Machine Vision and Applications manuscript No.
(will be inserted by the editor)

Multi-model particle filter-based tracking with switching dynamical state to study bedload transport

Hugo Lafaye de Micheaux · Christophe Ducottet · Philippe Frey

Received: 21 December 2016 / Accepted: 25 March 2018

Abstract Multi-object tracking is a difficult problem underlying many computer vision applications. In this work, we focus on bedload sediment transport experiments in a turbulent flow were sediments are represented by small spherical calibrated glass beads. The aim is to track all beads over long time sequences to obtain sediment velocities and concentration. Classical algorithms used in fluid mechanics fail to track the beads over long sequences with a high precision because they incorrectly handle both miss-detections and detector imprecision. Our contribution is to propose a particle filter-based algorithm including a multiple motion model adapted to our problem. Additionally, this algorithm includes several improvements such as the estimation of the detector confidence to account for the lack of precision of the detector. The evaluation was made using two test sequences - one from our experimental setup and one from a simulation created numerically - with their dedicated ground truths. The results show that this algorithm outperforms state-of-the-art concurrent algorithms.

Keywords Visual object tracking · multiple targets tracking · particle filter · switching dynamical state · detector confidence · bedload transport

1 Introduction

Visual target tracking is a recurrent problem in computer vision, it has been used in many applications, *e.g.* video surveillance, sports analysis, or traffic safety. In particular, object tracking is used in fluid mechanics to track particles with techniques such as Particle Image Velocimetry (PIV) [43] and Particle Tracking Velocimetry (PTV) [36]. We focus here on bedload sediment transport experiments conducted with a laboratory facility, as illustrated by the sketch in Fig. 1. Experiments consist in a two-dimensional steep channel where mixtures of two-size spherical glass beads are entrained by a turbulent supercritical water flow with a mobile bed. The aim is to track all beads (see Fig. 2) over a long time to obtain trajectories, particle velocities and concentrations [16], for studying bedload granular rheology [31], size segregation [5] and associated morphology [39].

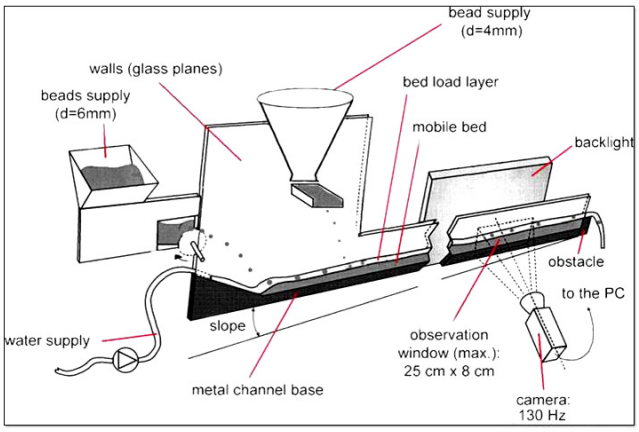


Fig. 1 Sketch of the experimental arrangement (modified from [9, 19]). Water flows from left to right, entraining a mixture of two-size spherical glass beads. Black beads of diameter d=6mm are introduced upstream and transparent beads of diameter d=4mm further downstream. The mobile bed is observed through a backlighted window and recorded by a high-speed video camera at 130Hz.

In this context, PIV with background modeling-based trackers can be used to detect moving objects by background subtraction [20, 42]. But the main problem with these methods is that it does not resolved strong gradients (*e.g.* of ve-

H. Lafaye de Micheaux · Christophe Ducottet
Laboratoire Hubert Curien UMR5516, UJM-Saint-Étienne, CNRS, IOGS, Université de Lyon, 42023, Saint-Étienne, France
E-mail: hugo.lafaye-de-micheaux@irstea.fr

H. Lafaye de Micheaux · Philippe Frey
Université Grenoble Alpes, Irstea, UR ETGR, 2 rue de la Papeterie-BP 76, 38402 St-Martin-d'Hères, France

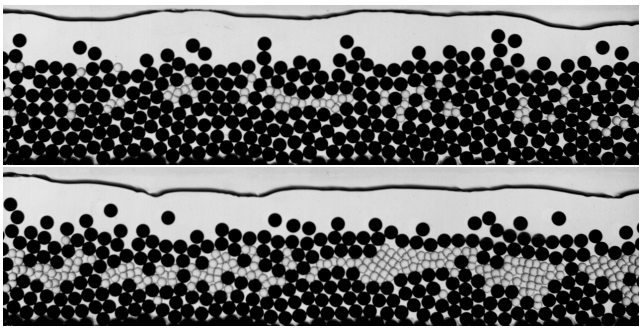


Fig. 2 Example of images from the sequence recorded during an experiment. An image with a small number of transparent beads at the beginning of the sequence (top), and an image with a much higher number of transparent beads later in the sequence (bottom). The black line on top of the images is the water free surface. The image resolution is 1280x320.

locity). Moreover, it is very sensitive to framerate, changing background, moving cameras and can remove locally non-moving objects that are not really part of the background. In body-part, person or car tracking applications, these problems can be faced through tracking-by-detection approaches to gain robustness. These approaches have become increasingly popular by global optimization based on position, size and appearance similarity [2, 18, 27]. Among these tracking methods, we focused on multiple object tracking where the aim is to follow a high number of objects in a scene individually, like in crowd tracking for example [1, 47]. In fluid mechanics, PTV algorithms are deterministic (or data association-based tracking) and rely on a standard two step approach: (1) object detection, (2) object tracking by solving a data association problem [19]. However, they are not robust enough to track objects over very long sequences with a good precision for object positions. This is mainly because they fully trust the object detector output, so they incorrectly handle miss-detections and do not take into account the lack of precision of the detector.

In recent years, particle filter-based approaches have been successfully used in many applications [12, 29, 38]. One of the main benefits of these methods is to enable the estimation of the distribution of the target's system state incorporating a non linear motion model. In this context, Breitenstein et al. [10] proposed a very efficient online multi-target tracking approach accounting for the detector confidence to handle miss-detections. However, this method is not directly adapted to our problem mainly because it uses a single motion model unable to describe all occurring object interactions.

In this paper, using particle filters and with this idea of detector confidence, we propose to introduce a Bayesian mixed-state framework [23] dedicated to our simulation of bedload transport. It is based on multiple motion models where each motion model corresponds to a specific state of motion linked with the mechanics of our objects: resting, rolling or bouncing also called saltation. Such an approach permits to follow the natural behavior of spherical beads in a fluid flow. We also introduce several improvements to correctly handle the lack of precision of the detector providing tracking over long sequences with a high precision for object positions.

The paper is structured as follows. After discussing the related work in the next section, Sec. 3 describes our proposed multiple model particle filter-based tracking algorithm and design choices. Sec. 4 gives details about the experiments and datasets we study. Sec. 5 presents the evaluation and a discussion of the results of tracking performances and a comparison with concurrent algorithms. Finally, Sec. 6 provides a conclusion of the paper.

2 Related work

This work stands at the intersection of computer vision, fluid mechanics and Bayesian statistical inference; it builds on previous approaches discussed below.

Various visual tracking approaches have been reviewed [40, 45] and they can be separated in two categories: deterministic methods and probabilistic ones. In fluid mechanics, researchers usually employ the first one, *i.e.* taking the object detector output as a straight truth and then solving the association problem on pairwise frames [9, 19]. It is also the case for the Lagrangian Particle Tracking [21, 37] widely used in experimental fluid dynamics. This kind of deterministic approach can lack precision since the association problem between detections and targets strongly relies on the object detector that can be unreliable and imprecise.

To handle this problem, several recent algorithms optimize detection assignments over a large temporal window in an offline step [3, 27, 48], *i.e.* by using information from future frames and after observing the complete sequence. Nevertheless, these methods are time-critical and unsuitable for online applications[1], unlike Sequential Monte Carlo (SMC) methods. SMC or particle filters (PF) were introduced to provide better estimations and predictions in an online manner [14, 17, 22]. Basically, they consist of a dynamic model for prediction and an observation model to evaluate the likelihood of a predicted state. The main steps are: random generation of a sample of potential states called particles, propagation of these particles through the motion model and resampling of the distribution according to the observation model.

In order to cope with difficult tracking situations in a multi-object context, Breitenstein et al. [10] proposed an automatic PF-based algorithm that exploits the detector confidence (*i.e.* the intermediate output of the object detector) as a graded observation model. It allows using a combination

[1] An application is online when the inputs to the associated algorithm is provided as a stream, piece-by-piece.

of unreliable information sources for robust multi-person tracking. As most particle filter-based algorithms, it relies on a single motion model which can be problematic in an environment with complex dynamics.

To face this limitation, a multiple model (MM) paradigm [4, 28], also called switching dynamical model approach, has been introduced and is used for several applications such as robotics and navigation [8] or video surveillance [46]. It assumes that, given a finite state of dynamic models (each of them corresponding to a particular behavior), a particular target can jump from one model to the other according to a set of transition probabilities. The interacting multiple model (IMM) algorithm [7, 24, 33] was the first to propose this approach through jump Markov process modeling. Then, the multiple model particle filter (MMPF) framework [23, 34] came as a promising alternative outperforming the IMM algorithm in a bearings-only tracking problem [4]. Both methods have also been recently combined in an IMMPF algorithm [11, 13]. The MMPF method uses the idea of MM, *i.e.* object motions can be classified in different motion states depending on the situation, in a PF framework. Each motion state is dealt with a specific state evolution model for prediction. However, it requires to know all motion states beforehand with a mathematical description for each. In contrast, our method exploits the mechanistic knowledge of behaviors we have in our application.

3 Multi-object multi-model tracking algorithm

In this section, we first discuss about the global goal and the proposed approach. Then we give the overview of our algorithm before focusing on each if its parts.

3.1 Problem formulation and proposed approach

The purpose of the experimental setup is to study the collective behavior of beads with respect to their size. More precisely we are interested in the phenomenon of infiltration called kinetic sieving [5]. At the beginning of an experiment, the bed is composed of an arrangement of large black beads. Small transparent beads are then introduced gradually and start to infiltrate in the mobile bed until a two-size equilibrium is reached (in approximately 50 min). See Sec. 4.1 for further details.

The global mechanical goal is to study the role of interactions at the bead scale in the formation of bed structures during the transition to equilibrium. This goal requires the statistical analysis of all individual trajectories along the experiment. A real time processing is not needed, but a fast post-processing is required to be able to analyze very long sequences (1 million images) in a reasonable time.

Three different motion types are used to describe the dynamics in bedload transport: resting when a bead is not moving, mostly encountered when it is deeply included inside the bed, rolling when it is rolling near the bed surface and saltating when it is bouncing on others above the bed surface. A given bead can suddenly switch from one motion type to another depending on its velocity, neighborhood and interactions with other beads.

From the tracking point of view, the main bottlenecks to solve are twofold. The first one is the diversity and abruptness of motion. Depending on the motion type, the dynamics are very different. Additionally the motion type can suddenly change and even in a given type, the bead motion is altered by interactions with neighboring beads. The second bottleneck is the weakness of the transparent beads detector. Due to 3D effects, the bead pattern is slightly changing resulting in miss-detections and lack of precision in the detection of transparent beads. Combined with the abruptness of motion, this constraint can lead to wrong associations and bad trajectories.

To face these problems, our approach is to use MMPFs. Compared to Kalman or IMM filters, the motivation to use particle filters is a non-linear observation model to handle miss-detections through the use of detector confidence information (see Sec. 3.3.2). Additionally, the multiple model approach allows incorporating the three motion models inspired by the mechanics and a switching mechanism to consider sporadic motion model changes. Compared to an IMM approach, the switching mechanism appears closer to the mechanics because a linear combination of states is not very realistic.

3.2 Algorithm overview

Our fully automatic multi-object tracking algorithm uses the same principle as [35] by embedding multiple motion models in a particle filter-based approach (MMPF). Moreover, it uses the technique of switching dynamical model introduced by [23] to increase the motion models flexibility of visual trackers. Thus, each physical trajectory is estimated by an individual particle filter (tracker), automatically initialized when a new object is detected and terminated when the object is leaving the field of view. A global overview of the algorithm is shown in Fig. 3. At each time step we perform: (1) object detection, (2) estimation of the detector confidence density, (3) association of each detection to a tracker, (4) update of the state of each particle filter according to the specific motion model and the new detection. The principle of the MMPF tracking algorithm is to predict the target state (*i.e.* target location, velocity and motion state) by applying an adapted motion model depending on the target behavior. This target state is then corrected with measurements made on the observed image to estimate the

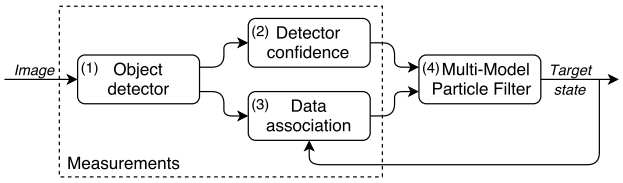


Fig. 3 Overview of the different parts of the multi-object tracking algorithm.

final target state. The target estimated at time t is further used for data association at time $t+1$. Each part of the algorithm is explained below.

3.3 Object detection

For each image, the object detection phase aims at detecting all beads by image processing. It consists in 3 steps: (1) detect black beads, (2) remove them from the image, (3) detect transparent beads. Facing lacks of reliability especially with the transparent bead detector, a special treatment is applied to the detectors (cf. Sec. 3.3.2).

3.3.1 Object detectors

First, black beads are detected by thresholding the image at level τ_b to keep only black pixels, and by taking the center of connected components having an area close to the mean bead area. This detector is very efficient as black beads are easily distinguishable from other kind of beads and they are not overlapping (6mm diameter beads in a channel width of 6.5mm). So it always returns the entire set of black bead centers D_b. Then, black beads are erased by replacing their gray levels by those of the background. The background is reconstructed through a morphological closing with a disk shaped structuring element (radius just greater than bead radius). This makes transparent beads more discernible. Secondly, since transparent beads appear as faint dark rings of different shapes because of their neighboring configurations (4mm diameter beads that can be partially overlapped by others), we use a specific chain of morphological operations to detect them (initially developed by [19]). We apply a *hconvex* operator [41] on the image obtained at the previous step, then a normalized cross-correlation with a ring-shape model, local maxima extraction and maxima selection according to the adjusted threshold τ_t. It returns the set of transparent bead centers D_t with some fallibility due to partially occluded beads. How this problem is dealt with is detailed in next section.

3.3.2 Detector confidence estimation

To handle missing detections, we introduce a detector confidence estimation and we use it as a graded observation model [10]. The principle is: based on the raw output of the detector, the detector confidence must give an estimation of the likelihood of a detection at each candidate pixel.

Assuming an exponential distribution for the position of each detection, we can estimate the likelihood of observing a bead at location $\boldsymbol{x}^{obs}$ given the set of detections D as:

$$\begin{aligned} p(\boldsymbol{x}^{obs}|D) &= d_c(\boldsymbol{x}^{obs}) \\ &= \frac{1}{k_D}\sum\nolimits_{d\in D}\mathcal{L}(d)\exp(-\lambda\left\|\boldsymbol{x}^{obs}-\boldsymbol{x}^{d}\right\|_2) \end{aligned} \quad (1)$$

where $\|.\|_2$ denotes the Euclidean distance between two points in Euclidean space, λ denotes the exponential decay constant (see Sec. 4.3), $\boldsymbol{x}^d$ is the location of a detection $d \in D$, $\mathcal{L}(d)$ is the likelihood of the detection at location $\boldsymbol{x}^d$ and k_D is the normalization coefficient.

For transparent beads, the set D corresponds to the entire set of local maxima outputs of the transparent bead detector (down to a very low threshold lower than τ_t) and $\mathcal{L}(d)$ is the value of the normalized cross-correlation with the ring shape model.

For black beads, $D = D_b$ and $\mathcal{L}(d) = 1$ as all detections have the same reliability. In practice, black beads are never missed since the corresponding detector is efficient and very reliable.

3.4 Data association

To assign at most one detection to at most one tracker, a detection-filter association process is needed. Due to the high number of detections and the long time series, the data association was done deterministically, independently from the particle filtering. We used a greedy algorithm [44] which was found to be an effective and sufficient solution with a lower computational cost than the optimal Hungarian algorithm [25].

3.4.1 Greedy data association

Given the matching cost $g(tr, d)$ between all tracker-detection pairs computed as described below, the greedy algorithm iteratively selects the best candidate and removes the corresponding concurrent associations. To limit the number of possible tracker-detection pairs, the set of possible detections associated to a given tracker is limited to the detections located inside a circular region centered at the predicted position $\hat{\boldsymbol{x}}^{tr}$ given by this tracker. This predicted position is made assuming a motion model corresponding to the motion state of the tracker at previous time step.

3.4.2 Matching cost

The matching cost $g(tr, d)$ between a detection d and a tracker tr consists of a combination of two factors: one based

on the distance of the detection position $\boldsymbol{x}^d$ to the predicted position of the tracker $\hat{\boldsymbol{x}}^{tr}$, the other based on the tracker velocity at the previous time step $\boldsymbol{u}^{tr}$. It is defined as:

$$g(tr,d) = \underbrace{\alpha \frac{\left\|\boldsymbol{x}^d - \hat{\boldsymbol{x}}^{tr}\right\|_2}{r_s}}_{\text{distance term}} + \underbrace{\beta \frac{|\boldsymbol{u}^{tr}|}{|\boldsymbol{u}|_{max}}}_{\text{velocity term}} \tag{2}$$

where $|.|$ denotes the Euclidean norm of a vector. The distance term promotes the detections closest to the prediction. With the velocity term, we promote low velocities as it allows better associations in case of bead collisions. Indeed, for example when a bead bounces, its predicted position (assuming a constant velocity model) can fall very close to another bead leading to the same distance term for the corresponding trackers. In this case, adding a velocity term in the matching cost permits to increase the cost of higher velocities much more than lower ones, and so to favor choosing lower velocities first during the greedy data association.

When associating the first two frames, the velocity term is set to 0 since we do not have a velocity value yet. The value $|\boldsymbol{u}|_{max}$ is a normalization coefficient equal to the highest velocity norm in the previous time step, the parameters α and β represent the weight of each term (see Sec. 4.3), and r_s is the radius of the circular searching region calculated from the fluid velocity, it is used here as a normalization coefficient. Therefore, the greedy algorithm returns the subset of detections associated to a tracker. The location $\boldsymbol{x}^d$ (resp. velocity $\boldsymbol{u}^d$) of the detection is considered as the observation variable $\boldsymbol{x}^{obs}$ (resp. $\boldsymbol{u}^{obs}$) of this tracker. The velocity $\boldsymbol{u}^{obs}$ is measured between the raw detection of the target at previous time step (the direct output of the detector before filtering) and the raw detection of the target in the current image. Additionally, motion state s^{obs} will be associated to this tracker (see Sec. 3.5.1).

3.5 Multi-model particle filtering

Each tracker (or trajectory) is described by a separate particle filter of N particles. At time step t the state space $\boldsymbol{c}_t^i$ of a particle i is defined as $\{\boldsymbol{x}_t^i, \boldsymbol{u}_t^i, s_t^i\}$ where $\boldsymbol{x}_t^i = (x, y)$ denotes the position, $\boldsymbol{u}_t^i = (u, v)$ the velocity and s_t^i the motion state. The observation variable at time t is given by $\boldsymbol{z}_t = \{\boldsymbol{x}_t^{obs}, \boldsymbol{u}_t^{obs}, s_t^{obs}\}$ resulting from measurement of location, velocity and motion state. We employ the Sequential Importance Resampling filter, also known as the bootstrap filter [17] where the state transition density is used as importance distribution, to approximate the probability function. For one target, the algorithm goes through different steps described in Fig. 4. It works as follows: (1) Measurement of target motion state according to neighbors configuration and target velocity, (2) Random selection of new motion states according to conditional probability table, (3) Update of particle positions and velocities according to motion models, (4) Particle importance weighting, normalizing and resampling, (5) Estimation of final motion state, position and velocity of the target. The main difference with the algorithm in [26], is the use of the target motion state estimation as an observation in the importance weighting (see Sec. 3.5.2).

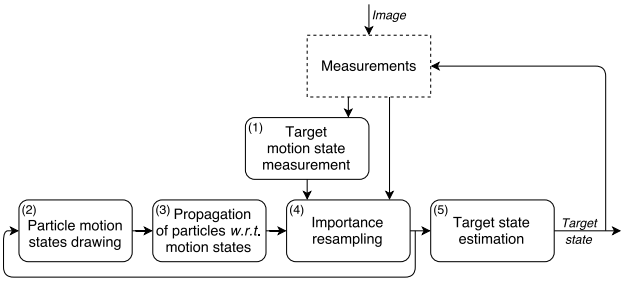


Fig. 4 The multi-model particle filter-based tracking algorithm.

3.5.1 Motion models based on mechanical dynamics

In motion-based stochastic tracking [35], explicit motion measurements are used to guide predictions. In our application, beads have different behaviors according to their location, velocity and neighborhood. We can distinguish three distinct motion states linked to the dynamics in bedload transport: resting (not moving), rolling (rolling/sliding on others) and saltating (bouncing on others). In the saltating state, we can discriminate two sub-states: a constant motion and a rebound motion. To propagate a particle i at time step t, we first update its previous motion state s_{t-1}^i, and then we apply a specific motion model to update its position and velocity. The new updated motion state s_t^i is drawn according to a conditional probability table π (see Sec. 4.3). The motion models are:

$$\begin{aligned}
\text{resting: } (x,y)_t &= (x,y)_{t-1} + \varepsilon_{rest}^{(x,y)} \\
\text{rolling: } (x,y)_t &= (x,y)_{t-1} + (u,0)_{t-1}\Delta t + \varepsilon_{roll}^{(x,y)} \\
(u,v)_t &= (u,0)_{t-1} + \varepsilon_{roll}^{(u)} \\
\text{saltating-constant: } (x,y)_t &= (x,y)_{t-1} + (u,v)_{t-1}\Delta t + \varepsilon_{salt}^{(x,y)} \\
(u,v)_t &= (u,v)_{t-1} + \varepsilon_{salt}^{(u,v)} \\
\text{saltating-rebound: } (x,y)_t &= (x,y)_{t-1} + (u,-v)_{t-1}\Delta t + \varepsilon_{salt}^{(x,y)} \\
(u,v)_t &= (u,-v)_{t-1} + \varepsilon_{salt}^{(u,v)}
\end{aligned} \tag{3}$$

where $\varepsilon_{rest}^{(x,y)}$, $\varepsilon_{roll}^{(x,y)}$, $\varepsilon_{salt}^{(x,y)}$ are the process noises on the position and $\varepsilon_{roll}^{(u)}$, $\varepsilon_{salt}^{(u,v)}$ are the process noises on the velocity (see Sec. 4.3), they are all independently drawn from zero-mean normal distributions ; Δt depends on the framerate of the sequence. At initialisation, the velocity of particles drawn in the saltating motion state is taken equal to a fraction of the velocity of the water flow.

Motion state measurement. Before applying the sequential importance resampling filtering on a target, it is possible to make an estimation of its motion state with the observations. The measurement of this motion state is computed by inspecting the neighbors configuration and velocity of the target with the help of the data association output. Indeed the idea is to use our mechanistic knowledge to best fit the mechanical dynamics of our objects. Given the number of neighboring beads nbN^{obs} touching the observed bead, we can consider the motion state to be "resting" if either nbN^{obs} is significantly high or the velocity norm $|\boldsymbol{u}^{obs}|$ of the bead is low. Similarly, the motion state will be "saltating" if nbN^{obs} is low or $|\boldsymbol{u}^{obs}|$ is high. Formally, we have:

$$s^{obs} = \begin{cases} \text{resting}, & \text{if } \text{nbN}^{obs} \geq \text{nbN}_{rest} \text{ or } |\boldsymbol{u}^{obs}| \leq u_{rest}, \\ \text{saltating}, & \text{if } \text{nbN}^{obs} \leq \text{nbN}_{salt} \text{ or } |\boldsymbol{u}^{obs}| \geq u_{salt}, \\ \text{rolling}, & \text{otherwise.} \end{cases} \tag{4}$$

where nbN_{rest} (resp. nbN_{salt}) is the threshold number of neighbors for resting (resp. saltating) and u_{rest} (resp. u_{salt}) is the threshold velocity norm for resting (resp. saltating). These values are determined experimentally on the test sequence (see Sec. 4.3).

The estimation of the motion state is used as a cue in the particle weighting (in Sec. 3.5.2) or to set the final motion state of the target (in Sec. 3.5.3).

3.5.2 Particle importance weighting

Observation model. The importance weight w_t^i for a particle i at time step t is updated from its value at previous time step w_{t-1}^i and from the conditional likelihood of the new observation $\boldsymbol{z}_t$ given the previous state $\boldsymbol{c}_{t-1}^i$ of the particle:

$$w_t^i \propto w_{t-1}^i \, p(\boldsymbol{z}_t | \boldsymbol{c}_{t-1}^i) \tag{5}$$

Thanks to resampling in each time step, all weights are the same at the end and $w_{t-1}^i = 1/N$, thus this term can be ignored. Moreover, in practice, the prior is usually chosen as the importance distribution to approximate the probability density function [17]. With these simplifications applied to Eq. (5), the particle weighting only depends on the propagated particle state and is so quite simpler:

$$w_t^i \propto p(\boldsymbol{z}_t | \boldsymbol{c}_t^i) \tag{6}$$

Importance weighting. Assuming independence between observation variables, we have:

$$\begin{aligned} p(\boldsymbol{z}_t | \boldsymbol{c}_t^i) &= p(\{\boldsymbol{x}_t^{obs}, \boldsymbol{u}_t^{obs}, s_t^{obs}\} | \boldsymbol{c}_t^i) \\ &= p(\boldsymbol{x}_t^{obs} | \boldsymbol{c}_t^i) p(\boldsymbol{u}_t^{obs} | \boldsymbol{c}_t^i) p(s_t^{obs} | \boldsymbol{c}_t^i) \end{aligned} \tag{7}$$

We have chosen to give a high confidence to the observed velocity which corresponds to select only particles having a velocity component equal to $\boldsymbol{u}_t^{obs}$. Practically, this comes down to remove the velocity likelihood in Eq. (7) and set each velocity component of the state to $\boldsymbol{u}_t^{obs}$ at the weighting step. This choice aims to better estimate the state density for locations and state variables with a low number of particles (*e.g.* 100) in particle filters.

Sometimes, there is not a detection associated to the tracker and this is most often caused by a false negative. In such cases, we need to continue having an estimation of the tracker state not to lose it until a detection is associated to it again. To do so, the likelihood of particle i at location $\boldsymbol{x}_t^i$ of the tracker tr is given by the detector confidence (see Eq. (1)). But in most cases, *i.e.* when a detection is associated to the tracker, it is estimated by a normal distribution centered on the corresponding detection d (see Sec. 4.3 for the variance value). Formally:

$$p(\boldsymbol{x}_t^{obs} | \boldsymbol{c}_t^i) = \begin{cases} p_{\mathcal{N}}(\boldsymbol{x}_t^i - \boldsymbol{x}_t^{obs}) & \text{if } \mathcal{I}(tr) = 1, \\ d_c(\boldsymbol{x}_t^i) & \text{otherwise.} \end{cases} \tag{8}$$

where $\mathcal{I}(tr)$ is an indicator function that returns 1 if a detection is associated to the tracker and 0 otherwise, it is given by the data association.

For the motion state likelihood $p(s_t^{obs} | \boldsymbol{c}_t^i)$, we assume a constant value γ if $s_t^i = s_t^{obs}$ and $1 - \gamma$ otherwise. This value is set experimentally (see Sec. 4.3). Formally:

$$p(s_t^{obs} | \boldsymbol{c}_t^i) = p_\gamma(s_t^i) = \begin{cases} \gamma, & \text{if } s_t^i = s_t^{obs}, \\ 1 - \gamma, & \text{otherwise.} \end{cases} \tag{9}$$

Thus, the update of weights is given by:

$$w_t^i = \begin{cases} p_\gamma(s_t^i) p_{\mathcal{N}}(\boldsymbol{x}_t^i - \boldsymbol{x}_t^{obs}) & \text{if } \mathcal{I}(tr) = 1, \\ p_\gamma(s_t^i) d_c(\boldsymbol{x}_t^i) & \text{otherwise.} \end{cases} \tag{10}$$

Normalizing and resampling. After weighting all particles, the weights are normalized so that $\sum_{i=1}^N w_t^i = 1$. Then, the resampling is used to solve particle degeneracy, namely removing particles with low weights and multiplying those with high weights. The approach used here is the multinomial resampling based on the bootstrap method [15].

3.5.3 Final estimation

Finally, the position of the target is estimated by averaging the resampled particles. For the velocity, according to the choice of promoting only particles having the observed velocity, the final estimate is exactly equal to this observed velocity. After all positions and velocities of the targets are estimated, we measure again their motion state the same way it was done before the particle importance weighting (see Sec. 3.5.1 and Eq. (4)). A classical approach could be to choose the most represented state but we chose to reassess the motion state to ensure to be consistent with the mechanical motion state.

Indeed, we observed a slight mismatch between the predicted state and the re-estimated state based on the predicted position and velocity. A detailed analysis of this mismatch shows it is due to discretization issue. Thus, in some situations and when the number of particles is too low (*e.g.* 100) regarding the number of state variables, after importance weighting, the correct motion state is not the most represented one. This mismatch does not occur with more particles in the particle filter (*e.g.* 500).

4 Experiments

In this section, we present the detailed experimental protocol we used to produce the experimental image sequences of bedload transport. Then, we describe the datasets used for the evaluation of our algorithm. Finally, we give some details about the implementation of the algorithm with the parameter settings.

4.1 Experimental protocol

The experimental protocol to reproduce bedload transport in the experimental flume illustrated in Fig. 1 consists in forming a two-size mobile bed to promote exchanges between the moving beads and the bed. First of all, a mobile bed of black beads is created by feeding water and black beads at constant flow rates into the empty flume which ends with an obstacle. Once the equilibrium is reached (after approx. 10 min), *i.e.* when the output rate of black beads is equal to the input rate, the feeding of transparent beads begins at constant flow rate, while water and black beads continue to be fed. The transparent beads rebound, move and start to infiltrate in the mobile bed. This phenomenon of infiltration is called kinetic sieving [5]. After a little time, a new two-size equilibrium is finally reached (after approx. 50 min). Each experiment is recorded over the entire duration and a sequence of images is obtained. These images are subsequently analysed with the procedure described in this study.

4.2 Datasets and ground truths

To test our tracking algorithm, we worked on two datasets: a short experimental sequence obtained with the previously detailed experiment and a longer numerical one obtained with a simulation. Both following datasets are freely available with their ground truth[2].

Experimental sequence. The experimental sequence is a 1,000-frame sequence recorded at 130 fps with approximately 400 beads per frame (about 300 coarse and 100 small beads). The image resolution is 1280x320. The sequence is split in two parts, one for optimizing and estimating the parameters, and the other part for evaluation. We created a ground truth on this sequence by first detecting beads with our object detectors on each frame (Sec. 3.3.1), and then by editing visually the detections, *i.e.* by removing the false positives and adding manually the false negatives. To get the ground truth of trajectories, we applied our data association algorithm (Sec. 3.4) on the ground truth of detections just created and then computed the motion states of each target by watching neighbors configuration (as in [19]). The limitation of this ground truth is the precision of the positions. Indeed, the center of missing detections were selected manually to fit visually the center of beads as good as possible. So they can be inaccurate by one or two pixels. Moreover, as we exploited the raw output of the detector to create this ground truth, the tracking algorithms cannot use it directly to study precision of estimated positions. To overcome this problem, we considered the detected locations as the truth and we added white noise on the locations given by the detector to simulate real measurements. Thus, all the classical metrics can be used to evaluate the performance of algorithms on this experimental sequence denoted ExpSeq, an example of trajectories is given in Fig. 5 (top).

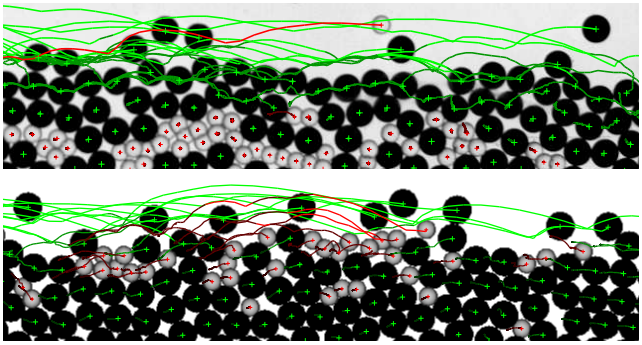


Fig. 5 Example of trajectories for the experimental ExpSeq (top) and simulated NumSeq (bottom) datasets (cropped images). Green color is for a coarse black bead and red is for a smaller transparent bead. The green and red color gradations from darker to lighter show respectively the three motion states (resting, rolling and saltating).

Numerical sequence. In order to study a longer sequence and lower frequency phenomena, we created a numerical sequence of bedload transport of 10,000 images thanks to a model developed at Irstea based on a coupled fluid discrete element method (cf. [30], [31] and [32]) and validated against our experiments (cf. [16]). The idea of this simulation is to reproduce as best as possible our experiments (*e.g.* flume size, number of each type of beads, feeding rates, fluid velocity, image frequency, etc.). Moreover, thanks to ParaView, an open source software used for scientific visualization, we were able to have a rendering of images of the obtained sequence really close to our experimental images. Thanks to

[2] https://perso.univ-st-etienne.fr/ducottet/

this rendering, we were able to use this numerical sequence in the same manner as an experimental sequence and apply all the previously presented algorithms. As for ExpSeq, half of the sequence is used to optimize and estimate the parameters, and the other for evaluation. The numerical sequence will be denoted NumSeq, an example of trajectories is given in Fig. 5 (bottom).

Ground truth motion states. On the obtained ground truth sequences, we need to assign a motion state to each tracker at each time step. To do this, we follow the offline method initially developed by Böhm et al. [9] which consists in combining two criteria: first testing the tracker velocity (averaged over five consecutive frames) against a threshold velocity and second testing the distance between the tracker and its nearest neighbor against a reference distance. The main difference with the way we measure the motion states in our algorithm is that here it is done after the tracking execution in a post-processing as it uses information from future frames, while we do it during the tracking execution in an online manner using current velocities. So the velocity thresholds here do not have the same values, the ones of the offline way of doing are more restrictive. We consider that the offline method gives motion states close enough to real motion behaviors to be used as ground truth.

4.3 Implementation details

Free parameter tuning. The number of particles N was fixed to $N = 100$, being a good compromise between computational cost and tracking reliability. The parameters in Eqs. (1) and (2) have been set experimentally and remained the same for the different experiments used in Sec. 4.2. λ was set to 1 in Eq. (1). In Eq. (2), α and β were set respectively to 0.75 and 0.25 to give more importance to the distance term. The variance of the normal distribution in Eq. (8) was set experimentally to 3 pixels for both x- and y-coordinates. The reward parameter γ in Eq. (9) was fixed to $2/3$ in order to promote particles having the same motion state as the measurement.

Process noise and tracker parameters. Each motion state in Eqs. (3) has its own variance for the process noise on position and velocity. They were set automatically by analyzing representative training sequences corresponding to the first halves of the studied sequences given in Sec. 4.2. The initial particle positions and velocities were drawn from a Normal distribution with positions centered at the detection center and velocities centered to null velocity. To handle the difficulty of dealing with new trackers, we increased the variances to make the motion model more flexible during the first 3 frames. To stop bad predictions, a tracker at resting state (resp. rolling and saltating) survives only 100 frames (resp. 10 frames and 5 frames) without associated detection and is then automatically terminated.

Transition probability table. For a given particle i, the transition from one motion state s^i_{t-1} to another s^i_t is controlled with conditional probabilities. We have estimated these probabilities on representative training sequences corresponding to the first halves of the studied sequences given in Sec. 4.2. As the probabilities were found to be nearly the same for all datasets, we present only one case here. The transition probability table π required for the switching state process presented in Sec. 3.5.1, was estimated as:

$$\pi = p(s^i_t | s^i_{t-1}) = \begin{bmatrix} 0.95 & 0.05 & 0.00 \\ 0.07 & 0.92 & 0.01 \\ 0.00 & 0.09 & 0.91 \end{bmatrix} \tag{11}$$

As N was fixed to 100 and we want each state to have a chance to be represented at each time step, we modified the transition probability table to be slightly less restrictive by imposing a minimal probability of 0.05, leading to:

$$\pi \approx \begin{bmatrix} 0.80 & 0.15 & 0.05 \\ 0.10 & 0.80 & 0.10 \\ 0.05 & 0.15 & 0.80 \end{bmatrix} \tag{12}$$

The saltating state being divided in two sub-states, when it is drawn after getting through π, there is then a probability of 0.5 to be in the saltating-constant sub-state (resp. saltating-bouncing sub-state). At initialization, the particle motion states were set as *unknown* states. At the next step, the particles had an equivalent chance to go in any of the motion states.

Motion state measurement thresholds. To measure the motion state of a target (Eq. (4)), we need to calculate its number of neighbors. We consider a detection as a neighbor of the target if they are in contact, *i.e.* when the distance between both centers is equal to the sum of their corresponding radius. The thresholds on the number of neighbors for the resting and saltating states were set experimentally to $\text{nbN}_{rest} = 5$ and $\text{nbN}_{salt} = 0$ respectively. The thresholds on velocity norms were set experimentally to $u_{rest} = 0.015m/s$ and $u_{salt} = 0.25m/s$ for the sequences studied in this paper (cf. Sec. 4.2). u_{rest} and u_{salt} are specific to each sequence. These four thresholds were adjusted by trial and error to minimize the differences between the motion state determined by the algorithm and the motion state determined by the naked eye.

5 Results and discussion

In this section, we present an analysis of some execution information and the results of the proposed tracking algorithm against other algorithms of the literature. They are all

compared to a ground truth of a short experimental sequence ExpSeq and a longer numerical one NumSeq (see Sec. 4.2). Indeed, as our case study is very specific, it is impossible to test our tracking algorithm on open datasets. We also use the CLEAR MOT metrics [6] to evaluate the tracking performance. Moreover, we study how the algorithms deal with different detector qualities and how well they estimate the motion states.

5.1 Execution analysis

Sensitivity of parameters. As mentioned before, some parameters were individually adjusted on both ExpSeq and NumSeq. We saw that these parameters were nearly identical. So we tested the parameters used for a given dataset on the other one and vice-versa and we observed that it did not change the results. From this observation, we have chosen to use only one set of parameters for both datasets. The only parameters to adjust are the thresholds on the detectors as they depend on the imaging setup.

Execution time. For a sequence of 5,000 images and a number of $N = 100$ particles, the execution time on a workstation with an Intel Core i7 2.70GHz and 8GB of memory is approximately 10 hours. On this duration, 15% is used for the detection of the beads and 85% for their tracking across the sequence. Note that this duration is reasonable for the remote analysis of an experiment and it could be reduced significantly since the algorithm has been implemented in Matlab.

5.2 Algorithms

To study the reliability and efficiency of our method, we tested it against four other algorithms of the literature. First of all, we have our particle filter-based tracking algorithm (see Sec. 3) which is based on the use of three motion models (or states), the detector confidence and a measurement of the motion state as a cue in the likelihood (MMPF+). Second, we have our preliminary work on three motion models-based particle filters with detector confidence [26] (MMPF). Third, we have an adaptation of the algorithm of Breitenstein et al. [10] based on a single constant velocity model and detector confidence particle filters (PF). Fourth, we have an interacting multiple model tracking algorithm [7] that uses our three motion models described as Kalman filters (IMM). The model and noise parameters are set the same as the MMPF+ ones. Finally, we have an initial and simple deterministic tracking approach previously developed in our laboratory [19] (DET). This last algorithm blindly relies on the output of the detector, without any special treatment for the false positives and false negatives.

5.3 Evaluation of the tracking algorithm

To evaluate the tracking performance, we use the CLEAR MOT metrics [6]. It gives a precision score MOTP (intersection over union (IoU) of the estimated bead and the ground truth) and an accuracy score MOTA (accounts for all object configuration errors, *i.e.* false negatives, false positives and mismatches[3]). The IoU threshold used here was 0.43, which rejects the estimated beads having their center outside the area of the ground truth bead. We also use the percentage of correct tracks, where a track is considered as correct on its total length if more than 95% of it has no false negatives or mismatches. In order to see the influence of the detector output on the different tracking algorithms, we vary the threshold for the detection of the transparent beads. It gives different detector qualities and allows seeing how the tracking algorithms deal with more false negatives or false positives. At last, we study how well the multiple motion model-based algorithms estimate the motion state of the beads.

Quantitative analysis. In Table 1, we present the evaluation results of the five tracking algorithms MMPF+, MMPF, PF, IMM and DET on both datasets ExpSeq and NumSeq. They show that MMPF+ has the best results in terms of accuracy in light of the percentage of correct tracks, number of mismatches and MOTA score. This comes from its good management of the false negatives. In terms of precision through the MOTP score, IMM and DET seem better than the others. However, with more particles (*e.g.* $N = 500$), the MMPF+'s precision climbs up to be as good as the IMM's. When compared to the other particle filter-based algorithms MMPF and PF, MMPF+ is the most performing in all metrics. For example, our precision MOTP and accuracy MOTA results are higher for both datasets. Note that all precision and accuracy scores are already high. This is due to the good quality of the transparent bead detectors for the chosen thresholds. We chose the threshold of the transparent bead detector $\tau_t = 0.25$ for ExpSeq and $\tau_t = 0.27$ for NumSeq as they give the best tracking results and as we know the importance of good detectors for reliable tracking operations. But the detector is not always that efficient as it depends on the image quality of the sequences. The analysis of the effect of different τ_t is done below.

We can also observe that the number of identity switches is the lowest for MMPF+. This property is one of the most important for a good tracking algorithm. In our images, ID switches can happen for tracks changing abruptly of direction (*e.g.* rebounding tracks) or velocity (*e.g.* acceleration $>> 0$), or for newly appearing beads located really close to where a track should be according to its motion model. The implementation of multiple motion models in our method allows better dealing with these

[3] A mismatch is a mistaken switch of tracker identifiers.

Dataset	Algorithm	Correct Tracks	MOTP	FN	FP	Id Sw.	MOTA
ExpSeq	MMPF+	**99.57%**	97.74% / **98.05%***	**0.21%**	0.27%	**9**	**99.52%**
	MMPF [26]	99.36%	97.55%	0.23%	0.34%	9	99.42%
	PF [10]	99.15%	96.11%	0.24%	0.36%	13	99.39%
	IMM [7]	99.57%	98.05%	0.30%	**0.19%**	32	99.49%
	DET [19]	99.57%	97.00%	0.30%	0.21%	34	99.47%
NumSeq	MMPF+	**98.50%**	97.38% / 97.60%*	**0.14%**	0.61%	**85**	**99.24%**
	MMPF [26]	98.50%	97.21%	0.15%	0.66%	111	99.18%
	PF [10]	97.59%	96.40%	0.15%	0.66%	91	99.19%
	IMM [7]	98.05%	97.60%	0.16%	0.59%	144	99.23%
	DET [19]	98.05%	**97.62%**	0.17%	**0.59%**	145	99.23%

Table 1 Tracking evaluation results for the five algorithms (our MMPF+, the MMPF from [26], the PF from [10], the IMM from [7], the DET from [19]) on both ExpSeq and NumSeq datasets (respectively an experimental and a numerical sequence). It shows the percentage of correct tracks (*i.e.* track correct when more than 95% of its length is correct), the false negative rate (FN), the false positive rate (FP), the number of mismatches (Id Sw.) and the CLEAR MOT metrics [6] such as the precision score (MOTP) and the accuracy score (MOTA). In bold the best score for each metrics. * corresponds to values where the MMPF+ algorithm was executed with 500 particles.

situations: this was one of the motivations of the MMPF+ algorithm.

In terms of false negative rates, our algorithm surpasses the others too. Indeed, in absence of an associated detection to a tracker, the DET algorithm stops immediately the corresponding tracker leading to a false negative, contrary to particle filter algorithms that estimate its position with the help of its previous behavior. Our algorithm is better than the PF method since it uses multiple adapted motion models to predict positions. And it is also better than MMPF since it estimates the motion states more closely to the real behavior of the beads.

Nevertheless, the deterministic and IMM approaches seem to be the best in terms of false positive rate. This is mainly because of the termination of trackers at borders (*i.e.* side effects). Indeed, in the particle filter algorithms, when there is no associated detection to a tracker because the bead left the field of view, the tracker tries to estimate its new state. If its estimated position is still inside the field of view, the tracker continues, resulting in a false positive. It survives like this a limited number of frames without associated detection and is then automatically terminated (see Sec. 4.3 for the survival times). On the contrary, DET and IMM algorithms immediately stop a tracker when there is no associated detection. So, in this case of side effect and in both studied datasets, it appears that these two methods are better than the particle filter methods.

Impact of detector quality. In order to study the tracking performance with different levels of detection, we varied the threshold τ_t of the transparent bead detector. It allows assessing how the detection quality influences the percentage of correct tracks and the accuracy MOTA on transparent beads. We made this analysis for the five algorithms and on both ExpSeq (Fig. 6, left) and NumSeq (Fig. 6, right) datasets.

The results show that our algorithm globally outperforms the four others in terms of accuracy for both datasets. Indeed, we can observe that according to the MOTA score, MMPF+ is the second best performing method for low thresholds (*i.e.* when more false positives) and the best performing one for high thresholds (*i.e.* when more false negatives). Only IMM is better for low thresholds because of the problem of false positives previously mentionned. The good results for high thresholds come from the good management of detection errors: quickly stopping trackers wrongly created by false detections and continuing predicting trackers without associated beads when missing detections.

Whatever the algorithm used, the tracking performances depend on the detector quality: the higher τ_t, the smaller the percentage of correct tracks. Especially, the DET and IMM algorithms without the propagation of undetected beads are the most impacted by high values of τ_t, as expected.

Motion state estimation. The motion states of trackers are not only used to set specific motion models for the particle filters. They are also used to study the behavior of trackers in each motion state. The aim is to analyze, for each motion state separately, the number of trackers at each time step, the velocity profiles, and the infiltration and concentration inside the mobile bed. For reliable analysis, we need the estimated motion states to be as close as possible to mechanical states. Therefore, we compare them against the motion states of the ground truths.

In Fig. 7, we present the F-score of the state identification obtained by our algorithm against MMPF for both ExpSeq and NumSeq. For each motion state, the F-score is a measure of accuracy of the classification in this motion state that considers both the precision[4] and recall[5]. We choose F-score instead of accuracy since accuracy yields misleading results if the dataset is unbalanced, *i.e.* when the number of samples in different classes vary greatly, as it is the case for our datasets with approximately 2 times less rolling beads and 20 times less saltating beads than resting beads.

[4] The precision is the fraction of retrieved instances that are relevant.
[5] The recall is the fraction of relevant instances that are retrieved.

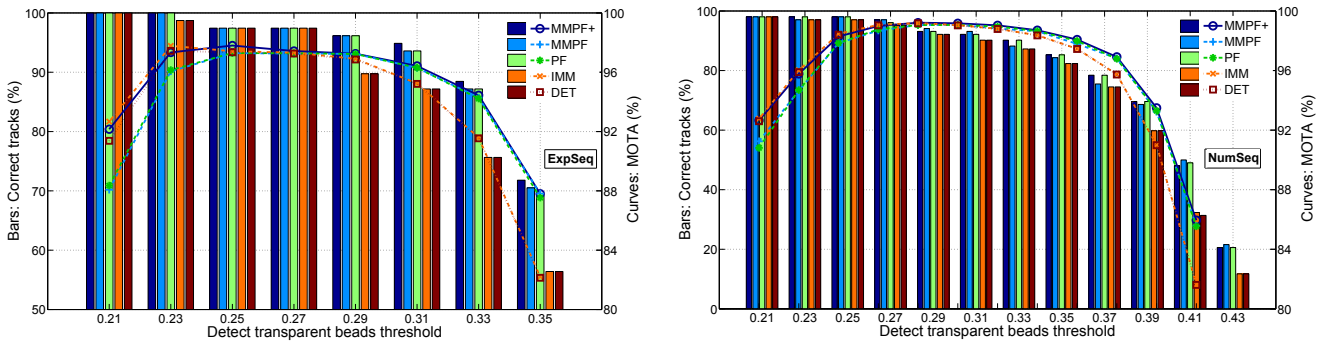


Fig. 6 Evaluation of performances of the tracking of transparent beads for the ExpSeq (left graph) and NumSeq (right graph) datasets, and for different qualities of the transparent bead detector that are controlled by the detection thresholds. The percentage of correct tracks (bars, left scales) and the accuracy score MOTA (curves, right scales) are plotted.

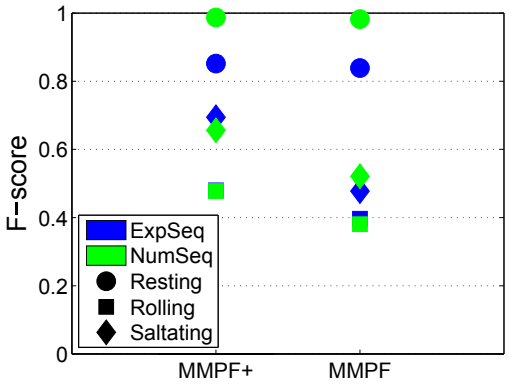


Fig. 7 Evaluation of the reliability of the estimated motion state of the trackers given by F-scores for both datasets ExpSeq and NumSeq and for the three motion states resting, rolling and saltating.

Firstly, we can observe that results are almost the same for both datasets. This confirms that the numerical dataset can be used to evaluate multiple model tracking algorithms. Secondly, the MMPF+ algorithm outperforms MMPF for all states which confirms its better ability to predict the state. In particular, the accuracy of saltating beads is increased by 0.2 for the MMPF+ algorithm especially thanks to the introduction of the two saltating sub-states 'constant' and 'rebound' (see Sec. 3.5.1). Thirdly, algorithms perform more or less well depending on the motion state. Indeed, resting beads are very well classified (0.85) compared to saltating ones (0.70) or rolling ones (0.45). This is not surprising since resting beads are easy to classify thanks to their low velocity and high number of neighbors. On the contrary, rolling beads can be confused either with resting beads when they have low velocity or with saltating beads when the are moving fast on top of the bed.

6 Conclusions

We have presented a new online particle filter algorithm based on multiple dynamic models for automatic multi-object tracking over long sequences. Our motivation was the study of bedload transport in an experimental channel with the tracking of spherical beads entrained by a water flow. The first contribution of our work is the use of several motion models to guide bead trackers with adapted predictions. Having *a priori* information about the object mechanical dynamics, we are able to better estimate object trajectories. The second contribution concerns the handling of unreliable detections, *i.e.* false positives and false negatives (missing detections). A robust bead tracking is obtained through the estimation of the detector confidence to guide particles of the filter when no discrete high-confidence detection is issued by the detector. Furthermore, we use the state of neighboring objects as an information in the likelihood of the particles to help choosing between motion states.

To evaluate the tracking performances, we carefully prepared two ground truth datasets, one created manually on an experimental sequence of images and one created numerically through a simulation reproducing the experimental conditions. When compared to state-of-the-art algorithms, our algorithm provides the highest tracking precision and accuracy. Moreover, it appears to be the least impacted by missing detections and presents less identity switches than the others when applied on detectors of different qualities. In terms of classification of the bead behaviors in motion states, our classifier outperforms another one from a similar multiple model algorithm thanks to inspection of neighbors configurations and velocities.

In our images, the main problem is to detect efficiently the transparent beads as they appear sometimes partially occluded. A possible extension would be to train the detector on several configuration problems instead of using a single appearance model of the bead. Furthermore, particles of the filter could be used in the data association step with their specific motion model for predictions. The tracking algorithm will be applied on very long sequences to observe both high and lower frequency phenomena which appear in sediment transport.

Acknowledgements This research was supported by Irstea; labex OSUG@2020; INSU programme EC2CO-BIOHEFECT and DRIL Modsed; the French national research agency project SegSed ANR-16-CE01-0005; and the Rhône-Alpes region as part of its higher education, research and innovation regional Strategy (Environment Academic Research Community).

References

1. Ali, S., Shah, M.: Floor fields for tracking in high density crowd scenes. In: European Conference on Computer Vision (ECCV), pp. 1–14. Springer (2008). doi: 10.1007/978-3-540-88688-4_1
2. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 1–8. IEEE (2008). doi: 10.1109/CVPR.2008.4587583
3. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. Signal Processing, IEEE Transactions on **50**(2), 174–188 (2002). doi: 10.1109/78.978374
4. Arulampalam, M.S., Ristic, B., Gordon, N., Mansell, T.: Bearings-only tracking of manoeuvring targets using particle filters. Journal on Advances in Signal Processing, EURASIP **2004**(15), 1–15 (2004). doi: 10.1155/S1110865704405095
5. Bacchi, V., Recking, A., Eckert, N., Frey, P., Piton, G., Naaim, M.: The effects of kinetic sorting on sediment mobility on steep slopes. Earth Surface Processes and Landforms **39**(8), 1075–1086 (2014). doi: 10.1002/esp.3564
6. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the CLEAR MOT metrics. Journal on Image and Video Processing **2008**(246309), 1–10 (2008). doi: 10.1155/2008/246309
7. Blom, H.A.P., Bar-Shalom, Y.: The interacting multiple model algorithm for systems with Markovian switching coefficients. Automatic Control, IEEE transactions on **33**(8), 780–783 (1988). doi: 10.1109/9.1299
8. Boers, Y., Driessen, J.N.: Interacting multiple model particle filter. Radar, Sonar and Navigation, IEE Proceedings **150**(5), 344–349 (2003). doi: 10.1049/ip-rsn:20030741
9. Böhm, T., Frey, P., Ducottet, C., Ancey, C., Jodeau, M., Reboud, J.L.: Two-dimensional motion of a set of particles in a free surface flow with image processing. Experiments in Fluids **41**(1), 1–11 (2006). doi: 10.1007/s00348-006-0134-9
10. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.: Online multiperson tracking-by-detection from a single, uncalibrated camera. Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on **33**(9), 1820–1833 (2011). doi: 10.1109/TPAMI.2010.232
11. Chang, D.C., Fan, M.W.: Interacting multiple model particle filtering using new particle resampling algorithm. In: Global Communications Conference (GLOBECOM), pp. 3215–3219. IEEE (2014). doi: 10.1109/glocom.2014.7037301
12. Choset, H., Nagatani, K.: Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization. Robotics and Automation, IEEE Transactions on **17**(2), 125–137 (2001). doi: 10.1109/70.928558
13. Dou, J., Li, J.: Robust visual tracking based on interactive multiple model particle filter by integrating multiple cues. Neurocomputing **135**, 118–129 (2014). doi: 10.1016/j.neucom.2013.12.049
14. Doucet, A., de Freitas, A., Gordon, N.: Sequential Monte Carlo methods in practice. Springer Science & Business Media (2001). doi: 10.1007/978-1-4757-3437-9
15. Efron, B., Tibshirani, R.J.: An introduction to the bootstrap. CRC press (1994). doi: 10.1007/978-1-4899-4541-9
16. Frey, P.: Particle velocity and concentration profiles in bedload experiments on a steep slope. Earth Surface Processes and Landforms **39**(5), 646–655 (2014). doi: 10.1002/esp.3517
17. Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. Radar and Signal Processing, IEE Proceedings F **140**(2), 107–113 (1993). doi: 10.1049/ip-f-2.1993.0015
18. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: European Conference on Computer Vision (ECCV), vol. 7575, pp. 702–715. Springer (2012). doi: 10.1007/978-3-642-33765-9_50
19. Hergault, V., Frey, P., Métivier, F., Barat, C., Ducottet, C., Böhm, T., Ancey, C.: Image processing for the study of bedload transport of two-size spherical particles in a supercritical flow. Experiments in Fluids **49**(5), 1095–1107 (2010). doi: 10.1007/s00348-010-0856-6
20. Honkanen, M., Nobach, H.: Background extraction from double-frame PIV images. Experiments in Fluids **38**(3), 348–362 (2005). doi: 10.1007/s00348-004-0916-x
21. Houssais, M., Ortiz, C.P., Durian, D.J., Jerolmack, D.J.: Onset of sediment transport is a continuous transition driven by fluid shear and granular creep. Nature communications **6**, 6527 (2015). doi: 10.1038/ncomms7527
22. Isard, M., Blake, A.: Condensation–conditional density propagation for visual tracking. International Journal of Computer Vision **29**(1), 5–28 (1998). doi: 10.1023/A:1008078328650
23. Isard, M., Blake, A.: A mixed-state condensation tracker with automatic model-switching. In: Computer Vision (ICCV), IEEE 6th International Conference on, pp. 107–112. IEEE (1998). doi: 10.1109/ICCV.1998.710707
24. Khalid, S.S., Abrar, S.: A low-complexity interacting multiple model filter for maneuvering target tracking. AEU-International Journal of Electronics and Communications **73**, 157–164 (2017). doi: 10.1016/j.aeue.2017.01.011
25. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955). doi: 10.1007/978-3-540-68279-0_2
26. Lafaye de Micheaux, H., Ducottet, C., Frey, P.: Online multi-model particle filter-based tracking to study bedload transport. In: Image Processing (ICIP), IEEE International Conference on, pp. 3489–3493. IEEE (2016). doi: 10.1109/ICIP.2016.7533008
27. Leibe, B., Schindler, K., Cornelis, N., Van Gool, L.: Coupled object detection and tracking from static cameras and moving vehicles. Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on **30**(10), 1683–1698 (2008). doi: 10.1109/TPAMI.2008.170
28. Li, X.R., Jilkov, V.P.: Survey of maneuvering target tracking. Part V. Multiple-model methods. Aerospace and Electronic Systems, IEEE Transactions on **41**(4), 1255–1321 (2005). doi: 10.1109/taes.2005.1561886
29. Lu, W.L., Okuma, K., Little, J.J.: Tracking and recognizing actions of multiple hockey players using the boosted particle filter. Image and Vision Computing **27**(1), 189–205 (2009). doi: 10.1016/j.imavis.2008.02.008
30. Maurin, R., Chauchat, J., Chareyre, B., Frey, P.: A minimal coupled fluid-discrete element model for bedload transport. Physics of Fluids (1994-present) **27**(11), 113,302 (2015). doi: 10.1063/1.4935703
31. Maurin, R., Chauchat, J., Frey, P.: Dense granular flow rheology in turbulent bedload transport. Journal of Fluid Mechanics **804**, 490–512 (2016). doi: 10.1017/jfm.2016.520
32. Maurin, R., Chauchat, J., Frey, P.: Revisiting slope influence in turbulent bedload transport: consequences for vertical flow structure and transport rate scaling. Journal of Fluid Mechanics **839**, 135–156 (2018). doi: 10.1017/jfm.2017.903

33. Mazor, E., Averbuch, A., Bar-Shalom, Y., Dayan, J.: Interacting multiple model methods in target tracking: a survey. Aerospace and Electronic Systems, IEEE Transactions on **34**(1), 103–123 (1998). doi: 10.1109/7.640267
34. McGinnity, S., Irwin, G.W.: Multiple model bootstrap filter for maneuvering target tracking. Aerospace and Electronic Systems, IEEE Transactions on **36**(3), 1006–1012 (2000). doi: 10.1109/7.869522
35. Odobez, J.M., Gatica-Perez, D., Ba, S.O.: Embedding motion in model-based stochastic tracking. Image Processing, IEEE Transactions on **15**(11), 3514–3530 (2006). doi: 10.1109/TIP.2006.877497
36. Ohmi, K., Li, H.Y.: Particle-tracking velocimetry with new algorithms. Measurement Science and Technology **11**(6), 603–616 (2000). doi: 10.1088/0957-0233/11/6/303
37. Ouellette, N.T., Xu, H., Bodenschatz, E.: A quantitative study of three-dimensional Lagrangian particle tracking algorithms. Experiments in Fluids **40**(2), 301–313 (2006). doi: 10.1007/s00348-005-0068-7
38. Rathi, Y., Vaswani, N., Tannenbaum, A., Yezzi, A.: Tracking deforming objects using particle filtering for geometric active contours. Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on **29**(8), 1470–1475 (2007). doi: 10.1109/TPAMI.2007.1081
39. Recking, A., Bacchi, V., Naaim, M., Frey, P.: Antidunes on steep slopes. Journal of Geophysical Research: Earth Surface **114**, F04,025 (2009). doi: 10.1029/2008JF001216
40. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: An experimental survey. Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on **36**(7), 1442–1468 (2014). doi: 10.1109/TPAMI.2013.230
41. Soille, P.: Morphological image analysis: Principles and Applications. Springer-Verlag Berlin Heidelberg (1999). doi: 10.1007/978-3-662-05088-0
42. Theunissen, R., Scarano, F., Riethmuller, M.: On improvement of PIV image interrogation near stationary interfaces. Experiments in Fluids **45**(4), 557–572 (2008). doi: 10.1007/s00348-008-0481-9
43. Westerweel, J., Elsinga, G.E., Adrian, R.J.: Particle image velocimetry for complex and turbulent flows. Annual Review of Fluid Mechanics **45**(1), 409–436 (2013). doi: 10.1146/annurev-fluid-120710-101204
44. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. International Journal of Computer Vision **75**(2), 247–266 (2007). doi: 10.1007/s11263-006-0027-7
45. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 2411–2418. IEEE (2013). doi: 10.1109/CVPR.2013.312
46. Zhai, Y., Yeary, M.B., Cheng, S., Kehtarnavaz, N.: An object-tracking algorithm based on multiple-model particle filtering with state partitioning. Instrumentation and Measurement, IEEE Transactions on **58**(5), 1797–1809 (2009). doi: 10.1109/tim.2009.2014511
47. Zhan, B., Monekosso, D.N., Remagnino, P., Velastin, S.A., Xu, L.Q.: Crowd analysis: a survey. Machine Vision and Applications **19**(5), 345–357 (2008). doi: 10.1007/s00138-008-0132-4
48. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 1–8. IEEE (2008). doi: 10.1109/CVPR.2008.4587584