The size of \( C \) is equal to the size of the original image, \( m \times n \). We start by setting \( C_{ij} = 0 \), \( \forall i, j \). Image \( C \) is then obtained by applying the following algorithm, where \( r \) is the supposed radius of an object.

\[\begin{align*}
&\text{for } i = 1, \ldots, m \\
&\text{for } j = 1, \ldots, n \\
&\quad i^* = i + \sin(\Phi_{ij} + 90), \quad j^* = j + \cos(\Phi_{ij} + 90) \\
&\quad i^* = i - \sin(\Phi_{ij} + 90), \quad j^* = j - \cos(\Phi_{ij} + 90) \\
&\quad \text{if } 0 < i^* \leq m \text{ and } 0 < j^* \leq n \\
&\quad \quad C_{ij^*} = C_{ij^*} + M_{ij} \\
&\quad \text{end (if)} \\
&\quad \text{if } 0 < i^* \leq m \text{ and } 0 < j^* \leq n \\
&\quad \quad C_{ij^*} = C_{ij^*} + M_{ij} \\
&\quad \text{end (if)} \\
&\text{end (for)} \\
&\text{end (for)}
\end{align*}\]

Fig. 4 illustrates the generation process of \( C \) based on information available in \( \Phi_{ij} \) and \( M_{ij} \). The idea is similar to that used in the Hough transform. However, the technique we developed is much more robust to noise. Based on information available in \( \Phi_{ij} \), the normal to the edge is determined and the centre certainty \( C_{ij^*} \) and \( C_{ij^*} \) are increased by \( M_{ij} \) at two possible centre positions, \( [i,j] \) and \( [i^*, j^*] \). For the case shown in Fig. 4, many edge pixels will contribute to the centre position at \([i^*, j^*]\). Fig. 5 presents the C image computed from \( M \) and \( \Phi \) images shown in Fig. 3 using \( r = 45 \). As can be seen from Fig. 5, the approximate centre position manifests itself by high \( C_{ij} \) values.

Since shape of \( P. \) minimum cells deviates from a circle, many pixels in the vicinity of the centre position exhibit high \( C_{ij} \) values. Moreover, \( P. \) minimum cells differ in size. Therefore, \( C \) images are filtered by a rotationally symmetric Gaussian low-pass filter with the standard deviation \( \sigma = 3 \). Fig. 6 presents five examples of a filtered \( C \) image computed from \( M \) and \( \Phi \) images shown in Fig. 3 for five different values of \( r \): 35, 40, 45, 50, and 55.
As can be seen from Fig. 6, the C image is rather insensitive to the r value. When varying r from 40 to 50, rather similar C images, with a clear centre position, are obtained.

3.3. Determining contour of circular objects

3.3.1. An iterative algorithm

Having a centre position of a circular-shaped object, a square of size $(2r+\beta) \times (2r+\beta)$, where $\beta$ is a parameter, is clipped out around the centre together with corresponding M image for further analysis, see Fig. 7. The analysis is based on the assumption that a contour does not contain high curvatures and an approximate object radius r is known. When determining an approximate object contour, we assume that the contour is given by the brightest line in the corresponding M image converted to the polar coordinate system, using the determined centre position. Fig. 8 presents the M image shown in Fig. 7 converted to the polar coordinates system, where the angle $\alpha$ is computed clockwise starting at “3 o’clock” and bottom corresponds to the centre position in the original image.

To emphasize image elements making horizontal lines and to suppress isolated image elements, M image in the polar coordinate system is filtered by a Gaussian filter with a much larger standard deviation in the angle direction than in the r direction. The M image shown in Fig. 8 filtered by a Gaussian filter with a standard deviation equal to 8 and 1 in the angle and r directions, respectively, is given in Fig. 9.

An iterative algorithm is used to find a contour line $c$. The algorithm aims at placing a contour line in high intensity points of a filtered M image $E$ in the polar coordinate system. For each $\alpha$, the initial contour position $c^{0}_{\alpha}$, $\alpha = 0, \ldots, 360^{\circ}$, is given by the largest r value, such that $E_{2\pi, r} > \text{mean}(E_{2\pi, r})$:

$$c^{0}_{\alpha} = \max(\arg r, E_{2\pi, r} > \text{mean}(E_{2\pi, r}))$$  \hspace{1cm} (9)

where $E_{2\pi, r}$ is an image $E$ element – intensity at angle $\alpha$ and radius r. At each iteration, the contour line $c$ is updated by $\Delta c$:

$$c^{i+1} = c^{i} + \Delta c$$  \hspace{1cm} (10)

where i denotes an iteration index and $\Delta c = \text{Bstep} \ast G$  \hspace{1cm} (11)

where Bstep stands for a vector of best step sizes at different $\alpha$ and the convolution operation, denoted by $\ast$, with a Gaussian filter G is applied aiming the adjustments at different $\alpha$ to be dependent on neighbours. The best step size at angle $\alpha$ is given by:

$$\text{Bstep}_{\alpha} = \arg \max_{\Delta c} \sum_{j=0}^{w} E_{2\pi, (c^{i+1} + \text{step}_{\alpha})}$$  \hspace{1cm} (12)

where G is a Gaussian windows of $2w+1$ size with maximum at $G(0)$, step is varied from $-\text{step}_{\alpha}$ to $\text{step}_{\alpha}$ and $\text{step}_{\alpha}$ was set to 5.

Fig. 10 presents the initial contour line (shown in yellow) and the contour line determined after 15 iterations (shown in blue). The determined contour superimposed on the M image and the original image is shown in Fig. 11. As can be seen from Fig. 11, noise in
the bottom part of the contour line was eliminated. However, the contour line does not follow the “true contour” accurately in the region of $140^\circ - 220^\circ$.

3.3.2. Contour correction by stochastic search

The proposed iterative contour detection algorithm experiences difficulties when contour lines of other objects appear in the vicinity of a contour line being considered. The dark object in the left upper part of Fig. 3 is an example. As can be seen from the image shown in the middle of Fig. 3—the $M$ image—other objects appear above and below the object being considered. Fig. 12 presents the object in the polar coordinate system, where the contour line is seen in the middle of the figure and the “other objects” appear at about $90^\circ$ and $270^\circ$. These “other objects” hinder the contour line being searched from following the actual contour line. This deficiency is clearly seen in Fig. 13, where the initial contour line given to the algorithm is shown in yellow (light), while in blue shown is the contour line determined by the algorithm. As can be seen from Fig. 13, the “other objects” appearing at about $90^\circ$ and $270^\circ$, disturb the algorithm.

Stochastic search is applied to eliminate this deficiency. The main discrepancy of the stochastic search-based algorithm from the one presented in Section 3.3.1 is computation of the adaptation step. Instead of using the whole contour length, only one, randomly selected, angle value is used to compute the adaptation step in the stochastic version of the algorithm. Thus, at $i$th iteration, the angle $a_r$ and the width $w_r$ of the Gaussian window are randomly selected and the adaptation step $B_{step_a}$ is computed analogously to (12):

$$B_{step_a} = \arg \max_{\theta_p} \sum_{j=\theta_p}^{j+w_r} \sum_{j=-w_r}^{j \neq \theta_p} E_{f(x_{j+step}G_{ij})} \quad (13)$$

Fig. 7. An image fragment containing an object (left) and its corresponding $M$ image (right).

Fig. 8. An object in the polar coordinate system.

Fig. 9. A filtered $M$ image of the object shown in Fig. 8.

Fig. 10. The initial contour line shown in yellow (light) and the final shown in blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
where $\tilde{f} = \alpha' + j$. If $B_{\text{step}} \neq 0$, contour adaptation takes place at $\alpha'$ and in the neighbourhood of $\alpha'$, given by $\alpha' \pm w'$:

$$c_j^{i+1} = c_j^i + B_{\text{step}} \cdot G(\tilde{f} - \alpha'), \quad \tilde{f} = \alpha' - w', \ldots, \alpha' + w'$$

The width of the Gaussian window $w'$ is randomly selected from the interval $w_{\text{min}} = 20$ and $w_{\text{max}} = 100$. The search continues for $n_s$ successful iterations, $B_{\text{step}} \neq 0$, or for $n_o$ iterations in total. Values of $n_s = 40$ and $n_o = 1000$, found experimentally, worked well in all the tests. The line shown in green in Fig. 13 represents the contour line found by the algorithm. The improvement, if compared to the contour line determined by the iterative algorithm should be obvious from Figs. 13 and 14, where the two contour lines were superimposed on the original image.

3.4. Segmenting original images

To detect objects, not only phase congruency preprocessed images, but also original colour images were used. Since the RGB colour space is rather nonhomogeneous, original RGB images were first transformed to the Lab colour space and then used in the segmentation process, based on the Fuzzy C-Means clustering (FCM) algorithm (Bezdek, 1981). In fact, segmentation into two clusters, “objects” and “background” is needed in this study. However, aiming to obtain more accurate object boundaries, a larger number of clusters was used in the FCM algorithm. Then, in the next step, clusters were merged and too large regions, representing the background, were eliminated. Fig. 15 presents an image and five sub-images, clusters of pixels, obtained by grouping pixels of the image into five clusters. It is easy to see that clusters in the middle column belong to the background, while pixels of the other three clusters make objects. The final clustering result is shown in the middle of Fig. 16, while on the right-hand side shown is the result of clustering into two clusters. It is easy to see that the middle image reflects objects seen in the original image better than the one shown on the right.
3.5. Combining analysis results for object detection

It is well known that fusing results obtained from different analysis techniques is an efficient way to improve the accuracy of the analysis (Aslantas & Kurban, 2010; Verikas, Lipnickas, Malmqvist, Bacauskiene, & Gelzinis, 1999; Verikas, Gelzinis, Kovalenko, & Bacauskiene, 2010). Therefore, the techniques developed to find centres of circular objects, to determine contour lines of such objects, and to segment an image into objects and background were combined into one algorithm. The algorithm can be summarized by the following steps:

(i) create an image of objects \( O \) by segmenting an \( \text{Lab} \) image;
(ii) determine centres of circular objects;
(iii) determine contour lines of the circular objects;
(iv) eliminate pixels belonging to the circular objects from the image \( O \); this step ends the phase I of the algorithm.
(v) eliminate pixels belonging to very small and very large objects from the image \( O \); objects much smaller than \( P. \) minimum cells are considered as being very small;
(vi) if the image \( O \) contains at least one object, take an object from \( O \) and perform the following steps (phase II)
   (a) determine a centre point of the object; if the object is larger than twice the average \( P. \) minimum cell size, the most distant point from the object edge is assumed to be the centre point;
   (b) determine a contour line of the object;
   (c) eliminate pixels of the object from \( O \);
   (d) apply the morphological opening operation on the image \( O \);
   (e) eliminate very small objects from \( O \);
   (f) if \( O \) contains at least one object goto Step (vi); otherwise stop.

We do not consider very small and very large objects. Objects with centre points too close to image boundaries are also left unprocessed.

4. Experimental investigations

4.1. Parameters of the algorithms

There are several parameters governing the behaviour of the algorithms. The appropriate values of the parameters are chosen experimentally. Using the a priori knowledge about the size of \( P. \) minimum cells, the supposed radius of an object \( r \) was set to \( r = 45 \). Three Gaussian filters are used by the algorithms. A \( C \) image is filtered by a rotationally symmetric Gaussian low-pass filter with the standard deviation \( \sigma \). The parameter \( \sigma \) was set to \( \sigma = 3 \) and worked well in all the tests. An \( M \) image in the polar coordinate system is filtered by a Gaussian filter with a much larger standard deviation in the angle direction than in the \( r \) direction. The standard deviation equal to 8 and 1 in the angle and \( r \) directions, respectively, was a good choice. The width of the Gaussian window \( w \) used by the stochastic contour correction algorithm is randomly selected from the interval \( w_{\min} = 20 \) and \( w_{\max} = 100 \). The number of successful iterations \( n_s \) and the total number of iterations \( n_o \) used by the algorithm were set to \( n_s = 40 \) and \( n_o = 1000 \). Objects larger than five times the average cell size are considered as being very small.
large. It is worth mentioning that the algorithms are rather insensitive to the choice of the parameters.

4.2. Results

The performance of the algorithm is evaluated by computing the percentage of detected *P. minimum* cells among all detected objects. This measure is equivalent to the sensitivity measure used to characterize the performance of a binary classifier. Other objects not belonging to the class of *P. minimum* cells are also detected. Algorithms to discriminate between *P. minimum* cells and other objects will be developed in future studies.

Fig. 17 presents an image fragment used in previous examples. Two objects were detected in this image. The object detected in phase I is shown in blue, while the other one, detected in phase II, is shown in red. As can be seen from Fig. 17, the *P. minimum* cell present in the image was detected in phase I. The object detected in phase II does not belong to the class of *P. minimum* cells.

Fig. 18 presents an example of object detection results for a full-size image. As can be seen from Fig. 18, majority of small “uninteresting” objects were eliminated and all *P. minimum* cells, except those with centre points too close to the image boundaries, were detected. All objects detected in phase II do not belong to the class of *P. minimum* cells. Figs. 19–22 present more examples of object detection results. Objects marked by yellow crosses and do not having a detected contour line were left aside from the analysis, as being too close to image boundaries if compared to the chosen value of parameter $r$.

Fig. 17. An image fragment with objects detected in phase I (blue) and phase II (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 18. An image containing objects detected in phase I (blue) and phase II (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 19. An image containing objects detected in phase I (blue) and phase II (red). Objects marked by yellow crosses were not analyzed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 20. An image containing objects detected in phase I (blue) and phase II (red). Objects marked by yellow crosses were not analyzed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 21. An image containing objects detected in phase I (blue) and phase II (red). Objects marked by yellow crosses were not analyzed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
In total, 114 full-size images were processed and the detection results were verified by manual inspection. The manual inspection has shown that there are 2088 P. minimum cells in these images in total. The algorithms found 2412 objects. Among these objects, 1947 were P. minimum cells. Thus, 93.2% of P. minimum cells present in the images were detected. Bearing in mind the simplicity of the imaging system used the result is rather encouraging.

5. Conclusions and future work

This article addressed a very important object detection task, which is to be solved if one is aiming to develop an automated system for detection, recognition, and derivation of quantitative concentration estimates of different phytoplankton species using ordinary inverted microscope phytoplankton images. It is worth noting that the object detection problem is almost never addressed in the literature. Robust object detection, however, is a prerequisite for obtaining a robust system for automated analysis of plankton images.

This article was mainly concerned with detection of objects representing one invasive species, a dinoflagellate Prorocentrum minimum (Pavillard) Schiller, which is known to cause harmful blooms in many estuarine and coastal environments. In contrast to many previous techniques, a simple imaging system was used to obtain phytoplankton images in this study. A new technique, combining phase congruency-based detection of circular objects, stochastic optimization, and image segmentation was developed for solving the task. Experimental tests have shown that robust object detection is possible even in images, where image area occupied by objects was much larger than 3%. On average 93.2% of objects representing P. minimum cells were detected. In future work, algorithms for automatic classification of objects into classes representing various phytoplankton species will be developed.

Acknowledgement

This research was funded by a Grant (No. LEK-05/2010) from the Research Council of Lithuania. The authors are grateful to Dr. Adrianna Ianora and Dr. Francesco Esposito (Stazione Zoologica, Naples, Italy) for providing the living culture of P. minimum.

References


