Understanding what we cannot see: automatic analysis of 4D digital in-line holographic microscopy data

Laura Leal-Taixé[1,*], Matthias Heydt[2], Axel Rosenhahn[2,3], and Bodo Rosenhahn[1]

[1] Leibniz Universität Hannover, Appelstr. 9A, Hannover, Germany
[2] Applied Physical Chemistry, University of Heidelberg, INF 253, Heidelberg, Germany
[3] Institute of Functional Interfaces, Karlsruhe Institute of Technology, Campus Nord, 76344 Eggenstein-Leopoldshafen, Germany

Abstract. Digital in-line holography is a microscopy technique which got an increasing attention over the last few years in the fields of microbiology, medicine and physics, as it provides an efficient way of measuring 3D microscopic data over time. In this paper, we present a complete system for the automatic analysis of digital in-line holographic data; we detect the 3D positions of the microorganisms, compute their trajectories over time and finally classify these trajectories according to their motion patterns. Tracking is performed using a robust method which evolves from the Hungarian bipartite weighted graph matching algorithm and allows us to deal with newly entering and leaving particles and compensate for missing data and outliers. In order to fully understand the behavior of the microorganisms, we make use of Hidden Markov Models (HMMs) to classify four different motion patterns of a microorganism and to separate multiple patterns occurring within a trajectory. We present a complete set of experiments which show that our tracking method has an accuracy between 76% and 91%, compared to ground truth data. The obtained classification rates on four full sequences (2500 frames) range between 83.5% and 100%.

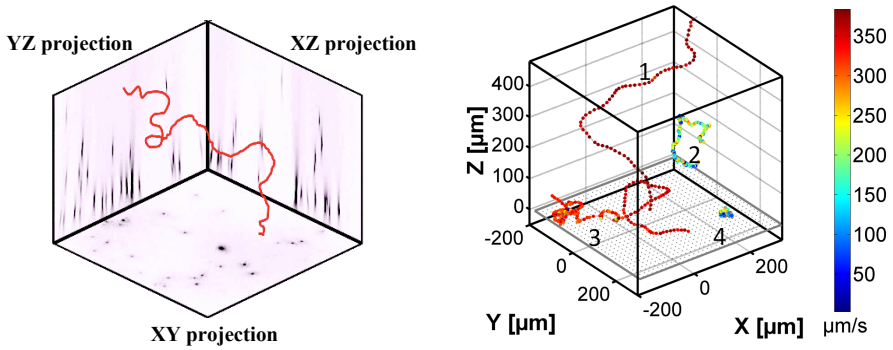
Keywords: digital in-line holographic microscopy, particle tracking, graph matching, multi-level Hungarian, Hidden Markov Models, motion pattern classification

1 Introduction

Many fields of interest in biology and other scientific research areas deal with intrinsically three-dimensional problems. The motility of swimming microorganisms such as bacteria or algae is of fundamental importance for topics like pathogen-host interactions [1], predator-prey interactions [1], biofilm-formation [2], or biofouling by marine microorganisms [3, 4].

Understanding the motility and behavioral patterns of microorganisms allows us to understand their interaction with the environment and thus to control environmental parameters to avoid unwanted consequences such as infections or biofouling. To study these effects in 3D several attempts have been made: tracking light microscopy, capable of tracking one bacterium at a time [5], stereoscopy [6] or confocal microscopy [7].

* Corresponding author: L. Leal-Taixé, email: leal@tnt.uni-hannover.de



(a) (b)

Fig. 1: (a) The input data, the projections obtained with digital in-line holography (inverted colors for better visualization). Sample trajectory in red. (b) The output data we want to obtain from each volume, the classification into four motion patterns, colored according to speed: orientation (1), wobbling (2), gyration (3) and intensive surface probing (4)

Berg built a pioneering tracking light microscope, capable of tracking one bacterium at a time in 3D. This has been used to investigate bacteria like Escherichia Coli [5]. Another way of measuring 3D trajectories is stereoscopy, which requires two synchronized cameras [6]. Confocal microscopy has also been used to study the motion of particles in colloidal systems over time, however the nature of this scanning technique limits the obtainable frame rate [7].

For any of these techniques, in order to draw statistically relevant conclusions, thousands of images have to be analyzed. Nowadays, this analysis is still heavily dependent on manual intervention. Recent work [8] presents a complete vision system for 2D cell tracking, which proves the increasing demand for efficient computer vision approaches in the field of microscopy as an emerging discipline. The search for a nearly automatic analysis of biological images has been extensively studied [9] but most of the work focuses on position as well as on the shape of the particle [10]. Several methods exist for multiple object detection based on methods such as Markov Chain Monte Carlo (MCMC) [11], inference in Bayesian networks [12] or the Nash Equilibrium of game theory [13]. These have been proven useful to track a fairly small number of targets, but are less appropriate when the number of targets is very large, as in our case. Statistical methods like Kalman filters [8], particle filters or recursive Bayesian filters [14] are widely used for tracking but they need a dynamical model of the target, a task that can be challenging depending on the microorganism under study and to which we dedicate the second part of this paper. In contrast to [8, 14], we do not use the output predictions of the filters to deal with occlusions, but rather use past and future information to complete broken trajectories and detect false alarms. Therefore, we do not need an extra track linking step as in [8]. Furthermore, we deal with 3D trajectories and random

and fast motions which are unsuited for a prediction type of approach. In this work we propose a global optimal matching solution and not a local one as suggested in [15].

Besides generating motion trajectories from microscopic data, a subsequent classification allows the biologists to get in a compact and compressed fashion the desired information from the large image sets. Indeed, the classification of motion patterns in biology is a well-studied topic [16] but identifying these patterns manually is a complicated and time consuming task. Recently, machine learning and pattern recognition techniques have been introduced to analyze in detail such complex movements. These techniques include: Principal Component Analysis (PCA) [17], a linear transformation used to analyze high dimensional data; Bayesian models [18] which use a graph model and the rules of probability theory to select among different hypotheses; or Support Vector Machines (SVM) [19], which use training data to find the optimum parameters of the model representing each class. A comparison of machine learning approaches applied to biology can be found in [20]. In order to classify biological patterns, we need to use an approach able to handle time-varying signals. Hidden Markov Models [21] are statistical models especially known for their application in temporal pattern recognition. They were first used in speech recognition and since then, HMMs have been extensively applied to vision. Applications vary from handwritten word recognition [22], face recognition [23] or human action recognition [24, 25].

In this paper, we present a complete system for the automatic analysis of digital in-line holographic data. This microscopy technique provides videos of a 3D volume and is used to study complex movements of microorganisms. The huge amount of information that we can extract from holographic images makes it necessary to have an automatic method to analyze this complex 4D data. Our system performs the detection of the 3D positions, tracking of the complete trajectories and classification of motion patterns. For multiple microorganism tracking, we propose a geometrically motivated and globally optimal multi-level Hungarian to compensate for leaving and entering particles, recover from missing data and erase the outliers to reconstruct the whole trajectory of the microorganisms [26] . Afterwards, we focus on the classification of four motion patterns of the green alga *Ulva linza* with the use of Hidden Markov Models [27]. Furthermore, our system is able to find and separate different patterns within a single sequence. Besides classification of motion patterns, a key issue is the choice of features used to classify and distinguish the involved patterns. For this reason we perfom an extensive analysis of the importance of typical motion parameters, such as velocity, curvature, orientation, etc. Our developed system is highly flexible and can easily be extended. Especially for forthcoming work on cells, microorganisms or human behavior, such automated algorithms are of pivotal importance as they allow high throughput analysis of individual segments in motion data.

2 Detection of 3D positions

In this section we present the details of digital in-line holography, how this microscopy technique allows us to obtain 3D positions of the microorganisms as well as the image processing methods used to robustly extract these positions from the images.

2.1 Digital in-line holographic microscopy (DIHM)

Digital in-line holographic microscopy provides an alternative, lensless microscopy technique which intrinsically contains three dimensional information about the investigated volume. It does not require a feedback control which responds to the motion and it uses only one CCD chip. This makes the method very straightforward and can be implemented with a very simple setup as shown in Figure 2.

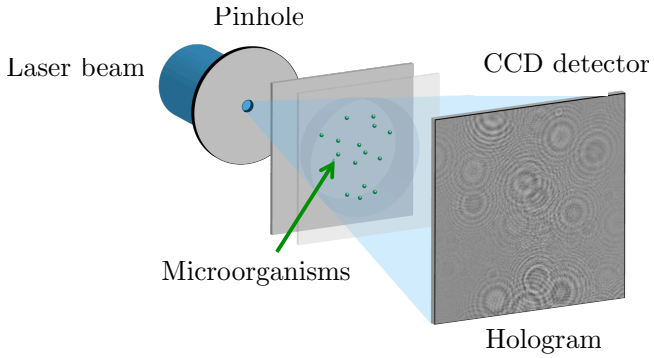


Fig. 2: Schematic setup for a digital in-line holographic experiment consisting of the laser, a spatial filter to create the divergent light cone, the objects of interest (e.g. microorganisms) and a detector which records the hologram

The holographic microscope requires only a divergent wavefront which is produced by diffraction of laser light from a pinhole. A CCD chip finally captures the hologram. The holographic microscope setup follows directly Gabors initial idea [28] and has been implemented for laser radiation by Xu et al. [29]. A hologram recorded without the presence of particles, called the *source* is subtracted from each hologram. This is used to reduce the constant illumination background and other artifacts; there are filtering methods [30, 31] to achieve this in case a *source* image is not readily available. These resulting holograms can then be reconstructed back into real space by a Kirchhoff-Helmholtz transformation [29] shown in Equation (1).

$$K(\mathbf{r}) = \int_S d^2\xi I(\xi) e^{\frac{ik\mathbf{r}\cdot\xi}{|\xi|}} \tag{1}$$

The integration extends over the 2D surface of the screen with coordinates $\xi = (X, Y, L)$, where L is the distance from the source (pinhole) to the center of the detector (CCD chip), $I(\xi)$ is the contrast image (hologram) on the screen obtained by subtracting the images with and without the object present and k the wave number: $k = 2\pi/\lambda$.

As we can see in Figure 3, the idea behind the reconstruction is to obtain a series of stacked XY projections from the hologram image. These projections contain the information at different depth values. From these images, we can obtain the 3 final projections XY, XZ and YZ, as described in [32]. These projections contain the image

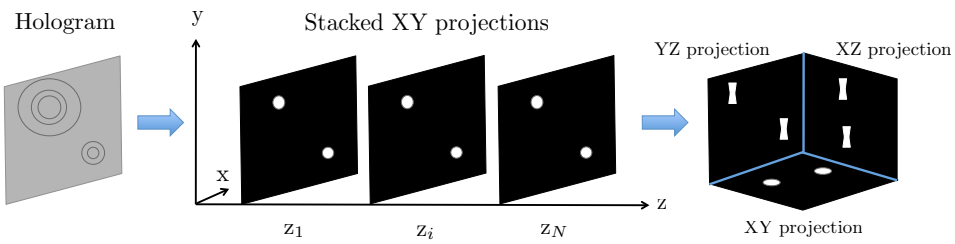


Fig. 3: Illustration of the reconstruction process. From the hologram a stack of XY projections is obtained in several depths and from those, the final 3 projections (XZ, XZ and YZ) are obtained.

information of the complete observation volume, i.e. from every object located in the light cone between pinhole and detector. The resolution in X and Y is $\delta_{x,y} = \frac{\lambda}{NA}$, where NA stands for the numerical aperture given by $NA = \frac{D}{2L}$, where D is the detector's side length. The resolution in the Z direction, which is the direction of the laser, is worse, $\delta_z = \frac{\lambda}{2NA^2}$. This is because the third dimension, Z, is obtained with a mathematical reconstruction, unlike confocal microscopy, where the value of every voxel is returned. On the other hand, confocal microscope takes a long time to return the values of all the voxels in a volume, and therefore is unsuited for tracking at a high frame rate.

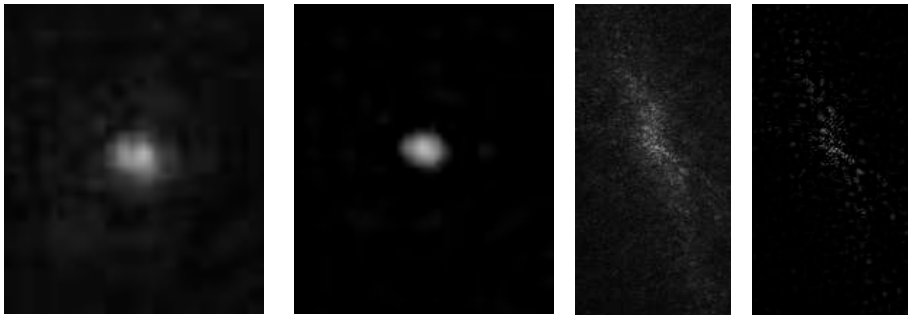
Using video sequences of holograms, it is possible to track multiple objects in 3D over time at a high frame rate, and multiple spores present in a single frame can be tracked simultaneously [3, 15, 33]. Using this advantage of digital in-line holographic microscopy a number of 3D phenomena in microbiology have been investigated: Lewis et al. [34] examined the swimming speed of Alexandrium (Dinophyceae), Sheng et al. [35, 36] studied the swimming behavior of predatory dinoflagellates in the presence of prey, and Sun et al. [37] used a submersible device to investigate in situ plankton in the ocean.

2.2 Detection of the microorganisms

In our sequences we are observing the green algae *Ulva linza*, which has a spherical spore body and four flagella. Since the body scatters most of the light, in the projected images the particles have a circular shape. In order to preserve and enhance the particle shape (see Figure 4(a)) but reduce noise and illumination irregularities of the image (see Figure 4(b)), we apply the Laplacian of Gaussian filter (LoG) which, for its shape, is a blob detector [38]:

$$LoG(x,y) = \frac{-1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

Due to the divergent nature of the light cone, the particles can appear smaller or larger in the projections depending on the z-plane. Therefore, the LoG filter is applied in several scales [38] according to the magnification. Note that the whole algorithm



(a) (b)

Fig. 4: (a) Enhancement of the shape of the microorganisms. (b) Reduction of the noise.

is extremely adaptable, since we can detect particles with any shape by just changing the filter. After this, we use thresholding on each projection to have the position of the candidate particles on the image. The final 3D positions (Figure 6, green box labeled "Candidate particles") are determined by thresholding each projection XY, XZ and YZ to find the particles in each image and crossing the information of the three projections.

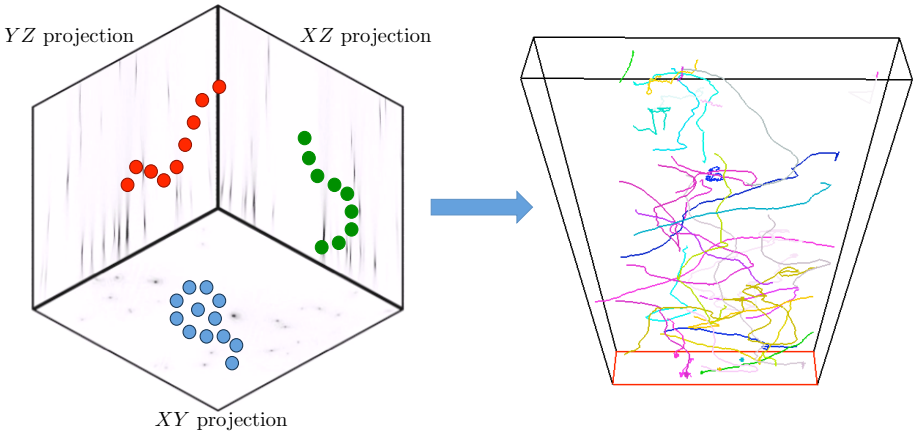


Fig. 5: From the 3D positions obtained at each time frame, we use the method in Section 3 to obtain the full trajectory of each microorganism

Once we have computed the 3D positions of all the microorganisms in all frames, we are interested in linking these 3D positions in order to find their complete 3D trajectories over time, a problem that is generally called Multiple object tracking (see Figure 5).

3 Automatic extraction of 3D trajectories

In this section we present the complete method to estimate the 3D trajectories of the microorganisms over time. Our algorithm, the Multi-level Hungarian, is a robust method evolved from the Hungarian-Munkre's assignment method, and is capable of dealing with entering and leaving particles, missing data and outliers. The diagram of the method is presented in Figure 6.

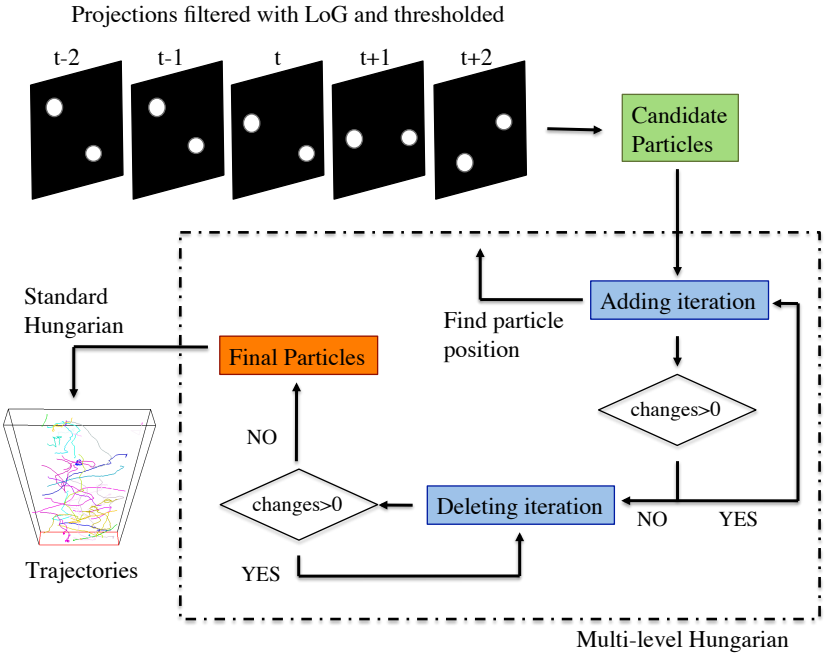


Fig. 6: Diagram of the algorithm described in Section 3.2

3.1 Cost function and bipartite graph matching

Graph Matching is one of the fundamental problems in Graph Theory and it can be defined as: given a graph $G = (V, E)$, where E represents its set of edges and V its set of nodes, a matching M in G is a set of pairwise non-adjacent edges, which means that no edges share a common vertex. For our application, we are specially interested in the *Assignment Problem*, which consists in finding a maximum weight matching in a weighted bipartite graph. In a general form, the problem can be expressed as: "There are N jobs and N workers. Any worker can be assigned to any job, incurring some cost that varies depending on the job-worker assignment. All jobs must be performed by assigning exactly one worker to each job in such a way that the total cost is minimized

(or maximized)". For the subsets of vertices X and Y, we build a cost matrix in which the element $C(i, j)$ will represent the weight or cost related to vertex i in X and vertex j in Y.

For numerical optimization we use the Hungarian or Munkres' assignment algorithm, a combinatorial optimization algorithm [39, 40] that solves the bipartite graph matching problem in polynomial time. For implementation details on the Hungarian we recommend [41]. Our initial problem configuration is: there are M particles in frame t_1 and N particles in frame t_2. The Hungarian will help us to find which particle in t_1 corresponds to which particle in t_2, allowing us to reconstruct their full trajectories in 3D space. Nonetheless, the Hungarian algorithm has some disadvantages which we should be aware of. In the context of our project, we summarize in Table 1 some of the advantages and disadvantages of the Hungarian algorithm.

ADVANTAGES
Finds a global solution for all vertices
Cost matrix is versatile
Easy to solve, bipartite matching is the simplest of all graph problems
DISADVANTAGES
Cannot handle missing vertices (a)
Cannot handle entering or leaving particles (b)
No discrimination of matches even if the cost is very high (c)

Table 1: Summary of the advantages and disadvantages of the Hungarian algorithm

In the following sections, we present how to solve the three disadvantages: (a) is solved with the multi-level Hungarian method explained in Section 3.2, (b) is solved with the IN/OUT states of Section 3.1 and finally a solution for (c) is presented in Section 3.1 as a maximum cost restriction.

The cost function C as key input for the Hungarian algorithm is created using the Euclidean distances between particles, that is, element $C(i, j)$ of the matrix represents the distance between particle i of frame t_1 and particle j of frame t_2. With this matrix, we need to solve a *minimum assignment problem* since we are interested in matching those particles which are close to each other.

Note that it is also possible to include in the cost function other characteristics of the particle like speed, size or gray level distribution. Such parameters can act as additional regularizers during trajectory estimation.

IN and OUT states In order to include more knowledge about the environment to the Hungarian algorithm and avoid matches with very high costs, we have created a variation for the cost matrix. In our experiments, particles can only enter and leave the scene by crossing the borders of the Field Of View (FOV) of the holographic microscope, therefore, the creation and deletion of particles depends on their distance to the borders

of the FOV. Nonetheless, the method can be easily extended to situations where trajectories are created (for example by cell division) or terminated (when the predator eats the prey) away from the FOV borders.

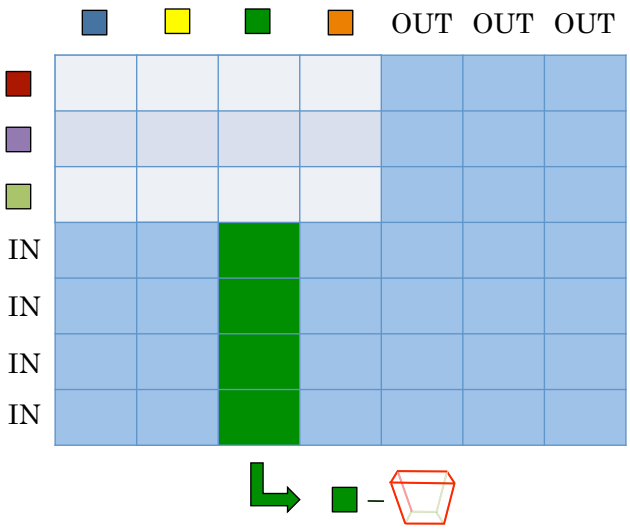


Fig. 7: Change in the cost matrix to include the IN/OUT states. Each particle is represented by a different color. The value of each extra element added is the distance between the particle position and the closest volume boundary.

As shown in Figure 7, we introduce the IN/OUT states in the cost matrix by adding extra row and columns. If we are matching the particles in frame f to particles in frame $f+1$, we will add as many columns as particles in frame f and as many rows as particles in frame $f+1$. This way, all the particles have the possibility to enter/leave the scene. Additionally, this allows us to obtain a square matrix, needed for the matching algorithm, even if the number of particles is not the same in consecutive frames.

$$C_{BB}(i,k) = \begin{cases} min(|P_i - \{M_x, m_x, M_y, m_y, M_z\}|), & 1 \leq i \leq M \text{ and } k > N \\ min(|P_k - \{M_x, m_x, M_y, m_y, M_z\}|) & 1 \leq k \leq N \text{ and } i > M \\ min(C_{BB}(i, 1:k-1), C_{BB}(1:i-1, k)) & i > N \text{ and } k > N \end{cases} \quad (3)$$

The cost of the added elements includes the information of the environment by calculating the distance of each particle to the nearest edge of the FOV as in Equation (3), where M is the number of particles in frame t_1 and N is the number of particles in frame t_2, m_x, m_y, m_z are the low borders and M_x, M_y, M_z are the high borders for each of the axis. Note that the low border in the z axis is not included as it represents the surface where the microorganisms might settle and, therefore, no particles can enter or leave from there.

If the distance is small enough, the Hungarian algorithm matches the particle with an IN/OUT state.

In Figure 8 we consider the simple scenario in which we have 4 particles in one frame and 4 in the next frame. As we can see, there is a particle which leaves the scene from the lower edge and a particle which enters the scene in the next frame from the right upper corner. As shown in Figure 8(a), the Hungarian algorithm finds a wrong matching since the result is completely altered by the entering/leaving particles. With the introduction of the IN/OUT state feature, the particles are now correctly matched (see Figure 8(b)) and the ones which enter/leave the scene are identified as independent particles.

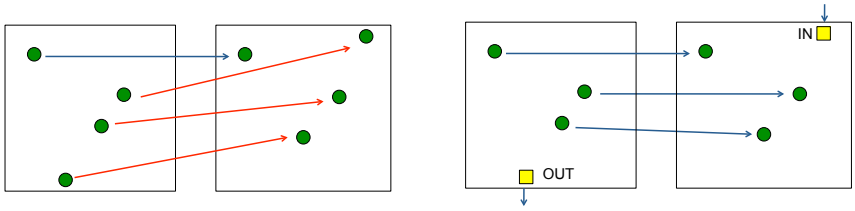


Fig. 8: Representation of the particles in frame t_1 (left) and t_2 (right). The lines represent the matchings. (a) Wrongly matched. (b) Correctly matched as a result of the IN/OUT state feature.

Maximum cost restriction Due to noise and illumination irregularities of the holograms, it is common that a particle is not detected in several frames, which means a particle can virtually disappear in the middle of the scene. If a particle is no longer detected, all the matches can be greatly affected. That is why we introduce a maximum cost restriction for the cost matrix, which does not allow matches which have costs higher than a given threshold V. This threshold is the observed maximum speed of the algae spores under study [32]. The restriction is guaranteed by using the same added elements as the ones used for the IN/OUT states, therefore, the final value of the added elements of the cost matrix will be:

$$C(i,k) = min(C_{BB}, V\Delta t) \tag{4}$$

Overall, if a particle is near a volume border or cannot be matched to another particle which is within a reachable distance, it will be matched to an IN/OUT state. This ensures that the resulting matches are all physically possible. Still, if we have missing data and a certain particle is matched to an IN/OUT state, we will recover two trajectories instead of the complete one. In the next section we present a hierarchical solution to recover missing data by extending the matching to the temporal dimension.

3.2 Multi-level Hungarian for missing data

If we consider just the particles detected using the thresholding, we see that there are many gaps within a trajectory (see Figure 12(a)). These gaps can be a result of morphing (different object orientations yield to different contrast), changes in the illumination, etc. The standard Hungarian is not capable of filling in the missing data and creating full trajectories, therefore, we now introduce a method based on the standard Hungarian that allows us to treat missing data, outliers and create complete trajectories. The general routine of the algorithm, the multi-level Hungarian, is:

- Find the matchings between particles in frames $[i-2 \dots i+2]$, so we know the position of each particle in each of these frames (if present). (Section 3.2).
- Build a table with all these positions and fill the gaps given some strict conditions. Let the algorithm converge until no particles are added. (Section 3.2).
- On the same table and given some conditions, erase the outliers. Let the algorithm converge until no particles are deleted. (Section 3.2).

The levels of the multi-level Hungarian The multi-level Hungarian takes advantage of the temporal information in 5 consecutive frames and is able to recover from occlusions and gaps in up to two consecutive frames. The standard Hungarian gives us the matching between the particles in frame t_1 and frame t_2 and we use this to find matchings of the same particle in 5 consecutive frames, $[i-2, \dots, i+2]$. In order to find these matchings, the Hungarian is applied on different levels. The first two levels, represented in Figure 9 by red arrows, are created to find the matching of the particles in the frame of study, frame i. But it can also be the case that a particle is not present in frame i but is present in the other frames. To solve all the possible combinations given this fact, we use Levels 3, 4 and 5, represented in Figure 9 by green arrows.

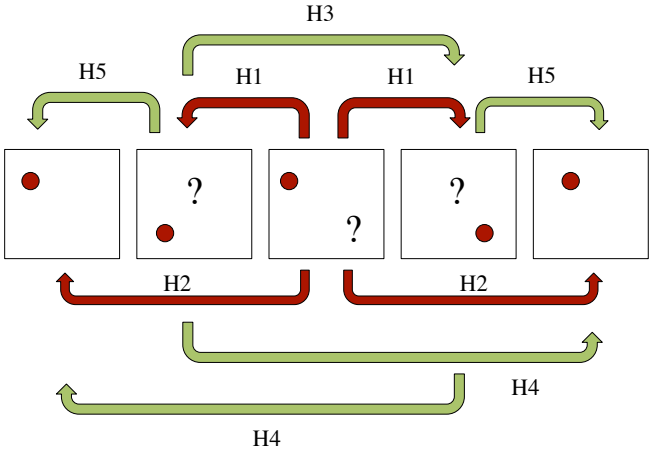


Fig. 9: Represented frames: [i-2,i-1,i,i+1,i+2]. Levels of the multi-level Hungarian.

Below we show a detailed description and purpose of each level of the multi-level Hungarian:

- Level 1: Matches particles in frame i with frames $i \pm 1$.
- Level 2: Matches particles in frame i with frames $i \pm 2$. With the first two levels, we know, for all the particles in frame i, their position in the neighboring frames (if they appear).
- Level 3: Matches particles in frame $i - 1$ with frame $i + 1$.
- Level 4: Matches particles in frame $i \pm 1$ with frame $i \mp 2$. Level 3 and 4 solve the detection of matchings when a particle appears in frames $i \pm 1$ and might appear in $i \pm 2$, but is not present in frame i.
- Level 5: Matches particles in frame $i \pm 1$ with frame $i \pm 2$.

Conditions to add/delete particles Once all the levels are applied hierarchically, a table with the matching information is created. On one axis we have the number of particles and on the other the 5 frames from $[i - 2 \dots i + 2]$, as shown in Figure 10.

To change the table information, we use two iterations: the adding iteration and the deleting iteration which appear in Figure 6 as blue boxes. During the *adding iteration*, we look for empty cells in the table where there is likely to be a particle. A new particle position is added if, and only if, two conditions are met:

1. There are at least 3 particles present in the row. Particles have continuity while noise points do not.
2. It is not the first or last particle of the row. We use this strict condition to avoid the creation of false particle positions or the incorrect elongation of trajectories.

If we look at particle 6 of the table in Figure 10. In this case, we do not want to add any particle in frames $i - 2$ and $i - 1$, since the trajectory could be starting at frame i. In the case of particle 4, we do not want to add a particle in frame $i + 2$ because the trajectory could be ending at $i + 1$. Each iteration repeats this process for all frames, and we iterate until the number of particles added converges.

After convergence, the *deleting iteration* starts and we erase the outliers considered as noise. A new particle position is deleted if, and only if, two conditions are met:

1. The particle is present in the frame of study i.
2. There are less than 3 particles in the same row.

We only erase particles from the frame of study i because it can be the case that a particle appears blurry in the first frames but is later correctly detected and has more continuity. Therefore, we only delete particles from which we know the complete neighborhood. Each iteration repeats this process for all frames, and we iterate until the number of particles deleted converges. The resulting particles are shown in Figure 10.

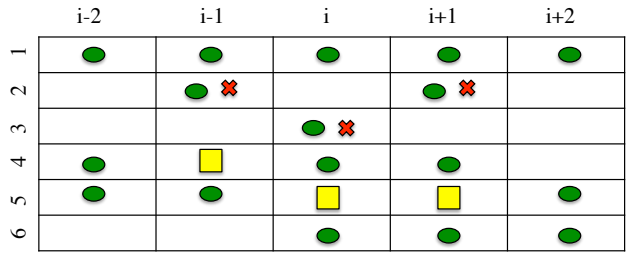


Fig. 10: Table with: the initial particles detected by the multi-level Hungarian (green ellipses), the ones added in the adding iteration (yellow squares) and the ones deleted in the deleting iteration (red crosses). In the blank spaces no position has been added or deleted.

Missing data interpolation During the adding iteration, we use the information of the filtered projection in order to find the correct position of the new particle (Figure 6) . For example, if we want to add a particle in frame $i-1$, we go to the filtered projections XY, XZ, YZ in $t = i-1$, take the position of the corresponding particle in $t = i$ or $t = i-2$ and search for the maximum value within a window w. If the position found is already present in the candidate particles' list of that frame, we go back to the projection and determine the position of the second maximum value. This allows us to distinguish two particles which are close to each other.

There are many studies on how to improve the particle depth-position resolution (z-position). As in [42] we use the traditional method of considering the maximum value of the particle as its center. Other more complex methods [31] have been developed which also deal with different particle sizes, but the flexibility of using morphological filtering already allows us to easily adapt our algorithm.

3.3 The final Hungarian

Once the final particle positions are obtained (in Figure 6, orange box labeled "Final particles"), we perform one last step to determine the trajectories. We use the standard Hungarian to match particles from frame i to frame $i+1$.

4 Motion pattern classification

In this section we describe the different types of motion patterns, as well as the design of the complete HMM and the features used for their classification.

4.1 Hidden Markov Models

Hidden Markov Models [21] are statistical models of sequencial data widely used in many applications in artificial intelligence, speech and pattern recognition and modeling of biological processes.

In an HMM it is assumed that the system being modeled is a Markov process with unobserved states. This hidden stochastic process can only be observed through another set of stochastic processes that produce the sequence of symbols $O = o_1, o_2, ..., o_M$. An HMM consists of a number N of states $S_1, S_2, ..., S_N$. The system is at one of the states at any given time. Every HMM can be defined by the triple $\lambda = (\Pi, A, B)$. $\Pi = \{\pi_i\}$ is the vector of initial state probabilities. Each transition from S_i to S_j can occur with a probability of a_{ij}, where $\sum_j a_{ij} = 1$. $A = \{a_{ij}\}$ is the state transition matrix. In addition, each state S_i generates an output o_k with a probability distribution $b_{ik} = P(o_k|S_i)$. $B = \{b_{ik}\}$ is the emission matrix.

There are three main problems related to HMMs:

1. *The evaluation problem*: for a sequence of observations O compute the probability $P(O|\lambda)$ that an HMM λ generated O. This is solved using the Forward-Backward algorithm.
2. *The estimation problem*: given O and an HMM λ, recover the most likely state sequence $S_1, S_2, ..., S_N$ that generated O. Problem 2 is solved by the Viterbi algorithm, a dynamic programming algorithm that computes the most likely sequence of hidden states in $O(N^2T)$ time.
3. *The optimization problem*: find the parameters of the HMM λ which maximize $P(O|\lambda)$ for some output sequence O. A local maximum likelihood can be derived efficiently using the Baum-Welch algorithm.

For a more detailed introduction to HMM theory, we refer to [21].

4.2 Types of patterns

In our experimental setup we are interested in four patterns shown by the green alga *Ulva linza* as depicted in Figure 1(b): Orientation(1), Wobbling(2), Gyration(3) and intensive surface probing or Spinning(4). These characteristic swimming patterns are highly similar to the patterns observed before in [43] for the brown algae *Hincksia irregularis*.

Orientation Trajectory 1 in Figure 1(b) is an example of the Orientation pattern. This pattern typically occurs in solution and far away from surfaces. The most important characteristics of the pattern are the high swimming speed (a mean of $150\mu m/s$) and a straight swimming motion with moderate turning angles.

Wobbling Pattern 2 is called the Wobbling pattern and its main characteristic is a much slower mean velocity of around $50\mu m/s$. The spores assigned to the pattern often change their direction of movement and only swim in straight lines for very short distances. Compared to the orientation pattern this leads to less smooth trajectories.

Gyration Trajectory 3 is an example of the Gyration pattern. This pattern is extremely important for the exploration of surfaces as occasional surface contacts are observable. The behavior in solution is similar to the Orientation pattern. Since in this pattern spores

often switch between swimming towards and away from the surfaces, it can be interpreted as a pre-stage to surface probing.

Intensive surface probing and Spinning Pattern 4 involves swimming in circles close to the surface within a very limited region. After a certain exploration time, the spores can either permanently attach or leave the surface to the next position and start swimming in circular patterns again. This motion is characterized by decreased mean velocities of about $30\mu m/s$ in combination with a higher tendency to change direction (see Figure 1(b), case 4).

4.3 Features used for classification

An analysis of the features used for classification is presented in this section. Most of the features are generally used in motion analysis problems. An intrinsic characteristic of digital in-line holographic microscopy is the lower resolution of the Z position compared to the X,Y resolution [31]. Since many of the following features depend on the depth value, we compute the average measurements within 5 frames in order to reduce the noise of such features. The four characteristic features used are:

- v, *velocity*: the speed of the particles is an important descriptive feature as we can see in Figure 1(b). We use only the magnitude of the speed vector, since the direction is described by the next two parameters. Range is $[0, maxSpeed]$. $maxSpeed$ is the maximum speed of the particles as found experimentally in [32].
- α, *angle between velocities*: it measures the change in direction, distinguishing stable patterns from random ones. Range is $[0, 180]$.
- β, *angle to normal of the surface*: it measures how the particles approaches the surface or how it swims above it. Range is $[0, 180]$.
- D, *distance to surface*: this can be a key feature to differentiate surface-induced movements from general movements. Range is $(m_z, M_z]$, where m_z and M_z are the z limits of the volume under study.

In order to work with Hidden Markov Models, we need to represent the features for each pattern with a fixed set of symbols. The number of total symbols will depend on the number of symbols used to represent each feature $N_{symbols} = N_v N_\alpha N_\beta N_D$.

In order to convert every symbol for each feature into a unique symbol for the HMM, we use Equation (5), where J is the final symbol we are looking for, $J_{1..4}$ are the symbols for each of the features, ranged $[1..N_{J_{1..4}}]$, where $N_{J_{1..4}}$ are the number of symbols per feature.

$$J = J_1 + (J_2 - 1)N_{J_1} + (J_3 - 1)N_{J_1}N_{J_2} + (J_4 - 1)N_{J_1}N_{J_2}N_{J_3} \tag{5}$$

In the next sections we present how to use the resulting symbols to train the HMMs. The symbols are the observations of the HMM, therefore, the training process gives us the probability of emitting each symbol for each of the states.

4.4 Building and training the HMMs

In speech recognition, an HMM is trained for each of the phonemes of a language. Later, words are constructed by concatenating several HMMs of the phonemes that form the word. HMMs for sentences can even be created by concatenating HMMs of words, etc. We take a similar hierarchical approach in this paper. We train one HMM for each of the patterns and then we combine them into a unique Markov chain with a simple yet effective design that will be able to describe any pattern or combination of patterns. This approach can be used in any problem where multiple motion patterns are present.

Individual HMM per pattern In order to represent each pattern, we build a Markov chain with N states and we only allow the model to stay in the same state or move one state forward. Finally, from state N we can also go back to state 1. The number of states N is found empirically using the training data (we use $N = 4$ for all the experiments, see Section 5.4). The HMM is trained using the Baum-Welch algorithm to obtain the transition and emission matrices.

Complete HMM The idea of having a complete HMM that represents all the patterns is that we can not only classify sequences where there is one pattern present, but sequences where the particle makes transitions between different patterns. In Figure 11(a) we can

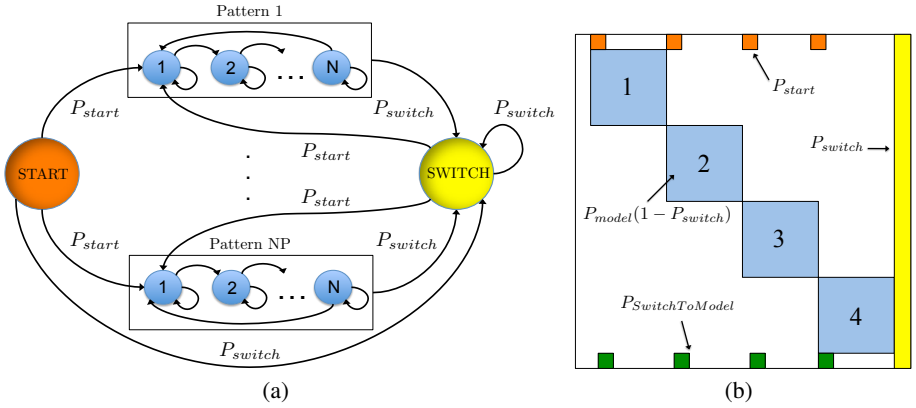


Fig. 11: (a) Complete HMM created to include changes between patterns within one trajectory. (b) Transition matrix of the complete HMM

see a representation of the complete model while the design of the transition matrix is depicted in Figure 11(b). The four individual HMMs for each of the patterns are placed in parallel (blue). In order to deal with the transitions we create two special states: the START and the SWITCH state.

The START state is just created to allow the system to begin at any pattern (orange). We define $P_{start} = P_{SwitchToModel} = \frac{1-P_{switch}}{N_P}$ where N_P is the number of patterns. As START does not contain any information of the pattern, it does not emit any symbol.

The purpose of the new state SWITCH is to make transitions easier. Imagine a given trajectory which makes a transition from Pattern 1 to Pattern 2. While transitioning, the features create a symbol that neither belongs to Pattern 1 nor 2. The system can then go to state SWITCH to emit that symbol and continue to Pattern 2. Therefore, all SWITCH emission probabilities are $\frac{1}{N_{symbols}}$. Since SWITCH is such a convenient state, we need to impose restrictive conditions so that the system does not go or stay in SWITCH too often. This is controlled by the parameter P_{switch}, set at the minimum value of all the P_{model} minus a small ϵ. This way, we ensure that P_{switch} is the lowest transition probability in the system.

Finally, the sequence of states given by the Viterbi algorithm determines the motion pattern observed. Our implementation uses the standard MatLab HMM functions.

5 Experimental results

In order to test our algorithm we use 6 sequences (labeled S1 to S6) in which the swimming motion of *Ulva linza* spores is observed [3]. All the sequences have some particle positions which have been semi-automatically reconstructed and manually labeled and inspected (our ground truth) for later comparison with our fully-automatic results.

5.1 Performance of the standard Hungarian

First of all, we want to show the performance of the final standard Hungarian described in Section 3.3. For this, we use the ground truth particle positions and apply the Hungarian algorithm to determine the complete trajectories of the microorganisms. Comparing the automatic matches to the ground truth, we can see that in 67% of all the sequences, the total number of particles in the sequence is correctly detected, while in the remaining 33%, there is just a 5% difference in the number of particles. The average accuracy of the matchings reaches 96.61%.

To further test the robustness of the Hungarian algorithm, we add random noise to each position of our particles. The added noise is in the same order as the noise intrinsically present in the reconstructed images, determined experimentally in [32]. $N = 100$ experiments are performed on each of the sequences and the accuracy is recorded. Results show that the average accuracy of the matching is just reduced from 96.61% to 93.51%, making the Hungarian algorithm very robust to the noise present in the holographic images and therefore well suited to find the trajectories of the particles.

5.2 Performance of the multi-level Hungarian

To test the performance of the multi-level Hungarian we apply the method to three sets of particles:

- Set A: particles determined by the threshold (pre multi-level Hungarian)
- Set B: particles corrected after multi-level Hungarian
- Set C: ground truth particles, containing all the manually labeled particles

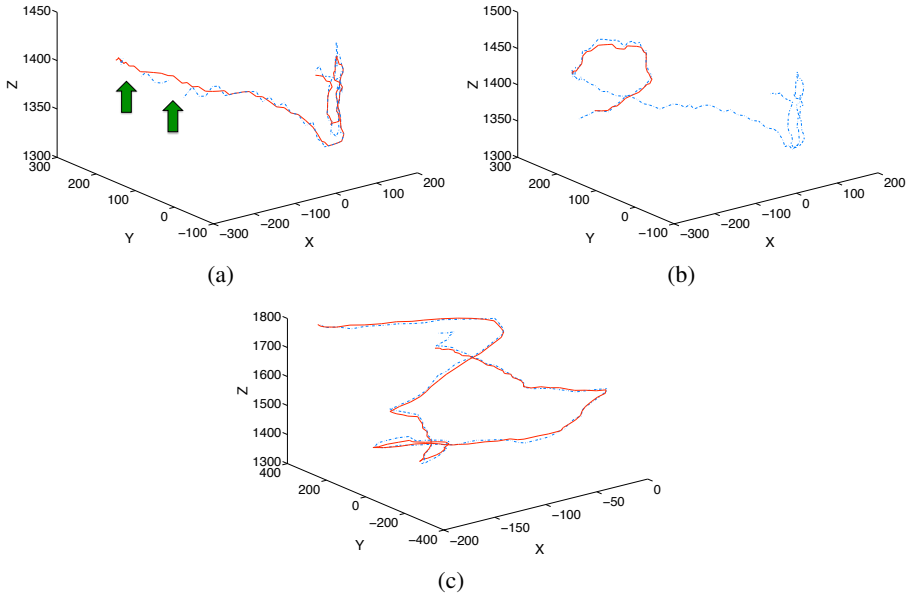


Fig. 12: (a) 3 separate trajectories are detected with the standard Hungarian (blue dashed line). Merged trajectory detected with our method (with a smoothing term, red line). Missing data spots marked by arrows. (b), (c) Ground truth trajectories (blue dashed line). Trajectories automatically detected with our method (red line).

	S1	**S2**	**S3**	**S4**	**S5**	**S6**
Set A	1599	1110	579	668	1148	2336
Set B	236	163	130	142	189	830
Set C	40	143	44	54	49	48

Table 2: Comparison of the number of particles detected by thresholding, by the multi-level Hungarian and the ground truth.

We then start by comparing the number of particles detected, as shown in Table 2.

As shown in Table 2, the number of particles detected in Set A is drastically reduced in Set B, after applying the multi-level Hungarian, demonstrating its abilities to compensate for missing data and merging trajectories. If we compare it to Set C, we see that the number is still too high, indicating possible tracks which were not merged and so detected as independent.

Nonetheless, as we do not know exactly the amount of particles present in a volume (not all particle positions have been labeled), it is of great value for us to compare the average length of the trajectories, defined as the number of frames in which the same particle is present. The results are shown in Table 3 where we can clearly see that the average length of a trajectory is greatly improved with the multi-level Hungarian,

which is crucial since long trajectories give us more information on the behavior of the particles.

	S1	S2	S3	S4	S5	S6
Set A	3	5	5	4	6	7
Set B	19	31	27	23	38	23
Set C	58	54	54	70	126	105

Table 3: Comparison of the trajectories' average length

Now let us consider just useful trajectories for particle analysis, that is, trajectories with a length of more than 25 frames, which are the trajectories that will be useful later for motion pattern classification. Tracking with the standard Hungarian returns 20.7% of useful trajectories from a volume, while the multi-level Hungarian allows us to extract 30.1%. In the end, this means that we can obtain more useful information from each analyzed volume.

Ultimately, this means that fewer volumes have to be analyzed in order to have enough information to draw conclusions of the behavior of a microorganism.

5.3 Performance of the complete algorithm

Finally, we are interested in determining the performance of the complete algorithm, including detection and tracking. For this comparison, we are going to present two values:

- Missing: percentage of ground truth particles which are not present in the automatic determination
- Extra: percentage of automatic particles that do not appear in the ground truth data

In Table 4 we show the detailed results for each surface.

	S1	S2	S3	S4	S5	S6
Missing (%)	8.9	20.7	19.1	23.6	11.5	12.9
Extra (%)	54.9	34.1	46.5	13.3	25.8	74.6

Table 4: Missing labeled and extra automatic particles

Our automatic algorithm detects between 76% and 91% of the particles present in the volume. This gives us a measure of how reliable our method is, since it is able to detect most of our verified particle positions. Putting this information together with the percentage of particles detected by our algorithm but not labeled, we can see that our method extracts much more information from the volume of study. This is clear in the

case of S6, where we have a volume with many crossing particles which are difficult to label manually and where our algorithm gives us almost 75% more information.

We now consider the actual trajectories and particle position and measure the position error of our method. The error is measured as the Euclidean distance between each point of the ground truth and the automatic trajectories, both at time t. In Figure 12(a) we can see the 3 independent trajectories found with the standard Hungarian and the final merged trajectory which proves the power of our algorithm to fill in the gaps (pointed by arrows). In Figure 12(b) we can see that the automatic trajectory is much shorter (there is a length difference of 105 frames), although the common part is very similar with an error of just 4,2μm. Figure 12(c) on the other hand, shows a perfectly matched trajectory with a length difference of 8 frames and error of 6,4μm for the whole trajectory, which is around twice the diameter of the spore body. This proves that the determination of the particle position is accurate but the merging of trajectories can be improved.

The next sections are dedicated to several experimental results on the automatic classification of biological motion patterns. All the trajectories used from now on are obtained automatically with the method described in Section 3 and are classified manually by experts, which we refer to as our ground truth classification data.

5.4 Evaluation of the features used for classification

The experiments in this section have the purpose of determining the impact of each feature for the correct classification of each pattern. We perform leave-one-out tests on our training data which consists of 525 trajectories: 78 for wobbling, 181 for gyration, 202 for orientation and 64 for intensive surface probing.

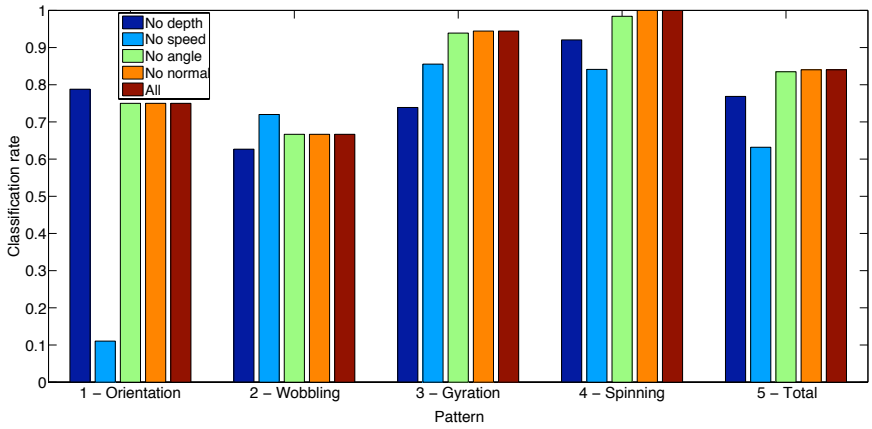


Fig. 13: Classification rate for parameters $N = 4$, $N_v = 4$, $N_\alpha = 3$, $N_\beta = 3$ and $N_D = 3$. On each experiment, one of the features is not used. In the last experiment all features are used.

The first experiment that we conduct (see Figure 13) is to determine the effect of each parameter for the classification of all the patterns. The number of symbols and states can only be determined empirically since they depend heavily on the amount of training data. In our experiments, we found the best set of parameters to be $N = 4$, $N_v = 4$, $N_\alpha = 3$, $N_\beta = 3$ and $N_D = 3$, for which we obtain a classification rate of 83.86%.

For each test, we set one parameter to 1, which means that the corresponding feature has no effect in the classification process. For example, the first bar in blue labeled "No Depth" is done with $N_D = 1$. The classification rate for each pattern (labeled from 1 to 4) as well as the mean for all the patterns (labeled Total) is recorded.

As we can see, the angle α and the normal β information are the less relevant features, since the classification rate with and without these features is almost the same. The angle information depends on the z component and, as explained in section 4.3, the lower resolution in z can result in noisy measurements. In this case, the trade-off is between having a noisy angle data which can be unreliable, or an average measure which is less discriminative for classification. The most distinguishing feature according to Figure 13 is the speed. Without it, the total classification rate decreases to 55.51% and down to just 11.05% for the orientation pattern.

Based on the previous results, we could think of just using the depth and speed information for classification. But if $N_\alpha = N_\beta = 1$, the rate goes down to 79.69%. That means that we need one of the two measures for correct classification. The parameters used are: $N = 4$, $N_v = 4$, $N_\alpha = 1$, $N_\beta = 3$ and $N_D = 3$, for which we obtain a classification rate of 83.5%. This rate is very close to the result with $N_\alpha = 3$, with the advantage that we now use less symbols to represent the same information. Several tests lead us to choose $N = 4$ number of states.

	1 - Ori	2 - Wob	3 - Gyr	4 - Spin
1 - Ori	0.75	0.09	0.16	
2 - Wob	0.07	0.68	0.01	0.24
3 - Gyr	0.01		0.94	0.05
4 - Spin			0.02	0.98

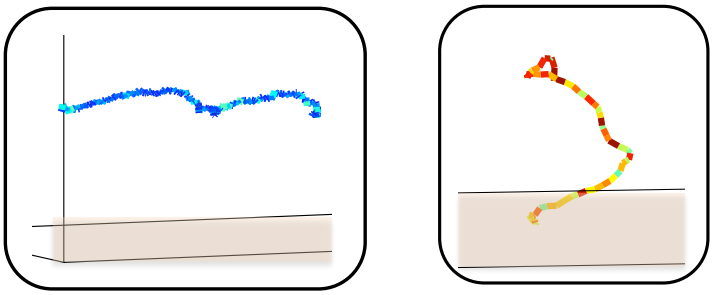
Fig. 14: Confusion matrix; parameters $N = 4$, $N_v = 4$, $N_\alpha = 1$, $N_\beta = 3$ and $N_D = 3$.

The confusion matrix for these parameters is shown in Figure 14. As we can see, patterns 3 and 4 are correctly classified. The common misclassifications occur when Orientation (1) is classified as Gyration (3), or when Wobbling (2) is classified as Spinning (4). In the next section we discuss these misclassifications in detail.

5.5 Classification on other sequences

In this section, we present the performance of the algorithm when several patterns appear within one trajectory and also analyze the typical misclassifications. As test data

we use four sequences which contain 27, 40, 49 and 11 trajectories, respectively. We obtain classification rates of 100%, 85%, 89.8% and 100%, respectively. Note that for the third sequence, 60% of the misclassifications are only partial, which means that the model detects that there are several patterns but only one of them is misclassified.



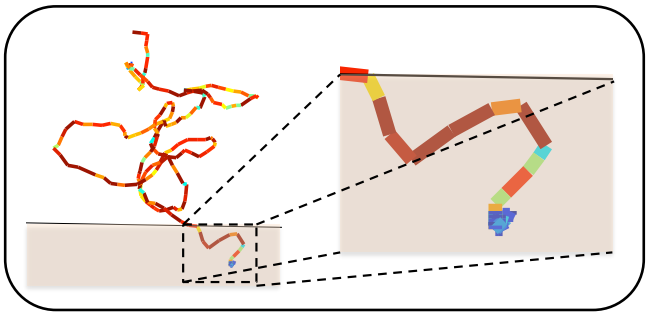
(a) (b)

Fig. 15: (a) Wobbling (pattern 2) misclassified as Spinning (4). (b) Gyration (3) misclassified as Orientation (1). Color coded according to speed as in Figure 1(b)

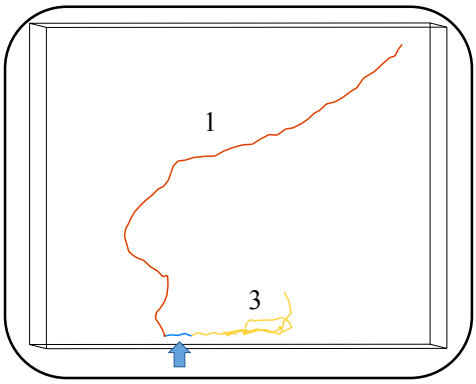
One of the misclassifications that can occur is that Wobbling (2) is classified as Spinning (4). Both motion patterns have similar speed values and the only truly differentiating characteristics are the depth and the angle α. Since we use 3 symbols for depth, the fact that the microorganism touches the surface or swims near the surface leads to the same classification. That is the case of Figure 15(a), in which the model chooses pattern Spinning (4) because the speed is very low (dark blue) and sometimes the speed in the Wobbling pattern can be a little higher (light blue).

As commented in section 4.2, Gyration (3) and Orientation (1) are two linked patterns. The behavior of gyration in solution is similar to the orientation pattern, that is why the misclassification shown in Figure 15(b) can happen. In this case, since the microorganism does not interact with the surface and the speed of the pattern is high (red color), the model detects it as an orientation pattern. We note that this pattern is difficult to classify, even for a trained expert, since the transition from orientation into gyration usually occurs gradually as spores swim towards the surface and interrupt the swimming pattern (which is very similar to the orientation pattern) by short surface contacts.

In general, the model has been proven to handle changes between patterns extremely well. In Figure 16(a), we see the transition between Gyration (3) and Spinning (4). In Figure 16(b), color coded according to classification, we can see how the model detects the Orientation part (red) and the Gyration part (yellow) perfectly well. The model performs a quick transition (marked in blue) and during this period the model stays in the SWITCH state. We have verified that all the transition periods detected by the model lie within the manually annotated transition boundaries marked by experts, even when there is more than one transition present in a trajectory.



(a) Gyration (3) + Spinning (4). Zoom on the spinning part. Color coded according to speed as in Figure 1(b).



(b) Orientation (1, red) + Gyration (3, yellow). Transition marked in blue and pointed by an arrow.

Fig. 16: Sequences containing two patterns within one trajectory

The classification results on a full sequence are shown in Figure 17.

Finally, we can obtain the probability of each transition (e.g. from Orientation to Spinning) for a given dataset under study. This is extremely useful for experts to understand the behavior of a certain microorganism under varying conditions.

6 Conclusions

In this paper we presented a fully-automatic method to analyze 4D digital in-line holographic microscopy videos of moving microorganisms by detecting the microorganisms, tracking their full trajectories and classifying the obtained trajectories into meaningful motion patterns.

The detection of the microorganisms is based on a simple blob detector and can be easily adapted for any microorganism shape. To perform multiple object tracking, we modified the standard Hungarian graph matching algorithm, so that it is able to overcome the disadvantages of the classical approach. The new multi-level Hungarian

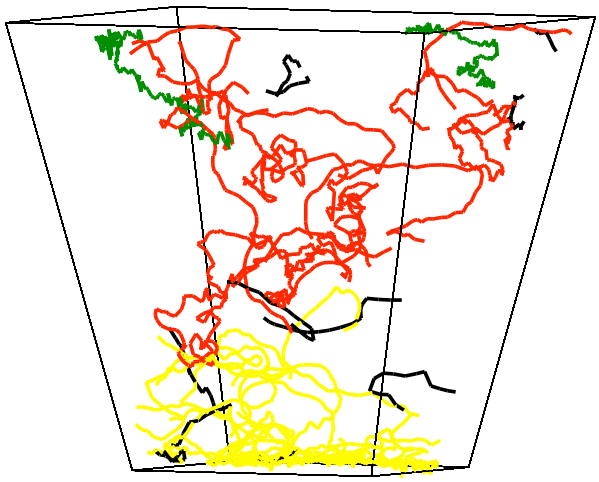


Fig. 17: Complete volume with patterns: Orientation (1, red), Wobbling (2, green), Gyration (3, yellow). The Spinning (4) pattern is not present in this sequence. Patterns which are too short to be classified are plotted in black.

recovers from missing data, discards outliers and is able to incorporate geometrical information in order to account for entering and leaving particles. The automatically determined trajectories are compared with ground truth data, proving the method detects between 75% and 90% of the labeled particles.

For motion pattern classification, we presented a simple yet effective hierarchical design which combines multiple trained Hidden Markov Models (one for each of the patterns), and has proved successful to identify different patterns within one single trajectory. The experiments performed on four full sequences result in a total classification rate between 83.5% and 100%.

Our system is proved to be a helpful tool for biologists and physicists as it provides a vast amount of analyzed data in an easy and fast way. As future work, we plan on further improving the tracking results by using a network flow approach, which will be specially useful for volumes with high density of microorganisms.

Acknowledgements. This work has been funded by the German Research Foundation, DFG projects RO 2497/7-1 and RO 2524/2-1 and by the Office of Naval Research, grant N00014-08-1-1116.

References

1. Ginger, M., Portman, N., McKean, P.: Swimming with protists: perception, motility and flagellum assembly. Nature Reviews Microbiology **6**(11) (2008) 838–850
2. Stoodley, P., Sauer, K., Davies, D., Costerton, J.: Biofilms as complex differentiated communities. Annual Review of Microbiology **56** (2002) 187–209

3. Heydt, M., Rosenhahn, A., Grunze, M., Pettitt, M., Callow, M.E., Callow, J.A.: Digital inline holography as a 3d tool to study motile marine organisms during their exploration of surfaces. The Journal of Adhesion **83**(5) (2007) 417–430
4. Rosenhahn, A., Ederth, T., Pettitt, M.: Advanced nanostructures for the control of biofouling: The fp6 eu integrated project ambio. Biointerphases **3**(1) (2008) IR1–IR5
5. Frymier, P., Ford, R., Berg, H., Cummings, P.: 3d tracking of motile bacteria near a solid planar surface. Proc. Natl. Acad. Sci. U.S.A. **92**(13) (1995) 6195–6199
6. Baba, S., Inomata, S., Ooya, M., Mogami, Y., Izumikurotani, A.: 3-dimensional recording and measurement of swimming paths of microorganisms with 2 synchronized monochrome cameras. Review of Scientific Instruments **62**(2) (1991) 540–541
7. Weeks, E., Crocker, J., Levitt, A., Schofield, A., Weitz, D.: 3d direct imaging of structural relaxation near the colloidal glass transition. Science **287**(5452) (2000) 627–631
8. Li, K., Miller, E., Chen, M., Kanade, T., Weiss, L., Campbell, P.: Cell population tracking and lineage construction with spatiotemporal context. Medical Image Analysis **12**(5) (October 2008) 546–566
9. Miura, K.: Tracking movement in cell biology. Microscopy Techniques (2005) 267–295
10. Tsechpenakis, G., Bianchi, L., Metaxas, D., Driscoll, M.: A novel computation approach for simultaneous tracking and feature extraction of c. elegans populations in fluid environments. IEEE Transactions on Biomedical Engineering **55**(5) (May 2008) 1539–49
11. Khan, Z., Balch, T., Dellaert, F.: Mcmc-based particle filtering for tracking a variable number of interacting targets. TPAMI (2005)
12. Nillius, P., Sullivan, J., Carlsson, S.: Multi-target tracking - linking identities using bayesian network inference. CVPR (2006)
13. Yang, M., Yu, T., Wu, Y.: Game-theoretic multiple target tracking. ICCV (2007)
14. Betke, M., Hirsh, D., Bagchi, A., Hristov, N., Makris, N., Kunz, T.: Tracking large variable number of objects in clutter. CVPR (2007)
15. Lu, J., Fugal, J., Nordsiek, H., Saw, E., Shaw, R., Yang, W.: Lagrangian particle tracking in three dimensions via single-camera in-line digital holography. New J. Phys. **10** (2008)
16. Berg, H.: Random walks in biology. Princeton University Press (1993)
17. Hoyle, D., Rattay, M.: Pca learning for sparse high-dimensional data. Europhysics Letters **62**(1) (2003)
18. Wang, X., Grimson, E.: Trajectory analysis and semantic region modeling using a nonparametric bayesian model. CVPR (2008)
19. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning **46**(1-3) (2004) 389–442
20. Sbalzariniy, I., Theriot, J., Koumoutsakos, P.: Machine learning for biological trajectory classification applications. Center for Turbulence Research (2002) 305–316
21. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2) (1989)
22. Chen, M., Kundu, A., Zhou, J.: Off-line handwritten word recognition using a hidden markov model type stochastic network. TPAMI **16** (1994)
23. Nefian, A., M.H.Hayes: Hidden markov models for face recognition. ICASSP (1998)
24. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden markov model. CVPR (1992)
25. Brand, M., Kettnaker, V.: Discovery and segmentation of activities in video. TPAMI **22**(8) (2000) 844–851
26. Leal-Taixé, L., Heydt, M., Rosenhahn, A., Rosenhahn, B.: Automatic tracking of swimming microorganisms in 4d digital in-line holography data. IEEE WMVC (2009)
27. Leal-Taixé, L., Heydt, M., Weisse, S., Rosenhahn, A., Rosenhahn, B.: Classification of swimming microorganisms motion patterns in 4d digital in-line holography data. DAGM (2010)

28. Gabor, D.: A new microscopic principle. Nature **161**(8) (1948) 777
29. Xu, W., Jericho, M., Meinertzhagen, I., Kreuzer, H.: Digital in-line holography for biological applications. Proc. Natl. Acad. Sci. U.S.A. **98**(20) (2001) 11301–11305
30. Raupach, S., Vossing, H., Curtius, J., Borrman, S.: Digital crossed-beam holography for in situ imaging of athmospheric particles. J. Opt. A: Pure Appl. Opt. **8** (2006) 796–806
31. Fugal, J., Schulz, T., Shaw, R.: Practical methods for automated reconstruction and characterization of particles in digital in-line holograms. Meas. Sci. Technol. **20** (2009) 075501
32. Heydt, M., Divós, P., Grunze, M., Rosenhahn, A.: Analysis of holographic microscopy data to quantitatively investigate three dimensional settlement dynamics of algal zoospores in the vicinity of surfaces. Eur. Phys. J. E: Soft Matter and Biological Physics (2009)
33. Garcia-Sucerquia, J., Xu, W., Jericho, S., Jericho, M.H., Tamblyn, I., Kreuzer, H.: Digital in-line holography: 4d imaging and tracking of microstructures and organisms in microfluidics and biology. Proc. SPIE **6026** (2006) 267–275
34. Lewis, N.I., Xu, W., Jericho, S., Kreuzer, H., Jericho, M., Cembella, A.: Swimming speed of three species of alexandrium (dinophyceae) as determined by digital in-line holography. Phycologia **45**(1) (2006) 61–70
35. Sheng, J., Malkiel, E., Katz, J., Adolf, J., Belas, R., Place, A.: Digital holographic microscopy reveals prey-induced changes in swimming behavior of predatory dinoflagellates. Proc. Natl. Acad. Sci. U.S.A. **104**(44) (2007) 17512–17517
36. Sheng, J., Malkiel, E., Katz, J., Adolf, J., Place, A.: A dinoflagellate exploits toxins to immobilize prey prior to ingestion. Proc. Natl. Acad. Sci. U.S.A. **107**(5) (2010) 2082–2087
37. Sun, H., Hendry, D., Player, M., Watson, J.: In situ underwater electronic holographic camera for studies of plankton. IEE Journal of Oceanic Engineering **32**(2) (2007) 373–382
38. Lindeberg, T.: Scale-space theory in computer vision. Springer (1994)
39. Kuhn, H.: The hungarian method for the assignment problem. Nav. Res. Logist. **2** (1955) 83–87
40. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society of Industrial and Applied Mathematics **5**(1) (1957) 32–38
41. Pilgrim, R.: Munkres' assignment algorithm; modified for rectangular matrices. Course Notes, Murray State University. http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html
42. Masuda, N., Ito, T., Kayama, K., Kono, H., Satake, S., Kunugi, T., Sato, K.: Special purpose computer for digital holographic particle tracking velocimetry. Optics Express **14** (2006) 587–92
43. Iken, K., Amsler, C., Greer, S., McClintock, J.: Qualitative and quantitative studies of the swimming behaviour of hincksia irregularis (phaeophyceae) spores: ecological implications and parameters for quantitative swimming assays. Phycologia **40** (2001) 359–366